



รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบปฏิบัติการปัญญาประดิษฐ์เชิงบูรณาการ
Robotic Processes Integration

นางสาวนันทิชาพร สีแดง

ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560



รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบปฏิบัติการปัญญาประดิษฐ์เชิงบูรณาการ

Robotic Processes Integration

นางสาวนันทชพร สีแดง

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ชื่อโครงการสหกิจศึกษา ระบบปฏิบัติการปัญญาประดิษฐ์เชิงบูรณาการ

ชื่อ-สกุล นักศึกษา นางสาวนันท์ชพร สีแดง

คณะ วิศวกรรมศาสตร์

ภาควิชา วิศวกรรมคอมพิวเตอร์

ชื่อ-สกุล อาจารย์นิเทศ ผศ.ดร.พิกุลแก้ว ตั้งติสานนท์

ชื่อ-สกุล ผู้นิเทศงาน พริยะ ดุสิตานนท์

ชื่อสถานประกอบการ บริษัท เอ็กซอนโมบิล จำกัด

บทคัดย่อ

ในปัจจุบันฝ่ายจัดซื้อของบริษัทมีโปรเจกต์ที่เพิ่มขึ้นเป็นจำนวนมาก ทำให้ต้องใช้เวลาในการทดสอบโปรแกรมมาก เพื่อให้สามารถมั่นใจได้ว่าโปรแกรมนั้นสามารถทำงานได้ถูกต้องตามที่ต้องการ นอกจากนี้ผู้ใช้โปรแกรมยังมีคำถามเกี่ยวกับโปรเจกต์ที่เพิ่มขึ้นมากมาย ทำให้ทางฝ่ายจัดซื้อต้องใช้เวลาในการตอบคำถามซ้ำ ๆ ที่มากเกินไป เพื่อเป็นการพัฒนาประสิทธิภาพการทำงานของฝ่าย จึงได้นำเทคโนโลยีระบบอัตโนมัติต่าง ๆ เข้ามาช่วยในการพัฒนาการทำงานของทีม ได้แก่ การสร้างระบบกระบวนการทำงานอัตโนมัติโดยหุ่นยนต์และระบบแชทบอทในการแก้ไขปัญหาดังกล่าว เพื่อช่วยให้พนักงานสามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น โดยสามารถลดเวลาที่ใช้ได้ประมาณ 5 เท่า เมื่อเทียบกับการทำงานแบบปกติ และยังช่วยเพิ่มเวลาให้กับพนักงาน เพื่อที่จะได้มีเวลาในการทำงานที่สำคัญกว่ามากขึ้น

คำสำคัญ : แชทบอท, การทดสอบอัตโนมัติ, การทำงานอัตโนมัติโดยหุ่นยนต์, การทำงานซ้ำ, เอสเอพี

Co-operative Title: Robotic Processes Integration

Student Intern Name: Nantatchaporn Seedang

Faculty: Engineering

Department: Computer Engineering

Advisor Name: Asst.Prof. Dr. Pikulkaew Tangtisanon

Mentor Name: Piriya Dusitanont

Company: Exxon Mobil Limited

ABSTRACT

Current situation in cooperate process procurement team is we have many project that teams had made, they have to do repetitive testing to make sure that the program work correctly as it should. And team member or users can have many question about the project or want to find the document. So teams have to answer a repetitive question and this is taking time. I implement automation process to help improve their work including test automation using Robotic Process Automation (RPA) technology and chat bot to help teams can do their job correctly and effectively. And reduce the time by 5 times that teams have to do a repetitive work so they can have more time to do other important jobs.

Keywords : Chat bot, Test automation, Robotic Process Automation (RPA), Repetitive work, SAP

กิตติกรรมประกาศ

ข้าพเจ้าได้มีโอกาสปฏิบัติงานโครงการสหกิจศึกษาที่บริษัท เอ็กซอนโมบิล จำกัด ในทีม Cooperate processes procurement ตั้งแต่วันที่ 7 สิงหาคม พ.ศ.2560 จนถึงวันที่ 22 ธันวาคม พ.ศ.2560 ซึ่งทำให้ข้าพเจ้าได้ความรู้และประสบการณ์มากมายเกี่ยวกับการทำงาน ซึ่งจะเป็นประโยชน์ในการประกอบอาชีพของข้าพเจ้าต่อไปในอนาคต

ข้าพเจ้าต้องขอขอบคุณบริษัท เอ็กซอนโมบิล จำกัด ที่ให้โอกาสข้าพเจ้าได้มาปฏิบัติงานสหกิจศึกษาที่บริษัท และพนักงานในทีมทุกคนที่ให้ความรู้และคำแนะนำต่าง ๆ แก่ข้าพเจ้า ที่สามารถนำไปปรับใช้กับการทำงานและชีวิตประจำวันได้ต่อไปในอนาคต

ขอขอบคุณ ผศ.ดร.พิกุลแก้ว ดั่งติสานนท์ อาจารย์ที่ปรึกษาที่คอยให้คำแนะนำและช่วยเหลือในการทำสหกิจศึกษาครั้งนี้สำเร็จลุล่วงไปได้ด้วยดี

นันทชพร สีแดง

สารบัญ

บทที่	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ขอบเขตของการวิจัย.....	2
1.4 วิธีการดำเนินการวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 ทฤษฎีที่เกี่ยวข้อง.....	4
2.2 โปรแกรมที่ใช้.....	12
2.3 ภาษาโปรแกรมที่ใช้.....	16
บทที่ 3 วิธีดำเนินการวิจัย.....	17
3.1 การวางแผนการทำงาน.....	17
3.2 การศึกษาขั้นตอนการทำงาน.....	17
3.3 การออกแบบการทำงาน.....	33
บทที่ 4 ผลการวิจัย.....	38
4.1 ระบบแชทบอท.....	38

4.2 Vendor invoice management (VIM) automated testing tool.....	47
4.3 Optical character recognition (OCR) automated posting tool	56
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	70
5.1 สรุปผลการวิจัย.....	70
5.2 ข้อเสนอแนะ	71
เอกสารอ้างอิง	72
ภาคผนวก ก.....	74
ภาคผนวก ข.....	79

สารบัญตาราง

ตารางที่	หน้า
3.1 แผนการทำงานโครงการสหกิจศึกษา.....	17
5.1 เปรียบเทียบเวลาที่ใช้ในการทำ VIM automated testing tool	70
5.2 เปรียบเทียบเวลาที่ใช้ในการทำ OCR automated testing tool.....	71

สารบัญภาพ

ภาพที่	หน้า
2.1 SkypeX Framework Block Diagram	4
2.2 SkypeX Program Flow Diagram	5
2.3 SkypeX MessageHandler Diagram.....	5
2.4 การทำงานของ Natural Language Processing ร่วมกับ Rule-based.....	7
2.5 ตัวอย่างวิธีการทำงานของ RPA.....	8
2.6 ERP modules Diagram.....	9
2.7 SAP ERP Application Modules.....	9
2.8 โครงสร้างการใช้งาน UCMA 4.0.....	13
2.9 ตัวอย่างการใช้งาน UCMA 4.0 ใน contact center application.....	13
2.10 ตัวอย่างการทำ Version control ใน TFS.....	14
2.11 ตัวอย่างการแสดงผล Kanban board ใน TFS.....	15
2.12 logo โปรแกรม UiPath Studio	15
2.13 ตัวอย่างโปรแกรม UiPath Studio.....	16
3.1 กระบวนการการจัดซื้อและการจ่ายเงินทั้งหมด.....	18
3.2 หน้า Process scanned incoming vendor invoices	19
3.3 หน้า Storing documents.....	19
3.4 หน้า Drag & Drop	20
3.5 หน้า Business workplace.....	20
3.6 หน้าโปรแกรม VIM (Basic Data).....	21
3.7 หน้าโปรแกรม VIM (Line Item Info).....	21
3.8 หน้าโปรแกรม VIM (Vendor Info).....	22
3.9 หน้าโปรแกรม VIM (Additional Info)	22
3.10 หน้า VIM Analytics	23
3.11 หน้า VIM Analytics ตาม Document view.....	23
3.12 หน้า Table.....	23
3.13 หน้า Table EDID4.....	24

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.14 หน้าข้อมูล IDoc.....	24
3.15 หน้า Table Views	24
3.16 หน้า Maintain Document ID	25
3.17 หน้า OCR Document Header.....	25
3.18 หน้าแรก IDoc processing	25
3.19 หน้าใส่ข้อมูล IDoc processing.....	26
3.20 หน้าใส่ข้อมูล header EDIDC	26
3.21 หน้าใส่ข้อมูล Segment name (E1EDK01).....	26
3.22 หน้า inbound function module.....	26
3.23 หน้า VIM Analytics	27
3.24 หน้า VIM Analytics ตาม Document view	27
3.25 หน้ามุมมอง Timeline.....	27
3.26 หน้ามุมมอง Timeline หลังจากแบ่งตามข้อมูลของงาน	28
3.27 หน้า Calendar	28
3.28 หน้า Calendar หลังจากกรองข้อมูลแล้ว	28
3.29 หน้าเพิ่มข้อมูล.....	29
3.30 หน้าแก้ไขหรือลบข้อมูล.....	29
3.31 หน้าลบข้อมูลสมบูรณ์.....	30
3.32 หน้า Team site	30
3.33 หน้าแรก PTP Support Library	31
3.34 หน้าแสดงเอกสารใน PTP Support Library	31
3.35 หน้าแรกในระบบ GSP	32
3.36 หน้าแรกของ SMBB.....	32
3.37 หน้าแสดงเอกสารใน SMBB	32
3.38 Use case ของระบบแชทบอท	33
3.39 ภาพรวมการทำงานของระบบแชทบอท	33

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.40 Use case ของกระบวนการทำงานอัตโนมัติโดยหุ่นยนต์.....	34
3.41 แผนผังแสดงขั้นตอนการทำงานของโปรแกรม VIM automated testing tool.....	35
3.42 แผนผังแสดงขั้นตอนการทำงานของโปรแกรม OCR automated testing tool	37
4.1 คำสั่งแบบ rule-based	38
4.2 การใช้คำสั่งแบบ rule-based (proj)	39
4.3 การใช้คำสั่งแบบ rule-based (smbb)	39
4.4 การใช้คำสั่งแบบ rule-based (ptplib).....	40
4.5 การใช้คำสั่งแบบ rule-based (role)	40
4.6 การใช้คำสั่งแบบ rule-based (query).....	41
4.7 การใช้คำสั่ง teachme บันทึกข้อมูลคำถามและคำตอบลงใน sharepoint list.....	41
4.8 การใช้คำสั่งแบบ rule-based (teachme)	41
4.9 การใช้คำสั่งแบบ rule-based (whois).....	42
4.10 การใช้คำสั่งแบบ rule-based (runvim).....	43
4.11 ตัวอย่างอีเมลล์จากคำสั่ง runvim	43
4.12 หน้าแรกของ LUIS.ai	44
4.13 ตัวอย่าง training data ของ intent help	45
4.14 ตัวอย่าง training data ของ intent projectinfo	45
4.15 ตัวอย่าง training data ของ intent document.....	45
4.16 ตัวอย่างการใช้ LUIS ในการประมวลผลให้ได้มาเป็น intent help.....	46
4.17 ตัวอย่างการใช้ LUIS ในการหาข้อมูล project.....	47
4.18 ตัวอย่างการใช้ LUIS ในการหาข้อมูลใน PTP support library.....	47
4.19 flow การใช้ API ในการดึงข้อมูลของผู้ใช้ปัจจุบัน (current logon user).....	48
4.20 โปรแกรมหลัก (main flow)	48
4.21 ตัวอย่างการทำ invoke workflow.....	49
4.22 ตัวอย่างการทำ invoke send email workflow	50
4.23 ตัวอย่าง main flow ของ system APB.....	51

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.24 การทำงานใน flowchart ชื่อ Open SAP.....	51
4.25 การทำงานใน flowchart ชื่อ /nsbwp.....	52
4.26 การทำงานใน flowchart ชื่อ Posting.....	52
4.27 การทำงานในรูปการทำ VIM posting.....	53
4.28 VIM posting แบบ PO และแบบ NPO	53
4.29 การทำงานใน flowchart ชื่อ /n/opt/vim_analytics.....	54
4.30 การทำงานใน flowchart ชื่อ sendEmail.....	54
4.31 ภายใน Activity ชื่อ Assign ใน flowchart ชื่อ sendEmail	55
4.32 ภายใน Activity ชื่อ send Outlook Mail Message ใน flowchart ชื่อ sendEmail.....	55
4.33 ตัวอย่างอีเมลล์รายงานผลการทำ VIM Posting	56
4.34 ตัวอย่างอีเมลล์รายงานผลการทำ VIM Posting (กรณีมีข้อผิดพลาดจากโปรแกรม)	56
4.35 หน้าโปรแกรมหลัก	57
4.36 การใช้ API ในการดึงข้อมูลของผู้ใช้งานปัจจุบัน (current logon user).....	58
4.37 การกำหนดค่าเริ่มต้นของโปรแกรมใน sequence ชื่อ Assign.....	58
4.38 การอ่านข้อมูลจากไฟล์เอ็กเซลใน sequence ชื่อ Read excel file.....	59
4.39 การเปิดโปรแกรม SAP ตามระบบที่ได้กำหนดใน flowchart ชื่อ Open SAP	59
4.40 การทำงานใน activity ชื่อ For each row.....	60
4.41 flowchart หลักในการทำ IDoc testing	61
4.42 กำหนดค่า และ เช็คว่ามีไฟล์อยู่แล้วหรือไม่	61
4.43 การทำงานใน sequence ชื่อ EDID4.....	62
4.44 การทำงานของ sequence ชื่อ Notepad	63
4.45 การทำงานของ sequence ชื่อ sm30.....	63
4.46 การทำงานของ for each activity ใน sequence ชื่อ sm30	64
4.47 การทำงานของ sequence ชื่อ Test tool for Idoc processing.....	64
4.48 การทำงานของ sequence ชื่อ Delete default SEGNAM.....	65
4.49 การทำงานของ sequence ชื่อ IDoc processing.....	65

สารบัญญภาพ (ต่อ)

ภาพที่	หน้า
4.50 การทำงานของ for each activity ใน sequence ชื่อ IDoc processing.....	66
4.51 การทำงานภายใน activity switch สำหรับแต่ละ segment name	66
4.52 ตัวอย่างการทำงานของ segment name ชื่อ E1EDK01.....	67
4.53 การทำงานของ sequence ชื่อ Function module	67
4.54 การทำงานของ sequence ชื่อ Open VIM Analytics	68
4.55 การทำงานของ sequence ชื่อ Send email	68
4.56 ตัวอย่างไฟล์ข้อความที่ใช้ทดสอบ	69
4.57 ตัวอย่างอีเมลล์รายงานผลการทำ OCR	69

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

บริษัท เอ็กซอนโมบิล จำกัด เป็นบริษัทมหาชนขนาดใหญ่ที่ประกอบธุรกิจเกี่ยวกับก๊าซธรรมชาติ น้ำมันปิโตรเลียม และผลิตภัณฑ์ปิโตรเคมี เป็นบริษัทชั้นนำด้านปิโตรเลียมและปิโตรเคมีของโลก โดยการทำงานของบริษัทจะเริ่มตั้งแต่ การขุดเจาะน้ำมันดิบ แล้วนำมาผ่านกระบวนการในโรงกลั่น แล้วส่งออกจำหน่ายไปยังปั้มน้ำมันต่างๆ ภายใต้เครื่องหมายการค้าชื่อ “เอสโซ่” ซึ่งตั้งอยู่ในหลายประเทศ รวมถึงประเทศไทยด้วย

การเข้าร่วมโครงการสหกิจศึกษากับบริษัท เอ็กซอนโมบิล จำกัด ในแผนกจัดซื้อ (Cooperate process procurement team) ซึ่งเป็นแผนกที่ดูแลเกี่ยวกับการจัดซื้อและการจ่ายเงินของบริษัท ซึ่งจะเริ่มตั้งแต่กระบวนการการจัดซื้อไปจนถึงกระบวนการการจ่ายเงิน โดยลักษณะของการทำงานของทีมเรียกว่าเป็นทีมประสานงาน (Process team) ซึ่งจะทำหน้าที่ประสานงานระหว่างผู้ใช้ที่อยู่ในฝั่งธุรกิจ (Business users) และทีมนักพัฒนา (Technical team)

การทำงานในโครงการสหกิจศึกษานี้แบ่งเป็น 2 ส่วนย่อย ได้แก่

1.1.1 ระบบแชทบอท

เนื่องจากทีมต้องตอบคำถามซ้ำ ๆ บ่อยครั้ง ไม่ว่าจะเป็นคำถามที่เกี่ยวกับการค้นหาเอกสารต่าง ๆ หรือมีคำถามที่ได้รับการถามบ่อย ๆ การนำเทคโนโลยีแชทบอทมาใช้ในการช่วยตอบคำถามที่เกิดขึ้นซ้ำ ๆ นั้น จะช่วยลดเวลาที่ทีมต้องตอบคำถามซ้ำ ๆ ได้ และยังช่วยให้ทีมมีสมาธิกับการทำงานมากขึ้น โดยไม่มีสิ่งรบกวนเวลาทำงาน อีกทั้งยังเพิ่มความสะดวกสบายให้กับผู้ใช้ เพราะระบบแชทบอทนั้นสามารถทำงานได้ตลอดเวลาที่ผู้ใช้งานต้องการด้วย

1.1.2 การทดสอบโปรแกรมอัตโนมัติโดยใช้เทคโนโลยี Robotic Process Automation (RPA)

เนื่องจากการทำการทดสอบระบบของทีมแต่ละครั้งนั้นมีข้อมูลที่ต้องใช้ทดสอบจำนวนมากและใช้เวลาในการทดสอบนาน อีกทั้งยังเป็นการทำงานแบบซ้ำ ๆ การนำเทคโนโลยี RPA มาใช้นั้นจะช่วยลดเวลาในการทดสอบระบบ และยังเพิ่มประสิทธิภาพในการทำงานให้มากขึ้นอีกด้วย

1.2 วัตถุประสงค์ของการวิจัย

- 1.2.1 เพื่อลดความผิดพลาดที่เกิดขึ้นจากการทำงานของมนุษย์
- 1.2.2 เพื่อลดเวลาที่ต้องใช้ในการทำงานซ้ำ ๆ
- 1.2.3 เพื่อเพิ่มประสิทธิภาพในการทำงานให้มากขึ้น
- 1.2.4 เพื่อเพิ่มความสะดวกสบายให้กับผู้ใช้

1.3 ขอบเขตของการวิจัย

ขอบเขตของการดำเนินงานที่ได้รับมอบหมายจากบริษัทแบ่งเป็น 3 ส่วน ดังนี้

- 1.3.1 การศึกษาเกี่ยวกับปัญหาที่เกิดขึ้น
- 1.3.2 การศึกษาเทคโนโลยีที่ต้องใช้ในการพัฒนาเครื่องมือ
- 1.3.3 การพัฒนาเครื่องมือและการทดสอบเครื่องมือ

1.4 วิธีการดำเนินการวิจัย

- 1.4.1 การศึกษาเกี่ยวกับปัญหาที่เกิดขึ้น
 - 1.4.1.1 เรียนรู้กระบวนการทำงาน
 - 1.4.1.2 วิเคราะห์กระบวนการทำงาน
 - 1.4.1.3 หาเทคโนโลยีที่เหมาะสมในการพัฒนาและเพิ่มประสิทธิภาพให้เหมาะสมกับกระบวนการนั้น ๆ
- 1.4.2 การศึกษาเทคโนโลยีที่ต้องใช้ในการพัฒนาเครื่องมือ
 - 1.4.2.1 การเข้าร่วมอบรมเกี่ยวกับเทคโนโลยีต่าง ๆ
 - 1.4.2.2 การทดลองใช้เทคโนโลยี
- 1.4.3 การพัฒนาเครื่องมือและการทดสอบเครื่องมือ
 - 1.4.3.1 การเตรียมข้อมูลที่ใช้ในการทดสอบ
 - 1.4.3.2 การพัฒนาเครื่องมือ
 - 1.4.3.2 การทดสอบเครื่องมือ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากการเข้าร่วมโครงการสหกิจศึกษากับบริษัท เอ็กซอนโมบิล จำกัด แบ่งออกเป็น 2 ส่วน คือ

1.5.1 ประโยชน์ต่อบริษัท

1.5.1.1 เพื่อลดเวลาที่ต้องใช้ในการทำงานซ้ำ ๆ

1.5.1.2 เพื่อเพิ่มประสิทธิภาพในการทำงานให้มากขึ้น

1.5.2 ประโยชน์ต่อผู้วิจัย

1.5.2.1 ได้ศึกษาเทคโนโลยีใหม่ ๆ ที่ได้เคยรู้มาก่อน

1.5.2.2 เรียนรู้การวางแผนการทำงานและจัดลำดับความสำคัญในการทำงานต่าง ๆ

1.5.2.3 สามารถนำความรู้ที่ได้รับไปใช้ในอนาคตได้

บทที่ 2

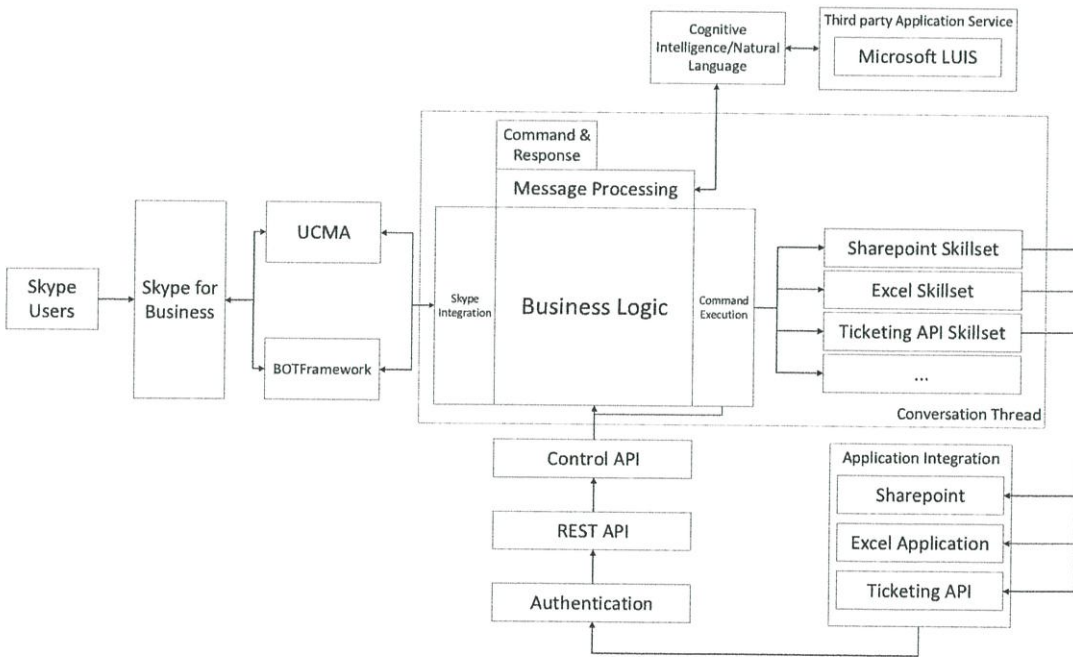
แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เนื่องจากโครงการงานสหกิจศึกษาชั้นนี้เป็นการนำเทคโนโลยีใหม่ ๆ มาปรับใช้กับงานที่ทำอยู่ในปัจจุบัน นอกจากนั้นยังจะต้องศึกษาเกี่ยวกับกระบวนการทางธุรกิจต่าง ๆ เพื่อให้ได้ประสิทธิภาพการทำงานที่สูงขึ้น ดังนั้นผู้วิจัยจึงต้องศึกษาและค้นคว้าเกี่ยวกับแนวคิดและทฤษฎีต่าง ๆ เพื่อนำมาปรับใช้กับโครงการงานสหกิจศึกษาชั้นนี้

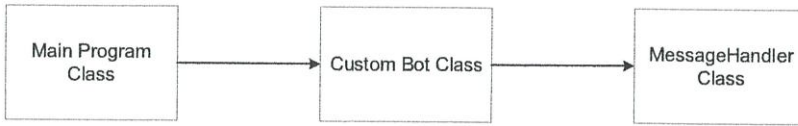
2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 SkypeX Framework

SkypeX framework คือ Framework ที่ใช้เพื่อให้โปรแกรมสามารถติดต่อกับโปรแกรม Skype for business ได้ โดยใช้ UCMA Microsoft framework และการเขียนโค้ดสำหรับการประมวลผลข้อความ (Message processing) ทำให้ SkypeX framework สามารถเปลี่ยนบทสนทนา (Conversation) เป็นอินเทนต์ (Intents) ซึ่งเป็นคำสั่งที่บอทสามารถเข้าใจและประมวลผลได้ง่าย ส่วนการประมวลผลนั้นใช้ Skill Sets ร่วมกับระบบอื่น ๆ ของบริษัท นอกจากนี้ Framework ยังทำให้บอทสามารถใช้งาน API เพื่อให้บอทสามารถติดต่อกับผู้ใช้งานได้ ภาพที่ 2.1 แสดงการทำงานทั้งหมดภายใน SkypeX framework



ภาพที่ 2.1 SkypeX Framework Block Diagram

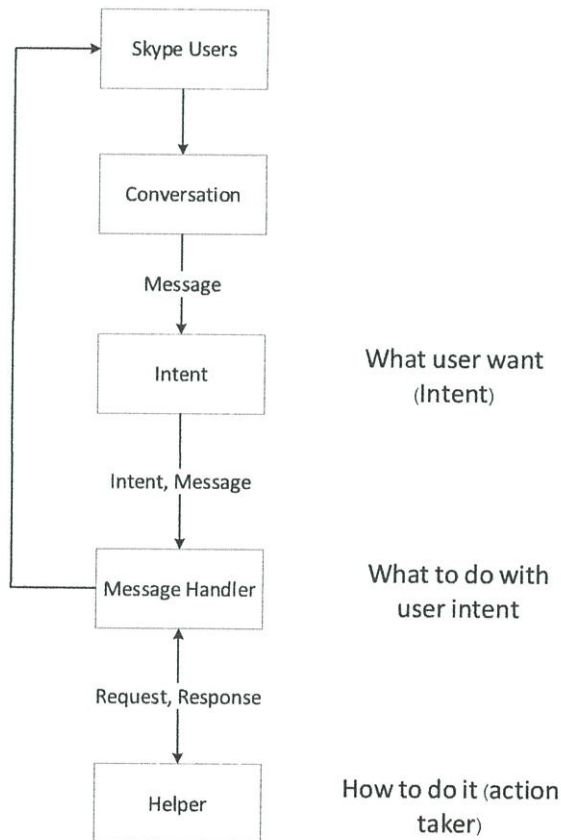


ภาพที่ 2.2 SkypeX Program Flow Diagram

จากภาพที่ 2.2 นั้น จุดประสงค์หลักของโปรแกรมหลัก (Main Program Class) คือการสร้างและรันบอท นอกจากนั้นโปรแกรมหลักยังสามารถใช้ในการเก็บประวัติการใช้ (Log) ตั้งแต่เริ่มต้นและจบโปรแกรม หรือตั้งค่าการใช้งานอื่น ๆ ได้

Custom Bot Class จะสืบทอด (Inherit) มาจาก Bot class และฟังก์ชันอื่น ๆ ที่ใช้ในการเชื่อมต่อและติดต่อกับ Skype ใน Custom Bot Class จะต้องตั้งค่าของข้อมูล (Configuration) ได้แก่ข้อมูล username และ password ในการใช้งานโปรแกรม Skype for business ของบอท โดยการใช้การยืนยันตัวตนของบอท (Bot's credentials) เพื่อให้บอทสามารถเชื่อมต่อกับ Skype และนักพัฒนาสามารถจัดการกับข้อความที่เข้ามาได้

เมื่อบอทรับข้อความมาจากผู้ใช้ บอทจะจัดการข้อความจาก Skype และ ส่งต่อไปให้กับ MessageHandler Class ซึ่ง MessageHandler Class จะเป็นตัวจัดการกับข้อความตามกฎหมายที่นักพัฒนาได้เขียนไว้ โดยอินเทนต์อ็อบเจกต์ (Intent object) จะให้ข้อมูลเกี่ยวกับข้อความที่รับเข้ามา ส่วนคอลอ็อบเจกต์ (Call object) จะแทนการสนทนา (Conversation) กับผู้ใช้ โดยสามารถใช้ส่งข้อความกลับ สิ้นสุดการสนทนา หรือฟังก์ชันต่างๆที่เกี่ยวกับการเชื่อมต่อได้ ดังภาพที่ 2.3



ภาพที่ 2.3 SkypeX MessageHandler Diagram

2.1.2 การประมวลภาษาธรรมชาติ (Natural Language Processing)

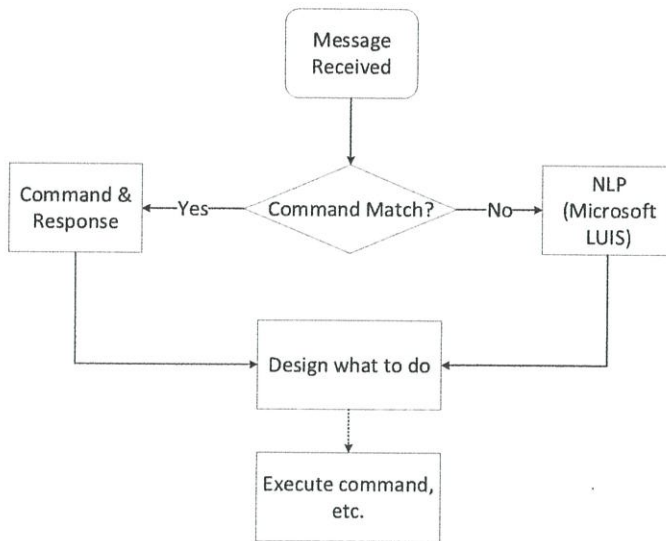
Natural Language Processing (NLP) เป็นสาขาย่อยของปัญญาประดิษฐ์ (Artificial intelligence) ที่ศึกษาเกี่ยวกับการประมวลผล การใช้งานภาษาของมนุษย์ และการทำความเข้าใจภาษามนุษย์ เพื่อให้คอมพิวเตอร์สามารถเข้าใจภาษามนุษย์ได้

2.1.3 Language Understanding Intelligent Service (LUIS)

Language Understanding Intelligent Service (LUIS) เป็นแอปพลิเคชันที่อนุญาตให้แอปพลิเคชันที่ผู้พัฒนาพัฒนาขึ้นนั้นสามารถเข้าใจสิ่งที่ผู้ใช้ต้องการได้ในภาษาของผู้ใช้เอง แอปพลิเคชัน LUIS ใช้เทคโนโลยีการเรียนรู้ของเครื่อง (Machine learning) ในการให้นักพัฒนาสามารถสร้างแอปพลิเคชันที่สามารถรับข้อมูลจากผู้ใช้ในภาษาปกติและค้นหาความหมายจากข้อมูลนั้น โดยผู้ใช้สามารถส่งข้อมูลผ่านแอปพลิเคชัน และแอปพลิเคชันนั้นจะส่งข้อมูลของผู้ใช้มาให้แอปพลิเคชัน LUIS และแอปพลิเคชัน LUIS จะประมวลผลและส่งข้อมูลกลับไปให้ผู้ใช้ผ่านแอปพลิเคชันนั้น

