



รายงานสหกิจศึกษาฉบับสมบูรณ์

การพิสูจน์แนวคิดของระบบก่อนการขาย
Proof of Concept for Presale Approach

นางสาวมณฑนา พิมพ์รัตน์

ภาควิชาวิศวกรรมคอมพิวเตอร์ สาขาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560



รายงานสหกิจศึกษาฉบับสมบูรณ์

การพิสูจน์แนวคิดของระบบก่อนการขาย

Proof of Concept for Presale Approach

นางสาวมณฑนา พิมพิรัตน์

ภาควิชา วิศวกรรมคอมพิวเตอร์ สาขา วิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ชื่อโครงการสหกิจศึกษา การพิสูจน์แนวคิดของระบบก่อนการขาย

ชื่อ-สกุล นักศึกษา นางสาวมณฑนา พิมพิรัตน์

คณะ วิศวกรรมศาสตร์ ภาควิชา วิศวกรรมคอมพิวเตอร์ สาขาวิชา วิศวกรรมสารสนเทศ

ชื่อ-สกุล อาจารย์นิเทศ ผศ.ไพศาล สิทธิโยภาสกุล

ชื่อ-สกุล ผู้นิเทศงาน นายเอกชัย ร่มสิน

ชื่อสถานประกอบการ บริษัท อีฟวาแลนท์ จำกัด

บทคัดย่อ

ในการเสนอขายสินค้าด้านไอทีหรือการเสนอขายโปรแกรมขนาดใหญ่ จะมีกระบวนการ Proof of Concept เพื่อแสดงให้เห็นว่าโปรแกรมสามารถทำงานได้อย่างที่ลูกค้าต้องการ ดังนั้นโครงการนี้ จึงเป็นการทำเพื่อการพิสูจน์แนวคิด (Proof of Concept) ของระบบ โดยการสร้างระบบด้วย ONEWEB 4.0 แพลตฟอร์ม ร่วมกับการเขียนโปรแกรม (Customize) ด้วยภาษา HTML, Java, JavaScript, jQuery, Ajax, JSON โดยภายหลังการพัฒนาเรียบร้อยแล้ว ทำให้ได้ระบบที่มีประสิทธิภาพ และมีความสามารถในการทำงานที่ตรงกับความต้องการของลูกค้า และเพิ่มความมั่นใจให้ลูกค้าในการตัดสินใจซื้อระบบมากยิ่งขึ้น

คำสำคัญ : การพิสูจน์แนวคิด, การเขียนโปรแกรมเพิ่มเติม, ความต้องการของลูกค้า

Co-operative Title: Proof of Concept for Presale Approach
Student Intern Name: Ms.Mantana Pimpirut
Faculty: Engineering **Department:** Computer Engineering
Program: Information Engineering
Advisor Name: Asst.Prof.Paisan Sithiyopasakul
Mentor Name: Mr.Ekachai Rumsin
Company: Avalant Co., Ltd.

ABSTRACT

The product offering in technology and large program marketing needs to use Proof of Concept (POC) for being attractive in customer's decision firstly. Therefore, we have created the program which be able to present and prove the efficiency of our concept that will fulfill the requirements of our customers completely. This program, Proof of Concept (POC), is made by ONEWEB 4.0 Platform in combination with HTML, Java, JavaScript, jQuery, Ajax and JSON program. After developing the system perfectly, this program would meet the customer demand and increase confidence in customer's decision to purchasing.

Keywords : Proof of Concept, Customize, Customer demand

กิตติกรรมประกาศ

การศึกษาค้นคว้าในการทำโครงการครั้งนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์และช่วยเหลือจากบุคคลหลายๆท่าน โดยเฉพาะอย่างยิ่ง ผศ.ไพศาล สิทธิโยภาสกุล อาจารย์ที่ปรึกษา ที่คอยชี้แนะแนวทางการปฏิบัติสหกิจศึกษา ให้คำแนะนำและข้อเสนอแนะ จนทำให้โครงการนี้เสร็จสมบูรณ์

ขอขอบคุณ คุณเอกชัย รัมสิน พนักงานที่ปรึกษา รวมทั้งบุคลากรในบริษัท อีฟวาแลนท์ จำกัด ที่กรุณาเสียสละเวลาอันมีค่ามาให้ความรู้ คำแนะนำในการทำงาน และสนับสนุนการทำงานอย่างเต็มที่ ผู้จัดทำทราบซึ่งในความกรุณาอย่างยิ่ง

สุดท้ายนี้ ขอขอบคุณบิดา-มารดา และบุคคลรอบข้าง ที่คอยช่วยเหลือ สนับสนุน และเป็นกำลังใจให้เสมอมา

นางสาวมณฑนา พิมพิรัตน์

วิศวกรรมสารสนเทศ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง	3
2.1 ทฤษฎีซอฟต์แวร์คอมพิวเตอร์ที่เกี่ยวข้อง	3
2.1.1 ONEWEB 4.0	3
2.1.2 ระบบจัดการฐานข้อมูล (Database Management System: DBMS).....	11
2.2 ทฤษฎีภาษาคอมพิวเตอร์ที่เกี่ยวข้อง	12
2.2.1 ภาษา SQL (Structured Query Language).....	12
2.2.2 ภาษา Java (Java Programming Language)	19
2.2.3 ภาษา HTML (Hypertext Markup Language)	20
2.2.4 ภาษา JavaScript	21
2.2.5 ภาษา jQuery	21
2.2.6 Servlet.....	23

บทที่ 3 วิธีดำเนินงาน	25
3.1 แนวทางการออกแบบระบบ.....	25
3.1.1 แบบร่างหน้าจอ.....	25
3.2 ส่วนที่ต้องเขียนโปรแกรมเพิ่มเติม (Customize)	35
3.3 การดำเนินงาน.....	37
3.3.1 สร้างหน้า User Interface โดยใช้ ONEWEB 4.0.....	37
3.3.2 เขียน Reject popup	39
3.3.3 เขียนโค้ดให้บันทึกค่าสถานะ ‘Submit’ ลงในฐานข้อมูล เมื่อกดปุ่ม Submit.....	47
3.3.4 บันทึกเหตุผลในการ reject ลงในฐานข้อมูล.....	48
3.3.5 การ Redirect screen.....	51
3.3.6 Validate record.....	53
3.3.7 Validate field	55
3.3.8 Filter to show Address.....	56
3.3.9 Filter Search VAS ภายใต้ product id ใน package ที่เลือก.....	61
บทที่ 4 ผลการดำเนินงาน	64
4.1 ผลการเขียน Reject popup.....	64
4.1.1 แสดง Reject popup.....	64
4.1.2 บันทึกค่าสถานะ ‘Reject’ และเหตุผลในการ Reject ลงในฐานข้อมูล	65
4.2 บันทึกค่าสถานะ ‘Submit’	66
4.3 แจ้งเตือนให้ระบุ Address อย่างน้อย 1 record ในหน้า Input Install Address	67
4.4 แจ้งเตือนให้ระบุ Package อย่างน้อย 1 record ในหน้า Package Offering	67
4.5 Filter Search VAS ภายใต้ product id.....	68
4.6 แจ้งเตือนให้ Mapping Address ให้ครบทุก Package/Product.....	70
4.7 Filter to show Address.....	70
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ.....	72

5.1 สรุปผลการดำเนินงาน	72
5.2 ปัญหาและแนวทางแก้ไข	72
5.2.1 ปัญหา	72
5.2.2 แนวทางแก้ไข	72
5.3 ข้อเสนอแนะ.....	72
เอกสารอ้างอิง	73

สารบัญตาราง

	หน้า
ตารางที่ 2.1 สรุปคำสั่งในชุดคำสั่ง SELECT Statement.....	15
ตารางที่ 3.1 ตารางสรุปหน้าจอ (Screen) และ Entity ที่เกี่ยวข้อง.....	34
ตารางที่ 3.2 สรุปงานที่ต้องทำการ Customize.....	35

สารบัญภาพ

	หน้า
ภาพที่ 2.1 ส่วนการพัฒนาแอปพลิเคชัน.....	3
ภาพที่ 2.2 การปรับแต่ง User interface โดยการลากวางปุ่มและฟอร์มต่างๆ.....	4
ภาพที่ 2.3 ตัวอย่าง Entity ชื่อ Component	4
ภาพที่ 2.4 ตัวอย่างหน้า search.....	5
ภาพที่ 2.5 ตัวอย่างหน้า insert.....	5
ภาพที่ 2.6 ตัวอย่างหน้า update.....	6
ภาพที่ 2.7 เมนูที่ใช้ในการสร้างหน้า User interface	6
ภาพที่ 2.8 ประเภทของ Tab	7
ภาพที่ 2.9 ประเภทของ Module	7
ภาพที่ 2.10 ประเภทของ Button	8
ภาพที่ 2.11 ตัวอย่าง Entity Button และ Module Button.....	9
ภาพที่ 2.12 ประเภทของ Action.....	9
ภาพที่ 2.13 ประเภทของ Field.....	10
ภาพที่ 2.14 ตาราง project_customer	14
ภาพที่ 2.15 ผลลัพธ์จากการ query.....	15
ภาพที่ 2.16 ตาราง customer(1).....	16
ภาพที่ 2.17 ตาราง customer(1) ที่มีการเพิ่มข้อมูลทุกคอลัมน์	16
ภาพที่ 2.18 ตาราง customer(1) ที่มีการเพิ่มข้อมูลบางคอลัมน์	17
ภาพที่ 2.19 ตาราง customer(2).....	17
ภาพที่ 2.20 ตาราง customer(2) ที่มีการแก้ไขข้อมูล	18
ภาพที่ 2.21 ตาราง customer(3).....	18
ภาพที่ 2.22 ตาราง customer(3) ที่มีการลบข้อมูล.....	19
ภาพที่ 2.23 คลาส ซึ่งประกอบไปด้วยตัวแปร และเมธอดในการทำงาน	20
ภาพที่ 2.24 การทำงานของเว็บแอปพลิเคชันแบบดั้งเดิม	22
ภาพที่ 2.25 การทำงานของเว็บแอปพลิเคชันที่ใช้เทคนิค Ajax	23
ภาพที่ 2.26 หลักการทำงานของ JSP และ Servlet.....	24
ภาพที่ 3.1 แสดงลำดับการทำงานของหน้าจอต่างๆ (New Service Order Screen Workflow)...	25

สารบัญภาพ (ต่อ)

	หน้า
ภาพที่ 3.2 ServiceRequest หน้า search.....	25
ภาพที่ 3.3 ServiceRequest หน้า update	26
ภาพที่ 3.4 Reject Popup	26
ภาพที่ 3.5 Input Install Address.....	27
ภาพที่ 3.6 Package Offering.....	27
ภาพที่ 3.7 Add OTC.....	28
ภาพที่ 3.8 Search OTC.....	28
ภาพที่ 3.9 Product VAS	29
ภาพที่ 3.10 Add VAS	29
ภาพที่ 3.11 Search VAS	30
ภาพที่ 3.12 Mapping Product.....	31
ภาพที่ 3.13 Mapping Popup	31
ภาพที่ 3.14 Input Service ID	32
ภาพที่ 3.15 Service ID popup.....	32
ภาพที่ 3.16 Payment Profile.....	33
ภาพที่ 3.17 Create Quote	34
ภาพที่ 3.18 Create New Entity	37
ภาพที่ 3.19 Entity Configuration ตั้งชื่อ Entity	38
ภาพที่ 3.20 การลากวางปุ่มและฟอร์มต่างๆ เพื่อสร้างหน้า User Interface (1)	38
ภาพที่ 3.21 การลากวางปุ่มและฟอร์มต่างๆ เพื่อสร้างหน้า User Interface (2)	39
ภาพที่ 3.22 เขียน DAO interface.....	40
ภาพที่ 3.23 เขียนโค้ดเพื่อติดต่อกับฐานข้อมูลและการทำงานของเมธอด loadMSRejectReason()	40
ภาพที่ 3.24 ไฟล์ AjaxServletRejectPopup.java	42
ภาพที่ 3.25 ไฟล์ order_new_service_request.js	44
ภาพที่ 3.26 ทำการ configuration ปุ่ม Entity Action	45
ภาพที่ 3.27 Action Field Configuration	46

สารบัญภาพ (ต่อ)

	หน้า
ภาพที่ 3.28 Button Configuration	46
ภาพที่ 3.29 เพิ่ม function submitBtnAction() ในไฟล์ order_new_service_request.js	47
ภาพที่ 3.30 Button Configuration	47
ภาพที่ 3.31 เขียนเมธอด modifyResult() ในไฟล์ OrderNewServiceRequestSubmit.java	48
ภาพที่ 3.32 ตัวอย่างรูปแบบการแบบข้อมูลแบบ Vector	49
ภาพที่ 3.33 ทำการ configuration ปุ่ม Entity Action	50
ภาพที่ 3.34 Action Field Configuration	50
ภาพที่ 3.35 Function พื้นฐานของ Button.....	51
ภาพที่ 3.36 เมธอด dynamicURL() ใน Class OrderNewServiceRequestSubmit	51
ภาพที่ 3.37 เมธอด dynamicURL() ใน Class OrderNewRequestDetailSubmit	52
ภาพที่ 3.38 function loadForBack().....	53
ภาพที่ 3.39 เมธอด validateResult() เพื่อ Validate record	53
ภาพที่ 3.40 เมธอด validateResult() เพื่อ Validate record	55
ภาพที่ 3.41 เมธอด validateResult() เพื่อ Validate field	56
ภาพที่ 3.42 ประกาศเมธอดเพิ่มใน DAO interface	57
ภาพที่ 3.43 เขียนโค้ดเพื่อติดต่อกับฐานข้อมูลและการทำงานของเมธอด loadMSInstallAddress(String request_no)	58
ภาพที่ 3.44 ไฟล์ AjaxServletInstallAddress.java	59
ภาพที่ 3.45 function loadUpdateManyManual(moduleID,rowID).....	60
ภาพที่ 3.46 Product-VAS	61
ภาพที่ 3.47 Add VAS	62
ภาพที่ 3.48 Search VAS	62
ภาพที่ 3.49 ไฟล์ OrderNewVASSearch.java	63
ภาพที่ 4.1 แสดง popup เมื่อกดปุ่ม Reject	65
ภาพที่ 4.2 บันทึกค่า Reject และเหตุผลในการ Reject ลงในฐานข้อมูล	66
ภาพที่ 4.3 บันทึกค่า Submit	66
ภาพที่ 4.4 แจ้งเตือนให้ระบุ Address อย่างน้อย 1 record	67

สารบัญภาพ (ต่อ)

	หน้า
ภาพที่ 4.5 แจ้งเตือนให้ระบุ Package อย่างน้อย 1 record	68
ภาพที่ 4.6 หน้า Product-VAS.....	68
ภาพที่ 4.7 หน้า Add VAS.....	69
ภาพที่ 4.8 หน้า Search VAS ที่ filter ตาม PRODUCT_ID.....	69
ภาพที่ 4.9 แจ้งเตือนให้ระบุ Address ให้ครบทุก Package/Product.....	70
ภาพที่ 4.10 แสดง Address ทั้งหมด	70
ภาพที่ 4.11 แสดงเฉพาะ Address ที่ยังไม่ถูกเลือก	71

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

บริษัท อีฟวาแลนท์ จำกัด เป็นผู้ประกอบการด้านเทคโนโลยีสารสนเทศ ที่ให้คำปรึกษาและพัฒนาระบบซอฟต์แวร์ แอปพลิเคชันให้กับหน่วยงานราชการและบริษัทเอกชนต่างๆ และเป็นผู้นำในการพัฒนาซอฟต์แวร์แพลตฟอร์ม (Software Platform) จนได้รับรางวัลต่างๆ ล่าสุดได้รับรางวัลชนะเลิศ Thailand ICT Awards 2013 (TICTA Award 2013) ใน 2 สาขา คือ Financial Industry Applications และ Application and Infrastructure Tools จากสำนักงานส่งเสริมอุตสาหกรรมซอฟต์แวร์แห่งชาติ (องค์การมหาชน) และบริษัทฯ ยังเป็น Premium Level Partner ของ IBM Thailand Software Group อีกทั้งยังเป็นบริษัทของคนไทยแห่งแรกในประเทศไทยที่ได้รับมาตรฐานการบริหารจัดการซอฟต์แวร์ CMMI Level 5 [1]

การเสนอขายสินค้าด้านไอที หรือโปรแกรมขนาดใหญ่ มักจะต้องมีกระบวนการ Proof of Concept (POC) ก่อนเสมอ คือต้องทำให้เห็นว่าโปรแกรมที่จะขายให้ นั้น เข้ากันได้กับระบบลูกค้า และทำงานได้อย่างที่ลูกค้าต้องการ เพื่อให้ลูกค้าสามารถทำการตัดสินใจได้อย่างถูกต้อง และไม่มีปัญหาตามมา สืบเนื่องด้วยข้อจำกัดต่างๆ ของสินค้าและโปรแกรมนั้นๆ คนที่ทำ POC ก็จะเป็นทีมงานจากผู้ขายสินค้า และเป็นกลุ่มคนที่มีความเข้าใจและสามารถใช้งานอุปกรณ์นั้นๆ ได้อย่างมีความเชี่ยวชาญ

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อทำการพิสูจน์แนวคิด (Proof of Concept) ของระบบว่าสามารถทำงานได้อย่างถูกต้อง และตรงความต้องการของลูกค้าก่อนการเสนอขาย

1.2.2 เพื่อเพิ่มประสิทธิภาพให้ระบบที่สร้างด้วยแพลตฟอร์ม ONEWEB 4.0 ให้สามารถทำงานตรงกับตามความต้องการของลูกค้ามากที่สุด

1.3 ขอบเขตของโครงการ

1.3.1 การปรับแต่งโปรแกรมเพิ่มเติม (Customize) ในสิ่งที่การ Configuration พื้นฐานของแพลตฟอร์ม ONEWEB 4.0 ทำไม่ได้ โดยการเขียนภาษา HTML, Java, JavaScript, jQuery, Ajax, JSON โดยใช้ซอฟต์แวร์ Eclipse

1.3.2 การเขียนคำสั่ง SQL เพื่อให้โปรแกรมสามารถรับ-ส่งข้อมูลกับฐานข้อมูลแบบ PostgreSQL ได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ช่วยให้สามารถเสนอขายระบบที่ตรงกับความต้องการของลูกค้า

1.4.2 เพื่อให้ได้ความเชื่อมั่นว่า สินค้า หรือโปรแกรมที่ลูกค้าจะซื้อนั้น สามารถทำงานได้อย่างมีประสิทธิภาพ และเหมาะสมกับองค์กรมากที่สุด

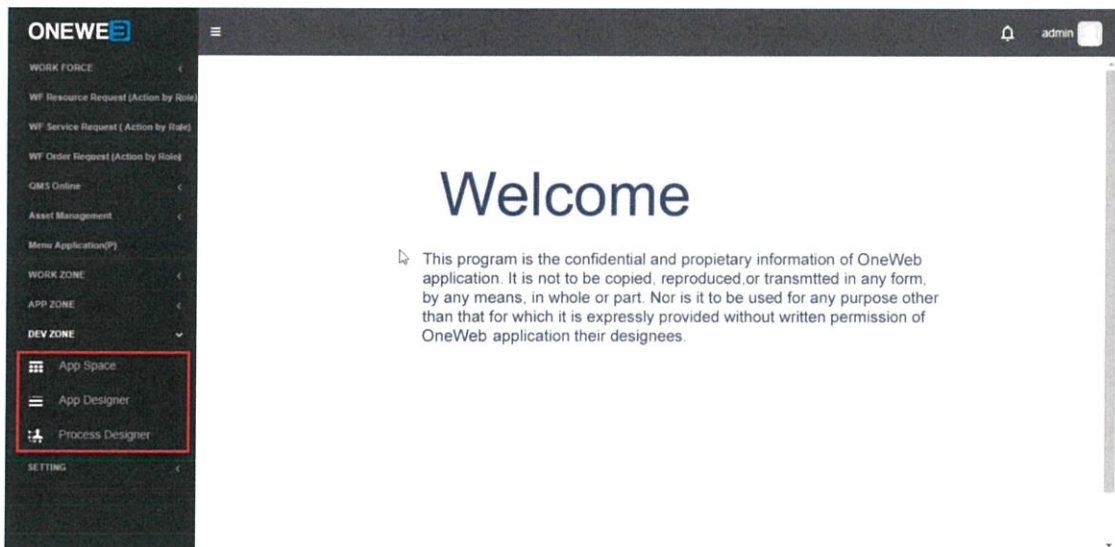
บทที่ 2

ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

2.1 ทฤษฎีซอฟต์แวร์คอมพิวเตอร์ที่เกี่ยวข้อง

2.1.1 ONEWEB 4.0

ONEWEB 4.0 เป็นแพลตฟอร์มที่พัฒนาขึ้นด้วยบริษัท อีฟวาแลนท์ จำกัด โดยเป็น All-In-One Digital Platform สามารถปฏิรูปความคิดของคนในองค์กรให้ออกมาเป็นแอปพลิเคชัน และสามารถตอบโจทย์ธุรกิจ การใช้งานของคนในองค์กรได้อย่างครอบคลุม มั่นคง และยั่งยืนในยุคดิจิทัล ทรานส์ฟอร์มเมชัน จากการพัฒนาผสมผสานกับประสบการณ์อันยาวนานของบริษัทฯ ทำให้ ONEWEB 4.0 รองรับการเติบโตขององค์กรทุกขนาดทุกอุตสาหกรรมให้สามารถออกแบบและพัฒนา แอปพลิเคชันธุรกิจขององค์กรเพื่อใช้ในการแข่งขันทำธุรกิจอย่างรวดเร็วและมีประสิทธิภาพ อีกทั้งยังสามารถประหยัดค่าใช้จ่ายในการบำรุงรักษาและการปรับปรุงระบบในระยะยาวอีกด้วย

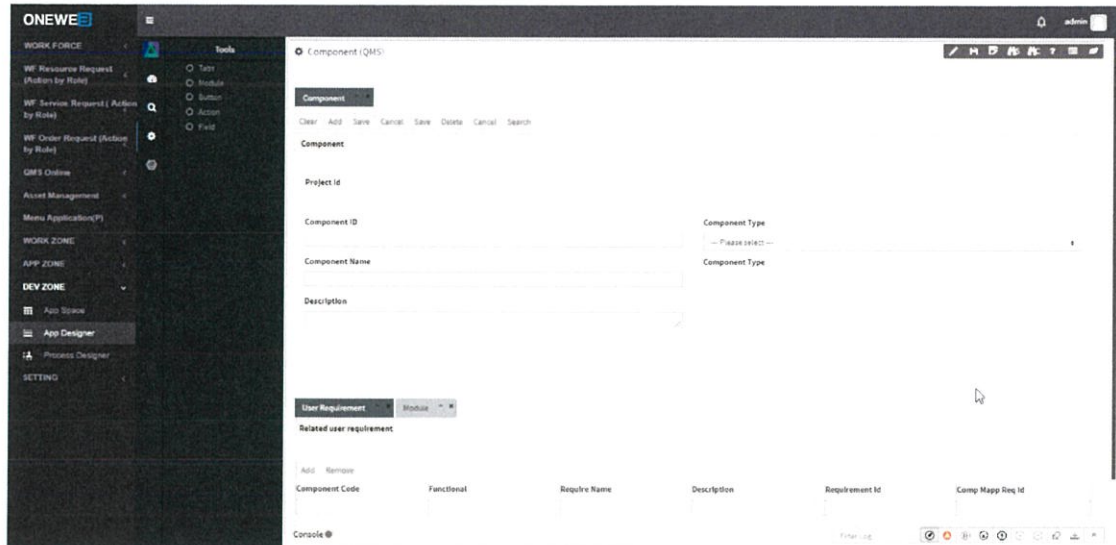


ภาพที่ 2.1 ส่วนการพัฒนาแอปพลิเคชัน

หน้าที่การพัฒนาแอปพลิเคชันแบ่งเป็น 3 ส่วน คือ ส่วนของ App Designer จะดูแลเรื่องฐานข้อมูล และการปรับแต่ง User interface (UI) และในส่วนของกระบวนการออกแบบ Process Designer คือ เครื่องมือในการออกแบบกระบวนการของธุรกิจ ส่วนสุดท้าย คือ App Space เป็นโมดูลในการบริหารจัดการแอปพลิเคชันที่พัฒนาขึ้นมาในองค์กรอย่างบูรณาการ โดยทั้ง 3 ส่วนเป็นกระบวนการพัฒนาแอปพลิเคชันที่จะช่วยลดความยุ่งยากและเวลาในการจัดทำเอกสารของ

โครงการในการพัฒนาแอปพลิเคชันทั้งไป โดยเน้นผลลัพธ์ที่สามารถพัฒนาแอปพลิเคชันให้ตรงตามความต้องการของธุรกิจได้อย่างรวดเร็ว

โดยในส่วนของ App Designer ที่ใช้ในการสร้าง Web Application แบบ Responsive นั้น สามารถสร้างได้ด้วยการลากวางปุ่มและฟอร์มต่างๆ ได้ตามต้องการ



ภาพที่ 2.2 การปรับแต่ง User interface โดยการลากวางปุ่มและฟอร์มต่างๆ

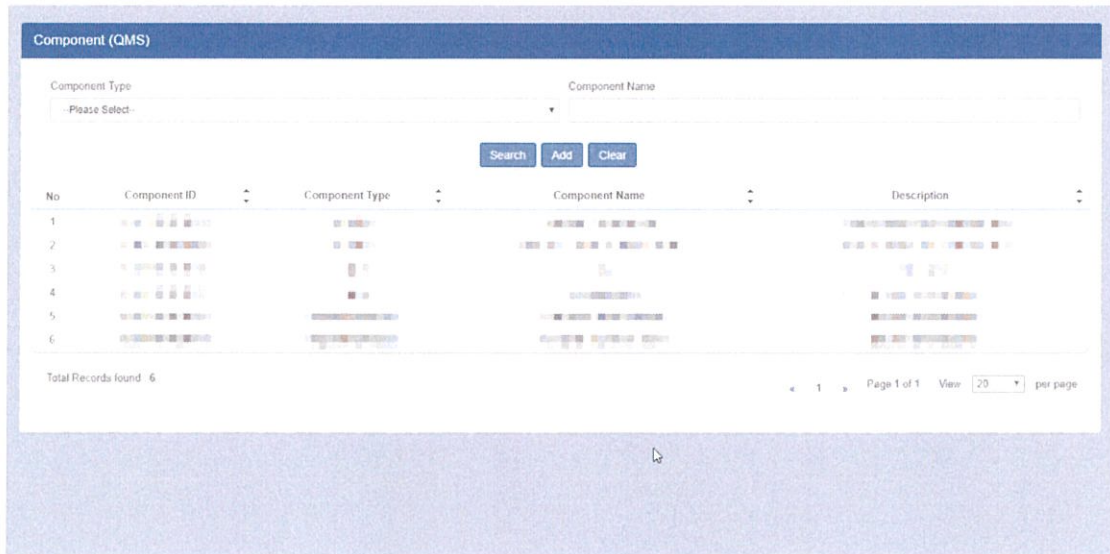
2.2.1.1 Entity

Entity คือ กลุ่มของข้อมูลทางธุรกิจและธุรกรรมที่เกี่ยวข้องในกระบวนการทางธุรกิจ (Business Processes) โดยการสร้าง entity จะมีองค์ประกอบทั่วไปดังภาพที่ 2.3

