

ออกแบบการทดสอบระบบ แบบ Manual Testing
และ Re-Test ระบบโดยใช้เครื่องมือ Automation Testing

Manual Test and Automation Test using PhantomJS



สหกิจศึกษาเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

ออกแบบการทดสอบระบบ แบบ Manual Testing
และ Re-Test ระบบโดยใช้เครื่องมือ Automation Testing
Manual Test and Automation Test using PhantomJS



สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Manual Test and Automation Test using PhantomJS



COOPERATIVE EDUCATION SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR
THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE , FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงานสหกิจศึกษา ออกแบบการทดสอบระบบ แบบ Manual Testing
และ Re-Test ระบบโดยใช้เครื่องมือ Automation Testing
Manual Test and Automation Test using PhantomJS

ชื่อนักศึกษา นางสาววิชโรบล คู่กระสังข์ รหัสนักศึกษา 56050370

ปริญญา วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)

ภาควิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2559

อาจารย์ที่ปรึกษา ผศ.กฤษฎา บุศรา

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้สหกิจ
ศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ประจำปีการศึกษา 2559

| คณะกรรมการสอบ | ลายมือชื่อ |
|---|---|
| ผศ.กฤษฎา บุศรา ประธานกรรมการ และอาจารย์ที่ปรึกษา |  |

ลิขสิทธิ์ของคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
๓ 1496 9 ใ้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|-------------------------|--|
| หัวข้อโครงการสหกิจศึกษา | ออกแบบการทดสอบระบบแบบ Manual Testing และ Re-Test ระบบโดยใช้เครื่องมือ Automation Testing |
| ชื่อนักศึกษา | นางสาววัชโรบล คู่กระสังข์ รหัสนักศึกษา 56050370 |
| ปริญญา | วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์) |
| ภาควิชา | วิทยาการคอมพิวเตอร์ |
| คณะ | วิทยาศาสตร์ |
| มหาวิทยาลัย | สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) |
| ปีการศึกษา | 2559 |
| อาจารย์ที่ปรึกษา | ผศ.กฤษฎา บุศรา |

บทคัดย่อ

โครงการสหกิจศึกษานี้นำเสนอเกี่ยวกับออกแบบการทดสอบระบบ แบบทดสอบด้วยตัวเอง (Manual Testing) และการทดสอบระบบแบบวนซ้ำ (Re-Test) โดยใช้เครื่องมือนำมาช่วยในการทดสอบระบบ (Automation Testing) ซึ่งการทดสอบระบบแบบวนซ้ำ (Re-Test) อาจทำให้เกิดข้อผิดพลาด ทำให้ระบบมีประสิทธิภาพลดลง และใช้เวลาในการทดสอบระบบแบบวนซ้ำ (Re-Test) ทางบริษัทจึงได้เลือกเครื่องมือที่จะนำมาช่วยในการทดสอบระบบ (Automation Testing) คือ PhantomJS ซึ่งมีการใช้งานง่ายโดยใช้โปรแกรมที่สามารถเขียนได้ไฟล์เป็นภาษา JavaScript และเขียนคำสั่ง (Script) ให้สามารถรันต่อกันได้ และแสดงข้อมูลที่ต้องการ (Pass) และข้อมูลที่ผิดพลาด (Error) เป็นตารางผ่านไฟล์ .csv จะแสดงข้อมูลทั้งหมดให้ผู้ทดสอบระบบ (Software Tester) ได้ตรวจสอบได้ง่าย โดยการทดสอบได้ออกแบบเอกสารวางแผนการทดสอบระบบ (Test Scenario) การทดสอบระบบทั้งถูกและผิด (Test Case) และขั้นตอนการทดสอบระบบตั้งแต่แรกจนจบ (Test Script) โดยทดสอบบน SharePoint คือเว็บไซต์ที่สามารถเก็บข้อมูล จัดระเบียบและสร้างลักษณะ (Feature) ต่างๆได้หลากหลาย เพื่อเก็บข้อมูลภายในองค์กร ซึ่งจากการทดสอบระบบโดยใช้เครื่องมือ PhantomJS สามารถนำมาใช้งานได้ดี ช่วยในการลดระยะเวลาในการทดสอบแบบวนซ้ำ (Re-Test) เพื่อให้เกิดข้อผิดพลาดที่น้อยลงได้ และติดตั้งง่ายสะดวกต่อการนำไปใช้งานกับลูกค้านอกสถานที่ได้

คำสำคัญ : การทดสอบระบบทั้งถูกและผิด การทดสอบระบบแบบวนซ้ำ ผู้ทดสอบระบบ PhantomJS SharePoint

| | |
|----------------------|---|
| Title | Manual Test and Automation Test using PhantomJS |
| Students | MissWacharobol Kukrasang Student ID 56050370 |
| Degree | Bachelor of Science (Computer Science) |
| Department | Computer Science |
| Faculty | Science |
| University | King Mongkut's Institute of Technology Ladkrabang (KMITL) |
| Academic Year | 2016 |
| Advisor | Asst.Prof.Krudsada Budsara |

Abstract

This cooperative project is about designing Manual testing and re-testing systems by using equipment to test Automation Testing. Re-Test can cause a mistake such as decreasing effective, taking long time for Re-Test .Therefore, Company choose the equipment to test Automation Testing which is PhantomJS. It is easy to use by writing file program to be JavaScript, running via Command Lind, and script to be able to run in order to show pass and error data by .csv file. This file is separated by comma(.). When the text and number are recorded in csv file, it will show all data to Software Tester to check easily, know the mistaken position in order to adjust data, and can see picture including pass and error data. Test Scenario, Test Case and Test Script is designed via Sharepoint which can keep, arrange data and create several features in order to keep data of sharepoint. Testing by PhantomJS is available, make to decrease the time to Re-Test and mistake. Moreover, it is installed easily and convenient to use with customer outside.

Keywords : PhantomJS , SharePoint , Sofeware Tester , Re-Test , Test Case

กิตติกรรมประกาศ

จากการที่ข้าพเจ้าได้ไปโครงการสหกิจศึกษา โดยได้เข้าร่วมทำงานศึกษาในด้านออกแบบการทดสอบระบบ แบบ Manual Testing และ Re-Test ระบบโดยใช้เครื่องมือ Automation Testing ตั้งแต่วันที่ 1 สิงหาคม 2559 ถึง 16 ธันวาคม 2559 กับทางบริษัท Revolic Tech Co.,Ltd. โดยใช้ภาษา JavaScript เป็นภาษาหลักในการทำงานในระบบ ซึ่งส่งผลให้ได้รับประสบการณ์การทำงานในชีวิตจริง และความรู้เพิ่มมากขึ้น ซึ่งโครงการสหกิจศึกษาเล่มนี้จะเกิดขึ้นไม่ได้ หากไม่ได้รับการสนับสนุน และช่วยเหลือจากบุคลากรภายในบริษัท ที่ได้ให้คำปรึกษา แนะนำ และอำนวยความสะดวกในการดำเนินงาน

ทางข้าพเจ้าต้องขอขอบพระคุณนายอนุสรณ์ ผลภาค ตำแหน่ง Senior Software Architect นางสาววรรณฤดี สุขสงวน ตำแหน่ง Software Tester และพี่ๆ ที่บริษัททุกท่าน ที่ให้การสนับสนุน และคอยช่วยเหลือมาโดยตลอดในการทำโครงการสหกิจศึกษา และตลอดการปฏิบัติงานที่บริษัท

ขอขอบพระคุณ ผศ.กฤษฎา บุศรา ที่ได้คำปรึกษาตลอดการทำโครงการสหกิจศึกษาเล่มนี้ และอบรมให้ความรู้แก่ข้าพเจ้า ในด้านการปฏิบัติงาน และแก้ไขปัญหาต่างๆในการทำงาน

สุดท้ายนี้ ขอขอบพระคุณบิดา มารดา สมาชิกในครอบครัว ครูบาอาจารย์ รวมทั้งผู้มีพระคุณ ที่คอยให้กำลังใจ รวมถึงเพื่อนๆ และบุคคลอื่นๆ ที่ไม่ได้กล่าวมา ผู้จัดทำโครงการขอขอบคุณเป็นอย่างสูงมา ณ โอกาสนี้

วิชโรบล คู่กระสังข์

สารบัญ

| | หน้า |
|---|----------|
| บทคัดย่อภาษาไทย | ก |
| บทคัดย่อภาษาอังกฤษ..... | ข |
| กิตติกรรมประกาศ | ค |
| สารบัญ | ง |
| สารบัญตาราง..... | ฉ |
| สารบัญรูป | ช |
| คำย่อ/สัญลักษณ์ | ฅ |
| บทที่ 1 บทนำ | 1 |
| 1.1 ความเป็นมาและความสำคัญ..... | 1 |
| 1.2 วัตถุประสงค์..... | 2 |
| 1.3 ขอบเขต..... | 2 |
| 1.4 ประโยชน์ที่คาดว่าจะได้รับ..... | 3 |
| 1.5 ขั้นตอนการดำเนินงาน | 3 |
| 1.6 เครื่องมือ/ภาษาที่ใช้ในงานวิจัย | 4 |
| บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง | 5 |
| 2.1 ทฤษฎีการออกแบบการทดสอบระบบ..... | 5 |
| 2.1.1 การวางแผนการทำงาน (Test Plan)..... | 5 |
| 2.1.2 ขั้นตอนการทำงาน (Test Scenario)..... | 5 |
| 2.1.3 ขั้นตอนการทดสอบ (Test Case)..... | 5 |
| 2.1.4 กระบวนการทดสอบก่อนใช้งานจริง (Test Script (UAT))..... | 5 |
| 2.1.5 การทดสอบระบบ (Test Execution (Actual Result))..... | 6 |
| 2.1.6 การทดสอบระบบแบบวนซ้ำ (Retest Defect Issue Log)..... | 6 |
| 2.1.7 การทำคู่มือการใช้งาน (User Manual Document)..... | 6 |
| 2.2 ทฤษฎีการออกแบบระบบโดยใช้แผนภาพความสัมพันธ์ (Use Case Diagram) | 6 |
| 2.2.1 หน้าที่ที่ต้องกระทำ (Use Case) | 6 |
| 2.2.2 ผู้เกี่ยวข้องกับระบบ (Actor) | 7 |
| 2.2.3 เส้นแบ่งขอบเขตการทำงานในระบบ (System Boundary)..... | 7 |

สารบัญ (ต่อ)

| | หน้า |
|--|-----------|
| 2.2.4 เส้นการเชื่อมต่อ (Connection) | 7 |
| 2.2.5 ความสัมพันธ์แบบขยายหรือเพิ่ม (Extend Relationship) | 7 |
| 2.2.6 ความสัมพันธ์อีกรูปแบบหนึ่ง (Include Relationship) | 7 |
| 2.3 ทฤษฎีการออกแบบระบบโดยใช้แผนภาพ (Class Diagram) | 8 |
| 2.3.1 แผนภาพที่ใช้แสดง Class Diagram | 8 |
| 2.3.2 ความสัมพันธ์ระหว่าง Class diagram..... | 8 |
| 2.3.3 องค์ประกอบของ Class diagram | 8 |
| 2.3.4 สัญลักษณ์ของ Class diagram..... | 9 |
| 2.4 ความสัมพันธ์ระหว่าง Object | 9 |
| 2.4.1 ความสัมพันธ์แบบ Association..... | 9 |
| 2.4.2 ความสัมพันธ์แบบ Aggregation | 10 |
| 2.4.3 ความสัมพันธ์แบบ Composition..... | 10 |
| 2.4.4 ความสัมพันธ์แบบ Generalization..... | 10 |
| 2.4.5 ความสัมพันธ์แบบ Specialization..... | 11 |
| 2.5 หลักในการสร้าง Class Diagram | 11 |
| 2.6 ทฤษฎีของ Web SharePoint | 12 |
| 2.7 ทฤษฎีเกี่ยวกับ PhantomJS | 14 |
| 2.8 ทฤษฎีเกี่ยวกับ JavaScript | 17 |
| 2.9 ทฤษฎีเกี่ยวกับ NodeJS | 18 |
| 2.10 ทฤษฎีเรื่อง CMD | 20 |
| 2.11 ทฤษฎีเรื่อง Sublime Text 2 | 21 |
| บทที่ 3 วิธีการดำเนินงานวิจัย | 22 |
| 3.1 ขั้นตอนการศึกษาและรวบรวมข้อมูลของระบบ | 22 |
| 3.2 ขั้นตอนการวิเคราะห์ระบบ | 23 |
| 3.3 ขั้นตอนการออกแบบระบบ..... | 23 |
| 3.4 ขั้นตอนการเขียนการทดสอบระบบ (Test Case)..... | 33 |
| 3.4.1 ขั้นตอนการทดสอบระบบ | 33 |

สารบัญ (ต่อ)

| | หน้า |
|--|-----------|
| 3.5 การวางแผนการทำงาน (Work Plan) | 37 |
| 3.5.1 Week 1 – Week 2: การปรับพื้นฐาน | 37 |
| 3.5.2 Week 3: การกำหนดขอบเขต | 37 |
| 3.5.3 Week 4 - Week 5: ศึกษาขั้นตอนการทำงาน..... | 38 |
| 3.5.4 Week 6 – Week 8: การทำเอกสารการทดสอบระบบ | 38 |
| 3.5.5 Week 9 – Week 11: ทดสอบระบบด้วยตนเอง (Manual Test) | 38 |
| 3.5.6 Week 12 – Week 17: การเขียนโปรแกรม (Test Script) | 38 |
| 3.5.7 Week 18 – Week 19: ทดสอบโดยใช้เครื่องมือ (Automation Test) | 38 |
| 3.5.8 Week 20: การจัดทำเอกสารคู่มือ (User Manual Document) | 39 |
| 3.6 สถาปัตยกรรมคอมพิวเตอร์ (2-Tier Architecture) | 39 |
| บทที่ 4 ผลการวิจัยและการอภิปรายผล | 40 |
| 4.1 ผลการดำเนินงานหน้า Export Space | 40 |
| 4.2 ผลการดำเนินงานหน้า Import Space | 42 |
| 4.3 ผลการดำเนินงานหน้า Check List Space | 43 |
| บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ | 44 |
| 5.1 สรุปผลกรณีศึกษา | 44 |
| 5.2 ประโยชน์ที่ได้รับ | 44 |
| 5.2.1 ด้านองค์กร | 44 |
| 5.2.2 ด้านผู้ศึกษาและวิจัย..... | 44 |
| 5.3 ข้อเสนอแนะ | 44 |
| เอกสารอ้างอิง | 46 |

สารบัญตาราง

| ตารางที่ | หน้า |
|---|------|
| 3.1 ตารางขั้นตอนการทดสอบระบบ (Test Case)..... | 34 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

| รูปที่ | หน้า |
|--|------|
| 2.1 องค์ประกอบของ Class diagram..... | 8 |
| 2.2 สัญลักษณ์ของ Class diagram..... | 9 |
| 2.3 ความสัมพันธ์ระหว่าง Object หรือ Class แบบ 2 ทิศทาง..... | 9 |
| 2.4 ความสัมพันธ์ระหว่าง Object หรือ Class แบบ “Whole-Part”..... | 10 |
| 2.5 ความสัมพันธ์ระหว่าง Object หรือ Class แบบ Composition..... | 11 |
| 2.6 ความสัมพันธ์ระหว่าง Object หรือ Class แบบ Generalization..... | 11 |
| 2.7 ความสัมพันธ์ระหว่าง Object หรือ Class แบบ Specialization..... | 12 |
| 2.8 การเขียนโปรแกรมใช้งาน Page Loading..... | 14 |
| 2.9 การเขียนโปรแกรมใช้งาน Screen Capture แบบกำหนด URL..... | 15 |
| 2.10 การเขียนโปรแกรมใช้งาน Screen Capture แบบไม่กำหนด URL..... | 15 |
| 2.11 การเขียนโปรแกรมรันใช้งาน Screen Capture..... | 15 |
| 2.12 การเขียนโปรแกรมใช้งาน Page Automation..... | 16 |
| 2.13 การเขียนโปรแกรมใช้งาน Page Automation โดยใช้ jQuery..... | 16 |
| 2.14 การเขียนโปรแกรมใช้งาน Network Monitoring..... | 17 |
| 3.1 Flowchart Diagram แสดงขั้นตอนการทดสอบระบบแบบ Automation Test..... | 24 |
| 3.2 Flowchart Diagram แสดงขั้นตอนการทดสอบระบบแบบ Manual Test..... | 25 |
| 3.3 Flowchart Diagram แสดงการทำงานของระบบที่นำมาทดสอบ..... | 26 |
| 3.4 Flowchart Diagram แสดงการทำงานของระบบ Module Export Space..... | 27 |
| 3.5 Flowchart Diagram แสดงการทำงานของระบบ Module Import Space..... | 28 |
| 3.6 Flowchart Diagram แสดงการทำงานของระบบ Module Check List Space..... | 29 |
| 3.7 Use Case แสดงการทำงานของระบบ Automation Test..... | 31 |
| 3.8 Use Case แสดงการทำงานของระบบ Manual Test..... | 32 |
| 3.9 ตัวอย่างการเขียนแบบฟอร์มการทดสอบระบบ (Test Case)..... | 33 |
| 3.10 สถาปัตยกรรมคอมพิวเตอร์ (2-Tier Architecture) ของระบบ..... | 39 |
| 4.1 แสดงหน้าจอหลัก Page..... | 40 |
| 4.2 แสดงหน้าจอการกรอกข้อมูล SiteNo..... | 40 |

สารบัญรูป (ต่อ)

| รูปที่ | หน้า |
|---|------|
| 4.3 แสดงหน้าจอข้อมูลแบบฟอร์มที่ถูกต้อง..... | 41 |
| 4.4 หน้าแบบฟอร์มที่ถูกนำออกมาแก้ไข..... | 41 |
| 4.5 แสดงหน้าจอการเลือกไฟล์ข้อมูลและแสดงข้อมูลที่ถูกแก้ไข | 42 |
| 4.6 แสดงข้อมูลที่แก้ไข (Update) ถูกต้อง | 42 |
| 4.7 แสดงหน้าข้อมูลที่ถูกแก้ไข (Update) โดยตรวจสอบที่ LocationNameEN | 43 |
| 4.8 แบบฟอร์มในการแสดงผลข้อมูลทั้ง 3 ส่วน..... | 43 |



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในโลกของการพัฒนาโปรแกรมในปัจจุบัน จะเห็นได้ว่าโปรแกรมที่เราเขียนออกมาจะมีความซับซ้อนมากกว่าสมัยก่อนเป็นอย่างมาก ทำให้การสร้างโปรแกรมที่มีขนาดใหญ่จะต้องพึ่งความสามารถในการสร้างพอสสมควรและการทดสอบระบบ (Testing) ก็ย่อมทำได้ยากกว่าโปรแกรมเล็กๆ ในปัจจุบันเลยมีวิธีการทดสอบโปรแกรมอยู่หลายวิธีแต่ด้วยความซับซ้อนที่มากขึ้นเรื่อยๆ ของโปรแกรมที่เราออกแบบ จึงทำให้ความเสี่ยงที่จะผิดพลาดนั้นย่อมสูงขึ้นตามไปด้วยเช่นกัน

ในปัจจุบันได้ มีคำว่าทดสอบระบบโดยใช้เครื่องมือมาช่วย (Automation Testing) เกิดขึ้นมาเพื่อช่วยลดข้อผิดพลาดของมนุษย์ (Human Error) ที่อาจจะเกิดได้ในการพัฒนาโปรแกรมและการทดสอบ ซึ่งในปัจจุบันนี้เครื่องมือที่ช่วยให้เราสามารถทดสอบระบบ (Automation Testing) มีหลากหลายรูปแบบ เมื่อนำมาใช้กับการทำงาน จะทำให้สามารถทำงานได้ราบรื่นขึ้น และไม่ต้องกังวลในเรื่องโปรแกรมจะไม่สามารถทำงานได้สำเร็จ ส่วนการทดสอบระบบด้วยตนเอง (Manual Testing) ก็ยังต้องใช้ควบคู่กันอยู่ ไม่สามารถทดสอบระบบ (Testing) แบบใช้เครื่องมือมาช่วย (Automation Testing) ได้ทั้งหมด 100% เพราะยังต้องใช้การทดสอบระบบด้วยตนเอง (Manual Testing) อยู่บางส่วน แต่สามารถช่วยลดความสำคัญของการทดสอบด้วยตนเอง (Manual Testing) ลงได้ แล้วสามารถไปสนใจ กับการใช้งานโดยใช้เครื่องมือมาช่วย (Automation Testing) ได้มากกว่า

การแก้ปัญหาต่างๆ สำหรับคนที่กำลังประสบปัญหา เรื่องการเสียเวลาในการทำทดสอบระบบ (Testing) หรือโปรแกรมที่มีข้อผิดพลาดมากๆ การเริ่มทดสอบระบบโดยใช้เครื่องมือ (Automation Testing) มาใช้ จะสามารถช่วยลดระยะเวลาในการทดสอบโปรแกรมลงได้ สามารถให้ความสำคัญกับงานหลักในตัวโปรแกรมที่เรากำลังออกแบบและจัดการอยู่ จะได้ใช้เวลากับการทดสอบได้ถูกต้องและมีประสิทธิภาพกับงานมากขึ้น และเนื่องด้วยความก้าวหน้าทางด้านเทคโนโลยีมีการพัฒนาแบบไม่สิ้นสุด มีการเปลี่ยนแปลงและมีความก้าวหน้ามากขึ้นอยู่ตลอดเวลา ดังนั้นเราจึงต้องศึกษาและนำความก้าวหน้าของเทคโนโลยีมาใช้ในการออกแบบระบบของการทำงานต่างๆ เพื่อลดความยุ่งยากซับซ้อนในขั้นตอนการดำเนินงาน และลดระยะเวลาในการทำงานแบบทดสอบด้วยตนเอง (Manual Testing)

ดังนั้น ทางบริษัท เรโวลิว เทคโนโลยี จำกัด จึงได้เล็งเห็นความสำคัญของปัญหาดังกล่าว จึงเป็นที่มาของการจัดการโครงการสหกิจศึกษาเรื่อง “ออกแบบการทดสอบระบบแบบ Manual Testing และ Re-Test ระบบโดยใช้เครื่องมือ Automation Testing” ทั้งนี้โดยคาดหวังว่าจะได้นำกระบวนการ

ทดสอบโดยใช้เครื่องมือ (Automation Testing) มาช่วยในกระบวนการทำงานแบบวนซ้ำ (Re-Test) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะช่วยลดระยะเวลาในการทดสอบด้วยตนเอง (Manual Testing) และยังช่วยลดความเสี่ยงของโปรแกรมเพื่อไม่ให้เกิดข้อผิดพลาด โดยจะนำกระบวนการทดสอบโดยใช้เครื่องมือมาช่วย (Automation Testing) มาปรับใช้ทำงานควบคู่กับระบบปัจจุบัน โดยจะทดสอบบนระบบที่รองรับ (Platform) เว็บไซต์ที่สามารถรวบรวมเก็บข้อมูลภายในองค์กร(SharePoint) เพื่อระบบการทำงานได้สะดวก รวดเร็ว และมีประสิทธิภาพมากยิ่งขึ้น

1.2 วัตถุประสงค์ของงานวิจัย

- 1) เพื่อสามารถวิเคราะห์และออกแบบการทดสอบระบบ เพื่อนำมาทดสอบระบบแบบนำเครื่องมือมาช่วย (Automation Testing) ได้
- 2) เพื่อสามารถนำการทดสอบโดยใช้เครื่องมือ (Automation Testing) มาช่วยในการทดสอบระบบในกระบวนการทดสอบแบบวนซ้ำ (Re-Test) ได้
- 3) เพื่อสามารถบอกถึงข้อดีและข้อเสียของการทดสอบที่ใช้เครื่องมือ (Automation Testing) มาใช้งานในแต่ละโปรแกรมย่อยๆ (Function) เพื่อลดระยะเวลาในการทำงานให้รวดเร็วขึ้น ถูกต้องและมีประสิทธิภาพมากขึ้นได้

1.3 ขอบเขตของงานวิจัย

- 1) ศึกษากระบวนการทดสอบระบบแบบทดสอบด้วยตนเอง (Manual Testing) และการทดสอบโดยใช้เครื่องมือ (Automation Testing)
- 2) วิเคราะห์และออกแบบเอกสารการทดสอบระบบ เพื่อนำมาทดสอบโดยใช้เครื่องมือ (Automation Testing)
 - 2.1) Test Plan การวางแผนขั้นตอนการทดสอบระบบ ลักษณะที่ควรจะเป็นมุมมองกว้างๆ
 - 2.2) Test Scenario การเขียนจำลองเหตุการณ์ ที่เกิดขึ้นจริงที่ระบบจะนำไปใช้ โดยในแต่ละ Test Scenario อาจจะมีการแบ่งเป็นข้อมูลที่ถูกต้อง(Positive) และข้อมูลที่อาจจะเกิดข้อผิดพลาด (Negative)
 - 2.3) Test Case การกำหนดสถานการณ์จำลองที่จะต้องเกิด ซึ่งต้องครอบคลุมทุกกรณี เพื่อทดสอบการตอบสนอง (Response) ที่ระบบจะแจ้งเตือนแก่ ผู้ใช้งาน(user)
- 3) สามารถเลือกเครื่องมือมาใช้ทดสอบระบบแบบ Automation Testing ให้เหมาะสมกับ บนแพลตฟอร์มที่สามารถรองรับ เว็บไซต์ (SharePoint) ได้
- 4) เขียนโปรแกรม (Script) เพื่อทดสอบระบบแบบนำเครื่องมือมาใช้ (Automation Testing) บนแพลตฟอร์มที่สามารถรองรับ เว็บไซต์ (SharePoint) ได้
- 5) ทดสอบระบบแบบนำเครื่องมือมาช่วย (Automation Testing) บนแพลตฟอร์มที่สามารถรองรับ เว็บไซต์ (SharePoint) ได้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษานั่น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถทราบถึงกระบวนการทำงานของหน้าที่ ผู้ทดสอบระบบ (Software Tester)
2. สามารถทราบถึงกระบวนการทดสอบระบบแบบทดสอบด้วยตนเอง (Manual Testing) และการทดสอบโดยใช้เครื่องมือ (Automation Testing) เพื่อนำมาทดสอบบนแพลตฟอร์มที่สามารถรองรับ เว็บไซต์ (SharePoint) ได้
3. สามารถนำการทดสอบระบบโดยใช้เครื่องมือ (Automation Testing) มาช่วยในการทดสอบระบบ ในส่วนโปรแกรมย่อยๆ (Function) และ คุณสมบัติต่างๆ บนแพลตฟอร์มที่สามารถรองรับ เว็บไซต์ (SharePoint) ได้
4. สามารถนำการทดสอบระบบโดยใช้เครื่องมือ (Automation Testing) มาช่วยให้ตรวจสอบความถูกต้องของระบบ ลดความผิดพลาดในการทำงานของคน (Human Error) ได้ดีขึ้น
5. สามารถนำโปรแกรม (Script) ที่ใช้ในการทดสอบระบบโดยใช้เครื่องมือ (Automation Testing) กลับมาใช้งานในภายหลังได้อีก เมื่อมีการกลับมาใช้งานระบบที่จบไปแล้วเป็นเวลานาน หรือมีการเพิ่มโปรแกรมย่อยๆ (Function) จากระบบเดิม เพื่อดูผลลัพธ์การทำงานของระบบ