แอปพลิเคชัน LUIS เป็นโปรแกรมภาษาเฉพาะด้าน (domain-specific language) ที่สร้างและปรับปรุงตามสิ่งที่ผู้พัฒนาต้องการ การเริ่มต้นใช้งานแอปพลิเคชัน LUIS นั้นเริ่มต้นจากการออกแบบและสร้างโมเดล โดยใช้ข้อมูลของผู้พัฒนาเองในการเทรนโมเดลให้เหมาะสมกับข้อมูลของผู้พัฒนา การสร้างโมเดลเริ่มจากการสร้างลิส (list) ของข้อมูลผู้ใช้ จากนั้นแบ่งประเภทคำในข้อมูลเป็น อินเทนต์ (intent) และ เอ็นทิตี (entities) เมื่อการสร้างโมเดล (create) การสอนโมเดล (training) และ การนำโมเดลไปใช้ (publish) เสร็จแล้วแอปพลิเคชัน LUIS ก็พร้อมที่จะทำงาน โดยแอปพลิเคชัน LUIS จะรับข้อมูลของผู้ใช้ผ่าน HTTP request และส่งข้อมูลการประมวลผลของแอปพลิเคชัน LUIS กลับไปเป็น JSON object ให้โปรแกรมหรือแอปพลิเคชันนั้นจัดการกับข้อมูลต่อไป

การใช้งานแอปพลิเคชัน LUIS ร่วมกับการเขียนกฎแบบ Rule-based นั้น จะเริ่มจากตรวจสอบว่าข้อความที่รับเข้ามานั้นมีกฎรองรับสำหรับการประมวลผลหรือไม่ ถ้าไม่รองรับจะส่งให้แอปพลิเคชัน LUIS ประมวลผลว่าต้องใช้กฎใดในการประมวลผลข้อความนั้น จากนั้นแอปพลิเคชัน LUIS จะตอบกลับมาที่แอปพลิเคชันว่าต้องใช้กฎใดในการประมวลผลข้อความนั้น และให้แอปพลิเคชันใช้กฎนั้นประมวลผลข้อความต่อไป ดังภาพที่ 2.4



ภาพที่ 2.4 การทำงานของ Natural Language Processing ร่วมกับ Rule-based

2.1.4 เอพีไอ (application programming interface: API)

เอพีไอ หมายถึง วิธีการที่ระบบปฏิบัติการหรือไลบรารีเปิดให้โปรแกรมคอมพิวเตอร์สามารถเรียกใช้งานได้จากในฝั่งลูกข่าย (Client) และทำงานที่ฝั่งแม่ข่าย (Server) โดยหน้าที่หลักของ API คือคอยรับคำสั่งจากฝั่งลูกข่าย ที่เป็นแอปพลิเคชันต่างๆ เมื่อเกิดคำสั่งใด ๆ จากฝั่งลูกข่ายแล้ว API จะรับคำสั่งนั้นไปประมวลผล และส่งข้อมูลที่ตรงกับกรร้องขอกลับไปฝั่งลูกข่ายเรียกว่า Response โดยข้อมูลที่ถูกส่งกลับมาส่วนใหญ่จะอยู่ในรูปแบบของ JSON Object หรือ XML เป็นต้น API แบ่งเป็น 2 แบบ คือ

- เอพีไอที่ขึ้นกับภาษา (language-dependent API) คือ API ที่สามารถเรียกใช้จากโปรแกรมที่เขียนขึ้นด้วยภาษาใดภาษาหนึ่ง
- เอพีไอไม่ขึ้นกับภาษา (language-independent API) คือ API ที่สามารถเรียกได้จากโปรแกรมหลาย ๆ ภาษา

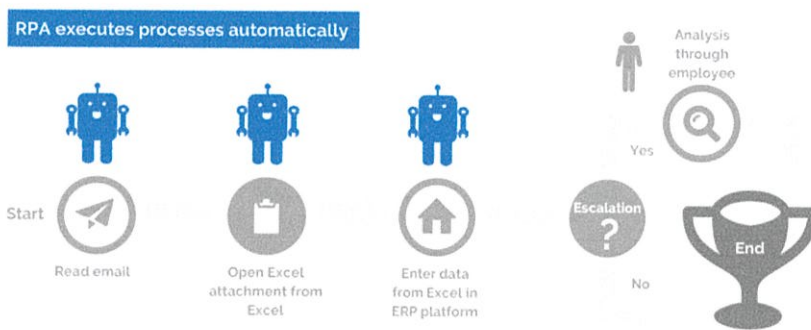
2.1.5 Robotic Process Automation (RPA)

Robotic process automation (RPA) คือ ซอฟต์แวร์ในการทำกระบวนการอัตโนมัติ (automated process) ที่ปกติแล้วจะทำด้วยคนเปลี่ยนมาเป็นการทำงานแบบอัตโนมัติ เครื่องมือ RPA นี้สามารถทำให้มีการแปลงกระบวนการที่ต้องใช้คนในการทำ (Manual process) ไปเป็นอาร์พีเอสคริปต์ (RPA script) โดยทำงานเหมือนวิธีที่คนใช้แอปพลิเคชันนั้น ๆ ตามภาพที่ 2.5 โดยปกติแล้ว

อาร์พีเอสคริปต์จะทำงานที่ชั้นส่วนต่อประสานกับผู้ใช้ (user interface layer) เช่น คลิกรูป, พิมพ์ข้อความ, คัดลอกและวางข้อมูลระหว่างแอปพลิเคชัน

ประโยชน์ของการใช้งานเทคโนโลยี RPA ได้แก่

- ลดต้นทุนการทำงาน (Lower Costs)
- สามารถทำงานได้ตลอดเวลา (24/7 Availability)
- ลดความผิดพลาดจากการทำงานโดยมนุษย์ (Eliminate Errors)
- สามารถเพิ่มกำลังการทำงานได้อย่างรวดเร็ว (Scale Up Quickly)
- สามารถใช้วิเคราะห์ข้อมูลขั้นสูงได้ (Enables Advanced Analytics)

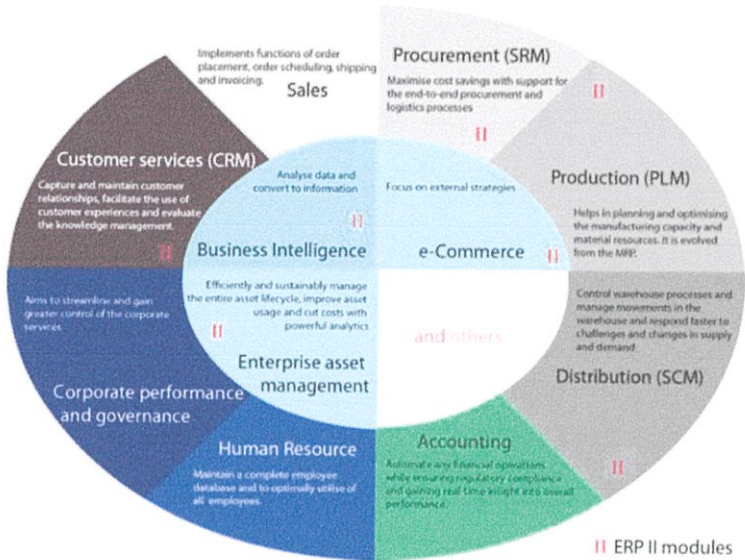


ภาพที่ 2.5 ตัวอย่างวิธีการทำงานของ RPA

(ที่มา : <https://www.roboyo.de/en/robotic-process-automation/>)

2.1.6 การวางแผนทรัพยากรองค์กร (Enterprise Resource Planning)

การวางแผนทรัพยากรองค์กร (Enterprise Resource Planning หรือ ERP) คือ เครื่องมือที่นำมาใช้ในการบริหารทางธุรกิจ เนื่องจากในหนึ่งบริษัทจะประกอบไปด้วยหลาย ๆ แผนก เช่น แผนกจัดซื้อ แผนกบัญชี แผนกการผลิต และแผนกบุคคล เป็นต้น ซึ่งในแต่ละแผนกก็จะมีข้อมูลอยู่มากมาย ซึ่งข้อมูลนั้นอาจจะต้องนำไปใช้ในแผนกอื่นด้วย ทำให้เกิดซอฟต์แวร์ที่ใช้ในการดูแลและจัดการข้อมูลให้เป็นระบบ เพื่อเป็นการดูแลและบริหารข้อมูลเหล่านั้นให้มีประสิทธิภาพ และยังป้องกันการซ้ำซ้อนกันของข้อมูล และยังช่วยในการวางแผนทรัพยากรต่าง ๆ ของธุรกิจอีกด้วย ดังภาพที่ 2.6 นั้นจะเห็นได้ว่า ERP นั้นประกอบด้วยหลาย ๆ โมดูลสำหรับแต่ละแผนก โดยแต่ละโมดูลนั้นก็ใช้ในงานที่แตกต่างกันออกไป

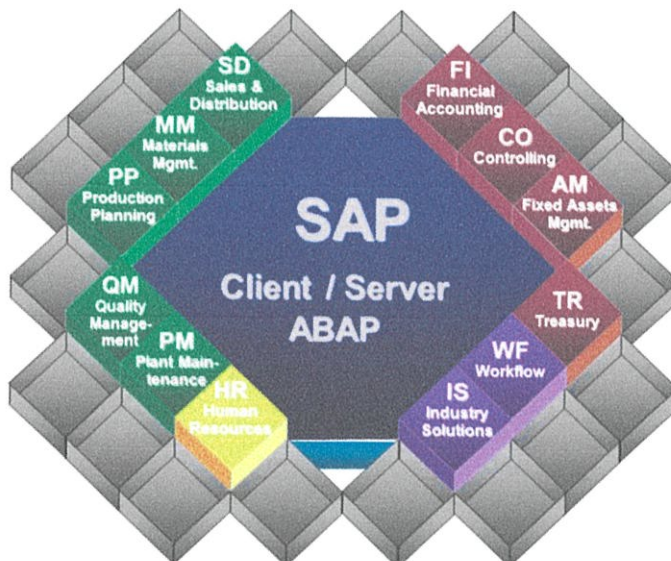


ภาพที่ 2.6 ERP modules Diagram

(ที่มา : https://en.wikipedia.org/wiki/Enterprise_resource_planning)

2.1.7 เอสเอพี (Systems, Applications and Products: SAP)

SAP (System Application and Products in Data Processing) เป็นโปรแกรมทางธุรกิจประเภท ERP ชั้นนำตัวหนึ่งของประเทศเยอรมนี SAP เป็นระบบที่เชื่อมโยงข้อมูลของโมดูลการทำงานย่อย ๆ ทั้งหมดเข้าด้วยกัน ทำให้ข้อมูลสามารถเชื่อมโยงถึงกันได้ ช่วยลดปัญหาการบันทึกข้อมูลซ้ำ นอกจากนี้ยังสามารถทำให้ประสิทธิภาพการทำงานในธุรกิจนั้นเพิ่มขึ้นอีกด้วย



ภาพที่ 2.7 SAP ERP Application Modules

(ที่มา : <https://sap-certification.info/how-does-sap-work>)

จากภาพที่ 2.7 นั้นโมดูลของโปรแกรม SAP (SAP Modules) นั้นจะประกอบไปด้วย 12 โมดูล ได้แก่

1. Sales and Distribution (SD) เป็นโมดูลที่เกี่ยวกับกระบวนการการขายสินค้า
2. Material Management (MM) เป็นโมดูลที่ดูแลและบริหารวัตถุดิบ
3. Production Planning (PP) เป็นโมดูลเกี่ยวกับการวางแผนการผลิต
4. Quality Management (QM) เป็นโมดูลเกี่ยวกับการตรวจสอบคุณภาพ
5. Plant Maintenance (PM) เป็นโมดูลเกี่ยวกับการดูแลและซ่อมบำรุงโรงงาน
6. Human Resource (HR) เป็นโมดูลด้านการจัดการและการบริหารทรัพยากรบุคคล
7. Financial Account (FI) เป็นโมดูลที่ดูแลทางด้านบัญชีและการเงินทั้งหมดของบริษัท
8. Controlling (CO) เป็นโมดูลที่ดูแลด้านการควบคุมและการจัดการภายในบริษัท ทำบัญชีเพื่อเสนอข้อมูลที่ใช้ในการดำเนินงานภายในบริษัท
9. Asset Management (AM) เป็นโมดูลที่ดูแลและบริหารสินทรัพย์ถาวรต่าง ๆ เช่น เครื่องจักร เป็นต้น
10. Treasury (TR) เป็นโมดูลที่ดูแลทางด้านการบริหารการเงิน
11. Workflow (WF) เป็นโมดูลของกระบวนการทำงานแบบอัตโนมัติ
12. Industry Solutions (IS) เป็นโมดูลเฉพาะของธุรกิจนั้น ๆ ซึ่งไม่ใช่โมดูลมาตรฐานของ SAP

2.1.8 การพัฒนาซอฟต์แวร์แบบเอจายล์ (Agile Software Development)

การพัฒนาซอฟต์แวร์แบบเอจายล์ คือ เป็นแนวคิดแบบใหม่ที่ใช้ในการพัฒนาซอฟต์แวร์ที่เน้นการพัฒนาแบบยืดหยุ่น สามารถรองรับการเปลี่ยนแปลงได้ตลอดเวลา มีการพัฒนาแบบรวดเร็ว การพัฒนาซอฟต์แวร์แบบเอจายล์นั้นออกมาเพื่อแก้ปัญหาของวิธีการพัฒนาแบบ Waterfall ที่มีความยากในการวางแผนเพื่อให้ทุกอย่างลงตัวและไม่ยืดหยุ่นกับความเปลี่ยนแปลงต่าง ๆ เช่น ผู้ใช้สามารถทดสอบซอฟต์แวร์ได้เมื่อการพัฒนาทั้งหมดเสร็จแล้วเท่านั้น ทำให้ถ้าเกิดการแก้ไขบางอย่างก็เป็นไปได้ยาก ถ้าใช้การพัฒนาซอฟต์แวร์แบบเอจายล์นั้นจะส่งงานให้กับผู้ใช้บ่อย ๆ ให้ผู้ใช้สามารถทดสอบได้เรื่อย ๆ ว่าเป็นไปตามที่ผู้ใช้งานต้องการหรือไม่ ทำให้การแก้ไขเปลี่ยนแปลงนั้นทำได้ง่ายและรวดเร็วกว่า framework ที่เป็นที่ยอมรับได้แก่ Scrum และ Kanban

เจตนารมณ์ของการพัฒนาซอฟต์แวร์แบบเอจายล์ (Manifesto for Agile Software Development)

- คนและการมีปฏิสัมพันธ์กัน สำคัญมากกว่า การทำตามขั้นตอนและเครื่องมือ
- ซอฟต์แวร์ที่นำไปใช้งานได้จริง สำคัญมากกว่า เอกสารที่ครบถ้วนสมบูรณ์
- ร่วมมือทำงานกับลูกค้า สำคัญมากกว่า การต่อรองให้เป็นไปตามสัญญา
- การตอบรับกับการเปลี่ยนแปลง สำคัญมากกว่า การทำตามแผนที่วางไว้

2.1.9 DevOps

DevOps (เป็นการรวมกันของคำว่า “development” และ “operations”) คือการปฏิบัติงานด้านซอฟต์แวร์ มีจุดมุ่งหมายเพื่อรวมการพัฒนาซอฟต์แวร์ (Dev) และกระบวนการการทำงาน of ซอฟต์แวร์ (Ops) คุณลักษณะสำคัญของ DevOps คือ สนับสนุนให้ทำ automation และมีการทำ monitoring ในทุกขั้นตอนของการสร้าง software ตั้งแต่การทำ integration, testing, releasing จนถึงขั้นตอนการ deployment และการบริหารจัดการระบบ (infrastructure)

DevOps มีเป้าหมายในการพัฒนาดังนี้

- ทำให้เวลาที่ใช้ในการพัฒนา (development cycles) สั้นลง
- เพิ่มความถี่ในการ deployment
- มี releases ที่มีความน่าเชื่อถือมากขึ้น
- ทำให้ซอฟต์แวร์มีความใกล้เคียงกับความต้องการของธุรกิจ (business objectives)

2.1.10 User-centered design (UCD)

การออกแบบที่มุ่งเน้นผู้ใช้เป็นศูนย์กลาง User-centered design (UCD) หรือ user-driven development (UDD) เป็น framework ของกระบวนการการผลิตที่ไม่ได้ขึ้นกับ อินเทอร์เน็ต (interface) หรือ เทคโนโลยี (technology) ซึ่งจะกำหนด เป้าหมายการใช้งาน คุณลักษณะของผู้ใช้ สภาพแวดล้อม งาน และ เวิร์คโฟลว์ (workflow) ของผลิตภัณฑ์และบริการ และให้ความสนใจอย่างมากในแต่ละขั้นตอนของกระบวนการออกแบบ

UCD เป็นขั้นตอนการแก้ปัญหาหลายขั้นตอนซึ่งไม่เพียงแต่ต้องใช้การออกแบบในการวิเคราะห์ถึงวิธีที่ผู้ใช้มีแนวโน้มที่จะใช้ผลิตภัณฑ์ แต่ยังคงตรวจสอบสมมติฐานเกี่ยวกับพฤติกรรมของผู้ใช้ในการทดสอบจริง

การทดสอบเหล่านี้อาจดำเนินการโดยผู้ใช้งานจริงหรือไม่ก็ได้ ในแต่ละขั้นตอนของกระบวนการตั้งแต่ข้อกำหนดรูปแบบการผลิต ก่อนการผลิต และหลังการผลิตเสร็จสมบูรณ์ เพื่อให้มั่นใจว่า การพัฒนานั้นดำเนินไปโดยมีการคำนึงถึงผู้ใช้งานเป็นหลัก

ความแตกต่างหลักจากปรัชญาการออกแบบผลิตภัณฑ์แบบอื่นๆ คือ UCD เป็นการออกแบบเพื่อเพิ่มประสิทธิภาพผลิตภัณฑ์โดยคำนึงถึงความต้องการของผู้ใช้ แทนที่จะบังคับให้ผู้ใช้งานเปลี่ยนพฤติกรรมเพื่อให้เหมาะกับผลิตภัณฑ์

2.2 โปรแกรมที่ใช้

2.2.1 Skype for Business

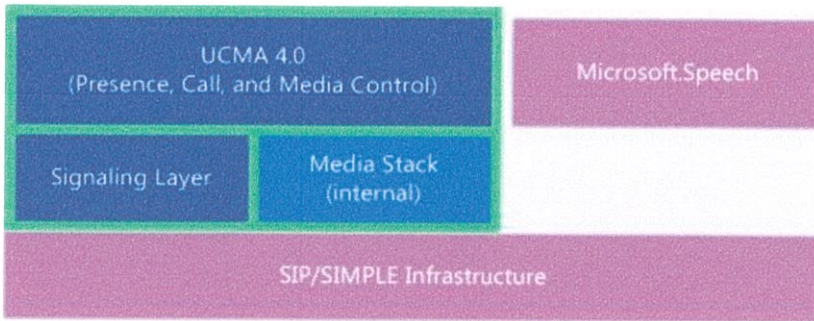
โปรแกรม Skype for Business (ชื่อเดิมคือ Microsoft Office Communicator และ Microsoft Lync) เป็นโปรแกรมสำหรับการรับส่งข้อความโต้ตอบแบบทันที (instant-messaging client) ชนิดหนึ่ง โปรแกรม Skype for Business เป็นซอฟต์แวร์สำหรับองค์กร เมื่อเทียบกับ Skype จะมีคุณลักษณะต่าง ๆ เพิ่มเติมขึ้นมาสำหรับเพื่อใช้ในการทำงานด้านธุรกิจ

2.2.2 Microsoft Unified Communications Managed API 4.0 (UCMA 4.0)

Microsoft Unified Communications Managed API 4.0 ทำให้ผู้พัฒนาสามารถสร้างแอปพลิเคชันโดยใช้ประโยชน์จาก Microsoft Lync Server 2013 platform ได้อย่างเต็มที่ ตามโครงสร้างในภาพที่ 2.8 ตัวอย่างแอปพลิเคชันที่สร้างบน UCMA 4.0 และใช้งานร่วมกับ unified communications concepts เช่น presence, call, conversation, และ conference ตามภาพที่ 2.9

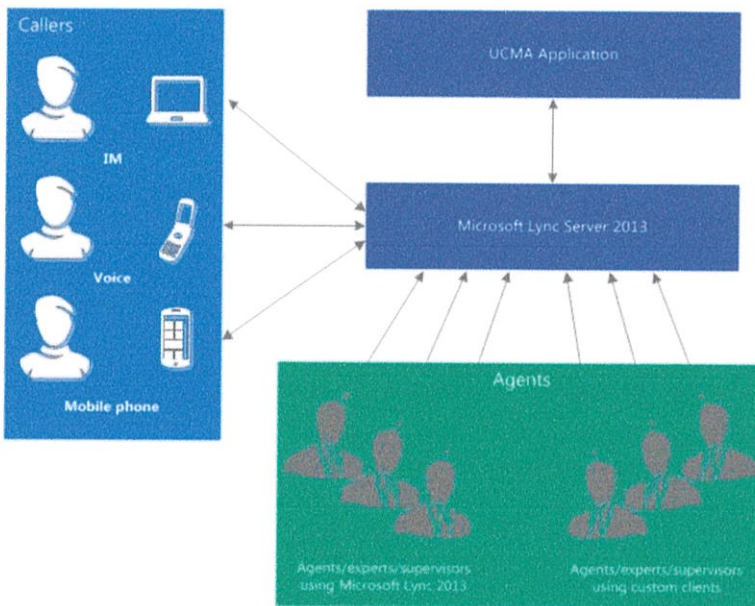
UCMA 4.0 เป็น software-based มากกว่า hardware-based ทำให้ง่ายต่อการเพิ่มหรือแก้ไข feature ต่างๆ แอปพลิเคชันเหล่านี้จึงสามารถตอบสนองความต้องการของผู้ใช้ (specific requirements) ได้ดี

UCMA 4.0 เป็น C# API ที่มีการรวม development และ runtime components เข้าไว้ด้วยกัน นอกจากนี้ UCMA 4.0 ทำให้นักพัฒนาสามารถพัฒนา communication solutions สำหรับองค์กรได้ง่ายขึ้นอีกด้วย



ภาพที่ 2.8 โครงสร้างการใช้งาน UCMA 4.0

(ที่มา : <https://msdn.microsoft.com/en-us/library/office/dn465943.aspx>)



ภาพที่ 2.9 ตัวอย่างการใช้งาน UCMA 4.0 ใน contact center application

(ที่มา : <https://msdn.microsoft.com/en-us/library/office/dn465936.aspx>)

2.2.3 Visual Studio

Microsoft Visual Studio เป็น integrated development environment (IDE) จาก Microsoft ใช้สำหรับพัฒนาโปรแกรมคอมพิวเตอร์สำหรับ Microsoft Windows เช่นเดียวกับเว็บไซต์ เว็บแอปพลิเคชัน เว็บเซอร์วิสและแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ โปรแกรม Visual Studio ใช้แพลตฟอร์มการพัฒนาซอฟต์แวร์ของ Microsoft เช่น Windows API, Windows Forms, Windows Presentation Foundation, Windows Store และ Microsoft Silverlight สามารถเขียนได้ทั้งแบบ native code และ managed code

Visual Studio ประกอบด้วย code editor supporting IntelliSense (the code completion component) และการทำ code refactoring การทำดีบั๊กแบบบูรณาการ (integrated debugger) ทำงานได้ทั้งในรูปแบบดีบั๊กเกอร์ระดับต้นและดีบั๊กเกอร์ระดับเครื่อง (source-level debugger and a machine-level debugger) เครื่องมือที่มีอยู่แล้วในตัวอื่น ๆ (built-in tools) ได้แก่ code profiler, forms designer for building GUI applications, web designer, class designer, และ database schema designer ยอมรับปลั๊กอินที่ช่วยเพิ่มฟังก์ชันการทำงานในเกือบทุกระดับ รวมถึงการเพิ่มการสนับสนุนระบบ source control เช่นการทำ subversion และเพิ่มชุดเครื่องมือใหม่ ๆ เช่น โปรแกรมแก้ไขและนักออกแบบภาพสำหรับภาษาหรือชุดเครื่องมือเฉพาะสำหรับโดเมน (editors and visual designers for domain-specific languages) หรือชุดเครื่องมือสำหรับด้านอื่น ๆ ของการพัฒนาซอฟต์แวร์ เช่น Team Foundation Server: Team Explorer

Visual Studio สนับสนุนโปรแกรมภาษาต่างๆ 36 ภาษาและโปรแกรมสำหรับการแก้ไขโค้ดและดีบั๊กเกอร์สามารถสนับสนุนภาษาเขียนโปรแกรมได้เกือบทุกภาษา สำหรับภาษาที่มีรองรับอยู่แล้ว (Built-in languages) ได้แก่ C, C++ และ C++/CLI, VB.NET, C#, F# และ TypeScript ส่วนภาษาอื่นๆ เช่น Python, Ruby, Node.js สามารถใช้งานได้โดยการติดตั้งแยกเพิ่มเติม

2.2.4 Team Foundation Server

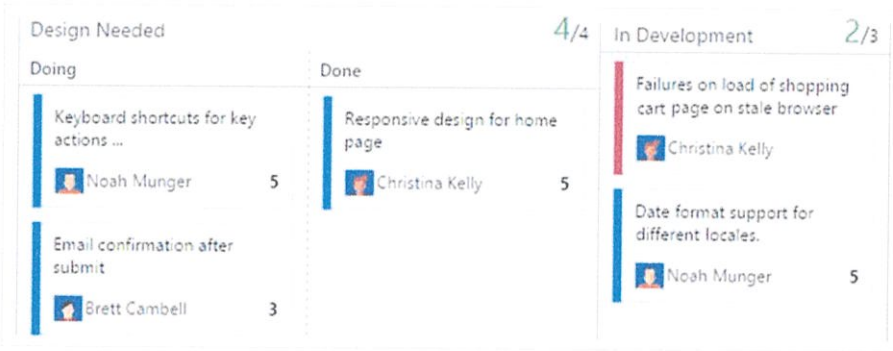
Team Foundation Server คือเครื่องมือที่มีความสามารถในการทำ source control, work item tracking, Team Foundation Build, a team project portal Web site, reporting, and project management ได้ตามตัวอย่างในภาพที่ 2.10 ซึ่งเป็นการทำ version control และภาพที่ 2.11 เป็นตัวอย่างการแสดงผลงานในรูปแบบของ Kanban board

นอกจากนี้ Team Foundation Server ยังมีคลังข้อมูล (data warehouse) ที่เก็บข้อมูลจาก work item tracking, source control, builds และ testing tools ได้อีกด้วย



ภาพที่ 2.10 ตัวอย่างการทำ Version control ใน TFS

(ที่มา : <https://www.visualstudio.com/tfs/>)



ภาพที่ 2.11 ตัวอย่างการแสดง Kanban board ใน TFS

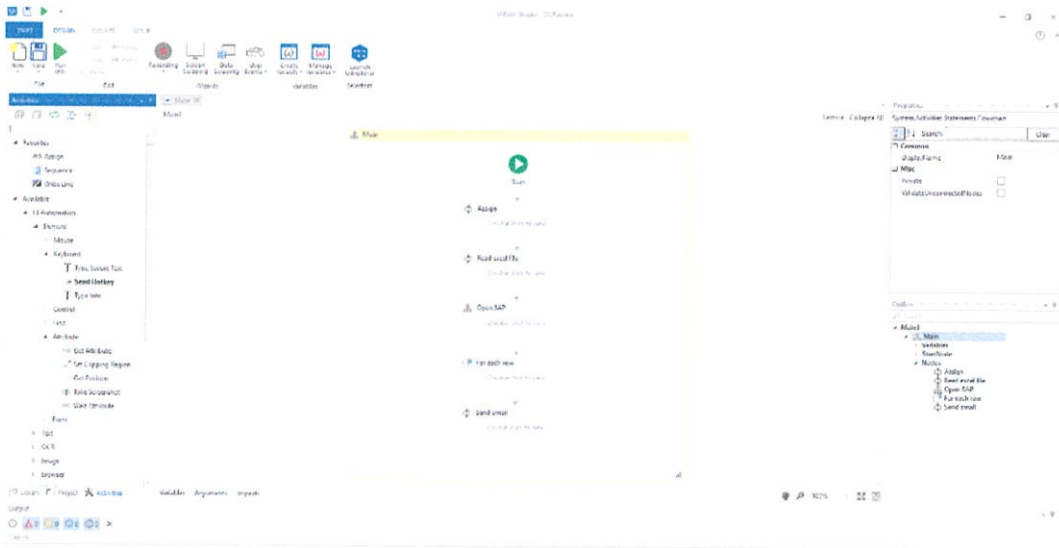
(ที่มา : <https://www.visualstudio.com/tfs/>)

2.2.5 UiPath Studio

UiPath Studio คือ โปรแกรมที่ใช้ทำระบบอัตโนมัติโดย UiPath Studio นั้นจะช่วยให้สามารถทำงานตามกฎที่เขียนขึ้น (rule-based processes) ได้โดยอัตโนมัติ โดยการควบคุมลำดับการทำงานและความสัมพันธ์ระหว่างการทำงานแต่ละขั้นตอนนี้สามารถกำหนดได้โดยผู้ใช้งาน โดยลำดับการทำงานและความสัมพันธ์นี้เรียกว่า activities โดยแต่ละ activities ประกอบด้วย action หลาย ๆ action เช่น การคลิกปุ่มอ่านไฟล์หรือเขียนข้อมูลลงในไฟล์ โดยมีโลโก้ของโปรแกรมตามภาพที่ 2.12 และตัวอย่างหน้าโปรแกรมตามภาพที่ 2.13



ภาพที่ 2.12 logo โปรแกรม UiPath Studio



ภาพที่ 2.13 ตัวอย่างโปรแกรม UiPath Studio

2.3 ภาษาโปรแกรมที่ใช้

2.3.1 ภาษา C#

C# เป็นภาษาที่มีรากฐานมาจากภาษา C และมีความคล้ายคลึงกับภาษา C, C++, Java และ JavaScript ได้รับการพัฒนาโดย Microsoft ภายใต้แนวคิดของ .NET C# เป็นหนึ่งในภาษาเขียนโปรแกรมที่ออกแบบมาสำหรับ Common Language Infrastructure

