



รายงานสหกิจศึกษาฉบับสมบูรณ์

การพัฒนาระบบอัตโนมัติสำหรับแชทบอตและล็อกเกอร์อัจฉริยะ

Automated System Development for Chatbot and Smart Locker

นายธนิชรุ์ โฆษิตศรีคุณากร

สาขาวิชาวิศวกรรมสารสนเทศ

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561



รายงานสหกิจศึกษาฉบับสมบูรณ์

การพัฒนาระบบอัตโนมัติสำหรับแชทบอตและล็อกเกอร์อัจฉริยะ

Automated System Development for Chatbot and Smart Locker

นายธนิชฐ์ โฆษิตศรีคุณากร

สาขาวิชาวิศวกรรมสารสนเทศ

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจศึกษา การพัฒนาระบบอัตโนมัติสำหรับแชตบอตและลือกเกอร์อัจฉริยะ

ชื่อ-สกุล นักศึกษา นายธนินธุ์ โฆษิตศรีคุณากร

คณะ วิศวกรรมศาสตร์ ภาควิชา วิศวกรรมคอมพิวเตอร์ สาขาวิชา วิศวกรรมสารสนเทศ

ชื่อ-สกุล อาจารย์นิเทศ รศ.ดร.ชวลิต เบญจางคประเสริฐ

ชื่อ-สกุล ผู้นิเทศงาน นายนันทฤกษ์ สมัญญารณ

ชื่อสถานประกอบการ บริษัท เอ็กซอนโมบิล จำกัด

บทคัดย่อ

ระบบอัตโนมัติเป็นสิ่งหนึ่งที่มีบทบาทสำคัญในการสร้างกระบวนการการทำงานที่ดีเป็นอย่างมาก เนื่องจากระบบอัตโนมัติช่วยเพิ่มประสิทธิภาพและความสะดวกสบายในการทำงานและการให้บริการ ลดข้อผิดพลาดในการทำงาน ทำงานได้รวดเร็วและทำงานได้ตลอดเวลา ในโครงการฉบับนี้ผู้พัฒนาได้พัฒนาต่อยอดระบบอัตโนมัติสำหรับโปรแกรมแชตบอตบนแพลตฟอร์มของ Skype for Business ของแผนกการดูแลและจัดการฐานข้อมูล เพื่อปรับปรุงและพัฒนาการทำงานของโปรแกรมให้สอดคล้องกับความต้องการทางธุรกิจในปัจจุบันของหน่วยงาน และพัฒนาระบบอัตโนมัติสำหรับลือกเกอร์อัจฉริยะ ที่เป็นระบบอัตโนมัติที่จะมาทดแทนระบบการยืมคืนอุปกรณ์ภายในบริษัทที่ยังอาศัยมนุษย์เป็นหลัก โดยจากผลการพัฒนาพบว่า ระบบอัตโนมัติสำหรับแชตบอตมีประสิทธิภาพการทำงานที่ดี และประสิทธิผลของการทำงานตอบสนองต่อความต้องการของหน่วยงานได้ และระบบอัตโนมัติสำหรับลือกเกอร์อัจฉริยะสามารถพิสูจน์การทำงานทดแทนการทำงานของมนุษย์ได้

คำสำคัญ : ระบบอัตโนมัติ แชตบอต ลือกเกอร์อัจฉริยะ การเพิ่มประสิทธิภาพ ความต้องการทางธุรกิจ

Cooperative Title: Automated System Development for Chatbot and Smart Locker

Student Intern Name: Tanit Kositsrikunakorn

Faculty: Engineering **Department:** Computer Engineering

Advisor Name: Assoc.Prof.Dr. Chawalit Benjangkaprasert

Mentor Name: Nantaruek Samanyaporn

Company: ExxonMobil Limited

ABSTRACT

Automation is one of the key factors of generating great outcomes out of any systems. It improves overall performance, reduces errors, and increases speed. This paper will show how the developer successfully made performance improvements by utilizing automation processes to a Chatbot program, which had been developed by Database Administration and Management (DBAM) function inside ExxonMobil Limited, including how the developer was able to identify and develop the Chatbot to fit the business needs of the function. This paper will also show how a new automated equipment loaning system called Smart Locker was developed through the process of problem identification and analysis, design and planning, product development, and testing. The result of the developments shows that the Chatbot performs well and fits the business needs and the Smart Locker prove the concept of an autonomous loaning system that can significantly reduce the number of workforces.

Keywords: Automation, Chatbot, Smart Locker, Performance Improvement, Business Needs

กิตติกรรมประกาศ

ข้าพเจ้าได้มีโอกาสเข้าร่วมโครงการสหกิจศึกษาที่ บริษัท เอ็กซอนโมบิล จำกัด ตั้งแต่วันที่ 2 กรกฎาคม 2560 ถึงวันที่ 28 ธันวาคม 2560 โดยได้รับฝึกหัดและปฏิบัติหน้าที่ในแผนกการดูแลและจัดการฐานข้อมูล (Database Administration and Management) ในหัวข้อโครงการการพัฒนาเว็บไซต์อัตโนมัติสำหรับแชทบอตและล็อกเกอร์อัจฉริยะ ซึ่งข้าพเจ้าได้รับความรู้และประสบการณ์ในการทำงานที่เป็นประโยชน์ต่อตัวข้าพเจ้าอย่างมาก อีกทั้งยังได้รับประสบการณ์ที่ดีจากกิจกรรมต่าง ๆ ที่บริษัทได้จัดให้ตลอดระยะเวลาการทำงาน ข้าพเจ้าขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ บริษัท เอ็กซอนโมบิล จำกัด และพนักงานที่เกี่ยวข้องที่ให้โอกาสข้าพเจ้าได้ร่วมเข้าทำงานในบริษัท ตลอดจนการดูแลและให้ความช่วยเหลือต่าง ๆ แก่ข้าพเจ้าตลอดระยะเวลาโครงการสหกิจศึกษาในครั้งนี้

ขอขอบพระคุณ คุณนันทฤกษ์ สมัญญามร คุณสุทธิพันธ์ รุจิรพิพัฒน์ และคุณสันต์ศนีย์ ทรัพย์ศิริ ที่เป็นผู้ดูแลข้าพเจ้า คอยให้คำปรึกษา และคำแนะนำไม่ว่าจะเป็นด้านการทำงานหรือการใช้ชีวิตภายในบริษัท ข้าพเจ้านำสิ่งที่ข้าพเจ้าได้รับไปปรับใช้ในการทำงานและการใช้ชีวิตของข้าพเจ้าในอนาคตต่อไป

ขอขอบพระคุณพนักงานแผนกการดูแลและจัดการฐานข้อมูลทุกท่าน ที่คอยช่วยเหลือข้าพเจ้าในเรื่องต่าง ๆ ตลอดจนดูแล ให้ความรู้ และให้คำแนะนำเสมอมา

ขอขอบพระคุณ รศ.ดร.ชวลิต เบญจางคประเสริฐ อาจารย์ที่ปรึกษาที่ให้คำปรึกษาและคำแนะนำแก่ข้าพเจ้า ทำให้การฝึกงานโครงการสหกิจศึกษา และรายงานฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณบิดา มารดา และครอบครัวของข้าพเจ้า ที่สนับสนุนเต็มกำลัง และเป็นกำลังใจที่ดีให้แก่ข้าพเจ้าอยู่ตลอดเวลา

ขอบคุณเพื่อน ๆ นักศึกษาร่วมงานของข้าพเจ้าทุกคน ที่เป็นกำลังใจให้กันและกันเสมอมา และทำให้บรรยากาศการฝึกงานครั้งนี้มีแต่ความสนุกสนาน

สุดท้ายนี้ ข้าพเจ้าขอขอบพระคุณอาจารย์ทุกท่านของข้าพเจ้าที่ได้ประสิทธิ์ประสาทวิชาความรู้ให้แก่ข้าพเจ้า และขอขอบพระคุณ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สำหรับโอกาสในการทำโครงการสหกิจศึกษาครั้งนี้

ธนัชฐ์ โฆษิตศรีคุณากร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย)	I
บทคัดย่อ (ภาษาอังกฤษ)	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์ของการพัฒนา	2
1.3 ขอบเขตของการพัฒนา	2
1.3.1 ระบบอัตโนมัติสำหรับแชทบอต	2
1.3.2 ระบบอัตโนมัติสำหรับบล็อกเกอร์อัจฉริยะ	2
1.4 วิธีการดำเนินการพัฒนา	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
1.5.1 ประโยชน์ต่อบริษัท	3
1.5.2 ประโยชน์ต่อผู้พัฒนา	3
บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 Chatbot	4
2.2 Skype for Business	5
2.3 Microsoft Unified Communications Managed API 5.0	6
2.4 API	7
2.5 .Net Framework	8
2.6 โปรแกรม Microsoft Visual Studio	9
2.7 ระบบปฏิบัติการ Windows Server	10
2.8 โปรแกรม Remote Desktop Connection	10
2.9 โปรแกรม Microsoft SQL Server Management Studio	11
2.10 ภาษา C#	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.11 ภาษา XML	12
2.12 รูปแบบข้อมูลประเภท JSON	13
2.13 ภาษา SQL	14
2.14 DDR5	15
2.15 Microsoft Team Foundation Server.....	15
2.16 ระบบ ITSM Remedy.....	16
2.17 Microsoft Azure	16
2.18 Raspberry Pi	17
2.19 ภาษา Python	17
บทที่ 3 ขั้นตอนการดำเนินงานพัฒนา	19
3.1 ขั้นตอนการดำเนินงานพัฒนาระบบอัตโนมัติสำหรับแชทบอต.....	19
3.1.1 ศึกษาการทำงานของระบบ	19
3.1.2 ศึกษาและรวบรวมความต้องการปัจจุบัน	24
3.1.3 วางแผนการพัฒนาระบบ.....	26
3.1.4 พัฒนาระบบ	26
3.1.5 ทดสอบการทำงานของระบบ	50
3.1.6 นำระบบไปใช้งานจริง.....	50
3.2 ขั้นตอนการดำเนินงานพัฒนาระบบอัตโนมัติสำหรับล็อกเกอร์อัจฉริยะ	51
3.2.1 ศึกษาความต้องการต่าง ๆ	52
3.2.2 วางแผนการพัฒนาระบบ.....	52
3.2.3 พัฒนาระบบ	54
3.2.4 ทดสอบการทำงานของระบบ	75
บทที่ 4 ผลการพัฒนา.....	76
4.1 ผลการพัฒนาในระบบอัตโนมัติสำหรับแชทบอต.....	76
4.2 ผลการพัฒนาในระบบอัตโนมัติสำหรับล็อกเกอร์อัจฉริยะ.....	80
บทที่ 5 สรุปผลการพัฒนาและข้อเสนอแนะ	81
5.1 สรุปการพัฒนา.....	81
5.2 ปัญหาที่พบ และวิธีการแก้ไข.....	81

เอกสารอ้างอิง 83



สารบัญตาราง

	หน้า
ตารางที่ 3.1 แผนการทำงานพัฒนาระบบแช็ตบอต	19
ตารางที่ 3.2 การแบ่งภาระงานที่รับผิดชอบของงานแช็ตบอต.....	26
ตารางที่ 3.3 แผนการทำงานพัฒนาระบบล็อกเกอร์อัจฉริยะในระยะที่ 1	52
ตารางที่ 3.4 การแบ่งภาระงานที่รับผิดชอบของงานล็อกเกอร์อัจฉริยะ	54



สารบัญรูป

	หน้า
รูปที่ 2.1 การสนทนาระหว่างมนุษย์และแช็ตบอต	4
รูปที่ 2.2 Skype for Business logo	5
รูปที่ 2.3 หน้าต่างหลักของโปรแกรม Skype for Business.....	5
รูปที่ 2.4 หน้าต่างสนทนาของโปรแกรม Skype for Business	6
รูปที่ 2.5 ภาพรวมการทำงานของ Microsoft UCMA 5.0.....	7
รูปที่ 2.6 ภาพรวมการใช้งาน API.....	8
รูปที่ 2.7 โครงสร้างทางสถาปัตยกรรมของ .Net Framework 4	8
รูปที่ 2.8 Microsoft Visual Studio logo	9
รูปที่ 2.9 หน้าต่างโปรแกรม Microsoft Visual Studio.....	9
รูปที่ 2.10 ระบบปฏิบัติการ Windows Server.....	10
รูปที่ 2.11 หน้าต่างการล็อกอินเพื่อเข้าควบคุมเครื่องคอมพิวเตอร์ระยะไกลของโปรแกรม Remote Desktop Connection	11
รูปที่ 2.12 หน้าต่างของโปรแกรม Microsoft SQL Server Management Studio.....	11
รูปที่ 2.13 C# logo	12
รูปที่ 2.14 ตัวอย่าง code ภาษา XML	13
รูปที่ 2.15 ตัวอย่างการเก็บข้อมูลในรูปแบบของ JSON	14
รูปที่ 2.16 ตัวอย่างโครงสร้างภาษา SQL	14
รูปที่ 2.17 Microsoft Team Foundation Server	15
รูปที่ 2.18 หน้าตาเว็บไซต์ Microsoft Azure Portal.....	16
รูปที่ 2.19 Raspberry Pi 3 Model B.....	17
รูปที่ 2.20 Python logo	18
รูปที่ 3.1 ภาพรวมโครงสร้างของระบบแช็ตบอต	20
รูปที่ 3.2 Code ภาษา C# ของ class Program.....	21
รูปที่ 3.3 Code ภาษา C# ของ class DatabaseBot.....	22
รูปที่ 3.4 Code ภาษา XML ของไฟล์ App.config.....	22
รูปที่ 3.5 การเรียกใช้งาน method calculateIntent() เพื่อหา Intent ของข้อความจากผู้ใช้งาน.....	23
รูปที่ 3.6 การหา Intent โดยอาศัยการเปรียบเทียบกับคำสำคัญ.....	23
รูปที่ 3.7 ภาพรวมการทำงานภายในของโปรแกรมแช็ตบอต.....	24

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 3.8 หน้าต่าง Team Explorer	27
รูปที่ 3.9 ปุ่ม Manage Connection ในหน้าต่าง Team Explorer	27
รูปที่ 3.10 หน้าต่าง Add Team Foundation Server	28
รูปที่ 3.11 ปุ่ม Source Control Explorer ภายในหน้าต่าง Team Explorer.....	28
รูปที่ 3.12 ตัวแปรที่สำคัญสำหรับการทำงานของเซิร์ฟเวอร์ในไฟล์ App.config	29
รูปที่ 3.13 การแปลง object จาก C# class format เป็น JSON format.....	30
รูปที่ 3.14 การทำงานของ code เชื่อมต่อกับระบบอื่น ๆ ผ่าน HTTP Request.....	31
รูปที่ 3.15 การเรียก method getUserData()	32
รูปที่ 3.16 การเก็บข้อมูลที่ได้จาก DBAM Automation API ลงในตัวแปร currentSession.....	32
รูปที่ 3.17 แผนผังรวมการทำงานของคำสั่ง Access Review	33
รูปที่ 3.18 แผนผังรวมการทำงานของคำสั่ง Access Review (2).....	34
รูปที่ 3.19 โปรแกรม Microsoft SQL Server Management Studio เพื่อการจัดการฐานข้อมูล SQL..	35
รูปที่ 3.20 แผนผังรวมการทำงานของคำสั่ง Backup Database	36
รูปที่ 3.21 แผนผังรวมการทำงานของคำสั่ง Backup Database (2).....	37
รูปที่ 3.22 แผนผังรวมการทำงานของคำสั่ง Restore Database.....	39
รูปที่ 3.23 แผนผังรวมการทำงานของคำสั่ง Restore Database (2).....	40
รูปที่ 3.24 แผนผังรวมการทำงานของคำสั่ง Restore Database (3).....	41
รูปที่ 3.25 แผนผังรวมการทำงานของคำสั่ง Restore Database (4).....	42
รูปที่ 3.26 แผนผังรวมการทำงานของคำสั่ง Refresh Database	44
รูปที่ 3.27 แผนผังรวมการทำงานของคำสั่ง Refresh Database (2)	45
รูปที่ 3.28 แผนผังรวมการทำงานของคำสั่ง Refresh Database (3)	46
รูปที่ 3.29 แผนผังรวมการทำงานของคำสั่ง Refresh Database (4).....	47
รูปที่ 3.30 แผนผังรวมการทำงานของคำสั่ง Refresh Database (5).....	48
รูปที่ 3.31 แผนผังรวมการทำงานของคำสั่ง Refresh Database (6).....	49
รูปที่ 3.32 การทดสอบการทำงานของคำสั่ง Access Review.....	50
รูปที่ 3.33 ตัวอย่างเอกสารข้อมูลเกี่ยวกับการพัฒนาเซิร์ฟเวอร์ DBAM Bot.....	51
รูปที่ 3.34 ภาพรวมการทำงานของระบบล็อกเกอร์อัจฉริยะ	53
รูปที่ 3.35 รูปแบบมาตรฐานของการตอบกลับของ Smart Locker API.....	54

สารบัญรูป (ต่อ)

หน้า

รูปที่ 3.36	คำอธิบาย Status และ StatusDescription ของการตอบกลับของ Smart Locker API	55
รูปที่ 3.37	รูปแบบการตอบกลับของ Get Categories endpoints	56
รูปที่ 3.38	JSON ที่ใช้เพื่อเรียกใช้ Add New Category endpoint	56
รูปที่ 3.39	รูปแบบการตอบกลับของ Get Subcategories endpoints	57
รูปที่ 3.40	JSON ที่ใช้เพื่อเรียกใช้ Add New Subcategory endpoint	58
รูปที่ 3.41	รูปแบบการตอบกลับของ Get Item endpoints	59
รูปที่ 3.42	รูปแบบการตอบกลับของ Get Item Details endpoints	60
รูปที่ 3.43	JSON ที่ใช้เพื่อเรียกใช้ Add New Item endpoint	61
รูปที่ 3.44	รูปแบบการตอบกลับของ Get User's Watchlist endpoint	61
รูปที่ 3.45	JSON ที่ใช้เพื่อเรียกใช้ Notify/unnotify endpoint	62
รูปที่ 3.46	รูปแบบการตอบกลับของ Get User's Loaned Object endpoint	63
รูปที่ 3.47	รูปแบบการตอบกลับของ Get Loaned Item Details endpoint	64
รูปที่ 3.48	รูปแบบการตอบกลับของ Get Country endpoint	65
รูปที่ 3.49	JSON ที่ใช้เพื่อเรียกใช้ Add New Country endpoint	65
รูปที่ 3.50	รูปแบบการตอบกลับของ Get Site endpoints	66
รูปที่ 3.51	JSON ที่ใช้เพื่อเรียกใช้ Add New Site endpoint	66
รูปที่ 3.52	รูปแบบการตอบกลับของ Check if User Exists endpoint	67
รูปที่ 3.53	JSON ที่ใช้เพื่อเรียกใช้ Create New User endpoint	68
รูปที่ 3.54	JSON ที่ใช้เพื่อเรียกใช้ Update User's Details endpoint	68
รูปที่ 3.55	ตัวอย่าง code สำหรับจำลองข้อมูลลงฐานข้อมูลของระบบ Smart Locker	69
รูปที่ 3.56	หน้าต่าง Account Settings ของโปรแกรม Visual Studio	70
รูปที่ 3.57	หน้าต่าง Sign in to your account ของโปรแกรม Visual Studio	70
รูปที่ 3.58	หน้าต่าง Publish ในโปรแกรม Visual Studio	71
รูปที่ 3.59	หน้าต่าง Create New Connection ของโปรแกรม Visual Studio	72
รูปที่ 3.60	หน้าต่างการทำงานระหว่างการ Publish	72
รูปที่ 3.61	หน้าต่างเว็บเบราว์เซอร์หน้าเว็บหลักของ API บนเว็บไซต์ของ Azure	73
รูปที่ 3.62	หน้าเข้าสู่ระบบ Query Editor ของ Azure SQL Database	74
รูปที่ 3.63	ตัวอย่างการใช้คำสั่งภาษา SQL ผ่าน Query Editor บน Azure	74

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 3.64 ผลการทดสอบการใช้งาน API โดยการเข้าถึง URL โดยตรงผ่านเว็บเบราว์เซอร์.....	75
รูปที่ 3.65 ผลการทดสอบการใช้งาน API ผ่านหน้าเว็บไซต์ที่สร้างขึ้น	75
รูปที่ 4.1 คำทักทายเมื่อเริ่มต้นสนทนาของแชทบอต.....	76
รูปที่ 4.2 ผลลัพธ์การทำงานคำสั่ง Access Review	77
รูปที่ 4.3 แชทบอตรับรู้ข้อมูลของคุณสนทนาได้	77
รูปที่ 4.4 ผลลัพธ์การทำงานคำสั่ง Backup Database	78
รูปที่ 4.5 แชทบอตเชื่อมต่อกับระบบ ITSM API เพื่อตรวจสอบ Ticket	78
รูปที่ 4.6 แชทบอตเชื่อมต่อกับระบบ ITSM API เพื่อสร้าง Ticket ขึ้นใหม่.....	79
รูปที่ 4.7 แชทบอตค้นหาและสร้างตารางข้อมูลให้ผู้ใช้เลือกเพื่อสร้างความสะดวกแก่ผู้ใช้งาน และ ตรวจสอบสิทธิ์การเข้าถึงให้โดยอัตโนมัติ.....	79
รูปที่ 4.8 ตัวอย่างผลลัพธ์จากการเรียกใช้งาน Get All Categories endpoints แสดงถึง category ทั้งหมดภายในระบบ.....	80
รูปที่ 4.9 ตัวอย่างผลลัพธ์จากการเรียกใช้งาน Get Filtered Categories endpoints โดยการกรอง category ภายในระบบโดยใช้ Site ID.....	80

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

บริษัท เอ็กซอนโมบิล จำกัด เป็นบริษัทในเครือเอ็กซอนโมบิล ซึ่งเป็นบริษัทชั้นนำด้านพลังงาน สัญชาติอเมริกัน ประกอบธุรกิจก๊าซธรรมชาติและน้ำมันปิโตรเลียมที่มีขนาดใหญ่ที่สุดในโลก โดยบริษัท เอ็กซอนโมบิล จำกัด ก่อตั้งขึ้นเพื่อดำเนินการศูนย์บริการธุรกิจกรุงเทพ (Bangkok Business Support Center: Bangkok BSC) ซึ่งเป็นศูนย์บริการธุรกิจที่ใหญ่ที่สุดในเครือข่ายศูนย์บริการธุรกิจทั่วโลกของ เอ็กซอนโมบิล ปัจจุบันมีพนักงานมากกว่า 2,200 คน มีสำนักงานอยู่ 3 แห่งในประเทศไทย ให้บริการแก่ บริษัทในเครือเอ็กซอนโมบิลและลูกค้าในกว่า 60 ประเทศทั่วโลก ครอบคลุมทั้งทวีปเอเชีย ทวีปยุโรป และ ทวีปอเมริกาเหนือ-ใต้ บริการของศูนย์ฯ ครอบคลุมงานด้านบัญชี เทคโนโลยีสารสนเทศ ศูนย์บริการลูกค้า สำหรับน้ำมันเชื้อเพลิงและน้ำมันหล่อลื่น ศูนย์บริการลูกค้าเคมีภัณฑ์ ศูนย์จัดซื้อ ศูนย์ทรัพยากรบุคคล ฝ่ายภาษี และเครดิตลูกค้า

การเข้าร่วมโครงการสหกิจศึกษารั้งนี้ ผู้พัฒนาได้เข้าร่วมงานในแผนกการดูแลและจัดการ ฐานข้อมูล (Database Administration and Management: DBAM) ที่มีหน้าที่ดูแลข้อมูลของบริษัทและ ควบคุมการใช้งานเครื่องเซิร์ฟเวอร์มากกว่า 2,000 เครื่อง ซึ่งในแต่ละวันจะมีผู้ใช้งาน (End user) เป็น จำนวนมาก โดยหากผู้ใช้งานได้รับปัญหาจากการใช้งานหรือมีข้อสงสัยที่เกี่ยวข้องกับฐานข้อมูล (Database) ผู้ใช้งานจะติดต่อมายังผู้ดูแลฐานข้อมูล (Database Administrator: DBA) เพื่อขอคำแนะนำ ซึ่งในบางครั้งผู้ดูแลฐานข้อมูลอาจต้องให้ความสำคัญกับงานที่เร่งด่วนกว่า หรือทำงานให้กับผู้ใช้งานท่านอื่น อยู่ อาจส่งผลให้ผู้ใช้งานบางท่านได้รับการตอบกลับที่ล่าช้าหรือไม่ครบถ้วน อีกทั้งการทำงานเดิมซ้ำ ๆ ใน ทุก ๆ วันถือเป็นการเพิ่มภาระงานให้กับผู้ดูแลฐานข้อมูลโดยไม่จำเป็น

จากปัญหาข้างต้น DBAM จึงได้พัฒนาระบบตอบคำถามอัตโนมัติสำหรับการจัดการฐานข้อมูล (Skype bot for Database Administration: DBAM Bot) ขึ้น เพื่อใช้สำหรับตอบคำถามต่าง ๆ จาก ผู้ใช้งานบนแพลตฟอร์มของ Skype for Business และเนื่องจากมีความต้องการที่เปลี่ยนไปในแต่ละปี ทำให้ต้องมีการปรับเปลี่ยนและปรับปรุงการทำงานของระบบให้ตอบรับกับความต้องการในปัจจุบันมากยิ่งขึ้น

นอกจาก DBAM Bot แล้ว ผู้พัฒนายังได้มีโอกาสเข้าร่วมโครงการ EMIT Bootcamp 2018 ของทางบริษัท ที่จัดขึ้นเพื่อให้พนักงานของบริษัทได้รวมกลุ่มและเสนอแนวความคิดใหม่ ๆ เพื่อผลักดันและพัฒนาบริษัทให้ก้าวไปข้างหน้า โดยกลุ่มของผู้พัฒนาได้สังเกตเห็นถึงปัญหาการยืมคืนอุปกรณ์ภายในบริษัท ที่ในปัจจุบันยังคงเป็นการยืมคืนอุปกรณ์ที่ต้องอาศัยเจ้าหน้าที่และไม่ได้มีการบันทึกรวมถึงติดตามสถานะ จึงทำให้ปัญหาอุปกรณ์เสียรวมถึงสูญหายเกิดขึ้นบ่อยครั้ง และเนื่องจากไม่มีการลงบันทึกจึงทำให้ทั้งผู้ดูแลและผู้ที่ต้องการยืมสิ่งของไม่สามารถทราบถึงจำนวนสิ่งของที่มีสถานะว่างและพร้อมให้ยืม ส่งผลให้สูญเสยทั้งเวลาและทรัพยากรแก่ทุกฝ่าย

จากปัญหาที่กล่าวมา ทางกลุ่มของผู้พัฒนาจึงได้คิดค้นและนำเสนอระบบล็อกเกอร์อัจฉริยะ (Smart Locker) ซึ่งจะเป็นการพัฒนากระบวนการยืมคืนที่สามารถทำได้ด้วยตัวเอง (self-service) รวมถึงเว็บไซต์เพื่อค้นหาและติดตามสถานะของอุปกรณ์ขึ้น เพื่อลดภาระงานของผู้ดูแล ยกกระดับประสบการณ์การใช้งานของผู้ใช้งาน และช่วยประหยัดค่าใช้จ่ายในการซ่อมบำรุงหรือซื้ออุปกรณ์ทดแทนให้แก่บริษัทได้

1.2 วัตถุประสงค์ของการพัฒนา

- 1.2.1 เพื่อลดภาระงานของผู้ดูแลระบบทั้งสองระบบ
- 1.2.2 เพื่อเพิ่มประสิทธิภาพในการทำงานโดยรวมและลดระยะเวลาในการทำงาน
- 1.2.3 เพื่อเพิ่มระดับความพึงพอใจของผู้ใช้งานระบบ

1.3 ขอบเขตของการพัฒนา

- 1.3.1 ระบบอัตโนมัติสำหรับแช็ตบอต
 - 1.3.1.1 แช็ตบอตสามารถเชื่อมต่อและทำงานเกี่ยวกับฐานข้อมูลได้
 - 1.3.1.2 แช็ตบอตสามารถระบุตัวตนผู้ใช้งานได้โดยอัตโนมัติ
 - 1.3.1.3 แช็ตบอตสามารถทำงานร่วมกับระบบอื่น ๆ ได้
- 1.3.2 ระบบอัตโนมัติสำหรับล็อกเกอร์อัจฉริยะ
 - 1.3.2.1 ระบบล็อกเกอร์สามารถเชื่อมต่อและทำงานเกี่ยวกับฐานข้อมูลได้
 - 1.3.2.2 ระบบล็อกเกอร์สามารถประมวลผลเพื่อให้ได้ข้อมูลที่ต้องการได้

1.4 วิธีการดำเนินการพัฒนา

- 1.4.1 รวบรวมความต้องการ
- 1.4.2 วางแผนและออกแบบการพัฒนาระบบ
- 1.4.3 พัฒนาระบบ
- 1.4.4 ทดสอบการทำงานของระบบ
- 1.4.5 นำระบบไปใช้งานจริง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่ผู้พัฒนาคาดว่าจะได้รับจากการเข้าร่วมโครงการสหกิจศึกษากับ บริษัท เอ็กซอนโมบิล จำกัด จำแนกออกเป็น 2 ส่วน คือ

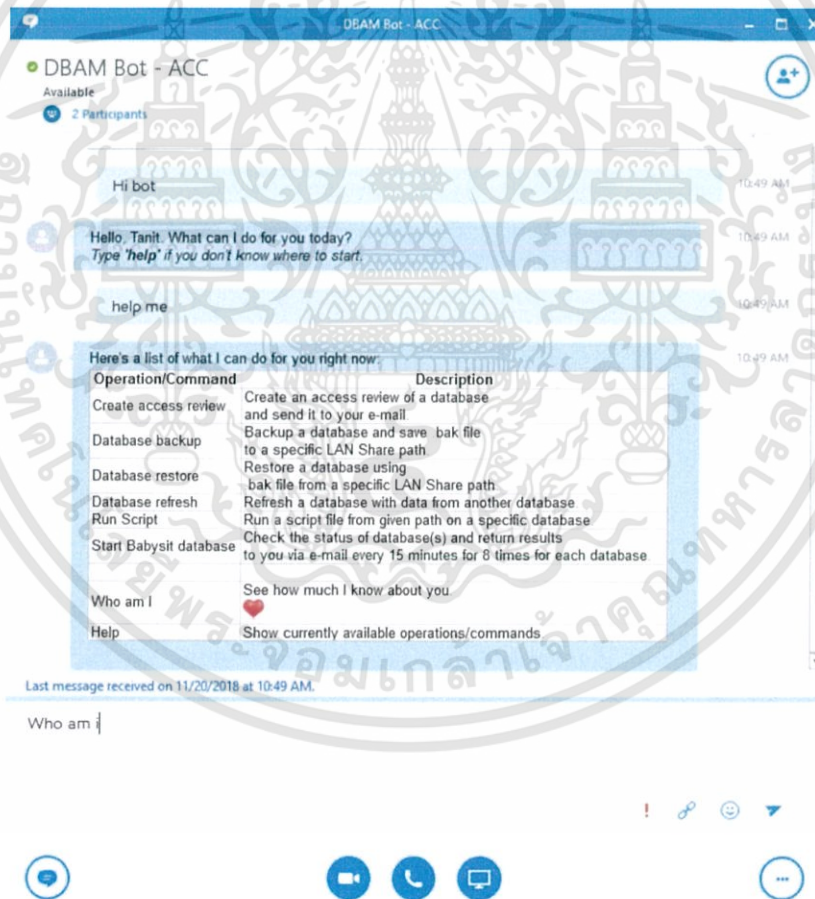
- 1.5.1 ประโยชน์ต่อบริษัท
 - 1.5.1.1 ลดการใช้ทรัพยากรบุคคล
 - 1.5.1.2 ลดค่าใช้จ่ายด้านอุปกรณ์ของบริษัท
 - 1.5.1.3 เพิ่มประสิทธิภาพการทำงานโดยรวม
- 1.5.2 ประโยชน์ต่อผู้พัฒนา
 - 1.5.2.1 ได้พัฒนาทักษะการสื่อสารและการทำงานร่วมกับผู้อื่น
 - 1.5.2.2 ได้พัฒนาทักษะการสื่อสารด้วยภาษาอังกฤษอย่างมีประสิทธิภาพ
 - 1.5.2.3 มีโอกาสได้ทำงานร่วมกับชาวต่างชาติ
 - 1.5.2.4 ได้เรียนรู้สภาพแวดล้อม และวัฒนธรรมภายในองค์กร
 - 1.5.2.5 สามารถคิด วิเคราะห์ วางแผน และทำงานได้อย่างเป็นระบบ

บทที่ 2

แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 Chatbot

Chatbot (แช็ตบอต) คือโปรแกรมคอมพิวเตอร์ชนิดหนึ่ง ที่สร้างขึ้นเพื่อใช้สื่อสารโดยการสนทนากับมนุษย์โดยจำลองการตอบโต้บทสนทนาให้คล้ายคลึงกับการตอบโต้ของมนุษย์ โดย Chatbot นั้นได้มีการพัฒนาขึ้นมา 2 รูปแบบ คือ Rule-Based Bot (กำหนดด้วยกฎต่าง ๆ) และ AI Bot (ปัญญาประดิษฐ์) โดย Rule-Based Bot นั้น จะทำงานตามกฎและคำสำคัญที่ถูกกำหนดไว้ ถ้าหากคำถามไม่ตรงกับที่กำหนดไว้ โปรแกรมอาจให้คำตอบที่ไม่ดีหรือทำงานผิดพลาดได้ ส่วน AI Bot นั้น จะใช้ปัญญาประดิษฐ์ (AI) และ Machine Learning เข้ามาช่วยให้ Chatbot นั้นฉลาดและเข้าใจภาษาของมนุษย์มากขึ้น



