



การทดสอบประสิทธิภาพการใช้งานเว็บแอปพลิเคชัน

กรณีศึกษา บริษัท เมโทรซิสเต็มส์คอร์ปอเรชัน จำกัด (มหาชน) พ.ศ. 2559

PERFORMANCE LOAD TESTING ON WEB APPLICATION

A CASE STUDY OF METRO SYSTEM CORPORATION CO, LTD.

นาย ธนวัต ชูดีไกรวัล

โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

พ.ศ. 2559

การทดสอบประสิทธิภาพการใช้งานเว็บแอปพลิเคชัน
กรณีศึกษา บริษัท เมโทรซิสเต็มส์คอร์ปอเรชัน จำกัด (มหาชน)
PERFORMANCE LOAD TESTING ON WEB APPLICATION
CASE STUDY METRO SYSTEM CORPORATION CO LTD.

นาย ธนวัต ชุตีไกรวัล

โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรบัณฑิต สาขาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีไทย - ญี่ปุ่น
ปีการศึกษา 2559

คณะกรรมการสอบ

..... ประธานกรรมการสอบ
(ผศ.ดร. อรรณพ หมั่นสกุล)

..... กรรมการสอบ
(อ.ลลิตา ณ หนองคาย)

..... อาจารย์ที่ปรึกษา
(อ.อมรพันธ์ ชมกลีน)

..... ประธานสหกิจศึกษาสาขาวิชา
(อ.อมรพันธ์ ชมกลีน)

ลิขสิทธิ์ของสถาบันเทคโนโลยีไทย - ญี่ปุ่น

ชื่อโครงการ

การทดสอบประสิทธิภาพการใช้งานเว็บแอปพลิเคชันการศึกษา บริษัท

เมโทรซิสเต็มส์คอร์ปอเรชัน จำกัด (มหาชน) พ.ศ. 2559

Performance Load Testing on Web Application

Case Study Metro System Corporation Co Ltd.

ผู้เขียน

นาย ธนวัต ชูดีไกรวัล

คณะวิชา

เทคโนโลยีสารสนเทศ

สาขาวิชา เทคโนโลยีสารสนเทศ

อาจารย์ที่ปรึกษา

อาจารย์ อมรพันธุ์ ชมกลิ่น

พนักงานที่ปรึกษา

นาย วิชาญ โสดาธาตุ

ชื่อบริษัท

เมโทรซิสเต็มส์คอร์ปอเรชัน จำกัด (มหาชน)

ประเภทธุรกิจ/สินค้า

Web Application, Software Application

บทสรุป

หน้าที่ที่ได้รับมอบหมายที่การทำสภากิจครั้งนี้คือการทดสอบประสิทธิภาพการใช้งานของเว็บแอปพลิเคชันที่บริษัทกำลังทำการพัฒนาอยู่ เพื่อให้เว็บเซิร์ฟเวอร์สามารถรองรับการเข้าใช้งานของผู้ใช้งานระบบในจำนวนที่มากได้ และเพื่อให้ระบบมีประสิทธิภาพการใช้งานที่คงที่ ไม่พบปัญหาในการเข้าใช้งานเป็นระยะเวลาต่อเนื่องที่มีการเข้าใช้งาน

ในการทำการทดสอบประสิทธิภาพ ผลที่ได้นั้นทำให้เราได้รู้ถึงจำนวนจำกัดในการเข้าใช้งานของผู้ใช้งาน ระยะเวลาในการทำรายการภายในแอปพลิเคชัน และประสิทธิภาพในการทำงานของเซิร์ฟเวอร์ ซึ่งเราสามารถนำผลการทดสอบที่ได้นั้นไปแก้ไขปัญหที่เกิดขึ้นระหว่างการทำการทดสอบได้ไม่ว่าจะเป็นการพัฒนาเว็บแอปพลิเคชันในส่วนที่ใช้งานแล้วติดขัด เพิ่มจำนวนเซิร์ฟเวอร์เพื่อให้เพียงพอต่อการใช้งานของผู้ใช้งาน เป็นต้น

กิตติกรรมประกาศ

จากการที่ผมได้ไปสหกิจศึกษาที่ ณ บริษัท เมโทรซิสเต็มส์คอร์ปอเรชันจำกัด (มหาชน) นับตั้งแต่วันที่ 30 พฤษภาคม พ.ศ. 2559 จนถึง 30 กันยายน พ.ศ. 2559 ทำให้ผมได้รับความรู้และประสบการณ์จากการทำงานจริงที่สำคัญต่อการพัฒนาตัวเองในอนาคต และสำหรับรายงานปฏิบัติงานสหกิจเล่มนี้ ผมขอกราบขอบคุณผู้ที่ให้ความรู้และสนับสนุนการฝึกสหกิจจนสำเร็จลุล่วงไปได้ด้วยดี ดังนี้

1. นาง อุไร สุขตระกูล (Production Manager) ผู้ที่ให้โอกาสผมเข้ามาทำสหกิจศึกษาภายในบริษัท เมโทรซิสเต็มส์คอร์ปอเรชันจำกัด(มหาชน) แผนก MIT
2. นาย ธวัชชัย สุขตระกูล (Production Manager) ผู้ให้การอบรมฝึกสอนตลอดการทำสหกิจศึกษา
3. นาย ธนากรณ สัมภ์ทอง (Scrum Master) ผู้เปิดโอกาสให้เข้าทีม QA1 และได้ให้ประสบการณ์ในการทำงานจริง
4. นาย วิชาญ โสดาธาตุ (Program Developer) ผู้ให้คำแนะนำและให้ความช่วยเหลือตลอดการทำสหกิจศึกษา

และสุดท้ายนี้ขอขอบคุณพี่ๆทุกท่านในบริษัทเมโทรซิสเต็มส์คอร์ปอเรชันจำกัด (มหาชน) ที่ให้ความช่วยเหลือและประสบการณ์ในการทำงานจริงตลอดการทำสหกิจศึกษา 4 เดือนที่ผ่านมาจนสำเร็จร่วมนไปได้ด้วยดี และผมจะนำสิ่งที่ได้จากการฝึกสอนทั้งหมดมาประยุกต์ใช้ในการทำงานในอนาคต

นาย ธนวัต ชูดีไกรวัล
ผู้จัดทำ

สารบัญ

หน้า

บทสรุป	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญรูป	ฅ
สารบัญตาราง	ช
บทที่ 1	1
1.1 ชื่อและที่ตั้งของสถานประกอบการ	1
1.2 ลักษณะธุรกิจของสถานประกอบการ	2
1.3 รูปแบบการจัดการองค์กรและการบริหารองค์กร	4
1.4 ตำแหน่งและหน้าที่งานที่ได้รับมอบหมาย	4
1.5 พนักงานที่ปรึกษา และตำแหน่งของพนักงานที่ปรึกษา	5
1.6 ระยะเวลาที่ปฏิบัติงาน	5
1.7 ที่มาและความสำคัญของปัญหา	5
1.8 วัตถุประสงค์หรือจุดมุ่งหมายของโครงการ	5
1.9 ผลที่คาดว่าจะได้รับการปฏิบัติงาน	6
บทที่ 2 ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน	7
2.1 เทคโนโลยีที่ใช้ในการปฏิบัติ	7
2.1.1 โปรแกรม Selenium WebDriver	7
2.1.1.1 Selenium Core	8
2.1.1.2 Selenium RC	9

สารบัญ(ต่อ)

บทที่	หน้า
2.1.1.3 Selenium Grid	10
2.1.1.4 Selenium IDE	10
2.1.1.5 WebDriver	11
2.1.1.6 Selenium 2	13
2.1.2 Java Lanaguge	13
2.1.3 Apache Jmeter	15
2.1.4 Spring Tool suite	16
บทที่ 3 แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน	17
3.1 แผนงานการปฏิบัติงาน	17
3.2 รายละเอียดที่นักศึกษาปฏิบัติในการฝึกงาน	17
3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงาน	18
บทที่ 4 สรุปผลการดำเนินงาน การวิเคราะห์และสรุปผลต่าง ๆ	26
4.1 ขั้นตอนและผลการดำเนินงาน	26
4.2 ผลการวิเคราะห์ข้อมูล	28
4.3 วิจัยข้อมูลโดยเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์และจุดมุ่งหมายการปฏิบัติงาน	29
บทที่ 5 บทสรุปและข้อเสนอแนะ	30
5.1 สรุปผลการดำเนินโครงการ	30
5.2 แนวทางการแก้ไขปัญหา	30
5.3 ข้อเสนอแนะจากการดำเนินงาน	31

สารบัญ(ต่อ)

บทที่

หน้า

ภาคผนวก

ก. รายงานประจำสัปดาห์

ประวัติผู้จัดทำโครงการ

53



สารบัญภาพ

ภาพที่	หน้า
ภาพที่ 1.1 สถานที่ตั้งบริษัท	1
ภาพที่ 1.2 รางวัลที่บริษัทได้รับ	3
ภาพที่ 1.3 คณะผู้บริหารบริษัท	4
ภาพที่ 2.1 เว็บเบราว์เซอร์ที่สามารถใช้ร่วมกับ WebDriver ได้	7
ภาพที่ 2.2 กลุ่มของเครื่องมือทดสอบที่ใช้ใน โปรแกรม Selenium ได้	8
ภาพที่ 2.3 แสดงการทำงานของ Selenium Grid	10
ภาพที่ 2.4 เปรียบเทียบการทำงานระหว่าง Selenium RC และ WebDriver	12
ภาพที่ 2.5 Java Programming	15
ภาพที่ 2.6 Spring Tool Suite Logo	16
ภาพที่ 3.1 หน้าเว็บไซต์ที่ทำการทดสอบ Login	19
ภาพที่ 3.2 หน้าเว็บไซต์ที่ทำการทดสอบ Logout	19
ภาพที่ 3.3 Selenium Coding	20
ภาพที่ 3.4 Selenium Coding	21
ภาพที่ 3.5 Selenium Coding	21
ภาพที่ 3.6 Selenium Coding	22
ภาพที่ 3.7 Selenium Coding	23
ภาพที่ 3.8 Coding Selenium	23
ภาพที่ 3.9 Jmeter JUnit Request	24
ภาพที่ 3.10 Jmeter HTTP Request	25
ภาพที่ 4.1 ผลการทดสอบ Performance Test ด้วย Jmeter	26
ภาพที่ 4.2 ผลการทดสอบ Load Test ด้วยโปรแกรม Jmeter	27
ภาพที่ 4.3 ผลการทดสอบ Load Test ด้วยโปรแกรม Jmeter	27
ภาพที่ 4.4 ผลการทดสอบ Load Test ด้วยโปรแกรม Jmeter	28

สารบัญตาราง

ตารางที่

หน้า

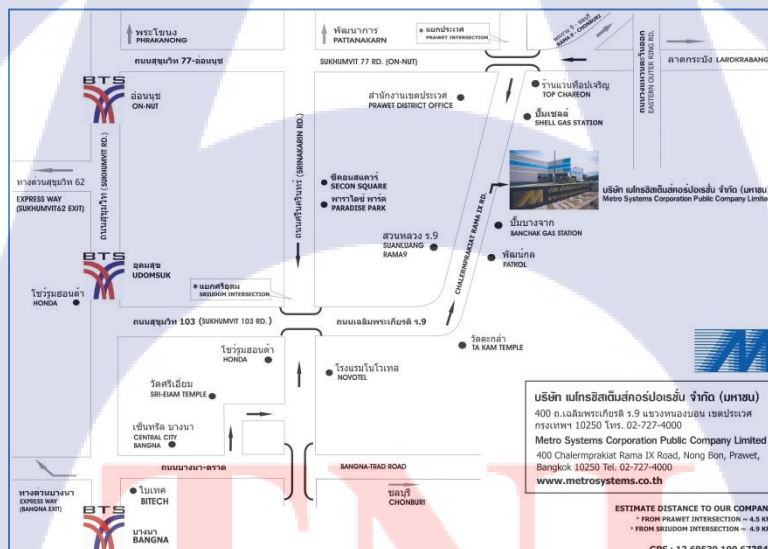
ตารางที่ 2.1	ข้อดีข้อเสียของการใช้งานร่วมกับ Selenium RC	11
ตารางที่ 2.3	ข้อดีและข้อเสียของการใช้งานร่วมกับ WebDriver	12
ตารางที่ 3.1	ตารางการปฏิบัติงาน	17