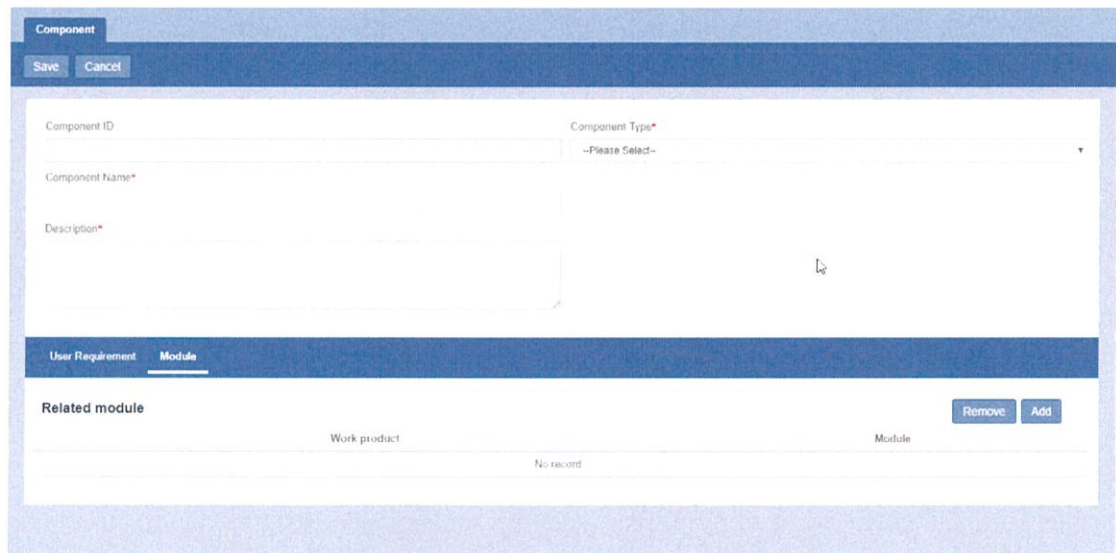


ภาพที่ 2.3 ตัวอย่าง Entity ชื่อ Component

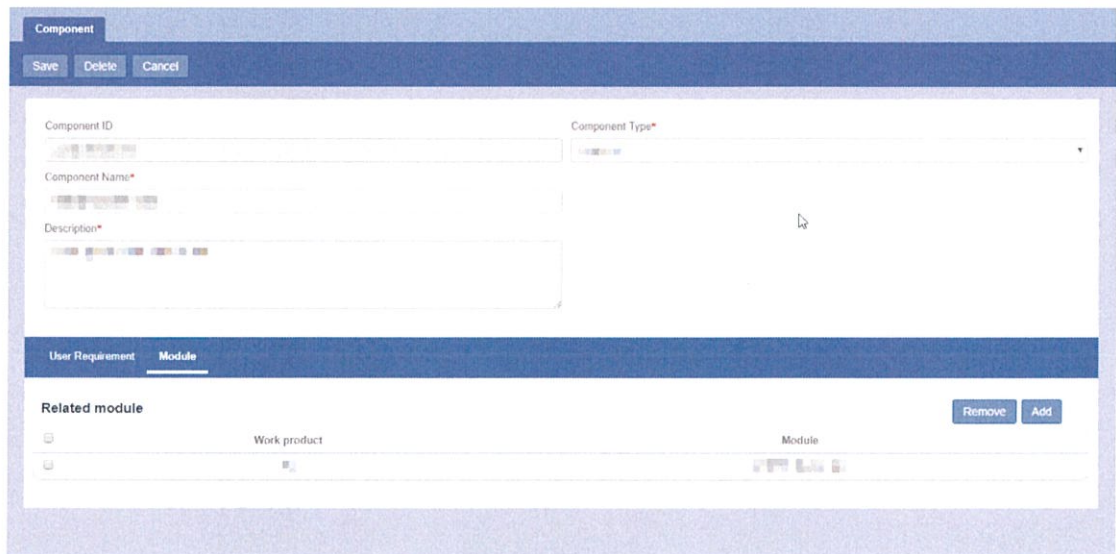
โดยการสร้างหนึ่ง entity จะได้หน้า User interface ทั้งหมด 3 หน้า
ด้วยกัน คือ หน้า search, หน้า insert และ หน้า update



ภาพที่ 2.4 ตัวอย่างหน้า search



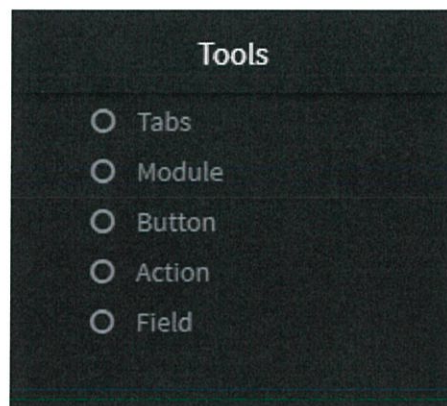
ภาพที่ 2.5 ตัวอย่างหน้า insert



ภาพที่ 2.6 ตัวอย่างหน้า update

2.2.1.2 Menu

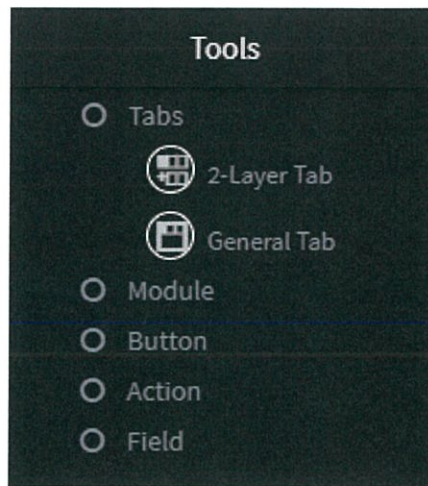
เมนูที่ใช้ในการสร้างหน้า User interface มีทั้งหมด 5 เมนู ได้แก่ เมนู Tabs, Module, Button, Action และ Field



ภาพที่ 2.7 เมนูที่ใช้ในการสร้างหน้า User interface

2.2.1.3 Tab

Tab ที่ใช้ในการสร้างหน้า User interface แบ่งออกเป็น 2 ประเภท ได้แก่ 2-Layer Tab และ General Tab

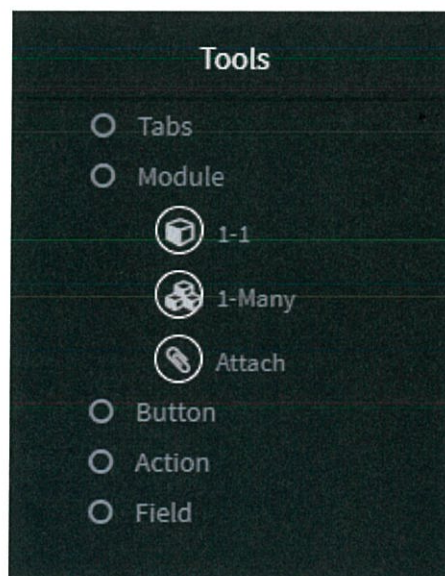


ภาพที่ 2.8 ประเภทของ Tab

- 2-Layer Tab: สำหรับหน้าจอที่มีการออกแบบไว้ 2 Layer โดยมี Tab บนและ Tab ล่าง ซึ่ง Tab ประเภทนี้จะใช้สำหรับ Tab Layer บนเท่านั้น (หากต้องการวาง Tab ล่างจะใช้ General Tab)
- General Tab: สำหรับหน้าจอที่มีการออกแบบไว้ Layer เดียว โดยมี Tab บน Tab เดียวเท่านั้น

2.2.1.4 Module

Module หรือ ส่วนการทำงานเฉพาะ เป็นส่วนประกอบของระบบที่ใช้ในการเชื่อมต่อกับระบบอื่น โดยตัวโมดูลจะมีการออกแบบและควบคุมตัดแปลงภายในตัวโมดูลเอง โดยจะแบ่งเป็น 3 ประเภท ได้แก่ 1-1, 1-Many และ Attach

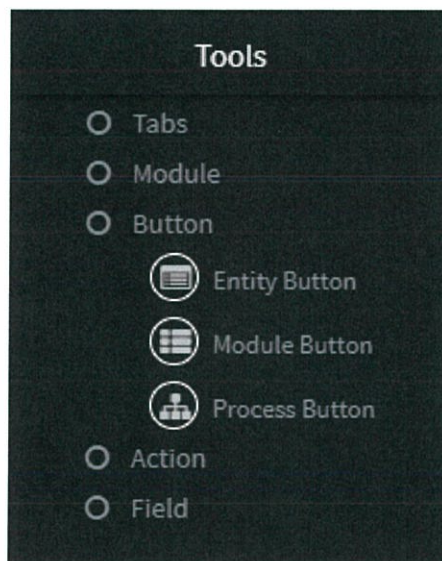


ภาพที่ 2.9 ประเภทของ Module

- 1-1: สำหรับ Module ที่มีลักษณะการบันทึกข้อมูลเพียง Record เดียวลง Database
- 1-Many: สำหรับ Module ที่มีลักษณะการบันทึกข้อมูล Multi Record ลง Database
- Attach: สำหรับ Module ที่ใช้แนบเอกสาร

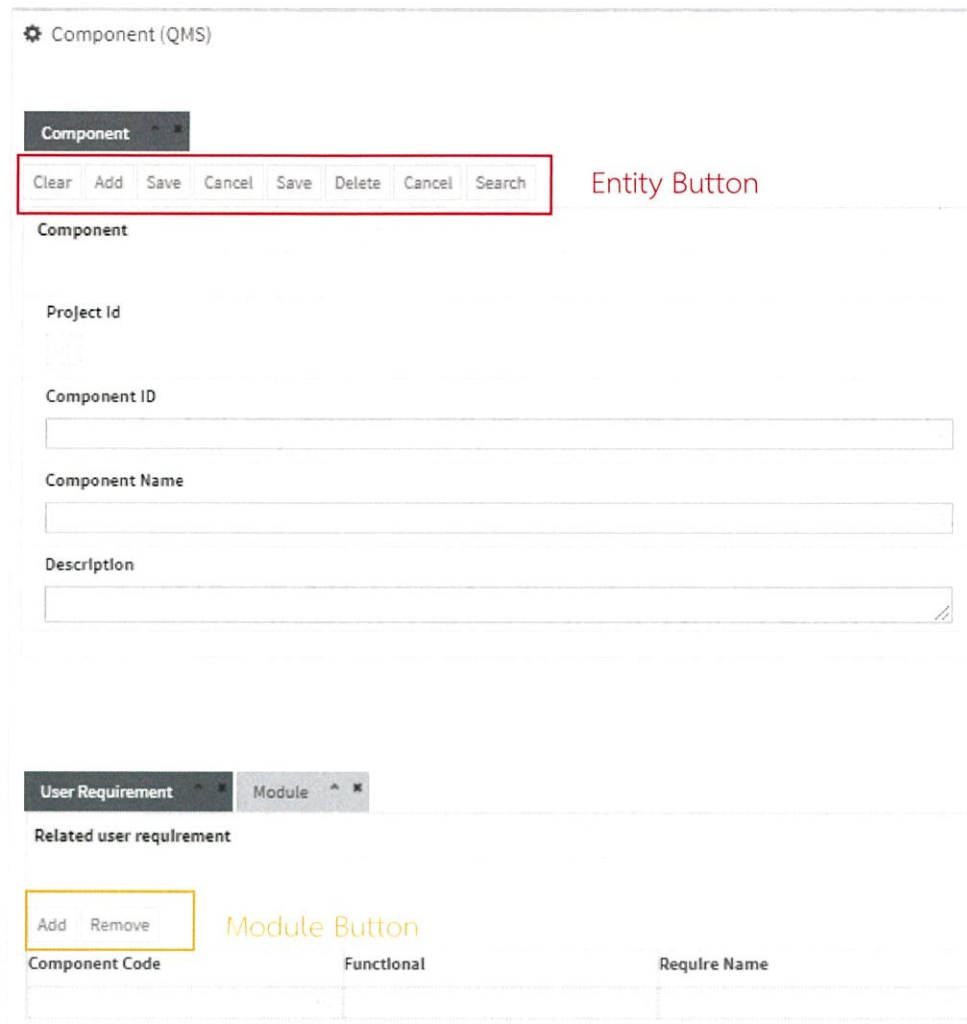
2.2.1.5 Button

Button หรือปุ่มที่จะทำให้เกิดการทำงานหรือเกิด Action ขึ้นนั้น มี 3 ประเภท แต่ที่ใช้เป็นหลักมี 2 ประเภท ได้แก่ Entity Button และ Module Button



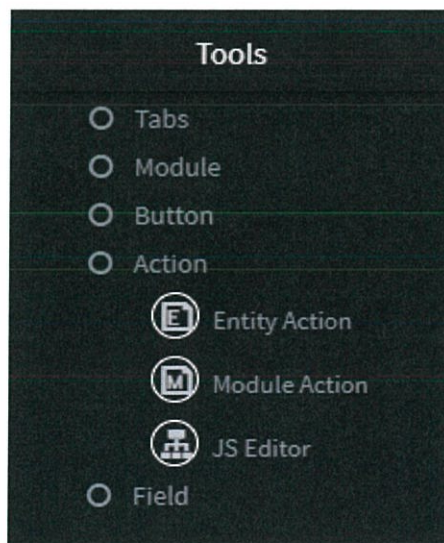
ภาพที่ 2.10 ประเภทของ Button

- Entity Button: เป็นปุ่มที่อยู่ในส่วนของ Entity ใช้สำหรับ Function ที่จะมี Action กระทบกับทั้งหน้า Entity
- Module Button: เป็นปุ่มที่อยู่ในส่วนของ Module 1-Many เท่านั้น ใช้สำหรับ Function ที่จะมี Action กระทบกับ Module 1-Many ที่วางปุ่มไว้



ภาพที่ 2.11 ตัวอย่าง Entity Button และ Module Button

2.2.1.6 Action

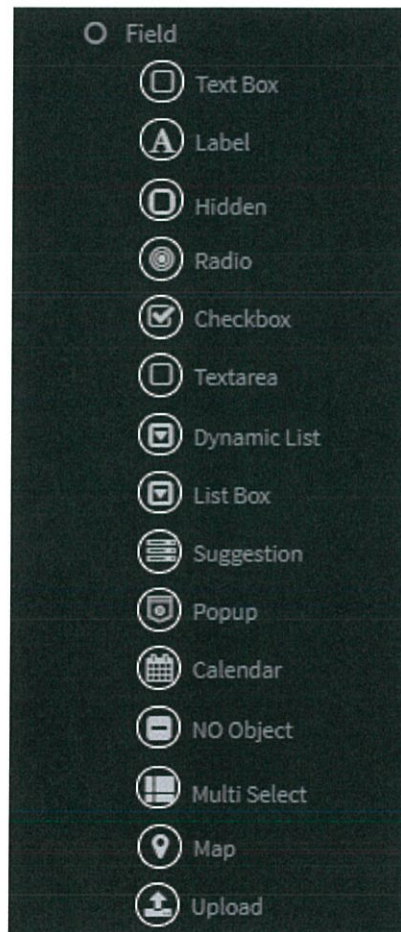


ภาพที่ 2.12 ประเภทของ Action

Action จะเป็นส่วนที่เราเอาไฟล์โค้ดต่างๆ ที่เราเขียนเพื่อปรับแต่งการทำงานของโปรแกรมเพิ่มเติม(Customize) มาใส่ไว้

- Entity Action: เมื่อการทำงานนั้นอยู่ในระดับ Entity
- Module Action: เมื่อการทำงานนั้นอยู่ในระดับ Module หรือเป็นการทำงานงานเฉพาะส่วน

2.2.1.7 Field



ภาพที่ 2.13 ประเภทของ Field

- Text box: Field ที่ใช้สำหรับกรอกข้อมูล รับผิดชอบทั้งตัวเลขและตัวอักษร
- Label: Field ที่ใช้สำหรับระบุป้ายชื่อหรือข้อความ เพื่อสร้างความเพิ่มเติมบนฟอร์มให้มีความสมบูรณ์มากขึ้น
- Hidden: Field ที่ใช้สำหรับบันทึกข้อมูลลง Database แต่ไม่ต้องการแสดงที่หน้าจอ (เหมาะสำหรับ Key ของ Module)

- Radio: Field ที่ใช้สำหรับให้เลือกข้อมูล ซึ่งมีลักษณะหลายตัวเลือก แต่ให้เลือกเพียงแค่ตัวเลือกเดียว (Choice)
- Checkbox: Field ที่ใช้สำหรับให้เลือกข้อมูลซึ่งมีลักษณะหลายตัวเลือก และ 1 ครั้งสามารถเลือกได้หลายตัวเลือก
- Textarea: Field ที่ใช้สำหรับกรอกข้อมูลที่มีขนาดเยอะๆ ประมาณ 100 ตัวอักษรขึ้นไป รับผิดชอบทั้งตัวเลขและตัวอักษร
- Dynamic List (Dropdown List): Field ที่ใช้สำหรับแสดงข้อมูลให้เลือกเป็น Dropdown โดยข้อมูลใน Dropdown จะนำมาจาก Table (ที่อยู่ในฐานข้อมูล) ที่เป็น Master Setup
- List Box: Field ที่ใช้สำหรับแสดงข้อมูลให้เลือกเป็น Dropdown โดยข้อมูลใน Dropdown จะมาจากการกำหนดค่าเองในตอน Configuration ได้
- Suggestion Box: Field ที่ใช้สำหรับแสดงข้อมูลให้เลือกเป็น Dropdown และสามารถ Key คำค้นหาได้ที่ Textbox
- Popup: Field ที่ใช้สำหรับแสดงข้อมูลให้เลือกเป็น Popup โดยข้อมูลใน Popup จะนำมาจาก Table (ที่อยู่ในฐานข้อมูล) ที่เป็น Master Setup และสามารถค้นได้ที่ Textbox และค้นหาได้จาก Popup
- Calendar: Field ที่ใช้สำหรับเก็บวันที่
- No Object: Field ที่ใช้สำหรับแสดงข้อมูลที่ Module 1-Many และเป็น Field ไม่มีอยู่ใน Table ของ Module นี้ แต่จะเกิดจากการเขียน SQL Command Join แล้วดึงขึ้นมาแสดง (Field นี้จะไม่ Save ลง Database)
- Multi Select: Field ที่ใช้สำหรับเลือกข้อมูลได้ครั้งละหลายๆอันในครั้งเดียว

2.1.2 ระบบจัดการฐานข้อมูล (Database Management System: DBMS) [2]

ระบบจัดการฐานข้อมูล คือซอฟต์แวร์สำหรับบริหารจัดการฐานข้อมูล ซึ่งมีหน้าที่ช่วยให้ผู้ใช้สามารถเข้าถึงข้อมูลได้ง่ายและมีประสิทธิภาพ โดยที่ผู้ใช้ไม่จำเป็นต้องรู้เกี่ยวกับรายละเอียดโครงสร้างของฐานข้อมูล เปรียบเสมือนเป็นสื่อกลางระหว่างผู้ใช้และโปรแกรมต่างๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล

หน้าที่ของระบบการจัดการฐานข้อมูล

- แปลงคำสั่งที่ใช้จัดการกับข้อมูลภายในฐานข้อมูล ให้อยู่ในรูปแบบที่ฐานข้อมูลเข้าใจ
- ป้องกันความเสียหายที่จะเกิดขึ้นกับข้อมูลภายในฐานข้อมูล โดยจะมีการตรวจสอบว่าคำสั่งไหนสามารถใช้ได้ และคำสั่งไหนไม่สามารถใช้ได้
- อำนวยความสะดวกให้ผู้ที่ต้องการใช้ฐานข้อมูล ทำให้ง่ายต่อการปรับปรุงแก้ไขข้อมูล
- รักษาความสัมพันธ์ของข้อมูลภายในฐานข้อมูลให้มีความถูกต้องอยู่เสมอ
- เก็บรายละเอียดต่างๆ ที่เกี่ยวข้องกับข้อมูลภายในฐานข้อมูล เรียกว่า Meta Data ซึ่งหมายถึง "ข้อมูลของข้อมูล"
- มีการกำหนดสิทธิ์การเข้าใช้ข้อมูล เพื่อความปลอดภัยของข้อมูล โดยป้องกันไม่ให้ผู้ที่ไม่ได้รับอนุญาตเข้ามาเรียกใช้หรือแก้ไขข้อมูลในส่วนป้องกันเอาไว้
- ช่วยจัดการการเข้าใช้ข้อมูลร่วมกันในเวลาเดียวกัน (สภาพที่มีผู้ใช้พร้อมๆ กันหลายคน)
- DBMS ทำให้โปรแกรมเป็นอิสระจากระบบปฏิบัติการและฮาร์ดแวร์

2.2 ทฤษฎีภาษาคอมพิวเตอร์ที่เกี่ยวข้อง

2.2.1 ภาษา SQL (Structured Query Language) [3]

SQL ย่อมาจาก Structured Query Language เป็นภาษาที่ใช้ในการจัดการกับฐานข้อมูลโดยเฉพาะ ภาษา SQL มีโครงสร้างที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง แบ่งออกเป็น 3 ส่วน คือ

- ภาษาที่ใช้สำหรับนิยามข้อมูล (Data Definition Language: DDL)
- ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language: DML)
- ภาษาควบคุมข้อมูล (Data Control Language: DCL)

2.2.1.1 ภาษาที่ใช้สำหรับนิยามข้อมูล (Data Definition Language: DDL)

Data Definition Language (DDL) [4] เป็นภาษาที่ใช้กำหนดโครงสร้างข้อมูล เพื่อเปลี่ยนแปลง หรือยกเลิกโครงสร้างฐานข้อมูลตามที่ต้องการ ออกแบบไว้ โครงสร้างดังกล่าวเรียกว่า Schema ตัวอย่างเช่น การกำหนดให้ฐานข้อมูลประกอบด้วยตารางอะไรบ้าง ชื่ออะไร เป็นประเภทใด ซึ่งภาษา DDL ประกอบด้วย 3 คำสั่ง ได้แก่ ได้แก่ CREATE, ALTER และ DROP

1) คำสั่งการสร้าง (CREATE) เป็นคำสั่งที่ใช้ในการสร้างโครงสร้างของตาราง โดยมีรูปแบบดังนี้

```
CREATE TABLE <ชื่อตาราง>
(ชื่อคอลัมน์1 ประเภทข้อมูล1,
ชื่อคอลัมน์2 ประเภทข้อมูล2,
...);
```

เช่น

```
CREATE TABLE project
(project_id int(10) NOT NULL,
project_code varchar(10),
project_name varchar(20)
);
```

2) คำสั่งเปลี่ยนแปลงโครงสร้าง (ALTER) เป็นคำสั่งที่ใช้ในการเปลี่ยนแปลงโครงสร้างตาราง โดยมีรูปแบบดังนี้

```
ALTER TABLE <ชื่อตาราง>
<คำสั่งการเปลี่ยนแปลง> (<ชื่อคอลัมน์ ประเภทข้อมูล>);
```

เช่น

- การเพิ่มคอลัมน์

```
ALTER TABLE project
ADD (partner_code varchar(10));
```

- การเปลี่ยนแปลงขนาดความกว้างของคอลัมน์

```
ALTER TABLE project
MODIFY (project_name varchar(50));
```

- การเปลี่ยนชื่อคอลัมน์

```
ALTER TABLE project
RENAME (project_id to project_no);
```

- การลบคอลัมน์ออกจากตาราง

```
ALTER TABLE project
DROP COLUMN (partner_code);
```

3) คำสั่งยกเลิก (DROP) เป็นคำสั่งที่ใช้ในการลบตารางออกจากฐานข้อมูล โดยมีรูปแบบดังนี้

```
DROP TABLE <ชื่อตาราง>
```

เช่น

```
DROP TABLE project;
```

2.2.1.2 ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language: DML)

Data Manipulation Language (DML) เป็นภาษาที่ใช้จัดการข้อมูลภายในตารางในฐานข้อมูล เช่น คำสั่งในการป้อนข้อมูลลงในฐานข้อมูลและเปลี่ยนแปลงข้อมูลในฐานข้อมูล ซึ่งภาษา DML ประกอบด้วยชุดคำสั่ง 4 คำสั่ง คือ

1) คำสั่งค้นหาข้อมูล (SELECT) เป็นการเรียกหาข้อมูลจากฐานข้อมูลตามเงื่อนไขที่ระบุ ประโยคคำสั่ง SELECT มี 5 clause ให้เลือกใช้ แต่มีเฉพาะ FROM เป็น clause บังคับ โดยมีรูปแบบดังนี้ [5]

```
SELECT <ชื่อคอลัมน์ที่ต้องการดูข้อมูล>
```

```
FROM <ชื่อตาราง>
```

```
[WHERE <เงื่อนไขตามที่ระบุ>]
```

```
[GROUP BY <ชื่อคอลัมน์1, ชื่อคอลัมน์2, ...>]
```

```
[HAVING <เงื่อนไขตามที่ระบุ>]
```