1.5 ขั้นตอนการดำเนินงาน

- 1) ศึกษาข้อมูลและความรู้เบื้องต้นเกี่ยวกับการทำงานของผู้ทดสอบระบบ (Software Tester)
 - 1.1) ศึกษาทฤษฎีการออกแบบเอกสารแบบทดสอบด้วยตนเอง (Manual Test)
 - 2) ศึกษาซอฟต์แวร์ที่ใช้ทดสอบระบบและทำความเข้าใจเกี่ยวกับระบบ
 - 2.1) ศึกษาการใช้งานของ เว็บไซต์ (SharePoint) รวมถึงกระบวนการทำงานของระบบ
 - 2.2) ศึกษาเครื่องมือที่จะนำมาใช้ในการทดสอบแบบใช้เครื่องมือ(Automation Test)
 - 2.3) ศึกษาการใช้งานระบบและภาษาที่สามารถนำมาใช้งานเพื่อทดสอบระบบ
 - 2.4) ออกแบบระบบการทำงานเพื่อเขียนโปรแกรมทดสอบโดยใช้เครื่องมือ (Automation Test)
 - 3) การเขียนโปรแกรมเพื่อนำมาใช้ในทดสอบโดยใช้เครื่องมือ (Automation Test)
 - 3.1) ติดตั้งเครื่องมือและภาษาที่นำมาใช้เขียนโปรแกรม
 - 3.2) เขียนโปรแกรมเพื่อนำมาใช้ในการทดสอบระบบ
 - 4) ทดสอบระบบหาข้อผิดพลาดและการพัฒนา
 - 4.1) แก้ไขและปรับปรุงข้อบกพร่องที่เกิดจากการทำงาน
 - 5) นำมาทดสอบใช้บนแพลตฟอร์มที่สามารถรองรับ เว็บไซต์(SharePoint) ได้
 - 5.1) ทดสอบการทำงานระบบบน Web App SharePoint
 - 6) จัดทำคู่มือการใช้งาน
 - 7) จัดทำรูปเล่มสหกิจศึกษา

เอกสารนี้เป็นเอกสารที่ส่งมอบให้แก่นักเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 เครื่องมือ/ภาษาที่ใช้ในงานวิจัย

1.6.1) ฮาร์ดแวร์

1.) เครื่องคอมพิวเตอร์ใช้เป็นเครื่องควบคุม

หน่วยประมวลผลกลาง (CPU) Core i7 4510U 2.00 GHz, 4MB L3 Cache, up to 3.10 GHz

หน่วยความจำหลัก (RAM) 8 GB

หน่วยความจำสำรอง 1TB

จอภาพ (Display) 15.6 inches Full HD 1920x1080

การ์ดจอ (Graphics Card) NVIDIA GeForce 840M

ซีดีรอม (CD-ROM) DVD Writer (Dual Layer Support)

1.6.2) ซอฟต์แวร์

1.) PhantomJS เป็นเครื่องมือที่ช่วยในการทดสอบแบบใช้เครื่องมือ (Automation Test)

2.) Node.JS ช่วยใช้เขียน JavaScript เอาไว้ที่ฝั่ง server ได้

3.) SharePoint ใช้ในการจัดการฐานข้อมูล ค้นหาข้อมูล แก้ไขข้อมูล เพิ่ม ข้อมูล ลบข้อมูล ในฐานข้อมูลที่ใช้งานในเครือข่าย

4.) Sublime Text เป็นโปรแกรมที่นำมาเขียนโค้ด .js

1.6.3) ภาษา

1.) ภาษา Java Script เป็นภาษาที่ใช้ในการเขียน Script

2.) ภาษา CMD เป็นภาษาที่ใช้ในการรัน Script เพื่อแสดงผลข้อมูลเป็น file.csv

3.) ภาษา PhantomJS มาใช้ในการเขียนจบการทำงานของฟังก์ชัน

4.) ภาษา JQuery เป็นภาษา library ที่นำมาช่วยในการหา ID ของปุ่มบน SharePoint

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในการพัฒนาระบบ Automation Test สำหรับการทำงาน เพื่อลดปัญหาการทดสอบระบบแบบ Re-Test ลดขั้นตอนความยุ่งยากแบบ Manual Test บน Web SharePoint และ Mobile นั้น ได้ทำการศึกษาค้นคว้า แนวคิด และทฤษฎีต่างๆที่เกี่ยวข้องกับการพัฒนาระบบเพื่อนำไปประยุกต์เป็นแนวทางในการพัฒนา โดยมีประเด็นตามลำดับหัวข้อดังนี้

2.1 ทฤษฎีการออกแบบการทดสอบระบบ

2.1.1 การวางแผนการทำงาน (Test Plan)

เป็นมุมมองแบบกว้างๆ (high level) ที่เอาไว้ใช้บอกเราว่า การทำ Test จะเกิดขึ้นในลักษณะไหน อะไรจะถูกนำมา Test บ้าง รวมถึง ใครเป็นคนทำ ทำอย่างไร ทำเมื่อไหร่ และ Quality level อยู่ในระดับไหนด้วย

2.1.2 ขั้นตอนการทำงาน (Test Scenario)

จะเป็นการเขียนจำลองเหตุการณ์ ที่เกิดขึ้นจริงที่ระบบจะนำไปใช้ในแต่ละ Business โดยในแต่ละ Test Scenario อาจจะมีการแบ่งเป็น Positive และ Negative ด้วยก็ได้ ส่วนมากนิยมเขียนตั้งแต่เริ่ม Business Flow จนจบ เพื่อให้แน่ใจว่า Application ที่ทดสอบสามารถนำมาใช้ควบคู่ไปกับ Business ในแต่ละเหตุการณ์ที่เกิดขึ้นจริงๆได้

2.1.3 ขั้นตอนการทดสอบ (Test Case)

การเขียน Test case จะต้องมีการกำหนดสถานการณ์จำลองที่จะต้องเกิด ซึ่งต้องครอบคลุมถึงกรณีที่เป็น Negative หรือกรณีที่ทำการการผิด กรอกข้อมูลผิดพลาดเพื่อทดสอบ Response ที่ระบบจะแจ้งเตือนแก่ user

2.1.4 กระบวนการทดสอบก่อนใช้งานจริง (Test Script (UAT))

Test script จะเป็นตัวที่บ่งบอกผลการทดสอบระบบทั้งหมด โดย UAT (User Acceptance Test) คือ กระบวนการการทดสอบระบบก่อนใช้งานจริง เพื่อตรวจสอบว่ามันสามารถตอบสนองตามความต้องการของลูกค้า (Requirement) ตรงตาม Business Flow จริงๆของลูกค้า ในระดับที่ยอมรับได้ และทดสอบที่สภาพ Environment ที่ใกล้เคียงกับ Production มากที่สุด โดยบางที่อาจจะต้องมีให้ทำเอกสารยืนยัน (UAT Sign Off) ว่าระบบใช้ได้จริง พร้อมนำไปขึ้น Production ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 การทดสอบระบบ (Test Execution (Actual Result))

ดำเนินการทดสอบตาม Test Execution Schedule ดำเนินการบันทึก, แก้ไข, ติดตาม Defect ที่เกิดขึ้น ติดตามความคืบหน้าการทดสอบ ควบคุมการทดสอบให้เป็นไปตามแผน เป็นการตรวจสอบว่าผลการทดสอบเป็นอย่างไร เช่น ได้ทำการทดสอบครบถ้วนตาม Test Execution Schedule แล้วหรือไม่ หรือมีการแก้ไข Defect ที่เกิดจากการทดสอบทั้งหมดแล้วหรือไม่ เป็นต้น และที่สำคัญยังเป็นการรายงานความคืบหน้าการทดสอบให้ทราบ อาทิเช่น

- จำนวน Test Case ที่ผ่าน/ไม่ผ่านการทดสอบ หรืออยู่ระหว่างการทดสอบ
- จำนวน Defect ที่พบจำแนกตามระดับความรุนแรง (Severity) และ สถานะ (Status)
- จำนวน Change Request (CR) ที่เกิดขึ้น เป็นต้น

2.1.6 การทดสอบแบบวนซ้ำ (Retest Defect Issue Log)

Re-testing คือ เมื่อมีการทดสอบระบบแล้ว ระบบนั้นล้มเหลวจาก defect ต่างๆ และได้มีการแก้ไขเกิดขึ้น จึงจำเป็นต้องทดสอบระบบใหม่อีกครั้งเพื่อยืนยันว่า defect นั้นๆ ได้มีการแก้ไขแล้ว สิ่งสำคัญคือ ต้องมั่นใจว่าในการทดสอบอีกครั้ง ทดสอบเหมือนกับครั้งแรกที่เคยทดสอบ ทั้ง input, ข้อมูล และ สิ่งแวดล้อมต่างๆ

2.1.7 การทำคู่มือการใช้งาน (Guideline Presentations หรือ User Manual Document)

หลักการทำคู่มือการใช้งาน (User manual)

1. อธิบายภาพรวมโดยเป็นคำบรรยาย หรือ อธิบายให้เห็นถึง องค์ประกอบโดยรวม และ นำ Function หรือ หน้าจอหลักๆ มาอธิบายการทำงานทั้งหมด (โปรแกรมนี้ใช้ทำอะไร มีประโยชน์อย่างไร)
2. อธิบาย แต่ละหน้าจอ โดยการ Capture หน้าจอ (กด Print Screen) เพื่อบันทึกภาพ หรือ อาจใช้โปรแกรมที่สำหรับเก็บภาพหน้าจอการทำงานของโปรแกรม ซึ่งต้องอธิบายอย่างละเอียดเพื่อให้ผู้ใช้เกิดความเข้าใจ
3. แต่ละภาพที่อธิบายอาจมี การกำหนดลำดับก่อน หรือหลัง เพื่อให้ผู้ใช้ เข้าใจขั้นตอนของการใช้หน้าจอ
4. เมื่อเขียนคู่มือจนเสร็จแล้ว ควรใส่ชื่อผู้พัฒนาโปรแกรมลงไปด้วย

2.2 ทฤษฎีการออกแบบระบบโดยใช้แผนภาพความสัมพันธ์ (Use Case Diagram)

2.2.1 หน้าທີ່ระบบต้องกระทำ (Use Case) คือ ใช้สัญลักษณ์รูปวงรี พร้อมทั้งเขียนชื่อ Use Case ซึ่งต้องใช้คำกริยาหรือกริยาวลีก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ผู้เกี่ยวข้องกับระบบ (Actor) ซึ่งรวมทั้ง Primary Actor และ Stakeholder Actor ที่เป็นมนุษย์ ในที่นี้จะใช้สัญลักษณ์รูปคน (Stick Man Icon) เหมือนกัน พร้อมทั้งเขียนชื่อ Actor ไว้ด้านล่างของสัญลักษณ์ด้วย แต่หากเป็น Actor ที่ไม่ใช่มนุษย์ เช่น ระบบงานอื่นที่อยู่นอกเหนือระบบที่เราสนใจ จะใช้รูปสี่เหลี่ยมแล้วเขียนคำว่า “<<actor>>” ไว้ด้านบนแทน

2.2.3 เส้นแบ่งขอบเขตการทำงานในระบบ (System Boundary) เส้นแบ่งขอบเขตระหว่างระบบกับผู้กระทำต่อระบบ (Use Case กับ Actor) ใช้รูปสี่เหลี่ยมเป็นสัญลักษณ์ พร้อมทั้งเขียนชื่อระบบไว้ด้านใน ซึ่งสำคัญมากทุกการเขียนจะ ต้องไม่ลืมเขียน System Boundary

2.2.4 เส้นการเชื่อมต่อ (Connection) คือ เส้นที่ลากเชื่อมต่อระหว่าง Actor กับ Use Case ที่มีปฏิสัมพันธ์กัน ใช้เส้นตรงไม่มีหัวลูกศรเป็นสัญลักษณ์ของ Connection ส่วน Connection ที่ใช้เชื่อมต่อระหว่าง Use Case กับ Use Case กรณีที่ Use Case นั้นมีความสัมพันธ์ซึ่งกันและกัน จะใช้สัญลักษณ์เส้นตรงมีหัวลูกศรพร้อมทั้งเขียนชื่อความสัมพันธ์ไว้ตรงกลางเส้นด้วย โดยเขียนไว้ภายในเครื่องหมาย <<...>>

2.2.5 ความสัมพันธ์แบบขยายหรือเพิ่ม (Extend Relationship) เป็นความสัมพันธ์แบบขยายหรือเพิ่ม เกิดขึ้นในกรณีที่บาง Use Case ดำเนินกิจกรรมของตนเองไปตามปกติ แต่อาจจะจะมีเงื่อนไขหรือสิ่งกระตุ้นบางอย่างที่ส่งผลให้กิจกรรมตามปกติของ Use Case นั้นถูกรบกวนจนเบี่ยงเบนไป ซึ่งเราสามารถแสดงเงื่อนไขหรือสิ่งกระตุ้นเหล่านั้นได้ในรูปของ “Use Case” และเรียกความสัมพันธ์ระหว่าง Use Case ในลักษณะนี้ว่า “Extend Relationship” โดยเรียก Use Case ที่ถูกรบกวนหรือ Use Case ที่ดำเนินงานตามปกติว่า “Base Use Case” และเรียก Use Case ที่ทำหน้าที่รบกวนหรือกระตุ้น Base Use Case ว่า “Extending Use Case” ซึ่งการเขียนสัญลักษณ์ Extend Relationship จะเขียนใน Connection เช่น <<extend>>

2.2.6 ความสัมพันธ์อีกรูปแบบหนึ่ง (Include Relationship) ความสัมพันธ์อีกรูปแบบหนึ่งของ Use Case Diagram ก็คือ ความสัมพันธ์แบบเรียกใช้เกิดขึ้นในกรณีที่ Use Case หนึ่งไปเรียกหรือดึงกิจกรรมของอีก Use Case หนึ่งมาใช้เพื่อให้กิจกรรมนั้นเกิดขึ้นจริงใน Use Case ของตนเอง หรือกล่าวให้ง่ายกว่านั้นคือกิจกรรมใน Use Case หนึ่ง อาจจะถูกผนวกเข้าไปรวมกับกิจกรรมของอีก Use Case หนึ่ง เราเรียกความสัมพันธ์ระหว่าง Use Case ในลักษณะนี้ว่า “Include Relationship” โดย Use Case ที่ทำหน้าที่ดึงกิจกรรมมาจาก Use Case อื่นๆ เรียกว่า “Base Use Case” ในขณะที่ Use Case ที่ถูกเรียก หรือถูกดึงกิจกรรมมาใช้ เรียกว่า “Included Use Case” สามารถเขียนเส้น Connection ได้ในทิศทางตรงกันข้ามกับ Extend Relationship โดยเริ่มต้นลากเส้นตรงจาก Base Use Case หันลูกศรชี้ไปที่ Included Use Case แล้วเขียนชื่อว่า <<include>> ไว้ตรงกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ทฤษฎีการออกแบบระบบโดยใช้แผนภาพที่แสดง class และความสัมพันธ์ (relationship) ระหว่าง class (Class Diagram)

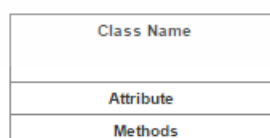
2.3.1 แผนภาพที่ใช้แสดง (Class Diagram) คือแผนภาพที่ใช้แสดง Class และความสัมพันธ์ในแง่ต่างๆ (Relation) ระหว่าง Class เหล่านั้น ซึ่งความสัมพันธ์ที่กล่าวถึงใน Class Diagram นี้ถือเป็นความสัมพันธ์เชิงสถิต (Static Relationship) หมายถึงความสัมพันธ์ที่มีอยู่แล้วเป็นปกติในระหว่าง Class ต่างๆ ไม่ใช่ความสัมพันธ์ที่เกิดขึ้นเนื่องจากกิจกรรมต่างๆ ซึ่งเรียกว่าความสัมพันธ์เชิงกิจกรรม (Dynamic Relationship) สิ่งที่ปรากฏใน Class Diagram นั้นประกอบด้วยกลุ่มของ Class และกลุ่มของ Relationship โดยสัญลักษณ์ที่ใช้ในการแสดง Class นั้นจะแทนด้วยสี่เหลี่ยมแบ่งออกเป็น 3 ส่วน โดยแต่ละส่วนนั้น (จากบนลงล่าง) จะใช้ในการแสดง ชื่อของ Class , Attribute และฟังก์ชันต่างๆตามลำดับ

2.3.2 ความสัมพันธ์ระหว่าง Class

ความสัมพันธ์ระหว่าง Class (Relationship) คือ ความสัมพันธ์ระหว่าง Class ที่ทำงานร่วมกัน สามารถจำแนกได้ดังนี้

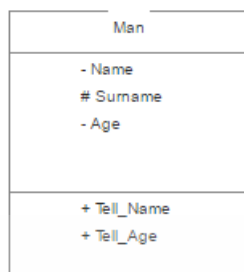
- ความสัมพันธ์แบบพึ่งพา (Dependency) เช่น “Class ลูกค้า” กับ “Class ขายสินค้า” กล่าวได้ว่า “Class ขายสินค้า” ขึ้นอยู่กับ “Class ลูกค้า” เพราะเมื่อลูกค้ามีการเปลี่ยนแปลงคำสั่งซื้อ หรือคำสั่งผลิตรายการขายก็จะต้องถูกเปลี่ยนแปลง (Update) ตามลูกค้า
- ความสัมพันธ์แบบสืบทอดคุณสมบัติ (Inheritance) เช่น “Class แม่” (super class) สืบทอดคุณลักษณะเฉพาะที่ตนมีอยู่ไปยัง “Class ลูก” (sub class)
- ความสัมพันธ์แบบร่วมกัน (Association) คือความสัมพันธ์ที่เกี่ยวข้องมีความสัมพันธ์ซึ่งกันและกัน เช่น “Class นักเรียน” สัมพันธ์กับ “Class รายวิชา” ในเรื่องของการลงทะเบียนเรียน

2.3.3 องค์ประกอบของ Class diagram



รูปที่ 2.1 องค์ประกอบของ Class diagram

2.3.4 สัญลักษณ์ของ Class diagram



รูปที่ 2.2 สัญลักษณ์ของ Class diagram

ในการเขียนสัญลักษณ์แทน Class สิ่งที่ต้องคำนึงถึงอีกสิ่งหนึ่งคือระดับการเข้าถึงเรียกสัญลักษณ์ที่ใช้แทนการเข้าถึงนี้ว่า Visibility แบ่งออกได้เป็น 3 ประเภท ได้แก่

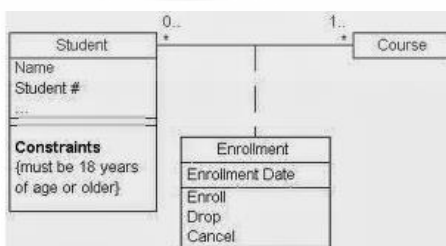
1. **Private** เขียนแทนด้วยสัญลักษณ์ - หมายถึง Attribute หรือ ฟังก์ชัน ที่ไม่สามารถมองเห็นได้จากภายนอก แต่สามารถมองเห็นได้จากภายในตัวของ Class เองเท่านั้น

2. **Protect** เขียนแทนด้วยสัญลักษณ์ # หมายถึง Attribute หรือ ฟังก์ชัน ที่สงวนไว้สำหรับการทำ Inheritance โดยเฉพาะ Attribute หรือ ฟังก์ชันเหล่านี้ จะเป็นของ Super class เมื่อทำการ Inheritance แล้ว Attribute หรือ ฟังก์ชัน ที่มี Visibility แบบ Protect จะกลายเป็น Private Attribute/ฟังก์ชัน หรือ Protected ขึ้นอยู่กับภาษา Programming ที่นำไปใช้ 3.

3. **Public** เขียนแทนด้วยสัญลักษณ์ + หมายถึง Attribute หรือ ฟังก์ชัน ที่สามารถมองเห็นได้จากภายนอก และสามารถเข้าไปเปลี่ยนค่า อ่านค่าหรือเรียกใช้งาน Attribute หรือ ฟังก์ชัน นั้นได้ทันทีโดยอิสระจากภายนอก (โดยทั่วไปแล้ว Visibility แบบ Public มักจะใช้กับฟังก์ชันมากกว่า Attribute)

2.4 ความสัมพันธ์ระหว่าง Object ประกอบด้วย

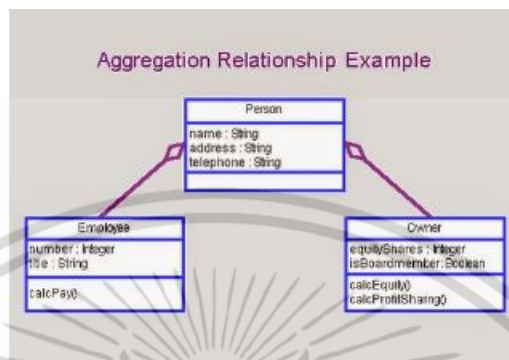
2.4.1 ความสัมพันธ์แบบ Association เป็นความสัมพันธ์ระหว่าง Object หรือ Class แบบ 2 ทิศทาง



รูปที่ 2.3 ความสัมพันธ์ระหว่าง Object หรือ Class แบบ 2 ทิศทาง

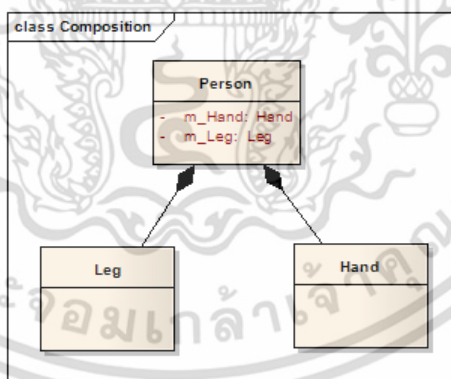
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ความสัมพันธ์แบบ Aggregation เป็นความสัมพันธ์ระหว่าง Object หรือ Class แบบ “Whole-Part” หรือ “is part of” โดยจะมี Class ที่ใหญ่ที่สุดที่เป็น Object หลัก และมี Class อื่นเป็นส่วนประกอบ



รูปที่ 2.4 ความสัมพันธ์ระหว่าง Object หรือ Class แบบ “Whole-Part”

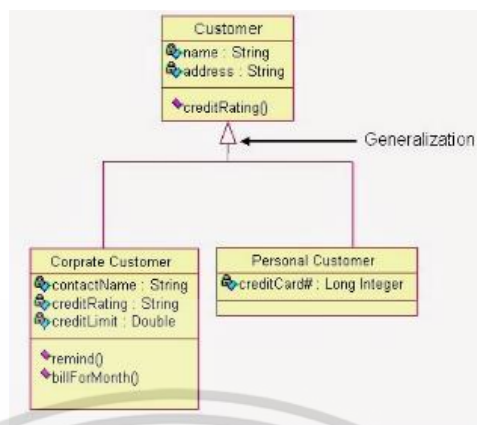
2.4.3 ความสัมพันธ์แบบ Composition เป็นความสัมพันธ์ระหว่าง Object หรือ Class แบบขึ้นต่อกันและมีความเกี่ยวข้องกันเสมอ โดยจะมี Class ซึ่งเป็นองค์ประกอบของ Class อื่นที่ใหญ่กว่า เมื่อ Class ที่ใหญ่กว่าถูกทำลาย Class ที่เป็นองค์ประกอบก็จะถูกทำลายไปด้วย



รูปที่ 2.5 ความสัมพันธ์ระหว่าง Object หรือ Class แบบ Composition

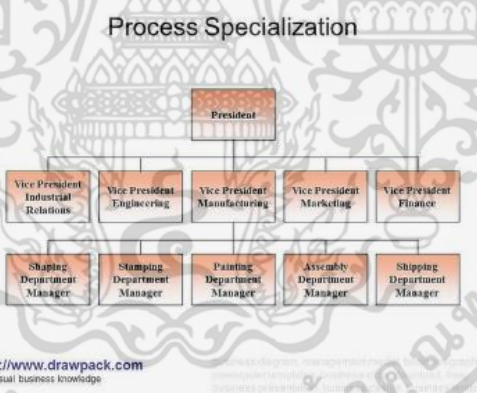
2.4.4 ความสัมพันธ์แบบ Generalization เป็นความสัมพันธ์ระหว่าง Object หรือ Class ในลักษณะของการสืบทอดคุณสมบัติจาก Class หนึ่ง (Super class) ไปยังอีก Class หนึ่ง (Subclass)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 ความสัมพันธ์ระหว่าง Object หรือ Class แบบ Generalization

2.4.5 ความสัมพันธ์แบบ Specialization คือกระบวนการที่ตรงกันข้ามกับ กระบวนการ Generalization Abstraction คือกระบวนการที่ตรงกันข้ามกับกระบวนการ Generalization Abstraction กล่าวคือ ถ้าต้องการสร้าง Class ใหม่ โดยอาศัย Concept ของ Class เก่าบางส่วน และเพิ่มเติมใหม่บางส่วนจนเกิดเป็น Class ใหม่



