

เครื่องมือทดสอบซอฟต์แวร์แบบกึ่งอัตโนมัติด้วยยูนิเวอร์แซลเทสไดรเวอร์

Semi-Automated Software Testing Tool

Using Universal Test Driver Framework



วัน เดือน ปี.....	08 ก.พ. 2550
เลขทะเบียน.....	02205
เลขเรียกหนังสือ.....	วทศ.ศ/๒๕๔๗ '๕๕๔๗
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2547

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	เครื่องมือทดสอบซอฟต์แวร์แบบกึ่งอัตโนมัติด้วยยูนิเวอร์แซล เทสต์ไครเวอร์
นักศึกษา	นางสาวจุพภา ชูอำไพ
อาจารย์ที่ปรึกษา	ดร. พรฤดี เนติโสภาคกุล
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2547

บทคัดย่อ

การทดสอบซอฟต์แวร์แบบอัตโนมัติเป็นแนวทางหนึ่งที่ทำให้ผู้ทดสอบซอฟต์แวร์สามารถทำงานได้ครอบคลุมและรวดเร็วยิ่งขึ้น โดยผู้ทดสอบจะมีความเกี่ยวข้องกับกระบวนการทดสอบมากเท่าที่จำเป็นเท่านั้น ดังนั้นจึงได้พัฒนาเครื่องมือทดสอบซอฟต์แวร์แบบกึ่งอัตโนมัติขึ้นเป็นเครื่องมือนำร่อง เพื่อใช้เป็นแนวทางการพัฒนาเครื่องทดสอบซอฟต์แวร์แบบอัตโนมัติในอนาคตต่อไป เครื่องมือทดสอบซอฟต์แวร์แบบกึ่งอัตโนมัตินี้จะสร้างจากพื้นฐานหลักการทำงานของยูนิเวอร์แซลเทสต์ไครเวอร์ที่มีการออกแบบไว้เพื่อสนับสนุนการทดสอบในรูปแบบอัตโนมัติ โดยในหลักการทำงานข้างต้นนั้นจะทำการทดสอบซอฟต์แวร์ด้วยเอกสารสคริปต์เป็นหลัก ซึ่งเอกสารสคริปต์นี้ได้มาจากการแปลงเอกสารรูปแบบภาษาเอกซ์เอ็มแอล (eXtensible Markup Language) ที่เก็บข้อมูลของการทดสอบให้อยู่ในรูปแบบที่เหมาะสมต่อการนำมาใช้ทดสอบซอฟต์แวร์ เครื่องมือที่พัฒนาขึ้นนี้อาศัยแนวคิดเชิงวัตถุในการวิเคราะห์และออกแบบระบบซึ่งใช้ UML (Unified Modeling Language) สำหรับอธิบายรายละเอียดต่าง ๆ ในขั้นตอนการออกแบบ และเครื่องมือที่พัฒนานี้จะทำงานในรูปแบบของเว็บที่พัฒนาด้วย Java โดยมีฐานข้อมูลกลางสำหรับเก็บข้อมูลที่เกี่ยวข้องในการทดสอบต่าง ๆ

Title	Semi-Automated Software Testing Tool Using Universal Test Driver Framework
Student	Ms. Julapa Chu-Ampai
Advisor	Dr. Ponrudee Netisopakul
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2004

ABSTRACT

Automated software testing is a method that can make testers work quickly and conveniently. Tester concerns with only the process that needed. Thus, semi-automated software testing tool would be developed for serving the tester requirement, increasing test efficiency and improving to an automated software testing.

The semi-automated software testing tool is designed from the process of Universal Test Driver which supports the automate software testing. It will test software by script. The script is translated from the XML document which has the form of testing information. This tool is developed by using the object-oriented concept. The system uses UML for describe the details design. The developing tools have the processes in the form of Java website and center database for testing information.

กิตติกรรมประกาศ

โครงการนี้สำเร็จล่วงได้ด้วยการได้รับความช่วยเหลือและความกรุณาจากบุคคลต่างๆ
ดังนี้

1. ขอขอบพระคุณบิดา มารดา ที่ให้โอกาสในการศึกษาเล่าเรียนอย่างเต็มที่ รวมทั้งคอยให้กำลังใจ ช่วยเหลือ และให้คำปรึกษาต่างๆ
2. ขอขอบพระคุณ ดร.พรฤดี เนติโสภากุล อาจารย์ที่ปรึกษาเป็นอย่างมาก ที่กรุณาให้คำปรึกษา คำแนะนำ และแก้ไขในสิ่งที่บกพร่องในการพัฒนาโครงการนี้
3. ขอขอบพระคุณ คณาจารย์ทุกท่านที่ให้ความรู้ต่างๆ เพื่อนำความรู้มาใช้ในการพัฒนาโครงการนี้
4. ขอขอบพระคุณเจ้าหน้าที่ของคณะเทคโนโลยีสารสนเทศทุกคนที่ให้ความช่วยเหลือในด้านต่างๆ เป็นอย่างดี
5. ขอขอบคุณเพื่อนๆ และญาติๆ ที่ให้กำลังใจและช่วยเหลือในการพัฒนาโครงการนี้

นางสาว จุฬภา ชูอำไพ

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	X
บทที่	
1 บทนำ	1
1.1 ความเป็นมาของปัญหาและความสำคัญของโครงการพัฒนาระบบงาน	1
1.2 แนวคิดที่ใช้ในการพัฒนาระบบ	2
1.3 วัตถุประสงค์ของการพัฒนาระบบ	2
1.4 สมมติฐานของการพัฒนาระบบ	3
1.5 ขอบเขตของการพัฒนาระบบ	3
1.6 ขั้นตอนของพัฒนาระบบ	4
1.7 ผลที่คาดว่าจะได้รับ	4
2 ทฤษฎีที่เกี่ยวข้อง	5
2.1 การทดสอบซอฟต์แวร์	5
2.2 หลักการทำงานของยูนิเวอร์แซลเทสต์โคเวอรัจ	10
2.3 หลักการวิเคราะห์และออกแบบเชิงวัตถุด้วย UML	21
3 การศึกษาและวิเคราะห์ระบบ	32
3.1 การศึกษาหลักการทำงานของยูนิเวอร์แซลเทสต์โคเวอรัจ	32
3.2 ยูสเคสไดอะแกรม	35
4 การออกแบบระบบงาน	60
4.1 การออกแบบฐานข้อมูล (Entity Relationship Diagrams)	60
4.2 การออกแบบอินเตอร์แอกชันไดอะแกรม (Interaction Diagrams)	78

สารบัญ (ต่อ)

เรื่อง	หน้า
4.3 การออกแบบคลาสและแพ็คเกจของระบบ (Designing Classes and Package).....	98
4.4 การวิเคราะห์กิจกรรมของระบบ (Activity Analysis)	104
5 การพัฒนาระบบงาน	114
5.1 เครื่องมือที่ใช้ในการพัฒนาระบบ.....	114
5.2 การทำงานของระบบ (หน้าจอ)	118
5.3 การติดตั้งระบบ (System Deployment).....	126
6 บทสรุปและข้อเสนอแนะ	127
6.1 สรุปการพัฒนาระบบ.....	127
6.2 ข้อเสนอแนะ.....	128
บรรณานุกรม.....	130
ประวัติผู้เขียน.....	131
ภาคผนวก.....	132

สารบัญรูป

รูปที่	หน้า
2.1 แผนผังการทำงานของยูนิเวอร์แซลเทสต์ไครเวอร์	10
2.2 ตัวอย่างโครงสร้าง DTD.....	12
2.3 แสดงตัวอย่าง Use-case Diagram ในระบบงานห้องสมุด	22
2.4 ตัวอย่างคลาสเครื่องบิน	23
2.5 ตัวอย่างความสัมพันธ์แบบ Association.....	24
2.6 ตัวอย่างความสัมพันธ์แบบ Aggregation.....	24
2.7 ตัวอย่างความสัมพันธ์แบบ Composition	24
2.8 ตัวอย่างความสัมพันธ์แบบ Generalization	25
2.9 ตัวอย่างชีเค้นซ์ไคอะแกรม	25
2.10 ตัวอย่างคอลลาบอเรชั่นไคอะแกรม	26
2.11 ตัวอย่างสเตซาร์คไคอะแกรม.....	27
2.12 ตัวอย่างแอกทิวตี้ไคอะแกรม.....	27
2.13 ตัวอย่างคอมโพเนนต์ไคอะแกรม.....	28
2.14 ตัวอย่างดีพลอยเม้นต์ไคอะแกรม.....	29
3.1 แผนผังแสดงภาพรวมของการทำงานของระบบในมุมมองของการไหลของข้อมูล (Context Diagram).....	33
3.2 แอกทิวตี้ไคอะแกรมแสดงการทำงานของระบบที่พัฒนา	34
3.3 ยูสเคสไคอะแกรมหลักของระบบที่พัฒนา.....	37
3.4 ยูสเคสไคอะแกรมย่อยเกี่ยวกับการจัดการข้อมูลผู้ใช้ระบบ (Manage User Profile).....	38
3.5 ยูสเคสไคอะแกรมย่อยเกี่ยวกับการจัดการข้อมูลสำหรับการทดสอบ (Manage Test Info).....	38
3.6 ยูสเคสไคอะแกรมย่อยเกี่ยวกับการจัดการข้อมูลรายงานผลการทดสอบที่ได้จากการทดสอบด้วย สคริปต์ (Manage Test Results)	39
4.1 Entity Relationship Diagram สำหรับหลักการทำงานของยูนิเวอร์แซลเทสต์ไครเวอร์.....	62
4.2 Entity Relationship Diagram สำหรับระบบที่กำลังพัฒนามบนพื้นฐานหลักการทำงานของ แซลเทสต์ไครเวอร์.....	72
4.3 ชีเค้นซ์ไคอะแกรมของการเข้าใช้ระบบและตรวจสอบสิทธิ์ (Authenticate).....	78

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.4 ซีเควินชีไออะแกรมของการแสดงรายการข้อมูลสำหรับการทดสอบ (List Test Info)	79
4.5 ซีเควินชีไออะแกรมของการแสดงรายละเอียดของข้อมูลสำหรับการทดสอบ(View Test Info)	80
4.6 ซีเควินชีไออะแกรมของการสร้างข้อมูลสำหรับการทดสอบ (Create Test Info)	81
4.7 ซีเควินชีไออะแกรมของการปรับเปลี่ยนข้อมูลสำหรับการทดสอบ (Edit Test Info)	82
4.8 ซีเควินชีไออะแกรมของการลบข้อมูลสำหรับการทดสอบ (Delete Test Info)	82
4.9 ซีเควินชีไออะแกรมของการสร้างสคริปต์สำหรับการทดสอบ (Generate Test Script)	83
4.10 ซีเควินชีไออะแกรมของการแสดงรายการสคริปต์สำหรับการทดสอบ (List Test Script)	84
4.11 ซีเควินชีไออะแกรมของการแสดงรายละเอียดของสคริปต์สำหรับการทดสอบ (View Test Script)	85
4.12 ซีเควินชีไออะแกรมของการลบสคริปต์สำหรับการทดสอบ (Delete Test Scripts)	86
4.13 ซีเควินชีไออะแกรมของการเรียกสคริปต์ทำการทดสอบ โปรแกรม (Execute Test Script).....	87
4.14 ซีเควินชีไออะแกรมของการแสดงรายการผลการทดสอบที่ได้จากการทดสอบด้วยสคริปต์ (List Test Results)	88
4.15 ซีเควินชีไออะแกรมของการแสดงรายละเอียดของผลการทดสอบที่ได้จากการทดสอบด้วย สคริปต์ (View Test Result)	89
4.16 ซีเควินชีไออะแกรมของการสร้างผลการทดสอบ (Create Test Results)	90
4.17 ซีเควินชีไออะแกรมของการลบผลการทดสอบ (Delete Test Results)	91
4.18 ซีเควินชีไออะแกรมของการสร้างข้อมูลผู้ใช้ระบบ (Create User Profile)	92
4.19 ซีเควินชีไออะแกรมของการแก้ไขข้อมูลผู้ใช้ระบบ (Edit User Profile)	92
4.20 ซีเควินชีไออะแกรมของการลบข้อมูลผู้ใช้ระบบ (Delete User Profile)	93
4.21 ซีเควินชีไออะแกรมของการแสดงรายการผู้ใช้ระบบ (List User Profile)	94
4.22 ซีเควินชีไออะแกรมของการแสดงรายละเอียดข้อมูลผู้ใช้ระบบ (View User Profile).....	94
4.23 ซีเควินชีไออะแกรมของการเชื่อมต่อกับฐานข้อมูลในการกระจายข้อมูลจากเอกสาร เอกซ์เอ็มแอล (Interface XML with Repository-Extracting)	95
4.24 ซีเควินชีไออะแกรมของการเชื่อมต่อกับฐานข้อมูลเพื่อสร้างเอกสารเอกซ์เอ็มแอล (Interface XML with Repository-Getting)	96

สารบัญรูป (ต่อ)

รูปที่

หน้า

4.25 ซีเคັນซ์ไคอะแกรมของการประมวลผล Style Sheet เพื่อสร้างเอกสารเอกซ์เอ็มแอล (Style Sheet Process-HTML)	96
4.26 ซีเคັນซ์ไคอะแกรมของการประมวลผล Style Sheet เพื่อสร้างเอกสารสคริปต์ (Style Sheet Process-Test Script)	97
4.27 ซีเคັນซ์ไคอะแกรมของการแปลงสคริปต์เพื่อใช้เรียกประมวลผล (Script Transformation)..	98
4.28 คลาสไคอะแกรมที่เกี่ยวกับหน้าที่การทำงานของระบบ.....	99
4.29 คลาสไคอะแกรมที่เกี่ยวกับการเชื่อมต่อฐานข้อมูล (1)	100
4.30 คลาสไคอะแกรมที่เกี่ยวกับการเชื่อมต่อฐานข้อมูล (2)	101
4.31 คลาสไคอะแกรมแสดงจาวาบี๋นของระบบ.....	102
4.32 คลาสไคอะแกรมเกี่ยวกับเซิร์ฟเลต	103
4.33 แผนภาพแสดงแพ็คเกจของระบบ.....	104
4.34 แอคทิวิตี้ไคอะแกรมแสดงการสร้างและแก้ไขข้อมูลการทดสอบ.....	105
4.35 คลาสไคอะแกรมแสดงคลาสที่เกี่ยวกับการสร้างและแก้ไขข้อมูลการทดสอบ	105
4.36 แอคทิวิตี้ไคอะแกรมแสดงการลบข้อมูลการทดสอบ	106
4.37 คลาสไคอะแกรมแสดงคลาสที่เกี่ยวกับการลบข้อมูลการทดสอบ	106
4.38 แอคทิวิตี้ไคอะแกรมแสดงการค้นรายการข้อมูลการทดสอบ.....	107
4.39 คลาสไคอะแกรมแสดงคลาสที่เกี่ยวกับการค้นรายการข้อมูลการทดสอบ	107
4.40 แอคทิวิตี้ไคอะแกรมแสดงการค้นรายละเอียดข้อมูลการทดสอบ	107
4.41 คลาสไคอะแกรมแสดงคลาสที่เกี่ยวกับการค้นรายละเอียดข้อมูลการทดสอบ	107
4.42 แอคทิวิตี้ไคอะแกรมแสดงการนำข้อมูลการทดสอบในฐานะข้อมูลมาสร้างเป็นเอกสาร เอกซ์เอ็มแอล.....	108
4.43 แอคทิวิตี้ไคอะแกรมแสดงการนำข้อมูลการทดสอบในเอกสารเอกซ์เอ็มแอลมากระจาย	109
4.44 คลาสไคอะแกรมแสดงคลาสที่เกี่ยวกับการเชื่อมต่อเอกสารเอกซ์เอ็มแอลและฐานข้อมูล	109
4.45 แอคทิวิตี้ไคอะแกรมแสดงการสร้างเอกสาร สคริปต์สำหรับการทดสอบ.....	110
4.46 คลาสไคอะแกรมแสดงคลาสที่เกี่ยวกับการสร้างเอกสาร สคริปต์สำหรับการทดสอบ	110

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.47 แอคทีวิตีไดอะแกรมแสดงการแปลงเอกสารด้วย Style Sheet.....	111
4.48 แอคทีวิตีไดอะแกรมแสดงการเรียกทดสอบ โปรแกรมตามเอกสารสคริปต์.....	112
4.49 คลาสไดอะแกรมแสดงคลาสที่เกี่ยวกับการเรียกทดสอบ โปรแกรมตามเอกสารสคริปต์	113
5.1 ส่วนการทำงานในรูปแบบการทำงานแบบ Struts	116
5.2 สถาปัตยกรรมการทำงานแบบ Struts.....	117
5.3 ซีคว็่นซ์ไดอะแกรมแสดงลำดับการทำงานของระบบที่ใช้แนวคิด Struts.....	117
5.4 หน้าจอการทำงานหลักของระบบ	119
5.5 หน้าจอแสดงรายการข้อมูลกรณีทดสอบ (Test).....	120
5.6 หน้าจอแสดงรายละเอียดของกรณีทดสอบ (Test Detail).....	121
5.7 หน้าจอแสดงรายการเอกสารสคริปต์ (Test Script List).....	122
5.8 หน้าจอแสดงรายละเอียดในเอกสารสคริปต์ (Test Script Detail)	123
5.9 หน้าจอการสร้างเอกสารสคริปต์ (Generate Test Script).....	124
5.10 หน้าจอการเรียกทำการทดสอบ โปรแกรม (Test Execution).....	125
5.11 หน้าจอการจัดการเกี่ยวกับผลการทดสอบ (Test Result List).....	126
5.12 ดีพลอยमेंต์ไดอะแกรมของระบบ	126

สารบัญตาราง

ตารางที่	หน้า
2.1 ตัวอย่างรายการ Utilities ในการสร้าง Script.....	17
4.1 แสดงตารางทั้งหมดบนฐานข้อมูลตามหลักการทำงานยูนิเวอร์แซลเทสต์ไดรเวอร์	60
4.2 ตารางกลุ่มการทดสอบ (UTD_USER).....	63
4.3 ตารางโครงการ (UTD_PROJECT).....	63
4.4 ตารางกลุ่มการทดสอบ (UTD_TESTLIST).....	63
4.5 ตารางกรณีการทดสอบ (UTD_TEST).....	64
4.6 ตารางความสัมพันธ์ของกลุ่มและกรณีการทดสอบ (UTD_TESTSET).....	65
4.7 ตารางขั้นตอนการทดสอบ (UTD_STEP).....	65
4.8 ตารางความสัมพันธ์ของขั้นตอนและกรณีการทดสอบ (UTD_STEPTEST).....	65
4.9 ตารางข้อมูลการทดสอบที่ได้กระทำไป (UTD_RUN).....	66
4.10 ตารางเอกสารสำหรับการทดสอบ (UTD_TESTSCRIPT).....	66
4.11 ตารางความสัมพันธ์ของเอกสารและกลุ่มการทดสอบ (UTD_SCRIPTTESTLIST).....	66
4.12 ตารางเอกสารผลการทดสอบ (UTD_XMLRESULT).....	67
4.13 ตารางคำสั่งที่ช่วยในการทดสอบ (UTD_UTILITIES).....	67
4.14 ตารางความสัมพันธ์ของกลุ่มคำสั่งที่ช่วยในการทดสอบและ Platform (UTD_CODEUTIL).67	
4.15 ตารางแสดงรายละเอียดของ Platform (UTD_PLATFORM).....	67
4.16 ตารางความสัมพันธ์ของ UTD_REPOSITORY, UTD_PROJECT, และ UTD_TESTSCRIPT (UTD_SCRIPTINFO).....	68
4.17 ตารางแสดงรายละเอียดของเอกสารสคริปต์ (UTD_SCRIPTFILE).....	68
4.18 ตารางความสัมพันธ์ของเอกสารสคริปต์ และ Folder สำหรับการจัดเก็บเอกสารสคริปต์ (UTD_SCRIPTFOLDER).....	69
4.19 ตารางแสดงรายละเอียดของ Folder (UTD_FOLDER).....	69
4.20 ตารางแสดงรายละเอียดของคลังข้อมูล (UTD_REPOSITORY).....	69
4.21 ตารางแสดงรายละเอียดของเครื่องที่ใช้ในการทดสอบ (UTD_MACHINES).....	70
4.22 แสดงตารางทั้งหมดบนฐานข้อมูลของระบบที่กำลังพัฒนา (KMTD).....	71

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
4.23 ตารางกลุ่มการทดสอบ (USER).....	73
4.24 ตารางโครงการ (PROJECT).....	73
4.25 ตารางกลุ่มการทดสอบ (TESTLIST).....	73
4.26 ตารางกรณีการทดสอบ (TEST).....	74
4.27 ตารางความสัมพันธ์ของกลุ่มและกรณีการทดสอบ (TESTSET_REL).....	74
4.28 ตารางขั้นตอนการทดสอบ (STEP).....	75
4.29 ตารางความสัมพันธ์ของขั้นตอนและกรณีการทดสอบ (STEPTEST_REL).....	75
4.30 ตารางข้อมูลการทดสอบที่ได้กระทำไป (RUN).....	75
4.31 ตารางเอกสารสำหรับการทดสอบ (TESTSCRIPT).....	76
4.32 ตารางความสัมพันธ์ของเอกสารและกลุ่มการทดสอบ (SCRIPT_REL).....	76
4.33 ตารางเอกสารผลการทดสอบ (XMLRESULT).....	77
4.34 ตารางคำสั่งที่ช่วยในการทดสอบ (UTILITIES).....	77

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหาและความสำคัญของโครงการพัฒนาระบบงาน

ขั้นตอนการทดสอบซอฟต์แวร์หรือส่วนชุดคำสั่งที่พัฒนาขึ้นจัดว่าเป็นขั้นตอนหนึ่งที่สำคัญ ซึ่งไม่ควรละเลยไม่ว่าจะใช้กระบวนการพัฒนาซอฟต์แวร์แบบใดก็ตาม แต่ในการที่จะทดสอบให้ครอบคลุมทุกเงื่อนไขหรือครอบคลุมทุกระบบปฏิบัติการที่สนับสนุนสำหรับซอฟต์แวร์หนึ่ง ๆ นั้น ผู้ทดสอบอาจต้องใช้เวลามากพอสมควรสำหรับการทดสอบแต่ละครั้ง อีกทั้งยังต้องทดสอบในเงื่อนไขเดิมซ้ำ ๆ กันจนกว่าเงื่อนไขนั้นจะผ่านหรือได้ผลที่สามารถยอมรับได้ ดังนั้นแนวคิดที่จะทำการทดสอบให้เป็นไปในแบบอัตโนมัติจึงเกิดขึ้นเพื่อให้ผู้ทดสอบสามารถทำงานได้อย่างมีประสิทธิภาพและสะดวกเร็วกว่าการทดสอบแบบธรรมดาทั่วไปเมื่อต้องการทดสอบเงื่อนไขเดิมซ้ำ ๆ กัน

ด้วยที่แนวคิดการทดสอบแบบอัตโนมัติได้รับความนิยมมากขึ้นในปัจจุบันซึ่งเป็นยุคที่อัตราการพัฒนาและผลิตซอฟต์แวร์เพิ่มขึ้นจึงได้มีการค้นคว้าวิจัยถึงหลักการการทำงานที่จะทำให้เกิดการทดสอบแบบอัตโนมัติขึ้น ดังเช่นทีมงานของ BMC Software Inc. ที่ได้คิดหลักการทำงานของยูนิเวอร์แซลเทสต์ไครเวอร์ซึ่งมีการทำงานตามแนวคิดของการทดสอบแบบอัตโนมัติขึ้น โดยที่หลักการนี้จัดได้ว่าเป็นแนวทางการทำการทดสอบที่ตั้งอยู่บนพื้นฐานของการใช้เอกสารสคริปต์ที่ได้จากเอกสารรูปแบบภาษาเอกซ์เอ็มแอล และสามารถประยุกต์ให้มีการตรวจสอบถึงขั้นตอนการพัฒนาของซอฟต์แวร์ที่กำลังสนใจเพื่อที่จะทำการทดสอบซอฟต์แวร์นั้นแบบอัตโนมัติได้อีกด้วย

โครงการพัฒนาเครื่องมือการทดสอบซอฟต์แวร์แบบกึ่งอัตโนมัติจึงได้นำเอาหลักการทำงานของยูนิเวอร์แซลเทสต์ไครเวอร์มาเป็นพื้นฐานของการออกแบบและพัฒนาระบบ โดยใช้นโยบายของการวิเคราะห์ออกแบบและพัฒนาระบบในเชิงวัตถุที่ได้รับการยอมรับในปัจจุบันซึ่งมีจุดเด่นในเรื่องของการนำกลับมาใช้ใหม่ นอกจากนั้นจะนำเอา UML ซึ่งเป็นภาษาเชิงรูปภาพและสัญลักษณ์มาใช้อธิบายระบบที่ออกแบบให้เข้าใจง่าย ซึ่งระบบที่พัฒนาขึ้นนี้เป็นกรณีศึกษาหนึ่งในด้านการพัฒนาเครื่องมือสำหรับการทดสอบซอฟต์แวร์ดังนั้นผลการพัฒนาโครงการจะเป็นประโยชน์ในการนำหลักการยูนิเวอร์แซลเทสต์ไครเวอร์นี้ไปประยุกต์ใช้จริงกับการทดสอบ

ซอฟต์แวร์สำหรับองค์กรผู้ผลิตซอฟต์แวร์ในอนาคต อีกทั้งยังเป็นแนวทางการศึกษาออกแบบเครื่องมือในการทดสอบซอฟต์แวร์ในรูปแบบอื่นต่อไป

1.2 แนวคิดที่ใช้ในการพัฒนาระบบ

แนวคิดหลักที่ใช้ในการพัฒนาระบบคือการนำเอาหลักการทำงานของยูนิเวอร์แซลเทสต์ไคโรเวอร์ที่ทำการทดสอบบนพื้นฐานของการใช้ออกสารสคริปต์จากเอกสารรูปแบบภาษาเอกซ์เอ็มแอล มาพัฒนาให้เกิดระบบที่ใช้ทดสอบได้จริงแบบกึ่งอัตโนมัติโดยที่ผู้ใช้งานมีส่วนเกี่ยวข้องกับระบบเพื่อเรียกทำการทดสอบซอฟต์แวร์ และด้วยที่เอกสารรูปแบบภาษาเอกซ์เอ็มแอล มีความยืดหยุ่นด้าน โครงสร้างของข้อมูลในเอกสารดังนั้นผู้ที่นำหลักการทำงานนี้ไปประยุกต์ใช้จึงสามารถปรับเปลี่ยนโครงสร้างเอกสารได้ตามต้องการเพื่อให้สอดคล้องกับการนำไปใช้มากที่สุด นอกจากนี้ตามหลักการทำงานของยูนิเวอร์แซลเทสต์ไคโรเวอร์แล้วจะมีการใช้ระบบฐานข้อมูลกลางในการเก็บรักษาข้อมูลที่เกี่ยวข้องกับการทดสอบเพื่อให้สามารถทำการทดสอบซ้ำ ๆ กันได้โดยไม่ต้องกังวลเกี่ยวกับผู้ทดสอบว่าต้องเป็นคนที่เคยทดสอบซอฟต์แวร์นั้นมาก่อนจึงจะเข้าใจความต้องการและเงื่อนไขที่ต้องทดสอบหรือกังวลในเรื่องการสูญหายของข้อมูลการทดสอบครั้งก่อนเป็นต้น สำหรับการขยายการทำงานเพื่อสนับสนุนการทดสอบแบบอัตโนมัติให้สมบูรณ์โดยที่ผู้ทำการทดสอบมีส่วนเกี่ยวข้องน้อยมากในการเรียกการทดสอบหนึ่ง ๆ นั้นจะต้องมีการเพิ่มส่วนของการตรวจสอบถึงการเกิดขึ้นใหม่หรือการเปลี่ยนแปลงในซอฟต์แวร์ที่อยู่ในขั้นตอนของการทดสอบเพื่อเรียกใช้สคริปต์ที่เกี่ยวข้องทำการทดสอบซอฟต์แวร์ที่กำลังทดสอบซอฟต์แวร์ดังกล่าว

ในการพัฒนาเครื่องมือการทดสอบซอฟต์แวร์แบบกึ่งอัตโนมัติตามโครงการนี้ผู้พัฒนาจะใช้หลักการของยูนิเวอร์แซลเทสต์ไคโรเวอร์เป็นพื้นฐานของการออกแบบและพัฒนาโดยจะเป็นการพัฒนาแบบบนาร่องที่ยังไม่มีส่วนขยายการทำงานที่สนับสนุนการทดสอบแบบอัตโนมัติอย่างสมบูรณ์ ซึ่งจะใช้การวิเคราะห์และออกแบบในเชิงวัตถุ (Object Oriented) และใช้ UML (Unified Model Language) ในการอธิบาย ส่วนของการพัฒนาเลือกใช้ภาษาจาวาโดยจะพัฒนาเครื่องมือที่มึการทำงานในรูปแบบของโปรแกรมประยุกต์บนเว็บและใช้ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เป็นฐานข้อมูลกลาง

1.3 วัตถุประสงค์ของการพัฒนาระบบ

- เพื่อให้ได้เครื่องมือทดสอบซอฟต์แวร์แบบกึ่งอัตโนมัติซึ่งช่วยให้การทดสอบซอฟต์แวร์เป็นไปได้อย่างรวดเร็วและมีประสิทธิภาพมากยิ่งขึ้น

- เพื่อศึกษาแนวทางการพัฒนาเครื่องมือทดสอบซอฟต์แวร์ที่มีการทำงานอยู่บนพื้นฐานของการใช้เอกสารสคริปต์จากเอกสารรูปแบบภาษาเอกซ์เอ็มแอล และการใช้ฐานข้อมูลกลางในการเก็บข้อมูลการทดสอบ

1.4 สมมติฐานของการพัฒนาระบบ

- ระบบที่พัฒนามบนพื้นฐานหลักการทำงานของยูนิเวอร์แซลเทสต์ไครเวอร์โดยสามารถรองรับผู้ใช้ 4 ประเภท ได้แก่
 - 1) ผู้ดูแลระบบ (System Admin) ซึ่งทำหน้าที่ดูแลเกี่ยวกับการเข้าใช้งานระบบ
 - 2) ผู้ควบคุมการทดสอบ (Tester) ซึ่งทำหน้าที่จัดเตรียมกรณีสำหรับการทดสอบโปรแกรม สร้างเอกสารสคริปต์สำหรับการทดสอบ และทำการทดสอบโปรแกรมที่ต้องการด้วยสคริปต์ที่สร้างไว้
 - 3) เจ้าหน้าที่ทดสอบซอฟต์แวร์ (Test Operator) ซึ่งทำหน้าที่ทดสอบโปรแกรมที่ต้องการด้วยสคริปต์ที่สร้างไว้แล้ว
- สำหรับการสร้างเอกสารรูปแบบภาษาเอกซ์เอ็มแอล สำหรับข้อมูลจากฐานข้อมูล และการบันทึกข้อมูลไปยังฐานข้อมูลด้วยเอกสารรูปแบบภาษาเอกซ์เอ็มแอล สามารถนำโมดูลที่มีในตลาดหรือมีอยู่ในองค์กรมาประยุกต์ใช้ได้ในขั้นตอนการพัฒนาระบบ
- ผลของการทดสอบซอฟต์แวร์ที่ได้จากเครื่องมือทดสอบซอฟต์แวร์จะสามารถแสดงผ่านเว็บไซต์และเก็บไว้ในรูปแบบไฟล์ได้

1.5 ขอบเขตของการพัฒนาระบบ

การพัฒนาเครื่องมือทดสอบซอฟต์แวร์นี้เป็นการนำเอาหลักการทำงานของยูนิเวอร์แซลเทสต์ไครเวอร์มาวิเคราะห์ ออกแบบและพัฒนา เพื่อนำมาเป็นเครื่องมือทดสอบซอฟต์แวร์แบบกึ่งอัตโนมัติที่พัฒนาด้วยภาษาจาวา โดยจะสามารถทดสอบซอฟต์แวร์ได้บนระบบปฏิบัติการวินโดวส์ ซึ่งการทำงานของเครื่องมือที่จะพัฒนาสามารถแบ่งการทำงานได้ดังต่อไปนี้

- 1) การจัดการเกี่ยวกับข้อมูลของการทดสอบในฐานข้อมูลกลาง
- 2) การเชื่อมต่อการอ่านข้อมูลระหว่างข้อมูลในเอกสารภาษาเอกซ์เอ็มแอลกับข้อมูลในฐานข้อมูลกลาง
- 3) การสร้างเอกสารสคริปต์สำหรับการทดสอบจากเอกสารภาษาเอกซ์เอ็มแอลที่เป็นข้อมูลสำหรับการทดสอบ
- 4) การเรียกกลุ่มโปรแกรมเพื่อทำการทดสอบด้วยเอกสารสคริปต์ที่ได้

- 5) การจัดการเกี่ยวกับเอกสารผลการทดสอบ
- 6) การจัดการเกี่ยวกับข้อมูลผู้ใช้

1.6 ขั้นตอนของพัฒนาระบบ

- ศึกษาทฤษฎีของการทดสอบซอฟต์แวร์เพื่อเป็นแนวทางการศึกษาและพัฒนาระบบงาน
- ศึกษาหลักการทำงานของยูนิเวอร์แซลเทสต์โคเวอริจ
- ศึกษาเครื่องมือและซอฟต์แวร์ที่ใช้ในการพัฒนาระบบงาน
- วิเคราะห์ระบบงานที่จะทำการพัฒนามบนพื้นฐานของหลักการของยูนิเวอร์แซลเทสต์โคเวอริจ เพื่อกำหนดขอบเขตพัฒนาระบบ
- ออกแบบระบบงาน โดยใช้ Unified Model Language
- ออกแบบฐานข้อมูลโดยใช้ Entity Relationship Diagram
- ออกแบบหน้าจอโปรแกรมประยุกต์บนเว็บในส่วนที่ติดต่อกับผู้ใช้
- พัฒนาส่วนชุดคำสั่งของระบบตามที่ได้ออกแบบไว้
- ทดสอบและแก้ไขปรับปรุงระบบให้ทำงานตามที่ออกแบบไว้
- ใช้ระบบที่ได้จากการพัฒนาเพื่อสรุปผลโครงการ
- จัดทำเอกสารสรุปการทำโครงการพัฒนา

1.7 ผลที่คาดว่าจะได้รับ

- ได้รับความรู้ความเข้าใจเกี่ยวกับการทดสอบซอฟต์แวร์แบบอัตโนมัติ โดยคาดหวังที่จะนำความรู้ไปประยุกต์ใช้ในอนาคต
- สามารถสร้างเครื่องมือที่ทำการทดสอบซอฟต์แวร์แบบกึ่งอัตโนมัติบนพื้นฐานของหลักการทำงานของยูนิเวอร์แซลเทสต์โคเวอริจเพื่อช่วยให้ทำการทดสอบได้รวดเร็วขึ้น
- เครื่องมือดังกล่าวสามารถอำนวยความสะดวกให้แก่ผู้ทดสอบโปรแกรม

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 การทดสอบซอฟต์แวร์

2.1.1 การทดสอบซอฟต์แวร์แบบทั่วไป (Traditional Software Testing)

Poston (1994 a:124-129) ในยุคแรกของการพัฒนาซอฟต์แวร์จะมีรูปแบบของการทดสอบเป็นการทดสอบที่ทำด้วยมือ นั่นคือการที่ผู้ทดสอบทำการป้อนข้อมูล 2-3 ค่าให้กับคอมพิวเตอร์แล้วดูการตอบสนองของซอฟต์แวร์ที่จะส่งผลลัพธ์กลับมา ซึ่งแบ่งการทำงาน 3 ขั้นตอน คือ

- *Setup* เป็นการตั้งสถานะให้อยู่ในสถานะพร้อมที่จะทำการทดสอบ โดยมีการให้ค่าข้อมูลเข้า (input) ที่จำเป็นในการทดสอบนั้น ๆ แก่ซอฟต์แวร์
- *Execute* เป็นการเรียกใช้การทำงานที่ต้องการ เพื่อให้ได้ซึ่งผลลัพธ์ ซึ่งจะนำไปพิจารณาต่อไปว่าตรงตามที่ต้องการหรือไม่ภายหลัง
- *Cleanup* เป็นการยืนยันผลที่ได้และเปลี่ยนสถานะของซอฟต์แวร์จากสถานะที่ทำการทดสอบเป็นสถานะปกติ นั่นคือหากมีการเปลี่ยนแปลงระบบใดๆ จากขั้นตอน Setup ก็ต้องทำการเปลี่ยนกลับให้อยู่ในสถานะปกติก่อนการทดสอบ

1) 'กรณีในการทดสอบ (Poston . 1994 b :48-58)

ปัจจุบันการทดสอบจะมีการสร้าง Test case หรือ กรณีในการทดสอบ ซึ่งเป็นชุดข้อมูลเข้าที่ใช้ในการทดสอบ (input) ขึ้นมาก่อนจากนั้นจึงทำการประมวลผล Test case และ การประเมินผลของการทดสอบ สำหรับการสร้าง Test case อาจออกแบบได้ด้วย 6 เทคนิคได้แก่

การทดสอบการทำงาน (Functional testing) แต่ละการทำงานหลัก ๆ นั้นจะมีการทดสอบอย่างน้อยหนึ่งครั้ง นั่นคือจะมีอย่างน้อยหนึ่ง Test case ที่มีชื่อ ค่าข้อมูลเข้าหรืออินพุต และค่าที่คาดหวัง

การวิเคราะห์ค่าขอบเขต (Boundary value Analysis) การหาข้อผิดพลาดในเรื่องของขอบเขตของค่าต่าง ๆ หรือ ตรวจสอบในช่วงของข้อมูลที่มากกว่าและน้อยกว่าค่าขอบเขต

การวิเคราะห์ชั้นสมมูล (Equivalence Class) เป็นการวิเคราะห์ตรวจสอบค่าที่เป็นไปได้ที่ถูกแบ่งแยกออกเป็นกลุ่มเล็ก ๆ ที่เรียกว่า Equivalence classes ซึ่งเราจะสร้าง Test case ที่มีข้อมูลเข้าอย่างน้อยหนึ่งค่าสำหรับแต่ละ sub domain และ equivalence class ที่ระบุ

Cause-effect Graphing การทดสอบเชิงตรรกะหรือเงื่อนไข ซึ่งทุก ๆ เงื่อนไขจะถูกตรวจสอบค่าความจริงที่เป็นจริง หรือ ค่าความจริงที่เป็นเท็จ กล่าวได้ว่าเป็นการทดสอบ Boolean Logic ซึ่ง test case จะถูกสร้างสำหรับตรวจสอบทุก ๆ เงื่อนไข ในทุก ๆ การกระทำในค่าที่เป็นจริง และ ค่าที่เป็นเท็จ

การทดสอบเหตุการณ์ (Event-directed Testing) เป็นการสร้าง test case ที่ทำทุก ๆ เหตุการณ์อย่างน้อยหนึ่งครั้ง โดยเหตุการณ์หมายถึงสิ่งที่เกิดขึ้นจากภายนอกระบบแล้วมีผลทำให้เกิดการกระทำขึ้นภายในระบบโปรแกรม นั่นคือ ต้องหาเหตุการณ์หรือสิ่งภายนอกระบบที่สามารถก่อให้เกิดระบบเกิดการกระทำหรือการกระทำขึ้น

การทดสอบสถานะ (State-directed Testing) คือ test case ที่ทำให้มีการเปลี่ยนสถานะ และจะต้องมีการสร้าง test case อย่างน้อย 1 ชุดต่อ 1 สถานะในการทดสอบ ซึ่งสถานะคือกลุ่มของข้อมูลหรือคุณลักษณะ หรือความสัมพันธ์ระหว่างค่านั้น ๆ และการเปลี่ยนสถานะคือการเปลี่ยนค่าเหล่านั้น

2) ประเภทของการทดสอบซอฟต์แวร์ (Patton . 2000)

สำหรับการทดสอบซอฟต์แวร์อาจแบ่งประเภทได้ 2 ประเภทจากวิธีการที่ใช้ในการทดสอบ คือการทดสอบแบบ *Black-Box* และ *White-Box* ซึ่งทั้งสองประเภทต่างกันว่า *black-box* ผู้ทดสอบจะรู้เพียงสิ่งที่ซอฟต์แวร์ควรจะได้โดยไม่ว่าทำได้ด้วยวิธีใด นั่นคือถ้าพิมพ์อินพุตบางค่าให้ก็จะได้รับเอาท์พุตสำหรับอินพุตนั้นกลับมาโดยที่ไม่รู้ว่ามันเกิดขึ้นได้อย่างไร ดังนั้นเพียงแต่ดูว่าค่าที่ได้รับนั้นมันถูกต้องสำหรับอินพุตหรือไม่เท่านั้น แต่สำหรับ *white-box* แล้วผู้ทดสอบจะต้องเข้าไปดูถึงโค้ดของ โปรแกรมและใช้เป็นแนวทางในการทดสอบซึ่งขึ้นอยู่กับว่าจะได้อะไรมาบ้างจากโค้ดก็จะนำมาเป็นข้อมูลสำหรับการทดสอบต่อไป

ในอีกมุมมองหนึ่งอาจแบ่งได้เป็นการทดสอบแบบ Static และการทดสอบแบบ Dynamic ซึ่งการทดสอบแบบ Static จะใช้ในการทดสอบอะไรก็ตามที่ไม่สามารถสั่งให้ประมวลผลได้ ส่วนมากจะเป็นการตรวจสอบและวิจารณ์ เช่นการตรวจสอบเกี่ยวกับเอกสารข้อกำหนดเบื้องต้น (Specification) ของโปรแกรม เป็นต้น ส่วนการทดสอบแบบ Dynamic จะเป็นการทดสอบสิ่งที่สามารถเรียกประมวลผลได้หรือต้องมีการใช้ซอฟต์แวร์ในการทดสอบซึ่งแบบนี้มักเป็นที่คุ้นเคยกันจากทั้งสองมุมมองหากนำมาใช้ร่วมกันจะเกิดประเภทของการทดสอบ 4 ประเภทดังนี้

- การทดสอบแบบ **Static Black-Box** มักใช้ในการตรวจสอบ Specification ซึ่งเป็นเอกสารที่ไม่สามารถประมวลผลได้ โดยไม่จำเป็นต้องรู้ที่มาของข้อมูลในเอกสารหรือเหตุผลของการสร้างเอกสาร
- การทดสอบแบบ **Dynamic Black-Box** มักใช้ในการทดสอบซอฟต์แวร์โดยที่ไม่รู้ถึงรายละเอียดของโค้ดภายในและใช้การทดสอบด้วยการเรียกประมวลผลเหมือนกับการใช้โปรแกรมตามปกติ
- การทดสอบแบบ **Static White-Box** มักใช้ในการตรวจสอบการออกแบบและโค้ดของโปรแกรม โดยไม่ต้องทำการเรียกประมวลผล แต่จะเป็นการวิเคราะห์โค้ดของโปรแกรมที่ต้องการทดสอบแทน
- การทดสอบแบบ **Dynamic White-Box** เป็นการทดสอบโปรแกรมที่ต้องรู้รายละเอียดของโค้ดเพื่อช่วยให้ตรวจสอบได้ถึงการทำงานภายใน ซึ่งยังคงมีการเรียกประมวลผลซอฟต์แวร์ตามเงื่อนไขของโค้ดที่มีโอกาสก่อให้เกิดข้อผิดพลาดต่าง ๆ

2.1.2 การทดสอบซอฟต์แวร์ด้วยบุคคลปฏิบัติ (Manual Software Testing)

เป็นการทดสอบซอฟต์แวร์โดยผู้ทดสอบกระทำเองทั้งหมดตามกระบวนการของการทดสอบซอฟต์แวร์ทั่วไปดังที่ได้กล่าวไว้ข้างต้นทุกครั้งที่ทำการทดสอบ

2.1.3 การทดสอบซอฟต์แวร์แบบอัตโนมัติ (Automated Software Testing)

Poston (1994 b:124-129)

การทดสอบแบบทั่วไปจะมีความยุ่งยากพอสมควรสำหรับผู้ทำการทดสอบที่ต้องทดสอบซอฟต์แวร์หลาย ๆ ครั้งหรือเมื่อมีการปรับปรุงซอฟต์แวร์ให้ดีขึ้นก็จะต้องทำการทดสอบอีกครั้งเพื่อให้มั่นใจว่าซอฟต์แวร์ดังกล่าวได้แก้ปัญหาที่พบในครั้งก่อนแล้ว ซึ่งการทดสอบเช่นนี้เรียกว่า การทำทดสอบแบบ Regression ซึ่งการทำการทดสอบซ้ำกันในกรณีเดิม ๆ ก็เป็นเรื่องที่น่าเบื่อหน่าย และเสียเวลาหากทำด้วยวิธีทั่วไป

แนวทางหนึ่งในการแก้ปัญหาก็คือการพัฒนาการทดสอบแบบอัตโนมัติ ซึ่งมีการใช้เครื่องมือช่วยในการทดสอบที่เรียกว่า Test Execution Tool โดยหลักการทำงานโดยทั่วไปของการทดสอบแบบอัตโนมัติก็จะคล้ายกับเครื่องเล่นวีดีโอเทป นั่นคือมีการบันทึก และการนำสิ่งที่บันทึกไว้มาเล่นซ้ำได้

การบันทึกของการทดสอบแบบอัตโนมัติ คือการบันทึกค่าต่าง ๆ ในการทำการทดสอบทั่วไป ทั้ง Test case หรือค่าสถานะต่าง ๆ ที่ต้องตั้งค่าใหม่ทุกครั้งเพื่อใช้ในการทดสอบ ซึ่งผู้ทำการทดสอบจะต้องทำสิ่งเหล่านี้ในขั้นตอน Setup ในครั้งแรกเท่านั้น ซึ่งหลังจากเริ่มการบันทึกค่าดังกล่าวแล้วผู้ทดสอบก็จะสามารถทำขั้นตอน Execute และ Cleanup ไปจนจบ แล้วจึงเลิกการบันทึก ซึ่งข้อมูลต่าง ๆ ที่ถูกบันทึกไว้หรือ Test Script ก็จะถูกเก็บไว้เพื่อนำกลับมาใช้ใหม่ ในการทดสอบครั้งต่อ ๆ ไปหรือที่เรียกว่าการเล่นซ้ำ กล่าวคือ Test Script จะถูกนำมาใช้ในการทดสอบแบบ Regression เพื่อประหยัดเวลาในการต้องตั้งค่าในขั้นตอน Setup นั่นเอง

ประโยชน์ของการทดสอบแบบอัตโนมัติ ได้แก่

- ความสะดวกรวดเร็ว
- ประสิทธิภาพในการทำงานมากขึ้นสำหรับผู้ทดสอบ
- ถูกต้องแม่นยำในการทดสอบซ้ำ ๆ กัน
- เครื่องมือสำหรับทดสอบแบบอัตโนมัติสามารถทำงานได้อย่างต่อเนื่องโดยไม่เหนื่อยล้า

2.1.4 การทดสอบซอฟต์แวร์แบบกึ่งอัตโนมัติ (Semi-Automated Software Testing)

เป็นการทดสอบซอฟต์แวร์แบบอัตโนมัติที่ยังต้องมีผู้ทดสอบกระทำงานบางกระบวนการอยู่บ้างแต่ไม่ทั้งหมด เช่น ผู้ทดสอบต้องทำการเรียกการทดสอบซอฟต์แวร์ที่มีการเตรียมข้อมูลไว้แล้วขึ้นมากระทำเพียงแต่กระบวนการตามที่ได้เตรียมข้อมูลไว้แล้วนั้นผู้ทดสอบไม่จำเป็นต้องกระทำเอง ระบบจะกระทำให้องแบบอัตโนมัติ

2.1.5 เครื่องมือในการทดสอบซอฟต์แวร์ (Test Tools) (Patton, 2000)

เครื่องมือสำหรับช่วยในการทดสอบนั้นมีมากมายหลายประเภทให้เลือกใช้ซึ่งในการตัดสินใจเลือกใช้ก็ขึ้นอยู่กับประเภทของซอฟต์แวร์ที่จะทดสอบและวิธีการที่ใช้ในการทดสอบว่าเป็นแบบ Black-box หรือ White-box ในที่นี้จะกล่าวถึงประเภทของเครื่องมือในหมวดหมู่ใหญ่ ๆ และวิธีการใช้งาน ดังนี้

1) Viewers และ Monitors เป็นเครื่องมือที่ทำให้ผู้ทดสอบเห็นรายละเอียดของการทำงานของซอฟต์แวร์ที่กำลังทดสอบซึ่งปกติไม่สามารถมองเห็น เช่น สามารถเห็นว่ากำลังได้ครบรหัสไหนที่กำลังประมวลผลอยู่ หรือฟังก์ชันใดกำลังทำงานอยู่ หรือรู้ว่ากำลังทดสอบไปตามเส้นทาง

ทำงานของโค้ดเส้นทางไหน หรือดูข้อมูลที่ถูกส่งผ่านเครือข่าย เป็นต้น ตัวอย่างของเครื่องมือประเภทนี้ได้แก่ Code coverage analyzer และ Communications analyzer

2) **Drivers** เป็นเครื่องมือที่ใช้ในการควบคุมและเรียกซอฟต์แวร์ขึ้นมาทดสอบ ตัวอย่างเช่น Batch file ที่เป็นรายการของโปรแกรมหรือคำสั่งง่าย ๆ ที่จะถูกเรียกประมวลผลเป็นลำดับ ซึ่งจะเรียก Batch file นี้ขึ้นมาทดสอบซอฟต์แวร์โดยที่ไม่จำเป็นต้องเฝ้าดูตลอดเวลา ในเวลาต่อมา Batch file ได้ถูกแทนที่ด้วยการใช้เอกสารสคริปต์แทนด้วยที่ซอฟต์แวร์มีความซับซ้อนมากขึ้น ตัวอย่างของเครื่องมือประเภทนี้อีกตัวอย่างคือการนำคอมพิวเตอร์อีกเครื่องหนึ่งที่มีโปรแกรมขนาดเล็กทำการสร้างข้อมูลเข้า (Input) สำหรับซอฟต์แวร์ที่ต้องทดสอบด้วยข้อมูลจำนวนมาก คอมพิวเตอร์ดังกล่าวจะทำงานเป็น driver ซึ่งสามารถสร้างข้อมูลได้เอง