2.3.2 ภาษา Visual Basic .NET

Visual Basic .NET (VB.NET) เป็นภาษาเชิงวัตถุ (object-oriented programming language) ซึ่งใช้กับ .NET Framework แม้ว่าจะพัฒนามาจากภาษา Visual Basic แต่ไม่สามารถใช้ร่วมกันกับโค้ดที่compile โดยใช้ VB.NET ได้

บทที่ 3

วิธีดำเนินการวิจัย

3.1 วางแผนการทำงาน

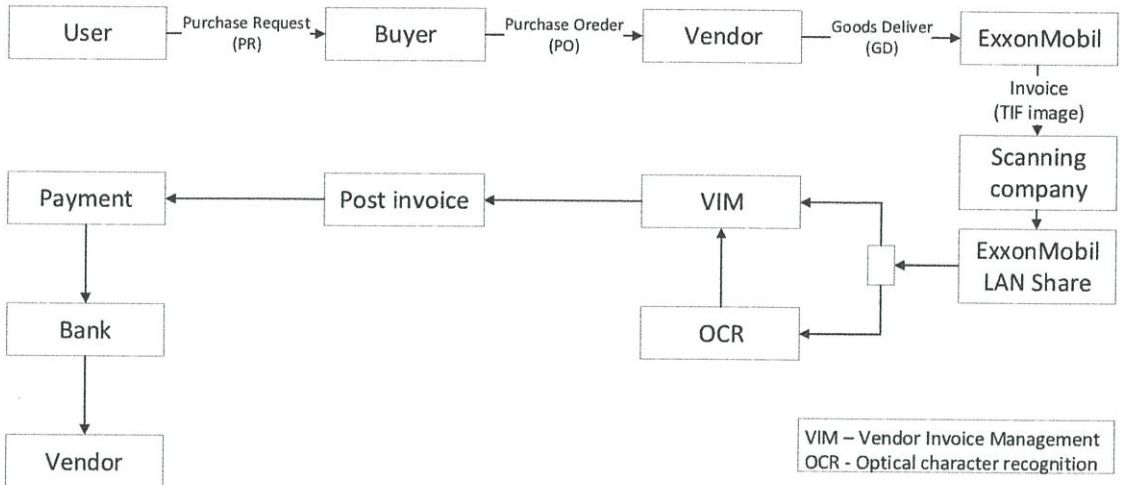
แผนการทำงานของโครงการสหกิจศึกษาชั้นนี้ แบ่งเป็น 3 ส่วนหลัก ได้แก่ การพัฒนาการทดสอบโปรแกรมอัตโนมัติโดยใช้เทคโนโลยี RPA การพัฒนาระบบแซทบอท และการทำเอกสาร โดยจะเริ่มต้นจากการพัฒนาระบบแซทบอทก่อน ซึ่งจะเป็นการเริ่มจากขั้นตอนการเก็บข้อมูลความต้องการ ศึกษาวิธีการจัดทำระบบแซทบอท และเริ่มต้นพัฒนาระบบแซทบอทบางส่วนก่อน จากนั้นจะเริ่มพัฒนาการทดสอบโปรแกรมอัตโนมัติโดยใช้เทคโนโลยี RPA โดยการทดสอบโปรแกรมอัตโนมัติโดยใช้เทคโนโลยี RPA นั้นจะมีการพัฒนา 2 เครื่องมือ คือการทดสอบบันทึกใบแจ้งหนี้แบบใช้ Vendor invoice management (VIM) และแบบใช้ Optical character recognition (OCR) เมื่อพัฒนาทั้งสองระบบเสร็จแล้ว จึงพัฒนาระบบแซทบอทต่อเพื่อให้มีฟังก์ชันการทำงานอื่นๆ เพิ่มขึ้น

ตารางที่ 3.1 แผนการทำงานโครงการสหกิจศึกษา

	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน				ธันวาคม			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
RPA																				
ระบบแซทบอท																				
จัดทำเอกสาร																				

3.2 การศึกษาขั้นตอนการทำงาน

กระบวนการการซื้อของในบริษัท เอ็กซอนโมบิล ตามภาพที่ 3.1 นั้น จะเริ่มตั้งแต่ผู้ใช้ต้องการซื้อของ จึงส่งใบสั่งซื้อ (Purchase request: PR) ไปให้กับผู้จัดซื้อ (Buyer) จากนั้นผู้จัดซื้อจะส่งใบจัดซื้อ (Purchase order: PO) ไปยังบริษัทผู้ผลิต เมื่อผู้ผลิตได้ดำเนินการจัดส่งของ (Goods deliver: GD) กลับมาให้กับบริษัท และผู้ผลิตจะส่งใบแจ้งหนี้ (invoice) กลับมายังบริษัท เพื่อให้บริษัททำการจ่ายเงิน โดยบริษัทจะส่งใบแจ้งหนี้ให้กับบริษัทแสกน (scanning company) เพื่อทำการสแกนใบแจ้งหนี้ให้เป็นรูปภาพ จากนั้นส่งกลับมาที่ LAN Share จากนั้นจะแยกเป็น 2 ส่วนได้แก่ การใช้ VIM ในการบันทึกใบแจ้งหนี้ลงไปในระบบ SAP เอง หรือผ่านการใช้ OCR เพื่อบันทึกใบแจ้งหนี้ เมื่อมีการบันทึกใบแจ้งหนี้แล้วจะมีการส่งไฟล์ไปให้ธนาคาร เพื่อให้ธนาคารดำเนินการจ่ายเงินผู้ผลิตต่อไป



ภาพที่ 3.1 กระบวนการการจัดซื้อและการจ่ายเงินทั้งหมด

3.2.1 การศึกษาขั้นตอนการบันทึกใบแจ้งหนี้แบบใช้ VIM

ในการบันทึกใบแจ้งหนี้ (invoice) ลงในระบบแบบมาตรฐานของระบบ SAP จะไม่มีขั้นตอนการทำงาน (workflow) ดังนั้น VIM จะเป็นระบบที่ช่วยในการนำขั้นตอนการทำงาน หรือ workflow มาใช้ร่วมกับการบันทึกใบแจ้งหนี้ลงในระบบ SAP โดยที่ VIM จะเป็นส่วนติดต่อประสานกับผู้ใช้เพื่อช่วยให้ผู้ใช้สามารถบันทึกข้อมูลผ่าน VIM แล้ว VIM จะทำการคำนวณ ประมวลผล และสร้าง workflow ส่งไปหาผู้เกี่ยวข้อง เช่น ส่ง workflow นั้นให้ผู้ที่ทำหน้าที่อนุมัติ หรือเรียกว่า Approval ทำการอนุมัติใบแจ้งหนี้ (approve) เมื่อทำงานเสร็จแล้วจะบันทึกใบแจ้งหนี้กลับไปทีระบบ SAP

ขั้นตอนการทำบันทึกใบแจ้งหนี้แบบใช้ VIM มีดังนี้

1. เปิดโปรแกรม SAP และเข้าระบบที่ต้องการ
2. พิมพ์ t-code : zf_vim_oawd เพื่อเปิดหน้า Process scanned incoming vendor invoices โดยเลือก business line เป็น PTP Payables ตามภาพที่ 3.2

Process Scanned Incoming Vendor Invoices

Please enter the following scan batch attributes:

Business Line (Mandatory)

Indexer ID

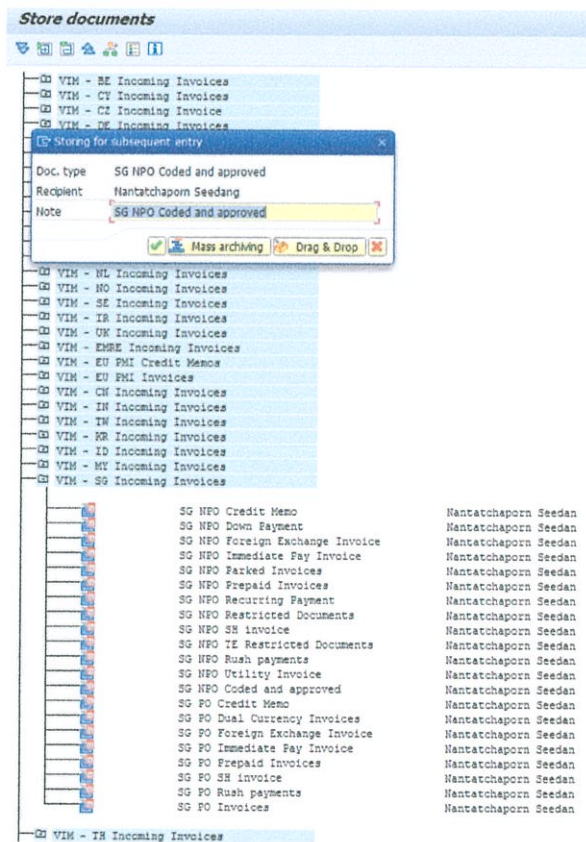
Invoice Receipt Date / Time / 00:00:00

Scan Note

Scan Location

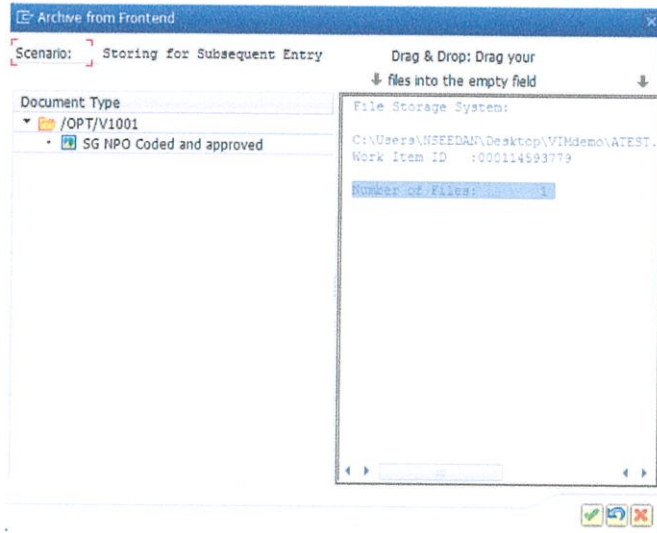
ภาพที่ 3.2 หน้า Process scanned incoming vendor invoices

3. เลือกประเทศและประเภทของ invoice จากนั้นคลิก drag & drop ตามภาพที่ 3.3



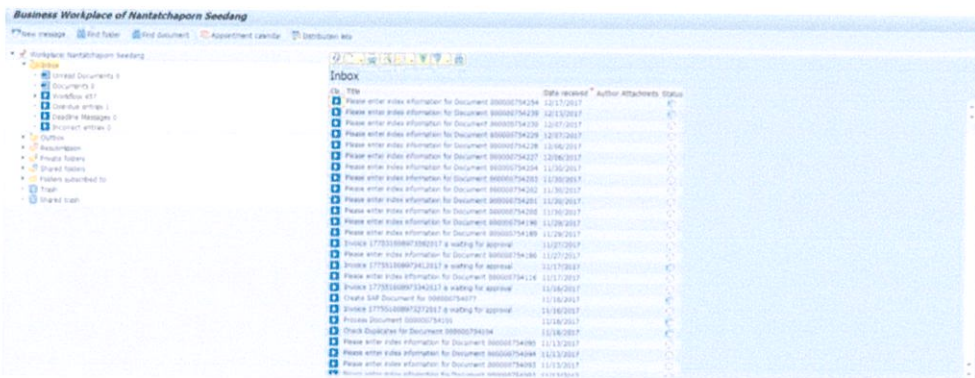
ภาพที่ 3.3 หน้า Storing documents

4. เลือกgrup invoice มาวางในโปรแกรม SAP ตามภาพที่ 3.4



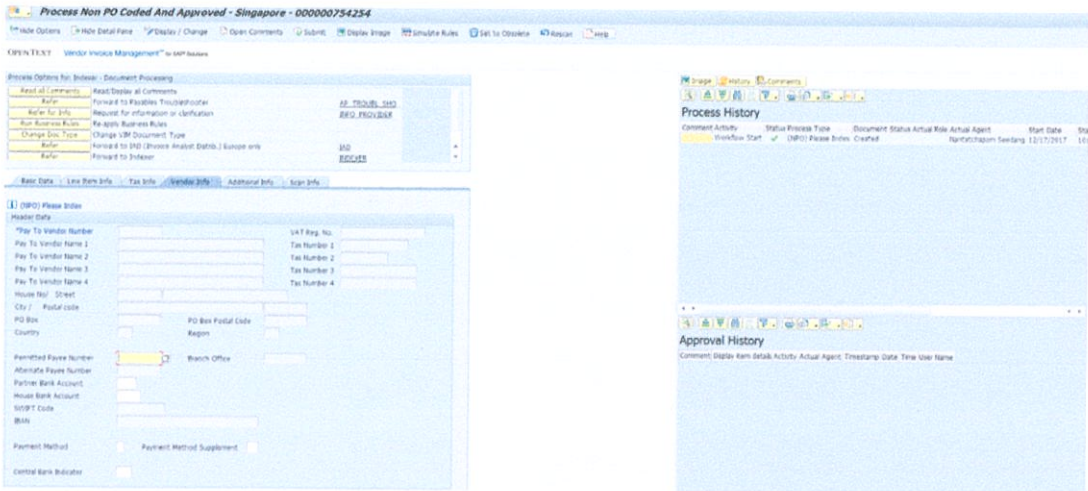
ภาพที่ 3.4 หน้า Drag & Drop

5. เปิดหน้า Business workplace โดยพิมพ์ t-code : sbwp แล้วเลือกใบแจ้งหนี้ที่ต้องการใส่ข้อมูลตามในภาพที่ 3.5

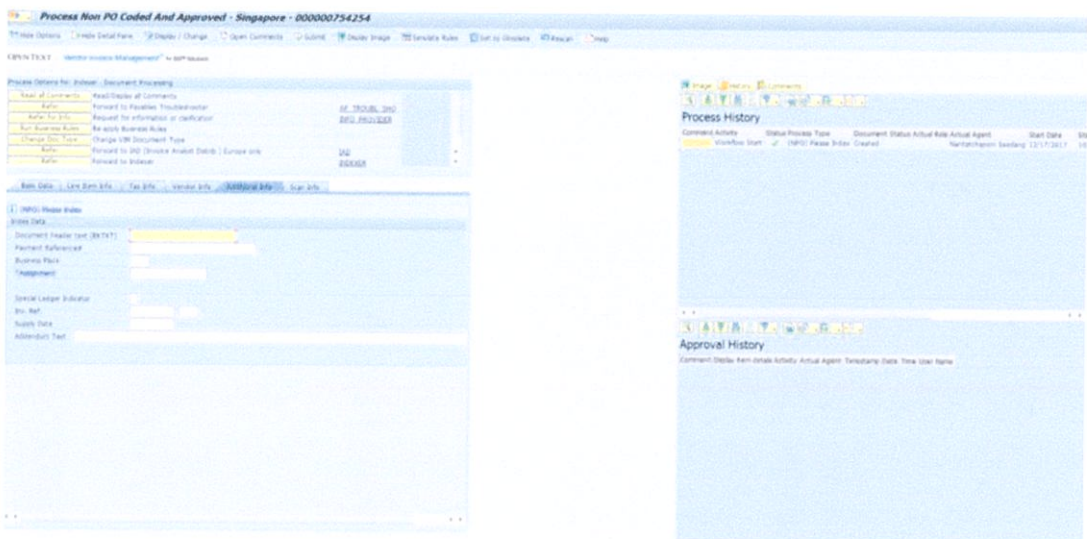


ภาพที่ 3.5 หน้า Business workplace

6. ใส่ข้อมูลของใบแจ้งหนี้ นั้น ได้แก่ ข้อมูลทั่วไป ตามภาพที่ 3.6, ข้อมูล line item ตามภาพที่ 3.7 ข้อมูลของผู้ขาย ตามภาพที่ 3.8 และข้อมูลเพิ่มเติมอื่นๆ ตามภาพที่ 3.9 เพื่อบันทึกใบแจ้งหนี้ลงในระบบ SAP

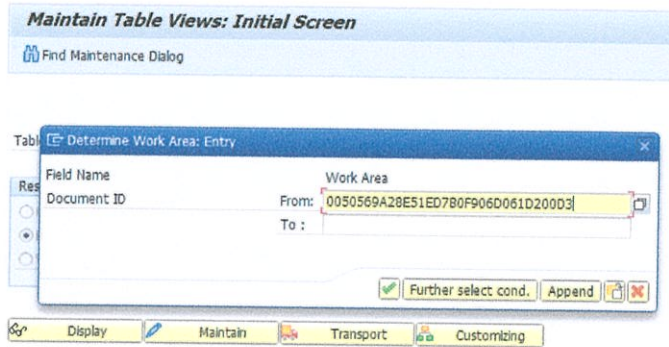


ภาพที่ 3.8 หน้าโปรแกรม VIM (Vendor Info)



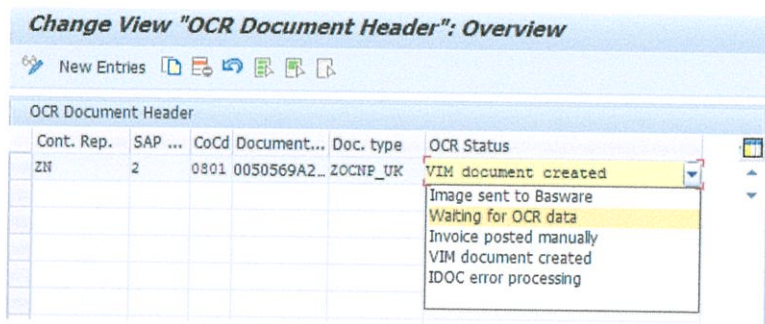
ภาพที่ 3.9 หน้าโปรแกรม VIM (Additional Info)

4. เปิดหน้า VIM Analytics โดยใช้ t-code : /opt/vim_analytics โดยการใส่เลขของ VIM ที่ช่อง Document processing number ดังภาพที่ 3.10 เพื่อให้สามารถดูสถานะของเอกสาร VIM ที่ได้บันทึกข้อมูลไปแล้วได้ ตามภาพที่ 3.11



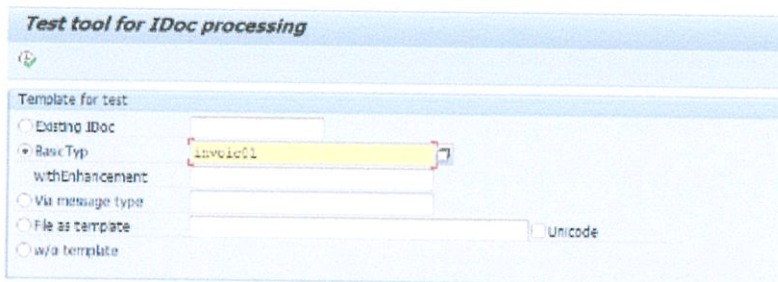
ภาพที่ 3.16 หน้า Maintain Document ID

3. เปลี่ยนสถานะของ document จาก completed เป็น waiting เพื่อให้ document สามารถนำมาใช้ประมวลผลได้ใหม่ ตามภาพที่ 3.17



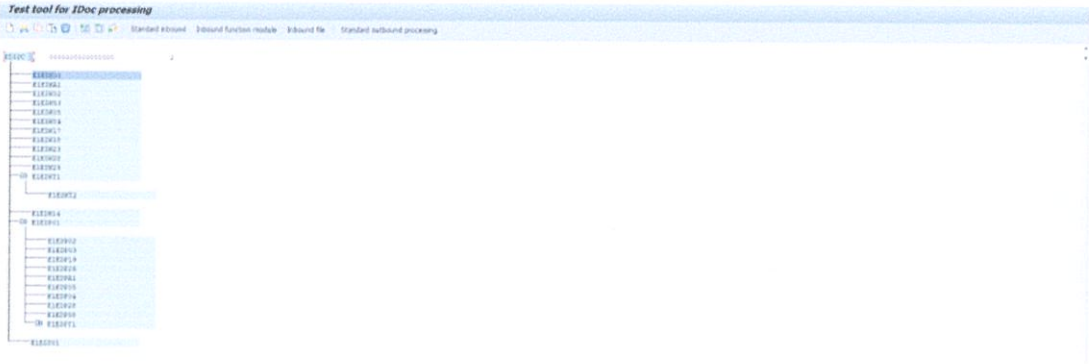
ภาพที่ 3.17 หน้า OCR Document Header

3. พิมพ์ t-code : we19 เลือก basic type ในช่องใส่ invoic01 เพื่อทำการโพสต์ใบแจ้งหนี้แบบ OCR ตามภาพที่ 3.18

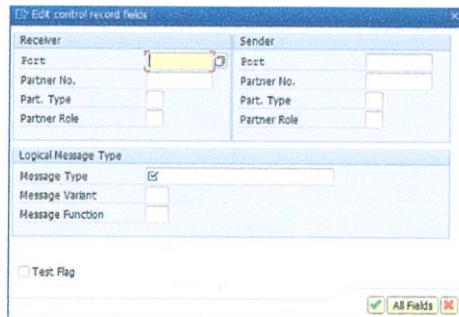


ภาพที่ 3.18 หน้าแรก IDoc processing

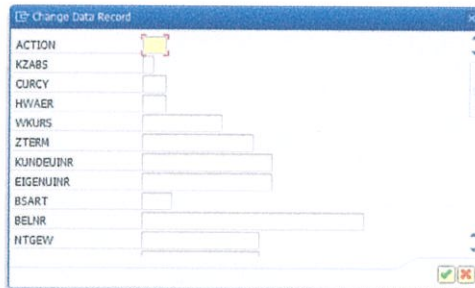
4. เมื่อเข้ามาแล้วจะเจอหน้าใส่ข้อมูลตามภาพที่ 3.19 จากนั้นใส่ข้อมูลตามไฟล์ข้อความที่ได้คัดลอกไว้ได้แก่ข้อมูล EDIDC header ตามภาพที่ 3.20 และข้อมูลของ segment name แต่ละตัวตามภาพที่ 3.21



ภาพที่ 3.19 หน้าใส่ข้อมูล IDoc processing

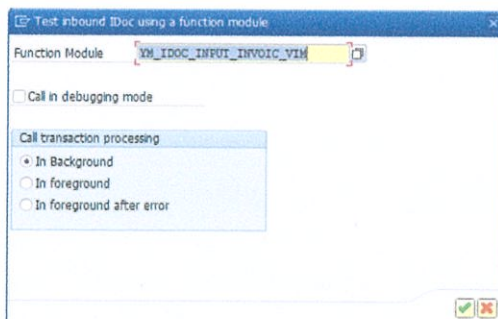


ภาพที่ 3.20 หน้าใส่ข้อมูล header EDICD



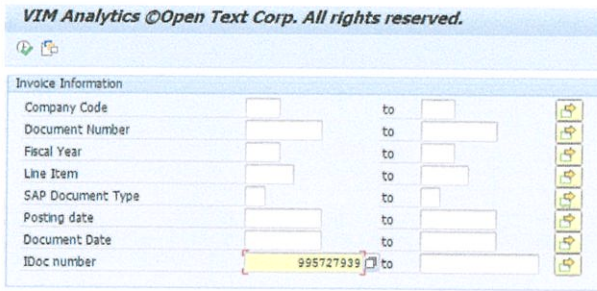
ภาพที่ 3.21 หน้าใส่ข้อมูล Segment name (E1EDK01)

5. รัน inbound function module ชื่อ ym_idoc_input_invoic_vim โดยเลือกการทำงานแบบ run in background ตามภาพที่ 3.22 เพื่อเป็นการบันทึกใบแจ้งหนี้แบบ OCR ลงในระบบ SAP



ภาพที่ 3.22 หน้า inbound function module

6. พิมพ์ t-code : /opt/vim_analytics เพื่อเปิดหน้า vim analytics ใส่เลข idoc ที่ได้หลังจากรัน inbound function module ในช่อง IDoc number ตามภาพที่ 3.23



ภาพที่ 3.23 หน้า VIM Analytics

7. ดูสถานะของ idoc ที่โพลไปหน้า VIM Analytics ตามภาพที่ 3.24

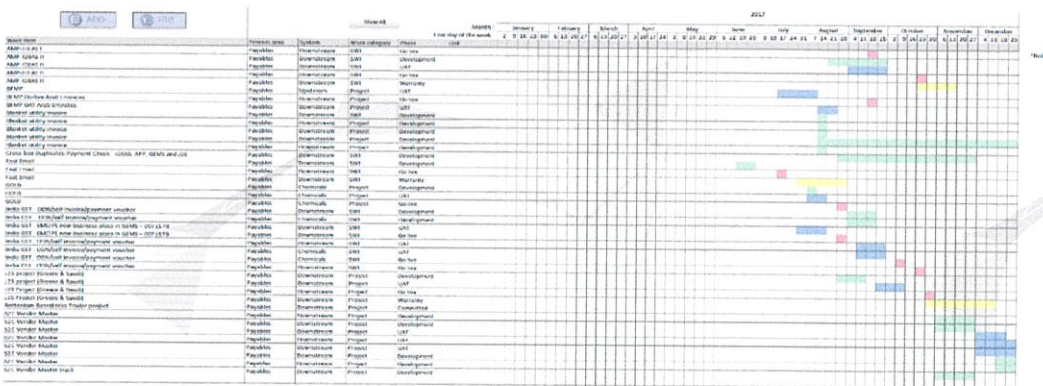


ภาพที่ 3.24 หน้า VIM Analytics ตาม Document view

3.2.3 การเก็บข้อมูลความต้องการและการศึกษาวิธีใช้งาน

- การดึงข้อมูลจากปฏิทินของทีม

เริ่มต้นเมื่อเปิดปฏิทินของทีมขึ้นมาจะเริ่มที่หน้าแรก ซึ่งจะมีส่วนติดต่อผู้ใช้เป็นลักษณะเหมือนเป็นแผนผังแสดงลำดับเหตุการณ์ที่เกิดขึ้นเรียงตามชื่อของโปรเจค ตามภาพที่ 3.25 และมีฟังก์ชันในการจำแนกงานต่างๆ ออกตามประเภทของข้อมูล ได้แก่ ขอบเขตกระบวนการทำงาน ระบบที่ใช้ ประเภทของงาน และสถานะของงานตามตัวอย่างในภาพที่ 3.26 และมีการใช้สีตามสถานะของโปรเจคในการแบ่งแต่ละโปรเจค



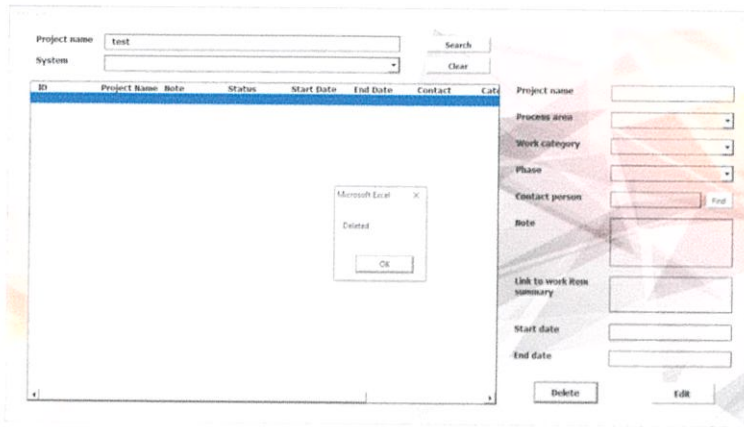
ภาพที่ 3.25 หน้ามุมมอง Timeline

ภาพที่ 3.29 หน้าเพิ่มข้อมูล

หน้าแก้ไขข้อมูลของปฏิทินงานของทีมนั้นเริ่มจากการที่ผู้ใช้ค้นหาข้อมูลที่ต้องการแก้ไขโดยค้นหาจากชื่อโปรเจกต์หรือชื่อระบบ จากนั้นเลือกโปรเจกต์ที่ต้องการแก้ไข และแก้ไขข้อมูล เมื่อแก้ไขข้อมูลเสร็จแล้วจึงกดปุ่ม “Edit” ตามภาพที่ 3.30

ภาพที่ 3.30 หน้าแก้ไขหรือลบข้อมูล

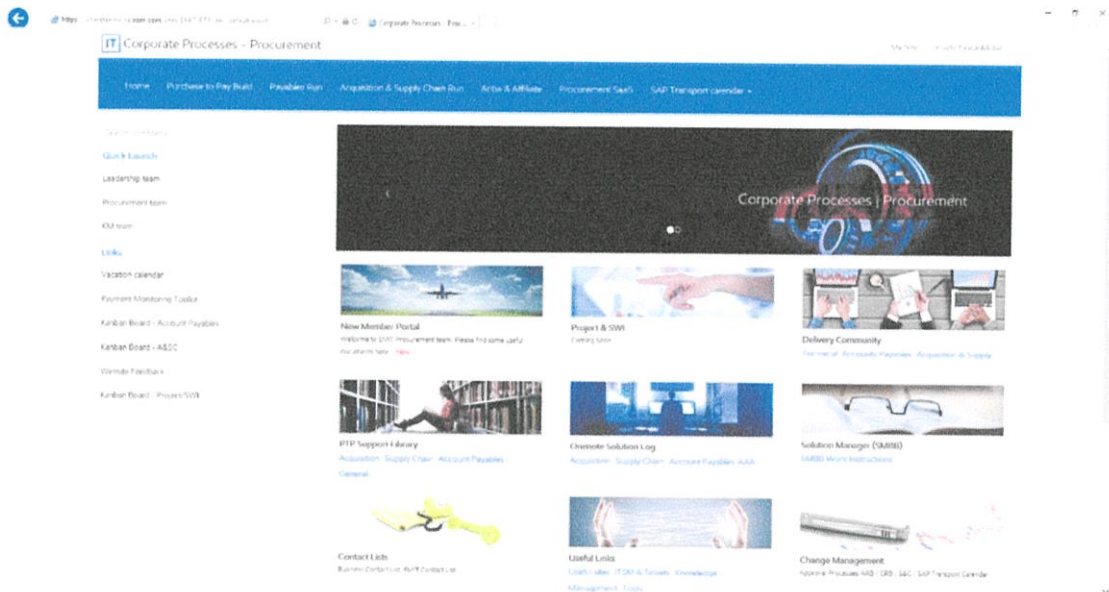
การลบข้อมูลโปรเจกต์นั้นทำได้ในหน้าแก้ไขข้อมูลของปฏิทินงานของทีม เริ่มจากผู้ใช้ค้นหาข้อมูลที่ต้องการลบโดยค้นหาจากชื่อโปรเจกต์หรือชื่อระบบ จากนั้นเลือกโปรเจกต์ที่ต้องการลบ และกดปุ่ม “Delete” โปรแกรมจะถามเพื่อยืนยันว่าจะลบข้อมูลหรือไม่ ให้กดปุ่ม “Yes” เพื่อยืนยันการลบข้อมูล หรือกดปุ่ม “No” เพื่อยกเลิกการทำงาน ตามตัวอย่างในภาพที่ 3.31