บทที่ 1 บทนำ

1.1 ชื่อและที่ตั้งของสถานประกอบการ

ชื่อหน่วยงาน	บริษัท เมโทรซิสเต็มส์คอร์ปอเรชัน จำกัด (มหาชน)
ที่ตั้ง	(Metro Systems Corporation Public Company Limited) เลขที่ 400 ถนนเฉลิมพระเกียรติ ร.9 แขวงหนองบอน เขตประเวศ กรุงเทพมหานคร 10250
เว็บไซต์	http://www.metrosystems.co.th
อีเมลล์	recruitment@metrosystems.co.th



ภาพที่ 1.1 สถานที่ตั้ง บริษัท เมโทรซิสเต็มส์คอร์ปอเรชัน จำกัด (มหาชน)

1.2 ลักษณะธุรกิจของสถานประกอบการ หรือการให้บริการหลักขององค์กร

บริษัท เมโทรซิสเต็มส์คอร์ปอเรชั่น จำกัด (มหาชน) ได้ก่อตั้งขึ้นในปี พ.ศ. 2529 ด้วยทุนจดทะเบียน 4 ล้านบาท เริ่มต้นดำเนินธุรกิจจากการเป็นตัวแทนจำหน่ายผลิตภัณฑ์ไอพีเอ็มเป็นรายแรกในภูมิภาคเอเชียตะวันออกเฉียงใต้ ต่อมาได้มีการขยายการจัดจำหน่ายโซลูชันและบริการจากบริษัทไอทีชั้นนำอื่น ๆ เพื่อการบริการที่ครบวงจรและครอบคลุมความต้องการของลูกค้าได้มากยิ่งขึ้น

ในปี 2538 บริษัทได้เพิ่มทุนจดทะเบียนเป็น 180 ล้านบาท เพื่อปรับเปลี่ยนรูปแบบธุรกิจโดยจดทะเบียนเป็นบริษัทมหาชนในตลาดหลักทรัพย์แห่งประเทศไทยและทำการซื้อขายหุ้นสามัญครั้งแรกเมื่อวันที่ 9 พฤษภาคม พ.ศ. 2539 ตลอดระยะเวลาการดำเนินงาน บริษัทมีการปรับเปลี่ยนมูลค่าทุนจดทะเบียนให้ตามความเหมาะสมกับขนาดของธุรกิจ โดย ณ ปัจจุบันบริษัทมีทุนจดทะเบียน 360 ล้านบาท

ภายหลังการได้รับรองระบบคุณภาพ ISO 9001: 2008 ในทุกขั้นตอนการทำงาน และระบบ ISO/IEC 20000:2005 และ ISO/IEC 27001:2005 ซึ่งเกี่ยวข้องกับการป้องกันสารสนเทศในองค์กรที่อาจส่งผลต่อการดำเนินงาน ทำให้ในปี 2554 ทางบริษัทได้ดำเนินจัดทำระบบบริหารความต่อเนื่องทางธุรกิจ ((Business Continuity Management System : BCMS) และกำหนดเพิ่มเติมไว้ในพันธกิจหลักขององค์กร กล่าวคือ “...การมุ่งมั่นสู่ความเป็นเลิศและบริหารความต่อเนื่องในธุรกิจเทคโนโลยีสารสนเทศครบวงจร โดยบุคลากรระดับมืออาชีพ เพื่อความสำเร็จของลูกค้า ตลอดจนมีส่วนร่วมในการพัฒนาสังคมแห่งความรู้ ทั้งนี้ เป็นการป้องกันบรรเทาอุบัติเหตุที่อาจก่อให้เกิดความเสียหายต่อธุรกิจ และทำให้บริษัทสามารถดำเนินธุรกิจได้อย่างต่อเนื่อง ซึ่งมุ่งเน้นการดูแลความปลอดภัย การปฏิบัติตามกฎหมาย กฎระเบียบ ข้อตกลงที่เกี่ยวข้อง ตลอดจนดูแลตรวจสอบความถูกต้องของข้อมูล ภาพลักษณ์และความน่าเชื่อถือของบริษัท

ด้วยแนวคิดที่ยึดเอาลูกค้าเป็นศูนย์กลาง บริษัทฯ จึงมุ่งแสวงหาแนวทางใหม่ๆ เพื่อตอบสนองความต้องการของลูกค้า และยังคงเดินหน้าแสวงหาผลิตภัณฑ์และบริการต่างๆ เพื่อให้ทันกับการเปลี่ยนแปลงของอุตสาหกรรมไอที การมุ่งแสวงหาบริการที่จะทำให้ลูกค้าดำเนินธุรกิจได้อย่างมีประสิทธิภาพและต่อเนื่องทั้งในสถานะปกติ หรือในยามเผชิญวิกฤต และยังให้ความสำคัญในการการร่วมมือกับพันธมิตรธุรกิจเพื่อนำเสนอโซลูชันให้ครอบคลุมความต้องการของกลุ่มลูกค้า บริษัทฯ จะยังคงเดินหน้าพัฒนาบุคลากร และผสมผสานความสามารถของซัพพลายเออร์ คู่ค้า ตลอดจนนวัตกรรมใหม่ๆ ที่เกี่ยวข้อง เพื่อตอบสนองความต้องการ สร้างความสำเร็จ และสร้างคุณค่าเพิ่มให้แก่ธุรกิจของลูกค้า

จากการเติบโตทางธุรกิจอย่างรวดเร็ว ทำให้บริษัท เมโทรซิสเต็มส์ คอร์ปอเรชั่น จำกัด(มหาชน) ได้รับมอบรางวัล ” IBM Business Partner of the Year” ซึ่งเป็นรางวัลที่ได้รับติดต่อกันเป็นเวลาถึง 25 ปี พร้อมทั้งได้รับรางวัล IBM The Longest Relation Business Partner, Citrix - Platinum Partner of the Year และรางวัลอื่นๆ จากคู่ค้าอีกมากมาย ดังภาพที่ 1.2



ภาพที่ 1.2 Metro Proud Awards

ปัจจุบันบริษัท เมโทรซิสเต็มส์คอร์ปอเรชั่น จำกัด (มหาชน) มีประเภทของผลิตภัณฑ์และบริการ ซึ่งสามารถแบ่งออกเป็น 3 กลุ่มใหญ่ๆ ได้แก่

- กลุ่มผลิตภัณฑ์ฮาร์ดแวร์ (Enterprise Systems Group – ESG) จำหน่ายผลิตภัณฑ์คอมพิวเตอร์ ฮาร์ดแวร์ ประกอบด้วย คอมพิวเตอร์ขนาดกลาง พีซี อุปกรณ์จัดเก็บข้อมูล ระบบค้ำปติก เครื่องพิมพ์สำหรับระบบคอมพิวเตอร์ขนาดใหญ่
- กลุ่มผลิตภัณฑ์ซอฟต์แวร์โซลูชั่น (Solutions Integration Group – SIG) จำหน่ายซอฟต์แวร์โซลูชั่นและบริการด้านไอทีเพื่อการบริหารงานที่มีประสิทธิภาพ
- กลุ่มผลิตภัณฑ์วัสดุสิ้นเปลือง (Office Supplies Group – OSG) จำหน่ายวัสดุสิ้นเปลืองที่เกี่ยวข้องกับคอมพิวเตอร์ เครื่องพิมพ์ขนาดเล็ก ก่อตั้งถ่ายรูป และเครื่องโปรเจ็คเตอร์ เป็นต้น

1.3 รูปแบบการจัดองค์กรและการบริหารองค์กร



ภาพที่ 1.3 คณะผู้บริหารบริษัท เมโทรซิสเต็มส์คอร์ปอเรชั่น จำกัด (มหาชน)

1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย

ตำแหน่ง	โปรแกรมเมอร์ (Programmer)
หน้าที่งานที่ได้รับมอบหมาย	เขียนโค้ด, แก้ไขบั๊กต่างๆ, จัดการตารางงาน

1.5 พนักงานที่ปรึกษา และ ตำแหน่งของพนักงานที่ปรึกษา

ชื่อ – นามสกุล

วิชาญ โสดาธาตุ

ตำแหน่ง

Program Developer

1.6 ระยะเวลาที่ปฏิบัติงาน

ปฏิบัติงานสหกิจศึกษาเป็นระยะเวลา 4 เดือน

นับตั้งแต่วันที่ 30 พฤษภาคม 2559 – 30 กันยายน 2559

1.7 ที่มาและความสำคัญของปัญหา

เนื่องจากในปัจจุบันเว็บแอปพลิเคชัน (Web Application) ส่วนมาก ไม่ได้มีการทดสอบประสิทธิภาพของเซิร์ฟเวอร์ในการรองรับการทำงานของผู้ใช้ที่มีการเข้ามาใช้แอปพลิเคชันและเมื่อทางผู้ผลิตได้ทำการปล่อยแอปพลิเคชันออกสู่สาธารณะ จึงเกิดเหตุเซิร์ฟเวอร์มีปัญหาเนื่องจากไม่สามารถรองรับผู้ใช้งานระบบ ที่มีจำนวนมากเกินกว่าที่คาดการณ์ไว้ได้ ทำให้เกิดการปรับปรุงอยู่บ่อยครั้ง

โปรแกรม Selenium WebDriver จึงได้ถูกพัฒนาขึ้นมาเพื่อทดสอบประสิทธิภาพของเว็บแอปพลิเคชันเพื่อที่จะสามารถให้แอปพลิเคชันทำงานได้อย่างมีประสิทธิภาพและรองรับจำนวนผู้ใช้งานระบบ ได้มากขึ้น โดยนำมาใช้ร่วมกับโปรแกรม Jmeter โดยมีการทดสอบ 2 แบบ คือการทำ Performance Test และการทำ Functional Test ซึ่งในโครงการนี้ทำในส่วนของ Performance Test เนื่องจากเป็นกรณีศึกษาจากทางบริษัท จึงได้ยกตัวอย่างการทำการทดสอบเพียงแค่การทำ Performance Test เท่านั้น

1.8 วัตถุประสงค์หรือจุดมุ่งหมายของโครงการ

1. เพื่อทดสอบประสิทธิภาพของเว็บแอปพลิเคชันได้
2. เพื่อทดสอบประสิทธิภาพของเว็บเซิร์ฟเวอร์
3. เพื่อป้องกันการเกิดเซิร์ฟเวอร์มีปัญหาเมื่อมีผู้ใช้งานจำนวนที่มากเกินไปกำหนด

1.9 ผลที่คาดว่าจะได้รับจากการปฏิบัติงานหรือโครงการที่ได้รับมอบหมาย

1. สามารถแก้ไขปัญหาเซิร์ฟเวอร์ในขณะที่มีผู้ใช้งานหลายคน
2. สามารถวางแผนการใช้งานของผู้ใช้งานที่จะเข้ามาใช้งานเว็บแอปพลิเคชันได้
3. สามารถรองรับจำนวนผู้ใช้งานเป็นจำนวนมากได้
4. สามารถตรวจสอบความถูกต้องในการทำงานของเว็บแอปพลิเคชันได้

1.10 นิยามศัพท์เฉพาะ

1. โปรแกรม Selenium WebDriver คือ โปรแกรมที่ใช้ในการเขียนคำสั่งสำหรับการทำการทดสอบ
2. โปรแกรม Apache Jmeter คือ โปรแกรมที่ใช้สำหรับการทำการทดสอบโดยเฉพาะ โดยโปรแกรม Apache Jmeter จะรองรับแค่ภาษาจาวาเท่านั้น
3. Load Test คือ การทำการทดสอบเพื่อวัดประสิทธิภาพการใช้งานเมื่อมีผู้เข้าใช้งานเป็นจำนวนมาก ส่วนใหญ่จะใช้งานกับการทดสอบสำหรับการ Login เข้าสู่ระบบ
4. Functional Test คือ การทำการทดสอบระบบฟังก์ชันการทำงานภายในเว็บแอปพลิเคชันว่าฟังก์ชันการทำงาน ยังสามารถใช้งานได้เป็นปกติอยู่หรือไม่
5. JUnit คือ ฟังก์ชันการทำงานในการทดสอบของโปรแกรม Apache Jmeter

