ระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม

A Support System for Elementary Test Case Generation Based on Form-Based Requirements Specification

สิริมา นาคเตี้ย

Sirima Naktia

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science
Prince of Songkla University
2557

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

ชื่อวิทยานิพนธ์ ระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม

ผู้เขียน นางสาวสิริมา นาคเตี้ย

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

.....

(ผู้ช่วยศาสตราจารย์ ดร.อำนาจ เปาะทอง)

คณะกรรมการสอบ

.....ประธานกรรมการ

(ดร.นพมาศ ปักเข็ม)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.อำนาจ เปาะทอง)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.สุภาภรณ์ กานต์สมเกียรติ)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

.....

(รองศาสตราจารย์ ดร.ธีระพล ศรีชนะ)

คณบดีบัณฑิตวิทยาลัย

ขอรับรองว่า ผลงานวิจัยนี้มาจากการศึกษาวิจัยของนักศึกษาเอง และได้แสดงความขอบคุณบุคคลที่มีส่วนช่วยเหลือแล้ว

ลงชื่อ……...……
(ผู้ช่วยศาสตราจารย์ ดร.อำนาจ เปาะทอง)
อาจารย์ที่ปรึกษาวิทยานิพนธ์

ลงชื่อ……...……
(นางสาวสิริมา นาคเตี้ย)
นักศึกษา

ข้าพเจ้าขอรับรองว่า ผลงานวิจัยนี้ไม่เคยเป็นส่วนหนึ่งในการอนุมัติปริญญาในระดับใดมาก่อน และไม่ได้ถูกใช้ในการยื่นขออนุมัติปริญญาในขณะนี้

ลงชื่อ.....
(นางสาวสิริมา นาคเตี้ย)
นักศึกษา

ชื่อวิทยานิพนธ์ ระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม
ผู้เขียน นางสาวสิริมา นาคเตี้ย
สาขาวิชา วิทยาการคอมพิวเตอร์
ปีการศึกษา 2556

บทคัดย่อ

ในกระบวนการทดสอบซอฟต์แวร์ กรณีทดสอบถือเป็นส่วนสำคัญที่จะนำมาซึ่งซอฟต์แวร์ที่มีคุณภาพ และสามารถวางแผนการดำเนินงานได้ตั้งแต่ขั้นตอนวิศวกรรมความต้องการ โดยเฉพาะการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ การสร้างกรณีทดสอบสำหรับการตรวจรับซอฟต์แวร์ของลูกค้าหรือผู้ใช้ที่ขาดประสบการณ์เป็นเรื่องยุ่งยาก ประกอบกับผู้จัดทำข้อกำหนดคุณลักษณะของซอฟต์แวร์ส่วนใหญ่มักเป็นฝั่งของผู้พัฒนาซอฟต์แวร์ไม่ใช่ฝั่งของผู้ต้องการซอฟต์แวร์ ดังนั้นงานวิจัยนี้จึงนำเสนอกรอบแนวคิดของการสร้างกรณีทดสอบจากคุณลักษณะความต้องการในรูปแบบฟอร์ม โดยสามารถสร้างกรณีทดสอบได้โดยอัตโนมัติตามกรอบแนวคิดที่ได้นำเสนอไว้ 5 ขั้นตอน คือ (1) นิยามความต้องการ (2) กำหนดคุณลักษณะความต้องการในรูปแบบฟอร์ม (3) ระบุชนิดข้อมูลและค่าข้อมูลที่เป็นไปได้ของข้อมูลนำเข้า (4) สร้างตารางตัดสินใจ (5) สร้างกรณีทดสอบ โดยลูกค้าสามารถนำกรณีทดสอบที่สร้างขึ้นไปใช้ประโยชน์ในการตรวจรับซอฟต์แวร์ในเบื้องต้น และยังสามารถตรวจสอบความถูกต้องของซอฟต์แวร์ได้แต่เนิ่นๆ เนื่องจากกรณีทดสอบที่ได้จะสะท้อนว่าข้อกำหนดคุณลักษณะของซอฟต์แวร์ชัดเจนถูกต้องเพียงใด ระบบต้นแบบที่พัฒนาขึ้นได้มีการประเมินแบ่งเป็น 2 ส่วน คือ (1) ประเมินกรณีทดสอบที่ถูกผลิตโดยระบบ (ชุด B) เทียบกับกรณีทดสอบที่ผลิตโดยผู้เชี่ยวชาญที่เป็นนักวิเคราะห์ระบบงานคอมพิวเตอร์ จำนวน 10 ราย (ชุด A) พบว่าความตรงกันของกรณีทดสอบที่สร้างด้วยนักวิเคราะห์ระบบมีค่าเฉลี่ยคิดเป็นร้อยละ 62.81 ของกรณีทดสอบที่สร้างด้วยระบบที่พัฒนาขึ้น และเวลาที่ใช้ในการสร้างกรณีทดสอบโดยระบบลดลงโดยเฉลี่ยคิดเป็นร้อยละ 51 ระบบประมวลผลของเครื่องฝาก – ถอนเงินอัตโนมัติอย่างง่ายถูกเลือกใช้เป็นกรณีตัวอย่างในการประเมินส่วนนี้ (2) ประเมินความพึงพอใจของผู้ใช้กลุ่มต่างๆ ที่มีต่อระบบโดยผู้เข้าร่วมทดลองจำนวน 33 ราย ผลการประเมินจากผู้ใช้งานระบบมีความพึงพอใจต่อการใช้งานระบบโดยรวมอยู่ในระดับมาก

Thesis Title	A Support System for Elementary Test Case Generation Based on Form-Based Requirements Specification
Author	Miss Sirima Naktia
Major Program	Computer Science
Academic Year	2013

ABSTRACT

In the process of software testing, test cases are important to meet quality software and can be planed from the early phase at requirements engineering, especially test cases for acceptance testing. To create test cases for acceptance testing is quite difficult for an inexperienced customer. Software specifications preparation is the duty of developers. This process is less concerned by most customers. Therefore, this research proposes the framework and system for generating test cases based on form-based requirements specification. The system can generate test cases automatically on a concept that is presented in five steps: (1) Generating requirements definition (2) Generating form-based requirements specification (3) Generating possible value of input (4) Generating a decision table (5) Generating test cases. The generated test cases can be primarily used by a customer for acceptance testing. Moreover, this saves time and the cost of the software development since the generated test cases can feedback the clarity of requirements specification from the early phase. The prototype system was evaluated as two parts: (1) Test cases evaluation was performed by the ten selected analysts (set A) comparing to test cases generated by the system (set B). The Simple Automated Teller Machines (ATM) System was selected as a case study for benchmarking. Two criteria were considered as the results of this evaluation: (a) the conformation of test cases between the two sets, A and B. (b) the time used by analysts comparing to the controlled time used by the system. The first criterion shows that the average conformation is 62.81% and the average time used by analysts over the system is 51%. (2) The system evaluation was performed by thirty – three subjects acting as each group of users. The overall result shows that the system is rated as good quality.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความช่วยเหลือและสนับสนุนจากบุคคลหลายฝ่าย ผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง ดังนี้

ผู้ช่วยศาสตราจารย์ ดร.อำนาจ เปาะทอง อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่กรุณาให้ความรู้ คำแนะนำ และช่วยเหลือในการแก้ปัญหาต่างๆ ให้แก่ผู้วิจัยเสมอมา พร้อมทั้งตรวจทานและแก้ไขวิทยานิพนธ์ให้แก่ผู้วิจัย

ดร.นพมาศ ปักเข็ม ประธานกรรมการสอบวิทยานิพนธ์ ที่กรุณาให้ข้อเสนอแนะและช่วยตรวจทานแก้ไขวิทยานิพนธ์ให้มีความถูกต้องสมบูรณ์

ผู้ช่วยศาสตราจารย์ ดร.สุภาภรณ์ กานต์สมเกียรติ กรรมการในการสอบวิทยานิพนธ์ ที่กรุณาให้ข้อเสนอแนะและช่วยตรวจทานแก้ไขวิทยานิพนธ์ให้มีความถูกต้องสมบูรณ์

อาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ทุกท่านที่ให้ความรู้ทางด้านวิชาการ และคำแนะนำในการทำวิทยานิพนธ์

นักศึกษาปริญญาโทที่ช่วยเป็นผู้ประเมินกรณีทดสอบนำร่อง นักวิเคราะห์ระบบและผู้เข้าร่วมทดลองประเมินระบบในห้องปฏิบัติการในฐานะผู้ใช้กลุ่มต่างๆ ทุกท่าน ตลอดถึงผู้เชี่ยวชาญที่ช่วยประเมินการยอมรับได้ของกรณีทดสอบที่สร้างโดยระบบ

เจ้าหน้าที่ภาควิชาวิทยาการคอมพิวเตอร์ และเจ้าหน้าที่บัณฑิตวิทยาลัยทุกท่านที่ให้ความช่วยเหลือ และอำนวยความสะดวกเกี่ยวกับเอกสารและอุปกรณ์ต่างๆ

เพื่อนๆ พี่ๆ และน้องๆ ในภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ที่ช่วยประเมินความพึงพอใจในการใช้งานระบบ และกำลังใจในการทำวิทยานิพนธ์

คุณพ่อ คุณแม่ และครอบครัว ที่สนับสนุนและให้กำลังใจแก่ผู้วิจัยเสมอมา

ผู้วิจัยขอขอบคุณทุกท่านเป็นอย่างสูงมา ณ โอกาสนี้

สิริมา นาคเตี้ย

สารบัญ

หน้า

สารบัญ ..... (8)
รายการตาราง ..... (12)
รายการภาพประกอบ ..... (14)
บทที่
1 บทนำ 1
1.1 ความสำคัญและที่มาของงานวิจัย 1
1.2 งานวิจัยที่เกี่ยวข้อง 2
1.3 วัตถุประสงค์ของงานวิจัย 3
1.4 ขอบเขตการดำเนินงานของการวิจัย 3
1.5 ขั้นตอนและระยะเวลาการดำเนินงาน 4
1.5.1 ขั้นตอนการดำเนินงาน 4
1.5.2 ระยะเวลาการดำเนินงาน 4
1.5.3 แผนการดำเนินการวิจัย 5
1.6 สถานที่ดำเนินงาน 5
1.7 เครื่องมือที่ใช้ในการดำเนินงานวิจัย 5
1.7.1 ด้านฮาร์ดแวร์ 5
1.7.2 ด้านซอฟต์แวร์ 5
1.8 ประโยชน์ที่คาดว่าจะได้รับ 6
2 แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง 7
2.1 การกระบวนการวิศวกรรมซอฟต์แวร์ 7
2.1.1 กิจกรรมพื้นฐานของกระบวนการวิศวกรรมซอฟต์แวร์ 9
2.1.2 กระบวนการวิศวกรรมความต้องการ 10
2.1.3 การกำหนดรูปแบบความต้องการและมาตรฐานที่ปรากฏ 12
2.1.4 การจัดทำพจนานุกรมข้อมูล 12
2.2 การทดสอบซอฟต์แวร์ 16
2.2.1 ระดับการทดสอบซอฟต์แวร์ 16
2.2.2 ขั้นตอนการทดสอบซอฟต์แวร์ 18
2.2.3 เทคนิคการทดสอบซอฟต์แวร์ 19
2.2.4 การกำหนดรูปแบบการสร้างกรณีทดสอบ 22

สารบัญ (ต่อ)

หน้า

2.3 แนวคิดของระบบสนับสนุนผู้ใช้ 22
3 การวิเคราะห์ระบบงาน…… 25
3.1 ปัญหาการทดสอบในการตรวจรับระบบของผู้ใช้หรือผู้ต้องการซอฟต์แวร์..... 25
3.2 การวิเคราะห์ความต้องการของระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม. 25
3.2.1 การกำหนดกลุ่มผู้เกี่ยวข้อง 26
3.2.2 รายการความต้องการ…. 26
3.2.3 นิยามความต้องการเชิงฟังก์ชันงานของระบบ 27
3.2.4 ความต้องการที่ไม่ใช่เชิงฟังก์ชันงาน..... 29
3.3 กระบวนการสร้างกรณีทดสอบเบื้องต้นจากคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม 29
3.4 ตัวอย่างการสร้างกรณีทดสอบจากคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม..... 37
3.4.1 การสร้างกรณีทดสอบจากคุณลักษณะความต้องการในรูปแบบฟอร์มของฟังก์ชันการเข้าสู่ระบบด้วยตู้เอทีเอ็ม.... 39
3.4.2 การสร้างกรณีทดสอบจากคุณลักษณะความต้องการในรูปแบบฟอร์มของฟังก์ชันการสอบถามยอดเงินคงเหลือด้วยตู้เอทีเอ็ม... 43
3.4.3 การสร้างกรณีทดสอบจากคุณลักษณะความต้องการในรูปแบบฟอร์มของฟังก์ชันการฝากเงินสดด้วยตู้เอทีเอ็ม….. 46
3.4.4 การสร้างกรณีทดสอบจากคุณลักษณะความต้องการในรูปแบบฟอร์มของฟังก์ชันการถอนเงินสดด้วยตู้เอทีเอ็ม…..... 50
4 การออกแบบระบบ 54
4.1 กรอบแนวคิดของระบบ 54
4.2 การออกแบบกระบวนการทำงานโดยรวม 57
4.3 การออกแบบโครงสร้างฐานข้อมูลของระบบ 63
4.4 การออกแบบโครงสร้างของระบบ 64
4.5 ออกแบบส่วนติดต่อกับผู้ใช้ 66
4.5.1 การเข้าใช้งานระบบ 66
4.5.2 เมนูหลักของระบบ….. 66

สารบัญ (ต่อ)

หน้า

4.5.3 การสร้างโครงงานใหม่ 68
4.5.4 การนิยามความต้องการ 68
4.5.5 การกำหนดคุณลักษณะความต้องการ ระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า และกำหนดผลลัพธ์ที่คาดหวังในตารางตัดสินใจ... 69
4.5.6 การแสดงกรณีทดสอบของโครงงานที่ต้องการ..... 71
4.5.7 การแสดงส่วนช่วยเหลือสำหรับผู้ใช้..... 73
5 การพัฒนาระบบ..... 75
5.1 เครื่องมือซอฟต์แวร์ที่ใช้พัฒนาระบบ 75
5.2 การพัฒนาระบบ 75
6 การประเมินประสิทธิภาพ 79
6.1 การประเมินประสิทธิผลของกรณีทดสอบ 79
6.1.1 ขั้นตอนการประเมินประสิทธิผลของกรณีทดสอบ 79
6.1.2 การออกแบบการประเมินกรณีทดสอบและทดลองใช้งานเบื้องต้น..... 80
6.1.3 การเลือกกลุ่มผู้ประเมิน..... 82
6.1.4 ผลการประเมินประสิทธิผลของกรณีทดสอบ..... 84
6.1.5 สรุปผลการประเมินประสิทธิผลของกรณีทดสอบ..... 84
6.1.6 ข้อสังเกตจากการประเมินประสิทธิผลของกรณีทดสอบ..... 85
6.2 การประเมินประสิทธิภาพของระบบ 86
6.2.1 การประเมินระบบแบบอิสระที่ไม่มีการควบคุม 86
6.2.2 การเลือกกลุ่มประเมินสำหรับการทดลองในสิ่งแวดล้อมที่ควบคุม..... 87
6.2.3 เครื่องมือที่ใช้ในการประเมิน..... 89
6.2.4 วิธีวิเคราะห์ข้อมูลการประเมินความพึงพอใจของผู้ใช้ระบบ 89
6.2.5 ผลการประเมินประสิทธิภาพของระบบ..... 90
6.2.6 สรุปผลการประเมินประสิทธิภาพของระบบ..... 95
6.2.7 ข้อเสนอแนะ..... 96
6.3 การดำเนินการหลังการประเมินกรณีทดสอบและการประเมินระบบ..... 96
7 บทสรุปและข้อเสนอแนะ..... 97
7.1 บทสรุป..... 97
7.2 ปัญหาและอุปสรรค 98

สารบัญ (ต่อ)

หน้า

7.3 ข้อเสนอแนะและงานในอนาคต 99
บรรณานุกรม..... 100
ภาคผนวก..... 102
ก ผลงานตีพิมพ์ในการประชุมวิชาการ ACTIS 2012..... 103
ข พจนานุกรมข้อมูล..... 110
ค แบบฟอร์มประเมินกรณีทดสอบ 113
ง โจทย์ปัญหาสำหรับการประเมินการใช้งานระบบ..... 121
จ แบบฟอร์มการประเมินการใช้งานระบบโดยรวม 127
ฉ คู่มือการใช้งานระบบ..... 130
ประวัติผู้เขียน..... 140

รายการตาราง

ตาราง		หน้า
1.1	โครงสร้างพื้นฐานของคุณลักษณะความต้องการที่อยู่ในรูปแบบฟอร์ม	3
1.2	ระยะเวลาการดำเนินการวิจัย	5
2.1	การเขียนคุณลักษณะความต้องการบนพื้นฐานแบบฟอร์มมาตรฐาน	12
3.1	ตัวอย่างการระบุค่าที่เป็นไปได้ของข้อมูลนำเข้าประเภทตัวเลข	31
3.2	ตัวอย่างการระบุค่าที่เป็นไปได้ของข้อมูลนำเข้าประเภทตัวอักษร	32
3.3	ตัวอย่างการระบุค่าที่เป็นไปได้ของข้อมูลนำเข้าประเภทข้อความ	32
3.4	ตัวอย่างการระบุค่าที่เป็นไปได้ของข้อมูลนำเข้าประเภทตรรกะ	33
5.1	รายชื่อโมดูลย่อยในโมดูลแนะนำการใช้งานในส่วนต่างๆ ของระบบ	77
5.2	รายชื่อโมดูลย่อยในโมดูลการกำหนดค่าต่างๆ ที่จำเป็นในการสร้างกรณีทดสอบเบื้องต้นให้กับระบบ	77
5.3	รายชื่อโมดูลย่อยในโมดูลสำหรับการจัดการข้อมูลต่างๆ ของผู้ใช้โดยผู้ดูแลระบบ	78
6.1	ข้อมูลนำเข้าและผลลัพธ์ที่คาดหวังจากกรณีทดสอบที่สร้างด้วยระบบของฟังก์ชันงาน Login……	81
6.2	การกำหนดคะแนนของกรณีทดสอบที่สร้างโดยระบบสำหรับแต่ละฟังก์ชันงานของระบบ ATM อย่างง่าย	81
6.3	ผลการประเมินประสิทธิผลที่ได้จากการสร้างกรณีทดสอบกับกลุ่มผู้ทดลองนำร่อง	82
6.4	ผลการประเมินประสิทธิผลที่ได้จากการสร้างกรณีทดสอบกับนักวิเคราะห์ระบบ	85
6.5	เวลาที่นักวิเคราะห์ระบบใช้ในการสร้างกรณีทดสอบด้วยมือเปรียบเทียบกับการสร้างกรณีทดสอบด้วยระบบที่พัฒนาขึ้น	86
6.6	ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านการทำงานในฐานะผู้ดูแลระบบ	91
6.7	ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านการติดต่อกับผู้ใช้งานในฐานะผู้ดูแลระบบ	91
6.8	ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านคุณค่าของระบบในฐานะผู้ดูแลระบบ……	92
6.9	ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านการทำงานในฐานะนักวิเคราะห์ระบบ	92
6.10	ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านการติดต่อกับผู้ใช้งานในฐานะนักวิเคราะห์ระบบ	93

รายการตาราง (ต่อ)

ตาราง		หน้า
6.11	ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านคุณค่าของระบบในฐานะนักวิเคราะห์ระบบ….	93
6.12	ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านการทำงานในฐานะผู้ใช้ทั่วไป	94
6.13	ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านการติดต่อกับผู้ใช้งานในฐานะผู้ใช้ทั่วไป	94
6.14	ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านคุณค่าของระบบในฐานะผู้ใช้ทั่วไป	95
ข-1	ข้อมูล User (ข้อมูลผู้ใช้)…	110
ข-2	ข้อมูล Project (ข้อมูลโครงงาน)….	110
ข-3	ข้อมูล ReqDef (ข้อมูลนิยามความต้องการ)….	111
ข-4	ข้อมูล ReqSpec (ข้อมูลคุณลักษณะความต้องการ)….	111
ข-5	ข้อมูล DecisionTable (ข้อมูลตารางตัดสินใจ)…	112
ข-6	ข้อมูล ValueInput (ข้อมูลค่าที่เป็นไปได้ข้องข้อมูลนำเข้า)…..	112

รายการภาพประกอบ

ภาพประกอบ **หน้า**

2.1 กระบวนการวิศวกรรมความต้องการ 11
2.2 คุณลักษณะของซอฟต์แวร์ตามฟังก์ชันงานที่ระบบซอฟต์แวร์จะต้องมีตาม มาตรฐาน IEEE Std 830-1998 13
2.3 แผนภาพระดับการทดสอบซอฟต์แวร์และขั้นตอนการพัฒนาซอฟต์แวร์ V-Model 16
2.4 ขั้นตอนการทดสอบซอฟต์แวร์ 18
2.5 ตัวอย่างการสร้างกรณีทดสอบเชิงฟังก์ชัน 19
2.6 ตัวอย่างโปรแกรมสำหรับสร้างกรณีทดสอบเชิงโครงสร้าง……. 20
2.7 ตัวอย่างผังงานและกราฟการไหลของงานของโปรแกรมในภาพประกอบ 2.6… 21
3.1 โครงสร้างตารางการตัดสินใจ 35
3.2 วิธีสร้างตารางการตัดสินใจของฟังก์ชัน Login ระบบ ATM 40
3.3 วิธีใส่ค่าในตารางการตัดสินใจของฟังก์ชัน Login ระบบ ATM 41
3.4 ตารางการตัดสินใจของฟังก์ชัน Login ระบบ ATM 42
3.5 วิธีใส่ค่าในตารางการตัดสินใจของฟังก์ชัน Check balance ระบบ ATM..... 45
3.6 ตารางการตัดสินใจของฟังก์ชัน Check balance ระบบ ATM..... 45
3.7 วิธีใส่ค่าในตารางการตัดสินใจของฟังก์ชัน Deposit ระบบ ATM..... 48
3.8 ตารางการตัดสินใจของฟังก์ชัน Deposit ระบบ ATM..... 48
3.9 วิธีใส่ค่าในตารางการตัดสินใจของฟังก์ชัน Withdrawal ระบบ ATM 52
3.10 ตารางการตัดสินใจของฟังก์ชัน Withdrawal ระบบ ATM 52
4.1 กรอบแนวคิดต้นแบบระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบน พื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม 55
4.2 แผนภาพบริบทของระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบน พื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม 57
4.3 แผนภาพกระแสข้อมูลระดับที่ 1 ของระบบสนับสนุนสำหรับการสร้างกรณีทดสอบ เบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม ... 58
4.4 แผนภาพกระแสข้อมูลระดับที่ 2 ของการดำเนินการสร้างนิยามความต้องการ .. 59
4.5 แผนภาพกระแสข้อมูลระดับที่ 2 ของการดำเนินการสร้างคุณลักษณะความ ต้องการที่เขียนอยู่ในรูปแบบฟอร์ม 60
4.6 แผนภาพกระแสข้อมูลระดับที่ 2 ของการดำเนินการระบุชนิดข้อมูลและค่าที่ เป็นไปได้ของข้อมูลนำเข้า 61

รายการภาพประกอบ (ต่อ)

ภาพประกอบ **หน้า**

4.7 แผนภาพกระแสข้อมูลระดับที่ 2 ของการดำเนินการสร้างตารางตัดสินใจ..... 62
4.8 แผนภาพกระแสข้อมูลระดับที่ 2 ของการดำเนินการแสดงกรณีทดสอบ 63
4.9 แผนภาพอี-อาร์ ของฐานข้อมูลระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม.... 64
4.10 โครงสร้างของระบบในภาพรวม 65
4.11 หน้าจอการเข้าใช้งานระบบ 66
4.12 หน้าจอหลักของระบบ 67
4.13 หน้าจอสำหรับสร้างโครงงานใหม่โดยนักวิเคราะห์ระบบ..... 68
4.14 หน้าจอสำหรับสร้างนิยามความต้องการโดยผู้ใช้ทั่วไปหรือนักวิเคราะห์ระบบ... 69
4.15 หน้าจอสำหรับสร้างคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม 70
4.16 หน้าจอสำหรับระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า..... 70
4.17 หน้าจอสำหรับระบุผลลัพธ์ที่คาดหวังในตารางตัดสินใจ 71
4.18 หน้าจอสำหรับแสดงกรณีทดสอบที่สร้างโดยระบบ..... 72
4.19 ตัวอย่างแฟ้มข้อมูลของกรณีทดสอบในรูป Text file..... 72
4.20 หน้าจอสำหรับแสดงคู่มือการใช้งานระบบสำหรับผู้ใช้ทั่วไป..... 73
4.21 หน้าจอสำหรับแสดงคู่มือการใช้งานระบบสำหรับนักวิเคราะห์ระบบ 74
5.1 โครงสร้างของระบบที่พัฒนาโดย Visual Studio C# 2008 Express Edition 76
6.1 ขั้นตอนการประเมินประสิทธิผลของกรณีทดสอบที่สร้างโดยระบบ 80
6.2 อัตราส่วนร้อยละของผู้เข้าร่วมประเมินประสิทธิผลของกรณีทดสอบแบ่งตามเพศ 83
6.3 อายุและอายุงานของผู้เข้าร่วมประเมินประสิทธิผลของกรณีทดสอบแต่ละคน.... 84
6.4 อัตราส่วนร้อยละของผู้เข้าร่วมประเมินประสิทธิภาพของระบบแบ่งตามเพศ 87
6.5 อายุของผู้เข้าร่วมประเมินประสิทธิภาพของระบบแต่ละกลุ่ม 87
6.6 อายุงานของผู้เข้าร่วมประเมินประสิทธิภาพของระบบแต่ละกลุ่ม 88
6.7 ระดับการศึกษาสูงสุดของผู้เข้าร่วมประเมินประสิทธิภาพของระบบแต่ละกลุ่ม .. 88
6.8 อัตราส่วนของผู้เข้าร่วมประเมินประสิทธิภาพของระบบแบ่งตามสาขาวิชาที่จบการศึกษาซึ่งเกี่ยวข้องและไม่เกี่ยวข้องกับด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ..... 88
ฉ-1 หน้าจอการเข้าใช้งานระบบ…… 130
ฉ-2 หน้าจอหลักของระบบ…..... 131

รายการภาพประกอบ (ต่อ)

ภาพประกอบ **หน้า**

ฉ-3 หน้าจอสำหรับสร้างโครงงานใหม่โดยนักวิเคราะห์ระบบ..... 132
ฉ-4 หน้าจอสำหรับสร้างนิยามความต้องการโดยผู้ใช้ทั่วไปหรือนักวิเคราะห์ระบบ... 133
ฉ-5 หน้าจอสำหรับสร้างคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม..... 134
ฉ-6 หน้าจอสำหรับระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า..... 134
ฉ-7 หน้าจอสำหรับระบุผลลัพธ์ที่คาดหวังในตารางตัดสินใจ..... 135
ฉ-8 หน้าจอสำหรับแสดงกรณีทดสอบที่สร้างโดยระบบ..... 135
ฉ-9 ตัวอย่างแฟ้มข้อมูลของกรณีทดสอบในรูป Text file..... 136
ฉ-10 หน้าจอสำหรับแสดงคู่มือการใช้งานระบบสำหรับผู้ใช้ทั่วไป..... 137
ฉ-11 หน้าจอสำหรับแสดงคู่มือการใช้งานระบบสำหรับนักวิเคราะห์ระบบ 137
ฉ-12 หน้าจอสำหรับการแก้ไขข้อมูลส่วนตัวผู้ใช้..... 138
ฉ-13 หน้าจอสำหรับการสร้างบัญชีผู้ใช้และข้อมูลผู้ใช้เบื้องต้นให้กับผู้ใช้รายใหม่..... 138
ฉ-14 หน้าจอสำหรับการสร้างรหัสผ่านใหม่ให้กับผู้ใช้รายเดิมในกรณีที่ผู้ใช้ลืม
รหัสผ่าน 139
ฉ-15 หน้าจอสำหรับการแก้ไขรายการนักวิเคราะห์ระบบที่สามารถเข้าดำเนินการกับ
แต่ละโครงงาน 139

บทที่ 2

แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

เนื้อหาในบทนี้กล่าวถึงทฤษฎีและหลักการต่างๆ โดยส่วนแรกเกี่ยวข้องกับกระบวนการวิศวกรรมซอฟต์แวร์ กระบวนการวิศวกรรมความต้องการ รวมถึงการกำหนดรูปแบบความต้องการในลักษณะแบบฟอร์ม และมาตรฐานที่ปรากฏ ในส่วนที่สองเป็นเนื้อหาเกี่ยวกับการทดสอบซอฟต์แวร์ ระดับในการทดสอบซอฟต์แวร์ กระบวนการทดสอบซอฟต์แวร์ เทคนิคการทดสอบซอฟต์แวร์ การกำหนดรูปแบบการสร้างกรณีทดสอบ และแนวคิดของระบบแนะนำ

2.1 กระบวนการวิศวกรรมซอฟต์แวร์

วัฏจักรชีวิตของการพัฒนาซอฟต์แวร์ (Software Development Life Cycle: SDLC) เป็นวงจรที่แสดงถึงกิจกรรมต่างๆ ในแต่ละขั้นตอน ตั้งแต่เริ่มต้นจนกระทั่งสิ้นสุดกระบวนการ ซึ่ง IEEE Std 1074-1995 ได้กำหนดมาตรฐานสำหรับวัฏจักรชีวิตของการพัฒนาซอฟต์แวร์ประกอบด้วยกิจกรรมดังนี้ (Peters, 2000)

- **การบริหารโครงการ (Project management activity)**

การดำเนินการในช่วงนี้ครอบคลุมกิจกรรมต่อไปนี้

- การริเริ่มโครงการ การกำหนดแผนงานกิจกรรมและงบประมาณด้านต่างๆ การควบคุม และการติดตามตรวจสอบโครงการ
- การจัดการคุณภาพของซอฟต์แวร์ เป็นกิจกรรมที่เกี่ยวข้องกับการวางแผนคุณภาพ การควบคุมคุณภาพซอฟต์แวร์ และการประกันคุณภาพให้กับโครงการในการพัฒนาซอฟต์แวร์ เพื่อสร้างความมั่นใจว่าผลิตภัณฑ์ซอฟต์แวร์ที่ได้มีคุณภาพน่าเชื่อถือ

- **การดำเนินงานก่อนการพัฒนา (Pre-development activity)**

การดำเนินการในช่วงนี้ครอบคลุมกิจกรรมต่อไปนี้

- การสำรวจแนวคิดเกี่ยวกับระบบ เป็นขั้นตอนการกำหนดขอบเขตของปัญหาที่ชัดเจน มีรายละเอียดมากขึ้น โดยการพิจารณาวัตถุประสงค์หรือนโยบายขององค์กร การพิจารณากิจกรรมต่างๆ เทียบกับแนวทางการพัฒนาซอฟต์แวร์ที่เลือก การกำหนดค่าใช้จ่ายที่สอดคล้องกับกิจกรรมและกำหนดการที่วางแผนไว้ให้ชัดเจนมากยิ่งขึ้น

- การจัดสรรระบบ เป็นการเชื่อมโยงแนวคิดที่สำรวจไว้กับกระบวนการพัฒนาซอฟต์แวร์ มักมี 3 กิจกรรมหลัก คือ (1) วิเคราะห์ฟังก์ชันงาน (2) ระบุสถาปัตยกรรมที่ใช้ และ (3) จัดสรรความต้องการด้านฮาร์ดแวร์และซอฟต์แวร์

- **การดำเนินงานในการพัฒนา (Development activity)**

การดำเนินการในช่วงนี้ครอบคลุมกิจกรรมต่อไปนี้

- การวิเคราะห์ความต้องการ โดยพิจารณาว่าระบบที่จะพัฒนาต้องให้บริการอะไรบ้าง พร้อมทั้งพิจารณาขั้นตอนการดำเนินงาน ความเสี่ยง และแผนในการทดสอบความต้องการต่างๆ

- การออกแบบเพื่อให้สอดคล้องกับความต้องการที่ได้กำหนดไว้ การกำหนดรูปแบบการทำงาน และโครงสร้างของระบบ ตลอดถึงรูปแบบของข้อมูลต่างๆ ที่จะใช้กับระบบ และรูปแบบของผลลัพธ์ที่นำไปสู่การพัฒนาขั้นตอนวิธีที่เป็นแบบสำหรับการพัฒนาโปรแกรม

- การดำเนินงานพัฒนาโปรแกรมต้นฉบับ การทดสอบซอฟต์แวร์ระดับต่างๆ ตลอดถึงการจัดทำเอกสารต่างๆ

- **การดำเนินงานหลังการพัฒนา (Post-development activity)**

การดำเนินการในช่วงนี้ครอบคลุมกิจกรรมต่อไปนี้

- การติดตั้งระบบและดำเนินการใช้งานระบบ พร้อมจัดสิ่งสนับสนุนที่ทำให้ระบบสามารถดำเนินงานได้อย่างมีประสิทธิภาพ

- การบำรุงรักษา เป็นการปรับปรุงเปลี่ยนแปลงซอฟต์แวร์ภายหลังการส่งมอบเพื่อแก้ไขข้อผิดพลาด หรือปรับปรุงซอฟต์แวร์ให้เหมาะสมกับสภาพแวดล้อมที่เปลี่ยนแปลงไป หรือขยายความต้องการใหม่ๆ เพิ่มเติมเข้าไปในระบบเดิม

- การเลิกใช้งานซอฟต์แวร์ ซึ่งผู้พัฒนาหรือผู้ดูแลระบบจะต้องมีการศึกษาถึงความเป็นไปได้ของระบบงานที่ใช้อยู่กับการเปลี่ยนแปลงต่างๆ ที่เกิดขึ้นในระบบงานธุรกิจและเทคโนโลยีปัจจุบัน ว่าระบบงานนั้นควรใช้งานต่อไป หรือยกเลิกระบบที่ได้มีการพัฒนาขึ้น เมื่อต้นทุนการบำรุงรักษาระบบสูงมาก หรือเมื่อความต้องการของผู้ใช้เปลี่ยนแปลงไปอย่างมาก ระบบงานเดิมไม่สามารถจะทำให้เป็นไปตามความต้องการของผู้ใช้ระบบได้ การยกเลิกระบบงานก็เป็นสิ่งจำเป็น

- **การดำเนินงานอื่นที่สำคัญในลักษณะบูรณาการ (Integral activity)**

การดำเนินกิจกรรมในกลุ่มนี้อาจเกิดขึ้นช่วงใดช่วงหนึ่งของการพัฒนาระบบ โดยมีกิจกรรมที่มักดำเนินการดังนี้

- Verification & Validation

Verification เป็นการตรวจสอบว่าการดำเนินงานพัฒนาระบบมีการทำงานตามที่กำหนดไว้ถูกต้องหรือไม่ ส่วน Validation เป็นการตรวจสอบว่าผลิตภัณฑ์ที่ได้สอดคล้องตามความต้องการของผู้ใช้หรือไม่ ทั้งนี้มักดำเนินการเมื่อเสร็จสิ้นการพัฒนาซอฟต์แวร์แต่ละระยะ หรือมีการผลิตผลลัพธ์ใดๆ ในกระบวนการผลิต

- การจัดการตั้งค่าซอฟต์แวร์ (Software configuration management) เนื่องจากซอฟต์แวร์เป็นระบบที่มีความซับซ้อนสูง การกำหนดค่าต้องได้รับการจัดการตามมาตรฐานและกรรมวิธีที่ถูกต้อง

- การจัดทำเอกสารการพัฒนาระบบ ทุกขั้นตอนของการพัฒนาระบบต้องมีการจัดทำเอกสารเพื่อใช้อ้างอิงเสมอ การขาดการทำเอกสารประกอบหรือเอกสารอ้างอิงมักจะส่งผลให้เกิดข้อผิดพลาดต่อระบบงานและต่อนักวิเคราะห์ระบบ เพราะการจัดทำเอกสารมักจะถูกมองข้ามไป เนื่องจากเห็นว่าการจัดทำเอกสารเป็นสิ่งที่เสียเวลา แม้กระทั่งในส่วนของโปรแกรมเอง โปรแกรมเมอร์มักจะไม่นิยมเขียนคำอธิบายการทำงานเล็กๆ น้อยๆ ว่าโปรแกรมส่วนนั้นๆ ทำอะไร เพื่ออะไร ซึ่งทำให้ยากลำบากสำหรับการกลับมาแก้ไขโปรแกรมในขั้นตอนต่างๆ และมีผลทำให้การบำรุงรักษาระบบเป็นการยากและเสียเวลา บางครั้งอาจจะไม่สามารถแก้ไขระบบได้ถึงขนาดที่จะต้องเริ่มการศึกษาวิเคราะห์ และออกแบบระบบกันใหม่ การจัดทำเอกสารในที่นี้ หมายรวมถึงการบันทึกเหตุการณ์ต่างๆ และแนวความคิดให้กับผู้ใช้ รวมทั้งข้อสรุปที่เกิดขึ้นในแต่ละขั้นตอนของการพัฒนาระบบงานด้วย

- การฝึกอบรม เมื่อมีการส่งมอบซอฟต์แวร์ให้กับผู้ใช้ ควรมีการฝึกอบรมให้กับผู้ใช้ระบบ เพื่อให้ผู้ใช้ระบบสามารถใช้ระบบงานได้อย่างมีประสิทธิภาพและถูกต้อง นอกจากนี้ ในกระบวนการพัฒนาซอฟต์แวร์อาจมีการอบรมเทคนิค วิธีการใหม่ หรือการจัดการต่างๆ ให้กับนักพัฒนาระบบและผู้เกี่ยวข้องด้วยก็ได้

2.1.1 กิจกรรมพื้นฐานของกระบวนการวิศวกรรมซอฟต์แวร์

กลุ่มกิจกรรมของการพัฒนาซอฟต์แวร์แต่ละแนวทางที่ถูกเสนอโดยนักวิทยาการคอมพิวเตอร์อาจมีความแตกต่างกันทั้งในเรื่องวิธีการและเทคนิคที่ใช้ แต่อย่างไรก็ตามแนวทางส่วนใหญ่จะมีกิจกรรมพื้นฐานดังนี้ (Sommerville, 2011)

1) การกำหนดคุณลักษณะของซอฟต์แวร์ (Software specification)

กิจกรรมช่วงนี้เป็นกิจกรรมที่จะกำหนดรายละเอียดความต้องการของระบบงานและเงื่อนไขหรือข้อบังคับของซอฟต์แวร์ที่จะถูกสร้าง

2) การพัฒนาซอฟต์แวร์ (Software development)

กิจกรรมช่วงนี้เป็นการพัฒนาหรือสร้างผลิตภัณฑ์ให้ตรงตามข้อกำหนดคุณลักษณะของซอฟต์แวร์ การพัฒนาซอฟต์แวร์ในที่นี้หมายถึงขั้นตอนการออกแบบและสร้างซอฟต์แวร์ เพื่อให้กระบวนการพัฒนาซอฟต์แวร์นั้นมีมาตรฐาน และได้ผลิตภัณฑ์ซอฟต์แวร์ที่มีคุณภาพ

3) การตรวจสอบความถูกต้องของซอฟต์แวร์ (Software validation)

ผลิตภัณฑ์ซอฟต์แวร์จะต้องได้รับการตรวจสอบความถูกต้อง เพื่อให้แน่ใจว่าซอฟต์แวร์ที่ได้รับการพัฒนาขึ้น เป็นผลิตภัณฑ์ที่ตรงกับความต้องการของลูกค้าหรือผู้ใช้งาน มีความถูกต้อง ปลอดภัย และน่าเชื่อถือต่อการใช้งาน

4) การวิวัฒนาการของซอฟต์แวร์ (Software evolution)

ระบบซอฟต์แวร์ต่างๆ มักมีปัญหาหลายๆ อย่าง หลังจากพัฒนาระบบเสร็จแล้วจึงจำเป็นอย่างยิ่งที่จะต้องมีการวางแผนและกำหนดกฎเกณฑ์ในการบำรุงรักษาอย่างสม่ำเสมอ มีการแก้ไขข้อผิดพลาด ปรับเปลี่ยนระบบตามความเหมาะสม และเพิ่มลักษณะเฉพาะใหม่ๆ ในสิ่งที่เป็นประโยชน์กับระบบ เช่น อาจมีการปรับเปลี่ยนตามเทคโนโลยีใหม่ หรือจากการขยายความต้องการของผู้ใช้ เป็นต้น

2.1.2 กระบวนการวิศวกรรมความต้องการ

วิศวกรรมความต้องการ (Requirements engineering) คือ กระบวนการทำงานอย่างมีระบบเพื่อกำหนดว่าระบบซอฟต์แวร์ที่เราต้องการพัฒนาจะต้องให้บริการอะไรบ้าง และดำเนินการภายใต้เงื่อนไขใด (Sommerville, 1998) ซึ่งประกอบด้วย 4 ขั้นตอนหลัก ดังนี้

1) การศึกษาความเป็นไปได้ (Feasibility study)

กิจกรรมช่วงนี้เป็นขั้นตอนของการรวบรวม หรือค้นหาปัญหาและความต้องการโดยรวมในเบื้องต้น กำหนดขอบเขตของปัญหาและศึกษาความเป็นไปได้ด้านต่างๆ พร้อมสร้างทางเลือกที่เป็นไปได้ และจัดทำรายงานการศึกษาความเป็นไปได้

2) การวิเคราะห์ความต้องการ (Requirements analysis)

กิจกรรมช่วงนี้เป็นการประเมินความต้องการที่รวบรวมได้ จัดลำดับความสำคัญของความต้องการ จากนั้นสร้างแบบจำลองความต้องการในระดับแนวคิด และนำเสนอผู้เกี่ยวข้องทั้งหมดให้ยอมรับในความต้องการที่ได้ หากไม่ยอมรับ ต้องมีการเจรจาต่อรอง แก้ไขและนำเสนอจนกว่าจะได้รับการยอมรับ

3) การนิยามความต้องการ (Requirements definition)

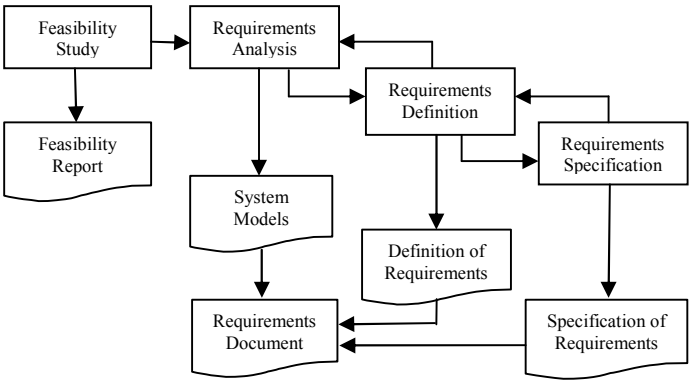
การนิยามความต้องการอาจแบ่งเป็น 2 ประเภท คือ

3.1) ความต้องการเชิงฟังก์ชัน (Functional requirements) เป็นความต้องการที่เกี่ยวกับการให้ระบบซอฟต์แวร์ทำงานตามหน้าซึ่งสอดคล้องกับระบบงานธุรกิจนั้นๆ

3.2) ความต้องการไม่เป็นเชิงฟังก์ชัน (Non-functional requirements) เป็นความต้องการซึ่งเป็นเงื่อนไขข้อกำหนดของการทำงานตามหน้าที่ต่างๆ ของระบบซอฟต์แวร์เพิ่มเติมนอกเหนือจากความต้องการเชิงฟังก์ชัน

4) การกำหนดคุณลักษณะความต้องการ (Requirements specification)

กิจกรรมช่วงนี้เป็นขั้นตอนของการขยายนิยามความต้องการแต่ละรายการให้มีรายละเอียดชัดเจน สามารถนำไปสู่การออกแบบระบบซอฟต์แวร์ได้ ซึ่งมักเรียกรายละเอียดนี้ว่าข้อกำหนดคุณลักษณะความต้องการ (Specification of requirements)



ภาพประกอบ 2.1 กระบวนการวิศวกรรมความต้องการ (Sommerville, 1998)

จากภาพประกอบ 2.1 กระบวนการวิศวกรรมความต้องการเริ่มจากการกำหนดขอบเขตและศึกษาความเป็นไปได้ของระบบ (Feasibility study) แล้วจัดทำรายงานการศึกษาความเป็นไปได้ (Feasibility report) จากนั้นวิเคราะห์ความต้องการ (Requirements analysis) จะได้แบบจำลองระบบ (System models) พร้อมการระบุความต้องการซึ่งนำมาขยายความโดยสรุป การจัดทำนิยามความต้องการ (Requirements definition) จะได้นิยามความต้องการ (Definition of requirements) ซึ่งหากเกิดปัญหาในขั้นตอนนี้ เช่น ความไม่ครบถ้วนสมบูรณ์ของความต้องการ ก็สามารถย้อนกลับไปแก้ไขขั้นตอนก่อนหน้าได้ หลังจากนั้นจะเป็นการจัดทำคำอธิบายรายละเอียดของความต้องการ (Requirements specification) ซึ่งจะได้ข้อกำหนดคุณลักษณะความต้องการ (Specification of requirements) สุดท้ายก็จะรวบรวมผลลัพธ์ทั้งหมดไว้ในเอกสารความต้องการของระบบ (Requirements document) ที่ประกอบด้วยส่วนหลักๆ ได้แก่ แบบจำลองระบบ นิยามความต้องการ และคุณลักษณะความต้องการ

2.1.3 การกำหนดรูปแบบความต้องการและมาตรฐานที่ปรากฏ

แนวทางการเขียนคุณลักษณะความต้องการที่เข้าใจง่าย ไม่ซับซ้อน มักอยู่บนพื้นฐานแบบฟอร์มมาตรฐาน ซึ่งเป็นการสร้างข้อกำหนดคุณลักษณะความต้องการ ตามรายการหัวข้อที่กำหนดเป็นส่วนๆ ชัดเจน โดยแต่ละส่วนจะปรากฏรายละเอียด สิ่งที่ต้องมีหรือต้องดำเนินการตามความต้องการของระบบ ประกอบด้วยหัวข้อที่นิยมใช้กำหนดรายละเอียดต่างๆ ดังตาราง 2.1

ตาราง 2.1 การเขียนคุณลักษณะความต้องการบนพื้นฐานแบบฟอร์มมาตรฐาน (Sommerville, 1998)

Function	ระบุชื่อฟังก์ชันงาน
Description	ระบุรายละเอียดของฟังก์ชันงานโดยสรุป
Inputs	ระบุข้อมูลนำเข้า
Source	ระบุอุปกรณ์นำเข้าข้อมูล
Outputs	ระบุข้อมูลนำออก/ผลลัพธ์
Destination	ระบุอุปกรณ์แสดงผลลัพธ์
Requires	ระบุสิ่งที่จำเป็นในการทำงานของฟังก์ชัน
Pre-condition	ระบุเงื่อนไขก่อนการดำเนินงานของฟังก์ชัน
Post-condition	ระบุเงื่อนไขหลังการดำเนินงานของฟังก์ชัน

แนวทางการอธิบายคุณลักษณะของซอฟต์แวร์ตามฟังก์ชันงานที่ระบบซอฟต์แวร์จะต้องมีสอดคล้องกับรูปแบบข้อกำหนดความต้องการด้านซอฟต์แวร์ (Software Requirements Specification: SRS) ตามมาตรฐาน IEEE Std 830-1998 มีรูปแบบดังภาพประกอบ 2.2 (IEEE, 1998)

2.1.4 การจัดทำพจนานุกรมข้อมูล

พจนานุกรมข้อมูล (Data dictionary) เป็นเอกสารอ้างอิงลักษณะหนึ่งที่มีความสำคัญยิ่งต่อการจัดการกับข้อมูลในระบบฐานข้อมูล เนื่องจากพจนานุกรมข้อมูลเป็นการผสมผสานระหว่างรูปแบบของพจนานุกรมโดยทั่วไปกับรูปแบบของข้อมูลในระบบงานคอมพิวเตอร์ เพื่อการอธิบายการใช้งานและชนิดของข้อมูลแต่ละรายการว่า เป็นตัวเลข ตัวอักขระ ข้อความหรือวันที่ เป็นต้น ทั้งนี้เพื่อตอบสนองต่อความต้องการในการอ้างอิงหรือค้นหารายละเอียดเกี่ยวกับข้อมูลทั้งหมด ความหมายของแต่ละชื่อที่ใช้ในระบบฐานข้อมูล ในการ

พัฒนาระบบตามแนวทางโดยทั่วไป มักจัดทำพจนานุกรมข้อมูลเชื่อมโยงในการอธิบายข้อมูลในแผนภาพกระแสข้อมูล (Data flow diagram) และโครงสร้างการออกแบบฐานข้อมูล เพื่อความเข้าใจตรงกันของผู้เกี่ยวข้องในกรณีที่มีการใช้ข้อมูลร่วมกัน (Kendall, 2005) การจัดทำพจนานุกรมข้อมูลประกอบด้วย 5 ส่วน ดังนี้

Specific requirements

1. External interface requirements
    - 1.1 User interfaces
    - 1.2 Hardware interfaces
    - 1.3 Software interfaces
    - 1.4 Communications interfaces
2. Functional requirements
    - 2.1 Mode 1
        - 2.1.1 Functional requirement 1.1
        - .
        - .
        - .
        - 2.1.*n* Functional requirement 1.*n*
    - 2.2 Mode 2
    - .
    - .
    - .
    - 2.*m* Mode *m*
        - 2.*m.*1 Functional requirement *m.*1
        - .
        - .
        - .
        - 2.*m.n* Functional requirement *m.n*
3. Performance requirements
4. Design constraints
5. Software system attributes
6. Other requirements

ภาพประกอบ 2.2 คุณลักษณะของซอฟต์แวร์ตามฟังก์ชันงานที่ระบบซอฟต์แวร์จะต้องมีตามมาตรฐาน IEEE Std 830-1998 (IEEE, 1998)

1) Data element

Data element คือ รายการที่แสดงรายละเอียดของฟิลด์ข้อมูลในตารางข้อมูลใดๆ โดยมีส่วนประกอบย่อยดังนี้