ภาพที่ 3.31 หน้าลบบข้อมูลสมบูรณ์

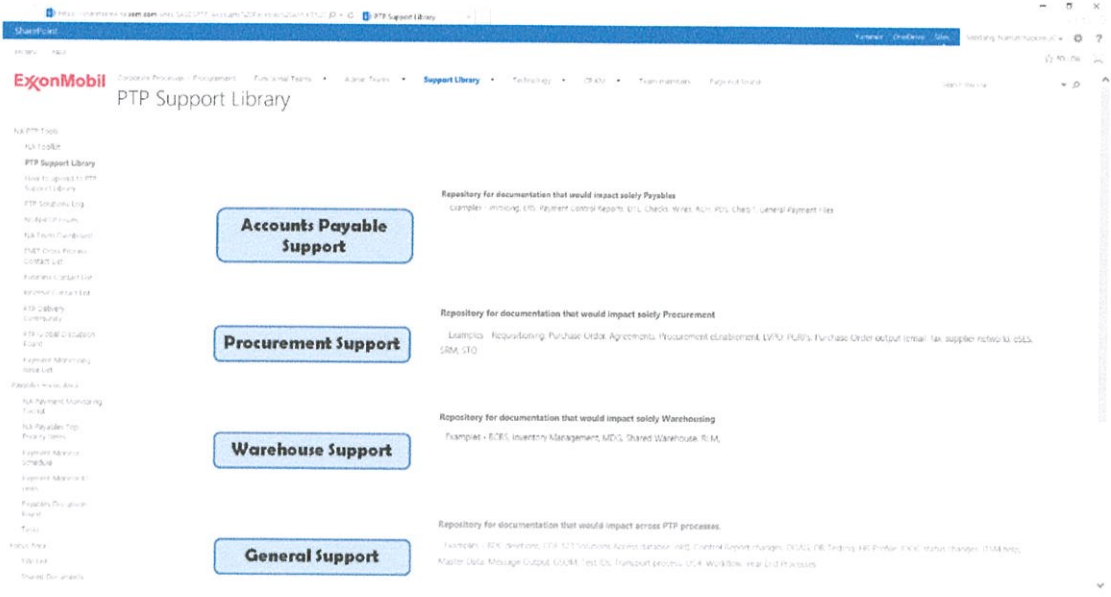
- วิธีการค้นหาเอกสารจาก PTP Support Library

1. เปิดหน้าเว็บไซต์ของทีม แล้วเลือกที่ PTP Support Library ตามภาพที่ 3.32



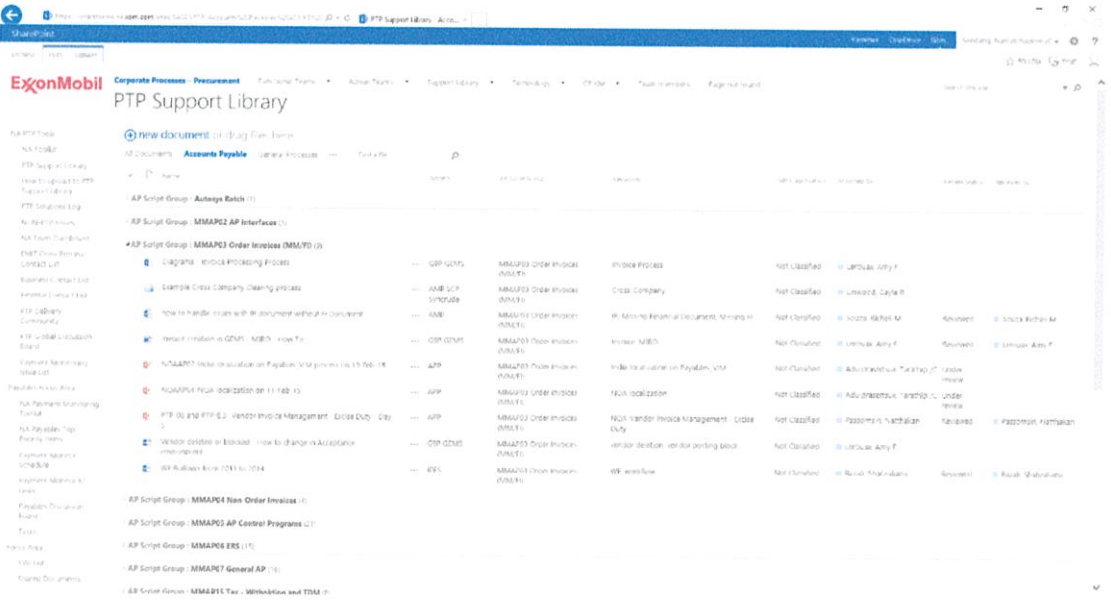
ภาพที่ 3.32 หน้า Team site

2. เลือกประเภทของเอกสารที่ต้องการค้นหาในหน้า PTP Support Library ตามภาพที่ 3.33



ภาพที่ 3.33 หน้าแรก PTP Support Library

3. เลือกเอกสารที่ต้องการจากหน้า PTP Support Library ตามภาพที่ 3.34



ภาพที่ 3.34 หน้าแสดงเอกสารใน PTP Support Library

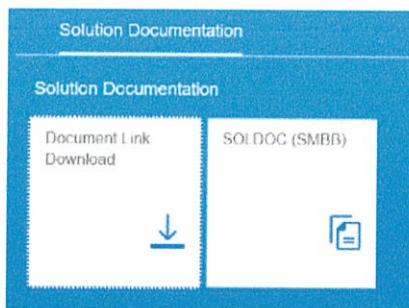
- วิธีการค้นหาเอกสารจาก SMBB

1. เปิดโปรแกรม SAP เลือก system GSP แล้วพิมพ์ t-code : sm_workcenter ตามภาพที่ 3.35 เพื่อเปิดหน้า SMBB



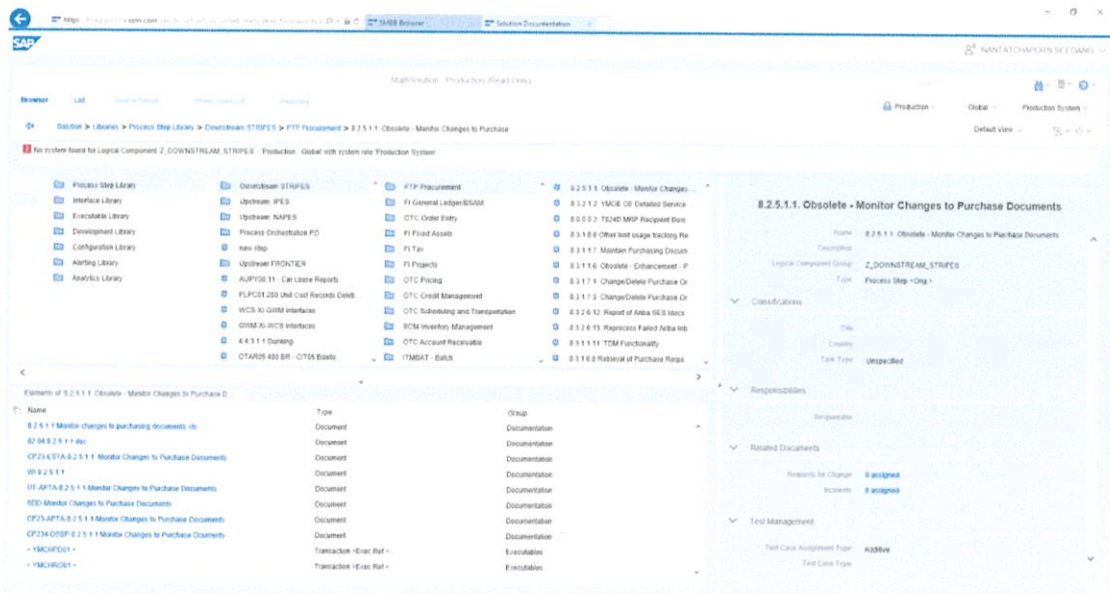
ภาพที่ 3.35 หน้าแรกในระบบ GSP

2. คลิก SOLDOC (SMBB) ตามภาพที่ 3.36 เพื่อเริ่มค้นหาเอกสารใน SMBB



ภาพที่ 3.36 หน้าแรกของ SMBB

3. ค้นหาเอกสารตามหมวดของเอกสารที่ต้องการหาใน SMBB ตามที่แสดงในภาพที่ 3.37

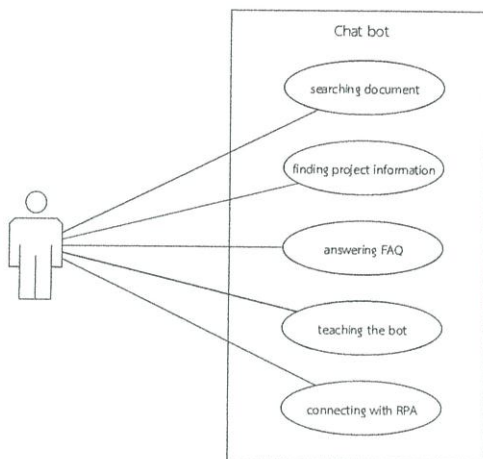


ภาพที่ 3.37 หน้าแสดงเอกสารใน SMBB

3.3 การออกแบบการทำงาน

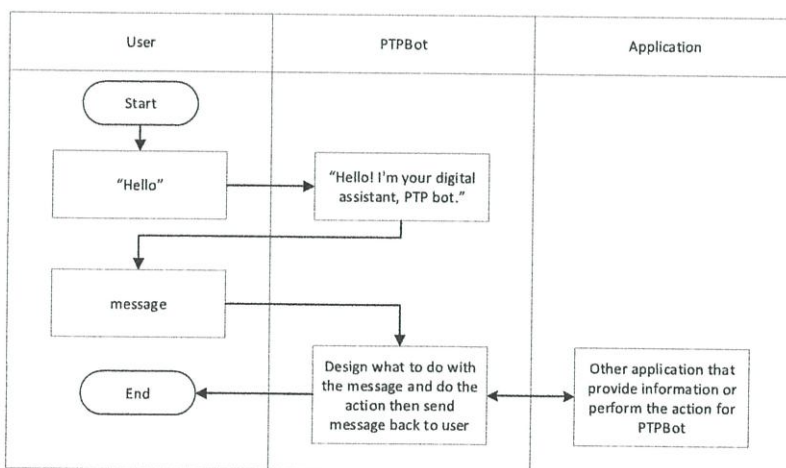
3.3.1 การออกแบบการทำงานของระบบแชทบอท

การออกแบบการทำงานของระบบแชทบอทนั้น ฟังก์ชันหลัก ๆ ที่ต้องสามารถทำได้นั้น ประกอบด้วย การค้นหาเอกสาร การค้นหาข้อมูลจากปฏิทินงานของทีม การตอบคำถามที่ได้รับซ้ำ ๆ การทำให้แชทบอทสามารถเรียนรู้จากข้อมูลที่ผู้ใช้ส่งเข้ามาได้ และการเชื่อมต่อกับระบบการทดสอบ โปรแกรมอัตโนมัติโดยใช้เทคโนโลยี RPA ได้ โดยการออกแบบการทำงานแสดงโดยการเขียน use case ในการเขียนฟังก์ชันคร่าว ๆ ที่ต้องทำให้ได้ก่อน ตามภาพที่ 3.38



ภาพที่ 3.38 Use case ของระบบแชทบอท

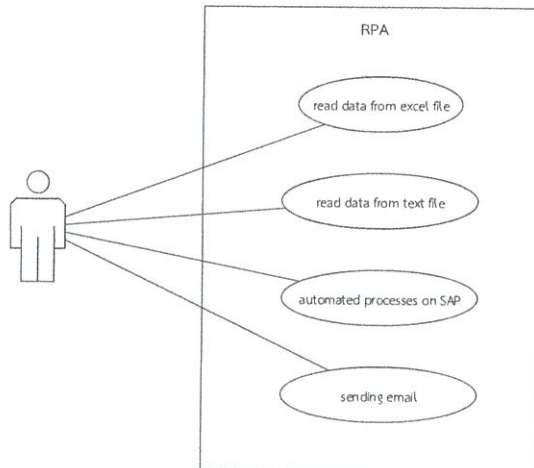
การสนทนาระหว่างผู้ใช้งานกับระบบแชทบอทผ่านโปรแกรม Skype นั้นในทุกคำสั่งจะมีความคล้ายกันคือจะเริ่มต้นจากผู้ใช้งานส่งข้อความเริ่มต้นเข้ามา จากนั้นผู้ใช้จะส่งคำสั่งเข้ามาเพื่อให้บอททำงาน โดยบอทจะประมวลผลคำสั่งที่ได้รับจากผู้ใช้งานและทำงานตามที่ใช้ต้องการ เมื่อบอททำงานเสร็จแล้วจะตอบข้อความกลับไปหาผู้ใช้งานผ่านโปรแกรม Skype ตามภาพที่ 3.39



ภาพที่ 3.39 ภาพรวมการทำงานของระบบแชทบอท

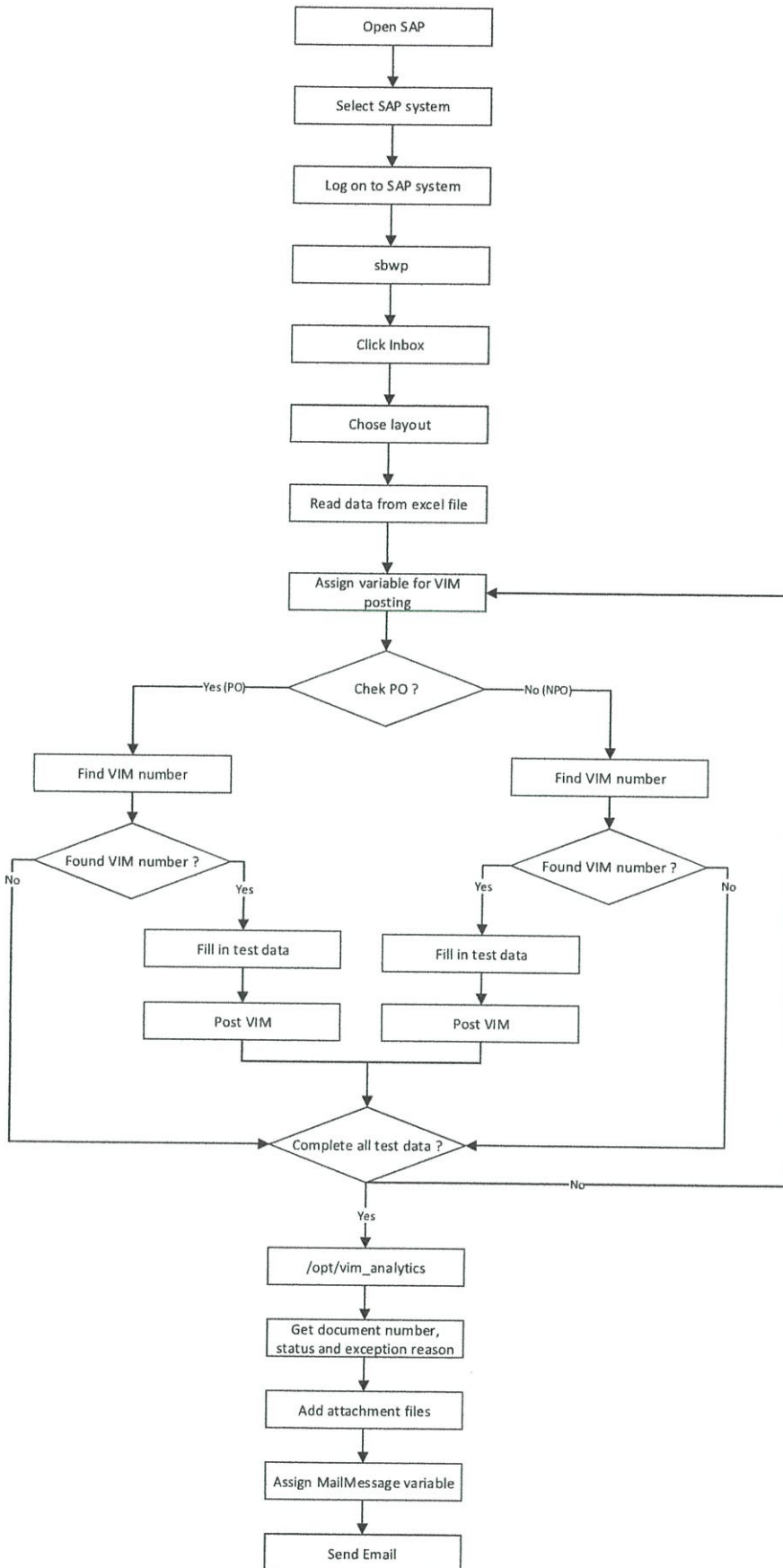
3.3.2 การออกแบบระบบการทดสอบโปรแกรมอัตโนมัติโดยใช้เทคโนโลยี RPA

การออกแบบการทำงานของระบบการทดสอบโปรแกรมอัตโนมัติโดยใช้เทคโนโลยี RPA นั้น ฟังก์ชันหลัก ๆ ที่ต้องสามารถทำได้นั้น ประกอบด้วย การอ่านข้อมูลจากไฟล์เอ็กเซลและไฟล์ข้อความ การทำให้โปรแกรม UiPath สามารถทำงานบนโปรแกรม SAP ได้ถูกต้อง เช่น การคลิกปุ่มที่ถูกต้อง เป็นต้น การออกแบบการทำงานใช้ use case ในการเขียนฟังก์ชันคร่าว ๆ ที่ต้องทำได้ก่อน ตามภาพที่ 3.40



ภาพที่ 3.40 Use case ของกระบวนการทำงานอัตโนมัติโดยหุ่นยนต์

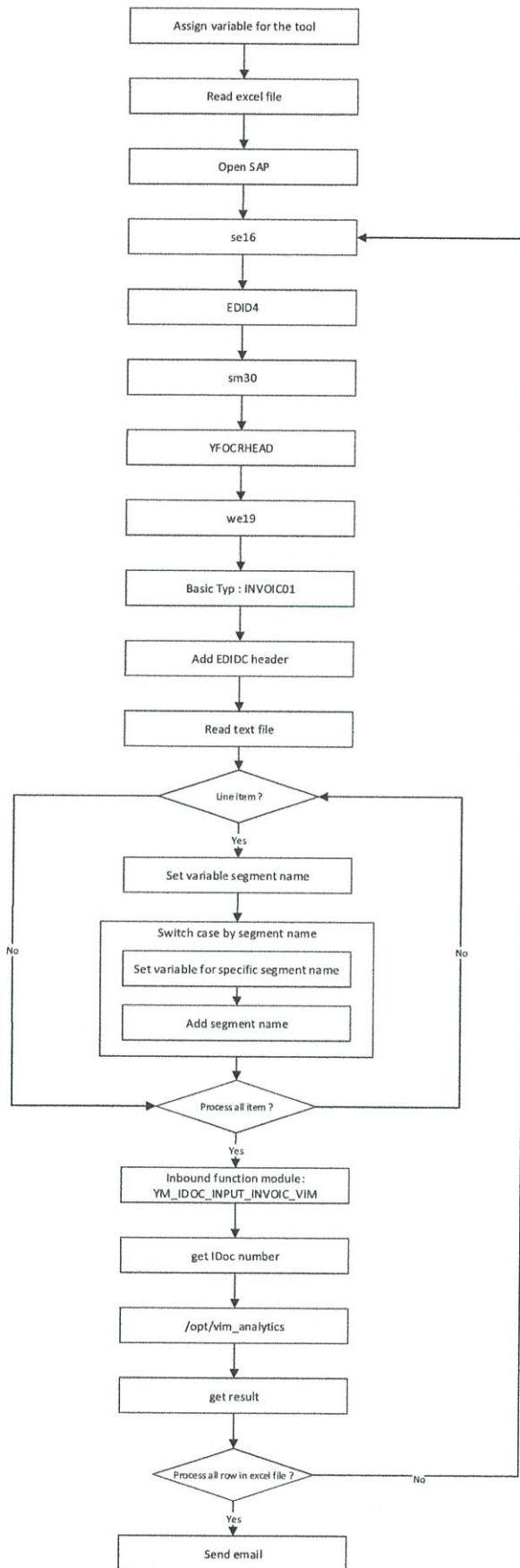
การทำงานของ Vendor invoice management (VIM) automated testing tool นั้น แสดงได้ตามภาพที่ 3.41 โดยจะเริ่มจากการเปิดโปรแกรม SAP และทำการ logon เพื่อเข้าสู่ระบบ SAP จากนั้นโปรแกรมจะเปิดหน้า business workplace เพื่อทำการโพส VIM จากนั้นอ่านข้อมูลที่ต้องการโพสจากไฟล์ข้อมูลที่เป็นไฟล์เอ็กเซลและโพสตามชนิดของใบแจ้งหนี้ เมื่อโพสครบทุกตัวแล้ว จะเปิดหน้า VIM analytics tool เพื่อดึงข้อมูลของ VIM number นั้น ๆ และทำการส่งอีเมลล์หาผู้ใช้



ภาพที่ 3.41 แผนผังแสดงขั้นตอนการทำงานของโปรแกรม VIM automated testing tool

การทำงานของ Optical character recognition (OCR) automated testing tool นั้น จะเริ่มจากให้ผู้ใช้ใส่ค่าชื่อไฟล์อิเล็กทรอนิกส์ที่มีเลข IDoc number อยู่ และตั้งค่าตัวแปรในโปรแกรมใน sequence ชื่อ Assign ก่อน จากนั้นผู้ใช้จะรันโปรแกรม

เมื่อผู้ใช้รันโปรแกรมแล้ว โปรแกรมจะเริ่มจากการเซตตัวแปรเริ่มต้น แล้วจะเปิดไฟล์อิเล็กทรอนิกส์เพื่ออ่านค่า IDoc number จากไฟล์อิเล็กทรอนิกส์ หลังจากนั้นโปรแกรมจะเปิดโปรแกรม SAP เลือกระบบตามค่าตัวแปร “system” ที่กำหนดไว้ใน sequence Assign แล้วดูตามค่า IDoc number ในไฟล์อิเล็กทรอนิกส์ แต่ละค่าจะถูกนำไปค้นหาใน table ชื่อ EDID4 เพื่อสร้างเป็นไฟล์ข้อความออกมาจากโปรแกรม SAP จากนั้นจะนำไฟล์ข้อความที่ได้จากโปรแกรม SAP จะถูกนำไปค้นหา Document ID จากนั้นเปิด t-code sm30 และใส่ Document ID ขั้นตอนนี้ทำเพื่อเปลี่ยนสถานะของ OCR จาก completed เป็น waiting เพื่อให้ Document ID นั้นสามารถนำไปใช้ประมวลผลอีกครั้งได้ เปิด t-code we19 เพื่อเปิด IDoc testing tool ในโปรแกรม SAP เพื่อทำการโพส OCR จากนั้นทำการลบ segment name ที่มีอยู่ก่อนหน้าเพื่อเพิ่ม segment name ใหม่ แล้วอ่านข้อมูลจากไฟล์ข้อความเพื่อให้ได้ segment name มาโพส จากนั้นดูตามแต่ละบรรทัดของข้อมูลในไฟล์ข้อความและเช็คว่ามี line item ที่ต้องนำมาประมวลผลหรือไม่ ถ้าใช่ให้นำข้อมูลบรรทัดนั้นไปเข้า switch case เพื่อทำการโพสตาม segment name หลังจากทำครบทุก line item แล้ว ให้เรียกใช้ inbound function module ชื่อ YM_IDOC_INPUT_INVOIC_VIM และตั้งค่า IDoc number ใหม่ จากหน้าต่างที่โปรแกรมแสดงขึ้น จากนั้นเปิด VIM Analytics เพื่อดึงข้อมูลผลลัพธ์ของ IDoc number ใหม่ ได้แก่ IDOC number, Document ID, Document number, Document status และ Exception reason ขั้นตอนสุดท้ายคือการส่งอีเมลให้กับผู้ใช้โดยเริ่มจากการเซตตัวแปรที่ใช้ส่งอีเมล และส่งอีเมลให้ผู้ใช้ โดยขั้นตอนการทำงานทั้งหมดนั้นแสดงอยู่ในแผนผังแสดงขั้นตอนการทำงานของโปรแกรมตามภาพที่ 3.42



ภาพที่ 3.42 แผนผังแสดงขั้นตอนการทำงานของโปรแกรม OCR automated testing tool

บทที่ 4

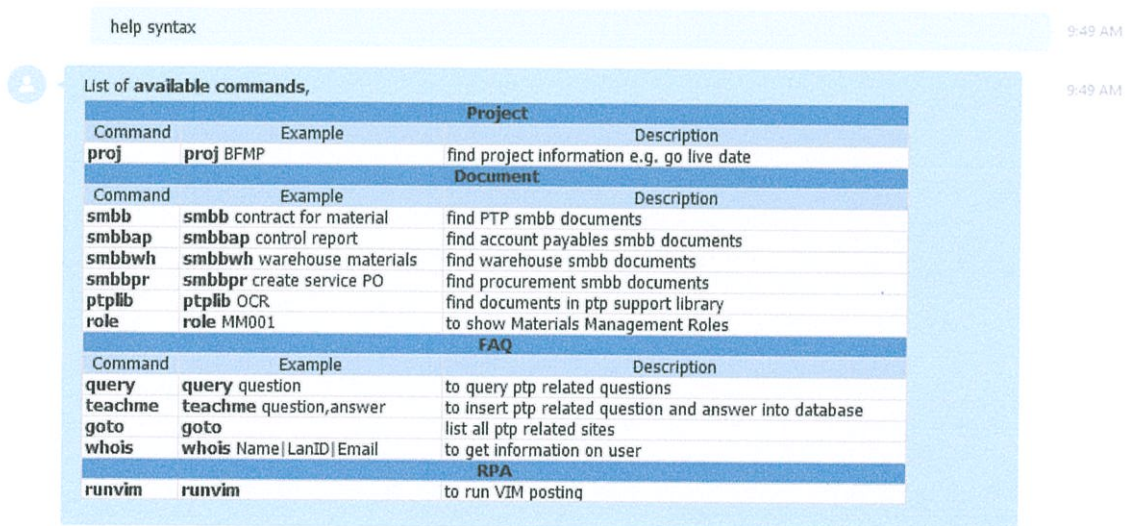
ผลการวิจัย

4.1 ระบบแชทบอท

ระบบแชทบอทนี้มีชื่อว่า “PTPBot” เป็นระบบแชทบอทสำหรับทีม cooperate process procurement มีฟังก์ชันหลักๆ ทั้งหมดแบ่งออกเป็น 4 ประเภท ได้แก่ การค้นหาข้อมูลจากปฏิทินงานของทีม การค้นหาเอกสาร การตอบคำถามซ้ำ ๆ (Frequency asking question) และการส่งงานการทดสอบโปรแกรมอัตโนมัติโดยใช้เทคโนโลยี RPA โดยการส่งอีเมลให้กับผู้ที่เกี่ยวข้อง

การทำงานของแชทบอทเริ่มจากผู้ใช้เปิดโปรแกรม Skype และเริ่มสนทนากับแชทบอท โดยผู้ใช้จะส่งข้อความผ่านโปรแกรม Skype แล้วระบบแชทบอทจะประมวลผลตามกฎที่ได้เขียนไว้ จากนั้นบอทจะส่งข้อความตอบกลับผู้ใช้ผ่าน Skype เหมือนการสนทนาโดยใช้โปรแกรม Skype ปกติ

เมื่อผู้ใช้ส่งข้อความ help syntax มาที่ระบบแชทบอท บอทจะประมวลผลข้อความ help syntax เป็น intent ที่ชื่อว่า help จากนั้นจะทำงานตามที่กฎกำหนดไว้ และส่งผลลัพธ์กลับไปแสดงผลที่โปรแกรม Skype ตามภาพที่ 4.1

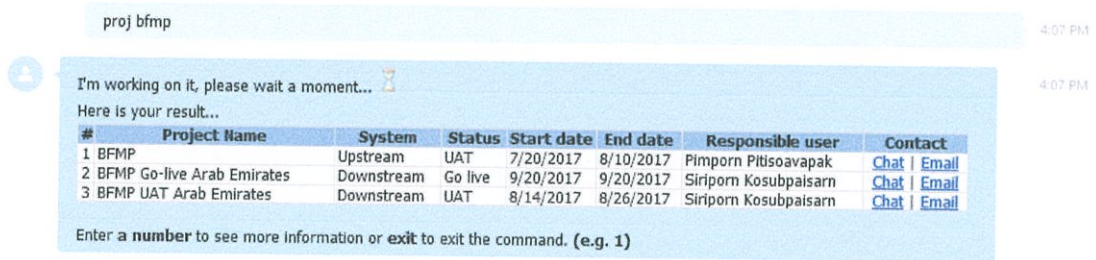


ภาพที่ 4.1 คำสั่งแบบ rule-based

4.1.1 การค้นหาข้อมูลจากปฏิทินงานของทีม

- การทำงานของคำสั่ง “proj” แบบ rule-based

การค้นหาข้อมูลโปรเจกต์ผ่านระบบแชทบอทจะเริ่มจากผู้ใช้ส่งข้อความที่มีคำว่า proj ตามด้วยชื่อโปรเจกต์ที่ต้องการค้นหาข้อมูลเข้ามาใช้โปรแกรม Skype จากนั้นบอทจะประมวลผลเป็น intent และประมวลผลคำสั่ง จากนั้นบอทจะไปค้นข้อมูลจากไฟล์เอ็กเซลและส่งผลลัพธ์กลับไปในรูปแบบของ html string และแสดงผลในโปรแกรม skype โดยผู้ใช้สามารถเห็นข้อมูลเดียวกับข้อมูลบนปฏิทินงานของทีม และสามารถกด “Chat” หรือ “Email” เพื่อทำการสนทนากับผู้รับผิดชอบโปรเจกต์นั้นได้ทันที ตัวอย่างดังภาพที่ 4.2

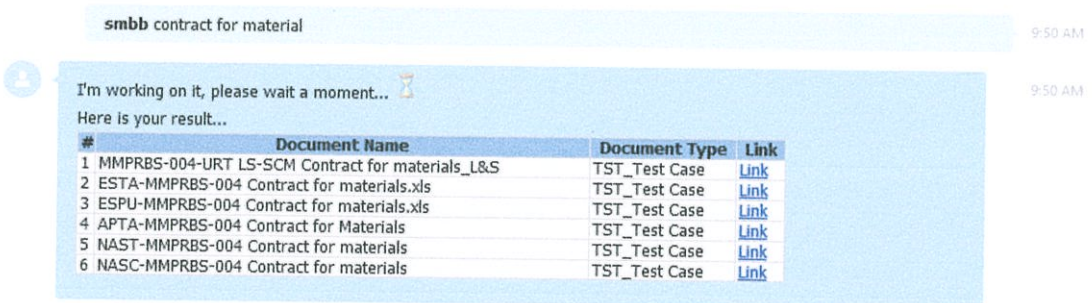


ภาพที่ 4.2 การใช้คำสั่งแบบ rule-based (proj)

