

แนวทางการทดสอบเรสท์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอไจล์



นางสาวสรिता คุปตยานนท์

จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2556

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR) are the thesis authors' files submitted through the University Graduate School.

AN APPROACH TO TESTING RESTFUL WEB SERVICES IN AGILE DEVELOPMENT

Miss Sarita Khuptayanon



จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

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 2013

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	แนวทางการทดสอบเรสเต็มเพล็กซ์ใยแก้วเสริมใยคาร์บอนในการพัฒนา
	แบบเอไอจี
โดย	นางสาวสรिता คุปตยานนท์
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร. ญาใจ ลีมีปะการณ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วน
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร. บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(ศาสตราจารย์ ดร. บุญเสริม กิจศิริกุล)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร. ญาใจ ลีมีปะการณ)

.....กรรมการภายนอกมหาวิทยาลัย
(อาจารย์ ดร. ภาสกร อภิรักษ์วรพินิต)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สรिता คุปตยานนท์ : แนวทางการทดสอบเรสฟูลเว็บเซอร์วิสในการพัฒนาแบบเอจิล์.
(AN APPROACH TO TESTING RESTFUL WEB SERVICES IN AGILE DEVELOPMENT) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร. ญาใจ ลิ้มปิยะกรณ์, 102 หน้า.

เซอร์วิสต่างๆ ที่ให้บริการส่วนมากบนแอปพลิเคชันมือถือที่กำลังเติบโตอยู่ในขณะนี้ มักนิยมพัฒนาเป็นเรสฟูลเว็บเซอร์วิส เนื่องจากขนาดข้อมูลที่รับส่งผ่านเรสฟูลเว็บเซอร์วิสมีขนาดเล็ก ระเบียบวิธีพัฒนาซอฟต์แวร์แบบเอจิล์เหมาะสมกับการพัฒนาแอปพลิเคชันมือถือที่มีการเปลี่ยนแปลงความต้องการรวดเร็ว ด้วยเทคนิคการพัฒนาขับเคลื่อนด้วยการทดสอบ กรณีทดสอบจะถูกพัฒนาก่อนการเขียนรหัสชุดคำสั่งในทุกๆรอบของการพัฒนา ถ้าหากมีเครื่องมือสนับสนุนให้การทดสอบสามารถกระทำได้อย่างอัตโนมัติ จะส่งผลให้สามารถพัฒนาบริการต่างๆได้มากและเร็วยิ่งขึ้น รวมถึงสามารถลดค่าใช้จ่ายในการทดสอบ งานวิจัยนี้จึงนำเสนอแนวทางและพัฒนาระบบเพื่อสนับสนุนการทดสอบเรสฟูลเว็บเซอร์วิส นอกจากนี้ กรณีทดสอบที่สร้างขึ้นแบบอัตโนมัติในบริบทการพัฒนาซอฟต์แวร์แบบเอจิล์ในงานวิจัยนี้ สามารถนำกลับมาใช้ใหม่สำหรับการทดสอบในลักษณะอื่นๆได้ในภายหลัง

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาควิชา วิศวกรรมคอมพิวเตอร์

ลายมือชื่อนิสิต

สาขาวิชา วิศวกรรมซอฟต์แวร์

ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก

ปีการศึกษา 2556

5571000521 : MAJOR SOFTWARE ENGINEERING

KEYWORDS: TEST CASE GENERATION / RESTFUL WEB SERVICES / AGILE /
REUSABILITY

SARITA KHUPTAYANON: AN APPROACH TO TESTING RESTFUL WEB SERVICES IN AGILE DEVELOPMENT. ADVISOR: ASSOC. PROF. YACHAI LIMPIYAKORN, Ph.D., 102 pp.

Today various growing services provided as mobile applications are often developed with RESTful web services as the size of data transferred over the RESTful web services is small. Agile software development methodology suits for the development of mobile applications with rapid changing requirements. When using the technique of Test -Driven Development, the test cases will be developed prior to coding during each cycle of development or sprint. If a tool exists to support automated testing, it will lead to more and faster development of these services. The cost of testing will be reduced as well. This research thus presents an approach and develops a system to support the RESTful web service test. Moreover, the test cases automatically generated in the context of Agile software development presented in this research work can be reused later for other types of testing.



Department: Computer Engineering Student's Signature

Field of Study: Software Engineering Advisor's Signature

Academic Year: 2013

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความอนุเคราะห์อย่างยิ่งของ รองศาสตราจารย์ ดร. ญาใจ ลิ้มปิยะกรณ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งได้ให้ความรู้ คำแนะนำ ข้อคิดเห็นต่างๆ และช่วยแก้ไขข้อบกพร่องต่างๆ ด้วยความเอาใจใส่ เพื่อให้การทำวิทยานิพนธ์ฉบับนี้สมบูรณ์ ผู้วิจัยมีความซาบซึ้งในความกรุณาอันดีจากอาจารย์ และขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล และอาจารย์ ดร.ภาสกร อภิรักษัรพินิต กรรมการสอบวิทยานิพนธ์ ที่กรุณาเสียสละเวลา ให้คำแนะนำ ตรวจสอบ และแก้ไขวิทยานิพนธ์ฉบับนี้

ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ และครอบครัว ที่ให้กำลังใจและการสนับสนุนช่วยเหลือในด้านต่างๆ กราบขอบพระคุณอาจารย์ทุกท่านที่ได้อบรมสั่งสอนวิชาความรู้ให้ผู้วิจัยจนสามารถทำวิทยานิพนธ์นี้ได้สำเร็จ

สุดท้ายนี้ผู้วิจัยขอขอบคุณเพื่อนๆ ทุกคนที่คอยได้ให้กำลังใจและความช่วยเหลือต่างๆ ตลอดจนผู้มีพระคุณทุกท่านที่ได้กล่าวถึง คุณประโยชน์และคุณค่าอันเกิดจากการศึกษาวิจัยครั้งนี้ ผู้วิจัยขอมอบแต่บิดา มารดา ครู อาจารย์ และผู้มีพระคุณทุกท่านด้วยความซาบซึ้งใจเป็นอย่างยิ่ง

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญรูป.....	ต
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	1
1.3 ขอบเขตของการวิจัย	1
1.4 ขั้นตอนและวิธีดำเนินการวิจัย	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์.....	2
1.7 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1.1 เรสท์ (REST : Representational State Transfer) [1].....	4
2.1.2 เจสัน (JSON : JavaScript Object Notation) [2].....	4
2.1.3 JUnit [3].....	5
2.2 งานวิจัยที่เกี่ยวข้อง	6
2.2.1 A Test Automation Framework Based on WEB [4].....	6
2.2.2 Test-the-REST: An Approach to Testing RESTful Web Service [5].....	7
บทที่ 3 แนวคิดวิธีดำเนินการวิจัย	10
3.1 ภาพรวมของแนวทางทดสอบเรสท์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอไจล์.....	10
3.1.1 Create User Story.....	11
3.1.2 Review.....	13
3.1.3 Change User Story	13

3.1.4 Generate Test Cases.....	13
3.1.5 Sprint Test Cases.....	16
3.1.6 Transform with Test Case Converter	17
3.1.7 Execute JUnit Source Code	17
3.1.8 Report Test Result.....	18
บทที่ 4 การออกแบบและพัฒนาระบบ.....	19
4.1 ข้อกำหนดเบื้องต้นของระบบ.....	19
4.1.1 ผู้ใช้งาน (User).....	19
4.1.2 ข้อมูลนำเข้า (Input)	19
4.1.3 ข้อมูลนำออก (Output)	19
4.1.4 ข้อจำกัดของระบบ (Constraint)	19
4.2 ความต้องการเชิงหน้าที่ (Functional Requirements).....	19
4.3 การออกแบบระบบ.....	21
4.3.1 แผนภาพยูสเคส.....	21
4.3.2 คำอธิบายยูสเคส.....	21
4.4 การพัฒนาระบบ.....	25
4.4.1 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา	25
4.4.2 การติดตั้งซอฟต์แวร์ในการพัฒนาระบบ.....	26
4.4.3 การพัฒนาส่วนต่อประสาน	26
4.4.3.1 ส่วนของการนำเข้าไฟล์และแสดงผล.....	26
4.4.3.2 ส่วนของรายการไฟล์เอกสารเอกซ์เอ็มแอล	27
4.4.3.3 ส่วนของรายการสคริปต์ทดสอบ	27
บทที่ 5 การทดสอบและประเมินระบบ.....	29
5.1 การทดสอบระบบ	29
5.2 การประเมินผลระบบ.....	62
5.2.1 ข้อมูลนำเข้า	62
5.2.2 ข้อมูลนำออก.....	65

5.2.3 การประมวลผลสคริปต์ทดสอบ.....	71
5.2.3.1 การนำเข้าสคริปต์ทดสอบ	72
5.2.3.2 การประมวลผลสคริปต์ทดสอบ	75
บทที่ 6 สรุปผลการวิจัย และข้อเสนอแนะ.....	76
6.1 สรุปผลการวิจัย.....	76
6.2 ข้อจำกัด	76
6.3 แนวทางการวิจัยต่อ	76
รายการอ้างอิง	77
ภาคผนวก.....	78
ภาคผนวก ก ผลการทดสอบ ระบบทดสอบเรสต์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอไจล์.....	79
ภาคผนวก ข การติดตั้งเครื่องมือและสร้างโปรเจกอะปาเช่ มาเว่น เวอร์ชัน 3.2.2.....	89
ภาคผนวก ค การติดตั้งเครื่องมืออะปาเช่ ทอมแคท เวอร์ชัน 6.0.41 และติดตั้งระบบทดสอบ เรสต์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอไจล์.....	95
ประวัติผู้เขียนวิทยานิพนธ์	102

สารบัญตาราง

ตารางที่ 1	ลักษณะการทำงานของ REST Method.....	4
ตารางที่ 2	รูปแบบการเขียน User story ที่นำเสนอในงานวิจัย	12
ตารางที่ 3	ข้อมูลที่สกัดจาก User Story ในรูปแบบเอกซ์เอ็มแอล	14
ตารางที่ 4	ข้อมูลที่สกัดจาก User Story เพื่อสร้างลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอล	16
ตารางที่ 5	ความต้องการเชิงหน้าที่.....	20
ตารางที่ 6	ค่าที่กำหนดใน User story	20
ตารางที่ 7	คำอธิบายยูสเคส Import Excel	22
ตารางที่ 8	คำอธิบายยูสเคส Extract Excel User Story	22
ตารางที่ 9	คำอธิบายยูสเคส Generate Test Cases in XML	23
ตารางที่ 10	คำอธิบายยูสเคส Generate Sequence Test Cases in XML	23
ตารางที่ 11	คำอธิบายยูสเคส Generate Script Test in JUnit.....	24
ตารางที่ 12	คำอธิบายยูสเคส Request Restful Web Service	24
ตารางที่ 13	คำอธิบายยูสเคส Display Result Script Test on User Interface.....	25
ตารางที่ 14	การทดสอบการกรนำเข้าไฟล์ User story ในรูปแบบเอกซ์เซล 2010.....	32
ตารางที่ 15	ทดสอบการนำเข้า User story empty	33
ตารางที่ 16	ทดสอบค่า Request Method ไม่ได้ระบุเป็น GET หรือ POST.....	33
ตารางที่ 17	ทดสอบค่า Request URL เป็นว่าง.....	33
ตารางที่ 18	ทดสอบค่า Assertion method ไม่ได้เป็นค่าตามที่กำหนด.....	34
ตารางที่ 19	ทดสอบค่า Assertion method เป็น NotNull	34
ตารางที่ 20	ทดสอบค่า Assertion method เป็น Null	35
ตารางที่ 21	ทดสอบค่า Assertion method เป็น True	35
ตารางที่ 22	ทดสอบค่า Assertion method เป็น False.....	35

ตารางที่ 23 ทดสอบค่า Assertion method เป็น NotEquals	36
ตารางที่ 24 ทดสอบค่า Assertion method เป็น Equals	36
ตารางที่ 25 ทดสอบค่า Assertion method เป็น EqualsOrMoreThan.....	37
ตารางที่ 26 ทดสอบค่า Assertion method เป็น EqualsOrLessThan.....	37
ตารางที่ 27 ทดสอบค่า Assertion method เป็น MoreThan	37
ตารางที่ 28 ทดสอบค่า Assertion method เป็น NotEquals	38
ตารางที่ 29 ทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล.....	38
ตารางที่ 30 ทดสอบจำนวนกรณีทดสอบ.....	39
ตารางที่ 31 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Request Method แบบ GET	39
ตารางที่ 32 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Request Method แบบ POST	39
ตารางที่ 33 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Request Method แบบ PUT	40
ตารางที่ 34 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Request Method แบบ DELETE	40
ตารางที่ 35 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น NotNull	41
ตารางที่ 36 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น Null	41
ตารางที่ 37 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น True	42
ตารางที่ 38 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น False	42
ตารางที่ 39 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น NotEquals	43
ตารางที่ 40 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น Equals	43
ตารางที่ 41 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น EqualsOrMoreThan	44
ตารางที่ 42 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น EqualsOrLessThan.....	44
ตารางที่ 43 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น EqualsOrLessThan.....	45
ตารางที่ 44 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น NotEquals	45

ตารางที่ 45 ทดสอบความถูกต้องของกรณีทดสอบ.....	46
ตารางที่ 46 ทดสอบการสร้างลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอล.....	46
ตารางที่ 47 ทดสอบความถูกต้องของลำดับการทดสอบ.....	46
ตารางที่ 48 ทดสอบการสร้างสคริปต์ทดสอบ ในรูปแบบ JUnit source code.....	47
ตารางที่ 49 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Request Method แบบ GET.....	47
ตารางที่ 50 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Request Method แบบ POST.....	48
ตารางที่ 51 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Request Method แบบ PUT.....	48
ตารางที่ 52 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Request Method แบบ DELETE.....	49
ตารางที่ 53 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น NotNull.....	49
ตารางที่ 54 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น Null.....	50
ตารางที่ 55 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น True.....	50
ตารางที่ 56 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น False.....	51
ตารางที่ 57 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น That.....	51
ตารางที่ 58 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น Equals.....	52
ตารางที่ 59 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น EqualsOrMoreThan.....	52

ตารางที่ 60 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น EqualsOrLessThan.....	53
ตารางที่ 61 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น MoreThan.....	53
ตารางที่ 62 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น NotEquals	54
ตารางที่ 63 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิสที่ต้องการจะทดสอบและสร้างผลลัพธ์....	54
ตารางที่ 64 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Request Method แบบ GET และสร้างผลลัพธ์.....	55
ตารางที่ 65 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Request Method แบบ POST และสร้างผลลัพธ์.....	55
ตารางที่ 66 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Request Method แบบ PUT และสร้างผลลัพธ์.....	56
ตารางที่ 67 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Request Method แบบ PUT และสร้างผลลัพธ์.....	56
ตารางที่ 68 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น NotNull และสร้างผลลัพธ์.....	57
ตารางที่ 69 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น Null และสร้างผลลัพธ์.....	57
ตารางที่ 70 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น True และสร้างผลลัพธ์.....	58
ตารางที่ 71 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น False และสร้างผลลัพธ์.....	58
ตารางที่ 72 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น That และสร้างผลลัพธ์.....	59
ตารางที่ 73 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น Equals และสร้างผลลัพธ์.....	59

ตารางที่ 74 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น EqualsOrMoreThan และสร้างผลลัพธ์	60
ตารางที่ 75 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น EqualsOrLessThan และสร้างผลลัพธ์	60
ตารางที่ 76 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น MoreThan และสร้างผลลัพธ์	61
ตารางที่ 77 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น NotEquals และสร้างผลลัพธ์	61
ตารางที่ 78 ทดสอบการแสดงผลลัพธ์ที่หน้าจการทำงานหลักจากไฟล์ลำดับกรณีทดสอบ	62
ตารางที่ 79 ทดสอบการแสดงผลลัพธ์ที่หน้าจการทำงานหลักจากไฟล์ลำดับกรณีทดสอบ	62
ตารางที่ 80 ตาราง User story	79

สารบัญรูป

รูปที่ 1 การรับส่งข้อมูลบนเครือข่ายในรูปแบบเจสัน.....	5
รูปที่ 2 ตัวอย่างสคริปต์ทดสอบในรูปแบบ JUnit source code	6
รูปที่ 3 กรอบงานการทดสอบอัตโนมัติบนพื้นฐานเว็บ [4].....	7
รูปที่ 4 กรณีทดสอบ [5].....	7
รูปที่ 5 การระบุปลายทางในการแสดงผลลัพธ์ [5].....	8
รูปที่ 6 การระบุลำดับกรณีทดสอบ [5].....	8
รูปที่ 7 ผลลัพธ์ของกรณีทดสอบ TTR-6-1.xml [5].....	8
รูปที่ 8 ขั้นตอนวิธีการทำงานของระบบ.....	10
รูปที่ 9 กรณีทดสอบ TEST-CASE-ID-001.xml	15
รูปที่ 10 กรณีทดสอบ TEST-CASE-ID-002.xml.....	15
รูปที่ 11 ลำดับการทดสอบ Sequence.xml	16
รูปที่ 12 สคริปต์ทดสอบ.....	17
รูปที่ 13 กำหนดค่า Library.....	18
รูปที่ 14 ผลลัพธ์ของการทดสอบใน Sequence.xml	18
รูปที่ 15 แผนภาพยูสเคสของระบบทดสอบเรสท์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอไจล์	21
รูปที่ 16 หน้าจอของระบบทดสอบเรสท์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอไจล์	27
รูปที่ 17 รายการเพิ่มเอกสาร User Story.....	27
รูปที่ 18 รายการเพิ่มเอกสารเอกซ์เอ็มแอล.....	27
รูปที่ 19 รายการเพิ่มเอกสารเอกซ์เอ็มแอล.....	28
รูปที่ 20 ต้นแบบสำหรับสร้าง JUnit source code	29
รูปที่ 21 ต้นแบบสำหรับเรียกการทดสอบเรสท์ฟูลเว็บเซอร์วิส.....	31
รูปที่ 22 กรณีทดสอบ TEST-CASE-ID-001 ในรูปแบบเอกซ์เอ็มแอลแบบ GET	65
รูปที่ 23 กรณีทดสอบ TEST-CASE-ID-002 ในรูปแบบเอกซ์เอ็มแอลแบบ POST.....	66

รูปที่ 24 กรณีทดสอบ TEST-CASE-ID-003 ในรูปแบบเอกซ์เอ็มแอลแบบ GET	66
รูปที่ 25 กรณีทดสอบ TEST-CASE-ID-004 ในรูปแบบเอกซ์เอ็มแอลแบบ POST	67
รูปที่ 26 กรณีทดสอบ TEST-CASE-ID-005 ในรูปแบบเอกซ์เอ็มแอลแบบ GET	67
รูปที่ 27 ลำดับทดสอบในรูปแบบเอกซ์เอ็มแอล	68
รูปที่ 28 สคริปต์ทดสอบในรูปแบบ JUnit source code	70
รูปที่ 29 ลำดับการทดสอบแบบแสดงผลลัพธ์	71
รูปที่ 30 หน้าจอแสดงผลลัพธ์ที่อ่านค่าจากไฟล์ลำดับการทดสอบ	71
รูปที่ 31 เครื่องมือสปริง ทูล สวีท เวอร์ชัน 3.2.2	72
รูปที่ 32 หน้าจอการทำงานของเครื่องมือสปริง ทูล สวีท	72
รูปที่ 33 การนำเข้าสคริปต์ทดสอบ.....	73
รูปที่ 34 หน้าจอการเลือกเพิ่มเอกสารสคริปต์ทดสอบ	73
รูปที่ 35 หน้าจอการตั้งค่า Maven plugin connectors	74
รูปที่ 36 หน้าจอแสดงสคริปต์ทดสอบตามโครงสร้าง Maven project.....	74
รูปที่ 37 การประมวลผลสคริปต์ทดสอบ.....	75
รูปที่ 38 ผลลัพธ์การประมวลผลด้วย JUnit.....	75
รูปที่ 39 ผลลัพธ์การประมวลผลด้วย มาเว่น.....	75
รูปที่ 40 กรณีทดสอบ TEST-CASE-ID-001 ในรูปแบบเอกซ์เอ็มแอลแบบ GET	83
รูปที่ 41 กรณีทดสอบ TEST-CASE-ID-001 ในรูปแบบเอกซ์เอ็มแอลแบบ POST	83
รูปที่ 42 กรณีทดสอบ TEST-CASE-ID-001 ในรูปแบบเอกซ์เอ็มแอลแบบ PUT	84
รูปที่ 43 กรณีทดสอบ TEST-CASE-ID-001 ในรูปแบบเอกซ์เอ็มแอลแบบ DELETE	85
รูปที่ 44 ลำดับกรณีทดสอบ.....	85
รูปที่ 45 สคริปต์ทดสอบ.....	88
รูปที่ 46 ผลลัพธ์ในไฟล์ลำดับกรณีทดสอบ	88
รูปที่ 47 การกำหนดค่า MAVEN_HOME.....	89

รูปที่ 48 หน้าจอแสดงเวอร์ชัน.....	90
รูปที่ 49 หน้าจอสร้างมาเว่น โพรเจค.....	90
รูปที่ 50 หน้าจอเลือกส่วนประกอบ.....	91
รูปที่ 51 กำหนดชื่อมาเว่น โพรเจค และไฟล์นำออก.....	91
รูปที่ 52 โครงสร้างมาเว่น โพรเจค.....	92
รูปที่ 53 ไฟล์ pom.xml.....	92
รูปที่ 54 ผลลัพธ์การดาวน์โหลด Library ต่างๆ.....	93
รูปที่ 55 เพิ่มเอกสาร Library	93
รูปที่ 56 เพิ่มเอกสารไฟล์ที่บีบอัด.....	94
รูปที่ 57 หน้าจอเริ่มต้นอะปาเช่ ทอมแคท เวอร์ชัน 6.0.41	95
รูปที่ 58 หน้าจอแสดงข้อตกลง.....	96
รูปที่ 59 หน้าจอการเลือกส่วนประกอบ.....	96
รูปที่ 60 หน้าจอสำหรับตั้งค่า	97
รูปที่ 61 หน้าจอการเลือกเพิ่มปลายทางของ Java Virtual Machine	97
รูปที่ 62 หน้าจอการเลือกสถานที่ที่จะติดตั้ง	98
รูปที่ 63 สถานะระหว่างติดตั้ง.....	98
รูปที่ 64 การติดตั้งอะปาเช่ ทอมแคท เสร็จสมบูรณ์.....	99
รูปที่ 65 หน้าจอคุณลักษณะอะปาเช่ ทอมแคท	99
รูปที่ 66 หน้าจอหลักการจัดการทอมแคท.....	100
รูปที่ 67 หน้าจอหลักการจัดการแอปพลิเคชัน.....	100
รูปที่ 68 หน้าจอการเลือกไฟล์เพื่อติดตั้งแอปพลิเคชัน.....	101
รูปที่ 69 หน้าจอการทำงานหลักของระบบ	101

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

การสื่อสารทางอินเทอร์เน็ตทำให้เกิดระบบกระจาย (Distributed System) ซึ่งได้ให้บริการ (Provide services) และข้อมูลต่างๆ ไร้มากมายทั้งในรูปแบบที่เป็น Public และ Private ระบบกระจายต่างๆเหล่านี้ สามารถเรียกใช้เซอร์วิสและเข้าถึงข้อมูลซึ่งกันและกัน ทำให้เป็นเรื่องยากที่จะสามารถทดสอบว่า ระบบหนึ่งๆ ทำงานได้ผลลัพธ์ตามที่ต้องการโดยสมบูรณ์ อันเนื่องมาจากปัจจัยหลายประการ ที่เห็นเด่นชัดที่สุด คือ เซอร์วิส ต่างๆที่ระบบเหล่านี้ให้บริการ เมื่อมองในมุมมองของผู้ใช้งานแล้วมักจะถูกซ่อนเอาไว้ ผู้วิจัยจึงมีแนวคิดที่จะทำให้การทดสอบความถูกต้องของระบบกระจายเหล่านี้ง่ายขึ้น โดยการนำเข้ากรณีทดสอบ(Test Case) ผ่านเครื่องมือที่พัฒนาขึ้นเพื่อให้ได้ผลการทดสอบโดยอัตโนมัติ เช่นนี้แล้วก็จะทำให้การทดสอบเซอร์วิสต่างๆ เป็นไปได้ง่ายขึ้น

การพัฒนาซอฟต์แวร์แบบเอจิล (Agile) ด้วยเทคนิคการพัฒนาขับเคลื่อนด้วยการทดสอบ (Test-Driven Development— TDD) จะต้องมีการทดสอบทุกๆ Sprint ซึ่งในแต่ละ Sprint นักพัฒนาจะสร้างกรณีทดสอบจาก User Story ก่อนการเขียนโค้ดและทำการทดสอบเอง แต่สุดท้ายแล้วต้องให้นักทดสอบ (Tester) มาทดสอบอีกครั้ง เนื่องจากในมุมมองของนักพัฒนา อาจจะไม่สามารถทดสอบได้ครบถ้วนทุกกรณี และนักทดสอบอาจมีมุมมองการทดสอบที่แตกต่างออกไป ดังนั้นเมื่อสิ้นสุดในแต่ละ Sprint แล้ว นักทดสอบจึงต้องทดสอบทุกๆฟังก์ชันใหม่หมดทุกครั้ง หากมีข้อผิดพลาด หรือมีการเปลี่ยนแปลงความต้องการกะทันหัน กรณีทดสอบชุดเดิมอาจจะต้องถูกนำมาใช้อีกครั้งเพื่อทดสอบในกรณีดังกล่าว หรือเมื่อเซอร์วิสของผู้ให้บริการมีการเปลี่ยน API ก็ สามารถทดสอบได้อย่างรวดเร็วว่าส่งผลกระทบต่อการพัฒนาหรือไม่

ปัจจุบัน เซอร์วิสต่างๆที่ให้บริการส่วนมากมักนิยมพัฒนาเป็นเรสต์ฟูลเว็บเซอร์วิสเนื่องจากข้อมูลที่ส่งให้เรสต์ฟูลเว็บเซอร์วิสมีขนาดเล็ก ถ้าหากมีเครื่องมือสนับสนุนให้การทดสอบที่เรียกใช้เรสต์ฟูลเว็บเซอร์วิสเซอร์วิสสามารถกระทำได้อย่างรวดเร็ว จะเป็นการขับเคลื่อนธุรกิจองค์กรให้สามารถพัฒนาบริการต่างๆได้มากและเร็วยิ่งขึ้น เช่น แอปพลิเคชันบนมือถือที่กำลังเติบโตอยู่ในขณะนี้ ก็จะสามารถให้บริการต่างๆ ได้ง่าย สะดวก รวดเร็ว สอดคล้องกับชีวิตประจำวัน รวมทั้งสามารถตอบสนองต่อธุรกิจที่มีการเปลี่ยนแปลงอย่างรวดเร็ว

1.2 วัตถุประสงค์ของการวิจัย

เพื่อนำเสนอแนวทางการทดสอบเรสต์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอจิล และพัฒนาระบบต้นแบบเพื่อสนับสนุนแนวทางที่นำเสนอ

1.3 ขอบเขตของการวิจัย

1. User Story จะต้องให้รายละเอียดเพียงพอกับการสร้าง Sprint Test Cases แบบอัตโนมัติ
2. รองรับข้อมูลนำเข้า User Story ที่สร้างด้วย Excel

3. กรณีทดสอบและผลลัพธ์จากการทดสอบจะต้องอยู่ในรูปแบบตามที่กำหนดเท่านั้น
4. รองรับการทดสอบบนเรสท์ฟูลเว็บเซอร์วิส ที่ถูกพัฒนาด้วย JAVA โดยใช้ Spring Framework
5. Message Body อยู่ในรูปแบบของเจสัน (JSON) หรือเอกซ์เอ็มแอล (XML)