1.1) Data element names เป็นชื่อข้อมูล

1.2) Alternative / alias names เป็นชื่ออื่นที่ใช้เรียก Data element

1.3) Type and length เป็นชนิดและความยาวของข้อมูล

1.4) Output format เป็นรูปแบบของข้อมูลที่แสดงผล

1.5) Default value เป็นค่าที่กำหนดให้ถ้าไม่กำหนดเป็นอย่างอื่น เช่น ค่าวันที่ปัจจุบัน (Default date) หรือ Fact value (Yes / No)

1.6) Prompt / Column header เป็นหัวข้อคอลัมน์ที่แสดงออกทางหน้าจอ

1.7) Source เป็นแหล่งที่มาของข้อมูล

1.8) Security เป็นการกำหนดสิทธิ์ในการแก้ไขหรือเรียกใช้ข้อมูล

1.9) Responsible end user เป็นชื่อของผู้รับผิดชอบข้อมูล

1.10) Acceptable values เป็นการบอกค่าที่ยอมรับได้ตามเงื่อนไขในการบันทึกข้อมูล เช่น การยอมรับค่าตัวเลขใด ๆ จะต้องระบุว่า “any number”

1.11) Other validation เป็นการกำหนดวิธีการตรวจสอบข้อมูลก่อนการจัดเก็บ เช่น การตรวจสอบค่าของข้อมูลกับตารางสัมพันธ์ ถ้าข้อมูลในตารางมีค่าไม่ตรงกันกับข้อมูลที่ป้อน ระบบจะไม่บันทึกข้อมูลนั้นๆ

1.12) Derivation formula เป็นการกำหนดสูตรคำนวณ เช่น ข้อมูลที่ได้มาจากการคำนวณของฟิลด์อื่นจะต้องใส่สูตรคำนวณที่ช่องนี้

1.13) Description and comments เป็นการใส่หมายเหตุ หรือคำอธิบายเพิ่มเติม

2) Data flow

Data flow คือ เส้นทางการไหลของข้อมูล ซึ่งแสดงถึงรายละเอียดของข้อมูลหรือเอกสารที่ไหลจากส่วนหนึ่งไปยังอีกส่วนหนึ่งที่ปรากฏในผังภาพกระแสข้อมูล โดยมีส่วนประกอบย่อย ดังนี้

2.1) Data flow name เป็นชื่อของการไหลของข้อมูล

2.2) Alternative / alias names เป็นชื่ออื่นที่ใช้เรียก

2.3) Abbreviation เป็นชื่อย่อ เช่น F1

2.4) Record เป็นรายการของข้อมูล ซึ่งใช้เส้นทางการไหลของข้อมูลนี้

2.5) Description เป็นคำอธิบายการไหลของข้อมูล เช่น ใบรับของที่เป็นเอกสารที่ใช้เป็นหลักฐานในการรับสินค้าจากผู้ขาย

2.6) Origin เป็นจุดต้นทางของเส้นทาง เช่น Process 2.2 ออกใบรับของ

2.7) Destination เป็นจุดปลายทางของเส้นทาง เช่น Data store ของใบรับของ

2.8) Volume & Frequency เป็นความถี่หรือปริมาณที่จะเกิดขึ้นของข้อมูลนี้ เช่น ประมาณ 20 รายการต่อสัปดาห์

3) Data store

Data store คือ การแสดงรายละเอียดของแหล่งจัดเก็บข้อมูลที่ปรากฏในแผนภาพกระแสข้อมูล โดยมีส่วนประกอบย่อย ดังนี้

3.1) Data store name เป็นชื่อของส่วนที่ใช้จัดเก็บข้อมูล

3.2) Alternative / alias names เป็นชื่ออื่นที่ใช้เรียกแทนส่วนจัดเก็บข้อมูลนี้

3.3) Abbreviation เป็นชื่อย่อ

3.4) Record เป็นรายการของข้อมูลที่จัดเก็บในแหล่งข้อมูลนี้

3.5) Description เป็นคำอธิบายรายละเอียดของส่วนจัดเก็บข้อมูล เช่น ใบรับของ เป็นเอกสารที่ใช้เป็นหลักฐานในการรับสินค้าจากผู้ขาย

3.6) Input data flow เป็นชื่อของข้อมูลที่ไหลเข้าสู่ส่วนจัดเก็บข้อมูล

3.7) Output data flow เป็นชื่อของข้อมูลที่ไหลออกจากส่วนจัดเก็บข้อมูล

3.8) Volume & Frequency เป็นปริมาณข้อมูล และความถี่ในการทำงานกับข้อมูล

4) Process

Process คือ รายละเอียดของการประมวลผลข้อมูลที่ปรากฏในแผนภาพกระแสข้อมูล โดยมีส่วนประกอบย่อย ดังนี้

4.1) Process name เป็นชื่อของการประมวลผล เช่น ออกใบรับของ

4.2) Purpose เป็นจุดมุ่งหมายของการประมวลผลโดยย่อ เช่น เป็นการออกเอกสารหลักฐานหลังการตรวจรับสินค้า

4.3) Input data flow เป็นชื่อของข้อมูลที่ไหลเข้าสู่การประมวลผล

4.4) Output data flow เป็นชื่อของข้อมูลที่ไหลออกจากการประมวลผล

4.5) Description เป็นคำอธิบายของการประมวลผล

5) External entity

External entity คือ รายละเอียดของหน่วยงาน หรือสิ่งที่อยู่ภายนอกระบบที่ปรากฏในผังทางเดินข้อมูล โดยมีส่วนประกอบย่อย ดังนี้

5.1) External entity name เป็นชื่อสิ่งภายนอกระบบ เช่น แผนกคลังสินค้า

5.2) Alternative / alias names เป็นชื่ออื่น เช่น Store room

5.3) Acronym เป็นชื่อย่อ เช่น W

5.4) Input data flow เป็นชื่อของข้อมูลที่ไหลเข้า เช่น ใบรับของ

5.5) Output data flow เป็นชื่อของข้อมูลที่ไหลออกจาก External entity

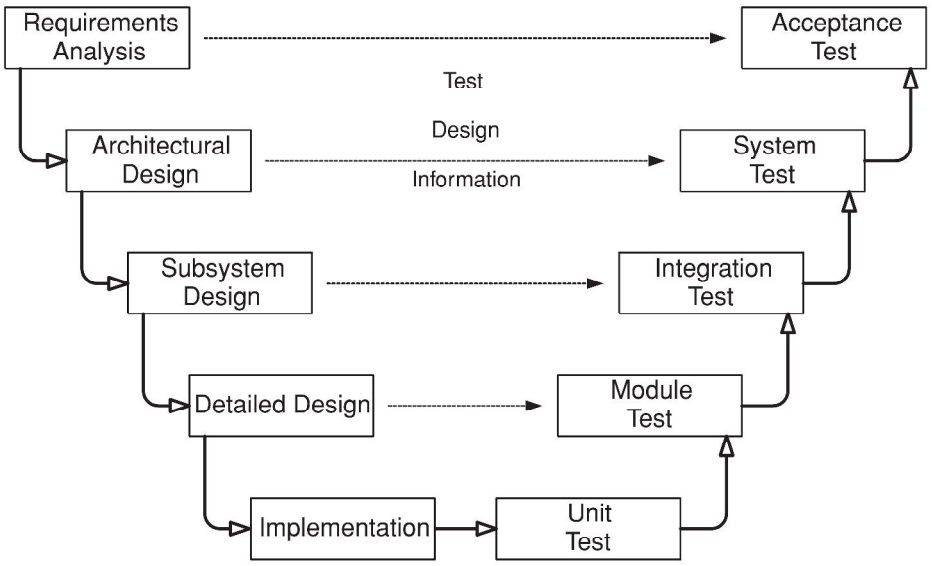
5.6) Description เป็นคำอธิบายของ External entity เช่น หน้าที่การทำงาน

2.2 การทดสอบซอฟต์แวร์

การทดสอบซอฟต์แวร์ (Software testing) เป็นกิจกรรมที่จัดทำขึ้นเพื่อประเมินและปรับปรุงคุณภาพของซอฟต์แวร์ โดยการตรวจหาข้อผิดพลาดและปัญหาที่เกิดขึ้น แล้วทำการแก้ไขข้อผิดพลาดหรือปัญหาดังกล่าวให้ถูกต้อง วัตถุประสงค์ของการทดสอบซอฟต์แวร์เป็นการพิสูจน์ว่าซอฟต์แวร์ทำงานได้ครบทุกฟังก์ชันตามคุณลักษณะความต้องการ

2.2.1 ระดับการทดสอบซอฟต์แวร์

การทดสอบซอฟต์แวร์แต่ละระดับจะทำควบคู่ไปกับขั้นตอนการพัฒนาซอฟต์แวร์ซึ่งสามารถแสดงให้เห็นถึงระดับการทดสอบที่เกี่ยวข้องกับกิจกรรมการพัฒนาซอฟต์แวร์ได้โดยการใช้แผนภาพวี (V-Model) ดังภาพประกอบ 2.3 ซึ่งสามารถแบ่งระดับการทดสอบออกตามขั้นตอนการพัฒนาซอฟต์แวร์ได้ 5 ระดับ (Ammann, 2008) ดังต่อไปนี้



ภาพประกอบ 2.3 แผนภาพระดับการทดสอบซอฟต์แวร์และขั้นตอนการพัฒนาซอฟต์แวร์ V-Model (Ammann, 2008)

1) การทดสอบระดับหน่วย (Unit testing)

การทดสอบในระดับนี้เป็นการตรวจสอบความถูกต้อง และหาข้อผิดพลาดของแต่ละส่วนโปรแกรม (Component) โดยพิจารณาถึงรหัสต้นฉบับ (Source code) ตรวจสอบความสมเหตุสมผลของกระบวนงาน ซึ่งการทดสอบระดับหน่วยนี้จะสอดคล้องกับขั้นตอนการดำเนินงาน (Implementation)

2) การทดสอบระดับโมดูล (Module testing)

การทดสอบในระดับนี้เป็นการทดสอบการทำงานแต่ละโมดูลว่าสามารถทำงานได้ถูกต้องหรือไม่ เช่น วัตถุ (Object) กระบวนงาน (Procedure) และฟังก์ชัน (Function) ซึ่งการทดสอบระดับโมดูลนี้จะสอดคล้องกับขั้นตอนการออกแบบรายละเอียด (Detailed design)

3) การทดสอบระดับรวมหน่วย (Integration testing)

การทดสอบในระดับนี้เป็นการทดสอบการทำงานของหลายโมดูลที่ทำงานร่วมกัน เพื่อหน้าที่ใดหน้าที่หนึ่ง ถึงแม้ว่าแต่ละโปรแกรมย่อยจะผ่านการทดสอบมาแล้วก็ตาม แต่เมื่อต้องมาทำงานร่วมกันอาจเกิดข้อผิดพลาดบางประการขึ้นได้ จึงต้องมีการทดสอบการรวมหน่วยเพื่อให้มั่นใจว่าไม่มีข้อผิดพลาดเกิดขึ้น จึงกล่าวได้ว่าการทดสอบระดับรวมหน่วยเป็นการทดสอบส่วนประสานการทำงานระหว่างหน่วยย่อยนั่นเอง ซึ่งสอดคล้องกับขั้นตอนการออกแบบระบบย่อย (Subsystem design)

4) การทดสอบระบบ (System testing)

การทดสอบในระดับนี้เป็นการทดสอบการทำงานของระบบทั้งระบบ ซึ่งเกิดจากการรวมกันของระบบย่อยหลายๆ ส่วน นักพัฒนาจะต้องตรวจสอบประสิทธิภาพในการทำงาน ความน่าเชื่อถือ และระบบรักษาความปลอดภัยจากผู้ไม่มีสิทธิ์เข้าใช้ระบบ นอกจากนั้นจะต้องตรวจสอบว่าสามารถทำงานได้บนเครื่องมืออุปกรณ์ที่ได้กำหนดไว้โดยไม่มีข้อผิดพลาด ซึ่งจะสอดคล้องกับขั้นตอนการออกแบบสถาปัตยกรรมของซอฟต์แวร์ (Architectural design)

5) การทดสอบเพื่อการยอมรับ (Acceptance testing)

การทดสอบในระดับนี้เป็นการทดสอบเพื่อประเมินความสมบูรณ์และประสิทธิภาพของระบบว่าสามารถทำงานได้ถูกต้องตามความต้องการและเป็นที่พึงพอใจของผู้ใช้หรือไม่ ซึ่งการทดสอบจะสอดคล้องกับขั้นตอนการวิเคราะห์ความต้องการของผู้ใช้ (Requirements analysis) และมักเป็นการทดสอบระบบในการตรวจรับของลูกค้าหรือผู้ใช้งาน สำหรับการทดสอบเพื่อการยอมรับสามารถทำได้ 2 ขั้นตอน ดังนี้

5.1) การทดสอบขั้นแอลฟา (Alpha testing)

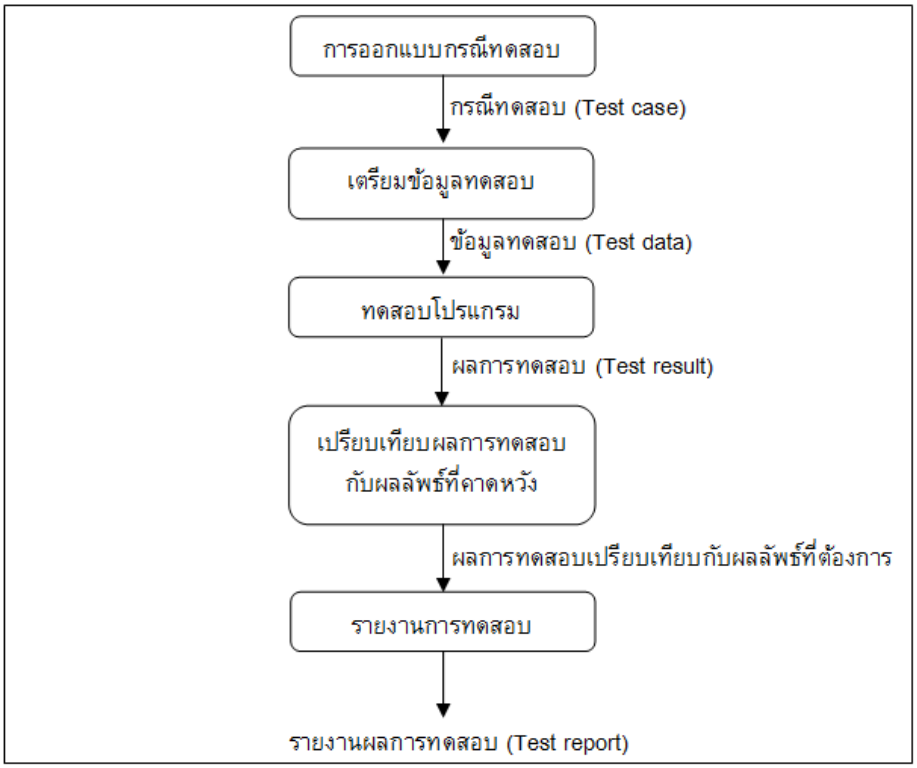
การทดสอบขั้นนี้เป็นการทดสอบการยอมรับจากผู้ผลิตหรือผู้พัฒนาโดยการจำลองสถานการณ์ กำหนดผู้ใช้ และจำลองข้อมูลให้ระบบประมวลผล แล้วพิจารณาถึงผลลัพธ์ที่ได้จากระบบนั้นว่าตรงตามความต้องการหรือไม่

5.2) การทดสอบขั้นบีตา (Beta testing)

การทดสอบขั้นนี้เป็นการทดสอบการยอมรับจากผู้ใช้ที่ทำต่อจากการทดสอบขั้นแอลฟา โดยใช้สถานที่จริงและทดสอบด้วยข้อมูลจริง ถ้ามีข้อผิดพลาดผู้ใช้จะรายงานปัญหานั้นให้กับผู้พัฒนาระบบ เพื่อจะได้ปรับปรุงแก้ไขระบบจนกว่าจะได้รับการยอมรับจากผู้ใช้หรือลูกค้า

2.2.2 ขั้นตอนการทดสอบซอฟต์แวร์

โดยทั่วไปการทดสอบซอฟต์แวร์จะเริ่มต้นจากการออกแบบ “กรณีทดสอบ” ซึ่งกำหนดข้อมูลนำเข้าและผลลัพธ์ที่คาดว่าจะได้รับจากการทำงานของโปรแกรมในสถานการณ์สมมติต่างๆ จากนั้นนำกรณีทดสอบที่ออกแบบไว้มากำหนดชุดข้อมูลทดสอบ (Test data) ให้สอดคล้องกัน เพื่อนำไปใช้ในการประมวลผลของโปรแกรม จากนั้นทำการทดสอบโดยการปฏิบัติการโปรแกรม แล้วนำผลลัพธ์ไปเปรียบเทียบกับผลลัพธ์ที่คาดหวังและสรุปเป็นรายงานการทดสอบ ซึ่งแสดงขั้นตอนการทดสอบ ได้ดังภาพประกอบ 2.4



ภาพประกอบ 2.4 ขั้นตอนการทดสอบซอฟต์แวร์

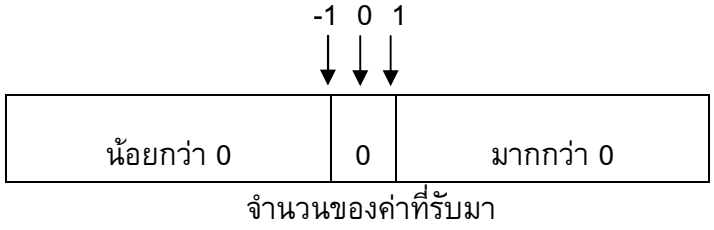
2.2.3 เทคนิคการทดสอบซอฟต์แวร์

ในการทดสอบซอฟต์แวร์แต่ละระดับนั้นจะมีเทคนิคที่ใช้แตกต่างกันตามความเหมาะสม ปัจจุบันเทคนิคที่นิยมใช้ในการทดสอบมี 2 วิธี (สกาวรัตน์, 2550) ดังนี้

1) การทดสอบเชิงฟังก์ชัน (Functional testing) เป็นการทดสอบเชิงฟังก์ชันการทำงานของระบบ โดยพิจารณาจากการรับข้อมูลเข้าสู่ระบบแล้วพิจารณาผลลัพธ์ที่ได้รับ แต่ไม่ได้พิจารณาถึงส่วนของโปรแกรม (Component) ว่ามีการทำงานอย่างไร เปรียบเสมือนเป็นกล่องดำที่ไม่ได้สนใจโครงสร้างหรือรายละเอียดภายใน บางครั้งจึงเรียกการทดสอบแบบนี้ว่า **“Black-box testing”** การทดสอบระบบด้วยวิธีนี้ ผู้ทดสอบไม่จำเป็นต้องมีความรู้เกี่ยวกับภาษาในการเขียนโปรแกรม จะมีหน้าที่ในลักษณะผู้ใช้แล้วกำหนดข้อมูลนำเข้าในลักษณะต่างๆ แล้วพิจารณาถึงผลลัพธ์ที่ได้จากระบบว่าควรเป็นอย่างไร มีกรณีใดบ้างที่ผลลัพธ์แตกต่างจากที่ได้คาดการณ์ไว้ การสร้างกรณีทดสอบมักพิจารณาจากคุณลักษณะความต้องการของแต่ละฟังก์ชันงาน ดังตัวอย่างในภาพประกอบ 2.5

ตัวอย่างการทดสอบเชิงฟังก์ชัน

โปรแกรมที่สามารถอ่านค่าจำนวนเต็มจากแป้นพิมพ์คราวละ 1 ค่า แล้วแสดงข้อความตามค่าที่รับมาออกทางจอภาพ โดยหากค่าที่รับมาน้อยกว่า 0 ให้แสดงข้อความ “Negative number” หากค่าที่รับมามากกว่า 0 ให้แสดงข้อความ “Positive number” และหากค่าที่รับมาคือ 0 ให้แสดงข้อความ “Zero”



ผลการทดสอบโปรแกรม

กรณีที่	ข้อมูลนำเข้า	ผลลัพธ์ที่คาดหวังได้	ผลการทดสอบ
1	ป้อนค่า -1	แสดงข้อความ “Negative number”	Pass
2	ป้อนค่า 1	แสดงข้อความ “Positive number”	Pass
3	ป้อนค่า 0	แสดงข้อความ “Zero”	Pass

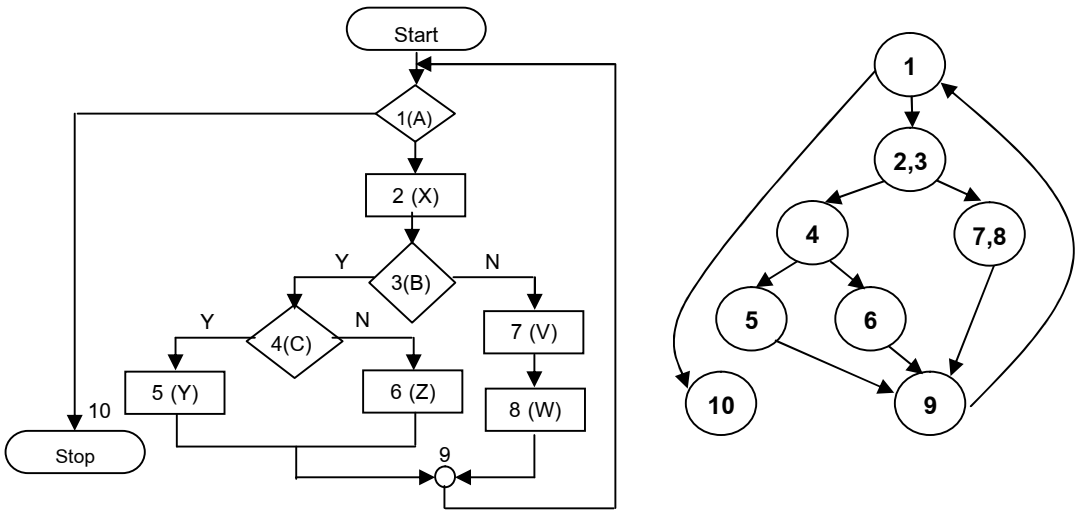
ภาพประกอบ 2.5 ตัวอย่างการสร้างกรณีทดสอบเชิงฟังก์ชัน

2) การทดสอบเชิงโครงสร้าง (Structural testing) เป็นการทดสอบที่ผู้ทดสอบต้องมีความรู้ความเข้าใจเกี่ยวกับภาษาที่ใช้สำหรับการพัฒนาระบบ เพราะจะต้องวิเคราะห์โครงสร้างของโปรแกรมว่ามีข้อผิดพลาดทางตรรกะหรือไม่ ฟังก์ชันการทำงานของโปรแกรมมีความถูกต้องและสมบูรณ์เพียงใด การทดสอบวิธีนี้เปรียบเสมือนกล่องขาวที่มุ่งสนใจถึงกระบวนการทำงานภายใน บางครั้งจึงเรียกการทดสอบแบบนี้ว่า **"White-box testing"** นอกจากนี้ยังมีการทดสอบเชิงโครงสร้างที่ใช้การตรวจสอบเส้นทางของการดำเนินงานของโปรแกรมว่ามีข้อผิดพลาดอะไรเกิดขึ้นบ้าง ที่เรียกว่า "Path testing" โดยทดสอบจากโครงสร้างควบคุมการทำงานของโปรแกรม เช่น If - then - else, While – do และ Case statement

การสร้างกรณีทดสอบสำหรับการทดสอบด้วยเทคนิคนี้ มักวิเคราะห์จากรายละเอียดของโปรแกรม หรือขั้นตอนวิธี (Algorithm) ของโปรแกรม ดังนั้นการกำหนดชุดข้อมูลเพื่อการทดสอบเชิงโครงสร้าง มักเริ่มจากการสร้างผังงาน (Flowchart) หรือ กราฟการไหลของงาน จากประโยคคำสั่งในโปรแกรม ซึ่งผังงานนั้นสามารถเป็นผังงานแสดงการทำงานของโปรแกรมหรือผังงานแสดงโครงสร้างควบคุมโปรแกรม ดังภาพประกอบ 2.7 (ก) แสดงตัวอย่างผังงานที่สอดคล้องกับตัวอย่างโปรแกรมในภาพประกอบ 2.6 ส่วนกราฟการไหลของงานเป็นแผนภาพแสดงให้เห็นโครงสร้างควบคุมของโปรแกรม โดยสัญลักษณ์ในกราฟการไหลของงานจะประกอบไปด้วยวงกลมแทนโหนด (Node) และเส้นที่เชื่อมระหว่างโหนด เรียกว่า เส้นเชื่อม (Edge) โดยทั่วไปจะแทนประโยคคำสั่งแต่ละประโยคในโปรแกรมด้วยโหนดแต่ละโหนด แล้วลากเส้นเชื่อมระหว่างโหนดตามทิศทางการทำงานของโปรแกรม ตัวอย่างกราฟการไหลของงานแสดงในภาพประกอบ 2.7 (ข)

```
Function f1()
{       (1) while A {
                (2) process X;
                (3) if B
                                (4) if C
                                (5) process Y;
                                (6) else  process Z;
                    else        (7) process V;
                                (8) process W;
        (9) } // while
} // function (10)
```

ภาพประกอบ 2.6 ตัวอย่างโปรแกรมสำหรับสร้างกรณีทดสอบเชิงโครงสร้าง (Ammann, 2008)



(ก) (ข)

ภาพประกอบ 2.7 ตัวอย่างผังงานและกราฟการไหลของงานของโปรแกรมในภาพประกอบ 2.6 (Ammann, 2008)

จากผังงานหรือกราฟการไหลของงานการกำหนดชุดข้อมูลจะเริ่มต้นจากการกำหนดเส้นทางการทดสอบโปรแกรมก่อนเพื่อกำหนดเส้นทางการทดสอบจากกราฟ จึงจะสามารถกำหนดชุดข้อมูลทดสอบของแต่ละเส้นทางได้อย่างครบถ้วนเพื่อให้ทุกๆ เส้นทางต้องผ่านการทดสอบอย่างน้อย 1 ครั้ง

จากภาพประกอบ 2.7 เมื่อกำหนดเส้นทางการทดสอบจากเส้นทางการทำงานของโปรแกรมทุกเส้นทางแล้วพบว่ามีทั้งหมด 4 เส้นทางที่จะต้องทดสอบดังนี้

1) 1, 10

2) 1, 2, 3, 4, 6, 9, 1, 10

3) 1, 2, 3, 4, 5, 9, 1, 10

4) 1, 2, 3, 7, 8, 9, 1, 10

กำหนดชุดข้อมูลทดสอบที่สอดคล้องกับทุกเส้นทาง ทำการทดสอบโดยใช้ชุดข้อมูลทดสอบแล้วนำผลที่ได้จากการทดสอบมาเปรียบเทียบกับผลลัพธ์ที่พึงประสงค์ หากผลลัพธ์ที่ได้จากการทดสอบไม่เป็นไปตามผลลัพธ์ที่พึงประสงค์แสดงให้เห็นว่าโปรแกรมจะต้องได้รับการแก้ไขให้ถูกต้อง การทดสอบเชิงโครงสร้างมีข้อดีคือ สามารถตรวจหาข้อผิดพลาดของโปรแกรมได้อย่างละเอียด เนื่องจากตรวจสอบจากภายในโปรแกรมจึงมั่นใจได้ว่าโปรแกรมจะมีความผิดพลาดน้อย

2.2.4 การกำหนดรูปแบบการสร้างกรณีทดสอบ

โดยทั่วไปการทดสอบซอฟต์แวร์จะเริ่มต้นจากการออกแบบกรณีทดสอบ ซึ่งหมายถึงข้อกำหนดของข้อมูลนำเข้าและผลลัพธ์ที่คาดว่าจะได้รับจากการทำงานของโปรแกรมในสถานการณ์สมมติต่างๆ เป็นเอกสารที่จัดทำขึ้นเพื่อไว้สำหรับใช้ในการทดสอบระบบ โดยหลักการทำกรณีทดสอบอย่างง่ายที่สุดนั้น จะต้องอยู่บนพื้นฐานของความต้องการทางธุรกิจ (Business requirements) และวัตถุประสงค์ของระบบ ในเอกสารจะระบุถึงขั้นตอนและวิธีการอย่างละเอียด เพื่อที่ผู้ที่ทำการทดสอบตามกรณีทดสอบจะสามารถทำตามได้ง่าย โดยหลักการของการสร้างกรณีทดสอบจะต้องคำนึงถึงกรณีด้านลบหรือกรณีที่ผิดพลาด (Negative case / Invalid case) และกรณีด้านบวกหรือกรณีถูกต้อง (Positive case / Valid case)

ในส่วนของเอกสารกรณีทดสอบอาจประกอบด้วยหัวข้อดังต่อไปนี้ (Williams, 2006)

1) ชื่อกรณีทดสอบ (Test case name) โดยปกติแล้วจะตั้งชื่อให้สื่อความหมาย เช่น การเข้าสู่ระบบ

2) คำอธิบายกรณีต่างๆ ว่าต้องการทดสอบกรณีใดบ้าง (Description)

3) ขั้นตอนในการทดสอบ (Step)

4) ข้อมูลนำเข้า (Input) ที่จะใช้ทดสอบในกรณีต่างๆ

5) ผลลัพธ์ที่คาดหวัง (Expected result) แสดงความคาดหวังสิ่งที่จะเป็นผลลัพธ์ของการดำเนินการทดสอบซึ่งจะต้องตรงกับความต้องการ

6) ผลลัพธ์ที่ได้จริงจากการทดสอบ (Actual result) แสดงผลลัพธ์จากการทดสอบ หากการทดสอบเป็นไปตาม Expected result จะแสดงด้วย “Pass” แต่หากการทดสอบไม่เป็นไปตาม Expected result ให้แสดงด้วย “Fail” และอธิบายข้อผิดพลาด

2.3 แนวคิดของระบบสนับสนุนผู้ใช้

การพัฒนาระบบซอฟต์แวร์ให้ผู้ใช้ทั่วไปใช้งาน มักพบส่วนสนับสนุนการใช้งานระบบ โดยเฉพาะระบบซอฟต์แวร์ที่มีความซับซ้อน การคาดหวังว่าผู้ใช้มีสัญชาตญาณในการใช้ระบบได้ หรือมีการฝึกฝนการใช้ระบบให้กับผู้ใช้ทุกรายเป็นเรื่องที่เป็นไปได้ยาก ดังนั้นการมีกลไกในการแนะนำและช่วยเหลือผู้ใช้ทุกกลุ่มให้สามารถใช้งานระบบได้อย่างมีประสิทธิภาพตลอดเวลา การใช้งานระบบตามที่ผู้ใช้ร้องขอโดยไม่ก้าวรุกล้ำเกินเลยจนผู้ใช้เกิดความรำคาญนั้น นับว่ามีความสำคัญยิ่งต่อการใช้ระบบของผู้ใช้รายใหม่และผู้ใช้รายเดิม

ส่วนของกลไกสำหรับการแนะนำจะช่วยเหลือสนับสนุนผู้ใช้อาจมีได้หลายแบบ ได้แก่ เอกสารอ้างอิง (Reference) การช่วยเหลือเฉพาะงาน (Task-specific help) การอธิบาย (Explanation) แบบฝึกหัด (Tutorial) คำถามที่พบบ่อย (Frequency Asked Question: FAQ) หรือแหล่งความรู้ (Knowledge resource) เป็นต้น

แนวทางการออกแบบส่วนสนับสนุนผู้ใช้อาจอยู่ในรูปแบบแนวทางสำหรับการสนับสนุน ดังนี้ (Dix, 1993)

1) ความช่วยเหลือในการใช้คำสั่ง (Command assistance)

ส่วนสนับสนุนนี้เป็นความช่วยเหลือให้ผู้ใช้ใช้คำสั่งที่ต้องการได้อย่างถูกต้อง เช่น ระบบช่วยเหลือโดยคำสั่ง Man ในระบบปฏิบัติการยูนิกซ์ (UNIX)

2) การเตือน (Prompts)

ส่วนสนับสนุนนี้เป็นการเตือนกรณีที่เกิดความผิดพลาดจากการใช้คำสั่ง หรือการเข้าถึงข้อมูลผิด เป็นต้น

3) ความช่วยเหลือในบริบทที่มีความอ่อนไหว (Context-sensitive help)

ส่วนสนับสนุนนี้เป็นการเตือนย้ำในกรณีที่มีการดำเนินงานใดที่มีความอ่อนไหว เช่น การลบข้อมูล มักมีการแจ้งเตือนให้ผู้ใช้ระวังมากขึ้น หรือการเปลี่ยนรหัสผ่านในการเข้าสู่ระบบ เป็นต้น

4) แบบฝึกหัดออนไลน์ (On-line tutorial)

ส่วนสนับสนุนนี้เป็นการให้ตัวอย่างที่ชัดเจนในการดำเนินงานบางอย่าง ที่เมื่อผู้ใช้เห็นแล้วทำให้เข้าใจการดำเนินงานได้ดียิ่งขึ้น เช่น การกรอกแบบฟอร์มในการเสียภาษีเงินได้

5) เอกสารออนไลน์ (On-line documentation)

ส่วนสนับสนุนนี้เป็นเอกสารที่สามารถให้คำแนะนำ หรืออธิบายรายละเอียดการใช้งานระบบในแง่มุมต่างๆ ได้มากยิ่งขึ้น เช่น คู่มือการใช้งานระบบ

6) วิดีโอออนไลน์ (On-line video)

ปัจจุบันมีความก้าวหน้าทางสื่อดิจิตอลมากยิ่งขึ้น ดังนั้นบางระบบอาจมีการใช้วิดีโอเพื่อสนับสนุนช่วยเหลือการใช้ระบบ โดยสามารถเรียกดูแบบออนไลน์ได้ทางระบบ

การออกแบบส่วนสนับสนุนผู้ใช้อาจอยู่ในรูปแบบที่เหมาะสมในลักษณะเมนู (Menu) หน้าต่าง (Window) หรือบทข้อความสนทนา (Text dialogue) แต่อย่างไรก็ตาม การออกแบบส่วนสนับสนุนผู้ใช้ควรคำนึงคุณลักษณะ ดังนี้ (Dix, 1993)

1) ความพร้อมสำหรับการใช้งาน (Availability)

ผู้ใช้จะต้องมีความสามารถในการเข้าถึงความช่วยเหลือได้ตลอดเวลาในระหว่างการทำงานกับระบบ โดยเฉพาะอย่างยิ่ง ผู้ใช้ไม่ต้องการให้โปรแกรมที่กำลังใช้งานอยู่ถูกปิดลง

เมื่อมีการเปิดโปรแกรมช่วยเหลือ หน้าต่างของระบบช่วยเหลือต้องเป็นอิสระจากโปรแกรมที่กำลังใช้งานอยู่ อย่างไรก็ตามหน้าต่างการทำงานหลักของระบบไม่ควรมีส่วนช่วยเหลือแสดงขึ้นมาบ่อยเกินไปโดยที่ผู้ใช้ไม่ได้กดปุ่มร้องขอ

2) ความถูกต้องและครบถ้วน (Accuracy and Completeness)

ความช่วยเหลือที่มีให้นั้นควรจะถูกต้องและครบถ้วน แต่ในยุคที่การใช้งานมีการปรับปรุงเวอร์ชันบ่อยๆ ความแตกต่างในการใช้งานอาจเกิดขึ้นในเวลาเดียวกัน ความช่วยเหลือที่มีให้จึงอาจถูกมองข้ามไปในที่สุด เช่นเดียวกับการสะท้อนให้เห็นถึงความถูกต้องของสถานะปัจจุบันของระบบช่วยเหลือควรจะสมบูรณ์และครอบคลุมทั้งระบบ ซึ่งเป็นสิ่งสำคัญมาก หากความช่วยเหลือที่ให้ สามารถถูกนำไปใช้อย่างมีประสิทธิภาพโดยผู้ใช้

3) ความคงเส้นคงวา (Consistency)

เนื่องจากผู้ใช้ต้องการความช่วยเหลือที่แตกต่างกันตามวัตถุประสงค์ของตน ซึ่งความช่วยเหลือนั้นอาจถูกสนองตอบจากส่วนต่างๆ ที่จัดให้ ที่ต้องคงเส้นคงวาสอดคล้องตรงกัน

4) ความแข็งแรง (Robustness)

ระบบช่วยเหลือมักถูกใช้โดยผู้ใช้ที่ประสบปัญหาในการใช้งานระบบ บางครั้งระบบซอฟต์แวร์ที่ใช้งานอาจล่ม หรือแสดงพฤติกรรมที่ไม่เป็นไปตามความคาดหวัง ดังนั้นระบบช่วยเหลือควรมีความแข็งแรงคงทน สามารถจัดการปัญหาและเป็นไปตามความคาดหวังของผู้ใช้

5) มีความยืดหยุ่น (Flexibility)

ระบบความช่วยเหลือจำนวนมากมีความเข้มงวดในการสร้างข้อความช่วยเหลือ โดยไม่คำนึงถึงความเชี่ยวชาญของบุคคลที่ใช้งานระบบ ระบบความช่วยเหลือที่มีความยืดหยุ่นนั้นจะช่วยให้ผู้ใช้แต่ละคนใช้งานส่วนช่วยเหลือได้ในรูปแบบที่เหมาะสมกับความต้องการของเขา ระบบความช่วยเหลือใด ๆ ควรออกแบบมาเพื่อการติดต่อสื่อสารกับผู้ใช้ มีการโต้ตอบมากขึ้น และมีความยืดหยุ่นในระดับของความช่วยเหลือที่นำเสนอ เช่น การใช้ไฮเปอร์เท็กซ์ (Hypertext) ที่สามารถให้ผู้ใช้เลือกดูและเชื่อมโยงไปยังส่วนที่ตนเองต้องการได้

6) ความไม่รุกล้ำ (Unobtrusiveness)

หลักการสุดท้ายสำหรับการออกแบบระบบความช่วยเหลือผู้ใช้นั้น ไม่ควรจะแทรกแซงผู้ใช้ขณะมีการใช้งานระบบตามปกติ เพราะถือเป็นการขัดจังหวะการทำงานของผู้ใช้ แต่หากจำเป็นจะต้องแสดงส่วนช่วยเหลือในบางขณะ ควรแยกหน้าจอออกจากการทำงานปกติ โดยแสดงส่วนช่วยเหลือแยกออกมาอีกหน้าจอหนึ่ง หรืออาจใช้การช่วยเหลือแบบชาญฉลาด (Intelligent help) ที่จัดการช่วยเหลืออัตโนมัติในสถานการณ์ที่จำเป็นโดยผู้ใช้ไม่ต้องร้องขอ

บทที่ 3

การวิเคราะห์ระบบงาน

เนื้อหาในบทนี้กล่าวถึงการสร้างกรณีทดสอบจากคุณลักษณะความต้องการที่เขียนอยู่ในรูปของแบบฟอร์ม ซึ่งส่วนแรกจะกล่าวถึงกรอบแนวคิดของหลักการที่นำเสนอและขั้นตอนต่าง ๆ ในการสร้างกรณีทดสอบ จากนั้นจะนำเสนอตัวอย่างการสร้างกรณีทดสอบตามหลักการที่นำเสนอ และท้ายสุดจะนำเสนอการวิเคราะห์ความต้องการของระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม

3.1 ปัญหาการทดสอบในการตรวจรับระบบของผู้ใช้หรือผู้ต้องการซอฟต์แวร์

ปัญหาที่มักพบในการตรวจรับระบบซอฟต์แวร์ของผู้ใช้หรือผู้ต้องการซอฟต์แวร์อาจพอสรุปได้ดังนี้

1) การสร้างกรณีทดสอบสำหรับการตรวจรับซอฟต์แวร์ของลูกค้าหรือผู้ใช้ที่ขาดประสบการณ์เป็นเรื่องที่ยุ่งยาก จะพบข้อผิดพลาดในซอฟต์แวร์เมื่อใช้ผลิตภัณฑ์ไปสักระยะหนึ่งแล้ว หรือค่อนข้างนาน ซึ่งบางครั้งอาจสิ้นสุดการรับประกันซอฟต์แวร์ไปแล้ว

2) สำหรับบางหน่วยงานอาจใช้นักวิชาการคอมพิวเตอร์ หรือผู้มีความรู้ทางคอมพิวเตอร์เป็นกรรมการตรวจรับระบบซอฟต์แวร์ ซึ่งบางครั้งขาดความเข้าใจในการสร้างกรณีทดสอบ และอาจไม่ได้เป็นผู้ใช้ซอฟต์แวร์โดยตรงจึงมีความเข้าใจในตัวระบบงานเชิงธุรกิจไม่มากเท่าที่ควร

3) ผู้จัดทำข้อกำหนดคุณลักษณะของซอฟต์แวร์ส่วนใหญ่มักเป็นฝั่งของผู้พัฒนาซอฟต์แวร์ ไม่ใช่ฝั่งของผู้ต้องการซอฟต์แวร์

3.2 การวิเคราะห์ความต้องการสำหรับระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม

3.2.1 การกำหนดกลุ่มผู้เกี่ยวข้อง

กลุ่มผู้ที่เกี่ยวข้องกับระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม แบ่งออกเป็น 3 กลุ่ม ดังนี้

1) ผู้ใช้ทั่วไป ซึ่งเป็นกลุ่มผู้ใช้งานที่มีความต้องการกรณีทดสอบ เพื่อใช้ในการตรวจรับซอฟต์แวร์จากนักพัฒนาว่าตรงตามความต้องการของผู้ใช้หรือไม่

2) นักวิเคราะห์ระบบ ซึ่งเป็นกลุ่มผู้ใช้งานเฉพาะทาง ที่มีความรู้ความเข้าใจในการวิเคราะห์ระบบงานคอมพิวเตอร์เป็นอย่างดี มักเป็นผู้ที่เกี่ยวข้องในช่วงวิศวกรรมความต้องการของกระบวนการพัฒนาระบบซอฟต์แวร์

3) ผู้ดูและระบบ ซึ่งเป็นผู้ใช้งานที่คอยสร้างและแก้ไขข้อมูลที่จำเป็นให้กับผู้ใช้ระบบ

3.2.2 รายการความต้องการ

การวิเคราะห์ความต้องการของระบบในเบื้องต้นได้ดำเนินการตามกลุ่มผู้เกี่ยวข้องกับระบบ โดยสามารถแสดงรายการความต้องการ (Requirements list) ตามกลุ่มผู้ใช้ระบบกลุ่มต่างๆ ได้ดังนี้

1) ผู้ใช้ทั่วไป

ผู้ใช้ทั่วไปจะเป็นใครก็ได้ที่มีความต้องการกรณีทดสอบ แต่ขาดความรู้ความชำนาญในการสร้างกรณีทดสอบ โดยใช้นิยามความต้องการเป็นข้อมูลนำเข้าเบื้องต้นในการสร้างกรณีทดสอบเบื้องต้น เพื่อนำไปตรวจรับซอฟต์แวร์ที่ได้รับการส่งมอบว่าตรงตามความต้องการของผู้ใช้หรือไม่ ซึ่งมีส่วนรายการความต้องการในภาพรวมโดยทั่วไป ดังนี้

1.1) การเข้าใช้งานระบบ

1.2) การแนะนำวิธีการใช้งานระบบและส่วนช่วยเหลือสนับสนุน

1.3) การนิยามความต้องการ

1.4) การดูผลกรณีทดสอบที่ระบบสร้างให้

1.5) การจัดการข้อมูลส่วนตัว

1.6) การออกจากระบบ

2) นักวิเคราะห์ระบบ

นักวิเคราะห์ระบบถือได้ว่ามีบทบาทสำคัญอย่างยิ่ง ในการที่จะได้มาซึ่งกรณีทดสอบที่สอดคล้องกับความต้องการของผู้ใช้ ซึ่งมีส่วนรายการความต้องการในภาพรวมโดยทั่วไป ดังนี้

2.1) การเข้าใช้งานระบบ

2.2) การแนะนำวิธีการใช้งานระบบและส่วนช่วยเหลือสนับสนุน

2.3) การสร้างโครงงานใหม่

2.4) การนิยามความต้องการ

2.5) การกำหนดคุณลักษณะความต้องการในรูปแบบฟอร์ม

2.6) การระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า

2.7) การเลือกข้อมูลนำออกหรือผลลัพธ์ที่เป็นไปได้ของแต่ละกรณี

2.8) การดูผลกรณีทดสอบที่ระบบสร้างให้

2.9) การจัดการข้อมูลส่วนตัว

2.10) การออกจากระบบ

3) ผู้ดูแลระบบ

ผู้ดูแลระบบมีหน้าที่ในการจัดการข้อมูลให้กับผู้ใช้ระบบกลุ่มอื่นๆ ซึ่งมีส่วนรายการความต้องการในภาพรวมโดยทั่วไป ดังนี้

3.1) การเข้าใช้งานระบบ

3.2) การสร้างบัญชีผู้ใช้

3.3) การสร้างรหัสผ่านใหม่ให้กับผู้ใช้รายเดิม

3.4) การแก้ไขรายการนักวิเคราะห์ระบบที่สามารถเข้าดำเนินการกับแต่ละโครงงาน

3.5) การออกจากระบบ

3.2.3 นิยามความต้องการเชิงฟังก์ชันงานของระบบ

จากการวิเคราะห์ความต้องการของระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม สามารถนิยามความต้องการของระบบ (Functional requirements definition) ดังนี้

1) การเข้าใช้งานระบบ

ฟังก์ชันการเข้าใช้งานระบบ เป็นฟังก์ชันงานที่ใช้ในการเข้าใช้งานระบบโดยการป้อนรหัสผู้ใช้ระบบและรหัสผ่าน

2) การแนะนำวิธีการใช้งานระบบและส่วนช่วยเหลือสนับสนุน

ฟังก์ชันการแนะนำวิธีการใช้งานระบบและส่วนช่วยเหลือสนับสนุน เป็นฟังก์ชันงานสำหรับให้ผู้ใช้ได้ทำความเข้าใจถึงวิธีการทำงานของระบบ ว่ามีการใช้งานอย่างไร อธิบายถึงวิธีการทำงานในขั้นตอนต่างๆ ที่ผู้ใช้ระบบจะต้องเรียนรู้ก่อนการใช้งานระบบ และตัวอย่างประกอบในแต่ละขั้นตอนการทำงาน

3) การสร้างโครงงานใหม่

ฟังก์ชันการสร้างโครงงานใหม่ เป็นฟังก์ชันงานสำหรับให้นักวิเคราะห์ระบบใช้สร้างโครงงานของระบบใหม่ที่ต้องการ ถือเป็นขั้นตอนแรกที่ต้องทำในการสร้างกรณีทดสอบ

4) การนิยามความต้องการ

ฟังก์ชันการนิยามความต้องการ เป็นฟังก์ชันงานสำหรับให้ผู้ใช้ทั่วไปหรือนักวิเคราะห์ระบบนิยามความต้องการ ซึ่งผู้ใช้จะต้องระบุชื่อฟังก์ชันงาน และคำอธิบายการทำงานของฟังก์ชันงาน ทั้งนี้ต้องผ่านการดำเนินงานในขั้นตอนการสร้างโครงงานใหม่มาก่อน ผู้ใช้จึงจะสามารถดำเนินงานในขั้นตอนนี้ได้

5) การกำหนดคุณลักษณะความต้องการในรูปแบบฟอร์ม

ฟังก์ชันการกำหนดคุณลักษณะความต้องการในรูปแบบฟอร์ม เป็นฟังก์ชันงานสำหรับให้นักวิเคราะห์ระบบกำหนดคุณลักษณะความต้องการในรูปแบบฟอร์ม ซึ่งนักวิเคราะห์จะต้องระบุคำอธิบายขั้นตอนในการดำเนินงานของฟังก์ชัน ข้อมูลนำเข้า อุปกรณ์นำเข้าข้อมูล ผลลัพธ์ที่ต้องการจะได้หลังจากที่การดำเนินงานของฟังก์ชันเสร็จสิ้น อุปกรณ์แสดงผลลัพธ์ สิ่งที่จำเป็นในการทำงานของฟังก์ชัน เงื่อนไขก่อนดำเนินการของฟังก์ชัน และเงื่อนไขหลังการดำเนินงานของฟังก์ชัน โดยขั้นตอนนี้จะต้องดำเนินงานหลังจากการนิยามความต้องการ

6) การระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า

ฟังก์ชันการระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า เป็นฟังก์ชันงานสำหรับให้นักวิเคราะห์ระบบระบุชนิดของข้อมูลนำเข้าแต่ละตัว และป้อนค่าที่เป็นไปได้ของข้อมูลนำเข้า โดยขั้นตอนนี้จะต้องดำเนินงานหลังจากการกำหนดคุณลักษณะความต้องการในรูปแบบฟอร์ม

7) การเลือกข้อมูลนำออกหรือผลลัพธ์ที่เป็นไปได้ของแต่ละกรณี

ฟังก์ชันการเลือกข้อมูลนำออกหรือผลลัพธ์ที่เป็นไปได้ของแต่ละกรณี เป็นฟังก์ชันงานสำหรับให้นักวิเคราะห์ระบบทำการเลือกข้อมูลนำออกหรือผลลัพธ์ที่เป็นไปได้ของแต่ละกรณีในตารางการตัดสินใจที่สร้างขึ้นโดยระบบ โดยขั้นตอนนี้จะต้องดำเนินงานหลังจากการระบุชนิดข้อมูลนำเข้าและป้อนค่าที่เป็นไปได้ของข้อมูลนำเข้า

8) การดูผลกรณีทดสอบที่ระบบสร้างให้

ฟังก์ชันการดูผลกรณีทดสอบที่ระบบสร้างให้ เป็นฟังก์ชันงานสำหรับให้นักวิเคราะห์ระบบหรือผู้ใช้ทั่วไปทำการเลือกแสดงกรณีทดสอบของฟังก์ชันงานที่ต้องการ โดยขั้นตอนนี้จะต้องดำเนินงานหลังจากที่นักวิเคราะห์ได้เลือกข้อมูลนำออกหรือผลลัพธ์ที่เป็นไปได้ของแต่ละกรณีในตารางการตัดสินใจแล้ว

9) การจัดการข้อมูลส่วนตัว

ฟังก์ชันการจัดการข้อมูลส่วนตัว เป็นฟังก์ชันงานสำหรับแสดง และแก้ไขข้อมูลส่วนตัวและรหัสผ่านของผู้ใช้

10) การสร้างบัญชีผู้ใช้

ฟังก์ชันการสร้างบัญชีผู้ใช้ เป็นฟังก์ชันงานสำหรับให้ผู้ดูแลระบบสร้างบัญชีผู้ใช้และข้อมูลผู้ใช้เบื้องต้น