3) **Stubs** เป็นเครื่องมือคล้าย Drivers โดยต่างกันว่า Stubs จะไม่ได้ไปควบคุมหรือเรียกซอฟต์แวร์อยู่ขึ้นมาทดสอบ แต่มันจะรับหรือตอบสนองต่อข้อมูลที่ซอฟต์แวร์ส่งมาแทน เช่น การทดสอบการพิมพ์ของเครื่องพิมพ์ โดยต้องพิมพ์ข้อมูลจำนวนมาก แทนที่จะใช้การพิมพ์ออกที่เครื่องพิมพ์เพื่อผลการพิมพ์เหมือนการใช้งานจริงซึ่งอาจเสียเวลาและไม่มีประสิทธิภาพ เราอาจใช้เครื่องคอมพิวเตอร์แทนเครื่องพิมพ์โดยที่เรียกโปรแกรมพิเศษขึ้นมาทำงานที่สามารถทำการอ่านและประมวลผลข้อมูลที่ส่งมาให้เครื่องพิมพ์ก็จะสามารถทำการทดสอบการพิมพ์ได้รวดเร็วและถูกต้องมากขึ้น

4) เครื่องมือ **Stress** และ **Load** เป็นเครื่องมือในที่ใช้สร้างสถานการณ์ที่ดึงเครียดและการทำงานที่หนักสำหรับซอฟต์แวร์ที่กำลังทดสอบ เพื่อใช้ทดสอบในสภาวะที่บีบบังคับซึ่งอาจทำให้เกิดการทำงานที่ผิดพลาดได้

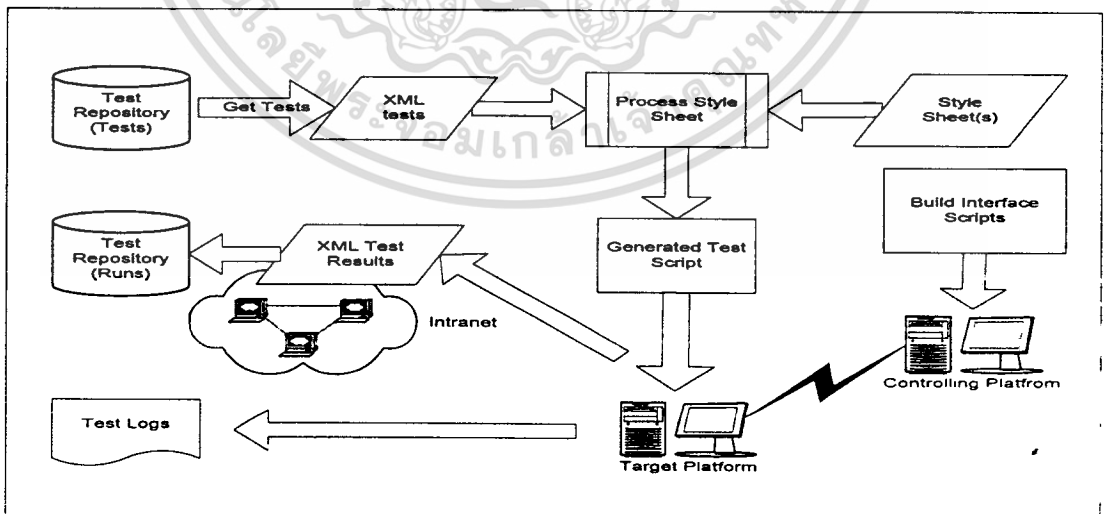
5) **Interference Injectors** และ **Noise Generators** เป็นเครื่องมือที่คล้ายกับเครื่องมือ Stress และ Load แต่มีการสร้างสถานการณ์แบบสุ่มมากกว่า ซึ่งมักใช้ในการทดสอบที่มีการเปลี่ยนแปลงสถานการณ์อย่างฉับพลัน เช่นมีการเพิ่มและลดจำนวนหน่วยความจำสำหรับการใช้ซอฟต์แวร์ด้วยจำนวนที่ต่างกันอย่างมาก ในการใช้เครื่องมือประเภทนี้อาจมีการใช้ Viewer มาประยุกต์ในการเฝ้ามองข้อมูลและทำการเปลี่ยนข้อจำกัดต่าง ๆ ของระบบด้วยตัวเอง

6) เครื่องมือสำหรับกรวิเคราะห้ เป็นเครื่องมือทั่วไปที่ผู้ทดสอบใช้ร่วมเพื่อให้การทดสอบง่ายขึ้น ตัวอย่างเช่น โปรแกรมงานเอกสาร (Word Processing Software) โปรแกรมสเปรดชีต (Spreadsheet Software) โปรแกรมฐานข้อมูล (Database Software) โปรแกรมเปรียบเทียบไฟล์ (File Comparison Software) โปรแกรมจับหน้าจอและเปรียบเทียบหน้าจอ (Screen Capture and Comparison Software) โปรแกรมตรวจสอบข้อผิดพลาด (Debugger) เครื่องมือในการคำนวณเลขฐานสองและฐานสิบหก (Binary-Hex Calculator) และกล้องถ่ายภาพ (VCR or Camera) เป็นต้น

2.2 หลักการทำงานของยูนิเวอร์แซลเทสโคโรเวอร์ (Cordrey, 2002)

โดยปกติแล้วเราจะมีซอฟต์แวร์ที่ต้องทำการทดสอบทั้งในเรื่องสถาปัตยกรรม (Architecture), ระบบปฏิบัติการ หรือ เวอร์ชันที่ต่าง ๆ กันเป็นจำนวนมาก ซึ่งในการพัฒนาซอฟต์แวร์แต่ละส่วนนั้นควรทำการทดสอบในหลาย ๆ รูปแบบด้วยกัน เช่น การทดสอบแต่หน่วยย่อย ๆ ของซอฟต์แวร์ (Unit Testing), การทดสอบโดยรวมแต่ละหน่วยย่อย ๆ เข้าด้วยกัน (Integration Testing), การทดสอบซ้ำ ๆ กันในเรื่องเดิมหลังจากที่มีการปรับปรุงซอฟต์แวร์เรื่องนั้น ๆ ให้ดีขึ้น (Regression Testing), การทดสอบประสิทธิภาพการทำงานของซอฟต์แวร์ (Performance Testing) และการทดสอบเกี่ยวกับขนาดของซอฟต์แวร์ (Scalability Testing) เป็นต้น นั่นคือเราต้องทดสอบให้ครบในทุก ๆ ระบบปฏิบัติการที่ซอฟต์แวร์สนับสนุน หรือทดสอบหลาย ๆ เวอร์ชัน ซึ่งถ้าหากทำการทดสอบโดยใช้คนเพียงอย่างเดียวอาจต้องมีการปรับเกี่ยวกับระยะเวลาที่จะทำการทดสอบ และสภาพแวดล้อมที่ใช้ให้เหมาะสมเนื่องจากช่วงเวลาก่อนที่เราจะส่งซอฟต์แวร์ให้แก่อูกค้านั้นมีจำกัด หรืออาจมีน้อย ดังนั้นการทดสอบแบบอัตโนมัติจึงเป็นทางเลือกหนึ่งที่จะช่วยให้การทดสอบเสร็จเร็วขึ้นและคนที่ทำหน้าที่รับผิดชอบก็ไม่ต้องทำการทดสอบเองทุกขั้นตอนซ้ำ ๆ กันซึ่งเป็นสิ่งที่น่าเบื่อหน่าย แต่กระนั้นการทดสอบที่ทำนั้นก็ยังต้องคำนึงว่าได้ทดสอบครอบคลุมในทุกเรื่อง หรือเพียงพอที่จะประกันได้ถึงคุณภาพของซอฟต์แวร์ที่จะส่งให้ลูกค้าด้วย

หลักการการทำงานของยูนิเวอร์แซลเทสโคโรเวอร์นี้เป็นการทดสอบบนพื้นฐานของการทำ Script หรือ Batch ไฟล์เพื่อใช้ในการทดสอบ ตัวอย่างเช่นการที่เราเรียกเครื่องมือที่ใช้ในการทำทดสอบด้วย Script ไฟล์และส่งข้อมูลของการทดสอบจากไฟล์นั้นไปในรูปแบบของคำสั่ง (command line) ซึ่งจะถูกนำไปเป็นข้อมูลเข้าสำหรับโปรแกรมที่ต้องการทดสอบ



รูปที่ 2.1 แผนผังการทำงานของยูนิเวอร์แซลเทสโคโรเวอร์

การทำงานของยูนิเวอร์แซลเทสไครเวอร์ประกอบไปด้วยส่วนต่าง ๆ ที่ใช้ในการทำการทดสอบแบบอัตโนมัติบนพื้นฐานของ Script ไฟล์บนหลาย ๆ ระบบปฏิบัติการในการทำงานบนระบบเครือข่าย ดังรูปที่ 2.1 ซึ่งแสดงหลักการทำงานของยูนิเวอร์แซลเทสไครเวอร์

2.2.1 องค์ประกอบของการทำงาน

องค์ประกอบของการทำงานของยูนิเวอร์แซลเทสไครเวอร์แบ่งได้ 6 องค์ประกอบหลักดังนี้

- คลังข้อมูลในการทดสอบ (Test Repository)
- เอกสารในการทดสอบ (Test Document)
- การเชื่อมต่อ เอกสารของการทดสอบแบบเอกซ์เอ็มแอล กับคลังข้อมูลในการทดสอบ (XML Interface to Repository)
- Style Sheets ในการแปลงข้อมูลของการทดสอบ (Test Transformation Style Sheets)
- ส่วนประมวลผลรูปแบบ (Style Processor)
- ตัวช่วยในการสร้าง Script (Script Building Utilities)

1) คลังข้อมูลในการทดสอบ (Test Repository)

เป็นส่วนที่อำนวยความสะดวกในการจัดดูแลและจัดเก็บการทดสอบต่าง ๆ ซึ่งโดยหลักการแล้วจะไม่เก็บในรูปแบบคำบรรยายรายละเอียดเป็นขั้นเป็นตอนแต่จะเก็บเพียงค่าตัวแปรที่จำเป็นในการแทนค่าใน Script ไฟล์ที่ใช้ในการทดสอบเท่านั้น ซึ่งคลังข้อมูลในการทดสอบมีข้อดีคือเป็นการเก็บรวบรวมข้อมูลในการทดสอบไว้ที่เดียวทำให้สามารถนำข้อมูลของการทดสอบมาใช้ใหม่อีกครั้งได้ (Test reuse) อีกทั้งยังสามารถเข้าข้อมูลถึงได้สะดวก ซึ่งเปรียบเสมือนว่าข้อมูลของการทดสอบนั้นเป็นทรัพยากรที่มีค่าที่จะยังคงอยู่แม้ว่าบุคลากรที่ทำงานด้านนี้จะมีการสลับเปลี่ยนไปก็ตาม นอกจากนี้การเลือกใช้คลังข้อมูลในการทดสอบนั้นอาจเป็นสร้างขึ้นใหม่โดยใช้แพคเกจจากที่อื่นหรือใช้คลังข้อมูลที่สร้างไว้แล้วก็ได้

โดยสรุปแล้วคลังข้อมูลในการทดสอบจะถูกใช้เป็นที่เก็บข้อมูลของการทดสอบ ซึ่งอาจประกอบไปด้วยตัวทดสอบ (Tests) หรือของกลุ่มทดสอบ (Test Groups) และรายชื่อเครื่องที่ใช้ในระบบ (Machine Lists) เป็นต้น

- **ตัวทดสอบ (Tests)** ตัวทดสอบประกอบไปด้วยความสัมพันธ์ระหว่าง Keyword กับค่าของมัน ซึ่งมีความสัมพันธ์กับทุก ๆ ค่าข้อมูลที่เป็นตัวแปรใน Script ของตัวทดสอบ กล่าวคือเมื่อทำการระบุค่าตัวแปรต่าง ๆ ใน Script ได้แล้วก็จะกำหนด Keyword สำหรับแต่ละตัวแปรนั้น โดย

เป็นไปในรูปแบบของ “พารามิเตอร์” ของตัวทดสอบซึ่งจะสามารถกำหนดได้จากการทำการวิเคราะห์ Script ที่จะใช้งานและจากการกำหนดค่าต่าง ๆ ที่จะมีการเปลี่ยนค่าจากตัวทดสอบหนึ่ง ๆ ไปยังตัวทดสอบตัวต่อไป ตัวอย่างเช่น

`test_program = test_move`

(หมายถึงโปรแกรมที่จะทดสอบชื่อ test_move)

นอกจากนี้เพื่อความสะดวกและประโยชน์ในการใช้งานเราอาจเพิ่ม Attribute ของตัวทดสอบคือ “Subtype” ที่ใช้จัดกลุ่มการทดสอบโดยใช้ลักษณะการทำงานหรือตามพฤติกรรมในการทดสอบเช่นแบ่งเป็น Unit Tests, Integration Tests และ Smoke Tests ซึ่งในแต่ละ Subtype ก็จะมี Script ที่ต่างกันออกไป

- **กลุ่มตัวทดสอบ (Test Groups)** ในคลังข้อมูลยังประกอบไปด้วยกลุ่มต่าง ๆ ของการทดสอบซึ่งทั้งหมดในกลุ่มจะถูกเรียกทำงานพร้อมกัน ซึ่งอาจจัดกลุ่มโดยใช้ลักษณะการทำงานหรือตามพฤติกรรมในการทดสอบเช่นแบ่งเป็น Unit Tests, Regression Tests, Integration Tests และ Smoke Tests หรืออาจจัดกลุ่มโดยแบ่งตามระบบปฏิบัติการ หรือจัดตามคุณสมบัติอื่น ๆ ที่สะดวกในการใช้งานก็ได้
- **รายชื่อเครื่องที่ใช้ในระบบ (Machine Lists)** เป็นรายชื่อเครื่องในการทดสอบด้วยตัวกลุ่มทดสอบหนึ่ง ๆ ซึ่งแต่ละเครื่องอาจใช้ชื่อ Domain Name Service (DNS) สำหรับการเข้าถึงเครื่องนั้น ๆ นอกจากนี้ชื่อที่ถูกตั้งขึ้นมาสำหรับสภาพแวดล้อมในการทดสอบจะสามารถบอกได้ถึงประเภทของระบบปฏิบัติการ, เวอร์ชัน หรือคอมพิวเตอร์ที่ใช้ด้วย

2) เอกสารในการทดสอบ (Test Document)

เอกสารในการทดสอบรูปแบบ Extensible Markup Language (XML) จะเป็นตัวช่วยให้ข้อมูลของการทดสอบสามารถแปลงเป็นรูปแบบของไฟล์ Script หรือรูปแบบที่เหมาะสมสำหรับจัดเก็บและเรียกใช้จากคลังข้อมูลได้ง่าย ที่สำคัญคือการใช้เอกสารแบบ เอกสารของการทดสอบแบบเอกซ์เอ็มแอล นี้ยังช่วยให้ทำการกระจายข้อมูลไปทางเว็บง่ายขึ้น ส่งผลให้ผู้ใช้สามารถเข้าถึงได้ด้วยบราวเซอร์ในการเรียกดูข้อมูลหรือผลของการทดสอบ จึงอำนวยความสะดวกให้สามารถทำงานได้เป็นวงกว้างมากกว่าเดิมที่เป็นการเข้าถึงคลังข้อมูลโดยตรง

ในเอกสาร เอกสารของการทดสอบแบบเอกซ์เอ็มแอล นี้จะประกอบไปด้วยข้อมูลทั้งหมดที่เกี่ยวกับโครงสร้างที่ได้อธิบายไว้ในส่วนของคลังข้อมูลก่อนหน้านี้แล้ว นอกจากนั้นเพื่อให้เอกสารมีรูปแบบที่เป็นมาตรฐานในการสร้างก็จะต้องมีการกำหนดข้อตกลงในการเขียนข้อมูลในเอกสารนี้ด้วยการให้นิยามของข้อมูลแต่ละชนิดในเอกสารที่เรียกว่า Data Type Definition: DTD

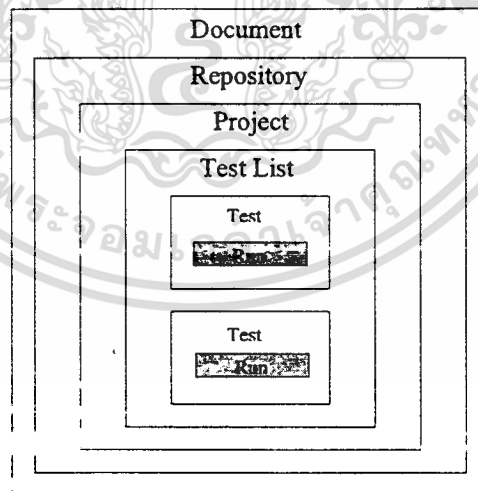
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรืออาจใช้ XML schemas แต่สำหรับ DTD จะมีการใช้งานที่แพร่หลายมากกว่าในปัจจุบันแม้ว่า XML schemas จะมีความยืดหยุ่นมากกว่าก็ตาม

ส่วนประกอบพื้นฐาน (Basic Elements) ในเอกสารของการทดสอบประกอบไปด้วย

- เอกสาร (Documents)
- คลังข้อมูล (Repository) ซึ่งอาจมีหลายที่ขึ้นอยู่กับการจัดรูปแบบองค์กร
- โครงการ (Project) ซึ่งเป็นไปได้ที่จะแตกต่างกันไปตามแผนก หรือกลุ่มการทำงาน
- รายการที่จะทดสอบ (Test List)
- ตัวทดสอบ (Tests)
- ตัวทดสอบที่ทำไปแล้ว (Runs)
- ลำดับขั้นการทดสอบ (Steps) ซึ่งอาจอยู่ในตัวทดสอบ หรือ ตัวทดสอบที่ทำไปแล้ว

นอกจากที่กล่าวแล้วอาจมีส่วนประกอบเสริมด้วยเพื่ออธิบายความหมายต่าง ๆ ได้แก่ ค่าคาดหวัง (Expected Result), ค่าจริงที่ได้ (Actual Result) และ ค่าสถานะ (Status) ในที่นี้สามารถดูตัวอย่าง DTD ของเอกสารการทดสอบในภาคผนวก ก. ซึ่งมีจะมีโครงสร้างเป็นไปตามรูปที่ 2.2



รูปที่ 2.2 ตัวอย่างโครงสร้าง DTD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการใช้งานจริงอาจเป็นที่โต้แย้งกันได้ระหว่างการใช้อยู่ XML Elements หรือ XML Attributes ในเอกสารของการทดสอบแบบเอกซ์เอ็มแอล ซึ่งที่จริงแล้วขึ้นอยู่กับความเหมาะสมและความถนัด แต่ในที่นี้มีความเห็นว่าจะใช้ XML Elements มากกว่าเนื่องจากผู้ออกแบบเชื่อว่าการเขียนโปรแกรมที่เกี่ยวกับเอกสารในยุคนี้สามารถเข้าถึง Element ได้มากกว่า ตัวอย่างที่เห็นได้ง่ายที่สุดก็คือ STATUS element ของ RUN เนื่องจากเราจะยังไม่ทราบสถานะของการทดสอบ RUN จนกว่าการทดสอบจะสิ้นสุดลง ซึ่งการสร้าง XML tags ในขณะที่การประมวลผลไปเราจะสามารถเพิ่ม STATUS element ตามหลัง STEP tags ได้เพื่อเป็นค่าสถานะของ RUN แต่ถ้าเขียนแบบใช้ Attribute ก็จะต้องเตรียมส่วนข้อมูลสำรอง (buffer) ของ RUN tags และ STEP tags ไว้จนกระทั่งการทดสอบสิ้นสุดลงเราสามารถจึงจะใส่ค่า STATUS attribute ใน RUN tag นั้น ๆ ได้ ดังนั้นจะเห็นได้ว่าข้อมูลเดียวกันอาจทำได้หลายแบบ ดังตัวอย่างที่ได้ยกมา

สังเกตได้ว่าเราควรคำนึงถึงการทำให้ Script ที่เข้าใจง่าย และควรเขียน DTD เพื่อความสะดวกในการตรวจสอบเอกสารของการทดสอบแบบเอกซ์เอ็มแอล ทั้งนี้ทั้งนั้นก็ควรคำนึงถึงความเป็นไปได้ในการนำไปใช้จริงด้วย

3) การเชื่อมต่อเอกซ์เอ็มแอล กับคลังข้อมูลในการทดสอบ(XML Interface to Repository)

ส่วนการทำงานหนึ่งที่สำคัญก็คือการเชื่อมต่อที่เป็นกลไกในการคัดลอกข้อมูล (Extracting) จากคลังข้อมูลให้เป็นรูปแบบของ เอกสารของการทดสอบแบบเอกซ์เอ็มแอล และทำกลไกในการแทรกข้อมูลจากผลของการทดสอบที่อยู่ในเอกสารของการทดสอบแบบเอกซ์เอ็มแอล กลับไปยังคลังข้อมูล ซึ่งการเชื่อมต่อนี้มีประโยชน์มากไม่ใช่ว่าเพียงในหลักการทำงานของยูนิเวอร์แซลเทสต์ไครเวอร์เท่านั้นแต่ยังสามารถนำไปใช้ในกรณีที่เป็นการทำทดสอบซอฟต์แวร์แบบธรรมชาติที่ไม่เป็นแบบอัตโนมัติ ในการเก็บข้อมูลของการทดสอบเข้าคลังข้อมูลอีกด้วย

คลังข้อมูลที่มีขายอยู่ในตลาดมักจะมีส่วนเชื่อมต่อกับการเขียนโปรแกรม (Application Programming Interfaces: APIs) ซึ่งสามารถนำมาใช้สร้างการเชื่อมต่อที่ทำการอ่านเขียนข้อมูล และดูแลรักษาเอกสารเอกซ์เอ็มแอล ซึ่งในปัจจุบันก็มีพวกฐานข้อมูลเอกซ์เอ็มแอล เช่น ฐานข้อมูลของ Oracle บางเวอร์ชันที่ไม่จำเป็นต้องสร้างการเชื่อมต่อดังกล่าว หรืออาจใช้พวก SQL procedure เป็นอีกทางเลือกหนึ่งในการทำก็ได้

ในที่นี้ผู้ออกแบบได้ใช้ SAX-based parser ซึ่งเป็น API ง่ายสำหรับเอกซ์เอ็มแอล และสามารถหาได้ทั่วไปจาก World Wide Web ซึ่งเราจะนำมารวมเข้ากับส่วนการทำงานที่ต้องเชื่อมต่อ API กับคลังข้อมูล โดยที่ผู้ออกแบบได้แนะนำให้อยู่ในประเภท SAX event-oriented parser

มากกว่าแบบอื่นเพราะว่าไฟล์เอกซ์เอ็มแอล ของผลการทดสอบนั้นอาจมีขนาดใหญ่และรูปแบบของ event based ก็ทำให้ไม่จำเป็นต้องสร้าง object model ทั้งหมดไว้ในหน่วยความจำซึ่งใช้ในการ Parse แบบ DOM (Document Object Model)

สรุปคือสำหรับการเรียกใช้ข้อมูลการทดสอบขึ้นมา (Get Tests) ส่วนการเชื่อมต่อเอกซ์เอ็มแอล จะถูกใช้เพื่อทำการคัดลอกข้อมูลรายการการทดสอบ (Test Lists) และรายการเครื่องที่จะทดสอบ (Machine Lists) จากคลังข้อมูล โดยจะมีเอกสารของการทดสอบแบบเอกซ์เอ็มแอลสำหรับนำเข้า (input) ไปยังตัวเชื่อมต่อกับคลังข้อมูลเพื่อให้ได้รายการการทดสอบที่เรียกว่า “Get Test XML” ดูตัวอย่างได้จากภาคผนวก ก. ซึ่งอาจจะมีการแทรกคำสั่งที่ระบุได้ว่าข้อมูลอะไรบ้างที่จะต้องมีในเอกสารของการทดสอบแบบเอกซ์เอ็มแอล ของการทดสอบที่จะนำไปสร้างเป็นเอกสาร Script สำหรับทดสอบต่อไป

4) Style Sheets ในการแปลงข้อมูลของการทดสอบ (Test Transformation Style Sheets)

การที่จะแปลงข้อมูลพารามิเตอร์ของการทดสอบในเอกสารของการทดสอบแบบเอกซ์เอ็มแอล เป็น Shell Script จริง ๆ อาจใช้การเขียน Extensible Style Sheet Language (XSL) ช่วย ซึ่ง XSL Style Sheet จะระบุถึง XML schemas ที่เป็นรูปแบบการแสดงผลข้อมูลในเอกสารเอกซ์เอ็มแอล โดยจะมีการอธิบายว่าจะแปลงกลุ่มข้อมูลต่าง ๆ ของเอกซ์เอ็มแอล อย่างไรให้ออกมาในรูปแบบตามที่ต้องการได้

ตัวอย่างที่เห็นได้ชัดคือการนำเสนอข้อมูลด้วย Elements ต่าง ๆ ของ HTML (Hyper-Text Markup Language) ที่มีความสัมพันธ์เกี่ยวข้องกับ Elements ของ เอกสารเอกซ์เอ็มแอล กล่าวคือจะสร้าง Style Sheet หนึ่งขึ้นมาที่สามารถอ้างอิงคู่กับเอกสารการทดสอบเพื่อให้เห็นผลบนบราวเซอร์ง่ายขึ้น ซึ่งในการใช้งานอาจทำโดยเรียกไฟล์ Style Sheet รวมเข้ากับไฟล์เอกซ์เอ็มแอล โดยตรงหรืออาจใช้ JavaScript หรือ ASP (Active Server Pages) ช่วยในการโหลดไฟล์นั้น Style Sheet ให้มาทำงานร่วมกับเอกซ์เอ็มแอล ก็ได้เพื่อให้ได้ไฟล์ในรูปแบบ HTML

แต่ที่เป็นประโยชน์ที่สุดสำหรับการทำงานของยูนิเวอร์แซลเทสต์ไคเวอร์ก็คือนำไปใช้ในการสร้างข้อความในรูปแบบของ Shell Script ซึ่งในการประมวลผล Element ในเอกสารของการทดสอบแบบเอกซ์เอ็มแอล ของการทดสอบ Style Sheet ตัวหลักก็มีการบอกว่าจะประมวลผลอย่างไร

นอกจากนี้ยังอาจทำการแยก Style Sheet ออกมาสำหรับแต่ละการทดสอบประเภทย่อย (Test Subtype) และทำการรวม Style Sheet เหล่านั้นไว้กับ Style Sheet ตัวหลัก กล่าวคือ Style Sheet ย่อย ๆ เหล่านี้จะเป็นตัวบอกว่าจะแปลง Test Elements และ Step Elements ในเอกสารของการ

ทดสอบแบบเอกซ์เอ็มแอล ได้อย่างไร ซึ่ง Style Sheet ยังนำไปใช้สำหรับการแปลงข้อมูลเกี่ยวกับรายการเครื่อง (Machine List) ที่ใช้ในการทดสอบอีกด้วย

5) ส่วนประมวลผลรูปแบบ (Style Processor)

ในการแปลงภาษาในรูปแบบ Tree ของเอกซ์เอ็มแอลให้เป็นข้อความรูปแบบอื่นเช่น HTML หรือเอกซ์เอ็มแอล ในรูปแบบอื่นนั้นทาง W3C ได้แนะนำตัวประมวลผลที่มีประสิทธิภาพ ได้แก่ XSLT (the Extensible Style Sheet Language Transformations) และ XPath (the XML Path Language) ซึ่งตัวประมวลผลตามที่ทาง W3C แนะนำสามารถทำงานได้ดีแต่บางตัวประมวลผลอาจทำการคอมไพล์ได้ไม่สมบูรณ์ในตัวเอง [1] อย่างเช่นในตัวประมวลผลแบบ XSLT ซึ่งมักต้องใช้ร่วมกับภาษาจาวาดังนั้นจึงจำเป็นต้องมี JRE (Java Runtime Engine) หรือตัวที่ทำการประมวลผลเกี่ยวกับภาษาจาวาคือ

สำหรับเงื่อนไขในการเลือกตัวประมวลผลเราอาจให้ความสำคัญกับระบบปฏิบัติการที่จะนำตัวประมวลผลนั้นไปทำงานไม่ว่าจะเป็น UNIX หรือ Windows แต่จริงแล้วมันก็ไม่ได้ทำให้เกิดการทำงานที่ต่างกันแต่อย่างใด เพราะกระบวนการควบคุมในการสร้าง Script ก็ยังคงสร้าง Script ที่เหมาะสมได้จากเครื่องแม่ (host) ซึ่งจะสร้าง Script ด้วยการแตกข้อมูลจากคลังข้อมูลของการทดสอบโดยตรง หรือ ไม่ก็อาจใช้การแปลงข้อมูลด้วย XSLT ช่วย

การทำงานในส่วนนี้ไม่ซับซ้อนมากนัก กล่าวคือตัวประมวลผลจะอ่านไฟล์เอกซ์เอ็มแอลที่เป็นข้อมูลนำเข้า (input) ซึ่งประกอบไปด้วยตัวทดสอบ ต่าง ๆ และอ่านไฟล์ Style Sheet ซึ่งเป็นไปได้ว่าจะมี Style Sheet ของการทดสอบประเภทย่อย ๆ รวมอยู่ด้วย จากนั้นก็จะประมวลผลสร้างไฟล์ที่มี Shell Script แทรกอยู่ด้วยขึ้นมา หรือที่เรียกว่า Script ของการทดสอบ ดังตัวอย่าง Script ในภาคผนวก ข.

นอกจากนั้นส่วนนี้ยังจัดการเกี่ยวกับรายการเครื่อง (Machine List) ที่ใช้ในการทดสอบซึ่งรายการจะถูกระบุไว้ในไฟล์เอกซ์เอ็มแอล โดยจะสร้างเป็นไฟล์รายการในรูปแบบ ASCII สำหรับใช้ในการทดสอบในตอนเริ่มทดสอบด้วย Script บนเครื่องที่กำหนด

6) ตัวช่วยในการสร้าง Script (Script Building Utilities)

ในการเขียน Style Sheets เพื่อใช้เป็นรูปแบบในการสร้าง Script สำหรับการทดสอบถ้ามี Utility Scripts ด้วยจะเป็นประโยชน์ในการจัดการเกี่ยวกับรายละเอียดของ XML tags, จัดการเกี่ยวกับ Test Mode และจัดการเกี่ยวกับการสร้าง Script ซึ่งตัวอย่างรายการของ Utilities มีดังตารางที่ 2.1

ตารางที่ 2.1 ตัวอย่างรายการ Utilities ในการสร้าง Script

Script	Description
add_command	Add a simple command to the script
add_step	Add a testing step to the script
create_test_script	Create the test script
daily_testing.ksh	Initiate the testing following detection of the Build
detect_new_build.ksh	Monitor the inbox for a new Build
killzombies	Terminates all user's processes whose parent PID=1
logevent	Write a date/time stamped record
print_end_tag_driver	Print Document XML end tag
print_end_tag_project	Print Project XML end tag
print_end_tag_repository	Print Repository XML end tag
print_end_tag_run	Print Run XML end tag
print_end_tag_step	Print Step XML end tag
print_end_tag_test	Print Test XML end tag
print_end_tag_testlist	Print Testlist XML end tag
print_start_tag_driver	Print Document XML start tag
print_start_tag_project	Print Project XML start tag
print_start_tag_repository	Print Repository XML start tag
print_start_tag_run	Print Run XML start tag
print_start_tag_step	Print Step XML start tag
print_start_tag_test	Print Test XML start tag
print_start_tag_testlist	Print Testlist XML start tag
run_test_script	Run the generated script
set_test_mode	Set Stop or Continue or Fail Mode
time_out	Kill a given process if it still exists after a time value
waitfor.ksh	Wait for a given file to be created
waitforgz.ksh	Wait for a compressed file to be valid

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในแต่ละ Tags ของเอกสารของการทดสอบแบบเอกซ์เอ็มแอล จะมีการนำ Utilities เกี่ยวกับการสร้าง Start tag และ End tag มาใช้ตัวอย่างเช่น ใน XSL Style Sheet จะมีการเขียน

```
.print_start_tag_test name folder description
```

ซึ่งจะทำให้ได้โค้ดดังนี้

```
echo " <TEST"
echo "      name = \"\$1\""
echo "      folder = \"\$2\""
echo " >"
echo " <DESCRIPTION><![CDATA[\"$3\"]></DESCRIPTION>"
```

ในส่วนที่เกี่ยวกับการจัดการ Test mode จะเป็นตัวบอกว่าหากการประมวลผลลำดับขั้นการทดสอบ (Steps) ล้มเหลวควรจะประมวลผล Script ต่อหรือไม่ โดยจะมีการกำหนดสถานะของ Test mode ไว้

นอกจากเกี่ยวกับ Test mode แล้ว Utilities ที่สำคัญต่อการทำงานในการเริ่มต้นสร้าง Script ได้แก่

- **add_command:** ใช้เพื่อใส่ Shell Command ใน Script ที่สร้างขึ้น เช่นพวก Change Directory command หรือ cd เป็นต้น
- **add_step:** ใช้เพื่อใส่ command สำหรับการทดสอบใน Script เช่นคำสั่งให้ประมวลผล หรือให้เก็บและแยกกลุ่มข้อผิดพลาด (error output) สำหรับ "Actual Result" tag ในเอกสารการทดสอบ แล้วนำไปเปรียบเทียบกับค่า "Expected Value" แล้วจึงนำผลการเปรียบเทียบไปใส่ใน Status tag ของ Run เป็นต้น

ตัวอย่างของ Script ดูจากภาคผนวก ข.

2.2.2 สภาพแวดล้อมของการทดสอบ

เครื่องคอมพิวเตอร์ที่จะทดสอบจะมีการเชื่อมต่อผ่านระบบเครือข่ายทั้งหมด ดังนั้นจึงมีเครื่องที่คอยควบคุม (Controlling machine) ที่ทำหน้าที่เริ่มทดสอบ แม้ว่าตามหลักการงานยูนิเวอร์แซลเทสใดเวอร์จจะมีการติดตั้งระบบในทุกเครื่องก็ตามแต่ในการทำงานจริงยังคงต้องสร้างการติดต่อสื่อสารระหว่างเครื่องให้ค่อนข้างยืดหยุ่นด้วย เนื่องจากผู้ใช้สามารถเข้าถึงทุก ๆ เครื่องได้ซึ่งได้มีการนำ File Server มาใช้ในการจัดเก็บ Home Directory ของผู้ใช้ เพื่อให้สามารถ

เข้าถึงได้จากทุกเครื่องในแบบ NFS automount ดังนั้นเครื่องที่ใช้ระบบปฏิบัติการ Windows อยู่ จะต้องมีการเข้าถึง Home Directory ด้วยการทำ drive mapping ไปยังเครื่องที่ใช้ Samba Software ในการเข้าถึงระบบ File system ของ UNIX โดยที่ Script ทั้งหมดที่ใช้ในการประมวลผลจะเป็น Script เดียวกันที่เขียนด้วย Korn shell ทั้งสองระบบ และเพื่อที่จะทำให้สามารถเข้าถึงแต่ละเครื่องที่ใช้ทดสอบจะต้องมีการกำหนดค่าต่าง ๆ ในแบบ remote shell (remsh) command จากเครื่องที่ควบคุมการทดสอบด้วย

ในทางตรงกันข้ามการติดต่อจากเครื่อง UNIX ไปยังเครื่องที่ใช้ระบบปฏิบัติการ Windows สามารถทำโดยใช้ utility ที่จัดการการเข้าถึงแบบ remote logon จากเครื่อง UNIX ไปยังเครื่อง Windows ซึ่ง utility นี้จะกำหนดค่าในการเรียกใช้สภาพแวดล้อม Cygwin bash shell ใน UNIX เอง (เฉพาะ UNIX ที่สนับสนุน Red Hat) ซึ่ง Script แบบนี้จะเขียนด้วย bash shell

2.2.3 การประยุกต์ใช้การทดสอบให้ทำงานร่วมไปกับการพัฒนาระบบและการรวมระบบ (Interface to Build and Packaging)

เพื่อให้ยูนิเวอร์ซัลเทสต์ไควเวอร์มีประสิทธิภาพมากที่สุดอาจเพิ่มความสามารถในการตรวจสอบว่าโปรแกรมที่จะต้องทดสอบนั้นมีการพัฒนาหรือรวมกลุ่มเสร็จแล้วด้วย ซึ่งจะช่วยให้ทำการทดสอบได้เองช่วงเวลาใดก็ได้เมื่อโปรแกรมพร้อมที่จะทดสอบ

ส่วนมากการนำการทดสอบมารวมกับกระบวนการพัฒนาโปรแกรมมักจะทำบนเครื่อง UNIX เพราะง่ายต่อการใช้ Home Directory หรือ Directory ในการพัฒนาร่วมกัน และมีความสามารถในการจัดลำดับการทำงานที่ดี

ในการทำเช่นนี้จะต้องมีส่วนที่ทำการตรวจสอบ "Imbox" ซึ่งเป็นที่ที่กำหนดไว้เพื่อเก็บไฟล์ที่บอกได้ว่าการพัฒนาหรือการรวมกลุ่มที่เสร็จและพร้อมที่จะทำการทดสอบ (Trigger files) โดยไฟล์ดังกล่าวถูกสร้างมาจากฝ่ายพัฒนาโปรแกรม ดังนั้นจะต้องมีการร่วมมือกับฝ่ายพัฒนาโปรแกรมในการเพิ่มเงื่อนไข logic ในกลุ่ม procedures ที่พัฒนาอยู่เพื่อให้มีการสร้างไฟล์ดังกล่าวไว้ในที่ที่กำหนดไว้ และเมื่อมีการตรวจพบ Trigger file ก็จะมีการเรียกใช้ Script ของการทดสอบในเรื่องนั้น ๆ เพื่อแสดงรายละเอียดเกี่ยวกับงานของการทดสอบในครั้งก่อน ๆ เช่นผลการทดสอบ หรือ logs ให้กับฝ่ายที่ควบคุมการทดสอบเห็น

จากนั้นกระบวนการทดสอบก็จะเริ่มขึ้นโดยการเอา Script มาประมวลผลเพื่อเอาข้อมูลการทดสอบเช่น ตัวทดสอบ และรายการเครื่องที่ใช้ในการทดสอบ ซึ่งข้อมูลดังกล่าวจะมีการประมวลผลผ่าน XSL Translator เพื่อสร้าง Test Script และ Machine List

นอกจากนี้สำหรับทุก ๆ เครื่องในรายการแล้ว Script จะถูกนำไปใช้เริ่มทำงานบนเครื่องนั้น ๆ ด้วยการทำ Remote Logon โดยจะมีการรองรับโปรแกรมที่พัฒนาเสร็จที่จะต้องทำงานในระบบปฏิบัติการบนเครื่องนั้น ๆ และเมื่อ Test Script มีการประมวลผลบนเครื่องนั้นเสร็จก็จะได้ Logs Files และ XML Files ที่มีรายละเอียดของการทดสอบและผลการทดสอบ ทั้ง Logs และผลการทดสอบจะมีการแสดงผลบนเว็บแบบ online และผลการทดสอบในรูปแบบเอกซ์เอ็มแอล จะถูกนำผ่านกระบวนการแปลงข้อมูลเพื่อเก็บในคลังข้อมูลอีกที

2.2.4 ข้อดีและความเสี่ยงที่ควรจัดการ

1) ข้อดี

- สามารถพัฒนาเพื่อทดสอบ Command line ตาม Scenario ที่ต้องการได้
- ส่งเสริมการใช้คลังข้อมูลกลางในการเก็บ Test Cases, Test Execution และรายงาน
- สามารถสร้างการทดสอบที่ซ้ำ ๆ กันได้โดยไม่ต้องใช้คนทำงานนั้นซ้ำ กันบ่อย ๆ
- เป็นการประมวลผลและทำรายงานแบบอัตโนมัติทั้งหมด
- สามารถทำงานในขณะไต่บนหลาย ๆ ระบบปฏิบัติการด้วยการทำงานแบบอัตโนมัติ
- Script ที่ใช้ทดสอบบำรุงรักษาง่ายและเป็น Script กลางที่เรียกใช้ได้ทั่วไป
- มีการแสดงผลการทดสอบในรูปแบบเว็บ และบันทึกในคลังข้อมูล
- ฝ่ายทดสอบโปรแกรมสามารถเพิ่มตัวทดสอบได้ง่ายโดยไม่ต้องพึ่งการสนับสนุนจากฝ่ายอื่นมากนัก

2) ความเสี่ยง

มีความเสี่ยงเกี่ยวกับเรื่องเทคโนโลยีซึ่งขึ้นกับประสิทธิภาพขององค์กรการทำงานเกี่ยวกับเอกซ์เอ็มแอล ซึ่งในระยะเริ่มแรกของการพัฒนา XSL จะค่อนข้างซับซ้อนมากกว่าการทำ Script สำหรับการทดสอบหนึ่ง เพราะต้องสร้างให้สามารถใช้งานได้ครอบคลุมการทดสอบทุกเรื่องด้วย

2.3 หลักการวิเคราะห์และออกแบบเชิงวัตถุด้วย UML (สุนทริน วงศ์ศิริกุล. ม.ป.ป.)

2.3.1 Unified Modeling Language (UML)

UML เป็นภาษาเชิงสัญลักษณ์ที่ใช้อธิบายระบบโดยจะประกอบไปด้วยแผนภาพทั้งหมด 8 รูปแบบ ดังนี้

- ยูสเคสไดอะแกรม (Use-case Diagram)
- คลาสไดอะแกรม (Class Diagram) : *(Static)*
- ไดอะแกรมเชิงพฤติกรรม (Behavior Diagram) : *(Dynamic)* จำนวน 4 รูปแบบ
 - ซีควีนซ์ไดอะแกรม (Sequence Diagram)
 - คอลลาบอเรชันไดอะแกรม (Collaboration Diagram)
 - สเตตชาร์ตไดอะแกรม (Statechart Diagram)
 - แอกทิวิตีไดอะแกรม (Activity Diagram)
- ไดอะแกรมเชิงการโปรแกรม (Implementation Diagram) จำนวน 2 รูปแบบ
 - คอมโพเนนต์ไดอะแกรม (Component Diagram)
 - ดีพลอยเมนต์ไดอะแกรม (Deployment Diagram)

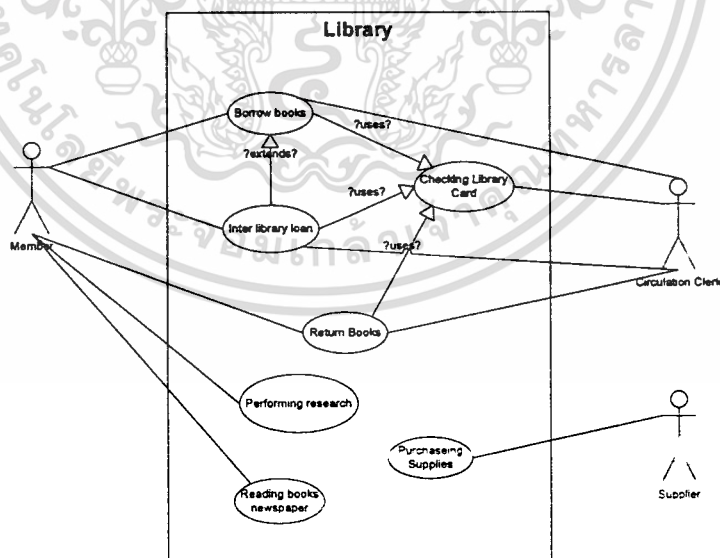
โดยที่ซีควีนซ์และคอลลาบอเรชันไดอะแกรม เป็นแผนภาพที่แสดงการทำงานร่วมกันหรือมีผลต่อกันระหว่างออบเจกต์เช่นเดียวกัน เพียงแต่จะต่างกันที่มุมมองของการสร้างแผนภาพ ดังนั้นทั้งสองไดอะแกรมจึงอาจเรียกรวมว่ากลุ่มอินเตอร์แอกชันไดอะแกรม (Interaction Diagrams)

1) ยูสเคสไดอะแกรม

ยูสเคสไดอะแกรมเป็นแผนภาพที่แสดงความต้องการของระบบ นั่นคือแสดงให้เห็นได้ว่าระบบนั้นมีการดำเนินงานอะไรบ้าง โดยในแผนภาพจะประกอบด้วย

- แอกเตอร์ (Actor) หมายถึงบุคคล หรือ สิ่งต่าง ๆ ที่เข้ามาใช้ระบบ และมีการแลกเปลี่ยนข่าวสาร (Message) กับระบบ แทนด้วยสัญลักษณ์รูปคน
- ยูสเคส (Use case) หมายถึงกลุ่มกิจกรรมหลักที่เกิดขึ้นในระบบซึ่งอาจเป็นกิจกรรมที่เกิดขึ้นระหว่างผู้ใช้กับระบบหรือระบบกับระบบ แทนด้วยสัญลักษณ์รูปวงรี
- ความสัมพันธ์ (Relationships) เป็นความสัมพันธ์ระหว่าง แอกเตอร์และยูสเคส หรือระหว่างยูสเคสด้วยกันเอง หรือระหว่างยูสเคสด้วยกันเอง ซึ่งแบ่งความสัมพันธ์ได้เป็น 3 รูปแบบคือ

- **Communication** เป็นความสัมพันธ์ระหว่างแอกเตอร์กับยูสเคส กล่าวคือ แอกเตอร์มีการสื่อสารแลกเปลี่ยนกับยูสเคส ซึ่งจะใช้เส้นตรงที่บิเชื่อมต่อกันระหว่างสัญลักษณ์แอกเตอร์และสัญลักษณ์ยูสเคส
- **Uses** เป็นความสัมพันธ์ที่อธิบายถึงการเรียกใช้ยูสเคสอื่นร่วมในการทำงาน ด้วยทุกครั้ง กล่าวคือต้องใช้การทำงานของอีกยูสเคสเป็นส่วนหนึ่งของการทำงาน ซึ่งเกิดในกรณีที่หลากหลาย ๆ ยูสเคสมีการทำงานซ้ำกันหรือคล้ายกันแล้ว แยกการทำงานนั้นออกมาเป็นยูสเคสใหม่เพื่อให้ยูสเคสอื่นเรียกใช้ แทนด้วยลูกศร โดยมีคำว่า “use” หรือ “include” กำกับ
- **Extends** เป็นความสัมพันธ์ที่อธิบายถึงการเรียกใช้ยูสเคสอื่นทำงานสำหรับกรณีพิเศษ หรือเหตุการณ์เฉพาะเจาะจง ซึ่งยูสเคสที่ใช้ในกรณีเฉพาะจะมีการทำงานเหมือนยูสเคสเดิมที่มีอยู่แต่จะมีการเพิ่มการทำงานบางอย่างด้วย กล่าวคือเป็นความสัมพันธ์คล้ายการถ่ายทอดคุณสมบัติหรือพฤติกรรมจากซูเปอร์คลาส (Super Class) ไปยังซับคลาส (Sub Class) แทนด้วยลูกศร โดยมีคำว่า “extend” กำกับ



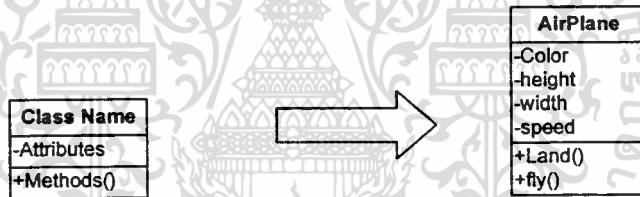
รูปที่ 2.3 แสดงตัวอย่าง Use-case Diagram ในระบบงานห้องสมุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) คลาสไดอะแกรม

คลาสไดอะแกรมเป็นแผนภาพที่อธิบายเกี่ยวกับออบเจกต์ที่เป็นการวิเคราะห์เชิงสแตติกที่สำคัญในรูปแบบของโครงสร้างที่มีส่วนประกอบที่สำคัญคือ คลาส และความสัมพันธ์ของคลาสเหล่านั้น โดยนำมาคลาสมาเชื่อมต่อกันด้วยความสัมพันธ์เพื่ออธิบายระบบในเชิงวัตถุ ซึ่งรายละเอียดที่เป็นข้อมูลแบบไม่คงสภาพหรือมีการเปลี่ยนแปลงจะไม่แสดงในแผนภาพนี้ แต่จะถูกนำไปอธิบายในส่วนของไดอะแกรมอื่นที่เป็นไดนามิกดังจะกล่าวในหัวข้อต่อไป

- คลาส (Classes) เป็นการอธิบายถึงกลุ่มของออบเจกต์ที่มีคุณลักษณะ รูปแบบการทำงาน ความสัมพันธ์ และความหมายบางอย่างร่วมกันหรือเหมือนกัน ซึ่งสามารถแทนคลาสได้ด้วยสัญลักษณ์รูปสี่เหลี่ยมผืนผ้า ดังตัวอย่างในรูปที่ 2.4 ที่มีการแบ่งพื้นที่ออกเป็น 3 ส่วนได้แก่
 - ชื่อคลาส (Class name)
 - คุณสมบัติหรือแอททริบิวต์ (Attributes)
 - พฤติกรรมหรือเมธอด (Methods)