4.1.2 การค้นหาเอกสาร

- การทำงานของคำสั่ง “smbb” แบบ rule-based

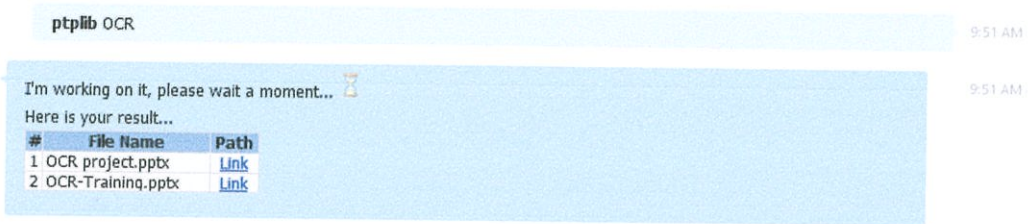
การค้นหาเอกสารแบบ SMBB เริ่มจากผู้ใช้ส่งข้อความ smbب ตามด้วยชื่อเอกสารที่ต้องการค้นหาข้อมูล จากนั้นบอทจะประมวลผลเป็น intent และประมวลผลคำสั่ง จากนั้นบอทจะไปค้นข้อมูลจากไฟล์ xml ที่เก็บชื่อเอกสารและ link ไปยังเอกสารนั้น และส่งผลลัพธ์ที่ประกอบด้วยชื่อเอกสาร ประเภทของเอกสาร และ URL ของเอกสารนั้น ๆ กลับไปในรูปแบบของ html string และแสดงผลในโปรแกรม skype โดยมีตัวอย่างตามภาพที่ 4.3



ภาพที่ 4.3 การใช้คำสั่งแบบ rule-based (smbb)

- การทำงานของคำสั่ง “ptplib” แบบ rule-based

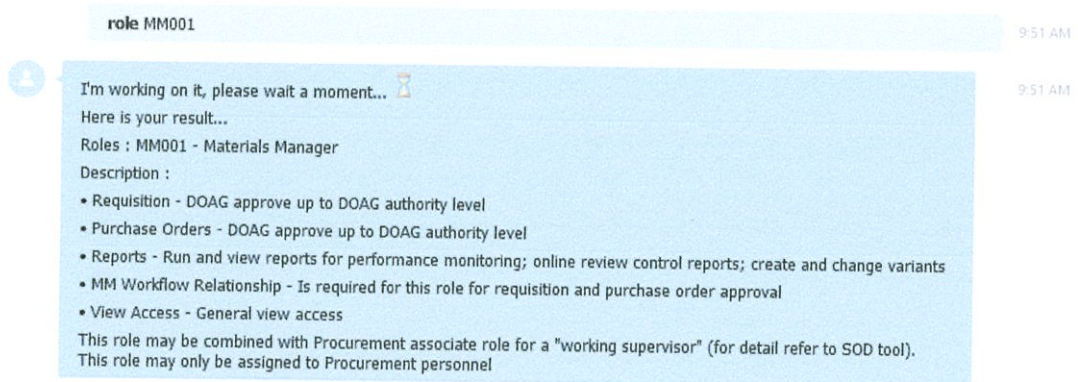
การค้นหาเอกสารใน PTP Support library นั้นเริ่มจากผู้ใช้ส่งข้อความที่มีคำว่า ptplib ตามด้วยชื่อเอกสารที่ต้องการค้นหาข้อมูล จากนั้นบอทจะประมวลผลเป็น intent ptplib และประมวลผลคำสั่ง จากนั้นบอทจะไปค้นหาข้อมูลจากโฟลเดอร์ PTP Support library และส่งผลลัพธ์กลับไปในรูปแบบของ html string และแสดงผลในโปรแกรม Skype ตามภาพที่ 4.4



ภาพที่ 4.4 การใช้คำสั่งแบบ rule-based (ptplib)

- การทำงานของคำสั่ง “role” แบบ rule-based

การค้นหาข้อมูล material management role นั้น เริ่มจากผู้ใช้ส่งคำสั่ง role ตามด้วยชื่อ role ที่ต้องการค้นหาข้อมูล จากนั้นบอทจะประมวลผลเป็น intent และประมวลผลคำสั่ง จากนั้นบอทจะไปค้นข้อมูลจาก share point list ที่เก็บข้อมูลชื่อของ role และคำอธิบายของ role นั้นๆ และส่งผลลัพธ์กลับไปในรูปแบบของ html string และแสดงผลในโปรแกรม Skype ตามภาพที่ 4.5



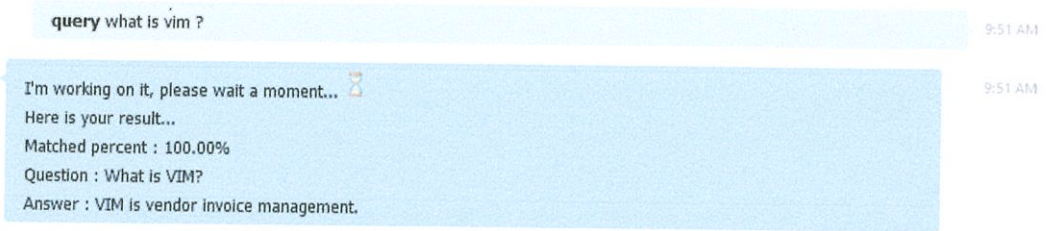
ภาพที่ 4.5 การใช้คำสั่งแบบ rule-based (role)

4.1.3 การตอบคำถามซ้ำ ๆ (Frequency asking question)

- การทำงานของคำสั่ง “query” แบบ rule-based

การค้นหาคำถามที่ได้รับการถามบ่อย ๆ นั้นจะเริ่มจากผู้ใช้ส่งคำสั่ง query ตามด้วยคำถามที่ต้องการทราบ จากนั้นบอทจะประมวลผลเป็น intent และประมวลผลคำสั่ง จากนั้นบอทจะไปค้นหาคำถามจาก share point list และส่งผลลัพธ์กลับไปในรูปแบบของ string ประกอบด้วย

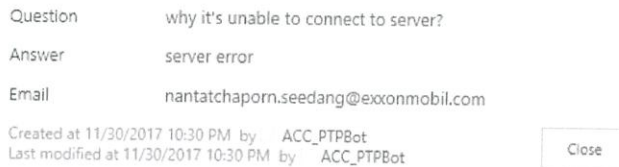
คำถาม คำตอบ และเปอร์เซ็นต์ของคำถามว่าคำถามนี้ตรงกับคำถามที่ผู้ใช้ต้องการมากน้อยเพียงใด และแสดงผลในโปรแกรม skype ตามภาพที่ 4.6



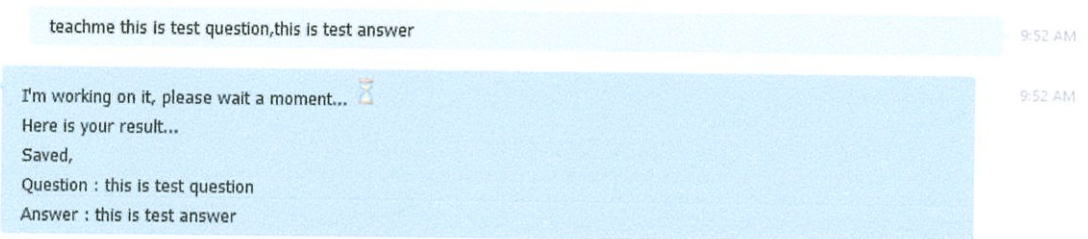
ภาพที่ 4.6 การใช้คำสั่งแบบ rule-based (query)

- การทำงานของคำสั่ง “teachme” แบบ rule-based

การเพิ่มคำถามให้กับฐานข้อมูลของระบบแชทบอทนั้น เริ่มต้นด้วยการที่ผู้ใช้ส่งคำสั่ง teachme ตามด้วยคำถามที่ต้องการเพิ่มเข้าไปในฐานข้อมูล จากนั้นบอทจะประมวลผลเป็น intent และประมวลผลคำสั่ง จากนั้นบอทจะไปเพิ่มคำถามใน share point list ตามภาพที่ 4.7 และส่งผลลัพธ์เป็นการยืนยันข้อมูลของคำถามที่ผู้ใช้ส่งเข้ามา กลับไปในรูปแบบของ string และแสดงผลในโปรแกรม skype ตามภาพที่ 4.8



ภาพที่ 4.7 การใช้คำสั่ง teachme บันทึกข้อมูลคำถามและคำตอบลงใน sharepoint list

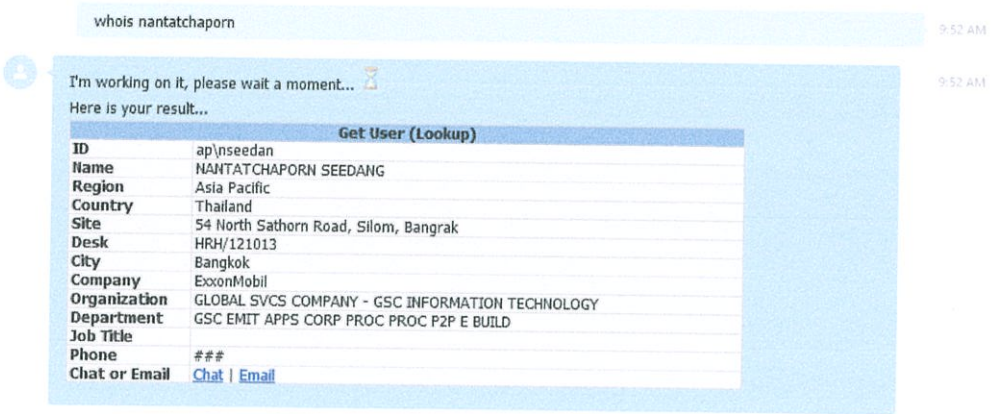


ภาพที่ 4.8 การใช้คำสั่งแบบ rule-based (teachme)

- การทำงานของคำสั่ง “whois” แบบ rule-based

การค้นหาข้อมูลของบุคคลจากชื่อ อีเมลล์ หรือ LAN ID นั้นเป็นอีกหนึ่งคำถามที่ เกิดขึ้นบ่อย ดังนั้นผู้จัดทำจึงเพิ่มฟังก์ชันนี้เข้ามาในระบบแชทบอทด้วย การค้นหาข้อมูลบุคคลจะเริ่ม จากผู้ใช้ส่งคำสั่ง whois ตามด้วยชื่อของคนที่ผู้ใช้ต้องการค้นหาข้อมูล จากนั้นบอทจะประมวลผลเป็น

intent และประมวลผลคำสั่ง จากนั้นบอทจะป้อนข้อมูลของคนที่ผู้ใช้ต้องการค้นหาจาก API และส่งผลลัพธ์กลับไปในรูปแบบของ html string และแสดงผลในโปรแกรม skype โดยผู้ใช้สามารถกด “Chat” หรือ “Email” เพื่อเป็นการสนทนากับคนที่ผู้ใช้ค้นหาข้อมูลได้ทันที โดยมีตัวอย่างตามภาพที่ 4.9

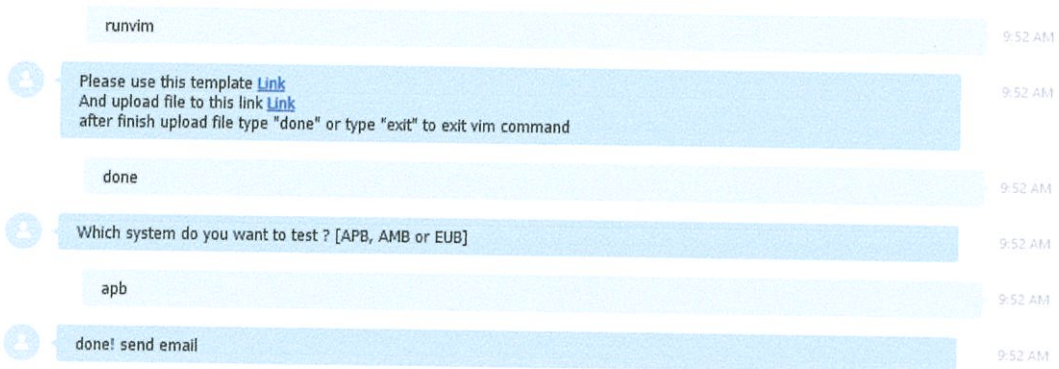


ภาพที่ 4.9 การใช้คำสั่งแบบ rule-based (whois)

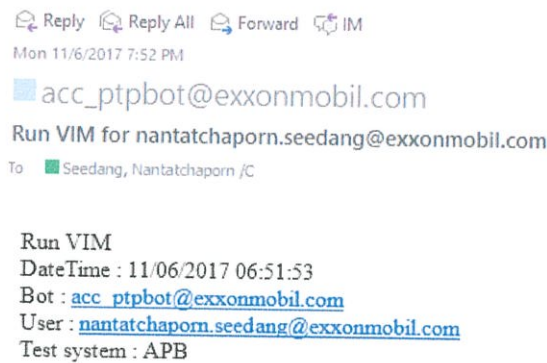
4.1.4 การสั่งงานการทดสอบโปรแกรมอัตโนมัติโดยใช้เทคโนโลยี RPA โดยการส่งอีเมลล์ให้กับผู้ที่เกี่ยวข้อง

- การทำงานของคำสั่ง “runvim” แบบ rule-based

การสั่งงานการทดสอบโปรแกรมอัตโนมัติโดยใช้เทคโนโลยี RPA โดยการส่งอีเมลล์ให้กับผู้ที่เกี่ยวข้องผ่านระบบแชทบอทนั้นจะเริ่มจากผู้ใช้ส่งคำสั่ง runvim เข้ามาในระบบ จากนั้นบอทจะประมวลผลเป็น intent และประมวลผลคำสั่ง จากนั้นบอทจะส่ง URL ของไฟล์ตัวอย่าง และ URL ที่จะอัปโหลดไฟล์ที่ใช้เป็นข้อมูลในการรันโปรแกรมอัตโนมัติโดยใช้ RPA เมื่อผู้ใช้ทำการอัปโหลดไฟล์เสร็จแล้ว ผู้ใช้จะส่งคำว่า done กลับมา จากนั้นบอทจะประมวลผล state และส่งข้อความตอบกลับเพื่อรับค่าชื่อระบบที่ผู้ใช้ต้องการทำทดสอบ ตามภาพที่ 4.10 เมื่อผู้ใช้ส่งค่าชื่อระบบกลับมาแล้ว บอทจะส่งอีเมลล์ไปให้ผู้รับผิดชอบในการทำการทดสอบ ตามภาพที่ 4.11 เพื่อให้ผู้ที่รับผิดชอบทำการรันโปรแกรมที่ใช้ทดสอบ



ภาพที่ 4.10 การใช้คำสั่งแบบ rule-based (runvim)

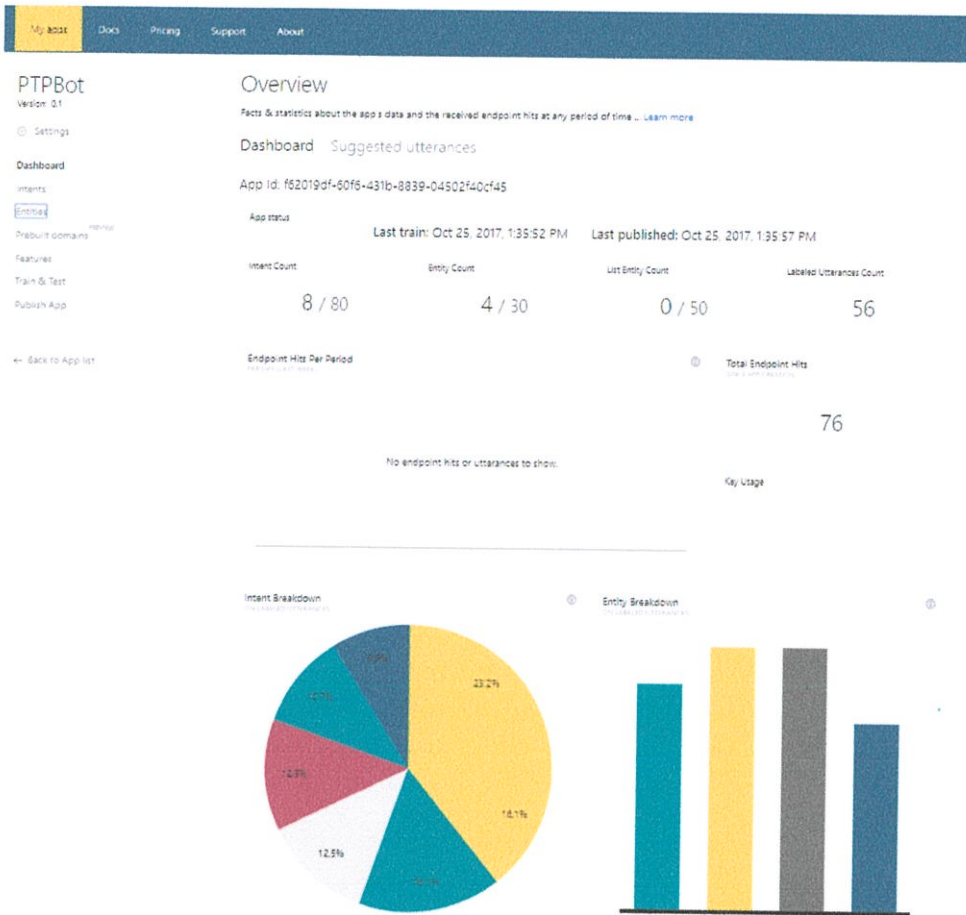


ภาพที่ 4.11 ตัวอย่างอีเมลล์จากคำสั่ง runvim

4.1.5 Language Understanding Intelligent Service (LUIS)

ระบบแชทบอทนี้ ไม่เพียงมีแต่การใช้คำสั่งแบบกฎ (rule-based) เท่านั้น ผู้จัดทำได้ทดลองนำเทคโนโลยี machine learning มาใช้งานร่วมด้วย โดยการนำเทคโนโลยี LUIS มาใช้นั้น เป็นเพียงแค่การทำ proof of concept (POC) เท่านั้น ส่วนการนำไปใช้จริงบน production environment นั้นเป็นแผนงานในอนาคต

เครื่องมือที่ผู้จัดทำเลือกสำหรับการทำคือ Language Understanding Intelligent Service (LUIS) ซึ่งเป็นการใช้เทคโนโลยี machine learning ในการประมวลผลข้อความที่ได้รับจากผู้ใช้ ในการหาว่าจากข้อความที่ได้รับนั้นผู้ต้องการให้บอททำอะไร โดยที่ผู้ใช้ไม่ต้องพิมพ์คำสั่งต่างๆ เพื่อเพิ่มความสะดวกสบายในการใช้งานให้กับตัวผู้ใช้อเอง และได้รู้สึกเหมือนกับกำลังสนทนาอยู่กับคนมากขึ้น หน้าแรกของการเริ่มต้นใช้งานจะเป็นแบบภาพที่ 4.12 โดยในหน้าแรกนี้จะแสดงข้อมูลเกี่ยวกับการใช้งานแอปพลิเคชัน LUIS ของระบบแชทบอทที่ผู้จัดทำกำลังพัฒนาอยู่



ภาพที่ 4.12 หน้าแรกของ LUIS.ai

การใช้งาน LUIS นั้นต้องเริ่มจากการเทรนข้อมูลเข้าไปในระบบ จากภาพที่ 4.13 นั้นเป็นการเทรนข้อมูลสำหรับคำสั่ง help ภาพที่ 4.14 นั้นเป็นการเทรนข้อมูลสำหรับคำสั่ง projectinfo ภาพที่ 4.15 นั้นเป็นการเทรนข้อมูลสำหรับคำสั่ง document เพื่อให้ machine learning สามารถเรียนรู้จากข้อมูลของเราได้ เมื่อเทรนข้อมูลเรียบร้อยแล้วสามารถทำการ Publish ข้อมูลมาใช้กับระบบของเราได้ โดยระบบจะใช้ข้อมูลนี้ผ่านการเรียกใช้แบบ API ที่จะให้ผลลัพธ์แบบ JSON Object

help

Here you are in full control of this intent, you can manage its utterances, used entities and suggested utterances. [Learn more](#)

Utterances (13) Entities in use Suggested utterances

Type a new utterance to your intent	
<input type="checkbox"/> what is your role?	1
<input type="checkbox"/> what will you do for me?	1
<input type="checkbox"/> tell me your feature	1
<input type="checkbox"/> tell me about your functionality	1
<input type="checkbox"/> can you tell me what else you can do	1
<input type="checkbox"/> what functionality you have	1
<input type="checkbox"/> things you can do	1
<input type="checkbox"/> do you understand me	1
<input type="checkbox"/> what else you can do	1
<input type="checkbox"/> what can you do for me?	1

ภาพที่ 4.13 ตัวอย่าง training data ของ intent help

projectinfo

Here you are in full control of this intent, you can manage its utterances, used entities and suggested utterances. [Learn more](#)

Utterances (7) Entities in use (1) Suggested utterances

Type a new utterance to your intent	
<input type="checkbox"/> find project [SPRONAME] in team calendar	0.94
<input type="checkbox"/> info about [SPRONAME] in calendar	0.97
<input type="checkbox"/> find [SPRONAME] in project time sheet	0.97
<input type="checkbox"/> search [SPRONAME] in calendar	0.95
<input type="checkbox"/> can you find information about [SPRONAME]	0.94
<input type="checkbox"/> information about [SPRONAME]	0.93
<input type="checkbox"/> i want to know about [SPRONAME] project	0.92

ภาพที่ 4.14 ตัวอย่าง training data ของ intent projectinfo

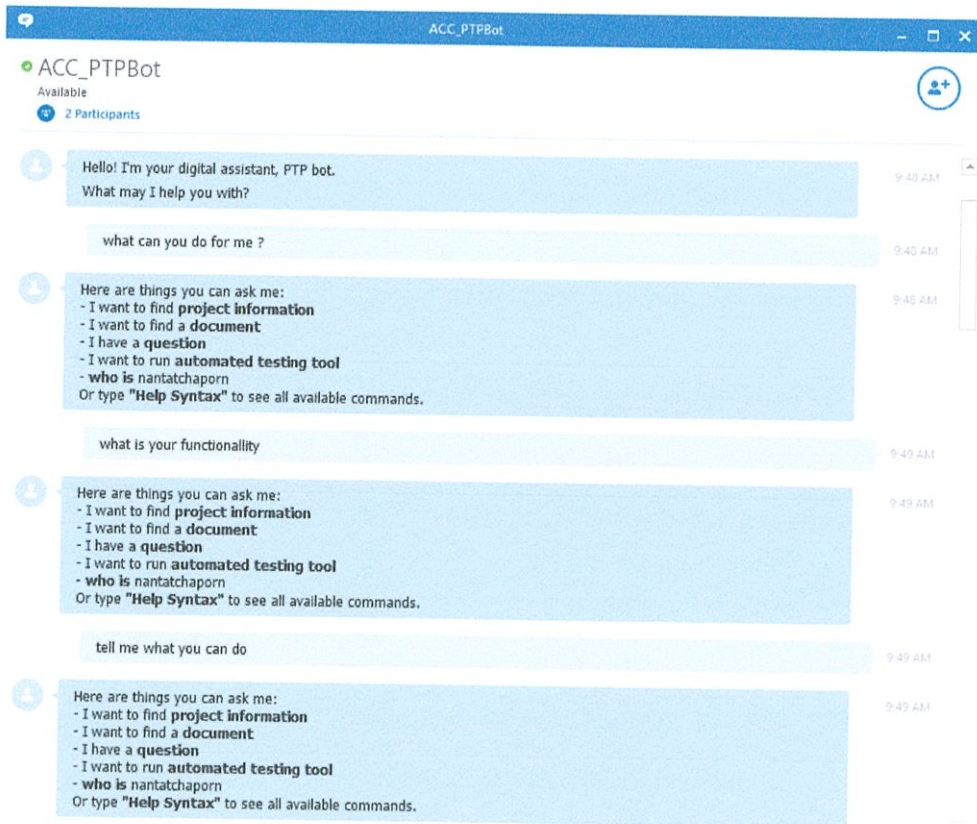
document

Here you are in full control of this intent, you can manage its utterances, used entities and suggested utterances. [Learn more](#)

Utterances (7) Entities in use (1) Suggested utterances

Type a new utterance to your intent	
<input type="checkbox"/> find document about [SDOENAME] in support library	0.97
<input type="checkbox"/> search [SDOENAME] library	0.97
<input type="checkbox"/> find [SDOENAME] in library	0.99
<input type="checkbox"/> can i get link of [SDOENAME] in teamsite	0.99
<input type="checkbox"/> i want to know about [SDOENAME]	0.96
<input type="checkbox"/> where is document about [SDOENAME]	0.98
<input type="checkbox"/> how can i find [SDOENAME]	0.97

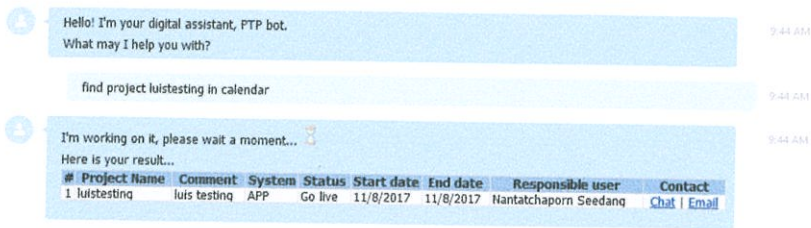
ภาพที่ 4.15 ตัวอย่าง training data ของ intent document



ภาพที่ 4.16 ตัวอย่างการใช้ LUIS ในการประมวลผลให้ได้มาเป็น intent help

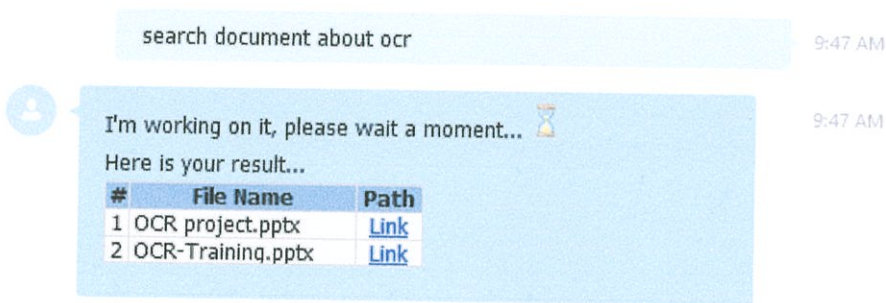
จากภาพที่ 4.16 จะเป็นการแสดงการใช้งานเมื่อผู้ใช้งานต้องการทราบว่าบอทสามารถทำอะไรได้บ้าง ถ้าเป็นการใช้แบบคำสั่ง (rules based) ข้อความของผู้ใช้ต้องมีคำว่า help อยู่ในข้อความเพื่อให้บอทสามารถรับรู้ได้ว่าจากข้อความนี้ผู้ใช้งานต้องการอะไร แต่ถ้าเป็นการใช้ LUIS จะไม่จำเป็นต้องมีคำว่า help ในประโยค บอทก็สามารถเรียนรู้ได้ว่าจากข้อความนี้สิ่งที่ผู้ใช้งานต้องการคืออะไร โดยไม่ต้องมีคำสั่งอยู่ในข้อความที่ผู้ใช้งานเข้ามา

การทำงานของแอปพลิเคชัน LUIS ร่วมกับการใช้คำสั่งแบบปกติในการค้นหาข้อมูลจากปฏิทินของทีม โดยเมื่อรับข้อความจากผู้เข้ามาแล้ว ถ้าไม่สามารถประมวลผลเป็นกฎที่เขียนไว้ได้ จะทำการส่งข้อความนั้นให้แอปพลิเคชัน LUIS ในการประมวลผล เมื่อแอปพลิเคชัน LUIS ประมวลผลจากข้อมูลที่เทรนไว้ได้เป็น intent “projectinfo” แอปพลิเคชัน LUIS จะส่งข้อมูลกลับมาให้เซทบอทประมวลผล อยู่ในรูปของ JSON Object โดยใน JSON Object นั้นจะประกอบด้วย intent และ entity คือสิ่งที่เซทบอทต้องใช้ในการประมวลผล เมื่อเซทบอทรับข้อมูลมาแล้ว เซทบอทจะใช้ข้อมูลนั้นในการประมวลผลตามกฎที่เขียนไว้ โดยการไปค้นหาข้อมูลจากปฏิทินงานของทีมเหมือนการใช้คำสั่งปกติ และส่งข้อมูลที่ผู้ใช้งานต้องการไปแสดงผลในโปรแกรม Skype ตามภาพที่ 4.17



ภาพที่ 4.17 ตัวอย่างการใช้ LUIS ในการหาข้อมูล project

อีกตัวอย่างหนึ่งในการค้นหาเอกสารโดยไม่ต้องใช้คำสั่ง การทำงานของการใช้แอปพลิเคชัน LUIS ร่วมกับการใช้คำสั่งแบบปกติในการค้นหาเอกสาร โดยเมื่อรับข้อความจากผู้เข้ามาแล้ว และไม่สามารถประมวลผลเป็นกฎที่เขียนไว้ได้ จะทำการส่งข้อความนั้นให้แอปพลิเคชัน LUIS ในการประมวลผล เมื่อแอปพลิเคชัน LUIS ประมวลผลจากข้อมูลที่เทรนไว้ได้เป็น intent “document” แอปพลิเคชัน LUIS จะส่งข้อมูลกลับมาให้เซทบอทประมวลผล อยู่ในรูปของ JSON Object เมื่อเซทบอทรับข้อมูลมาแล้ว เซทบอทจะใช้ข้อมูลนั้นในการประมวลผลตามกฎที่เขียนไว้ โดยการไปค้นหาเอกสารจาก PTP Support Library เหมือนการใช้คำสั่งปกติ และส่งข้อมูลที่ผู้ใช้ต้องการไปแสดงผลในโปรแกรม Skype ตามภาพที่ 4.18

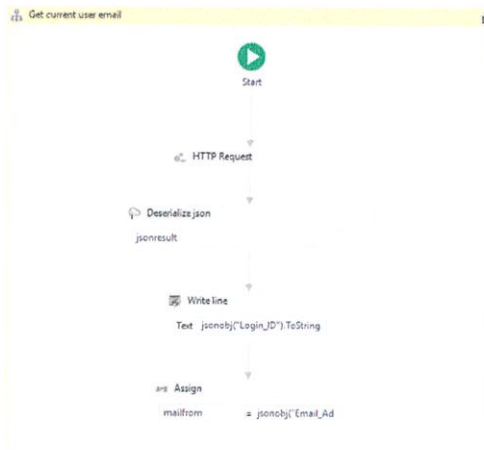


ภาพที่ 4.18 ตัวอย่างการใช้ LUIS ในการหาข้อมูลใน PTP support library

4.2 Vendor invoice management (VIM) automated testing tool

การทำงานของโปรแกรมนั้นจะเริ่มที่ไฟล์ Main.xaml โดยผู้ใช้ต้องตั้งค่าตัวแปรเริ่มต้นให้กับโปรแกรมใน Activity ชื่อ Assign ค่าที่ผู้ใช้ต้องกำหนดให้กับโปรแกรมได้แก่