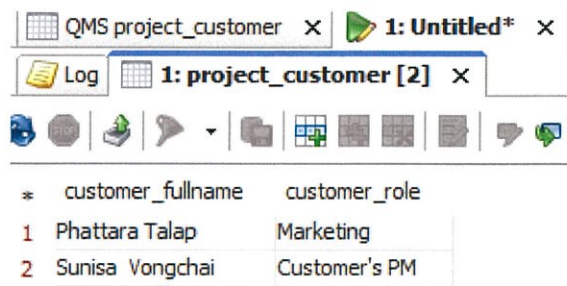
```
[ORDER BY <ชื่อคอลัมน์1, ชื่อคอลัมน์2, ...> [ASC | DESC]];
```

เช่น

project_cust_id	project_id	customer_fullname	customer_role	customer_email	customer_telephone
1	61	201 Somboon Phochana	Customer's BA	somboon@gmail.com	0809999955
2	62	205 Phattara Talap	Marketing	phat.ta@gmail.com	0818003000
3	63	205 Sunisa Vongchai	Customer's PM	sunisa@gmail.com	0904444440
4	101	0 Nithi Paisankul	User	nithi@mail.com	0900000000

ภาพที่ 2.14 ตาราง project_customer

```
SELECT customer_fullname, customer_role  
FROM project_customer  
WHERE project_id = 205;
```



ภาพที่ 2.15 ผลลัพธ์จากการ query

ตารางที่ 2.1 สรุปคำสั่งในชุดคำสั่ง SELECT Statement

คำสั่ง	ความหมาย
SELECT	เป็นคำสั่งเรียกดูข้อมูลในคอลัมน์ที่ระบุ ซึ่งอาจจะมากกว่าหนึ่งคอลัมน์ก็ได้ หรือถ้าต้องการเรียกดูข้อมูลทุกคอลัมน์ สามารถใช้เครื่องหมายดอกจัน (*) แทนการเขียนชื่อคอลัมน์ทั้งหมด
FROM	เป็นการระบุตารางที่ต้องการดู ซึ่งอาจจะมากกว่าหนึ่งตารางก็ได้
WHERE	ระบุเงื่อนไขที่จะใช้ในการค้นหาข้อมูลจากตารางที่อยู่หลังคำสั่ง FROM
GROUP BY	ใช้ในการจัดกลุ่มภายในคอลัมน์
HAVING	เป็นการระบุเงื่อนไขคล้ายกับ WHERE เพียงแต่จะถูกใช้ร่วมกับ GROUP BY แทน
ORDER BY	เป็นการกำหนดให้ผลลัพธ์ของการ query แสดงแบบเรียงลำดับ ตามคอลัมน์ที่ระบุให้เรียงลำดับ ASC: เรียงลำดับจากน้อยไปมาก (เป็นค่าเริ่มต้น) DESC: เรียงลำดับจากมากไปน้อย

2) คำสั่งเติมข้อมูล (INSERT) เป็นคำสั่งเพื่อเพิ่มข้อมูลเข้าไปในตาราง โดยมีรูปแบบดังนี้

INSERT INTO <ชื่อตาราง>

VALUES (< ค่าของคอลัมน์ 1>, < ค่าของคอลัมน์ 2>, ...);

เช่น

ลูกค้า	ชื่อลูกค้า	ชื่อผู้ติดต่อ	ที่อยู่	เมือง	รหัสไปรษณีย์	ประเทศ
87	Wartian Herkku	Pirkko Koskitalo	Torikatu 38	Oulu	90110	Finland
88	Wellington Importadora	Paula Parente	Rua do Mercado, 12	Resende	08737-363	Brazil
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland

ภาพที่ 2.16 ตาราง customer(1) [6]

การเพิ่มข้อมูลโดยไม่กำหนดคอลัมน์ นั่นคือ เวลาเพิ่มข้อมูล ต้องเพิ่มให้ครบทุกคอลัมน์

```
INSERT INTO customer
```

```
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');
```

ลูกค้า	ชื่อลูกค้า	ชื่อผู้ติดต่อ	ที่อยู่	เมือง	รหัสไปรษณีย์	ประเทศ
87	Wartian Herkku	Pirkko Koskitalo	Torikatu 38	Oulu	90110	Finland
88	Wellington Importadora	Paula Parente	Rua do Mercado, 12	Resende	08737-363	Brazil
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland
92	Cardinal	Tom B. Erichsen	Skagen 21	Stavanger	4006	Norway

ภาพที่ 2.17 ตาราง customer(1) ที่มีการเพิ่มข้อมูลทุกคอลัมน์ [6]

การเพิ่มข้อมูลแบบกำหนดชื่อคอลัมน์ โดยเวลาเพิ่มต้องบอกชื่อคอลัมน์ที่ต้องการเพิ่ม ไม่จำเป็นต้องเพิ่มทุกคอลัมน์ ดัง Syntax ด้านล่าง

```
INSERT INTO customer (ชื่อลูกค้า, เมือง, ประเทศ)
```

```
VALUES ('Cardinal', 'Stavanger', 'Norway');
```

ลูกค้า	ชื่อลูกค้า	ชื่อผู้ติดต่อ	ที่อยู่	เมือง	รหัสไปรษณีย์	ประเทศ
87	Wartian Herkku	Pirkko Koskitalo	Torikatu 38	Oulu	90110	Finland
88	Wellington Importadora	Paula Parente	Rua do Mercado, 12	Resende	08737-363	Brazil
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland
92	Cardinal	null	null	Stavanger	null	Norway

ภาพที่ 2.18 ตาราง customer(1) ที่มีการเพิ่มข้อมูลบางคอลัมน์ [6]

3) คำสั่งแก้ไขข้อมูล (UPDATE) เป็นคำสั่งเพื่อปรับปรุงแก้ไขข้อมูลที่เคยบันทึกไว้แล้วในตาราง โดยมีรูปแบบดังนี้

UPDATE <ชื่อตาราง>

SET <คอลัมน์ 1 = ค่าของคอลัมน์ 1, คอลัมน์ 2 = ค่าของคอลัมน์ 2, ...>

WHERE <เงื่อนไข>;

เช่น

ลูกค้า	ชื่อลูกค้า	ชื่อผู้ติดต่อ	ที่อยู่	เมือง	รหัสไปรษณีย์	ประเทศ
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitucion 2222	Mexico D.F.	05021	Mexico
3	Antonio Moreno Taqueria	Antonio Moreno	Mataderos 2312	Mexico D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbkop	Christina Berglund	Berguvsvagen 8	Lulea	S-958 22	Sweden

ภาพที่ 2.19 ตาราง customer(2) [7]

หากต้องการแก้ไขข้อมูลของลูกค้าที่ชื่อ Alfreds Futterkiste โดยปรับปรุงข้อมูล ผู้ติดต่อ และ เมือง ใหม่ สามารถเขียนได้ดังนี้

UPDATE customer

SET ชื่อผู้ติดต่อ='Alfred Schmidt', เมือง='Hamburg'

WHERE ชื่อลูกค้า='Alfreds Futterkiste';

ลูกค้า	ชื่อลูกค้า	ชื่อผู้ติดต่อ	ที่อยู่	เมือง	รหัสไปรษณีย์	ประเทศ
1	Alfreds Futterkiste	Alfred Schmidt	Obere Str. 57	Hamburg	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitucion 2222	Mexico D.F.	05021	Mexico
3	Antonio Moreno Taqueria	Antonio Moreno	Mataderos 2312	Mexico D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbkop	Christina Berglund	Berguvsvagen 8	Lulea	S-958 22	Sweden

ภาพที่ 2.20 ตาราง customer(2) ที่มีการแก้ไขข้อมูล [7]

4) คำสั่งลบแถวข้อมูล (DELETE) เป็นคำสั่งเพื่อลบแถวข้อมูลออกจากตาราง ซึ่งจะลบข้อมูลเฉพาะที่ตรงกับเงื่อนไขที่ผู้ใช้ต้องการ โดยมีรูปแบบดังนี้

```
DELETE FROM <ชื่อตาราง>
```

```
WHERE <เงื่อนไข>;
```

เช่น

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitucion 2222	Mexico D.F.	05021	Mexico
3	Antonio Moreno Taqueria	Antonio Moreno	Mataderos 2312	Mexico D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbkop	Christina Berglund	Berguvsvagen 8	Lulea	S-958 22	Sweden

ภาพที่ 2.21 ตาราง customer(3) [8]

ต้องการลบข้อมูลของลูกค้าที่ชื่อ Alfreds Futterkiste สามารถเขียนได้ดังนี้

```
DELETE FROM customer
```

```
WHERE CustomerName='Alfreds Futterkiste'
```

```
AND ContactName='Maria Anders';
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitucion 2222	Mexico D.F.	05021	Mexico
3	Antonio Moreno Taqueria	Antonio Moreno	Mataderos 2312	Mexico D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbkop	Christina Berglund	Berguvsvagen 8	Lulea	S-958 22	Sweden

ภาพที่ 2.22 ตาราง customer(3) ที่มีการลบข้อมูล [8]

2.2.1.3 ภาษาควบคุมข้อมูล (Data Control Language: DCL)

เป็นคำสั่งที่ใช้ในการกำหนดสิทธิการอนุญาต หรือ ยกเลิก การเข้าถึงฐานข้อมูล เพื่อป้องกันความปลอดภัยของฐานข้อมูล คำสั่งที่จัดอยู่ในประเภท DCL ได้แก่ คำสั่ง GRANT, REVOKE เป็นต้น

1) GRANT เป็นคำสั่งเพื่อกำหนดสิทธิของการเข้าถึงข้อมูลในฐานข้อมูลว่าผู้ใช้แต่ละคนมีสิทธิกระทำการใดกับข้อมูลได้บ้าง โดยมีรูปแบบดังนี้

```
GRANT <SELECT, INSERT, UPDATE, DELETE>
ON <ชื่อตาราง>
TO <ชื่อผู้ใช้สิทธิ>;
```

2) REVOKE เป็นคำสั่งที่มีไว้สำหรับยกเลิกสิทธิการเข้าถึงข้อมูลในฐานข้อมูล โดยมีรูปแบบดังนี้

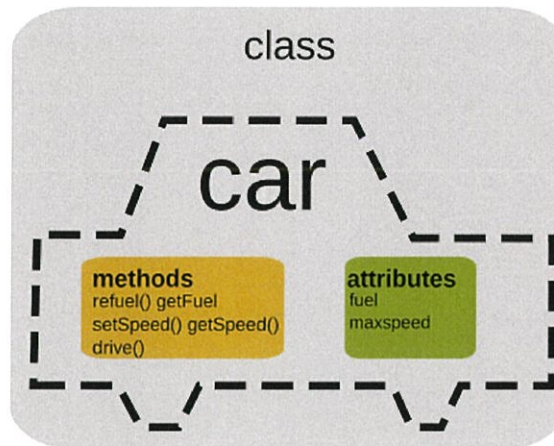
```
REVOKE <SELECT, INSERT, UPDATE, DELETE>
ON <ชื่อตาราง>
FROM <ชื่อผู้ใช้สิทธิ>;
```

2.2.2 ภาษา Java (Java Programming Language) [9]

ภาษา Java เป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP : Object-Oriented Programming) มีวิธีการเขียนที่ไม่ยุ่งยากซับซ้อน สามารถตรวจสอบหาข้อผิดพลาดในโปรแกรมได้ง่าย มีประสิทธิภาพในการทำงานและมีความยืดหยุ่นสูง

การเขียนโปรแกรมเชิงวัตถุ เป็นการเขียนโปรแกรมในลักษณะที่รวมการทำงานและข้อมูลที่เกี่ยวข้องกันไว้ด้วยกัน และในการเขียนโปรแกรมด้วยภาษาจาวานั้น Object จะถูกสร้างขึ้นมาจากคลาส (Class) เสมอ หรือกล่าวได้ว่า คลาส เป็นต้นกำเนิดของ Object

ภายในคลาสประกอบด้วย คุณลักษณะ (Attribute) และ พฤติกรรม (method) ของวัตถุ ยกตัวอย่างเช่น คุณลักษณะของรถ ได้แก่ ความเร็ว เชื้อเพลิง สี จำนวนที่นั่ง เป็นต้น ส่วน พฤติกรรมของรถ ได้แก่ การออกตัวด้วยความเร็วเท่าไร ปริมาณเชื้อเพลิงที่ใช้ตามระยะทาง เป็นต้น



ภาพที่ 2.23 คลาส ซึ่งประกอบไปด้วยตัวแปร และเมธอดในการทำงาน [10]

2.2.2.1 โครงสร้างโปรแกรมของภาษา Java [11]

```
import ... ;
public class Name {
    public static void main (String[] agrs) {
        statement;
    }
}
```

ตัวอย่างโปรแกรม Hello World

```
import java.lang*;
public class HelloWorld {
    public static void main (String[] args) {
        System.out.println("Hello World");
    }
}
```

2.2.3 ภาษา HTML (Hypertext Markup Language) [12]

HTML เป็นภาษาคอมพิวเตอร์ที่ใช้ในการแสดงผลของเอกสารบนเว็บไซต์ มีโครงสร้างการเขียนโดยอาศัยตัวกำกับ (Tag) ควบคุมการแสดงผล

โครงสร้างหลักของ HTML ก็จะเริ่มด้วย <html> และจบด้วย </html> เสมอ ซึ่งชุดคำสั่งที่ใช้จะแยกเป็น 2 ส่วนคือ

1) head คำสั่งที่อยู่ในส่วนนี้จะใช้บรรยายรายละเอียดเกี่ยวกับเว็บเพจ ซึ่งจะไม่แสดงผลที่เว็บเพจโดยตรง

2) body คำสั่งที่อยู่ในส่วนนี้จะใช้ในการจัดรูปแบบตัวอักษร จัดหน้า ใส่รูปภาพ ซึ่งตัวอักษรในส่วนนี้จะแสดงที่เว็บเพจโดยตรง

```
<html>
  <head>
    คำสั่งในหัวข้อของ head (Head Section)
  </head>
  <body>
    คำสั่งในหัวข้อของ body (Body Section) ในส่วนนี้จะเป็นส่วนที่ใช้แสดงผล
  </body>
</html>
```

2.2.4 ภาษา JavaScript [13]

JavaScript เป็นภาษาสคริปต์เชิงวัตถุ ที่ใช้ในการสร้างและพัฒนาเว็บไซต์ (ใช้ร่วมกับ HTML) เพื่อให้เว็บไซต์ดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งเราสามารถเขียนโปรแกรม JavaScript เพิ่มเข้าไปในเว็บเพจเพื่อใช้ประโยชน์สำหรับงานด้านต่างๆ เช่น การแสดงผล การรับ-ส่งข้อมูล และยังสามารถโต้ตอบกับผู้ใช้ได้อย่างทันทีทันใด

2.2.5 ภาษา jQuery [14]

jQuery คือ JavaScript Library ซึ่งถูกออกแบบมาเพื่อให้การเขียน JavaScript นั้นมีความสะดวกและง่ายขึ้น ทำให้การเขียนโค้ดที่อาจจะใช้ JavaScript หลายๆ บรรทัด สามารถเขียนให้สั้นลงได้

2.2.5.1 jQuery Basic Syntax

jQuery การทำงานส่วนมากในเว็บฟอร์มจะให้ความสำคัญในการอ้างถึง (Selectors) element ในรูปแบบต่างๆ หากจะเข้าใจง่ายๆ ในระบบการเขียน JavaScript แบบดั้งเดิมจะใช้ document.form.element หรือ document.getElementById ในการอ้างถึง element เช่น

```
<input type="text" id="txtname">
```

ถ้าในระบบเดิมของ JavaScript เราจะเรียกใช้ element นี้ได้โดยการใช้คำสั่ง

```
document.form.txtname.value = 'myText';  
document.getElementById("").value = 'myText';
```

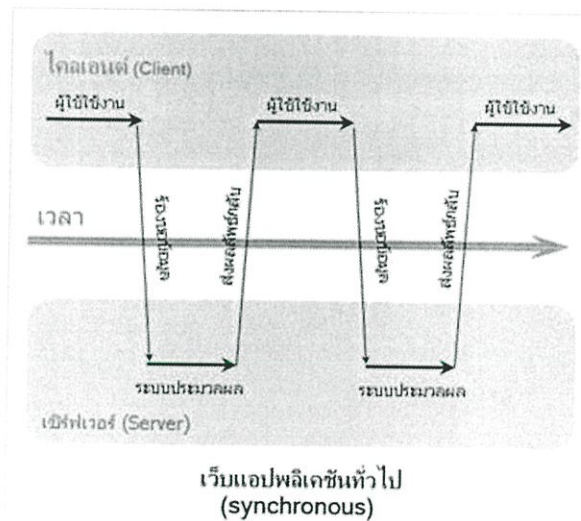
แต่ใน Syntax ของ jQuery คำสั่งพวกนี้จะถูกใช้เพียงสั้นๆ เท่านั้น