11) การสร้างรหัสผ่านใหม่ให้กับผู้ใช้รายเดิม

ฟังก์ชันการสร้างรหัสผ่านใหม่ให้กับผู้ใช้รายเดิม เป็นฟังก์ชันงานสำหรับให้ผู้ดูแลระบบสร้างรหัสผ่านใหม่ให้กับผู้ใช้ในกรณีที่ผู้ใช้ลืมรหัสผ่าน

12) การแก้ไขรายการนักวิเคราะห์ระบบที่สามารถเข้าดำเนินการกับแต่ละโครงงาน

ฟังก์ชันการแก้ไขรายการนักวิเคราะห์ระบบที่สามารถเข้าดำเนินการกับแต่ละโครงงาน เป็นฟังก์ชันงานสำหรับให้ผู้ดูแลระบบเพิ่มหรือลบจำนวนนักวิเคราะห์ระบบที่สามารถเข้าดำเนินการกับแต่ละโครงงาน

13) การออกจากระบบ

ฟังก์ชันการออกจากระบบ เป็นฟังก์ชันที่ผู้ใช้ระบบใช้ในการออกจากระบบเมื่อสิ้นสุดการใช้งาน

3.2.4 ความต้องการที่ไม่ใช่เชิงฟังก์ชันงาน

ความต้องการในส่วนนี้เป็นเงื่อนไขของผู้ต้องการระบบที่ไม่เกี่ยวข้องกับฟังก์ชันการทำงานของระบบงานเชิงธุรกิจ ตัวอย่างความต้องการที่ไม่ใช่เชิงฟังก์ชันงาน (Non-functional requirements) ในระบบนี้ ได้แก่

1) ส่วนติดต่อกับผู้ใช้ มีการเลือกใช้ตัวอักษร การใช้สี และการวางตำแหน่งของส่วนต่างๆ อย่างเหมาะสม

2) เมนูใช้งานง่ายไม่ซับซ้อน

3.3 กระบวนการสร้างกรณีทดสอบเบื้องต้นจากคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม

งานวิจัยนี้นำเสนอระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นที่อยู่บนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม เพื่อสนับสนุนนักพัฒนา

ซอฟต์แวร์และผู้ใช้หรือผู้ต้องการซอฟต์แวร์ โดยเสนอแนวคิดการดำเนินงานกับระบบที่มีรายละเอียดขั้นตอนดังนี้

ขั้นตอนที่ 1 **นิยามความต้องการ**

นิยามความต้องการในลักษณะฟังก์ชันงานโดยสรุปว่าต้องการให้ฟังก์ชันงานนั้นมีการดำเนินงานอย่างไรบ้าง ซึ่งมักให้ข้อมูลหรือจัดทำโดยผู้ใช้ทั่วไป (Customer) หรือนักวิเคราะห์ระบบ (Analyst) ประกอบด้วยหัวข้อดังนี้

1) Function: การระบุชื่อฟังก์ชันงาน จะต้องตั้งชื่อให้สื่อถึงหน้าที่ในการทำงานว่าฟังก์ชันงานนั้นทำอะไร

2) Description: การเขียนอธิบายการทำงานของฟังก์ชันงาน โดยควรระบุขั้นตอนการดำเนินงานหลักๆ โดยละเอียด เพื่อให้เกิดความเข้าใจในภาพรวมเกี่ยวกับขั้นตอนการดำเนินงาน องค์ประกอบของการทำงาน และเงื่อนไขที่สำคัญสำหรับการทำงาน

ขั้นตอนที่ 2 **กำหนดคุณลักษณะความต้องการให้อยู่ในรูปแบบฟอร์ม**

นักวิเคราะห์ระบบกำหนดคุณลักษณะความต้องการในรูปแบบฟอร์มสำหรับแต่ละนิยามความต้องการ ซึ่งประกอบด้วยหัวข้อดังนี้ (Sommerville, 1998)

1) Function: การระบุชื่อฟังก์ชันงาน จะต้องตั้งชื่อให้สื่อถึงหน้าที่ในการทำงานว่าฟังก์ชันงานนั้นทำอะไร มักนำมาจากชื่อฟังก์ชันงานของนิยามความต้องการ

2) Description: การเขียนระบุรายละเอียดของฟังก์ชันงาน ต้องระบุขั้นตอนการดำเนินงานเป็นข้อๆ โดยจะต้องสอดคล้องกับ Description ในนิยามความต้องการ

3) Inputs: การระบุข้อมูลนำเข้า ให้ระบุข้อมูลนำเข้าที่ต้องใช้ในการดำเนินงานของฟังก์ชันว่ามีข้อมูลอะไรบ้าง โดยใช้เครื่องหมายจุลภาค (,) คั่นแต่ละข้อมูล

4) Source: การระบุอุปกรณ์นำเข้าข้อมูล เช่น เมาส์ คีย์บอร์ด เป็นต้น

5) Outputs: การระบุผลลัพธ์หรือข้อมูลนำออกที่ต้องการจะได้หลังจากที่การดำเนินงานของฟังก์ชันเสร็จสิ้น โดยให้ใช้เครื่องหมายทับ (/) คั่นแต่ละข้อมูลนำออก ในกรณีที่ต้องการให้แสดงผลอย่างใดอย่างหนึ่งตามเงื่อนไขของข้อมูลนำเข้า

6) Destination: การระบุอุปกรณ์แสดงผลลัพธ์ ให้ระบุว่าเมื่อการทำงานของฟังก์ชันเสร็จสิ้นระบบจะแสดงผลลัพธ์ออกทางอุปกรณ์ใดบ้าง เช่น เครื่องพิมพ์ หน้าจอแสดงผล เป็นต้น

7) Requires: ระบุสิ่งที่จำเป็นในการทำงานของฟังก์ชัน เป็นการระบุสิ่งที่จำเป็นในการดำเนินงาน หรือสิ่งที่ผู้ใช้จำเป็นต้องทำโดยการระบุสิ่งที่ต้องการเป็นข้อๆ

8) Pre-condition: การระบุเงื่อนไขก่อนการดำเนินงานของฟังก์ชัน ให้ผู้ใช้ระบุว่าระบบต้องดำเนินงานอะไร หรือมีเงื่อนไขใดที่ต้องทำก่อนการดำเนินงานของฟังก์ชันหรือไม่ แต่หากไม่มีเงื่อนไขก่อนการดำเนินงานให้ผู้ใช้ระบุเครื่องหมายขีด (-) แทน

9) Post-condition: การระบุเงื่อนไขหลังการดำเนินงานของฟังก์ชัน เป็นการระบุว่าระบบจะต้องดำเนินการอะไรเมื่อการดำเนินงานของฟังก์ชันเสร็จสิ้น

ขั้นตอนที่ 3 **ระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า**

นักวิเคราะห์ระบบระบุชนิดข้อมูลของข้อมูลนำเข้าแต่ละตัว แล้วป้อนค่าที่เป็นไปได้ของแต่ละข้อมูลนำเข้า โดยมีรายละเอียดดังนี้

1) ชนิดข้อมูลประเภทตัวเลข (Number) นักวิเคราะห์จะต้องระบุค่าข้อมูลที่เป็นไปได้โดยคั่นแต่ละค่าของข้อมูลที่เป็นไปได้ด้วยเครื่องหมายจุลภาค (,) แล้วระบุค่าสูงสุด (Max) และค่าต่ำสุด (Min) ของข้อมูล โดยมีวิธีการระบุค่าข้อมูล 3 วิธี ดังนี้

1.1) ในกรณีที่มีค่าข้อมูลที่เป็นไปได้ แต่ไม่มีค่า Max และ Min ให้ระบุค่าข้อมูลที่เป็นไปได้ในช่อง Possible value แล้วใส่เลข 0 ในช่อง Max และ Min แสดงตัวอย่างดัง Case 1 ในตาราง 3.1

1.2) ในกรณีที่ไม่มีค่าข้อมูลที่เป็นไปได้ แต่มีค่า Max และ Min ให้ใส่เครื่องหมายขีด (-) ในช่อง Possible value แล้วใส่ค่า Max และ Min ตามต้องการ แสดงตัวอย่างดัง Case 2 ในตาราง 3.1

1.3) ในกรณีที่มีค่าข้อมูลที่เป็นไปได้ มีค่า Max และ Min ให้ระบุค่าข้อมูลที่เป็นไปได้ในช่อง Possible value แล้วใส่ค่า Max และ Min ตามต้องการ แสดงตัวอย่างดัง Case 3 ในตาราง 3.1

ตาราง 3.1 ตัวอย่างการระบุค่าที่เป็นไปได้ของข้อมูลนำเข้าประเภทตัวเลข

Case	Input	Possible value	Max	Min
1	Number1	<Number2 , =Number2 , >Number2	0	0
2	Number1	-	10000	100
3	Number1	<Number2 , =Number2 , >Number2	10000	100

2) ชนิดข้อมูลประเภทตัวอักษร (Character) นักวิเคราะห์จะต้องป้อนค่าข้อมูลที่เป็นไปได้ โดยคั่นแต่ละค่าของข้อมูลด้วยเครื่องหมายจุลภาค (,) แสดงตัวอย่างดังตาราง 3.2

ตาราง 3.2 ตัวอย่างการระบุค่าที่เป็นไปได้ของข้อมูลนำเข้าประเภทตัวอักษร

Case	Input	Possible value
1	Sex	F, M
2	Grade	A, B, C, D, E

3) ชนิดข้อมูลประเภทข้อความหรือชุดอักษร (String)

3.1) หากค่าข้อมูลที่เป็นไปได้นั้นเป็นค่าใดก็ได้เพียง 1 ค่า เช่น ที่อยู่ ภูมิลำเนา หรือข้อเสนอแนะ ให้นักวิเคราะห์ป้อนค่าที่เป็นไปได้เพียง 1 ค่านั้น โดยไม่ต้องมีเครื่องหมายจุลภาค (,) แสดงตัวอย่างดัง Case 1 ในตาราง 3.3

3.2) กรณีที่มีค่าข้อมูลที่เป็นไปได้หลายค่า ให้นักวิเคราะห์ป้อนค่าข้อมูลที่เป็นไปได้ทั้งหมด โดยคั่นแต่ละค่าของข้อมูลด้วยเครื่องหมายจุลภาค (,) แสดงตัวอย่างดัง Case 2 ในตาราง 3.3

3.3) หากข้อมูลนำเข้านั้นมีความสำคัญต่อการดำเนินงานของระบบ กล่าวคือ หากผู้ใช้ไม่ป้อนค่าให้กับข้อมูลนำเข้านั้น ระบบจะไม่สามารถดำเนินงานได้ ให้นักวิเคราะห์เพิ่มค่า null ไว้ท้ายสุดของข้อมูลนำเข้านั้น แสดงตัวอย่างดัง Case 3 ในตาราง 3.3

ตาราง 3.3 ตัวอย่างการระบุค่าที่เป็นไปได้ของข้อมูลนำเข้าประเภทข้อความ

Case	Input	Possible value
1	ระดับการศึกษาสูงสุด	ปริญญาตรี
2	ระดับการศึกษาสูงสุด	ปริญญาตรี, ปริญญาโท , ปริญญาเอก
3	ระดับการศึกษาสูงสุด	ปริญญาตรี, ปริญญาโท , ปริญญาเอก, null

4) ชนิดข้อมูลประเภทตรรกะ (Boolean) เป็นค่าข้อมูลที่ได้จากการตรวจสอบค่าความจริงของประพจน์ว่ามีค่าจริงหรือเท็จ โดยค่าที่เก็บจะมีแค่ True หรือ False เท่านั้น นักวิเคราะห์จึงไม่ต้องป้อนค่าข้อมูลที่เป็นไปได้อีก เช่น ข้อมูลนำเข้าประเภทที่ต้องนำไปตรวจสอบความถูกต้องของข้อมูลอย่าง User name และ Password จะต้องมีการตรวจสอบค่าว่า User name มีค่าถูกต้องหรือผิด Password มีค่าถูกต้องหรือผิด แสดงตัวอย่างดังตาราง 3.4

ตาราง 3.4 ตัวอย่างการระบุค่าที่เป็นไปได้ของข้อมูลนำเข้าประเภทตรรกะ

Case	Input	Possible value
1	User name	True, False
2	Password	True, False

<u>ขั้นตอนที่ 4</u> **สร้างตารางตัดสินใจ**

ในขั้นตอนนี้จะมีการทำงานร่วมกันจาก 2 ส่วน คือ ระบบจะสร้างตารางตัดสินใจ (Decision table) จากนั้นนักวิเคราะห์ระบบจะต้องทำการเลือกข้อมูลนำออกหรือผลลัพธ์ที่เป็นไปได้ของข้อมูลนำเข้าในแต่ละกรณี โดยระบบจะมีวิธีสร้างตารางตัดสินใจ ดังนี้

1) คำนวณหาจำนวนแถวของตาราง (NumRowDecisionTable) โดยระบบ ซึ่งจะมีค่าเท่ากับจำนวนข้อมูลนำเข้า (NumInput) บวกด้วยหนึ่ง

NumRowDecisionTable = NumInput + 1

2) คำนวณหาจำนวนคอลัมน์ของตาราง (NumColumnDecisionTable) โดยระบบ ซึ่งจะมีค่าเท่ากับผลคูณของค่าที่เป็นไปได้ของข้อมูลนำเข้า (NumValueInput) แต่ละตัวบวกด้วย 1

NumColumnDecisionTable = (NumValueInput [1] x
NumValueInput [2] x … x
NumValueInput [n]) + 1

โดยมีขั้นตอนการหาจำนวนค่าข้อมูลที่เป็นไปได้ของข้อมูลนำเข้า ดังนี้

2.1) ข้อมูลประเภทตัวเลขจะมีหลักในการพิจารณา ดังนี้

2.1.1) หากข้อมูลนำเข้านั้นไม่มีค่า Max และ Min จะพิจารณาจากจำนวนค่าที่เป็นไปได้ใน Possible value เช่น ตัวอย่างจากตาราง 3.1 Case 1 มีค่าที่เป็นไปได้คือ <Number2, =Number2 และ >Number2 ดังนั้นจำนวนค่าข้อมูลที่เป็นไปได้จะเท่ากับ 3

2.1.2) หากข้อมูลนำเข้านั้นไม่มีค่า Possible value จะพิจารณาจำนวนค่าที่เป็นไปจาก Max และ Min โดยใช้หลักการจะใช้หลักการการแบ่งส่วนสมมูล (Equivalence partitioning) ในการกำหนดค่าข้อมูลที่เป็นไปได้ของ Input นั้น ดังนี้

- ค่าข้อมูลต่ำสุดของพิสัยข้อมูล นั่นคือ ค่า Min
- ค่าข้อมูลสูงสุดขอพิสัยข้อมูล นั่นคือ ค่า Max
- ค่าข้อมูลตัวแทนกลุ่ม เป็นค่าที่ใกล้เคียงกับค่ากลางของพิสัยข้อมูล นั่นคือ ค่า Min + ((Max – Min) / 2)
- ค่าข้อมูลเกินพิสัย หรือเกินขอบเขตข้อมูลในแต่ละช่วง ประกอบด้วย 2 ค่า ซึ่งในที่นี้จะใช้ค่า Max บวกด้วย 1 และใช้ค่า Min ลบด้วย 1

เช่น ตัวอย่างจากตาราง 3.1 Case 2 มีค่า Max คือ 10000 และ Min คือ 100 ค่าที่เป็นไปได้โดยใช้หลักการแบ่งส่วนสมมูล ได้แก่ 99, 100, 5050, 10000 และ 10001 เพราะฉะนั้นจำนวนค่าข้อมูลที่เป็นไปได้จะเท่ากับ 5

2.1.3) หากข้อมูลนำเข้านั้นมีทั้ง Possible value ค่า Max และ Min จะพิจารณาจากจำนวนค่าที่เป็นไปได้ใน Possible value ทั้งหมด บวกด้วยค่าที่เป็นไปได้โดยใช้หลักการแบ่งส่วนสมมูลของ Max และ Min เช่น ตัวอย่างจากตาราง 3.1 Case 3 มีค่าที่เป็นไปได้ใน Possible value คือ <Number2, =Number2 และ >Number2 และค่าที่เป็นไปได้โดยใช้หลักการแบ่งส่วนสมมูล ได้แก่ 99, 100, 5050, 10000 และ 10001 ดังนั้นจำนวนค่าข้อมูลที่เป็นไปได้จะเท่ากับ 8

2.2) ข้อมูลประเภทตัวอักษรและประเภทข้อความจะพิจารณาจากจำนวนค่าที่เป็นไปได้ใน Possible value เท่านั้น เช่น ตัวอย่างจากตาราง 3.2 Case 1 มีค่าที่เป็นไปได้ คือ F และ M ดังนั้นจำนวนค่าข้อมูลที่เป็นไปได้จะเท่ากับ 2

2.3) ข้อมูลประเภทตรรกะจะมีจำนวนค่าที่เป็นไปได้ของข้อมูลนำเข้าเท่ากับ 2 เสมอ เนื่องจากค่าที่เป็นไปได้ของข้อมูลประเภทนี้มีเพียง True และ False

3) การใส่ค่าลงในตารางตัดสินใจ จะประกอบด้วย 4 ส่วน ดังนี้

3.1) ใส่เลข 1 ลงในแถวที่ 1 คอลัมน์ที่ 2 ของตารางตัดสินใจ แล้วเพิ่มค่าทีละ 1 เรื่อยไปในแถวที่ 1 จนถึงคอลัมน์สุดท้าย

3.2) ใส่ค่าของข้อมูลนำเข้าตัวที่ 1 (Input[1]) ลงในแถวที่ 2 คอลัมน์ที่ 1 ของตารางตัดสินใจ แล้วใส่ข้อมูลนำเข้าตัวถัดไปในคอลัมน์ที่ 1 เรื่อยไปจนถึงแถวสุดท้าย

3.3) ใส่ค่าต่างๆ ที่เป็นไปได้สำหรับข้อมูลนำเข้าแต่ละตัว ดังนี้

3.3.1) หาจำนวนการเขียนซ้ำของค่าข้อมูลที่เป็นไปได้ของ Input[1] (NumRepeatValueInput[1]) โดยนำจำนวนคอลัมน์ของตารางตัดสินใจลบด้วย 1 มาหารด้วยจำนวนค่าที่เป็นไปได้ของ Input[1] (NumValueInput[1])

$$\text{NumRepeatValueInput}[1] = (\text{NumColumnDecisionTable} - 1) / \text{NumValueInput}[1]$$

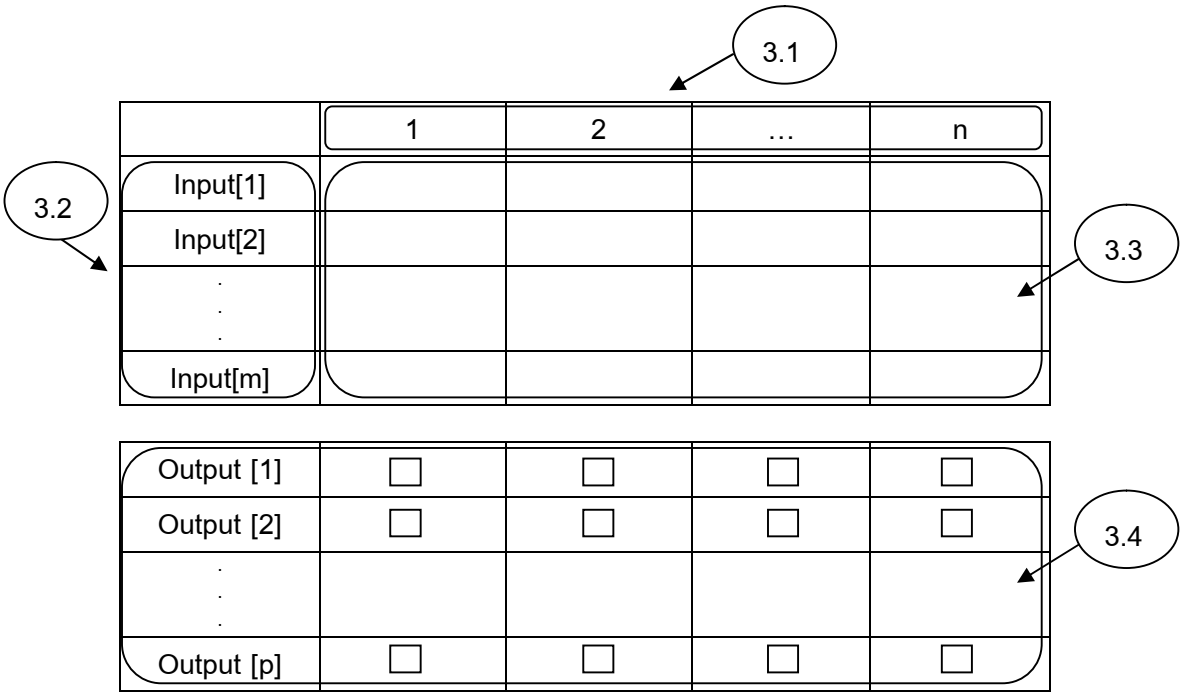
3.3.2) หาจำนวนการเขียนซ้ำของค่าข้อมูลที่เป็นไปได้ของ Input[i] (ในที่นี้ให้ i แทนจำนวนเต็มใดๆ) คำนวณจากจำนวนที่เขียนซ้ำของข้อมูลนำเข้าที่ (i-1) หารด้วยจำนวนค่าที่เป็นไปได้ของ Input[i] โดยวิธีการนี้จะนำไปใช้กับ NumRepeatValueInput[2] ถึง NumRepeatValueInput[NumColumnDecisionTable – 1]

NumRepeatValueInput[i] = NumRepeatValueInput[i-1] / NumValueInput[i]

3.3.3) เมื่อได้จำนวนการเขียนซ้ำของค่าข้อมูลที่เป็นไปได้ของแต่ละ Input แล้ว จึงใส่ค่าข้อมูลที่เป็นไปได้ของแต่ละ Input ลงในตาราง โดยเขียนซ้ำค่าที่เป็นไปได้ของแต่ละ input ตามจำนวนการเขียนซ้ำ

3.4) เลือกข้อมูลนำออกหรือผลลัพธ์ที่เป็นไปได้ของแต่ละกรณี โดยระบบจะแสดงข้อมูลนำออกหรือผลลัพธ์ที่ต้องการ (Output) นักวิเคราะห์ระบบจะต้องเลือกว่า Output ใดบ้างที่จะเป็นผลลัพธ์ของแต่ละกรณีทดสอบ

แสดงโครงสร้างและขั้นตอนการสร้างตารางตัดสินใจจากหัวข้อที่ 3.1 ถึง 3.4 ดังภาพประกอบ 3.1



ภาพประกอบ 3.1 โครงสร้างตารางการตัดสินใจ

<u>ขั้นตอนที่ 5</u> **สร้างกรณีทดสอบ**

ระบบสร้างกรณีทดสอบ (Test case) โดยมีขั้นตอนดังนี้

1) นำชื่อฟังก์ชันงานมาเป็นชื่อกรณีทดสอบ

2) นำคำอธิบายฟังก์ชันงานของนิยามความต้องการมาเป็นคำอธิบายการทำงานของกรณีทดสอบ

3) นำเงื่อนไขก่อนการดำเนินงาน (ถ้ามี) และคำอธิบายการทำงานของคุณลักษณะความต้องการมาเป็นขั้นตอนในการทดสอบ

4) นำข้อมูลจากตารางตัดสินใจมาสร้างกรณีทดสอบ ดังนี้

4.1) วิธีการสร้าง Input ของกรณีทดสอบ

```
for (i = 1; i <= NumInput[i]; i++)
{   count = 0;
    do{  for (j = 1; j <= NumValueInput[i]; j++)
         {   for (k = 1; k <= NumRepeatValueInput[i]; k++)
             {
               ValueRepeatArray[i] = ValueRepeatArray[i] & ValueInputArray[j] + ",";
             }
         }
         temp = NumValueInput[i] * NumRepeatValueInput[i];
         count = count + temp;
    } while (count != NumTestCase);
}
// NumInput[i] คือ จำนวน Input ของฟังก์ชันงาน
// count คือ ค่าจำนวนเต็มที่ใช้ตรวจสอบว่าได้ค่าครบตามจำนวนกรณีทดสอบแล้วหรือไม่
// NumValueInput[i] คือ จำนวนค่าที่เป็นไปได้ของ Input[i]
// NumRepeatValueInput[i] คือ จำนวนการเขียนซ้ำของค่าที่เป็นไปได้ของ Input[i]
// ValueRepeatArray[i] คือ ค่าที่เป็นไปได้ทั้งหมดของ Input[i] เก็บตามจำนวนการเขียนซ้ำ
// ValueInputArray[j] คือ ค่าที่เป็นไปได้ตัวที่ j ของ Input[i]
// temp คือ จำนวนกรณีทดสอบที่เก็บค่าได้ใน 1 รอบ do…while
// NumTestCase คือ จำนวนกรณีทดสอบของฟังก์ชันงาน
```

4.2) วิธีการสร้าง Expected result ของกรณีทดสอบ

```
for (i = 1; i <= NumTestCase; i++)
{    for (j = 0; j < NumOutput; j++)
     {    if (CheakOutput[i][j] == true)
          {    ExpectedResultTestCase[i] = ExpectedResultTestCase[i] & Output[j];    }}}
// NumTestCase คือ จำนวนกรณีทดสอบของฟังก์ชันงาน
// NumOutput คือ จำนวนผลลัพธ์ที่เป็นไปได้ทั้งหมดของฟังก์ชันงาน
// CheakOutput[i][j] คือ ค่าการตรวจสอบผลลัพธ์ที่คาดหวังของกรณีทดสอบที่ i ผลลัพธ์ที่ j
// ExpectedResultTestCase[i] คือ ค่าผลลัพธ์ที่คาดหวังของกรณีทดสอบที่ i
// Output[j] คือ ค่าของผลลัพธ์ที่ j
```

3.4 ตัวอย่างการสร้างกรณีทดสอบจากคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม

แม้ว่าแนวคิดการสร้างกรณีทดสอบจากข้อกำหนดคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม จะสามารถนำไปใช้กับความต้องการซอฟต์แวร์ได้หลายประเภท แต่เพื่อให้ง่ายต่อการสาธิตแนวคิดที่นำเสนอ งานวิจัยนี้ได้เลือกใช้ซอฟต์แวร์สำหรับการประมวลผลเครื่องฝาก-ถอนเงินอัตโนมัติอย่างง่าย (Simple Automatic Teller Machine: ATM) เป็นกรณีตัวอย่าง ทั้งนี้ เงื่อนไขในฟังก์ชันการทำงานต่างๆ อาจไม่สอดคล้องกับความเป็นจริงของแต่ละธนาคารที่มีความแตกต่างกัน ผู้พัฒนาระบบจะถือว่าเงื่อนไขเหล่านี้เป็นเงื่อนไขตัวอย่างที่แต่ละธนาคารสามารถนำไปปรับเปลี่ยนได้ตามความต้องการ เช่น การอนุญาตให้ป้อนรหัสผิดได้ไม่เกิน 3 ครั้ง หากเกิน 3 ครั้ง แล้วเครื่อง ATM จะยึดบัตร

ปัญหากรณีตัวอย่าง

The software to be designed will control a simulated automated teller machine (ATM) having a magnetic stripe reader for reading the ATM card, a keyboard and display for interaction with the customer, a slot for depositing, a dispenser for cash, a printer for printing customer receipts, and a key-operated switch to allow an operator to start or stop the machine. The ATM will communicate with the computer of a bank over an appropriate communication link. (The software on the latter is not part of the requirements for this problem.)

The ATM will service one customer at a time. A customer will be required to insert the ATM card and enter a personal identification number (PIN) - both of which will be sent to the bank for validation as a part of each transaction. The customer will be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned. If the ATM card or customer PIN is invalid, an error message will be display on the monitor and return the ATM card.

The ATM must be able to provide the following services to a customer:

1) A customer must be able to check a balance inquiry of any account linked to the ATM card. After performing this function, the customer chooses whether to continue or not. If the customer selects “No”, the ATM will exit and return the ATM card. If the customer selects “Yes”, the customer will select another transaction.

2) A customer chooses the deposit button, then puts the right cash in the slot and presses the confirmation button. The ATM will check the balance and display the net balance to the customer. The ATM will accept a deposit from 100 Bahts to 100,000 Bahts, when the customer presses a deposit confirmation. If condition of deposit is true, the ATM will display the total deposit and net balance. Then the ATM prints a receipt and returns the ATM card. If condition of deposit is false, the ATM will display a warning message and return the ATM card.

3) The ATM will accept a withdrawal from 100 Bahts to 20,000 Bahts for each time, when a customer chooses to withdraw and enter an amount of money. If the amount of withdrawals is less than or equal to the remaining balance, the system will allow a withdrawal from the ATM, then displays money to withdraw and balance on the monitor, then prints a receipt and returns the ATM card. But if the customer withdraws more than the balance, the ATM will not be allowed to withdraw money from the ATM, then displays a warning message and return the ATM card.

จากปัญหากรณีตัวอย่างนักวิเคราะห์ระบบ (Analyst) ได้จัดแบ่งนิยามความต้องการเชิงฟังก์ชันเป็น 4 ฟังก์ชัน คือ ฟังก์ชันเข้าการสู่ระบบ (Login) ฟังก์ชันสอบถามยอดเงิน (Check balance) ฟังก์ชันฝากเงิน (Deposit) และฟังก์ชันถอนเงิน (Withdrawal) สามารถสร้างกรณีทดสอบจากวิธีการที่นำเสนอได้ ดังนี้

3.4.1 การสร้างกรณีทดสอบจากคุณลักษณะความต้องการในรูปแบบฟอร์มของฟังก์ชันการเข้าสู่ระบบด้วยตู้เอทีเอ็ม

ขั้นตอนที่ 1 นิยามความต้องการ

Function	Login
Description	The ATM will service one customer at a time. A customer will be required to inserts the ATM card and enters a personal identification number (PIN) - both of which will be sent to the bank for validation as a part of each transaction. The customer will be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned. If the ATM card or the customer PIN is invalid, the monitor will display an error message and return the ATM card.

ขั้นตอนที่ 2 กำหนดคุณลักษณะความต้องการให้อยู่ในรูปแบบฟอร์ม

Function	Login
Description	1. A customer will be required to insert the ATM card. 2. The customer enters a personal identification number (PIN).
Inputs	The ATM card data, PIN
Source	The ATM card reader, keyboard
Outputs	The monitor shows the main menu for performing transactions. / The monitor displays an error message and returns the ATM card.
Destination	The monitor
Requires	The ATM card, PIN, Database of customer account
Pre-condition	The ATM is ready for use.
Post-condition	The monitor shows message on the screen or the menu corresponding to its output.

ขั้นตอนที่ 3 ระบุชนิดของข้อมูลนำเข้าและป้อนค่าที่เป็นไปได้ของข้อมูลนำเข้า

Inputs	Type	Possible value
The ATM card data	Boolean	True, False
PIN	Boolean	True, False

ขั้นตอนที่ 4 สร้างตารางตัดสินใจ

1) คำนวณหาจำนวนแถวของตาราง

NumRowDecisionTable	=	NumInput + 1
	=	2 + 1
	=	3

2) คำนวณหาจำนวนคอลัมน์ของตาราง

NumColumnDecisionTable	=	(NumValueInput[1] x NumValueInput[2]) + 1
	=	(2 x 2) + 1
	=	5

3) การใส่ค่าลงในตารางตัดสินใจ

3.1) ใส่เลข 1 ลงในแถวที่ 1 คอลัมน์ที่ 2 ของตารางตัดสินใจ แล้วเพิ่มค่าทีละ 1 เรื่อยไปในแถวที่ 1 จนถึงคอลัมน์สุดท้าย

3.2) ใส่ค่าของข้อมูลนำเข้าตัวที่ 1 (Input[1]) ลงในแถวที่ 2 คอลัมน์ที่ 1 ของตารางตัดสินใจ แล้วใส่ข้อมูลนำเข้าตัวถัดไปในคอลัมน์ที่ 1 เรื่อยไปจนถึงแถวสุดท้าย

จากข้อ 1, 2, 3.1 และ 3.2 สามารถแสดงได้ดังภาพประกอบ 3.2

	1	2	3	4
The ATM card data				
PIN				

ภาพประกอบ 3.2 วิธีสร้างตารางการตัดสินใจของฟังก์ชัน Login ระบบ ATM

3.3) ใส่ค่าต่างๆ ที่เป็นไปได้สำหรับข้อมูลนำเข้าแต่ละตัว ดังนี้

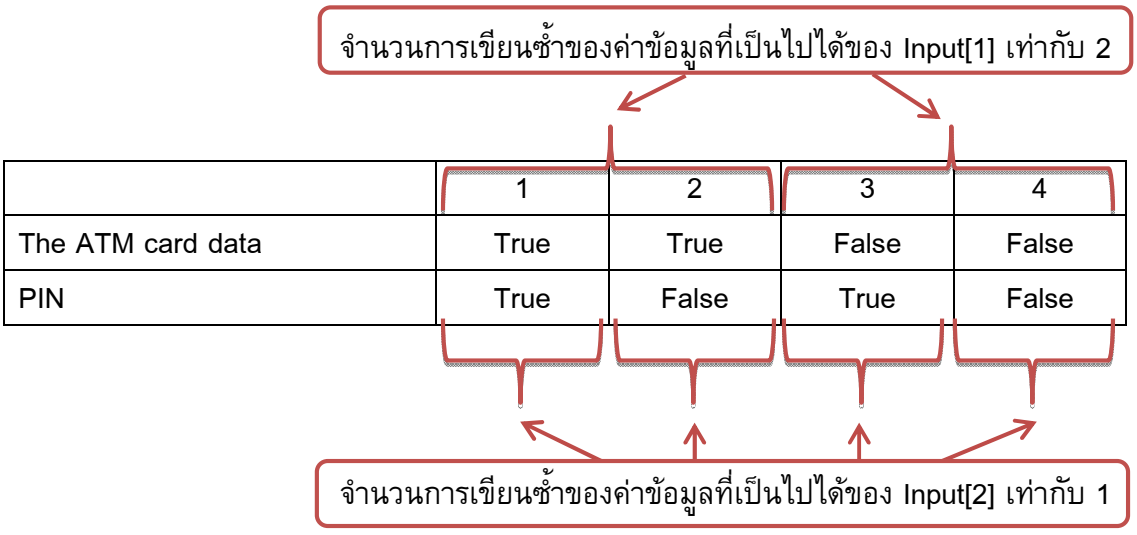
3.3.1) หาจำนวนการเขียนซ้ำของค่าข้อมูลที่เป็นไปได้ของ Input[1] นั่นคือ The ATM card data

```
NumRepeatValueInput[1]  = (NumColumnDecisionTable – 1) /
                            NumValueInput[1]
                        = (5 – 1) / 2
                        = 2
```

3.3.2) หาจำนวนการเขียนซ้ำของค่าข้อมูลที่เป็นไปได้ของ Input[2] นั่นคือ PIN

```
NumRepeatValueInput[2]  = NumRepeatPossibleInput[2-1] /
                            NumValueInput[2]
                        = 2 / 2
                        = 1
```

3.3.3) เมื่อได้จำนวนการเขียนซ้ำของค่าข้อมูลที่เป็นไปได้ของแต่ละ Input แล้ว จึงใส่ค่าข้อมูลที่เป็นไปได้ของแต่ละ Input ลงในตาราง โดยเขียนซ้ำค่าที่เป็นไปได้ของแต่ละ Input ตามจำนวนการเขียนซ้ำ ดังภาพประกอบ 3.3



ภาพประกอบ 3.3 วิธีใส่ค่าในตารางการตัดสินใจของฟังก์ชัน Login ระบบ ATM

4) เลือกข้อมูลนำออกหรือผลลัพธ์ที่เป็นไปได้ของแต่ละกรณี โดยระบบจะแสดงข้อมูลนำออกหรือผลลัพธ์ที่ต้องการ นักวิเคราะห์ระบบจะต้องเลือกว่า Output ใด จะเป็นผลของกรณีทดสอบใด ดังภาพประกอบ 3.4

	1	2	3	4
The ATM card data	True	True	False	False
PIN	True	False	True	False
The monitor shows the main menu for performing transactions.	x			
The monitor displays an error message and returns the ATM card.		x	x	x

ภาพประกอบ 3.4 ตารางการตัดสินใจของฟังก์ชัน Login ระบบ ATM

ขั้นตอนที่ 5 สร้างกรณีทดสอบ

จากคุณลักษณะความต้องการสามารถสร้างกรณีทดสอบเบื้องต้นได้ ดังนี้

Test case : Login

Description : The ATM will service one customer at a time. A customer will be required to inserts the ATM card and enters a personal identification number (PIN) - both of which will be sent to the bank for validation as a part of each transaction. The customer will be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned. If the ATM card or the customer PIN is invalid, the monitor will display an error message and return the ATM card.

Step : Pre-condition is the ATM is ready for use.

1. A customer will be required to insert the ATM card.
2. The customer enters a personal identification number (PIN).

Input	Expected result
The ATM card data: True PIN: True	The monitor shows the main menu for performing transactions.
The ATM card data: True PIN: False	The monitor displays an error message and returns the ATM card.
The ATM card data: False PIN: True	The monitor displays an error message and returns the ATM card.
The ATM card data: False PIN: False	The monitor displays an error message and returns the ATM card.

3.4.2 การสร้างกรณีทดสอบจากคุณลักษณะความต้องการในรูปแบบฟอร์มของฟังก์ชันการสอบถามยอดเงินคงเหลือด้วยตู้เอทีเอ็ม

ขั้นตอนที่ 1 นิยามความต้องการ

Function	Check balance
Description	A customer must be able to check a balance inquiry of any account linked to the ATM card. After performing this function, the customer chooses whether to continue or not. If the customer selects “No”, the ATM will exit and return the ATM card. If the customer selects “Yes”, the customer will select another transaction.

ขั้นตอนที่ 2 กำหนดคุณลักษณะความต้องการให้อยู่ในรูปแบบฟอร์ม

Function	Check balance
Description	1. A customer selects the “check balance” button. 2. The ATM displays net balance. 3. The customer chooses whether to continue or not.
Inputs	The button selection
Source	The ATM button
Outputs	The ATM displays net balance, exits and returns the ATM card. / The ATM displays net balance and displays transaction for a customer selection.

Destination	The monitor
Requires	Database of customer account
Pre-condition	The monitor displays transaction for a customer selection.
Post-condition	The monitor displays the result of the “Yes” or “No” selection

ขั้นตอนที่ 3 ระบุชนิดของข้อมูลนำเข้าและป้อนค่าที่เป็นไปได้ของข้อมูลนำเข้า

Inputs	Type	Possible value
The button selection	Boolean	True, False

ขั้นตอนที่ 4 สร้างตารางตัดสินใจ

1) คำนวณหาจำนวนแถวของตาราง

NumRowDecisionTable	=	NumInput + 1
	=	1 + 1
	=	2

2) คำนวณหาจำนวนคอลัมน์ของตาราง

NumColumnDecisionTable	=	NumValueInput[1] + 1
	=	2 + 1
	=	3

3) การใส่ค่าลงในตารางตัดสินใจ

3.1) ใส่เลข 1 ลงในแถวที่ 1 คอลัมน์ที่ 2 ของตารางตัดสินใจ แล้วเพิ่มค่าทีละ 1 เรื่อยไปในแถวที่ 1 จนถึงคอลัมน์สุดท้าย

3.2) ใส่ค่าของข้อมูลนำเข้าตัวที่ 1 (Input[1]) ลงในแถวที่ 2 คอลัมน์ที่ 1 ของตารางตัดสินใจ แล้วใส่ข้อมูลนำเข้าตัวถัดไปในคอลัมน์ที่ 1 เรื่อยไปจนถึงแถวสุดท้าย

3.3) ใส่ค่าต่าง ๆ ที่เป็นไปได้สำหรับข้อมูลนำเข้าแต่ละตัว โดยไม่ต้องหาจำนวนการเขียนซ้ำของค่าข้อมูลที่เป็นไปได้ของ Input แต่ละตัว เนื่องจากมี Input เพียง 1 ตัว

สามารถแสดงการสร้างตารางการตัดสินใจของฟังก์ชัน Check balance ได้ดังภาพประกอบ 3.5

	1	2
The button selection	True	False

จำนวนการเขียนซ้ำของค่าข้อมูลที่เป็นไปได้ของ Input[1] เท่ากับ 1 เสมอ
หากฟังก์ชันงานนั้นมีจำนวนข้อมูลนำเข้าเพียง 1 ตัว

ภาพประกอบ 3.5 วิธีใส่ค่าในตารางการตัดสินใจของฟังก์ชัน Check balance ระบบ ATM

4) เลือกข้อมูลนำออกหรือผลลัพธ์ที่เป็นไปได้ของแต่ละกรณี โดยระบบจะแสดงข้อมูลนำออกหรือผลลัพธ์ที่ต้องการ นักวิเคราะห์ระบบจะต้องเลือกว่า Output ใด จะเป็นผลของกรณีทดสอบใด ดังภาพประกอบ 3.6

	1	2
The button selection	True	False
The ATM displays net balance, exits and returns the ATM card.		x
The ATM displays net balance and displays transaction for a customer selection.	x	

ภาพประกอบ 3.6 ตารางการตัดสินใจของฟังก์ชัน Check balance ระบบ ATM

ขั้นตอนที่ 5 สร้างกรณีทดสอบ

จากคุณลักษณะความต้องการสามารถสร้างกรณีทดสอบเบื้องต้นได้ ดังนี้

Test case : Check balance

Description : A customer must be able to check a balance inquiry of any account linked to the ATM card. After performing this function, the customer chooses whether to continue or not. If the customer selects “No”, the ATM will exit and return the ATM card. If the customer selects “Yes”, the customer will select another transaction.

Step : Pre-condition is the monitor displays transaction for a customer selection.

1. A customer selects the “check balance” button.
2. The ATM displays net balance.
3. The customer chooses whether to continue or not.

Input	Expected result
The button selection : True	The ATM displays net balance and displays transaction for a customer selection.
The button selection : False	The ATM displays net balance, exits and returns the ATM card.

3.4.3 การสร้างกรณีทดสอบจากคุณลักษณะความต้องการในรูปแบบฟอร์มของฟังก์ชันการฝากเงินสดด้วยตู้เอทีเอ็ม

ขั้นตอนที่ 1 นิยามความต้องการ

Function	Deposit
Description	A customer chooses the deposit button, then puts the right cash in the slot and presses the confirmation button. The ATM will check the balance and display the net balance to the customer. The ATM will accept a deposit from 100 Bahts to 100,000 Bahts, when the customer presses a deposit confirmation. If condition of deposit is true, the ATM will display the total deposit and net balance. Then the ATM prints a receipt and returns the ATM card. If condition of deposit is false, the ATM will display a warning message and return the ATM card.

ขั้นตอนที่ 2 กำหนดคุณลักษณะความต้องการให้อยู่ในรูปแบบฟอร์ม

Function	Deposit
Description	1. A customer chooses the “deposit” button 2. The customer puts the right cash in the slot and presses the confirmation button.
Inputs	An amount of deposit

Source	The ATM
Outputs	The ATM displays the total deposit and net balance in the customer account, prints a receipt and returns the ATM card. / The ATM displays a warning message and returns the ATM card.
Destination	The ATM
Requires	The deposit cash, Database of customer account
Pre-condition	The monitor displays transactions for a customer selection.
Post-condition	The monitor shows a message on the screen.

ขั้นตอนที่ 3 ระบุชนิดของข้อมูลนำเข้าและป้อนค่าที่เป็นไปได้ของข้อมูลนำเข้า

Inputs	Type	Possible value	Min	Max
An amount of deposit	Number	-	100	100000

ขั้นตอนที่ 4 สร้างตารางตัดสินใจ

1) คำนวณหาจำนวนแถวของตาราง

NumRowDecisionTable	=	NumInput + 1
	=	1 + 1
	=	2

2) คำนวณหาจำนวนคอลัมน์ของตาราง

ในกรณีที่ Input ใด มีการระบุค่าข้อมูลสูงสุด (Max) และค่าข้อมูลต่ำสุด (Min) จะใช้หลักการการแบ่งส่วนสมมูลในการกำหนดค่าข้อมูลที่เป็นไปได้ของ Input นั้น ดังนี้

- ค่าข้อมูลต่ำสุดของพิสัยข้อมูล นั่นคือ ค่า Min เท่ากับ 100
- ค่าข้อมูลสูงสุดขอพิสัยข้อมูล นั่นคือ ค่า Max เท่ากับ 100000
- ค่าข้อมูลตัวแทนกลุ่ม เป็นค่าที่ใกล้เคียงกับค่ากลางของพิสัยข้อมูล นั่นคือ ค่า 100 + ((100000 – 100) / 2) เท่ากับ 50050
- ค่าข้อมูลเกินพิสัย หรือเกินขอบเขตข้อมูลในแต่ละช่วง ประกอบด้วย 2 ค่า คือ 99 และ 100001

จำนวนค่าข้อมูลที่เป็นไปได้ของ Input จึงเท่ากับ 5

NumColumnDecisionTable	=	NumValueInput[1] + 1
	=	5 + 1
	=	6

3) การใส่ค่าลงในตารางตัดสินใจ

3.1) ใส่เลข 1 ลงในแถวที่ 1 คอลัมน์ที่ 2 ของตารางตัดสินใจ แล้วเพิ่มค่าทีละ 1 เรื่อยไปในแถวที่ 1 จนถึงคอลัมน์สุดท้าย

3.2) ใส่ค่าของข้อมูลนำเข้าตัวที่ 1 (Input[1]) ลงในแถวที่ 2 คอลัมน์ที่ 1 ของตารางตัดสินใจ แล้วใส่ข้อมูลนำเข้าตัวถัดไปในคอลัมน์ที่ 1 เรื่อยไปจนถึงแถวสุดท้าย

3.3) ใส่ค่าต่าง ๆ ที่เป็นไปได้สำหรับข้อมูลนำเข้าแต่ละตัว โดยไม่ต้องหาจำนวนการเขียนซ้ำของค่าข้อมูลที่เป็นไปได้ของ Input แต่ละตัว เนื่องจากมี Input เพียง 1 ตัว

สามารถแสดงการสร้างตารางการตัดสินใจของฟังก์ชัน Deposit ได้ดังภาพประกอบ 3.7

	1	2	3	4	5
An amount of deposit	99	100	50050	100000	100001

ภาพประกอบ 3.7 วิธีใส่ค่าในตารางการตัดสินใจของฟังก์ชัน Deposit ระบบ ATM

	1	2	3	4	5
An amount of deposit	99	100	50050	100000	100001
The ATM displays the total deposit and net balance in the customer account, prints a receipt and returns the ATM card.		x	x	x	
The ATM displays a warning message and returns the ATM card.	x				x

ภาพประกอบ 3.8 ตารางการตัดสินใจของฟังก์ชัน Deposit ระบบ ATM

4) เลือกข้อมูลนำออกหรือผลลัพธ์ที่เป็นไปได้ของแต่ละกรณี โดยระบบจะแสดงข้อมูลนำออกหรือผลลัพธ์ที่ต้องการ นักวิเคราะห์ระบบจะต้องเลือกว่า Output ใด จะเป็นผลของกรณีทดสอบใด ดังภาพประกอบ 3.8

ขั้นตอนที่ 5 สร้างกรณีทดสอบ

จากคุณลักษณะความต้องการสามารถสร้างกรณีทดสอบเบื้องต้นได้ ดังนี้

Test case : Deposit

Description : A customer chooses the deposit button, then puts the right cash in the slot and presses the confirmation button. The ATM will check the balance and display the net balance to the customer. The ATM will accept a deposit from 100 Bahts to 100,000 Bahts, when the customer presses a deposit confirmation. If condition of deposit is true, the ATM will display the total deposit and net balance. Then the ATM prints a receipt and returns the ATM card. If condition of deposit is false, the ATM will display a warning message and return the ATM card.

Step : Pre-condition is the monitor displays transactions for a customer selection.

1. A customer chooses the “deposit” button.
2. The customer payments and presses the confirmation button.

Input	Expected result
An amount of deposit: 99	The ATM displays a warning message and returns the ATM card.
An amount of deposit : 100	The ATM displays the total deposit and net balance in the customer account, prints a receipt and returns the ATM card.
An amount of deposit : 50050	The ATM displays the total deposit and net balance in the customer account, prints a receipt and returns the ATM card.
An amount of deposit : 100000	The ATM displays the total deposit and net balance in the customer account, prints a receipt and returns the ATM card.
An amount of deposit : 100001	The ATM displays a warning message and returns the ATM card.

3.4.4 การสร้างกรณีทดสอบจากคุณลักษณะความต้องการของการถอนเงินสดจากตู้เอทีเอ็ม

ขั้นตอนที่ 1 นิยามความต้องการ

Function	Withdrawal
Description	A customer chooses the deposit button, then puts the right cash in the slot and presses the confirmation button. The ATM will check the balance and display the net balance to the customer. The ATM will accept a deposit from 100 Bahts to 100,000 Bahts, when the customer presses a deposit confirmation. If condition of deposit is true, the ATM will display the total deposit and net balance. Then the ATM prints a receipt and returns the ATM card. If condition of deposit is false, the ATM will display a warning message and return the ATM card.

ขั้นตอนที่ 2 กำหนดคุณลักษณะความต้องการให้อยู่ในรูปแบบฟอร์ม

Function	Withdrawal
Description	1. A customer chooses the “withdrawal” button. 2. The customer enters an amount of money.
Inputs	An amount of withdrawal
Source	The ATM button
Outputs	The ATM will allow a withdrawal from the ATM, then displays money to withdraw and balance on the monitor, then prints a receipt and return the ATM card. / The ATM will not be allowed to withdraw money from the ATM, then displays a warning message and return the ATM card.
Destination	The ATM
Requires	The amount of withdrawal, Database of customer account
Pre-condition	The monitor displays transactions for a customer selection.
Post-condition	The monitor shows a message on the screen.