1.4 ขั้นตอนและวิธีดำเนินการวิจัย

1. ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการทดสอบเรสท์ฟูลเว็บเซอร์วิส
2. ศึกษาแนวทางการสร้าง JUnit Source Code และรูปแบบของข้อมูลที่จะใช้ในการทดสอบ รวมถึงข้อจำกัด
3. ศึกษาและทำความเข้าใจเกี่ยวกับการทดสอบเรสท์ฟูลเว็บเซอร์วิส
4. ศึกษาหลักการส่งผ่านข้อมูลในรูปแบบ JSON และ XML
5. วิเคราะห์และออกแบบระบบให้ตรงตามวัตถุประสงค์ของงานวิจัย
6. พัฒนาระบบและคอมโพเนนต์ต่างๆ รวมทั้งเครื่องมือส่งข้อมูลไปยังบริการที่พัฒนาด้วยเรสท์ฟูลเว็บเซอร์วิส
7. ทดสอบระบบและประเมินผลระบบที่พัฒนา
8. ตีพิมพ์ผลงานวิชาการ
9. สรุปผลการวิจัยและข้อเสนอแนะ และจัดทำวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้วิธีการและเครื่องมือสร้างกรณีทดสอบแบบอัตโนมัติจาก User Story ในการพัฒนาซอฟต์แวร์แบบขับเคลื่อนด้วยการทดสอบ
2. ได้เครื่องมือสำหรับทดสอบการบูรณาการกับบริการที่พัฒนาด้วยเรสท์ฟูลเว็บเซอร์วิสได้อย่างรวดเร็ว
3. วิธีการที่นำเสนอช่วยให้การพัฒนาเชื่อมต่อบริการต่างๆ เข้าด้วยกันเป็นไปได้ง่ายขึ้น
4. วิธีการที่นำเสนอช่วยลดความซ้ำซ้อน ในการทดสอบบริการที่พัฒนาด้วยเรสท์ฟูลเว็บเซอร์วิส
5. กรณีทดสอบที่สร้างขึ้นสามารถนำกลับมาใช้ได้ใหม่ เนื่องจากอยู่ในรูปแบบ Source ของ JUnit

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความวิจัยในหัวข้อเรื่อง “แนวทางการทดสอบเรสท์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอจิล” โดย สรिता คุปตยานนท์ และ ญาใจ ลิ้มปิยะกรณ์ ในวารสารการประชุมวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ครั้งที่ 10 (The 10th National Conference on Computing and Information

Technology: NCCIT 2014) ซึ่งจัดขึ้นโดยคณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ณ โรงแรมอัสสนา ลากูน่า ภูเก็ต ระหว่างวันที่ 8-9 พฤษภาคม 2557

1.7 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

วิทยานิพนธ์นี้แบ่งเนื้อหาออกเป็น 6 บท ดังต่อไปนี้ บทที่ 1 เป็นบทนำกล่าวถึงที่มาและความสำคัญของปัญหา วัตถุประสงค์ ขอบเขต และประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย บทที่ 2 กล่าวถึงกล่าวถึงทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้องกับงานวิจัยนี้ บทที่ 3 อธิบายถึงแนวทางในการสร้างกรณีทดสอบแบบบูรณาการจากแผนภาพส่วนประกอบ บทที่ 4 อธิบายถึงรายละเอียดการออกแบบของระบบต้นแบบ บทที่ 5 กล่าวถึงวิธีการทดสอบระบบ และบทที่ 6 กล่าวถึงสรุปผลการวิจัยและข้อเสนอแนะ

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 เรสท์ (REST : Representational State Transfer) [1]

เรสท์เป็นแบบสถาปัตยกรรม (Architecture style) ของระบบเน็ตเวิร์ก โดยเรียกใช้ผ่านทางโปรโตคอลเอชทีทีพี (HTTP) ซึ่งสามารถดำเนินการกับรับส่งข้อมูลผ่านเมทอด (Method) GET/POST/PUT/DELETE ดังสรุปลักษณะการทำงานของแต่ละเมทอดในตารางที่ 1 และส่งออกข้อมูลในรูปแบบต่างๆได้ ทำให้ปริมาณข้อมูลที่รับส่งน้อยกว่าการใช้โปรโตคอล SOAP อยู่มาก ซึ่งเป็นข้อดีของเรสท์นี้ ทำให้นักพัฒนาสนใจการเขียนโปรแกรมแบบใช้เรสท์กันมากขึ้น เพราะมีผลกับประสิทธิภาพของการใช้งานโปรแกรมด้วย แต่เนื่องจากเรื่องเรสท์นี้เพิ่งเกิดขึ้นมาเมื่อปี 2000 ทำให้ยังไม่มีมาตรฐานที่กำหนดให้บังคับใช้ รูปแบบการแสดงผล (Representation) ของทรัพยากรนั้นจะถูกนำมาวางลงบน Client application ในรูปแบบของสถานะ (State) รูปแบบการแสดงผลใหม่ที่ถูกวางลงบน Client application อีกครั้งก็จะถือว่าเป็นอีกสถานะหนึ่ง ดังนั้น Client application จะเปลี่ยน (Transfer) สถานะในแต่ละรูปแบบการแสดงผลของทรัพยากร จึงได้ชื่อว่า Representational State Transfer

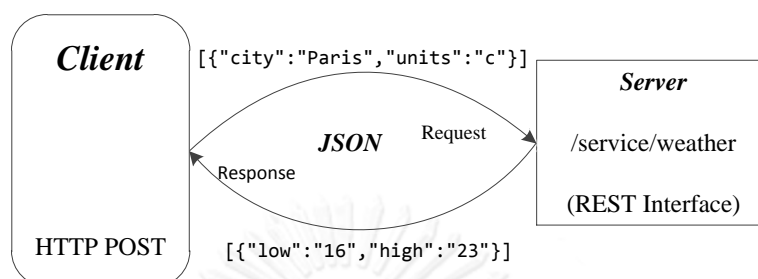
ตารางที่ 1 ลักษณะการทำงานของ REST Method

Method	ลักษณะการทำงาน	รูปแบบ URL
GET	เรียกข้อมูลมาแสดงแบบหลายรายการ	http://develop.com/platform /
POST	เรียกข้อมูลมาแสดงแบบทีละรายการ	http://develop.com/platform/12
PUT	ส่งข้อมูลจากฟอร์มหรือโปรแกรม เพื่อเพิ่มข้อมูล	http://develop.com/platform/
DELETE	ส่งข้อมูลจากฟอร์มหรือโปรแกรม เพื่อแก้ไขข้อมูล	http://develop.com/platform/12

2.1.2 เจสัน (JSON : JavaScript Object Notation) [2]

รูปแบบเจสัน คือ RFC 4627 มี Content-Type เป็น application/json มีนามสกุลของไฟล์เป็น .json และมี Internet media type เป็น application/json ซึ่งเป็นรูปแบบสำหรับแลกเปลี่ยนข้อมูลคอมพิวเตอร์ JSON นั้นย่อมาจากคำว่า "JavaScript Object Notation" ข้อมูลในรูปแบบของเจสันเป็นข้อความธรรมดา (Plain text) ที่ถูกห่อหุ้มด้วยเครื่องหมายปีกกา { } และถูกจัดเรียงให้อยู่ในรูปแบบ Object Array ปัจจุบัน เจสันเป็นรูปแบบข้อมูลที่นิยมใช้ในเว็บแอปพลิเคชัน เนื่องจากกระชับเข้าใจง่าย สาเหตุที่เจสันเป็นที่ได้รับความนิยมเพราะกระชับและเข้าใจง่าย โดยเจสันเป็นฟอร์แมตทางเลือกในการส่งข้อมูล นอกเหนือไปจากเอกซ์เอ็มแอล ซึ่งนิยมใช้กันอยู่แต่เดิม และมี

ไลบรารีของภาษาโปรแกรมอื่นๆ ที่ใช้ประมวลผลข้อมูลในรูปแบบเจสันมากมาย เจสันไม่ถูกมองว่าเป็นภาษาโปรแกรม แต่กลับถูกมองว่าเป็นภาษาในการแลกเปลี่ยนข้อมูลมากกว่า ดังรูปที่ 1 แสดงตัวอย่างรูปแบบเจสันที่ใช้ในการรับส่งข้อมูลบนเครือข่าย



รูปที่ 1 การรับส่งข้อมูลบนเครือข่ายในรูปแบบเจสัน

2.1.3 JUnit [3]

JUnit เป็นกรอบงานทดสอบ (Testing framework) พัฒนาโดย Kent Beck และ Erich Gamma มีจุดประสงค์เพื่อให้เป็นกรอบงานการทดสอบหน่วย (Unit Testing) สำหรับโปรแกรมภาษาจาวา ตัวอย่างโค้ด JUnit ดังรูปที่ 2 มีการสร้าง Response ที่ API enquiryBill ต้องการชื่อ enquiryBillRespM ที่มีข้อมูลที่คาดหวังคือ getBiller เรียก enquiryBill() แล้วเก็บใส่ตัวแปรชื่อ enquiryBillResponse นำเข้าตัวแปรทั้งคู่เข้ามา assertEquals(enquiryBillRespM.getBiller(), enquiryBillResponse.getBillerCode()) เพื่อทดสอบว่าผ่าน Unit test หรือไม่ ซึ่งถ้า Response ทั้งคู่มีค่าเท่ากัน Process การ Unit test ผ่านการทดสอบ แต่ถ้าไม่เท่ากันจะเกิด exception และหยุดการ Unit test ใน enquiryBillServicesSuccess()

JUnit จะมี assert แบบต่างๆที่มีอยู่ใน class org.junit.Assert ได้แก่

- assertEquals() เป็นเมทอดสำหรับทดสอบ Array 2 ตัวว่า “เท่ากัน” หรือไม่ โดยคำว่า “เท่ากัน” นี้หมายถึง Array 2 ตัวนี้ต้องมี “จำนวนสมาชิก” เท่ากัน และ “สมาชิกในแต่ละตำแหน่งต้องมีค่าเท่ากัน” ด้วย
- assertEquals() เป็นการทดสอบหน่วย ระหว่าง Object กับ Object โดยผ่านการเรียก equals()
- assertTrue() + assertFalse() เป็นเมทอดสำหรับทำการทดสอบหน่วยแบบใช้ตัวแปรเดียวว่ามีค่าเป็น true หรือ false
- assertNull() + assertNotNull() เป็นเมทอดสำหรับทำการทดสอบหน่วยแบบใช้ตัวแปรตัวเดียวอีกเช่นกัน โดยจะสนใจว่ามีค่าเป็น null หรือไม่
- assertEquals() + assertEquals() สำหรับเมทอดทดสอบว่าการอ้างถึงทั้งสองตัวนี้ระบุไปที่ Object ตัวเดียวกันหรือไม่ ซึ่งมันไม่เหมือนกับ assertEquals() ซึ่งเทียบกันด้วย equals() แต่เมทอดนี้ต้องเป็นตัวเดียวกันเท่านั้น

- `assertThat()` เป็นเมทอดที่ใช้เปรียบเทียบระหว่าง Object กับ `org.hamcrest.Matcher` โดยการทดสอบหน่วยจะผ่านเมื่อ Object นั้นต้อง Match กับ Matcher ที่เลือก

`assert` สามารถเรียกใช้งานได้ง่าย แต่ `assert` มีความเหมาะสมที่จะใช้ในงานวิจัยที่จะใช้ทดสอบการทำงานของเรสตร์ฟูลเว็บเซอร์วิสได้ และข้อดีของ JUnit มีความยืดหยุ่น สามารถปรับแก้หรือสามารถนำกลับมาใช้ซ้ำได้สำหรับบางกรณีทดสอบ ทำให้เป็นประโยชน์ที่จะสามารถนำ Script เดิมกลับมาใช้ได้อีกในอนาคต

```
@Test
public void enquiryBillServicesSuccess() throws
RemoteException {
    when(kioskProxyRequestFactory.billEnquiryRequest()).the
nReturn(enquiryBillReqM);
    when(proxy.billEnquiry(enquiryBillReqM)).thenReturn(enq
uiryBillRespM);
    enquiryBillResponse =
kioskProxyBillService.enquiryBill(enquiryBillRequest);
    assertEquals(enquiryBillRespM.getBiller(),enquiryBillRe
sponse.getBillerCode());
}
```

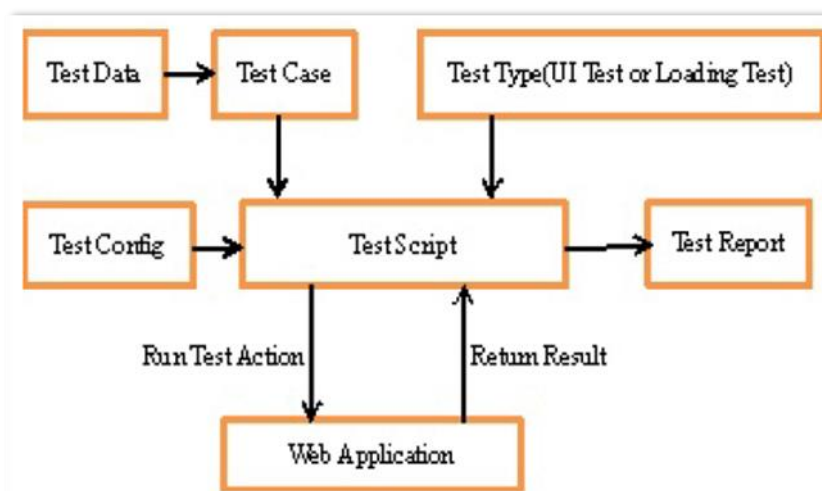
รูปที่ 2 ตัวอย่างสคริปต์ทดสอบในรูปแบบ JUnit source code

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 A Test Automation Framework Based on WEB [4]

การพัฒนาซอฟต์แวร์ให้เป็นเว็บแอปพลิเคชันได้รับความนิยม บทความนี้ได้กล่าวถึงการออกแบบกรอบงานทดสอบแบบอัตโนมัติสำหรับเว็บแอปพลิเคชัน เพื่อให้มีความสะดวกในการทดสอบฟังก์ชัน, การทดสอบประสิทธิภาพ, การนำกลับมาใช้ใหม่ และตรวจสอบความถูกต้อง โดยใช้เครื่องมือทดสอบ Selenium เขียนทดสอบ UI(User Interface)บนเว็บที่ไม่จำกัดภาษาที่ใช้พัฒนา และใช้เครื่องมือ Jmeter ในการสนับสนุนการร้องขอไปยังเว็บเซิร์ฟเวอร์หรือฐานข้อมูลเซิร์ฟเวอร์สำหรับทดสอบและสามารถทดสอบความประสิทธิภาพที่แตกต่างกันของเซิร์ฟเวอร์

การทดสอบอัตโนมัติควรมีไฟล์ที่ใช้ในการระบุค่า configuration เพียงไฟล์เดียว เพื่อที่หลีกเลี่ยงข้อมูลที่มีการเปลี่ยนแปลง เป็นอิสระต่อกันและป้องกันการรบกวนการทดสอบ โดยงานวิจัยได้ออกแบบกรอบงานทดสอบแบบอัตโนมัติ ดังรูปที่ 3 โดยสามารถนำมาประยุกต์ใช้ในงานวิจัยได้ โดยแต่ละส่วนประกอบเป็นอิสระต่อกันเพื่อรองรับเครื่องมือทดสอบอื่นๆในอนาคต โดยใช้กรอบงานทดสอบแบบอัตโนมัติสามารถกำหนดประเภทการทดสอบได้อย่างง่ายดายเพียงแค่ผ่านการปรับเปลี่ยนประเภทการทดสอบ และสามารถสลับไปมาระหว่างการทดสอบที่แตกต่างกันโดยไม่ต้องเปลี่ยนใด ๆ องค์ประกอบที่เกี่ยวข้องกับการทดสอบ



รูปที่ 3 กรอบงานการทดสอบอัตโนมัติบนพื้นฐานเว็บ [4]

2.2.2 Test-the-REST: An Approach to Testing RESTful Web Service [5]

งานวิจัยดังกล่าวได้นำเสนอวิธีการทดสอบเรสต์ฟูลเว็บเซอร์วิส ใช้ชื่อโครงการว่า Test-the-REST (TTR) โดยใช้รูปแบบของเอกซ์เอ็มแอลมาช่วยในการสร้าง ระบุรายละเอียดอื่นๆที่จำเป็นในการดำเนินการทดสอบ ดังรูปที่ 4 และตรวจสอบความถูกต้องของกรณีทดสอบ รวมถึงการนำผลลัพธ์ของการทดสอบมาแสดงผลด้วย

TTR specification language สามารถตรวจสอบถูกต้องของกรณีทดสอบ รองรับการตอบสนองด้วย เอกซ์เอ็มแอลหรือเจสันโดยสามารถระบุผลลัพธ์ ด้วยเงื่อนไข (" &", "|") ใน MediaTypeExpression ดังรูปที่ 5 และการกำหนดลำดับการทดสอบ ดังรูปที่ 6 การดำเนินการกรณีทดสอบ TTR-3 จะผ่านก็ต่อเมื่อทดสอบ TTR-1 และ TTR-2 ผ่านทั้งสองกรณี และแสดงผลลัพธ์กรณีทดสอบในเอกซ์เอ็มแอล ดังรูปที่ 7 แสดง "true"และ "false" ตามลำดับ

```

<testcase>
  <id>TTR-1</id>
  <URI>http://localhost/cgi-bin/hello-sujit.cgi</URI>
  <atomicity>atomic</atomicity>
  <method>GET</method>
  <representation mediatype="XML"/>
</testcase>
  
```

รูปที่ 4 กรณีทดสอบ [5]


```

<testcase>
  <id>TTR-COMPOSITE-VALIDATION</id>
  <URI>http://localhost/cgi-bin/hello-sujit.cgi</URI>
  <atomicity>atomic</atomicity>
  <method>GET</method>
  <response>
    <representation mediaTypeExpression="xml | plaintext"/>
    <mediaType id="xml" value="XML"/>
    <mediaType id="plaintext" value="PLAINTEXT"
      src="D:\ResultValidationData.txt"/>
  </response>
</testcase>

```

รูปที่ 5 การระบุปลายทางในการแสดงผลลัพธ์ [5]

```

<testcase>
  <id>TTR-3</id>
  <atomicity>composite</atomicity>
  <sequence>
    <testcase>
      <src>TTR-1.xml</src>
    </testcase>
    <testcase>
      <src>TTR-2.xml</src>
    </testcase>
  </sequence>
</testcase>

```

รูปที่ 6 การระบุลำดับกรณีทดสอบ [5]

```

<testcase>
  <id>TTR-6</id>
  <atomicity>composite</atomicity>
  <var>x</var>
  <var>y</var>
  <sequence>
    <testcase>
      <src>TTR-6-1.xml</src>
      <return>{x}</return>
      <return>{y}</return>
    </testcase>
  </sequence>
</testcase>

```

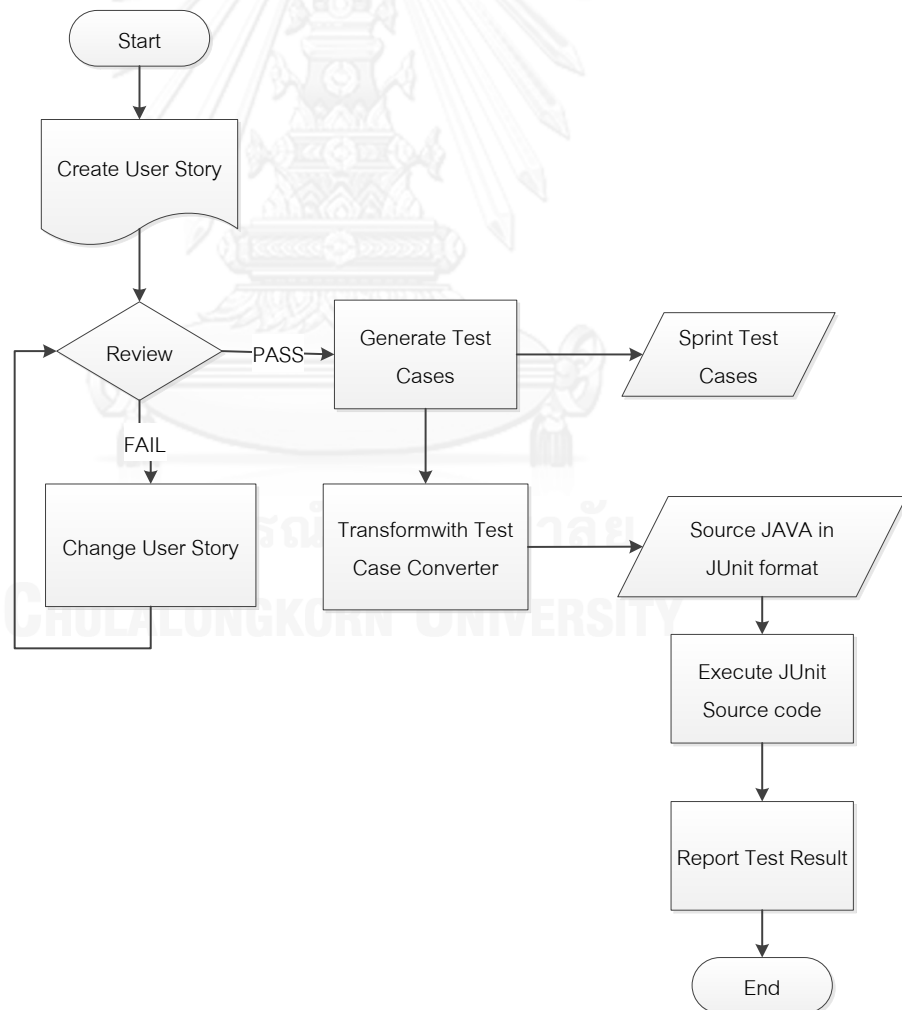
รูปที่ 7 ผลลัพธ์ของกรณีทดสอบ TTR-6-1.xml [5]

บทที่ 3

แนวคิดวิธีดำเนินการวิจัย

3.1 ภาพรวมของแนวทางทดสอบเรสต์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอจิล์

งานวิจัยนี้นำเสนอแนวทางในการทดสอบบูรณาการกับ Services ที่พัฒนาด้วยเรสต์ ในการพัฒนาซอฟต์แวร์แบบเอจิล์ ด้วยเทคนิคการพัฒนาขับเคลื่อนด้วยการทดสอบ ประกอบด้วยขั้นตอนต่างๆ ดังรูปที่ 8 การพัฒนาซอฟต์แวร์จะแบ่งรอบการพัฒนา เรียกว่า Sprint โดยแบ่งเป็น Sprint ละ 10 วัน โดยที่ความต้องการของซอฟต์แวร์จะถูกแบ่งออกเป็นส่วนย่อยๆ โดยความต้องการที่สำคัญ จะถูกเลือกมาทำใน Sprint แรกๆ โดยผู้ที่มีส่วนเกี่ยวข้องกับซอฟต์แวร์จะเป็นผู้จัดลำดับความสำคัญ โดยภาพรวมของงานมุ่งวิจัย จะมุ่งเน้นทดสอบบริการที่พัฒนาด้วยเรสต์ โดยที่ไม่ต้องเขียนโค้ด ทำได้โดยเริ่มต้นจากการสร้าง User Story ที่อยู่ในรูปแบบ Excel และสามารถทดสอบได้บ่อยเท่าที่ ต้องการก่อนที่จะเริ่มเขียนโค้ด หรือเริ่มเข้า Sprint เพื่อเป็นการลดระยะเวลาการทดสอบภายใน Sprint และเป็นการตรวจสอบการทำงานของบริการว่าเป็นไปตามผลลัพธ์ที่คาดหวัง



รูปที่ 8 ขั้นตอนวิธีการทำงานของระบบ

3.1.1 Create User Story

User story คือ การเขียนความต้องการของระบบ ให้อยู่ในรูปแบบประโยคที่เข้าใจได้ง่าย สามารถเป็นตัวกลางสื่อสารกันระหว่างทีมพัฒนาซอฟต์แวร์กับลูกค้าหรือผู้ใช้ซอฟต์แวร์ และนำไปสู่การยืนยันความต้องการที่ตรงกันของทั้งสองฝ่าย โดยในงานวิจัยจะออกแบบให้ User story อยู่ในรูปแบบ Excel และมีช่องกำหนดไว้สำหรับช่องที่ต้องการข้อมูลที่จำเป็นในการดำเนินการทดสอบ ต้องใส่ค่าทุกช่อง เพื่อให้ง่ายต่อการตรวจสอบและสามารถ Generate test case ได้ และรูปแบบ Excel นี้ได้ออกแบบให้เหมาะสมสำหรับความต้องการที่จะทดสอบเรสตฟูลเว็บเซอร์วิส ดังตารางที่ 2 โดยในขั้นตอนนี้จำเป็นต้องอาศัยประสบการณ์ของนักทดสอบ มาช่วยในการสร้าง User Story ด้วย เนื่องจากจะต้องมีเงื่อนไขที่ยอมรับ (Acceptance Criteria) ซึ่งประกอบด้วยกรณีทดสอบที่หลากหลายเงื่อนไขเพื่อง่ายต่อการนำไปใช้ในขั้นตอนถัดไป

คุณลักษณะต่างๆใน User story ประกอบด้วย

- Feature ชื่อคุณสมบัติของซอฟต์แวร์
- Description คำอธิบายโดยสังเขปของ User story
- TEST-CASE-ID-XXX รหัสกรณีทดสอบ
- Request Method ได้แก่ GET หรือ POST
- Request URL (Uniform Resource Locator) ที่อยู่ของเรสตฟูลเว็บเซอร์วิส
- Request Attribute ค่าที่จำเป็นต้องส่งไป
- Response Attribute ค่าที่ผลลัพธ์ที่ถูกตอบกลับ
- Name, Value ชื่อและค่าของข้อมูล
- Expected ข้อมูลที่คาดหวัง
- Assertion Method การเปรียบเทียบกับค่าที่จะเกิดขึ้นจริง ประกอบด้วย 9 ค่า ได้แก่
 - assertNull
 - assertNotNull
 - assertTrue
 - assertFalse
 - assertThat
 - assertEquals
 - assertEqualsOrMoreThan
 - assertEqualsOrLessThan
 - assertMoreThan

- assertNotEquals

ตารางที่ 2 รูปแบบการเขียน User story ที่นำเสนอในงานวิจัย

Feature	BillPayment		
Description	Bill payment service.		
TEST-CASE-ID-001	InquiryTrueMoveBillSuccess		
Request Method	GET		
Request URL	http://localhost:8080/counter/online/bill/inquiry/truemove/0809001044		
Request Attribute	Name	Value	
Response Attribute	Name	Assertion method	Expected
	biller_code	Equals	truemove
	account_no	Equals	0809001044
	account_status	NotNull	
	fullname	NotNull	
	total_outstanding	EqualsOrMoreThan	0
TEST-CASE-ID-002	PaymentSuccess		
Request Method	POST		
Request URL	http://localhost:8080/counter/online/bill/pay		
Request Attribute	Name	Value	
	biller_code	truemove	
	account_no	0809001044	
	amount	534.47	
Response Attribute	Name	Assertion method	Expected
	payment_id	NotNull	
	payment_date	NotNull	
	result_code	Equals	0

3.1.2 Review

ขั้นตอนนี้เป็น การตรวจทานความถูกต้องสมบูรณ์ของ User story ประกอบด้วย 2 ขั้นตอน

- 1) ตรวจทานโดยอาศัยความรู้ ความชำนาญเฉพาะด้านของนักทดสอบ
- 2) ตรวจทานความถูกต้องสมบูรณ์ของข้อมูล เช่น ค่าทุกช่องห้ามว่าง และประเภทข้อมูลของค่าถูกต้องเหมาะสม เป็นต้น เพื่อให้สามารถดำเนินการทดสอบในขั้นตอนต่อไปได้

3.1.3 Change User Story

ขั้นตอนนี้เป็น การให้ผู้ทำหน้าที่ทำหน้าที่มีส่วนเกี่ยวข้องกับซอฟต์แวร์ ได้แก่ นักพัฒนา นักทดสอบ นักออกแบบ ผู้เชี่ยวชาญด้านเทคโนโลยีสารสนเทศ และผู้จัดการโครงการ เข้ามาทำการปรับปรุงแก้ไข และเพิ่มเติมข้อมูลใน User Story ที่สร้างขึ้นให้ครบถ้วนสมบูรณ์ ภายหลังจากพบข้อผิดพลาดในขั้นตอนก่อนหน้า

3.1.4 Generate Test Cases

ภาษาเอกซ์เอ็มแอลจะใช้ Markup Tag เพื่ออธิบายความหมายของค่าข้อมูล โดย Tag คือ ข้อความที่อยู่ระหว่างเครื่องหมาย < และ > และค่าของข้อมูลจะต้องอยู่ระหว่าง Tag เปิดและ Tag ปิด โดย Tag ปิดจะมีเครื่องหมาย "/" นำหน้าข้อความ สามารถสรุปเป็นขั้นตอนได้ดังนี้