```
$("#txtname").val("myText");
```

2.2.5.2 jQuery and Ajax

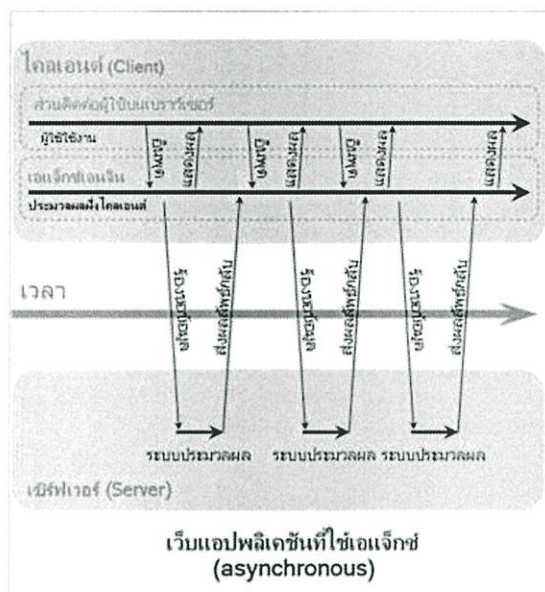
Ajax (Asynchronous JavaScript and XML) เป็นชื่อของเทคนิคที่จะเปลี่ยนแปลงบางส่วนของหน้าเว็บเฉพาะส่วนที่เปลี่ยนไปจริงๆเท่านั้น โดยจะไม่เปลี่ยนส่วนที่ไม่เกี่ยวข้อง

วิธีการทำงานของเว็บแอปพลิเคชันแบบดั้งเดิมนั้น เป็นวิธีการแบบการร้องขอและการตอบรับ (Request and Response) คือหลังจากที่ผู้ใช้ทำการร้องขอข้อมูลจากเซิร์ฟเวอร์แล้ว ผู้ใช้จะต้องรอระหว่างที่เซิร์ฟเวอร์ประมวลผลอยู่ เมื่อเว็บเซิร์ฟเวอร์ทำการประมวลผลเสร็จ จะส่งผลลัพธ์เป็นหน้า HTML กลับไปให้ผู้ใช้ ซึ่งเป็นหลักการทำงานแบบ Synchronous



ภาพที่ 2.24 การทำงานของเว็บแอปพลิเคชันแบบดั้งเดิม [15]

ส่วนการทำงานของเว็บแอปพลิเคชันที่ใช้เทคนิค Ajax จะเป็นการทำงานแบบ Asynchronous โดยที่เซิร์ฟเวอร์จะทำการส่งผลลัพธ์เป็นเว็บเพจให้ผู้ใช้ทันทีโดยไม่ต้องรอให้ประมวลผลเสร็จก่อน หลังจากนั้นเว็บเพจที่ผู้ใช้ได้รับจะทำการดึงข้อมูลในส่วนต่างๆที่หลัง หรือจะดึงข้อมูลก็ต่อเมื่อผู้ใช้ต้องการเท่านั้น (ทำงานอยู่เบื้องหลัง)



ภาพที่ 2.25 การทำงานของเว็บแอปพลิเคชันที่ใช้เทคนิค Ajax [15]

ใน jQuery ก็ได้ออกแบบ function ที่ทำงานร่วมกับ Ajax ได้อย่างมีประสิทธิภาพ ด้วยโค้ดคำสั่งสั้นๆ คือ `jQuery.ajax()` ซึ่งเป็นการใช้ jQuery กับ Ajax เพื่อติดต่อระหว่าง Server กับ Client ในรูปแบบต่าง ๆ

2.2.5.3 jQuery, Ajax and JSON

การส่งข้อมูล Ajax ด้วย jQuery สามารถส่งข้อมูลได้หลายประเภท เช่น ข้อมูลที่อยู่รูปแบบ String, JSON Object, JSON String เป็นต้น

1) การส่งค่าจาก jQuery โดยส่งไปเป็นแบบ JSON Object

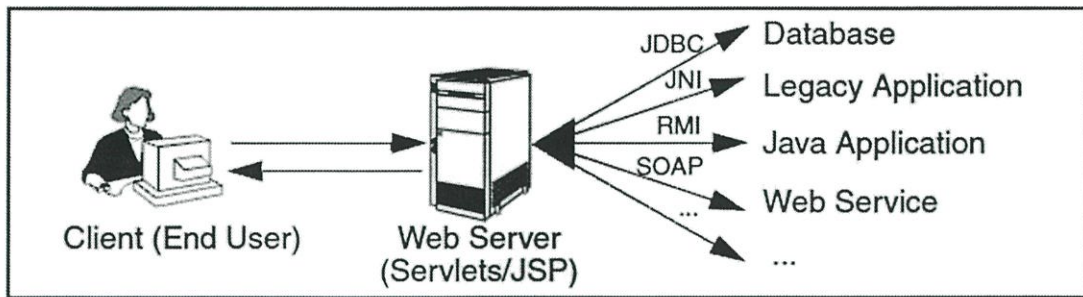
สำหรับการส่งค่าในรูปแบบนี้คือชุดของข้อมูลที่จะส่งอยู่ในรูปแบบของ JSON เรียบร้อยแล้วและพร้อมที่จะส่งไปกับ Ajax ได้เลย

2) การส่งค่าจาก jQuery โดยส่งไปเป็นแบบ JSON String

สำหรับการส่งค่าในรูปแบบนี้คือข้อมูล JSON จะอยู่ในรูปแบบของ String คือ ข้อความ ซึ่งในการส่งผ่าน Ajax เราจะต้องระบุ `contentType: 'application/json'` ไปด้วย

2.2.6 Servlet [16]

Servlet เป็นโปรแกรม Java ที่รันบนเว็บเซิร์ฟเวอร์ทำหน้าที่สร้างหน้าเว็บ ส่งไปยังผู้ใช้ (client) ตามคำร้องขอ (request) หลักการทำงานของ Servlet เป็นแบบ Server-side processing คือ ฟังก์ชันเซิร์ฟเวอร์เป็นผู้รันโปรแกรมและส่งผลลัพธ์ให้กับผู้ใช้



The role of Web middleware.

ภาพที่ 2.26 หลักการทำงานของ JSP และ Servlet [17]

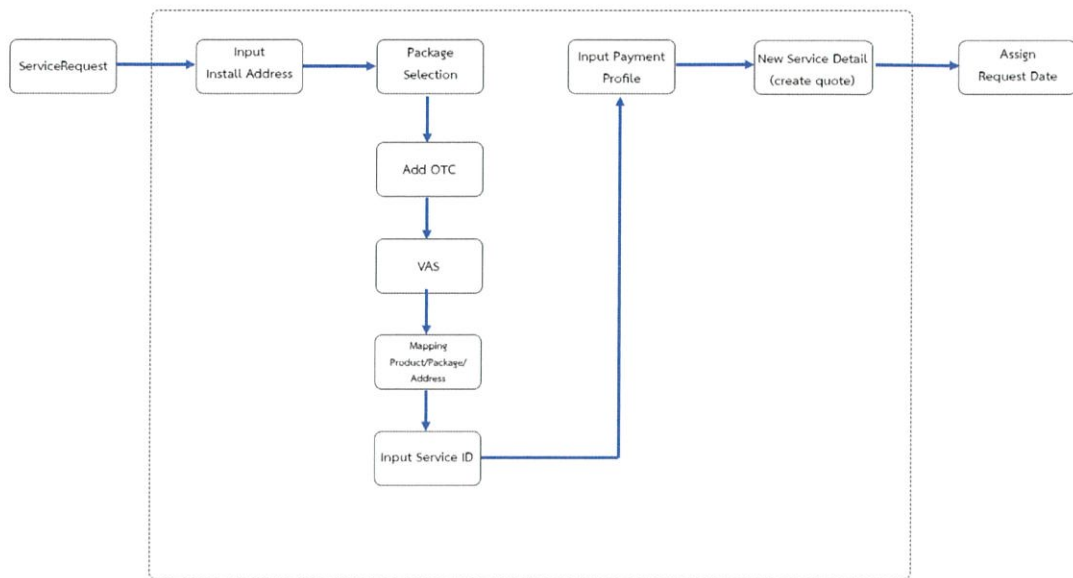
ดังนั้น เราใช้ Servlet เพื่อให้บริการอะไรบางอย่าง แก่ผู้รับบริการที่อยู่บนเครื่องคอมพิวเตอร์เครื่องหนึ่ง

บทที่ 3

วิธีดำเนินงาน

3.1 แนวทางการออกแบบระบบ

ในโครงการนี้ ส่วนที่รับผิดชอบจะเป็นการเขียนโปรแกรมเพื่อการพิสูจน์แนวคิดของระบบ (Proof of Concept) เฉพาะการทำงานในโมดูลที่เกี่ยวกับ New Service Request เท่านั้น



ภาพที่ 3.1 แสดงลำดับการทำงานของหน้าจอต่างๆ (New Service Order Screen Workflow)

3.1.1 แบบร่างหน้าจอ

1) ServiceRequest

Service Request

SR No:

Customer Name:

Received:

Home Location:

Order Type:

To:

No.	SR No	Customer Name	Order Type	Package	Products	Asset No	Install Address	Received	Description	Home Location	Initiator	Status
1	SR-202202-202202	Artith Samrit	ขอติดตั้ง (New)					20/09/2017	รายละเอียดขอ Service Request	0101	Automatic Handle	New
2	SR-202202-202202	Artith Samrit	ขอติดตั้ง (New)					20/09/2017	รายละเอียดขอ Service Request	0102	Automatic Handle	New
3	SR-202202-202202	Artith Samrit	ขอติดตั้ง (New)					20/09/2017	รายละเอียดขอ Service Request	0103	Automatic Handle	New
4	SR-202202-202202	Wirat Ja-dee	ขอติดตั้ง (New)					20/09/2017	รายละเอียดขอ Service Request	0104	Automatic Handle	New
5	SR-202202-202202	Wirat Ja-dee	ขอติดตั้ง (New)					20/09/2017	รายละเอียดขอ Service Request	0105	Automatic Handle	New
6	SR-202202-202202	Wirat Ja-dee	ขอติดตั้ง (New)					20/09/2017	รายละเอียดขอ Service Request	0106	Automatic Handle	New

ภาพที่ 3.2 ServiceRequest หน้า search

- เมื่อกดปุ่ม Search จะแสดงข้อมูลออกมาตามเงื่อนไขที่ระบุ

- เลือก record ที่ต้องการ จากนั้นจะแสดงหน้า update

Service Request Detail

SR No	<input type="text" value="SR-22222-22222"/>	Order Type	<input type="text" value="ชนิดตัวใหม่ (New)"/>
SR Date	<input type="text" value="#####"/>	Sale	<input type="text" value="นางสาวเกสรภรณ์ แสงแก้ว"/>
SR Owner	<input type="text" value="Agent2039 CRM"/>	Customer Sub Type	<input type="text" value="ลูกค้าบุคคลทั่วไป (บ้านพักอาศัย)"/>
Customer Name	<input type="text" value="นายจักรพันธ์ นฤวรวัฒน์"/>	Project Name	<input type="text"/>
Customer Type	<input type="text" value="ลูกค้าบ้านพักอาศัยประเภททั่วไป"/>	Billing Name	<input type="text" value="นายจักรพันธ์ นฤวรวัฒน์"/>
Cover Sheet Id	<input type="text"/>	Bill Layout	<input type="text" value="ใบแจ้งหนี้ปกติ"/>
Billing Account	<input type="text"/>	Contact Tel	<input type="text" value="0850909090"/>
Billing Address	<input type="text" value="1/150 อาคารเฟรนด์ สแควร์
เหนือ สยาม สแควร์ กรุงเทพฯ
10500"/>		
Service Type	<input type="text" value="Fixed Line (Telco)"/>		
Contact Person	<input type="text" value="นายจักรพันธ์ นฤวรวัฒน์"/>		

ภาพที่ 3.3 ServiceRequest หน้า update

- หน้า update จะแสดงข้อมูลของ record ที่เลือก
- หากกดปุ่ม Reject จะมี popup แสดงขึ้นมาให้กรอกเหตุผลในการ reject

Popup X

Service Request Reject

Reason

Remark

ภาพที่ 3.4 Reject Popup

- เมื่อกด Confirm จะเก็บสถานะ 'Reject' ลงในฐานข้อมูล และจบการทำงาน

- หากกดปุ่ม Submit จะทำการเก็บสถานะ 'Submit' ลงในฐานข้อมูล และ redirect screen ไปหน้า Input Install Address

2) Input Install Address

Back Submit Reject

Service Request Detail

SR No: SR Date:

Order Type: Customer Type:

CA Account: Customer Name:

Billing Account: Billing Name:

Billing Address:

Install Address

<input type="checkbox"/>	No	Moo	Village/Building	Soi	Road	SubDistrict	District	Province	Zipcode	Nearly Place	Service Location
<input type="checkbox"/>	171/131	11	อาคารพหลโยธิน ชั้น 15	2	สาทร	บางรัก	สีลม	กทม	10500	บ่อน้ำผุดหัว	ศูนย์หลัก
<input type="checkbox"/>	123/34	22	อาคารพหลโยธิน ชั้น 16	2	สาทร	บางรัก	สีลม	กทม	10500	อาคารคิงเฮาส์	ศูนย์หลัก

avalant: Default 1 record from legal address which get from Service Request

ภาพที่ 3.5 Input Install Address

- กด Add ที่ layer ล่าง เพื่อเพิ่มที่อยู่
- ปุ่ม Reject ทำงานเหมือนในข้อ 1)
- ปุ่ม Submit ทำงานเหมือนในข้อ 1) แต่มีเงื่อนไขว่า ต้องระบุ Address อย่างน้อย 1 record ถึงจะสามารถ redirect screen ไปหน้า Package Offering ได้

3) Package Offering

Back Submit Reject

Service Request Detail

SR No: SR Date:

Order Type: Customer Type:

CA Account: Customer Name:

Billing Account: Billing Name:

Billing Address:

Package Offering

Package

<input type="checkbox"/>	Package Name	Product Name	Price	Period Date	Promotion Bill Cycle	Quantity
<input type="checkbox"/>	บริการ Strong Life Up to 40Mb/20Mb_Fixed Line	Normal Telephone	500.00	01/01/2011 - 30/04/2018	1	1
<input checked="" type="checkbox"/>		General Discount Product(n)	0	01/01/2011 - 30/04/2018	1	1
<input type="checkbox"/>		Fiber 2U(n)	590.00	01/01/2011 - 30/04/2018	1	1

ภาพที่ 3.6 Package Offering

- กด Add ที่ layer ล่าง เพื่อเลือก Package อื่นๆ
- ปุ่ม Reject ทำงานเหมือนในข้อ 1)
- ปุ่ม Submit ทำงานเหมือนในข้อ 1) แต่มีเงื่อนไขว่า ต้องระบุ Package อย่างน้อย 1 record ถึงจะสามารถ redirect screen ไป หน้า Add OTC ได้

4) OTC (One Time Charge)

Service Request Detail

Back
Submit
Reject

SR No

Order Type

CA Account

Billing Account

Billing Address

SR Date

Customer Type

Customer Name

Billing Name

One Time Charge

One Time Charge

Add
Delete

<input type="checkbox"/>	OTC ID	OTC Name	Total	Payment Method	
<input checked="" type="checkbox"/>	1	ค่าติดตั้ง/ย้าย-โทรศัพท์	3,350.00	Pay-First	/
<input checked="" type="checkbox"/>	166	ค่าธรรมเนียมแรกเข้า / ค่าขอใช้บริการ PrivateNet	1,000.00	Next Bill	/

ภาพที่ 3.7 Add OTC

- กด Add ที่ layer ล่าง เพื่อเลือก OTC จากหน้า Search OTC

One Time Charge

Search
Clear

! warning:
Search OTC under all
selected package

<input type="checkbox"/>	OTC ID	OTC Name	Total	
<input type="checkbox"/>	1	ค่าติดตั้ง/ย้าย-โทรศัพท์	3,350.00	+
<input type="checkbox"/>	166	ค่าธรรมเนียมแรกเข้า / ค่าขอใช้บริการ Free Phone	1,000.00	+
<input type="checkbox"/>	167	ค่าธรรมเนียมแรกเข้า / ค่าขอใช้บริการ UAN	1,000.00	+
<input type="checkbox"/>	168	ค่าธรรมเนียมแรกเข้า / ค่าขอใช้บริการ PrivateNet	1,000.00	+
<input type="checkbox"/>	169	ค่าธรรมเนียมแรกเข้า / ค่าขอใช้บริการ Vote Now	1,000.00	+
<input type="checkbox"/>	225	ค่าติดตั้งโทรศัพท์มือถือ	500.00	+

Submit
Cancel

ภาพที่ 3.8 Search OTC

- กดปุ่ม Submit (หน้า Search OTC) ข้อมูลที่เลือกจะไปขึ้นที่ layer ล่างของหน้า Add OTC
- ปุ่ม Reject ทำงานเหมือนในข้อ 1)
- ปุ่ม Submit (หน้า Add OTC) ทำงานเหมือนในข้อ 1) และ redirect screen ไปหน้า Product VAS

5) VAS (Value Added Service)

Back
Submit
Reject

Service Request Detail

SR No	<input type="text" value="SR-222222-222222"/>	SR Date	<input type="text" value="02/09/2017"/>
Order Type	<input type="text" value="สมัครใช้บริการ(ใหม่)"/>	Customer Type	<input type="text" value="ลูกค้าบ้านพักอาศัยประเภททั่วไป"/>
CA Account	<input type="text" value="CA-000000-000000"/>	Customer Name	<input type="text" value="นายจักษิณ บุญวรรณ"/>
Billing Account	<input type="text" value="123456789012"/>	Billing Name	<input type="text" value="นายจักษิณ บุญวรรณ"/>
Billing Address	<input type="text" value="1150 อาคารพาณิชย์"/>		

Package Offering

Package/Product - VAS

No.	Package Name	Product Name	Price	Period Date	Promotion Bill Cycle	Quantity	VAS
1	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	Normal Telephone	500.00	01/01/2011 - 30/04/2018	1	1	VAS
2	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	General Discount Product(n)	0	01/01/2011 - 30/04/2018	1	1	VAS
3	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	Fiber 2U(n)	590.00	01/01/2011 - 30/04/2018	1	1	VAS

ภาพที่ 3.9 Product VAS

- ที่ layer ล่าง กดปุ่ม VAS (เพื่อแก้ไข) จะไปหน้า Add VAS

Submit
Cancel

Package/Product Detail

Package Name	<input type="text" value="ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Lin"/>	Period Date	<input type="text" value="1/01/2012 - 04/04/201"/>
Product Name	<input type="text" value="Normal Telephone"/>	Promotion Bill Cycle	<input type="text" value="1"/>
Price Plan Name	<input type="text" value="Mthly Tel LC.LD Free 0800-1700 MB 1 Baht"/>		

VAS

Value Added Service
Add
Delete

<input type="checkbox"/>	VAS Group	VAS Code	VAS Name
<input type="checkbox"/>	SPC	1	SPC-Pack 1(Free : รับสายเรียกข้อมโอนสาย)
<input type="checkbox"/>	SPC	2	SPC-Pack 8 (Free : รับสายเรียกข้อมโอนสายไปรษณีย์ผ่านโทรศัพท์ที่เรียกซ้ำอัตโนมัติ)
<input type="checkbox"/>	SPC	3	SPC-Pack 3 (เสริมหมายเลข/เสริมหมายเลข/แสดงเลขหมายโทรศัพท์)
<input type="checkbox"/>	SPC	4	Std. SPC-Call Forwarding (บริการโอนสาย)
<input type="checkbox"/>	SPC	5	Std. SPC-Automated Call Repetition (บริการโทรซ้ำอัตโนมัติ)
<input type="checkbox"/>	SPC	6	Std. SPC-Abbreviated Dialling (บริการขอเลขหมาย)
<input type="checkbox"/>	SPC	7	Std. SPC-Conference Call (บริการสนทนา 3 สาย)
<input type="checkbox"/>	SPC	9	Std. SPC-Call Waiting (บริการรับสายเรียกข้อม)
<input type="checkbox"/>	SPC	10	Std. SPC-Immediate Charge Info (บริการแจ้งค่าชุดโทรศัพท์ทางไกลทันที)

ภาพที่ 3.10 Add VAS

- กด Add ที่ layer ล่างของหน้า Add VAS จะแสดงหน้า Search VAS

Value Added Service

VAS Group VAS Name

<input type="checkbox"/>	VAS Group	VAS Code	VAS Name	
<input type="checkbox"/>	SPC	1	SPC-Peak 1(Free:รับสายบริการด่วน/โทรมา)	+
<input type="checkbox"/>	SPC	2	SPC-Peak 2(Free:รับสายบริการด่วน/โทรมา/บริการลูกค้าโทรศัพท์/รับสายอัตโนมัติ)	+
<input type="checkbox"/>	SPC	3	SPC-Peak 3 (Service Charge: ค่าบริการตามอัตราค่าบริการตามโทรมา)	+
<input type="checkbox"/>	SPC	4	SM-SPC-Call Forwarding (บริการโอนสายตาม)	+
<input type="checkbox"/>	SPC	5	SM-SPC-Redacted Call Rejection (บริการโทรขัดอัตโนมัติ)	+
<input type="checkbox"/>	SPC	6	SM-SPC-Abbreviated Dialling (บริการกดเลขตาม)	+
<input type="checkbox"/>	SPC	7	SM-SPC-Conferee Call (บริการรับสาย 3 ฝ่าย)	+
<input type="checkbox"/>	SPC	9	SM-SPC-Call Waiting (บริการรับสายบริการด่วน)	+
<input type="checkbox"/>	SPC	10	SM-SPC-Immediate Charge Infa (บริการแจ้งค่าโทรทันทีก่อนโทรออก)	+
<input type="checkbox"/>	SPC	11	SM-SPC-Outgoing Call Barring (บริการจำกัดการโทรออก)	+
<input type="checkbox"/>	SPC	13	SM-SPC-Slip Service In display call number (บริการรับสายแสดงเลขตามโทรออก)	+
<input type="checkbox"/>	Miscellaneous	112	Normal Telephone - ตามบริการอื่น	+
<input type="checkbox"/>	Miscellaneous	113	N-ISDN - DDI ตามบริการอื่น	+
<input type="checkbox"/>	Miscellaneous	114	ADSL - ตามบริการอื่น	+

ภาพที่ 3.11 Search VAS

- เลือก record ที่ต้องการ กด Submit (หน้า Search VAS) ข้อมูลจะไปขึ้นที่ layer ล่างของหน้า Add VAS
- กดปุ่ม Submit หน้า Add VAS จะกลับไปหน้า Product VAS
- ปุ่ม Reject ทำงานเหมือนในข้อ 1)
- ปุ่ม Submit (หน้า Product VAS) ทำงานเหมือนในข้อ 1) และ redirect screen ไปหน้า Mapping Product

6) Mapping Product

Back Submit Reject

Service Request Detail

SR No	SR-222222-222222	SR Date	02/09/2017
Order Type	สมัครรับใหม่ (New)	Customer Type	ลูกค้าบ้านพักอาศัยประเภททั่วไป
CA Account	CA-000000-000000	Customer Name	นายจักรพันธ์ นฤวรทัศน์
Billing Account	123456789012	Billing Name	นายจักรพันธ์ นฤวรทัศน์
Billing Address	1/150 อาคารบุฟไฟด์ สาทรเหนือ สยาม สาทร กรุงเทพฯ 10500		

Package/Product - Install Address

No.	Package Name	Product Name	Install Address
1	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	Normal Telephone	1/150 อาคารบุฟไฟด์ สาทรเหนือ สยาม สาทร กรุงเทพฯ 10500
2	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	General Discount Product(n)	
3	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	Fiber 2U(n)	11/229 เคซี เนเชอรัลส์ ซิมเกล้า 11 แสวนแสน มิ้นบุรี กรุงเทพฯ 10510

ภาพที่ 3.12 Mapping Product

- กดแก้ไข หรือ ที่ layer ล่างของหน้า Mapping Product จะแสดงหน้าต่าง Popup ให้เลือก Address

Package/Product - Install Address

Package Name	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line
Product Name	Normal Telephone
Install Address	1/150 อาคารบุฟไฟด์ สาทรเหนือ สยาม สาทร กรุงเทพฯ 10500

Save Cancel

ภาพที่ 3.13 Mapping Popup

- เลือก Address ในช่อง Install Address กด Save ข้อมูลจะไปขึ้นที่ layer ล่างของหน้า Mapping Product
- ที่นี่ หากจะทำการเพิ่ม Address ให้กับ record อื่นๆ มีเงื่อนไขว่า เมื่อ Mapping Popup แสดงขึ้นมา ตรงช่อง Install Address จะไม่ปรากฏ Address ที่ถูกเลือกไปแล้ว
- ปุ่ม Reject ทำงานเหมือนในข้อ 1)

- ปุ่ม Submit (หน้า Mapping Product) ทำงานเหมือนในข้อ 1) แต่จะต้องระบุ Address ให้ครบทุก record จึงจะสามารถ redirect screen ไปหน้า Input Service ID ได้

7) Input Service ID

Back Submit Reject

Service Request Detail

SR No	<input type="text" value="SR-22222-22222"/>	SR Date	<input type="text" value="02/09/2017"/>
Order Type	<input type="text" value="สมัครใช้งาน (New)"/>	Customer Type	<input type="text" value="ลูกค้าบ้านพักอาศัยประเภททั่วไป"/>
CA Account	<input type="text" value="CA-000000-000000"/>	Customer Name	<input type="text" value="นายจักรพันธ์ นฤพรพัฒน์"/>
Billing Account	<input type="text" value="123456789012"/>	Billing Name	<input type="text" value="นายจักรพันธ์ นฤพรพัฒน์"/>
Billing Address	<input type="text" value="1/150 อาคารพหลโยธิน สาทรเหนือ สยาม สหกรณ์ กรุงเทพ 10500"/>		

Service ID

Input Service ID

No.	Service ID	Package Name	Product Name	Install Address	Service Location
1		ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	Normal Telephone	1/150 อาคารพหลโยธิน สาทรเหนือ สยาม สหกรณ์ กรุงเทพ 10500	ศูนย์ฯ หลักสี่ ✓
2		ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	General Discount Product(n)		ศูนย์ฯ หลักสี่ ✓
3		ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	Fiber 2U(n)	11229 เคซี เนเชอรัลส์ ร่มเกล้า 11 แสงสาม มินนิชิ กรุงเทพ 10510	ศูนย์ฯ หลักสี่ ✓

ภาพที่ 3.14 Input Service ID

- กดแก้ไข หรือ ที่ layer ล่าง จะแสดงหน้า Popup ให้ระบุ Service ID

Popup X

Service ID

Package Name	<input type="text" value="ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line"/>
Product Name	<input type="text" value="Normal Telephone"/>
Install Address	<input type="text" value="1/150 อาคารพหลโยธิน สาทรเหนือ สยาม สหกรณ์ กรุงเทพ 10500"/>
Service ID	<input type="text"/>

Save Cancel

ภาพที่ 3.15 Service ID popup

- ปุ่ม Reject ทำงานเหมือนในข้อ 1)
- ปุ่ม Submit ทำงานเหมือนในข้อ 1) และ redirect screen ไปหน้า Payment Profile

8) Payment Profile

Service Request Detail

Back Submit Reject

SR No	<input type="text" value="SR-222222-222222"/>	SR Date	<input type="text" value="02/05/2017"/>
Order Type	<input type="text" value="ซ่อมบำรุง (SR)"/>	Customer Type	<input type="text" value="ลูกค้าบ้านพักอาศัย/ผู้ประกอบการทั่วไป"/>
CA Account	<input type="text" value="CA-000000-000000"/>	Customer Name	<input type="text" value="นายจักรพันธ์ บุตรทรัพย์"/>
Billing Account	<input type="text" value="123456789012"/>	Billing Name	<input type="text" value="นายจักรพันธ์ บุตรทรัพย์"/>
Billing Address	<input type="text" value="1/158 ซอยบางขุนนนท์ ซอยประเวศ ซอย
บางขุนนนท์ 10500"/>		

Payment Profile

Billing Address

No	<input type="text" value="1/158"/>	Mo	<input type="text"/>
Village/Building	<input type="text" value="บางขุนนนท์"/>	Soi	<input type="text"/>
Road	<input type="text" value="บางขุนนนท์"/>	SubDistrict	<input type="text" value="..."/>
District	<input type="text" value="..."/>	Province	<input type="text" value="..."/>
Zipcode	<input type="text" value="10500"/>		

Payment Profile

รวมทุกสาขาใน DR เดียวกัน แยก DR ตามเขตสาขา (Ready Package สาขาอื่น)
 แยก DR ตามเขตสาขา (สาขาอื่นมี 30 วันก่อนเปิดให้บริการ)

Issuer Type	<input type="text" value="ใบแจ้งหนี้ทั่วไป"/>	Payment Method	<input type="text" value="Chequer/Cash"/>	Issuer Language	<input type="text" value="TH (TH)"/>
Home Location	<input type="text"/>	Drkt. Notification Channel	<input type="text"/>	Sum of Issuer	<input type="text" value="1"/>
Bill Cycle	<input type="text"/>	Bill Type	<input type="text"/>	Bill Media	<input type="text"/>
Job Type	<input type="text" value="ช่างทั่วไป"/>	VAT	<input type="text"/>	E-mail	<input type="text"/>
เบอร์โทรศัพท์ (1)	<input type="text"/>	เบอร์โทรศัพท์ (2)	<input type="text"/>	เบอร์โทร	<input type="text"/>
เบอร์โทรfax	<input type="text"/>	เบอร์โทรfax	<input type="text"/>		
Company	<input type="text"/>	Bill Period	<input type="text" value="1"/>		

ภาพที่ 3.16 Payment Profile

- ปุ่ม Reject ทำงานเหมือนในข้อ 1)
- ปุ่ม Submit ทำงานเหมือนในข้อ 1) และ redirect screen ไปหน้า Create Quote

9) Create Quote

Service Request Detail

Back Submit Reject Print

SR No:

Order Type:

CR Number:

Billing Number:

Billing Address:

SR Date:

Customer Type:

Customer Name:

Billing Name:

Quotation

Install Address

Service ID	Package Name	Product	Price	Price Plan	Period Date	Install Address	Service Inactive
021001000	บริการสาย Stream Life Up to 40Mbps/20Mbps, Fixed Line	Normal Telephone	1,000.00	High Tel LC, LD Fee 0000-1700 HD 1 D, 0.00	02/03/2017	1/158 ซอยบางพลีใหญ่ แขวงบางพลี เขต บางพลี กรุงเทพมหานคร 10500	ศูนย์แม่ข่าย
021001000	บริการสาย Stream Life Up to 40Mbps/20Mbps, Fixed Line	Fiber 2Upl	1,000.00	Stream Life Up to 40Mbps/20Mbps, Fib	02/03/2017	174/151 ซอยบางพลีใหญ่ ชั้น 45 แขวงบางพลีใหญ่ เขต บางพลี กรุงเทพมหานคร 10500	ศูนย์แม่ข่าย

VAS

No	VAS Group	VAS Code	VAS Name
1	SPC	1	SPC-Peak 1 [Fee: 3 บาท/ชม./วัน/สาย]
2	SPC	2	SPC-Peak 2 [Fee: 3 บาท/ชม./วัน/สาย / ค่าบริการรายชั่วโมง / 1 บาท/ชม./วัน/สาย]
3	SPC	3	SPC-Peak 3 [ค่าบริการรายชม. / ค่าบริการรายวัน/สาย]
4	SPC	4	SID-SPC-Call Forwarding [ค่าบริการรายชม.]
5	SPC	5	SID-SPC-Redundant Call Repeating [ค่าบริการรายชม./วัน/สาย]
6	SPC	6	SID-SPC-Redundant Call Billing [ค่าบริการรายชม.]
7	SPC	7	SID-SPC-Conference Call [ค่าบริการ/ชม./สาย]
8	SPC	8	SID-SPC-Call Waiting [ค่าบริการรายชม./สาย]
9	SPC	10	SID-SPC-Immediate Charge Info [ค่าบริการรายชม./สาย/วัน/สาย]

Use Time Charge

No	OTC ID	OTC Name	Total	Payment Method
1	1	ค่าบริการรายชม./วัน/สาย	3,350.00	Pay Final
2	100	ค่าบริการรายชม./สาย/วัน/สาย / ค่าบริการรายชม./วัน/สาย	1,000.00	Real Bill

Quotation Language

Thai English

ภาพที่ 3.17 Create Quote

- ปุ่ม Reject ทำงานเหมือนในข้อ 1)
- ปุ่ม Submit ทำงานเหมือนในข้อ 1) และเป็นการจบการทำงานในส่วนของโมดูล New Service Request

ตารางที่ 3.1 ตารางสรุปหน้าจอ (Screen) และ Entity ที่เกี่ยวข้อง

Screen Name	Entity Name	Entity ID
ServiceRequest	SERqstDetail_1	EN_171005121322672_v001
Input Install Address	SERqstDetailNew_2	EN_6137580665_v001
Package Offering	SEPackageOffering_3	EN_3790213085_v001
Add OTC	SeAddOTC3.1	EN_4885979025_v001
Search OTC	Search_OTC3.2	EN_5013526585_v001
Product-VAS	Product_VAS	EN_4875773690_v001
Add VAS	Add_VAS	EN_3631304132_v001

Search VAS	Search_Vas	EN_171115053015078_v001
Mapping Product	Mapping_product	EN_171115063912816_v001
Input Service ID	InputServiceID_8	EN_7084170929_v001
Payment Profile	SEPayment Profile_5 new	EN_2236143642_v001
Create Quote	SEReqCreateQuote_6	EN_2257727007_v001

3.2 ส่วนที่ต้องเขียนโปรแกรมเพิ่มเติม (Customize)

ตารางที่ 3.2 สรุปรงานที่ต้องทำการ Customize

Screen Name	Customize
ServiceRequest (หน้า update)	เขียน Reject popup และให้บันทึกค่าสถานะ 'Reject' และเหตุผลในการ reject ลงในฐานข้อมูล
	กด Submit ให้บันทึกค่าสถานะ 'Submit' ลงในฐานข้อมูล และ redirect screen
Input Install Address	เขียน Reject popup และให้บันทึกค่าสถานะ 'Reject' และเหตุผลในการ reject ลงในฐานข้อมูล
	ทำการ Validate ว่าต้องมี Address อย่างน้อย 1 record เมื่อกด Submit
	กด Submit ให้บันทึกค่าสถานะ 'Submit' ลงในฐานข้อมูล และ redirect screen
Package Offering	เขียน Reject popup และให้บันทึกค่าสถานะ 'Reject' และเหตุผลในการ reject ลงในฐานข้อมูล
	ทำการ Validate ว่าต้องมี Package อย่างน้อย 1 record เมื่อกด Submit
	กด Submit ให้บันทึกค่าสถานะ 'Submit' ลงในฐานข้อมูล และ redirect screen
Add OTC	เขียน Reject popup และให้บันทึกค่าสถานะ 'Reject' และเหตุผลในการ reject ลงในฐานข้อมูล

	กด Submit ให้บันทึกค่าสถานะ 'Submit' ลง ในฐานข้อมูล และ redirect screen
Product VAS	เขียน Reject popup และให้บันทึกค่าสถานะ 'Reject' และเหตุผลในการ reject ลงใน ฐานข้อมูล
	กด Submit ให้บันทึกค่าสถานะ 'Submit' ลง ในฐานข้อมูล และ redirect screen
Search VAS	ทำการ Filter VAS ที่อยู่ภายใต้ product id ใน package ที่เลือก
Mapping Product	เขียน Reject popup และให้บันทึกค่าสถานะ 'Reject' และเหตุผลในการ reject ลงใน ฐานข้อมูล
	หากจะทำการเพิ่ม Address ให้กับ record อื่นๆ มีเงื่อนไขว่า จะต้องไม่ปรากฏ Address ที่ ถูกเลือกไปแล้ว
	ทำการ Validate ว่าต้องระบุ Address ครบทุก record เมื่อกด Submit
	กด Submit ให้บันทึกค่าสถานะ 'Submit' ลง ในฐานข้อมูล และ redirect screen
Input Service ID	เขียน Reject popup และให้บันทึกค่าสถานะ 'Reject' และเหตุผลในการ reject ลงใน ฐานข้อมูล
	กด Submit ให้บันทึกค่าสถานะ 'Submit' ลง ในฐานข้อมูล และ redirect screen
Payment Profile	เขียน Reject popup และให้บันทึกค่าสถานะ 'Reject' และเหตุผลในการ reject ลงใน ฐานข้อมูล
	กด Submit ให้บันทึกค่าสถานะ 'Submit' ลง ในฐานข้อมูล และ redirect screen
Create Quote	เขียน Reject popup และให้บันทึกค่าสถานะ 'Reject' และเหตุผลในการ reject ลงใน ฐานข้อมูล

	กด Submit ให้บันทึกค่าสถานะ 'Submit' ลงในฐานข้อมูล และจบการทำงานของโมดูล (complete flow)
--	--

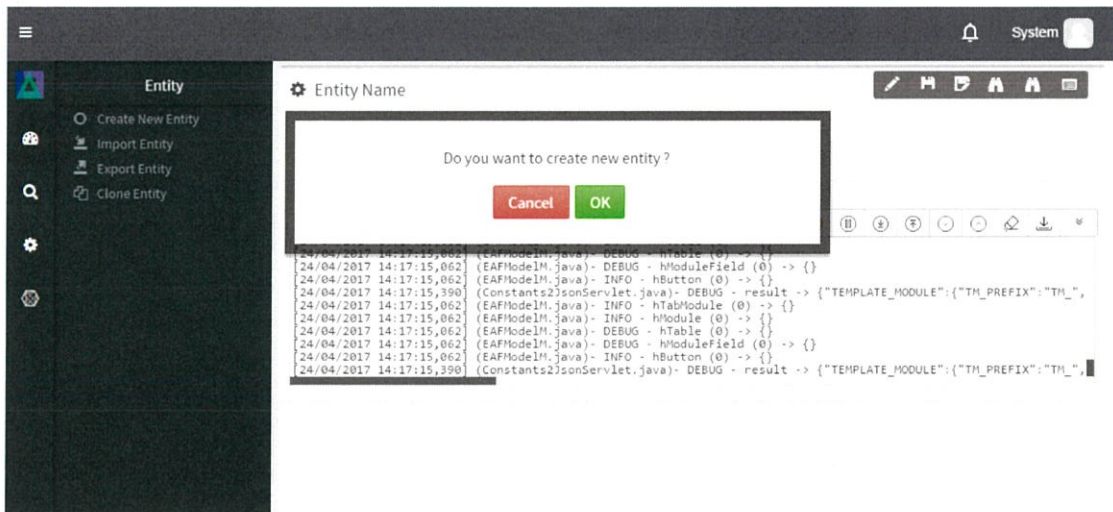
3.3 การดำเนินงาน

3.3.1 สร้างหน้า User Interface โดยใช้ ONEWEB 4.0

สร้างหน้า User Interface ตามแบบร่างหน้าจอในหัวข้อ 3.1.1

3.3.1.1 สร้าง Entity

- 1) เลือก Create New Entity ด้านซ้าย
- 2) มี Popup ขึ้นมา ให้กด OK



ภาพที่ 3.18 Create New Entity

- 3) กด Edit



- 4) ทำการตั้งชื่อ Entity ในช่อง Entity name จากนั้น กด OK

Entity Configuration

Entity ID: EN_171005121322672_v001

Entity name: SERReqDetail_1

Template code: Responsive2016

Theme code: Responsive2016

System code: EAF@GEDB_10G

Version: v001

Default search: N

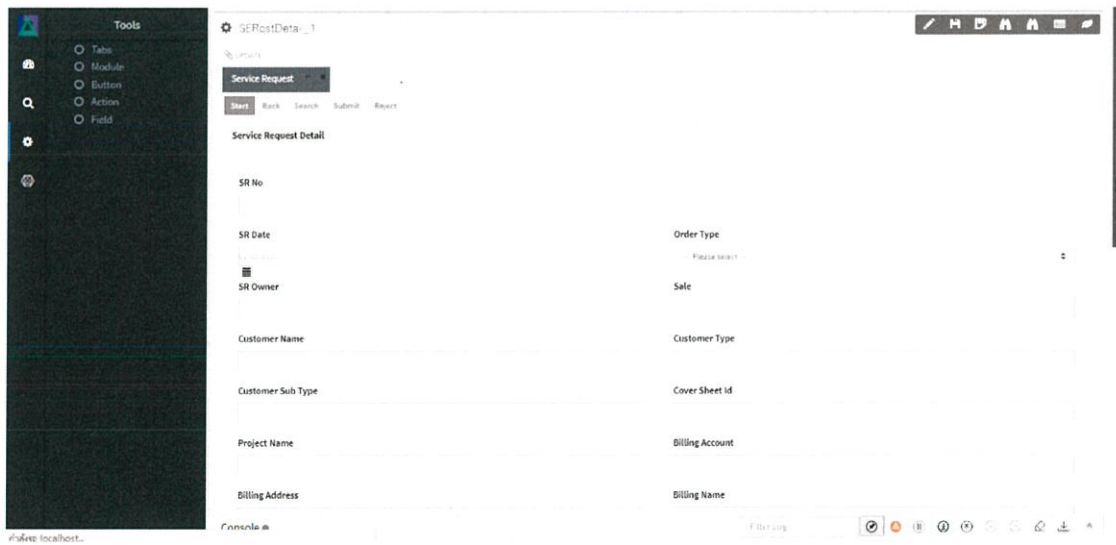
Default insert: N

Require workflow: N

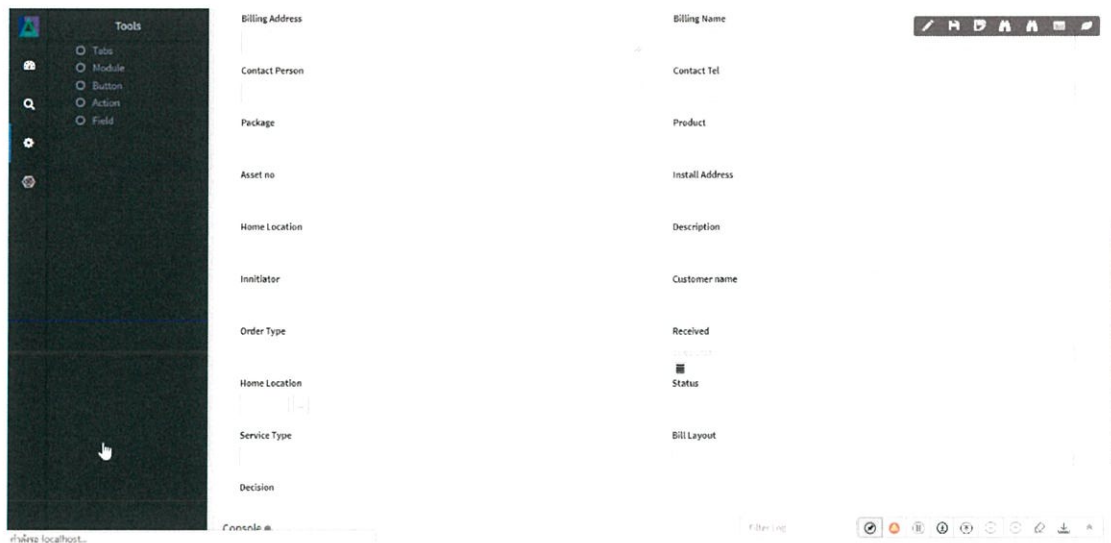
OK Cancel

ภาพที่ 3.19 Entity Configuration ตั้งชื่อ Entity

5) จากนั้นทำการลากวางปุ่มและฟอร์มต่างๆ ตามแบบร่างหน้าจอ



ภาพที่ 3.20 การลากวางปุ่มและฟอร์มต่างๆ เพื่อสร้างหน้า User Interface (1)



ภาพที่ 3.21 การลากวางปุ่มและฟอร์มต่างๆ เพื่อสร้างหน้า User Interface (2)

6) กด Save Entity



จากนั้นทำการสร้างหน้าต่างต่างๆ ที่เหลือจนครบ ขั้นตอนต่อไปจะเป็นการเขียนโปรแกรมเพิ่มเติมในส่วนที่ ONEWEB ไม่สามารถทำได้

3.3.2 เขียน Reject popup

เมื่อกดปุ่ม Reject แล้วให้เปิด popup เพื่อให้ระบุเหตุผลในการ Reject และเมื่อกด Confirm ในหน้า popup ให้ทำการบันทึกค่าสถานะ 'Reject' ลงใน attribute DECISION ในฐานข้อมูล

3.3.2.1 ส่วนติดต่อกับฐานข้อมูล

เราจะใช้ DAO (Data Access Object) ในการติดต่อกับฐานข้อมูล ซึ่ง DAO คือ Object ที่มีหน้าที่ในการเข้าถึง (Access) ข้อมูล เพราะฉะนั้นภายในคลาสนี้จะประกอบไปด้วยเมธอดที่ใช้ในการเข้าถึงข้อมูลเท่านั้น [18]

เราใช้ DAO เพื่อแยก Data Layer ออกจาก Business Layer ให้ชัดเจน จะทำให้การพัฒนาของเราดูง่ายขึ้นและสะดวกต่อการใช้งาน หรือจะกล่าวอีกอย่างหนึ่งว่า DAO จะคอยเป็นตัวกลางในการเชื่อมข้อมูลระหว่าง ฐานข้อมูลเรากับหน้าจอที่ใช้ในการแสดงผลข้อมูล

สร้าง DAO interface ชื่อ ManualOrderDAO มีเมธอดดังนี้