รูปที่ 2.1 การสนทนาระหว่างมนุษย์และแช็ตบอต

2.2 Skype for Business

Skype for Business มีชื่อเดิมว่า Microsoft Lync เป็นบริการที่ใช้เพื่อการสื่อสารและการทำงานร่วมกันอย่างมีประสิทธิภาพภายในองค์กร โดยมีคุณสมบัติรองรับการสนทนาผ่านข้อความ เสียง วิดีโอ รวมทั้งการประชุมออนไลน์ ผู้ใช้งานสามารถค้นหา เชื่อมต่อ และสื่อสารกับผู้ใช้งาน Skype ทั้งในและนอกองค์กรได้ มีระบบความปลอดภัยและการบริหารจัดการที่เหมาะสมกับระดับองค์กร Skype for Business เป็นส่วนหนึ่งของชุดโปรแกรม Microsoft Office (Office 365) ซึ่งจะช่วยให้ผู้ใช้สามารถติดต่อสื่อสารหรือเข้าประชุมควบคู่กับการทำงานในโปรแกรม Office อย่างอื่นได้อย่างง่ายดาย

Skype for Business

รูปที่ 2.2 Skype for Business logo

(ที่มา: <http://www.mvpskill.com/kb/addingusercontactsinskypeforbusiness.html>)



CALL FORWARDING OFF

รูปที่ 2.3 หน้าต่างหลักของโปรแกรม Skype for Business

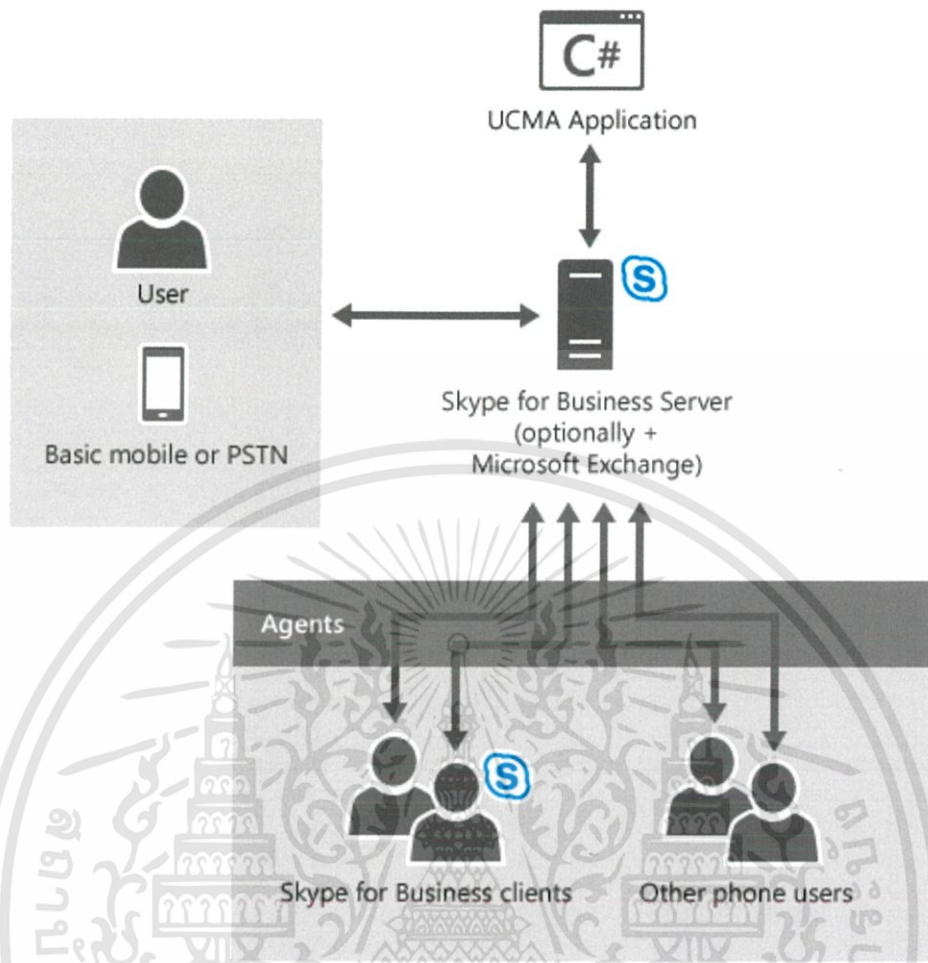
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 หน้าต่างสนทนาของโปรแกรม Skype for Business

2.3 Microsoft Unified Communications Managed API 5.0

Microsoft Unified Communications Managed API 5.0 (UCMA 5.0) คือโปรแกรมและชุดคำสั่งที่สร้างขึ้นโดย Microsoft เพื่อให้ให้นักพัฒนาซอฟต์แวร์ได้ใช้ในการสร้างแอปพลิเคชันที่สามารถเข้าถึงและใช้ประโยชน์จาก Skype for Business platform ได้เต็มประสิทธิภาพ เช่น แอปพลิเคชันสามารถจัดการการโทรและข้อความ แอปพลิเคชันระบบโต้ตอบอัตโนมัติ (IVR หรือ Chatbot) การบันทึกการสนทนา เป็นต้น



รูปที่ 2.5 ภาพรวมการทำงานของ Microsoft UCMA 5.0

(ที่มา: <https://docs.microsoft.com/en-us/skype-sdk/ucma/ucma-5-0-business-scenarios>)

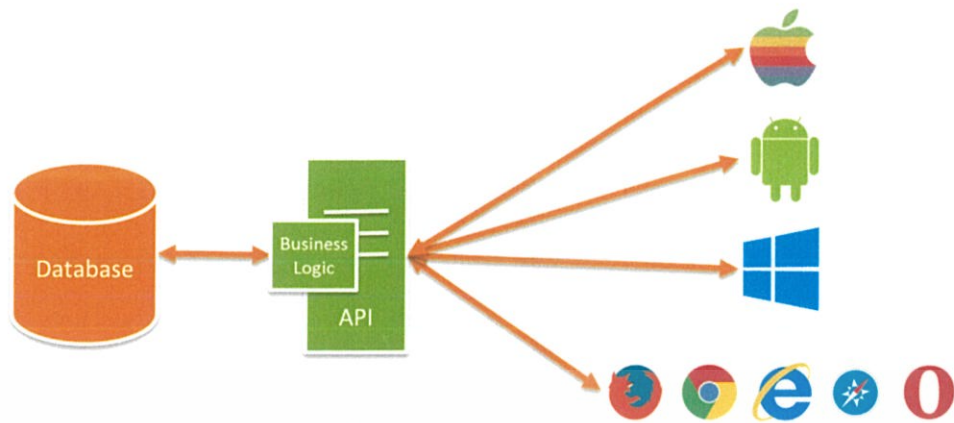
2.4 API

API (Application Programming Interface) หมายถึงวิธีการที่ระบบปฏิบัติการ ไลบรารี หรือบริการอื่น ๆ เปิดให้โปรแกรมคอมพิวเตอร์สามารถติดต่อเรียกใช้งานได้

API สามารถแบ่งออกได้เป็น 2 ประเภท คือ

1. API ที่ขึ้นกับภาษา (Language-Dependent API) คือ API ที่สามารถเรียกใช้จากโปรแกรมที่เขียนขึ้นด้วยภาษาเพียงภาษาใดภาษาหนึ่ง
2. API ที่ไม่ขึ้นกับภาษา (Language-Independent API) คือ API ที่สามารถเรียกได้จากโปรแกรมหลาย ๆ ภาษา

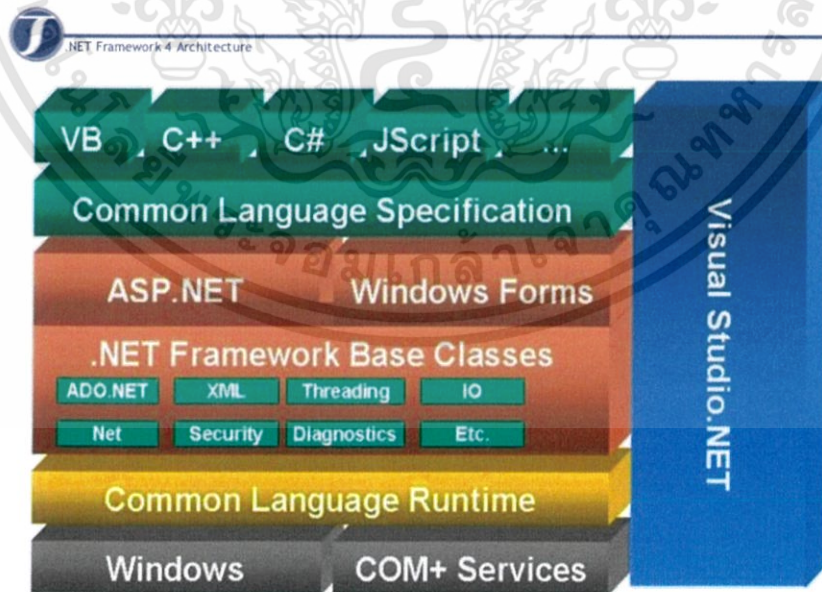
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 ภาพรวมการใช้งาน API

2.5 .Net Framework

.NET Framework คือ รูปแบบการพัฒนาโปรแกรมแบบใหม่ ที่ไมโครซอฟต์ได้พัฒนาขึ้นมา โดยมีจุดประสงค์คือเพื่อให้โปรแกรมที่พัฒนาขึ้นมาสามารถใช้งานในสภาวะของฮาร์ดแวร์หรือระบบปฏิบัติการที่แตกต่างกันได้อย่างไม่มีปัญหา และทำให้นักพัฒนาสามารถพัฒนาโปรแกรมใหม่ ๆ ได้ด้วยภาษาที่หลากหลายและสามารถทำงานร่วมกันได้ เช่น ภาษา C กับ ภาษา Java เป็นต้น .Net Framework ยังเป็นเครื่องมือในการพัฒนาโปรแกรมให้สามารถเชื่อมต่อกับโปรแกรมต่าง ๆ ของไมโครซอฟต์ได้โดยง่าย รวมไปถึงการทำงานภายในของระบบปฏิบัติการวินโดวส์เองด้วย ผู้พัฒนาจึงสามารถพัฒนาโปรแกรมใหม่ ๆ ได้โดยง่าย และรวดเร็ว ไม่ติดข้อจำกัดต่าง ๆ อย่างเช่นการพัฒนาโปรแกรมในสมัยก่อนอีกต่อไป



รูปที่ 2.7 โครงสร้างทางสถาปัตยกรรมของ .Net Framework 4

(ที่มา: <http://pankajs.com/microsoft-net-framework-architecture/>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

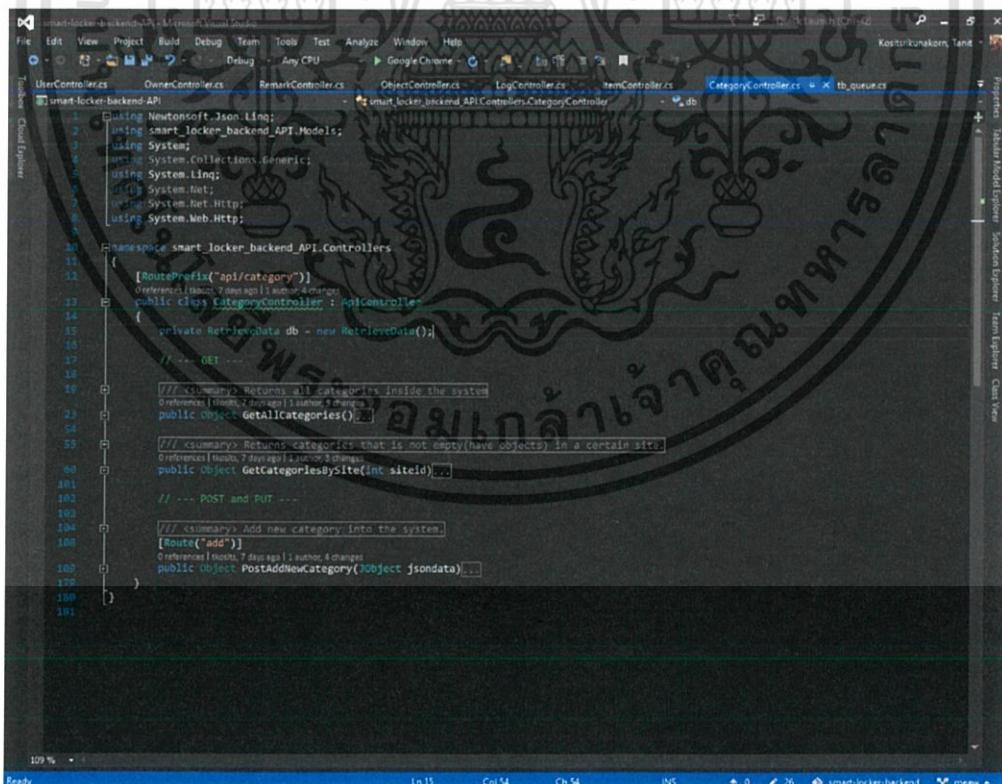
2.6 โปรแกรม Microsoft Visual Studio

โปรแกรม Microsoft Visual Studio เป็น Integrated Development Environment (IDE) ที่ถูกพัฒนาขึ้นโดย บริษัท ไมโครซอฟต์ เป็นเครื่องมือที่ช่วยให้นักพัฒนาซอฟต์แวร์สามารถพัฒนาซอฟต์แวร์และระบบต่าง ๆ ซึ่งสามารถติดต่อสื่อสารพูดคุยกับคอมพิวเตอร์ได้ เช่น เว็บไซต์ เว็บแอปพลิเคชัน และเว็บเซอร์วิส เป็นต้น ระบบที่รองรับการทำงานมีหลากหลาย เช่น ไมโครซอฟต์ วินโดวส์ ฟ็อกเกตพีซี โทรศัพท์สมาร์ทโฟน และ เว็บเบราวเซอร์



รูปที่ 2.8 Microsoft Visual Studio logo

(ที่มา: <http://jtower.com/blog/undocking-parts-of-team-explorer-in-visual-studio>)

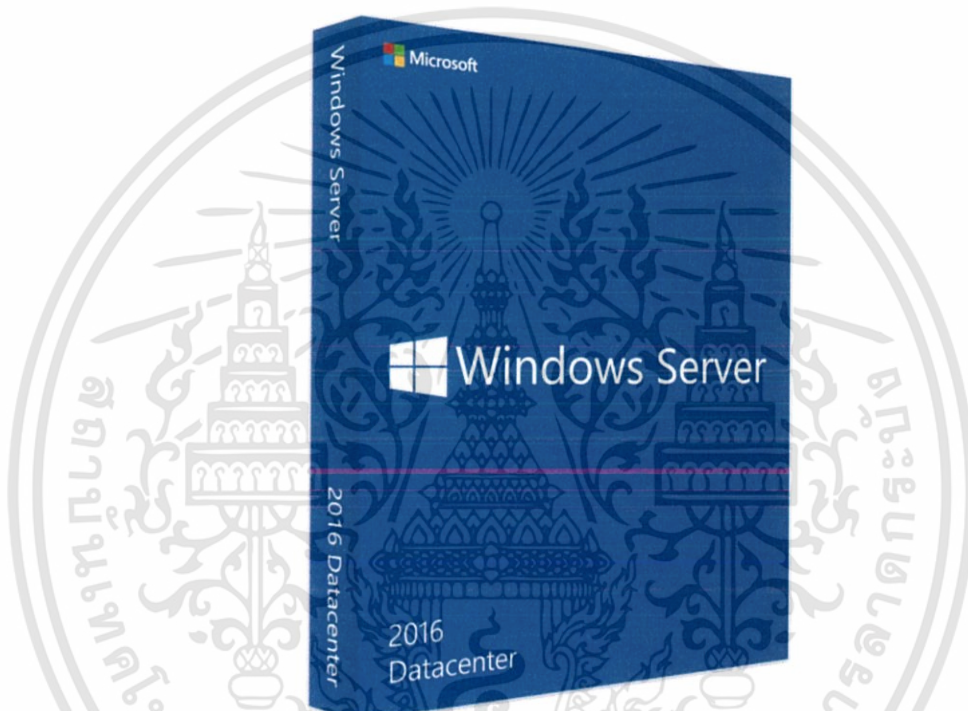


รูปที่ 2.9 หน้าต่างโปรแกรม Microsoft Visual Studio

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ระบบปฏิบัติการ Windows Server

Windows server คือ ระบบปฏิบัติที่พัฒนาขึ้นโดย บริษัท ไมโครซอฟต์ เพื่อใช้สำหรับคอมพิวเตอร์ที่ทำหน้าที่ให้บริการแก่คอมพิวเตอร์ที่เป็นลูกข่ายหรือโปรแกรมคอมพิวเตอร์อื่น โดยภายในเซิร์ฟเวอร์ที่มีทั้งฐานข้อมูลและตัวจัดการฐานข้อมูล มีการแบ่งปัน การประมวลผล โดยผ่านทางเครื่องลูกข่าย สนับสนุนการจัดการระดับองค์กร การจัดเก็บข้อมูลแอปพลิเคชันและการติดต่อสื่อสาร Windows Server ยังมีความสำคัญต่อเสถียรภาพความปลอดภัยระบบเครือข่ายและการปรับปรุงระบบไฟล์ต่าง ๆ ด้วย

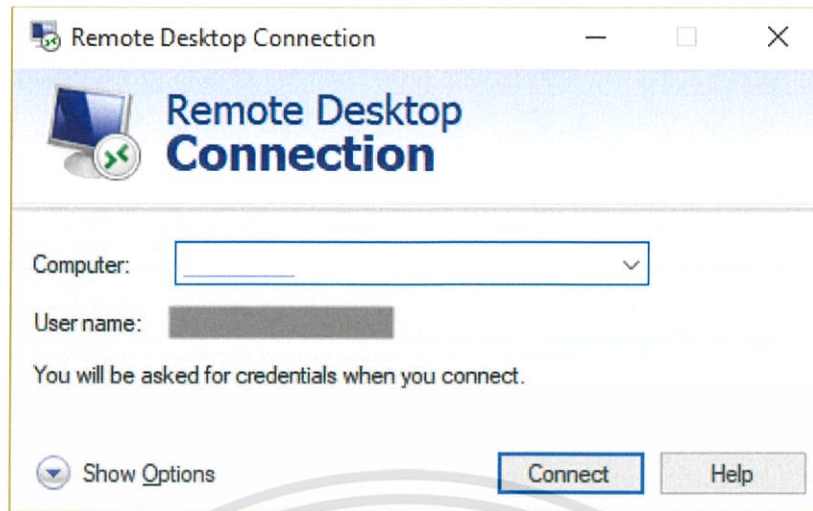


รูปที่ 2.10 ระบบปฏิบัติการ Windows Server

(ที่มา: <https://www.lizengo.co.uk/microsoft/windows-server-2016-datacenter>)

2.8 โปรแกรม Remote Desktop Connection

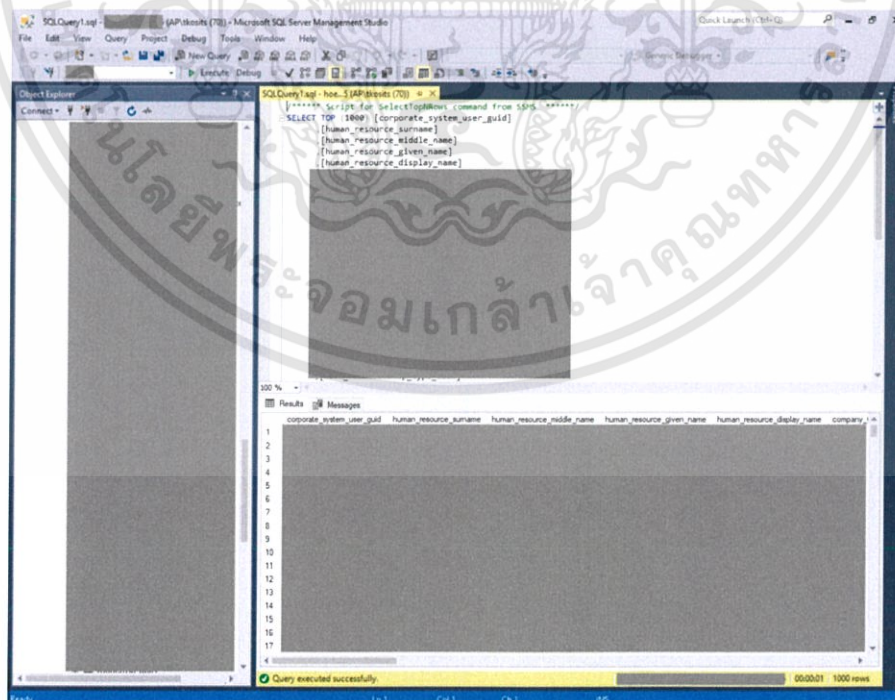
Remote Desktop Connection เป็นโปรแกรมภายในระบบปฏิบัติการ Microsoft Windows ที่มีหน้าที่สำหรับควบคุมเครื่องคอมพิวเตอร์ระยะไกล โดยสามารถควบคุมเครื่องคอมพิวเตอร์ที่เป็นเครื่องปลายทางได้ โดยจะสามารถมองเห็นหน้าจอและควบคุมการทำงานได้ ในการจะใช้โปรแกรมนี้เข้าไปควบคุมเครื่องใด ๆ นั้น จะต้องมีการตั้งค่าเครื่องฝั่งปลายทางให้อนุญาตให้เครื่องอื่นสามารถเข้าไปควบคุมจากระยะไกลได้ และต้อง มี Account ที่มีการรักษาความปลอดภัยด้วยรหัสผ่าน เพื่อใช้ในการล็อกอินเข้าสู่ระบบเพื่อไปควบคุม



รูปที่ 2.11 หน้าต่างการล็อกอินเพื่อเข้าควบคุมเครื่องคอมพิวเตอร์ระยะไกลของโปรแกรม Remote Desktop Connection

2.9 โปรแกรม Microsoft SQL Server Management Studio

โปรแกรม Microsoft SQL Server Management Studio (SSMS) เป็นโปรแกรมสำหรับการเข้าถึงข้อมูล (access), กำหนดค่า (configure), จัดการ (manage), บริหาร (administrate) และพัฒนา (develop) องค์ประกอบต่าง ๆ ของ Microsoft SQL Server โดยที่ SSMS เป็นโปรแกรมที่รวบรวมเครื่องมือที่ใช้ในการจัดการฐานข้อมูลเหล่านี้ไว้ในรูปแบบของรูปภาพกราฟิก



รูปที่ 2.12 หน้าต่างของโปรแกรม Microsoft SQL Server Management Studio

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 ภาษา C#

C# คือ ภาษาคอมพิวเตอร์ประเภท object-oriented programming ที่พัฒนาขึ้นโดยบริษัท ไมโครซอฟต์ โดยมีจุดมุ่งหมายในการรวมความสามารถการคำนวณของภาษา C++ เข้ากับรูปแบบการโปรแกรมที่ง่ายกว่าของ Visual Basic โดย C# มีพื้นฐานจาก C++ และเก็บส่วนการทำงานคล้ายกับ Java

C# ได้รับการออกแบบให้ทำงานกับ .NET platform ของ Microsoft โดยมีจุดมุ่งหมายคือ อำนวยความสะดวกในการแลกเปลี่ยนสารสนเทศและบริการผ่านเว็บ และทำให้ผู้พัฒนาสร้างโปรแกรมประยุกต์ในขนาดกะทัดรัด C# ทำให้โปรแกรมง่ายขึ้นผ่านการใช้ Extensible Markup Language (XML) และ Simple Object Access Protocol (SOAP) ซึ่งยอมให้เข้าถึง object ของโปรแกรมหรือ method โดยผู้เขียนโปรแกรมไม่ต้องเขียนคำสั่งเพิ่มในแต่ละขั้นตอน เนื่องจากผู้เขียนโปรแกรมสามารถสร้างบนคำสั่งที่มีอยู่ แทนที่การคัดลอก C# ได้รับการรับรองจากหน่วยงาน ECMA (หน่วยงานกำหนดมาตรฐานสากลด้านสารสนเทศ) และ ISO และปัจจุบันไมโครซอฟต์ยังพัฒนาภาษานี้อย่างต่อเนื่อง



รูปที่ 2.13 C# logo

2.11 ภาษา XML

คำว่า XML ย่อมาจาก Extensible Markup Language คือภาษาหนึ่งที่ใช้ในการแสดงผลข้อมูลที่เรียกว่า Meta Data ใช้สำหรับกำหนดรูปแบบของคำสั่ง Markup ต่าง ๆ มีโครงสร้างที่ประกอบด้วยแท็กเปิดและแท็กปิด ภาษา XML ผู้ใช้สามารถสร้างแท็กรวมทั้งกำหนดโครงสร้างของข้อมูลตัวเอง XML นั้นไม่สามารถแสดงผลได้ด้วยตัวของมันเอง ต้องมีการใช้ร่วมกับภาษาอื่น ๆ เช่น HTML, JSP, PHP, ASP เป็นต้น

```

<?xml version="1.0"?>
<IC_SPEC>
  <WIDTH>32</WIDTH>
  <BLOCK_LIST>
    <BLOCK>
      <NAME>Buffer</NAME>
      <REGISTER Key="BufferBC_CONFIG">
        <ADDR>0</ADDR>
        <REGISTER_ACRONYM>BC_CONFIG</REGISTER_ACRONYM>
        <ACCESS>RW</ACCESS>
        <BIT_LIST NumberOfEntries="20">
          <BIT HighestIndexOfAcronyms="0">
            <BIT_MSB>8</BIT_MSB>
            <BIT_ACRONYM>RefrCnt
          ..
        </REGISTER>
      <REGISTER Key="BufferBC_CONFIG">
        ..
      </REGISTER>
    ..
  </BLOCK>
</BLOCK_LIST>
</IC_SPEC>

```

รูปที่ 2.14 ตัวอย่าง code ภาษา XML

(ที่มา: https://www.eetimes.com/document.asp?doc_id=1266772)

2.12 รูปแบบข้อมูลประเภท JSON

JSON (JavaScript Object Notation) คือ รูปแบบของข้อมูลที่ใช้สำหรับแลกเปลี่ยนข้อมูลที่มีขนาดเล็ก ซึ่งมนุษย์สามารถอ่านทำความเข้าใจได้ง่าย และสามารถถูกสร้างและอ่านโดยเครื่องได้ง่าย เป็นรูปแบบข้อมูลตัวอักษรที่มีความเป็นอิสระอย่างสมบูรณ์ ไม่ขึ้นกับภาษาใด แต่จะมีหลักการการเขียนที่คุ้นเคยกับนักเขียนโปรแกรมภาษาต่าง ๆ ได้ ไม่ว่าจะเป็น ภาษา C, C++, C#, Java, JavaScript, Perl, Python และอื่น ๆ คุณสมบัติเหล่านี้ทำให้ JSON เป็นภาษาแลกเปลี่ยนข้อมูลที่มีสมบูรณ์แบบ โดย JSON จะมีรูปแบบการเก็บข้อมูลแบบเป็นคู่ Key-Value หรือเป็นแบบ Array และ สามารถนำมาใช้แทนการเก็บข้อมูลรูปแบบของภาษา XML ได้

```

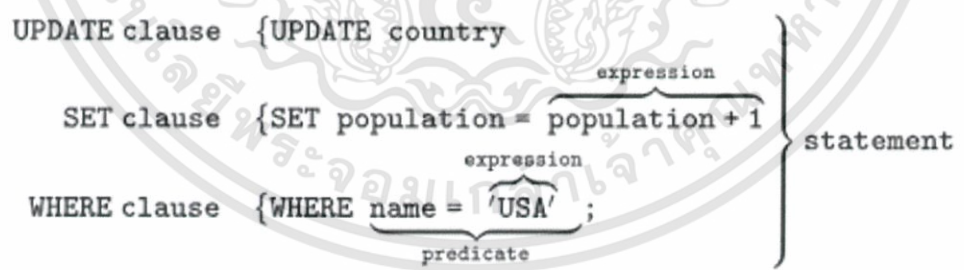
{
  "Query": "SELECT * FROM tb1",
  "Destination": {
    "Name": "db1",
    "Server": "test1"
  },
  "ExportFormat": "JSON"
}

```

รูปที่ 2.15 ตัวอย่างการเก็บข้อมูลในรูปแบบของ JSON

2.13 ภาษา SQL

SQL มาจากคำว่า Structured Query Language เป็นภาษามาตรฐานในการเข้าถึงในฐานข้อมูล เราสามารถใช้งานภาษา SQL ได้จากโปรแกรมต่าง ๆ ที่ต้องทำการกับระบบฐานข้อมูล เช่น ใช้ MS-SQL ในการทำการดึงข้อมูล (Retrieve Data) จากฐานข้อมูล และ SQL เป็นมาตรฐานกลางที่ใช้ในระบบฐานข้อมูลต่าง ๆ โดยเป็นมาตรฐานของ ANSI (American National Standard Institute) สามารถอ่านเข้าใจได้ง่าย เนื่องจากคล้ายคลึงกับการใช้ภาษาของมนุษย์



รูปที่ 2.16 ตัวอย่างโครงสร้างภาษา SQL

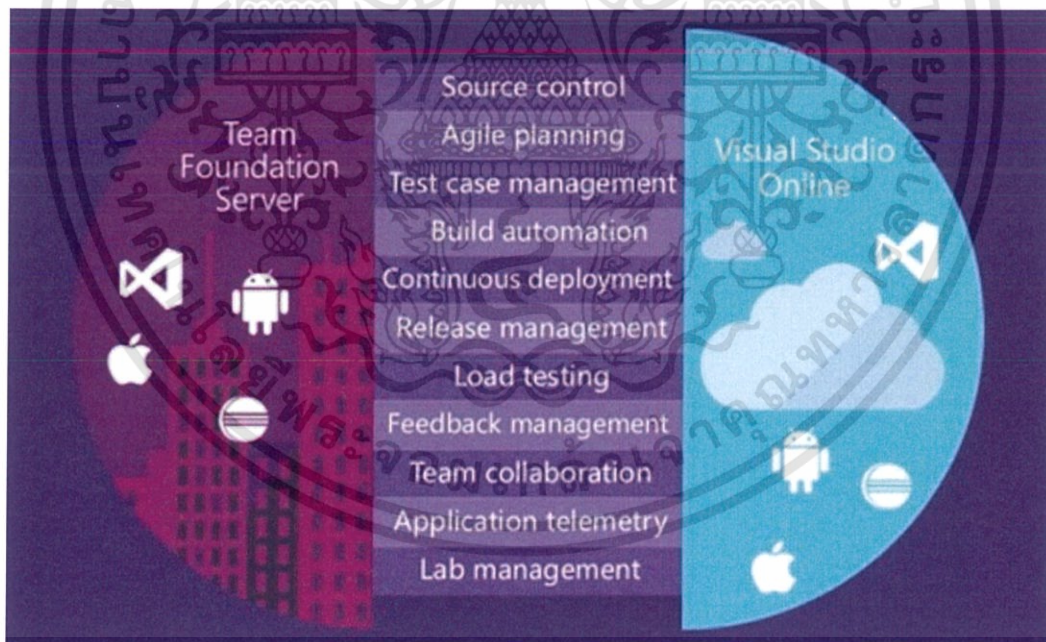
(ที่มา: <https://en.wikipedia.org/wiki/SQL>)

2.14 DDR5

DDR ย่อมาจาก Database Administration Data Repository เป็นชื่อของฐานข้อมูลที่ใช้สำหรับเก็บข้อมูลต่าง ๆ ที่เกี่ยวข้องกับเครื่องเซิร์ฟเวอร์ ฐานข้อมูล แอปพลิเคชัน และผู้ดูแลฐานข้อมูล โดย DDR5 จะมีหน้าเว็บไซต์เพื่อให้ผู้ใช้งานสามารถเข้าถึงหรือติดต่อกับผู้ดูแลฐานข้อมูลได้ ภายในเว็บไซต์จะมีคำสั่งต่าง ๆ ที่เกี่ยวข้องกับการดูแลฐานข้อมูล ให้ผู้ใช้งานเลือกใช้งานได้ตามความต้องการ เช่น คำสั่ง OnCall เพื่อตรวจสอบรายชื่อของผู้ดูแลฐานข้อมูลที่ประจำการอยู่ในขณะนั้น เป็นต้น ส่วนผู้ดูแลฐานข้อมูลมักจะใช้ DDR5 ผ่านโปรแกรม Microsoft SQL Server Management Studio ใน เข้าถึงฐานข้อมูลของ DDR5 แล้วทำการค้นหาข้อมูลต่าง ๆ เกี่ยวกับฐานข้อมูล เช่นมีฐานข้อมูลอะไรในเซิร์ฟเวอร์บ้าง หรือฐานข้อมูลแต่ละอันใช้ สำหรับแอปพลิเคชันใดบ้าง

2.15 Microsoft Team Foundation Server

Team Foundation Server หรือ TFS เป็นเครื่องมือช่วยในกระบวนการของการพัฒนาระบบผลิตโดยบริษัทไมโครซอฟต์ ซึ่งประกอบด้วยเครื่องมือต่าง ๆ ที่ช่วยในการพัฒนาระบบในหลาย ๆ ส่วน ตั้งแต่ส่วนการบริหารโครงการ ส่วนการพัฒนาระบบ เป็นต้น