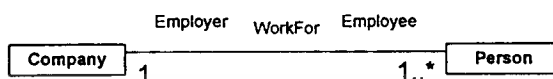


รูปที่ 2.4 ตัวอย่างคลาสรเครื่องบิน

- ความสัมพันธ์ระหว่างคลา (Relationships) สามารถเขียนแทนได้ด้วยเส้นตรงที่บ่งชี้โอกาสการกำหนดคิมตัวเลขแสดงความสัมพันธ์ (Multiplicity) เขียนไว้ที่ด้านปลายของเส้นแสดงความสัมพันธ์ ซึ่งอาจจะระบุได้ในรูปแบบต่าง ๆ ดังนี้
 - เลขจำนวนเดียว ซึ่งเป็นค่าที่แน่นอน
 - ช่วงของตัวเลข (ค่าต่ำสุด..ค่าสูงสุด) ซึ่งเป็นช่วงค่าที่เป็นไปได้
 - สัญลักษณ์ดอกจัน (*) ซึ่งเป็นการไม่ระบุจำนวนหรือช่วงค่าที่เป็นไปได้

นอกจากนี้ความสัมพันธ์ระหว่างคลาสยังสามารถแบ่งได้หลายรูปแบบดังนี้

- **Association** เป็นความสัมพันธ์ระหว่างคลาสทั่วไป ซึ่งอาจเป็นแบบสองทางหรือมากกว่า (Binary or N-ary association) ดังตัวอย่างในรูปที่ 2.5 ซึ่งหน้าที่หรือบทบาท (Role) ของแต่ละคลาสจะถูกกำหนดไว้ที่ด้านปลายของเส้นความสัมพันธ์ที่อยู่ติดกับคลาสนั้น ๆ และชื่อของความสัมพันธ์จะมีความหมายจากซ้ายไปขวา



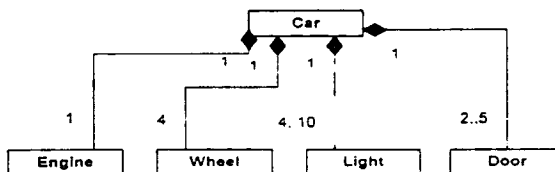
รูปที่ 2.5 ตัวอย่างความสัมพันธ์แบบ Association

- **Aggregation** เป็นความสัมพันธ์ที่แสดงให้เห็นถึงคลาสหนึ่งเป็นส่วนหนึ่งของอีกคลาสหนึ่ง (Whole-Part) ซึ่งเป็นความสัมพันธ์แบบรวมกลุ่มกันรูปแบบหนึ่ง ซึ่งใช้สัญลักษณ์ที่มีหัวเป็นรูปสี่เหลี่ยมขนมเปียกปูน โปร่งในด้านปลายที่เป็นคลาสส่วนเจ้าของ (Whole) และใช้ลูกศรปลายเปิดในด้านปลายที่เป็นคลาสที่เป็นส่วนประกอบของส่วนเจ้าของ (Part) ดังตัวอย่างในรูปที่ 2.6



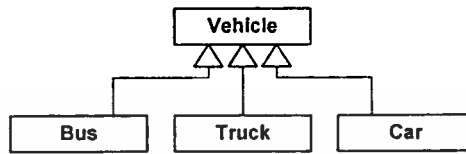
รูปที่ 2.6 ตัวอย่างความสัมพันธ์แบบ Aggregation

- **Composition Aggregation** เป็นความสัมพันธ์แบบรวมกลุ่มกันรูปแบบหนึ่งเช่นกัน แต่หากว่าไม่มีคลาสทางด้านส่วนเจ้าของ (Whole) แล้วจะไม่สามารถมีคลาสด้านที่เป็นส่วนประกอบ (Part) ได้ ซึ่งจะใช้สัญลักษณ์รูปสี่เหลี่ยมขนมเปียกปูนทึบในด้านที่เป็นคลาสส่วนเจ้าของ ดังรูปที่ 2.7



รูปที่ 2.7 ตัวอย่างความสัมพันธ์แบบ Composition

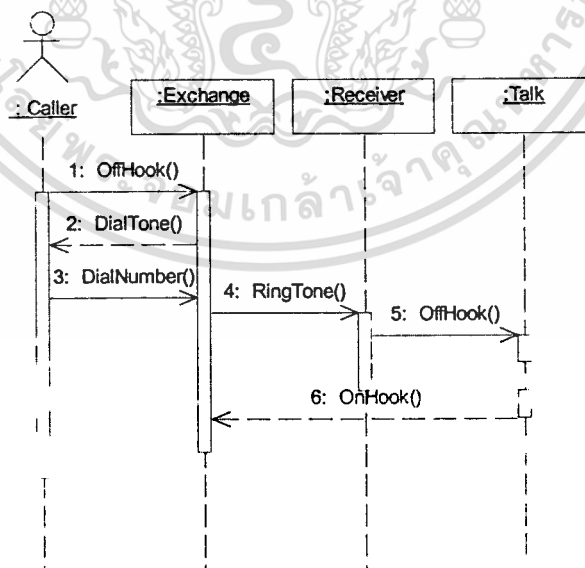
- **Generalization** เป็นความสัมพันธ์ระหว่างคลาสทั่วไปหรือซูเปอร์คลาส (Super Class) กับคลาสที่มีความเฉพาะมากกว่าหรือซบคลาส (Sub Class) ซึ่งจะได้รับการถ่ายทอดคุณสมบัติและพฤติกรรมจากซูเปอร์คลาส และอาจเรียกได้ว่าเป็นความสัมพันธ์แบบ “is-a” โดยสัญลักษณ์ที่ใช้จะเป็นรูปลูกศรปลายปิด โปรงและด้านที่ปลายลูกศรชี้อยู่จะเป็นซูเปอร์คลาสส่วนอีกด้านคือซบคลาสนั่นเอง ดังตัวอย่างรูปที่ 2.8



รูปที่ 2.8 ตัวอย่างความสัมพันธ์แบบ Generalization

3) ซีคว็่นซ์ไดอะแกรม

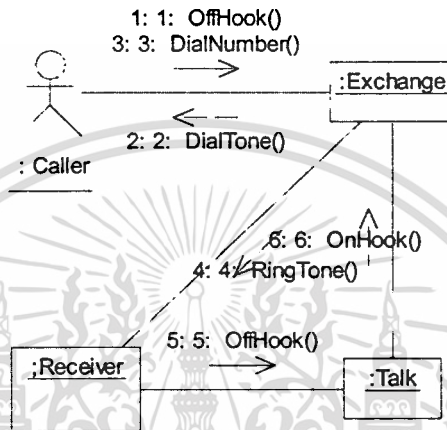
ซีคว็่นซ์ไดอะแกรมเป็นแผนภาพแสดงการติดต่อในการทำงานระหว่างออบเจ็กต์ในระบบตามลำดับของเวลา (Time Sequence) ไดอะแกรมนี้จะมี 2 มิติ คือ ในแนวตั้งแสดงลำดับเวลา และแนวนอนแสดงออบเจ็กต์ที่แตกต่างกัน และเส้นตามแนวตั้งจะเรียกว่าเส้นชีวิต (Life Line) ซึ่งจะบอกว่าออบเจ็กต์ที่ปรากฏขึ้นในการทำกิจกรรมกับออบเจ็กต์อื่นตามลำดับก่อนหลังอย่างไร และจะมีเส้นลูกศรในแนวนอนที่เชื่อมระหว่างเส้นชีวิตเพื่อแสดงการส่งผ่านข้อความที่ติดต่อกัน ดังรูปที่ 2.9



รูปที่ 2.9 ตัวอย่างซีคว็่นซ์ไดอะแกรม

4) คอลลาบอเรชันไดอะแกรม

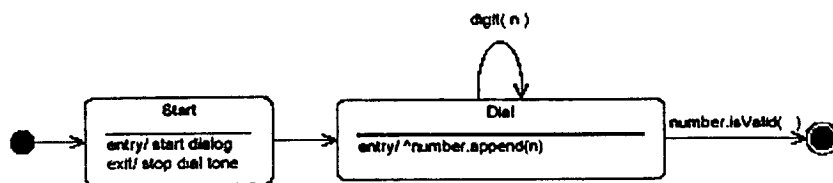
คอลลาบอเรชันไดอะแกรมเป็นแผนภาพแสดงการทำงานประสานกันระหว่างออบเจ็กต์ ซึ่งสอดคล้องกับซีควีนซ์ไดอะแกรม แต่สำหรับซีควีนซ์ไดอะแกรมจะเน้นที่ลำดับเวลา แต่ในคอลลาบอเรชันไดอะแกรมจะเน้นที่การทำงานร่วมกันระหว่างออบเจ็กต์ ตัวอย่างคอลลาบอเรชันไดอะแกรมดังรูปที่ 2.10



รูปที่ 2.10 ตัวอย่างคอลลาบอเรชันไดอะแกรม

5) สเตตชาร์ตไดอะแกรม

สเตตชาร์ตไดอะแกรมเป็นแผนภาพที่ใช้แสดงการเปลี่ยนสถานะภาพ (State) ของออบเจ็กต์ ตั้งแต่เริ่มต้นจนถึงสิ้นสุดการเปลี่ยนแปลงในแต่ละรอบ (1 Sequence) เพื่อช่วยให้เข้าใจอัลกอริทึมที่เกี่ยวข้องกับการทำงานในแต่ละเมธอดของคลาส สถานะภาพ (State) เป็นกลุ่มของค่าที่ซึ่งใช้อธิบายออบเจ็กต์ ณ เวลาหนึ่ง ๆ ซึ่งแทนด้วยสัญลักษณ์รูปสี่เหลี่ยมขอบมน และจะมีเส้นแสดงการเปลี่ยนสถานะภาพหรือมีเหตุการณ์เกิดขึ้นในการเชื่อมโยงระหว่างสเตตซึ่งแทนด้วยลูกศรที่เชื่อมต่อระหว่างสัญลักษณ์ของสเตต นอกจากนี้ยังในแผนภาพจะมีจุดเริ่มต้นซึ่งแทนด้วยสัญลักษณ์วงกลมทึบ และจุดสิ้นสุดแทนด้วยสัญลักษณ์วงกลมทึบที่มีวงกลมโปร่งล้อมรอบ ดังตัวอย่างรูปที่ 2.11



รูปที่ 2.11 ตัวอย่างสเตตชาร์ตไดอะแกรม

6) แอคทิวิตีไดอะแกรม

แอกทิวิตีไดอะแกรมเป็นแผนภาพแสดงขั้นตอนการกระทำ (Action) และกิจกรรม (Activity) รวมถึงการสร้างการดำเนินการ และผลลัพธ์ของการกระทำนั้น ๆ ในรูปแบบที่คล้ายกับโฟลว์ชาร์ต ซึ่งมีส่วนคล้ายกับสเตตชาร์ตไดอะแกรมแต่ต่างกันตรงที่ แอกทิวิตีไดอะแกรมเน้นไปที่การกระทำ และผลลัพธ์ ส่วนสเตตชาร์ตไดอะแกรมจะเน้นที่การเปลี่ยนแปลงสถานะของออบเจกต์เมื่อมีเหตุการณ์ต่าง ๆ เข้ามา สำหรับแอกทิวิตีไดอะแกรมจะมีสัญลักษณ์ คล้ายกับสเตตชาร์ตไดอะแกรม แต่เพิ่มสัญลักษณ์ของการตัดสินใจ (Decision) เข้ามาซึ่งเป็นรูปสี่เหลี่ยมขนมเปียกปูน และอาจมีการแสดงการทำงานที่ทำพร้อมกันหลาย ๆ งาน โดยจะใช้เส้นตรงแนวนอนเส้นหนาในการจัดกลุ่มงานที่มีการทำพร้อม ๆ กัน นอกจากนี้ยังสามารถแสดงให้เห็นการแบ่งงานตามผู้รับผิดชอบได้ด้วยการแบ่งได้โดยใช้สวิมเลน (Swimlanes) ตัวอย่างแอกทิวิตีไดอะแกรมแสดงในรูปที่ 2.12

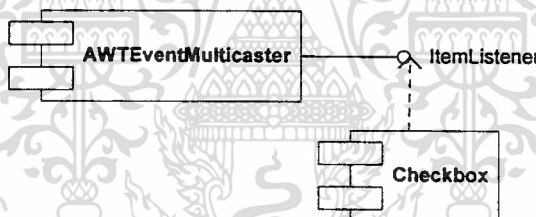


รูปที่ 2.12 ตัวอย่างแอกทิวิตีไดอะแกรม

7) คอมโพเนนต์ไคอะแกรม

คอมโพเนนต์ไคอะแกรมเป็นแผนภาพอธิบายองค์ประกอบทางกายภาพของการออกแบบ เช่น โค้ดต้นฉบับ (Source Code) โปรแกรมที่ใช้งาน (Executable Program) หรือการติดต่อกับผู้ใช้ (User Interface) เป็นต้น สำหรับคอมโพเนนต์ตาม UML ได้แบ่งตามลักษณะการใช้งานออกเป็น 3 ประเภทได้แก่

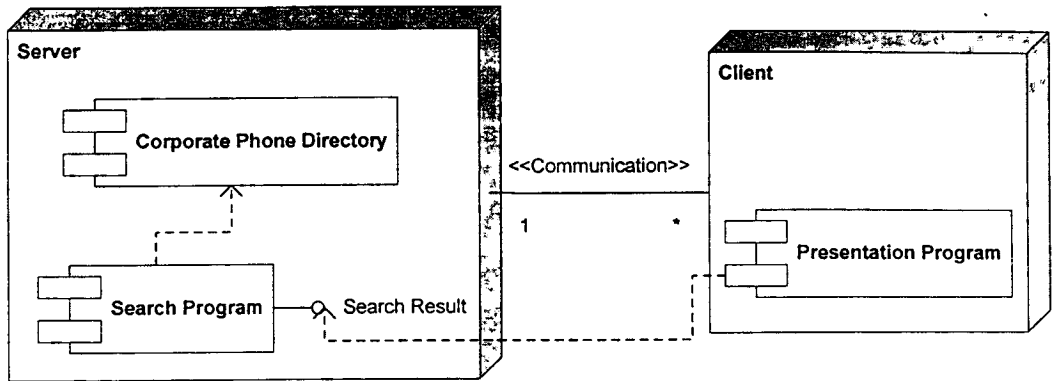
- Deployment components ซึ่งเป็นคอมโพเนนต์ของระบบในส่วนที่ทำหน้าที่ทำงานจริง ๆ เช่น DLL, executable, ActiveX controls, Java Beans เป็นต้น
 - Work product components ซึ่งเป็นคอมโพเนนต์ที่อธิบายว่า Deployment components นั้นได้มาจากที่ไหนบ้าง เช่น ไฟล์ข้อมูล, ไฟล์ซอร์สโค้ด เป็นต้น
 - Execution components ซึ่งเป็นคอมโพเนนต์ที่เป็นผลลัพธ์ของระบบที่กำลังทำงานอยู่
- องค์ประกอบหลักของคอมโพเนนต์ไคอะแกรมได้แก่ คอมโพเนนต์ (Components), อินเทอร์เฟซ (Interface) และ ความสัมพันธ์ (Relationships) โดยสัญลักษณ์ที่ใช้แทนคอมโพเนนต์คือรูปสี่เหลี่ยมผืนผ้าที่มีสี่เหลี่ยมผืนผ้าขนาดเล็ก ๆ 2 รูปอยู่ด้านซ้ายมือและมีชื่อคอมโพเนนต์อยู่ในสี่เหลี่ยมรูปใหญ่ ตัวอย่างดังรูปที่ 2.13



รูปที่ 2.13 ตัวอย่างคอมโพเนนต์ไคอะแกรม

8) ดิพลอยเมนต์ไคอะแกรม

ดิพลอยเมนต์ไคอะแกรมเป็นแผนภาพแสดงการสถาปนาปัตยกรรมของระบบขณะที่โปรแกรมทำงาน (Run - Time) และโปรโตคอล (Protocol) ที่เชื่อมต่อระหว่างโหนด (Node) ซึ่งได้แก่ โปรเซสเซอร์ อุปกรณ์ และ ซอฟต์แวร์ต่าง ๆ ไว้ด้วยกัน สัญลักษณ์ที่ใช้แทนโหนดคือกล่องสี่เหลี่ยม และใช้เส้นทึบลากเชื่อมระหว่างโหนดพร้อมเขียนกำกับบอกถึงโปรโตคอลที่ใช้แสดงถึงการเชื่อมต่อ ดังตัวอย่างรูปที่ 2.14



รูปที่ 2.14 ตัวอย่างดีพลอยเมตไดอะแกรม

2.3.2 การวิเคราะห์เชิงวัตถุ (Object Oriented Analysis) (Bahrami . 1999)

หลักการวิเคราะห์เชิงวัตถุจะใช้แอกเตอร์และยูสเคสในการอธิบายระบบงานในมุมมองของผู้ใช้ โดยแอกเตอร์เป็นปัจจัยภายนอกที่ติดต่อกับระบบและยูสเคสเป็นเหตุการณ์ที่อธิบายว่าแอกเตอร์ใช้ระบบอย่างไร และหลังจากนั้นจึงวิเคราะห์ระบบเพื่อจัดกลุ่มออบเจกต์ที่มีความเกี่ยวข้องกันเป็นกลุ่มเดียวกันเพื่อให้เกิดเป็นคลาสของกลุ่มออบเจกต์

กระบวนการในการวิเคราะห์เชิงวัตถุประกอบด้วยขั้นตอนดังนี้

1) สร้างยูสเคสไดอะแกรม โดย

- ระบุแอกเตอร์ซึ่งอาจเป็นผู้ใช้หรือผู้เชื่อมต่อกับระบบ
- สร้างภาพรวมของระบบในเชิงผู้ใช้ด้วยการวาดแอกทิวิตีไดอะแกรม
- สร้างยูสเคส โดยการพิจารณาจากการกระทำของผู้ใช้ที่มีต่อระบบ
- สร้างเป็นยูสเคสไดอะแกรมซึ่งแสดงถึงความสัมพันธ์ระหว่างแอกเตอร์กับยูสเคสต่าง ๆ

2) สร้างไดอะแกรมเชิงพฤติกรรมระหว่างกลุ่มออบเจกต์ โดย

- จัดกลุ่มออบเจกต์ที่เกี่ยวข้องกับระบบ โดยใช้รูปแบบ (Pattern) ต่าง ๆ เป็นเกณฑ์การพิจารณา ซึ่งกลุ่มที่ได้จะถูกเรียกว่าคลาส
- ระบุพฤติกรรม หน้าที่ความรับผิดชอบ และการทำงานร่วมกันของแต่ละคลาส
- สร้างซีควีนซ์ไดอะแกรมหรือคอลลาบอเรชันไดอะแกรม สำหรับอธิบายพฤติกรรม ความรับผิดชอบ และการทำงานร่วมกันของแต่ละคลาส

3) สร้างคลาสไคอะแกรม

- ระบุความสัมพันธ์ (Association) แอททริบิวต์ (Attributes) และ เมธอด (Methods) ของแต่ละคลาสโดยวิเคราะห์ข้อมูลจากยูสเคสไคอะแกรมหรือไคอะแกรมอื่น ๆ
- สร้างคลาสไคอะแกรมสำหรับอธิบายถึงความสัมพันธ์ แอททริบิวต์ และเมธอดของแต่ละคลาสที่ได้จากการวิเคราะห์

2.3.3 การออกแบบเชิงวัตถุ (Object Oriented Design) (Bahrami . 1999)

การออกแบบเชิงวัตถุจะเป็นการปรับปรุงคลาสที่ได้จากขั้นตอนการวิเคราะห์ระบบเพื่อให้เหมาะสำหรับการนำไปพัฒนาเป็นโปรแกรม โดยอาจมีการสร้างคลาส แอททริบิวต์ หรือเมธอดเพิ่มเติมจากเดิม นอกจากนี้ยังรวมถึงการออกแบบส่วนติดต่อกับผู้ใช้ (View Layer) และส่วนเชื่อมต่อกับข้อมูล (Access Layer)

กระบวนการในการออกแบบเชิงวัตถุประกอบด้วยขั้นตอนดังนี้

1) ปรับปรุงคลาสไคอะแกรม

- ระบุระดับการติดต่อกับสาธารณะระหว่างคลาส เช่น Public, Private และ Protected เป็นต้น
- ปรับและแก้ไขแอททริบิวต์ของแต่ละคลาสจากข้อมูลที่ได้จากการวิเคราะห์
- ออกแบบเมธอดจากไคอะแกรมเชิงพฤติกรรมจากการวิเคราะห์โดยระบุรายละเอียดให้ชัดเจนยิ่งขึ้น ซึ่งรวมถึงการกำหนดระดับการเข้าถึงเมธอดแต่ละเมธอดด้วย
- จัดกลุ่มคลาสเข้าอยู่ในแพ็คเกจ (Package) และปรับคลาสไคอะแกรมให้ละเอียดขึ้น

2) ออกแบบส่วนเชื่อมต่อกับข้อมูล

- ออกแบบฐานข้อมูลของระบบจากข้อมูลที่ผ่านมาขั้นตอนวิเคราะห์ ซึ่งอาจเลือกใช้โมเดลฐานข้อมูลได้ต่าง ๆ กันไป เช่น Relational Model หรือ Network Model เป็นต้น
- ออกแบบคลาสที่ใช้เชื่อมต่อกับฐานข้อมูล
- ปรับปรุงคลาสไคอะแกรมโดยเพิ่มส่วนเชื่อมต่อกับข้อมูลเข้าไป

3) ออกแบบส่วนติดต่อกับผู้ใช้

- ระบุคลาสที่ใช้ติดต่อกับผู้ใช้โดยวิเคราะห์จากยูสเคสต่างๆ
- ออกแบบคลาสที่ใช้ติดต่อกับผู้ใช้ให้เหมาะสมกับการใช้งานของผู้ใช้
- ปรับปรุงคลาสไดอะแกรมโดยเพิ่มส่วนติดต่อกับผู้ใช้เข้าไป
- สร้างส่วนติดต่อกับผู้ใช้ต้นแบบเพื่อทดลองใช้ ซึ่งอาจมีการปรับเปลี่ยนได้



บทที่ 3

การศึกษาและวิเคราะห์ระบบ

3.1 การศึกษาหลักการทำงานของยูนิเวอร์แซลเทสต์ไดเรกทอรี

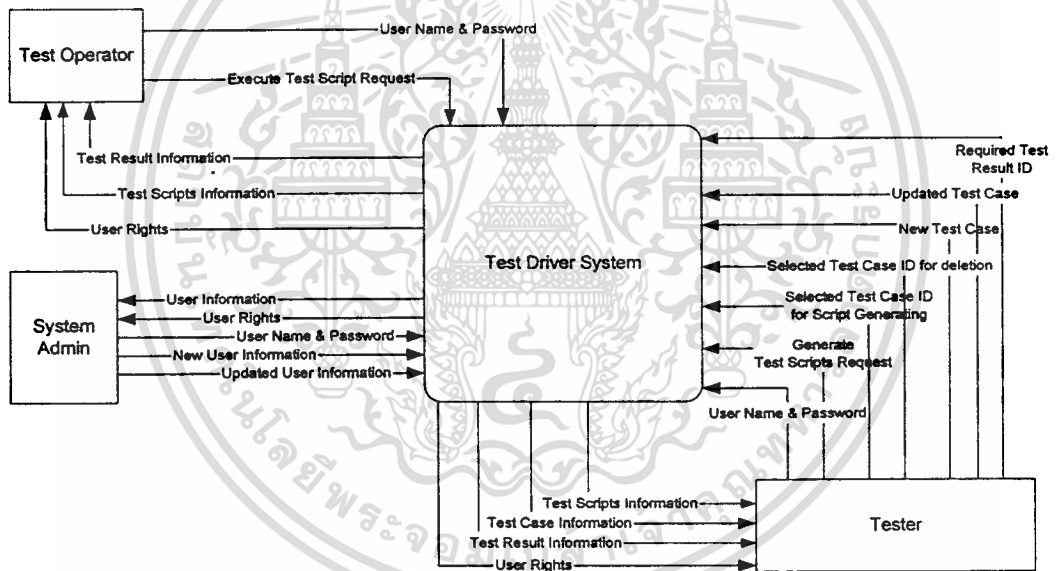
การทดสอบซอฟต์แวร์โดยทั่วไปมีหลักการทำงานคือ ทำการตั้งค่าต่าง ๆ ที่เกี่ยวกับการทดสอบ เช่น ค่าข้อมูลเข้าที่เป็น Test case จากนั้นจึงทำการเรียกซอฟต์แวร์หรือ โปรแกรมที่ต้องการทดสอบขึ้นมาประมวลผลตามค่าที่ได้ตั้งไว้เพื่อทดสอบให้ได้ผลลัพธ์ค่าหนึ่ง ๆ ที่ต่อไปจะนำไปประเมินว่าถูกต้องหรือไม่ ซึ่งเมื่อได้ทำการทดสอบเสร็จแล้วจะมีการล้างค่าที่ได้ตั้งไว้ในตอนแรกออกไปเพื่อให้ซอฟต์แวร์กลับสู่สถานะปกติ การทดสอบซอฟต์แวร์แบบนี้จะไม่มีการเก็บค่าต่าง ๆ ที่ตั้งไว้เพื่อใช้ในครั้งต่อไป ทำให้หากผู้ทดสอบซอฟต์แวร์ต้องทดสอบซอฟต์แวร์หลาย ๆ ครั้งหรือทดสอบใหม่อีกครั้งเมื่อมีการปรับปรุงซอฟต์แวร์ให้ดีขึ้นเพื่อให้มั่นใจว่าซอฟต์แวร์ดังกล่าวได้แก้ปัญหาที่พบในครั้งก่อนแล้ว ที่เรียกว่าการทำทดสอบแบบ Regression ซึ่งต้องเสียเวลาในการตั้งค่าใหม่ทุก ๆ ครั้งของการทดสอบ และเป็นการทำงานที่ไม่มีประสิทธิภาพสำหรับผู้ทดสอบ

ดังนั้นการทดสอบแบบอัตโนมัติซึ่งมีการเก็บบันทึกค่าเริ่มต้นของการทดสอบไว้เพื่อใช้งานครั้งต่อไปจึงทำให้ผู้ทดสอบทำงานได้อย่างมีประสิทธิภาพมากขึ้น โดยการบันทึกนั้นจะเก็บบันทึกเป็นประเภทไฟล์ข้อความ (Text Files) ที่จัดเป็นการเก็บข้อมูลแบบถาวรจึงทำให้สามารถทดสอบได้ทันทีแม้ว่าจะมีการเปลี่ยนผู้ทดสอบหรือไม่ก็ตามเพราะไม่จำเป็นต้องให้ผู้ทดสอบสร้างค่าเริ่มต้นสำหรับการทดสอบใหม่แต่สามารถอ่านจากไฟล์ที่บันทึกไว้ได้เลย ไฟล์ดังกล่าวอาจเรียกอีกอย่างว่า Script ไฟล์สำหรับการทดสอบซึ่งมีการเขียนได้หลายรูปแบบขึ้นกับผู้ออกแบบคิดค้น ซึ่งการสร้าง Script จากเอกสาร XML ก็เป็นแนวทางหนึ่งที่นำข้อดีของ XML มาประยุกต์ใช้ นั่นคือเป็นเอกสารกลางที่ใช้สื่อสารได้กับทุกระบบ ดังนั้นหากต้องทำการสร้าง Script ขึ้นจากข้อมูลที่มาจากหลายที่ก็สามารถใช้ XML เป็นเอกสารกลางในการสื่อสารและเตรียมข้อมูลได้

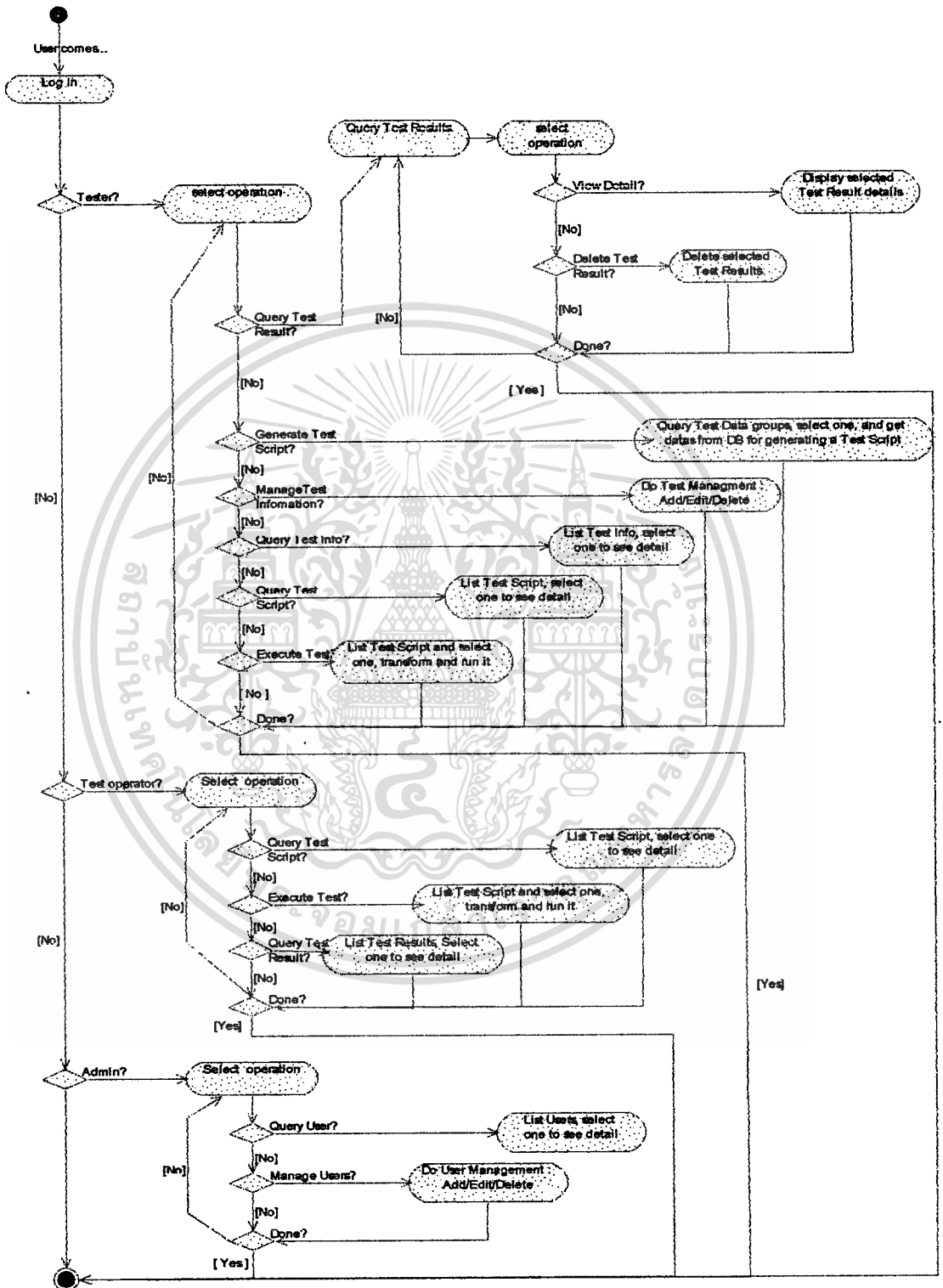
สำหรับการพัฒนาระบบบนพื้นฐานของการทำงานยูนิเวอร์แซลเทสต์ไดเรกทอรีซึ่งเป็นหลักการทำงานที่สนับสนุนการทดสอบบนพื้นฐานของการทำเอกสาร Script จากเอกสาร XML นี้จะมีการทำงานในรูปแบบกึ่งอัตโนมัติ นั่นคือมีการสร้าง Script ของการทดสอบเก็บไว้แต่ยังไม่รวมถึงการเรียกใช้ Script ดังกล่าวแบบอัตโนมัติ โดยการสร้าง Script ของการทดสอบจะนำข้อมูลจาก

ฐานข้อมูลกลางที่เก็บบันทึกรายละเอียดและข้อกำหนดต่าง ๆ ในการทดสอบออกมาเป็นเอกสาร XML ก่อนในขั้นต้นจากนั้นจึงนำไปแปลงเป็น Script รูปแบบที่ต้องการด้วยตัวช่วยแปลงเอกสาร XML ส่วนการบันทึกผลของการทดสอบซอฟต์แวร์หลังจากการทดสอบสำหรับการเปรียบเทียบในการทดสอบจะบันทึกไว้ในรูปแบบเอกสาร XML ซึ่งสามารถใช้แสดงผลผ่าน Web Browser ได้

กระบวนการทำงานของยูนิเวอร์แซลเทสตีโคโนโลยีได้กล่าวไว้ในบทที่ 2 หัวข้อที่ 2.2 ซึ่งจะอธิบายถึงองค์ประกอบที่สำคัญและกระบวนการทำงาน ภาพรวมของการทำงานของระบบที่พัฒนาบนหลักการการทำงานของยูนิเวอร์แซลเทสตีโคโนโลยีแสดงได้ในไดอะแกรมรูปที่ 3.1 และแอกทิวิตี้ไดอะแกรมรูปที่ 3.2 และจะนำหลักการดำเนินงานนี้มาวิเคราะห์และออกแบบเพื่อพัฒนาเป็นระบบงานซึ่งจะใช้แนวคิดเชิงวัตถุและการอธิบายด้วยแผนภาพต่าง ๆ ของ UML โดยจะนำเสนอในบทที่ 4 ต่อไป



รูปที่ 3.1 แผนผังแสดงภาพรวมของการทำงานของระบบในมุมมองของการไหลของข้อมูล (Context Diagram)



รูปที่ 3.2 แอททิวต์ไอคอนแอมแสดงการทำงาน โดยรวมของระบบที่พัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ยูสเคสไดอะแกรม (สุนทริน วงศ์ศิริกุล. ม.ป.ป.)

3.2.1 ระบุแอกเตอร์และยูสเคส

สำหรับ Actor และ Use case ที่จำแนกตาม Actor ที่กระทำ Use case นั้น ๆ ซึ่งได้พิจารณาเพื่อใช้สำหรับระบบที่พัฒนาในโครงการดังนี้

- **เจ้าหน้าที่ทดสอบซอฟต์แวร์ (Test Operator)**
 - 1) เข้าใช้ระบบและตรวจสอบสิทธิ์ (Authenticate)
 - 2) เรียกดูสคริปต์สำหรับการทดสอบ (Query Test Scripts)
 - แสดงรายการสคริปต์สำหรับการทดสอบ (List Test Script)
 - แสดงรายละเอียดของสคริปต์สำหรับการทดสอบ (View Test Script)
 - 3) เรียกสคริปต์ทำการทดสอบ โปรแกรม (Execute Test Script)
 - 4) เรียกดูรายงานผลการทดสอบที่ได้จากการทดสอบด้วยสคริปต์ (Query Test Results)
- **ผู้ควบคุมการทดสอบ (Tester)**
 - 1) เข้าใช้ระบบและตรวจสอบสิทธิ์ (Authenticate)
 - 2) เรียกดูข้อมูลสำหรับการทดสอบ (Query Test Info)
 - แสดงรายการข้อมูลสำหรับการทดสอบ (List Test Info)
 - แสดงรายละเอียดของข้อมูลสำหรับการทดสอบ (View Test Info)
 - 3) จัดการเกี่ยวกับข้อมูลสำหรับการทดสอบ (Manage Test Info)
 - สร้างข้อมูลสำหรับการทดสอบ (Create Test Info)
 - ปรับเปลี่ยนข้อมูลสำหรับการทดสอบ (Edit Test Info)
 - ลบข้อมูลสำหรับการทดสอบ (Delete Test Info)
 - 4) สร้างสคริปต์สำหรับการทดสอบ (Generate Test Script)
 - 5) เรียกดูรายละเอียดในสคริปต์สำหรับการทดสอบ (Query Test Scripts)
 - แสดงรายการสคริปต์สำหรับการทดสอบ (List Test Script)
 - แสดงรายละเอียดของสคริปต์สำหรับการทดสอบ (View Test Script)
 - 6) จัดการเกี่ยวกับสคริปต์สำหรับการทดสอบ (Manage Test Scripts)
 - ลบสคริปต์สำหรับการทดสอบ (Delete Test Scripts)
 - 7) เรียกสคริปต์ทำการทดสอบ โปรแกรม (Execute Test Script)
 - 8) เรียกดูผลการทดสอบที่ได้จากการทดสอบด้วยสคริปต์ (Query Test Results)

- แสดงรายการผลการทดสอบที่ได้จากการทดสอบด้วยสคริปต์ (List Test Results)
 - แสดงรายละเอียดของผลการทดสอบที่ได้จากการทดสอบด้วยสคริปต์ (View Test Result)
- 9) จัดการเกี่ยวกับผลการทดสอบที่ได้จากการทดสอบด้วยสคริปต์(Manage Test Results)
- ลบผลการทดสอบ (Delete Test Results)

■ ผู้ดูแลระบบ (System Admin)

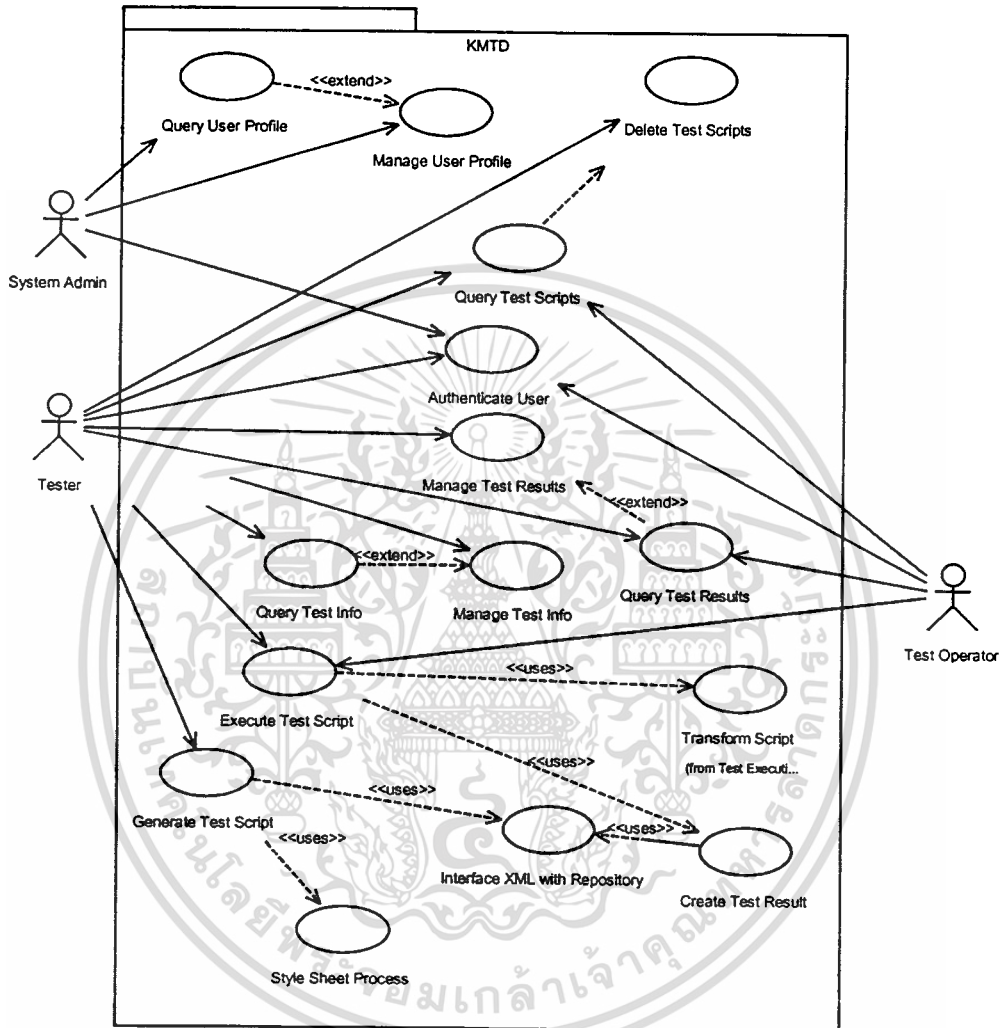
- 1) เข้าสู่ระบบและตรวจสอบสิทธิ์ (Authenticate)
- 2) จัดการเกี่ยวกับข้อมูลผู้ใช้งานระบบ (Manage User Profile)
 - สร้างข้อมูลผู้ใช้งานระบบ (Create User Profile)
 - แก้ไขข้อมูลผู้ใช้งานระบบ (Edit User Profile)
 - ลบข้อมูลผู้ใช้งานระบบ (Delete User Profile)
- 3) เรียกดูข้อมูลผู้ใช้งานระบบ (Query User Profile)
 - แสดงรายการผู้ใช้งานระบบ (List User Profile)
 - แสดงรายละเอียดข้อมูลผู้ใช้งานระบบ (View User Profile)

นอกจากยูสเคสที่ได้แสดงข้างต้นยังมียูสเคสที่ภายในซึ่งไม่ได้ถูกกระทำด้วย Actor แต่เป็นยูสเคสที่มีความสัมพันธ์กับยูสเคสข้างต้นแบบ extend และ use ได้แก่

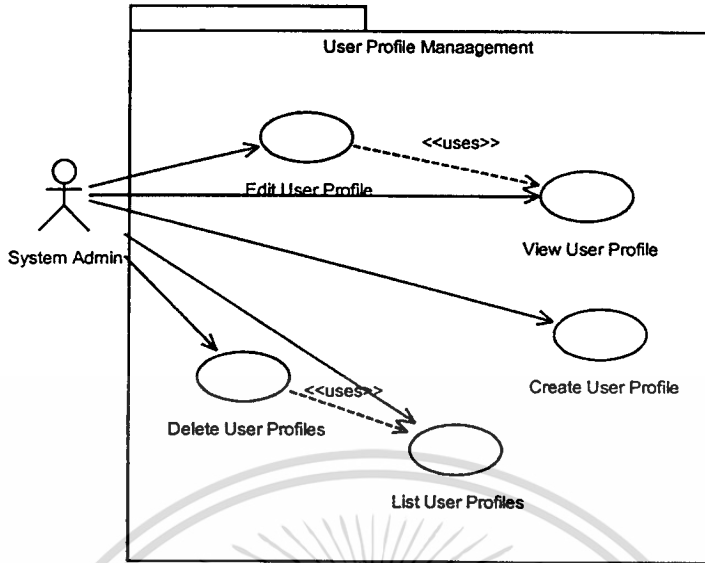
- 1) การเชื่อมต่อกับฐานข้อมูลด้วยเอกสาร XML (Interface XML with Repository)
- 2) การประมวลผล Style Sheet (Style Sheet Process)
- 3) การสร้างสคริปต์ส่วน Utilities (Build Interface Utilities)
- 4) การแปลงสคริปต์เพื่อใช้เรียกประมวลผล (Script Transformation)
- 5) การสร้างผลการทดสอบ (Create Test Results)

3.2.2 ยูสเคสไดอะแกรม (Use-Case Diagram)

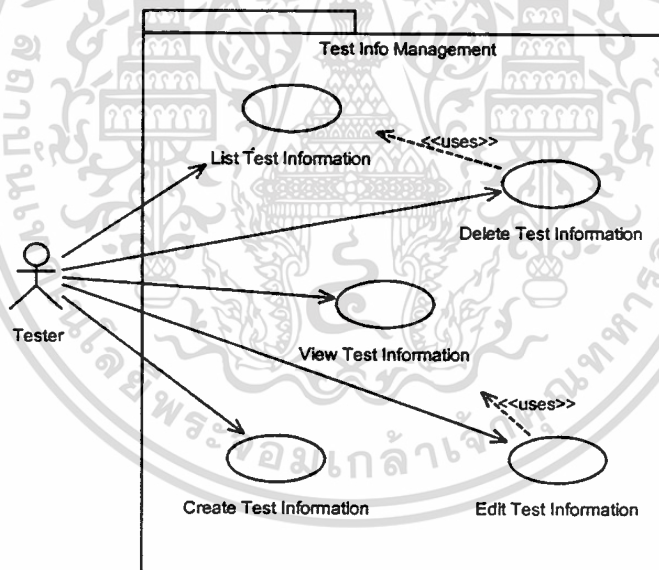
ยูสเคสไดอะแกรมของระบบที่พัฒนามาหลักการของยูนิเวอร์แซลเทสเคสไดอะแกรมที่ได้ตั้งรูปที่ 3.3, 3.4, 3.5 และ 3.6 ซึ่งประกอบด้วยยูสเคสต่าง ๆ ที่ได้กล่าวไว้ในหัวข้อที่แล้ว



รูปที่ 3.3 ยูสเคสไดอะแกรมหลักของระบบที่พัฒนา

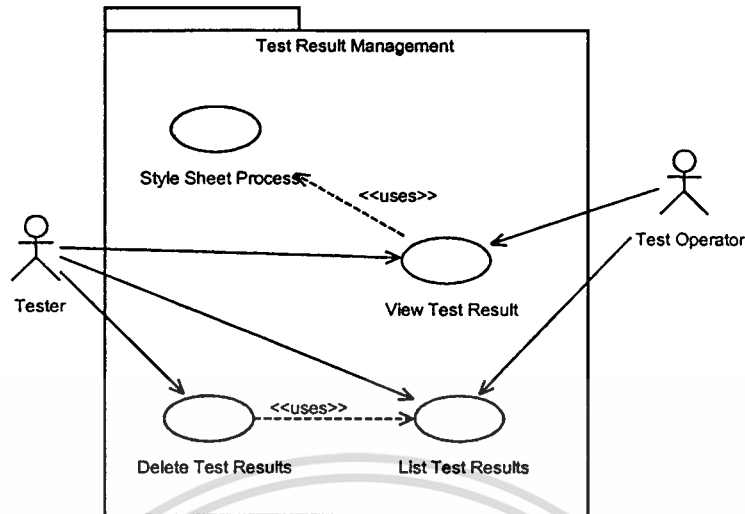


รูปที่ 3.4 ยูสเคสไดอะแกรมย่อยเกี่ยวกับการจัดการข้อมูลผู้ใช้งานระบบ (Manage User Profile)



รูปที่ 3.5 ยูสเคสไดอะแกรมย่อยเกี่ยวกับการจัดการข้อมูลสำหรับการทดสอบ (Manage Test Info)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 ยูสเคสไดอะแกรมย่อยเกี่ยวกับการจัดการข้อมูลรายงานผลการทดสอบที่ได้จากการทดสอบด้วยสคริปต์ (Manage Test Results)

3.2.3 การวิเคราะห์ลำดับการทำงานจากยูสเคส (Bahrami . 1999)

เป็นการวิเคราะห์ลำดับขั้นตอนการดำเนินงานภายในยูสเคสว่ามีลำดับเหตุการณ์เป็นอย่างไรหรือมีเงื่อนไขต่าง ๆ ที่เกี่ยวข้องอะไรบ้าง สามารถอธิบายได้ด้วยตารางดังต่อไปนี้

ตารางที่ 3.1 ยูสเคสเกี่ยวกับการสร้างข้อมูลผู้ใช้ระบบ

USE CASE NAME:	Create User Profile	
ACTOR(S):	System Admin	
DESCRIPTION:	ยูสเคสนี้เป็นการสร้างข้อมูลของผู้ใช้ระบบ	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	Step 1: System Admin ส่งข้อมูล User Profile	Step 2: ระบบ User Profile ใหม่ในฐานข้อมูล
	Step 4: ยูสเคสเสร็จสิ้นเมื่อ	Step 3: แสดงในรายการ User Profile

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ได้รับสิทธิใช้งานจากระบบ
ALTERNATE COURSES:	Step 2: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ
PRE-CONDITION:	ไม่มี
POST-CONDITION	ผู้ใช้ระบบได้รับสิทธิการใช้งาน
ASSUMPTIONS:	ไม่มี

ตารางที่ 3.2 ยูสเคสเกี่ยวกับการลบข้อมูลผู้ใช้ระบบ

USE CASE NAME:	Delete User Profile	
ACTOR(S):	System Admin	
DESCRIPTION:	ยูสเคสนี้เป็นการลบข้อมูลของผู้ใช้ระบบ	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: System Admin ทำการร้องขอทำการลบข้อมูลของผู้ใช้ระบบ Step 4: ยูสเคสเสร็จสิ้นเมื่อได้ทำการลบข้อมูลผู้ใช้เรียบร้อยแล้ว	System Response Step 2: ระบบทำการลบข้อมูลของผู้ใช้ที่ต้องการลบออกจากฐานข้อมูล Step 3: แสดงผลออกมาทางรายการ User Profile
ALTERNATE COURSES:	Step 2: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	User Profile ที่ต้องการลบถูกลบ	
ASSUMPTIONS:	มีข้อมูลผู้ใช้ในระบบ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 ยูสเคสเกี่ยวกับการแก้ไขข้อมูลผู้ใช้ระบบ

USE CASE NAME:	Edit User Profile	
ACTOR(S):	System Admin	
DESCRIPTION:	ยูสเคสนี้เป็นการแก้ไขข้อมูลของผู้ใช้ระบบ	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: System Admin ร้องขอทำการแก้ไขข้อมูลของผู้ใช้ระบบ Step 4: ยูสเคสเสร็จสิ้นเมื่อได้ทำการแก้ไขข้อมูลผู้ใช้เรียบร้อยแล้ว	System Response Step 2: ระบบทำการแก้ไขข้อมูลของผู้ใช้ระบบ Step 3: แสดงผลออกมาทางรายการ User Profile เมื่อทำการแก้ไขเรียบร้อยแล้ว
ALTERNATE COURSES:	Step 2: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	User Profile ที่ต้องการ ได้ทำการแก้ไขเรียบร้อยแล้ว	
ASSUMPTIONS:	มีข้อมูลผู้ใช้อยู่ในระบบ	