1. ทำการรับค่า ไฟล์ Excel และอ่านโดยใช้ Library Apache POI
2. อ่านค่าในแถวแต่ละแถว getworkbook(), getSheet(), getColumn(), cellliterator(), getStringCellValue() ตามลำดับ
3. ตรวจสอบค่าให้ถูกต้อง เหมาะสมที่จะดำเนินการทดสอบ
4. เก็บค่าลง Map เรียกใช้งานผ่านเมทอด getter ,setter
5. ใช้ DocumentBuilder class ส่งค่า Map ไปยังแฟ้มผลลัพธ์เอกซ์เอ็มแอล

จากงานวิจัยนี้ได้ออกแบบรูปแบบของเอกซ์เอ็มแอล ที่ได้จาก User Story ทั้งหมด 2 ไฟล์

1. ไฟล์แสดงข้อมูลที่จำเป็นในการทดสอบดังรูปที่ 9 และ 10 อธิบายความหมายของ XML Tag ดังแสดงสรุปในตารางที่ 3
2. ไฟล์ที่ระบุลำดับในการดำเนินการทดสอบดังรูปที่ 11 อธิบายความหมายของ XML Tag ดังแสดงสรุปในตารางที่ 4

ตารางที่ 3 ข้อมูลที่สกัดจาก User Story ในรูปแบบเอกซ์เอ็มแอล

XML Tag	คำอธิบาย
testcase	Root Element
id	รหัสกรณีทดสอบและชื่อไฟล์
name	ชื่อเมทอด
method	ชนิดของการร้อง และรูปแบบของข้อมูล
url	Resource URI
request	ค่าที่จำเป็นต้องส่งไปสามารถอธิบายข้อมูลเพิ่มเติมของ Element <code><request></code> ในรูปแบบของ Attribute เช่น Tag ชื่อ <code><param></code> คำอธิบายเพิ่มเติม ตัวอย่าง <code>name="biller_code" value="truemove"</code> name คือ ชื่อของข้อมูลที่ร้องขอ value คือ ค่าของข้อมูล
response	ผลลัพธ์ที่ตอบกลับมาสามารถอธิบายข้อมูลเพิ่มเติมของ Element <code><response></code> ในรูปแบบของ Attribute เช่น Tag ชื่อ <code>param</code> คำอธิบายเพิ่มเติม ตัวอย่าง <code>name="result_code" expected="0" assert_type="Equals"</code> name คือ ชื่อของข้อมูลที่ตอบกลับ expected คือ ผลลัพธ์ที่คาดหวัง assert_type คือ Assertion method

```

<?xml version="1.0" encoding="UTF-8"?>
<testcase>
  <id>TEST-CASE-ID-001</id>
  <name>InquiryTrueMoveBillSuccess</name>
  <method>GET</method>
  <url>http://localhost:8080/counter/online/bill/inquiry/truemove/0809001044</url>
  <response>
    <param name="biller_code" expected="truemove" assert_type="Equals"/>
    <param name="account_no" expected="0809001044" assert_type="Equals"/>
    <param name="account_status" assert_type="NotNull"/>
    <param name="fullname" assert_type="NotNull"/>
    <param name="total_outstanding" expected="0" assert_type="EqualsOrMoreThan"/>
  </response>
</testcase>

```

รูปที่ 9 กรณีทดสอบ TEST-CASE-ID-001.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<testcase>
  <id>TEST-CASE-ID-002</id>
  <name>PaymentSuccess</name>
  <method>POST</method>
  <url>http://localhost:8080/counter/online/bill/pay</url>
  <request>
    <param name="biller_code" value="truemove"/>
    <param name="account_no" value="0809001044"/>
    <param name="amount" value="534.47"/>
  </request>
  <response>
    <param name="payment_id" assert_type="NotNull"/>
    <param name="payment_date" assert_type="NotNull"/>
    <param name="result_code" expected="0" assert_type="Equals"/>
  </response>
</testcase>

```

รูปที่ 10 กรณีทดสอบ TEST-CASE-ID-002.xml

ตารางที่ 4 ข้อมูลที่สกัดจาก User Story เพื่อสร้างลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอล

XML Tag	คำอธิบาย
testsuite	Root Element
id	รหัสลำดับการทดสอบและชื่อคลาส JUnit source code
sequence	ลำดับการทดสอบ
testcase	กรณีทดสอบ
src	ชื่อไฟล์กรณีทดสอบ
result	ผลลัพธ์เมื่อดำเนินการทดสอบ จะแสดงหลังการทดสอบ

```
<?xml version="1.0" encoding="UTF-8"?>
<testsuite <id>BillPayment</id> <sequence>
  <testcase>
    <src>TEST-CASE-ID-001.xml</src>
  </testcase>
  <testcase>
    <src>TEST-CASE-ID-002.xml</src>
  </testcase>
  <testcase>
    <src>TEST-CASE-ID-003.xml</src>
  </testcase>
  <testcase>
    <src>TEST-CASE-ID-004.xml</src>
  </testcase>
  <testcase>
    <src>TEST-CASE-ID-005.xml</src>
  </testcase>
</sequence></testsuite>
```

รูปที่ 11 ลำดับการทดสอบ Sequence.xml

3.1.5 Sprint Test Cases

กรณีทดสอบทั้งหมดของรอบการพัฒนาอยู่ในรูปแบบเอกซ์เอ็มแอลที่สมบูรณ์เหมาะสมใช้ได้ในช่วงต้นถัดไป นักพัฒนาสร้างสคริปต์ทดสอบ อ้างอิงจาก User Story ในช่วงต้นก่อนหน้า ตามหลักการพัฒนาขับเคลื่อนด้วยการทดสอบ ที่สามารถสร้างกรณีทดสอบได้ก่อนการเขียนโค้ด ซึ่งมีข้อดี คือ นักพัฒนาได้สคริปต์ทดสอบก่อนการเริ่มต้นพัฒนา ทำให้ประหยัดเวลาในการพัฒนา ในงานวิจัยนี้ได้นำเสนอวิธีการและพัฒนาเครื่องมือในการสร้างกรณีทดสอบอัตโนมัติจาก User Story

3.1.6 Transform with Test Case Converter

กระบวนการนี้ทำการแปลงกรณีทดสอบในรูปแบบเอกซ์เอ็มแอลจากขั้นตอนก่อนหน้าให้เป็นสคริปต์ทดสอบที่อยู่ในรูปแบบ JUnit source code ตัวอย่างดังรูปที่ 12 ซึ่งสะดวกต่อการนำไปใช้งานผ่าน IDE Tools ต่างๆ โดยเครื่องมือเหล่านี้ส่วนมากจะสนับสนุน JUnit โดยในงานวิจัยนี้ได้มี JUnit Template ที่นำมาเป็นต้นแบบในการสร้างสคริปต์ทดสอบ

สคริปต์ทดสอบที่ได้สามารถนำกลับมาใช้ทดสอบได้บ่อยครั้งเท่าที่ต้องการ โดยเวลาทดสอบจะแยกทดสอบเฉพาะส่วน Request attribute และ Response attribute ของ API Name หนึ่งๆ เท่านั้น ทั้งนี้ สคริปต์ทดสอบจะมุ่งเน้นใช้ในการทดสอบความพร้อมของเรสท์ฟูลเว็บเซอร์วิส ก่อนที่เริ่มจะเข้าพัฒนาซอฟต์แวร์ ข้อดีคือ การแบ่งการทดสอบเป็นส่วนย่อยๆ ทำให้ปรับเปลี่ยนแก้ไขได้ง่าย หากซอฟต์แวร์มีความผิดพลาดหรือพัฒนาแล้วไม่เป็นไปตามความต้องการ และด้วยลักษณะการพัฒนาที่เน้นให้ซอฟต์แวร์ผ่านการทดสอบก่อน แล้วนำมาปรับเปลี่ยนภายหลัง ทำให้ลดระยะเวลาการพัฒนา สามารถพัฒนาส่วนต่อไปได้ ในขณะที่ส่วนแรกกำลังทำการปรับปรุงเซอร์วิสหรือซอฟต์แวร์

```
public void testInquiryTrueMoveBillSuccess() {
    Map<String, String> responseDatas = RESTClientUtils.callTo("GET",
        "http://localhost:8080/counter/online/bill/inquiry/truemove/0809001044");

    TestCaseAssertionUtils.assertEquals("truemove", responseDatas.get("biller_code")
    );
    TestCaseAssertionUtils.assertEquals("0809001044",
        responseDatas.get("account_no"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("account_status"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("fullname"));
    TestCaseAssertionUtils.assertEqualsOrMoreThan("0",
        responseDatas.get("total_outstanding"));
}
```

รูปที่ 12 สคริปต์ทดสอบ

3.1.7 Execute JUnit Source Code

ทำการประมวลผล JUnit Source Code เพื่อทดสอบเรสท์ฟูลเว็บเซอร์วิสที่ต้องการ กล่าวคือ ตรวจสอบความถูกต้องในการเรียกใช้แต่ละ API และตรวจสอบความถูกต้อง Assertion methods ตามข้อมูลที่ระบุจาก User Story การประมวลผลจะใช้ Maven ซึ่งเป็นเครื่องมือการจัดสร้างโครงสร้างของซอฟต์แวร์ และสามารถดาวน์โหลด Java Class Library ที่ต้องการสำหรับการประมวลผลมาให้ โดยผู้ใช้งานจะต้องกำหนดค่า JUnit ในไฟล์ pom.xml ก่อน รวมทั้งระบุเวอร์ชันที่ต้องการ ดังรูปที่ 13 JUnit Library นี้จะทำการประมวลผล JUnit ให้อยู่ในรูปแบบเจสัน ซึ่งมีขนาดข้อมูลเล็กและเหมาะสมสำหรับการทดสอบเรสท์ฟูลเว็บเซอร์วิส

```

<dependencies>
  [...]
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.8.1</version>
    <scope>test</scope>
  </dependency>
  [...]
</dependencies>

```

รูปที่ 13 กำหนดค่า Library

3.1.8 Report Test Result

ทำการเปรียบเทียบผลลัพธ์ที่ได้จากการประมวลผล JUnit Source Code กับค่า Response attribute ที่ระบุใน User story และสร้างค่าลง Sequence.xml ใน XML Tag Element <result>{result test}</result> มีค่าเป็น PASS หรือ FAIL ดังรูปที่ 14 เพื่อบอกการอ่านค่าและกลับแสดงผลลัพธ์ที่หน้าจอ โดยการอ่านค่าที่เอกซ์เอ็มแอล จะประมวลผลในขั้นตอนนี้

```

<?xml version="1.0" encoding="UTF-8"?>
<testsuite>
  <id>BillPayment</id>
  <sequence>
    <testcase>
      <src>TEST-CASE-ID-001.xml</src>
      <result>PASS</result>
    </testcase>
    <testcase>
      <src>TEST-CASE-ID-002.xml</src>
      <result>PASS</result>
    </testcase>
    <testcase>
      <src>TEST-CASE-ID-003.xml</src>
      <result>PASS</result>
    </testcase>
    <testcase>
      <src>TEST-CASE-ID-004.xml</src>
      <result>PASS</result>
    </testcase>
    <testcase>
      <src>TEST-CASE-ID-005.xml</src>
      <result>FAILED</result>
    </testcase>
  </sequence>
</testsuite>

```

รูปที่ 14 ผลลัพธ์ของการทดสอบใน Sequence.xml

บทที่ 4

การออกแบบและพัฒนาระบบ

รายละเอียดในบทนี้จะอธิบายการออกแบบและพัฒนาระบบต้นแบบเพื่อสนับสนุนแนวทางในการทดสอบเรสต์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอจิล์ที่ได้นำเสนอในบทที่ 3 โดยเนื้อหาประกอบด้วยข้อกำหนดเบื้องต้นของระบบ ความต้องการเชิงหน้าที่ แผนภาพยูสเคส คำอธิบายยูสเคส การออกแบบ และการพัฒนาระบบ

4.1 ข้อกำหนดเบื้องต้นของระบบ

4.1.1 ผู้ใช้งาน (User)

ควรเป็นผู้ที่ทำหน้าที่มีส่วนเกี่ยวข้องกับซอฟต์แวร์ นักพัฒนา นักทดสอบ นักออกแบบ ผู้เชี่ยวชาญด้านเทคโนโลยีสารสนเทศ และผู้จัดการโครงการ ซึ่งเป็นผู้ที่มีความรู้เกี่ยวกับภาพรวมของระบบ ทั้งในด้านข้อกำหนดความต้องการ การออกแบบ และการทดสอบ

4.1.2 ข้อมูลนำเข้า (Input)

User Story ได้มาจากผู้ที่ทำหน้าที่มีส่วนเกี่ยวข้องกับซอฟต์แวร์ ระบุข้อกำหนดความต้องการ อยู่ในรูปแบบ Excel ด้วยเครื่องมือไมโครซอฟท์ เอกซ์เซล 2010 รวมถึงรายละเอียดที่เพียงพอต่อการดำเนินการทดสอบ และค่าที่ระบบรองรับได้

4.1.3 ข้อมูลนำออก (Output)

- รายการกรณีทดสอบในรูปแบบของเอกซ์เอ็มแอล (XML)
- รายการลำดับทดสอบและผลลัพธ์ในรูปแบบของเอกซ์เอ็มแอล (XLS)
- สคริปต์ทดสอบในรูปแบบ JUnit

4.1.4 ข้อจำกัดของระบบ (Constraint)

ระบบสามารถรองรับการนำเข้าไฟล์ได้เพียงครั้งละหนึ่งไฟล์เท่านั้น และผลลัพธ์ที่ได้จากการทดสอบยังแสดงได้ไม่ละเอียด แสดงได้เพียง PASS หรือ FAILED แค่นั้น ในส่วนของการดักจับข้อผิดพลาดระหว่างการทดสอบยังคงมีความซับซ้อนอยู่มาก ไม่สามารถระบุถึงข้อผิดพลาดออกมาได้อย่างชัดเจน และรองรับการทดสอบเรสต์ฟูลเว็บเซอร์วิส แบบ GET หรือ POST เท่านั้น

4.2 ความต้องการเชิงหน้าที่ (Functional Requirements)

ระบบทดสอบเรสต์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอจิล์มีความต้องการเชิงหน้าที่ดังตาราง

ตารางที่ 5 ความต้องการเชิงหน้าที่

รหัสอ้างอิง	ชื่อ	คำอธิบายความต้องการเชิงหน้าที่
FREQ-01	นำเข้า User Story ในรูปแบบไฟล์เอกซ์เซล	ระบบสามารถนำเข้า User Story ในรูปแบบไฟล์เอกซ์เซล เวอร์ชัน 2010 ได้
FREQ-02	สกัดคำอธิบายข้อกำหนดความต้องการ	ระบบสามารถสกัดคำอธิบายข้อกำหนดความต้องการ และตรวจสอบค่าที่ได้จาก User Story ได้ โดยมีค่าที่กำหนดดังตารางที่ 6
FREQ-03	สร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล	ระบบสามารถสร้างกรณีทดสอบจากคำอธิบายข้อกำหนดความต้องการ เพื่อสร้างสคริปต์ทดสอบ
FREQ-04	สร้างลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอล	ระบบสามารถสร้างลำดับการทดสอบ เพื่อดำเนินการทดสอบ
FREQ-05	สร้างสคริปต์ทดสอบในรูปแบบ JUnit	ระบบสามารถสร้างสคริปต์ทดสอบจากกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล โดยใช้ JUnit Template ที่สร้างขึ้นในงานวิจัยนี้ เพื่อต้นแบบในการสร้างสคริปต์ทดสอบที่เหมาะสม
FREQ-06	ส่งคำร้องขอไปยังเรสท์ฟูลเว็บเซอร์วิสและสร้างผลลัพธ์	ระบบสามารถส่งคำร้องขอไปยังเรสท์ฟูลเว็บเซอร์วิสได้ เพื่อทดสอบการทำงานของเรสท์ฟูลเว็บเซอร์วิสและบันทึกผลลัพธ์การทดสอบในรูปแบบเอกซ์เอ็มแอลได้
FREQ-07	แสดงผลลัพธ์จากกรณีทดสอบ	แสดงผลลัพธ์จากกรณีทดสอบ จากลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอล

ตารางที่ 6 ค่าที่กำหนดใน User story

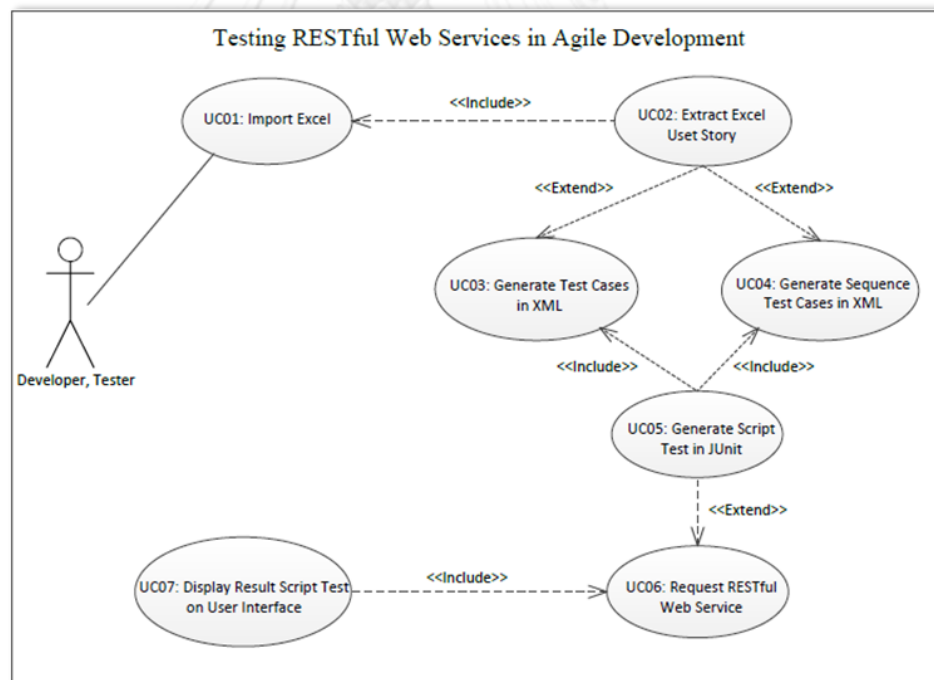
ข้อมูล	รายละเอียด
Feature	ห้ามเป็นค่าว่าง
TEST-CASE-ID-xxx	ห้ามเป็นค่าว่าง
Request Method	GET หรือ POST
Request URL	ห้ามเป็นค่าว่าง
Assertion method	Assertion มีดังนี้ <ul style="list-style-type: none"> - NotNull, - Null,

	<ul style="list-style-type: none"> - True, - False, - That, - Equals, - EqualsOrMoreThan, - EqualsOrLessThan, - MoreThan, - NotEquals
--	---

4.3 การออกแบบระบบ

4.3.1 แผนภาพยูสเคส

จากความต้องการเชิงหน้าที่สามารถแสดงเป็นแผนภาพยูสเคส (Use Case) ระบบทดสอบเรสต์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอจิลได้ดังรูปที่ 15



รูปที่ 15 แผนภาพยูสเคสของระบบทดสอบเรสต์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอจิล

4.3.2 คำอธิบายยูสเคส

อธิบายถึงฟังก์ชันความต้องการเชิงหน้าที่ ว่าผู้ใช้งานจะเกี่ยวข้องกับอย่างไรกับระบบ และระบบจะตอบสนองผู้ใช้งานอย่างไร และระบบทำงานมีขั้นตอนอย่างไร โดยคำอธิบายยูสเคสแสดงดังตารางที่ 7 ถึง ตารางที่ 13

ตารางที่ 7 คำอธิบายยูสเคส Import Excel

หมายเลขยูสเคส:	UC01	เวอร์ชัน:	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (High)		
ผู้เกี่ยวข้อง:	นักพัฒนา หรือนักทดสอบ		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการนำเข้า User Story ในรูปแบบเอกซ์เซล		
เงื่อนไขขั้นต้น:	User Story ต้องอยู่ในรูปแบบเอกซ์เซล ด้วยเครื่องมือไมโครซอฟท์ เอกซ์เซล 2010 และมีรายละเอียดที่เหมาะสมที่จะการดำเนินการทดสอบ		
การทำงานโดยปกติ:	<p>ผู้ใช้งานคลิกปุ่ม เลือกไฟล์</p> <p>เลือกไฟล์เอกซ์เซลที่จะนำเข้า</p> <p>ระบบตรวจสอบไฟล์และข้อมูล</p> <p>ระบบจัดเก็บไฟล์</p>		
ทางเลือกเพิ่มเติมในการทำงาน:	<p>3-1 นำเข้าไฟล์ไม่ถูกต้อง</p> <p>3-2 ระบบแสดงข้อความเตือนนำเข้าไฟล์ไม่ถูกต้อง</p> <p>3-3 กลับไปทำขั้นตอนที่ 2</p>		

ตารางที่ 8 คำอธิบายยูสเคส Extract Excel User Story

หมายเลขยูสเคส:	UC02	เวอร์ชัน:	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (High)		
ผู้เกี่ยวข้อง:	-		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการสกัดคำอธิบายข้อกำหนดความต้องการ		
เงื่อนไขขั้นต้น:	นำเข้า User Story ในรูปแบบเอกซ์เซล		
การทำงานโดยปกติ:	ระบบอ่านไฟล์เอกซ์เซลแล้วสกัดข้อมูลที่ต้องการ และตรวจสอบข้อมูลที่เหมาะสมต่อการทดสอบ		
ทางเลือกเพิ่มเติมในการทำงาน:	<p>1-1 ข้อมูลไม่ถูกต้อง</p> <p>1-2 ระบบแสดงข้อความเตือนนำข้อมูลที่ไม่ถูกต้อง</p> <p>1-3 กลับไปทำ UC01 ขั้นตอนที่ 2</p>		

ตารางที่ 9 คำอธิบายยูสเคส Generate Test Cases in XML

หมายเลขยูสเคส:	UC03	เวอร์ชัน:	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (Moderate)		
ผู้เกี่ยวข้อง:	-		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล		
เงื่อนไขขั้นต้น:	ระบบจะต้องสกัดคำอธิบายข้อกำหนดความต้องการ		
การทำงานโดยปกติ:	ระบบดึงข้อมูลคำอธิบายข้อกำหนดความต้องการ ระบบสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล ระบบจัดเก็บไฟล์		
ทางเลือกเพิ่มเติมใน การทำงาน:	-		

ตารางที่ 10 คำอธิบายยูสเคส Generate Sequence Test Cases in XML

หมายเลขยูสเคส:	UC04	เวอร์ชัน:	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (Moderate)		
ผู้เกี่ยวข้อง:	-		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการสร้างลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอล		
เงื่อนไขขั้นต้น:	ระบบจะต้องสกัดลำดับการทดสอบจากคำอธิบายข้อกำหนดความต้องการ		
การทำงานโดยปกติ:	ระบบดึงลำดับการทดสอบจากคำอธิบายข้อกำหนดความต้องการ ระบบสร้างลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอล ระบบจัดเก็บไฟล์		
ทางเลือกเพิ่มเติมใน การทำงาน:	-		

ตารางที่ 11 คำอธิบายยูสเคส Generate Script Test in JUnit

หมายเลขยูสเคส:	UC05	เวอร์ชัน:	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (High)		
ผู้เกี่ยวข้อง:	-		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการสร้างสคริปต์ทดสอบ		
เงื่อนไขขั้นต้น:	ระบบจะต้องสร้างกรณีทดสอบและลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอล		
การทำงานโดยปกติ:	<p>ระบบดึงข้อมูลจากกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล</p> <p>ระบบดึงข้อมูลจากลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอล</p> <p>ระบบสร้างกรณีสคริปต์ทดสอบจาก JUnit Template</p> <p>ระบบจัดเก็บไฟล์</p>		
ทางเลือกเพิ่มเติมในการทำงาน:	-		

ตารางที่ 12 คำอธิบายยูสเคส Request Restful Web Service

หมายเลขยูสเคส:	UC06	เวอร์ชัน:	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (High)		
ผู้เกี่ยวข้อง:	-		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการส่งคำร้องขอไปยังเรสท์ฟูลเว็บเซอร์วิสเพื่อทดสอบ		
เงื่อนไขขั้นต้น:	ระบบต้องสร้างสคริปต์ทดสอบ		
การทำงานโดยปกติ:	<p>ระบบอ่านข้อมูลกรณีสคริปต์ทดสอบที่สร้างขึ้น แล้วส่งคำร้องขอไปยังเรสท์ฟูลเว็บเซอร์วิส</p> <p>ระบบบันทึกผลลัพธ์ของการทดสอบ ในไฟล์ลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอล</p>		
ทางเลือกเพิ่มเติมในการทำงาน:	-		

ตารางที่ 13 คำอธิบายยูสเคส Display Result Script Test on User Interface

หมายเลขยูสเคส:	UC07	เวอร์ชัน:	1.0
ความสำคัญ:	ส่วนการทำงานหลัก (Moderate)		
ผู้เกี่ยวข้อง:	-		
รายละเอียดโดยย่อ:	ยูสเคสนี้อธิบายการแสดงผลการทดสอบ		
เงื่อนไขขั้นต้น:	ระบบต้องสร้างผลลัพธ์ของการทดสอบ ในไฟล์ลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอล		
การทำงานโดยปกติ:	ระบบอ่านข้อมูลผลลัพธ์ในไฟล์ลำดับการทดสอบที่อยู่ในรูปแบบเอกซ์เอ็มแอล ระบบแสดงผลการดำเนินการทดสอบ		
ทางเลือกเพิ่มเติมในการทำงาน:	-		

4.4 การพัฒนาระบบ

4.4.1 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา

สภาพแวดล้อมที่ใช้ในการพัฒนาระบบมีสภาพแวดล้อมทางด้านฮาร์ดแวร์และซอฟต์แวร์ดังต่อไปนี้

- ฮาร์ดแวร์
 - หน่วยประมวลผลกลาง (CPU) อินเทล คอร์ไอสาม 2.5 กิกะเฮิร์ตซ์ (CPU Intel Core i3 2.5GHz)
 - หน่วยความจำ (RAM) 2 DDR3 SDRAM กิกะไบต์ (1600 GHz)
 - จานบันทึกข้อมูล (Hard Disk) 320 กิกะไบต์ (Hard disk 320 GB)
- ซอฟต์แวร์
 - ระบบปฏิบัติการ ไมโครซอฟต์วินโดวส์เซเว่น โพรเฟสชันแนล 32 บิต (Microsoft Windows 7 Professional 32 bit)
 - เจดีเค 1.6.0_24 (Java JDK 1.6.0_24)
 - โปรแกรม สปริง ทูล สวีท เวอร์ชัน 3.3.0 (Spring Tool Suite 3.3.0)
 - ไมโครซอฟท์ เอกซ์เซล 2010 (Microsoft excel 2010)
 - อะปาเช่ มาเว่น เวอร์ชัน 3.2.2 (Apache Maven 3.2.2)

- อะปาเช่ ทอมแคท เวอร์ชัน 6.0.41 (Apache Tomcat 6.0.41)
- กิจ เวอร์ชัน 2.0.1 (GIT 2.0.1)
- กูเกิลโครม เว็บเบราว์เซอร์ 35.0.1916.153 m (Google Chrome 35.0.1916.153 m)

4.4.2 การติดตั้งซอฟต์แวร์ในการพัฒนาระบบ

ทำการติดตั้งเครื่องมือในการพัฒนาระบบทั้งหมดลงในเครื่องคอมพิวเตอร์ที่ใช้พัฒนาระบบ โดยมีลำดับการติดตั้งเครื่องมือเป็นไปตามขั้นตอนต่อไปนี้

1. ติดตั้งระบบปฏิบัติการ ไมโครซอฟต์วินโดวส์ต้า โสมพีเอ็มเอ็ม
2. ติดตั้งเจดีเค 1.6.0_24
3. ติดตั้งสปริง ทูล สวีท เวอร์ชัน 3.2.2
4. ติดตั้งไมโครซอฟท์ เอกซ์เซล 2010
5. ติดตั้งอะปาเช่ มาเว่น เวอร์ชัน 3.2.2
6. ติดตั้งอะปาเช่ ทอมแคท เวอร์ชัน 6.0.41
7. ติดตั้งกิจ เวอร์ชัน 2.0.1
8. ติดตั้งกูเกิลโครม เวอร์ชัน 35.0.1916.153 m

4.4.3 การพัฒนาส่วนต่อประสาน

พัฒนาส่วนต่อประสานผู้ใช้ ให้สอดคล้องกับขอบเขตของระบบจากบทที่ 1 และมีการทำงานตรงกับตามความต้องการด้านหน้าที่ที่กำหนดไว้ในบทที่ 4 ในการพัฒนาส่วนต่อประสานจะใช้ แอ ลก ลู ล่า เจเอส (AngularJS JavaScript Framework) โดยใช้โปรแกรมสปริง ทูล สวีท (Spring Tool Suite 3.3.0) ในการพัฒนา และใช้ติดตั้งอะปาเช่ ทอมแคท (Apache Tomcat 6.0.41) เป็นเซิร์ฟเวอร์ แสดงผลบน กูเกิลโครม เว็บเบราว์เซอร์ (Google Chrome 35.0.1916.153 m)

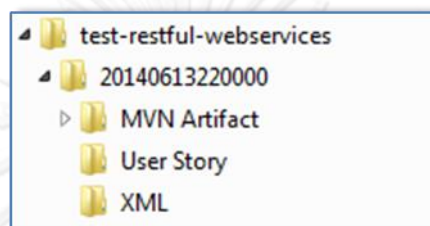
4.4.3.1 ส่วนของการนำเข้าไฟล์และแสดงผล

เมื่อเปิดระบบทดสอบเรสต์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอจิล์ ระบบจะแสดงหน้าจอหลัก ดังรูปที่ 16 ซึ่งเป็นส่วนหน้าจอทำงานหลัก สำหรับนำเข้า User Story ในรูปแบบเอกซ์เซลและแสดงผลการดำเนินการทดสอบ สำหรับไฟล์นำเข้าจะแสดงในแฟ้มเอกสาร ดังรูปที่ 17 แสดงที่ D:\test-restful-webservices\{\YYYYMMDDHHMMSS}\User Story

The screenshot shows a web browser window with the URL 'localhost:8080/test-upload-api/'. The page title is 'Testing RESTful Web Services'. Below the title, there is a 'User Story' input field, a 'เลือกไฟล์' (Select File) button, a 'ไม่ได้เลือกไฟล์' (File not selected) label, and a 'Cancel Upload' button. Below this is a section titled 'ผลลัพธ์การทดสอบ' (Test Results) containing a table with the following data:

ID	Name	Method	URL	Result
TEST-CASE-ID-001	InquiryTrueMoveBillSuccess	GET	http://localhost:8080/counter/online/bill/inquiry/truemove/0809001044	PASS
TEST-CASE-ID-002	PaymentSuccess	POST	http://localhost:8080/counter/online/bill/pay	PASS
TEST-CASE-ID-003	InquiryPaymentSuccess	GET	http://localhost:8080/counter/online/bill/pay/20140613212411	PASS
TEST-CASE-ID-004	CancelPaymentSuccess	POST	http://localhost:8080/counter/online/bill/pay/20140613212411/cancel	PASS
TEST-CASE-ID-005	InquiryAISBillSuccess	GET	http://localhost:8080/counter/online/bill/inquiry/ais/0809001044	FAILED

รูปที่ 16 หน้าจอของระบบทดสอบเรสทูลเว็บเซอร์วิสในการพัฒนาแบบเอจิล์



รูปที่ 17 รายการเพิ่มเอกสาร User Story

4.4.3.2 ส่วนของรายการไฟล์เอกสารเอกซ์เอ็มแอล

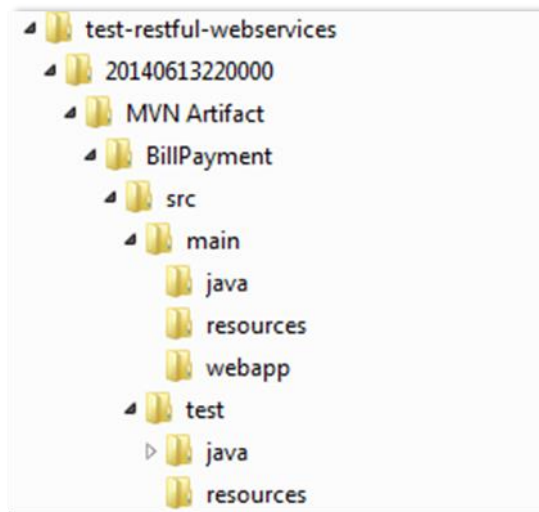
เป็นส่วนที่แสดงรายการของไฟล์เอกสารเอกซ์เอ็มแอลที่ส่งออกทั้งหมด โดยจะแสดงในเพิ่มเอกสารที่ D:\test-restful-webservices\{YYYYMMDDHHMMSS}\XML ซึ่งชื่อของเอกสารเอกซ์เอ็มแอลจะมีชื่อตาม TEST-CASE-ID-xxx และ Sequence.xml ด้วยทุกครั้ง ดังรูปที่ 18

Name	Date modified	Type	Size
Sequence	13/7/2557 22:19	XML Document	1 KB
TEST-CASE-ID-001	14/7/2557 1:18	XML Document	1 KB
TEST-CASE-ID-002	13/7/2557 22:08	XML Document	1 KB
TEST-CASE-ID-003	13/7/2557 22:08	XML Document	1 KB
TEST-CASE-ID-004	13/7/2557 22:09	XML Document	1 KB
TEST-CASE-ID-005	13/7/2557 22:09	XML Document	1 KB

รูปที่ 18 รายการเพิ่มเอกสารเอกซ์เอ็มแอล

4.4.3.3 ส่วนของรายการสคริปต์ทดสอบ

เป็นส่วนที่แสดงรายการของสคริปต์ทดสอบ โดยอ่านจากไฟล์กรณีทดสอบและลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอลและส่งออกในรูปแบบ JUnit Source code โดยจะแสดงในเพิ่มเอกสารที่ D:\test-restful-webservices\{YYYYMMDDHHMMSS}\MVN Artifact ซึ่งส่งออกตามโครงสร้างของ Maven project เพื่อที่จะสามารถนำไปประมวลผลต่อได้ ดังรูปที่ 19



รูปที่ 19 รายการเพิ่มเอกสารเอกซ์เอ็มแอล

สคริปต์ทดสอบในรูปแบบ Junit source code โดยชื่อคลาสสคริปต์ทดสอบจะระบุมาจาก User Story ในช่อง Feature และชื่อของเมทอดมาจากชื่อของกรณีทดสอบ จะส่งออกในแฟ้มเอกสารที่ D:\test-restful-webservices\20140613220000\MVN Artifact\BillPayment\src\test\java\th\ac\cu\testsuite โดยในงานวิจัยนี้ได้สร้าง JUnit Template เพื่อที่จะนำมาเป็นต้นแบบในการสร้างสคริปต์ทดสอบ ดังรูปที่ 20 และ Call REST Template ดังรูปที่ 21