รูปที่ 2.7 ความสัมพันธ์ระหว่าง Object หรือ Class แบบ Specialization

2.5 หลักในการสร้าง Class Diagram

สิ่งที่ต้องคำนึงถึงสำหรับการจำลอง Class และ Relationship ต่างๆใน Class Diagram ใน OOA คือ

- 1.) Class ทั้งหมดที่ต้องมีอยู่ในระบบหรือใน Real World
- 2.) ต้องมีอยู่ครบ ไม่ขาดหาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.) ไม่มากเกินไปจนความจำเป็น

1. กำหนดกรอบของ Problem Domain ให้ชัดเจน
 - 1.) ให้ยึดถือ Problem Domain นี้เป็นบรรทัดฐานในการวิเคราะห์ระบบ
 - 2.) เขียน Use Case Diagram ของ Problem Domain
 - 3.) พิจารณาว่า ในแต่ละ Use Case จะมี Objects อะไรอยู่บ้าง
2. พิจารณาหา Objects ที่สามารถจับต้องได้เห็นได้สัมผัสได้ ซึ่งเรียกว่า Tangible Objects
3. พิจารณาหา Objects ที่ไม่สามารถจับต้องได้ซึ่งเรียกว่า Intangible Objects
4. ใช้ Classification Abstraction เพื่อแยกแยะและสร้าง Class จาก Objects ที่มีอยู่
 - 1.) พยายามหา Attributes และ Functions ของ Class เท่าที่จะหาได้
 - 2.) วาด Class ทั้งหมดที่ได้ ลงใน Class Diagram
5. หา Aggregation Abstraction (โดยพิจารณาการเป็นส่วนประกอบ)
 - 1.) เพิ่มเติมสัญลักษณ์
 - 2.) ใส่ Cardinality ให้ถูกต้อง
6. ใช้ Generalization มาพิจารณา
 - 1.) เพิ่มเติมสัญลักษณ์
 - 2.) อาจเกิด Class ใหม่เพื่อเป็น Generalized Class ได้
7. ใช้ Association มาพิจารณา
 - 1.) เพิ่มเติมสัญลักษณ์
 - 2.) พิจารณาประเภทของความสัมพันธ์และ Cardinality ให้ถูกต้อง
8. พิจารณา Class Diagram ว่ามี Class หรือ กลุ่มของ Class ที่ไม่มีความสัมพันธ์กับ Class อื่นๆ หรือไม่
 - 1.) อาจจะมี Class ที่ไม่จำเป็นสำหรับระบบ
 - 2.) อาจจะมี Class อื่นๆที่จำเป็นในระบบ

2.6 ทฤษฎีของ Web SharePoint

กลุ่ม SharePoint คือ กลุ่มของผู้ใช้ SharePoint โดยทั่วไปแล้วบุคคลทั้งหมดในกลุ่ม SharePoint จะทำงานที่คล้ายคลึงกันบนเว็บไซต์และมีความต้องการระดับสิทธิ์ที่เหมือนกัน คุณสามารถจัดกลุ่มผู้ใช้ ออกเป็นกี่กลุ่มก็ได้ ตามขนาดและความสลับซับซ้อนขององค์กรหรือไซต์ของคุณ โดยทั่วไปแล้วผู้ดูแลระบบ SharePoint จะสร้างและจัดการกลุ่ม SharePoint ที่ระดับไซต์คอลเลกชัน

การจัดเป็นกลุ่มต่างๆ ช่วยให้คุณสามารถรวมบุคคลที่ใช้ไซต์ในแบบเดียวกันได้อย่างง่ายดาย ผู้ใช้บางคนต้องการแคร์ริวเนื้อหาบนไซต์ บางคนอาจต้องการแก้ไขเนื้อหา และบางคนอาจต้องการเพิ่มหรือแก้ไขเนื้อหาหรือองค์ประกอบอื่นๆ ของไซต์ ไดอะแกรมต่อไปนี้แสดงผู้ใช้กลุ่มต่างๆ และชนิดของการเข้าถึงที่จำเป็นต้องมี ทั้งนี้ขึ้นอยู่กับความซับซ้อนของความต้องการทางธุรกิจ องค์กรหลายแห่งสร้างกลยุทธ์สำหรับการจัดการการเข้าถึงทรัพยากรเอง ตัวอย่างเช่น คุณอาจมีไซต์ย่อยพิเศษสำหรับเก็บข้อมูลที่มีความสำคัญ และออกแบบโซลูชันที่ไม่เหมือนใครเพื่อเก็บข้อมูลนั้นให้ปลอดภัยบนไซต์ย่อย Web Application ที่ทำหน้าที่ในการให้บริการการสื่อสาร การจัดการเนื้อหา การอำนวยความสะดวกให้กับทีม โดยการสร้าง website จาก template สำเร็จรูป เพื่อรวบรวมเอกสารต่างๆ เข้าด้วยกันสำหรับการทำงานร่วมกันเป็นทีม เป็นโปรแกรมที่ช่วยในเรื่อง ของการบริหารจัดการทรัพยากรต่างๆ ขององค์กรให้มีประสิทธิภาพมากยิ่งขึ้น กล่าวคือ เป็นการรวบรวมสถานที่ในการจัดเก็บเอกสาร และ ทรัพยากรอื่นๆ ขององค์กรไว้ที่แหล่งเดียวกัน, สร้าง Workflows, กำหนด ขั้นตอนการดำเนินงานในรูปแบบอัตโนมัติ, การสื่อสารกันภายในองค์กรโดยอาศัย Blogs และ Wikis, เพิ่มวิสัยทัศน์ในการท างานของผู้ใช้งานและทีม โดยอาศัย Team Sites และ My Sites อีกทั้งยังช่วยให้ผู้ใช้งานสามารถใช้ Office SharePoint Server 2007 เป็นสื่อกลางในการแสดงความคิดเห็นในการ ทำงาน แบ่งปันความรู้ เพื่อก่อให้เกิดมาตรฐานการตัดสินใจในธุรกิจได้ดียิ่งขึ้น

Microsoft SharePoint คือชุดผลิตภัณฑ์ของบริษัท Microsoft สำหรับการประสานงานร่วมกัน และบริหารเอกสารผ่าน web browser ทำหน้าที่เสมือนเป็นตัวกลางการเก็บ, แลกเปลี่ยน, ค้นหาข้อมูลที่ใช้ภายใน องค์กร ซึ่งหน้าที่หลักของมันคือการสร้างพื้นที่ทำงานร่วมกัน ให้แชร์เอกสาร และมีการเชิญสมาชิกเข้ามาสู่พื้นที่ทำงานร่วมกัน โดยทุกอย่างเป็นระบบเซิร์ฟเวอร์เป็นตัวจัดการทั้งหมด

ประโยชน์ของ Microsoft SharePoint

1. แชร์พื้นที่การทำงาน ให้ระบบการทำงานที่รวมเป็นหนึ่งสำหรับเว็บไซต์ธุรกิจทั้งหมดของคุณ คุณสามารถใช้เอกสารร่วมกับผู้ร่วมงาน จัดการโครงการร่วมกับบริษัทคู่ค้า และประกาศข้อมูลให้ลูกค้าทราบได้
2. มีเครื่องมือการทำงานร่วมกันที่มีประสิทธิภาพ และ platform เพียงหนึ่งเดียวสำหรับการจัดการเครื่องมือต่างๆ ทุกคนจะสามารถแลกเปลี่ยนความคิดเห็นและทำงานร่วมกันอย่างที่ต้องการได้อย่างง่ายดาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ช่วยให้ผู้ใช้ทุกคนเข้าถึงข้อมูลในฐานข้อมูล รายงาน และโปรแกรมประยุกต์ทางธุรกิจได้ ทั้งยังช่วยในการค้นหาตำแหน่งที่ตั้งของข้อมูลที่ใช้ต้องการเพื่อใช้ ประกอบการตัดสินใจได้
4. ทำให้การจัดการเนื้อหาเป็นเรื่องง่าย เพียงกำหนดมาตรการด้านการปฏิบัติตามข้อกำหนดที่ทำงานอยู่เบื้องหลัง ด้วยคุณลักษณะต่างๆ เช่น ชนิดเอกสาร นโยบายการเก็บรักษา และการเรียงลำดับเนื้อหาอัตโนมัติ แล้วให้ผู้ใช้ทำงานได้ตามปกติใน Microsoft Office
5. ระบบที่สามารถเก็บข้อมูล แชรข้อมูล ค้นหา แลกเปลี่ยน รวมถึงการจัดการต่างๆ ไว้เป็นศูนย์กลาง และใช้ได้บน Web Browser ทั้งหมดโดยที่เครื่อง Client ของคุณไม่ต้องติดตั้ง Software ใดๆทั้งสิ้น ทำให้ประหยัดด้านค่าใช้จ่าย ด้านเวลา และง่ายในการจัดการ Web Portal ของคุณ

2.7 ทฤษฎีเกี่ยวกับ PhantomJS

PhantomJS โดยปกติแล้ว เวลาเราจะเข้าเว็บเราก็จะต้องเปิด web browser ขึ้นมา แล้วก็ใส่ url ลงไป แต่ PhantomJS นั้นจะแตกต่างจาก web browser อื่นๆ ตรงที่มันจะต้องรันผ่าน command-line PhantomJS นั้นเป็น tool ที่ใช้สำหรับเข้าเว็บเหมือนกับ Chrome, Internet Explorer หรือ Firefox เลย แม้ฟังแล้วอาจดูเหมือนไม่มีอะไรน่าสนใจนัก แต่ความพิเศษของมันนั้นอยู่ตรงที่การรันผ่าน command-line ซึ่งมาพร้อมกับ JavaScript API ที่จะทำให้เราสามารถจัดการกับผลลัพธ์ของเว็บนั้นๆ ได้ด้วย ด้วยความสามารถในการจัดการกับผลลัพธ์ของหน้าเว็บได้ คนจึงนิยมนำ PhantomJS ไปใช้ใน งานต่างๆ ดังนี้

- 1.) **Page Loading** สามารถเปิดหน้าเว็บนั้นเอง โดยเราจะสามารถระบุ URL ของหน้าเว็บที่ต้องการจะเปิดได้ พร้อมกับกำหนด callback function เพื่อที่จะให้ทำอะไรบางอย่าง เมื่อหน้าเว็บนั้นได้ถูกโหลดมาเรียบร้อยแล้ว ลองดูตัวอย่างโค้ดด้านล่างนี้

```
var page = require('webpage').create();
// เปิดหน้าเว็บ http://www.siamhtml.com ขึ้นมา
page.open('http://www.siamhtml.com', function() {
  // ทำอะไรกับผลลัพธ์ที่ได้ ?
});
```

รูปที่ 2.8 การเขียนโปรแกรมใช้งาน Page Loading

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) Screen Capture การเซฟภาพหน้าเว็บเป็นไฟล์ภาพได้ รวมไปถึงกราฟิกอย่าง SVG หรือ Canvas เราก็แค่เพิ่มโค้ดลงไป ใน callback function เพียงไม่กี่บรรทัดแบบนี้

```
var page = require('webpage').create();

page.open('http://www.siamhtml.com', function() {
  page.render('siamhtml.png'); // เซฟหน้าเว็บเป็นไฟล์ siamhtml.png
  phantom.exit(); // สั่งให้ phantomJS เลิกทำงาน
});
```

รูปที่ 2.9 การเขียนโปรแกรมใช้งาน Screen Capture แบบกำหนด URL

หากไม่ต้องการกำหนด URL ลงไปในโค้ดเลย เราก็อาจจะให้อ่านจาก argument ที่จะส่งเข้าไปตอนรัน PhantomJS ก็ได้ ให้แก่โค้ดเป็นแบบนี้

```
var page = require('webpage').create(),
    system = require('system'),
    url = system.args[1], // url คือ argument ตัวที่ 1
    filename = system.args[2] || 'screenshot';

page.open(url, function() {
  page.render(filename + '.png');
  phantom.exit();
});
```

รูปที่ 2.10 การเขียนโปรแกรมใช้งาน Screen Capture แบบไม่กำหนด URL

ที่นี้เวลาจะรัน ก็ให้กำหนด URL และชื่อของไฟล์รูปที่ต้องการจะเซฟไปด้วย แบบนี้

```
phantomjs someScript.js http://www.siamhtml.com siamhtml
```

รูปที่ 2.11 การเขียนโปรแกรมรันใช้งาน Screen Capture

จากโค้ดด้านบน จะได้ว่า `http://www.siamhtml.com` และ `siamhtml` จะเป็น argument ตัวที่ 1 และ 2 ตามลำดับ (url และ filename)

3.) Page Automation สามารถจัดการกับ DOM Manipulation ของหน้าเว็บนั้นๆ ได้ โดยจะทำผ่านฟังก์ชัน evaluate() สมมติ อยากรู้ว่า title ของหน้าเว็บนั้นคืออะไร ก็จะเขียนแบบนี้

```
var page = require('webpage').create();
page.open('http://www.siamhtml.com', function() {
  var title = page.evaluate(function() {
    return $('title').html(); // ดึงค่า title ออกมาด้วย jQuery
  });
  console.log('Page title is ' + title);
  phantom.exit();
});
```

รูปที่ 2.12 การเขียนโปรแกรมใช้งาน Page Automation

จากตัวอย่างด้านบน จะสังเกตว่าดึง title มาโดยการใช้ jQuery ซึ่งถ้าหน้าเว็บนั้นๆ มีการใช้ jQuery อยู่แล้วก็จะไม่มีปัญหาอะไร แต่ถ้ายังไม่มี ก็จะต้องใส่ jQuery ให้มันก่อนโดย ใช้ includeJS() แบบนี้

```
var page = require('webpage').create();
page.open('http://www.siamhtml.com', function() {
  // โหลด jQuery ใส่หน้าเว็บ
  page.includeJs('//code.jquery.com/jquery-1.11.0.min.js', function() {
    var title = page.evaluate(function() {
      return $('title').html();
    });
    console.log('Page title is ' + title);
    phantom.exit();
  });
});
```

รูปที่ 2.13 การเขียนโปรแกรมใช้งาน Page Automation โดยใช้ jQuery

และจากความสามารถเกี่ยวกับ DOM Manipulation นี้เอง ที่ทำให้สามารถใช้ PhantomJS ในการทำ Page Automation ได้ เช่น หลังจากโหลดหน้าเว็บมาแล้ว ก็ให้กรอกข้อความลงในฟอร์ม แล้วสั่งให้ submit ฟอร์มนั้นๆ เป็นต้น หรือจะใช้เพื่อประโยชน์ในการ test ฟังก์ชันต่างๆ ก็ได้

4.) Network Monitoring สามารถตรวจดูได้อย่างละเอียดว่าในการโหลดหน้าเว็บนั้นๆ มา มีข้อมูล request และ receive อะไรบ้าง ซึ่งจะมีประโยชน์มากในการใช้วิเคราะห์เรื่อง performance ของหน้าเว็บ ให้เราลองรันโค้ดด้านล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

var page = require('webpage').create();

page.onResourceRequested = function(request) {
  console.log('Request ' + JSON.stringify(request, undefined, 4));
};
page.onResourceReceived = function(response) {
  console.log('Receive ' + JSON.stringify(response, undefined, 4));
};