รูปที่ 2.17 Microsoft Team Foundation Server

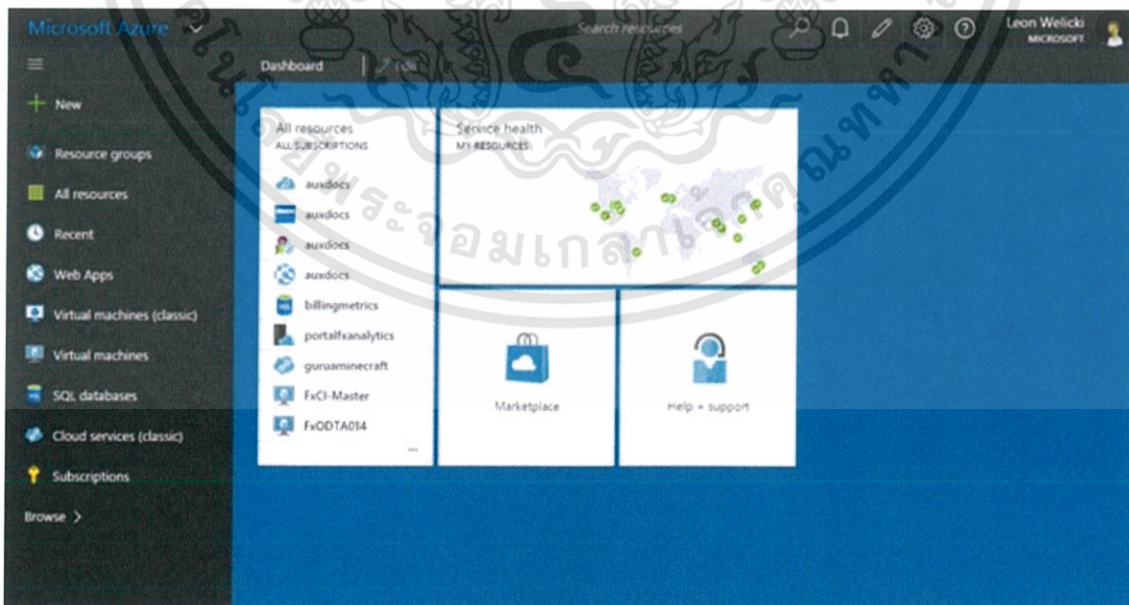
มาที่): <http://www.rlslog.net/microsoft-team-foundation-server-2018-with-update2-iso-tbe/>

2.16 ระบบ ITSM Remedy

ITSM เป็นระบบหนึ่งภายในบริษัทที่ใช้สำหรับการสร้างและส่งคำร้อง (Ticket) มายังผู้ดูแลระบบ ผังไอที (EMIT support group) เมื่อเจอปัญหา หรือต้องการให้ผู้ดูแลระบบทำงานตามคำร้อง (Ticket) โดยเมื่อผู้ใช้มีคำร้องหรือต้องการความช่วยเหลือจากผู้ดูแลระบบ ผู้ใช้จะทำการส่งคำร้องเพื่อขอความช่วยเหลือจากผู้ดูแลระบบ ITSM จะเก็บคำร้องทั้งหมดของผู้ใช้ ทำให้ง่ายต่อการจัดการตามลำดับความสำคัญ โดยจะมีการตั้งสถานะของคำร้องนั้น ๆ ว่าขณะนี้ดำเนินการถึงขั้นตอนไหน โดยบริษัทจะมีคำร้องหลักอยู่ 4 ประเภท คือ Incident (INC), Work Order (WO), Task (TAS) และ Change Request (CRQ)

2.17 Microsoft Azure

Microsoft Azure คือ คลาวด์แพลตฟอร์ม (Cloud Platform) ที่เปิดกว้างและมีความยืดหยุ่นสูง เพื่อทำหน้าที่เป็นรากฐานสำหรับการสร้าง ติดตั้ง รวมถึงจัดการ solution รองรับการใช้งานในรูปแบบ IAAS (Infrastructure as a Service) และ PASS (Platform as a Service) นั่นคือผู้ใช้งานสามารถเช่าใช้บริการโดยให้ไมโครซอฟต์ช่วยดูแลและจัดการด้านฮาร์ดแวร์ ซอฟต์แวร์ และระบบเครือข่าย เหมาะสำหรับองค์กรทุกขนาดไม่ว่าจะเป็นองค์กรขนาดเล็ก ขนาดกลาง หรือขนาดใหญ่ที่ต้องการพัฒนาและปรับปรุงศูนย์ข้อมูลให้มีความยืดหยุ่นและรองรับกับความต้องการทางธุรกิจได้อย่างรวดเร็วและมีประสิทธิภาพ การพัฒนาแอปพลิเคชันเพื่อใช้งานบน Azure ทำได้โดยใช้ภาษาที่หลากหลาย เช่น Java, PHP, .NET ผ่านชุดเครื่องมือ และเฟรมเวิร์กต่าง ๆ



รูปที่ 2.18 หน้าตาเว็บไซต์ Microsoft Azure Portal

(ที่มา: <https://azure.microsoft.com/en-us/blog/announcing-azure-portal-general-availability>)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.18 Raspberry Pi

Raspberry Pi คือ บอร์ดคอมพิวเตอร์ขนาดเล็ก (Single-Board Computer หรือ SBC) ที่ถูกพัฒนาขึ้นโดย Raspberry Pi Foundation มีคุณสมบัติเด่น คือ ติดต่อ และควบคุมอุปกรณ์อิเล็กทรอนิกส์ได้ โดยก่อนที่จะใช้ Raspberry Pi ได้นั้น จำเป็นต้องติดตั้งระบบปฏิบัติการให้กับ Raspberry Pi ก่อน โดยระบบปฏิบัติการที่นิยมใช้กัน คือ ระบบปฏิบัติการ Raspbian เนื่องจากเป็นระบบปฏิบัติการที่ถูกระบุสนับสนุนโดยตรงจากทาง Raspberry Pi Foundation บนบอร์ด Raspberry Pi จะมีสิ่งที่เรียกว่า GPIO (General Purpose Input-Output) ซึ่งมันคือส่วนที่เอาไว้ต่อสัญญาณ Input/output เข้ากับวงจรอิเล็กทรอนิกส์ภายนอกได้ การรับ/ส่งสัญญาณ Input/output ผ่าน GPIO จำเป็นต้องเขียนโปรแกรมสั่งงาน โดยภาษาที่นิยมใช้กันทั่วไป ได้แก่ ภาษา Python แต่นอกจากภาษา Python แล้ว ก็ยังมีภาษาอื่น ๆ ให้เลือกใช้กันอีก เช่น C/C++, Shell Script และภาษาอื่น ๆ



รูปที่ 2.19 Raspberry Pi 3 Model B

(ที่มา: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>)

2.19 ภาษา Python

Python คือชื่อภาษาที่ใช้ในการเขียนโปรแกรม ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษา Python ได้ทั้งบนระบบปฏิบัติการได้หลากหลาย เช่น Unix, Linux , Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD เป็นต้น อีกอย่างหนึ่งภาษานี้เป็น Open Source เหมือนอย่าง PHP ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมได้โดยไม่ต้องเสียค่าใช้จ่าย และความเป็น Open Source ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ Python มีความสามารถสูงขึ้น และใช้งานได้ครอบคลุมกับทุกลักษณะงาน ภาษา Python นั้นถูกพัฒนาขึ้นมาโดยมีความตั้งใจว่าจะให้เป็น

ภาษาที่อ่านง่าย ถูกออกแบบมาให้มีโครงสร้างที่มองเห็นได้โดยไม่ซับซ้อน โดยมักจะใช้คำในภาษาอังกฤษใน
ขณะที่ภาษาอื่นใช้เครื่องหมายวรรคตอน นอกจากนี้ Python มีข้อยกเว้นของโครงสร้างทางภาษาน้อยกว่า
ภาษา C และ Pascal



รูปที่ 2.20 Python logo

(ที่มา: <https://www.python.org/>)



ขั้นตอนการดำเนินงานพัฒนา

3.1 ขั้นตอนการดำเนินงานพัฒนาระบบอัตโนมัติสำหรับแชทบอต

DBAM Bot เป็นแชทบอตที่ถูกพัฒนาขึ้นโดย DBAM และได้ถูกนำมาใช้เพื่อช่วยตอบคำถามที่เกี่ยวข้องกับฐานข้อมูล ทั้งนี้เนื่องจากความต้องการของผู้ใช้งานและความต้องการทางธุรกิจของหน่วยงานที่เปลี่ยนไป ทำให้ต้องมีการปรับปรุงการทำงานของแชทบอตให้ตอบรับกับสถานการณ์ให้มากที่สุด โดยในการพัฒนาระบบครั้งนี้ ผู้พัฒนาได้มีโอกาสร่วมกันกับเพื่อนร่วมงานทั้งในประเทศและต่างประเทศ ผู้พัฒนาจึงได้ยึดรูปแบบการทำงานของ Scrum Framework เป็นหลัก เพื่อให้การทำงานร่วมกันสามารถเป็นไปได้อย่างลื่นไหล และสามารถแก้ไขปัญหา รวมถึงปรับเปลี่ยนแผนการทำงานได้อย่างทันท่วงที ทั้งนี้เพื่อให้การทำงานร่วมกันกับผู้ร่วมงานในต่างประเทศเป็นไปได้อย่างมีประสิทธิภาพ จึงได้มีการจัดประชุมสองครั้งต่อสัปดาห์ เพื่อติดตามความคืบหน้า วางแผน และร่วมกันแก้ไขปัญหา โดยแผนการทำงานทั้งหมด แสดงได้ ดังตารางที่ 3.1

ตารางที่ 3.1 แผนการทำงานพัฒนาระบบแชทบอต

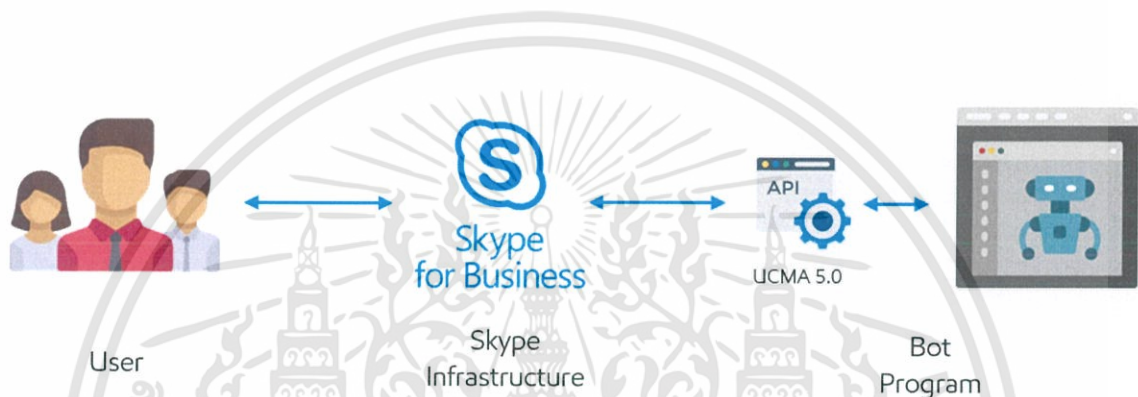
งาน \ ระยะเวลา	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
ประชุมงาน	2 ครั้ง/สัปดาห์					
ศึกษาระบบ	■					
ศึกษาความต้องการ	■	■				
วางแผนการพัฒนา			■	■	■	
พัฒนาระบบ			■	■	■	
ทดสอบการทำงาน				■	■	■
นำไปใช้งานจริง						■

3.1.1 ศึกษาการทำงานของระบบ

เนื่องจากโปรแกรมแชทบอตนี้ได้ถูกพัฒนาขึ้นและใช้งานอยู่เดิม ผู้พัฒนาจึงจำเป็นที่จะต้องเข้าใจถึงหลักการการทำงานของโปรแกรมเพื่อที่จะสามารถพัฒนาโปรแกรมได้ ผู้วิจัยได้ศึกษาหลักการการทำงานของโปรแกรมโดยอ่านเอกสารประกอบการพัฒนา รวมไปถึงอ่าน code ของโปรแกรมร่วมด้วย โดยผู้วิจัยสามารถสรุปความเข้าใจได้ดังนี้ โปรแกรมแชทบอต DBAM Bot พัฒนาขึ้นจากโปรแกรม Visual Studio

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ **19** ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยใช้ภาษา C# ซึ่งเมื่อพัฒนาโปรแกรมและทำการ build จะได้ชุดไฟล์ที่จำเป็นสำหรับการเรียกใช้งาน โดยเมื่อโปรแกรมถูกเรียกใช้งานขึ้นมาจะทำการเชื่อมต่อเข้ากับระบบของ Skype for Business (Skype for Business Server) โดยอาศัยการทำงานของ UCMA 5.0 เข้าช่วย และเมื่อมีผู้ใช้งานเริ่มต้นการสนทนา กับแชทบอตโดยการส่งข้อความถึง DBAM Bot ผ่านโปรแกรม Skype for Business ข้อความนั้นจะถูกส่งผ่านจาก Skype for Business ผ่านทาง UCMA 5.0มายังโปรแกรมแชทบอตที่กำลังทำงานอยู่ แชทบอตจะทำการประมวลผลและส่งข้อความตอบสนองที่เหมาะสมกลับไปหาผู้ใช้งาน ภาพรวมโครงสร้างของระบบแสดงได้ ดังรูปที่ 3.1



รูปที่ 3.1 ภาพรวมโครงสร้างของระบบแชทบอต

สำหรับการทำงานของโปรแกรมนั้นจะเริ่มต้นที่ class Program ซึ่งเป็น class หลักของโปรแกรมซึ่งประกอบไปด้วย method ที่มีชื่อว่า Start() และ Stop() ใช้สำหรับเริ่มต้นการทำงานและหยุดการทำงานตามลำดับ ภายใน method Start() จะประกอบไปด้วยคำสั่งสำหรับสร้าง object ชื่อว่า Bot ขึ้นจาก class DatabaseBot(), เริ่มต้นการทำงานของส่วนเก็บข้อมูลการใช้งาน (log) นอกจากนี้การสร้าง method เพิ่มเติมภายใน class เพื่อจัดการการทำงานก็สามารถทำได้ เช่น การสั่งเริ่มต้นการทำงานของคำสั่งบางคำสั่งให้เป็นกิจวัตร เป็นต้น

```

15 namespace XomBot.DatabaseBot
16 {
17     class Program : BotProgram
18     {
19         public override string ServiceName
20         {
21             get { return "XomBot.DatabaseBot"; }
22         }
23         static void Main(string[] args)
24         {
25             Program prog = new Program();
26             prog.MainProgram(prog);
27             Console.WriteLine("Debug path: main program.cs");
28         }
29         public override void Start()
30         {
31
32             base.Start();
33             String logpath = ConfigurationManager.AppSettings["logpath"];
34             Log.LogLevel loglevel;
35             loglevel = Log.LogLevel.DEBUG;
36             Log.LogInit(logpath, "MyBot", loglevel);
37             Console.WriteLine("saving log to: " + logpath);
38             Console.WriteLine("log level: " + loglevel);
39
40             Console.WriteLine("Debug path: 'Start' 'Program.cs'");
41             try
42             {
43                 Bot = new DatabaseBot();
44             }
45             catch (RealTimeException ex)
46             {
47                 Log.Error("Fail to get bot: " + ex.Message, typeof(Program));
48             }
49             Log.Info("Started Test bot.", typeof(Program));
50         }
51         public override void Stop()
52         {
53             Console.WriteLine("Debug path: 'Stop' 'Program.cs'");
54             base.Stop();
55         }
56     }
57 }

```

รูปที่ 3.2 Code ภาษา C# ของ class Program

การเชื่อมต่อเข้ากับระบบของ Skype for Business นั้นจะถูกทำภายใน class DatabaseBot โดย method DatabaseBot() จะถูกเรียกใช้งานโดย method Start() ใน class Program เมื่อถูกเรียกใช้งานแล้ว DatabaseBot() จะทำการดึงข้อมูลที่จำเป็นสำหรับการเชื่อมต่อได้แก่ ชื่อผู้งานและรหัสผ่าน ซึ่งถูกเก็บไว้ที่ไฟล์ App.config เมื่อได้ข้อมูลมาแล้วจึงเชื่อมต่อเข้ากับระบบ เมื่อเชื่อมต่อแล้ว เซ็ตบอตจึงสามารถรับและส่งข้อความผ่านระบบ Skype for Business ได้

```

13 namespace Kombot.DatabaseBot
14 {
15     class DatabaseBot : Bot
16     {
17         public DatabaseBot() : base()
18         {
19             Console.WriteLine("Debug path: DatabaseBot DatabaseBot.cs");
20
21             botURI = ConfigurationManager.AppSettings["boturi"];
22             serverFQDN = ConfigurationManager.AppSettings["priserverfqdn"];
23             botName = ConfigurationManager.AppSettings["botname"];
24
25             username = ConfigurationManager.AppSettings["botname"];
26             pwd = ConfigurationManager.AppSettings["botpassword"]; // leave password blank for service authentication
27             domain = ConfigurationManager.AppSettings["botdomain"];
28             string photouri = ConfigurationManager.AppSettings["avatar"];
29
30             //set the MessageHandler to use for responding to the chat
31             setMessageHandlerType(typeof(BotMessageHandler));
32
33             //set the total number of concurrent chat sessions
34             resetMaxConcurrentCalls(System.Convert.ToInt32(ConfigurationManager.AppSettings["maxConcurrentSessions"]));
35
36             //not set up the chat bot and you're good
37             Connect(username, pwd, domain, "", photouri);
38         }
39
40         public void Run()
41         {
42         }
43     }
44 }
45 }

```

รูปที่ 3.3 Code ภาษา C# ของ class DatabaseBot

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <configuration>
3   <startup>
4     <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0.1" />
5   </startup>
6   <appSettings>
7     <!-- service info (not in use) -->
8     <add key="username" value="DBAM Bot" />
9     <!-- bot info - for production, the password should be left BLANK and set at the windows service level -->
10    <!-- COMMENT: if you're moving from acceptant to production, change 's' values: botname, botpassword, and boturi -->
11    <add key="botname" value="DBAM Bot" />
12    <add key="botpassword" value="" />
13    <add key="botdomain" value="" />
14    <add key="boturi" value="" />
15    <add key="displayname" value="DBAM Bot" />
16    <add key="avatar" value="" />
17    <!-- serving server and application info -->
18    <!-- COMMENT: if you're moving from acceptant to production, edit production application and server name below, also 's' in uris. -->
19    <add key="servingDatabaseName" value="" />
20    <add key="servingServerName" value="" />
21
22
23    <!-- sharepoint url -->
24    <add key="spConsultantUrl" value="" />
25    <!-- add key="priserverfqdn" value="" />
26    <add key="priserverfqdn" value="" />
27    <!-- ITSM template for issues - it should be a template that creates issues to the bot owner's queue -->
28    <!-- test -->
29    <add key="issueTemplate" value="" />
30
31
32
33
34
35    <!-- log configuration -->
36    <add key="logpath" value="" />
37    <add key="loglevel" value="DEBUG" />
38    <add key="maxConcurrentSessions" value="10" />
39    <add key="ClientSettingsProvider.ServiceUri" value="" />
40    <add key="ClientSettingsProvider.ConnectionStringName" value="DefaultConnection" />
41  </appSettings>
42  <system.web>
43    <membership defaultProvider="ClientAuthenticationMembershipProvider">
44      <providers>
45        <add name="ClientAuthenticationMembershipProvider" type="System.Web.ClientServices.Providers.ClientWindowsAuthenticationMembershipProvider, System.Web.Extensions, Version=4.0.10931.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
46      </providers>
47    </membership>

```

รูปที่ 3.4 Code ภาษา XML ของไฟล์ App.config

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทุกครั้งที่โปรแกรมได้รับข้อความจากผู้ใช้งานผ่าน Skype for Business method getIntentFromMessage() ใน class BotMessageHandler จะถูกเรียกใช้งาน และจะทำการเรียก method calculateIntent() จาก class IntentProcessor ดังรูปที่ 3.5 เพื่อทำการประมวลผลหาความต้องการของผู้ใช้งานหรือเรียกได้ว่า Intent โดยอาศัยจากการเปรียบเทียบกับคำสำคัญที่ถูกกำหนดไว้ เช่น หากผู้ใช้งานพิมพ์ว่า “Create an access review for me” แชนด์บอตจะพบว่าคำสำคัญของประโยคนี้นี้คือ “access review” และจะได้ Intent ของประโยคนี้นี้เป็น “accessreview” ดังรูปที่ 3.6

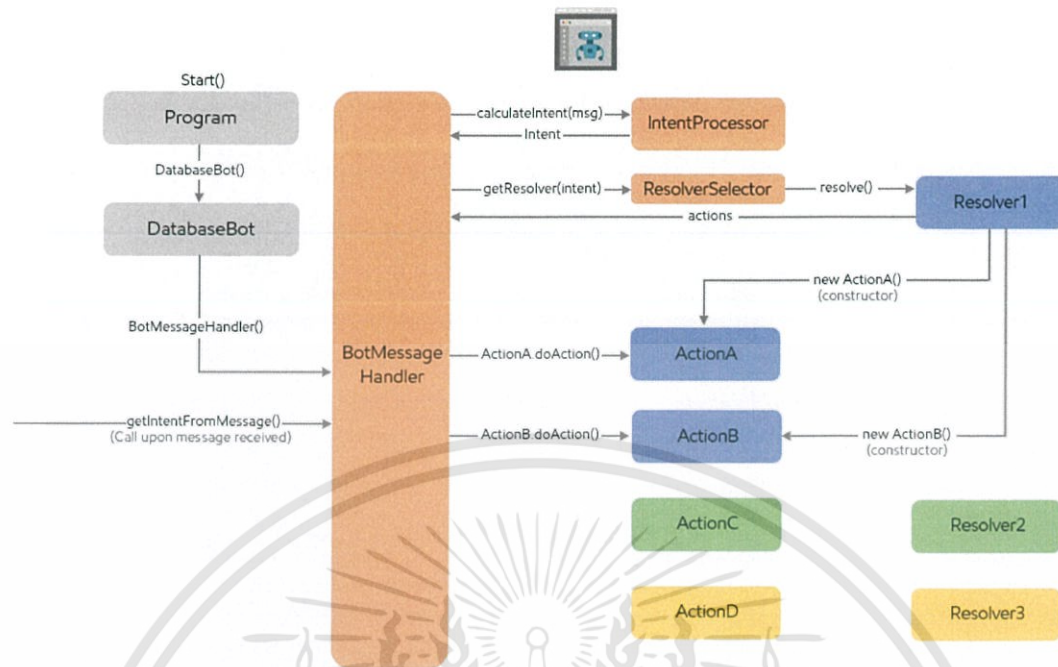
```
IIntentProcessor IntentProcessor = new IntentProcessor();
IIntent result;
result = (Intent)IntentProcessor.calculateIntent(message);
```

รูปที่ 3.5 การเรียกใช้งาน method calculateIntent() เพื่อหา Intent ของข้อความจากผู้ใช้งาน

Create an access review for me → Intent = “accessreview”
Keywords

รูปที่ 3.6 การหา Intent โดยอาศัยการเปรียบเทียบกับคำสำคัญ

เมื่อได้ Intent แล้ว method takeActionOnIntent() ใน class BotmessageHandler จะถูกเรียกใช้งานและเรียกใช้ Resolver ที่เกี่ยวข้องกับ Intent นั้น ภายใน Resolver คำสั่งที่เหมาะสมจะถูกกำหนดขึ้นเตรียมให้เรียกใช้งานเพื่อให้บรรลุความต้องการของผู้ใช้งานหรือเรียกได้ว่า action จากนั้น Resolver จะส่ง object ของ action ที่ต้องการเรียกใช้งานกลับมายัง BotMessageHandler และจะถูกเรียกใช้งานเพื่อให้แชทบอตเริ่มดำเนินการตามคำสั่งที่กำหนดขึ้นเตรียมไว้ และส่งผลลัพธ์กลับไปหาผู้ใช้งาน ทั้งนี้ก่อน Resolver จะถูกเรียกใช้งาน โปรแกรมจะทำการเช็คค่าตัวแปรที่มีชื่อว่า state โดยหากค่าภายในตัวแปรไม่ใช่ “NORMAL” Resolver ก่อนหน้าที่ถูกเรียกใช้งานจะถูกเรียกขึ้นโดยอัตโนมัติโดยไม่สนใจค่า Intent ล่าสุด เนื่องจากหากค่า state ไม่เท่ากับ “NORMAL” แล้ว หมายความว่า Resolver ก่อนหน้าที่ถูกเรียกใช้งานยังต้องการข้อความต่อไปจากผู้ใช้งานเพื่อทำงานให้บรรลุจุดประสงค์ ภาพรวมแสดงการทำงานภายในทั้งหมดของโปรแกรมแชทบอตแสดงได้ ดังรูปที่ 3.7



รูปที่ 3.7 ภาพรวมการทำงานภายในของโปรแกรมแชทบอท

3.1.2 ศึกษาและรวบรวมความต้องการปัจจุบัน

ในงานพัฒนานี้ ผู้พัฒนาได้ศึกษาและรวบรวมถึงความต้องการต่าง ๆ ที่จำเป็น และได้จำแนกประเภทของความต้องการในการพัฒนาระบบออกเป็น 2 ส่วน ได้แก่ ความต้องการของผู้พัฒนาเพื่อการพัฒนา ระบบ และความต้องการทางธุรกิจของหน่วยงาน

3.1.2.1 ความต้องการของผู้พัฒนาเพื่อการพัฒนา ระบบ

เพื่อให้การพัฒนาระบบเป็นไปตามข้อกำหนดของบริษัท และตรงต่อความต้องการทางธุรกิจของหน่วยงาน ผู้พัฒนาจึงได้จัดเตรียมสิ่งที่จำเป็นต่อการพัฒนาระบบไว้ ดังนี้

1. ขอและติดตั้งโปรแกรม Microsoft Visual Studio Professional 2015 พร้อม Microsoft Developer Network Subscription (MSDN)
2. ขอและติดตั้งโปรแกรม UCMA 5.0 SDK
3. ขอสิทธิ์เข้าถึง Team Foundation Server directory ที่เก็บ code DBAM Bot สำหรับการพัฒนา และเชื่อมต่อเข้ากับ Visual Studio
4. ขอสิทธิ์เข้าถึงกลุ่มของนักพัฒนาที่เกี่ยวข้อง

5. ขอสติธิ์เข้าถึงระบบ ITSM Remedy เพื่อการจัดการ Ticket ที่เกี่ยวข้อง
6. ขอสติธิ์เข้าถึงเครื่องเซิร์ฟเวอร์และฐานข้อมูลที่เป็นสำเนาสำหรับการทำงานของโปรแกรมแช็ตบอต

3.1.2.2 ความต้องการทางธุรกิจของหน่วยงาน

สำหรับความต้องการทางธุรกิจของหน่วยงาน เนื่องจากทีมของผู้พัฒนาทำงานโดยใช้รูปแบบการทำงานจาก Scrum Framework ซึ่งในแต่ละผลิตภัณฑ์จะมีผู้ที่รับผิดชอบหลักของผลิตภัณฑ์หรือเรียกว่าเจ้าของผลิตภัณฑ์ (Product Owner) ในแต่ละช่วงของปีจะมีการวางแผน รวบรวมความเห็นของผู้ใช้งาน วิเคราะห์บริบททางธุรกิจ เพื่อให้ได้ผลิตภัณฑ์ที่ตอบโจทย์กับธุรกิจปัจจุบันให้มากที่สุด สำหรับความต้องการของเจ้าของผลิตภัณฑ์ที่มีต่อ DBAM Bot ที่ผู้พัฒนาได้รับมานั้น สามารถจำแนกออกได้เป็น 4 ขอบเขตหลัก โดยมีรายละเอียดดังนี้

1. DBAM Bot สามารถรับรู้รายละเอียดของคู่สนทนาได้
 - ข้อมูลพื้นฐานของผู้ใช้งาน
 - ข้อมูลฐานข้อมูลของผู้ใช้งาน
2. DBAM Bot สามารถทำงานร่วมกับระบบอื่น ๆ ได้
 - ITSM API
 - DBAM Automation API
3. DBAM Bot สามารถเชื่อมต่อและทำงานกับฐานข้อมูลได้
 - คำสั่ง Access Review
 - คำสั่ง Backup Database
 - คำสั่ง Restore Database
 - คำสั่ง Refresh Database
 - คำสั่ง Run Script
4. DBAM Bot สามารถตรวจสอบสถานะต่าง ๆ ของเครื่องเซิร์ฟเวอร์ได้
 - คำสั่ง Babysit Database

3.1.3 วางแผนการพัฒนาระบบ

จากการศึกษาและรวบรวมข้อมูลความต้องการทางธุรกิจตามหัวข้อที่ 3.1.2.2 นั้น ผู้พัฒนาและเพื่อนร่วมงานได้ทำการวางแผนและออกแบบการพัฒนา และได้แบ่งภาระงานออกเป็น 2 ส่วนและแจกแจงให้ผู้พัฒนาและเพื่อนร่วมงาน โดยอาศัยความคล้ายคลึงของการพัฒนาโปรแกรม และความง่ายต่อการประยุกต์เข้าด้วยกันของโปรแกรม โดยผู้พัฒนารับผิดชอบงานในส่วนที่ 1 ดังตารางที่ 3.2

ตารางที่ 3.2 การแบ่งภาระงานที่รับผิดชอบของงานแช่ตบอต

ส่วนที่ 1	ส่วนที่ 2
<ul style="list-style-type: none">- พัฒนาการเชื่อมต่อเข้ากับระบบอื่น ๆ- พัฒนาให้ DBAM Bot รับรู้ข้อมูลของคู่สนทนาได้- คำสั่ง Access Review- คำสั่ง Backup Database- คำสั่ง Restore Database- คำสั่ง Refresh Database	<ul style="list-style-type: none">- พัฒนาให้ DBAM Bot รับรู้ข้อมูลฐานข้อมูลของคู่สนทนา- คำสั่ง Run Script- คำสั่ง Babysit Database

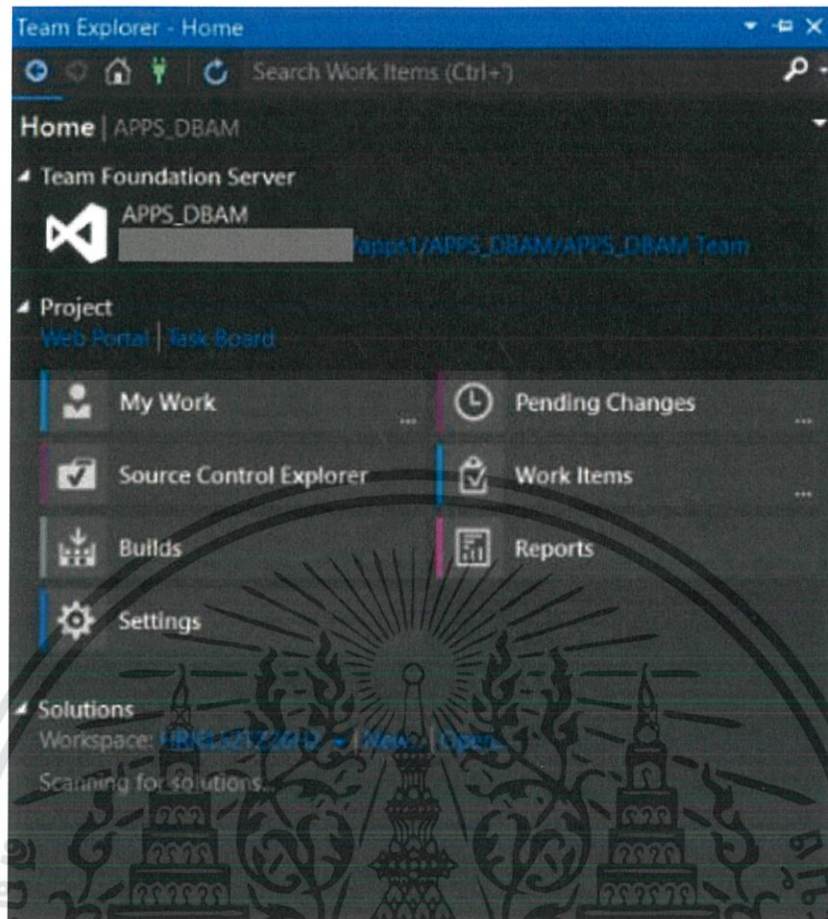
3.1.4 พัฒนาระบบ

จากการวางแผนงานและแบ่งภาระรับผิดชอบในหัวข้อที่ 3.1.3 ผู้พัฒนารับผิดชอบงานในส่วนที่ 1 ดังแสดงตามตารางที่ 3.2 โดยสามารถแจกแจงรายละเอียดการพัฒนาได้ ดังนี้

3.1.4.1 การเชื่อมต่อเข้า Team Foundation Server

ในการพัฒนาครั้งนี้ ผู้พัฒนาและเพื่อนร่วมงานได้ใช้บริการของ Team Foundation Server เพื่อทำ Version Control โดยได้ทำการสร้าง Branch master ขึ้นเพื่อเก็บ code ส่วนที่พร้อมจะนำไปใช้งาน branch develop เพื่อเก็บ code หลัก และเมื่อผู้พัฒนาหรือเพื่อนร่วมงานจะพัฒนาฟังก์ชันการทำงานใหม่ จะทำการสร้าง branch ขึ้นมาใหม่ เพื่อให้สามารถทำงานที่แตกต่างกันไปได้พร้อม ๆ กัน โดยที่ code ไม่ทับซ้อนกัน เมื่อเสร็จสิ้นการพัฒนาฟังก์ชันการทำงานนั้น ๆ แล้ว จึงจะทำการรวม branch ที่สร้างขึ้น กลับเข้าสู่ branch develop และรวมจาก branch develop สู่ master เมื่อพร้อมที่จะนำไปใช้จริงต่อไป สำหรับขั้นตอนการเชื่อมต่อ Visual Studio กับ Team Foundation Server ทำได้ดังนี้

1. เปิดโปรแกรม Microsoft Visual Studio
2. ไปที่ View > Team Explorer จะปรากฏหน้าต่าง ดังรูปที่ 3.8



รูปที่ 3.8 หน้าต่าง Team Explorer

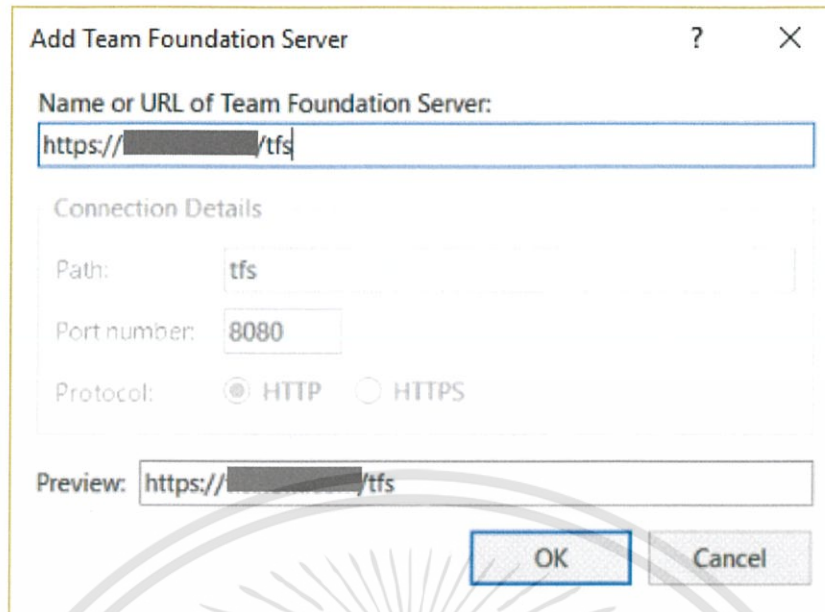
3. ที่หน้าต่าง Team Explorer กด Manage Connection > Connect to Team Project แล้วกดที่ Server..



รูปที่ 3.9 ปุ่ม Manage Connection ในหน้าต่าง Team Explorer

4. จะปรากฏหน้าต่าง ดังรูปที่ 3.10 กรอก URL ของ Team Foundation Server ที่ต้องการ และกด OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

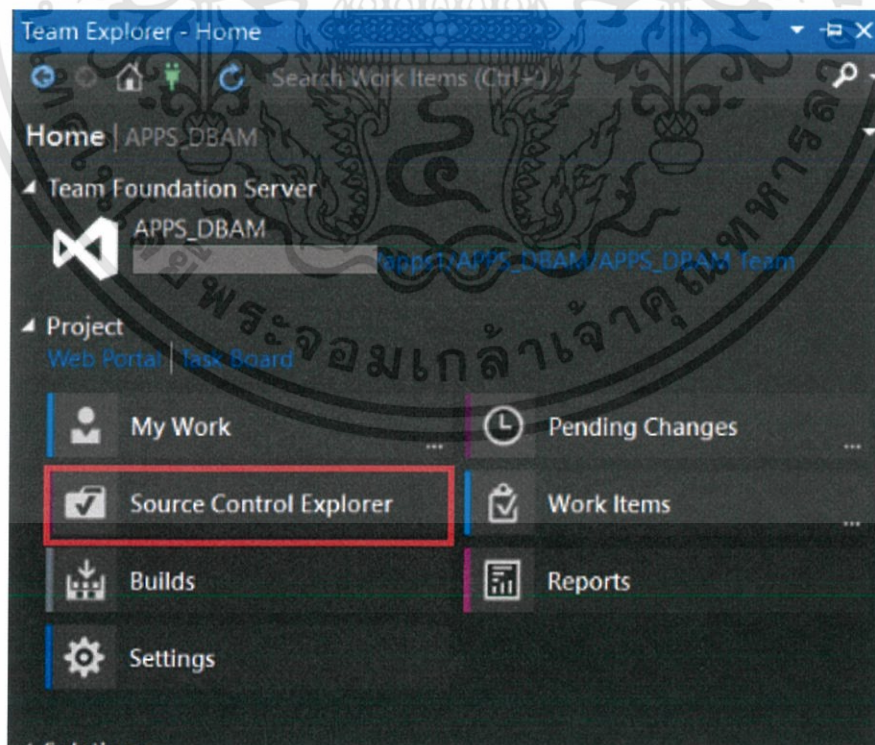


รูปที่ 3.10 หน้าต่าง Add Team Foundation Server

5. เลือก Project ที่ต้องการคือ APPS1 > APPS_DBAM และกด Connect

6. หลังจากเสร็จสิ้นการสร้างการเชื่อมต่อ ไปที่ Source Control Explorer ดัง

รูปที่ 3.11



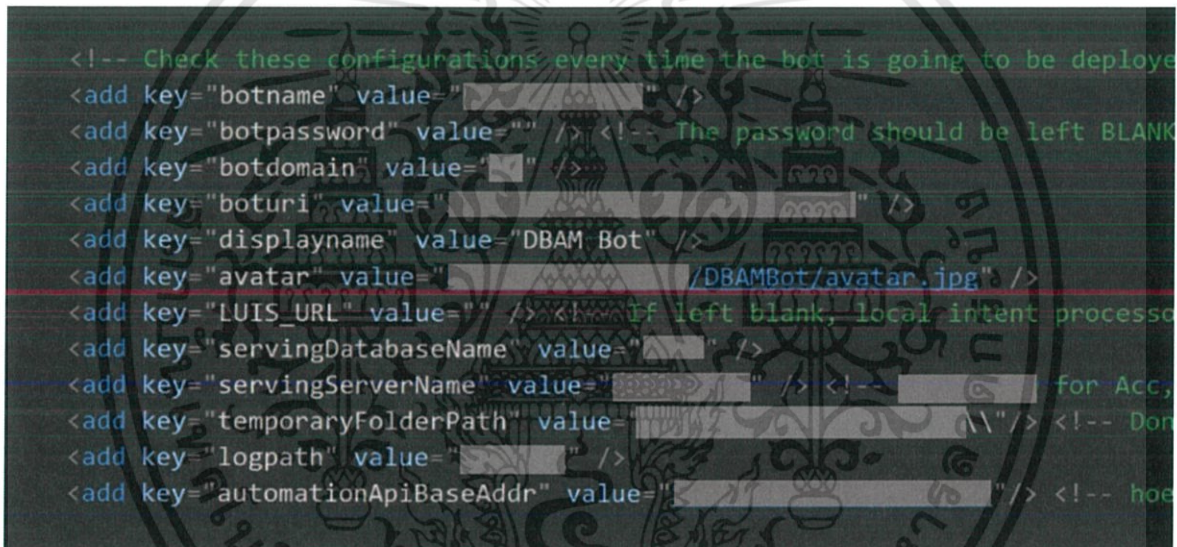
รูปที่ 3.11 ปุ่ม Source Control Explorer ภายในหน้าต่าง Team Explorer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ไปที่ APPS_DBAM คลิกขวาที่ DBAM Bot-2018 > Get Latest Version
8. จะได้ไฟล์สำหรับพัฒนาทั้งหมดของโปรแกรม DBAM Bot

3.1.4.2 การตั้งค่าภายในโปรแกรมแชทบอต

ในการพัฒนาโปรแกรมแชทบอต DBAM Bot ไฟล์ชื่อ App.config มีหน้าที่เป็นที่เก็บรวบรวมตัวแปรที่สำคัญสำหรับการกำหนดค่าการทำงานต่าง ๆ ภายในแชทบอต รวมถึงสามารถใช้เก็บค่าตัวแปรที่จะมีการถูกเรียกใช้จากหลาย ๆ ส่วนที่ต่างกันของโปรแกรมแต่มีค่าเหมือนกัน เพื่อให้สะดวกต่อการแก้ไขในภายหลังเนื่องจากจะสามารถแก้ไขได้ในทีเดียว และเนื่องจากแชทบอตใช้ค่าจากตัวแปรต่าง ๆ จากไฟล์นี้เพื่อทำงาน ผู้พัฒนาจึงจำเป็นต้องทำการกำหนดค่าให้ถูกต้องเพื่อให้แชทบอตทำงานได้ ตัวแปรต่าง ๆ ที่สำคัญต่อการทำงานของแชทบอตและต้องถูกกำหนดค่าให้ถูกต้อง แสดงได้ ดังรูปที่ 3.12