```
ManualOrderDAO.java
1 package com.manual.order.dao;
2
3 import org.json.JSONArray;
4
5 public interface ManualOrderDAO {
6     public JSONArray loadMSRejetReason() throws Exception;
7
8 }
```

ภาพที่ 3.22 เขียน DAO interface

ต่อมาสร้างไฟล์ชื่อว่า ManualOrderPostgresDAOImpl.java เพื่อทำการ implements interface ManualOrderDAO.java มาเขียนโค้ดการทำงานจริง

```
ManualOrderPostgresDAOImpl.java
1 package com.manual.order.dao;
2
3 import java.sql.Connection;
4
5 public class ManualOrderPostgresDAOImpl extends ManualObjectDAO implements ManualOrderDAO {
6
7     Logger logger = Logger.getLogger(ManualOrderPostgresDAOImpl.class);
8
9     @Override
10    public JSONArray loadMSRejetReason() throws Exception {
11        JSONArray resultAll = new JSONArray();
12        Connection conn = null;
13        PreparedStatement ps = null;
14        ResultSet rs = null;
15        try{
16            StringBuilder sql = new StringBuilder();
17            sql.append("select reason_id, reason_desc ");
18            sql.append("from order_ms_reject_reason");
19            logger.debug("#####"+sql);
20
21            String dSql = String.valueOf(sql);
22
23            conn = ManualJDBCServiceLocator.getInstance().getConnection(ManualJDBCServiceLocator.ORDER_DB);
24            ps = conn.prepareStatement(dSql);
25            rs = ps.executeQuery();
26
27            while(rs.next()){
28                JSONObject result = new JSONObject();
29                result.put("reason_id", rs.getString("reason_id"));
30                result.put("reason_desc", rs.getString("reason_desc"));
31                resultAll.put(result);
32            }
33        }catch(Exception e){
34            e.printStackTrace();
35            throw e;
36        }finally{
37            closeConnection(conn, ps, rs);
38        }
39        return resultAll;
40    }
41 }
```

ภาพที่ 3.23 เขียนโค้ดเพื่อติดต่อกับฐานข้อมูลและการทำงานของเมธอด loadMSRejetReason()

ในเมธอด loadMSRejectReason() จะทำการเชื่อมต่อกับฐานข้อมูลและเขียน SQL เพื่อทำการ query ข้อมูลใน attribute reason_id และ reason_desc จากตาราง order_ms_reject_reason

```
28     StringBuilder sql = new StringBuilder();
29     sql.append("select reason_id, reason_desc ");
30     sql.append("from order_ms_reject_reason");
31     logger.debug("#####"+sql);
32
33     String dSql = String.valueOf(sql);
34
35     conn = ManualJDBCServiceLocator.getInstance().getConnection(ManualJDBCServiceLocator.ORDER_DB);
36     ps = conn.prepareStatement(dSql);
37     rs = ps.executeQuery();
```

จากนั้น ทำการเก็บผลลัพธ์ให้อยู่ในรูปแบบ JSONObject และ return เป็น JSONArray

```
39     while(rs.next()){
40         JSONObject result = new JSONObject();
41         result.put("reason_id", rs.getString("reason_id"));
42         result.put("reason_desc", rs.getString("reason_desc"));
43         resultAll.put(result);
44     }
```

3.3.2.2 เขียน Servlet เพื่อให้บริการแก่ผู้ใช้ (Client) ตามคำร้องขอ

(Request)

```
1 package com.manual.order.ajax;
2
3 import javax.servlet.http.HttpServletRequest;
4
5 public class AjaxServletRejectPopup implements ManualInterface {
6     HttpServletRequest request;
7     Logger logger = Logger.getLogger(AjaxServletRejectPopup.class);
8
9     @Override
10    public String processManual() {
11        String entityID = (String)getRequest().getSession().getAttribute("entityID");
12        String entitySession = entityID+"_session";
13        EntityFormHandler entityForm = (EntityFormHandler)getRequest().getSession().getAttribute(entitySession);
14        String moduleID = entityForm.getMainModuleID();
15        logger.info("-----processManual test-----"+moduleID);
16
17        StringBuilder popHtml = new StringBuilder();
18        try{
19            popHtml.append("<div class='boxboldbelow box-body'>");
20            popHtml.append("<div class='panel-subheading02' style='width:600px;'>");
21            popHtml.append("<div class='subheading'>Service Request Reject</div>");
22            popHtml.append("</div>");
23            popHtml.append("<div class='"+moduleID+"-content-header'></div>");
24            //End P-Subhead
25
26            popHtml.append("<div id='"+moduleID+"-content' style='width:100%'>");
27            popHtml.append("<div class='form-group-sm textblock1' id='"+moduleID+"boxField_0'>");
28            popHtml.append("<div class='field-padding'>");
29            popHtml.append("<div class='col-xs-12' id='"+moduleID+"boxLeft_0'>");
30            popHtml.append("<div class='col-sm-2' id='"+moduleID+"_REJECT_REASON_LabelField'><label>Reason </label></div>");
31            popHtml.append("<div class='col-sm-10 margintable' id='"+moduleID+"_REJECT_REASON_InputField'>");
32
33            //start select Reject Reason
34            popHtml.append("<select class='form-control listboxtype btn btn-default dropdown' name='REJECT_REASON'>");
35            //loop JSON
36            JSONObject result = new JSONObject();
37            JSONArray rejReasonArray = ManualDAOFactory.getManualOrderDAO().loadMSRejetReason();
38            popHtml.append("<option value=''>--Please Select--</option>");
39            logger.info("-----rejReasonArray>>>"+ rejReasonArray);
40
41            for (int i=0; i<rejReasonArray.length(); i++){
42                result = rejReasonArray.getJSONObject(i);
43                String reason_id = result.getString("reason_id");
44                String reason_desc = result.getString("reason_desc");
45
46                popHtml.append("<option value='"+reason_id+"'> "+reason_desc+ " </option>");
47            }
48
49            popHtml.append("</select>");
50            popHtml.append("</div>");
51            //end select Reject Reason
52            popHtml.append("</div>");
53            popHtml.append("</div>");
54
55            //start text area Reject Remark
56            popHtml.append("<div class='form-group-sm textblock1' id='"+moduleID+"boxField_1'>");
57            popHtml.append("<div class='field-padding'>");
58            popHtml.append("<div class='col-xs-12' id='"+moduleID+"boxLeft_1' style='overflow: hidden; min-height: 30px;'>");
59            popHtml.append("<div class='col-sm-2' id='"+moduleID+"_REMARK_LabelField'><label>Remark </label></div>");
60            popHtml.append("<div class='col-sm-10 margintable' id='"+moduleID+"_REMARK_InputField'> ");
61            popHtml.append("<textarea name='REMARK' cols='20' rows='4' id='"+moduleID+"_117951338' class='form-control textinput-boxarea2'>");
62            popHtml.append("</textarea>");
63            popHtml.append("</div>");
64            popHtml.append("</div>");
65            popHtml.append("</div>");
66            popHtml.append("</div>");
67            popHtml.append("</div>");
68            popHtml.append("</div>");
69
70        }catch(Exception e){
71        }
72        return popHtml.toString();
73    }
74
75    @Override
76    public void setRequest(HttpServletRequest arg0) {
77        // TODO Auto-generated method stub
78        this.request = arg0;
79    }
80
81    public HttpServletRequest getRequest() {
82        return this.request;
83    }
84
85 }
```

ภาพที่ 3.24 ไฟล์ AjaxServletRejectPopup.java

ในการที่จะกำหนดให้ Popup แสดงที่หน้า Entity ไหนบ้างนั้น เราจำเป็นต้องระบุเลข Entity ID โดยเราจะเขียนโค้ดเพื่อดึงค่า Entity ID จากหน้าจอเลย เพื่อให้ใช้ได้กับหน้า Entity หลายๆหน้า

```

24 String entityID = (String)getRequest().getSession().getAttribute("entityID");
25 String entitySession = entityID+"_session";
26 EntityFormHandler entityForm = (EntityFormHandler)getRequest().getSession().getAttribute(entitySession);
27 String moduleID = entityForm.getMainModuleID();

```

โค้ดบรรทัด 24 เป็นการดึงค่า Entity ID จากหน้าจอบอก มาเก็บไว้ในตัวแปร entityID

โค้ดบรรทัด 26 เป็นการใช้คลาส EntityFormHandler เพื่อที่เราจะสามารถดึงข้อมูลที่อยู่ในระดับ Entity มาใช้งานได้

โค้ดบรรทัด 27 ดึงค่า Module ID หลัก มาเก็บไว้ในตัวแปร moduleID

นอกจากนี้ เราทำการเขียน Popup โดยใช้ภาษา HTML และ return HTML ออกไปเป็น String (บรรทัดที่ 30-87)

และเมื่อดูจากแบบร่างหน้าจอของ Reject Popup แล้ว เมื่อคลิกที่ Dropdown List ของช่อง Reason จะต้องแสดงข้อมูลเหตุผลในการ Reject จากฐานข้อมูล

ดังนั้น เราจึงเรียกใช้เมธอด loadMSRejectReason() ที่เขียนไว้ในไฟล์

ManualOrderPostgresDAOImpl.java

```

46 //start select Reject Reason
47 popHtml.append("<select class='form-control listboxtype btn btn-default dropdown' name='REJECT_REASON'>");
48 //loop JSON
49 JSONObject result = new JSONObject();
50 JSONArray rejReasonArray = ManualDAOFactory.getManualOrderDAO().loadMSRejectReason();
51 popHtml.append("<option value=''>--Please Select--</option>");
52 logger.info("-----rejReasonArray>>>"+ rejReasonArray);
53
54 for(int i=0; i<rejReasonArray.length(); i++){
55     result = rejReasonArray.getJSONObject(i);
56     String reason_id = result.getString("reason_id");
57     String reason_desc = result.getString("reason_desc");
58
59     popHtml.append("<option value='"+reason_id+"'> "+reason_desc+" </option>");
60 }
61
62 popHtml.append("</select>");
63 popHtml.append("</div>");
64 //end select Reject Reason

```

โค้ดบรรทัด 50 เรียกใช้เมธอด loadMSRejectReason() และเก็บค่าไว้ในตัวแปร rejReasonArray ซึ่งเป็น JSONArray

โค้ดบรรทัด 55 ใช้ for loop เพื่อดึงค่าที่เก็บเป็น JSONObject ออกมาเก็บไว้ในตัวแปร result

โค้ดบรรทัด 56 และ 57 ดึงค่าข้อมูลออกมาเป็น String

โค้ดบรรทัด 59 ใน <option></option> จะแสดงข้อมูลที่เก็บในตัวแปร reason_desc และมี value เท่ากับค่าที่เก็บไว้ในตัวแปร reason_id

3.3.2.3 เขียนไฟล์ JavaScript เพื่อให้แสดง Popup

```
order_new_service_request.js
1 $(document).ready(function(){
2     $('#name=masterForm').append('<div id="confirm_dialog"></div>');
3     $('#confirm_dialog').dialog({
4         width: 500,
5         autoOpen: false,
6         height: 'auto',
7         modal: true,
8         title: '',
9         close: function(ev, ui) {
10             $('body').css("overflow", "auto");
11         }
12     }).parent().appendTo('[name=masterForm]');
13 });
14
15 function confirmRejBtnAction(){
16     $('#confirm_dialog').dialog('close');
17     $('#MD1171634317-content [name="DECISION"]').val('Reject');
18     saveEntity();
19 }
20
21 function cancelRejBtnAction(){
22     $('#confirm_dialog').dialog('close');
23 }
24
25 function getReasonReject(){
26     var htmlBuild='';
27     var uri = "/MasterWeb/ManualServlet";
28     var dataString = "className=com.manual.order.ajax.AjaxServletRejectPopup";
29     dataString += "&functionName=processManual";
30     jQuery.ajax({
31         type : "POST",
32         url : uri,
33         data : dataString,
34         async : false,
35         success : function(data) {
36             console.log("-----data ###"+ data);
37             htmlBuild +=data;
38             htmlBuild += '<div style="text-align:center;margin-top:25px">'
39                 + '<input class="btn btn-success" style="margin-right:10px" '
40                 + 'type="button" id="confirmBtn" value="Confirm" onclick="confirmRejBtnAction()">'
41                 + '<input class="btn btn-warning" type="button" id="cancelBtn" '
42                 + 'value="Cancel" onclick="cancelRejBtnAction()">'
43                 + '</div>';
44
45             $('#confirm_dialog').html(htmlBuild);
46             $('#confirm_dialog').dialog('open');
47         }
48     });
49 }
```

ภาพที่ 3.25 ไฟล์ order_new_service_request.js

โค้ดบรรทัด 1-13 เป็นการสร้าง dialog และแทรก <div id="confirm_dialog"></div> ไปใน Form ที่ name=masterForm

โค้ดบรรทัด 25 สร้าง function getReasonReject() ใช้ jQuery.ajax() เพื่อส่ง request ไปยัง AjaxServletRejectPopup.java

โค้ดบรรทัด 37 data ที่ได้รับ response กลับมา เป็น HTML ที่เป็น String เก็บในตัวแปร htmlBuild

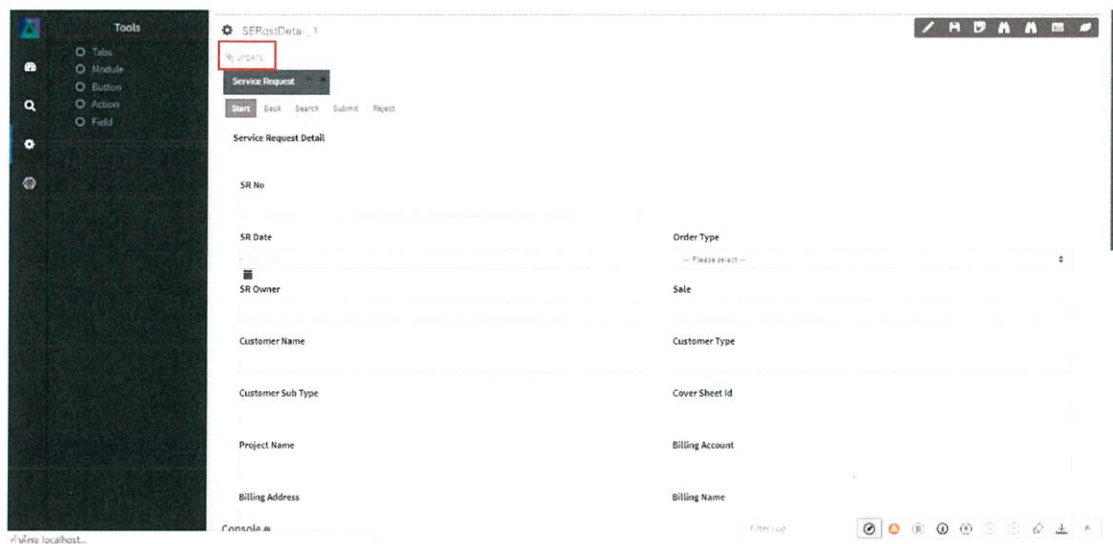
โค้ดบรรทัด 39-42 เป็นการเขียนปุ่ม Confirm และ Cancel เพิ่ม โดยเมื่อคลิกปุ่ม Confirm จะไปทำงานใน function confirmRejBtnAction() คือทำการปิด dialog และกำหนดค่าให้ field name="DECISION" มีค่าเป็น 'Reject' และ save ลงในฐานข้อมูล แต่ถ้าคลิกปุ่ม Cancel จะไปทำงานใน function cancelRejBtnAction() คือทำการปิด dialog

โค้ดบรรทัด 45 แปลง String ให้อยู่ในรูปแบบโครงสร้างภาษา HTML

โค้ดบรรทัด 46 ทำการเปิด dialog

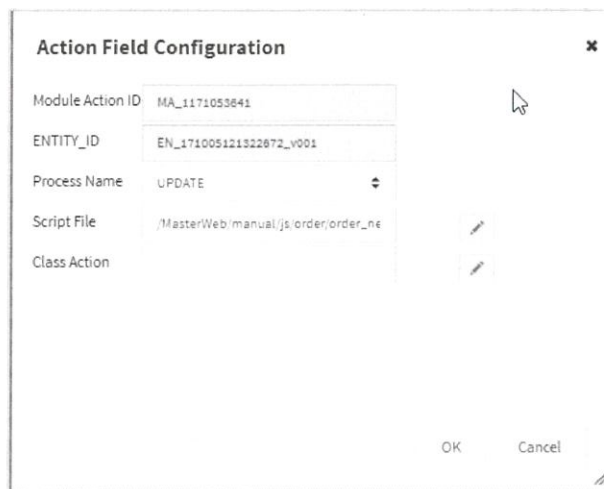
3.3.2.4 Insert ไฟล์ JavaScript ไปใน ONEWEB

กลับไป ONEWEB App Designer เปิดหน้า Entity ที่จะให้แสดง Reject Popup จากนั้นลาก Entity Action มาวาง



ภาพที่ 3.26 ทำการ configuration ปุ่ม Entity Action

คลิก  เพื่อทำการ insert ไฟล์ JavaScript เลือก Process Name เป็น UPDATE และช่อง Script File ใส่ /MasterWeb/manual/js/order/order_new_service_request.js



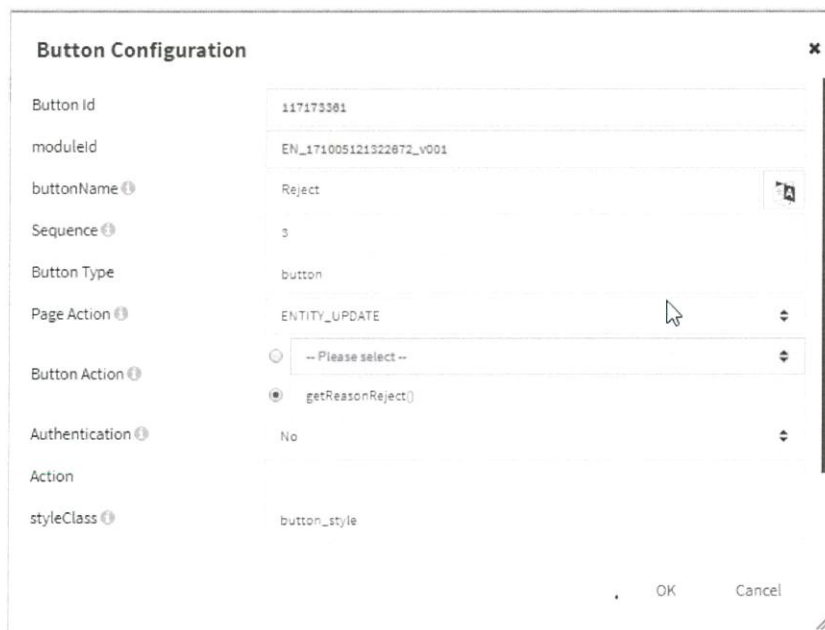
ภาพที่ 3.27 Action Field Configuration

3.3.2.5 กำหนด function ในปุ่ม Reject

กำหนด function ในปุ่ม Reject



ใส่ `getReasonReject()` ลงในช่อง Button Action



ภาพที่ 3.28 Button Configuration

หากต้องการให้แสดง Reject Popup ในหน้าอื่น ให้สร้างไฟล์ JavaScript ใหม่และทำการเปลี่ยนโค้ด ในไฟล์ JavaScript หัวข้อ 3.3.2.3 เปลี่ยนเลขโมดูล (บรรทัดที่ 17 ตรง #MD1171634317-content) ให้ตรงกับ Module ID ใน Entity นั้นๆ และทำซ้ำขั้นตอนในหัวข้อ 3.3.2.4 และ 3.3.2.5

3.3.3 เขียนโค้ดให้บันทึกค่าสถานะ 'Submit' ลงในฐานข้อมูล เมื่อกดปุ่ม Submit

3.3.3.1 เพิ่ม function submitBtnAction() ในไฟล์ JavaScript

ในไฟล์ JavaScript หัวข้อ 3.3.2.3 ให้เพิ่ม function submitBtnAction()

ลงไป