page.open('http://www.siamhtml.com');

```

รูปที่ 2.14 การเขียนโปรแกรมใช้งาน Network Monitoring

เมื่อลองดูผลลัพธ์ที่ได้ เราก็จะเห็นข้อมูลดิบออกมามากมาย ซึ่งเราสามารถนำมาคำนวณเพื่อทำเป็นรายงานได้

2.8 ทฤษฎีเกี่ยวกับ JavaScript

JavaScript เป็นภาษาที่เป็น Script ที่อยู่ในเว็บไซต์ (ใช้ร่วมกับ HTML) เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ยกตัวอย่าง Hellomyweb.com ตรงเมนูด้านซ้ายมือจะเห็นว่าสามารถคลิกเพื่อดูหัวข้อภายในได้ และสามารถคลิกที่ลูกศรสีเขียวเพื่อปิดดูทั้งหมด และลูกศรสีแดงเพื่อเปิดทั้งหมด ข้อดีของ JavaScript คือสามารถทำให้ผู้ใช้งานใช้เว็บไซต์ของเราได้ง่ายขึ้น รวมถึงดึงดูดความสนใจของผู้ใช้งานได้อีกด้วย ปัจจุบันนี้ JavaScript นั้นเป็นมาตรฐานที่อยู่ใน W3C จึงมั่นใจได้ว่าทุกๆ Web browser รองรับการทำงานของ JavaScript แน่แน่นอน เนื้อหาเบื้องต้นที่ผู้ใช้ต้องเข้าใจมาก่อนล่วงหน้าคือ HTML เพื่อให้สามารถทำความเข้าใจในเนื้อหาเรื่องนี้ได้ง่ายขึ้น

- JavaScript นั้นออกแบบให้ใช้งานร่วมกับ HTML นั่นคือต้องอยู่รวมไปกับ HTML Code
- JavaScript เป็น script language ทำให้ผู้ใช้งานใช้งานได้ง่าย ไม่จำเป็นต้องมีพื้นฐานโปรแกรมมากนัก
- JavaScript เป็นภาษาที่ใช้ทรัพยากรเครื่องน้อยมาก (JavaScript นั้นจะประมวลผลที่ฝั่งของเครื่องผู้ใช้ทำให้ไม่เป็นภาระกับเครื่องมากนักเมื่อเทียบกับ Flash
- JavaScript Java กับ JavaScript แตกต่างกัน หลายนๆคนมักคิดว่าเป็นตัวเดียวกัน แต่จริงๆแล้วแตกต่างกันโดยสิ้นเชิงทั้งความซับซ้อนของภาษา การใช้งาน ประสิทธิภาพ รวมถึงผู้พัฒนา โดย Java นั้นพัฒนาโดย Sun ซึ่งตอนนี้โดย Oracle ซื้อไปเรียบร้อยแล้ว ส่วน JavaScript นั้นพัฒนาโดยทีมงาน Netscape (Mozilla Foundation) ผู้พัฒนา Firefox Web browser ให้เราได้ใช้กันฟรีๆ แถมคุณภาพคับแก้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- JavaScript ทำให้สามารถใช้เขียนโปรแกรมแบบง่ายๆได้ โดยไม่ต้องพึ่งภาษาอื่น เช่น PHP เน้นว่าแบบง่ายๆ โดยส่วนใหญ่จะเป็นรูปแบบของการแสดงผลมากกว่า
- JavaScript มีคำสั่งที่ตอบสนองกับผู้ใช้งาน เช่นเมื่อผู้ใช้คลิกที่ปุ่ม หรือ Checkbox ก็สามารถสั่งให้เปิดหน้าต่างใหม่ได้ ทำให้เว็บไซต์ของเรามีปฏิสัมพันธ์กับผู้ใช้งานมากขึ้น นี่คือข้อดีของ JavaScript เลยก็ว่าได้ที่ทำให้เว็บไซต์ต่างๆทั้งหลายเช่น Google Map ต่างหันมาใช้
- JavaScript สามารถเขียนหรือเปลี่ยนแปลง HTML Element ได้นั้นคือสามารถเปลี่ยนแปลงรูปแบบการแสดงผลของเว็บไซต์ได้นั่นเอง สังเกตจากเมนูต่างๆใน Hellomyweb.com สามารถเลื่อนขึ้นลงได้หรือหน้าต่างแสดงเนื้อหาสามารถซ่อนหรือแสดงเนื้อหาได้แบบง่ายๆนั่นเอง
- JavaScript สามารถใช้ตรวจสอบข้อมูลได้ สังเกตว่าเมื่อเรากรอกข้อมูลบางเว็บไซต์ เช่น Email เมื่อเรากรอกข้อมูลผิดจะมีหน้าต่างฟ้องขึ้นมาว่าเรากรอกผิด หรือลิมกรอกอะไรบางอย่าง ส่วนใหญ่เกือบทั้งหมดใช้ JavaScript ตรวจสอบ
- JavaScript สามารถใช้ในการตรวจสอบผู้ใช้ได้เช่น ตรวจสอบว่าผู้ใช้ใช้ Web browser อะไร
- JavaScript สร้าง Cookies (เก็บข้อมูลของผู้ใช้ในคอมพิวเตอร์ของผู้ใช้เอง) ได้

2.9 ทฤษฎีเกี่ยวกับ NodeJS

node.js คือ environment เพื่อให้เราเขียน software มารันตามที่เราต้องการ ภายใต้ environment ของ node นี้ โดยภาษาที่เราจะใช้เขียน มันคือภาษา JavaScript เหมือนที่เราใช้เขียนบนหน้าเว็บเลย โดยจะมีตัว compiler คือ google JavaScript engine V8 ก็คือตัวประมวลผลภาษา JavaScript ที่ทาง google พัฒนาขึ้นมาตัวเอง (เป็น open source) สรุปคือ เราเขียนโปรแกรมด้วยภาษา JavaScript โดยความสามารถมันทำได้หลายอย่างมาก ไม่ว่าจะเป็นประมวลผลงานต่างๆ หรือติดต่อผ่าน socket หรือไปจนถึงสั่ง command line บนเครื่องเลย ทำได้หมด

node.js หยิบ V8 มาทำเป็นตัว compiler เพราะว่า มันประมวลผลได้รวดเร็วมาก ผลที่ได้คือมันทำตัวคล้ายๆกับจะเป็น parallel execution ด้วย คือ ทำงานหลายอย่างพร้อมกันในเวลาเดียวกัน แต่ความจริงคือไม่ใช่ เพราะว่ามันเป็น single thread แต่มันปรับกระบวนการโดยเอาแต่ละงานไปเข้าคิวเวลาเอาไว้ก่อน (ยังไม่ประมวลผล) แล้วค่อยประมวลผลงานตามคิวเวลา ไม่ได้เรียงตามขั้นตอนที่เสร็จหรือไม่เสร็จ ซึ่งกระบวนการเข้าคิวเวลามันใช้เวลาน้อยมากๆ เริ่มต้นทำงานได้พร้อมกันส่วนใหญ่เค้าจะนิยมใช้ node.js ในงานที่ทำเป็นเบื้องหลัง ก็คือ งานที่ประมวลผลส่ง server ซึ่งเป็นงานที่อาจจะต้อง interface กับผู้ใช้ หรือไม่ต้อง interface เลยก็ได้ ตัวอย่างงานที่ต้อง interface กับผู้ใช้ก็คือ การทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวเองเป็น http server ในการดึงหน้าเว็บมาแสดงผลให้กับ user หรือว่า การเปิด socket เพื่อรับส่งข้อมูลกันระหว่าง server กับคนใช้ ที่อาจจะเอาไปทำเป็นห้อง chat , ทำเกม, ทำระบบที่ป้อนข้อมูลเพื่อคำนวณเอาผลลัพธ์ เป็นต้น จุดนี้มันไม่ได้เอาไว้ทำหน้าที่อะไรครับ แต่มันเอาไว้ประมวลผลล้วนๆ หน้าตาที่ปล่อยให้มันเป็นหน้าที่ของ html + css + javascript ที่เขียนหน้าเว็บตามปกติ หรือ ตัวอย่างงานที่ไม่ต้อง interface กับคนใช้ก็เช่นเอาไปทำ spider crawler เว็บ คือให้มันเปิดเว็บอ่านแล้วเก็บข้อมูลไปเรื่อยๆ หรือว่า โปรแกรมที่ รอรับค่าจาก streaming ที่ต่างๆ เพื่อเอามาบันทึกเอาไว้ เหล่านี้ไม่จำเป็นต้อง interface กับคนใช้ ต่างก็ใช้ node.js ทำงานด้วยกันทั้งนั้น

node.js มี ส่วนเสริม ที่ชื่อว่า node package management หรือมองภาพง่ายๆว่า มันเป็นเหมือน google play ใน android หรือ app store ใน iOS นั่นเอง ที่เราสามารถเอา package ที่คนอื่นเขียนเอาไว้แล้วแจกฟรี มาต่อยอดเพื่อใช้ในงานของเราได้ โดยตัวอย่างที่ได้รับความนิยมมากๆ ก็จะเป็น underscore, async, request, express เป็นต้น อย่างเช่น ตัว underscore หากเราเอามาใช้ เราก็จะสามารถใช้งาน function เช่น `_.each([1, 2, 3], alert);` ได้เลย หรือ `_.filter(list, iterator, [context])` ได้เลย หรือ express ก็เหมือนเป็น web framework ตัวหนึ่งเลยติดตั้งตัวนี้แล้วตั้งค่านิดหน่อย ก็รันเว็บได้โดยไม่ต้องมี Apache เลย ซึ่ง package เหล่านี้ช่วยให้เราไม่ต้องมาเขียน function พื้นฐานเหล่านี้ด้วยตัวเราเอง ปัจจุบัน มีให้เราได้เลือกใช้เยอะมาก (ณ วันที่เขียน มี 37,784 ตัวแล้ว) จริงๆ เราก็ใช้เท่าที่งานเราจำเป็นนั้นแหละครับ (เท่าที่ผมเขียนมาก็ใช้ 1-4 ตัว ต่องานประมาณนี้) การติดตั้ง ก็แสนง่าย แค่สั่ง `npm install underscore` เท่านั้นเอง ก็จะได้ package underscore มาใช้

การจะใช้งาน node.js

node.js เวลาเราจะใช้งานมัน จะเหมือนภาษา script ที่ต้องใช้ command เพื่อสั่งให้มันทำงาน เพราะมันไม่มีตัวกลางที่คอยสร้าง request เหมือน web browser ที่ไปทำหน้าที่เรียกหาเว็บต่างๆ ดังนั้น คนที่จะใช้งาน node.js จำเป็นที่จะต้องรู้เรื่องของ linux command เพื่อให้พอเข้าใจ และเรียกใช้งานได้ โดยประมาณว่า จะใช้ command ของ linux เช่น `cd, ls, ll, &, nohup command &, ps aux | grep node, kill, nano (vi)` นอกเหนือจากนี้ไม่ได้ใช้เลย หรือถ้าใช้ก็น้อยมากๆ เวลาใช้งานเราก็สร้างไฟล์ `test.js` แล้วพิมพ์ว่า `console.log("yes you can do it");` แล้ว save จากนั้นไปที่ command แล้วสั่งรันด้วย `node test.js` เท่านั้น เราจะได้ข้อความดังกล่าวปรากฏที่หน้าจอแล้ว

การ debug node.js

node.js built in การ debug มาให้ด้วย วิธีใช้ก็แสนง่ายอีก คือ โดยปกติเวลาเราจะรันงานของเรา เราจะสั่ง `node file.js` เราก็แค่สั่งแทนด้วย `node debug file.js` เท่านั้นเอง หรืออีกวิธีเพื่อให้เรา

มองเห็นสิ่งที่ node มันทำก็คือ การใช้ `console.log('message');` เพื่อ print message ออกมาให้เราเห็นผลลัพธ์ ระหว่างที่เรารันไฟล์

node.js ทำงานแบบ event driven

การทำงานของ node จะเรียกมันว่า การขับเคลื่อนด้วย event ต่างๆที่เกิดขึ้น ทำให้เรากระโดดจาก event หนึ่งเสร็จแล้วไปอีก event หนึ่งได้ด้วยการส่งงานมันต่อเนื่องกันไปเรื่อยๆ หรือว่าการสั่งให้หลายๆ event เริ่มทำงานในเวลาใกล้ๆเคียงกันเลยก็ได้เช่นกัน ประโยชน์อีกอย่างที่ได้จาก event driven ก็คือ การสั่งให้มันรอรับ event นั้นไปตลอดการณ โดยไม่เปลือง resource เช่น การเชื่อมต่อไปยัง streaming channel สักที่หนึ่ง อาจจะเป็น text หรือข้อมูลบางอย่างเช่นปริมาณน้ำฝน ที่ง้อเอาไว้ หากต้นทางของ streaming ยังไม่มีข้อมูลส่งมา มันก็จะไม่เกิด event ใดๆ node.js ก็จะอยู่นิ่งๆ แต่หากต้นทาง streaming มาแล้ว node.js ก็จะทำงานเพื่อตอบสนองต่อ event เกิดขึ้นทันที

ด้วยประโยชน์ของ event driven นี้ทำให้เราเอามาต่อยอดได้อีกหลายอย่างเช่น การ subscribe pubsub (เช่น pubsub ใน redis) หากเกิด event publish เมื่อไร node.js ที่ subscribe รอเอาไว้อยู่นานแล้ว ก็จะถูกระตุ้นแล้วทำงานตามที่เราเขียนเอาไว้โดยทันที ซึ่งแบบนี้ จะทำให้ลดการสูญเสีย header ในการเริ่มต้นประมวลตั้งแต่จุดแรกไปได้มาก รวมทั้ง ไม่ต้องไปหน่วงเวลาเพื่อคอยเช็ค event เหมือนเวลาเราเขียน ajax เลยแม้แต่น้อย เรียกได้ว่า อยู่เฉยๆเถอะ ถ้ามีข้อมูลมาเดี๋ยวไปกระตุ้นให้ทำงานเอง

2.10 ทฤษฎีเรื่อง CMD

cmd คือ คำสั่งที่ใช้เปิด Windows Command interpreter สามารถเรียกใช้โปรแกรมต่าง ๆ ผ่าน text mode ปกติเวลาเราใช้งานวินโดวส์มักจะใช้งานอยู่ในโหมดของกราฟิกโหมด (GUI) ซึ่งระบบ GUI นี้ทำให้เราสามารถทำงานทุกอย่างไม่ว่าจะเป็น การสั่งรันโปรแกรม ก็อปปีไฟล์หรือลบไฟล์ ผ่านทางวินโดวส์โหมดได้ แต่ระบบปฏิบัติการก็มีโหมดที่เรียกว่า Command Prompt หรือบางทีก็เรียกว่า Dos Prompt ซึ่งเป็นโหมดการทำงานในสภาพแวดล้อมที่เป็นตัวอักษร (Text Mode) นั้นหมายความว่าเมื่อต้องการใช้คำสั่งใด ท่านต้องพิมพ์คำสั่งต่างๆเอาเอง ซึ่งจะพบเห็นได้ในโปรแกรมเก่าๆหลายตัว หรือแม้แต่โปรแกรมใหม่ๆบางตัว จะเรียกใช้งานหรือปรับแต่งค่าได้เมื่ออยู่ในโหมด Command Prompt เท่านั้น

2.11 ทฤษฎีเรื่อง Sublime Text 2

Sublime Text เหมาะกับทุกคนที่เขียนโปรแกรม ทั้งผู้เขียนระบบ (Developer) และผู้ออกแบบ (Designer) ที่ชอบแต่งอะไรๆ ด้วยตัวเองและคนที่สนใจเฉพาะด้านเพราะมันสามารถพัฒนาต่อยอด (Customize) ได้เยอะมาก สามารถเปลี่ยนรูปแบบ (Theme) ได้ สามารถลง package ต่างๆ เพิ่มเข้าไปได้ และ Sublime Text2 ไม่เน้นการปรับแต่งแบบหน้าจอการใช้งาน (GUI) ทำให้ เวลาที่ยาก กำหนด (Config) แต่ละครั้ง ต้องไปแก้ที่ตัวโค้ดกำหนด (Config) เาเอง ซึ่งบางคนอาจจะไม่ชอบ แต่ชัดเจน โปรแกรมนี้เปิดให้ดาวโหลดไปใช้ฟรีๆ แบบครบทุกลักษณะ (Full feature) แต่จะมีหน้าต่างแสดงขึ้นมา (pop-up) ให้เราซื้อลิขสิทธิ์ (License) อยู่เป็นระยะๆ ถ้าใช้ประจำควรซื้อลิขสิทธิ์ (LC) เป็นการสนับสนุน โปรแกรมดีๆ โดยราคาลิขสิทธิ์ (License) อยู่ที่ \$59 ที่สำคัญโปรแกรมนี้มีในทุกลักษณะ (Platform) ทั้ง Window และ Mac OSX เร็วมาก ทั้งตอนเปิดโปรแกรม เรียกไฟล์หรือฟังก์ชันต่างๆ

- คุณสมบัติ (Multiple Cursors) สามารถ แกะไขหลายๆ ที่ในทีเดียว ช่วยประหยัดเวลาได้มาก
- แต่งรูปแบบ (Theme) ได้เอง และมีแบบที่คนทำไว้ให้เยอะมาก
- ลักษณะเด่นของหน้าจอ (Split Screen) สามารถแบ่งหน้าจอการทำงานได้ แบบเป็นคอลัมน์ เป็นแถว หรือเป็นการออกแบบเลย์เอาต์ (Grid)
- การค้นหาข้อมูล (Command Palette) ทำหน้าที่คล้ายๆ Spotlight ใน Mac ที่หาอะไรไม่เจอ ก็พิมพ์เข้าไป เดียวจะหาคำสั่งนั้นมาให้เอง สะดวกมากๆ
- พื้นที่ขนาดเล็ก (Minimap) สำหรับดูว่าแก้ไขโค้ดส่วนไหนของไฟล์อยู่
- Sublime Package Control เป็นการติดตั้ง (plugin) ที่ช่วยให้เราควบคุมรายการชุดคัม (package) ต่างๆ ที่เราจะลงเพิ่มใน Sublime text ได้

บทที่ 3

วิธีการดำเนินงานวิจัย

วิธีการดำเนินงานของการออกแบบและการพัฒนาการทดสอบระบบแบบการนำเครื่องมือมาช่วย (Automation Test) สำหรับการทำงาน เพื่อลดปัญหาการทดสอบระบบแบบวนซ้ำ (Re-Test) ลดขั้นตอนความยุ่งยากแบบการทดสอบด้วยตนเอง (Manual Test) บนเว็บไซต์ที่สามารถเก็บข้อมูล ภายในองค์กร (SharePoint) ได้ทำการศึกษาค้นคว้า แนวคิด และทฤษฎีต่างๆที่เกี่ยวข้องกับการพัฒนาระบบ เพื่อนำไปประยุกต์เป็นแนวทางในการพัฒนา โดยมีประเด็นตามลำดับหัวข้อดังนี้

3.1 ขั้นตอนการศึกษาและรวบรวมข้อมูลของระบบ

1) ศึกษาปัญหาและความต้องการของระบบ

ศึกษาปัญหาและรวบรวมข้อมูลของการพัฒนาระบบแบบการนำเครื่องมือมาช่วย (Automation Test) สำหรับการทำงาน เพื่อลดปัญหาการทดสอบระบบแบบวนซ้ำ (Re-Test) และลดขั้นตอนความยุ่งยากกว่าการทดสอบด้วยตนเองโดยไม่ใช้เครื่องมือ (Manual Test) บนเว็บไซต์ที่สามารถเก็บข้อมูล ภายในองค์กร (SharePoint) รวมถึงศึกษาความต้องการของระบบเพื่อนำมาวิเคราะห์และออกแบบระบบในขั้นต่อไป

2) ศึกษาขั้นตอนการพัฒนาของระบบ

ศึกษาขั้นตอนการพัฒนาระบบ เริ่มจากการศึกษาและรวบรวมข้อมูลต่างๆที่ใช้ในการทำระบบแบบการนำเครื่องมือมาช่วย (Automation Test) จากนั้นทำการวิเคราะห์ระบบงานเดิมกว่าการทดสอบด้วยตนเองโดยไม่ใช้เครื่องมือ (Manual Test) t เพื่อนำมาปรับปรุงในการทำเป็นระบบ แล้วทำการออกแบบระบบ และทำการทดสอบระบบ

3) ศึกษาข้อมูลและเครื่องมือที่ใช้ในการพัฒนาระบบ

ศึกษาการพัฒนาระบบแบบการนำเครื่องมือมาช่วย (Automation Test) เพื่อนำไปใช้งานบนเว็บไซต์ที่สามารถเก็บข้อมูล ภายในองค์กร (SharePoint) ด้วยโปรแกรม PhantomJS และ ใช้โปรแกรม Visual Studio โดยใช้ภาษา JavaScript เพื่อพัฒนาการทำงานแบบการทดสอบด้วยตนเอง (Manual Test) และลดปัญหาการทดสอบระบบแบบวนซ้ำ (Re-Test) และศึกษาการใช้งานที่มีประสิทธิภาพโดยการนำ เครื่องมือต่างๆ มาช่วยเพื่อให้ระบบมีประสิทธิภาพและแม่นยำมากขึ้น