```
<!-- Check these configurations every time the bot is going to be deployed
<add key="botname" value=" " />
<add key="botpassword" value=" " /> <!-- The password should be left BLANK
<add key="botdomain" value=" " />
<add key="boturi" value=" " />
<add key="displayname" value="DBAM Bot" />
<add key="avatar" value=" /DBAMBot/avatar.jpg" />
<add key="LUIS_URL" value=" " /> <!-- If left blank, local intent processor
<add key="servingDatabaseName" value=" " />
<add key="servingServerName" value=" " /> <!-- for Acc,
<add key="temporaryFolderPath" value=" " /> <!-- Don
<add key="logpath" value=" " />
<add key="automationApiBaseAddr" value=" " /> <!-- hoe
```

รูปที่ 3.12 ตัวแปรที่สำคัญสำหรับการทำงานของแชทบอตในไฟล์ App.config

3.1.4.3 พัฒนาการเชื่อมต่อเข้ากับระบบอื่น ๆ

จากการศึกษาและรวบรวมความต้องการในหัวข้อ 3.1.2.2 เกือบทุกฟังก์ชันจำเป็นต้องมีการเชื่อมต่อผ่านอินเทอร์เน็ตภายในองค์กรเพื่อให้บรรลุความต้องการ เช่น การสร้าง Ticket ในระบบ ITSM Remedy ซึ่งต้องทำผ่าน ITSM API เป็นต้น และสำหรับฟังก์ชันที่เกี่ยวข้องกับฐานข้อมูล จากข้อกำหนดของบริษัท บริษัทไม่ต้องการให้แอปพลิเคชันใด ๆ มีการติดต่อกับฐานข้อมูลโดยตรง จึงทำให้การเชื่อมต่อแชทบอตกับฐานข้อมูลนั้นต้องทำผ่าน DBAM Automation API เท่านั้น ผู้พัฒนาจึงได้พัฒนา code ต้นแบบที่สามารถนำไปประยุกต์ใช้ในการเชื่อมต่อกับระบบอื่น ๆ ภายในโปรแกรมแชทบอตได้โดยใน code ต้นแบบนี้เริ่มต้นด้วยการสร้าง C# class ขึ้นมาโดยจะใช้ชื่อตัวแปรต่าง ๆ เหมือนกับตัวแปรที่ API

ต้องการ จากนั้นจะทำการสร้าง object จาก class นั้นและกำหนดค่าให้ตัวแปร แล้วนำ object ไปแปลงเป็นรูปแบบ JSON เพื่อให้สามารถส่งผ่านต่อไปยัง API ได้ ดังรูปที่ 3.13

```
public class Destination
{
    public string Name { get; set; }
    public string Server { get; set; }
}

public class ExeQueryApiObject
{
    public string Query { get; set; }
    public Destination Destination = new Destination();
    public string ExportFormat { get; set; }
}
```

C# class format



```
{
  "Query": "SELECT * FROM tb1",
  "Destination": {
    "Name": "db1",
    "Server": "test1"
  },
  "ExportFormat": "JSON"
}
```

JSON format

รูปที่ 3.13 การแปลง object จาก C# class format เป็น JSON format

เมื่อแปลง object เป็น JSON แล้ว โปรแกรมจะสร้างการเชื่อมต่อพร้อมทั้งส่ง JSON ไปยัง URL ของ API ที่กำหนด เมื่อ API ตอบสนองกลับมา โปรแกรมจะเก็บค่าตอบสนองที่ได้รับไว้ในตัวแปร เพื่อใช้งานต่อไป การทำงานของกระบวนการนี้แสดงได้ ดังรูปที่ 3.14

```

ExeQueryApiObject queryObject = new ExeQueryApiObject();
queryObject.Query = "SELECT * FROM tb_1";
queryObject.Destination.Name = servingDatabaseName;
queryObject.Destination.Server = servingServerName;
queryObject.ExportFormat = "json";
var jsonObject = Newtonsoft.Json.JsonConvert.SerializeObject(queryObject);
var stringContent = new StringContent(jsonObject);

//Create connection
var handler = new HttpClientHandler();
var client = new HttpClient(handler);
handler.UseDefaultCredentials = true;

//Connect to API
var response = await client.PostAsync("https://[redacted]/api/SQL/ExecuteQuery", stringContent);
JsonObject parsedResponse = JsonObject.Parse(await response.Content.ReadAsStringAsync());
JArray messages = JArray.Parse(parsedResponse["Message"].ToString());
foreach (JToken message in messages)
{
    |   currentSession.storeData("User", message);
}

```

รูปที่ 3.14 การทำงานของ code เชื่อมต่อกับระบบอื่น ๆ ผ่าน HTTP Request

3.1.4.4 พัฒนาให้ DBAM Bot รับรู้ข้อมูลของคู่สนทนาได้

เมื่อผู้ใช้งานสนทนากับแชทบอต หากแชทบอตสามารถรับรู้ถึงข้อมูลของคู่สนทนาได้ จะทำให้การทำงานในขั้นตอนบางขั้นตอนทำได้ง่ายและปลอดภัยมากยิ่งขึ้น ซึ่งจะส่งผลให้ผู้ใช้งานได้รับประสบการณ์การใช้งานที่ดี ตัวอย่างการลดขั้นตอนในการทำงาน เช่น แชทบอตสามารถรับรู้ข้อมูลชื่อผู้ใช้งานและข้อมูลคอมพิวเตอร์ที่ผู้ใช้งานกำลังใช้งานอยู่ เพื่อใช้ในการยืนยันตัวตนและเข้าถึงฐานข้อมูลโดยที่ผู้ใช้งานไม่จำเป็นต้องกรอกข้อมูลเพิ่มเติมใด ๆ แชทบอตสามารถแสดงฐานข้อมูลที่เกี่ยวข้องกับผู้ใช้งานได้เป็นต้น

จากการศึกษาการทำงานของระบบในหัวข้อ 3.1.1 ผู้พัฒนาได้พัฒนา code เพิ่มเติมลงไปในส่วนของ method `takeActionOnIntent()` โดยให้ตรวจสอบว่า session ปัจจุบัน เป็น session ใหม่หรือไม่ ถ้าหากใช่แชทบอตจะทำการเรียก method `getUserData()` ดังรูปที่ 3.15 โดย method นี้จะทำการเชื่อมต่อไปยัง DBAM Automation API โดยอาศัย code การเชื่อมต่อจากหัวข้อ 3.1.4.3 เพื่อทำการ query ข้อมูลของผู้ใช้งาน session ปัจจุบันจากฐานข้อมูล และนำมาเก็บไว้ในโปรแกรมแชทบอต ภายใต้ตัวแปรชื่อ `currentSession` ดังรูปที่ 3.16 ภายหลังจากมีจุดใดในโปรแกรมที่ต้องใช้ งานข้อมูลของผู้ใช้งาน โปรแกรมจะสามารถนำข้อมูลในตัวแปรนี้ไปใช้งานได้ทันที