```

package th.ac.cu.util;

import java.math.BigDecimal;
import org.hamcrest.Matcher;

public class TestCaseAssertionUtils {

    public static void assertEquals(String expected, String actual) {
        org.junit.Assert.assertEquals(expected, actual);
    }

    public static void assertNotNull(String object) {
        org.junit.Assert.assertNotNull(object);
    }

    public static void assertNull(String object) {
        org.junit.Assert.assertNull(object);
    }

    public static void assertTrue(boolean condition) {

```

```

        org.junit.Assert.assertTrue(condition);
    }
    public static void assertFalse(boolean condition) {
        org.junit.Assert.assertFalse(condition);
    }
    public static void assertThat(String expected, String actual) {
        org.junit.Assert.assertThat(expected, is(equalTo(actual)));
    }
    public static void assertEquals(String expected, String actual) {
        org.junit.Assert.assertEquals(expected, actual);
    }
    public static void assertEqualsOrMoreThan(String expected, String actual) {
        org.junit.Assert.assertTrue(new BigDecimal(actual).compareTo(new
BigDecimal(expected))>=0);
    }

    public static void EqualsOrLessThan(String expected, String actual) {
        org.junit.Assert.assertTrue(new BigDecimal(actual).compareTo(new
BigDecimal(expected))>=0);
    }
    public static void assertMoreThan(String expected, String actual) {
        org.junit.Assert.assertTrue(new BigDecimal(actual).compareTo(new
BigDecimal(expected))>=0);
    }
    public static void assertNotEquals(String expected, String actual) {
        org.junit.Assert.assertNotEquals(actual, expected);
    }
}

```

รูปที่ 20 ต้นแบบสำหรับสร้าง JUnit source code

```

package th.ac.cu.util;

import java.util.HashMap;
import java.util.Map;
import net.minidev.json.JSONObject;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.sun.jersey.api.client.Client;
import com.sun.jersey.api.client.ClientResponse;

public class RESTClientUtils {

    public static Map<String, String> callTo(String method, String url) {
        try {
            Client client = Client.create();
            client.setConnectTimeout(60000);
            client.setReadTimeout(60000);
            ClientResponse clientResponse =
client.resource(url).type("application/json").get(ClientResponse.class);
            return new
ObjectMapper().readValue(clientResponse.getEntity(String.class), HashMap.class);
        } catch (Exception e) {
            throw new Error(e);
        }
    }

    public static Map<String, String> callTo(String method, String url, Map<String, String>
requestDatas) {
        try {
            Client client = Client.create();
            client.setConnectTimeout(60000);
            client.setReadTimeout(60000);
            JSONObject json = new JSONObject();
            json.putAll(requestDatas);
            ClientResponse clientResponse =
client.resource(url).type("application/json").post(ClientResponse.class, json.toJSONString());
            return new
ObjectMapper().readValue(clientResponse.getEntity(String.class), HashMap.class);
        } catch (Exception e) {
            throw new Error(e);
        }
    }

    public static Map<String, String> callTo(String method, String url, Map<String, String>
requestDatas) {
        try {
            Client client = Client.create();
            client.setConnectTimeout(60000);
            client.setReadTimeout(60000);
            JSONObject json = new JSONObject();
            json.putAll(requestDatas);
            ClientResponse clientResponse =
client.resource(url).type("application/json").put(ClientResponse.class, json.toJSONString());
            return new
ObjectMapper().readValue(clientResponse.getEntity(String.class), HashMap.class);
        } catch (Exception e) {
            throw new Error(e);
        }
    }
}

```