- กำหนดชื่อระบบที่ต้องการใช้ในโปรแกรม โดยกำหนดในตัวแปร “system”
- กำหนดชื่อไฟล์ของข้อมูลที่จะใช้ในการทดสอบ โดยกำหนดในตัวแปร “excelfilename”
- กำหนดอีเมลล์ของผู้ส่ง โดยใช้การดึงค่าจาก API เพื่อดึงข้อมูลผู้ใช้ปัจจุบัน (current logon user) และกำหนดค่าในตัวแปร “mailfrom” ตามภาพที่ 4.19



ภาพที่ 4.19 flow การใช้ API ในการดึงข้อมูลของผู้ใช้ปัจจุบัน (current logon user)

- กำหนดอีเมลล์ของผู้รับในตัวแปร “mailto”

หลังจากนั้นโปรแกรมจะเรียก activity ชื่อ switch ในการเลือกว่าจะต้องเรียกใช้ไฟล์ไหนในการประมวลผลต่อไป ค่าที่ใช้ในการเลือกใน switch คือค่าตัวแปร “system” ตามภาพที่ 4.20

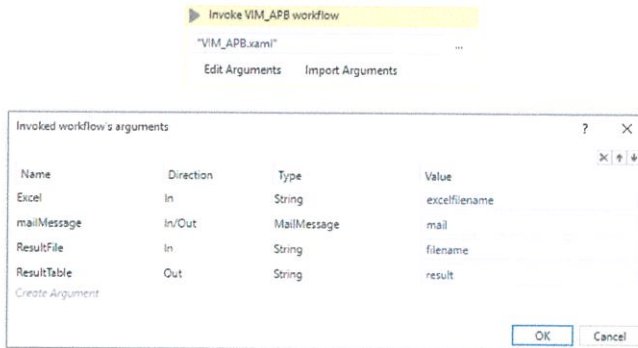


ภาพที่ 4.20 โปรแกรมหลัก (main flow)

หลังจากกำหนดค่าต่างๆแล้ว ถัดไปใช้ activity ชื่อ invoke workflow ในการเรียกใช้ไฟล์ที่ใช้ทำ VIM posting สำหรับแต่ละระบบ และส่งค่าตัวแปรเข้าไปในไฟล์นั้นๆผ่านการใช้อาร์กิวเมนต์ตามภาพที่ 4.21

- Argument ชื่อ Excel ใช้ส่งตัวแปร excelfilename ซึ่งคือตัวแปรเก็บชื่อไฟล์เอ็กเซลที่ใช้ประมวลผล
- Argument ชื่อ mailMessage ใช้ส่งตัวแปร mail ซึ่งคือตัวแปรที่ใช้ในการส่งอีเมลล์

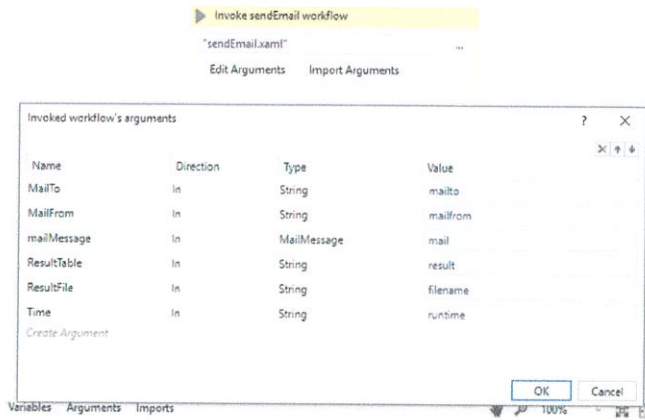
- Argument ชื่อ ResultTable ใช้ส่งตัวแปร result ซึ่งคือตัวแปรเก็บค่าผลลัพธ์ของการทำ VIM posting
- Argument ชื่อ ResultFile ใช้ส่งตัวแปร filename ซึ่งคือตัวแปรที่เป็นชื่อไฟล์เอ็กเซลของผลลัพธ์



ภาพที่ 4.21 ตัวอย่างการทำ invoke workflow

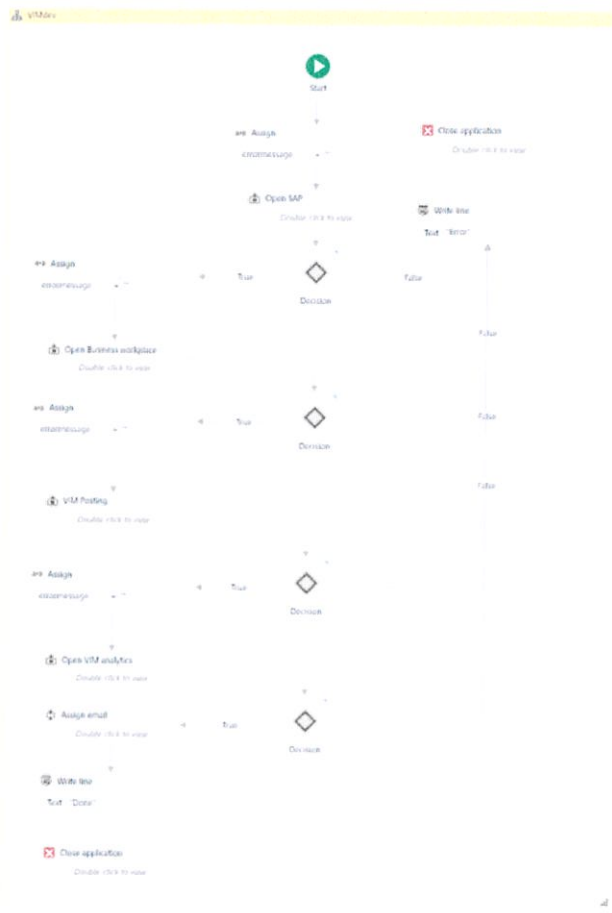
หลังจากไฟล์ที่ VIM posting สำหรับแต่ละระบบทำงานเสร็จ โปรแกรมจะกลับมาที่หน้าโปรแกรมหลัก นั่นคือไฟล์ Main.xaml ถัดไปใช้ activity invoke workflow ในการ เรียกไฟล์ที่ทำหน้าที่ส่งอีเมลผลลัพธ์หาผู้ใช้ และส่งค่าตัวแปรเข้าไปในไฟล์นั้นๆผ่านการ ใช้ argument ตามภาพที่ 4.22

- Argument ชื่อ MailTo ใช้ส่งตัวแปร mailto ซึ่งคือตัวแปรที่ใช้เก็บอีเมลของผู้รับ
- Argument ชื่อ MailFrom ใช้ส่งตัวแปร mailfrom ซึ่งคือตัวแปรที่ใช้เก็บอีเมลของผู้ส่ง
- Argument ชื่อ mailMessage ใช้ส่งตัวแปร mail ซึ่งคือตัวแปรเก็บ object ที่ใช้ส่งอีเมล
- Argument ชื่อ ResultTable ใช้ส่งตัวแปร result ซึ่งคือตัวแปรเก็บค่าผลลัพธ์ของการทำ VIM posting
- Argument ชื่อ ResultFile ใช้ส่งตัวแปร filename ซึ่งคือตัวแปรที่เป็นชื่อไฟล์เอ็กเซลของผลลัพธ์
- Argument ชื่อ Time ใช้ส่งตัวแปร runtime ซึ่งคือตัวแปรที่เก็บเวลาเริ่มต้นของการเริ่มโปรแกรม



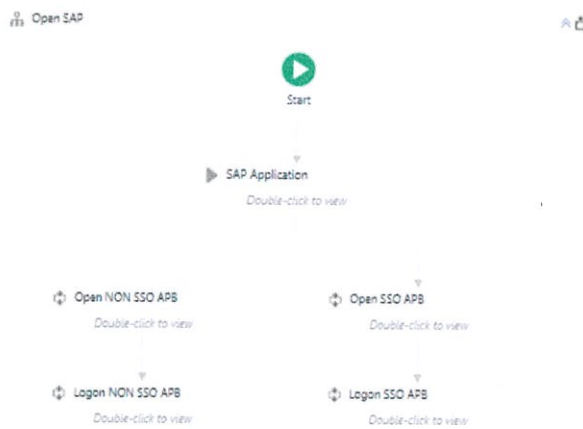
ภาพที่ 4.22 ตัวอย่างการทำ invoke send email workflow

จากภาพที่ 4.23 Main flow ของไฟล์ที่ใช้ทำ VIM posting สำหรับแต่ละระบบนั้นจะประกอบด้วย 5 ขั้นตอนหลัก ได้แก่ การเปิดโปรแกรม SAP การเปิด business workplace การทำ VIM Posting การเปิด VIM analytics และการกำหนดค่าให้กับตัวแปรที่ใช้ในการส่งอีเมลล์ โดย flow จะเริ่มจากการเปิดโปรแกรม SAP และทำการ logon โปรแกรม หลังจากนั้นเปิด business workplace เพื่อหา VIM number ที่ต้องการ process จากนั้นทำ VIM posting โดยการใส่ข้อมูลตามช่องต่างๆแล้ว submit ซึ่งแต่ละ system และประเภทของ invoice จะมีวิธีการโพส VIM ที่ไม่เหมือนกัน ทำให้แต่ละ system จะมี flow ที่ต่างกันในส่วนนี้ เมื่อทำการโพส VIM ครบทุกตัวแล้วโปรแกรมจะเปิด VIM analytics เพื่อเก็บค่าผลลัพธ์ของการทำโพส VIM แต่ละตัว และขั้นตอนสุดท้ายคือการนำผลลัพธ์ไปกำหนดในตัวแปรที่จะใช้ในการส่งอีเมลล์ผู้ใช้ต่อไป



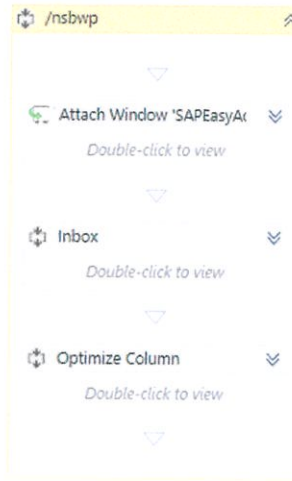
ภาพที่ 4.23 ตัวอย่าง main flow ของ system APB

Flowchart แรกที่จะเริ่มทำงานคือ flowchart ชื่อ Open SAP โดยจะทำการเปิดโปรแกรม SAP และเลือกการล็อกคอนแบบเป็น single sign on ตามภาพที่ 4.24



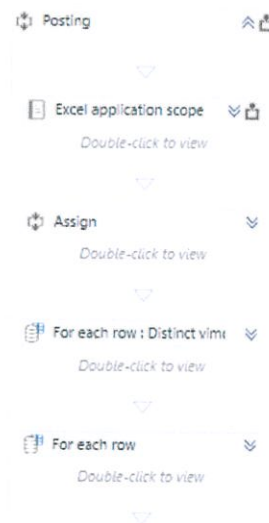
ภาพที่ 4.24 การทำงานใน flowchart ชื่อ Open SAP

Flowchart ถัดมาคือ flowchart ชื่อ /nsbwp ตามภาพที่ 4.25 โดย flowchart นี้จะเปิด business workplace โดยการใส่ t-code sbwp จากนั้นเลือก inbox และเลือก layout ใหม่ เพื่อให้โปรแกรมสามารถมองเห็น vim number เพื่อที่จะสามารถ click มาโผล่ได้



ภาพที่ 4.25 การทำงานใน flowchart ชื่อ /nsbwp

Flowchart ถัดมาคือ flowchart ชื่อ Posting ตามภาพที่ 4.26 โดย flowchart นี้จะเป็นการทำ VIM posting โปรแกรมจะเริ่มจากการเปิด excel โดยใช้ activity ชื่อ Excel application scope และใช้ activity ชื่อ Read Range ในการอ่านข้อมูล excel มาเก็บในตัวแปรชนิด datatable จากนั้นประกาศตัวแปรใน sequence ชื่อ assign หลังจากนั้นทำการลบแถวที่มี VIM number ซ้ำกันออกโดยสร้างเป็น datatable ตัวใหม่ ที่มี VIM number ไม่ซ้ำกัน



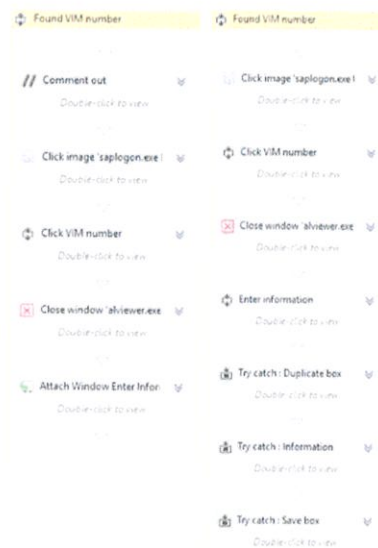
ภาพที่ 4.26 การทำงานใน flowchart ชื่อ Posting

ทำการรูปตามข้อมูลแต่ละแถว และกำหนดค่าตัวแปรแต่ละตัวโดยใช้ค่าจากใน excel กำหนดให้ตัวแปรต่างๆ ใน activity assign จากนั้นตรวจสอบว่าเป็น document ชนิด PO หรือ NPO เนื่องจากสองชนิดมีวิธีการโพสใบแจ้งหนี้ที่แตกต่างกัน ตามภาพที่ 4.27



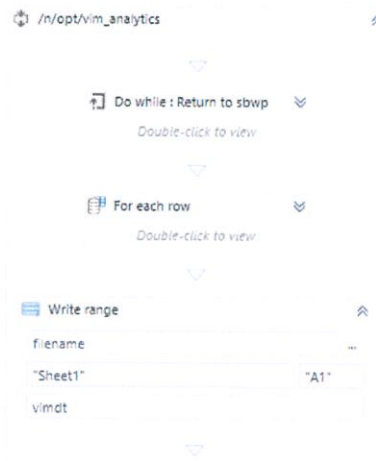
ภาพที่ 4.27 การทำงานในรูปการทำ VIM posting

สำหรับภาพที่ 4.28 นั้นแสดงการทำงานในการโพสใบแจ้งหนี้แต่ละแบบ ซึ่งมีส่วนติดต่อผู้ใช้งานต่างกัน ทำให้ขั้นตอนในการทำส่วนนี้ต่างกันไปด้วย



ภาพที่ 4.28 VIM posting แบบ PO และแบบ NPO

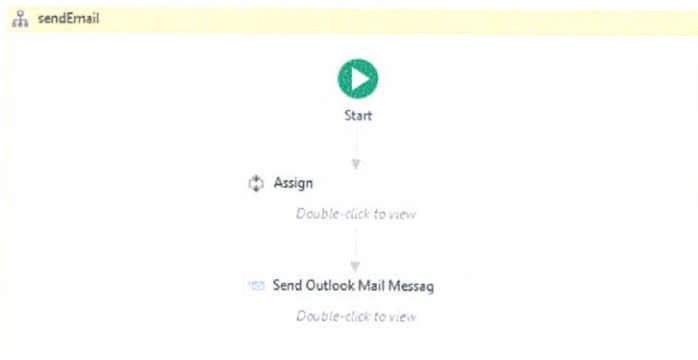
หลังจากทำการโพส VIM ครบทุกตัวแล้ว โปรแกรมจะทำการเก็บผลลัพธ์โดยการเปิด VIM analytics และใส่ VIM number จากนั้นโปรแกรมจะดึงข้อมูลผลลัพธ์ออกจากโปรแกรม SAP ที่ละตัวจนครบทุกตัว แล้วเพิ่มผลลัพธ์ลงในไฟล์เอ็กเซล ตามที่แสดงในภาพที่ 4.29



ภาพที่ 4.29 การทำงานใน flowchart ชื่อ /n/opt/vim_analytics

จากนั้นทำการรวบรวมข้อมูลแล้วส่งกลับไปโปรแกรมหลัก จากนั้นโปรแกรมหลักจะไป invoke ไฟล์ชื่อ sendEmail เพื่อทำการส่งอีเมลให้กับผู้ใช้

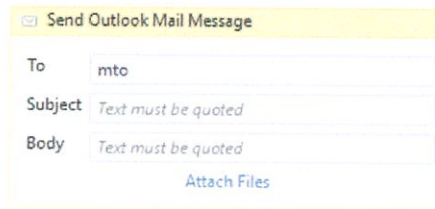
ขั้นตอนสุดท้ายคือการส่งอีเมลโดยใน sequence ชื่อ assign จะเป็นการกำหนดค่าให้กับตัวแปรต่างๆที่ต้องใช้ในการส่งอีเมลได้แก่ body และ subject ของอีเมล และสุดท้ายใช้ activity ชื่อ Send Outlook Mail Message ในการส่ง email โดยภาพที่ 4.30 จะแสดงการทำงานภายใน flowchart ชื่อ sendEmail ทั้งหมด ภาพที่ 4.31 จะแสดงการตั้งค่าตัวแปรต่าง ๆ และภาพที่ 4.32 จะเป็นการแสดงการใช้ activity ชื่อ send Outlook Mail Message



ภาพที่ 4.30 การทำงานใน flowchart ชื่อ sendEmail



ภาพที่ 4.31 ภายใน Activity ชื่อ Assign ใน flowchart ชื่อ sendEmail



ภาพที่ 4.32 ภายใน Activity ชื่อ send Outlook Mail Message ใน flowchart ชื่อ sendEmail

ภาพที่ 4.33 และ ภาพที่ 4.34 นั้นเป็นการแสดงตัวอย่างของอีเมลที่โปรแกรมส่งให้ผู้ใช้ หลังจากโปรแกรมทำงานเสร็จ ในอีเมลประกอบด้วย

- VIM Number คือเลข VIM ที่ทำการโพสต์ข้อมูล
- Status คือ สถานะของใบแจ้งหนี้ นั้น ๆ
- Screenshot กรณีที่เกิดข้อผิดพลาดใน VIM number นั้น ๆ โปรแกรมจะจับภาพหน้าจอเอาไว้ เพื่อแจ้งผู้ใช้งานว่าเกิดข้อผิดพลาดที่ขั้นตอนไหน
- Attachment file ใน attachment นั้นจะประกอบด้วยข้อมูลที่ใช้ทดสอบและผลลัพธ์ของ VIM นั้นๆ

Reply Reply All Forward IM
 Mon 11/27/2017 8:35 AM
Seedang, Nantatchaporn /C
VIM Posting - 11/27/2017 08:30:05
 To **Seedang, Nantatchaporn /C**
 Message Result 2017-11-27_083005.xlsx (9 KB)

Automated test on 11/27/2017 08:30:05
 System : APB
 Please see the result below,

VIM Number	Status	Screenshot (if error occurred)
754177	Posted	
754178	Blocked	
754179	Awaiting Approval	
754181	Posted	
754182	Blocked	

For more information please see the attachment file.

ภาพที่ 4.33 ตัวอย่างอีเมลล์รายงานผลการทำ VIM Posting

Reply Reply All Forward IM
 Fri 10/27/2017 3:15 PM
Seedang, Nantatchaporn /C
VIM Posting - 10/27/2017 15:12:28
 To **Seedang, Nantatchaporn /C**
 Message Result 2017-10-27_151228.xlsx (6 KB)

To: nanatchaporn.seedang@ecooonmobd.com
 Automated test on 10/27/2017 15:12:28
 Please see the result below,

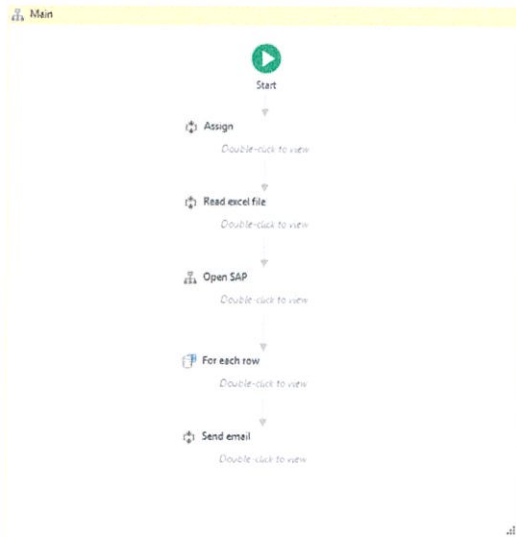
VIM Number	Status	Screenshot (if error occurred)
1620182	Created (fail)	

For more information please see the attachment file.

ภาพที่ 4.34 ตัวอย่างอีเมลล์รายงานผลการทำ VIM Posting (กรณีมีข้อผิดพลาดจากโปรแกรม)

4.3 Optical character recognition (OCR) automated posting tool

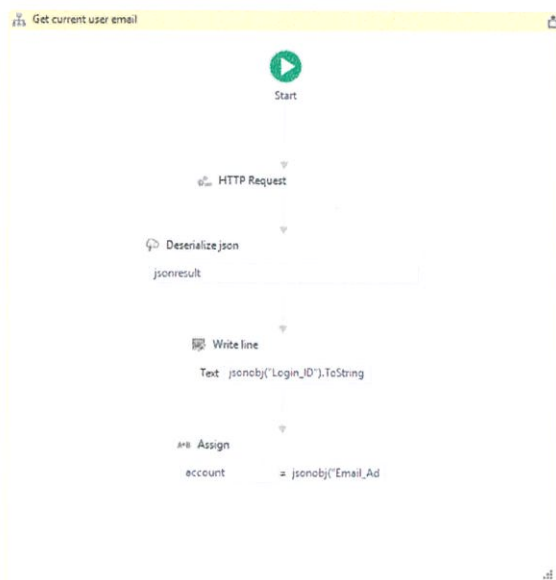
Flow การทำงานของโปรแกรมแบ่งเป็น 5 ส่วนหลักๆ ตามภาพที่ 4.35 เริ่มจาก กำหนดค่า เริ่มต้นของโปรแกรม จากนั้นอ่านค่า idoc number ที่ต้องการทดสอบจากไฟล์เอ็กเซล แล้วเปิดโปรแกรม SAP เพื่อวนลูปตามค่า idoc number ในไฟล์เอ็กเซลและสุดท้ายส่ง email ให้กับผู้ใช้งานโดยขั้นตอนทั้งหมดจะอยู่ใน activity flowchart ชื่อ Main



ภาพที่ 4.35 หน้าโปรแกรมหลัก

การกำหนดตัวแปรเริ่มต้นของโปรแกรมใน sequence ชื่อ Assign ตามภาพที่ 4.36 ทั้งหมดนั้นจะใช้ Activity ชื่อ assign

- กำหนดชื่อไฟล์เอ็กเซล ที่ใช้ในโปรแกรมในตัวแปร “excelfile”
- กำหนดชื่อระบบ ที่ต้องการ automated ในตัวแปร “system”
- กำหนดที่อยู่ของ text file (directory) ในตัวแปร “path”
- กำหนด email ของผู้รับในตัวแปร “email”
- กำหนด email account โดยใช้การดึงค่าจาก API เพื่อดึงข้อมูลของผู้ใช้งานปัจจุบัน (current logon user) และกำหนดค่าในตัวแปร “account” ตามภาพที่ 4.37

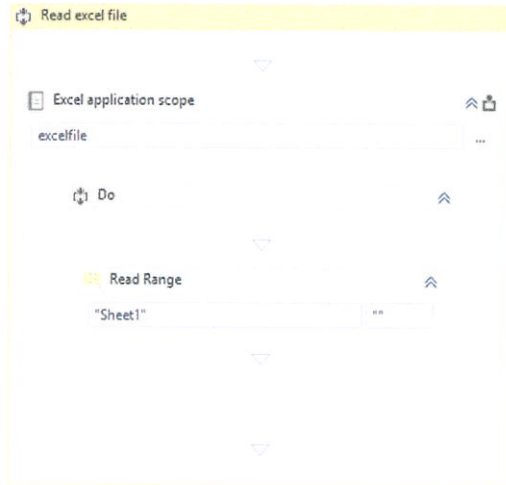


ภาพที่ 4.36 การใช้ API ในการดึงข้อมูลของผู้ใช้งานปัจจุบัน (current logon user)



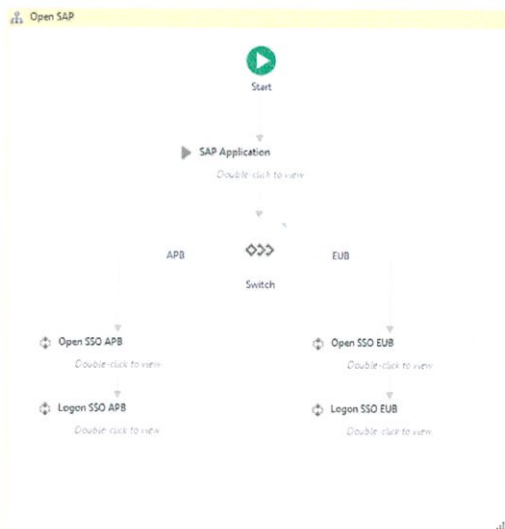
ภาพที่ 4.37 การกำหนดค่าเริ่มต้นของโปรแกรมใน sequence ชื่อ Assign

Flowchart ชื่อ Read excel file ตามภาพที่ 4.38 นั้นใช้ activity ชื่อ Excel application scope ในการเปิดไฟล์เอ็กเซล ใช้ตัวแปร “excelfile” เป็นชื่อไฟล์เอ็กเซลที่ต้องการเปิด ใช้ activity ชื่อ Read Range ในการอ่านค่าในไฟล์เอ็กเซลและเก็บลงในตัวแปรที่เป็นชนิด datatable



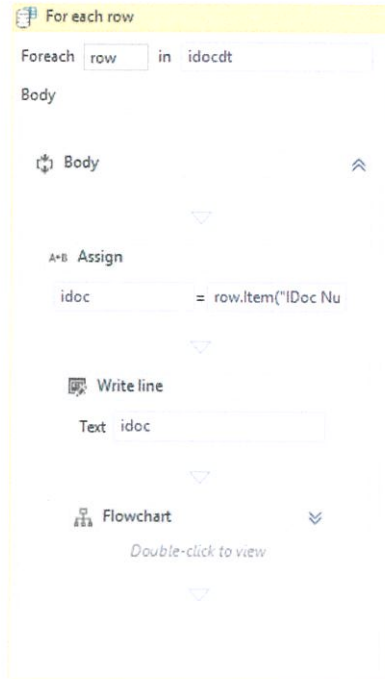
ภาพที่ 4.38 การอ่านข้อมูลจากไฟล์เอ็กเซลใน sequence ชื่อ Read excel file

จากนั้นจะเริ่มที่ Flowchart ชื่อ Open SAP ตามภาพที่ 4.39 โดยเริ่มจากการเปิดโปรแกรม SAP โดยเริ่มจากการใช้ activity ชื่อ Open Application เพื่อเปิดโปรแกรม SAP จากนั้นใช้ activity switch case ในการเลือกว่าต้องเปิดระบบตัวไหน ตามค่าตัวแปร “system” ที่กำหนดไว้ในการตั้งค่าเริ่มต้นของโปรแกรม หลังจากนั้นเปิดระบบที่ต้องการและ logon ผ่านการใช้ single sign on (SSO)



ภาพที่ 4.39 การเปิดโปรแกรม SAP ตามระบบที่ได้กำหนดใน flowchart ชื่อ Open SAP

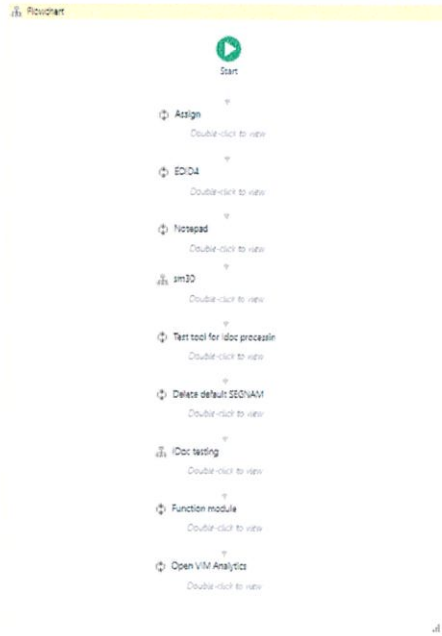
จากนั้นใช้ activity For each row ในการลูปตามแต่ละค่าในไฟล์เอ็กเซล ใช้ activity assign ในการกำหนดค่า IDoc number เก็บไว้ในตัวแปรชื่อ “idoc” จากนั้นทำการใส่ข้อมูลตาม segment name โดยจะทำกระบวนการนี้ใน activity ชื่อ flowchart ตามภาพที่ 4.40



ภาพที่ 4.40 การทำงานใน activity ชื่อ For each row

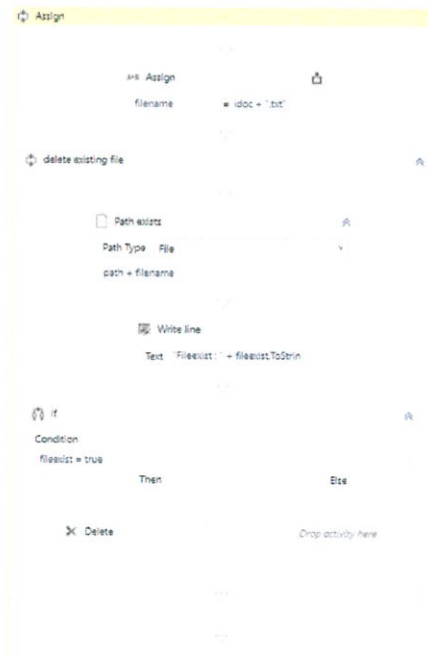
การทำงานใน activity flowchart ตามภาพที่ 4.41 นั้นจะประกอบด้วย 9 ขั้นตอนหลัก ได้แก่

- การกำหนดค่าให้ตัวแปรที่ต้องใช้
- table edid4 เพื่อดึงข้อมูลออกมาเป็นไฟล์ข้อความจากโปรแกรม SAP
- โปรแกรม notepad เปิดไฟล์ข้อความเพื่อบันทึกไฟล์ข้อความ
- t-code sm30 เพื่อเปลี่ยนสถานะ OCR จาก completed เป็น waiting
- t-code we19 เพื่อเปิด IDoc testing tool
- ลบค่า segment name ที่มีมาให้
- ทำการใส่ข้อมูลแต่ละ segment name
- การเรียกใช้ function module
- โปรแกรม VIM Analytics