ขั้นตอนที่ 3 ระบุชนิดของข้อมูลนำเข้าและป้อนค่าที่เป็นไปได้ของข้อมูลนำเข้า

Inputs	Type	Possible value	Min	Max
An amount of withdrawal	Number	<balance, =balance, >balance	100	20000

ขั้นตอนที่ 4 สร้างตารางตัดสินใจ

1) คำนวณหาจำนวนแถวของตาราง

NumRowDecisionTable	=	NumInput + 1
	=	1 + 1
	=	2

2) คำนวณหาจำนวนคอลัมน์ของตาราง

ในกรณีที่ Input ใด มีการระบุค่าข้อมูลสูงสุด (Max) และค่าข้อมูลต่ำสุด (Min) และมีการระบุเงื่อนไขในช่องของค่าข้อมูลที่เป็นไปได้ของ Input นั้น จำนวนของค่าข้อมูลที่เป็นไปได้ของ Input นั้นจะเท่ากับจำนวนข้อมูลที่เป็นไปได้บวกด้วยค่าจากการใช้หลักการการแบ่งส่วนสมมูล ดังนั้นจำนวนค่าข้อมูลที่เป็นไปได้ของ An amount of withdrawal จึงเท่ากับ 8 ซึ่งได้แก่ <balance, =balance, >balance, Min - 1, Min, Min + ((Max – Min) / 2), Max และ Max + 1

NumColumnDecisionTable	=	NumValueInput[1] + 1
	=	8 + 1
	=	9

3) การใส่ค่าลงในตารางตัดสินใจ

3.1) ใส่เลข 1 ลงในแถวที่ 1 คอลัมน์ที่ 2 ของตารางตัดสินใจ แล้วเพิ่มค่าทีละ 1 เรื่อยไปในแถวที่ 1 จนถึงคอลัมน์สุดท้าย

3.2) ใส่ค่าของข้อมูลนำเข้าตัวที่ 1 (Input[1]) ลงในแถวที่ 2 คอลัมน์ที่ 1 ของตารางตัดสินใจ แล้วใส่ข้อมูลนำเข้าตัวถัดไปในคอลัมน์ที่ 1 เรื่อยไปจนถึงแถวสุดท้าย

3.3) ใส่ค่าต่าง ๆ ที่เป็นไปได้สำหรับข้อมูลนำเข้าแต่ละตัว โดยไม่ต้องหาจำนวนการเขียนซ้ำของค่าข้อมูลที่เป็นไปได้ของ Input แต่ละตัว เนื่องจากมี Input เพียง 1 ตัว

สามารถแสดงการสร้างตารางการตัดสินใจของฟังก์ชัน Withdrawal ได้ดังภาพประกอบ 3.9

	1	2	3	4	5	6	7	8
An amount of withdrawal	<balance	=balance	>balance	99	100	10050	20000	20001

ภาพประกอบ 3.9 วิธีใส่ค่าในตารางการตัดสินใจของฟังก์ชัน Withdrawal ระบบ ATM

4) เลือกข้อมูลนำออกหรือผลลัพธ์ที่เป็นไปได้ของแต่ละกรณี โดยระบบจะแสดงข้อมูลนำออกหรือผลลัพธ์ที่ต้องการ นักวิเคราะห์ระบบจะต้องเลือกว่า Output ใด จะเป็นผลของกรณีทดสอบใด แสดงดังภาพประกอบ 3.10 โดยใช้ Output 1 แทน The ATM will allow a withdrawal from the ATM, then displays money to withdraw and balance on the monitor, then prints a receipt and return the ATM card. และใช้ Output 2 แทน The ATM will not be allowed to withdraw money from the ATM, then displays a warning message and return the ATM card.

	1	2	3	4	5	6	7	8
An amount of withdrawal	<balance	=balance	>balance	99	100	10050	20000	20001
Output 1	x	x			x	x	x	
Output 2			x	x				x

ภาพประกอบ 3.10 ตารางการตัดสินใจของฟังก์ชัน Withdrawal ระบบ ATM

ขั้นตอนที่ 5 สร้างกรณีทดสอบ

จากคุณลักษณะความต้องการสามารถสร้างกรณีทดสอบเบื้องต้นได้ ดังนี้

Test case : Withdrawal

Description : A customer chooses the deposit button, then puts the right cash in the slot and presses the confirmation button. The ATM will check the balance and display the net balance to the customer. The ATM will accept a deposit from 100 Bahts to 100,000 Bahts, when the customer presses a deposit confirmation. If condition of deposit is true, the ATM will display the total deposit and net balance. Then the ATM prints a receipt and returns the ATM

card. If condition of deposit is false, the ATM will display a warning message and return the ATM card.

Step : Pre-condition is the monitor displays transactions for a customer selection.

1. A customer chooses the “withdrawal” button.
2. The customer enters an amount of money.

Input	Expected result
An amount of withdrawal: <balance	The ATM will allow a withdrawal from the ATM, then displays money to withdraw and balance on the monitor, then prints a receipt and return the ATM card.
An amount of withdrawal: =balance	The ATM will allow a withdrawal from the ATM, then displays money to withdraw and balance on the monitor, then prints a receipt and return the ATM card.
An amount of withdrawal: >balance	The ATM will not be allowed to withdraw money from the ATM, then displays a warning message and return the ATM card.
An amount of withdrawal: 99	The ATM will not be allowed to withdraw money from the ATM, then displays a warning message and return the ATM card.
An amount of withdrawal: 100	The ATM will allow a withdrawal from the ATM, then displays money to withdraw and balance on the monitor, then prints a receipt and return the ATM card.
An amount of withdrawal: 10050	The ATM will allow a withdrawal from the ATM, then displays money to withdraw and balance on the monitor, then prints a receipt and return the ATM card.
An amount of withdrawal: 20000	The ATM will allow a withdrawal from the ATM, then displays money to withdraw and balance on the monitor, then prints a receipt and return the ATM card.
An amount of withdrawal: 20001	The ATM will not be allowed to withdraw money from the ATM, then displays a warning message and return the ATM card.

บทที่ 4

การออกแบบระบบ

เนื้อหาในบทนี้กล่าวถึงการออกแบบระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม (A Support System for Elementary Test Case Generation Based on Form-Based Requirements Specification) ซึ่งเป็นเครื่องมือที่พัฒนาขึ้นเพื่อสนับสนุนการสร้างกรณีทดสอบจากคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม โดยจะกล่าวถึงกรอบแนวคิดของระบบ การออกแบบกระบวนการทำงานโดยรวม การออกแบบโครงสร้างฐานข้อมูลของระบบ และการออกแบบส่วนติดต่อกับผู้ใช้งาน

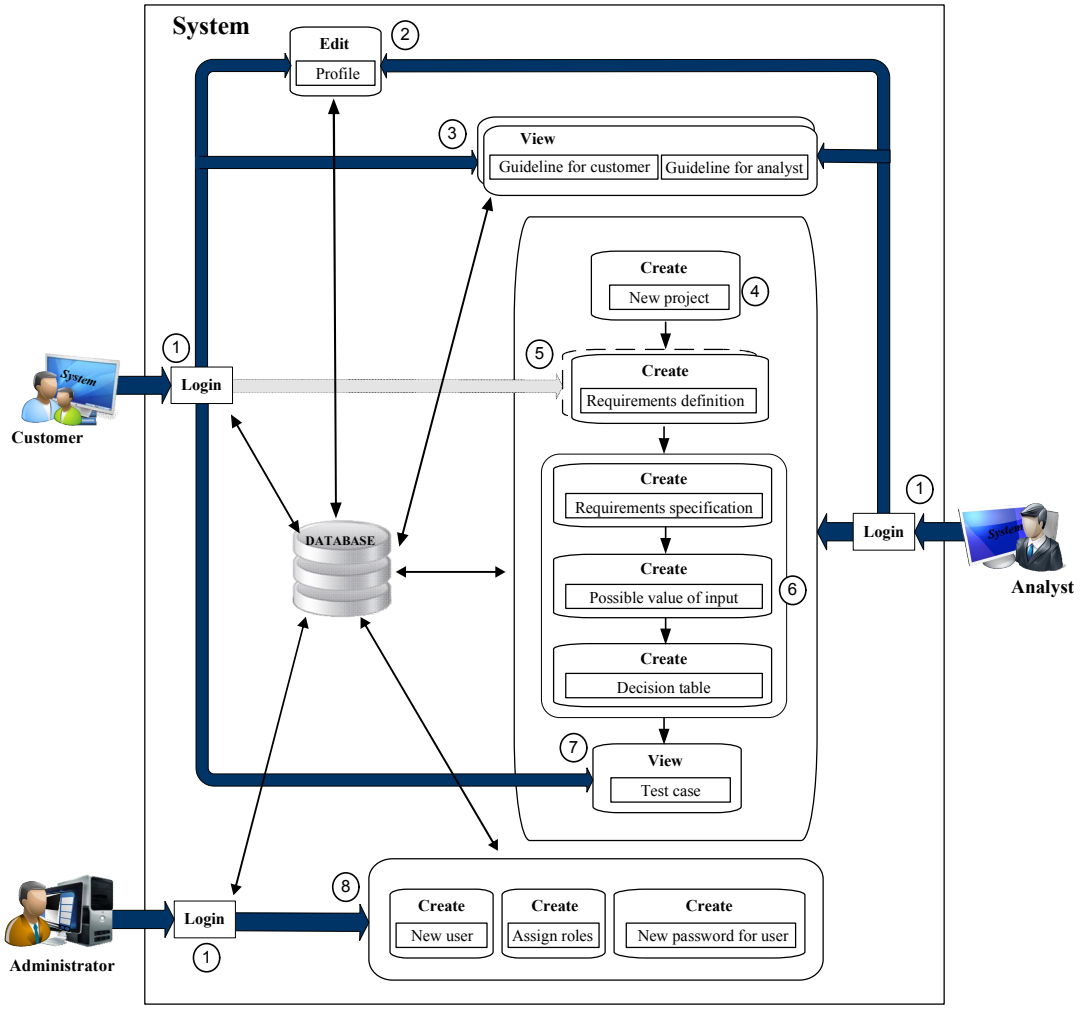
4.1 กรอบแนวคิดของระบบ

จากการวิเคราะห์ความต้องการของกลุ่มผู้ที่เกี่ยวข้องที่กล่าวไว้ในบทที่ 3 สามารถออกแบบกรอบแนวคิดโดยทั่วไปของระบบ ที่มีกลุ่มผู้เกี่ยวข้อง 3 กลุ่ม ดังภาพประกอบ 4.1 ประกอบด้วยส่วนของผู้ใช้ทั่วไป (Customer) ซึ่งมีหน้าที่ในการป้อนนิยามความต้องการและเลือกดูผลกรณีทดสอบที่ระบบสร้างให้ กลุ่มที่สอง คือ นักวิเคราะห์ระบบ (Analyst) ซึ่งมีบทบาทสำคัญในระบบเป็นอย่างยิ่ง เพราะมีหน้าที่ในการป้อนคุณลักษณะความต้องการในรูปแบบฟอร์มซึ่งขยายจากนิยามความต้องการของผู้ใช้ ระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า และเลือกผลลัพธ์ที่คาดหวังของแต่ละกรณีในตารางตัดสินใจ ซึ่งเป็น 3 ส่วนหลักๆ ที่จะใช้เป็นข้อมูลในการสร้างกรณีทดสอบเบื้องต้นให้กับผู้ใช้ ส่วนกลุ่มที่สาม คือ ผู้ดูและระบบ (Administrator) ซึ่งมีหน้าที่สร้างบัญชีผู้ใช้และแก้ไขข้อมูลที่จำเป็นให้กับผู้ใช้ในส่วนต่างๆ

จากกรอบแนวคิดของระบบในภาพประกอบ 4.1 สามารถจำแนกองค์ประกอบออกเป็น 3 ส่วน ดังนี้

1) โมดูล

โมดูลเป็นส่วนการทำงานย่อยของระบบ ที่สร้างขึ้นเพื่อให้ผู้ใช้เลือกดำเนินการต่างๆ กับระบบตามบทบาทและหน้าที่ของผู้ใช้แต่ละกลุ่ม เช่น ส่วนของการนิยามความต้องการ ส่วนของการกำหนดคุณลักษณะความต้องการในรูปแบบฟอร์ม หรือส่วนแสดงคำแนะนำในการใช้งานระบบ เป็นต้น



ภาพประกอบ 4.1 กรอบแนวคิดต้นแบบระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม

2) กลุ่มผู้ที่เกี่ยวข้อง

2.1) ผู้ใช้ทั่วไป เป็นบุคคลที่ทำหน้าที่ในการนิยามความต้องการ ดูวิธีการใช้งานระบบและส่วนช่วยเหลือสนับสนุน รวมถึงการเลือกดูผลกรณีทดสอบที่ระบบสร้างให้ และแก้ไขข้อมูลส่วนตัว

2.2) นักวิเคราะห์ระบบ เป็นบุคคลที่ทำหน้าที่ในการสร้างโครงงานใหม่ นิยามความต้องการ กำหนดคุณลักษณะความต้องการในรูปแบบฟอร์ม ระบุชนิดของข้อมูลนำเข้าและป้อนค่าที่เป็นไปได้ของข้อมูลนำเข้า เลือกผลลัพธ์ที่คาดหวังของแต่ละกรณีในตารางตัดสินใจ ดูวิธีการใช้งานระบบและส่วนช่วยเหลือสนับสนุน รวมถึงการเลือกดูผลกรณีทดสอบที่ระบบสร้างให้ และแก้ไขข้อมูลส่วนตัว

2.3) ผู้ดูและระบบ เป็นบุคคลที่ทำหน้าที่ในการสร้างบัญชีผู้ใช้และระบุข้อมูลเบื้องต้นให้กับผู้ใช้ คอยสร้างรหัสผ่านใหม่ให้กับผู้ใช้ในกรณีที่ผู้ใช้ลืมรหัสผ่าน และแก้ไขจำนวนนักวิเคราะห์ระบบที่สามารถเข้าดำเนินการกับแต่ละโครงงานได้

3) ฐานข้อมูล

ส่วนของฐานข้อมูลเป็นส่วนที่ใช้สำหรับเก็บข้อมูลต่างๆ ที่เกี่ยวข้องกับระบบซึ่งข้อมูลแบ่งออกเป็น 6 ส่วน คือ ข้อมูลผู้ใช้ ข้อมูลโครงงาน ข้อมูลนิยามความต้องการ ข้อมูลคุณลักษณะความต้องการ ข้อมูลนำเข้าของฟังก์ชันงาน และข้อมูลตารางตัดสินใจ

จากกรอบแนวคิดของระบบแต่ละส่วนมีขั้นตอนการทำงานร่วมกันดังนี้

1) การเข้าใช้งานระบบ (Login)

ผู้ใช้ระบบจำเป็นต้องใช้รหัสผู้ใช้ระบบและรหัสผ่านเป็นการยืนยันตัวตนเพื่อเข้าใช้ระบบในส่วนต่างๆ ที่ตนเกี่ยวข้อง

2) การจัดการข้อมูลส่วนตัว (Profile)

ผู้ใช้ระบบสามารถแสดงข้อมูลส่วนตัว และแก้ไขข้อมูลที่ต้องการได้

3) แนะนำวิธีการใช้งานระบบและส่วนช่วยเหลือสนับสนุน (Guideline)

ผู้ใช้ระบบทั้ง 2 กลุ่มหลัก คือนักวิเคราะห์ระบบและผู้ใช้ทั่วไป จำเป็นจะต้องศึกษาวิธีการใช้งานระบบว่ามีการทำงานอย่างไร ประกอบด้วยขั้นตอนใดบ้าง และแต่ละขั้นตอนมีการระบุข้อมูลนำเข้าอย่างไร ทั้งนี้ผู้ใช้สามารถศึกษาวิธีการเพิ่มเติมได้จากตัวอย่าง

4) สร้างโครงงานใหม่ (New project)

นักวิเคราะห์ระบบจะต้องทำการสร้างโครงงานใหม่ทุกครั้งที่มีโครงงานใหม่เข้ามาในระบบ ก่อนที่จะทำการนิยามความต้องการโดยผู้ใช้ทั่วไปหรือนักวิเคราะห์ระบบ

5) นิยามความต้องการ (Requirements definition)

เมื่อนักวิเคราะห์สร้างโครงงานใหม่เรียบร้อยแล้ว ขั้นตอนต่อมาจะเป็นการนิยามความต้องการ ผู้ใช้ทั่วไปจะนิยามความต้องการหรือไม่ก็ได้ ซึ่งหากผู้ใช้ทั่วไปได้ทำการนิยามความต้องการแล้ว นักวิเคราะห์ระบบจะต้องปรับปรุงนิยามความต้องการนั้นให้ถูกต้องอีกครั้ง แต่หากผู้ใช้ทั่วไปไม่ได้นิยามความต้องการ นักวิเคราะห์ระบบจะเป็นผู้นิยามความต้องการเอง

6) ระบุค่าต่าง ๆ ที่จำเป็นต่อการสร้างกรณีทดสอบ

ในขั้นตอนนี้จะประกอบด้วยการทำงานหลัก 3 ขั้นตอน ที่นักวิเคราะห์ระบบจะต้องทำต่อเนื่องจนครบ โดยเริ่มจากกำหนดคุณลักษณะความต้องการ (Requirements specification) ในรูปแบบฟอร์ม จากนั้นระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า (Possible value of input) และเลือกผลลัพธ์ที่คาดหวังของแต่ละกรณีในตารางตัดสินใจ (Decision table)

7) แสดงผลกรณีทดสอบ (Test case) ที่ระบบสร้างให้

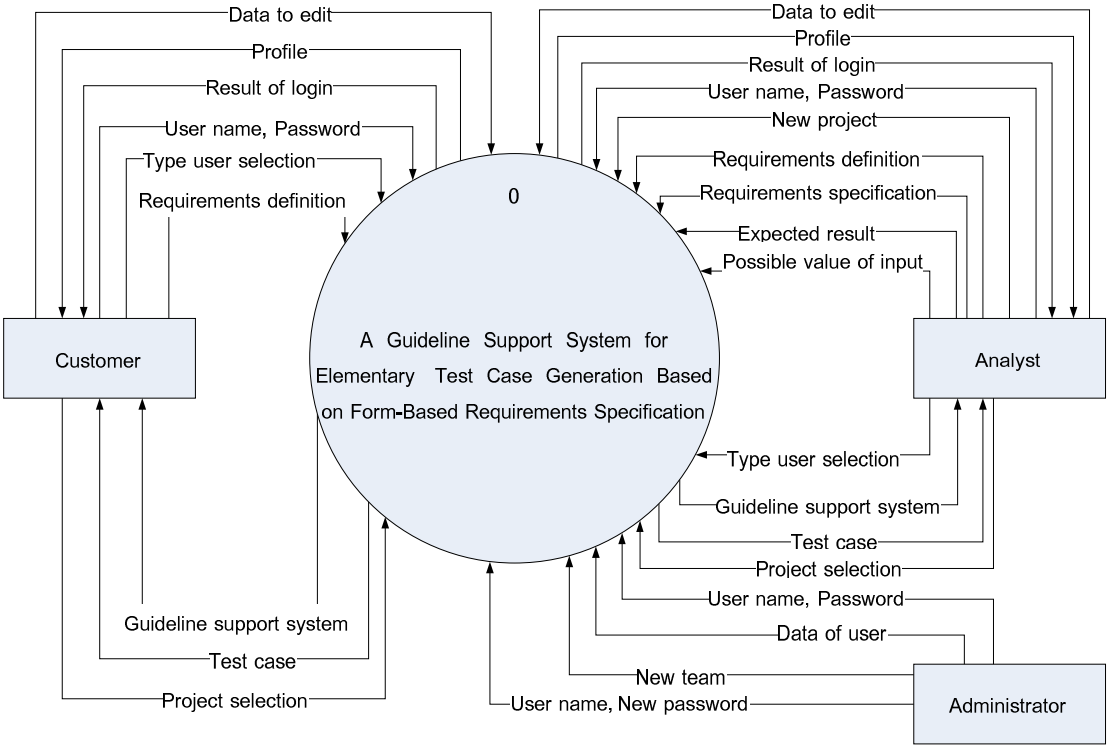
การจะแสดงผลกรณีทดสอบที่ระบบสร้างให้ โครงงานนั้นจะต้องผ่านขั้นตอนการทำงานที่ 6 ครบถ้วนสมบูรณ์ก่อน ผู้ใช้ทั่วไปหรือนักวิเคราะห์จึงจะสามารถเลือกดูผลกรณีทดสอบของโครงงานนั้นได้

8) การจัดการข้อมูลของผู้ใช้

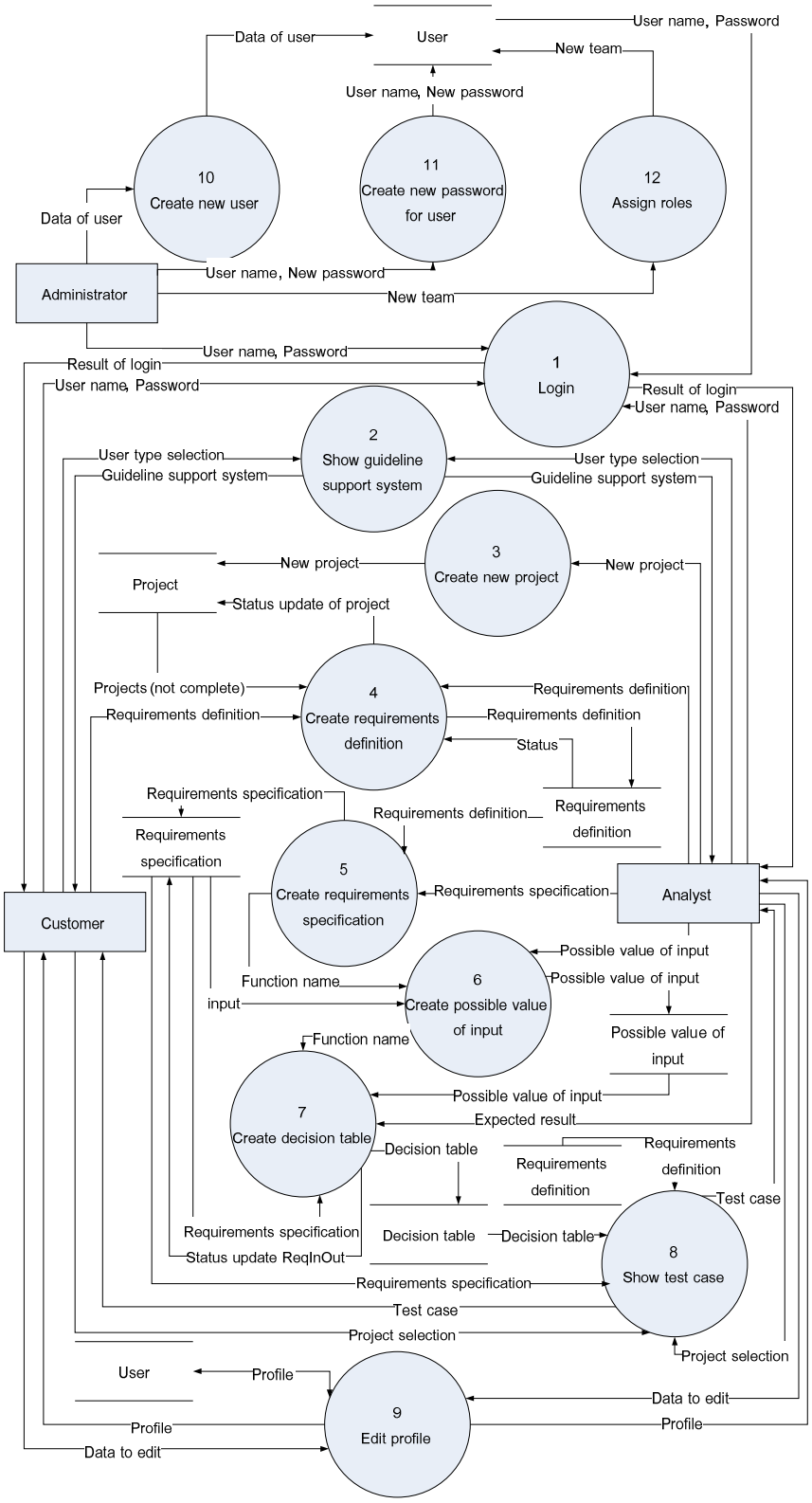
การจัดการข้อมูลผู้ใช้จะเป็นหน้าที่ของผู้ดูแลระบบ โดยจะมีหน้าที่หลัก 3 อย่าง คือ สร้างบัญชีผู้ใช้ให้กับผู้ใช้รายใหม่ (New user) สร้างรหัสผ่านใหม่ (New password) ให้กับผู้ใช้รายเดิมที่ลืมรหัสผ่าน และแก้ไขรายการนักวิเคราะห์ระบบที่สามารถเข้าดำเนินการกับแต่ละโครงงาน (Assign roles)

4.2 การออกแบบกระบวนการทำงานโดยรวม

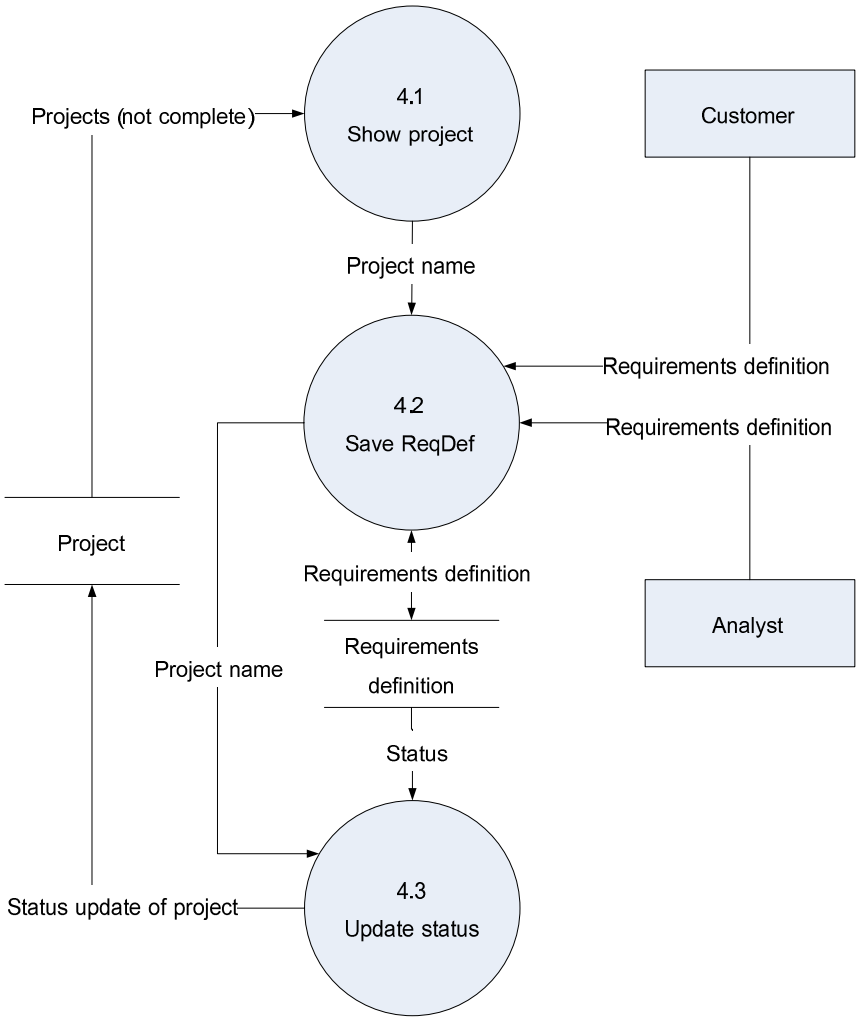
จากการวิเคราะห์ความต้องการในบทที่ 3 สามารถอธิบายกระบวนการทำงานโดยรวมของระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์มด้วยแผนภาพบริบท (Context diagram) และแผนภาพกระแสข้อมูล (Data flow diagram) ดังภาพประกอบ 4.2 ถึงภาพประกอบ 4.8



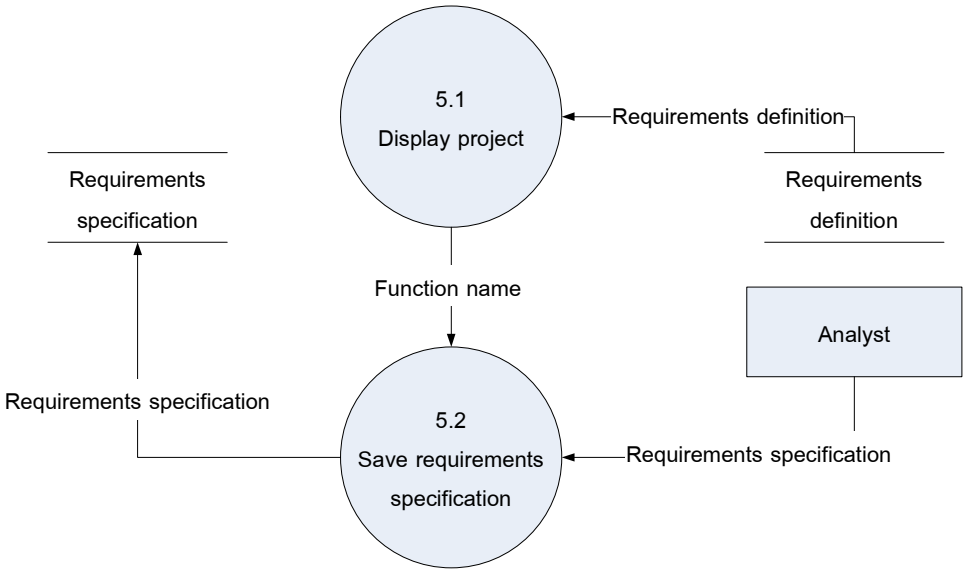
ภาพประกอบ 4.2 แผนภาพบริบทของระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม



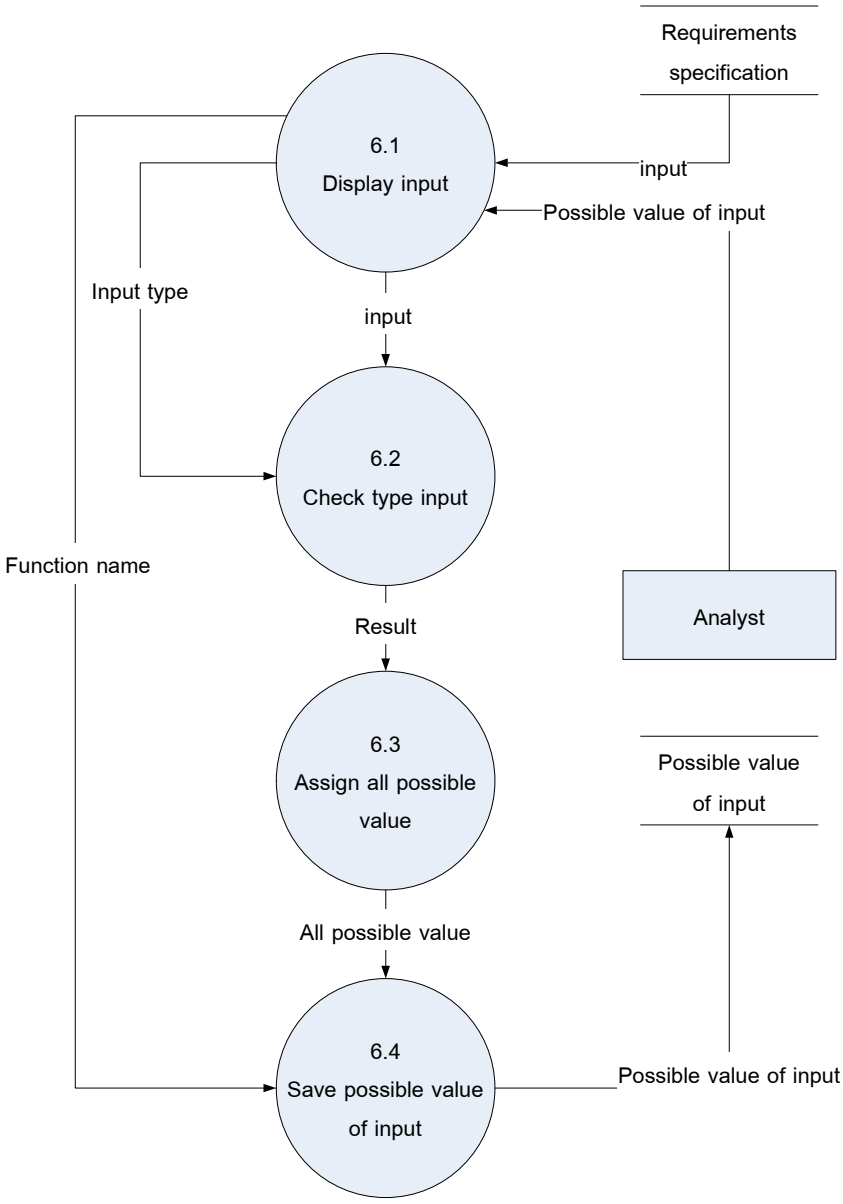
ภาพประกอบ 4.3 แผนภาพกระแสข้อมูลระดับที่ 1 ของระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม



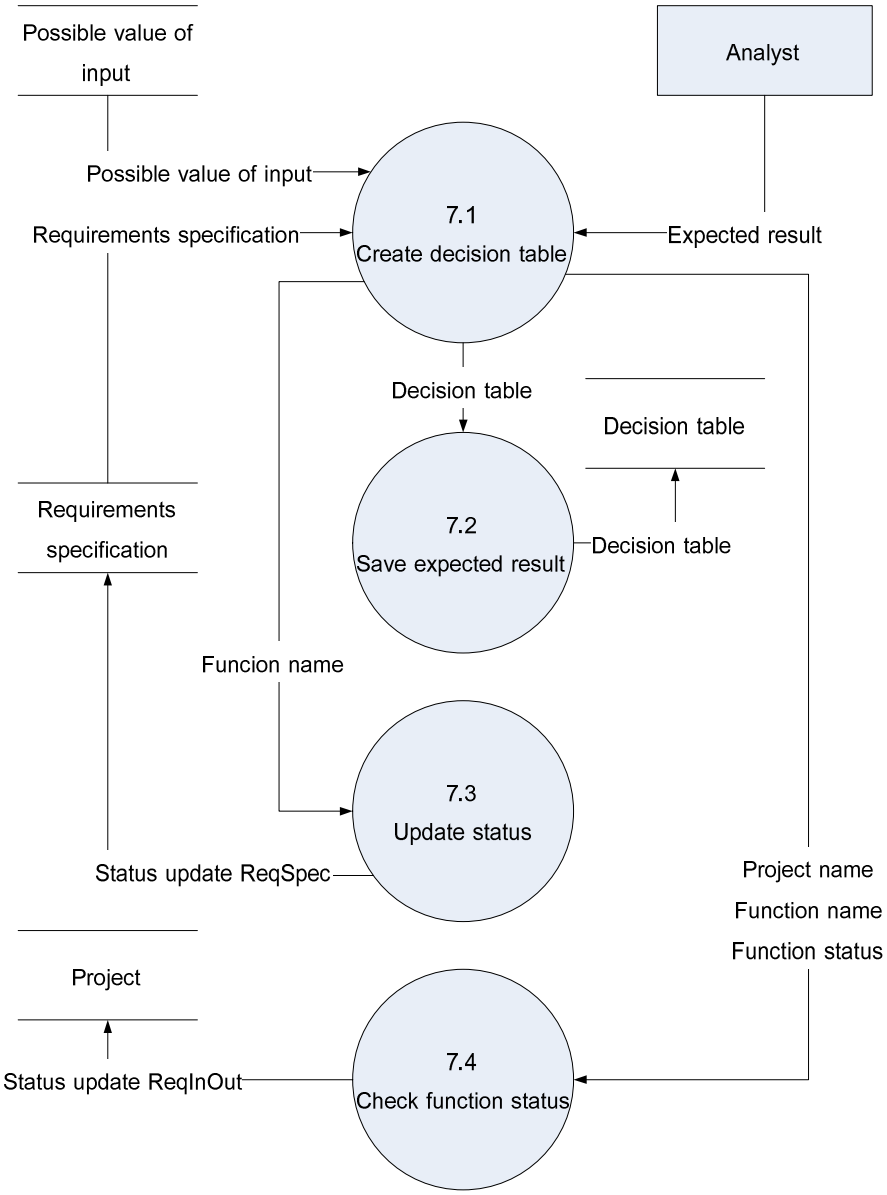
ภาพประกอบ 4.4 แผนภาพกระแสข้อมูลระดับที่ 2 ของการดำเนินการสร้างนิยามความต้องการ



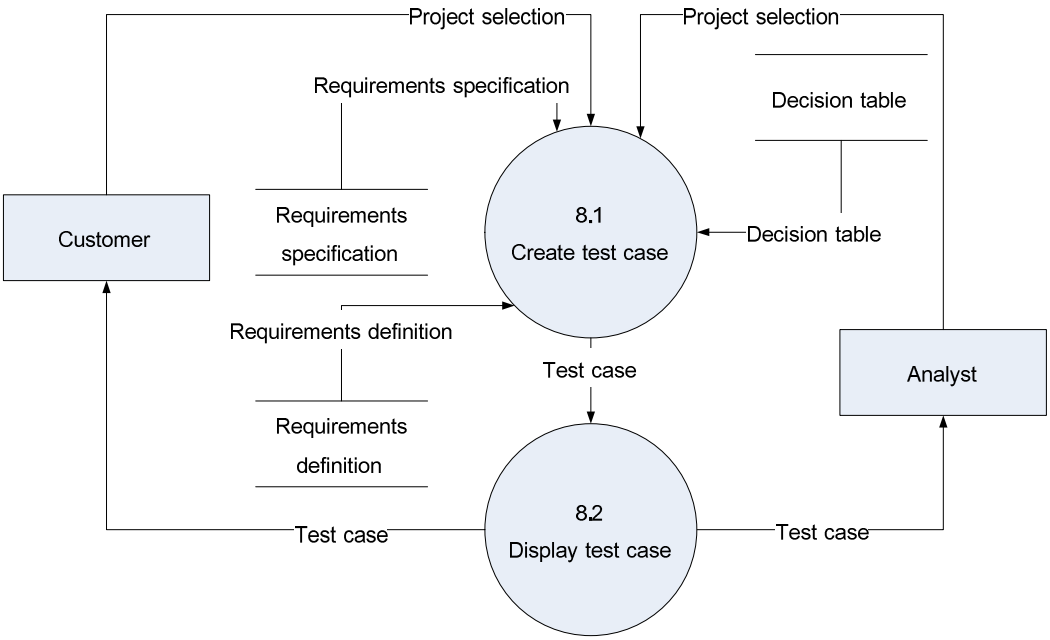
ภาพประกอบ 4.5 แผนภาพกระแสข้อมูลระดับที่ 2 ของการดำเนินการสร้างคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม



ภาพประกอบ 4.6 แผนภาพกระแสข้อมูลระดับที่ 2 ของการดำเนินการระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า



ภาพประกอบ 4.7 แผนภาพกระแสข้อมูลระดับที่ 2 ของการดำเนินการสร้างตารางตัดสินใจ



ภาพประกอบ 4.8 แผนภาพกระแสข้อมูลระดับที่ 2 ของการดำเนินการแสดงกรณีทดสอบ

4.3 การออกแบบโครงสร้างฐานข้อมูลของระบบ

การออกแบบฐานข้อมูลของระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม จะสัมพันธ์กับแผนภาพกระแสข้อมูลของระบบ โดยแสดงโครงสร้างฐานข้อมูลด้วยแผนภาพอี-อาร์ (Entity-Relationship (E-R) Diagram) ได้ดังภาพประกอบ 4.9

จากแผนภาพอี-อาร์ ซึ่งแสดงโครงสร้างฐานข้อมูลเชิงสัมพันธ์ (Relation database structure) สามารถแปลงเป็นแบบแผนเชิงสัมพันธ์ (Relational schema) ที่สมนัย ซึ่งสอดคล้องกับพจนานุกรมข้อมูลที่แสดงดังภาคผนวก ข ดังนี้

User(U_id, U_name, U_userName, U_password, U_mail, U_numPhone, U_type)

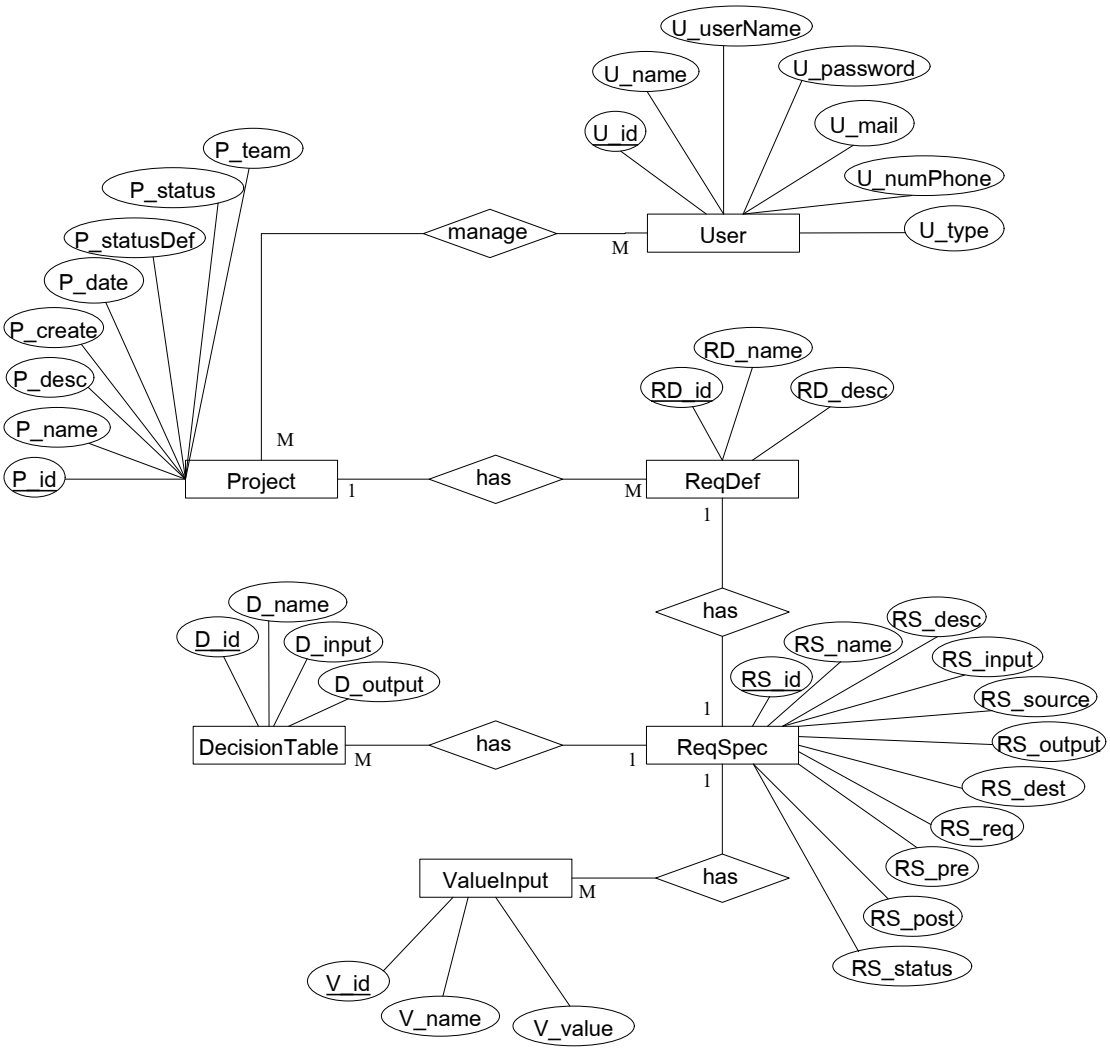
Project(P_id, P_name, P_desc, P_create, P_date, P_team, P_statusDef, P_status)

ReqDef(RD_id, P_id, RD_name, RD_desc, RD_create)

ReqSpec(RS_id, RD_id, RS_name, RS_desc, RS_input, RS_source, RS_output, RS_dest, RS_req, RS_pre, RS_post, RS_status)

ValueInput(V_id, RS_id, V_name, V_value)

DecisionTable(D_id, RS_id, D_name, D_input, D_output)



ภาพประกอบ 4.9 แผนภาพอี-อาร์ ของฐานข้อมูลระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม

4.4 การออกแบบโครงสร้างของระบบ

จากการออกแบบกระบวนการทำงานโดยรวม และสถาปัตยกรรมของต้นแบบระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม จะได้โครงสร้างของระบบในภาพรวมสำหรับผู้ใช้งานแต่ละกลุ่มดังแสดงในภาพประกอบ 4.10

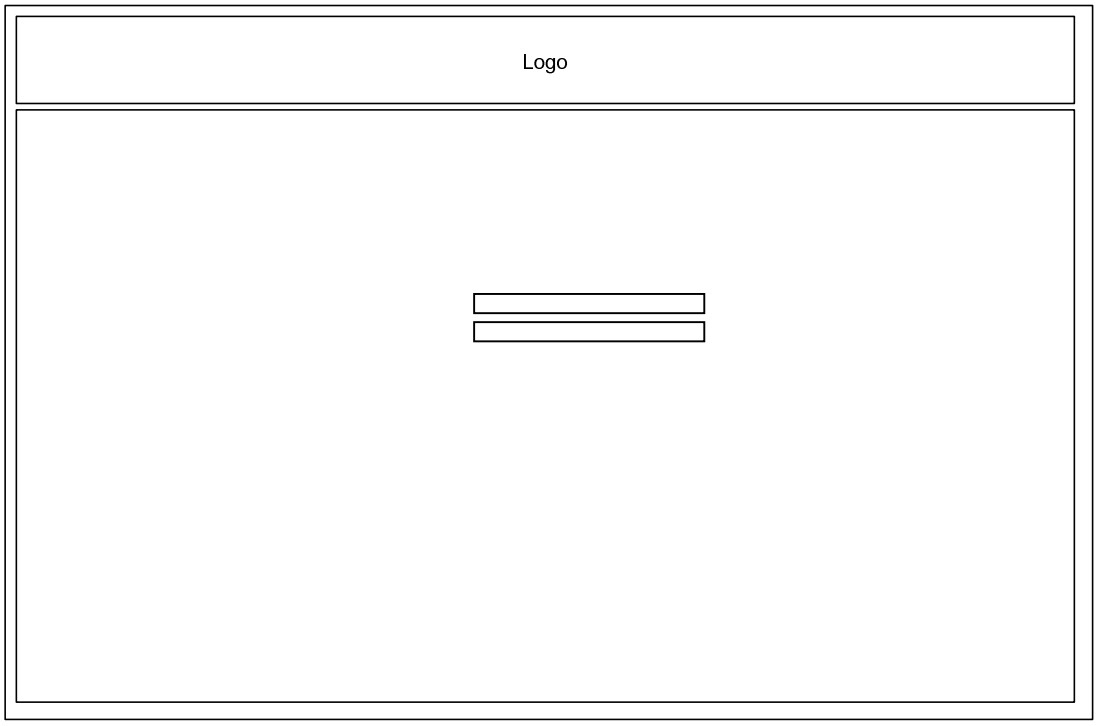
ภาพประกอบ 4.10 โครงสร้างของระบบในภาพรวม

4.5 การออกแบบส่วนติดต่อกับผู้ใช้

การดำเนินงานและการแสดงผลต่างๆ ของระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม ได้รับการพัฒนาให้มีฟังก์ชันรองรับการใช้งานตามที่ได้ออกแบบไว้ข้างต้น โดยมีรายละเอียดของส่วนต่างๆ ดังต่อไปนี้

4.5.1 การเข้าใช้งานระบบ

ผู้ใช้ทุกกลุ่มจะต้องใช้ชื่อผู้ใช้และรหัสผ่านในการยืนยันตัวตนเพื่อเข้าสู่การใช้งานระบบในส่วนต่างๆ ที่ตนเกี่ยวข้อง แสดงดังภาพประกอบ 4.11

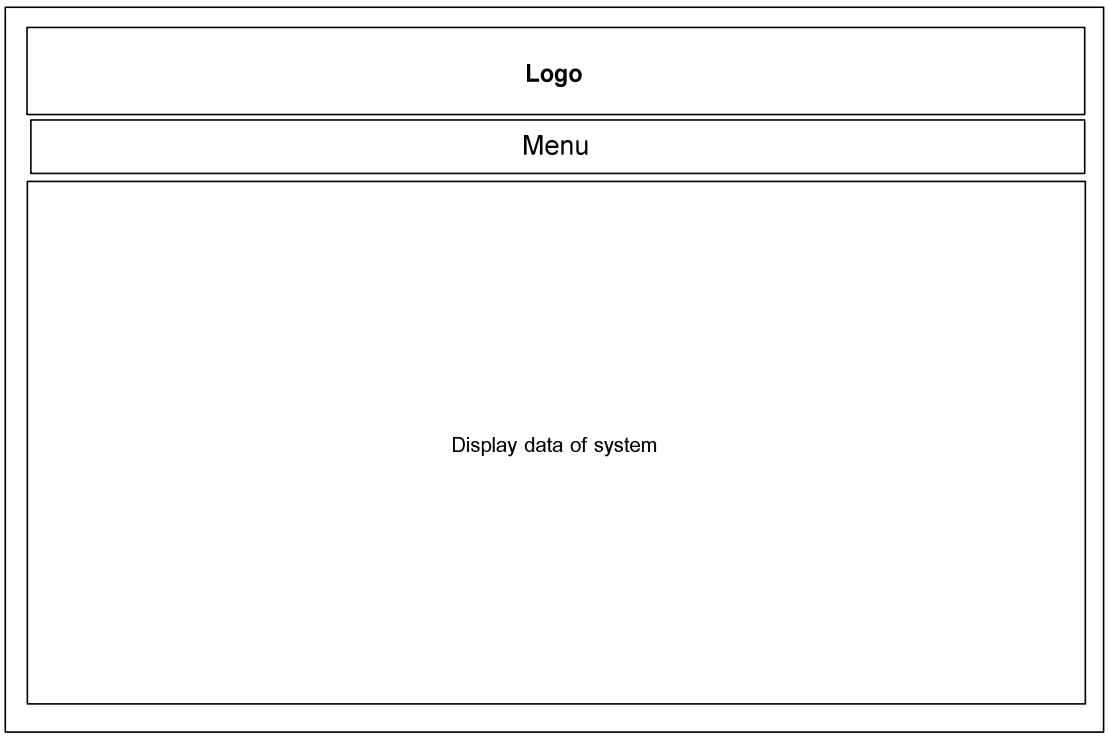


ภาพประกอบ 4.11 หน้าจอการเข้าใช้งานระบบ

4.5.2 เมนูหลักของระบบ

เมนูหลักของระบบจะประกอบด้วย 7 ส่วนหลักๆ คือ (1) เมนู “New Project” เพื่อสร้างโครงงานใหม่ (2) เมนู “Requirements Definition” เพื่อนิยามความต้องการ (3) เมนู