3.2 ขั้นตอนการวิเคราะห์ระบบ

จากการศึกษาและวิเคราะห์ปัญหาของระบบการทำงานแบบการทดสอบด้วยตนเอง (Manual Test) แล้ว แต่เดิมนั้นส่งผลกระทบต่อกระบวนการทำงานหลายส่วน ดังนี้

1. ในปัจจุบันการทำงานแบบการทดสอบด้วยตนเอง (Manual Test) เป็นกระบวนการทำงานแบบใช้มือในการทดสอบระบบทั้งหมดแต่ละครั้งจนกว่าจะครบ (Test Case) ซึ่งจะมีปัญหาในการทำงานที่ล่าช้า
2. ต้องอาศัยระยะเวลาในการทดสอบระบบแต่ละครั้งเป็นเวลานาน ในกรณีที่มีการเปลี่ยนแปลง ความต้องการ (requirement) การทำให้สูญเสียเวลาในการไปทดสอบระบบซ้ำๆ (Re-Test) หรือ การบันทึกข้อมูล การเปลี่ยนแปลง ในแต่ละครั้งลงเครื่องคอมพิวเตอร์
3. ข้อจำกัดในการทำงานที่การทดสอบระบบ บางอย่างไม่สามารถทำโดยวิธีการใช้แบบ แบบการทดสอบด้วยตนเอง (Manual Test) ในกรณีที่ต้องการทดสอบเว็บแอปพลิเคชันที่รองรับผู้ใช้เป็นจำนวนมากและผู้ใช้มีการเข้าถึงแอปพลิเคชันในเวลาใกล้เคียงกัน เป็นต้น
4. การทดสอบด้วยมือเป็นการทำงานซ้ำๆ (Re-Test) แก่นักทดสอบส่งผลให้ความถูกต้องของการทดสอบลดลง อาจได้ผลลัพธ์ไม่เท่ากันทุกครั้งที่มีการรันการทดสอบ

จากการศึกษาระบบงานแบบเดิม และ วิเคราะห์ความเป็นไปได้ในการพัฒนาความต้องการของระบบงานใหม่ เพื่อให้กระบวนการทำงานมีประสิทธิภาพและมีความแม่นยำในการทดสอบระบบทุกครั้ง มีข้อมูลที่ต้องการ สามารถตอบสนองความต้องการของนักทดสอบระบบขององค์กรในการจัดการระบบ ข้อมูลให้มีความถูกต้อง และมีการเกิดปัญหาน้อยที่สุด สามารถลดเวลาและค่าใช้จ่ายในการทดสอบระบบแต่ละครั้งได้ เช่น ลดระยะเวลาในการทดสอบระบบ และเพิ่มประสิทธิภาพความถูกต้องในการรันทดสอบระบบแบบซ้ำๆ เมื่อมีการเปลี่ยนแปลงความต้องการ (requirement) และให้สามารถทำงานบนแพลตฟอร์ม ของเว็บไซต์ (SharePoint) ได้ นอกจากนี้ยังมีการบันทึกวันที่และเวลาเพื่อตรวจสอบการทำงานว่าใช้เวลาเท่าไรในการทดสอบ ช่วยให้ทดสอบได้มีประสิทธิภาพมากขึ้น

3.3 ขั้นตอนการออกแบบระบบ

3.3.1 การออกแบบการทดสอบระบบแบบการนำเครื่องมือมาช่วย (Automation Test) การเริ่มต้นสร้างระบบนั้นหลังจากรับความต้องการของลูกค้า (Requirement) จะนำมาวิเคราะห์ออกแบบและประเมิน ผู้เขียนโปรแกรม (Programmer) จะเขียนโปรแกรมเพื่อสร้างระบบ เมื่อทำระบบเสร็จเรียบร้อย ผู้ทดสอบระบบ (Tester) จะทดสอบระบบแบบการนำเครื่องมือมาช่วย (Automation Test) โดยการสร้างเอกสารการทดสอบทั้งหมด (Test Case) และต้องทดสอบการทดสอบด้วยตนเอง (Manual Test) และเขียนโปรแกรม (Script) ตามภาษาที่ใช้งานเพื่อนำไปรันทดสอบระบบ มีการทำงานเพื่อเปรียบเทียบการใช้งานที่เหมาะสมบนแพลตฟอร์ม เว็บไซต์ (SharePoint) จะทดสอบระบบตั้งแต่ขั้นตอน

แรกจนจบกระบวนการ ถ้าระบบเกิดข้อผิดพลาด (error) ส่วนไหนจะมีข้อมูลแจ้งเตือนข้อมูลที่ผิดพลาด (error) ระหว่างการทำงาน เพื่อส่งกลับไปให้ ผู้เขียนโปรแกรม (Programmer) แก้ไข และส่งกลับมาเพื่อ ทดสอบระบบอีกครั้ง ถ้าระบบรันผ่านทั้งหมดจะมีการผลออกทางหน้าจอเพื่อให้บันทึกข้อมูล โดยลำดับ การทำงานของระบบจะแสดงดังรูปที่ 3.1

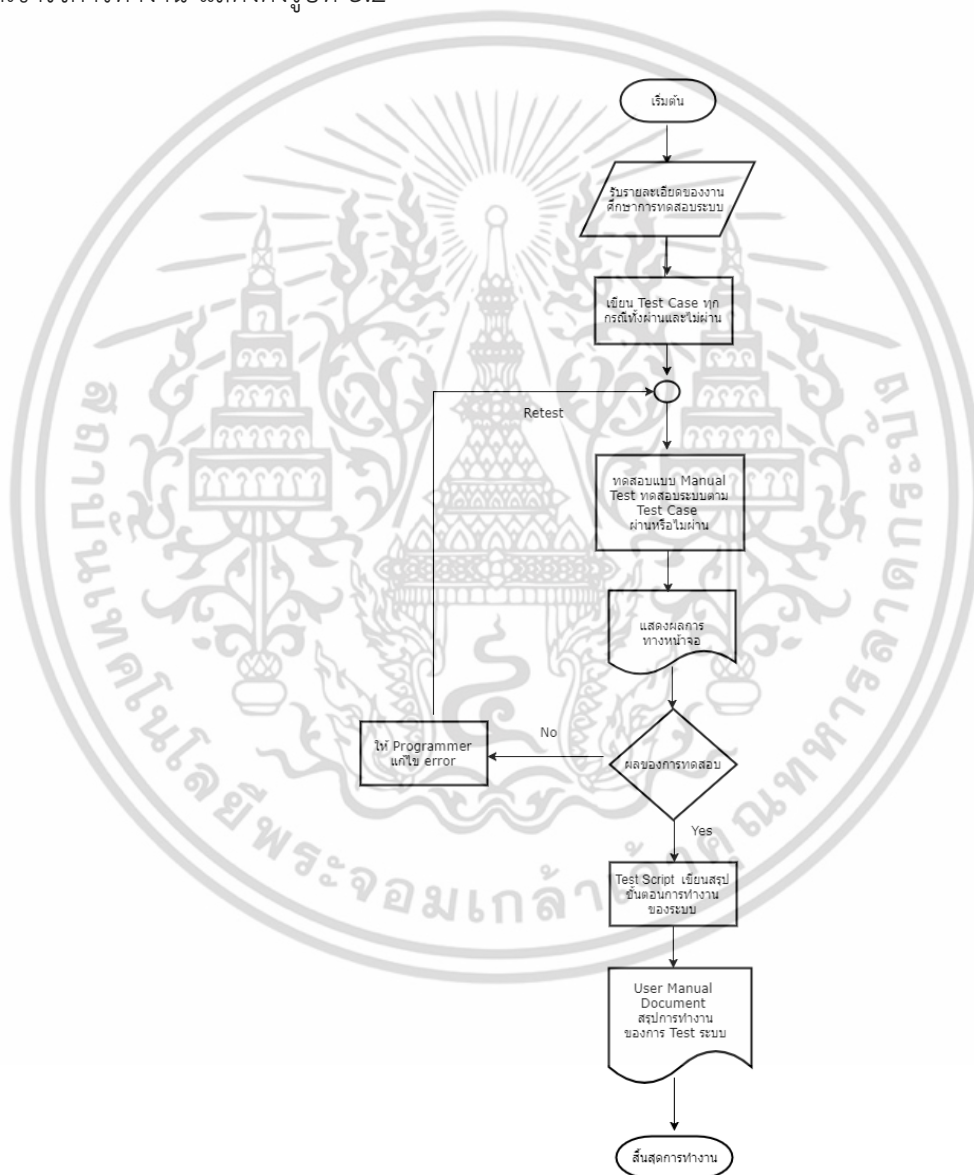


รูปที่ 3.1 Flowchart Diagram แสดงขั้นตอนการทดสอบระบบแบบ Automation Test

จากรูปที่ 3.1 สามารถนำมาแสดงการทำงานของแต่ละระบบย่อยได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

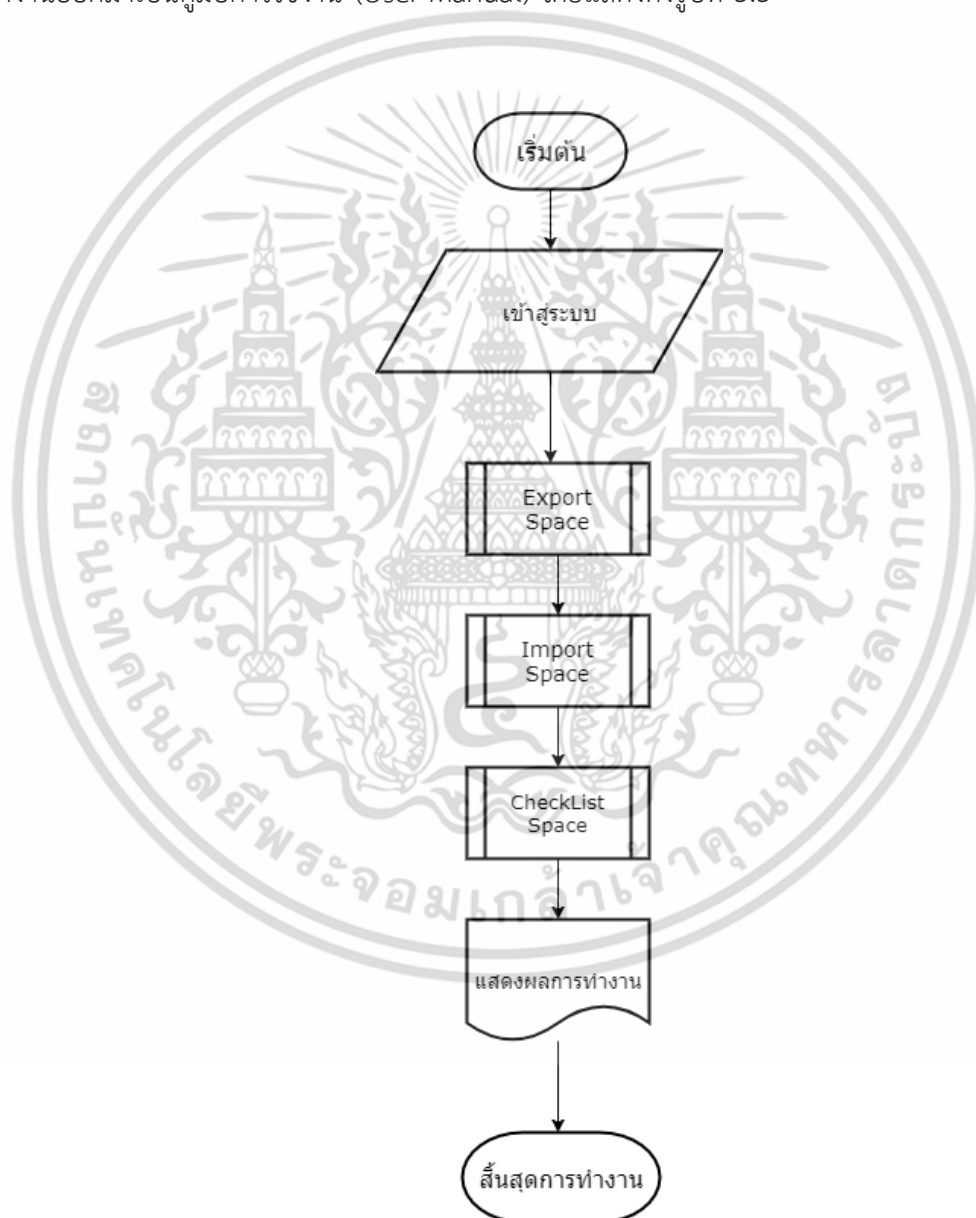
เมื่อระบบเสร็จแล้วจะมีการนำมาทดสอบระบบ โดยขั้นตอนการทดสอบโดยการสร้างเอกสารวางแผนการทำงาน (Test Case) ขึ้นมาก่อนเพื่อนำมาเขียนโปรแกรมเพื่อเรียงลำดับขั้นตอนในการรันทดสอบระบบ ถ้าระหว่างการทดสอบมีข้อผิดพลาด (Error) จะมีการแสดงการแจ้งเตือนเพื่อให้ทราบข้อผิดพลาดและนำส่งกลับไปให้ผู้เขียนโปรแกรม (Programmer) แก้ไขแล้วกลับมาทดสอบระบบอีกครั้ง (Retest) เพื่อยืนยันการแก้ไขว่าผ่านแล้วเรียบร้อยแล้ว จึงนำมาเขียนขั้นตอนการทำงานระบบทั้งหมด (Test Script) และบันทึกผลที่ออกมาเพื่อทำคู่มือการใช้งาน (User Manual) ให้ผู้ใช้งาน (User) ได้เข้าใจการทำงาน แสดงดังรูปที่ 3.2



รูปที่ 3.2 Flowchart Diagram แสดงขั้นตอนการทดสอบระบบแบบ Manual Test

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

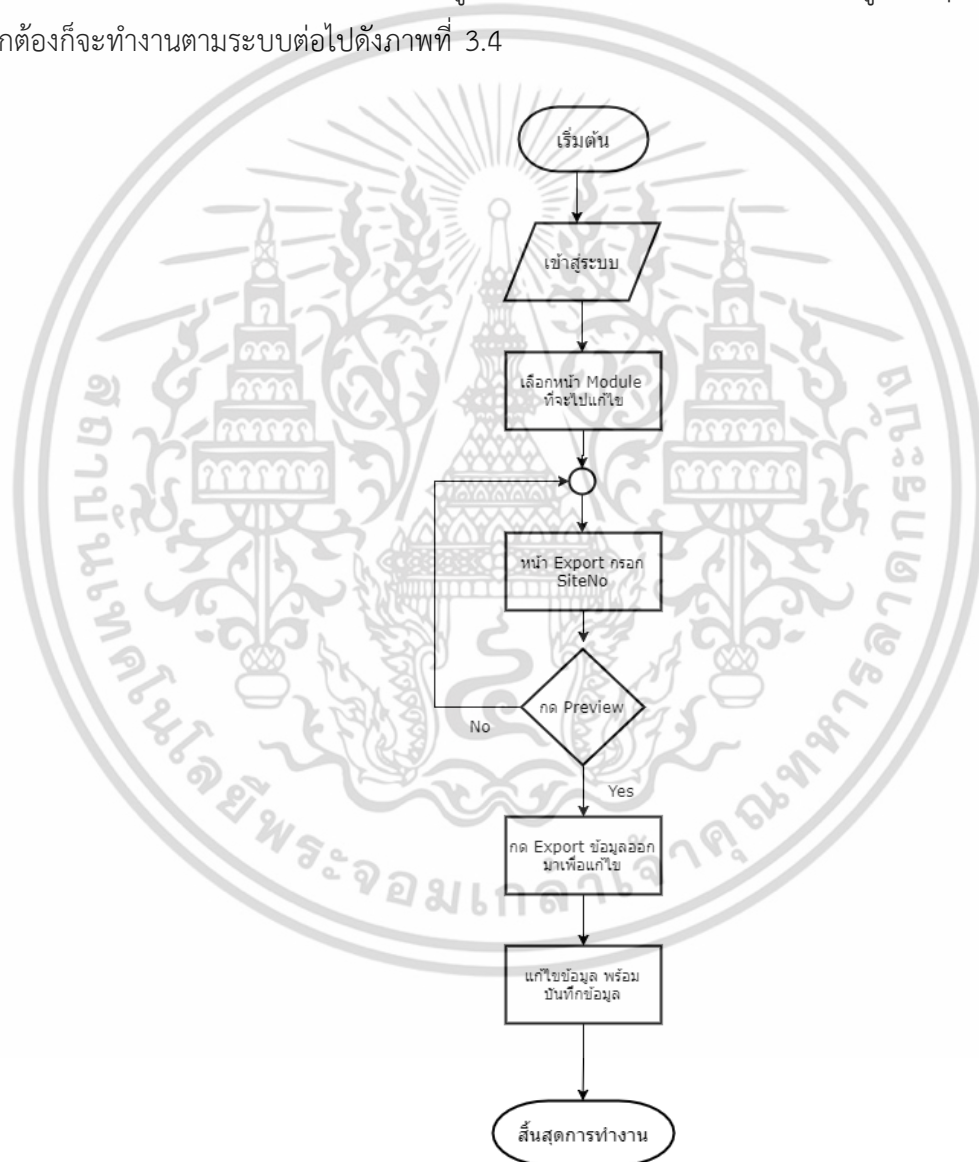
เมื่อทราบการทำงานของระบบทดสอบระบบด้วยตนเอง (Manual Test) และการทดสอบโดยใช้เครื่อง (Automation Test) แล้ว เราจะนำมาทดสอบกับระบบเพื่อสามารถใช้งานได้จริง โดยการเขียนโปรแกรม (Script) ด้วยภาษา JavaScript จะให้รันตามขั้นตอนของระบบตั้งแต่ต้นจบโดยมีการทำงานตามระบบ โดยเริ่มจากการเข้าสู่ระบบ เข้าหน้าเข้าสู่ข้อมูลระบบนำข้อมูลออกจากระบบ (Export) เพื่อดาวน์โหลดข้อมูล (Upload) ออกมาแก้ไขและนำข้อมูลเข้าสู่ระบบ (Import) ข้อมูล และมีการตรวจสอบ (Checklist) เพื่อตรวจสอบการทำงานว่าแก้ไข (Update) ได้ถูกต้องตามที่แก้ไข สุดท้ายจะแสดงผลการทำงานออกมาเป็นคู่มือการใช้งาน (User Manual) โดยแสดงดังรูปที่ 3.3



รูปที่ 3.3 Flowchart Diagram แสดงการทำงานของระบบที่นำมาทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

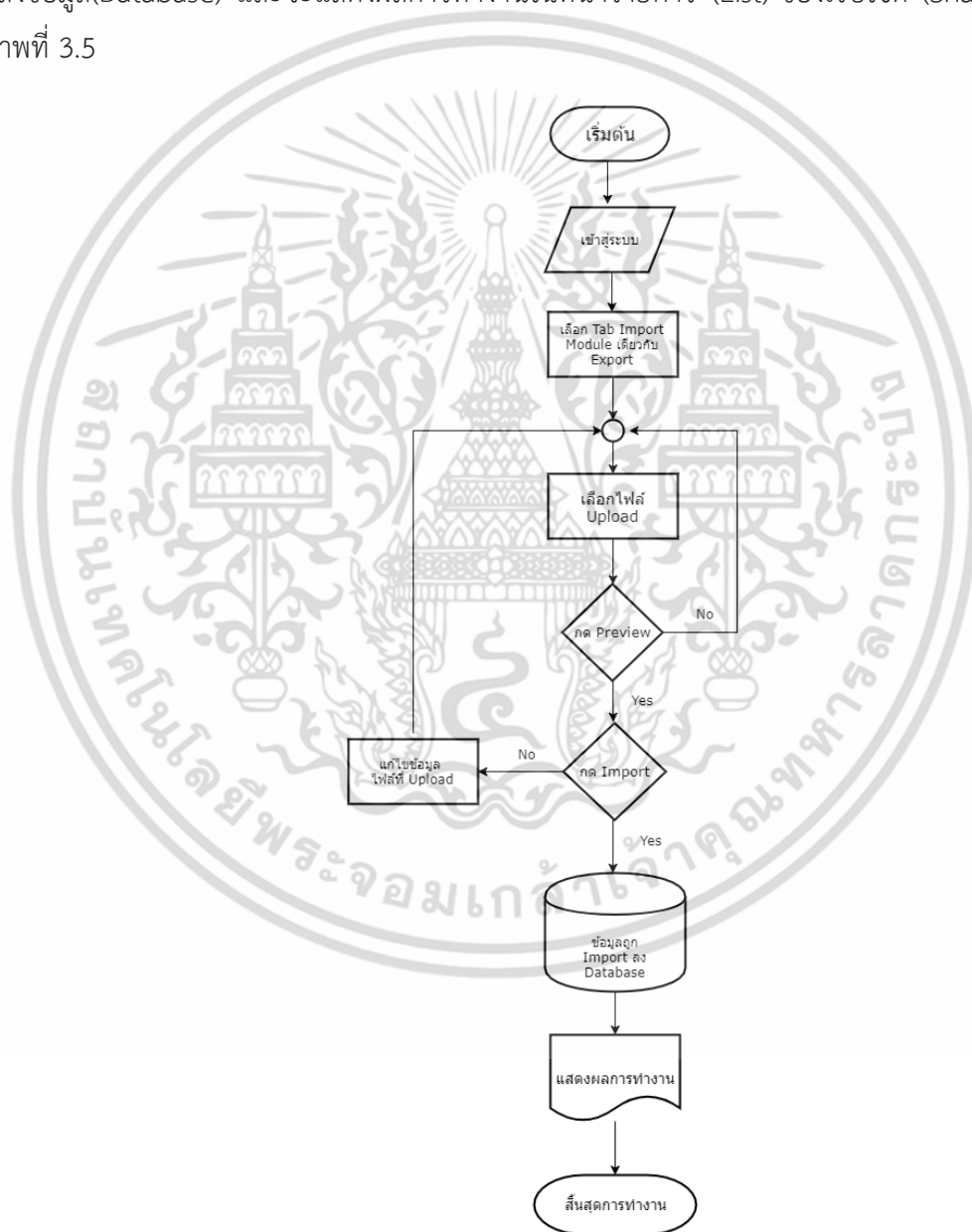
จากการแสดงการทำงานของระบบที่นำมาทดสอบ ขั้นตอนไปคือการแสดงการทำงานของระบบส่วนส่งออกของข้อมูล (Module Export Space) เพื่อนำมาแก้ไขและเข้าสู่ระบบ โดยขั้นตอนแรกจะมีการกรอกชื่อผู้ใช้งานและรหัสผ่านเพื่อการเข้าสู่ระบบ แล้วให้ไปเลือกเข้าสู่หน้าระบบย่อย (Module) ที่จะแก้ไข โดยหน้าส่งออกข้อมูล (Export) จะมีให้กรอกรหัสของข้อมูล (SiteNo) เมื่อกดปุ่มแสดงข้อมูล (Preview) จะแสดงข้อมูลที่ถูกต้องและข้อมูลที่ผิดพลาด ข้อมูลที่ถูกต้องจะสามารถนำข้อมูลออก (Export) มาเป็นไฟล์ เพื่อนำไปแก้ไขข้อมูลใหม่และบันทึกเพื่อนำกลับเข้าสู่ระบบ ส่วนข้อมูลที่แสดงผิดพลาด (Error) จะให้กรอกรหัสของข้อมูล (SiteNo) ใหม่ ที่หน้าส่งออกข้อมูล (Export) ถ้าข้อมูลถูกต้องก็จะทำงานตามระบบต่อไปดังภาพที่ 3.4



รูปที่ 3.4 Flowchart Diagram แสดงการทำงานของระบบ Module Export Space

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

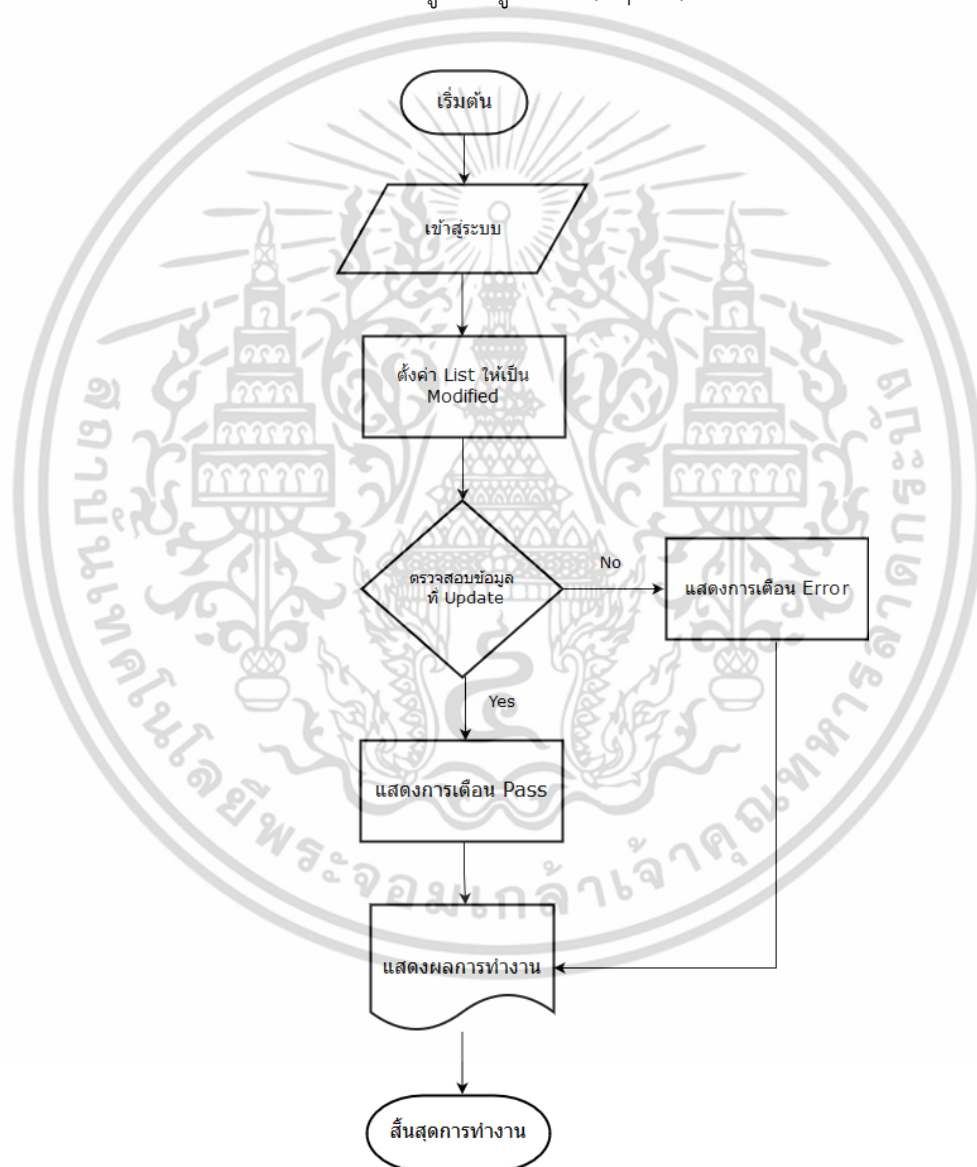
จากการแสดงการทำงานนำข้อมูลออกสู่ระบบเพื่อแก้ไข ขั้นต่อไปคือการแสดงการทำงาน นำไฟล์ที่แก้ไขเข้าสู่ระบบ (Module Import Space) โดยต่อจากการนำไฟล์ที่แก้ไขข้อมูลแล้วให้ไปเลือกเข้าสู่หน้าระบบย่อย (Module) ที่จะนำไฟล์เข้าสู่ระบบ (Import) และให้เลือกไฟล์ที่จะนำเข้าสู่ระบบ กดแสดงข้อมูล (Preview) ถ้าข้อมูลถูกต้องก็จะสามารถกดบันทึก (Import) เพื่อเข้าสู่ระบบ และเมื่อกดแสดงข้อมูล (Preview) หรือ เมื่อบันทึก (Import) ไม่ได้ ข้อมูลก็ต้องถูกส่งกลับไปแก้ไขใหม่และนำไฟล์เข้าสู่ระบบใหม่อีกครั้ง และถ้าข้อมูลถูกการบันทึก (Import) ข้อมูลจะไปถูกแทนที่แทนข้อมูลเดิมในคลังข้อมูล(Database) และจะแสดงผลการทำงานในหน้ารายการ (List) ของเว็บไซต์ (SharePoint) ดังภาพที่ 3.5



รูปที่ 3.5 Flowchart Diagram แสดงการทำงานของระบบ Module Import Space

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

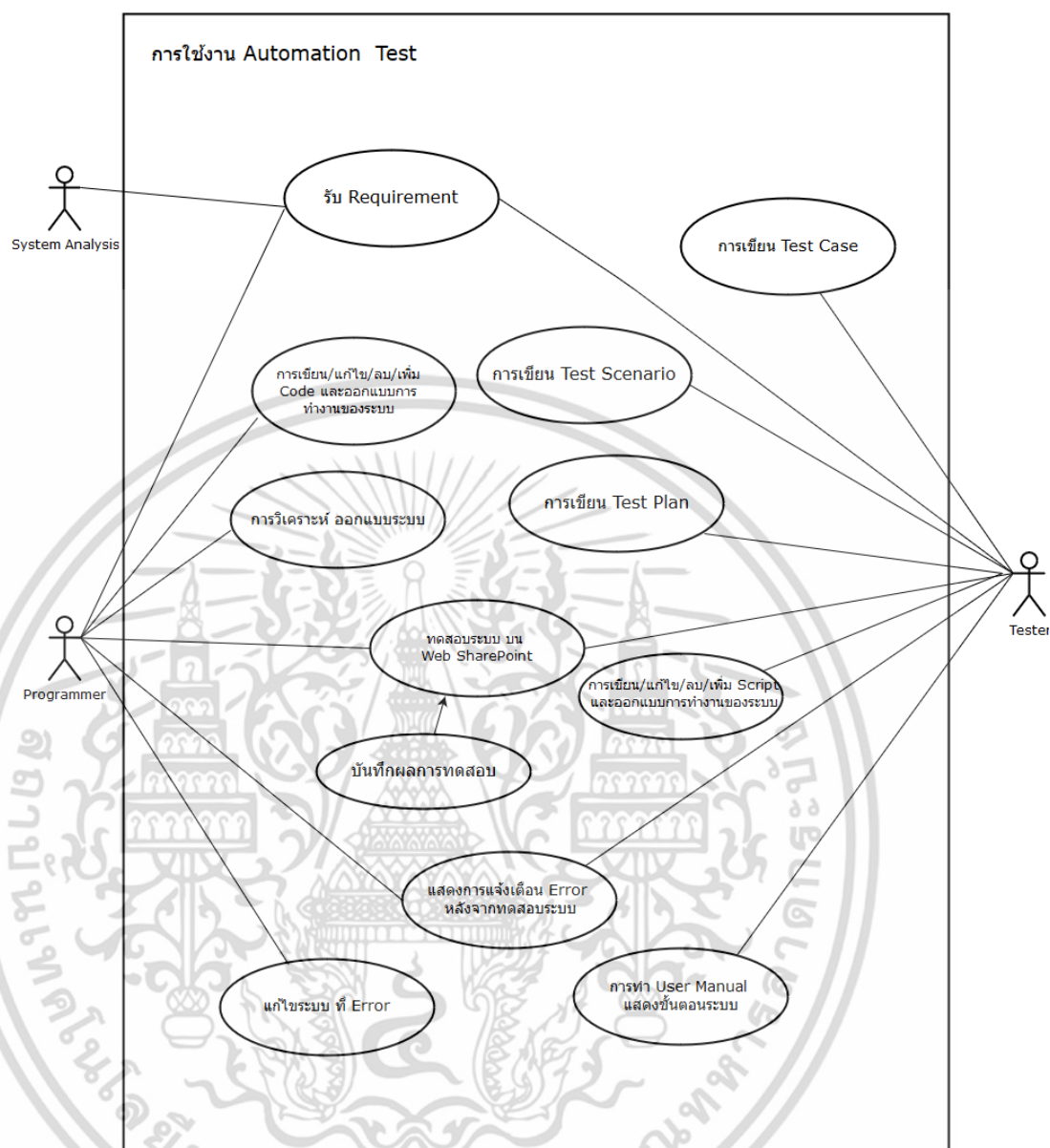
จากการที่ข้อมูลถูกบันทึก (Import) เข้าสู่ระบบ เราจะเขียนโปรแกรมเพื่อตรวจสอบ (Check) ว่าข้อมูลที่เรแก้ไขได้ถูกบันทึกในรายการ (List) ในคลังข้อมูล (Database) ของเว็บไซต์ (SharePoint) โดยเราจะตั้งค่าหน้ารายการ (List) ที่เราจะไปตรวจสอบของเว็บไซต์ (SharePoint) โดยตั้งค่าให้เลือก รายการที่ถูกแก้ไขล่าสุดอยู่ด้านบนสุด (Modified) และเขียนโปรแกรมให้ตรวจสอบว่าด้านบนสุดมีข้อมูลตรงกันกับที่แก้ไขและถูกข้อมูลใหม่แทนที่ข้อมูลเดิมเรียบร้อยแล้ว ถ้าข้อมูลถูกต้องจะถูกแสดงผลทางหน้าจอว่าผ่าน (Pass) ถ้าไม่ถูกต้องจะแสดงผลทางหน้าจอว่าผิด (Error) และถ้าจะแก้ไขต้องกลับไปตรวจสอบตั้งแต่ขั้นตอนแรกคือการนำข้อมูลเข้าสู่ระบบ (Export) ดังภาพที่ 3.6



รูปที่ 3.6 Flowchart Diagram แสดงการทำงานของระบบ Module Check List Space

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทำงานของ Flowchart ขึ้นต่อไปจะแสดงการทำงาน หน้าที่ของแต่ละฝ่าย (Use Case) ของระบบโดยเริ่มที่ระบบแบบใช้เครื่องมือทดสอบ (Automation Test) จะมีฝ่ายนักวิเคราะห์ออกแบบระบบ (System Analysis) ซึ่งจะมีหน้าที่รับข้อมูลตามความต้องการ (Requirement) ของลูกค้า เพื่อมาสร้างระบบตามที่ถูกคำต้องการ และส่งความต้องการของลูกค้าให้ผู้เขียนระบบ (Programmer) และผู้ทดสอบระบบ (Tester) โดยผู้เขียนระบบ (Programmer) จะมีการวิเคราะห์ออกแบบ วางแผนการสร้างระบบเพื่อนำมาเขียนโปรแกรมตามภาษาต่างๆที่ถูกคำต้องการมีการทดสอบระบบและบันทึกผลการทดสอบในขั้นต้นและแก้ไขระบบที่ผิดพลาด (Error) ตามที่ตนเองทดสอบและผู้ทดสอบระบบ (Tester) ทดสอบ ส่วนหน้าที่ของผู้ทดสอบระบบ (Tester) หลังจากรับความต้องการ (Requirement) ของลูกค้า จะมีการวางแผน ออกแบบ (Test Plan) , จำลองสถานการณ์ของระบบ (Test Scenario) และเขียนกรณีขั้นตอนการทดสอบระบบทั้งผิดและถูก (Test Case) และเมื่อผู้เขียนระบบ (Programmer) เขียนระบบเสร็จ จะส่งให้ผู้ทดสอบ (Tester) ทดสอบตามขั้นตอนการทดสอบระบบทั้งกรณีที่ดีและถูก (Test Case) ถ้ามีข้อผิดพลาด (Error) และส่งให้ผู้เขียนระบบ (Programmer) แก้ไขแล้วจะกลับมาทดสอบระบบอีกครั้ง (Re-test) และจะเขียนโปรแกรมเพื่อทดสอบระบบเพื่อทดสอบตั้งแต่ขั้นตอนแรกจนจบโดยถ้ามีการแก้ไขก็สามารถใช้โปรแกรมทดสอบได้โดยไม่ต้องทดสอบด้วยตนเองอีกรอบ ซึ่งจะช่วยลดระยะเวลาและแสดงผลการทำงานเป็นไฟล์ที่สามารถตรวจสอบข้อมูลที่ถูกและผิด เพื่อทำเป็นคู่มือการใช้งานให้ผู้เข้าใจได้ง่าย ดังภาพที่ 3.7

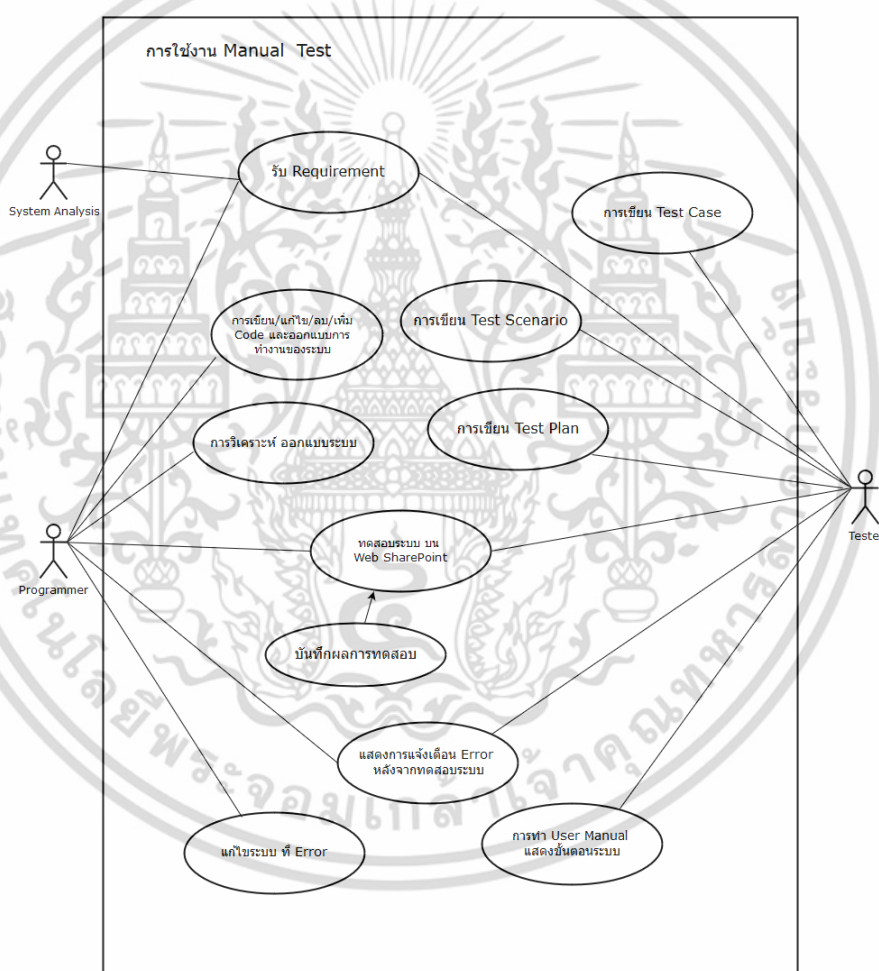


รูปที่ 3.7 Use Case แสดงการทำงานของระบบ Automation Test

จากการทำงานของหน้าที่ของแต่ละฝ่าย (Use Case) แสดงการทำงานของระบบ Automation Test ขึ้นต่อไปจะแสดงการทำงานของการทำงานของการทดสอบด้วยตนเอง (Manual Test) จะมีฝ่ายนักวิเคราะห์ ออกแบบระบบ (System Analysis) ซึ่งจะมีหน้าที่รับข้อมูลตามความต้องการ (Requirement) ของลูกค้า เพื่อมาสร้างระบบตามที่ลูกค้าต้องการ และส่งความต้องการของลูกค้าให้ผู้เขียนระบบ (Programmer) และผู้ทดสอบระบบ (Tester) โดยผู้เขียนระบบ (Programmer) จะมีการวิเคราะห์ ออกแบบ วางแผนการสร้างระบบเพื่อนำมาเขียนโปรแกรมตามภาษาต่างๆที่ลูกค้าต้องการมีการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบและบันทึกผลการทดสอบในขั้นต้นและแก้ไขระบบที่ผิดพลาด (Error) ตามที่ตนเองทดสอบและผู้ทดสอบระบบ (Tester) ทดสอบ ส่วนหน้าที่ของผู้ทดสอบระบบ (Tester) หลังจากรับความต้องการ (Requirement) ของลูกค้า จะมีการวางแผน ออกแบบ (Test Plan) , จำลองสถานการณ์ของระบบ (Test Scenario) และเขียนกรณีขั้นตอนการทดสอบระบบทั้งผิดและถูก (Test Case) และเมื่อผู้เขียนระบบ (Programmer) เขียนระบบเสร็จ จะส่งให้ผู้ทดสอบ (Tester) ทดสอบตามขั้นตอนการทดสอบระบบทั้งกรณีที่ดีและถูก (Test Case) ถ้ามีข้อผิดพลาด (Error) และส่งให้ผู้เขียนระบบ (Programmer) แก้ไขแล้วจะกลับมาทดสอบระบบอีกครั้ง (Re-test) และโดยการทดสอบด้วยตนเอง (Manual Test) การทดสอบระบบอีกครั้งต้องทดสอบใหม่ทั้งหมดทำให้ใช้เวลานานและประสิทธิภาพอาจลดลง



รูปที่ 3.8 Use Case แสดงการทำงานของระบบ Manual Test

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| Scenario ID | Test Case ID | Test Case Condition |
|-------------|--|---|
| UTS01 | Export File Excel FTTH (แก้ไข Information ชื่อหมู่บ้าน) | |
| | UTC01-005 | Upload File Excel FTTH file นามสกุลที่กำหนด |
| | UTC01-006 | Upload File Excel FTTH file นอกเหนือนามสกุลที่กำหนด |
| | Data Excel FTTH | |
| | UTC01-007 | คอลัมน์ "New Site Name(EN)" ใน Excel File ไม่กรอกข้อมูล |
| | UTC01-008 | คอลัมน์ "New Site Name(EN)" โดย คอลัมน์ "New Site Name(EN)" ใน Excel File กรอกข้อมูลเป็นภาษาอังกฤษและต้องกรอก "New Site Name(TH)" ด้วย |
| | UTC01-009 | คอลัมน์ "New Site Name(EN)" โดย คอลัมน์ "New Site Name(EN)" ใน Excel File กรอก ข้อมูลเป็นภาษาอังกฤษ และ ไม่กรอก "New Site Name(TH)" |
| | UTC01-010 | คอลัมน์ "New Site Name(EN)" โดย คอลัมน์ "New Site Name(EN)" ใน Excel File กรอกข้อมูลเป็นนอกเหนือจากภาษาอังกฤษ และ ต้องกรอก "New Site Name(TH)" ด้วย |
| | UTC01-011 | คอลัมน์ "New Site Name(EN)" โดย คอลัมน์ "New Site Name(EN)" ใน Excel File กรอก ข้อมูลเป็นนอกเหนือจากภาษาอังกฤษ และ ไม่กรอก "New Site Name(TH)" |
| | UTC01-012 | คอลัมน์ "New Site Name(TH)" ใน Excel File ไม่กรอกข้อมูล |
| | UTC01-013 | คอลัมน์ "New Site Name(TH)" โดย คอลัมน์ "New Site Name(TH)" ใน Excel File กรอกข้อมูลเป็นภาษาอังกฤษ และ ต้องกรอก "New Site Name(EN)" ด้วย |
| | UTC01-014 | คอลัมน์ "New Site Name(TH)" โดย คอลัมน์ "New Site Name(TH)" ใน Excel File กรอกข้อมูลเป็นภาษาอังกฤษ และ ไม่กรอก "New Site Name(EN)" |
| | UTC01-015 | คอลัมน์ "New Site Name(TH)" โดย คอลัมน์ "New Site Name(TH)" ใน Excel File กรอกข้อมูลเป็นภาษาอังกฤษ และ ต้องกรอก "New Site Name(EN)" ด้วย |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| Scenario ID | Test Case ID | Test Case Condition |