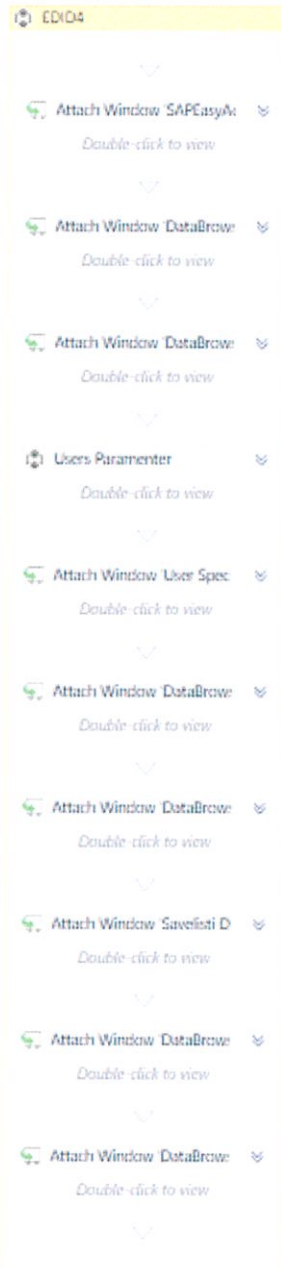
ภาพที่ 4.41 flowchart หลักในการทำ IDoc testing

เริ่มจากการกำหนดค่าตัวแปรที่ชื่อ “filename” โดยใช้ activity assign และเช็คว่ามีไฟล์อยู่แล้วหรือไม่ โดยใช้ activity path exists ที่จะให้ค่าออกมาเป็น Boolean หลังจากนั้นนำค่า Boolean ที่ได้ไปเช็คใน activity if ถ้าค่า Boolean นั้นเท่ากับ true แสดงว่ามีไฟล์อยู่แล้ว ก็จะใช้ activity delete ให้ลบไฟล์นั้นทิ้งไป ถ้าค่า Boolean นั้นเท่ากับ false ก็ไม่ต้องทำอะไรให้ข้ามไปตามภาพที่ 4.42



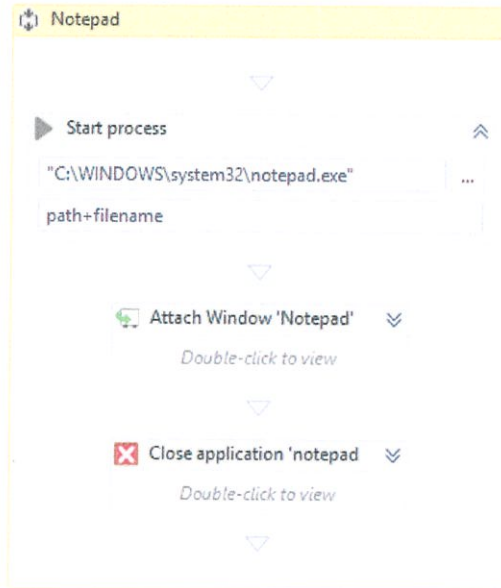
ภาพที่ 4.42 กำหนดค่า และ เช็คว่ามีไฟล์อยู่แล้วหรือไม่

จากนั้นจะเริ่มทำงานต่อไปที่ sequence ชื่อ EDID4 ตามภาพที่ 4.43 โดยใน sequence นี้จะทำการเปิดข้อมูลใบแจ้งหนี้ที่ต้องการคัดลอกข้อมูล และบันทึกข้อมูลใบแจ้งหนี้เป็นไฟล์ข้อความเพื่อใช้ต่อไป



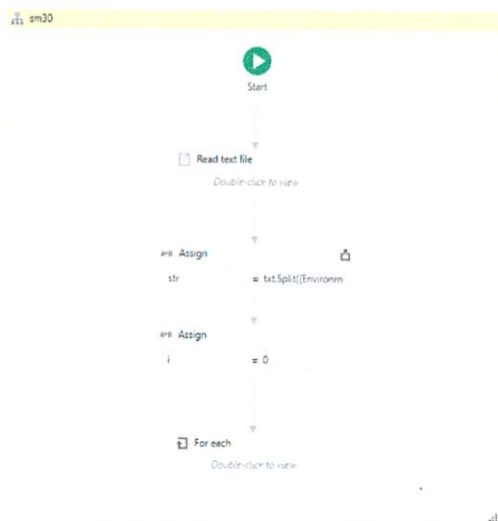
ภาพที่ 4.43 การทำงานใน sequence ชื่อ EDID4

เนื่องจากไฟล์ข้อความที่ได้จากโปรแกรม SAP ไม่สามารถนำมาใช้ประมวลผลได้เลยทันที จึงต้องทำการเปิดไฟล์ข้อความนั้นก่อน โดยใช้โปรแกรม notepad แล้วทำการบันทึกไฟล์ใหม่อีกครั้ง ก่อนนำไฟล์ไปใช้ตามภาพที่ 4.44



ภาพที่ 4.44 การทำงานของ sequence ชื่อ Notepad

จากนั้นตามภาพที่ 4.45 โปรแกรมจะอ่านไฟล์ข้อความเพื่อหา Document ID โดยเริ่มจากใช้ activity read text file ในการอ่านไฟล์ข้อความมาเก็บในตัวแปร string หลังจากนั้นแยก string โดยใช้ new line เป็นตัวแยก string แล้วเก็บในตัวแปรที่เป็น array string หลังจากนั้นตามภาพที่ 4.46 จะใช้ activity for each ในการloop string แต่ละตัว เพื่อหา Document ID เมื่อเจอแล้วให้เปิด t-code sm30 เพื่อเปลี่ยนสถานะของ OCR จาก completed ให้เป็น waiting เพื่อให้ Document ID นั้นสามารถนำกลับมาใช้ประมวลผลได้อีกครั้ง

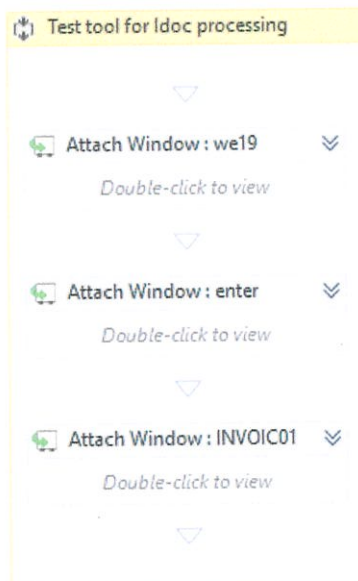


ภาพที่ 4.45 การทำงานของ sequence ชื่อ sm30



ภาพที่ 4.46 การทำงานของ for each activity ใน sequence ชื่อ sm30

ถัดไปใน activity ชื่อ Test tool for Idoc processing ตามภาพที่ 4.47 นั้นจะทำการเปิดหน้าโปรแกรม IDoc processing tool และเลือก Basic type เป็น INVOIC01 เพื่อทำการโพส OCR ใหม่



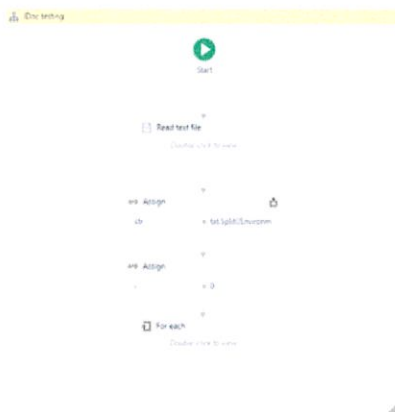
ภาพที่ 4.47 การทำงานของ sequence ชื่อ Test tool for Idoc processing

เมื่อเข้ามาในหน้า Idoc testing tool แล้ว ก่อนโพส segment name ใหม่ นั้นต้องลบค่า segment name ที่ระบบมีไว้ให้ก่อน หลังจากนั้นทำการเพิ่ม header ของ edidc โดย header ของ EDIDC จะเปลี่ยนไปตามแต่ละ system โดยการใช้ switch activity ในการกำหนดค่า header ตามตัวแปร system ที่ได้กำหนดไว้ตอนเริ่มต้น ตามภาพที่ 4.48



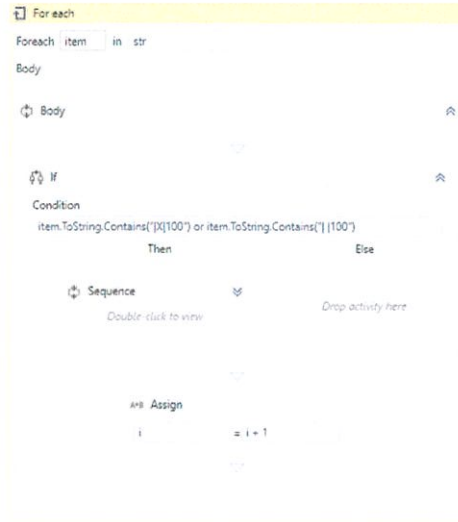
ภาพที่ 4.48 การทำงานของ sequence ชื่อ Delete default SEGNAME

หลังจากใส่ header แล้ว โปรแกรมจะอ่านไฟล์ข้อความโดยใช้ activity read text file ในการอ่านไฟล์ข้อความและเก็บค่าลงในตัวแปร string จากนั้นทำการแยก string โดยใช้ new line เป็นตัวแยก แล้วเก็บในตัวแปรชนิด array string แล้วใช้ activity for each ในการลูบผ่านค่าแต่ละตัวในตัวแปร array string ตามภาพที่ 4.49



ภาพที่ 4.49 การทำงานของ sequence ชื่อ IDoc processing

แต่ละบรรทัดของข้อมูลจะถูกนำมาเช็คเงื่อนไขเพื่อดูว่าใช้ line item ที่ต้องนำมาใช้หรือไม่ ถ้าใช้ line item ที่ต้องการ โปรแกรมจะทำการ substring เพื่อดึงค่าต่างๆในบรรทัดนั้นมาใช้ในโปรแกรม ตามภาพที่ 4.50



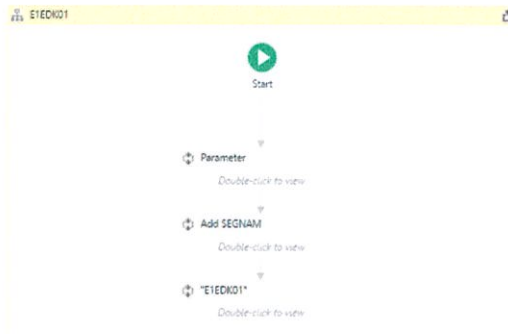
ภาพที่ 4.50 การทำงานของ for each activity ใน sequence ชื่อ IDoc processing

ใน for each activity ในภาพที่ 4.51 จะมี activity switch เพื่อใช้ในการเพิ่ม segment name แต่ละตัวเนื่องจากแต่ละ segment name จะมีการใช้ตัวแปรที่แตกต่างกัน ทำให้ต้องสร้าง case สำหรับแต่ละ segment name ด้วย



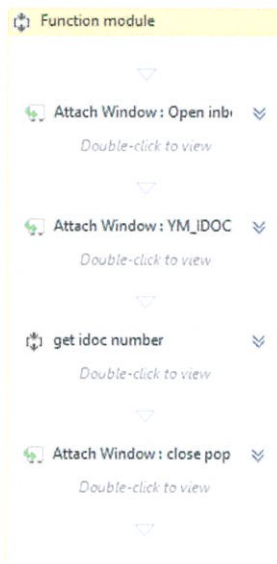
ภาพที่ 4.51 การทำงานภายใน activity switch สำหรับแต่ละ segment name

ในแต่ละ case นั้นจะประกอบด้วยการทำงานทั้งหมด 3 ขั้นตอน ตามภาพที่ 4.52 ขั้นตอนแรกคือการกำหนดค่าตัวแปรที่ต้องใส่ โดยแต่ละ segment name จะใช้ไม่เหมือนกัน ขั้นตอนที่สองคือการเพิ่ม segment name นั้น ขั้นตอนที่สามคือการกรอกข้อมูลของ segment name นั้นโดยใช้ตัวแปรที่ได้กำหนดไว้ในขั้นตอนที่หนึ่ง



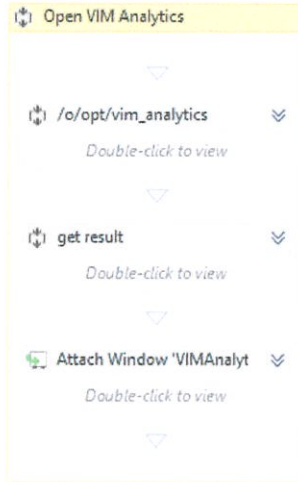
ภาพที่ 4.52 ตัวอย่างการทำงานของ segment name ชื่อ E1EDK01

เมื่อวนลูปครบทุก line item แล้วโปรแกรมจะสั่งงานให้เรียก inbound function module เพื่อทำการประมวลผลและจะได้หน้าต่าง pop up แสดง IDoc number ตัวใหม่ ให้ทำการเก็บค่า IDoc number ตัวใหม่นั้นไว้ การทำงานในโปรแกรมนั้นตามภาพที่ 4.53



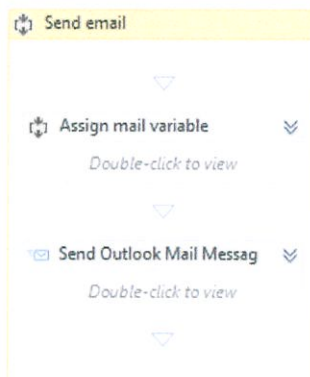
ภาพที่ 4.53 การทำงานของ sequence ชื่อ Function module

ถัดไปเป็นการเรียกใช้ Flowchart ชื่อ Open VIM Analytics ตามภาพที่ 4.54 การทำงานใน Flowchart นี้จะเริ่มจากเปิดหน้า VIM Analytics และใส่ IDoc number เพื่อเอาผลลัพธ์การทำงานออกมาจากโปรแกรม SAP โดยใช้ activity get text เพื่อดึงค่าผลลัพธ์ที่ต้องการแต่ละตัวออกมา ได้แก่ IDoc number ของข้อมูลที่นำมาทดสอบ, IDoc number, Document ID, Document number, Document status, Exception reason



ภาพที่ 4.54 การทำงานของ sequence ชื่อ Open VIM Analytics

ขั้นตอนสุดท้ายคือการส่งอีเมล ใน Flowchart ชื่อ Send email ตามภาพที่ 4.55 โดยเริ่มจากการกำหนดค่าให้กับตัวแปรที่ต้องใช้ในการส่งอีเมล ได้แก่ body และ subject ของอีเมล แล้วใช้ activity send outlook mail message ในการส่งอีเมล



ภาพที่ 4.55 การทำงานของ sequence ชื่อ Send email

ภาพที่ 4.56 เป็นการแสดงตัวอย่างของไฟล์ข้อความที่ใช้ทดสอบ โดยเป็นไฟล์ที่ได้จากโปรแกรม SAP และเป็นไฟล์นามสกุล .txt

IDOC number (Test data)	IDOC number	Document ID	Document number	Document status	Exception reason
996805357	000000997251685	1433107		Suspected Duplicate	(NPO) Suspected Duplicate
995727939	000000997251686	1433108		Suspected Duplicate	(PO) Suspected Duplicate

ภาพที่ 4.56 ตัวอย่างไฟล์ข้อความที่ใช้ทดสอบ

ภาพที่ 4.57 นั้นเป็นการแสดงตัวอย่างอีเมลที่ส่งออกไปหลังจากโปรแกรมทำงานจบ ในอีเมลนี้จะประกอบด้วย

- IDOC number (Test data) คือ idoc number เริ่มต้นที่ผู้ใช้ใส่มาให้ใน excel file
- IDOC number คือ idoc number ใหม่ที่ได้จากการรันโปรแกรม
- Document ID คือ vim number
- Document number คือ invoice document number
- Document status คือ สถานะของ document นั้น
- Exception reason คือ เหตุผลถ้า document post ไม่สำเร็จ จะบอกว่าไม่สำเร็จเพราะอะไร

Reply Reply All Forward IM

Tue 11/21/2017 11:32 AM

Seedang, Nantatchaporn /C

OCR - 11/21/2017 11:23:20

To Seedang, Nantatchaporn /C

Automated test on 11/21/2017 11:23:20

Please see the result below,

IDOC number (Test data)	IDOC number	Document ID	Document number	Document status	Exception reason
996805357	000000997251685	1433107		Suspected Duplicate	(NPO) Suspected Duplicate
995727939	000000997251686	1433108		Suspected Duplicate	(PO) Suspected Duplicate

ภาพที่ 4.57 ตัวอย่างอีเมลรายงานผลการทำ OCR

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากผลการพัฒนาเครื่องมือทั้ง 2 ส่วน ได้แก่ ระบบแชทบอท เพื่อใช้ในการค้นหาข้อมูล เอกสาร และการตอบคำถาม และระบบการทำงานอัตโนมัติโดยหุ่นยนต์ที่นำมาใช้กับกระบวนการบันทึกใบแจ้งหนี้ นั้น สามารถสรุปผลการทำงานของเครื่องมือแต่ละชนิดได้ดังนี้

5.1.1 สรุปผลการใช้งานระบบแชทบอท

ผลจากการใช้ระบบแชทบอทในการช่วยตรวจสอบข้อมูลจากปฏิทินของทีม การค้นหาเอกสาร และการตอบคำถามนั้น สำหรับพนักงานที่เข้ามาทำงานใหม่ สามารถลดระยะเวลาที่ใช้ในการค้นหาข้อมูลได้เฉลี่ย 1 ชั่วโมงต่ออาทิตย์ สำหรับพนักงานที่ทำงานมานานแล้วนั้น พบว่ายังไม่เห็นผลลัพธ์ที่ชัดเจนว่าใช้เวลาลดลง เนื่องจากพนักงานที่ทำงานมาอยู่แล้วนั้นคุ้นเคยกับการค้นหาแบบเก่ามากกว่า ทำให้ตอนนี้ระบบแชทบอทเหมาะสมกับผู้ใช้งานที่เป็นพนักงานใหม่ หรือผู้ใช้ที่ไม่คุ้นเคยกับระบบเก่ามากกว่า

5.1.2 สรุปผลการใช้งานกระบวนการการทำงานอัตโนมัติโดยหุ่นยนต์

ผลการใช้งานกระบวนการการทำงานอัตโนมัติโดยหุ่นยนต์ในการนำมาใช้กับกระบวนการบันทึกใบแจ้งหนี้ลงในระบบ SAP แบ่งเป็น 2 เครื่องมือ ได้แก่

5.1.2.1 Vendor invoice management (VIM) automated testing tool

ตารางที่ 5.1 เปรียบเทียบเวลาที่ใช้ในการทำ VIM automated testing tool

	การทำงานด้วยตนเอง	การทำงานอัตโนมัติ
การทำโพสต์ (posting) 5 ครั้ง	9 นาที	3 นาที
การทำโพสต์ (posting) 1 ครั้ง	1.8 นาที	0.6 นาที
การรายงานผลลัพธ์ 5 ครั้ง	10 นาที	0.35 นาที
การรายงานผลลัพธ์ 1 ครั้ง	2 นาที	0.07 นาที
เวลาทั้งหมดที่ใช้ทำงาน 5 ครั้ง	19 นาที	3.35 นาที
เวลาทั้งหมดที่ใช้ทำงาน 1 ครั้ง	3.8 นาที	0.67 นาที

การทำงานแบบอัตโนมัติ (automated) ใช้เวลาดลดลงประมาณ 5 เท่าเมื่อเทียบกับการทำงานด้วยตนเอง

5.1.2.2 Optical character recognition (OCR) automated posting tool

ตารางที่ 5.2 เปรียบเทียบเวลาที่ใช้ในการทำ OCR automated testing tool

	การทำงานด้วยตนเอง	การทำงานอัตโนมัติ
การเตรียมข้อมูลเพื่อทดสอบ	4 นาที	1.5 นาที
การทำโพสต์ (posting)	10 นาที	3 นาที
การรายงานผลลัพธ์	2 นาที	0.2 นาที
เวลาทั้งหมด	16 นาที	4.7 นาที

การทำงานแบบอัตโนมัติ (automated) ใช้เวลาลดลงประมาณ 3 เท่าเมื่อเทียบกับการทำงานด้วยตนเอง

5.2 ข้อเสนอแนะ

ข้อเสนอแนะของโครงการงานสหกิจศึกษานี้แบ่งตามเครื่องมือต่างๆ ที่ได้พัฒนาขึ้นนั้น มีดังนี้

5.2.1 ระบบแชทบอท

การพัฒนาแชทบอทโดยใช้เครื่องมือ LUIS ตอนนี้สามารถใช้ได้แค่ development environment เท่านั้น การนำไปใช้บน acceptance และ production environment ยังติดเรื่องความปลอดภัย (security) และยังต้องศึกษาความคุ้มค่าในการพัฒนาให้มากกว่านี้

5.2.2 การทดสอบโปรแกรมอัตโนมัติโดยใช้เทคโนโลยี RPA

การทดสอบโปรแกรมอัตโนมัติโดยใช้เทคโนโลยี RPA โดยใช้โปรแกรม UiPath ในการใช้งานร่วมกับระบบเอสเอพี (SAP) นั้นยังไม่เสถียรนัก ทำให้ต้องใช้เวลาในการพัฒนาเพิ่มขึ้นเพื่อให้โปรแกรมสามารถทำงานได้อย่างราบรื่น และยังต้องจัดการกับข้อผิดพลาดต่างๆ ที่อาจจะเกิดขึ้นได้ ซึ่งทำให้เวลาที่โปรแกรมใช้ในการทำกระบวนการอัตโนมัติ (automated) ใช้เวลานานขึ้น

เอกสารอ้างอิง

- [1] Application programming interface. (10 ธันวาคม 2560). เข้าถึงได้จาก:
https://en.wikipedia.org/wiki/Application_programming_interface
- [2] C Sharp (programming language). (14 ธันวาคม 2560). เข้าถึงได้จาก:
[https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))
- [3] DevOps: the Gartner Toolchain. (11 ธันวาคม 2560). เข้าถึงได้จาก:
<https://www.bluebridgesoftware.com/blog/26-devops-the-gartner-toolchain.html>
- [4] Enterprise resource planning. (11 ธันวาคม 2560). เข้าถึงได้จาก:
https://en.wikipedia.org/wiki/Enterprise_resource_planning
- [5] How Does SAP Work? (12 ธันวาคม 2560). เข้าถึงได้จาก:
<https://sap-certification.info/how-does-sap-work>
- [6] Introducing RPA. (11 ธันวาคม 2560). เข้าถึงได้จาก:
<https://www.uipath.com/automate/robotic-process-automation>
- [7] Language Understanding (LUIS) Documentation. (10 ธันวาคม 2560). เข้าถึงได้จาก:
<https://docs.microsoft.com/en-us/azure/cognitive-services/LUIS/>
- [8] Natural language processing. (10 ธันวาคม 2560). เข้าถึงได้จาก:
https://en.wikipedia.org/wiki/Natural_language_processing
- [9] Robotic Process Automation (RPA). (11 ธันวาคม 2560). เข้าถึงได้จาก:
<https://www.roboyo.de/en/robotic-process-automation/>
- [10] SAP Modules Overview. (13 ธันวาคม 2560). เข้าถึงได้จาก:
<http://www.traininginchennai.net/sap-training-in-chennai/sap-modules-overview>
- [11] Team Foundation Server. (12 ธันวาคม 2560). เข้าถึงได้จาก:
<https://www.visualstudio.com/tfs/>

- [12] THE EMERGENCE OF ROBOTIC PROCESS AUTOMATION SOFTWARE. (11 ธันวาคม 2560). เข้าถึงได้จาก: <https://www.accenture.com/sg-en/robotic-process-automation-software>
- [13] The Importance of Different Agile Methodologies Included in Agile Manifesto. (11 ธันวาคม 2560). เข้าถึงได้จาก: <https://www.linkedin.com/pulse/importance-different-agile-methodologies-included-manifesto-kapoor>
- [14] The UiPath Studio Guide. (12 ธันวาคม 2560). เข้าถึงได้จาก: <https://studio.uipath.com/>
- [15] UCMA 4.0 business scenarios. (12 ธันวาคม 2560). เข้าถึงได้จาก: <https://msdn.microsoft.com/en-us/library/office/dn465936.aspx>
- [16] User-centered Design: A cooperative process. (11 ธันวาคม 2560). เข้าถึงได้จาก: <http://www.ergonomen.ch/en/services/user-centered-design/>
- [17] Visual Basic .NET. (14 ธันวาคม 2560). เข้าถึงได้จาก: https://en.wikipedia.org/wiki/Visual_Basic_.NET
- [18] Visual Studio IDE. (12 ธันวาคม 2560). เข้าถึงได้จาก: <https://www.visualstudio.com/vs/>
- [19] What is UCMA 4.0. (12 ธันวาคม 2560). เข้าถึงได้จาก: <https://msdn.microsoft.com/en-us/library/office/dn465943.aspx>

ภาคผนวก ก

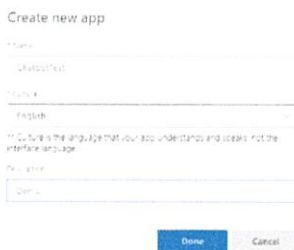
การเริ่มต้นพัฒนาระบบแชทบอท

ขั้นตอนการเริ่มต้นพัฒนาระบบแชทบอทนี้เป็นขั้นตอนที่ใช้ภายในบริษัทเท่านั้น สำหรับการพัฒนาระบบแชทบอทที่ไม่ใช่ภายในบริษัทนั้นจะมีขั้นตอนที่แตกต่างออกไปจากนี้ ส่วนขั้นตอนการพัฒนาแอปพลิเคชัน LUIS นั้นสามารถทำตามได้

1. ขอเป็นผู้พัฒนาแบบ SDO (Service Delivery Organization developer) ผ่านแบบฟอร์มออนไลน์ของทีม Robotics Automation Center of Expertise (RACE) เพื่อให้มีสิทธิ์และสามารถเข้าถึงข้อมูลที่จำกัดแก่ผู้พัฒนาเท่านั้นได้
2. ขอใช้งานซอฟต์แวร์ Visual Studio 2015 เวอร์ชัน Professional ผ่านระบบ IT Services หรือ Software Request ของบริษัท เพื่อให้สามารถลงโปรแกรมลงในคอมพิวเตอร์ของบริษัทได้ การลงโปรแกรม Visual Studio 2015 นั้นไว้ใช้เขียนโค้ดในการพัฒนาแชทบอท
3. ขอใช้งานซอฟต์แวร์ UCMA เวอร์ชัน 4.0 ผ่านระบบ IT Services หรือ Software Request ของบริษัท โปรแกรม UCMA นั้นจะเป็นตัวที่ทำให้โปรแกรมแชทบอทที่พัฒนาขึ้นสามารถเชื่อมต่อกับโปรแกรม Skype for business ได้
4. ขอผู้ใช้งานแบบเป็น Services (Service IDs) บน Production และ Acceptance environment เป็นการขอชื่อผู้ใช้และรหัสผ่านให้กับแชทบอทไว้ใช้ในการเข้าถึงข้อมูลต่าง ๆ เช่นการเข้าถึงข้อมูลบนเว็บไซต์ของทีม เป็นต้น
5. ดาวน์โหลด SkypeX Framework SDK ซึ่งเป็นไฟล์ template ที่บริษัทได้พัฒนาโครงสร้างเริ่มต้นเอาไว้แล้ว เพื่อใช้ในการเริ่มต้นพัฒนาระบบแชทบอท โดยใน SkypeX Framework SDK นั้นจะมีโค้ดที่ช่วยในการเชื่อมต่อกับ Skype for business มาให้ด้วย
6. การเริ่มต้นพัฒนาแอปพลิเคชัน LUIS เพื่อใช้ในระบบแชทบอท แอปพลิเคชัน LUIS นั้น สามารถเริ่มต้นได้ด้วยตนเอง โดยการเข้าไปที่เว็บไซต์ <https://www.luis.ai/> จากนั้นล็อกอินด้วยไมโครซอฟท์แอคเคาท์ จากนั้นให้สร้างแอปพลิเคชัน LUIS ขึ้นมาโดยการกด Create new app ตามภาพที่ 1 และใส่ข้อมูลของแอปพลิเคชันตามภาพที่ 2 และกด Done

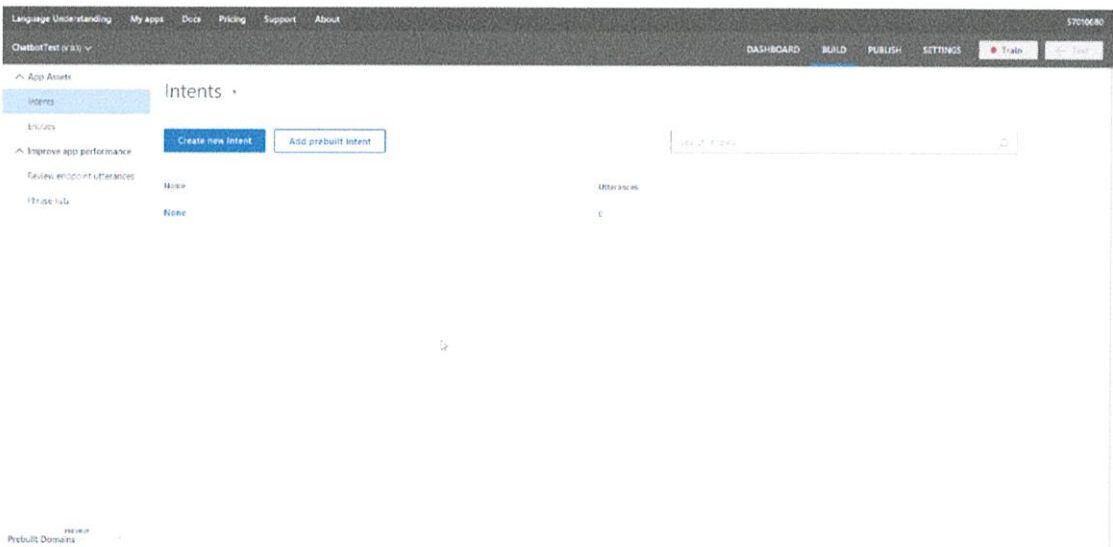


ภาพที่ 1 หน้า My Apps



ภาพที่ 2 หน้า Create new app

เมื่อสร้างเสร็จแล้วให้คลิกเข้าไปในแอปพลิเคชันที่สร้างขึ้น จากนั้นทำการเทรนข้อมูลโดยการคลิก Create new Intent ตามภาพที่ 3 ใส่ชื่ออินเทนต์และกด Done ตามภาพที่ 4