บทที่ 2

ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน

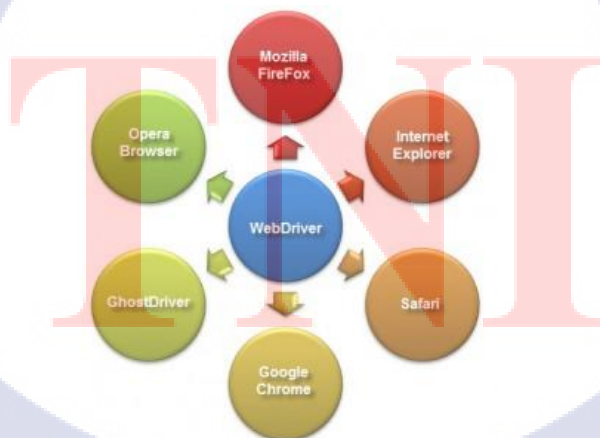
โครงการทดสอบประสิทธิภาพการใช้งานของเว็บแอปพลิเคชัน นี้จัดทำขึ้นมาโดยมีเครื่องมือเป็นโปรแกรมในการเขียนโค้ด และโปรแกรมที่ใช้ทำการทดสอบเว็บแอปพลิเคชัน โดยเฉพาะ ซึ่งโปรแกรมที่ใช้สำหรับทำเว็บแอปพลิเคชันมีดังนี้

2.1 เทคโนโลยีที่ใช้ในการปฏิบัติงาน

2.1.1 โปรแกรม Selenium WebDriver

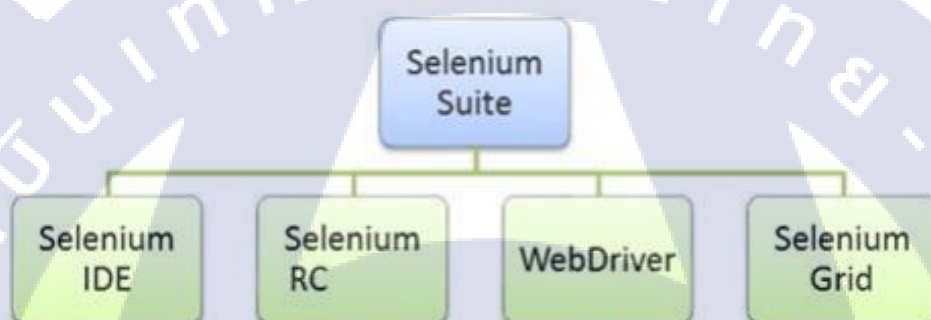
โปรแกรม Selenium WebDriver เป็นโปรแกรมเครื่องมือที่ช่วยให้เราสามารถสร้างโปรแกรมที่ใช้ในการทดสอบเว็บแอปพลิเคชันกับเว็บเบราว์เซอร์ได้หลายตัว ซึ่งถือเป็นคุณสมบัติเด่นที่ดีกว่า Selenium IDE (Selenium IDE จะใช้งานได้กับเฉพาะ Firefox เท่านั้น) โดยจะมี Web driver เป็นตัวกลางที่มี library ที่ช่วยให้เราสามารถติดต่อกับ เว็บเบราว์เซอร์ ได้

โดยใช้ภาษาโปรแกรมต่างๆ นำมาเขียน เช่น Java , .NET , C# และ Ruby มาติดต่อกับ Library ของ WebDriver เพื่อเข้าถึง Control ที่แสดงผ่าน เว็บเบราว์เซอร์ ได้ ทำให้เราสามารถสร้างโปรแกรมการทดสอบได้หลากหลายมากขึ้น เช่น ใช้ทดสอบในกรณีที่ต้องการทดสอบอย่างต่อเนื่อง และ ถึงข้อมูลที่ใช้สำหรับกรอกข้อมูลบนฟอร์มจากฐานข้อมูลได้ ดังภาพที่ 2.1



ภาพที่ 2.1 เว็บเบราว์เซอร์ ที่สามารถใช้ร่วมกับ WebDriver ได้

Selenium เป็นกลุ่มของเครื่องมือทดสอบต่างๆ ซึ่งแต่ละชนิดมีวิธีการทำงานร่วมกับการทดสอบอัตโนมัติที่แตกต่างกัน โดยนักทดสอบส่วนใหญ่จะเน้นไปที่การใช้เครื่องมือเพียงหนึ่งหรือสองเครื่องมือเท่านั้น แต่อย่างไรก็ตามการใช้เครื่องมือทั้งหมดร่วมกับการทดสอบอัตโนมัติจะช่วยให้การทดสอบสามารถครอบคลุมการทำงานของเว็บแอปพลิเคชันได้ในทุกๆ มิติ ข้อดีของการใช้ Selenium ในการทดสอบคือความสามารถในการรองรับการทำงานร่วมกับเว็บเบราว์เซอร์และระบบปฏิบัติการที่แตกต่างกันได้ ส่งผลให้ Tester สามารถทดสอบระบบที่มีความหลากหลายได้มากขึ้น Selenium ถูกแบ่งออกมาเพื่อวัตถุประสงค์ในการใช้งานที่แตกต่างกัน และสามารถแบ่งได้ออกเป็น 4 ชนิด ได้แก่ Selenium Integrated Development Environment (IDE) , Selenium Remote Control (RC) , WebDriver และ Selenium Grid ดังภาพที่ 2.2



ภาพที่ 2.2 กลุ่มของเครื่องมือทดสอบที่ใช้ในโปรแกรม Selenium

Selenium ถูกพัฒนาขึ้นเป็นครั้งแรกในปี 2004 และมีการพัฒนาการต่อเนื่องมาจนถึงปัจจุบันซึ่งในแต่ละช่วงเวลาของการพัฒนาได้มีการเพิ่มส่วนประกอบต่างๆ เข้าไปเพื่อให้มีความสามารถในการทำงานมากยิ่งขึ้น ขณะเดียวกันก็ได้มีการตัดบางส่วนหรือส่วนที่ล้าสมัยหรือไม่จำเป็นออกไปเพื่อให้เครื่องมือมีความทันสมัยมากขึ้น ซึ่งการพัฒนาดังกล่าวสามารถแบ่งออกได้ตามช่วงเวลาที่สำคัญๆ ดังนี้

2.1.1.1 Selenium Core

ในปี 2004 Jason Huggins วิศวกรของบริษัท Thought Works ซึ่งทำหน้าที่ในการพัฒนาเว็บแอปพลิเคชันที่ต้องการทดสอบบ่อยๆ ซึ่งจากการทำงานดังกล่าวทำให้เกิดแนวคิดที่ว่าการทำงานด้วยมือซ้ำๆ ส่งผลให้ประสิทธิภาพในการทดสอบลดลงๆ ดังนั้นจึงได้พัฒนาโปรแกรมด้วยภาษาสคริปต์โดยใช้ชื่อว่า

JavaScriptTestRunner เพื่อใช้สำหรับควบคุมการทำงานของเว็บเบราว์เซอร์โดยอัตโนมัติ ซึ่งในเวลาต่อมาได้ถูกเปลี่ยนชื่อเป็น Selenium Core และถือเป็นพื้นฐานการทำงานที่สำคัญสำหรับ Selenium Remote Control (RC) และ Selenium IDE ที่ถูกพัฒนาขึ้นในเวลาต่อมา

2.1.1.2 Selenium RC

ย้อนมาจากรากว่า Selenium Remote Control ซึ่งถูกพัฒนาขึ้นเพื่อแก้ไขข้อจำกัดจากแนวคิดแบบ Same Origin Policy โดย Paul Hammant วิศวกรอีกคนหนึ่งจากบริษัท Thought Works ได้พัฒนาโปรแกรมที่ใช้เป็นเซิร์ฟเวอร์เพื่อทำหน้าที่เป็น HTTP Proxy เพื่อใช้สำหรับหลอกเว็บเบราว์เซอร์ให้เชื่อว่า Selenium Core และเว็บแอปพลิเคชันได้ถูกทดสอบจากโดเมนเดียวกันซึ่งการทำงานในลักษณะดังกล่าวจึงเป็นที่มาของ Selenium RC หรือ Selenium 1 ในเวลาต่อมา

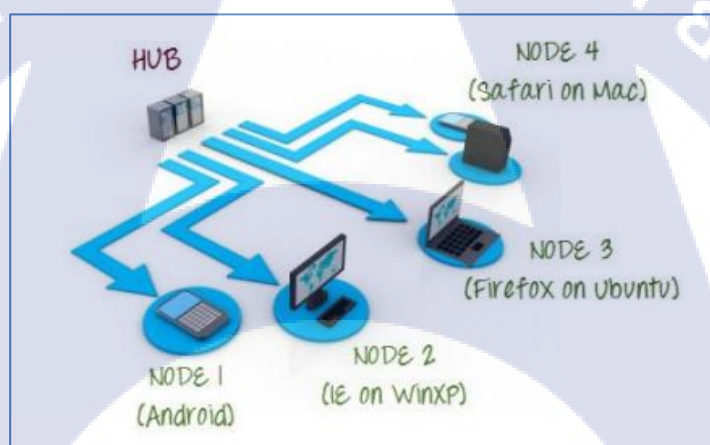
Selenium RC นอกจากจะใช้เป็นเครื่องมือทดสอบอัตโนมัติสำหรับเว็บแอปพลิเคชันในช่วงต้นๆ แล้ว ยังสนับสนุนการทำงานร่วมกับโปรแกรมภาษาต่างๆ ได้แก่ Java, C#, PHP, Python, Perl, Ruby, เป็นต้น นอกจากนี้ Selenium RC อาศัยการทำงานแบบ Client Server ที่ยอมให้ผู้ใช้ควบคุมการทำงานของเว็บเบราว์เซอร์จากภายในเครื่องเดียวหรือจากเครื่องคอมพิวเตอร์อื่นๆ ได้

ตารางที่ 2.1 ข้อดีข้อเสียของการใช้งานร่วมกับ Selenium RC

ข้อดี	ข้อเสีย
สามารถทำงานร่วมกับเว็บเบราว์เซอร์และระบบปฏิบัติการที่แตกต่างกันได้	การติดตั้งมีความยุ่งยากกว่า Selenium IDE
สนับสนุนการทำงานแบบวนซ้ำ (Loop) และเงื่อนไขต่างๆ (if/switch)	นักทดสอบต้องมีประสบการณ์ทางด้านโปรแกรมมาก่อน
สนับสนุนการทำงานแบบ Data Driven Testing	ต้องการ Selenium RC Server สำหรับ Run การทำงาน
มี API ที่สมบูรณ์และพร้อมใช้งาน	มี API มากเกินไปและบางคำสั่งซ้ำซ้อนกันก่อให้เกิดความสับสน
สนับสนุนการทำงานของเว็บเบราว์เซอร์รุ่นใหม่ๆ	ไม่สามารถลำดับการ Run การทดสอบในกรณีที่มีหลายๆ Test Case

2.1.1.3 Selenium Grid

ถูกพัฒนาขึ้นโดย Patrick Lightbody เพื่อจุดประสงค์ในการลดเวลาที่ใช้ในการประมวลผลการทดสอบ โดยใช้แนวคิดของการประมวลผลการทดสอบแบบขนาน นั่นคือการรันหลายๆการทดสอบที่แตกต่างกันในเวลาเดียวกันสามารถเกิดขึ้นได้ในเว็บเบราว์เซอร์และระบบปฏิบัติการที่แตกต่างกันโดยการผ่านคำสั่ง Selenium ไปยังคอมพิวเตอร์อื่นๆเพื่อการประมวลผลการทดสอบในเวลาเดียวกันได้ นั่นคือผู้ใช้สามารถใช้ Selenium Grid เป็นเครื่องมือที่ใช้ในการทดสอบ ร่วมกับ Selenium RC เพื่อรันการทำงานแบบขนานได้ ปัจจุบัน Selenium Grid สามารถแบ่งออกได้เป็น 2 เวอร์ชัน ได้แก่ Grid 1 ซึ่งเป็นเวอร์ชันเก่า และ Grid 2 เป็นเวอร์ชันที่ใช้งานอยู่ในปัจจุบัน Selenium Grid ใช้แนวคิดการทำงานแบบ Hub-Node โดยผู้ใช้สามารถรันการทดสอบเฉพาะบนเครื่องคอมพิวเตอร์ซึ่งทำหน้าที่เป็น Hub และผ่านคำสั่งไปยังคอมพิวเตอร์ซึ่งทำหน้าที่เป็น Nodes เพื่อประมวลผลการทดสอบในเวลาเดียวกัน ดังภาพที่ 2.3