ตารางที่ 3.4 ยูสเคสเกี่ยวกับแสดงรายการข้อมูลผู้ใช้ระบบ

USE CASE NAME:	List User Profile	
ACTOR(S):	System Admin	
DESCRIPTION:	ยูสเคสนี้เป็นการเรียกดูข้อมูลของผู้ใช้ระบบแบบรายการ	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: System Admin ร้องขอทำการเรียกดูข้อมูลของ	System Response

	ผู้ใช้ระบบ Step 4: ยูสเคสเสร็จสิ้นเมื่อได้ทำการแสดงรายการเรียบร้อยแล้ว	Step 2: ระบบทำการค้นหาตามเงื่อนไขที่กำหนด Step 3: แสดงผลออกมาทางรายการ User Profile
ALTERNATE COURSES:	Step 2: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	User Profile ที่ต้องการ ได้ทำการแสดงเรียบร้อยแล้ว	
ASSUMPTIONS:	ไม่มี	

ตารางที่ 3.5 ยูสเคสเกี่ยวกับแสดงข้อมูลผู้ใช้ระบบ

USE CASE NAME:	View User Profile	
ACTOR(S):	System Admin	
DESCRIPTION:	ยูสเคสนี้เป็นการเรียกดูข้อมูลของผู้ใช้ระบบตาม ID	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: System Admin ร้องขอทำการเรียกดูข้อมูลของผู้ใช้ระบบ Step 4: ยูสเคสเสร็จสิ้นเมื่อได้ทำการแสดงข้อมูลผู้ใช้เรียบร้อยแล้ว	System Response Step 2: ระบบทำการค้นหาตาม User Profile ID Step 3: แสดงผลออกมาทาง User Profile Detail
ALTERNATE COURSES:	Step 2: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POST-CONDITION	User Profile ที่ต้องการ ได้ทำการแสดงเรียบร้อยแล้ว
ASSUMPTIONS:	ไม่มี

ตารางที่ 3.6 ยูสเคสการเข้าใช้ระบบและตรวจสอบสิทธิ์

USE CASE NAME:	Authenticate	
ACTOR(S):	System Admin, Test Operator, Tester	
DESCRIPTION:	ยูสเคสนี้เป็นการตรวจสอบสิทธิ์การเข้าใช้ระบบ	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: Actor ต้องการเข้าใช้ระบบ Step 4: ยูสเคสเสร็จสิ้นเมื่อได้ให้สิทธิ์แก่ผู้ใช้เรียบร้อยแล้ว	System Response Step 2: ระบบทำการค้นหาข้อมูลของผู้ใช้ระบบจากฐานข้อมูลโดยใช้ Username และ Password Step 3: ระบบตรวจสอบและให้สิทธิ์แก่ผู้ใช้
ALTERNATE COURSES:	Step 2,3: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	ผู้ใช้ได้รับสิทธิ์และสามารถเข้าใช้ระบบได้ตามสิทธิ์ที่ได้รับ	
ASSUMPTIONS:	ไม่มี	

ตารางที่ 3.7 ยูสเคสเกี่ยวกับการสร้างข้อมูลสำหรับการทดสอบ

USE CASE NAME:	Create Test Information
ACTOR(S):	Tester
DESCRIPTION:	ยูสเคสนี้เป็นการสร้างข้อมูลสำหรับการทดสอบ

TYPICAL COURSE OF EVENTS:	Actor Action Step 1: Tester ร้องขอทำการสร้างข้อมูลสำหรับการทดสอบใหม่ Step 4: ยูสเคสเสร็จสิ้นเมื่อได้ทำการแสดงรายการข้อมูลสำหรับการทดสอบ	System Response Step 2: ระบบทำการสร้างข้อมูลสำหรับการทดสอบใหม่ และบันทึกข้อมูลลงฐานข้อมูล Step 3: แสดงผลออกมาทางรายการข้อมูลสำหรับการทดสอบ
ALTERNATE COURSES:	Step 2: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	สร้างข้อมูลสำหรับการทดสอบเรียบร้อยแล้ว	
ASSUMPTIONS:	ไม่มี	

ตารางที่ 3.8 ยูสเคสเกี่ยวกับการลบข้อมูลสำหรับการทดสอบ

USE CASE NAME:	Delete Test Information	
ACTOR(S):	Tester	
DESCRIPTION:	ยูสเคสนี้เป็นการลบข้อมูลสำหรับการทดสอบที่ต้องการ	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: Tester ร้องขอทำการลบข้อมูลสำหรับการทดสอบ	System Response Step 2: ระบบทำการลบข้อมูลสำหรับการทดสอบออกจาก ฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Step 4: ยูสเคสเสร็จสิ้นเมื่อได้ทำการแสดงรายการข้อมูลสำหรับการทดสอบ	Step 3: แสดงผลออกมาทางรายการข้อมูลสำหรับการทดสอบ
ALTERNATE COURSES:	Step 2: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	ลบข้อมูลสำหรับการทดสอบเรียบร้อยแล้ว	
ASSUMPTIONS:	มีข้อมูลสำหรับการทดสอบในฐานข้อมูล	

ตารางที่ 3.9 ยูสเคสเกี่ยวกับการแก้ไขข้อมูลสำหรับการทดสอบ

USE CASE NAME:	Edit test information	
ACTOR(S):	Tester	
DESCRIPTION:	ยูสเคสนี้เป็นการแก้ไขข้อมูลสำหรับการทดสอบ	
TYPICAL COURSE OF EVENTS:	<p>Actor Action</p> <p>Step 1: Tester ร้องขอทำการแก้ไขข้อมูลสำหรับการทดสอบ</p> <p>Step 4: ยูสเคสเสร็จสิ้นเมื่อได้ทำการแสดงรายการข้อมูลสำหรับการทดสอบ</p>	<p>System Response</p> <p>Step 2: ระบบทำการแก้ไขข้อมูลสำหรับการทดสอบและบันทึกข้อมูลลงฐานข้อมูล</p> <p>Step 3: แสดงผลออกมาทางรายการข้อมูลสำหรับการทดสอบ</p>
ALTERNATE COURSES:	Step 2: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	แก้ไขข้อมูลสำหรับการทดสอบเรียบร้อยแล้ว	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ASSUMPTIONS:	มีข้อมูลสำหรับการทดสอบเดิมอยู่ในฐานข้อมูล
---------------------	---

ตารางที่ 3.10 ยูสเคสเกี่ยวกับการแสดงรายการข้อมูลสำหรับการทดสอบ

USE CASE NAME:	List test information	
ACTOR(S):	Tester	
DESCRIPTION:	ยูสเคสนี้เป็นการแสดงรายการข้อมูลสำหรับการทดสอบ	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: Tester ร้องขอทำการแสดงรายการข้อมูลสำหรับการทดสอบ Step 4: ยูสเคสเสร็จสิ้นเมื่อได้ทำการแสดงรายการข้อมูลสำหรับการทดสอบ	System Response Step 2: ระบบทำการแสดงรายการข้อมูลสำหรับการทดสอบตามเงื่อนไขที่กำหนด Step 3: แสดงผลออกมาทางรายการข้อมูลสำหรับการทดสอบ
ALTERNATE COURSES:	Step 2: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	แสดงรายการ Test Information เรียบร้อย	
ASSUMPTIONS:	ไม่มี	

ตารางที่ 3.11 ยูสเคสเกี่ยวกับการแสดงข้อมูลสำหรับการทดสอบ

USE CASE NAME:	View test information
ACTOR(S):	Tester
DESCRIPTION:	ยูสเคสนี้เป็นการแสดงข้อมูลสำหรับการทดสอบ โดยระบุ ID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL COURSE OF EVENTS:	Actor Action Step 1: Tester ร้องขอทำการแสดงข้อมูลสำหรับการทดสอบ Step 4: ยูสเคสเสร็จสิ้นเมื่อได้ทำการแสดงรายการข้อมูลสำหรับการทดสอบ	System Response Step 2: ระบบทำการแสดงรายละเอียดของข้อมูลสำหรับการทดสอบตาม ID Step 3: แสดงผลออกมาทางรายการข้อมูลสำหรับการทดสอบ
ALTERNATE COURSES:	Step 2: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	แสดงข้อมูลสำหรับการทดสอบเรียบร้อยแล้ว	
ASSUMPTIONS:	ไม่มี	

ตารางที่ 3.12 ยูสเคสเกี่ยวกับการสร้างสคริปต์สำหรับการทดสอบ

USE CASE NAME:	Generate Test Script	
ACTOR(S):	Tester	
DESCRIPTION:	ยูสเคสนี้เป็นการสร้างสคริปต์สำหรับการทดสอบจากข้อมูลที่อยู่ในฐานข้อมูล	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: Tester ร้องขอทำการสร้างสคริปต์สำหรับการทดสอบ	System Response Step 2: ระบบทำการค้นหาข้อมูลสำหรับการทดสอบตามที่ผู้ใช้เลือก โดยทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		<p>แปลงให้อยู่ในรูปแบบเอกสารเอกซ์เอ็มแอล และบันทึกไว้</p> <p>Step 3: นำเอกสารเอกซ์เอ็มแอลที่ได้มาแปลงด้วยกระบวนการแปลง Style Sheet เพื่อให้ได้สคริปต์สำหรับการใช้ในการทดสอบ</p> <p>Step 4: บันทึกค่าอ้างอิงต่าง ๆ สำหรับเอกสารสคริปต์ในฐานข้อมูล</p> <p>Step 5: ค้นหาและแสดงรายการเอกสารสคริปต์แก่ผู้ใช้</p>
	<p>Step 6: บุคลากรเสร็จสิ้นเมื่อได้ทำการแสดงรายการเอกสารสคริปต์สำหรับการทดสอบ</p>	
ALTERNATE COURSES:	Step 2,3,4: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	แสดงรายการเอกสารสคริปต์สำหรับการทดสอบเรียบร้อยแล้ว	
ASSUMPTIONS:	ไม่มี	

ตารางที่ 3.13 บุคลากรเกี่ยวกับการแสดงรายการสคริปต์สำหรับการทดสอบ

USE CASE NAME:	List Test Script	
ACTOR(S):	Tester Operator, Tester	
DESCRIPTION:	บุคลากรนี้เป็นการแสดงรายการเอกสารสคริปต์สำหรับการทดสอบที่มีการสร้างไว้ในระบบ	
TYPICAL COURSE OF EVENTS:	<p>Actor Action</p> <p>Step 1: Actor ร้องขอทำการ</p>	System Response

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	<p>แสดงข้อมูลสำหรับการทดสอบ</p> <p>Step 4: ยูสเคสเสร็จสิ้นเมื่อได้ทำการแสดงรายการเอกสารสคริปต์สำหรับการทดสอบ</p>	<p>Step 2: ระบบทำการค้นหาข้อมูลอ้างอิงของเอกสารสคริปต์ในฐานข้อมูล</p> <p>Step 3: แสดงผลออกมาทางรายการเอกสารสคริปต์สำหรับการทดสอบ</p>
ALTERNATE COURSES:	Step 2: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	แสดงรายการเอกสารสคริปต์สำหรับการทดสอบเรียบร้อยแล้ว	
ASSUMPTIONS:	ไม่มี	

ตารางที่ 3.14 ยูสเคสเกี่ยวกับการแสดงรายละเอียดของสคริปต์สำหรับการทดสอบ

USE CASE NAME:	View Test Script	
ACTOR(S):	Tester Operator, Tester	
DESCRIPTION:	ยูสเคสนี้เป็นการแสดงรายละเอียดของเอกสารสคริปต์สำหรับการทดสอบที่มีการสร้างไว้ในระบบ	
TYPICAL COURSE OF EVENTS:	<p>Actor Action</p> <p>Step 1: Actor ร้องขอทำการแสดงรายละเอียดของข้อมูลสำหรับการทดสอบ</p>	<p>System Response</p> <p>Step 2: ระบบทำการค้นหาข้อมูลอ้างอิงของเอกสารสคริปต์ในฐานข้อมูลจาก ID ที่ผู้ใช้เลือก</p>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Step 4: ยูสเคสเสร็จสิ้นเมื่อได้ทำการแสดงรายละเอียดของเอกสารสคริปต์สำหรับการทดสอบ	Step 3: แสดงผลรายละเอียดของเอกสารสคริปต์สำหรับการทดสอบแก่ผู้ใช้
ALTERNATE COURSES:	Step 2: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	แสดงรายการเอกสารสคริปต์สำหรับการทดสอบเรียบร้อยแล้ว	
ASSUMPTIONS:	ไม่มี	

ตารางที่ 3.15 ยูสเคสเกี่ยวกับการลบสคริปต์สำหรับการทดสอบ

USE CASE NAME:	Delete Test Scripts	
ACTOR(S):	Tester	
DESCRIPTION:	ยูสเคสนี้เป็นการลบข้อมูลเกี่ยวกับเอกสารสคริปต์สำหรับการทดสอบ	
TYPICAL COURSE OF EVENTS:	<p>Actor Action</p> <p>Step 1: Tester ร้องขอทำการลบข้อมูลเกี่ยวกับเอกสารสคริปต์สำหรับการทดสอบ</p> <p>Step 4: ยูสเคสเสร็จสิ้นเมื่อได้ทำการแสดงรายการข้อมูลเกี่ยวกับเอกสารสคริปต์สำหรับการทดสอบ</p>	<p>System Response</p> <p>Step 2: ระบบทำการลบข้อมูลเกี่ยวกับเอกสารสคริปต์สำหรับการทดสอบตาม ID ที่เลือก</p> <p>Step 3: แสดงผลรายการข้อมูลเกี่ยวกับเอกสารสคริปต์สำหรับการทดสอบหลังการลบแล้วแก่ผู้ใช้</p>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ALTERNATE COURSES:	Step 2: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ
PRE-CONDITION:	ไม่มี
POST-CONDITION	แสดงรายการข้อมูลเกี่ยวกับเอกสารสคริปต์สำหรับการทดสอบเรียบร้อยแล้ว
ASSUMPTIONS:	ไม่มี

ตารางที่ 3.16 ยูสเคสการเรียกสคริปต์ทำการทดสอบโปรแกรม

USE CASE NAME:	Execute Test Script	
ACTOR(S):	Tester, Test Operator	
DESCRIPTION:	ยูสเคสนี้เป็นการเรียกโปรแกรมที่ต้องการทดสอบขึ้นมาทำงานตามสคริปต์ที่เลือก	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: Actor ต้องการทดสอบโปรแกรมตามเอกสารสคริปต์ที่เลือก	System Response Step 2: ระบบทำการค้นหาสคริปต์ไฟล์และเปิดไฟล์ Step 3: นำสคริปต์มาแปลงให้เป็นคำสั่งที่สามารถนำไปประมวลผลได้ โดยการเรียกทำผ่านส่วนการแปลงสคริปต์ Step 4: ทำการประมวลผลโปรแกรมที่ต้องการทดสอบด้วยคำสั่งที่ได้ Step 5: สร้างเอกสารเอกซ์เอ็มแอลของผลการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Step 7: ยูสเคสเสร็จสิ้นเมื่อได้รับผลการทดสอบเรียบร้อยแล้ว	Step 6: เรียกยูสเคสการแสดงผลการทดสอบเพื่อนำเอกสารเอกซ์เอ็มแอลของผลการทดสอบไปแปลงให้อยู่ในรูปแบบที่แสดงผลได้
ALTERNATE COURSES:	Step 4: หากระบบไม่สามารถเรียกประมวลผลได้ต้องแจ้งให้ Actor ทราบ Step 6: อาจมีการส่ง E-mail ให้ Actor ทราบถึงผลการทดสอบด้วย	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	ได้รับผลการทดสอบโปรแกรมตามสคริปต์ได้เรียบร้อยแล้ว	
ASSUMPTIONS:	เอกสารสคริปต์สำหรับเรียกทดสอบ โปรแกรมต้องมีการสร้างไว้ก่อนแล้ว และ โปรแกรมที่ต้องการทดสอบอยู่ในไคลเรกทอรีที่กำหนดไว้ในเอกสารสคริปต์เรียบร้อยแล้ว	

ตารางที่ 3.17 ยูสเคสการแปลงสคริปต์เพื่อใช้เรียกประมวลผล

USE CASE NAME:	Script Transformation	
ACTOR(S):	Test Executor	
DESCRIPTION:	ยูสเคสนี้เป็นการแปลง	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: ส่วนการเรียกโปรแกรมขึ้นมาทดสอบต้องการคำสั่งที่เหมาะสม	System Response Step 2: ระบบทำการค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	สำหรับการประมวลผล	คำสั่งที่เหมาะสมด้วย Utilities Tag Step 3: นำคำสั่งที่ได้มารวม กับพารามิเตอร์ที่มี Step 3: ส่งคำสั่งที่เหมาะสม กลับ
	Step 4: ยูสเคสเสร็จสิ้นเมื่อ ได้รับคำสั่งที่เหมาะสม เรียบร้อยแล้ว	
ALTERNATE COURSES:	Step 2,3: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	
POST-CONDITION:	ผลจากการแปลงสคริปต์ถูกต้อง	
ASSUMPTIONS:	มีการกำหนดคำสั่งสำหรับแต่ละ utilities tag	

ตารางที่ 3.18 ยูสเคสเกี่ยวกับการสร้างผลการทดสอบ

USE CASE NAME:	Create Test Result	
ACTOR(S):	ยูสเคส Execute Test Script	
DESCRIPTION:	ยูสเคสนี้เป็นการสร้าง Test Result XML	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: ยูสเคสได้รับผลการ ทดสอบจากการทดสอบ	System Response Step 2: ระบบทำการเปิด เอกสารเอกซ์เอ็มแอล สำหรับการทดสอบ Step 3: สร้างข้อมูลที่เป็นผล ของการทดสอบลงใน เอกสารเอกซ์เอ็มแอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	<p>Step 4: บันทึกเอกสารเอกซ์เอ็มแอลสำหรับผลการทดสอบ</p> <p>Step 5: บันทึกข้อมูลเกี่ยวกับ Path ของผลการทดสอบในฐานข้อมูล</p> <p>Step 6: เรียกส่วนการเชื่อมต่อกับฐานข้อมูลทำการกระจายข้อมูลผลการทดสอบและบันทึกลงในฐานข้อมูล</p>
	<p>Step 7: บุสเคสเสร็จสิ้นเมื่อทำการสร้างผลการทดสอบเรียบร้อยแล้ว</p>
ALTERNATE COURSES:	Step 2,3,4,5: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ
PRE-CONDITION:	ทำการเรียกโปรแกรมที่ต้องการทดสอบมาทำงานตามเอกสารสคริปต์แล้ว
POST-CONDITION	สร้าง XML ของผลการทดสอบเรียบร้อยแล้ว
ASSUMPTIONS:	มีผลการทดสอบจากการประมวลผลเอกสารสคริปต์

ตารางที่ 3.19 บุสเคสเกี่ยวกับการลบผลการทดสอบ

USE CASE NAME:	Delete Test Result	
ACTOR(S):	, Tester	
DESCRIPTION:	บุสเคสนี้เป็นการลบผลการทดสอบ	
TYPICAL COURSE OF EVENTS:	<p>Actor Action</p> <p>Step 1: User ร้องขอทำการลบผลการทดสอบ ที่ต้องการ</p>	<p>System Response</p> <p>Step 2: ระบบทำการค้นหาผลการทดสอบ ที่ต้องการ</p>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		ลบ ตาม PK Step 3: ทำการลบผลการทดสอบและเอกสารที่เกี่ยวข้อง
ALTERNATE COURSES:	Step 2: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	ลบผลการทดสอบเรียบร้อยแล้ว	
ASSUMPTIONS:	มีข้อมูลผลการทดสอบในฐานข้อมูล	

ตารางที่ 3.20 ยูสเคสเกี่ยวกับการแสดงรายการผลการทดสอบ

USE CASE NAME:	List Test Result	
ACTOR(S):	Test Operator , Test Manger, Tester	
DESCRIPTION:	ยูสเคสนี้เป็นการแสดงรายการผลการทดสอบ	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: User ร้องขอทำการแสดงรายการผลการทดสอบ Step 4: ยูสเคสเสร็จสิ้นเมื่อทำการแสดงรายการผลการทดสอบ	System Response Step 2: ระบบทำการค้นหาผลการทดสอบ ที่ต้องการตามเงื่อนไขที่กำหนด Step 3: แสดงรายการผลการทดสอบ
ALTERNATE COURSES:	ไม่มี	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	แสดงรายการผลการทดสอบ เรียบร้อย	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ASSUMPTIONS:	ไม่มี
---------------------	-------

ตารางที่ 3.21 ยูสเคสเกี่ยวกับการแสดงผลการทดสอบ

USE CASE NAME:	View Test Result	
ACTOR(S):	Test Operator , Test Manger, Tester	
DESCRIPTION:	ยูสเคสนี้เป็นการแสดงผลการทดสอบที่ต้องการ	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: User ร้องขอทำการแสดงผลการทดสอบ Step 5: ยูสเคสเสร็จสิ้นเมื่อทำการแสดงผลการทดสอบ	System Response Step 2: ระบบทำการค้นหาผลการทดสอบ ที่ต้องการตาม PK Step 3: ระบบทำการแปลงเอกสารเอกซ์เอ็มแอล ของผลการทดสอบเป็น HTML Step 4: แสดงผลการทดสอบเป็น HTML
ALTERNATE COURSES:	Step 3: ถ้าระบบทำไม่สำเร็จจะแจ้งให้ Actor ทราบ	
PRE-CONDITION:	ไม่มี	
POST-CONDITION	แสดงผลการทดสอบ เรียบร้อย	
ASSUMPTIONS:	มีผลการทดสอบเป็นเอกสารเอกซ์เอ็มแอลในระบบ	

ตารางที่ 3.22 ยูสเคสการประมวลผล Style Sheet

USE CASE NAME:	Style Sheet Process
ACTOR(S):	Generate Test Script
DESCRIPTION:	ยูสเคสนี้เป็นการอ่าน XSL Style Sheet ที่เหมาะสมมาอ่านและประมวลผลร่วมกับเอกสาร XML ที่เป็นข้อมูลการทดสอบ เพื่อให้ได้เอกสารรูปแบบใหม่ เช่น สคริปต์ในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หรือ HTML	
TYPICAL COURSE OF EVENTS:	<p>Actor Action</p> <p>Step 1: ยูสเคสการสร้างสคริปต์ของการทดสอบต้องการนำ XSL Style Sheet เข้าร่วมกับเอกสารเอกซ์เอ็มแอลเพื่อสร้างสคริปต์</p> <p>Step 2: ยูสเคสการเรียกดูผลการทดสอบต้องการนำ XSL Style Sheet เข้าร่วมกับเอกสารเอกซ์เอ็มแอลเพื่อสร้างเป็นเอกสารเอชทีเอ็มแอล</p> <p>Step 3: ยูสเคสเสร็จสิ้นเมื่อได้รับผลการทำงาน</p>	<p>System Response</p> <p>Step 1.1: ส่วนประมวลผล Style Sheet ทำการรวม XSL Style Sheet สำหรับแปลงสคริปต์เข้ากับเอกสารเอกซ์เอ็มแอล</p> <p>Step 1.2: ระบบส่งข้อมูลในรูปแบบเอกสารสคริปต์กลับ</p> <p>Step 2.1: ส่วนประมวลผล Style Sheet ทำการรวม XSL Style Sheet สำหรับเอกสารเอชทีเอ็มแอลเข้ากับเอกสารเอกซ์เอ็มแอล</p> <p>Step 2.2: ระบบส่งข้อมูลในรูปแบบเอกสารเอชทีเอ็มแอลกลับ</p>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	เรียบร้อย	
ALTERNATE COURSES:	Step 1.1,2.1: ถ้าระบบไม่สามารถทำการรวม XSL Style Sheet ที่เหมาะสมเข้ากับเอกสารเอกซ์เอ็มแอลได้ ต้องแจ้งให้ Actor ทราบ	
PRE-CONDITION:	มีการเรียกใช้จากยูสเคสการสร้างสคริปต์ หรือ ยูสเคสการเรียกใช้สคริปต์การทดสอบทำงาน	
POST-CONDITION	ได้รับเอกสารรูปแบบที่ต้องการจากระบบ	
ASSUMPTIONS:	มีการสร้าง XSL Style Sheet ไว้แล้ว	

ตารางที่ 3.23 ยูสเคสการเชื่อมต่อกับฐานข้อมูลด้วยเอกสาร XML

USE CASE NAME:	Interface XML with Repository	
ACTOR(S):	Test Script Generator , Result Creator	
DESCRIPTION:	ยูสเคสนี้เป็นการทำการเชื่อมต่อระหว่างข้อมูลในเอกสารเอกซ์เอ็มแอลกับข้อมูลในฐานข้อมูล ซึ่งมีการทำงานสองรูปแบบ คือ การกระจายข้อมูลลงฐานข้อมูล และการเก็บรวบรวมข้อมูลจากฐานข้อมูลเป็นไฟล์	
TYPICAL COURSE OF EVENTS:	Actor Action Step 1: ยูสเคสการสร้างสคริปต์ของการทดสอบต้องการเอกสารเอกซ์เอ็มแอลของข้อมูลสำหรับการทดสอบ	System Response Step 1.1: ส่วนการเชื่อมต่อข้อมูลทำการเก็บรวบรวมข้อมูลจากฐานข้อมูล Step 1.2: เรียกส่วนการแปลงข้อมูลจาก Data Object ให้อยู่ในรูปแบบเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	<p>Step 2: ส่วนสร้างผลการทดสอบต้องการนำข้อมูลผลการทดสอบจากเอกสารเอกซ์เอ็มแอลของข้อมูลสำหรับการทำการทดสอบ</p> <p>Step 3: ยูสเคสเสร็จสิ้นเมื่อได้รับผลการทำงานเรียบร้อยแล้ว</p>	<p>เอกซ์เอ็มแอล</p> <p>Step 1.3: ระบบส่งเอกสารเอกซ์เอ็มแอลกลับ</p> <p>Step 2.1: ส่วนการเชื่อมต่อข้อมูลทำการกระจายข้อมูลจากเอกสารเอกซ์เอ็มแอล</p> <p>Step 2.2: เรียกส่วนการจัดการข้อมูลในฐานข้อมูลทำการบันทึกข้อมูลที่ได้จากเอกสารเอกซ์เอ็มแอล</p> <p>Step 2.3: ระบบส่งผลการบันทึกข้อมูลกลับ</p>
ALTERNATE COURSES:	<p>Step 1.2 ถ้าระบบไม่สามารถทำการแปลงเป็นเอกสาร XML ได้ ต้องแจ้งให้ Actor ทราบ</p> <p>Step 1.3 ถ้าระบบไม่สามารถทำการบันทึกข้อมูลในฐานข้อมูลได้ ต้องแจ้งให้ Actor ทราบ</p>	
PRE-CONDITION:	มีการเรียกใช้จากยูสเคสการสร้างสคริปต์ หรือ ยูสเคสการสร้างผลการทดสอบทำงาน	
POST-CONDITION	ได้รับเอกสารเอกซ์เอ็มแอลที่ต้องการ หรือผลการบันทึกข้อมูลจากระบบ	
ASSUMPTIONS:	ไม่มี	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบระบบงาน

4.1 การออกแบบฐานข้อมูล

4.1.1 ฐานข้อมูลของหลักการทำงานยูนิเวอร์แซลเทสต์ไครเวอร์ (Cordrey. 2002)

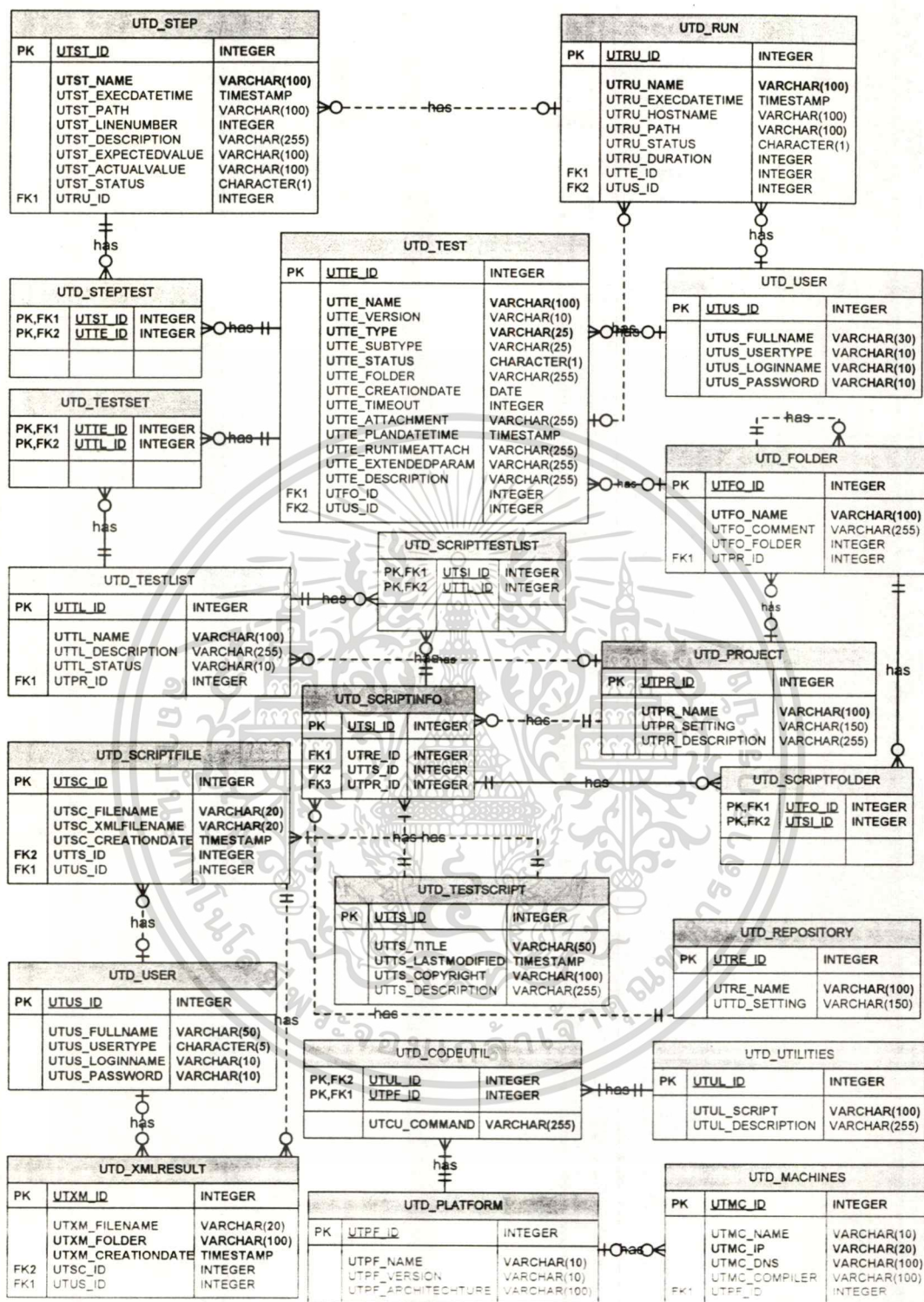
ในการวิเคราะห์เพื่อออกแบบฐานข้อมูลของระบบ และหาความสัมพันธ์ของข้อมูลที่ใช้ในการทำงานของยูนิเวอร์แซลเทสต์ไครเวอร์มีรายละเอียดของตารางของฐานข้อมูลดังแสดงในตารางที่ 4.1 และพิจารณาในรูปแบบของ Entity Relationship Diagram ดังรูปที่ 4.1

ตารางที่ 4.1 แสดงตารางทั้งหมดบนฐานข้อมูลตามหลักการทำงานยูนิเวอร์แซลเทสต์ไครเวอร์

ลำดับที่	ชื่อตาราง	คำอธิบาย
1.	UTD_USER	ตารางแสดงรายละเอียดข้อมูลการเข้าสู่ระบบของผู้ใช้ระบบ
2.	UTD_PROJECT	ตารางแสดงรายละเอียดโครงการที่จะทำการทดสอบ
3.	UTD_TESTLIST	ตารางแสดงรายละเอียดกลุ่มการทดสอบ
4.	UTD_TEST	ตารางแสดงรายละเอียดกรณีการทดสอบ
5.	UTD_TESTSET	ตารางแสดงความสัมพันธ์ระหว่างกลุ่มและกรณีของการทดสอบ
6.	UTD_STEP	ตารางแสดงรายละเอียดขั้นตอนการทดสอบ
7.	UTD_STEPTEST	ตารางแสดงความสัมพันธ์ระหว่างขั้นตอนการทดสอบและกรณีการทดสอบ
8.	UTD_RUN	ตารางแสดงรายละเอียดของการทดสอบที่ได้กระทำไป
9.	UTD_TESTSCRIPT	ตารางแสดงรายละเอียดข้อมูลพื้นฐานในเอกสารของการทดสอบ
10.	UTD_SCRIPTTESTLIST	ตารางแสดงความสัมพันธ์ระหว่าง UTD_SCRIPTINFO และกลุ่มการทดสอบที่ได้กำหนดไว้

11.	UTD_XMLRESULT	ตารางแสดงรายละเอียดของเอกสารผลการทดสอบ
12.	UTD_UTILITIES	ตารางแสดงรายละเอียดของกลุ่มคำสั่งที่ช่วยในการทดสอบ
13.	UTD_CODEUTIL	ตารางแสดงความสัมพันธ์ระหว่างกลุ่มคำสั่งที่ช่วยในการทดสอบและ Platform ที่ใช้
14.	UTD_PLATFORM	ตารางแสดงรายละเอียดของ Platform ที่ใช้ในการทดสอบ
15.	UTD_SCRIPTINFO	ตารางแสดงความสัมพันธ์ระหว่าง UTD_REPOSITORY, UTD_PROJECT, และ UTD_TESTSCRIPT
16.	UTD_SCRIPTFILE	ตารางแสดงรายละเอียดของเอกสารสคริปต์สำหรับการทดสอบ
17.	UTD_SCRIPTFOLDER	ตารางแสดงความสัมพันธ์ระหว่างเอกสารสคริปต์และ Folder สำหรับการจัดเก็บเอกสารสคริปต์
18.	UTD_FOLDER	ตารางแสดงรายละเอียดของ Folder ที่ใช้ในการเก็บข้อมูลต่าง ๆ ที่เกี่ยวข้องกับการทดสอบ
19.	UTD_REPOSITORY	ตารางแสดงรายละเอียดของคลังข้อมูลที่จัดเก็บกรณีการทดสอบ
20.	UTD_MACHINES	ตารางแสดงรายละเอียดของเครื่องที่ใช้ในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 Entity Relationship Diagram สำหรับหลักการทำงานยูนิเวอร์แซลเทสต์ไดเรกทอรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ตารางกลุ่มการทดสอบ (UTD_USER)

Table Description: แสดงรายละเอียดข้อมูลการเข้าสู่ระบบของผู้ใช้ระบบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTUS_ID	รหัสผู้ใช้	INTEGER	PK	/	
2	UTUS_FULLNAME	ชื่อผู้ใช้	VARCHAR(50)		/	
3	UTUS_USERTYPE	ประเภทผู้ใช้	VARCHAR(10)		/	
4	UTUS_LOGINNAME	ชื่อในการเข้าใช้ระบบ	VARCHAR(10)		/	
5	UTUS_PASSWORD	รหัสในการเข้าใช้ระบบ	VARCHAR(10)		/	

ตารางที่ 4.3 ตารางโครงการ (UTD_PROJECT)

Table Description: แสดงรายละเอียดโครงการที่จะทำการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTPR_ID	รหัสโครงการ	INTEGER	PK	/	
2	UTPR_NAME	ชื่อโครงการ	VARCHAR(100)		/	
3	UTPR_SETTING	ข้อกำหนดเพื่อการเข้าถึง	VARCHAR(150)			
4	UTPR_DESCRIPTION	หมายเหตุ	VARCHAR(255)			

ตารางที่ 4.4 ตารางกลุ่มการทดสอบ (UTD_TESTLIST)

Table Description: ตารางแสดงรายละเอียดกลุ่มการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTTL_ID	รหัสกลุ่มการทดสอบ	INTEGER	PK	/	
2	UTTL_NAME	ชื่อกลุ่มการทดสอบ	VARCHAR(100)		/	
3	UTTL_DESCRIPTION	หมายเหตุ	VARCHAR(255)			
4	UTTL_STATUS	สถานะ เช่น "Open"	VARCHAR(10)			
5	UTPR_ID	รหัสโครงการ	INTEGER	FK		UTD_PROJECT

ตารางที่ 4.5 ตารางกรณีกการทดสอบ (UTD_TEST)

Table Description: ตารางแสดงรายละเอียดกรณีกการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTTE_ID	รหัสกรณีกการทดสอบ	INTEGER	PK	/	
2	UTTE_NAME	ชื่อกรณีกการทดสอบ	VARCHAR(100)		/	
3	UTTE_VERSION	รุ่นของการทดสอบ	VARCHAR(10)			
4	UTTE_TYPE	ประเภทของการทดสอบ เช่น "Manual"	VARCHAR(25)		/	
5	UTTE_SUBTYPE	ประเภทย่อยของการทดสอบ	VARCHAR(25)			
6	UTTE_STATUS	สถานะ (R=ready, D=Design)	CHARACTER(1)		/	
7	UTTE_FOLDER	Folder ในการทดสอบ	VARCHAR(255)			
8	UTTE_CREATIONDATE	วันที่สร้าง	DATE			
9	UTTE_TIMEOUT	ระยะเวลาที่กำหนด	INTEGER			
10	UTTE_ATTACHMENT	เอกสารแนบ	VARCHAR(255)			
11	UTTE_PLANDATETIME	วันเวลาที่กำหนดสำหรับทดสอบ	TIMESTAMP			
12	UTTE_RUNTIMEATTACH	เอกสารแนบในการ RUN	VARCHAR(255)			
13	UTTE_EXTENDEDPARAM	พารามิเตอร์เพิ่มเติม	VARCHAR(255)			
14	UTTE_DESCRIPTION	หมายเหตุ	VARCHAR(255)			
15	UTFO_ID	รหัส FOLDER	INTEGER	FK		UTD_FOLDER
16	UTUS_ID	รหัสผู้รับผิดชอบ	INTEGER	FK		UTD_USER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.6 ตารางความสัมพันธ์ของกลุ่มและกรณีการทดสอบ (UTD_TESTSET)

Table Description: ตารางแสดงความสัมพันธ์ระหว่างกลุ่มและกรณีของการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTTE_ID	รหัส TEST	INTEGER	PK,FK	/	UTD_TEST
2	UTTL_ID	รหัส TEST LIST	INTEGER	PK,FK	/	UTD_TESTLIST

ตารางที่ 4.7 ตารางขั้นตอนการทดสอบ (UTD_STEP)

Table Description: ตารางแสดงรายละเอียดขั้นตอนการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTST_ID	รหัส STEP	INTEGER	PK	/	
2	UTST_NAME	ชื่อ STEP	VARCHAR(100)		/	
3	UTST_EXECDATETIME	วันเวลาที่ ประมวลผล	TIMESTAMP			
4	UTST_PATH	Path ที่เกี่ยวข้อง	VARCHAR(100)			
5	UTST_LINENUMBER	บรรทัดที่ทดสอบ	INTEGER			
6	UTST_DESCRIPTION	หมายเหตุ	VARCHAR(255)			
7	UTST_EXPECTEDVALUE	ค่าที่คาดหวังไว้	VARCHAR(100)			
8	UTST_ACTUALVALUE	ค่าที่ได้จริง	VARCHAR(100)			
9	UTST_STATUS	สถานะ (P=passed,F=failed)	CHARACTER(1)			
10	UTRU_ID	รหัส RUN	INTEGER	FK		UTD_RUN

ตารางที่ 4.8 ตารางความสัมพันธ์ของขั้นตอนและกรณีการทดสอบ (UTD_STEPTEST)

Table Description: ตารางแสดงความสัมพันธ์ระหว่างขั้นตอนการทดสอบและกรณีการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTST_ID	รหัส STEP	INTEGER	PK,FK	/	UTD_STEP
2	UTTE_ID	รหัส TEST	INTEGER	PK,FK	/	UTD_TEST

ตารางที่ 4.9 ตารางข้อมูลการทดสอบที่ได้กระทำไป (UTD_RUN)

Table Description: ตารางแสดงรายละเอียดของการทดสอบที่ได้กระทำไป

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTRU_ID	รหัส RUN	INTEGER	PK	/	
2	UTRU_NAME	ชื่อ RUN	VARCHAR(100)		/	
3	UTRU_EXECDATETIME	วันที่ ประมวลผล	TIMESTAMP			
4	UTRU_HOSTNAME	host_name ที่ ทดสอบ	VARCHAR(100)			
5	UTRU_PATH	path ที่เกี่ยวข้อง	VARCHAR(100)			
6	UTRU_STATUS	สถานะ (P=passed,F=failed)	CHARACTER(1)			
7	UTRU_DURATION	ระยะเวลาที่ทดสอบ	INTEGER			
8	UTTE_ID	รหัส TEST	INTEGER	FK		UTD_TEST
9	UTUS_ID	รหัส ผู้ทดสอบ	INTEGER	FK		UTD_USER

ตารางที่ 4.10 ตารางเอกสารสำหรับการทดสอบ (UTD_TESTSCRIPT)

Table Description: ตารางแสดงรายละเอียดเอกสารของการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTTS_ID	รหัส TEST DOCUMENT	INTEGER	PK	/	
2	UTTS_TITLE	ชื่อเอกสารสคริปต์	VARCHAR(50)		/	
3	UTTS_LASTMODIFIED	วันที่แก้ไขล่าสุด	TIMESTAMP		/	
4	UTTS_COPYRIGHT	ลิขสิทธิ์	VARCHAR(100)		/	
5	UTTS_DESCRIPTION	หมายเหตุ	VARCHAR(255)			

ตารางที่ 4.11 ตารางความสัมพันธ์ของเอกสารและกลุ่มการทดสอบ (UTD_SCRIPTTESTLIST)

Table Description: ตารางแสดงความสัมพันธ์ระหว่างเอกสารและกลุ่มการทดสอบที่ได้กำหนดไว้

No.	Attribute	Description	Data Type	Key	Req	Reference
1	LTSI_ID	รหัสข้อมูลในเอกสาร สคริปต์	INTEGER	PK,FK	/	UTD_SCRIPTINFO
2	UTTL_ID	รหัส TEST LIST	INTEGER	PK,FK	/	UTD_TESTLIST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.12 ตารางเอกสารผลการทดสอบ (UTD_XMLRESULT)

Table Description: ตารางแสดงรายละเอียดของเอกสารผลของการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTXM_ID	รหัสไฟล์ผลการทดสอบ	INTEGER	PK	/	
2	UTXM_FILENAME	ชื่อไฟล์	VARCHAR(20)		/	
3	UTXM_FOLDER	Folder ที่เก็บ	VARCHAR(100)		/	
4	UTXM_CREATIONDATE	วันที่สร้าง	TIMESTAMP		/	
5	UTSC_ID	รหัส SCRIPT FILE	INTEGER	FK	/	UTD_SCRIPTFILE
6	UTUS_ID	รหัสผู้ทดสอบ สคริปต์	INTEGER	FK		UTD_USER

ตารางที่ 4.13 ตารางคำสั่งที่ช่วยในการทดสอบ (UTD_UTILITIES)

Table Description: ตารางแสดงรายละเอียดกลุ่มคำสั่งที่ช่วยในการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTUL_ID	รหัส Utilities	INTEGER	PK	/	
2	UTUL_SCRIPT	สคริปต์ของ Utilities	VARCHAR(100)		/	
3	UTUL_DESCRIPTION	หมายเหตุ	VARCHAR(100)			

ตารางที่ 4.14 ตารางความสัมพันธ์ของกลุ่มคำสั่งที่ช่วยในการทดสอบและ Platform (UTD_CODEUTIL)

Table Description: ตารางแสดงความสัมพันธ์ระหว่างกลุ่มคำสั่งที่ช่วยในการทดสอบและ Platform ที่ใช้

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTUL_ID	รหัส Utilities	INTEGER	PK,FK	/	UTD_UTILITIES
2	UTPF_ID	รหัส Platform	INTEGER	PK,FK	/	UTD_PLATFORM
3	UTCU_COMMAND	คำสั่งในการทำงาน	VARCHAR(255)		/	

ตารางที่ 4.15 ตารางแสดงรายละเอียดของ Platform (UTD_PLATFORM)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table Description: ตารางแสดงรายละเอียดของ Platform ที่ใช้ในการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTPF_ID	รหัสประเภท Platform	INTEGER	PK	/	
2	UTPF_NAME	ชื่อ Platform	VARCHAR(10)		/	
3	UTPF_VERSION	เวอร์ชันของ Platform	VARCHAR(10)			
4	UTPF_ARCHITECH	สถาปัตยกรรม	VARCHAR(100)			

ตารางที่ 4.16 ตารางความสัมพันธ์ของ UTD_REPOSITORY, UTD_PROJECT, และ UTD_TESTSCRIPT (UTD_SCRIPTINFO)**Table Description:** ตารางแสดงความสัมพันธ์ระหว่าง UTD_REPOSITORY, UTD_PROJECT, และ UTD_TESTSCRIPT

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTSI_ID	รหัสข้อมูลในเอกสารสคริปต์	INTEGER	PK	/	
2	UTRE_ID	รหัสคลังข้อมูล	INTEGER	FK	/	UTD_REPOSITORY
3	UTTS_ID	รหัส TEST SCRIPT	INTEGER	FK	/	UTD_TESTSCRIPT
4	UTPR_ID	รหัสโครงการ	INTEGER	FK	/	UTD_PROJECT

ตารางที่ 4.17 ตารางแสดงรายละเอียดของเอกสารสคริปต์ (UTD_SCRIPTFILE)**Table Description:** ตารางแสดงรายละเอียดของเอกสารสคริปต์สำหรับการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTSC_ID	รหัสไฟล์สคริปต์การทดสอบ	INTEGER	PK	/	
2	UTSC_FILENAME	ชื่อไฟล์สคริปต์	VARCHAR(20)		/	
3	UTSC_XMLFILENAME	ชื่อไฟล์เอกสาร XML การทดสอบ	VARCHAR(20)		/	
4	UTSC_CREATIONDATE	วันที่สร้างสคริปต์	TIMESTAMP		/	
5	UTTS_ID	รหัส TEST SCRIPT	INTEGER	FK	/	UTD_TESTSCRIPT
6	UTUS_ID	รหัสสร้างสคริปต์	INTEGER	FK		UTD_USER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.18 ตารางความสัมพันธ์ของเอกสารสคริปต์ และ Folder สำหรับการจัดเก็บเอกสาร
สคริปต์ (UTD_SCRIPTFOLDER)**

Table Description: ตารางแสดงความสัมพันธ์ระหว่างเอกสารสคริปต์ และ Folder สำหรับการ
จัดเก็บเอกสารสคริปต์

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTSI_ID	รหัสข้อมูลในเอกสาร สคริปต์	INTEGER	PK,FK	/	UTD_SCRIPTINFO
2	UTFO_ID	รหัส FOLDER	INTEGER	PK,FK	/	UTD_FOLDER

ตารางที่ 4.19 ตารางแสดงรายละเอียดของ Folder (UTD_FOLDER)

Table Description: ตารางแสดงรายละเอียดของ Folder ที่ใช้ในการเก็บข้อมูลต่าง ๆ ที่เกี่ยวข้องกับ
การทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTFO_ID	รหัส Folder	INTEGER	PK	/	
2	UTFO_NAME	ชื่อ Folder	VARCHAR(100)		/	
3	UTFO_COMMENT	หมายเหตุ	VARCHAR(255)			
4	UTFO_FOLDER	Folder ที่อยู่ก่อนหน้า	INTEGER	FK		UTD_FOLDER
5	UTPR_ID	รหัสโครงการ	INTEGER	FK		UTD_PROJECT

ตารางที่ 4.20 ตารางแสดงรายละเอียดของคลังข้อมูล (UTD_REPOSITORY)

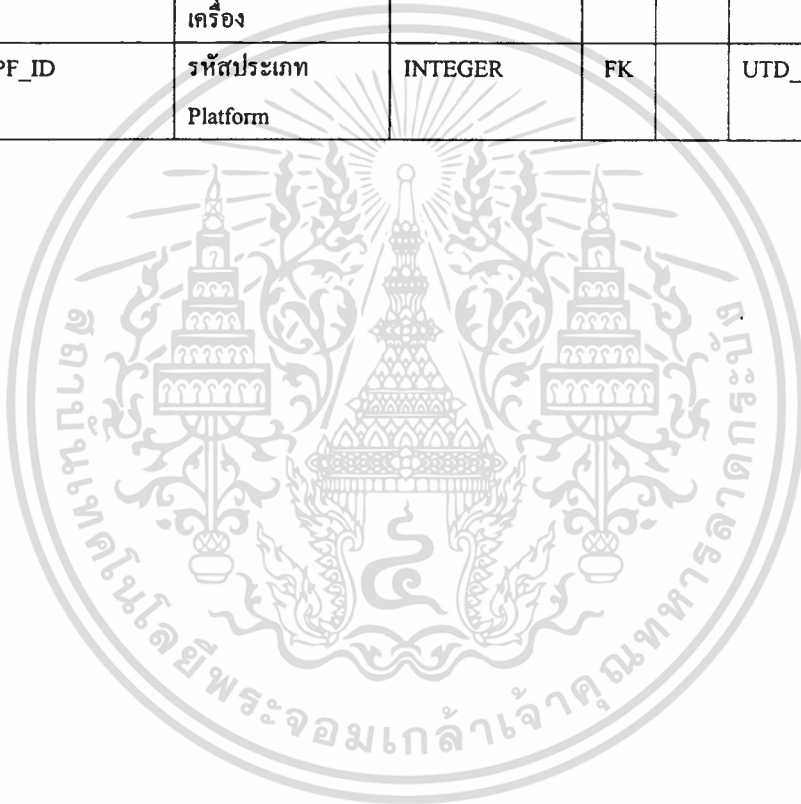
Table Description: ตารางแสดงรายละเอียดของคลังข้อมูลที่จัดเก็บกรณีการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTRE_ID	รหัสคลังข้อมูล	INTEGER	PK	/	
2	UTRE_NAME	ชื่อคลังข้อมูล	VARCHAR(100)		/	
3	UTTD_SETTING	ข้อกำหนดในการเข้าถึง คลังข้อมูล	VARCHAR(150)			

