

โครงการนิสิตชั้นปีที่ 4

การพัฒนาระบบ Common Reporting Standard (CRS)

เสนอโดย

ณัฐภัทร

ดั่งดี

6330169921

คณะกรรมการสอบโครงการนิสิตชั้นปีที่ 4

| | | |
|---------------|---------------|------------------|
| รศ.ดร. ณัฐ | ลีละวัฒน์ | อาจารย์ที่ปรึกษา |
| ผศ. ภูมิ | เหลื่องจามิกร | อาจารย์กรรมการ |
| คุณ กิตติพงษ์ | จรัสลักษณ์ | กรรมการ |

ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2566

บทคัดย่อ

บริษัท ทรูศึกษา เป็นองค์กรที่ขับเคลื่อนธุรกิจด้วยข้อมูล (Data Driven Organization) เพื่อใช้ประกอบการตัดสินใจในการให้บริการ ทำให้ข้อมูลเป็นหัวใจหลักในการทำงาน โดยแผนกที่มีหน้าที่จัดการข้อมูลของบริษัทมีชื่อว่า Enterprise data services(EDS) ซึ่งมีบทบาทอย่างมากในการขับเคลื่อนธุรกิจขององค์กร โดยแผนก EDS มีลักษณะการทำงานทั้งแบบแมนนวลและแบบอัตโนมัติ ในรูปแบบกิจวัตร รายวัน รายเดือน และ รายปี การศึกษาครั้งนี้มีจุดประสงค์เพื่อพัฒนาระบบอัตโนมัติที่มีชื่อว่า Common Reporting Standard(CRS) ซึ่งเป็นระบบที่มีผลลัพธ์เป็นรายงานที่เกี่ยวข้องกับข้อมูลทางภาษีในบัญชีต่างประเทศโดยมีการเรียกใช้งานในรูปแบบรายปี และอำนวยความสะดวกให้แก่พนักงานแผนก EDS ในการใช้งานระบบ CRS โดยมีวิธีการดำเนินงานคือ ศึกษารายละเอียดของรายงานและสร้างระบบอัตโนมัติ ซึ่งจะวัดผลการทำงานของระบบด้วย 1.ความถูกต้องของรายงาน CRS 2.ความสะดวกในการใช้งาน และ 3.ความรวดเร็วในการทำงานของระบบ CRS จากการดำเนินงานพบว่าระบบสามารถสร้างรายงานได้ถูกต้องตามข้อกำหนดของรายงานและสะดวกต่อการใช้งานและมีการทำงานที่รวดเร็ว

สารบัญ

| | |
|--------------------------------------------------------|----|
| บทคัดย่อ | ก |
| สารบัญ..... | ข |
| สารบัญรูป..... | ง |
| สารบัญตาราง..... | ฉ |
| บทที่ 1 บทนำ | 1 |
| 1.1 ที่มาและความสำคัญ..... | 1 |
| 1.2 วัตถุประสงค์การทำโครงการ | 5 |
| 1.3 ขอบเขตโครงการ | 5 |
| 1.4 ผลผลิตที่คาดว่าจะได้รับ (Expected Output)..... | 6 |
| 1.5 ผลลัพธ์ที่คาดว่าจะได้รับ (Expected Outcome)..... | 6 |
| 1.6 แผนการดำเนินงาน..... | 6 |
| บทที่ 2 ทฤษฎีที่เกี่ยวข้อง..... | 7 |
| 2.1. ETL (Extract-Transforms-Load) | 7 |
| 2.2. Database | 10 |
| 2.3 Big O Notation..... | 12 |
| 2.4 The Open Group Architecture Framework (TOGAF)..... | 14 |
| 2.5 Zachman Framework for Enterprise Architecture..... | 15 |
| บทที่ 3 วิธีการดำเนินงานวิจัย | 16 |
| 3.1 ระยะการดำเนินโครงการ..... | 16 |
| 3.2 Develop..... | 16 |
| 3.3 System Integration Test | 20 |
| 3.4 User Acceptance Test | 20 |
| บทที่ 4 ผลการดำเนินงานวิจัย..... | 21 |
| 4.1 Result of Develop..... | 21 |

| | |
|---------------------------------------------------|----|
| 4.2 Result of System Integration Test | 22 |
| 4.3 Result of User Acceptance Test..... | 24 |
| บทที่ 5 สรุปและอภิปรายผล | 25 |
| 5.1. สรุปผล..... | 25 |
| 5.2. อภิปรายผล | 25 |
| 5.3 ข้อเสนอแนะและข้อจำกัด..... | 25 |
| บรรณานุกรม..... | 26 |
| ภาคผนวก ก..... | 27 |
| CRS Architecture..... | 27 |
| Transform algorithm..... | 36 |
| CRS control framework & configuration design..... | 41 |
| Power Bi Dashboard | 46 |

สารบัญรูป

| | |
|-----------------------------------------------------------------------------------------|----|
| รูปที่ 1: azure services environment | 2 |
| รูปที่ 2: Data tiers ของบริษัทกรณีสึกษา..... | 3 |
| รูปที่ 3: บทบาทของ azure data factory | 3 |
| รูปที่ 4: CRS control framework overview | 5 |
| รูปที่ 5: กระบวนการ Extract-Transforms-Load | 7 |
| รูปที่ 6: การเปรียบเทียบการจัดข้อมูลระหว่าง Operational database กับ Datawarehouse..... | 10 |
| รูปที่ 7: การรวมกันของข้อมูล | 11 |
| รูปที่ 8: คุณสมบัติการไม่เปลี่ยนแปลงของข้อมูลในคลังข้อมูล..... | 11 |
| รูปที่ 9: ความสัมพันธ์ของ runtime และ input size ระหว่าง Order of ต่างๆ..... | 13 |
| รูปที่ 10: 9 Principle of Enterprise Architecture by TOGAF | 14 |
| รูปที่ 11: Phase การดำเนินงานของ CRS Project | 16 |
| รูปที่ 12: ตัวอย่าง requirement ของ OrganizationParty_Type | 17 |
| รูปที่ 13: องค์ประกอบของ azure data factory | 20 |
| รูปที่ 14: CRS report example..... | 21 |
| รูปที่ 15: time consume for created report..... | 22 |
| รูปที่ 16: error message จากการ validate ด้วย XML schema..... | 23 |
| รูปที่ 17: example of sql table in data tier 2 | 24 |
| รูปที่ 18: Architecture vision model | 28 |
| รูปที่ 19: CRS business model canvas | 29 |
| รูปที่ 20: business capability model..... | 30 |
| รูปที่ 21: legend categories of business capability model..... | 31 |
| รูปที่ 22: target business capability map | 31 |
| รูปที่ 23: sample input data..... | 32 |
| รูปที่ 24: baseline CRS data migration diagram | 33 |
| รูปที่ 25: target CRS data migration diagram..... | 34 |

| | |
|----------------------------------------------------------|----|
| รูปที่ 26: baseline use-case diagram | 35 |
| รูปที่ 27: target use-case diagram | 36 |
| รูปที่ 28: XML schema expand diagram | 37 |
| รูปที่ 29: Baseline operation process of algorithm | 39 |
| รูปที่ 30: Target operation process of algorithm | 40 |
| รูปที่ 31: Configure setup detail..... | 41 |
| รูปที่ 32: CRS Data pipeline detail | 44 |
| รูปที่ 33: dynamic Power bi dashboard..... | 47 |

สารบัญตาราง

| | |
|-------------------------------------------------------------|----|
| ตารางที่ 1: แผนการดำเนินงานของงานวิจัย | 6 |
| ตารางที่ 2: การดำเนินการตามกระบวนการ Zachman Framework..... | 15 |
| ตารางที่ 3: ชนิดของข้อมูลที่เกี่ยวข้องในการรายงาน | 18 |
| ตารางที่ 4: mapping information characteristic | 38 |
| ตารางที่ 5: parameter table | 45 |

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

1.1.1 ภาพรวมขององค์กร

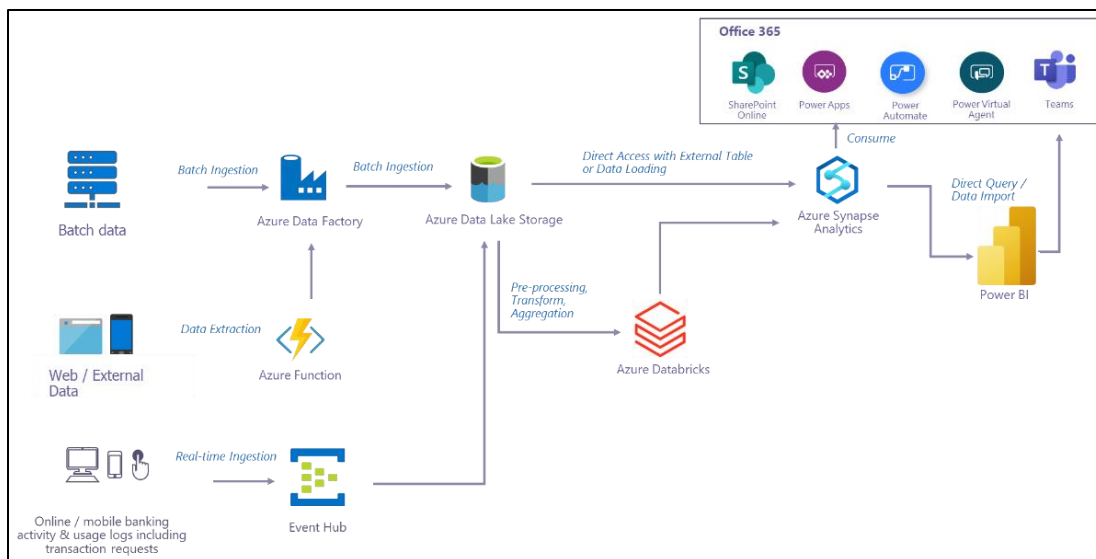
บริษัทธนศึกษาเป็นธนาคารพาณิชย์ที่ให้บริการด้านการเงิน สินเชื่อและเงินฝาก มีบริการที่หลากหลายครอบคลุมทุกกลุ่มลูกค้า โดยยังมีธุรกิจที่เกี่ยวข้องอื่นๆผ่านบริษัทย่อย ตัวอย่างเช่น ธุรกิจการถือหุ้น การให้เช่าหลักทรัพย์และสัญญาซื้อขายล่วงหน้า การเงินและการลงทุน เป็นต้น อีกทั้งยังเป็นองค์กรที่ขับเคลื่อนธุรกิจด้วยข้อมูล(data driven organization) โดยใช้ข้อมูลการวิเคราะห์ประกอบกับการตัดสินใจในการให้บริการ

1.1.2 ภาพรวมของแผนกที่ดำเนินงาน

แผนกที่ได้เข้าไปดำเนินงานมีชื่อว่า Enterprise data services(EDS) ซึ่งมีบทบาทในการดูแล จัดการ เคลื่อนย้ายข้อมูลของทั้งองค์กร(data engineering) โดยมีหน้าที่ปฏิบัติตามคำร้องขอของผู้ใช้งาน โดยจัดทำระบบอัตโนมัติทั้งงานที่เป็นกิจวัตรและไม่เป็นกิจวัตรโดยแบ่งเป็น รายวัน รายเดือน และรายปี ซึ่งมีจุดประสงค์เพื่ออำนวยความสะดวกให้กับผู้ใช้งานเมื่อมีการร้องขอ

1.1.3 ระบบการดำเนินงานของบริษัทธนศึกษา

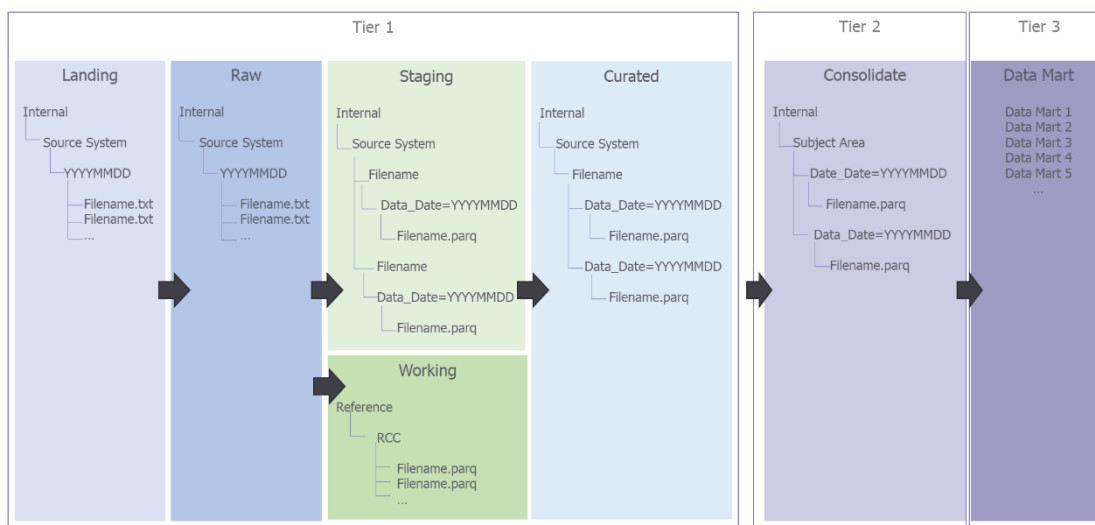
การดำเนินงานของแผนก EDS เป็นการดำเนินงานทางด้าน data engineer โดยดำเนินงานผ่าน Microsoft azure services ดังรูปที่ 1 โดยข้อมูลที่มีที่มาแตกต่างกันจะมีวิธีการนำเข้าข้อมูลที่แตกต่างกันโดยข้อมูลที่เป็นลักษณะ batch จะนำเข้าข้อมูลผ่าน azure data factory แต่ข้อมูลที่มีลักษณะ real-time จะนำเข้าข้อมูลผ่าน Event-hub โดยทำการจัดเก็บข้อมูลดิบ(Raw data) ที่ azure data lake storage และมี environment ที่ใช้สำหรับการแปลงข้อมูลที่ azure databricks และผู้ใช้งานสามารถใช้งานข้อมูลผ่านการแปลงแล้วได้ที่ azure synapse analytics



รูปที่ 1: azure services environment

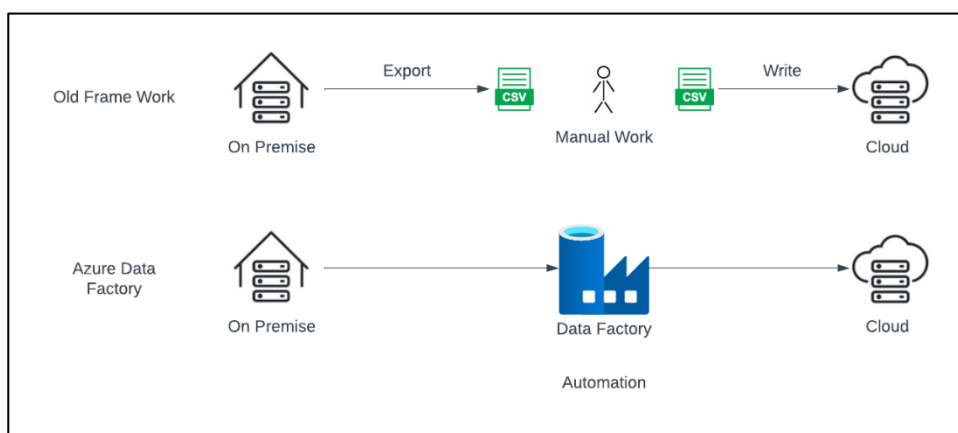
โดย data flow ของบริษัทกรณีศึกษาสามารถแบ่งออกได้เป็น 3 tiers ได้แก่ 1.data tier 1(Raw), 2.data tier 2(Consolidate) และ 3.data tier 3(Data Mart) โดย 1.data tier 1 จะเป็น tier ที่ข้อมูลเป็นลักษณะของข้อมูลดิบ(raw data) โดยมีได้หลากหลายรูปแบบทั้ง structure data, semi-structure data และ unstructured data เช่น ไฟล์ excel, ไฟล์ csv, ไฟล์ text เป็นต้น ซึ่งข้อมูลดิบเมื่อทำการนำเข้ามาที่ tier 1 แล้ว จะทำการแปลงข้อมูลให้อยู่รูปแบบไฟล์ parquet แล้วแปลงข้อมูลจากไฟล์ parquet ให้อยู่ในรูปแบบของตารางข้อมูล จากนั้นทำการลบข้อมูลดิบที่ไม่ใช่ไฟล์ parquet ออก หลังจากผ่านไป 30 วันโดยไฟล์ parquet มีประโยชน์คือเป็นไฟล์ที่ถูกทำการ encode ให้ไม่สามารถอ่านได้ด้วยมนุษย์ทำให้มีขนาดไฟล์ที่เล็กลงอย่างมาก รวมถึงทำหน้าที่เป็น Log file ในเวลาเดียวกันอีกทั้งการอ่านไฟล์ parquet แปลงเป็นตารางข้อมูลยังสามารถทำได้รวดเร็วกว่าข้อมูลดิบชนิดอื่นๆ 2.data tier 2 จะเป็น tier ที่ข้อมูลเป็นลักษณะของตารางข้อมูลที่มีการ join table, cast type, rename หรือผ่านการทำ standardize ของตารางข้อมูลในเบื้องต้น 3.data tier 3 จะเป็น tier ที่ข้อมูลเป็นลักษณะของตารางข้อมูลที่ผ่านมากระบวนการตรวจสอบหรือทำการ standardize โดยสมบูรณ์โดยจะมีลักษณะตารางข้อมูลที่มีชื่อเป็นมาตรฐาน ผ่านการตรวจสอบชนิดข้อมูลหรือมีการกำหนดช่วงเวลาของข้อมูลให้เป็นปัจจุบัน โดยจากรูปที่ 1 เราสามารถแบ่ง environment ในการดำเนินงานให้อยู่ใน data tiers ได้ดังต่อไปนี้

- 1.azure data lake storage จัดอยู่ใน data tier 1
- 2.azure databricks จัดอยู่ใน data tier 2
- 3.azure synapse analytics จัดอยู่ใน data tier 3



รูปที่ 2: Data tiers ของบริษัทกรณีศึกษา

โดยโครงการจะเป็นการออกแบบระบบงานที่มีชื่อว่า Common Reporting Standard(CRS) บน azure data factory โดย azure data factory ทำหน้าที่เป็นเครื่องมือที่ช่วยอำนวยความสะดวกสำหรับงานกิจวัตรทั้งกิจวัตรประจำวัน เดือน ปี โดยจะมีลักษณะเป็นท่อข้อมูล(data pipeline) ตัวอย่างเช่น การอัปเดตข้อมูลประจำวันของจำนวนรายชื่อผู้ถือบัญชีจากระบบฐานข้อมูลแบบ on premise ไปยังระบบฐานข้อมูลแบบ cloud โดยในแต่ละวันข้อมูลจะมีการเพิ่มลดของจำนวนรายชื่อผู้ถือบัญชีที่แตกต่างกัน หากระบบการทำงานยังไม่ได้ถูกพัฒนาใน azure data factory พนักงานมีความจำเป็นต้องส่งออกข้อมูลเป็นรูปแบบไฟล์ csv จาก on premise และนำข้อมูลไปยังระบบ cloud โดยทำเป็นประจำทุกวันด้วยตัวพนักงานเอง ซึ่ง azure data factory สามารถเข้ามาทำงานทดแทนการงานแบบวนซ้ำเหล่านี้ได้ โดยทำการออกแบบ control framework นี้ให้ดึงข้อมูลอัตโนมัติจาก on premise แล้วทำการอัปเดตข้อมูลแบบ incremental load ไปยังระบบ cloud เพื่ออัปเดตข้อมูลที่มีการเปลี่ยนแปลงในแต่ละวัน



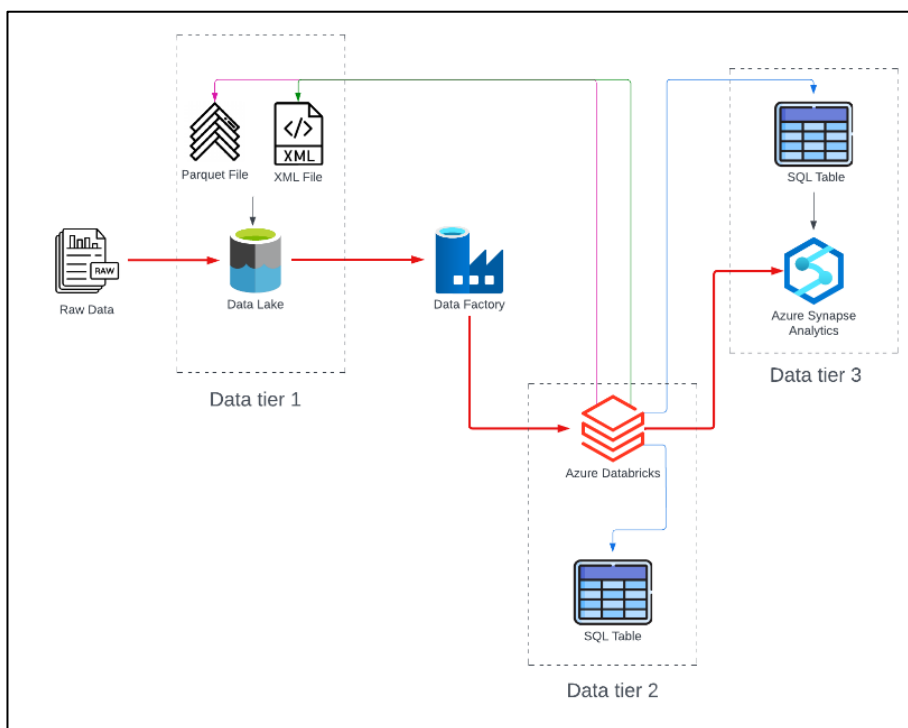
รูปที่ 3: บทบาทของ azure data factory

1.1.4 ที่มาของโครงการ Common Reporting Standard (CRS)

ในปี 2560 ประเทศไทยได้เข้าเป็นสมาชิกกรอบความร่วมมือเกี่ยวกับการแลกเปลี่ยนข้อมูลทางภาษี (Global Forum on Transparency and Exchange of Information for Tax Purposes หรือ Global Forum) ขององค์การเพื่อความร่วมมือทางเศรษฐกิจและการพัฒนา (Organization for Economic Co-operation and Development : OECD) เพื่อเป็นการยืนยันเจตนารมณ์ของประเทศไทยที่พร้อมจะร่วมมือกับ OECD ในการป้องกันการหลบหลีกหรือ หลีกเลียงภาษีและมีการแลกเปลี่ยนข้อมูลทางภาษีเป็นไปตามมาตรฐานสากลเช่นเดียวกับประเทศอื่นๆ ซึ่งจะช่วยเพิ่มประสิทธิภาพและความโปร่งใส ในการบริหารจัดการเก็บภาษีของประเทศ ทั้งนี้ ในการเป็นสมาชิก Global Forum ดังกล่าว ประเทศไทยจะต้องดำเนินการยกระดับการแลกเปลี่ยนข้อมูลตามร้องขอ (Exchange of Information on Request: EOIR) และดำเนินการแลกเปลี่ยนข้อมูลแบบอัตโนมัติ (Automatic Exchange of Information: AEOI) รายปีตามที่ OECD กำหนด โดยข้อมูลที่ทำกรแลกเปลี่ยนมีชื่อว่า Common Reporting Standard (CRS) ซึ่งเป็นรายงานที่บันทึกข้อมูลทางภาษีในบัญชีต่างประเทศให้กับประเทศคู่สัญญาเป็นประจำรายปี โดยประเทศไทยได้กำหนดจะเริ่มแลกเปลี่ยนข้อมูลครั้งแรกภายในเดือน กันยายน 2566 (พัชรินทร์ เปลี่ยนสังข์ และ บัญชา คลังป่า, 2566)

1.1.5 ภาพรวมการดำเนินโครงการ Common Reporting Standard (CRS)

CRS control framework เป็นระบบการสร้างรายงาน CRS(XML file) ซึ่งเป็นการทำงานในรูปแบบรายปี โดยมีภาพรวมของการดำเนินงานคือนำเข้าข้อมูลดิบไปไว้ที่ azure data lake storage(data tier1) จากนั้นทำการแปลงข้อมูลดิบเป็นไฟล์ parquet แล้วแปลงข้อมูลจากไฟล์ parquet ให้อยู่ในรูปแบบของตารางข้อมูลโดยทำการ standardize ในเบื้องต้นแล้วจัดเก็บที่ azure databricks(data tier2) จากนั้นนำตารางข้อมูลในที่บันทึกอยู่ที่ azure databricks มาทำเป็นรายงาน CRS(XML file) และทำการจัดเก็บที่ azure data lake storage ขั้นตอนต่อไปคือทำการ standardize ตารางข้อมูลที่ azure databricks(data tier2) ให้สมบูรณ์แล้วทำการเคลื่อนย้ายไปยัง azure synapse analytics(data tier 3) โดยไฟล์รายงาน CRS คือผลลัพธ์หลักของโครงการ แต่ไฟล์ parquet และตารางข้อมูลที่ azure databricks และ azure synapse analytics คือผลลัพธ์จากการปฏิบัติตามระบบการทำงานหลักของบริษัท โดยการจัดเก็บตารางข้อมูลที่ azure databricks และใช้เป็นข้อมูลตั้งต้นในการสร้างรายงาน CRS นั้น เนื่องจากสภาพแวดล้อมของ azure databricks ถูกออกแบบมาสำหรับการแปลงข้อมูลโดยเฉพาะโดย azure databricks นั้นรองรับการทำงานของ algorithm ที่หลากหลายอีกทั้งยังเป็น collaboration environment ที่ให้คนในองค์กรสามารถเข้ามามีส่วนร่วมในการพัฒนา ทว่าที่ azure synapse analytics นั้นมีจุดประสงค์เพื่อใช้สำหรับการวิเคราะห์และสร้าง report สำหรับข้อมูลขนาดใหญ่ ด้วยจุดประสงค์การใช้งานที่ต่างกันจึงทำให้มีการแยกการจัดเก็บตารางข้อมูลออกเป็น 2 environment ดังกล่าว โดยไฟล์รายงานจะถูกจัดเก็บแยกจากไฟล์ parquet และไฟล์ข้อมูลดิบ ดังรูปที่ 3 โดยไฟล์ข้อมูลดิบจะถูกจัดเก็บที่ Landing และ/หรือ Raw ไฟล์ parquet จะถูกจัดเก็บที่ Staging และ/หรือ Curated ไฟล์รายงาน CRS จะถูกจัดเก็บที่ Working



รูปที่ 4: CRS control framework overview

1.2 วัตถุประสงค์การทำโครงการ

1.2.1 เพื่อสร้างระบบงาน CRS(CRS Control Framework) ให้กับผู้ใช้งาน(พนักงานแผนก EDS) โดยทำงานได้รวดเร็วและมีความสะดวกในการใช้งาน

1.3 ขอบเขตโครงการ

1.3.1 โครงการนี้ ใช้ข้อมูลจริงของบุคคลที่ทำธุรกรรมกับบริษัทกรณีศึกษา เป็นผลให้เปิดเผยเนื้อหาไม่ได้อย่างเด็ดขาด โดยจะมีตัวชี้วัด ได้แก่ 1.ในด้านคุณภาพ ระบบสามารถสร้างรายงาน Common Reporting Standard (CRS) ได้อย่างถูกต้องตามที่ OECD ได้กำหนด 2.ในด้านเวลา สามารถนำส่งรายงานได้ตามเวลาที่กำหนดไว้ โดยการออกแบบระบบจะต้องสอดคล้องไปกับระบบการทำงานของบริษัทและใช้ภายในบริษัทเท่านั้น

1.3.2 โครงการนี้ จำกัดขอบเขตเนื้อหาของข้อมูลที่ต้องรายงานเฉพาะที่เกี่ยวข้องกับ Common Reporting Standard (CRS) โดยช่วงเวลาของข้อมูลทั้งหมดจะใช้ข้อมูลความเคลื่อนไหวของบัญชีที่เข้าเกณฑ์ตั้งแต่อดีตจนถึงปัจจุบัน

1.3.3 โครงการนี้ ในส่วนของการออกแบบระบบ CRS นั้นจะใช้ทฤษฎีที่ได้ระบุไว้และจะประยุกต์ใช้ทฤษฎีให้สอดคล้องและเหมาะสมต่อระบบ CRS

1.3.4 โครงการนี้ ข้อมูลที่ใช้ในการพัฒนาเป็น Mock up data ของข้อมูลจริง

1.4. ผลผลิตที่คาดว่าจะได้รับ (Expected Output)

สามารถสร้างระบบงาน CRS(CRS Control Framework) ที่สามารถสร้างไฟล์รายงานได้อย่างถูกต้องโดยเป็นไปตามที่ OECD กำหนดและระบบงาน CRS สอดคล้องกับระบบการทำงานหลักของบริษัทประกันภัยศึกษา

1.5. ผลลัพธ์ที่คาดว่าจะได้รับ (Expected Outcome)

ผู้ใช้งานระบบ(พนักงานแผนก EDS) สามารถใช้งานระบบได้สะดวกและระบบใช้เวลาน้อยในการผลิตรายงาน CRS เพื่อให้พนักงานสามารถส่งต่อรายงาน CRS ให้ผู้ที่เกี่ยวข้องต่อไป

1.6 แผนการดำเนินงาน

จากข้อมูลของตารางที่ 1 แสดงให้เห็นถึงกระบวนการดำเนินงานและระยะเวลาในการทำโครงการโดยแบ่งเป็น 2 ช่วงได้แก่ช่วงนำเสนอหัวข้อโครงการโดยมีระยะเวลาคือเดือน มิถุนายน ถึง สิงหาคม และ ช่วงลงมือปฏิบัติโดยมีระยะเวลาคือเดือน กรกฎาคม ถึง พฤศจิกายน ปีพุทธศักราช 2566

| Activity | มิ.ย. | | | | ก.ค. | | | | ส.ค. | | | | ก.ย. | | | | ต.ค. | | | | พ.ย. | | | |
|-------------------------------------------------------------------|-------|---|---|---|------|---|---|---|------|---|---|---|------|---|---|---|------|---|---|---|------|---|---|---|
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| ศึกษา data architecture และการทำงานของ บริษัทและเลือกปัญหาที่สนใจ | | | | | | | | | | | | | | | | | | | | | | | | |
| ศึกษารายละเอียดของปัญหา | | | | | | | | | | | | | | | | | | | | | | | | |
| วางแผนการดำเนินงานวิจัย | | | | | | | | | | | | | | | | | | | | | | | | |
| ศึกษาทฤษฎีที่เกี่ยวข้อง | | | | | | | | | | | | | | | | | | | | | | | | |
| นำเสนอโครงร่างงานวิจัย | | | | | | | | | | | | | | | | | | | | | | | | |
| Proposal report | | | | | | | | | | | | | | | | | | | | | | | | |
| ศึกษา Business requirement ของงานวิจัย | | | | | | | | | | | | | | | | | | | | | | | | |
| พัฒนา Prototype | | | | | | | | | | | | | | | | | | | | | | | | |
| ออกแบบ CRS Data Architecture | | | | | | | | | | | | | | | | | | | | | | | | |
| รวบรวม, Algorithm optimization และสร้าง CRS control framework | | | | | | | | | | | | | | | | | | | | | | | | |
| SIT: จัดทำ Testcase และทำการ validate XML | | | | | | | | | | | | | | | | | | | | | | | | |
| UAT: จัดทำ Dashboard และทำการวัด User Satisfaction | | | | | | | | | | | | | | | | | | | | | | | | |
| จัดทำรายงาน | | | | | | | | | | | | | | | | | | | | | | | | |
| สื่อนำเสนอ Final Project | | | | | | | | | | | | | | | | | | | | | | | | |
| Senior Project | | | | | | | | | | | | | | | | | | | | | | | | |

ตารางที่ 1: แผนการดำเนินงานของงานวิจัย

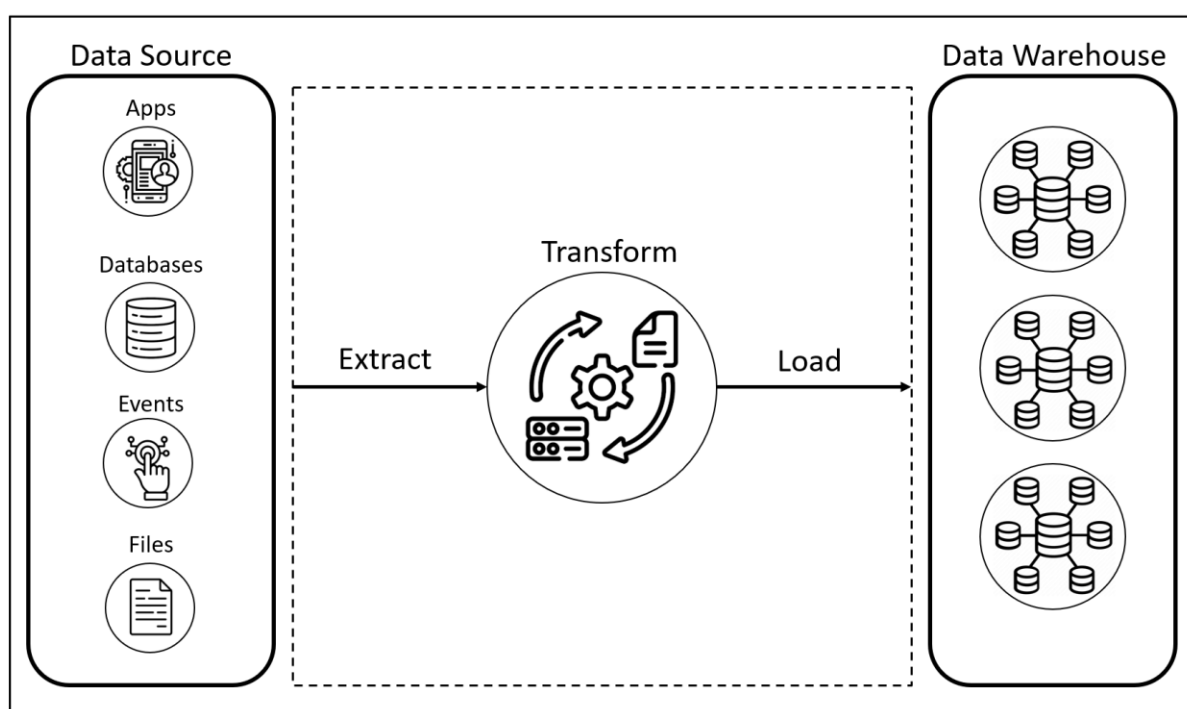
บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องที่ใช้ในโครงการ Common Reporting Standard(CRS) เพื่อสร้างรายงานให้มีความสมบูรณ์ จะพิจารณาทั้งด้าน แนวคิด วิธีการ และคุณสมบัติซึ่งจะประกอบไปด้วย 5 ทฤษฎี ได้แก่

2.1. ETL (Extract-Transforms-Load)

ETL Definition : ETL คือขั้นตอนในกระบวนการประมวลผลข้อมูลในระบบฐานข้อมูลที่มีการกระจายข้อมูล (Distributed Database System) หรือระบบฐานข้อมูลที่มีข้อมูลจำนวนมาก (Big Data) ซึ่งมาจากคำย่อของคำว่า Extract, Transform, Load หรือการดึงข้อมูล (Extract) การแปลงข้อมูล (Transform) และการโหลดข้อมูล (Load) ดังแสดงในรูป



รูปที่ 5: กระบวนการ Extract-Transforms-Load

ที่มา fusionsol (2566)

โดยมีจุดมุ่งหมายของการดำเนินการคือ นำข้อมูลดิบที่ได้จากการรวบรวมจากแหล่งที่มาต่างๆ ตัวอย่างเช่น เว็บไซต์ หน่วยงาน หรือ ฐานข้อมูล มาทำการเปลี่ยนแปลงตามจุดประสงค์ที่ต้องการแล้วโหลดข้อมูลไปไว้ในที่ที่ผู้ใช้งานสามารถใช้งานได้ง่ายไม่ซับซ้อน โดยข้อมูลที่ผ่านมากระบวนการนี้ต้องมีลักษณะตรงตามที่ต้องการแล้วมีคุณภาพของข้อมูลที่ดี (Lung Yiu & Vista, 2008)

2.1.1 Extraction

1. Extract Explained : Extract (ดึงข้อมูล): ขั้นตอนแรกของกระบวนการ ETL คือการดึงข้อมูลจากแหล่งที่เก็บข้อมูลต้นทาง เช่น ฐานข้อมูล, ไฟล์ข้อมูล, API, หรือแหล่งข้อมูลอื่น ๆ ข้อมูลที่ถูกดึงมาจะอยู่ในรูปแบบที่ไม่ได้ถูกจัดระเบียบเพื่อการวิเคราะห์หรือนำไปใช้งาน การดึงข้อมูลเป็นขั้นตอนที่สำคัญ เนื่องจากต้องให้แน่ใจว่าข้อมูลถูกดึงมาในรูปแบบที่ถูกต้องและเติมเต็มความต้องการ(Lung Yiu & Vista, 2008)

2. Extract architecture design

2.1.Full extraction. การดึงข้อมูลแบบ Full extraction ในแต่ละครั้งจะเป็นการดึงข้อมูลทั้งหมดในครั้งเดียว มักใช้ในขั้นตอนแรกของการเริ่มกระบวนการ ETL หรือข้อมูลนั้นเป็นลักษณะ shot live data หรือ ขนาดของไฟล์ข้อมูลไม่ได้ใหญ่มาก เป็นผลให้ข้อมูลที่ได้นั้นมีความสดใหม่ แต่ในกรณีที่ข้อมูลมีขนาดใหญ่อาจส่งผลกระทบต่อทรัพยากรมากในการนำเข้าสู่ข้อมูลแต่ละครั้ง

2.2.Incremental extraction. การดึงข้อมูลแบบ Incremental extraction ในแต่ละครั้งจะเป็นการดึงเฉพาะข้อมูลที่เป็น new data หรือ ข้อมูลที่ถูกเปลี่ยนแปลงไปเท่านั้นเช่น โดยตัวอย่างการดึงข้อมูลแบบ Incremental extraction เช่น การดึงข้อมูลโดย API โดยการดึงข้อมูลลักษณะนี้เหมาะกับ ข้อมูลที่มีความสำคัญโดยส่วนใหญ่ และมีข้อดีคือมีความสมดุลระหว่างข้อมูลที่สดใหม่และใช้ข้อมูลการดึงในแต่ละครั้งน้อย

2.3.Source-driven extraction. การดึงข้อมูลแบบ Source-driven extraction ในแต่ละครั้งจะเป็นการดึงข้อมูลที่เปลี่ยนแปลงไป เหมาะสำหรับข้อมูลที่เปลี่ยนแปลงบ่อยและความสำคัญไม่มาก โดยประโยชน์ของวิธีนี้ ใช้ข้อมูลการดึงในแต่ละครั้งน้อย แต่ได้ข้อมูลเร็วและคุณภาพข้อมูลที่จะตกลง

2.1.2 Transformation

1. Transformation Explained : Transform (แปลงข้อมูล): หลังจากข้อมูลถูกดึงมาแล้ว เราต้องนำข้อมูลมาทำการแปลงเพื่อให้ข้อมูลเหมาะสมและใช้งานได้(Lung Yiu & Vista, 2008) โดยในขั้นตอนนี้ประกอบไปด้วยหลากหลายกระบวนการ ดังนี้

1.Data cleaning: การทำความสะอาดข้อมูลมีด้วยกันหลากหลายวิธี ตัวอย่างเช่น:

- 1.1 กำจัด missing data ตัวอย่างเช่น NULL NA เป็นต้น
- 1.2 กำจัด outliers
- 1.3 เปลี่ยนข้อมูล ตัวอย่างเช่น เปลี่ยน “M”, 1, “male”, เป็น “Male”

2.Data enriching: การเพิ่มคุณค่า/มูลค่าให้แก่ข้อมูลที่เก็บมา ตัวอย่างเช่น:

- 2.1 นำข้อมูลไป join เข้ากับฐานข้อมูลอื่นๆ
- 2.2 กำจัดข้อมูลที่ซ้ำออกไป

2.3 สร้าง fields ใหม่ที่เป็นประโยชน์เพิ่มเข้ามา ตัวอย่างเช่น คำนวณ Revenue จาก

$$\text{Revenue} = \text{Unit Price} \times \text{Unit Amount}$$
 เป็นต้น

2. Transformation architecture design

1.Order of operations. การออกแบบให้ทำการแปลงข้อมูลตามลำดับที่ถูกต้อง เป็นปัจจัยสำคัญที่ส่งผลต่อผลลัพธ์สุดท้ายของข้อมูล การต่อเรียงการแปลงข้อมูลที่ไม่ถูกต้องอาจทำให้ผลลัพธ์ของการคำนวณหรือการแปลงข้อมูลเปลี่ยนแปลงไปจากที่คาดคิดไว้

2.Business logic. บ่อยครั้งการ Transforms มักจะมี Business logic เข้ามาเกี่ยวข้อง เช่น customer's lifetime value ดังนั้น architecture จำเป็นต้องได้รับการออกแบบเพื่อให้สามารถจัดการกับข้อมูลที่ เป็น missing value หรือ corrupt value ทำให้ algorithm ที่ออกแบบมานั้นต้องมีประสิทธิภาพ เพราะต้องรับ load ของข้อมูลจำนวนมาก

3.Algorithmic efficiency. ในการออกแบบการ transforms นั้นต้องมี Algorithmic ที่มีประสิทธิภาพ เพราะ algorithm ที่ต่างกันจะส่งผลถึง time consumption ที่ต่างกัน ตัวอย่างเช่น การแปลง data 1 ล้านแถว โดยใช้ dictionary มีประสิทธิภาพมากกว่าการแปลงโดยใช้ For loop เป็นต้น

4.Quality assurance. บ่อยครั้งหลังการ Transforms ข้อมูลที่ได้มามักได้รับการตรวจสอบความถูกต้องตามเกณฑ์ที่กำหนด (เช่น ห้ามนำเข้าข้อมูลลูกค้าเว้นแต่เราจะมีอีเมลของพวกเขา) และติดตามเพื่อคุณภาพ หรือ encrypt ข้อมูลก่อนนำไปใช้จริง

2.1.3 Loading

1. Loading Explained : Load (โหลดข้อมูล): ขั้นตอนสุดท้ายของ ETL คือการนำข้อมูลที่ถูกลบแล้วมาเก็บไว้ในระบบฐานข้อมูลปลายทาง เพื่อให้ผู้ใช้งานสามารถเข้าถึงและใช้งานข้อมูลได้ ข้อมูลที่โหลดเข้าไปในระบบฐานข้อมูลปลายทาง อาจถูกจัดเก็บในรูปแบบของตาราง (Table) หรือโครงสร้างข้อมูลที่เหมาะสมกับการนำไปใช้งานต่อไป(Lung Yiu & Vista, 2008)

2. Loading architecture design : การ Load มีวิธีหลักด้วยกัน 3 วิธี ดังนี้

1.Full load. การใช้งานหลักๆคือใช้สำหรับ โหลดข้อมูลทั้งหมดลงฐานข้อมูลในครั้งเดียว ซึ่งมีข้อดีคือไม่ยุ่งยากในการทำงาน แต่ต้องการเวลาและทรัพยากรจำนวนมาก

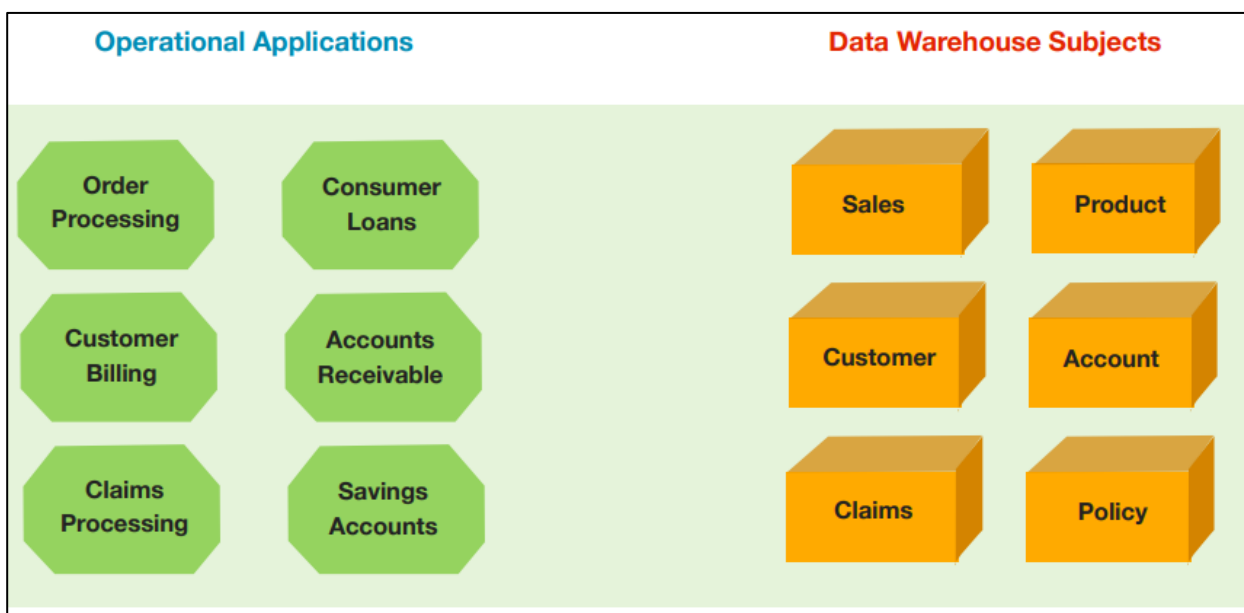
2.Incremental batch load การใช้งานหลักๆคือใช้สำหรับโหลดข้อมูลบางส่วนฐานข้อมูล ซึ่งมีข้อดีคือใช้เวลาและทรัพยากรน้อย แต่ต้องการทักษะการเขียนโปรแกรมในการโหลดข้อมูลในระดับกลาง

3.Incremental stream load การใช้งานหลักๆคือใช้สำหรับโหลดข้อมูลเวลาที่ข้อมูลใหม่เกิดขึ้นมาหรือข้อมูลเก่าต้องทำการอัปเดต ซึ่งมีข้อดีคือใช้เวลาและทรัพยากรน้อย แต่ต้องการทักษะการเขียนโปรแกรมในการโหลดข้อมูลในระดับสูง

2.2. Database

2.2.1 Data warehouse Definition : Data Warehouse หรือคลังข้อมูล เป็นระบบจัดการข้อมูลประเภทหนึ่ง ซึ่งออกแบบมาเพื่อสนับสนุนการสร้าง Business Intelligence (BI) และการวิเคราะห์ข้อมูลโดยเฉพาะ นอกจากนั้นมันยังมีความสามารถเรียก/สืบค้นข้อมูล (Query) และรวบรวมข้อมูลในอดีต (Historical data) จำนวนมาก โดยข้อมูลภายในมักมาจากแหล่งข้อมูลที่หลากหลาย เช่น ล็อกไฟล์ รายการข้อมูลธุรกรรม เป็นต้น Data Warehouse จะมีคุณสมบัติอยู่ 5 อย่างดังนี้

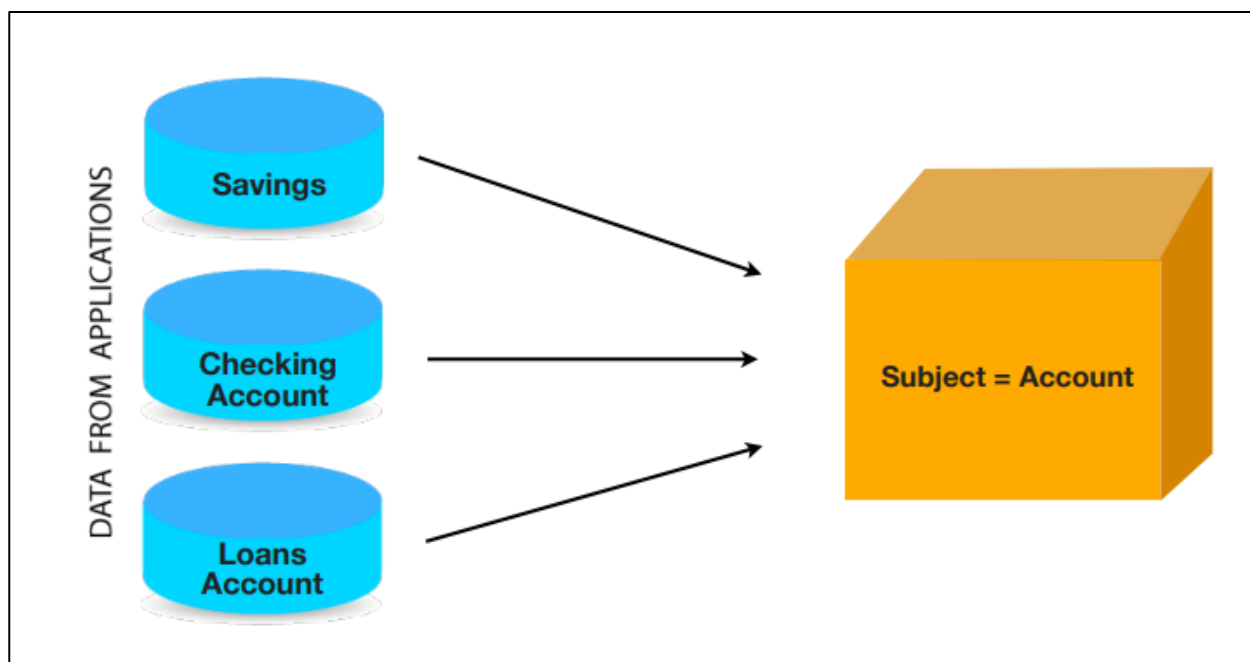
1. Subject-oriented: ข้อมูลถูกจัดเก็บตามหัวข้อที่สนใจ โดยข้อมูลในคลังข้อมูลจะถูกเก็บและเชื่อมโยงด้วยหัวข้อทางธุรกิจ(Business subject) ตัวอย่างเช่น บริษัทผู้ผลิตสินค้า จะมีความเกี่ยวเนื่องกับข้อมูลการผลิตสินค้า การขายสินค้า การส่งสินค้า การจัดเก็บสินค้าเข้าสู่คลังสินค้า เป็นต้น



รูปที่ 6: การเปรียบเทียบการจัดข้อมูลระหว่าง Operational database กับ Datawarehouse

ที่มา โกเมศ อัมพวัน (2566)

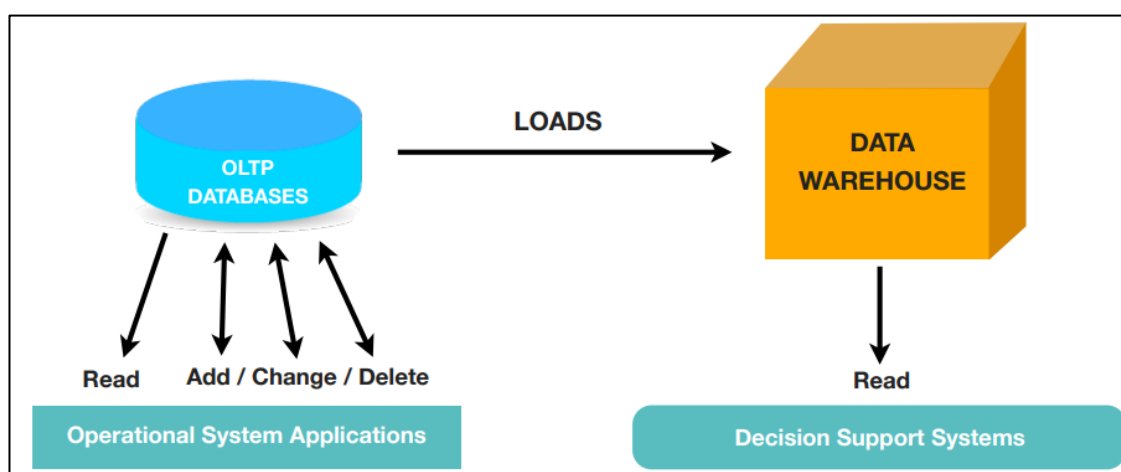
2. Integrated: ข้อมูลที่ถูกกรวบรวมมาจากหลายแหล่งข้อมูล โดยข้อมูลที่มาจากหลายระบบอาจมีความแตกต่างกันในเรื่องของระบบการจัดการฐานข้อมูลที่ใช้ แฟ้มข้อมูล หรือการจัดเก็บข้อมูลส่วนย่อยๆ ที่มีความแตกต่างในเรื่องของโครงสร้างของแฟ้มข้อมูล ตัวอย่างเช่น ในบริษัทผู้ผลิตสินค้า สำหรับสินค้าหนึ่งๆ รหัสการขายสินค้า อาจจะเป็น sale.001 ส่วนรหัสการส่งสินค้าอาจจะเป็น ship.001 แต่ว่าทั้ง 2 รหัสนี้หมายถึงสินค้าตัวเดียวกัน แต่หากจะต้องทำการนำข้อมูลเข้าสู่คลังข้อมูล จำเป็นต้อง transforms เป็นรูปแบบเดียวกันเช่น prod.001 เป็นต้น



รูปที่ 7: การรวมกันของข้อมูล

ทีมา โกเมศ อัมพวัน (2566)

3. Time-variant: ข้อมูลที่เกี่ยวข้องกับช่วงเวลาต่างๆ โดยในคลังข้อมูลจะต้องทำการเก็บข้อมูลที่เป็นปัจจุบัน และข้อมูลย้อนหลัง โดยมีแกนเวลาเข้ามาเกี่ยวข้องซึ่งจะทำให้ผู้ใช้ทราบความเปลี่ยนแปลงของข้อมูลต่อช่วงเวลาต่างๆได้
4. Nonvolatile: ข้อมูลที่ไม่เปลี่ยนแปลง เป็นที่ทราบกันดีว่าใน operational database จะอนุญาตให้สามารถกระทำการใดๆ ในฐานข้อมูลได้ด้วยอย่างเช่น select, insert, delete และ update แต่ว่าในคลังข้อมูล (Datawarehouse) จะไม่สามารถกระทำการใดๆได้เนื่องจากการ select เท่านั้น



รูปที่ 8: คุณสมบัติการไม่เปลี่ยนแปลงของข้อมูลในคลังข้อมูล

ทีมา โกเมศ อัมพวัน (2566)

5. Granularity: ข้อมูลที่มีรายละเอียดหลายระดับ ตัวอย่างเช่น ในบริษัทผู้ผลิตสินค้า ความละเอียดของข้อมูลจะหมายถึง ข้อมูลยอดขายของสินค้าชนิดหนึ่งในสาขาหนึ่งในภาคตะวันออก เป็นต้น

2.2.2 Entity Relationship Diagram (ER Diagram)

ER (Entity-Relationship) diagram เป็นเครื่องมือในการออกแบบฐานข้อมูลที่ใช้ในการแสดงความสัมพันธ์ระหว่าง Entity (ข้อมูลหรือองค์กร) และ Relationship (ความสัมพันธ์) โดยใช้สัญลักษณ์และรูปแบบที่กำหนดตามข้อกำหนดของแบบจำลองข้อมูลส่วนต่อประสาน (Entity-Relationship Model)

2.3 Big O Notation

Big O notation คือ สัญกรณ์ทางคณิตศาสตร์ที่ใช้ในทาง Computer Science และ คณิตศาสตร์เพื่ออธิบายพฤติกรรมของ algorithm หรือ function ซึ่งมีส่วนช่วยในการวิเคราะห์ประสิทธิภาพและศักยภาพของ algorithm โดยกำหนดขอบเขตอัตราการเติบโต (เวลา) ของ algorithm ที่สัมพันธ์กับขนาดของข้อมูลที่ใช้ในการประมวลผล โดยตัว notation เองจะถูกกำหนดด้วย $O(g(n))$ โดย 'O' หมายถึง "order of" ซึ่งหมายถึงอัตราการเติบโตของประสิทธิภาพของ algorithm ที่เกี่ยวเนื่องกับขนาด input (Massachusetts Institute of Technology, 2022) โดยมีสมการดังนี้

$$f(n) = O(g(n)) \quad \dots (1)$$

โดย

f & g คือ จำนวนจริงที่เป็นบวก, $\mathbb{R} \in [0, \infty)$

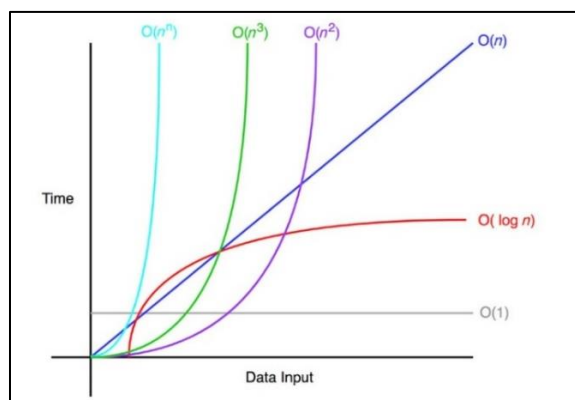
$g(n)$ คือความซับซ้อนของ algorithm ที่ขึ้นกับ Input Size

n คือ Input Size ซึ่งเป็นจำนวนนับที่เป็นบวก $\mathbb{N} \in [1, \infty)$

และมีลักษณะของ Order of ต่างๆดังนี้

1. $O(1)$ - Constant time complexity: ประสิทธิภาพของ algorithm คงที่ โดยไม่คำนึงถึงขนาดของ input ตัวอย่างเช่น การเข้าถึง element ใน array โดยใช้ index
2. $O(\log n)$ - Logarithmic time complexity: ประสิทธิภาพของ algorithm เพิ่มขึ้นเป็นลักษณะ logarithmic ตามขนาดของ input โดยมักจะพบใน algorithm ที่สามารถแบ่งปัญหาออกเป็นส่วนย่อยๆ ตัวอย่างเช่น Binary search ใน sorted array.
3. $O(n)$ - Linear time complexity: ประสิทธิภาพของ algorithm เพิ่มขึ้นเป็นลักษณะ linear ตามขนาดของ input ซึ่งหมายความว่า runtime นั้นจะเกี่ยวข้องโดยตรงกับขนาดของ input ตัวอย่างเช่น For loop ใน list
4. $O(n \log n)$ - Linearithmic time complexity: ประสิทธิภาพของ algorithm เพิ่มขึ้นโดยอยู่ระหว่าง linear และ logarithmic โดยปกติแล้วมักพบเจอใน sorting algorithms ตัวอย่างเช่น Merge Sort และ Quick Sort

5. $O(n^2)$ - Quadratic time complexity: ประสิทธิภาพของ algorithm เพิ่มขึ้นเป็นลักษณะ quadratically ตามขนาดของ input หมายความว่า runtime ของ algorithm จะเป็นกำลังสองของขนาด input ตัวอย่างเช่น Nested loops
6. $O(n^c)$ - Polynomial time complexity: ประสิทธิภาพของ algorithm เพิ่มขึ้นเป็นลักษณะ polynomial ตามขนาดของ input ยกกำลัง c โดย c คือเลขชี้กำลัง ตัวอย่างเช่น Bubble Sort ด้วย $O(n^2)$.
7. $O(2^n)$ - Exponential time complexity: ประสิทธิภาพของ algorithm เพิ่มขึ้นเป็นลักษณะ exponentially ตามขนาดของ input โดยส่วนมากมักพบใน recursive algorithm ที่มี subproblems จำนวนมาก ตัวอย่างเช่น Fibonacci sequence calculation
8. $O(n!)$ - Factorial time complexity: ประสิทธิภาพของ algorithm เพิ่มขึ้นเป็นลักษณะ factorial ตามขนาดของ input โดยส่วนมากพบใน algorithm ที่พยายามหา all possible combinations ตัวอย่างเช่น brute-force algorithms

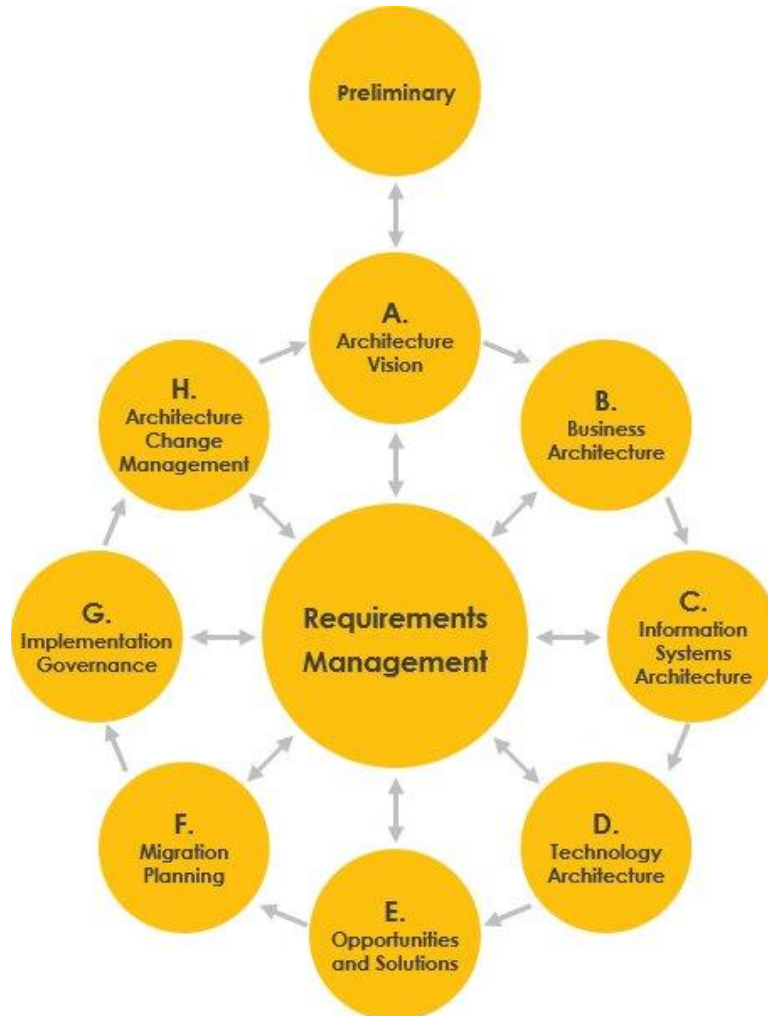


รูปที่ 9: ความสัมพันธ์ของ runtime และ input size ระหว่าง Order of ต่างๆ

ที่มา DroidTechKnow (2566)

2.4 The Open Group Architecture Framework (TOGAF)

TOGAF คือ enterprise architecture methodology ที่สามารถนำมาเป็นแนวทางในการออกแบบ data architecture ได้ (The Open Group Standard, 2018) โดยมี 9 หลักการในการออกแบบดังรูปที่ 10







































รูปที่ 10: 9 Principle of Enterprise Architecture by TOGAF
ที่มา TOGAF Open Group (2018)

2.5 Zachman Framework for Enterprise Architecture

Zachman จะให้ classification schema โดยประกอบไปด้วยส่วนสำคัญคือคอลัมต์ 6 คอลัมต์ซึ่งประกอบไปด้วย

- 1 What - Data and information representation (Ontological perspective).
- 2 How - Processes and activities (Functional perspective).
- 3 Where - Locations and distribution (Geographical perspective).
- 4 Who - People and roles (Organizational perspective).
- 5 When - Time and schedules (Temporal perspective).
- 6 Why - Motivations and goals (Contextual perspective)

ซึ่งเป็นตัวยึดหลักและจะทำการเปรียบเทียบควบคู่ไปกับ Artifacts(แถว) หรือคือสิ่งประดิษฐ์หรือแบบจำลองที่เกี่ยวข้องที่ได้รับการพัฒนาขึ้นมา แล้วตอบคำถามในแต่ละเซลล์เพื่อให้เป็นแนวทางที่สมบูรณ์ในการระบุ data Architecture (Zachman, 2003) โดย Zachman framework เองนั้นสามารถใช้ควบคู่กับ TOGAF ทำให้การออกแบบ data architecture นั้นสมบูรณ์มากยิ่งขึ้น

| | WHAT | HOW | WHERE | WHO | WHEN | WHY | |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|
| SCOPE CONTEXTS | Inventory Identification  Inventory Types | Process Identification  Process Types | Network Identification  Network Types | Organization Identification  Organization Types | Timing Identification  Timing Types | Motivation Identification  Motivation Types | STRATEGISTS AS THEORISTS |
| BUSINESS CONCEPTS | Inventory Definition  Business Entity Business Relationship | Process Definition  Business Transform Business Input | Network Definition  Business Location Business Connection | Organization Definition  Business Role Business Work | Timing Definition  Business Cycle Business Moment | Motivation Definition  Business End Business Means | EXECUTIVE LEADERS AS OWNERS |
| SYSTEM LOGIC | Inventory Representation  System Entity System Relationship | Process Representation  System Transform System Input | Network Representation  System Location System Connection | Organization Representation  System Role System Work | Timing Representation  System Cycle System Moment | Motivation Representation  System End System Means | ARCHITECTS AS DESIGNERS |
| TECHNOLOGY PHYSICS | Inventory Specification  Technology Entity Technology Relationship | Process Specification  Technology Transform Technology Input | Network Specification  Technology Location Technology Connection | Organization Specification  Technology Role Technology Work | Timing Specification  Technology Cycle Technology Moment | Motivation Specification  Technology End Technology Means | ENGINEERS AS BUILDERS |
| COMPONENT ASSEMBLIES | Inventory Configuration  Component Entity Component Relationship | Process Configuration  Component Transform Component Input | Network Configuration  Component Location Component Connection | Organization Configuration  Component Role Component Work | Timing Configuration  Component Cycle Component Moment | Motivation Configuration  Component End Component Means | TECHNICIANS AS IMPLEMENTERS |
| OPERATIONS CLASSES | Inventory Instantiation  Operations Entity Operations Relationship | Process Instantiation  Operations Transform Operations Input | Network Instantiation  Operations Location Operations Connection | Organization Instantiation  Operations Role Operations Work | Timing Instantiation  Operations Cycle Operations Moment | Motivation Instantiation  Operations End Operations Means | WORKERS AS PARTICIPANTS |
| | INVENTORY SETS | PROCESS TRANSFORMATIONS | NETWORK NODES | ORGANIZATION GROUPS | TIMING PERIODS | MOTIVATION REASONS | |

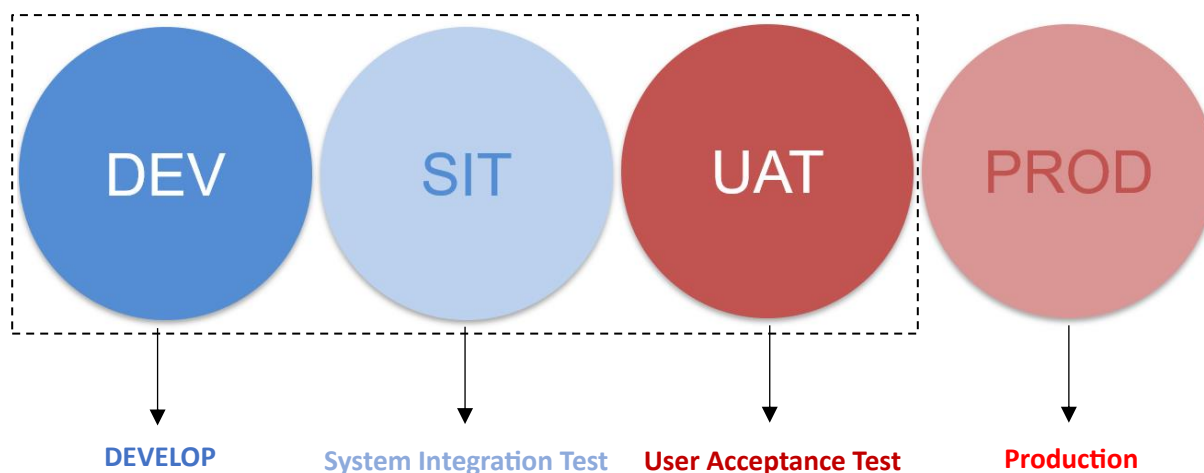
ตารางที่ 2: การดำเนินการตามกระบวนการ Zachman Framework
ที่มา Visual Paradigm (2023)

บทที่ 3

วิธีการดำเนินงานวิจัย

3.1 ระยะการดำเนินโครงการ

ระยะการดำเนินงานวิจัยประกอบไปด้วย 3 จาก 4 ขั้นตอนของกระบวนการ QA Testing ดังแสดงในรูปที่ 11 โดยขอบเขตการดำเนินงานจะเริ่มจาก DEV ไป SIT และจบที่เฟส UAT โดยในเฟส DEV นั้นหากทำการสร้าง CRS control framework เรียบร้อยแล้วจะทำการไปยังเฟส SIT โดยในเฟสนี้หลังจากทำการตรวจสอบรายงานด้วย XML Schema เรียบร้อยจะไปยังที่เฟส UAT โดยในเฟสนี้จะทำการตรวจสอบโครงสร้างและทำการ standardize ตารางข้อมูลอย่างสมบูรณ์ จากนั้นทำการจัดเก็บใน azure synapse analytics ตามระบบการดำเนินงานหลักของบริษัทต่อไป



รูปที่ 11: Phase การดำเนินงานของ CRS Project

3.2 Develop

ในขั้นตอนของ Develop นั้นมีจุดประสงค์เพื่อสร้างระบบ CRS control framework บน azure data factory โดยประกอบไปด้วย 4 อย่างได้แก่ 1.Business requirement analysis 2.Prototype (Transform algorithm) 3.Design architecture 4.Combination and Create CRS control framework

3.2.1 Business requirements analysis

ทำความเข้าใจของรูปแบบรายงานที่ต้องการและเงื่อนไขต่างๆที่ต้องนำมาพิจารณาในการสร้างรายงานที่ต้องการ โดยทำการศึกษา requirement document ของงานเพื่อระบุถึงรายละเอียดของเนื้อหาเพื่อที่จะพัฒนารายงานและซอฟต์แวร์ทางธุรกิจ การทำความเข้าใจความต้องการของผู้มีส่วนเกี่ยวข้อง(หน่วยงานจัดเก็บภาษีของประเทศในพันธกิจ) จึงมีสำคัญส่วนสำคัญที่จะทำให้การพัฒนารายงานและซอฟต์แวร์ทางธุรกิจเป็นไปตามความต้องการและถูกต้องตามหลักสากลตัวอย่างเช่น จากรูปที่ 12 พบว่า OrganizationParty_Type

ประกอบไปด้วย RescountryCode ซึ่งมีลักษณะข้อมูลเป็นตัวอักษร 2 ตัว โดยสามารถมีหรือไม่มีในองค์ประกอบย่อยก็ได้, IN และ Name เป็นตัวอักษรไม่เกิน 200 ตัว โดย IN สามารถมีหรือไม่มีในองค์ประกอบย่อยที่ทำการสร้างในรายงานได้ แต่ Name ต้องมีอยู่ในองค์ประกอบย่อยที่ทำการสร้างรายงาน สุดท้าย Address จะเป็นตัวที่สามารถแยกย่อยต่อไปได้โดยสังเกตจาก Size ที่ไม่ได้มีการระบุไว้ โดยทำซ้ำทุกองค์ประกอบเพื่อให้สามารถระบุลักษณะข้อมูลได้อย่างถูกต้อง เป็นต้น อีกทั้งยังต้องทำการศึกษาภาพรวมของรูปแบบไฟล์ตามที่ OECD ได้กำหนด โดยทำการระบุเงื่อนไขสำคัญทุกอันที่สามารถส่งผลกระทบต่อรูปแบบของไฟล์รายงานตัวอย่างเช่น หาก MessageRefId เป็น “CRS703” จะส่งผลให้ส่วนของ Crsbody ไม่ต้องทำการรายงานเป็นผลให้รายงานนั้นไม่ต้องทำการนำเสนอ (Organization for Economic Co-operation and Development , 2019) เป็นต้น

III. OrganisationParty_Type

This complex type identifies the name of an Account Holder that is an Entity as opposed to an Individual.

It is comprised of the following four data elements:

| Element | Attribute | Size | Input Type | Requirement |
|----------------|-----------|---------------------|---------------------------|----------------------|
| ResCountryCode | | 2-character | iso:CountryCode_Type | (Optional) Mandatory |
| Element | Attribute | Size | Input Type | Requirement |
| IN | | 1 to 200 characters | crs: OrganisationIN_Type | (Optional) Mandatory |
| Element | Attribute | Size | Input Type | Requirement |
| Name | | 1 to 200 characters | cfc:NameOrganisation_Type | Validation |
| Element | Attribute | Size | Input Type | Requirement |
| Address | | | cfc:Address_Type | Validation |

รูปที่ 12: ตัวอย่าง requirement ของ OrganizationParty_Type
ที่มา OECD guide book (2019)

3.2.2 Phase I : Prototype(Transform algorithm)

เมื่อดำเนินการในขั้นตอนของ Business requirement เรียบร้อยแล้ว จะทำการวิเคราะห์องค์ประกอบของข้อมูลแล้วออกแบบ Algorithms สำหรับการ Transforms data โดยเน้นให้สามารถแก้ไขได้ง่ายหากพบปัญหาเพิ่มหรือลดองค์ประกอบได้ง่าย และใช้งานได้อย่างมีประสิทธิภาพ โดยขั้น Prototype จะทำการออกแบบโดยยึดข้อมูล 3 ประเภทดังแสดงในตารางที่ 3 โดยในแต่ละประเภทจะมีการรายงานในรูปแบบ XML ที่แตกต่างกันไปซึ่งจะมี Flag Column ในการบอกประเภทได้แก่ Institution_under_CRS และ Passive โดยประเภทบัญชีของบุคคลธรรมดาจะไม่มีการระบุข้อมูลใน Flag Column สำหรับนิติบุคคลที่ไม่อยู่ในภายใต้ CRS จะถูกระบุด้วย NO และ NO และ นิติบุคคลแบบ Passive จะถูกระบุด้วย NO หรือ YES ในทั้ง 2 flag column โดยทำการสร้าง transform notebook บน azure databrick โดยใช้ Pandas และ pyspark library ในการออกแบบ algorithm ซึ่งจะถูกใช้ใน CRS control framework บน azure data factory ต่อไป

| | | |
|---------------------------------|-----------------------|---------|
| Data Type Report | Institution_under_CRS | Passive |
| บุคคลธรรมดา | N/A | N/A |
| นิติบุคคลที่ไม่อยู่ในภายใต้ CRS | NO | NO |
| นิติบุคคลแบบ Passive | NO/YES | NO/YES |

ตารางที่ 3: ชนิดของข้อมูลที่เกี่ยวข้องในการรายงาน

3.2.3 Phase II : Design architecture

สืบเนื่องจากกระบวนการทำงานซึ่งประกอบไปด้วยหลากหลายองค์ประกอบจึงจำเป็นต้องออกแบบ CRS Architecture เพื่อใช้เป็นโครงสร้างหลักของการทำงานโดย CRS Architecture จะมีหน้าที่อธิบายว่ากระบวนการทำงานนั้นมีวิธีการจัดการข้อมูล ตั้งแต่ collection(การเก็บรวบรวมข้อมูล), transformation(การแปลงข้อมูล), distribution(การแยกข้อมูลหรือเคลื่อนย้ายข้อมูล), and consumption(การนำข้อมูลไปใช้ประโยชน์) อย่างไร โดยทำหน้าที่เสมือนพิมพ์เขียวที่เป็นแนวทางสำหรับการทำงาน ซึ่งเป็นพื้นฐานของการดำเนินการประมวลผลข้อมูลโดยใช้ TOGAF Framework ในการออกแบบโดยจะทำการ simplify ให้เหมาะกับ CRS control framework เนื่องด้วยแต่เดิมนั้น TOGAF Framework เป็น Framework ที่ใช้สำหรับการออกแบบระดับ Enterprise ทว่าในส่วนของ TOGAF principle เป็นส่วนสำคัญที่จะช่วยระบอบองค์ประกอบต่างๆในโครงการให้ครบและไม่ตกหล่น ทำให้โปรเจกต์มีความชัดเจนว่าทำไปเพื่ออะไร ทำอย่างไร ทำแล้วได้อะไร ทำให้ใคร ทำแล้วได้ประโยชน์อย่างไร

3.2.4 Phase III : Combination and Create CRS Control Framework

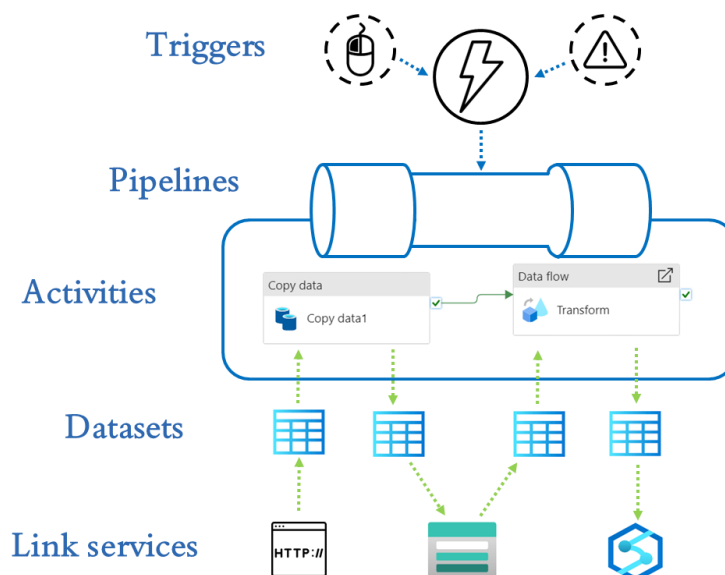
จากการออกแบบ CRS Architecture ทำให้ทราบว่าถ้าจะมีสถาปัตยกรรมของข้อมูลเป็นอย่างไร โดยสามารถระบุอย่างชัดเจนว่าจะดึงข้อมูลจากไหน แปลงข้อมูลให้เป็นรูปแบบใด และนำไปเก็บไว้ที่ใด ขั้นตอนต่อมาจะทำการรวบรวม Transforms Algorithms ในขั้นตอน Prototype และทำการสร้าง ETL pipeline(CRS control framework) ที่ azure data factory โดยจะมีการพัฒนาให้เข้ากับระบบการดำเนินงานหลักของบริษัทการศึกษา และปรับปรุงกระบวนการทำงานอย่างต่อเนื่องเพื่อพัฒนา CRS control framework เช่น การปรับปรุง transforms algorithms เพื่อลดระยะเวลาการทำงานของ CRS control framework

1. Algorithms optimization

หลังจากออกแบบและทำการวัดผลในขั้นตอน Prototype เสร็จสิ้นแล้ว ขั้นตอนต่อมาคือ Algorithms optimization เนื่องมาจากการทำงานที่ซับซ้อนของ Algorithms จึงจำเป็นต้องทำการปรับปรุงเพื่อให้ runtime ออกมาน้อยที่สุดแต่ยังคงรักษาคุณภาพของรายงานไว้ โดยยังคงได้ผลลัพธ์ดั้งเดิม หลังจากวิเคราะห์แล้วว่า Algorithms มีเวลาการทำงานอย่างไร จะทำการปรับแก้ในส่วนที่เป็นเป็นคอขวดหรือเป็นปัญหาหลัก เพื่อให้ runtime ของ Algorithms ออกมาน้อยที่สุด โดยเหตุผลที่ต้องทำการพัฒนาเป็นเพราะ ในการดำเนินงานอาจมีบางช่วงเวลา que server ของบริษัทมีการทำงานหนัก ถ้าออกแบบ CRS control framework ได้ไม่ดีอาจจะส่งผลให้เกิดความล่าช้าของการทำงานได้ โดยจะใช้ Data structure ในการ pre-screen ในเบื้องต้นในการหลีกเลี่ยง object ที่สร้างภาระหนักให้แก่ server และใช้ Big o Notation ในการระบุเวลาการทำงานที่แย่ที่สุดในทุกส่วนเพื่อให้สามารถปรับลด runtime ให้ไม่มากจนเกินไป

2. Create CRS Control Framework

ทำการสร้าง CRS control framework โดยใช้ app services ของ Microsoft ที่มีชื่อว่า azure data factory ซึ่งทำหน้าที่เป็นตัวสร้าง data pipeline ผ่าน activity ต่างๆ โดยการเดินทางของข้อมูลจะเป็นไปตาม CRS Data Architecture และมีกระบวนการทำงานที่สอดคล้องไปกับระบบการทำงานหลักของบริษัทการศึกษา จากรูปที่ 13 แสดงให้เห็นถึงองค์ประกอบของ azure data factory ซึ่งประกอบไปด้วย 1.triggers 2.pipelines 3.activities 4.datasets 5.link services โดยเมื่อทำการ trigger แล้ว pipeline นั้นจะประกอบไปด้วย activity ซึ่งเชื่อมข้อมูลกับ dataset โดย dataset มีที่มาจาก link services ซึ่งสามารถอธิบายดังตัวอย่างต่อไปนี้ ใน pipeline นี้มีการเรียกใช้ activity copy data และ dataflow โดยจะทำการ copy data จาก dataset ที่มีที่มาจาก HTTP เว็บไซต์โดยจะทำการ copy data ไปวางที่ storage account เมื่อเสร็จ copy data activity จะไปยัง dataflow activity โดยดึงข้อมูลจาก storage account แล้วทำการ transform และจัดเก็บที่ synapse analytics โดย CRS control framework จะมีการทำงานในลักษณะดังกล่าวแต่จะมี activity, dataset และ link services ที่ต่างออกไป



รูปที่ 13: องค์ประกอบของ azure data factory

3.3 System Integration Test

ในขั้นตอนของ SIT นั้นจะทำการทดสอบ algorithm ด้วยข้อมูลทดสอบที่แตกต่างกันเพื่อเป็นการตรวจทานข้อมูลอีกชั้นหนึ่งก่อนจะไปเฟสต่อไป โดยจะเป็นการเน้นความถูกต้องของข้อมูลเป็นหลักเช่น AccountNumber ต้องเป็นลักษณะข้อมูลของตัวเลข ดังนั้น ข้อมูลชนิดอื่นจะต้องไม่สามารถอยู่ในองค์ประกอบนี้ได้ เป็นต้นและข้อมูลที่บันทึกผลงานต้องตรงกับแหล่งที่มาเพื่อป้องกันการปลอมแปลงข้อมูลระหว่างทางเช่นที่ต้นทางข้อมูลของเงินในบัญชีเป็น 600.00 แต่ที่ข้อมูลที่บันทึกในรายงานเป็น 550.00 ถ้าเป็นลักษณะแบบนี้จะต้องไม่สามารถสร้างรายงาน CRS ออกมาจาก CRS control framework ได้ สุดท้ายทำการ Validate structure ของไฟล์รายงาน(XML file) ด้วย XML Schema เพื่อตรวจสอบว่า ชื่อขององค์ประกอบ โครงสร้างลำดับชั้น และลักษณะข้อมูลถูกต้องเป็นไปตามที่ทาง OECD ได้กำหนดไว้

3.4 User Acceptance Test

ในขั้นตอนของ UAT จะทำการตรวจสอบว่าระบบสามารถสร้างไฟล์รายงานได้จริงและมีการจัดเก็บข้อมูลตามระบบการทำงานขององค์กร โดยต้องมีการเรียกใช้งานระบบที่ง่ายและสะดวกต่อการใช้งานสำหรับผู้ใช้งาน

บทที่ 4

ผลการดำเนินงานวิจัย

จากการดำเนินงานตามวิธีการดำเนินงานซึ่งประกอบไปด้วย 1.Develop 2.System Integration Test 3. User Acceptance Test ได้ผลลัพธ์ของการดำเนินงาน ดังนี้

4.1 Result of Develop

4.1.1 transform algorithm

จากผลการดำเนินงานในขั้น business requirement analysis, prototype และ algorithm optimization ผลลัพธ์คือ algorithm สามารถสร้างไฟล์รายงานได้ตรงตามที่ OECD และมีเวลาการทำงานที่รวดเร็ว โดยรูปที่ 14 แสดงถึงตัวอย่างของไฟล์รายงาน และรูปที่ 15 แสดงถึงเวลาการทำงานของ algorithm ที่ผ่านการ optimized แล้วโดยมีเวลาการทำงานที่เร็วกว่าขั้น prototype 7 เท่า

```
<?xml version="2.0" encoding="UTF-8" ?>
<crs:CRS_OECD xmlns="urn:oecd:ties:crs:v2" xmlns:crs="urn:oecd:ties:crs:v2" xmlns:cfc="urn:oecd:ties:commonatypesfatcacrsv2" xmlns:ftc="urn:oecd:ties:fatca:v1"
xmlns:stf="urn:oecd:ties:crs:stf:v5" xmlns:iso="urn:oecd:ties:isocrstypes:v1" xmlns:xsd="http://www.w3.org/2001/XMLSchema" version="2.0">
  <crs:MessageSpec>
    <crs:SendingCompanyIN>107536000000</crs:SendingCompanyIN>
    <crs:TransmittingCountry>TH</crs:TransmittingCountry>
    <crs:ReceivingCountry>TH</crs:ReceivingCountry>
    <crs:MessageType>CRS</crs:MessageType>
    <crs:Contact>puttachard.soon@kkpfg.com</crs:Contact>
    <crs:MessageRefId>107536000000.2023.TH.0001</crs:MessageRefId>
    <crs:MessageTypeIndic>CRS703</crs:MessageTypeIndic>
    <crs:ReportingPeriod>2023-12-31</crs:ReportingPeriod>
    <crs:Timestamp>2023-07-10T19:33:41</crs:Timestamp>
  </crs:MessageSpec>
  <crs:CrsBody>
    <crs:ReportingFI>
      <crs:ResCountryCode>TH</crs:ResCountryCode>
      <crs:IN_INTYPE="TIN" issuedBy="MY":107536000000</crs:IN>
      <crs:Name>Kiatnakin Phatra Bank Plc.</crs:Name>
      <crs:Address legalAddressTypes="OECD302">
        <cfc:CountryCode>TH</cfc:CountryCode>
        <cfc:AddressFix>
          <cfc:Street>SUKHUMVIT 21 (ASOKE)</cfc:Street>
          <cfc:BuildingIdentifier>KKP TOWER A</cfc:BuildingIdentifier>
          <cfc:SuiteIdentifier>209</cfc:SuiteIdentifier>
          <cfc:FloorIdentifier>17TH</cfc:FloorIdentifier>
          <cfc:DistrictName>KHLONG TOEY NUA</cfc:DistrictName>
          <cfc:PostCode>10110</cfc:PostCode>
          <cfc:City>WATTANA</cfc:City>
          <cfc:CountrySubentity>Bangkok</cfc:CountrySubentity>
        </cfc:AddressFix>
      </crs:Address>
    </crs:ReportingFI>
  </crs:CrsBody>
  <crs:DocSpec>
    <stf:DocTypeIndic>OECD11</stf:DocTypeIndic>
    <stf:DocRefId>107536000000.2023.TH.0001.DR0001</stf:DocRefId>
  </crs:DocSpec>
</crs:CRS_OECD>
```

รูปที่ 14: CRS report example

```

320     file.write(ET.tostring(xml_doc, encoding="unicode")) # .decode('utf-8') ET.to
321
Command took 1.08 minutes -- by nattapat.dung@kkpfg.com at 11/5/2023, 8:32:15 PM on dbc-kkps-test-edp

```

```

75
▶ (50) Spark Jobs
Command took 9.19 seconds -- by nattapat.dung@kkpfg.com at 11/5/2023, 8:32:17 PM on dbc-kkps-test-edp

```

รูปที่ 15: time consume for created report

4.1.2 CRS Architecture

จากผลการดำเนินงานในขั้น design architecture ผลลัพธ์คือ ได้ CRS Architecture ที่ประกอบไปด้วย 1.Architecture Vision ที่ช่วยระบุวิสัยทัศน์และกำหนดทิศทางการพัฒนาโครงการที่จะดำเนินไป 2.Business architecture ที่ช่วยระบุผลประโยชน์ทางธุรกิจของโครงการที่มีประโยชน์ต่อบริษัทกรณีศึกษาและระบุขีดความสามารถทางธุรกิจของโครงการ 3.Information architecture ที่ช่วยระบุ data architecture ที่บ่งบอกลักษณะการเคลื่อนไหวของข้อมูลของโครงการและ application architecture ที่บ่งบอกวิธีการใช้งานแก่ผู้ใช้งาน (พนักงานแผนก EDS) (สามารถดูรายละเอียดเพิ่มเติมได้ที่ภาคผนวก ก CRS Architecture)

4.1.3 CRS Data pipeline

จากผลการดำเนินงานในขั้น create CRS control framework ผลลัพธ์คือ ได้ CRS data pipeline ที่เป็นระบบอัตโนมัติบน azure data factory โดยมีการทำงานที่สอดคล้องกับระบบการดำเนินงานหลักของบริษัทซึ่งช่วยแบ่งเบาภาระในการดำเนินงานของพนักงานเมื่อถึงกำหนดส่งรายงานประจำปีตามข้อบังคับจากกรมสรรพากร (สามารถดูรายละเอียดเพิ่มเติมได้ที่ภาคผนวก ก CRS control framework & configuration design)

4.2 Result of System Integration Test

4.2.1 Validation of algorithm with test data

จากผลการดำเนินงาน ผลลัพธ์คือ transform algorithm สามารถแสดงผลได้ถูกต้องต่อข้อมูลทดสอบที่จัดทำขึ้นเพื่อตรวจสอบการทำงานของ algorithm ตัวอย่างเช่น ข้อมูลทดสอบข้อมูลหนึ่งจะทำการทดสอบว่า algorithm สามารถสร้างองค์ประกอบย่อยภายในองค์ประกอบ Address ได้อย่างถูกต้องหรือไม่ โดยในองค์ประกอบของ Address นั้นมีทางเลือกการบันทึกด้วยกัน 2 แบบได้แก่ 1.AddressFree 2.AddressFix โดยมีลักษณะข้อมูลดังนี้

1.<AddressFree>Mandatory Information</AddressFree>

2.<AddressFix>

<Street>Option Information</Street>

<BuildingIdentifier>Option Information</BuildingIdentifier>

<SuiteIdentifier>Option Information</SuiteIdentifier>

<FloorIdentifier>Option Information</FloorIdentifier>

<DistrictName>Option Information</DistrictName>

<PostCode>Option Information</PostCode>

<City>Mandatory Information</City>

<CountrySubentity>Option Information</CountrySubentity>

</AddressFix>

โดยปกติแล้ว algorithm จะทำการสร้างองค์ประกอบ AddressFix ก่อนเสมอ ทว่าหากมีกรณีที่องค์ประกอบย่อยที่ชื่อว่า City ไม่มีข้อมูลอยู่ ตามข้อบังคับของรายงาน รายงานจะต้องระบุเป็น AddressFree แทน AddressFix โดยหาก algorithm ไม่สามารถเปลี่ยนแปลงการดำเนินการในเงื่อนไขเหล่านี้ได้ จะทำให้เกิด error ขึ้นในขั้นตอนของการตรวจสอบโครงสร้างและลักษณะของข้อมูลต่อไป

4.2.2 Report Validation By XML Schema

จากผลการดำเนินงาน ผลลัพธ์คือ algorithm สามารถตรวจสอบโครงสร้างและลักษณะของข้อมูลได้ถูกต้อง โดยมี Invalid notification แจ้งเตือนเมื่อเกิดปัญหาขึ้นจากการตรวจสอบโครงสร้างและลักษณะของข้อมูล ตัวอย่างเช่น บรรทัดที่ 33 บอกว่า DocRefID นั้นไม่ได้ถูกระบุด้วย xml schema แต่เป็น DocRefId หรือ บรรทัดที่ 58 บอกว่า Controllingperson ควรจะมี Individual เป็นองค์ประกอบย่อยแต่ทว่าในรายงานนั้นไม่มี Individual ประกอบอยู่ในรายงาน เป็นต้น

```
Validation failed against /dbfs/mt/uat/working/TMP/JSON_CONFIG/INPUT/CommonTypesFatcaCrs_v2.0.xsd.
Validation error message:
Line 1:Element '{urn:oecd:ties:crs:v2}CRS_OECD': No matching global declaration available for the validation root.
Validation failed against /dbfs/mt/uat/working/TMP/JSON_CONFIG/INPUT/CrsXML_v2.0.xsd.
Validation error message:
Line 20:Element '{urn:oecd:ties:crs:v2}AddressFix': This element is not expected. Expected is one of { (urn:oecd:ties:commontypesfatcacrs:v2)AddressFree, (urn:oecd:ties:commontypesfatcacrs:v2)AddressFix }.
Line 33:Element '{urn:oecd:ties:crs:v2}DocRefID': This element is not expected. Expected is { (urn:oecd:ties:crs:v2)DocRefId }.
Line 40:Element '{urn:oecd:ties:crs:v2}DocRefID': This element is not expected. Expected is { (urn:oecd:ties:crs:v2)DocRefId }.
Line 42:Element '{urn:oecd:ties:crs:v2}AccountNumber', attribute 'ClosedAccount': 'FALSE' is not a valid value of the atomic type 'xs:boolean'.
Line 42:Element '{urn:oecd:ties:crs:v2}AccountNumber', attribute 'DormantAccount': 'FALSE' is not a valid value of the atomic type 'xs:boolean'.
Line 42:Element '{urn:oecd:ties:crs:v2}AccountNumber', attribute 'AccountNumberType': The attribute 'AccountNumberType' is not allowed.
Line 50:Element '{urn:oecd:ties:commontypesfatcacrs:v2}AddressFix': Missing child element(s). Expected is one of { (urn:oecd:ties:commontypesfatcacrs:v2)FloorIdentifier, (urn:oecd:ties:commontypesfatcacrs:v2)DistrictName, (urn:oecd:ties:commontypesfatcacrs:v2)PostCode, (urn:oecd:ties:commontypesfatcacrs:v2)City }.
Line 58:Element '{urn:oecd:ties:crs:v2}ControllingPerson': Missing child element(s). Expected is { (urn:oecd:ties:crs:v2)Individual }.
Line 59:Element '{urn:oecd:ties:crs:v2}AccountBalance': The attribute 'currCode' is required but missing.
Line 59:Element '{urn:oecd:ties:crs:v2}AccountBalance': Element content is not allowed, because the content type is a simple type definition.
Line 59:Element '{urn:oecd:ties:crs:v2}AccountBalance': ' ' is not a valid value of the atomic type '{urn:oecd:ties:commontypesfatcacrs:v2}TwoDigitType'.
Line 62:Element '{urn:oecd:ties:crs:v2}Payment': Missing child element(s). Expected is { (urn:oecd:ties:crs:v2)PaymentAmt }.
```

รูปที่ 16: error message จากการ validate ด้วย XML schema

4.3 Result of User Acceptance Test

จากผลการดำเนินงาน ผลลัพธ์คือ ระบบมีการใช้งานที่ง่ายและสะดวกต่อการเรียกใช้ และสามารถจัดเก็บข้อมูลได้ตรงตามระบบการทำงานหลักของบริษัทโดยทำการจัดเก็บข้อมูลใน data tier 1, data tier 2 และ data tier 3 อย่างถูกต้อง โดยที่ data tier 1 หรือ azure data lake storage ทำการจัดเก็บไฟล์ parquet ที่ Staging และ Curated area และทำการจัดเก็บไฟล์รายงาน CRS ที่ Working area อีกทั้งยังทำการจัดเก็บตารางข้อมูลที่ data tier 2 หรือ azure databricks และ data tier 3 หรือ azure synapse analytics ทั้งหมดได้อย่างถูกต้อง

| | BirthDate ▲ | DistrictName ▲ | City ▲ | TransmittingCountry ▲ | ReceivingCountry ▲ | MessageType ▲ |
|---|-------------|----------------|--------------|-----------------------|--------------------|---------------|
| 1 | 1973-12-08 | Thawiwatthana | Bangkok | TH | TH | CRS |
| 2 | 1973-05-18 | Muang | Chiang Mai | TH | TH | CRS |
| 3 | 1990-02-04 | Suan Luang | Bangkok | TH | TH | CRS |
| 4 | 2000-09-22 | Muang | Pathum Thani | TH | TH | CRS |
| 5 | 1988-07-07 | Lat Phrao | Bangkok | TH | TH | CRS |
| 6 | 1979-11-08 | Phasi Charoen | Bangkok | TH | TH | CRS |
| 7 | 2000-10-29 | Muang | Nonthaburi | TH | TH | CRS |

รูปที่ 17: example of sql table in data tier 2

บทที่ 5

สรุปและอภิปรายผล

5.1. สรุปผล

โครงการนี้มีวัตถุประสงค์เพื่อสร้างระบบงาน CRS(CRS Control Framework) ให้กับผู้ใช้งาน(พนักงานแผนก EDS) โดยทำงานได้รวดเร็วและมีความสะดวกในการใช้งาน จากผลการดำเนินงานพบว่า CRS control framework สามารถสร้างไฟล์รายงานได้อย่างถูกต้องตามที่ OECD ได้กำหนดและมีระบบการทำงานที่สอดคล้องกับระบบการทำงานหลักของบริษัททำให้ผู้ใช้งานใช้งานได้สะดวกอีกทั้งยังสามารถทำงานได้รวดเร็ว ซึ่งช่วยแบ่งเบาภาระในการดำเนินงานของพนักงานเมื่อถึงกำหนดการส่งรายงานประจำปี

5.2. อภิปรายผล

จากผลการดำเนินงานเพื่อพัฒนาระบบ CRS สำหรับกระบวนการออกแบบจนถึงเริ่มทดลองทดสอบ ประกอบไปด้วยหลายขั้นตอนโดยต้องอาศัยความร่วมมือกับพนักงานบริษัทรวมถึงผู้ทำการพัฒนา โดยในการดำเนินงาน ผู้จัดทำได้ใช้เทคโนโลยีของ Microsoft ในการดำเนินงานซึ่งมีบริการครอบคลุมการทำงานทั้ง การวิเคราะห์ การสื่อสาร และ technical ต่างๆครบวงจรทำให้ระบบการทำงานเชื่อมถึงกันเป็นผลให้การดำเนินงานออกมาโดยไม่มีข้อผิดพลาด อีกทั้งยังสามารถนำไปพัฒนาต่อยอดสู่ Production ได้ในอนาคต

5.3 ข้อเสนอแนะและข้อจำกัด

เนื่องจากโครงการนี้ได้จัดทำอยู่ในระยะ UAT และยังไม่ได้มีการใช้งานจริงเกิดขึ้นทางผู้จัดทำจึงได้มีข้อเสนอแนะเพิ่มเติมดังต่อไปนี้

5.3.1 ทดสอบกับบุคคลที่เป็นผู้ใช้งานในจำนวนที่มากขึ้น เนื่องจากรายงานนั้นมีระยะการส่งเป็นแบบรายปี ส่งผลให้การวัดและประเมินผลการใช้งานเป็นไปได้ยาก ทำได้เพียงแค่มุ่งเน้นที่ความถูกต้องของรายงานเท่านั้น

5.3.2 เนื่องจากแผนการพัฒนา ณ วันที่ทำการพัฒนานั้นถูกเลื่อนออกไปเป็นผลให้การนำเข้าข้อมูลยังคงเป็นระบบแบบแมนนวล ทว่าหากสามารถทำเป็นอัตโนมัติได้ จะช่วยให้การใช้งานของผู้ใช้งานนั้นง่ายขึ้นและสะดวกสบายยิ่งขึ้น

5.3.3 สืบเนื่องจากกระบวนการดำเนินงานหลักของบริษัทกรณีศึกษาใน data tier 3(data mart) จะอนุญาตให้ผู้ใช้งานเข้ามาใช้งานข้อมูลเพื่อไปทำการวิเคราะห์หรือสร้าง dashboard ได้ ทางผู้จัดทำจึงได้มีข้อเสนอแนะเพิ่มเติมโดยได้ทำการสร้าง dynamic dashboard ซึ่งอาจจะเป็นประโยชน์ต่อผู้ใช้งานในการลดระยะเวลาในการออกแบบ dashboard ลง (สามารถดูรายละเอียดเพิ่มเติมได้ที่ภาคผนวก ก Power BI Dashboard)

บรรณานุกรม

กรมสรรพกร. (2566, กรกฎาคม). *หลักการรายงานข้อมูล-crs*. www.rd.go.th. Retrieved July 5, 2023, from <https://www.rd.go.th/64547/%E0%B8%AB%E0%B8%A5%E0%B8%B1%E0%B8%81%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%A3%E0%B8%B2%E0%B8%A2%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B8%82%E0%B9%89%E0%B8%AD%E0%B8%A1%E0%B8%B9%E0%B8%A5-crs.html>

โกเมศ อัมพวัน. (2566). *Datawarehouse design*. staff.informatics.buu.ac.th. Retrieved July 5, 2023, from <https://staff.informatics.buu.ac.th/~komate/886452/data%20warehouse-ch2.pdf>

Lung Yiu, & Vista. (2008). *Extract, Transform, Load (ETL)*. people.cs.aau.dk. Retrieved July 5, 2023, from https://people.cs.aau.dk/~tdn/itev/uploads/media/DW4_ETL.pdf

Massachusetts Institute of Technology. (2022). *The Big O notation*. web.mit.edu. Retrieved July 11, 2023, from https://web.mit.edu/16.070/www/lecture/big_o.pdf

Organization for Economic Co-operation and Development . (2019, June). *Common Reporting Standard (CRS)*. www.oecd.org. Retrieved June 19, 2023, from <https://www.oecd.org/tax/exchange-of-tax-information/common-reporting-standard-xml-schema-user-guide-for-tax-administrations-june-2019.pdf>

Song, Evans, & Park. (1995). *A Comparative Analysis of Entity-Relationship Diagrams*. www.researchgate.net. Retrieved July 21, 2023, from https://www.researchgate.net/publication/243781001_A_Comparative_Analysis_of_Entity-Relationship_Diagrams

The Open Group Standard. (2018). *The TOGAF® Standard, Version 9.2*. Firebrand.Training. Retrieved July 25, 2023, from <https://firebrand.training/au/pdf/learn/open-group/togaf-standard.pdf>

Visual Paradigm. (2023). *What is Zachman Framework?* www.visual-paradigm.com. Retrieved August 21, 2023, from <https://www.visual-paradigm.com/guide/enterprise-architecture/what-is-zachman-framework/>

Wes McKinney and the Pandas Development Team. (2022). *pandas: powerful Python data analysis*. pandas.pydata.org. Retrieved July 11, 2023, from <https://pandas.pydata.org/pandas-docs/version/1.4.4/pandas.pdf>

Zachman. (2003). *The Zachman framework for enterprise architecture*. www.dragon1.com. Retrieved July 25, 2023, from <https://www.dragon1.com/downloads/ZachmanBookRfExtract.pdf>

ภาคผนวก ก

CRS Architecture

Architecture ที่ให้สอดคล้องกับ CRS Project จะประกอบไปด้วย 1.CRS Architecture Vision 2.CRS Business Architecture 3. CRS Information System Architecture โดยแต่ละ Architecture มีจุดมุ่งหมายเพื่อบรรลุจุดประสงค์ของตัวเอง ดังต่อไปนี้

1. CRS Architecture Vision

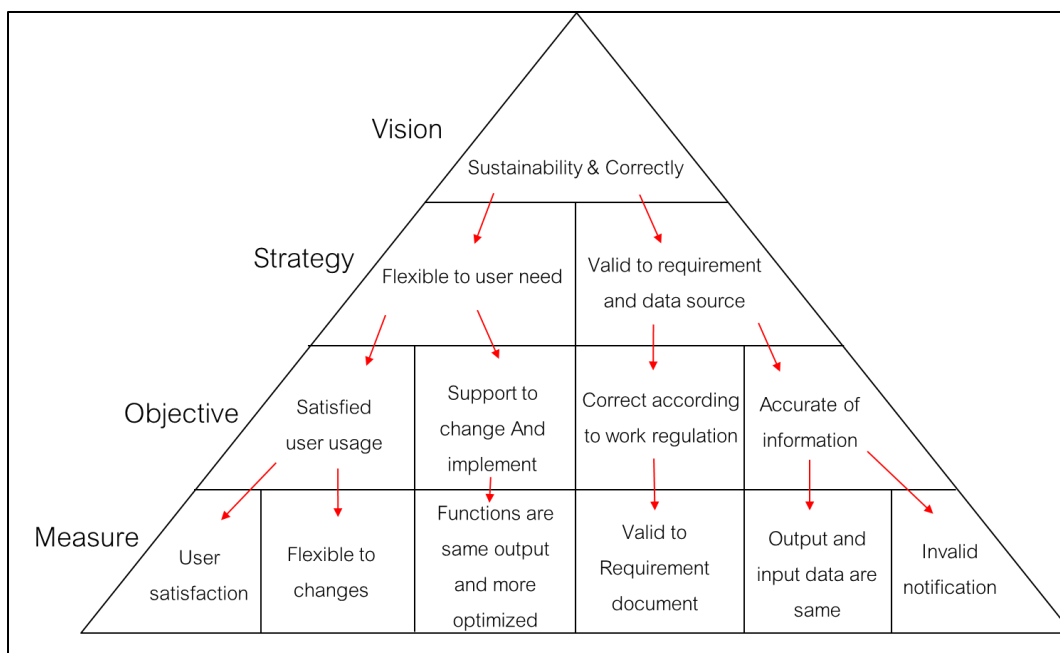
ในเฟสนี้มีจุดประสงค์คือ เพื่อระบุ business value และ พัฒनावิสัยทัศน์ของความสามารถของโปรเจกต์นี้

1.1. Business Value

เนื่องด้วยรายงาน CRS เป็นรายงานที่เกี่ยวข้องทางภาษีและกฎหมาย ทำให้มีหลากหลายฝ่ายเข้ามาเกี่ยวข้องตัวอย่างเช่น Business Analyst, Branch Operation & Planning, Anti-Money Laundering Compliance เป็นต้น ที่จะเข้ามามีส่วนร่วมในการทำงานร่วมกันโดยทุกฝ่ายที่นอกเหนือจากฝ่าย Enterprise data services(EDS) จะเป็นคนรวบรวมข้อมูลและ mapping condition สำหรับรายงาน CRS ให้กับฝ่าย EDS โดยฝ่าย EDS จะทำหน้าที่ในการผลิตรายงาน CRS ให้ตรงกับเงื่อนไขของ requirement document โดยจะมีค่าปรับเกิดขึ้นหากไฟล์รายงานผิดเมื่อนำส่งสรรพากรและจะเกิดปัญหาที่กระทบกับหลายฝ่ายเป็นวงกว้าง ดังนั้นจุดประสงค์หลักของโครงการนี้จึงเป็นการพยายามหลีกเลี่ยงค่าปรับโดย **Business Value คือ ความถูกต้องและครบถ้วนของข้อมูลของไฟล์รายงาน CRS ตามหลักที่ OECD ได้กำหนด**

1.2. Architecture Vision

จากรูปที่ 18 แสดงให้เห็นถึงวิสัยทัศน์(vision)ของโครงการนี้โดยมีกลยุทธ์(strategy) ที่จะกำหนดแนวทางที่เป็นรูปธรรมเพื่อให้บรรลุวิสัยทัศน์(vision) ที่กำหนด โดยระบุ จุดประสงค์(objectives) ที่ต้องทำในแต่ละกลยุทธ์(strategy) และวัดผลว่าบรรลุจุดประสงค์(objectives) ด้วยตัวชี้วัด(measure) ที่กำหนด โดยการจัดทำ Architecture Vision มีจุดมุ่งหมายคือ ทำให้ทุกการพัฒนาใดๆต้องสอดคล้องกับ Architecture Vision



รูปที่ 18: Architecture vision model

2. CRS Business Architecture

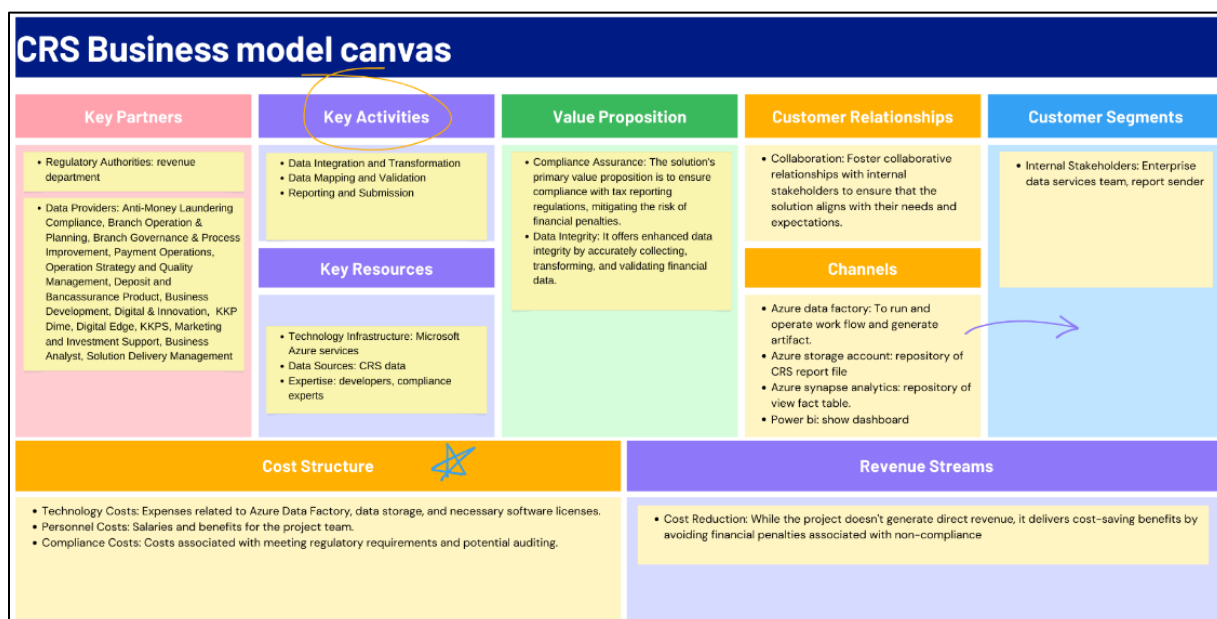
ในเฟสนี้มีจุดประสงค์คือ 1.ระบุ baseline business Description 2.กำหนด target business Description และ 3.สร้าง gap analysis ระหว่าง baseline and target business Description

2.1. Baseline Business Description

ในเฟสนี้จะเป็นการระบุถึงสถานการณ์ทางธุรกิจในปัจจุบันของโครงการ โดยอธิบายถึงภาพรวมและขีดความสามารถทางธุรกิจของโครงการ

2.1.1 Business Model Canvas

จากรูปที่ 19 แสดงถึง business model canvas ของ CRS โดยมีจุดประสงค์เพื่อระบุถึงองค์ประกอบทางธุรกิจโดยรวมของโครงการ ได้แก่ 1.คุณค่าของโครงการที่มีผลต่อธุรกิจของบริษัทการศึกษา 2.บุคคลที่จะเข้ามาใช้งาน services 3.วิธีการสร้างความสัมพันธ์กับกลุ่มผู้ใช้งาน 4.ช่องทางการใช้งาน service 5.สิ่งที่ต้องทำเพื่อขับเคลื่อนโครงการ 6.พาหนะหลักที่ช่วยให้โครงการนี้ประสบความสำเร็จ 7.ทรัพยากรที่ใช้ในการขับเคลื่อนโครงการ 8.ค่าใช้จ่ายในการขับเคลื่อนโครงการ 9.ประโยชน์ที่โครงการนี้มีต่อบริษัท



รูปที่ 19: CRS business model canvas

2.1.2 Business Capability Map

จากรูปที่ 20 แสดงถึงขีดความสามารถ ณ ตอนเริ่มโครงการซึ่งมีจุดประสงค์เพื่อประมาณขีดความสามารถในขั้นต้นและใช้เป็นข้อมูลสำหรับการวางแผนปรับปรุงแก้ไขให้ดียิ่งขึ้นโดยสามารถอธิบายได้ดังนี้

1.Product โดย product จะหมายถึง data pipeline บน azure data factory

1.1.Product design ให้ความหมายว่าที่การออกแบบที่ baseline เมื่อเปรียบเทียบกับ target ยังมีความห่างไกลจาก target ในระดับกลางแต่ไม่กระทบต่อการดำเนินงานทางธุรกิจ หมายความว่าสามารถ run data pipeline ตัวนี้ได้แต่ยังคงไม่มีความสมบูรณ์ในเรื่องขององค์ประกอบภายในแต่ก็ไม่ได้กระทบหรือมีปัญหาหากเรียกใช้งานซึ่งความไม่สมบูรณ์นี้(medium gap) มีความเสี่ยงในระดับกลางหากต่อการดำเนินงานทางธุรกิจแต่ยังอยู่ในระดับที่สามารถจัดการได้ โดยการออกแบบ product มีความสำคัญต่อการดำเนินธุรกิจมาก แต่ในขั้นต้นนี้ยังคงมีศักยภาพต่ำอยู่

1.2.Road map generation ให้ความหมายว่าที่การออกแบบที่ baseline เมื่อเปรียบเทียบกับ target ยังมีความห่างไกลจาก target ในระดับกลางแต่ไม่กระทบต่อการดำเนินงานทางธุรกิจ หมายความว่าเมื่อเปรียบเทียบกับ target แล้วตัว Road map generation ยังมีความสามารถที่ยังห่างจากเป้าหมายที่ควรเป็นในระดับกลาง มีความเสี่ยงในระดับกลางหากออกแบบได้ไม่ดี ส่งผลต่อการดำเนินธุรกิจต่ำและในขั้นต้นนี้ยังคงมีศักยภาพต่ำ

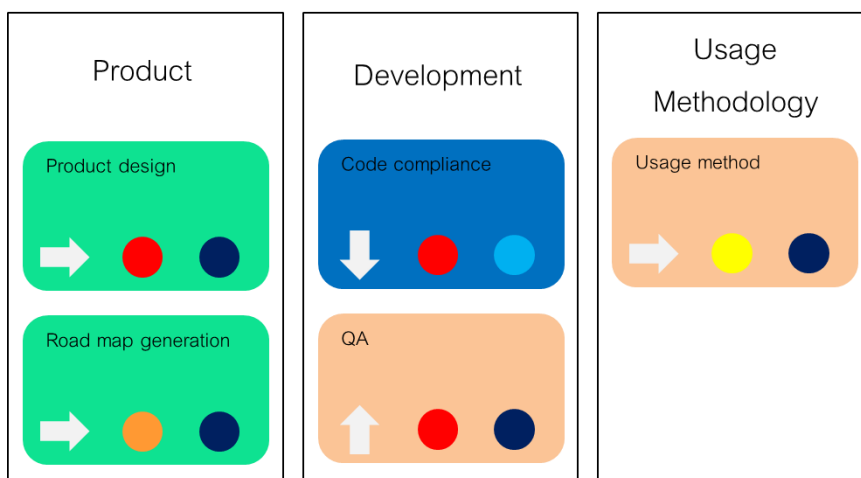
2.Development โดย development จะหมายถึงการพัฒนา product

2.1.Code compliance ให้ความหมายว่า การปฏิบัติตามมาตรฐานที่กำหนดไว้สำหรับการเขียนโปรแกรมเป็นไปตามข้อกำหนดหรือไม่ ตัวอย่างเช่น 1.standard language 2.code style โดยที่ baseline เมื่อเปรียบเทียบกับ target มีความห่างน้อย หมายความว่า code compliance ค่อนข้างใกล้เคียงกับเป้าหมาย ซึ่งจะมีความเสี่ยงต่ำเพราะหากผิดพลาดจะสามารถแก้ไขได้ทันที โดยส่งผลกระทบต่อการดำเนินธุรกิจสูงและในขั้นต้นมีศักยภาพในระดับกลาง

2.2.QA ให้ความหมายว่ากระบวนการตรวจสอบคุณภาพที่ baseline เมื่อเปรียบเทียบกับ target ยังคงมีความห่างอยู่มาก หมายความว่า การตรวจสอบคุณภาพ ในขั้นต้นยังห่างไกลจากเป้าหมายอยู่มาก โดยจะมีความเสี่ยงสูงหากไม่ได้รับการปรับปรุงที่ดีพอเพราะ output ที่ออกมาอาจจะผิดพลาดได้ โดยส่งผลกระทบต่อการดำเนินธุรกิจสูงและในขั้นต้นมีศักยภาพในระดับต่ำ

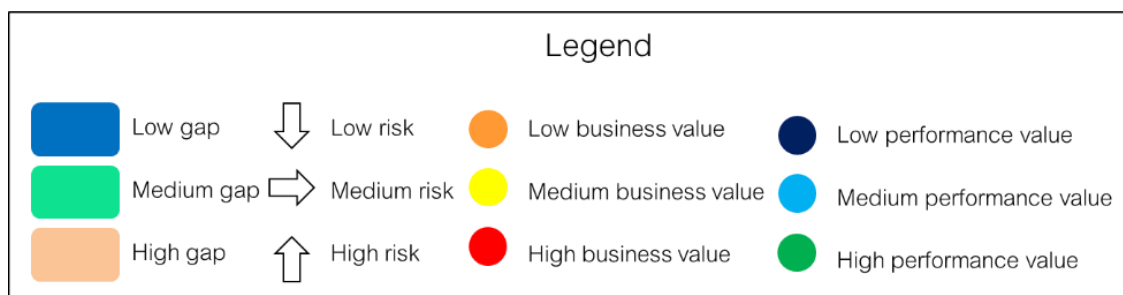
3.Usage methodology โดย usage methodology จะหมายถึงความสะดวกในการใช้งานในมุมมองของผู้ใช้งาน

3.1.Usage method ให้ความหมายว่าความสะดวกในการใช้งานที่ baseline เมื่อเปรียบเทียบกับ target มีความห่างมาก หมายความว่าความสะดวกในการใช้งานในขั้นต้นยังคงห่างไกลจากเป้าหมายอยู่มาก โดยจะมีความเสี่ยงในระดับกลางและต้องได้รับการปรับปรุงแก้ไขเพราะต้องออกแบบให้ผู้ใช้งานใช้งานได้ง่ายตามวิสัยทัศน์ที่ระบุไว้ โดยส่งผลกระทบต่อการดำเนินธุรกิจในระดับกลางและในขั้นต้นมีศักยภาพในระดับต่ำ



รูปที่ 20: business capability model

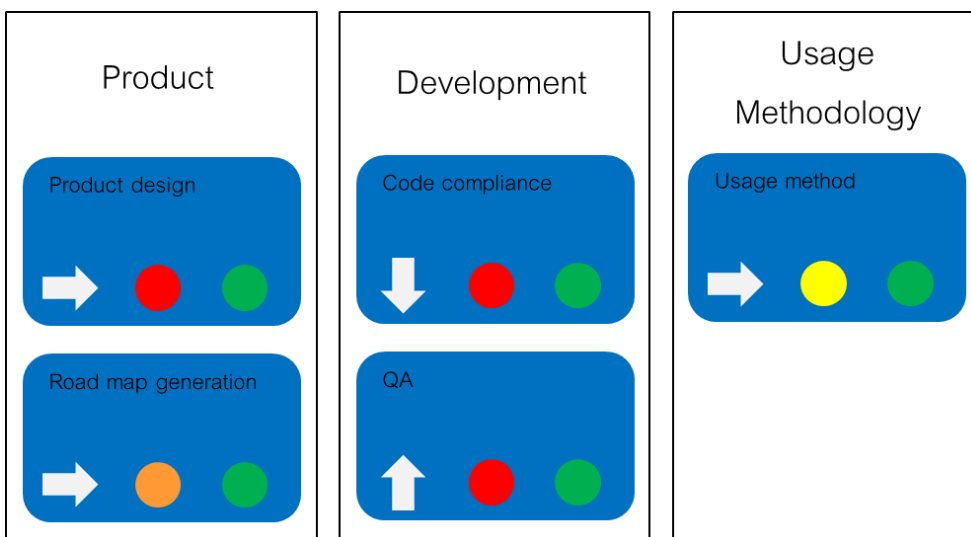
จากรูปที่ 21: แสดงถึงการบ่งบอกระดับของ ความห่าง(gap), ความเสี่ยง(risk), มูลค่าที่กระทบต่อธุรกิจ(business vale) และ ศักยภาพ(performance value) ตามสีและสัญลักษณ์ที่กำหนด



รูปที่ 21: legend categories of business capability model

2.2. Target Business Description

จากการระบุขีดความสามารถของโครงการในขั้นต้น ในขั้นตอนนี้จะเป็นการกำหนดเป้าหมายในทางธุรกิจที่โครงการนี้จะดำเนินไป จากรูปที่ 20 ทำให้ทราบได้ว่า product design, road map generation, QA และ usage methodology ต้องมีการลด gab ระหว่าง baseline และ target และเพิ่มศักยภาพให้สูงขึ้นในทุกองค์ประกอบ ดังรูปที่ 22 เพื่อให้มีความใกล้เคียงกับ target มากยิ่งขึ้น



รูปที่ 22: target business capability map

2.3. Gap Analysis

จากการดำเนินการระบุ baseline business description และ กำหนด target business description ทำให้ทราบจุดที่ต้องทำการพัฒนาเพื่อให้โครงการมีความสามารถที่บรรลุเป้าหมายที่อยู่ภายใต้วิสัยทัศน์ที่กำหนด โดยสิ่งที่ต้องพัฒนาเพิ่มเติมได้แก่ 1.product design (สามารถดูรายละเอียดเพิ่มเติมได้ที่บทที่ 4 - 4.2. Data pipeline & configuration design) 2.Road map generation (สามารถดูรายละเอียดเพิ่มเติมได้ที่บทที่ 4 - 4.2. Data pipeline & configuration design) 3.code compliance (สามารถดูรายละเอียดได้ที่บทที่ 4 - 4.1.Transform Algorithm) 4.QA (สามารถดูรายละเอียดเพิ่มเติมได้ที่บทที่ 4 - 4.2.Transform Algorithm) 5.usage methodology (สามารถดูรายละเอียดเพิ่มเติมได้ที่ภาคผนวก 3.2 CRS application architecture)

3. CRS Information Architecture

ในเฟสนี้จะประกอบไปด้วย 2 เฟสย่อยได้แก่ CRS data architecture และ CRS application architecture ดังต่อไปนี้

3.1 CRS Data Architecture

ในเฟสนี้มีจุดประสงค์คือ 1.ระบุ baseline data Description 2.กำหนด target data Description และ 3.สร้าง gap analysis ระหว่าง baseline and target data Description

3.1.1 Baseline Data Description

ในเฟสนี้จะเป็นการระบุถึงสถานการณ์การดำเนินงานทางข้อมูลในปัจจุบันซึ่งจะระบุถึงลักษณะของข้อมูลที่น่าสนใจและการไหลของข้อมูล

1. Sample Input Data

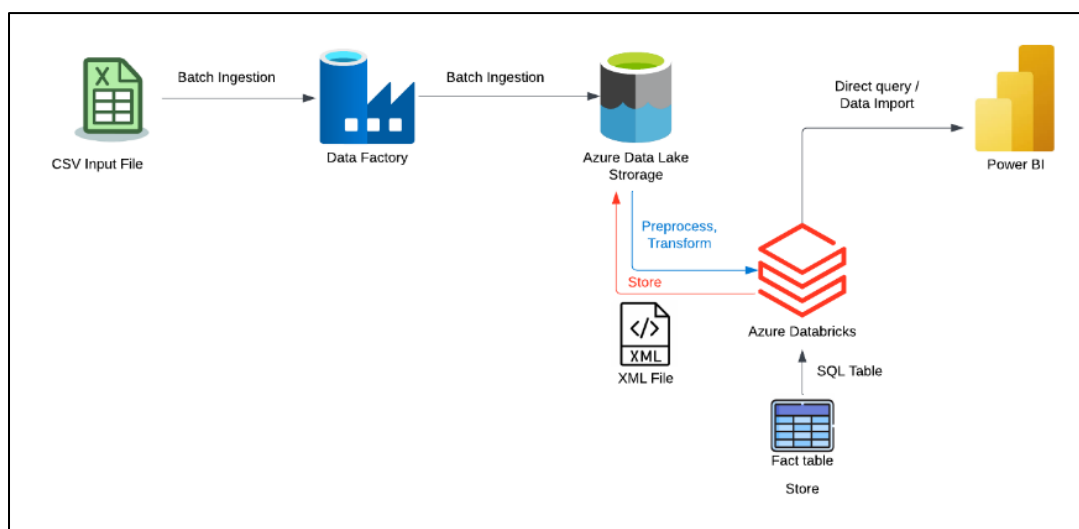
ในขั้นต้นข้อมูลที่ใช้พัฒนาจะเป็นข้อมูลทดสอบที่เกิดจากการนำข้อมูลจริงบางส่วนมาใช้ในการพัฒนาและใช้ทดสอบการทำงานของ algorithm

| crs:SendingCompanyIN | crs:TransmittingCountry | crs:ReceivingCountry | crs:MessageType | crs:Warning | crs:Contact | crs:MessageRefID |
|----------------------|-------------------------|----------------------|-----------------|-------------|-------------|------------------|
| | TH | TH | CRS | | | |
| | TH | TH | CRS | | | |
| | TH | TH | CRS | | | |

รูปที่ 23: sample input data

2. CRS Data Migration

ในขั้นต้นจะทำการ batch ingestion ไฟล์ข้อมูลจากนั้น trigger data pipeline บน azure data factory แล้วทำการเก็บไฟล์รายงาน ที่ Azure storage account และจะมีการสำรองเก็บตารางข้อมูลที่ azure databricks จากนั้นจะถูกนำไปสร้าง dashboard โดยดึงข้อมูลโดยตรงผ่าน azure databricks

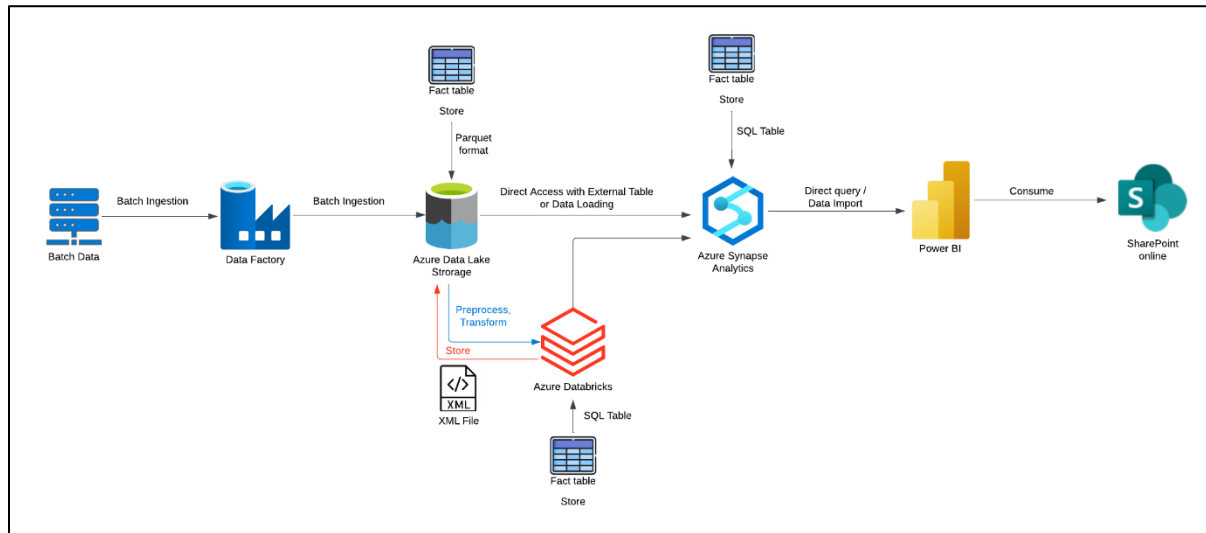


รูปที่ 24: baseline CRS data migration diagram

3.1.2 target data Description

1. Target CRS Data Migration

ลักษณะของ target data migration จะทำการ batch ingestion จาก azure storage account จากนั้นทำการ trigger data pipeline บน azure data factory โดยทำการสำรองเก็บตารางข้อมูลและสร้างไฟล์รายงานจาก processing notebook ที่ azure databricks และทำการจัดเก็บไฟล์รายงานในรูปแบบ XML และตารางข้อมูลในรูปแบบ parquet ที่ azure storage account สุดท้ายทำการ sync ข้อมูลในรูปแบบตารางไปที่ azure synapse analytics และใช้ข้อมูลเพื่อทำ dashboard ที่ power bi จาก azure synapse analytics และทำการ push dashboard ไปที่คลาวด์ต่อไป



รูปที่ 25: target CRS data migration diagram

3.1.3 Gap Analysis

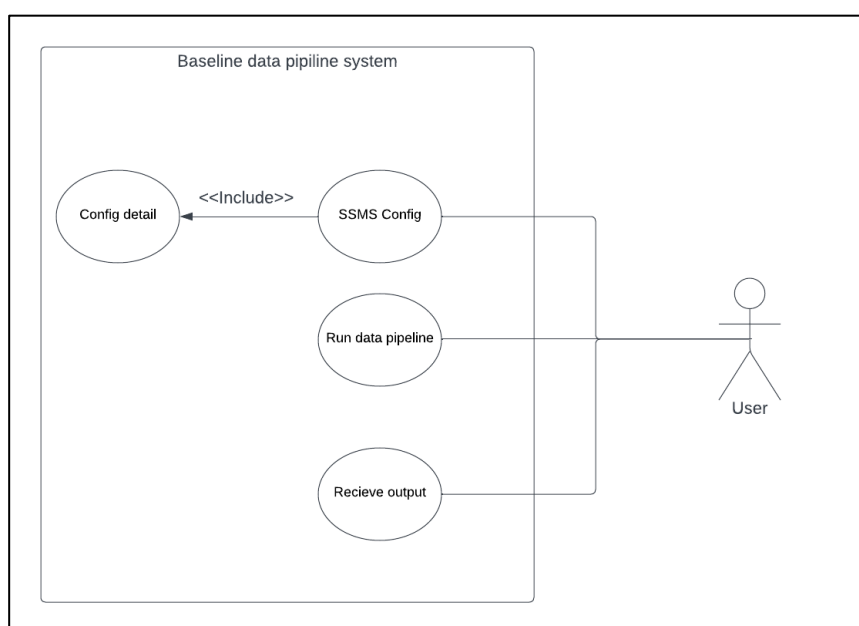
จากการระบุ baseline และกำหนด target data description ทำให้ทราบว่าลักษณะการไหลของข้อมูลไม่เหมาะสม จากรูปที่ 24 จะเห็นได้ว่าสถานที่ที่ใช้เก็บตารางข้อมูลนั้นไม่เหมาะสมและน้อยเกินไปโดยถ้าอ้างอิงตามระบบการเก็บข้อมูลของบริษัทนั้นต้องเก็บข้อมูลตามรูปที่ 25 อีกทั้งการเรียกใช้งานผ่าน azure databricks เพื่อสร้าง dashboard นั้นจะทำให้ user เกิดความสับสนในการใช้งานเพราะสถานที่ที่ใช้ในการเรียกข้อมูลเพื่อทำ dashboard ของบริษัทจะเป็นการเรียกใช้ที่ azure synapse analytics

3.2. CRS Application Architecture

ในเฟสนี้มีจุดประสงค์คือ 1.ระบุ baseline Application Description 2.กำหนด target Application Description และ 3.สร้าง gap analysis ระหว่าง baseline and target Application Description

3.2.1. Baseline Application Description

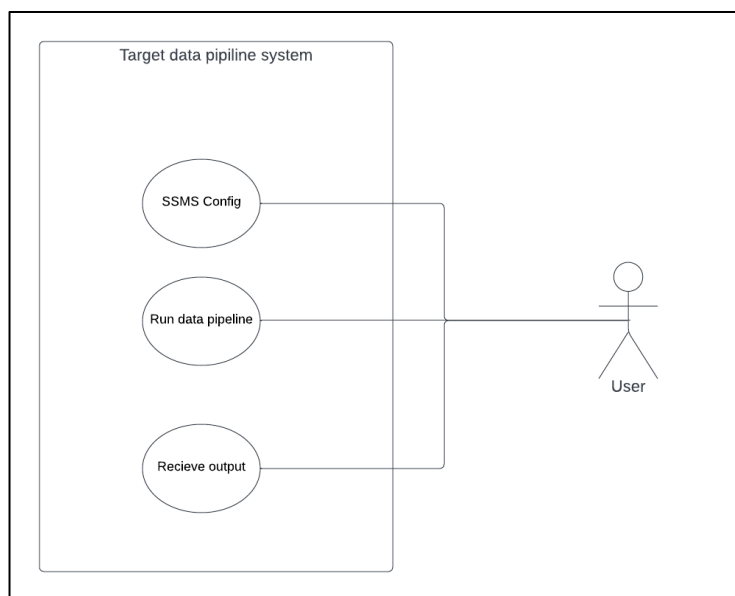
ในการใช้งานผู้ใช้งานจำเป็นต้องทำการกำหนดค่าในตารางกำหนดค่าที่ SQL Server Management Studio(SSMS) ก่อนเพื่อใช้เป็นตัวแปรในการ trigger data pipeline บน azure data factory จากรูปที่ 26 แสดงให้เห็นถึงการใช้งานว่าผู้ใช้งานต้องทำการกำหนดค่าตัวแปรก่อนโดยต้องกำหนดรายละเอียดทั้งหมดของตารางกำหนดค่าและต้องทำการ trigger data pipeline บน azure data factory และรับผลลัพธ์



รูปที่ 26: baseline use-case diagram

3.2.2 Target Application Description

จากรูปที่ 27 แสดง target data pipeline system ที่มุ่งหมายโดยตัวเป้าหมายนี้จะทำการลดขั้นตอนของ config detail ออกไปเปลี่ยนเป็นการกำหนดไว้ล่วงหน้าให้กับผู้ใช้งาน และให้ผู้ใช้งานกำหนดตัวแปรบางส่วนที่สำคัญได้แก่ ปีที่ทำการรายงานและข้อมูลที่ผ่านมาการ mapping ในปีนั้นๆที่จะเปลี่ยนแปลงไปในแต่ละปี



รูปที่ 27: target use-case diagram

3.2.3 Gap Analysis

จากการระบุ baseline และกำหนด target application description ทำให้ทราบว่าลักษณะของการใช้งานของในชั้น baseline นั้นจะเป็นการกำหนดตัวแปรทั้งหมดสำหรับการtrigger data pipeline แต่ในส่วนของ target จะทำการกำหนดไว้ล่วงหน้าและทำการเปลี่ยนเฉพาะข้อมูลที่ต้องเปลี่ยนในแต่ละรอบการรายงาน(yearly) ทำให้ผู้ใช้งานใช้งานได้สะดวกมากยิ่งขึ้น

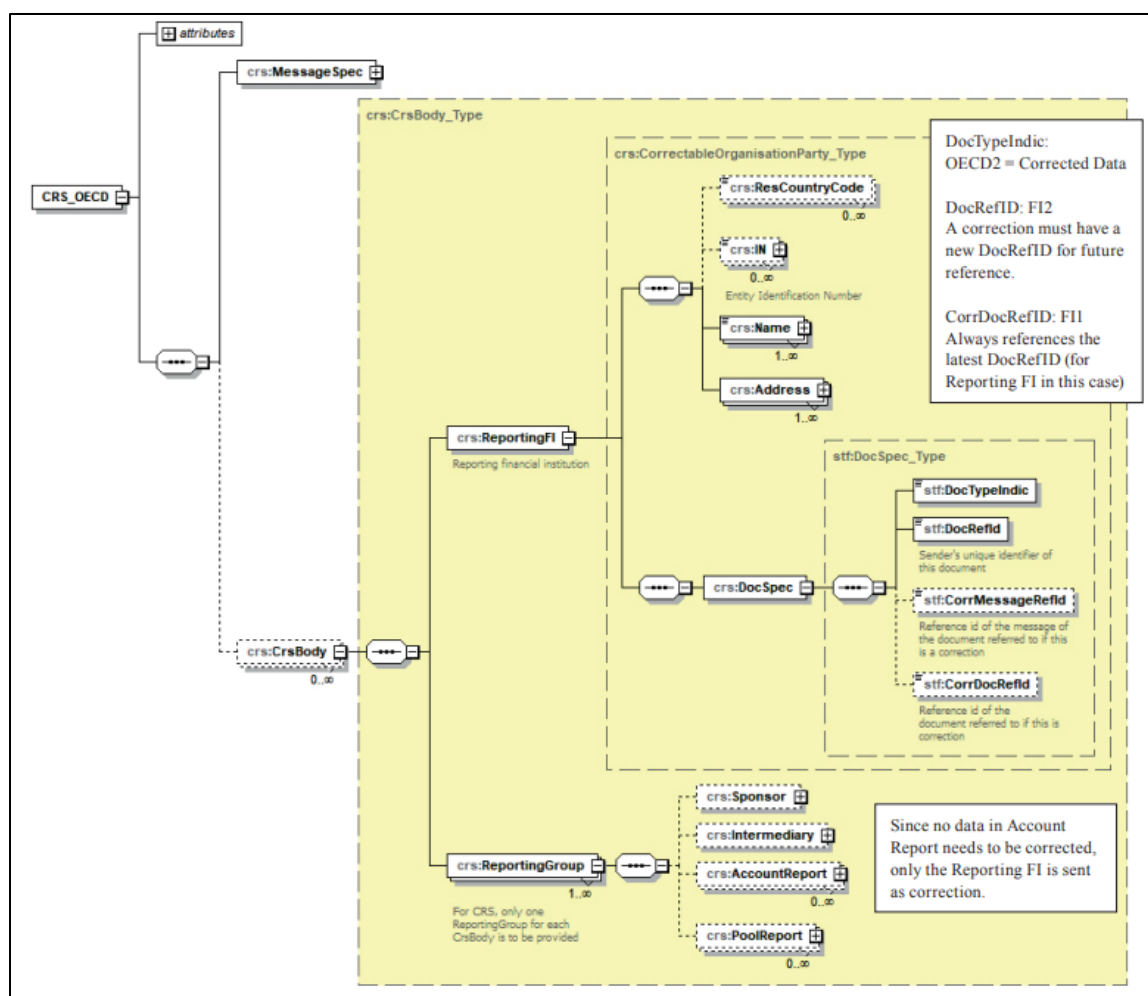
Transform algorithm

การออกแบบ transform algorithm นั้นประกอบไปด้วย 2 ส่วนได้แก่ 1.business requirements analysis โดยมีจุดประสงค์เพื่อวิเคราะห์โครงสร้างและองค์ประกอบที่จำเป็นสำหรับรายงาน CRS 2.transform algorithm design โดยมีจุดประสงค์เพื่อสร้าง algorithm ที่มีประสิทธิภาพในการทำงานโดยสามารถทำงานได้รวดเร็วและถูกต้อง ดังนี้

1. Business requirements analysis

จากการศึกษาคู่มือการออกแบบรายงาน CRS ทำให้ทราบรายละเอียดองค์ประกอบของโครงสร้างข้อมูล และลักษณะของข้อมูลดังต่อไปนี้ โครงสร้างโดยรวมของรายงาน CRS จะประกอบไปด้วย องค์ประกอบหลัก (Parent) และ องค์ประกอบย่อย(Child) โดยองค์ประกอบหลักและ องค์ประกอบย่อยสามารถมี “องค์ประกอบย่อย” เพิ่มเติมจากโครงสร้างเดิม ตัวอย่างเช่นโครงสร้าง CRS_OECD มี องค์ประกอบย่อยเป็น Crsbody และ Crsbody มีองค์ประกอบย่อยเป็น ReportingGroup และ ReportingGroup มีองค์ประกอบย่อยเป็น

AccountReport และ AccountReport สามารถมีองค์ประกอบย่อยได้เพิ่มเติม โดยองค์ประกอบย่อยใดที่สามารถขยายเพิ่มเติมได้ จะถูกทำสัญลักษณ์ “บวก” ต่อท้ายองค์ประกอบนั้นดังรูปที่ 28



รูปที่ 28: XML schema expand diagram

ที่มา OECD guide book (2019)

หลังจากทำการระบุรายละเอียดโครงสร้างข้อมูลของรายงาน CRS เรียบร้อยแล้วขั้นตอนต่อไปคือ การระบุรายละเอียดลักษณะของข้อมูล ดังแสดงในตารางที่ 4 ดังต่อไปนี้ 1.TAG หมายถึงองค์ประกอบย่อยที่เล็กที่สุดที่มีข้อมูลบรรจุอยู่ในองค์ประกอบนั้น 2.Option/Mandatory หมายถึงความจำเป็นที่ต้องระบุข้อมูลภายในรายงาน CRS โดย 1.option คือสามารถระบุข้อมูลหรือไม่ก็ได้ 2.mandatory คือจำเป็นต้องระบุข้อมูล 3.Example data หมายถึงตัวอย่างลักษณะของข้อมูลที่ต้องรายงานในแต่ละองค์ประกอบย่อย 4.Description หมายถึงคำอธิบายบริบทของข้อมูลว่ามีความหมายอย่างไรต่อรายงาน CRS

| TAG | Option/Mandatory | Example data | Description |
|----------------------|------------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SendingCompanyIN | Mandatory | 010711111111 | Number element |
| TransmittingCountry | Mandatory | | TH 2-character alphabetic country code and country name list1 based on the ISO 3166-1 Alpha 2 standard. |
| ReceivingCountry | Mandatory | | TH 2-character alphabetic country code based on the ISO 3166-1 Alpha 2 standard |
| MessageType | Mandatory | | CRS Allow only 'CRS' |
| Warning | Option | | data element is a free text field |
| Contact | Option | 633xx@student.chula.ac.th | data element is a free text field |
| MessageRefID | Mandatory | 01071111111111.2022.TH.0001 | data element is a free text field capturing the sender's unique message identifier (created by the sender) that identifies the particular message being sent |
| MessageTypelndic | Mandatory | CRS[701-703] | data element allows the sender to define the type of message sent allow for CRS[701-703] |
| CorrMessageRefID | Option | | This element is not to be used in the context of the CRS XML Schema. |
| ReportingPeriod | Mandatory | 2014-12-31 | yyyy-MM-DD format |
| Timestamp | Mandatory | 2015-03-15T09:45:30 | yyyy-MM-DD'T'h:mm:ss.nnn format |
| ResCountryCode | Mandatory | | TH 2-character alphabetic country code based on the ISO 3166-1 Alpha 2 standard |
| TIN | Option | 314000000 | This data element identifies the Tax Identification Number (TIN) used by the receiving tax administration to identify the Individual Account Holder |
| TIN issuedBy | Option | TW | 2-character alphabetic country code based on the ISO 3166-1 Alpha 2 standard |
| NamePerson_Type nz | Option | OECD[201-208] | It is possible for a CRS individual or entity to have several names. This is a qualifier to indicate the type of a particular name. |
| PrecedingTitle | Option | | data element is a free text field |
| Title | Option | MR | data element is a free text field |
| FirstName | Mandatory | NATTAPAT | data element is a free text field |
| FirstName xnlNameTy | Option | | data element is a free text field |
| MiddleName | Option | | data element is a free text field |
| MiddleName xnlName | Option | | data element is a free text field |
| NamePrefix | Option | | data element is a free text field |
| NamePrefix xnlName | Option | | data element is a free text field |
| LastName | Mandatory | DUNGDEE | data element is a free text field |
| LastName xnlNameTy | Option | | data element is a free text field |
| GenerationIdentifier | Option | | data element is a free text field |
| Suffix | Option | | data element is a free text field |
| GeneralSuffix | Option | | data element is a free text field |
| CountryCode | Mandatory | | TH 2-character alphabetic country code based on the ISO 3166-1 Alpha 2 standard |
| Address legalAddress | Option | OECD[301-305] | This is a datatype for an attribute to an address. It serves to indicate the legal character of that address (residential, business etc.) |
| AddressFree | Option* | hum Wan , Bangkok 10330 | data element is a free text field, *mean by if cannot fill the addressfix format only then will use addressfree |
| Street | Option | Phaya Tha1 Rd | data element is a free text field |
| BuildingIdentifier | Option | | data element is a free text field |
| SubIdentifier | Option | | data element is a free text field |
| FloorIdentifier | Option | | data element is a free text field |
| DistrictName | Option | Wang Ma1 | data element is a free text field |
| POB | Option | | data element is a free text field |
| PostCode | Option | 10330 | data element is a free text field |
| City | Mandatory | Pathum Wan | data element is a free text field |
| CountrySubentity | Option | Bangkok | data element is a free text field |
| Nationality | Option | | TH This data element is not required for CRS and should not be completed |
| BirthDate | Mandatory | 2014-12-31 | yyyy-MM-DD format |
| City | Option | | data element is a free text field |
| CitySubentity | Option | | data element is a free text field |
| CountryInfo | Option | TH/Thailand | choice between a current jurisdiction (identified by 2-character country code) or a former jurisdiction (identified by name). |
| CountryCode | Option | | TH 2-character alphabetic country code based on the ISO 3166-1 Alpha 2 standard |
| FormerCountryName | Option | | data element is a free text field |
| ResCountryCode | Mandatory | | TH 2-character alphabetic country code based on the ISO 3166-1 Alpha 2 standard |
| IN | Mandatory | US GIIN TIN | number (IN) used by the sending and/or receiving tax administration to identify the Entity Account Holder. For CRS this may be the US GIIN, a TIN, company registration number |
| IN issuedBy | Option | TH | 2-character alphabetic country code based on the ISO 3166-1 Alpha 2 standard |
| IN INTy | Option | TIN | This Attribute defines the type of identification number being sent (e.g. US GIIN, EIN, TIN). Possible values should normally be agreed between Competent Authorities. |
| Name | Mandatory | | data element is a free text field |
| Name nameType | Option | OECD[201-208] | It is possible for a CRS individual or entity to have several names. This is a qualifier to indicate the type of a particular name. |
| ResCountryCode | Mandatory | | TH 2-character alphabetic country code based on the ISO 3166-1 Alpha 2 standard |
| AccountNumber | Mandatory | 2008892290000 | Provide the account number used by the financial institution to identify the account |
| AccountNumber Acctf | Option | OECD60[1-5] | There is an option to include information about the account number type as an enumeration. The possible values are OECD60[1-5] |
| AccountNumber Undr | Mandatory | True/False | This attribute is for use in CRS domestic reporting to indicate that the account is undocumented |
| AccountNumber Clos | Mandatory | True/False | This attribute is for use in CRS reporting to indicate that the account is closed |
| AccountNumber Dorm | Option | True/False | This attribute may be used in CRS reporting to indicate that the account is dormant |
| Account Report Acctf | Mandatory | CRS[101-103] | a element may identify an entity account holder who is CRS[101-103] |
| CtrlPersonType | Option | CRS[801-813] | This data element allows the identification of the type of each Controlling Person ("CP") when available CRS[801-813] |
| AccountBalance | Mandatory | 2837 .10 | Provide the account balance or value of the reported financial account |
| AccountBalance currC | Mandatory | THB | All amounts must be accompanied by the appropriate 3 character currency code2 based on the ISO 4217 Alpha 3 standard |
| Type | Mandatory | CRS[501-504] | Select the proper code to identify the payment type. Specific payment types listed are CRS[501-504] |
| PaymentAmnt | Mandatory | 2837 .10 | Payment Amounts are entered with 2-digit fractional amounts of the currency in question. For example, USD 1 000 would be entered as 1000.00 |
| PaymentAmnt curr | Mandatory | THB | All payment amounts must be accompanied by the appropriate 3 character currency code based on the ISO 4217 Alpha 3 standard. |

ตารางที่ 4: mapping information characteristic

2. Transform Algorithm Design

จากการระบุโครงสร้างและลักษณะของข้อมูลทำให้ทราบถึงรายละเอียดของรูปแบบรายงาน CRS ในขั้นตอนต่อไปคือทำการออกแบบ algorithm โดยสามารถแบ่งออกเป็น 3 ระยะได้แก่ 1.Prototype Algorithm 2.Optimized Algorithm(final algorithm)

2.1. Prototype algorithm

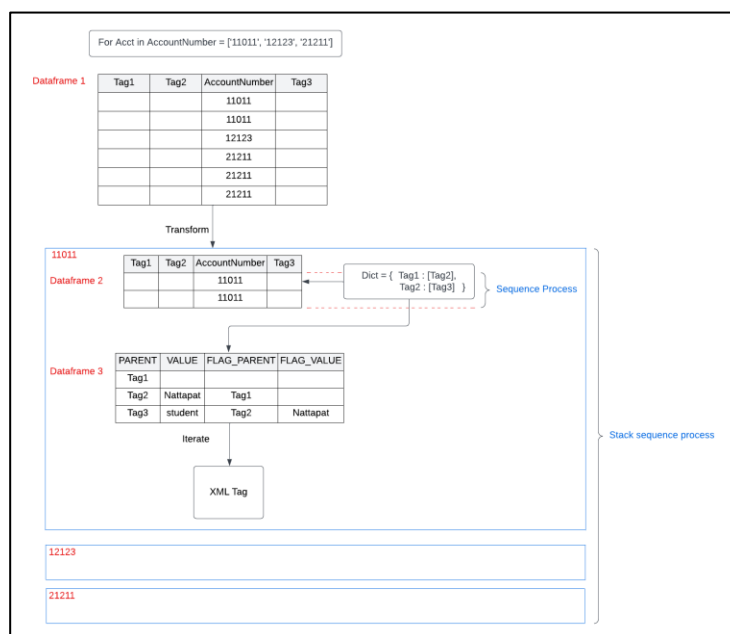
ในขั้นตอนการทำงานของ algorithm กำหนดให้มีการใช้งาน pandas library ในการทำงาน

ขั้นตอนที่ 1 ทำการวนลูป Dataframe 1 ซึ่งเป็นข้อมูลดิบด้วย AccountNumber ที่ไม่ซ้ำกัน และทำการแปลงข้อมูลเป็น Dataframe 2

ขั้นตอนที่ 2 ทำการวนลูป Dictionary ที่บรรจุองค์ประกอบของรายงานไว้ภายใน จากนั้นทำการนำข้อมูลในคอลัมน์ที่มีชื่อคอลัมน์ตรงกับองค์ประกอบนั้นๆ บรรจุลงใน Dataframe 3 โดยทำซ้ำจนครบทุก record ใน Dataframe 2

ขั้นตอนที่ 3 ทำการวนลูป Dataframe 3 เพื่อสร้างองค์ประกอบและเขียนบันทึกลงในรายงาน

โดยทำซ้ำ 3 ขั้นตอนดังกล่าวจนครบทุก AccountNumber ซึ่งการทำงานของ algorithm ในขั้นตอนนี้ในทุกๆการทำงานเป็นการทำงานแบบ sequence process ซึ่งต้องรอให้การทำงานก่อนหน้าเสร็จสิ้นก่อนจึงจะสามารถดำเนินการการทำงานถัดไป ดังแสดงในรูปที่ 29



รูปที่ 29: Baseline operation process of algorithm

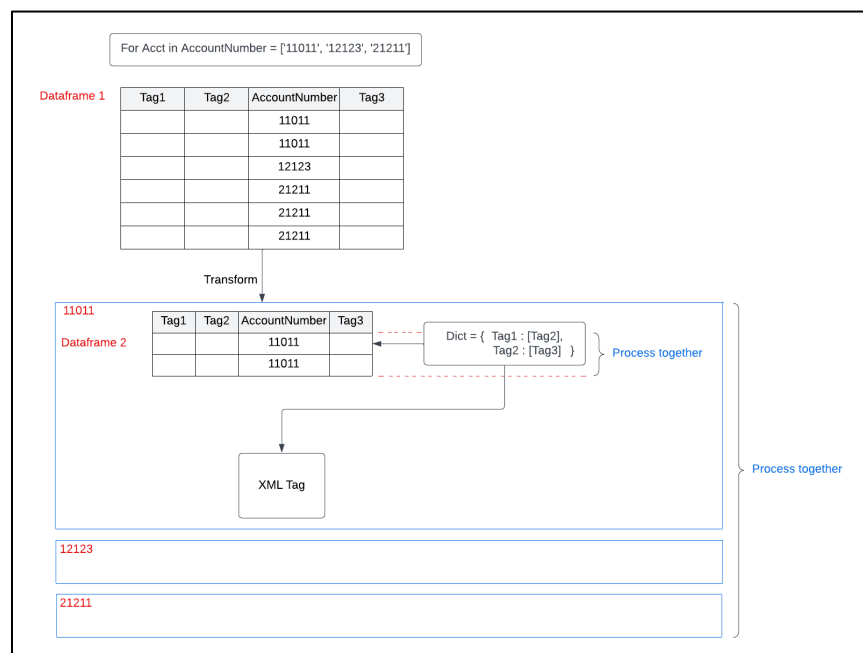
2.2. Optimized Algorithm

จากการวิเคราะห์การทำงานของ algorithm ในขั้นต้นพบว่าการทำงำนนั้นเป็นการทำงำนแบบ sequence process ซึ่งต้องรอให้การทำงานก่อนหน้าเสร็จสิ้นก่อนจึงจะสามารถดำเนินการการทำงานถัดไป ดังนั้นจึงได้ทำกำรมุ่งเน้นในการแก้ปัญหาดังกล่าวโดยการเปลี่ยนแปลง data structure ของ algorithm ซึ่งเปลี่ยนจาก pandas library เป็น pyspark library โดยการทำงานของ pyspark library นั้นจะเป็นการทำงำนแบบ Hadoop Distributed File System(HDFS) โดยมีขั้นตอนการทำงานดังนี้

ขั้นตอนที่ 1 ทำกำรวนรูป Dataframe 1 ซึ่งเป็นข้อมูลดิบด้วย AccountNumber ที่ไม่ซ้ำกันและทำกำรแปลงข้อมูลเป็น Dataframe 2

ขั้นตอนที่ 2 ทำกำรวนรูป Dictionary ที่บรรจุองค์ประกอบของรายงานไว้ภายใน จากนั้นทำกำรนำข้อมูลในคอลัมน์ที่มีชื่อคอลัมน์ตรงกับองค์ประกอบนั้นและชื่อขององค์ประกอบ เขียนบันทึกลงในรายงาน

โดยการทำงานทั้ง 2 ขั้นตอนจะเป็นการทำงานแบบ HDFS ซึ่งเป็นผลทำให้ทุก record ทำงานพร้อมกัน ดังแสดงในรูปที่ 30



รูปที่ 30: Target operation process of algorithm

จากการประยุกต์ใช้ในเรื่องของ data structure และ big o notation ในการช่วยลดความซับซ้อนของการทำงานใน algorithm ทำให้ได้ผลลัพธ์ในเรื่องของ runtime ดีขึ้น 7 เท่า โดยระยะเวลาลดลงจาก 1.08 นาที เหลือเพียง 9.19 วินาที ภายใต้เงื่อนไขการทดสอบในสภาพแวดล้อมการทำงานที่เหมือนกัน

CRS control framework & configuration design

1. CRS control framework on Azure Data Factory

จากรูปที่ 32 กระบวนการทำงานของ CRS data pipeline เริ่มต้นด้วยการนำเข้าข้อมูลดิบจากนั้นทำการ trigger data factory โดยก่อนเริ่มการทำงานผู้ใช้งานจำเป็นที่จะต้องทำการกำหนดค่าตัวแปรที่จำเป็นสำหรับการ process data factory และ processing notebook

| CNTL_UAT.CNTL_CFG_STEM | | | | |
|------------------------|------------|-------------|-------------|--------------|
| STREM_NM | DATA_DT | NXT_DATA_DT | PRV_DATA_DT | CALC_DATA_DT |
| EDP_CRS | 2022-12-31 | 2023-01-01 | 2022-12-30 | 2022-12-31 |

| CNTL_UAT.CNTL_CFG_PRCES_GRP | | | | | |
|-----------------------------|----------|------|-------|-------------------------|---------|
| PRCS_GRP | STREM_NM | PRIR | ACT_F | UPDT_DTTM | UPDT_BY |
| EDP_CRS_D | EDP_CRS | 1 | 1 | 2023-09-28 14:46:31.440 | INITIAL |

| CNTL_UAT.CNTL_CFG_PRCES | | | | | | | | | |
|-------------------------|-----------|------|---------|--------|-------|------------|-------------|---------|----------|
| PRCS_NM | PRCS_GRP | PRIR | CNCT_NM | SYS_NM | FI_ID | TGT_SYS_NM | TGT_SCHM_NM | TGT_TBL | PRCS_TYP |
| EDP_CRS_D_99 | EDP_CRS_D | 1 | NULL | NULL | NULL | NULL | NULL | NULL | 3 |

| ADB_NM | | ADB_PARM | | BYPS_F | ACT_F |
|----------------------------------|--|-----------------------------------------------------|--|--------|-------|
| /01_Staging/edp/adf/crs/CRS_TEST | | CRS_TEST1~\dbfs:/mnt/uat/working/TMP/JSON_CONFIG... | | 0 | 1 |

| CNTL_UAT.CNTL_CFG_DPND | | | | |
|------------------------|---------------|-------|-----------|---------|
| PRCS_NM | DPND_PRCES_NM | ACT_F | UPDT_DTTM | UPDT_BY |

รูปที่ 31: Configure setup detail

เมื่อทำการกำหนดข้อมูลใน SQL database แล้วจะสามารถเริ่มการทำงานโดยกรอกข้อมูลดังต่อไปนี้
 1.STREAM_NAME = EDP_CRS และ 2.RUN_MODE = F(Force Run) ซึ่งการทำงาน CRS data pipeline จะประกอบไปด้วย data pipeline ย่อยทั้งหมด 7 อันและ 1 processing notebook ดังต่อไปนี้

00_STREAM_COMMON มีหน้าที่ในการทำงานคือรับตัวแปร STREAM_NAME และ RUN_MODE หลักการทำงานคือจะทำการหาชื่อ STREAM_NM ที่เท่ากับ 'EDP_CRS' ในตาราง CNTL_UAT.CNTL_CFG_STEM ใน SQL database และทำการส่งต่อตัวแปร STREAM_NM, RUN_MODE, STREAM_ID เพื่อใช้สำหรับ Execute pipeline 05_COMMON_RUN_PCS_GRP_BY_PRIOR

05_COMMON_RUN_PCS_GRP_BY_PRIOR มีหน้าที่ในการทำงานคือดึงข้อมูลจากตาราง CNTL_UAT.CNTL_CFG_PCRS_GRP และทำการดึงชื่อ PCRS_GRP ที่มีชื่อ STREAM_NM เท่ากับ EDP_CRS จากตาราง CNTL_UAT.CNTL_CFG_PCRS_GRP และทำการเก็บค่าตัวแปร ได้แก่ PCRS_GRP, RUN_MODE และ STREAM_ID โดย pipeline 05_COMMON_RUN_PCS_GRP_BY_PRIOR สามารถทำการรัน job แบบ parallel ได้ หากในหนึ่ง STREAM_NAME มีหลาย PCRS_GRP ประกอบอยู่ จากนั้นจึงส่งต่อตัวแปรและทำการ Execute pipeline 10_COMMON_RUN_PCS_GRP

10_COMMON_RUN_PCS_GRP มีหน้าที่ในการทำงานคือดึงข้อมูลจากตาราง CNTL_UAT.CNTL_CFG_PCRS และทำการดึงค่า PCRS_NM จากตาราง CNTL_UAT.CNTL_CFG_PCRS ที่มีชื่อ PCRS_GRP เท่ากับตัวแปรที่รับมา และทำการเก็บค่าตัวแปร ได้แก่ PCRS_NM, RUN_MODE และ STREAM_ID จากนั้นจึงส่งต่อตัวแปรและทำการ Execute pipeline 20_PCS_COMMON_START

20_PCS_COMMON_START มีหน้าที่ในการทำงานคือตรวจสอบตัวแปร SKIP_PCRS ว่าเป็น True หรือ False ถ้าหากเป็น True จะรับตัวแปร PCRS_NM, RUN_MODE และทำการ Execute pipeline 30_PCS_COMMON_VALIDATE แต่หากเป็น False จะทำการ Cancel pipeline

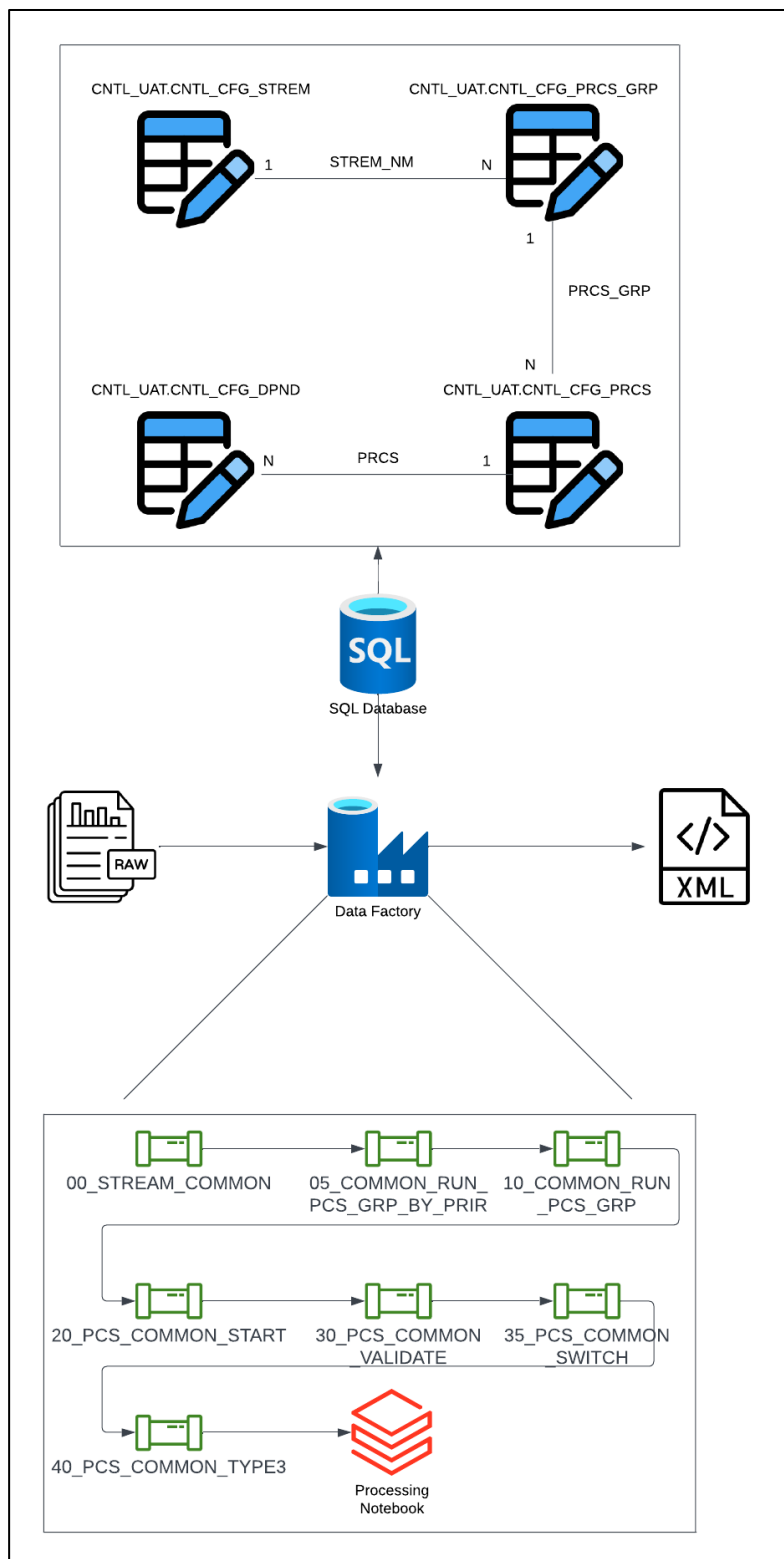
30_PCS_COMMON_VALIDATE มีหน้าที่ในการทำงานคือทำการรับตัวแปร PCRS_NM, RUN_MODE จากนั้นทำการตรวจสอบ dependency process กับตาราง CNTL_UAT.CNTL_CFG_DPND ถ้า DPND_FALG เท่ากับ N จะทำการยกเลิกการตรวจสอบ dependency process และทำการเก็บค่าตัวแปร ได้แก่ PCRS_NM, PCRS_TYPE, PCRS_OBJ โดย PCRS_TYPE คือ process type ที่ได้ทำการกำหนดค่าในตาราง CNTL_UAT.CNTL_CFG_PCRS และ PCRS_OBJ คือ ข้อมูลทั้งหมดจากตาราง CNTL_UAT.CNTL_CFG_PCRS และทำการ Execute pipeline 35_PCS_COMMON_SWITCH ถ้า DPND_FALG เท่ากับ Y จะทำการ Execute dependency process

35_PCS_COMMON_SWITCH มีหน้าที่ในการทำงานคือตรวจสอบเช็ค PCRS_TYPE ที่ได้รับจาก pipeline 30_PCS_COMMON_VALIDATE จากนั้นทำการเก็บค่าตัวแปร ได้แก่ PCRS_NM, PCRS_OBJ และทำการ Execute pipeline 40_PCS_COMMON_{PCRS_TYPE} ที่มีค่า PCRS_TYPE เท่ากับค่าที่กำหนด

40_PCS_COMMON_TYPE3 มีหน้าที่ในการทำงานคือ รัน processing notebook บน azure databricks ที่มีชื่อเดียวกับตัวแปร ADB_NM ที่ได้ทำการกำหนดค่าในตาราง CNTL_UAT.CNTL_CFG_PCRS

และส่งตัวแปรที่ชื่อ ADB_PARM ที่ได้ทำการกำหนดค่าในตาราง CNTL_UAT. CNTL_CFG_PCRS เป็นตัวแปรเริ่มต้นใน processing notebook และทำการรันเพื่อสร้างรายงานต่อไป

PROCESSING NOTEBOOK มีหน้าที่ในการทำงานคือ จะทำการ execute notebook ที่มีชื่อเดียวกับ ADB_NM และมีตัวแปรเริ่มต้นในการ process เท่ากับ ADB_PARAM โดยมีฟังก์ชันการทำงาน 4 แบบได้แก่ 1. แปลงข้อมูลดิบเป็นไฟล์ parquet โดยมีจุดประสงค์เพื่อทำการจัดเก็บสำรองไฟล์และเรียกใช้งาน 2. แปลงข้อมูลดิบให้อยู่ในรูปแบบตารางข้อมูลโดยทำการบันทึกไว้ที่ azure databricks และ synapse analytics 3. นำตารางข้อมูลใน azure databricks มาแปลงให้เกิดเป็นไฟล์รายงาน CRS 4. ทำการตรวจสอบโครงสร้างและรูปแบบข้อมูลของไฟล์รายงาน CRS ด้วยไฟล์ XML schema



รูปที่ 32: CRS Data pipeline detail

2. configuration design

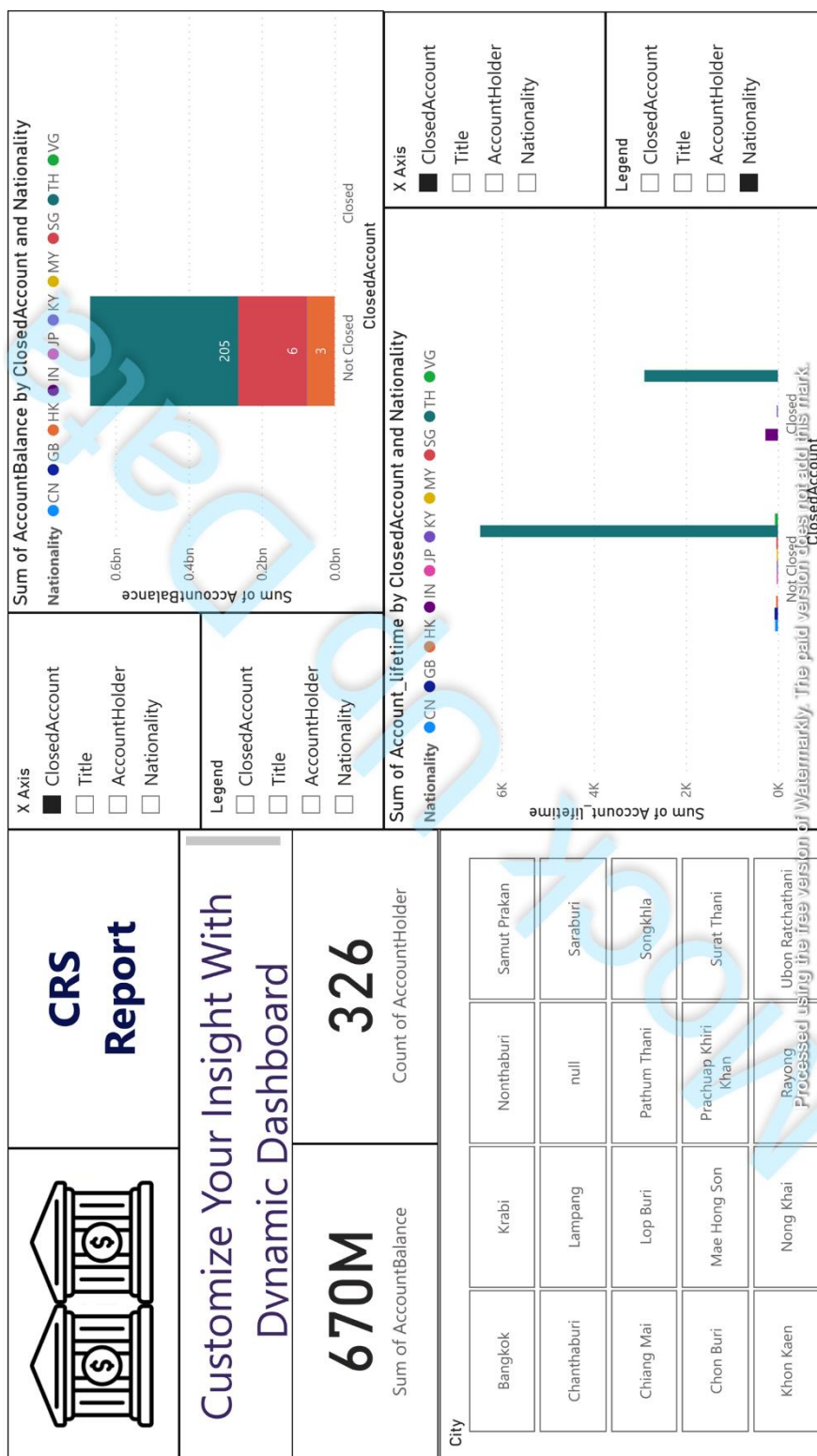
จากรูปที่ 31 ได้ทำการออกแบบการ กำหนดค่า ให้มีความยืดหยุ่นต่อการทำงานเนื่องจากผู้ใช้งานอาจมีความต้องการในเรื่องของสถานที่สำหรับวางไฟล์ต้นทางหรือไฟล์ปลายทางทั้งสำหรับจัดเก็บรายงานและไฟล์ parquet ในสถานที่ที่ต่างออกไปตามความต้องการของผู้ใช้งานโดยสิ่งที่อนุญาตให้ผู้ใช้งานสามารถปรับแต่งได้จะเป็นตัวแปรชื่อ ADB_PARM ซึ่งจะประกอบไปด้วย “source_input_file_name/source_input_file_path/target_table_name/target_xml_file_path/target_parquet_file_path/data_dt” โดยแต่ละตัวองค์ประกอบใน ADB_PARM มีความหมายดังนี้

| Parameter Name | Description |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| source_input_file_name | ชื่อของไฟล์ข้อมูลดิบที่ใช้สำหรับการสร้างรายงาน CRS |
| source_input_file_path | path ที่เก็บไฟล์ข้อมูลดิบใน azure storage account |
| target_table_name | ชื่อของตารางข้อมูล SQL ที่ใช้สำหรับจัดเก็บข้อมูลที่ azure databricks และ azure synapse analytics |
| target_xml_file_path | path ที่เก็บไฟล์รายงานใน azure storage account |
| target_parquet_file_path | path ที่เก็บไฟล์ parquet ใน azure storage account |
| target_parquet_file_path | path ที่เก็บไฟล์ parquet ใน azure storage account |
| data_dt | วันที่ที่เริ่มทำการ execute pipeline ที่ได้ทำการกำหนดค่าไว้ในตาราง CNTL_UAT.CNTL_CFG_STEM ใน SQL database ที่มีชื่อตัวแปรว่า DATA_DT |

ตารางที่ 5: parameter table

Power Bi Dashboard

หลังจากนำข้อมูลเก็บในรูปแบบตารางข้อมูลได้สำเร็จ จะทำการดึงข้อมูลของตารางข้อมูลจาก azure synapse analytics มาทำ data visualization โดยมีจุดประสงค์เพื่อบ่งบอกข้อมูลเชิงลึกที่ซ่อนอยู่ภายใต้ข้อมูล โดยได้มีการออกแบบเป็น dynamic dashboard ซึ่งสามารถให้ผู้ใช้งานสามารถปรับแต่งการแสดงผลของ dashboard ได้อย่างอิสระตามที่ผู้ใช้งานต้องการโดยเมื่อออกแบบเรียบร้อยแล้วจะทำการอัปโหลด dashboard ขึ้น คลาวด์เพื่อให้สามารถใช้งานจากที่ใดก็ได้ โดยได้ทำการจัดตั้ง refresh schedule ให้ refresh dashboard อัตโนมัติเพื่ออัปเดตข้อมูลหากข้อมูลมีการเปลี่ยนแปลงไป



รูปที่ 33: dynamic Power bi dashboard