ภาพที่ 2.3 แสดงการทำงานของ Selenium Grid

2.1.1.4 Selenium IDE

ถูกพัฒนาโดยชาวญี่ปุ่นที่ชื่อ Shinya Kasatani ในรูปส่วนต่อขยายของ Firefox ที่สามารถใช้ในการทำงานแบบอัตโนมัติร่วมกับเว็บเบราว์เซอร์ผ่านเทคนิคการทดสอบที่เรียกว่า Record & Playback โดยมีจุดประสงค์เพื่อให้เป็นเครื่องมือที่ใช้งานได้ง่าย โดยผู้ใช้ไม่จำเป็นต้องมีพื้นความรู้ทางด้านโปรแกรมมาก่อน นอกจากนั้นยังเป็นเครื่องมือที่ช่วยเพิ่มความเร็วในการสร้าง Test Case ซึ่งในเวลาต่อมาจึงได้มอบ Selenium IDE ให้กับโครงการพัฒนา Selenium ในปี 2006

Selenium IDE จัดเก็บ Script สำหรับการทดสอบในรูปแบบตารางที่สามารถนำกลับไปใช้ใหม่เพื่อการประมวลผลซ้ำได้ในภายหลัง แต่อย่างไรก็ตามตัวโปรแกรมเองยังขาดคุณสมบัติที่สำคัญคือ คุณสมบัติในการ

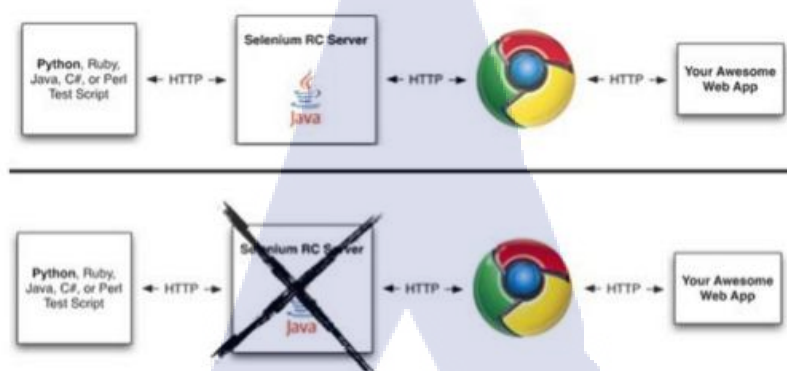
การ Run ผลลัพธ์การทดสอบหรือความสามารถในการสร้างการทดสอบอัตโนมัติที่ผู้ใช้งานต้องการ นอกจากนั้น Selenium IDE ยังไม่สนับสนุนการทำงานประเภททำซ้ำหรือเงื่อนไขต่างๆสำหรับการใช้งานร่วมกับ Script สำหรับการทดสอบ ทั้งนี้เนื่องจากเหตุผลบางส่วนที่เป็นปัญหาทางเทคนิคและบางส่วนจากตัวนักพัฒนาเอง อย่างไรก็ตามเนื่องจากความง่ายต่อการใช้ Selenium IDE จึงถูกจำกัดให้เป็นได้เฉพาะเครื่องมือต้นแบบที่ใช้สำหรับการทดสอบแบบพื้นฐานเท่านั้น ส่วนในกรณีที่ต้องสร้าง Test Case ที่มีการทำงานแบบซับซ้อนจำเป็นต้องเลือกใช้ Selenium RC หรือ WebDriver แทน

ตาราง 2.2 ข้อดีข้อเสียของการใช้งานร่วมกับ Selenium IDE

ข้อดี	ข้อเสีย
สะดวกต่อการติดตั้งและใช้งาน	ใช้ได้เฉพาะกับ เว็บเบราว์เซอร์ Firefox
ไม่จำเป็นต้องมีประสบการณ์ทางด้านโปรแกรมมาก่อน ใช้ความรู้เฉพาะ HTML และ DOM	ออกแบบมาเพื่อใช้เป็นเครื่องมือต้นแบบเท่านั้น
สามารถส่งออกการทดสอบไปยังรูปแบบที่ใช้ได้กับ Selenium RC และ WebDriver	ไม่สนับสนุนการทำงานแบบทำซ้ำหรือแบบมีเงื่อนไข
มี Help และ Module ที่ถูกสร้างไว้แล้วสำหรับรายงานผลการทดสอบ	การประมวลผลการทดสอบช้าเมื่อเปรียบเทียบกับ Selenium RC และ WebDriver

2.1.1.5 WebDriver

ในปี 2006 Simon Stewart ได้พัฒนา WebDriver ขึ้นโดยเพิ่ม API ที่เป็นทางเลือกสำหรับฟังก์ชันการทำงานที่ไม่ได้รับการสนับสนุนจาก Selenium RC โดยการทำงานของ WebDriver จะไม่ขึ้นอยู่กับ JavaScript ที่ฝังตัวอยู่ในเว็บเบราว์เซอร์ ดังนั้นจึงสามารถลดข้อจำกัดที่เกิดขึ้นจากการทำงานของ Selenium ได้ นอกจากนั้น WebDriver ยังมีเป้าหมายหลักในการพัฒนาเครื่องมือทดสอบอัตโนมัติเพื่อให้สามารถทำงานข้าม Platform ได้ รวมถึงความสามารถในการควบคุมการทำงานร่วมกับเว็บเบราว์เซอร์ได้ในระดับระบบปฏิบัติการ ดังภาพที่ 2.4



ภาพที่ 2.4 เปรียบเทียบการทำงานระหว่าง Selenium RC และ WebDriver

WebDriver ได้ถูกออกแบบมาเพื่อแก้ไขข้อจำกัดต่างๆที่เพิ่มมากขึ้นจากการใช้ JavaScript รวมไปถึงข้อบกพร่องต่างๆที่เกิดขึ้นทั้งจาก Selenium RC และ Selenium IDE โดยมีชุดคำสั่งที่ถูกออกแบบมาเป็นแบบเชิงวัตถุส่งผลให้การนำไปใช้สามารถทำได้ง่ายขึ้น รวมทั้งยังสนับสนุนการทำงานกับโปรแกรมภาษาต่างๆ เช่นเดียวกับ Selenium RC นอกจากนี้ยังมีการทำงานที่เสถียรมากกว่าในการทดสอบอัตโนมัติร่วมกับเว็บเบราว์เซอร์ชนิดต่างๆ

ตาราง 2.3 ข้อดีและข้อเสียของการใช้งานร่วมกับ WebDriver

ข้อดี	ข้อเสีย
ติดตั้งง่ายกว่า Selenium RC	การติดตั้งยุ่งยากกว่า Selenium IDE
ติดต่อโดยตรงกับเว็บเบราว์เซอร์	ต้องการความรู้ด้านโปรแกรมมาก่อน
การติดต่อกับเว็บเบราว์เซอร์มีความเสมือนจริง	ไม่สนับสนุนเว็บเบราว์เซอร์รุ่นใหม่ๆ
ไม่ต้องใช้ Component อื่น เช่นเดียวกับ Selenium RC Server	ไม่มีกลไกสำหรับแสดงข้อความเตือนในช่วงเวลารันไทม์และสร้างผลลัพธ์การทดสอบ
ประมวลผลได้เร็วกว่า Selenium IDE และ Selenium RC	

2.1.1.6 Selenium 2

ในปี 2008 ทีมพัฒนา Selenium ทั้งหมดได้ตัดสินใจรวม WebDriver และ Selenium RC เข้าด้วยกัน เพื่อสร้างเครื่องมือชนิดใหม่ที่มีประสิทธิภาพมากขึ้น เรียกว่า Selenium 2 โดยใช้ WebDriver เป็นแกนหลักในการทำงาน ปัจจุบัน Selenium RC ไม่มีการพัฒนาเวอร์ชันใหม่อีกต่อไปแล้ว มีแต่เพียงส่วนของการบำรุงรักษาเท่านั้น ดังนั้นแนวโน้มของการพัฒนาเครื่องมือทดสอบปัจจุบันจะเน้นไปที่ Selenium 2 เสมอ

2.1.2 ภาษา Java

Java เป็นภาษาโปรแกรมเชิงวัตถุ โดยมีจุดเด่นคือสามารถใช้หลักของ OOP (Object-Oriented Programming) มาพัฒนาโปรแกรมได้ ภาษาจาวา เดิมชื่อ ภาษาโอ๊ค (Oak) โดยกำเนิดขึ้นมาจากบริษัท ซัน ไมโครซิสเต็มส์ ซึ่งในปี ค.ศ. 1990 แพททริก นอตัน โปรแกรมเมอร์คนหนึ่งในบริษัท ซันฯ รายงานกับประธานบริษัทว่าจะลาออกและจะไปทำงานอีกบริษัทหนึ่งแทน ประธานบริษัทจึงขอให้เขาช่วยระบุข้อผิดพลาดของบริษัทซันฯ และให้ช่วยเขียนวิธีแก้ข้อผิดพลาดที่ว่านั้นด้วย

นอตันจึงได้เขียนรายงานถึงปัญหาและวิธีการแก้ปัญหานั้น และได้ส่งอีเมลล์ให้กับประธานบริษัท และคณะผู้บริหารอื่นๆทั้งหมด วันถัดมา พนักงานบริษัททั้งหมดที่ได้อ่านสำเนาจดหมายก็ได้เขียนอีเมลล์กลับไปหานอตันด้วยความเห็นตรงกันทั้งหมด และหนึ่งในนั้นก็มี “เจมส์ กอสลิง” ผู้ที่ในอนาคตจะให้กำเนิดภาษาจาวาขึ้นมา นอตันเปลี่ยนใจที่จะลาออก โดยเขาเข้าร่วมกับกลุ่มผู้สร้างโครงการใหม่ในชื่อ “กรีนทิม” เป็นทีมที่ไม่ได้มีจุดมุ่งหมายเพื่อที่จะพัฒนาภาษา แต่มีวัตถุประสงค์คือการสร้างคอมพิวเตอร์รุ่นใหม่ ซึ่งเป็นคอมพิวเตอร์สำหรับคนทั่วไป นอตันคาดการณ์ว่า ยุคถัดจากคอมพิวเตอร์ส่วนบุคคล ก็คือยุคคอมพิวเตอร์ในอุปกรณ์อิเล็กทรอนิกส์

อุปกรณ์อิเล็กทรอนิกส์รุ่นใหม่ๆ ทักจะมีชิปคอมพิวเตอร์ติดตั้งอยู่ ตั้งแต่อุปกรณ์ที่เห็นได้ชัดเจนเช่น สมาร์ทการ์ด โทรศัพท์มือถือ ไปจนถึงอุปกรณ์ที่คนทั่วไปคาดไม่ถึงหรือไม่ได้มองเป็นคอมพิวเตอร์ เช่น เครื่องปิ้งขนมปัง เครื่องเล่นวิดีโอ และรถยนต์ เนื่องจากการใช้งานอุปกรณ์อิเล็กทรอนิกส์เหล่านี้ไม่ได้มีมาตรฐานเดียวกัน และอุปกรณ์บางอย่างนั้นใช้งานยาก กรีนทิมจึงคิดที่จะสร้างอุปกรณ์ที่ใช้ควบคุมอุปกรณ์เหล่านั้น โดยอุปกรณ์ตัวนี้มีชื่อว่า “สตาร์เซเวน (*7)”

สตาร์เซเวน เป็นรีโมตคอนโทรลที่มีลักษณะคล้าย “พีดีเอ (Personal Digital Assistant)” อุปกรณ์ตัวนี้จะมีแต่จอภาพไม่มีปุ่มกดใดๆ เมื่อผู้นำปากกาไปสัมผัสกับหน้าจอ ตัวตุ๊ก (Duke) จะปรากฏขึ้นและเป็นไกด์ในโลกเสมือนจริง สตาร์เซเวนสามารถควบคุมอุปกรณ์ต่างชนิดกันได้เนื่องจากอุปกรณ์เหล่านั้นรู้จักภาษาเดียวกัน นั่นคือ “ภาษาโอ๊ค”