|-------------|--------------------------------|---|
| UTS01 | UTC01-016 | คอลัมน์ "New Site Name(TH)" โดย คอลัมน์ "New Site Name(TH)" ใน Excel File กรอกข้อมูลเป็นภาษาอังกฤษ และไม่กรอก "New Site Name(EN)" |
| | UTC01-017 | คอลัมน์ "New Region" ใน Excel File ไม่กรอกข้อมูล |
| | UTC01-018 | คอลัมน์ "New Region" โดย กรอกข้อมูลคอลัมน์ "New Region" ใน Excel File ตรงกับ ข้อมูล คอลัมน์ "RegionName" ที่ List:Region |
| | UTC01-019 | คอลัมน์ "New Region" โดย กรอกข้อมูลคอลัมน์ "New Region" ใน Excel File ไม่ตรงกับ ข้อมูลคอลัมน์ "RegionName" ที่ List: Region |
| | UTC01-020 | คอลัมน์ "New Address" ใน Excel File ไม่กรอกข้อมูล |
| | UTC01-021 | คอลัมน์ "New Address" ใน Excel File กรอกข้อมูล ไม่เกิน 255 ตัวอักษร |
| | UTC01-022 | คอลัมน์ "New Address" ใน Excel File กรอกข้อมูล เท่ากับ 255 ตัวอักษร |
| | UTC01-023 | คอลัมน์ "New Address" ใน Excel File กรอกข้อมูล มากกว่า 255 ตัวอักษร |
| | Preview Data Excel FTTH | |
| | UTC01-024 | คอลัมน์ "SiteNo" ใน Excel File ไม่กรอกข้อมูล |
| | UTC01-025 | คอลัมน์ "SiteNo" ใน Excel File โดย กรอกข้อมูลคอลัมน์ "SiteNo" ใน Excel File ตรงกับ ข้อมูลคอลัมน์ "SiteNo" ที่ List: Space |
| | UTC01-026 | คอลัมน์ "SiteNo" ใน Excel File โดย กรอกข้อมูลคอลัมน์ "SiteNo" ใน Excel File ไม่ตรงกับ ข้อมูล คอลัมน์ "SiteNo" ที่ List: Space |
| | UTC01-027 | คอลัมน์ "Current Site Name(EN)" ใน Excel File ไม่กรอก ข้อมูล |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| Scenario ID | Test Case ID | Test Case Condition |
|-------------|--------------|---|
| UTS01 | UTC01-028 | ข้อมูลคอลัมน์ " Current Site Name(EN)" โดยข้อมูลคอลัมน์ "Current Site Name(EN)" ใน Excel Fileตรงกับ ข้อมูลคอลัมน์ "SiteNameEN" ที่ List:Space |
| | UTC01-029 | ข้อมูลคอลัมน์ " Current Site Name(EN)" โดย ข้อมูลคอลัมน์ " Current Site Name(EN)" ใน Excel File ไม่ตรงกับ ข้อมูลคอลัมน์ "SiteNameEN" ที่ List:Space |
| | UTC01-030 | คอลัมน์ " Current Site Name(TH)" ใน Excel File ไม่กรอกข้อมูล |
| | UTC01-031 | ข้อมูลคอลัมน์ " Current Site Name(TH)" โดย ข้อมูลคอลัมน์ "Current Site Name(TH)" ใน Excel File ตรงกับ ข้อมูลคอลัมน์ "SiteNameTH" ที่ List:Space |
| | UTC01-032 | ข้อมูลคอลัมน์ " Current Site Name(TH)" โดยข้อมูลคอลัมน์ "Current Site Name(TH)" ใน Excel File ไม่ตรงกับ ข้อมูลคอลัมน์ "SiteNameTH" ที่ List:Space |
| | UTC01-033 | คอลัมน์ "CurrentRegion" ใน Excel File ไม่กรอกข้อมูล |
| | UTC01-034 | ข้อมูลคอลัมน์ "Current Region"โดยข้อมูลคอลัมน์ "Current Region" ใน Excel File ตรงกับ ข้อมูลคอลัมน์ "RegionName" ที่ List:Region |
| | UTC01-035 | ข้อมูลคอลัมน์ "Current Region" โดยข้อมูลคอลัมน์ "Current Region" ใน Excel File ไม่ตรงกับ ข้อมูล คอลัมน์ "RegionName"ที่ List: Region |
| | UTC01-036 | คอลัมน์ " Current Address" ใน Excel File ไม่กรอกข้อมูล |
| | UTC01-037 | ข้อมูลคอลัมน์ "Current Address"โดย ข้อมูลคอลัมน์ " Current Address" ใน Excel File ตรงกับ ข้อมูลคอลัมน์ " Address" ที่ List:Location |
| | UTC01-038 | ข้อมูลคอลัมน์ "Current Address"โดย ข้อมูลคอลัมน์ " Current Address" ใน Excel File ไม่ตรงกับ ข้อมูลคอลัมน์ " Address" ที่ List:Location |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| Scenario ID | Test Case ID | Test Case Condition | |
|-------------|--|---|--|
| UTS01 | UTC01-039 | ข้อมูลใน Excel File มีชื่อ Column เหมือนกันมากกว่า 1 คอลัมน์ | |
| | UTC01-040 | เมื่อเพิ่ม หน้า Excel file Sheet 2 เป็นหน้าว่างแล้วนำมาไว้หน้าแรก | |
| | UTC01-041 | เมื่อใส่หัวข้อคอลัมน์ใน Excel File แต่ไม่มีข้อมูล | |
| | UTC01-042 | เมื่อใส่หัวข้อคอลัมน์ใน Excel file ไม่ครบตามที่กำหนด | |
| | UTC01-043 | เมื่อใส่หัวข้อไม่ครบคอลัมน์ใน Excel file เกินที่กำหนด | |
| | UTC01-044 | เมื่อกรอกหัวข้อคอลัมน์ ใน Excel file เป็นตัวเล็กทั้งหมด | |
| | UTC01-045 | เมื่อกรอกหัวข้อคอลัมน์ใน Excel file เป็นตัวใหญ่ทั้งหมด | |
| | UTC01-046 | เมื่อชื่อคอลัมน์ใน Excel file ไม่ตรงตาม รูปแบบที่กำหนด | |
| | Update Data Excel FTTH to SharePoint | | |
| | UTC01-047 | Import ข้อมูลคอลัมน์ "SiteNo" ใน Excel File | |
| UTC01-048 | Import ข้อมูลคอลัมน์ "Current Region" ใน Excel File | | |
| UTC01-049 | Import ข้อมูลคอลัมน์ "New Region" ใน Excel File | | |
| UTC01-050 | Import ข้อมูลคอลัมน์ "Current Site Name(EN)" ใน Excel File | | |
| UTC01-051 | Import ข้อมูลคอลัมน์ "New Site Name(EN)" ใน Excel File | | |
| UTC01-052 | Import ข้อมูลคอลัมน์ "Current Site Name(TH)" ใน Excel File | | |
| UTC01-053 | Import ข้อมูลคอลัมน์ "New Site Name(TH)" ใน Excel File | | |
| UTC01-054 | Import ข้อมูลคอลัมน์ "Current Address" ใน Excel File | | |
| UTC01-055 | Import ข้อมูลที่ไม่ถูกต้อง จากคอลัมน์ ใน Excel File | | |

3.5 การวางแผนการทำงาน (Work Plan)

ในการดำเนินงานได้วางแผนการระยะดำเนินงานประมาณ 20 สัปดาห์ โดยมีรายละเอียด งานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1 Week 1 – Week 2: การปรับพื้นฐาน

ก่อนเริ่มโครงการ มีการปรับพื้นฐานโปรแกรมและภาษาที่ใช้ในองค์กร เพื่อทดสอบความชอบในการทำงานที่ถูกต้องกับบุคลิกและความสามารถของตน มีการอบรมรับฟัง การทำงานปฏิบัติจริงและเลือกหัวข้อสทกิจ ซึ่งจะตอบโจทย์ในการทำงานและการศึกษาเพื่อทำหัวข้อต่อไป

3.5.2 Week 3: การกำหนดขอบเขต

จากการเลือกหัวข้อการทำสทกิจ จะมีการกำหนดขอบเขตของการศึกษาเกี่ยวกับสทกิจที่ทำ การศึกษาเครื่องมือและภาษาที่จะนำมาใช้ในการทำสทกิจ เพื่อนำมาใช้ในการปฏิบัติงานได้จริง

3.5.3 Week 4 - Week 5: ศึกษาขั้นตอนการทำงาน

ศึกษาเกี่ยวกับกระบวนการทำงานของผู้ทดสอบระบบ (Software Tester) เรียนรู้การออกแบบเอกสารการทดสอบทุกขั้นตอน ตั้งแต่ การวางแผนการทำงาน (Test Scenario) จนถึง การทำคู่มือการใช้งาน (User Manual Document) เพื่อส่งมอบให้ลูกค้าได้ศึกษาและเข้าใจ

3.5.4 Week 6 – Week 8: การทำเอกสารการทดสอบระบบ

มีการเริ่มวางแผนการทำงานทดสอบระบบ การเขียนเอกสารการทดสอบระบบทั้งหมด (Test Case) เพื่อนำมาทดสอบตามจริงกับระบบ ทดสอบโดยตนเอง(Manual Test) ทั้งหมดทุกระบบย่อย (Module) ในเว็บไซต์ที่รองรับการทำงานขององค์กร (SharePoint) ก่อนที่จะนำมาเขียนโปรแกรม (Test Script) และทดสอบโดยนำเครื่องมือมาช่วยลดระยะเวลา (Automation Test)

3.5.5 Week 9 – Week 11: ทดสอบระบบด้วยตนเอง (Manual Test)

การทดสอบในขั้นตอนนี้จะทดสอบตามขั้นตอนการทดสอบระบบ (Test Case) ที่ได้ออกแบบไว้ เพื่อหาข้อมูลที่ต้องการและตรวจสอบข้อมูลที่ผิดพลาดของระบบทั้งหมดเพื่อส่งข้อมูลที่ผิดพลาดของระบบให้ผู้เขียนระบบ(Programmer) ได้แก้ไขและนำกลับมาทดสอบตามขั้นตอนการทดสอบระบบ (Test Case) อีกครั้ง (Re-Test)

3.5.6 Week 12 – Week 17: การเขียนโปรแกรม (Test Script)

การเขียนโปรแกรมโดยใช้ภาษา JavaScript โดยนำเครื่องมือ PhantomJS ที่ได้ศึกษาว่าสามารถใช้งานได้ง่าย และนำไปใช้ทดสอบระบบให้ลูกค้าได้จริง โดยมีการศึกษาภาษาในการใช้เขียนโปรแกรมและนำไปเขียนตามขั้นตอนการทดสอบระบบ (Test Case) โดยให้สามารถใช้ระบบได้ถูกต้อง โดยข้อมูลจากการทดสอบระบบทั้งหมดจะถูกนำมาเป็นเอกสารเดียวกัน เพื่อให้ นักทดสอบระบบ (Software Tester) ได้ดูข้อมูลที่ต้องการและผิดพลาด พร้อมทั้งสามารถดูรูปภาพประกอบข้อมูลที่ถูกต้องและผิดพลาดได้ เพื่อทราบระบบการทำงานและนำไปแก้ไข

3.5.7 Week 18 – Week 19: ทดสอบโดยใช้เครื่องมือ (Automation Test)

นำโปรแกรมที่เขียน (Test Script) ในระบบย่อย (Module Space) ทดสอบบนเครื่องมือ PhantomJS ว่าสามารถทดสอบตามระบบโดยผ่านทั้งหมด และถ้าพบข้อมูลที่ผิดพลาด (Error) จะนำโปรแกรมมาแก้ไขและกลับมาทดสอบใหม่ (Re-Test) ทั้งหมด โดยการนำไฟล์ที่เขียนโปรแกรม (Test Script) ทดสอบผ่านเครื่องมือ PhantomJS อีกครั้ง

3.5.8 Week 20: การจัดทำเอกสารคู่มือการใช้งานของระบบ (User Manual Document)

จากการทดสอบทั้งหมดจะมีการจัดทำเอกสารการเขียนตัวอย่างพร้อมข้อมูลเพื่อให้ลูกค้าทดสอบตามระบบ (Test Script) และจัดทำคู่มือการใช้งานของระบบ (User Manual Document) เพื่อให้ลูกค้าได้เข้าใจการทำงานของระบบ และทดสอบกระบวนการของการทดสอบระบบก่อนใช้งานจริง (User Acceptance Test) เพื่อตรวจสอบว่าสามารถตอบสนองต่อความต้องการของลูกค้า (Requirement) ตรงตาม Business Flow จริงของลูกค้า ในระดับที่ยอมรับได้ และทดสอบที่สภาพ Environment ที่ใกล้เคียงกับ Production มากที่สุด โดยการทำงานทั้งหมดจะมีพี่เลี้ยงของบริษัทในตำแหน่งนักทดสอบระบบ (Software Tester) คอยดูแลและให้คำปรึกษาอย่างใกล้ชิดตลอดการทำงาน

3.6 สถาปัตยกรรมคอมพิวเตอร์ (2-Tier Architecture)



รูปที่ 3.10 สถาปัตยกรรมคอมพิวเตอร์ (2-Tier Architecture) ของระบบ

การดำเนินการของซอฟต์แวร์ในเครื่องผู้ใช้บริการ (client) ที่เชื่อมต่อกันอยู่ แล้วส่งไปดำเนินการต่อที่เครื่องเซิร์ฟเวอร์ที่มีไว้เพื่อรันระบบที่เป็นฐานข้อมูล (Database server) ก็จะเป็นการเข้าถึงได้โดยตรง และต้องรับผิดชอบความเสี่ยงในระบบอินเทอร์เน็ต (Network) พวก รหัสผ่าน (Password) ก็จะถูกใช้งานได้โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

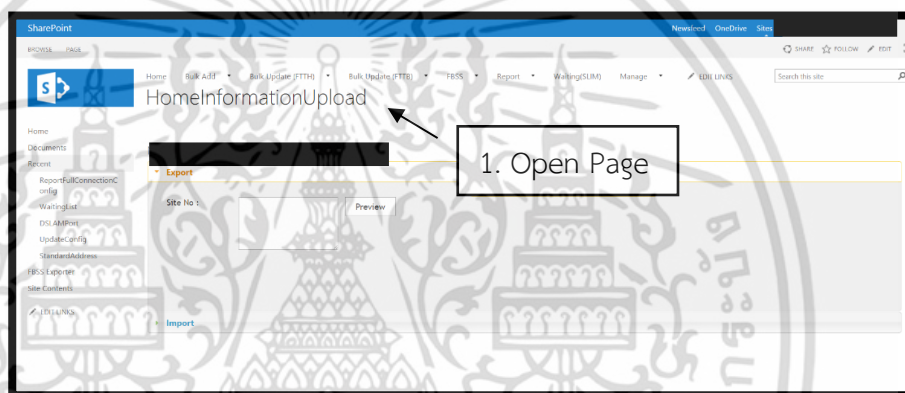
บทที่ 4

ผลการวิจัยและการอภิปรายผล

จากที่ได้กล่าวไว้ในบทที่ 3 ว่า มีการแบ่งการทำงานออกเป็น 3 ส่วน คือส่วนของการทำงานนำข้อมูลออกสู่ระบบ (Export Space) ข้อมูลส่วนการนำไฟล์เข้าสู่ระบบ (Import Sapce) ส่วนสุดท้ายคือการตรวจสอบข้อมูลที่ถูกบันทึกเข้าสู่ระบบ (Check List Space) และการนำทั้ง 3 ส่วน แสดงผลออกเป็นไฟล์เดียวกัน ดังนั้น จึงนำมาสรุปผลการดำเนินงานได้ ดังนี้

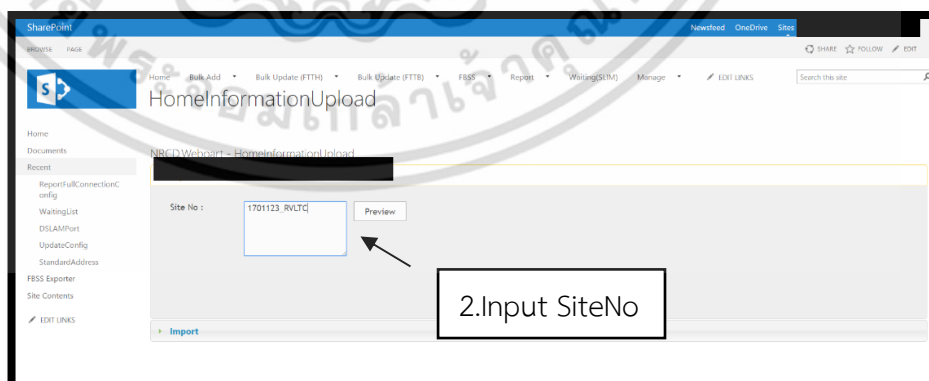
4.1 ผลการดำเนินงานหน้า Export Space

4.1.1 การเข้าหน้า Page โดยกำหนด URL และ Username , Password



รูปที่ 4.1 แสดงหน้าจอหลัก Page

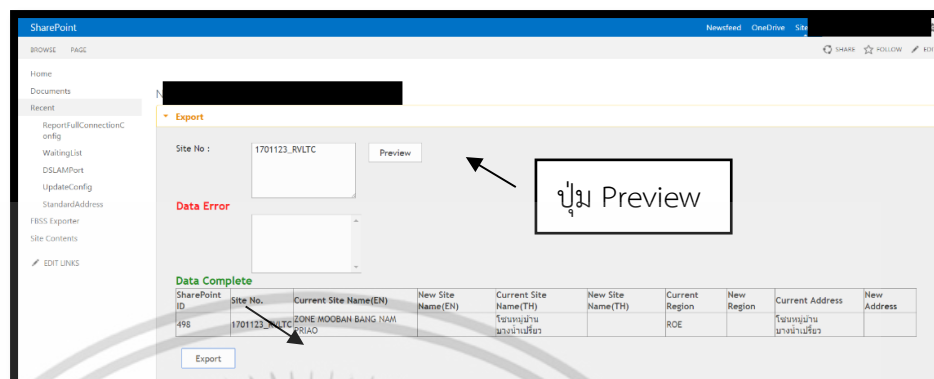
4.1.2 การกรอกข้อมูลรหัส SiteNo ในหน้า Export Space



รูปที่ 4.2 แสดงหน้าจอการกรอกข้อมูล SiteNo

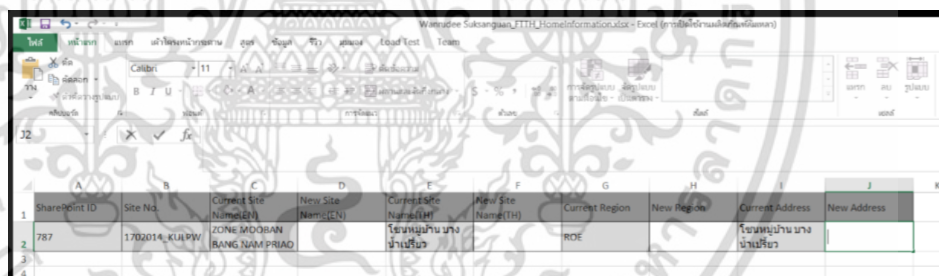
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 การกดปุ่ม Preview เพื่อแสดงข้อมูลของระบบที่ต้องการ



รูปที่ 4.3 แสดงหน้าจอข้อมูลแบบฟอร์มที่ต้องการ

4.1.4 ข้อมูลถูกนำออก (Export) ไฟล์ excel ออกมาเป็นแบบฟอร์มที่ต้องการเพื่อนำมาแก้ไขข้อมูล

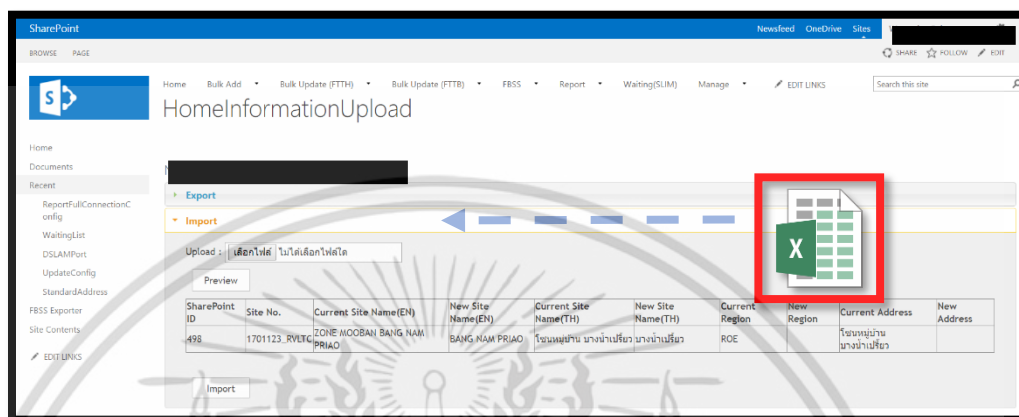


รูปที่ 4.4 หน้าแบบฟอร์มที่ถูกนำออกมาแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

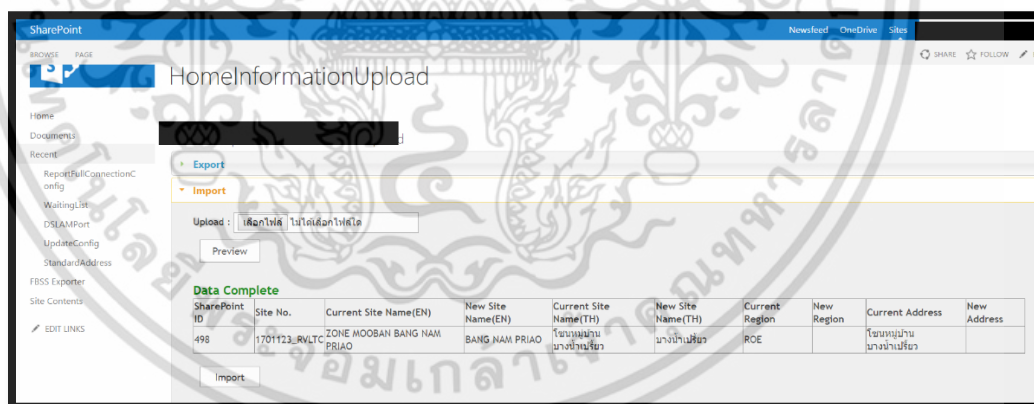
4.2 ผลการดำเนินงานหน้า Import Space

4.2.1 แสดงหน้า Tab Import เพื่อเลือกไฟล์ข้อมูล พอกดปุ่ม Preview จะแสดงข้อมูลที่ถูกแก้ไขเพื่อให้บันทึก (Import) เข้าสู่ระบบ



รูปที่ 4.5 แสดงหน้าจอการเลือกไฟล์ข้อมูลและแสดงข้อมูลที่ถูกแก้ไข

4.2.2 เลือกไฟล์ข้อมูลที่ถูกแก้ไข เมื่อเลือกเสร็จแล้วกดปุ่ม Preview เพื่อแสดงข้อมูลที่ถูกแก้ไข (Update) ข้อมูลที่ถูกต้องจะถูกบันทึก (Import) เข้าสู่ระบบ



รูปที่ 4.6 แสดงข้อมูลที่แก้ไข (Update) ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการดำเนินงานหน้า Check List Space

4.3.1 แสดงหน้าจอรายการ List: Space เพื่อตรวจสอบข้อมูลที่แก้ไขถูกต้อง

| SiteCode | SiteNo | SiteType | Latitude | Longitude | PropertyRight | Location | LocationID | LocationNameTH | LocationNameEN | SiteNameEN | SiteNameTH |
|----------|---------------|----------|-----------|-----------|---------------|---------------|------------|--------------------------|----------------|----------------------------|--------------------------|
| RVLTB | 1701092_RVLTB | OLT | 19.851919 | 99.999206 | AWN | 1701092_RVLTB | 1701092 | โซมหมู่บ้าน นางน้ำเป็ริอ | BANG NAM PRIAO | ZONE MCOBAN BANG NAM PRIAO | โซมหมู่บ้าน นางน้ำเป็ริอ |
| REVOL | 111216_REVOL | OLT | 6.111111 | 98.111111 | AWN | 111216_REVOL | 111216 | สำเร็จ2 | Success2 | Success2 | สำเร็จ2 |
| RVLTA | 1701121_RVLTA | OLT | 19.851919 | 99.999206 | AWN | 1701121_RVLTA | 1701121 | โซมหมู่บ้าน นางน้ำเป็ริอ | BANG NAM PRIAO | ZONE MCOBAN BANG NAM PRIAO | โซมหมู่บ้าน นางน้ำเป็ริอ |
| RVLTC | 1701123_RVLTC | SPT | 19.851919 | 99.999206 | AWN | 1701123_RVLTC | 1701123 | โซมหมู่บ้าน นางน้ำเป็ริอ | BANG NAM PRIAO | ZONE MCOBAN BANG NAM PRIAO | โซมหมู่บ้าน นางน้ำเป็ริอ |

รูปที่ 4.7 แสดงหน้าข้อมูลที่ถูกรแก้ไข (Update) โดยตรวจสอบที่ LocationNameEN

4.3.2 แสดงผลทั้ง 3 ส่วน นำมารวมกันเป็นไฟล์ .csv จะแสดงวันที่และเวลาในการทดสอบ พร้อมแสดงรูปภาพในการทดสอบทั้งกรณีที่ถูกและกรณีที่ผิด

| ScenarioNo | ScenarioName | Action | DateTime | status | Images |