```
    }  
  
    public static Map<String, String> callTo(String method, String url, Map<String, String>  
requestDatas) {  
        try {  
            Client client = Client.create();  
            client.setConnectTimeout(60000);  
            client.setReadTimeout(60000);  
            JSONObject json = new JSONObject();  
            json.putAll(requestDatas);  
            ClientResponse clientResponse =  
client.resource(url).type("application/json").delete(ClientResponse.class, json.toJSONString());  
            return new  
ObjectMapper().readValue(clientResponse.getEntity(String.class), HashMap.class);  
        } catch (Exception e) {  
            throw new Error(e);  
        }  
    }  
}
```

รูปที่ 21 ต้นแบบสำหรับเรียกการทดสอบเรสท์ฟูลเว็บเซอร์วิส

บทที่ 5

การทดสอบและประเมินระบบ

รายละเอียดในบทนี้จะอธิบายการทดสอบและประเมินระบบต้นแบบเพื่อสนับสนุนแนวทางในการทดสอบเรสต์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอจิล์ตามที่ได้ออกแบบและพัฒนาในบทที่ 4 โดยเนื้อหาประกอบด้วย การทดสอบระบบ การประเมินระบบผลระบบ ตั้งแต่ นำเข้าเอกสาร User Story ตลอดจนผลลัพธ์ของเอกสารไฟล์นำออก รวมถึงการนำสคริปต์ทดสอบไปใช้อีกด้วย

5.1 การทดสอบระบบ

การทดสอบระบบเรสต์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอจิล์ได้ออกแบบวิธีการทดสอบแบบ (Black Box Testing) ซึ่งจะทดสอบตามความต้องการเชิงหน้าที่ และมุ่งเน้นทดสอบฟังก์ชันการทำงาน รวมถึงเอกสารนำออก ซึ่งมีกรณีทดสอบและผลการทดสอบ ดังต่อไปนี้

1. ทดสอบการนำเข้าไฟล์ User story ในรูปแบบเอกซ์เซล 2010
 - 1.1) ทดสอบนำเข้า User story empty
 - 1.2) ทดสอบค่า Request Method ไม่ได้ระบุเป็น GET, POST, PUT และ DELETE
 - 1.3) ทดสอบค่า Request URL เป็นว่าง
 - 1.4) ทดสอบค่า Assertion method ไม่ได้เป็นค่าตามที่กำหนด
 - 1.5) ทดสอบค่า Assertion method เป็น NotNull
 - 1.6) ทดสอบค่า Assertion method เป็น Null
 - 1.7) ทดสอบค่า Assertion method เป็น True
 - 1.8) ทดสอบค่า Assertion method เป็น False
 - 1.9) ทดสอบค่า Assertion method เป็น That
 - 1.10) ทดสอบค่า Assertion method เป็น Equals
 - 1.11) ทดสอบค่า Assertion method เป็น EqualsOrMoreThan
 - 1.12) ทดสอบค่า Assertion method เป็น EqualsOrLessThan
 - 1.13) ทดสอบค่า Assertion method เป็น MoreThan
 - 1.14) ทดสอบค่า Assertion method เป็น NotEquals
2. ทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล
 - 2.1) ทดสอบจำนวนกรณีทดสอบ
 - 2.2) ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Request Method แบบ GET

- 2.3) ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Request Method แบบ POST
 - 2.4) ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Request Method แบบ PUT
 - 2.5) ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Request Method แบบ DELETE
 - 2.6) ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น NotNull
 - 2.7) ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น Null
 - 2.8) ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น True
 - 2.9) ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น False
 - 2.10) ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น That
 - 2.11) ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น Equals
 - 2.12) ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น EqualsOrMoreThan
 - 2.13) ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น EqualsOrLessThan
 - 2.14) ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น MoreThan
 - 2.15) ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น NotEquals
3. ทดสอบการสร้างลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอล
 - 3.1) ทดสอบความถูกต้องของลำดับการทดสอบ
 4. ทดสอบการสร้างสคริปต์ทดสอบ ในรูปแบบ JUnit source code
 - 4.1) ทดสอบการทำงานสคริปต์ทดสอบ Request Method แบบ GET
 - 4.2) ทดสอบการทำงานสคริปต์ทดสอบ Request Method แบบ POST
 - 4.3) ทดสอบการทำงานสคริปต์ทดสอบ Request Method แบบ PUT

- 4.4) ทดสอบการทำงานสคริปต์ทดสอบ Request Method แบบ DELETE
- 4.5) ทดสอบความถูกต้องของสคริปต์ทดสอบ โดย Assertion method เป็น NotNull
- 4.6) ทดสอบความถูกต้องของสคริปต์ทดสอบ โดย Assertion method เป็น Null
- 4.7) ทดสอบความถูกต้องของสคริปต์ทดสอบ โดย Assertion method เป็น True
- 4.8) ทดสอบความถูกต้องของสคริปต์ทดสอบ โดย Assertion method เป็น False
- 4.9) ทดสอบความถูกต้องของสคริปต์ทดสอบ โดย Assertion method เป็น That
- 4.10) ทดสอบความถูกต้องของสคริปต์ทดสอบ โดย Assertion method เป็น Equals
- 4.11) ทดสอบความถูกต้องของสคริปต์ทดสอบ โดย Assertion method เป็น EqualsOrMoreThan
- 4.12) ทดสอบความถูกต้องของสคริปต์ทดสอบ โดย Assertion method เป็น EqualsOrLessThan
- 4.13) ทดสอบความถูกต้องของสคริปต์ทดสอบ โดย Assertion method เป็น MoreThan
- 4.14) ทดสอบความถูกต้องของสคริปต์ทดสอบ โดย Assertion method เป็น NotEquals
5. ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิสที่ต้องการจะทดสอบและสร้างผลลัพธ์
- 5.1) ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Request Method แบบ GET และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ
- 5.2) ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Request Method แบบ POST และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ
- 5.3) ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Request Method แบบ PUT และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ
- 5.4) ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Request Method แบบ DELETE และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ
- 5.5) ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น NotNull และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ
- 5.6) ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น Null และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ

5.7) ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น True และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ

5.8) ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น False และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ

5.9) ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น That และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ

5.10) ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น Equals และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ

5.11) ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น EqualsOrMoreThan และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ

5.12) ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น EqualsOrLessThan และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ

5.13) ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น MoreThan และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ

5.14) ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น NotEquals และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ

6. ทดสอบการแสดงผลที่หน้าจการทำงานหลักจากไฟล์ลำดับการทดสอบ

ตารางที่ 14 การทดสอบการการนำเข้าไฟล์ User story ในรูปแบบเอกซ์เซล 2010

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC01	การนำเข้าไฟล์เอกสาร User story	นำเข้าไฟล์ User story ในรูปแบบเอกซ์เซล 2010	แสดงชื่อไฟล์เอกสาร User story ที่หน้าจอ	ถูกต้อง
			จัดเก็บเอกสารที่แฟ้มปลายทางได้ถูกต้อง	ถูกต้อง
			สามารถเปิดไฟล์เอกสารได้ถูกต้อง	ถูกต้อง

ตารางที่ 15 ทดสอบการนำเข้า User story empty

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC02	การนำเข้าไฟล์เอกสาร User story ในรูปแบบเอกซ์เซล 2010	เพื่อทดสอบการนำเข้า User story ในรูปแบบเอกซ์เซล 2010 โดยเนื้อหาว่าง	หน้าจอการทำงานหลักแสดงข้อความ “ขอภัย เอกสารนำเข้าของท่านไม่ถูกต้อง”	ถูกต้อง
			ระบบสามารถทำงานต่อได้	ถูกต้อง

ตารางที่ 16 ทดสอบค่า Request Method ไม่ได้ระบุเป็น GET หรือ POST

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC03	เอกสาร User story ในช่อง Request Method ต้องเป็นค่าตามที่กำหนดเท่านั้น	เพื่อตรวจสอบค่าในช่อง Request Method ที่ไม่ได้ระบุ GET หรือ POST ที่ระบบรองรับได้เท่านั้น	หน้าจอการทำงานหลักแสดงข้อความ “ขอภัย เอกสารนำเข้าของท่านไม่ถูกต้อง”	ถูกต้อง

ตารางที่ 17 ทดสอบค่า Request URL เป็นว่าง

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC04	เอกสาร User story ในช่อง Request URL ต้องเป็นค่าตามที่กำหนดเท่านั้น	เพื่อตรวจสอบค่าในช่อง Request URL เป็นค่าว่าง ระบบไม่สามารถรองรับได้	หน้าจอการทำงานหลักแสดงข้อความ “ขอภัย เอกสารนำเข้าของท่านไม่ถูกต้อง”	ถูกต้อง

ตารางที่ 18 ทดสอบค่า Assertion method ไม่ได้เป็นค่าตามที่กำหนด

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC05	เอกสาร User story ในช่อง Assertion method ต้องเป็นค่าตามที่กำหนดเท่านั้น	เพื่อตรวจสอบค่าในช่อง Assertion method ว่าเป็นค่าสามารถรองรับได้เท่านั้น ได้แก่ NotNull, Null True False NotEquals Equals EqualsOrMoreThan EqualsOrLessThan MoreThan LessThan	หน้าจอการทำงานหลัก แสดงข้อความ “ขอภัย เอกสารนำเข้าของท่านไม่ถูกต้อง”	ถูกต้อง

ตารางที่ 19 ทดสอบค่า Assertion method เป็น NotNull

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC06	เอกสาร User story ในช่อง Assertion method มีค่าเป็น NotNull	เพื่อตรวจสอบค่าในช่อง Assertion method NotNull	สามารถนำเข้าไฟล์เอกสาร User story และจัดเก็บเอกสารที่แฟ้มปลายทางได้ ถูกต้อง	ถูกต้อง
			สามารถเปิดไฟล์เอกสารได้ ถูกต้อง	ถูกต้อง

ตารางที่ 20 ทดสอบค่า Assertion method เป็น Null

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC07	เอกสาร User story ในช่อง Assertion method มีค่าเป็น Null	เพื่อตรวจสอบค่าในช่อง Assertion method Null	สามารถนำเข้าไฟล์เอกสาร User story และจัดเก็บเอกสารที่แฟ้มปลายทางได้ถูกต้อง	ถูกต้อง
			สามารถเปิดไฟล์เอกสารได้ถูกต้อง	ถูกต้อง

ตารางที่ 21 ทดสอบค่า Assertion method เป็น True

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC08	เอกสาร User story ในช่อง Assertion method มีค่าเป็น True	เพื่อตรวจสอบค่าในช่อง Assertion method True	สามารถนำเข้าไฟล์เอกสาร User story และจัดเก็บเอกสารที่แฟ้มปลายทางได้ถูกต้อง	ถูกต้อง
			สามารถเปิดไฟล์เอกสารได้ถูกต้อง	ถูกต้อง

ตารางที่ 22 ทดสอบค่า Assertion method เป็น False

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC09	เอกสาร User story ในช่อง Assertion method มีค่า	เพื่อตรวจสอบค่าในช่อง Assertion method False	สามารถนำเข้าไฟล์เอกสาร User story และจัดเก็บเอกสารที่แฟ้มปลายทางได้ถูกต้อง	ถูกต้อง

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการ ทดสอบ จริง
	เป็น False		สามารถเปิดไฟล์เอกสารได้ ถูกต้อง	ถูกต้อง

ตารางที่ 23 ทดสอบค่า Assertion method เป็น NotEquals

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการ ทดสอบ จริง
TC10	เอกสาร User story ในช่อง Assertion method มีค่า เป็น NotEquals	เพื่อตรวจสอบค่าในช่อง Assertion method NotEquals	สามารถนำเข้าไฟล์เอกสาร User story และจัดเก็บ เอกสารที่แฟ้มปลายทางได้ ถูกต้อง	ถูกต้อง
			สามารถเปิดไฟล์เอกสารได้ ถูกต้อง	ถูกต้อง

ตารางที่ 24 ทดสอบค่า Assertion method เป็น Equals

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการ ทดสอบ จริง
TC11	เอกสาร User story ในช่อง Assertion method มีค่า เป็น Equals	เพื่อตรวจสอบค่าในช่อง Assertion method Equals	สามารถนำเข้าไฟล์เอกสาร User story และจัดเก็บ เอกสารที่แฟ้มปลายทางได้ ถูกต้อง	ถูกต้อง
			สามารถเปิดไฟล์เอกสารได้ ถูกต้อง	ถูกต้อง

ตารางที่ 25 ทดสอบค่า Assertion method เป็น EqualsOrMoreThan

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC12	เอกสาร User story ในช่อง Assertion method มีค่าเป็น EqualsOrMoreThan	เพื่อตรวจสอบค่าในช่อง Assertion method EqualsOrMoreThan	สามารถนำเข้าไฟล์เอกสาร User story และจัดเก็บเอกสารที่แฟ้มปลายทางได้ถูกต้อง	ถูกต้อง
			สามารถเปิดไฟล์เอกสารได้ถูกต้อง	ถูกต้อง

ตารางที่ 26 ทดสอบค่า Assertion method เป็น EqualsOrLessThan

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC13	เอกสาร User story ในช่อง Assertion method มีค่าเป็น EqualsOrLessThan	เพื่อตรวจสอบค่าในช่อง Assertion method EqualsOrLessThan	สามารถนำเข้าไฟล์เอกสาร User story และจัดเก็บเอกสารที่แฟ้มปลายทางได้ถูกต้อง	ถูกต้อง
			สามารถเปิดไฟล์เอกสารได้ถูกต้อง	ถูกต้อง

ตารางที่ 27 ทดสอบค่า Assertion method เป็น MoreThan

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC14	เอกสาร User story ในช่อง Assertion method มีค่า	เพื่อตรวจสอบค่าในช่อง Assertion method MoreThan	สามารถนำเข้าไฟล์เอกสาร User story และจัดเก็บเอกสารที่แฟ้มปลายทางได้ถูกต้อง	ถูกต้อง

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการ ทดสอบ จริง
	เป็น MoreThan		สามารถเปิดไฟล์เอกสารได้ ถูกต้อง	ถูกต้อง

ตารางที่ 28 ทดสอบค่า Assertion method เป็น NotEquals

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการ ทดสอบ จริง
TC15	เอกสาร User story ในช่อง Assertion method มีค่า เป็น NotEquals	เพื่อตรวจสอบค่าในช่อง Assertion method NotEquals	สามารถนำเข้าไปไฟล์เอกสาร User story และจัดเก็บเอกสารที่แฟ้มปลายทางได้ ถูกต้อง	ถูกต้อง
			สามารถเปิดไฟล์เอกสารได้ ถูกต้อง	ถูกต้อง

ตารางที่ 29 ทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการ ทดสอบ จริง
TC16	การสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล	เพื่อทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล	จัดเก็บเอกสารที่แฟ้มปลายทางได้ถูกต้อง	ถูกต้อง

ตารางที่ 30 ทดสอบจำนวนกรณีทดสอบ

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC17	จำนวนกรณีทดสอบที่สร้างขึ้น	เพื่อทดสอบจำนวนกรณีทดสอบที่สร้างขึ้น	แสดงจำนวนกรณีทดสอบได้ครบถ้วน	ถูกต้อง

ตารางที่ 31 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Request Method แบบ GET

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC18	การสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล Request Method แบบ GET	เพื่อทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล เมื่อ User Story ระบุ Request Method เป็น GET	แสดงค่า GET ใน Tag <method> ได้ถูกต้อง	ถูกต้อง

ตารางที่ 32 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Request Method แบบ POST

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC19	การสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล Request Method แบบ POST	เพื่อทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล เมื่อ User Story ระบุ Request Method เป็น POST	แสดงค่า POST ใน Tag <method> ได้ถูกต้อง	ถูกต้อง

ตารางที่ 33 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Request Method แบบ PUT

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC20	การสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล Request Method แบบ PUT	เพื่อทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล เมื่อ User Story ระบุ Request Method เป็น PUT	แสดงค่า PUT ใน Tag <method> ได้ถูกต้อง	ถูกต้อง

ตารางที่ 34 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Request Method แบบ DELETE

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC21	การสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล Request Method แบบ DELETE	เพื่อทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล เมื่อ User Story ระบุ Request Method เป็น DELETE	แสดงค่า DELETE ใน Tag <method> ได้ถูกต้อง	ถูกต้อง

ตารางที่ 35 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น NotNull

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC22	การสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล Assertion Method แบบ NotNull	เพื่อทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล เมื่อ User Story ระบุ Assertion Method แบบ NotNull	กรณีทดสอบ แสดงค่า <param assert_type="NotNull"> ได้ถูกต้อง	ถูกต้อง

ตารางที่ 36 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น Null

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC23	การสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล Assertion Method แบบ Null	เพื่อทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล เมื่อ User Story ระบุ Assertion Method แบบ Null	กรณีทดสอบ แสดงค่า <param assert_type="Null"> ได้ถูกต้อง	ถูกต้อง

ตารางที่ 37 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น True

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC24	การสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล Assertion Method แบบ True	เพื่อทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล เมื่อ User Story ระบุ Assertion Method แบบ True	กรณีทดสอบ แสดงค่า <param assert_type="True"> ได้ถูกต้อง	ถูกต้อง

ตารางที่ 38 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น False

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC25	การสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล Assertion Method แบบ False	เพื่อทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล เมื่อ User Story ระบุ Assertion Method แบบ False	กรณีทดสอบ แสดงค่า <param assert_type="False"> ได้ถูกต้อง	ถูกต้อง

ตารางที่ 39 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น NotEquals

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC26	การสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล Assertion Method แบบ NotEquals	เพื่อทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล เมื่อ User Story ระบุ Assertion Method แบบ NotEquals	กรณีทดสอบ แสดงค่า <param assert_type="NotEquals"> ได้ถูกต้อง	ถูกต้อง

ตารางที่ 40 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น Equals

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC27	การสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล Assertion Method แบบ Equals	เพื่อทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล เมื่อ User Story ระบุ Assertion Method แบบ Equals	กรณีทดสอบ แสดงค่า <param assert_type="Equals"> ได้ถูกต้อง	ถูกต้อง

ตารางที่ 41 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น EqualsOrMoreThan

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC28	การสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล Assertion Method แบบ EqualsOrMoreThan	เพื่อทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล เมื่อ User Story ระบุ Assertion Method แบบ EqualsOrMoreThan	กรณีทดสอบ แสดงค่า <param assert_type="EqualsOrMoreThan"> ได้ถูกต้อง	ถูกต้อง

ตารางที่ 42 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น EqualsOrLessThan

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC29	การสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล Assertion Method แบบ EqualsOrLessThan	เพื่อทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล เมื่อ User Story ระบุ Assertion Method แบบ EqualsOrLessThan	กรณีทดสอบ แสดงค่า <param assert_type="EqualsOrLessThan"> ได้ถูกต้อง	ถูกต้อง

ตารางที่ 43 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น EqualsOrLessThan

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC30	การสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล Assertion Method แบบ MoreThan	เพื่อทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล เมื่อ User Story ระบุ Assertion Method แบบ MoreThan	กรณีทดสอบ แสดงค่า <param assert_type="MoreThan"> ได้ถูกต้อง	ถูกต้อง

ตารางที่ 44 ทดสอบสอบความถูกต้องของกรณีทดสอบ โดย Assertion method เป็น NotEquals

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC31	การสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล Assertion Method แบบ NotEquals	เพื่อทดสอบการสร้างกรณีทดสอบในรูปแบบเอกซ์เอ็มแอล เมื่อ User Story ระบุ Assertion Method แบบ NotEquals	กรณีทดสอบ แสดงค่า <param assert_type="NotEquals"> ได้ถูกต้อง	ถูกต้อง

ตารางที่ 45 ทดสอบความถูกต้องของกรณีทดสอบ

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC32	ทดสอบความถูกต้องของกรณีทดสอบ	เพื่อทดสอบความถูกต้องว่าแสดงผลที่ตรงกับ User story ที่ระบุ	แสดงข้อมูลในรูปแบบเอกซ์เอ็มแอลได้ตรงกับที่ระบุใน User story	ถูกต้อง

ตารางที่ 46 ทดสอบการสร้างลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอล

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC33	ทดสอบการสร้างลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอล	เพื่อทดสอบการสร้างลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอล	จัดเก็บเอกสารที่เพิ่มปลายทางได้ถูกต้อง	ถูกต้อง

ตารางที่ 47 ทดสอบความถูกต้องของลำดับการทดสอบ

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC34	ทดสอบความถูกต้องของลำดับทดสอบ	เพื่อทดสอบความถูกต้องว่าลำดับการทดสอบตรงกับ User story ที่ระบุ	แสดงลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอลได้ตรงกับที่ระบุใน User story	ถูกต้อง

ตารางที่ 48 ทดสอบการสร้างสคริปต์ทดสอบ ในรูปแบบ JUnit source code

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC35	ทดสอบการสร้างสคริปต์ทดสอบในรูปแบบ JUnit source code	เพื่อทดสอบสคริปต์ทดสอบให้อยู่ในรูปแบบ JUnit source code	จัดเก็บเอกสารที่เพิ่มปลายทางได้ถูกต้อง	ถูกต้อง
			สคริปต์ทดสอบแสดงข้อมูลได้ตรงกับที่ User story ระบุ	ถูกต้อง
			สคริปต์ทดสอบสามารถประมวลผลได้	ถูกต้อง

ตารางที่ 49 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Request Method แบบ GET

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC36	ทดสอบการสร้างสคริปต์ทดสอบในรูปแบบ JUnit source code	เพื่อทดสอบการสร้างและการทำงานของสคริปต์ทดสอบที่อยู่ในรูปแบบ JUnit source code โดยระบุ User Story Request Method เป็นแบบ GET	สคริปต์ทดสอบ สามารถสร้างได้ถูกต้องตรงตาม User story ระบุ	ถูกต้อง
			สคริปต์ทดสอบสามารถประมวลผลได้	ถูกต้อง

ตารางที่ 50 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Request Method แบบ POST

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC37	ทดสอบการสร้างสคริปต์ทดสอบในรูปแบบ JUnit source code	เพื่อทดสอบการสร้างและการทำงานของสคริปต์ทดสอบที่อยู่ในรูปแบบ JUnit source code โดยระบุ User Story Request Method เป็นแบบ POST	สคริปต์ทดสอบ สามารถสร้างได้ถูกต้องตรงตาม User story ระบุ	ถูกต้อง
			สคริปต์ทดสอบสามารถประมวลผลได้	ถูกต้อง

ตารางที่ 51 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Request Method แบบ PUT

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC38	ทดสอบการสร้างสคริปต์ทดสอบในรูปแบบ JUnit source code	เพื่อทดสอบการสร้างและการทำงานของสคริปต์ทดสอบที่อยู่ในรูปแบบ JUnit source code โดยระบุ User Story Request Method เป็นแบบ PUT	สคริปต์ทดสอบ สามารถสร้างได้ถูกต้องตรงตาม User story ระบุ	ถูกต้อง
			สคริปต์ทดสอบสามารถประมวลผลได้	ถูกต้อง

ตารางที่ 52 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Request Method แบบ DELETE

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC39	ทดสอบการสร้างสคริปต์ทดสอบในรูปแบบ JUnit source code	เพื่อทดสอบการสร้างและการทำงานของสคริปต์ทดสอบที่อยู่ในรูปแบบ JUnit source code โดยระบุ User Story Request Method เป็นแบบ DELETE	สคริปต์ทดสอบ สามารถสร้างได้ถูกต้องตรงตาม User story ระบุ	ถูกต้อง
			สคริปต์ทดสอบสามารถประมวลผลได้	ถูกต้อง

ตารางที่ 53 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น NotNull

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC40	ทดสอบการสร้างสคริปต์ทดสอบในรูปแบบ JUnit source code	เพื่อทดสอบการสร้างและการทำงานของสคริปต์ทดสอบที่อยู่ในรูปแบบ JUnit source code โดยระบุ User Story Assertion Method เป็นแบบ NotNull	สคริปต์ทดสอบ สามารถสร้างได้ถูกต้องตรงตาม User story ระบุ	ถูกต้อง
			สคริปต์ทดสอบสามารถประมวลผลได้	ถูกต้อง

ตารางที่ 54 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น Null

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC41	ทดสอบการสร้างสคริปต์ทดสอบในรูปแบบ JUnit source code	เพื่อทดสอบการสร้างและการทำงานของสคริปต์ทดสอบที่อยู่ในรูปแบบ JUnit source code โดยระบุ User Story Assertion Method เป็นแบบ Null	สคริปต์ทดสอบ สามารถสร้างได้ถูกต้องตรงตาม User story ระบุ	ถูกต้อง
			สคริปต์ทดสอบสามารถประมวลผลได้	ถูกต้อง

ตารางที่ 55 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น True

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC42	ทดสอบการสร้างสคริปต์ทดสอบในรูปแบบ JUnit source code	เพื่อทดสอบการสร้างและการทำงานของสคริปต์ทดสอบที่อยู่ในรูปแบบ JUnit source code โดยระบุ User Story Assertion Method เป็นแบบ True	สคริปต์ทดสอบ สามารถสร้างได้ถูกต้องตรงตาม User story ระบุ	ถูกต้อง
			สคริปต์ทดสอบสามารถประมวลผลได้	ถูกต้อง

ตารางที่ 56 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น False

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC43	ทดสอบการสร้างสคริปต์ทดสอบในรูปแบบ JUnit source code	เพื่อทดสอบการสร้างและการทำงานของสคริปต์ทดสอบที่อยู่ในรูปแบบ JUnit source code โดยระบุ User Story Assertion Method เป็นแบบ False	สคริปต์ทดสอบ สามารถสร้างได้ถูกต้องตรงตาม User story ระบุ	ถูกต้อง
			สคริปต์ทดสอบสามารถประมวลผลได้	ถูกต้อง

ตารางที่ 57 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น That

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC44	ทดสอบการสร้างสคริปต์ทดสอบในรูปแบบ JUnit source code	เพื่อทดสอบการสร้างและการทำงานของสคริปต์ทดสอบที่อยู่ในรูปแบบ JUnit source code โดยระบุ User Story Assertion Method เป็นแบบ That	สคริปต์ทดสอบ สามารถสร้างได้ถูกต้องตรงตาม User story ระบุ	ถูกต้อง
			สคริปต์ทดสอบสามารถประมวลผลได้	ถูกต้อง

ตารางที่ 58 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น Equals

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการ ทดสอบ จริง
TC45	ทดสอบการสร้างสคริปต์ทดสอบในรูปแบบ JUnit source code	เพื่อทดสอบการสร้างและการทำงานของสคริปต์ทดสอบที่อยู่ในรูปแบบ JUnit source code โดยระบุ User Story Assertion Method เป็นแบบ Equals	สคริปต์ทดสอบ สามารถสร้างได้ถูกต้องตรงตาม User story ระบุ	ถูกต้อง
			สคริปต์ทดสอบสามารถประมวลผลได้	ถูกต้อง

ตารางที่ 59 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น EqualsOrMoreThan

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการ ทดสอบ จริง
TC46	ทดสอบการสร้างสคริปต์ทดสอบในรูปแบบ JUnit source code	เพื่อทดสอบการสร้างและการทำงานของสคริปต์ทดสอบที่อยู่ในรูปแบบ JUnit source code โดยระบุ User Story Assertion Method เป็นแบบ EqualsOrMoreThan	สคริปต์ทดสอบ สามารถสร้างได้ถูกต้องตรงตาม User story ระบุ	ถูกต้อง
			สคริปต์ทดสอบสามารถประมวลผลได้	ถูกต้อง

ตารางที่ 60 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น EqualsOrLessThan

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC47	ทดสอบการสร้างสคริปต์ทดสอบในรูปแบบ JUnit source code	เพื่อทดสอบการสร้างและการทำงานของสคริปต์ทดสอบที่อยู่ในรูปแบบ JUnit source code โดยระบุ User Story Assertion Method เป็นแบบ EqualsOrLessThan	สคริปต์ทดสอบ สามารถสร้างได้ถูกต้องตรงตาม User story ระบุ	ถูกต้อง
			สคริปต์ทดสอบสามารถประมวลผลได้	ถูกต้อง

ตารางที่ 61 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น MoreThan

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC48	ทดสอบการสร้างสคริปต์ทดสอบในรูปแบบ JUnit source code	เพื่อทดสอบการสร้างและการทำงานของสคริปต์ทดสอบที่อยู่ในรูปแบบ JUnit source code โดยระบุ User Story Assertion Method เป็นแบบ MoreThan	สคริปต์ทดสอบ สามารถสร้างได้ถูกต้องตรงตาม User story ระบุ	ถูกต้อง
			สคริปต์ทดสอบสามารถประมวลผลได้	ถูกต้อง

ตารางที่ 62 ทดสอบการสร้างและการทำงานของสคริปต์ทดสอบ โดย Assertion method เป็น NotEquals

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC49	ทดสอบการสร้างสคริปต์ทดสอบในรูปแบบ JUnit source code	เพื่อทดสอบการสร้างและการทำงานของสคริปต์ทดสอบที่อยู่ในรูปแบบ JUnit source code โดยระบุ User Story Assertion Method เป็นแบบ NotEquals	สคริปต์ทดสอบ สามารถสร้างได้ถูกต้องตรงตาม User story ระบุ	ถูกต้อง
			สคริปต์ทดสอบสามารถประมวลผลได้	ถูกต้อง

ตารางที่ 63 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิสที่ต้องการจะทดสอบและสร้างผลลัพธ์

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC50	ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส	เพื่อทดสอบการส่งคำร้องขอที่สามารถประมวลผลได้	สามารถเรียกการทดสอบไปยังเรสตร์พูลเว็บเซอร์วิสที่ระบุใน User Story ได้	ถูกต้อง

ตารางที่ 64 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Request Method แบบ GET และสร้างผลลัพธ์

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC51	ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิสโดย Request Method เป็นแบบ GET และสร้างผลลัพธ์	เพื่อทดสอบการส่งคำร้องขอว่าสามารถประมวลผลได้และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ	สามารถเรียกการทดสอบไปยังเรสตร์พูลเว็บเซอร์วิสในรูปแบบ GET และผลลัพธ์ลงในไฟล์ลำดับการทดสอบได้ถูกต้อง	ถูกต้อง

ตารางที่ 65 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Request Method แบบ POST และสร้างผลลัพธ์

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC52	ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิสโดย Request Method เป็นแบบ POST และสร้างผลลัพธ์	เพื่อทดสอบการส่งคำร้องขอว่าสามารถประมวลผลได้และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ	สามารถเรียกการทดสอบไปยังเรสตร์พูลเว็บเซอร์วิสในรูปแบบ POST และผลลัพธ์ลงในไฟล์ลำดับการทดสอบได้ถูกต้อง	ถูกต้อง

ตารางที่ 66 ทดสอบส่งคำร้องขอไปยังเรสท์ฟูลเว็บเซอร์วิส โดย Request Method แบบ PUT และสร้างผลลัพธ์

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC53	ทดสอบส่งคำร้องขอไปยังเรสท์ฟูลเว็บเซอร์วิสโดย Request Method เป็นแบบ PUT และสร้างผลลัพธ์	เพื่อทดสอบการส่งคำร้องขอว่าสามารถประมวลผลได้และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ	สามารถเรียกการทดสอบไปยังเรสท์ฟูลเว็บเซอร์วิสในรูปแบบ PUT และผลลัพธ์ลงในไฟล์ลำดับการทดสอบได้ถูกต้อง	ถูกต้อง

ตารางที่ 67 ทดสอบส่งคำร้องขอไปยังเรสท์ฟูลเว็บเซอร์วิส โดย Request Method แบบ PUT และสร้างผลลัพธ์

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC54	ทดสอบส่งคำร้องขอไปยังเรสท์ฟูลเว็บเซอร์วิสโดย Request Method เป็นแบบ DELETE และสร้างผลลัพธ์	เพื่อทดสอบการส่งคำร้องขอว่าสามารถประมวลผลได้และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ	สามารถเรียกการทดสอบไปยังเรสท์ฟูลเว็บเซอร์วิสในรูปแบบ DELETE และผลลัพธ์ลงในไฟล์ลำดับการทดสอบได้ถูกต้อง	ถูกต้อง

ตารางที่ 68 ทดสอบส่งคำร้องขอไปยังเรสตร์ฟูลเว็บเซอร์วิส โดย Assertion method เป็น NotNull และสร้างผลลัพธ์

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC55	ทดสอบส่งคำร้องขอไปยังเรสตร์ฟูลเว็บเซอร์วิสโดยมีค่าข้อมูลอย่างน้อย 1 ค่าที่ Assertion method เป็น NotNull และสร้างผลลัพธ์	เพื่อทดสอบการส่งคำร้องขอว่าสามารถประมวลผลได้และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ	สามารถเรียกการทดสอบไปยังเรสตร์ฟูลเว็บเซอร์วิสและสามารถเปรียบเทียบข้อมูลกับค่าที่ระบุใน User story ได้และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบได้ถูกต้อง	ถูกต้อง

ตารางที่ 69 ทดสอบส่งคำร้องขอไปยังเรสตร์ฟูลเว็บเซอร์วิส โดย Assertion method เป็น Null และสร้างผลลัพธ์

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC56	ทดสอบส่งคำร้องขอไปยังเรสตร์ฟูลเว็บเซอร์วิสโดยมีค่าข้อมูลอย่างน้อย 1 ค่าที่ Assertion method เป็น Null และสร้างผลลัพธ์	เพื่อทดสอบการส่งคำร้องขอว่าสามารถประมวลผลได้และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ	สามารถเรียกการทดสอบไปยังเรสตร์ฟูลเว็บเซอร์วิสและสามารถเปรียบเทียบข้อมูลกับค่าที่ระบุใน User story ได้และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบได้ถูกต้อง	ถูกต้อง

ตารางที่ 70 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น True และสร้างผลลัพธ์

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC57	ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิสโดยมีค่าข้อมูลอย่างน้อย 1 ค่าที่ Assertion method เป็น True และสร้างผลลัพธ์	เพื่อทดสอบการส่งคำร้องขอที่สามารถประมวลผลได้และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ	สามารถเรียกการทดสอบไปยังเรสตร์พูลเว็บเซอร์วิสและสามารถเปรียบเทียบข้อมูลกับค่าที่ระบุใน User story ได้และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบได้ถูกต้อง	ถูกต้อง

ตารางที่ 71 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น False และสร้างผลลัพธ์

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC58	ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิสโดยมีค่าข้อมูลอย่างน้อย 1 ค่าที่ Assertion method เป็น False และสร้างผลลัพธ์	เพื่อทดสอบการส่งคำร้องขอที่สามารถประมวลผลได้และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบ	สามารถเรียกการทดสอบไปยังเรสตร์พูลเว็บเซอร์วิสและสามารถเปรียบเทียบข้อมูลกับค่าที่ระบุใน User story ได้และสร้างผลลัพธ์ลงในไฟล์ลำดับการทดสอบได้ถูกต้อง	ถูกต้อง

ตารางที่ 72 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น That และ สร้างผลลัพธ์

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการ ทดสอบ จริง
TC59	ทดสอบส่งคำ ร้องขอไป ยังเรสตร์พูลเว็บ เซอร์วิสโดยมี ค่าข้อมูลอย่าง น้อย 1 ค่าที่ Assertion method เป็น That และ สร้างผลลัพธ์	เพื่อทดสอบการส่งคำ ร้องขอที่สามารถ ประมวลผลได้และสร้าง ผลลัพธ์ลงในไฟล์ลำดับ การทดสอบ	สามารถเรียกการทดสอบไป ยังเรสตร์พูลเว็บเซอร์วิสและ สามารถเปรียบเทียบข้อมูล กับค่าที่ระบุใน User story ได้และสร้างผลลัพธ์ลงในไฟล์ ลำดับการทดสอบได้ถูกต้อง	ถูกต้อง

ตารางที่ 73 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น Equals และสร้างผลลัพธ์

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการ ทดสอบ จริง
TC60	ทดสอบส่งคำ ร้องขอไป ยังเรสตร์พูลเว็บ เซอร์วิสโดยมี ค่าข้อมูลอย่าง น้อย 1 ค่าที่ Assertion method เป็น Equals และ สร้างผลลัพธ์	เพื่อทดสอบการส่งคำ ร้องขอที่สามารถ ประมวลผลได้และสร้าง ผลลัพธ์ลงในไฟล์ลำดับ การทดสอบ	สามารถเรียกการทดสอบไป ยังเรสตร์พูลเว็บเซอร์วิสและ สามารถเปรียบเทียบข้อมูล กับค่าที่ระบุใน User story ได้และสร้างผลลัพธ์ลงในไฟล์ ลำดับการทดสอบได้ถูกต้อง	ถูกต้อง

ตารางที่ 74 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น EqualsOrMoreThan และสร้างผลลัพธ์

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการ ทดสอบ จริง
TC61	ทดสอบส่งคำ ร้องขอไป ยังเรสตร์พูลเว็บ เซอร์วิสโดยมี ค่าข้อมูลอย่าง น้อย 1 ค่าที่ Assertion method เป็น EqualsOrMo reThan และ สร้างผลลัพธ์	เพื่อทดสอบการส่งคำ ร้องขอที่สามารถ ประมวลผลได้และสร้าง ผลลัพธ์ลงในไฟล์ลำดับ การทดสอบ	สามารถเรียกการทดสอบไป ยังเรสตร์พูลเว็บเซอร์วิสและ สามารถเปรียบเทียบข้อมูล กับค่าที่ระบุใน User story ได้และสร้างผลลัพธ์ลงในไฟล์ ลำดับการทดสอบได้ถูกต้อง	ถูกต้อง

ตารางที่ 75 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น EqualsOrLessThan และสร้างผลลัพธ์

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการ ทดสอบ จริง
TC62	ทดสอบส่งคำ ร้องขอไป ยังเรสตร์พูลเว็บ เซอร์วิสโดยมี ค่าข้อมูลอย่าง น้อย 1 ค่าที่ Assertion method เป็น EqualsOrLes sThan และ สร้างผลลัพธ์	เพื่อทดสอบการส่งคำ ร้องขอที่สามารถ ประมวลผลได้และสร้าง ผลลัพธ์ลงในไฟล์ลำดับ การทดสอบ	สามารถเรียกการทดสอบไป ยังเรสตร์พูลเว็บเซอร์วิสและ สามารถเปรียบเทียบข้อมูล กับค่าที่ระบุใน User story ได้และสร้างผลลัพธ์ลงในไฟล์ ลำดับการทดสอบได้ถูกต้อง	ถูกต้อง

ตารางที่ 76 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น MoreThan และสร้างผลลัพธ์

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการ ทดสอบ จริง
TC63	ทดสอบส่งคำ ร้องขอไป ยังเรสตร์พูลเว็บ เซอร์วิสโดยมี ค่าข้อมูลอย่าง น้อย 1 ค่าที่ Assertion method เป็น MoreThan และสร้าง ผลลัพธ์	เพื่อทดสอบการส่งคำ ร้องขอที่สามารถ ประมวลผลได้และสร้าง ผลลัพธ์ลงในไฟล์ลำดับ การทดสอบ	สามารถเรียกการทดสอบไป ยังเรสตร์พูลเว็บเซอร์วิสและ สามารถเปรียบเทียบข้อมูล กับค่าที่ระบุใน User story ได้และสร้างผลลัพธ์ลงในไฟล์ ลำดับการทดสอบได้ถูกต้อง	ถูกต้อง

ตารางที่ 77 ทดสอบส่งคำร้องขอไปยังเรสตร์พูลเว็บเซอร์วิส โดย Assertion method เป็น NotEquals และสร้างผลลัพธ์

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการ ทดสอบ จริง
TC64	ทดสอบส่งคำ ร้องขอไป ยังเรสตร์พูลเว็บ เซอร์วิสโดยมี ค่าข้อมูลอย่าง น้อย 1 ค่าที่ Assertion method เป็น NotEquals และสร้าง ผลลัพธ์	เพื่อทดสอบการส่งคำ ร้องขอที่สามารถ ประมวลผลได้และสร้าง ผลลัพธ์ลงในไฟล์ลำดับ การทดสอบ	สามารถเรียกการทดสอบไป ยังเรสตร์พูลเว็บเซอร์วิสและ สามารถเปรียบเทียบข้อมูล กับค่าที่ระบุใน User story ได้และสร้างผลลัพธ์ลงในไฟล์ ลำดับการทดสอบได้ถูกต้อง	ถูกต้อง

ตารางที่ 78 ทดสอบการแสดงผลลัพธ์ที่หน้าจอการทำงานหลักจากไฟล์ลำดับกรณีทดสอบ

หมายเลขกรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลการทดสอบที่คาดหวัง	ผลการทดสอบจริง
TC65	ทดสอบแสดงผลลัพธ์ที่หน้าจอ	ทดสอบแสดงผลลัพธ์ที่หน้าจอ หลังจากส่งคำร้องไปยังเรสต์ฟูลเว็บเซอร์วิส	สามารถแสดงผลลัพธ์และรายละเอียดอื่นๆ ที่หน้าจอการทำงานหลักได้	ถูกต้อง

5.2 การประเมินผลระบบ

การประเมินผลระบบ จะประเมินโดยทดสอบระบบทดสอบเรสต์ฟูลเว็บเซอร์วิสกับ User story ถูกสร้างขึ้น

5.2.1 ข้อมูลนำเข้า

User story ดังตารางที่ 79 ประกอบด้วยคำอธิบายความต้องการที่จะทดสอบเรสต์ฟูลเว็บเซอร์วิส 5 กรณีทดสอบ

ตารางที่ 79 ทดสอบการแสดงผลลัพธ์ที่หน้าจอการทำงานหลักจากไฟล์ลำดับกรณีทดสอบ

Feature	BillPayment		
Description	Bill payment service.		
TEST-CASE-ID-001	InquiryTrueMoveBillSuccess		
Request Method	GET		
Request URL	http://localhost:8080/counter/online/bill/inquiry/truemove/0809001044		
Request Attribute	Name	Value	
Response Attribute	Name	Assertion method	Expected
	biller_code	Equals	truemove

	account_no	Equals	0809001044
	account_status	NotNull	
	fullname	NotNull	
	total_outstanding	EqualsOrMoreThan	0
TEST-CASE-ID-002	PaymentSuccess		
Request Method	POST		
Request URL	http://localhost:8080/counter/online/bill/pay		
Request Attribute	Name	Value	
	biller_code	truemove	
	account_no	0809001044	
	amount	534.47	
Response Attribute	Name	Assertion method	Expected
	payment_id	NotNull	
	payment_date	NotNull	
	result_code	Equals	0
TEST-CASE-ID-003	InquiryPaymentSuccess		
Request Method	GET		
Request URL	http://localhost:8080/counter/online/bill/pay/20140613212411		
Request Attribute	Name	Value	
Response Attribute	Name	Assertion method	Expected
	payment_id	Equals	20140613212411

	payment_date	Equals	13/6/2014
	payment_status	Equals	paid
	biller_code	Equals	truemove
	account_no	Equals	0809001044
	paid_amount	Equals	534.47
TEST-CASE-ID-004	CancelPaymentSuccess		
Request Method	POST		
Request URL	http://localhost:8080/counter/online/bill/pay/20140613212411/cancel		
Request Attribute	Name	Value	
Response Attribute	Name	Assertion method	Expected
	payment_id	Equals	20140613212411
	cancel_date	NotNull	
	result_code	Equals	0
TEST-CASE-ID-005	InquiryAISBillSuccess		
Request Method	GET		
Request URL	http://localhost:8080/counter/online/bill/inquiry/ais/0809001044		
Request Attribute	Name	Value	
Response Attribute	Name	Assertion method	Expected
	biller_code	Equals	ais

	account_no	Equals	0809001044
	account_status	NotNull	
	fullname	NotNull	
	total_outstanding	EqualsOrMoreThan	0

5.2.2 ข้อมูลนำออก

1. กรณีทดสอบในรูปแบบเอกซ์เอ็มแอลทั้งหมด 5 ไฟล์ 5 กรณีทดสอบ
 - TEST-CASE-ID-001 InquiryTrueMoveBillSuccess ดังรูปที่ 22
 - TEST-CASE-ID-002 PaymentSuccess ดังรูปที่ 3
 - TEST-CASE-ID-003 InquiryPaymentSuccess ดังรูปที่ 24
 - TEST-CASE-ID-004 CancelPaymentSuccess ดังรูปที่ 25
 - TEST-CASE-ID-005 InquiryAISBillSuccess ดังรูปที่ 26

```
<?xml version="1.0" encoding="UTF-8"?>
<testcase>
  <id>TEST-CASE-ID-001</id>
  <name>InquiryTrueMoveBillSuccess</name>
  <method>GET</method>
  <url>http://localhost:8080/counter/online/bill/inquiry/truemove/0809001044</url>
  <response>
    <param name="biller_code" expected="truemove" assert_type="Equals"/>
    <param name="account_no" expected="0809001044" assert_type="Equals"/>
    <param name="account_status" assert_type="NotNull"/>
    <param name="fullname" assert_type="NotNull"/>
    <param name="total_outstanding" expected="0" assert_type="EqualsOrMoreThan"/>
  </response></testcase>
```

รูปที่ 22 กรณีทดสอบ TEST-CASE-ID-001 ในรูปแบบเอกซ์เอ็มแอลแบบ GET

```

<?xml version="1.0" encoding="UTF-8"?>
<testcase>
  <id>TEST-CASE-ID-002</id>
  <name>PaymentSuccess</name>
  <method>POST</method><url>http://localhost:8080/counter/online/bill/pay</url>
  <request>
    <param name="biller_code" value="truemove"/>
    <param name="account_no" value="0809001044"/>
    <param name="amount" value="534.47"/>
  </request>
  <response>
    <param name="payment_id" assert_type="NotNull"/>
    <param name="payment_date" assert_type="NotNull"/>
    <param name="result_code" expected="0" assert_type="Equals"/>
  </response></testcase>

```

รูปที่ 23 กรณีทดสอบ TEST-CASE-ID-002 ในรูปแบบเอกซ์เอ็มแอลแบบ POST