```
getUserData(currentSession);
getUserAppData(currentSession, client);
```

รูปที่ 3.15 การเรียก method getUserData()

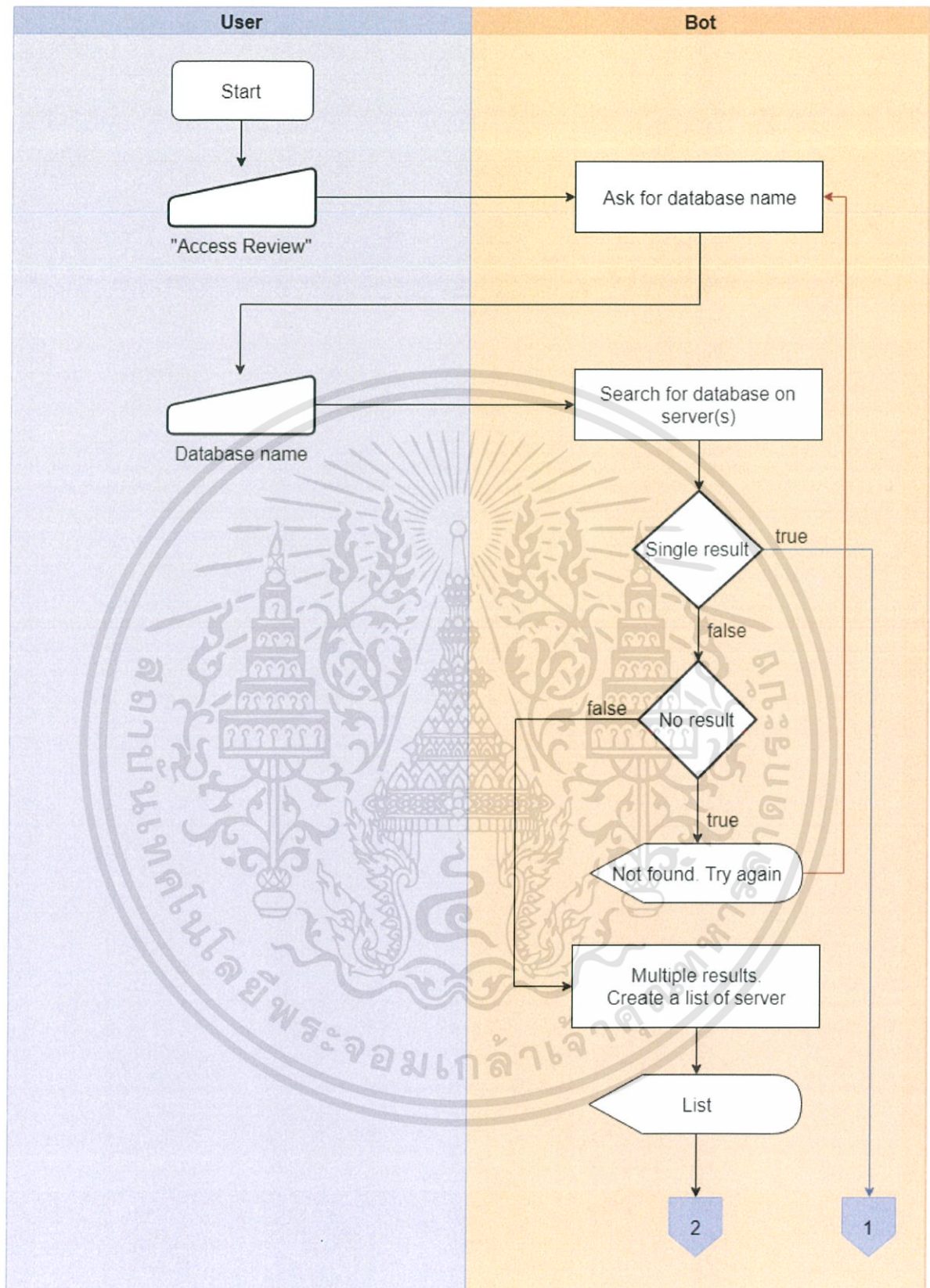
```
try
{
    JObject parsedResponse = JObject.Parse(content);
    JArray messages = JArray.Parse(parsedResponse["Message"].ToString());
    currentSession.storeData("User", messages[0]);
    JObject user = JObject.Parse(messages[0].ToString());
    string userLanId = user["active_directory_domain_code"].ToString() + "\\";
    currentSession.storeData("USERLANID", userLanId);
}
catch (Exception e)
```

รูปที่ 3.16 การเก็บข้อมูลที่ได้จาก DBAM Automation API ลงในตัวแปร currentSession

3.1.4.5 คำสั่ง Access Review

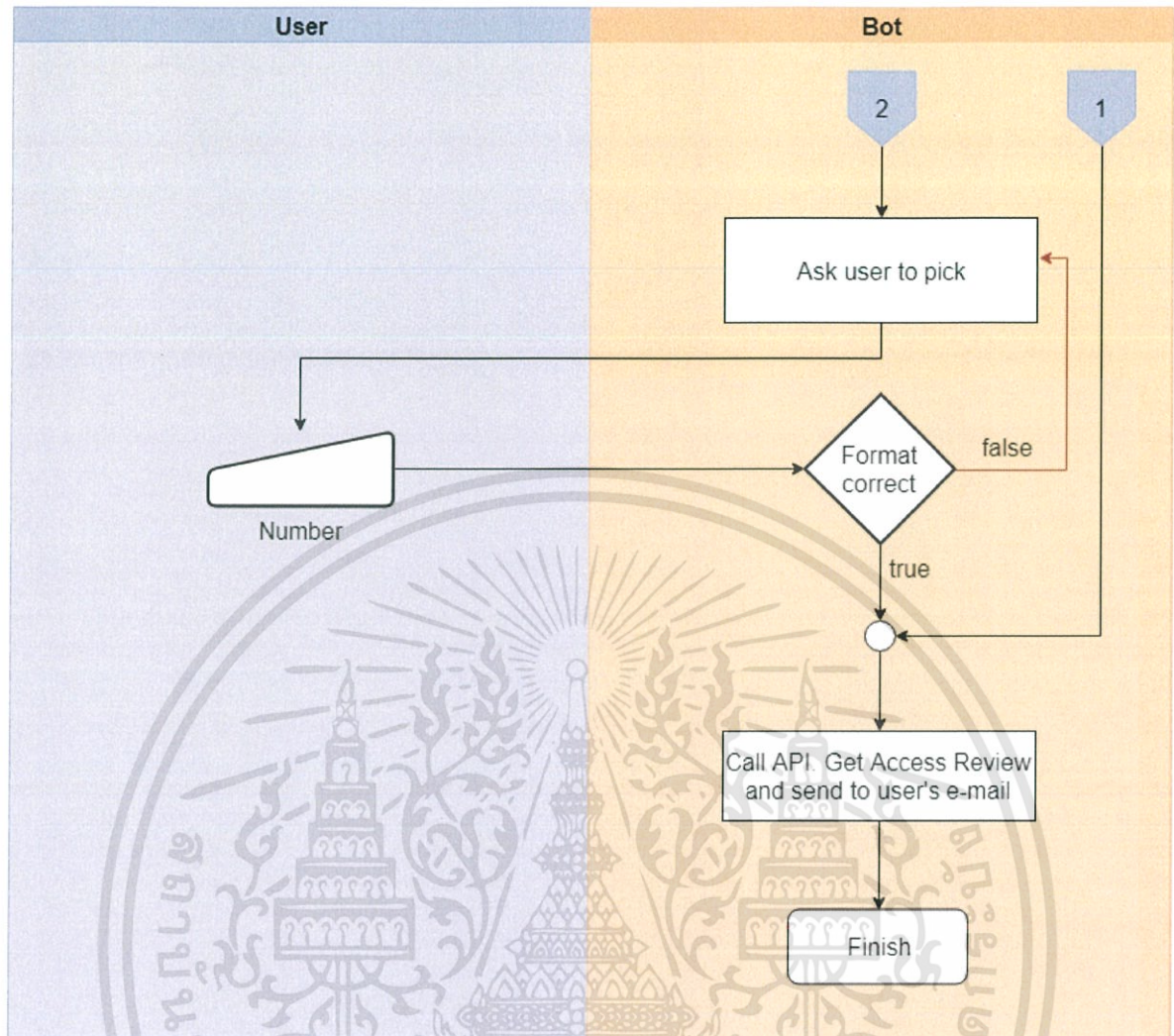
เมื่อผู้ใช้งานต้องการสรุปผลสิทธิ์ของการเข้าถึง (Access Review) ของฐานข้อมูลใด ๆ ผู้ใช้งานจะต้องเปิดเข้าไปที่หน้า Access Review ของ DDR5 และทำการกรอกข้อมูลเพื่อค้นหาโดยใช้ชื่อแอปพลิเคชันหรือชื่อเซิร์ฟเวอร์ โดยไม่สามารถค้นหาโดยใช้ชื่อฐานข้อมูลโดยตรงได้ คำสั่ง Access Review ถูกพัฒนาขึ้นเพื่อให้ผู้ใช้งานสามารถเข้าถึง Access Review ได้ง่ายและไม่จำเป็นต้องเข้าเว็บไซต์อีกต่อไป

สำหรับการทำงานของแชทบอต เมื่อผู้ใช้งานแชทบอตเรียกใช้คำสั่ง Access Review DBAM Bot จะถามชื่อของฐานข้อมูลและชื่อของเซิร์ฟเวอร์ตามลำดับ โดยถ้าหากชื่อของฐานข้อมูลที่ผู้ใช้งานส่งมามีอยู่เพียงหนึ่งเดียวและอยู่บนเซิร์ฟเวอร์เพียงเซิร์ฟเวอร์เดียว ขั้นตอนการถามชื่อเซิร์ฟเวอร์จากแชทบอตจะถูกข้ามไป หากมีมากกว่าหนึ่งแชทบอตจะสร้างตารางและส่งให้ผู้ใช้งานเลือก จากนั้นแชทบอตจะทำการติดต่อ DBAM Automation API เพื่อสร้าง Access Review และส่ง Access Review ไปให้ผู้ใช้งานทาง E-mail โดยจากการพัฒนาให้แชทบอตรับรู้ข้อมูลของคุณสนทนาในหัวข้อที่ 3.1.4.4 แล้ว ผู้ใช้งานไม่จำเป็นต้องระบุ E-mail address เนื่องจากแชทบอตมีข้อมูลพื้นฐานของผู้ใช้งานอยู่แล้ว และสามารถเรียกใช้งานได้ทันที แผนผังการทำงานของคำสั่ง Access Review เป็นไป ดังรูปที่ 3.17 และ รูปที่ 3.18



รูปที่ 3.17 แผนผังรวมการทำงานของคำสั่ง Access Review

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

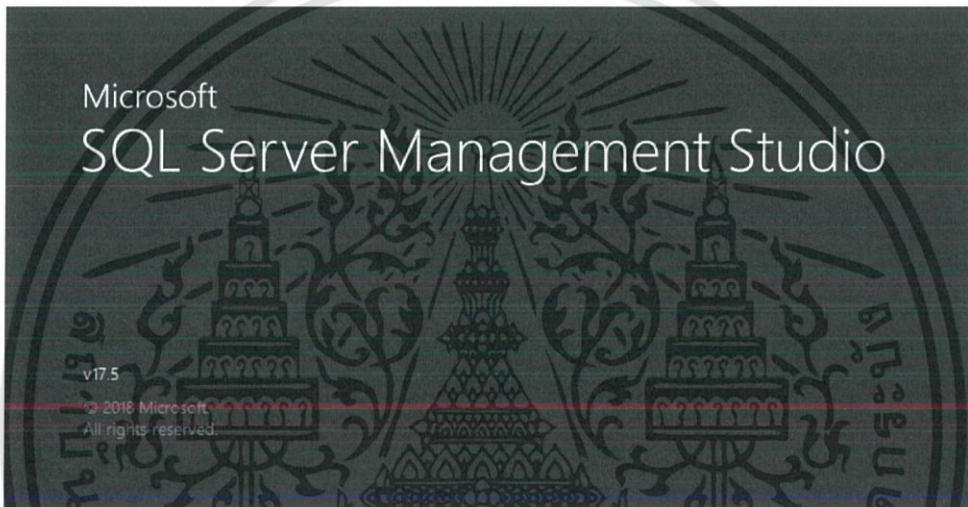


รูปที่ 3.18 แผนผังรวมการทำงานของคำสั่ง Access Review (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4.6 คำสั่ง Backup Database, Restore Database, Refresh Database

โดยปกติแล้วเมื่อผู้ใช้งานต้องการที่จะกระทำการใด ๆ กับฐานข้อมูลของผู้ใช้งาน ผู้ใช้งานจำเป็นต้องติดต่อผู้ดูแลฐานข้อมูลใน DBAM เพื่อให้ผู้ดูแลฐานข้อมูลเป็นผู้กระทำการให้เพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นระหว่างกระบวนการ เนื่องจากมีกระบวนการมีความซับซ้อนและมีความต้องการโปรแกรมเฉพาะด้าน คำสั่งที่เกี่ยวข้องกับฐานข้อมูลคือ Backup Database, Restore Database, Refresh Database จึงถูกพัฒนาขึ้นเพื่อให้ทั้งผู้ใช้งานสามารถกระทำการกับฐานข้อมูลของผู้ใช้งานได้โดยตรง และผู้ดูแลฐานข้อมูลสามารถกระทำการกับฐานข้อมูลได้ง่ายและรวดเร็ว เนื่องจากกระบวนการทั้งหมดในแชตบอตเป็นอัตโนมัติและป้องกันความผิดพลาดไว้แล้ว

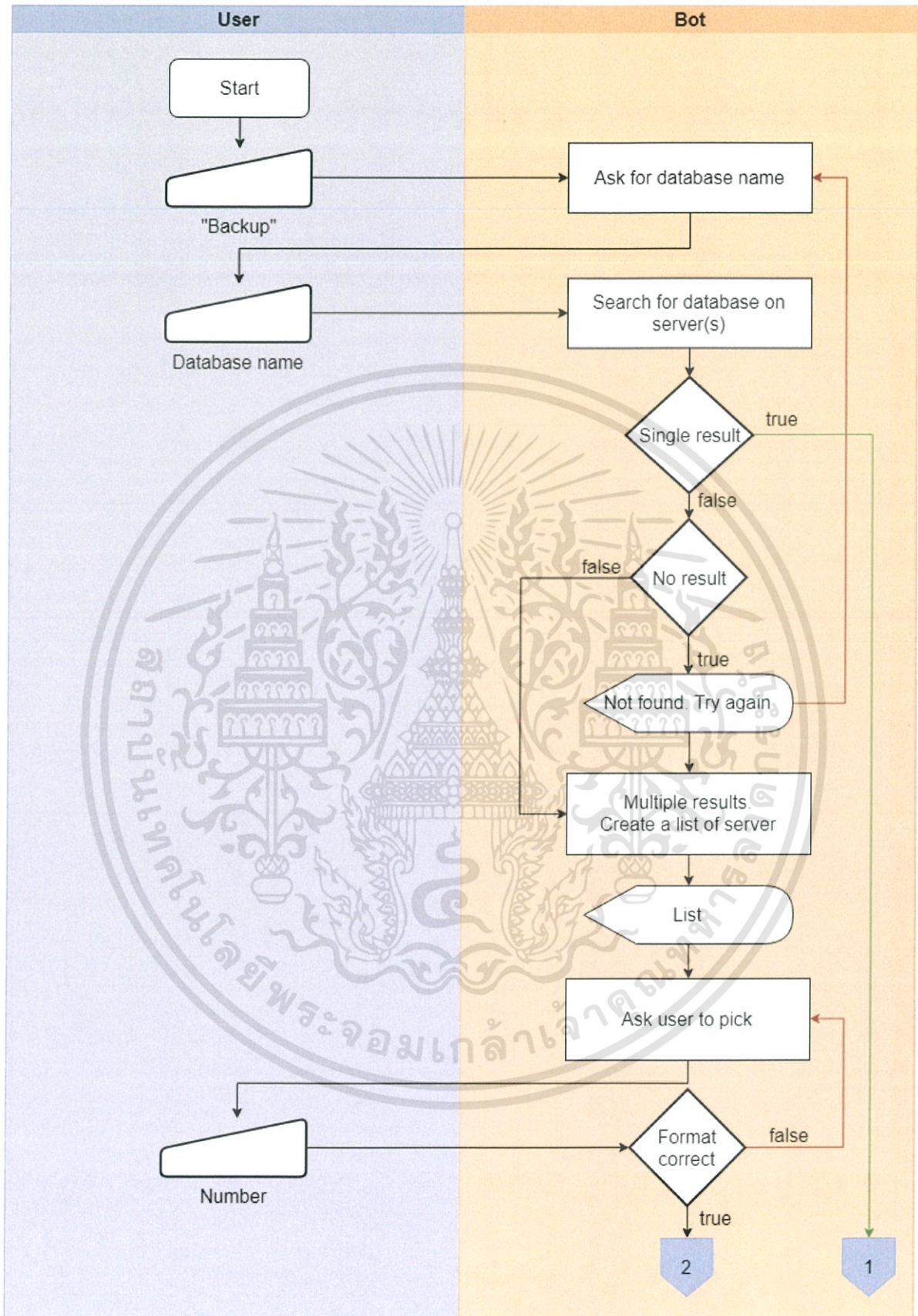


รูปที่ 3.19 โปรแกรม Microsoft SQL Server Management Studio เพื่อการจัดการฐานข้อมูล SQL

การพัฒนาแต่ละคำสั่งที่เกี่ยวข้องกับฐานข้อมูลมีรายละเอียด ดังนี้

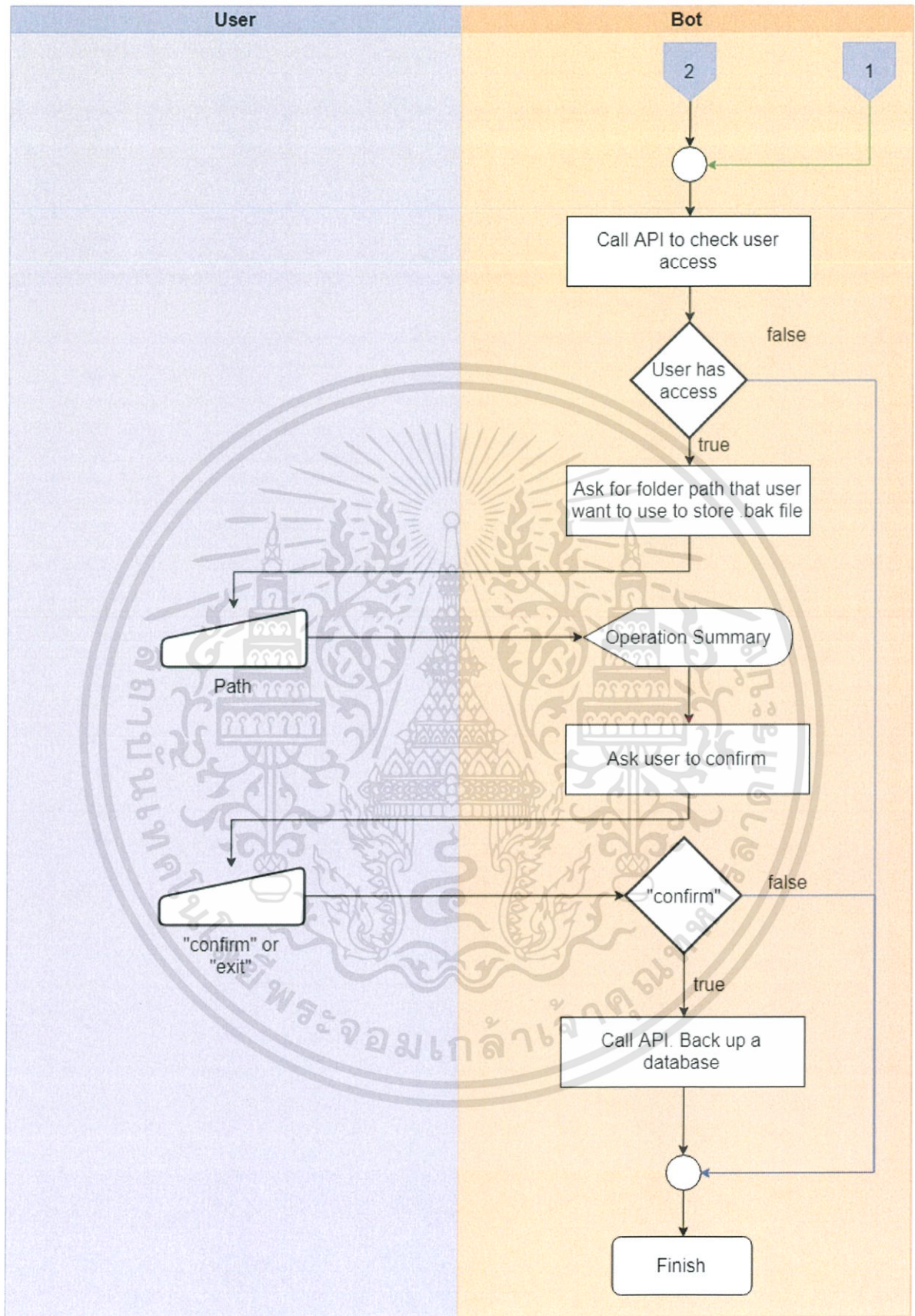
1. คำสั่ง Backup Database

คำสั่ง Backup Database จะทำการ backup ฐานข้อมูลให้ผู้ใช้งาน โดยเมื่อผู้ใช้งานเรียกใช้งานคำสั่ง Backup Database DBAM Bot จะถามชื่อของฐานข้อมูลและชื่อของเซิร์ฟเวอร์ตามลำดับ โดยถ้าหากชื่อของฐานข้อมูลที่ผู้ใช้งานส่งมามีอยู่เพียงหนึ่งเดียวและอยู่บนเซิร์ฟเวอร์เพียงเซิร์ฟเวอร์เดียว ขั้นตอนการถามชื่อเซิร์ฟเวอร์จากแชตบอตจะถูกข้ามไป หากมีมากกว่าหนึ่งแชตบอตจะสร้างตารางและส่งให้ผู้ใช้งานเลือก จากนั้นแชตบอตจะทำการเรียก DBAM Automation API เพื่อทำการเช็กสิทธิ์การเข้าถึงของผู้ใช้งานโดยอาศัยข้อมูลผู้ใช้งานที่แชตบอต ระบุในหัวข้อที่ 3.1.4.4 หากผู้ใช้งานมีสิทธิ์เข้าถึงแชตบอตจะสอบถามตำแหน่งที่ผู้ใช้งานต้องการใช้เก็บไฟล์ backup เมื่อผู้ใช้งานระบุตำแหน่ง แชตบอตจะส่งข้อมูลสรุปการทำงานและให้ผู้ใช้งานให้ยืนยันความถูกต้องของข้อมูล เมื่อผู้ใช้งานยืนยัน แชตบอตจะติดต่อ DBAM Automation API เพื่อ backup ฐานข้อมูล แสดงผลลัพธ์ให้ผู้ใช้งานและเสร็จสิ้นกระบวนการ แผนผังการทำงานของคำสั่ง Backup Database เป็นไป ดังรูปที่ 3.20 และ รูปที่ 3.21



รูปที่ 3.20 แผนผังรวมการทำงานของคำสั่ง Backup Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 แผนผังรวมการทำงานของคำสั่ง Backup Database (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

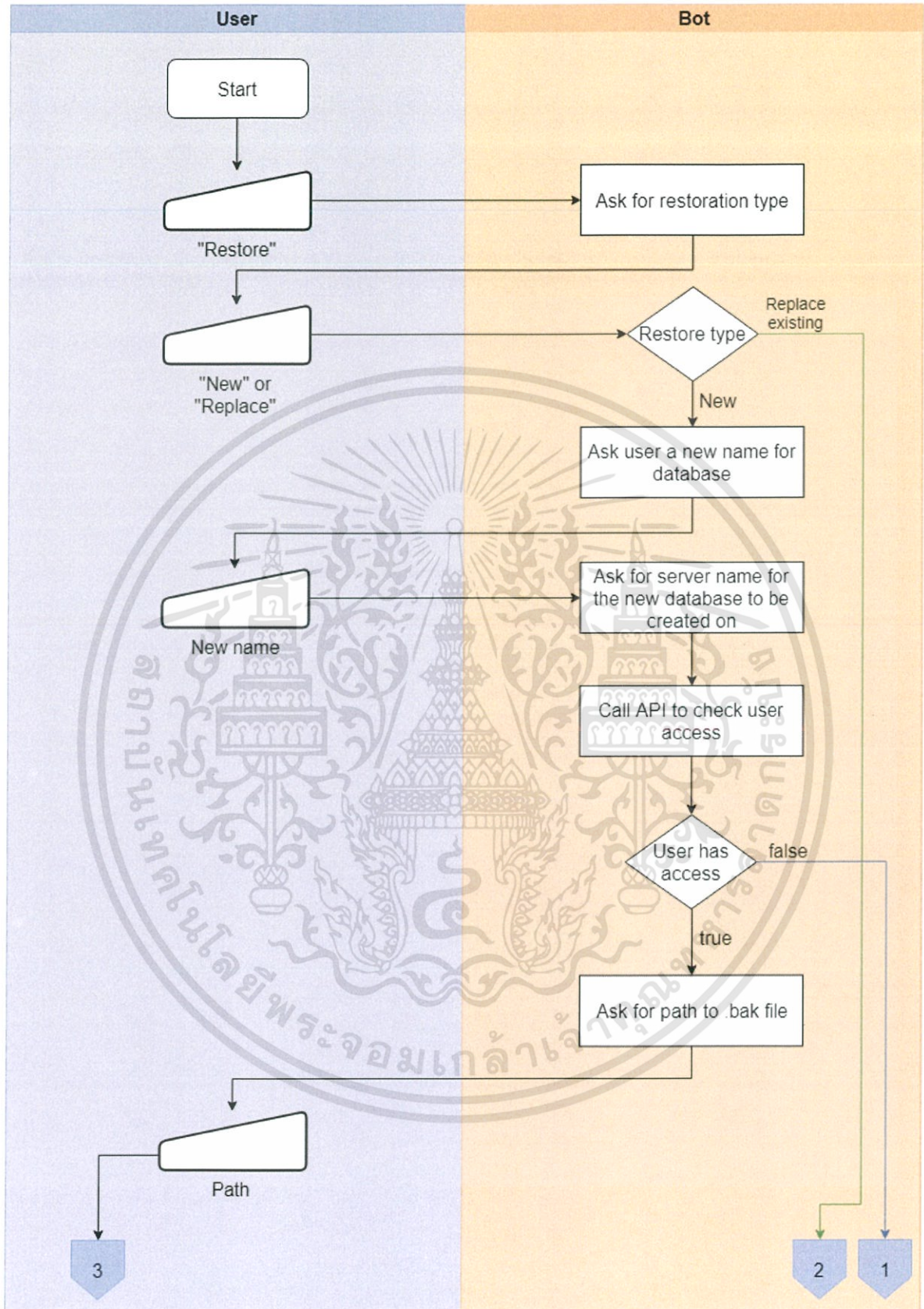
2. คำสั่ง Restore Database

คำสั่ง Restore Database จะทำการคืนค่าข้อมูลในฐานข้อมูลให้ผู้ใช้งานโดยอาศัยข้อมูลจากไฟล์ backup โดยเมื่อผู้ใช้งานเรียกใช้งานคำสั่ง Restore Database DBAM Bot จะถามว่าผู้ใช้งานต้องการจะคืนค่าฐานข้อมูลที่มีอยู่แล้วหรือสร้างฐานข้อมูลขึ้นใหม่และคืนค่าลงในฐานข้อมูลนั้นโดย

2.1 หากผู้ใช้งานเลือกคืนค่าฐานข้อมูลที่มีอยู่แล้ว แค็ตบอตจะถามชื่อไฟล์ backup และตำแหน่งของไฟล์ จากนั้นแค็ตบอตจะทำการประมวลผลหาชื่อฐานข้อมูลและเซิร์ฟเวอร์จากชื่อไฟล์โดยอัตโนมัติและทำการเรียก DBAM Automation API เพื่อทำการเช็คสิทธิ์การเข้าถึงของผู้ใช้งานโดยอาศัยข้อมูลผู้ใช้งานที่แค็ตบอตรับรู้ในหัวข้อที่ 3.1.4.4 หากผู้ใช้งานมีสิทธิ์เข้าถึง แค็ตบอตจะสอบถามว่าผู้ใช้งานมี ITSM Ticket ที่เกี่ยวข้องหรือไม่ หากมี Ticket ให้ผู้ใช้งานระบุหมายเลข Ticket และแค็ตบอตจะทำการติดต่อกับ ITSM API เพื่อตรวจสอบความถูกต้องของ Ticket หากไม่มี ticket แค็ตบอตจะทำการติดต่อกับ ITSM API เพื่อสร้าง Ticket ประเภท Work Order ให้โดยอัตโนมัติพร้อมระบุรายละเอียดลงใน Ticket จากนั้นแค็ตบอตจะสอบถามว่าผู้ใช้งานต้องการเก็บการตั้งค่าสิทธิ์ผู้เข้าถึงฐานข้อมูลที่มีอยู่แล้วไว้คงเดิมหรือไม่ จากนั้นแค็ตบอตจะส่งข้อมูลสรุปการทำงานและให้ผู้ใช้งานให้ยืนยันความถูกต้องของข้อมูล เมื่อผู้ใช้งานยืนยัน แค็ตบอตจะติดต่อ DBAM Automation API เพื่อ restore ข้อมูลสู่ฐานข้อมูล แสดงผลลัพธ์ให้ผู้ใช้งานและเสร็จสิ้นกระบวนการ

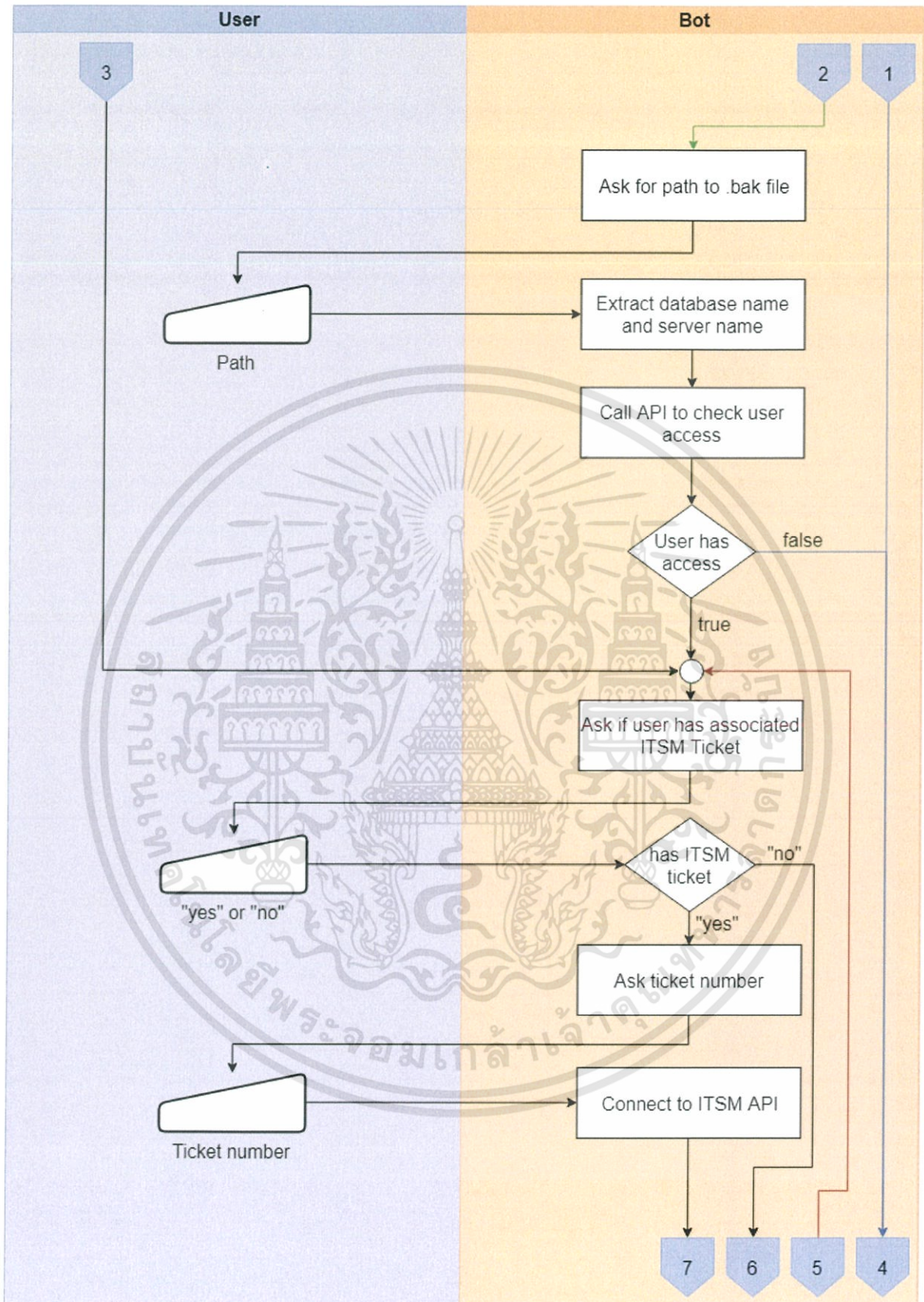
2.2 หากผู้ใช้งานเลือกสร้างฐานข้อมูลขึ้นใหม่และคืนค่าลงในฐานข้อมูลนั้น แค็ตบอตจะถามชื่อของฐานข้อมูลใหม่และชื่อของเซิร์ฟเวอร์ตามลำดับ จากนั้นแค็ตบอตจะทำการเรียก DBAM Automation API เพื่อทำการเช็คสิทธิ์การเข้าถึงของผู้ใช้งานโดยอาศัยข้อมูลผู้ใช้งานที่แค็ตบอต รู้ในหัวข้อที่ 3.1.4.4 หากผู้ใช้งานมีสิทธิ์เข้าถึงแค็ตบอตจะถามชื่อไฟล์ backup และตำแหน่งของไฟล์ เมื่อผู้ใช้งานระบุตำแหน่ง แค็ตบอตจะสอบถามว่าผู้ใช้งานมี ITSM Ticket ที่เกี่ยวข้องหรือไม่ หากมี Ticket ให้ผู้ใช้งานระบุหมายเลข Ticket และแค็ตบอตจะทำการติดต่อกับ ITSM API เพื่อตรวจสอบความถูกต้องของ Ticket หากไม่มี ticket แค็ตบอตจะทำการติดต่อกับ ITSM API เพื่อสร้าง Ticket ประเภท Work Order ให้โดยอัตโนมัติพร้อมระบุรายละเอียดลงใน Ticket จากนั้นแค็ตบอตจะส่งข้อมูลสรุปการทำงานและให้ผู้ใช้งานให้ยืนยันความถูกต้องของข้อมูล เมื่อผู้ใช้งานยืนยัน แค็ตบอตจะติดต่อ DBAM Automation API เพื่อสร้างฐานข้อมูลใหม่และคืนค่า แสดงผลลัพธ์ให้ผู้ใช้งานและเสร็จสิ้นกระบวนการ

แผนผังการทำงานของคำสั่ง Restore Database ทั้ง 2 รูปแบบคือคืนค่าสู่ฐานข้อมูลที่มีอยู่แล้ว และสร้างฐานข้อมูลขึ้นใหม่และคืนค่าลงในฐานข้อมูลนั้น เป็นไป ดัง รูปที่ 3.22 ถึงรูปที่ 3.25



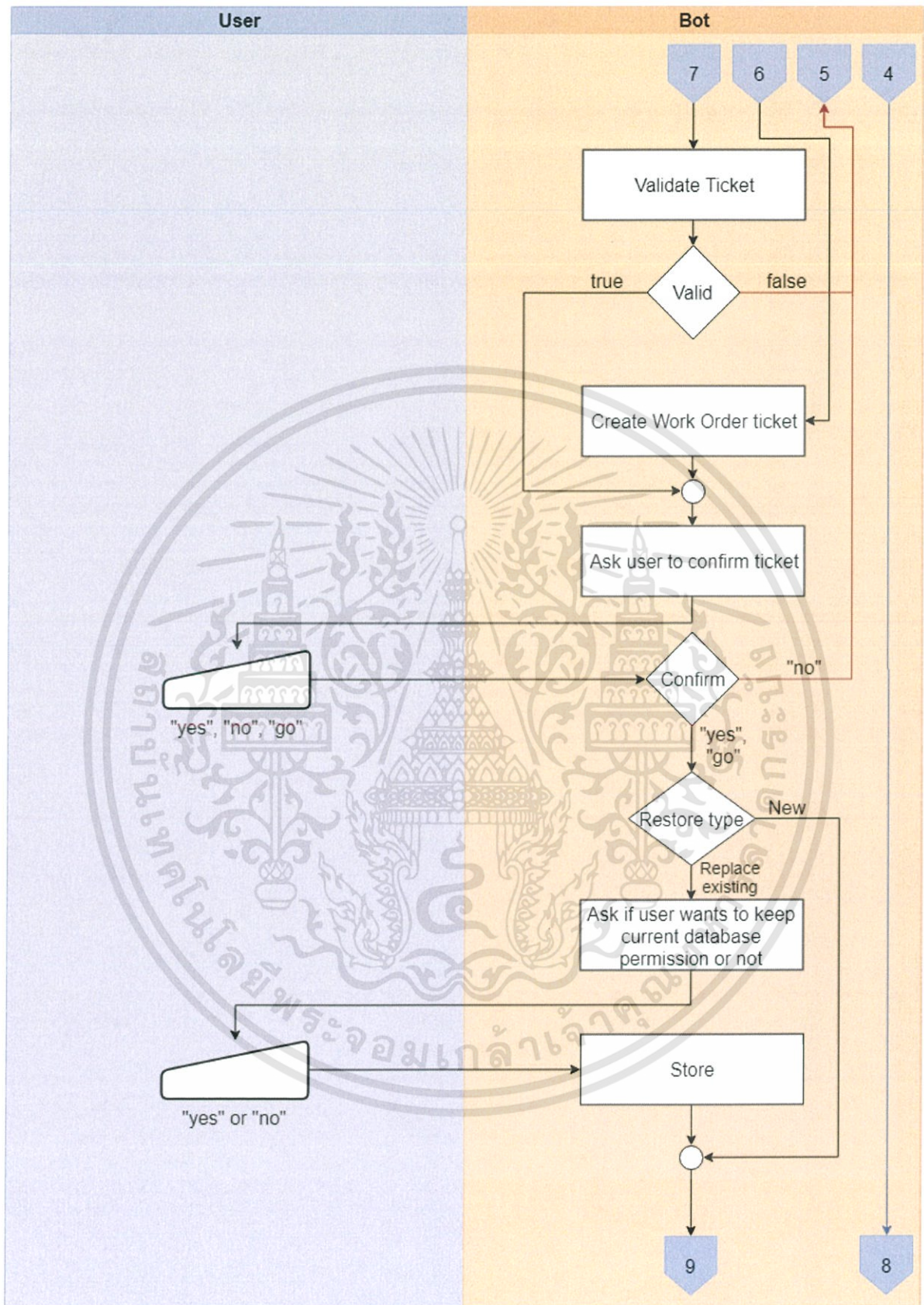
รูปที่ 3.22 แผนผังรวมการทำงานของคำสั่ง Restore Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อภา39ษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



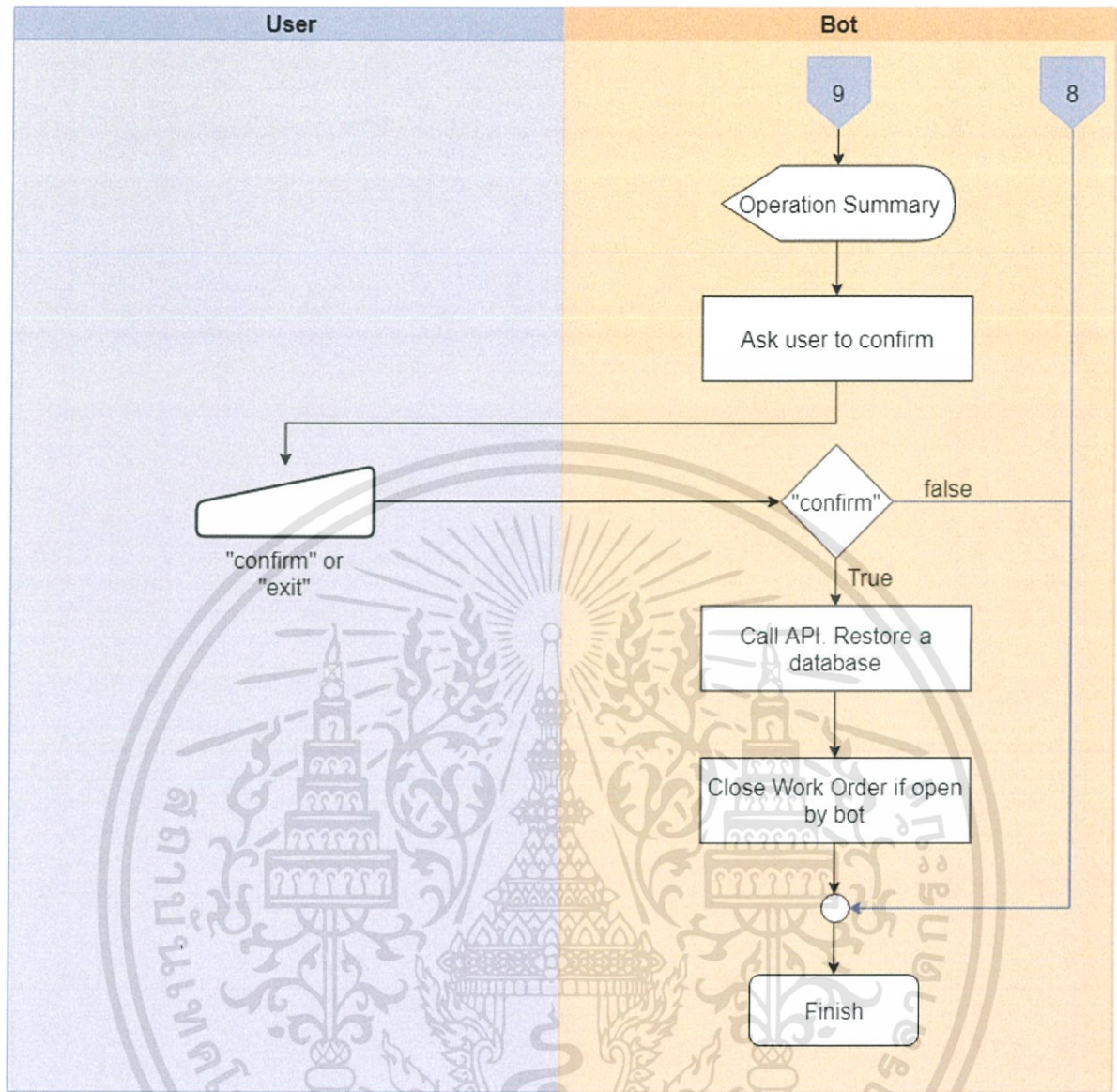
รูปที่ 3.23 แผนผังรวมการทำงานของคำสั่ง Restore Database (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 แผนผังรวมการทำงานของคำสั่ง Restore Database (3)

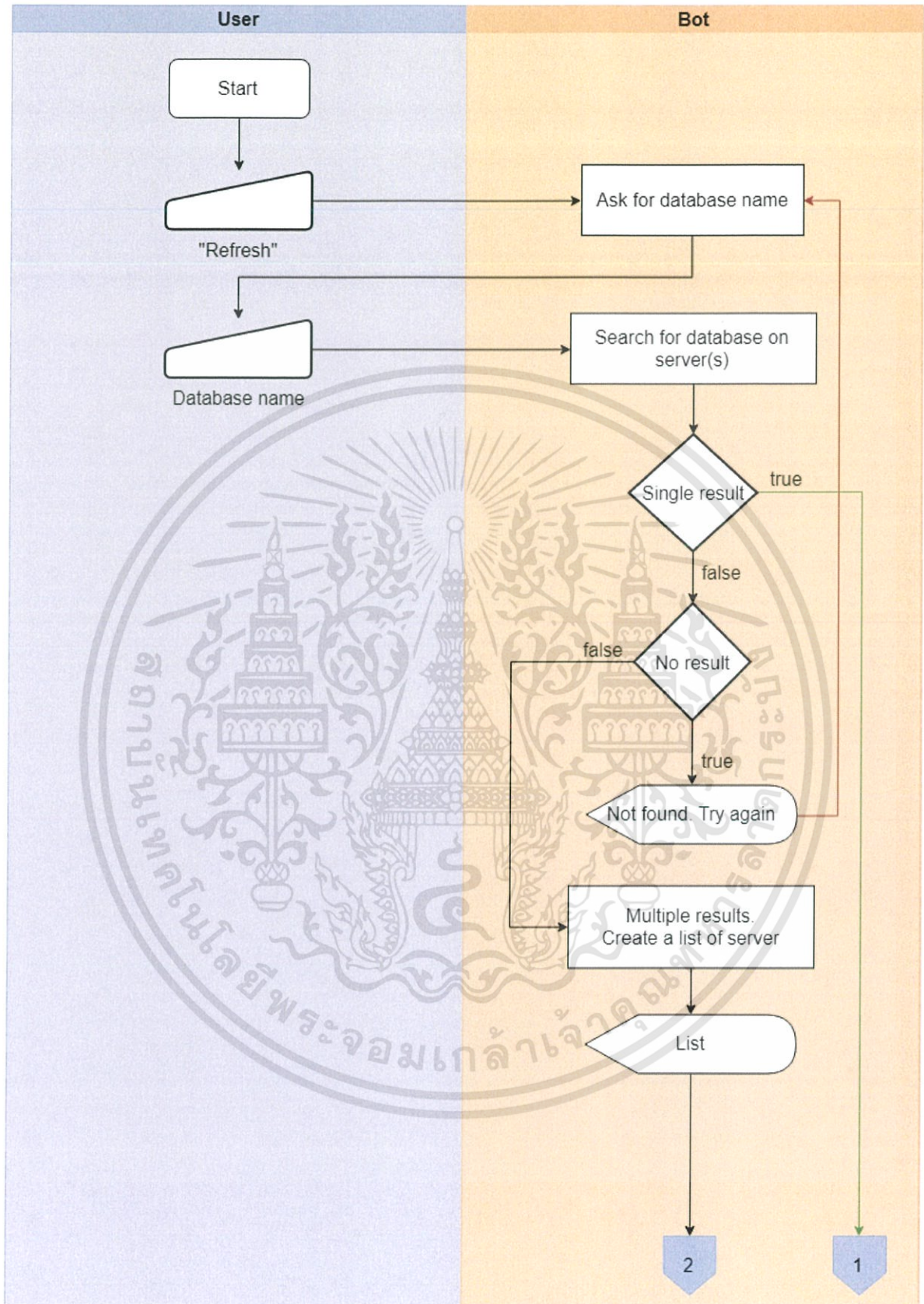
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.25 แผนผังรวมการทำงานของคำสั่ง Restore Database (4)

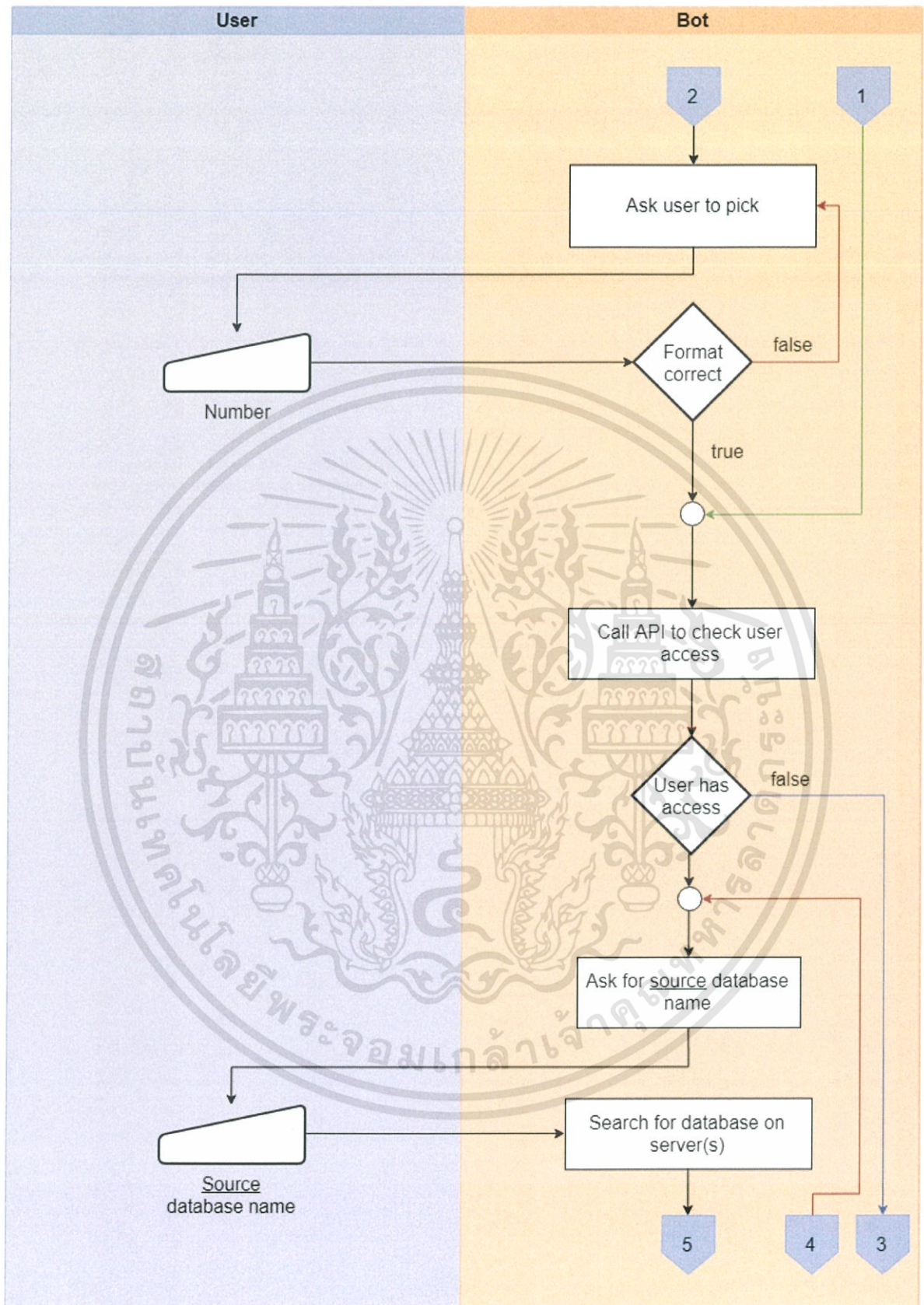
3. คำสั่ง Refresh Database

คำสั่ง Refresh Database จะทำการคืนค่าข้อมูลในฐานข้อมูลให้ผู้ใช้งานโดยอาศัยข้อมูลฐานข้อมูลอื่น โดยเมื่อผู้ใช้งานเรียกใช้งานคำสั่ง Refresh Database DBAM Bot จะถามชื่อของฐานข้อมูลที่ต้องการ Refresh และชื่อของเซิร์ฟเวอร์ตามลำดับ โดยถ้าหากชื่อของฐานข้อมูลที่ใช้ส่งมามีอยู่เพียงหนึ่งเดียวและอยู่บนเซิร์ฟเวอร์เพียงเซิร์ฟเวอร์เดียว ขั้นตอนการถามชื่อเซิร์ฟเวอร์จากแช็ตบอตจะถูกข้ามไป หากมีมากกว่าหนึ่งแช็ตบอตจะสร้างตารางและส่งให้ผู้ใช้งานเลือก จากนั้นแช็ตบอตจะทำการเรียก DBAM Automation API เพื่อทำการเช็คสิทธิ์การเข้าถึงของผู้ใช้งานโดยอาศัยข้อมูลผู้ใช้งานที่แช็ตบอต รับรู้ในหัวข้อที่ 3.1.4.4 หากผู้ใช้งานมีสิทธิ์เข้าถึง จากนั้นแช็ตบอตจะถามชื่อของฐานข้อมูลต้นทางและชื่อของเซิร์ฟเวอร์ตามลำดับ โดยถ้าหากชื่อของฐานข้อมูลที่ใช้ส่งมามีอยู่เพียงหนึ่งเดียวและอยู่บนเซิร์ฟเวอร์เพียงเซิร์ฟเวอร์เดียว ขั้นตอนการถามชื่อเซิร์ฟเวอร์จากแช็ตบอตจะถูกข้ามไป หากมีมากกว่าหนึ่งแช็ตบอตจะสร้างตารางและส่งให้ผู้ใช้งานเลือก จากนั้นแช็ตบอตจะทำการเรียก DBAM Automation API เพื่อทำการเช็คสิทธิ์การเข้าถึงของผู้ใช้งาน หากผู้ใช้งานมีสิทธิ์เข้าถึงแช็ตบอตจะสอบถามว่าผู้ใช้งานมี ITSM Ticket ที่เกี่ยวข้องหรือไม่ หากมี Ticket ให้ผู้ใช้งานระบุหมายเลข Ticket และแช็ตบอตจะทำการติดต่อกับ ITSM API เพื่อตรวจสอบความถูกต้องของ Ticket หากไม่มี ticket แช็ตบอตจะทำการติดต่อกับ ITSM API เพื่อสร้าง Ticket ประเภท Work Order ให้โดยอัตโนมัติพร้อมระบุรายละเอียดลงใน Ticket จากนั้นแช็ตบอตจะส่งข้อมูลสรุปการทำงานและให้ผู้ใช้งานให้ยืนยันความถูกต้องของข้อมูล เมื่อผู้ใช้งานยืนยัน แช็ตบอตจะติดต่อ DBAM Automation API เพื่อ Refresh ฐานข้อมูล แสดงผลลัพธ์ให้ผู้ใช้งานและเสร็จสิ้นกระบวนการ แผนผังการทำงานของคำสั่ง Refresh Database เป็นไป ดังรูปที่ 3.26 ถึงรูปที่ 3.31



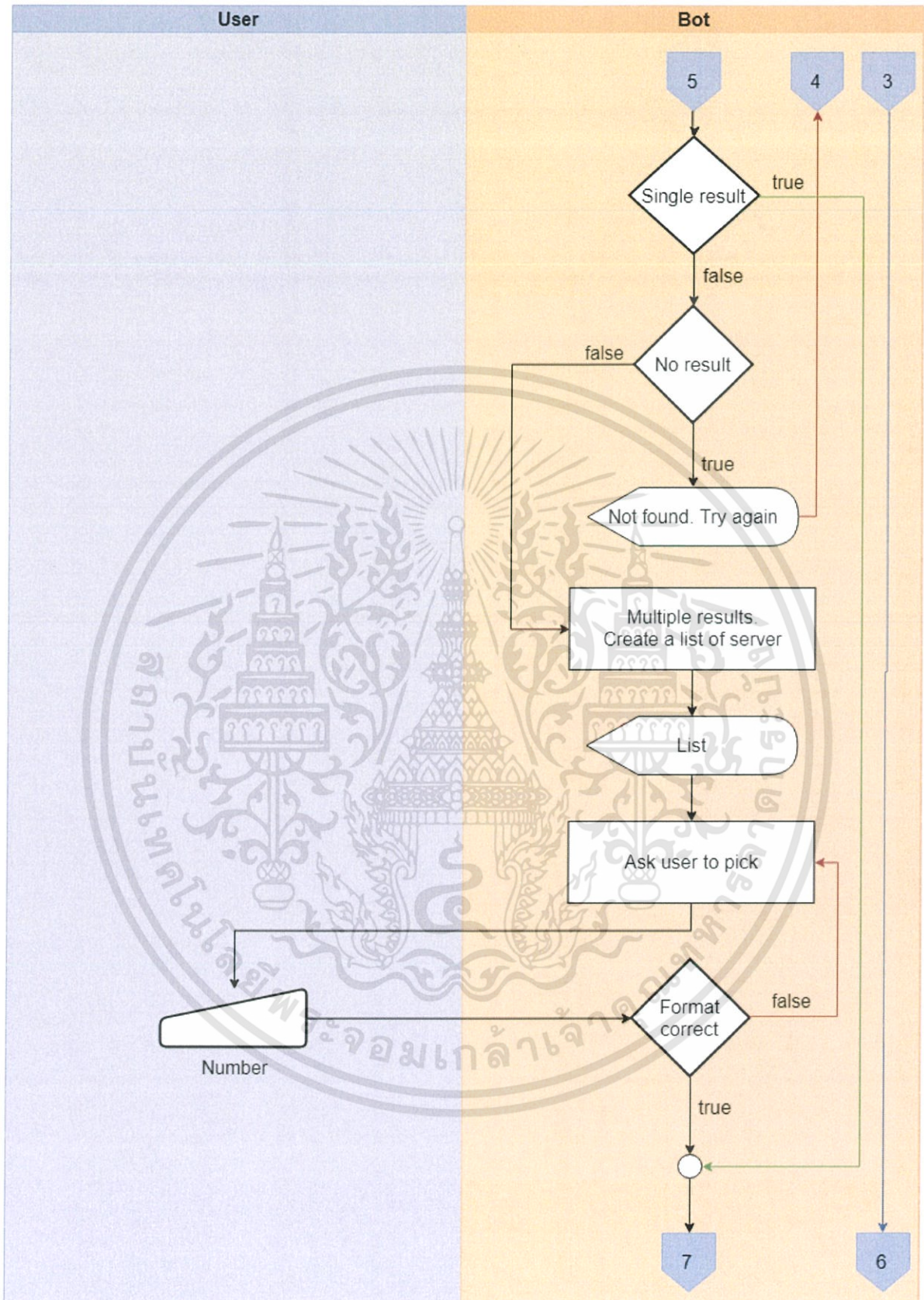
รูปที่ 3.26 แผนผังรวมการทำงานของคำสั่ง Refresh Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



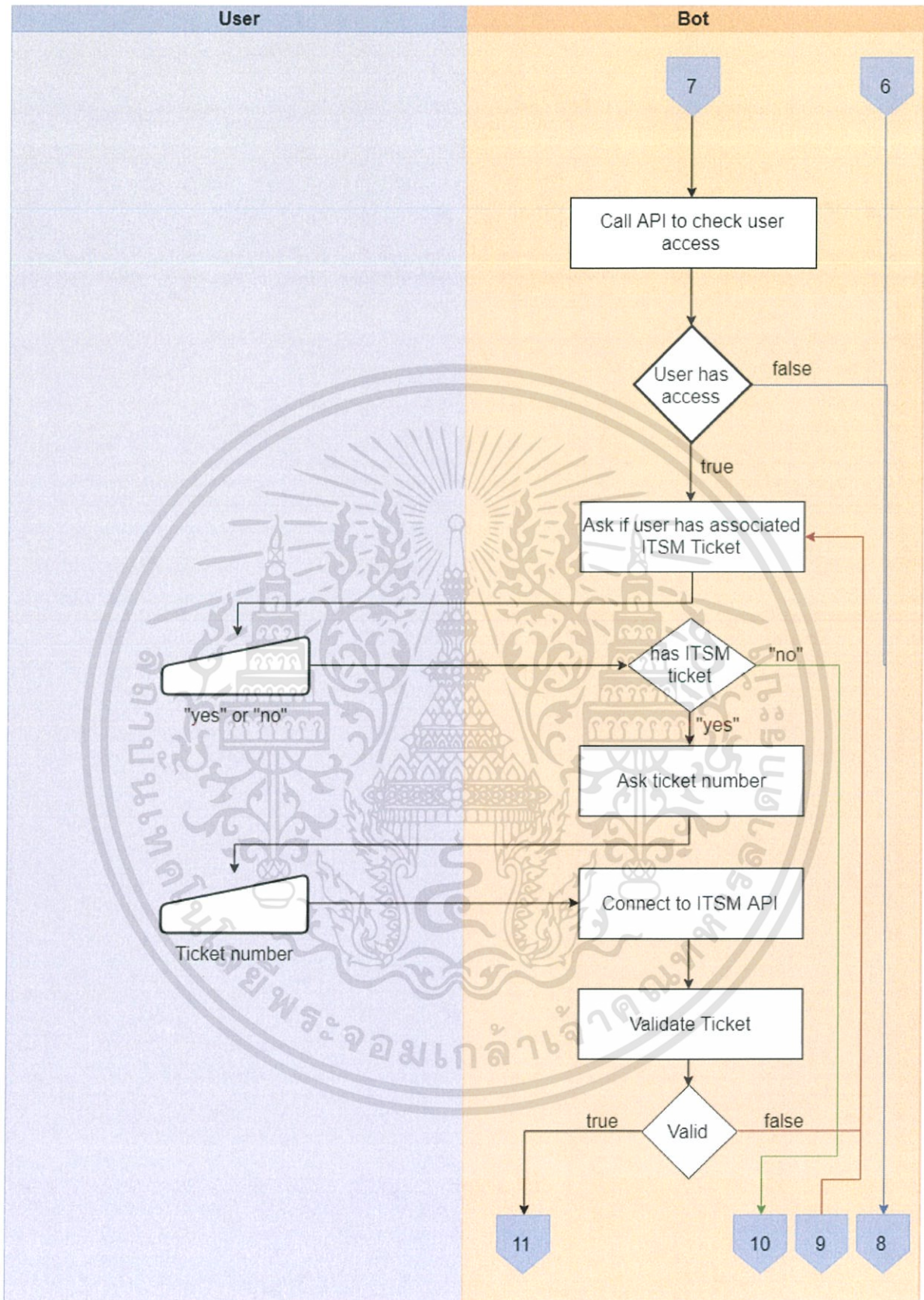
รูปที่ 3.27 แผนผังรวมการทำงานของคำสั่ง Refresh Database (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



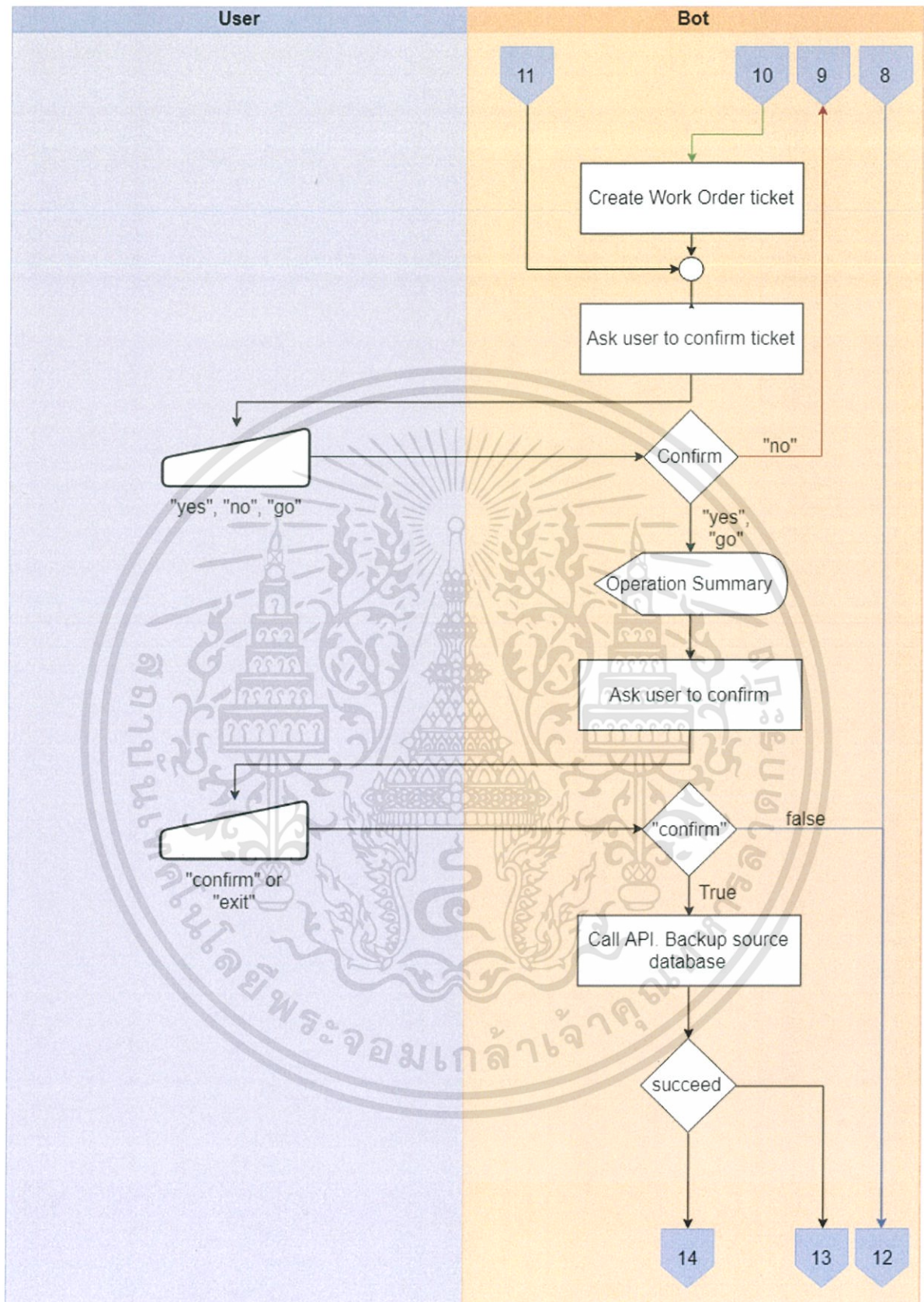
รูปที่ 3.28 แผนผังรวมการทำงานของคำสั่ง Refresh Database (3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



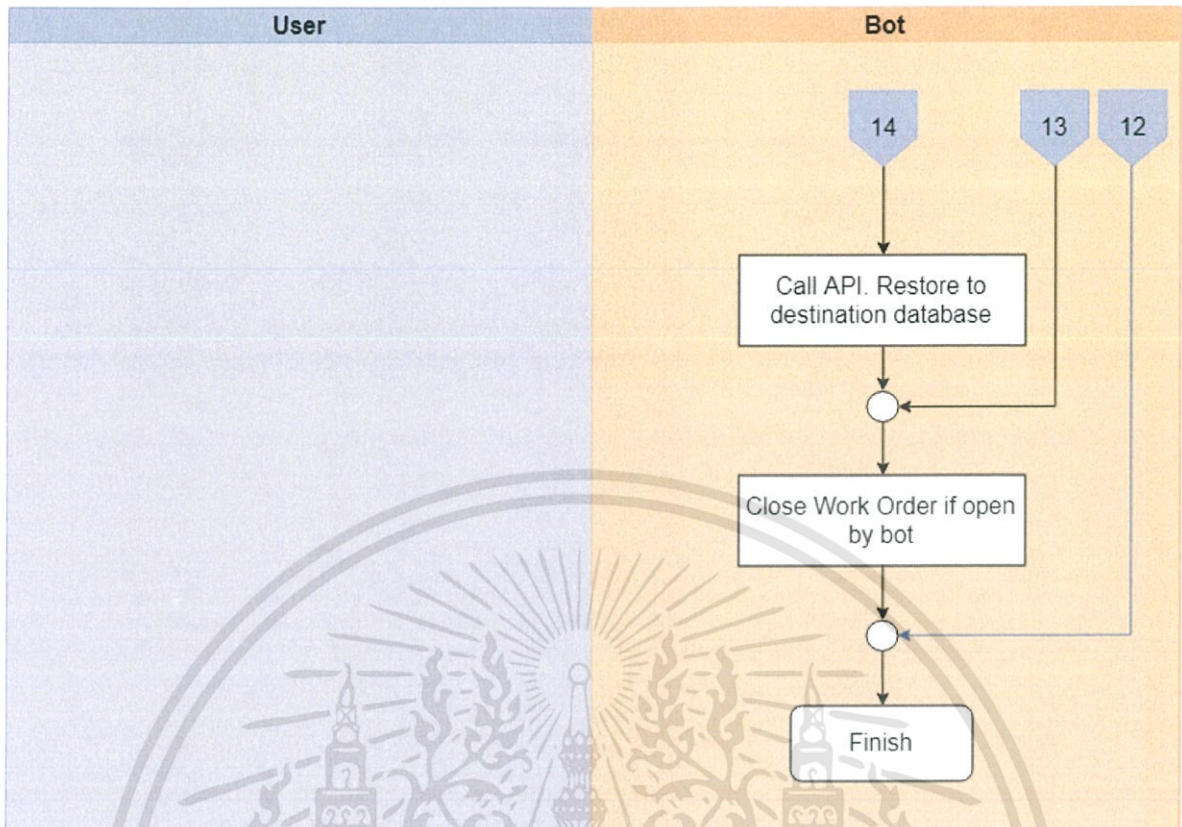
รูปที่ 3.29 แผนผังรวมการทำงานของคำสั่ง Refresh Database (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.30 แผนผังรวมการทำงานของคำสั่ง Refresh Database (5)

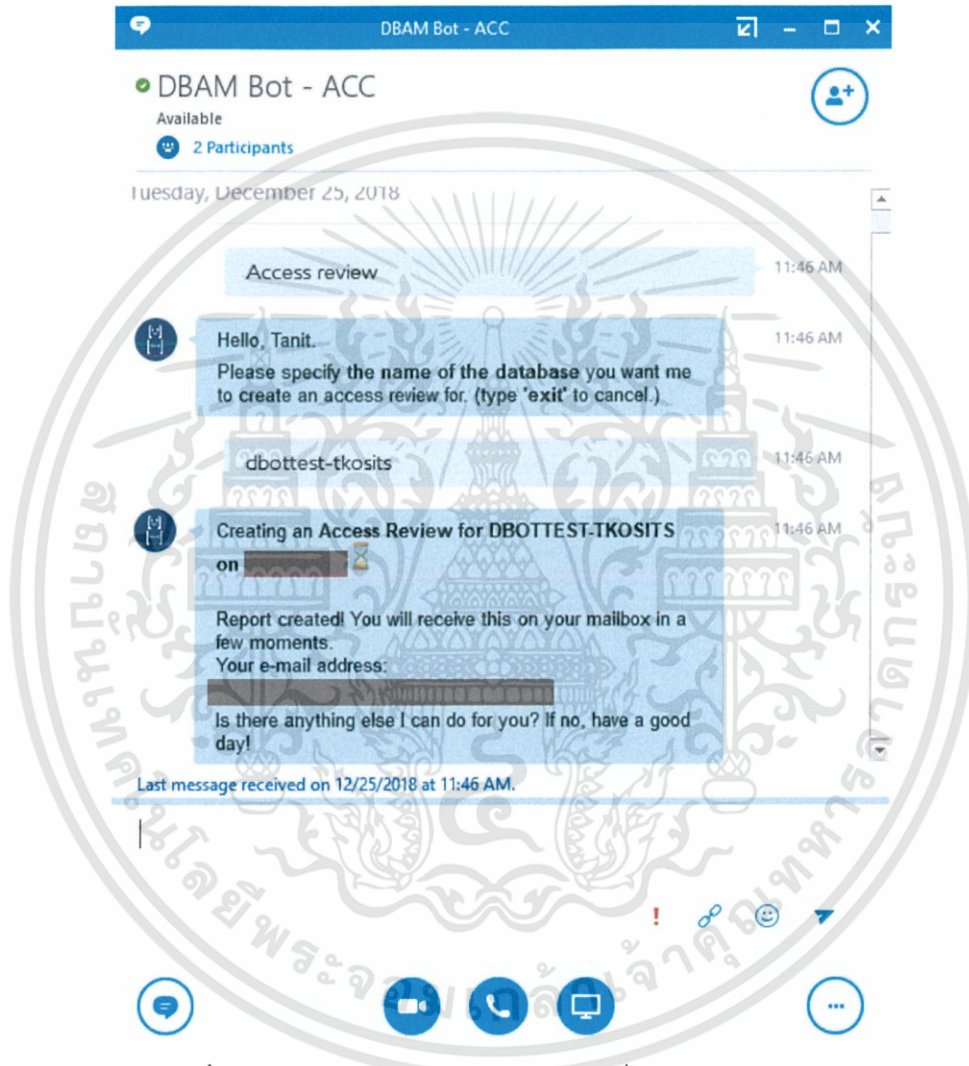
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.31 แผนผังรวมการทำงานของคำสั่ง Refresh Database (6)

3.1.5 ทดสอบการทำงานของระบบ

ผู้พัฒนาได้ทำการทดสอบการใช้งานของโปรแกรมตลอดช่วงระยะเวลาของการพัฒนา โดยการทดสอบด้วยตนเอง ทดสอบพร้อมกับเพื่อนร่วมงาน และให้พนักงานภายในแผนกรวมถึงพนักงานที่เกี่ยวข้องกับฐานข้อมูลช่วยทดสอบ โดยผลจากการทดสอบในแต่ละครั้ง ผู้พัฒนาจะนำผลการทดสอบมาวิเคราะห์เพื่อพัฒนาและปรับปรุงการทำงานของแช็ตบอตต่อไป



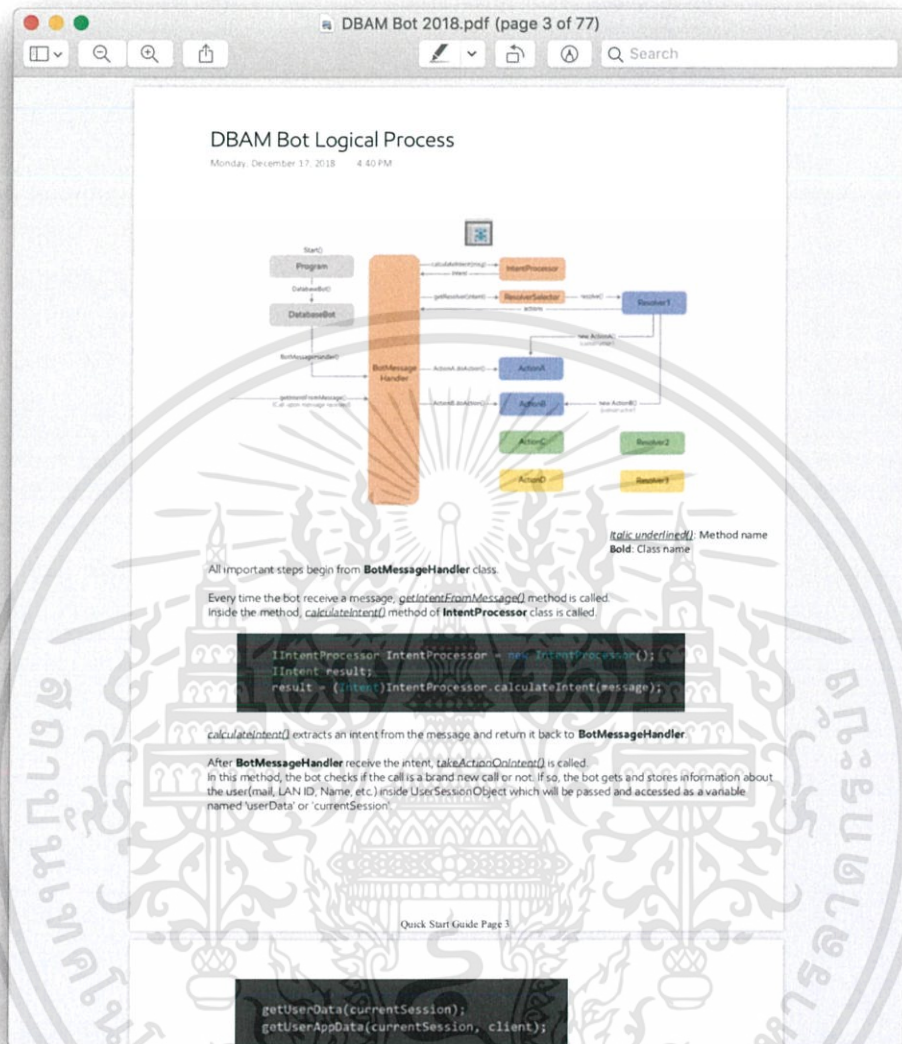
รูปที่ 3.32 การทดสอบการทำงานของคำสั่ง Access Review

3.1.6 นำระบบไปใช้งานจริง

หลังจากการทดสอบการทำงานตามคำสั่งต่างแล้ว ได้รับผลลัพธ์ที่น่าพึงพอใจ ผู้พัฒนาและเพื่อนร่วมงานได้ทำการปรึกษากับพนักงานที่ปรึกษาและเพื่อนร่วมงานต่างประเทศ และได้นำแช็ตบอตขึ้นทำงานเป็นเซิร์ฟวิสนบนเครื่องเซิร์ฟเวอร์เพื่อให้แช็ตบอตสามารถออนไลน์และพร้อมใช้งานตลอดเวลา นอกจากนี้ผู้พัฒนายังได้เขียนเอกสารข้อมูลเกี่ยวกับการพัฒนาแช็ตบอต DBAM Bot เพื่อส่งต่อให้ผู้พัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คนต่อไป ดังรูปที่ 3.33 และทำ Knowledge Transfer เพื่อส่งต่อความรู้ให้เพื่อนร่วมงานต่างประเทศที่รับผิดชอบดูแลงานนี้ต่อไป



รูปที่ 3.33 ตัวอย่างเอกสารข้อมูลเกี่ยวกับการพัฒนาแชทบอท DBAM Bot

3.2 ขั้นตอนการดำเนินงานพัฒนาระบบอัตโนมัติสำหรับล็อกเกอร์อัจฉริยะ

ล็อกเกอร์อัจฉริยะ (Smart Locker) เป็นโครงการที่พัฒนาขึ้น เนื่องจากกลุ่มผู้พัฒนาได้เล็งเห็นถึงปัญหาของระบบการยืมคืนสิ่งของภายในบริษัท ที่ยังคงเป็นระบบ manual อาศัยการทำงานของมนุษย์ในเกือบทุกขั้นตอน ส่งผลให้การยืมคืนอุปกรณ์มีความล่าช้า อุปกรณ์ตกหล่น สูญหาย และยังเป็นการเพิ่มภาระงานให้ผู้ดูแลระบบยืมคืนซึ่งมีภาระหน้าที่ประจำอยู่แล้ว ไม่เป็นผลดีแก่ทั้งผู้ดูแล พนักงานผู้ต้องการยืมอุปกรณ์ และบริษัท กลุ่มของผู้พัฒนาจึงได้คิดค้นโครงการนี้ขึ้น และนำเสนอโครงการภายในกิจกรรม EMIT Bootcamp 2018 ที่จัดขึ้นโดยบริษัทเพื่อสนับสนุนความคิดใหม่ ๆ ที่เป็นประโยชน์ต่อบริษัท ทั้งนี้เพื่อให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานร่วมกันภายในกลุ่มเป็นไปได้อย่างมีประสิทธิภาพ จึงได้มีการจัดการประชุมเป็นประจำ หนึ่งครั้งต่อสัปดาห์ เพื่อติดตามความคืบหน้า วางแผน และร่วมกันแก้ไขปัญหา โดยแผนการทำงานพัฒนา สามารถแสดงได้ ดังตารางที่ 3.3

ตารางที่ 3.3 แผนการทำงานพัฒนาระบบล็อกเกอร์อัจฉริยะในระยะที่ 1

งาน \ ระยะเวลา	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
ประชุมงาน	1 ครั้ง/สัปดาห์					
ศึกษาความต้องการ						
วางแผนการพัฒนา						
พัฒนาระบบ						
ทดสอบการทำงาน						

3.2.1 ศึกษาความต้องการต่าง ๆ

ในขั้นตอนการศึกษาความต้องการ กลุ่มของผู้พัฒนาได้ทำการรวบรวมข้อมูลจากผู้ใช้งาน ผู้ดูแลระบบ และหน่วยงานต่าง ๆ รวมถึงศึกษาความเป็นไปได้ในหลาย ๆ แนวทาง เพื่อค้นหาวิธีที่จะตอบโจทยมากที่สุด โดยกลุ่มผู้พัฒนาสรุปความต้องการขั้นพื้นฐานสำหรับระบบที่จะพัฒนาขึ้น ดังนี้

3.2.1.1 ต้องเป็นอัตโนมัติ

3.2.1.2 ลดความเกี่ยวข้องของมนุษย์ระหว่างกระบวนการยืมคืนได้อย่างมีนัยยะ

3.2.1.3 มีการเก็บข้อมูลการยืมคืน ตลอดจนมีระบบติดตามอุปกรณ์ที่ยืมเกินเวลา

3.2.1.4 สามารถนับและตรวจเช็คสถานะสิ่งของได้จากหน้าเว็บไซต์ เพื่อลดโอกาสการ

มายืมของแล้วไม่มีอุปกรณ์ให้ยืม รวมถึงเพื่อให้ผู้ดูแลระบบสามารถบริหารจัดการอุปกรณ์ได้

3.2.1.5 ให้ประสบการณ์การใช้งานที่ดีแก่ผู้ใช้งานและผู้ดูแลระบบ

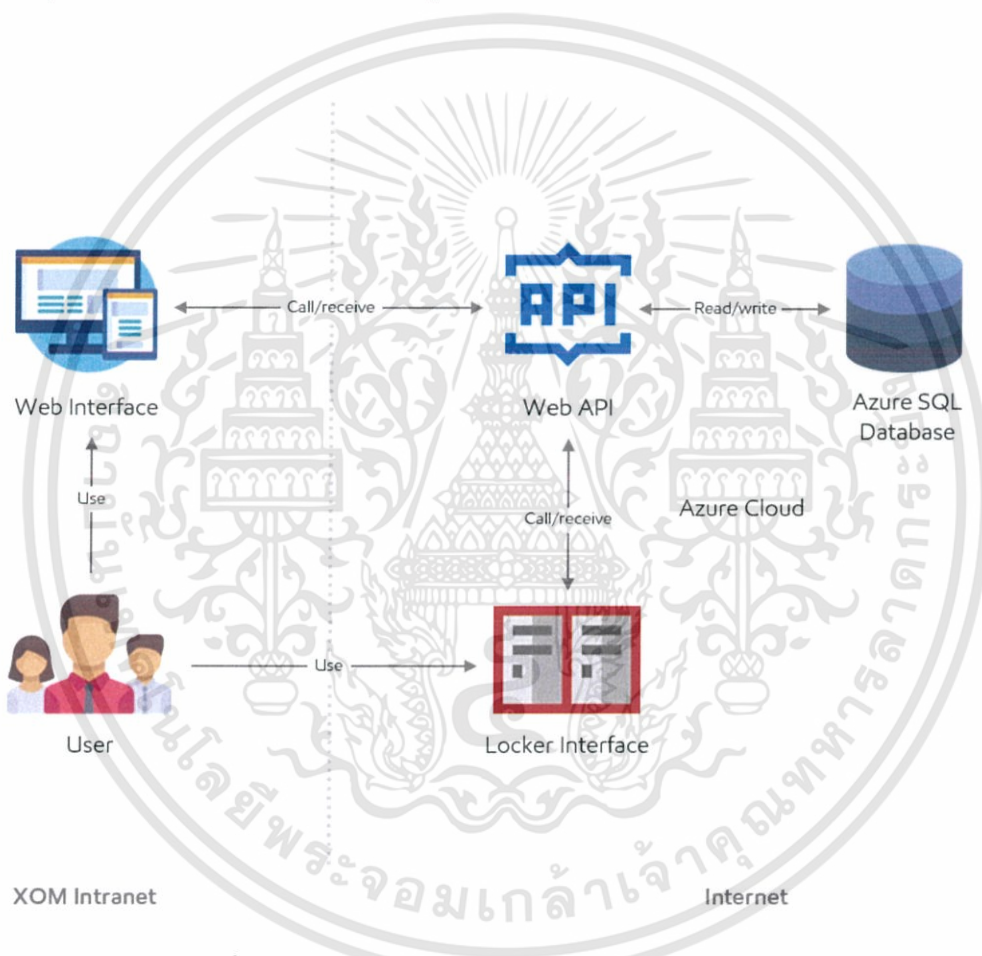
3.2.1.6 ระบบต้องมีความยืดหยุ่น รองรับารรูปแบบที่หลากหลายได้

3.2.2 วางแผนการพัฒนาระบบ

จากผลการศึกษาความต้องการในหัวข้อที่ 3.2.1 กลุ่มผู้พัฒนาเลือกที่จะสร้างหน้าเว็บไซต์ขึ้นสำหรับทั้งผู้ใช้งานทั่วไปและผู้ดูแลระบบ โดยสำหรับผู้ใช้งานทั่วไปจะสามารถเข้าดูรายละเอียดอุปกรณ์ต่าง ๆ ที่มีให้ยืมอยู่ในระบบ รายละเอียดของอุปกรณ์ที่ผู้ใช้งานกำลังยืมและยืมไปแล้ว รับแจ้งเตือนวันเวลาดินอุปกรณ์ รวมถึงเพิ่มอุปกรณ์เข้า Watchlist หากอุปกรณ์นั้นไม่พร้อมให้ยืม โดยระบบจะแจ้งเตือน

ผู้ใช้งานเมื่ออุปกรณ์นั้น ๆ พร้อมให้ยืมอีกครั้ง สำหรับเว็บไซต์ผู้ดูแลระบบ จะสามารถเพิ่ม/ลดอุปกรณ์ รวมถึงกำหนดค่าต่าง ๆ ในระบบได้ เมื่อผู้ใช้งานตรวจเช็คสถานะสิ่งของแล้วและต้องการยืม/คืนสิ่งของ ผู้ใช้งานสามารถไปที่ล็อกเกอร์ แสกนบัตรประจำตัวพนักงาน เลือกสิ่งของที่ต้องการยืม/คืน หยิบของจาก ล็อกเกอร์หรือใส่ของเข้าล็อกเกอร์ และปิดตู้ เป็นอันเสร็จกระบวนการยืม/คืน โดยล็อกเกอร์แต่ละล็อกเกอร์ สามารถมีขนาดที่ต่างกันออกไปได้ตามความต้องการ และพัฒนาบน Raspberry Pi โดยใช้ภาษา Python

เว็บไซต์และล็อกเกอร์จะมีการเชื่อมต่อกับ Web API ซึ่งพัฒนาและ Publish อยู่บน Azure โดยที่ Web API มีหน้าที่เป็นตัวกลางเพื่อจัดการข้อมูลของระบบบนฐานข้อมูล SQL ที่อยู่บน Azure กลุ่ม ผู้พัฒนาสรุปภาพรวมแนวคิดของระบบได้ ดังรูปที่ 3.34



รูปที่ 3.34 ภาพรวมการทำงานของระบบล็อกเกอร์อัจฉริยะ

เมื่อสามารถสรุปภาพรวมแนวคิดของระบบได้แล้ว เนื่องจากโครงการครั้งนี้เป็นโครงการขนาดใหญ่ และกลุ่มของผู้พัฒนาประกอบไปด้วยนักศึกษาฝึกงานสหกิจศึกษาจำนวน 4 คน ทำให้มีข้อจำกัดทางด้านเวลา กลุ่มของผู้พัฒนาจึงได้วางแผนการพัฒนาโดยเลือกนำส่วนที่สำคัญที่สุดของโครงการขึ้นมาทำก่อนในระยะที่ 1 คือ Web Interface, Web API, Locker และ SQL Database และแบ่งงานออกเป็น 3 ส่วนเพื่อแจกแจงงานให้กลุ่มนักพัฒนาที่เป็นนักศึกษาได้ทำงาน โดยรายละเอียดเป็นไป ดังตารางที่ 3.4

ตารางที่ 3.4 การแบ่งภาระงานที่รับผิดชอบของงานล็อกเกอร์อัจฉริยะ

ส่วนที่ 1	ส่วนที่ 2	ส่วนที่ 3
<ul style="list-style-type: none">- ออกแบบและพัฒนา API- จัดเตรียมข้อมูลจำลอง- Publish ขึ้น Azure	<ul style="list-style-type: none">- พัฒนาฐานข้อมูล SQL- ออกแบบและพัฒนา API- พัฒนาล็อกเกอร์ต้นแบบ	<ul style="list-style-type: none">- พัฒนาหน้า User Interface ส่วนเว็บไซต์ผู้ใช้งาน

3.2.3 พัฒนาระบบ

จากการวางแผนงานและแบ่งภาระรับผิดชอบในหัวข้อที่ 3.2.2 ผู้พัฒนารับผิดชอบงานในส่วนที่ 1 ดังแสดงตามตารางที่ 3.4 โดยสามารถแจกแจงรายละเอียดการพัฒนาได้ ดังนี้

3.2.3.1 ออกแบบและพัฒนา API

ในส่วนการออกแบบและพัฒนา API ผู้พัฒนาได้ร่วมกับเพื่อนร่วมงานได้พัฒนา API Endpoints โดยใช้โปรแกรม Visual Studio ในภาษา C# ขึ้นมาทั้งหมด 23 endpoints แบ่งออกเป็นทั้งสิ้น 7 หมวดหมู่ โดยทุก Endpoints เมื่อถูกเรียกใช้งานจะมีรูปแบบของการตอบกลับของข้อมูลเป็น JSON ที่ผู้พัฒนาได้วางรูปแบบไว้ ดังรูปที่ 3.35 โดย Status จะส่งกลับเป็นตัวเลขเพื่อระบุสถานะการเรียกใช้งาน API และ StatusDescription เป็นคำอธิบายตัวเลข Status และ Message เป็นข้อมูลที่ส่งกลับมาจาก API Endpoints เพื่อให้ง่ายต่อการนำไปใช้งานของผู้เรียกใช้ API โดย Base URL ในที่นี้จะ เป็น localhost