```
51 function submitBtnAction() {  
52     $('#confirm_dialog').dialog('close');  
53     $('#MD1171634317-content [name="DECISION"]').val('Submit');  
54     saveEntity();  
55 }
```

ภาพที่ 3.29 เพิ่ม function submitBtnAction() ในไฟล์ order_new_service_request.js

เมื่อคลิกปุ่ม Submit จะไปทำงานใน function submitBtnAction() คือทำการปิด dialog และกำหนดค่าให้ field name="DECISION" มีค่าเป็น 'Submit' และ save ลงในฐานข้อมูล

3.3.3.2 กำหนด function ในปุ่ม Submit

กำหนด function ในปุ่ม Submit ใส่ submitBtnAction() ลงในช่อง

Button Action

Field	Value
Button Id	1171732814
moduleId	EN_171005121322672_v001
buttonName	Submit
Sequence	2
Button Type	button
Page Action	ENTITY_UPDATE
Button Action	submitBtnAction()
Authentication	No
Action	
styleClass	button_style

ภาพที่ 3.30 Button Configuration

3.3.4 บันทึกรหัสเหตุการณ์ในการ reject ลงในฐานข้อมูล

3.3.4.1 สร้าง Java Class ชื่อ OrderNewServiceRequestSubmit

```
OrderNewServiceRequestSubmit.java
21 public Vector modifyResult() {
22     try{
23         String reason = getRequest().getParameter("REJECT_REASON");
24         String remark = getRequest().getParameter("REMARK");
25         logger.debug("##### reason---- : " + reason);
26         logger.debug("##### remark---- : " + remark);
27
28         String entityID = (String)getRequest().getSession().getAttribute("entityID");
29         String entitySession = entityID+"_session";
30         EntityFormHandler entityForm = (EntityFormHandler)getRequest().getSession().getAttribute(entitySession);
31         String moduleID = entityForm.getMainModuleID();
32
33         MasterFormHandler moduleForm = (MasterFormHandler)getRequest().getSession().getAttribute(moduleID+ "_session");
34         String tableName = moduleForm.getModuleM().getTableName();
35         String processMode = moduleForm.getProcessMode();
36
37         Vector<HashMap> data = getResultForProcess();
38         logger.debug("##### Vector data1 : " + data);
39
40         int indexOfMainModuleTable = this.getHashMapByModeAndTable(data, processMode, tableName);
41         HashMap mainModuleHash = data.get(indexOfMainModuleTable);
42         HashMap processModeHash = (HashMap)mainModuleHash.get(processMode);
43         HashMap tableHash = (HashMap)processModeHash.get(tableName);
44
45         tableHash.put("REJECT_REASON", reason);
46         tableHash.put("REMARK", remark);
47
48     }catch(Exception e){
49         e.printStackTrace();
50     }
51     return super.modifyResult();
52 }
53
54 private int getHashMapByModeAndTable(Vector<HashMap> processVect, String processMode, String tableName){
55     if(processVect != null && processVect.size() > 0){
56         for(int i=0; i<processVect.size(); i++){
57             HashMap recordHash = processVect.get(i);
58             if(recordHash.containsKey(processMode)){
59                 HashMap modeHash = (HashMap)recordHash.get(processMode);
60                 if(modeHash.containsKey(tableName)){
61                     return i;
62                 }
63             }
64         }
65     }
66     return 0;
67 }
```

ภาพที่ 3.31 เขียนเมธอด modifyResult() ในไฟล์ OrderNewServiceRequestSubmit.java

โค้ดบรรทัด 23 ดึงค่า “REJECT_REASON” จากหน้าจอ มาเก็บไว้ในตัวแปร reason

โค้ดบรรทัด 24 ดึงค่า “REMARK” จากหน้าจอ มาเก็บไว้ในตัวแปร remark

โค้ดบรรทัด 28 เป็นการดึงค่า Entity ID จากหน้าจอ มาเก็บไว้ในตัวแปร entityID

โค้ดบรรทัด 30 เป็นการใช้คลาส EntityFormHandler เพื่อที่เราจะสามารถดึงข้อมูลที่อยู่ในระดับ Entity มาใช้งานได้

โค้ดบรรทัด 31 ดึงค่า Module ID หลัก มาเก็บไว้ในตัวแปร moduleID

โค้ดบรรทัด 33 เป็นการใช้คลาส MasterFormHandler เพื่อที่เราจะสามารถดึงข้อมูลที่อยู่ในระดับ Module มาใช้งานได้

โค้ดบรรทัด 34 ดึงชื่อตารางที่ Module หลักใช้ (ตารางในฐานข้อมูล)

โค้ดบรรทัด 35 ดึงโหมดการทำงานของหน้าจอ เช่น INSERT หรือ UPDATE

โค้ดบรรทัด 37 ดึงข้อมูลจากหน้าจอ มาเก็บเป็น Vector

```
[{   โหมดการทำงาน
  UPDATE = {
    MODULE_FIELD_KEY = [REQUEST_NO, INSTALL_ADDRESS_ID],
    ORDER_REQUEST_INSTALL_ADDRESS = {
      INSTALL_ADDRESS_ID = 000000006,   ชื่อตาราง
      UPDATE_DATE = 2017 - 11 - 28,
      ROAD = ลาดพร้าว,
      ...
    }
  }
}, {
  UPDATE = {
    MODULE_FIELD_KEY = [REQUEST_NO, INSTALL_ADDRESS_ID],
    ORDER_REQUEST_INSTALL_ADDRESS = {
      INSTALL_ADDRESS_ID = 000000008,
      UPDATE_DATE = 2017 - 11 - 28,
      ROAD = สาทรเหนือ,
      ...
    }
  }
}, {
  UPDATE = {
    ORDER_SERVICE_REQUEST = {
      REQUEST_NO = SR - 601002 - 0000007,
      DECISION = Reject,
      UPDATE_DATE = 2017 - 11 - 28,
      UPDATE_BY = ,
      CA_ACCOUNT = CA - 000000 - 000000
    },
    MODULE_FIELD_KEY = [REQUEST_NO]
  }
}]
```

ภาพที่ 3.32 ตัวอย่างรูปแบบการเก็บข้อมูลแบบ Vector

โค้ดบรรทัด 40 เรียกใช้เมธอด `getHashMapByModeAndTable()` เพื่อทำการวนลูปหา index ของ HashMap ของตารางที่ Module หลักใช้

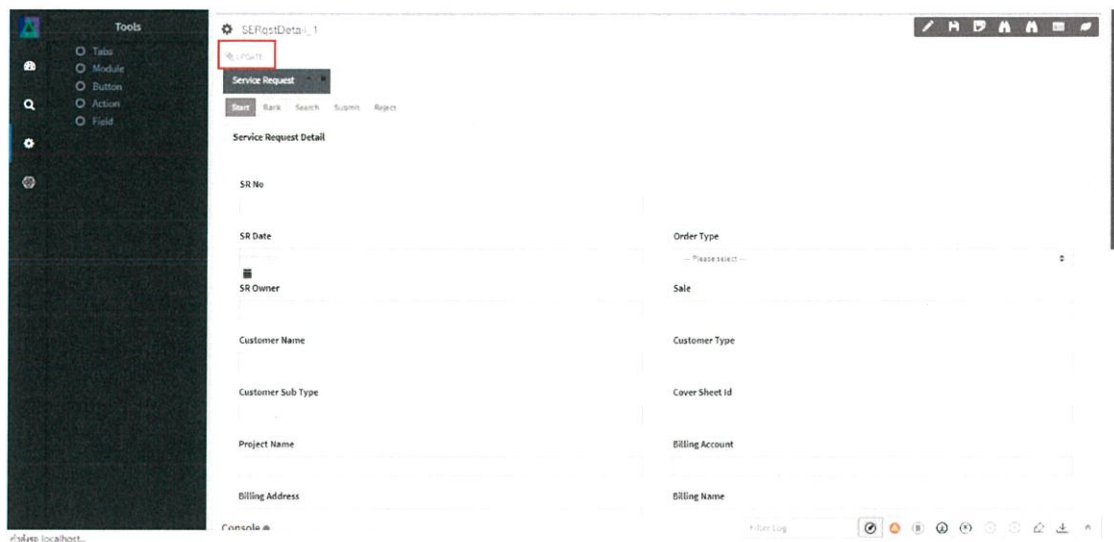
โค้ดบรรทัด 43 เข้าถึง HashMap ของตาราง

โค้ดบรรทัด 45-46 เพิ่มค่า reason และ remark เข้าไปใน HashMap

3.3.4.2 Insert ไฟล์ Java ไปใน ONEWEB

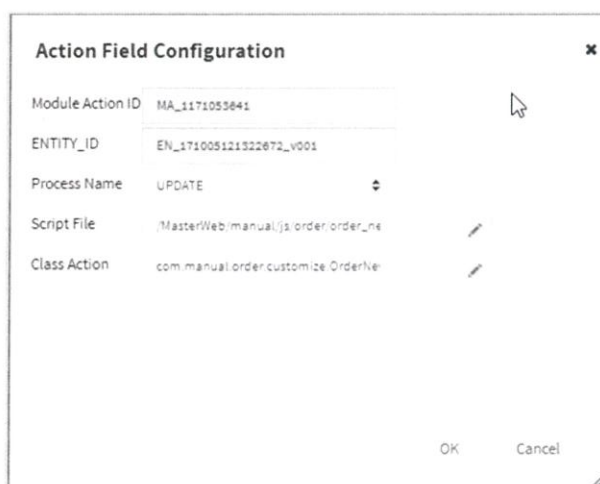
กลับไปที่ ONEWEB App Designer เปิดหน้า Entity ที่จะให้แสดง Reject

Popup คลิกแก้ไข Entity Action



ภาพที่ 3.33 ทำการ configuration ปุ่ม Entity Action

ช่อง Class Action ใส่ `com.manual.order.customize.OrderNewServiceRequestSubmit`

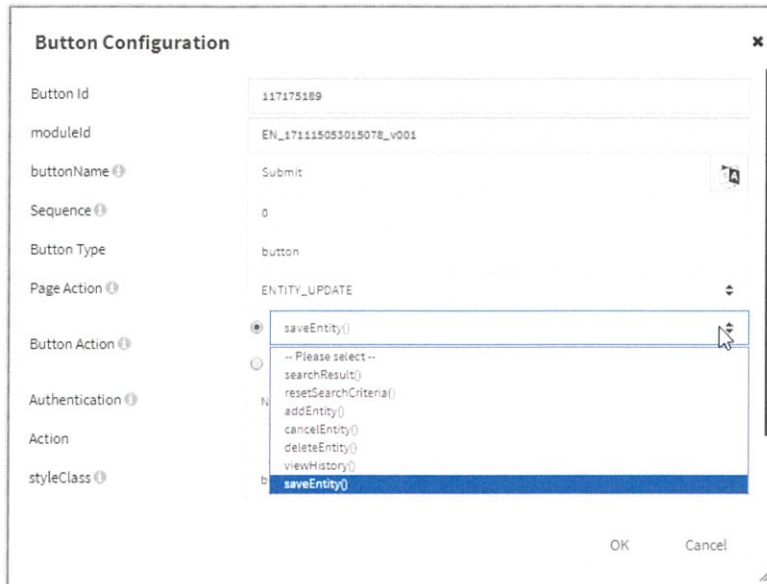


ภาพที่ 3.34 Action Field Configuration

สำหรับ Entity อื่น ที่ต้องการให้มีการทำงานเช่นเดียวกัน ให้ทำการสร้างไฟล์ Java ใหม่ เขียนโค้ดตามหัวข้อ 3.3.4.1 และ insert ไฟล์ ตามหัวข้อ 3.3.4.2

3.3.5 การ Redirect screen

โดยปกติการ Configuration Button พื้นฐานของ ONEWEB จะมีการทำงานให้เลือกอยู่ 7 functions ซึ่งทุก function เมื่อทำงานเสร็จจะกลับไปยังหน้า Search ของ Entity ตัวเอง



ภาพที่ 3.35 Function พื้นฐานของ Button

แต่ในที่นี้ ต้องการให้ เมื่อกดปุ่มและจบการทำงาน ให้มันทำการเปลี่ยนไปหน้าอื่น ที่ไม่ใช่หน้า Search ของตัวมันเอง

3.3.5.1 เพิ่มเมธอด dynamicURL()

เขียนเมธอด dynamicURL() เพิ่มเข้าไปใน Java Class

OrderNewServiceRequestSubmit ในหัวข้อ 3.3.4.1

```
public String dynamicURL() {
    String entityID = (String)getRequest().getSession().getAttribute("entityID");
    EntityFormHandler entityForm = (EntityFormHandler)getRequest().getSession().getAttribute(entityID + "_session");
    String moduleID = entityForm.getMainModuleID();
    String requestNo = (String)EAFManualUtil.getDataHashMapFromSession(moduleID, request).get("REQUEST_NO");
    String decision = (String)EAFManualUtil.getDataHashMapFromSession(moduleID, request).get("DECISION");
    logger.debug("##### dynamicURL requestNo : " + requestNo);
    logger.debug("##### dynamicURL decision : " + decision);
    if("Submit".equals(decision)){
        return "/MasterWeb/FrontController?action=loadUpdateEntity&entityID="
            + "EN_6137580665_v001&tabID=TAB_ID_7557575452&REQUEST_NO=" + requestNo;
    }else if("Reject".equals(decision)){
        return "/MasterWeb/FrontController?action=loadSearchEntity&entityID="
            + "EN_171005121322672_v001&tabID=TAB_ID_1171634765&searchForUpdate=Y";
    }else{
        return super.dynamicURL();
    }
}
```

ภาพที่ 3.36 เมธอด dynamicURL() ใน Class OrderNewServiceRequestSubmit

ใน if...else จะทำการเช็คค่า ถ้า field DECISION มีค่าเท่ากับ “Submit” (มีค่าเท่ากับ “Submit” เมื่อกดปุ่ม Submit) จะทำการเปลี่ยนไปหน้าถัดไปคือ Entity ID = EN_6137580665_v001 (Entity: SERqstDetailNew_2, Screen: Input Install Address)

ถ้า field DECISION มีค่าเท่ากับ “Reject” (มีค่าเท่ากับ “Reject” เมื่อกดปุ่ม Confirm ใน Reject popup) จะทำการเปลี่ยนกลับไปหน้าแรกสุดเสมอคือ Entity ID = EN_171005121322672_v001 (Entity: SERqstDetail_1, Scree: ServiceRequest)

ไฟล์ Java ที่ยกตัวอย่างข้างต้น (ภาพที่ 3.31, 3.36) เป็นไฟล์ที่ใช้กับ Entity SERqstDetail_1 (Screen: ServiceRequest) ถ้าหากจะใช้กับ Entity อื่น จะต้องเปลี่ยน URL ที่ return ด้วย ตัวอย่างเช่น

```
public String dynamicURL() {
    String entityID = (String)getRequest().getSession().getAttribute("entityID");
    EntityFormHandler entityForm = (EntityFormHandler)getRequest().getSession().getAttribute(entityID + "_session");
    String moduleID = entityForm.getMainModuleID();
    String requestNo = (String)EAFManualUtil.getDataHashMapFromSession(moduleID, request).get("REQUEST_NO");
    String decision = (String)EAFManualUtil.getDataHashMapFromSession(moduleID, request).get("DECISION");
    logger.debug("@@@@@ dynamicURL requestNo : " + requestNo);
    logger.debug("@@@@@ dynamicURL decision : " + decision);
    if("Submit".equals(decision)){
        return "/MasterWeb/FrontController?action=loadUpdateEntity&entityID="
            + "EN_3790213085_v001&tabID=TAB_ID_117159502&REQUEST_NO=" + requestNo;
    }else if("Reject".equals(decision)){
        return "/MasterWeb/FrontController?action=loadSearchEntity&entityID="
            + "EN_171005121322672_v001&tabID=TAB_ID_1171634765&searchForUpdate=Y";
    }else{
        return super.dynamicURL();
    }
}
```

ภาพที่ 3.37 เมธอด dynamicURL() ใน Class OrderNewRequestDetailSubmit

ไฟล์ในภาพที่ 3.37 ใช้กับ Entity SERqstDetailNew_2 (Screen: Input Install Address) ถ้า field DECISION มีค่าเท่ากับ “Submit” จะทำการเปลี่ยนไปหน้าถัดไปคือ EN_3790213085_v001 (Entity: SEPackageOffering_3, Screen: Package Offering) ส่วนค่า “Reject” จะทำงานเหมือนกัน

ยกเว้นเมธอด dynamicURL() ใน Entity สุดท้าย SEReqCreateQuote_6 (Screen: Create Quote) คือไม่ว่า field DECISION มีค่าเท่ากับ “Submit” หรือ “Reject” ก็จะกลับไปหน้าแรกสุดเสมอ

ส่วนการกดปุ่ม Back แล้วให้ย้อนกลับไปยังหน้าก่อนหน้า จะเขียนในไฟล์ JavaScript

```

function loadForBack() {
    blockScreen();
    windowFocus();
    //Find service request number from current entity
    var requestNo = $('#MD1538762421-content [name="REQUEST_NO"]').val();
    //Load previous entity with service request number
    window.location = "/MasterWeb/FrontController?action=loadUpdateEntity&entityID=" +
        "EN_171005121322672_v001&tabID=TAB_ID_1171634765&REQUEST_NO=" + requestNo;
}

```

ภาพที่ 3.38 function loadForBack()

3.3.6 Validate record

3.3.6.1 ตรวจสอบว่าต้องมีการระบุ Address อย่างน้อย 1 record

ต้องระบุ Address อย่างน้อย 1 record ถึงจะกดปุ่ม Submit ได้ (ใน

Entity SERqustDetailNew_2, Screen: Input Install Address)

เขียนเมธอด validateResult() ใน Java Class ของ Entity

SERqustDetailNew_2

```

public boolean validateResult() {
    boolean result = true;
    String entityID = (String)getRequest().getSession().getAttribute("entityID");
    String entitySession = entityID+"_session";
    String nextEntity = request.getParameter("nextEntity");
    logger.debug("##### nextEntity : " + nextEntity);
    if(nextEntity == null || "".equals(nextEntity)){
        EntityFormHandler entityForm = (EntityFormHandler)getRequest().getSession().getAttribute(entitySession);
        Vector vError = entityForm.getFormErrors();
        String moduleID = entityForm.getMainModuleID();
        String decision = (String)EAFManualUtil.getDataHashMapFromSession(moduleID, request).get("DECISION");

        MasterFormHandler moduleForm = (MasterFormHandler)request.getSession().getAttribute("MD4223903684_session");
        if(moduleForm.getStoreActionList().size()==0 && "Submit".equals(decision)){
            vError.add("Require at least 1 Install Address");
            result = false;
        }
    }
    return result;
}

```

ภาพที่ 3.39 เมธอด validateResult() เพื่อ Validate record

เนื่องจากการตรวจสอบนี้ เป็นการตรวจสอบที่โมดูลกลาง การใช้คลาส MasterFormHandler จึงต้องระบุเซสชันโมดูลของโมดูลข้างล่าง ไม่ใช่โมดูลหลัก

```
if(moduleForm.getStoreActionList().size()==0 && "Submit".equals(decision)){
    vError.add("Require at least 1 Install Address");
    result = false;
}
```

If function นี้ จะตรวจสอบ list (record) ในโมดูลข้างล่าง ถ้าหากมี size เป็น 0 (ไม่มี record เลย) และ field DECISION มีค่าเท่ากับ "Submit" จะแจ้งข้อความเตือนว่า "Require at least 1 Install Address"

3.3.6.2 ตรวจสอบว่าต้องมีการระบุอย่างน้อย 1 package

ต้องระบุ Package อย่างน้อย 1 record ถึงจะกดปุ่ม Submit ได้ (ใน Entity SEPackageOffering_3, Screen: Package Offering)

Package Name	Product Name	Price	Period Date	Promotion Bill Cycle	Quantity
บริการ Strong Life Up to 40Mb/20Mb_Fixed Line	Normal Telephone	500.00	01/01/2011 - 30/04/2018	1	1
	General Discount Product(n)	0	01/01/2011 - 30/04/2018	1	1
	Fiber 2U(n)	590.00	01/01/2011 - 30/04/2018	1	1

เขียนเมธอด validateResult() ใน Java Class ของ Entity SEPackageOffering_3 คล้ายกับที่เขียนในหัวข้อ 3.3.6.1 แต่เปลี่ยนเซสชันโมดูลตอนเรียกใช้คลาส MasterFormHandler

```

public boolean validateResult() {
    boolean result = true;
    String entityID = (String)getRequest().getSession().getAttribute("entityID");
    String entitySession = entityID+"_session";
    String nextEntity = request.getParameter("nextEntity");
    logger.debug("##### nextEntity : " + nextEntity);
    if(nextEntity == null || "".equals(nextEntity)){
        EntityFormHandler entityForm = (EntityFormHandler)getRequest().getSession().getAttribute(entitySession);
        Vector vError = entityForm.getFormErrors();
        String moduleID = entityForm.getMainModuleID();
        String decision = (String)EAFManualUtil.getDataHashMapFromSession(moduleID, request).get("DECISION");

        MasterFormHandler moduleForm = (MasterFormHandler)request.getSession().getAttribute("MD1171510917_session");
        if(moduleForm.getStoreActionList().size()==0 && "Submit".equals(decision)){
            vError.add("Require at least 1 Package Offering");
            result = false;
        }
    }
    return result;
}

```

ภาพที่ 3.40 เมธอด validateResult() เพื่อ Validate record

หากตรวจสอบแล้วไม่มี record เลย และ field DECISION มีค่าเท่ากับ "Submit" จะแจ้งข้อความเตือนว่า "Require at least 1 Package Offering"

3.3.7 Validate field

3.3.7.1 ต้องระบุ Address ให้ครบทุก Package

ต้องระบุ Address ให้ครบทุก Package ถึงจะกดปุ่ม Submit ได้ (ใน Entity Mapping_product, Screen: Mapping Product)

Service Request Detail

SR No	<input type="text" value="SR-222222-222222"/>	SR Date	<input type="text" value="02/09/2017"/>
Order Type	<input type="text" value="สมัครใหม่ (New)"/>	Customer Type	<input type="text" value="ลูกค้าบ้านพักอาศัยประเภททั่วไป"/>
CA Account	<input type="text" value="CA-000000-000000"/>	Customer Name	<input type="text" value="นายจักรพันธ์ นฤวรพันธ์"/>
Billing Account	<input type="text" value="123456789012"/>	Billing Name	<input type="text" value="นายจักรพันธ์ นฤวรพันธ์"/>
Billing Address	<input type="text" value="1/150 อาคารพาณิชย์ สาทรเหนือ สยาม สาทร กรุงเทพฯ 10500"/>		

Package/Product - Install Address

No.	Package Name	Product Name	Install Address
1	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	Normal Telephone	1/150 อาคารพาณิชย์ สาทรเหนือ สยาม สาทร กรุงเทพฯ 10500
2	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	General Discount Product(n)	
3	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	Fiber 2U(n)	11/229 เคหะ วิลล่า ริมเกล้า 11 แสวนแสบ มินบุรี กรุงเทพฯ 10510

เขียนเมธอด validateResult() ใน Java Class ของ Entity

Mapping_product

Service Request Detail

SRNo:
 SR Date:

Order Type:
 Customer Type:

CA Account:
 Customer Name:

Billing Account:
 Billing Name:

Billing Address:

Package/Product - Install Address

No.	Package Name	Product Name	Install Address
1	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	Normal Telephone	1/150 อาคารพาณิชย์ สหกรณ์อสังหาริมทรัพย์ สหกรณ์ อสังหาริมทรัพย์ 10500
2	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	General Discount Product(n)	
3	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	Fiber 2U(n)	11/229 เคหะชวลิต ริมเกล้า 11 แสงแสน มินนิจิ กรุงเทพ 10510

Popup

Package/Product - Install Address

Package Name:

Product Name:

Install Address:

ตรงช่อง Install Address ในหน้า popup จะต้องแสดงเฉพาะ Address ที่ยังไม่ถูกเลือกเท่านั้น และถ้าหากไปกดแก้ไขตรง record ที่มีการระบุ Address อยู่แล้ว ช่อง Install Address ในหน้า popup จะต้องแสดง Address ที่ยังไม่ถูกเลือก และ Address เดิมของตัวเองด้วย

3.3.8.1 ส่วนติดต่อกับฐานข้อมูล

ใน DAO interface ชื่อ ManualOrderDAO ประกาศเมธอด

loadMSInstallAddress(String request_no) เพิ่ม