ภาพที่ 3 หน้า Intents

Create new intent

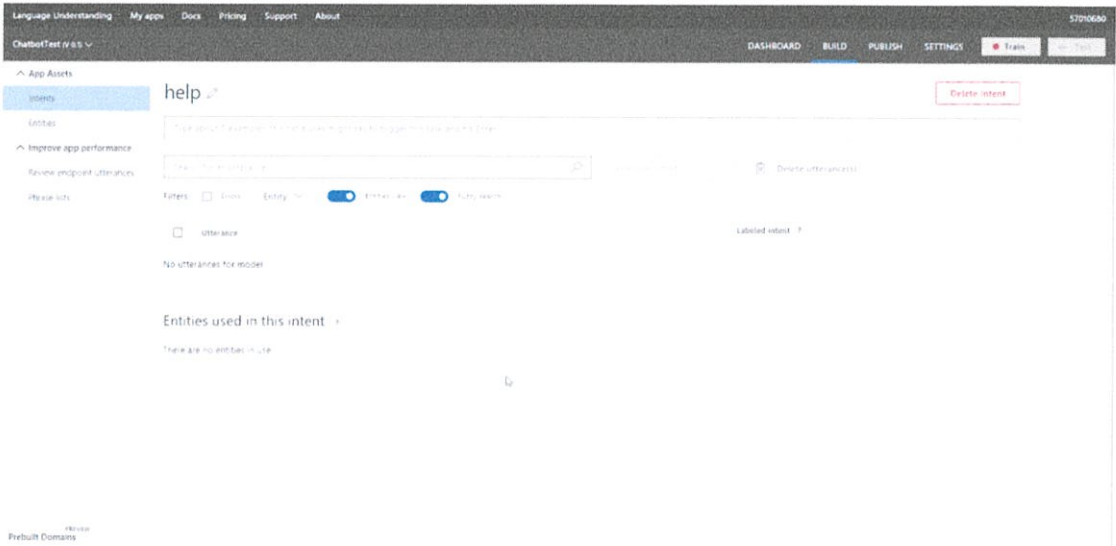
Intent name

help

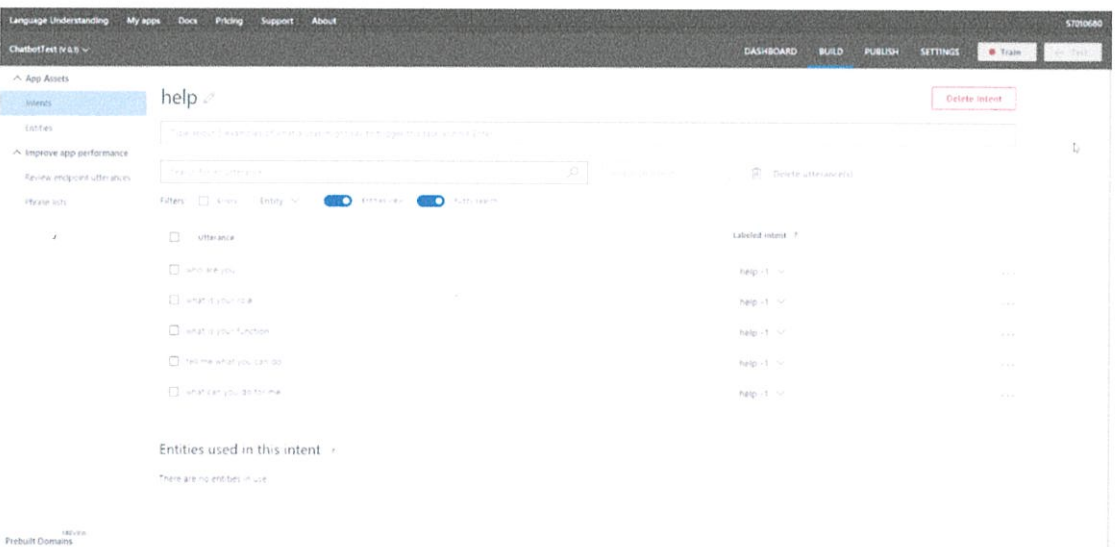
Cancel Done

ภาพที่ 4 หน้า Create new intent

จากนั้นใส่ข้อมูลในหน้า intent ที่สร้างขึ้นตามภาพที่ 5 ซึ่งเป็นข้อมูลประโยคตัวอย่างและกด Enter เมื่อกด Enter แล้วจะมีประโยคขึ้นมาตามภาพที่ 6

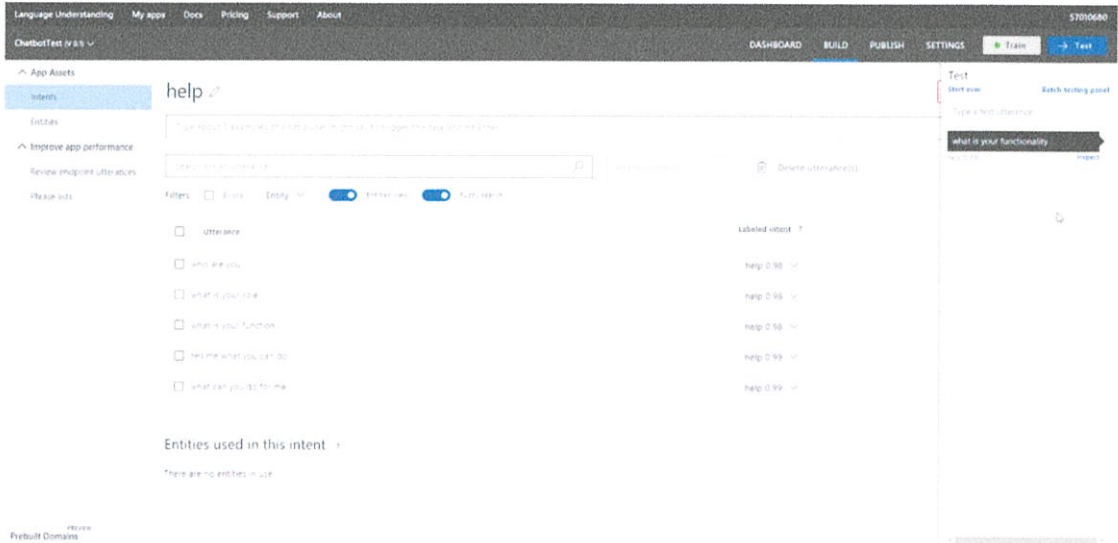


ภาพที่ 5 หน้า intent help



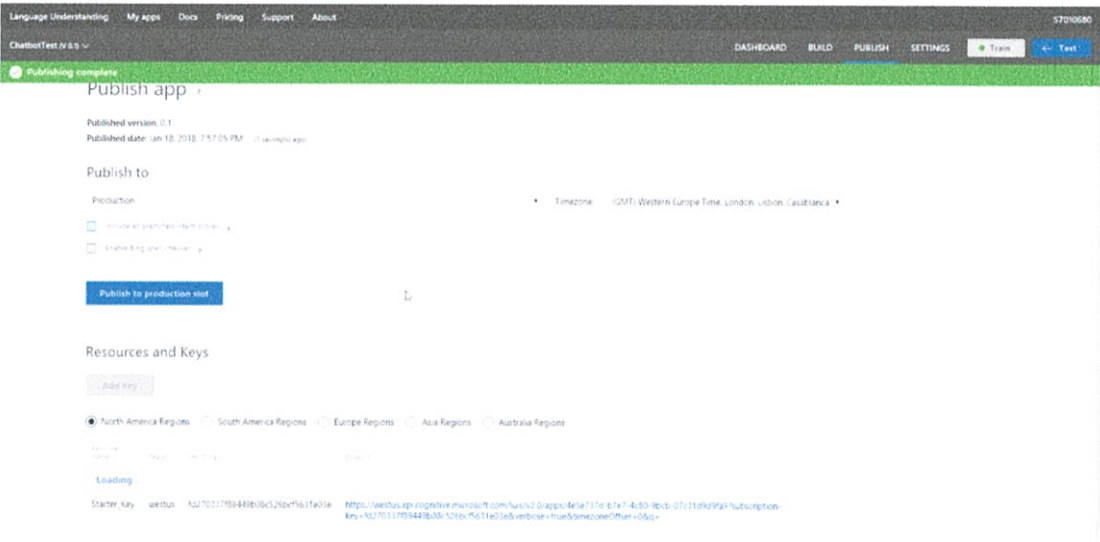
ภาพที่ 6 หน้า intent help เมื่อใส่ข้อมูลแล้ว

เมื่อใส่ข้อมูลที่ต้องการเทรนครบแล้วให้คลิก Train จากนั้นให้คลิก Test และลองใส่ประโยคตัวอย่างลงไป แอปพลิเคชัน LUIS จะแสดงอินเทนต์ที่ทำนายได้ออกมาตามภาพที่ 7



ภาพที่ 7 หน้า intent Test แอปพลิเคชัน LUIS

เมื่อเทรนและทดสอบเสร็จแล้วให้ทำการคลิก Publish แล้วคลิก Publish to production slot จากนั้นคัดลอก Endpoint URL และ Key String ไปไว้ในโปรแกรมแชทบอท ตามภาพที่ 8 เมื่อทำเสร็จแล้ว ระบบแชทบอทก็จะสามารถใช้งานแอปพลิเคชัน LUIS นี้ได้ หรือสามารถใช้แอปพลิเคชัน LUIS นี้กับโปรแกรมอื่น ๆ ได้โดยวิธีใช้นั้นเหมือนกับการใช้ API ทั่วไปเนื่องจากแอปพลิเคชัน LUIS นั้นให้ Response กลับมาเป็นแบบ JSON Object



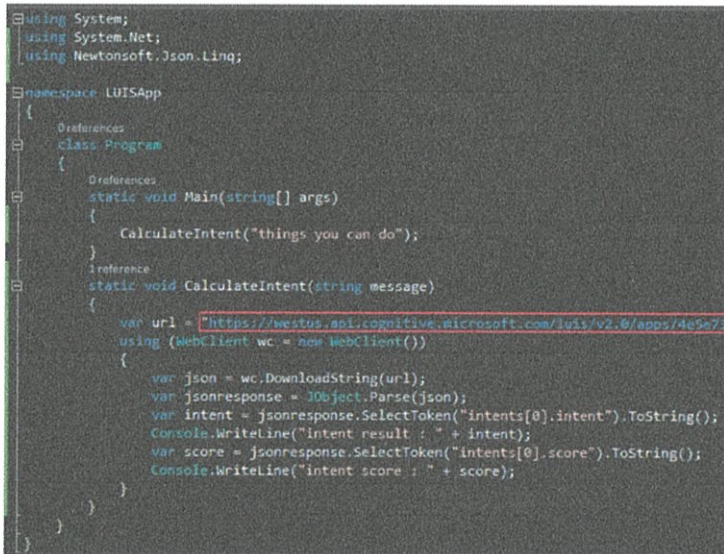
ภาพที่ 8 หน้า Publish app

ตัวอย่างการนำแอปพลิเคชัน LUIS ไปใช้งานในแอปพลิเคชันอื่น ๆ นั้นแสดงได้ตามตัวอย่างด้านล่างนี้ โดยในตัวอย่างนี้จะเป็นการเรียกใช้งานแอปพลิเคชัน LUIS ด้วยภาษา C#

การใช้งานนั้นให้นำ Endpoint URL มาใส่ในโปรแกรมที่ต้องการเรียกใช้แอปพลิเคชัน LUIS จากนั้นแปลง response ที่ได้เป็น JSON Object ตามภาพที่ 9 จากนั้นให้ดึงค่าที่ต้องการใช้จาก JSON Object ตามตัวอย่างการเขียนโปรแกรมในภาพที่ 10

```
{
  "query": "things you do",
  "topScoringIntent": {
    "intent": "help",
    "score": 0.5126804
  },
  "intents": [
    {
      "intent": "help",
      "score": 0.5126804
    },
    {
      "intent": "hello",
      "score": 0.188667387
    },
    {
      "intent": "None",
      "score": 0.0443957672
    }
  ],
  "entities": []
}
```

ภาพที่ 9 ตัวอย่าง JSON Object



```
using System;
using System.Net;
using Newtonsoft.Json.Linq;

namespace LUISApp
{
    class Program
    {
        static void Main(string[] args)
        {
            CalculateIntent("things you can do");
        }

        static void CalculateIntent(string message)
        {
            var url = "https://westus.api.cognitive.microsoft.com/luis/v2.0/apps/4e5e73...";
            using (WebClient wc = new WebClient())
            {
                var json = wc.DownloadString(url);
                var jsonresponse = JObject.Parse(json);
                var intent = jsonresponse.SelectToken("intents[0].intent").ToString();
                Console.WriteLine("intent result : " + intent);
                var score = jsonresponse.SelectToken("intents[0].score").ToString();
                Console.WriteLine("intent score : " + score);
            }
        }
    }
}
```

ภาพที่ 10 การเขียนโปรแกรม

เมื่อสั่งรันโปรแกรมในภาพที่ 10 แล้ว จะได้ผลลัพธ์เป็นอินเทนต์ที่มีคะแนนสูงสุดและคะแนนของอินเทนต์นั้นออกมา ตามภาพที่ 11

```
intent result : help
intent score : 0.823002636
```

ภาพที่ 11 ผลลัพธ์ของโปรแกรม

ภาคผนวก ข

การเริ่มต้นพัฒนาระบบการทดสอบโปรแกรมอัตโนมัติโดยใช้เทคโนโลยี Robotic Process Automation (RPA)

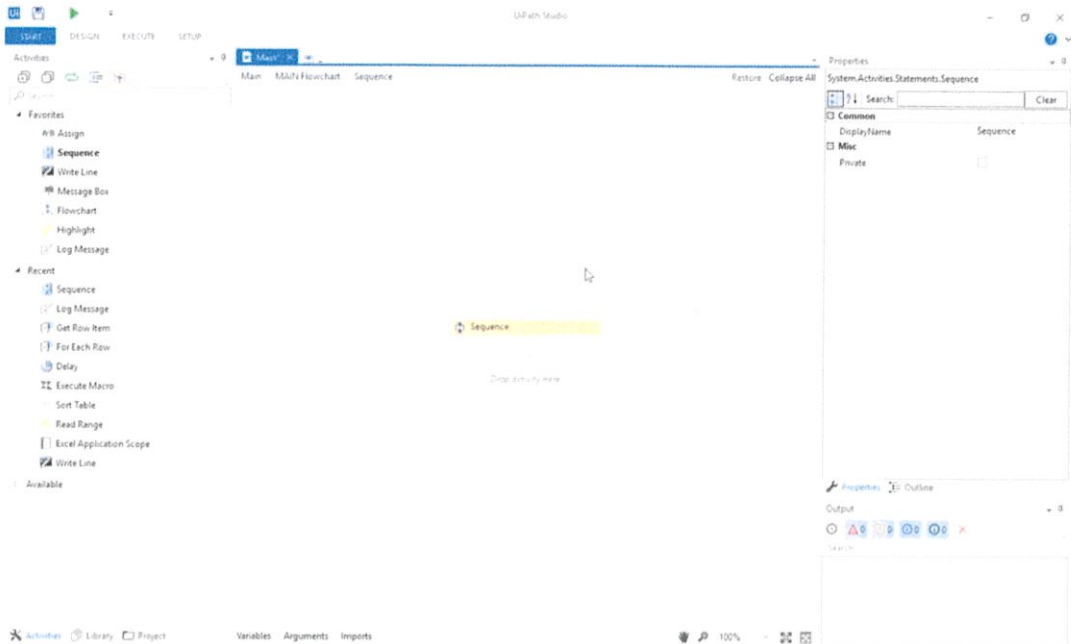
ขั้นตอนการเริ่มต้นพัฒนาระบบการทดสอบโปรแกรมอัตโนมัตินี้เป็นขั้นตอนที่ใช้ภายในบริษัทเท่านั้น สำหรับการพัฒนาที่ไม่ใช่ภายในบริษัทนั้นจะมีขั้นตอนที่แตกต่างออกไปจากนี้ โดยการพัฒนา ระบบการทดสอบโปรแกรมอัตโนมัติใช้โปรแกรม UiPath Studio ร่วมกับโปรแกรมที่ต้องการให้ RPA Script ไปทำงานอัตโนมัติให้ในที่นี้คือโปรแกรม SAP

1. ขอใช้งานโปรแกรม UiPath ผ่านระบบ IT Services หรือ Software Request ของบริษัท โดยขอเป็น trial version เท่านั้น โดยโปรแกรม UiPath เป็นโปรแกรมที่ใช้ในการสร้างระบบอัตโนมัติด้วยการใช้การลากและวาง activity ต่าง ๆ เป็นเหมือน Flowchart เลียนแบบวิธีที่คนใช้แอปพลิเคชันนั้น ๆ โดยสามารถใช้งานได้กับหลากหลายแอปพลิเคชัน เช่น เว็บเบราว์เซอร์ โปรแกรม Excel หรือ โปรแกรม SAP เป็นต้น
2. ติดตั้งโปรแกรม UiPath และ Activate โปรแกรมเป็นแบบ trial version โดยโปรแกรม UiPath แบบ trail version นั้นสามารถใช้งานได้ 3 เดือนเท่านั้น หลังจากนั้นถ้าต้องการใช้งานต่อจะต้องเสียค่าใบอนุญาตที่ใช้พัฒนา
3. ติดตั้ง library อื่น ๆ ที่ต้องการใช้งานเพิ่มเติมในโปรแกรม UiPath เช่น library ที่ใช้ร่วมกับโปรแกรม Microsoft Excel หรือ library ที่ใช้ส่งอีเมลล์ เป็นต้น
4. สามารถเริ่มพัฒนาโปรแกรมอัตโนมัติได้เลย หรือสามารถเริ่มศึกษาการใช้งานเบื้องต้นได้จากเว็บไซต์ <https://academy.uipath.com/> จะมีการสอนการเริ่มใช้งานเบื้องต้น เพื่อให้สามารถนำไปพัฒนาเป็นโปรแกรมอัตโนมัติอื่น ๆ ต่อไปได้

ตัวอย่างการใช้โปรแกรม UiPath Studio พัฒนาระบบการทดสอบโปรแกรมอัตโนมัติ

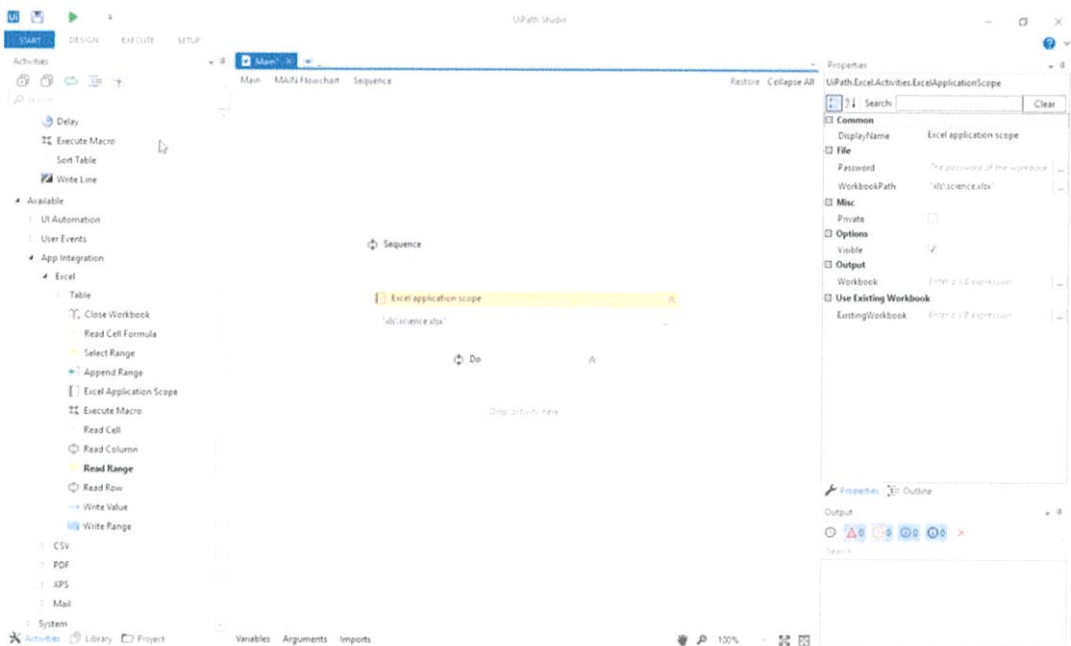
ตัวอย่างนี้จะเป็นการคัดลอกข้อมูลจากไฟล์เอ็กเซลหนึ่งและเขียนข้อมูลนั้นลงในไฟล์เอ็กเซลอีกไฟล์โดยอัตโนมัติ โดยการใช้โปรแกรม UiPath Studio ร่วมกับโปรแกรม Microsoft Excel

1. เมื่อเปิดโปรแกรม UiPath Studio ขึ้นมาแล้วจะเจอกับไฟล์ Main.xaml เป็นไฟล์แรก ให้ลาก activity ชื่อ Sequence จากแถบ Activities มาใส่ในไฟล์ Main.xaml ตามภาพที่ 1



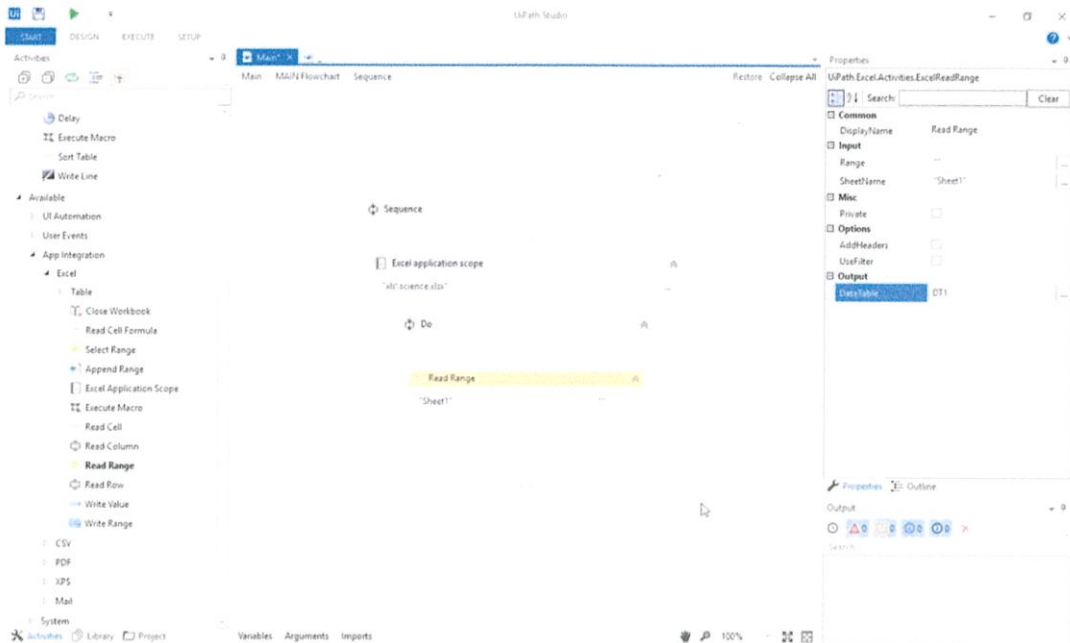
ภาพที่ 1 activity ชื่อ Sequence

2. ต่อไปให้ลาก activity ชื่อ Excel application scope จากแถบ Activities มาใส่ใน activity ชื่อ Sequence ตามภาพที่ 2 ใส่ชื่อของไฟล์เอ็กเซลที่ต้องการให้คัดลอกข้อมูล



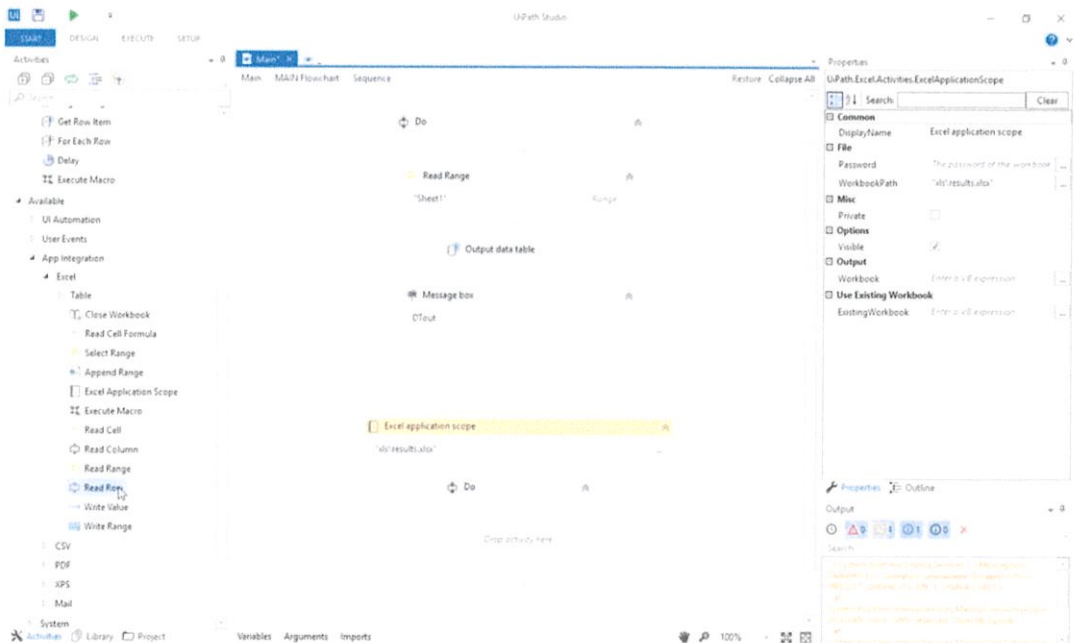
ภาพที่ 2 activity ชื่อ Excel application scope

3. ต่อกำหนด activity ชื่อ Read Range จากแถบ Activities มาใส่ใน activity Do ใน activity Excel application scope ตามภาพที่ 3 กำหนดตัวแปรที่ใช้สำหรับข้อมูลที่คัดลอกออกมา เก็บเป็นตัวแปรชนิด DataTable และเลือกช่วงที่ต้องการคัดลอกข้อมูลในช่อง Range ที่ส่วน Input



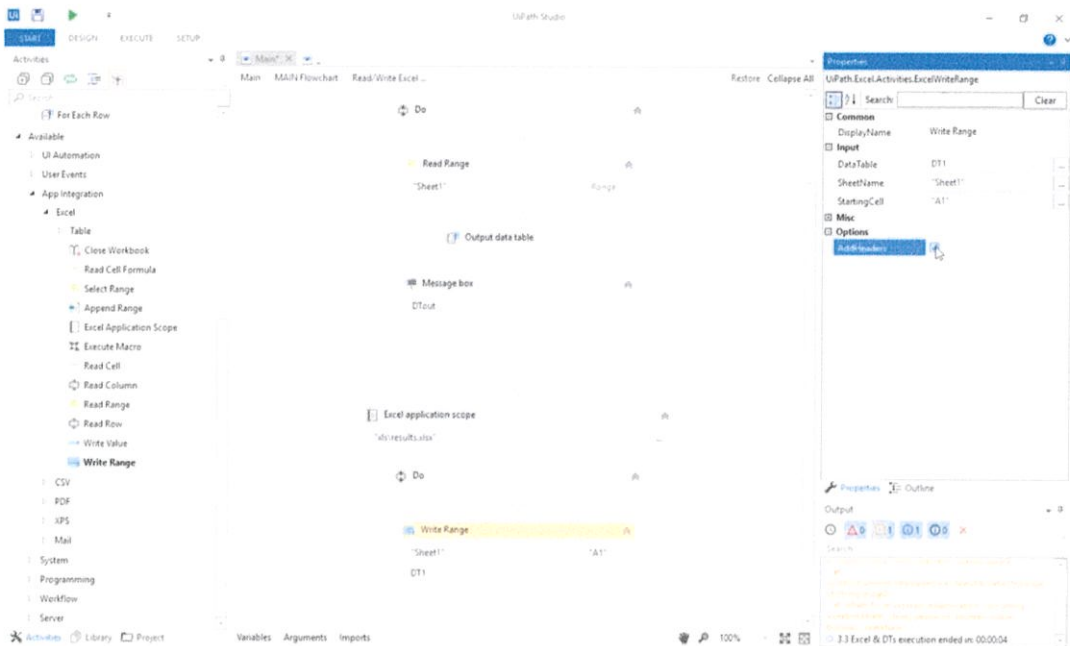
ภาพที่ 3 activity ชื่อ Read Range

4. ต่อกำหนด activity ชื่อ Excel application scope จากแถบ Activities มาใส่ใน activity ชื่อ Sequence ตามภาพที่ 4 ใส่ชื่อของไฟล์เอ็กเซลที่ต้องการให้ไปวางข้อมูลที่คัดลอกไว้ในขั้นตอนที่ 3



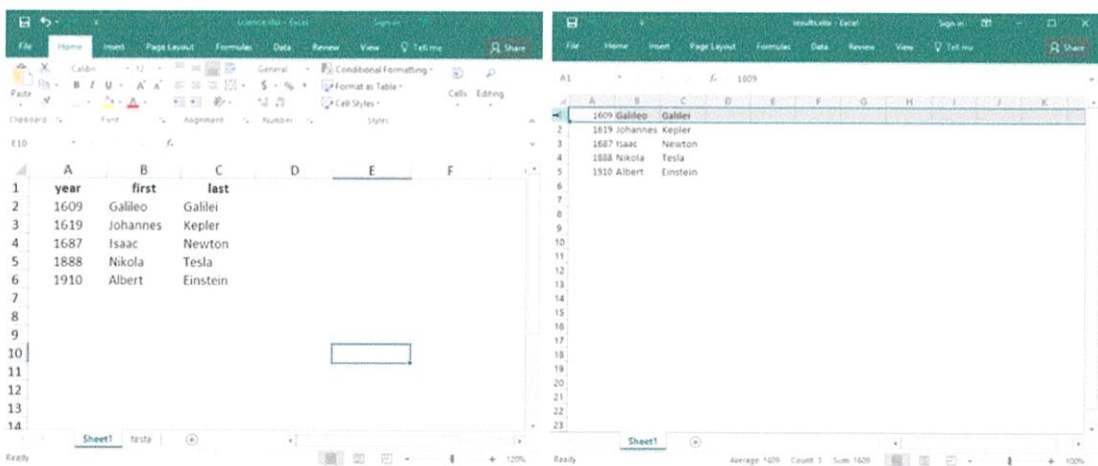
ภาพที่ 4 activity ชื่อ Excel application scope

5. ต่อไปให้ลาก activity ชื่อ Write Range จากแถบ Activities มาใส่ใน activity Do ใน activity Excel application scope ตามภาพที่ 5 จากนั้นใส่ชื่อตัวแปร DataTable ที่สร้างไว้ในขั้นตอนที่ 3 ในช่อง DataTable ในส่วนที่เป็น Input เมื่อเสร็จแล้วให้กดรันหรือ F5 เพื่อรันโปรแกรม



ภาพที่ 5 activity ชื่อ Write Range

6. เมื่อรันโปรแกรม UiPath แล้ว โปรแกรมจะทำงานตามที่เขียนสคริปไว้คือเริ่มจากเปิดไฟล์เอ็กเซลไฟล์แรก คัดลอกข้อมูล สร้างไฟล์เอ็กเซลใหม่แล้ววางข้อมูลที่คัดลอกไว้ เมื่อโปรแกรมทำงานเสร็จแล้วก็จะได้ไฟล์เอ็กเซลที่มีข้อมูลเหมือนไฟล์แรกตามภาพที่ 6



ภาพที่ 6 เปรียบเทียบไฟล์เอ็กเซลที่ทำการคัดลอกข้อมูล