```
[{  
  "Status": int,  
  "StatusDescription": string,  
  "Message": string  
}]
```

รูปที่ 3.35 รูปแบบมาตรฐานของการตอบกลับของ Smart Locker API

Status	StatusDescription
1	SUCCESS
2	VALIDATION_ERROR
3	INTERNAL_ERROR
4	INVALID_PARAMS
5	not assigned

รูปที่ 3.36 คำอธิบาย Status และ StatusDescription ของการตอบกลับของ Smart Locker API

API Endpoints ทั้ง 7 หมวดหมู่สามารถแบ่งได้ ดังนี้

1. API ที่เกี่ยวข้องกับฐานข้อมูล Category

เมื่อเรียก endpoints ในหมวดหมู่นี้แล้ว โปรแกรมจะทำการติดต่อฐานข้อมูล Category ในระบบซึ่งเก็บข้อมูลเกี่ยวกับ Category หรือหมวดหมู่ของอุปกรณ์ทั้งหมดภายในระบบ API ในหมวดหมู่นี้ประกอบด้วย

- Get All Categories

เรียกใช้งานผ่าน URL localhost/api/category โดยใช้ HTTP Method GET API จะส่งข้อมูล Category ที่มีทั้งหมดในระบบกลับมาให้ผู้เรียก มีรูปแบบการตอบกลับ ดังรูปที่ 3.37

- Get Filtered Categories

เรียกใช้งานผ่าน URL localhost/api/category?siteid={siteid} โดยใช้ HTTP Method GET โดยที่ผู้เรียกใช้จะต้องระบุ Site ID แล้ว API จะส่งข้อมูล Category ที่มีทั้งหมดภายใน Site นั้น ๆ กลับมาให้ผู้เรียก มีรูปแบบการตอบกลับ ดังรูปที่ 3.37