```

<?xml version="1.0" encoding="UTF-8"?>
<testcase>
  <id>TEST-CASE-ID-003</id>
  <name>InquiryPaymentSuccess</name>
  <method>GET</method>
  <url>http://localhost:8080/counter/online/bill/pay/20140613212411</url>
  <response>
    <param name="payment_id" expected="20140613212411" assert_type="Equals"/>
    <param name="payment_date" expected="13/06/2014" assert_type="Equals"/>
    <param name="payment_status" expected="paid" assert_type="Equals"/>
    <param name="biller_code" expected="truemove" assert_type="Equals"/>
    <param name="account_no" expected="0809001044" assert_type="Equals"/>
    <param name="paid_amount" expected="534.47" assert_type="Equals"/>
  </response>
</testcase>

```

รูปที่ 24 กรณีทดสอบ TEST-CASE-ID-003 ในรูปแบบเอกซ์เอ็มแอลแบบ GET

```

<?xml version="1.0" encoding="UTF-8"?>
<testcase>
  <id>TEST-CASE-ID-004</id>
  <name>CancelPaymentSuccess</name>
  <method>POST</method>
  <url>http://localhost:8080/counter/online/bill/pay/20140613212411/cancel</url>
  <response>
    <param name="payment_id" expected="20140613212411" assert_type="Equals"/>
    <param name="cancel_date" assert_type="NotNull"/>
    <param name="result_code" expected="0" assert_type="Equals"/>
  </response>
</testcase>

```

รูปที่ 25 กรณีทดสอบ TEST-CASE-ID-004 ในรูปแบบเอกซ์เอ็มแอลแบบ POST

```

<?xml version="1.0" encoding="UTF-8"?>
<testcase>
  <id>TEST-CASE-ID-005</id>
  <name>InquiryAISBillSuccess</name>
  <method>GET</method>
  <url>http://localhost:8080/counter/online/bill/inquiry/ais/0809001044</url>
  <response>
    <param name="biller_code" expected="ais" assert_type="Equals"/>
    <param name="account_no" expected="0809001044" assert_type="Equals"/>
    <param name="account_status" assert_type="NotNull"/>
    <param name="fullname" assert_type="NotNull"/>
    <param name="total_outstanding" expected="0" assert_type="EqualsOrMoreThan"/>
  </response>
</testcase>

```

รูปที่ 26 กรณีทดสอบ TEST-CASE-ID-005 ในรูปแบบเอกซ์เอ็มแอลแบบ GET

2. ลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอลทั้งหมด 1 ไฟล์เท่านั้น ดังรูปที่ 27

```
<?xml version="1.0" encoding="UTF-8"?>
<testsuite<id>BillPayment</id><sequence>
    <testcase>
        <src>TEST-CASE-ID-001.xml</src>
    </testcase>
    <testcase>
        <src>TEST-CASE-ID-002.xml</src>
    </testcase>
    <testcase>
        <src>TEST-CASE-ID-003.xml</src>
    </testcase>
    <testcase>
        <src>TEST-CASE-ID-004.xml</src>
    </testcase>
    <testcase>
        <src>TEST-CASE-ID-005.xml</src>
    </testcase>
</sequence></testsuite>
```

รูปที่ 27 ลำดับทดสอบในรูปแบบเอกซ์เอ็มแอล

3. สคริปต์ทดสอบในรูปแบบ JUnit source code ดังรูปที่ 28

```
package th.ac.cu.testsuite;
import java.util.HashMap;
import java.util.Map;
import org.junit.Test;
import th.ac.cu.util.RESTClientUtils;
import th.ac.cu.util.TestCaseAssertionUtils;

public class BillPaymentTests {

    // TEST-CASE-ID-001
    @Test
    public void testInquiryTrueMoveBillSuccess() {

        Map<String, String> responseDatas = RESTClientUtils.callTo("GET",
            "http://localhost:8080/counter/online/bill/inquiry/truemove/0809001044");

        TestCaseAssertionUtils.assertEquals("truemove", responseDatas.get("biller_code"));
        TestCaseAssertionUtils.assertEquals("0809001044", responseDatas.get("account_no"));
    }
}
```

```

        TestCaseAssertionUtils.assertNotNull(responseDatas.get("account_status"));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("fullname"));
        TestCaseAssertionUtils.assertEqualsOrMoreThan("0", responseDatas.get("total_outstanding"));
    }

// TEST-CASE-ID-002
@Test
public void testPaymentSuccess() {

    Map<String, String> requestDatas = new HashMap<String, String>();
        requestDatas.put("biller_code", "truemove");
        requestDatas.put("account_no", "0809001044");
        requestDatas.put("amount", "534.47");

        Map<String, String> responseDatas = RestClientUtils.callTo("POST",
"http://localhost:8080/counter/online/bill/pay", requestDatas);

        TestCaseAssertionUtils.assertNotNull(responseDatas.get("payment_id"));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("payment_date"));
        TestCaseAssertionUtils.assertEquals("0", responseDatas.get("result_code"));
    }

// TEST-CASE-ID-003
@Test
public void testInquiryPaymentSuccess() {

    Map<String, String> responseDatas = RestClientUtils.callTo("GET",
"http://localhost:8080/counter/online/bill/pay/20140613212411");

        TestCaseAssertionUtils.assertEquals("20140613212411", responseDatas.get("payment_id"));
        TestCaseAssertionUtils.assertEquals("13/06/2014", responseDatas.get("payment_date"));
        TestCaseAssertionUtils.assertEquals("paid", responseDatas.get("payment_status"));
        TestCaseAssertionUtils.assertEquals("truemove", responseDatas.get("biller_code"));
        TestCaseAssertionUtils.assertEquals("0809001044", responseDatas.get("account_no"));
        TestCaseAssertionUtils.assertEquals("534.47", responseDatas.get("paid_amount"));
    }

// TEST-CASE-ID-004
@Test
public void testCancelPaymentSuccess() {

    Map<String, String> requestDatas = new HashMap<String, String>();

```



```

        Map<String, String> responseDatas = RESTClientUtils.callTo("POST",
"http://localhost:8080/counter/online/bill/pay/20140613212411/cancel", requestDatas);

        TestCaseAssertionUtils.assertEquals("20140613212411", responseDatas.get("payment_id"));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("cancel_date"));
        TestCaseAssertionUtils.assertEquals("0", responseDatas.get("result_code"));
    }
// TEST-CASE-ID-005
@Test
public void testInquiryAISBillSuccess() {
    Map<String, String> responseDatas = RESTClientUtils.callTo("GET",
"http://localhost:8080/counter/online/bill/inquiry/ais/0809001044");

    TestCaseAssertionUtils.assertEquals("ais", responseDatas.get("biller_code"));
    TestCaseAssertionUtils.assertEquals("0809001044", responseDatas.get("account_no"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("account_status"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("fullname"));
    TestCaseAssertionUtils.assertEqualsOrMoreThan("0", responseDatas.get("total_outstanding"));
}
}

```

รูปที่ 28 สคริปต์ทดสอบในรูปแบบ JUnit source code

4. หลังจากส่งคำร้องขอไปยังเรสตร์ฟูลเว็บเซอร์วิสเพื่อทดสอบเรียบร้อยแล้ว ระบบจำทำการสร้างค่าสคริปต์ทดสอบลงไฟล์ลำดับทดสอบ ในรูปแบบเอกซ์เอ็มแอล ดังรูปที่ 29

```

<?xml version="1.0" encoding="UTF-8"?>
<testsuite <id>BillPayment</id><sequence>
  <testcase>
    <src>TEST-CASE-ID-001.xml</src>
    <result>PASS</result>
  </testcase>
  <testcase>
    <src>TEST-CASE-ID-002.xml</src>
    <result>PASS</result>
  </testcase>
  <testcase>
    <src>TEST-CASE-ID-003.xml</src>
    <result>PASS</result>
  </testcase>
  <testcase>
    <src>TEST-CASE-ID-004.xml</src>

```

```

        <result>PASS</result>
    </testcase>
    <testcase>
        <src>TEST-CASE-ID-005.xml</src>
        <result>FAILED</result>
    </testcase>
</sequence></testsuite>

```

รูปที่ 29 ลำดับการทดสอบแบบแสดงผลลัพธ์

5. แสดงผลลัพธ์การทดสอบและข้อมูลรายละเอียดอื่นๆ ของกรณีทดสอบทางหน้าจอการทำงานหลัก

The screenshot shows a web browser window with the URL localhost:8080/test-upload-api/. The page title is "Testing RESTful Web Services". Below the title, there is a "User Story" input field, a "เลือกไฟล์" button, a "ไม่ได้เลือกไฟล์ใด" button, and a "Cancel Upload" button. Below this is a section titled "ผลลัพธ์การทดสอบ" (Test Results) containing a table with the following data:

ID	Name	Method	URL	Result
TEST-CASE-ID-001	InquiryTrueMoveBillSuccess	GET	http://localhost:8080/counter/online/bill/inquiry/truemove/0809001044	PASS
TEST-CASE-ID-002	PaymentSuccess	POST	http://localhost:8080/counter/online/bill/pay	PASS
TEST-CASE-ID-003	InquiryPaymentSuccess	GET	http://localhost:8080/counter/online/bill/pay/20140613212411	PASS
TEST-CASE-ID-004	CancelPaymentSuccess	POST	http://localhost:8080/counter/online/bill/pay/20140613212411/cancel	PASS
TEST-CASE-ID-005	InquiryAISBillSuccess	GET	http://localhost:8080/counter/online/bill/inquiry/ais/0809001044	FAILED

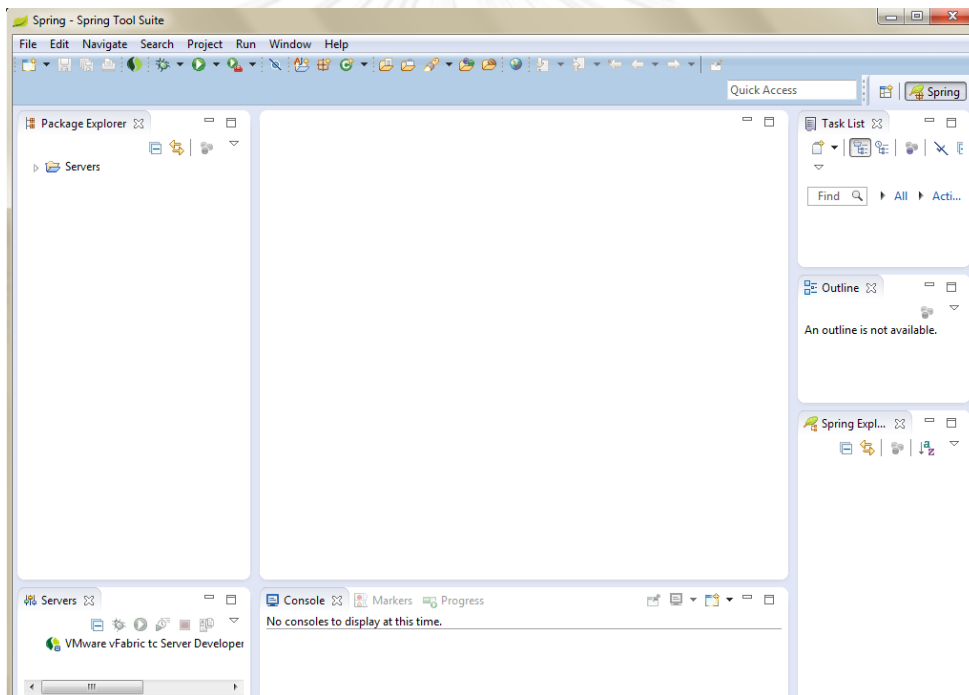
รูปที่ 30 หน้าจอแสดงผลลัพธ์ที่อ่านค่าจากไฟล์ลำดับการทดสอบ

5.2.3 การประมวลผลสคริปต์ทดสอบ

สคริปต์ทดสอบที่ได้จากระบบทดสอบเรสต์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอจิล์ สามารถนำไปประมวลผลได้บ่อยครั้งเท่าที่ต้องการ เครื่องมือสปริง ทูล สวีท เวอร์ชัน 3.2.2 ดังรูปที่ 31 เป็นเครื่องมือในการนำเข้าสู่สคริปต์ทดสอบในรูปแบบ JUnit source code



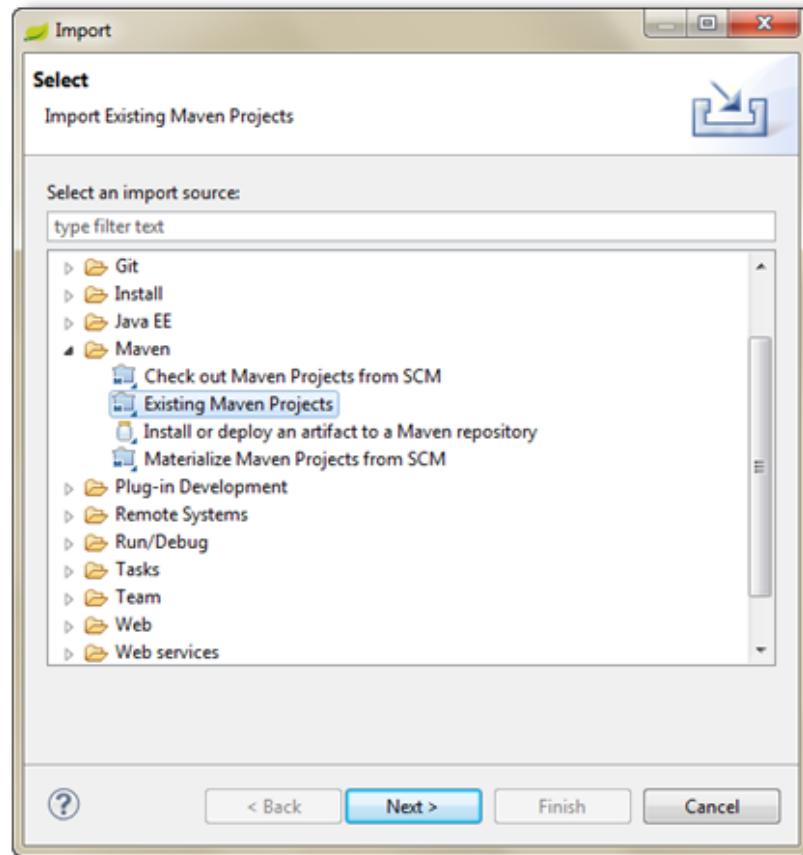
รูปที่ 31 เครื่องมือสปริง ทูล สวีท เวอร์ชัน 3.2.2



รูปที่ 32 หน้าจอการทำงานของเครื่องมือสปริง ทูล สวีท

5.2.3.1 การนำเข้าสคริปต์ทดสอบ

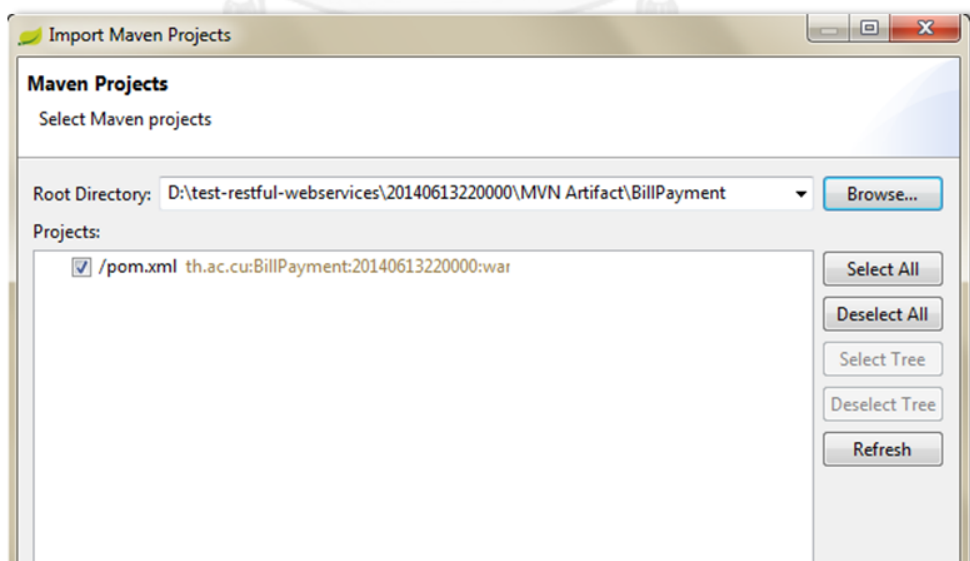
1. เลือกที่แถบเมนูด้านบนดังนี้ File > Import > Maven > Existing Maven Project ดังรูปที่ 33



รูปที่ 33 การนำเข้าสคริปต์ทดสอบ

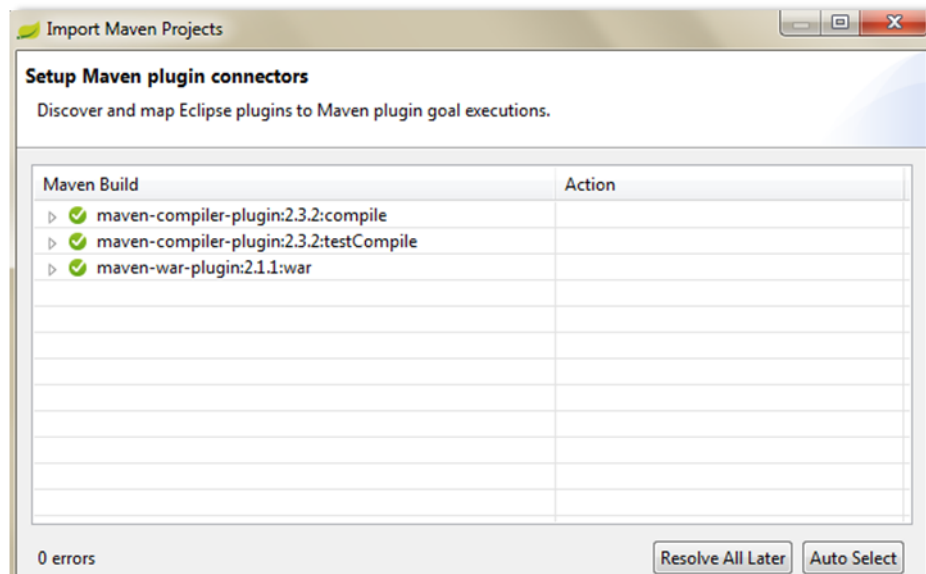
2. Root Directory เลือกไปยังเพิ่มเอกสารสคริปต์ทดสอบ > Next > Finish

ดังรูปที่ 34



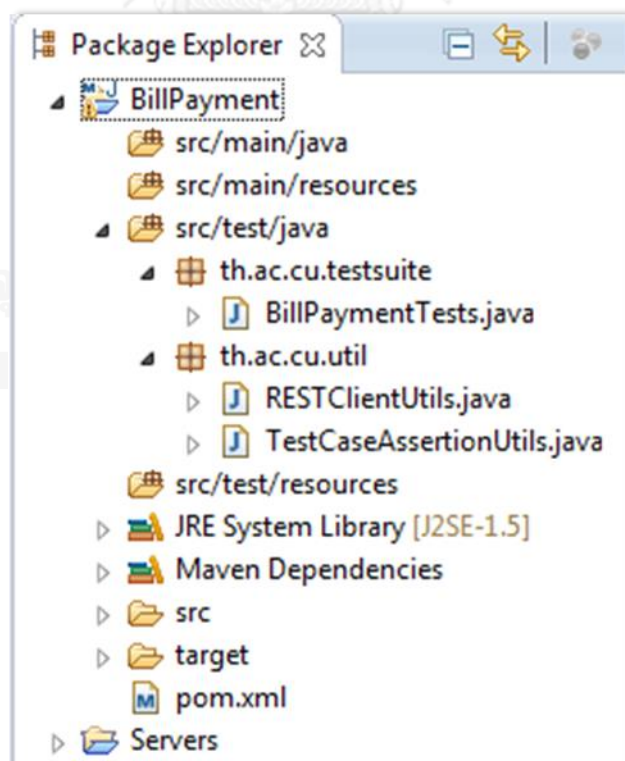
รูปที่ 34 หน้าจอการเลือกเพิ่มเอกสารสคริปต์ทดสอบ

3. แสดงรายการของ Maven Build ตามโครงสร้างสคริปต์ทดสอบ ดังรูปที่ 35



รูปที่ 35 หน้าจอการตั้งค่า Maven plugin connectors

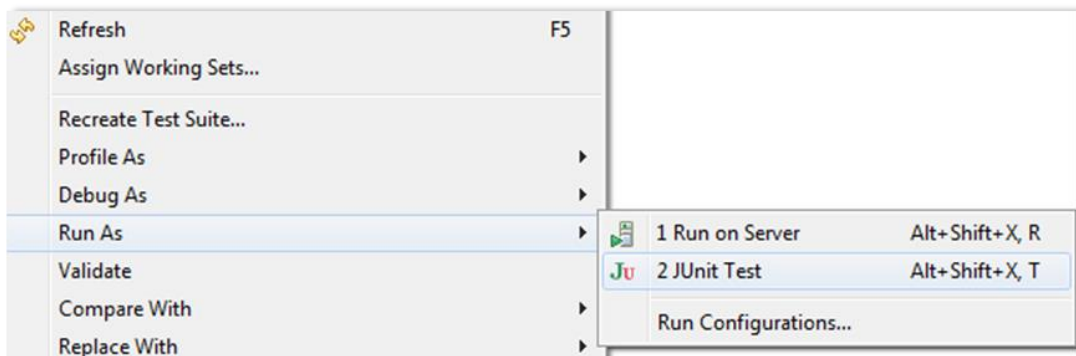
4. ที่ช่อง Package Explorer จะแสดงสคริปต์ทดสอบตามโครงสร้าง Maven project ดังรูปที่ 36



รูปที่ 36 หน้าจอแสดงสคริปต์ทดสอบตามโครงสร้าง Maven project

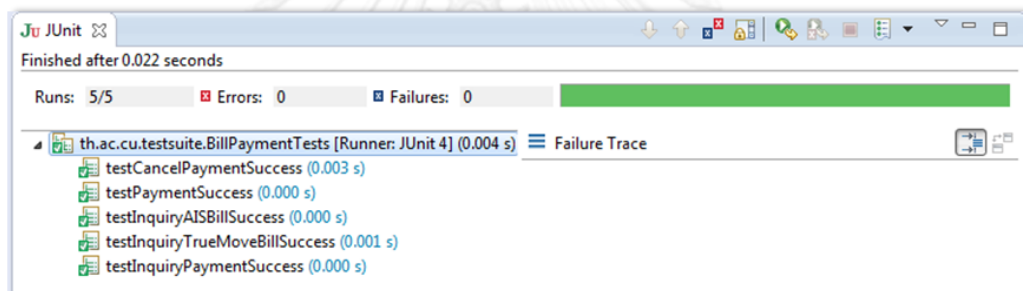
5.2.3.2 การประมวลผลสคริปต์ทดสอบ

1. คลิกขวาที่ไฟล์ BillPaymentTest.java เลือก Run As > JUnit Test ดังรูปที่ 37



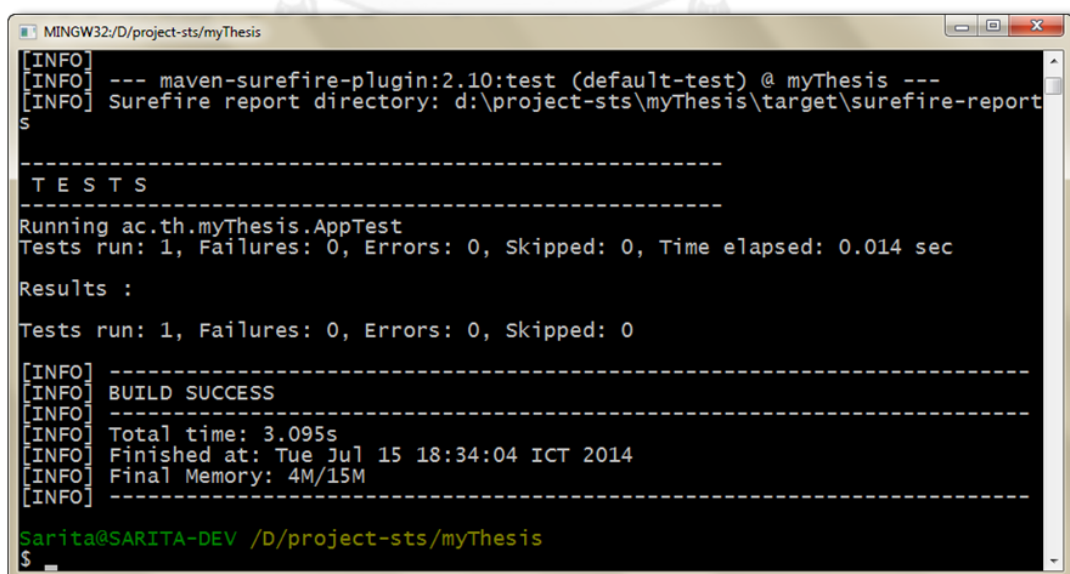
รูปที่ 37 การประมวลผลสคริปต์ทดสอบ

2. ผลการทำงานของของสคริปต์ทดสอบ ในรูปแบบ JUnit source code ดังรูปที่ 38



รูปที่ 38 ผลลัพธ์การประมวลผลด้วย JUnit

3. ผลของการทดสอบผ่าน Command Line “maven test”



รูปที่ 39 ผลลัพธ์การประมวลผลด้วย มาเว่น

บทที่ 6

สรุปผลการวิจัย และข้อเสนอแนะ

6.1 สรุปผลการวิจัย

ผลลัพธ์โดยสรุปที่ได้จากงานวิจัยนี้ มีดังนี้

1. ได้แนวทางสำหรับการทดสอบเรสท์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอไจล์
2. ได้ระบบต้นแบบทดสอบเรสท์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอไจล์
3. ได้เครื่องมือในการสร้างกรณีทดสอบและลำดับการทดสอบ
4. ได้เครื่องมือในการสร้างสคริปต์ทดสอบ
5. ได้เครื่องมือทดสอบเรสท์ฟูลเว็บเซอร์วิส เพื่อใช้ในการทดสอบจริง
6. ได้เครื่องมือสนับสนุนการพัฒนาซอฟต์แวร์ ซึ่งช่วยลดระยะเวลา ลดทรัพยากรและข้อผิดพลาดในกระบวนการพัฒนาซอฟต์แวร์

6.2 ข้อจำกัด

- User Story ที่นำเข้ามาในงานวิจัยนี้รองรับเฉพาะไมโครซอฟท์ เอกซ์เซล 2010
- รองรับบริการเรสท์ฟูลเว็บเซอร์วิสที่จะทดสอบในรูปแบบเจสัน ที่เป็น GET, POST, PUT และDELETE เท่านั้น

6.3 แนวทางการวิจัยต่อ

แนวทางการวิจัยนี้สามารถนำไปพัฒนาเครื่องมือทดสอบให้อยู่ในรูปแบบอัตโนมัติ เพื่อเพิ่มความสะดวกและแม่นยำในการทดสอบ และสามารถตรวจสอบความซับซ้อนของบริการที่จะทำการทดสอบได้ สามารถพยากรณ์ จำนวนกรณีทดสอบได้จาก User Story เพิ่มความปลอดภัยในการรักษาข้อมูลระหว่างทำการทดสอบ

รายการอ้างอิง

1. Rodriguez, A. RESTful Web services: The basic. 2008; Available from: <http://www.ibm.com/developerworks/webservices/library/ws-restful/>.
2. Sriparasa, S.S., JavaScript and JSON Essentials. 2013, UK: Packt Publishing Ltd.
3. P. Tahchiev, F.L., V. Massol, and G. Gregory, JUnit IN ACTION. 2011, Stamford: MANNING.
4. Fei, W. and D. Wencai. A Test Automation Framework Based on WEB. in Computer and Information Science (ICIS), 2012 IEEE/ACIS 11th International Conference on. 2012.
5. Sujit, C., and Prashant, K, Test-the-REST: An Approach to Testing RESful Web-Services, in Proceedings of the 2009 Computation World. 2009. p. 302-308.



ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาคผนวก ก

ผลการทดสอบ ระบบทดสอบเรสท์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอไจล์

ผลการประเมินผลระบบ จะแสดง User story ไฟล์นำออก สคริปต์ทดสอบ และผลลัพธ์กการทดสอบ โดยทดสอบระบบทดสอบเรสท์ฟูลเว็บเซอร์วิสกับ User story ถูกสร้างขึ้นตามกรณีทดสอบ

1. กรณีทดสอบเรสท์ฟูลเว็บเซอร์วิส โดย Request Method แบบ GET, POST, PUT, DELETE และมีค่าข้อมูลอย่างน้อย 1 ค่า ที่ Assertion method เป็น NotNull, Null, True, False , That, Equals, EqualsOrMoreThan, EqualsOrLessThan, MoreThan, NotEquals

- 1.1 ข้อมูลนำเข้า User story ดังตารางที่ 80 ประกอบด้วยคำอธิบายความต้องการที่จะทดสอบเรสท์ฟูลเว็บเซอร์วิส 4 กรณีทดสอบ

ตารางที่ 80 ตาราง User story

Feature	Transaction Report		
Description	As a operation, I want a search, Add, Update, Delete report transaction so that I can see transaction details and export report to report her request		
TEST-CASE-ID-001	InquiryTrueMoveSuccess		
Request Method	GET		
Request URL	http://localhost:8080/counter/online/report/transaction/transactionID/1000479015		
Request Attribute	Name	Value	
Response Attribute	Name	Assertion method	Expected
	trans_id	Equals	1000479015
	account_no	NotNull	
	terminal_id	NotNull	
	svc_id	NotNull	
	state	MoreThan	0
	code	NotNull	

	namespace	NotNull	
	pay_amount	NotNull	
	svc_fee	EqualsOrLessThan	10
	service_code	NotNull	
	pay_ref1	NotNull	
	pay_txn	NotNull	
	sof_type	NotNull	
	sof_ref1	NotNull	
	sof_ref2	NotNull	
	debit_txn	NotNull	
	l2_comm	Null	
	sof_fee	EqualsOrMoreThan	10
TEST-CASE-ID-002	UpdateTransaction		
Request Method	POST		
Request URL	http://localhost:8080/counter/report/transaction		
Request Attribute	Name	Value	
	transactionid	1000477826	
	status	commit	
Response Attribute	Name	Assertion method	Expected
	response_code	Equals	0
	response_desc	NotNull	
	tr_desc	NotNull	
	conf_instance	True	0
	km.sessionid	NotEquals	-1
TEST-CASE-ID-003	AdjustTransaction		
Request Method	PUT		

Request URL	http://localhost:8080/counter/report/transaction/		
Request Attribute	Name	Value	
	transactionid	1000477826	
	adjust	True	
Response Attribute	Name	Assertion method	Expected
	response_code	Equals	0
	response_desc	NotNull	
	km.sessionid	That	0
	km.wait	False	0
	token	NotNull	
	initiator	NotNull	
	pin	NotNull	
	cashto	NotNull	
	result_namespace	NotNull	
	vtrid	NotNull	
TEST-CASE-ID-004	CancelTransaction		
Request Method	DELETE		
Request URL	http://localhost:8080/counter/report/transaction/ 1000477861/Cancel		
Request Attribute	Name	Value	
Response Attribute	Name	Assertion method	Expected
	response_code	Equals	0
	response_desc	NotNull	
	ext_id	NotNull	
	iotype	True	0

	iomse	NotNull	
	created	NotNull	
	dkey	NotNull	
	dval	NotNull	
	ext_desc	NotNull	
	ext_login	NotEquals	-1

1.2 ข้อมูลนำออก กรณีทดสอบในรูปแบบเอกซ์เอ็มแอลทั้งหมด 4 ไฟล์ 4 กรณีทดสอบและลำดับการทดสอบในรูปแบบเอกซ์เอ็มแอลทั้งหมด 1 ไฟล์ ดังรูปที่ 44 และ สคริปต์ทดสอบในรูปแบบ JUnit source code ดังรูปที่

- TEST-CASE-ID-001 InquiryTransaction ดังรูปที่ 40
- TEST-CASE-ID-002 UpdateTransaction ดังรูปที่ 41
- TEST-CASE-ID-003 CancelTransaction ดังรูปที่ 42
- TEST-CASE-ID-004 AdjustTransaction ดังรูปที่ 43

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<testcase>
  <id>TEST-CASE-ID-001</id>
  <method>GET</method>
  <name>InquiryTransaction</name>
  <request/>
  <response>
    <param assert_type="Equals" expected="1000479015" name="TRANS_ID"/>
    <param assert_type="NotNull" name="account_no"/>
    <param assert_type="NotNull" name="TERMINAL_ID"/>
    <param assert_type="NotNull" name="SVC_ID"/>
    <param assert_type="MoreThan" expected="0" name="STATE"/>
      <param assert_type="NotNull" name="CODE"/>
      <param assert_type="NotNull" name="NAMESPACE"/>
      <param assert_type="NotNull" name="PAY_AMOUNT"/>
      <param assert_type="EqualsOrLessThan" expected="10" name="SVC_FEE"/>
      <param assert_type="NotNull" name="SERVICE_CODE"/>
      <param assert_type="NotNull" name="PAY_REF1"/>
      <param assert_type="NotNull" name="PAY_TXN"/>
      <param assert_type="NotNull" name="SOF_TYPE"/>
  </response>
</testcase>
```

```

    <param assert_type="NotNull" name="SOF_REF1"/>
    <param assert_type="NotNull" name="SOF_REF2"/>
    <param assert_type="NotNull" name="DEBIT_TXN"/>
    <param assert_type="Null" name="L2_COMM"/>
    <param assert_type="EqualsOrMoreThan" expected="10" name="SOF_FEE"/>
  </response>
  <url>http://localhost:8080/counter/online/report/transaction/transactionID/1000479015</url>
</testcase>