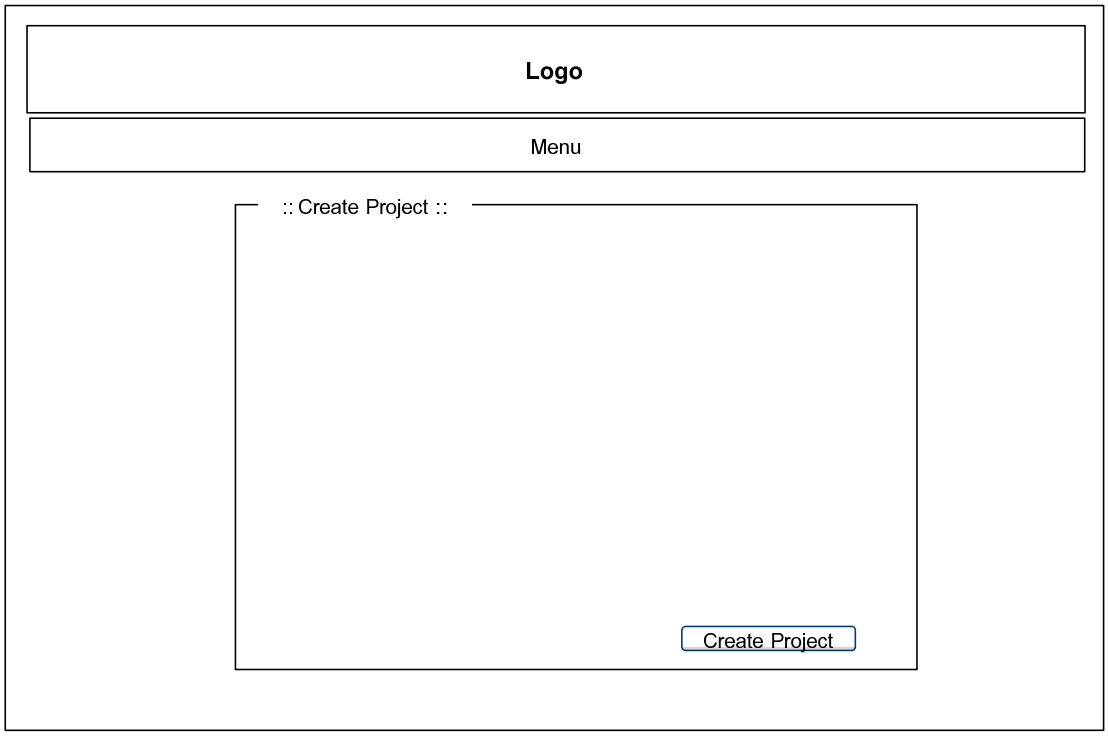
“Requirements Specification and Possible Value” เพื่อกำหนดคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม จากนั้นก็ระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า แล้วกำหนดผลลัพธ์ที่คาดหวังของแต่ละกรณีในตารางตัดสินใจตามลำดับ (4) เมนู “Test Case” เพื่อเลือกแสดงกรณีทดสอบของโครงงานที่ต้องการ และ (5) เมนู “User Manual” เป็นส่วนของการอธิบายการทำงานของระบบและขั้นตอนต่างๆ พร้อมทั้งตัวอย่างประกอบแยกตามกลุ่มผู้ใช้งาน (6) เมนู “Profile” เป็นส่วนของการแสดงข้อมูลผู้ใช้ ผู้ใช้สามารถปรับปรุงข้อมูลส่วนตัวได้ (7) เมนู “Logout” เป็นส่วนของการออกจากระบบเมื่อเสร็จสิ้นการทำงาน หน้าจอแรกของระบบแสดงดังภาพประกอบ 4.12 ทั้งนี้กลุ่มผู้ใช้ที่เป็นนักวิเคราะห์ระบบจะสามารถใช้งานเมนูต่างๆ ได้ทั้งหมด แต่กลุ่มผู้ใช้ที่เป็นผู้ใช้งานทั่วไปจะไม่สามารถใช้งานเมนู “New Project” และเมนู “Requirements Specification and Possible Value” เพราะส่วนนี้มีไว้สำหรับนักวิเคราะห์ระบบเท่านั้น



ภาพประกอบ 4.12 หน้าจอหลักของระบบ

4.5.3 การสร้างโครงงานใหม่

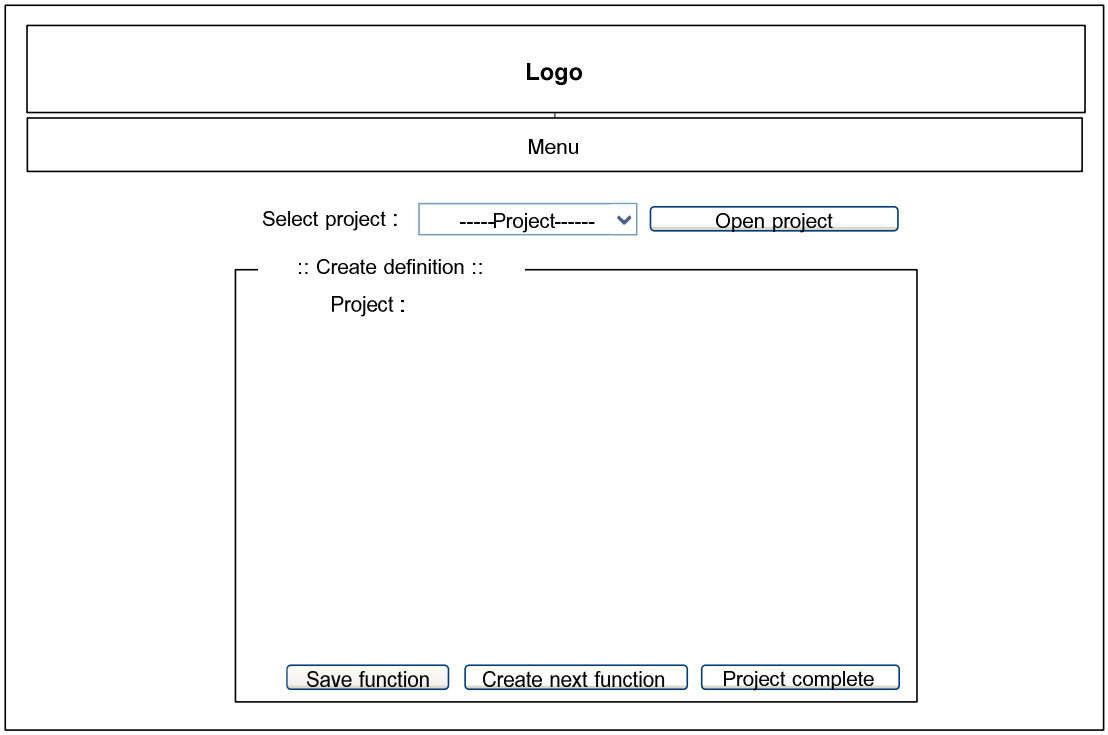
เมื่อต้องการสร้างโครงงานใหม่ นักวิเคราะห์ระบบจะเลือกเมนู “New Project” จากนั้นทำการกรอกข้อมูลให้ครบถ้วน แล้วกดปุ่ม “Create Project” เพื่อบันทึกโครงงาน ดังภาพประกอบ 4.13



ภาพประกอบ 4.13 หน้าจอสำหรับสร้างโครงงานใหม่โดยนักวิเคราะห์ระบบ

4.5.4 การนิยามความต้องการ

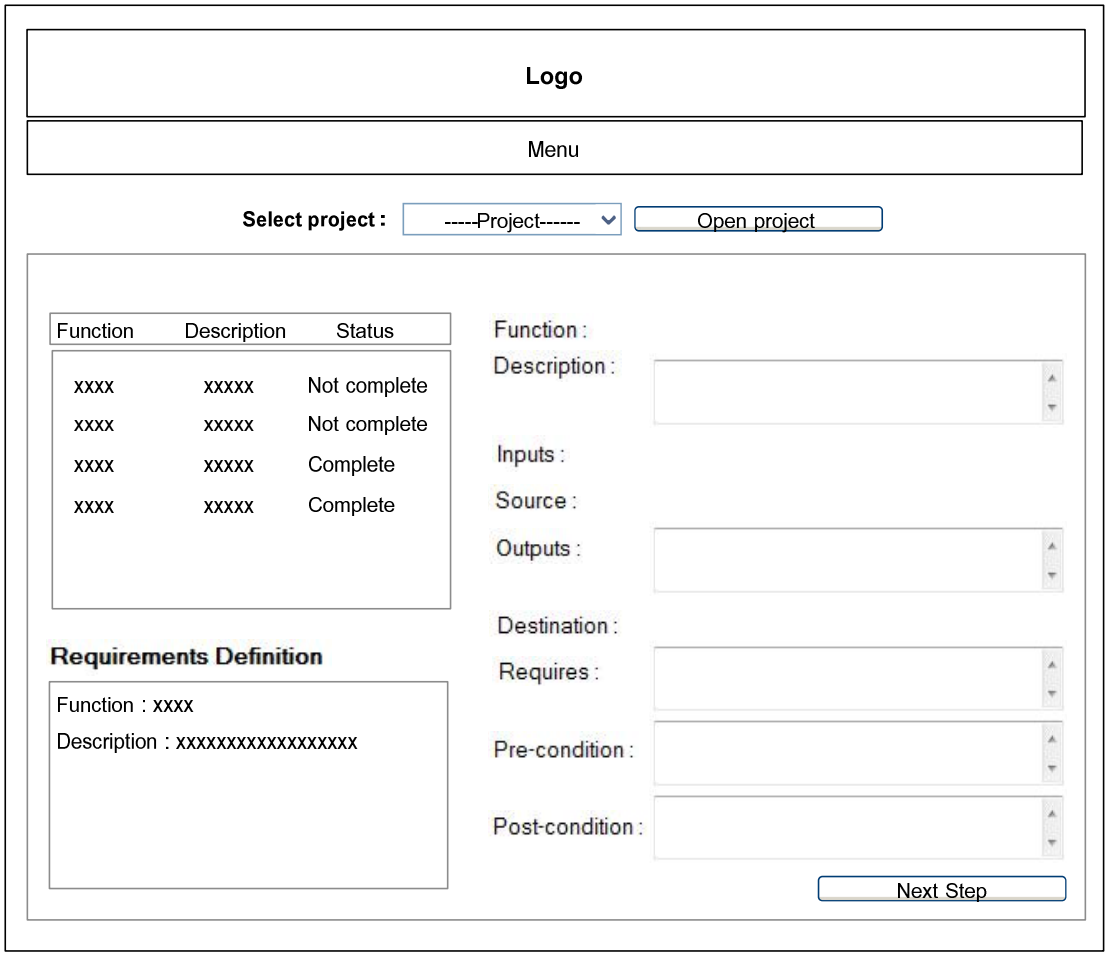
ในส่วนของการนิยามความต้องการสามารถเข้าใช้งานได้ทั้งผู้ใช้ทั่วไปและนักวิเคราะห์ระบบ โดยเลือกที่เมนู “Requirements Definition” จากนั้นทำการเลือกโครงงานที่ต้องการ แล้วเลือกปุ่ม “Open project” เมื่อกรอกข้อมูลครบถ้วนแล้วให้กดปุ่ม “Save function” จากนั้นกดปุ่ม “Create next function” เพื่อนิยามความต้องการให้กับฟังก์ชันงานต่อไป ทำอย่างนี้ไปเรื่อยๆ จนกว่าจะกรอกข้อมูลครบทุกฟังก์ชันในโครงงาน เมื่อกรอกข้อมูลครบทุกฟังก์ชันแล้ว จึงกดปุ่ม “Project complete” เพื่อจบการทำงานในส่วนนี้ ดังภาพประกอบ 4.14



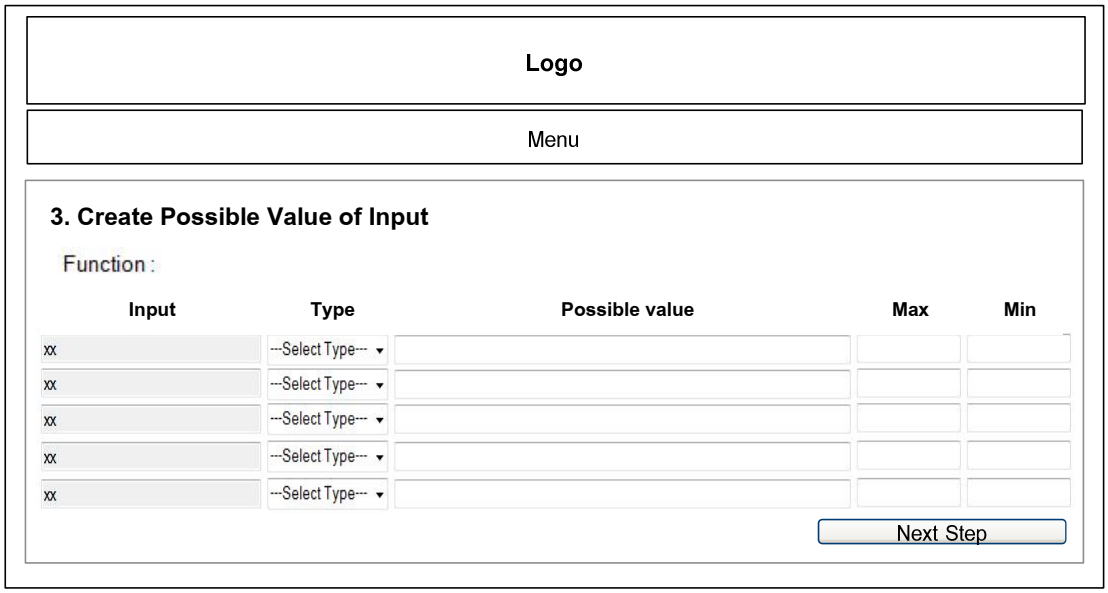
ภาพประกอบ 4.14 หน้าจอสำหรับสร้างนิยามความต้องการโดยผู้ใช้ทั่วไป หรือนักวิเคราะห์ระบบ

4.5.5 การกำหนดคุณลักษณะความต้องการ ระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า และกำหนดผลลัพธ์ที่คาดหวังในตารางตัดสินใจ

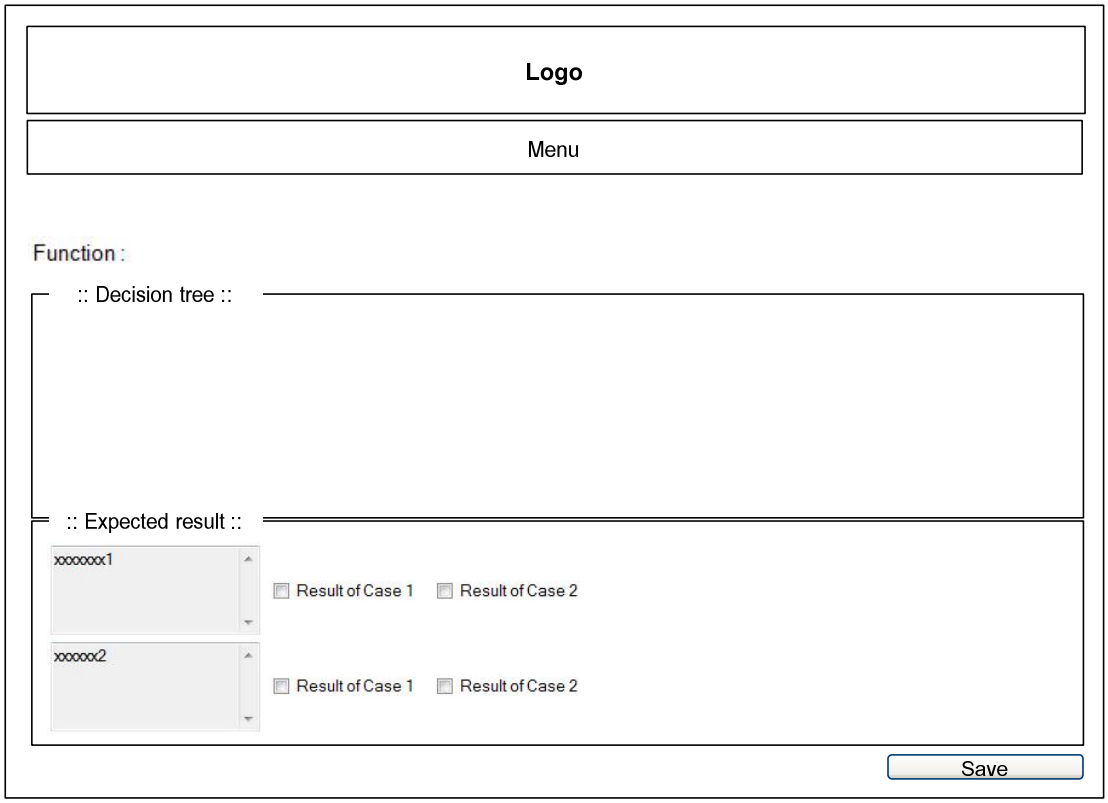
ขั้นตอนนี้ถือเป็นส่วนสำคัญอย่างยิ่งที่จะได้มาซึ่งกรณีทดสอบที่มีประสิทธิภาพ จำเป็นต้องใช้นักวิเคราะห์ระบบดำเนินการในส่วนนี้เท่านั้น โดยการเลือกเมนู “Requirements Specification and Possible Value” จากนั้นให้ทำการเลือกโครงงานและฟังก์ชันงานที่ต้องการระบุ โดยเริ่มจากการกำหนดคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม ดังภาพประกอบ 4.15 จากนั้นระบุชนิดและค่าข้อมูลที่เป็นไปได้ของข้อมูลนำเข้า ดังภาพประกอบ 4.16 แล้วกำหนดผลลัพธ์ที่คาดหวังของแต่ละกรณีในตารางตัดสินใจดังภาพประกอบ 4.17



ภาพประกอบ 4.15 หน้าจอสำหรับสร้างคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม



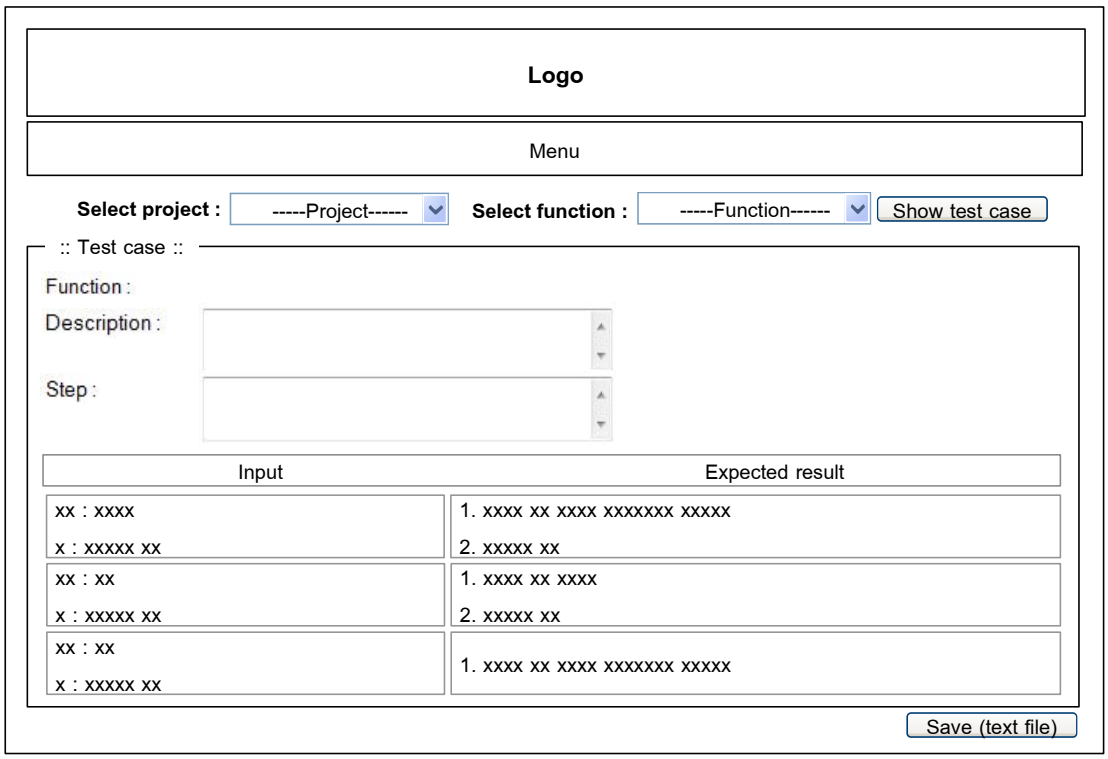
ภาพประกอบ 4.16 หน้าจอสำหรับระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า



ภาพประกอบ 4.17 หน้าจอสำหรับระบุผลลัพธ์ที่คาดหวังในตารางตัดสินใจ

4.5.6 การแสดงกรณีทดสอบของโครงงานที่ต้องการ

ผู้ใช้ทั้งสองกลุ่มสามารถเลือกดูกรณีทดสอบที่ต้องการได้โดยกดเลือกเมนู “Test Case” จากนั้นเลือกโครงงานและฟังก์ชันงานที่ต้องการแสดงกรณีทดสอบ ดังภาพประกอบ 4.18 และสามารถบันทึกข้อมูลกรณีทดสอบของฟังก์ชันงานที่เลือกในรูป Text file ได้โดยการกดปุ่ม “Save (text file)” แสดงตัวอย่างแฟ้มข้อมูลที่ได้ดังภาพประกอบ 4.19



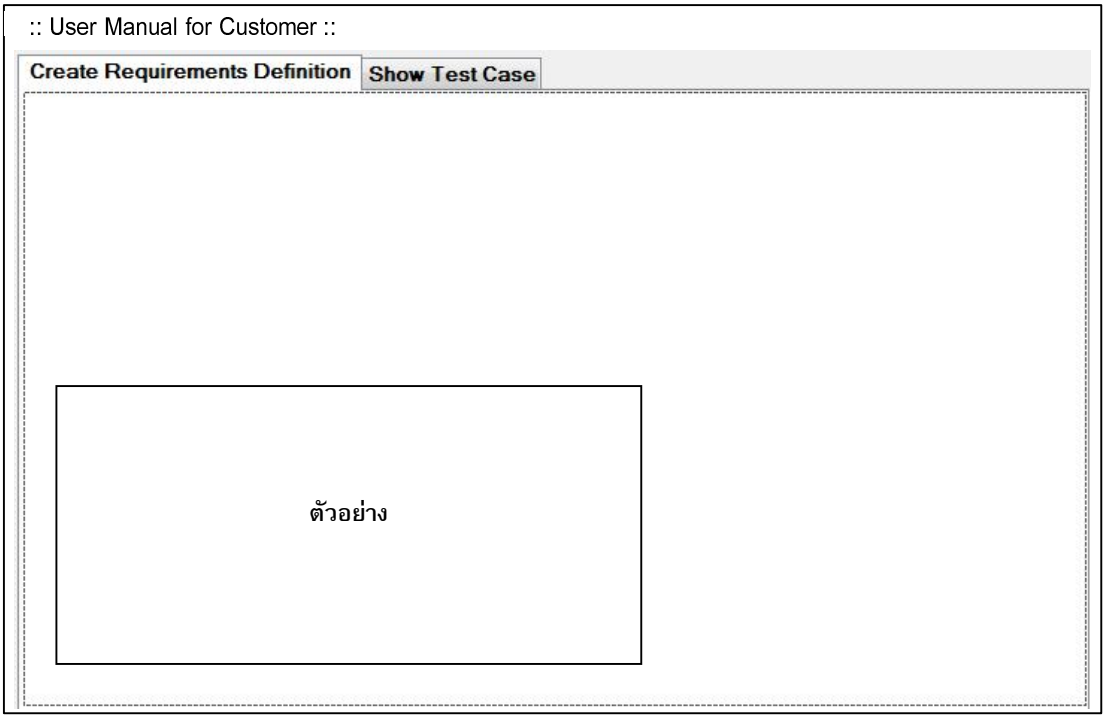
ภาพประกอบ 4.18 หน้าจอสำหรับแสดงกรณีทดสอบที่สร้างโดยระบบ

```
================================ Test Case ================================

Test Case:
Deposit

Description:
A customer chooses the deposit button, then payment and confirm. The ATM will
check the balance and display to the customer. The ATM will accept a deposit
from 100 Baht to 100,000 Baht. When the customer presses the deposit
confirmation. If condition of deposit is true, so the ATM will displays the
total deposit and net balance. Then the ATM prints a receipt and returns a cash
card. But it is not conditions then the ATM displays a warning message and
returns the cash card.

Step:
Pre-condition : The monitor displays transactions for a customer select.
1. A customer chooses deposit. Then the payment and confirm.
2. The ATM will check the balance and display to a customer.
3. If condition of deposit is true, so the ATM will displays the total deposit
and net balance in the customer account. Then the ATM prints a receipt and
return cash card.
4. But it is not conditions, the ATM displays a warning message and return cash
card.

--------------------------------------------------------------------------------
#Test case 1
Input:
deposit : 50
Expected Result:
1. The ATM displays a warning message.
2. Returns a cash card.

--------------------------------------------------------------------------------
#Test case 2
Input:
deposit : 100
Expected Result:
1. The ATM displays the total deposit and net balance in the customer account.
2. Prints a receipt..
3. Returns a cash card.

--------------------------------------------------------------------------------
```

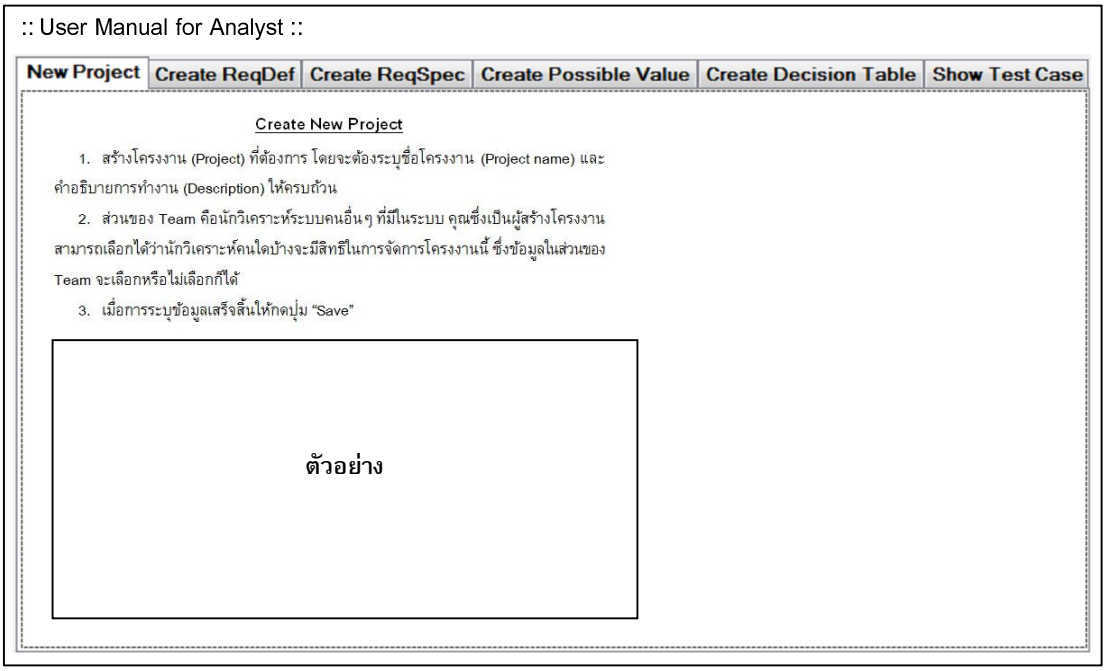
ภาพประกอบ 4.19 ตัวอย่างแฟ้มข้อมูลของกรณีทดสอบในรูป Text file

4.5.7 การแสดงส่วนช่วยเหลือสำหรับผู้ใช้

ผู้ใช้ทั้งสองกลุ่มสามารถศึกษาวิธีการใช้งานระบบ คำอธิบายขั้นตอนการทำงานต่างๆ ของระบบ พร้อมตัวอย่างประกอบ โดยการกดเลือกเมนู “User Manual” จากนั้นระบบจะแสดงคู่มือการใช้งานระบบให้ผู้ใช้ได้ทำการศึกษาตามกลุ่มของผู้ใช้ ได้แก่ คู่มือการใช้งานระบบสำหรับผู้ใช้ทั่วไป ดังตัวอย่างในภาพประกอบ 4.20 และคู่มือการใช้งานระบบสำหรับนักวิเคราะห์ระบบ ดังตัวอย่างในภาพประกอบ 4.21



ภาพประกอบ 4.20 หน้าจอสำหรับแสดงคู่มือการใช้งานระบบสำหรับผู้ใช้ทั่วไป



ภาพประกอบ 4.21 หน้าจอสำหรับแสดงคู่มือการใช้งานระบบสำหรับนักวิเคราะห์ระบบ

บทที่ 5

การพัฒนาระบบ

5.1 เครื่องมือซอฟต์แวร์ที่ใช้พัฒนาระบบ

Visual Studio C# 2008 Express Edition

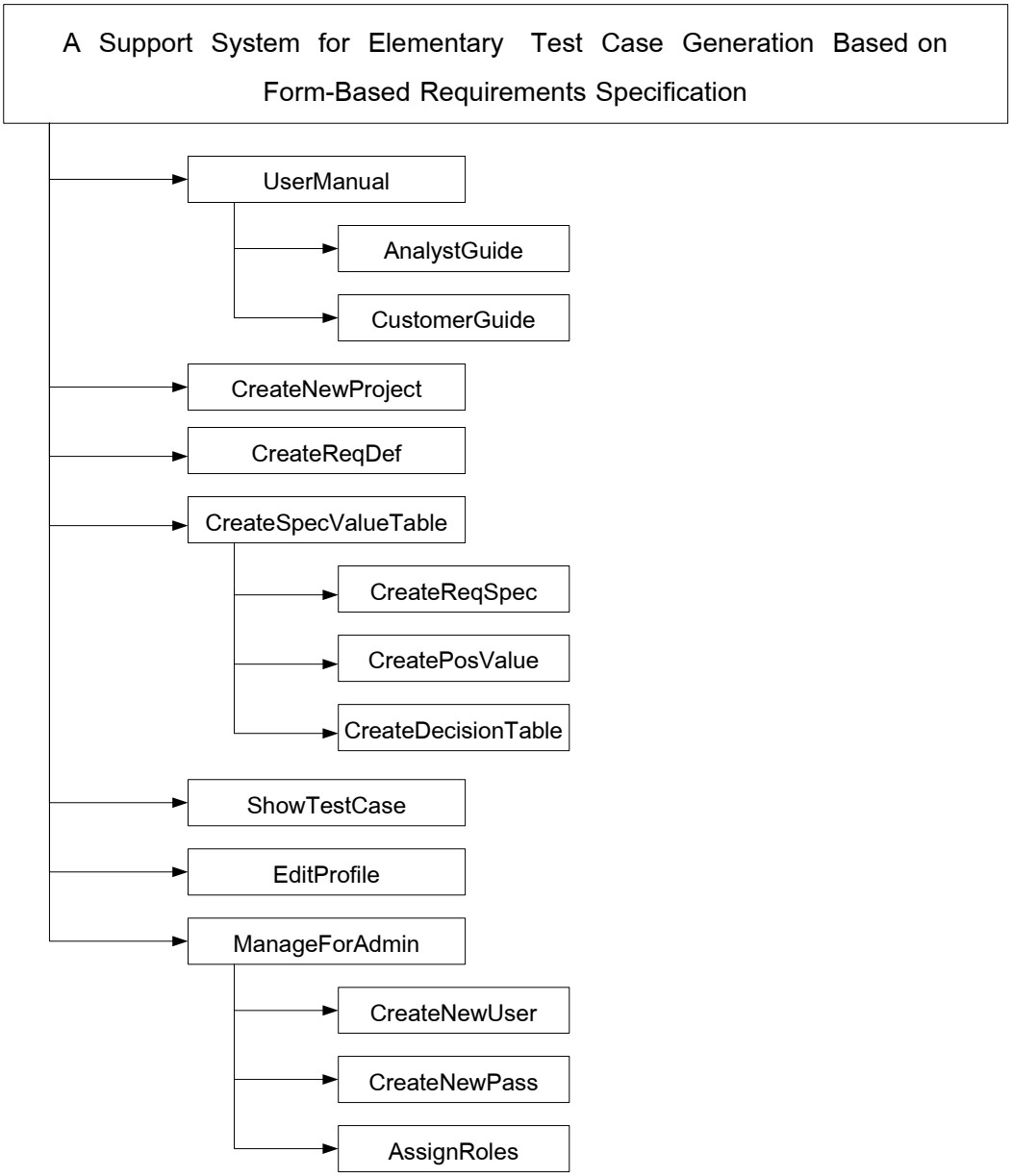
Visual Studio คือ ซอฟต์แวร์ประเภท IDE (Integrated development Environment) ช่วยให้ผู้พัฒนาโปรแกรมสามารถเขียนโปรแกรมด้วยความสะดวกรวดเร็วขึ้น สามารถแก้ไขข้อผิดพลาดในการเขียนโปรแกรมได้ง่าย เนื่องจากเป็นการเขียนโปรแกรมในลักษณะเชิงทัศน์ (Visual programming)

C# เป็นภาษาคอมพิวเตอร์ที่พัฒนาโดยบริษัทไมโครซอฟต์ โดยมีรากฐานของภาษา C/C++ เป็นหลัก ดังนั้นรูปแบบโครงสร้างทางภาษาของ C# จึงคล้ายกับ C/C++ และยังแก้ไขปัญหาและข้อบกพร่องหลายประการที่มีอยู่ใน C++ ให้หมดไป จึงทำให้ภาษา C# นั้นกลายเป็นภาษาที่เรียนรู้ได้ง่ายและสามารถทำงานได้อย่างมีประสิทธิภาพ

เนื่องจาก C# นั้นเกิดขึ้นมาพร้อมกับเทคโนโลยี .NET ดังนั้นการทำงานของ C# จึงขึ้นกับ .NET Framework เป็นหลัก โดยมีชุดเครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชั่นด้วยภาษา C# เรียกว่า Visual C# ซึ่งสามารถใช้ในการพัฒนาแอปพลิเคชั่นทั่วๆ ไปในระดับเดียวกับ Visual Basic เพื่อให้ผู้เขียนโปรแกรมที่คุ้นเคยกับรูปแบบโครงสร้างในแบบ C/C++ สามารถเลือกใช้ VC# แทน VB ได้

5.2 การพัฒนาระบบ

จากการที่ได้วิเคราะห์และออกแบบระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม สามารถนำมาพัฒนาระบบโดยแยกการทำงานของระบบออกเป็นโมดูล ซึ่งสามารถแสดงโครงสร้างดังภาพประกอบ 5.1



ภาพประกอบ 5.1 โครงสร้างของระบบที่พัฒนาโดย Visual Studio C# 2008 Express Edition

จากภาพประกอบ 5.1 สามารถอธิบายการทำงานแต่ละโมดูลได้ดังนี้

5.2.1 UserManual

เป็นโมดูลที่ใช้สำหรับแนะนำการใช้งานในส่วนต่างๆ ของระบบ ซึ่งประกอบด้วย 2 โมดูลย่อย ดังแสดงในตาราง 5.1

ตาราง 5.1 รายชื่อโมดูลย่อยในโมดูลแนะนำการใช้งานในส่วนต่างๆ ของระบบ

ชื่อโมดูลย่อย	คำอธิบาย
AnalystGuide	เป็นโมดูลสำหรับแนะนำการใช้งานระบบให้กับนักวิเคราะห์ระบบ
CustomerGuide	เป็นโมดูลสำหรับแนะนำการใช้งานระบบให้กับผู้ใช้ทั่วไป

5.2.2 CreateNewProject

เป็นโมดูลสำหรับการสร้างโครงงานใหม่โดยนักวิเคราะห์ระบบ

5.2.3 CreateReqDef

เป็นโมดูลสำหรับการนิยามความต้องการของผู้ใช้หรือนักวิเคราะห์ระบบ

5.2.4 CreateSpecValueTable

เป็นโมดูลสำหรับการกำหนดค่าต่างๆ ที่จำเป็นในการสร้างกรณีทดสอบเบื้องต้นให้กับระบบ โดยมีโมดูลย่อยดังแสดงในตาราง 5.2

ตาราง 5.2 รายชื่อโมดูลย่อยในโมดูลการกำหนดค่าต่างๆ ที่จำเป็นในการสร้างกรณีทดสอบเบื้องต้นให้กับระบบ

ชื่อโมดูลย่อย	คำอธิบาย
CreateReqSpec	เป็นโมดูลสำหรับการกำหนดคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม
CreatePosValue	เป็นโมดูลสำหรับการระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า
CreateDecisionTable	เป็นโมดูลสำหรับการกำหนดผลลัพธ์ที่คาดหวังของแต่ละกรณีในตารางตัดสินใจ

5.2.5 ShowTestCase

เป็นโมดูลสำหรับการดำเนินการแสดงกรณีทดสอบเบื้องต้นที่สร้างโดยระบบ

5.2.6 EditProfile

เป็นโมดูลสำหรับการแสดงและแก้ไขข้อมูลผู้ใช้

5.2.7 ManageForAdmin

เป็นโมดูลสำหรับการจัดการข้อมูลต่างๆ ของผู้ใช้โดยผู้ดูแลระบบ โดยมีโมดูลย่อยดังแสดงในตาราง 5.3

ตาราง 5.3 รายชื่อโมดูลย่อยในโมดูลสำหรับการจัดการข้อมูลต่างๆ ของผู้ใช้โดยผู้ดูแลระบบ

ชื่อโมดูลย่อย	คำอธิบาย
CreateNewUser	เป็นโมดูลสำหรับการสร้างบัญชีผู้ใช้
CreateNewPass	เป็นโมดูลสำหรับการสร้างรหัสผ่านใหม่ให้กับผู้ใช้ ในกรณีที่ผู้ใช้ลืมรหัสผ่าน
AssignRoles	เป็นโมดูลสำหรับการแก้ไขรายการนักวิเคราะห์ระบบที่สามารถเข้าดำเนินการกับแต่ละโครงงาน

บทที่ 6

การประเมินประสิทธิภาพ

เนื้อหาในบทนี้กล่าวถึงการประเมินประสิทธิภาพของงานวิจัย ซึ่งถูกดำเนินการเป็น 2 ส่วนหลัก ดังนี้

1) การประเมินกรณีทดสอบ

ในส่วนนี้เป็นการคัดเลือกผู้เชี่ยวชาญทางด้านการวิเคราะห์และพัฒนาระบบงานคอมพิวเตอร์ เพื่อสร้างกรณีทดสอบสำหรับกรณีศึกษาตามขอบเขตของงานวิจัย โดยเป็นการคัดเลือกแบบเจาะจง

2) การประเมินความพึงพอใจของผู้ใช้ต่อระบบโดยรวม

ในส่วนนี้เป็นการประเมินระบบโดยตัวแทนของกลุ่มผู้ใช้ประเภทต่างๆ ที่คัดเลือกมาแบบเจาะจง เพื่อประเมินการใช้งานระบบและความพึงพอใจที่มีต่อระบบ

6.1 การประเมินประสิทธิผลของกรณีทดสอบ

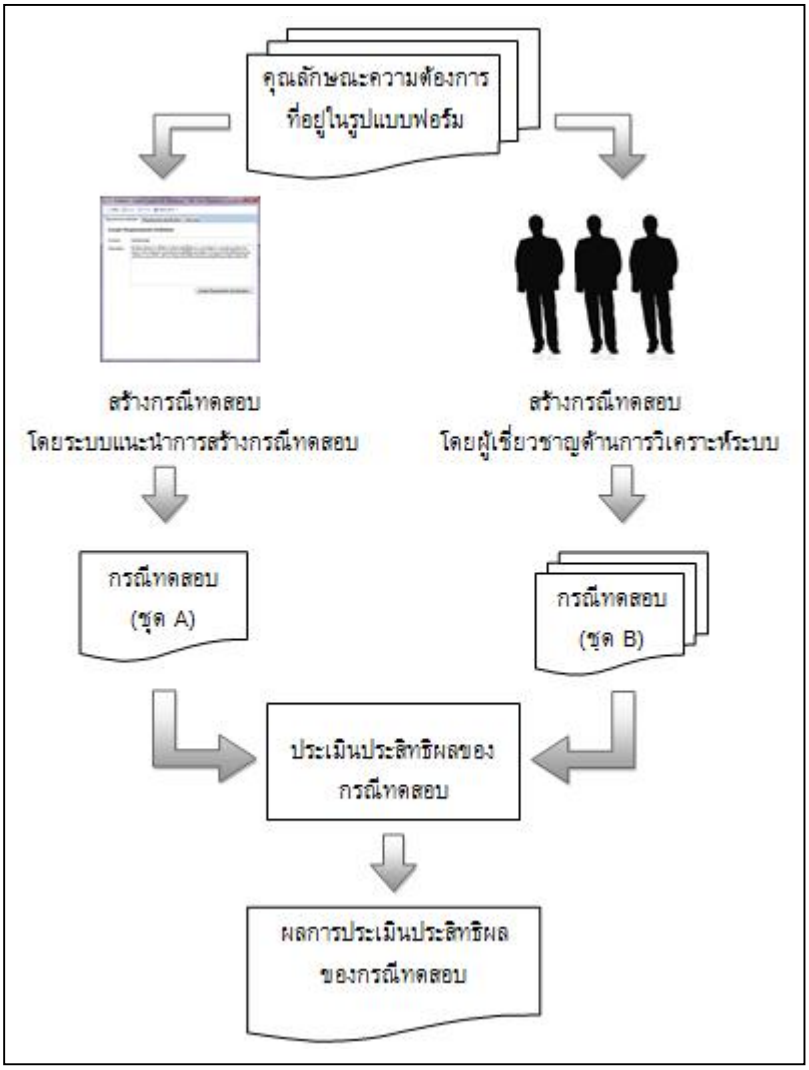
แม้ว่าระบบที่ได้รับการพัฒนาขึ้นใช้งานจะสามารถสร้างโครงงาน นิยามและกำหนดคุณลักษณะความต้องการ เพื่อสร้างกรณีทดสอบได้ทั้งภาษาไทยและภาษาอังกฤษ แต่ตามขอบเขตของงานวิจัยนี้ได้กำหนดระบบกรณีตัวอย่าง คือ ระบบสำหรับการประมวลผลเครื่องฝาก – ถอนเงินอัตโนมัติอย่างง่าย (Simple ATM System) ไว้เป็นกรณีศึกษาขั้นพื้นฐาน (Based-line case study) เพื่อเปรียบเทียบประสิทธิผลการสร้างกรณีทดสอบโดยนักวิเคราะห์ระบบกับกรณีทดสอบที่สร้างโดยระบบที่พัฒนาขึ้น

การประเมินประสิทธิผลของกรณีทดสอบที่สร้างจากระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม มีขั้นตอนการดำเนินงานและผลจากการดำเนินงานดังนี้

6.1.1 ขั้นตอนการประเมินประสิทธิผลของกรณีทดสอบ

ในงานวิจัยนี้ได้ประเมินประสิทธิผลของกรณีทดสอบโดยผู้เชี่ยวชาญซึ่งเป็นนักวิเคราะห์ระบบ โดยการให้ผู้เชี่ยวชาญสร้างกรณีทดสอบจากคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม โดยใช้กรณีศึกษา คือระบบฝาก-ถอนเงินอัตโนมัติอย่างง่ายในการ

ดำเนินงานครั้งนี้ แล้วนำคุณลักษณะความต้องการที่เขียนในรูปแบบฟอร์มชุดเดียวกันนี้ มาสร้างกรณีทดสอบด้วยระบบที่พัฒนาขึ้น จากนั้นนำกรณีทดสอบที่ได้ทั้ง 2 ชุด มาทำการประเมินประสิทธิผลของกรณีทดสอบที่ได้จากระบบ (ชุด A) เปรียบเทียบกับกรณีทดสอบที่สร้างโดยผู้เชี่ยวชาญ (ชุด B) ดังภาพประกอบ 6.1



ภาพประกอบ 6.1 ขั้นตอนการประเมินประสิทธิผลของกรณีทดสอบที่สร้างโดยระบบ

6.1.2 การออกแบบการประเมินกรณีทดสอบและทดลองใช้งานเบื้องต้น

ก่อนมีการประเมินกรณีทดสอบจริงได้นำแบบประเมินที่สร้างขึ้นไปทดลองใช้กับนักศึกษาปริญญาโท ของภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์

จำนวน 4 คน โดยใช้วิธีการตรวจให้คะแนนการประเมินประสิทธิผลของการสร้างกรณีทดสอบสำหรับระบบกรณีตัวอย่าง เช่นเดียวกับที่ใช้ในการทดลองประเมินจริงกับผู้เชี่ยวชาญที่เป็นนักวิเคราะห์ระบบ

ในการตรวจสอบเพื่อเปรียบเทียบกรณีทดสอบที่สร้างด้วยนักวิเคราะห์กับกรณีทดสอบที่สร้างโดยระบบ ได้กำหนดคะแนนจากจำนวนข้อมูลนำเข้า (Input) และผลลัพธ์ที่คาดหวัง (Expected result) เป็น 1 คะแนนต่อรายการ เช่น ฟังก์ชันงาน Login ดังตาราง 6.1 จะมีจำนวนข้อมูลนำเข้า 4 รายการ และผลลัพธ์ที่คาดหวัง 4 รายการ รวมคะแนนเป็น 8 คะแนน

ตาราง 6.1 ข้อมูลนำเข้าและผลลัพธ์ที่คาดหวังจากกรณีทดสอบที่สร้างด้วยระบบของฟังก์ชันงาน Login

Input	Expected result
The ATM card data: true PIN: true	The monitor shows the main menu for performing transactions.
The ATM card data: true PIN: false	The monitor displays an error message and returns the ATM card.
The ATM card data: false PIN: true	The monitor displays an error message and returns the ATM card.
The ATM card data: false PIN: false	The monitor displays an error message and returns the ATM card.