```

ManualOrderDAO.java
1 package com.manual.order.dao;
2
3 import org.json.JSONArray;
4
5 public interface ManualOrderDAO {
6     public JSONArray loadMSRejetReason() throws Exception;
7     public JSONArray loadMSInstallAddress(String request_no) throws Exception;
8 }

```

ภาพที่ 3.42 ประกาศเมธอดเพิ่มใน DAO interface

และในไฟล์ชื่อว่า ManualOrderPostgresDAOImpl.java เขียนการ

ทำงานของเมธอด loadMSInstallAddress(String request_no) เพิ่มอีก

```
ManualOrderPostgresDAOImpl.java  OrderNewMappingProductSubmit.java  AjaxServletInstallAddress.java  order_new_mapping_product.js
57 public JSONArray loadMSInstallAddress(String request_no) throws Exception {
58     JSONArray resultAll = new JSONArray();
59     Connection conn = null;
60     PreparedStatement ps = null;
61     ResultSet rs = null;
62     try{
63         StringBuilder sql = new StringBuilder();
64         sql.append("select install_address_id,");
65         sql.append("address_detail ");
66         sql.append("from vw_request_install_address ");
67         sql.append("where ");
68         sql.append("request_no = ");
69         sql.append(request_no);
70         sql.append("");
71         logger.debug("#####"+sql);
72
73         String dSql = String.valueOf(sql);
74
75         conn = ManualJDBCServiceLocator.getInstance().getConnection(ManualJDBCServiceLocator.ORDER_DB);
76         ps = conn.prepareStatement(dSql);
77         rs = ps.executeQuery();
78
79         while(rs.next()){
80             JSONObject result = new JSONObject();
81             result.put("install_address_id", rs.getString("install_address_id"));
82             result.put("address_detail", rs.getString("address_detail"));
83             resultAll.put(result);
84         }
85
86     }catch(Exception e){
87         e.printStackTrace();
88         throw e;
89     }finally{
90         closeConnection(conn, ps, rs);
91     }
92     return resultAll;
93 }
94 }
```

ภาพที่ 3.43 เขียนโค้ดเพื่อติดต่อกับฐานข้อมูลและการทำงานของเมธอด

loadMSInstallAddress(String request_no)

ในเมธอด loadMSInstallAddress(String request_no) จะทำการเชื่อมต่อกับฐานข้อมูลและเขียน SQL เพื่อทำการ query ข้อมูลใน attribute install_address_id และ address_detail ที่มี request_no = request_no ที่รับเข้ามา

จากนั้น ทำการเก็บผลลัพธ์ให้อยู่ในรูปแบบ JSONObject และ return เป็น JSONArray

3.3.8.2 เขียน Servlet เพื่อติดต่อกับ DAO

```
ManualOrderPostgresDAOImpl.java AjaxServletInstallAddress.java order_new_mapping_product.js OrderNewMappingProductSubmit.java
1 package com.manual.order.ajax;
2
3 import java.util.HashMap;
4
18
19 public class AjaxServletInstallAddress implements ManualInterface {
20     HttpServletRequest request;
21     Logger logger = Logger.getLogger(AjaxServletRejectPopup.class);
22
23     @Override
24     public String processManual() {
25         String rowID = request.getParameter("rowID");
26         String entityID = (String)request.getSession().getAttribute("entityID");
27         String entitySession = entityID+"_session";
28         EntityFormHandler entityForm = (EntityFormHandler)request.getSession().getAttribute(entitySession);
29         String moduleID = entityForm.getMainModuleID();
30         String requestNo = (String)EAFManualUtil.getDataHashMapFromSession(moduleID, request).get("REQUEST_NO");
31
32         MasterFormHandler moduleForm = (MasterFormHandler)request.getSession().getAttribute("MD1171858697_session");
33         Vector<HashMap> dataV = moduleForm.getStoreActionList();
34         dataV.size();
35         StringBuilder addressHtml = new StringBuilder();
36         try{
37             JSONObject result = new JSONObject();
38             JSONArray installAddressArray = ManualDAOFactory.getManualOrderDAO().loadMSInstallAddress(requestNo);
39
40             addressHtml.append("<select class='form-control listboxtype btn btn-default dropdown' name='INSTALL_ADDRESS_ID'>");
41             addressHtml.append("<option value=''>--Please Select--</option> ");
42
43             for(int i=0; i<installAddressArray.length(); i++){
44                 result = installAddressArray.getJSONObject(i);
45                 String install_address_id = result.getString("install_address_id");
46                 String address_detail = result.getString("address_detail");
47
48                 int matchType = checkNotUseAddress(install_address_id,dataV,rowID);
49                 if(matchType == 1){ //Case found in edit line
50                     addressHtml.append("<option selected value='"+install_address_id+"' "+address_detail+" </option>");
51                 }else if(matchType == 2){ //Case not found in all record
52                     addressHtml.append("<option value='"+install_address_id+"' "+address_detail+" </option>");
53                 }
54             }
55             addressHtml.append("</select>");
56
57         }catch(Exception e){
58         }
59         return addressHtml.toString();
60     }
61
62     private int checkNotUseAddress(String address, Vector dataV, String rowID){
63         for(int i=0; i<dataV.size(); i++){
64             HashMap hStore = (HashMap)dataV.get(i);
65             HashMap process = (HashMap)hStore.get("UPDATE");
66             HashMap table = (HashMap)process.get("ORDER_REQUEST_PRICE_PLAN_ADDRESS");
67             String InstallAddressID = (String)table.get("INSTALL_ADDRESS_ID");
68
69             if(address.equals(InstallAddressID) && i==Integer.valueOf(rowID)){
70                 return 1;
71             }else if(address.equals(InstallAddressID) && i!=Integer.valueOf(rowID)){
72                 return 3;
73             }
74         }
75         return 2;
76     }
}
```

ภาพที่ 3.44 ไฟล์ AjaxServletInstallAddress.java

โค้ดบรรทัด 38 เรียกใช้เมธอด loadMSInstallAddress(String request_no) และเก็บค่าไว้ในตัวแปร installAddressArray ซึ่งเป็น JSONArray

โค้ดบรรทัด 43 ใช้ for loop เพื่อดึงค่าที่เก็บเป็น JSONObject ออกมาเก็บไว้ในตัวแปร result

โค้ดบรรทัด 45 และ 46 ดึงค่าข้อมูลออกมาเป็น String

โค้ดบรรทัด 48 เรียกใช้เมธอด checkNotUseAddress(String address, Vector dataV, String rowID) เพื่อเช็คค่า Address ไหนยังไม่ถูกเลือก

โค้ดบรรทัด 69 ถ้าค่าของตัวแปร address (ค่าของ install_address_id ในฐานข้อมูล) เท่ากับ ค่าของตัวแปร InstallAddressID (ค่าของ field INSTALL_ADDRESS_ID ใน vector ที่ดึงมาจากหน้าจอ) และค่า i เท่ากับ rowID (record ที่กดแก้ไข) ให้ return 1 (แสดงว่า Address นั้น ถูกเลือกแล้วที่ record ตัวมันเอง)

โค้ดบรรทัด 71 ถ้าค่าของตัวแปร address (ค่าของ install_address_id ในฐานข้อมูล) เท่ากับ ค่าของตัวแปร InstallAddressID (ค่าของ field INSTALL_ADDRESS_ID ใน vector ที่ดึงมาจากหน้าจอ) แต่ค่า i ไม่เท่ากับ rowID (record ที่กดแก้ไข) ให้ return 3 (แสดงว่า Address นั้น ถูกเลือกแล้วที่ record อื่น)

โค้ดบรรทัด 75 ให้ return 2 หากไม่เข้า 2 เงื่อนไขบน (แสดงว่า Address นั้น ยังไม่ถูกเลือก)

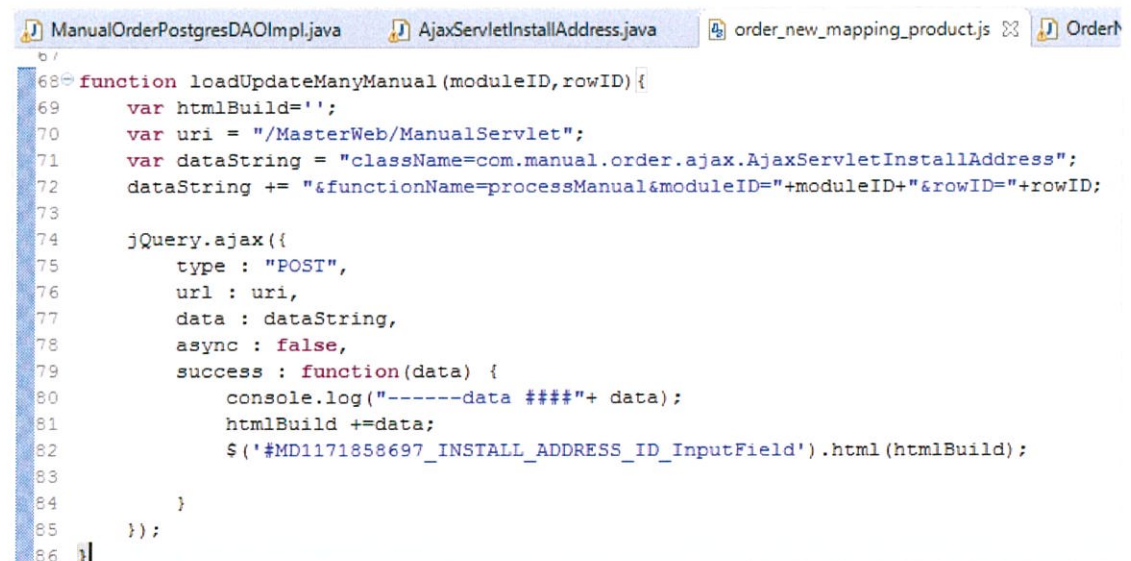
โค้ดบรรทัด 49-53 ให้ใน <option></option> แสดงข้อมูลที่เก็บในตัวแปร address_detail และมี value เท่ากับค่าที่เก็บไว้ในตัวแปร install_address_id แต่ต่างกันว่า ถ้า matchType เป็น 1 ให้ช่อง Install Address ขึ้น default ค่าเป็น Address เดิมที่เลือกไว้

โค้ดบรรทัด 59 return HTML ออกไปเป็น String

3.3.8.3 เพิ่ม function ในไฟล์ JavaScript

function loadUpdateManyManual() จะทำงานอัตโนมัติเมื่อกดแก้ไข

record ในโมดูลข้างล่าง



```
ManualOrderPostgresDAOImpl.java AjaxServletInstallAddress.java order_new_mapping_product.js Order
68 function loadUpdateManyManual (moduleID, rowID) {
69     var htmlBuild='';
70     var uri = "/MasterWeb/ManualServlet";
71     var dataString = "className=com.manual.order.ajax.AjaxServletInstallAddress";
72     dataString += "&functionName=processManual&moduleID="+moduleID+"&rowID="+rowID;
73
74     jQuery.ajax({
75         type : "POST",
76         url : uri,
77         data : dataString,
78         async : false,
79         success : function(data) {
80             console.log("-----data ####"+ data);
81             htmlBuild +=data;
82             $('#MD1171858697_INSTALL_ADDRESS_ID_InputField').html(htmlBuild);
83
84         }
85     });
86 }
```

ภาพที่ 3.45 function loadUpdateManyManual(moduleID,rowID)

โค้ดบรรทัด 68 สร้าง function loadUpdateManyManual(moduleID,rowID) ใช้ jQuery.ajax() เพื่อส่ง request ไปยัง AjaxServletInstallAddress.java

โค้ดบรรทัด 81 data ที่ได้รับ response กลับมา เป็น HTML ที่เป็น String เก็บในตัวแปร htmlBuild
 โค้ดบรรทัด 82 แปลง String ให้อยู่ในรูปแบบโครงสร้างภาษา HTML และใส่เข้าไปใน field ID =
 MD1171858697_INSTALL_ADDRESS_ID_InputField (เป็น ID ของช่อง Install Address ในหน้า
 popup)

3.3.9 Filter Search VAS ภายใต้ product id ใน package ที่เลือก

ต้องการให้หน้า Search VAS ทำการ search VAS ขึ้นมาเฉพาะ VAS ที่อยู่ภายใต้
 product id ใน package ที่เลือกเท่านั้น

Back
Submit
Reject

Service Request Detail

SR No	<input type="text" value="SR-222222-222222"/>	SR Date	<input type="text" value="02/09/2017"/>
Order Type	<input type="text" value="ชนิดบริการ (New)"/>	Customer Type	<input type="text" value="ลูกค้าบ้านพักอาศัยประเภททั่วไป"/>
CA Account	<input type="text" value="CA-000000-000000"/>	Customer Name	<input type="text" value="นายจักษิตอน นฤพรพัฒน์"/>
Billing Account	<input type="text" value="123456789012"/>	Billing Name	<input type="text" value="นายจักษิตอน นฤพรพัฒน์"/>
Billing Address	<input type="text" value="1150 อาคารพิเศษ"/>		

Package Offering

Package/Product - VAS

No.	Package Name	Product Name	Price	Period Date	Promotion Bill Cycle	Quantity	VAS
1	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	Normal Telephone	500.00	01/01/2011 - 30/04/2018	1	1	<input type="button" value="VAS"/>
2	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	General Discount Product(n)	0	01/01/2011 - 30/04/2018	1	1	<input type="button" value="VAS"/>
3	ค่าบริการ Strong Life Up to 40Mb/20Mb_Fixed Line	Fiber 2U(n)	530.00	01/01/2011 - 30/04/2018	1	1	<input type="button" value="VAS"/>

ภาพที่ 3.46 Product-VAS

จากหน้า Product-VAS เมื่อกดแก้ไขที่ โมดูลล่าง มันจะไปเปิดหน้า Add VAS ซึ่งเป็นอีก Entity หนึ่ง พร้อมทั้งส่งค่า PRODUCT_ID ไปด้วย (ทำการ configuration ใน ONEWEB)

Package/Product Detail

Package Name: Period Date:

Product Name: Promotion Bill Cycle:

Price Plan Name:

Value Added Service

<input type="checkbox"/>	VAS Group	VAS Code	VAS Name
<input type="checkbox"/>	SPC	1	SPC-Pack 1(Free: รับสายเรียกซ้อน(เกินสาม)
<input type="checkbox"/>	SPC	2	SPC-Pack 8(Free: รับสายเรียกซ้อน(เกินสาม)พร้อมผ่านโทรศัพท์(เรียกซ้ำอัตโนมัติ)
<input type="checkbox"/>	SPC	3	SPC-Pack 9 (เลขหมายย่อ/เลขหมายตาม/แสดงเลขหมายโทรศัพท์)
<input type="checkbox"/>	SPC	4	Std. SPC-Call Forwarding (บริการโอนสาย)
<input type="checkbox"/>	SPC	5	Std. SPC-Automated Call Repetition (บริการโทรซ้ำอัตโนมัติ)
<input type="checkbox"/>	SPC	6	Std. SPC-Abbreviated Dialing (บริการย่อเลขหมาย)
<input type="checkbox"/>	SPC	7	Std. SPC-Conference Call (บริการสนทนา 3 สาย)
<input type="checkbox"/>	SPC	9	Std. SPC-Call Waiting (บริการรับสายเรียกซ้อน)
<input type="checkbox"/>	SPC	10	Std. SPC-Immediate Charge Info (บริการแจ้งค่าชุดโทรศัพท์ทางไกลทันที)

ภาพที่ 3.47 Add VAS

ที่หน้า Add VAS ตรงโมดูลข้างบนจะมีค่า PRODUCT_ID ที่มาจากหน้า Product-VAS และเมื่อกดปุ่ม Add จะเปิดหน้า Search VAS ซึ่งหน้านี้เราจะให้มีการ filter VAS ตาม PRODUCT_ID

Value Added Service

VAS Group: VAS Name:

<input type="checkbox"/>	VAS Group	VAS Code	VAS Name	
<input type="checkbox"/>	SPC	1	SPC-Pack 1(Free: รับสายเรียกซ้อน/เกินสาม)	*
<input type="checkbox"/>	SPC	2	SPC-Pack 8(Free: รับสายเรียกซ้อน/เกินสาม)พร้อมผ่านโทรศัพท์(เรียกซ้ำอัตโนมัติ)	*
<input type="checkbox"/>	SPC	3	SPC-Pack 9 (เลขหมายย่อ/เลขหมายตาม/แสดงเลขหมายโทรศัพท์)	*
<input type="checkbox"/>	SPC	4	Std. SPC-Call Forwarding (บริการโอนสาย)	*
<input type="checkbox"/>	SPC	5	Std. SPC-Automated Call Repetition (บริการโทรซ้ำอัตโนมัติ)	*
<input type="checkbox"/>	SPC	6	Std. SPC-Abbreviated Dialing (บริการย่อเลขหมาย)	*
<input type="checkbox"/>	SPC	7	Std. SPC-Conference Call (บริการสนทนา 3 สาย)	*
<input type="checkbox"/>	SPC	9	Std. SPC-Call Waiting (บริการรับสายเรียกซ้อน)	*
<input type="checkbox"/>	SPC	10	Std. SPC-Immediate Charge Info (บริการแจ้งค่าชุดโทรศัพท์ทางไกลทันที)	*
<input type="checkbox"/>	SPC	11	Std. SPC-Abbreviated Call Forwarding (บริการแจ้งค่าการโทรตาม)	*
<input type="checkbox"/>	SPC	13	Std. SPC-Slip Service to display call number (บริการรับสายตามแสดงเลขหมายโทรศัพท์)	*
<input type="checkbox"/>	Miscellaneous	112	Normal Telephone -ค่าบริการอื่นๆ	*
<input type="checkbox"/>	Miscellaneous	113	N-ISDN - บริการอื่นๆ	*
<input type="checkbox"/>	Miscellaneous	114	ADSL -ค่าบริการอื่นๆ	*

ภาพที่ 3.48 Search VAS

3.3.9.1 สร้าง Java class OrderNewVASSearch