```
.
"Message": {[
  "intCategoryId": int,
  "strCategoryName": string,
  "strDescription": string,
  "strImage": string,
  "strNote": string
]}
.
```

รูปที่ 3.37 รูปแบบการตอบกลับของ Get Categories endpoints

- Add New Category

เรียกใช้งานผ่าน URL localhost/api/category/add โดยใช้ HTTP Method POST พร้อมแนบไฟล์ JSON ใช้รูปแบบ ดังรูปที่ 3.38 แล้ว API จะเพิ่ม category เข้าสู่ระบบ และจะส่งผลลัพธ์การทำงานกลับมาให้ผู้เรียก

```
{
  "CategoryName": string,
  "Description": string,
  "strImage": string,
  "strNote": string,
  "UpdaterId": int,
  "UpdaterBadgeId": string
}
```

รูปที่ 3.38 JSON ที่ใช้เพื่อเรียกใช้ Add New Category endpoint

2. API ที่เกี่ยวข้องกับฐานข้อมูล Subcategory

เมื่อเรียก endpoints ในหมวดหมู่นี้แล้ว โปรแกรมจะทำการติดต่อฐานข้อมูล Subcategory ในระบบซึ่งเก็บข้อมูลเกี่ยวกับ Subcategory หรือหมวดหมู่ย่อยของอุปกรณ์ทั้งหมดภายในระบบ API ในหมวดหมู่นี้ประกอบด้วย

- Get All Subcategories

เรียกใช้งานผ่าน URL localhost/api/subcategory โดยใช้ HTTP Method GET API จะส่งข้อมูล Subcategory ที่มีทั้งหมดในระบบกลับมาให้ผู้เรียก มีรูปแบบการตอบกลับ ดังรูปที่ 3.39

- Get Filtered Subcategories

เรียกใช้งานผ่าน URL localhost/api/subcategory?siteid={siteid}&categoryid={categoryid} โดยใช้ HTTP Method GET โดยที่ผู้เรียกใช้จะต้องระบุ Site ID หรือ Site ID และ Category ID แล้ว API จะส่งข้อมูล Subcategory ที่มีทั้งหมดที่กรองแล้ว กลับมาให้ผู้เรียก มีรูปแบบการตอบกลับ ดังรูปที่ 3.39

```
·  
"Message": { [  
  "intSubCategoryId": int,  
  "strSubCategoryName": string,  
  "intCategoryId": int,  
  "strCategoryName": string,  
  "strNote": string,  
  "strImage": string,  
  "strDescription": string,  
  ]  
}
```

รูปที่ 3.39 รูปแบบการตอบกลับของ Get Subcategories endpoints

- Add New Subcategory

เรียกใช้งานผ่าน URL localhost/api/subcategory/add โดยใช้ HTTP Method POST พร้อมแนบไฟล์ JSON ใช้รูปแบบ ดังรูปที่ 3.40 แล้ว API จะเพิ่ม subcategory เข้าสู่ระบบและจะส่งผลลัพธ์การทำงานกลับมาให้ผู้เรียก

```

{
  "SubCategoryName": string,
  "CategoryId": int
  "Description": string,
  "Image": string,
  "Note": string,
  "UpdaterId": int,
  "UpdaterBadgeId": string
}

```

รูปที่ 3.40 JSON ที่ใช้เพื่อเรียกใช้ Add New Subcategory endpoint

3. API ที่เกี่ยวข้องกับฐานข้อมูล Item

เมื่อเรียก endpoints ในหมวดหมู่นี้แล้ว โปรแกรมจะทำการติดต่อฐานข้อมูล Item ในระบบซึ่งเก็บข้อมูลเกี่ยวกับ Item ภายในระบบ API ในหมวดหมู่นี้ประกอบด้วย

- Get All Items

เรียกใช้งานผ่าน URL localhost/api/item โดยใช้ HTTP Method GET API จะส่งข้อมูล Item ที่มีทั้งหมดภายในระบบกลับมาให้ผู้เรียก มีรูปแบบการตอบกลับ ดังรูปที่ 3.41

- Get Filtered Items

เรียกใช้งานผ่าน URL localhost/api/item?siteid={siteid}&categoryid={categoryid} || &subcategoryid={subcategoryid} โดยใช้ HTTP Method GET โดยที่ผู้เรียกใช้จะต้องระบุ Site ID หรือ Site ID และ Category ID หรือ Site ID และ Subcategory ID แล้ว API จะส่งข้อมูล Item ที่กรองแล้ว กลับมาให้ผู้เรียก มีรูปแบบการตอบกลับ ดังรูปที่ 3.41

```
·  
"Message": [{  
  "intItemId": int,  
  "strItemName": string,  
  "intSubcategoryId": int,  
  "strSubCategoryName": string,  
  "intCategoryId": int,  
  "strCategoryName": string,  
  "strImage": string,  
  "strNote": string,  
  "intLoanTimeDays": int,  
  "strDescription": string,  
  "bInDisable": bool  
}, {...}]  
·
```

รูปที่ 3.41 รูปแบบการตอบกลับของ Get Item endpoints

Get Item Details

เรียกใช้งานผ่าน URL localhost/api/item/details?siteid={siteid}&userid={userid}&itemid={itemid} โดยใช้ HTTP Method GET โดยที่ผู้เรียกใช้จะต้องระบุ Site ID, User ID และ Item ID แล้ว API จะส่งข้อมูลรายละเอียดของ Item พร้อมรายละเอียดของผู้ใช้งานที่เกี่ยวข้องกับ Item นั้น ๆ กลับมาให้ผู้เรียก มีรูปแบบการตอบกลับ ดังรูปที่ 3.42

```

{
  "Message": {
    "Details": [{
      "intItemId": int,
      "strItemName": string,
      "strDescription": string,
      "intLoanTimeDays": int,
      "intSubcategoryId": int,
      "strSubCategoryName": string,
      "intCategoryId": int,
      "strCategoryName": string,
      "intOwnerId": int,
      "objCount": int
    }],
    "Images": [{
      "intImageId": int,
      "strDescription": string,
      "strImageAddress": string,
      "strImageName": string
    }, {...}],
    "IsNotifyThisSite": bool,
    "NotifyLocation": [{
      "intLocationId": int,
      "strLocationName": string,
      "strSiteName": string
    }, {...}],
    "ObjectAvailableInLocker": [{
      "intLockerId": int,
      "strLockerName": string,
      "strDescription": string,
      "intMaxSlot": int,
      "objCount": int
    }, {...}]
  }
}

```

รูปที่ 3.42 รูปแบบการตอบกลับของ Get Item Details endpoints

- Add New Item

เรียกใช้งานผ่าน URL localhost/api/item/add โดยใช้ HTTP Method POST พร้อมแนบไฟล์ JSON ใช้รูปแบบ ดังรูปที่ 3.43 แล้ว API จะเพิ่ม item เข้าสู่ระบบและจะส่งผลลัพธ์การทำงานกลับมาให้ผู้เรียก

```
{
  "ItemName": string,
  "Description": string,
  "LoanTimeDay": int,
  "OwnerId": int,
  "SubCategoryId": int,
  "ReqTicket": bool,
  "UpdaterId": int,
  "UpdaterBadgeId": string
}
```

รูปที่ 3.43 JSON ที่ใช้เพื่อเรียกใช้ Add New Item endpoint

- Get User's Watchlist

เรียกใช้งานผ่าน URL localhost/api/item/watchlist?userid={userid}

โดยใช้ HTTP Method GET โดยที่ผู้เรียกใช้จะต้องระบุ User ID API จะส่งข้อมูล Item ทั้งหมดที่อยู่ใน Watchlist ของ User กลับมาให้ผู้เรียก มีรูปแบบการตอบกลับ ดังรูปที่ 3.44

```
{
  "Message": [{
    "intItemId": int,
    "strItemName": string,
    "intLocationId": int,
    "strLocationName": str,
    "strSiteName": str,
    "intSubcategoryId": int,
    "strSubcategoryName": string,
    "intCategoryId": int,
    "strCategoryName": string,
    "strImage": string,
    "strNote": string
  }, {..}]
}
```

รูปที่ 3.44 รูปแบบการตอบกลับของ Get User's Watchlist endpoint

- Add User to Watchlist

เรียกใช้งานผ่าน URL localhost/api/item/watchlist/notify โดยใช้ HTTP Method POST พร้อมแนบไฟล์ JSON ใช้รูปแบบ ดังรูปที่ 3.45 แล้ว API จะทำการเพิ่ม item เข้าสู่ watchlist ของ user และจะส่งผลลัพธ์การทำงานกลับมาให้ผู้เรียก

- Remove User from Watchlist

เรียกใช้งานผ่าน URL localhost/api/item/watchlist/unnotify โดยใช้ HTTP Method POST พร้อมแนบไฟล์ JSON ใช้รูปแบบ ดังรูปที่ 3.45 แล้ว API จะทำการลบ item ออกจาก watchlist ของ user และจะส่งผลลัพธ์การทำงานกลับมาให้ผู้เรียก

```
{  
  "SiteId": int,  
  "UserId": int,  
  "ItemId": int  
}
```

รูปที่ 3.45 JSON ที่ใช้เพื่อเรียกใช้ Notify/unnotify endpoint

4. API ที่เกี่ยวข้องกับฐานข้อมูล Object

เมื่อเรียก endpoints ในหมวดหมู่นี้แล้ว โปรแกรมจะทำการติดต่อฐานข้อมูล Object ในระบบซึ่งเก็บข้อมูลเกี่ยวกับ Object หรืออุปกรณ์ทั้งหมดภายในบริษัท API ในหมวดหมู่นี้ประกอบด้วย

- Get Loaned Objects of User

เรียกใช้งานผ่าน URL localhost/api/object/userid={userid} โดยใช้ HTTP Method GET โดยที่ผู้เรียกใช้จะต้องระบุ User ID API จะส่งข้อมูล Object ทั้งหมดที่ผู้ใช้กำลังยืมหรือเคยยืมไปแล้วกลับมาให้ผู้เรียก มีรูปแบบการตอบกลับ ดังรูปที่ 3.46

```
    "Message": [{  
      "intLoanId": int,  
      "intItemId": int,  
      "strItemName": str,  
      "strSubCategoryName": string,  
      "strImage": string,  
      "strNote": string,  
      "strLocationName": string,  
      "strSiteName": string,  
      "strDescription": string,  
      "datLoanDate": datetime,  
      "datDueDate": datetime  
    }, {...}]
```

รูปที่ 3.46 รูปแบบการตอบกลับของ Get User's Loaned Object endpoint

- Get Loaned Object Details

เรียกใช้งานผ่าน URL localhost/api/object/loaned={loaned} โดยใช้ HTTP Method GET โดยที่ผู้เรียกใช้จะต้องระบุ Loan ID API จะส่งข้อมูลรายละเอียดของ Object นั้นกลับมาให้ผู้เรียก มีรูปแบบการตอบกลับ ดังรูปที่ 3.47

```

{
  "Message": {
    "Details": [{
      "intLoanId": int,
      "intObjectId": int,
      "strBarcodeId": string,
      "strSerialNumber": string,
      "intItemId": int,
      "strItemName": string,
      "intOwnerId": int,
      "strDescription": string,
      "intSubCategoryId": int,
      "strSubCategoryName": string,
      "intCategoryId": int,
      "strCategoryName": string,
      "intLockerId": int,
      "strLockerName": string,
      "strLocationName": string,
      "strSiteName": string,
      "intUserLoanId": int,
      "strREQId": string,
      "strWId": string,
      "datLoanDate": datetime,
      "datDueDate": datetime
    }, {...}],
    "Images": [{
      "intImageId": int,
      "intItemId": int,
      "strDescription": string,
      "strImageAddress": string,
      "strImageName": string
    }, {...}]
  }
}

```

รูปที่ 3.47 รูปแบบการตอบกลับของ Get Loaned Item Details endpoint

5. API ที่เกี่ยวข้องกับฐานข้อมูล Country

เมื่อเรียก endpoints ในหมวดหมู่แล้ว โปรแกรมจะทำการติดต่อฐานข้อมูล Country ในระบบซึ่งเก็บข้อมูลเกี่ยวกับ Country หรือประเทศทั้งหมดภายในระบบ API ในหมวดหมู่นี้ ประกอบด้วย

- Get All Country

เรียกใช้งานผ่าน URL localhost/api/country โดยใช้ HTTP Method GET API จะส่งข้อมูล Country ที่มีทั้งหมดในระบบกลับมาให้ผู้เรียก มีรูปแบบการตอบกลับ ดังรูปที่ 3.48

```
.
"Message": [
  {
    "intCountryId": int,
    "strCountryName": string,
    "strDescription": string,
    "bInDisable": bool
  },
  {...},
  ..
]
.
```

รูปที่ 3.48 รูปแบบการตอบกลับของ Get Country endpoint

- Add New Country

เรียกใช้งานผ่าน URL localhost/api/country/add โดยใช้ HTTP Method POST พร้อมแนบไฟล์ JSON ใช้รูปแบบ ดังรูปที่ 3.49 แล้ว API จะเพิ่ม country เข้าสู่ระบบและจะส่งผลลัพธ์การทำงานกลับมาให้ผู้เรียก

```
{
  "CountryName": string,
  "Description": string,
  "UpdaterId": int,
  "UpdaterBadgeId": string
}
```

รูปที่ 3.49 JSON ที่ใช้เพื่อเรียกใช้ Add New Country endpoint

6. API ที่เกี่ยวข้องกับฐานข้อมูล Site

เมื่อเรียก endpoints ในหมวดหมู่นี้แล้ว โปรแกรมจะทำการติดต่อฐานข้อมูล Site ในระบบซึ่งเก็บข้อมูลเกี่ยวกับ Site ทั้งหมดภายในระบบ API ในหมวดหมู่นี้ประกอบด้วย

- Get All Sites

เรียกใช้งานผ่าน URL localhost/api/site โดยใช้ HTTP Method GET API จะส่งข้อมูล Site ที่มีทั้งหมดในระบบกลับมาให้ผู้เรียก มีรูปแบบการตอบกลับ ดังรูปที่ 3.50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Get Sites by Country

เรียกใช้งานผ่าน URL localhost/api/site?countryid={countryid} โดยใช้ HTTP Method GET API จะส่งข้อมูล Site ใน Country กลับมาให้ผู้เรียก มีรูปแบบการตอบกลับ ดังรูปที่ 3.50

```
Message: {[
  intLocationId: int,
  strLocationName: string,
  strSiteName: string,
  intCountryId: int,
  strCountryName: string,
  strDescription: string,
  blnDisable: bool
]}
```

รูปที่ 3.50 รูปแบบการตอบกลับของ Get Site endpoints

- Add New Sites

เรียกใช้งานผ่าน URL localhost/api/site/add โดยใช้ HTTP Method POST พร้อมแนบไฟล์ JSON ใช้รูปแบบ ดังรูปที่ 3.51 แล้ว API จะเพิ่ม site เข้าสู่ระบบและจะส่งผลลัพธ์การทำงานกลับมาให้ผู้เรียก

```
{
  SiteName: string,
  Description: string,
  ShortSiteName: string,
  CountryId: int
  UpdaterId: int,
  UpdaterBadgeId: string
}
```

รูปที่ 3.51 JSON ที่ใช้เพื่อเรียกใช้ Add New Site endpoint

7. API ที่เกี่ยวข้องกับฐานข้อมูล User

เมื่อเรียก endpoints ในหมวดหมู่นี้แล้ว โปรแกรมจะทำการติดต่อฐานข้อมูล User ในระบบซึ่งเก็บข้อมูลเกี่ยวกับ User หรือผู้ใช้งานทั้งหมดภายในระบบ API ในหมวดหมู่นี้ประกอบด้วย

- Check if User Exists

เรียกใช้งานผ่าน URL localhost/api/user?badgeid={badgeid} โดยใช้ HTTP Method GET โดยที่ผู้เรียกใช้จะต้องระบุ Badge ID ของ User API จะตรวจสอบว่ามีข้อมูลผู้ใช้ที่มี Badge ID นี้อยู่ในระบบหรือไม่ ถ้าหากไม่ส่งค่าใด ๆ กลับมา ถ้ามีในระบบจะส่งข้อมูล User นั้นกลับมาให้ผู้เรียก มีรูปแบบการตอบกลับ ดังรูปที่ 3.52

```
· "Message": [{  
  "intUserId": int,  
  "strUserName": string,  
  "strRole": string,  
  "strDescription": string,  
  "intLocationId": int  
}]  
·
```

รูปที่ 3.52 รูปแบบการตอบกลับของ Check if User Exists endpoint

- Create New User

เรียกใช้งานผ่าน URL localhost/api/user/create โดยใช้ HTTP Method POST พร้อมแนบไฟล์ JSON ใช้รูปแบบ ดังรูปที่ 3.53 แล้ว API จะเพิ่ม user เข้าสู่ระบบและจะส่งผลลัพธ์การทำงานกลับมาให้ผู้เรียก

```
{  
  "BadgeId": string,  
  "Description": string,  
  "SiteId": int  
}
```

รูปที่ 3.53 JSON ที่ใช้เพื่อเรียกใช้ Create New User endpoint

- Update User's Details

เรียกใช้งานผ่าน URL localhost/api/user/update โดยใช้ HTTP Method PUT พร้อมแนบไฟล์ JSON ใช้รูปแบบ ดังรูปที่ 3.54 แล้ว API จะเช็คว่าคุณอัปเดตมีสิทธิ์ระดับผู้ดูแลระบบหรือไม่ หากใช่ โปรแกรมจะ update ข้อมูลของ user เข้าสู่ระบบและจะส่งผลลัพธ์การทำงานกลับมาให้ผู้เรียก