ตารางที่ 4.21 ตารางแสดงรายละเอียดของเครื่องที่ใช้ในการทดสอบ (UTD_MACHINES)

Table Description: ตารางแสดงรายละเอียดของเครื่องที่ใช้ในการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTMC_ID	รหัสเครื่อง	INTEGER	PK	/	
2	UTMC_NAME	ชื่อเครื่อง	VARCHAR(10)		/	
3	UTMC_IP	IP ของเครื่อง	VARCHAR(20)		/	
4	UTMC_DNS	Domain Name Server ของเครื่อง	VARCHAR(100)		/	
5	UTMC_COMPILER	ตัวแปลภาษาของ เครื่อง	VARCHAR(100)			
6	UTPF_ID	รหัสประเภท Platform	INTEGER	FK		UTD_PLATFORM



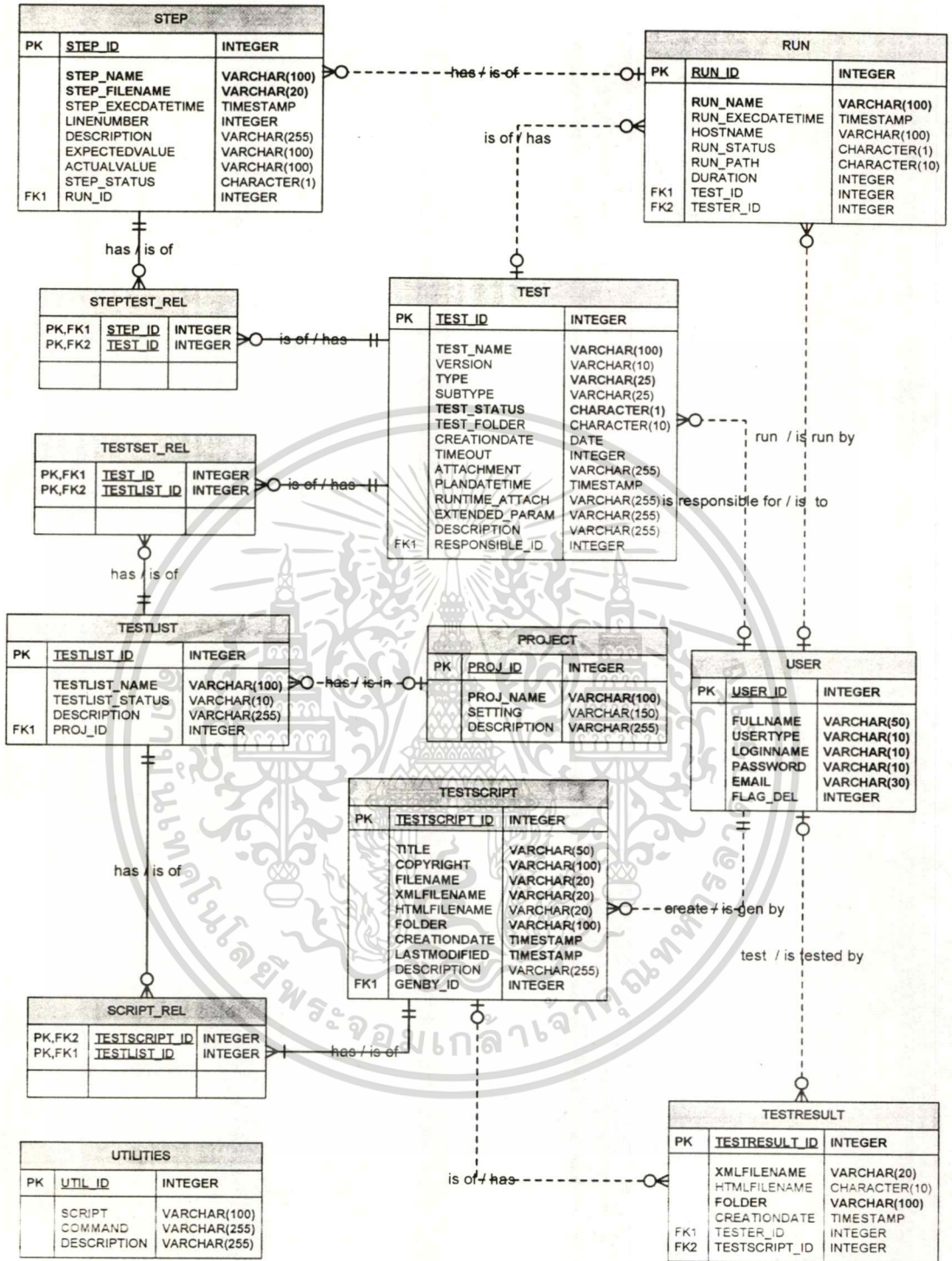
4.1.2 ฐานข้อมูลของระบบที่กำลังพัฒนา (KMTD)

ในการวิเคราะห์เพื่อออกแบบฐานข้อมูลของระบบ และหาความสัมพันธ์ของข้อมูลที่ใช้ในระบบที่กำลังพัฒนาที่มีชื่อว่า KMTD จะมีรายละเอียดของตารางของฐานข้อมูลดังแสดงในตารางที่ 4.1 และพิจารณาในรูปแบบของ Entity Relationship Diagram ดังรูปที่ 4.1

ตารางที่ 4.22 แสดงตารางทั้งหมดบนฐานข้อมูลของระบบที่กำลังพัฒนา (KMTD)

ลำดับที่	ชื่อตาราง	คำอธิบาย
1.	USER	ตารางแสดงรายละเอียดข้อมูลการเข้าสู่ระบบของผู้ใช้ระบบ
2.	PROJECT	ตารางแสดงรายละเอียดโครงการที่จะทำการทดสอบ
3.	TESTLIST	ตารางแสดงรายละเอียดกลุ่มการทดสอบ
4.	TEST	ตารางแสดงรายละเอียดกรณีการทดสอบ
5.	TESTSET_REL	ตารางแสดงความสัมพันธ์ระหว่างกลุ่มและกรณีของการทดสอบ
6.	STEP	ตารางแสดงรายละเอียดขั้นตอนการทดสอบ
7.	STEPTEST_REL	ตารางแสดงความสัมพันธ์ระหว่างขั้นตอนการทดสอบและกรณีการทดสอบ
8.	RUN	ตารางแสดงรายละเอียดของการทดสอบที่ได้กระทำไป
9.	TESTSCRIPT	ตารางแสดงรายละเอียดเอกสารของการทดสอบ
10.	SCRIPT_REL	ตารางแสดงความสัมพันธ์ระหว่างเอกสารและกลุ่มการทดสอบที่ได้กำหนดไว้
11.	TESTRESULT	ตารางแสดงรายละเอียดของเอกสารผลการทดสอบ
12.	UTILITIES	ตารางแสดงรายละเอียดกลุ่มคำสั่งที่ช่วยในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 Entity Relationship Diagram สำหรับระบบที่กำลังพัฒนาบนพื้นฐานหลักการงาน ยูนิเวอร์แซลเทสตีโดรเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.23 ตารางกลุ่มการทดสอบ (USER)

Table Description: แสดงรายละเอียดข้อมูลการเข้าสู่ระบบของผู้ใช้ระบบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	USER_ID	รหัสผู้ใช้	INTEGER	PK	/	
2	FULLNAME	ชื่อผู้ใช้	VARCHAR(50)		/	
3	USERTYPE	ประเภทผู้ใช้	VARCHAR(10)		/	
4	LOGINNAME	ชื่อในการเข้าใช้ระบบ	VARCHAR(10)		/	
5	PASSWORD	รหัสในการเข้าใช้ระบบ	VARCHAR(10)		/	
6	EMAIL	E-mail Address ของผู้ใช้	VARCHAR(30)		/	
7	FLAG_DEL	รหัสการยกเลิกผู้ใช้	INTEGER		/	

ตารางที่ 4.24 ตารางโครงการ (PROJECT)

Table Description: แสดงรายละเอียดรายละเอียดโครงการที่จะทำการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	PROJ_ID	รหัสโครงการ	INTEGER	PK	/	
2	PROJ_NAME	ชื่อโครงการ	VARCHAR(100)		/	
3	SETTING	ข้อกำหนดเพื่อการเข้าถึง	VARCHAR(150)			
4	DESCRIPTION	หมายเหตุ	VARCHAR(255)			

ตารางที่ 4.25 ตารางกลุ่มการทดสอบ (TESTLIST)

Table Description: ตารางแสดงรายละเอียดกลุ่มการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	TESTLIST_ID	รหัสกลุ่มการทดสอบ	INTEGER	PK	/	
2	TESTLIST_NAME	ชื่อกลุ่มการทดสอบ	VARCHAR(100)		/	
3	TESTLIST_STATUS	สถานะของกลุ่ม TEST เช่น 'open'	VARCHAR(10)			
4	DESCRIPTION	หมายเหตุ	VARCHAR(255)			
5	PROJ_ID	รหัสโครงการ	INTEGER	FK		PROJECT

ตารางที่ 4.26 ตารางกรณีการทดสอบ (TEST)

Table Description: ตารางแสดงรายละเอียดกรณีการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	TEST_ID	รหัส TEST	INTEGER	PK	/	
2	TEST_NAME	ชื่อ TEST	VARCHAR(100)		/	
3	VERSION	เวอร์ชัน	VARCHAR(10)			
4	TYPE	ประเภทของ TEST เช่น "Manual"	VARCHAR(25)		/	
5	SUBTYPE	ประเภทย่อยของ TEST	VARCHAR(25)			
6	TEST_STATUS	สถานะ (R=ready, D=Design)	CHARACTER(1)		/	
7	FOLDER	Folder ในการ ทดสอบ	VARCHAR(255)			
8	CREATIONDATE	วันที่สร้าง	DATE			
9	TIMEOUT	ระยะเวลาที่กำหนด	INTEGER			
10	ATTACHMENT	เอกสารแนบ	VARCHAR(255)			
11	PLAN_DATETIME	วันเวลาที่กำหนด สำหรับทดสอบ	TIMESTAMP			
12	RUNTIME_ATTACH	เอกสารแนบในการ RUN	VARCHAR(255)			
13	EXTENDED_PARAM	พารามิเตอร์เพิ่มเติม	VARCHAR(255)			
14	DESCRIPTION	หมายเหตุ	VARCHAR(255)			
15	RESPONSIBLE_ID	รหัสผู้รับผิดชอบ	INTEGER	FK		USER

ตารางที่ 4.27 ตารางความสัมพันธ์ของกลุ่มและกรณีการทดสอบ (TESTSET_REL)

Table Description: ตารางแสดงความสัมพันธ์ระหว่างกลุ่มและกรณีของการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	TEST_ID	รหัส TEST	INTEGER	PK,FK	/	TEST
2	TESTLIST_ID	รหัส TEST LIST	INTEGER	PK,FK	/	TESTLIST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.28 ตารางขั้นตอนการทดสอบ (STEP)

Table Description: ตารางแสดงรายละเอียดขั้นตอนการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	STEP_ID	รหัส STEP	INTEGER	PK	/	
2	STEP_NAME	ชื่อ STEP	VARCHAR(100)		/	
3	STEP_EXECDATETIME	วันเวลาที่ ประมวลผล	TIMESTAMP			
4	STEP_FILENAME	ชื่อไฟล์สำหรับการ ทำ Step	VARCHAR(100)			
5	LINENUMBER	บรรทัดที่ทดสอบ	INTEGER			
6	DESCRIPTION	หมายเหตุ	VARCHAR(255)			
7	EXPECTEDVALUE	ค่าที่คาดหวังไว้	VARCHAR(100)			
8	ACTUALVALUE	ค่าที่ได้จริง	VARCHAR(100)			
9	STEP_STATUS	สถานะ(P=passed, F=failed)	CHARACTER(1)			
10	RUN_ID	รหัส RUN	INTEGER			

ตารางที่ 4.29 ตารางความสัมพันธ์ของขั้นตอนและกรณีการทดสอบ (STEPTEST_REL)

Table Description: ตารางแสดงความสัมพันธ์ระหว่างขั้นตอนการทดสอบและกรณีการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	STEP_ID	รหัส STEP	INTEGER	PK,FK	/	STEP
2	TEST_ID	รหัส TEST	INTEGER	PK,FK	/	TEST

ตารางที่ 4.30 ตารางข้อมูลการทดสอบที่ได้กระทำไป (RUN)

Table Description: ตารางแสดงรายละเอียดของการทดสอบที่ได้กระทำไป

No.	Attribute	Description	Data Type	Key	Req	Reference
1	RUN_ID	รหัส RUN	INTEGER	PK	/	
2	RUN_NAME	ชื่อ RUN	VARCHAR(100)		/	
3	RUN_EXECDATETIME	วันเวลาที่ ประมวลผล	TIMESTAMP			
4	HOSTNAME	host_name ที่ ทดสอบ	VARCHAR(100)			

5	RUN_PATH	path ที่เกี่ยวข้อง	VARCHAR(100)			
6	RUN_STATUS	สถานะ (P=passed,F=failed)	CHARACTER(1)			
7	DURATION	ระยะเวลาที่ทดสอบ	INTEGER			
8	TEST_ID	รหัส TEST	INTEGER	FK		TEST
9	TESTER_ID	รหัส ผู้ทดสอบ	INTEGER	FK		USER

ตารางที่ 4.31 ตารางเอกสารสำหรับการทดสอบ (TESTSCRIPT)

Table Description: ตารางแสดงรายละเอียดเอกสารของการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	TESTSCRIPT_ID	รหัสเอกสารสคริปต์	INTEGER	PK	/	
2	TITLE	ชื่อเอกสารสคริปต์	VARCHAR(50)		/	
3	COPYRIGHT	ลิขสิทธิ์	VARCHAR(100)		/	
4	FILENAME	ชื่อไฟล์สคริปต์	VARCHAR(20)		/	
5	XMLFILENAME	ชื่อไฟล์ XML การทดสอบ	VARCHAR(20)		/	
6	HTMLFILENAME	ชื่อไฟล์ HTML สคริปต์	VARCHAR(20)		/	
7	FOLDER	Folder ที่เก็บสคริปต์	VARCHAR(100)		/	
8	CREATIONDATE	วันที่สร้างสคริปต์	TIMESTAMP		/	
9	LASTMODIFIED	วันที่เปลี่ยนแปลงสคริปต์	TIMESTAMP		/	
10	DESCRIPTION	หมายเหตุ	VARCHAR(255)			
11	GENBY_ID	รหัสสร้างสคริปต์	INTEGER	FK	/	USER

ตารางที่ 4.32 ตารางความสัมพันธ์ของเอกสารและกลุ่มการทดสอบ (SCRIPT_REL)

Table Description: ตารางแสดงความสัมพันธ์ระหว่างเอกสารและกลุ่มการทดสอบที่ได้กำหนดไว้

No.	Attribute	Description	Data Type	Key	Req	Reference
1	TESTSCRIPT_ID	รหัสเอกสารสคริปต์สำหรับ ทดสอบ	INTEGER	PK,FK	/	TESTSCRIPT
2	TESTLIST_ID	รหัสกลุ่ม TEST	INTEGER	PK,FK	/	TESTLIST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.33 ตารางเอกสารผลการทดสอบ (XMLRESULT)

Table Description: ตารางแสดงรายละเอียดของเอกสารผลของการทดสอบ

No.	Attribute	Description	Data Type	Key	Req	Reference
1	XMLRESULT_ID	รหัสไฟล์ผลการทดสอบ	INTEGER	PK	/	
2	XMLFILENAME	ชื่อไฟล์	VARCHAR(20)		/	
3	HTMLFILENAME	ชื่อไฟล์ HTML	VARCHAR(20)			
4	FOLDER	Folder ที่เก็บ	VARCHAR(100)		/	
5	CREATIONDATE	วันที่สร้าง	TIMESTAMP		/	
6	SCRIPTFILE_ID	รหัส SCRIPT FILE	INTEGER	FK	/	SCRIPTFILE
7	TESTER_ID	รหัสผู้ทดสอบสคริปต์	INTEGER	FK		USER

ตารางที่ 4.34 ตารางคำสั่งที่ช่วยในการทดสอบ (UTILITIES)

Table Description: ตารางแสดงรายละเอียดกลุ่มคำสั่งที่ช่วยในการทดสอบ

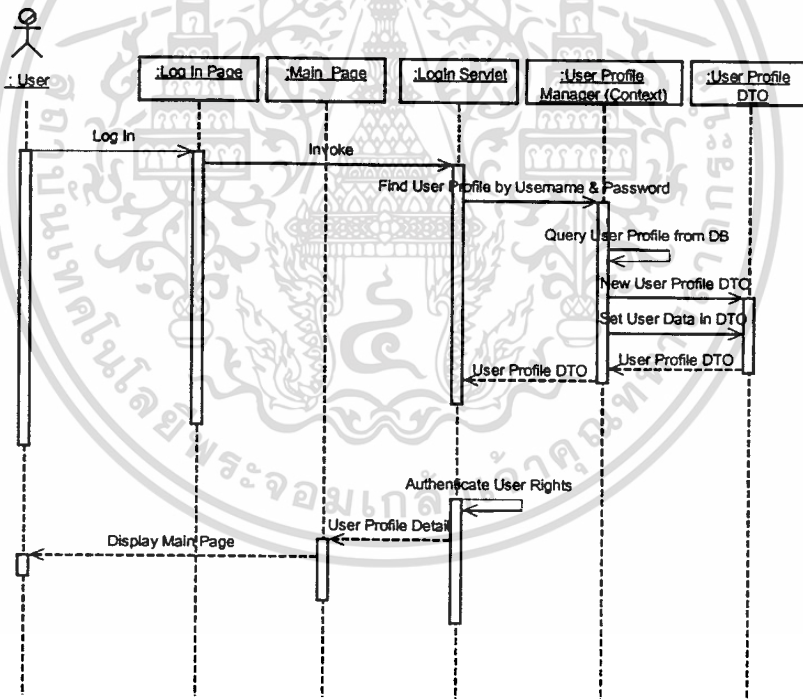
No.	Attribute	Description	Data Type	Key	Req	Reference
1	UTUL_ID	รหัส Utilities	INTEGER	PK	/	
2	SCRIPT	สคริปต์ของ Utilities	VARCHAR(100)		/	
3	COMMAND	คำสั่งในการทำงาน	VARCHAR(255)		/	
4	DESCRIPTION	หมายเหตุ	VARCHAR(255)			

4.2 การออกแบบอินเทอร์แอคชันไดอะแกรม (Interaction Diagrams)

จากลำดับการทำงานที่ได้วิเคราะห์ไว้ในแต่ละยูสเคสสามารถนำมาออกแบบรายละเอียดและเขียนเป็นไดอะแกรมโดยสามารถแสดงลำดับเหตุการณ์ต่าง ๆ ที่เกิดขึ้นภายในยูสเคสเพื่ออธิบายขั้นตอนของการดำเนินงานต่าง ๆ ได้ชัดเจนขึ้น ไดอะแกรมที่ใช้อธิบายประกอบด้วยไดอะแกรม 2 ประเภทได้แก่ ซีควีนซ์ไดอะแกรม และคอลลาบอเรนซ์ไดอะแกรม ซึ่งในการออกแบบระบบจะได้ซีควีนซ์ไดอะแกรมของแต่ละยูสเคสดังต่อไปนี้

4.2.1 ซีควีนซ์ไดอะแกรมของยูสเคสการเข้าใช้ระบบและตรวจสอบสิทธิ์ (Authenticate)

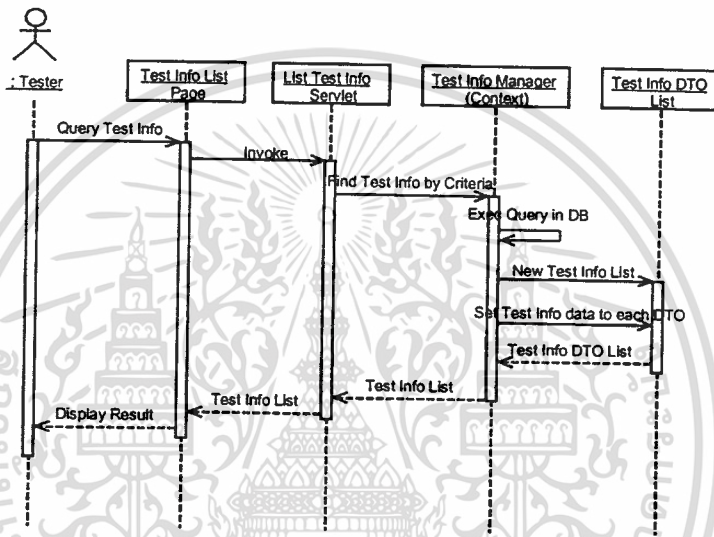
ซีควีนซ์ไดอะแกรมของยูสเคสการเข้าใช้ระบบและตรวจสอบสิทธิ์มีลำดับการทำงานดังนี้ User ทำการ Log in ไปยัง Log in Page และทำการเรียก User Profile โดยผ่าน Servlet ระบบทำการค้นหา User Profile จากฐานข้อมูลระบบจะส่ง User Profile DTO ที่บันทึกข้อมูลที่ค้นหาได้กลับไปจากนั้นระบบทำการให้สิทธิ์กับ User และส่ง User Profile Detail ผ่านทาง Servlet ระบบหน้าจอหลักให้กับผู้ใช้ตามสิทธิ์



รูปที่ 4.3 ซีควีนซ์ไดอะแกรมของการเข้าใช้ระบบและตรวจสอบสิทธิ์ (Authenticate)

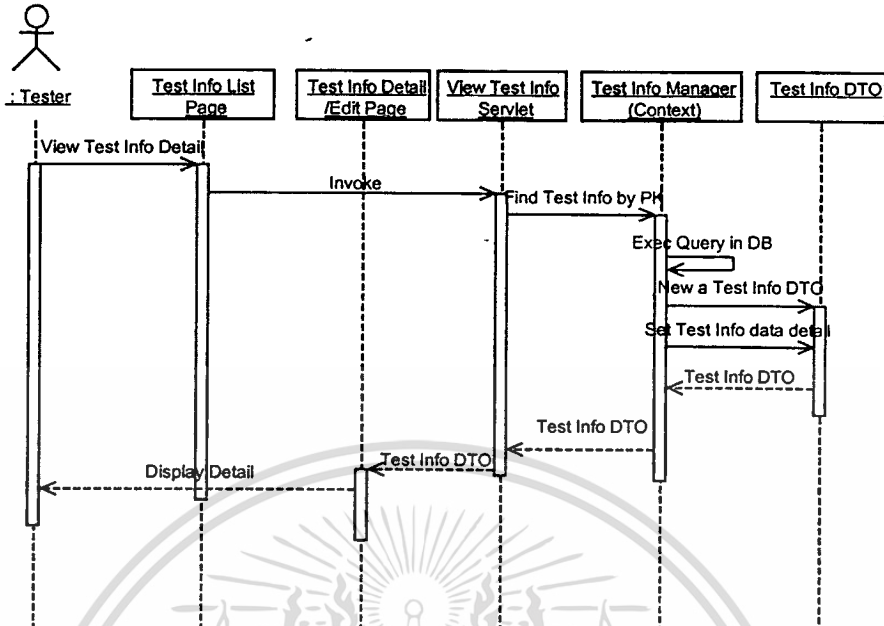
4.2.2 ซีเคิวนซ์ไคอะแกรมของยูสเคสการเรียกดูข้อมูลสำหรับการทดสอบ (Query Test Info)

ซีเคิวนซ์ไคอะแกรมของยูสเคสการเรียกดูข้อมูลสำหรับการทดสอบมีลำดับการทำงานแบ่งเป็นการแสดงรายการข้อมูลสำหรับการทดสอบ และการแสดงรายละเอียดของแต่ละรายการ โดยการแสดงรายการข้อมูลสำหรับการทดสอบมีลำดับการทำงานดังนี้ Tester ทำการค้นหาข้อมูลสำหรับการทดสอบผ่านทาง List test info servlet ระบบทำการค้นหาข้อมูลสำหรับการทดสอบตามเงื่อนไขที่กำหนดจากฐานข้อมูล และส่งรายการ Test Info DTO ที่บันทึกข้อมูลที่ค้นหาได้กลับมา จากนั้นระบบจะส่งรายการ Test Info ผ่านทาง Servlet มาแสดงผลให้ Tester



รูปที่ 4.4 ซีเคิวนซ์ไคอะแกรมของการแสดงรายการข้อมูลสำหรับการทดสอบ (List Test Info)

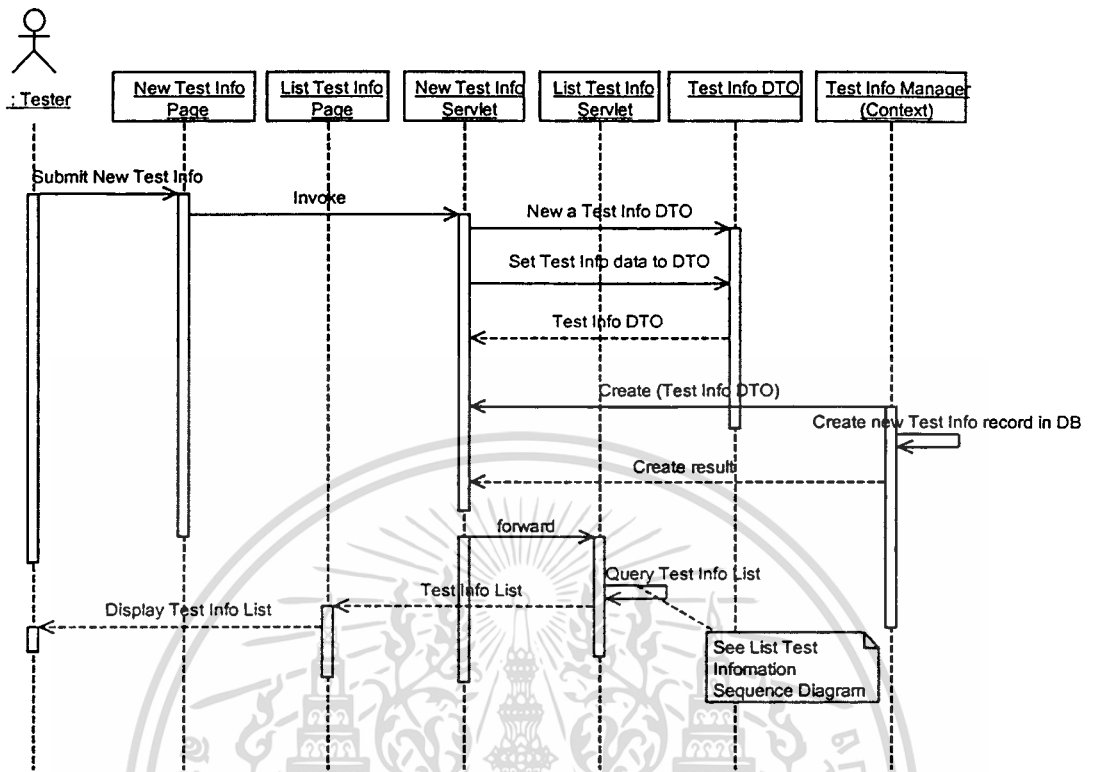
ซีเคิวนซ์ไคอะแกรมของการแสดงรายละเอียดของข้อมูลสำหรับการทดสอบมีลำดับการทำงานดังนี้ Tester เรียกดูรายละเอียดของข้อมูลสำหรับการทดสอบผ่าน Servlet ระบบค้นหาในข้อมูลตาม PK จากฐานข้อมูล และส่ง Test Info DTO ที่บันทึกข้อมูลที่ค้นหาได้กลับมาผ่านทาง Servlet ระบบแสดงรายละเอียดของข้อมูลสำหรับการทดสอบให้กับ Tester



รูปที่ 4.5 ซีควีนซ์ไดอะแกรมของการแสดงรายละเอียดของข้อมูลสำหรับการทดสอบ (View Test Info)

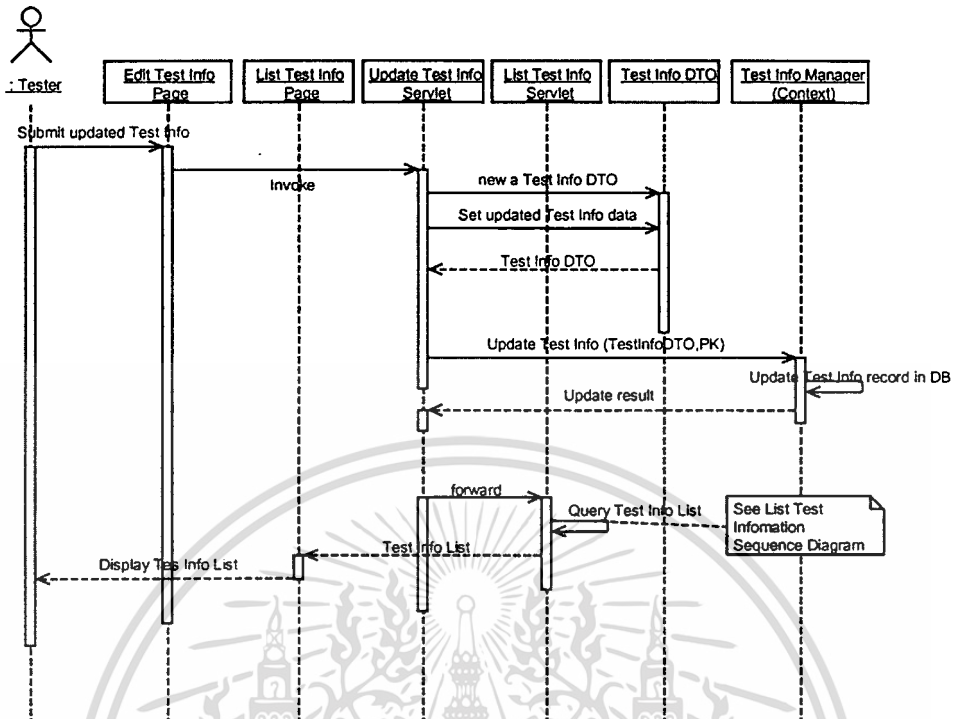
4.2.3 ซีควีนซ์ไดอะแกรมของยูสเคสจัดการเกี่ยวกับข้อมูลสำหรับการทดสอบ (Manage Test Info)

ซีควีนซ์ไดอะแกรมของยูสเคสจัดการเกี่ยวกับข้อมูลสำหรับการทดสอบ มีลำดับการทำงานแบ่งเป็น การสร้าง การแก้ไข และการลบข้อมูลสำหรับการทดสอบ สำหรับการสร้างข้อมูลสำหรับการทดสอบมีลำดับการทำงานดังนี้ Tester ส่งข้อมูลการทดสอบใหม่ที่จะทำการสร้างผ่าน New Test Info Servlet ระบบทำการ Set Test Info Data ลง DTO แล้ว Test Info Manager ทำการบันทึกข้อมูลใหม่ลงฐานข้อมูล และคืนหารายการข้อมูลสำหรับการทดสอบส่งกลับไปแสดงให้ Tester



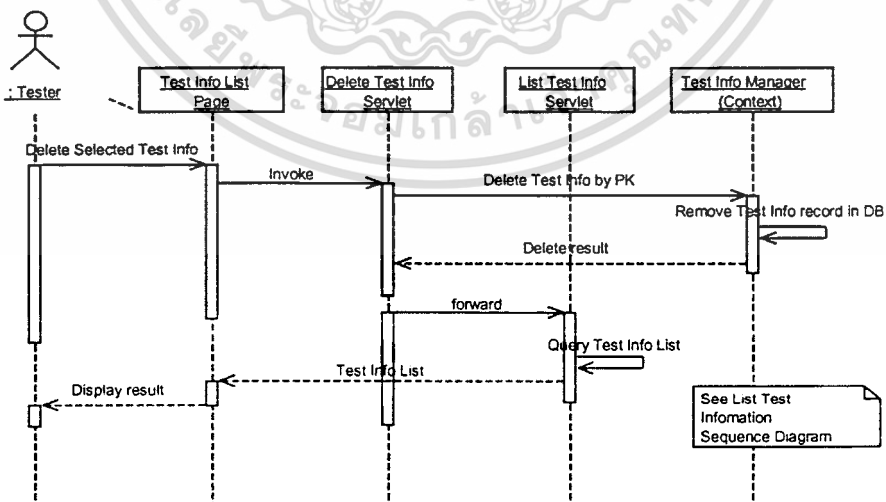
รูปที่ 4.6 ซีเค้นซ์ไดอะแกรมของการสร้างข้อมูลสำหรับการทดสอบ (Create Test Info)

ซีเค้นซ์ไดอะแกรมของการแก้ไขข้อมูลสำหรับการทดสอบ มีลำดับการทำงานดังนี้ Tester ส่งข้อมูลของการทดสอบที่จะทำการแก้ไขผ่านทาง servlet ระบบจะ Set ข้อมูลลง DTO และ บันทึกการแก้ไขลงฐานข้อมูล จากนั้นระบบจะค้นหารายการข้อมูลสำหรับการทดสอบส่งไปแสดงให้กับ Tester



รูปที่ 4.7 ซีควีนซ์ไดอะแกรมของการปรับเปลี่ยนข้อมูลสำหรับการทดสอบ (Edit Test Info)

ซีควีนซ์ไดอะแกรมของการลบข้อมูลสำหรับการทดสอบ มีลำดับการทำงานดังนี้ Tester ส่งข้อมูลสำหรับการทดสอบที่ต้องการลบ โดยผ่านทาง servlet ระบบทำการลบข้อมูลตาม PK จากฐานข้อมูล จากนั้นระบบจะค้นหารายการข้อมูลสำหรับการทดสอบส่งไปแสดงให้กับ Tester

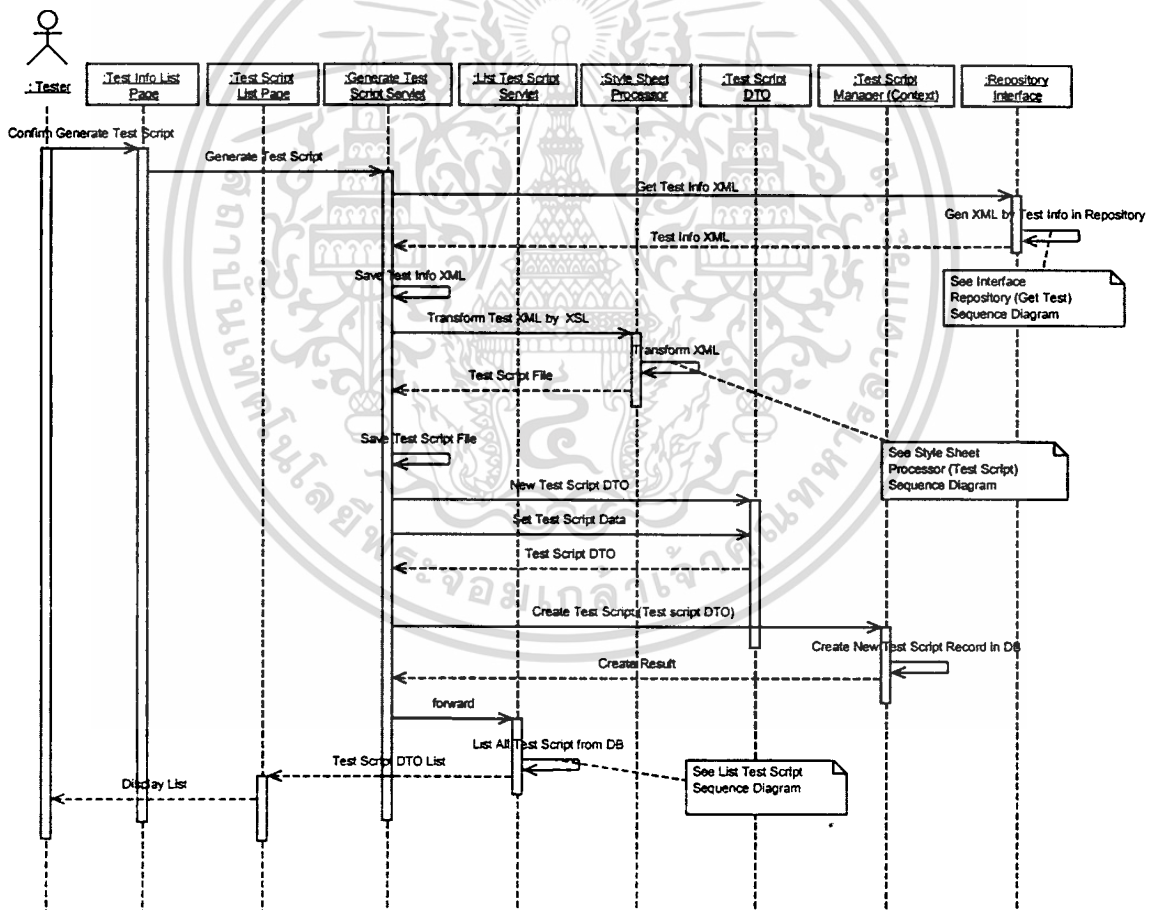


รูปที่ 4.8 ซีควีนซ์ไดอะแกรมของการลบข้อมูลสำหรับการทดสอบ (Delete Test Info)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 ซีเคิร์ฟเวอร์โค้ดของยูสเคสสร้างสคริปต์สำหรับการทดสอบ (Generate Test Script)

ซีเคิร์ฟเวอร์โค้ดของยูสเคสสร้างสคริปต์สำหรับการทดสอบ มีลำดับการทำงานดังนี้ Tester ทำการยืนยันการสร้างเอกสารสคริปต์สำหรับการทดสอบ จากนั้น Generate Test Script Servlet ระบบจะทำการดึงข้อมูลการทดสอบในรูปแบบเอกสารเอกซ์เอ็มแอลด้วยส่วนการเชื่อมต่อเอกซ์เอ็มแอลกับฐานข้อมูลและทำการบันทึกเป็นไฟล์ในระบบ จากนั้นจะทำการแปลงเอกสารเอกซ์เอ็มแอลที่ได้ด้วย XSL โดย Style Sheet Processor ก็จะได้เอกสารสคริปต์สำหรับการทดสอบ และทำการบันทึกเป็นไฟล์ในระบบ จากนั้นสร้าง Test Script DTO และ Set ข้อมูลที่เกี่ยวข้องกับ Test Script เพื่อให้ Test Script Manager ทำการบันทึกข้อมูลลงฐานข้อมูล จากนั้นระบบจะส่งต่อไปยัง List Test Script Servlet เพื่อแสดงรายการเอกสารสคริปต์ทั้งหมดจากฐานข้อมูล ซึ่งจะส่งไปแสดงให้กับ Tester

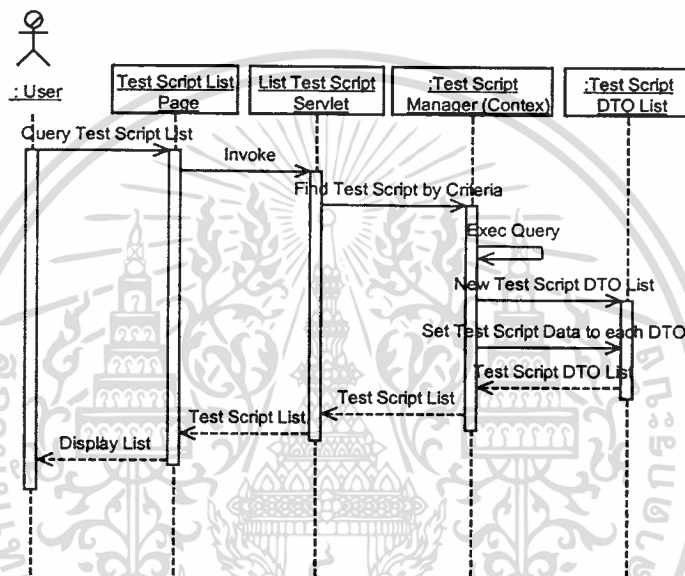


รูปที่ 4.9 ซีเคิร์ฟเวอร์โค้ดของยูสเคสการสร้างสคริปต์สำหรับการทดสอบ (Generate Test Script)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

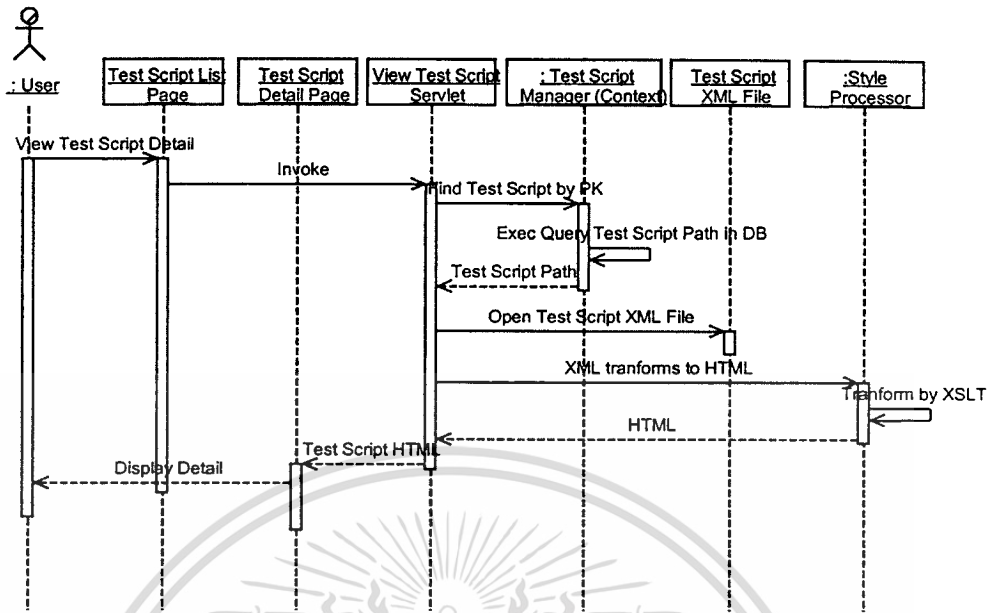
4.2.5 ซีเควินซ์ไคอะแกรมของยูสเคสเรียกดูรายละเอียดในสคริปต์สำหรับการทดสอบ (Query Test Scripts)

ซีเควินซ์ไคอะแกรมของยูสเคสเรียกดูรายละเอียดในสคริปต์สำหรับการทดสอบ มีลำดับการทำงานแบ่งเป็น การแสดงรายการข้อมูลเอกสารสคริปต์ และการแสดงรายละเอียดของเอกสารสคริปต์ ซึ่งการแสดงรายการมีลำดับการทำงานดังนี้ ผู้ใช้ต้องการค้นหารายการเอกสารสคริปต์สำหรับการทดสอบผ่านทาง servlet ระบบทำการค้นหาข้อมูลตามเงื่อนไขที่กำหนด ระบบส่งรายการเอกสารสคริปต์สำหรับการทดสอบกลับมาผ่าน Servlet ไปแสดงให้กับผู้ใช้



รูปที่ 4.10 ซีเควินซ์ไคอะแกรมของการแสดงรายการสคริปต์สำหรับการทดสอบ (List Test Script)

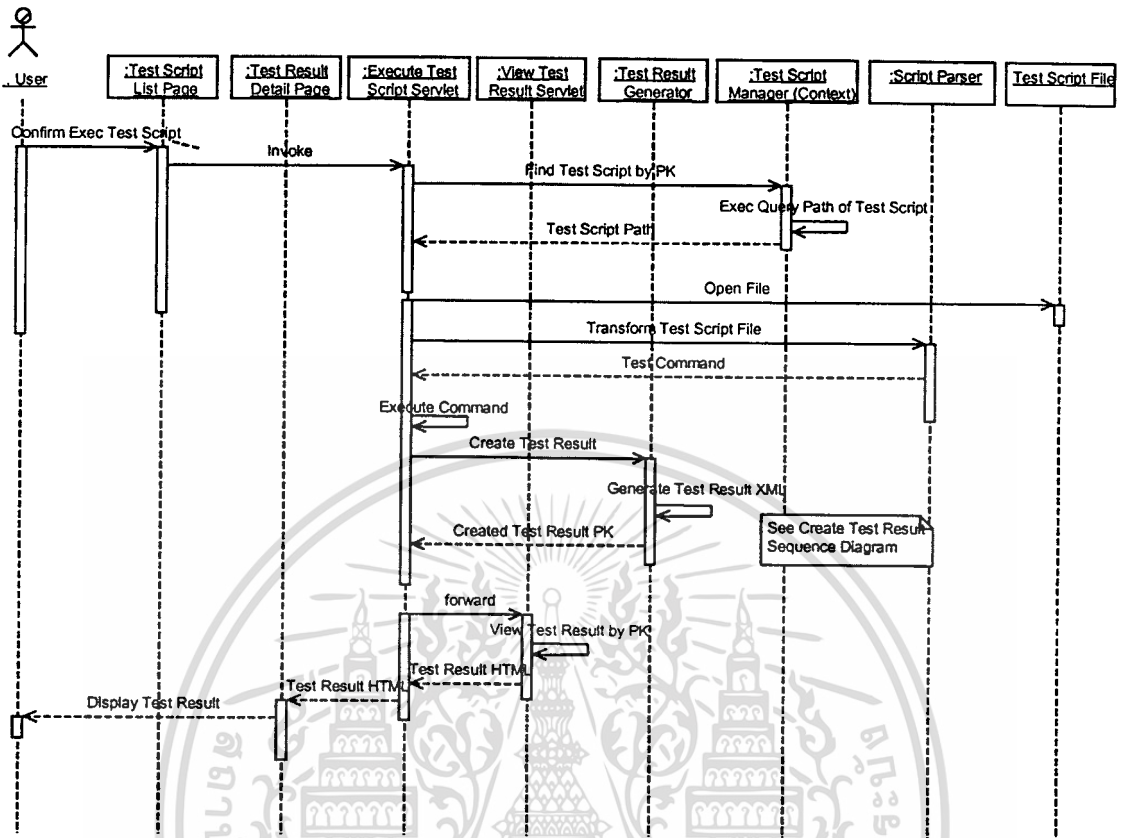
ซีเควินซ์ไคอะแกรมของการแสดงรายละเอียดของสคริปต์สำหรับการทดสอบ มีลำดับการทำงานดังนี้ ผู้ใช้ต้องการรายละเอียดของเอกสารสคริปต์โดยเรียกผ่าน Servlet ระบบทำการค้นหา ข้อมูลของเอกสารสคริปต์ที่เลือกในฐานข้อมูลตาม PK และส่งตำแหน่งที่เก็บเอกสารเอกซ์เอ็มแอลสำหรับการทดสอบ (path) กลับมา ระบบเปิดไฟล์ตาม path ที่ได้และแปลง XML ไปเป็น HTML โดย Style Processor และส่ง test script HTML ไปแสดงที่ผู้ใช้



รูปที่ 4.11 ซีควีนซ์ไดอะแกรมของการแสดงรายละเอียดของสคริปต์สำหรับการทดสอบ (View Test Script)

4.2.6 ซีควีนซ์ไดอะแกรมของยูสเคสจัดการเกี่ยวกับสคริปต์สำหรับการทดสอบ (Manage Test Scripts)

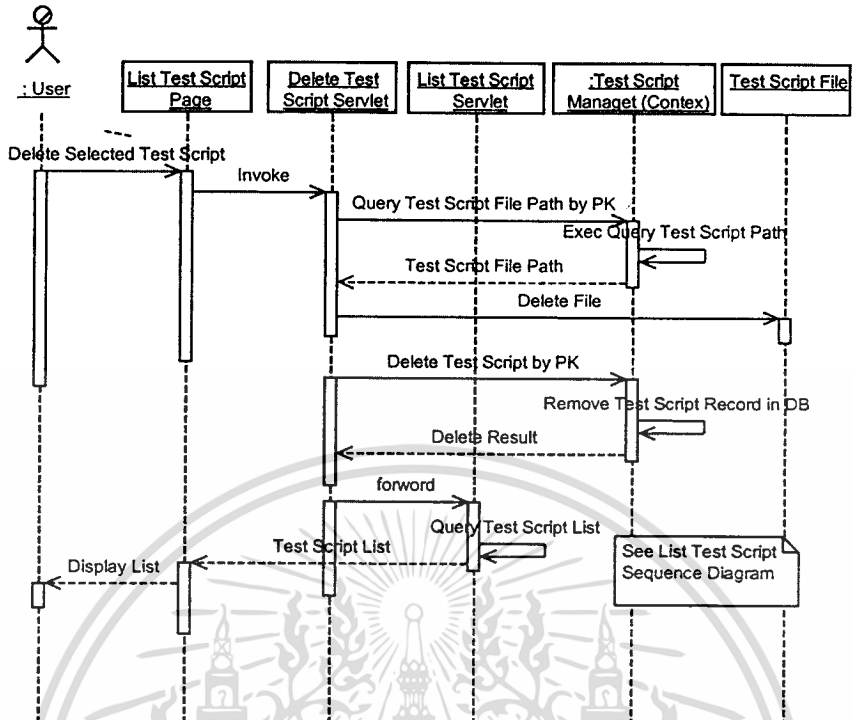
ซีควีนซ์ไดอะแกรมของการลบสคริปต์สำหรับการทดสอบ มีลำดับการทำงานดังนี้ ผู้ใช้ต้องการลบเอกสารสคริปต์สำหรับการทดสอบผ่านทาง Servlet ระบบทำการค้นหา Path ที่เก็บเอกสารสคริปต์ตาม PK และส่ง Path กลับไป จากนั้นทำการลบไฟล์ที่เกี่ยวข้องทั้งหมดจากระบบและ ลบข้อมูลอ้างอิงเกี่ยวกับเอกสารสคริปต์ออกจากฐานข้อมูล จากนั้นระบบจะค้นหารายการของเอกสารสคริปต์ในระบบแสดงผลให้กับผู้ใช้