สำหรับกรณีตัวอย่างระบบ ATM อย่างง่าย จะถูกกำหนดคะแนนของแต่ละฟังก์ชันงานได้ดังตาราง 6.2

ตาราง 6.2 การกำหนดคะแนนของกรณีทดสอบที่สร้างโดยระบบสำหรับแต่ละฟังก์ชันงานของระบบ ATM อย่างง่าย

คะแนนของกรณีทดสอบที่สร้างโดยระบบ	Login (คะแนน)	Check balance (คะแนน)	Deposit (คะแนน)	Withdrawal (คะแนน)	Total mark (คะแนน)
	8	4	10	10	32

ผลการประเมินประสิทธิผลที่ได้จากการสร้างกรณีทดสอบกับกลุ่มผู้ทดลองนำร่องทั้ง 4 คน ดังแสดงในตาราง 6.3 ซึ่งจะเห็นว่าความตรงกันของกรณีทดสอบที่ได้โดยเฉลี่ยคิดเป็น 64.06%

ตาราง 6.3 ผลการประเมินประสิทธิผลที่ได้จากการสร้างกรณีทดสอบกับกลุ่มผู้ทดลองนำร่อง

กลุ่มผู้ทดลอง	Login (คะแนน)	Check balance (คะแนน)	Deposit (คะแนน)	Withdrawal (คะแนน)	Total (คะแนน)	ความตรงกัน
คนที่ 1	8	4	4	6	22	68.75%
คนที่ 2	8	4	4	4	20	62.50%
คนที่ 3	8	4	4	4	20	62.50%
คนที่ 4	8	4	4	4	20	62.50%
ค่าเฉลี่ย	**8**	**4**	**4**	**4.50**	**20.50**	**64.06%**

6.1.3 การเลือกกลุ่มผู้ประเมิน

การประเมินประสิทธิผลของกรณีทดสอบนั้น ผู้วิจัยได้คัดเลือกผู้เชี่ยวชาญทางด้านการวิเคราะห์และพัฒนาระบบ จำนวน 10 ท่าน ได้แก่

1) นายพลชัย ลพานุสรณ์

ตำแหน่งนักวิชาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์ ระดับและวุฒิการศึกษาสูงสุด คือ ระดับปริญญาโท วท.ม.(วิทยาการคอมพิวเตอร์)

2) นายชัยวัฒน์ พานิชกุล

ตำแหน่งนักวิชาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์ ระดับและวุฒิการศึกษาสูงสุด คือ ระดับปริญญาตรี วท.บ. (เทคโนโลยีสารสนเทศ)

3) นางสาววรรณรักษ์ นุรักษ์

ตำแหน่งนักวิชาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์ ระดับและวุฒิการศึกษาสูงสุด คือ ระดับปริญญาโท วท.ม. (การจัดการเทคโนโลยีสารสนเทศ)

4) นางอรอุมา รัตนดิลก ณ ภูเก็ต

ตำแหน่งหัวหน้าหน่วยเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ ระดับและวุฒิการศึกษาสูงสุด คือ ระดับปริญญาโท วท.ม. (วิทยาการคอมพิวเตอร์)

5) นายสุขกมล สุขพิศิษฐ์

ตำแหน่งนักวิชาการคอมพิวเตอร์ ศูนย์คอมพิวเตอร์ ระดับและวุฒิการศึกษาสูงสุด คือ ระดับปริญญาตรี วท.บ. (เทคโนโลยีสารสนเทศและการสื่อสาร)

6) นางสาวอัญชลี พัฒนพันธ์ชัย

ตำแหน่งนักวิชาการคอมพิวเตอร์ หน่วยเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ ระดับและวุฒิการศึกษาสูงสุด คือ ระดับปริญญาโท วท.ม. (การจัดการเทคโนโลยีสารสนเทศ)

7) นางวิไลวรรณ หังสพฤกษ์

ตำแหน่งนักวิชาการคอมพิวเตอร์ ศูนย์คอมพิวเตอร์ ระดับและวุฒิการศึกษาสูงสุด คือ ระดับปริญญาโท วท.ม. (การจัดการเทคโนโลยีสารสนเทศ)

8) นางสาวเบญจวรรณ สุขหนู

ตำแหน่งนักวิชาการคอมพิวเตอร์ ศูนย์คอมพิวเตอร์ ระดับและวุฒิการศึกษาสูงสุด คือ ระดับปริญญาโท วท.ม. (การจัดการเทคโนโลยีสารสนเทศ)

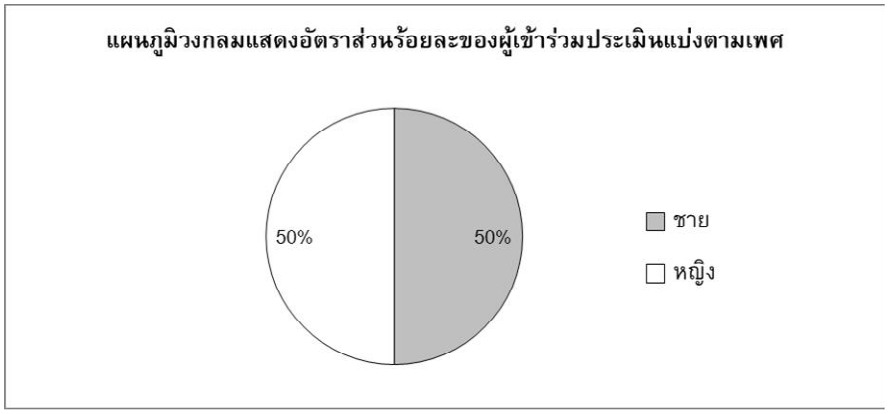
9) นายพงศพัฒน์ หังสพฤกษ์

ตำแหน่งนักวิชาการคอมพิวเตอร์ ศูนย์คอมพิวเตอร์ ระดับและวุฒิการศึกษาสูงสุด คือ ระดับปริญญาโท วท.ม. (การจัดการเทคโนโลยีสารสนเทศ)

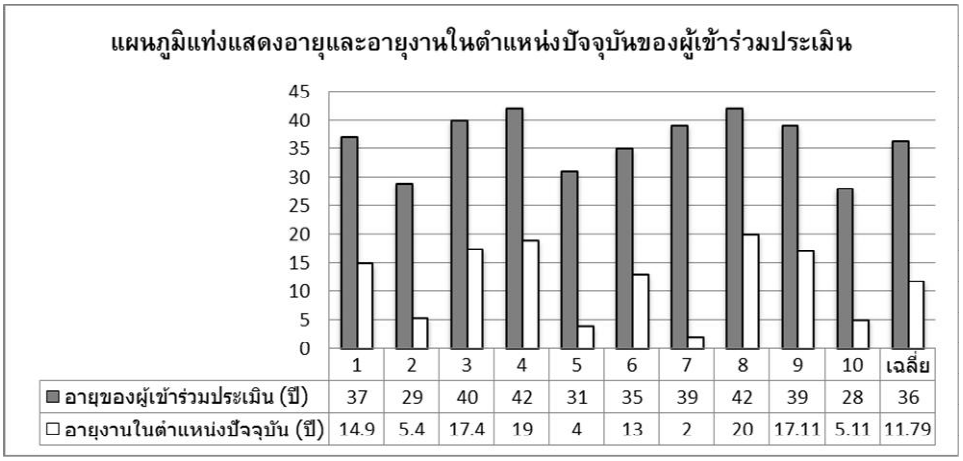
10) นายกัมปนาท โชติกุล

ตำแหน่งนักวิชาการคอมพิวเตอร์ ศูนย์คอมพิวเตอร์ ระดับและวุฒิการศึกษาสูงสุด คือ ระดับปริญญาตรี วท.บ. (วิทยาการคอมพิวเตอร์)

สามารถสรุปข้อมูลส่วนตัวในด้านเพศ อายุและอายุงานในตำแหน่งปัจจุบันของนักวิเคราะห์ที่เข้าร่วมประเมินได้ดังภาพประกอบ 6.2 และภาพประกอบ 6.3 ตามลำดับ



ภาพประกอบ 6.2 อัตราส่วนร้อยละของผู้เข้าร่วมประเมินประสิทธิผลของกรณีทดสอบแบ่งตามเพศ



	1	2	3	4	5	6	7	8	9	10	เฉลี่ย
■ อายุของผู้เข้าร่วมประเมิน (ปี)	37	29	40	42	31	35	39	42	39	28	36
□ อายุงานในตำแหน่งปัจจุบัน (ปี)	14.9	5.4	17.4	19	4	13	2	20	17.11	5.11	11.79

ภาพประกอบ 6.3 อายุและอายุงานของผู้เข้าร่วมประเมินประสิทธิผลของกรณีทดสอบแต่ละคน

6.1.4 ผลการประเมินประสิทธิผลของกรณีทดสอบ

ผลการเปรียบเทียบประสิทธิผลการสร้างกรณีทดสอบของนักวิเคราะห์ระบบด้วยมือกับการสร้างกรณีทดสอบด้วยระบบที่พัฒนาขึ้น สามารถแสดงดังตาราง 6.4 ทั้งนี้ ได้กำหนดหมายเลขรหัส A1 ถึง A10 แทนนักวิเคราะห์ระบบทั้ง 10 ราย โดยไม่เรียงลำดับตามรายชื่อในหัวข้อ 6.1.3 และช่วงเวลาที่เข้าร่วมประเมิน

นอกจากนี้ผู้วิจัยได้จับเวลาที่นักวิเคราะห์ระบบแต่ละรายใช้ในการสร้างกรณีทดสอบด้วยมือสำหรับกรณีระบบงานตัวอย่าง ได้ค่าเฉลี่ย 41 นาที เมื่อเทียบกับที่ผู้วิจัยได้ทดลองใช้งานระบบเพื่อสร้างกรณีทดสอบเองแล้วจับเวลาได้ ประมาณ 20 นาที โดยสามารถแสดงเวลาที่นักวิเคราะห์ระบบแต่ละรายใช้เปรียบเทียบกับการสร้างกรณีทดสอบด้วยระบบที่พัฒนาขึ้นดังตาราง 6.5

6.1.5 สรุปผลการประเมินประสิทธิผลของกรณีทดสอบ

จากผลการประเมินประสิทธิผลของกรณีทดสอบที่สร้างขึ้นด้วยระบบที่พัฒนาขึ้นเปรียบเทียบกับกรณีทดสอบที่สร้างโดยนักวิเคราะห์ระบบ สามารถสรุปได้ดังนี้

1) ด้านความตรงกันของกรณีทดสอบ

ความตรงกันของกรณีทดสอบที่สร้างด้วยนักวิเคราะห์ระบบโดยเฉลี่ยคิดเป็นร้อยละ 62.81 ของกรณีทดสอบที่สร้างด้วยระบบที่พัฒนาขึ้น

2) ด้านเวลาที่ใช้ในการสร้างกรณีทดสอบ

เวลาที่ใช้ในการสร้างกรณีทดสอบด้วยระบบเมื่อเทียบกับการสร้างด้วยมือของนักวิเคราะห์ระบบโดยเฉลี่ยลดลงร้อยละ 51

6.1.6 ข้อสังเกตจากการประเมินประสิทธิผลของกรณีทดสอบ

1) นักวิเคราะห์ผู้เข้าร่วมทดลองสร้างกรณีทดสอบสำหรับโครงงานกรณีตัวอย่างที่ประกอบด้วยฟังก์ชันงานต่างๆ ตามคำอธิบายในข้อกำหนดคุณลักษณะของฟังก์ชันงาน (Functional requirements specification) ได้สร้างกรณีทดสอบนอกเหนือจากข้อกำหนดที่ให้ โดยใช้ประสบการณ์ในการใช้งานจริงกับระบบ ATM ของธนาคารต่างๆ ที่มีการบริการหลากหลายรูปแบบ

2) นักวิเคราะห์ที่เข้าร่วมทดลองส่วนใหญ่จะให้ข้อมูลว่าไม่มีปัญหาในการใช้ภาษาอังกฤษในข้อกำหนดคุณลักษณะของฟังก์ชันงาน แต่ในความจริงนักวิเคราะห์บางรายเข้าใจศัพท์ภาษาอังกฤษเชิงเทคนิคที่ใช้กับการสร้างกรณีทดสอบคลาดเคลื่อน เช่น ผลลัพธ์ที่คาดหวัง (Expected result) จากข้อมูลนำเข้าสู่ฟังก์ชันงาน โดยนักวิเคราะห์บางรายใส่ผลลัพธ์ที่คาดว่าจะให้ระบบทำได้ ซึ่งไม่อยู่ในขอบเขตของในข้อกำหนดคุณลักษณะความต้องการของฟังก์ชันนั้น

ตาราง 6.4 ผลการประเมินประสิทธิผลที่ได้จากการสร้างกรณีทดสอบกับนักวิเคราะห์ระบบ

กลุ่มผู้ทดลอง	Login (คะแนน)	Check balance (คะแนน)	Deposit (คะแนน)	Withdrawal (คะแนน)	Total (คะแนน)	ความตรงกัน
A1	6	4	8	6	24	75.00%
A2	6	4	2	2	14	43.75%
A3	6	3	0	4	13	40.63%
A4	6	4	10	10	30	93.75%
A5	5	4	6	8	23	71.88%
A6	8	4	6	8	26	81.25%
A7	6	1	4	4	15	46.88%
A8	6	4	6	6	22	68.75%
A9	6	4	4	6	20	62.50%
A10	4	4	2	4	14	43.75%
ค่าเฉลี่ย	**6**	**4**	**5**	**6**	**20**	**62.81%**

ตาราง 6.5 เวลาที่นักวิเคราะห์ระบบใช้ในการสร้างกรณีทดสอบด้วยมือเปรียบเทียบกับการสร้างกรณีทดสอบด้วยระบบที่พัฒนาขึ้น

นักวิเคราะห์ระบบ	เวลาที่ใช้ในการสร้างกรณีทดสอบด้วยมือ	เวลาที่ลดลงเมื่อใช้ระบบในการสร้างกรณีทดสอบ
A1	50 นาที	60%
A2	65 นาที	69%
A3	50 นาที	60%
A4	41 นาที	51%
A5	46 นาที	57%
A6	25 นาที	20%
A7	25 นาที	20%
A8	38 นาที	47%
A9	25 นาที	20%
A10	40 นาที	50%
ค่าเฉลี่ย	41 นาที	51%

6.2 การประเมินประสิทธิภาพของระบบ

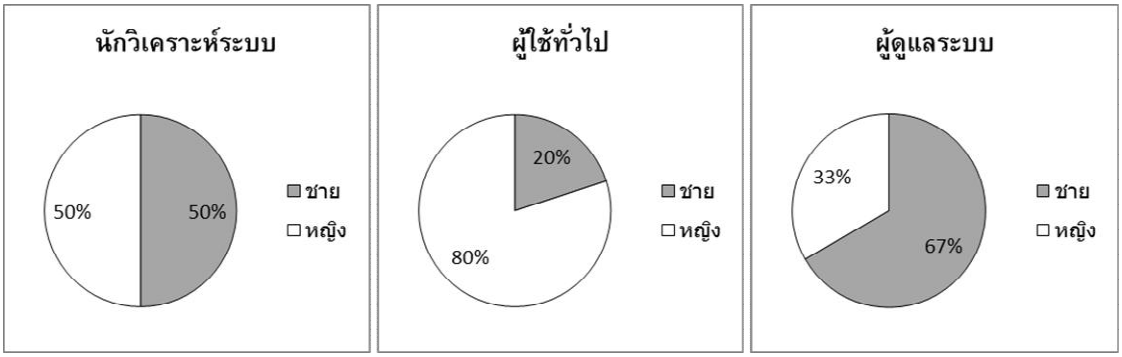
การประเมินประสิทธิภาพของระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม มีขั้นตอนการดำเนินงานและผลจากการดำเนินงานดังนี้

6.2.1 การประเมินระบบแบบอิสระที่ไม่มีการควบคุม

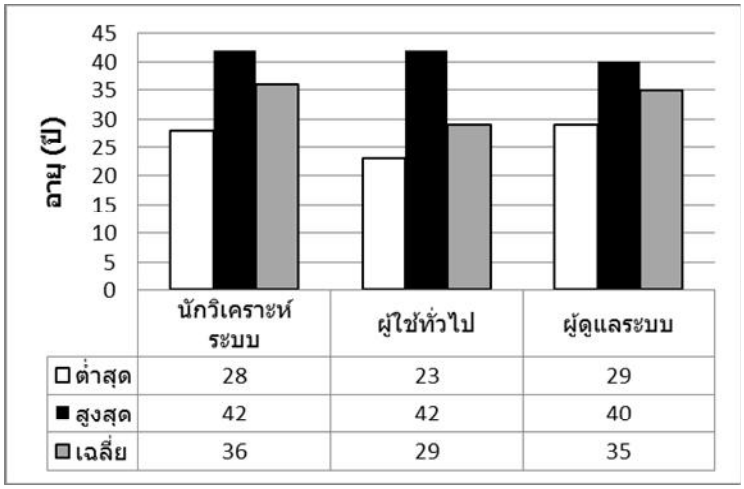
ก่อนมีการประเมินระบบงานจริงที่มีการควบคุมสิ่งแวดล้อม ได้มีการสอบถามความพึงพอใจต่อการใช้งานระบบเบื้องต้นในสถานการณ์อิสระที่ไม่มีการควบคุม จากกลุ่มผู้ใช้ทั่วไปจำนวน 20 คน แสดงผลการประเมินดังรายงานใน (สิริมา, 2555) ทั้งนี้ได้มีการปรับปรุงขั้นตอนวิธีให้เหมาะสมขึ้นกว่าเดิม ตามข้อเสนอแนะที่ได้จากการประเมินครั้งนี้ และจากคณะกรรมการสอบความก้าวหน้าของวิทยานิพนธ์ ก่อนมีการประเมินระบบในสิ่งแวดล้อมที่ควบคุม

6.2.2 การเลือกกลุ่มประเมินสำหรับการทดลองในสิ่งแวดล้อมที่ควบคุม

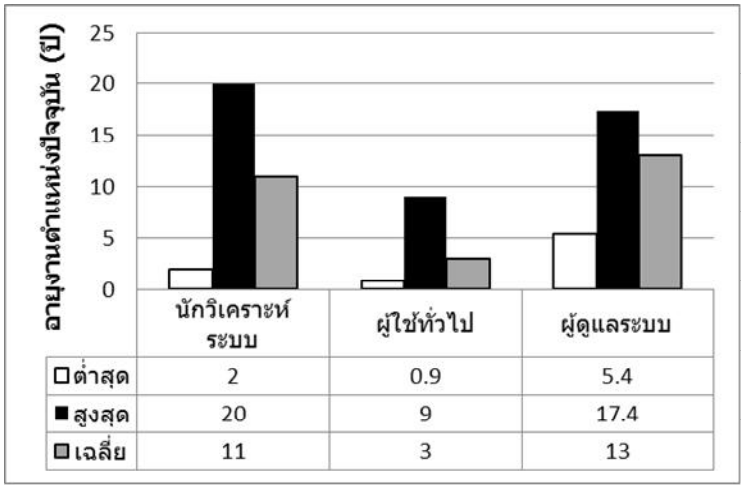
เนื่องจากข้อจำกัดของจำนวนผู้ประเมิน ระยะเวลาและค่าใช้จ่ายที่จะเกิดขึ้นจากการประเมินระบบ ดังนั้นจึงใช้การเลือกกลุ่มผู้ประเมินแบบเจาะจง โดยเลือกผู้เชี่ยวชาญทางด้านการวิเคราะห์และพัฒนาระบบงานคอมพิวเตอร์จากภาควิชาวิทยาการคอมพิวเตอร์ หน่วยเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ และศูนย์คอมพิวเตอร์ จำนวนทั้งสิ้น 10 คน เป็นตัวแทนของกลุ่มผู้ใช้ที่เป็นนักวิเคราะห์ระบบ เลือกบุคลากรและนักศึกษาปริญญาโทในมหาวิทยาลัยสงขลานครินทร์ รวมทั้งสิ้นจำนวน 20 คน เป็นตัวแทนของกลุ่มผู้ใช้ทั่วไป และเลือกผู้ดูแลระบบจากภาควิชาวิทยาการคอมพิวเตอร์ จำนวน 3 คน เป็นตัวแทนของกลุ่มผู้ใช้ที่เป็นผู้ดูแลระบบ ทั้งสามกลุ่มนี้จะเป็นผู้ทดสอบและประเมินประสิทธิภาพของระบบ แสดงข้อมูลส่วนตัวของผู้เข้าร่วมประเมินในแต่ละกลุ่มดังภาพประกอบ 6.4 ถึงภาพประกอบ 6.8



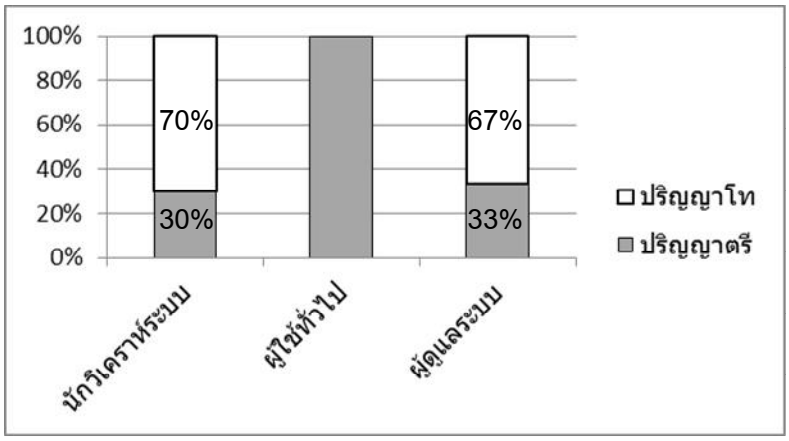
ภาพประกอบ 6.4 อัตราส่วนร้อยละของผู้เข้าร่วมประเมินประสิทธิภาพของระบบแบ่งตามเพศ



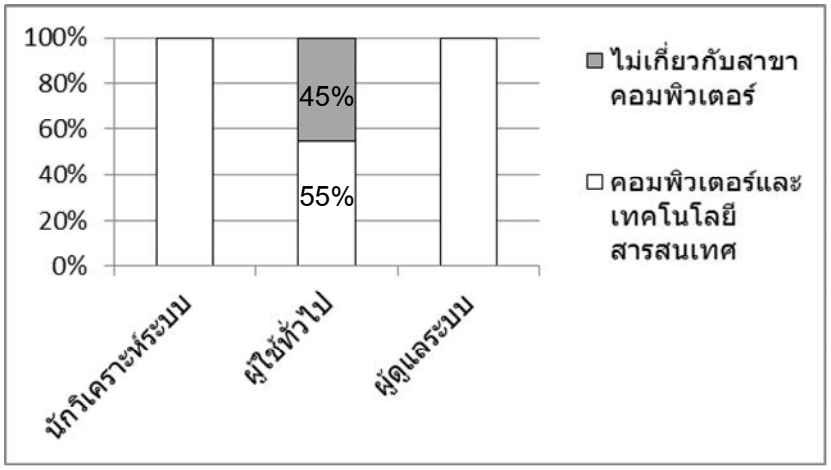
ภาพประกอบ 6.5 อายุของผู้เข้าร่วมประเมินประสิทธิภาพของระบบแต่ละกลุ่ม



ภาพประกอบ 6.6 อายุงานของผู้เข้าร่วมประเมินประสิทธิภาพของระบบแต่ละกลุ่ม



ภาพประกอบ 6.7 ระดับการศึกษาสูงสุดของผู้เข้าร่วมประเมินประสิทธิภาพของระบบแต่ละกลุ่ม



ภาพประกอบ 6.8 อัตราส่วนของผู้เข้าร่วมประเมินประสิทธิภาพของระบบแบ่งตามสาขาวิชาที่จบการศึกษาซึ่งเกี่ยวข้องและไม่เกี่ยวข้องกับด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

6.2.3 เครื่องมือที่ใช้ในการประเมิน

1) ระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม

2) โจทย์สำหรับการประเมินระบบ

ในการประเมินระบบภายใต้สิ่งแวดล้อมที่ควบคุม ได้ให้ผู้เข้าร่วมทดลองใช้ระบบตามโจทย์ปัญหาที่กำหนดให้ซึ่งสอดคล้องกับประเภทกลุ่มผู้ใช้ เพื่อให้ผู้ใช้ได้เข้าถึงทุกฟังก์ชันงานที่เกี่ยวข้องกับกลุ่มผู้ใช้แต่ละประเภท

3) แบบสอบถาม

การสร้างแบบสอบถามการประเมินประสิทธิภาพของระบบ ที่จะนำมาใช้กับระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม มีขั้นตอนการสร้างแบบสอบถาม 4 ขั้นตอน ดังนี้

3.1) รวบรวมข้อมูลและศึกษาเอกสารที่เกี่ยวข้อง เพื่อกำหนดแนวทางการสร้างแบบสอบถามความพึงพอใจ

3.2) กำหนดประเด็นและออกแบบแบบสอบถามการประเมินประสิทธิภาพของระบบ

3.3) สร้างแบบสอบถามการประเมินประสิทธิภาพของระบบเพื่อใช้กับผู้เข้าร่วมประเมิน

3.4) ใช้แบบสอบถามการประเมินประสิทธิภาพของระบบกับกลุ่มผู้เข้าร่วมการประเมิน จำนวน 33 คน โดยแบบสอบถามแบ่งออกเป็น 3 ตอน ดังนี้

ตอนที่ 1 ส่วนข้อมูลเกี่ยวกับผู้ตอบแบบสอบถาม

ตอนที่ 2 เป็นแบบสอบถามความพึงพอใจในการใช้งานระบบ มีลักษณะเป็นแบบสอบถามปลายปิดจำนวน 12 ข้อ

ตอนที่ 3 ส่วนคำถามปลายเปิดสำหรับข้อเสนอแนะจำนวน 1 ข้อ

6.2.4 วิธีวิเคราะห์ข้อมูลของการประเมินความพึงพอใจของผู้ใช้ระบบ

ผู้วิจัยรวบรวมแบบสอบถามและทำการวิเคราะห์ผลความพึงพอใจของกลุ่มผู้เข้าร่วมประเมิน ซึ่งผู้วิจัยกำหนดเกณฑ์การให้คะแนนตามแนวทาง Likert Scale ดังนี้

4.50-5.00 ความพึงพอใจมากที่สุด ระบบมีประสิทธิภาพในเกณฑ์ดีมาก
3.50-4.49 ความพึงพอใจมาก ระบบมีประสิทธิภาพในเกณฑ์ดี
2.50-3.49 ความพึงพอใจปานกลาง ระบบมีประสิทธิภาพในเกณฑ์ปานกลาง
1.50-2.49 ความพึงพอใจน้อย ระบบมีประสิทธิภาพในเกณฑ์พอใช้
0.50-1.49 ความพึงพอใจน้อยที่สุด ระบบมีประสิทธิภาพในเกณฑ์ควรปรับปรุง

สถิติที่ใช้ในการวิเคราะห์ความพึงพอใจของผู้ประเมิน ได้แก่

1) ค่าเฉลี่ย ($\bar{X}$)

$$\bar{X} = \frac{\sum_{i=1}^{n} x_i}{n}$$

โดย $\bar{X}$ คือ ค่าเฉลี่ย
x_i คือ ข้อมูลแต่ละจำนวน
$\sum_{i=1}^{n} x_i$ คือ ผลรวมคะแนนทั้งหมด
n คือ จำนวนตัวอย่าง

2) ค่าส่วนเบี่ยงเบนมาตรฐาน (Standard deviation)

$$SD = \sqrt{\frac{n\sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} (x_i)^2}{n(n-1)}}$$

โดย SD คือ ส่วนเบี่ยงเบนมาตรฐาน
x_i คือ ข้อมูลแต่ละจำนวน
$\sum_{i=1}^{n} x_i$ คือ ผลรวมคะแนนทั้งหมด
n คือ จำนวนตัวอย่าง

6.2.5 ผลการประเมินประสิทธิภาพของระบบ

1) ความพึงพอใจของผู้ดูแลระบบ

ผลจากการประเมินความพึงพอใจจากการใช้งานระบบของผู้ดูแลระบบ แสดงในตาราง 6.6 ถึงตาราง 6.8

ตาราง 6.6 ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านการทำงานในฐานะผู้ดูแลระบบ

1. ความพึงพอใจด้านการทำงานของระบบ	$\overline{X}$	**S.D.**	**ระดับความพึงพอใจ**
1.1 ความถูกต้องของการทำงาน	4.33	1.15	มาก
1.2 ความสมบูรณ์ของผลลัพธ์	4.33	1.15	มาก
1.3 ความเร็วในการประมวลผล	4.67	0.58	มากที่สุด
1.4 ความสามารถในการป้องกันความผิดพลาดจากผู้ใช้	4.00	1.00	มาก

จากตาราง 6.6 พบว่า ความพึงพอใจของผู้ดูแลระบบ ด้านการทำงานของระบบอยู่ในระดับมากที่สุด คือ ความเร็วในการประมวลผล นอกจากนั้นอยู่ในระดับมาก

ตาราง 6.7 ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านการติดต่อกับผู้ใช้งานในฐานะผู้ดูแลระบบ

2. ความพึงพอใจด้านการติดต่อกับผู้ใช้งาน	$\overline{X}$	**S.D.**	**ระดับความพึงพอใจ**
2.1 การใช้งานง่ายและเป็นมิตรกับผู้ใช้ เช่น การจัดลำดับข้อมูลบนหน้าจอ	4.33	1.15	มาก
2.2 ความเหมาะสมในการวางตำแหน่งองค์ประกอบบนหน้าจอ	4.67	0.58	มากที่สุด
2.3 ความชัดเจนในการสื่อความหมายของส่วนต่างๆ	4.33	0.58	มาก
2.4 ในกรณีที่มีข้อผิดพลาดมีการแสดงให้เห็นอย่างชัดเจน	4.33	1.15	มาก
2.5 มีส่วนช่วยเหลือผู้ใช้ในการใช้งานระบบ	4.00	1.00	มาก
2.6 ความสวยงามของการแสดงผลบนหน้าจอ เช่น การใช้สีหรือขนาดของตัวอักษร	4.33	0.58	มาก

จากตาราง 6.7 พบว่า ความพึงพอใจของผู้ดูแลระบบ ด้านการติดต่อกับผู้ใช้งานอยู่ในระดับมากที่สุด คือ ความเหมาะสมในการวางตำแหน่งองค์ ประกอบบนหน้าจอ นอกจากนั้นอยู่ในระดับมาก

ตาราง 6.8 ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านคุณค่าของระบบในฐานะผู้ดูแลระบบ

3. ความพึงพอใจด้านคุณค่าของระบบ	$\overline{X}$	**S.D.**	**ระดับความพึงพอใจ**
3.1 ช่วยลดเวลาในการสร้างกรณีทดสอบ ซึ่งจะเป็นร่างกรณีทดสอบเบื้องต้น	3.67	1.15	มาก
3.2 สามารถนำไปเป็นระบบสนับสนุนสำหรับการตรวจรับซอฟต์แวร์ โดยเฉพาะผู้ใช้ที่ขาดความรู้ในการสร้างกรณีทดสอบ	3.67	1.15	มาก

จากตาราง 6.8 พบว่า ความพึงพอใจของผู้ดูแลระบบ ด้านคุณค่าของระบบอยู่ในระดับมาก ทั้งนี้ผู้ดูและระบบทั้งหมดให้เหตุผลว่าไม่ได้มีหน้าที่ที่เกี่ยวข้องกับประโยชน์โดยตรงของระบบ จึงให้คะแนนความพึงพอใจในมุมมองของผู้ดูและระบบเท่านั้น

2) ความพึงพอใจของนักวิเคราะห์ระบบ

ผลจากการประเมินความพึงพอใจจากการใช้งานระบบของนักวิเคราะห์ระบบแสดงในตาราง 6.9 - 6.11

ตาราง 6.9 ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านการทำงานในฐานะนักวิเคราะห์ระบบ

1. ความพึงพอใจด้านการทำงานของระบบ	$\overline{X}$	**S.D.**	**ระดับความพึงพอใจ**
1.1 ความถูกต้องของการทำงาน	3.90	0.74	มาก
1.2 ความสมบูรณ์ของผลลัพธ์	3.80	0.79	มาก
1.3 ความเร็วในการประมวลผล	4.60	0.52	มากที่สุด
1.4 ความสามารถในการป้องกันความผิดพลาดจากผู้ใช้	3.20	0.92	ปานกลาง

จากตาราง 6.9 พบว่า ความพึงพอใจของนักวิเคราะห์ระบบ ด้านการทำงานของระบบอยู่ในระดับมากที่สุด คือ ความเร็วในการประมวลผล ส่วนประเด็นความถูกต้องของการทำงานและความสมบูรณ์ของผลลัพธ์อยู่ในระดับมาก และความสามารถในการป้องกันความผิดพลาดจากผู้ใช้อยู่ในระดับปานกลาง

ตาราง 6.10 ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านการติดต่อกับผู้ใช้งานในฐานะนักวิเคราะห์ระบบ

2. ความพึงพอใจด้านการติดต่อกับผู้ใช้งาน	$\overline{X}$	**S.D.**	**ระดับความพึงพอใจ**
2.1 การใช้งานง่ายและเป็นมิตรกับผู้ใช้ เช่น การจัดลำดับข้อมูลบนหน้าจอ	3.90	0.74	มาก
2.2 ความเหมาะสมในการวางตำแหน่งองค์ประกอบบนหน้าจอ	4.10	0.74	มาก
2.3 ความชัดเจนในการสื่อความหมายของส่วนต่างๆ	3.90	0.74	มาก
2.4 ในกรณีที่มีข้อผิดพลาดมีการแสดงให้เห็นอย่างชัดเจน	4.00	0.94	มาก
2.5 มีส่วนช่วยเหลือผู้ใช้ในการใช้งานระบบ	4.10	1.10	มาก
2.6 ความสวยงามของการแสดงผลบนหน้าจอ เช่น การใช้สีหรือขนาดของตัวอักษร	3.80	0.79	มาก

จากตาราง 6.10 พบว่า ความพึงพอใจของนักวิเคราะห์ระบบ ด้านการติดต่อกับผู้ใช้งานอยู่ในระดับมาก

ตาราง 6.11 ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านคุณค่าของระบบในฐานะนักวิเคราะห์ระบบ

3. ความพึงพอใจด้านคุณค่าของระบบ	$\overline{X}$	**S.D.**	**ระดับความพึงพอใจ**
3.1 ช่วยลดเวลาในการสร้างกรณีทดสอบ ซึ่งจะเป็นร่างกรณีทดสอบเบื้องต้น	4.00	0.67	มาก
3.2 สามารถนำไปเป็นระบบสนับสนุนสำหรับการตรวจรับซอฟต์แวร์ โดยเฉพาะผู้ใช้ที่ขาดความรู้ในการสร้างกรณีทดสอบ	3.90	0.99	มาก

จากตาราง 6.11 พบว่า ความพึงพอใจของนักวิเคราะห์ระบบ ด้านคุณค่าของระบบอยู่ในระดับมาก

3) ความพึงพอใจของผู้ใช้ทั่วไป

ผลจากการประเมินความพึงพอใจจากการใช้งานระบบของผู้ใช้ทั่วไป แสดงในตาราง 6.12 - 6.14

ตาราง 6.12 ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านการทำงานในฐานะผู้ใช้ทั่วไป

1. ความพึงพอใจด้านการทำงานของระบบ	$\overline{X}$	**S.D.**	**ระดับความพึงพอใจ**
1.1 ความถูกต้องของการทำงาน	4.55	0.51	มากที่สุด
1.2 ความสมบูรณ์ของผลลัพธ์	4.40	0.68	มาก
1.3 ความเร็วในการประมวลผล	4.55	0.60	มากที่สุด
1.4 ความสามารถในการป้องกันความผิดพลาดจากผู้ใช้	3.80	0.70	มาก

จากตาราง 6.12 พบว่า ความพึงพอใจของผู้ใช้ทั่วไป ด้านการทำงานของระบบอยู่ในระดับมากที่สุด คือ ความถูกต้องของการทำงานและความเร็วในการประมวลผล ส่วนความสมบูรณ์ของผลลัพธ์และความสามารถในการป้องกันความผิดพลาดจากผู้ใช้อยู่ในระดับมาก

ตาราง 6.13 ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านการติดต่อกับผู้ใช้งานในฐานะผู้ใช้ทั่วไป

2. ความพึงพอใจด้านการติดต่อกับผู้ใช้งาน	$\overline{X}$	**S.D.**	**ระดับความพึงพอใจ**
2.1 การใช้งานง่ายและเป็นมิตรกับผู้ใช้ เช่น การจัดลำดับข้อมูลบนหน้าจอ	4.30	0.66	มาก
2.2 ความเหมาะสมในการวางตำแหน่งองค์ประกอบบนหน้าจอ	4.40	0.75	มาก
2.3 ความชัดเจนในการสื่อความหมายของส่วนต่างๆ	4.20	0.77	มาก
2.4 ในกรณีที่มีข้อผิดพลาดมีการแสดงให้เห็นอย่างชัดเจน	4.30	0.73	มาก
2.5 มีส่วนช่วยเหลือผู้ใช้ในการใช้งานระบบ	4.55	0.51	มากที่สุด
2.6 ความสวยงามของการแสดงผลบนหน้าจอ เช่น การใช้สีหรือขนาดของตัวอักษร	4.25	0.72	มาก

จากตาราง 6.13 พบว่า ความพึงพอใจของผู้ใช้ทั่วไป ด้านการติดต่อกับผู้ใช้งานอยู่ในระดับมากที่สุด คือ การมีส่วนช่วยเหลือผู้ใช้ในการใช้งานระบบ นอกจากนั้นอยู่ในระดับมาก

ตาราง 6.14 ค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและระดับความพึงพอใจในด้านคุณค่าของระบบในฐานะผู้ใช้ทั่วไป

3. ความพึงพอใจด้านคุณค่าของระบบ	$\overline{X}$	**S.D.**	**ระดับความพึงพอใจ**
3.1 ช่วยลดเวลาในการสร้างกรณีทดสอบ ซึ่งจะเป็นร่างกรณีทดสอบเบื้องต้น	4.40	0.60	มาก
3.2 สามารถนำไปเป็นระบบสนับสนุนสำหรับการตรวจรับซอฟต์แวร์ โดยเฉพาะผู้ใช้ที่ขาดความรู้ในการสร้างกรณีทดสอบ	4.55	0.60	มากที่สุด

จากตาราง 6.14 พบว่า ความพึงพอใจของผู้ใช้ทั่วไป ด้านคุณค่าของระบบอยู่ในระดับมากที่สุด คือ สามารถนำไปเป็นระบบสนับสนุนสำหรับการตรวจรับซอฟต์แวร์โดยเฉพาะผู้ใช้ที่ขาดความรู้ในการสร้างกรณีทดสอบ ส่วนด้านช่วยลดเวลาในการสร้างกรณีทดสอบ ซึ่งจะเป็นร่างกรณีทดสอบเบื้องต้น อยู่ในระดับมาก

6.2.6 สรุปผลการประเมินประสิทธิภาพของระบบ

จากผลการประเมินความพึงพอใจในการใช้งานระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์มจากผู้ใช้ในด้านต่างๆ สามารถสรุปได้ดังนี้

1) ความพึงพอใจด้านการทำงานของระบบ

ความพึงพอใจของผู้ดูแลระบบ นักวิเคราะห์ระบบ และผู้ใช้ทั่วไป ด้านการใช้งานในส่วนของความเร็วในการประมวลผลอยู่ในระดับมากที่สุด ส่วนด้านอื่นๆ มีความพึงพอใจอยู่ในระดับมาก

2) ความพึงพอใจด้านการติดต่อกับผู้ใช้งาน

ความพึงพอใจของผู้ดูแลระบบ นักวิเคราะห์ระบบ และผู้ใช้ทั่วไป ด้านการติดต่อกับผู้ใช้งานอยู่ในระดับมาก เช่น การใช้งานง่ายและเป็นมิตรกับผู้ใช้ มีส่วนช่วยเหลือผู้ใช้ในการใช้งานระบบ ความสวยงามของการแสดงผลบนหน้าจอ

3) ความพึงพอใจด้านคุณค่าของระบบ

ความพึงพอใจของผู้ดูแลระบบ นักวิเคราะห์ระบบ และผู้ใช้ทั่วไป ด้านคุณค่าของระบบอยู่ในระดับมาก ทั้งการช่วยลดเวลาในการสร้างกรณีทดสอบ และสามารถนำไปเป็นระบบสนับสนุนสำหรับการตรวจรับซอฟต์แวร์ โดยเฉพาะผู้ใช้ที่ขาดความรู้ในการสร้างกรณีทดสอบ

6.2.7 ข้อเสนอแนะ

ข้อเสนอแนะที่ได้จากการประเมินความพึงพอใจของผู้ใช้ระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม มีดังนี้

1) การทำงานในส่วนของ Requirements definition เมื่อบันทึกฟังก์ชันงาน ‘Save function’ แล้ว ควรให้เพิ่มฟังก์ชันใหม่อัตโนมัติโดยไม่ต้องกดปุ่ม ‘Create next function’ อีก แม้จะกดปุ่ม ‘Project complete’ แล้ว ควรอนุญาตให้ผู้ใช้เพิ่มฟังก์ชันงานได้อีก ในกรณีที่โครงงานนั้นมีฟังก์ชันงานเพิ่มเติม และควรแสดงข้อมูลฟังก์ชันงานที่ได้บันทึกไปแล้ว เพื่อป้องกันการป้อนข้อมูลซ้ำซ้อน

2) การทำงานในส่วนของการระบุ Requirements specification, Possible value และ Expected result ในตารางตัดสินใจ เมื่อกดปุ่มบันทึกข้อมูลของฟังก์ชันงานนั้นแล้ว ควรอนุญาตให้แก้ข้อมูลได้ในภายหลัง

6.3 การดำเนินการหลังการประเมินกรณีทดสอบและการประเมินระบบ

ผู้วิจัยได้ปรับปรุงระบบตามข้อเสนอแนะของผู้ประเมินระบบ อีกทั้งได้ทบทวนปรับปรุงข้อกำหนดคุณลักษณะความต้องการของระบบกรณีตัวอย่างให้เหมาะสมยิ่งขึ้น แล้วทดลองสร้างกรณีทดสอบอีกครั้ง ซึ่งพบว่าได้กรณีทดสอบที่ดีขึ้นกว่าเดิม

บทที่ 7

บทสรุปและข้อเสนอแนะ

จากเนื้อหาในบทที่ผ่านมา ได้กล่าวถึงความเป็นมาของงานวิจัยและศึกษาเอกสารรวมทั้งแนวคิดที่เกี่ยวข้อง และหลักการที่ได้นำเสนอ จนถึงการพัฒนาเครื่องมือต้นแบบรวมทั้งการทดสอบการใช้งานระบบ สำหรับในบทนี้จะกล่าวถึงผลสรุปในการทำวิจัย ปัญหาและอุปสรรคในการวิจัย รวมถึงข้อเสนอแนะสำหรับผู้ที่สนใจในอนาคต

7.1 บทสรุป

ในกระบวนการทดสอบซอฟต์แวร์ กรณีทดสอบถือเป็นส่วนสำคัญที่จะนำมาซึ่งซอฟต์แวร์ที่มีคุณภาพ สามารถวางแผนดำเนินงานได้ตั้งแต่ขั้นตอนวิศวกรรมความต้องการ (Requirements engineering) โดยเฉพาะการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ (Acceptance testing)

งานวิจัยนี้ได้นำเสนอระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม (A Support System for Elementary Test Case Generation Based on Form-Based Requirements Specification) ซึ่งเป็นเครื่องมือที่พัฒนาขึ้นเพื่อสนับสนุนการสร้างกรณีทดสอบจากคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม โดยเกี่ยวข้องกับกลุ่มผู้ใช้งาน 3 กลุ่ม คือ (1) นักวิเคราะห์ระบบ (2) ผู้ใช้ทั่วไป (3) ผู้ดูแลระบบ

ระบบต้นแบบที่พัฒนาขึ้นใช้งานมีองค์ประกอบ 5 ส่วนหลัก คือ (1) คู่มือการใช้งานระบบ (2) การสร้างโครงงานใหม่ (3) การนิยามความต้องการของฟังก์ชันงาน (4) การระบุข้อมูลที่จำเป็นต่อการสร้างกรณีทดสอบ ซึ่งได้แก่ การกำหนดคุณลักษณะความต้องการในรูปแบบฟอร์ม การระบุชนิดและค่าข้อมูลที่เป็นไปได้ของข้อมูลนำเข้า และการสร้างตารางตัดสินใจ (5) การแสดงผลกรณีทดสอบ

กรณีทดสอบจากวิธีการที่นำเสนอจะถูกประเมินประสิทธิผลโดยใช้นักวิเคราะห์ระบบที่มีความรู้ความสามารถในการวิเคราะห์และพัฒนาระบบจำนวน 10 ราย นอกจากนี้ได้มีการประเมินระบบโดยผู้ใช้แต่ละกลุ่ม ได้แก่ (1) นักวิเคราะห์ระบบจำนวน 10 ราย (2) ผู้ใช้ทั่วไปจำนวน 20 ราย และ (3) ผู้ดูแลระบบจำนวน 3 ราย โดยผลการประเมินประสิทธิผลของกรณีทดสอบที่สร้างขึ้นด้วยระบบที่พัฒนาขึ้นเปรียบเทียบกับกรณีทดสอบที่สร้างโดยนักวิเคราะห์

ระบบพบว่า ความตรงกันของกรณีทดสอบที่สร้างด้วยนักวิเคราะห์ระบบโดยเฉลี่ยคิดเป็นร้อยละ 62.81 ของกรณีทดสอบที่สร้างด้วยระบบที่พัฒนาขึ้น และเวลาที่ใช้ในการสร้างกรณีทดสอบด้วยระบบลดลงโดยเฉลี่ยคิดเป็นร้อยละ 51 ของเวลาที่ใช้ในการสร้างกรณีทดสอบด้วยมือโดยนักวิเคราะห์ระบบ ส่วนผลการประเมินประสิทธิภาพของระบบโดยผู้ใช้ทั้ง 3 กลุ่มพบว่า ผู้ใช้งานระบบทั้ง 3 กลุ่ม มีความพึงพอใจต่อการใช้งานระบบโดยรวมอยู่ในระดับมาก

ผลลัพธ์ที่ได้จากวิธีการที่ได้นำเสนอนี้ช่วยให้ (1) การสร้างกรณีทดสอบเริ่มต้นได้ตั้งแต่ขั้นตอนของการระบุความต้องการโดยใช้คุณลักษณะความต้องการ จึงช่วยลดความเสี่ยงในการพัฒนาระบบในขั้นตอนต่อไป เนื่องจากแผนการทดสอบผลิตภัณฑ์ซอฟต์แวร์ เริ่มดำเนินการควบคู่กับการระบุคุณลักษณะความต้องการ (2) การใช้เครื่องมือเพื่อช่วยสร้างกรณีทดสอบ มีส่วนสำคัญที่จะช่วยลดระยะเวลาในการสร้างกรณีทดสอบ และเวลาโดยรวมที่ใช้ในการทดสอบซอฟต์แวร์ให้น้อยลงได้ โดยเฉพาะนำไปเป็นระบบสนับสนุนสำหรับการสร้างกรณีทดสอบระดับการตรวจรับผลิตภัณฑ์ของผู้ใช้หรือลูกค้า ที่มีข้อจำกัดด้านความรู้ในการสร้างกรณีทดสอบซอฟต์แวร์

7.2 ปัญหาและอุปสรรค

ปัญหาและอุปสรรคที่พบในงานวิจัยมีดังนี้

1) นักวิเคราะห์ระบบงานคอมพิวเตอร์ส่วนใหญ่ยังขาดความเข้าใจในด้านวิศวกรรมความต้องการ โดยเฉพาะการสร้างข้อกำหนดคุณลักษณะความต้องการของซอฟต์แวร์มักไม่ค่อยมีการดำเนินการอย่างจริงจังในแนวทางพัฒนาซอฟต์แวร์ปัจจุบัน ทำให้ผู้วิจัยต้องพยายามสร้างกรอบแนวคิดที่สามารถนำไปใช้ในทางปฏิบัติได้จริง

2) การหาวิธีที่จะนำมาซึ่งการประเมินประสิทธิผลของกรณีทดสอบนั้นเป็นเรื่องที่ค่อนข้างยาก เนื่องจากการประเมินขั้นตอนวิธีการประมวลผลด้วยคอมพิวเตอร์ใดๆ มักต้องมีกรณีที่เคยมีผู้เสนอแนวคิดไว้แล้วมาเทียบเคียง สำหรับงานวิจัยนี้ที่เกี่ยวข้องกับการสร้างกรณีทดสอบในช่วงวิศวกรรมความต้องการ มีความยุ่งยากในการหาแนวคิดที่เคยเสนอมาเทียบเคียง

7.3 ข้อเสนอแนะและงานในอนาคต

จากการวิจัยในครั้งนี้มีข้อเสนอแนะ ดังนี้

1) ปรับการพัฒนาขั้นตอนวิธีที่มีการสร้างกรณีทดสอบได้อย่างอัตโนมัติมากยิ่งขึ้น ซึ่งต้องเชื่อมโยงกับพจนานุกรมข้อมูล (Data dictionary) ที่ได้จากการวิเคราะห์ระบบในกระบวนการวิศวกรรมความต้องการได้

2) การประเมินกรณีทดสอบควรดำเนินการโดยอาศัยผู้เชี่ยวชาญ ที่นอกเหนือจากนักวิเคราะห์ระบบที่เข้าร่วมทดลองในการสร้างกรณีทดสอบ และประเมินระบบในฐานะนักวิเคราะห์ระบบ โดยผู้เชี่ยวชาญจะทำหน้าที่ในการประเมินกรณีทดสอบที่สร้างขึ้นโดยระบบ เพื่อยืนยันการยอมรับได้ของกรณีทดสอบที่สร้างขึ้นสำหรับระบบกรณีตัวอย่าง และให้ข้อเสนอแนะที่จำเป็น หรืออาจใช้กรณีทดสอบที่สร้างขึ้นสำหรับโครงงานอื่นก็ได้ นอกจากนี้ถ้าเป็นไปได้ ควรมีการทดลองประเมินซ้ำการสร้างกรณีทดสอบด้วยนักวิเคราะห์ระบบกลุ่มใหม่เทียบกับกรณีทดสอบที่ผู้เชี่ยวชาญกลุ่มนี้ได้ให้ข้อเสนอแนะในลักษณะการทดลองซ้ำ (Replication) ในทางวิทยาศาสตร์

3) ควรปรับระบบให้เป็นโปรแกรมที่สามารถทำงานบนเว็บ (Web - based application) ได้ เพื่อให้ง่ายต่อการใช้งานสำหรับผู้ใช้ทุกกลุ่ม

4) ระบบที่พัฒนาขึ้นนี้จะแสดงผลกรณีทดสอบที่สร้างโดยระบบทั้งหมด ซึ่งบางกรณีอาจซ้ำซ้อน จึงควรให้นักวิเคราะห์สามารถเลือกแสดงผลเฉพาะกรณีทดสอบที่ต้องการได้

บรรณานุกรม

กิตติ ภักดีวัฒนะกุล และพนิดา พานิชกุล. 2550. วิศวกรรมซอฟต์แวร์. เคทีพี คอมพ์ แอนด์ คอนซัลท์: กรุงเทพฯ.

ณัฏฐพันธ์ เขจรนันท์. 2551. การวิเคราะห์และออกแบบระบบสารสนเทศ. สำนักพิมพ์ซีเอ็ดยูเคชั่น จำกัด: กรุงเทพฯ.

บัญชา ปะสีละเตสัง. 2552. พัฒนาแอปพลิเคชั่นด้วย Visual C# 2008. สำนักพิมพ์ซีเอ็ดยูเคชั่น จำกัด: กรุงเทพฯ.

เศรษฐพงศ์ ลีหรัตนรักษ์. 2547. วิธีการสร้างกรณีทดสอบโดยอัตโนมัติจากยูสเคส. วิทยานิพนธ์มหาบัณฑิต (วิทยาศาสตร์คอมพิวเตอร์), วิทยาศาสตร์จุฬาลงกรณ์ มหาวิทยาลัย.

สกาวรัตน์ จงพัฒนากร. 2550. การวิเคราะห์และออกแบบระบบสารสนเทศ. สำนักพิมพ์มหาวิทยาลัยเกษตรศาสตร์: กรุงเทพฯ.

สิริมา นาคเตี้ย และ อำนาจ เปาะทอง. 2555. กรอบแนวคิดการสร้างกรณีทดสอบจากคุณลักษณะความต้องการในรูปแบบฟอร์ม. The 5th National Conference and 2012 International Conference on Applied Computer Technology and Information Systems (ACTIS 2012). สงขลา ประเทศไทย, 22 – 23 กันยายน 2555.

สุชาดา สุภผล. 2547. การสร้างกรณีทดสอบสำหรับโปรแกรมประยุกต์บนเว็บด้วยเทคนิคการทดสอบแบบแบล็กบอกซ์. วิทยานิพนธ์มหาบัณฑิต (วิทยาศาสตร์คอมพิวเตอร์), วิทยาศาสตร์จุฬาลงกรณ์ มหาวิทยาลัย.

สุชาย ธนวเสถียร และ มนู อรดีดลเชษฐ์. 2542. การพัฒนาซอฟต์แวร์. สำนักพิมพ์ดี แอล เอส: กรุงเทพฯ.

Ammann, P., and Offutt, J. 2008. Introduction to Software Testing. Cambridge University Press: USA.

Chen, T. Y., and Poon, P.L. 1998. Teaching black–box testing. Proceedings of the 1998 International Conference on Software Engineering: Education and Practice. IEEE Computer Society Press. January 1998. pp. 324-329.

Dix, A. 1993. Human-computer interaction: Help and documentation. Cambridge University: Great Britain.

Glenford, J., and Myers. 2004. The Art of Software Testing. Second Edition. John Wiley & Sons, Inc.: USA.

IEEE (The Institute of Electrical and Electronics Engineers). 1997. IEEE Standard for Developing Software Life Cycle Processes. IEEE Standard 1074. 1997 Edition. IEEE Computer Society.

IEEE (The Institute of Electrical and Electronics Engineers). 1998. IEEE Recommended Practice for Software Requirements Specifications. IEEE Standard 830. 1998 Edition. IEEE Computer Society.

Peters, J. F., and Pedrycz, W. 2000. Software Engineering : An Engineering Approach. Fifth Edition. John Wiley & Sons, Inc. : USA.

Raamesh, L., and Uma, G. V. 2010. Reliable Mining of Automatically Generated Test Cases from Software Requirements Specification. IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 1, No. 3. January 2010. pp. 87-91.

Sommerville, I. 1998. Software Engineering. Fifth Edition. Addison Wesley Longman Limited: England.

Sommerville, I. 2011. Software Engineering. Ninth Edition. Addison Wesley Longman Limited: America.