|------------|---------------------------------|---|-----------------|--------|------------|
| 1 | 1 Bulk Update Space Information | Export>> | | | |
| 4 | | ACTION 1/10: Open Page Update Information Export Space | | Pass | view image |
| 5 | | ACTION 2/10: input Site No | | Pass | view image |
| 6 | | ACTION 3/10: Preview Data UpdateSpace | | Pass | view image |
| 7 | | ACTION 4/10: Check Preview Data Update | | Pass | view image |
| 8 | | ACTION 5/10: Finished download | | Pass | view image |
| 9 | | | 1/13/2017 14:25 | | |
| 11 | 2 Bulk Update Space Information | Import>> | | | |
| 14 | | ACTION 6/10: Open Page Update Information Import Space | | Pass | view image |
| 15 | | ACTION 7/10: Bulk file UpdateSpace | | Pass | view image |
| 16 | | Preview Data Space Process.. | | | |
| 17 | | ACTION 8/10: Preview Data UpdateSpace | | Pass | view image |
| 18 | | Import Data Space Process.. | | | |
| 19 | | ACTION 9/10: Import Data Complete | | Pass | view image |
| 20 | | | 1/13/2017 14:25 | | |
| 22 | 3 Bulk Update Space Information | Checklist>> | | | |
| 24 | | ACTION 10/10: Check Update LocationEN:"BANG NAM PRIAO" at SiteNo:"1701121_RVLTA" Complete | | Pass | view image |
| 25 | | | 1/13/2017 14:25 | | |

รูปที่ 4.8 แบบฟอร์มในการแสดงผลข้อมูลทั้ง 3 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลกรณีศึกษา

จากการศึกษาค้นคว้าการนำเครื่องมือ PhantomJS มาช่วยในการทดสอบระบบแบบมีเครื่องมือมาช่วย (Automation Test) โดยจะมีการทดสอบด้วยตนเอง (Manual Test) ควบคู่กันไป และเขียนโปรแกรมเป็นภาษา JavaScript ซึ่งเป็นภาษาที่สามารถเรียนรู้และค้นคว้าเพื่อนำมาศึกษาในการเขียนได้เข้าใจง่าย โดยการทดสอบระบบที่จะทำให้เกิดประสิทธิภาพเนื่องจากมีการแก้ไขและต้องเกิดการทดสอบแบบวนซ้ำ (Re-Test) การนำระบบแบบมีเครื่องมือมาช่วย (Automation Test) จึงมีบทบาทสำคัญที่จะช่วยลดระยะเวลาในการทดสอบของผู้ทดสอบ (Tester) ได้เป็นอย่างดี และจากการทดสอบระบบการนำเครื่องมือ PhantomJS มาช่วยสามารถติดตั้งได้ง่ายและนำไปใช้กับลูกค้านอกสถานที่ได้จริงจึงเป็นประโยชน์แก่บริษัทและสามารถนำไปต่อยอดได้ดี

5.2 ประโยชน์ที่ได้รับ

5.2.1 ด้านองค์กร

- สามารถช่วยลดระยะเวลาในการทำงานให้มีประสิทธิภาพมากขึ้น
- สามารถช่วยตรวจสอบข้อผิดพลาดการทดสอบระบบแบบวนซ้ำ (Re-Test)
- สามารถนำโปรแกรม (Script) ที่ใช้ในการทดสอบแล้วกลับมาใช้งานได้อีก

5.2.2 ด้านผู้ศึกษาและวิจัย

- ได้เรียนรู้การทดสอบระบบแบบการทดสอบด้วยตนเอง (Manual Test)
- ได้รู้จักแก้ไขปัญหาในการทำงานที่จะต้องถูกเปลี่ยนความต้องการตลอดเวลา
- ได้เพิ่มคุณลักษณะให้ตนเอง เช่น การตรงต่อเวลา ความรับผิดชอบ ความอดทน
- ได้เพิ่มความสามารถในการใช้เครื่องมือต่างๆ ที่ใช้ในการทำงานจริง
- ได้วิเคราะห์การทำงานของระบบ สามารถเข้าใจหลักการคิดของระบบได้ง่ายขึ้น
- สามารถเข้าใจความต้องการของผู้ใช้งานระบบและอธิบายให้ผู้อื่นได้เข้าใจได้มากขึ้น

5.3 ข้อเสนอแนะ

1) ในการทำงานจริงต้องคิดขั้นตอนการทดสอบและทดสอบอย่างถี่ถ้วนและรอบคอบที่สุด เพราะถ้าเกิดความผิดพลาดอาจเกิดความเสียหายมากมายที่เกิดขึ้นในระบบ

2) ในการทดสอบจำเป็นต้องใช้ข้อมูลที่เหมือนจริงมากที่สุด เพื่อการทดสอบที่แม่นยำและมีประสิทธิภาพ

3) เนื่องจากความต้องการอาจมีการเปลี่ยนแปลงได้ตลอดเวลา ดังนั้นในการพัฒนาต่อควรมีการตรวจสอบความถูกต้องที่มีประสิทธิภาพ

4) ในความเป็นจริงชีวิตการทำงานต้องแบ่งส่วนการทำงานและต้องทำงานเป็นทีม การตัดสินใจอะไรก็ควรต้องได้รับความเห็นชอบจากทีมเสียก่อน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] ทศนีย์ คัดเจริญ. 2017. เรื่องงง ๆ ของ Tester มือใหม่. [Online].
Available : <http://www.stream.co.th/2016/06/เรื่องงงๆ-tester-ของ-tester-มือใหม่/>
- [2] รศ.รังสิต ศิริรังษี. 2014. Automation Tester คืออะไร ?. [Online].
Available : http://www.indythaitester.com/TesterTraining/TrainingBlog_6.jsp
- [3] เอกภิน ใจแก้วมา (แม็ก). 2015. ดาวโหลดและติดตั้ง Nodejs บน Windows. [Online].
Available : <https://drivesoftcenter.net/tutorial/javascript/nodejs/nodejs-ดาวโหลดและติดตั้ง-nodejs-บน-windows>
- [4] Accenture. 2014. Data Conversion for SAP. [Online]. Available : https://www.accenture.com/t20160419T225657w/us-en/_acnmedia/PDF-14/Accenture-Data-Conversion-For-SAP-Brochure.pdf
- [5] AorJoa. 2013. N tier (tier - 2 tier - 3 tier) มันคืออะไร?. [Online].
Available : <http://aorjoa.blogspot.com/2013/11/n-tier-tier-2-tier-3-tier.html>
- [6] Charathbank. 2010. QA Knowledge :: Testing Type มีอะไรบ้าง. [Online].
Available : <https://charathbank.wordpress.com/2010/10/04/qa-knowledge-testing-type/>
- [7] Chai Phonbopit. 2015. Node.js คืออะไร ? + เริ่มต้นใช้งาน Node.js. [Online].
Available : <https://devahoy.com/posts/getting-started-with-nodejs/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[8] Client/Server. 2017. Client Server คืออะไร ไคลเอ็นท์ เซิร์ฟเวอร์ คือระบบเครือข่ายคอมพิวเตอร์ซึ่งประกอบด้วยเครื่องคอมพิวเตอร์ที่ให้บริการ server เครื่องคอมพิวเตอร์ที่ขอรับบริการ client . [Online]. Available : <http://www.mindphp.com/คู่มือ/73คืออะไร/2052-client-serverคืออะไร.html>

[9] Goft Saisopa. 2014. วิธีการติดตั้ง SharePoint Server 2016. [Online].
Available : <https://www.nongit.com/blog/install-sharepoint-server-2016.html>

[10] Ms SharePoint. 2017. Microsoft SharePoint คืออะไร ไมโครซอฟท์ แชนร์พอยท์ คือระบบแชร์ ข้อมูล ผ่าน web browser . [Online]. Available : <http://www.mindphp.com/คู่มือ/73-คืออะไร/2437-ms-sharepoint-คืออะไร.html>

[11] Node.js. 2015. เจาะลึกกับ node.js แบบเริ่มต้นทำความรู้จัก. [Online].
Available : <http://meewebfree.com/site/nodejs/441-learn-about-node-js-with-basic-of-node-js>

[12] Premal Dave. 2016. Top 10 Automated Software Testing Tools. [Online].
Available : <https://dzone.com/articles/top-10-automated-software-testing-tools>

[13] Sarin Kesphanich. 2016. วิธีติดตั้งและใช้งาน Sublime Linter. [Online].
Available : <https://medium.com/hezagon/วิธีติดตั้งและใช้งาน-sublime-linter-1a9ad8e30699>

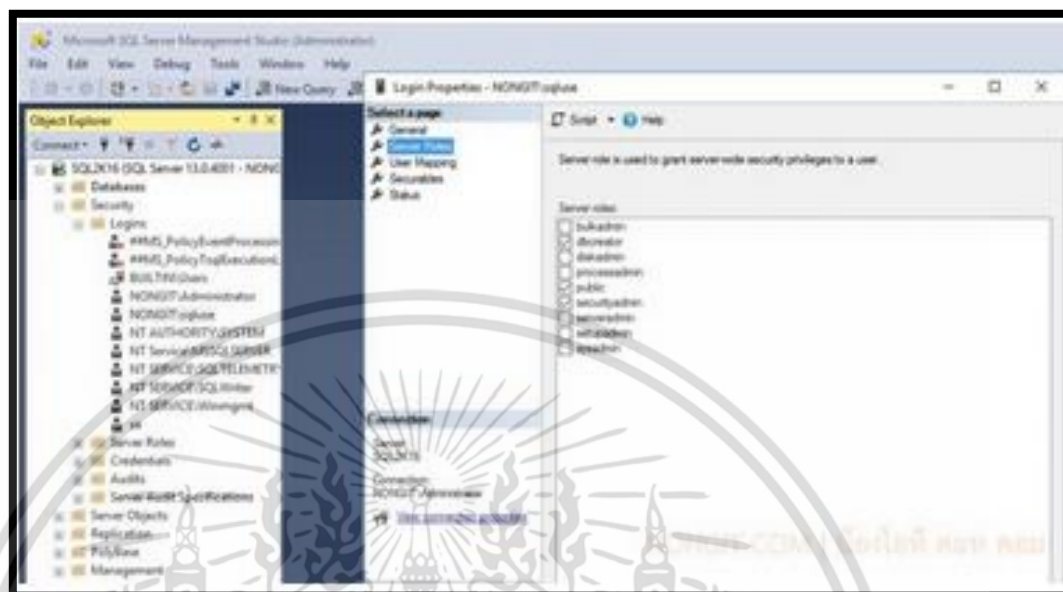
[14] Somkiat. 2017.ทำไมถึงต้องการ Automated Test กันจัง ? . [Online].
Available : <http://www.somkiat.cc/why-we-need-automated-test/>

- [15] Suranart Namcome. 2014. **รีวิวการติดตั้ง SharePoint Server 2016 บน Windows Server 2016**. [Online]. Available : <http://www.siamhtml.com/introduction-to-phantomjs/>
- [16] Suranart Niamcome. 2014. **Node.js คืออะไร ? + สอนวิธีใช้**. [Online]. Available : <http://www.siamhtml.com/introduction-to-node-js/>
- [17] Teerasej Jiraphatchandej. 2015. **เรียนรู้วิธีใช้ Sublime Text ง่ายๆ สำหรับ Designer/ Developer**. [Online]. Available : <http://nextflow.in.th/sublime-text/>
- [18] Zyracuze. 2016. **จะทดสอบอย่างไรให้รู้ผลได้ภายใน 10 นาที ก่อนไปมองหา Automate Testing Tools**. [Online]. Available : <http://www.welovebug.com/tag/automate-testing/>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) สิทธิ์สำหรับสร้าง data base ใน sql server โดยต้องมี securityadmin และ dbcreator



รูปที่ ก.2 Configure และ manage server farm / ใช้ Run Microsoft SharePoint

4) SharePoint Farm Account เป็น user ที่ใช้สำหรับจัดการ Configure และ manage server farm/ ใช้ Run Microsoft SharePoint Foundation Workflow Timer Service Software Requirements ที่ต้องติดตั้งให้เรียบร้อย

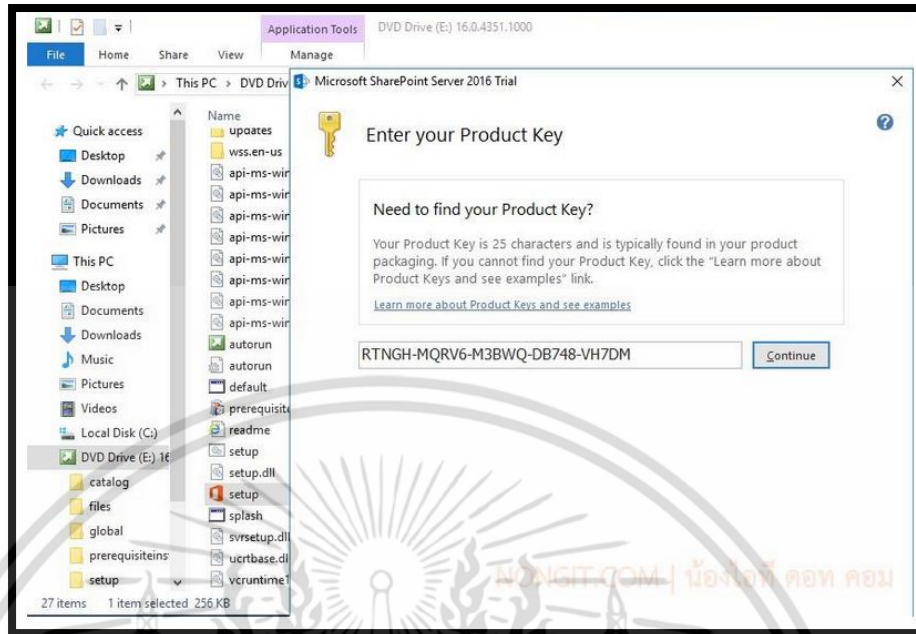
- SQL Server 2014 หรือ 2016 (วิธีติดตั้ง SQL Server 2016)
- SharePoint requirements (วิธีติดตั้ง sharepoint requirements offline)

ก.2 ขั้นตอนการติดตั้งโปรแกรม SAP GUI

1) เปิดไฟล์ setup.exe จากนั้นให้ใส่ Product key ของ SharePoint 2016

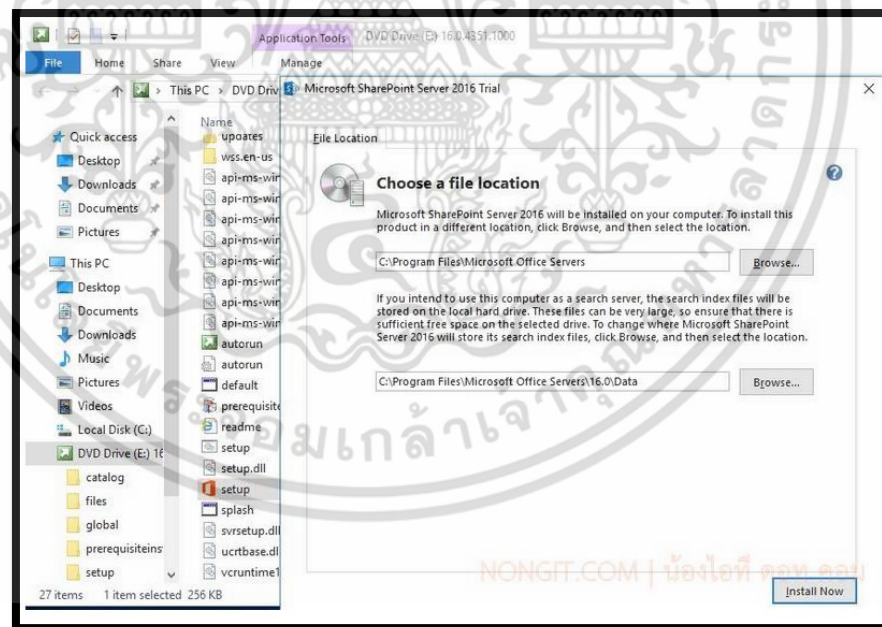
- Enterprise: NQGJR-63HC8-XCRQH-MYVCH-3J3QR
- Standard: RTNGH-MQRV6-M3BWQ-DB748-VH7DM
- 180 days

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.3 เปิดไฟล์ setup.exe

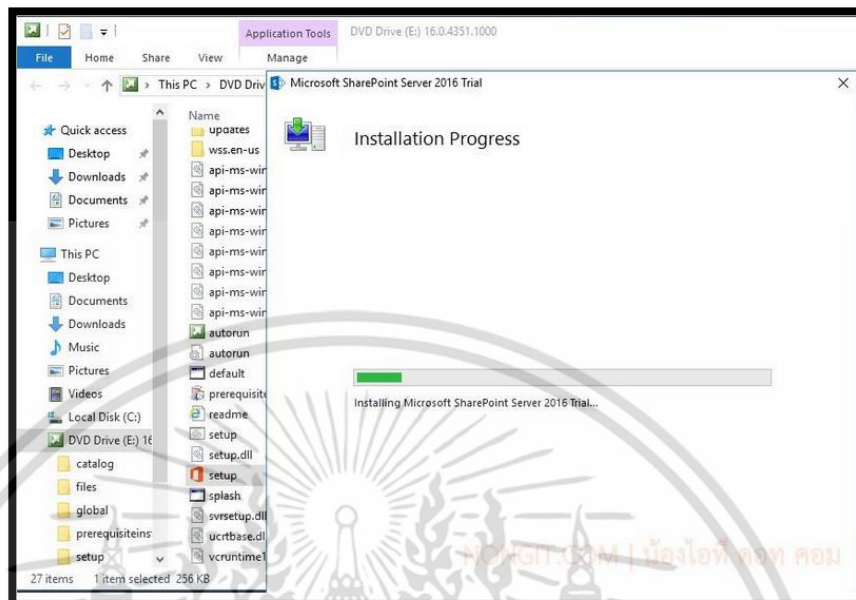
2) กด install now



รูปที่ ก.4 กด install now

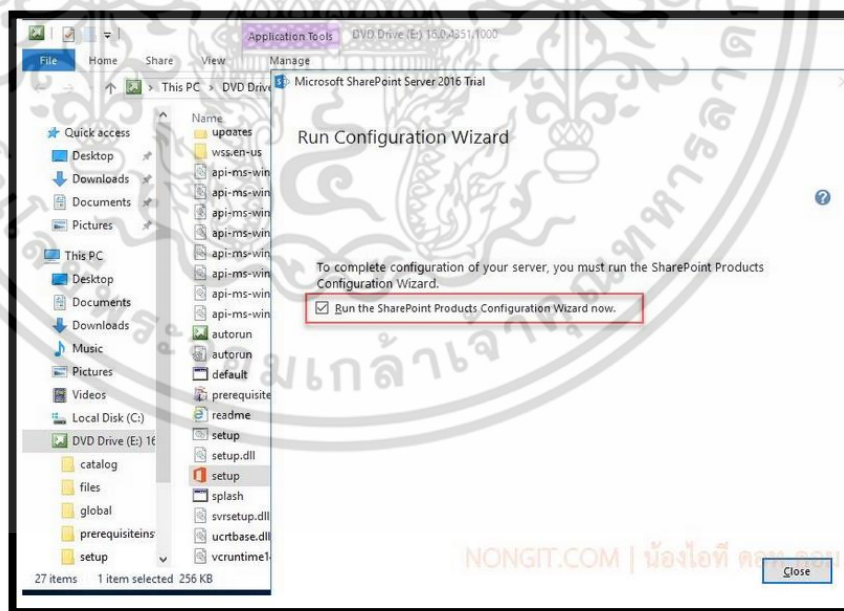
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) รอกการติดตั้ง



รูปที่ ก.5 รอกการติดตั้ง

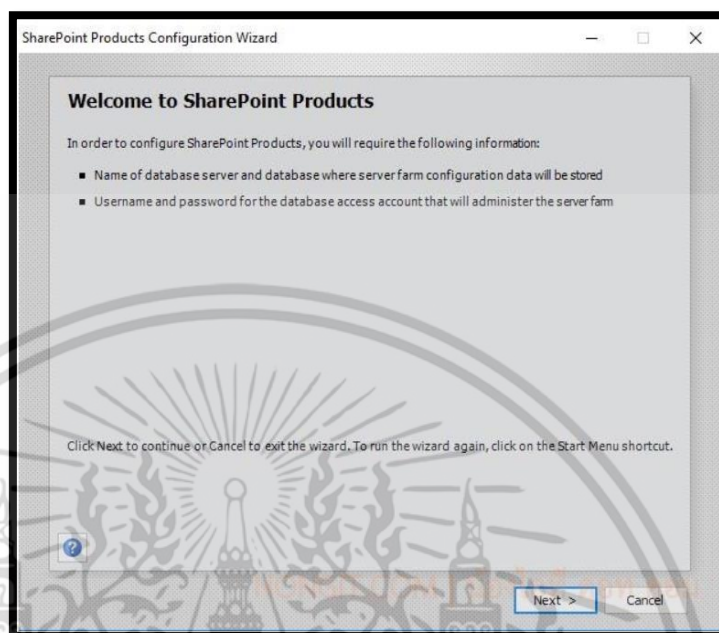
4) ติดตั้งเสร็จเรียบร้อย จากนั้นเลือกถูกที่ Run the SharePoint Products Configuration Wizard เพื่อสร้าง Farm



รูปที่ ก.6 รอกการติดตั้ง

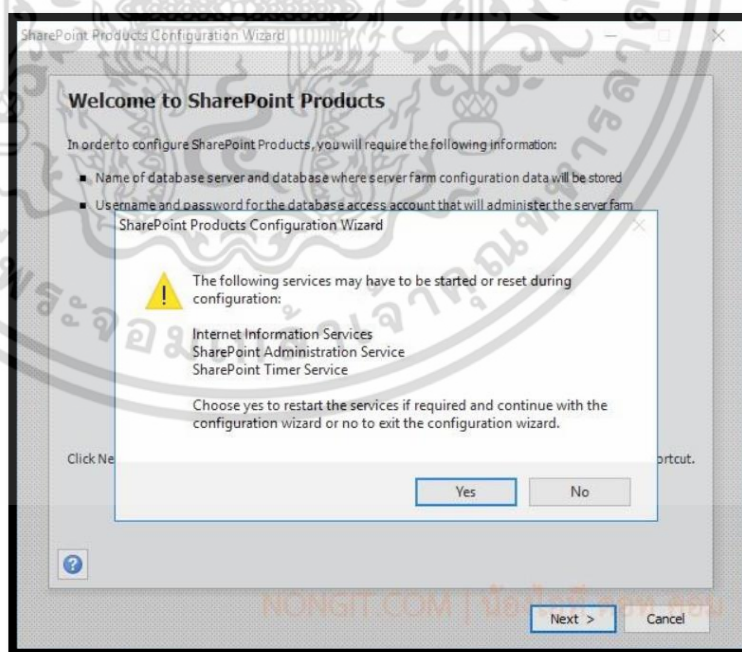
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) ที่หน้า SharePoint Products Configuration Wizard กด Next



รูปที่ ก.7 เริ่มเข้าหน้า SharePoint Products กด Next

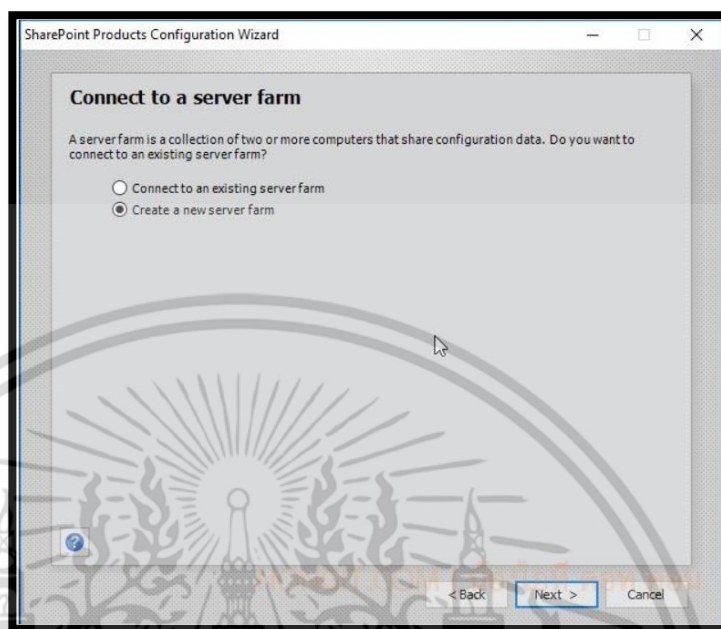
6) กด Yes ยืนยัน restart service



รูปที่ ก.8 เริ่มเข้าหน้า SharePoint Products กด Yes ยืนยัน restart service

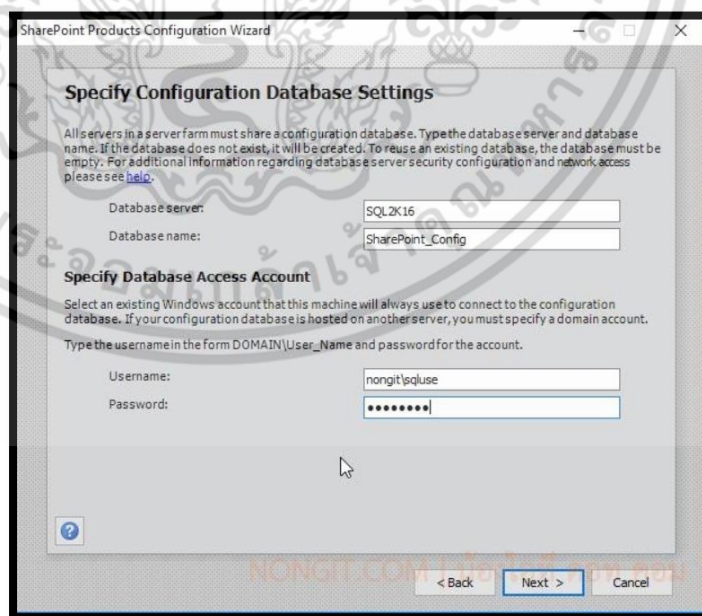
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) เลือก Create a new server farm กด next



รูปที่ ก.9 เลือก Create a new server farm กด next

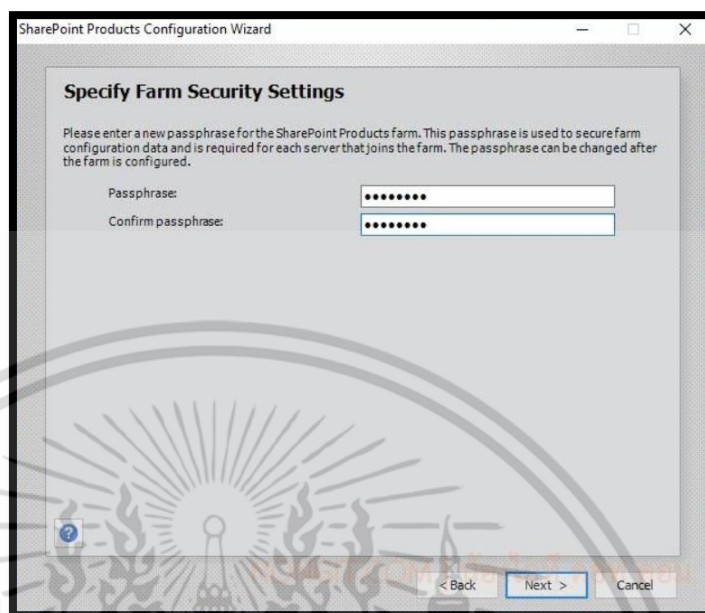
8) มาถึงหน้าในส่วนของ Specify configuration database settings ให้ใส่ Database server name และ ใส่ user password สำหรับ access data base



รูปที่ ก.10 ใส่ user password สำหรับ access data base

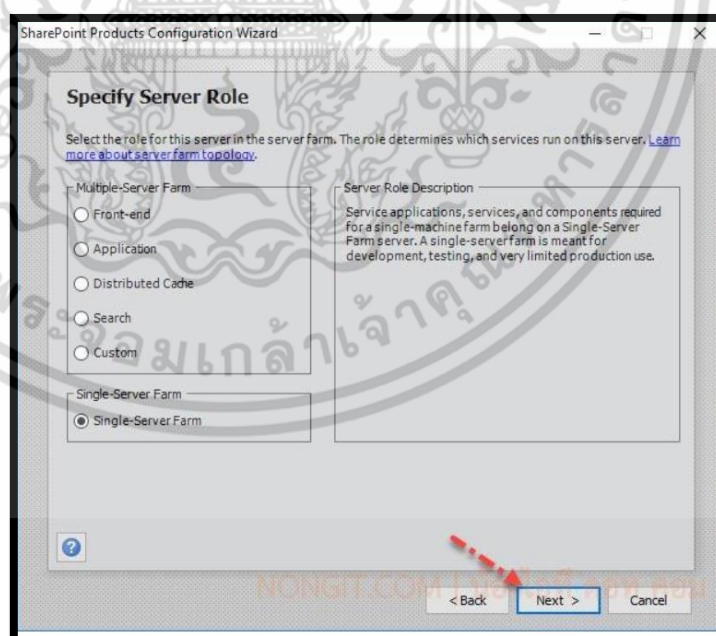
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9) กำหนด Passphrase



รูปที่ ก.11 กำหนด Passphrase

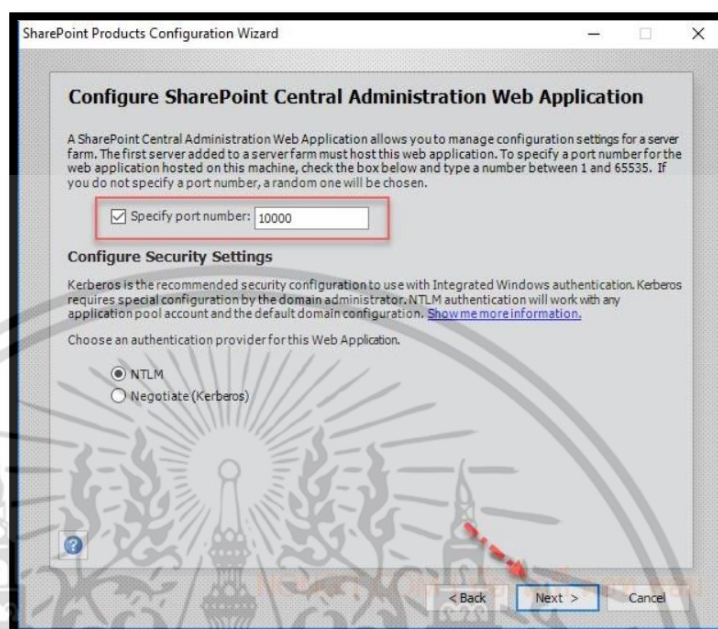
10) เลือก Single-Server Farm สำหรับ Server เครื่องเดียว และหากมี server หลายเครื่องโดยแยก Application server role ก็ให้เลือกตาม Role ที่กำหนดไว้



รูปที่ ก.12 หากมี server หลายเครื่องให้เลือกตาม Role ที่กำหนดไว้

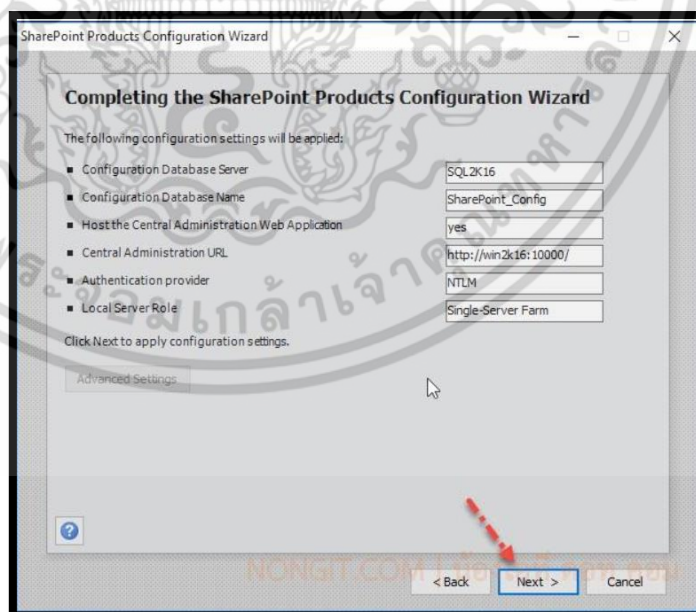
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 11) กำหนด Port สำหรับ Central Administration Site โดยที่ต้องไม่ซ้ำกับ service อื่นๆ และเลือก authentication provide ตามที่ต้องการ



รูปที่ ก.13 กำหนด Port สำหรับ Central Administration Site

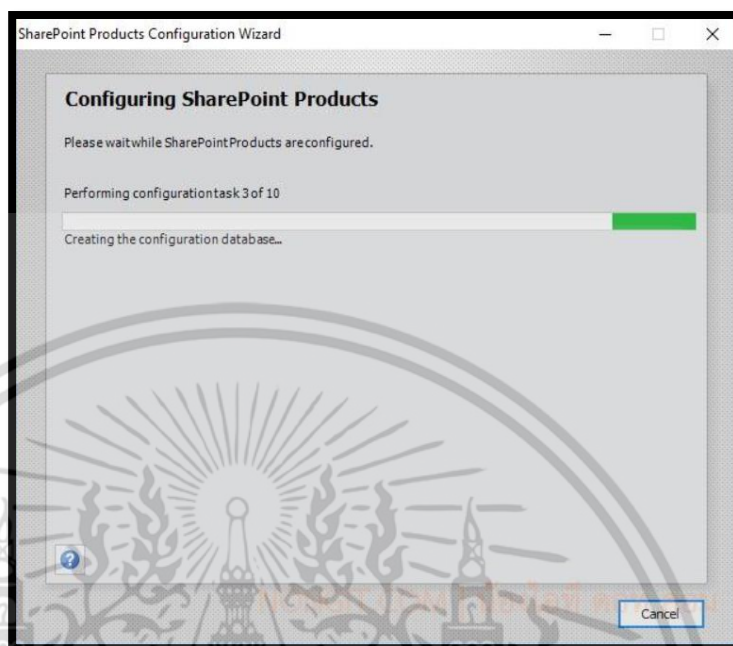
- 12) หน้าสรุปข้อมูลต่างๆที่จะตั้งค่า



รูปที่ ก.14 หน้าสรุปข้อมูลต่างๆที่จะตั้งค่า

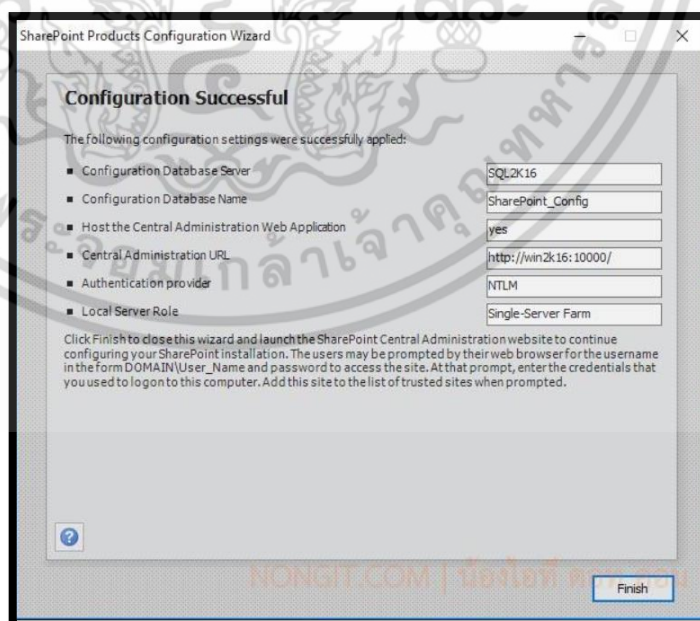
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13) รอการติดตั้งเข้า Server



รูปที่ ก.15 รอการติดตั้งเข้า Server

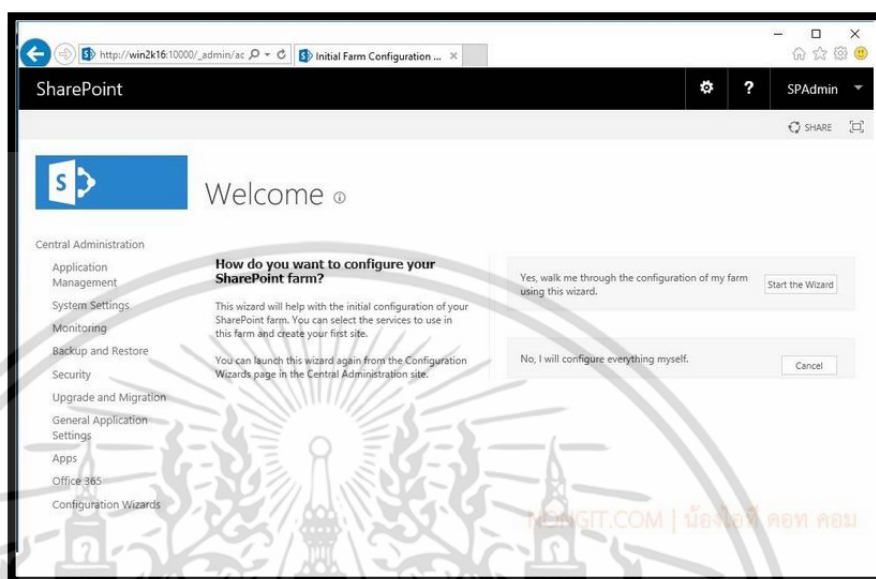
14) Configuration successful เมื่อเสร็จแล้วจะได้ข้อมูลที่ได้ทำการตั้งค่าไปและ data base ที่สร้าง



รูปที่ ก.16 การติดตั้งเสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15) หลังจากกด Finish จะเข้าหน้า SharePoint 2016 Central Administration จากนี้ก็เป็นขั้นตอนของการสร้าง Web Application สร้าง Site ต่อไป



รูปที่ ก.17 การเริ่มใช้งานหน้าแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

วิธีการติดตั้งและใช้งาน Node.js

ข.1 ขั้นตอนการติดตั้งโปรแกรม Node JS

1) เข้าเว็บไซต์ nodejs.org แล้วทำการดาวน์โหลดเวอร์ชัน v4.4.7



รูปที่ ข.1 การเริ่มใช้งานหน้าแรก

2) เมื่อดาวน์โหลดเสร็จแล้ว จะได้ไฟล์ install node js มา ให้ทำการ Double click เพื่อติดตั้ง



รูปที่ ข.2 ไฟล์ Install node js

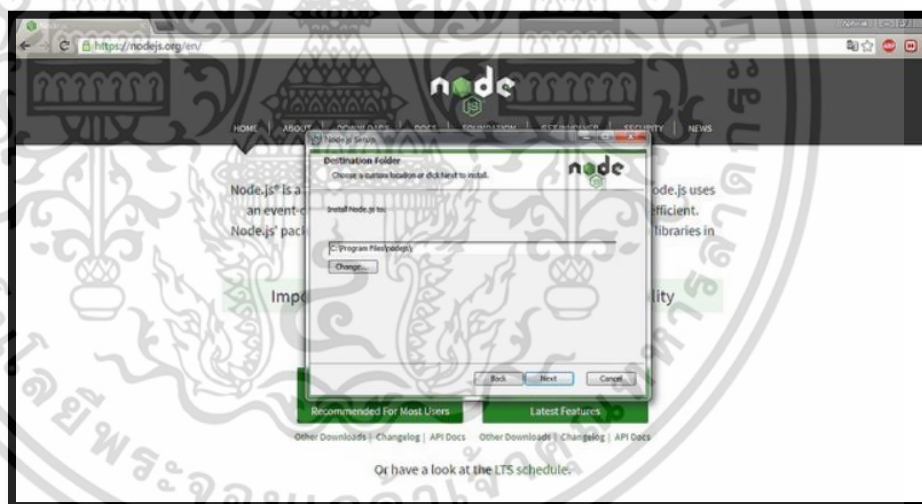
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) การลงโปรแกรมก็กด Next



รูปที่ ข.3 การลงโปรแกรมโดยการกด Next

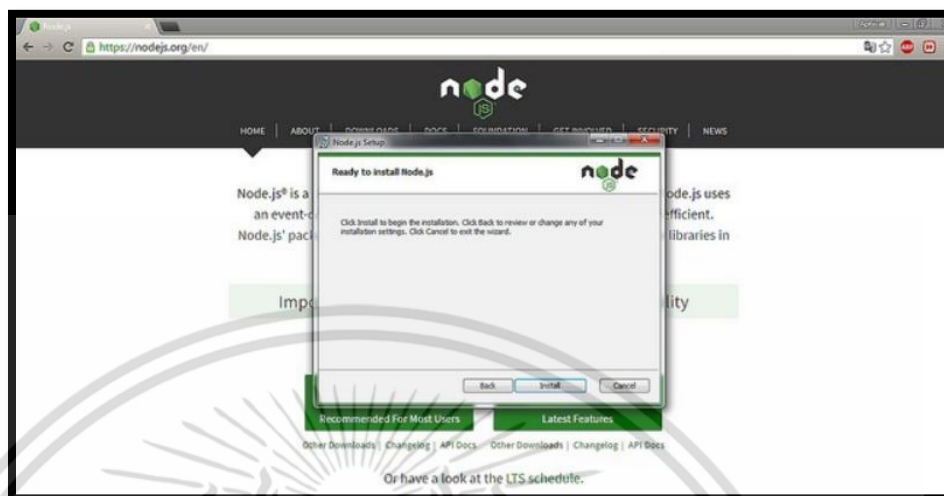
4) การเลือกโปรแกรมลงไดร์ฟ



รูปที่ ข.4 การเลือกไดร์ฟลงโปรแกรม

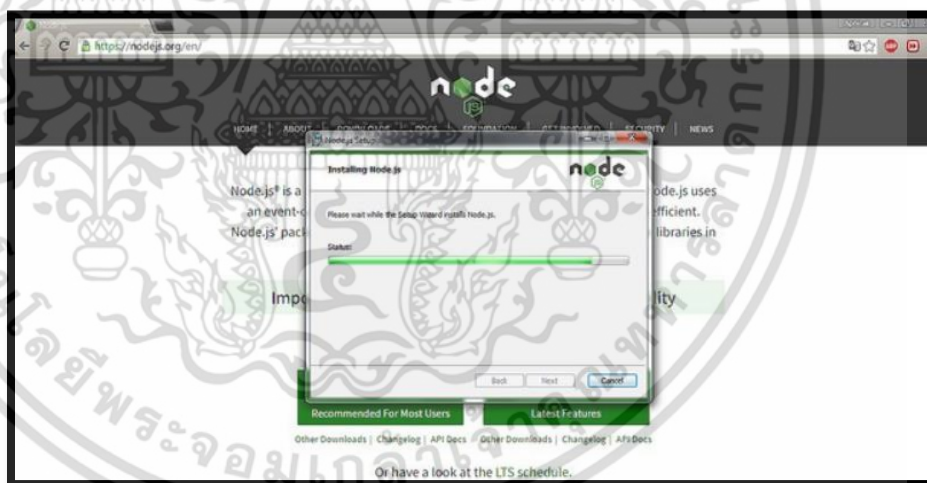
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) เลือกลงไฟล์เสร็จก็กด install



รูปที่ ข.5 การกด install Program

6) รอการโหลดจนเสร็จการติดตั้ง



รูปที่ ข.6 รอการโหลดติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) กด Finish เป็นการลงโปรแกรมเสร็จสมบูรณ์



รูปที่ ข.7 รอกการโหลดติดตั้ง

8) เข้า Start -> cmd -> node -v และ npm -v ก็ก็จะแสดงเวอร์ชันของ node js และ npm ที่ติดตั้ง



รูปที่ ข.8 การตั้งค่าการใช้งาน Node js

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

วิธีการติดตั้ง PhantomJS สำหรับ Windows

ค.1 ขั้นตอนการลงโปรแกรมเพื่อใช้งาน

1) Install Phantomjs (หลังจาก install Node js สำเร็จ)

- พิมพ์ `npm install -g phantomjs` ที่ command-line

