

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจส.

เทสต์ไดร์เวอร์ GUI สำหรับทดสอบดีเอ็มแอลของฐานข้อมูล

Test Driver GUI for Testing Database's DML



วัน เดือน ปี..... 15 ก.พ. 2550
เลขทะเบียน..... 02249
เลขเรียกหนังสือ..... อท. 0621ท 2547
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจส."

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคฤดูร้อน ปีการศึกษา 2547

คณะเทคโนโลยีสารสนเทศ

เอกสารนี้เป็นเอกสารที่สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	เทสต์ไควร์วอร์กซ์สำหรับทดสอบดีเอ็มแอลของฐานข้อมูล
นักศึกษา	นายอาทิตย์ จิตบำรุงธรรม
อาจารย์ที่ปรึกษา	ผศ.ดร. พรฤดี เนติโสภากุล
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2547

บทคัดย่อ

ในปัจจุบันนี้ระบบฐานข้อมูลเข้ามามีบทบาทสำคัญกับองค์กรสมัยใหม่ แต่เราจะทดสอบระบบฐานข้อมูลอย่างไรว่ามีการทำงานที่ถูกต้อง และให้ประสิทธิภาพที่ยอมรับได้ โดยบทความนี้จะศึกษา และปรับปรุงระบบเทสต์ไควร์วอร์กซ์สำหรับการทดสอบภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language -DML) โดยผ่านส่วนประสานกราฟิกกับผู้ใช้ (GUI) ซึ่งช่วยให้ผู้ทดสอบสามารถสร้างแฟ้มอินพุตสำหรับการทดสอบดีเอ็มแอลของฐานข้อมูลได้ง่ายขึ้น จัดเก็บแฟ้มเอาต์พุตเพื่อในการวิเคราะห์ภายหลัง และกำหนดให้แฟ้มอินพุต และเอาต์พุตมีรูปแบบแฟ้มเป็น XML เพื่อเป็นมาตรฐานในการแลกเปลี่ยนชุดทดสอบ

Title Test Driver GUI for Testing Database's DML
Student Mr. Arthit Jitbumrungthum
Advisor Asst.Prof.Dr. Ponrudee Netisopakul
Level of Study Master of Science in Information Technology
Major Information Science
Academic Year 2004

Abstract

Database systems play an important role in most of the modern organizations, but how can we test their functionalities and effectiveness? This project developed a Test Driver Graphical User Interface for testing Database's Data Manipulation Language (DML). This tool helps testers to create tests for testing DML. The tool is able to produce output files for future analysis. It uses XML format in all interface files as a standard in exchanging test cases among testers.

กิตติกรรมประกาศ

โครงการพัฒนาระบบงานฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยความกรุณาของ ผศ.ดร. พร
ฤดี เนติโสภาคกุล ซึ่งเป็นอาจารย์ที่ปรึกษาที่ได้กรุณาให้คำปรึกษา ตักเตือน และเสนอแนะ
ข้อคิดเห็นต่างๆ เพื่อเป็นแนวทางในการศึกษา และทำให้ดำเนินการเป็นไปอย่างต่อเนื่อง

นอกจากนี้ ผู้ศึกษาต้องขอขอบคุณทุกคนในครอบครัว รวมถึงเพื่อนๆทุกคนที่คอยเป็น
กำลังใจให้เสมอมา คอยให้คำแนะนำและช่วยเหลือจนกระทั่งโครงการพัฒนาระบบงานนี้สำเร็จ
ลุล่วงไปได้

อาทิตย์ จิตบำรุงธรรม

10 พฤษภาคม 2548



สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VIII
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ขอบเขตของ โครงการพัฒนาระบบงาน.....	2
1.3 ขั้นตอนการดำเนินงาน.....	2
1.4 ข้อจำกัดของระบบ.....	2
2. ระบบเทคโนโลยีเว็บสำหรับทดสอบดีเอ็มแอล และทฤษฎีที่เกี่ยวข้อง.....	3
2.1 ทฤษฎีต่างๆ ที่เกี่ยวข้อง.....	3
2.2 ความหมายของ เทคโนโลยีเว็บ, Oracle และการเชื่อมต่อฐานข้อมูล.....	7
2.3 เครื่องมือที่ใช้ในการพัฒนาระบบ.....	11
3. วิเคราะห์ และออกแบบระบบงาน.....	17
3.1 ความต้องการของระบบงาน.....	17
3.2 Use Case Diagram.....	17
3.3 Use Case Description.....	19
3.4 Class Diagram.....	30
3.5 Sequence Diagrams.....	35
3.6 Activity Diagram.....	50
4 การสร้างดีเอ็มแอลเทสต์เคส และการทำนาย.....	52
4.1 การสร้างดีเอ็มแอล.....	52
4.2 การทำนาย.....	54

สารบัญ (ต่อ)

บทที่	หน้า
4.3	ข้อจำกัดของการสร้างดีเอ็มแอลเทสต์เคส.....58
4.4	ข้อจำกัดของการทำนาย.....59
5	การพัฒนาระบบ.....60
5.1	หน้าจอหลัก (MainWindow).....60
5.2	เมนูกำหนดค่าติดต่อ DBMS ที่จะทดสอบ (Setting Connection).....62
5.3	เมนูสร้างโครงการใหม่ (New test).....63
5.4	เมนูเพิ่ม DBMS ที่จะทดสอบ.....63
5.5	เมนูเพิ่ม Test Set ที่จะทดสอบ.....64
5.6	เมนูเพิ่ม Test Case ที่จะทดสอบ.....67
5.7	เมนูทดสอบ.....72
5.8	เมนูผลลัพธ์.....74
5.9	เพิ่มอินพุต เทสต์เซต และเอาต์พุต.....75
6	บทสรุป.....83
6.1	สรุปโครงการ.....83
6.2	ประโยชน์ที่ได้รับ.....83
6.3	ข้อเสนอแนะ.....83
	บรรณานุกรม.....84
	ประวัติผู้เขียน.....85

สารบัญตาราง

ตารางที่	หน้า
3.1	Setting Connection Use Case Description.....19
3.2	Manage Project Use Case Description.....20
3.3	Manage DBMS Use Case Description.....20
3.4	Manage Test Set Use Case Description.....20
3.5	Manage Test Case Use Case Description.....20
3.6	Execute Test Set Use Case Description.....21
3.7	View Test Result Use Case Description.....22
3.8	Create New Project Use Case Description.....22
3.9	Open Project Use Case Description.....23
3.10	Save Project Use Case Description.....23
3.11	Add DBMS Use Case Description.....23
3.12	Edit DBMS Use Case Description.....24
3.13	Remove DBMS Use Case Description.....24
3.14	Create Test Set Use Case Description.....24
3.15	Add New Test Set Use Case Description.....24
3.16	Copy to new Test Set Use Case Description.....25
3.17	Add Exist Test Set Use Case Description.....25
3.18	Import Test Set Use Case Description.....25
3.19	Remove Test Set Use Case Description.....26
3.20	Add Test Case Use Case Description.....26
3.21	Add Create table Use Case Description.....27
3.22	Add Drop table Use Case Description.....27
3.23	Add Select Use Case Description.....28
3.24	Add Insert Use Case Description.....28
3.25	Add Update Use Case Description.....28

สารบัญตาราง (ต่อ)

	หน้า
ตารางที่	
3.26 Add Delete Use Case Description.....	29
3.27 Generate Test Case Use Case Description.....	29
3.28 Edit Test Case Use Case Description.....	29
3.29 Remove Test Case Use Case Description.....	30
5.1 รายละเอียดไอคอนบนทูลบาร์.....	60



สารบัญรูป

รูปที่	หน้า
2.1 Data Type Definition.....	6
2.2 XML Schema.....	7
2.3 กระบวนการทดสอบข้อบกพร่อง.....	8
2.4 TestDriver และ Stub.....	9
2.5 หลักการการทำงานของโปรแกรม JAVA.....	11
2.6 การติดต่อระหว่างโปรแกรม Java กับฐานข้อมูลผ่าน JDBC.....	12
3.1 Use Case Diagram หลักของระบบทดสอบโค้ด.....	17
3.2 Use Case Diagram ของระบบย่อย Manage Project.....	18
3.3 Use Case Diagram ของระบบย่อย Manage DBMS.....	18
3.4 Use Case Diagram ของระบบย่อย Manage Test Set.....	18
3.5 Use Case Diagram ของระบบย่อย Manage Test Case.....	19
3.6 Class Diagram ของระบบทดสอบโค้ด.....	31
3.7 Sequence Diagram ของ Setting Connection.....	36
3.8 Sequence Diagram ของ Create New Project.....	36
3.9 Sequence Diagram ของ Open Project.....	37
3.10 Sequence Diagram ของ Save Project.....	38
3.11 Sequence Diagram ของ Execute Test Set.....	38
3.12 Sequence Diagram ของ View Test Result.....	39
3.13 Sequence Diagram ของ Add DBMS.....	40
3.14 Sequence Diagram ของ Edit DBMS.....	40
3.15 Sequence Diagram ของ Remove DBMS.....	41
3.16 Sequence Diagram ของ Create Test Set.....	41
3.17 Sequence Diagram ของ Add New Test Set.....	42
3.18 Sequence Diagram ของ Copy to new Test Set.....	42
3.19 Sequence Diagram ของ Add Exist Test Set.....	43

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.20	Sequence Diagram ของ Import Test Set.....43
3.21	Sequence Diagram ของ Remove Test Set.....44
3.22	Sequence Diagram ของ Add Test Case.....44
3.23	Sequence Diagram ของ Add Create table.....45
3.24	Sequence Diagram ของ Add Drop table.....45
3.25	Sequence Diagram ของ Add Select.....46
3.26	Sequence Diagram ของ Add Insert.....46
3.27	Sequence Diagram ของ Add Update.....47
3.28	Sequence Diagram ของ Add Delete.....47
3.29	Sequence Diagram ของ Generate Test Case.....48
3.30	Sequence Diagram ของ Edit Test Case.....49
3.31	Sequence Diagram ของ Remove Test Case.....49
3.32	Activity Diagram ของระบบทดสอบโค้ดที่เวอร์รี่.....51
5.1	หน้าจอหลักของระบบทดสอบโค้ดที่เวอร์รี่.....62
5.2	ไดอะล็อกแสดงรายชื่อการติดต่อที่สร้างขึ้น.....62
5.3	ไดอะล็อกรับ และแสดงค่าต่างๆในการติดต่อ DBMS.....63
5.4	หน้าจอเมนูสร้างโครงการใหม่.....63
5.5	หน้าจอเมนูเพิ่ม DBMS ที่จะทดสอบ.....64
5.6	ไดอะล็อกให้เลือกในการสร้าง Test Set ใหม่.....65
5.7	ไดอะล็อกใส่ชื่อ Test Set ใหม่ ใหม่.....65
5.8	หน้าจอหลังจากเลือกสร้าง Test Set ใหม่.....65
5.9	ไดอะล็อกเลือก Test Set ที่มีอยู่ในโครงการ.....66
5.10	ไดอะล็อกเปิดเลือกไฟล์ Test Set XML.....66
5.11	หน้าจอหลังจากเลือกสร้างใช้ Test Set เดิมหรือนำเข้า Test Set.....66
5.12	หน้าจอแสดงเมนูย่อยของ Test Case.....67

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.13	หน้าจอเมื่อเลือกเมนูย่อย Create.....68
5.14	หน้าจอเมื่อเลือกเมนูย่อย Drop.....68
5.15	หน้าจอเมื่อเลือกเมนูย่อย Select.....69
5.16	หน้าจอเมื่อเลือกเมนูย่อย Insert.....69
5.17	หน้าจอเมื่อเลือกเมนูย่อย Update.....70
5.18	หน้าจอเมื่อเลือกเมนูย่อย Delete.....70
5.19	หน้าจอเมื่อเลือกเมนูย่อย Generate.....71
5.20	หน้าจอแสดง Test Case ที่ Generate.....71
5.21	หน้าจอหลังจากสร้าง Test Case เรียบร้อยแล้ว.....72
5.22	หน้าจอเลือก DBMS และ Test Set ที่จะทดสอบ.....73
5.23	ไดอะล็อกแสดงว่าทดสอบเรียบร้อยแล้ว.....73
5.24	หน้าจอเมื่อทดสอบเรียบร้อยแล้ว.....74
5.25	หน้าจอแสดงผลลัพธ์ที่เคยทดสอบในโครงการ.....75
5.26	เพิ่มอธิบายเพิ่มอินพุตแบบ DTD.....76
5.27	เพิ่มอธิบายเพิ่มอินพุตแบบ XML Schemas.....76
5.28	ตัวอย่างเพิ่มอินพุต.....77
5.29	เพิ่มอธิบายเพิ่มเทสต์เซตแบบ DTD.....77
5.30	เพิ่มอธิบายเพิ่มเทสต์เซตแบบ XML Schemas.....78
5.31	ตัวอย่างเพิ่มเทสต์เซต.....78
5.32	เพิ่มอธิบายเพิ่มเอาต์พุตแบบ DTD.....79
5.33	เพิ่มอธิบายเพิ่มเอาต์พุตแบบ XML Schemas.....80
5.34	ตัวอย่างเพิ่มเอาต์พุต.....82
5.35	โครงสร้างเพิ่ม.....82

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ฐานข้อมูลมีบทบาทในการทำงานของเกือบทุกองค์กรสมัยใหม่ โดยระบบการจัดการฐานข้อมูลทำให้องค์กรสามารถเข้าถึงข้อมูลจำนวนมากได้อย่างมีประสิทธิภาพ รวมทั้งการควบคุมความถูกต้องให้กับข้อมูล และลดความจำเป็นในการเข้าใจรายละเอียดปลีกย่อยในการจัดเก็บ และค้นคืน ปัจจัยสำคัญของระบบการจัดการฐานข้อมูลจะต้องรับประกันอัลกอริทึม และ โครงสร้างฐานข้อมูล โดยลักษณะความถูกต้องของระบบฐานข้อมูลมีดังต่อไปนี้

1. โปรแกรมประยุกต์ที่ติดต่อฐานข้อมูลกระทำตามที่กำหนดหรือไม่
2. โครงสร้างฐานข้อมูลที่ถูกต้องสะท้อนถึงข้อมูลจริงที่เป็นแบบจำลองขององค์กรหรือไม่
3. ข้อมูลในฐานข้อมูลถูกต้องหรือไม่
4. ระบบปกป้องความปลอดภัยและความเป็นส่วนตัวอย่างเหมาะสมหรือไม่
5. ระบบการจัดการข้อมูลกระทำการเพิ่ม ลบ และปรับปรุงของข้อมูลทั้งหมดอย่างถูกต้องหรือไม่ (Chays et.al. 2000)

ซึ่งในการพัฒนาระบบงานนี้เราจะเน้นไปที่การทำงานของระบบการจัดการข้อมูลว่ามีความถูกต้องตามลักษณะความถูกต้อง(ข้อ 5)ของระบบฐานข้อมูล เนื่องจากระบบทดสอบไคร์เวอร์ถูกใช้ในการสร้างชุดทดสอบภาษาการจัดการข้อมูล สำหรับทดสอบ DBMS ต่างๆ เท่านั้น ดังนั้นเราจะตั้งสมมติฐานลักษณะความถูกต้องของระบบฐานข้อมูล (ข้อ 1-4) ว่ามีความถูกต้องโดยเราจะใช้ทฤษฎีในการทดสอบซอฟต์แวร์ (Software Testing) นั่นคือการสร้าง เทสต์ไคร์เวอร์ ซึ่งเป็นซอฟต์แวร์โมดูล หรือ โปรแกรมประยุกต์ที่ใช้ทำให้เกิด ชุดทดสอบที่เป็นอินพุตทดสอบ(ข้อมูล) โดยเทสต์ไคร์เวอร์จะทำการป้อนข้อมูลอินพุตจำนวนมากแทนที่ผู้ทดสอบจะทำการป้อนข้อมูลเข้าสู่ระบบที่ทดสอบเอง แล้วทำการเปรียบเทียบผลลัพธ์ที่เกิดขึ้นจริงกับผลลัพธ์ที่ควรเป็น นอกจากนี้ยังใช้ทฤษฎีในการแลกเปลี่ยนเอกสารด้วยมาตรฐาน XML (eXtensible Markup Language) โดยเราจะนำทฤษฎีเหล่านี้มาพัฒนาระบบงานสำหรับการทดสอบภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language -DML) โดยผ่านส่วนประสานกราฟิกกับผู้ใช้ (GUI) ซึ่งช่วยให้ผู้ทดสอบสามารถสร้างเพิ่มอินพุตสำหรับการทดสอบดีเอ็มแอลของฐานข้อมูลได้ง่ายขึ้น จัดเก็บเพิ่มเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อในการวิเคราะห์ภายหลัง และกำหนดให้เพิ่มอินพุต และเอาต์พุตมีรูปแบบแฟ้มเป็น XML เพื่อเป็นมาตรฐานในการแลกเปลี่ยนชุดทดสอบ

1.2 ขอบเขตของโครงการพัฒนาระบบงาน

ระบบทดสอบโค้ดเวิร์กสำหรับทดสอบดีเอ็มแอลของฐานข้อมูลในปัจจุบันยังมีรายละเอียดคุณสมบัติต่างๆ ไม่ครบถ้วนสมบูรณ์ ดังนั้นเรามีการวางแผนที่จะปรับปรุงระบบนี้ดังต่อไปนี้

- ปรับปรุง GUI ให้ใช้งานง่าย และเป็นมิตรกับผู้ใช้มากขึ้น
- ขยายให้ใช้ได้กับ DBMSs ที่จัดการฐานข้อมูลที่เราทดสอบให้มากขึ้น ได้แก่ ODBC, MySQL, ORACLE DATABASE และ Microsoft SQL Server 2000
- สามารถสร้าง Test Driver มากกว่าหนึ่งชุดในหนึ่งชุดการทดสอบ เพื่อสะดวกในการเปรียบเทียบผลการทดสอบ
- สามารถนำชุดทดสอบที่เคยสร้างนำกลับมาใช้ได้
- จัดเก็บแฟ้มอินพุต และเอาต์พุตในรูปแบบ XML ที่รองรับทั้ง DTD และ XML Schemas

1.3 ขั้นตอนการดำเนินงาน

ในโครงการนี้ได้แบ่งขั้นตอนการศึกษาและการพัฒนาโปรแกรมไว้ดังต่อไปนี้

- ขั้นตอนที่1 ศึกษาทฤษฎีการทดสอบซอฟต์แวร์ โดยเฉพาะเทสต์ไคร์เวอร์ ทฤษฎีการแลกเปลี่ยนเอกสาร XML
- ขั้นตอนที่2 ศึกษาภาษาการโปรแกรม Java และเครื่องมือ IDE ที่ใช้ในการพัฒนาโปรแกรม
- ขั้นตอนที่3 วิเคราะห์ความต้องการของระบบ และออกแบบ GUI
- ขั้นตอนที่4 ออกแบบสถาปัตยกรรม และการทำงานของระบบตามความต้องการในขั้นตอนที่3
- ขั้นตอนที่5 ทำการพัฒนาตัวโปรแกรม และทดสอบโปรแกรมแต่ละส่วน ให้ได้ตามที่ออกแบบไว้
- ขั้นตอนที่6 ทำการทดสอบและปรับปรุงโดยรวม ให้เหมาะสมกับสภาพแวดล้อมที่จะเกิดขึ้น

1.4 ข้อจำกัดของระบบ

- ผลลัพธ์ที่เกิดจากการทำนาย อาจได้ผลลัพธ์ที่ผิดพลาดจากความเป็นจริงได้ สำหรับคำสั่งภาษาสำหรับการจัดการข้อมูลที่ซับซ้อน เนื่องจากรูปแบบการเขียนสามารถเขียนได้หลายรูปแบบ ซึ่งทำให้เกิดเหตุการณ์ข้อผิดพลาดได้หลายรูปแบบ ดังนั้นจึงต้องใช้เวลาในการพัฒนาอย่างต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ระบบทศตวรรษที่ 21 วิศวกรรมสำหรับทดสอบดีเอ็มแอล และทฤษฎีที่เกี่ยวข้อง

2.1 ทฤษฎีต่างๆ ที่เกี่ยวข้อง

2.1.1 ฐานข้อมูลเชิงสัมพันธ์ และภาษา DML ใน SQL

ฐานข้อมูลเชิงสัมพันธ์เป็นการใช้แบบจำลองข้อมูลเชิงสัมพันธ์ที่ซึ่งมองข้อมูลเป็นกลุ่มของความสัมพันธ์ ซึ่งความสัมพันธ์คือตาราง โดยในแต่ละตารางประกอบด้วยแถว (Row) ที่แทนข้อมูลเกี่ยวกับหนึ่งรายการ และแต่ละสดมภ์ (Column) แทนคุณลักษณะต่างๆ ของข้อมูล โดยโครงสร้างความสัมพันธ์เป็น $R(A_1, \dots, A_n)$ เป็นชื่อความสัมพันธ์ (ชื่อตาราง) ตามด้วยชื่อของคุณลักษณะ (ชื่อสดมภ์) แต่ละสดมภ์จะมีชื่อ A_i และ โดเมน (ชนิด) $dom(A_i)$ ซึ่งโครงสร้างความสัมพันธ์อธิบายถึงโครงสร้างของข้อมูล ในขณะที่ความสัมพันธ์อธิบายถึงสถานะของข้อมูลในขณะเวลานั้น โดยโครงสร้างความสัมพันธ์จะถูกกำหนดในช่วงเวลาออกแบบฐานข้อมูล และมีการเปลี่ยนแปลงน้อยที่สุด อย่างไรก็ตามสถานะความสัมพันธ์ถูกแก้ไขอย่างคงที่ในการส่งผลต่อการเปลี่ยนแปลงในข้อมูลจริงที่เป็นแบบจำลอง แต่ในบางครั้งสถานะความสัมพันธ์อาจจะเปลี่ยนแปลงเมื่อมีการเข้ามาใหม่ในฐานข้อมูล หรือเมื่อมีการลบ หรือแก้ไข โครงสร้างฐานข้อมูลเชิงสัมพันธ์เป็นเซตของโครงสร้างความสัมพันธ์ด้วยกันกับเซตของข้อบ่งชี้บังคับความถูกต้อง และสถานะฐานข้อมูลเชิงสัมพันธ์เป็นเซตของสถานะความสัมพันธ์ของการให้ความสัมพันธ์ เช่น ข้อบ่งชี้บังคับความถูกต้อง ซึ่งมีชนิดต่างๆ ของข้อบ่งชี้บังคับ ดังนี้

1. กำหนดข้อบ่งชี้บังคับโดเมน
2. กำหนดข้อบ่งชี้บังคับที่มีเอกลักษณ์
3. คุณลักษณะ A สามารถเป็นค่าพิเศษ (ว่างเปล่า) ที่เรียกว่า NULL โดยการกำหนดข้อบ่งชี้บังคับ not-NULL คือให้คุณลักษณะไม่สามารถให้มีค่าเป็น NULL ได้
4. ข้อบ่งชี้บังคับความถูกต้องอ้างอิง (Referential integrity)
5. ข้อบ่งชี้บังคับความถูกต้องซีเมนติก เช่น เงินเดือนหัวหน้าแผนกจะต้องสูงกว่าสมาชิกอื่นๆ ในแผนก (Chays et.al. 2000)

SQL (Structured Query Language) เป็นภาษาที่ใช้สำหรับการจัดการ และดึงข้อมูลจากฐานข้อมูล ซึ่งฐานข้อมูลที่จะใช้งานภาษา SQL ได้ต้องเป็นฐานข้อมูลเชิงสัมพันธ์ ซึ่ง SQL ไม่เพียงเป็นแต่เป็นภาษาที่ใช้สำหรับการคิวรีเท่านั้นแต่นำมาใช้ในการควบคุมการทำงานของ DBMS ได้ด้วย

เช่น การใช้คำสั่ง SQL เพื่อสร้างตารางในฐานข้อมูล หรือแม้แต่การอนุญาตให้ผู้ใช้แต่ละคนมีสิทธิในการใช้ฐานข้อมูลเป็นต้น โดยสามารถแบ่งกลุ่มภาษา SQL ได้ 3 กลุ่มดังนี้

1. ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language-DML) เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับการเข้าถึงข้อมูลและการแก้ไขข้อมูล เช่น SELECT ใช้เพื่อคิวรีหาข้อมูล, INSERT ใช้เพื่อเพิ่มข้อมูล, DELETE ใช้เพื่อลบ และ UPDATE ใช้เพื่ออัปเดตลงในตารางข้อมูล

2. ภาษาสำหรับการนิยามข้อมูล (Data Definition Language-DDL) เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้กำหนดขอบเขตฐานข้อมูล เช่น CREATE ใช้เพื่อเพิ่มตารางลงในฐานข้อมูล, DROP ใช้เพื่อลบตารางออก และ ALTER ใช้เพื่อเปลี่ยนแปลงโครงสร้างตารางในฐานข้อมูล

3. ภาษาการควบคุมข้อมูล (Data Control Language-DCL) เป็นกลุ่มคำสั่งในภาษา SQL ที่ควบคุมความปลอดภัยข้อมูลเช่น GRANT ใช้เพื่อให้สิทธิในการใช้ฐานข้อมูล และ REVOKE ใช้เพื่อยกเลิกสิทธิในการใช้ฐานข้อมูล (ยุทธนา ติลาศวัฒน์กุล และอมรพันธุ์ คำอรรด. 2545)

เนื่องจากในบทความนี้เราสนใจการทดสอบดีเอ็มแอลของฐานข้อมูล จึงนำเสนอรูปแบบเฉพาะภาษาการจัดการฐานเท่านั้น

```
SELECT [DISTINCT] <column(s)>
FROM <table(s)>
[WHERE <condition>]
[GROUP BY <column(s)>] [HAVING <condition>]
[ORDER BY <column(s)> [ASC|DECS] ]
```

```
INSERT INTO <table_name>
VALUES(v_column1, v_column2, ...) หรือ
INSERT INTO <table_name> (col_name1, col_name2)
VALUES(v_column1, v_column2, ...)
```

```
DELETE FROM <table_name>
WHERE <condition>
```

```
UPDATE <table_name>
SET <column_name1> = <new_value1> ,
```

<column_name2> = <new_value2>

WHERE <condition>

แต่เนื่องจากระบบทดสอบไคร์เวอร์กยูจำเป็นต้องมีการสร้าง และลบตารางจึงจำเป็นต้องใช้ภาษาการกำหนดข้อมูล (ดีดีแอล) เข้ามาเกี่ยวข้องด้วยคือ

CREATE TABLE <table name>

<column name1><data type>[CONSTRAINT<constraint name>

<integrity constraint>]

... <column nameN><data type>[CONSTRAINT<constraint name>

<integrity constraint>]

DROP TABLE <table name>

2.1.2 การแลกเปลี่ยนเอกสารระหว่างองค์กร-XML

มาตรฐาน XML (eXtensible Markup Language) เป็นภาษาที่ใช้ในการสร้างเอกสารที่มีการกำหนดความหมายของข้อมูลได้รับการออกแบบให้ใช้งานได้ดีบนเว็บ โดยเฉพาะการแลกเปลี่ยนข้อมูลระหว่างโปรแกรมประยุกต์ แต่ก็สามารถทำงานได้ดีแม้สภาพแวดล้อมการทำงานที่ไม่ใช่เว็บ ซึ่งจุดประสงค์หลักของ XML มีดังนี้

- XML ควรสนับสนุน โปรแกรมประยุกต์ได้หลากหลาย ทั้งในรูปแบบเป็นเครื่องมือสำหรับการสร้าง การแสดงผลข้อมูล และแม้แต่การจัดการเก็บข้อมูล เพื่อให้โปรแกรมประยุกต์ต่างๆ นำ XML มาใช้งาน
- ความสามารถเสริมที่มีอยู่ใน XML ต้องมีน้อยที่สุด หรือไม่ควรมีเลย เนื่องจากจะทำให้ทั้งเอกสารและโปรแกรมประมวลผลมีความเข้ากันได้ยาก เช่น โปรแกรมประมวลผลหนึ่งได้รับการออกแบบมาให้สามารถอ่านและประมวลผลเอกสาร XML ได้ด้วยความสามารถเสริมตัวหนึ่ง แต่เอกสารที่สร้างด้วยความสามารถเสริมอีกตัวหนึ่งอาจไม่สามารถทำงานได้ถูกต้องด้วยโปรแกรมประมวลผลนี้
- การออกแบบ XML จะต้องเป็นรูปแบบที่สั้นและกระทัดรัด

การสร้างเอกสาร XML จะต้องคำนึงถึงการมีลักษณะ “Well Formed” คือเอกสารตัวนั้นต้องทำตามหลักของภาษา XML ทุกประการ เช่น เมื่อมี tag เปิด ต้องมี tag ปิดเสมอ และจะต้องตามลำดับ ตัวอักษรเล็กใหญ่มีความแตกต่างกันเป็นต้น แต่การ Well Formed อย่างเดียวยังไม่พอ เอกสารเนื่องจากเอกสารที่สร้างขึ้นนี้มีความจำเป็นต้องมีการแลกเปลี่ยนไปยังองค์กรอื่น จึงต้องมีการไม่ว่ากรณีใดทั้งสั้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดข้อตกลงในหลักภาษาให้สอดคล้องเข้าใจความหมายตรงกัน ซึ่งขึ้นกับว่าจะเลือกใช้ DTD (Data Type Definition) หรือ XML Schemas

DTD (Data Type Definition) สามารถกำหนดได้เอง หรืออาจจะใช้ DTD ที่องค์กรอื่นได้มีการกำหนดไว้แล้วก็ได้ซึ่งจะเป็นการกำหนดว่า element หนึ่งๆ นั้นมีส่วนประกอบอะไรบ้าง ตัวอย่างในรูปที่ 2.1

```

1 <?xml version = "1.0">
2 <!DOCTYPE memo [
3 <!ELEMENT memo (to,from,subject?,body)>
4 <!ELEMENT to (#PCDATA)>
5 <!ELEMENT from (#PCDATA)>
6 <!ELEMENT subject (#PCDATA)>
7 <!ELEMENT body (#PCDATA)>
8 ]>

```

รูปที่ 2.1 Data Type Definition

จากรูปที่ 2.1 บรรทัดที่ 2 DOCTYPE เป็นตัวย่อของคำว่า Document Type ในบรรทัดที่ 3 เป็นการนิยามว่า element memo นี้ประกอบด้วย element ย่อย 4 ตัวคือ to, from, subject, body ซึ่ง element subject จะปรากฏในเอกสาร XML 1 ครั้งหรือไม่ปรากฏเลย ส่วนบรรทัดที่ 4-7 เป็นการนิยามว่า element นั้นๆเป็นชนิด PCDATA

ซึ่ง DTD ตามรูปที่ 2.1 นั้นสามารถรวมอยู่ในเอกสาร XML หรือแยกจากเอกสาร XML แล้วใช้อ้างอิงก็ได้ แต่การแยกจากเอกสาร XML เป็นวิธีที่สะดวกกว่าเนื่องจากเมื่อมีการแก้ไข DTD ก็จะไม่แก้ไขแค่เพียงที่เดียว

วิธีการในการจัดการรูปแบบเอกสาร XML ไม่ได้มีเพียง DTD แต่มีวิธีการจัดการรูปแบบอีกหนึ่ง คือ XML Schemas ดังตัวอย่างในรูปที่ 2.2

```

1 <?xml version="1.0">
2 <Schema name="memo" xmlns="urn:Schemas-microsoft-
com:xml-data"xmlns:dt="urn:schemas-microsoft-
com:datatypes">
3 <ElementType name="to" content="textOnly" />
4 <ElementType name="from" content="textOnly" />
5 <ElementType name="subject"content="textOnly" />
6 <ElementType name="body" content="textOnly" />
7 <ElementType name="memo" content="textOnly" />
8<element type="to"minOccurs="1" maxOccurs="1"/>
9 <element type= "from" minOccurs= "1" maxOccurs= "1" />
10 <element type= "subject"minOccurs= "0"maxOccurs= "1" />
11 <element type= "body" minOccurs= "1" maxOccurs= "1" />
12 </ElementType>

```

รูปที่ 2.2 XML Schema

XML Schema ตามรูปที่ 2.2 นั้นมีความหมายเหมือนกับ DTD ซึ่งในการพัฒนาระบบนี้เรา จะใช้ทั้งสองรูปแบบเนื่องจากระบบเดิมเป็นแบบ DTD ซึ่งในปัจจุบันไม่เป็นที่นิยมเท่า XML Schema ซึ่งจะทำให้ผู้ทดสอบดีเอ็มแอลของฐานข้อมูลสามารถเลือกได้ว่าจะเก็บชุดทดสอบ (input) และผลทดสอบ (output) แบบใด (ชนิศา สุจริตรัถยตระกูล, 2544)