Tahat, L. H., Vaysburg, B., Korel, B., and Bader, A. J. 2001. Requirement-Based Automated Black-Box Test Generation, IEEE Transaction on Software Engineering. pp. 489-495.

Williams, L. 2006. Testing Overview and Black-Box Testing Techniques. North Carolina State University Department of Computer Science.

ภาคผนวก

ภาคผนวก ก.

ผลงานตีพิมพ์

เรื่อง	กรอบแนวคิดการสร้างกรณีทดสอบจากคุณลักษณะความต้องการในรูปแบบฟอร์ม
งานประชุมวิชาการ	The 5th National Conference and 2012 International Conference on Applied Computer Technology and Information Systems (ACTIS 2012)
จัดโดย	มหาวิทยาลัยทักษิณ ประเทศไทย
สถานที่	สงขลา ประเทศไทย
วันที่	22 – 23 กันยายน 2555

กรอบแนวคิดการสร้างกรณีทดสอบจากคุณลักษณะความต้องการในรูปแบบฟอร์ม

A Framework for Generating Test Cases from Form-Based Requirements Specification

สิริมา นาคเตี้ย
สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่
อ.หาดใหญ่ จ.สงขลา 90110
E-mail: p.294@live.com

อำนาจ เปาะทอง
สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่
อ.หาดใหญ่ จ.สงขลา 90110
E-mail: amnart.p@psu.ac.th

บทคัดย่อ

บทความนี้ในกระบวนการทดสอบซอฟต์แวร์ กรณีทดสอบถือเป็นส่วนสำคัญที่จะนำมาซึ่งซอฟต์แวร์ที่มีคุณภาพ สามารถวางแผนดำเนินงานได้จากขั้นตอนวิศวกรรมความต้องการ โดยเฉพาะการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับ (acceptance testing) การสร้างกรณีทดสอบสำหรับการตรวจรับซอฟต์แวร์ของลูกค้าหรือผู้ใช้ที่ขาดประสบการณ์เป็นเรื่องยุ่งยาก ประกอบกับผู้จัดทำข้อกำหนดคุณลักษณะของซอฟต์แวร์ส่วนใหญ่มักเป็นฝั่งของผู้พัฒนาซอฟต์แวร์ ไม่ใช่ฝั่งของผู้ต้องการซอฟต์แวร์ ดังนั้นงานวิจัยนี้จึงนำเสนอกรอบแนวคิดของการสร้างกรณีทดสอบจากคุณลักษณะความต้องการในรูปแบบฟอร์ม (form-based requirements specification) โดยสามารถสร้างกรณีทดสอบได้โดยอัตโนมัติตามกรอบแนวคิดที่ได้นำเสนอ ลูกค้าสามารถนำกรณีทดสอบที่สร้างขึ้นไปใช้ประโยชน์ในการตรวจรับซอฟต์แวร์ ทั้งนี้ยังสามารถตรวจสอบความถูกต้องของซอฟต์แวร์ได้แต่เนิ่นๆ เป็นการช่วยประหยัดเวลาและค่าใช้จ่ายในกระบวนการผลิตซอฟต์แวร์

คำสำคัญ: การทดสอบซอฟต์แวร์ กรณีทดสอบ คุณลักษณะความต้องการในรูปแบบฟอร์ม

Abstract

In software testing process, test cases are very important that lead to software quality. Many errors found in software products usually come from unsuitable test cases. Test case generation can be planned from the requirements engineering phase, especially test cases for acceptance testing. However, many customers perform acceptance testing by ad hoc. They lack systematic approach for creating appropriate test cases. Therefore, this research paper proposes a framework of test-case generation based on form-based requirements specification. The test-case will be generated automatically by the generator, according to the proposed framework. Produced test cases would benefit customers during acceptance testing process. Errors in a software product can be found at the beginning of its operation or before.

Keywords: software testing, test case, form-based requirements specification

1. บทนำ

การทดสอบซอฟต์แวร์เป็นขั้นตอนที่มีความสำคัญและมีค่าใช้จ่ายสูง มักดำเนินการในช่วงการโปรแกรมและการส่งมอบผลิตภัณฑ์ การที่สามารถสร้างกรณีทดสอบที่สอดคล้องกับคุณลักษณะความต้องการของซอฟต์แวร์ และสามารถวางแผนตรวจสอบความถูกต้องของซอฟต์แวร์ได้แต่เนิ่นๆ จะทำให้ประหยัดเวลาและค่าใช้จ่ายในกระบวนการผลิตซอฟต์แวร์

ปัจจุบันพบว่ามีงานวิจัยที่นำคุณลักษณะความต้องการมาใช้ในการสร้างกรณีทดสอบ เช่น L. H. Tahat และคณะ [1] นำเสนอเทคนิคการสร้างการทดสอบบนพื้นฐานของความต้องการ ด้วยวิธีการรับ

คุณลักษณะซอฟต์แวร์จากแต่ละบุคคลแสดงในรูปของข้อความ แล้วนำข้อมูลนั้นมาสร้างแบบจำลองของระบบด้วย SDL (Specification Description Language) จากนั้นนำแบบจำลองที่ได้ไปสร้างกรณีทดสอบที่สัมพันธ์กับคุณลักษณะความต้องการ L. Raamesh และ G. V. Uma [2] นำเสนอการเขียนคุณลักษณะความต้องการ โดยแยกประเภทเป็นความต้องการเชิงฟังก์ชันและไม่เป็นฟังก์ชัน จากคุณลักษณะความต้องการซอฟต์แวร์ (Software Requirements Specification: SRS) จากนั้นจึงสร้างกรณีทดสอบจาก SRS โดยเริ่มจากการแปลง SRS ให้อยู่ในรูปของ UML (Unified Modeling Language) แล้วสร้างกรณีทดสอบจาก UML และคัดสรรกรณีทดสอบที่เหมาะสม

เมื่อพิจารณางานวิจัยที่กล่าวมาข้างต้นพบว่าคุณลักษณะความต้องการมักถูกนำมาสร้างแบบจำลองก่อน แล้วจึงนำแบบจำลองนั้นมาสร้างกรณีทดสอบ ไม่ได้มีการนำคุณลักษณะความต้องการมาสร้างกรณีทดสอบโดยตรง ทั้งที่คุณลักษณะความต้องการที่นิยมเขียนมักอยู่ในรูปของแบบฟอร์ม เนื่องจากง่ายต่อการสร้างในลักษณะโครงสร้าง และง่ายต่อการทำความเข้าใจของผู้เกี่ยวข้อง โดยเฉพาะลูกค้าหรือผู้ใช้ที่ต้องการซอฟต์แวร์ เพราะยังอยู่บนพื้นฐานของภาษาธรรมชาติ นอกจากนี้พบว่าการสร้างกรณีทดสอบสำหรับการตรวจรับซอฟต์แวร์ของลูกค้าซึ่งขาดประสบการณ์เป็นเรื่องยุ่งยากจะพบข้อผิดพลาดในซอฟต์แวร์เมื่อใช้ผลิตภัณฑ์ไปสักระยะหนึ่งแล้ว ประกอบกับผู้จัดทำคุณลักษณะของซอฟต์แวร์ส่วนใหญ่มักเป็นฝั่งของผู้พัฒนาซอฟต์แวร์ ไม่ใช่ฝั่งของผู้ต้องการซอฟต์แวร์ อีกทั้งการสื่อสารระหว่างนักพัฒนาซอฟต์แวร์กับผู้ใช้ระบบมักเกิดปัญหาเรื่องความเข้าใจไม่ตรงกัน

ดังนั้นงานวิจัยนี้จึงนำเสนอกรอบแนวคิดในการสร้างกรณีทดสอบจากคุณลักษณะความต้องการในรูปแบบฟอร์ม เพื่อสนับสนุนนักพัฒนาซอฟต์แวร์และผู้ใช้ ช่วยให้การสร้างกรณีทดสอบเริ่มต้นได้ตั้งแต่ขั้นตอนของการระบุความต้องการ โดยใช้ข้อกำหนดคุณลักษณะความต้องการ จึงช่วยลดความเสี่ยงในการพัฒนาระบบในขั้นตอนต่อไป อีกทั้งการใช้เครื่องมือเพื่อช่วยสร้างกรณีทดสอบมีส่วนสำคัญที่จะช่วยลดระยะเวลาในการสร้างกรณีทดสอบ นักพัฒนาสามารถใช้เครื่องมือนี้ช่วยในขั้นตอนการระบุความต้องการของผู้ใช้ และยังสามารถสร้างกรณีทดสอบจากความต้องการนี้ได้ในคราวเดียว จึงสามารถนำไปเป็นระบบแนะนำการสร้างกรณีทดสอบระดับการตรวจรับผลิตภัณฑ์ของผู้ใช้ได้ โดยผู้ใช้สามารถนำกรณีทดสอบที่ได้ไปใช้ทดสอบระบบในขั้นตอนของการทดสอบการยอมรับ

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ข้อกำหนดคุณลักษณะความต้องการ

ข้อกำหนดคุณลักษณะความต้องการเป็นกิจกรรมที่กำหนดรายละเอียดความต้องการของระบบ และเงื่อนไขของซอฟต์แวร์ที่จะถูกสร้าง [3] ในขั้นตอนนี้ผู้ผลิตซอฟต์แวร์จะขยายนิยามความต้องการแต่ละรายการให้มีรายละเอียดที่ชัดเจน สามารถนำไปสู่การออกแบบระบบซอฟต์แวร์ โดยกระบวนการวิศวกรรมความต้องการประกอบด้วยขั้นตอนดังนี้

1) การศึกษาความเป็นไปได้ (feasibility study)

2) การวิเคราะห์ความต้องการ (requirements analysis)

3) การนิยามความต้องการ (requirements definition)

4) การกำหนดคุณลักษณะความต้องการ (requirements specification)

การเขียนคุณลักษณะความต้องการในรูปแบบฟอร์มได้รับความนิยมในกลุ่มนักพัฒนา เนื่องจากมีลักษณะโครงสร้างในการเขียนชัดเจน ง่ายต่อการทำความเข้าใจของผู้เกี่ยวข้อง โดยมีแนวทางการเขียนบนพื้นฐานแบบฟอร์มมาตรฐาน (form-based) ประกอบด้วยหัวข้อที่นิยมดังแสดงในตารางที่ 1

2.2 การกำหนดรูปแบบการสร้างกรณีทดสอบ

ในส่วนของเอกสารกรณีทดสอบมักมีรูปแบบแตกต่างกันไปตามวัตถุประสงค์ในการใช้งาน แต่หัวข้อส่วนใหญ่ที่นิยมใช้มักประกอบไปด้วยส่วนต่างๆ ดังนี้

1) ชื่อกรณีทดสอบ (Test case)

2) คำอธิบายรายละเอียดในการทดสอบ (Description)

3) ขั้นตอนในการทดสอบ (Step)

4) ข้อมูลนำเข้าที่จะใช้ทดสอบในกรณีต่างๆ (Input)

5) ความคาดหวังสิ่งที่จะเป็นผลลัพธ์ ซึ่งจะต้องตรงกับความต้องการของผู้ใช้ (Expected result)

6) ผลลัพธ์จากการทดสอบ (Actual result)

ตารางที่ 1: รูปแบบการเขียนคุณลักษณะความต้องการบนพื้นฐานแบบฟอร์ม (Form-based requirements specification) [3]

Function	ชื่อฟังก์ชันงาน
Description	อธิบายรายละเอียดของฟังก์ชันงานโดยสรุป
Inputs	ระบุข้อมูลนำเข้า
Source	ระบุอุปกรณ์นำเข้าข้อมูล
Outputs	ระบุข้อมูลนำออก/ผลลัพธ์
Destination	ระบุอุปกรณ์แสดงผลลัพธ์
Requires	ระบุสิ่งที่จำเป็นในการทำงานของฟังก์ชัน
Pre-condition	ระบุเงื่อนไขก่อนการดำเนินงานของฟังก์ชัน
Post-condition	ระบุเงื่อนไขหลังการดำเนินงานของฟังก์ชัน

3. วิธีการดำเนินการวิจัย

3.1 นิยามความต้องการ (Requirements definition)

นิยามความต้องการในลักษณะฟังก์ชันงานโดยสรุปว่าต้องการให้ฟังก์ชันงานนั้นมีการดำเนินงานอย่างไรบ้าง ประกอบด้วยชื่อฟังก์ชัน (Function) และคำอธิบายการทำงานโดยสรุป (Description) ซึ่งจะนำไปอธิบายในระดับรายละเอียดเป็นคุณลักษณะความต้องการต่อไป ทั้งนี้ผู้ใช้ควรเป็นผู้สร้างหรือกำหนดนิยามความต้องการเอง

3.2 เขียนคุณลักษณะความต้องการให้อยู่ในรูปแบบฟอร์ม (Form-based requirements specification)

เขียนคุณลักษณะความต้องการในรูปแบบฟอร์มสำหรับแต่ละนิยามความต้องการ แสดงตัวอย่างการเขียนคุณลักษณะความต้องการในรูปแบบฟอร์มดังตารางที่ 2

3.3 สร้างกรณีทดสอบ (Generate test case)

การสร้างกรณีทดสอบเบื้องต้นด้วยเครื่องมือที่พัฒนาขึ้นนี้ จะนำคุณลักษณะความต้องการในรูปแบบฟอร์มของผู้ใช้มาเป็นข้อมูลนำเข้าในการสร้างกรณีทดสอบ โดยมีขั้นตอนในการสร้างกรณีทดสอบดังนี้

1) ระบบจะนำข้อมูลในส่วนของชื่อฟังก์ชันงานจากแบบฟอร์มมาแสดงเป็นข้อมูลในส่วนของชื่อกรณีทดสอบ และนำข้อมูลในส่วนของคำอธิบายรายละเอียดของฟังก์ชันงานจากแบบฟอร์ม มาแสดงเป็นข้อมูลในส่วนของคำอธิบายกรณีทดสอบ

ตารางที่ 2: คุณลักษณะความต้องการของฟังก์ชันงานสอบถามยอดเงิน

Function	การสอบถามยอดเงิน
Description	เป็นฟังก์ชันงานที่ผู้ใช้บริการจะต้องกดเลือกสอบถามยอดเงิน จากนั้นระบบจะแสดงยอดเงินในบัญชีผู้ใช้ออกทางจอภาพของตู้เอทีเอ็ม
Inputs	การกดปุ่มที่ใช้สำหรับสอบสอบยอดเงิน
Source	ตู้เอทีเอ็ม
Outputs	ยอดเงินในบัญชีผู้ใช้
Destination	หน้าจอแสดงผลของตู้เอทีเอ็ม
Requires	ผู้ใช้กดเลือกสอบถามยอดเงิน
Pre-condition	ล๊อคอินเข้าสู่หน้าจอของการใช้งานระบบ
Post-condition	ปรากฏข้อความแจ้งการเลือกดำเนินงานต่อ

2) ระบบจะนำข้อมูลในส่วนของสิ่งที่ระบบต้องการ (Requires) และ เงื่อนไขก่อนการดำเนินงาน (Pre-condition) จากแบบฟอร์ม มาพิจารณาเพื่อสร้างเป็นขั้นตอนในการทดสอบ (Step) โดยการพิจารณา Pre-condition ว่ามีหรือไม่ หากไม่มีผู้ใช้จะระบุเป็น None ระบบจะนำ Requires มาแสดงเป็น Step ของกรณีทดสอบทันที แต่หากผู้ใช้มีการระบุ Pre-condition ไว้ ระบบจะนำ Pre-condition มาแสดงเป็นขั้นตอนแรกในการทดสอบระบบ จากนั้นนำ Requires มาแสดงเป็นขั้นตอนต่อไปในการทดสอบ

3) สำหรับการสร้างชุดข้อมูลนำเข้าและผลลัพธ์ที่คาดหวัง (Expected result) ของกรณีทดสอบ มีขั้นตอนดังนี้

3.1) นำคำอธิบายฟังก์ชันงานมาเปรียบเทียบกับกลุ่มคำแต่ละประเภทในฐานข้อมูลระบบ ว่าคุณลักษณะความต้องการเป็นฟังก์ชันงานประเภทใด ดังภาพที่ 1 ฟังชันก์การสอบถามยอดเงินจัดอยู่ในประเภทที่ 12 โดยจะเก็บค่าไว้ในตัวแปร TypeID เท่ากับ 12 ซึ่ง TypeID แต่ละค่าก็จะมีการดำเนินงานในการสร้างชุดข้อมูลนำเข้าและผลลัพธ์ที่คาดหวังของกรณีทดสอบแตกต่างกันไป

3.2) สำหรับการดำเนินงานของประเภทที่ 12 ระบบจะนำข้อมูลนำเข้า (Inputs) ของคุณลักษณะความต้องการมาตรวจสอบ ถ้าข้อมูลนำเข้าที่รับมาเป็นการกดปุ่มในการดำเนินงานจะเก็บค่า subID

เท่ากับ 1 แต่ถ้าเป็นการป้อนข้อมูลโดยผู้ใช้ หรือกดเลือกข้อมูลก่อนการดำเนินงาน ค่าของตัวแปร subID จะเท่ากับ 2

Description	เป็นฟังก์ชันงานที่ผู้ใช้บริการจะต้องกดเลือกสอบถามยอดเงิน จากนั้นระบบจะแสดงยอดเงินในบัญชีผู้ใช้ออกทางจอภาพของตู้เอทีเอ็ม

ประเภท	กลุ่มคำ
type[1]	เข้าใช้ระบบ,เข้าใช้งานระบบ,เข้าสู่ระบบ,เข้าสู่การใช้งานระบบ,ล๊อกอิน,ล๊อคอิน,ล้อกอิน,ล้อคอิน
. . .	. . .
type[12]	สอบถาม,ตรวจสอบ

ภาพที่ 1: การหาประเภทของฟังชันก์งาน

3.3) subID เท่ากับ 1 ชุดข้อมูลในการทดสอบจะมีจำนวน 1 ชุด โดยระบบจะนำข้อมูลในส่วนของข้อมูลนำเข้าและอุปกรณ์นำเข้าข้อมูล (Source) จากแบบฟอร์ม มาพิจารณาเพื่อสร้างข้อมูลนำเข้าของกรณีทดสอบ และนำข้อมูลในส่วนของข้อมูลนำออก (Outputs) อุปกรณ์แสดงผลลัพธ์ (Destination) และเงื่อนไขหลังการดำเนินงานจากแบบฟอร์ม (Post-condition) มาสร้างผลลัพธ์ที่คาดหวังของกรณีทดสอบ แสดงกรณีทดสอบของการสอบถามยอดเงินดังภาพที่ 2

Test case: การสอบถามยอดเงิน

Description: เป็นกรณีทดสอบที่ผู้ใช้บริการจะต้องกดเลือกสอบถามยอดเงิน จากนั้นระบบจะแสดงยอดเงินในบัญชีผู้ใช้ออกทางจอภาพของตู้เอทีเอ็ม

Step: 1.ล๊อคอินเข้าสู่หน้าจอของการใช้งานระบบ
2.ผู้ใช้กดเลือกสอบถามยอดเงิน

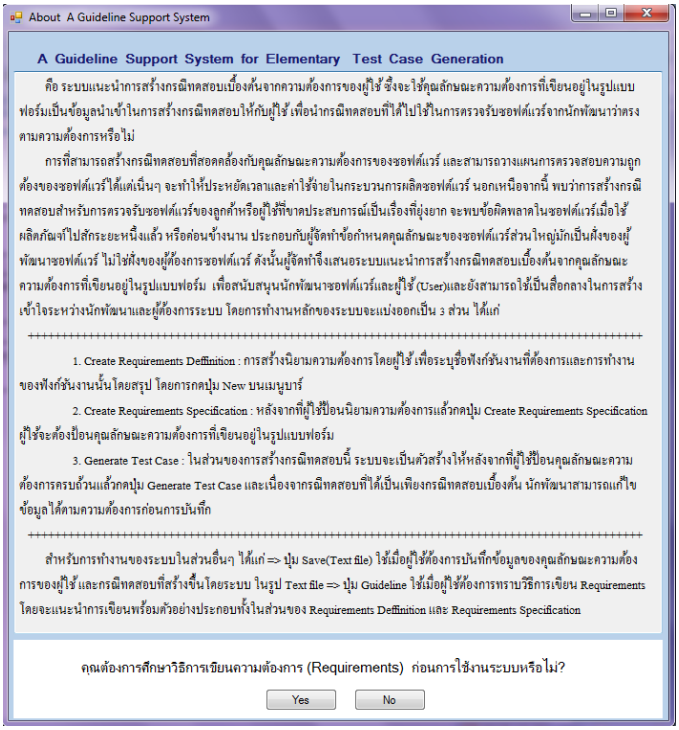
Input	Expected result
การกดปุ่มที่ใช้สำหรับสอบถามยอดเงิน (รับข้อมูลจากตู้เอทีเอ็ม)	1.แสดงยอดเงินในบัญชีผู้ใช้ (ออกทางหน้าจอแสดงผลของตู้เอทีเอ็ม) 2.ปรากฏข้อความแจ้งการเลือกดำเนินงานต่อ

ภาพที่ 2: กรณีทดสอบของฟังก์ชันการสอบถามยอดเงิน

4. ผลการดำเนินงาน

4.1 ผลการพัฒนาระบบ

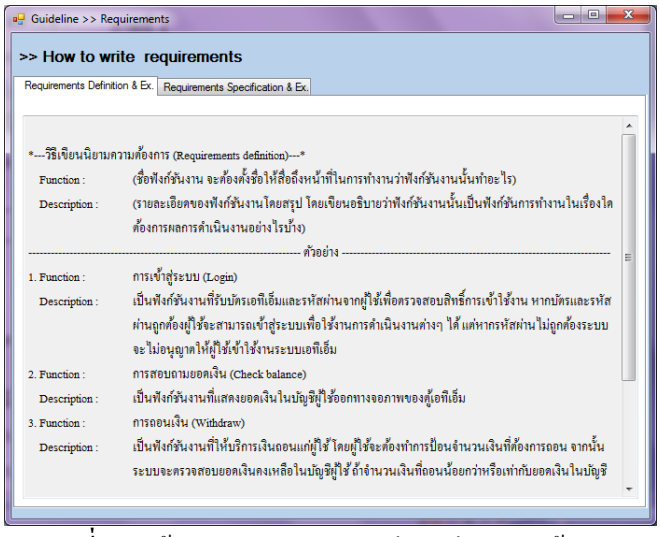
ผลการดำเนินงานของการพัฒนาเครื่องมือสร้างกรณีทดสอบจากคุณลักษณะความต้องการในรูปแบบฟอร์ม หน้าจอแรกของระบบเป็นคำอธิบายการทำงานเกี่ยวกับระบบ แสดงดังภาพที่ 3



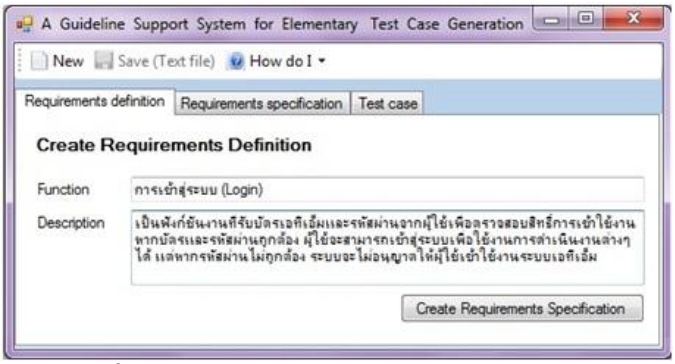
ภาพที่ 3: หน้าจอแสดงคำอธิบายการใช้งานระบบ

และในกรณีที่ผู้ใช้ขาดความรู้ความชำนาญในการเขียนนิยามความต้องการและคุณลักษณะความต้องการ ระบบมีส่วนของการแนะนำการเขียนความต้องการ พร้อมตัวอย่างประกอบเพื่อให้ผู้ใช้ได้ศึกษา แสดงดังภาพที่ 4

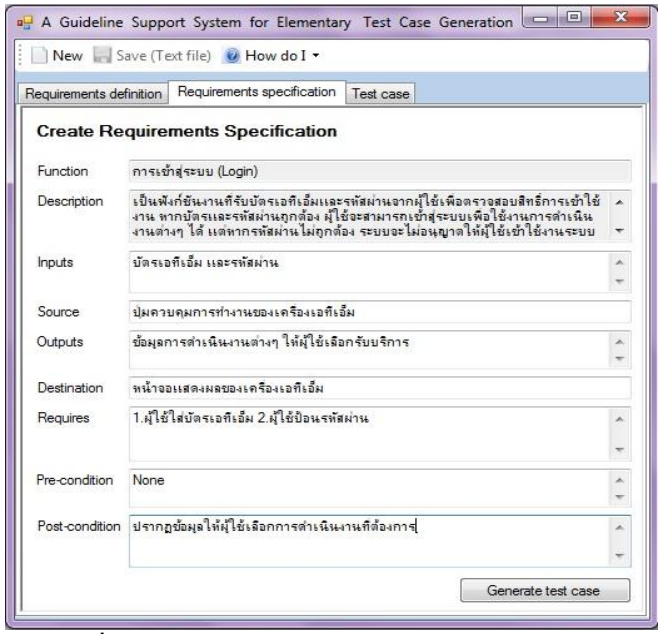
ในขั้นแรกผู้ใช้จะต้องกดปุ่ม "New" เพื่อทำการป้อนข้อมูลนิยามความต้องการของฟังก์ชันงาน ดังภาพที่ 5 แล้วกดปุ่ม "Create Requirements Specification" เพื่อเริ่มทำการป้อนข้อมูลคุณลักษณะความต้องการในรูปแบบฟอร์ม ดังภาพที่ 6 โดยผู้ใช้จะต้องป้อนข้อมูลให้ครบถ้วน ยกเว้นในส่วนของชื่อฟังก์ชัน (Function) และคำอธิบายการทำงานของฟังก์ชันโดยสรุป (Description) เนื่องจากระบบจะนำข้อมูลจากนิยามความต้องการที่ผู้ใช้ได้ระบุไว้ข้างต้นมาแสดงในส่วนนี้แล้ว



ภาพที่ 4: หน้าจอแสดงการแนะนำวิธีการเขียนความต้องการ

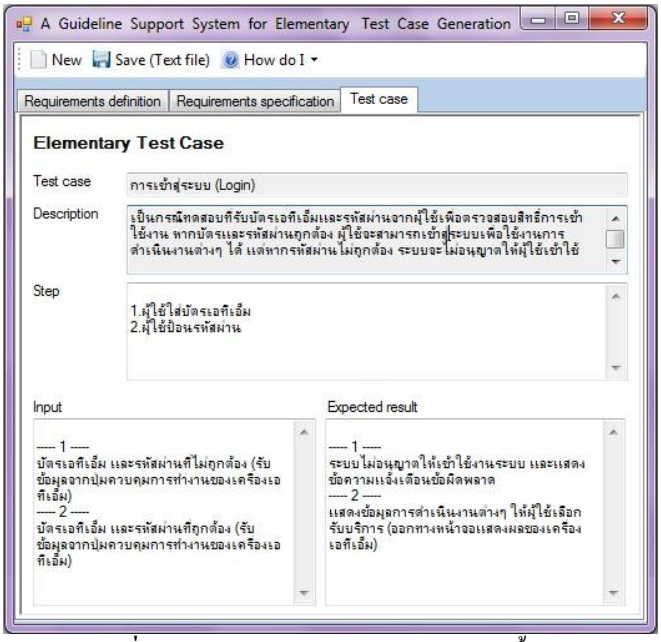


ภาพที่ 5: หน้าจอการป้อนข้อมูลของนิยามความต้องการ



ภาพที่ 6: หน้าจอการป้อนข้อมูลคุณลักษณะความต้องการ

หลังจากผู้ใช้ได้ป้อนข้อมูลทั้ง 2 ส่วนดังภาพที่ 5 และภาพที่ 6 แล้ว ระบบจะนำข้อมูลคุณลักษณะความต้องการของผู้ใช้มาวิเคราะห์ และสร้างกรณีทดสอบเบื้องต้นดังแสดงในภาพที่ 7



ภาพที่ 7: หน้าจอแสดงข้อมูลกรณีทดสอบเบื้องต้น

4.2 ประเมินคุณภาพระบบโดยผู้ใช้ทั่วไป

ในการเตรียมพร้อมสำหรับการประเมินการใช้งานระบบจริง ได้มีการสอบถามความพึงพอใจต่อการใช้งานระบบเบื้องต้นในสถานการณ์ที่มีความอิสระและไม่มีการควบคุม ด้วยวิธีการส่งซอฟต์แวร์และแบบสอบถามให้กับผู้ใช้ทั่วไปที่ใกล้เคียงกับกลุ่มเป้าหมายจำนวน 20 ท่าน ผลการประเมินในด้านต่างๆ สามารถแสดงดังตารางที่ 3

ตารางที่ 3: ผลการประเมินคุณภาพของระบบโดยผู้ใช้ทั่วไป

รายการ	Mean	S.D.	ระดับความพึงพอใจ
1. ด้านการใช้งานระบบ	4.05	0.75	ดี
2. ด้านการแสดงผลข้อมูล	3.81	0.70	ดี
3. ด้านประสิทธิภาพของระบบ	4.27	0.70	ดี
4. ด้านคุณค่าของระบบ	3.90	0.75	ดี
สรุปผล	4.00	0.73	ดี

จากตารางผลการประเมินคุณภาพของระบบโดยผู้ใช้ทั่วไปพบว่าผลการประเมินในแต่ละด้านมีค่าเฉลี่ย (Mean) โดยรวมเท่ากับ 4.00 และมีส่วนเบี่ยงเบนมาตรฐาน (S.D.) เท่ากับ 0.73 ดังนั้นผู้ใช้ทั่วไปมีความคิดเห็นว่าระบบที่พัฒนาขึ้นมีคุณภาพอยู่ในระดับดี (ตามเกณฑ์การประเมินความพึงพอใจ 5 ช่วง คือ ดีมาก: 4.50-5.00 ดี: 3.50-4.49 ปานกลาง: 2.50-3.49 พอใช้: 1.50-2.49 และควรปรับปรุง: 0.50-1.49)

ทั้งนี้ผู้ใช้ทั่วไปได้เสนอความคิดเห็นว่าส่วนของคำอธิบายวิธีการเขียนความต้องการ (Guideline) ควรอยู่ในหน้าเดียวกันกับส่วนของการเขียนคุณลักษณะความต้องการเพื่อความสะดวกในการทำความเข้าใจระบบ เป็นโปรแกรมที่มีแนวคิดดี สามารถนำไปต่อยอดเพิ่มประสิทธิภาพการทำงานให้เพิ่มขึ้นหรือหลากหลายมากขึ้น ส่วนของการทำงานก็ให้ผลลัพธ์ที่น่าพอใจ

5. บทสรุปและงานในอนาคต

การสร้างกรณีทดสอบเป็นขั้นตอนหนึ่งที่สำคัญในการทดสอบซอฟต์แวร์ งานวิจัยนี้ได้นำเสนอวิธีการสร้างกรณีทดสอบจากคุณลักษณะความต้องการในรูปแบบฟอร์ม ซึ่งสามารถเริ่มขั้นตอนการทดสอบซอฟต์แวร์ได้เร็วขึ้นตั้งแต่ขั้นตอนการระบุความต้องการ โดยการนำคุณลักษณะความต้องการมาเขียนในรูปแบบฟอร์ม เพื่อให้ง่ายต่อการทำความเข้าใจ มีการระบุข้อมูลในส่วนต่างๆ ของแบบฟอร์มอย่างชัดเจน จึงง่ายต่อการนำมาสร้างกรณีทดสอบ อีกทั้งยังช่วยลดความเสี่ยงในการพัฒนาระบบในขั้นตอนต่อไป เพราะหากทำการทดสอบได้เร็วก็จะมีโอกาสพบข้อบกพร่องในการพัฒนาซอฟต์แวร์ได้เร็วขึ้นเช่นกัน และเนื่องจากเครื่องมือที่พัฒนาขึ้นเพื่อมารองรับการทำงานภายใต้กรอบแนวคิดของผู้วิจัยนี้ อาจไม่ครอบคลุมความต้องการที่มีอยู่มากมาย ดังนั้นในอนาคตผู้วิจัยจะพัฒนาเครื่องมือในการสร้างกรณีทดสอบจากคุณลักษณะความต้องการนี้ให้มีประสิทธิภาพมากขึ้น อีกทั้งสามารถบันทึกข้อมูลการสร้างกรณีทดสอบของผู้ใช้ในแต่ละครั้งไว้ในฐานข้อมูล เพื่อนำมาสร้างกรณีทดสอบได้โดยอัตโนมัติในกรณีที่คุณลักษณะความต้องการของผู้ใช้มีความใกล้เคียงกับข้อมูลที่มีอยู่ในฐานข้อมูลของระบบ

ในส่วนของการประเมินผู้วิจัยจะดำเนินการประเมินเพิ่มเติม โดยแบ่งการประเมินออกเป็น 2 ส่วนหลัก คือ การประเมินประสิทธิผลของกรณีทดสอบที่ระบบสร้างขึ้นโดยผู้เชี่ยวชาญ และการประเมินความพึงพอใจของผู้ใช้งานระบบในห้องปฏิบัติการที่มีการควบคุมกึ่งการทดลอง

เอกสารอ้างอิง

[1] L. H. Tahat, B. Vaysburg, B. Korel, and A. J. Bader, "Requirement-Based Automated Black-Box Test Generation", *IEEE Transaction on Software Engineering*, 2001, pp. 489-495.

[2] L. Raamesh, and G. V. Uma, "Reliable Mining of Automatically Generated Test Cases from Software Requirements Specification", *IJCSI*, Vol 7, Issue 1, No. 3, January 2010.

[3] I. Sommerville, *"Software Engineering"*, Addison Wesley, 1995.

[4] P. Ammann, and J. Offutt, *"Introduction to Software Testing"*, Cambridge University Press, USA, 2008.

[5] L. Williams, *"Testing Overview and Black-Box Testing Techniques"*, North Carolina State University Department of Computer Science, 2006.

[6] J. Glenford, and Myers, *"The Art of Software Testing. Second Edition",* John Wiley & Sons, Inc. USA, 2004.

ภาคผนวก ข.

พจนานุกรมข้อมูล

พจนานุกรมข้อมูล (Data Dictionary) ในระดับรายละเอียดดังตาราง ข-1 ถึง ข-6

สัญลักษณ์ที่ใช้ในการกำหนดชนิดของข้อมูล ได้แก่
int แทน ข้อมูลจำนวนตัวเลข 2^{32}
varchar แทน ข้อมูลอักขระ
text แทน ข้อมูลข้อความ

สัญลักษณ์ที่ใช้แทนคีย์ ได้แก่
PK แทน คีย์หลัก (Primary Key)
FK แทน คีย์นอก (Foreign Key)

ตาราง ข-1 ข้อมูล User (ข้อมูลผู้ใช้)

ลำดับ	ชื่อเขตข้อมูล	ชนิด	คำอธิบาย	คีย์
1	U_id	int (10)	รหัสผู้ใช้	PK
2	U_name	varchar (20)	ชื่อ-สกุล	
3	U_userName	varchar (10)	ชื่อเข้าใช้งาน	
4	U_password	varchar (10)	รหัสผ่าน	
5	U_mail	varchar (20)	อีเมล	
6	U_numPhone	varchar (10)	เบอร์โทรศัพท์	
7	U_type	varchar (10)	ประเภทผู้ใช้	
8	P_id	int (10)	รหัสโครงงาน	FK (Project)

ตาราง ข-2 ข้อมูล Project (ข้อมูลโครงงาน)

ลำดับ	ชื่อเขตข้อมูล	ชนิด	คำอธิบาย	คีย์
1	P_id	int (10)	รหัสโครงงาน	PK
2	P_name	varchar (50)	ชื่อ	
3	P_desc	text (255)	คำอธิบายการทำงาน	

ตาราง ข-2 ข้อมูล Project (ข้อมูลโครงงาน) (ต่อ)

ลำดับ	ชื่อเขตข้อมูล	ชนิด	คำอธิบาย	คีย์
4	P_create	varchar (50)	ผู้สร้างโครงงาน	
5	P_date	varchar (20)	วัน/เดือน/ปี ที่สร้าง	
6	P_team	varchar (50)	ทีมนักพัฒนา	
7	P_status	varchar (10)	สถานะโครงงาน	
8	P_statusDef	varchar (10)	สถานะนิยามความต้องการ	
9	RD_id	int (10)	รหัสนิยามความต้องการ	FK (ReqDef)
10	U_id	int (10)	รหัสผู้ใช้	FK (User)

ตาราง ข-3 ข้อมูล ReqDef (ข้อมูลนิยามความต้องการ)

ลำดับ	ชื่อเขตข้อมูล	ชนิด	คำอธิบาย	คีย์
1	RD_id	int (10)	รหัสนิยามความต้องการ	PK
2	RD_name	varchar (20)	ชื่อ	
3	RD_desc	text (255)	คำอธิบายการทำงาน	
4	RS_id	int (10)	รหัสคุณลักษณะความต้องการ	FK (ReqSpec)
5	P_id	int (10)	รหัสโครงงาน	FK (Project)

ตาราง ข-4 ข้อมูล ReqSpec (ข้อมูลคุณลักษณะความต้องการ)

ลำดับ	ชื่อเขตข้อมูล	ชนิด	คำอธิบาย	คีย์
1	RS_id	int (10)	รหัสคุณลักษณะความต้องการ	PK
2	RS_name	varchar (20)	ชื่อ	
3	RS_desc	Text (255)	ขั้นตอนการทำงาน	
4	RS_input	Text (255)	ข้อมูลนำเข้า	
5	RS_source	Text (255)	อุปกรณ์นำเข้าข้อมูล	
6	RS_output	Text (255)	ข้อมูลนำออก	
7	RS_dest	varchar (50)	อุปกรณ์แสดงผลลัพธ์	

ตาราง ข-4 ข้อมูล ReqSpec (ข้อมูลคุณลักษณะความต้องการ) (ต่อ)

ลำดับ	ชื่อเขตข้อมูล	ชนิด	คำอธิบาย	คีย์
8	RS_req	Text (255)	สิ่งที่ต้องการ	
9	RS_pre	Text (255)	เงื่อนไขก่อนการดำเนินงาน	
10	RS_post	Text (255)	เงื่อนไขหลังการดำเนินงาน	
11	RS_status	varchar (10)	สถานะ	
12	RD_id	int (10)	รหัสนิยามความต้องการ	FK (ReqDef)
13	D_id	int (10)	รหัสตารางตัดสินใจ	FK (DecisionTable)
14	V_id	int (10)	รหัสค่าข้อมูลที่เป็นไปได้	FK (ValueInput)

ตาราง ข-5 ข้อมูล DecisionTable (ข้อมูลตารางตัดสินใจ)

ลำดับ	ชื่อเขตข้อมูล	ชนิด	คำอธิบาย	คีย์
1	D_id	int (10)	รหัสตารางตัดสินใจ	PK
2	D_name	varchar (20)	ชื่อ	
3	D_input	Text (255)	ข้อมูลนำเข้า	
4	D_output	Text (255)	ข้อมูลนำออก	
5	RS_id	int (10)	รหัสคุณลักษณะความต้องการ	FK (ReqSpec)

ตาราง ข-6 ข้อมูล ValueInput (ข้อมูลค่าที่เป็นไปได้ข้องข้อมูลนำเข้า)

ลำดับ	ชื่อเขตข้อมูล	ชนิด	คำอธิบาย	คีย์
1	V_id	int (10)	รหัสค่าข้อมูลที่เป็นไปได้	PK
2	V_name	varchar (20)	ชื่อ	
3	V_value	Text (255)	ค่าที่เป็นไปได้	
5	RS_id	int (10)	รหัสคุณลักษณะความต้องการ	FK (ReqSpec)

ภาคผนวก ค.

แบบฟอร์มการประเมินกรณีทดสอบ

<u>คำชี้แจง</u> แบบประเมินนี้ใช้เพื่อการประเมินกรณีทดสอบสำหรับงานวิจัยในหัวข้อ “ระบบสนับสนุนสำหรับการสร้างกรณีทดสอบจากข้อกำหนดคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม” ของนางสาวสิริมา นาคเตี้ย นักศึกษาระดับปริญญาโทในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ โดยมี ผศ.ดร.อำนาจ เปาะทอง เป็นอาจารย์ที่ปรึกษา ขอให้ท่านตอบแบบประเมินตามความเป็นจริง ผู้วิจัยจะนำเสนอผลในภาพรวมโดยไม่กระทบต่อผู้ตอบแบบประเมิน และขอขอบคุณที่เสียสละเวลาในการตอบแบบประเมินนี้

แบบสอบถามนี้ประกอบด้วย 3 ตอน

ตอนที่ 1 ข้อมูลส่วนตัว

ตอนที่ 2 การสร้างกรณีทดสอบสำหรับระบบ ATM อย่างง่าย

ตอนที่ 3 ข้อเสนอแนะ

<u>ตอนที่ 1</u> ข้อมูลส่วนตัว

1) ชื่อ-สกุล ……..

2) เพศ ☐ ชาย ☐ หญิง อายุ.....ปี

3) ระดับและวุฒิการศึกษาสูงสุดที่ได้รับ .....

4) ชื่อหน่วยงาน / ภาควิชาที่สังกัด .....

5) ชื่อตำแหน่งงานปัจจุบัน .....

6) ลักษณะงานที่รับผิดชอบ .....

.....

7) อายุงานในตำแน่งปัจจุบัน .....ปี ..... เดือน

<u>ตอนที่ 2</u> การสร้างกรณีทดสอบสำหรับระบบ ATM อย่างง่าย

กรุณาสร้างกรณีทดสอบสำหรับแต่ละฟังก์ชันงานที่อธิบายข้อกำหนดคุณลักษณะความต้องการ (Requirements specification) ที่อยู่ในรูปแบบฟอร์มภาษาอังกฤษ โดยกรอกข้อมูลส่วน Input และ Expected result ลงในตารางของ Test case ที่ 1 – 4 โดยตอบเป็นภาษาไทยและพิจาณาจากคำอธิบายข้อกำหนดคุณลักษณะความต้องการที่อธิบายไว้ก่อนในรายงานการสร้างกรณีทดสอบสำหรับแต่ละฟังก์ชันงาน ทั้งนี้ระบบ ATM อย่างง่าย

ประกอบด้วย 4 ฟังก์ชันงานหลัก ได้แก่ Login, Check balance, Deposit และ Withdrawal ดังนี้

หมายเหตุ

Input : ข้อมูลนำเข้าแต่ละรายการจะถูกแยกด้วยเครื่องหมายจุลภาค (,)

Output : ข้อมูลผลลัพธ์แต่ละรายการจะถูกแยกด้วยเครื่องหมายทับ (/)

1. ฟังก์ชันงาน : Login (การเข้าสู่ระบบ)

Function	Login
Description	The ATM will service one customer at a time. A customer will be required to inserts the ATM card and enters a personal identification number (PIN) - both of which will be sent to the bank for validation as a part of each transaction. The customer will be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned. If the ATM card or the customer PIN is invalid, the monitor will display an error message and return the ATM card.
Inputs	The ATM card data, PIN
Source	The ATM card reader, a keyboard
Outputs	The monitor shows the main menu for performing transactions. / The monitor displays an error message and returns the ATM card.
Destination	The monitor
Requires	The ATM card, PIN, Database of customer account
Pre-condition	The ATM is ready for use.
Post-condition	The monitor show message on the screen or the menu corresponding to its output.

Test case ที่ 1

Test case : Login

Description : The ATM will service one customer at a time. A customer will be required to inserts the ATM card and enters a personal identification number (PIN) -

both of which will be sent to the bank for validation as a part of each transaction. The customer will be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned. If the ATM card or the customer PIN is invalid, the monitor will display an error message and return the ATM card.

Step : 1. A customer will be required to insert the ATM card.

2. The customer enters a personal identification number (PIN).

Input	Expected result

2. ฟังก์ชันงาน : Check balance (การสอบถามยอดเงินในบัญชี)

Function	Check balance
Description	1. A customer selects the “check balance” button. 2. The ATM displays net balance. 3. The customer chooses whether to continue or not.
Inputs	The button selection
Source	The ATM button
Outputs	The ATM displays net balance, exits and returns the ATM card. / The ATM displays net balance and displays transaction for a customer selection.

Destination	The monitor
Requires	Database of customer account
Pre-condition	The monitor displays transaction for a customer selection.
Post-condition	The monitor displays the result of the “Yes” or “No” selection

Test case ที่ 2

Test case : Check balance

Description : A customer must be able to check a balance inquiry of any account linked to the ATM card. After performing this function, the customer chooses whether to continue or not. If the customer selects “No”, the ATM will exit and return the ATM card. If the customer selects “Yes”, the customer will select another transaction.

Step : 1. A customer selects the “check balance” button.

2. The ATM displays net balance.

3. The customer chooses whether to continue or not.

Input	Expected result

3. ฟังก์ชันงาน : Deposit (การฝากเงิน)

Function	Deposit
Description	1. A customer chooses the “deposit” button 2. The customer puts the right cash in the slot and presses the confirmation button.
Inputs	An amount of deposit
Source	The ATM
Outputs	The ATM displays the total deposit and net balance in the customer account, prints a receipt and returns the ATM card. / The ATM displays a warning message and returns the ATM card.
Destination	The ATM
Requires	The deposit cash, Database of customer account
Pre-condition	The monitor displays transactions for a customer selection.
Post-condition	The monitor shows a message on the screen.

Test case ที่ 3

Test case : Deposit

Description : A customer chooses the deposit button, then puts the right cash in the slot and presses the confirmation button. The ATM will check the balance and display the net balance to the customer. The ATM will accept a deposit from 100 Bahts to 100,000 Bahts, when the customer presses a deposit confirmation. If condition of deposit is true, the ATM will display the total deposit and net balance. Then the ATM prints a receipt and returns the ATM card. If condition of deposit is false, the ATM will display a warning message and return the ATM card.

Step : 1. A customer chooses the “deposit” button.

2. The customer payments and presses the confirmation button.

Input	Expected result

Input	Expected result

4. ฟังก์ชันงาน : Withdrawal (การถอนเงิน)

Function	Withdrawal
Description	1. A customer chooses the “withdrawal” button. 2. The customer enters an amount of money.
Inputs	An amount of withdrawal
Source	The ATM button
Outputs	The ATM will allow a withdrawal from the ATM, then displays money to withdraw and balance on the monitor, then prints a receipt and return the ATM card. / The ATM will not be allowed to withdraw money from the ATM, then displays a warning message and return the ATM card.

Destination	The ATM
Requires	The amount of withdrawal, Database of customer account
Pre-condition	The monitor displays transactions for a customer selection.
Post-condition	The monitor shows a message on the screen.

Test case ที่ 4

Test case : Withdrawal

Description : A customer chooses the deposit button, then puts the right cash in the slot and presses the confirmation button. The ATM will check the balance and display the net balance to the customer. The ATM will accept a deposit from 100 Bahts to 100,000 Bahts, when the customer presses a deposit confirmation. If condition of deposit is true, the ATM will display the total deposit and net balance. Then the ATM prints a receipt and returns the ATM card. If condition of deposit is false, the ATM will display a warning message and return the ATM card.

Step : 1. A customer chooses the “withdrawal” button.

2. The customer enters an amount of money.

Input	Expected result

<u>ตอนที่ 3</u> ข้อเสนอแนะ

3.1 คำอธิบายฟังก์ชันงานเป็นภาษาอังกฤษมีผลต่อการสร้างกรณีทดสอบหรือไม่

☐ มีเพราะ……

☐ ไม่มี เพราะ……

3.2 ข้อเสนอแนะอื่นๆ

.....

.....

.....

.....

.....

.....

.....

.....

ขอบคุณที่เสียสละเวลาในการตอบแบบประเมิน

ภาคผนวก ง.

โจทย์ปัญหาสำหรับการประเมินการใช้งานระบบ

โจทย์ปัญหาการประเมินระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม (สำหรับนักวิเคราะห์ระบบ)

<u>คำชี้แจง</u>

โจทย์ปัญหาแบ่งเป็น 3 ตอน คือ

1. รายการดำเนินการ
2. ข้อเสนอแนะสำหรับรายการดำเนินการที่ไม่ผ่าน
3. คำรับรองการเข้าร่วมประเมินระบบ

1. รายการดำเนินการ

<u>ข้อแนะนำ</u> กรุณาดำเนินการตามลำดับรายการในตารางด้านล่าง แล้วใส่เครื่องหมายระบุผลลัพธ์การดำเนินการในส่วนของผลลัพธ์ ซึ่งอยู่ด้านหลังของหัวข้อรายการดำเนินการ โดยให้ใส่เครื่องหมายดังนี้

✓ หมายถึง สามารถทำรายการได้จนเสร็จสมบูรณ์ (ผ่าน)

✗ หมายถึง ไม่สามารถทำรายการได้จนเสร็จสมบูรณ์ (ไม่ผ่าน)

รายการดำเนินการ	ผลลัพธ์ (ผ่าน/ไม่ผ่าน)
อ่านคำแนะนำเกี่ยวกับระบบ (About System)	
เข้าสู่ระบบ โดยใช้ User Name: A.Bob และใช้ Password: abob	
เปิดดูคู่มือการใช้งานระบบ (User Manual)	
ศึกษาคู่มือการใช้งานสำหรับนักวิเคราะห์ในส่วนของการสร้างโครงงานใหม่	
ศึกษาคู่มือการใช้งานสำหรับนักวิเคราะห์ในส่วนของการสร้างคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม	
ศึกษาคู่มือการใช้งานสำหรับนักวิเคราะห์ในส่วนของการระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า	

รายการดำเนินการ	ผลลัพธ์ (ผ่าน/ไม่ผ่าน)
เลือกเมนู “New Project” เพื่อสร้างโครงงานใหม่ ชื่อ “ระบบสั่งซื้อออนไลน์สำหรับร้านขายดอกไม้”	
เลือกเมนู “Requirements Definition” แล้วเลือกโครงงาน “ระบบยืม-คืนหนังสือของห้องสมุด” เพื่อแก้ไขนิยามความต้องการทุกฟังก์ชันงานที่สร้างโดยผู้ใช้ทั่วไป (Customer)	
เลือกเมนู “Requirements Definition” แล้วเลือกโครงงาน “ระบบสั่งซื้อออนไลน์สำหรับร้านขายดอกไม้” เพื่อสร้างนิยามความต้องการให้กับทุกฟังก์ชันงาน	
กำหนดคุณลักษณะความต้องการ (Requirements Specification) ให้กับโครงงาน “ระบบสั่งซื้อออนไลน์สำหรับร้านขายดอกไม้” ทุกฟังก์ชันงาน	
ระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า (Possible Value of Input) ในโครงงาน “ระบบสั่งซื้อออนไลน์สำหรับร้านขายดอกไม้” ทุกฟังก์ชันงาน	
ระบุผลลัพธ์ที่คาดหวัง (Expected Result) ในตารางตัดสินใจ (Decision table)ในโครงงาน “ระบบสั่งซื้อออนไลน์สำหรับร้านขายดอกไม้” ทุกฟังก์ชันงาน	
แสดงผลกรณีทดสอบ (Test case) ของโครงงาน “ระบบสั่งซื้อออนไลน์สำหรับร้านขายดอกไม้” ทุกฟังก์ชันงานของระบบ	
บันทึกกรณีทดสอบของฟังก์ชันงานที่แสดงในรูป Text file	
เปิดแฟ้มข้อมูลกรณีทดสอบที่ได้บันทึกไว้แล้ว	
แก้ไขข้อมูลส่วนตัว (Profile)	
ออกจากระบบ	

2. ข้อเสนอแนะสำหรับรายการดำเนินการที่ไม่ผ่าน

……

……

……

……

……

3. คำรับรองการเข้าร่วมประเมินระบบ

ข้าพเจ้า ชื่อ-สกุล……..……..

