

เครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูล

นายชาญวิทย์ ศรีอภานนท์



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
are the thesis authors' files submitted through the University Graduate School.

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2557

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A Tool for Analyzing Impacts to Source Code and Test Cases for Database Schema
Changes

Mr. Chanwit Sriarpanon



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Software Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2014

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์

เครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณี

ทดสอบสำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูล

โดย

นายชาญวิทย์ ศรีอาภาภานนท์

สาขาวิชา

วิศวกรรมซอฟต์แวร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

รองศาสตราจารย์ ดร. ธาราทิพย์ สุวรรณศาสตร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้รับวิทยานิพนธ์ฉบับนี้เป็นส่วน
หนึ่งของการศึกษาตามหลักสูตรปริญญาโทบริหารธุรกิจ

.....คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร. บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. อาทิตย์ ทองทักษ์)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(รองศาสตราจารย์ ดร. ธาราทิพย์ สุวรรณศาสตร์)

.....กรรมการ

(รองศาสตราจารย์ ดร. วิวัฒน์ วัฒนาวุฒิ)

.....กรรมการภายนอกมหาวิทยาลัย

(ดร. อภินิหาร เมธาวัชรนันท์)

ชาญวิทย์ ศรีอาภาภานนท์ : เครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบ สำหรับการเปลี่ยนแปลงสคีมามาฐานข้อมูล (A Tool for Analyzing Impacts to Source Code and Test Cases for Database Schema Changes) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร. ธาราทิพย์ สุวรรณศาสตร์, 97 หน้า.

การทดสอบซอฟต์แวร์ เป็นขั้นตอนหนึ่งที่สำคัญในกระบวนการพัฒนาซอฟต์แวร์ แต่อย่างไรก็ตามการเปลี่ยนแปลงเป็นสิ่งที่ไม่หลีกเลี่ยงไม่ได้และสามารถเกิดขึ้นได้ตลอดเวลาในวัฏจักรของการพัฒนาซอฟต์แวร์ การเปลี่ยนแปลงสามารถเกิดขึ้นได้กับสคีมามาฐานข้อมูล และหากโปรแกรมหรือ ซอร์สโค้ดใดๆ ก็ตามที่ทำงานสัมพันธ์กับสคีมามาฐานข้อมูลดังกล่าว อาจจะได้รับผลกระทบตามมา เนื่องจากสคีมามาฐานข้อมูลเกิดการเปลี่ยนแปลง และกรณีทดสอบที่สร้างด้วยวิธีการแบบไวท์บ็อกซ์ที่อ้างอิงจากซอร์สโค้ดนั้นก็ย่อมเกิดผลกระทบเช่นกัน

ผู้วิจัยจึงพัฒนาเครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมามาฐานข้อมูล โดยเครื่องมือนี้จะวิเคราะห์หาข้อมูลจากล็อกไฟล์ เพื่อหาการเปลี่ยนแปลงที่เกิดขึ้น แล้ววิเคราะห์หาว่าซอร์สโค้ดและกรณีทดสอบใดบ้างที่ได้รับผลกระทบจากการกระทำดังกล่าว พร้อมทั้งแสดงตำแหน่งที่เกิดผลกระทบในซอร์สโค้ด ปรับปรุงแก้ไขข้อมูลทดสอบของกรณีทดสอบที่ได้รับผลกระทบ ให้สามารถนำไปใช้งานต่อได้ หลังจากนั้นผู้วิจัยได้ทดสอบเครื่องมือนี้กับระบบจำลองการทำงาน 3 ระบบ ซึ่งเครื่องมือสามารถวิเคราะห์หาผลกระทบต่อซอร์สโค้ดและกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมามาฐานข้อมูลได้อย่างถูกต้อง อีกทั้งยังสามารถสร้างกรณีทดสอบได้ตามที่คาดไว้

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาควิชา วิศวกรรมคอมพิวเตอร์

ลายมือชื่อนิสิต

สาขาวิชา วิศวกรรมซอฟต์แวร์

ลายมือชื่อ อ.ที่ปรึกษาหลัก

ปีการศึกษา 2557

5570481521 : MAJOR SOFTWARE ENGINEERING

KEYWORDS: DATABASE SCHEMA,STRUCTURAL TESTING,SOFTWARE TESTING,TEST CASE

CHANWIT SRIARPANON: A Tool for Analyzing Impacts to Source Code and Test Cases for Database Schema Changes. ADVISOR: ASSOC. PROF. TARATIP SUWANNASART, Ph.D., 97 pp.

Software testing is an important phase of software development process. However, changes are inevitable and can happen throughout any phase of software development cycle. Changes can occur to database schema. Software or source code which is used database schema must be affected when the database schema is changed. Test cases which are generated from white box technique must be changed as well.

We implement a tool for analyzing impacts to source code and test cases for database schema changes. The tool analyzes changes from a log file, then analyze which a source code and test cases that affected by changed. Moreover, the tool can analyzes the source code and test cases that are affected, as well as fixes and updates affected test cases for ready to be used. After that, we test this tool with 3 systems. The tool can analyze impacts to source code and test cases for database schema changes correctly. It can fix and update test cases and also generate additional test cases.

Department: Computer Engineering Student's Signature

Field of Study: Software Engineering Advisor's Signature

Academic Year: 2014

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความช่วยเหลืออย่างดียิ่งจาก รองศาสตราจารย์ ดร. ธาราทิพย์ สุวรรณศาสตร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่สละเวลาช่วยเหลือ ให้คำแนะนำ และคำปรึกษาเกี่ยวกับการทำวิจัย รวมถึงการให้ความรู้ทั้งทางด้านวิชาการ และการใช้ชีวิตประจำวัน ด้วยความห่วงใยเอาใจใส่จนวิทยานิพนธ์สำเร็จได้ด้วยดี

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. อาทิตย์ ทองทักษ์ ประธานกรรมการสอบ รองศาสตราจารย์ ดร. วิวัฒน์ วัฒนาวุฒิ และ ดร. อภินันท์พร เมธาวีชนันนันท คณะกรรมการสอบ ที่ช่วยชี้แนะและตรวจสอบวิทยานิพนธ์นี้

ขอขอบคุณอาจารย์ทุกท่านที่ได้ให้ความรู้ ให้คำปรึกษาในด้านต่างๆ และเป็นประโยชน์ในการทำวิจัยนี้เป็นอย่างมาก ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ทุกท่านที่ให้คำแนะนำ และให้กำลังใจที่ดีตลอดมาระหว่างการทำวิจัย

สุดท้ายนี้ ขอขอบพระคุณบิดา มารดา ของข้าพเจ้าที่ช่วยเหลือและสนับสนุนในด้านต่างๆ และเป็นกำลังใจสำคัญระหว่างการทำวิจัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญภาพ	ฎ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของงานวิจัย	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของงานวิจัย.....	3
1.4 ขั้นตอนและวิธีดำเนินงานวิจัย	4
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	4
1.6 ลำดับขั้นตอนในการนำเสนอ.....	4
1.7 ผลงานตีพิมพ์จากงานวิจัย.....	5
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	6
2.1 ทฤษฎีที่เกี่ยวข้อง	6
2.1.1 การทดสอบซอฟต์แวร์	6
2.1.2 ภาษาเอสคิวแอล.....	6
2.1.3 เอสคิวแอลแบบฝังตัว.....	8
2.1.4 กรณีทดสอบ	8
2.1.5 การทดสอบแบบไวท์บ็อกซ์	9
2.2 งานวิจัยที่เกี่ยวข้อง.....	9

2.1.1 งานวิจัย A Two-folded Impact Analysis of Schema Changes on Database Applications	9
2.1.2 งานวิจัย Identify Impact of Database Schema on Application.....	10
2.1.3 งานวิจัย Impact Analysis of Database Schema Changes	10
2.1.4 งานวิจัย A Tool for Generating Test Case for Relational Database Constraints Testing	11
2.1.5 งานวิจัย A Tool for Displaying Effect on C# Program from Database Schema Changes	11
บทที่ 3 การวิเคราะห์และออกแบบเครื่องมือ.....	12
3.1 ภาพรวมการทำงานของเครื่องมือ.....	12
3.1.1 วิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล	13
3.1.2 วิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ.....	15
3.1.3 วิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ	24
3.1.4 แก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ.....	27
3.2 การวิเคราะห์และออกแบบเครื่องมือ	28
3.2.1 แผนภาพยูสเคส.....	28
3.2.2 แผนภาพคลาส.....	33
3.2.3 แผนภาพกิจกรรม	40
3.2.4 โครงสร้างฐานข้อมูล	45
บทที่ 4 การพัฒนาเครื่องมือ.....	47
4.1 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ	47
4.1.1 ฮาร์ดแวร์.....	47
4.1.2 ซอฟต์แวร์.....	47
4.2 โครงสร้างส่วนต่อประสานกับผู้ใช้ของเครื่องมือ.....	48

บทที่ 5 การทดสอบเครื่องมือ	56
5.1 สภาพแวดล้อมที่ใช้ทดสอบ	56
5.1.1 ฮาร์ดแวร์	56
5.1.2 ซอฟต์แวร์	56
5.2 การทดสอบเครื่องมือ	57
5.3 ผลการทดสอบ	58
5.3.1 การทดสอบการวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบ	58
5.4 สรุปผลการทดสอบ	59
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ	60
6.1 สรุปผลการวิจัย	60
6.2 ข้อจำกัดของเครื่องมือ	60
6.3 แนวทางการพัฒนาต่อ	61
รายการอ้างอิง	62
ภาคผนวก ก พจนานุกรมข้อมูล	64
ภาคผนวก ข ตัวอย่างระบบทดสอบ	70
ภาคผนวก ค วิธีการใช้เครื่องมือ	88
ประวัติผู้เขียนวิทยานิพนธ์	97

สารบัญตาราง

	หน้า
ตารางที่ 3.1 ผลกระทบต่อ SQL statement ตั้งต้นกรณีเพิ่มฟิลด์.....	17
ตารางที่ 3.2 ผลกระทบต่อ SQL statement ตั้งต้นกรณีลบฟิลด์.....	18
ตารางที่ 3.3 ผลกระทบต่อ SQL statement ตั้งต้นกรณีแก้ไขชื่อฟิลด์.....	19
ตารางที่ 3.4 ผลกระทบต่อ SQL statement ตั้งต้นกรณีแก้ไขประเภทข้อมูลของฟิลด์.....	20
ตารางที่ 3.5 ผลกระทบต่อ SQL statement ตั้งต้นกรณีแก้ไขขนาดของฟิลด์.....	22
ตารางที่ 3.6 ตัวอย่างไฟล์แมปปิง.....	23
ตารางที่ 3.7 ผลกระทบต่อกรณีทดสอบกรณีแก้ไขชื่อฟิลด์.....	24
ตารางที่ 3.8 ผลกระทบต่อกรณีทดสอบกรณีแก้ไขขนาดของฟิลด์แบบลดขนาด.....	25
ตารางที่ 3.9 ผลกระทบต่อกรณีทดสอบกรณีแก้ไขขนาดของฟิลด์แบบเพิ่มขนาด.....	25
ตารางที่ 3.10 ผลกระทบต่อกรณีทดสอบกรณีแก้ไขประเภทข้อมูลของฟิลด์.....	25
ตารางที่ 3.11 ผลกระทบต่อกรณีทดสอบกรณีลบฟิลด์.....	26
ตารางที่ 3.12 ผลกระทบต่อกรณีทดสอบกรณีเพิ่มฟิลด์.....	26
ตารางที่ 3.13 ตัวอย่างกรณีทดสอบที่ได้รับผลกระทบ.....	26
ตารางที่ 3.14 ตัวอย่างกรณีทดสอบที่ได้รับการแก้ไขปรับปรุง.....	28
ตารางที่ 3.15 รายละเอียดยูสเคสการวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล.....	29
ตารางที่ 3.16 รายละเอียดยูสเคสการวิเคราะห์หาข้อผิดพลาดที่ได้รับผลกระทบ.....	30
ตารางที่ 3.17 รายละเอียดยูสเคสการวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ.....	31
ตารางที่ 3.18 รายละเอียดยูสเคสการแก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ.....	32
ตารางที่ 5.1 ผลการทดสอบการวิเคราะห์หาการเปลี่ยนแปลงและผลกระทบกับระบบจำลอง...58	

ตารางที่ ก. 1 พจนานุกรมข้อมูลตาราง File_storage.....	64
ตารางที่ ก. 2 พจนานุกรมข้อมูลตาราง History.....	64
ตารางที่ ก. 3 พจนานุกรมข้อมูลตาราง Log_after_process.....	65
ตารางที่ ก. 4 พจนานุกรมข้อมูลตาราง Mapping_table.....	66
ตารางที่ ก. 5 พจนานุกรมข้อมูลตาราง Schema_file.....	66
ตารางที่ ก. 6 พจนานุกรมข้อมูลตาราง Source_code.....	67
ตารางที่ ก. 7 พจนานุกรมข้อมูลตาราง Testcase.....	68
ตารางที่ ข-1 ตัวอย่างไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลง.....	70
ตารางที่ ข-2 ตัวอย่างไฟล์ล็อก.....	71
ตารางที่ ข-3 ตัวอย่างไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลง.....	77
ตารางที่ ข-4 ตัวอย่างไฟล์ล็อก.....	78
ตารางที่ ข-5 ตัวอย่างไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลง.....	81
ตารางที่ ข-6 ตัวอย่างไฟล์ล็อก.....	82

สารบัญภาพ

	หน้า
ภาพที่ 3-1 ภาพรวมการทำงานของเครื่องมือ.....	12
ภาพที่ 3-2 ตัวอย่างของไฟล์ล็อก.....	13
ภาพที่ 3-3 ตัวอย่างเอกสารสคีมาฐานข้อมูลก่อนเปลี่ยนแปลง.....	14
ภาพที่ 3-4 ตัวอย่างการวิเคราะห์ซอร์สโค้ดแบบฝังตัวในภาษาจาวา.....	22
ภาพที่ 3-5 แผนภาพยูสเคสของเครื่องมือ.....	29
ภาพที่ 3-6 แผนภาพคลาสของเครื่องมือ.....	34
ภาพที่ 3-7 แอตทริบิวต์ของคลาส Main.....	35
ภาพที่ 3-8 แอตทริบิวต์และเมทอดของคลาส New_Project.....	35
ภาพที่ 3-9 แอตทริบิวต์ของคลาส State1.....	36
ภาพที่ 3-10 เมทอดของคลาส State1.....	36
ภาพที่ 3-11 แอตทริบิวต์ของคลาส State2.....	37
ภาพที่ 3-12 เมทอดของคลาส State2.....	37
ภาพที่ 3-13 แอตทริบิวต์และเมทอดของคลาส State3.....	38
ภาพที่ 3-14 แอตทริบิวต์และเมทอดของคลาส State4.....	38
ภาพที่ 3-15 แอตทริบิวต์และเมทอดของคลาส History.....	39
ภาพที่ 3-16 แอตทริบิวต์ของคลาส Testcase.....	39
ภาพที่ 3-17 เมทอดของคลาส Testcase.....	40
ภาพที่ 3-18 แผนภาพกิจกรรมวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล.....	41
ภาพที่ 3-19 แผนภาพกิจกรรมวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ.....	42

ภาพที่ 3-20 แผนภาพกิจกรรมวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ.....	43
ภาพที่ 3-21 แผนภาพกิจกรรมแก้ไขข้อมูลทดสอบ.....	44
ภาพที่ 3-22 โครงสร้างฐานข้อมูลของเครื่องมือ.....	45
ภาพที่ 4-1 แผนภาพ Window Navigation.....	48
ภาพที่ 4-2 หน้าต่างป๊อปอัพ WELCOME.....	49
ภาพที่ 4-3 หน้าจอหลัก.....	49
ภาพที่ 4-4 หน้าต่างติดต่อผู้พัฒนา.....	50
ภาพที่ 4-5 หน้าต่างการใช้งานโปรแกรม.....	50
ภาพที่ 4-6 หน้าต่างประวัติการใช้งาน.....	51
ภาพที่ 4-7 หน้าต่างป๊อปอัพสร้างโปรเจค.....	51
ภาพที่ 4-8 หน้าต่างวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล.....	52
ภาพที่ 4-9 หน้าต่างวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ.....	52
ภาพที่ 4-10 หน้าต่างวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ.....	53
ภาพที่ 4-11 หน้าต่างแก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ.....	54
ภาพที่ 4-12 หน้าต่างป๊อปอัพสถานะแจ้งเตือน (กรณีเพิ่มฟิลด์).....	54
ภาพที่ 4-13 หน้าต่างป๊อปอัพสถานะแจ้งเตือน (บันทึกกรณีทดสอบใหม่เสร็จสิ้น).....	55
ภาพที่ ข-1 เครื่องมือแสดงรายละเอียดการเปลี่ยนแปลง.....	72
ภาพที่ ข-2 ตัวอย่างของซอร์สโค้ด newMember.....	72
ภาพที่ ข-3 ตัวอย่างของไฟล์แมปปิง newMember.....	73
ภาพที่ ข-4 เครื่องมือแสดงตำแหน่งและหมายเลขบรรทัดของซอร์สโค้ดที่ได้รับผลกระทบ.....	73

ภาพที่ ข-5 ตัวอย่างกรณีทดสอบ.....	74
ภาพที่ ข-6 เครื่องมือแสดงรายละเอียดกรณีทดสอบที่ได้รับผลกระทบ.....	74
ภาพที่ ข-7 เครื่องมือแก้ไขข้อมูลทดสอบและส่งออกเสร็จสิ้น.....	75
ภาพที่ ข-8 ตัวอย่างกรณีทดสอบที่แก้ไขข้อมูลทดสอบ.....	76
ภาพที่ ข-9 เครื่องมือแสดงรายละเอียดการเปลี่ยนแปลง.....	79
ภาพที่ ข-10 ตัวอย่างซอร์สโค้ด delPassenger.....	79
ภาพที่ ข-11 ตัวอย่างไฟล์แมปปิง delPassenger.....	80
ภาพที่ ข-12 เครื่องมือวิเคราะห์หาผลกระทบต่อซอร์สโค้ด.....	80
ภาพที่ ข-13 เครื่องมือแสดงรายละเอียดการเปลี่ยนแปลง.....	83
ภาพที่ ข-14 ตัวอย่างซอร์สโค้ด checkPaid.....	84
ภาพที่ ข-15 ตัวอย่างไฟล์แมปปิง checkPaid.....	84
ภาพที่ ข-16 เครื่องมือแสดงตำแหน่งและหมายเลขบรรทัดของซอร์สโค้ดที่ได้รับผลกระทบ.....	85
ภาพที่ ข-17 ตัวอย่างกรณีทดสอบ.....	85
ภาพที่ ข-18 เครื่องมือแสดงรายละเอียดกรณีทดสอบที่ได้รับผลกระทบ.....	86
ภาพที่ ข-19 เครื่องมือแก้ไขข้อมูลทดสอบ.....	87
ภาพที่ ข-20 ตัวอย่างกรณีทดสอบที่แก้ไขข้อมูลทดสอบเสร็จสิ้น.....	87
ภาพที่ ค-1 ข้อความยินดีต้อนรับ.....	88
ภาพที่ ค-2 หน้าต่างหลักของเครื่องมือ.....	88
ภาพที่ ค-3 หน้าต่างสร้างโปรเจคใหม่.....	89
ภาพที่ ค-4 การวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล.....	89

ภาพที่ ค-5 การวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ.....	90
ภาพที่ ค-6 เครื่องมือแสดงตำแหน่งที่เกิดผลกระทบในซอร์สโค้ด.....	91
ภาพที่ ค-7 เครื่องมือแสดงหมายเลขบรรทัดของซอร์สโค้ดที่ได้รับผลกระทบ.....	91
ภาพที่ ค-8 การวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ.....	92
ภาพที่ ค-9 การแก้ไขข้อมูลทดสอบ.....	93
ภาพที่ ค-10 เครื่องมือแสดงกรณีทดสอบที่ปรับปรุงแก้ไขเสร็จสิ้น.....	93
ภาพที่ ค-11 บันทึกกรณีทดสอบใหม่เสร็จสิ้น.....	94
ภาพที่ ค-12 ติดต่อผู้พัฒนา.....	94
ภาพที่ ค-13 การใช้งานโปรแกรม.....	95
ภาพที่ ค-14 ประวัติการใช้งานเครื่องมือ.....	95
ภาพที่ ค-15 ออกจากโปรแกรม.....	96

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของงานวิจัย

การทดสอบซอฟต์แวร์ (Software Testing) ตามหลักของวิศวกรรมซอฟต์แวร์นั้น ถือเป็นขั้นตอนหนึ่งที่สำคัญของวงจรชีวิตการพัฒนาซอฟต์แวร์ (Software Development Life Cycle) เพื่อช่วยให้ซอฟต์แวร์ที่พัฒนาถูกต้องตรงตามความต้องการของผู้ใช้ มีประสิทธิภาพ และความน่าเชื่อถือ โดยในขั้นตอนการทดสอบซอฟต์แวร์นั้นจะต้องสร้างกรณีทดสอบ (Test Case) ที่มีความครอบคลุมกับระบบให้ได้มากที่สุด และเป็นขั้นตอนที่ใช้ทั้งแรงงาน ระยะเวลา และงบประมาณค่อนข้างสูง ซึ่งการทดสอบซอฟต์แวร์สามารถแบ่งออกเป็น 2 ประเภท คือ แบบแบล็กบ็อกซ์ (Black-box) ซึ่งจะเน้นเรื่องของฟังก์ชันการทำงานของซอฟต์แวร์ ซึ่งมองข้ามกลไกภายในของซอฟต์แวร์หรือส่วนประกอบ แต่มุ่งเน้นไปที่ผลลัพธ์ (Output) ที่ออกมาหลังจากการตอบสนองของซอฟต์แวร์เพียงอย่างเดียว และแบบไวท์บ็อกซ์ (White-box) ซึ่งจะเน้นไปที่กลไกการทำงานภายในของซอฟต์แวร์ ทั้งนี้การออกแบบกรณีทดสอบใช้ระยะเวลานานและงบประมาณที่ค่อนข้างสูง

นอกจากนี้ในปัจจุบันฐานข้อมูล (Database) ได้เข้ามามีบทบาทสำคัญในองค์กรต่างๆ มากมาย และได้มีการปรับปรุงให้ใช้งานหรือการติดต่อกับฐานข้อมูลได้ง่ายยิ่งขึ้น ผ่านแอปพลิเคชันฐานข้อมูล และยังเป็นองค์ประกอบสำคัญส่วนหนึ่งในการพัฒนาซอฟต์แวร์ เมื่อระยะเวลาผ่านไป จะยิ่งทำให้ซอฟต์แวร์และฐานข้อมูลมีขนาดใหญ่ขึ้น อีกทั้งหากมีการเปลี่ยนแปลงเกิดขึ้นกับสคีมาฐานข้อมูลหรือโครงสร้างของฐานข้อมูล (Database Schema) อาทิเช่น การลบฟิลด์ (Field) ในบางตาราง การเพิ่มตาราง (Table) ในฐานข้อมูล เป็นต้น ซึ่งการเปลี่ยนแปลงดังกล่าวอาจส่งผลกระทบต่อส่วนต่างๆ ของซอฟต์แวร์และกรณีทดสอบได้

การสร้างกรณีทดสอบ พบว่าเครื่องมือ (Tool Case) ที่ช่วยสร้างกรณีทดสอบเหล่านั้นจะเน้นการสร้างกรณีทดสอบจากขอบเขตของข้อมูลที่รับเข้าเท่านั้น โดยไม่คำนึงถึงข้อมูล หรือสคีมาของฐานข้อมูลที่ใช้สร้างกรณีทดสอบ ถ้าหากเกิดการเปลี่ยนแปลงสคีมาของฐานข้อมูล ย่อมส่งผลกระทบต่อซอร์สโค้ด และกระทบต่อกรณีทดสอบ เพราะกรณีทดสอบที่สร้างจากไวท์บ็อกซ์นั้น เมื่อซอร์สโค้ดเปลี่ยนแปลง ก็อาจจะส่งผลกระทบต่อกรณีทดสอบด้วย จึงทำให้ผู้ทดสอบไม่สามารถทราบได้ว่ากรณีเปลี่ยนแปลงที่เกิดขึ้นนั้นส่งผลกระทบต่อกรณีทดสอบใด และเมื่อผู้ทดสอบนำกรณีทดสอบที่ได้รับ

ผลกระทบไปใช้งาน จะทำให้เกิดความผิดพลาดหรือไม่ถูกต้องต่อการทดสอบ โดยที่ผู้ทดสอบไม่รู้ว่ากรณีทดสอบนั้นๆ ไม่สามารถใช้งานต่อได้ อีกทั้งถ้าหากซอฟต์แวร์นั้นมีขนาดใหญ่ และมีความซับซ้อนมาก จำนวนกรณีทดสอบที่ต้องนำมาทดสอบซอฟต์แวร์ก็จะมีมากตามไปด้วยจึงทำให้เสียเวลา แรงงาน และงบประมาณในการพัฒนาซอฟต์แวร์อย่างมาก

ดังนั้นการวิเคราะห์หาผลกระทบต่อกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาของฐานข้อมูล เพื่อตรวจสอบถึงกรณีทดสอบที่ได้รับผลกระทบและไม่สามารถใช้งานได้ รวมถึงการสร้างกรณีทดสอบใหม่ขึ้นมาทดแทนกรณีทดสอบที่ได้รับผลกระทบนั้น ถือเป็นสิ่งสำคัญอย่างยิ่งในการพัฒนาซอฟต์แวร์

งานวิจัยมีเป้าหมายที่จะเสนอการวิเคราะห์ผลกระทบต่อกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาของฐานข้อมูล โดยมุ่งเน้นการวิเคราะห์ผ่านกลไกการทำงานภายในของซอฟต์แวร์ ดังนั้นงานวิจัยนี้จึงเสนอ การวิเคราะห์ผลกระทบต่อกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาของฐานข้อมูล ด้วยการใช้ไฟล์ล็อก (Log File) โดยเครื่องมือจะวิเคราะห์การเปลี่ยนแปลงที่เกิดขึ้นกับสคีมาฐานข้อมูลผ่านไฟล์ล็อก เพื่อใช้เปรียบเทียบกับเอกสารสคีมาของฐานข้อมูลก่อนเกิดการเปลี่ยนแปลง และวิเคราะห์กลไกการทำงานภายในของซอฟต์แวร์ เพื่อตรวจหาผลกระทบต่อซอร์สโค้ด และกรณีทดสอบเป็นลำดับต่อไป เมื่อเกิดผลกระทบต่อกรณีทดสอบขึ้น ส่งผลให้กรณีทดสอบนั้นไม่สามารถนำไปใช้งานต่อได้ เครื่องมือจึงสร้างกรณีทดสอบใหม่ เพื่อให้ข้อมูลในกรณีทดสอบใหม่ที่สร้างขึ้นมีความสอดคล้องกับข้อมูลภายในสคีมาของฐานข้อมูลที่เกิดการเปลี่ยนแปลงแล้วนั่นเอง จึงทำให้ผู้ทดสอบสามารถนำกรณีทดสอบนี้ไปใช้งานได้ทันที

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อออกแบบ และพัฒนาเครื่องมือวิเคราะห์หาซอร์สโค้ด และกรณีทดสอบที่ได้รับผลกระทบจากการเปลี่ยนแปลงของสคีมาฐานข้อมูล
2. เพื่อแก้ไขกรณีทดสอบที่ใช้งานไม่ได้ โดยสร้างข้อมูลทดสอบใหม่ให้กับกรณีทดสอบที่ได้รับผลกระทบ จากการเปลี่ยนแปลงของสคีมาฐานข้อมูล

1.3 ขอบเขตของงานวิจัย

1. เครื่องมือสามารถรับไฟล์ที่ใช้วิเคราะห์ได้ 4 ชนิด คือ ไฟล์ล็อก ไฟล์เอกสารสคีมามาฐานข้อมูล ก่อนเปลี่ยนแปลง ซอร์สโค้ดแบบฝังตัว และกรณีทดสอบ
2. ไฟล์ล็อกต้องอยู่ในรูปแบบไฟล์เอกสาร และไฟล์เอกสารสคีมามาฐานข้อมูลก่อนเปลี่ยนแปลงต้อง อยู่ในรูปแบบเอสคิวแอลสคริปต์
3. เครื่องมือสามารถรับซอร์สโค้ดแบบฝังตัวในภาษาจาวา และกรณีทดสอบเดิมที่มีอยู่แล้วใน ฐานข้อมูลมาใช้ โดยฐานข้อมูลที่ใช้งานต้องเป็นฐานข้อมูลเชิงสัมพันธ์
4. ซอร์สโค้ดแบบฝังตัวในภาษาจาวา และกรณีทดสอบต้องอยู่ในรูปแบบที่งานวิจัยกำหนดขึ้น เท่านั้น โดยซอร์สโค้ดแบบฝังตัวนั้นจะมีการฝังส่วนที่มีการเชื่อมต่อกับฐานข้อมูลด้วยเครื่องหมาย #sql ส่วนกรณีทดสอบนั้นอยู่ในรูปแบบไฟล์เอกซ์เซลล์ (.xlsx)
5. เครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดแบบฝังตัวในภาษาจาวาได้ครั้งละ 1 คลาส
6. ผู้ใช้ต้องระบุและนำเข้าไฟล์แมปปิง (Mapping File) อธิบายความสัมพันธ์ระหว่างซอร์สโค้ด กับหมายเลขบรรทัดของซอร์สโค้ด ซึ่งอยู่ในรูปแบบไฟล์ซีเอสวี (.csv)
7. ไฟล์ล็อกของฐานข้อมูลที่ใช้รองรับกับ MySQL
8. วัน/เดือน/ปี : เวลา ของไฟล์ล็อกที่นำมาวิเคราะห์ต้องอยู่ในช่วงหลังจากวันที่สร้างเอกสาร สคีมามาฐานข้อมูลก่อนเปลี่ยนแปลง
9. การเปลี่ยนแปลงของสคีมามาฐานข้อมูลจะครอบคลุมเฉพาะการเปลี่ยนแปลงที่เกิดขึ้นภายใน ฟิลด์ของแต่ละตารางเท่านั้น เช่น การแก้ไขชื่อฟิลด์ การลบฟิลด์ และการเพิ่มฟิลด์ การแก้ไขประเภท ข้อมูลของฟิลด์ และการแก้ไขขนาดข้อมูลของฟิลด์ เป็นต้น โดยไม่รวมถึงกรณีการเพิ่มตาราง การ แก้ไขตาราง และการลบตาราง
10. การเปลี่ยนแปลงสคีมามาฐานข้อมูลจะไม่ครอบคลุมเงื่อนไขบังคับบูรณาภาพ (Integrity Constraints)
11. ผลลัพธ์ที่ได้จากเครื่องมือ ประกอบด้วย หมายเลขกรณีทดสอบที่ได้รับผลกระทบ หมายเลขที่อ้างถึงการไหลของข้อมูลภายในโปรแกรม ข้อมูลทดสอบที่ได้รับผลกระทบ และข้อมูล ทดสอบใหม่ที่ถูกแก้ไข ตัวแปรและตำแหน่งที่เกิดผลกระทบในซอร์สโค้ด

12. เครื่องมือที่พัฒนาเสร็จแล้วจะถูกนำไปทดสอบกับระบบงานอย่างน้อย 3 ระบบงาน

1.4 ขั้นตอนและวิธีดำเนินงานวิจัย

1. สำรองและศึกษาเครื่องมือทดสอบ
2. ศึกษารูปแบบสคีมาของฐานข้อมูล
3. ศึกษามาตรฐานของภาษาเอสคิวแอล
4. ศึกษารูปแบบและรายละเอียดของซอร์สโค้ดแบบฝังตัวในภาษาจาวา
5. ออกแบบและกำหนดลักษณะข้อมูลนำเข้าที่นำมาใช้พัฒนาเครื่องมือ
6. กำหนดขอบเขตและความสามารถของเครื่องมือ
7. พัฒนาเครื่องมือตามที่ได้ออกแบบไว้
8. ทดสอบเครื่องมือที่สร้างขึ้นตามขอบเขตที่กำหนดไว้
9. สรุปผลงานวิจัย ข้อเสนอแนะ และจัดทำรายงานวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถวิเคราะห์หาซอร์สโค้ด และกรณีทดสอบที่ได้รับผลกระทบจากการเปลี่ยนแปลงของสคีมาฐานข้อมูล
2. เพื่อลดระยะเวลาและงบประมาณที่ใช้วิเคราะห์หาซอร์สโค้ดและกรณีทดสอบที่ได้รับผลกระทบจากการเปลี่ยนแปลงของสคีมาฐานข้อมูล
3. แก้ไขข้อมูลทดสอบที่ได้รับผลกระทบให้สามารถนำไปใช้งานต่อได้
4. เพิ่มความสะดวกให้กับนักพัฒนา เพราะสามารถนำซอร์สโค้ดที่ได้ระบุตำแหน่งของผลกระทบโดยสามารถนำไปแก้ไขปรับปรุง และนำไปใช้งานต่อได้
5. เครื่องมือจะรายงานผลกระทบที่เกิดขึ้นกับซอร์สโค้ด และกรณีทดสอบให้ผู้ใช้งานทราบ

1.6 ลำดับขั้นตอนในการนำเสนอ

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 6 บทดังต่อไปนี้ บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปัญหา รวมทั้งวัตถุประสงค์ของงานวิจัย ขอบเขตของงานวิจัย ขั้นตอนและวิธีการ

ดำเนินงานวิจัย และประโยชน์ที่คาดว่าจะได้รับ บทที่ 2 กล่าวถึงทฤษฎีที่เกี่ยวข้องรวมถึงงานวิจัยที่เกี่ยวข้อง บทที่ 3 นำเสนอ การวิเคราะห์และออกแบบเครื่องมือ รวมทั้งภาพรวมการทำงานของเครื่องมือ บทที่ 4 การพัฒนาเครื่องมือ บทที่ 5 การทดสอบและสภาพแวดล้อมการทดสอบ และ บทที่ 6 บทสรุป

1.7 ผลงานตีพิมพ์จากงานวิจัย

งานวิจัยที่ได้รับการตีพิมพ์เป็นบทความวิชาการดังต่อไปนี้

1. บทความวิชาการเรื่อง “Impact Analysis of Source Code and Test Cases for Database Schema Changes” โดย Chanwit Sriarpanon and Taratip Suwannasart ได้รับคัดเลือกและตีพิมพ์ในการประชุมวิชาการ “International Computer Science and Engineering Conference (ICSEC2014) : Thai Track” ระหว่างวันที่ 30 กรกฎาคม – 1 สิงหาคม 2557 ณ โรงแรมพูลแมนขอนแก่น จังหวัด ขอนแก่น ประเทศไทย

2. บทความวิชาการเรื่อง “A Source Code and Test Cases Impact Analysis Tool for Database Schema Changes” โดย Chanwit Sriarpanon and Taratip Suwannasart ได้รับคัดเลือกและตีพิมพ์ในการประชุมวิชาการ “International MultiConference of Engineerings and Computer Scientists (IMECS2015)” ระหว่างวันที่ 18 มีนาคม – 20 มีนาคม 2558 ณ โรงแรมเดอะรอยัลการ์เดน ประเทศฮ่องกง

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องในงานวิจัยนี้ ได้แก่ การทดสอบซอฟต์แวร์ ภาษาเอสคิวแอล (SQL) เอสคิวแอลแบบฝังตัว (Embedded SQL) กรณีทดสอบ การทดสอบแบบไวท์บ็อกซ์

2.1.1 การทดสอบซอฟต์แวร์

การทดสอบซอฟต์แวร์ [1] เป็นกระบวนการเพื่อช่วยให้ซอฟต์แวร์ที่พัฒนามี ความถูกต้อง สมบูรณ์ ปลอดภัย และมีคุณภาพที่ดี เพื่อมุ่งหวังให้ซอฟต์แวร์เป็นไปตามความต้องการ หรือเพื่อที่จะ ระบุความแตกต่างระหว่างผลลัพธ์ที่คาดหวังกับผลลัพธ์ที่เกิดขึ้นจริงจากซอฟต์แวร์

การทดสอบซอฟต์แวร์ แบ่งได้เป็น 4 ระดับดังนี้

- การทดสอบระดับหน่วย - เป็นการทดสอบหน่วยย่อยที่สุดของโปรแกรม เพื่อเป็นการ ยืนยันการทำงานระดับย่อยว่าทำงานได้ถูกต้อง เป็นการทดสอบโดยนักเขียนโปรแกรม ผู้เขียนซอร์สโค้ด
- การทดสอบระดับรวมหน่วย - เป็นการทดสอบการเชื่อมต่อส่วนย่อยที่นำมาประกอบกัน ให้ได้ซอฟต์แวร์ที่สมบูรณ์
- การทดสอบระบบ - เป็นการทดสอบการเชื่อมต่อหรือติดต่อสื่อสารกันระหว่างซอฟต์แวร์ หรือระบบอื่นๆ
- การทดสอบการยอมรับ - เป็นการทดสอบโดยผู้ใช้ หรือลูกค้า เพื่อดูว่าซอฟต์แวร์ทำงาน ถูกต้องตามต้องการหรือไม่

2.1.2 ภาษาเอสคิวแอล

ภาษาเอสคิวแอล [2, 3] ภาษาสอบถามเชิงโครงสร้าง (Structured Query Language: SQL, อ่านอย่างย่อว่า เอสคิวแอล, ซีเควล, ซีควล) เป็นภาษาสอบถามที่นิยมมากที่สุดของการจัดการ

ฐานข้อมูล สำหรับสร้าง แก๊ไขและเรียกใช้ฐานข้อมูล โดยใช้มาตรฐานของแอนซี (ANSI) และ ไอเอสโอ (ISO) เป็นต้น

- การใช้งานคำสั่งเอสคิวแอลสามารถนำไปใช้งานได้ 2 แบบคือ
 - แบบโต้ตอบ (Interactive SQL) เป็นรูปแบบที่ผู้ใช้สามารถเขียนคำสั่งภาษาเอสคิวแอลโต้ตอบกันบนจอภาพ เพื่อทำงานกับฐานข้อมูลได้ทันที โดยผู้ใช้จำเป็นต้องมีความรู้ในระดับเบื้องต้นเพื่อสามารถเขียนคำสั่งทำงานกับฐานข้อมูล
 - แบบฝังตัวในโปรแกรม เป็นการนำเอาคำสั่งเอสคิวแอลไปทำงานร่วมกับการเขียนโปรแกรมด้วยภาษาอื่น ซึ่งมีภาษาโปรแกรมเป็นจำนวนมากที่สนับสนุนการทำงานด้วยคำสั่งเอสคิวแอล ในการเขียนลักษณะนี้สามารถลดข้อจำกัดที่คำสั่งเอสคิวแอลไม่สามารถทำได้ เช่น คำสั่งวนลูป (Loop) เป็นต้น
- ชนิดของข้อมูล สามารถแบ่งชนิดของข้อมูลพื้นฐานในภาษาเอสคิวแอล ได้ดังนี้
 - อักขระ (Character)
 - อักขระแบบความยาวคงที่ (Fixed-Length Character) จะใช้ char (n) แทนประเภทของข้อมูลที่เป็นตัวอักขระใด ๆ ที่มีความยาวของข้อมูลคงที่โดยมีความยาว n ตัวอักขระประเภทนี้จะมีการจองเนื้อที่ตามความยาวที่คงที่ตามที่กำหนดไว้ ชนิดของข้อมูลประเภทนี้จะเก็บความยาวของข้อมูลได้มากที่สุด คือ 2000 ตัวอักขระ
 - อักขระแบบความยาวไม่คงที่ (Variable-Length Character) จะใช้ varchar (n) แทนประเภทของข้อมูลที่เป็นตัวอักขระใด ๆ ที่มีความยาวของข้อมูลไม่คงที่โดยมีความยาว n ตัวอักขระประเภทนี้ จะมีการจองเนื้อที่ตามความยาวของข้อมูล ชนิดของข้อมูลประเภทนี้จะเก็บความยาวของข้อมูลได้มากที่สุด คือ 4000 ตัวอักขระ
 - จำนวนตัวเลข (Numeric)

- จำนวนเลขที่ไม่มีจุดทศนิยม จะใช้ Integer เป็นเลขจำนวนเต็มบวกหรือลบขนาดใหญ่ เป็นตัวเลข 10 หลัก ที่มีค่าตั้งแต่ -2,147,583,648 ถึง +2,147,483,647
- เลขจำนวนจริง อาจใช้ Number (n) แทนจำนวนเลขที่ไม่มีจุดทศนิยม และจำนวนเลขที่มีจุดทศนิยม
- วันที่และเวลา (Date/Time) เป็นชนิดวันที่หรือเวลา จะใช้ Date เป็นข้อมูลวันที่และ Time เป็นข้อมูลเวลา

2.1.3 เอสคิวแอลแบบฝังตัว

เอสคิวแอลแบบฝังตัว [2, 3] เป็นวิธีการในการเขียนโปรแกรมที่มีการผสมผสานระหว่างภาษาของโปรแกรมหนึ่งๆ ที่ทำงานร่วมกับภาษาสอบถามเชิงโครงสร้าง หรือ “SQL” หรือเรียกอีกอย่างหนึ่ง คือ ภาษา SQL ที่ฝังตัวในโปรแกรม โดยวิธีการนี้มีภาษาของโปรแกรมที่รองรับมากมาย ได้แก่ C/C++, COBOL, Fortran และ Java เป็นต้น ซึ่งเป็นการเพิ่มความสามารถและประสิทธิภาพของภาษาสอบถามเชิงโครงสร้างให้มากยิ่งขึ้น โดยผลลัพธ์ของคำสั่งที่เกิดจากวิธีการนี้จะถูกส่งผ่านให้กับตัวแปรหรือพารามิเตอร์ในโปรแกรม

2.1.4 กรณีทดสอบ

กรณีทดสอบ [1] เป็นชุดข้อมูลทดสอบที่ใช้ทดสอบโปรแกรมหรือการทำงานของโปรแกรม ขึ้นอยู่กับวัตถุประสงค์ของระบบหรือความต้องการเชิงธุรกิจ (Business Requirement) ซึ่งมีส่วนประกอบหลักดังนี้

- หมายเลขกรณีทดสอบ (Test Case ID)
- หมายเลขที่อ้างถึงการไหลของข้อมูลภายในโปรแกรม (Path)
- ข้อมูลนำเข้าที่ใช้ทดสอบ (Input)
- ผลลัพธ์ที่คาดหวัง (Expected Output)

กรณีทดสอบมีหลากหลายรูปแบบ ทั้งนี้ขึ้นกับวัตถุประสงค์ของระบบ เพื่อให้เป็นไปตามแนวทางที่วางแผนไว้ โดยชนิดของการทดสอบนั้นจะก่อให้เกิดความแตกต่างของกรณีทดสอบที่นำมาทดสอบโปรแกรม หรืออาจจะออกแบบกรณีทดสอบให้เกิดความยืดหยุ่น เพื่อเป็นไปตามวัตถุประสงค์ของแต่ละระบบงาน ซึ่งออกแบบโดยผู้ทดสอบโปรแกรม (Tester)

2.1.5 การทดสอบแบบไวท์บ็อกซ์

การทดสอบแบบไวท์บ็อกซ์ [1] เทคนิคที่ใช้ทดสอบซอฟต์แวร์ หรือเรียกว่า การทดสอบเชิงโครงสร้าง ซึ่งวิธีการนี้จะพิจารณาภายในของระบบหรือทางเดินในโปรแกรม โดยจะมุ่งเน้นไปที่โครงสร้างการทำงานภายในของโปรแกรม ซึ่งนักพัฒนา (Developer) จะทราบถึงรายละเอียดการทำงานภายในของโปรแกรม และผู้ทดสอบจะต้องออกแบบกรณีทดสอบโดยขึ้นกับการทำงานภายในของโปรแกรม เพื่อให้การทดสอบเป็นไปตามผลลัพธ์ที่คาดหวังของวัตถุประสงค์เชิงธุรกิจ ซึ่งบ่งชี้ให้เห็นถึงการทำงานที่ถูกต้อง ทั้งนี้เพื่อให้ได้ซอฟต์แวร์ที่มีประสิทธิภาพตรงตามความต้องการของผู้ใช้

2.2 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องสำหรับงานวิจัยนี้ประกอบด้วย เนื้อหาที่เกี่ยวข้องกับการเปลี่ยนแปลงของสคีมาฐานข้อมูล และการสร้างกรณีทดสอบ ดังนี้

2.1.1 งานวิจัย A Two-folded Impact Analysis of Schema Changes on Database Applications

A Two-folded Impact Analysis of Schema Changes on Database Applications [4] งานวิจัยนี้เสนอวิธีการวิเคราะห์ผลกระทบที่เกิดจากการเปลี่ยนแปลงสคีมาฐานข้อมูล โดยแบ่งออกเป็น 2 ส่วน คือ ส่วนที่กระทบกับซอร์สโค้ด และส่วนที่กระทบกับกรณีทดสอบ ซึ่งงานวิจัยนี้ได้พัฒนาเครื่องมือชื่อ DATA โดยการทำงานจะเริ่มจากวิเคราะห์ผลกระทบต่อซอร์สโค้ดเมื่อมีการเปลี่ยนแปลงสคีมาฐานข้อมูลเกิดขึ้น ซึ่งการทำงานส่วนนี้จะขึ้นกับการสไลซ์โปรแกรม โดยข้อกำหนดของงานวิจัยนี้ คือ การเพิ่มฟิลด์ในตาราง การลบตาราง เป็นต้น ผลลัพธ์ที่ได้จากการวิเคราะห์จะอยู่ในรูปของ CFG (Control Flow Graph) และลำดับถัดไปจะวิเคราะห์ผลกระทบต่อกรณีทดสอบ โดยจะมีกรณีทดสอบเดิมที่เรียงตามลำดับความสำคัญ เมื่อเกิดการเปลี่ยนแปลงสคีมาฐานข้อมูลเกิดขึ้น จึงทำให้เกิดการเรียงลำดับของกรณีทดสอบใหม่ ทั้งนี้ขึ้นกับความสำคัญที่มีผลต่อกรณีทดสอบ งานวิจัยนี้บอกถึงผลกระทบต่อซอร์สโค้ดและกรณีทดสอบ แต่ไม่ได้นำเสนอการสร้างกรณีทดสอบใหม่ขึ้นมาทดแทนของเดิมที่ได้รับผลกระทบเมื่อเกิดการเปลี่ยนแปลงสคีมาฐานข้อมูล

2.1.2 งานวิจัย Identify Impact of Database Schema on Application

Identify Impact of Database Schema on Application [5] งานวิจัยนี้สร้างเครื่องมือเพื่อระบุผลกระทบที่เกิดขึ้นจากการเปลี่ยนแปลงของสคีมาฐานข้อมูลบนแอปพลิเคชันเชิงวัตถุ โดยมีชื่อว่า SEMT (Schema Evolution Management Tool) ซึ่งจะแสดงส่วนประกอบของระบบออกมาในรูปของกราฟ โดยเงื่อนไขของการเปลี่ยนแปลงในงานวิจัยนี้จะแบ่งออกเป็น 3 ระดับ คือ

- การเปลี่ยนแปลงระดับคุณสมบัติคลาส (Properties of Class)
 - เพิ่ม/ลบ/แก้ไข ฟิลด์ (Field)
 - เพิ่ม/ลบ/แก้ไข เมทอด (Method)
- การเปลี่ยนแปลงระดับการสืบทอดกราฟ (Inheritance Graph)
 - เพิ่มคลาสให้กับรายชื่อซูเปอร์คลาส (Super Class)
 - ลบคลาสออกจากรายชื่อซูเปอร์คลาส
- การเปลี่ยนแปลงระดับคลาส (Class)
 - เพิ่ม/ลบ คลาส

ผลลัพธ์จากการวิจัยนี้คือ การระบุผลกระทบที่เกิดจากการเปลี่ยนแปลงสคีมาฐานข้อมูล ในระดับฟิลด์ เมทอด และคลาส ซึ่งเครื่องมือในงานวิจัยนี้พัฒนาขึ้นเพื่อลดระยะเวลาในการจัดการการเปลี่ยนแปลงของสคีมาฐานข้อมูล ลดจำนวนข้อผิดพลาดที่มีโอกาสเกิดขึ้น แต่ยังไม่มีการกล่าวถึง เมื่อพัฒนาระบบส่วนใดบ้างที่ได้รับผลกระทบ เมื่อเกิดการเปลี่ยนแปลงของสคีมาฐานข้อมูล

2.1.3 งานวิจัย Impact Analysis of Database Schema Changes

Impact Analysis of Database Schema Changes [6] งานวิจัยนี้นำเสนอวิธีการวิเคราะห์โปรแกรม แบบสถิต (Static Program) โดยใช้การไหลของข้อมูล (Data Flow Analysis) มาสกัดข้อมูลที่ติดต่อกับฐานข้อมูล โดยนำข้อมูลส่วนนี้มาใช้วิเคราะห์และทำนายผลกระทบที่อาจจะเกิดขึ้นจากการเปลี่ยนแปลงสคีมาของฐานข้อมูล และนำเทคนิคที่เรียกว่า “การสไลซ์โปรแกรม” (Program Slicing) มาปรับใช้กับอัลกอริทึม (Algorithm) เพื่อช่วยลดขนาดของโปรแกรมที่จะใช้วิเคราะห์ ให้เหลือแต่ส่วนที่ต้องการหรือส่วนที่ติดต่อกับฐานข้อมูลเพื่อให้โปรแกรมทำงานเร็วขึ้น ซึ่งงานวิจัยนี้ได้ใช้ Context-sensitive มาช่วยเพิ่มความแม่นยำในการวิเคราะห์ผลกระทบ โดยงานวิจัยนี้ได้สร้างเครื่องมือชื่อ SUITE (Schema Update Impact Tool Environment) โดยใช้เทคนิค

ต่างๆ ที่กล่าวมา ผลลัพธ์ที่ได้จากงานวิจัยนี้ คือ การนำเทคนิคการไหลของข้อมูล การสไลซ์โปรแกรม และ Context sensitivity มาใช้เพื่อเพิ่มความสามารถให้กับเครื่องมือนี้ โดยจะวิเคราะห์ผ่านซอร์สโค้ดและมุ่งเน้นพัฒนาเครื่องมือให้มีประสิทธิภาพเพื่อสร้างรายงานการวิเคราะห์ผลกระทบที่อาจเกิดจากการเปลี่ยนแปลงของสคีมารฐานข้อมูลให้แม่นยำมากยิ่งขึ้น

2.1.4 งานวิจัย A Tool for Generating Test Case for Relational Database Constraints Testing

A Tool for Generating Test Case for Relational Database Constraints Testing [7] งานวิจัยนี้นำเสนอ เครื่องมือสร้างกรณีทดสอบสำหรับทดสอบเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ โดยกรณีทดสอบที่สร้างขึ้นจะอยู่ในรูปแบบของ คำสั่งการเพิ่มข้อมูล การลบข้อมูล และการปรับปรุงข้อมูล จากนั้นกรณีทดสอบจะถูกบันทึกลงในฐานข้อมูลของเครื่องมือ เพื่อใช้ทดสอบกับฐานข้อมูลในลำดับต่อไป หลังจากทดสอบเสร็จสิ้น เครื่องมือทดสอบจะบันทึกผลลัพธ์ และแสดงผลการทดสอบให้ผู้ใช้ทราบ หากเกิดข้อผิดพลาดขึ้น เครื่องมือสามารถแก้ไขปรับปรุงเงื่อนไขของฐานข้อมูลเชิงสัมพันธ์ได้ ผลลัพธ์ที่ได้จากงานวิจัยนี้ คือ กรณีทดสอบที่สามารถตรวจสอบความถูกต้องเงื่อนไขบังคับของฐานข้อมูล พร้อมทั้งปรับปรุงเงื่อนไขของฐานข้อมูลที่ทดสอบ แต่ไม่ได้เน้นถึงประสิทธิภาพ และความแม่นยำของเครื่องมือ

2.1.5 งานวิจัย A Tool for Displaying Effect on C# Program from Database Schema Changes

A Tool for Displaying Effect on C# Program from Database Schema Changes [8] งานวิจัยนี้นำเสนอ เครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดในโปรแกรมภาษาซีชาร์ปจากการเปลี่ยนแปลงสคีมารฐานข้อมูล โดยแบ่งออกเป็น 2 ส่วน คือ กฎไวยากรณ์ที่ค้นหารายละเอียดข้อมูลส่วนที่เชื่อมต่อกับฐานข้อมูล และกฎไวยากรณ์ที่ค้นหาสคีมารฐานข้อมูล โดยวิเคราะห์ซอร์สโค้ดเพื่อหาสคีม่าเก่า และนำข้อมูลสคีม่าใหม่มาเปรียบเทียบ และระบายสีให้กับตำแหน่งที่เกิดผลกระทบ ประกอบด้วย ชื่อตาราง ชื่อฟิลด์ และตัวแปรที่เกิดผลกระทบ ผลลัพธ์ที่ได้จากงานวิจัยนี้ คือ ตำแหน่งของซอร์สโค้ดที่ได้รับผลกระทบ แต่ไม่ได้กล่าวถึงว่ากรณีทดสอบใดบ้างที่ได้รับผลกระทบจากการเปลี่ยนแปลงสคีมารฐานข้อมูล

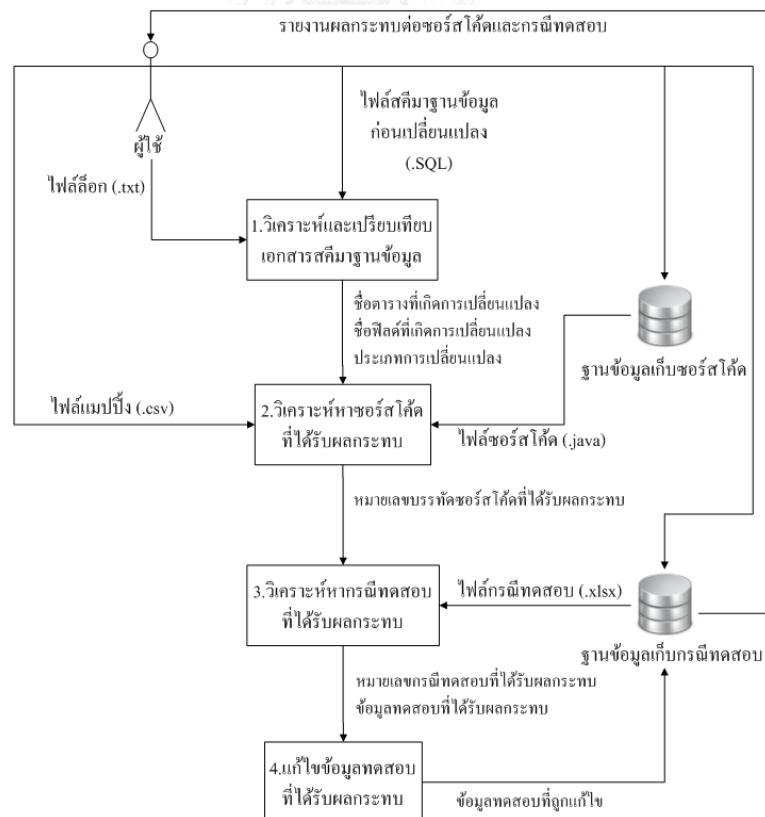
บทที่ 3

การวิเคราะห์และออกแบบเครื่องมือ

ในบทนี้จะอธิบายการวิเคราะห์และการออกแบบเครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ด และกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูล โดยจะกล่าวถึงภาพรวมของเครื่องมือ และอธิบายการวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล การวิเคราะห์ผลกระทบต่อซอร์สโค้ด การวิเคราะห์ผลกระทบต่อกรณีทดสอบ การแก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ รวมทั้งอธิบายโครงสร้างของฐานข้อมูล ซึ่งมีรายละเอียดดังต่อไปนี้

3.1 ภาพรวมการทำงานของเครื่องมือ

วิทยานิพนธ์นี้ได้พัฒนาเครื่องมือที่ใช้วิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบ สำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูล เพื่อใช้วิเคราะห์ถึงตำแหน่งที่เกิดผลกระทบต่อซอร์สโค้ด และกรณีทดสอบใดบ้างที่เกิดผลกระทบ รวมถึงการแก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ ให้สามารถนำไปใช้งานต่อไป ภาพรวมการทำงานของเครื่องมือสามารถแสดงได้ดังภาพที่ 3-1



ภาพที่ 3-1 ภาพรวมการทำงานของเครื่องมือ

ภาพที่ 3-1 แสดงถึงแผนภาพเชิงแนวคิดของเครื่องมือที่งานวิจัยนี้ได้นำเสนอ ซึ่งประกอบไปด้วย 4 ขั้นตอน ส่วนการทำงานในแต่ละขั้นตอนแรกคือ การวิเคราะห์และเปรียบเทียบเอกสารสคีมามาตรฐานข้อมูล โดยให้ผู้ใช้นำเข้า เอกสารสคีมามาตรฐานข้อมูลก่อนเปลี่ยนแปลง และไฟล์ล็อก เพื่อนำมาวิเคราะห์และเปรียบเทียบ ซึ่งทำให้ทราบถึงการเปลี่ยนแปลงที่เกิดขึ้นกับสคีมามาตรฐานข้อมูล โดยผลลัพธ์ที่ได้คือ ชื่อตาราง ชื่อของฟิลด์ที่เกิดการเปลี่ยนแปลง และประเภทของการเปลี่ยนแปลง การทำงานในขั้นตอนที่สอง คือ การวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ โดยขั้นตอนนี้ให้ผู้ใช้นำเข้าไฟล์ซอร์สโค้ดแบบฝังตัว (A Source Code with Embedded SQL in JAVA) เพื่อใช้วิเคราะห์หาผลกระทบและทำเครื่องหมายด้วยสีต่างๆ (Highlight) เพื่อบอกถึงตำแหน่งที่เกิดผลกระทบในซอร์สโค้ด และนำเข้าไฟล์แมปปิง (Mapping File) เพื่อเปรียบเทียบหาหมายเลขบรรทัดที่เกิดผลกระทบในซอร์สโค้ด โดยผลลัพธ์ที่ได้คือ หมายเลขบรรทัดในซอร์สโค้ดที่ได้รับผลกระทบ การทำงานในขั้นตอนที่สาม คือ การวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ โดยผู้ใช้นำเข้ากรณีทดสอบ (Test Cases) ในรูปแบบไฟล์เอกซ์เซลล์ (Excel File) เพื่อวิเคราะห์หมายเลขการไหลของข้อมูลภายในโปรแกรม และชื่อข้อมูลนำเข้าของกรณีทดสอบ (Input Name) และแสดงผลลัพธ์ คือ หมายเลขกรณีทดสอบที่ได้รับผลกระทบ และการทำงานในขั้นตอนที่สี่ คือ การแก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ โดยขั้นตอนนี้จะแก้ไขปรับปรุงเฉพาะข้อมูลทดสอบของกรณีทดสอบที่ได้รับผลกระทบเท่านั้น และสร้างกรณีทดสอบใหม่ สำหรับการเปลี่ยนแปลงบางกรณีเช่น การเพิ่มฟิลด์ เป็นต้น พร้อมทั้งแจ้งเตือนให้ผู้ใช้ทราบถึงผลลัพธ์ของแต่ละขั้นตอนผ่านทางหน้าจอของเครื่องมือ

3.1.1 วิเคราะห์และเปรียบเทียบเอกสารสคีมามาตรฐานข้อมูล

ขั้นตอนการวิเคราะห์และเปรียบเทียบเอกสารสคีมามาตรฐานข้อมูล โดยผู้ใช้นำเข้าไฟล์ล็อกดังภาพที่ 3-2 ที่อยู่ในรูปแบบของไฟล์เอกสาร (Text File) เพื่อนำไปวิเคราะห์และเปรียบเทียบกับเอกสารสคีมามาตรฐานข้อมูลก่อนเปลี่ยนแปลง ซึ่งอยู่ในรูปแบบเอกสารเอสคิวแอลสคริปต์ (.SQL) ดังภาพที่ 3-3 ผลลัพธ์ที่ต้องการคือ การเปลี่ยนแปลงที่เกิดขึ้น ซึ่งประกอบไปด้วย ชื่อตารางที่เกิดการเปลี่ยนแปลง ชื่อฟิลด์ที่เกิดการเปลี่ยนแปลง และประเภทของการเปลี่ยนแปลง เป็นต้น

```
ALTER TABLE `member` CHANGE `password` `password` VARCHAR(5) NOT NULL
ALTER TABLE `member` CHANGE `firstname` `firstname` CHAR(50) NOT NULL
ALTER TABLE `member` CHANGE `cash_balance` `cash_bal` INT(11) NOT NULL
ALTER TABLE `member` ADD `description` VARCHAR(50) NOT NULL
ALTER TABLE `member` DROP `lastname`
```

ภาพที่ 3-2 ตัวอย่างของไฟล์ล็อก

จากภาพที่ 3-2 ประกอบด้วย ชื่อตารางที่เกิดการเปลี่ยนแปลง (Member) ประเภทการเปลี่ยนแปลง (CHANGE, ADD, DROP) ชื่อฟิลด์ (Password, Firstname, Cash_balance, Description, Lastname) ประเภทของฟิลด์ (Field Type) และขนาดของฟิลด์ (Field Size) ที่เกิดการเปลี่ยนแปลง

เครื่องมือจะวิเคราะห์จากคำสั่ง ALTER TABLE เพื่อวิเคราะห์รายละเอียดการเปลี่ยนแปลงที่เกิดขึ้นกับสคีมาฐานข้อมูล จากตัวอย่างนี้มีการเปลี่ยนแปลงกับตาราง Member 5 ครั้ง และนำไปเปรียบเทียบกับเอกสารสคีมาฐานข้อมูลก่อนเปลี่ยนแปลง ดังภาพที่ 3-3 เพื่อให้ทราบถึงการเปลี่ยนแปลงที่เกิดขึ้น

จากภาพที่ 3-2 ขอยกตัวอย่างการเปลี่ยนแปลงที่เกิดขึ้นคือ การลบฟิลด์ Lastname โดยเครื่องมือจะวิเคราะห์ข้อมูลจากไฟล์ล็อก และนำไปเปรียบเทียบกับเอกสารสคีมาฐานข้อมูลก่อนเปลี่ยนแปลง เพื่อให้ทราบถึงการเปลี่ยนแปลง จากตัวอย่างนี้ทำให้ทราบว่าฟิลด์ Lastname varchar(50) ถูกลบออกจากตาราง Member

```
CREATE TABLE IF NOT EXISTS `member` (
  `mid` varchar(10) NOT NULL,
  `password` varchar(10) NOT NULL,
  `firstname` varchar(50) NOT NULL,
  `lastname` varchar(50) NOT NULL,
  `address` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `cash_balance` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

ภาพที่ 3-3 ตัวอย่างเอกสารสคีมาฐานข้อมูลก่อนเปลี่ยนแปลง

เครื่องมือจะนำข้อมูลที่วิเคราะห์ได้จากไฟล์ล็อก ดังภาพที่ 3-2 เปรียบเทียบกับภาพที่ 3-3 เพื่อทราบถึงรายละเอียดการเปลี่ยนแปลงที่เกิดขึ้นของสคีมารฐานข้อมูล ตั้งแต่วันที่สร้างเอกสาร สคีมารฐานข้อมูลก่อนเกิดการเปลี่ยนแปลง โดยสรุปสคีมารฐานข้อมูลหลังเกิดการเปลี่ยนแปลงได้ ดังต่อไปนี้

1. ตารางที่เกิดการเปลี่ยนแปลงคือ ตาราง Member
2. 필ด์และประเภทของการเปลี่ยนแปลงคือ
 - a. 필ด์ Password (แก้ไขขนาดของฟิลด์)
 - b. 필ด์ Firstname (แก้ไขประเภทของฟิลด์)
 - c. 필ด์ Cash_balance (แก้ไขชื่อฟิลด์)
 - d. 필ด์ Description (เพิ่มฟิลด์ใหม่)
 - e. 필ด์ Lastname (ลบฟิลด์)

จากผลลัพธ์ดังกล่าว นำไปหาซอร์สโค้ดที่ได้รับผลกระทบที่เกิดจากการเปลี่ยนแปลงสคีมารฐานข้อมูล โดยใช้ชื่อตาราง ชื่อฟิลด์ และประเภทที่เกิดการเปลี่ยนแปลง

3.1.2 วิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ

ขั้นตอนการวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ โดยผู้ใช้ต้องนำเข้าซอร์สโค้ดแบบฝังตัว เพื่อใช้วิเคราะห์หาผลกระทบที่อาจเกิดขึ้น เมื่อเกิดการเปลี่ยนแปลงสคีมารฐานข้อมูล โดยใช้ผลลัพธ์จากขั้นตอนที่ 3.1.1

จากการวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ เพื่อระบุถึงตำแหน่งที่เกิดผลกระทบในซอร์สโค้ดประกอบด้วย ชื่อตาราง ชื่อฟิลด์ และตัวแปรที่รองรับค่าจากชื่อฟิลด์ โดยประเภทของการเปลี่ยนแปลงแต่ละชนิดส่งผลกระทบต่อซอร์สโค้ดต่างกัน ซึ่งมีรายละเอียดและรูปแบบของผลกระทบต่อ SQL Statement ดังนี้ ที่ใช้วิเคราะห์ในส่วนที่เชื่อมต่อกับฐานข้อมูล แบ่งได้ 5 ประเภท ดังตารางที่ 3.1 ผลกระทบต่อ SQL Statement ตั้งต้นกรณีเพิ่มฟิลด์ ตารางที่ 3.2 ผลกระทบต่อ SQL Statement ตั้งต้นกรณีลบฟิลด์ ตารางที่ 3.3 ผลกระทบต่อ SQL Statement ตั้งต้นกรณีแก้ไขชื่อฟิลด์ ตารางที่ 3.4 ผลกระทบต่อ SQL Statement ตั้งต้นกรณีแก้ไขประเภทข้อมูลของฟิลด์ และ ตารางที่ 3.5 ผลกระทบต่อ SQL Statement ตั้งต้นกรณีแก้ไขขนาดของฟิลด์

- 1) การเปลี่ยนแปลงแบบ เพิ่มฟิลด์ (Add Attribute) จะเกิดผลกระทบต่อ SQL Statement ตั้งต้น ดังตารางที่ 3.1
 - a. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Select เฉพาะ Field Name หรือ Select แบบ Where Clause เป็น Query แบบเฉพาะเจาะจงกับฟิลด์ใดฟิลด์หนึ่ง ซึ่งไม่ก่อให้เกิดผลกระทบต่อซอร์สโค้ด เมื่อเกิดการเปลี่ยนแปลงโดยการเพิ่มฟิลด์ เช่น สมมติซอร์สโค้ด Select Field Name (ชื่อฟิลด์ A) หรือแบบ Where Clause (ชื่อฟิลด์ใดๆ) จะ Query เฉพาะชื่อฟิลด์ดังกล่าว และเมื่อเกิดการเปลี่ยนแปลงโดยการเพิ่มฟิลด์ใหม่เข้ามา ก็ไม่ส่งผลกระทบต่อซอร์สโค้ด เพราะคำสั่งที่ Select นั้นๆ ไม่ได้สนใจต่อฟิลด์ใหม่ที่เพิ่มเข้ามาแต่อย่างใด เป็นต้น
 - b. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Insert แบบระบุชื่อฟิลด์และจำนวนของฟิลด์ ซึ่งเมื่อเกิดการเปลี่ยนแปลงโดยการเพิ่มฟิลด์จะส่งผลให้ซอร์สโค้ดที่มี SQL Statement ดังกล่าวเกิดผลกระทบได้ เช่น Insert ข้อมูลลงในตารางชื่อ Member โดยมีฟิลด์ A, B และ C แต่เมื่อมีการเพิ่มฟิลด์ใหม่เข้ามาคือฟิลด์ D ก็จะส่งผลให้ SQL Statement เกิดผลกระทบ เพราะไม่สามารถเพิ่มค่าในฐานข้อมูลได้ เพราะจำนวนฟิลด์ไม่ตรงกัน เป็นต้น
 - c. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Update แบบ Multiple Field หรือแบบ Where Clause จะเป็นการปรับปรุ่ค่าข้อมูลในฐานข้อมูล โดยเลือกปรับปรุ่จากฟิลด์ใดฟิลด์หนึ่งหรือหลายๆ ฟิลด์ และแบบ Where Clause ซึ่งเมื่อเกิดการเปลี่ยนแปลงกรณีเพิ่มฟิลด์ใหม่ ก็จะไม่ส่งผลกระทบต่อซอร์สโค้ดที่มี SQL Statement แบบนี้ เพราะสนใจแค่ฟิลด์เดิมในฐานข้อมูล โดยไม่ส่งผลกระทบต่ออะไรกับฟิลด์ใหม่ที่เพิ่มเข้ามา เช่น กรณี Update ฟิลด์ A และฟิลด์ B ซอร์สโค้ดก็เน้นการทำงานกับสองฟิลด์ดังกล่าว โดยไม่คำนึงถึงฟิลด์อื่นๆ รวมถึงฟิลด์ใหม่ที่ถูกเพิ่มเข้ามาคือฟิลด์ D ก็จะไม่ส่งผลกระทบต่ออะไรกับซอร์สโค้ดในส่วนนี้ เป็นต้น
 - d. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Delete * หรือ Delete แบบ Where Clause ไม่ส่งผลกระทบต่อซอร์สโค้ด ถึงแม้เกิดการเปลี่ยนแปลงโดยการเพิ่มฟิลด์ใหม่ เช่น Delete * จะลบข้อมูลทั้งหมดออกจากตาราง โดยไม่สนใจฟิลด์ใดฟิลด์หนึ่ง ส่วน Delete แบบ Where Clause จะสนใจฟิลด์ใดฟิลด์หนึ่งเท่านั้น แต่ถึงอย่างไรก็ไม่เกี่ยวข้องใดๆกับฟิลด์ใหม่ที่เพิ่มเข้ามา จึงทำให้ซอร์สโค้ดกรณี Delete ไม่เกิดผลกระทบ เป็นต้น

ตารางที่ 3.1 ผลกระทบต่อ SQL Statement ตั้งต้นกรณีเพิ่มฟิลด์

Change Type (Attribute Level)	SQL Statement	SQL Statement type	กระทบ/ไม่กระทบ (Source Code)
Add new field	Select	Select Field_name	ไม่กระทบ
		Select Where_Clause	ไม่กระทบ
	Insert	-	กระทบ
	Update	Multiple_update	ไม่กระทบ
		Where_clause	ไม่กระทบ
	Delete	Delete *	ไม่กระทบ
		Where_clause	ไม่กระทบ

2) การเปลี่ยนแปลงแบบ ลบฟิลด์ (Drop Attribute) จะเกิดผลกระทบต่อ SQL Statement ตั้งต้น ดังตารางที่ 3.2

- a. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Select แบบ Select Field Name และแบบ Where Clause การเปลี่ยนแปลงแบบลบฟิลด์ ส่งผลให้เกิดผลกระทบต่อซอร์สโค้ด ดังภาพที่ 3-4 ตัวอย่างการวิเคราะห์ซอร์สโค้ดแบบฝังตัวในภาษาจาวา เพราะเมื่อเกิดการเปลี่ยนแปลงโดยการลบฟิลด์ใดๆ (Lastname) จะทำให้ซอร์สโค้ดที่ยังมีการทำงานกับฟิลด์ที่ถูกลบยังคงอยู่ จึงทำให้เกิดข้อผิดพลาดที่จะเชื่อมต่อกับฐานข้อมูลที่ไม่มีฟิลด์ที่ถูกลบอยู่แล้ว เช่น กรณี Select Field Name ของฟิลด์ Mid ฟิลด์ Password ฟิลด์ Firstname และฟิลด์ Lastname ส่งผลให้ซอร์สโค้ดดังกล่าวเกิดผลกระทบ เพราะซอร์สโค้ดยังมีการทำงานเชื่อมต่อกับฐานข้อมูลที่มีฟิลด์ Lastname ซึ่งถูกลบออกจากสคีมารฐานข้อมูลแล้ว เป็นต้น
- b. ผลกระทบต่อ SQL Statement ตั้งต้น: การเปลี่ยนแปลงแบบลบฟิลด์จะส่งผลกระทบต่อซอร์สโค้ดที่มีการเชื่อมต่อกับฐานข้อมูล โดยคำสั่ง Insert จะเกิดผลกระทบต่อซอร์สโค้ด หากเกิดการเปลี่ยนแปลงโดยการลบฟิลด์ใดฟิลด์หนึ่ง เพราะการเพิ่มค่าหรือ Insert ข้อมูลในฐานข้อมูลจะทำได้ ถ้าหากสคีมารฐานข้อมูล มี 3 ฟิลด์ แต่ในซอร์สโค้ดมีการ Insert ถึง 4 ฟิลด์ เป็นต้น
- c. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Update แบบ Multiple Field และแบบ Where Clause จะเกิดผลกระทบ ถ้าหากซอร์สโค้ดนั้นๆ มีการปรับปรุงข้อมูลกับฟิลด์ที่ถูกลบ เช่น Update ฟิลด์ A และฟิลด์ B แต่เมื่อเกิดการเปลี่ยนแปลงสคีมารฐานข้อมูลโดยการลบฟิลด์ B ก็จะมีผลกระทบต่อซอร์สโค้ดนั้นๆ เป็นต้น

- d. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Delete * ไม่ส่งผลกระทบ หากเกิดการเปลี่ยนแปลงโดยการลบฟิลด์ แต่ถ้าเป็น Delete แบบ Where Clause จะส่งผลกระทบกับซอร์สโค้ด เพราะเงื่อนไขนั้นอาจจะไปสัมพันธ์กับฟิลด์ที่ถูกลบ เป็นต้น

ตารางที่ 3.2 ผลกระทบต่อ SQL Statement ตั้งต้นกรณีลบฟิลด์

Change Type (Attribute Level)	SQL Statement	SQL Statement type	กระทบ/ไม่กระทบ (Source Code)
Drop field	Select	Select Field_name	กระทบ
		Select Where_Clause	กระทบ
	Insert	-	กระทบ
	Update	Multiple_update	กระทบ
		Where_clause	กระทบ
	Delete	Delete *	ไม่กระทบ
Where_clause		กระทบ	

- 3) การเปลี่ยนแปลงแบบ แก้ไขชื่อฟิลด์ (Change Name of Attribute) จะเกิดผลกระทบต่อ SQL Statement ตั้งต้น ดังตารางที่ 3.3

- a. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Select แบบ Select Field Name และแบบ Where Clause ทำให้ซอร์สโค้ดเกิดผลกระทบ หากมีการเปลี่ยนแปลงแบบแก้ไขชื่อฟิลด์ เพราะทำให้ซอร์สโค้ดไม่สามารถเชื่อมต่อกับฟิลด์ในฐานข้อมูลได้ เพราะฟิลด์ถูกแก้ไขชื่อ เป็นต้น
- b. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Insert ที่ระบุชื่อฟิลด์ จะส่งผลกระทบต่อซอร์สโค้ด หากเกิดการแก้ไขชื่อฟิลด์ เพราะชื่อฟิลด์ในซอร์สโค้ดจะไม่สอดคล้องกันกับฟิลด์ในสคีมาฐานข้อมูลที่ถูกแก้ไขชื่อแล้ว แต่ถ้า Insert แบบไม่กำหนดชื่อฟิลด์ หากเกิดการเปลี่ยนแปลงแบบนี้ ก็จะไม่ส่งผลกระทบต่อซอร์สโค้ด เป็นต้น
- c. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Update ทั้งแบบ Multiple Field และแบบ Where Clause จะเกิดผลกระทบต่อซอร์สโค้ด หากเกิดการเปลี่ยนแปลงโดยการแก้ไขชื่อฟิลด์ เพราะถ้าซอร์สโค้ดนั้นมีการทำงานกับฟิลด์ที่

ถูกแก้ไขชื่อแล้ว ย่อมส่งผลกระทบต่อโดยตรงกับซอร์สโค้ด เพราะไม่สามารถ Update Field ดังกล่าวได้ เป็นต้น

- d. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Delete แบบ Where Clause จะเกิดผลกระทบต่อซอร์สโค้ด หากเกิดการเปลี่ยนแปลงแบบแก้ไขชื่อฟิลด์ ซึ่งถ้าซอร์สโค้ดมีการเรียกใช้งานฟิลด์ที่ถูกแก้ไขชื่อ ย่อมส่งผลกระทบต่อโดยตรงกับซอร์สโค้ด เพราะไม่สามารถ Delete Field ดังกล่าวได้ เป็นต้น

ตารางที่ 3.3 ผลกระทบต่อ SQL Statement ตั้งต้นกรณีแก้ไขชื่อฟิลด์

Change Type (Attribute Level)	SQL Statement	SQL Statement type	กระทบ/ไม่กระทบ (Source Code)
Change name of field	Select	Select Field_name	กระทบ
		Select Where_Clause	กระทบ
	Insert	Not specify column name	ไม่กระทบ
		Specify column name	กระทบ
	Update	Multiple_update	กระทบ
		Where_clause	กระทบ
	Delete	Delete *	ไม่กระทบ
		Where_clause	กระทบ

- 4) การเปลี่ยนแปลงแบบ แก้ไขประเภทของฟิลด์ (Change Field Type) จะเกิดผลกระทบต่อ SQL Statement ตั้งต้น ดังตารางที่ 3.4

- a. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Select แบบ Select Field Name และแบบ Where Clause จะเกิดผลกระทบต่อซอร์สโค้ด หากเกิดการเปลี่ยนแปลงแบบแก้ไขประเภทข้อมูลของฟิลด์ เพราะจะทำให้ซอร์สโค้ดที่ทำงานกับฟิลด์ที่ถูกแก้ไขประเภทข้อมูล ไม่สามารถ Query ข้อมูลได้ เพราะขอบเขตของข้อมูลอาจจะผิดเพี้ยนไปจากเดิม เช่น การเปลี่ยนประเภทข้อมูลจาก VARCHAR เป็น CHAR เป็นต้น
- b. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Insert จะไม่สามารถเพิ่มข้อมูลในฐานข้อมูลได้ เพราะขอบเขตประเภทข้อมูลอาจจะไม่ตรงกัน และการประกาศประเภทข้อมูลในซอร์สโค้ดที่สัมพันธ์กับตัวแปรของฟิลด์ดังกล่าวอาจจะ

ไม่สอดคล้องกับการเปลี่ยนแปลงประเภทของข้อมูลที่เกิดขึ้นในสคีมาฐานข้อมูล
ย่อมส่งผลกระทบต่อโดยตรงกับซอร์สโค้ดส่วนที่เชื่อมต่อกับฐานข้อมูล เป็นต้น

- c. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Update ทั้งแบบ Multiple Field และแบบ Where Clause จะส่งผลกระทบต่อ หากการเปลี่ยนแปลงสคีมาเกิดขึ้นกับฟิลด์ที่มีการใช้งานในซอร์สโค้ด เพราะทำให้ประเภทของตัวแปรในซอร์สโค้ดไม่สอดคล้องกับประเภทของข้อมูลในฐานข้อมูล ซึ่งทำให้เกิดการผิดพลาดในการ Update ได้
- d. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Delete แบบ Where Clause จะเกิดผลกระทบ หากเกิดการเปลี่ยนแปลงสคีมาฟิลด์ที่ใช้งานอยู่ในซอร์สโค้ด เพราะทำให้ประเภทของตัวแปรในซอร์สโค้ดที่ทำงานสัมพันธ์กับฟิลด์ที่ถูกแก้ไขประเภทไม่สอดคล้องกับประเภทของข้อมูลในฐานข้อมูล จึงเกิดการขัดแย้งกัน และส่งผลกระทบต่อโดยตรงกับซอร์สโค้ดได้

ตารางที่ 3.4 ผลกระทบต่อ SQL Statement ตั้งต้นกรณีแก้ไขประเภทข้อมูลของฟิลด์

Change Type (Attribute Level)	SQL Statement	SQL Statement type	กระทบ/ไม่กระทบ (Source Code)
Change data type	Select	Select Field_name	กระทบ
		Select Where_Clause	กระทบ
	Insert	-	กระทบ
	Update	Multiple_update	กระทบ
		Where_clause	กระทบ
	Delete	Delete *	ไม่กระทบ
Where_clause		กระทบ	

- 5) การเปลี่ยนแปลงแบบ แก้ไขขนาดของฟิลด์ (Change Field Size) จะเกิดผลกระทบต่อ SQL Statement ตั้งต้น ดังตารางที่ 3.5

- a. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Select แบบ Select Field Name และแบบ Where Clause ถ้าเกิดการเปลี่ยนแปลงโดยการเพิ่มขนาดของฟิลด์ จะส่งผลกระทบต่อซอร์สโค้ด เพราะประเภทของตัวแปรที่ประกาศในซอร์สโค้ดอาจจะไม่สามารถรองรับค่าได้เพียงพอ เพราะค่าที่ Query มานั้นอาจจะเกินขอบเขตที่สามารถรองรับได้ รวมถึงการแก้ไขแบบ ลดขนาดของฟิลด์ก็จะส่งผลกระทบต่อซอร์สโค้ดเช่นกัน เพราะขนาดที่เปลี่ยนแปลงอาจจะไม่

สัมพันธ์กับขนาดในซอร์สโค้ด ทั้งนี้ในบางครั้งเมื่อมีการ Insert ค่าลงในฐานข้อมูลที่มีมากเกินขนาดที่ฟิลด์นั้นๆสามารถรองรับได้ ก็เกิดผลกระทบตามมา เป็นต้น

- b. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Insert ถ้าเกิดการเปลี่ยนแปลงโดยการเพิ่มขนาดของฟิลด์ จะไม่ส่งผลกระทบต่อซอร์สโค้ด เพราะค่าที่เพิ่มลงฐานข้อมูลก็สามารถทำได้ดังเดิม แต่ถ้าหากมีการเปลี่ยนแปลงโดยการลดขนาดของฟิลด์ ก็จะส่งผลกระทบต่อซอร์สโค้ด เพราะการ Insert ค่าลงฐานข้อมูลนั้น จะไม่สามารถทำได้ ถ้าหากค่าขนาดเกินขอบเขตที่จะ Insert ลงฐานข้อมูล หรือค่าที่ถูกลดขนาด อาจจะไม่สัมพันธ์กับสคีม่าได้เช่น ค่าเดิมคือ CHAR(10) แต่แก้ไขขนาดเหลือ 5 แล้ว Insert ค่าในฐานข้อมูลเพียง 5 ตัวอักษร ย่อมเกิดผลกระทบได้ เป็นต้น
- c. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Update แบบ Multiple Field และแบบ Where Clause จะเกิดผลกระทบต่อซอร์สโค้ด หากเกิดการเปลี่ยนแปลงโดยการลดขนาดฟิลด์ เพราะเมื่อปรับปรุงข้อมูล หรือ Update ข้อมูลในฐานข้อมูล อาจเกิดข้อผิดพลาดได้ หากค่าที่ปรับปรุงมีค่าเกินขนาดของฟิลด์ที่ถูกแก้ไข เช่น ฟิลด์ B มีขนาดคือ 50 และเมื่อเกิดการเปลี่ยนแปลงโดยการลดขนาดเป็น 10 และเมื่อมีการปรับปรุงข้อมูล ย่อมส่งผลกระทบกับซอร์สโค้ด เพราะขนาดฟิลด์เปลี่ยน แต่ซอร์สโค้ดยังเหมือนเดิม จึงทำให้เกิดข้อผิดพลาดได้
- d. ผลกระทบต่อ SQL Statement ตั้งต้น: กรณี Delete แบบ Where Clause เกิดผลกระทบต่อซอร์สโค้ด เพราะค่าของข้อมูลในฟิลด์ที่ถูกแก้ไขขนาด อาจผิดพลาดไปจากเดิม และส่งผลให้ซอร์สโค้ดตั้งกล่าว ทำงานผิดพลาดได้ เป็นต้น

ตารางที่ 3.5 ผลกระทบต่อ SQL statement ตั้งต้นกรณีแก้ไขขนาดของฟิลด์

Change Type (Attribute Level)	SQL Statement	SQL Statement type	เพิ่ม/ลด (กระทบ)	กระทบ/ไม่กระทบ (Source Code)
Change data size	Select	Select Field_name	เพิ่ม	กระทบ
			ลด	กระทบ
		Select Where_Clause	เพิ่ม	กระทบ
			ลด	กระทบ
	Insert	Not specify column name	เพิ่ม	ไม่กระทบ
			ลด	กระทบ
		Specify column name	เพิ่ม	ไม่กระทบ
			ลด	กระทบ
	Update	Multiple update	เพิ่ม	ไม่กระทบ
			ลด	กระทบ
		Where clause	เพิ่ม	ไม่กระทบ
			ลด	กระทบ
Delete	Delete *	เพิ่ม	ไม่กระทบ	
		ลด	กระทบ	

```
private static void memberLogIn()
throws SQLException, IOException
{
String mmid1, pass1, fname1, lname1;
String lname2 = readEntry("Account#: ");
String pass2 = readEntry("Password: ");
Try
{
#sql {select mid, password, firstname, lastname into :mmid1, :pass1, :fname1,
lname1 from member where lastname = :lname2 and password = :pass2};
}
}
```

ภาพที่ 3-4 ตัวอย่างการวิเคราะห์ซอร์สโค้ดแบบฝังตัวในภาษาจาวา

จากภาพที่ 3-4 เป็นการยกตัวอย่างการวิเคราะห์ซอร์สโค้ดเพื่อหาผลกระทบ เมื่อเกิดการเปลี่ยนแปลงสคีมารฐานข้อมูล โดยเริ่มวิเคราะห์จากส่วนที่มีการเชื่อมต่อกับฐานข้อมูล หรือส่วนที่ขึ้นต้น Statement ด้วยเครื่องหมาย “แฮช” (Hash) หรือเรียกว่า SQL Statement ตั้งต้น ซึ่งเครื่องมือจะสนใจเฉพาะซอร์สโค้ดที่มีชื่อตารางที่สอดคล้องกับการเปลี่ยนแปลงเท่านั้น จาก

ภาพที่ 3-2 ตัวอย่างข้อมูลส่วนที่เกิดการเปลี่ยนแปลงจากไฟล์ล็อก คือ ตาราง Member โดยวิเคราะห์ หา Statement และตัวแปรที่ได้รับผลกระทบ โดยวิเคราะห์จากชื่อฟิลด์ Lastname ที่สอดคล้องกับการเปลี่ยนแปลงดังภาพที่ 3-2 เครื่องมือจะวิเคราะห์ซอร์สโค้ดโดยใช้การเปลี่ยนแปลงที่ได้จากผลลัพธ์ในขั้นตอนที่ 3.1.1 โดยการระบายสี (Highlight) กับชื่อฟิลด์ที่ได้รับผลกระทบคือฟิลด์ Lastname และตัวแปรที่ได้รับค่าจากฟิลด์ดังกล่าว คือ lname1 และlname2 รวมถึงตำแหน่งที่ประกาศตัวแปร และการนำตัวแปรดังกล่าวที่เกิดผลกระทบไปใช้งานในส่วนอื่นๆ ของซอร์สโค้ดในครั้งแรกเท่านั้นเช่น $A = lname1 + lname2 + 5$ (เครื่องมือจะระบายสีเฉพาะตัวแปร lname1, lname2 เท่านั้น) โดยเครื่องมือจะไม่รองรับการนำตัวแปร A ดังกล่าวไปใช้งานต่อในส่วนอื่นๆ และผู้ใช้นำเข้าไฟล์แมปปิง (Mapping File) โดยไฟล์แมปปิงสร้างมาจากซอร์สโค้ด ดังภาพที่ 3-4 ที่ใช้วิเคราะห์ในขั้นตอนที่ 3.1.2 การวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ เพื่อบอกถึงรายละเอียดความสัมพันธ์ระหว่างซอร์สโค้ดกับหมายเลขบรรทัดของซอร์สโค้ด ซึ่งอยู่ในรูปแบบของไฟล์ซีเอสวี (CSV File) โดยผู้ใช้งานต้องสร้างไฟล์แมปปิงด้วยมือ (Manual) โดยองค์ประกอบของไฟล์แมปปิงประกอบด้วย Line of Code Number, Statement ซึ่งสรุปได้ดังตารางที่ 3.6

ตารางที่ 3.6 ตัวอย่างไฟล์แมปปิง

Line of code number	Statement
34	private static void memberLogIn() throws SQLException, IOException
35	{
36	String mmid1, pass1, fname1, lname1;
37	String lname2 = readEntry("Account#: ");
38	String pass2 = readEntry("Password: ");
39	Try
40	{
41	#sql {select mid, password, firstname, lastname into :mmid1, :pass1, :fname1, lname1 from member where lastname = :lname2 and password = :pass2};
42	}

จากตารางที่ 3.6 อธิบายความสัมพันธ์ระหว่าง Statement ในซอร์สโค้ดกับหมายเลขบรรทัดในซอร์สโค้ด โดยดูจากคอลัมน์ด้านขวา คือ ซอร์สโค้ดเป็นหลักคือชื่อตาราง ชื่อฟิลด์ และตัวแปรที่ได้รับผลกระทบ จากนั้นเปรียบเทียบกับคอลัมน์ด้านซ้ายคือ หมายเลขบรรทัดในซอร์สโค้ดที่ได้รับผลกระทบคือ หมายเลข 36, 37 และ 41 ซึ่งจะปรากฏในหมายเลขการไหลของข้อมูลภายในโปรแกรมของกรณีทดสอบที่ทำงานสัมพันธ์กับซอร์สโค้ดดังกล่าว ซึ่งอธิบายได้ว่ากรณีทดสอบดังกล่าว

ครอบคลุมบรรทัดไหนบ้างในซอร์สโค้ด ดังกรณีทดสอบในขั้นตอนที่ 3.1.3 ดังนั้นผลลัพธ์ของขั้นตอนนี้คือ หมายเลขบรรทัดที่ได้รับผลกระทบจากการเปลี่ยนแปลงสคีมาฐานข้อมูลคือ หมายเลข 36, 37 และ 41 ซึ่งจะใช้วิเคราะห์หาผลกระทบต่อกรณีทดสอบในลำดับต่อไป

3.1.3 วิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ

การวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ เครื่องมือจะนำกรณีทดสอบที่มีอยู่เดิมมาใช้วิเคราะห์ โดยกรณีทดสอบในงานวิจัยนี้สร้างจากซอร์สโค้ด และมีรูปแบบของผลกระทบต่อกรณีทดสอบแบ่งได้ 5 ประเภท โดยมีรายละเอียดต่างๆ ดังตารางที่ 3.7 ผลกระทบต่อกรณีทดสอบกรณีแก้ไขชื่อฟิลด์ ตารางที่ 3.8 ผลกระทบต่อกรณีทดสอบกรณีแก้ไขขนาดของฟิลด์แบบลดขนาด ตารางที่ 3.9 ผลกระทบต่อกรณีทดสอบกรณีแก้ไขขนาดของฟิลด์แบบเพิ่มขนาด ตารางที่ 3.10 ผลกระทบต่อกรณีทดสอบกรณีแก้ไขประเภทข้อมูลของฟิลด์ ตารางที่ 3.11 ผลกระทบต่อกรณีทดสอบกรณีลบฟิลด์ และตารางที่ 3.12 ผลกระทบต่อกรณีทดสอบกรณีเพิ่มฟิลด์

จากตารางที่ 3.7 การเปลี่ยนแปลงประเภทนี้จะส่งผลกระทบต่อซอร์สโค้ด เพราะจะทำให้ซอร์สโค้ดไม่สามารถเชื่อมต่อกับฟิลด์ในฐานข้อมูลได้ เนื่องจากชื่อฟิลด์ในซอร์สโค้ดและชื่อฟิลด์ในสคีมาของฐานข้อมูลไม่สอดคล้องกัน แต่ข้อมูลทดสอบในกรณีทดสอบยังสามารถใช้งานต่อได้ เพราะการเปลี่ยนแปลงกรณีแก้ไขชื่อฟิลด์นั้นไม่ส่งผลกระทบต่อข้อมูลทดสอบ แต่ก็ต้องมีการปรับปรุงชื่อของข้อมูลนำเข้าให้กับกรณีทดสอบด้วย

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 3.7 ผลกระทบต่อกรณีทดสอบกรณีแก้ไขชื่อฟิลด์

Change Type (Attribute Level)	SQL Statement	(กระทบ/ไม่กระทบ)	Description
Change name of field	Select field & where clause	ไม่กระทบ	ข้อมูลทดสอบใช้งานได้
	Insert	ไม่กระทบ	ข้อมูลทดสอบใช้งานได้
	Update Field & where clause	ไม่กระทบ	ข้อมูลทดสอบใช้งานได้
	Delete Where clause	ไม่กระทบ	ข้อมูลทดสอบใช้งานได้

จากตารางที่ 3.8 การเปลี่ยนแปลงแบบลดขนาด จะส่งผลกระทบต่อกรณีทดสอบ หากนำกรณีทดสอบนั้นไปใช้ทดสอบย่อมเกิดข้อผิดพลาดได้ เพราะขนาดของข้อมูลทดสอบไม่สอดคล้องกับการเปลี่ยนแปลงที่เกิดขึ้น โดยจะแจ้งให้ผู้ใช้ทราบว่ามีการเปลี่ยนแปลงแบบนี้เกิดขึ้น และผู้ใช้สามารถแก้ไขขนาดของข้อมูลทดสอบให้สอดคล้องกับการเปลี่ยนแปลงได้ในขั้นตอนถัดไป คือ แก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ

ตารางที่ 3.8 ผลกระทบต่อกรณีทดสอบกรณีแก้ไขขนาดของฟิลด์แบบลดขนาด

Change Type (Attribute Level)	SQL Statement	(กระทบ/ไม่กระทบ)	Description
Change data size (ลด)	Select field & where clause	กระทบ	แจ้งเตือนผู้ใช้ (ลดขนาด)
	Insert	กระทบ	แจ้งเตือนผู้ใช้ (ลดขนาด)
	Update Field & where clause	กระทบ	แจ้งเตือนผู้ใช้ (ลดขนาด)
	Delete Where clause	กระทบ	แจ้งเตือนผู้ใช้ (ลดขนาด)

จากตารางที่ 3.9 การเปลี่ยนแปลงแบบเพิ่มขนาด จะไม่ส่งผลกระทบต่อข้อมูลทดสอบในกรณีทดสอบ โดยจะแจ้งเตือนให้ผู้ใช้ทราบว่ามีการเปลี่ยนแปลงแบบเพิ่มขนาด และสามารถแก้ไขขนาดของข้อมูลทดสอบให้สอดคล้องกับการเปลี่ยนแปลงได้ในขั้นตอนถัดไป คือ แก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ

ตารางที่ 3.9 ผลกระทบต่อกรณีทดสอบกรณีแก้ไขขนาดของฟิลด์แบบเพิ่มขนาด

Change Type (Attribute Level)	SQL Statement	(กระทบ/ไม่กระทบ)	Description
Change data size (เพิ่ม)	Select field & where clause	ไม่กระทบ	แจ้งเตือนผู้ใช้ (เพิ่มขนาด)
	Insert	ไม่กระทบ	แจ้งเตือนผู้ใช้ (เพิ่มขนาด)
	Update Field & where clause	ไม่กระทบ	แจ้งเตือนผู้ใช้ (เพิ่มขนาด)
	Delete Where clause	ไม่กระทบ	แจ้งเตือนผู้ใช้ (เพิ่มขนาด)

จากตารางที่ 3.10 การเปลี่ยนแปลงประเภทนี้ส่งผลกระทบต่อกรณีทดสอบ เพราะเมื่อเปลี่ยนแปลงประเภทข้อมูลของฟิลด์อาจจะส่งผลกระทบต่อประเภทของข้อมูลทดสอบในกรณีทดสอบได้ และอาจเกิดข้อผิดพลาด หากนำกรณีทดสอบที่ได้รับผลกระทบไปใช้งานต่อ ซึ่งผู้ใช้สามารถแก้ไขประเภทของข้อมูลทดสอบให้สอดคล้องกับการเปลี่ยนแปลงได้ในขั้นตอนถัดไป คือ แก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ

ตารางที่ 3.10 ผลกระทบต่อกรณีทดสอบกรณีแก้ไขประเภทข้อมูลของฟิลด์

Change Type (Attribute Level)	SQL Statement	(กระทบ/ไม่กระทบ)	Description
Change data type	Select field & where clause	กระทบ	ประเภทของข้อมูลทดสอบ
	Insert	กระทบ	ประเภทของข้อมูลทดสอบ
	Update Field & where clause	กระทบ	ประเภทของข้อมูลทดสอบ
	Delete Where clause	กระทบ	ประเภทของข้อมูลทดสอบ
		ไม่กระทบ	Where clause ที่นำมาสร้างข้อมูลทดสอบไม่ถูกเปลี่ยนประเภท

จากตารางที่ 3.11 การเปลี่ยนแปลงประเภทนี้ หากฟิลด์ใดที่ถูกลบจากสคีมารฐานข้อมูลดังตัวอย่างที่กล่าวมาในขั้นตอนที่ 3.1.1 และขั้นตอนที่ 3.1.2 คือฟิลด์ Lastname ที่ถูกลบจากสคีมารฐานข้อมูล จะส่งผลกระทบต่อข้อมูลทดสอบในกรณีทดสอบ เพราะว่าข้อมูลทดสอบในกรณีทดสอบที่

มีความสัมพันธ์กับฟิลด์ที่ถูกลบยังคงอยู่ ซึ่งต้องแจ้งเตือนให้ผู้ใช้ทราบถึงข้อมูลทดสอบที่ได้รับผลกระทบและไม่สามารถนำไปใช้งานต่อได้ แต่ก็ยังมีข้อมูลทดสอบที่เหลืออยู่ที่ไม่ได้รับผลกระทบสามารถนำไปใช้งานต่อได้ดังเดิม ดังตัวอย่างกรณีทดสอบที่ได้รับผลกระทบในตารางที่ 3.13

ตารางที่ 3.11 ผลกระทบต่อกรณีทดสอบกรณีลบฟิลด์

Change Type (Attribute Level)	SQL Statement	(กระทบ/ไม่กระทบ)	Description
Drop field	Select field & where clause	กระทบ	แก้ไขปรับปรุงข้อมูลทดสอบให้เป็นปัจจุบัน
	Insert	กระทบ	แก้ไขปรับปรุงข้อมูลทดสอบให้เป็นปัจจุบัน
	Update Field & where clause	กระทบ	แก้ไขปรับปรุงข้อมูลทดสอบให้เป็นปัจจุบัน
	Delete Where clause	กระทบ	แก้ไขปรับปรุงข้อมูลทดสอบให้เป็นปัจจุบัน

จากตารางที่ 3.12 การเปลี่ยนแปลงประเภทนี้ ส่งผลกระทบต่อกรณีทดสอบในบางเงื่อนไขเท่านั้น เช่น กรณี insert ถ้ามีการเพิ่มฟิลด์ใหม่เข้ามา ก็จะส่งผลกระทบทั้งซอร์สโค้ดและกรณีทดสอบ ซึ่งจะแจ้งเตือนให้ผู้ใช้ทราบถึงการเปลี่ยนแปลงที่เกิดขึ้น และในขั้นตอนถัดไป คือ แก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ เครื่องมือจะสร้างข้อมูลทดสอบสำหรับฟิลด์ใหม่

ตารางที่ 3.12 ผลกระทบต่อกรณีทดสอบกรณีเพิ่มฟิลด์

Change Type (Attribute Level)	SQL Statement	(กระทบ/ไม่กระทบ)	Description
Add new field	Select field & where clause	ไม่กระทบ	-
	Insert	กระทบ	เพิ่มเติมกรณีทดสอบใหม่
	Update Field & where clause	ไม่กระทบ	-
	Delete Where clause	ไม่กระทบ	-

ตารางที่ 3.13 ตัวอย่างกรณีทดสอบที่ได้รับผลกระทบ

Test case ID	TC1		
Path	34-35-36-37-38-39-40-41-42		
Input			
Name	Type	Size	Value
lastname	STRING	10	Sriarpanon
password	STRING	10	aaaaaaaaa
Expected output	Member login success		

จากตารางที่ 3.13 เป็นตัวอย่างกรณีทดสอบที่สร้างขึ้นจากซอร์สโค้ดในภาพที่ 3-4 เพื่อใช้หาผลกระทบต่อกรณีทดสอบ อันเนื่องมาจากเกิดการเปลี่ยนแปลงสคีมาฐานข้อมูล เครื่องมือจะนำเข้า

กรณีทดสอบเดิม มาใช้วิเคราะห์โดยดูจากหมายเลขการไหลข้อมูลภายในโปรแกรม ที่ระบุหมายเลขบรรทัด ซึ่งอ้างอิงถึงเลขบรรทัดของซอร์สโค้ด โดยจะทำเครื่องหมายเฉพาะหมายเลขที่สอดคล้องกับหมายเลขของซอร์สโค้ดที่เกิดผลกระทบจากขั้นตอนที่ 3.1.2 ซึ่งชี้ให้เห็นว่าการทดสอบนี้มีการไหลผ่านข้อมูลตามบรรทัดที่เกิดผลกระทบในซอร์สโค้ด ซึ่งก่อให้เกิดผลกระทบตามมา หลังจากนั้นวิเคราะห์ข้อมูลทดสอบของกรณีทดสอบเป็นลำดับต่อไป จากรูปจะเห็นว่าหมายเลขบรรทัดการไหลของข้อมูลภายในโปรแกรมที่เกิดผลกระทบคือ หมายเลข 36, 37 และ 41 ซึ่งในบรรทัดนี้ก่อให้เกิดผลกระทบต่อข้อมูลทดสอบในกรณีทดสอบคือชื่อข้อมูลนำเข้า Lastname เพราะการเปลี่ยนแปลงที่ลูปฟิลด์ Lastname ส่งผลให้ข้อมูลทดสอบในกรณีทดสอบไม่สามารถใช้งานต่อได้ เนื่องจากชื่อฟิลด์ดังกล่าวสัมพันธ์กับฟิลด์ที่ถูกลบ (Lastname) ส่งผลให้ข้อมูลทดสอบ “Sriarpanon” ไม่สามารถใช้งานต่อได้

เมื่อกรณีทดสอบเกิดผลกระทบและไม่สามารถนำไปใช้งานต่อได้ จึงต้องมีการแก้ไขข้อมูลทดสอบที่ได้รับผลกระทบหรือปรับปรุงข้อมูลทดสอบให้เป็นปัจจุบัน เพื่อให้สามารถนำไปใช้งานต่อได้ ซึ่งในงานวิจัยนี้จะสร้างข้อมูลทดสอบในกรณีทดสอบเฉพาะตัวที่เกิดผลกระทบและไม่สามารถใช้งานต่อได้เท่านั้น หลังจากนั้นเครื่องมือจะแก้ไขและปรับปรุงข้อมูลทดสอบใหม่ขึ้น เพื่อทดแทนข้อมูลทดสอบที่เกิดผลกระทบและไม่สามารถนำไปใช้งานได้

ผลลัพธ์ของขั้นตอนนี้คือ หมายเลขกรณีทดสอบที่ได้รับผลกระทบ คือ TC1 และชื่อข้อมูลนำเข้าของกรณีทดสอบที่ได้รับผลกระทบคือ Lastname และข้อมูลทดสอบที่ได้รับผลกระทบคือ “Sriarpanon”

3.1.4 แก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ

จากผลลัพธ์ของขั้นตอนที่ 3.1.3 คือ หมายเลขกรณีทดสอบ และข้อมูลทดสอบที่ได้รับผลกระทบและไม่สามารถนำไปใช้งานต่อได้ ซึ่งเครื่องมือจะแก้ไขปรับปรุงข้อมูลทดสอบ ดังนี้

- หมายเลขกรณีทดสอบที่ได้รับผลกระทบคือ TC1
- Path คือ หมายเลข 36, 37 และ 41
- ชื่อข้อมูลนำเข้าคือ Lastname ประเภทข้อมูลคือ STRING ขนาดเท่ากับ 10 และ ค่าข้อมูลทดสอบคือ “Sriarpanon” (ไม่สามารถนำไปใช้งานต่อได้)

เครื่องมือจะแก้ไขปรับปรุงข้อมูลทดสอบเดิมที่ไม่สามารถใช้งานต่อได้ โดยผลกระทบที่เกิดขึ้นกับกรณีทดสอบดังตารางที่ 3.13 คือ เครื่องมือจะลบชื่อข้อมูลนำเข้า Lastname ประเภท(ขนาด) คือ STRING (10) และข้อมูลทดสอบคือ “Sriarpanon” เพื่อให้กรณีทดสอบเป็นปัจจุบัน โดยอ้างอิงจากการเปลี่ยนแปลงที่เกิดขึ้นดังภาพที่ 3-2 และกรณีทดสอบที่ได้รับการแก้ไขปรับปรุง แสดงดังตารางที่ 3.14

ตารางที่ 3.14 ตัวอย่างกรณีทดสอบที่ได้รับการแก้ไขปรับปรุง

Test case ID	TC1		
Path	34-35-36-37-38-39-40-41-42		
Input			
Name	Type	Size	Value
password	STRING	10	aaaaaaaaaa
Expected output	Member login success		

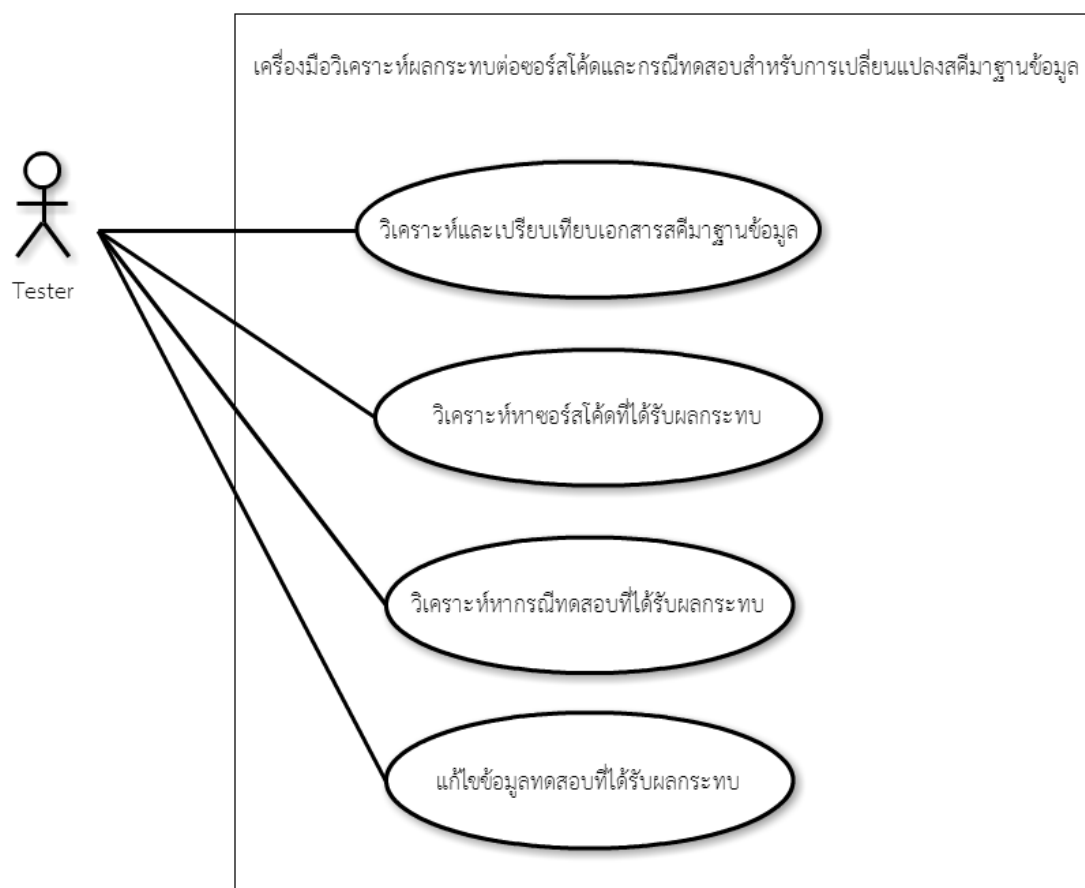
3.2 การวิเคราะห์และออกแบบเครื่องมือ

การวิเคราะห์และออกแบบเครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูลสามารถแสดงได้ด้วยแผนภาพยูสเคส (Use Case Diagram) แผนภาพคลาส (Class Diagram) แผนภาพกิจกรรม (Activity Diagram) และแผนภาพอีอาร์ (E-R Diagram) ซึ่งมีรายละเอียดดังต่อไปนี้

3.2.1 แผนภาพยูสเคส

แผนภาพยูสเคสเป็นแผนภาพที่ได้แสดงให้เห็นถึงฟังก์ชันการทำงานของระบบ โดยแผนภาพยูสเคสของเครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูลประกอบไปด้วย 4 ยูสเคส คือ ยูสเคสการวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล ยูสเคสการวิเคราะห์ผลกระทบต่อซอร์สโค้ด ยูสเคสการวิเคราะห์ผลกระทบต่อกรณีทดสอบ และยูสเคสการแก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ ซึ่งสามารถแสดงได้ดังภาพที่ 3-5 โดยรายละเอียดยูสเคสของการวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูลสามารถแสดงได้ดังตารางที่ 3.15 รายละเอียดยูสเคสของการวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบแสดงได้ดังตารางที่ 3.16 รายละเอียดยูสเคสของการวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบแสดง

ได้ดังตารางที่ 3.17 และรายละเอียดยูสเคสของการแก้ไขข้อมูลทดสอบที่ได้รับผลกระทบแสดงได้ดังตารางที่ 3.18



ภาพที่ 3-5 แผนภาพยูสเคสของเครื่องมือ

ตารางที่ 3.15 รายละเอียดยูสเคสการวิเคราะห์และเปรียบเทียบเอกสารสคีมามาฐานข้อมูล

ยูสเคส	วิเคราะห์และเปรียบเทียบเอกสารสคีมามาฐานข้อมูล
แอกเตอร์	Tester
รายละเอียดยูสเคส	เพื่อวิเคราะห์หาการเปลี่ยนแปลงที่เกิดขึ้น ที่ได้จาก การวิเคราะห์และเปรียบเทียบเอกสารสคีมามาฐานข้อมูลที่ผู้ทดสอบนำเข้าสู่ระบบ
ยูสเคสที่สัมพันธ์	-
เงื่อนไขก่อนหน้า	-

ตารางที่ 3.15 รายละเอียดยวดยุสเคสการวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล (ต่อ)

ขั้นตอน	<ol style="list-style-type: none"> 1. ผู้ทดสอบเลือกไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลงเข้าสู่ระบบผ่านหน้าจอของเครื่องมือ และบันทึกไฟล์เข้าสู่ฐานข้อมูล 2. ผู้ทดสอบเลือกล็อกไฟล์เข้าสู่ระบบผ่านทางหน้าจอของเครื่องมือและบันทึกไฟล์เข้าสู่ฐานข้อมูล 3. ผู้ทดสอบกดปุ่มวิเคราะห์ 4. ระบบวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูลก่อนเปลี่ยนแปลง เพื่อหาการเปลี่ยนแปลงที่เกิดขึ้น
เงื่อนไขภายหลัง	ระบบแสดงผลพัทธ์การเปลี่ยนแปลงผ่านทางหน้าจอของเครื่องมือ

ตารางที่ 3.16 รายละเอียดยวดยุสเคสการวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ

ยูสเคส	วิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ
แอกเตอร์	Tester
รายละเอียดยวดยุสเคส	เพื่อวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบจากการเปลี่ยนแปลงที่เกิดขึ้นกับสคีมาฐานข้อมูล
ยูสเคสที่สัมพันธ์	-
เงื่อนไขก่อนหน้า	ผลลัพธ์จากการเปลี่ยนแปลงของขั้นตอนก่อนหน้านำมาใช้วิเคราะห์หาผลกระทบต่อซอร์สโค้ด

ตารางที่ 3.16 รายละเอียดยวดยุสเคสการวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ (ต่อ)

ขั้นตอน	<ol style="list-style-type: none"> 1. ผู้ทดสอบเลือกไฟล์ซอร์สโค้ดแบบฝังตัวในภาษาจาวาเข้าสู่ระบบผ่านทางหน้าจอของเครื่องมือและบันทึกไฟล์เข้าสู่ฐานข้อมูล 2. ผู้ทดสอบกดปุ่มวิเคราะห์ 3. ระบบวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ แสดงตำแหน่งที่เกิดผลกระทบ พร้อมทั้งไฮไลน์สีจําแนกตามประเภทของการเปลี่ยนแปลง ผ่านทางหน้าจอของเครื่องมือ 4. ผู้ทดสอบเลือกไฟล์แมปปิงเข้าสู่ระบบผ่านทางหน้าจอของเครื่องมือและบันทึกไฟล์เข้าสู่ฐานข้อมูล 5. ผู้ทดสอบกดปุ่มวิเคราะห์ 6. ระบบวิเคราะห์หาเลขบรรทัดที่ได้รับผลกระทบจากการเปลี่ยนแปลง
เงื่อนไขภายหลัง	ระบบแสดงหมายเลขบรรทัดของซอร์สโค้ดที่เกิดผลกระทบผ่านทางหน้าจอของเครื่องมือ

ตารางที่ 3.17 รายละเอียดยวดยุสเคสการวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ

ยวดยุสเคส	วิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ
แอกเตอร์	Tester
รายละเอียดยวดยุสเคส	เพื่อวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบจากการเปลี่ยนแปลงที่เกิดขึ้นกับสคีมาฐานข้อมูล
ยวดยุสเคสที่สัมพันธ์	-
เงื่อนไขก่อนหน้า	หมายเลขบรรทัดของซอร์สโค้ดที่เกิดผลกระทบ

ตารางที่ 3.17 รายละเอียดยวดยุสเคสการวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ (ต่อ)

ขั้นตอน	<ol style="list-style-type: none"> 1. ผู้ทดสอบเลือกไฟล์กรณีทดสอบเข้าสู่ระบบผ่านทางหน้าจอของเครื่องมือและบันทึกไฟล์เข้าสู่ระบบฐานข้อมูล 2. ผู้ทดสอบกดปุ่มวิเคราะห์ 3. ระบบวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ
เงื่อนไขภายหลัง	ระบบแสดงหมายเลขของกรณีทดสอบที่เกิดผลกระทบผ่านทางหน้าจอของเครื่องมือ

ตารางที่ 3.18 รายละเอียดยวดยุสเคสการแก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ

ยูสเคส	แก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ
แอกเตอร์	Tester
รายละเอียดยวดยุสเคส	เพื่อปรับปรุงแก้ไขข้อมูลทดสอบให้กับกรณีทดสอบที่ได้รับผลกระทบ
ยูสเคสที่สัมพันธ์	-
เงื่อนไขก่อนหน้า	หมายเลขของกรณีทดสอบที่เกิดผลกระทบ
ขั้นตอน	<ol style="list-style-type: none"> 1. ผู้ทดสอบเลือกหมายเลขกรณีทดสอบที่ต้องการจะปรับปรุงแก้ไขผ่านทางหน้าจอของเครื่องมือ 2. ผู้ทดสอบกดปุ่มปรับปรุงแก้ไข 3. ระบบปรับปรุงแก้ไขกรณีทดสอบ 4. ระบบแสดงผลรหัสหมายเลขกรณีทดสอบที่ได้ปรับปรุงแก้ไขเสร็จสิ้นผ่านทางหน้าจอ 5. ผู้ทดสอบสามารถเลือกดูรายละเอียดของกรณีทดสอบที่ปรับปรุงแก้ไขเสร็จสิ้นได้ 6. ผู้ทดสอบกดปุ่มเลือกโฟลเดอร์ 7. ผู้ทดสอบระบุตำแหน่งที่ต้องการบันทึกกรณีทดสอบใหม่ 8. ผู้ทดสอบกดส่งออกกรณีทดสอบใหม่

ตารางที่ 3.18 รายละเอียดยูสเคสการแก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ (ต่อ)

เงื่อนไขภายหลัง	กรณีทดสอบใหม่ถูกบันทึกลงในโพลเดอร์
-----------------	------------------------------------

3.2.2 แผนภาพคลาส

แผนภาพคลาสเป็นแผนภาพที่แสดงรายละเอียดของคลาสและความสัมพันธ์ระหว่างคลาส ซึ่งแผนภาพคลาสของเครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูลแสดงได้ดังภาพที่ 3-6 ซึ่งมีรายละเอียดดังต่อไปนี้

1) คลาส Main คือคลาสที่ทำหน้าที่ติดต่อผู้ทดสอบ โดยคลาสนี้เป็นคลาสหลักของเครื่องมือ เพื่อนำไปสู่การใช้งาน เมนูหลัก ช่วยเหลือ และประวัติการใช้งาน รายละเอียดของ Main แสดงได้ดังภาพที่ 3-7

2) คลาส History คือคลาสที่ทำหน้าที่แสดงประวัติการใช้งานของเครื่องมือให้กับผู้ทดสอบ ผ่านทางหน้าจอ โดยคลาสนี้จะมีรายการให้เลือกเช่น ครั้งที่ รายการ เพื่อค้นหาประวัติการใช้งานต่างๆ จากระบบฐานข้อมูล โดยแสดงผลพร้อมออกมาในรูปแบบตาราง แสดงได้ดังภาพที่ 3-15

3) คลาส Step คือคลาสที่อธิบายถึงรายละเอียดขั้นตอนวิธีการใช้งานของเครื่องมือ

4) คลาส Contact_Us คือ คลาสที่แสดงรายละเอียดของผู้พัฒนา เวอร์ชันของเครื่องมือ

5) คลาส New_Project คือ คลาสที่ติดต่อกับผู้ทดสอบผ่านทางหน้าจอ โดยให้ผู้ทดสอบกรอกข้อมูล ชื่อโปรเจกต์ และชื่อผู้สร้างโปรเจกต์ แสดงได้ดังภาพที่ 3-8

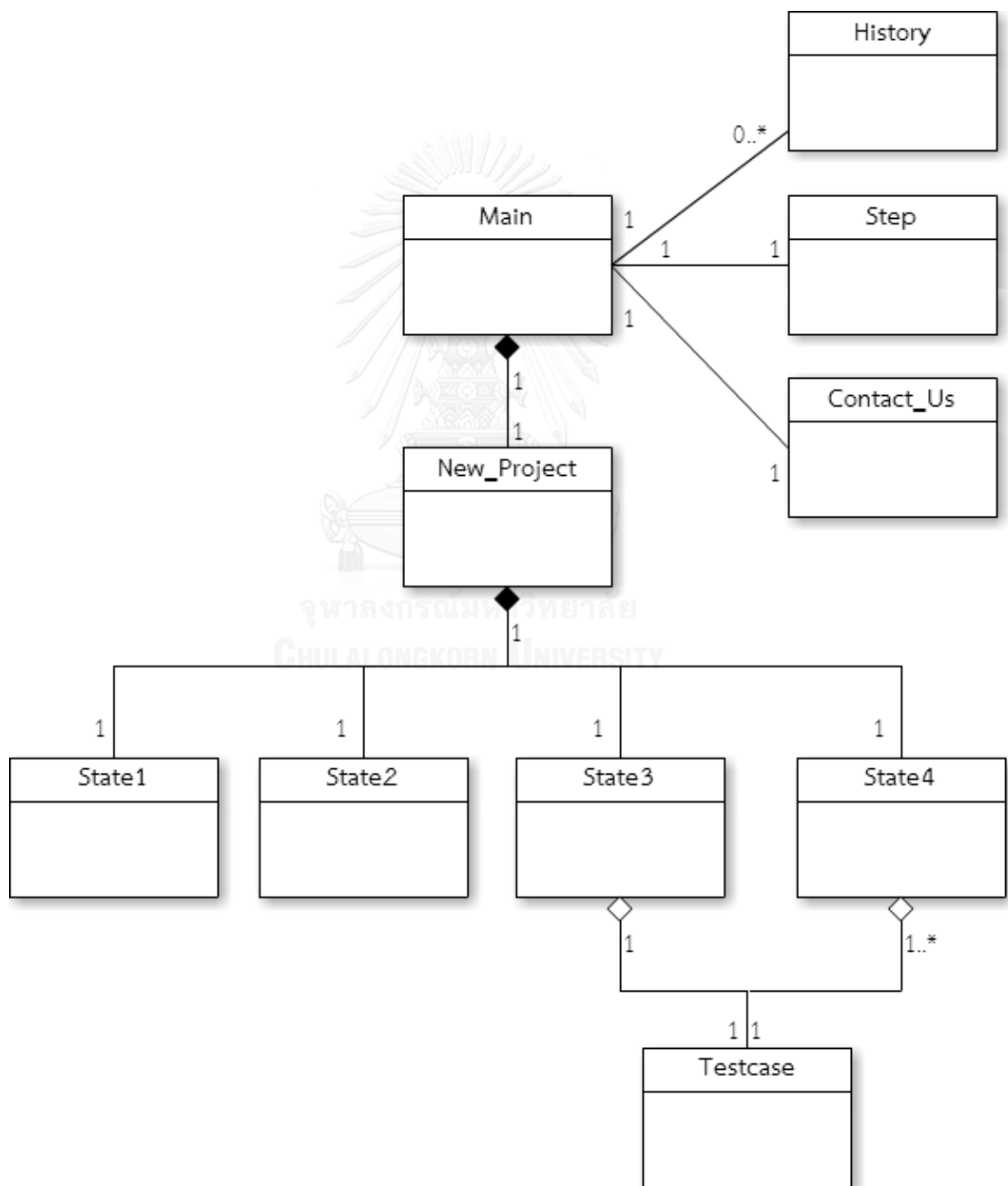
6) คลาส State1 คือ คลาสที่มีการทำงานในส่วนของการวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล ซึ่งให้ผู้ทดสอบอัปโหลด ไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลง (.SQL) และ ล็อกไฟล์ (.txt) เพื่อใช้วิเคราะห์หาผลลัพธ์จากการเปลี่ยนแปลง โดยแสดงผลพร้อมออกมาอยู่ในรูปแบบตาราง แสดงได้ดังภาพที่ 3-9 และ ภาพที่ 3-10

7) คลาส State2 คือ คลาสที่มีการทำงานในส่วนของการวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ โดยจะแสดงตำแหน่งที่เกิดผลกระทบในซอร์สโค้ด ด้วยการระบายสี (Hilight) รวมถึงแสดงเลขบรรทัดของซอร์สโค้ดที่ได้รับผลกระทบ แสดงได้ดังภาพที่ 3-11 และภาพที่ 3-12

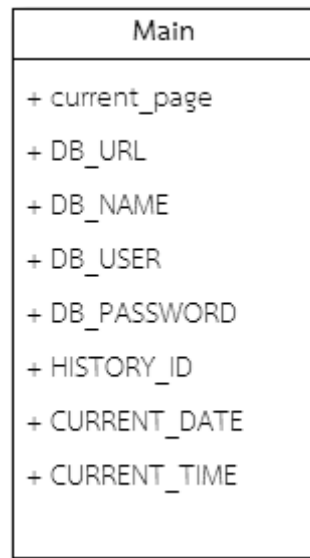
8) คลาส State3 คือ คลาสที่มีการทำงานในส่วนของการวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ โดยจะแสดงหมายเลขของกรณีทดสอบ และรายละเอียดของกรณีทดสอบที่ได้รับผลกระทบ แสดงได้ดังภาพที่ 3-13

9) คลาส State4 คือ คลาสที่มีการทำงานในส่วนของการแก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ โดยจะปรับปรุงแก้ไขกรณีทดสอบที่ได้รับผลกระทบ และแสดงรายละเอียดของกรณีทดสอบที่แก้ไขเสร็จสิ้น รวมทั้งสามารถส่งออกไฟล์กรณีทดสอบใหม่ได้ แสดงได้ดังภาพที่ 3-14

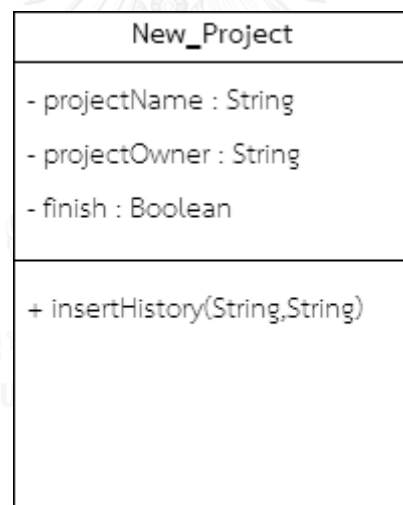
10) คลาส Testcase คือ คลาสที่ถูกเรียกใช้งานจากการทำงานในส่วนของการวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ และการแก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ แสดงได้ดังภาพที่ 3-16 และภาพที่ 3-17



ภาพที่ 3-6 แผนภาพคลาสของเครื่องมือ



ภาพที่ 3-7 แอตทริบิวต์ของคลาส Main



ภาพที่ 3-8 แอตทริบิวต์และเมทอดของคลาส New_Project

State1
- log_group_by_field : ArrayList
+ schema_compare : ArrayList
+ log_group_by_field_relate_schema_table : ArrayList
+ changename_history : ArrayList
- schemaTableName : String
- schemaFieldName : String
- schemaFieldType : String
- schemaFieldSize : String
- schemaFieldPosition : String
- logTableName : String
- logFieldOriginalName : String
- logFieldName : String
- logFieldType : String
- logFieldSize : String
- logFieldStatus : String

ภาพที่ 3-9 แอตทริบิวต์ของคลาส State1

จุฬาลงกรณ์มหาวิทยาลัย

+ insertSchemaFile(File) : void
+ insertSchemaData(String,String,String,String,String) : void
+ insertLogFile(File) : void
+ insertLogData(String,String,String,String,String,String) : void
+ printLogData() : void
+ printSchemaData() : void
+ printChangeNameHistory() : void
+ printScreen() : void

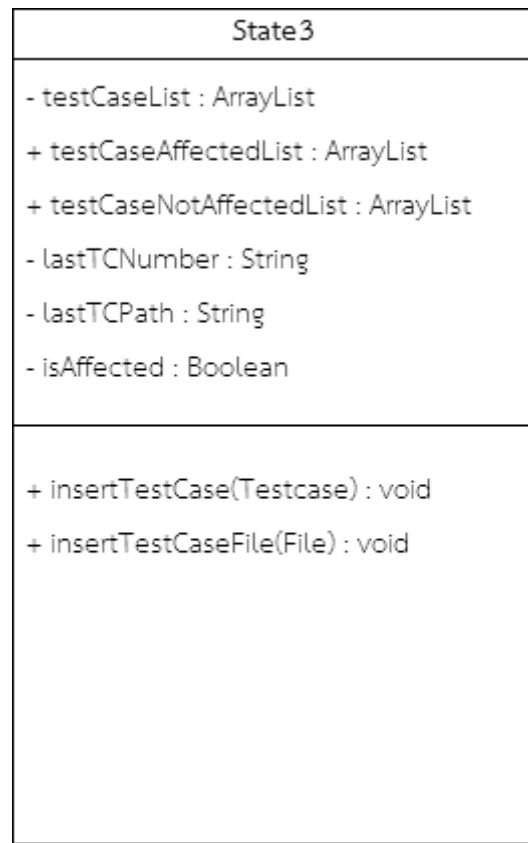
ภาพที่ 3-10 เมทอดของคลาส State1

State2
<ul style="list-style-type: none"> - listSql : ArrayList + listObjectSQLAfterProcess : ArrayList - listSourceCode : ArrayList - listPositionToHighlight : ArrayList - listMappingStatement : ArrayList - listParameter : ArrayList - listInsertSQLToHighlight : ArrayList - listAlertText : ArrayList - listSQLMappingTableBeforeProcess : ArrayList - listObjectSQLMappingTableAfterProcess : ArrayList - listSQLMappingParamName : ArrayList - listSQLMappingLineNumber : ArrayList + listSQLMappingParamWithLineNumber : ArrayList

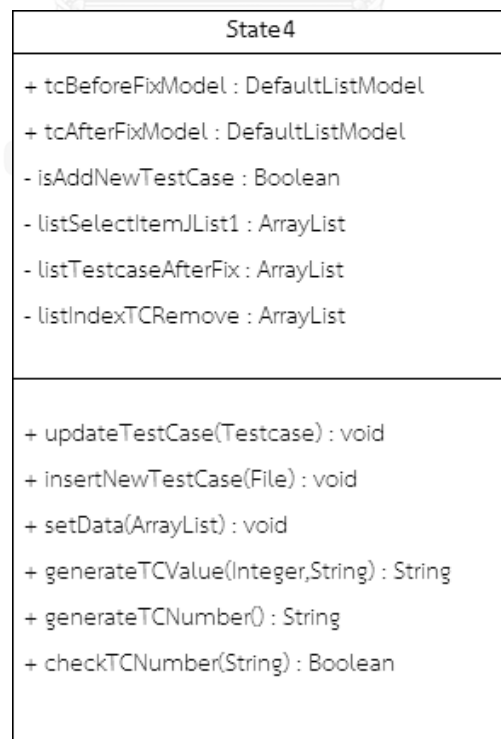
ภาพที่ 3-11 แอตทริบิวต์ของคลาส State2

<ul style="list-style-type: none"> + insertSourceCodeData(String,String,String) : void + insertMappingTableData(String,String,String,String) : void + insertSourceCodeFile(File) : void + insertMappingTableFile(File) : void + whereProcess(String) : ArrayList + findTextPositionToHighlight(String,String,Integer,String) : void + isInsertWithFieldName(String) : boolean
--

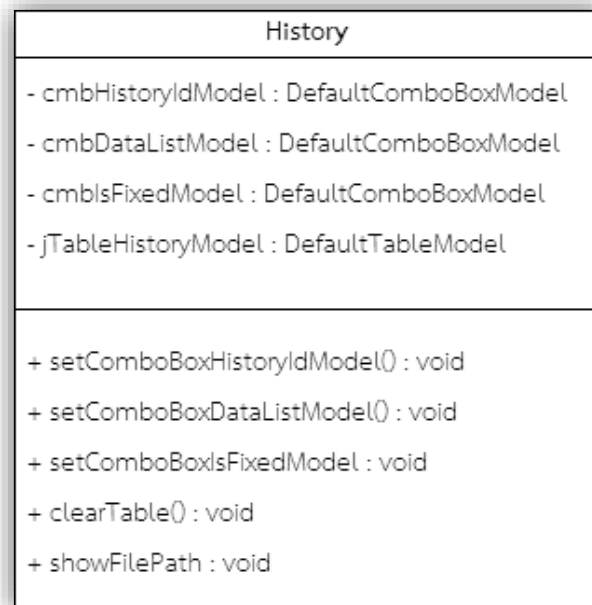
ภาพที่ 3-12 เมทอดของคลาส State2



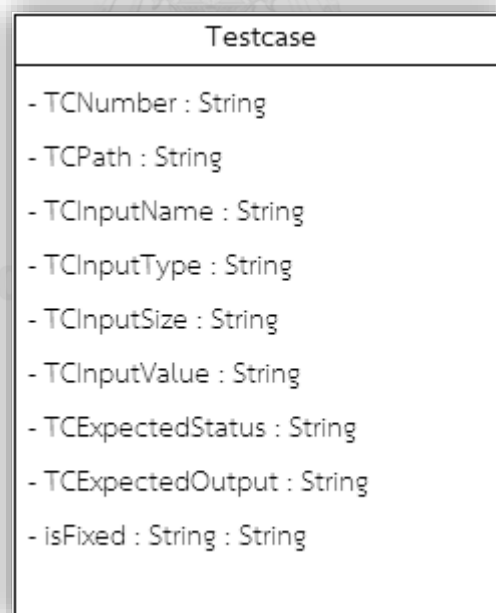
ภาพที่ 3-13 แอตทริบิวต์และเมทอดของคลาส State3



ภาพที่ 3-14 แอตทริบิวต์และเมทอดของคลาส State4



ภาพที่ 3-15 แอตทริบิวต์และเมทอดของคลาส History



ภาพที่ 3-16 แอตทริบิวต์ของคลาส Testcase

```

+ getTCExpectedOutput() : String
+ setTCExpectedOutput(String) : void
+ getTCNumber() : String
+ setTCNumber(String) : void
+ getTCPATH() : String
+ setTCPATH(String) : void
+ getTCInputName() : String
+ setTCInputName(String) : void
+ getTCInputType() : String
+ setTCInputType(String) : void
+ getTCInputSize() : String
+ setTCInputSize(String) : void
+ getTCInputValue() : String
+ setTCInputValue(String) : void
+ getTCExpectedStatus() : String
+ setTCExpectedStatus(String) : void
+ getIsFixed() : String
+ setIsFixed(String) : void

```

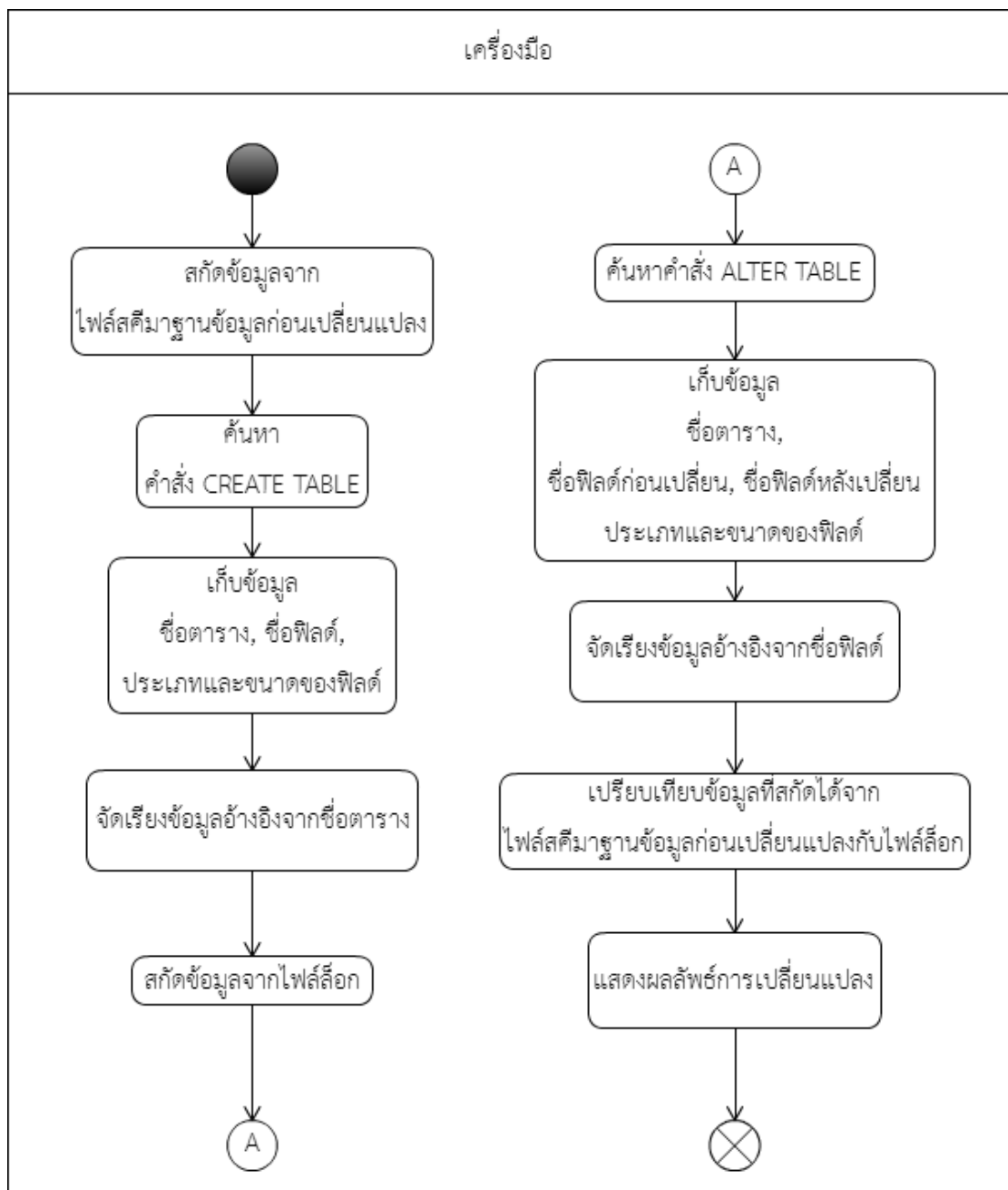
ภาพที่ 3-17 เมทอดของคลาส Testcase

3.2.3 แผนภาพกิจกรรม

แผนภาพกิจกรรมเป็นแผนภาพที่ใช้แสดงขั้นตอนการทำกิจกรรมของเครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูล อธิบายด้วยแผนภาพกิจกรรม 4 แผนภาพ คือ แผนภาพกิจกรรมวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล แสดงดังภาพที่ 3-18 แผนภาพกิจกรรมวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ แสดงดังภาพที่ 3-19 แผนภาพกิจกรรมวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ แสดงดังภาพที่ 3-20 แผนภาพกิจกรรมแก้ไขข้อมูลทดสอบ แสดงดังภาพที่ 3-21

1) แผนภาพกิจกรรมวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล

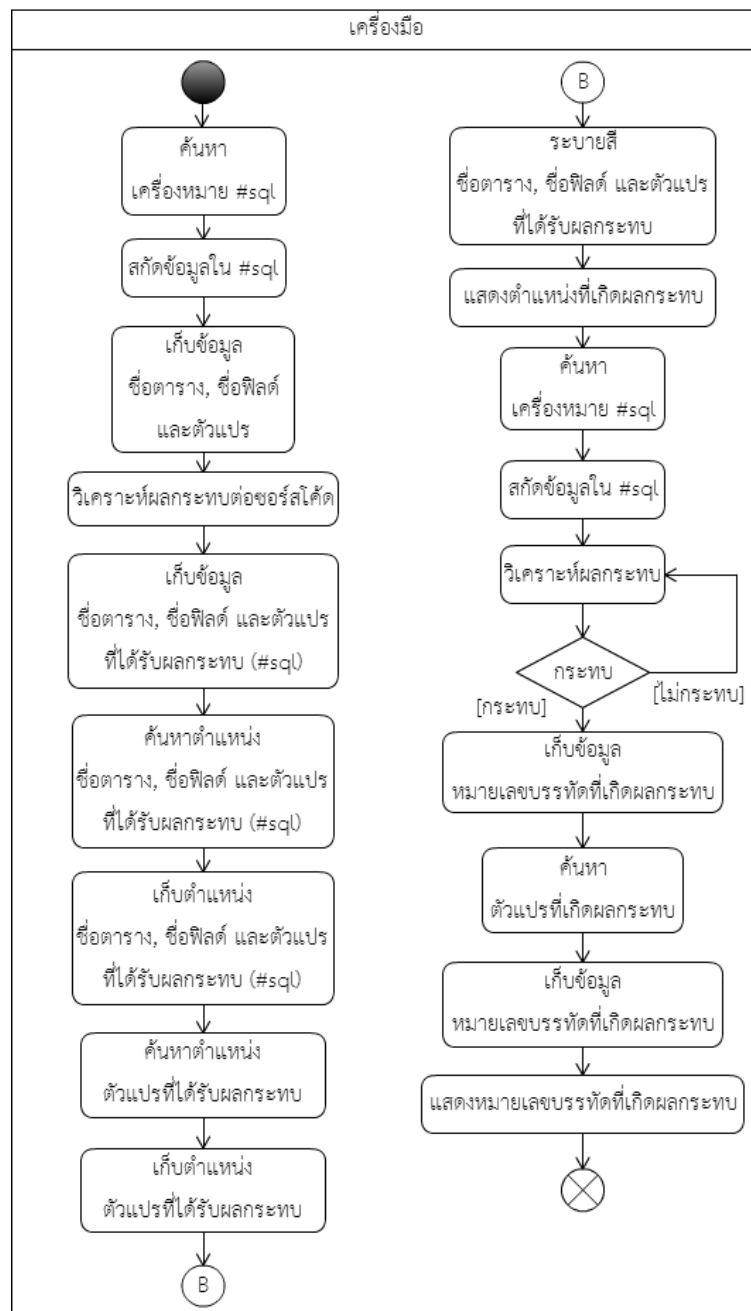
แผนภาพกิจกรรมวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล ดังภาพที่ 3-18 แสดงขั้นตอนของเครื่องมือ โดยเครื่องมือจะแสดงหน้าจอ เพื่อให้ผู้ทดสอบกดเลือกอัปโหลดไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลง และเครื่องมือจะสกัดข้อมูลจากไฟล์ที่อ่านได้ จากนั้นอัปโหลดล็อกไฟล์ ตามลำดับ และเครื่องมือจะสกัดข้อมูลจากล็อกไฟล์ที่อ่านได้ จากนั้นให้ผู้ทดสอบกดปุ่มวิเคราะห์ เพื่อให้เครื่องมือวิเคราะห์หากการเปลี่ยนแปลงที่เกิดขึ้นกับสคีมาฐานข้อมูล และเครื่องมือจะแสดงผลที่ได้ผ่านทางหน้าจอ



ภาพที่ 3-18 แผนภาพกิจกรรมวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล

2) แผนภาพกิจกรรมวิเคราะห์หาข้อผิดพลาดที่ได้รับผลกระทบ

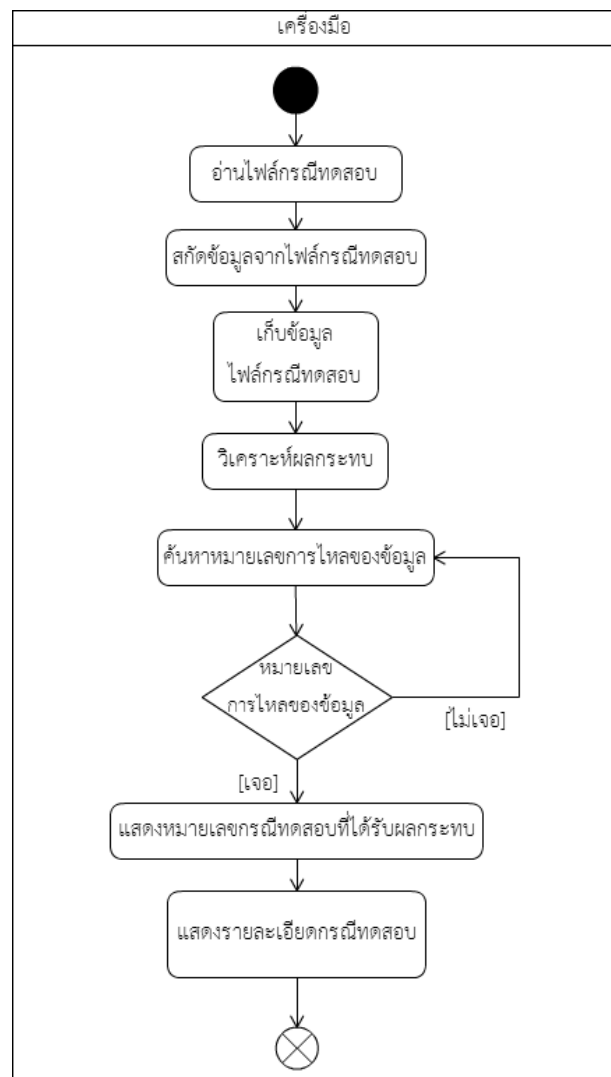
แผนภาพกิจกรรมวิเคราะห์หาข้อผิดพลาดที่ได้รับผลกระทบ ดังภาพที่ 3-19 แสดงขั้นตอนของเครื่องมือ โดยเครื่องมือจะแสดงหน้าจอ เพื่อให้ผู้ทดสอบเลือกอัปโหลดไฟล์ซอร์สโค้ด เครื่องมือจะสกัดข้อมูลซอร์สโค้ดเก็บไว้ จากนั้นกดปุ่มวิเคราะห์เพื่อให้เครื่องมือนำข้อมูลซอร์สโค้ดที่สกัดมาประมวลผลเพื่อวิเคราะห์และแสดงตำแหน่งของข้อผิดพลาดที่ได้รับผลกระทบผ่านทางหน้าจอ ลำดับถัดไปคือ ผู้ทดสอบเลือกอัปโหลดไฟล์แมปปีง เครื่องมือจะสกัดข้อมูลจากไฟล์แมปปีง จากนั้นกดปุ่มวิเคราะห์ เครื่องมือจะแสดงผลลัพธ์หมายเลขบรรทัดของข้อผิดพลาดที่ได้รับผลกระทบผ่านทางหน้าจอ



ภาพที่ 3-19 แผนภาพกิจกรรมวิเคราะห์หาข้อผิดพลาดที่ได้รับผลกระทบ

3) แผนภาพกิจกรรมวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ

แผนภาพกิจกรรมวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ ดังภาพที่ 3-20 แสดงขั้นตอนของเครื่องมือ โดยเครื่องมือจะแสดงหน้าจอ เพื่อให้ผู้ทดสอบเลือกอัปโหลดไฟล์กรณีทดสอบ เครื่องมือจะสกัดข้อมูลจากไฟล์กรณีทดสอบเก็บไว้ จากนั้นกดปุ่มวิเคราะห์ เครื่องมือจะค้นหาหมายเลขการไหลของข้อมูล เพื่อค้นหาหมายเลขที่ได้รับผลกระทบจากกรณีทดสอบ จากนั้นเครื่องมือจะแสดงผลลัพธ์ คือหมายเลขกรณีทดสอบที่ได้รับผลกระทบ และผู้ทดสอบสามารถคลิกดูรายละเอียดของกรณีทดสอบที่ได้รับผลกระทบได้ผ่านทางหน้าจอ

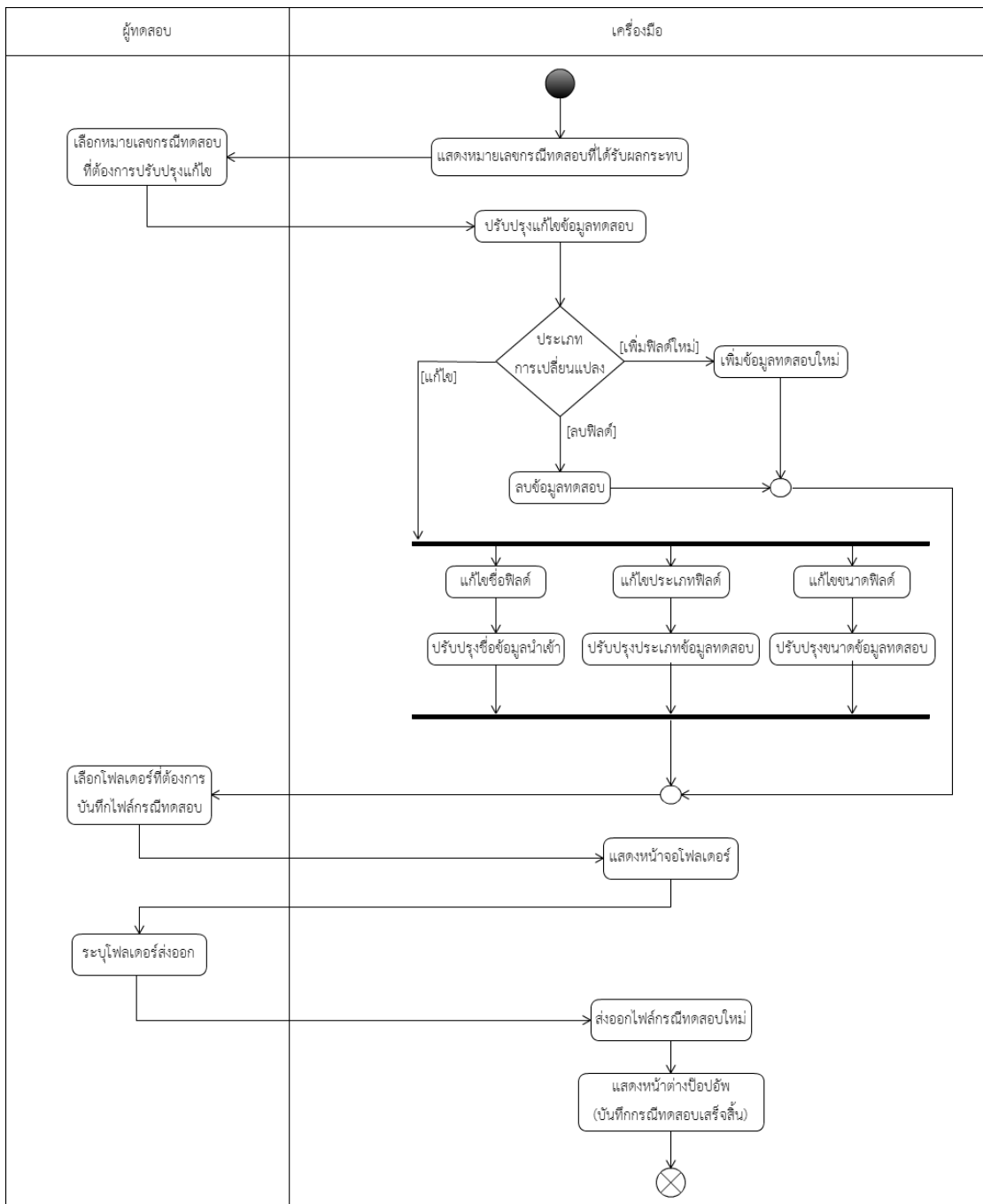


แผนภาพที่ 3-20 แผนภาพกิจกรรมวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ

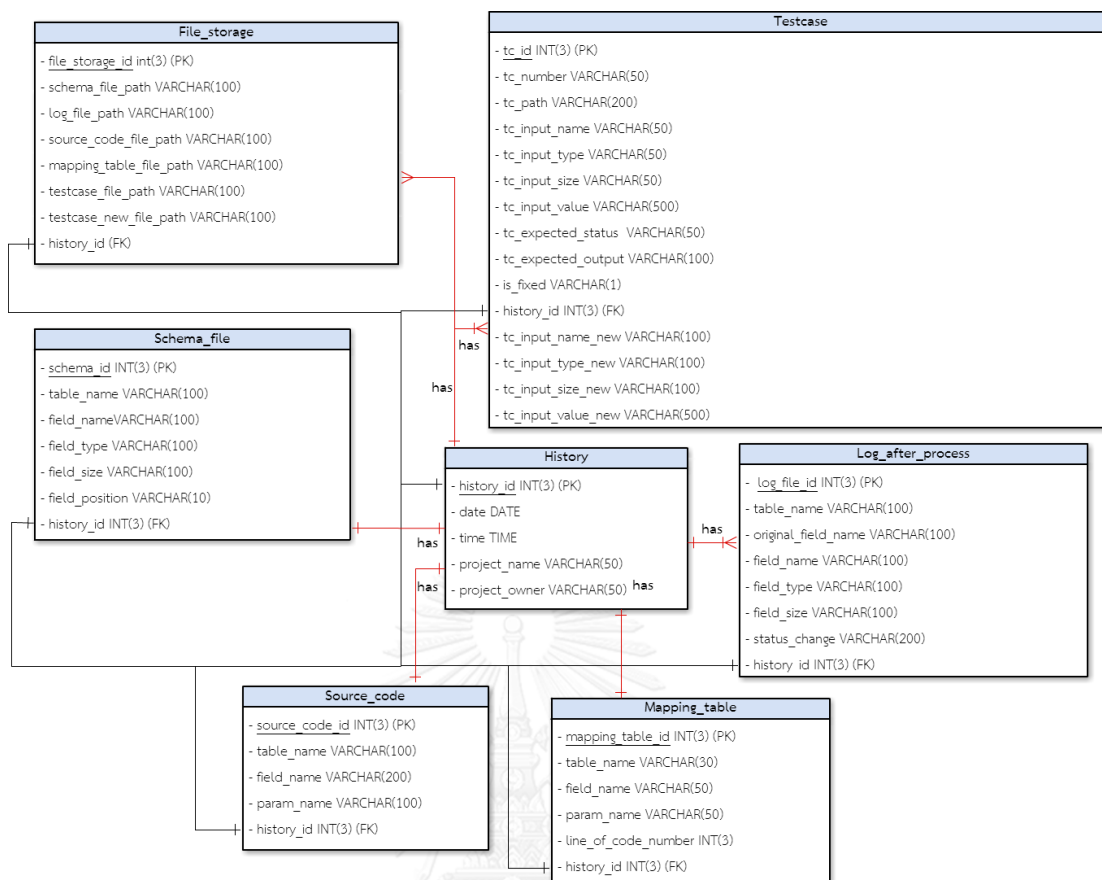
4) แผนภาพกิจกรรมแก้ไขข้อมูลทดสอบ

แผนภาพกิจกรรมแก้ไขข้อมูลทดสอบ ดังภาพที่ 3-21 แสดงขั้นตอนของเครื่องมือ โดยเครื่องมือจะแสดงหมายเลขกรณีทดสอบที่ได้รับผลกระทบ จากขั้นตอนก่อนหน้า โดยให้ผู้ทดสอบ

เลือกหมายเลขกรณีทดสอบที่ต้องการปรับปรุงแก้ไข จากนั้นผู้ทดสอบกดปุ่มปรับปรุงแก้ไข เครื่องมือจะแสดงหมายเลขกรณีทดสอบที่ปรับปรุงแก้ไขข้อมูลทดสอบเสร็จสิ้นผ่านทางหน้าจอ และสามารถคลิกดูรายละเอียดกรณีทดสอบที่ปรับปรุงแก้ไขแล้วได้ พร้อมทั้งผู้ทดสอบสามารถส่งออกกรณีทดสอบใหม่ได้ ในรูปแบบไฟล์เอกสาร (.xlsx) ผ่านทางหน้าจอ โดยผู้ทดสอบเลือกโฟลเดอร์ที่ต้องการจะบันทึก และกดปุ่มส่งออกกรณีทดสอบใหม่ เมื่อบันทึกกรณีทดสอบใหม่เสร็จสิ้น เครื่องมือจะแสดงข้อความแจ้งเตือนให้ผู้ทดสอบทราบ



ภาพที่ 3-21 แผนภาพกิจกรรมแก้ไขข้อมูลทดสอบ



ภาพที่ 3-22 โครงสร้างฐานข้อมูลของเครื่องมือ

3.2.4 โครงสร้างฐานข้อมูล

โครงสร้างฐานข้อมูลของเครื่องมือ สามารถแสดงได้ดังแผนภาพอีอาร์ ดังภาพที่ 3-22 โดยรายละเอียดของโครงสร้างฐานข้อมูลเป็นดังนี้

- 1) ตาราง File_storage เป็นตารางที่ใช้เก็บตำแหน่งที่อยู่ของไฟล์
- 2) ตาราง Schema_file เป็นตารางที่ใช้เก็บรายละเอียดข้อมูลที่ได้จากไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลง
- 3) ตาราง Source_code เป็นตารางที่ใช้เก็บรายละเอียดข้อมูลที่ได้จากวิเคราะห์ไฟล์ซอร์สโค้ด
- 4) ตาราง Testcase เป็นตารางที่ใช้เก็บรายละเอียดข้อมูลที่ได้จากการวิเคราะห์กรณีทดสอบ
- 5) ตาราง History เป็นตารางที่ใช้เก็บประวัติและข้อมูลต่างๆ ของผู้ใช้งานเครื่องมือ

6) ตาราง Log_after_process เป็นตารางที่เก็บรายละเอียดข้อมูลการเปลี่ยนแปลงที่เกิดขึ้นที่ได้จากการวิเคราะห์ล็อกไฟล์

7) ตาราง Mapping_table เป็นตารางที่เก็บรายละเอียดข้อมูลที่วิเคราะห์ได้จากไฟล์แมปปิง



บทที่ 4

การพัฒนาเครื่องมือ

ในบทการพัฒนาเครื่องมือจะกล่าวถึงสภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ รวมทั้งโครงสร้างส่วนต่อประสานกับผู้ทดสอบของเครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูล ซึ่งมีรายละเอียดดังต่อไปนี้

4.1 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ

สภาพแวดล้อมที่ใช้ทดสอบเครื่องมือ มีรายละเอียดดังต่อไปนี้

4.1.1 ฮาร์ดแวร์

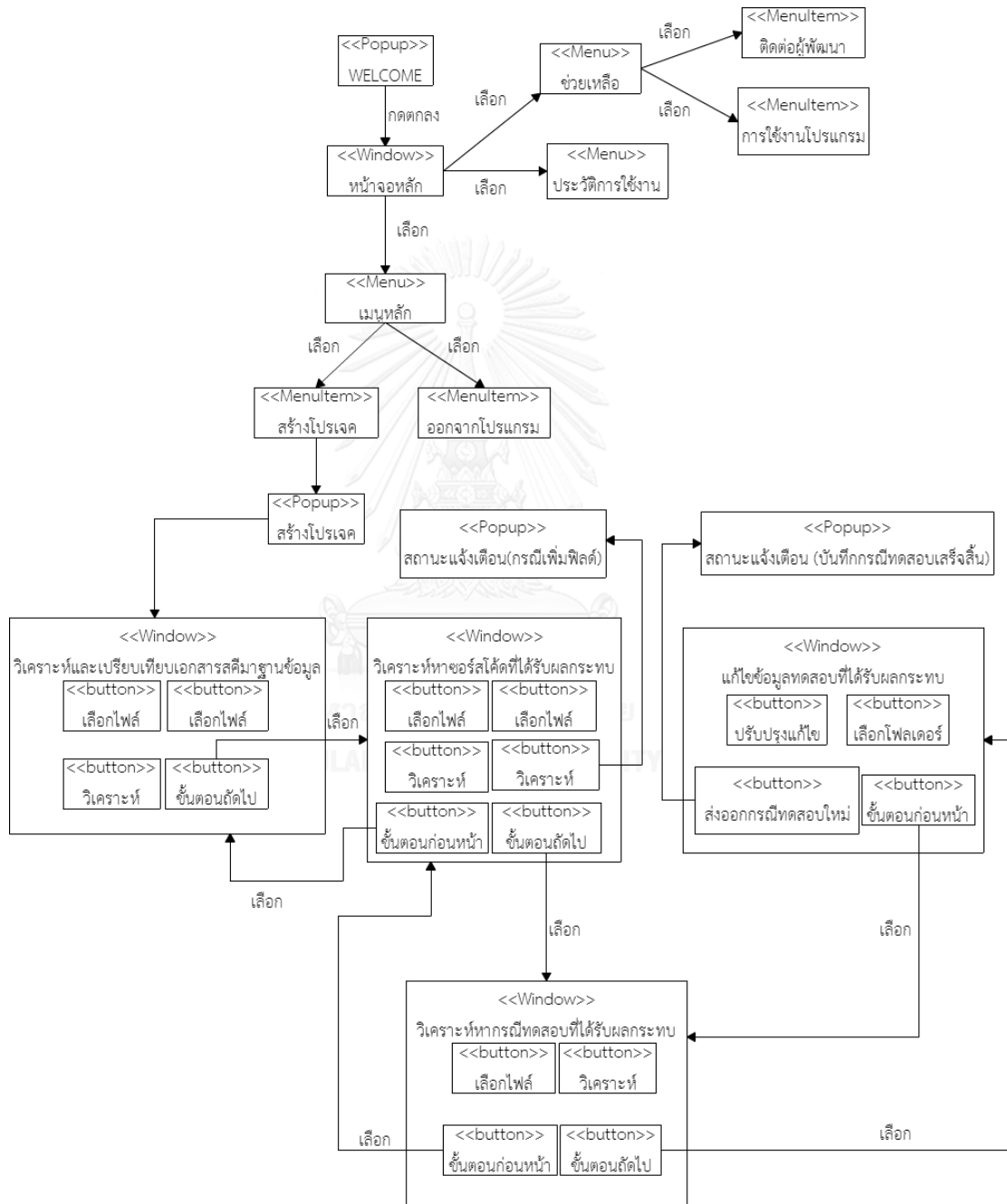
- 1) เครื่องคอมพิวเตอร์แบบโน้ตบุ๊ก (Notebook) หน่วยประมวลผลอินเทลคอร์ไอที 1.8 กิกะเฮิร์ตซ์ (Intel Core i3 GHz)
- 2) หน่วยความจำสำรอง 4.0 กิกะไบต์ (4.0 GB)
- 3) ฮาร์ดดิสก์ (Harddisk) 500 กิกะไบต์ (500 GB)

4.1.2 ซอฟต์แวร์

- 1) ระบบปฏิบัติการ (Operating System) ไมโครซอฟท์วินโดวส์ (Microsoft Windows 8)
- 2) เนทเบินไอดีอี เวอร์ชัน 8.0 (Netbeans IDE 8.0)
- 3) จาวาเจดีเค เวอร์ชัน 7.6 (Java JDK 7.6)
- 4) ฐานข้อมูลมายเอสคิวแอล เวอร์ชัน 5.6.1 (MySQL 5.6.1)
- 5) พีโอไอ ไลบรารี เวอร์ชัน 3.12 (POI 3.12)
- 6) มายเอสคิวแอลคอนเนคเตอร์ เวอร์ชัน 5.1.34 (MySQL Connector 5.1.34)

4.2 โครงสร้างส่วนต่อประสานกับผู้ใช้ของเครื่องมือ

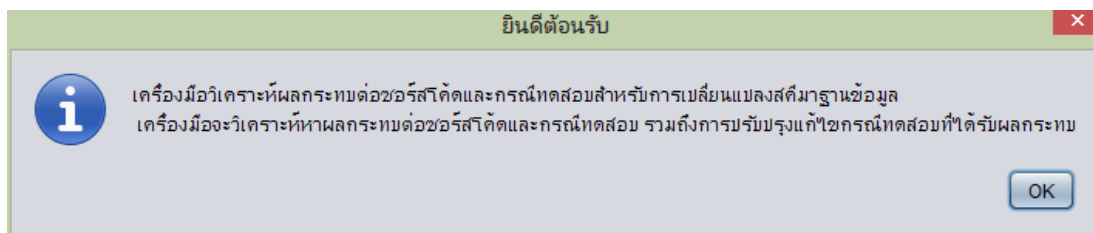
โครงสร้างส่วนต่อประสานของเครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบ สำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูล อธิบายด้วยแผนภาพ Window Navigation ซึ่งใช้อธิบายถึงความสัมพันธ์ต่างๆของส่วนต่อประสานทั้งหมดในโปรแกรมแบ่งตามหน้าที่ต่างๆ ดังภาพที่ 4.1



ภาพที่ 4-1 แผนภาพ Window Navigation

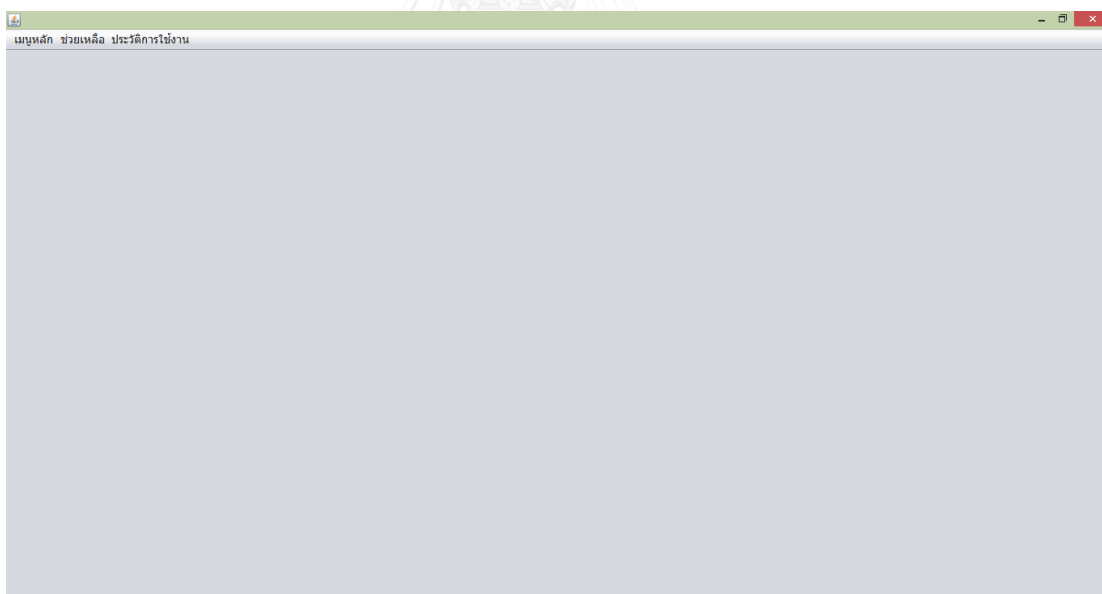
จากภาพที่ 4-1 แผนภาพ Window Navigation แสดงส่วนประกอบของส่วนต่อประสานของเครื่องมือ ซึ่งจะประกอบไปด้วย หน้าต่าง เมนู แต่ละส่วนประกอบมีรายละเอียดดังต่อไปนี้

- 1) WELCOME เป็นป๊อปอัพแรกที่แสดงขึ้นมาเมื่อเปิดเครื่องมือ เพื่อแสดงรายละเอียดของเครื่องมือดังภาพที่ 4-2



ภาพที่ 4-2 หน้าต่างป๊อปอัพ WELCOME

- 2) หน้าจอหลัก เป็นหน้าต่างหลักของเครื่องมือ ซึ่งประกอบด้วย 3 เมนู ได้แก่ เมนูหลัก เมนูช่วยเหลือ และเมนูประวัติการใช้งาน ดังภาพที่ 4-3



ภาพที่ 4-3 หน้าจอหลัก

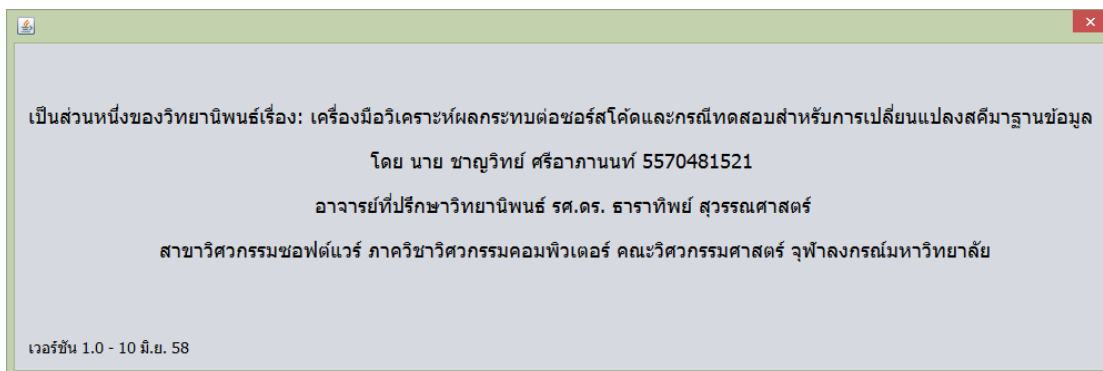
- 2.1) เมนูหลัก เป็นเมนูที่ประกอบไปด้วยเมนูย่อย 2 เมนู คือ

2.1.1) สร้างโปรเจค เป็นเมนูที่ใช้สร้างโปรเจคใหม่

2.1.2) ออกจากโปรแกรม เป็นเมนูที่ออกจากโปรแกรม

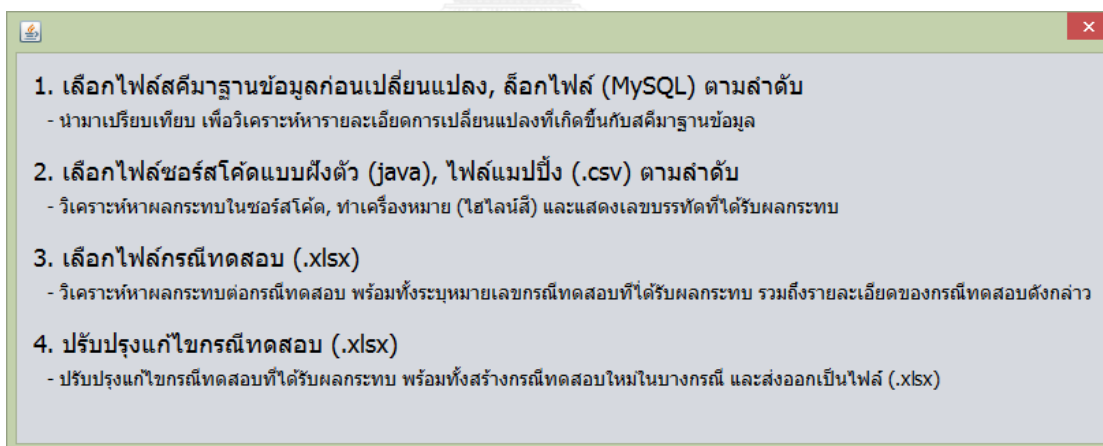
2.2) ช่วยเหลือ เป็นเมนูที่ประกอบไปด้วยเมนูย่อย 2 เมนู คือ

2.2.1) ติดต่อผู้พัฒนา เป็นเมนูที่อธิบายรายละเอียดเกี่ยวกับผู้พัฒนาเครื่องมือ ดังภาพที่ 4-4



ภาพที่ 4-4 หน้าต่างติดต่อผู้พัฒนา

2.2.2) การใช้งานโปรแกรม เป็นหน้าต่างที่อธิบายขั้นตอนการใช้งานของเครื่องมือ ดังภาพที่ 4-5



ภาพที่ 4-5 หน้าต่างการใช้งานโปรแกรม

2.3) ประวัติการใช้งาน เป็นหน้าต่างที่แสดงรายละเอียดการใช้งานของเครื่องมือ ดังภาพที่ 4-6

ประวัติการใช้งานเครื่องมือ

ครั้งที่ 1 รายการ ไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลง ค้นหา

วันที่หารายการ : 2558-06-03 เวลา 08:51:16 น. ชื่อโปรเจค : CW Project ชื่อผู้สร้างโปรเจค : Chanwit

No.	TableName	FieldName	FieldType	FieldSize	FieldPosition
1	member	m_id	int	3	1
2	member	m_surname	varchar	50	2
3	member	m_lastname	varchar	50	3
4	member	m_password	varchar	50	4
5	member	m_email	varchar	50	5
6	member	m_address	varchar	50	6
7	member	m_cash	varchar	100	7
8	security	s_cname	varchar	50	1
9	security	s_current_price	varchar	100	2
10	security	s_offer_price	varchar	100	3
11	security	s_bid_price	varchar	100	4
12	security	s_symbol	varchar	10	5

ไฟล์เดอริทีเก็บไฟล์ : D:/CWTool/filestorage/25580603085116/schemaFile.sql

ภาพที่ 4-6 หน้าต่างประวัติการใช้งาน

3) สร้างโปรเจค เป็นป๊อปอัพที่ให้ผู้ใช้กรอกข้อมูล ชื่อโปรเจคใหม่ และชื่อผู้สร้างโปรเจค
ดังภาพที่ 4-7

สร้างโปรเจค

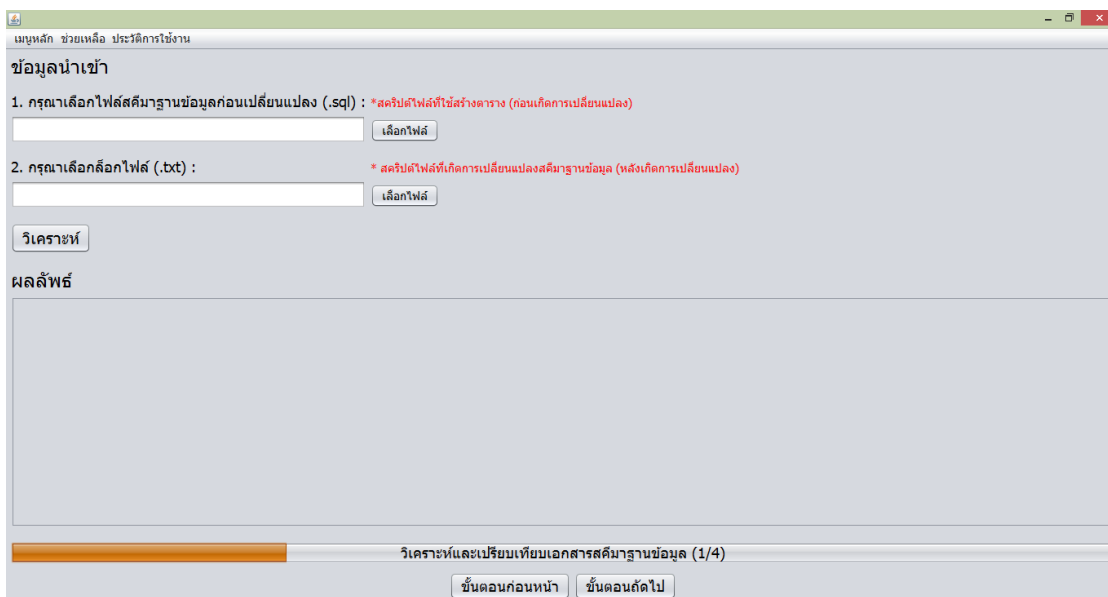
กรณาระบุชื่อโปรเจค :

กรณาระบุชื่อผู้สร้างโปรเจค :

ตกลง

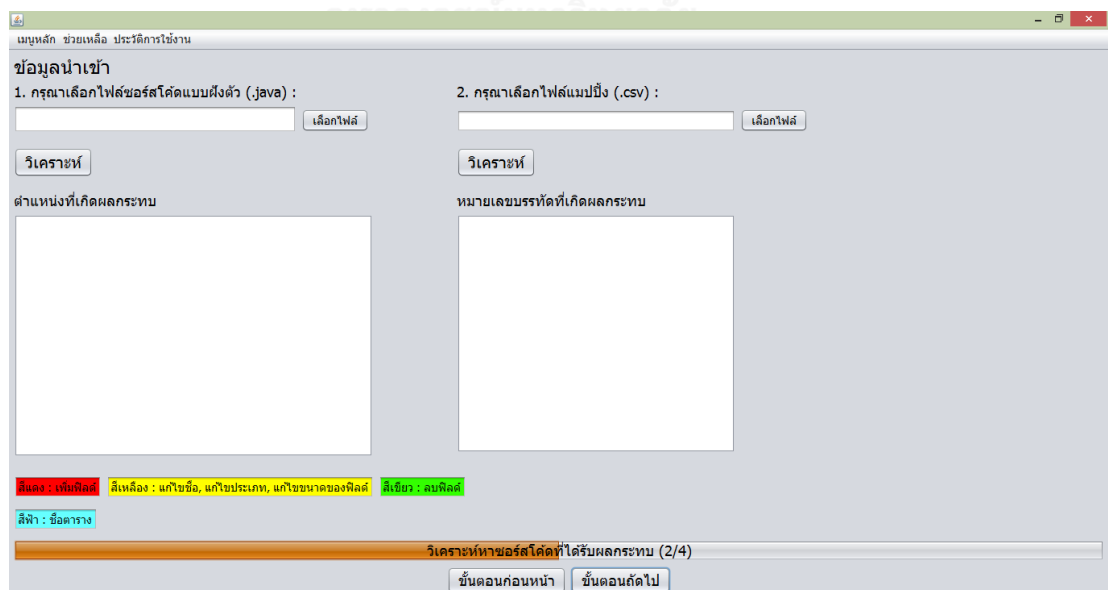
ภาพที่ 4-7 หน้าต่างป๊อปอัพสร้างโปรเจค

3) หน้าต่างวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล เป็นส่วนที่ใช้วิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล เพื่อหาการเปลี่ยนแปลงที่เกิดขึ้น ดังภาพที่ 4-8



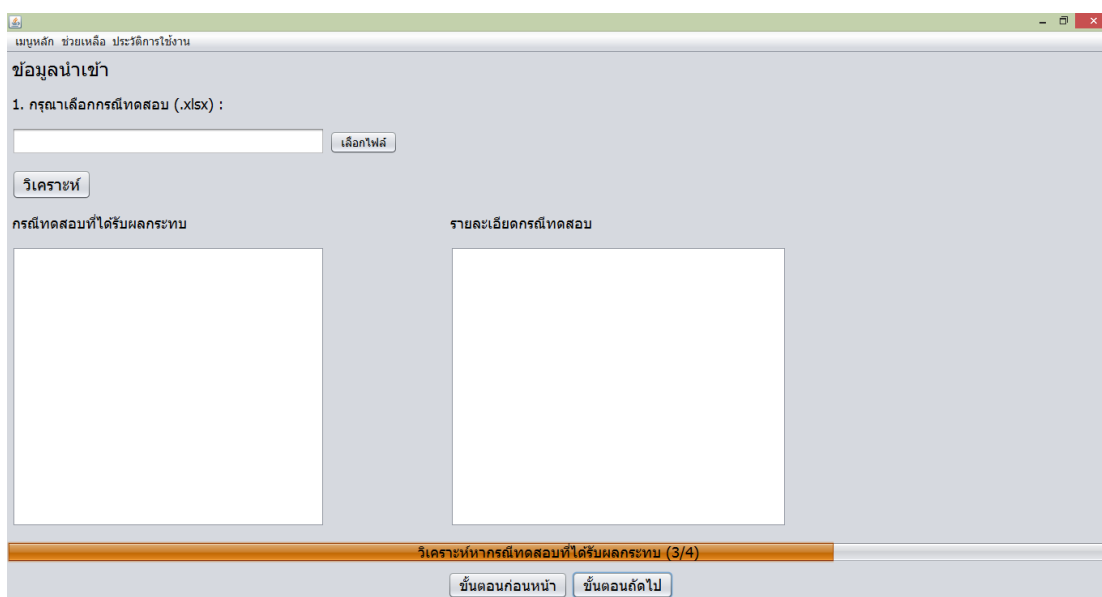
ภาพที่ 4-8 หน้าต่างวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล

4) หน้าต่างวิเคราะห์หาข้อผิดพลาดที่ได้รับผลกระทบ เป็นส่วนที่วิเคราะห์หาข้อผิดพลาดที่ได้รับผลกระทบจากการเปลี่ยนแปลงที่เกิดขึ้น พร้อมทั้งแสดง ตำแหน่งที่เกิดผลกระทบ ด้วยการระบายสี (Highlight) และเลขบรรทัดที่เกิดผลกระทบ ผ่านทางหน้าจอของเครื่องมือ ดังภาพที่ 4-9



ภาพที่ 4-9 หน้าต่างวิเคราะห์หาข้อผิดพลาดที่ได้รับผลกระทบ

5) หน้าต่างวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ เป็นส่วนที่วิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ โดยแสดงผลลัพธ์คือ หมายเลขของกรณีทดสอบที่ได้รับผลกระทบ และแสดงรายละเอียดของกรณีทดสอบที่ได้รับผลกระทบ ดังภาพที่ 4-10

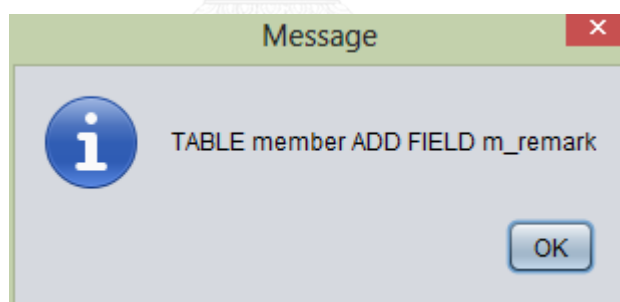


ภาพที่ 4-10 หน้าต่างวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ

6) หน้าต่างแก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ เป็นส่วนที่ปรับปรุงแก้ไขกรณีทดสอบที่ได้รับผลกระทบ ในส่วนของข้อมูลทดสอบ รวมถึงการสร้างกรณีทดสอบใหม่ในบางกรณี เพื่อให้กรณีทดสอบที่ได้รับผลกระทบที่ได้รับการปรับปรุง สามารถนำไปใช้งานต่อได้ทันที ดังภาพที่ 4-11

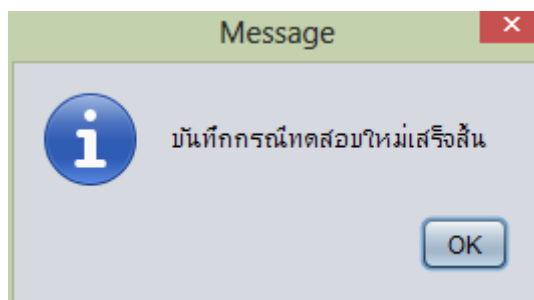
ภาพที่ 4-11 หน้าต่างแก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ

7) สถานะแจ้งเตือน (กรณีเพิ่มฟิลด์) เป็นป๊อปอัพสถานะแจ้งเตือนว่าซอร์สโค้ดได้รับผลกระทบจากการเปลี่ยนแปลงกรณีเพิ่มฟิลด์ ดังภาพที่ 4-12



ภาพที่ 4-12 หน้าต่างป๊อปอัพสถานะแจ้งเตือน (กรณีเพิ่มฟิลด์)

8) สถานะแจ้งเตือน (บันทึกกรณีทดสอบเสร็จสิ้น) เป็นป๊อปอัพสถานะแจ้งเตือนว่าส่งออกกรณีทดสอบใหม่เสร็จสิ้น ดังภาพที่ 4-13



ภาพที่ 4-13 หน้าต่างป๊อปอัพสถานะแจ้งเตือน (บันทึกกรณีทดสอบใหม่เสร็จสิ้น)



บทที่ 5

การทดสอบเครื่องมือ

หัวข้อนี้กล่าวถึงการทดสอบเครื่องมือเพื่อตรวจสอบความถูกต้อง โดยจะกล่าวถึงสภาพแวดล้อมที่ใช้ในการทดสอบ รายละเอียดของการทดสอบเครื่องมือ และผลการทดสอบเครื่องมือ โดยมีรายละเอียดดังต่อไปนี้

5.1 สภาพแวดล้อมที่ใช้ทดสอบ

สภาพแวดล้อมที่ใช้ทดสอบเครื่องมือ มีรายละเอียดดังต่อไปนี้

5.1.1 ฮาร์ดแวร์

- 1) เครื่องคอมพิวเตอร์แบบโน้ตบุ๊ก (Notebook) หน่วยประมวลผลอินเทลคอร์ไอทรี 1.8 กิกะเฮิร์ตซ์ (Intel Core i3 GHz)
- 2) หน่วยความจำสำรอง 4.0 กิกะไบต์ (4.0 GB)
- 3) ฮาร์ดดิสก์ (Harddisk) 500 กิกะไบต์ (500 GB)

5.1.2 ซอฟต์แวร์

- 1) ระบบปฏิบัติการ (Operating System) ไมโครซอฟท์วินโดวส์ (Microsoft Windows 8)
- 2) เนทเบินไอดีอี เวอร์ชัน 8.0 (Netbeans IDE 8.0)
- 3) จาวาเจดีเค เวอร์ชัน 7.6 (Java JDK 7.6)
- 4) ฐานข้อมูลมายเอสคิวแอล เวอร์ชัน 5.6.1 (MySQL 5.6.1)
- 5) พีโอไอ ไลบรารี เวอร์ชัน 3.12 (POI 3.12)
- 6) มายเอสคิวแอลคอนเนคเตอร์ เวอร์ชัน 5.1.34 (MySQL Connector 5.1.34)

5.2 การทดสอบเครื่องมือ

การทดสอบเครื่องมือ เพื่อตรวจสอบว่าเครื่องมือสามารถวิเคราะห์ผลกระทบต่อซอร์สโค้ด และกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูล เช่น วิเคราะห์หาการเปลี่ยนแปลง ตำแหน่งที่เกิดผลกระทบในซอร์สโค้ด วิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ และสามารถปรับปรุงแก้ไข ข้อมูลทดสอบ รวมถึงการสร้างกรณีทดสอบใหม่ในบางกรณี ได้หรือไม่ โดยการทดสอบนี้จะกระทำกับระบบจำลองการทำงานที่สร้างขึ้นเพื่อทดสอบเครื่องมือนี้โดยเฉพาะ ซึ่งระบบจำลองนั้นประกอบไปด้วย 3 ระบบ คือ ระบบการลงทุน ระบบสั่งซื้อตัวออนไลน์ และระบบสั่งซื้อสินค้าออนไลน์ แต่ละระบบมีรายละเอียดดังนี้

1) ระบบการลงทุน

ระบบนี้จะจำลองระบบการลงทุน โดยระบบนี้สามารถ เพิ่มสมาชิกใหม่ สมาชิกล็อกอิน ดูราคาหุ้นปัจจุบัน และลบข้อมูลสมาชิก เป็นต้น ระบบนี้ประกอบไปด้วย ไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลง 1 ไฟล์ ไฟล์ล็อก 1 ไฟล์ ไฟล์ซอร์สโค้ด 6 ไฟล์ ไฟล์แมปปิง 6 ไฟล์ และไฟล์กรณีทดสอบ 1 ไฟล์ ประกอบไปด้วย 10 กรณีทดสอบ สามารถดูรายละเอียดเพิ่มเติมได้ในภาคผนวก ข.

2) ระบบจองตั๋วเครื่องบินโดยสารออนไลน์

ระบบนี้จะจำลองระบบการจองตั๋วเครื่องบินโดยสารออนไลน์ โดยระบบนี้สามารถ ดูราคาตั๋วเครื่องบินปัจจุบัน จองตั๋ว แก้ไขข้อมูลผู้โดยสาร ลบข้อมูลผู้โดยสาร เป็นต้น ระบบนี้ประกอบไปด้วย ไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลง 1 ไฟล์ ไฟล์ล็อก 1 ไฟล์ ไฟล์ซอร์สโค้ด 4 ไฟล์ ไฟล์แมปปิง 4 ไฟล์ และไฟล์กรณีทดสอบ 1 ไฟล์ ประกอบไปด้วย 8 กรณีทดสอบ สามารถดูรายละเอียดเพิ่มเติมได้ในภาคผนวก ข.

3) ระบบสั่งซื้อสินค้าออนไลน์

ระบบนี้จะจำลองระบบการสั่งซื้อสินค้าออนไลน์ โดยระบบนี้สามารถ แสดงรายละเอียดราคา สินค้า เช็การชำระเงิน เป็นต้น ระบบนี้ประกอบไปด้วย ไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลง 1 ไฟล์ ไฟล์ล็อก 1 ไฟล์ ไฟล์ซอร์สโค้ด 2 ไฟล์ ไฟล์แมปปิง 2 ไฟล์ และไฟล์กรณีทดสอบ 1 ไฟล์ ประกอบไปด้วย 4 กรณีทดสอบ สามารถดูรายละเอียดเพิ่มเติมได้ในภาคผนวก ข.

5.3 ผลการทดสอบ

5.3.1 การทดสอบการวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบ

เมื่อได้ทดสอบเครื่องมือโดยใช้ไฟล์ซอร์สโค้ดและกรณีทดสอบที่สร้างขึ้นเฉพาะ ได้รับผลการทดสอบ ดังตารางที่ 5.1 และสามารถดูรายละเอียดเพิ่มเติมได้ในภาคผนวก ข.

ตารางที่ 5.1 ผลการทดสอบการวิเคราะห์หาการเปลี่ยนแปลงและผลกระทบกับระบบจำลอง

ชื่อระบบ	ชื่อไฟล์ซอร์สโค้ด	จำนวนกรณีทดสอบที่ไม่ได้รับผลกระทบ	จำนวนกรณีทดสอบที่ปรับปรุงแก้ไข	จำนวนกรณีทดสอบที่สร้างใหม่	จำนวนกรณีทดสอบที่ถูกลบ
ระบบการลงทะเบียน	deleteMember	8	2	1	0
	memberLogin	10	0	0	0
	newMember	8	2	1	1
	newMember_no_fieldname	8	2	1	1
	selectBySymbol	8	2	1	0
	updateMember	10	0	0	0
ระบบจองตั๋วเครื่องบินโดยสารออนไลน์	delPassenger	8	0	0	0
	reserveTicket	6	2	0	2
	selectFlight	8	0	0	0
	updatePassenger	6	2	1	2
ระบบสั่งซื้อสินค้าออนไลน์	checkPaid	2	2	0	2
	showProduct	4	0	0	0

5.4 สรุปผลการทดสอบ

จากการทดสอบเครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูล ได้ผลการทดสอบดังตารางที่ 5.1 ซึ่งประกอบไปด้วย ชื่อระบบ ชื่อไฟล์ซอร์สโค้ด จำนวนกรณีทดสอบที่ไม่ได้รับผลกระทบ จำนวนกรณีทดสอบที่ปรับปรุงแก้ไข จำนวนกรณีทดสอบที่สร้างใหม่ และจำนวนกรณีทดสอบที่ถูกลบ หมายความว่า ในการทดสอบแต่ละครั้งนั้น เครื่องมือจะวิเคราะห์ไฟล์ซอร์สโค้ดได้ครั้งละ 1 ไฟล์ เช่น ไฟล์ newMember ของระบบการลงทุน เมื่อวิเคราะห์หาผลกระทบต่อซอร์สโค้ดแล้ว ย่อมมีการเปลี่ยนแปลงที่ส่งผลกระทบต่อกรณีทดสอบเช่นกัน ประกอบไปด้วย จำนวนกรณีทดสอบที่ไม่ได้รับผลกระทบ คือ 8 กรณีทดสอบ จำนวนกรณีทดสอบที่ปรับปรุงแก้ไข คือ 2 กรณีทดสอบ จำนวนกรณีทดสอบที่สร้างใหม่ คือ 1 กรณีทดสอบ และจำนวนกรณีทดสอบที่ถูกลบ คือ 1 ข้อมูลทดสอบในกรณีทดสอบ

เครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูลสามารถสรุปผลการทดสอบได้ว่า เครื่องมือสามารถวิเคราะห์หาการเปลี่ยนแปลงและผลกระทบ รวมถึงการปรับปรุงแก้ไขข้อมูลทดสอบและการสร้างกรณีทดสอบใหม่ได้

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

จากการศึกษา วิจัย และพัฒนาเครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบ สำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูล สามารถสรุปผลการวิจัย ข้อจำกัดของเครื่องมือ และแนวทางในการพัฒนาต่อ โดยมีรายละเอียดดังนี้

6.1 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอเครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูล โดยวิเคราะห์การเปลี่ยนแปลงที่เกิดขึ้นจากล็อกไฟล์ จากนั้นจึงดำเนินการสร้างเครื่องมือตามวิธีการที่ได้นำเสนอไว้ ซึ่งเครื่องมือสามารถวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบทำงานได้อย่างถูกต้องตามวิธีการที่ได้นำเสนอไว้ โดยเครื่องมือสามารถวิเคราะห์การเปลี่ยนแปลง วิเคราะห์หาผลกระทบต่อซอร์สโค้ดและกรณีทดสอบได้ อีกทั้งยังสามารถสร้างกรณีทดสอบใหม่ในบางกรณี พร้อมทั้งปรับปรุงแก้ไขเฉพาะข้อมูลทดสอบของกรณีทดสอบที่ได้รับผลกระทบให้สามารถพร้อมใช้งานต่อได้ สามารถนำไปใช้ในการทดสอบได้จริง

6.2 ข้อจำกัดของเครื่องมือ

เครื่องมือวิเคราะห์หาผลกระทบต่อซอร์สโค้ดและกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูล มีข้อจำกัดดังต่อไปนี้

1. เครื่องมือสามารถรับไฟล์ที่ใช้วิเคราะห์ข้อมูลได้ 4 ชนิด คือ ไฟล์ล็อก ไฟล์เอกสารสคีมาฐานข้อมูลก่อนเปลี่ยนแปลง ซอร์สโค้ดแบบฝังตัว และกรณีทดสอบ
2. ไฟล์ล็อกต้องอยู่ในรูปแบบไฟล์เอกสาร และไฟล์เอกสารสคีมาฐานข้อมูลก่อนเปลี่ยนแปลงต้องอยู่ในรูปแบบเอสคิวแอลสคริปต์
3. ซอร์สโค้ดแบบฝังตัวในภาษาจาวา และกรณีทดสอบต้องอยู่ในรูปแบบที่งานวิจัยกำหนดขึ้นเท่านั้น โดยซอร์สโค้ดแบบฝังตัวนั้นจะมีการฝังส่วนที่มีการเชื่อมต่อกับฐานข้อมูลด้วยเครื่องหมาย #sql ส่วนกรณีทดสอบนั้นอยู่ในรูปแบบไฟล์เอกซ์เซลล์ (.xlsx) โดยงานวิจัยนี้จะไม่ครอบคลุมในการ

แก้ไขซอร์สโค้ดที่ได้รับผลกระทบ แต่จะแก้ไขปรับปรุงข้อมูลทดสอบเฉพาะกรณีทดสอบที่ได้รับผลกระทบเท่านั้น

4. เครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดแบบฝังตัวในภาษาจาวาได้ครั้งละ 1 คลาส
5. ผู้ใช้ต้องระบุและนำเข้าไฟล์แมปปิง ซึ่งอธิบายความสัมพันธ์ระหว่างซอร์สโค้ดกับหมายเลขบรรทัดของซอร์สโค้ด ซึ่งอยู่ในรูปแบบไฟล์ซีเอสวี (.csv)
6. ไฟล์ล็อกของฐานข้อมูลที่ใช้รองรับกับ MySQL
7. การเปลี่ยนแปลงของสคีมาฐานข้อมูลจะครอบคลุมเฉพาะการเปลี่ยนแปลงที่เกิดขึ้นภายในฟิลด์ของแต่ละตารางเท่านั้น เช่น การแก้ไขชื่อฟิลด์ การลบฟิลด์ และการเพิ่มฟิลด์ การแก้ไขประเภทข้อมูลของฟิลด์ และการแก้ไขขนาดข้อมูลของฟิลด์ เป็นต้น โดยไม่รวมถึงกรณีการเพิ่มตาราง การแก้ไขตาราง และการลบตาราง
8. ในอนาคตพัฒนาเครื่องมือให้สามารถวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ โดยไม่ต้องใช้ไฟล์แมปปิง

6.3 แนวทางการพัฒนาต่อ

1. พัฒนาเครื่องมือให้สามารถรองรับกับล็อกไฟล์ของฐานข้อมูลอื่นๆ ได้
2. พัฒนาเครื่องมือให้สามารถรองรับกับการเปลี่ยนแปลงของสคีมาฐานข้อมูลในรูปแบบต่างๆ ได้ เช่น การเพิ่ม ลบ ตาราง และการเพิ่มลบเงื่อนไขอื่นๆ เป็นต้น
3. พัฒนาเครื่องมือให้สามารถเชื่อมต่อกับฐานข้อมูลของระบบที่ถูกทดสอบได้
4. พัฒนาเครื่องมือให้รองรับการแก้ไขซอร์สโค้ดที่ได้รับผลกระทบ

รายการอ้างอิง

- [1] P. C. Jorgensen, *Software Testing: A Craftsman's Approach, Second Edition*: Taylor & Francis, 2002.
- [2] "American National Standards Institute, SQL, October 1918. Internet: www.ansi.org," ed.
- [3] "International Organization for Standardization, SQL, December 2011. Internet: www.iso.org," ed.
- [4] S. K. Gardikiotis and N. Malevris, "A two-folded impact analysis of schema changes on database applications," *International Journal of Automation and Computing*, vol. 6, pp. 109-123, 2009.
- [5] A. Karahasanovic, "Identifying impacts of database schema changes on applications," *Proceedings of the 8th Doctoral Consortium at the CAISE*, vol. 1, pp. 93-104, 2001.
- [6] A. Maule, W. Emmerich, and D. S. Rosenblum, "Impact analysis of database schema changes," in *Software Engineering, 2008. ICSE '08. ACM/IEEE 30th International Conference on*, 2008, pp. 451-460.
- [7] P. Tongrak and T. Suwannasart, "A tool for generating test case from relational database constraints testing," in *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, 2009, pp. 435-439.
- [8] T. S. K. Adump, "A Tool for Displaying Effect on C# Program from Database Schema Changes," presented at the *International Computer Science and Engineering Conference*, Hotel Pullman Khon Kaen Raja Orchid.



ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาคผนวก ก
พจนานุกรมข้อมูล

ตารางที่ ก. 1 พจนานุกรมข้อมูลตาราง File_storage

ชื่อ	คำอธิบาย	ชนิดข้อมูล	คีย์	ค่าว่าง
file_storage_id	หมายเลขไฟล์ที่จัดเก็บ	INT(3)	PK	ไม่ ว่าง
schema_file_path	ชื่อโฟลเดอร์ที่เก็บไฟล์ สคีมาฐานข้อมูลก่อน เปลี่ยนแปลง	VARCHAR(100)		ค่า ว่าง
log_file_path	ชื่อโฟลเดอร์ที่เก็บล็อกไฟล์	VARCHAR(100)		ค่า ว่าง
source_code_file_path	ชื่อโฟลเดอร์ที่เก็บซอร์สโค้ด	VARCHAR(100)		ค่า ว่าง
mapping_table_file_path	ชื่อโฟลเดอร์ที่เก็บไฟล์แมปปิง	VARCHAR(100)		ค่า ว่าง
testcase_file_path	ชื่อโฟลเดอร์ที่เก็บไฟล์ กรณีทดสอบ	VARCHAR(100)		ค่า ว่าง
testcase_new_file_path	ชื่อโฟลเดอร์ที่เก็บกรณีทดสอบ ใหม่	VARCHAR(100)		ค่า ว่าง
history_id	หมายเลขประวัติ	INT(3)	FK	ไม่ ว่าง

ตารางที่ ก. 2 พจนานุกรมข้อมูลตาราง History

ชื่อ	คำอธิบาย	ชนิดข้อมูล	คีย์	ค่าว่าง
history_id	หมายเลขประวัติ	INT(3)	PK	ไม่ว่าง
date	วันเดือนปี	DATE		ไม่ว่าง
Time	เวลา	TIME		ไม่ว่าง
project_name	ชื่อโปรเจค	VARCHAR(50)		ไม่ว่าง

ตารางที่ ก.2 พจนานุกรมข้อมูลตาราง History (ต่อ)

ชื่อ	คำอธิบาย	ชนิดข้อมูล	คีย์	ค่าว่าง
project_owner	ชื่อผู้สร้างโปรเจค	VARCHAR(50)		ไม่ว่าง

ตารางที่ ก. 3 พจนานุกรมข้อมูลตาราง Log_after_process

ชื่อ	คำอธิบาย	ชนิดข้อมูล	คีย์	ค่าว่าง
log_file_id	หมายเลข ล็อกไฟล์	INT(3)	PK	ไม่ว่าง
table_name	ชื่อตาราง	VARCHAR(100)		ค่าว่าง
original_field_name	ชื่อฟิลด์ ก่อนเปลี่ยน	VARCHAR(100)		ค่าว่าง
field_name	ชื่อฟิลด์ หลังเปลี่ยน	VARCHAR(100)		ค่าว่าง
field_type	ประเภทของ ฟิลด์	VARCHAR(100)		ค่าว่าง
field_size	ขนาดของฟิลด์	VARCHAR(100)		ค่าว่าง
status_change	ประเภท การ เปลี่ยนแปลง	VARCHAR(200)		ค่าว่าง
history_id	หมายเลข ประวัติ	INT(3)	FK	ไม่ว่าง

ตารางที่ ก. 4 พจนานุกรมข้อมูลตาราง Mapping_table

ชื่อ	คำอธิบาย	ชนิดข้อมูล	คีย์	ค่าว่าง
mapping_table_id	หมายเลขแมป ปิ้ง	INT(3)	PK	ไม่ว่าง
table_name	ชื่อตารางที่ เกิดผลกระทบ	VARCHAR(30)		ค่าว่าง
field_name	ชื่อฟิลด์ที่ เกิดผลกระทบ	VARCHAR(50)		ค่าว่าง
param_name	ชื่อพารามิเตอร์ ที่เกิดผล กระทบ	VARCHAR(50)		ค่าว่าง
line_of_code_number	หมายเลข บรรทัดของ ซอร์สโค้ด ที่เกิดผล กระทบ	INT(3)		ค่าว่าง
history_id	หมายเลข ประวัติ	INT(3)	FK	ไม่ว่าง

ตารางที่ ก. 5 พจนานุกรมข้อมูลตาราง Schema_file

ชื่อ	คำอธิบาย	ชนิดข้อมูล	คีย์	ค่า ว่าง
schema_id	หมายเลข สคีมารฐานข้อมูล	INT(3)	PK	ไม่ ว่าง
table_name	ชื่อตาราง สคีมารฐานข้อมูลก่อนเปลี่ยนแปลง	VARCHAR(100)		ค่า ว่าง
field_name	ชื่อฟิลด์ใน สคีมารฐานข้อมูลก่อนเปลี่ยนแปลง	VARCHAR(100)		ค่า ว่าง

ตารางที่ ก.5 พจนานุกรมข้อมูลตาราง Schema_file (ต่อ)

ชื่อ	คำอธิบาย	ชนิดข้อมูล	คีย์	ค่าว่าง
field_type	ชื่อประเภทของฟิลด์ในสคีมารฐานข้อมูลก่อนเปลี่ยนแปลง	VARCHAR(100)		ค่าว่าง
field_size	ขนาดของฟิลด์ในสคีมารฐานข้อมูลก่อนเปลี่ยนแปลง	VARCHAR(100)		ค่าว่าง
field_position	ตำแหน่งของฟิลด์ในสคีมารฐานข้อมูลก่อนเปลี่ยนแปลง	VARCHAR(10)		ค่าว่าง
history_id	หมายเลขประวัติ	INT(3)	FK	ไม่ว่าง

ตารางที่ ก. 6 พจนานุกรมข้อมูลตาราง Source_code

ชื่อ	คำอธิบาย	ชนิดข้อมูล	คีย์	ค่าว่าง
source_code_id	หมายเลขซอร์สโค้ด	INT(3)	PK	ไม่ว่าง
table_name	ชื่อตารางที่เกิดผลกระทบในซอร์สโค้ด	VARCHAR(100)		ค่าว่าง
field_name	ชื่อฟิลด์ที่เกิดผลกระทบในซอร์สโค้ด	VARCHAR(200)		ค่าว่าง
param_name	พารามิเตอร์ที่เกิดผลกระทบในซอร์สโค้ด	VARCHAR(100)		ค่าว่าง
history_id	หมายเลขประวัติ	INT(3)	FK	ไม่ว่าง

ตารางที่ ก. 7 พจนานุกรมข้อมูลตาราง Testcase

ชื่อ	คำอธิบาย	ชนิดข้อมูล	คีย์	ค่าว่าง
tc_id	เลขหัวชุดหมายเลขกรณีทดสอบ	INT(3)	PK	ไม่ว่าง
tc_number	หมายเลขกรณีทดสอบ	VARCHAR(50)		ค่าว่าง
tc_path	เส้นทางการไหลของข้อมูลในซอร์สโค้ด	VARCHAR(200)		ค่าว่าง
tc_input_name	ชื่อข้อมูลนำเข้าของกรณีทดสอบ	VARCHAR(50)		ค่าว่าง
tc_input_type	ประเภทข้อมูลนำเข้าของกรณีทดสอบ	VARCHAR(50)		ค่าว่าง
tc_input_size	ขนาดข้อมูลนำเข้าของกรณีทดสอบ	VARCHAR(50)		ค่าว่าง
tc_input_value	ข้อมูลทดสอบของกรณีทดสอบ	VARCHAR(500)		ค่าว่าง
tc_expected_status	เคสสำเร็จหรือล้มเหลว	VARCHAR(50)		ค่าว่าง
tc_expected_output	ผลลัพธ์ที่คาดหวัง	VARCHAR(100)		ค่าว่าง
is_fixed	ฟิลด์ถูกแก้ไขหรือไม่	VARCHAR(1)		ค่าว่าง
history_id	หมายเลขประวัติ	INT(3)	FK	ไม่ว่าง

ตารางที่ ก. 7 พจนานุกรมข้อมูลตาราง Testcase (ต่อ)

ชื่อ	คำอธิบาย	ชนิดข้อมูล	คีย์	ค่าว่าง
tc_input_name_new	ชื่อข้อมูลนำเข้าของกรณีทดสอบที่ถูกปรับปรุงแก้ไขเสร็จสิ้น	VARCHAR(100)		ค่าว่าง
tc_input_type_new	ประเภทข้อมูลนำเข้าของกรณีทดสอบที่ถูกปรับปรุงแก้ไขเสร็จสิ้น	VARCHAR(100)		ค่าว่าง
tc_input_size_new	ขนาดข้อมูลนำเข้าของกรณีทดสอบที่ถูกปรับปรุงแก้ไขเสร็จสิ้น	VARCHAR(100)		ค่าว่าง
tc_input_value_new	ข้อมูลทดสอบของกรณีทดสอบที่ถูกปรับปรุงแก้ไขเสร็จสิ้น	VARCHAR(500)		ค่าว่าง

ภาคผนวก ข
ตัวอย่างระบบทดสอบ

1) ระบบการลงทุน

การทดสอบด้วยระบบการลงทุน ประกอบด้วย ไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลง 1 ไฟล์ ไฟล์ล็อก 1 ไฟล์ ไฟล์ซอร์สโค้ด newMember 1 ไฟล์ (เพิ่มสมาชิกใหม่) ไฟล์แมปปิง 1 ไฟล์ และไฟล์กรณีทดสอบ 1 ไฟล์ ประกอบไปด้วย 10 กรณีทดสอบ ดังนี้

1.1) วิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล

ขั้นตอนแรกเครื่องมือจะวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล สามารถแสดงรายละเอียดของไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลงได้ ดังตารางที่ ข-1 และไฟล์ล็อกดังตารางที่ ข-2

ตารางที่ ข-1 ตัวอย่างไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลง

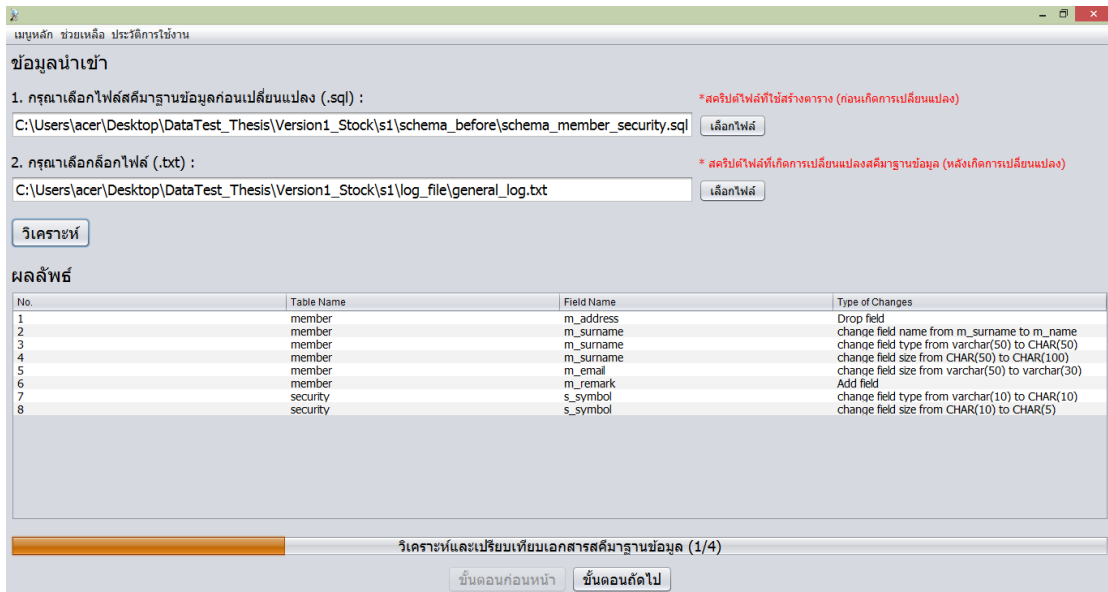
ชื่อตาราง	ชื่อฟิลด์	ประเภทของฟิลด์	ขนาดของฟิลด์
member	m_id	INT	3
	m_surname	VARCHAR	50
	m_lastname	VARCHAR	50
	m_password	VARCHAR	50
	m_email	VARCHAR	50
	m_address	VARCHAR	50
	m_cash	VARCHAR	100
security	s_cname	VARCHAR	50
	s_current_price	VARCHAR	100
	s_offer_price	VARCHAR	100
	s_bid_price	VARCHAR	100
	s_symbol	VARCHAR	10

ตารางที่ ข-2 ตัวอย่างไฟล์ล็อก

คำสั่ง เอสคิว แอล	ชื่อตาราง	ประเภทการ เปลี่ยนแปลง	ชื่อฟิลด์ ก่อนเปลี่ยน	ชื่อฟิลด์ หลัง เปลี่ยน	ประเภท ของฟิลด์	ขนาด ของ ฟิลด์
ALTER TABLE	member	DROP	m_address	-	-	-
ALTER TABLE	member	CHANGE	m_surname	m_name	CHAR	100
ALTER TABLE	member	CHANGE	m_email	m_email	VARCHAR	30
ALTER TABLE	member	ADD	m_remark	-	VARCHAR	50
ALTER TABLE	security	CHANGE	s_symbol	s_symbol	CHAR	5

จากตารางที่ ข-1 และ ตารางที่ ข-2 เครื่องมือจะวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูลดังกล่าว เพื่อวิเคราะห์หาการเปลี่ยนแปลงที่เกิดขึ้น และเครื่องมือจะแสดงรายละเอียดการเปลี่ยนแปลงที่เกิดขึ้น ดังภาพที่ ข-1 สามารถสรุปการเปลี่ยนแปลงได้ดังนี้

- ตาราง Member
 - การลบฟิลด์ m_address
 - การแก้ไขชื่อฟิลด์ m_surname เป็น m_name
 - การแก้ไขประเภทของฟิลด์ m_surname VARCHAR(50) เป็น CHAR(50)
 - การแก้ไขขนาดของฟิลด์ m_surname CHAR(50) CHAR(100)
 - การแก้ไขขนาดของฟิลด์ m_email VARCHAR(50) เป็น VARCHAR(30)
 - การเพิ่มฟิลด์ m_remark
- ตาราง Security
 - การแก้ไขประเภทของฟิลด์ s_symbol
 - การแก้ไขขนาดของฟิลด์ s_symbol



ภาพที่ ข-1 เครื่องมือแสดงรายละเอียดการเปลี่ยนแปลง

1.2) วิเคราะห์หาข้อผิดพลาดที่ได้รับผลกระทบ

ขั้นตอนที่สองเครื่องมือจะวิเคราะห์หาข้อผิดพลาดที่ได้รับผลกระทบ จากภาพที่ ข-2 ตัวอย่างของซอร์สโค้ด newMember และจากภาพที่ข-3 ตัวอย่างไฟล์แมปปิงของ newMember

```

1  class newMember
2  {
3      private static void newMember()
4      {
5          char m_surname [] = new char[10];
6          char m_lastname [] = new char[10];
7          char m_password [] = new char[10];
8          char m_email [] = new char[15];
9          char m_address [] = new char[20];
10         char m_cash [] = new char[5];
11         m_surname = readEntry("Enter first name: ");
12         m_lastname = readEntry("Enter last name: ");
13         m_email = readEntry("Enter email: ");
14         m_address = readEntry("Enter address: ");
15         m_password = readEntry("Enter password: ");
16         m_cash = readEntry("Enter initial cash: ");
17         try
18         {
19             char m_id [] = new char[3];
20             #sql (select m_seq.nextval into :m_id from dual);
21             #sql (insert into member ( m_id , m_surname , m_lastname , m_password , m_email , m_address , m_cash ) values ( :m_
22             #sql (commit);
23             System.out.println("Add New Account Successfully");
24             System.out.println("\nYour Account Number is " + m_id);
25         }
26         catch (SQLException e)
27         {
28             System.out.println("Failed !! could not add new account ");
29             System.out.println("Message:" + e.getMessage());
30         }
31         return;
32     }
33 }

```

ภาพที่ ข-2 ตัวอย่างของซอร์สโค้ด newMember

```

1 line of code number,Statement
2 1, class newMember
3 2, {
4 3, private static void newMember()
5 4, {
6 5, char m_surname [] = new char[10];
7 6, char m_lastname [] = new char[10];
8 7, char m_password [] = new char[10];
9 8, char m_email [] = new char[15];
10 9, char m_address [] = new char[20];
11 10, char m_cash [] = new char[5];
12 11, m_surname = readEntry("Enter first name: ");
13 12, m_lastname = readEntry("Enter last name: ");
14 13, m_email = readEntry("Enter email: ");
15 14, m_address = readEntry("Enter address: ");
16 15, m_password = readEntry("Enter password: ");
17 16, m_cash = readEntry("Enter initial cash: ");
18 17, try
19 18, {
20 19, char m_id [] = new char[3];
21 20, #sql {select m_seq.nextval into :m_id from dual};
22 21, #sql {insert into member ( m_id , m_surname , m_lastname , m_password , m_email , m_address , m_cash ) values ( :m_id , :m_surname , :m_lastname , :m_password , :m_email , :m_address , :m_cash )};
23 22, #sql {commit};
24 23, System.out.println("Add New Account Successfully");
25 24, System.out.println("\nYour Account Number is " + m_id);
26 25, }
27 26, catch (SQLException e)
28 27, {
29 28, System.out.println("Failed !! could not add new account ");
30 29, System.out.println("Message:"+e.getMessage ());
31 30, return;
32 31, }
33 32, }
34 33, }

```

ภาพที่ ข-3 ตัวอย่างของไฟล์แมปปิ้ง newMember

เครื่องมือจะวิเคราะห์ไฟล์ดังกล่าวที่ ข-2 และภาพที่ ข-3 เพื่อหาผลกระทบต่อซอร์สโค้ด และแสดงหมายเลขบรรทัดของซอร์สโค้ดที่เกิดผลกระทบ โดยเครื่องมือจะแสดงตำแหน่งที่เกิดผลกระทบต่อซอร์สโค้ด และหมายเลขบรรทัดของซอร์สโค้ดที่เกิดผลกระทบ แสดงได้ดังภาพที่ ข-4

The screenshot shows a software analysis tool interface with the following components:

- เมนูหลัก** (Main Menu): ประกอบด้วยปุ่ม **ช่วยเหลือ** (Help) และ **ประวัติการใช้งาน** (Usage History).
- ข้อมูลนำเข้า** (Input Information):
 - 1. กรณำเลือกไฟล์ซอร์สโค้ดแบบฝังตัว (.java): `n1_Stock\s2\source code\newMember.java` (เลือกไฟล์)
 - 2. กรณำเลือกไฟล์แมปปิ้ง (.csv): `n1_Stocks2\mapping table\mapping_table_newMember.csv` (เลือกไฟล์)
- วิเคราะห์** (Analyze): ปุ่มกดเพื่อเริ่มการวิเคราะห์.
- ตำแหน่งที่เกิดผลกระทบ** (Affected Locations): แสดงโค้ดที่ถูกระบุด้วยไฮไลต์สีเหลืองและสีเขียว:


```

6 char m_surname [] = new char[10];
7 char m_lastname [] = new char[10];
8 char m_password [] = new char[10];
9 char m_email [] = new char[15];
10 char m_address [] = new char[20];
11 char m_cash [] = new char[5];
12 m_surname = readEntry("Enter first name: ");
13 m_lastname = readEntry("Enter last name: ");
14 m_email = readEntry("Enter email: ");
15 m_address = readEntry("Enter address: ");
16 m_password = readEntry("Enter password: ");
17 m_cash = readEntry("Enter initial cash: ");
18 try
19 {
20 char m_id [] = new char[3];
21 #sql {select m_seq.nextval into :m_id from dual};
22 #sql {insert into member ( m_id , m_surname , m_lastname , m_password , m_email , m_address , m_cash ) values ( :m_id , :m_surname , :m_lastname , :m_password , :m_email , :m_address , :m_cash )};
23 #sql {commit};

```
- หมายเลขบรรทัดที่เกิดผลกระทบ** (Affected Line Numbers): รายการตัวเลข 5, 8, 9, 11, 13, 14, 21.
- สีเหลือง**: เพิ่มฟิลด์ (Yellow: Add Field)
- สีเขียว**: ลบฟิลด์ (Green: Remove Field)
- สีฟ้า**: ชื่อตาราง (Blue: Table Name)
- วิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ (2/4)**: แถบแสดงสถานะการวิเคราะห์.
- ขั้นตอนก่อนหน้า** (Previous Step) และ **ขั้นตอนถัดไป** (Next Step): ปุ่มนำทาง.

ภาพที่ ข-4 เครื่องมือแสดงตำแหน่งและหมายเลขบรรทัดของซอร์สโค้ดที่ได้รับผลกระทบ

1.3) วิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ

ขั้นตอนที่สามเครื่องมือจะวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ จากภาพที่ ข-5 ตัวอย่างของไฟล์กรณีทดสอบ ซึ่งประกอบด้วย 10 กรณีทดสอบ

TCNumber	TCPath	TCInputName	TCInputType	TCInputSize	TCInputValue	TCExpectedStatus	TCExpectedOutput
TC-003	4-5-6-7-8-9-10-11-12-13-14	m_id	char	5	Jacky	1	Login success
		m_password	char	10	aaaaaaaaaa	1	Login success
TC-004	8-9-10-11-12-13-14-15-16-1	m_id	char	5	somchai	0	Invalid Account
		m_password	char	10	aaa	0	Invalid Account
TC-001	-23-24-25-26-27-28-29-30-3	m_id	char	2	aa	1	Updated successfully
TC-002	-10-11-12-13-14-15-16-17-1	m_id	char	2	Chanwit	0	Invalid account
TC-005	9-10-11-12-13-14-20-21-24	s_symbol	char	3	Joe	1	showPriceQuote
TC-006	8-9-10-11-12-13-14-15-16-1	s_symbol	char	3	joe123	0	Invalid
TC-007	7-8-9-10-11-13-14-15-16-17	m_id	char	2	aaaa	0	Delete Failed
		m_surname	char	5	abc	0	Delete Failed
TC-008	3-4-5-6-7-8-9-10-11-12-13-2	m_id	char	2	a1	1	Delete Successfully
		m_surname	char	5	owen1	1	Delete Successfully
TC-009	-12-13-14-15-16-17-18-19-2	m_surname	char	10	somjai1234	1	Add New Account Successfully
		m_lastname	char	10	chokdee123	1	Add New Account Successfully
		m_password	char	10	aaaaaaaaaa	1	Add New Account Successfully
		m_email	char	15	somjai1@hotmail.com	1	Add New Account Successfully
		m_address	char	20	11111/1111 Lucky Rd.	1	Add New Account Successfully
		m_cash	char	5	50000	1	Add New Account Successfully
TC-010	3-14-15-16-17-18-19-20-21	m_surname	char	10	aaa	0	Failed !! could not add new account
		m_lastname	char	10	bbb	0	Failed !! could not add new account
		m_password	char	10	ccc	0	Failed !! could not add new account
		m_email	char	15	sss@hotmail.com	0	Failed !! could not add new account
		m_address	char	20	1/111 Lucky Rd.	0	Failed !! could not add new account
		m_cash	char	5	500	0	Failed !! could not add new account

ภาพที่ ข-5 ตัวอย่างกรณีทดสอบ

จากภาพที่ ข-5 เครื่องมือวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบคือ หมายเลขกรณีทดสอบ TC-009 และ TC-010 ดังภาพที่ ข-6

เมนูหลัก ช่วยเหลือ ประวัติการใช้งาน

ข้อมูลนำเข้า

1. กรุณาเลือกกรณีทดสอบ (.xlsx) :

t_case_version1_field_name_InputTcname.xlsx

กรณีทดสอบที่ได้รับผลกระทบ

- TC-007
- TC-008
- TC-009
- TC-010

รายละเอียดกรณีทดสอบ

TCNumber: TC-009
 TCPath: 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-2

TCName : m_surname
 TCInputType : char
 TCInputSize : 10
 TCInputValue : somjai1234
 TCExpectedOutput : Add New Account Successfully

TCName : m_email
 TCInputType : char
 TCInputSize : 15
 TCInputValue : somjai1@hotmail.com
 TCExpectedOutput : Add New Account Successfully

TCName : m_address
 TCInputType : char
 TCInputSize : 20
 TCInputValue : 11111/1111 Lucky Rd.

วิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ (3/4)

ภาพที่ ข-6 เครื่องมือแสดงรายละเอียดกรณีทดสอบที่ได้รับผลกระทบ

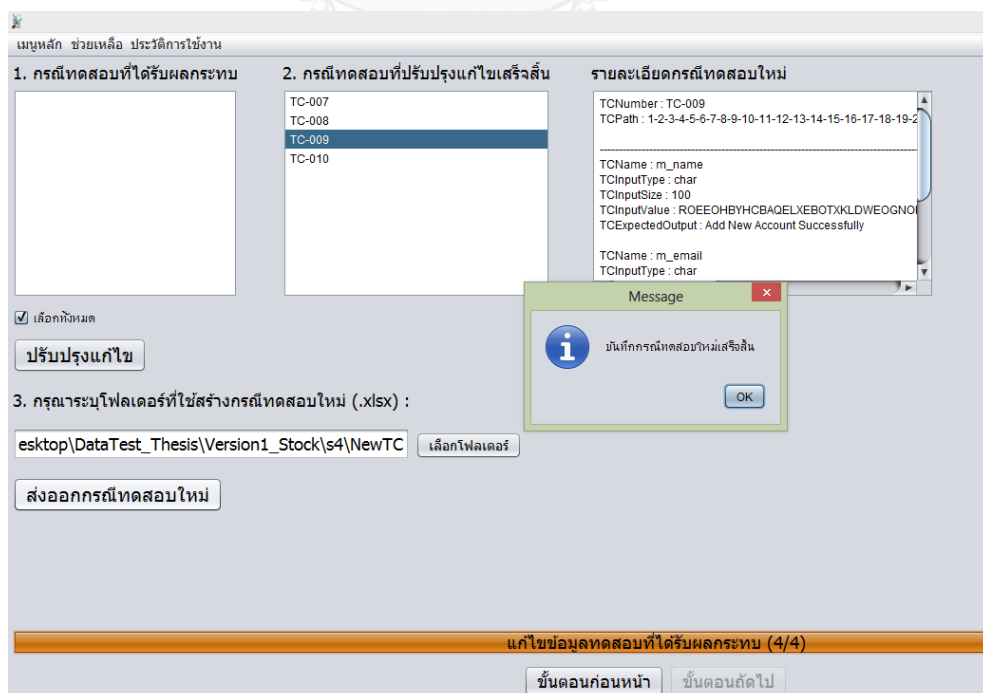
1.4) แก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ

ขั้นตอนที่สี่เครื่องมือจะแก้ไขข้อมูลทดสอบของกรณีทดสอบที่ได้รับผลกระทบ ดังภาพที่ ข-7

โดยจะปรับปรุงแก้ไขข้อมูลทดสอบของฟิลด์ที่เกิดการเปลี่ยนแปลง ดังนี้

- ฟิลด์ m_surname ปรับปรุงแก้ไขประเภท และขนาดของฟิลด์จาก VARCHAR(50) เป็น CHAR(100)
- ฟิลด์ m_email ปรับปรุงแก้ไขขนาดของฟิลด์จาก VARCHAR(50) เป็น VARCHAR(30)
- ฟิลด์ m_address ปรับปรุงแก้ไขด้วยการลบข้อมูลทดสอบออกจากกรณีทดสอบ เพราะฟิลด์ m_address จากสคีมาฐานข้อมูล
- เพิ่มฟิลด์ m_remark จึงต้องมีการสร้างข้อมูลทดสอบให้กับฟิลด์ใหม่ด้วย
- ฟิลด์ s_symbol ปรับปรุงแก้ไขประเภท และขนาดของฟิลด์จาก VARCHAR(10) เป็น CHAR(5)

จากภาพที่ ข-8 เป็นตัวอย่างกรณีทดสอบที่ถูกสร้างขึ้นใหม่ โดยมีการปรับปรุงแก้ไขข้อมูลทดสอบ แล้วจากที่กล่าวมา โดยการระบายสีเหลือง เพื่อแสดงให้เห็นว่ากรณีทดสอบใดที่ได้รับการปรับปรุงแก้ไข



ภาพที่ ข-7 เครื่องมือแก้ไขข้อมูลทดสอบและส่งออกเสร็จสิ้น

TCNumber	TCPath	TCName	TCType	TCSize	TCValue	TCExpected	TCExpectedOutput
TC-003	1-2-3-4-5-	m_id	char	5	Jacky	1	Login success
		m_passwo	char	10	aaaaaaaaa	1	Login success
TC-004	1-2-3-4-5-	m_id	char	5	somchai	0	Invalid Account
		m_passwo	char	10	aaa	0	Invalid Account
TC-001	1-2-3-4-5-	m_id	char	2	aa	1	Updated successfully
TC-002	1-2-3-4-5-	m_id	char	2	Chanwit	0	Invalid account
TC-005	1-2-3-4-5-	s_symbol	char	3	Joe	1	showPriceQuote
TC-006	1-2-3-4-5-	s_symbol	char	3	joe123	0	Invalid
TC-007	1-2-3-4-5-	m_id	char	2	aaaa	0	Delete Failed
		m_name	char	100	ASATEJCG	0	Delete Failed
TC-008	1-2-3-4-5-	m_id	char	2	a1	1	Delete Successfully
		m_name	char	100	KTXBWMC	1	Delete Successfully
TC-009	1-2-3-4-5-	m_name	char	100	ROEEOHB	1	Add New Account Successfully
		m_lastname	char	10	chokdee12	1	Add New Account Successfully
		m_passwo	char	10	aaaaaaaaa	1	Add New Account Successfully
		m_email	char	30	NJOFKPW	1	Add New Account Successfully
		m_cash	char	5	50000.0	1	Add New Account Successfully
TC-010	1-2-3-4-5-	m_name	char	100	GQROAES	0	Failed !! could not add new account
		m_lastname	char	10	bbb	0	Failed !! could not add new account
		m_passwo	char	10	ccc	0	Failed !! could not add new account
		m_email	char	30	SGJZNMFV	0	Failed !! could not add new account
		m_cash	char	5	500.0	0	Failed !! could not add new account
TC-NEW-1		m_remark	char	50	AOASWYZ	1	

ภาพที่ ข-8 ตัวอย่างกรณีทดสอบที่แก้ไขข้อมูลทดสอบ

2) ระบบจองตั๋วเครื่องบินโดยสารออนไลน์

การทดสอบด้วยระบบจองตั๋วเครื่องบินโดยสารออนไลน์ ประกอบด้วย ไฟล์สคีมาฐานข้อมูล ก่อนเปลี่ยนแปลง 1 ไฟล์ ไฟล์ล็อก 1 ไฟล์ ไฟล์ซอร์สโค้ด delPassenger 1 ไฟล์ (ลบสมาชิก) ไฟล์แมปปิง 1 ไฟล์ และไฟล์กรณีทดสอบ 1 ไฟล์ ประกอบไปด้วย 8 กรณีทดสอบ ดังนี้

2.1) วิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล

ขั้นตอนแรกเครื่องมือจะวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล สามารถแสดงรายละเอียดของไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลงได้ ดังตารางที่ ข-3 และไฟล์ล็อก ดังตารางที่ ข-4

ตารางที่ ข-3 ตัวอย่างไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลง

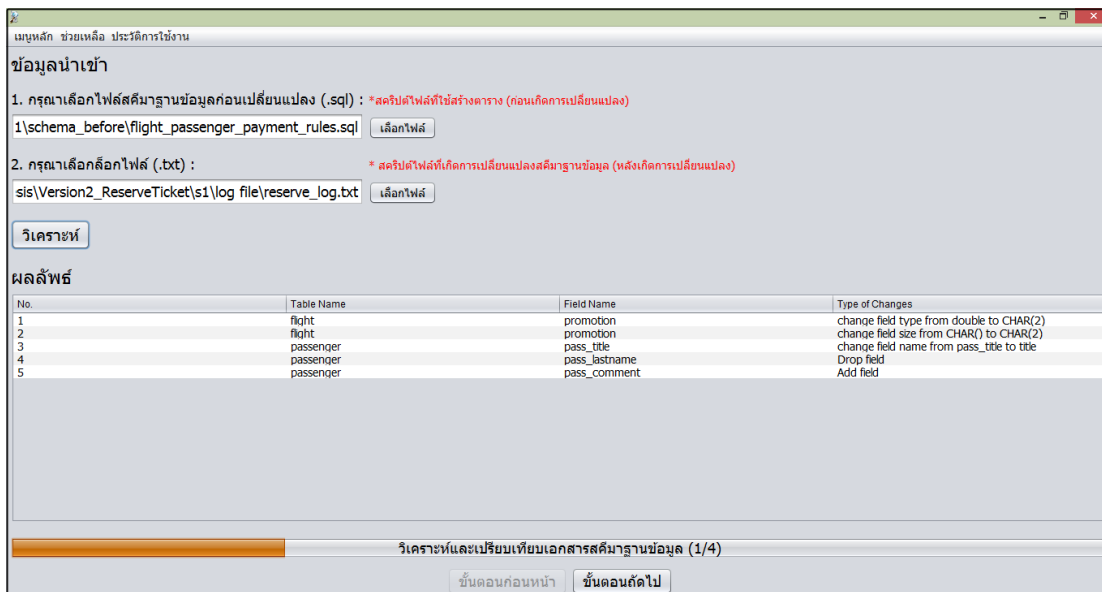
ชื่อตาราง	ชื่อฟิลด์	ประเภทของฟิลด์	ขนาดของฟิลด์
flight	flight_id	INT	3
	flight_operator	VARCHAR	50
	departs	VARCHAR	100
	arrives	VARCHAR	100
	promotion	DOUBLE	-
	economy	DOUBLE	-
passenger	passenger_id	INT	3
	pass_title	VARCHAR	20
	pass_firstname	VARCHAR	50
	pass_lastname	VARCHAR	50
	pass_address	VARCHAR	100
	pass_email	VARCHAR	50
	pass_tel	VARCHAR	10
	pass_dob	DATE	-
	nationality	VARCHAR	50
payment_method	payment_method_id	INT	3
	directDebit	VARCHAR	50
	counterPayment	VARCHAR	50
	atmPayment	VARCHAR	50
rule_regulations	rule_regulations_id	INT	3
	rules	TEXT	-
	isCheck	INT	1

ตารางที่ ข-4 ตัวอย่างไฟล์ล็อก

คำสั่ง เอสคิว แอล	ชื่อตาราง	ประเภทการ เปลี่ยนแปลง	ชื่อฟิลด์ ก่อนเปลี่ยน	ชื่อฟิลด์ หลังเปลี่ยน	ประเภท ของฟิลด์	ขนาด ของ ฟิลด์
ALTER TABLE	flight	CHANGE	promotion	promotion	CHAR	2
ALTER TABLE	passenger	CHANGE	pass_title	title	VARCHAR	20
ALTER TABLE	passenger	DROP	pass_lastname	-	-	-
ALTER TABLE	passenger	ADD	pass_comment	-	VARCHAR	100

จากตารางที่ ข-3 และ ตารางที่ ข-4 เครื่องมือจะวิเคราะห์และเปรียบเทียบเอกสาร
สคีมาฐานข้อมูลดังกล่าว เพื่อวิเคราะห์หาการเปลี่ยนแปลงที่เกิดขึ้น และเครื่องมือจะแสดง
รายละเอียดการเปลี่ยนแปลงที่เกิดขึ้น ดังภาพที่ ข-9 สามารถสรุปการเปลี่ยนแปลงได้ดังนี้

- ตาราง Flight
 - ฟิลด์ promotion แก้ไขประเภท และขนาดจาก DOUBLE เป็น CHAR(2)
- ตาราง Passenger
 - ฟิลด์ pass_title แก้ไขชื่อ จาก pass_title เป็น title
 - ฟิลด์ pass_lastname ถูกลบ
 - เพิ่มฟิลด์ใหม่ pass_comment



ภาพที่ ข-9 เครื่องมือแสดงรายละเอียดการเปลี่ยนแปลง

1.2) วิเคราะห์หาข้อผิดพลาดที่ได้รับผลกระทบ

ขั้นตอนที่สองเครื่องมือจะวิเคราะห์หาข้อผิดพลาดที่ได้รับผลกระทบ จากภาพที่ ข-10 ตัวอย่างของข้อผิดพลาด delPassenger และจากภาพที่ข-11 ตัวอย่างไฟล์แม่พิมพ์ของ delPassenger

```

1 class delPassenger
2 {
3     public static void delPassenger()
4     {
5         char payment_method_id1 [] = new char[2];
6         payment_method_id1 = readEntry("Enter your payment method id : ");
7         try
8         {
9             #sql { delete from payment_method where payment_method_id = :payment_method_id1 };
10            System.out.println("Delete Successfully");
11        }
12        catch (Exception e)
13        {
14            System.out.println("Delete Failed !!");
15            System.out.println("Message:"+e.getMessage());
16            return;
17        }
18    }
19 }

```

ภาพที่ ข-10 ตัวอย่างข้อผิดพลาด delPassenger

```

1 1, class delPassenger
2 2, {
3 3, public static void delPassenger()
4 4, {
5 5, char payment_method_id1 [] = new char[2];
6 6, payment_method_id1 = readEntry("Enter your payment method id : ");
7 7, try
8 8, {
9 9, #sql { delete from payment_method where payment_method_id = :payment_method_id1 };
10 10, System.out.println("Delete Successfully");
11 11, }
12 12, catch (Exception e)
13 13, {
14 14, System.out.println("Delete Failed !!");
15 15, System.out.println("Message:"+e.getMessage());
16 16, return;
17 17, }
18 18, }
19 19, }

```

ภาพที่ ข-11 ตัวอย่างไฟล์แมปปิ้ง delPassenger

เครื่องมือจะวิเคราะห์ไฟล์ดังกล่าวที่ ข-10 และภาพที่ ข-11 เพื่อหาผลกระทบต่อซอร์สโค้ด และแสดงหมายเลขบรรทัดของซอร์สโค้ดที่เกิดผลกระทบ แต่ในกรณีนี้ขอยกตัวอย่างไฟล์ซอร์สโค้ดที่ไม่ได้รับผลกระทบจากการเปลี่ยนแปลงคือ delPassenger เพราะว่าฟิลด์ payment_method_id จากตาราง payment_method ไม่เกิดการเปลี่ยนแปลงจากขั้นตอนการวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล จึงทำให้ซอร์สโค้ดนี้ไม่ได้รับผลกระทบ แสดงได้ดังภาพที่ ข-12

The screenshot shows a software analysis tool window with the following elements:

- File Selection:** Two steps are shown. Step 1: 'กรุณาเลือกไฟล์ซอร์สโค้ดแบบฝังตัว (.java) :' with a file selected as 'veTicket\s2\source_code\delPassenger.java'. Step 2: 'กรุณาเลือกไฟล์แมปปิ้ง (.csv) :'
- Code Editor:** Displays the Java code for the `delPassenger` method, including the SQL delete statement and exception handling.
- Message Dialog:** A dialog box titled 'Message' with an information icon and the text 'ไม่ได้รับผลกระทบ' (Not affected), with an 'OK' button.
- Footer:** A status bar at the bottom indicates 'วิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ (2/4)' and navigation buttons for 'ขั้นตอนก่อนหน้า' and 'ขั้นตอนถัดไป'.

ภาพที่ ข-12 เครื่องมือวิเคราะห์หาผลกระทบต่อซอร์สโค้ด

จากภาพที่ ข-12 เครื่องมือวิเคราะห์หาผลกระทบต่อซอร์สโค้ด ในกรณีนี้คือไฟล์ซอร์สโค้ด ไม่ได้รับผลกระทบ เพราะไม่มีการเปลี่ยนแปลงที่เกี่ยวข้องกัน ทั้งนี้เพื่ออยากแสดงให้เห็นว่าการวิเคราะห์ผลกระทบต่อซอร์สโค้ดนั้น สามารถวิเคราะห์ได้หลากหลายรูปแบบ ขึ้นอยู่กับว่าซอร์สโค้ดนั้นมีการทำงานที่เกี่ยวข้องกับฟิลต์ที่เกิดการเปลี่ยนแปลงหรือไม่ หากไม่เกี่ยวข้องกัน ก็ย่อมไม่ส่งผลกระทบต่อซอร์สโค้ดแต่อย่างใด ดังนั้นกรณีทดสอบก็ไม่ได้รับผลกระทบเช่นกัน

3) ระบบสั่งซื้อสินค้าออนไลน์

การทดสอบด้วยระบบสั่งซื้อสินค้าออนไลน์ ประกอบด้วย ไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลง 1 ไฟล์ ไฟล์ล็อก 1 ไฟล์ ไฟล์ซอร์สโค้ด checkPaid 1 ไฟล์ (เช็คการชำระเงิน) ไฟล์แมปปิง 1 ไฟล์ และไฟล์กรณีสอบ 1 ไฟล์ ประกอบไปด้วย 4 กรณีสอบ ดังนี้

1.1) วิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล

ขั้นตอนแรกเครื่องมือจะวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล สามารถแสดงรายละเอียดของไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลงได้ ดังตารางที่ ข-5 และไฟล์ล็อกดังตารางที่ ข-6

ตารางที่ ข-5 ตัวอย่างไฟล์สคีมาฐานข้อมูลก่อนเปลี่ยนแปลง

ชื่อตาราง	ชื่อฟิลต์	ประเภทของฟิลต์	ขนาดของฟิลต์
customer	customer_id	INT	3
	customer_name	VARCHAR	50
	customer_lastname	VARCHAR	50
	customer_address	VARCHAR	100
	customer_tel	VARCHAR	10
	customer_paid	VARCHAR	5
product	product_id	INT	3
	product_name	VARCHAR	50
	product_price	DOUBLE	-
	product_detail	VARCHAR	50

ตารางที่ ข-6 ตัวอย่างไฟล์ล็อก

คำสั่ง เอสคิวแอล	ชื่อตาราง	ประเภท การ เปลี่ยนแปลง	ชื่อฟิลด์ ก่อน เปลี่ยน	ชื่อฟิลด์ หลังเปลี่ยน	ประเภท ของฟิลด์	ขนาดของ ฟิลด์
ALTER TABLE	customer	DROP	customer _paid	-	-	-
ALTER TABLE	product	CHANGE	product_ detail	product_ detail	CHAR	5
ALTER TABLE	customer	CHANGE	customer _address	customer _address	TEXT	-
ALTER TABLE	customer	CHANGE	customer _name	customer _name	DOUBLE	-

จากตารางที่ ข-5 และ ตารางที่ ข-6 เครื่องมือจะวิเคราะห์และเปรียบเทียบเอกสาร
สคีมาฐานข้อมูลดังกล่าว เพื่อวิเคราะห์หาการเปลี่ยนแปลงที่เกิดขึ้น และเครื่องมือจะแสดง
รายละเอียดการเปลี่ยนแปลงที่เกิดขึ้น ดังภาพที่ ข-13 สามารถสรุปการเปลี่ยนแปลงได้ดังนี้

- ตาราง Customer
 - ฟิลด์ customer_paid ถูกลบ
 - ฟิลด์ customer_address แก้ไขประเภทของฟิลด์ จาก VARCHAR(100) เป็น TEXT

- ฟิลด์ customer_name แก้ไขประเภทของฟิลด์ จาก VARCHAR(50) เป็น DOUBLE
- ตาราง Product
 - ฟิลด์ product_detail แก้ไขประเภทของฟิลด์ จาก VARCHAR(50) เป็น CHAR(50)
 - ฟิลด์ product_detail แก้ไขขนาดของฟิลด์ จาก CHAR(50) เป็น CHAR(5)

เมนูหลัก ช่วยเหลือ ประสิทธิภาพใช้งาน

ข้อมูลนำเข้า

1. กรุณาเลือกไฟล์สคริปต์ฐานข้อมูลก่อนเปลี่ยนแปลง (.sql) : * สคริปต์ไฟล์ที่ใช้สร้างตาราง (ก่อนเกิดการเปลี่ยนแปลง)

C:\Users\acer\Desktop\DataTest_Thesis\Version3_buyOnline\s1\schema_before\schema_before_buyOnline.sql

2. กรุณาเลือกล็อกไฟล์ (.txt) : * สคริปต์ไฟล์ที่เกิดการเปลี่ยนแปลงสคริปต์ฐานข้อมูล (หลังเกิดการเปลี่ยนแปลง)

C:\Users\acer\Desktop\DataTest_Thesis\Version3_buyOnline\s1\log file\buyOnline_log.txt

ผลลัพธ์

No.	Table Name	Field Name	Type of Changes
1	customer	customer_paid	Drop field
2	customer	customer_address	change field type from varchar(100) to TEXT
3	customer	customer_name	change field type from varchar(50) to DOUBLE
4	product	product_detail	change field type from varchar(50) to CHAR(50)
5	product	product_detail	change field size from CHAR(50) to CHAR(5)

วิเคราะห์และเปรียบเทียบเอกสารสคริปต์ฐานข้อมูล (1/4)

ภาพที่ ข-13 เครื่องมือแสดงรายละเอียดการเปลี่ยนแปลง

1.2) วิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ

ขั้นตอนที่สองเครื่องมือจะวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ จากภาพที่ ข-14 ตัวอย่างของซอร์สโค้ด checkPaid และจากภาพที่ ข-15 ตัวอย่างไฟล์แม่ปั๊งของ checkPaid

```

1 class checkPaid
2 {
3     public static void checkPaid()
4     {
5         char cp2 [] = new char[3];
6         cp2 = readEntry("Check Paid : ").toUpperCase();
7         char cn2 [] = new char[4];
8         cn2 = readEntry("Enter Customer name : ").toUpperCase();
9         try
10        {
11            #sql { select customer_name , customer_lastname , customer_address , customer_tel into :cn , :cl , :ca , :ct
12        }
13        catch (Exception e)
14        {
15            System.out.println("Not found customer !!");
16            return;
17        }
18        System.out.println("Customer name has Paid : " + cn);
19        System.out.println("Customer lastname has Paid : " + cl);
20        System.out.println("Customer address has Paid : " + ca);
21        System.out.println("Customer telephone number has Paid : " + ct);
22    }
23 }

```

ภาพที่ ข-14 ตัวอย่างซอร์สโค้ด checkPaid

```

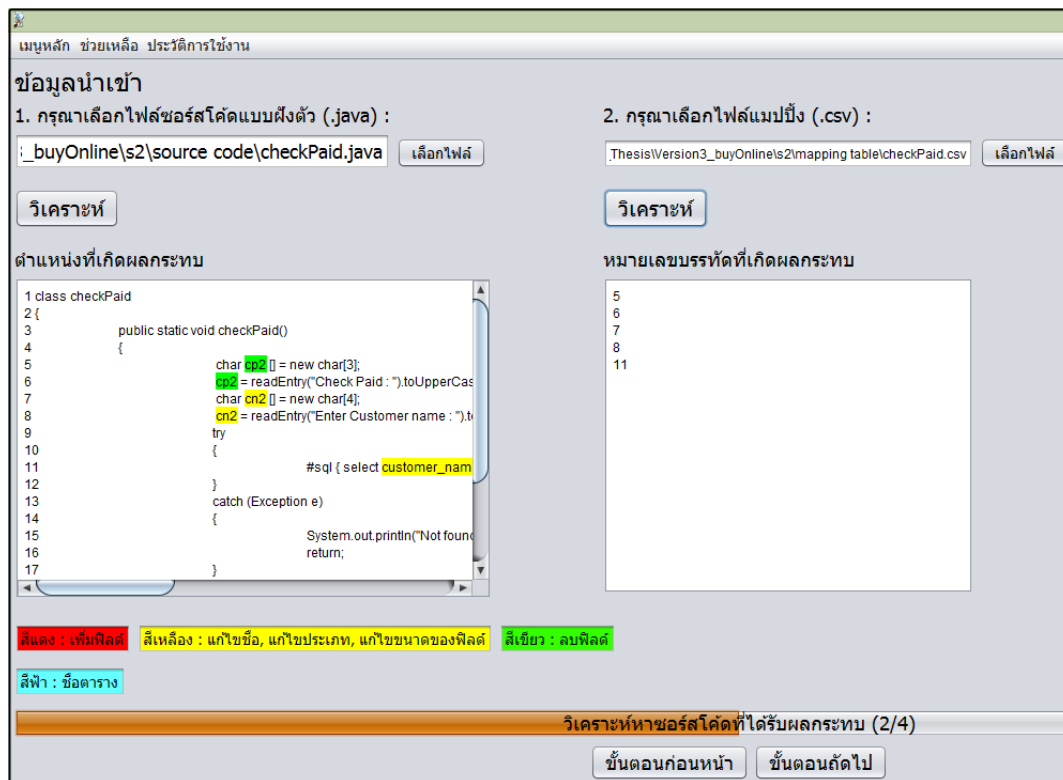
1 1, class checkPaid
2 2, {
3 3, public static void checkPaid()
4 4, {
5 5, char cp2 [] = new char[3];
6 6, cp2 = readEntry("Check Paid : ").toUpperCase();
7 7, char cn2 [] = new char[4];
8 8, cn2 = readEntry("Enter Customer name : ").toUpperCase();
9 9, try
10 10, {
11 11, #sql { select customer_name , customer_lastname , customer_address , customer_tel into :cn , :cl , :ca , :ct
12 12, }
13 13, catch (Exception e)
14 14, {
15 15, System.out.println("Not found customer !!");
16 16, return;
17 17, }
18 18, System.out.println("Customer name has Paid : " + cn);
19 19, System.out.println("Customer lastname has Paid : " + cl);
20 20, System.out.println("Customer address has Paid : " + ca);
21 21, System.out.println("Customer telephone number has Paid : " + ct);
22 22, }
23 23, }

```

ภาพที่ ข-15 ตัวอย่างไฟล์แม็บปิ้ง checkPaid

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

เครื่องมือจะวิเคราะห์ไฟล์ดังกล่าวที่ ข-14 และภาพที่ ข-15 เพื่อหาผลกระทบต่อซอร์สโค้ด และแสดงหมายเลขบรรทัดของซอร์สโค้ดที่เกิดผลกระทบ โดยเครื่องมือจะแสดงตำแหน่งที่เกิดผลกระทบต่อซอร์สโค้ด และหมายเลขบรรทัดของซอร์สโค้ดที่เกิดผลกระทบ แสดงได้ดังภาพที่ ข-16



ภาพที่ ข-16 เครื่องมือแสดงตำแหน่งและหมายเลขบรรทัดของซอร์สโค้ดที่ได้รับผลกระทบ

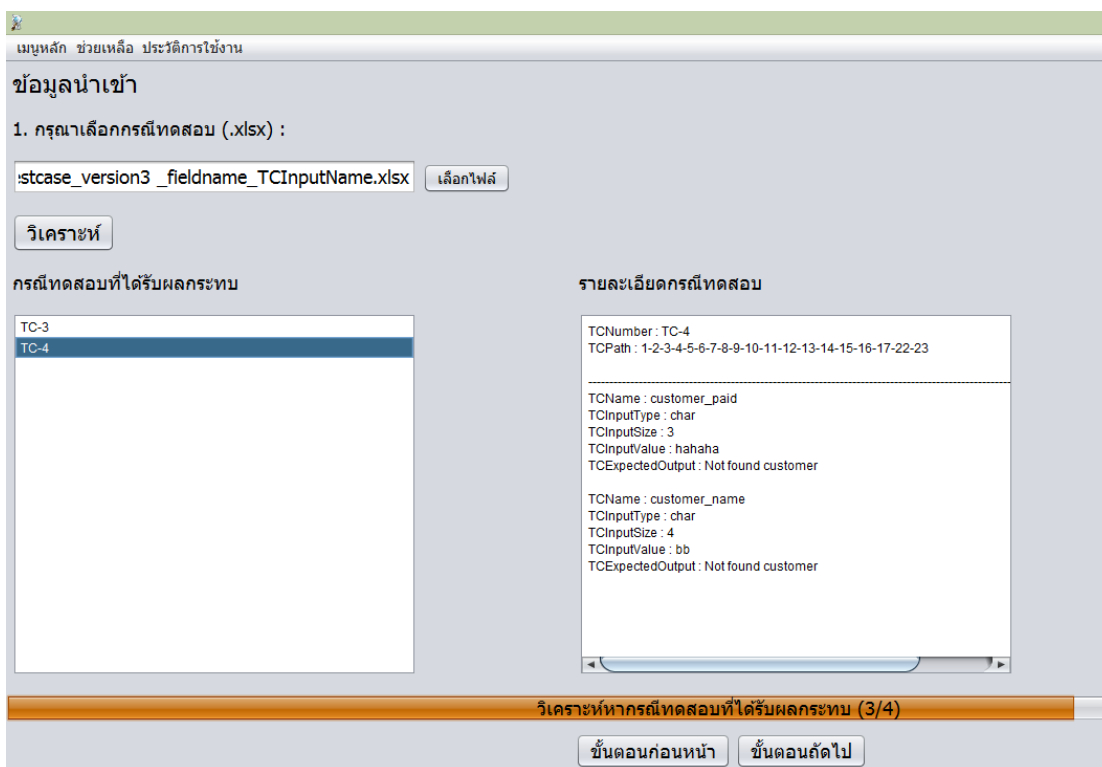
1.3) วิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ

ขั้นตอนที่สามเครื่องมือจะวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ จากภาพที่ ข-17 ตัวอย่างของไฟล์กรณีทดสอบ ซึ่งประกอบด้วย 4 กรณีทดสอบ

TCNumber	TCPath	TCInputName	TCInputType	TCInputSize	TCInputValue	TCExpectedStatus	TCExpectedOutput
TC-1	1-2-3-4-5-6-7-8-9-	product_name	char	3	bag	1	showProduct
TC-2	1-2-3-4-5-6-7-8-9- 10-11-12-13-14-15-	product_name	char	3	shirt	0	Not found product
TC-3	1-2-3-4-5-6-7-8-9-	customer_paid	char	3	yes	1	checkPaid
		customer_name	char	4	ball	1	checkPaid
TC-4	1-2-3-4-5-6-7-8-9- 10-11-12-13-14-15-	customer_paid	char	3	hahaha	0	Not found customer
		customer_name	char	4	bb	0	Not found customer

ภาพที่ ข-17 ตัวอย่างกรณีทดสอบ

จากภาพที่ ข-17 เครื่องมือวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบคือ หมายเลขกรณีทดสอบ TC-3 และ TC-4 ดังภาพที่ ข-18



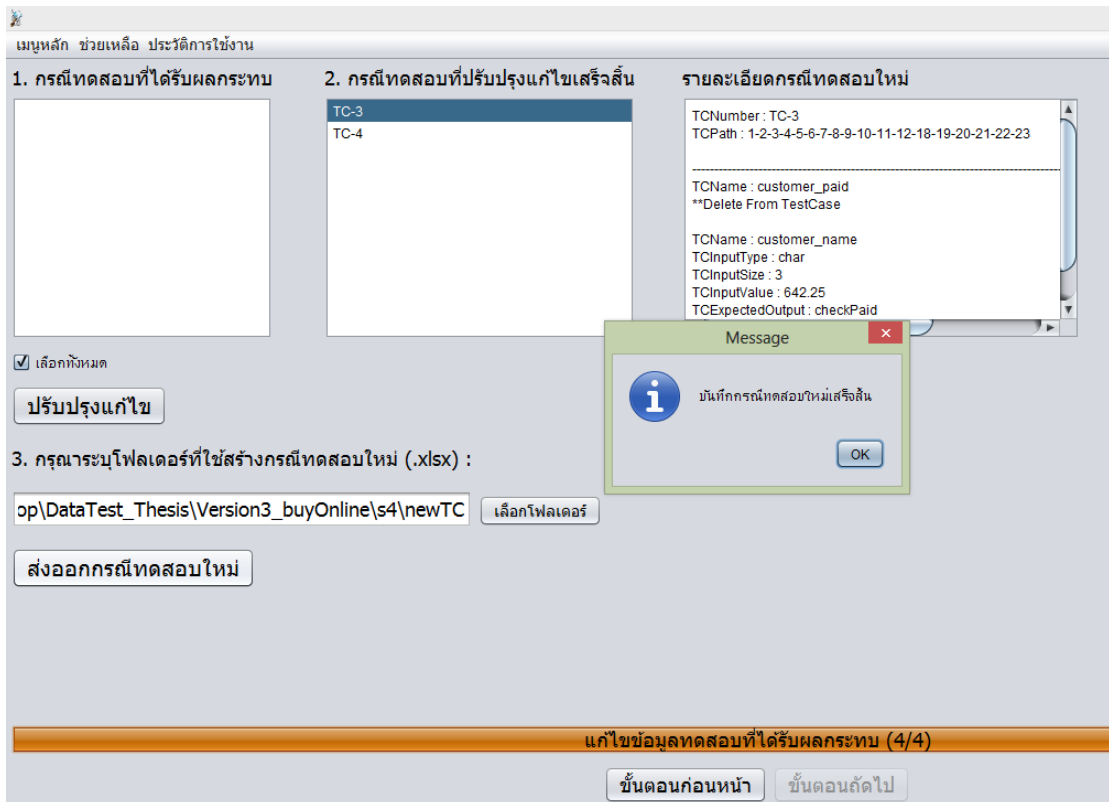
ภาพที่ ข-18 เครื่องมือแสดงรายละเอียดกรณีทดสอบที่ได้รับผลกระทบ

1.4) แก้ไขข้อมูลทดสอบที่ได้รับผลกระทบ

ขั้นตอนที่สี่ เครื่องมือจะแก้ไขข้อมูลทดสอบของกรณีทดสอบที่ได้รับผลกระทบ ดังภาพที่ ข-19 โดยจะปรับปรุงแก้ไขข้อมูลทดสอบของฟิลด์ที่เกิดการเปลี่ยนแปลง ดังนี้

- ตาราง customer
 - ฟิลด์ customer_paid ปรับปรุงแก้ไขโดยการลบข้อมูลทดสอบที่เกี่ยวข้องกับฟิลด์ดังกล่าวออกจากกรณีทดสอบ
 - ฟิลด์ customer_name ปรับปรุงแก้ไขข้อมูลทดสอบให้สอดคล้องกับการเปลี่ยนแปลงคือ จาก VARCHAR(50) เป็น DOUBLE

การเปลี่ยนแปลงที่เกิดขึ้นกับฟิลด์อื่นๆ นอกเหนือจากที่กล่าวมา เครื่องมือจะไม่สนใจ เพราะซอร์สโค้ดที่ใช้วิเคราะห์นั้นไม่ได้มีการทำงานร่วมกับฟิลด์ดังกล่าว สามารถดูรายละเอียดกรณีทดสอบที่ปรับปรุงแก้ไขข้อมูลทดสอบเสร็จสิ้นได้ ดังภาพที่ ข-20 โดยการระบายสีเหลือง เพื่อแสดงให้เห็นว่ากรณีทดสอบใดที่ได้รับการปรับปรุงแก้ไข



ภาพที่ ข-19 เครื่องมือแก้ไขข้อมูลทดสอบ

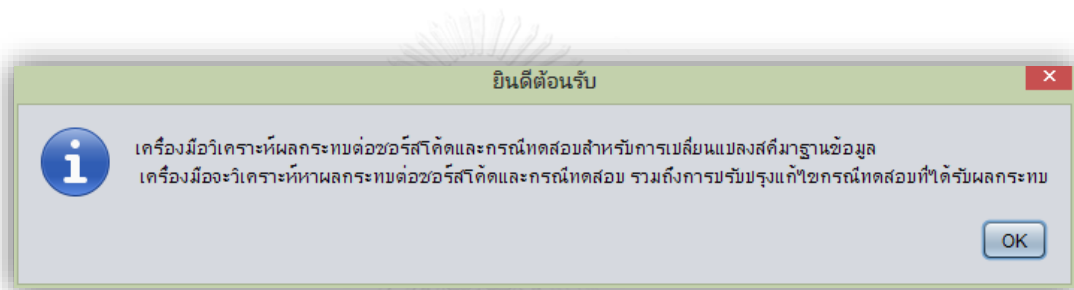
TCNumber	TCPath	TCName	TCType	TCSIZE	TCValue	ExpectedSta	TCEXPECTEDOUTPUT
TC-1	1-2-3-4-5-6-7-8-9-10-11-12-13-19-20-21-22-23	product_name	char	3	bag	1	showProduct
TC-2	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-22-23	product_name	char	3	shirt	0	Not found product
TC-3	1-2-3-4-5-6-7-8-9-10-11-12-18-19-20-21-22-23	customer_name	char	3	642.25	1	checkPaid
TC-4	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-22-23	customer_name	char	3	638999.25	0	Not found customer

ภาพที่ ข-20 ตัวอย่างกรณีทดสอบที่แก้ไขข้อมูลทดสอบเสร็จสิ้น

ภาคผนวก ค วิธีการใช้เครื่องมือ

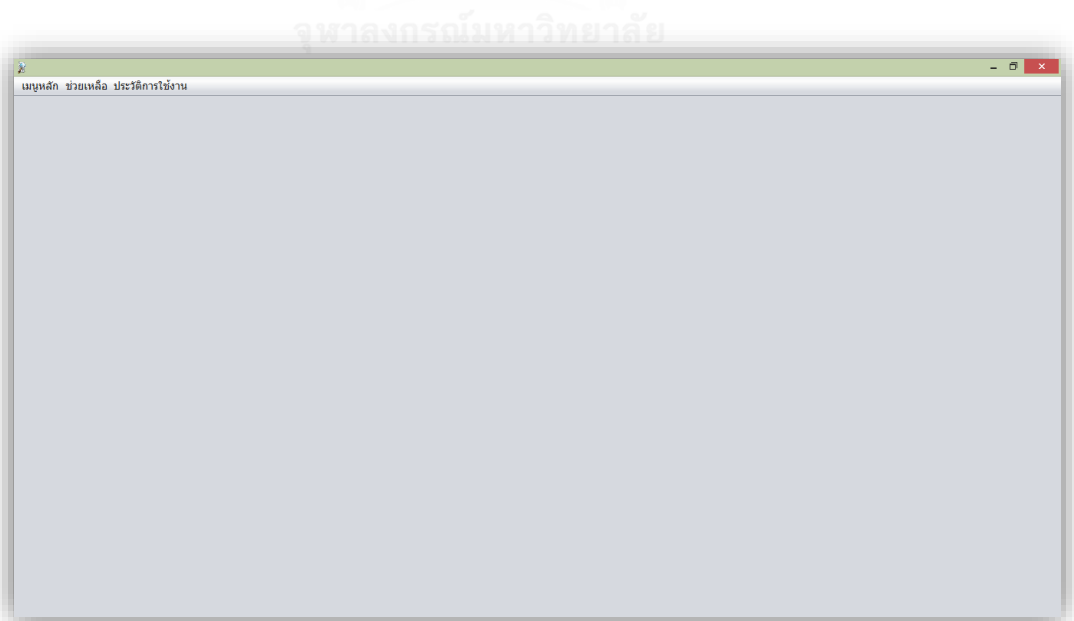
เครื่องมือวิเคราะห์ผลกระทบต่อซอร์สโค้ดและกรณีทดสอบสำหรับการเปลี่ยนแปลงสคีมาฐานข้อมูล แบ่งออกเป็น 4 ส่วน คือ การวิเคราะห์และเปรียบเทียบเอกสารสคีมาฐานข้อมูล การวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ การวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ และการแก้ไขข้อมูลทดสอบ ซึ่งมีรายละเอียดดังต่อไปนี้

- เมื่อเข้าสู่เครื่องมือจะมีข้อความแสดง “ยินดีต้อนรับ” ดังภาพที่ ค-1



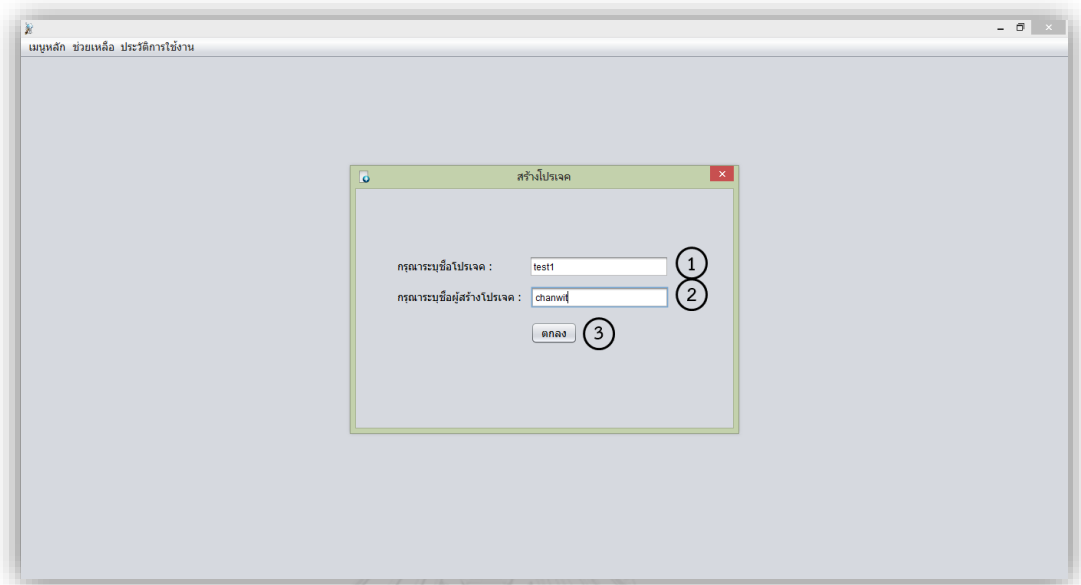
ภาพที่ ค-1 ข้อความยินดีต้อนรับ

- เข้าสู่หน้าต่างหลักของเครื่องมือ คลิกเมนูหลักเพื่อสร้างโปรเจกต์ใหม่ ดังภาพที่ ค-2



ภาพที่ ค-2 หน้าต่างหลักของเครื่องมือ

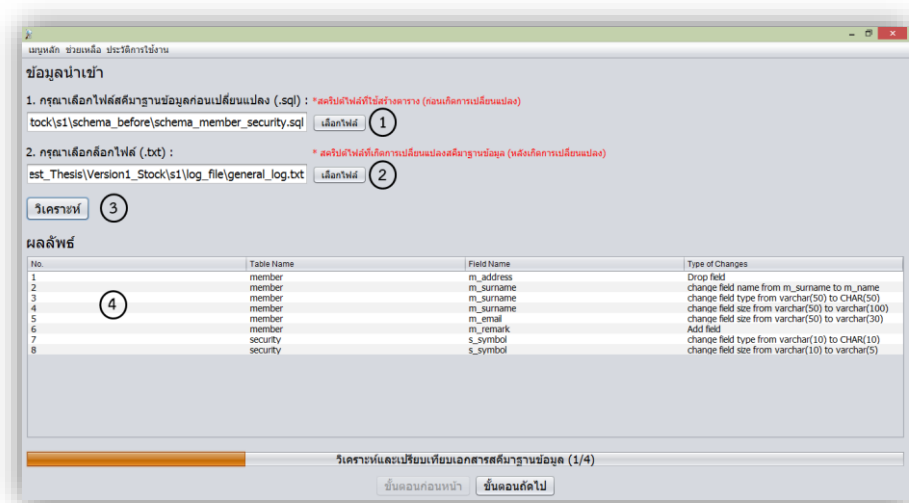
- หน้าต่างที่ใช้สร้างโปรเจกใหม่จะแสดงขึ้นมา เพื่อให้กรอกชื่อโปรเจก และชื่อผู้สร้างโปรเจก ดังภาพที่ ค-3



ภาพที่ ค-3 หน้าต่างสร้างโปรเจกใหม่

1) การวิเคราะห์และเปรียบเทียบเอกสารสคีมารฐานข้อมูล

เมื่อกรอกข้อมูลเสร็จสิ้น ก็จะไปสู่ขั้นตอนที่หนึ่ง คือ การวิเคราะห์และเปรียบเทียบเอกสารสคีมารฐานข้อมูล ดังภาพที่ ค-4

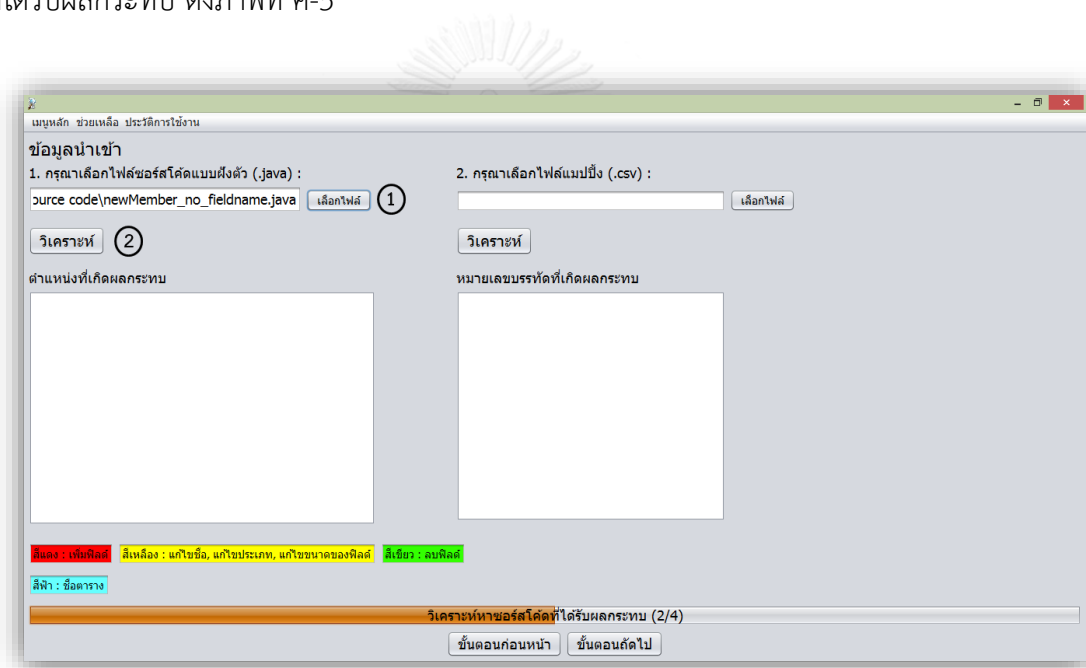


ภาพที่ ค-4 การวิเคราะห์และเปรียบเทียบเอกสารสคีมารฐานข้อมูล

จากภาพที่ ค-4 เครื่องมือจะให้ผู้ใช้อัปโหลดไฟล์คือ อัปโหลดไฟล์สคริปต์มาตรฐานข้อมูลก่อนเปลี่ยนแปลง (.SQL) และล็อกไฟล์ ตามลำดับหมายเลขในภาพ จากนั้นกดปุ่มวิเคราะห์เพื่อให้เครื่องมือวิเคราะห์หาการเปลี่ยนแปลงที่เกิดขึ้น และแสดงผลลัพธ์ผ่านทางหน้าจอ จากนั้นกดปุ่มขึ้นตอนถัดไป

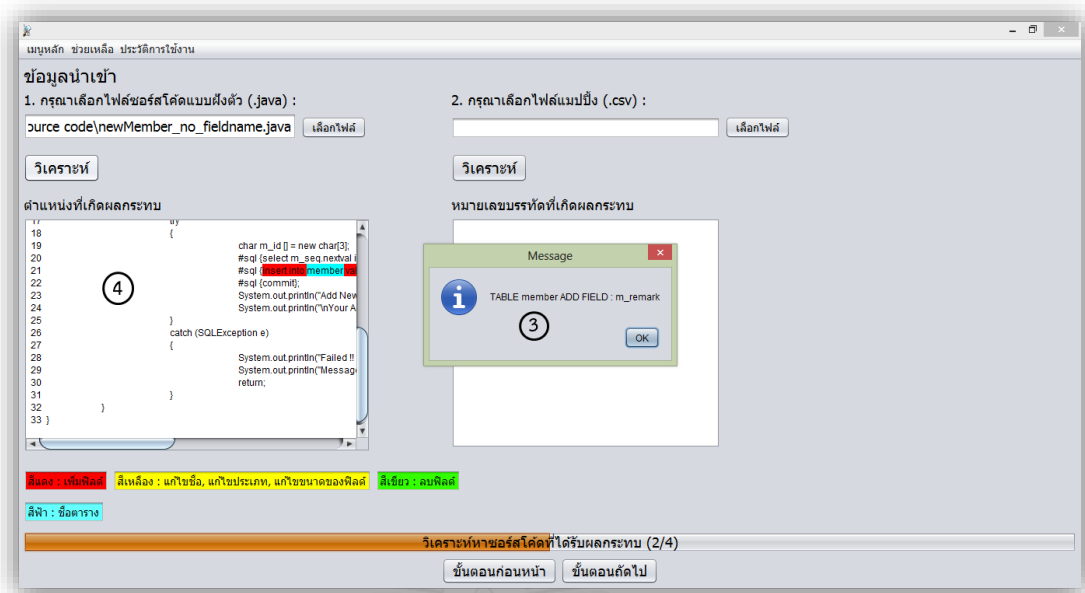
2) การวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ

หลังจากที่กดปุ่มขึ้นตอนถัดไปแล้ว จะเข้าสู่ขั้นตอนที่สองคือ การวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ ดังภาพที่ ค-5



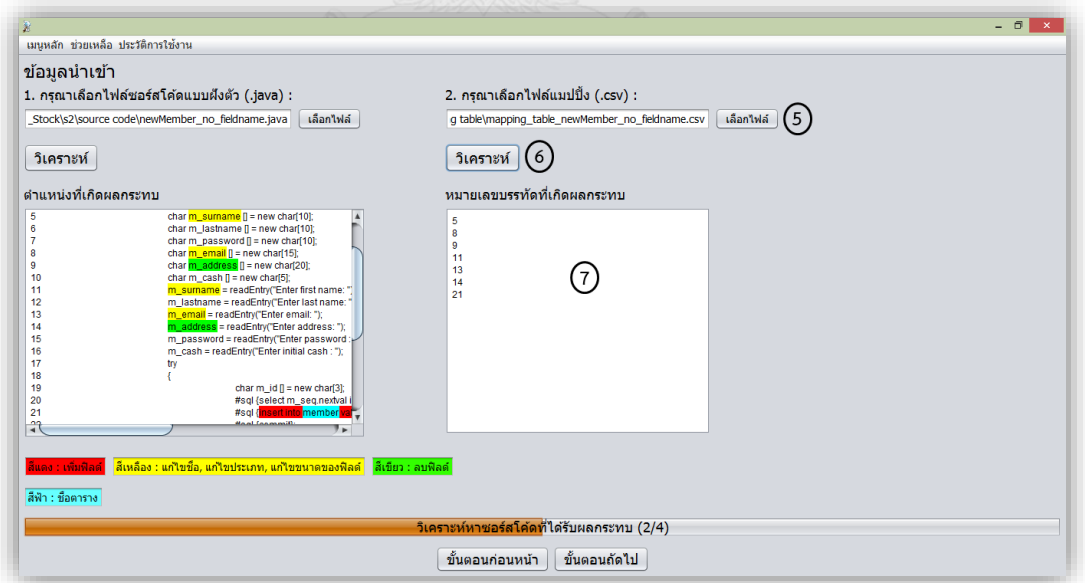
ภาพที่ ค-5 การวิเคราะห์หาซอร์สโค้ดที่ได้รับผลกระทบ

จากภาพที่ ค-5 เครื่องมือจะให้ผู้ทดสอบอัปโหลดไฟล์ซอร์สโค้ดที่ต้องการวิเคราะห์หาผลกระทบ จากการเปลี่ยนแปลงที่เกิดขึ้น กดปุ่มวิเคราะห์ ตามลำดับหมายเลข เครื่องมือจะแสดงตำแหน่งที่เกิดผลกระทบ และข้อความแจ้งเตือนกรณีเพิ่มฟิลด์ เพราะเป็นส่วนหนึ่งที่ทำให้ซอร์สโค้ดได้รับผลกระทบ ดังภาพที่ ค-6



ภาพที่ ค-6 เครื่องมือแสดงตำแหน่งที่เกิดผลกระทบในซอร์สโค้ด

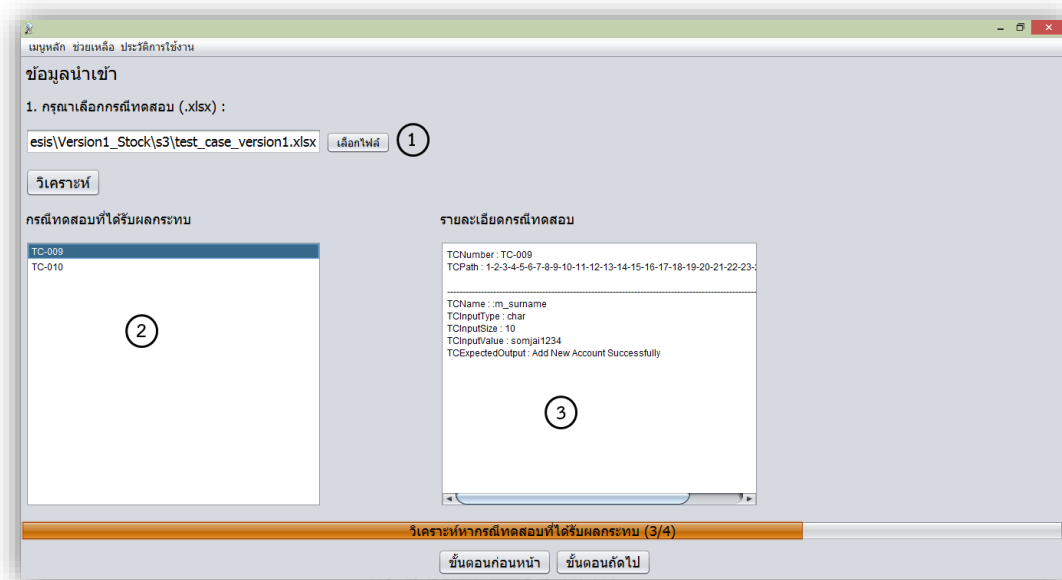
จากภาพที่ ค-7 ให้ผู้ทดสอบอัปโหลดไฟล์แมปมิ่ง (.csv) แล้วกดปุ่มวิเคราะห์ เครื่องมือจะวิเคราะห์หมายเลขบรรทัดของซอร์สโค้ดที่ได้รับผลกระทบ ดังภาพที่ ค-7 จากนั้นกดปุ่มขั้นตอนถัดไป



ภาพที่ ค-7 เครื่องมือแสดงหมายเลขบรรทัดของซอร์สโค้ดที่ได้รับผลกระทบ

3) การวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ

หลังจากที่กดปุ่มขั้นตอนถัดไปแล้ว จะเข้าสู่ขั้นตอนที่สาม คือ การวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ ดังภาพที่ ค-8

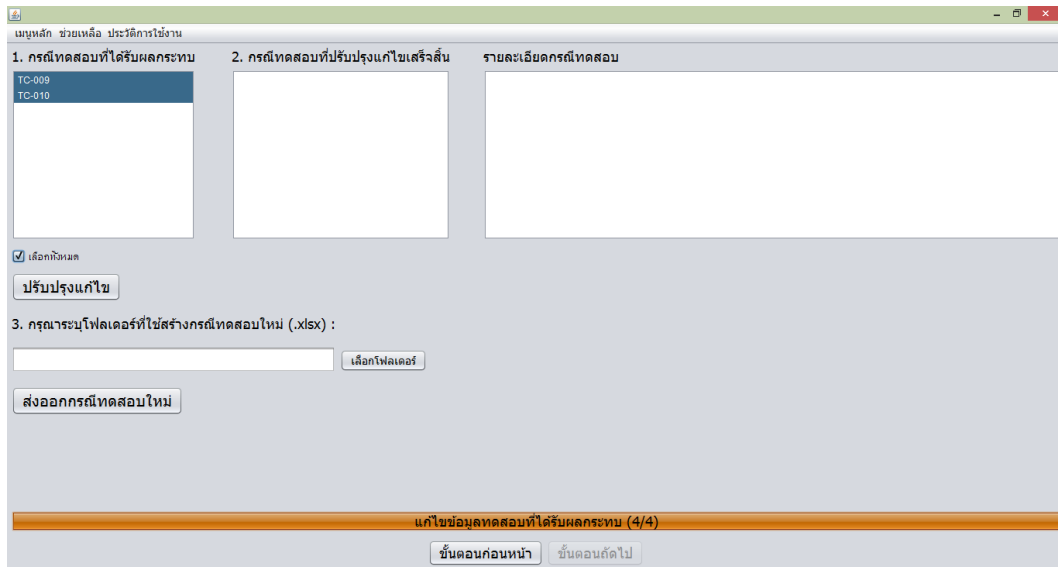


ภาพที่ ค-8 การวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ

จากภาพที่ ค-8 การวิเคราะห์หากรณีทดสอบที่ได้รับผลกระทบ ผู้ทดสอบอัปโหลดไฟล์กรณีทดสอบ (.xlsx) เพื่อให้เครื่องมือวิเคราะห์และแสดงหมายเลขกรณีทดสอบที่ได้รับผลกระทบผ่านทางหน้าจอ รวมถึงแสดงรายละเอียดกรณีทดสอบ จากนั้นกดปุ่มขั้นตอนถัดไป

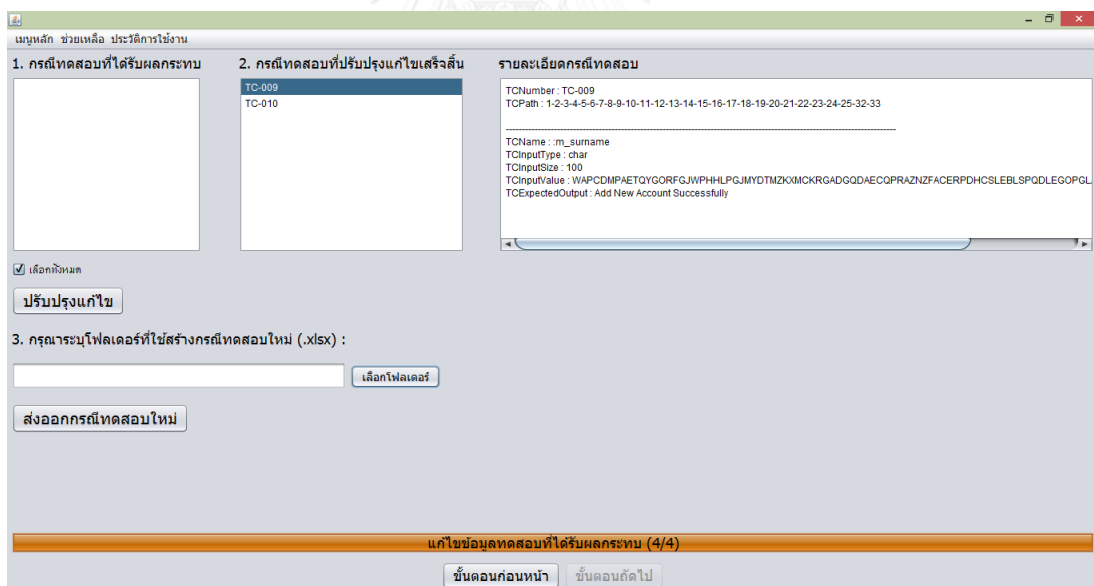
4) การแก้ไขข้อมูลทดสอบ

หลังจากที่กดปุ่มขั้นตอนถัดไปแล้ว จะเข้าสู่ขั้นตอนที่สี่ คือ การแก้ไขข้อมูลทดสอบ ดังภาพที่ ค-9



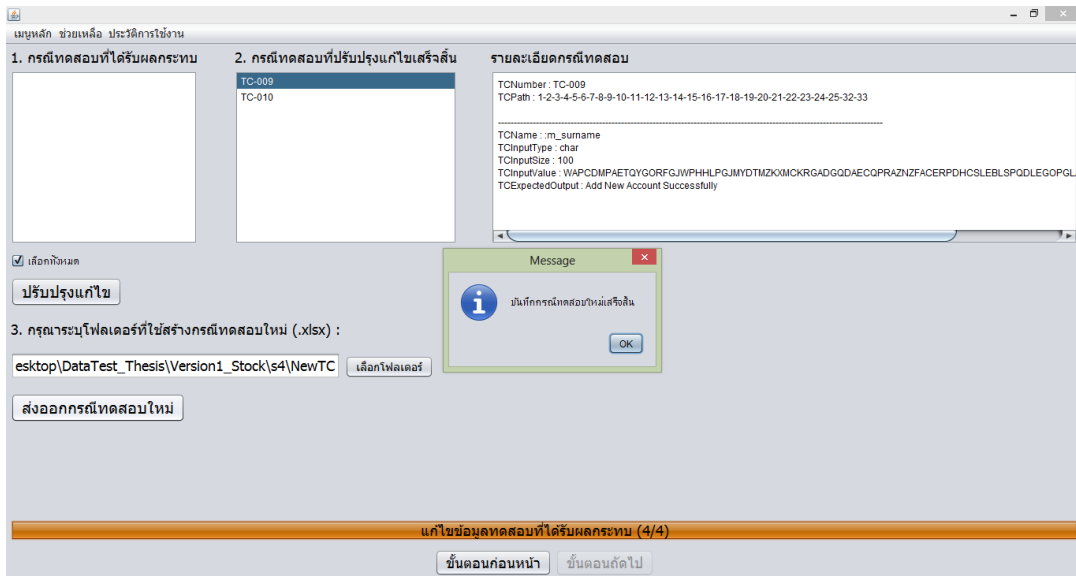
ภาพที่ ค-9 การแก้ไขข้อมูลทดสอบ

จากภาพที่ ค-9 การแก้ไขข้อมูลทดสอบ โดยให้ผู้ทดสอบเลือกกรณีทดสอบที่ได้รับผลกระทบ เพื่อปรับปรุงแก้ไข ดังภาพที่ ค-10



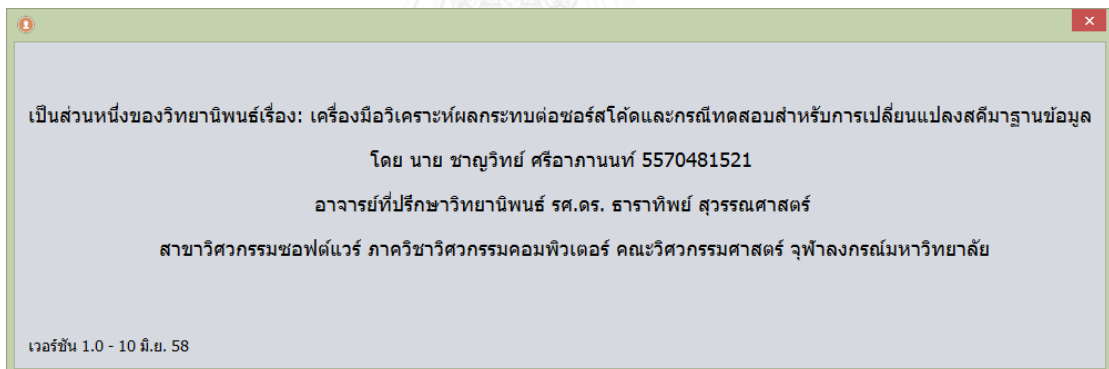
ภาพที่ ค-10 เครื่องมือแสดงกรณีทดสอบที่ปรับปรุงแก้ไขเสร็จสิ้น

จากภาพที่ ค-10 หลังจากที่ผู้ทดสอบกดปุ่มปรับปรุงแก้ไขแล้ว เครื่องมือจะแสดงหมายเลขกรณีทดสอบที่ปรับปรุงแก้ไขเสร็จสิ้น พร้อมทั้งแสดงรายละเอียดกรณีทดสอบ ลำดับถัดไปคือ ให้ผู้ทดสอบระบุไฟล์เดอร์ที่ใช้บันทึกกรณีทดสอบใหม่ และกดปุ่มส่งออกกรณีทดสอบใหม่ ดังภาพที่ ค-11



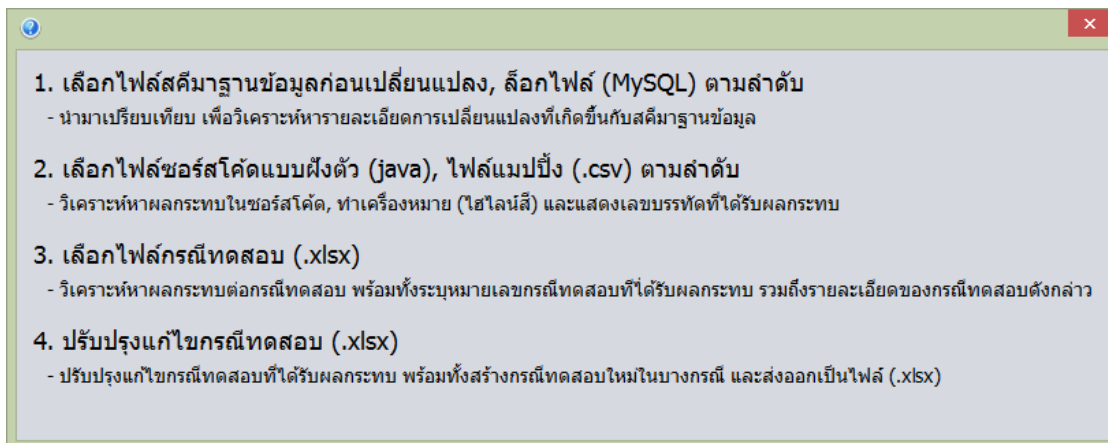
ภาพที่ ค-11 บันทึกกรณีทดสอบใหม่เสร็จสิ้น

หากผู้ทดสอบต้องการติดต่อผู้พัฒนา สามารถกดเมนูช่วยเหลือ และเลือกติดต่อผู้พัฒนา จะแสดงรายละเอียด ดังภาพที่ ค-12



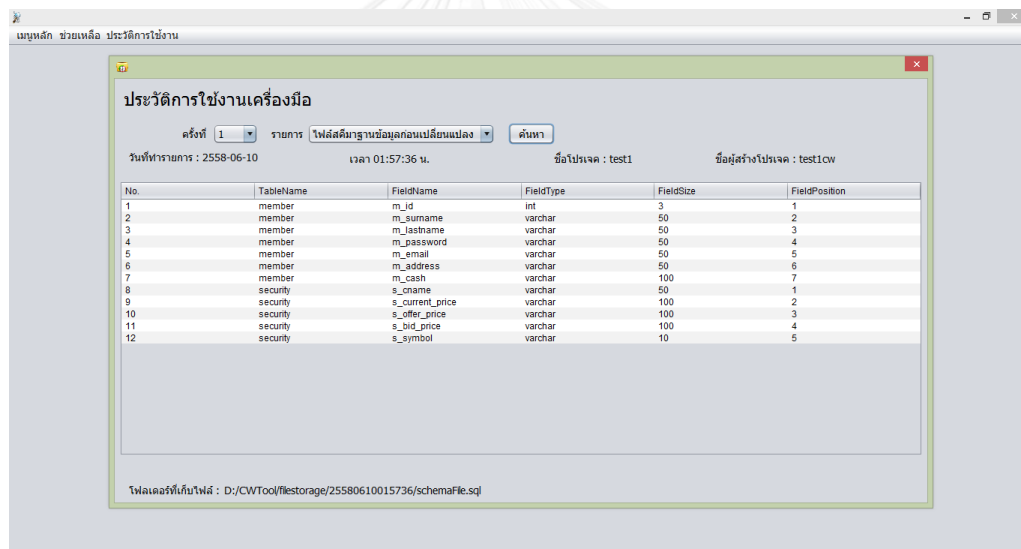
ภาพที่ ค-12 ติดต่อผู้พัฒนา

หากผู้ทดสอบมีข้อสงสัยในการทำงานโปรแกรม สามารถกดเมนูช่วยเหลือ และเลือกการใช้งานโปรแกรม จะแสดงรายละเอียด ดังภาพที่ ค-13



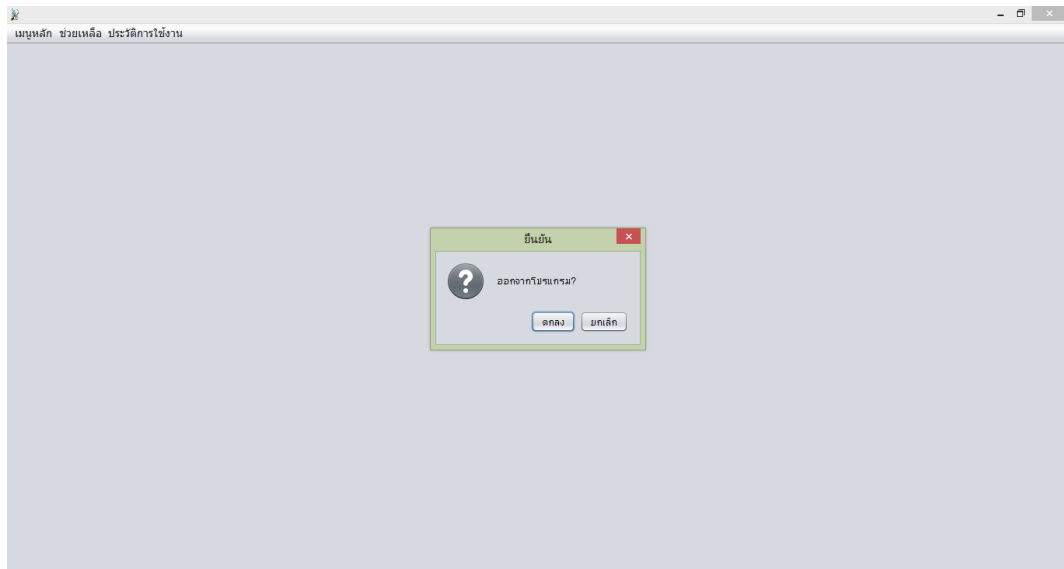
ภาพที่ ค-13 การใช้งานโปรแกรม

หากผู้ทดสอบต้องการดูประวัติการใช้งานของเครื่องมือ สามารถกดดูได้ผ่านทางเมนูประวัติการใช้งาน ดังภาพที่ ค-14



ภาพที่ ค-14 ประวัติการใช้งานเครื่องมือ

หากผู้ทดสอบต้องการออกจากโปรแกรม ให้กดเลือกเมนูหลัก และเลือกออกจากโปรแกรม ดังภาพที่ ค-15



ภาพที่ ค-15 ออกจากโปรแกรม



ประวัติผู้เขียนวิทยานิพนธ์

นาย ชาญวิทย์ ศรีอภานนท์ เกิดวันที่ 17 ตุลาคม 2532 ที่จังหวัดสุราษฎร์ธานี สำเร็จการศึกษาในหลักสูตรวิทยาศาสตรบัณฑิต คณะเทคโนโลยีและสิ่งแวดล้อม สาขา วิศวกรรมซอฟต์แวร์

มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต เมื่อปีการศึกษา 2554 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ จุฬาลงกรณ์มหาวิทยาลัยในปีการศึกษา 2554