2.2 ความหมายของ เทสต์ไคร์เวอร์, Oracle และ การเชื่อมต่อฐานข้อมูล

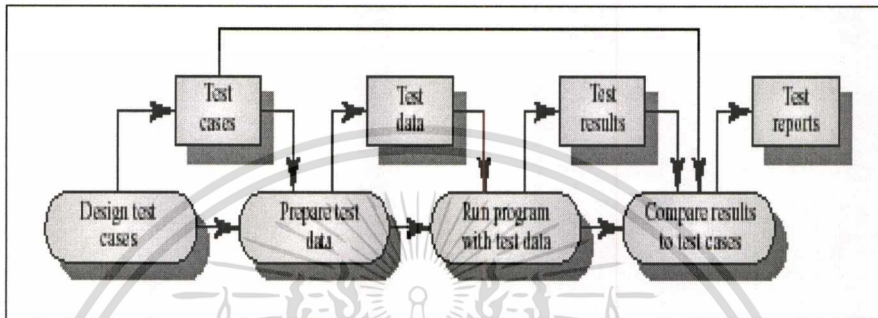
2.2.1 ความหมายของเทสต์ไคร์เวอร์

การทดสอบหมายถึงการตรวจสอบซอฟต์แวร์ โดยการทดสอบโปรแกรมจะทำการเลือก อินพุต และทำการตรวจสอบหาข้อผิดพลาดของซอฟต์แวร์ โดยมีวัตถุประสงค์หลักของการ ทดสอบซอฟต์แวร์ คือการพัฒนาอย่างมีระบบในการเลือกกลุ่มทดสอบให้มีขนาดเล็กเพียงพอ เหมาะสมในการทดสอบให้ครอบคลุมสถานการณ์ต่างๆ ที่มีความเป็นไปได้ (เทคนิคในการเลือก เทสต์เคสน้อยที่สุดให้เจอข้อผิดพลาดมากที่สุด เช่น Domain Testing และ Category-partition Testing เป็นต้น) ซึ่งจะได้ผลลัพธ์เป็นแฮตพุด โดยกลุ่มทดสอบคือ อินพุตหนึ่งชุดที่ประกอบด้วย

พารามิเตอร์สองตัวคือ (อินพุต และสิ่งแวดล้อม) และเอาต์พุตที่คาดหวังจะได้ สามารถเขียนกลุ่มทดสอบเป็นเซตดังนี้

Test cases = { input, expected output}

ในการทดสอบซอฟต์แวร์มีกระบวนการทดสอบข้อบกพร่องสามารถแสดงได้ดังรูปที่ 2.3

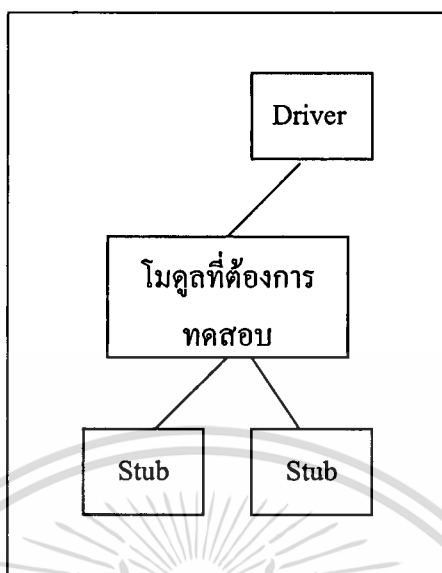


รูปที่ 2.3 กระบวนการทดสอบข้อบกพร่อง

โดยปกติผู้ทดสอบจะสร้างข้อมูลที่ใช้ในการทดสอบขึ้นมาเอง เรียกว่า Test Data หรือ Test Set โดยข้อมูลที่สร้างขึ้นจะต้องมีปริมาณ และมีความหลากหลายมากพอที่จะทำให้สรุปได้ว่าซอฟต์แวร์ ทำงานถูกต้องในทุกสถานการณ์ที่จะเกิดขึ้นได้ ในการเตรียม Test Data เป็นชุดขึ้นมาทดสอบ เพื่อตรวจสอบกระบวนการตั้งแต่อินพุต และการตอบสนองในระหว่างกระบวนการแต่ละขั้นตอน จนถึงการผลิตเอาต์พุต ว่าข้อมูลเหล่านั้นประกอบกันขึ้นแล้วทำให้เหตุการณ์หรือผลลัพธ์เป็นไปตามที่คาดหวัง หรือไม่ Test Data ลักษณะนี้ เรียกว่า Test case ที่ได้อธิบายแล้วในข้างต้น

จากการที่ซอฟต์แวร์มักถูกออกแบบให้มีการแยกโมดูล ซึ่งโมดูล ต่างๆในระบบมักมีความเกี่ยวข้องเชื่อมโยงกัน ในการทดสอบขั้นทดสอบโมดูล จะเน้นการทำงานของโมดูลใดๆ ครั้งละโมดูลเดียว จึงต้องให้ความสนใจในการทำงานของโมดูลอื่นๆ ไว้ ในกรณีที่โมดูลที่ควบคุมอยู่ด้านบน มีการเรียก โมดูลล่าง หากต้องการทดสอบโมดูลล่างนั้น แทนที่การทดสอบจะต้องเรียกจากโมดูลจริงๆ มักทำการสร้างโมดูลบนเป็น โมดูลจำลองเพื่อเป็นตัวขับโมดูลล่าง เรียกโมดูลขับนี้ว่า driver ส่วนกรณีโมดูลที่ต้องการทดสอบ มีการเรียกโมดูลล่าง ก็ต้องสร้างโมดูลล่างเป็นโมดูลจำลอง แทนการเรียกโมดูลล่างตัวจริง เรียกโมดูลจำลองที่สร้างขึ้นเพื่อใช้ในการทดสอบ โมดูลบนนี้เรียกว่า stub ดังแสดงดังรูปที่ 2.4

เทสต์ไดร์เวอร์ เป็นซอฟต์แวร์โมดูล หรือโปรแกรมประยุกต์ที่ผู้ทดสอบ สร้างขึ้นเพื่อทดสอบ โมดูล ระดับ Unit Testing และ Integration Testing โดย Test Driver จะเป็นตัว generate input แล้วทำการเปรียบเทียบผลลัพธ์ที่ได้ผลส่งกลับมาเปรียบกับ Expected Output (ผลลัพธ์ที่ควรเป็น) ที่เกิดจาก Oracle (Patton, 2001)



รูปที่ 2.4 TestDriver และ Stub

2.2.2 ความหมายของ Oracle

Oracle เป็นเซตของ Expected Output หรือเป็นวิธีการผลิตผลลัพธ์การทำงานเปรียบเทียบกับผลลัพธ์จริงของซอฟต์แวร์ภายใต้การทดสอบ (Patton, 2001)

2.2.3 การเชื่อมต่อฐานข้อมูล

ในระบบการทดสอบไดร์เวอร์เป็นการใช้พื้นฐานบน JDBC (Java Database Connection) ในการเชื่อมต่อฐานข้อมูล โดย JDBC แบ่งออกเป็นสองส่วนคือ JDBC api เป็นชุดของคลาส และ Interface เพื่อให้โปรแกรมภาษาจาวาใช้ติดต่อกับระบบฐานข้อมูลใด ๆ อีกส่วนหนึ่ง คือ ข้อกำหนดของ JDBC Drivers เพื่อให้บริษัทผู้ผลิตระบบฐานข้อมูลนำไปสร้าง drivers ที่ติดต่อกับระบบฐานข้อมูลของเขา ทำให้โปรแกรมไม่ผูกติดกับระบบฐานข้อมูลที่ใช้ รวมทั้งไม่จำเป็นต้องทราบรายละเอียดวิธีติดต่อกับระบบฐานข้อมูลนั้น โดยกลไกพื้นฐานของ JDBC คือ เริ่มต้นติดต่อกับระบบฐานข้อมูล แล้วส่งคำสั่ง SQL ไประบบฐานข้อมูล สุดท้ายรับผลลัพธ์จากระบบฐานข้อมูลเข้ามาจัดการ

เมื่อโปรแกรมภาษาจาวา ต้องการติดต่อกับระบบฐานข้อมูลใด ก็จะต้องมีไดร์เวอร์สำหรับระบบฐานข้อมูลนั้น เพื่อทำหน้าที่เชื่อมต่อระหว่างโปรแกรมกับฐานข้อมูล แต่ระบบฐานข้อมูลมีความแตกต่างกันไปในแต่ละยี่ห้อ ดังนั้นไดร์เวอร์ที่สร้างขึ้นสำหรับต่อกับ JDBC จึงถูกสร้างขึ้นด้วยวิธีที่แตกต่างกันไป ซึ่งส่งผลถึงสภาพแวดล้อมในการทำงานด้วยเราสามารถแบ่งไดร์เวอร์ตามวิธีที่ถูกสร้างขึ้นมาได้ 4 ประเภท ดังนี้

Type 1. JDBC-ODBC bridge driver เป็นประเภทที่ถูกสร้างขึ้นโดยใช้ native code เพื่อทำหน้าที่เป็นตัวกลางระหว่างโปรแกรมกับ ODBC drivers ข้อดีของการใช้ bridge drivers คือ ODBC drivers สามารถต่อกับระบบฐานข้อมูลได้หลายยี่ห้อ โดยเฉพาะที่ทำงานได้ใน Microsoft Windows แต่ข้อเสียคือ มีการแปลคำสั่งเพิ่มขึ้นอีกระดับหนึ่งทำให้ทำงานช้าเมื่อเปรียบเทียบกับไดรเวอร์ประเภทอื่นๆ และที่เครื่องด้านไคร์เอนต์ ต้องมีการติดตั้งโปรแกรมของ ODBC รวมทั้งต้องทำ configuration ให้มองเห็นฐานข้อมูลที่จะใช้ ตัวอย่าง ไดรเวอร์ประเภทนี้คือ JDBC-ODBC driver ซึ่งอยู่ใน package “un.jdbc.odbc.JdbcOdbcDriver” ที่มากับ JDK นับตั้งแต่รุ่น 1.1 เป็นต้นมา ที่จริงแล้ว จุดประสงค์ของการสร้าง JDBC-ODBC driver คือเพื่อให้ผู้เขียนโปรแกรมทดลองเปลี่ยนจาก ODBC เป็น JDBC แต่ในปัจจุบันระบบฐานข้อมูลเกือบทุกยี่ห้อที่มีไดรเวอร์ประเภทอื่นๆ ซึ่งทำงานได้เร็วกว่า และเชื่อถือได้มากกว่า ดังนั้นจึงแนะนำว่า ควรใช้ JDBC-ODBC driver ในการทดลองเรียนรู้เท่านั้น ส่วนในการใช้งานจริงควรใช้ไดรเวอร์ประเภทอื่นๆ

Type 2. Native API party Java driver เป็นประเภทที่ถูกสร้างขึ้นโดยใช้ native code แล้วนำมาสร้างเป็นคลาสของภาษาจาวาในการใช้งานจะต้องติดตั้งทั้ง native code และคลาสของจาวาที่เครื่องด้านไคร์เอนต์ทุกตัวที่ใช้งาน เช่นเดียวกับแบบแรก แต่เนื่องจากไม่ต้องผ่านไดรเวอร์ตัวกลางอย่าง ODBC และมีบางส่วนที่ทำงานด้วย native code จึงทำงานได้เร็วกว่าแบบอื่นๆ ยกเว้นแบบที่ 4 ตัวอย่าง ไดรเวอร์ประเภทนี้ได้แก่ Oracle Call Interface (OCI) driver และยังมีของบริษัท Sybase, Infomix และ IBM ด้วย

Type 3. JDBC-Net pure Java driver เป็นประเภทที่สร้างด้วยภาษาจาวาทั้งหมด โดยไดรเวอร์จะแปลคำสั่งจาก JDBC เป็นคำสั่งที่ใช้ส่งผ่านเครือข่ายไปที่เซิร์ฟเวอร์ของระบบฐานข้อมูล จากนั้นเซิร์ฟเวอร์จะแปลคำสั่งที่ได้รับเป็นคำสั่งสำหรับเชื่อมต่อกับฐานข้อมูล ข้อดีคือ ไดรเวอร์ชนิดนี้สามารถติดต่อกับระบบฐานข้อมูลที่ต่างยี่ห้อได้ หากเซิร์ฟเวอร์ของระบบฐานข้อมูลนั้นใช้โปรโตคอลเครือข่ายเดียวกัน และเนื่องจากไดรเวอร์ประเภทนี้ถูกสร้างขึ้นด้วยภาษาจาวาทั้งหมด จึงสามารถถูกดาวน์โหลดไปที่เครื่องด้านไคร์เอนต์ โดยอัตโนมัติ แต่ก็มีข้อเสียคือเพิ่มความยุ่งยากในการจัดการด้านความปลอดภัย ตัวอย่างของไดรเวอร์ประเภทนี้ได้แก่ Web logic ของ Tengah

Type. 4 Native protocol pure Java driver เป็นประเภทที่สร้างด้วยภาษาจาวาทั้งหมด เหมือนกับแบบที่ 3 แต่ไดรเวอร์จะทำหน้าที่ติดต่อกับระบบฐานข้อมูลโดยตรง ทำให้มีประสิทธิภาพการทำงานมาก เรามักเรียกไดรเวอร์ ประเภทนี้ว่า thin drivers ระบบฐานข้อมูลที่มีไดรเวอร์ประเภทนี้ได้แก่ Oracle, Sybase, Infomix, IBM, DB2, Inprise Interbase และ Microsoft SQL Server

ข้อแนะนำในการเลือกใช้งานไดรเวอร์ให้เหมาะสมกับงาน มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้เฉพาะในพิธีกรรทำเท่านั้น เมื่อผู้ใดเห็นนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

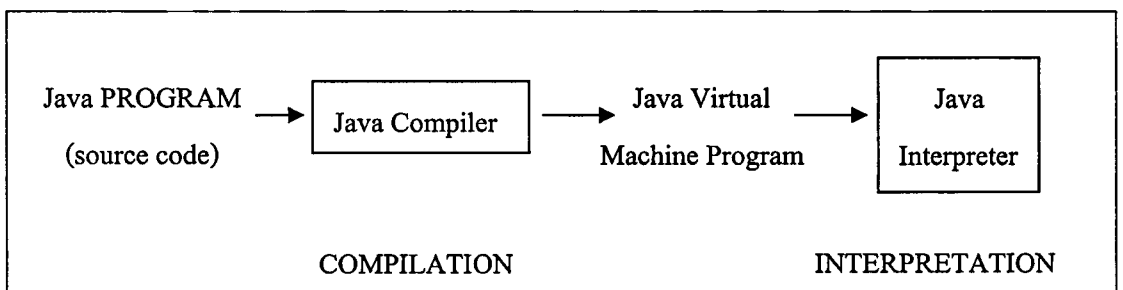
- ถ้าระบบของเราใช้ระบบฐานข้อมูลหนึ่งอยู่แล้ว เราก็ต้องใช้ไคร์เวอร์ที่มีมากับระบบฐานข้อมูลนั้น ซึ่งโดยปกติเกือบทุกยี่ห้อจะให้ฟรี และอาจมีรุ่นปรับปรุงใหม่ให้ดาวน์โหลดได้เรื่อยๆ หากไม่จำเป็นแล้วไม่ควรใช้ไคร์เวอร์ ที่ผลิตโดยบริษัทที่ไม่ใช่ผู้ผลิตระบบฐานข้อมูลนั้น
- การเลือกประเภทของไคร์เวอร์มีผลเฉพาะในตอนใช้งาน โปรแกรม แต่ในระหว่างพัฒนาโปรแกรมเราจะใช้ไคร์เวอร์ประเภทใดก็ได้ หากในตอนใช้งานเราต้องการความเร็วควรเลือก ไคร์เวอร์ประเภทที่เป็น native API หรือ native protocol หากเราไม่สามารถติดตั้งโปรแกรมที่เครื่องของไคร์เอนต์ ก็ควรเลือกไคร์เวอร์ประเภทที่เป็น pure java เพื่อให้สามารถดาวน์โหลดไปที่เครื่องของไคร์เอนต์ได้ง่าย (วีระศักดิ์ ชิงถาวร. 2546)

2.3 เครื่องมือที่ใช้ในการพัฒนาระบบ

ในหัวข้อนี้จะกล่าวถึงเครื่องมือที่ใช้ในการพัฒนาระบบ ก่อนที่จะเริ่มต้นการพัฒนาโปรแกรมจะต้องทำการติดตั้งเครื่องมือให้เรียบร้อยก่อน ไม่ว่าจะเป็น Java compiler, Java interpreter, ODBC, MySQL, ORACLE DATABASE, Microsoft SQL Server และอื่นๆ โดยจะมีรายละเอียดดังนี้

2.3.1 โปรแกรมภาษาจาวา

ภาษาจาวานั้นเป็นภาษาแบบ OOP (Object-Oriented Programming) ซึ่งจะช่วยให้สามารถสร้างโปรแกรมที่มีความยืดหยุ่นมาก โค้ดที่สร้างแล้วจะเก็บเป็นจุด class (เป็นลักษณะ object) สามารถนำกลับมาใช้ได้ อีก ภาษาจาวานั้นเป็นอิสระต่อแพลตฟอร์ม (Platform-Independent) กล่าวคือใช้ source code ที่เขียนบน Operating System (OS) หนึ่งไปใช้อีก OS หนึ่งได้โดยไม่ต้องเขียนโค้ดใหม่ เพื่อให้โปรแกรมไม่ขึ้นกับระบบ จึงนำแนวคิด Virtual Machine มาใช้ โดย Java นั้นมีคอมไพเลอร์ทำการแปลภาษา (source code) ให้เป็นโปรแกรมของ Java Virtual Machine (JVM) แล้วนำโปรแกรมนั้นมาทำการใน JVM นั้นโดย Java Interpreter ดังรูปที่ 2.5



รูปที่ 2.5 หลักการทำงานของโปรแกรม JAVA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในโครงการวิจัยของคณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่ควรนำข้อมูลไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

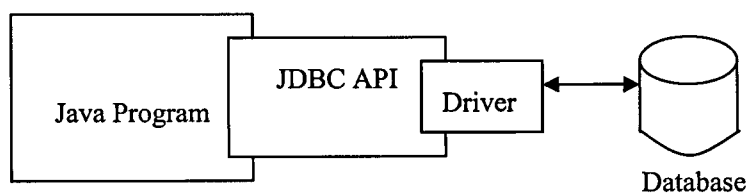
ในการสร้างโปรแกรมภาษา Java ก่อนอื่นจะต้องติดตั้ง Java Developer Kit (JDK) โดยปัจจุบันเป็น version 1.4.x แล้ว ซึ่งสามารถดาวน์โหลดได้แบบฟรีๆ จาก <http://java.sun.com> การติดตั้งก็เพียงแค่แตกไฟล์ที่ดาวน์โหลดมา จากนั้นก็ชี้พาธไปที่ <default directory>\bin เพื่อที่จะเรียกใช้โปรแกรมใน JDK เช่น javac.exe คือ Java Compiler และ java.exe คือ Java Interpreter เป็นต้น วิธีการชี้พาธนั้นจะขึ้นกับระบบปฏิบัติการที่ใช้ สามารถหาได้ในเอกสารของ JDK ที่ <http://java.sun.com> จากนั้นจะต้องทำการกำหนด CLASSPATH ให้ไปใช้ Java Compiler หรือ Java Interpreter ค้นหาคลาสที่เจอ โดยควรให้ Java ทำการค้นหาดังนี้

- เริ่มหาที่ Standard classes ที่อยู่ใน rt.jar และ i18n.jar (rt.jar เป็นไฟล์ที่เก็บคลาสที่เป็นมาตรฐาน ส่วน i18n.jar เก็บคลาสที่เกี่ยวข้องกับ internationalization)
- หากไม่พบแล้วจึงเริ่มหาคลาสที่จัดเป็นประเภท extension classes (อยู่ใน package ที่ขึ้นต้นด้วย javax)
- หากยังไม่พบจึงไปหาคลาสประเภท User defined classes ในไดเรกทอรีปัจจุบันที่ทำงาน

ในการพัฒนาระบบทดสอบไคร์เวอร์กยู เราใช้เครื่องมือภาษาจาวา คือ Borland Jbuilder X Enterprise เนื่องจากเป็นเครื่องมือที่ใช้ในการพัฒนาระบบจากจาวาได้ง่าย และสะดวก

2.3.2 Java Database Connection (JDBC)

วิระศักดิ์ ซึ่งถาวร(2546) JDBC คือตัวเชื่อมต่อระหว่างโปรแกรมกับฐานข้อมูล โดยแบ่งออกเป็นสองส่วนคือ JDBC api เป็นชุดของคลาส และ Interfaces เพื่อให้โปรแกรมภาษา Java ใช้ติดต่อกับระบบฐานข้อมูลใดๆ อีกส่วนหนึ่งคือ ข้อกำหนดของ JDBC drivers เพื่อให้บริษัทผู้ผลิตระบบฐานข้อมูลนำไปสร้าง drivers ที่ติดต่อกับระบบฐานข้อมูลของเขา ทำให้โปรแกรมไม่ผูกติดกับระบบฐานข้อมูลที่ใช้ รวมทั้งไม่จำเป็นต้องทราบรายละเอียดวิธีติดต่อเข้าระบบฐานข้อมูลนั้น การติดต่อระหว่างโปรแกรมกับระบบฐานข้อมูลแสดงได้ดังรูปที่ 2.6



รูปที่ 2.6 การติดต่อระหว่าง โปรแกรม Java กับฐานข้อมูลผ่าน JDBC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ผลิตระบบฐานข้อมูลแต่ละยี่ห้อจะสร้าง drivers สำหรับระบบฐานข้อมูลนั้น โดย drivers จะแปลคำสั่งจาก JDBC เป็นคำสั่งสำหรับเข้าใช้งานระบบฐานข้อมูล ส่วนโปรแกรมภาษา Java จะส่งคำสั่งสำหรับติดต่อกับระบบฐานข้อมูลเป็น SQL ไปที่ JDBC แล้ว JDBC เปลี่ยนคำสั่งนั้นเป็นคำสั่งสำหรับ drivers

โดยปกติ JDBC drivers มักจะถูกสร้างเป็นคลาสของภาษา Java เก็บอยู่ในไฟล์ .jar (หรือบางยี่ห้ออาจจะมีทั้งไฟล์ .jar และไฟล์ .zip) สำหรับ JDBC-ODBC bridge driver จะมีมาใน rt.jar ของ j2sdk แต่ drivers ยี่ห้ออื่นๆ จะมากับโปรแกรมของระบบฐานข้อมูลนั้นๆ ซึ่งเราต้องกำหนด CLASSPATH ไปที่ไฟล์ .jar ของมัน เพื่อจะใช้งานการตั้งชื่อ packages ของ JDBC drivers มักจะเป็นไปตามรูปแบบดังนี้

`companyName.driverType.driverName`

ตัวอย่าง เช่น

- JDBC-ODBC bridge มีชื่อเป็น `sun.jdbc.odbc.JdbcOdbcDriver`
- Oracle drivers มีชื่อเป็น (ก่อน 9i) `oracle.jdbc.driver.OracleDriver`
(9i) `oracle.jdbc.OracleDriver`
- MySQL driver มีชื่อเป็น `com.mysql.jdbc.Driver`
- SQL Server 2000 มีชื่อเป็น `com.microsoft.jdbc.sqlserver.SQLServerDriver`

รายชื่อของ JDBC drivers ของระบบฐานข้อมูลยี่ห้อต่างๆ ถูกเก็บไว้ที่

<http://java.sun.com/products/jdbc/drivers>

ในการพัฒนาระบบทดสอบไคลเอนต์ เราใช้ JDBC driver ในการติดต่อบริการข้อมูลดังนี้

- JDBC-ODBC Bridge เพื่อเป็นสะพานเชื่อมต่อการทำงานจากโปรแกรมภาษาจาวาไปยัง ODBC driver ซึ่งมีอยู่ใน JDK 1.4 อยู่แล้ว

- Oracle JDBC driver มีสองประเภทคือแบบ type 2 และ type 4 ซึ่งทั้งสองชนิดมีทั้งที่ฝั่งไคลเอนต์ และฝั่งเซิร์ฟเวอร์ ดังนี้

- OCI driver เป็น type2 มีบางส่วนเป็นจาวา และบางส่วนเป็น native code ซึ่งเป็น Oracle Call Interface สำหรับติดต่อกับระบบฐานข้อมูลของ Oracle เป็นไคลเอนต์สำหรับฝั่งไคลเอนต์ และต้องมีการติดตั้งโปรแกรมที่เครื่องไคลเอนต์ เพราะมี native code ระบบฐานข้อมูล Oracle เริ่มสนับสนุน OCI driver ตั้งแต่รุ่น 7.3.x มีชื่อเป็น OCI7 ส่วนของรุ่น 8.x และ 9.x มีชื่อเป็น OCI8 และ OCI9 ตามลำดับ

- Thin driver เป็น type4 สำหรับฝั่งไคลเอนต์ แต่ไม่ต้องมีการติดตั้งโปรแกรมที่

เอกสารเครื่องไคลเอนต์ เพราะเป็นจาวาทั้งหมด เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Internal driver เป็น type2 สำหรับฝั่งเซิร์ฟเวอร์ โดยส่วนที่เป็นจาวาจะทำงานใน JServer JVM ของระบบฐานข้อมูล Oracle 8i

- Server-side Thin driver เป็น type4 สำหรับฝั่งเซิร์ฟเวอร์ ปกติเซิร์ฟเวอร์จาวาที่ทำงานฝั่งเซิร์ฟเวอร์สามารถใช้ internal driver ติดต่อกับระบบฐานข้อมูลของ Oracle ได้อยู่แล้ว เราจะใช้ไคร์เวอร์ประเภทนี้ในกรณีที่โปรแกรมจาวานั้นต้องการติดต่อกับแหล่งข้อมูลอื่น

ซึ่งใน Oracle9i มีคลาสอยู่ใน package “oracle.jdbc” อยู่ในไฟล์ชื่อ .zip หรือ .jar ที่ชื่อ classesXXX ในไดเรกทอรี %ORACLE_HOME%/jdbc/lib ของ Oracle9i ซึ่งไฟล์ classes12 สนับสนุน JDK ตั้งแต่รุ่น 12 ขึ้นไป

- MySQL JDBC driver สามารถดาวน์โหลดไฟล์สำหรับ JDBC driver ของ MySQL จาก www.mysql.com สำหรับในการพัฒนาครั้งนี้ใช้ไฟล์ที่ชื่อ mysql-connector-java-3.1.6.jar

- Microsoft SQL Server 2000 สามารถดาวน์โหลดไฟล์สำหรับ JDBC driver ของ SQL Server 2000 จาก www.microsoft.com/sql/downloads สำหรับในการพัฒนาครั้งนี้ใช้ SQL Server 2000 Driver for JDBC Version 2.2 ซึ่ง Microsoft SQL Server 2000 จำเป็นต้อง ลง Service Pack3 (วีระศักดิ์ ชิงถาวร. 2546)

2.3.3 Microsoft ODBC

Microsoft ODBC (Open Database Connectivity) เป็น API ที่ถูกใช้อย่างแพร่หลาย สามารถติดต่อกับระบบฐานข้อมูลได้หลายยี่ห้อ และใช้ได้หลาย platform แต่สาเหตุที่เราไม่เขียนโปรแกรมเพื่อเรียกใช้ ODBC ตรงๆเองจากภาษา Java ก็เพราะ ODBC ถูกเขียนขึ้นด้วยภาษา C มีปัญหาในเรื่องของความปลอดภัยของข้อมูลและความเข้ากันได้ของโปรแกรม ดังนั้นจึงมี JDBC-ODBC bridge เพื่อเป็นสะพานเชื่อมต่อการทำงานจากโปรแกรมภาษา Java ไปยัง ODBC Driver

2.3.4 ORACLE DATABASE

Oracle Database Server จัดว่าเป็น Object-Relational Database Management System (ORDBMS) ซึ่งหมายความว่า สามารถทำงานได้ ทั้งในรูปแบบ Rational และบางคุณสมบัติของ Object Oriented ได้โดย Oracle Database Server นั้น มีความสามารถทำงานร่วมกันกับ ซอฟต์แวร์หลายๆตัวได้ จากผู้ผลิตหลายราย และสนับสนุนมาตรฐานระบบเปิดต่างๆ

Oracle Database Server จะประกอบด้วย 2 ส่วนหลัก คือ

- Oracle Database จะเป็นส่วนของการจัดเก็บข้อมูล
- Oracle Server Instance จะประกอบด้วย Memory และ Background Process

ในการติดต่อใช้งานกับ Oracle Database นั้นเราต้องใช้ภาษา SQL ซึ่งเป็นภาษาที่ใช้การกำหนด และจัดการกับ Database (DDL, DML) การทำงานกับ Database ในแบบ Relational Database

2.3.5 ระบบการจัดการฐานข้อมูล MySQL

ระบบการจัดการฐานข้อมูล MySQL เป็นโปรแกรมที่สามารถนำมาใช้งานได้ฟรี สามารถทำงานได้ทั้งในระบบปฏิบัติการวินโดวส์ และยูนิกซ์ใช้ในการจัดการฐานข้อมูลเชิงสัมพันธ์ ที่ทำงานในรูปแบบไคลเอนต์/เซิร์ฟเวอร์ ในการนำมาใช้งานจะต้องทำการติดตั้งส่วนของโปรแกรมที่เป็นเซิร์ฟเวอร์เพื่อให้ทำงานในเบื้องหลังของระบบปฏิบัติการหรือ เคมอน โพรเซสเพื่อดูแลจัดการฐานข้อมูล หลังจากนั้นจึงทำการสร้างฐานข้อมูล, ตาราง และกำหนดรูปแบบของข้อมูลที่ต้องการจัดเก็บในตารางรวมถึงคีย์หลักตามที่ได้ออกแบบไว้ เพื่อใช้ในการค้นหา เพิ่ม และแก้ไขข้อมูล ในกรณีที่ระบบปฏิบัติการยังไม่ได้ถูกติดตั้งระบบจัดการฐานข้อมูลสามารถทำได้โดยการดาวน์โหลดโปรแกรมที่ตรงกับระบบปฏิบัติการที่ต้องการ ได้ฟรีจากเว็บไซต์ <http://www.mysql.com> เพื่อนำมาติดตั้งตามขั้นตอน

2.3.6 Microsoft SQL Server

Microsoft SQL Server เป็นโปรแกรมจัดการฐานข้อมูลระดับเซิร์ฟเวอร์ที่มีขีดความสามารถในการรองรับข้อมูลขนาดใหญ่ที่ช่วยให้การบริหารจัดการฐานข้อมูลมีประสิทธิภาพ รวดเร็ว สนองตอบต่อความต้องการขององค์กรขนาดใหญ่

Microsoft SQL Server เป็นระบบฐานข้อมูลและโซลูชันการวิเคราะห์ที่สมบูรณ์แบบ ซึ่งนำเสนอความน่าเชื่อถือและประสิทธิภาพ ด้านการขยายระบบที่เว็บและองค์กรธุรกิจต้องการ ด้วยการรองรับ XML และ HTTP ทำให้การเข้าถึงและการแลกเปลี่ยนข้อมูลนั้นทำได้ง่ายขึ้น ในขณะที่ความสามารถในการวิเคราะห์อันทรงพลังยังช่วยเพิ่มคุณค่าของข้อมูล และด้วยความพร้อมของระบบที่ดีขึ้น ทำให้ความสามารถในการทำงานอย่างต่อเนื่อง ของระบบนั้นสูงขึ้น, จัดการกับงานในแต่ละวันโดยอัตโนมัติได้ดีขึ้น รวมทั้งปรับปรุงเครื่องมือด้านการเขียนโปรแกรม และการพัฒนาความเร็วด้านการบริการได้ดีขึ้นเช่นกัน

Microsoft SQL Server ประกอบไปด้วยเวอร์ชันต่าง ๆ ดังต่อไปนี้

- **Personal Edition** เป็นระบบจัดการฐานข้อมูลที่ใช้ได้เฉพาะเครื่องส่วนตัวเท่านั้น ไม่สามารถใช้งานร่วมกับ SQL Server 2000 เวอร์ชันอื่นได้โดยตรง ส่วนใหญ่นิยมนำมาใช้ในการทดสอบการเขียนแอปพลิเคชัน ใช้ติดตั้งบน Windows 9x, ME, XP

- **Standard Edition** เป็นระบบจัดการฐานข้อมูลที่สามารถรองรับเซิร์ฟเวอร์ได้สูงสุดถึง 4 CPU และอ้างหน่วยความจำได้ถึง 2 GB นิยมนำมาใช้งานกับองค์กรขนาดใหญ่ ต้อง

ติดตั้งบน Windows NT 4, 2000 หรือ 2003 Server แต่ถ้าติดตั้งบน Windows 9x, ME, XP จะติดตั้งให้เฉพาะส่วนที่เป็น Client Tools หรือ Connectivity เท่านั้น