รูปที่ 4.13 ซีควีนซ์ไดอะแกรมของการเรียกสคริปต์ทำการทดสอบโปรแกรม (Execute Test Script)

4.2.8 ซีควีนซ์ไดอะแกรมของยูสเคสเรียกดูผลการทดสอบที่ได้จากการทดสอบด้วยสคริปต์ (Query Test Results)

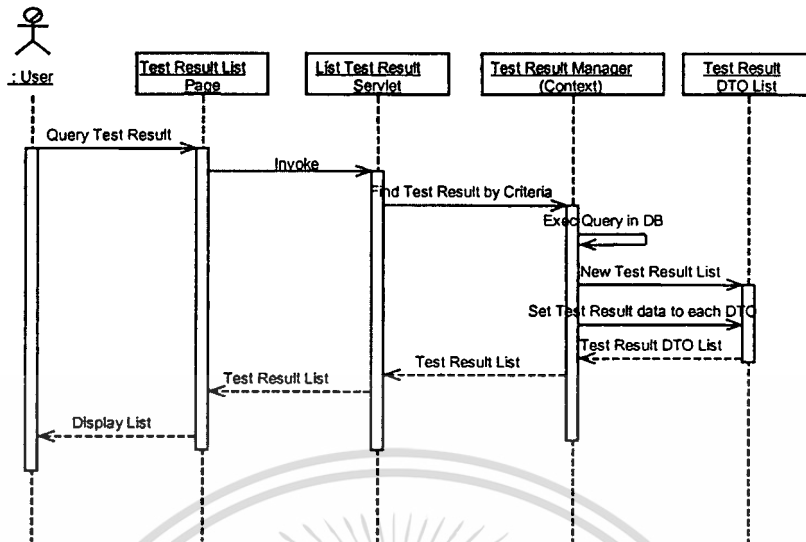
ซีควีนซ์ไดอะแกรมของของยูสเคสเรียกดูผลการทดสอบที่ได้จากการทดสอบด้วยสคริปต์ มีลำดับการทำงานแบ่งเป็น การแสดงรายการผลการทดสอบ และการแสดงรายละเอียดของผลการทดสอบ ซึ่งการแสดงผลการทดสอบมีลำดับการทำงานดังนี้ ผู้ใช้ต้องการให้แสดงผลการทดสอบโดยเรียกผ่าน List Test Results Servlet ระบบทำการค้นหาผลการทดสอบตามเงื่อนไขที่กำหนด และส่งมาแสดงให้กับผู้ใช้



รูปที่ 4.12 ซีเควินซ์โคอะแกรมของการลบสคริปต์สำหรับการทดสอบ (Delete Test Scripts)

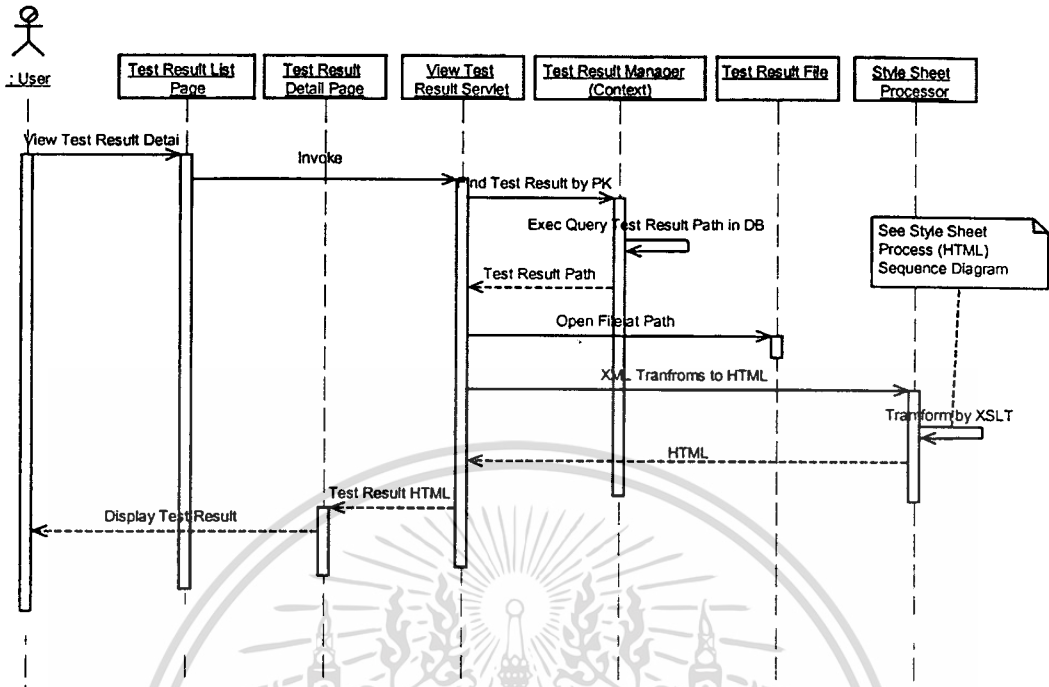
4.2.7 ซีเควินซ์โคอะแกรมของยูสเคสเรียกสคริปต์ทำการทดสอบโปรแกรม (Execute Test Script)

ซีเควินซ์โคอะแกรมของยูสเคสเรียกสคริปต์ทำการทดสอบโปรแกรมมีลำดับการทำงานดังนี้ ผู้ใช้ยืนยันการทำการทดสอบโปรแกรมผ่าน Execute Test Script Servlet ระบบทำการค้นหา Test Script ตาม PK และส่ง Path ของเอกสารสคริปต์กลับไป จากนั้นทำการเปิด File เอกสารสคริปต์และแปลงเอกสารให้อยู่ในรูปคำสั่งที่นำไปประมวลผลได้ จากนั้นทำการเรียกประมวลผลคำสั่งนั้น ๆ และสร้างผลการทดสอบโปรแกรมในรูปแบบเอกสารเอกซ์เอ็มแอล ด้วยส่วนการสร้างผลการทดสอบ ระบบจะแสดงรายละเอียดของผลการทดสอบในรูปแบบเอกซ์เอ็มแอลให้กับผู้ใช้



รูปที่ 4.14 ซีเควินซ์ไคอะแกรมของการแสดงรายการผลการทดสอบที่ได้จากการทดสอบด้วยสคริปต์ (List Test Results)

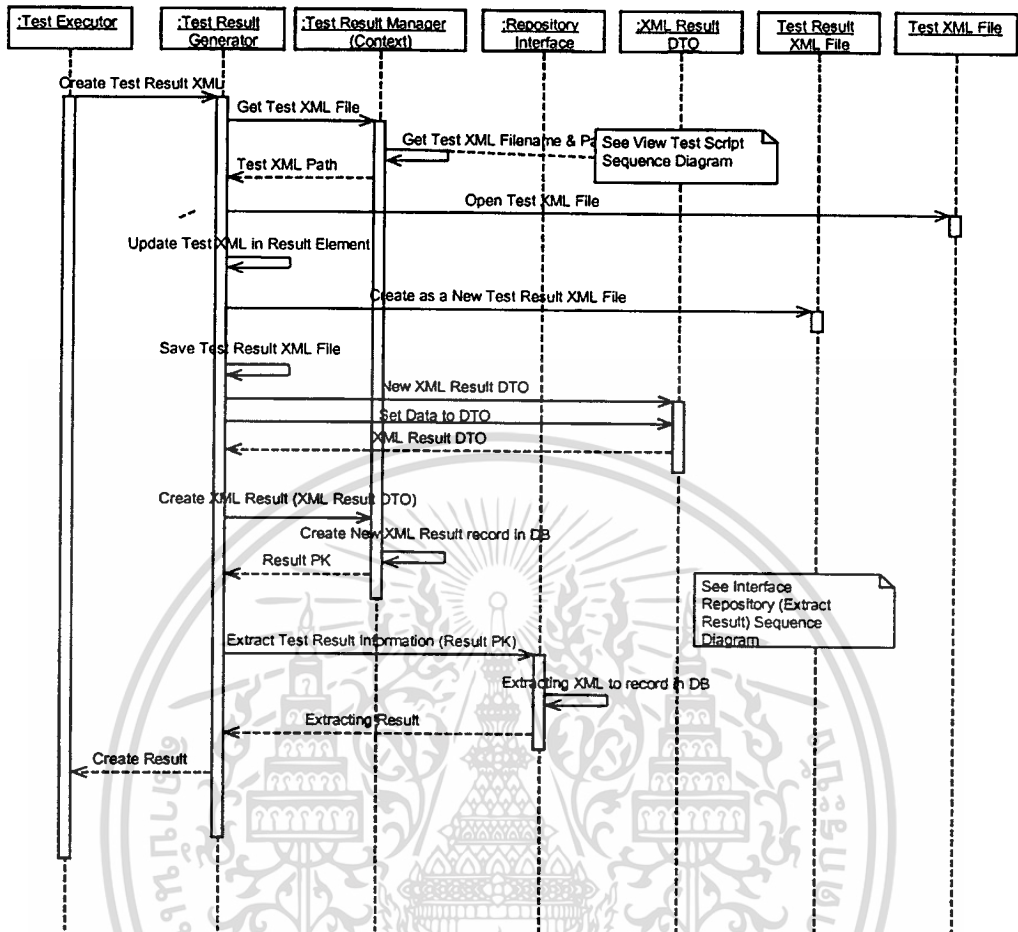
ซีเควินซ์ไคอะแกรมการแสดงผลรายละเอียดของผลการทดสอบที่ได้จากการทดสอบด้วยสคริปต์ มีลำดับการทำงานดังนี้ ผู้ใช้ต้องการดูรายละเอียด ระบบทำการค้นหา Path ที่เก็บเอกสารเอกซ์เอ็มแอลของผลการทดสอบ และเปิดไฟล์เพื่อนำไปแปลงให้อยู่ในรูปแบบเอกซ์เอ็มแอล จากนั้นแสดงผลให้กับผู้ใช้



รูปที่ 4.15 ซีเควินซ์ไคอะแกรมของการแสดงรายละเอียดของผลการทดสอบที่ได้จากการทดสอบด้วยสคริปต์ (View Test Result)

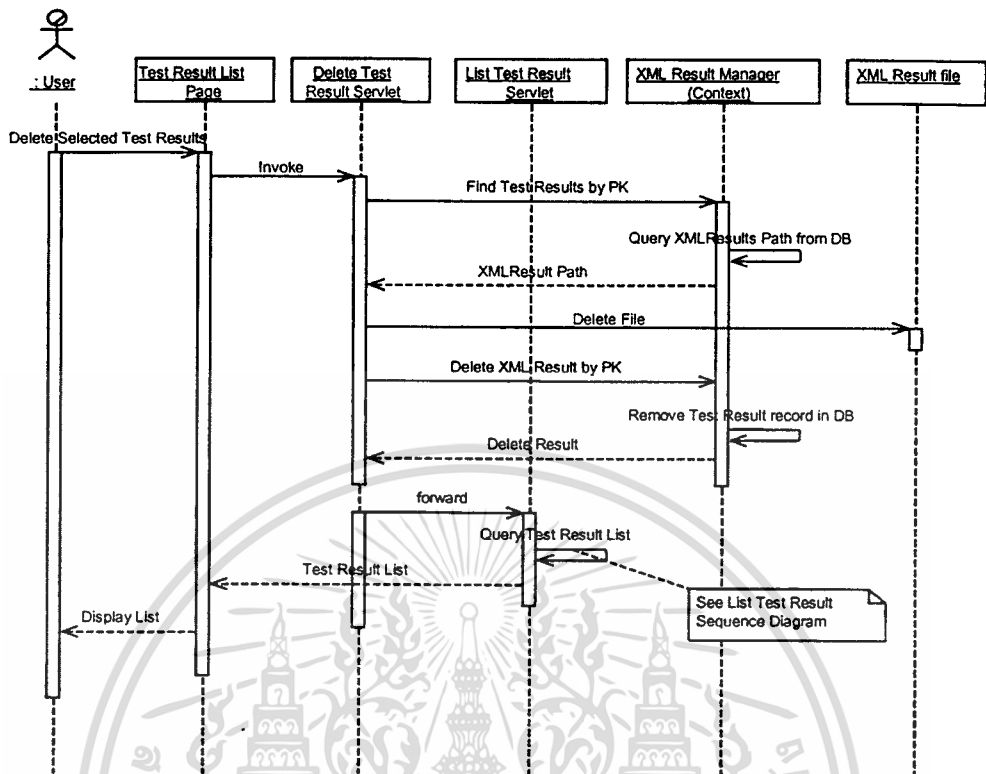
4.2.9 ซีเควินซ์ไคอะแกรมของยูสเคสจัดการเกี่ยวกับผลการทดสอบที่ได้จากการทดสอบด้วยสคริปต์(Manage Test Results)

ซีเควินซ์ไคอะแกรมของยูสเคสจัดการเกี่ยวกับผลการทดสอบที่ได้จากการทดสอบด้วยสคริปต์ มีลำดับการทำงานแบ่งเป็นการสร้างผลการทดสอบและการลบผลการทดสอบ ซึ่งการสร้างผลการทดสอบมีลำดับการทำงานดังนี้ ระบบทำการสร้างเอกสารเอกซ์เอ็มแอลของผลการทดสอบ โดยหน่วยการสร้างผลการทดสอบ ระบบจะดึงข้อมูลเกี่ยวกับการทดสอบเพื่อให้ได้ที่อยู่ของเอกสารเอกซ์เอ็มแอลของการทดสอบจากฐานข้อมูลและจะทำการเปิดเอกสารนั้นขึ้นมา จากนั้นจะทำการบันทึกข้อมูลเกี่ยวกับผลการทดสอบลงในเอกสาร (เป็น Element ของผลการทดสอบ) และทำการบันทึกเป็นเอกสารเอกซ์เอ็มแอลสำหรับการทดสอบอันใหม่ จากนั้น บันทึกข้อมูลอ้างอิงลงฐานข้อมูล และส่งหมายเลขอ้างอิงของเอกสารผลการทดสอบมาให้นำไปใช้ในการกระจายข้อมูลผลการทดสอบจากเอกสารเอกซ์เอ็มแอลบันทึกลงฐานข้อมูลด้วย ส่วนการเชื่อมต่อกับฐานข้อมูล (Interface to Repository)



รูปที่ 4.16 ซีควีนซ์ไดอะแกรมของการสร้างผลการทดสอบ (Create Test Results)

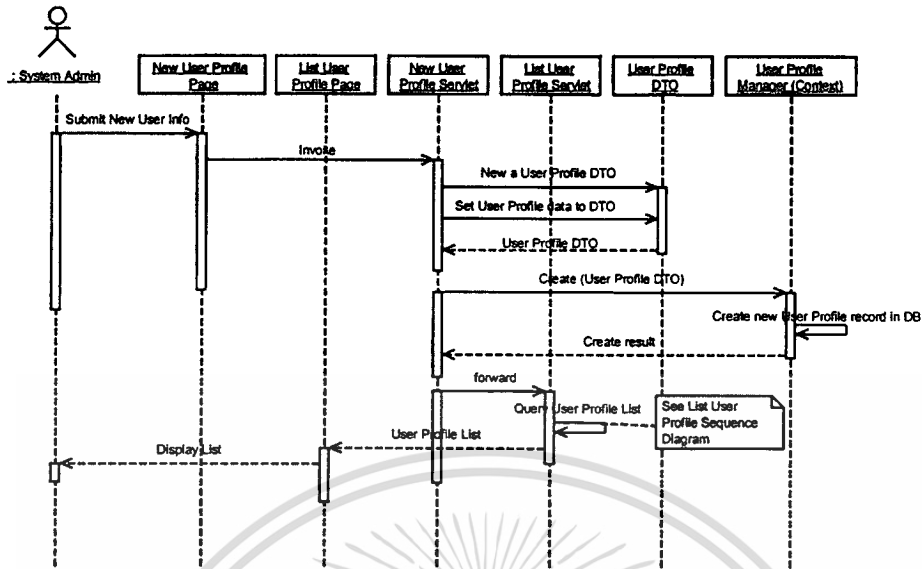
ซีควีนซ์ไดอะแกรมของการลบผลการทดสอบ มีลำดับการทำงานดังนี้ User เรียกทำการลบผลการทดสอบผ่าน Delete Test Result Servlet ระบบทำการค้นหา Test Result ตาม PK จากฐานข้อมูลและส่ง XML path กลับไป จากนั้นทำการลบไฟล์ที่อยู่ใน Path ดังกล่าวและลบข้อมูลอ้างอิงออกจากฐานข้อมูล จากนั้นแสดงรายการเอกสารผลการทดสอบให้กับผู้ใช้



รูปที่ 4.17 ซีควีนซ์ไดอะแกรมของการลบผลการทดสอบ (Delete Test Results)

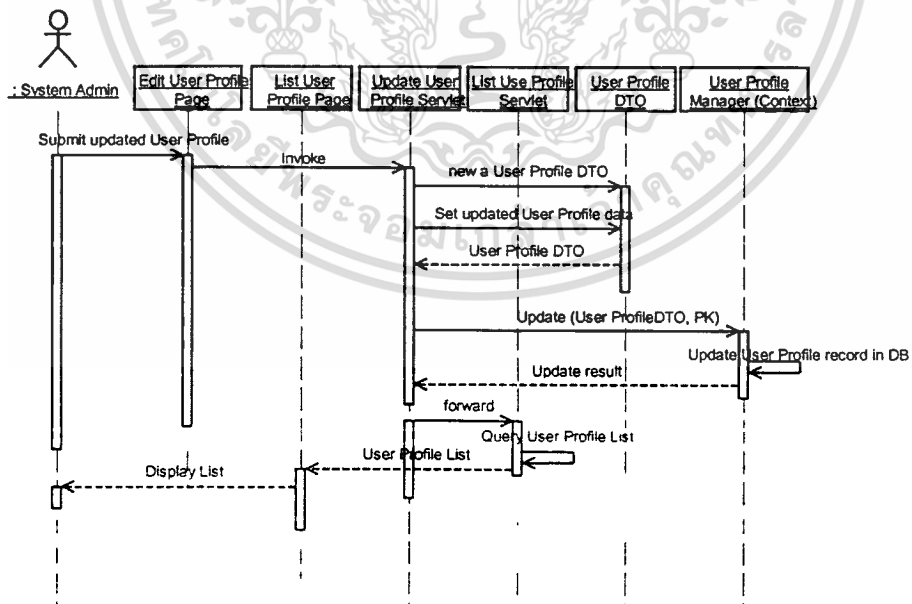
4.2.10 ซีควีนซ์ไดอะแกรมของยูสเคสจัดการเกี่ยวกับข้อมูลผู้ใช้ระบบ (Manage User Profile)

ซีควีนซ์ไดอะแกรมของยูสเคสจัดการเกี่ยวกับข้อมูลผู้ใช้ระบบ มีลำดับการทำงานแบ่งเป็นการสร้างผู้ใช้ในระบบ การแก้ไขข้อมูลผู้ใช้ การลบข้อมูลผู้ใช้ ซึ่งการสร้างผู้ใช้ในระบบมีลำดับการทำงานดังนี้ System Admin ทำการส่งข้อมูลผู้ใช้คนใหม่ให้กับระบบผ่าน New User Profile Servlet จากนั้น Set User Profile Data ลง DTO สร้าง New User Profile Record ลงในฐานข้อมูล จากนั้นทำการค้นหารายการผู้ใช้ส่งไปแสดงให้กับ System Admin



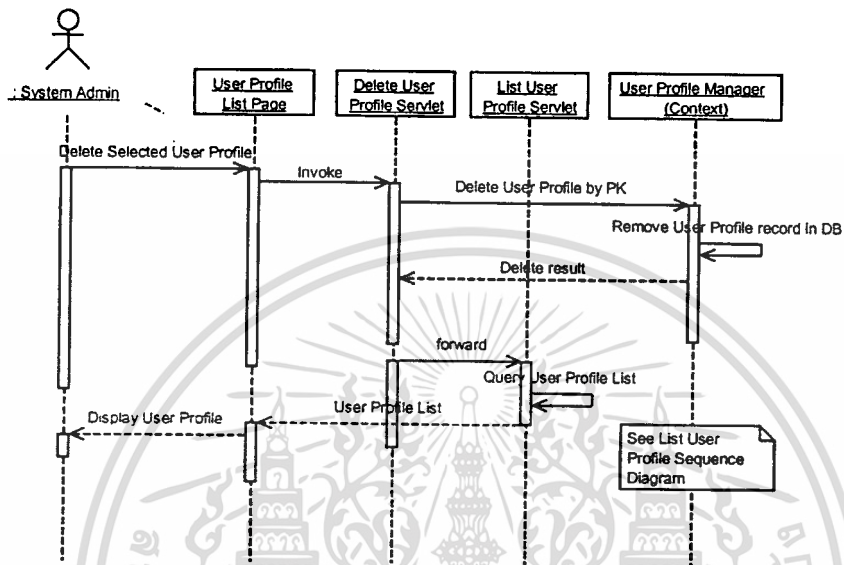
รูปที่ 4.18 ซีควีนซ์ไดอะแกรมของการสร้างข้อมูลผู้ใช้ระบบ (Create User Profile)

ลำดับการทำงานของแก้ไขข้อมูลผู้ใช้ระบบ มีลำดับการทำงานดังนี้ System Admin ทำการ Submit Update User Profile ให้กับระบบผ่าน Update User Profile Servlet จากนั้น Set Update User Profile Data ลง DTO จากนั้น Update User Profile Record ลงฐานข้อมูล จากนั้นทำการค้นหารายการผู้ใช้ส่งไปแสดงให้กับ System Admin



รูปที่ 4.19 ซีควีนซ์ไดอะแกรมของการแก้ไขข้อมูลผู้ใช้ระบบ (Edit User Profile)

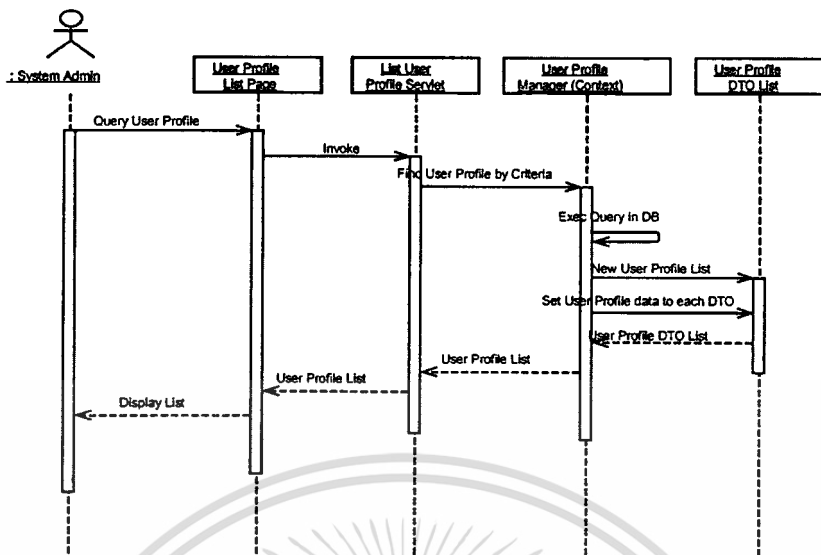
ซีเคิวนซ์ไคอะแกรมของการลบข้อมูลผู้ใช้ระบบ มีลำดับการทำงานดังนี้ System Admin ทำการลบข้อมูลผู้ใช้ที่เลือกผ่าน Delete User Profile Servlet และทำการลบข้อมูลออกจากรฐานข้อมูล จากนั้นทำการค้นหารายการผู้ใช้ส่ง ไปแสดงให้กับ System Admin



รูปที่ 4.20 ซีเคิวนซ์ไคอะแกรมของการลบข้อมูลผู้ใช้ระบบ (Delete User Profile)

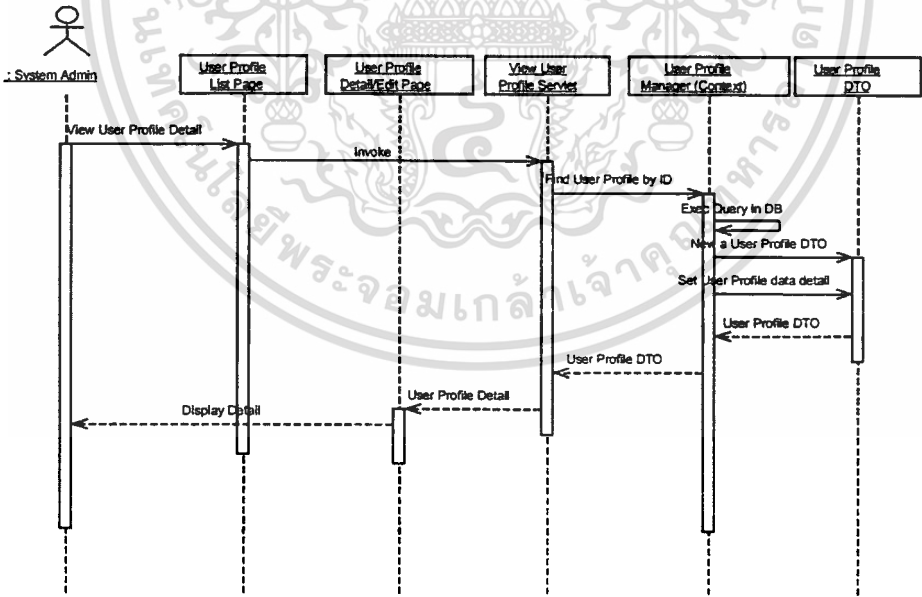
4.2.11 ซีเคิวนซ์ไคอะแกรมของยูสเควรี่ข้อมูลผู้ใช้ระบบ (Query User Profile)

ซีเคิวนซ์ไคอะแกรมของการเรียกดูข้อมูลผู้ใช้ระบบ มีลำดับการทำงานแบ่งเป็น การแสดงรายการข้อมูลผู้ใช้ และการแสดงรายละเอียดข้อมูลผู้ใช้ ซึ่งการแสดงรายการข้อมูลผู้ใช้มีลำดับการทำงานดังนี้ System Admin ต้องการค้นหารายการผู้ใช้ผ่าน List User Profile Servlet จากนั้นระบบทำการค้นหาจากรฐานข้อมูล และได้รับรายการข้อมูลผู้ใช้ส่ง ไปแสดงให้กับ System Admin



รูปที่ 4.21 ซีควีนซ์ไดอะแกรมของการแสดงรายการผู้ใช้ระบบ (List User Profile)

ซีควีนซ์ไดอะแกรมของการแสดงรายละเอียดข้อมูลผู้ใช้ระบบ มีลำดับการทำงานดังนี้ System Admin ต้องการรายละเอียดข้อมูลผู้ใช้ผ่าน View User Profile Servlet จากนั้นระบบทำการค้นหาจากฐานข้อมูล และได้รายละเอียดข้อมูลผู้ใช้ส่งไปแสดงให้กับ System Admin

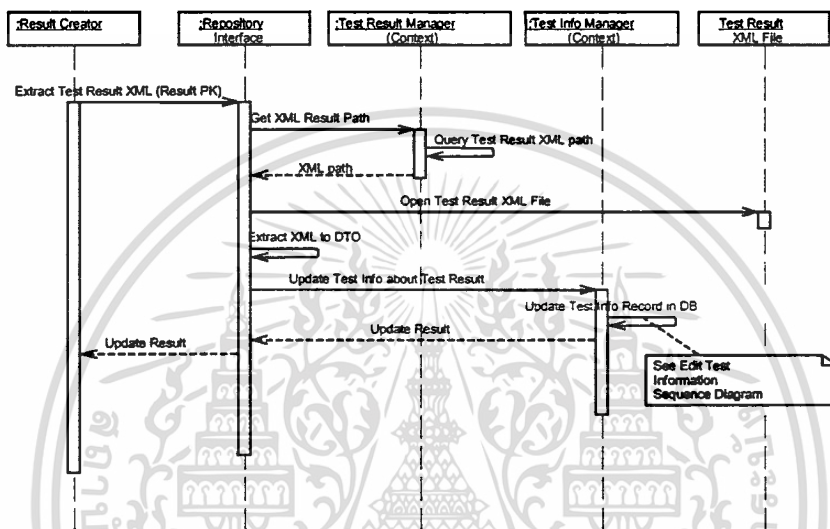


รูปที่ 4.22 ซีควีนซ์ไดอะแกรมของการแสดงรายละเอียดข้อมูลผู้ใช้ระบบ (View User Profile)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

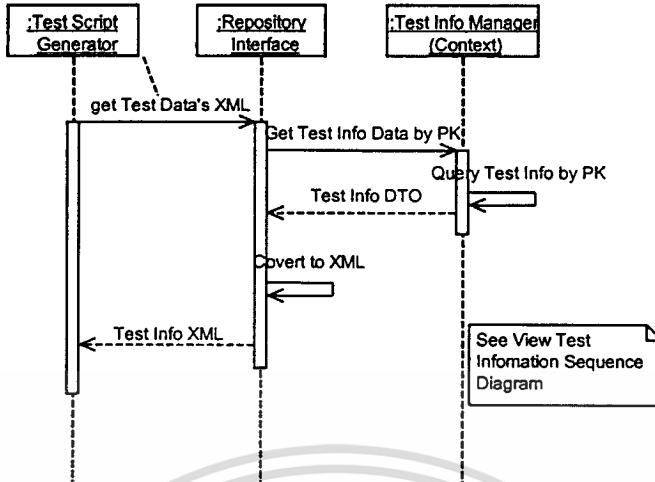
4.2.12 ซีเควินซี่โคอะแกรมของยูสเคสการเชื่อมต่อกับฐานข้อมูลด้วยเอกสารเอกซ์เอ็มแอล (Interface XML with Repository)

ซีเควินซี่โคอะแกรมของการเชื่อมต่อกับฐานข้อมูลในการกระจายข้อมูลจากเอกสารเอกซ์เอ็มแอลมีลำดับการทำงานดังนี้ ระบบทำการกระจายข้อมูลจากเอกสารเอกซ์เอ็มแอลของ Test Result เป็น DTO โดย Repository Interface จากนั้นบันทึกแก้ไขข้อมูลของการทดสอบเกี่ยวกับผลการทดสอบลงในฐานข้อมูลและส่งผลการบันทึกแก้ไขกลับ



รูปที่ 4.23 ซีเควินซี่โคอะแกรมของการเชื่อมต่อกับฐานข้อมูลในการกระจายข้อมูลจากเอกสารเอกซ์เอ็มแอล (Interface XML with Repository-Extracting)

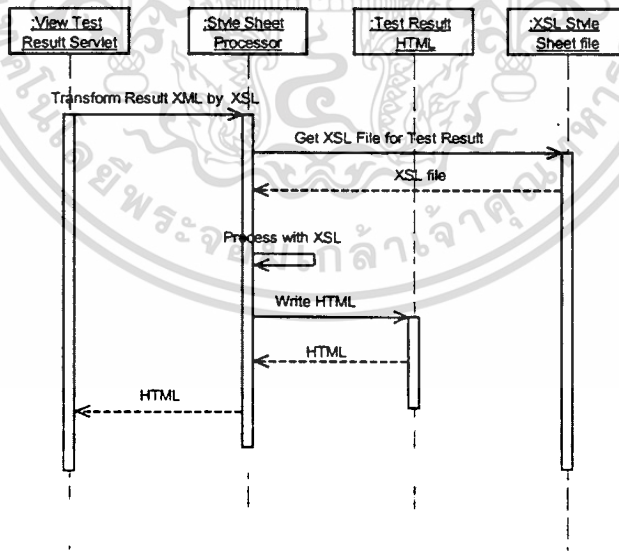
ซีเควินซี่โคอะแกรมของการเชื่อมต่อกับฐานข้อมูลเพื่อสร้างเอกสารเอกซ์เอ็มแอลมีลำดับการทำงานดังนี้ ระบบต้องการดึงข้อมูลการทดสอบในรูปแบบเอกสารเอกซ์เอ็มแอลโดยไปทำการ Query ข้อมูลออกมาจากฐานข้อมูล จากนั้นทำการแปลงให้อยู่ในรูปแบบเอกสารเอกซ์เอ็มแอลโดย Repository Interface และส่งเอกสารเอกซ์เอ็มแอลกลับไปยังส่วนการสร้างเอกสารสคริปต์สำหรับการทดสอบ



รูปที่ 4.24 ซีควีนซ์ไดอะแกรมของการเชื่อมต่อกับฐานข้อมูลเพื่อสร้างเอกสารเอกซ์เอ็มแอล (Interface XML with Repository-Getting)

4.2.13 ซีควีนซ์ไดอะแกรมของยูสเคสการประมวลผล Style Sheet (Style Sheet Process)

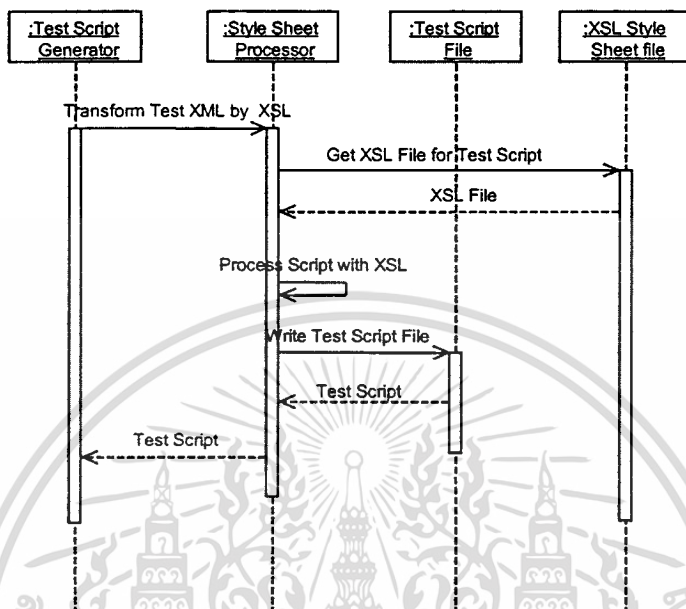
ซีควีนซ์ไดอะแกรมของการประมวลผล Style Sheet เพื่อสร้างเอกสารเอกซ์เอ็มแอลมีลำดับการทำงานดังนี้ ระบบทำการดึงเอกสาร XSL สำหรับแปลงเอกสารให้อยู่ในรูปเอกซ์เอ็มแอลและทำการแปลงเอกสารด้วย XSL จากนั้นจะได้ผลเป็นเอกซ์เอ็มแอลส่งไปกลับไป



รูปที่ 4.25 ซีควีนซ์ไดอะแกรมของการประมวลผล Style Sheet เพื่อสร้างเอกสารเอกซ์เอ็มแอล (Style Sheet Process-HTML)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

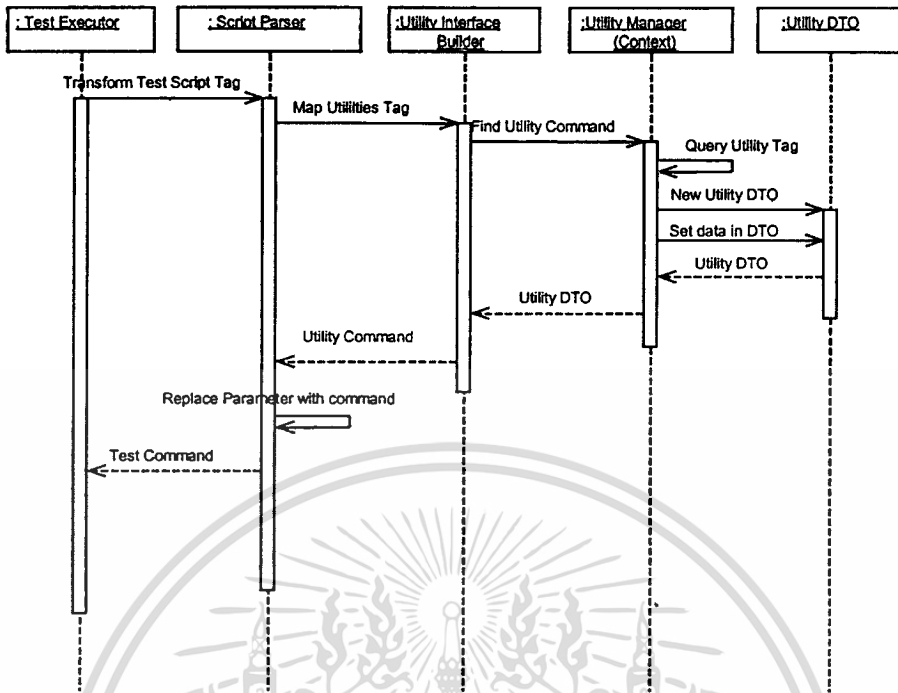
ซีเคิวนซ์ไคอะแกรมของการประมวลผล Style Sheet เพื่อสร้างเอกสารสคริปต์มีลำดับการทำงานดังนี้ ระบบทำการดึงเอกสาร XSL สำหรับแปลงเอกสารให้อยู่สคริปต์ที่ต้องการและทำการแปลงเอกสารด้วย XSL จากนั้นจะได้ผลเป็นเอกสารสคริปต์สำหรับการทดสอบส่งกลับไป



รูปที่ 4.26 ซีเคิวนซ์ไคอะแกรมของการประมวลผล Style Sheet เพื่อสร้างเอกสารสคริปต์ (Style Sheet Process-Test Script)

4.2.14 ซีเคิวนซ์ไคอะแกรมของยูสเคสการแปลงสคริปต์เพื่อใช้เรียกประมวลผล (Script Transformation)

ซีเคิวนซ์ไคอะแกรมของการแปลงสคริปต์เพื่อใช้เรียกประมวลผล มีลำดับการทำงานดังนี้ ส่วนการเรียกโปรแกรมทดสอบต้องการแปลง Test Script Tag โดยจะต้อง Map Utility Tag จากฐานข้อมูล และทำการแทนค่า Parameter ในคำสั่งที่ได้ และส่งคำสั่งสำหรับการทดสอบกลับไป



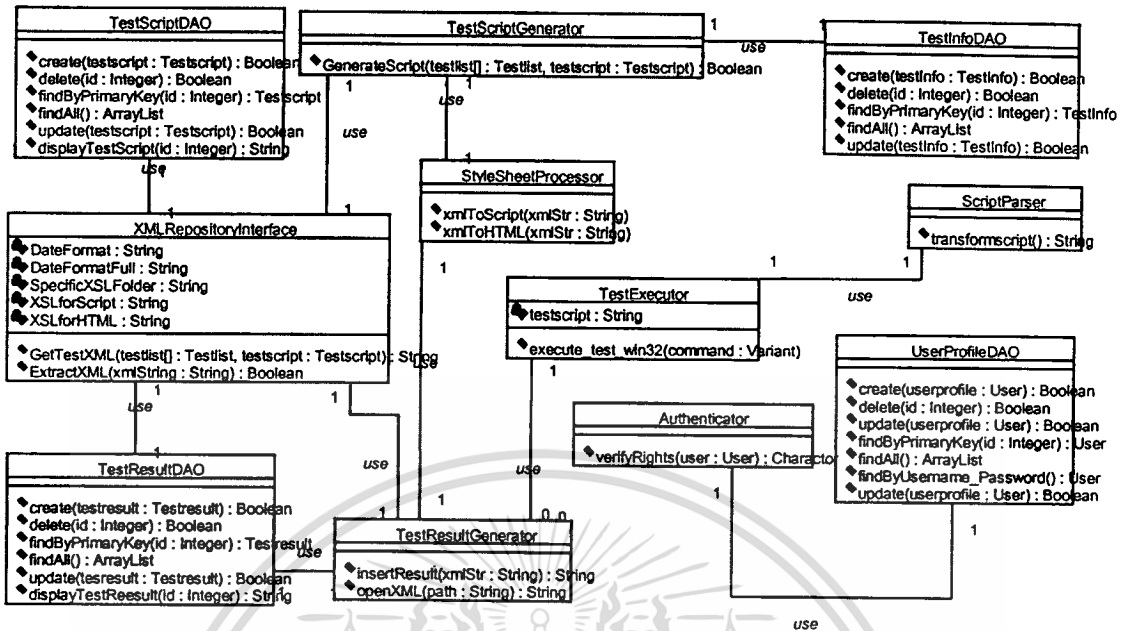
รูปที่ 4.27 ซีควีนซ์ไดอะแกรมของการแปลงสคริปต์เพื่อใช้เรียกประมวลผล (Script Transformation)

4.3 การออกแบบคลาสและแพ็คเกจของระบบ (Designing Classes and Package)

การออกแบบคลาสแบ่งออกเป็น 3 ส่วนใหญ่ เพื่อง่ายต่อการวิเคราะห์คือ คลาสที่เกี่ยวข้องหน้าที่การทำงานหลักของระบบ (Functional Classes), คลาสที่เกี่ยวข้องกับการเชื่อมต่อฐานข้อมูล (Interface Database Classes) และ คลาสที่เชื่อมต่อกับส่วนประสานกับผู้ใช้ (View Classes)

4.3.1 การออกแบบคลาสที่เกี่ยวข้องกับหน้าที่การทำงานของระบบ (Functional Classes)

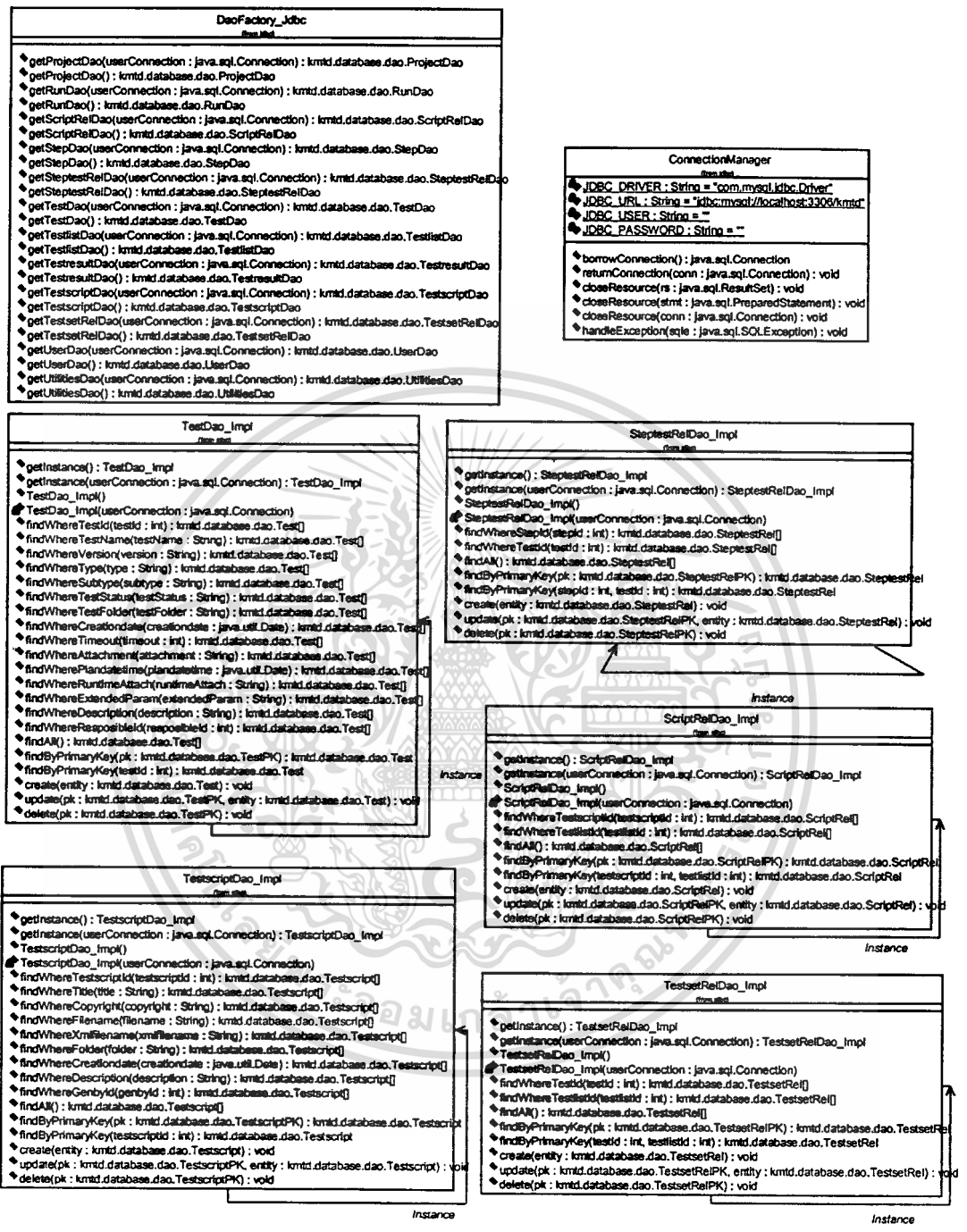
จากการวิเคราะห์ระบบในบทที่ 3 ซึ่งเป็นการวิเคราะห์พฤติกรรมการทำงานของระบบ เมื่อสามารถนำการวิเคราะห์ดังกล่าวมาพิจารณาเพิ่มเติมในรายละเอียดในการพัฒนาในเชิงวัตถุจะได้คลาสที่มีความสัมพันธ์ดังในรูปที่ 4.28



รูปที่ 4.28 คลาสไดอะแกรมที่เกี่ยวกับหน้าที่การทำงานของระบบ

4.3.2 การออกแบบคลาสที่เกี่ยวกับการเชื่อมต่อฐานข้อมูล (Interface Database Classes)

สำหรับการออกแบบระบบคลาสส่วนที่มีการเชื่อมต่อกับฐานข้อมูลกลางซึ่งนำเทคโนโลยี Java Database Connectivity (JDBC) มาใช้ในการเชื่อมต่อ โดยได้ทำการออกแบบให้สอดคล้องกับตารางในฐานข้อมูลของระบบที่กำลังพัฒนาและอ็อบเจกต์ในซีเคิร์ฟไออะแกรมข้างต้น กล่าวคือ อ็อบเจกต์ที่อยู่ในซีเคิร์ฟไออะแกรมต่าง ๆ ที่อยู่ในประเภท Context จะมีความหมายสอดคล้องกับคลาส Data Access Object (DAO) ต่างที่มีความเกี่ยวข้องเช่น สำหรับ Test Information Context จะประกอบไปด้วยคลาสต่าง ๆ ที่ใช้เชื่อมต่อกับตารางต่าง ๆ ในฐานข้อมูลที่จัดเป็นข้อมูลสำหรับการทดสอบ เช่น ตารางแสดงรายละเอียดโครงการที่จะทำการทดสอบ (Project), ตารางแสดงรายละเอียดกลุ่มการทดสอบ (Test List), ตารางแสดงรายละเอียดกรณีการทดสอบ (Test) เป็นต้น ซึ่งคลาสที่ใช้เชื่อมต่อกับฐานข้อมูลสามารถแสดงได้โดยคลาสไดอะแกรมตามรูปที่ 4.29, 4.30 และรูปที่ 4.31 แสดงถึงคลาสที่เป็นจาวาบีเอ็นทีใช้ในการถ่ายโอนข้อมูลระหว่างฐานข้อมูลกลางกับระบบ ซึ่งเรียกว่า Data Transfer Object (DTO)



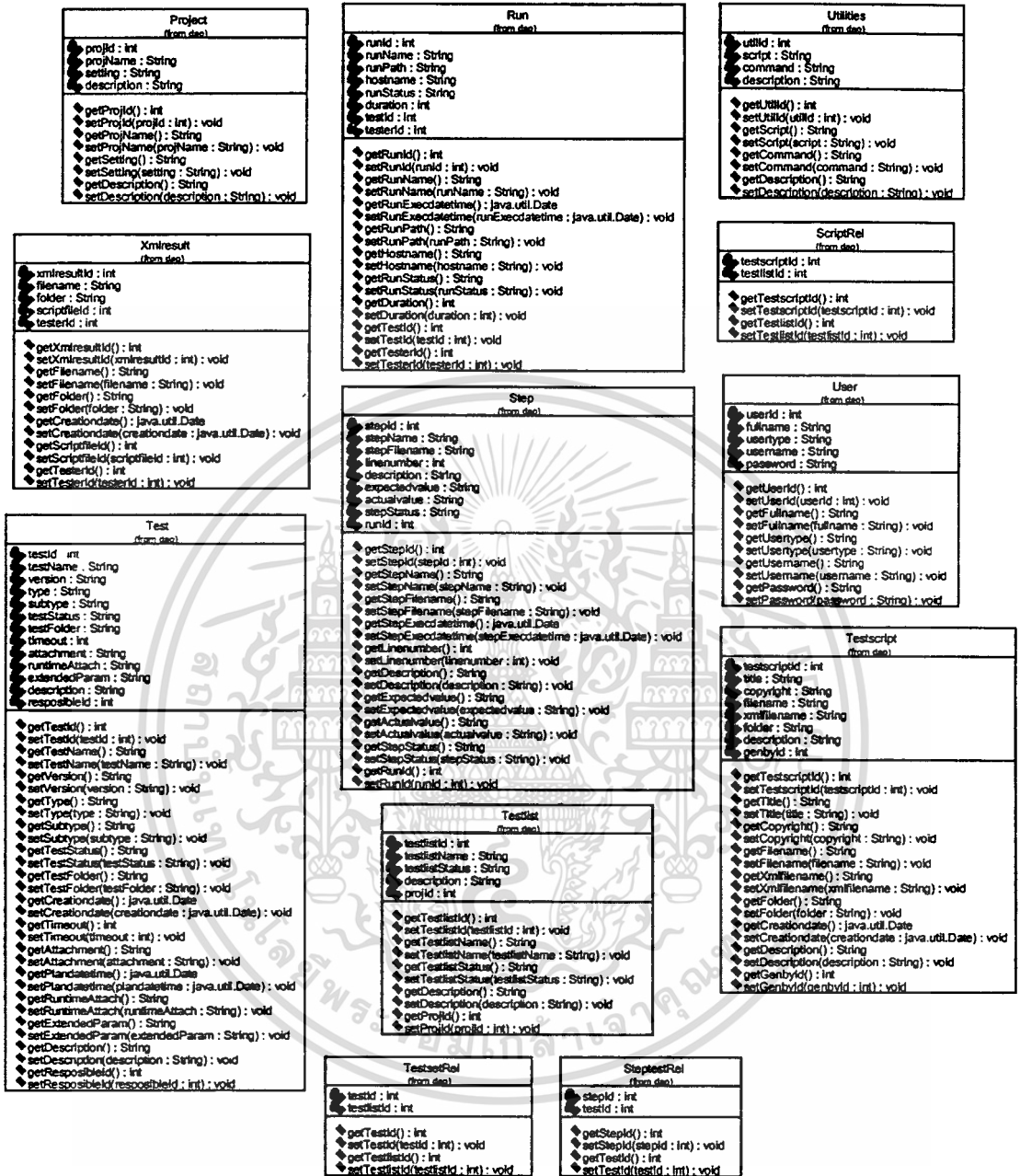
รูปที่ 4.29 คลาสไดอะแกรมที่เกี่ยวกับการเชื่อมต่อฐานข้อมูล (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.30 คลาสไลอะแกรมที่เกี่ยวกับการเชื่อมต่อฐานข้อมูล (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