```

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Wacha\Desktop\MyProject>npm install -g phantomjs
  
```

รูปที่ ค.1 Install Phantomjs

- รอจนโหลดสำเร็จจะมีการแสดงที่หน้า command-line

```

C:\Users\Wacha\Desktop\MyProject>npm install -g phantomjs
npm WARN deprecated phantomjs@2.1.7: Package renamed to phantomjs-prebuilt. Please update phantomjs package references to 'phantomjs-prebuilt'.
npm WARN deprecated node-uuid@1.4.8: Use uuid module instead
npm WARN deprecated tough-cookie@2.2.2: ReDoS vulnerability parsing Set-Cookie header
https://nodesecurity.io/advisories/120
C:\Users\Wacha\AppData\Roaming\npm\phantomjs -> C:\Users\Wacha\AppData\Roaming\npm\node_modules\phantomjs\bin\phantomjs

> phantomjs@2.1.7 install C:\Users\Wacha\AppData\Roaming\npm\node_modules\phantomjs
> node install.js

Considering PhantomJS found at C:\Users\Wacha\AppData\Roaming\npm\phantomjs.CMD
Looks like an npm install -g on windows; skipping installed version.
Downloading https://github.com/Medium/phantomjs/releases/download/v2.1.1/phantomjs-2.1.1-windows.zip
Saving to C:\Users\Wacha\AppData\Local\Temp\phantomjs\phantomjs-2.1.1-windows.zip
Receiving...
[#####] 97%
Received 17767K total.
Extracting zip contents
Removing C:\Users\Wacha\AppData\Roaming\npm\node_modules\phantomjs\lib\phantom
Copying extracted folder C:\Users\Wacha\AppData\Local\Temp\phantomjs\phantomjs-2.1.1-windows.zip-extract-149348893304\phantomjs-2.1.1-windows -> C:\Users\Wacha\AppData\Roaming\npm\node_modules\phantomjs\lib\phantom
Writing location.js file
Done. Phantomjs binary available at C:\Users\Wacha\AppData\Roaming\npm\node_modules\phantomjs\lib\phantom\bin\phantomjs.exe
phantomjs@2.1.7 C:\Users\Wacha\AppData\Roaming\npm\node_modules\phantomjs
├── progress@1.1.8
├── keccak@1.0
├── request-progress@2.0.1 (throttleit@1.0.0)
├── which1@2.14 (isexe@2.0.0)
├── hash@2.2.0 (is-stream@1.0, pinkie-promise@2.0.1)
├── extract-zip@1.5.0 (debug@0.7.4, yauz@2.4.1, concat-stream@1.5.0, mkdirp@0.5.0)
├── fs-extra@0.26.7 (path-is-absolute@1.0.1, klaw@1.3.1, graceful-fs@4.1.11, js-yaml@3.4.0, rimraf@2.6.1)
├── request@2.67.0 (tunnel-agent@0.4.3, aws-sign@2.0.6, oauth-sign@0.2, forever-agent@0.6.1, is-typedarray@1.0.0, caseless@0.11.0, stringstream@0.0.5, isstream@0.1.2, json-stringify-safe@0.1, extend@3.0.1, tough-cookie@2.2.2, qs@5.2.1, combined-stream@1.0.5, node-uuid@1.4.8, mime-types@2.1.15, b161@0.0, hawk@3.1.3, http-signature@1.1.1, har-validator@2.0.6, form-data@1.0.1)
C:\Users\Wacha\Desktop\MyProject>
  
```

รูปที่ ค.2 Install Phantomjs สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการทำงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) นำไฟล์ที่เขียนโปรแกรม (Script) ไฟล์ .js ซึ่งในโครงการสททเลือกใช้โปรแกรม Sublime

3) พิมพ์ command-line เพื่อเข้าถึงที่อยู่ของไฟล์ Script ที่จะทำการรัน

- ตัวอย่างอยู่ Desktop\MyProject พิมพ์ ดังนี้

- Desktop
- cd MyProject

```

Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Wacha\Desktop\MyProject>
  
```

รูปที่ ค.3 การเข้าถึงไฟล์ในโฟลเดอร์

4) การนำไฟล์โปรแกรม (Script) มาทดสอบ

- พิมพ์ phantomjs ตามด้วยชื่อไฟล์ (ตัวอย่างชื่อไฟล์ Test.js) พิมพ์ ดังนี้

- Phantomjs test.js

```

Command Prompt - Phantomjs 1ExportverSpace.js
C:\Users\Wacha\Desktop\MyProject>Phantomjs 1ExportverSpace.js
  
```

รูปที่ ค.4 การเข้าถึงไฟล์ในโฟลเดอร์

5) การนำไฟล์ที่ทดสอบนำออก (Export Output) เป็นไฟล์ .csv

- Phantomjs test.js > output.csv (create File)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Phantomjs test.js >> output.csv (Update File เดิม)

```

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Wacha\Desktop\MyProject>phantomjs Exportuer1.js >open.csv
C:\Users\Wacha\Desktop\MyProject>

```

รูปที่ ค.5 การนำไฟล์ออกมาเป็นไฟล์ .csv

- 6) แสดงไฟล์ .csv ที่ได้จากการทดสอบระบบ

| A | B | C | D | E |
|----|---------------------------------|---|------|---------------|
| 1 | ScenarioNo ScenarioName | Action | | status Images |
| 2 | 1 Bulk Update Space Information | | | |
| 3 | | Export>> | | |
| 4 | | ACTION 1/10: Open Page Update Information Export Space | Pass | view image |
| 5 | | ACTION 2/10: input Site No | Pass | view image |
| 6 | | ACTION 3/10: Preview Data UpdateSpace | Pass | view image |
| 7 | | ACTION 4/10: Check Preview Data Update | Pass | view image |
| 8 | | ACTION 5/10: Finished download | Pass | view image |
| 9 | | | | |
| 10 | | Import>> | | |
| 11 | | ACTION 6/10: Open Page Update Information Import Space | Pass | view image |
| 12 | | ACTION 7/10: Bulk file UpdateSpace | Pass | view image |
| 13 | | Preview Data Space Process.. | | |
| 14 | | ACTION 8/10: Preview Data UpdateSpace | Pass | view image |
| 15 | | Import Data Space Process.. | | |
| 16 | | ACTION 9/10: Import Data Complete | Pass | view image |
| 17 | | | | |
| 18 | | Checklist>> | | |
| 19 | | ACTION 10/10: Check Update LocationEN-"SANKHABURE" at SiteNo-"23289_COMET" Complete | Pass | view image |

รูปที่ ค.6 ไฟล์ .csv ที่ได้จากการทดสอบระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้