- **Enterprise Edition** เป็นระบบจัดการฐานข้อมูลที่สามารถรองรับเซิร์ฟเวอร์ได้สูงสุดถึง 32 CPU และอ้างหน่วยความจำได้ถึง 64 GB นิยมนำมาใช้งานกับองค์กรที่มีขนาดใหญ่และมีสาขามากมาย เพราะมีระบบสนับสนุนการทำงานมากมาย เช่น การทำ Data Mining, Data Warehouse เป็นต้น ต้องติดตั้งบน Windows NT 4, 2000 หรือ 2003 Server เท่านั้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิเคราะห์ และออกแบบระบบงาน

3.1 ความต้องการของระบบงาน

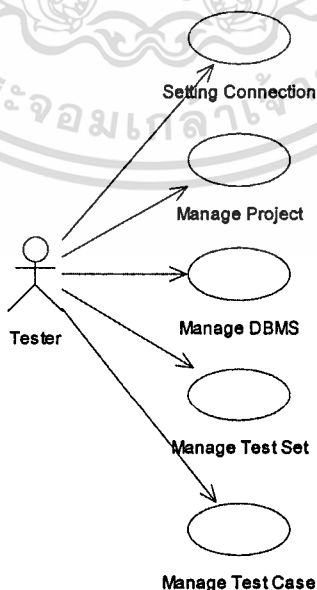
จากความต้องการที่จะพัฒนาระบบงานเพื่อแก้ไขปัญหาต่างๆ ตามที่ได้กล่าวมาข้างต้น จำเป็นต้องนำความต้องการมาทำการวิเคราะห์เพื่อให้ทราบถึงผู้เกี่ยวข้องภายในระบบงาน หน้าที่การทำงานของระบบ และเป็นแนวทางสำหรับการพัฒนาระบบงานต่อไป เครื่องมือที่จะใช้ในขั้นตอนนี้คือ Use Case Diagram ซึ่งในระบบ Test Driver GUI นี้มีผู้ที่เกี่ยวข้องกับระบบคือ

Tester: ผู้ทดสอบ

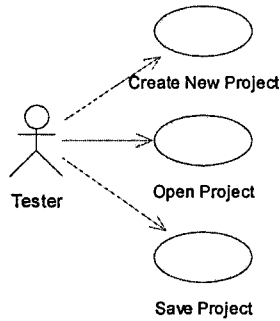
- เป็นผู้สร้าง test case input สำหรับการทดสอบ
- เป็นผู้ทดสอบ และกำหนดให้ result output เก็บแบบ DTD หรือ XML Schema

3.2 Use Case Diagram

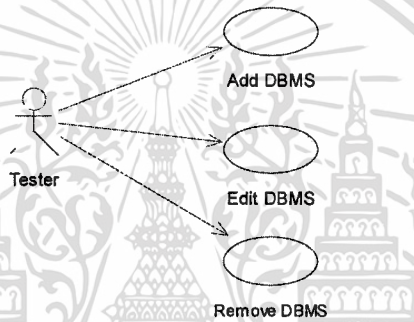
เมื่อนำความต้องการของระบบมาเขียนให้อยู่ในรูปของ Use Case Diagram สามารถแสดงได้ดังนี้



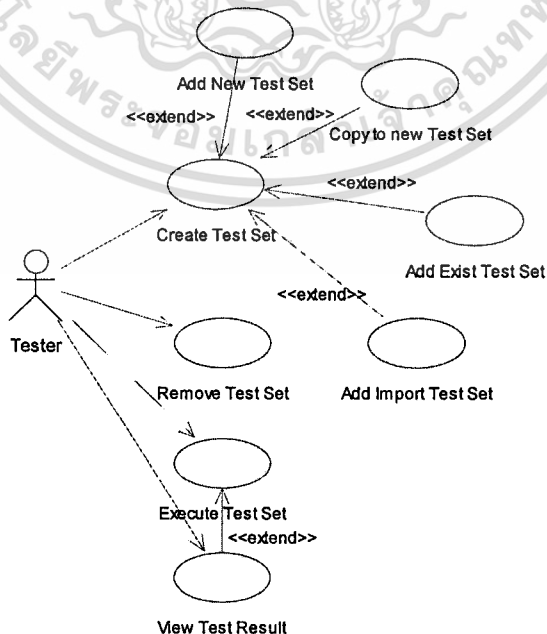
รูปที่ 3.1 Use Case Diagram หลักของระบบทดสอบโค้ดไคร์เวอร์กูย



รูปที่ 3.2 Use Case Diagram ของระบบย่อย Manage Project

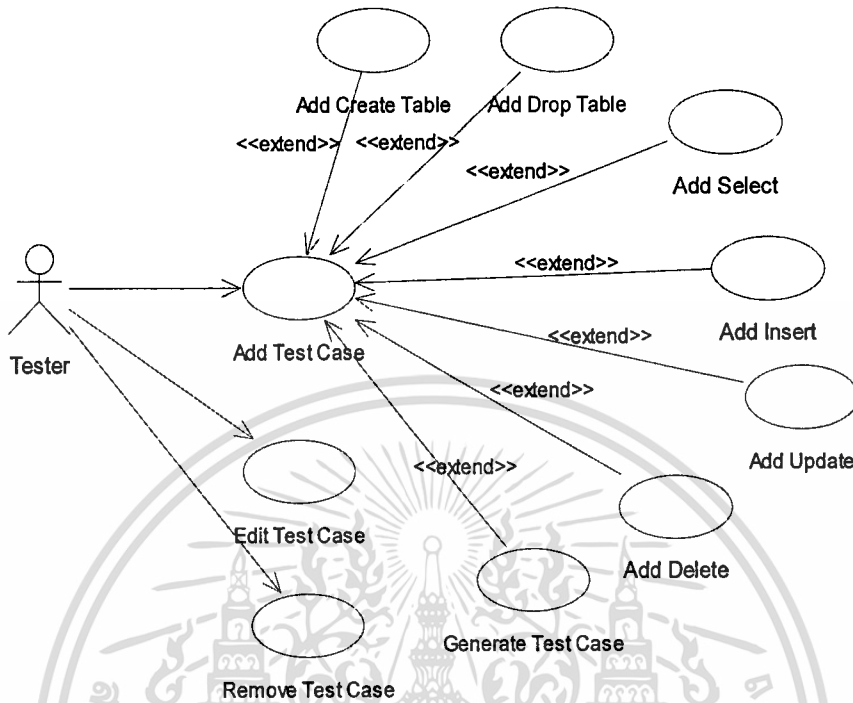


รูปที่ 3.3 Use Case Diagram ของระบบย่อย Manage DBMS



รูปที่ 3.4 Use Case Diagram ของระบบย่อย Manage Test Set

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้ผู้ใช้ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 Use Case Diagram ของระบบย่อย Manage Test Case

3.3 Use Case Description

จาก Use Case Diagram ที่ได้แสดงไว้ จะมีรายละเอียด ขั้นตอนการทำงานหรือ ขั้นตอนการใช้งานของแต่ละ Use Case ประกอบด้วย

3.3.1 Use Case – Setting Connection

ตารางที่ 3.1 Setting Connection Use Case Description

Actor :	Tester
Description :	Tester สร้างรายละเอียดในการติดต่อกับ DBMS ที่จะทำการทดสอบ
Basic Flow :	<ol style="list-style-type: none"> 1. Tester เลือก Connection ใน Setting menu 2. ชื่อ Connection ที่เคยสร้างไว้จะปรากฏในลิสต์ 3. Tester สามารถสร้าง connection ใหม่โดยการกดปุ่ม New ทำการลบ connection โดยการกดปุ่ม Delete หรือ แก้ไข connection โดยการกดปุ่ม Edit 4. กดปุ่ม Ok เมื่อทำการใส่รายละเอียดเรียบร้อยแล้ว 5. กดปุ่ม Cancel เมื่อต้องการยกเลิก

3.3.2 Use Case – Manage Project

ตารางที่ 3.2 Manage Project Use Case Description

Actor :	Tester
Description :	Tester ทำการสร้างโปรเจ็คใหม่ เปิดโปรเจ็คที่มีอยู่แล้ว หรือ จัดเก็บ โปรเจ็ค

3.3.3 Use Case – Manage DBMS

ตารางที่ 3.3 Manage DBMS Use Case Description

Actor :	Tester
Description :	Tester ทำการจัดการต่างๆ เกี่ยวกับ DBMS ที่จะทำการทดสอบ เช่น เพิ่ม DBMS แก้ไข DBMS หรือ ลบ DBMS
Pre-Condition :	ต้องเคยผ่าน Manage Project Use case ก่อน

3.3.4 Use Case – Manage Test Set

ตารางที่ 3.4 Manage Test Set Use Case Description

Actor :	Tester
Description :	Tester ทำการจัดการต่างๆ เกี่ยวกับชุดทดสอบที่จะทำการทดสอบ เช่น เพิ่มชุดทดสอบใหม่ เพิ่มชุดทดสอบโดยการสำเนาจากชุดทดสอบที่มีอยู่แล้ว เพิ่มชุดทดสอบจากชุดทดสอบที่มีอยู่แล้ว นำชุดทดสอบเข้ามาใน โครงการ ลบชุดทดสอบ ทำการทดสอบชุดทดสอบที่ได้สร้างไว้ในโครงการ หรือ คู่มือผลลัพธ์ของการทดสอบ
Pre-Condition :	ต้องเคยผ่าน Manage DBMS Use case ก่อน

3.3.5 Use Case – Manage Test Case

ตารางที่ 3.5 Manage Test Case Use Case Description

Actor :	Tester
Description :	Tester ทำการจัดการต่างๆ เกี่ยวกับ Test Case ที่จะทำการทดสอบ เช่น เพิ่มคำสั่ง CREATE, DROP, SELECT, INSERT, UPDATE และ DELETE ทำการ Generate Test Case แก้ไข Test Case หรือ ลบ Test Case
Pre-Condition :	ต้องเคยผ่าน Manage Test Set Use case ก่อน

3.3.6 Use Case – Execute Test Set

ตารางที่ 3.6 Execute Test Set Use Case Description

Actor :	Tester
Description :	ทำการทดสอบ DBMS โดยใช้ชุดทดสอบที่สร้างไว้
Basic Flow :	<ol style="list-style-type: none"> 1. ทำการเลือก DBMS และ Test Set ที่ต้องการทดสอบโดยการกดเลือกที่ Check Box ในต้นไม้ที่แสดงในส่วน Show Test Driver 2. เลือก Begin test ในเมนู Testing หรือ Start icon ใน Toolbar 3. หากยังไม่ได้บันทึกโครงการที่สร้างไว้ โปรแกรม TestDriver จะให้เราบันทึกโครงการเป็นไฟล์ XML ซึ่งสามารถเลือกได้ว่าบันทึกเป็น DTD หรือ Schema โดยกำหนดจากปุ่มบน Toolbar 4. TestDriver ส่งค่าที่เกี่ยวกับการเชื่อมต่อที่อ่านมาจากแฟ้มอินพุตส่งไปให้ DBClient 5. TestDriver โหลด Prediction1 ขึ้นมาโดยจะนำค่าที่เกี่ยวกับการเชื่อมต่อที่อ่านมาจากแฟ้มอินพุตส่งไปให้ Prediction1 ผ่านทางเมทอด prepare Connection จากอินเทอร์เฟซ IcheckDB ก่อนโดยจะส่งค่าต่างๆที่จำเป็นในการติดต่อฐานข้อมูลให้แก่ Prediction1 มีดังนี้ url (URL ของฐานข้อมูลทำการทดสอบ) dbDriver (คลาส JDBC Driver ของ DBMS ที่เราจะทดสอบ) user และ password (ชื่อผู้ใช้และรหัสผ่านที่ใช้สำหรับฐานข้อมูล) 6. จากนั้น TestDriver จะเรียกเมทอด beforeQuery เพื่อส่งคำสั่ง SQL ที่จะควรี ส่งไปให้ Prediction1 ก่อน โดยเมทอด beforeQuery จะทำการแยกคำสั่ง SQL เป็นกรณีต่างๆ ดังนี้ <ul style="list-style-type: none"> - SELECT จะทำการเรียกเมทอด beforeSelect - INSERT จะทำการเรียกเมทอด beforeInsert - UPDATE จะทำการเรียกเมทอด beforeUpdate - DELETE จะทำการเรียกเมทอด beforeDelete - CREATE จะทำการเรียกเมทอด beforeCreate - DROP จะทำการเรียกเมทอด beforeDrop 7. TestDriver อ่านคำสั่ง SQL ส่งไปให้คลาส DBClient เพื่อทำการควรีไปที่ DBMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในหน่วยงานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้วยวิธีการใด

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	<p>8. จากนั้น Test Driver จะรอ DBClient ส่งผลลัพธ์การคิวรีจนได้รับเรียบร้อยแล้ว</p> <p>9. แล้วจะเรียกเมทอด afterQuery ไปยัง Oracle เพื่อขอทราบผลลัพธ์การตรวจสอบความถูกต้องจากผลของ beforeQuery และ afterQuery และคืนค่าเป็น true ถ้าการทดสอบถูกต้อง และเป็น false ถ้าการทดสอบได้ผลผิดพลาดพร้อมด้วย ExpectedError</p> <p>10. เมื่อ Prediction1 ส่งผลลัพธ์การตรวจสอบกลับมา ก็จะรวบรวมผลลัพธ์ของทุกชุดทดสอบแล้วเขียนลงยังเอาต์พุต</p> <p>11. โปรแกรมจะแจ้งให้ทราบว่าการทดสอบเสร็จแล้ว</p> <p>12. TestDriver เรียก SaxParser ฟิล์เอาต์พุต แล้วจะส่งค่า DriverListResult</p> <p>13. แสดงผลลัพธ์จากการทดสอบแบบต้น ไม้ในกรอบ Show Test Result</p>
--	--

3.3.7 Use Case – View Test Result

ตารางที่ 3.7 View Test Result Use Case Description

Actor :	Tester
Description :	แสดงผลลัพธ์จากการทดสอบ
Basic Flow :	<ol style="list-style-type: none"> 1. เมื่อทำการ Test แล้วจะแสดงผลลัพธ์ในรูปแบบต้น ไม้ในกรอบ Show Test Result โดยอัตโนมัติ หรือเลือก Test result ในเมนู View หรือ Report icon ใน Toolbar จะได้ ตารางแสดงผลลัพธ์ของชุดทดสอบที่เคยทำการทดสอบในโครงการนี้

3.3.8 Use Case – Create New Project

ตารางที่ 3.8 Create New Project Use Case Description

Actor :	Tester
Description :	Tester สร้าง โครงการใหม่สำหรับเก็บชุดทดสอบอินพุต และผลลัพธ์เอาต์พุต
Basic Flow :	<ol style="list-style-type: none"> 1. Tester เลือก New test ใน File menu หรือเลือก New icon จาก Toolbar 2. Tester ใส่ชื่อโครงการ 3. Tester ใส่พารสำหรับสร้างไฟล์เคอร์โครงการ 4. กดปุ่ม Ok เมื่อทำการใส่รายละเอียดเรียบร้อยแล้ว 5. กดปุ่ม Cancel เมื่อต้องการยกเลิก

3.3.9 Use Case – Open Project

ตารางที่ 3.9 Open Project Use Case Description

Actor :	Tester
Description :	Tester เปิดโครงการที่เคยสร้างแล้วเพื่อทำการแก้ไขชุดทดสอบ ทดสอบชุดทดสอบอีกครั้ง หรือดูผลการทดสอบ
Basic Flow :	<ol style="list-style-type: none"> 1. Tester เลือก Open ใน File menu หรือเลือก Open icon จาก Toolbar 2. Tester เลือกไฟล์โครงการที่ต้องการเปิด(นามสกุล tdg) 3. ชุดทดสอบในโครงการที่เปิดจะแสดง ลักษณะแบบต้นไม้ในกรอบ Show Test Driver และหากโครงการนี้เคยทำการทดสอบแล้วจะแสดงผลลัพท์การทดสอบล่าสุดแบบต้นไม้ในกรอบ Show Test Result

3.3.10 Use Case – Save Project

ตารางที่ 3.10 Save Project Use Case Description

Actor :	Tester
Description :	Tester ทำการจัดเก็บชุดทำสอบอินพุตในพาธของโครงการที่กำหนดเมื่อตอนสร้างโครงการใหม่
Basic Flow :	<ol style="list-style-type: none"> 1. Tester เลือก Save หรือ Save as ใน File menu หรือเลือก Save icon จาก Toolbar

3.3.11 Use Case – Add DBMS

ตารางที่ 3.11 Add DBMS Use Case Description

Actor :	Tester
Description :	Tester เลือกประเภท DBMS ที่จะทดสอบ (โปรแกรมรองรับ ODBC, MySQL, ORACLE Database และ SQL Server) และใส่รายละเอียดต่างๆ ในการติดต่อกับ DBMS
Basic Flow :	<ol style="list-style-type: none"> 1. Tester เลือก Add DBMS Testing ใน DBMS menu หรือเลือก Add DBMS icon จาก Toolbar 2. Tester เลือกประเภท DBMS ที่จะทดสอบใน กรอบ Main 3. Tester เลือกคลาส ที่จะทำนายการทดสอบใน กรอบ Main 4. กดปุ่ม Ok เมื่อทำการใส่รายละเอียด DBMS เรียบร้อย

	5. เรียก Create Test Set Use Case 6. กดปุ่ม Cancel เมื่อต้องการยกเลิก
--	--

3.3.12 Use Case – Edit DBMS

ตารางที่ 3.12 Edit DBMS Use Case Description

Actor :	Tester
Description :	Tester ทำการแก้ไข DBMS เช่นเปลี่ยน DBMS ที่จะทำการทดสอบ หรือเปลี่ยนคลาสสำหรับทำการ Oracle(ทำนาย)
Basic Flow :	1. Tester คลิกไปยัง Node DBMS ที่จะแก้ไข 2. เมื่อแก้ไขเรียบร้อยแล้วกดปุ่ม Update เพื่อทำการยืนยัน

3.3.13 Use Case – Remove DBMS

ตารางที่ 3.13 Remove DBMS Use Case Description

Actor :	Tester
Description :	Tester ทำการลบ DBMS
Basic Flow :	1. Tester คลิกไปยัง โหนด DBMS ที่จะลบ 1. กดปุ่ม Remove เพื่อทำการลบ

3.3.14 Use Case – Create Test Set

ตารางที่ 3.14 Create Test Set Use Case Description

Actor :	Tester
Description :	Tester ทำการสร้างชุดทดสอบ
Basic Flow :	1. Tester เลือก Add Test Set icon จาก Toolbar (ถ้าเป็นการแทรก Test Set ให้คลิกไปยัง โหนด Test Set แล้ว Test Set icon จะสามารถเลือกได้)

3.3.15 Use Case – Add New Test Set

ตารางที่ 3.15 Add New Test Set Use Case Description

Actor :	Tester
Description :	Tester ทำการเพิ่มชุดทดสอบ โดยการ เพิ่มชุดทดสอบใหม่

Basic Flow :	<ol style="list-style-type: none"> 1. Tester เลือกประเภทในการสร้างชุดทดสอบ โดยเลือกสร้างชุดทดสอบใหม่ 2. เรียก Add Test Case Use Case
---------------------	--

3.3.16 Use Case – Copy to new Test Set

ตารางที่ 3.16 Copy to new Test Set Use Case Description

Actor :	Tester
Description :	Tester ทำการเพิ่มชุดทดสอบ โดยการสำเนาชุดทดสอบจากชุดทดสอบที่มีอยู่แล้ว
Basic Flow :	<ol style="list-style-type: none"> 1. Tester เลือกประเภทในการสร้างชุดทดสอบ โดยเลือกสร้างชุดทดสอบใหม่ 2. Tester คลิกถูก Create new test set by copy exist test set 3. เลือกชุดทดสอบจากรายชื่อชุดทดสอบที่ปรากฏในลิสต์โดยการคลิกถูก 4. กดปุ่ม Ok เมื่อทำการเลือกเรียบร้อยแล้ว 5. กดปุ่ม OK ใน Toolbar เพื่อยืนยันในการสร้างชุดทดสอบเรียบร้อยแล้ว 6. กดปุ่ม Cancel เมื่อต้องการยกเลิก

3.3.17 Use Case – Add Exist Test Set

ตารางที่ 3.17 Add Exist Test Set Use Case Description

Actor :	Tester
Description :	Tester ทำการเพิ่มชุดทดสอบ โดยการ เพิ่มชุดทดสอบจากชุดทดสอบที่มีอยู่แล้ว
Basic Flow :	<ol style="list-style-type: none"> 1. Tester เลือกประเภทในการสร้างชุดทดสอบ โดยเลือกสร้างชุดทดสอบจากชุดทดสอบที่มีอยู่แล้ว 2. เลือกชุดทดสอบจากรายชื่อชุดทดสอบที่ปรากฏในลิสต์โดยการคลิกถูก 3. กดปุ่ม Ok เมื่อทำการเลือกเรียบร้อยแล้ว 4. กดปุ่ม OK ใน Toolbar เพื่อยืนยันในการสร้างชุดทดสอบเรียบร้อยแล้ว 5. กดปุ่ม Cancel เมื่อต้องการยกเลิก

3.3.18 Use Case – Import Test Set

ตารางที่ 3.18 Import Test Set Use Case Description

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับคนใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Actor :	Tester
Description :	Tester ทำการเพิ่มชุดทดสอบ โดยการนำชุดทดสอบเข้ามาในโครงการ
Basic Flow :	<ol style="list-style-type: none"> 1. Tester เลือกประเภทในการสร้างชุดทดสอบ โดยเลือกนำชุดทดสอบเข้ามาในโครงการ 2. เลือกชุดทดสอบในพาร์ทที่อยู่ 3. กดปุ่ม Ok เมื่อทำการเลือกเรียบร้อยแล้ว 4. กดปุ่ม OK ใน Toolbar เพื่อยืนยันในการสร้างชุดทดสอบเรียบร้อยแล้ว 5. กดปุ่ม Cancel เมื่อต้องการยกเลิก

3.3.19 Use Case – Remove Test Set

ตารางที่ 3.19 Remove Test Set Use Case Description

Actor :	Tester
Description :	Tester ทำการลบ Test Set
Basic Flow :	<ol style="list-style-type: none"> 1. Tester คลิกไปยังไอคอน Test Set ที่จะลบ 2. กดปุ่ม Remove เพื่อทำการลบ

3.3.20 Use Case – Add Test Case

ตารางที่ 3.20 Add Test Case Use Case Description

Actor :	Tester
Description :	Tester ทำการเพิ่ม Test Case ลงในชุดทดสอบ Test Set
Basic Flow :	<ol style="list-style-type: none"> 1. เลือกประเภท Test Case ที่อยู่ใน TestCase menu หรือเลือก SQL icon จาก Toolbar ประกอบด้วย เมนูย่อยดังนี้ <ul style="list-style-type: none"> - Create table จะเป็นการเพิ่ม test case ที่จะทำการสร้างตารางลงใน database - Drop table จะเป็นการเพิ่ม test case ที่จะทำการลบตารางออกจาก database - Select จะเป็นการเพิ่ม test case ที่จะทำการเลือกข้อมูลจากตารางต่างๆ ด้วยคำสั่ง select - Insert จะเป็นการเพิ่ม test case ที่จะทำการเพิ่มข้อมูลลงในตาราง ด้วยคำสั่ง Insert

	<ul style="list-style-type: none"> - Update จะเป็นการเพิ่ม test case ที่จะทำการแก้ไขข้อมูลลงในตาราง ด้วยคำสั่ง Update - Delete จะเป็นการเพิ่ม test case ที่จะทำการลบข้อมูลลงในตาราง ด้วยคำสั่ง Delete - Generate จะเป็นการสร้าง test case โดยอัตโนมัติ
--	---

3.3.21 Use Case – Add Create table

ตารางที่ 3.21 Add Create table Use Case Description

Actor :	Tester
Description :	เพิ่ม Test Case ที่จะทำการสร้างตารางลงใน database
Basic Flow :	<ol style="list-style-type: none"> 1. ทำการสร้าง Test Case แบบ Create table โดยสามารถเลือกได้ 2 แบบ คือ <ul style="list-style-type: none"> - Basic คือมี GUI ช่วยสร้างคำสั่ง create - Advance คือ ผู้ทดสอบเขียนภาษา SQL เอง ซึ่งเหมาะกับ SQL ที่ซับซ้อนมากขึ้น 2. กดปุ่ม OK เมื่อทำการสร้างคำสั่งเสร็จ หรือ Cancel เพื่อยกเลิก 3. กดปุ่ม OK ใน Toolbar เพื่อยืนยันในการสร้างชุดทดสอบเรียบร้อยแล้ว

3.3.22 Use Case – Add Drop table

ตารางที่ 3.22 Add Drop table Use Case Description

Actor :	Tester
Description :	เพิ่ม Test Case ที่จะทำการลบตารางออกจาก database
Basic Flow :	<ol style="list-style-type: none"> 1. ทำการสร้าง Test Case แบบ Drop table โดยสามารถเลือกได้ 2 แบบ คือ <ul style="list-style-type: none"> - Basic คือมี GUI ช่วยสร้างคำสั่ง drop - Advance คือ ผู้ทดสอบเขียนภาษา SQL เอง ซึ่งเหมาะกับ SQL ที่ซับซ้อนมากขึ้น 2. กดปุ่ม OK เมื่อทำการสร้างคำสั่งเสร็จ หรือ Cancel เพื่อยกเลิก 3. กดปุ่ม OK ใน Toolbar เพื่อยืนยันในการสร้างชุดทดสอบเรียบร้อยแล้ว

3.3.23 Use Case – Add Select

ตารางที่ 3.23 Add Select Use Case Description

Actor :	Tester
Description :	เพิ่ม Test Case ที่จะทำการเลือกข้อมูลจากตารางต่างๆ ด้วยคำสั่ง Select
Basic Flow :	<ol style="list-style-type: none"> 1. ทำการสร้าง Test Case แบบ Select โดยสามารถเลือกได้ 2 แบบ คือ <ul style="list-style-type: none"> - Basic คือมี GUI ช่วยสร้างคำสั่ง select - Advance คือ ผู้ทดสอบเขียนภาษา SQL เอง ซึ่งเหมาะกับ SQL ที่ซับซ้อนมากขึ้น 2. กดปุ่ม OK เมื่อทำการสร้างคำสั่งเสร็จ หรือ Cancel เพื่อยกเลิก 3. กดปุ่ม OK ใน Toolbar เพื่อยืนยันในการสร้างชุดทดสอบเรียบร้อยแล้ว

3.3.24 Use Case – Add Insert

ตารางที่ 3.24 Add Insert Use Case Description

Actor :	Tester
Description :	เพิ่ม Test Case ที่จะทำการเพิ่มข้อมูลลงในตารางต่างๆ ด้วยคำสั่ง Insert
Basic Flow :	<ol style="list-style-type: none"> 1. ทำการสร้าง Test Case แบบ Insert โดยสามารถเลือกได้ 2 แบบ คือ <ul style="list-style-type: none"> - Basic คือมี GUI ช่วยสร้างคำสั่ง insert - Advance คือ ผู้ทดสอบเขียนภาษา SQL เอง ซึ่งเหมาะกับ SQL ที่ซับซ้อนมากขึ้น 2. กดปุ่ม OK เมื่อทำการสร้างคำสั่งเสร็จ หรือ Cancel เพื่อยกเลิก 3. กดปุ่ม OK ใน Toolbar เพื่อยืนยันในการสร้างชุดทดสอบเรียบร้อยแล้ว

3.3.25 Use Case – Add Update

ตารางที่ 3.25 Add Update Use Case Description

Actor :	Tester
Description :	เพิ่ม Test Case ที่จะทำการแก้ไขข้อมูลลงในตารางต่างๆ ด้วยคำสั่ง Update
Basic Flow :	<ol style="list-style-type: none"> 1. ทำการสร้าง Test Case แบบ Update โดยสามารถเลือกได้ 2 แบบ คือ <ul style="list-style-type: none"> - Basic คือมี GUI ช่วยสร้างคำสั่ง update - Advance คือ ผู้ทดสอบเขียนภาษา SQL เอง ซึ่งเหมาะกับ SQL ที่ซับซ้อนมากขึ้น

	<ol style="list-style-type: none"> 2. กดปุ่ม OK เมื่อทำการสร้างคำสั่งเสร็จ หรือ Cancel เพื่อยกเลิก 3. กดปุ่ม OK ใน Toolbar เพื่อยืนยันในการสร้างชุดทดสอบเรียบร้อยแล้ว
--	---

3.3.26 Use Case – Add Delete

ตารางที่ 3.26 Add Delete Use Case Description

Actor :	Tester
Description :	เพิ่ม Test Case ที่จะทำการลบข้อมูลลงในตารางต่างๆ ด้วยคำสั่ง Delete
Basic Flow :	<ol style="list-style-type: none"> 1. ทำการสร้าง Test Case แบบ Delete โดยสามารถเลือกได้ 2 แบบ คือ <ul style="list-style-type: none"> - Basic คือมี GUI ช่วยสร้างคำสั่ง delete - Advance คือ ผู้ทดสอบเขียนภาษา SQL เอง ซึ่งเหมาะกับ SQL ที่ซับซ้อนมากขึ้น 2. กดปุ่ม OK เมื่อทำการสร้างคำสั่งเสร็จ หรือ Cancel เพื่อยกเลิก 3. กดปุ่ม OK ใน Toolbar เพื่อยืนยันในการสร้างชุดทดสอบเรียบร้อยแล้ว

3.3.27 Use Case – Generate Test Case

ตารางที่ 3.27 Generate Test Case Use Case Description

Actor :	Tester
Description :	จะเป็นการสร้าง test case โดยอัตโนมัติ
Basic Flow :	<ol style="list-style-type: none"> 1. คลิกเลือกคำสั่งที่ต้องการ 2. เลือกประเภทในการทดสอบคือ Test to pass หรือ Test to fail 3. เลือก ตาราง และ ฟิวล์ 4. กดปุ่มปุ่ม Next 5. เลือกคำสั่งที่สร้างอัตโนมัติเสร็จแล้วโดยการคลิกถูก 6. กดปุ่ม OK เมื่อทำการสร้างคำสั่งเสร็จ หรือ Cancel เพื่อยกเลิก 7. กดปุ่ม OK ใน Toolbar เพื่อยืนยันในการสร้างชุดทดสอบเรียบร้อยแล้ว

3.3.28 Use Case – Edit Test Case

ตารางที่ 3.28 Edit Test Case Use Case Description

Actor :	Tester
----------------	--------

Description :	ทำการแก้ไข Test Case ที่เราสร้างไว้
Basic Flow :	<ol style="list-style-type: none"> 1. คลิกไปที่ไหนค Test Case ที่แสดงแบบต้นไม้ในกรอบ Show Test Driver ที่ต้องการแก้ไข <ol style="list-style-type: none"> 1.1 คำสั่ง SQL Command จะปรากฏใน Text Input(SQL Command)ในกรอบ Main 1.2 แก้ไขเรียบร้อยแล้วกดปุ่ม Update