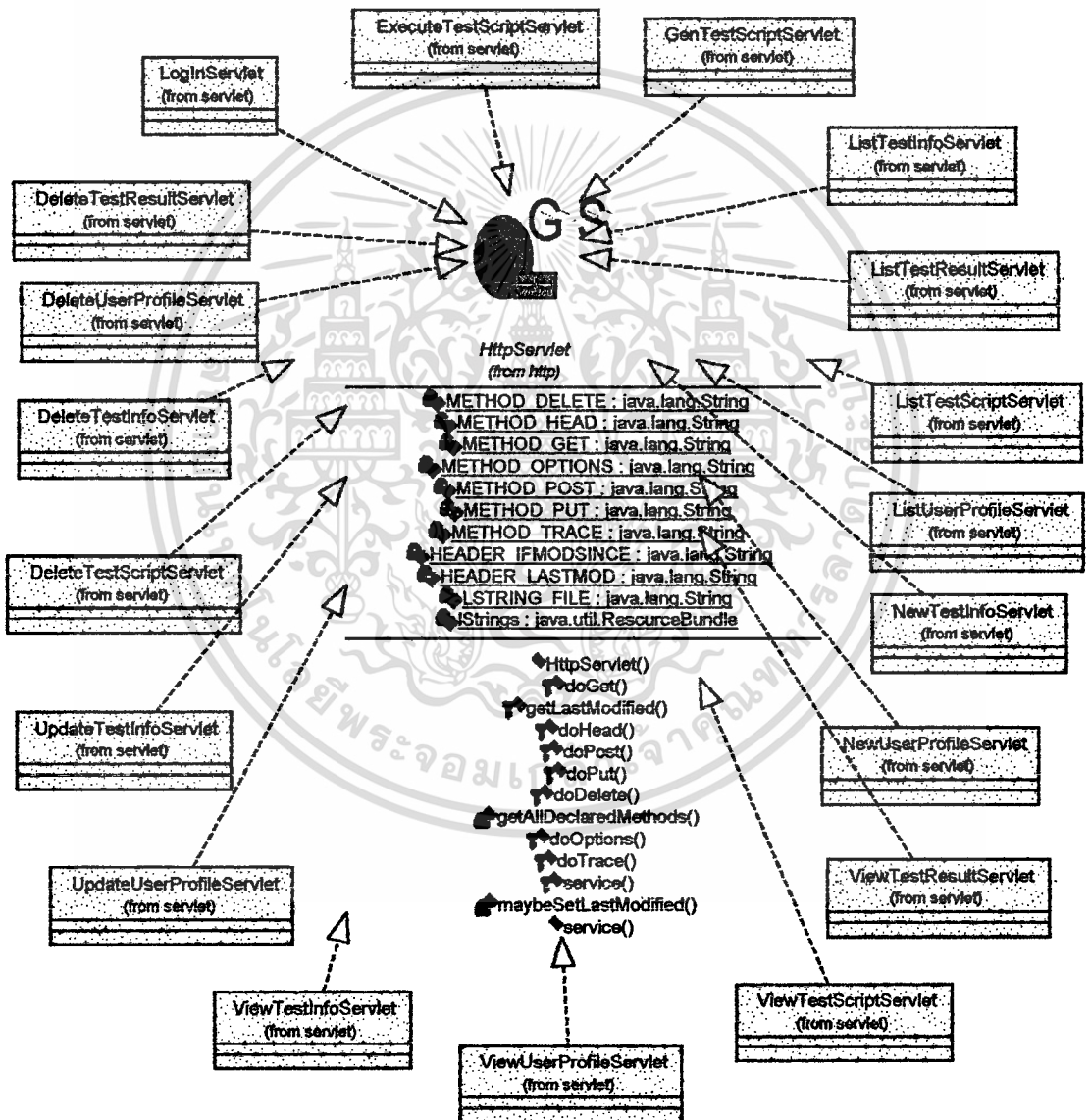


รูปที่ 4.31 คลาสโคออดิเนตแสดงจาวาบี๋นของระบบ

เอกสารนี้เป็นเอกสารที่สวอนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 คลาสที่เชื่อมต่อกับส่วนประสานกับผู้ใช้ (View Classes)

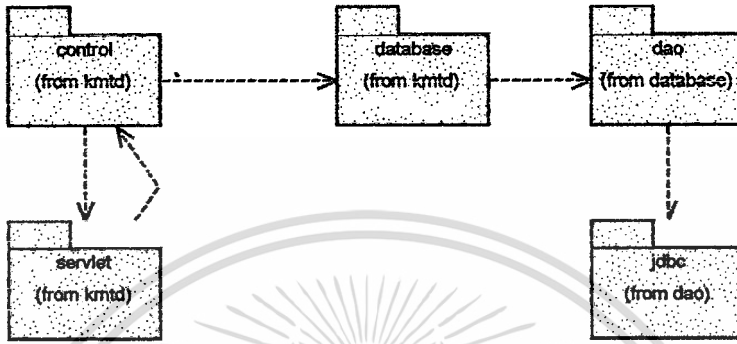
ในการออกแบบคลาสในฝั่งแม่ข่าย (Server) ที่ทำงานร่วมกับส่วนต่อประสาน (User Interface) จะใช้ออกแบบเป็นจาวาเซิร์ฟเลต (Java Servlet) เพื่อเป็นตัวกลางสำหรับควบคุมการรับข้อมูลจากหน้าเว็บและส่งข้อมูลไปประมวลผลต่อซึ่งเมื่อได้ผลลัพธ์ของการประมวลผลแล้วจะนำส่งต่อไปแสดงผลยังหน้าเว็บที่เกี่ยวข้อง คลาสในส่วนนี้จะต้องถ่ายทอดคุณสมบัติจากคลาส HttpServlet ซึ่งเป็นคลาสที่จาวามีให้ รูปที่ 4.32 แสดงคลาสเซิร์ฟเลต



รูปที่ 4.32 คลาสโคแอมกับเซิร์ฟเลต

4.3.4 การสร้างแพ็คเกจจากการวิเคราะห์รายละเอียด (Designing Package)

คลาสต่าง ๆ ข้างต้นที่ได้จากการออกแบบระบบสามารถจัดกลุ่มเป็นแพ็คเกจได้ตามหน้าที่การทำงานของแต่ละคลาสได้ตามแพ็คเกจดังรูปที่ 4.33



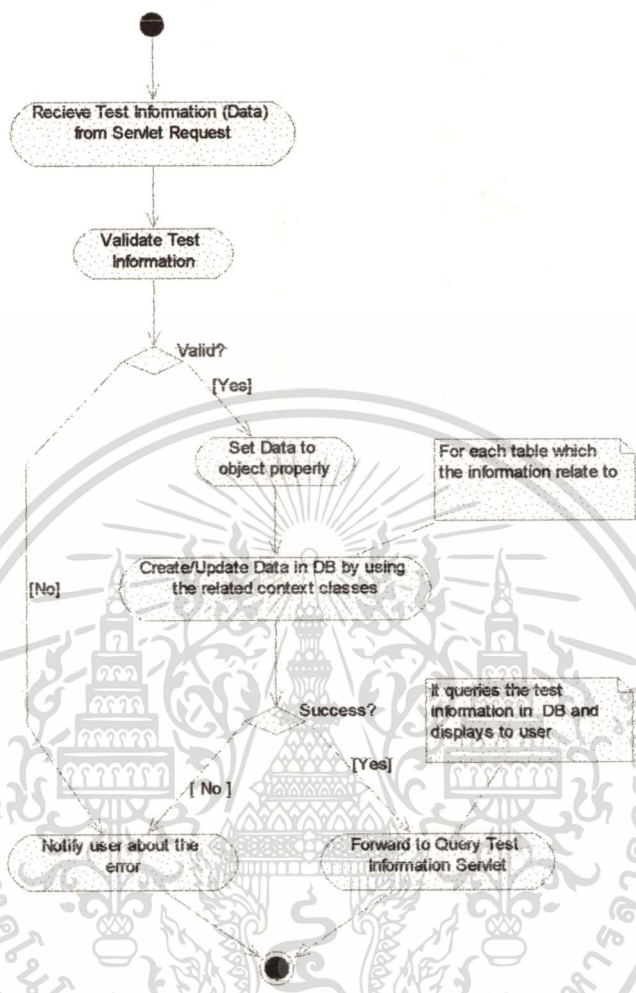
รูปที่ 4.33 แผนภาพแสดงแพ็คเกจของระบบ

4.4 การวิเคราะห์กิจกรรมของระบบ (Activity Analysis)

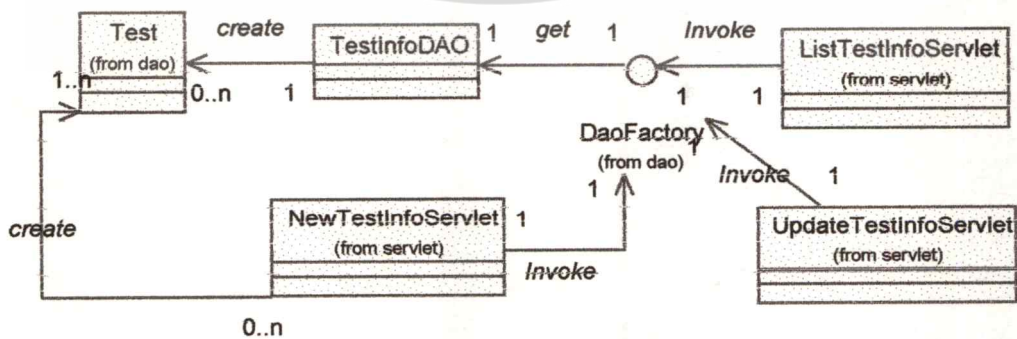
การวิเคราะห์กิจกรรมที่ของระบบในที่นี้จะกล่าวถึงกิจกรรมที่สำคัญและมีความซับซ้อนในการทำงาน โดยจะอธิบายด้วยเอกทวิดีโคอะแกรมดังต่อไปนี้

4.4.1 กิจกรรมการสร้างและแก้ไขข้อมูลสำหรับการทดสอบ (Test Information Creation)

การสร้างและแก้ไขข้อมูลจะเริ่มจากที่ผู้ใช้ Submit ข้อมูลที่ต้องการบันทึกในฐานะข้อมูลส่งผ่านมาทาง Servlet และระบบจะทำการตรวจสอบความถูกต้องของข้อมูลก่อนทำการบันทึก จากนั้นจะส่งการทำงานไปยังหน่วยที่ทำการค้นหาข้อมูลเพื่อแสดงข้อมูลในระบบที่ได้บันทึกแล้วหากการบันทึกถูกต้อง แต่ถ้าผิดพลาดใด ๆ จะทำการแจ้งแก่ผู้ใช้ทราบ



รูปที่ 4.34 แอคทิวิตี้ โค้ดแกรมแสดงการสร้างและแก้ไขข้อมูลการทดสอบ

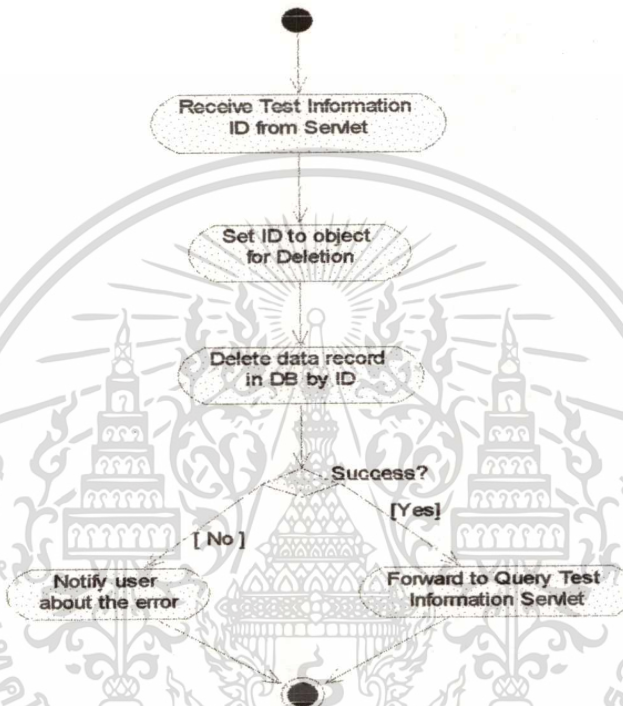


รูปที่ 4.35 คลาสโค้ดแกรมแสดงคลาสที่เกี่ยวข้องกับการสร้างและแก้ไขข้อมูลการทดสอบ

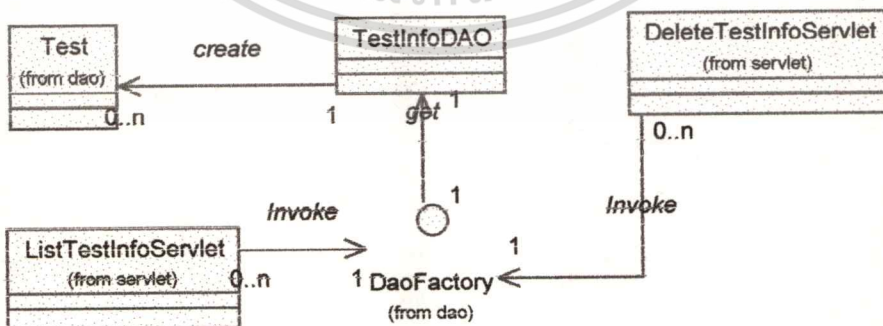
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 กิจกรรมการลบข้อมูลสำหรับการทดสอบ (Test Information Deletion)

การลบข้อมูลจะเริ่มจากที่ผู้ใช้ Submit เลขอ้างอิงข้อมูลที่ต้องการลบจากฐานข้อมูลส่งผ่านมาทาง Servlet จากนั้นจะส่งการทำงานไปยังหน่วยที่ทำการค้นหาข้อมูลเพื่อแสดงข้อมูลในระนาบที่ได้บันทึกแล้วหากการลบถูกต้อง แต่ถ้าผิดพลาดใด ๆ จะทำการแจ้งแก่ผู้ใช้ทราบ



รูปที่ 4.36 แอคทีวิตี้ไดอะแกรมแสดงการลบข้อมูลการทดสอบ

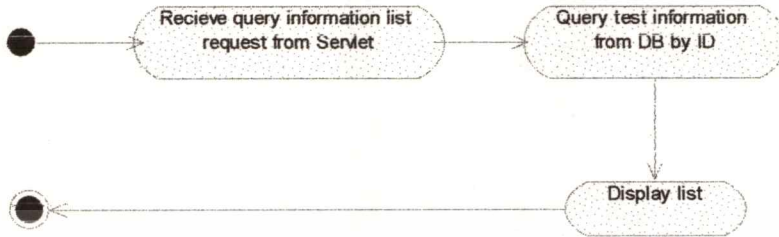


รูปที่ 4.37 คลาสไดอะแกรมแสดงคลาสที่เกี่ยวกับการลบข้อมูลการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.3 กิจกรรมการค้นหาข้อมูลสำหรับการทดสอบ (Test Information Query)

1) การค้นรายการข้อมูลสำหรับการทดสอบ (Test Information List)



รูปที่ 4.38 แอคทิวิตี้ไดอะแกรมแสดงการค้นรายการข้อมูลการทดสอบ

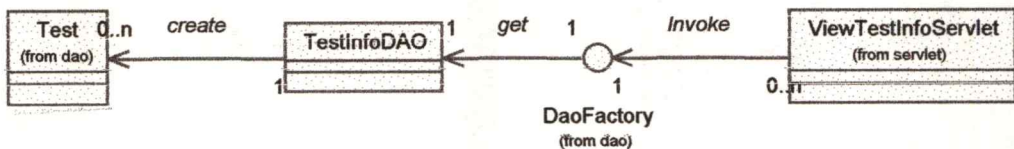


รูปที่ 4.39 คลาสไดอะแกรมแสดงคลาสที่เกี่ยวข้องกับการค้นรายการข้อมูลการทดสอบ

2) การค้นรายละเอียดของข้อมูลสำหรับการทดสอบ (Test Information Detail)



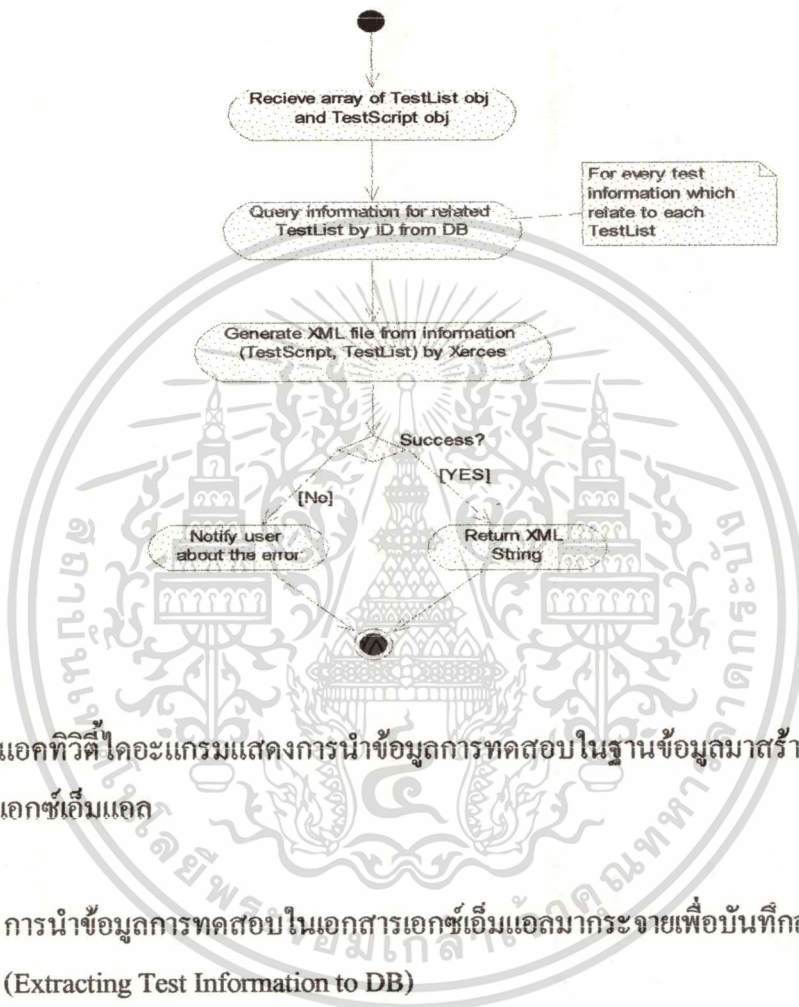
รูปที่ 4.40 แอคทิวิตี้ไดอะแกรมแสดงการค้นรายละเอียดข้อมูลการทดสอบ



รูปที่ 4.41 คลาสไดอะแกรมแสดงคลาสที่เกี่ยวข้องกับการค้นรายละเอียดข้อมูลการทดสอบ

4.4.4 กิจกรรมการเชื่อมต่อระหว่างเอกสารเอกซ์เอ็มแอลและฐานข้อมูล (XML Interface)

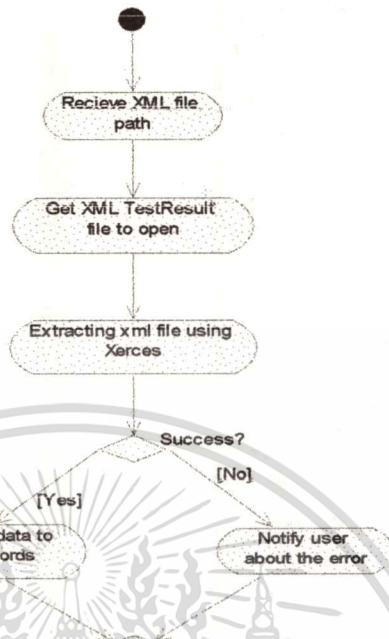
- 1) การนำข้อมูลการทดสอบในฐานข้อมูลมาสร้างเป็นเอกสารเอกซ์เอ็มแอล (Get Test Information to XML)



รูปที่ 4.42 แอคทีวิตีโคอะแกรมแสดงการนำข้อมูลการทดสอบในฐานข้อมูลมาสร้างเป็นเอกสารเอกซ์เอ็มแอล

- 2) การนำข้อมูลการทดสอบในเอกสารเอกซ์เอ็มแอลมากระจายเพื่อบันทึกลงฐานข้อมูล (Extracting Test Information to DB)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.43 แอคทิวิตีไดอะแกรมแสดงการนำข้อมูลการทดสอบในเอกสารเอกซ์เอ็มแอลมากระจาย

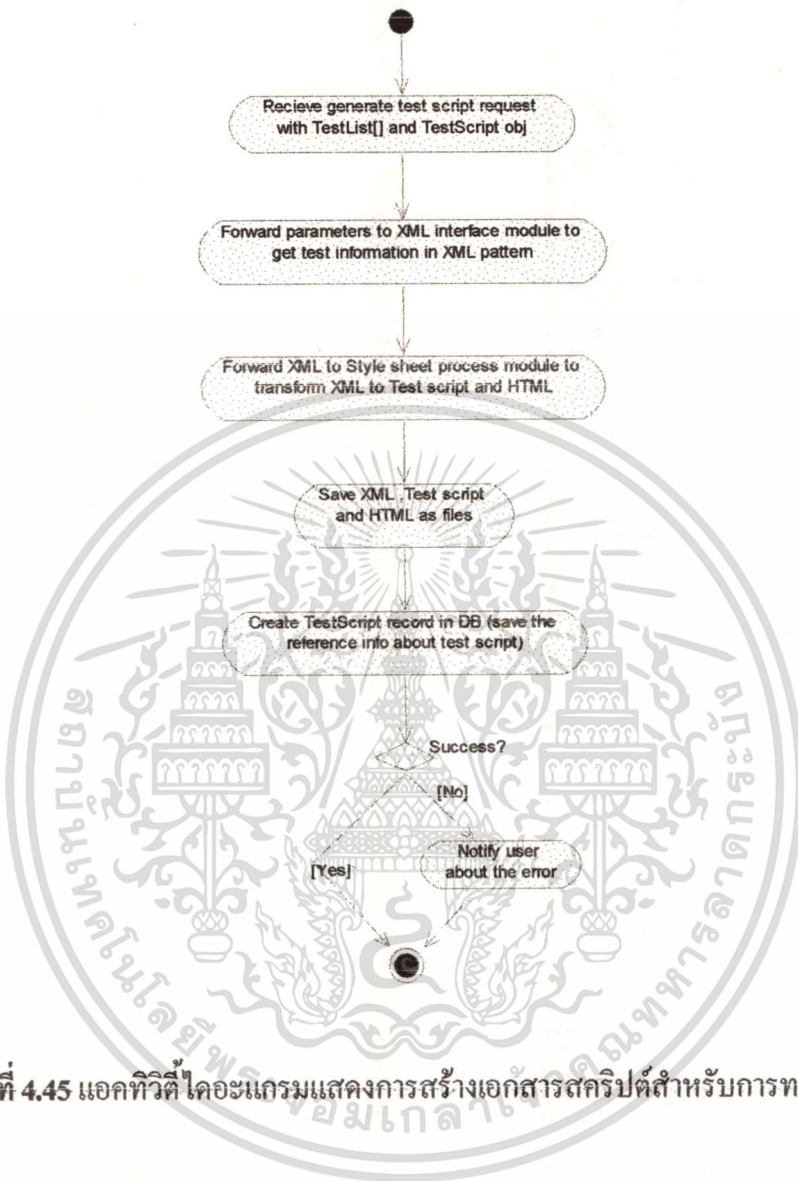


รูปที่ 4.44 คลาสไดอะแกรมแสดงคลาสที่เกี่ยวกับการเชื่อมต่อเอกสารเอกซ์เอ็มแอลและฐานข้อมูล

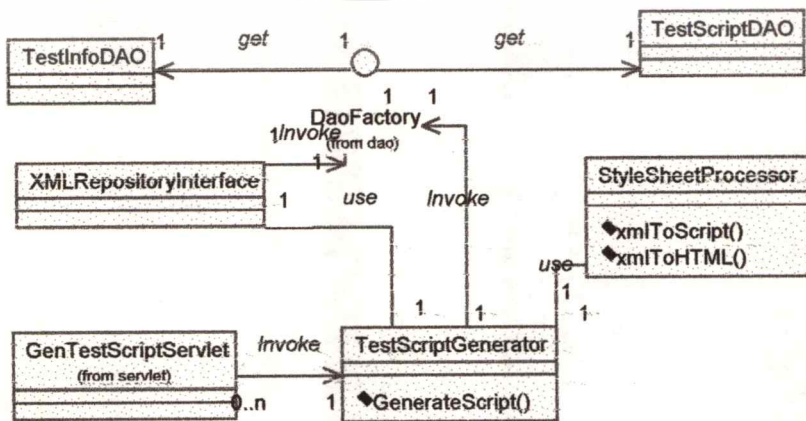
4.4.5 กิจกรรมการสร้างเอกสารสคริปต์สำหรับการทดสอบ (Test Script Generating)

การสร้างเอกสารสคริปต์จะมีค่าที่อ้างอิงที่ต้องการคือ ข้อมูลทั่วไปของเอกสารสคริปต์ และข้อมูลสำหรับการทดสอบ ซึ่งผู้ใช้จะทำการเลือกส่งมาให้เพื่อสร้างเอกสารสคริปต์ โดยมีลำดับการสร้างดังนี้ คือ ทำการนำข้อมูลการทดสอบในฐานข้อมูลมาสร้างเป็นเอกสารเอกซ์เอ็มแอล และนำเอกสารเอกซ์เอ็มแอลไปแปลงให้อยู่ในรูปแบบเอกสารสคริปต์และเอชทีเอ็มแอลด้วย ส่วนการแปลงเอกสารด้วย Style Sheet จากนั้นจะบันทึกทั้งเอกสารเอกซ์เอ็มแอล, เอกสารเอชทีเอ็มแอล และเอกสารสคริปต์ที่ได้ใน path ที่เหมาะสม และบันทึกข้อมูลอ้างอิงต่าง ๆ ลงในฐานข้อมูล หากทำไม่สำเร็จจะแจ้งให้ผู้ใช้ทราบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.45 แอคทิวิตี้ไดอะแกรมแสดงการสร้างเอกสารสคริปต์สำหรับการทดสอบ



รูปที่ 4.46 คลาสไดอะแกรมแสดงคลาสที่เกี่ยวกับการสร้างเอกสารสคริปต์สำหรับการทดสอบ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.6 กิจกรรมการแปลงเอกสารเอกซ์เอ็มแอลด้วย Style Sheet (Style Sheet Process)

การแปลงเอกสารเอกซ์เอ็มแอลให้อยู่ในรูปแบบเอกสารที่ต้องการ โดยใช้ XML Style Sheet (XSL) ซึ่งในที่นี้สามารถแปลงเป็น Test Script หรือ HTML ของ Test Script เนื่องจากมีการสร้าง XSL ไว้สองรูปแบบข้างต้นลำดับการทำงานเป็นดังรูป

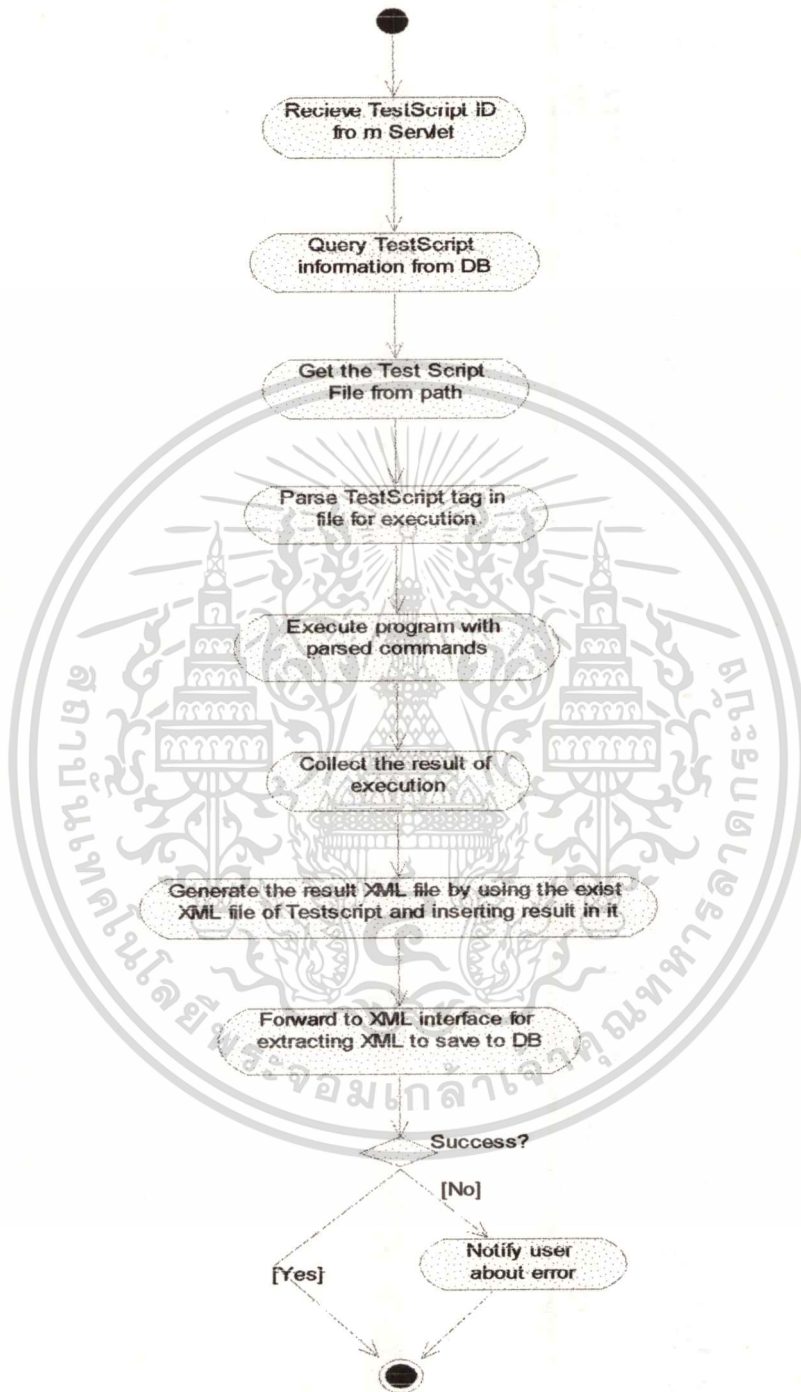


รูปที่ 4.47 แอคทีวิตีไดอะแกรมแสดงการแปลงเอกสารด้วย Style Sheet

4.4.7 กิจกรรมการเรียกทดสอบซอฟต์แวร์ตามเอกสารสคริปต์ (Test Execution)

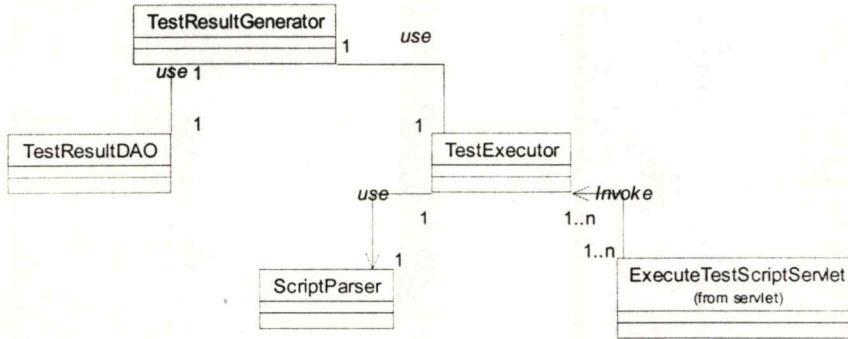
การเรียกทดสอบตามเอกสารสคริปต์จะเริ่มจากเปิดเอกสารสคริปต์ขึ้นมาและ Parse คำสั่งของแต่ละ Tag เพื่อให้ได้คำสั่ง จากนั้นนำคำสั่งมาประมวลผลและเก็บผลการทำงาน ซึ่งจะสร้างผลการทดสอบเป็นเอกสารเอกซ์เอ็มแอล โดยการเปิดเอกสารเอกซ์เอ็มแอลของการทดสอบขึ้นมา แทรกผลการทดสอบลงไป สุดท้ายจะเรียกส่วนการกระทำข้อมูลจากเอกสารเอกซ์เอ็มแอลเพื่อเก็บผลการทดสอบลงฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.48 แอคทีวิตีไดอะแกรมแสดงการเรียกทดสอบโปรแกรมตามเอกสารสคริปต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.49 คลาสไดอะแกรมแสดงคลาสที่เกี่ยวกับการเรียกทดสอบโปรแกรมตามเอกสารสคริปต์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การพัฒนาระบบงาน

5.1 เครื่องมือที่ใช้ในการพัฒนาระบบ

5.1.1 J2SDK1.4.1_06 (สัญญาบัตร มาศสุรางค์. 2545)

เนื่องด้วยผู้พัฒนาระบบเลือกใช้ภาษา Java ในการพัฒนาซึ่งมีหลักการใช้งานเป็นแบบเชิงวัตถุ และหนึ่งในชุดโปรแกรมที่ใช้พัฒนาโปรแกรมภาษา Java ออกมาคือ J2SDK โดยในที่นี้ผู้พัฒนาเลือกใช้รุ่น 1.4.1_06 ของบริษัท Sun สำหรับ J2SDK จะมี Java Compiler และ Java interpreter ที่จำเป็นต้องใช้สำหรับการสร้างโปรแกรมภาษา Java แต่ที่สำคัญคือมี Java Class พื้นฐานรวมอยู่ด้วย ดังนั้นเพื่อความสะดวกในการพัฒนาผู้พัฒนาจึงเลือกใช้ J2SDK ร่วมกับชุดพัฒนาโปรแกรมภาษา Java อื่นของบริษัท Borland ด้วย นั่นคือ JBuilder

สำหรับการติดตั้ง J2SDK รุ่น 1.4.1_06 บนระบบปฏิบัติการ Win32 ทำได้ดังต่อไปนี้

- ดาวน์โหลดไฟล์สำหรับการติดตั้งซึ่งเป็นไฟล์ที่มีชื่อในอย่างเช่น j2sdk-1_4_1_06-windows-i586.exe ได้จากเว็บไซต์ <http://java.sun.com>
- ติดตั้งลงในไดเรกทอรีที่ต้องการซึ่งโดยปกติจะได้ชื่อ j2sdk1.4.1_06 ใน drive C: ซึ่งเมื่อติดตั้งแล้วจะพบโปรแกรมที่เราใช้งานบ่อย ๆ ในไดเรกทอรี C:\j2sdk1.4.1_06\bin เช่น javac.exe (Java Compiler), java.exe (Java interpreter) และ appletviewer.exe (โปรแกรมสำหรับดู applet) เป็นต้น
- ควรตั้งค่าไดเรกทอรี path (Set Path) ไปที่ไดเรกทอรี C:\j2sdk1.4.1_06\bin เพื่อให้สามารถเรียกใช้โปรแกรมในไดเรกทอรีนี้ได้จากที่อื่น ๆ ได้ และตั้งค่าคลาส path (Set Classpath) ไปที่ไดเรกทอรี C:\j2sdk1.4.1_06\lib\Classes.zip ด้วยเพื่อบอกให้การค้นหาคลาสจากที่นี้ด้วย

5.1.2 Apache Tomcat 5.0.16 (Apache . 2004 a)

เป็น Web Server บนเครื่องแม่ข่ายที่สนับสนุนการประมวลผลโปรแกรมภาษา Java ซึ่งสามารถดาวน์โหลดชุดติดตั้งสำหรับระบบปฏิบัติการ Windows ได้ที่เว็บไซต์ <http://jakarta.apache.org/tomcat-5> โดยไม่เสียค่าใช้จ่ายใด ๆ เนื่องจากเป็น Open Source และสำหรับการติดตั้งสามารถเรียกชุดติดตั้งทำงานได้ทันที

5.1.3 Apache Jakarta Struts 1.1 Release Builds (Apache . 2004 b)

Struts Framework ของ Jakarta Apache เป็นแนวคิดการทำงานที่รวม Java Servlet, Java ServerPages, tags และ message resources เข้าด้วยกันสำหรับการพัฒนา Java web application ซึ่งผู้พัฒนาระบบใช้ในส่วนการทำ Java web application เพื่อให้สามารถทำการบำรุงรักษาได้ง่าย และสามารถนำไปประยุกต์ใช้สำหรับการพัฒนาระบบแบบที่มียกด้วย ดังนั้นเพื่อการพัฒนาระบบให้เป็นไปตามแนวคิดนี้จะใช้พื้นฐานแนวคิดของการพัฒนาระบบแบบ Java Servlet แบบเดิมแต่จะมีการออกแบบระบบให้เหมาะสมกับการพัฒนาระบบหลาย ๆ ส่วนงานพร้อมกัน นั่นคือจะแบ่งส่วนการทำงานออกเป็น 3 ระดับ ที่เรียกว่า Model-View-Controller (MVC) ซึ่ง Model จะเป็นส่วนกระบวนการที่จัดการเกี่ยวกับข้อมูล, View จะเป็นส่วนที่ใช้ในการแสดงผลข้อมูลแก่ผู้ใช้เช่น JSP หรือ HTML และ Controller จะเป็นส่วนที่เป็นตัวควบคุมและเชื่อมต่อที่ทำให้งานทั้งหมดไหลไปได้ ดังรูปที่ 5.1 และมีองค์ประกอบที่สำคัญในการทำงานได้แก่

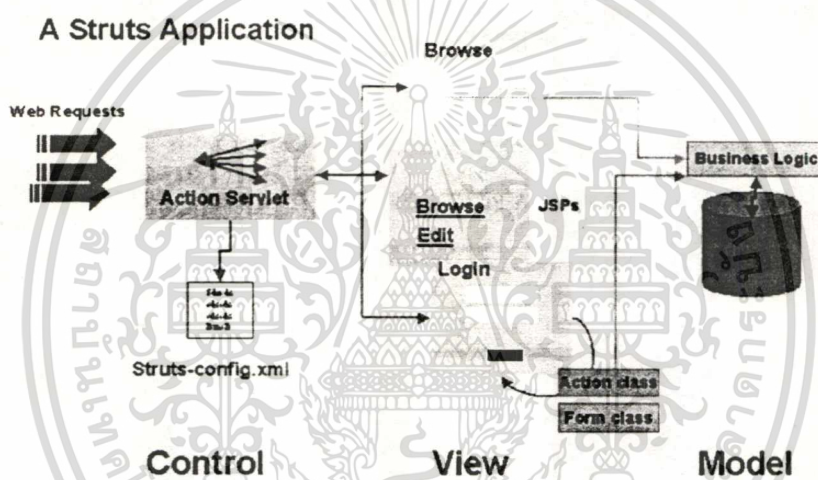
- The Controller เป็นส่วนที่รับ request จากผู้ใช้ระบบและมีหน้าที่ทำการจับคู่ระหว่างการกระทำที่ผู้ใช้องการกับคลาสที่ตอบสนองการกระทำนั้น (มีมากับ Framework)
- The struts-config.xml File เป็นเอกสารเอกซ์เอ็มแอลที่เก็บรูปแบบการจับคู่ที่กล่าวข้างต้นและข้อมูลต่าง ๆ ที่เกี่ยวข้องในการทำงาน ซึ่งจะถูกนำไปวางไว้ที่ WEB-INF ของ web application บนเครื่องแม่ข่าย (server)
- Action Classes เป็นคลาสที่ผู้พัฒนาระบบต้องสร้างขึ้นเอง โดยมีหน้าที่เป็นตัวเชื่อมระหว่างการร้องขอของผู้ใช้และการบริการของระบบ นั่นคือคลาสนี้จะรับ request และประมวลผลเรียกบริการเหมาะสมของระบบทำงานและส่ง Action Forward Object ที่บอกได้ว่าจะมีการเรียกส่วนการทำงานไหนต่อไปยังผู้เรียกใช้ คลาสนี้ซึ่งจะเป็นระดับการทำงานส่วน Controller
- View Resources เป็นส่วนทรัพยากรต่างสำหรับระดับ View ที่ประกอบไปด้วย Java Server Pages, HTML pages, JavaScript และ Stylesheet files, Resource bundles, JavaBeans และ Struts JSP tags
- ActionForms เป็นส่วนที่ทำการตรวจสอบข้อมูลโดยการเก็บข้อมูลที่ผู้ใช้ส่งทาง request ที่สามารถทำได้ง่าย ๆ ซึ่งเป็นส่วนที่ทำงานคล้าย firewall ระหว่างการ forms บนเว็บเพจกับ Action คลาส นั่นคือมันจะทำการตรวจสอบค่าข้อมูลนำเข้า (input) ก่อนที่จะนำไปใช้ประมวลผลใน Action คลาส ซึ่งถ้าข้อมูลไม่ถูกต้องจะสามารถบอกถึงความผิดพลาดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Model Components เป็นส่วนโปรแกรมย่อยต่าง ๆ ที่จัดการเกี่ยวกับข้อมูลซึ่งจะไม่ได้มีให้ใน Framework แต่สามารถนำมาใช้ร่วมกันได้เลย เช่น JavaBeans, EJB, CORBA และ JDO เป็นต้น

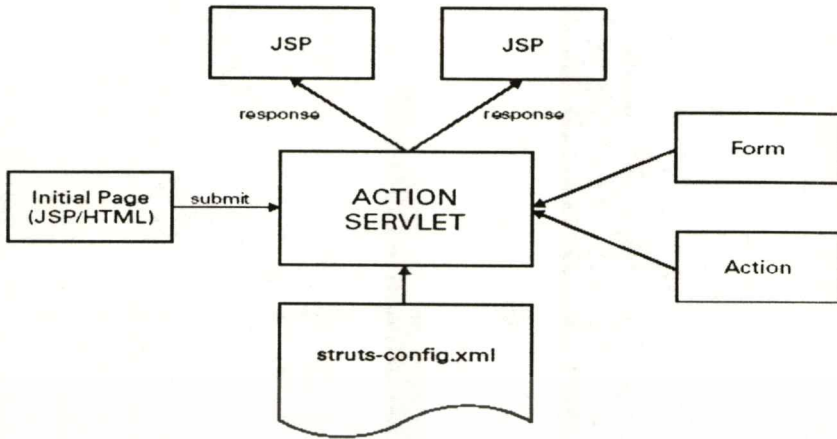
สำหรับลำดับการทำงานของแต่ละองค์ประกอบจะเป็นไปดังรูปที่ 5.3 และการติดตั้ง Framework สามารถทำได้โดยดาวน์โหลดไฟล์สำหรับติดตั้งบนเครื่องแม่ข่ายจากเว็บไซต์

<http://jakarta.apache.org/struts> และทำการกระจายไฟล์เพื่อนำ war ไฟล์ที่อยู่ในชุดติดตั้งไปวางไว้ที่ web container แล้วจึงเปิด Web Server อีกครั้งซึ่งถ้าเป็น Web Server รุ่นใหม่ก็จะทำการแตก war ไฟล์เป็น web application ที่สามารถศึกษาตัวอย่างและนำมาใช้ได้เลย



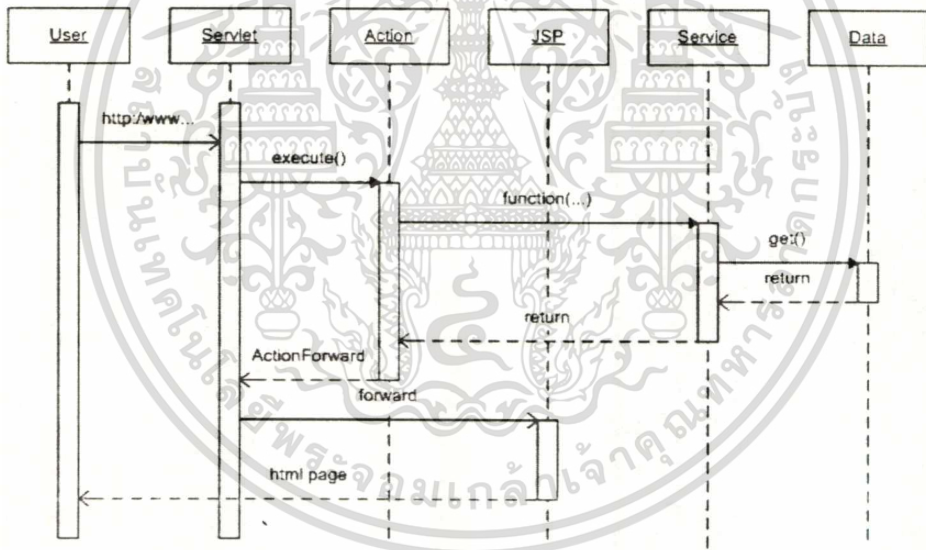
รูปที่ 5.1 ส่วนการทำงานในรูปแบบการทำงานแบบ Struts

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 สถาปัตยกรรมการทำงานแบบ Struts

Sequence diagram



รูปที่ 5.3 ซีควีนซ์ไดอะแกรมแสดงลำดับการทำงานของระบบที่ใช้แนวคิด Struts

5.1.4 mySql 4.0.17(สงกรานต์ ทองสว่าง. 2545)

เป็นระบบฐานข้อมูลบนเครื่องแม่ข่ายที่เป็น Open Source ซึ่งสามารถดาวน์โหลดติดตั้งได้จากเว็บไซต์ <http://www.mysql.com> โดยไม่มีค่าใช้จ่ายใดๆ ซึ่งในที่นี้ผู้พัฒนาระบบเลือกใช้รุ่น 4.0.17 สำหรับการติดตั้งบนระบบปฏิบัติการ Window โดยการติดตั้งสามารถนำชุดติดตั้งมากระจาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์และเรียกตัวติดตั้งโปรแกรมที่อยู่ภายในนั้นทำงาน หลังจากนั้นจะมีการติดตั้งที่ Drive C ในไดเรกทอรี mysql โดยปริยาย นอกจากนี้จะต้องดาวน์โหลด Library ไฟล์ของจาวาสำหรับการติดต่อกับฐานข้อมูลซึ่งมีชื่อไฟล์ลักษณะนี้ mysql-connector-java-3.1.0-alpha-bin.jar เพื่อใช้สำหรับสร้าง Connection

5.1.5 JBuilder X และ Eclipse 2.1

JBuilder เป็นชุดโปรแกรมสำหรับสร้างโปรแกรมภาษา Java ซึ่งเป็นของบริษัท Borland ในที่นี้ผู้พัฒนาระบบเลือกใช้รุ่น 10 ซึ่งมีส่วนการทำงานที่ช่วยเกี่ยวกับการพัฒนาระบบงานประเภท web application และ Struts Framework จึงทำให้สะดวกสำหรับผู้พัฒนาระบบ

Eclipse 2.1 เป็นชุดโปรแกรมสำหรับสร้างโปรแกรมภาษา Java เช่นเดียวกันแต่สามารถดาวน์โหลดชุดติดตั้งได้ที่เว็บไซต์ <http://www.eclipse.org/downloads/index.php> โดยไม่ต้องเสียค่าใช้จ่ายใด ๆ และมีการใช้ทรัพยากรพื้นที่หน่วยความจำในฮาร์ดดิสก์น้อยกว่า JBuilder แต่ไม่มีส่วนการทำงานช่วยเกี่ยวกับการพัฒนาระบบ web application และ Struts Framework อยู่ในชุดโปรแกรมต้องดาวน์โหลดส่วนนี้มาเชื่อมต่อกับชุดโปรแกรมเองภายหลัง สำหรับการติดตั้งสามารถกระจายไฟล์จากชุดติดตั้งไปยังไดเรกทอรีที่ต้องการและสามารถเรียกใช้งานได้จากที่นั่นได้เลย นอกจากนี้คู่มือการใช้งานสามารถค้นหาได้ภายในเว็บไซต์ต่าง ๆ ทั่วไป

5.1.6 Java Library

- Xerces-J สำหรับส่วนการเชื่อมต่อข้อมูลเข้าและออกระหว่างเอกสารเอกซ์เอ็มแอลและฐานข้อมูล
- Java Mail สำหรับการส่งอีเมล
- Javax.xml.transform ซึ่งอยู่ในกลุ่มคลาสพื้นฐานใน J2SDK อยู่แล้ว โดยเป็นส่วนการแปลงเอกสารเอกซ์เอ็มแอลด้วย Style Sheet ให้เป็นเอกสารในรูปแบบอื่น ๆ เช่น เอกสารสคริปต์ หรือ HTML เป็นต้น