```
OrderNewVASSearch.java
1 package com.manual.order.customize;
2
3 import java.util.HashMap;
4
5
6
7
8
9
10
11
12
13 public class OrderNewVASSearch extends ProcessHelper implements ProcessAction {
14     Logger logger = Logger.getLogger(OrderNewVASSearch.class);
15
16     public HashMap getSearchResult() {
17         String entityID = (String)getRequest().getSession().getAttribute("entityID");
18         EntityFormHandler entityForm = (EntityFormHandler)getRequest().getSession().getAttribute(entityID + "_session");
19         String productID = (String)EAFManualUtil.getDataHashMapFromSession("MD6991276030", request.get("PRODUCT_ID"));
20
21         MasterFormHandler moduleForm = (MasterFormHandler) request.getSession().getAttribute("MD1171741471_session");
22
23         String sql = moduleForm.getModuleM().getSqlWorkQueue();
24         logger.info("##### old sql : " + sql);
25         StringBuilder newSql = new StringBuilder(sql);
26         newSql.append(" where order_ms_vas.product_id =");
27         newSql.append(productID);
28         newSql.append("'");
29
30         logger.info("##### new sql : " + newSql.toString());
31
32         return EAFManualUtil.loadTableDataPopupList(request, newSql.toString(), moduleForm.getPage(),
33             moduleForm.getVolumePerPage(), moduleForm.getSearchOrderBy(), moduleForm.getSearchOrderByType(), "jdbc/order_app");
34
35         //return EAFManualUtil.loadTableData(request, newSql.toString(), moduleForm.getPage(), moduleForm.getVolumePerPage());
36     }
37 }
```

ภาพที่ 3.49 ไฟล์ OrderNewVASSearch.java

โค้ดบรรทัด 19 ดึงค่า PRODUCT_ID จาก Module ID = MD6991276030 ซึ่งเป็นโมดูลข้างบนของหน้า Add VAS

โค้ดบรรทัด 21 ใช้คลาส MasterFormHandler ระบุเซสชันโมดูล MD1171741471_session (เป็นเลขโมดูลของหน้า Search VAS) หากระบุเป็น MainModuleID เหมือนที่เคยเขียนในคลาสอื่น มันจะไปเรียกโมดูลหลักของหน้า Product-VAS แทน เพราะมันทำงานต่อเนื่องกัน

โค้ดบรรทัด 26-28 เขียน SQL เพิ่มเข้าไปจาก SQL เดิม โดยระบุเงื่อนไขให้ query ข้อมูลเฉพาะที่ product_id = PRODUCT_ID ที่ดึงค่ามาจากหน้า Add VAS

3.3.9.2 Insert ไฟล์ Java

กลับไปที่ ONEWEB App Designer เปิดหน้า Entity Search VAS ทำการแก้ไข Entity Action เลือก Process Name เป็น SEARCH และช่อง Class Action ใส่ com.manual.order.customize.OrderNewVASSearch

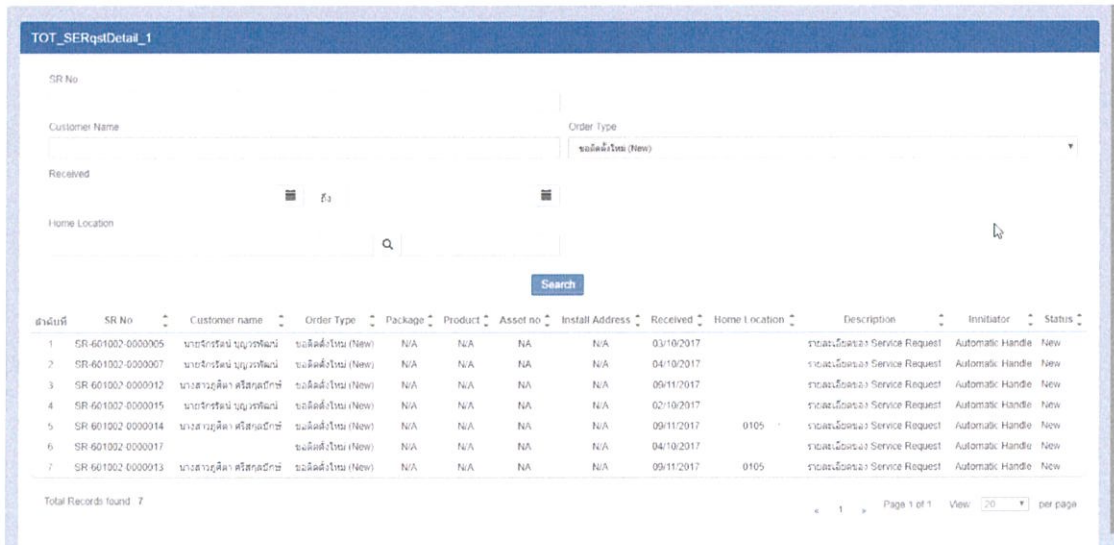
บทที่ 4

ผลการดำเนินงาน

4.1 ผลการเขียน Reject popup

4.1.1 แสดง Reject popup

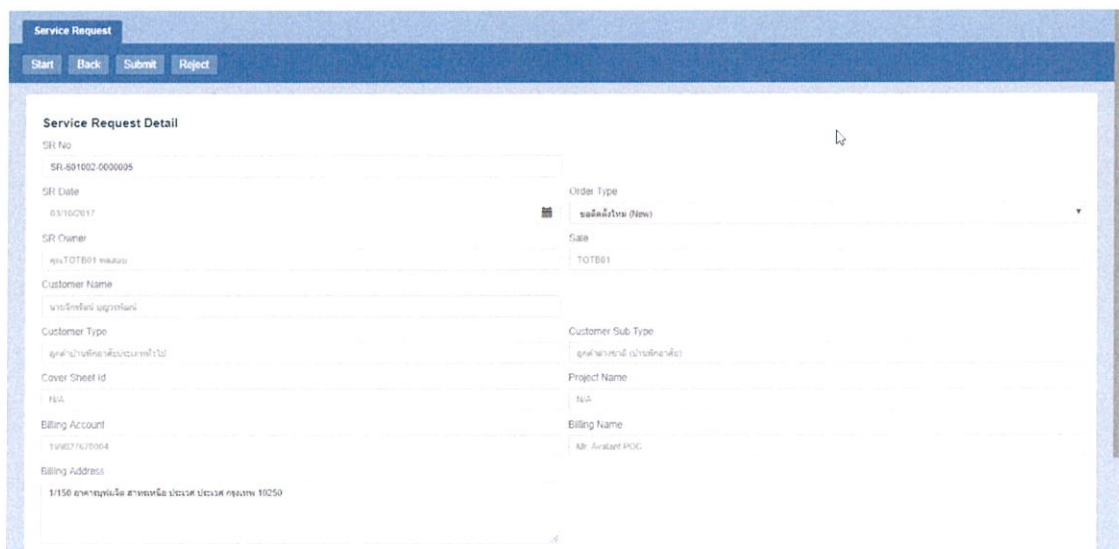
1) ทดลองโดยการ เปิดหน้า ServiceRequest (Search) ขึ้นมา



The screenshot shows a web application interface for searching Service Requests. At the top, there are input fields for SR No, Customer Name, Order Type (set to 'ขอติดตั้งใหม่ (New)'), Received, and Home Location. A 'Search' button is located below these fields. Below the search area is a table with 7 columns: SR No, Customer name, Order Type, Package, Product, Asset no, Install Address, Received, Home location, Description, Initiator, and Status. The table contains 7 rows of data, all with a status of 'New'. At the bottom, it indicates 'Total Records found: 7' and 'Page 1 of 1'.

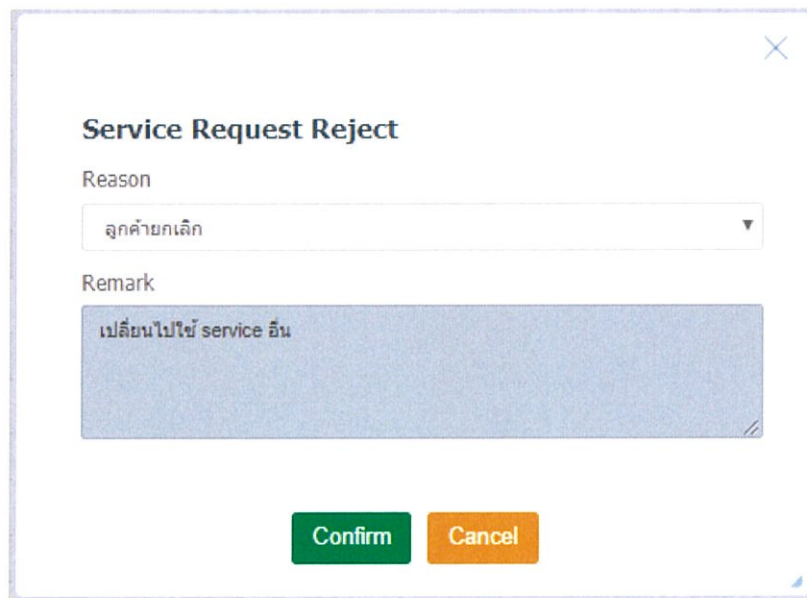
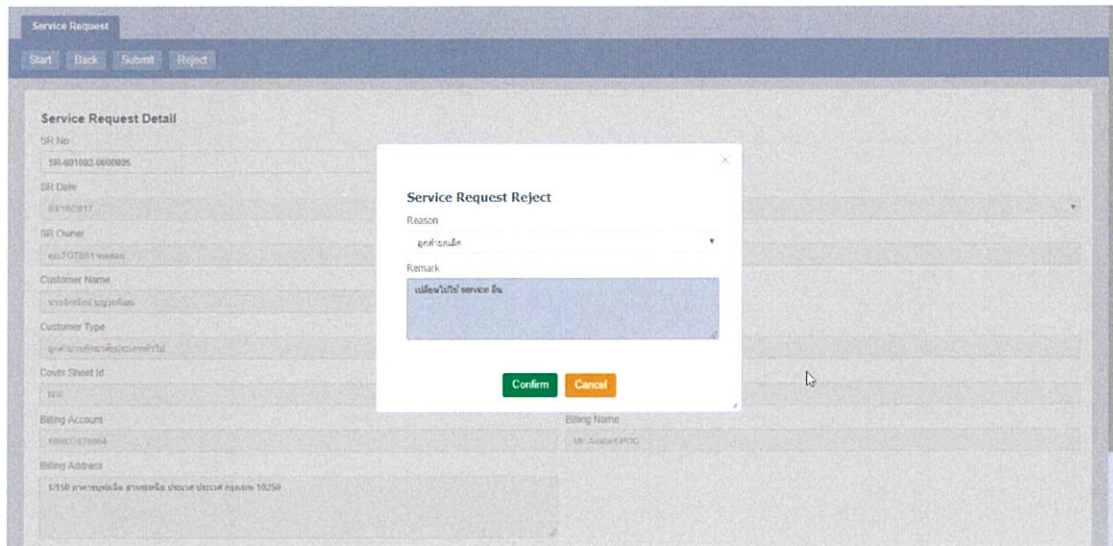
ลำดับที่	SR No	Customer name	Order Type	Package	Product	Asset no	Install Address	Received	Home location	Description	Initiator	Status
1	SR-601002-0000005	นายจักรพันธ์ บุญพรพัฒน์	ขอติดตั้งใหม่ (New)	N/A	N/A	NA	N/A	03/10/2017		รายละเอียดของ Service Request	Automatic Handle	New
2	SR-601002-0000007	นายจักรพันธ์ บุญพรพัฒน์	ขอติดตั้งใหม่ (New)	N/A	N/A	NA	N/A	04/10/2017		รายละเอียดของ Service Request	Automatic Handle	New
3	SR-601002-0000012	นางสาวจุติลา ศรีสกุลศรีชัย	ขอติดตั้งใหม่ (New)	N/A	N/A	NA	N/A	09/11/2017		รายละเอียดของ Service Request	Automatic Handle	New
4	SR-601002-0000015	นายจักรพันธ์ บุญพรพัฒน์	ขอติดตั้งใหม่ (New)	N/A	N/A	NA	N/A	02/10/2017		รายละเอียดของ Service Request	Automatic Handle	New
5	SR-601002-0000014	นางสาวจุติลา ศรีสกุลศรีชัย	ขอติดตั้งใหม่ (New)	N/A	N/A	NA	N/A	09/11/2017	0105	รายละเอียดของ Service Request	Automatic Handle	New
6	SR-601002-0000017	นายจักรพันธ์ บุญพรพัฒน์	ขอติดตั้งใหม่ (New)	N/A	N/A	NA	N/A	04/10/2017		รายละเอียดของ Service Request	Automatic Handle	New
7	SR-601002-0000013	นางสาวจุติลา ศรีสกุลศรีชัย	ขอติดตั้งใหม่ (New)	N/A	N/A	NA	N/A	09/11/2017	0105	รายละเอียดของ Service Request	Automatic Handle	New

2) กดเข้าไปที่ record ใด record หนึ่ง จะเข้าไปหน้า ServiceRequest (Update) ที่มีข้อมูลต่างๆ ของ record นั้นอยู่ (ยกตัวอย่าง record SR No = SR-601002-0000005)



The screenshot shows the 'Service Request Detail' page for SR No SR-601002-0000005. The page has a navigation bar with 'Start', 'Back', 'Submit', and 'Reject' buttons. The main content area is divided into two columns. The left column contains fields for SR No, SR Date (03/10/2017), SR Owner (นายจักรพันธ์ บุญพรพัฒน์), Customer Name (นายจักรพันธ์ บุญพรพัฒน์), Customer Type (ลูกค้าราชการ/หน่วยงานราชการ), Cover Sheet Id (N/A), Billing Account (1080276/10004), and Billing Address (1/150 ถนนสุขุมวิท แขวงคลองเตย เขตวัฒนา กรุงเทพฯ 10250). The right column contains fields for Order Type (ขอติดตั้งใหม่ (New)), Sub (TOTB61), Customer Sub Type (ลูกค้าราชการ (บ้านศึกษา/โรงเรียน)), Project Name (N/A), and Billing Name (Mr. Jirapatt POC).

3) ทดลองกดปุ่ม Reject



ภาพที่ 4.1 แสดง popup เมื่อกดปุ่ม Reject

4.1.2 บันทึกค่าสถานะ 'Reject' และเหตุผลในการ Reject ลงในฐานข้อมูล กดปุ่ม Confirm ใน Reject popup ภาพที่ 4.1 แล้วในฐานข้อมูล ที่ SR No = SR-601002-0000005 จะต้องมีการบันทึกค่าใน attribute decision = 'Reject', reject_reason และ remark

```

1 select order_service_request.request_no,
2 order_service_request.reject_reason,
3 order_service_request.remark,
4 order_service_request.decision
5 from order_service_request
6 where order_service_request.request_no = 'SR-601002-0000005'

```

* request_no	reject_reason	remark	decision
1 SR-601002-0000005	RJ01	เปลี่ยนไปใช้ service อื่น	Reject

ภาพที่ 4.2 บันทึกค่า Reject และเหตุผลในการ Reject ลงในฐานข้อมูล

และหลังจากกดปุ่ม Confirm Reject จะทำการกลับไปหน้าจอแรกสุดคือหน้า ServiceRequest (Search) เสมอ ไม่ว่าจะ Reject ที่หน้าไหนก็ตาม

4.2 บันทึกค่าสถานะ 'Submit'

จากหัวข้อ 4.1.1 ขั้นตอนที่ 3) กดปุ่ม Submit แทนการกดปุ่ม Reject แล้วในฐานข้อมูล ที่ SR No = SR-601002-0000005 จะต้องมีการบันทึกค่าใน attribute decision = 'Submit'

```

1 select order_service_request.request_no,
2 order_service_request.reject_reason,
3 order_service_request.remark,
4 order_service_request.decision
5 from order_service_request
6 where request_no = 'SR-601002-0000005'

```

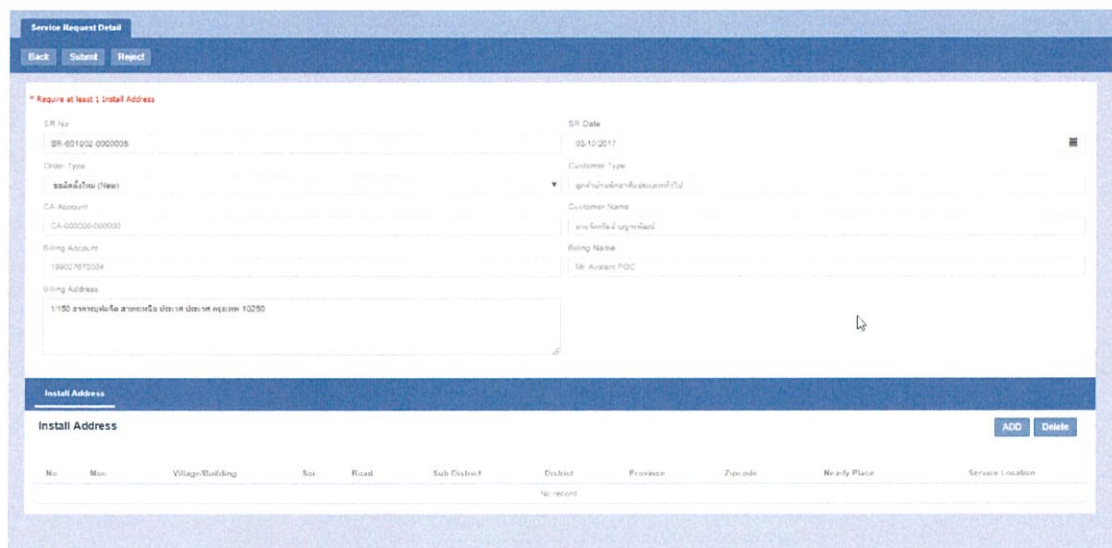
* request_no	reject_reason	remark	decision
1 SR-601002-0000005	(null)	(null)	Submit

ภาพที่ 4.3 บันทึกค่า Submit

หลังจากกดปุ่ม Submit แล้ว จะทำการ redirect screen ไปหน้าถัดไป คือหน้า Input Install Address และถ้ากด Submit ที่หน้า Input Install Address ก็จะไปหน้าถัดไป คือหน้า Package Offering เป็นแบบนี้เสมอ ยกเว้นการกดปุ่ม Submit หน้าสุดท้าย หน้า Create Quote จะกลับไปยังหน้าแรก คือหน้า ServiceRequest (Search)

4.3 แจ้งเตือนให้ระบุ Address อย่างน้อย 1 record ในหน้า Input Install Address

ในหน้า Input Install Address ที่โมดูลข้างล่าง หากไม่มีการระบุ Address แล้วกดปุ่ม Submit จะแจ้งข้อความสีแดงว่า Require at least 1 Install Address



The screenshot shows a web form titled "Service Request Detail" with a blue header. Below the header are buttons for "Back", "Submit", and "Reset". The form contains several input fields: "SR No" (SR-001902-0000005), "SR Date" (03/10/2017), "Order Type" (ระบุชนิดใหม่ (New)), "Customer Type" (ลูกค้าบ้านเดี่ยว-คอนโด-อพาร์ทเมนท์), "CA Account" (CA-000000-000000), "Customer Name" (นายสุวิทย์ ใจบุญใจดี), "Billing Account" (18902707004), "Billing Name" (Mr. Anant POC), and "Billing Address" (1/150 แขวงจตุจักร เขตจตุจักร กรุงเทพฯ 10250). A red error message "Require at least 1 Install Address" is displayed above the "Install Address" section. This section contains a table with columns: No., Moo, Village/Building, Soi, Road, Sub-District, District, Province, Zipcode, No. of Place, and Service Location. The table is currently empty, showing "No record".

ภาพที่ 4.4 แจ้งเตือนให้ระบุ Address อย่างน้อย 1 record

4.4 แจ้งเตือนให้ระบุ Package อย่างน้อย 1 record ในหน้า Package Offering

ในหน้า Package Offering ที่โมดูลข้างล่าง หากไม่มีการระบุ Package แล้วกดปุ่ม Submit จะแจ้งข้อความสีแดงว่า Require at least 1 Package Offering



ภาพที่ 4.5 แจ้งเตือนให้ระบุ Package อย่างน้อย 1 record

4.5 Filter Search VAS ภายใต้ product id

ในหน้า Product-VAS เมื่อกด แก้ไขที่ record ในโมดูลข้างล่าง จะเปิดหน้า Add VAS พร้อมทั้งส่งค่า PRODUCT_ID ไปด้วย (ทำการ configuration ใน ONEWEB)



ภาพที่ 4.6 หน้า Product-VAS

Service Request Detail Package/Product Detail

Submit Cancel

Package Name: 45726 400 user (Normal Telephone) Period Date: 01/01/2011 - 30/04/2018

Product Name: General Discount Product(C) Promotion Cycle: 1

Price Plan Name: Unlimited 400 Fixed Line

VAS

Value Added Service Add Delete

VAS Group	VAS Code	VAS Name
SPC	3	SPC Pack 1 (Free - บริการโทรศัพท์พื้นฐาน)

ภาพที่ 4.7 หน้า Add VAS

ที่หน้า Add VAS ตรงโมดูลข้างบนจะมีค่า PRODUCT_ID ที่มาจากหน้า Product-VAS และเมื่อกดปุ่ม Add จะเปิดหน้า Search VAS ซึ่งหน้านี้เราจะให้มีการ filter VAS ตาม PRODUCT_ID

Value Added Service

VAS Group VAS Name

Search Clear

VAS Group	VAS Code	VAS Name
SPC	3	SPC Pack 1 (Free - บริการโทรศัพท์พื้นฐาน)
SPC	4	SPC Pack 8 (Free - บริการโทรศัพท์พื้นฐาน/บริการโทรเลข/บริการโทรสาร)

Total Records found: 2

Page 1 of 1 View 20 per page

Submit

ภาพที่ 4.8 หน้า Search VAS ที่ filter ตาม PRODUCT_ID

เมื่อกดปุ่ม Search ที่หน้า Search VAS จะแสดง VAS ที่อยู่ภายใต้ PRODUCT_ID ที่กดเข้ามาจากหน้า Product-VAS

4.6 แจ้งเตือนให้ Mapping Address ให้ครบทุก Package/Product

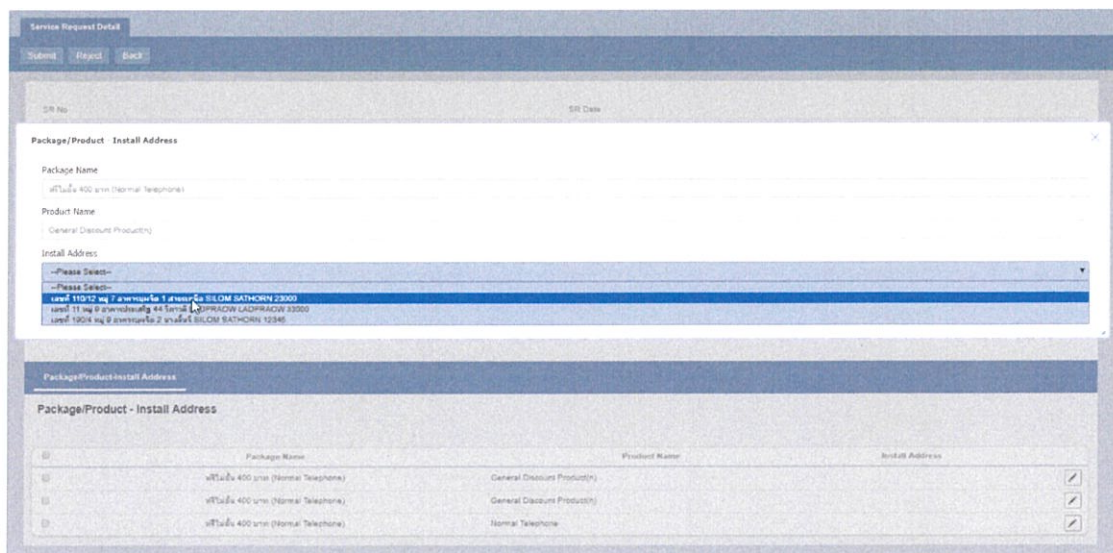
ในหน้า Mapping Product ที่โมดูลข้างล่าง หากระบุ Address ไม่ครบทุก Package/Product แล้วกดปุ่ม Submit จะแจ้งข้อความสีแดงว่า Please Update all Install Address



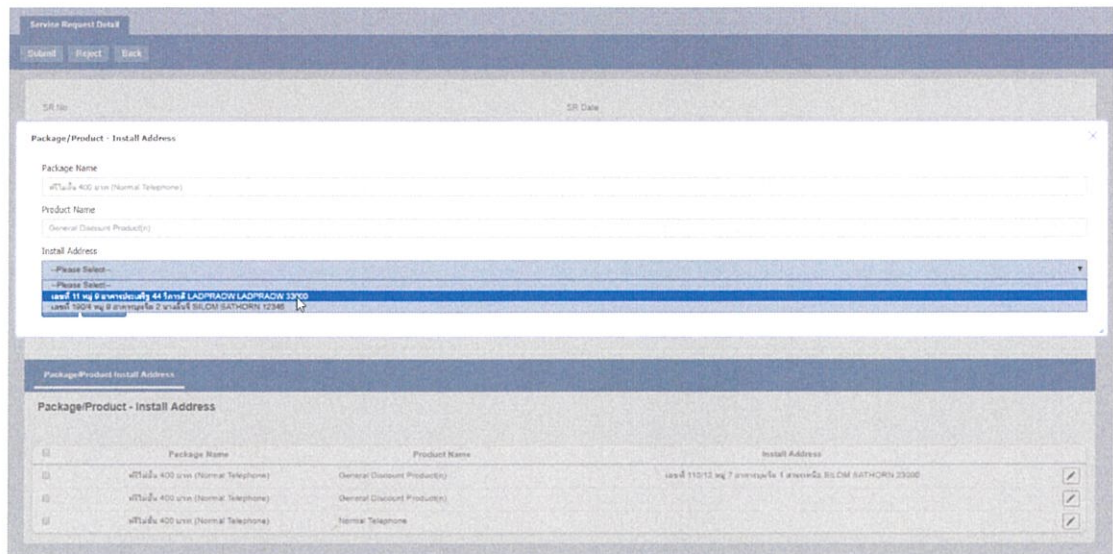
ภาพที่ 4.9 แจ้งเตือนให้ระบุ Address ให้ครบทุก Package/Product

4.7 Filter to show Address

ในหน้า Mapping Product ต้องมีการระบุ Address โดยให้กดแก้ไขที่ record ในโมดูลข้างล่าง แล้วจะมี popup ขึ้นมาให้เลือก Address



ภาพที่ 4.10 แสดง Address ทั้งหมด



ภาพที่ 4.11 แสดงเฉพาะ Address ที่ยังไม่ถูกเลือก

ตรงช่อง Install Address ในหน้า popup จะแสดงเฉพาะ Address ที่ยังไม่ถูกเลือกเท่านั้น และถ้าหากไปกดแก้ไขตรง record ที่มีการระบุ Address อยู่แล้ว ช่อง Install Address ในหน้า popup จะต้องแสดง Address ที่ยังไม่ถูกเลือก และ Address เดิมของตัวเองด้วย

บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการศึกษาและพัฒนาระบบด้วย ONEWEB 4.0 แพลตฟอร์ม ร่วมกับการเขียนโปรแกรม (Customize) ด้วยภาษา HTML, Java, JavaScript, jQuery, Ajax, JSON ทำให้ระบบมีประสิทธิภาพและมีความสามารถในการทำงานที่ตรงกับความต้องการของลูกค้า เป็นการพิสูจน์แนวคิด (Proof of Concept) ว่าระบบนี้สามารถทำให้เกิดขึ้นได้และใช้งานได้จริง เพื่อเพิ่มความมั่นใจให้ลูกค้าในการตัดสินใจซื้อระบบจากเรา

5.2 ปัญหาและแนวทางแก้ไข

5.2.1 ปัญหา

5.2.1.1 ปัญหาในขั้นตอนการเขียนโปรแกรม ส่วนใหญ่จะเป็นการใช้คลาสและเมธอดที่บริษัทมีอยู่แล้ว ซึ่งเราไม่รู้การทำการของคลาสหรือเมธอดนั้น บางครั้งจึงนำมาใช้ไม่ถูกหรือบางครั้งก็ไม่ได้นำมาใช้ให้เกิดประโยชน์ และทำให้เสียเวลาในการแก้ไข

5.2.1.2 ปัญหาในขั้นตอนที่ต้องนำโค้ดที่เขียนมาใช้ร่วมกับ ONEWEB 4.0 แพลตฟอร์ม เนื่องจากขั้นตอนการสร้างหน้า User Interface รับผิดชอบโดยผู้อื่น บางครั้งก็ไม่ได้ใส่ field ที่เราต้องใช้มา จึงทำให้เวลาที่เร preview หน้าจอมันขึ้น error

5.2.2 แนวทางแก้ไข

5.2.2.1 ศึกษาการใช้คลาสและเมธอดของบริษัทจากโค้ดใน project อื่นๆ รวมถึงการถามและขอความรู้จากพี่ๆในบริษัท

5.2.2.2 ก่อนการเขียนโค้ด จะมีการสอนการใช้ ONEWEB 4.0 แพลตฟอร์ม จากทางบริษัท ทำให้สามารถเข้าไปแก้ไขการ configuration ในตัว ONEWEB App Designer เองได้ โดยไม่ต้องแจ้งกลับไปให้ผู้สร้าง User Interface ทำให้สามารถพัฒนาระบบได้เร็วขึ้น

5.3 ข้อเสนอแนะ

สามารถนำความรู้ต่างๆ ที่ได้จากการศึกษาและใช้งาน ONEWEB ไปสร้างระบบอื่นๆ ได้อีกมากมาย และหากหมั่นศึกษาและพัฒนาการเขียนโปรแกรมอย่างต่อเนื่อง จะสามารถสร้างระบบที่มีการทำงานที่ซับซ้อนและมีประสิทธิภาพมากยิ่งขึ้น

เอกสารอ้างอิง

- [1] Avalant Co., Ltd [ออนไลน์] เข้าถึงได้จาก:
<http://www.avalant.co.th/index.html>
- [2] ระบบการจัดการฐานข้อมูล [ออนไลน์] เข้าถึงได้จาก:
<http://www.pongkorn.net/?article:194>
- [3] SQL [ออนไลน์] เข้าถึงได้จาก:
<http://www.mindphp.com/คู่มือ/73-คืออะไร/2088-sql-คืออะไร.html>
- [4] ภาษานิยามข้อมูล [ออนไลน์] เข้าถึงได้จาก:
<https://sqlbysayuri.wordpress.com/ภาษานิยามข้อมูล-data-definition-language-ddl/>
- [5] SELECT Statement [ออนไลน์] เข้าถึงได้จาก:
<http://www.widbase.net/database/sql/sqlquery/sqlquery01.shtml>
- [6] ภาพตัวอย่างการใช้คำสั่ง INSERT [ออนไลน์] เข้าถึงได้จาก:
http://www.w3im.com/th/sql/sql_insert.html
- [7] ภาพตัวอย่างการใช้คำสั่ง UPDATE [ออนไลน์] เข้าถึงได้จาก:
http://www.w3im.com/th/sql/sql_update.html
- [8] ภาพตัวอย่างการใช้คำสั่ง DELETE [ออนไลน์] เข้าถึงได้จาก:
http://www.w3im.com/th/sql/sql_delete.html
- [9] Java programming language [ออนไลน์] เข้าถึงได้จาก:
<http://www.mindphp.com/คู่มือ/73-คืออะไร/2185-java-คืออะไร.html>
- [10] องค์ประกอบของคลาส [ออนไลน์] เข้าถึงได้จาก:
<http://marcuscode.com/lang/java/object-oriented-programming>
- [11] โครงสร้างของภาษา Java [ออนไลน์] เข้าถึงได้จาก:
<http://marcuscode.com/lang/java/program-struct>
- [12] ความรู้เบื้องต้นภาษา HTML [ออนไลน์] เข้าถึงได้จาก:
<https://pirun.ku.ac.th/~agrtnk/web/units/unit1/unit1-2.htm>
- [13] JavaScript [ออนไลน์] เข้าถึงได้จาก:
<http://www.mindphp.com/คู่มือ/73-คืออะไร/2187-java-javascript-คืออะไร.html>
- [14] jQuery [ออนไลน์] เข้าถึงได้จาก:

- <http://www.thaicreate.com/jquery.html>
- [15] การทำงานของเว็บแอปพลิเคชัน [ออนไลน์] เข้าถึงได้จาก:
<http://www.thaicreate.com/tutorial/ajax-introduction.html>
- [16] JSP & Servlet [ออนไลน์] เข้าถึงได้จาก:
<http://know.9choo.in.th/2013/06/jsp-servlet-part1.html>
- [17] หลักการทำงานของ JSP และ Servlet [ออนไลน์] เข้าถึงได้จาก:
<http://know.9choo.in.th/2013/06/jsp-servlet-part1.html>
- [18] Create DAO [ออนไลน์] เข้าถึงได้จาก:
<http://mystou.com/basic-javaweb-6/>