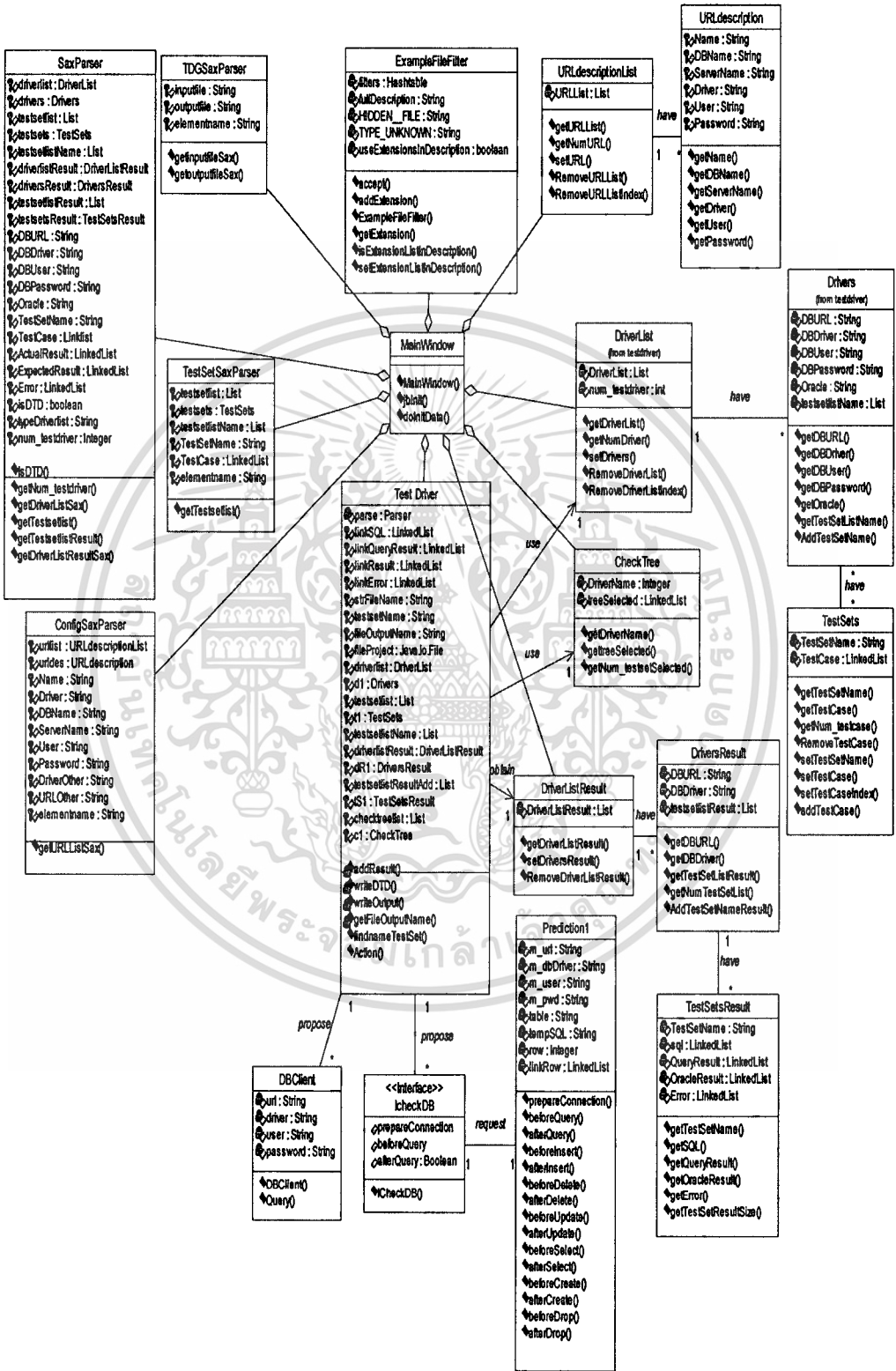
3.3.29 Use Case – Remove Test Case

ตารางที่ 3.29 Remove Test Case Use Case Description

Actor :	Tester
Description :	ทำการลบ Test Case หรือ Test Driver ที่เราสร้างไว้
Basic Flow :	<ol style="list-style-type: none"> 1. คลิกไปที่ไหนค Test Case ที่แสดงแบบต้นไม้ในกรอบ Show Test Driver ที่ต้องการลบ <ol style="list-style-type: none"> 1.1 คำสั่ง SQL Command จะปรากฏใน Text Input(SQL Command)ในกรอบ Main 1.2 ต้องการลบ Test Case กดปุ่ม Remove

3.4 Class Diagram

ขั้นตอนต่อไปเป็นขั้นตอนที่จะหา และกำหนดคลาส โดยหามาจาก Use case แล้วนำมาพิจารณาของ class ที่เป็นไปได้ซึ่งจะได้ Class Diagram ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 3.6 Class Diagram ของระบบทดสอบเครือข่ายที่ใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีมีการนำไปใช้

จากรูปที่ 3.6 Class Diagram ประกอบด้วย 19 คลาส โดย

3.4.1 คลาส MainWindow เป็นคลาสหน้าจอ Interface หลักกับผู้ใช้

3.4.2 คลาส SaxParser เป็นคลาสอ่านข้อมูลต่างๆจากไฟล์อินพุต และเอาต์พุตที่เก็บแบบ XML จะถูกเรียกใช้โดยคลาส MainWindow มีเมทอด

- IsDTD() ส่งค่ากลับ true ถ้าไฟล์ที่อ่านเป็น XML DTD ส่ง false เมื่อเป็น XML Schema

- getNum_testdriver() ส่งค่าเป็นจำนวน DBMS ที่ทำการทดสอบ

- getDriverListSax() ส่งค่าเป็นคลาส DriverList ซึ่งเก็บรายละเอียดต่างๆของหลายๆ Driver

- getTestsetlist() ส่งค่าเป็น List ซึ่งเก็บรายละเอียดต่างๆของหลายๆ Test Set

- getTestsetlistResult() ส่งค่าเป็น List ซึ่งเก็บรายละเอียดต่างๆของ Test Set ของผลลัพธ์

- getDriverListResultSax() ส่งค่าเป็นคลาส DriverListResult ซึ่งเก็บรายละเอียดต่างๆของหลายๆ Driver ของผลลัพธ์

3.4.3 คลาส ConfigSaxParser เป็นคลาสอ่านข้อมูลจากไฟล์ Config ซึ่งจะอ่านทุกครั้งเมื่อ MainWindow ถูกเรียกมาครั้งแรก โดยไฟล์ Config จะเก็บรายละเอียดในการ Connection ต่างๆ

- getURLListSax() ส่งค่าเป็นคลาส URLdescriptionList ซึ่งเก็บรายละเอียดต่างๆของหลายๆ Connection

3.4.4 คลาส TDGSaxParser เป็นคลาสอ่านข้อมูลจากไฟล์โครงการ(นามสกุล tdg) ซึ่งถูกใช้เมื่อ MainWindow เกิดเหตุการณ์เปิดโครงการ

- getInputFileSax() ส่งค่าเป็นพารของไฟล์อินพุต

- getoutputfileSax() ส่งค่าเป็นพารของไฟล์เอาต์พุต

3.4.5 คลาส TestSetSaxParser เป็นคลาสอ่านข้อมูลที่เก็บรายละเอียด Test Set ซึ่งถูกใช้เมื่อ MainWindow ต้องการนำ Test Set เข้ามาในโครงการ

- getTestsetlist() ส่งค่าเป็น List เก็บรายละเอียดของ Test Set

3.4.6 คลาส ExampleFileFilter เป็นคลาสที่ MainWindow เรียกใช้เมื่อต้องการกรองนามสกุลไฟล์

3.4.7 คลาส URLdescriptionList เป็นคลาสเก็บหลายๆ รายละเอียดของการ Connection

- getURLList() ส่งค่าเป็น List ของการ Connection

- getNumURL() ส่งขนาดของ List ของการ Connection

- setURL() เพิ่มรายละเอียดการ Connection ใน List
- RemoveURLList() ลบรายละเอียดการ Connection ใน List ทั้งหมด
- RemoveURLListIndex() ลบรายละเอียดการ Connection ในตำแหน่งที่กำหนด

3.4.8 คลาส URLdescription เป็นคลาสเก็บรายละเอียดการ Connection

- getName() ส่งค่าเป็นชื่อของการ Connection
- getDBName() ส่งค่าเป็นชื่อฐานข้อมูล
- getServerName() ส่งค่าเป็นชื่อเซิร์ฟเวอร์
- getDriver() ส่งค่าเป็นไดร์เวอร์
- getUser() ส่งค่าเป็นชื่อผู้ใช้
- getPassword() ส่งค่าเป็นพาสเวิร์ด

3.4.9 คลาส DriverList เป็นคลาสเก็บหลายๆรายละเอียด DBMS ที่ทดสอบคือ คลาส Drivers(อินพุต)

- getDriverList() ส่งค่าเป็น List คลาส Drivers ที่ทดสอบในโครงการ
- getNumDriver() ส่งค่าเป็นจำนวนคลาส Drivers ที่ทดสอบในโครงการ
- setDrivers() เพิ่มคลาส Driversเข้าไปใน List
- RemoveDriverList() ลบ Drivers ใน List ทั้งหมด
- RemoveDriverListIndex() ลบ Drivers ณ ตำแหน่งที่กำหนด

3.4.10 คลาส Drivers เป็นคลาสเก็บรายละเอียด DBMS ที่ทดสอบ (อินพุต)

- getDBURL() ส่งค่าเป็น URL
- getDBDriver() ส่งค่าเป็น Driver
- getDBUser() ส่งค่าเป็นชื่อผู้ใช้
- getDBPassword() ส่งค่าเป็น Password
- getOracle() ส่งค่าเป็นชื่อคลาสในการทำนาย
- getTestSetListName() ส่งค่าเป็น List ที่เก็บชื่อ Test Set
- AddTestSetName() เพิ่ม List ที่เก็บชื่อ Test Set ลงใน List ใน Drivers

3.4.11 คลาส TestSets เป็นคลาสเก็บรายละเอียดชุดทดสอบ Test Set (อินพุต)

- getTestSetName() ส่งค่าเป็นชื่อ Test Set
- getTestCase() ส่งค่าเป็นคำสั่ง SQL (Test Case) ณ ตำแหน่งที่กำหนด
- getNum_testcase() ส่งค่าเป็นจำนวน Test Case

- setTestSetName() กำหนดชื่อ Test Set ใหม่
- setTestCase() กำหนด Test Case ทั้ง List ใหม่
- setTestCaseIndex() กำหนด Test Case ณ ตำแหน่งที่กำหนด ใหม่
- addTestCase() เพิ่ม Test Case ณ ตำแหน่งที่กำหนด

3.4.12 คลาส **CheckTree** เป็นคลาสที่คลาส **TestDriver** เรียกใช้เพื่อเลือก DBMS หรือ Test Set ที่จะทดสอบตามที่ Tester เลือก check box บนต้นไม้ Test Driver

- getDriverName() ส่งค่าในตำแหน่งใน DriverList
- gettreeSelected() ส่งค่าตำแหน่งใน TestSetList
- getNum_testsetSelected() ส่งขนาดของ List ที่เก็บตำแหน่ง Test Set

3.4.13 คลาส **DriverListResult** เป็นคลาสเก็บหลายๆรายละเอียด DBMS ที่ทดสอบแล้ว คือคลาส **DriversResult**(เอาต์พุต)

- getDriverListResult() ส่งค่าเป็น List คลาส **DriversResult** ที่ทดสอบในโครงการแล้ว
- setDriverResult() เพิ่ม คลาส **DriversResult** เข้าไปใน List
- RemoveDriverListResult() ลบ คลาส **DriversResult** ใน List ทั้งหมด

3.4.14 คลาส **DriversResult** เป็นคลาสเก็บรายละเอียด DBMS ที่ทดสอบแล้ว (เอาต์พุต)

- getDBURL() ส่งค่าเป็น URL
- getDBDriver() ส่งค่าเป็น Driver
- getTestSetListResult() ส่งค่าเป็น List ที่เก็บ คลาส **TestSetsResult**
- getNumTestSetList() ส่งค่าเป็นขนาดจำนวน Test Set
- AddTestSetNameResult() เพิ่ม List ที่เก็บ คลาส **TestSetsResult**

3.4.15 คลาส **TestSetsResult** เป็นคลาสเก็บรายละเอียดชุดทดสอบ Test Set (เอาต์พุต)

- getTestSetName() ส่งค่าเป็นชื่อ Test Set
- getSQL() ส่งค่าเป็นคำสั่ง SQL
- getQueryResult() ส่งค่าเป็นผลลัพธ์ในการคิวรี(บูลีน)
- getOracleResult() ส่งค่าเป็นผลลัพธ์ในการทำนาย(บูลีน)
- getError() ส่งค่าเป็นข้อความผิดพลาด ถ้าไม่มีจะเป็น null
- getTestSetResultSize() ส่งค่าเป็นขนาดของ TestSetResult

3.4.16 คลาส **TestDriver** นำค่าต่างๆ เกี่ยวกับการเชื่อมต่อที่อ่านมาจากแฟ้มอินพุตส่งให้ Oracle ผ่านทางเมธอด `prepare Connection` แล้วส่งคำสั่ง SQL ไปให้ Oracle โดยเรียกใช้จากเมธอด

เอกสาร `beforeQuery` และส่งคำสั่ง SQL ไปให้คลาส `DBCLIENT` จากนั้นเรียกเมธอด `afterQuery` ไป
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยัง Oracle เพื่อขอทราบผลลัพธ์การตรวจสอบความถูกต้องของ DBMS เสร็จแล้วรวบรวมผลลัพธ์จากการคิวรี และการตรวจสอบเขียนลงยังแฟ้มเอาต์พุต

- addResult() นำคำสั่ง SQL เก็บลงใน linkSQL ผลลัพธ์คิวรี เก็บลงใน linkQueryResult และ ผลลัพธ์การทำนาย เก็บลงใน linkResult

- writeDTD() เขียน XML Specification ลงไฟล์

- writeOutput() เขียน ไฟล์เอาต์พุต XML ลงไฟล์

- getFileOutputName() ส่งชื่อไฟล์เอาต์พุต

- findnameTestSet() หาชื่อ Test Set

- Action() ทำการส่งคำสั่ง SQL ไปคิวรีไปยัง DBMS และ ทำนายด้วยคลาส Prediction1 แล้วรวบรวมอินพุต และผลลัพธ์ที่แท้จริงและทำนาย แล้วเรียกใช้เมธอด writeDTD และ writeOutput

3.4.17 DBClient คิวรีคำสั่ง SQL ที่ได้ไปยัง DBMS และส่งผลลัพธ์การคิวรีไปยังคลาส TestDriver

- DBClient() รับค่าต่างๆ ในการ Connection

- Query() ทำการคิวรีตามคำสั่ง SQL แล้วส่งผลลัพธ์เป็น true หรือ false

3.4.18 ICheckDB เป็นอินเตอร์เฟสระหว่างคลาส TestDriver กับ Prediction1

- prepareConnection เป็นเมธอดที่ใช้ในการรับ และเตรียมค่าต่างๆ ในการเชื่อมต่อ

- beforeQuery เป็นเมธอดที่ใช้ในการรับคำสั่ง SQL จาก TestDriver

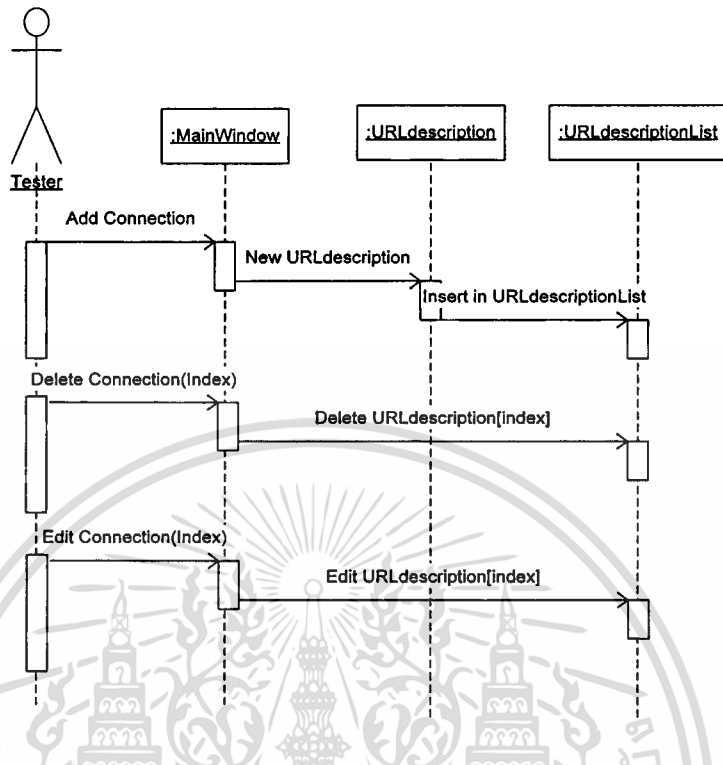
- afterQuery เป็นเมธอดที่ใช้ทดสอบคำสั่ง SQL ที่ได้จาก เมธอด beforeQuery เมื่อตรวจสอบเรียบร้อยแล้วจะคืนค่าออกมาเป็น true หรือ false ส่งกลับไปยังคลาส TestDriver

3.4.19 Prediction1 เป็นคลาสที่ใช้ในการทำนาย(Oracle)ซึ่งจะอธิบายรายละเอียดในบทที่4

3.5 Sequence Diagrams

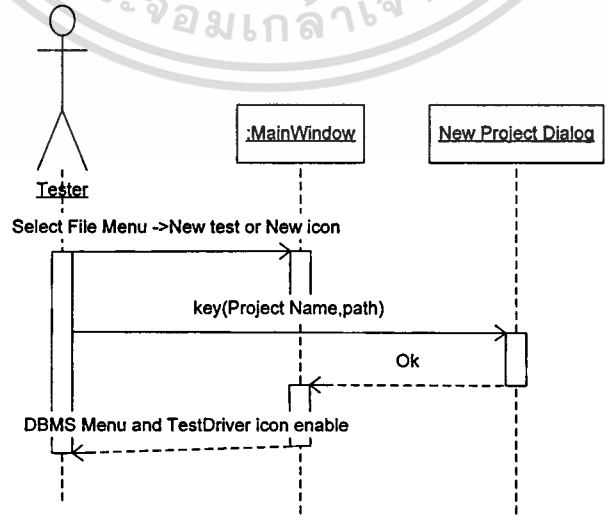
เมื่อได้ Scenario (Use Case Description) และ Class Diagram แล้วขั้นตอนถัดไปเป็นการกำหนดหน้าที่ความรับผิดชอบหรือพฤติกรรมต่างๆ ให้กับ Class ที่หามาได้ โดยจะพิจารณาจาก Scenario ของแต่ละ Use Case ว่าในแต่ละ Scenario ของแต่ละ Use case นั้นมีขั้นตอนการทำงานอะไรบ้าง แต่ละขั้นตอนจะกำหนดให้ Class ใดที่มีหน้าที่รับผิดชอบในการทำงาน

เทคนิควิธีที่ได้นำมาใช้คือ การสร้างแบบจำลอง Sequence Diagram ขึ้นมา ซึ่งโดยทั่วไปเราจะทำการสร้าง Sequence Diagram ต่อหนึ่ง Use case ดังในรูปที่ 3.7 ถึง รูปที่ 3.31



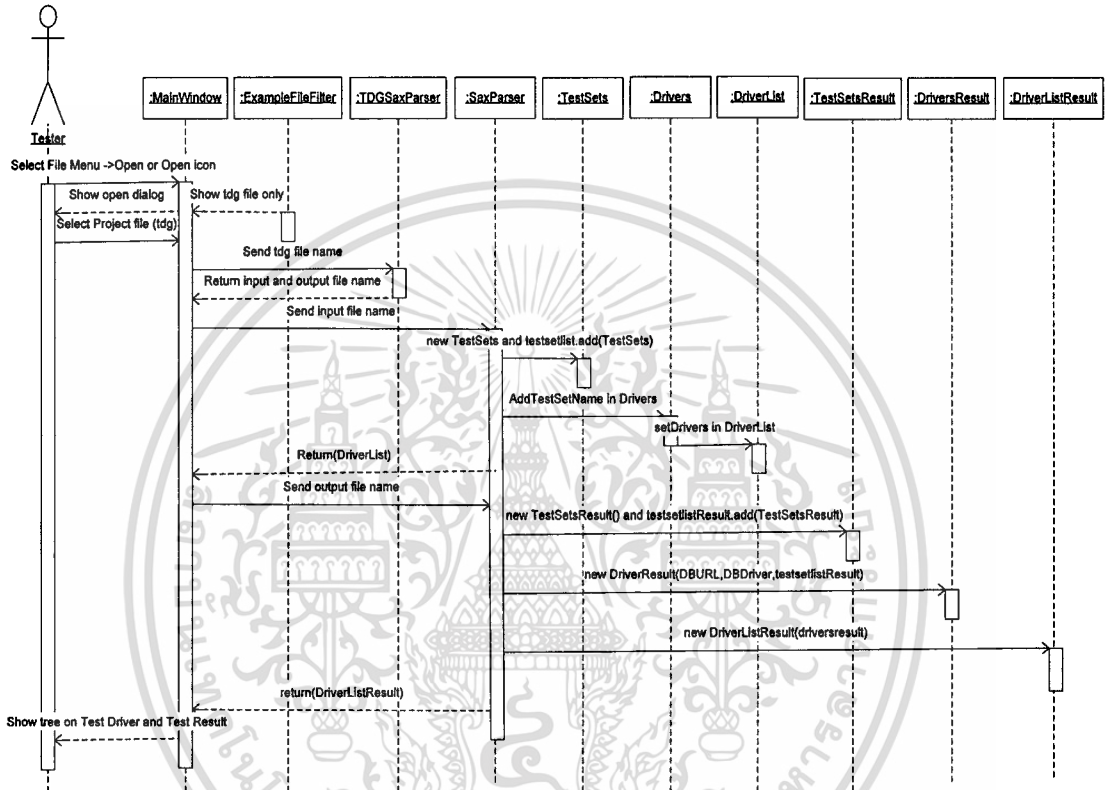
รูปที่ 3.7 Sequence Diagram ของ Setting Connection

จากรูปที่ 3.7 เป็นการอธิบายถึงลำดับการทำงานของการทำงานของการกำหนดค่าการติดต่อกับ DBMS ที่ จะทำการทดสอบซึ่ง Tester สามารถทำการ เพิ่ม ลบ หรือแก้ไขค่าการติดต่อได้ผ่านทาง MainWindow โดย MainWindow จะทำการบันทึกค่าต่างๆ ลงในคลาส URLdescription และ รวบรวมหลายๆ การติดต่อเป็น List แล้วเก็บในคลาส URLdescriptionList



รูปที่ 3.8 Sequence Diagram ของ Create New Project

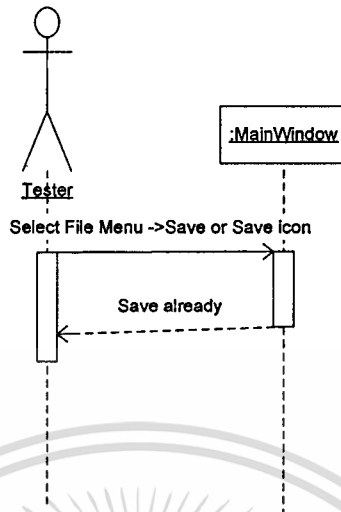
จากรูปที่ 3.8 เป็นการอธิบายถึงลำดับการทำงานของกรสร้างโครงการใหม่โดย Tester จะติดต่อกับ MainWindow คือเลือก New test ในเมนู File หรือ เลือก New icon จากนั้นจะทำการใส่ชื่อโครงการ และ พารโฟร์เคอร์ของโครงการเมื่อเรียบร้อย เมนู DBMS และ TestDriver icon จะปรากฏ



รูปที่ 3.9 Sequence Diagram ของ Open Project

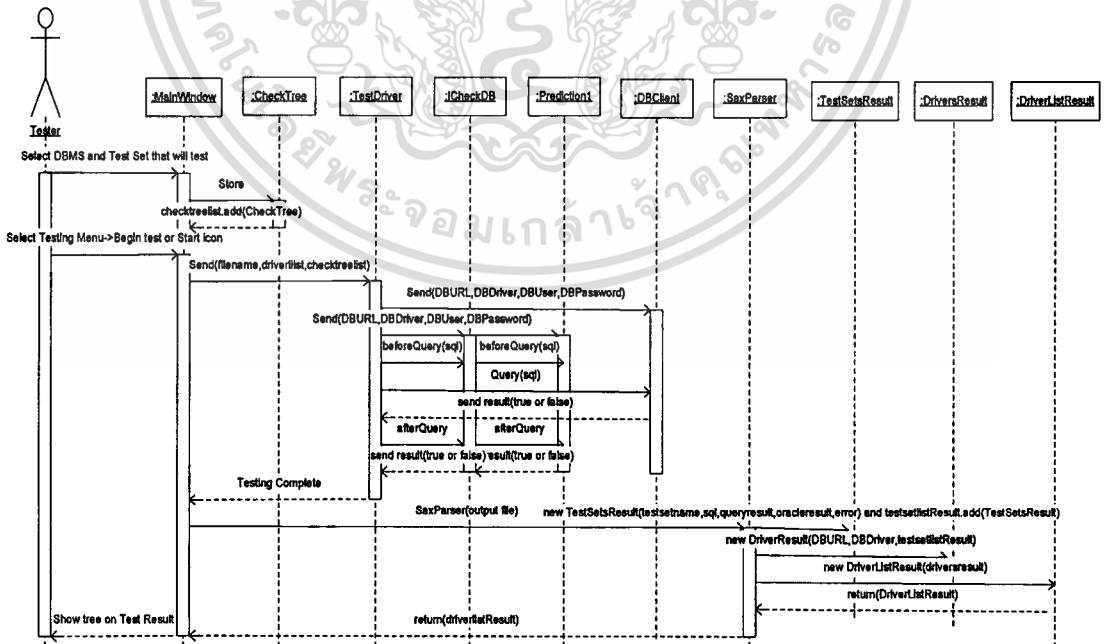
จากรูปที่ 3.9 เป็นการอธิบายถึงลำดับการทำงานของกรเปิดโครงการโดย Tester ทำการเลือก Open ในเมนู File หรือ Open icon จากนั้นคลาส ExampleFileFilter จะทำการกรองชื่อเฉพาะไฟล์ tdg แล้ว Tester เลือกไฟล์โครงการ คลาส MainWindow ส่งพารโครงการที่เลือกให้คลาส TDGSaxParser ซึ่งส่งค่ากลับมาเป็นพารของไฟล์อินพุตและเอาต์พุต จากนั้น MainWindow ส่งพารของไฟล์อินพุตให้ SaxParser จะทำการอ่านค่าต่างๆมาเก็บไว้ในคลาส TestSets, Drivers และ DriverList แล้วจะส่งค่ากลับมาที่ MainWindow เป็นคลาส DriverList จากนั้นจะทำการส่งพารของไฟล์เอาต์พุตให้ SaxParser จะทำการอ่านค่าต่างๆมาเก็บไว้ในคลาส TestSetsResult, DriversResult และ DriverListResult แล้วจะส่งค่ากลับมาที่ MainWindow เป็นคลาส DriverListResult เมื่อเสร็จคลาส MainWindow จะทำการแสดงค่าต่างๆในแบบต้นไม้ในกรอบ Test Driver และ Test Result

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 Sequence Diagram ของ Save Project

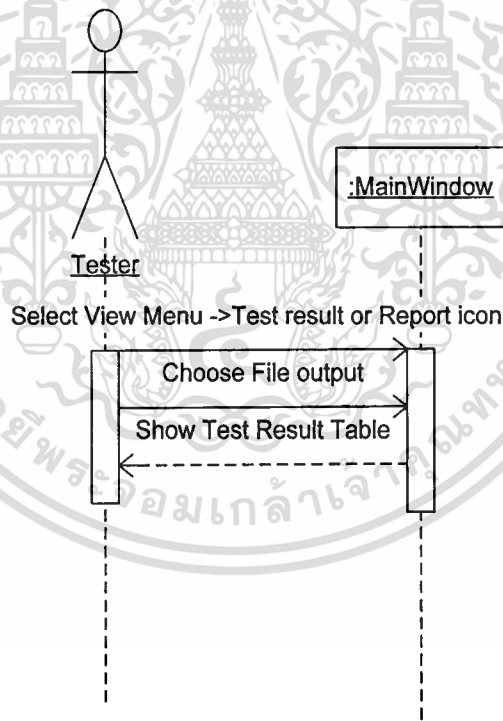
จากรูปที่ 3.10 เป็นการอธิบายถึงลำดับการทำงานของการทำงานของการบันทึกโครงการ โดย Tester ทำการเลือก Save ในเมนู File หรือ Save icon จากนั้น MainWindow จะทำการบันทึกไฟล์ในพุดเมื่อเสร็จสิ้นจะแจ้งให้ Tester รับทราบ



รูปที่ 3.11 Sequence Diagram ของ ExecuteTest Set

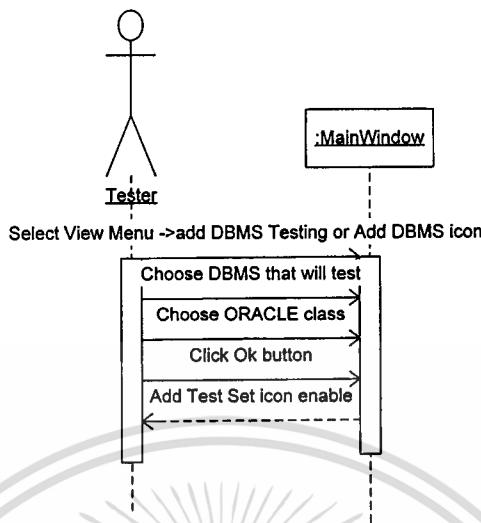
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.11 เป็นการอธิบายถึงลำดับการทำงานของ การทดสอบ โดย Tester ทำการเลือก DBMS และ Test Set ที่ต้องการทดสอบ โดยเก็บค่าไว้ในคลาส CheckTree แล้ว Tester ทำการเลือก Begin test ในเมนู Testing หรือ Start icon หลังจากนั้น MainWindow ตรวจสอบหากมีการแก้ไขค่าต่างๆ จะทำการบันทึกเพิ่มอินพุต แล้วค่อยส่ง พารามิเตอร์ให้ TestDriver แล้ว TestDriver ส่งค่าในการเชื่อมต่อให้กับ DBClient และ Prediction1 ผ่าน ICheckDB Interface ตามลำดับ หลังจากนั้นส่งคำสั่ง sql ไปทำนายให้กับ Prediction1 ก่อนครั้งแรก แล้วจึงส่งไปทดสอบกับ DBClient แล้วรอรับผลการทดสอบจริง หลังจากนั้นส่ง sql ไป Prediction1 อีกครั้ง โดยนำผลทำนายครั้งแรกและครั้งสองมาเปรียบเทียบกันได้แล้วจึงส่งผลทำนายไปให้ TestDriver หลังจากนั้นเขียนลงไฟล์เอาต์พุตแล้วแจ้งว่าทดสอบเรียบร้อยแล้ว MainWindow ส่งพารามิเตอร์ให้ SaxParser อ่านค่าเอาต์พุตมาเก็บไว้ใน TestSetsResult, DriversResult และ DriverListResult หลังจากนั้นส่งค่า DriverListResult ให้ MainWindow แล้วจึงแสดงค่าแบบต้นไม้ในกรอบ Test Result



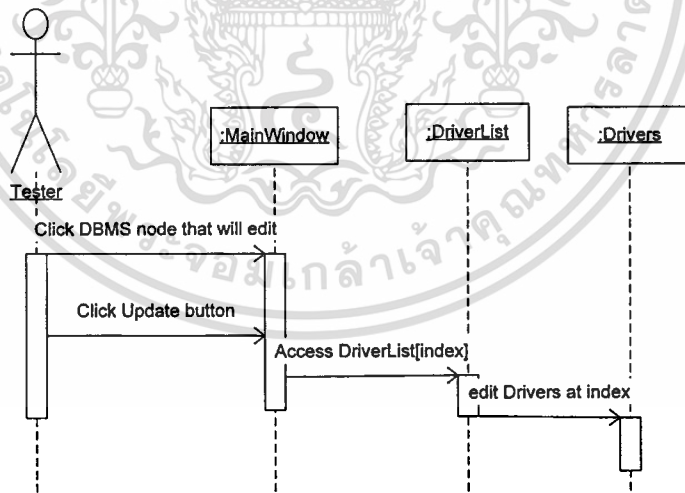
รูปที่ 3.12 Sequence Diagram ของ View Test Result

จากรูปที่ 3.12 เป็นการอธิบายถึงลำดับการทำงานของ การดูผลลัพธ์เอาต์พุต โดย Tester เลือก Test result ในเมนู View แล้ว Tester เลือกไฟล์เอาต์พุตที่ต้องการดูผลทดสอบ แล้ว MainWindow แสดงเป็นตาราง Test Result



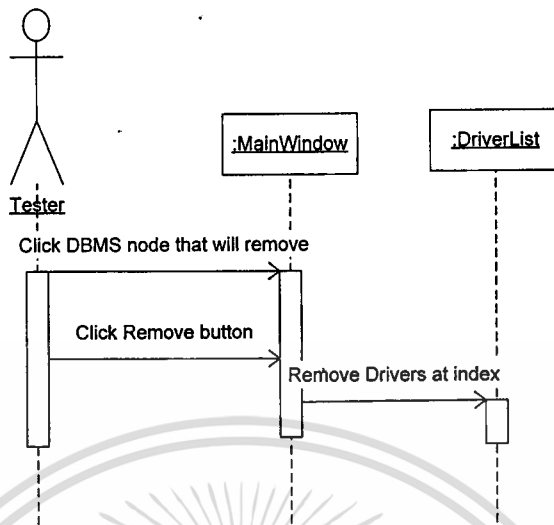
รูปที่ 3.13 Sequence Diagram ของ Add DBMS

จากรูปที่ 3.13 เป็นการอธิบายถึงลำดับการทำงานของการทำงานของการเพิ่ม DBMS ที่ทดสอบ โดย Tester เลือก add DBMS Testing ในเมนู View หรือ Add DBMS icon หลังจากนั้น Tester เลือก DBMS ที่จะทดสอบ และคลาสที่จะทำนายเสร็จแล้วคลิกปุ่ม Ok แล้ว Add Test Set icon จะปรากฏ



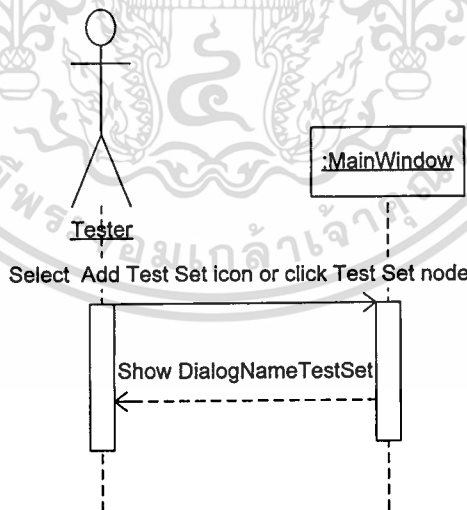
รูปที่ 3.14 Sequence Diagram ของ Edit DBMS

จากรูปที่ 3.14 เป็นการอธิบายถึงลำดับการทำงานของการทำงานของการแก้ไข DBMS โดย Tester คลิกไปที่ไหน DBMS ที่จะทำการแก้ไข เมื่อแก้ไขเรียบร้อยแล้วคลิกปุ่ม Update คลาส MainWindow จะเข้าถึง Drivers ใน DriverList แล้วทำการแก้ไข



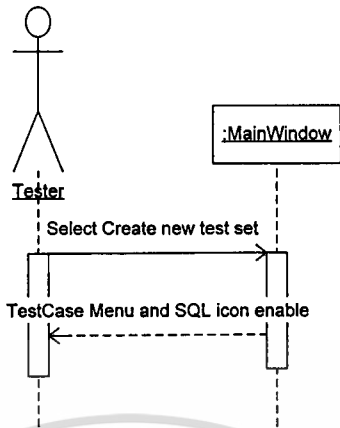
รูปที่ 3.15 Sequence Diagram ของ Remove DBMS

จากรูปที่ 3.15 เป็นการอธิบายถึงลำดับการทำงานของกรลบ DBMS โดย Tester คลิกไปที่ โหนด DBMS ที่จะทำการลบ แล้วคลิกปุ่ม Remove คลาส MainWindow จะเข้าถึง Drivers ใน DriverList แล้วทำการลบ



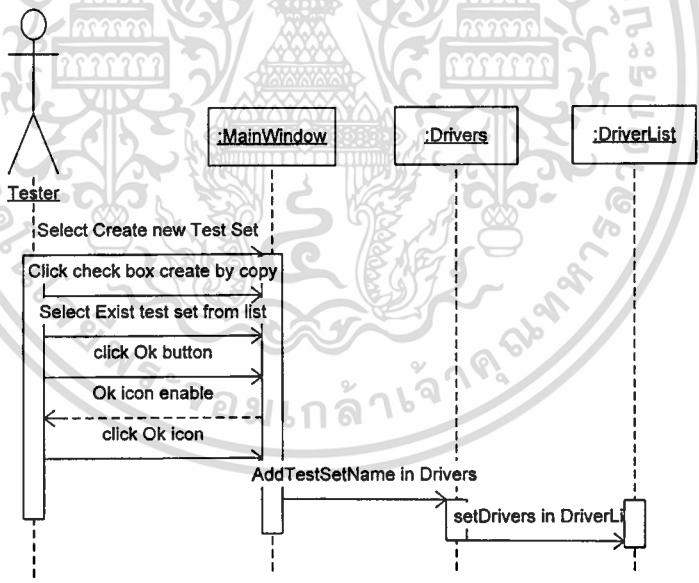
รูปที่ 3.16 Sequence Diagram ของ Create Test Set

จากรูปที่ 3.16 เป็นการอธิบายถึงลำดับการทำงานของกรสร้าง Test Set โดย Tester คลิก Test Set icon แล้วจะแสดง DialogNameTestSet



รูปที่ 3.17 Sequence Diagram ของ Add New Test Set

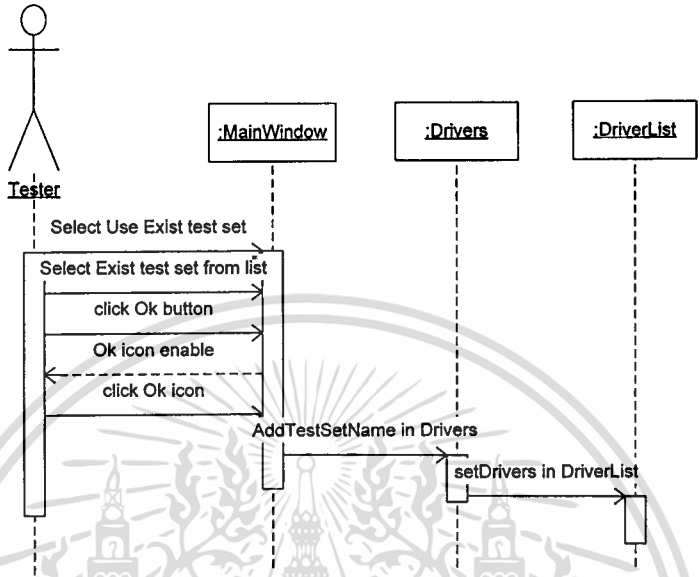
จากรูปที่ 3.17 เป็นการอธิบายถึงลำดับการทำงานของการทำงานของการเพิ่ม Test Set ใหม่ โดย Tester เลือก Create new test set แล้ว เมนู TestCase และ SQL icon จะปรากฏ



รูปที่ 3.18 Sequence Diagram ของ Copy to new Test Set

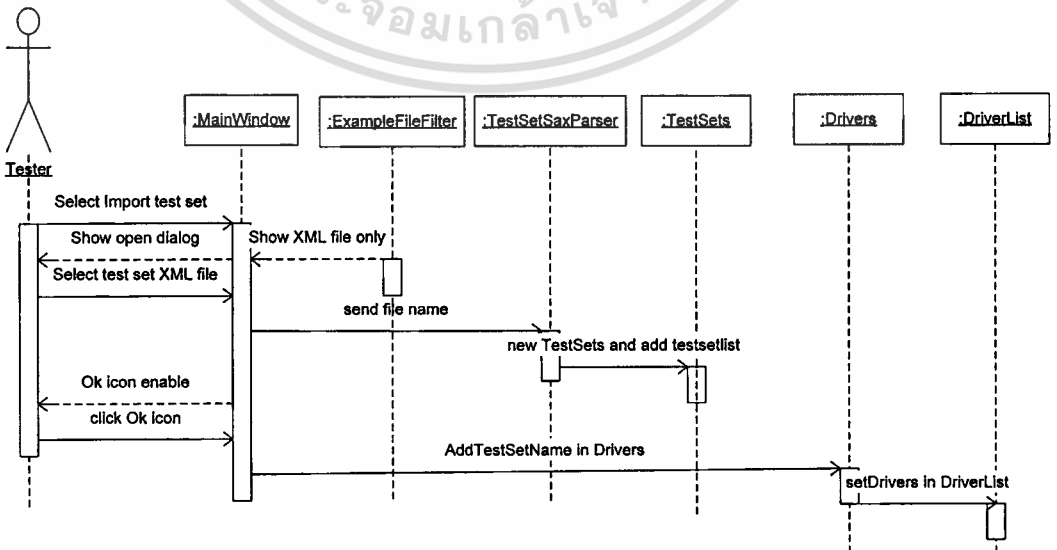
จากรูปที่ 3.18 เป็นการอธิบายถึงลำดับการทำงานของการทำงานของการเพิ่ม Test Set โดยการสำเนาจาก Test Set ที่มีอยู่แล้วในโครงการ โดย Tester เลือก Create new test set และคลิกถูก Create new test set by copy exist test set หลังจากนั้นเลือก Test Set จาก list แล้วคลิกปุ่ม Ok จะทำให้ Ok icon

ปรากฏ แล้ว Tester คลิก Ok Icon เพื่อยืนยันว่าสิ้นสุดการเพิ่ม Test Set ลงใน DBMS ที่จะทดสอบ แล้ว MainWindow จะทำการเพิ่มชื่อ TestSetName ลงใน Drivers



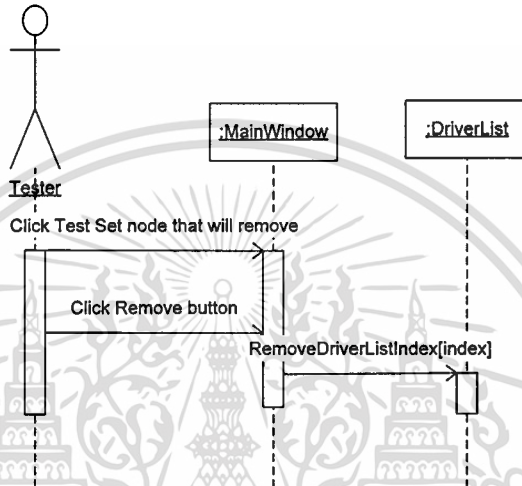
รูปที่ 3.19 Sequence Diagram ของ Add Exist Test Set

จากรูปที่ 3.19 เป็นการอธิบายถึงลำดับการทำงานของการทำงานของการเพิ่ม Test Set จาก Test Set ที่มีอยู่แล้วในโครงการ โดย Tester เลือก Use Exist test set หลังจากนั้นเลือก Test Set จาก list แล้วคลิกปุ่ม Ok จะทำให้ Ok icon ปรากฏ แล้ว Tester คลิก Ok Icon เพื่อยืนยันว่าสิ้นสุดการเพิ่ม Test Set ลงใน DBMS ที่จะทดสอบ แล้ว MainWindow จะทำการเพิ่มชื่อ TestSetName ลงใน Drivers



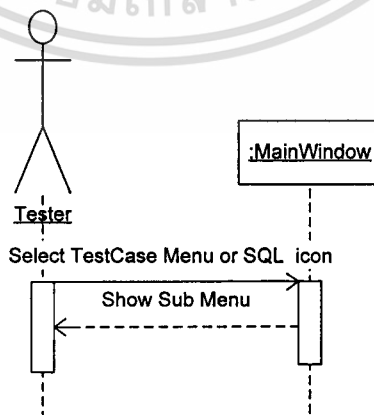
รูปที่ 3.20 Sequence Diagram ของ Import Test Set

จากรูปที่ 3.20 เป็นการอธิบายถึงลำดับการทำงานของกรนำเข้า Test Set มาในโครงการ โดย Tester เลือก Import test set แล้วคลาส ExampleFileFilter ทำการกรองชนิดไฟล์โดยแสดงเฉพาะไฟล์ XML จากนั้น Tester เลือก Test Set XML file แล้ว MainWindow ส่งพาธไฟล์ให้ TestSetSaxParser อ่านค่าต่างๆโดย ข้อมูลเกี่ยวกับ Test Set เก็บไว้ใน TestSets และ Ok icon ปรากฏ แล้ว Tester คลิก Ok icon ทำให้ MainWindow เพิ่มTestSetName เข้าไปใน Drivers



รูปที่ 3.21 Sequence Diagram ของ Remove Test Set

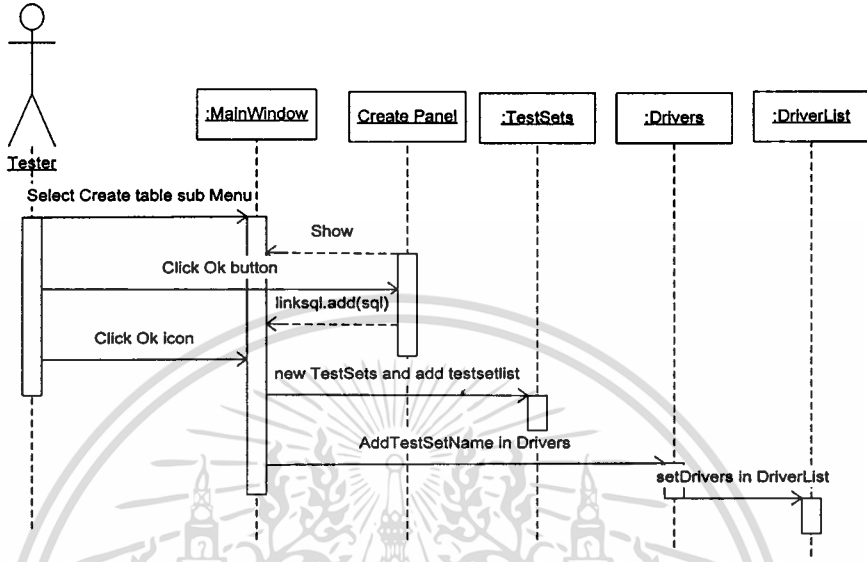
จากรูปที่ 3.21 เป็นการอธิบายถึงลำดับการทำงานของกรลบ Test Set โดย Tester คลิก โหนด Test Set ที่จะลบ แล้วคลิกปุ่ม Remove จากนั้น MainWindow ทำการลบ TestSetName ใน Drivers



รูปที่ 3.22 Sequence Diagram ของ Add Test Case

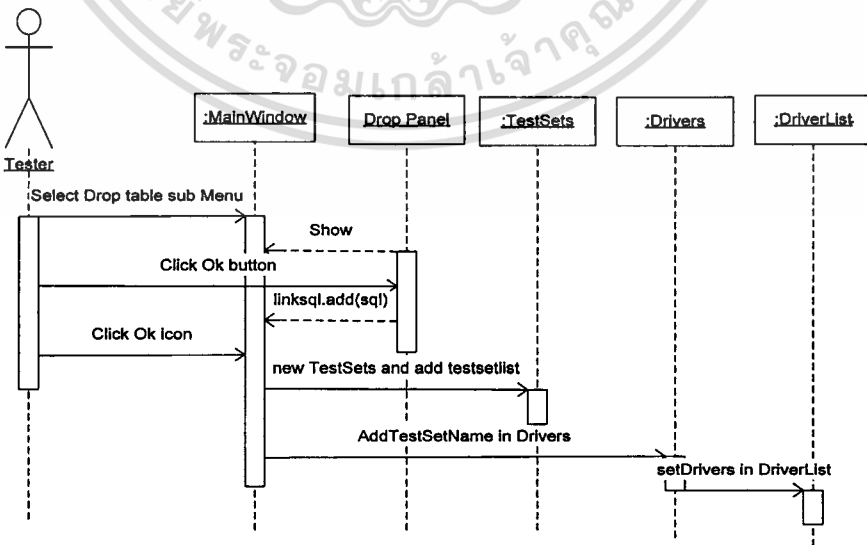
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.22 เป็นการอธิบายถึงลำดับการทำงานของกรเพิ่ม Test Case โดย Tester เลือกเมนู TestCase หรือ SQL icon แล้ว MainWindow จะแสดงเมนูย่อยเพื่อเลือกคำสั่ง SQL



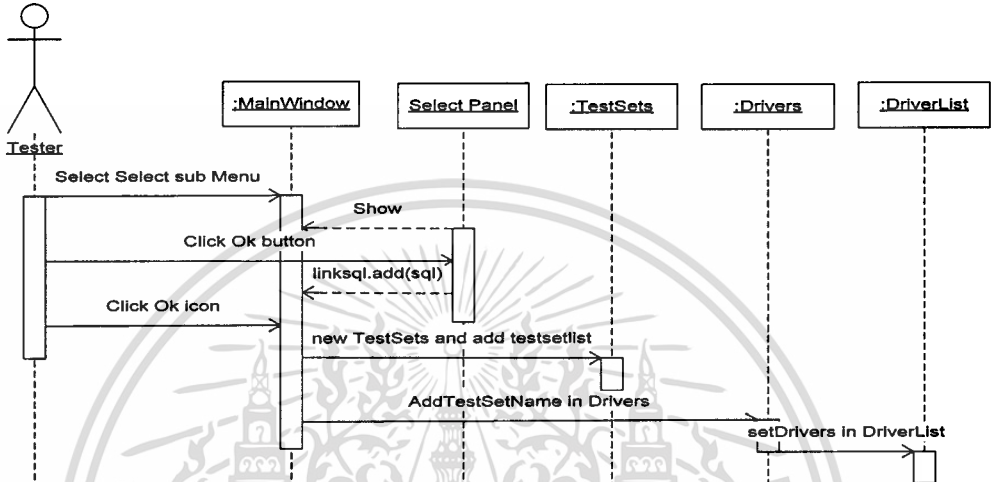
รูปที่ 3.23 Sequence Diagram ของ Add Create table

จากรูปที่ 3.23 เป็นการอธิบายถึงลำดับการทำงานของกรเพิ่มคำสั่ง Create table โดย Tester เลือกเมนูย่อย Create table จากนั้นแสดง Create Panel ซึ่ง Tester สร้างคำสั่งได้เมื่อเสร็จแล้วคลิกปุ่ม Ok จะเก็บคำสั่ง sql ลงใน linksql เมื่อเพิ่มคำสั่งเรียบร้อยแล้วคลิก Ok icon จะทำการบันทึก TestSet ลงใน TestSets และเพิ่ม TestSetName ลงใน Drivers



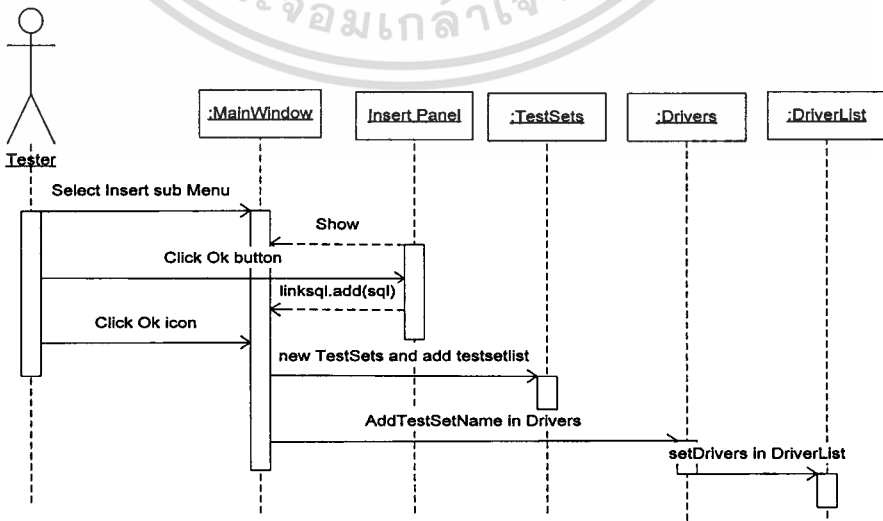
รูปที่ 3.24 Sequence Diagram ของ Add Drop table

จากรูปที่ 3.24 เป็นการอธิบายถึงลำดับการทำงานของกรเพิ่มคำสั่ง Drop table โดย Tester เลือกเมนูย่อย Drop table จากนั้นแสดง Drop Panel ซึ่ง Tester สร้างคำสั่งได้เมื่อเสร็จแล้วคลิกปุ่ม Ok จะเก็บคำสั่ง sql ลงใน linksql เมื่อเพิ่มคำสั่งเรียบร้อยแล้ว คลิก Ok icon จะทำการบันทึก TestSet ลงใน TestSets และเพิ่ม TestSetName ลงใน Drivers



รูปที่ 3.25 Sequence Diagram ของ Add Select

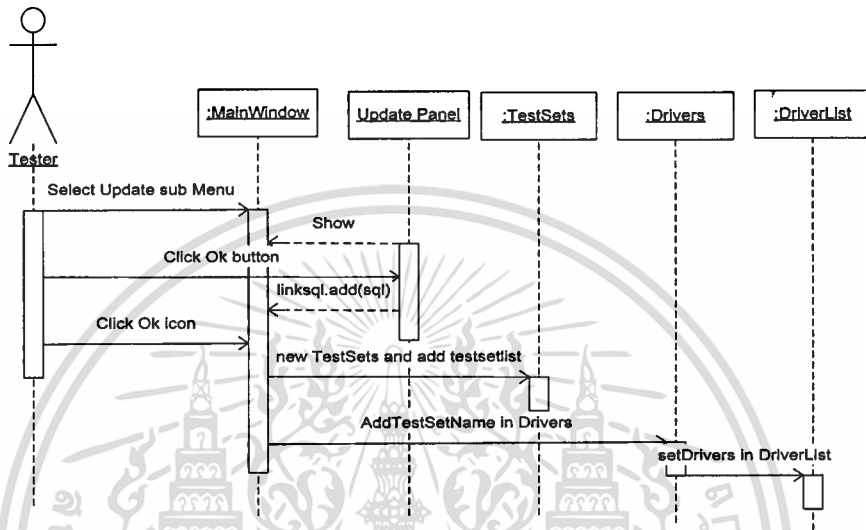
จากรูปที่ 3.25 เป็นการอธิบายถึงลำดับการทำงานของกรเพิ่มคำสั่ง Select โดย Tester เลือกเมนูย่อย Select จากนั้นแสดง Select Panel ซึ่ง Tester สร้างคำสั่งได้เมื่อเสร็จแล้วคลิกปุ่ม Ok จะเก็บคำสั่ง sql ลงใน linksql เมื่อเพิ่มคำสั่งเรียบร้อยแล้ว คลิก Ok icon จะทำการบันทึก TestSet ลงใน TestSets และเพิ่ม TestSetName ลงใน Drivers



รูปที่ 3.26 Sequence Diagram ของ Add Insert

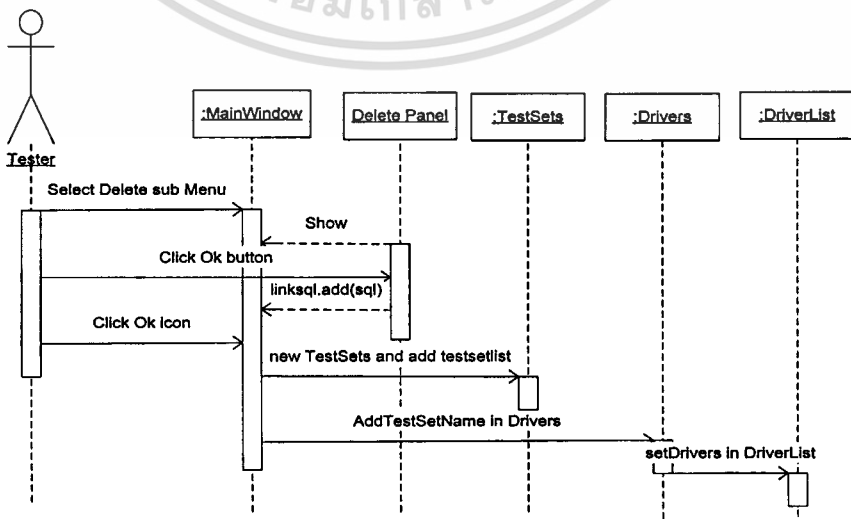
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.26 เป็นการอธิบายถึงลำดับการทำงานของการทำงานของการเพิ่มคำสั่ง Insert โดย Tester เลือกเมนูย่อย Insert จากนั้นแสดง Insert Panel ซึ่ง Tester สร้างคำสั่งได้เมื่อเสร็จแล้วคลิกปุ่ม Ok จะเก็บคำสั่ง sql ลงใน linksql เมื่อเพิ่มคำสั่งเรียบร้อยแล้ว คลิก Ok icon จะทำการบันทึก TestSet ลงใน TestSets และเพิ่ม TestSetName ลงใน Drivers



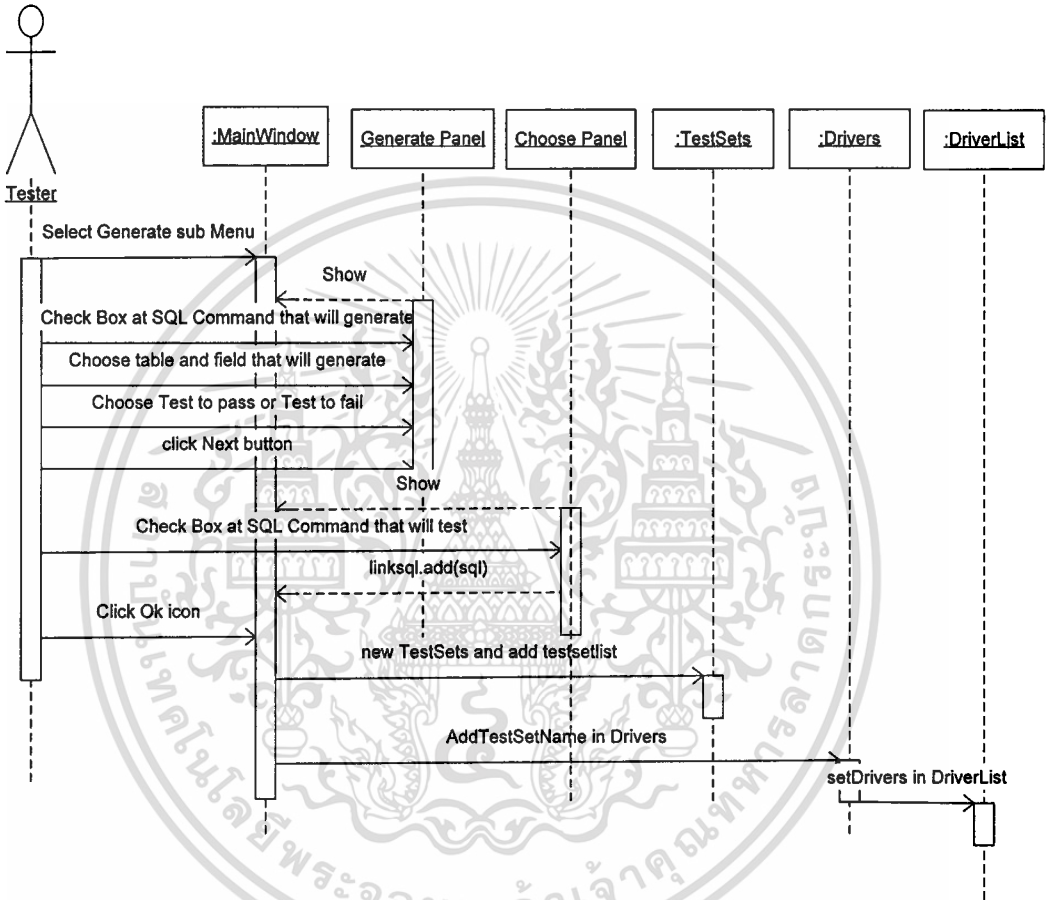
รูปที่ 3.27 Sequence Diagram ของ Add Update

จากรูปที่ 3.27 เป็นการอธิบายถึงลำดับการทำงานของการทำงานของการเพิ่มคำสั่ง Update โดย Tester เลือกเมนูย่อย Update จากนั้นแสดง Update Panel ซึ่ง Tester สร้างคำสั่งได้เมื่อเสร็จแล้วคลิกปุ่ม Ok จะเก็บคำสั่ง sql ลงใน linksql เมื่อเพิ่มคำสั่งเรียบร้อยแล้ว คลิก Ok icon จะทำการบันทึก TestSet ลงใน TestSets และเพิ่ม TestSetName ลงใน Drivers



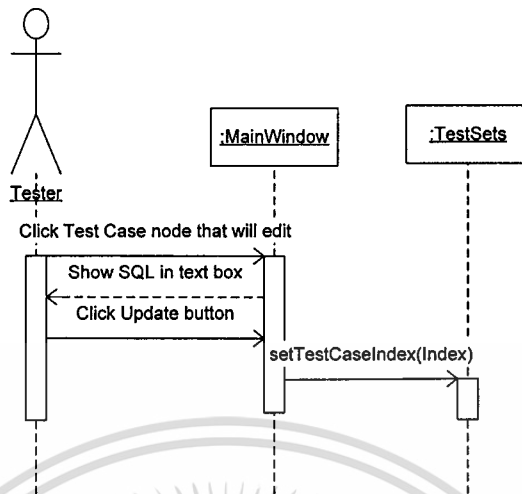
รูปที่ 3.28 Sequence Diagram ของ Add Delete

จากรูปที่ 3.28 เป็นการอธิบายถึงลำดับการทำงานของกรเพิ่มคำสั่ง Delete โดย Tester เลือกเมนูย่อย Delete จากนั้นแสดง Delete Panel ซึ่ง Tester สร้างคำสั่งได้เมื่อเสร็จแล้วคลิกปุ่ม Ok จะเก็บคำสั่ง sql ลงใน linksql เมื่อเพิ่มคำสั่งเรียบร้อยแล้ว คลิก Ok icon จะทำการบันทึก TestSet ลงใน TestSets และเพิ่ม TestSetName ลงใน Drivers



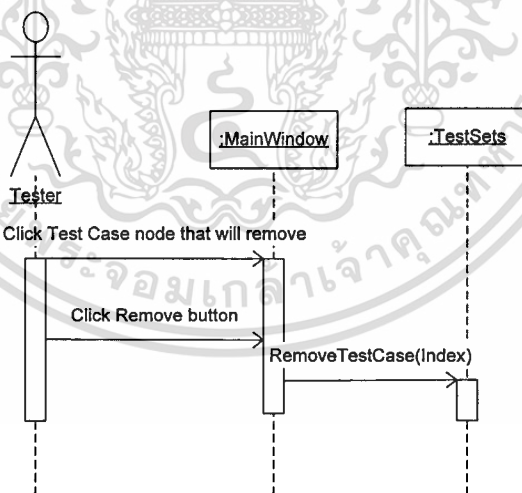
รูปที่ 3.29 Sequence Diagram ของ Generate Test Case

จากรูปที่ 3.29 เป็นการอธิบายถึงลำดับการทำงานของกรสร้าง Test Case อัตโนมัติ โดย Tester เลือกเมนูย่อย Generate จากนั้นแสดง Generate Panel แล้ว Tester ทำการ check box คำสั่ง sql ที่จะสร้างอัตโนมัติ เลือกตารางและฟิลด์ที่จะสร้างคำสั่ง sql และเลือกว่าต้องการ Test to pass หรือ Test to fail เมื่อเสร็จกดปุ่ม Next จากนั้นแสดง Choose Panel ซึ่ง Tester เลือกว่าต้องการคำสั่ง sql ใดบ้างสำหรับการทดสอบเมื่อเสร็จแล้วคลิกปุ่ม Ok จะเก็บคำสั่ง sql ลงใน linksql เมื่อเพิ่มคำสั่งเรียบร้อยแล้ว คลิก Ok icon จะทำการบันทึก TestSet ลงใน TestSets และเพิ่ม TestSetName ลงใน Drivers



รูปที่ 3.30 Sequence Diagram ของ Edit Test Case

จากรูปที่ 3.30 เป็นการอธิบายถึงลำดับการทำงานของกรแก้ไข Test Case โดย Tester คลิก โหนด Test Case ที่จะแก้ไข ซึ่งคำสั่ง SQL เก็บลงในช่องข้อความเมื่อแก้ไขเสร็จคลิกปุ่ม Update ซึ่งจะไปแก้ไขใน TestSets



รูปที่ 3.31 Sequence Diagram ของ Remove Test Case

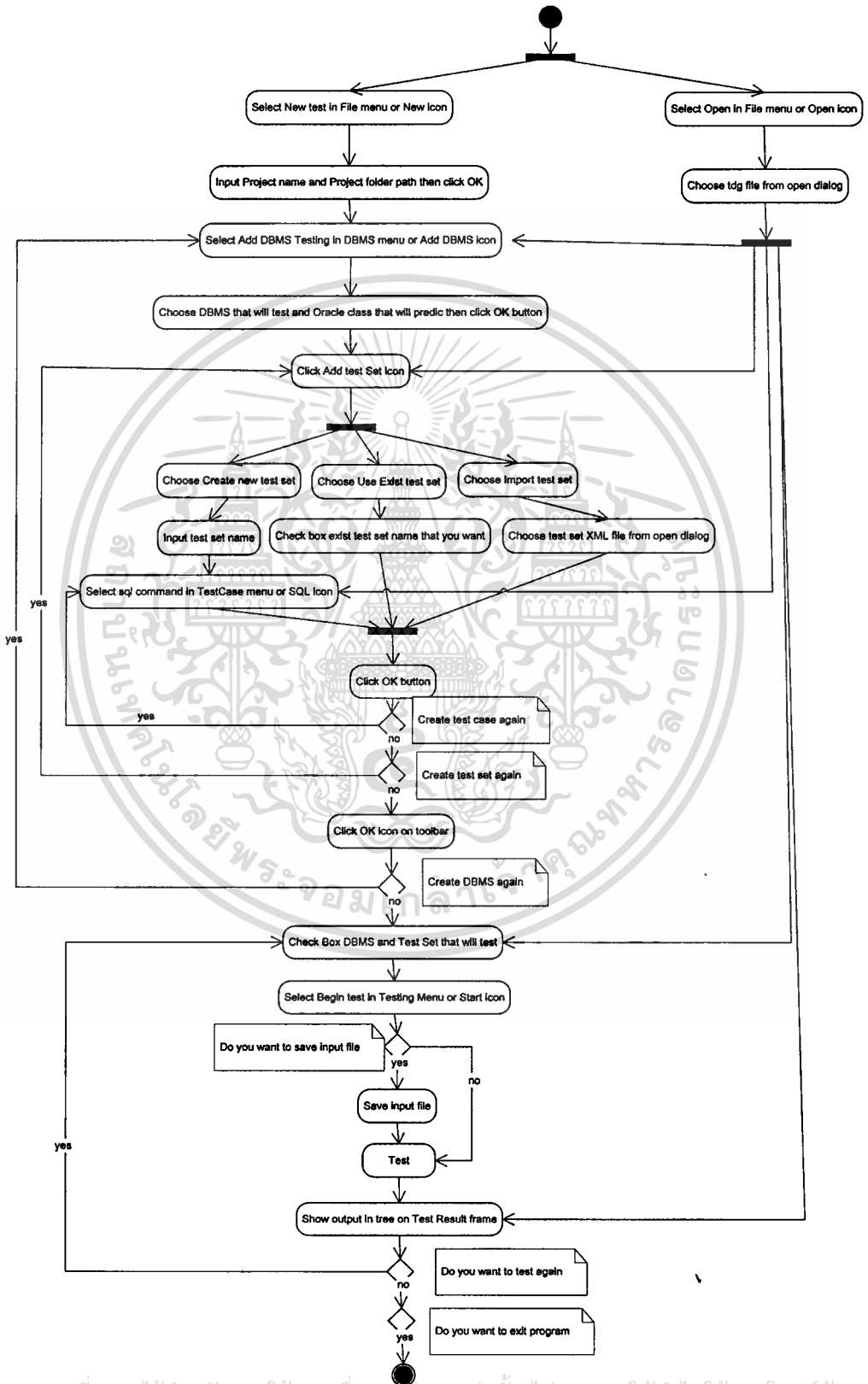
จากรูปที่ 3.31 เป็นการอธิบายถึงลำดับการทำงานของกรลบ Test Case โดย Tester คลิก โหนด Test Case ที่จะลบ แล้วคลิกปุ่ม Remove ซึ่งจะไปลบใน TestSets

3.6 Activity Diagram

Activity Diagram จะมีลักษณะเดียวกับ โพล์ชาร์ต คือ ใช้สำหรับแสดงขั้นตอนการทำงานของระบบ ดังนั้นเราจึงนำมาอธิบายการทำงานทั้งหมดของระบบทดสอบไดร์เวอร์กุก ซึ่งแสดงดังรูปที่ 3.32 ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3.32 Activity Diagram ของระบบทดสอบไดรเวอร์กัญ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การสร้างดีเอ็มแอลทดสอบ และการทำนาย

4.1 การสร้างดีเอ็มแอล

ระบบทดสอบไคร์เวอร์สามารถสร้างดีเอ็มแอลทดสอบโดยอัตโนมัติได้ โดยจะแบ่งกรณีในการสร้างเป็น 2 กรณี คือ Test to Pass และ Test to Fail

4.1.1 Test to Pass

เป็นการสร้างทดสอบเพื่อวัตถุประสงค์ในการทดสอบ DBMS ให้ผ่าน (คาดหวังว่าจะผ่าน) คือ ทดสอบที่อยู่ในโดเมนทั้งหมด โดยดีเอ็มแอลทดสอบ แบ่งตามคำสั่งภาษาสำหรับจัดการข้อมูล ดังนี้

4.1.1.1 SELECT

- SELECT ทุกฟิลด์ที่มีอยู่ในตาราง
- SELECT บางฟิลด์ที่มีอยู่ในตาราง
- SELECT โดยใช้ DISTINCT
- SELECT โดยมีเงื่อนไข WHERE
- SELECT ด้วย Aggregate function เช่น SUM, MIN, MAX, AVG และ COUNT
- SELECT โดยใช้ GROUP BY
- SELECT โดยใช้ HAVING

4.1.1.2 INSERT

- INSERT ในตารางลูกที่มี foreign key อ้างอิงไปตารางหลัก ค่าที่ INSERT ในฟิลด์ที่เป็น foreign key จะต้องตรวจสอบว่าที่ตารางหลักมีค่าที่อ้างอิงแล้ว
- INSERT ในฟิลด์ที่เป็น primary key จะต้องไม่ซ้ำ
- ต้อง INSERT .ค่าในฟิลด์ที่กำหนดว่า NOT NULL
- ชนิดข้อมูลของค่าที่จะ INSERT ต้องเป็นไปตามชนิดข้อมูลที่กำหนดไว้ในฟิลด์ที่จะ INSERT ข้อมูล

4.1.1.3 UPDATE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- UPDATE ฟิวด์ที่เป็น primary key โดยไม่ให้ซ้ำ
- UPDATE ในตารางลูกที่มี foreign key อ้างอิงไปตารางหลัก ค่าที่ UPDATE ใหม่ในฟิวด์ที่เป็น foreign key จะต้องตรวจสอบว่าที่ตารางหลักมีค่าที่อ้างอิงแล้ว
- ชนิดข้อมูลที่ UPDATE ใหม่ต้องเป็นไปตามชนิดข้อมูลที่กำหนดไว้ในฟิวด์ที่จะ UPDATE ข้อมูล
- ถ้ามีการ UPDATE ค่าในตารางหลักในฟิวด์ที่เป็น primary key จะต้องมีการตรวจสอบว่ามีการกำหนดข้อบับบังคับ ดังต่อไปนี้
 - Cascade, Set NULL หรือ Set Default ค่าที่ UPDATE ใหม่เป็นค่าอะไรก็ได้
 - Restrict หรือ ไม่มีการกำหนด กรณีนี้จะไม่สร้างเทสต์เคส

4.1.1.4 DELETE

- ลบแถวในตารางลูก
- ลบแถวในตารางหลัก จะต้องมีการตรวจสอบว่ามีการกำหนดข้อบับบังคับ ดังต่อไปนี้หรือไม่
 - Cascade, Set NULL หรือ Set Default จะ DELETE แถวอะไรก็ได้
 - Restrict หรือ ไม่มีการกำหนด กรณีนี้จะไม่สร้างเทสต์เคส

4.1.2 Test to Fail

เป็นการสร้างเทสต์เคส เพื่อวัตถุประสงค์ในการทดสอบ DBMS ให้ส่งผลกลับออกมา หรือเป็นอินพุตที่ Invalid คือ เทสต์เคสที่อยู่นอกโดเมนทั้งหมด โดยดีเอ็มแอลเทสต์เคสแบ่งตามคำสั่งภาษาสำหรับการจัดการข้อมูล ดังนี้

4.1.2.1 SELECT

- SELECT ฟิวด์ที่ไม่มีอยู่ในตาราง
- SELECT โดยใช้ทั้งชื่อฟิวด์ และ Aggregate function โดยไม่มี GROUP BY เช่น SELECT sale_id, COUNT(sale_values) FROM salesrep;

4.1.2.2 INSERT

- INSERT ฟิวด์ที่เป็น foreign key ในตารางลูก ไม่อ้างอิงไปตารางหลัก
- INSERT ในฟิวด์ที่เป็น primary key ให้มีค่าซ้ำ
- ไม่ INSERT .ค่าในฟิวด์ที่กำหนดว่า NOT NULL

- ชนิดข้อมูลของค่าที่จะ INSERT ต้องไม่เป็นไปตามชนิดข้อมูลที่กำหนดไว้ในฟิลด์ที่จะ INSERT ข้อมูล

4.1.2.3 UPDATE

- UPDATE ฟิลด์ที่เป็น Primary key ในตารางให้ซ้ำ
 - UPDATE ฟิลด์ที่เป็น foreign key ในตารางลูก ไม่อ้างอิงไปตารางหลัก
 - ชนิดข้อมูลที่ UPDATE ใหม่ต้องไม่เป็นไปตามชนิดข้อมูลที่กำหนดไว้ในฟิลด์ที่จะ UPDATE ข้อมูล

- ถ้ามีการ UPDATE ค่าในตารางหลักในฟิลด์ที่เป็น primary key จะต้องมีการตรวจสอบว่ามีการกำหนดข้อบับบังคับ ดังต่อไปนี้หรือไม่

- Restrict หรือ ไม่มีการกำหนด ค่าที่ UPDATE ใหม่เป็นค่าอะไรก็ได้

- Cascade, Set NULL หรือ Default กรณีนี้จะไม่สร้างเทสต์เคส

4.1.2.4 DELETE

- ลบแถวในตารางหลัก จะต้องมีการตรวจสอบว่ามีการกำหนดข้อบับบังคับ ดังต่อไปนี้หรือไม่

- Restrict หรือ ไม่มีการกำหนด จะDELETE แถวอะไรก็ได้

- Cascade, Set NULL หรือ Default กรณีนี้จะไม่สร้างเทสต์เคส

4.2 การทำนาย

คลาสในการทำนาย ในระบบเทสต์ไควร์วอร์กยูจะติดต่อกับคลาส TestDriver โดยผ่าน ICheckDB ซึ่งเป็น Interface ซึ่งมีเมทอดดังนี้

4.2.1 prepareConnection

เป็นเมทอดที่ใช้ในการรับ และเตรียมค่าต่างๆ ในการเชื่อมต่อ เช่น Driver, URL, User และ Password

4.2.2 beforeQuery

เป็นเมทอดที่ใช้ในการรับคำสั่ง SQL จาก TestDriver แล้วมาทำการทำการแยกคำสั่ง SQL เป็นกรณีต่างๆ ดังนี้

- SELECT จะทำการเรียกเมทอด beforeSelect
- INSERT จะทำการเรียกเมทอด beforeInsert

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ UPDATE จะทำการเรียกเมทอด beforeUpdate ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- DELETE จะทำการเรียกเมทริก beforeDelete
- CREATE จะทำการเรียกเมทริก beforeCreate
- DROP จะทำการเรียกเมทริก beforeDrop

4.2.3 beforeSelect

เป็นเมทริกในการทำนายคำสั่ง SELECT โดยจะทำการตรวจสอบตามกฎต่างๆ ดังนี้

กฎที่1 ตารางที่ถูก SELECT จะต้องมียู่ในฐานข้อมูล

กฎที่2 ฟิวล์ที่ถูก SELECT ต้องมียู่ในตารางที่อยู่หลัง FROM

กฎที่3 การ SELECT โดยใช้ทั้งชื่อ ฟิวล์ และ Aggregate function ต้องมี GROUP

BY

หมายเหตุ ถ้าเป็น ODBC จะทำการทำนายโดยการนับจำนวนแถวในตารางแล้วเก็บ ในตัวแปร row โดยจะนำไปใช้ในเมทริก afterSelect เนื่องจากไดร์เวอร์ของ Microsoft Access ไม่สามารถบอกได้ว่าฟิวล์ใดเป็น primary key และ foreign key

4.2.4 beforeInsert

เป็นเมทริกในการทำนายคำสั่ง INSERT โดยจะทำการตรวจสอบตามกฎต่างๆ ดังนี้

กฎที่1 ตารางที่ถูก INSERT จะต้องมียู่ในฐานข้อมูล

กฎที่2 ฟิวล์ที่ถูก INSERT ต้องมียู่ในตารางที่จะเพิ่ม

กฎที่3 ถ้า INSERT ในตารางลูกที่มี foreign key อ้างอิงไปตารางหลัก ค่าที่ INSERT ในฟิวล์ที่เป็น foreign key จะต้องตรวจสอบว่าที่ตารางหลักมีค่าที่อ้างอิงแล้วหรือยัง

กฎที่ 4 การ INSERT ในฟิวล์ที่เป็น primary key จะต้องไม่ซ้ำ

กฎที่ 5 ต้องมีการ INSERT .ค่าในฟิวล์ที่กำหนดว่า NOT NULL

กฎที่ 6 ชนิดข้อมูลของค่าที่จะ INSERT ต้องเป็นไปตามชนิดข้อมูลที่กำหนดไว้ในฟิวล์ที่จะ INSERT ข้อมูล

หมายเหตุ ถ้าเป็น ODBC จะทำการทำนายโดยการนับจำนวนแถวในตารางแล้วเก็บ ในตัวแปร row โดยจะนำไปใช้ในเมทริก afterInsert เนื่องจากไดร์เวอร์ของ Microsoft Access ไม่สามารถบอกได้ว่าฟิวล์ใดเป็น primary key และ foreign key

4.2.5 beforeUpdate

เป็นเมทริกในการทำนายคำสั่ง UPDATE โดยจะทำการตรวจสอบตามกฎต่างๆ

เอกสารนี้ เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กฎที่1 ตารางที่ถูก UPDATE จะต้องมียู่ในฐานข้อมูล

กฎที่2 ฟิวส์ที่ถูก UPDATE ต้องมียู่ในตารางที่จะปรับปรุง

กฎที่3 ถ้า UPDATE ในตารางลูกที่มี foreign key อ้างอิงไปตารางหลัก ค่าที่ UPDATE ใหม่ในฟิวส์ที่เป็น foreign key จะต้องตรวจสอบว่าที่ตารางหลักมีค่าที่อ้างอิงแล้วหรือยัง

กฎที่4 ชนิดข้อมูลที่ UPDATE ใหม่ต้องเป็นไปตามชนิดข้อมูลที่กำหนดไว้ในฟิวส์ที่จะ UPDATE ข้อมูล

กฎที่5 ถ้ามีการ UPDATE ค่าในตารางหลักในฟิวส์ที่เป็น primary key จะต้องมีการตรวจสอบว่ามีการกำหนดข้อบับบังคับ ค้างต่อไปหรือไม่

- Cascade, Set NULL หรือ Default ค่าที่ UPDATE ส่งผลลัพธ์เป็น true
- Restrict หรือ ไม่มีการกำหนด กรณีนี้จะส่งผลลัพธ์เป็น false

กฎที่6 การ UPDATE ฟิวส์ที่เป็น primary key ในตารางลูกจะต้องไม่ซ้ำ

หมายเหตุ ถ้าเป็น ODBC จะทำการทำนายโดยการนับจำนวนแถวในตารางแล้วเก็บ ในตัวแปร row โดยจะนำไปใช้ในเมทริก afterUpdate เนื่องจากไดร์เวอร์ของ Microsoft Access ไม่สามารถบอกได้ว่าฟิวส์ใดเป็น primary key และ foreign key

4.2.6 beforeDelete

เป็นเมทริกในการทำนายคำสั่ง DELETE โดยจะทำการตรวจสอบตามกฎต่างๆ ดังนี้

กฎที่1 ตารางที่ถูก DELETE จะต้องมียู่ในฐานข้อมูล

กฎที่ 2 ลบแถวในตารางหลัก จะต้องมีการตรวจสอบว่ามีการกำหนดข้อบับบังคับ ค้างต่อไปหรือไม่

- Cascade, Set NULL หรือ Default จะDELETE แถวอะไรก็ได้ ส่งผลลัพธ์เป็น true

- Restrict หรือ ไม่มีการกำหนด กำหนด กรณีนี้จะส่งผลลัพธ์เป็น false

หมายเหตุ ถ้าเป็น ODBC จะทำการทำนายโดยการนับจำนวนแถวในตารางแล้วเก็บ ในตัวแปร row โดยจะนำไปใช้ในเมทริก afterDelete เนื่องจากไดร์เวอร์ของ Microsoft Access ไม่สามารถบอกได้ว่าฟิวส์ใดเป็น primary key และ foreign key

4.2.7 beforeCreate

เป็นเมทริกในการทำนายคำสั่ง CREATE โดยจะทำการตรวจสอบตามกฎต่างๆ ดังนี้

กฎที่1 ตารางที่ถูก CREATE จะต้องไม่มีอยู่ฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของ บริษัท ทรู คอร์ปอเรชั่น จำกัด (มหาชน) ให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กฎที่2 ทำการนับจำนวนตารางที่อยู่ในฐานข้อมูล แล้วเก็บในตัวแปร row โดยจะนำไปใช้ในเมทอด afterCreate

4.2.8 beforeDrop

เป็นเมทอดในการทำนายคำสั่ง DROP โดยจะทำการตรวจสอบตามกฎต่างๆดังนี้
กฎที่1 ตารางที่ถูก DROP จะต้องมียู่ฐานข้อมูล

กฎที่2 ลบตารางที่เป็นตารางหลักจะต้องลบตารางลูกก่อน(ODBC จะไม่ตรวจสอบในกฎนี้)

กฎที่3 ทำการนับจำนวนตารางที่อยู่ในฐานข้อมูล แล้วเก็บในตัวแปร row โดยจะนำไปใช้ในเมทอด afterDrop

4.2.9 afterQuery

เป็นเมทอดทำการส่งผลลัพธ์ True หรือ False และ ExpectedError ที่ได้จากการตรวจสอบตามกฎต่างๆ ใน beforeQuery .ให้ TestDriver ผ่านอินเตอร์เฟส ICheckDB แต่ถ้าเป็น ODBC driver จะต้องทำการนำค่าในตัวแปร row มาเปรียบเทียบก่อน โดยจะแยกตามคำสั่งดังนี้ที่ใช้ทดสอบคำสั่ง SQL ตามกฎต่างๆ ซึ่งขึ้นอยู่กับคำสั่งนั้นๆ ดังนี้

- SELECT จะทำการเรียกเมทอด afterSelect
- INSERT จะทำการเรียกเมทอด afterInsert
- UPDATE จะทำการเรียกเมทอด afterUpdate
- DELETE จะทำการเรียกเมทอด afterDelete
- CREATE จะทำการเรียกเมทอด afterCreate
- DROP จะทำการเรียกเมทอด afterDrop

4.2.10 afterSelect

เป็นเมทอดที่ทำการนับจำนวนแถวหลังจากที่ DBClient ทำการทดสอบจริง มาเก็บในตัวแปร after แล้วนำค่าในตัวแปร row ที่ได้จาก beforeSelect มาเปรียบเทียบกับ after ดังนี้

- row เท่ากับ after ส่งกลับเป็น True
- row ไม่เท่ากับ after ส่งกลับเป็น False

4.2.11 afterInsert

เป็นเมทอดที่ทำการนับจำนวนแถวหลังจากที่ DBClient ทำการทดสอบจริง มาเก็บในตัวแปร after แล้วนำค่าในตัวแปร row ที่ได้จาก beforeInsert มาเปรียบเทียบกับ after ดังนี้

- row น้อยกว่า after ส่งกลับเป็น False

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ... นอกนั้น ส่งกลับเป็น True... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า... ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.12 afterUpdate

เป็นเมทริกซ์ที่ทำการนับจำนวนแถวหลังจากที่ DBClient ทำการทดสอบจริง มาเก็บในตัวแปร after แล้วนำค่าในตัวแปร row ที่ได้จาก beforeUpdate มาเปรียบเทียบกับ after ดังนี้

- row เท่ากับ after ส่งกลับเป็น True
- row ไม่เท่ากับ after ส่งกลับเป็น False

4.2.13 afterDelete

เป็นเมทริกซ์ที่ทำการนับจำนวนแถวหลังจากที่ DBClient ทำการทดสอบจริง มาเก็บในตัวแปร after แล้วนำค่าในตัวแปร row ที่ได้จาก beforeDelete มาเปรียบเทียบกับ after ดังนี้

- row มากกว่า after ส่งกลับเป็น True
- นอกนั้น ส่งกลับเป็น False

4.2.14 afterCreate

เป็นเมทริกซ์ที่ทำการนับจำนวนตารางหลังจากที่ DBClient ทำการทดสอบจริง มาเก็บในตัวแปร after แล้วนำค่าในตัวแปร row ที่ได้จาก beforeCreate มาเปรียบเทียบกับ after ดังนี้

- row น้อยกว่า after ส่งกลับเป็น True
- นอกนั้น ส่งกลับเป็น False

4.2.15 afterDrop

เป็นเมทริกซ์ที่ทำการนับจำนวนตารางหลังจากที่ DBClient ทำการทดสอบจริง มาเก็บในตัวแปร after แล้วนำค่าในตัวแปร row ที่ได้จาก beforeDrop มาเปรียบเทียบกับ after ดังนี้

- row มากกว่า after ส่งกลับเป็น True
- นอกนั้น ส่งกลับเป็น False

4.3 ข้อจำกัดของการสร้างดีเอ็มแอลเทสต์เคส

4.3.1 ไม่สามารถสร้างดีเอ็มแอลเทสต์เคสแบบอัตโนมัติ ที่ติดต่อกับ ODBC ได้เนื่องจากไคร์เวอร์ไม่สนับสนุนในการบอกว่าฟิลด์ใดเป็น primary key หรือ foreign key

4.3.2 .ในการสร้างดีเอ็มแอลเทสต์เคสแบบอัตโนมัติ จะสนับสนุนชนิดข้อมูล ดังต่อไปนี้ Integer, Varchar, Char, Bit, Real และ DateTime เนื่องจากในการประกาศชนิดข้อมูลเหล่านี้เหมือนกัน ซึ่งนอกนั้นจะมีวิธีการเขียนที่แตกต่างกัน

4.4 ข้อจำกัดของการทำนาย

4.4.1 ในการทำนาย ระบบฐานข้อมูลที่ติดต่อ ODBC จะให้คำทำนายที่ไม่ละเอียด หรืออาจให้ข้อผิดพลาดได้เนื่องจาก ไดรเวอร์ไม่สนับสนุนในการบอกว่าฟิลด์ใดเป็น primary key หรือ foreign key

4.4.2 ในการทำนายคำสั่ง SQL ที่ซับซ้อน เช่น มีคำสั่งซ้อนกัน อาจให้คำทำนายที่ผิดพลาดได้เนื่องจากในการพัฒนาระบบทดสอบไดรเวอร์ เป็นช่วงในการเริ่มพัฒนาจึงต้องพัฒนาให้ทำนายให้ครอบคลุมคำสั่งที่ไม่ซับซ้อนก่อน แล้วจึงค่อยๆ ขยายให้ครอบคลุมทั้งคำสั่ง SQL

4.4.3 ในการทำนายคำสั่ง CREATE อาจให้คำทำนายที่ผิดพลาดได้เนื่องจาก คำสั่ง CREATE มีไวยากรณ์ในการเขียนที่หลากหลาย

4.4.4 ในการทำนายคำสั่ง จะสมมติฐานว่า คำสั่งนั้นมีไวยากรณ์ที่ถูกต้อง แล้วเนื่องจากโมดูลในการตรวจสอบไวยากรณ์คำสั่ง SQL นั้นมีความซับซ้อนมากเพราะจะต้องตรวจสอบไวยากรณ์คำสั่ง SQL ได้ทุกๆ ระบบฐานข้อมูล ซึ่งจะต้องใช้ในการพัฒนา

บทที่ 5

การพัฒนาระบบ









การออกแบบหน้าจอของระบบทดสอบไควร์รีภาษาที่ใช้จัดการฐานข้อมูล (Data Manipulation Language -DML) สามารถอธิบายในรายละเอียดโดยเริ่มตั้งแต่หน้าจอหลักของระบบ และหน้าจอแบบฟอร์มการเข้าใช้งานในเมนูต่างๆได้ ดังนี้






5.1 หน้าจอหลัก (Main Window)

ตัวประสานกราฟิกกับผู้ใช้ หรือหน้าจอหลักของระบบทดสอบไควร์รีประกอบด้วย 4 ส่วนดังนี้

5.1.1 Toolbar

เป็นส่วนที่ประกอบด้วยไอคอนต่างๆ เพื่ออำนวยความสะดวกในการใช้งาน
ตารางที่ 5.1 รายละเอียดไอคอนบนทูลบาร์

รูปไอคอน	คำอธิบาย
	ทำการสร้างชุดทดสอบ DBMS ใหม่
	เปิดเพิ่มอินพุตชุดทดสอบที่เคยสร้างไว้แล้วเป็นชนิด XML
	บันทึกเพิ่ม
	ลบ DBMS, TestSet และ TestCase
	เริ่มทำการทดสอบ
	ดูรายงานผลการทดสอบแบบตาราง
	ทำการเพิ่ม DBMS (ชุดทดสอบแต่ละ DBMS)
	ทำการเพิ่มชุด Test Set(หนึ่ง DBMS มีได้หลาย ชุด Test Set)

	<p>ทำการเพิ่ม Test Case (คำสั่ง SQL)ซึ่งมีตัวเลือกดังนี้(หนึ่ง Test Set มีได้หลาย Test Case)</p> <ul style="list-style-type: none"> - Create table - Drop table - Select - Insert - Update - Delete - Generate
	<p>ทำการยืนยันเมื่อสร้างชุดทดสอบเรียบร้อยแล้ว</p>
	<p>เลือกเมื่อต้องการ ให้บันทึกอินพุต และเอาต์พุต แบบ DTD</p>
	<p>เลือกเมื่อต้องการ ให้บันทึกอินพุต และเอาต์พุต แบบ XML Schemas</p>
	<p>Help</p>

5.1.2 Main

เป็นส่วนที่ให้ผู้ทดสอบใส่รายละเอียดของ DBMS และ SQL Command ต่างๆที่จะทดสอบ

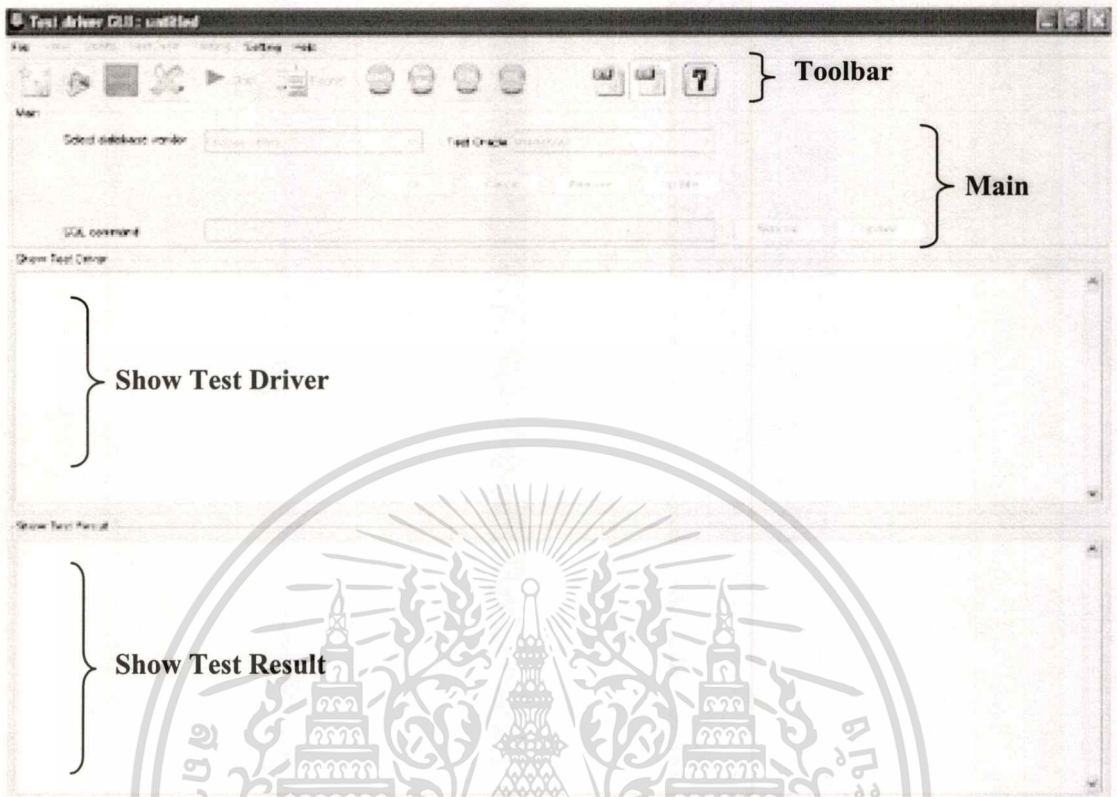
5.1.3 Show Test Driver

เป็นส่วนแสดงชุดทดสอบ DBMS ที่ผู้ใช้สร้างขึ้น โดยจะแสดงในแบบต้นไม้

5.1.4 Show Test Result

เป็นส่วนแสดงผลลัพธ์ชุดทดสอบ DBMS ที่ผู้ใช้สร้างขึ้น โดยจะแสดงในแบบต้นไม้

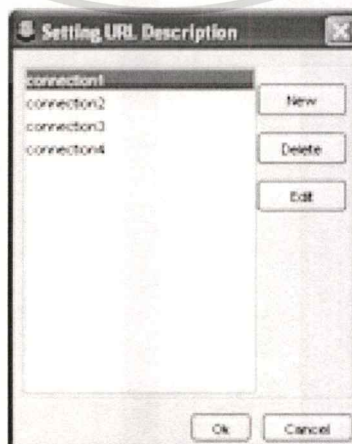
โดยหน้าจอหลักจะแสดงดังรูป 5.1



รูปที่ 5.1 หน้าจอหลักของระบบทดสอบไดรเวอร์ก्यू

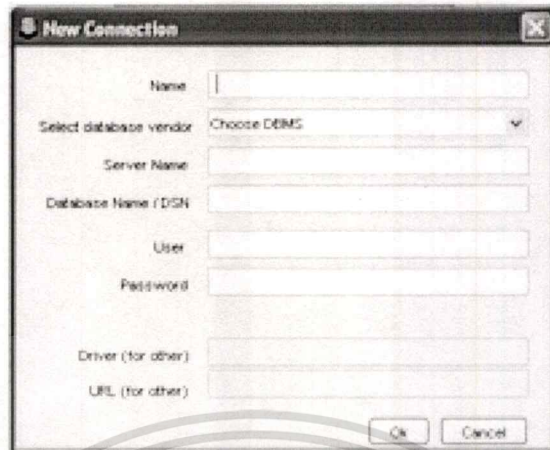
5.2 เมนูกำหนดค่าติดต่อ DBMS ที่จะทดสอบ (Setting Connection)

ในการใช้ระบบทดสอบไดรเวอร์ก्यूจำเป็นต้องกำหนดค่าต่างๆ ในการติดต่อกับ DBMS ที่จะทดสอบ โดยผู้ทดสอบเลือก Connection ในเมนู Setting จะปรากฏ Setting URL Description dialog ซึ่งแสดงรายชื่อการติดต่อที่เราสร้างขึ้นมาดังรูป ที่ 5.2 ซึ่งผู้ทดสอบสามารถเพิ่ม ลบ หรือ แก้ไขการติดต่อได้โดยกดปุ่ม New Delete หรือ Edit ตามลำดับ ดังรูปที่ 5.3



รูปที่ 5.2 ไดอะล็อกแสดงรายชื่อการติดต่อที่สร้างขึ้น

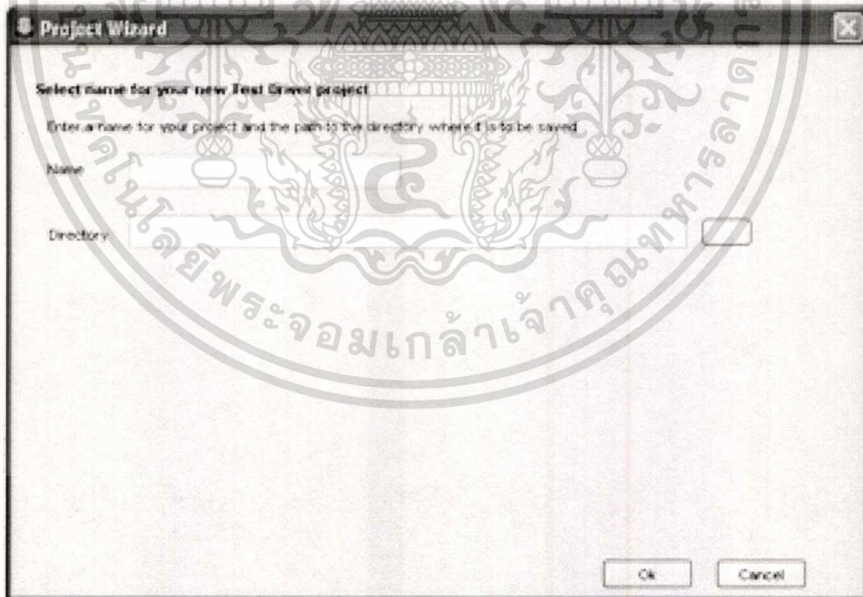
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 ไดอะล็อกบ็อกซ์ และแสดงค่าต่างๆในการติดต่อ DBMS

5.3 เมนูสร้างโครงการใหม่ (New test...)

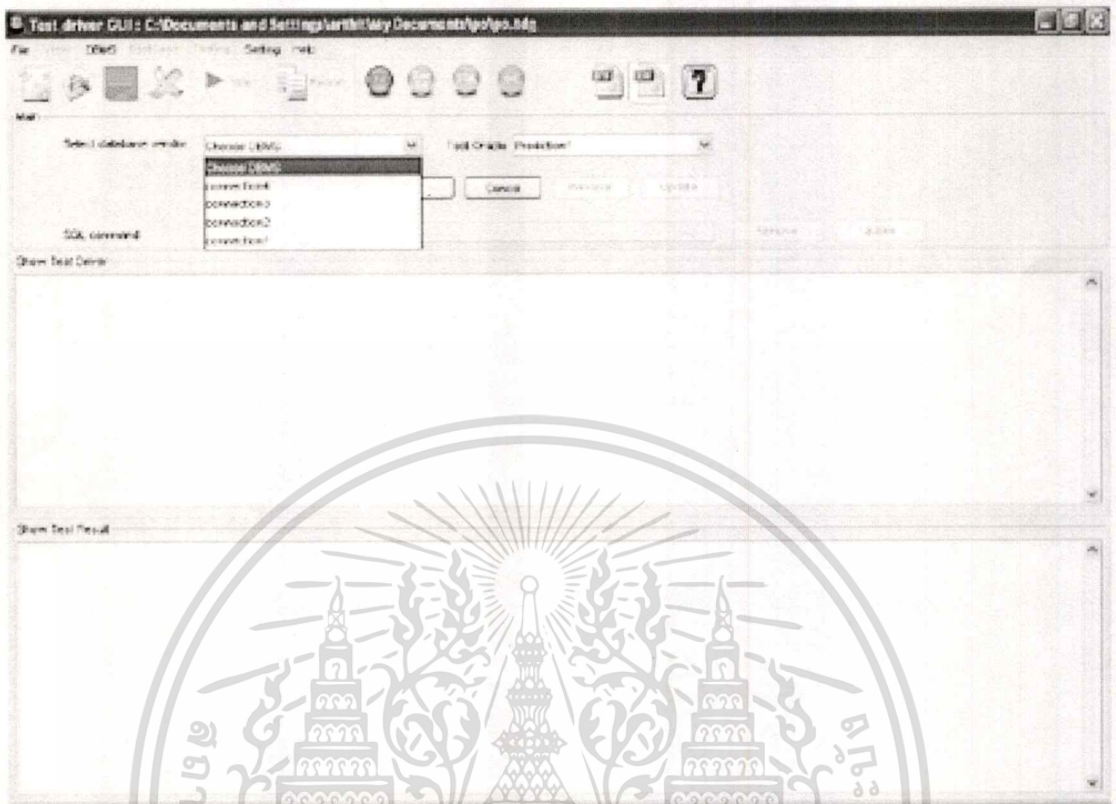
เมื่อผู้ทดสอบต้องการสร้างโครงการใหม่ ผู้ทดสอบจะเลือก New test ในเมนู File หรือเลือก New icon บน ทูลบาร์ จะปรากฏ dialog ให้ใส่ชื่อโครงการ และ พาทที่เก็บโครงการ โดยแสดงดังรูป 5.2



รูปที่ 5.4 หน้าจอเมนูสร้างโครงการใหม่

5.4 เมนูเพิ่ม DBMS ที่จะทดสอบ

เมื่อผู้ทดสอบต้องการเพิ่ม DBMS ที่จะทดสอบ ผู้ทดสอบจะเลือก Add DBMS Testing ใน DBMS menu หรือเลือก Add DBMS icon จาก Toolbar ทำให้ใน Main Panel สามารถทำงานได้ดังเอกสารรูปที่ 5.5 เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 หน้าจอเมนูเพิ่ม DBMS ที่จะทดสอบ

5.5 เมนูเพิ่ม Test Set ที่จะทดสอบ

เมื่อผู้ทดสอบต้องการเพิ่ม Test Set ที่จะทดสอบ ผู้ทดสอบจะเลือก Add Test Set icon จาก ทูลบาร์ (ถ้าเป็นการแทรก Test Set ให้คลิกไปยัง โหนด Test Set แล้ว Test Set icon จะสามารถเลือก ได้) จะปรากฏ dialog ให้เลือกว่าจะสร้าง Test Set แบบไหน ดังแสดงดังรูปที่ 5.6

- ถ้าเลือก Create new test set แล้ว dialog จะให้ใส่ชื่อ Test Set ดังแสดงดังรูปที่ 5.7 เมื่อเสร็จสิ้นจะได้หน้าจอแสดงดังรูปที่ 5.8 ขั้นตอนต่อไปจะเป็นการสร้าง Test Case แต่ จากรูปที่ 5.7 ถ้าคลิกเลือก Create new test set by copy exist test set จะได้ไดอะล็อก แสดงรายชื่อ Test Set ที่มีอยู่ในโครงการที่จะทำการคัดสำเนา ดังแสดงดังรูปที่ 5.9
- ถ้าเลือก Use Exist test set แล้ว dialog จะให้เลือกชื่อ Test Set ที่มีอยู่ในโครงการ ดังแสดงดังรูปที่ 5.9
- ถ้าเลือก Import test set แล้ว Open dialog จะให้เลือกไฟล์ test set xml ดังแสดง ดังรูปที่ 5.10

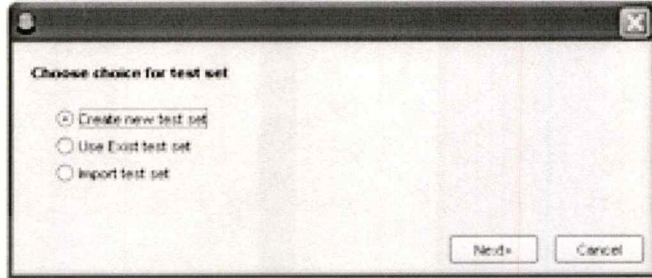
หมายเหตุ ถ้าเลือก Create new by copy ,Use Exist test set หรือ Import test set

เมื่อเสร็จสิ้นจะได้หน้าจอ ดังแสดงดังรูปที่ 5.11 ซึ่งผู้ทดสอบสามารถเพิ่ม Test Set ได้อีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปเผยแพร่โดยไม่ได้รับอนุญาตเป็นการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

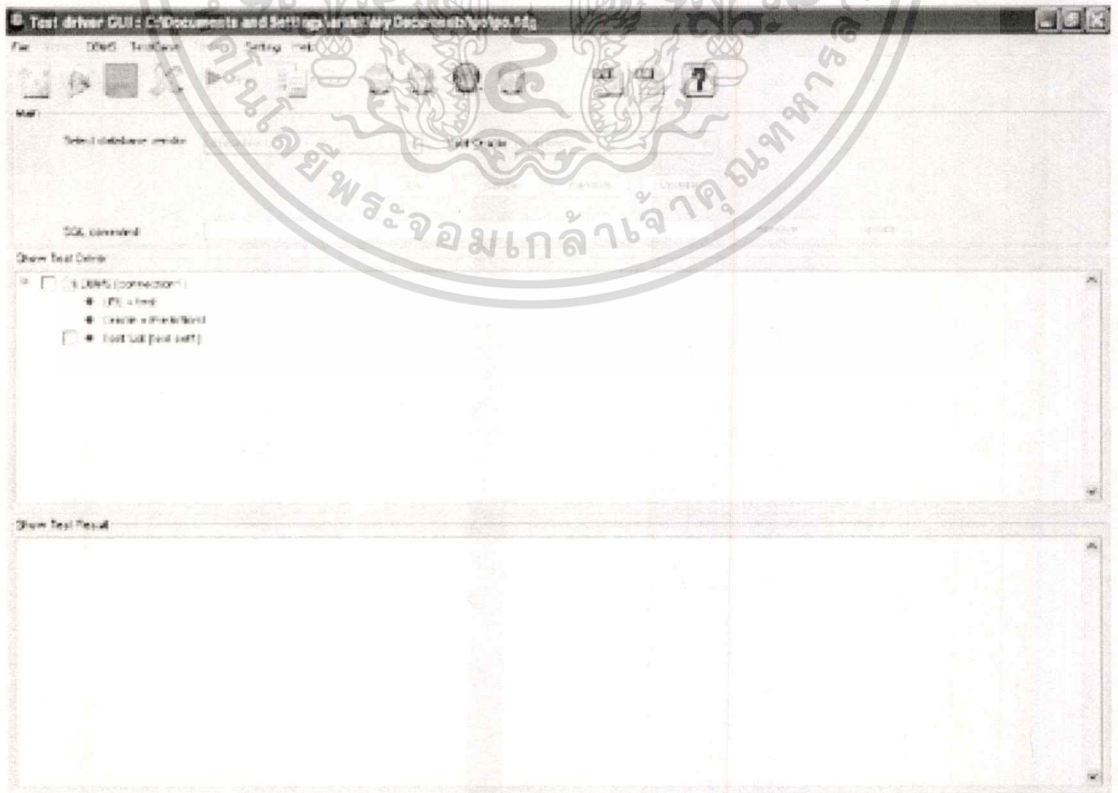
หากไม่ต้องการแล้วจะเป็นการยืนยันในการสร้าง Test Set เสร็จสิ้น โดยการคลิกปุ่ม Ok บน ทูลบาร์



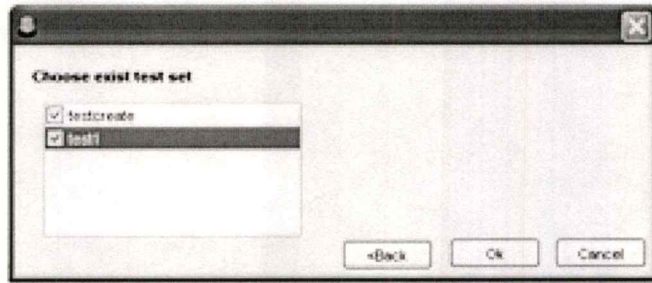
รูปที่ 5.6 โค้ดคลิกให้เลือกในการสร้าง Test Set ใหม่



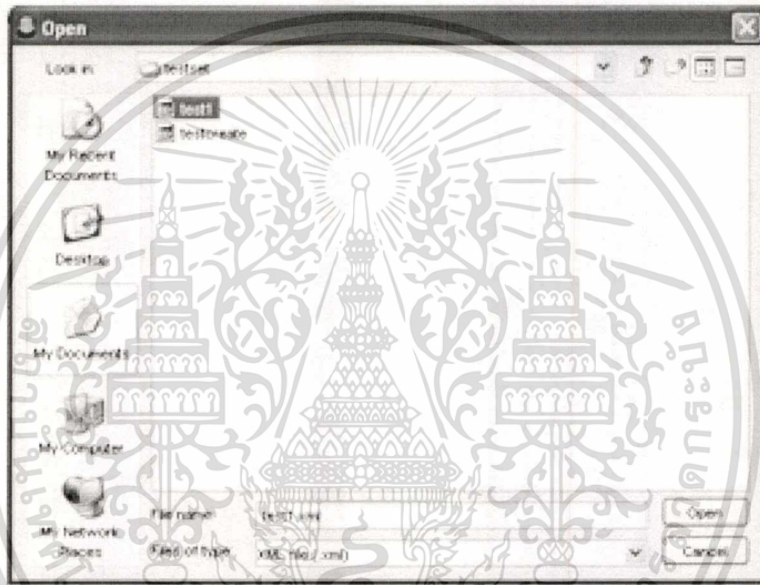
รูปที่ 5.7 โค้ดคลิกใส่ชื่อ Test Set ใหม่



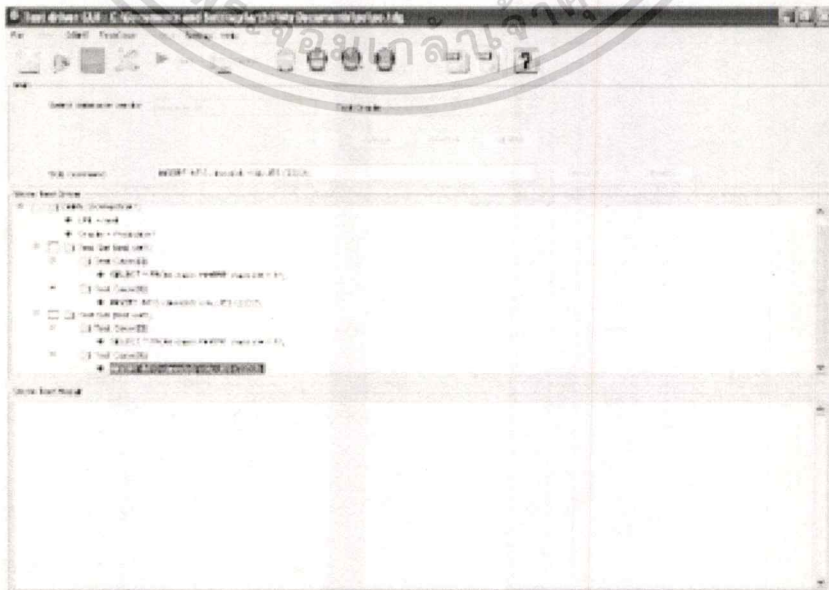
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้การอ้างอิงร่วมกันเท่านั้น ไม่ควรเอาไปใช้ประโยชน์ด้านการค้า
รูปที่ 5.8 หน้าจอหลังจากเลือกสร้าง Test Set ใหม่
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.9 โค้ดเลือกเลือก Test Set ที่มีอยู่ในโครงการงาน



รูปที่ 5.10 โค้ดเลือกเปิดเลือกไฟล์ Test Set XML



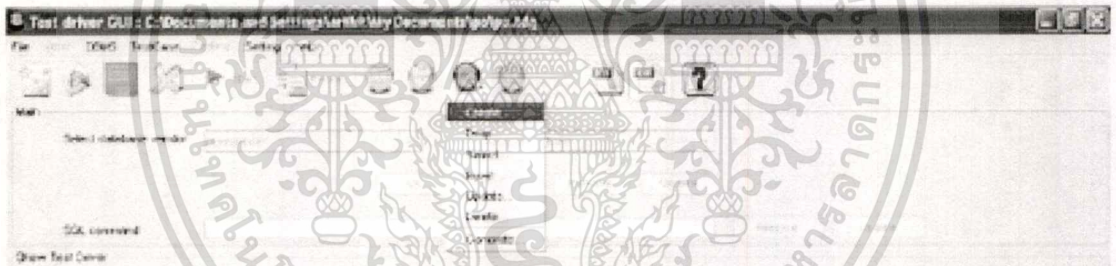
เอกสารนี้เป็นเอกสารรูปที่ 5.11 หน้าจอหลังจากเลือกสร้างใช้ Test Set เดิมหรือนำเข้า Test Set โดยขั้นตอนการดำเนินการ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.6 เมนูเพิ่ม Test Case ที่จะทดสอบ

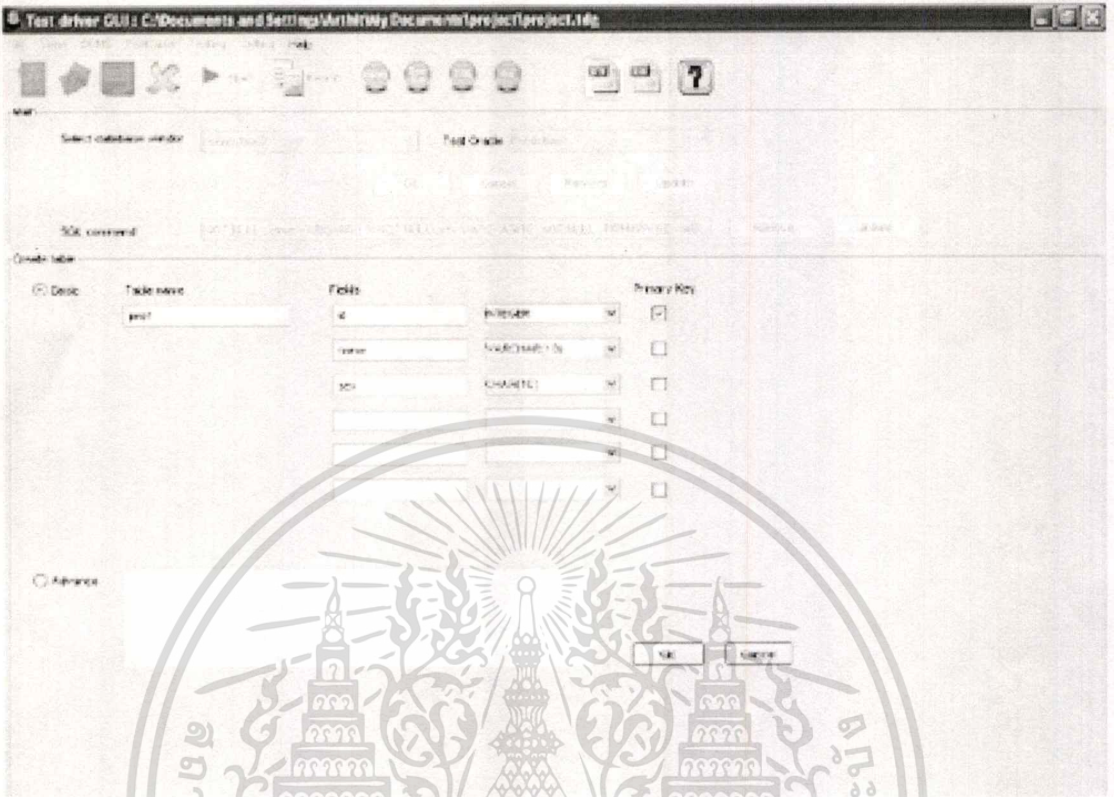
เมื่อผู้ทดสอบต้องการเพิ่ม Test Case ที่จะทดสอบ ผู้ทดสอบจะเลือกประเภท Test Case ที่อยู่ใน TestCase menu หรือเลือก SQL icon จากทูลบาร์ดังแสดงดังรูปที่ 5.12

- เลือก Create จะปรากฏหน้าจอแสดงดังรูปที่ 5.13
- เลือก Drop จะปรากฏหน้าจอแสดงดังรูปที่ 5.14
- เลือก Select จะปรากฏหน้าจอแสดงดังรูปที่ 5.15
- เลือก Insert จะปรากฏหน้าจอแสดงดังรูปที่ 5.16
- เลือก Update จะปรากฏหน้าจอแสดงดังรูปที่ 5.17
- เลือก Delete จะปรากฏหน้าจอแสดงดังรูปที่ 5.18
- เลือก Generate จะปรากฏหน้าจอแสดงดังรูปที่ 5.19 เมื่อสร้าง Test Case

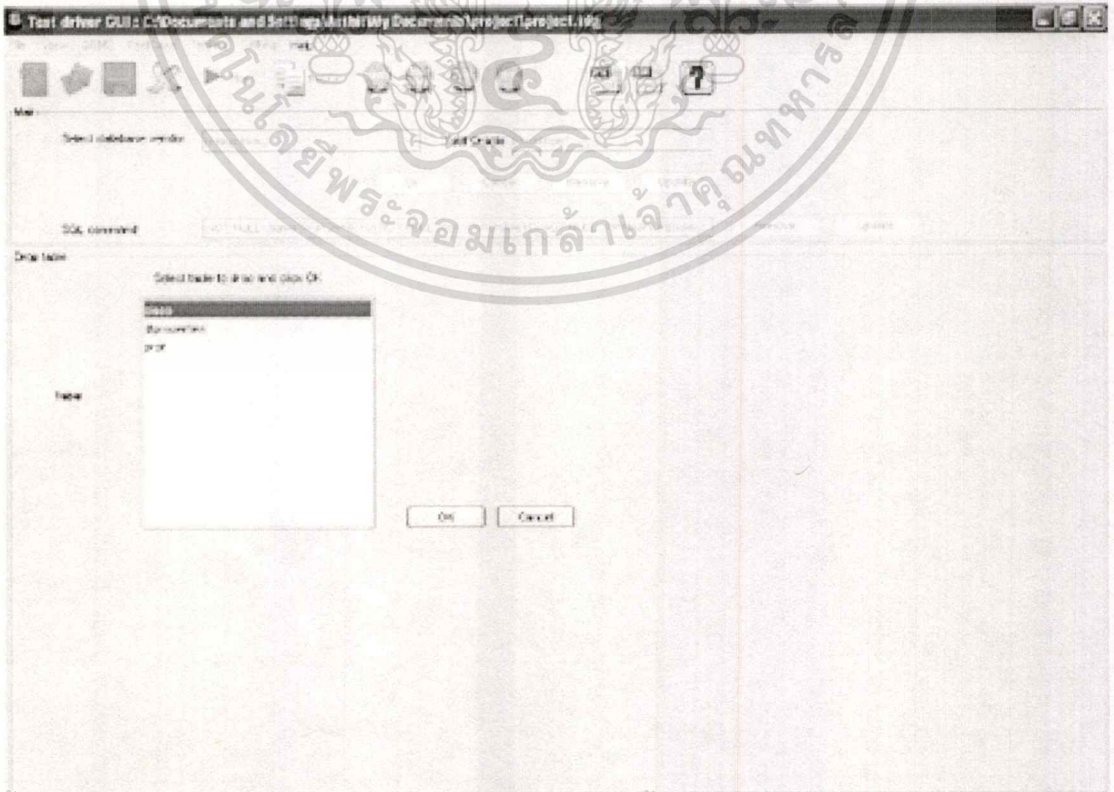
เรียบร้อยแล้วจะปรากฏหน้าจอแสดงรายชื่อ Test Case ที่ generate โดยผู้ทดสอบสามารถเลือก Test Case ที่ต้องการทดสอบได้ ดังแสดงดังรูปที่ 5.20



รูปที่ 5.12 หน้าจอแสดงเมนูย่อยของ Test Case

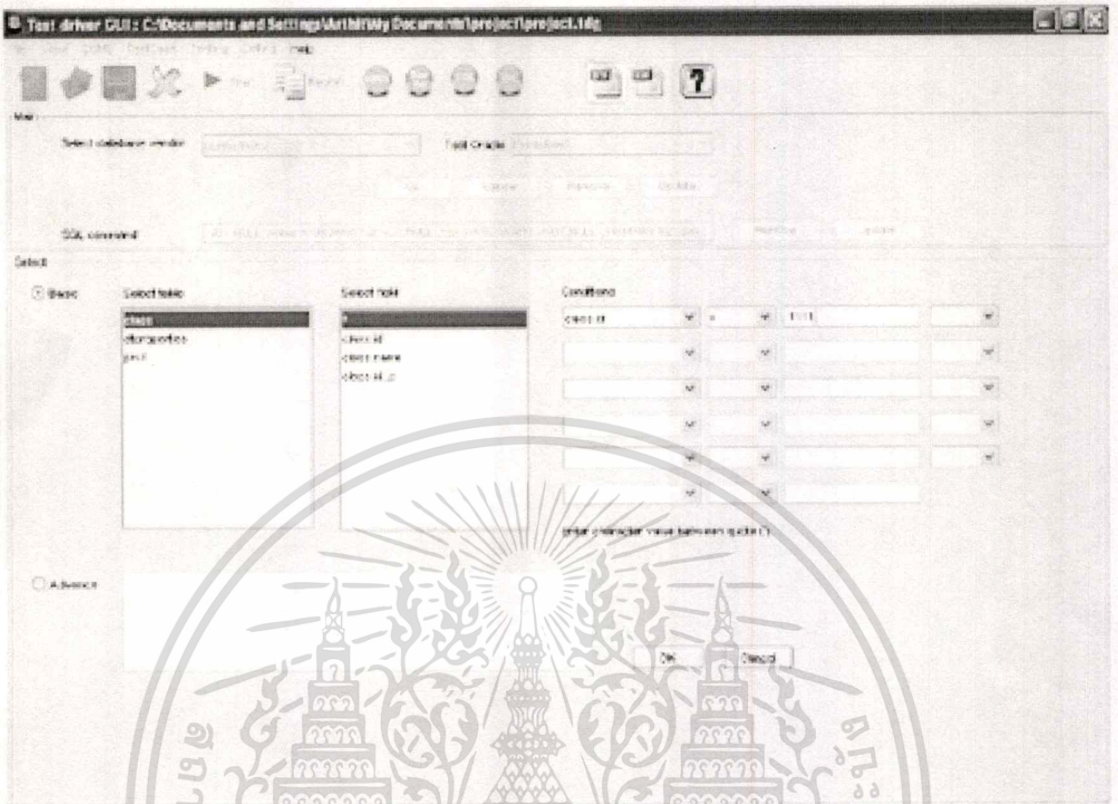


รูปที่ 5.13 หน้าจอเมื่อเลือกเมนูย่อย Create

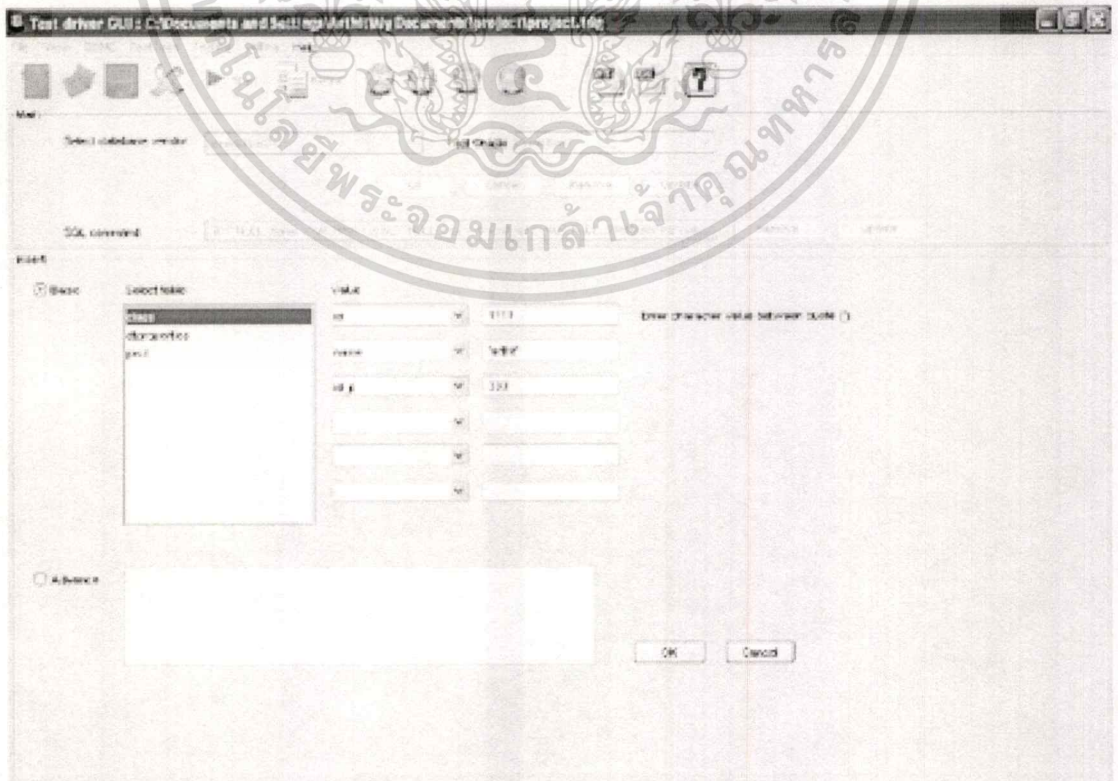


รูปที่ 5.14 หน้าจอเมื่อเลือกเมนูย่อย Drop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับองค์ความรู้ที่เผยแพร่โดยไม่จำกัดให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

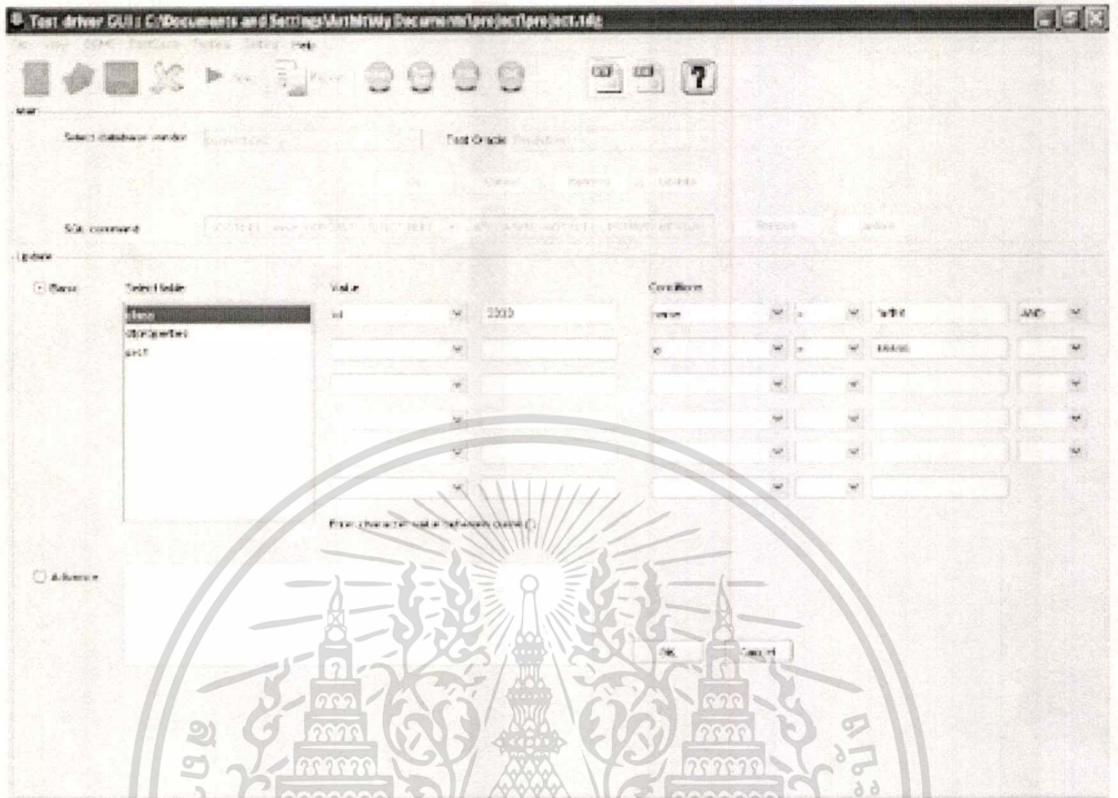


รูปที่ 5.15 หน้าจอเมื่อเลือกเมนูย่อย Select

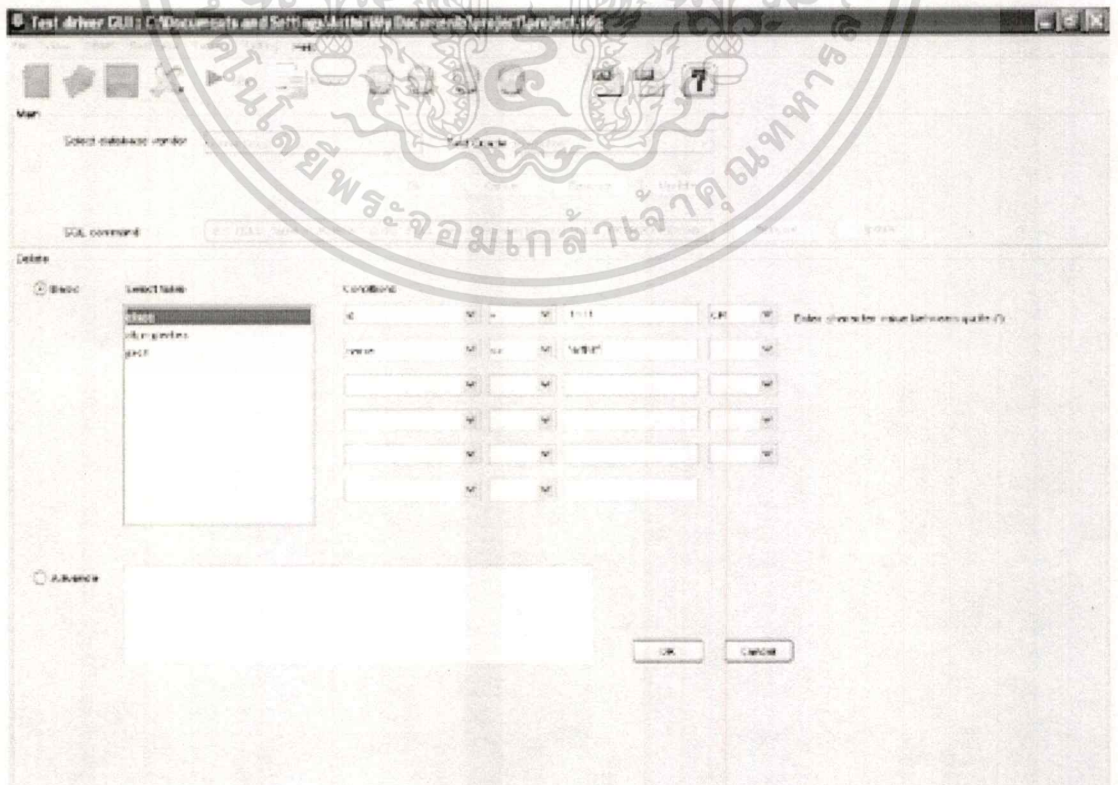


รูปที่ 5.16 หน้าจอเมื่อเลือกเมนูย่อย Insert

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคลากรในหน่วยงานเพื่อการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

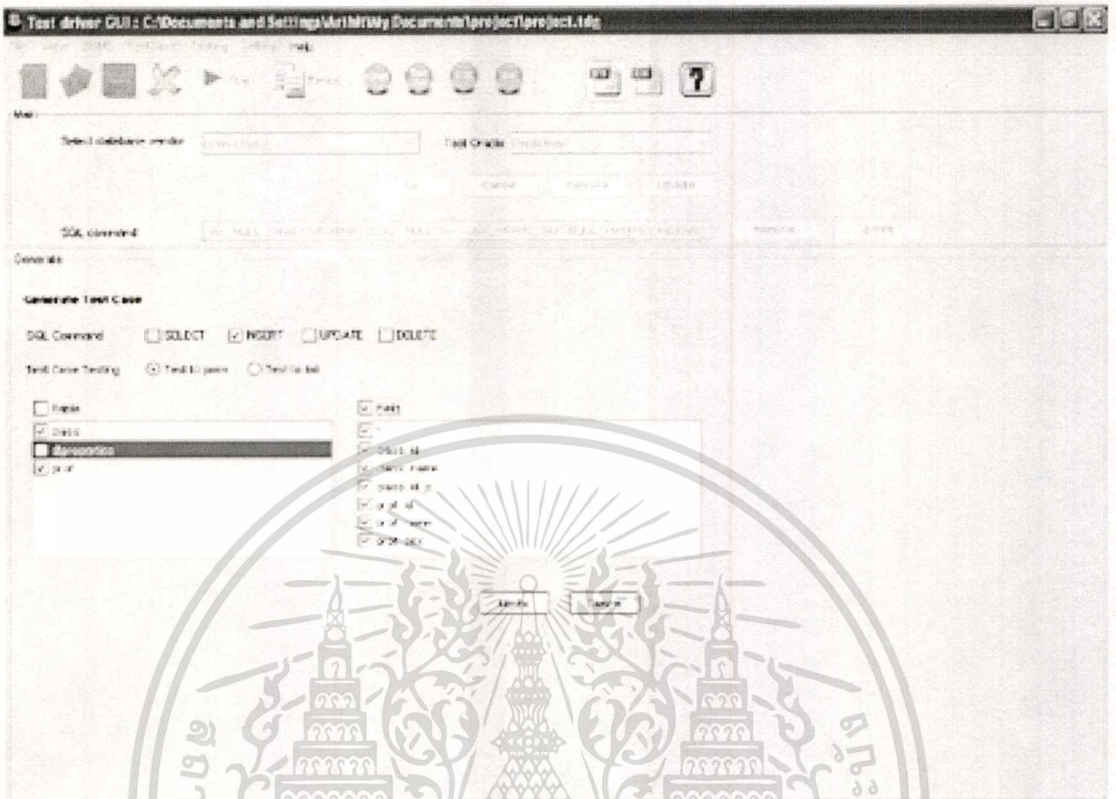


รูปที่ 5.17 หน้าจอเมื่อเลือกเมนูย่อย Update

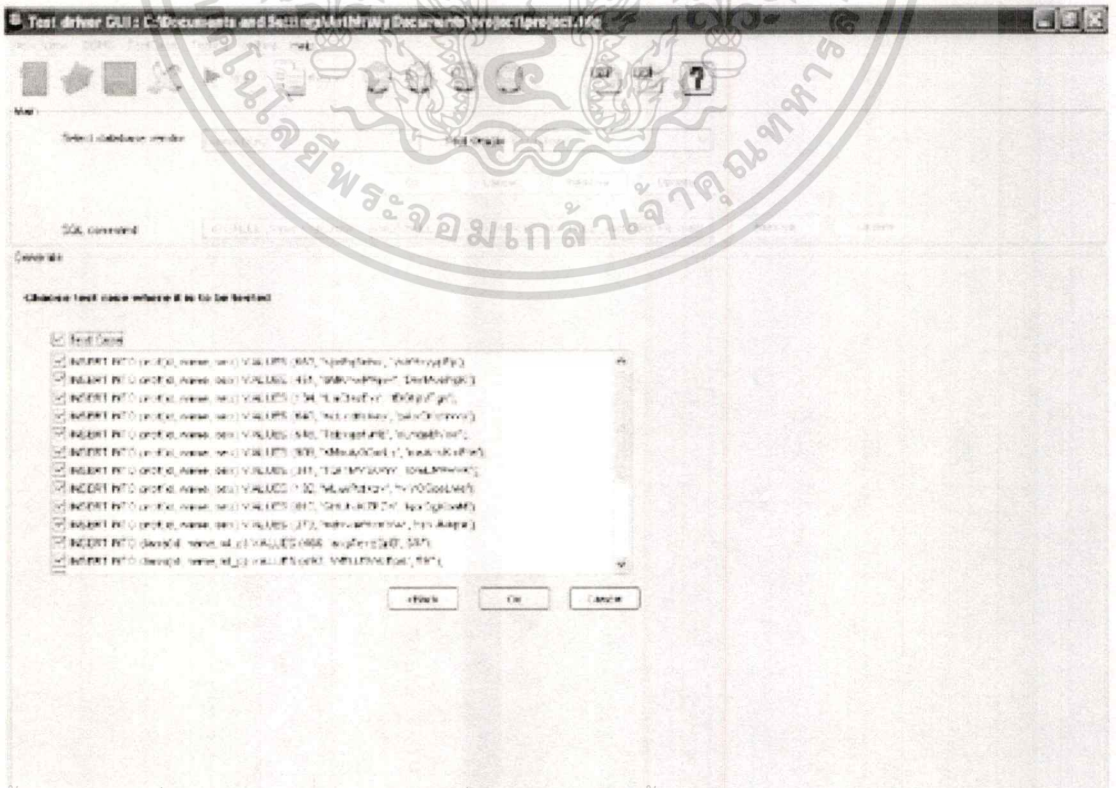


รูปที่ 5.18 หน้าจอเมื่อเลือกเมนูย่อย Delete

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 5.18 หน้าจอเมื่อเลือกเมนูย่อย Delete ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



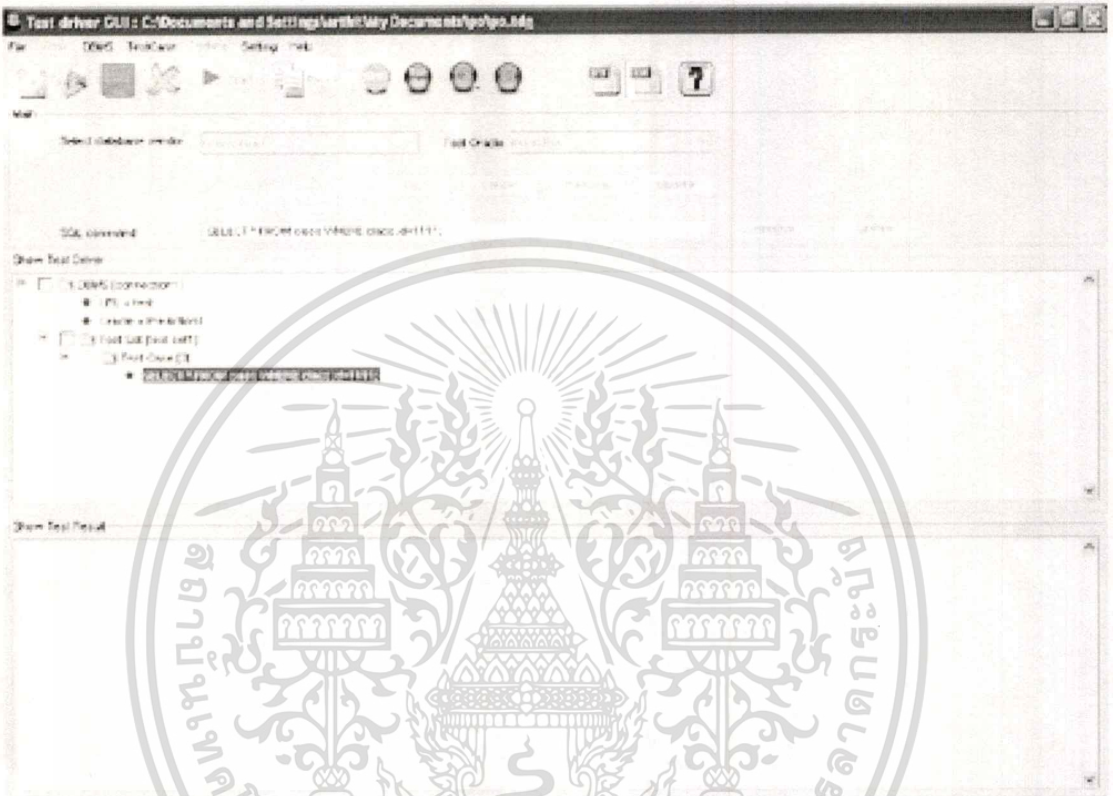
รูปที่ 5.19 หน้าจอเมื่อเลือกเมนูย่อย Generate



รูปที่ 5.20 หน้าจอแสดง Test Case ที่ Generate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

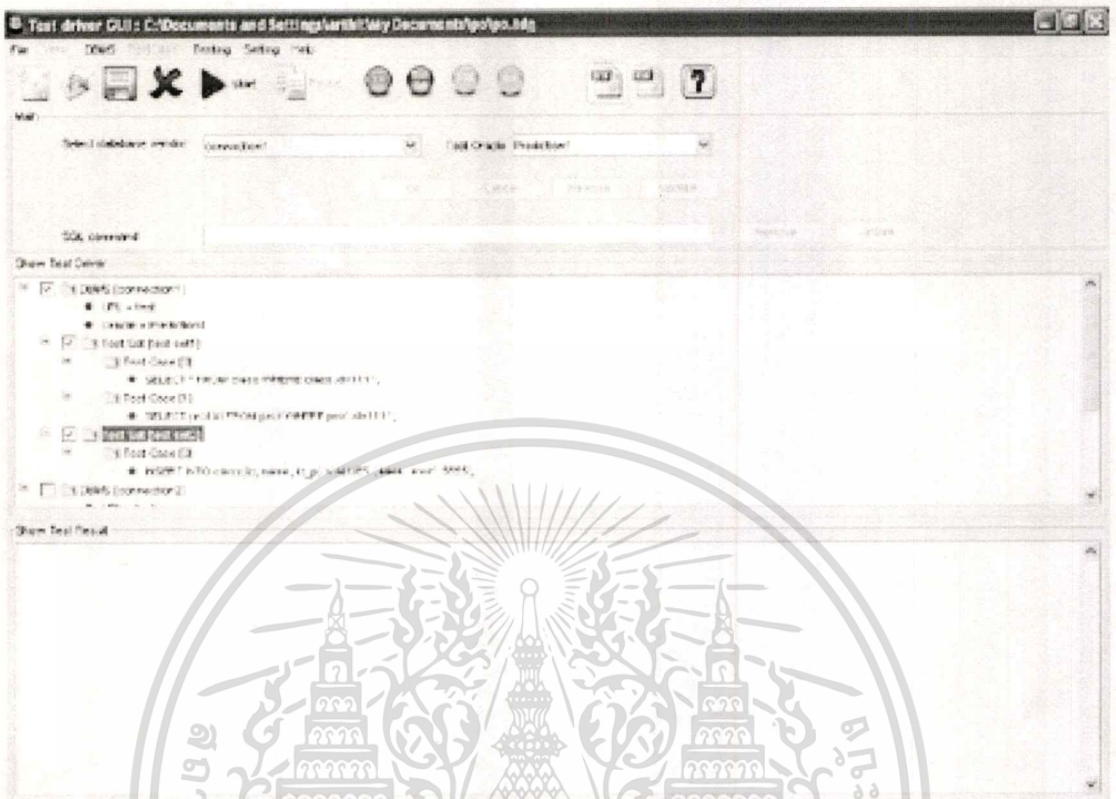
เมื่อเสร็จสิ้นจะได้หน้าจอ ดังแสดงดังรูปที่ 5.21 ซึ่งผู้ทดสอบสามารถเพิ่ม Test Case ได้อีก หากไม่ต้องการแล้วจะเป็นการยืนยันในการสร้าง Test Case เสร็จสิ้นโดยการคลิกปุ่ม Ok บนทูลบาร์



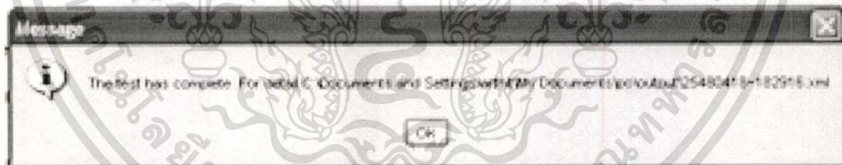
รูปที่ 5.21 หน้าจอหลังจากสร้าง Test Case เรียบร้อยแล้ว

5.7 เมนูทดสอบ

เมื่อผู้ทดสอบต้องการทดสอบ ผู้ทดสอบจะเลือก DBMS และ Test Set ที่ต้องการทดสอบ โดยการกดเลือกที่ Check Box ในต้นไม้ที่แสดงในส่วน Show Test Driver แล้ว เลือก Begin test ในเมนู Testing หรือ Start icon ใน Toolbar ดังแสดงดังรูปที่ 5.22 เมื่อทดสอบเรียบร้อยแล้วจะแจ้งให้ผู้ทดสอบทราบ ดังแสดงดังรูปที่ 5.23 และแสดงผลทดสอบครั้งสุดท้าย ดังแสดงรูปที่ 5.24



รูปที่ 5.22 หน้าจอเลือก DBMS และ Test Set ที่จะทดสอบ



รูปที่ 5.23 โดอะต็อกแสดงว่าทดสอบเรียบร้อยแล้ว

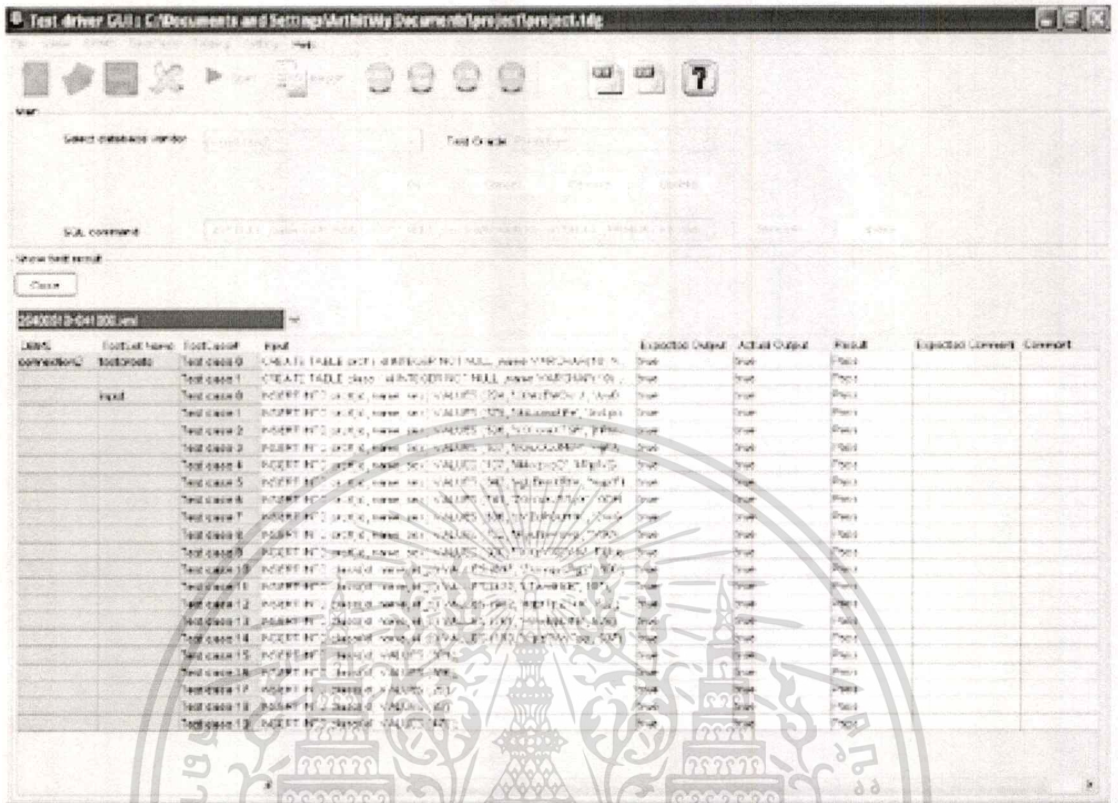
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.24 หน้าจอเมื่อทดสอบเรียบร้อยแล้ว

5.8 เมนูผลลัพธ์

เมื่อผู้ทดสอบต้องการดูผลทดสอบที่เกยทดสอบในโครงการ ผู้ทดสอบจะเลือก Test result ในเมนู View หรือ Report icon ในทูลบาร์ และเลือกไฟล์เอาต์พุตที่ต้องการแสดงผล ดังแสดงดังรูปที่ 5.25



รูปที่ 5.25 หน้าจอแสดงผลลัพธ์ที่เคชทดสอบในโครงการ

5.9 เพิ่มอินพุต เทสต์เซต และเอาต์พุต

ระบบเทสต์ไดร์เวอร์ถูกจัดเก็บเพิ่มโดยใช้ XML ซึ่งสนับสนุนทั้งแบบ DTD และ XML Schemas ดังนี้

5.9.1 เพิ่มอินพุต

เนื่องจากเพิ่มอินพุตจัดเก็บแบบ XML ดังนั้นจึงจำเป็นต้องมีเพิ่มอธิบายเพิ่มอินพุตซึ่งมีอยู่ด้วยกันสองแบบ คือ DTD ดังรูป 5.26 และ แบบ XML Schemas ดังรูป 5.27 และตัวอย่างเพิ่มอินพุตดังรูป 5.28

```

<?xml version='1.0' encoding='UTF-8'?>
<!ELEMENT DBDriver (#PCDATA)>
<!ELEMENT DBPassword (#PCDATA)>
<!ELEMENT DBURL (#PCDATA)>
<!ELEMENT DBUser (#PCDATA)>
<!ELEMENT Oracle (#PCDATA)>
<!ELEMENT xi:include EMPTY >

```

```

<!ATTLIST xi:include href CDATA #REQUIRED >
<!ATTLIST xi:include parse NMTOKEN #REQUIRED >
<!ELEMENT DBMS (DBURL, DBDriver, DBUser, DBPassword, Oracle, xi:include+)>
<!ELEMENT TestDriver (DBMS+)>
<!ATTLIST TestDriver xmlns:xi CDATA #REQUIRED >

```

รูปที่ 5.26 เพิ่มอธิบายเพิ่มอินพุตแบบ DTD

```

<?xml version='1.0' encoding='UTF-8'?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <xs:element name="TestDriver">
  <xs:complexType>
  <xs:sequence>
  <xs:element ref="DBMS" minOccurs="1" maxOccurs="unbounded"/>
  </xs:sequence>
  </xs:complexType>
  </xs:element>
  <xs:element name="DBMS">
  <xs:complexType>
  <xs:sequence>
  <xs:element name="DBURL" type="xs:string"/>
  <xs:element name="DBDriver" type="xs:string"/>
  <xs:element name="DBUser" type="xs:string"/>
  <xs:element name="DBPassword" type="xs:string"/>
  <xs:element name="Oracle" type="xs:string"/>
  <xs:element ref="xi:include" minOccurs="1" maxOccurs="unbounded"/>
  </xs:sequence>
  </xs:complexType>
  </xs:element>
  </xs:schema>

```

รูปที่ 5.27 เพิ่มอธิบายเพิ่มอินพุตแบบ XML Schemas

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TestDriver SYSTEM 'input.dtd'>
<TestDriver xmlns:xi="http://www.w3.org/2001/XInclude">
  <DBMS>
  <DBURL>jdbc:odbc:test</DBURL>
  <DBDriver>sun.jdbc.odbc.JdbcOdbcDriver</DBDriver>
  <DBUser></DBUser>
  <DBPassword></DBPassword>
  <Oracle>Prediction1 </Oracle>
  <xi:include href="testset/test set1.xml" parse="xml"/>
  <xi:include href="testset/test set2.xml" parse="xml"/>
  <DBMS>
  <DBMS>
  <DBURL>jdbc:microsoft:sqlserver://localhost:1433;DatabaseName=test;selectMethod=cursor;</DBURL>
  <DBDriver>com.microsoft.jdbc.sqlserver.SQLServerDriver</DBDriver>
  <DBUser>sa</DBUser>
  <DBPassword></DBPassword>
  <Oracle>Prediction1 </Oracle>
  <xi:include href="testset/test set1.xml" parse="xml"/>
  <xi:include href="testset/test set2.xml" parse="xml"/>
  <DBMS>
</TestDriver>

```

รูปที่ 5.28 ตัวอย่างเพิ่มอินพุต

5.9.2 เพิ่มเทสต์เซต

เนื่องจากเพิ่มเทสต์เซตจัดเก็บแบบ XML ดังนั้นจึงจำเป็นต้องมีเพิ่มอธิบายเพิ่มเทสต์เซต ซึ่งมีอยู่ด้วยกันสองแบบ คือ DTD ดังรูป 5.29 และ แบบ XML Schemas ดังรูป 5.30 และตัวอย่างเพิ่มเทสต์เซตดังรูป 5.31

```

<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT SQL (#PCDATA)>
<!ELEMENT TestSetName (#PCDATA)>
<!ELEMENT TestCase (SQL)>
<!ELEMENT TestSet ( TestSetName,TestCase+)>

```

รูปที่ 5.29 เพิ่มอธิบายเพิ่มเทสต์เซตแบบ DTD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้ใดเห็นนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<?xml version='1.0' encoding='UTF-8'?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="TestSet">
<xs:complexType>
<xs:sequence>
<xs:element name="TestSetName" type="xs:string"/>
<xs:element name="TestCase" minOccurs="1" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="SQL" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

รูปที่ 5.30 เพิ่มอธิบายเพิ่มทดสอบชุดแบบ XML Schemas

```

<?xml version ="1.0" encoding="UTF-8"?>
<!DOCTYPE TestSet SYSTEM 'testsetInput.dtd'>
<TestSet>
<TestSetName>test set1</TestSetName>
<TestCase>
<SQL>SELECT * FROM class WHERE class.id=1111;</SQL>
</TestCase>
</TestSet>

```

รูปที่ 5.31 ตัวอย่างเพิ่มทดสอบชุด

5.9.3 เพิ่มเอาต์พุต

เนื่องจากเพิ่มเอาต์พุตจัดเก็บแบบ XML ดังนั้นจึงจำเป็นต้องมีเพิ่มอธิบายเพิ่มเอาต์พุตซึ่งมีอยู่ด้วยกันสองแบบ คือ DTD ดังรูป 5.32 และ แบบ XML Schemas ดังรูป 5.33 และตัวอย่างเพิ่มอินพุตดังรูป 5.34

```
<?xml version='1.0' encoding='UTF-8'?>
<!ELEMENT TestDriver ( InputFile, DBMS+ ) >
<!ELEMENT InputFile ( #PCDATA ) >
<!ELEMENT DBMS ( DBURL, DBDriver, TestSet+ ) >
<!ELEMENT DBURL ( #PCDATA ) >
<!ELEMENT DBDriver ( #PCDATA ) >
<!ELEMENT TestSet ( TestSetName, TestCase+)>
<!ELEMENT TestSetName ( #PCDATA)>
<!ELEMENT TestCase ( SQL, ActualResult, ExpectedResult, ExpectedComment?, Result, Comment?)>
<!ELEMENT SQL ( #PCDATA)>
<!ELEMENT ActualResult ( #PCDATA)>
<!ELEMENT ExpectedResult ( #PCDATA)>
<!ELEMENT ExpectedComment ( #PCDATA)>
<!ELEMENT Result ( #PCDATA)>
<!ELEMENT Comment ( #PCDATA)>
```

รูปที่ 5.32 เพิ่มอธิบายเพิ่มเอาต์พุตแบบ DTD

```
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="TestDriver">
<xs:complexType>
<xs:sequence>
<xs:element name="InputFile" type="xs:string"/>
<xs:element ref="DBMS" minOccurs='1' maxOccurs='unbounded'/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="DBMS">
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่สามารถนำไปใช้ประโยชน์อื่นใด

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<xs:complexType>
<xs:sequence>
<xs:element name="DBURL" type="xs:string"/>
<xs:element name="DBDriver" type="xs:string"/>
<xs:element ref="TestSet" minOccurs="1" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="TestSet">
<xs:complexType>
<xs:sequence>
<xs:element name="TestSetName" type="xs:string"/>
<xs:element name="TestCase" minOccurs="1" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="SQL" type="xs:string"/>
<xs:element name="ActualResult" type="xs:string"/>
<xs:element name="ExpectedResult" type="xs:string"/>
<xs:element name="ExpectedComment" minOccurs="0" type="xs:string"/>
<xs:element name="Result" type="xs:string"/>
<xs:element name="Comment" minOccurs="0" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

รูปที่ 5.33 เพิ่มอธิบายเพิ่มเอาต์พุตแบบ XML Schemas

```

<?xml version="1.0" encoding="UTF-8"?>
<TestDriver xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  xsi:noNamespaceSchemaLocation="output.xsd">

```

```

<InputFile>C:\Documents and Settings\Arthit\My Documents\po3\input\input.xml</InputFile>
<DBMS>
<DBURL>jdbc:mysql://localhost/test</DBURL>
<DBDriver>com.mysql.jdbc.Driver</DBDriver>
<TestSet>
<TestSetName>testcreate</TestSetName>
<TestCase>
<SQL>CREATE TABLE prof ( id INTEGER NOT NULL ,name VARCHAR(10) NOT NULL ,sex
VARCHAR(10) NOT NULL , PRIMARY KEY(id));</SQL>
<ActualResult>>true</ActualResult>
<ExpectedResult>>true</ExpectedResult>
<Result>Pass</Result>
</TestCase>
<TestCase>
<SQL>CREATE TABLE class ( id INTEGER NOT NULL ,name VARCHAR(10) ,id_p INTEGER ,
PRIMARY KEY(id) , FOREIGN KEY(id_p) REFERENCES prof(id));</SQL>
<ActualResult>>true</ActualResult>
<ExpectedResult>>true</ExpectedResult>
<Result>Pass</Result>
</TestCase>
</TestSet>
<TestSet>
<TestSetName>insert pass</TestSetName>
<TestCase>
<SQL>INSERT INTO prof(id, name, sex) VALUES (990, 'eleWLgyAjQ', 'DLnuLbsbTR');</SQL>
<ActualResult>>true</ActualResult>
<ExpectedResult>>true</ExpectedResult>
<Result>Pass</Result>
</TestCase>
<TestCase>
<TestCase>
<SQL>INSERT INTO prof(id, name, sex) VALUES (990, 'ojnptFTFQi', 'FxfjXNBcZgq');</SQL>
<ActualResult>>false</ActualResult>
<ExpectedResult>>false</ExpectedResult>

```

```

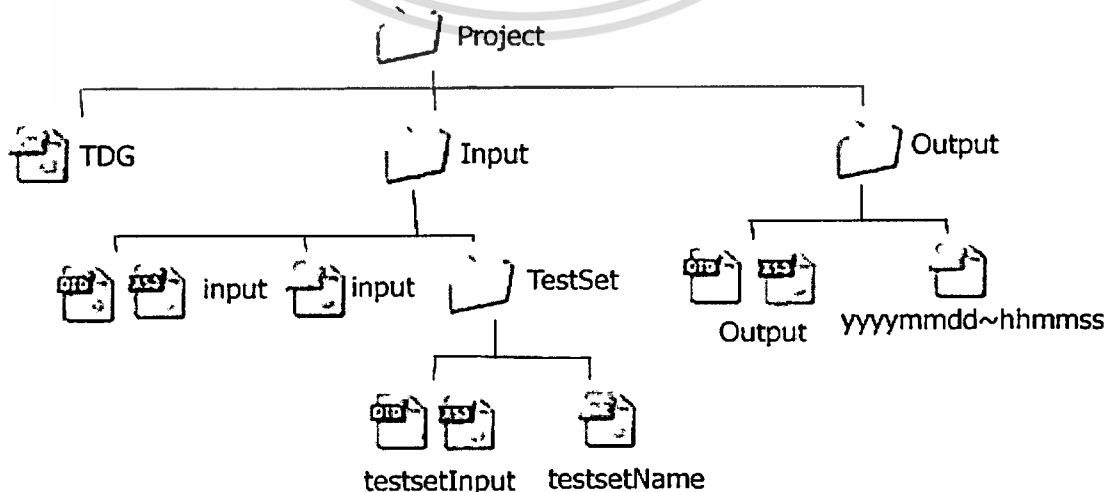
<ExpectedComment>Insert value in primary key must not duplicate</ExpectedComment>
<Result>Pass</Result>
</TestCase>
</TestSet>
<TestSet>
<TestSetName>insert fail</TestSetName>
<TestCase>
<SQL>INSERT INTO class(id, name, id_p) VALUES (755, 777, 177);</SQL>
<ActualResult>>true</ActualResult>
<ExpectedResult>false</ExpectedResult>
<ExpectedComment>insert missing type</ExpectedComment>
<Result>Fail</Result>
<Comment>insert missing type</Comment>
</TestCase>
</TestSet>
</DBMS>
</TestDriver>

```

รูปที่ 5.34 ตัวอย่างเพิ่มเอาต์พุต

5.9.4 โครงสร้างแฟ้ม

จากที่ได้กล่าวมาแล้วว่าในระบบทดสอบไคร์เวอร์กฤษจะประกอบด้วย แฟ้มโครงการ อินพุต เทสต์เซต และเอาต์พุต ซึ่งสามารถแสดงโครงสร้างรูปที่ 5.35



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้รูปที่ 5.35 โครงสร้างแฟ้มอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทสรุป

6.1 สรุปโครงการ

ระบบทดสอบดีไคร์เวอร์กึ่งสำหรับทดสอบดีเอ็มแอลของฐานข้อมูลช่วยให้ผู้ใช้สามารถสร้างชุดทดสอบต่างๆ ได้ง่ายและสะดวก โดยสามารถเก็บไว้ในรูปแบบ XML ซึ่งเป็นมาตรฐานในการแลกเปลี่ยนเพิ่ม เพราะฉะนั้นเราสามารถนำชุดทดสอบนี้ไปใช้ทดสอบกับระบบการจัดการฐานข้อมูล(DBMSs) อื่นๆ ซึ่งเราไม่จำเป็นต้องไปนั่งคีย์หรือสร้างชุดทดสอบใหม่อีกครั้ง และอาจนำไปใช้กับเครื่องทดสอบอื่นๆ ได้(ต้องสนับสนุน XML) นอกจากนี้จะช่วยสร้างเพิ่มอินพุตแล้วเมื่อทดสอบเสร็จระบบทดสอบดีไคร์เวอร์กึ่งยังสนับสนุนในการรวบรวมผลลัพธ์การทดสอบจากชุดทดสอบต่างๆ มาเก็บในรูปแบบของ XML ด้วยซึ่งเราสามารถเก็บผลลัพธ์นี้เพื่อใช้ในการวิเคราะห์หาจุดผิดพลาด หรือเปรียบเทียบผลได้ในภายหลัง

6.2 ประโยชน์ที่ได้รับ

สามารถนำระบบทดสอบดีไคร์เวอร์กึ่งสำหรับทดสอบดีเอ็มแอลของฐานข้อมูล เป็นเครื่องมือในการตรวจสอบ DBMS เพื่อค้นหาข้อผิดพลาดสำหรับภาษาการจัดการฐานข้อมูล ซึ่งเหมาะกับ DBMS ที่พัฒนาขึ้นมาใหม่ ซึ่งจะลดระยะเวลาในการทดสอบซอฟต์แวร์ (Software Testing) ให้เร็วขึ้น

6.3 ข้อเสนอแนะ

เนื่องจากระบบทดสอบดีไคร์เวอร์กึ่งสำหรับทดสอบดีเอ็มแอลของฐานข้อมูล จะทำการทดสอบเฉพาะภาษาสำหรับการจัดการฐานข้อมูล (Data Manipulation Language:DML) เท่านั้น ดังนั้นระบบควรได้รับการปรับปรุงให้ สามารถทดสอบได้ทั้งหมดของภาษา SQL คือ ขยายให้รองรับถึงภาษาสำหรับการนิยามข้อมูล (Data Definition Language:DDL) และ ภาษาการควบคุมข้อมูล (Data Control Language:DCL) ด้วย

บรรณานุกรม

ฉัตรคนัย ช้างพันธ์. 2546. “Test driver GUI.” รายงานทางเทคนิคประกอบงานวิจัย

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

ชนิดา สุจริตชัยตระกูล. 2544. “การแลกเปลี่ยนเอกสารระหว่างองค์กรโดยใช้ XML.”

บทความสัมมนา 2 คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ
ทหารลาดกระบัง.

บุษนา ทิลาศวัฒนกุล และอมรพันธุ์ คำอรรถ. 2545. **สร้างระบบฐานงานฐานข้อมูลด้วย PL/SQL
& Oracle Developer.** กรุงเทพฯ: อินโฟเพรส.

วีระศักดิ์ ชิงถาวร. 2546. **Java Programming Volume III.** กรุงเทพฯ : ซีเอ็ดยูเคชั่น.

Chays, D. et al. 2000. “A Framework for Testing Database Applications.” 147-157. in

Proc. International Symposium on Software Testing and Analysis (ISSTA2000)

Portland, Oregon. USA.

Patton, R. 2001. **Software Testing.** Indiana: Sams Publishing.

ประวัติผู้เขียน

ชื่อ นายอาทิตย์ จิตบำรุงธรรม
สถานที่เกิด อำเภอสีคิ้ว จังหวัดนครราชสีมา
ปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์
สถานที่สำเร็จการศึกษา มหาวิทยาลัยศรีปทุม
ปีที่สำเร็จการศึกษา 2542



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้