5.2 การทำงานของระบบ (หน้าจอ)

5.2.1 หน้าจอการทำงานหลัก

เป็นหน้าจอแรกของการเข้าใช้งานระบบซึ่งสามารถใส่ Username และ Password เพื่อเข้าใช้ระบบตามสิทธิ์ของแต่ละบุคคลได้ที่หน้าจอนี้ ซึ่งหลังจากสามารถล็อกอินแล้วจะปรากฏเมนูการทำงานให้เลือกใช้ได้ดังนี้

การจัดการเกี่ยวกับข้อมูลการทดสอบ (Test Management)

การจัดการเกี่ยวกับเอกสารสคริปต์ (Test Script)

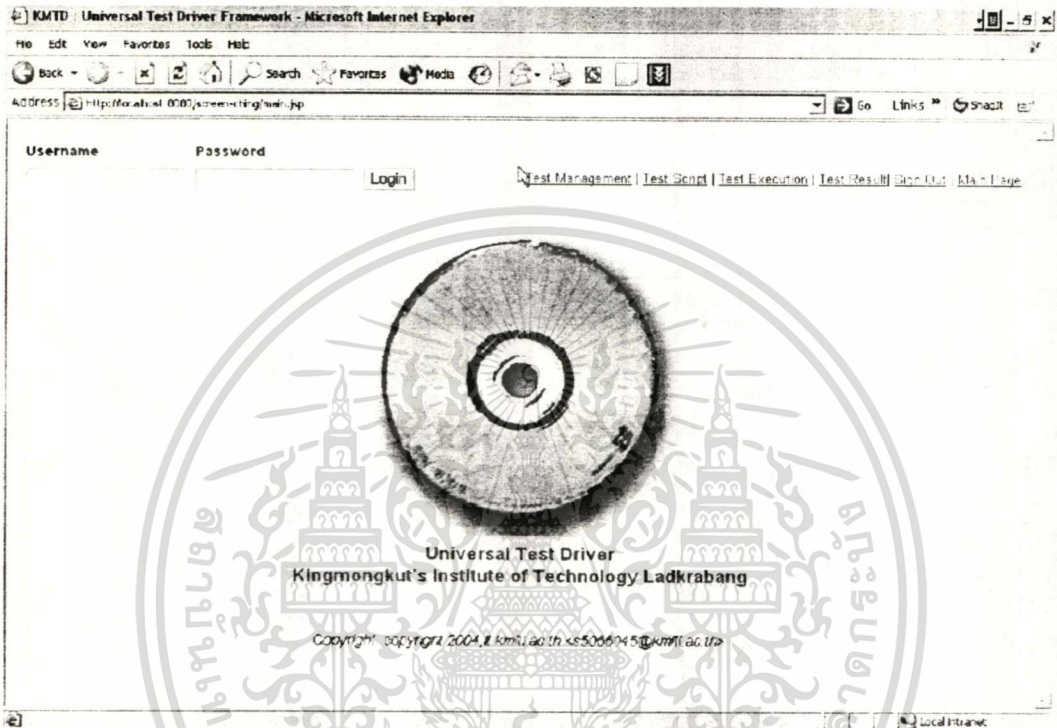
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเรียกทำการทดสอบโปรแกรม (Test Execution)

การจัดการเกี่ยวกับผลการทดสอบ (Test Result)

การออกจากระบบ (Sign Out)

การกลับสู่หน้าจอการทำงานหลัก (Main Page)



รูปที่ 5.4 หน้าจอการทำงานหลักของระบบ

5.2.2 หน้าจอการจัดการเกี่ยวกับข้อมูลการทดสอบ

เนื่องจากข้อมูลสำหรับการทดสอบมีรายละเอียดอยู่หลายส่วนซึ่งสามารถเข้าจัดการข้อมูลแต่ละส่วนได้จากเมนูย่อย ได้แก่

ข้อมูลโครงการพัฒนาซอฟต์แวร์ (Project)

ข้อมูลกลุ่มกรณีการทดสอบ (Test List)

ข้อมูลกรณีการทดสอบ (Test Case)

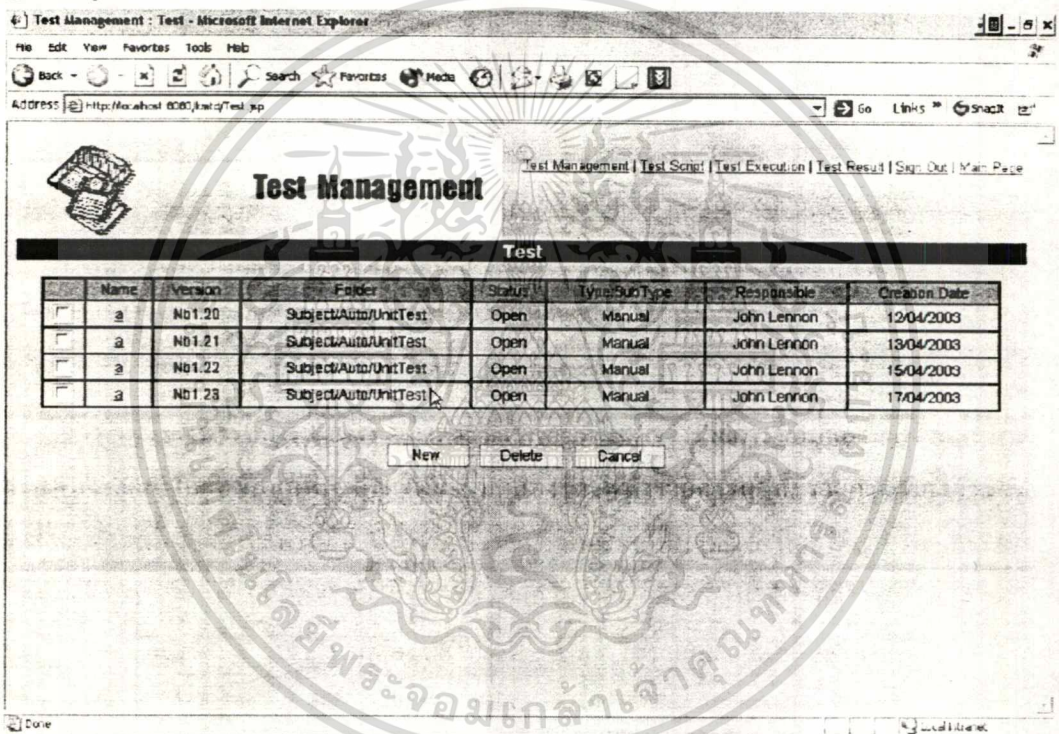
สำหรับข้อมูลกรณีการทดสอบจะมีการจัดการข้อมูลเกี่ยวกับขั้นตอนการทดสอบของแต่ละกรณีภายในด้วย (Step)

ในรายละเอียดแต่ละส่วนข้างต้นจะมีลักษณะการทำงานคล้ายกันต่างกันที่ข้อมูลรายละเอียดเท่านั้น กล่าวคือทุกส่วนจะประกอบด้วยการทำงานดังนี้ การแสดงรายการ, การสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลใหม่, การแก้ไขเปลี่ยนแปลงข้อมูลเดิม, การดูรายละเอียดในแต่ละชุดข้อมูล และการลบข้อมูล ดังนั้นลักษณะการออกแบบหน้าจอจะมีส่วนคล้ายกัน โดยในที่นี้จะขออธิบายถึงหน้าจอเกี่ยวกับข้อมูลส่วนกรณีการทดสอบซึ่งมีรายละเอียดของข้อมูลซับซ้อน

สำหรับหน้าจอการแสดงผลรายการจะประกอบด้วยตารางรายการข้อมูลกรณีการทดสอบซึ่งจะแสดงรายละเอียดข้อมูลที่สำคัญ โดยจะมีกล่องข้างหน้ารายการแต่ละรายการเพื่อไว้ใช้เลือกรายการข้อมูลที่ต้องการลบโดยจะทำการลบรายการนั้นถ้ากดปุ่ม Delete ที่อยู่ใต้ตาราง และสำหรับแต่ละรายการจะสามารถคลิกเพื่อเข้าไปดูหรือแก้ไขรายละเอียดที่ไม่ได้แสดงในตารางได้ และถ้ากดปุ่ม New ที่อยู่ใต้ตารางจะเชื่อมต่อไปยังหน้าจอทำการใส่รายละเอียดข้อมูลใหม่เพื่อบันทึกเข้าสู่ฐานข้อมูล



รูปที่ 5.5 หน้าจอแสดงผลรายการข้อมูลกรณีทดสอบ (Test)

สำหรับหน้าจอการแสดงผลรายละเอียดของกรณีทดสอบเพื่อการเพิ่มหรือแก้ไข จะมีลักษณะรูปแบบเหมือนกันแต่ต่างกันที่ผลการทำงานหลังจากกดปุ่มตกลง และหากเป็นการแก้ไขข้อมูลจะมีการแสดงข้อมูลเดิมในฐานข้อมูลให้ผู้ใช้ก่อน ซึ่งสำหรับหน้าจอนี้ผู้ใช้สามารถกรอกรายละเอียดต่างๆ ตามกล่องข้อความ และเนื่องจากกรณีการทดสอบแต่ละรายการจะมีขั้นตอน (Step) ในการทำการทดสอบรวมอยู่ด้วยจึงต้องเลือกขั้นตอนที่ต้องการเข้ามาด้วยการคลิก Attach ซึ่งจะทำการเปิดหน้าจอรายการข้อมูลขั้นตอนการทดสอบ (Step) ที่มีอยู่แล้วโดยผู้ใช้สามารถเลือกรายการที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รวมเข้ามาได้ หรือหากไม่มีขั้นตอนดังกล่าวอยู่ในระบบก็สามารถสร้างใหม่ได้เหมือนข้อมูลการทดสอบอื่นๆ ที่ได้กล่าวไว้ข้างต้น

The screenshot shows the 'Test Management' application window. At the top, there is a navigation bar with 'Test Management | Test Script | Test Execution | Test Result | Sign Out | Main Page'. Below this is a 'Test Detail' header. The main area contains a form with the following fields and controls:

- Name*: [Text Input]
- Version: [Text Input]
- Description: [Text Area]
- Type: [Text Input]
- Sub Type: [Text Input]
- Status: Ready Design
- Folder: [Text Input]
- Responsible: [Dropdown Menu]
- Creation Date: [Text Input]
- Time Out: [Text Input]
- Attachment: [Text Input] [Browse...]
- Plan-Scheduling Date: [Text Input]
- Plan-Scheduling Time: [Text Input] hh:mm:ss
- Runtime attachment: [Text Input] [Browse...]
- Extended parameter: [Text Input]
- Steps:
- Step Name 1: [Text Input]

At the bottom of the form are 'OK' and 'Cancel' buttons.

รูปที่ 5.6 หน้าจอแสดงรายละเอียดของกรณีทดสอบ (Test Detail)

5.2.3 หน้าจอการจัดการเกี่ยวกับข้อมูลเอกสารสคริปต์สำหรับการทดสอบ

หน้าจอที่เกี่ยวกับการจัดการข้อมูลของเอกสารสคริปต์แบ่งได้เป็น หน้าจอสำหรับแสดงรายการ หน้าจอสำหรับแสดงข้อมูลในเอกสารสคริปต์ และหน้าจอสำหรับการสร้างเอกสารสคริปต์ สำหรับหน้าจอการแสดงผลรายการเอกสารสคริปต์นั้นจะมีลักษณะคล้ายกับหน้าจอแสดงผลรายการข้อมูลการทดสอบที่กล่าวไปข้างต้น ต่างกันตรงที่ในหน้าจอนี้จะสามารถลบและลิงค์เพื่อดูรายละเอียดข้อมูลของแต่ละรายการเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test Management | Test Script | Test Execution | Test Result | Sign Out | Main Page

Test Script

Test Script List By:

Name	Host Name	Path	Responsible	Project	Creation Date
<input type="checkbox"/>	localhost	Subject/Auto/UnitTest	John Lennon	A project	12/04/2004
<input type="checkbox"/>	localhost	Subject/Auto/UnitTest	John Lennon	A project	13/04/2004
<input type="checkbox"/>	localhost	Subject/Auto/UnitTest	John Lennon	B project	15/04/2004
<input type="checkbox"/>	localhost	Subject/Auto/UnitTest	John Lennon	A project	17/04/2004

รูปที่ 5.7 หน้าจอแสดงรายการเอกสารสคริปต์ (Test Script List)

สำหรับหน้าจอแสดงรายละเอียดในเอกสารสคริปต์จะเป็นหน้าจอที่แสดงข้อมูลทั้งหมดในเอกสาร โดยการแปลงเอกสารสคริปต์ให้เป็นเอกสารเอชทีเอ็มแอลเพื่อแสดงผลในหน้าจอแก่ผู้ใช้ ซึ่งในหน้าจอนี้สามารถทำการบันทึกเป็นเพื่อเป็นต้นแบบแก่เอกสารสคริปต์ที่จะสร้างใหม่โดยกดปุ่ม Save As

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Test Script

Test Management | Test Script | Test Execution | Test Result | Sign Out | Main Page

Test Script Information

Test Script Name : ?Script Name?
Project : ?Project Name?

Test List :	?Test List Name?				
Test List Description :	?Description?				
Status :	?Test List Status?				
Test Folder :	?Subject\A\IC\UnitTest1?	Test :	?Test Name?	Status :	?Test Status?

Test :	?Test Name?	Folder :	?Test Folder?
Test Description :	?Description?		
Creation date :	?DD/MM/YYYY?	Type :	?Test Type?
Responsible to :	?Test Responsible User?	Attachment :	?Test Attachment?
Status :	?Test Status?	Run-time attachments :	?Test's Runtime Attachment?
Plan scheduling date :	?DDMMYYYY?	Timeouts :	?Test timeout?
Plan scheduling time :	?HH:MM?		

Step	Description	Expected Value
Step Name 1	'Step desc'	?Step 1 Expected?

Run :	?Run Name?		
Responsible to :	?Test Responsible User?	Exec Date :	?DDMMYYYY?
Host name :	?Run host name?	Exec Time :	?HH:MM:SS?
Path :	?Run path?	Duration :	?Run duration?

Step	Description	Expected	Actual	Status
'Step in Run 1 Name'	'Step in Run 1 Description'	'Step in Run 1 Expected'	'Step in Run 1 Actual'	'Step in Run 1 Status'

Save As

Cancel

รูปที่ 5.8 หน้าจอแสดงรายการละเอียดในเอกสารสคริปต์ (Test Script Detail)

สำหรับหน้าจอการสร้างเอกสารสคริปต์จะเป็นหน้าจอเพื่อใส่ข้อมูลทั่วไปสำหรับเอกสารสคริปต์ที่ต้องการสร้าง ซึ่งจะมีกล่องข้อความไว้ให้ใส่ข้อมูลและจะต้องเลือกกลุ่มการกรณิการทดสอบที่ต้องการจะรวมในเอกสารสคริปต์ดังกล่าวโดยวิธีการเลือกจะเหมือนกับการ Attach ที่กล่าวไว้ข้างต้น นอกจากนี้ก่อนเข้าใช้หน้าจอนี้ผู้ใช้สามารถเลือกเอกสารสคริปต์ที่เคยสร้างจากหน้าจอแสดงรายละเอียดเอกสารสคริปต์เพื่อเป็นต้นแบบ กล่าวคือจะสามารถนำข้อมูลทั่วไปของเอกสารสคริปต์ที่มีอยู่เดิมมาแก้ไขในรายละเอียดบางอย่างได้ซึ่งเป็นการนำมาใช้ใหม่นั้นเอง และสามารถคลิกเข้าไปดูรายละเอียดของข้อมูลการทดสอบที่รวมอยู่ในเอกสารสคริปต์ได้หากต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Report: - Microsoft Internet Explorer

Address: http://localhost:8080/screen-cding/Reports.jsp

Test Management | Test Script | Test Execution | Test Result | Sign Out | Main Page

Reports

Result Reports List By: Go

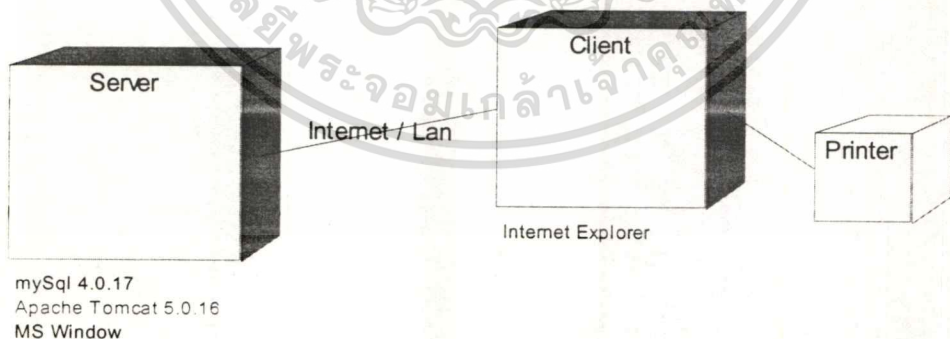
<input type="checkbox"/>	Test Result Name	Related Test Script	Test Script Desc	Test Operator	Project	Creation Date
<input type="checkbox"/>	TR_TC(1)12-08-04	TestScript Name1	Test Script Name 1 : Desc	John Lennon	A project	12/08/2004
<input type="checkbox"/>	TR_TC(1)22-08-04	TestScript Name1	Test Script Name 1 : Desc	John Lennon	A project	22/08/2004
<input type="checkbox"/>	TR_TC(2)12-08-04	TestScript Name2	Test Script Name 2 : Desc	John Lennon	B project	12/08/2004

Delete Cancel

รูปที่ 5.11 หน้าจอการจัดการเกี่ยวกับผลการทดสอบ (Test Result List)

5.3 การติดตั้งระบบ (System Deployment)

สำหรับการติดตั้งระบบสามารถจัดสถาปัตยกรรมได้ดังรูปที่ 5.12



รูปที่ 5.12 คือพลอยเม้นต์โคอะแกรมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทสรุปและข้อเสนอแนะ

6.1 สรุปการพัฒนาระบบ

โครงการพัฒนาระบบงานนี้ เป็นการพัฒนาขึ้นเพื่อจำลองการทำงานของเว็บแอปพลิเคชัน สำหรับทำการทดสอบซอฟต์แวร์แบบกึ่งอัตโนมัติ โดยการนำเอาแนวคิดเชิงวัตถุมาใช้ในการ ออกแบบและพัฒนา ซึ่งผลที่ได้จากการพัฒนาระบบมีดังนี้

- เว็บไซต์สำหรับให้ผู้ทำการทดสอบโปรแกรม (Tester, Test Operator) สามารถใช้เป็น เครื่องมือสำหรับการทดสอบ โปรแกรมแบบกึ่งอัตโนมัติบนระบบปฏิบัติการ Window โดยมีการทำงานตามลำดับดังต่อไปนี้
 - สร้างข้อมูลสำหรับการทดสอบซอฟต์แวร์ในฐานข้อมูล ได้แก่ กลุ่มการ ทดสอบ, กรณีการทดสอบ, ขั้นตอนการทดสอบ, โครงการที่ต้องการทดสอบ เป็นต้น
 - สร้างเอกสารสคริปต์สำหรับการทดสอบ โดยการเลือกข้อมูลสำหรับการ ทดสอบที่ได้สร้างไว้ในฐานข้อมูลในข้างต้น
 - เรียกทดสอบ โปรแกรมตามเอกสารสคริปต์ที่สร้างไว้แล้ว โดยต้องมีโปรแกรม ที่ต้องการทดสอบติดตั้งไว้ที่ตำแหน่งที่ไคลเอนท์ที่ระบุในเอกสารสคริปต์ ซึ่ง ระบบจะทำการเรียกโปรแกรมนั้น ๆ ขึ้นมาทดสอบ และสร้างเอกสารผลการ ทดสอบนั้น ๆ พร้อมแจ้งให้ผู้ทดสอบ โปรแกรมทราบถึงผลการทดสอบ
 - ผู้ใช้สามารถดูผลการทดสอบระบบของการเรียกทดสอบโปรแกรมสำหรับแต่ ละเอกสารสคริปต์ได้หากต้องการ
- การใช้เครื่องมือทดสอบซอฟต์แวร์แบบกึ่งอัตโนมัติจะช่วยให้ผู้ทดสอบระบบสามารถ นำเอกสารสคริปต์สำหรับการทดสอบมาใช้ทดสอบในเรื่องเดียวกันนั้นได้หลายครั้ง โดยไม่ต้องทำการเตรียมใหม่เหมือนการทดสอบแบบปกติ และข้อมูลสำหรับสร้าง เอกสารสคริปต์หนึ่ง ๆ สามารถนำไปสร้างเอกสารสคริปต์อื่นได้อีกนั่นคือสามารถนำ ข้อมูลนั้น ๆ ไปใช้ใหม่ได้ หรืออาจนำไปปรับเปลี่ยนแก้ไขและบันทึกเป็นข้อมูลอีกชุด หนึ่งก็สามารถทำได้

- เนื่องด้วยเครื่องมือการทดสอบซอฟต์แวร์แบบกึ่งอัตโนมัติได้ออกแบบบนพื้นฐานการทำงานของยูนิเวอร์แซลเทสต์ไครเวอร์ซึ่งมีการนำฐานข้อมูลกลางบนเครื่องแม่ข่ายมาใช้เพื่อเก็บข้อมูลการทดสอบ จึงทำให้สามารถนำข้อมูลดังกล่าวไปใช้ใหม่หรือนำไปปรับใช้ในอนาคตได้สะดวกโดยไม่ต้องขึ้นกับตัวบุคคลผู้สร้างข้อมูลดังกล่าวดังเช่นในอดีต
- ได้รับความรู้จากการศึกษากระบวนการทดสอบซอฟต์แวร์แบบต่าง ๆ รวมทั้งแบบที่มีการใช้เอกสารสคริปต์ในการทดสอบ และความรู้จากหลักการทำงานยูนิเวอร์แซลเทสต์ไครเวอร์ซึ่งเป็นแนวคิดหนึ่งในการสร้างเครื่องมือหรือระบบที่ช่วยในการทดสอบโปรแกรมแบบอัตโนมัติด้วยเอกสารสคริปต์ นอกจากนี้ยังได้รับความรู้เพิ่มเติมในการนำเอา Struts Framework และ Java Library ในบทที่ 5 มาใช้พัฒนาระบบงาน Web Application

6.2 ข้อเสนอแนะ

อย่างไรก็ตามระบบที่พัฒนานี้เป็นการจำลองการทำงานของยูนิเวอร์แซลเทสต์ไครเวอร์สำหรับการทดสอบซอฟต์แวร์แบบกึ่งอัตโนมัติโดยการใช้ Java Web Application เพื่อศึกษาการทำงานตามแนวคิดนี้ และทดลองสร้าง Web Application ที่รองรับแนวคิดการทำงานนี้ ในอนาคตถ้ามีการพัฒนาต่อไปเพื่อให้สามารถนำไปใช้จริงได้อย่างสมบูรณ์จำเป็นต้องปรับปรุงและพัฒนาเพิ่มในส่วนต่างๆ ดังนี้

- การพัฒนาให้การทดสอบเป็นแบบอัตโนมัติมากขึ้น ซึ่งอาจพัฒนาเพิ่มได้ดังนี้
 - การตรวจจับการเปลี่ยนแปลงข้อมูลสำหรับการทดสอบหากมีการเปลี่ยนแปลง ซึ่งในกรณีที่มีการตั้งค่าให้มีการสร้างสคริปต์สำหรับการทดสอบทุกครั้งการเปลี่ยนแปลง ระบบอาจทำการสร้างสคริปต์ให้ใหม่และหลังจากนั้นจะแจ้งการมีสคริปต์เกิดขึ้นใหม่ให้กับผู้ทดสอบระบบ โดยที่ในการสร้างสคริปต์ใหม่ลักษณะนี้อาจเป็นการสร้างเป็นรอบช่วงเวลา เช่น ทุก 7.30 น.ของทุกวัน เป็นต้น
 - การตั้งเวลาการทดสอบโปรแกรม ซึ่งอาจระบบทำการเรียกเอกสารสคริปต์ขึ้นมาทดสอบตามวันและเวลาที่ได้ตั้งไว้
 - การแจ้งให้ผู้ทดสอบทราบว่าเมื่อเอกสารสคริปต์ใดบ้างที่ตนรับผิดชอบอยู่ และมีกำหนดให้ทำการเรียกการทดสอบภายในวันนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การจัดการกับเอกสารของผลการทดสอบที่มีจำนวนมากเกินไปแบบอัตโนมัติ
เนื่องจากการเรียกทดสอบโปรแกรมหลายครั้งสำหรับเอกสารสคริปต์หนึ่ง ๆ
- การพัฒนาระบบให้สามารถทำการทดสอบซอฟต์แวร์ได้บนระบบปฏิบัติการอื่น ๆ
นอกจาก Windows และทำการเรียกโปรแกรมที่ต้องการทดสอบด้วยวิธี Remote เข้า
เครื่องได้ กล่าวคืออาจมีกระบวนการที่ต้องการให้โปรแกรมที่ต้องการทดสอบทำงาน
ด้วยนอกเหนือจากที่ระบุตำแหน่งของโปรแกรม
- การส่งอีเมลผลการทดสอบซอฟต์แวร์ในรูปแบบเอกสารแนบให้ผู้ทดสอบทราบ เพื่อ
อำนวยความสะดวกแก่ผู้ทดสอบในการตรวจสอบผล โดยไม่ต้องเข้าระบบมาดูผลการ
ทดสอบเอง
- การสร้างรายงานผลการทดสอบความถี่ต่าง ๆ ให้หลากหลาย เช่น ตามโครงการ
พัฒนาซอฟต์แวร์ต่าง ๆ, ตามประเภทของการทดสอบ หรือ ตามผู้รับผิดชอบการ
ทดสอบ เป็นต้น เพื่อใช้เป็นข้อมูลสำหรับการประเมินประสิทธิภาพการพัฒนา
ซอฟต์แวร์
- แนวคิดยูนิเวอร์แซลเทสต์เคสไดร์เวอร์ที่ได้ศึกษานั้น เป็นแนวคิดขององค์กรเอกชน
ต่างประเทศ ซึ่งข้อมูลสำหรับการศึกษานั้น ไม่ได้มีเป็นเอกสารทางวิชาการที่สามารถ
ค้นได้จากฐานข้อมูลวิชาการ ดังนั้นหากต้องการรายละเอียดมากกว่านี้สำหรับการ
พัฒนาระบบต่ออาจจำเป็นต้องติดต่อกับองค์กรนั้นหากทำได้ หนึ่งในการศึกษาแนวคิด
นี้ควรมีความรู้พื้นฐานในวิชา Software Engineering เกี่ยวกับการทดสอบซอฟต์แวร์
แบบอัตโนมัติ บนพื้นฐานของการใช้เอกสารสคริปต์สำหรับการทดสอบมาก่อน
- เทคโนโลยีและเครื่องมือที่ใช้ในการพัฒนา เช่น Struts Framework เป็นเทคโนโลยีใหม่
และเป็นแนวคิดที่ใช้สำหรับการพัฒนาระบบ Java Web Application โดยเฉพาะซึ่ง
จำเป็นต้องใช้เวลาในการศึกษาและเรียนรู้การใช้งานพอสมควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- ขยัน จันทรสถาวร. 2544. **XML Extensible Markup Language**. กรุงเทพฯ:
เอ อาร์ อินฟอร์เมชัน แอนด์ พับลิเคชัน.
- ชัยดำรงค์ อุทธิรัมย์,ผู้แปล. 2543. **ปฏิบัติการ XML เทคโนโลยีเว็บสุดร้อน**. กรุงเทพฯ:
สามย่าน.com.
- วีระศักดิ์ ชิงถาวร. 2543. **Java Programming. 2 เล่ม**. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.
- ศิวณัฐ มาศสุรางค์. 2545. **เรียนรู้ JSP ฉบับพร้อมใช้งาน**. กรุงเทพฯ:
เอ อาร์ อินฟอร์เมชัน แอนด์ พับลิเคชัน.
- สงกรานต์ ทองสว่าง. 2545. **My SQL ระบบฐานข้อมูลสำหรับอินเทอร์เน็ต**. กรุงเทพฯ:
ซีเอ็ดยูเคชั่น.
- สุนทริน วงศ์ศิริกุล. ม.ป.ป. **พัฒนาโมเดลยูทีใหม่ UML Unified Modeling Language**. กรุงเทพฯ:
ซัคเซส มีเดีย.
- Apache . 2004 a. **Apache Tomcat 5.0 Release Builds**. Available :
<http://jakarta.apache.org/tomcat-5> .
- Apache. 2004 b. **Jakarta Struts 1.1 Release Builds** Available : <http://jakarta.apache.org/struts> .
- Bahrami, A. 1999. **Object oriented systems development**. Singapore: Irwing/McGraw-Hill.
- Cordrey, G. L. 2002. “Universal Test Driver: A Data-Driven Framework for Script-
Based Testing”. Houston: Common Service Automation BMC Software
- Patton, R. 2000. **Software Testing**. Indiana: Sams Publishing.
- Poston, M. R. 1994 a. “Moving from Manual to Automated Test Execution”. **Comm. ACM**.
37, 9(Sept. 1994): 124-129.
- Poston, M. R. 1994 b. “Automating Testing from Object Models”.. **Comm. ACM**. 37,
9(Sept. 1994): 48-58.
- Silberschatz, A. et al. 2002. **Database System Concepts**. Fourth Edition. New York:
McGraw-Hill.

ประวัติผู้เขียน

ชื่อผู้เขียน	นางสาว จุฬภาภา ชูอำไพ
สถานที่เกิด	กรุงเทพมหานคร
ปริญญาตรี	วิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์ (เกียรตินิยมอันดับสอง)
สถานที่สำเร็จการศึกษา	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีที่สำเร็จการศึกษา	2544



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการเอกสาร

1. ภาคผนวก ก. ตัวอย่าง DTD ของเอกสารของการทดสอบ (UTD)
2. ภาคผนวก ข. ตัวอย่าง Generated Test Script ของ Universal Test Driver Framework
3. ภาคผนวก ค. Style Sheet สำหรับเอกสารการทดสอบ (UTD)
4. ภาคผนวก ง. Style Sheet สำหรับ Test และ Step (UTD)
5. ภาคผนวก จ. ตัวอย่าง เอกสาร XML ของการ Get Tests (UTD)
6. ภาคผนวก ฉ. ตัวอย่าง เอกสาร XML ของการทดสอบ (UTD)
7. ภาคผนวก ช. ตัวอย่าง DTD ของเอกสารของการทดสอบ (KMTD)
8. ภาคผนวก ซ. Style Sheet สำหรับ Test และ Step (KMTD)
9. ภาคผนวก ฌ. ตัวอย่าง เอกสาร XML ของการทดสอบ (KMTD)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ตัวอย่าง DTD ของเอกสารของการทดสอบ (UTD)

```

<!-- KMITL Test Document DTD - version 1.0 --->
<!--Copyright (c) 2003-2004, Infomation Technology @ KMITL.ac.th-->

<!ELEMENT BMCTD      (DESCRIPTION?, REPOSITORY? ) >
<!ELEMENT REPOSITORY (PROJECT) * >
<!ELEMENT PROJECT    (DESCRIPTION?, (TESTLIST | FOLDER) * ) >
<!ELEMENT TESTLIST   (DESCRIPTION?, (TEST) * ) >
<!ELEMENT FOLDER     (FOLDER | TEST) * >
<!ELEMENT TEST       (DESCRIPTION?, (RUN | STEP) * ) >
<!ELEMENT RUN        ((STEP) * , STATUS? , DURATION?) >
<!ELEMENT STEP       (DESCRIPTION? , EXPECTED? , ACTUAL? , STATUS?) >
<!ELEMENT DESCRIPTION (#PCDATA) >
<!ELEMENT EXPECTED   (#PCDATA) >
<!ELEMENT ACTUAL     (#PCDATA) >
<!ELEMENT STATUS     (#PCDATA) >
<!ELEMENT DURATION   (#PCDATA) >
<!ATTLIST BMCTD
    title          CDATA " (untitled) "
    author         CDATA " (unknown) "
    creation_date  CDATA " (unknown) "
    last_modified  CDATA " (unknown) "
    copyright      CDATA "Copyright (c) 2003-2004,Infomation Technology @
KMITL.ac.th "
>
<!ATTLIST REPOSITORY
    name          CDATA #REQUIRED
    settings      CDATA #IMPLIED
>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<!ATTLIST PROJECT

name CDATA #REQUIRED

settings CDATA #IMPLIED

>

<!ATTLIST TESTLIST

name CDATA #REQUIRED

status CDATA #IMPLIED

>

<!ATTLIST FOLDER

name CDATA #REQUIRED

folder CDATA #IMPLIED

>

<!ATTLIST TEST

name CDATA #REQUIRED

number CDATA #IMPLIED

type CDATA "MANUAL"

subtype CDATA #IMPLIED

status (Ready | Design) "Ready"

folder CDATA #IMPLIED

responsible CDATA " "

creation_date CDATA #IMPLIED

timeout CDATA #IMPLIED

attachment CDATA #IMPLIED

plan_scheduling_date CDATA #IMPLIED

plan_scheduling_time CDATA #IMPLIED

runtime_attachment CDATA #IMPLIED

extended_parameters CDATA #IMPLIED

>

<!ATTLIST RUN

name CDATA #REQUIRED

tester_name CDATA " "

execution_date CDATA #IMPLIED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

execution_time      CDATA #IMPLIED
host_name           CDATA #IMPLIED
path                CDATA " "

```

>

<!ATTLIST STEP

```

name                CDATA #REQUIRED
execution_date      CDATA #IMPLIED
execution_time      CDATA #IMPLIED
path                CDATA #IMPLIED
line_number         CDATA " 0 "

```

>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

ตัวอย่าง Generated Test Script ของ Universal Test Driver Framework

```

# -----
# Author:      Julapa ChuAmpai
# Created:     29/08/2003
# Last modified: Fri Aug 29 12:23:30 2003
# Copyright:   copyright © 2003-2004, Information Technology @ KMITL.ac.th
# -----
# This script was created by the KMITL Test Driver.
# For more information, contact the Faculty of Information Technology @ KMITL
#
# Requires the Universal Test Driver.
#
# To run this script:
# - log on to the testing machine
# - source the runtime profiles
# - create an empty directory and "cd" to it
# - source the script and redirect its output to a .xml file
# -----
. print_start_tag_driver 'Machine List for the SFA orderstatus Testing'
. print_start_tag_repository 'WCSDEV database' 'sfa2000'
. print_start_tag_project 'SFA_premier_marketing' 'salerep'
# -----
# TESTLIST: OrderStatus on PDA
# -----
. print_start_tag_testlist 'OrderStatus_PDA-'$TEST_SET_SUFFIX '$SFA_Orderstatus'
# -----
#TEST: 'listOrderStatus'
# -----
. print_start_tag_test 'list' 'premier/sfa/order/orderstatusform' ''

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
.print_start_tag_step 'test_program' 'test_listOrderStatus'
```

```
.print_end_tag_step
```

```
.print_start_tag_run
```

```
#TEST SUBTYPE: ORDER
```

```
TEST_SCRIPT = orderstatus_test
```

```
.create_test_scriptf $TEST_SCRIPT
```

```
.add_step $TEST_SCRIPT "test_listOrderStatus" ""
```

```
.run_test_script $TEST_SCRIPT 180 2> /dev/null
```

```
.print_end_tag_run
```

```
.print_end_tag_test
```

```
# -----
```

```
.print_end_tag_testlist
```

```
.print_end_tag_project
```

```
.print_end_tag_repository
```

```
.print_end_tag_driver
```

ภาคผนวก ค.

Style Sheet สำหรับเอกสารการทดสอบ (UTD)

```

<?xml version='1.0'?>

<!-- KMITL Test Document Style Sheet for the KMITL Test Driver -->
<!-- KMITL Test Driver v1.0 -->
<!-- Julapa ChuAmpai -01/21/04 -->
<!-- Copyright (c) 2004, Information Technology @ KMITL.ac.th -->
<!-- Compatible with the KMITL Test Document DTD v.1.0 -->

<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/xsl/" version="1.0">

<!-- Add your imports here-->
<xsl:import href="subtype1.xsl"/>

<xsl:preserve-space elements="*"/>

<xsl:template match="/">
<xsl:for-each select="KMTD">
# <xsl:value-of select="@title"/>
# -----
# Author: <xsl:value-of select="@author"/>
# Created: <xsl:value-of select="@creation_date"/>
# Last modified: <xsl:value-of select="@last_modified"/>
# Copyright: <xsl:value-of select="@copyright"/>
# -----
# This script was created by the KMITL Test Driver
# For more information, contact the Faculty of Information Technology @ KMITL
#
# Requires the Universal Test Driver.
#
# To run this script:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

# - log on to the testing machine
# - source the runtime profiles
# - create an empty directory and "cd" to it
# - source the script and redirect its output to a .xml file
# -----

.print_start_tag_kmttd '<xsl:value-of select="@title"/>'

<xsl:for-each select="REPOSITORY">
.print_start_tag_repository '<xsl:value-of select="@name"/>' '<xsl:value-of
select="@settings"/>'

<xsl:for-each select="PROJECT">
.print_start_tag_project '<xsl:value-of select="@name"/>' '<xsl:value-of select="@settings"/>'

<xsl:for-each select="TESTLIST">
# -----
# TESTLIST: <xsl:value-of select="@name"/>
# -----

.print_start_tag_testlist '<xsl:value-of select="@name"/>-'$TEST_SET_SUFFIX
$LOCAL_TARGET

<xsl:apply-templates select="TEST"/>

.print_end_tag_testlist
</xsl:for-each><!--TESTLIST-->

.print_end_tag_project
</xsl:for-each><!--PROJECT-->

.print_end_tag_repository
</xsl:for-each><!--REPOSITORY-->

.print_end_tag_kmttd
</xsl:for-each><!--KMTD-->

```

```

</xsl:template>

<xsl:template match="TEST">
# -----
# TEST: <xsl:value-of select="@name"/>
# -----

.print_start_tag_test '<xsl:value-of select="@name"/>' '<xsl:value-of select="@folder"/>'
'<xsl:value-of select="DESCRIPTION"/>'

<xsl:for-each select="STEP">
.print_start_tag_step '<xsl:value-of select="@name"/>' '<xsl:value-of
select="DESCRIPTION"/>' '<xsl:value-of select="EXPECTED"/>'
.print_end_tag_step
</xsl:for-each><!--STEP-->

.print_start_tag_run

<xsl:apply-imports/>

.print_end_tag_run
.print_end_tag_test
# -----
</xsl:template>

</xsl:stylesheet>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง.

Style Sheet สำหรับ Test และ Step (UTD)

```

<?xml version='1.0'?>

<!-- KMITL Test Driver Import Style Sheet for Subtype1 test cases -->
<!-- KMITL Test Driver v1.0 -->
<!-- Julapa ChuAmpai -01/21/04 -->
<!-- Copyright (c) 2004, Infomation Technology @ KMITL.ac.th -->
<!-- Compatible with the KMITL Test Document DTD v.1.0 -->
<!-- -->
<!-- Usage: This stylesheet must be imported by kmtd.xsl -->

<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/xsl/" version="1.0">
<xsl:preserve-space elements="*" />

<xsl:template match="TEST[@subtype='subtype1']">
# TEST SUBTYPE: subtype1

TEST_SCRIPT = subtype1_test
. create_test_script $TEST_SCRIPT

<xsl:apply-templates select="STEP" />

. run_test_script $TEST_SCRIPT 180 2> /dev/null

</xsl:template>

<xsl:template match="TEST@subtype='subtype1']/STEP[@name='test_location']">
. add_step $TEST_SCRIPT "cp <xsl:value-of select="DESCRIPTION"/>/*.mof ." ""
</xsl:template>

<xsl:template match="TEST@subtype='subtype1']/STEP[@name='copy_directory']">
. add_step $TEST_SCRIPT "cp -r <xsl:value-of select="DESCRIPTION"/> ." ""
</xsl:template>

```

```

<xsl:template match=s"TEST@subtype='subtype1']/STEP[@name='test_program']">
. add_step $TEST_SCRIPT "<xsl:value-of select="DESCRIPTION"/>" ""
</xsl:template>

</xsl:stylesheet>

```



/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ.

ตัวอย่าง เอกสาร XML ของการ Get Tests (UTD)

```

<?xml version= "1.0" encoding= "utf-8"?>
<?xml-stylesheet type= "text/xsl" href= http://localhost/utd/xml/kmtd_html.xsl?>
<!DOCTYPE KMTD SYSTEM "http://localhost:8080/utd/xml/kmtd.dtd">
<!-- Validates against the KMITL Test Document DTD v1.0-->
<!--Use with tdxml to extract sample tests from Test Director-->
<KMTD title = "KM Packaging Smoke Tests" author= "Automation Team" creation_date =
"04/30/2003">
  <REPOSITORY
    name = "Test Director database"
    settings= "TDAPIServer host name"
  >
    <PROJECT name= "TestDirectorProjectName">
      <TESTLIST name= "KMPkgTestset">
        <?tdxml include tests runs steps?>
      </TESTLIST>
    </PROJECT>
  </REPOSITORY>
</KMTD>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ตัวอย่าง เอกสาร XML ของการทดสอบ (UTD)

```

<?xml version="1.0"?>
<!DOCTYPE KMTD SYSTEM "http://localhost:8080/utd/xml/kmtd.dtd">
<!-- Validates against the KMITL Test Document DTD v1.0-->
<!-- This document contains output from the KMITL Test Driver-->
<!-- For more information, contact the Faculty of Information Technology @ KMITL-->

<KMTD title="(untitled)"
  author="(unknown author)"
  creation_date="Fri Jun 21 2004 13:00:00 "
  last_modified="Fri Jun 21 2004 13:00:00 "
  copyright="Copyright (c) 2003-2004, Information Technology @ KMITL.ac.th ">
  <REPOSITORY
    name="KMITL Test Driver Database"
    settings="utd">
  <PROJECT
    name="proj_name1"
    settings="quest">
  <TESTLIST
    name="testlist_name1">
    <DESCRIPTION><![CDATA[description about test_list ,ex: test group
description]]></DESCRIPTION>
  <TEST
    name="test_name 1"
    folder="root/kmitl test driver/subtype1/testlist1">
    <DESCRIPTION><![CDATA[description about test]]></DESCRIPTION>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิพนธ์ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    <STEP name="test_program_step">
      <DESCRIPTION><![CDATA[description about step such as
test_inputboundary]]></DESCRIPTION>
      <EXPECTED><![CDATA[expected value of this
step]]></EXPECTED>
    </STEP>

    <RUN name="01/19/04 15:00:00"
      host_name="host1"
      execution_date="01/19/04 "
      execution_time="15:00:00">

      <STEP name="step_name"
        execution_date="01/19/04 "
        execution_time="15:00:01">
        <DESCRIPTION>test_inputboundary</DESCRIPTION>
        <ACTUAL><![CDATA[actual values from execution of this
step]]></ACTUAL>
        <STATUS>Passed</STATUS>
      </STEP>
      <STATUS>Passed</STATUS>
      <DURATION>1</DURATION>
    </RUN>
  </TEST>
</TESTLIST>
</PROJECT>
</REPOSITORY>
</KMTD>

```

ภาคผนวก ข.

ตัวอย่าง DTD ของเอกสารของการทดสอบ (KMTD)

```

<!ELEMENT KMTD          (DESCRIPTION?, PROJECT ) >
<!ELEMENT PROJECT      (DESCRIPTION?, (TESTLIST) * ) >
<!ELEMENT TESTLIST     (DESCRIPTION?, TEST * ) >
<!ELEMENT TEST         (DESCRIPTION?, (RUN | STEP) * ) >
<!ELEMENT RUN          ((STEP) * , STATUS? , DURATION?) >
<!ELEMENT STEP         (DESCRIPTION? , EXPECTED? , ACTUAL? , STATUS?) >
<!ELEMENT DESCRIPTION  (#PCDATA) >
<!ELEMENT EXPECTED    (#PCDATA) >
<!ELEMENT ACTUAL      (#PCDATA) >
<!ELEMENT STATUS      (#PCDATA) >
<!ELEMENT DURATION    (#PCDATA) >
<!ATTLIST KMTD
    id          CDATA#REQUIRED
    title       CDATA " (untitled) "
    author      CDATA " (unknown) "
    date        CDATA " (unknown) "
    copyright   CDATA "copyright © 2004, Information Technology @
KMITL.ac.th">
<!ATTLIST PROJECT
    id          CDATA#REQUIRED
    name        CDATA #REQUIRED
    settings    CDATA #IMPLIED>
<!ATTLIST TESTLIST
    id          CDATA#REQUIRED
    name        CDATA #REQUIRED
    status      CDATA #IMPLIED>
<!ATTLIST TEST
    id          CDATA#REQUIRED

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

name	CDATA #REQUIRED
type	CDATA "MANUAL"
subtype	CDATA #IMPLIED
status	(Ready Design) "Ready"
folder	CDATA #IMPLIED
responsible	CDATA ""
creation_date	CDATA #IMPLIED
timeout	CDATA #IMPLIED
attachment	CDATA #IMPLIED
plan_datetime	CDATA #IMPLIED
runtime_attach	CDATA #IMPLIED
extended_param	CDATA #IMPLIED>

<!ATTLIST RUN

| | |
|---------------|-----------------|
| name | CDATA #REQUIRED |
| tester | CDATA "" |
| exec_datetime | CDATA #IMPLIED |
| host_name | CDATA #IMPLIED |
| path | CDATA ""> |

<!ATTLIST STEP

| | |
|---------------|-----------------|
| id | CDATA#IMPLIED |
| name | CDATA #REQUIRED |
| exec_datetime | CDATA #IMPLIED |
| filename | CDATA #IMPLIED |
| line_number | CDATA " 0 "> |

ภาคผนวก ข.

Style Sheet สำหรับ Test และ Step (KMTD)

```

<?xml version='1.0'?>

<!-- KMITL Test Driver Import Style Sheet for Subtype1 test cases -->
<!-- KMITL Test Driver v1.0 -->
<!-- Julapa ChuAmpai -01/21/04 -->
<!-- Copyright (c) 2004, Infomation Technology @ KMITL.ac.th -->
<!-- Compatible with the KMITL Test Document DTD v.1.0 -->
<!-- -->
<!-- Usage: This stylesheet must be imported by kmt.d.xsl -->

<xsl:stylesheet " version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:preserve-space elements="*"/>

<xsl:template match="TEST[@subtype='subtype1']">
# TEST SUBTYPE: subtype1

TEST_SCRIPT = subtype1_test
.create_test_script $TEST_SCRIPT

<xsl:apply-templates select="STEP"/>

.run_test_script $TEST_SCRIPT
</xsl:template>

<xsl:template match="TEST@subtype='subtype1']/STEP[@name='initial']">
.add_step $TEST_SCRIPT " <xsl:value-of select="DESCRIPTION"/> ," "<xsl:value-of
select="TEST@attachment"/>"
</xsl:template>

<xsl:template match="TEST@subtype='subtype1']/STEP[@name='runtime_attach']">

```

```

.add_step $TEST_SCRIPT " <xsl:value-of select="DESCRIPTION"/> ." "<xsl:value-of
select="TEST@runtime_attach"/>"
</xsl:template>

<xsl:template match=s"TEST@subtype='subtype1']/STEP[@name='test_program']">
.add_step $TEST_SCRIPT "<xsl:value-of select="DESCRIPTION"/> ," "<xsl:value-of
select="TEST@extended_param"/>"
</xsl:template>

</xsl:stylesheet>

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ฉ.

ตัวอย่าง เอกสาร XML ของการทดสอบ (KMTD)

```

<?xml version="1.0"?>
<!DOCTYPE KMTD SYSTEM "http://localhost:8080/utd/xml/kmttd.dtd">
<!-- Validates against the KMITL Test Document DTD v1.0-->
<!-- This document contains output from the KMITL Test Driver-->
<!-- For more information, contact the Faculty of Information Technology @ KMITL-->

<KMTD title="(untitled)"
      author="(unknow author)"
      creation_date="Fri Jun 21 2004 13:00:00 "
      last_modified="Fri Jun 21 2004 13:00:00 "
      copyright="Copyright (c) 2003-2004, Information Technology @ KMITL.ac.th ">
  <PROJECT
    id=projId
    name="proj_name1"
    settings="quest">
    <TESTLIST id=testlistId
      name="testlist_name1">
      <DESCRIPTION><![CDATA[description about test_list ,ex: test group
description]]></DESCRIPTION>
      <TEST
        id=testId
        name="test_name-1"
        folder="root/kmitl test driver/subtype1/testlist1">
        <DESCRIPTION><![CDATA[description about
test]]></DESCRIPTION>
        <STEP id=stepId >
          name="test_program_step"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<DESCRIPTION><![CDATA[description about step
such as test_inputboundary]]></DESCRIPTION>
```

```
<EXPECTED><![CDATA[expected value of this
step]]></EXPECTED>
```

```
</STEP>
```

```
<RUN name="01/19/04 15:00:00"
```

```
host_name="host1"
```

```
execution_datetime="01/19/04 15:00:00" >
```

```
<STEP name="step_name"
```

```
execution_datetime="01/19/04 15:00:01" >
```

```
<DESCRIPTION>test_inputboundary</DESCRIPTION>
```

```
<ACTUAL><![CDATA[actual values from
execution of this step]]></ACTUAL>
```

```
<STATUS>Passed</STATUS>
```

```
</STEP>
```

```
<STATUS>Passed</STATUS>
```

```
<DURATION>1</DURATION>
```

```
</RUN>
```

```
</TEST>
```

```
</TESTLIST>
```

```
</PROJECT>
```

```
</KMTD>
```