```

รูปที่ 40 กรณีทดสอบ TEST-CASE-ID-001 ในรูปแบบเอกซ์เอ็มแอลแบบ GET

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<testcase>
  <id>TEST-CASE-ID-002</id>
  <method>POST</method>
  <name>UpdateTransaction</name>
  <request>
    <param name="transactionid" value="1000477826"/>
    <param name="status" value="commit"/>
  </request>
  <response>
    <param assert_type="Equals" expected="0" name="response_code"/>
    <param assert_type="NotNull" name="response_desc"/>
    <param assert_type="NotNull" name="tr_desc"/>
    <param assert_type="True" expected="0" name="conf_instance"/>
    <param assert_type="NotEquals" expected="-1" name="km.sessionid"/>
  </response>
  <url>http://localhost:8080/counter/report/transaction/</url>
</testcase>

```

รูปที่ 41 กรณีทดสอบ TEST-CASE-ID-001 ในรูปแบบเอกซ์เอ็มแอลแบบ POST

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<testcase>
  <id>TEST-CASE-ID-003</id>
  <method>PUT</method>
  <name>AdjustTransaction</name>
  <request>
    <param name="transactionid" value="1000477861"/>
    <param name="adjust" value="true"/>
  </request>
  <response>
    <param assert_type="Equals" expected="0" name="response_code"/>
    <param assert_type="NotNull" name="response_desc"/>
    <param assert_type="That" expected="0" name="km.sessionid"/>
      <param assert_type="False" name="km.wait"/>
      <param assert_type="NotEquals" name="token"/>
      <param assert_type="NotEquals" name="initiator"/>
      <param assert_type="NotEquals" name="pin"/>
      <param assert_type="NotEquals" name="cashto"/>
      <param assert_type="NotEquals" name="result_namespace"/>
      <param assert_type="NotEquals" name="vtrid"/>
  </response>
  <url>http://localhost:8080/counter/report/transaction/</url>
</testcase>

```

รูปที่ 42 กรณีทดสอบ TEST-CASE-ID-001 ในรูปแบบเอกซ์เอ็มแอลแบบ PUT

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<testcase>
  <id>TEST-CASE-ID-004</id>
  <method>DELETE</method>
  <name>CancelTransaction</name>
  <response>
    <param assert_type="Equals" expected="0" name="response_code"/>
    <param assert_type="NotNull" name="response_desc"/>
    <param assert_type="NotNull" name="EXT_ID"/>
      <param assert_type="True" expected="0" name="IOTYPE"/>
      <param assert_type="NotNull" name="IOMSG"/>
      <param assert_type="NotNull" name="CREATED"/>
      <param assert_type="NotNull" name="DKEY"/>
      <param assert_type="NotNull" name="DVAL"/>
      <param assert_type="NotNull" name="EXT_DESC"/>
  </response>

```

```

        <param assert_type="NotEquals" expected="-1" name="EXT_LOGIN"/>
    </response>
    <url>http://localhost:8080/counter/report/transaction/1000477861/Cancel</url>
</testcase>

```

รูปที่ 43 กรณีทดสอบ TEST-CASE-ID-001 ในรูปแบบเอกซ์เอ็มแอลแบบ DELETE

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<testsuite>
  <id>TransactionReport</id>
  <sequence>
    <testcase>
      <src>TEST-CASE-ID-001.xml</src>
    </testcase>
    <testcase>
      <src>TEST-CASE-ID-002.xml</src>
    </testcase>
    <testcase>
      <src>TEST-CASE-ID-003.xml</src>
    </testcase>
    <testcase>
      <src>TEST-CASE-ID-004.xml</src>
    </testcase>
  </sequence>
</testsuite>

```

รูปที่ 44 ลำดับกรณีทดสอบ

1.3 สคริปต์ทดสอบในรูปแบบ JUnit source code ดังรูปที่ 45

```

package th.ac.cu.testsuite;

import java.util.HashMap;
import java.util.Map;

import org.junit.Test;

import th.ac.cu.util.RESTClientUtils;
import th.ac.cu.util.TestCaseAssertionUtils;

public class BillPaymentTests {

    // TEST-CASE-ID-001

```



```

@Test
public void testInquiryTransactionSuccess() {

    Map<String, String> responseDatas = RESTClientUtils.callTo("GET",
"http://localhost:8080/counter/online/report/transaction");

    TestCaseAssertionUtils.assertEquals("trans_id", responseDatas.get("1000479015"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("account_no"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("terminal_id"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("svc_id"));
    TestCaseAssertionUtils.assertMoreThan("0", responseDatas.get("state"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("code"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("namespace"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("pay_amount"));
    TestCaseAssertionUtils.EqualsOrLessThan("10", responseDatas.get("svc_fee"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("service_code"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("pay_ref1"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("pay_txn"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("sof_type"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("sof_ref1"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("sof_ref2"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("debit_txn"));
    TestCaseAssertionUtils.assertNull(responseDatas.get("l2_comm"));
    TestCaseAssertionUtils.assertEqualsOrMoreThan("10", responseDatas.get("sof_fee"));
}

// TEST-CASE-ID-002
@Test
public void testUpdateTransactionSuccess() {

    Map<String, String> requestDatas = new HashMap<String, String>();
    requestDatas.put("transactionid", "1000477826");
    requestDatas.put("status", "commit");

    Map<String, String> responseDatas = RESTClientUtils.callTo("POST",
"http://localhost:8080/counter/report/transaction", requestDatas);

    TestCaseAssertionUtils.assertEquals("0", responseDatas.get("response_code"));
    TestCaseAssertionUtils.assertNotNull(responseDatas.get("response_desc"));

```

```

        TestCaseAssertionUtils.assertNotNull(responseDatas.get("tr_desc"));
        TestCaseAssertionUtils.assertTrue("0".equals(responseDatas.get("conf_instance")));
        TestCaseAssertionUtils.assertEquals("-1", responseDatas.get("km.sessionid"));
    }

    // TEST-CASE-ID-003
    @Test
    public void testInquiryPaymentSuccess() {

        Map<String, String> requestDatas = new HashMap<String, String>();
        requestDatas.put("transactionid", "1000477826");
        requestDatas.put("adjust", "True");

        Map<String, String> responseDatas = RESTClientUtils.callTo("PUT",
"http://localhost:8080/counter/report/transaction/");

        TestCaseAssertionUtils.assertEquals("0", responseDatas.get("response_code"));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("response_desc"));
        TestCaseAssertionUtils.assertThat("0", responseDatas.get("km.sessionid"));
        TestCaseAssertionUtils.assertFalse("0".equals(responseDatas.get("km.wait")));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("token"));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("initiator"));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("pin"));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("cashto"));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("result_namespace"));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("vtrid"));
    }

    // TEST-CASE-ID-004
    @Test
    public void testCancelTransaction() {

        Map<String, String> requestDatas = new HashMap<String, String>();
        Map<String, String> responseDatas = RESTClientUtils.callTo("DELETE",
"http://localhost:8080/counter/report/transaction/1000477861/Cancel", requestDatas);

        TestCaseAssertionUtils.assertEquals("0", responseDatas.get("response_code"));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("response_desc"));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("ext_id"));
        TestCaseAssertionUtils.assertTrue("0".equals(responseDatas.get("iotype")));
    }

```

```

        TestCaseAssertionUtils.assertNotNull(responseDatas.get("iomse"));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("created"));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("dkey"));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("dval"));
        TestCaseAssertionUtils.assertNotNull(responseDatas.get("ext_desc"));
        TestCaseAssertionUtils.assertNotEquals("-1", responseDatas.get("ext_login"));
    }
}

```

รูปที่ 45 สคริปต์ทดสอบ

1.4 ผลลัพธ์ของการทดสอบเรสตร์พูลเว็บเซอร์วิสเพื่อทดสอบเรียบร้อยแล้ว ที่ระบบทำการสร้างค่าผลลัพธ์ลงในไฟล์ลำดับทดสอบ ในรูปแบบเอกซ์เอ็มแอล ดังรูปที่ 46

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<testsuite>
  <id>TransactionReport</id>
  <sequence>
    <testcase>
      <result>PASS</result>
      <src>TEST-CASE-ID-001.xml</src>
    </testcase>
    <testcase>
      <result>PASS</result>
      <src>TEST-CASE-ID-002.xml</src>
    </testcase>
    <testcase>
      <result>PASS</result>
      <src>TEST-CASE-ID-003.xml</src>
    </testcase>
    <testcase>
      <result>PASS</result>
      <src>TEST-CASE-ID-004.xml</src>
    </testcase>
  </sequence>
</testsuite>

```

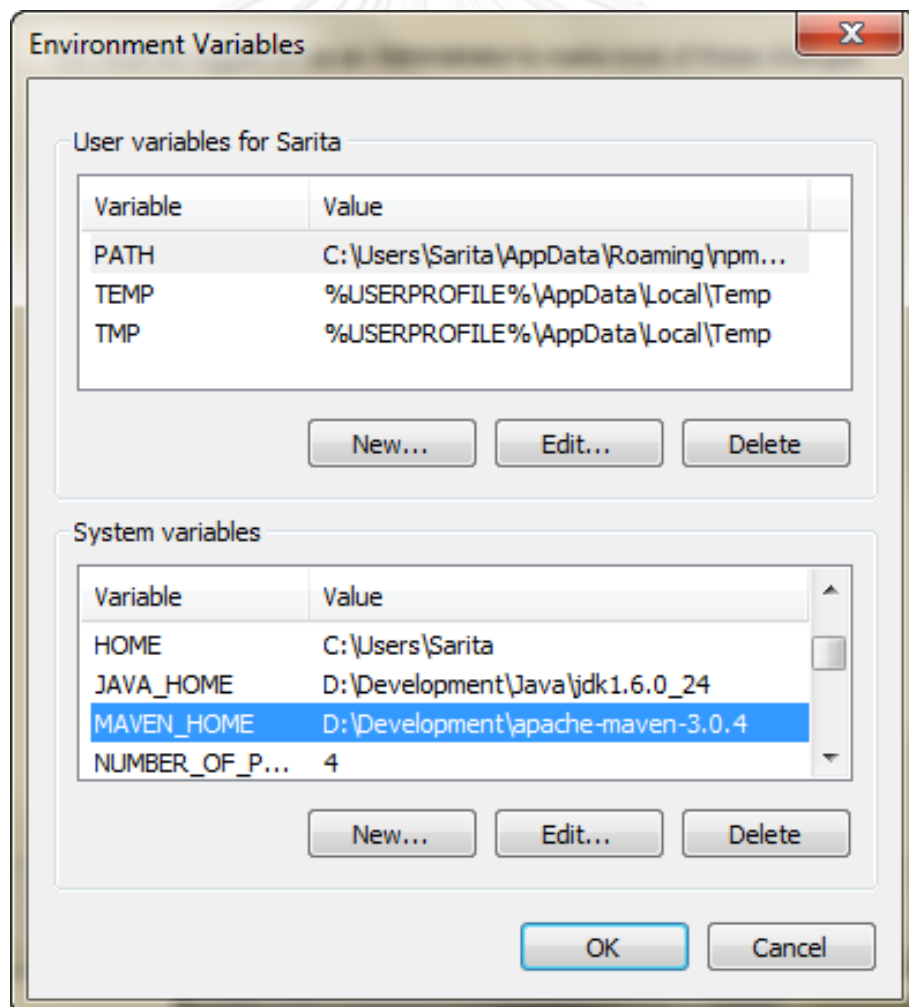
รูปที่ 46 ผลลัพธ์ในไฟล์ลำดับกรณีทดสอบ

ภาคผนวก ข

การติดตั้งเครื่องมือและสร้างโปรเจกอะปาเช่ มาเว่น เวอร์ชัน 3.2.2

อะปาเช่ มาเว่น เวอร์ชัน 3.2.2 (Apache Maven 3.2.2) เป็นเครื่องมือในการจัดการซอฟต์แวร์ และ Library ต่างๆ ในรูปแบบของวัตถุ โดยระบุกำหนดค่าในไฟล์ pom.xml ความสามารถของ มาเว่น ยังสามารถสร้างบีบอัดไฟล์ (Package) ได้อีกด้วย

1. การติดตั้งอะปาเช่ มาเว่น
 - 1.1 ดาวน์โหลดอะปาเช่ มาเว่น เวอร์ชัน 3.2.2
 - 1.2 หลังจาก download maven มาเรียบร้อยแล้วก็ทำการ set env :
MAVEN_HOME และ PATH ดังรูปที่ 47
 - 1.3 ตรวจสอบหลังการติดตั้ง “mvn -version” ดังรูปที่ 48



รูปที่ 47 การกำหนดค่า MAVEN_HOME

```

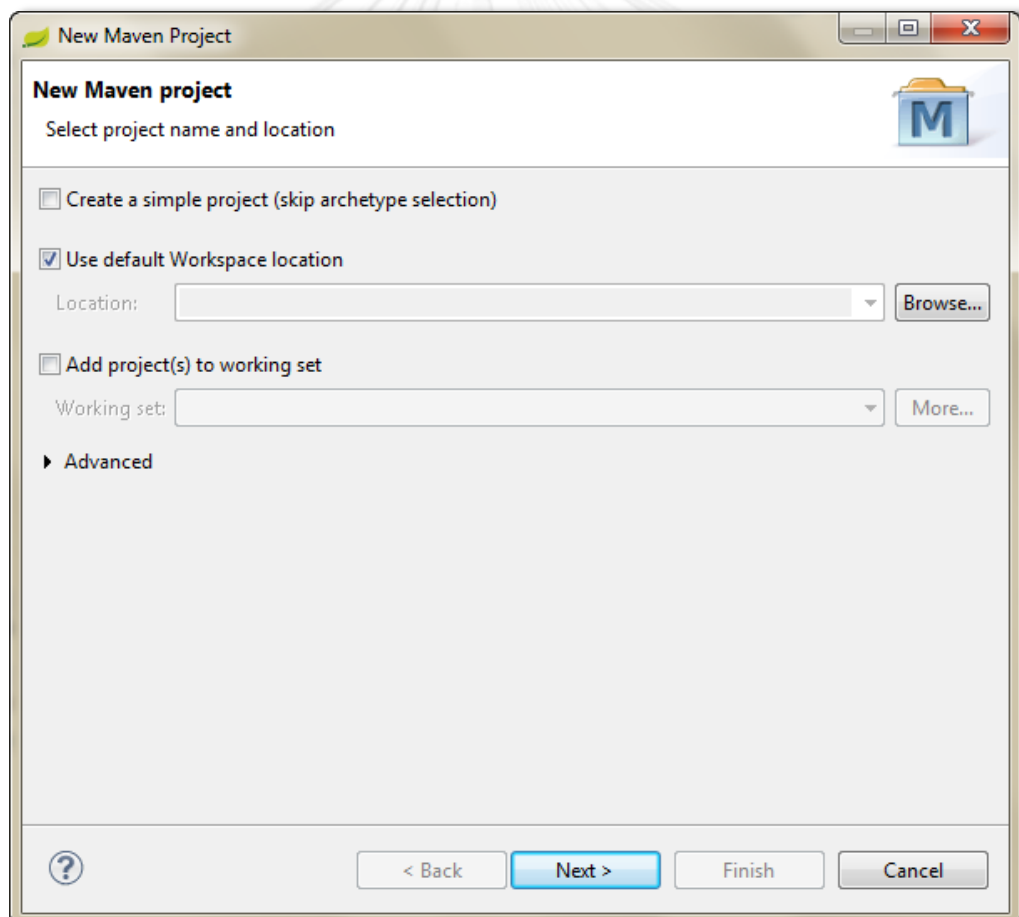
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Sarita>mvn -version
Apache Maven 3.0.4 (r1232337; 2555-01-17 15:44:56+0700)
Maven home: D:\Development\apache-maven-3.0.4\bin\..
Java version: 1.6.0_24, vendor: Sun Microsystems Inc.
Java home: D:\Development\Java\jdk1.6.0_24\jre
Default locale: th_TH, platform encoding: MS874
OS name: "windows 7", version: "6.1", arch: "x86", family: "windows"
C:\Users\Sarita>

```

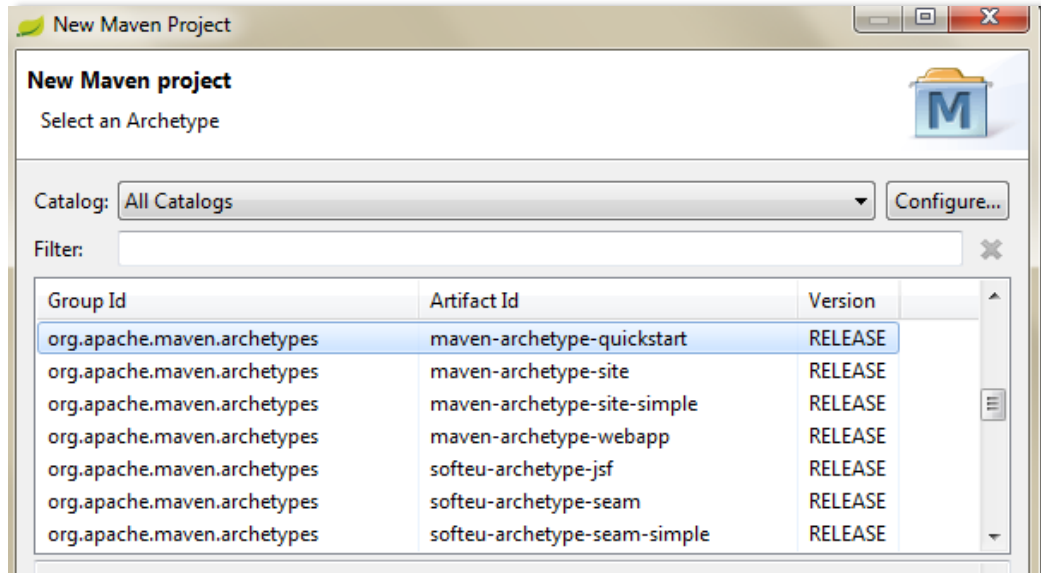
รูปที่ 48 หน้าจอแสดงเวอร์ชัน

2. การสร้างอะปาเช่ มาเว่น โปรเจคโดยใช้เครื่องมือสปริง ทูล สวีท เวอร์ชัน 3.2.2
 - 2.1 เลือก File > New > Maven Project ดังรูปที่ 49



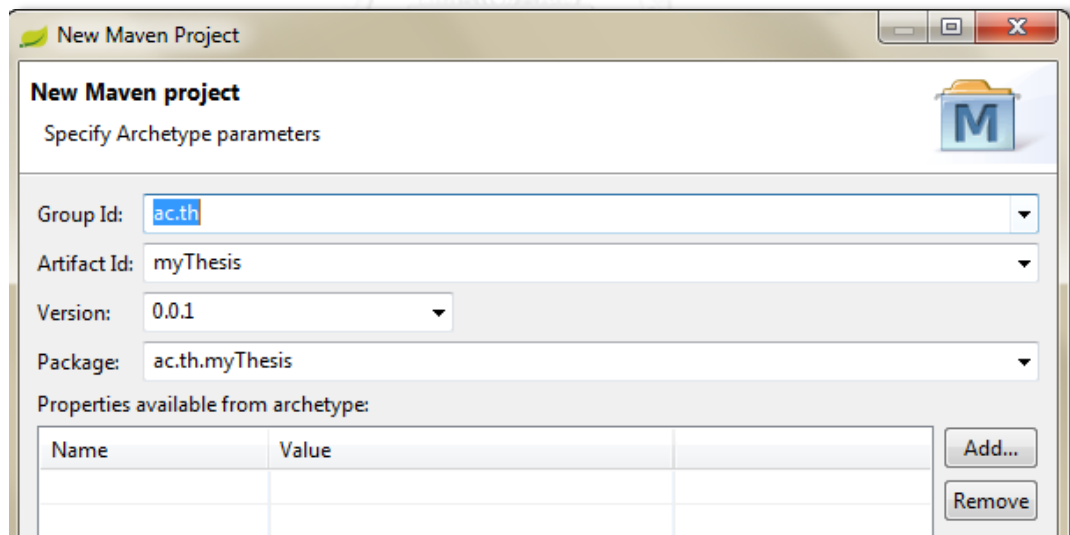
รูปที่ 49 หน้าจอสร้างมาเว่น โปรเจค

2.2 เลือก New Maven project เลือก Catalog เป็น All Catalogs > Next
 ดังรูปที่ 50



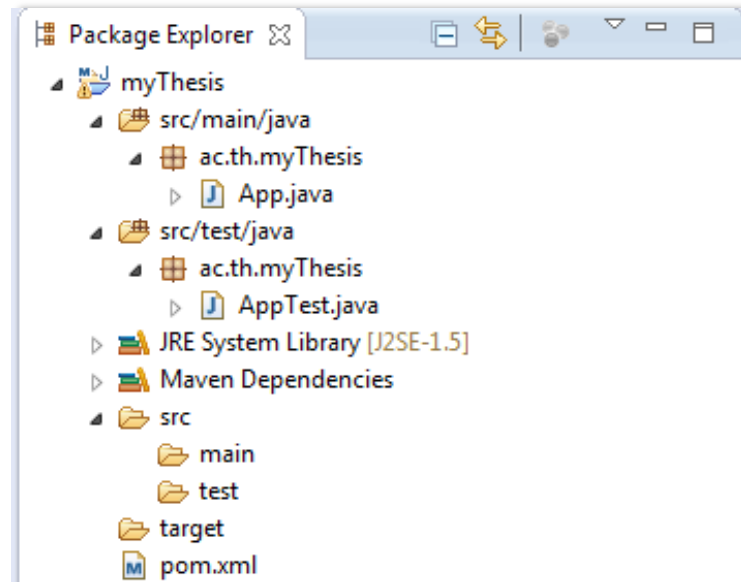
รูปที่ 50 หน้าจอเลือกส่วนประกอบ

2.3 ตั้งชื่อที่ Group Id และ ArtifactId > Finish ดังรูปที่ 51



รูปที่ 51 กำหนดชื่อมาเว่น โปรเจค และไฟล์นำออก

2.4 Maven project จะไปแสดงที่ Package Explorer ดังรูปที่ 52



รูปที่ 52 โครงสร้างมาเว่น โปรเจค

2.5 แสดงไฟล์ pom.xml สำหรับกำหนด Library ที่ต้องการ ดังรูปที่ 53