กอสลิง ได้ประดิษฐ์ภาษาไค้ขึ้นมา เพราะเห็นว่า ภาษาC++ ซึ่งเป็นภาษาที่ใช้เป็นมาตรฐานในอุตสาหกรรมซอฟต์แวร์นั้น ถึงแม้จะทำงานได้อย่างรวดเร็ว แต่ก็ไม่เหมาะที่จะใช้กับอุปกรณ์อิเล็กทรอนิกส์ ภาษาที่เหมาะสมกับอุปกรณ์อิเล็กทรอนิกส์ ต้องมีคุณสมบัติสำคัญคือความน่าเชื่อถือ ผู้ใช้คงไม่อยากจะใช้โปรแกรมที่หยุดทำงานบ่อยๆ โปรแกรมที่เขียนด้วยภาษาC++มีรูปแบบการอ้างอิงถึงข้อมูลที่เสี่ยงต่อการทำให้โปรแกรมหยุดการทำงาน ทำให้ภาษาC++ไม่เหมาะสมที่จะใช้โปรแกรมอุปกรณ์อิเล็กทรอนิกส์

อุปกรณ์อิเล็กทรอนิกส์ที่เป็นตลาดของภาษาไค้และสตาร์เซเวน คือ โฮมคอมพิวเตอร์ รถยนต์ โทรศัพท์ โทรศัพท์ หรือทุกสิ่งทำงานด้วยระบบดิจิทัลนั่นเอง แต่เนื่องจากไม่สามารถตกลงเงื่อนไขกับลูกค้าได้ ภาษาไค้และเครื่องสตาร์เซเวนก็เลยไม่ได้ออกสู่ท้องตลาด

ในเดือนมิถุนายน ปี ค.ศ.1993 มาร์ค แอนดริสเซน ซึ่งในขณะนั้นยังเป็นนักศึกษาที่มหาวิทยาลัยแห่งอิลลินอยส์ได้ออกแบบโปรแกรม Mosaic ซึ่งเป็นโปรแกรมเว็บเบราว์เซอร์ ตั้งแต่นั้นเป็นต้นมาความนิยมในการใช้เว็บ ก็ได้เริ่มต้นขึ้น

บิล จอย หนึ่งในผู้ร่วมก่อตั้งบริษัทซันฯ มองเห็นว่าเว็บเบราว์เซอร์เป็นหนทางที่จะกู้สถานการณ์ของภาษาไค้ เขาเสนอแผนการธุรกิจแนวแฟรนไชส์ในอินเทอร์เน็ต นั่นคือแจกจ่ายตัวพัฒนาภาษาออกไปฟรีๆ เพื่อสร้างส่วนแบ่งตลาด

ในยุคก่อนหน้าที่เว็บจะเป็นที่นิยม การที่บริษัทจะแจกซอฟต์แวร์ของตนฟรีๆออกไปนั้นเป็นเรื่องที่แหวกแนวมาก แต่ในยุคอินเทอร์เน็ตนั้น มีผู้ประสบความสำเร็จด้วยกลยุทธ์นี้มาแล้ว นั่นคือบริษัท Netscape ซึ่งแจกเว็บเบราว์เซอร์ของตนออกไปฟรีๆ เพื่อสร้างฐานลูกค้า รายได้ของบริษัทมาจากการขายโปรแกรมเว็บเซิร์ฟเวอร์

ต่อมาภาษาไค้ของกอสลิงได้เปลี่ยนชื่อไปเป็น ภาษาจาวา เพราะชื่อดั้งกล่าวนั้นไปซ้ำกับภาษาโปรแกรมที่มีอยู่แล้ว และนอตันได้พัฒนาเว็บเบราว์เซอร์ที่สนับสนุนภาษาจาวา โปรแกรมดังกล่าวมีชื่อว่า HotJava ซึ่งทางบริษัทซันฯได้แจกให้กับกลุ่มคนไม่กี่กลุ่ม

ซึ่งแอนดริสเซน เป็นหนึ่งในคนกลุ่มนั้น เขาได้ลองใช้โปรแกรม HotJava และชอบมาก ในขณะนั้นแอนดริสเซนไม่ได้เป็นนักศึกษาธรรมดาแล้ว เขาได้เป็นผู้ร่วมก่อตั้งบริษัท Netscape อันโด่งดัง เขาตัดสินใจซื้อสิทธิ์ในการใช้ภาษาจาวาในโปรแกรม Netscape Navigator

โปรแกรม Navigator นั้นเป็นเว็บเบราว์เซอร์ยอดนิยมในยุคนั้น ถ้า Navigator สามารถรันโปรแกรมภาษาจาวาที่เรียกว่า จาวาแอปเพล็ต ก็เป็นที่แน่ใจได้ว่าผู้ใช้อินเทอร์เน็ตส่วนใหญ่สามารถรันแอปเพล็ตได้ แต่เพื่อให้อินเทอร์เน็ตเต็มไปด้วยแอปเพล็ต บริษัทซันฯต้องรีบออกชุดพัฒนาภาษาจาวาและแจกจ่ายออกไป โดยชุดพัฒนาภาษาจาวาเวอร์ชันแรก (JDK1.0) ได้ถูกเผยแพร่ในวันที่ 23 มกราคม ค.ศ.1996



ภาพที่ 2.5 Java Programming

2.1.3 โปรแกรม Apache Jmeter

โปรแกรม Apache Jmeter เป็น Open Source Software ที่ใช้กับแอปพลิเคชันสำหรับภาษาจาวา โดยเฉพาะ สำหรับการทำ load test เพื่อวัดประสิทธิภาพการทำงาน ถูกออกแบบมาเพื่อทำการทดสอบเว็บแอปพลิเคชัน

โปรแกรม Apache Jmeter อาจจะใช้ในการทดสอบ ประสิทธิภาพทั้ง static และ dynamic resources สามารถนำมาใช้จำลอง heavy load บนเซิร์ฟเวอร์หรือเน็ตเวิร์กเพื่อทดสอบประสิทธิภาพ หรือนำมาวิเคราะห์ประสิทธิภาพโดยรวมภายใต้ประเภทการ Load ที่แตกต่างกัน สามารถใช้เพื่อทำเป็น graphical analysis ของประสิทธิภาพการทำงานได้

Stefano Mazzocchi แห่ง Apache Software Foundation ผู้พัฒนาต้นฉบับของโปรแกรม Jmeter เขาเขียนโปรแกรม Jmeter ขึ้นมาเพื่อใช้ในการทดสอบประสิทธิภาพของ Jserver และพวกเขาได้เพิ่ม GUI และเพิ่มความสามารถในการทำการทดสอบ Functional Test

โปรแกรม Jmeter กลายเป็นโครงการที่อยู่ระดับบนสุดของ Apache Project ในเดือนพฤศจิกายน 2011 ซึ่งทำให้โปรแกรม Jmeter ได้มีคณะกรรมการบริหารโครงการและเว็บไซต์โดยเฉพาะของตัวเองตั้งแต่นั้นเป็นต้นมา

2.1.4 โปรแกรม Spring Tool Suite

Spring Tool Suite คือ Eclipse-based development environment ที่กำหนดขึ้นมาสำหรับพัฒนา Spring Application ที่ให้ความพร้อมต่อการใช้งานในการ implement , debug , run และ deploy รวมไปถึงการผสานเข้ากับ Pivotal tc Server , Pivotal Cloud Foundry , Git , Maven , AspectJ และ อื่นๆที่จะตามมาในรุ่นล่าสุดของ Eclipse



ภาพที่ 2.6 Spring Tool Suite Logo

TNI

บทที่ 3

แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน

3.1 แผนงานการปฏิบัติงาน

ตารางที่ 3.1 ตารางการปฏิบัติงาน

หัวข้องาน	พ.ค. 59		มิ.ย. 59		ก.ค. 59		ส.ค. 59		ก.ย. 59	
ปฏิบัติงานที่ได้รับมอบหมาย										
ศึกษา Selenium WebDriver										
ศึกษา Apache Jmeter										
วางแผนการทำการทดสอบ										
ศึกษาระบบของเว็บแอปพลิเคชันที่จะทำการทดสอบ										
เขียนโค้ด Program Selenium WebDriver										
Load Test ด้วย Jmeter										
ทำผลวิเคราะห์ผลการทดสอบ										

3.2 รายละเอียดที่นักศึกษาปฏิบัติในการฝึกงาน

จากการที่ได้มาสหกิจศึกษาที่ Metro System Corporation จำกัด (มหาชน) ซึ่งในขณะที่ผมได้เข้ามาทำสหกิจศึกษาอยู่นั้น ทางบริษัทได้มีการทำโครงการใหญ่ที่ต้องจัดส่งมอบให้ลูกค้าอยู่ ผมจึงได้มีส่วนร่วมในการเข้าทำโครงการนี้ด้วย ซึ่งงานที่ผมได้รับมอบหมายนั้นมีตั้งแต่ จัดทำเอกสารในส่วนของ Unit Test, Screen Report, Program Specification ในรูปแบบของเอกสาร Excel จากนั้นก็ได้มีการนำจัดเข้าเพิ่มส่งมอบลูกค้า ได้ทำการขนย้ายข้อมูลจำนวนหนึ่งผ่านโปรแกรมที่ทางบริษัทจัดเตรียมไว้ให้ และได้ทำการทดสอบประสิทธิภาพของเว็บแอปพลิเคชันของทางบริษัท ซึ่งผมได้ขอกับทางหัวหน้างานและนำการทดสอบที่ว่านั้นมาจัดทำรูปเล่มโครงการนี้ การทดสอบที่ผมได้ทำที่บริษัทนั้น เป็นการทดสอบประสิทธิภาพของเว็บแอป

พลิเคชันของทางบริษัท โดยทดสอบว่าหากมีผู้ใช้งานระบบจำนวนมากเข้ามาใช้งานเว็บแอปพลิเคชัน นั้นเครื่องเซิร์ฟเวอร์จะ Over load หรือไม่ เป็นการทำให้ Load Test อย่างหนึ่งเพื่อทดสอบประสิทธิภาพของเซิร์ฟเวอร์ เพื่อป้องกันความล่าช้าในการใช้งานและปัญหาการรองรับจำนวนผู้ใช้งานระบบไม่พอใช้งาน

3.3 ขั้นตอนการดำเนินงานที่นักศึกษาปฏิบัติงาน

3.3.1 ศึกษาการทำ Load Test ด้วยโปรแกรม Selenium

Selenium เป็นโปรแกรมสำหรับการทำ Load Test โดยภาษาที่ใช้เขียนการทดสอบคือภาษา Java ซึ่งเป็นรูปแบบการเขียนที่ไม่เคยศึกษามาก่อน แต่รูปแบบการเขียนการทำ Load Test นั้นไม่ได้ต่างกันมากนัก ทำให้เข้าใจได้ง่ายและไม่ซับซ้อน

3.3.2 ศึกษาโปรแกรม Apache Jmeter

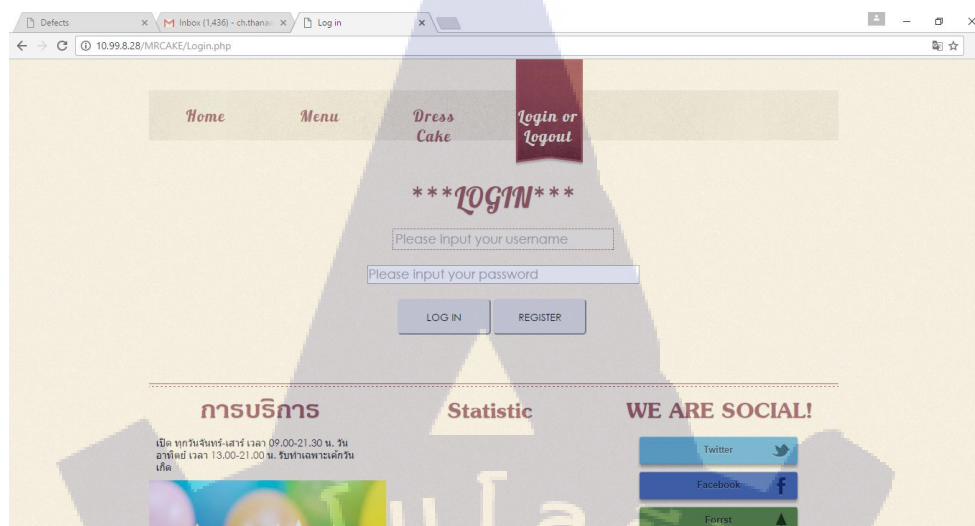
Jmeter แต่เดิมเป็นโปรแกรมที่มีไว้สำหรับการทดสอบอยู่แล้ว แต่สำหรับการทดสอบที่ผมทำนั้นผมได้ใช้งานร่วมกับ Selenium โดยใช้ Selenium เป็นเหมือนเครื่องนำทางเข้าสู่เว็บแอปพลิเคชันและใช้ Jmeter ในการทำวนซ้ำการทำงาน เพื่อทดสอบประสิทธิภาพของเซิร์ฟเวอร์ ว่า หากมีการวนซ้ำการทำงานแบบนี้เป็นเวลา 3 ชั่วโมงขึ้นไปเซิร์ฟเวอร์จะมีปัญหาการทำงานหรือไม่

3.3.3 วางแผนการทดสอบ

วางแผนการทดสอบว่าเราจะทำการทดสอบในส่วนไหนของเว็บแอปพลิเคชัน และเลือกที่จะทำการทดสอบ แบบ Performance Test หรือ Functional Test

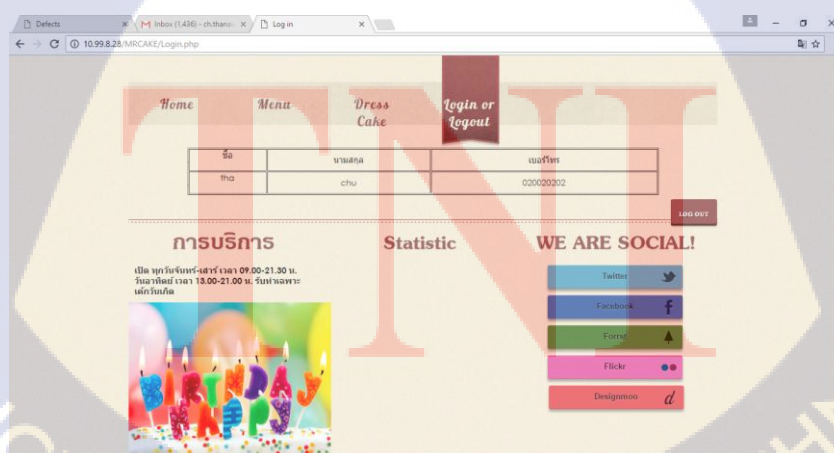
3.3.4 ศึกษาระบบของแอปพลิเคชันที่จะทำการทดสอบ

ศึกษาระบบของเว็บแอปพลิเคชันที่จะทำการทดสอบว่าหากกดลิงค์ตรงนี้แล้วจะไปไฟล์ไหน หน้าไหนแล้วต้องกดตรงไหนต่อ จึงจะไปถึงหน้าที่เป็นจุดสิ้นสุดของการทดสอบ ดังภาพที่ 3.1



ภาพที่ 3.1 หน้าเว็บไซต์ที่ทำการทดสอบ Login

จากภาพจะเป็นการทดสอบที่เราได้ทำนั้น เป็นการทดสอบโดยการ ถ้าหากมีการเข้าใช้ Login และ Logout อย่างต่อเนื่อง จะเกิดการล่าช้าหรือปัญหาของเว็บแอปพลิเคชันหรือไม่ โดยแบ่ง Scenario การทดสอบเป็น 2 Scenario คือ 1. 50 User และ 2. 100 User ซึ่ง Website ที่นำมา Test เป็น local ของ PC อีกเครื่องหนึ่ง ทั้งนี้เพราะไม่อาจทำการทดสอบบนเว็บแอปพลิเคชัน ของทางบริษัทได้เนื่องจากติดลิขสิทธิ์ของทางบริษัท โดยเราจะเริ่มจากการทำการ Login เข้าสู่ระบบของเว็บไซต์และเมื่อ Login สำเร็จแล้ว เราก็จะทำการ Logout และทำการทดสอบเช่นนี้จนกระทั่งครบ 2 Scenario



ภาพที่ 3.2 หน้าเว็บไซต์ที่ทำการทดสอบ Logout

3.3.5 Coding Selenium WebDriver

การเขียน Selenium นั้น ภาษาที่เขียนจะใช้ภาษา Java เนื่องจากภาษา Java ใช้งานง่าย รองรับได้หลายโปรแกรม และ Jmeter รองรับแค่ภาษา Java ภาษาเดียวเท่านั้น

การจะ Coding Selenium จะต้องมีการเตรียมไอดีของผู้ใช้งาน จำนวนหนึ่งไว้ทำการทดสอบ ซึ่งจำนวนไอดีผู้ใช้งานที่ต้องเตรียม หาได้จากการสมัครไอดีผู้ใช้งานของเว็บแอปพลิเคชันนั้นไว้ เมื่อเตรียมไอดีผู้ใช้งานเสร็จเรียบร้อยแล้ว ก็จะเริ่มการเขียนโค้ด โดยการการเขียนโค้ดจะเรียงลำดับจากขั้นตอนแรกของการท่องเว็บแอปพลิเคชัน ซึ่งจะเริ่มจากหน้า Login ะส่วนใหญ่ ไปจนถึงขั้นตอนการทำงานสุดท้ายของเว็บแอปพลิเคชัน เช่น การกดยืนยันการทำรายการใดรายการหนึ่ง หลังจากเสร็จสิ้นขั้นตอนสุดท้ายแล้ว โปรแกรม Selenium จะทำการวนไอดีผู้ใช้งานคนที่สองเข้ามาทำการทดสอบต่อ และจะวนเรื่อยจนครบจำนวนไอดีผู้ใช้งาน ที่กำหนดไว้

การทดสอบสามารถทำได้หลายๆเครื่องพร้อมๆกัน แต่จะต้องใช้ไอดีผู้ใช้งานระบบที่ต่างกัน หากใช้ไอดีผู้ใช้งานระบบ ที่เหมือนกัน อาจจะทำให้ไอดี ซกกันขณะทำการทดสอบและทำให้เกิด Error ได้ ดังภาพที่ 3.3 โดยจะเริ่มจากการสร้าง Class และ กำหนดค่าในการรับ ไฟล์ที่มีข้อมูล ID Password ที่ใช้สำหรับการ Login, เว็บเบราว์เซอร์ URL, Program browser ที่จะใช้สำหรับการ Test, Username และ Password

```
public class TestProject {
    String csvPathFile = "C:\\Works\\Users_test.txt";

    // public HtmlUnitDriver driver;
    public WebDriver driver;
    // public String baseUrl = "http://alpha.fsct.local";
    public String baseUrl = "http://10.99.8.28/MRCAKE/index.php";
    public String webDriver = "C:\\dev\\workspace\\libs\\chromedriver.exe";
    public String userName = ""; //utest5043-05";
    public String passWord = ""; //pwdtest5043";
    Wait<WebDriver> wait;
```

ภาพที่ 3.3 Selenium Coding

จากนั้นให้เขียนโค้ดคำสั่งเพื่อให้ Selenium ทำงานต่อหลังจากที่โหลดหน้า เว็บเบราว์เซอร์ เสร็จ โดยจะมีการเช็ค ว่า หากคำสั่งนี้ยังไม่สำเร็จ จะให้มีการวนลูปเช็คคำสั่งนี้ใหม่ทุกๆวินาทีภายในเวลาที่วินาที เพื่อป้องกันไม่ให้คำสั่งทำงานก่อนที่หน้า webpage จะโหลดเสร็จตามภาพที่ 3.4

```

public void waitForPageLoaded() {
    ExpectedCondition<Boolean> expectation = new ExpectedCondition<Boolean>() {
        public Boolean apply(WebDriver driver) {
            return (((org.openqa.selenium.JavascriptExecutor) driver).executeScript("return document.readyState"))
                .equals("complete");
        }
    };
    try {
        WebDriverWait wait = new WebDriverWait(driver, 60);
        wait.until(expectation);
    } catch (Throwable error) {
        Assert.fail("Timeout waiting for Page Load Request to complete.");
    }
}

```

ภาพที่ 3.4 Selenium Coding

ลำดับต่อไปคือการเขียนโค้ดคำสั่งสำหรับการทำงานเพื่อเปิดเว็บเบราว์เซอร์ ดังภาพที่ 3.5

```

try {
    System.setProperty("webdriver.chrome.driver", webdriver);
    driver = new HtmlUnitDriver(BrowserVersion.CHROME);true
    driver = new HtmlUnitDriver();
    driver.setJavaScriptEnabled(true);
    driver = new ChromeDriver();
    driver.manage().window().maximize();
    driver.manage().timeouts().implicitlyWait(180, TimeUnit.SECONDS);
    // Waiting 30 seconds for an element to be present on the page, checking for its presence once every 5 seconds.
    wait = new FluentWait<WebDriver>(driver)
        .withTimeout(180, TimeUnit.SECONDS)
        .pollingEvery(15, TimeUnit.SECONDS)
        .ignoring(NoSuchElementException.class);
}

```

ภาพที่ 3.5 Selenium Coding

จากภาพที่ 3.5 สามารถอธิบายการทำงานของโค้ดที่เขียนได้ดังนี้

- System.setProperty เป็น method ที่จะตั้งคุณสมบัติของระบบที่ระบุไว้โดยเรียกใช้จาก Key ที่มีการระบุไว้แล้ว
- driver = new ChromeDriver() เป็นคำสั่งที่ใช้รับค่าของ ChromeDriver() โดยจะใช้ variable ที่ชื่อ driver เป็นตัวเก็บค่า และเมื่อมีการเขียนคำสั่ง ก็จะใช้ variable ตัวนี้เป็นตัวในการรับคำสั่งแทน ดังคำสั่งในบรรทัดถัดไป driver.manage().window().maximize(); คือคำสั่งขยายขนาดหน้าจอของ Browser เป็นต้น

- wait เป็นคำสั่งให้รอหากคำสั่งที่เขียนไม่มีการทำงาน เช่น สั่งให้มีการกดกลับไปยังหน้า Home แต่หน้า Browser ยังโหลดไม่เสร็จ ทำให้ปุ่ม Home ยังไม่ปรากฏ คำสั่ง wait ก็จะทำการเช็คค่าถ้ายังกดปุ่ม Home ไม่ได้ ก็จะหยุดคำสั่งตัวนั้นให้รอ โดยเราสามารถตั้งเวลาในการรอได้ด้วยคำสั่ง withTimeout ซึ่งจะเป็นคำสั่งให้ทำการรอเป็นเวลาเท่าไร (หน่วยเป็นวินาที) และจะมีการเช็คคำสั่งใหม่ในทุกๆที่วินาที (สามารถเลือกหน่วยเวลาได้ จากภาพที่ 3.5 เลือกหน่วยเวลาเป็น วินาที โดยดูจากคำสั่ง TimeUnit.SECONDS)

คำสั่ง CSVReader คือคำสั่งที่ใช้อ่านไฟล์ txt ที่เราเก็บไว้ ซึ่งทางเข้าไฟล์ txt จะอยู่ที่ภาพที่ 3.3 ที่คำสั่ง String csvPathFile ซึ่งคำสั่งนี้จะเข้าไปนำข้อมูลจากไฟล์ txt ในโฟลเดอร์ปลายทางและนำมาวนลูปเพื่อเก็บค่า Username และ Password จากไฟล์ txt ดังภาพที่ 3.6

```
CSVReader readerCsv = new CSVReader(new FileReader(csvPathFile));
int len = readerCsv.toString().length();

List<String[]> lis = readerCsv.readAll();
System.out.println(lis.size()+" USERS !!");
for (int i = 0; i < lis.size(); i++) {
    System.out.println(i);
    userName = lis.get(i)[2];
    passWord = lis.get(i)[3];
}
```

ภาพที่ 3.6 Selenium Coding

จากนั้นจะเริ่มเขียนคำสั่งเพื่อให้โปรแกรมทำงาน โดยลักษณะการเขียนออกคำสั่งจะคล้ายๆกันทั้งหมด คือ กำหนด variable ขึ้นมาตัวหนึ่งเพื่อรับค่า Attribute ในแต่ละคำสั่ง ซึ่ง Attribute ที่นำมาจากเว็บไซต์ สามารถหาได้จาก ให้เมาส์ ไปชี้ที่ปุ่มที่ต้องการจะให้กด คลิกขวา เลือก Inspect จากนั้นแถบ Console จะแสดงขึ้นมาให้เห็น เราสามารถเลือก Attribute ได้ที่แถบ Console และนำมาใส่ไว้ในคำสั่ง findElement(By.name("xxxx")); ในส่วนของ "xxxx" นั้น ให้เรานำ Attribute ไปใส่แทน เสร็จแล้ว variable จะรับค่า Attribute ตัวนี้เอาไว้ให้นำ variable ที่กำหนดไว้ในแต่ละคำสั่ง ไปเขียนออกคำสั่งเพื่อให้มันทำงาน เช่น elementLogin.click(); คือสั่งเพื่อบอกให้ variable ตัวนั้น คลิกปุ่ม ที่เราได้เลือกจาก Attribute ไว้ เป็นต้น ทำเช่นนี้ไปจนครบกระบวนการทดสอบที่เราต้องการ ดังภาพที่ 3.7

```

Thread.sleep(1000);
WebElement elementClickLogin = wait.until(new ExpectedCondition<WebElement>() {
    public WebElement apply(WebDriver d){
        return d.findElement(By.xpath("//*[@id=\"nav\"]/li[4]/a"));
    }
});
elementClickLogin.click();
System.out.println("-----> Click Login Button.");

Thread.sleep(1000);
WebElement elementLogin = wait.until(new ExpectedCondition<WebElement>() {
    public WebElement apply(WebDriver d){
        return d.findElement(By.xpath("//*[@id=\"container\"]/table/tbody/tr[1]/td/center/input"));
    }
});
elementLogin.click();
elementLogin.sendKeys(userName);
System.out.println("-----> send ID.");

Thread.sleep(1000);
WebElement elementPassword = wait.until(new ExpectedCondition<WebElement>() {
    public WebElement apply(WebDriver d){
        return d.findElement(By.xpath("//*[@id=\"container\"]/table/tbody/tr[3]/td/center/input"));
    }
});
elementPassword.click();
elementPassword.sendKeys(passWord);
System.out.println("-----> send Password.");

```

ภาพที่ 3.7 Selenium Coding

และปิดท้ายด้วยคำสั่ง catch (Exception e) ซึ่งคำสั่งนี้จะคอยทำหน้าที่จับ Error ว่า กระบวนการทดสอบ นั้นไปหยุดที่ตรงไหน และส่งข้อความกลับมา ดังภาพที่ 3.8

```

readerCsv.close();
} catch (Exception e) {
    System.out.println("ERR: "+e.getMessage());
    //throw

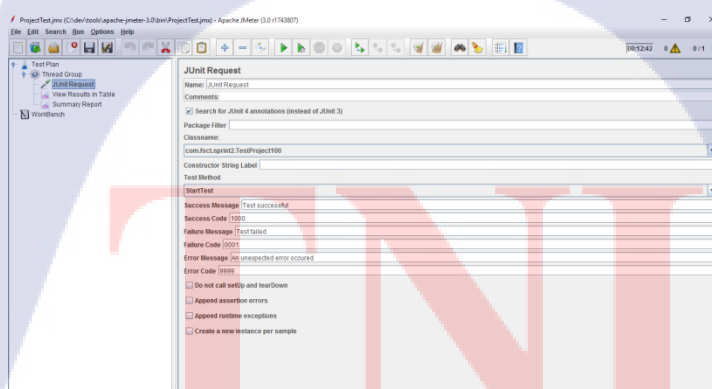
```

ภาพที่ 3.8 Coding Selenium

3.3.6 การทดสอบด้วยโปรแกรม Jmeter

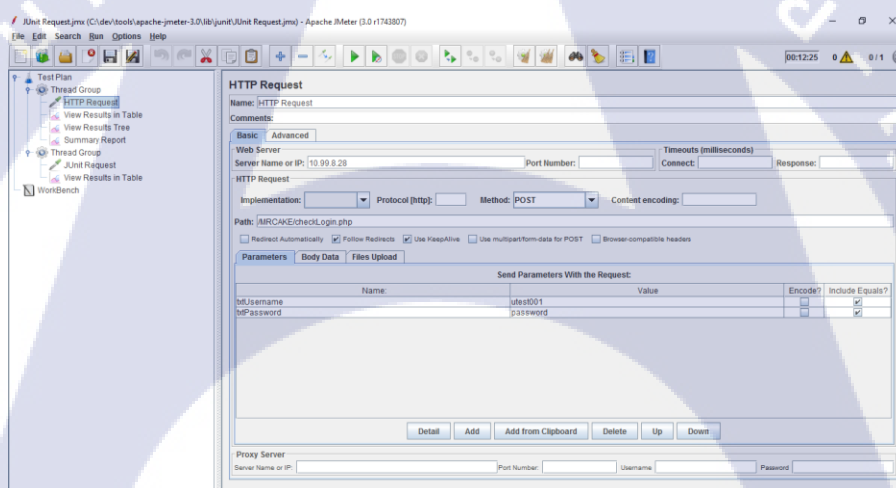
หลังทำการเขียนโค้ดโปรแกรม Selenium จนผ่านทุกขั้นตอนของกระบวนการที่จะทำการทดสอบแล้ว ให้ทำการแปลงไฟล์ในส่วนที่เขียนออกมาเป็นไฟล์ .jar เพื่อที่จะนำไปทำการทดสอบร่วมกับโปรแกรม Jmeter โดยโปรแกรม Jmeter จะเป็นตัวสั่งการทำงานทุกคำสั่งในโปรแกรม Selenium ที่เราเขียนลงไป และจะแสดงผลลัพธ์ของการทดสอบออกมา ซึ่งโปรแกรม Selenium นั้นไม่ได้แสดงผลลัพธ์ออกมาให้เราเห็น โปรแกรม Jmeter จะแสดงเวลาทั้งหมดของแต่ละผู้ใช้งานที่ทำการทดสอบว่าใช้เวลาทั้งหมดกี่นาที โดยจะแสดงในรูปแบบของ MS (Micro Second) การทำ Load Test ด้วยโปรแกรม Jmeter นั้น ได้มีการทำการทดสอบทั้งหมด 2 แบบด้วยกันคือ

1) การ Load Test ด้วย JUnit การทดสอบ ด้วย JUnit จะเริ่มได้ก็ต่อเมื่อทำการเขียนโค้ดโปรแกรม Selenium เสร็จสิ้นแล้ว เมื่อทำการเขียนโค้ดเสร็จสิ้น ให้เราทำการแปลงไฟล์ออกมาเป็นไฟล์นามสกุล .jar และนำไฟล์ที่ได้จากการแปลงไปไว้ในโฟลเดอร์ ของโปรแกรม Jmeter จากนั้น กลับมาที่โปรแกรม Jmeter ให้ทำการสร้าง Thread Group และใน Thread Group ให้เราทำการสร้าง JUnit Request เพื่อที่จะใช้สำหรับการทำการทดสอบประสิทธิภาพและสร้าง View Result in Table เพื่อใช้สำหรับการดูผลหลังจากทำการทดสอบเสร็จสิ้น ให้ไปที่ JUnit Request จากนั้นกดเลือกในช่อง Search for JUnit 4 annotation จากนั้นเลือก Classname ที่เป็นชื่อ Class ที่เรา แร่งไฟล์ออกมา จากนั้นกด start เพื่อเริ่มการทดสอบ โปรแกรม Jmeter จะทำการทดสอบในเว็บแอปพลิเคชันที่เลือกไว้ในโปรแกรม Selenium จนครบกระบวนการ ดังภาพที่ 3.9



ภาพที่ 3.9 Jmeter JUnit Request

2) Performance Test ด้วย HTTP Request การทดสอบแบบ HTTP Request นั้น ไม่จำเป็นต้องใช้โปรแกรม Selenium ในการนำทางเข้าสู่เว็บไซต์ เพียงแต่ การทดสอบแบบ HTTP Request นั้น เป็นการทดสอบประสิทธิภาพว่า หากมีการเข้าใช้งานเว็บไซต์เป็นจำนวนมากๆ เซิร์ฟเวอร์ของเว็บไซต์นั้น จะรับไหวหรือไม่ เป็นการทำการทดสอบแบบเช็คจำนวนการเข้าใช้งานเพื่อให้สามารถวางแผนการเข้าใช้งานของผู้ใช้งานและสามารถแบ่งช่วงเวลาการเข้าใช้งานได้ โดยจะเริ่มจาก ใส่ Server Name or IP จากนั้นใส่ Port หากมี และใส่ path คือช่องทางที่เราต้องการจะทำการทดสอบ ในช่องด้านล่าง ให้ไปที่หน้า parameters เลือก Add ในช่อง Name ให้ใส่ชื่อที่บ่งบอก Value และในช่อง Value ให้ใส่ค่าที่เราต้องการให้ป้อนลงไป เช่น อดีผู้ใช้งาน หรือ รหัสเวิร์ด เมื่อเพิ่มจนครบตามที่เรต้องการแล้ว ให้กดปุ่ม Start ด้านบน เพื่อให้โปรแกรมทำการทดสอบโดยโปรแกรม Jmeter จะทำการทดสอบจนครบตามจำนวนอดีผู้ใช้งานที่เรากรอกลงไป ดังภาพที่ 3.10



ภาพที่ 3.10 Jmeter HTTP Request

3.3.7 ทำผลวิเคราะห์การทดสอบ

หลังจากที่เราได้ผลลัพธ์จากการ Test ด้วย Jmeter แล้ว ให้เรานำผลลัพธ์ส่วนนั้นมาทำเป็นกราฟหรือแผนภูมิภาพ เพื่อดูค่าเฉลี่ยของเวลาที่ใช้ในการทดสอบทั้งหมด และเราสามารถนำส่วนที่เป็นกราฟ นำเสนอต่อผู้ที่พัฒนา Web Application อยู่ได้

บทที่ 4

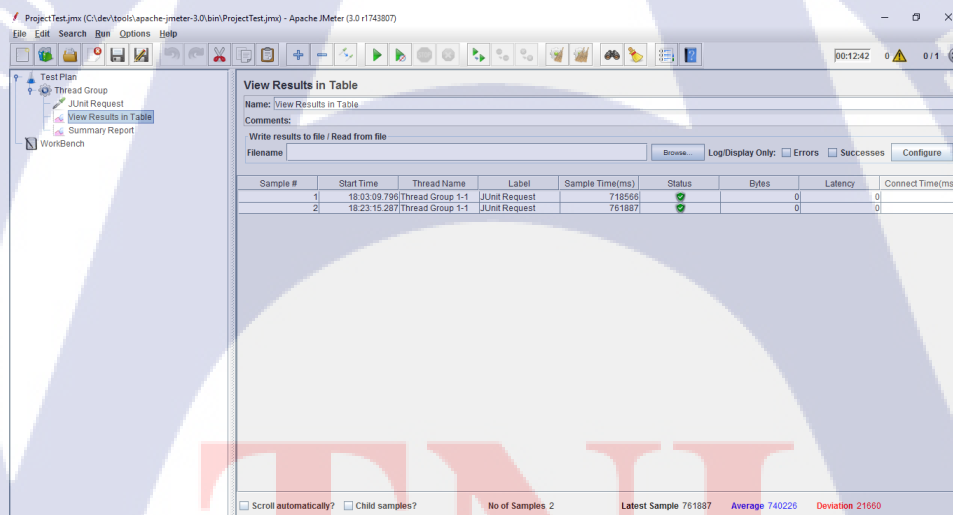
สรุปผลการดำเนินงาน การวิเคราะห์และสรุปผลต่าง ๆ

จากขั้นตอนการดำเนินโครงการทดสอบประสิทธิภาพเว็บแอปพลิเคชัน ในบทที่ 3 สามารถสรุปผลการดำเนินงานผลการวิเคราะห์ข้อมูล รวมถึงการวิจารณ์ข้อมูลโดยเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์ในการจัดทำโครงการดังนี้

4.1 ขั้นตอนและผลการดำเนินงาน

ผลการดำเนินงานของโครงการทดสอบประสิทธิภาพการใช้งานเว็บแอปพลิเคชัน

1) หน้าจอผลการทดสอบประสิทธิภาพด้วยโปรแกรม Jmeter ในประเภทของ JUnit Request ดังภาพที่ 4.1



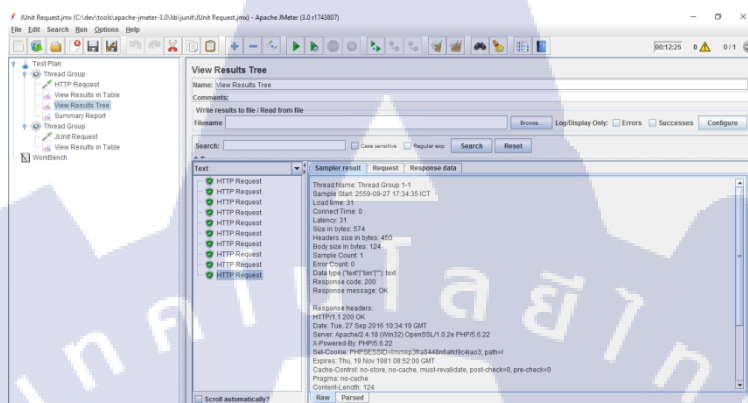
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Latency	Connected Time(ms)
1	18:03:09.796	Thread Group 1-1	JUnit Request	718566	✓	0	0	0
2	18:23:15.287	Thread Group 1-1	JUnit Request	761887	✓	0	0	0

ภาพที่ 4.1 ผลการทดสอบ Performance Test ด้วย Jmeter

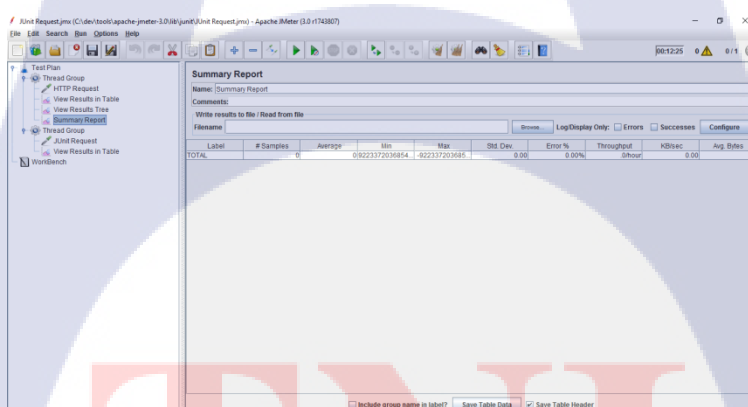
หลังจากทำการทดสอบเสร็จสิ้น โปรแกรม Jmeter จะแสดงผลการทดสอบของ JUnit ที่หน้า View Result in Table ซึ่งในหน้านี้ จะแสดงผลการทดสอบซึ่งจะแบ่งออกเป็นแต่ละ Scenario ดังภาพที่ 3.9 ได้ทำการทดสอบทั้งหมด 2 Scenario ด้วยกันคือ ทำการทดสอบ Login เข้าใช้อย่างต่อเนื่องเป็นจำนวน 50 และ

100 ไอดีผู้ใช้งาน ผลการทดสอบจะแสดงระยะเวลาทั้งหมดที่ใช้ในการทดสอบใน Scenario ที่ 1 และ 2 (หน่วยเป็น MilliSeconds) และ Status ว่ามี Scenario ที่เท่าไรที่ทำการทดสอบไม่ผ่าน

2) หน้าจอแอปพลิเคชันการทำงานของโครงการ ดังภาพที่ 4.2 – ภาพที่ 4.4



ภาพที่ 4.2 ผลการทดสอบ Load Test ด้วยโปรแกรม Jmeter



ภาพที่ 4.3 ผลการทดสอบ Load Test ด้วยโปรแกรม Jmeter

Sample #	Start Time	Thread Name	Label	Sample Time(s)	Status	Bytes	Latency	Connect Time(ms)
1	17-34-34.862	Thread Group 1-1	HTTP Request	190	Success	574	198	128
2	17-34-35.020	Thread Group 1-1	HTTP Request	27	Success	574	271	0
3	17-34-35.090	Thread Group 1-1	HTTP Request	36	Success	574	36	0
4	17-34-35.090	Thread Group 1-1	HTTP Request	10	Success	574	10	0
5	17-34-35.090	Thread Group 1-1	HTTP Request	18	Success	574	18	0
6	17-34-35.114	Thread Group 1-1	HTTP Request	13	Success	574	13	0
7	17-34-35.128	Thread Group 1-1	HTTP Request	20	Success	574	20	0
8	17-34-35.148	Thread Group 1-1	HTTP Request	66	Success	574	66	0
9	17-34-35.214	Thread Group 1-1	HTTP Request	18	Success	574	18	0
10	17-34-35.233	Thread Group 1-1	HTTP Request	31	Success	574	31	0

Summary: No of Samples: 10, Latest Sample: 31, Average: 39, Deviation: 42

ภาพที่ 4.4 ผลการทดสอบ Load Test ด้วยโปรแกรม Jmeter

เมื่อทำการทดสอบเสร็จสิ้น ระบบจะแสดงผลการทดสอบว่า แต่ละไอดีผู้ใช้งานนั้นใช้เวลาในการ Login นานเท่าไร เริ่มทำการทดสอบในแต่ละไอดีผู้ใช้งานในเวลาเมื่อใด และการทดสอบในแต่ละไอดีผู้ใช้งานนั้นผ่านหรือไม่ โดยจะแสดงในช่อง Status และใช้เวลาเฉลี่ยเท่าใดในการทำการทดสอบจนครบทุกไอดีผู้ใช้งาน

4.2 ผลการวิเคราะห์ข้อมูล

จากการทดสอบที่ได้ทำนั้น เป็นการทดสอบเพื่อทดสอบประสิทธิภาพของเว็บแอปพลิเคชันและเซิร์ฟเวอร์แอปพลิเคชันว่า หากมีจำนวนผู้ใช้เข้ามาใช้งานเว็บแอปพลิเคชันเป็นจำนวนมากเซิร์ฟเวอร์ จะสามารถทำงานได้เป็นปกติหรือไม่ จะเกิดปัญหาระหว่างการใช้งานหรือไม่ ซึ่งหากได้ผลวิเคราะห์จากการทดสอบมาแล้ว เราสามารถจัดการกับปัญหาเช่น หากมีผู้ใช้เข้ามาใช้งานจำนวนมาก เราสามารถจัดการกับปัญหานี้ได้อย่างไร เพื่อป้องกันให้เกิดปัญหากับทางเว็บเซิร์ฟเวอร์และผู้ใช้ระบบน้อยที่สุด เป็นต้น

4.3 วิจารณ์ข้อมูลโดยเปรียบเทียบผลที่ได้รับกับวัตถุประสงค์และจุดมุ่งหมายการปฏิบัติงานหรือการจัดทำโครงการ

จุดมุ่งหมายในการทำการทดสอบประสิทธิภาพของเว็บแอปพลิเคชัน เพื่อทดสอบประสิทธิภาพของเว็บแอปพลิเคชันว่า หากมีผู้ใช้งานระบบมาใช้งานเป็นจำนวนมาก เซิร์ฟเวอร์ของเว็บแอปพลิเคชันจะรับจำนวนผู้ใช้งานไหวหรือไม่ และจะทำงานได้เป็นปกติไม่มีการล่าช้าของเว็บแอปพลิเคชันเกิดขึ้นหรือไม่

ซึ่งผลที่ได้รับมาหลังจากการทำการทดสอบสามารถทำให้เราจัดการกับปัญหาจำนวนผู้ใช้งานเข้าใช้เป็นจำนวนมากจนเซิร์ฟเวอร์มีปัญหาได้และอาจมีการเพิ่มปริมาณเครื่องเซิร์ฟเวอร์เพื่อรองรับจำนวนผู้ใช้งานได้มากขึ้นรวมถึงการตรวจสอบฟังก์ชันการทำงานของเว็บแอปพลิเคชันว่าทำงานได้เป็นปกติหรือไม่



TNI

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการที่ได้ไปสหกิจศึกษาเป็นเวลา 4 เดือน ทำให้ได้รับความรู้และประสบการณ์ในการทำงานจริงมากมาย โดยงานที่ได้รับมอบหมายเป็นงานหลักสำหรับการไปสหกิจศึกษานี้คือการทำการทดสอบประสิทธิภาพคือการทดสอบประสิทธิภาพของเว็บแอปพลิเคชันที่สร้างขึ้นมา เพื่อป้องกันไม่ให้เกิดปัญหาของเซิร์ฟเวอร์ระหว่างการใช้งานเว็บแอปพลิเคชัน โดยการทดสอบจะทำทั้งหมด 2 แบบ คือ Performance Test และ Load Test ซึ่งการทดสอบทั้ง 2 แบบนั้น ไม่เหมือนกัน แบบ Performance Test จะเป็นการทดสอบประสิทธิภาพการทำงานของเว็บแอปพลิเคชันว่าหากมีการทำงานระบบนี้ซ้ำๆ กันเป็นเวลานานเซิร์ฟเวอร์จะเกิดปัญหาหรือไม่ และแบบ Load Test คือการทดสอบจำนวนผู้ใช้งานที่มีการเข้าใช้เว็บแอปพลิเคชันว่าหากมีผู้ใช้งานเข้ามาใช้งานเป็นจำนวนมาก เซิร์ฟเวอร์จะสามารถรองรับได้หรือไม่ การทดสอบทั้ง 2 แบบนี้ มีความสำคัญเหมือนกันคือทำเพื่อแก้ไขปัญหาที่จะเกิดกับเซิร์ฟเวอร์และเพื่อให้ผู้ใช้งานได้ใช้งานอย่างต่อเนื่องโดยที่ไม่มีปัญหามารบกวนระหว่างการใช้งาน ซึ่งหากเราทราบปัญหาที่เกิดขึ้นก่อนที่จะทำการนำออกไปใช้งานจริง เราก็สามารถจัดการกับปัญหานั้นได้รวดเร็วมากขึ้น

ปัญหาที่เกิดจากการทำการทดสอบ ส่วนมากเกิดจากการที่มีผู้พัฒนาอื่นกำลังทำการซ่อมแซมระบบภายในเว็บแอปพลิเคชันหรือผู้ใช้งานระบบ กำลังอยู่ระหว่างการทดสอบระบบ ทำให้ไม่สามารถทำการทดสอบได้ ซึ่งหากทำการทดสอบในช่วงเวลานี้ อาจเกิดปัญหาระหว่างการทำการพัฒนาหรือเกิดปัญหาความล่าช้าของเว็บแอปพลิเคชันได้

5.2 แนวทางการแก้ไขปัญหา

การแก้ปัญหาที่กล่าวมานั้น ทำได้เพียงแค่ออกให้ผู้พัฒนาหรือผู้ใช้งานระบบ ใช้งานให้เสร็จสิ้นในส่วนที่มีการทำการทดสอบเท่านั้น เพราะอาจจะสร้างปัญหาให้กับผู้พัฒนาระบบหรือผู้ทดสอบใช้งานระบบได้

5.3 ข้อเสนอแนะจากการดำเนินงาน

ควรศึกษาระบบของเว็บแอปพลิเคชันให้ละเอียด หากเป็นการทำการทดสอบประสิทธิภาพการใช้งานเพราะอาจจะทำให้ใช้เวลานานในการเขียนโค้ดโปรแกรม Selenium นาน หากเราไม่สามารถระบุได้ว่าเราควรนำทางให้ Selenium เข้าไปทำงานในส่วนไหนของเว็บแอปพลิเคชันซึ่งอาจจะเกิดปัญหาตามมาในระหว่างการการทดสอบเช่น ทำการทดสอบผิดระบบหรือผิดเป้าหมายที่กำหนดไว้ เป็นต้น



เอกสารอ้างอิง

วรเศรษฐ สุวรรณิก, 2556, เขียนโปรแกรม **Java** เบื้องต้น, ISBN:978-616-08-1347-6, ซีเอ็ดยูเคชั่น, หน้า 24-25.

Antonio Gomes, **Apache Jmeter**[Online], Available : <http://jmeter.apache.org/> [2013,May 5].



ภาคผนวก

TNI

ภาคผนวก ก.
รายงานประจำสัปดาห์

TNI

ประวัติผู้จัดทำโครงการ

ชื่อ – สกุล	ธนวัต ชุติไกรวัล
วัน เดือน ปีเกิด	27 มิถุนายน 2538
ประวัติการศึกษา	
ระดับประถมศึกษา	โรงเรียนจารุพัฒนานุกูล พ.ศ. 2544
ระดับมัธยมศึกษา	มัธยมศึกษาตอนต้น โรงเรียนกรพิทักษ์ศึกษา พ.ศ. 2550 มัธยมศึกษาตอนปลาย โรงเรียนบางปะกอกวิทยาคม พ.ศ. 2553
ระดับอุดมศึกษา	คณะเทคโนโลยีสารสนเทศ สาขาเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีไทย-ญี่ปุ่น พ.ศ. 2559
ทุนการศึกษา	- ไม่มี -
ประวัติการฝึกอบรม	1. Agile Methodology 2. ทำความเข้าใจการใช้ Progress Rollbase เบื้องต้น
ผลงานที่ได้รับการตีพิมพ์	- ไม่มี -