```
{  
  "UserId": string,  
  "UserBadgeId": string,  
  "UserRole": string,  
  "Description": string,  
  "SiteId": int,  
  "UpdaterId": string,  
  "UpdaterBadgeId": string  
}
```

รูปที่ 3.54 JSON ที่ใช้เพื่อเรียกใช้ Update User's Details endpoint

3.2.3.2 การจัดเตรียมข้อมูลจำลอง

ผู้พัฒนาได้เขียน code ภาษา SQL เพื่อจำลองข้อมูลลงไปในฐานข้อมูลของระบบ เนื่องจากระบบจำเป็นต้องมีข้อมูล นักพัฒนาจึงจะสามารถทำการทดสอบระบบได้ โดยการจำลองข้อมูลนี้ ผู้พัฒนาจำเป็นต้องทำให้ข้อมูลในแต่ละตารางของฐานข้อมูลสอดคล้องกัน เพื่อให้ไม่มีความผิดพลาดในการทดสอบ

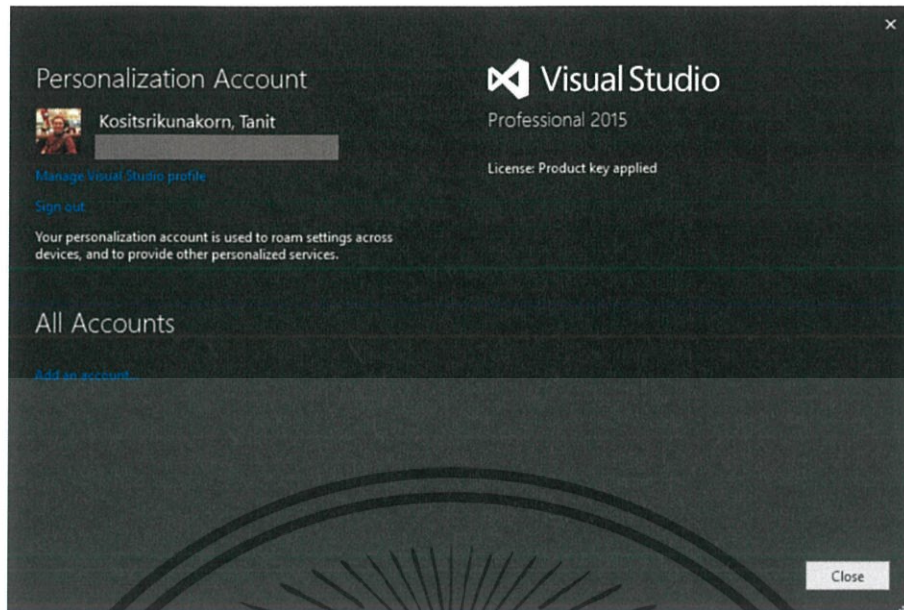
```
81
82 /* Item */
83 INSERT INTO [dbo].[tb_item]([intSubCategoryId], [intOwnerId], [strItemName], [strDescription], [intLoanTimeDays], [blnReqTicket])
84 VALUES ('3', '2', 'Microsoft Sculpt Ergonomics', '', '14', 'true')
85      ,(1, '1', 'Dell Ordinary Keyboard', '', '30', 'true')
86      ,(2, '2', 'Logitech MX Mouse', '', '14', 'true')
87      ,(2, '2', 'Logitech MX Performance Mouse', '', '14', 'true')
88      ,(2, '2', 'Logitech Comfort Mouse', '', '14', 'true')
89      ,(5, '1', 'Apple TV', '', '14', 'true')
90      ,(6, '1', 'Plantronics 325T Headset', '', '60', 'true')
91      ,(8, '1', 'Snowball', '', '14', 'true')
92      ,(4, '1', 'Dell Projector', '', '7', 'true')
93      ,(9, '1', 'Logitech Smart Conference', '', '14', 'true')
94
95 /* Image */
96 INSERT INTO [dbo].[tb_image]([intItemId], [strImgAddress], [strImageName], [strDescription])
97 VALUES ('1', 'www.test.com', '1', '')
98      ,(1, 'www.test.com', '2', '')
99      ,(1, 'www.test.com', '3', '')
100      ,(6, 'www.test.com', '1', '')
101      ,(8, 'www.test.com', '1', '')
102
103
```

รูปที่ 3.55 ตัวอย่าง code สำหรับจำลองข้อมูลลงฐานข้อมูลของระบบ Smart Locker

3.2.3.3 การ Publish ขึ้น Azure

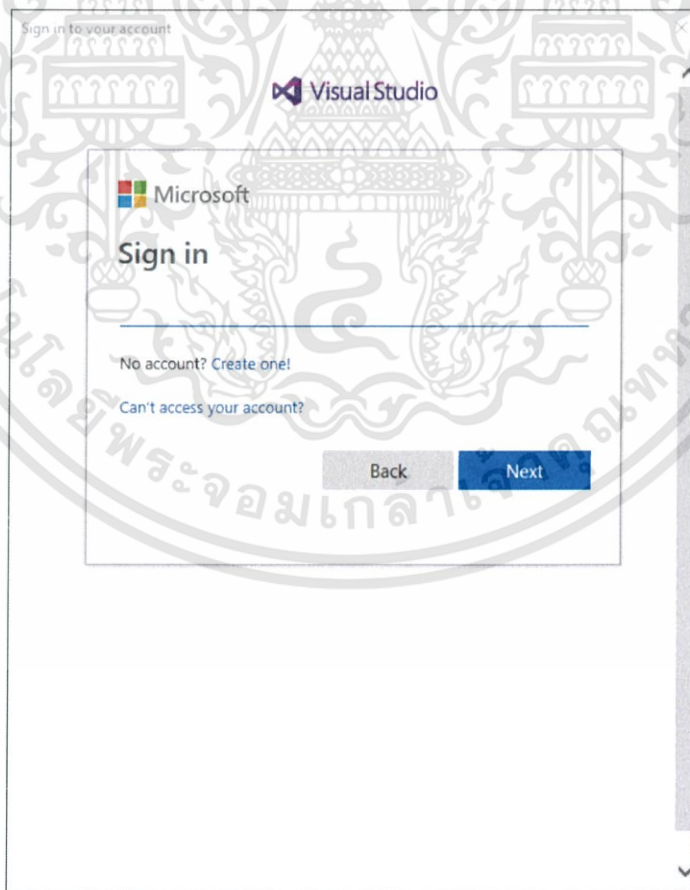
การเชื่อมต่อ Visual Studio เข้ากับ Azure และ Publish Web API นั้น ผู้พัฒนาทำได้ ดังนี้

1. ในโปรแกรม Visual Studio เปิดหน้าต่าง Account Setting โดยกดที่รูป Profile ที่ด้านขวาบนของโปรแกรม



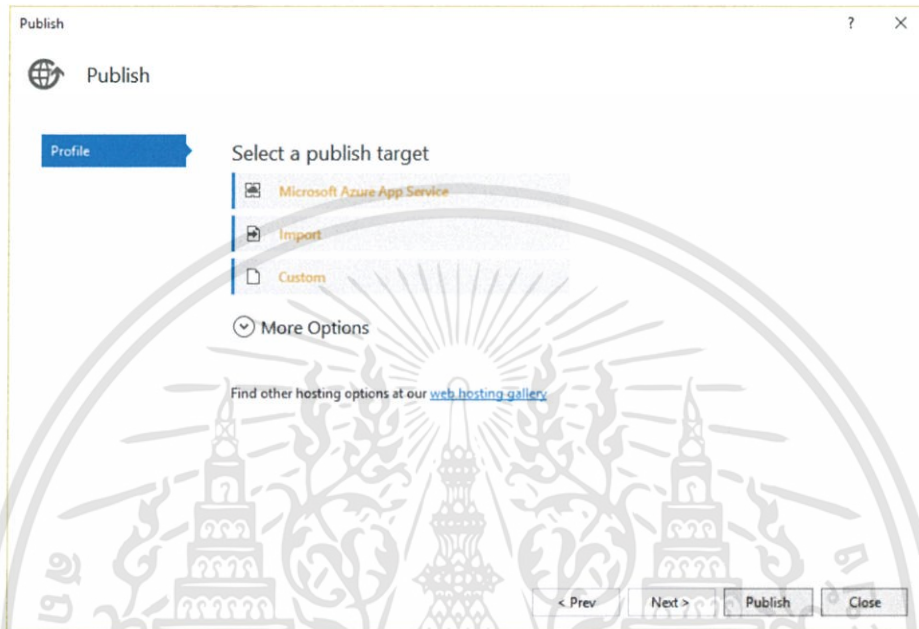
รูปที่ 3.56 หน้าต่าง Account Settings ของโปรแกรม Visual Studio

2. กด Add an account และล็อกอินด้วย Cloud ID ของบริษัท



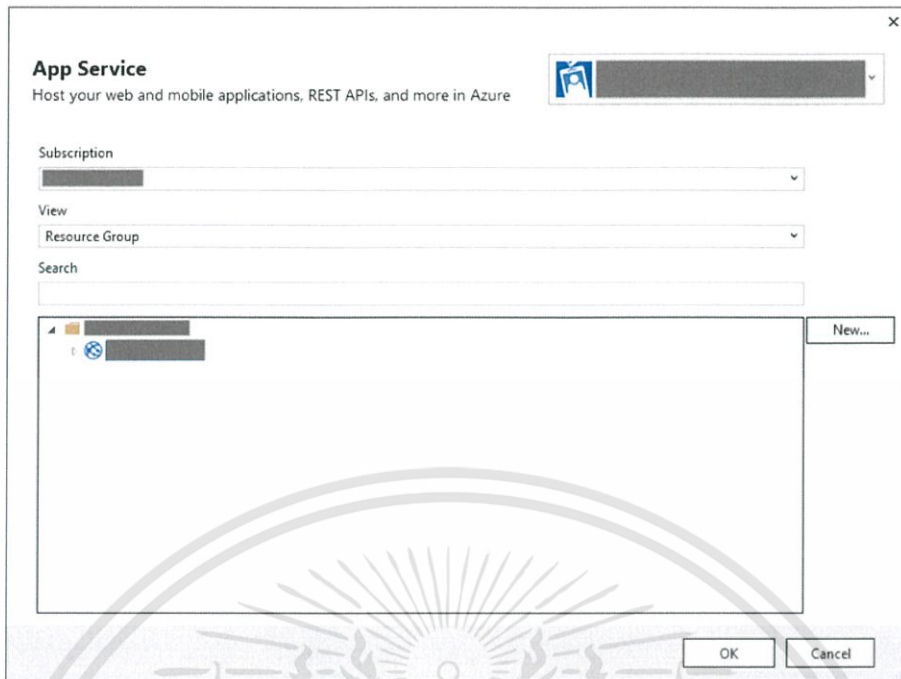
รูปที่ 3.57 หน้าต่าง Sign in to your account ของโปรแกรม Visual Studio

- เมื่อเสร็จสิ้นกระบวนการ ปิดหน้าต่าง Account Settings และไปที่ View > Solution Explorer
- ที่หน้าต่าง Solution Explorer คลิกขวาที่ชื่อโครงการ > Publish
- จะกฏหน้าต่าง ดังรูปที่ 3.58 กต Microsoft Azure App Service



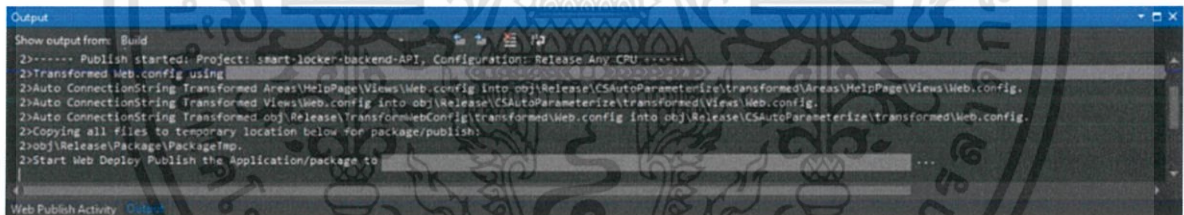
รูปที่ 3.58 หน้าต่าง Publish ในโปรแกรม Visual Studio

- ในหน้าต่าง ดังรูปที่ 3.59 ที่มุมขวบน เลือกผู้ใช้งานที่จะใช้งานให้ถูกต้อง แล้วเลือก Subscription และ เลือก App Service จากนั้นกด OK



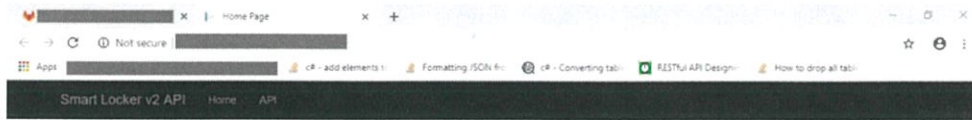
รูปที่ 3.59 หน้าต่าง Create New Connection ของโปรแกรม Visual Studio

7. ใน tab Connection และ Settings จะถูกตั้งค่าโดยอัตโนมัติ ไปที่ tab Preview จากนั้นกด Publish



รูปที่ 3.60 หน้าต่างแสดงการทำงานระหว่างการ Publish

8. เมื่อการ Publish API เสร็จสิ้น เว็บเบราว์เซอร์จะถูกเปิดขึ้นและนำทางไปยังเว็บไซต์ของ Azure ที่ผู้พัฒนาสร้างไว้



ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

[Learn more »](#)

Getting started

ASP.NET Web API is a framework that makes it easy to build HTTP services that reach a broad range of clients, including browsers and mobile devices. ASP.NET Web API is an ideal platform for building RESTful applications on the .NET Framework.

[Learn more »](#)

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

[Learn more »](#)

Web Hosting

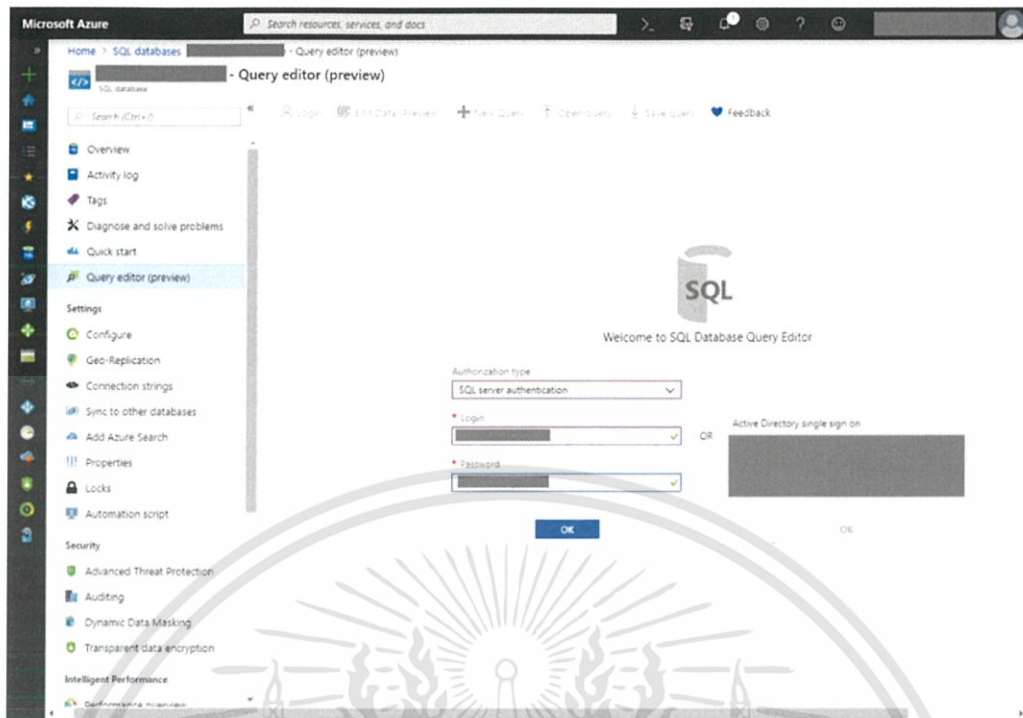
You can easily find a web hosting company that offers the right mix of features and price for your applications.

[Learn more »](#)

© 2016 - My ASP.NET Application

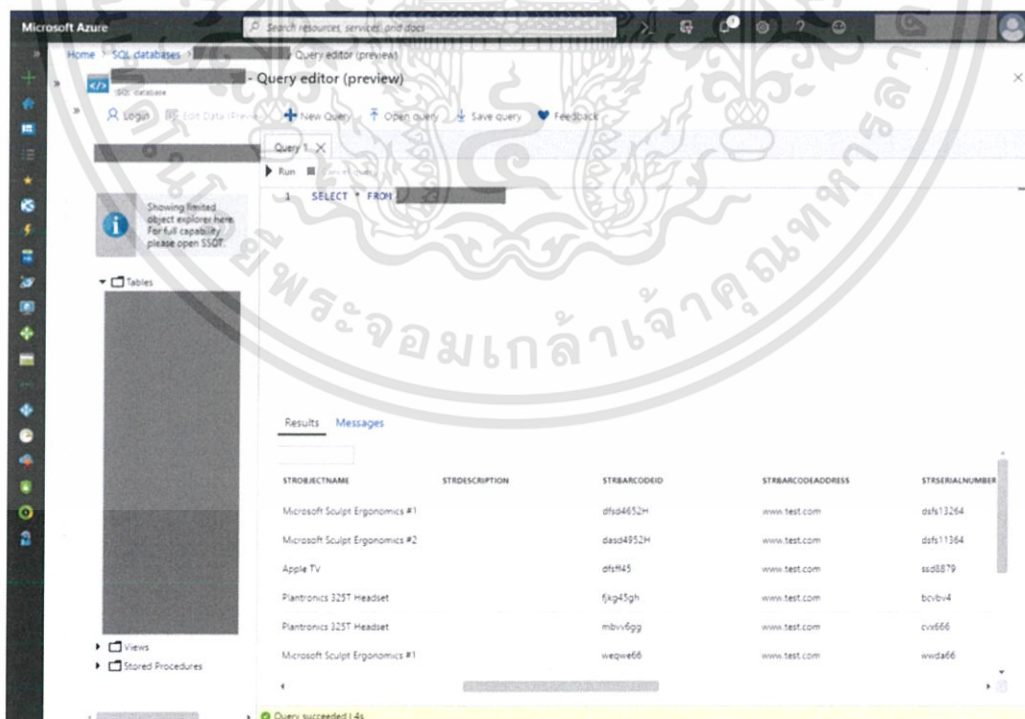
รูปที่ 3.61 หน้าต่างเว็บเบราว์เซอร์แสดงหน้าเว็บหลักของ API บนเว็บไซต์ของ Azure

เมื่อผู้พัฒนานำ API ขึ้นไปไว้บน Azure แล้ว ผู้พัฒนาได้สร้างฐานข้อมูล SQL ไว้บน Azure ด้วย โดยเมื่อเข้าสู่ระบบโดย Cloud ID บนเว็บไซต์ Azure แล้ว ไปที่ฐานข้อมูลที่สร้างไว้ แล้วไปที่ Query Editor และเข้าสู่ระบบอีกครั้ง โดยใช้ชื่อและรหัสผ่านที่สร้างขึ้นเฉพาะสำหรับเข้าถึง Azure SQL Database



รูปที่ 3.62 หน้าเข้าสู่ระบบ Query Editor ของ Azure SQL Database

เมื่อเข้าสู่ระบบแล้วจะสามารถใช้งานคำสั่งภาษา SQL ได้ ผู้พัฒนาได้สั่งใช้งาน Code สำหรับสร้างตารางบนฐานข้อมูล และ Code สำหรับจำลองข้อมูลลงฐานข้อมูลตามลำดับ

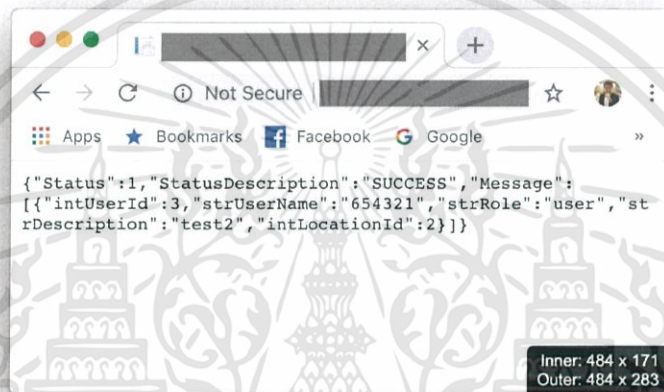


รูปที่ 3.63 ตัวอย่างการใช้คำสั่งภาษา SQL ผ่าน Query Editor บน Azure

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 ทดสอบการทำงานของระบบ

ผู้พัฒนาได้ทำการทดสอบการใช้งานของระบบตลอดช่วงระยะเวลาของการพัฒนา โดยจะเป็นการทดสอบโดยการทดลองเรียกใช้งาน API endpoint ต่าง ๆ เพื่อดูผลลัพธ์ และทดสอบโดยการทดลองเรียกใช้งาน API จากหน้าเว็บไซต์ที่ถูกพัฒนาขึ้น เพื่อสังเกตผลว่าได้ตรงตามที่ต้องการหรือไม่ โดยผลจากการทดสอบในแต่ละครั้ง ผู้พัฒนานำผลการทดสอบมาวิเคราะห์และส่งต่อให้กลุ่มผู้พัฒนาเพื่อพัฒนาและปรับปรุงการทำงานของระบบต่อไป



รูปที่ 3.64 ผลการทดสอบการใช้งาน API โดยการเข้าถึง URL โดยตรงผ่านเว็บเบราว์เซอร์



รูปที่ 3.65 ผลการทดสอบการใช้งาน API ผ่านหน้าเว็บไซต์ที่สร้างขึ้น

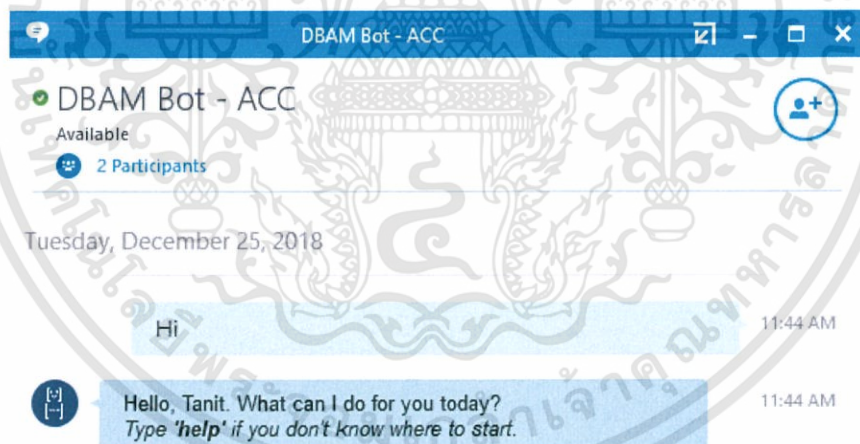
บทที่ 4

ผลการพัฒนา

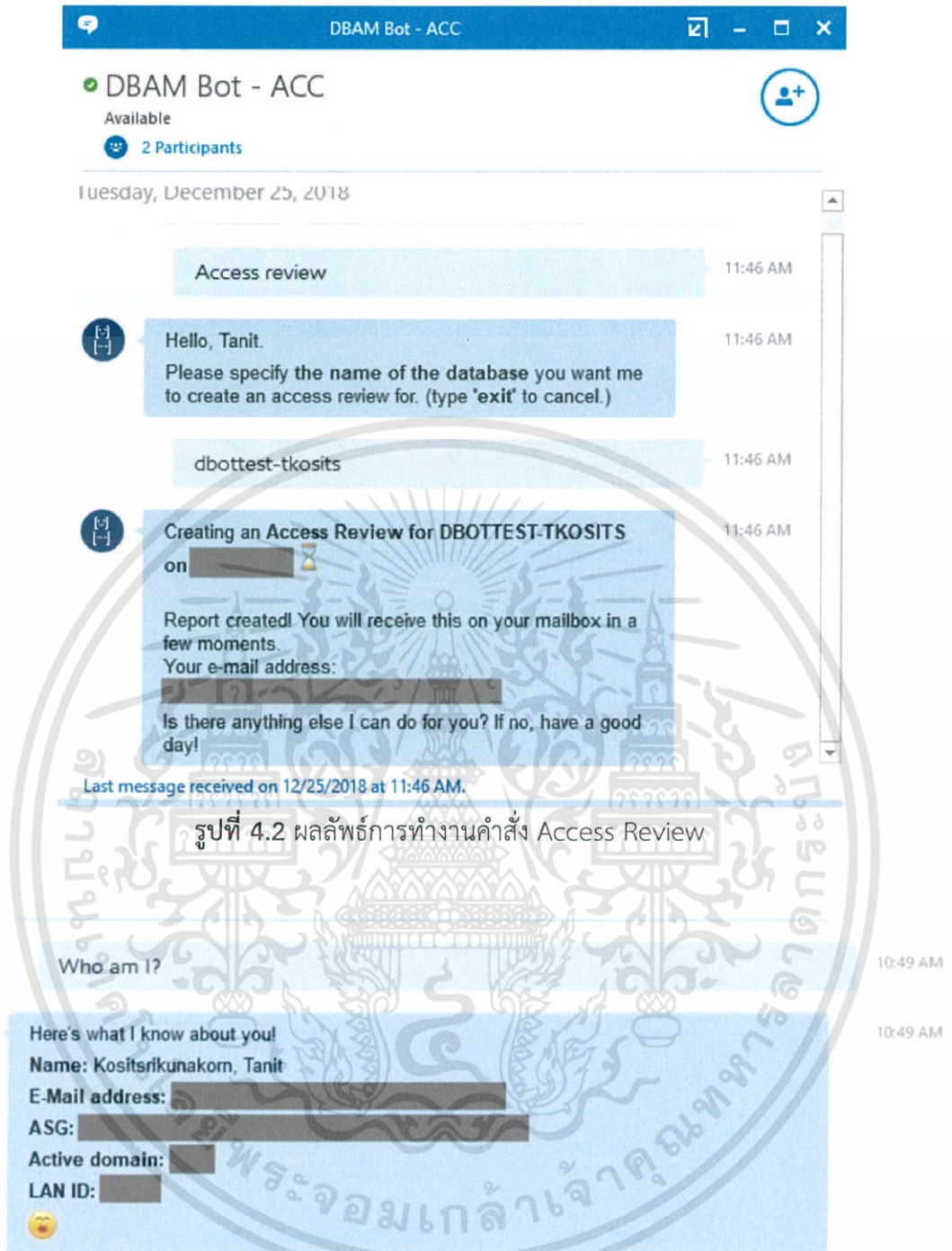
ในส่วนของผลการพัฒนาระบบ ผู้พัฒนาแบ่งผลการพัฒนาออกเป็นสองส่วน โดยแบ่งตามระบบที่พัฒนาได้แก่ ระบบอัตโนมัติสำหรับแชทบอต และระบบอัตโนมัติสำหรับล็อกเกอร์อัจฉริยะ

4.1 ผลการพัฒนาระบบอัตโนมัติสำหรับแชทบอต

ผู้พัฒนาได้วัดประสิทธิผลของการพัฒนาโปรแกรมแชทบอต โดยจากผลการทดสอบการใช้งาน โปรแกรมพบว่า แชทบอตสามารถสนทนารับรู้ข้อมูลของคู่สนทนา และสามารถใช้ประโยชน์จากข้อมูลนี้เพื่อเพิ่มประสิทธิภาพการทำงานของคำสั่งต่าง ๆ ได้ และแชทบอตสามารถตอบโต้กับผู้ใช้งานได้ในคำสั่ง Access Review, คำสั่ง Backup Database, คำสั่ง Restore Database และคำสั่ง Refresh Database นอกจากนี้แชทบอตยังสามารถเชื่อมต่อกับระบบอื่น ๆ ได้แก่ DBAM Automation API และ ITSM API เพื่อทำงานได้ การทำงานของแชทบอตในสถานการณ์ต่าง ๆ แสดงได้ดังรูปที่ 4.1 ถึงรูปที่ 4.7



รูปที่ 4.1 คำทักทายเมื่อเริ่มต้นสนทนาของแชทบอต



รูปที่ 4.3 แชทบอตรับรู้ข้อมูลของคุณสนทนาได้

Here's your operation details.
Operation: Backup a database
Database Name: dbotest-tkosits
Server Name: [REDACTED]
Lanshare Path: [REDACTED]
Please type 'confirm' to confirm and start the operation. You can type 'exit' if you want to cancel the operation.

10:50 AM

confirm

10:50 AM

Backup in progress ⌚
Operation completed successfully.
Your .bak file is saved at [REDACTED]
Is there anything else I can do for you? If no, have a good day!

10:50 AM

รูปที่ 4.4 ผลลัพธ์การทำงานคำสั่ง Backup Database

Do you have any ITSM ticket associated to this operation?
yes
Okay. Please enter the ITSM ticket ID for this operation.
[REDACTED]
I can only support WO, INC, TAS, and CRQ at the moment. Please try again. Type 'back' if you don't have any one of these types.

Last message received on 11/25/2018 at 11:04 PM.

รูปที่ 4.5 แช้ตบอตเชื่อมต่อกับระบบ ITSM API เพื่อตรวจสอบ Ticket

Do you have any ITSM ticket associated to this operation?.

no

Okay. No problem. I will create a Work Order for you.



Work order created.

Ticket ID: [REDACTED]

Type 'go' to confirm WO and continue.

Last message received on 11/25/2018 at 11:05 PM.

รูปที่ 4.6 แค็ตบอลเชื่อมต่อกับระบบ ITSM API เพื่อสร้าง Ticket ขึ้นใหม่

A

Here's a list of databases with the name

A

#	Database name	Server	Environment	Database status
1			Acceptance	Loading
2			Development	Active
3			Development	Loading
4			Production	Loading
5			Development	Active
6			Production	Loading
7			Production	Loading
8			Production	Active

Which one do you want to backup?

You can specify the server name if what you are looking for is not listed.

5

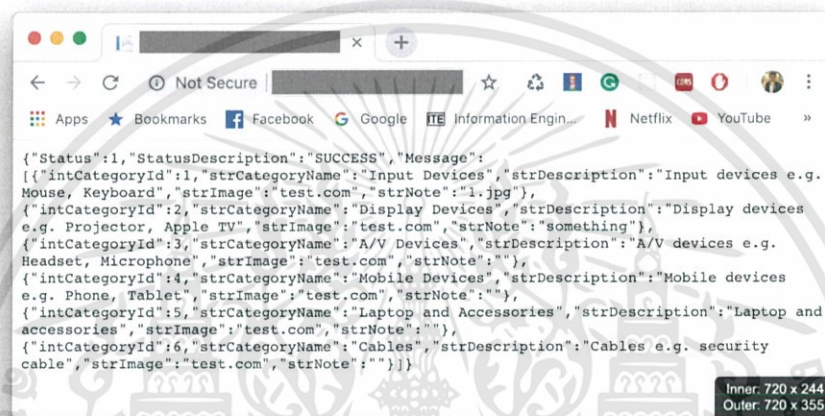
Checking your access for A on B

You have the right to access this database!

รูปที่ 4.7 แค็ตบอลค้นหาและสร้างตารางข้อมูลให้ผู้ใช้เลือกเพื่อสร้างความสะดวกแก่ผู้ใช้งาน และตรวจสอบสิทธิ์การเข้าถึงให้โดยอัตโนมัติ

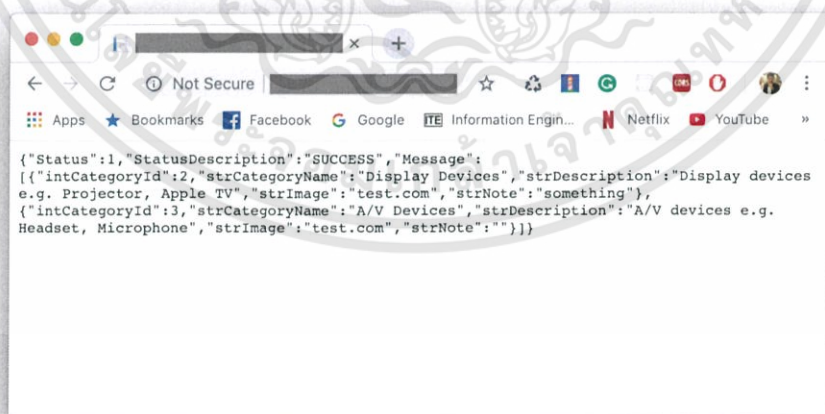
4.2 ผลการพัฒนาระบบอัตโนมัติสำหรับล็อกเกอร์อัจฉริยะ

ผู้พัฒนาได้ทำการวัดประสิทธิภาพการทำงานของระบบในส่วนที่ผู้พัฒนาได้พัฒนาขึ้น โดยจากการทดสอบพบว่าระบบ API ของล็อกเกอร์อัจฉริยะสามารถทำงานได้ตามที่วางแผนไว้ โดยสามารถเชื่อมต่อกับฐานข้อมูล และประมวลผลการร้องขอข้อมูลเพื่อให้ได้ข้อมูลที่ต้องการได้ถูกต้องและแม่นยำ และการสร้างข้อมูลจำลองลงในฐานข้อมูลทำให้ผู้พัฒนารวมถึงผู้พัฒนารายอื่น ๆ สามารถทดสอบการทำงานเสมือนจริงของระบบได้ตามความคาดหวัง การทำงานของระบบล็อกเกอร์อัจฉริยะในสถานการณ์ต่าง ๆ แสดงได้ดังรูปที่ 4.8 และรูปที่ 4.9



```
{
  "Status": 1,
  "StatusDescription": "SUCCESS",
  "Message": [
    {
      "intCategoryId": 1,
      "strCategoryName": "Input Devices",
      "strDescription": "Input devices e.g. Mouse, Keyboard",
      "strImage": "test.com",
      "strNote": "1.jpg"
    },
    {
      "intCategoryId": 2,
      "strCategoryName": "Display Devices",
      "strDescription": "Display devices e.g. Projector, Apple TV",
      "strImage": "test.com",
      "strNote": "something"
    },
    {
      "intCategoryId": 3,
      "strCategoryName": "A/V Devices",
      "strDescription": "A/V devices e.g. Headset, Microphone",
      "strImage": "test.com",
      "strNote": ""
    },
    {
      "intCategoryId": 4,
      "strCategoryName": "Mobile Devices",
      "strDescription": "Mobile devices e.g. Phone, Tablet",
      "strImage": "test.com",
      "strNote": ""
    },
    {
      "intCategoryId": 5,
      "strCategoryName": "Laptop and Accessories",
      "strDescription": "Laptop and accessories",
      "strImage": "test.com",
      "strNote": ""
    },
    {
      "intCategoryId": 6,
      "strCategoryName": "Cables",
      "strDescription": "Cables e.g. security cable",
      "strImage": "test.com",
      "strNote": ""
    }
  ]
}
```

รูปที่ 4.8 ตัวอย่างผลลัพธ์จากการเรียกใช้งาน Get All Categories endpoints แสดงถึง category ทั้งหมดภายในระบบ



```
{
  "Status": 1,
  "StatusDescription": "SUCCESS",
  "Message": [
    {
      "intCategoryId": 2,
      "strCategoryName": "Display Devices",
      "strDescription": "Display devices e.g. Projector, Apple TV",
      "strImage": "test.com",
      "strNote": "something"
    },
    {
      "intCategoryId": 3,
      "strCategoryName": "A/V Devices",
      "strDescription": "A/V devices e.g. Headset, Microphone",
      "strImage": "test.com",
      "strNote": ""
    }
  ]
}
```

รูปที่ 4.9 ตัวอย่างผลลัพธ์จากการเรียกใช้งาน Get Filtered Categories endpoints โดยการกรอง category ภายในระบบโดยใช้ Site ID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการพัฒนาและข้อเสนอแนะ

5.1 สรุปผลการพัฒนา

ผู้พัฒนาได้มีโอกาสเข้าร่วมโครงการสหกิจศึกษาที่ บริษัท เอ็กซ์คอนโมบิล จำกัด ในแผนกการดูแล และจัดการฐานข้อมูล โดยในขณะที่เข้าร่วมโครงการสหกิจศึกษาผู้พัฒนาได้รับมอบหมายให้พัฒนาระบบอัตโนมัติสำหรับแชทบอต DBAM Bot โดยได้รับความต้องการจากเจ้าของผลิตภัณฑ์ ผู้พัฒนาได้ศึกษา โปรแกรมแชทบอต รวบรวมความต้องการเพิ่มเติม ร่วมวางแผนการพัฒนาโปรแกรมกับเพื่อนร่วมงาน พัฒนาโปรแกรม ทดสอบการทำงานของโปรแกรม และนำโปรแกรมไปใช้งานจริง

เมื่อการดำเนินงานพัฒนาโปรแกรมแชทบอตเสร็จสิ้น พบว่าประสิทธิผลของการทำงานของ แชทบอตนั้นน่าพึงพอใจ ลดขั้นตอนและเวลาการทำงานได้ และแชทบอตสามารถทำงานได้ตามความต้องการทางธุรกิจของหน่วยงาน ตามที่ผู้พัฒนาได้รับมอบหมายความต้องการมาจากเจ้าของผลิตภัณฑ์

นอกจากนี้ผู้พัฒนายังได้มีโอกาสร่วมคิดค้น นำเสนอ และพัฒนาระบบอัตโนมัติสำหรับล็อกเกอร์อัจฉริยะ โดยที่ผู้พัฒนาได้มีโอกาสร่วมตั้งแต่จุดเริ่มต้น ตั้งแต่รับทราบและเข้าใจถึงปัญหาที่เกิดขึ้น รวบรวม ความต้องการต่าง ๆ นำมาคิด วิเคราะห์เพื่อค้นหาวิธีแก้ไขปัญหาที่เหมาะสม ออกแบบและวางแผนการ พัฒนาระบบ พัฒนาระบบ ไปจนถึงทดสอบการใช้งานของระบบที่ผู้พัฒนาพัฒนาขึ้น

เมื่อช่วงระยะเวลาของการทำสหกิจศึกษาเสร็จสิ้น ระบบ API และการเชื่อมต่อต่าง ๆ ที่อยู่ในส่วน ของผู้พัฒนานั้น สามารถทำงานและให้ผลลัพธ์ได้ตามที่ต้องการ

5.2 ปัญหาที่พบ และวิธีการแก้ไข

5.2.1 เนื่องจาก บริษัท เอ็กซ์คอนโมบิล จำกัด เป็นบริษัทที่อยู่ในเครือเอ็กซ์คอนโมบิล ซึ่งมีขนาดใหญ่ ทำให้การทำงานต่าง ๆ ต้องคำนึงถึงความปลอดภัยของข้อมูลและระบบอยู่ตลอดเวลา ทำให้การทำงาน เช่น การขอซอฟต์แวร์ที่จำเป็นในการพัฒนาโปรแกรม หรือ ขอสสิทธิ์การเข้าถึงฐานข้อมูลบาง ฐานข้อมูล ใช้เวลานาน ส่งผลกระทบถึงความล่าช้าในการทำงาน

วิธีการแก้ไข : ผู้พัฒนาได้วางแผนการใช้ซอฟต์แวร์ต่าง ๆ ล่วงหน้า เพื่อให้ได้ซอฟต์แวร์ในช่วงเวลาที่พอดีกับความ ต้องการ และคอยติดตามสถานะของคำร้องหรือการทำงานต่าง ๆ เพื่อให้มั่นใจว่าจะได้รับผลลัพธ์เร็วที่สุด เพื่อให้กระทบต่อการทำงานของผู้พัฒนาให้น้อยที่สุด

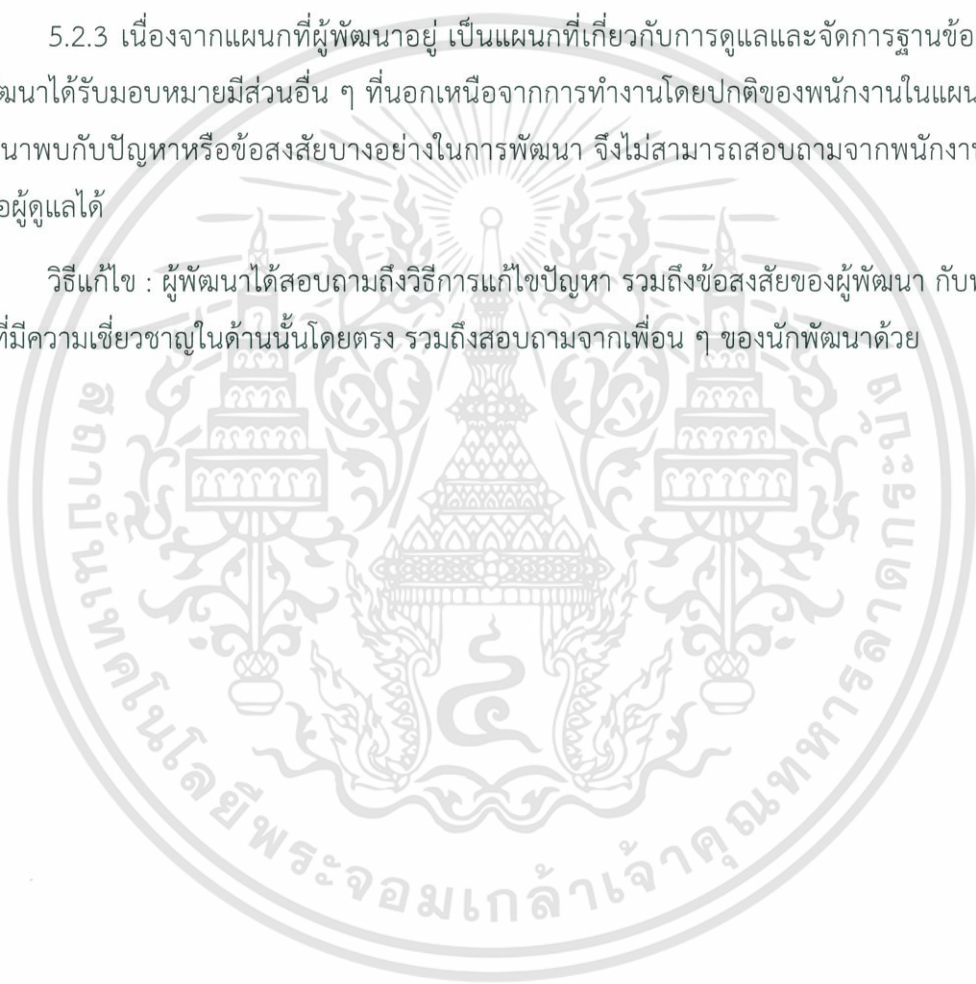
5.2.2 ผู้พัฒนามีโอกาสร่วมงานกับเพื่อนร่วมงานจากต่างประเทศ ซึ่งมีเขตเวลาที่ต่างกันถึง 14 ชั่วโมง ส่งผลให้ไม่มีช่วงเวลาทำงานที่ทับซ้อนกัน เมื่อต้องการสอบถามสิ่งต่าง ๆ จำเป็นต้องทำผ่าน

E-mail ซึ่งใช้เวลามากกว่าจะได้รับคำตอบกลับ รวมถึงการประชุม 2 ครั้งต่อสัปดาห์ที่เริ่มในเวลา 18.00 น. ทำให้เกิดความเหนื่อยล้าในการทำงาน

วิธีแก้ไข : เนื่องจากบริษัทของผู้พัฒนา ไม่ได้กำหนดเวลาเข้า-ออกงานเนื่องจากต้องการให้พนักงานปรับเปลี่ยนเวลาการทำงานและรูปแบบการทำงานให้เหมาะสมกับงานและตัวพนักงาน ผู้พัฒนาจึงปรับเวลาการทำงานในบางวัน โดยเข้าทำงานช้าลงเล็กน้อยในตอนเช้า และเลิกงานในช่วงระหว่าง 20.00 – 21.00 น. เพื่อให้สามารถติดต่อกับผู้ร่วมงานต่างประเทศผ่าน Skype for Business ได้ในทันที และลดความเหนื่อยล้าจากการต้องตื่นนอนเช้าของผู้พัฒนาได้

5.2.3 เนื่องจากแผนกที่ผู้พัฒนาอยู่ เป็นแผนกที่เกี่ยวกับการดูแลและจัดการฐานข้อมูล และงานที่ผู้พัฒนาได้รับมอบหมายมีส่วนอื่น ๆ ที่นอกเหนือจากการทำงานโดยปกติของพนักงานในแผนก ทำให้เมื่อผู้พัฒนาพบกับปัญหาหรือข้อสงสัยบางอย่างในการพัฒนา จึงไม่สามารถสอบถามจากพนักงานภายในแผนกหรือผู้ดูแลได้

วิธีแก้ไข : ผู้พัฒนาได้สอบถามถึงวิธีการแก้ไขปัญหา รวมถึงข้อสงสัยของผู้พัฒนา กับพนักงานในแผนกที่มีความเชี่ยวชาญในด้านนั้นโดยตรง รวมถึงสอบถามจากเพื่อน ๆ ของนักพัฒนาด้วย



เอกสารอ้างอิง

1. Linda Caputo. 2015. What is UCMA 5.0. สืบค้นเมื่อ 6 สิงหาคม 2561, จาก <https://docs.microsoft.com/en-us/skype-sdk/ucma/what-is-ucma-5-0>
2. Laem. 2011. .NET Framework คืออะไร มีที่มาและความสำคัญอย่างไร. สืบค้นเมื่อ 6 สิงหาคม 2561, จาก <https://notebookspec.com/net-framework/>
3. Mike Wasson. 2017. Get Started with ASP.NET Web API 2 (C#). สืบค้นเมื่อ 17 กันยายน 2561, จาก <https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api>
4. Watchanan Chantapakul. 2017. ตอน 1 : มารู้จัก Cloud Computing และ Cloud Services ของ Microsoft — Azure. สืบค้นเมื่อ 30 พฤศจิกายน 2561, จาก <https://medium.com/blogs-194/มารู้จัก-cloud-computing-และ-cloud-services-ของ-microsoft-azure-74849f4fb6a3>
5. PoundXI. 2017. Raspberry Pi คืออะไร ?. สืบค้นเมื่อ 21 ธันวาคม 2561, จาก <https://poundxi.com/raspberry-pi-คืออะไร>
6. Thumbsupteam. 2015. Microsoft เปิดตัว Skype for Business พลิกรูปแบบการสื่อสารและ การทำงานร่วมกันขององค์กรธุรกิจ. สืบค้นเมื่อ 8 มกราคม 2562, จาก <http://thumbsup.in.th/2015/04/microsoft-launches-skype-for-business-thailand/>
7. ศิริรัตน์ กรอดสุข. 2018. เริ่มต้นกับการบริหารโครงการโดยใช้เครื่องมือ Team Foundation Server (Phrase 1 : Overview). สืบค้นเมื่อ 8 มกราคม 2562, จาก <https://sysadmin.psu.ac.th/2018/04/10/เริ่มต้นกับการบริหารโค/>