ตำแหน่ง……..……..……..……..……..……..……..……..……..……..……..……

หน่วยงาน/สังกัด……..……

ขอรับรองว่าได้เข้าร่วมประเมินระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม

ลงนาม.....

(.....)

วันที่.....

โจทย์ปัญหาการประเมินระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม (สำหรับผู้ใช้ทั่วไป)

<u>คำชี้แจง</u>

โจทย์ปัญหาแบ่งเป็น 3 ตอน คือ

1. รายการดำเนินการ
2. ข้อเสนอแนะสำหรับรายการดำเนินการที่ไม่ผ่าน
3. คำรับรองการเข้าร่วมประเมินระบบ

1. รายการดำเนินการ

<u>ข้อแนะนำ</u> กรุณาดำเนินการตามลำดับรายการในตารางด้านล่าง แล้วใส่เครื่องหมายระบุผลลัพธ์การดำเนินการในส่วนของผลลัพธ์ ซึ่งอยู่ด้านหลังของหัวข้อรายการดำเนินการ โดยให้ใส่เครื่องหมายดังนี้

✓ หมายถึง สามารถทำรายการได้จนเสร็จสมบูรณ์ (ผ่าน)

✗ หมายถึง ไม่สามารถทำรายการได้จนเสร็จสมบูรณ์ (ไม่ผ่าน)

รายการดำเนินการ	ผลลัพธ์ (ผ่าน/ไม่ผ่าน)
อ่านคำแนะนำเกี่ยวกับระบบ (About System)	
เข้าสู่ระบบ โดยใช้ User Name: Customer และใช้ Password: customer	

รายการดำเนินการ	ผลลัพธ์ (ผ่าน/ไม่ผ่าน)
เปิดดูคู่มือการใช้งานระบบ (User Manual)	
ศึกษาคู่มือการใช้งานสำหรับผู้ใช้ทั่วไปทั้งหมด ประกอบด้วย 2 ขั้นตอน คือ การสร้างนิยามความต้องการ และการแสดงผลกรณีทดสอบ	
เลือกเมนู “Requirements Definition” แล้วเลือกโครงงาน “ระบบสั่งซื้อออนไลน์สำหรับร้านขายดอกไม้” เพื่อสร้างนิยามความต้องการให้กับทุกฟังก์ชันงาน	
แสดงผลกรณีทดสอบ (Test Case) ของโครงงาน ATM_test	
บันทึกกรณีทดสอบของฟังก์ชันงานที่ต้องการในรูป Text file	
เปิดแฟ้มข้อมูล (*.txt) ของกรณีทดสอบที่ได้บันทึกไว้	
แก้ไขข้อมูลส่วนตัว (Profile)	
ออกจากระบบ	

2. ข้อเสนอแนะสำหรับรายการดำเนินการที่ไม่ผ่าน

……

……

……

……

……

……

3. คำรับรองการเข้าร่วมประเมินระบบ

ข้าพเจ้า ชื่อ-สกุล……..

ตำแหน่ง……..……..……..……..……..……..……..……..……..……

หน่วยงาน/สังกัด……..……

ขอรับรองว่าได้เข้าร่วมประเมินระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม

ลงนาม.....

(.....)

วันที่.....

โจทย์ปัญหาการประเมินระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม (สำหรับผู้ดูแลระบบ)

<u>คำชี้แจง</u>

โจทย์ปัญหาแบ่งเป็น 3 ตอน คือ

1. รายการดำเนินการ
2. ข้อเสนอแนะสำหรับรายการดำเนินการที่ไม่ผ่าน
3. คำรับรองการเข้าร่วมประเมินระบบ

1. รายการดำเนินการ

<u>ข้อแนะนำ</u> กรุณาดำเนินการตามลำดับรายการในตารางด้านล่าง แล้วใส่เครื่องหมายระบุผลลัพธ์การดำเนินการในส่วนของผลลัพธ์ ซึ่งอยู่ด้านหลังของหัวข้อรายการดำเนินการ โดยให้ใส่เครื่องหมายดังนี้

✓ หมายถึง สามารถทำรายการได้จนเสร็จสมบูรณ์ (ผ่าน)

✗ หมายถึง ไม่สามารถทำรายการได้จนเสร็จสมบูรณ์ (ไม่ผ่าน)

รายการดำเนินการ	ผลลัพธ์ (ผ่าน/ไม่ผ่าน)
อ่านคำแนะนำเกี่ยวกับระบบ (About System)	
เข้าสู่ระบบ โดยใช้ User Name: Administrator และใช้ Password: admin	
สร้างบัญชีผู้ใช้ใหม่ (Create New User)	
สร้างรหัสผ่านใหม่ให้กับผู้ใช้รายเดิมที่มีอยู่ในระบบ (Create New Password for Existing User)	
จัดการรายการนักวิเคราะห์ระบบที่สามารถดำเนินการกับโครงงาน (Assign Roles) ระบบสั่งซื้อออนไลน์สำหรับร้านขายดอกไม้	
ออกจากระบบ	

2. ข้อเสนอแนะสำหรับรายการดำเนินการที่ไม่ผ่าน

.....

.....

.....

3. คำรับรองการเข้าร่วมประเมินระบบ

ข้าพเจ้า ชื่อ-สกุล……..……..

ตำแหน่ง……..……..……..……..……..……..……..……..……

หน่วยงาน/สังกัด……..……

ขอรับรองว่าได้เข้าร่วมประเมินระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม

ลงนาม.....

(.....)

วันที่.....

ภาคผนวก จ.

แบบฟอร์มการประเมินการใช้งานระบบโดยรวม

แบบประเมินประสิทธิภาพของระบบ

เรื่อง ระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม

คำชี้แจง ดิฉัน น.ส.สิริมา นาคเตี้ย เป็นนักศึกษาระดับปริญญาโท สาขาวิชาวิทยาการคอมพิวเตอร์ ได้ทำวิจัยระดับบัณฑิตศึกษาในหัวข้อเรื่อง “ระบบสนับสนุนสำหรับการสร้างกรณีทดสอบเบื้องต้นบนพื้นฐานของคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม” ภายใต้การให้คำปรึกษาวิทยานิพนธ์โดย ผศ.ดร.อำนาจ เปาะทอง ในการนี้จึงใคร่ขอความอนุเคราะห์ให้ท่านได้ตอบข้อมูลในแบบสอบถามนี้ตามความจริง เพื่อประโยชน์ในการนำผลการวิจัยไปใช้งานได้จริง และขอขอบคุณที่กรุณาสละเวลาในการตอบแบบสอบถาม

แบบประเมินแบ่งเป็น 2 ตอน คือ

1. ข้อมูลเกี่ยวกับผู้ใช้
2. แบบประเมินความพึงพอใจในการใช้งานระบบและข้อเสนอแนะ

ตอนที่ 1 ข้อมูลเกี่ยวกับผู้ใช้

1.1 ประเภทของกลุ่มผู้ใช้ ☐ ผู้เชี่ยวชาญด้านการวิเคราะห์และพัฒนาระบบ (นักวิเคราะห์ระบบ)
☐ ผู้ดูแลระบบ
☐ ผู้ใช้ทั่วไป (ผู้ต้องการซอฟต์แวร์)

1.2 เพศ ☐ ชาย ☐ หญิง อายุ.....ปี

1.4 ระดับการศึกษาสูงสุด ☐ ต่ำกว่าปริญญาตรี ☐ ปริญญาตรี ☐ ปริญญาโท
☐ ปริญญาเอก ☐ อื่นๆ ระบุ.....

1.5 วุฒิการศึกษาสูงสุด (เช่น วท.บ.คณิตศาสตร์).....

1.6 ชื่อหน่วยงาน / ภาควิชาที่สังกัด.....

1.7 ตำแหน่งงานปัจจุบัน ☐ นักศึกษา ☐ บุคลากร/บุคคลทั่วไป ตำแหน่ง.....

1.8 อายุงานในตำแหน่งปัจจุบัน .....ปี ..... เดือน

ตอนที่ 2 แบบประเมินความพึงพอใจของผู้ใช้

กรุณาใส่เครื่องหมาย ✓ ลงในช่องที่เห็นว่าใกล้เคียงกับความจริงมากที่สุด โดยมีเกณฑ์การให้คะแนนดังนี้

ระดับความพึงพอใจ 5 = ดีมาก 4 = ดี 3 = ปานกลาง 2 = พอใช้ 1 = ปรับปรุง

ข้อ	รายการประเมินผล	ระดับความพึงพอใจ				
		5	4	3	2	1
1	ด้านการทำงานของระบบ					
	1.1 ความถูกต้องของการทำงาน					
	1.2 ความสมบูรณ์ของผลลัพธ์					
	1.3 ความเร็วในการประมวลผล					
	1.4 ความสามารถในการป้องกันความผิดพลาดจากผู้ใช้					

ความคิดเห็นและข้อเสนอแนะ

.....

.....

.....

.....

.....

.....

ข้อ	รายการประเมินผล	ระดับความพึงพอใจ				
		5	4	3	2	1
2	ด้านการติดต่อกับผู้ใช้งาน					
	2.1 การใช้งานง่ายและเป็นมิตรกับผู้ใช้ เช่น การจัดลำดับข้อมูลบนหน้าจอ					
	2.2 ความเหมาะสมในการวางตำแหน่งองค์ประกอบบนหน้าจอ					
	2.3 ความชัดเจนในการสื่อความหมายของส่วนต่างๆ					
	2.4 ในกรณีที่มีข้อผิดพลาดมีการแสดงให้เห็นอย่างชัดเจน					
	2.5 มีส่วนช่วยเหลือผู้ใช้ในการใช้งานระบบ					
	2.6 ความสวยงามของการแสดงผลบนหน้าจอ เช่น การใช้สีหรือขนาดของตัวอักษร					

ความคิดเห็นและข้อเสนอแนะ

.....

.....

.....

.....

.....

.....

.....

ข้อ	รายการประเมินผล	ระดับความพึงพอใจ				
		5	4	3	2	1
3	ด้านคุณค่าของระบบ					
	3.1 ช่วยลดเวลาในการสร้างกรณีทดสอบ ซึ่งจะเป็นร่างกรณีทดสอบเบื้องต้น					
	3.2 สามารถนำไปเป็นระบบสนับสนุนสำหรับการตรวจรับซอฟต์แวร์ โดยเฉพาะผู้ใช้ที่ขาดความรู้ในการสร้างกรณีทดสอบ					

ความคิดเห็นและข้อเสนอแนะ

.....

.....

.....

.....

.....

.....

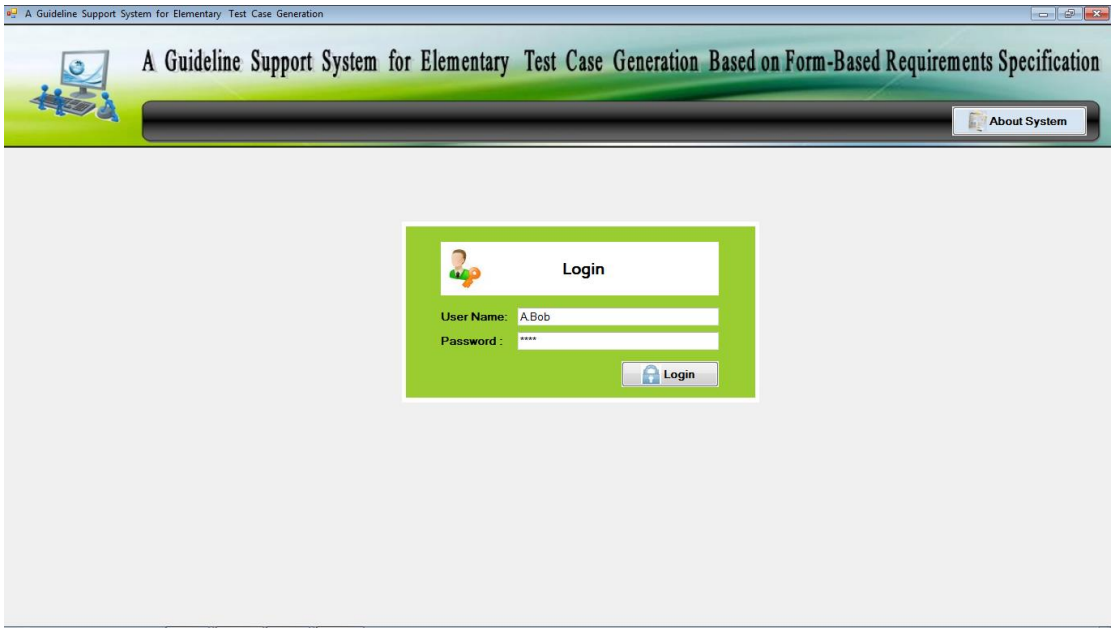
ขอบคุณที่เสียสละเวลาในการตอบแบบประเมิน

ภาคผนวก ฉ.

คู่มือการใช้งานระบบ

1. การเข้าใช้งานระบบ

ผู้ใช้ทุกกลุ่มจะต้องใช้ชื่อผู้ใช้และรหัสผ่านในการยืนยันตัวตนเพื่อเข้าสู่การใช้งานระบบในส่วนต่างๆ ที่ตนเกี่ยวข้อง แสดงดังภาพประกอบ ฉ-1

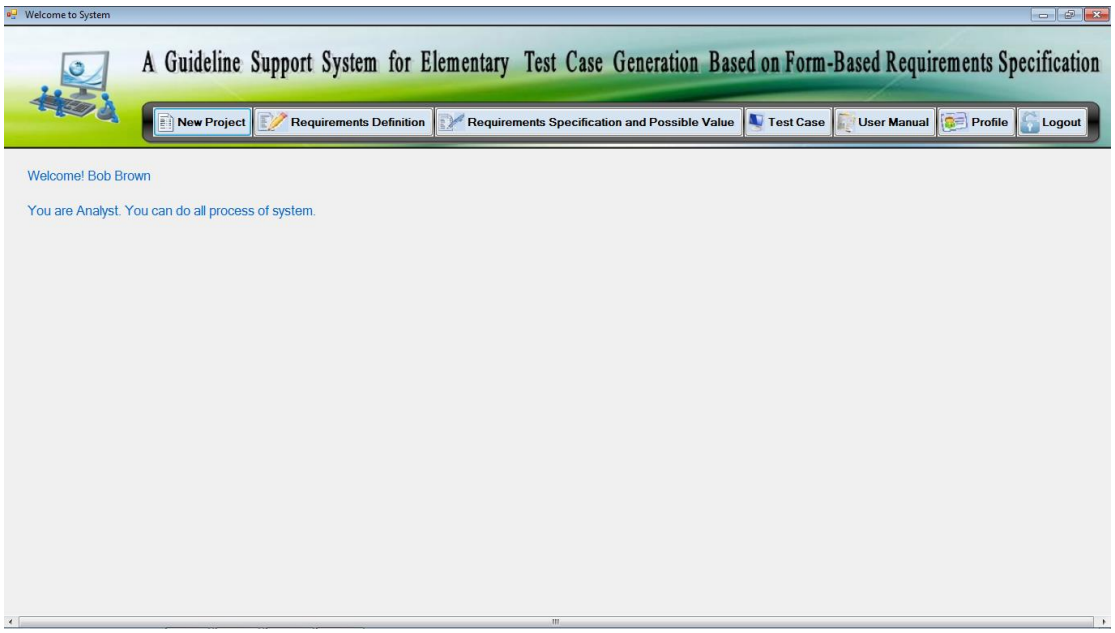


ภาพประกอบ ฉ-1 หน้าจอการเข้าใช้งานระบบ

2. เมนูหลักของระบบ

เมนูหลักของระบบจะประกอบด้วย 7 ส่วนหลักๆ คือ (1) เมนู “New Project” เพื่อสร้างโครงงานใหม่ (2) เมนู “Requirements Definition” เพื่อนิยามความต้องการ (3) เมนู “Requirements Specification and Possible Value” เพื่อกำหนดคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม จากนั้นก็ระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า แล้วกำหนดผลลัพธ์ที่คาดหวังของแต่ละกรณีในตารางตัดสินใจตามลำดับ (4) เมนู “Test Case” เพื่อเลือกแสดงกรณีทดสอบของโครงงานที่ต้องการ และ (5) เมนู “User Manual” เป็นส่วนของการ

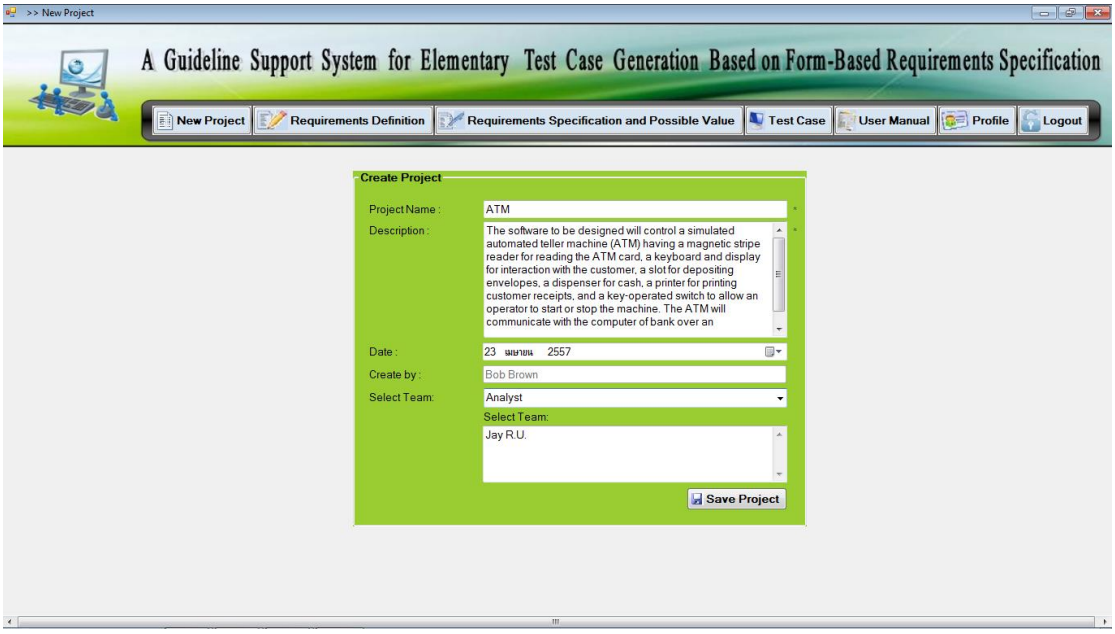
อธิบายการทำงานของระบบและขั้นตอนต่างๆ พร้อมทั้งตัวอย่างประกอบแยกตามกลุ่มผู้ใช้งาน (6) เมนู “Profile” เป็นส่วนของการแสดงข้อมูลผู้ใช้ ผู้ใช้สามารถปรับปรุงข้อมูลส่วนตัวได้ (7) เมนู “Logout” เป็นส่วนของการออกจากระบบเมื่อเสร็จสิ้นการทำงาน หน้าจอแรกของระบบแสดงดังภาพประกอบ ฉ-2 ทั้งนี้กลุ่มผู้ใช้ที่เป็นนักวิเคราะห์ระบบจะสามารถใช้งานเมนูต่างๆ ได้ทั้งหมด แต่กลุ่มผู้ใช้ที่เป็นผู้ใช้งานทั่วไปจะไม่สามารถใช้งานเมนู “New Project” และเมนู “Requirements Specification and Possible Value” เพราะส่วนนี้มีไว้สำหรับนักวิเคราะห์ระบบเท่านั้น



ภาพประกอบ ฉ-2 หน้าจอหลักของระบบ

3. การสร้างโครงงานใหม่

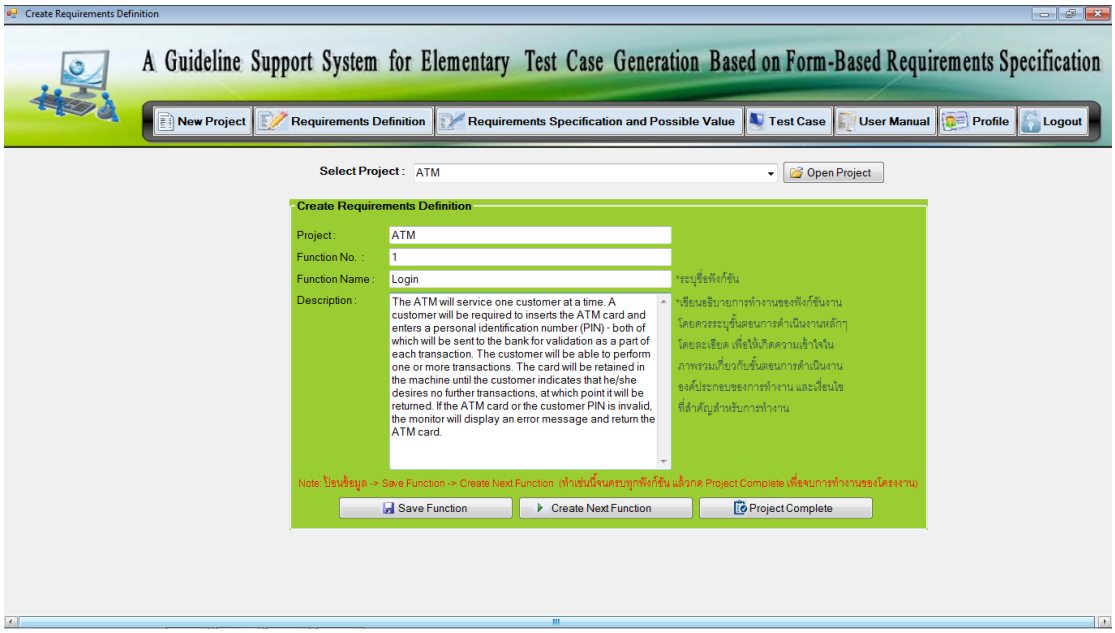
เมื่อต้องการสร้างโครงงานใหม่ นักวิเคราะห์ระบบจะเลือกเมนู “New Project” จากนั้นทำการกรอกข้อมูลให้ครบถ้วน แล้วกดปุ่ม “Create Project” เพื่อบันทึกโครงงาน ดังภาพประกอบ ฉ-3



ภาพประกอบ ฉ-3 หน้าจอสำหรับสร้างโครงงานใหม่โดยนักวิเคราะห์ระบบ

4. การนิยามความต้องการ

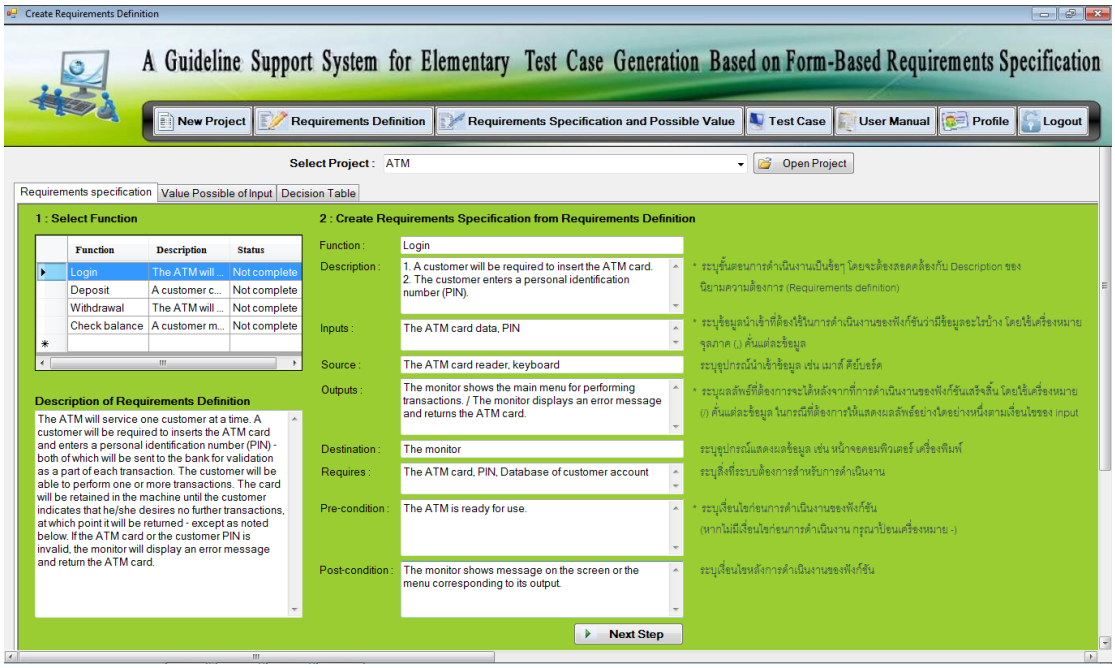
ในส่วนของการนิยามความต้องการสามารถเข้าใช้งานได้ทั้งผู้ใช้ทั่วไปและนักวิเคราะห์ระบบ โดยเลือกที่เมนู “Requirements Definition” จากนั้นทำการเลือกโครงงานที่ต้องการ แล้วเลือกปุ่ม “Open project” เมื่อกรอกข้อมูลครบถ้วนแล้วให้กดปุ่ม “Save function” จากนั้นกดปุ่ม “Create next function” เพื่อนิยามความต้องการให้กับฟังก์ชันงานต่อไป ทำอย่างนี้ไปเรื่อยๆ จนกว่าจะกรอกข้อมูลครบทุกฟังก์ชันในโครงงาน เมื่อกรอกข้อมูลครบทุกฟังก์ชันแล้ว จึงกดปุ่ม “Project complete” เพื่อจบการทำงานในส่วนนี้ ดังภาพประกอบ ฉ-4



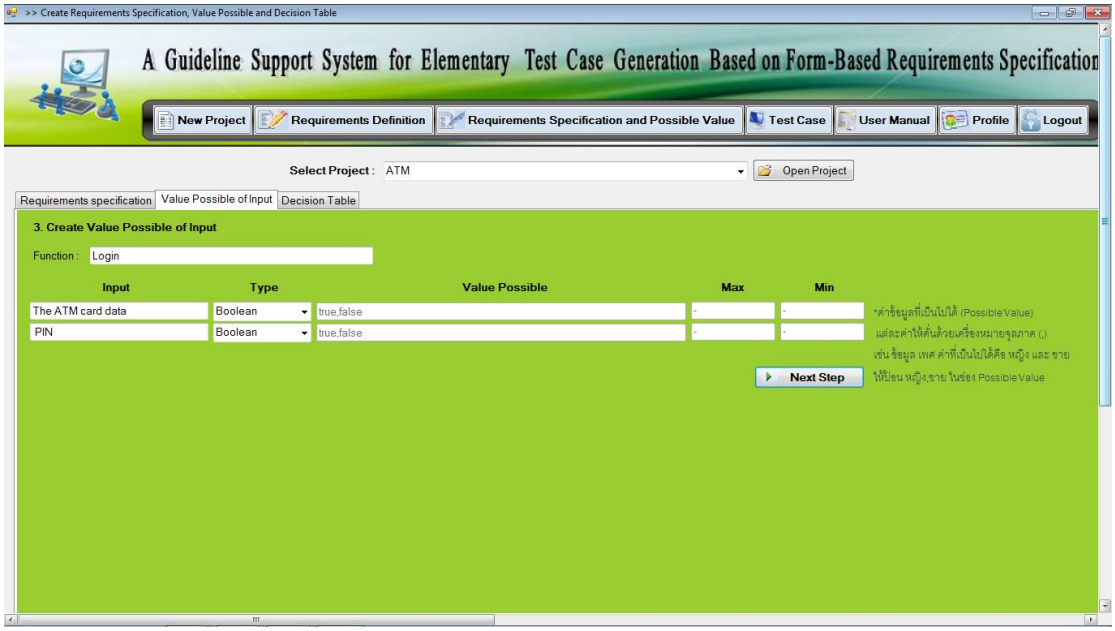
ภาพประกอบ ฉ-4 หน้าจอสำหรับสร้างนิยามความต้องการโดยผู้ใช้ทั่วไป หรือนักวิเคราะห์ระบบ

5. การกำหนดคุณลักษณะความต้องการ ระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า และกำหนดผลลัพธ์ที่คาดหวังในตารางตัดสินใจ

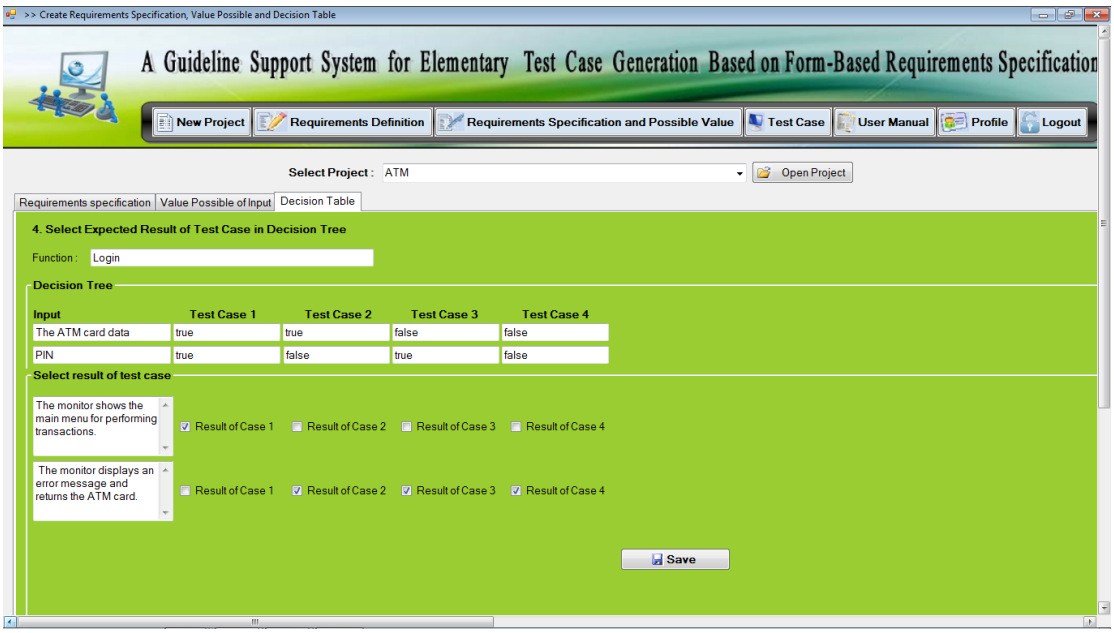
ขั้นตอนนี้ถือเป็นส่วนสำคัญอย่างยิ่งที่จะได้มาซึ่งกรณีทดสอบที่มีประสิทธิภาพ จำเป็นต้องใช้นักวิเคราะห์ระบบดำเนินการในส่วนนี้เท่านั้น โดยการเลือกเมนู “Requirements Specification and Possible Value” จากนั้นให้ทำการเลือกโครงงานและฟังก์ชันงานที่ต้องการระบุ โดยเริ่มจากการกำหนดคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม ดังภาพประกอบ ฉ-5 จากนั้นระบุชนิดและค่าข้อมูลที่เป็นไปได้ของข้อมูลนำเข้า ดังภาพประกอบ ฉ-6 แล้วกำหนดผลลัพธ์ที่คาดหวังของแต่ละกรณีในตารางตัดสินใจดังภาพประกอบ ฉ-7



ภาพประกอบ ฉ-5 หน้าจอสำหรับสร้างคุณลักษณะความต้องการที่เขียนอยู่ในรูปแบบฟอร์ม



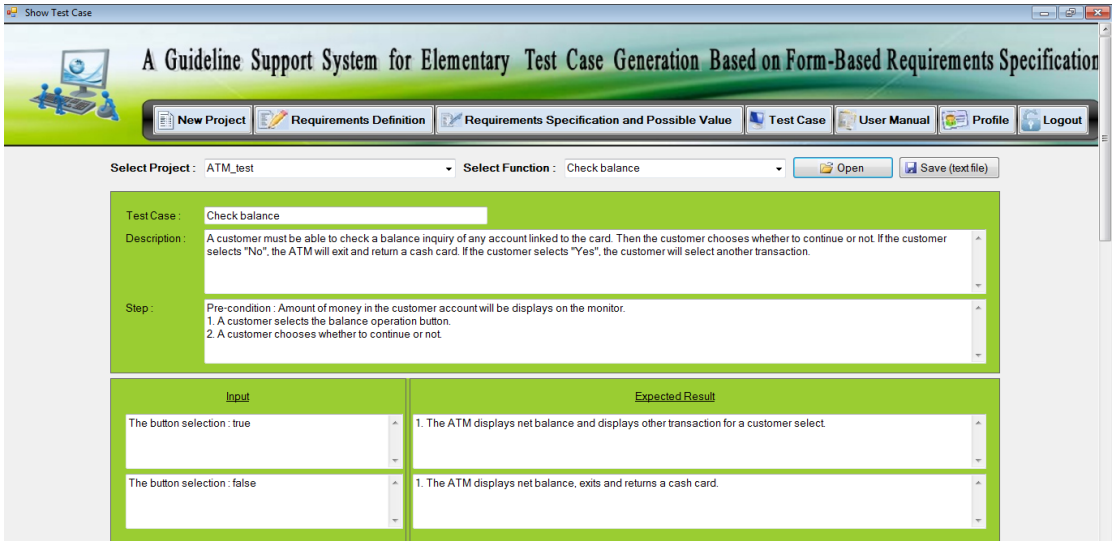
ภาพประกอบ ฉ-6 หน้าจอสำหรับระบุชนิดข้อมูลและค่าที่เป็นไปได้ของข้อมูลนำเข้า



ภาพประกอบ ฉ-7 หน้าจอสำหรับระบุผลลัพธ์ที่คาดหวังในตารางตัดสินใจ

6. การแสดงกรณีทดสอบของโครงงานที่ต้องการ

ผู้ใช้ทั้งสองกลุ่มสามารถเลือกดูกรณีทดสอบที่ต้องการได้โดยกดเลือกเมนู “Test Case” จากนั้นเลือกโครงงานและฟังก์ชันงานที่ต้องการแสดงกรณีทดสอบ ดังภาพประกอบ ฉ-8 และสามารถบันทึกข้อมูลกรณีทดสอบของฟังก์ชันงานที่เลือกในรูป Text file ได้โดยการกดปุ่ม “Save (text file)” แสดงตัวอย่างแฟ้มข้อมูลที่ได้ดังภาพประกอบ ฉ-9



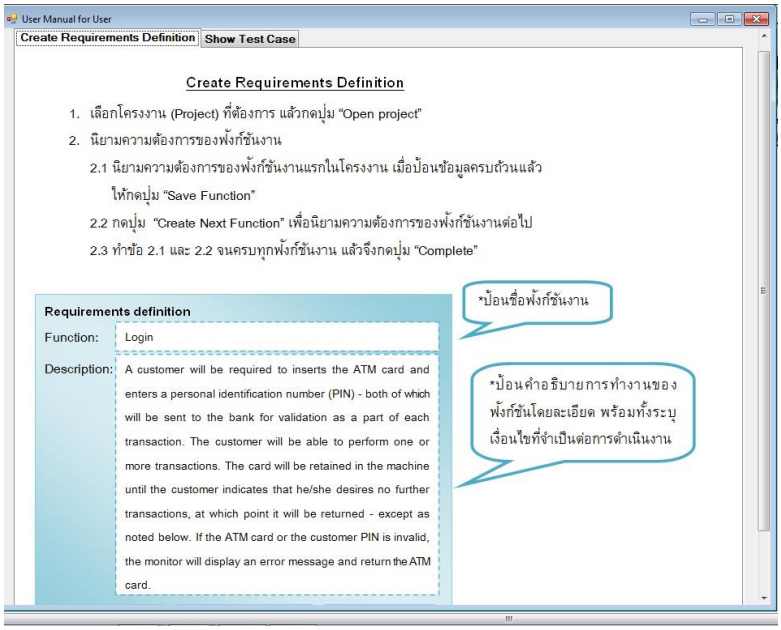
ภาพประกอบ ฉ-8 หน้าจอสำหรับแสดงกรณีทดสอบที่สร้างโดยระบบ

```
================================ Test Case ================================

Test Case:
Deposit

Description:
A customer chooses the deposit button, then payment and confirm. The ATM will
check the balance and display to the customer. The ATM will accept a deposit
from 100 Baht to 100,000 Baht. When the customer presses the deposit
confirmation. If condition of deposit is true, so the ATM will displays the
total deposit and net balance. Then the ATM prints a receipt and returns a cash
card. But it is not conditions then the ATM displays a warning message and
returns the cash card.

Step:
Pre-condition : The monitor displays transactions for a customer select.
1. A customer chooses deposit. Then the payment and confirm.
2. The ATM will check the balance and display to a customer.
3. If condition of deposit is true, so the ATM will displays the total deposit
and net balance in the customer account. Then the ATM prints a receipt and
return cash card.
4. But it is not conditions, the ATM displays a warning message and return cash
card.

--------------------------------------------------------------------------------
#Test case 1
Input:
deposit : 50
Expected Result:
1. The ATM displays a warning message.
2. Returns a cash card.

--------------------------------------------------------------------------------
#Test case 2
Input:
deposit : 100
Expected Result:
1. The ATM displays the total deposit and net balance in the customer account.
2. Prints a receipt..
3. Returns a cash card.

--------------------------------------------------------------------------------
```

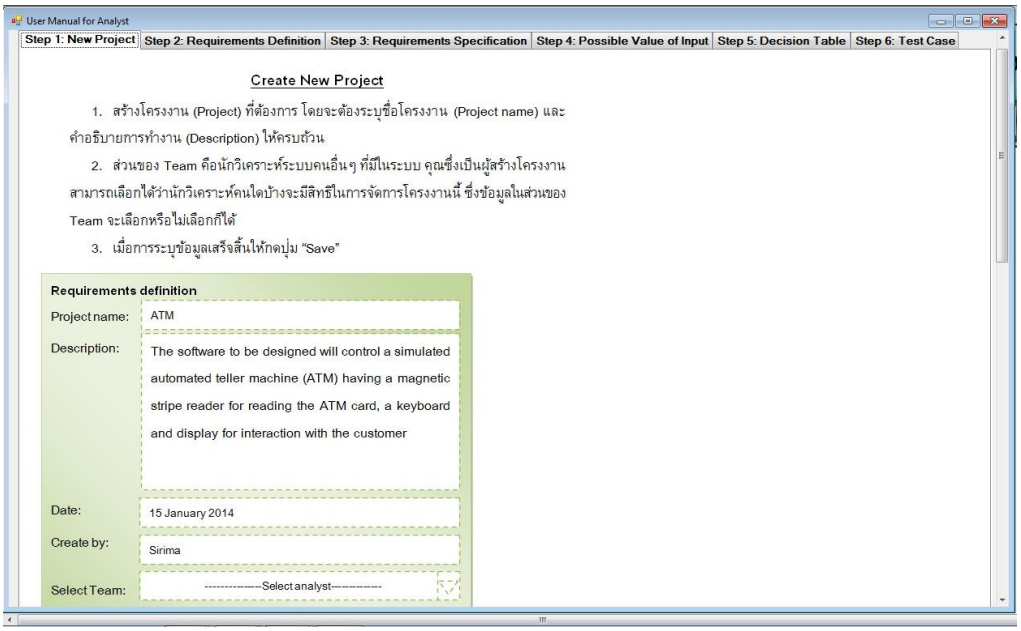
ภาพประกอบ ฉ-9 ตัวอย่างแฟ้มข้อมูลของกรณีทดสอบในรูป Text file

7. การแสดงส่วนช่วยเหลือสำหรับผู้ใช้

ผู้ใช้ทั้งสองกลุ่มสามารถศึกษาวิธีการใช้งานระบบ คำอธิบายขั้นตอนการทำงานต่างๆ ของระบบ พร้อมตัวอย่างประกอบ โดยการกดเลือกเมนู “User Manual” จากนั้นระบบจะแสดงคู่มือการใช้งานระบบให้ผู้ใช้ได้ทำการศึกษาตามกลุ่มของผู้ใช้ ได้แก่ คู่มือการใช้งานระบบสำหรับผู้ใช้ทั่วไป ดังตัวอย่างในภาพประกอบ ฉ-10 และคู่มือการใช้งานระบบสำหรับนักวิเคราะห์ระบบ ดังตัวอย่างในภาพประกอบ ฉ-11



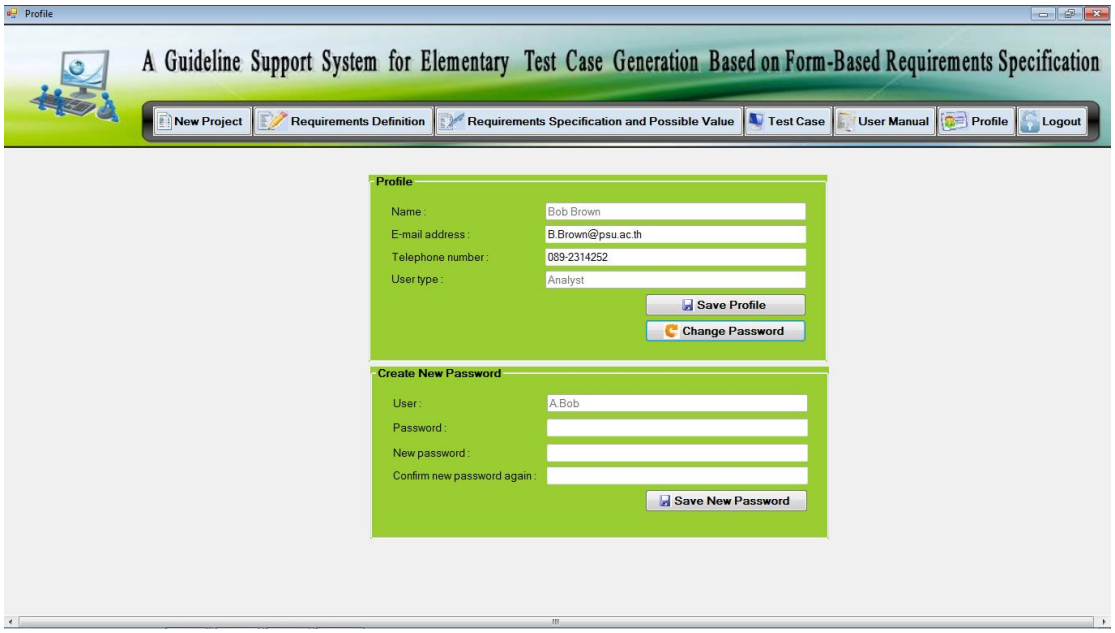
ภาพประกอบ ฉ-10 หน้าจอสำหรับแสดงคู่มือการใช้งานระบบสำหรับผู้ใช้ทั่วไป



ภาพประกอบ ฉ-11 หน้าจอสำหรับแสดงคู่มือการใช้งานระบบสำหรับนักวิเคราะห์ระบบ

8. การแก้ไขข้อมูลส่วนตัว

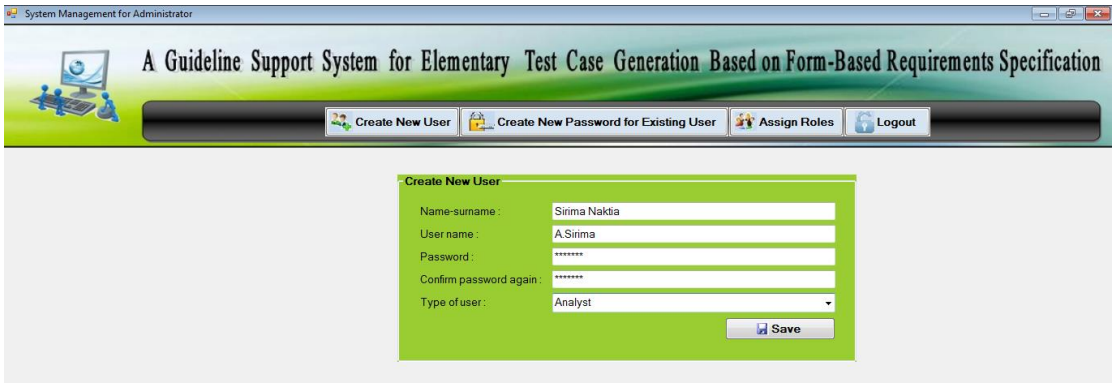
ผู้ใช้ทั่วไปและนักวิเคราะห์ระบบสามารถแก้ไขหรือเพิ่มเติมข้อมูลส่วนตัวได้ ดังตัวอย่างในภาพประกอบ ฉ-12



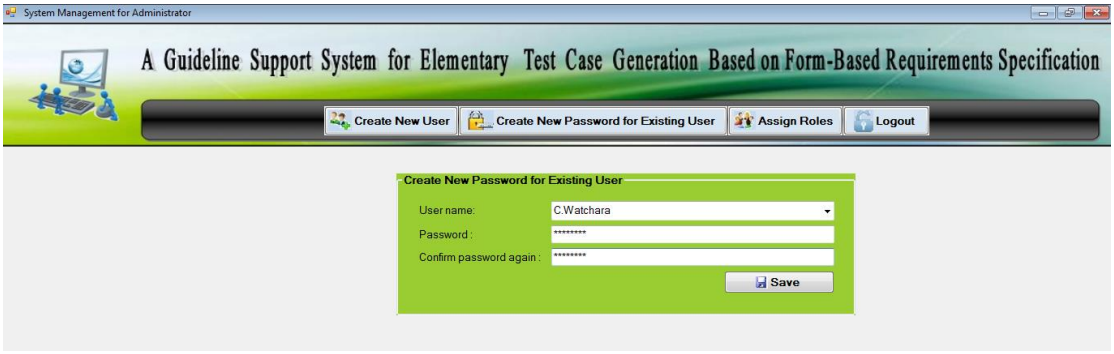
ภาพประกอบ ฉ-12 หน้าจอสำหรับการแก้ไขข้อมูลส่วนตัวผู้ใช้

9. การจัดการข้อมูลผู้ใช้สำหรับผู้ดูแลระบบ

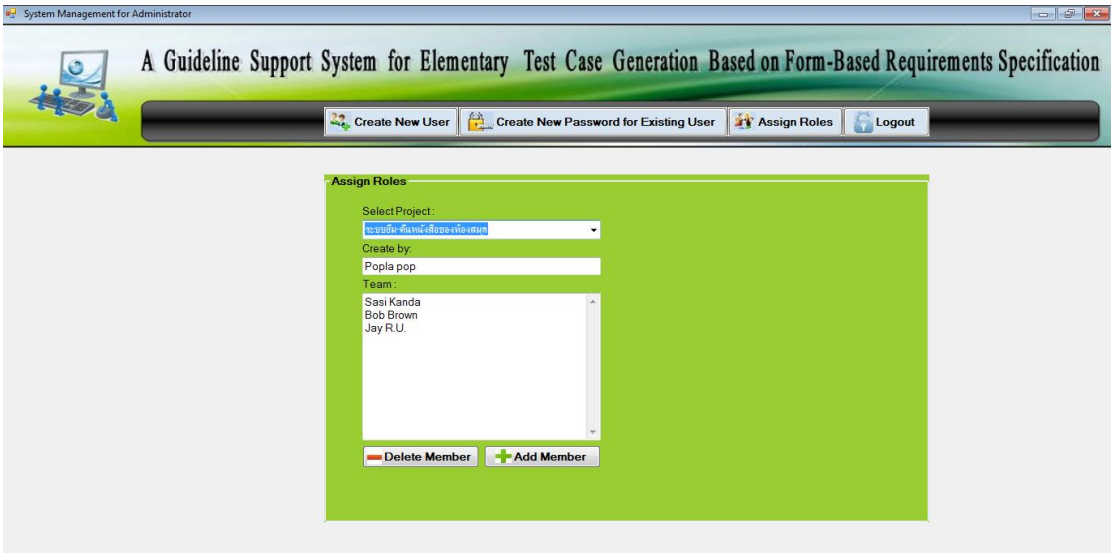
ในระบบที่พัฒนาขึ้นนี้ผู้ดูแลระบบจะมีหน้าที่ในการจัดการข้อมูลหลัก 3 ส่วน คือ การสร้างบัญชีผู้ใช้และข้อมูลผู้ใช้เบื้องต้นให้กับผู้ใช้รายใหม่ โดยการเลือกเมนู “Create New User” ดังตัวอย่างในภาพประกอบ ฉ-13 การสร้างรหัสผ่านใหม่ให้กับผู้ใช้รายเดิมในกรณีที่ผู้ใช้ลืมรหัสผ่าน โดยการเลือกเมนู “Create New Password for Existing User” ดังตัวอย่างในภาพประกอบ ฉ-14 และการแก้ไขรายการนักวิเคราะห์ระบบที่สามารถเข้าดำเนินการกับแต่ละโครงงาน โดยการเลือกเมนู “Assign Roles” ดังตัวอย่างในภาพประกอบ ฉ-15



ภาพประกอบ ฉ-13 หน้าจอสำหรับการสร้างบัญชีผู้ใช้และข้อมูลผู้ใช้เบื้องต้นให้กับผู้ใช้รายใหม่



ภาพประกอบ ฉ-14 หน้าจอสำหรับการสร้างรหัสผ่านใหม่ให้กับผู้ใช้รายเดิม ในกรณีที่ผู้ใช้ลืมรหัสผ่าน



ภาพประกอบ ฉ-15 หน้าจอสำหรับการแก้ไขรายการนักวิเคราะห์ระบบ ที่สามารถเข้าดำเนินการกับแต่ละโครงงาน