```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://maven.apache.org/xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsi:schemaLocation="
  <modelVersion>4.0.0</modelVersion>
  <groupId>th.ac.cu</groupId>
  <artifactId>BillPayment</artifactId>
  <version>20140613220000</version>
  <packaging>war</packaging>
  <name>BillPayment</name>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <plugins>

```

รูปที่ 53 ไฟล์ pom.xml

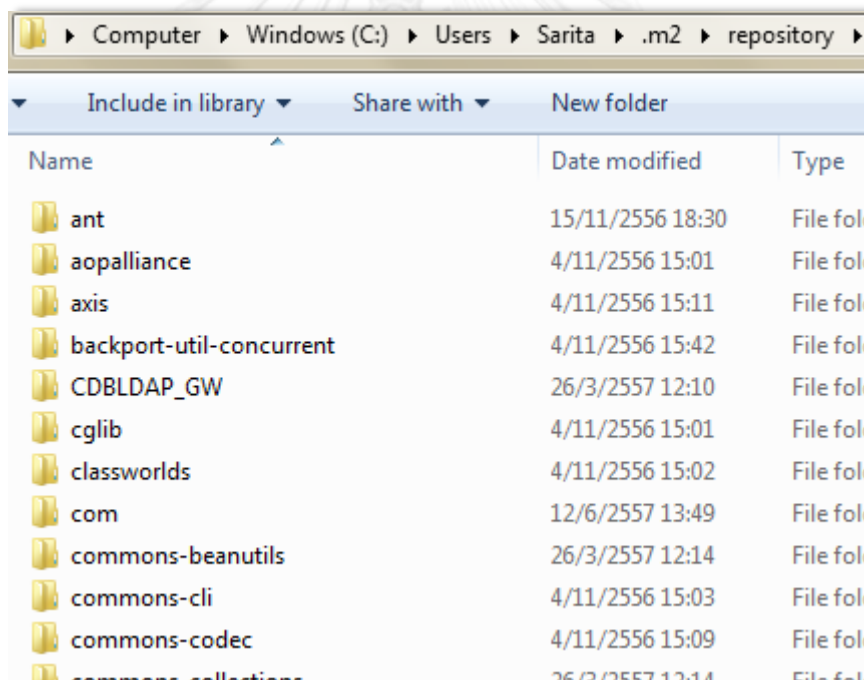
2.6 เปิดหน้าต่าง Command Prompt พิมพ์คำสั่ง “mvn install” อะปาเซ มาเว่นจะทำการดาวน์โหลด Library ดังรูปที่ 54 ที่กำหนดค่าไว้ใน pom.xml ไปเก็บไว้ที่ C:\Users\MyUsers\Downloads โดยอัตโนมัติ ดังรูปที่ 46

```

MINGW32/D:/test-restful-webservices/20140613220000/MVN Artifact/BillPayment
Artifact\BillPayment\src\main\webapp]
[INFO] Webapp assembled in [48 msecs]
[INFO] Building war: d:\test-restful-webservices\20140613220000\MVN Artifact\BillPayment\target\BillPayment-20140613220000.war
[WARNING] Warning: selected war files include a WEB-INF/web.xml which will be ignored (webxml attribute is missing from war task, or ignoreWebxml attribute is specified as 'true')
[INFO]
[INFO] --- maven-install-plugin:2.3.1:install (default-install) @ BillPayment ---
[INFO] Installing d:\test-restful-webservices\20140613220000\MVN Artifact\BillPayment\target\BillPayment-20140613220000.war to C:\Users\Sarita\.m2\repository\th\ac\cu\BillPayment\20140613220000\BillPayment-20140613220000.war
[INFO] Installing d:\test-restful-webservices\20140613220000\MVN Artifact\BillPayment\pom.xml to C:\Users\Sarita\.m2\repository\th\ac\cu\BillPayment\20140613220000\BillPayment-20140613220000.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.049s
[INFO] Finished at: Tue Jul 15 20:47:38 ICT 2014
[INFO] Final Memory: 5M/15M
[INFO] -----

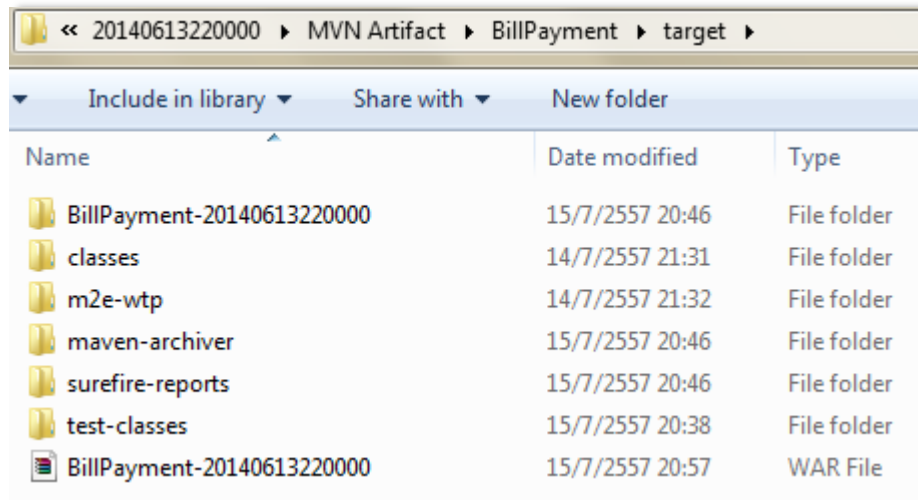
```

รูปที่ 54 ผลลัพธ์การดาวน์โหลด Library ต่างๆ



รูปที่ 55 แฟ้มเอกสาร Library

2.7 เปิดหน้าต่าง Command Prompt พิมพ์คำสั่ง “mvn package” อะปาเซ่ มาเว่นจะทำ บีบอัดไฟล์รวมถึงดาวน์โหลด Library ที่กำหนดค่าไว้ใน pom.xml ไปเก็บไว้ที่ D:\test-restful-webservices\20140613220000\MVN Artifact\BillPayment\target โดยอัตโนมัติ ดังรูปที่ 56



Name	Date modified	Type
BillPayment-20140613220000	15/7/2557 20:46	File folder
classes	14/7/2557 21:31	File folder
m2e-wtp	14/7/2557 21:32	File folder
maven-archiver	15/7/2557 20:46	File folder
surefire-reports	15/7/2557 20:46	File folder
test-classes	15/7/2557 20:38	File folder
BillPayment-20140613220000	15/7/2557 20:57	WAR File

รูปที่ 56 แฟ้มเอกสารไฟล์ที่บีบอัด

ภาคผนวก ค

การติดตั้งเครื่องมืออะปาเช่ ทอมแคท เวอร์ชัน 6.0.41 และติดตั้งระบบทดสอบ

เรสท์ฟูลเว็บเซอร์วิสในการพัฒนาแบบเอไจล์

อะปาเช่ ทอมแคท เวอร์ชัน 6.0.41 (Apache Tomcat 6.0.41) เป็นเครื่องมือประเภทผู้ให้บริการ (Server) ในการประมวลผลซอฟต์แวร์ และติดตั้งแอปพลิเคชัน ข้อดีคือ ทำงานเร็วสามารถรองรับผู้ใช้งานได้มาก ตั้งค่าได้ง่าย สามารถทำ Load balancing Clustering ได้สะดวก และไม่เสียค่าใช้จ่าย

1. การติดตั้งอะปาเช่ ทอมแคท

1.1 ดาวนโหลดอะปาเช่ ทอมแคท เวอร์ชัน 6.0.41

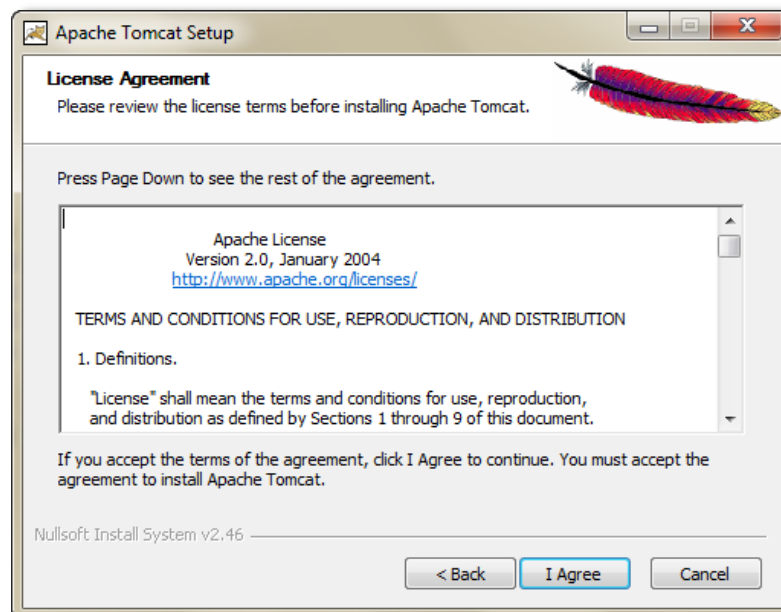
1.2 คลิก tomcat.exe แสดงหน้าจอเริ่มต้นอะปาเช่ ทอมแคท เลือก Next ดังรูปที่

57



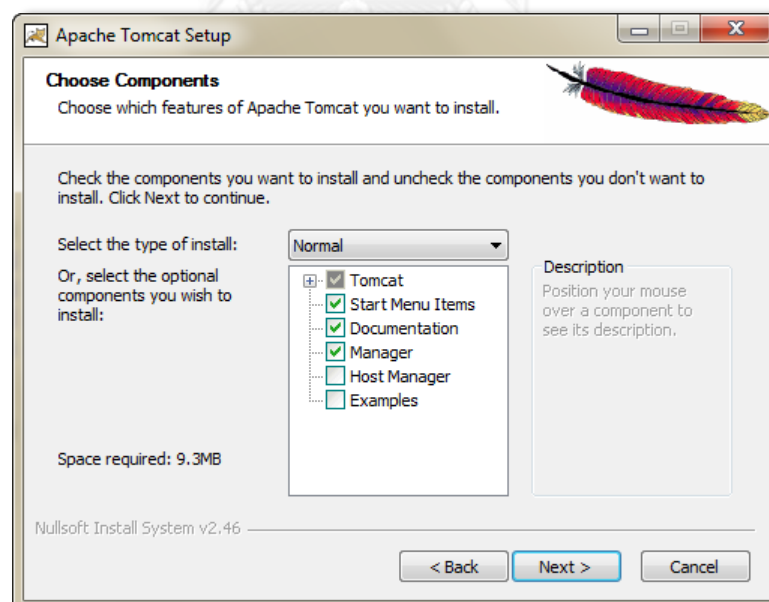
รูปที่ 57 หน้าจอเริ่มต้นอะปาเช่ ทอมแคท เวอร์ชัน 6.0.41

1.3 แสดงหน้าต่าง License Agreement เลือก I Agree



รูปที่ 58 หน้าจอแสดงข้อตกลง

1.4 Choose Components เลือก Tomcat, Start Menu Items, Documentation, Manager ตั้งรูปที่ 59 เลือก Next



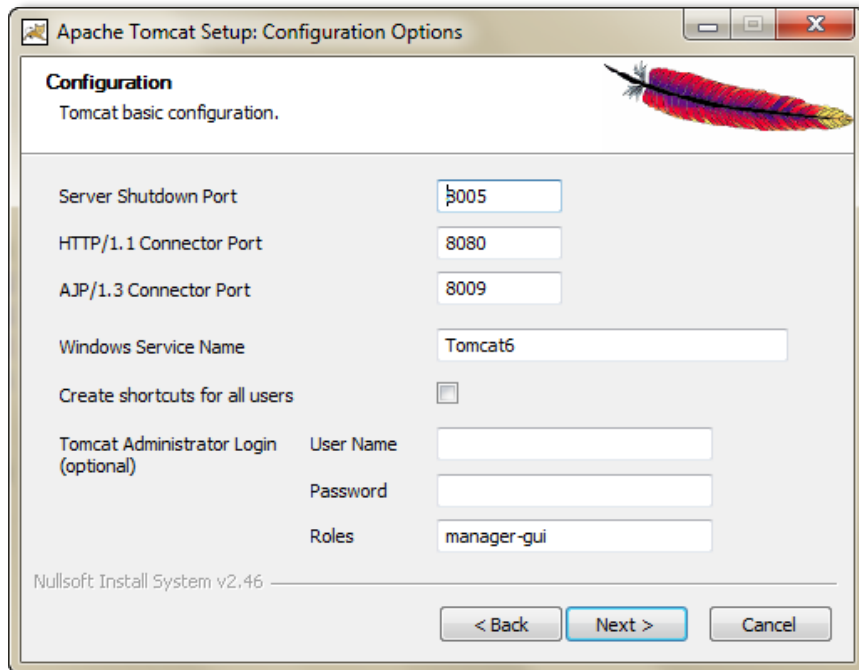
รูปที่ 59 หน้าจอการเลือกส่วนประกอบ

1.5 Configuration

- Server Shutdown Port เป็น 8005
- HTTP/1.1 Connector Port เป็น 8080

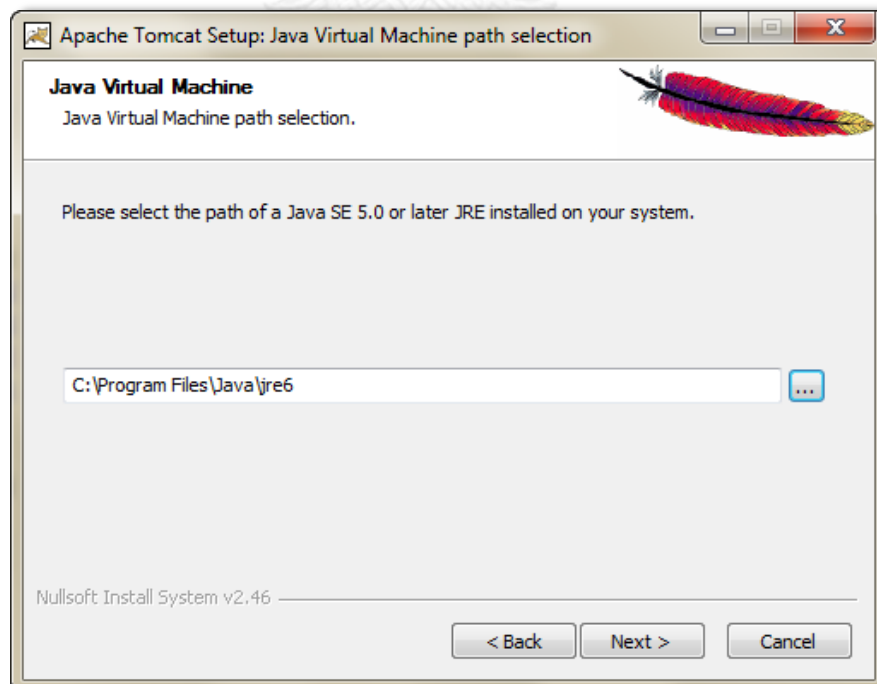
- AJP/1.3 Connector Port เป็น 8009

เลือก Next ดังรูปที่ 60



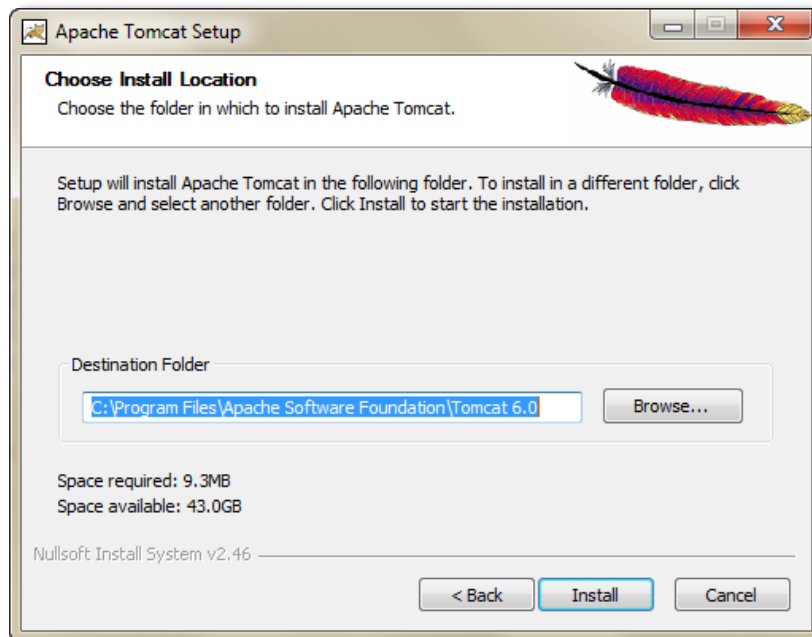
รูปที่ 60 หน้าจอสำหรับตั้งค่า

1.6 เลือก Java Virtual Machine เป็น jre6 เลือก Next ดังรูปที่ 61



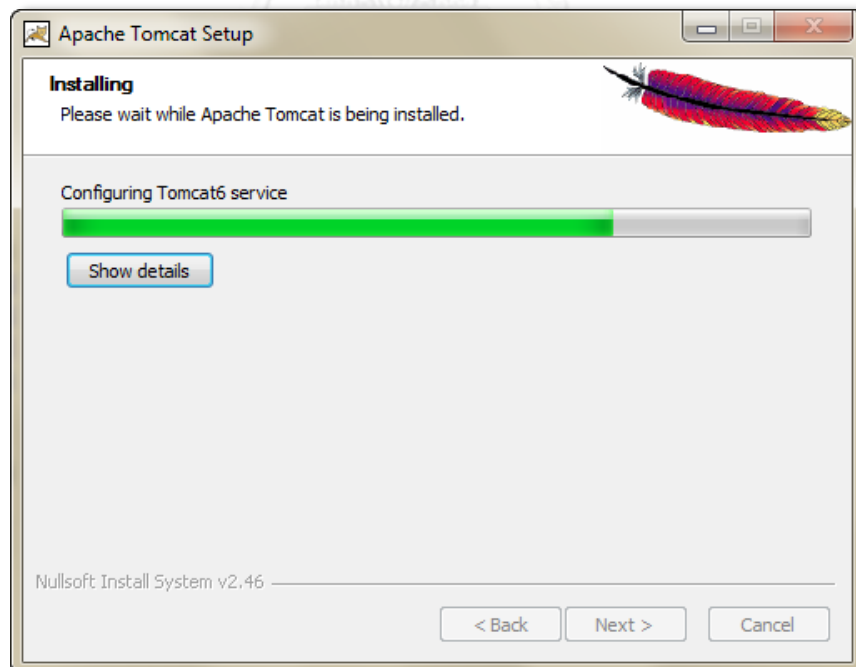
รูปที่ 61 หน้าจอการเลือกเพิ่มปลายทางของ Java Virtual Machine

1.7 เลือก Destination Folder เป็นเพิ่มปลายทางที่ต้องการจะติดตั้ง ดังรูปที่ 62



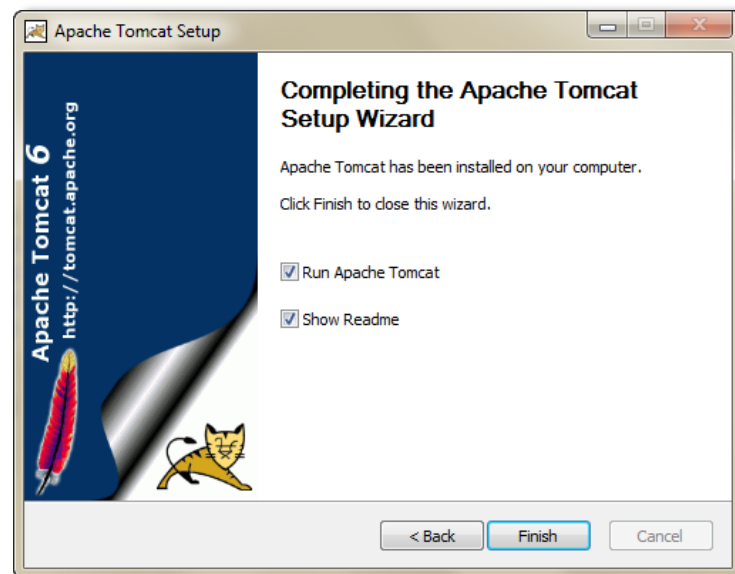
รูปที่ 62 หน้าจอการเลือกสถานที่ที่จะติดตั้ง

1.8 หน้าต่างแสดงสถานะระหว่างติดตั้ง



รูปที่ 63 สถานะระหว่างติดตั้ง

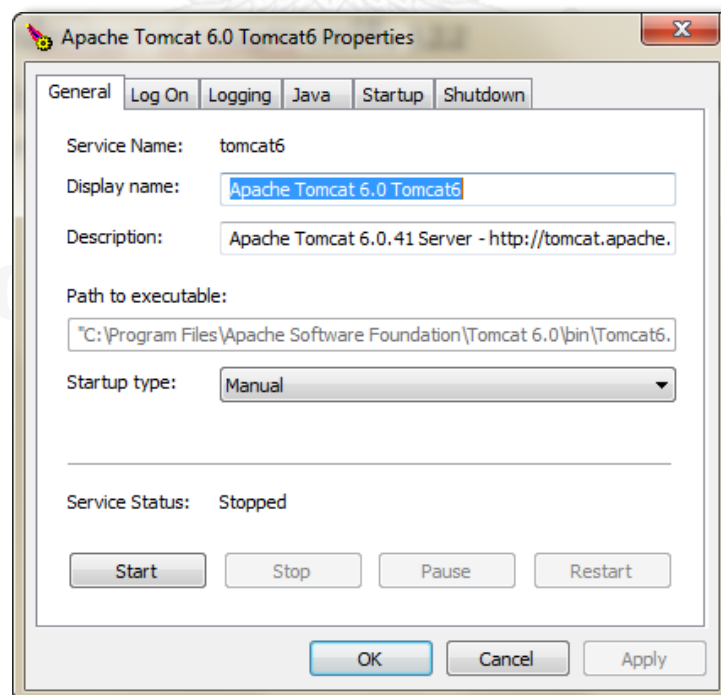
1.9 แสดงหน้าต่างการติดตั้งเสร็จสมบูรณ์



รูปที่ 64 การติดตั้งอะปาเช่ ทอมแคท เสร็จสมบูรณ์

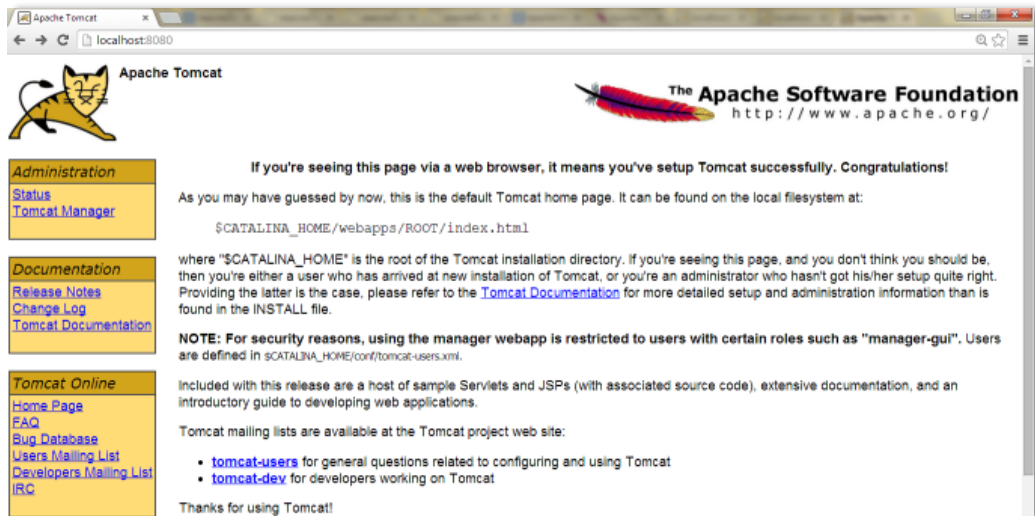
2. ขั้นตอนการติดตั้งแอปพลิเคชัน ระบบทดสอบเรสตร์พูลเว็บเซอร์วิสในการพัฒนาแบบ เอไจล์

2.1 เข้าไปที่แฟ้มเอกสาร apache-tomcat-6.0.41\bin > tomcat6w.exe > Start ดังรูปที่ 65



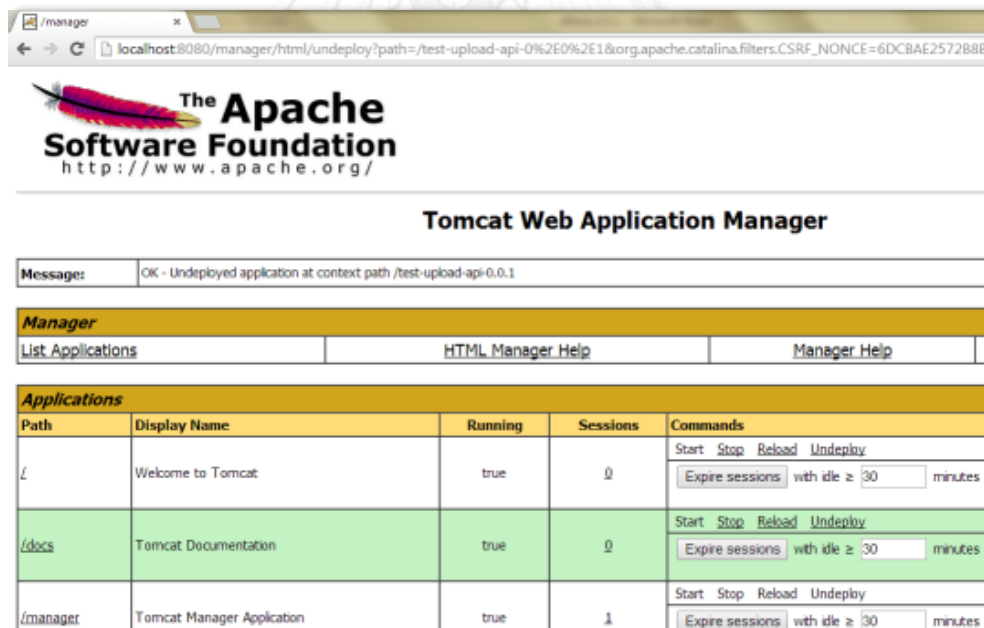
รูปที่ 65 หน้าจอคุณลักษณะอะปาเช่ ทอมแคท

2.2 เข้าไปเว็บเบราว์เซอร์ http://localhost:8080/ ดังรูปที่ 66



รูปที่ 66 หน้าจอหลักการจัดการทอมแคท

2.3 เลือก Tomcat Manager ดังรูปที่ 67



รูปที่ 67 หน้าจอหลักการจัดการแอปพลิเคชัน

2.4 เลือกไปยังไฟล์บีบอัด test-upload-api-0.0.1 ดังรูปที่ 68

Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

Deploy

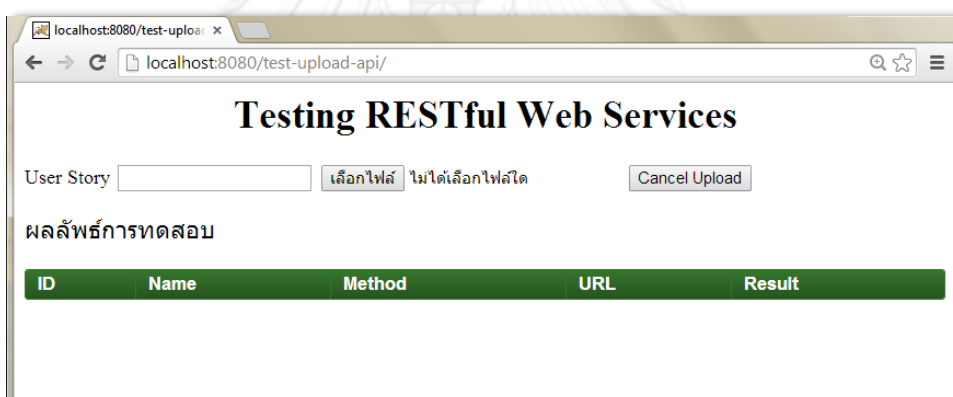
WAR file to deploy

Select WAR file to upload test-upload-api-0.0.1.war

Deploy

รูปที่ 68 หน้าจอการเลือกไฟล์เพื่อติดตั้งแอปพลิเคชัน

2.5 หลังจากติดตั้งสำเร็จ เข้าไปเว็บเบราว์เซอร์ <http://localhost:8080/test-upload-api/> เพื่อเข้าหน้าระบบทดสอบเรสตฟูลเว็บเซอร์วิสในการพัฒนาแบบเอจิล์ ดังรูปที่ 69



รูปที่ 69 หน้าจอการทำงานหลักของระบบ

ประวัติผู้เขียนวิทยานิพนธ์

นางสาว สรिता คุปตยานนท์ เกิดเมื่อวันที่ 11 พฤศจิกายน พ.ศ. 2530 ที่จังหวัดอุดรธานี สำเร็จการศึกษาปริญญาตรีหลักสูตรบริหารธุรกิจบัณฑิต (บธ.บ.) สาขาวิชาเทคโนโลยีสารสนเทศ ธุรกิจ ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยธุรกิจบัณฑิตย์ ในปีการศึกษา 2551 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2555



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY