



รายงานสหกิจศึกษาฉบับสมบูรณ์

การกู้ยืมเงินผ่านระบบดิจิทัลโดยระบบการอนุมัติสินเชื่อเบื้องต้น

Digital Lending (Lead Generation)

นางสาวชนนิกานต์ วรกานต์ศักดิ์

สาขาวิศวกรรมสารสนเทศ

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจศึกษา	การกู้ยืมเงินผ่านระบบดิจิทัลโดยระบบการอนุมัติสินเชื่อเบื้องต้น
ชื่อ-สกุล นักศึกษา	นางสาวชนนิกานต์ วรรณศักดิ์
คณะ วิศวกรรมศาสตร์	ภาควิชา วิศวกรรมคอมพิวเตอร์ สาขาวิชา วิศวกรรมสารสนเทศ
ชื่อ-สกุล อาจารย์นิเทศ	รศ.ดร.ชวลิต เบญจางคประเสริฐ
ชื่อ-สกุล ผู้นิเทศงาน	คุณเสฐียรพงษ์ เบญจมานุกูล
ชื่อสถานประกอบการ	บริษัท กลสิกร เทคโนโลยี กรุป เซเครเทรียต จำกัด

บทคัดย่อ

ปัจจุบันนี้ พบปัญหาการที่ประชาชนถูกตัดออกจากระบบการให้บริการทางการเงิน หรือการเข้าถึงการเงินขั้นพื้นฐาน จากปัญหาดังกล่าว จึงเป็นที่มาของความพยายามในการสร้างให้เกิด Financial Inclusion หรือการสร้างบริการที่ถือเป็นทางเลือกของโอกาสให้คนทุกกลุ่ม สามารถเข้าถึงบริการทางการเงินได้อย่างครบถ้วนและทั่วถึง ซึ่งสำหรับบริการการขอกู้สินเชื่อ นั้น จากการรวบรวมกรณีศึกษาพบว่ามีสองปัจจัยที่สามารถสร้างให้เกิด Financial Inclusion นั้น ได้แก่ การประเมินเครดิตของผู้กู้ โดยการใช้ข้อมูลรูปแบบใหม่ประกอบการพิจารณา เพื่อทำการเสนอสินเชื่อของผลิตภัณฑ์ต่าง ๆ ให้กับลูกค้าได้อย่างถูกกลุ่มเป้าหมาย และการสร้างช่องทางออนไลน์สำหรับการให้บริการขอกู้สินเชื่อ ดังนั้น เพื่อให้บรรลุจุดประสงค์ดังกล่าว จึงได้มีการจัดทำระบบการอนุมัติสินเชื่อเบื้องต้น เพื่อเอื้อให้ผู้กู้สามารถเข้าถึงบริการทางการเงินได้อย่างสะดวก ง่าย และโปร่งใส ผ่านแอปพลิเคชัน K-PLUS

คำสำคัญ : บริการทางการเงิน บริการขอกู้สินเชื่อ การอนุมัติผลิตภัณฑ์

Co-operative Title: Digital Lending (Lead Generation)
System Student Intern Name: Miss Chonnikarn Warakarnsak
Faculty: Engineering **Department:** Computer Engineering
Program: Information Engineering
Advisor Name: Assoc.Prof.Dr.Chawalit Benjangkprasert
Mentor Name: Mr.Sathianpong Benjamanukool
Company: Kasikorn Business-Technology Group (KBTG)

ABSTRACT

Nowadays, we have found the problem of people are eliminated from the financial services system or basic financial inaccessibility. From this problem, It is the source of the effort to create Financial Inclusion or creating services that give people access to financial services for all fully and thoroughly. Which for the loan request service From the collection of case studies, it is found that there are two factors that can create Financial Inclusion, including the credit evaluation of the borrower by using the new format of information for consideration. In order to offer credit for various products to customers correctly. And creating an online channel for providing loan financing. Therefore, there is a Digital Lending (Lead Generation) for specific groups of customers to enable borrowers to access financial services conveniently, easily, and transparently through the K-PLUS application.

keywords: Financial services system, Loan request service, Digital Lending (Lead Generation)

กิตติกรรมประกาศ

ข้าพเจ้าได้รับผิดชอบและปฏิบัติหน้าที่ในบริษัทสิกร เทคโนโลยี กรุ๊ป เซนครเทรียต จำกัด ระหว่างวันที่ 5 สิงหาคม ถึงวันที่ 22 พฤศจิกายน พ.ศ.2562 ในโครงการสหกิจศึกษาที่ทางคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และหน่วยงานฯ ร่วมมือกัน จัดตั้งขึ้น ในหัวข้อโครงการการกู้ยืมเงินผ่านระบบดิจิทัลโดยระบบการอนุมัติสินเชื่อเบื้องต้น ซึ่งข้าพเจ้าได้รับความรู้ ความเข้าใจและประสบการณ์ในการทำงานที่เป็นประโยชน์อย่างมาก อีกทั้งการดูแลและการช่วยเหลือต่าง ๆ ตลอดเวลาการทำงาน โดยการปฏิบัติงานสหกิจศึกษาในครั้งนี้สำเร็จลุล่วงได้ เพราะมีการชี้แนะ และได้รับความร่วมมือจากบุคคลต่าง ๆ ดังต่อไปนี้

วิศวกรข้อมูล นักวิเคราะห์ธุรกิจ และนักพัฒนาระบบ ทีม Application Development Groups

1. นายเสฐียรพงษ์ เบญจมานุกูล
2. นายสรวิทย์ พานิชประเสริฐ
3. นางอภาภรณ์ เรืองเทพ
4. นางสาวสุธารินี สาคำ
5. นายศิวกร อิทธิอาวัชกุล

และข้าพเจ้าขอขอบคุณอาจารย์ที่ปรึกษา รศ.ดร.ชวลิต เบญจางคประเสริฐ ที่คอยให้คำแนะนำ คำปรึกษาและคอยรับฟังและช่วยเหลือปัญหาต่าง ๆ ในการทำโครงการครั้งนี้ และท้ายที่สุดข้าพเจ้าขอขอบคุณครอบครัวที่คอยให้กำลังใจที่ดีแก่ข้าพเจ้าเสมอมา ทำให้ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

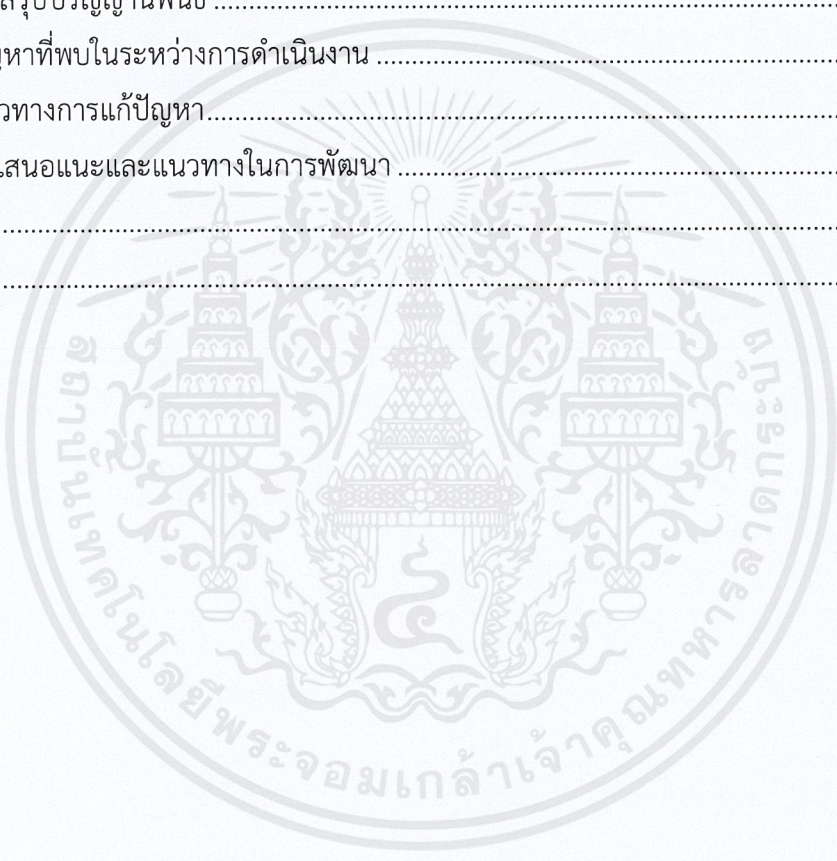
ชนนิกันต์ วรกานต์ศักดิ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ขอบเขตของการวิจัย.....	2
1.4 วิธีการดำเนินการวิจัย.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	5
1.5.1 ประโยชน์ต่อตนเอง.....	5
1.5.2 ประโยชน์ต่อองค์กร.....	5
บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	6
2.1 แนวคิดทฤษฎีเกี่ยวกับฐานข้อมูล.....	6
2.2 ทฤษฎีภาษา SQL (Structured Query Language).....	9
2.3 ทฤษฎีเกี่ยวกับเครื่องมือที่ใช้ในการพัฒนาโครงการ.....	11
2.3.1 เครื่องมือ Hadoop.....	11
2.3.2 เครื่องมือ Datastage.....	13
2.3.3 เครื่องมือ DbVisualizer.....	15
2.4 แนวคิดทฤษฎีเกี่ยวกับการทดสอบความถูกต้อง.....	16
บทที่ 3 วิธีดำเนินการวิจัย.....	18
3.1 การศึกษาข้อมูลขนาดใหญ่ขั้นพื้นฐาน (Standard Big Data) ของธนาคารกสิกรไทย.....	18
3.2 การศึกษาความต้องการของผู้ใช้ (Requirement).....	20
3.3 การออกแบบและวิเคราะห์ข้อมูล (Design and Analysis).....	21
3.3.1 เอกสาร CBR (Common Business Requirement).....	21
3.3.2 เอกสาร KBMF (K-Bank Message Format).....	23

สารบัญ (ต่อ)

	หน้า
3.4 การจัดการข้อมูลโดย Datastage.....	24
3.5 การทดสอบความถูกต้องโดยเอกสาร UIS	36
3.6 การนำงานไปใช้จริงบน Server (Deploy).....	39
3.7 การทดสอบความถูกต้อง.....	44
บทที่ 4 ผลการวิจัย	45
บทที่ 5 บทสรุปและวิจารณ์ผลการดำเนินงาน.....	49
5.1 บทสรุปปริญาานิพนธ์	49
5.2 ปัญหาที่พบในระหว่างการดำเนินงาน	49
5.3 แนวทางการแก้ปัญหา.....	49
5.4 ข้อเสนอแนะและแนวทางในการพัฒนา	50
เอกสารอ้างอิง	51
ประวัติผู้เขียน.....	53



สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 1.1 รายละเอียดการทำงานตั้งแต่เดือนสิงหาคม-พฤศจิกายน 2562	3
ตารางที่ 2.1 คำสั่งที่ใช้ในการนิยามข้อมูล	10
ตารางที่ 2.2 คำสั่งที่ใช้ในการจัดการข้อมูล.....	10
ตารางที่ 2.3 คำสั่งที่ใช้ในการควบคุมข้อมูล.....	10



สารบัญภาพ

ภาพที่	หน้า
ภาพที่ 2.1 ฐานข้อมูล.....	6
ภาพที่ 2.2 Relational database	6
ภาพที่ 2.3 Distributed database.....	7
ภาพที่ 2.4 Cloud database.....	7
ภาพที่ 2.5 NoSQL database	8
ภาพที่ 2.6 เครื่องคอมพิวเตอร์ที่ถูกจัดอยู่ในรูปแบบ Cluster.....	12
ภาพที่ 2.7 กระบวนการจัดการข้อมูล.....	13
ภาพที่ 2.8 สถาปัตยกรรมของ ETL.....	14
ภาพที่ 2.9 Test Pyramid.....	16
ภาพที่ 3.1 ข้อมูลขนาดใหญ่ขั้นพื้นฐาน (Standard Big Data) ของธนาคารกสิกรไทย.....	19
ภาพที่ 3.2 เว็บไซต์ฐานข้อมูลของบริษัท	19
ภาพที่ 3.3 การเก็บ Requirement จาก User	20
ภาพที่ 3.4 ดึงข้อมูลตาม Requirement ที่ได้มาโดยใช้ภาษา SQL.....	20
ภาพที่ 3.5 ส่วนบนของเอกสาร CBR.....	21
ภาพที่ 3.6 คอลัมน์ Source ของเอกสาร CBR.....	22
ภาพที่ 3.7 คอลัมน์ Physical Table ของเอกสาร CBR.....	22
ภาพที่ 3.8 คอลัมน์ View ของเอกสาร CBR.....	23
ภาพที่ 3.9 ส่วนบนของเอกสาร KBMF.....	23
ภาพที่ 3.10 ส่วนเนื้อหาของเอกสาร KBMF	24
ภาพที่ 3.11 ไฟล์ ddl สำหรับขา IN.....	25
ภาพที่ 3.12 ไฟล์ ddl สำหรับขา OUT	25
ภาพที่ 3.13 ไฟล์ ddl สำหรับขา VIEW	26
ภาพที่ 3.14 ไฟล์ Parallel ในโปรแกรม Datastage.....	27
ภาพที่ 3.15 การกรอกข้อมูลคอลัมน์จาก Source.....	28
ภาพที่ 3.16 การกรอกข้อมูลในเครื่องมือ Tranformer_LEAD_CFG	28
ภาพที่ 3.17 การกำหนดพารามิเตอร์ที่จะต้องใช้ในไฟล์ Parallel.....	29
ภาพที่ 3.18 การประมวลผลงานในโปรแกรม Datastage.....	29
ภาพที่ 3.19 พารามิเตอร์ทั่วไปของบริษัท.....	30

สารบัญญภาพ (ต่อ)

หน้า

ภาพที่ 3.20 การกำหนดพารามิเตอร์ที่ต้องใช้ในไฟล์ Sequence	31
ภาพที่ 3.21 การกำหนดค่าให้กับทุกเครื่องมือในไฟล์ Sequence.....	31
ภาพที่ 3.22 การกรอกข้อมูลในเครื่องมือ LoadEBAN_ LEAD_CFG_SELF_EMP_INL_HIST.....	32
ภาพที่ 3.23 การกรอกข้อมูลในเครื่องมือ BIGDATA_TRG_ LEAD_CFG_SELF_EMP_INL_HIST.....	32
ภาพที่ 3.24 การกรอกข้อมูลในเครื่องมือ RCNCL_TRG_ LEAD_CFG_SELF_EMP_INL_HIST.....	33
ภาพที่ 3.25 การกำหนดพารามิเตอร์ที่ต้องใช้ในงาน FM	34
ภาพที่ 3.26 การกำหนดค่าให้กับทุกเครื่องมือในไฟล์งาน FM	34
ภาพที่ 3.27 การกรอกข้อมูลในเครื่องมือ SEQ_ LEAD_CFG_SELF_EMP_INL_HIST.....	35
ภาพที่ 3.28 การประมวลผลงาน FM_ LEAD_CFG_SELF_EMP_INL_HIST สำเร็จ.....	35
ภาพที่ 3.29 เอกสาร UIS ส่วนแรก	36
ภาพที่ 3.30 ตัวอย่างการเทียบข้อมูลในเอกสาร UIS	37
ภาพที่ 3.31 ตรวจสอบงานด้วยโปรแกรม Director Client ในเอกสาร UIS.....	37
ภาพที่ 3.32 ตรวจสอบจำนวนข้อมูลด้วยโปรแกรม DbVisualizer ในเอกสาร UIS	38
ภาพที่ 3.33 การเทียบข้อมูลจากเอกสาร CBR กับข้อมูลจาก Datastage ในเอกสาร UIS.....	38
ภาพที่ 3.34 กรอกชื่องาน Datastage ที่จะทำการ Deploy เพื่อสร้างตารางใหม่ในฐานข้อมูล	39
ภาพที่ 3.35 หน้าต่างของ Generate_dstage_deploy_package.bat	40
ภาพที่ 3.36 ไฟล์ใหม่	40
ภาพที่ 3.37 รอกการประมวลผล.....	41
ภาพที่ 3.38 ชื่องาน Datastage ปรากฏในโพลเดอร์ deploy	41
ภาพที่ 3.39 การ Pack Code ในส่วนของ DDL	42
ภาพที่ 3.40 การสร้างไฟล์ 01 สำหรับ Pack Code.....	42
ภาพที่ 3.41 การสร้างไฟล์ 02 สำหรับ Pack Code.....	43
ภาพที่ 3.42 การสร้างไฟล์ 03 สำหรับ Pack Code.....	43
ภาพที่ 3.43 การสร้างไฟล์ 04 สำหรับ Pack Code.....	43
ภาพที่ 4.1 ช่องทางแรกของการสมัครสินเชื่อ.....	45
ภาพที่ 4.2 ช่องทางที่ 2 ของการสมัครสินเชื่อ.....	46
ภาพที่ 4.3 ช่องทางที่ 3 ของการสมัครสินเชื่อ.....	46

สารบัญญภาพ (ต่อ)

	หน้า
ภาพที่ 4.4 วงเงินสินเชื่อและอัตราดอกเบี้ย.....	47
ภาพที่ 4.5 กระบวนการยินยอมสมัครสินเชื่อ.....	48



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

บริษัท กลสิกร เทคโนโลยี กรุ๊ป เซเคเรเทรียต จำกัด ได้จัดโครงการสหกิจศึกษาาระหว่างบริษัท กลสิกร เทคโนโลยี กรุ๊ป เซเคเรเทรียต จำกัด กับคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โดย บริษัท กลสิกร เทคโนโลยี กรุ๊ป เซเคเรเทรียต จำกัด เป็นองค์กรที่เป็นผู้นำด้านเทคโนโลยีในการสร้างอาคารแห่งอนาคตแบบดิจิทัลที่สมบูรณ์แบบ เพื่อให้ลูกค้าสามารถเข้าถึงบริการและทำกิจกรรมทางการเงินทุกประเภท ในทุกช่องทาง เน้นการให้บริการที่ง่าย สะดวก รวดเร็วและปลอดภัยเป็นส่วนประกอบสำคัญในชีวิตประจำวันของลูกค้า และช่วยลูกค้าตัดสินใจทางการเงิน เพื่อให้การใช้ชีวิตในรูปแบบที่เป็นดิจิทัลในทุกวันนี้ได้อย่างไร้ขีดจำกัด

บริษัท กลสิกร เทคโนโลยี กรุ๊ป เซเคเรเทรียต จำกัด ^[1] มุ่งมั่นในการสร้างความสามารถด้านเทคโนโลยีและนวัตกรรมที่ให้บริการลูกค้าในรูปแบบอัตโนมัติและครบวงจร

- สร้างทีมงานที่มีความแข็งแกร่ง ทั้งด้านธุรกิจ และเทคโนโลยีสารสนเทศที่ทันสมัย
- ร่วมมือกับพันธมิตรที่มีแนวคิดและรูปแบบบริการด้านดิจิทัลที่น่าสนใจ สามารถมาเสริมความแข็งแกร่งและหลากหลายให้กับอาคารรูปแบบดิจิทัลได้
- สร้างวัฒนธรรมการทำงานที่เน้นการสร้างคุณค่า มีความคิดริเริ่มสร้างสรรค์ และกระบวนการทำงานที่รวดเร็ว มีความยืดหยุ่นสูง
- สร้างกระบวนการให้บริการลูกค้าที่เป็นแบบอัตโนมัติ มีบริการและข้อมูลที่ครบถ้วน หลากหลาย และตรงจุด สามารถตอบโจทย์ความต้องการของลูกค้าได้อย่างรวดเร็วและทันเวลา

บริษัท กลสิกร เทคโนโลยี กรุ๊ป เซเคเรเทรียตจำกัดนั้น มีความประสงค์ที่จะเตรียมความพร้อมก่อนเข้าทำงานจริงในอนาคต ด้วยการคิดจริง ปฏิบัติจริง โดยทีมงานมืออาชีพที่พร้อมด้วยประสบการณ์ คอยแนะนำและส่งต่อความรู้มากมาย เป็นความรู้ใหม่ที่ได้จากการทำงานจริง ซึ่งหาไม่ได้จากการเรียนในมหาวิทยาลัย ได้พัฒนาทักษะมากมาย ไม่ว่าจะเป็นทักษะในการพูด การกล้าแสดงออก การกล้าเสนอความคิดเห็นในห้องประชุม ความรู้ทางเทคนิคมากมาย

การเข้าร่วมโครงการสหกิจศึกษากับทางบริษัท กลสิกร เทคโนโลยี กรุ๊ป เซเคเรเทรียตจำกัด ในตำแหน่งวิศวกรข้อมูล (Data Engineer) ซึ่งวิศวกรข้อมูล หรือ Data Engineer นั้น คือคนที่ช่วยสร้างระบบในการส่งข้อมูล (Data Pipeline) และระบบจัดเก็บข้อมูล เช่น Data Warehouse หรือ Data Lake อาชีพนี้เน้นความสามารถด้านการเขียนโปรแกรม และความรู้เกี่ยวกับเครื่องมือการจัดการ Big Data ที่หลากหลายเครื่องมือให้เลือกใช้ คนที่เป็น Data Engineer ควรตอบได้ว่าในงานแบบนี้ควรใช้เครื่องมือแบบไหน จัดเตรียมข้อมูลเพื่อส่งต่อให้ผู้ที่ต้องการใช้ข้อมูลนำไปใช้ได้อย่างมีประสิทธิภาพ

ทางโครงการการกู้ยืมเงินผ่านระบบดิจิทัล ในส่วนระบบการอนุมัติสินเชื่อเบื้องต้น (Digital Lending : Lead Generation) มีจุดประสงค์ที่จะทำการตรวจสอบ และกรองลูกค้าที่สามารถได้รับการอนุมัติจากผลิตภัณฑ์ต่าง ๆ เช่น สินเชื่อบ้าน วงเงินกู้เบิกเกินบัญชี (Overdraft) บัตรเครดิต และบัตรเครดิตเงินสด เป็นต้น ซึ่งลูกค้าที่มีศักยภาพที่ทางธนาคารจะให้เงินกู้ ลูกค้าที่ได้แสดงความสนใจในสิ่งที่แอปพลิเคชันนำเสนอ และมีข้อมูลช่วยให้ระบบตัวต้นได้แน่นอน ซึ่งลูกค้าดังกล่าวสามารถเรียกสั้น ๆ ได้ว่า Lead ^[2] ลูกค้าเหล่านี้จะเข้ามาเห็นการเสนอการอนุมัติผลิตภัณฑ์ต่าง ๆ ในแอปพลิเคชัน K-PLUS เมื่อลูกค้าสนใจที่จะทำสินเชื่อต่าง ๆ ผ่านระบบดิจิทัล ลูกค้าจะสามารถทำได้อย่างง่ายดาย สะดวก รวดเร็ว และปลอดภัย นอกจากนี้ สิ่งที่สำคัญคือ การทำให้ผู้ใช้เกิดความคุ้นเคยและรู้สึกสบายใจที่จะใช้งาน ผ่านการสร้างความปลอดภัยให้เกิดขึ้น เพราะท้ายที่สุดแล้ว กลุ่มที่จะได้รับประโยชน์สูงสุดก็คือตัวลูกค้าหรือผู้บริโภคนั่นเอง หากผู้คนจำนวนมากขึ้นสามารถเข้าถึงบริการทางการเงินได้อย่างทั่วถึง นั่นนอกระบบก็จะลดลง และเป็นผลไปสู่การแก้ปัญหาความยากจนของผู้คนได้ในระยะยาว

1.2 วัตถุประสงค์ของการวิจัย

- 1) เพื่อทำการอนุมัติผลิตภัณฑ์ต่าง ๆ เช่น สินเชื่อบ้าน วงเงินกู้เบิกเกินบัญชี (Overdraft) บัตรเครดิต และบัตรเครดิตเงินสด เป็นต้น ให้กับลูกค้าที่มีศักยภาพที่ทางธนาคารจะให้เงินกู้
- 2) เพื่อให้ลูกค้าสามารถเข้าถึงบริการและทำกิจกรรมทางการเงิน การทำสินเชื่อต่าง ๆ ผ่านระบบดิจิทัล ที่สามารถทำได้อย่างง่ายดาย สะดวก รวดเร็ว และปลอดภัย เป็นส่วนประกอบสำคัญในชีวิตประจำวันของลูกค้า
- 3) เพื่อช่วยลูกค้าตัดสินใจทางการเงิน ให้การใช้ชีวิตในรูปแบบที่เป็นดิจิทัลในทุกวันได้อย่างไร้ขีดจำกัด
- 4) เพื่อลดปัญหานั้นนอกระบบ และเป็นผลไปสู่การแก้ปัญหาความยากจนของผู้คนได้ในระยะยาว

1.3 ขอบเขตการวิจัย

ขอบเขตงานตลอดระยะเวลาเข้าร่วมโครงการสหกิจศึกษาที่ได้รับมอบหมายให้รับผิดชอบจากหน่วยงาน แบ่งออกเป็น 6 ส่วนดังนี้

- 1) ศึกษา Requirement ของ K-Bank Personal Loan (KPL) และ K-Bank Commercial Loan (KCL)
- 2) ศึกษา Standard Big Data ของธนาคารกสิกรไทย
- 3) จัดทำเอกสาร Mapping
 - เอกสาร CBR (Common Business Requirement) เป็นข้อมูลที่ทำการส่งให้กับผู้พัฒนา (Developer) ทำการพัฒนาระบบ

-เอกสาร KBMF (K-Bank Message Format) เป็นเอกสารให้ผู้ใช้ข้อมูลสามารถเข้าใจข้อมูลได้ดียิ่งขึ้น รู้ว่าข้อมูลนั้น ๆ ย่อมาจากคำว่าอะไร หมายความว่าอย่างไร และตัวอย่างของข้อมูลเป็นอย่างไร

4) Development Dastage เป็น ETL Tool ซึ่ง ETL ย่อมาจาก Extract Transform และ Load คือการดึงข้อมูลจาก Data Source ต่าง ๆ เข้าสู่ Data Warehouse

5) ทดสอบระบบ และ Deploy ระบบ

6) Performance Verification Test (PVT) เป็นการทดสอบระบบอีกครั้งกับ User ว่าตรงตามความต้องการของ User หรือไม่

7) Integration Verification Test (IVT) เป็นการทดสอบระบบอีกครั้ง โดย Developer หรือ Business Analyst เป็นผู้ทดสอบระบบด้วยตนเอง เพื่อตรวจดูว่าระบบถูกต้องหรือไม่

1.4 วิธีการดำเนินการวิจัย

ในการที่จะทำโครงการการกู้ยืมเงินผ่านระบบดิจิทัล ในส่วนของระบบการอนุมัติสินเชื่อเบื้องต้น (Digital Lending : Lead Generation) นั้น มีการวางแผนการทำงานในขั้นตอนต่าง ๆ ซึ่งสามารถจำแนกออกได้ตามแต่ละส่วนของขอบเขตการทำงานดังตารางที่ 1.1

ตารางที่ 1.1 รายละเอียดการทำงานตั้งแต่เดือนสิงหาคม-พฤศจิกายน 2562

หัวข้อ	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1) ศึกษาข้อมูลพื้นฐานขนาดใหญ่ (Standard Big Data) ของธนาคารกสิกรไทย																
2) ศึกษา Requirement ของ K-Bank Personal Loan (KPL)																
3) ศึกษา Requirement ของ K-Bank Commercial Loan (KCL)																

ตารางที่ 1.1 รายละเอียดการทำงานตั้งแต่เดือนสิงหาคม-พฤศจิกายน 2562

หัวข้อ	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
4) จัดทำเอกสาร Mapping เอกสาร KBMF (K-Bank Message Format)			■	■												
5) จัดทำเอกสาร Mapping เอกสาร CBR (Common Business Requirement)					■	■	■	■								
6) ประชุมกับ User เพื่อให้ได้ข้อสรุป Requirement					■	■	■	■								
7) ศึกษา Structured Query Language (SQL)					■	■	■	■								
8) เขียน SQL Query ของแต่ละ Requirement เพื่อให้เข้าใจ Requirement มากยิ่งขึ้น					■	■	■	■								
9) Development Datastage ทำการจัดการข้อมูล									■	■						
10) Pack Code เพื่อเตรียมนำงานไป Deploy											■					
11) ทดสอบระบบ และ Deploy ระบบ												■	■	■		
12) Performance Verification Test (PVT)															■	■
13) Integration Verification Test (IVT)															■	■

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่ได้รับจากโครงการการกู้ยืมเงินผ่านระบบดิจิทัล ในส่วนของระบบ การอนุมัติสินเชื่อเบื้องต้น (Digital Lending : Lead Generation) กับทางบริษัททศสิกร เทคโนโลยี กรุ๊ป เซครเทรียต จำกัด สามารถแบ่งได้ 2 ส่วนดังนี้

1.5.1 ประโยชน์ต่อตนเอง

- 1) ได้รับความรู้เกี่ยวกับกระบวนการทำงานด้าน Data ได้รู้ถึง Standard Big Data ของธนาคาร ทศสิกรไทย
- 2) ได้ฝึกฝนการ Query โดยใช้ภาษา SQL (Structured Query Language) ได้ฝึกฝนการใช้ ETL Tool นั่นคือ Datastage และได้ฝึกฝนการต่อรองและสื่อสารในที่ประชุม
- 3) ได้เรียนรู้การปฏิบัติงานในสถานที่จริง การปรับตัวเข้ากับสภาพแวดล้อมการทำงานในหน่วยงาน
- 4) ได้ฝึกการทำงานร่วมกับผู้อื่น การมีระเบียบวินัยตรงต่อเวลา มีความรับผิดชอบ ในหน้าที่ของตนเอง
- 5) เรียนรู้อุปสรรค และการแก้ไขปัญหาในเหตุการณ์ต่าง ๆ ในการทำงาน ทั้งจากการ ปรึกษาพี่ ๆ ในแผนก เพื่อนร่วมงาน และศึกษาเรียนรู้ด้วยตนเอง
- 6) ได้ฝึกการสื่อสาร การนำเสนอผลงาน ทำให้มีมนุษยสัมพันธ์ที่ดีขึ้น และเพิ่มความกล้าแสดงออก

1.5.2 ประโยชน์ต่อองค์กร

เนื่องจากบริษัท ทศสิกร เทคโนโลยี กรุ๊ป เซครเทรียต จำกัด เป็นองค์กรที่เป็นผู้นำด้านเทคโนโลยี ในการสร้างธนาคารแห่งอนาคตแบบดิจิทัลที่สมบูรณ์แบบ เพื่อให้ลูกค้าสามารถเข้าถึงบริการและทำกิจกรรมทางการเงินทุกประเภท ในทุกช่องทาง เน้นการให้บริการที่ง่าย สะดวก รวดเร็วและปลอดภัย เป็นส่วนประกอบสำคัญในชีวิตประจำวันของลูกค้า และช่วยลูกค้าตัดสินใจทางการเงิน เพื่อให้การใช้ชีวิตในรูปแบบที่เป็นดิจิทัลในทุกวันนี้ได้อย่างไร้ขีดจำกัด ดังนั้น ประโยชน์ที่หน่วยงานจะได้รับมีดังนี้

- 1) สามารถนำระบบการกู้ยืมเงินผ่านระบบดิจิทัล ในส่วนของระบบ การอนุมัติสินเชื่อเบื้องต้น (Digital Lending: Lead Generation) ไปใช้งานได้ เพื่อช่วยลูกค้าตัดสินใจทางการเงิน
- 2) ข้อมูลที่ถูกเก็บไว้ได้นำมาใช้ให้เกิดประโยชน์
- 3) สามารถตอบโจทย์ความต้องการของผู้ใช้งานได้มากขึ้น

บทที่ 2

แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดทฤษฎีเกี่ยวกับฐานข้อมูล

Database (ฐานข้อมูล) ^[3] คือส่วนที่ใช้เก็บข้อมูล เพื่อให้สามารถบริการการจัดเก็บ และการเข้าถึงข้อมูลส่วนกลางได้สะดวกรวดเร็ว ซึ่งข้อมูลจะถูกเก็บในรูปของ rows รูปของcolumns และ tables โดยอาศัยดัชนี (indexed) เพื่อให้สามารถค้นหาข้อมูลได้รวดเร็ว นอกจากนี้ยังสามารถทำการเพิ่ม (insert) ลด (delete) หรือ เปลี่ยนแปลง (update) ข้อมูลใหม่ได้อีกด้วย

ปัจจุบัน Computer database ถูกใช้ในการเก็บข้อมูลแทนในรูปแบบ files เช่น ยอดขาย จำนวนผลิตภัณฑ์ และ คลังเก็บสินค้า รวมถึงข้อมูลของลูกค้าเอง โดยระบบสามารถมีได้ตั้งแต่ระดับ mainframe ขนาดใหญ่ จนมาถึงแค่ PC หรือ Laptop ตามต้องการของ user

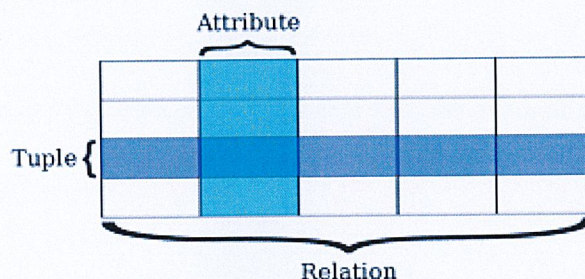


ภาพที่ 2.1 ฐานข้อมูล

Database ถูกพัฒนาขึ้นมาตั้งแต่ปี 1960 เริ่มต้นจาก hierarchical และ network databases จนมาถึงปี 1980 มีการนำเอา object-oriented-databases (OODBMS) มาใช้งาน ซึ่งเป็นพื้นฐานของระบบ relation database ที่ใช้กันอยู่ในปัจจุบันนี้

ในอีกมุมหนึ่ง สามารถจัดแบ่งประเภทของ database ตามรูปแบบของชนิดข้อมูลได้ เช่น ตัวเลข ตัวอักษร หรือ รูปภาพ บางครั้งก็อาจจะแบ่งตามความนิยมของ relational database เช่น distributed database หรือ cloud database หรือ NoSQL database.

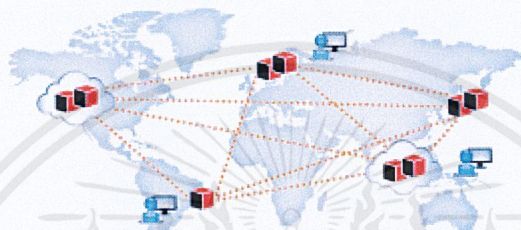
Relational database



ภาพที่ 2.2 Relational database

Relational database ถูกคิดค้นขึ้นโดย E.F. Codd (IBM) ในปี 1970 เริ่มต้นสร้างขึ้นมาจากกลุ่มของ table ที่มีข้อมูลภายในโดยแบ่งออกเป็นตามประเภทที่ตั้งไว้ แต่ละ table จะมีอย่างน้อย 1 ชนิดของ แต่ละ column และแต่ละ row จะมีข้อมูลตามที่ชนิดที่ columns ได้กำหนดไว้ Standard Query Language (SQL) เป็นภาษาที่ผู้ใช้งาน และ ระบบอื่น ๆ ทั่วไปเชื่อมต่อกับ relational database ซึ่งง่ายต่อการเพิ่มข้อมูลเข้าไป โดยไม่กระทบต่อโปรแกรมอื่นที่ใช้งานร่วมกันอยู่

Distributed database



ภาพที่ 2.3 Distributed database

Distributed database เป็นฐานข้อมูลที่ถูกเก็บกระจายออกไปหลาย ๆ ที่ โดยอาศัยกระบวนการแจกจ่าย และ สืบค้นข้อมูล ผ่านทางระบบ network ซึ่งมีอยู่ด้วยกัน 2 แบบคือ

- 1) homogeneous - ระบบทั้งหมดทุกที่ต้องเป็น OS และ database ชนิดเดียวกัน
- 2) heterogeneous - ระบบทั้งหมดจะเหมือนหรือต่างกันได้ในแต่ละที่

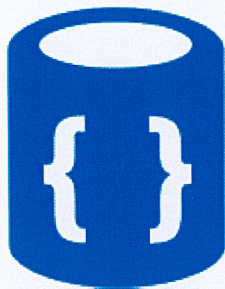
Cloud database



ภาพที่ 2.4 Cloud database

Cloud database เป็นฐานข้อมูลแบบใหม่ ที่ถูกปรับปรุงและสร้างขึ้นบนระบบ virtualized แบบเดียวกับ hybrid cloud และ public cloud หรือ private cloud โดยสามารถขยายขนาดเพิ่มขึ้น หรือ ปรับแต่ง resource ได้ตลอดเวลา ขึ้นอยู่กับความต้องการของผู้ใช้งาน

NoSQL database



ภาพที่ 2.5 NoSQL database

NoSQL database ถูกใช้ในรูปแบบ ที่เป็นการกระจายของข้อมูล จึงมีประสิทธิภาพสูงสำหรับ ข้อมูลขนาดใหญ่ (big data) เพราะ relational database ไม่ถูกออกมาให้รองรับข้อมูลขนาดใหญ่ จึงนิยม ใช้กับการวิเคราะห์ข้อมูลขนาดใหญ่ที่ไม่ค่อยมีรูปแบบตายตัว การ access database มีด้วยกัน 2 แบบคือ

- Database management system (DBMS) เป็น software ที่ควบคุมและบริหารข้อมูลภายใน ฐานข้อมูล

- Relational database management system (RDBMS) ถูกพัฒนาขึ้นในปี 1970 เพื่อเข้าถึง ฐานข้อมูลแบบ relational และยังคงได้รับความนิยมจนถึงปัจจุบัน

ปัจจุบันมี production database ให้เลือกใช้อยู่มากมาย เช่น โปรแกรม Microsoft SQL Server โปรแกรม Oracle Database โปรแกรม IBM โปรแกรม DB2 และโปรแกรม MySQL ที่เป็นที่ยอดนิยมใน ระดับองค์กร

Database server^[4] คือ เครื่องบริการข้อมูล ที่เปิดให้ผู้ใช้เพิ่มข้อมูล ลบ หรือแก้ไข สำหรับ โปรแกรมบริการระบบฐานข้อมูลที่ยอดนิยมใช้ได้แก่ MYSQL หรือ Microsoft Access เป็นต้น โดยผู้ใช้งาน ต้องเขียนโปรแกรมส่งประมวลผล ปรับปรุงข้อมูล หรือนำข้อมูลในส่วนที่ตนเองมีสิทธิ์ ไปใช้ตามต้องการ

ระบบฐานข้อมูล (Database System) หมายถึงระบบที่รวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกันเข้าไว้ด้วยกันอย่างมีระบบมีความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่ชัดเจน ในระบบฐานข้อมูลจะประกอบด้วย แฟ้มข้อมูลหลายแฟ้มที่มีข้อมูล เกี่ยวข้องสัมพันธ์กันเข้าไว้ด้วยกันอย่างเป็นระบบและเปิดโอกาสให้ผู้ใช้ สามารถใช้งานและดูแลรักษา ป้องกันข้อมูลเหล่านี้ ได้อย่างมีประสิทธิภาพ ในการนำข้อมูลมาจัดเก็บไว้ใน คอมพิวเตอร์ก็เช่นเดียวกันคือ จะต้องสร้างแฟ้มข้อมูลขึ้นมาสำหรับจัดเก็บข้อมูลที่เกี่ยวข้องกันเป็นเรื่อง ๆ โดยแต่ละแฟ้มข้อมูลจะประกอบด้วยข้อมูลส่วนย่อย ๆ หรือฟิลด์ (Field) หลายชิ้นที่มีความสัมพันธ์ เกี่ยวข้องกันและข้อมูลแต่ละชุดที่จัดเก็บในแฟ้มหรือระเบียบ (Record) การจัดเก็บในลักษณะนี้จะแยก

ออกเป็นแฟ้ม ๆ คล้ายกับการจัดเก็บในแฟ้มเอกสารที่เป็นกระดาษปกติ แต่มีประสิทธิภาพการใช้งานและการบำรุงรักษาที่ดีกว่า

2.2 ทฤษฎีภาษา SQL (Structured Query Language)

SQL ^[5] ย่อมาจาก structured query language คือภาษาที่ใช้ในการเขียนโปรแกรม เพื่อจัดการกับฐานข้อมูลโดยเฉพาะ เป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์และเป็นระบบเปิด (open system) หมายถึงสามารถใช้คำสั่ง SQL กับฐานข้อมูลชนิดใดก็ได้ และ คำสั่งงานเดียวกันเมื่อส่งงานผ่าน ระบบฐานข้อมูลที่แตกต่างกันจะได้ ผลลัพธ์เหมือนกัน ทำให้สามารถเลือกใช้ฐานข้อมูล ชนิดใดก็ได้ โดยไม่ติดขัดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง นอกจากนี้แล้ว SQL ยังเป็นชื่อโปรแกรมฐานข้อมูล ซึ่งโปรแกรม SQL เป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง โปรแกรม SQL จึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์ และเป็นภาษาหนึ่ง ซึ่งแบ่งการทำงานได้เป็น 4 ประเภท ดังนี้

1. Select query ใช้สำหรับดึงข้อมูลที่ต้องการ
2. Update query ใช้สำหรับแก้ไขข้อมูล
3. Insert query ใช้สำหรับการเพิ่มข้อมูล
4. Delete query ใช้สำหรับลบข้อมูลออกไป

ปัจจุบันมีซอฟต์แวร์ระบบจัดการฐานข้อมูล (DBMS) ที่สนับสนุนการใช้คำสั่ง SQL เช่น โปรแกรม Oracle โปรแกรม DB2 โปรแกรม MS-SQL และโปรแกรม MS-Access เป็นต้น

นอกจากนี้ภาษา SQL ถูกนำมาใช้เขียนร่วมกับโปรแกรมภาษาต่าง ๆ เช่น ภาษา C หรือ C++ โปรแกรม VisualBasic และภาษา Java

ประเภทของคำสั่งภาษา SQL

1. ภาษานิยามข้อมูล (Data Definition Language: DDL) เป็นคำสั่งที่ใช้ในการสร้างฐานข้อมูล กำหนดโครงสร้างข้อมูลว่ามี Attribute ใด ชนิดของข้อมูล รวมทั้งการเปลี่ยนแปลงตาราง และการสร้างดัชนี ดังตารางที่ 2.1

ตารางที่ 2.1 คำสั่งที่ใช้ในการนิยามข้อมูล

ลำดับ	คำสั่ง	คำอธิบาย
1	CREATE DATABASE	ใช้ในการสร้างฐานข้อมูล
2	ALTER DATABASE	ใช้ในการปรับเปลี่ยนฐานข้อมูล
3	CREATE TABLE	ใช้ในการสร้างตารางข้อมูล
4	ALTER TABLE	ใช้ในการปรับเปลี่ยนตาราง
5	DROP TABLE	ใช้ในการลบตาราง
6	CREATE INDEX	ใช้ในการสร้างดัชนี
7	DROP INDEX	ใช้ในการลบดัชนี

2. ภาษาจัดการข้อมูล (Data Manipulation Language: DML) เป็นคำสั่งที่ใช้ในการเรียกใช้ เพิ่ม ลบ และเปลี่ยนแปลงข้อมูลในตาราง ดังตารางที่ 2.2

ตารางที่ 2.2 คำสั่งที่ใช้ในการจัดการข้อมูล

ลำดับ	คำสั่ง	คำอธิบาย
1	SELECT	ใช้ในการดึงข้อมูลออกมาจากฐานข้อมูล
2	UPDATE	ใช้ในการแก้ไขเปลี่ยนแปลงข้อมูลในฐานข้อมูล
3	INSERT	ใช้ในการเพิ่มข้อมูลในฐานข้อมูล
4	DELETE	ใช้ในการลบข้อมูลในฐานข้อมูล

3. ภาษาควบคุมข้อมูล (Data Control Language: DCL) เป็นคำสั่งที่ใช้ในการกำหนดสิทธิ์การอนุญาต หรือ ยกเลิก การเข้าถึงฐานข้อมูล เพื่อป้องกันความปลอดภัยของฐานข้อมูล ดังตารางที่ 2.3

ตารางที่ 2.3 คำสั่งที่ใช้ในการควบคุมข้อมูล

ลำดับ	คำสั่ง	คำอธิบาย
1	GRANT	ใช้กำหนดสิทธิ์ของผู้ใช้ในการเข้าถึงทรัพยากรของระบบฐานข้อมูล
2	REVOKE	ใช้ยกเลิกหรือเรียกคืนสิทธิ์ที่เคยให้ไว้ ทำให้ผู้ใช้ที่ถูกยกเลิกสิทธิ์ ไม่สามารถใช้งานทรัพยากรเดิมได้

ประโยชน์ของภาษา SQL [6]

- ใช้ในการสืบค้นข้อมูลในฐานข้อมูล
- ใช้ใส่ข้อมูลเพิ่มเข้าไปยังฐานข้อมูล
- ใช้ปรับปรุงข้อมูลในฐานข้อมูล
- ใช้ลบรายการที่ไม่ต้องการออกจากฐานข้อมูล
- ใช้สร้างฐานข้อมูลขึ้นมาใหม่
- ใช้สร้างตาราง(Table) ในฐานข้อมูล
- ใช้สร้าง Stored Procedure ในฐานข้อมูล
- ใช้สร้าง Views ในฐานข้อมูล
- ใช้กำหนดสิทธิ์ให้กับตาราง (Table) รวมถึง Procedure และ Views

โดยแม้ว่า SQL เป็นมาตรฐานแต่ก็ยังมีหลายเวอร์ชัน ซึ่งก็จะมีแตกต่างกันออกไป ในแต่ละผลิตภัณฑ์อีกด้วย แต่โครงสร้างหลักในการใช้คำสั่งจัดการข้อมูลตามตารางที่ 2.2 จะมีโครงสร้างเดียวกัน

การใช้งานภาษา SQL ไปใช้งาน [7]

- ใช้กับเว็บไซต์ เพื่อแสดงผลข้อมูลจากฐานข้อมูล DBMS ไม่ว่าจะเป็นโปรแกรม Microsoft Access โปรแกรม SQL Server โปรแกรม MySQL และโปรแกรม Oracle
- ใช้ร่วมกับระบบฐานข้อมูล RDBMS ไม่ว่าจะเป็นโปรแกรม MS SQL Server โปรแกรม IBM โปรแกรม DB2 โปรแกรม Oracle โปรแกรม MySQL และโปรแกรม Microsoft Access
- ใช้ในการกำหนดในระบบวิเคราะห์ข้อมูล (Analysis Tools) ที่เปิดช่องทางให้สามารถทำการใส่ หรือ ปรับปรุงภาษา SQL ได้ด้วยตัวเอง

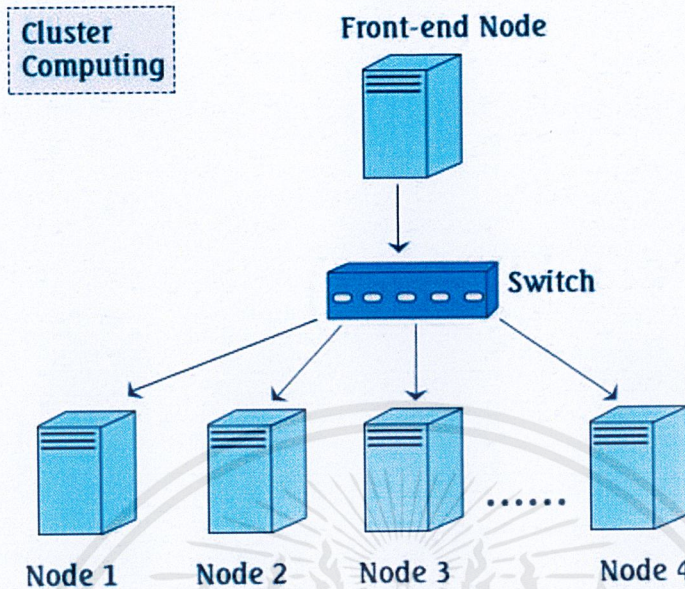
ดังนั้นหากผู้พัฒนาสามารถใช้งานภาษา SQL ได้ ย่อมจะเป็นประโยชน์ในการทำความเข้าใจในการทำงานกับระบบฐานข้อมูลมากยิ่งขึ้น อ่านความหมายของสิ่งที่ใช้งานได้อย่างลึกซึ้งขึ้นอีกด้วย

2.3 ทฤษฎีเกี่ยวกับเครื่องมือที่ใช้ในการพัฒนาโครงการงาน

2.3.1 เครื่องมือ Hadoop

เมื่อกกล่าวถึง Big Data เครื่องมือที่จะถูกสนใจเป็นอย่างมากในปัจจุบัน นั่นคือ Hadoop^[8] และ Hadoop คือ ซอฟต์แวร์ประเภท Open Source ที่จัดทำขึ้นเพื่อเป็นแพลตฟอร์มในการจัดเก็บข้อมูล ซึ่งมีการออกแบบมาเพื่อใช้ในการจัดเก็บข้อมูลและประมวลผลข้อมูลที่มีขนาดใหญ่มาก ๆ ที่เรียกกันว่า Big Data ซึ่ง Hadoop สามารถปรับขยาย ยืดหยุ่น เพื่อรองรับข้อมูลที่มีจำนวนมากมายมหาศาลได้ ทั้งนี้ก็เพราะมีกระบวนการประมวลผลที่แข็งแกร่ง ซึ่งเป็นผลมาจากการประมวลผลข้อมูลแบบกระจายผ่านเครื่อง

คอมพิวเตอร์ที่ถูกจัดอยู่ในรูปแบบ Cluster ดังภาพที่ 2.6 อันนำไปสู่ความสามารถในการรองรับข้อมูลที่ไม่จำกัด และยังมีความน่าเชื่อถือสูงอีกด้วย



ภาพที่ 2.6 เครื่องคอมพิวเตอร์ที่ถูกจัดอยู่ในรูปแบบ Cluster

ในส่วนของการทำงาน โครงสร้างของ Hadoop จะเป็นแบบ Cluster โดยแบ่งออกเป็น 2 ส่วนคือ Master server และ Slave server ซึ่งมีการทำงานหลัก ๆ อยู่ 3 ส่วนด้วยกัน ได้แก่

- 1) Cluster management
- 2) Storage
- 3) Processing

ประโยชน์ของ Hadoop มีดังนี้

- 1) ความสามารถในการรองรับการจัดเก็บข้อมูลขนาดใหญ่มาก ๆ ทุกประเภทได้อย่างรวดเร็ว ด้วยปริมาณข้อมูลในปัจจุบันที่เพิ่มขึ้นอย่างต่อเนื่อง โดยเฉพาะจากแหล่งอย่างโซเชียลมีเดีย และ Internet of Things (IoT) คุณสมบัติข้อนี้ของ Hadoop จึงสำคัญมาก
- 2) พลังแห่งการประมวลผล ด้วยรูปแบบการประมวลผลที่รวดเร็วจากการทำงานแบบ Cluster จึงทำให้ Hadoop กลายเป็นแพลตฟอร์มที่เป็นที่นิยมอย่างกว้างขวางในปัจจุบัน
- 3) มีระบบรองรับความผิดพลาด ด้วยการทำงานแบบ Cluster เมื่อ node ใด node หนึ่งพังลงงานที่มีการทำอยู่ในระบบจะถูกส่งไปยัง node อื่นทันทีเพื่อให้เกิดความต่อเนื่อง รวมถึงระบบเองยังมีการทำข้อมูลสำรองเก็บไว้แบบอัตโนมัติหลายชุดอีกด้วย

4) ความยืดหยุ่นในการใช้งาน Hadoop แตกต่างจากระบบฐานข้อมูลดั้งเดิม ที่ต้องมีการแยกประเภทของข้อมูลคร่าว ๆ ก่อนการจัดเก็บ สำหรับ Hadoop จะเก็บข้อมูลประเภทไหนก็ได้ มากเท่าไรก็ได้ทันที โดยไม่ต้องมีการแยกประเภทล่วงหน้าแถมยังสามารถเลือกได้อีกว่าจะเอาไปใช้งานด้านใด

5) ต้นทุนต่ำ เพราะเป็นแพลตฟอร์มแบบ Open Source จึงสามารถนำมาใช้งานได้ฟรี!

6) ความสามารถในการขยายการรองรับข้อมูลได้ไม่สิ้นสุด แค่เพิ่ม node เข้าไปก็สามารถรองรับการจัดเก็บข้อมูลไปได้เรื่อย ๆ ตามต้องการ
เครื่องมือที่สามารถทำงานร่วมกับ Hadoop ได้

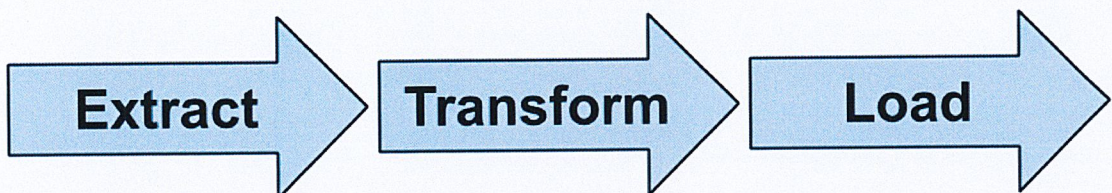
- Spark
- NoSQL
- Pig
- Samza
- Hive
- Cascading
- etc.

2.3.2 เครื่องมือ Datastage

การจัดการข้อมูลโดย ETL Tool ^[9] ซึ่ง ETL ย่อมาจาก Extract ย่อมาจาก Transform และย่อมาจาก Load คือการดึงข้อมูลจาก Data Source ต่าง ๆ เข้าสู่ Data Warehouse ซึ่ง Datastage ก็เป็น ETL Tool ชนิดหนึ่ง

ETL Tool คือเครื่องมือที่ใช้ในการจัดการกระบวนการสำคัญต่าง ๆ กับข้อมูล ดังนี้

1. Extract คือ การสกัดข้อมูล การดึงข้อมูลจากแหล่งข้อมูลที่แตกต่างกัน
2. Transform คือ การนำข้อมูลที่ได้มาจากการ Extract มาจัดรูปแบบให้ถูกต้อง
3. Load คือ การนำข้อมูลที่ผ่านการ Transform แล้ว เข้าสู่ Data Warehouse หรือ Data Mart หรือ Database



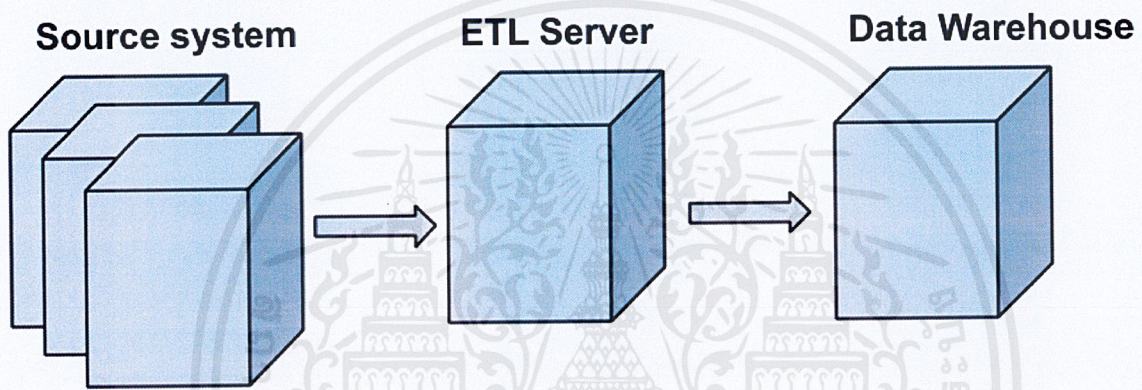
ภาพที่ 2.7 กระบวนการจัดการข้อมูล

ลักษณะพิเศษ

ETL ส่วนมากจะนิยมใช้ในระบบ integrate data จากหลาย ๆ แอปพลิเคชัน พัฒนาจากหลาย ๆ ผู้ขาย หรือจากหลาย ๆ แหล่งติดตั้งที่แตกต่างกัน

สถาปัตยกรรมของ ETL

1. การสกัดโดยใช้งาน staging area คือหลังจากเลือกข้อมูลบางส่วนจากแหล่งข้อมูล จะทำการโอนข้อมูลไปยัง staging area จากนั้นข้อมูลจะถูกประมวลผลก่อนส่งไปยังคลังข้อมูลอีกที
2. การสกัดโดยใช้หน่วยความจำ การสกัดข้อมูลบางส่วนจากแหล่งข้อมูล แล้วทำการเปลี่ยนแปลงข้อมูลในหน่วยความจำ แล้วจึงส่งไปยังคลังข้อมูลอีกที



ภาพที่ 2.8 สถาปัตยกรรมของ ETL

ขั้นตอนการทำงานของ ETL

ประกอบด้วย 3 ขั้นตอนการทำงานหลักคือ การสกัดข้อมูล การเปลี่ยนแปลงข้อมูล และการโอนถ่ายข้อมูล แต่แท้จริงยังสามารถแจ้งรายละเอียดได้ดังต่อไปนี้

- การวางแผนสำหรับการรวบรวมข้อมูล
- การกำหนดกฎสำหรับการเปลี่ยนแปลง หรือเปลี่ยนรูปแบบ และการทำความสะอาด
- การสร้างกฎในการสกัดข้อมูล
- การเตรียมการเชื่อมโยงข้อมูลจากแหล่งข้อมูล
- การรวบรวมข้อมูลจากแหล่งข้อมูลต่าง ๆ ทั้งแหล่งข้อมูลภายในและภายนอก
- การกำหนดข้อมูลทั้งหมดที่ต้องการเก็บไว้ในคลังข้อมูล
- การรวมโครงสร้างข้อมูลจากหลายแหล่งข้อมูลไปเป็นข้อมูลเพียงแถวเดียว
- การแยกโครงสร้างข้อมูลหนึ่ง ๆ ไปเป็นข้อมูลที่มีหลายโครงสร้างเพื่อสร้างเป็นข้อมูลหลาย ๆ แถว
- การอ่านข้อมูลจากพจนานุกรมข้อมูลที่ถูกเก็บไว้ในแหล่งข้อมูล

- การอ่านข้อมูลจากหลาย ๆ แหล่งข้อมูล
- การโหลดรายละเอียดสำหรับสร้าง Table
- การรวบรวมข้อมูลหรือ ทำผลสรุปข้อมูลให้กับ Table
- การแปลงข้อมูลจากรูปแบบหนึ่งของข้อมูลไปเป็นอีกรูปแบบหนึ่งของ Table
- การรับเอาข้อมูลที่เป็นเป้าหมายจาก Field ต่าง ๆ ของแหล่งข้อมูล
- การเปลี่ยนข้อมูลที่มีความกำกวมให้มีความหมายมากขึ้น

ลักษณะการใช้งาน ETL tool

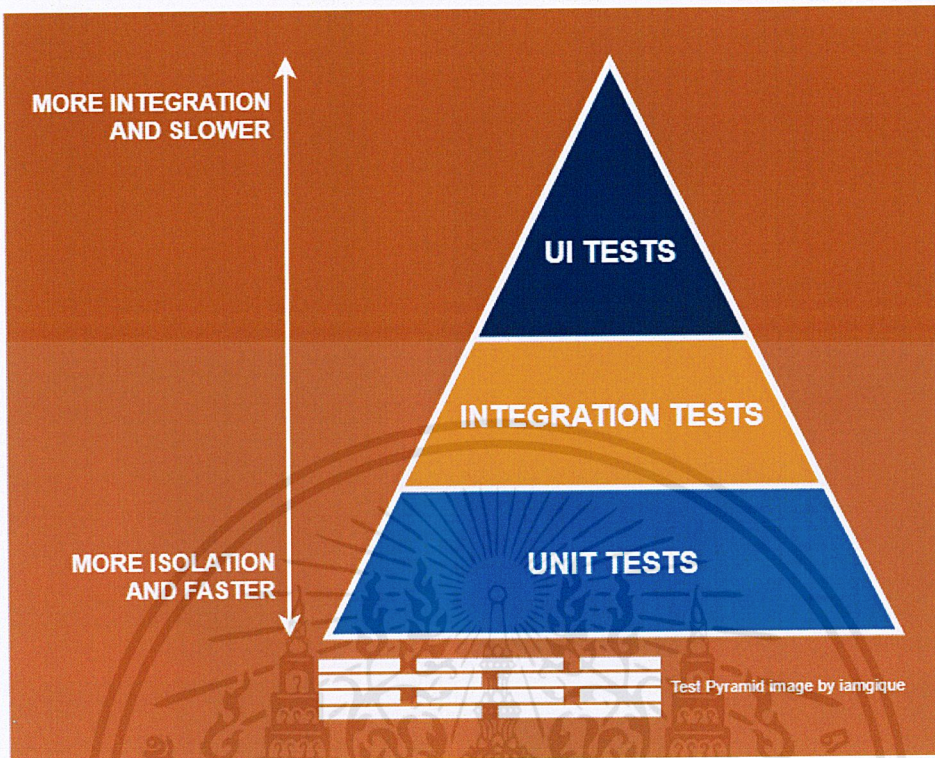
การใช้งาน ETL tool สามารถแบ่งออกเป็น 2 รูปแบบใหญ่ ๆ

1. การดึงข้อมูลจากระบบเก่ามาใช้ในระบบใหม่สำหรับ Migration
2. การดึงข้อมูลจากหลาย ๆ แหล่งมารวมกันเพื่อเตรียมใช้ใน BI

2.3.3 เครื่องมือ DbVisualizer

DbVisualizer^[10] เป็นเครื่องมือฐานข้อมูลข้ามแพลตฟอร์มที่มีเป้าหมายเพื่อเพิ่มประสิทธิภาพของผู้ใช้ จะนำเสนอโซลูชันเดียว เพื่อเพิ่มความเร็วในการพัฒนาทดสอบและบริหารจัดการฐานข้อมูลเชิงสัมพันธ์ สำหรับนักพัฒนาซอฟต์แวร์ DbVisualizer มีเครื่องมือที่หลากหลาย เพื่อช่วยในการพัฒนาฐานข้อมูลและปฏิบัติงานต่าง ๆ เช่น การเรียกค้นวัตถุ การสร้าง การเรียกใช้คำสั่ง SQL พร้อมด้วยการดู และแก้ไขข้อมูล DbVisualizer มีประสิทธิภาพสูงและสนับสนุนฐานข้อมูลเชิงพาณิชย์ที่สำคัญทั้งหมด เช่น โปรแกรม Oracle โปรแกรม DB2 โปรแกรม Mimer โปรแกรม SQL Server โปรแกรม Sybase ASE โปรแกรม Informix และ Open Source เช่น โปรแกรม MySQL โปรแกรม PostgreSQL และโปรแกรม JavaDB

2.4 แนวคิดทฤษฎีเกี่ยวกับการทดสอบความถูกต้อง



ภาพที่ 2.9 Test Pyramid

Test Pyramid เป็น concept หนึ่งที่ใช้เรียก กลุ่มของการทำ Software Tests โดยจะแบ่งการ Test ออกเป็น 3 ส่วน ดังนี้

- 1) Unit Tests
- 2) Integration Tests
- 3) User Interface Tests

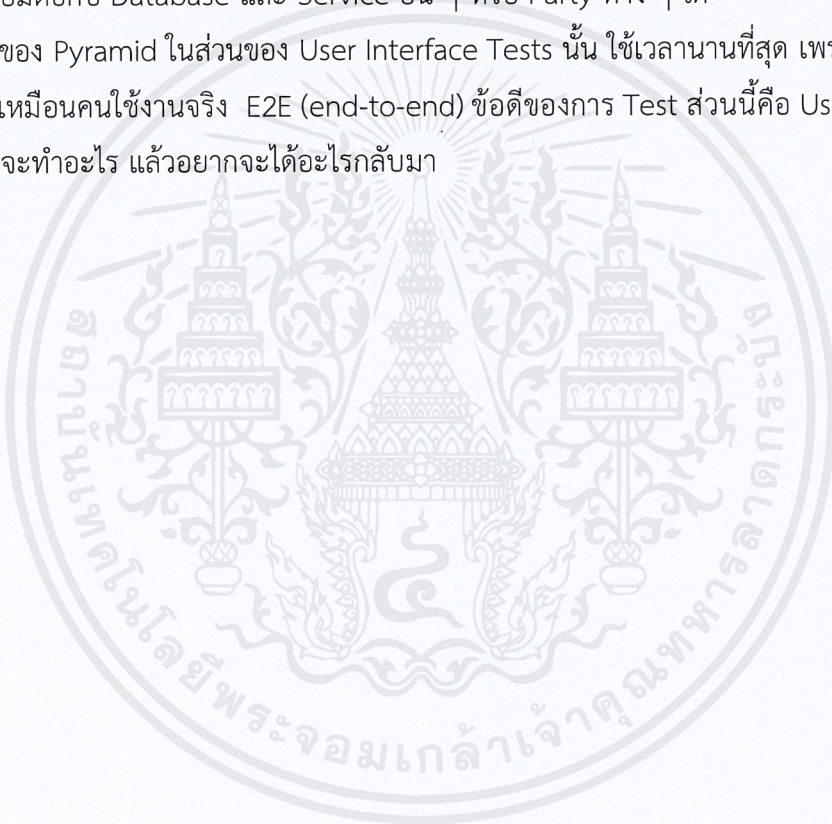
เริ่มกันที่ส่วนฐานของ Pyramid นั้นคือ Unit tests การทำ Unit tests นั้น เป็นการ Test function ของการทำงานในระดับ Unit หรือ คือ Test การทำงานในหน่วยเล็กที่สุด เห็นได้เร็วที่สุดว่าส่วนไหนทำงานผิดพลาด แต่ไม่การันตีความถูกต้อง เพราะการทำ Unit Tests มันไม่ได้พ่วงการทำงานกับจุดอื่น ปัจจุบันที่นิยมกันมักจะเป็นการทำ TDD (Test Driven Develop) การที่จะทำ Unit tests ได้นั้น สิ่งที่เราเหล่า Developer พึงกระทำต่อ Code ก็คือ เขียนให้มีลักษณะ Testable ดังนี้

- 1) Easy to write: เขียนให้ง่าย
- 2) Readable: ทุกคนที่อ่านสามารถเข้าใจได้ง่าย เมื่อกลับมาอ่านอีกครั้งก็ยังสามารถเข้าใจได้

- 3) Reliable: น่าเชื่อถือ นั่นคือหาก Developer มีการแก้ไข แล้วโปรแกรมเกิดการผิดพลาด Unit Tests ควรจะรู้
- 4) Fast: เขียนให้เรียกใช้งานหลายครั้งได้ง่าย รวดเร็ว หรือเป็น Function นั้นเอง
- 5) Truly unit. Not integration.: Unit Tests ไม่ใช้การ Integrate ไม่ต้องต่อ Database ไม่ต้องต่อกับ Service อื่น ดังนั้นสนใจเฉพาะหน่วยเล็กที่สุดนี้เท่านั้น

ส่วนกลางของ Pyramid เป็นส่วนของ Integration Tests ในส่วนนี้ จะเป็นการ Tests ระดับของ Service เป็นส่วนที่ไม่ใช้การทำ Test function เล็ก ๆ แต่เป็นการ Test function ทั้ง function ที่เหล่านักพัฒนาจัดทำมาให้ ซึ่งในส่วนนี้ จะเป็นการรวมถึงการที่ Service จะต้องทำงานได้ สามารถไปเชื่อมต่อกับ Network เชื่อมต่อกับ Database และ Service อื่น ๆ หรือ Party ต่าง ๆ ได้

ส่วนบนที่สุดของ Pyramid ในส่วนของ User Interface Tests นั้น ใช้เวลานานที่สุด เพราะส่วนนี้เป็นส่วนการทดสอบเหมือนคนใช้งานจริง E2E (end-to-end) ข้อดีของการ Test ส่วนนี้คือ User เห็น และรู้ว่าตัวเองต้องการจะทำอะไร แล้วอยากจะได้อะไรกลับมา



บทที่ 3

วิธีการดำเนินการวิจัย

เมื่อได้ศึกษาสภาพแวดล้อมและสิ่งต่าง ๆ ภายในองค์กรจนมีความรู้ความเข้าใจในกระบวนการทำงาน ตลอดจนถึงข้อจำกัดและข้อพึงระวัง ซึ่งได้รับมอบหมายในการพัฒนาระบบการอนุมัติสินเชื่อเบื้องต้น (Lead Generation) เพื่อให้บริษัทได้นำไปใช้ และในครั้งนี้ได้มีการศึกษาความต้องการต่าง ๆ ของผลิตภัณฑ์ K-Bank Commercial Loan (KCL) ในช่วงแรกของการทำงาน จะใช้เวลาส่วนใหญ่ไปกับเรื่องการศึกษาความต้องการของผลิตภัณฑ์และการศึกษาข้อมูลพื้นฐานขนาดใหญ่ (Standard Big Data) ของธนาคารกสิกรไทย โดยมีกระบวนการดำเนินการวิจัยต่าง ๆ ดังนี้

3.1 การศึกษาข้อมูลขนาดใหญ่ขั้นพื้นฐาน (Standard Big Data) ของธนาคารกสิกรไทย

ข้อมูลขนาดใหญ่ขั้นพื้นฐาน (Standard Big Data) ของธนาคารกสิกรไทยจะถูกจัดเก็บในฐานข้อมูล ซึ่งฐานข้อมูลที่ไม่ว่าจะต้องทำการดึงข้อมูลหรือใส่ข้อมูลเข้าไปนี้ เรียกว่า EBAN ซึ่งฐานข้อมูล EBAN นี้ จะประกอบไปด้วยข้อมูลหลากหลายฐานข้อมูล หลากหลายแฟ้มข้อมูล และหลากหลายชนิดข้อมูล ซึ่งประกอบไปด้วยฐานข้อมูลดังนี้

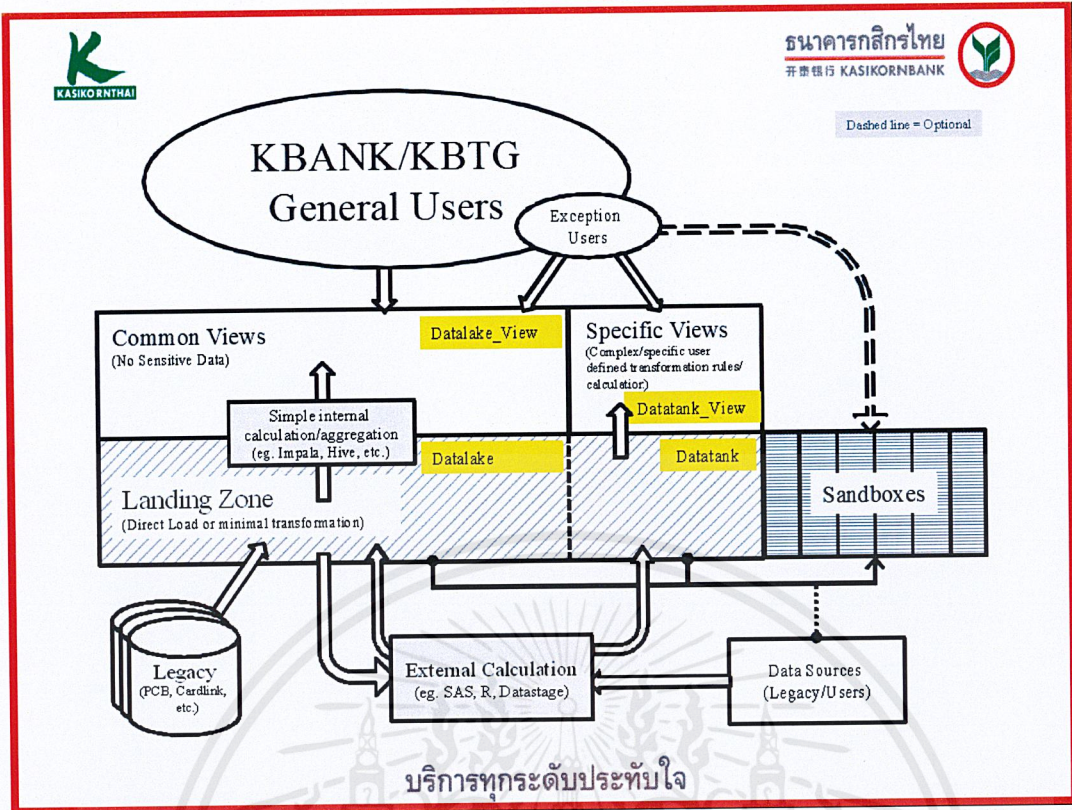
1) Datalake เป็นฐานข้อมูลที่เก็บข้อมูลจากแหล่งกำเนิด (Source) เช่น ข้อมูลลูกค้า เลขบัตรประชาชนของลูกค้า และเบอร์โทรศัพท์ เป็นต้น และเก็บข้อมูลแบบ Sensitive Data นั่นคือเป็นข้อมูลส่วนบุคคลที่เป็นความลับ เช่น ที่อยู่ เลขบัตรประชาชน เบอร์โทรศัพท์ และ วันเกิด เป็นต้น

2) Datalake_view เป็นฐานข้อมูลที่เก็บข้อมูลแบบ Non-Sensitive Data นั่นคือเป็นข้อมูลที่ไม่ใช่ข้อมูลส่วนบุคคลที่เป็นความลับ

3) Datatank เป็นฐานข้อมูลแบบ Specific Transformation คือการนำข้อมูลมาแปลงเป็นเฉพาะเรื่อง และเก็บข้อมูลแบบ Sensitive Data

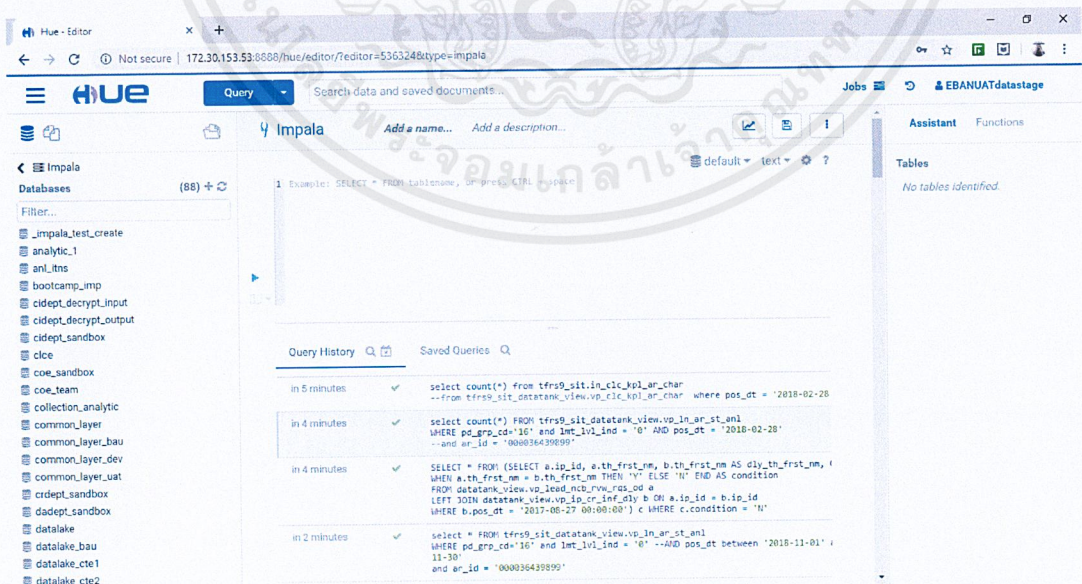
4) Datatank_view เป็นฐานข้อมูลที่เก็บข้อมูลแบบ Sensitive Data บุคคลทั่วไปไม่มีสิทธิ์เข้าไปใช้ฐานข้อมูลได้ ยกเว้นสำหรับผู้ที่มีสิทธิ์เท่านั้น

5) Common_layer เป็นฐานข้อมูลแบบ Data Transformation ทุกคนมีสิทธิ์เข้าไปใช้ได้



ภาพที่ 3.1 ข้อมูลขนาดใหญ่ขั้นพื้นฐาน (Standard Big Data) ของธนาคารกสิกรไทย

เมื่อต้องการดูข้อมูลหรือจัดการข้อมูล จะต้องเข้าไปทำการจัดการข้อมูลภายในเว็บไซต์ฐานข้อมูลของบริษัทได้ (ในที่นี้จะเรียกว่า HUE) โดยเขียน Query เป็นภาษา SQL เพื่อเรียกใช้ข้อมูลได้ตามต้องการ ดังภาพที่ 3.2



ภาพที่ 3.2 เว็บไซต์ฐานข้อมูลของบริษัท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การศึกษาความต้องการของผู้ใช้ (Requirement)

การทำ Lead Generation ในแต่ละผลิตภัณฑ์ได้นั้น จะต้องมีการเก็บ Requirement จาก User ก่อน เพื่อให้ทราบว่า User ต้องการเงื่อนไขหรือกฎอะไรบ้าง เป็นการหาข้อสรุปความต้องการของผู้ใช้นั้นเอง หากลูกค้าที่ไม่ผ่านกฎข้อใดข้อหนึ่งที่ User ได้ตั้งไว้ ก็จะมีการกรองออกไป ดังนั้น ในแต่ละผลิตภัณฑ์อาจมีลูกค้าที่ไม่เหมือนกัน เนื่องจากมี Requirement ที่ต่างกันนั่นเอง

A	B	C	D	E	F	G	L	M	
No.	Credit Criteria	Comment	Diff KCL McK	Check by DL	ผู้เกี่ยวข้อง	KCL	McK(OD)	Captured in PD Mod	com
1	อุตสาหกรรม			DL, RDM		เป็นอุตสาหกรรมดี, อุตสาหกรรมไม่เป็นที่ปอง, อุตสาหกรรม, เกษตรกรรายย่อย, อุตสาหกรรม, FL ไม่สนับสนุน	Y(Normal industry,null->Pass;seise Fail)		
2	FICO		D	DL, MLS		DL: FICO >= 600, 0 MLS: FICO >= 600, 0	RDM: FICO >= 650;Pass MLS: FICO >= 650;Pass		
3	Credit Status	ตรวจสอบข้อมูลเฉพาะผู้เท่านั้น		DL, RDM	Y	DL, RDM: Credit Status เป็น Good ที่ปกติจัดชั้น 1 ในระบบ LPM ธนาคาร (ผู้และผู้ที่เกี่ยวข้อง)	N	Y	Captured
4	Bad Bank			DL		DL: ไม่อยู่ในสังกัดดูแลของ Bad Bank	N	Y	Captured i pick, Crec over Bac
5	NPL	ตรวจสอบข้อมูลเฉพาะผู้เท่านั้น		DL, RDM, MLS	Y	MLS: ไม่เป็น NPL ในรอบ 12 เดือนล่าสุด ใน NCB (ผู้, ผู้เกี่ยวข้อง) DL, RDM: ไม่เป็น NPL ในธนาคาร ในรอบ 12 เดือนล่าสุด (ผู้, ผู้เกี่ยวข้อง)	RDM/MLS: ไม่เป็น NPL ในรอบ 12 เดือนล่าสุด ใน NCB (ผู้)		

ภาพที่ 3.3 การเก็บ Requirement จาก User

เมื่อได้ Requirement แล้ว ก็ทำการศึกษาภาษา SQL โดยการฝึกตั้งข้อมูลตาม Requirement ที่ได้มาโดยใช้ภาษา SQL เพื่อให้เข้าใจกฎต่าง ๆ ได้มากขึ้น และเตรียมตัวในการทำหน้าที่เป็น Developer ดังภาพที่ 3.4

J	K	L	O	P
Physical Table				
Table Name	Column Name	Data Type	Transformation Rules	
DATATANK_P_LEAD_SME	NOT_DEATH_F	STRING	ตรวจสอบเงื่อนไขลูกค้าที่ไม่เสียชีวิต 1. Use table LEAD_SME as base input 2. lookup IP_ID from DATATANK.P_IP_CR_INF_DLY filter DEATH_F = 'Y' and pos_dt = (orderdate) use table 1 left join table 2 by use TO_IP_ID = IP_ID if found set to 'Y'	SELECT ip_id, CASE WHEN a.DEATH_F = 'Y' THEN 'N' ELSE 'Y' END AS NOT_DEATH_F FROM Datatank.p_ip_cr_inf_dly a LEFT JOIN Datatank.p_lead_sme b ON a.ip_id = b.to_ip_id WHERE a.pos_dt=2017-12-31'
DATATANK_P_LEAD_SME	NOT_KCOM_F	STRING	เช็คลูกค้าไม่ใช่งานบริษัทในเครือองค์กรไหน 1. Use table LEAD_SME as base input 2. lookup IP_ID from DATATANK.P_IP_CR_INF_DLY filter KCOM_CD is not null and pos_dt = (orderdate) use table 1 left join table 2 by use TO_IP_ID = IP_ID if found set to 'Y'	SELECT ip_id, CASE WHEN a.KCOM_CD IS NOT NULL THEN 'N' ELSE 'Y' END AS NOT_KCOM_F FROM Datatank.p_ip_cr_inf_dly a LEFT JOIN Datatank.p_lead_sme b ON a.ip_id = b.to_ip_id WHERE a.pos_dt=2017-12-31'
DATATANK_P_LEAD_SME	NOT_IDY_RSK_IND_F	STRING	ตรวจสอบ Risk Industry ของลูกค้า 1. Use table LEAD_SME as base input 2. lookup IP_ID from DATATANK.P_IP_CR_INF_DLY filter SEG_IDY_RSK_IND <> 'Normal' and pos_dt = (orderdate) use table 1 left join table 2 by use TO_IP_ID = IP_ID if found set to 'Y'	P1 SELECT ip_id, CASE WHEN a.SEG_IDY_RSK_IND <> 'Normal' THEN 'N' ELSE 'Y' END AS NOT_IDY_RSK_IND_F FROM Datatank.p_ip_cr_inf_dly a

ภาพที่ 3.4 ตั้งข้อมูลตาม Requirement ที่ได้มาโดยใช้ภาษา SQL

3.3 การออกแบบและวิเคราะห์ข้อมูล (Design and Analysis)

การออกแบบและการวิเคราะห์ข้อมูล เป็นหน้าที่ของนักวิเคราะห์ธุรกิจ หรือที่เรียกว่า BA (Business Analyst) คือคนที่ทำหน้าที่ในการวิเคราะห์ข้อมูลขององค์กรและการดำเนินธุรกิจ ทั้งข้อมูลจริงและสมมติฐาน แล้วนำข้อมูลที่ได้ไปสู่ระบบหรือกระบวนการแก้ไขปัญหาต่อไป โดยขั้นตอนนี้ จะต้องทำการออกแบบเอกสาร 2 ประเภท คือ เอกสาร CBR (Common Business Requirement) และเอกสาร KBMF (K-Bank Message Format)

3.3.1 เอกสาร CBR (Common Business Requirement)

เอกสาร CBR (Common Business Requirement) เป็นข้อมูลที่ทำให้การส่งให้กับผู้พัฒนา (Developer) ทำการพัฒนา ระบบ เพื่อให้ Developer รู้ได้ว่าต้องดึงข้อมูลอะไรบ้าง และดึงข้อมูลมาจากแหล่งที่มาใด ด้านบนสุดของเอกสาร จะเป็นข้อมูลชื่อตารางที่จะให้ Developer ทำการพัฒนา จะเป็นส่วนที่อธิบายว่า จะต้องสร้างตารางไว้ในฐานข้อมูลใด ดังภาพที่ 3.5

Source Name			Table Schema	DATATANK
Table (IN)	DATATANK_IN_LEAD_SME_DLY	(KUDU table)	Table Name	LEAD_SME_DLY
Physical Table (OUT)	DATATANK_P_LEAD_SME_DLY	(Table Path) Shared:EBAN\DATA\LEAD\OUT\IP	Source App Acronym	LEAD
View Name	VP_LEAD_SME_DLY		Data Level	IP
Partition Key	PTN_YYYY.PTN_MM		View Schema	DATATANK_VIEW
Format file type	HDFS			
Ingestion Method	Load Append			

Seq.	Schema	Column Name	Data Type	Size	Mandatory	Transformation Rules	Table Name
------	--------	-------------	-----------	------	-----------	----------------------	------------

ภาพที่ 3.5 ส่วนบนของเอกสาร CBR

เมื่อเลื่อนไปดูเอกสารทางด้านขวามือ จะเจอคอลัมน์ที่ชื่อว่า Source จะเป็นส่วนที่อธิบายว่า ชื่อคอลัมน์นั้น ๆ มาจากที่ใด มาจากฐานข้อมูลไหนนั่นเอง ดังภาพที่ 3.6

Transformation Rules	Source App	Table / File Name	Column Name	Data Type	Size	Mandatory
Processing Logic : Delete from DATATANK.IN_LEAD_SME_DLY Insert into DATATANK.VP_IP_CR_INF_MO (TO_IP_ID, SPCL_PRJ_CD, INL_ID, INCM_AR_ID, INCM_AMT, LEAD_INL_DT, CR_F)						
Processing Logic : Delete from DATATANK.VP_IP_CR_INF_MO (TO_IP_ID, SPCL_PRJ_CD, INL_ID, INCM_AR_ID, INCM_AMT, LEAD_INL_DT, CR_F)						
Processing Logic : Delete from DATATANK.VP_IP_CR_INF_MO (TO_IP_ID, SPCL_PRJ_CD, INL_ID, INCM_AR_ID, INCM_AMT, LEAD_INL_DT, CR_F)	EBAN	DATATANK_VIEW.VP_LEAD_INL	TO_IP_ID	INT		M
Processing Logic : Delete from DATATANK.VP_IP_CR_INF_MO (TO_IP_ID, SPCL_PRJ_CD, INL_ID, INCM_AR_ID, INCM_AMT, LEAD_INL_DT, CR_F)	EBAN	DATATANK_VIEW.VP_LEAD_INL	SPCL_PRJ_CD	STRING		M
Processing Logic : Delete from DATATANK.VP_IP_CR_INF_MO (TO_IP_ID, SPCL_PRJ_CD, INL_ID, INCM_AR_ID, INCM_AMT, LEAD_INL_DT, CR_F)	EBAN	DATATANK_VIEW.VP_IP_CR_INF_MO	POS_DT	TIMESTAMP		M

ภาพที่ 3.6 คอลัมน์ Source ของเอกสาร CBR

เมื่อเลื่อนไปทางซ้ายมือ จะพบคอลัมน์ Physical Table จะเป็นส่วนที่บอกว่าในฐานข้อมูล (ในที่นี้คือ EBAN เป็นชื่อฐานข้อมูลของบริษัท) มีข้อมูลอะไรบ้าง ดังภาพที่ 3.7

Transformation Rules	Table Name	Column Name	Data Type	Size	Mandatory	Transformation Rules	Source App
Processing Logic : 1. Delete from DATATANK.IN_LEAD_SME_DLY 2. Insert into (TO_IP_ID, SPCL_PRJ_CD, INL_ID, INCM_AR_ID, INCM_AMT, LEAD_INL_DT, CR_F) select TO_IP_ID, SPCL_PRJ_CD, INL_ID, INCM_AR_ID, INCM_AMT, LEAD_INL_DT, CR_F							
Direct Move	DATATANK.P_LEAD_SME_DLY	POS_DT	TIMESTAMP		M(PK)	refer to Processing Logic	EBAN
Direct Move	DATATANK.P_LEAD_SME_DLY	TO_IP_ID	INT		M(PK)	refer to Processing Logic	EBAN
Direct Move	DATATANK.P_LEAD_SME_DLY	SPCL_PRJ_CD	STRING		M(PK)	refer to Processing Logic	EBAN
Direct Move	DATATANK.P_LEAD_SME_DLY	CFT_MO_DT	TIMESTAMP		M	update IN_LEAD_SME_DLY set CFT_MO_DT = select max(POS_DT) from DATATANK_VIEW.VP_IP_CR_INF_MO	EBAN

ภาพที่ 3.7 คอลัมน์ Physical Table ของเอกสาร CBR

ด้านซ้ายมือสุดของเอกสาร จะพบคอลัมน์ View จะเป็นส่วนที่บอก Developer ว่าต้องดึงข้อมูลชื่ออะไรบ้าง ดังภาพที่ 3.8

Seq.	Schema	Confid ent	Column Name	Data Type	Size	Mandat ory	Transformation Rules	Table Name
1	DATATANK_VIEW	No	POS_DT	TIMESTAMP	0	M(PK)	Direct Move	DATATANK_P_LEAD_SME_DLY
2	DATATANK_VIEW	No	TO_IP_ID	INT	0	M(PK)	Direct Move	DATATANK_P_LEAD_SME_DLY
3	DATATANK_VIEW	No	SPCL_PRJ_CD	STRING	0	M(PK)	Direct Move	DATATANK_P_LEAD_SME_DLY
4	DATATANK_VIEW	No	CFT_MO_DT	TIMESTAMP	0	M	Direct Move	DATATANK_P_LEAD_SME_DLY

ภาพที่ 3.8 คอลัมน์ View ของเอกสาร CBR

3.3.2 เอกสาร KBMF (K-Bank Message Format)

เอกสาร KBMF (K-Bank Message Format) เป็นเอกสารให้ผู้ใช้ข้อมูลสามารถเข้าใจข้อมูลได้ดียิ่งขึ้น เพื่อให้รู้ว่าข้อมูลนั้น ๆ ย่อมาจากคำว่าอะไร หมายความว่าอย่างไร และตัวอย่างของข้อมูลเป็นอย่างไร เป็นต้น ในส่วนบนของเอกสาร จะเป็นส่วนที่บอกว่า เอกสารนี้จะอธิบายความหมายของข้อมูลจากตารางชื่ออะไร และมาจากฐานข้อมูลไหน เป็นต้น ดังภาพที่ 3.9

Seq. Number	Field Name	Max. Occurrence	Type	Size	Mandatory	Offset	Description	Format
1	Service ID	1	Text	20	M		DATATANK_VIEW_VP_LEAD_SME_DLY	
2	Interface UUID	1	Text	36	M		KBNK-EBAN-718.02-000XXXX	
3	Message Type	1	Text	20	M		Outbound View	
4	Service Description	1	Text	255	M		Lead Generation for KCL Commercial Loan product	
5	Data Frequency	1	Text	10	M		Daily	
6	Data Retention Period	1	Text	10	M		10 Years	
7	Remark	1	Text	255	M		Character Encoding UTF-8	
8							Partition Key PTN_YYYY.PTN_MM	
9							Ingestion Method Load Append	
10							View Type Standard View	
11							View Condition N/A	

ภาพที่ 3.9 ส่วนบนของเอกสาร KBMF

ในส่วนล่าง จะเป็นการอธิบายข้อมูล ว่าชื่อคอลัมน์นั้น ๆ เป็นประเภทอะไร ขนาดเท่าไร สามารถเป็น NULL ได้หรือไม่ ย่อมาจากคำว่าอะไร และตัวอย่างข้อมูลเป็นอย่างไร เป็นต้น ดังภาพที่ 3.10

Seq. Number	Field Name	Max. Occurrence	Type	Size	Mandatory	Offset	Description	Format
1	POS_DT	1	TIMESTAMP	19	M (PK)	19	Def(En): Position Date	YYYY-MM-
2	TO_IP_ID	1	INT	30	M(PK)	49	Def(En): To Customer ID	
3	SPCL_PRJ_CD	1	STRING	10	M(PK)	59	Def(En): Special Project Code	
4	CFT_MO_DT	1	TIMESTAMP	19	M	78	Def(Th): Customer Foundation Monthly Date Def(Th): วันที่ของมูลนิธิจากทางคณะ CFT Monthly	YYYY-MM- HH MM SS
5	INL_ID	1	STRING	30	M	108	Def(En): Initial Id Def(Th)	
6	LEAD_INL_DT	1	TIMESTAMP	19	M	127	Def(En): Lead Initial Date Def(Th)	YYYY-MM- HH:MM:SS
7	LEAD_CRT_DT	1	TIMESTAMP	19	O	146	Def(En): Lead Create Date Def(Th)	YYYY-MM- HH:MM:SS
8	LEAD_GEN_EXP_DT	1	TIMESTAMP	19	O	165	Def(En): Lead Generation Expiry Date Def(Th)	YYYY-MM- HH:MM:SS
9	LEAD_SPPRS_END_DT	1	TIMESTAMP	19	O	184	Def(En): Lead Suppress End Date Def(Th)	YYYY-MM- HH:MM:SS
10	PD_TP_CD	1	STRING	10	M	194	Def(En): The Product type code Def(Th)	

ภาพที่ 3.10 ส่วนเนื้อหาของเอกสาร KBMF

3.4 การจัดการข้อมูลโดย Datastage

การจัดการข้อมูลโดย Datastage คือการดึงข้อมูลจาก Data Source ต่าง ๆ เข้าสู่ Data Warehouse หน้าที่ของ Datastage คือ

- 1) Extract คือ การสกัดข้อมูล การดึงข้อมูลจากแหล่งข้อมูลที่แตกต่างกัน
- 2) Transform คือ การนำข้อมูลที่ได้มาจากการ Extract มาจัดรูปแบบให้ถูกต้อง
- 3) Load คือ การนำข้อมูลที่ผ่านการ Transform แล้ว เข้าสู่ Data Warehouse หรือ Data Mart หรือ Database

ซึ่งการจัดการข้อมูลโดย Datastage นี้ เป็นหน้าที่ของ Developer ที่จะทำการดึงข้อมูลจาก Data Source ต่าง ๆ เข้าสู่ EBAN ซึ่งก่อนที่จะทำการพัฒนาใน Datastage จะต้องสร้างไฟล์ที่มีนามสกุลไฟล์คือ ddl หรือที่เรียกว่า Script ขึ้นมา ทำการสร้างพื้นที่เพื่อจัดเก็บไฟล์ข้อมูลของ โดยการสร้างไฟล์ ddl นั้น หลัก ๆ แล้วมีด้วยกัน 3 ไฟล์ คือ

- 1) ขา IN เป็นการเก็บข้อมูลจาก Source File จะสามารถดูได้จากเอกสาร CBR ในส่วนคอลัมน์ Source โดยจะไม่มีกรทำ Partition ที่ขา IN โดย Partition จะเป็นส่วนที่บอกว่าข้อมูลที่ Developer ต้องทำการพัฒนาอยู่ระดับไหน เป็นระดับปี ระดับเดือน หรือระดับวัน

```

1 SET SYNC_DDL=1;
2 DROP TABLE IF EXISTS DATATANK.IN_LEAD_CFG_SELF_EMP_INL_HIST;
3 CREATE EXTERNAL TABLE DATATANK.IN_LEAD_CFG_SELF_EMP_INL_HIST (
4   POS_DT          TIMESTAMP
5   ,PD_TP_CD       STRING
6   ,SPCL_PRJ_CD   STRING
7   ,MN_FNL_PD     DECIMAL(10,6)
8   ,MX_FNL_PD     DECIMAL(10,6)
9   ,OL_SELLR      DECIMAL(9,5)
10  ,OL_SELLR_WEB  DECIMAL(9,5)
11  ,OL_SELLR_KMKT DECIMAL(9,5)
12  ,OL_SELLR_EV   DECIMAL(9,5)
13  ,PTNL_OL_SELLR DECIMAL(9,5)
14  ,OL_SELLR_INFLOW DECIMAL(9,5)
15  ,OL_SELLR_TAX  DECIMAL(9,5)
16  ,FRLN          DECIMAL(9,5)
17  ,ORG_SEG       DECIMAL(9,5)
18  ,DUAL_SEG      DECIMAL(9,5)
19  ,IDV_SME       DECIMAL(9,5)
20  ,BSN_OWN       DECIMAL(9,5)
21  ,EDC_PD_HLDG  DECIMAL(9,5)

```

ภาพที่ 3.11 ไฟล์ ddl สำหรับขา IN

2. ขา OUT หรือขา P เป็นการดึงข้อมูลจากขา IN เป็นการบีบขนาดไฟล์ให้เล็กลง ซึ่งไฟล์นี้ใช้ได้แค่ Hadoop เท่านั้น ขา OUT จะแตกต่างจากขา IN ตรงที่มีการทำ Partition โดยสามารถดูได้ที่เอกสาร CBR ในคอลัมน์ Physical Table

```

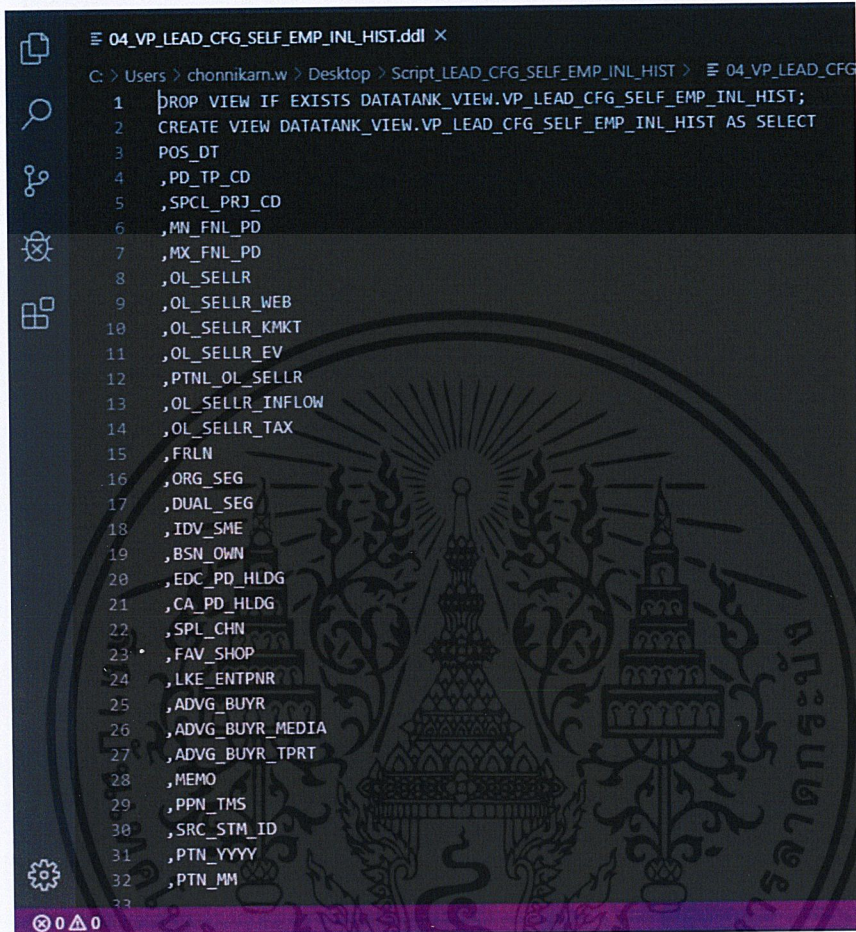
File Edit Selection View Go Debug Terminal Help
03_P_LEAD_CFG_SELF_EMP_INL_HIST.ddl x
C:\Users\chonnikarnw\Desktop>Script_LEAD_CFG_SELF_EMP_INL_HIST> 03_P_LEAD_CFG_SELF_EMP_INL_HIST.ddl
1 SET SYNC_DDL=1;
2 DROP TABLE IF EXISTS DATATANK.P_LEAD_CFG_SELF_EMP_INL_HIST;
3 REATE EXTERNAL TABLE DATATANK.P_LEAD_CFG_SELF_EMP_INL_HIST (
4   OS_DT          TIMESTAMP
5   ,PD_TP_CD       STRING
6   ,SPCL_PRJ_CD   STRING
7   ,MN_FNL_PD     DECIMAL(10,6)
8   ,MX_FNL_PD     DECIMAL(10,6)
9   ,OL_SELLR      DECIMAL(9,5)
10  ,OL_SELLR_WEB  DECIMAL(9,5)
11  ,OL_SELLR_KMKT DECIMAL(9,5)
12  ,OL_SELLR_EV   DECIMAL(9,5)
13  ,PTNL_OL_SELLR DECIMAL(9,5)
14  ,OL_SELLR_INFLOW DECIMAL(9,5)
15  ,OL_SELLR_TAX  DECIMAL(9,5)
16  ,FRLN          DECIMAL(9,5)
17  ,ORG_SEG       DECIMAL(9,5)
18  ,DUAL_SEG      DECIMAL(9,5)
19  ,IDV_SME       DECIMAL(9,5)
20  ,BSN_OWN       DECIMAL(9,5)
21  ,EDC_PD_HLDG  DECIMAL(9,5)
22  ,CA_PD_HLDG   DECIMAL(9,5)
23  ,SPL_CHN      DECIMAL(9,5)
24  ,FAV_SHOP     DECIMAL(9,5)
25  ,LKE_ENTPNR   DECIMAL(9,5)
26  ,ADVG_BUYR    DECIMAL(9,5)
27  ,ADVG_BUYR_MEDIA DECIMAL(9,5)
28  ,ADVG_BUYR_TPRT DECIMAL(9,5)
29  ,MEMO         DECIMAL(9,5)
30  ,PPN_TMS      TIMESTAMP
31  ,CRG_CTN_CD   STRING
32  PARTITIONED BY (PTN_YYYY STRING,PTN_MM STRING)
33  STORED AS PARQUET

```

ภาพที่ 3.12 ไฟล์ ddl สำหรับขา OUT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

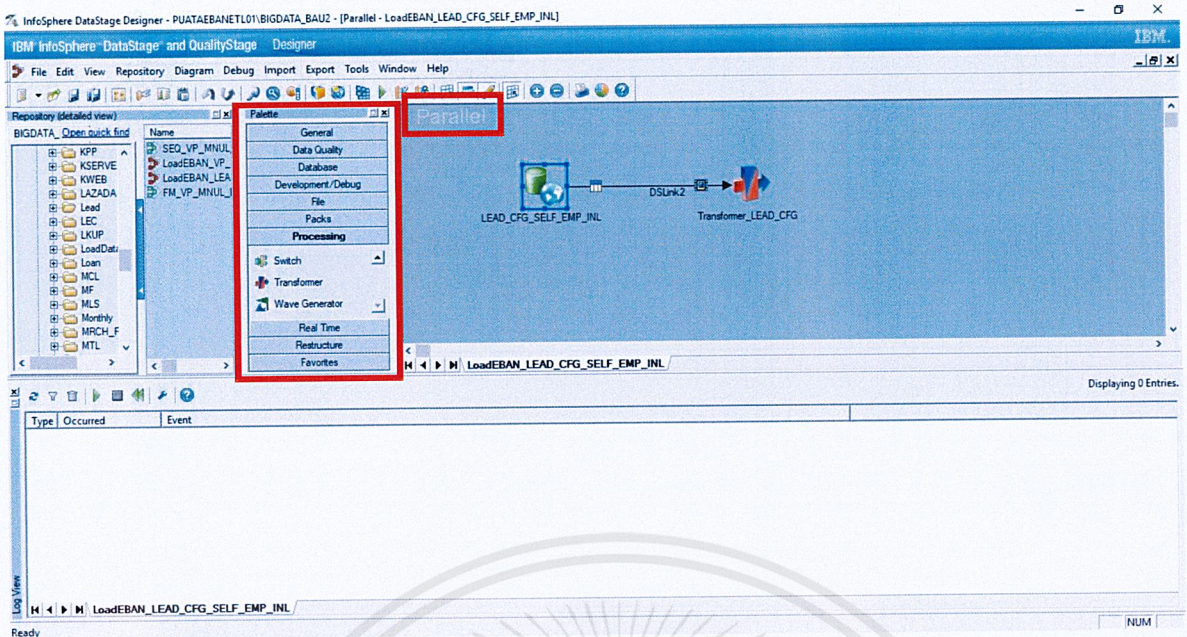
3. ขา V หรือขา VIEW เป็นการดึงข้อมูลจากขา P มา แต่ในบางครั้ง ก็ไม่ได้ดึงข้อมูลเหมือนขา P เสมอไป Developer จะต้องทำตามเอกสาร CBR เป็นหลัก ซึ่งส่วนนี้สามารถดูได้ในคอลัมน์ View ว่าจะดึงข้อมูลอะไรบ้าง



```
04_VP_LEAD_CFG_SELF_EMP_INL_HIST.ddl
C: > Users > chonnikarn.w > Desktop > Script_LEAD_CFG_SELF_EMP_INL_HIST > 04_VP_LEAD_CFG
1 | PROP VIEW IF EXISTS DATATANK_VIEW.VP_LEAD_CFG_SELF_EMP_INL_HIST;
2 | CREATE VIEW DATATANK_VIEW.VP_LEAD_CFG_SELF_EMP_INL_HIST AS SELECT
3 | POS_DT
4 | , PD_TP_CD
5 | , SPCL_PRJ_CD
6 | , MN_FNL_PD
7 | , MX_FNL_PD
8 | , OL_SELLR
9 | , OL_SELLR_WEB
10 | , OL_SELLR_KMKT
11 | , OL_SELLR_EV
12 | , PTNL_OL_SELLR
13 | , OL_SELLR_INFLOW
14 | , OL_SELLR_TAX
15 | , FRLN
16 | , ORG_SEG
17 | , DUAL_SEG
18 | , IDV_SME
19 | , BSN_OWN
20 | , EDC_PD_HLDG
21 | , CA_PD_HLDG
22 | , SPL_CHN
23 | , FAV_SHOP
24 | , LKE_ENTPNR
25 | , ADVG_BUYR
26 | , ADVG_BUYR_MEDIA
27 | , ADVG_BUYR_TPRT
28 | , MEMO
29 | , PPN_TMS
30 | , SRC_STM_ID
31 | , PTN_YYYY
32 | , PTN_MM
33
```

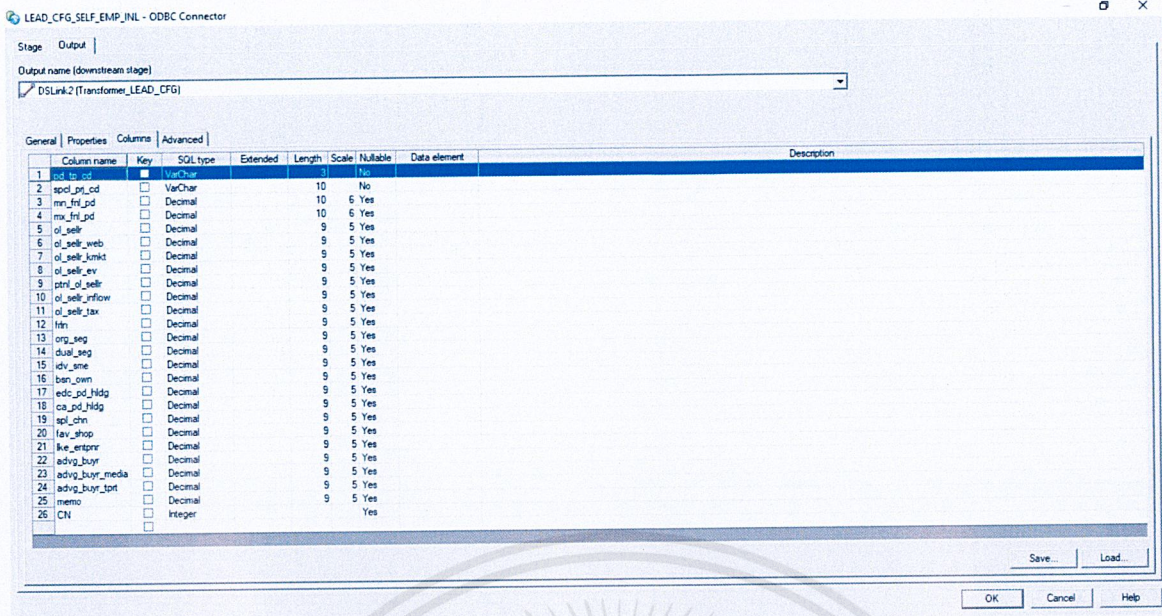
ภาพที่ 3.13 ไฟล์ ddl สำหรับขา VIEW

เมื่อสร้าง Script เสร็จแล้ว ก็นำไปประมวลผลในเว็บไซต์ฐานข้อมูลของบริษัท เมื่อประมวลผลผ่าน จะมีพื้นที่จัดเก็บข้อมูลของในเว็บไซต์ฐานข้อมูลของบริษัท จากนั้นก็สามารถทำ Datastage ได้ โดยการเริ่มสร้างไฟล์ใน Datastage ซึ่งไฟล์ใน Datastage นั้น จะมีหน่วยย่อยหรือเล็กสุด เรียกว่า Parallel และสามารถเลือกเครื่องมือใช้งานได้ตามต้องการที่ Palette ทางด้านซ้ายมือ สามารถเรียกใช้โดยการลากแล้ววางได้ ดังภาพที่ 3.14

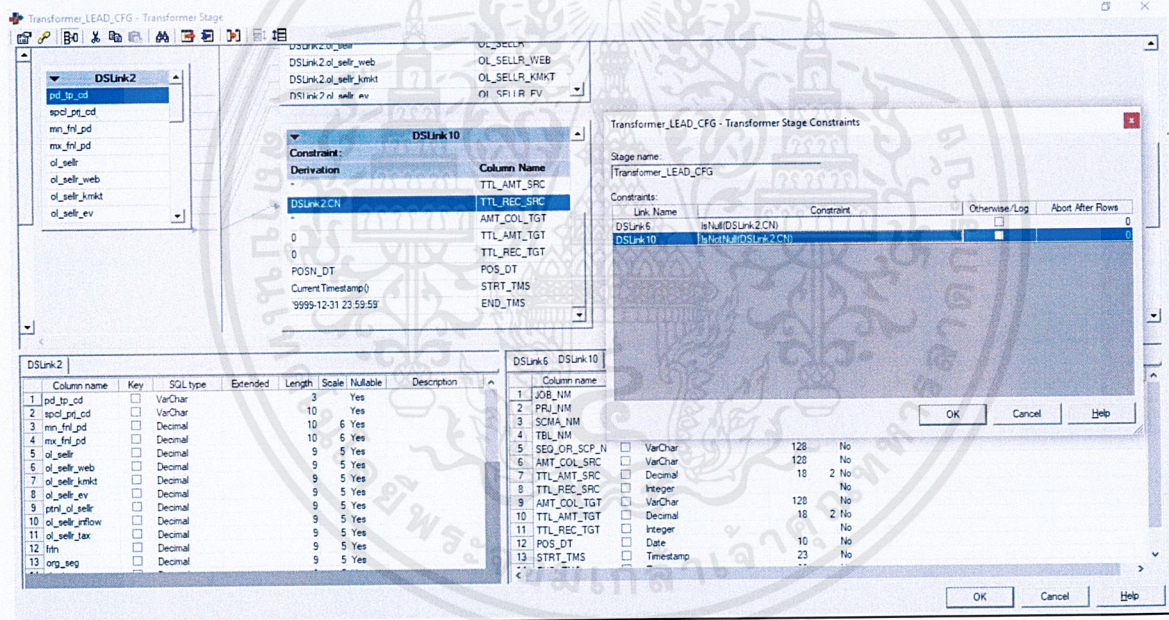


ภาพที่ 3.14 ไฟล์ Parallel ในโปรแกรม Datastage

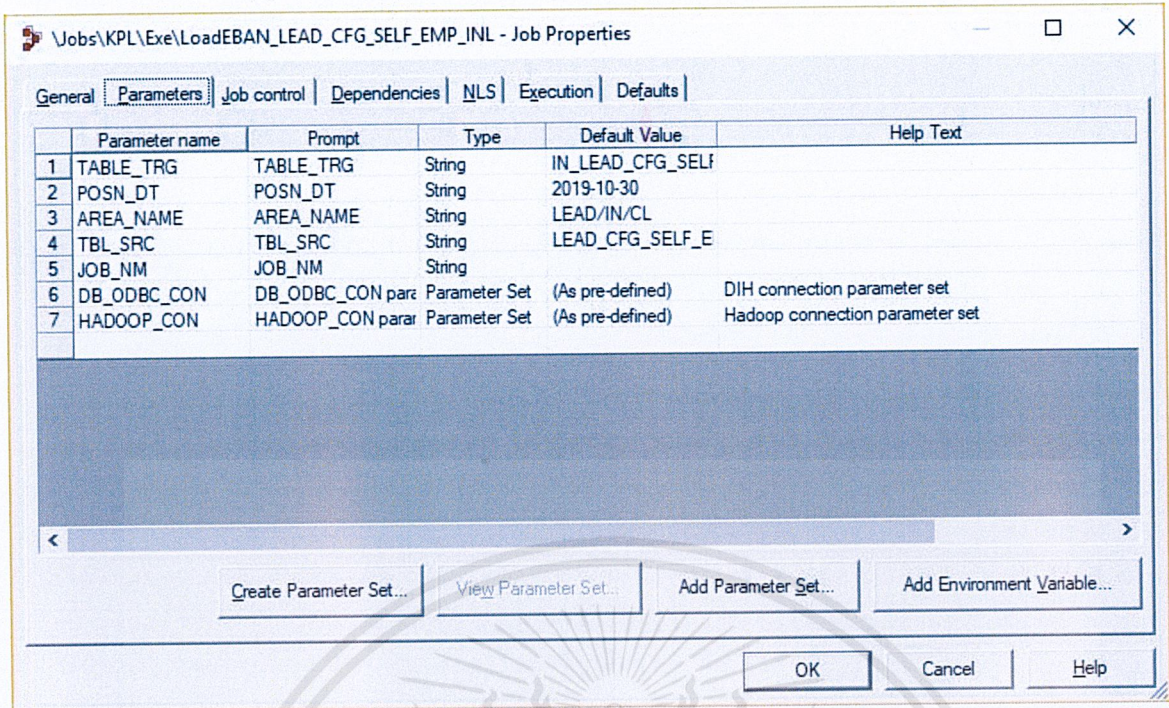
ซึ่งแต่ละเครื่องมือที่จะใช้งาน จะต้องมีการกรอกรายละเอียดทุกครั้ง โดยจะยกตัวอย่างจากงาน เครื่องมือ LEAD_CFG_SELF_EMP_INL จะเลือกใช้เป็น Database เนื่องจาก Source เป็น Database เมื่อทำการ Double Click เข้าไป จะต้องทำการกรอกข้อมูลให้ครบและถูกต้อง เช่น การกรอกข้อมูล คอลัมน์จาก Source จะต้องกรอกให้ถูกต้อง ไม่ว่าจะเป็นประเภทข้อมูล ขนาดข้อมูล และสามารถเป็น NULL ได้หรือไม่ ดังภาพที่ 3.15 และเครื่องมือ Tranformer_LEAD_CFG ที่เลือกมาจาก Palette ทางด้าน ซ้ายมือ จะเป็นตัวที่ทำการนำข้อมูลจาก Source นำเข้าสู่ขา IN โดยจะต้องกรอกรายละเอียดเช่นเดียวกัน กับเครื่องมือ LEAD_CFG_SELF_EMP_INL ดังภาพที่ 3.16 นอกจากการกรอกข้อมูลให้ถูกต้องแล้ว จะต้อง ทำการลาก DSLink ให้ถูกต้องอีกด้วย ดังนั้นเมื่อกดเข้ามาดูข้อมูลในเครื่องมือ Tranformer_LEAD_CFG จะเจอกับการเชื่อมโยงของข้อมูลในแต่ละ DSLink ว่ามีการเชื่อมโยงข้อมูลอะไรบ้าง ประเภทข้อมูลอะไร และขนาดข้อมูลเท่าไร เป็นต้น นอกจากนี้ทุกงานจะต้องมีการกำหนดพารามิเตอร์ที่จะต้องใช้ในงานด้วย ทุกครั้ง ดังภาพที่ 3.17



ภาพที่ 3.15 การกรอกข้อมูลคอลัมน์จาก Source

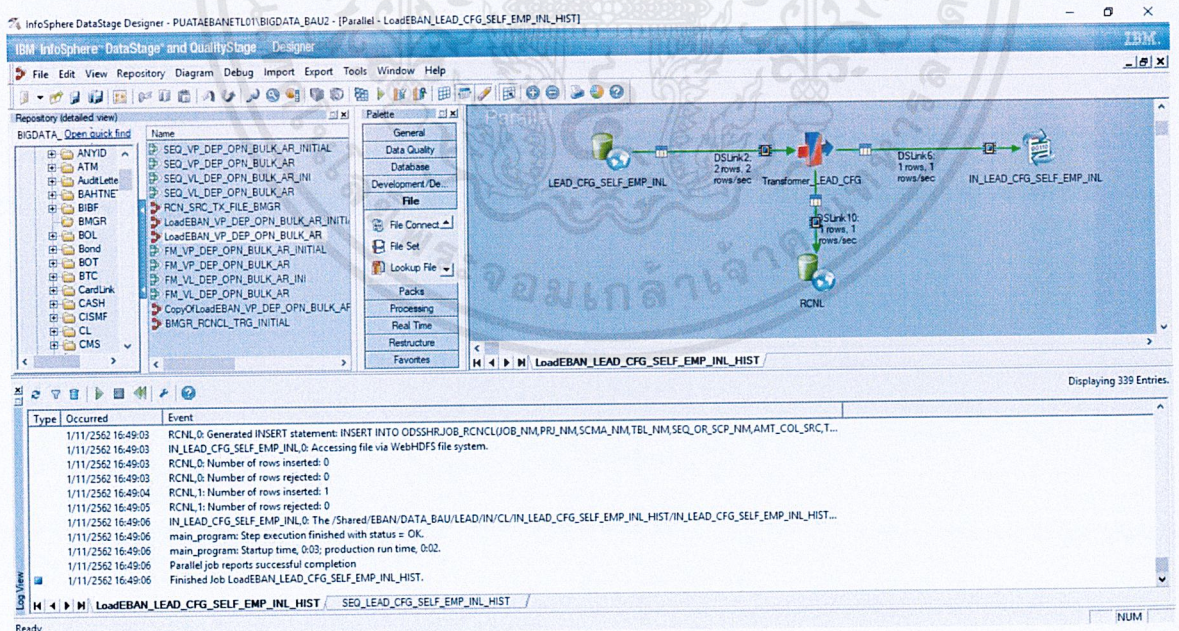


ภาพที่ 3.16 การกรอกข้อมูลในเครื่องมือ Tranformer_LEAD_CFG



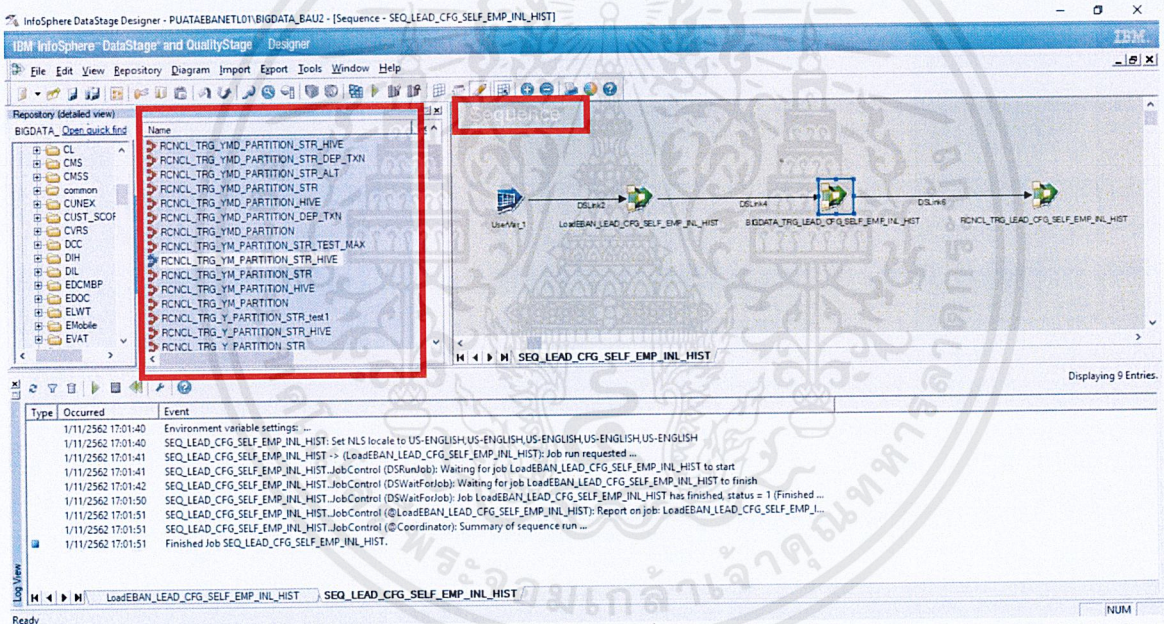
ภาพที่ 3.17 การกำหนดพารามิเตอร์ที่จะต้องใช้ในไฟล์ Parallel

เมื่อทำการเลือกใช้เครื่องมือ และกรอกรายละเอียดทุกอย่างที่ต้องการจนครบแล้ว ให้ทำการประมวลผล ว่าผ่านหรือไม่ หากประมวลผลแล้วเกิดความผิดพลาด จะต้องแก้ไขให้ถูกต้องให้ไม่มีความผิดพลาดใด ๆ เมื่อประมวลผลผ่านแล้ว ดังภาพที่ 3.18

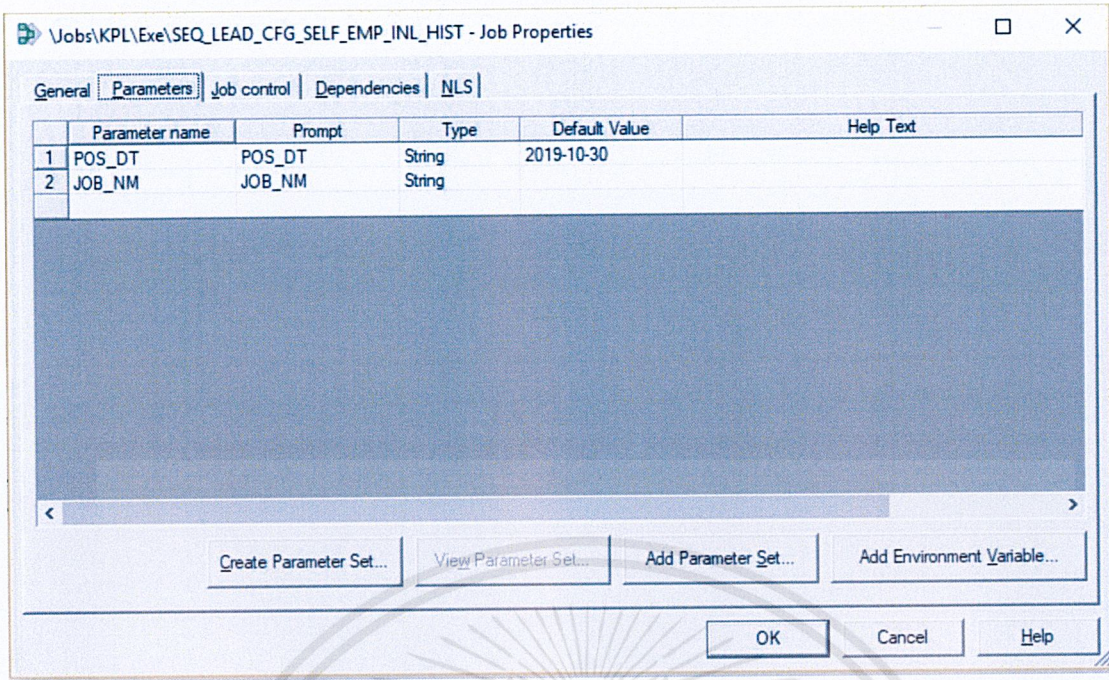


ภาพที่ 3.18 การประมวลผลงานในโปรแกรม Datastage

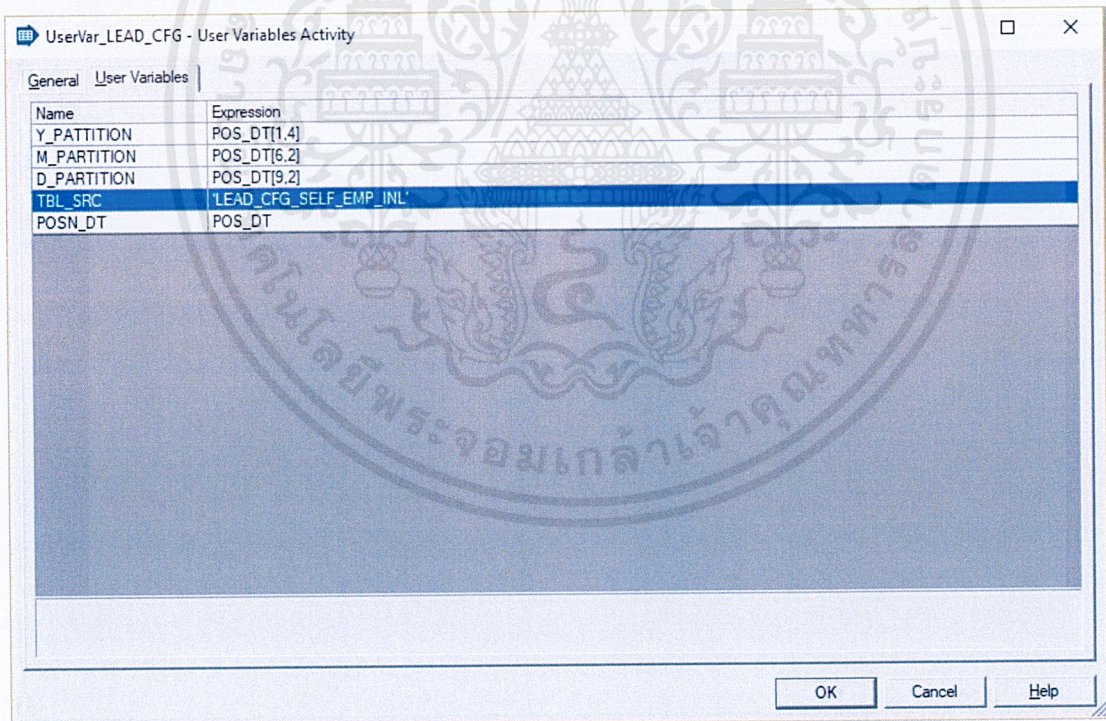
เมื่อทำในส่วนของ Parallel สำเร็จแล้ว ทำการสร้างไฟล์ใหม่ขึ้นมา เรียกว่า Sequence เป็นส่วนที่ใหญ่กว่า Parallel หรือเป็นส่วนที่ครอบ Parallel ไว้อีกทีนั่นเอง โดยทำการลากงาน Parallel ที่ได้ทำไว้มาวางที่ Sequence จะเห็นได้จากที่มีชื่องาน LoadEBAN_LEAD_CFG_SELF_EMP_INL_HIST ในไฟล์ Sequence จากนั้นก็สามารถใช้งานพารามิเตอร์ทั่วไป ที่ทางบริษัทได้สร้างไว้ให้แล้วได้ตามต้องการ โดยการลากและวางเช่นกัน เช่น RCNCL_TRG_YM_PARTITION_STR_HIVE เป็นตัวที่ทำหน้าที่ในการนับข้อมูลจาก Source และนับข้อมูลที่จะเข้ามา IN เพื่อตรวจสอบว่าได้นำข้อมูลเข้าเท่ากันหรือไม่นั่นเอง ดังภาพที่ 3.19 ทุกงานจะต้องมีการกำหนดพารามิเตอร์ที่จะต้องใช้งานด้วยทุกครั้ง ดังภาพที่ 3.20 มีการกำหนดค่าให้กับทุกเครื่องมือในไฟล์ Sequence นั่นคือทุกเครื่องมือสามารถใช้ค่าเหล่านี้ได้ โดยการเรียกใช้ชื่อตัวแปรนั้น ๆ ตัวอย่างการกำหนดค่าให้กับทุกเครื่องมือในไฟล์ Sequence จะเห็นได้ดังภาพที่ 3.21 เช่นเดียวกันกับงานใน Parallel แต่ละเครื่องมือที่ใช้งาน จะต้องมีการกรอกรายละเอียดให้ครบและถูกต้องทุกครั้ง ดังภาพที่ 3.22 ภาพที่ 3.23 และภาพที่ 3.24 จากนั้นทำการประมวลผลทำงานผ่านหรือไม่



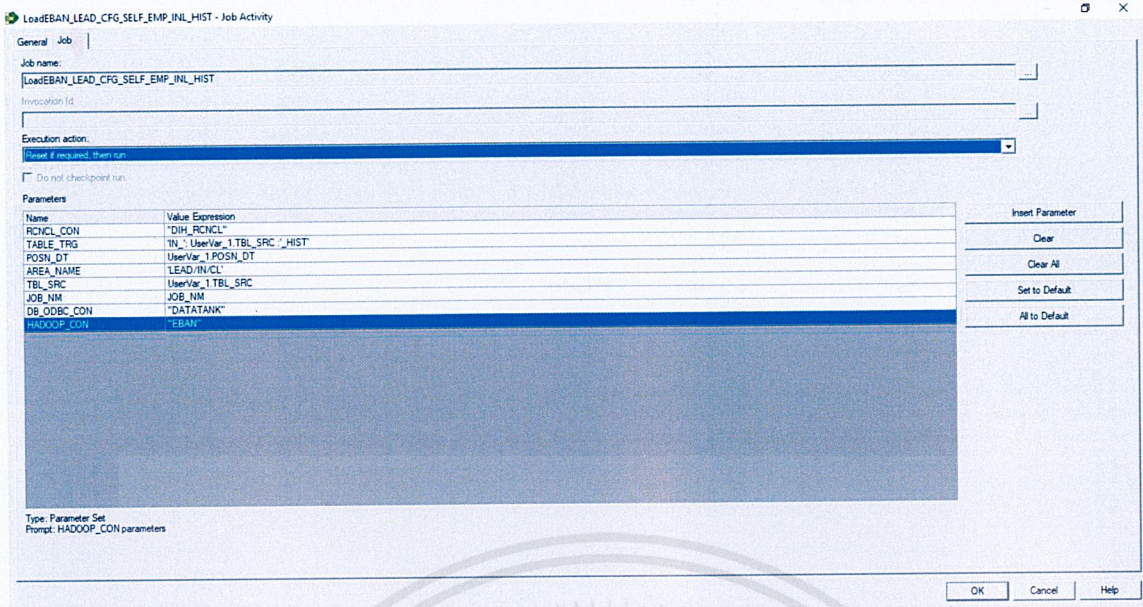
ภาพที่ 3.19 พารามิเตอร์ทั่วไปของบริษัท



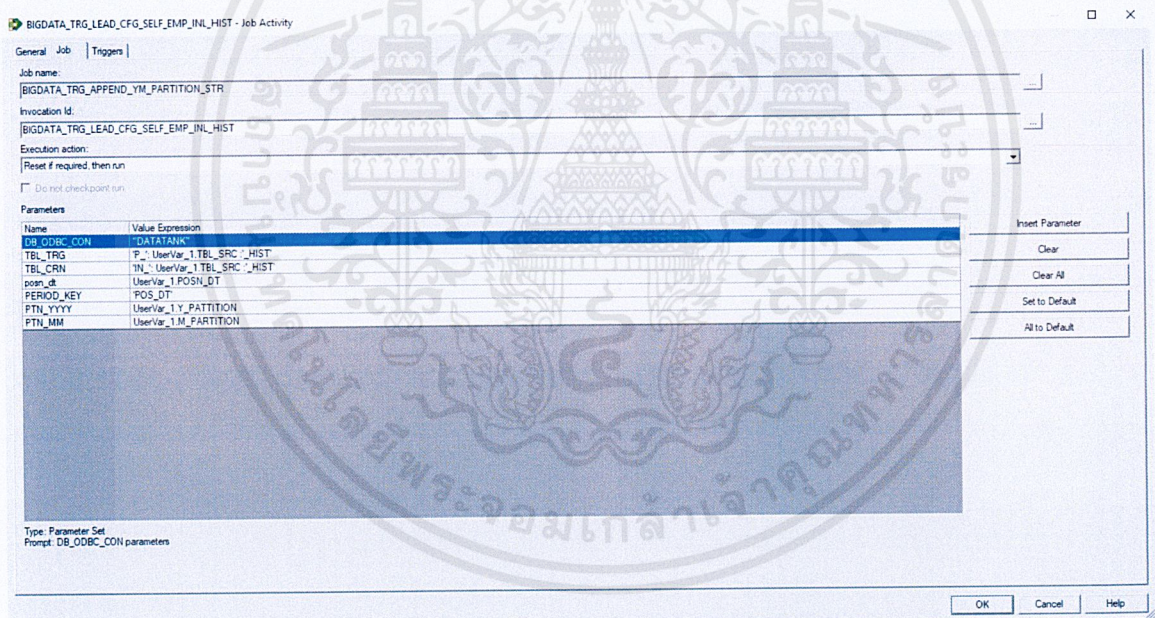
ภาพที่ 3.20 การกำหนดพารามิเตอร์ที่จะต้องใช้ในไฟล์ Sequence



ภาพที่ 3.21 การกำหนดค่าให้กับทุกเครื่องมือในไฟล์ Sequence



ภาพที่ 3.22 การกรอกข้อมูลในเครื่องมือ LoadEBAN_ LEAD_CFG_SELF_EMP_INL_HIST

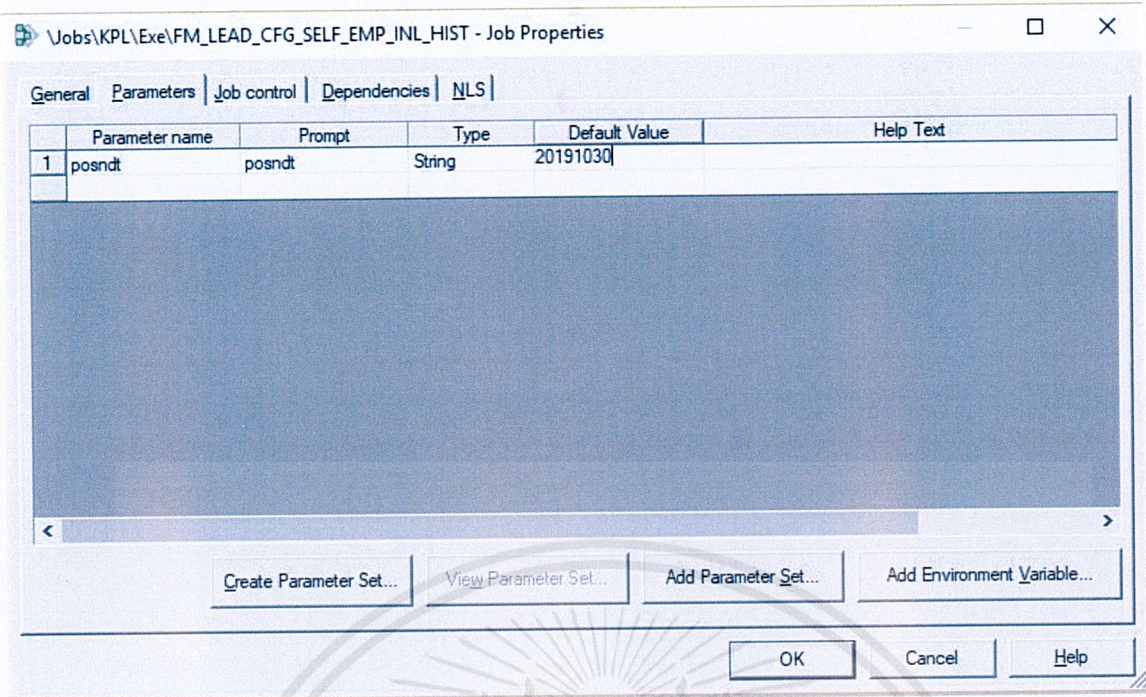


ภาพที่ 3.23 การกรอกข้อมูลในเครื่องมือ BIGDATA_TRG_ LEAD_CFG_SELF_EMP_INL_HIST

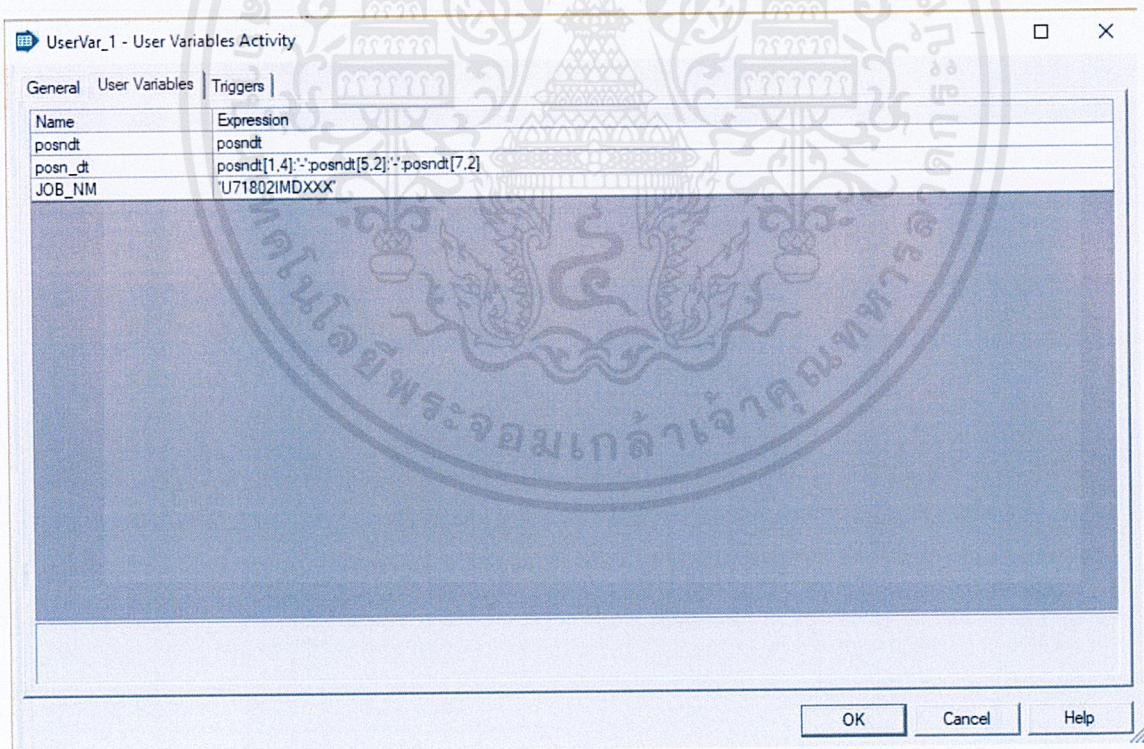


ภาพที่ 3.24 การกรอกข้อมูลในเครื่องมือ RCNCL_TRG_LEAD_CFG_SELF_EMP_INL_HIST

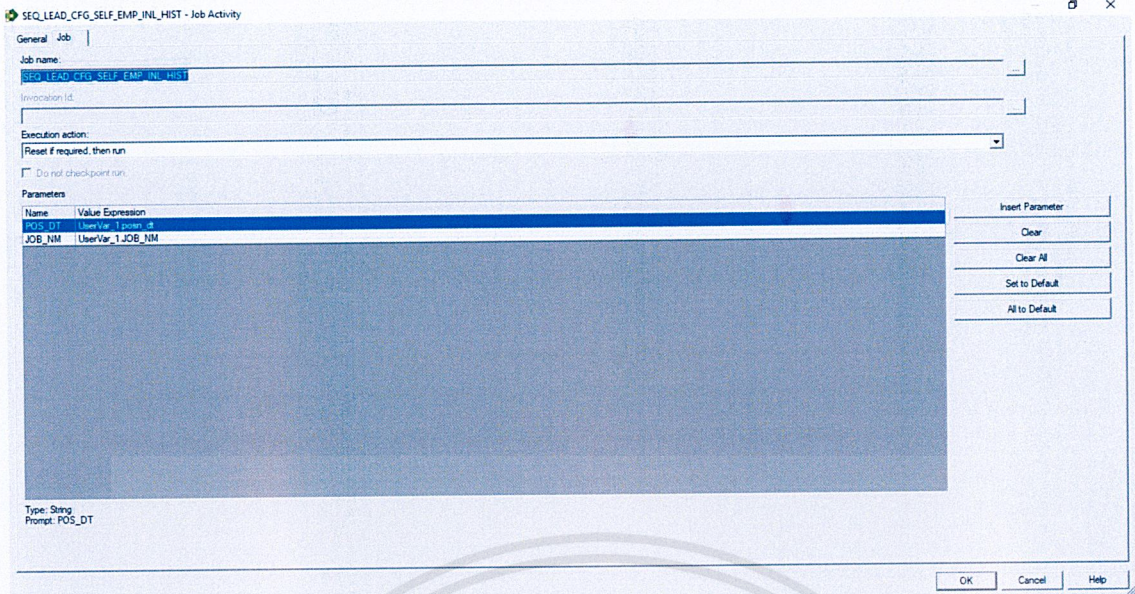
เมื่อประมวลผลเสร็จ ไม่มีข้อผิดพลาดอะไร ก็ทำการสร้าง Sequence เพิ่ม เพื่อทำการครอบงาน Sequence เดิมอีกทีหนึ่ง Sequence ใหม่จะเรียกว่า FM จะเห็นได้จากไฟล์งานที่มีชื่อว่า FM_LEAD_CFG_SELF_EMP_INL_HIST นั่นเอง เช่นเดียวกันทุกงานจะต้องมีการกำหนดพารามิเตอร์ที่จะต้องใช้ในการงานด้วยทุกครั้ง ดังภาพที่ 3.25 มีการกำหนดค่าให้กับทุกเครื่องมือในไฟล์งาน FM ดังภาพที่ 3.26 และจะต้องมีการกรอกรายละเอียดครบและถูกต้องให้เครื่องมือทุกครั้ง ดังภาพที่ 3.27 เมื่อทำการสร้าง FM เสร็จ จากนั้นทำการประมวลผล เมื่อไม่มีข้อผิดพลาดใด ๆ ก็ถือว่างานนี้สำเร็จ ดังภาพที่ 3.28 สามารถนำงานไปใช้ได้จริง หรือนำไป Deploy ได้



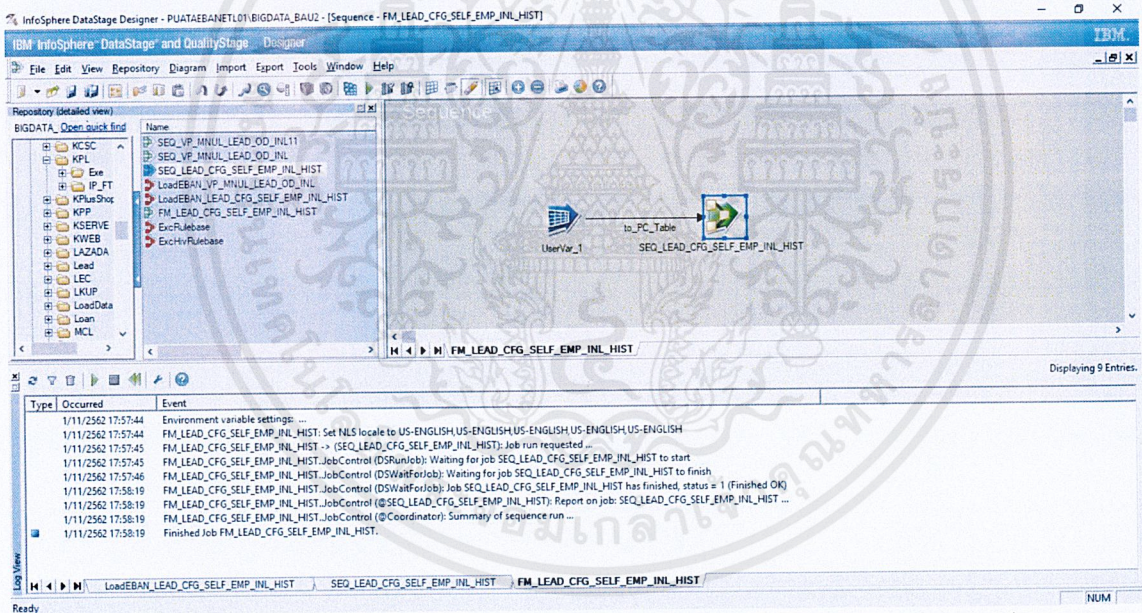
ภาพที่ 3.25 การกำหนดพารามิเตอร์ที่จะต้องใช้ในงาน FM



ภาพที่ 3.26 การกำหนดค่าให้กับทุกเครื่องมือในไฟล์งาน FM



ภาพที่ 3.27 การกรอกข้อมูลในเครื่องมือ SEQ_LEAD_CFG_SELF_EMP_INL_HIST



ภาพที่ 3.28 การประมวลผลงาน FM_LEAD_CFG_SELF_EMP_INL_HIST สำเร็จ

3.5 การทดสอบความถูกต้องโดยเอกสาร UIS

หลังจากที่ทำการจัดการข้อมูลโดย Datastage สำเร็จแล้ว ต้องทำการทดสอบความถูกต้องทั้งหมดของงาน โดยเรียกเอกสารที่ทำการทดสอบนี้ว่า เอกสาร UIS เป็นเอกสารที่ Developer ต้องทำการทดสอบงานด้วยตนเอง โดยหน้าที่หลักของเอกสาร UIS นั้น คือการอธิบายว่างานที่จะทำการทดสอบนี้ ชื่องานว่าอะไร ชื่องานในโปรแกรม Datastage มีชื่อว่าอะไรบ้าง หน้าตาของงานเป็นอย่างไร ใช้เครื่องมืออะไรบ้าง และตัวแปรที่ Developer ได้ใส่ในงานมีอะไรบ้าง เป็นต้น ดังภาพที่ 3.29

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Technical sheet	LEAD_CFG_SELF_EMP_INL_HIST											
2	Environment												
3	Server :	PUATAEBANETL01											
4	Project :	BIGDATA_EXT											
5	Category :	JOB\KPL\Ext\LEAD_CFG_TBU\											
6	Job Name :	FM_LEAD_CFG_SELF_EMP_INL_HIST											
7		SEQ_LEAD_CFG_SELF_EMP_INL_HIST											
8		LOB\EBAN_LEAD_CFG_SELF_EMP_INL_HIST											
9													
10	Source Table :	SRV\EP.S6.EDH.S61D011.SAM											
11	Target Table :	P_LEAD_CFG_SELF_EMP_INL_HIST											
12													
13													
14	Common Job :	BIGDATA_TRG_APPEND_YM_PARTITION_STR											
15		RCNCL_TRG_YM_PARTITION_STR_HIVE											
16													
17													
18	Main Sequencer	FM_LEAD_CFG_SELF_EMP_INL_HIST											
19													
20													
21													
22													
23													
24													
25													
26													
27													
28													
29													
30													
31													
32													
33													
34													
35													
36													
37													
38													
39													
40													
41													
42													
43													
44													
45													

Name	Expression
param8	param8
param_8	param8[1.4] ; param8[5.2] ; param8[7.2]
JOB_NM	U718021MDXXX

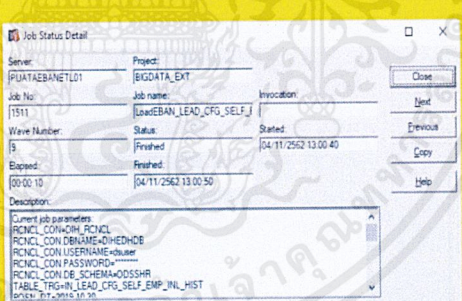
ภาพที่ 3.29 เอกสาร UIS ส่วนแรก

ภายในเอกสารยังมีส่วนที่บอกว่า ตัวอย่างข้อมูลของขา IN ขา OUT และขา V เป็นอย่างไร เหมือนกันหรือไม่ เพื่อทดสอบความถูกต้อง ดังภาพที่ 3.30

B36	R	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
5	Schema :			Schema :	DATATANK											
6	Source table :			Target table :	P_LEAD_CFG_SELF_EMP_INL_HIST											
7																
8																
9	Position Date = '2010-10-30'			Position Date = '2010-10-30'												
10																
12	Actual Result															
13	DATATANK.P_LEAD_CFG_SELF_EMP_INL_HIST															
14	2010-10-30 00:00:00	Success	620	0.94567	0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
15																
16																
17																
18																
19																
20	Actual Result															
21	DATATANK.P_LEAD_CFG_SELF_EMP_INL_HIST															
22	2010-10-30 00:00:00	Success	620	0.94567	0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
23																
24																
25																
26	Expected Result															
27	DATATANK.VIEW.VP_LEAD_CFG_SELF_EMP_INL_HIST															
28	2010-10-30 00:00:00	Success	620	0.94567	0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
29																
30																
31																
32	Verify Result															
33	DATATANK.P_LEAD_CFG_SELF_EMP_INL_HIST															
34	2010-10-30 00:00:00	Success	620	0.94567	0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
35																
36																
37																
38																
39																
40	Verify Result															
41	DATATANK.VIEW.VP_LEAD_CFG_SELF_EMP_INL_HIST															
42	2010-10-30 00:00:00	Success	620	0.94567	0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
43																

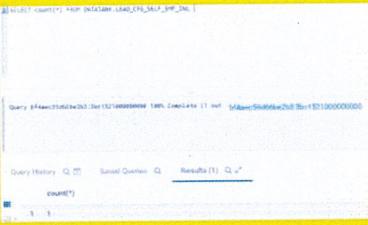
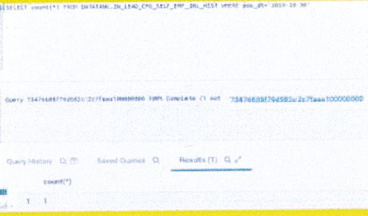
ภาพที่ 3.30 ตัวอย่างการเทียบข้อมูลในเอกสาร UIS

ภายในเอกสารมีส่วนที่ใช้โปรแกรม Director Client เพื่อทดสอบงานทุกหน่วย ไม่ว่าจะเป็นงาน Parallel หรืองาน Sequence โปรแกรมนี้จะทำการทดสอบงานว่าสำเร็จหรือไม่ เมื่องานสำเร็จ ก็สามารถถ่ายภาพหน้าจอ แล้วนำมาวางในเอกสาร UIS ดังภาพที่ 3.31

A	B	C	D	E	F	G	
9	Acknowledgment: Describe any restrictions, for example, the user has to have appropriate access authority.						
10	Pre-Navigation	1. Run load process completed	PASS				
11	Post-Navigation	Describe any steps required to exit the test conditions area and to reset the test environment for another test.					
12	Test Step	Action	Expected Result	Actual Result		Pass or Fail	
13							
14	1	Compile and Run Job LoadEBAN_LEAD_CFG_SELF_EMP_INL_HIS	Job run completely	PASS		PASS	

ภาพที่ 3.31 ตรวจสอบงานด้วยโปรแกรม Director Client ในเอกสาร UIS

และยังมีส่วนที่ Developer ต้องทำการนับจำนวนข้อมูลจาก Source จำนวนข้อมูลขา IN และจำนวนข้อมูลขา OUT ว่ามีจำนวนข้อมูลเท่ากันหรือไม่ โดยทำการ Query เรียกดูจำนวนข้อมูลโดยใช้ภาษา SQL ในเว็บไซต์ฐานข้อมูลของบริษัท หรือในโปรแกรม DbVisualizer หากมีจำนวนเท่ากันจะถือว่าถูกต้อง ดังภาพที่ 3.32

A	B	C	D	E	F
14	1			Pass	
15	2			Pass	

ภาพที่ 3.32 ตรวจสอบจำนวนข้อมูลด้วยโปรแกรม DbVisualizer ในเอกสาร UIS

นอกจากนี้ยังมีส่วนที่บอกว่าข้อมูลที่ Developer นำมาจากเอกสาร CBR มีข้อมูลอะไรบ้าง ชนิดข้อมูลเป็นอย่างไรบ้าง เมื่อทำการเทียบกับข้อมูลที่ผ่านการจัดการโดย Datastage แล้ว ข้อมูลเหมือนกันหรือไม่ ทำการตรวจสอบข้อมูลทั้งขา OUT และขา V หากข้อมูลเหมือนกันถือว่างานถูกต้อง ดังภาพที่ 3.33 เมื่อทำการทดสอบทั้งหมดสำเร็จ ก็แสดงว่างานที่ Developer ทำนั้นถูกต้อง แต่หากทำการตรวจสอบแล้วพบข้อผิดพลาด Developer ต้องทำการแก้ไขให้การทดสอบถูกต้องให้ได้

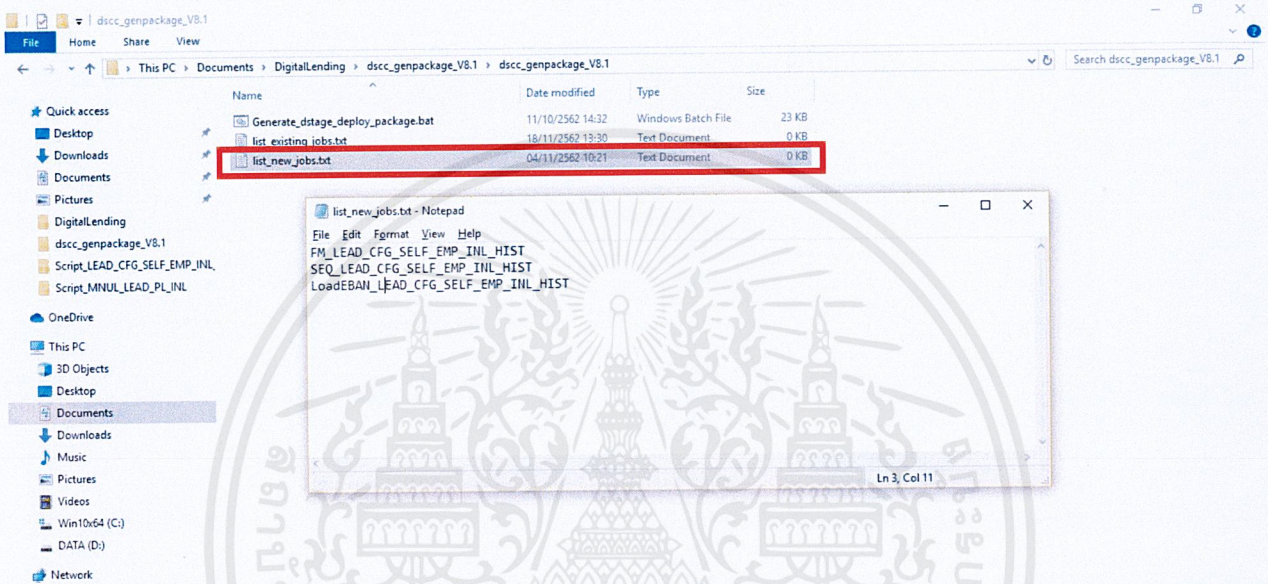
	A	B	C	D	E	F	G	H	I	J	K	L
1	Check Field name & Data type											
2												
3												
4												
5												
6		SB		datatank		Physical (datatank.p_lead_ctg_self_emp_inl_hist)						
7		POS_DT	TIMESTAMP			pos_dt	timestamp					1 TRUE
8		PD_TP_CD	STRING			pd_tp_cd	string					1 TRUE
9		SPCL_PRJ_CD	STRING			spd_prj_cd	string					1 TRUE
10		MN_FNL_PD	DECIMAL(10,0)			mn_fnl_pd	decimal(10,0)					1 TRUE
11		MX_FNL_PD	DECIMAL(10,0)			mx_fnl_pd	decimal(10,0)					1 TRUE
12		OL_SELLR	DECIMAL(9,5)			ol_sellr	decimal(9,5)					1 TRUE
13		OL_SELLR_WEB	DECIMAL(9,5)			ol_sellr_web	decimal(9,5)					1 TRUE
14		OL_SELLR_KMKT	DECIMAL(9,5)			ol_sellr_kmkt	decimal(9,5)					1 TRUE
15		OL_SELLR_EV	DECIMAL(9,5)			ol_sellr_ev	decimal(9,5)					1 TRUE
16		PTNL_OL_SELLR	DECIMAL(9,5)			ptnl_ol_sellr	decimal(9,5)					1 TRUE
17		OL_SELLR_INFLOW	DECIMAL(9,5)			ol_sellr_inflow	decimal(9,5)					1 TRUE
18		OL_SELLR_TAX	DECIMAL(9,5)			ol_sellr_tax	decimal(9,5)					1 TRUE
19		FRIN	DECIMAL(9,5)			frin	decimal(9,5)					1 TRUE
20		ORG_SEG	DECIMAL(9,5)			org_seg	decimal(9,5)					1 TRUE
21		DUAL_SEG	DECIMAL(9,5)			dual_seg	decimal(9,5)					1 TRUE
22		IDV_SME	DECIMAL(9,5)			idv_sme	decimal(9,5)					1 TRUE
23		BSN_OWN	DECIMAL(9,5)			bsn_own	decimal(9,5)					1 TRUE
24		EDC_PD_HLDG	DECIMAL(9,5)			edc_pd_hldg	decimal(9,5)					1 TRUE
25		CA_PD_HLDG	DECIMAL(9,5)			ca_pd_hldg	decimal(9,5)					1 TRUE
26		SPL_CHN	DECIMAL(9,5)			spl_chn	decimal(9,5)					1 TRUE
27		FAV_SHOP	DECIMAL(9,5)			fav_shop	decimal(9,5)					1 TRUE
28		LKE_ENTPNR	DECIMAL(9,5)			lke_entpnr	decimal(9,5)					1 TRUE
29		ADVG_BUYR	DECIMAL(9,5)			advb_buyr	decimal(9,5)					1 TRUE
30		ADVG_BUYR_MEDIA	DECIMAL(9,5)			advb_buyr_media	decimal(9,5)					1 TRUE
31		ADVG_BUYR_TPRT	DECIMAL(9,5)			advb_buyr_tprt	decimal(9,5)					1 TRUE
32		MEMO	DECIMAL(9,5)			memo	decimal(9,5)					1 TRUE
33		PPN_TMS	TIMESTAMP			ppn_tms	timestamp					1 TRUE
34		SRC_STM_ID	STRING			src_stm_id	string					1 TRUE
35		PTN_YYYY	STRING			ptn_yyy	string					1 TRUE
36		PTN_MM	STRING			ptn_mm	string					1 TRUE
37												
38												

ภาพที่ 3.33 การเทียบข้อมูลจากเอกสาร CBR กับข้อมูลจาก Datastage ในเอกสาร UIS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

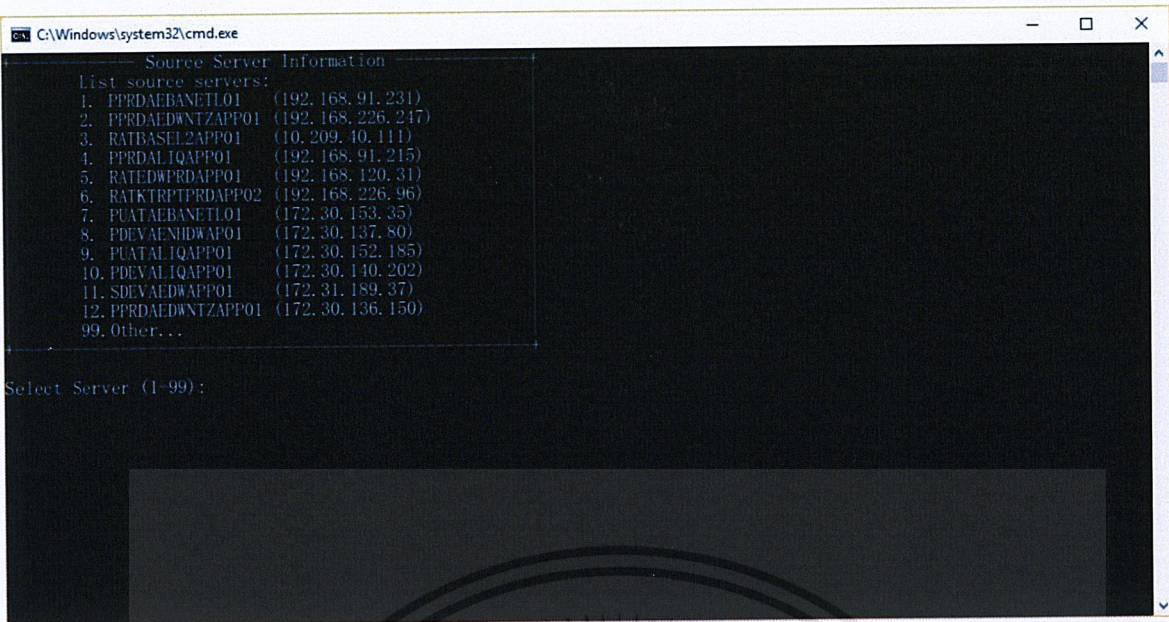
3.6 การนำงานไปใช้จริงบน Server (Deploy)

ก่อนที่จะทำการ Deploy จะต้องมีการ Pack Code งานทั้งหมด เริ่มจากงานใน Datastage โดยให้เข้าไปในไฟล์ dscg_genpackage_V8.1 เมื่อเข้าไปแล้วให้กรอกชื่องาน Datastage ว่ามีงานอะไรบ้าง หากงานที่จะทำการ Deploy เป็นการสร้างตารางใหม่ในฐานข้อมูลให้กรอกชื่องานใน list_new_jobs.txt ดังภาพที่ 3.34 แต่ถ้าหากงานที่ได้รับ เป็นการแก้ไขตารางข้อมูลเดิม ให้กรอกชื่องานใน list_existing_jobs.txt

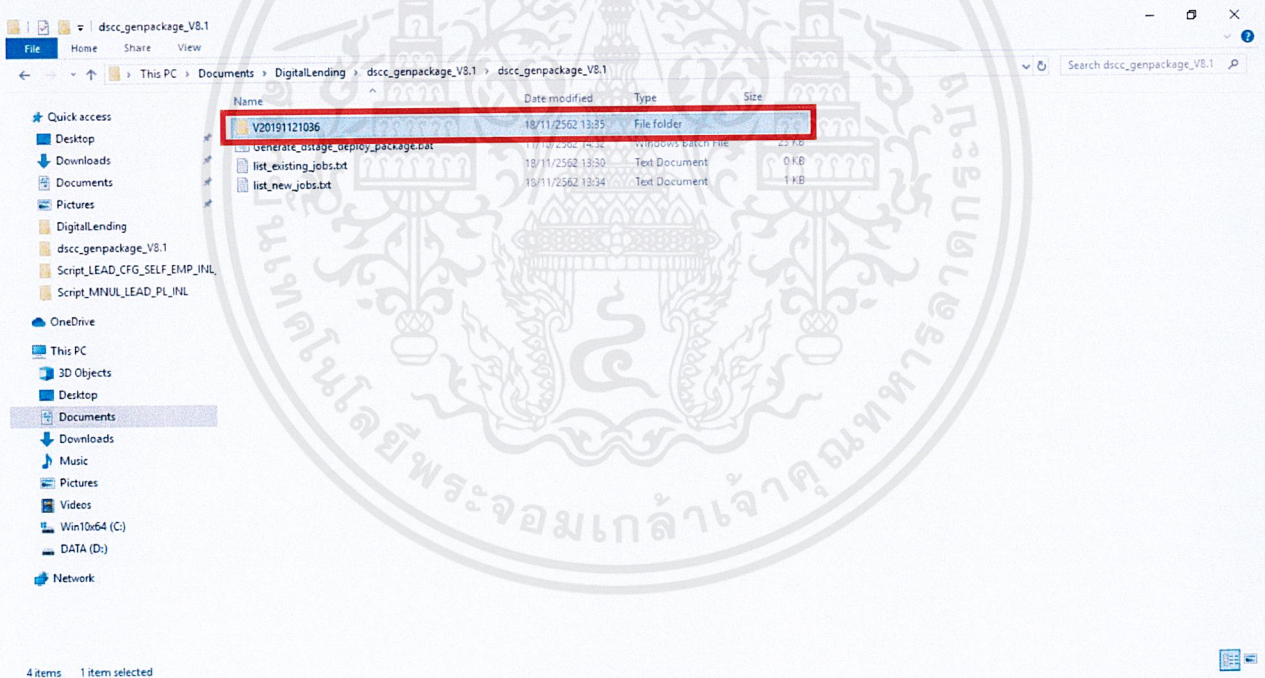


ภาพที่ 3.34 กรอกชื่องาน Datastage ที่จะทำการ Deploy เพื่อสร้างตารางใหม่ในฐานข้อมูล

จากนั้นกด Generate_dstage_deploy_package.bat จะมีหน้าจอเหมือนดังภาพที่ 3.35 จากนั้นทำการกรอกข้อมูลตามที่โปรแกรมต้องการ เมื่อกรอกครบแล้ว จะมีไฟล์ใหม่เกิดขึ้นมา ดังภาพที่ 3.36

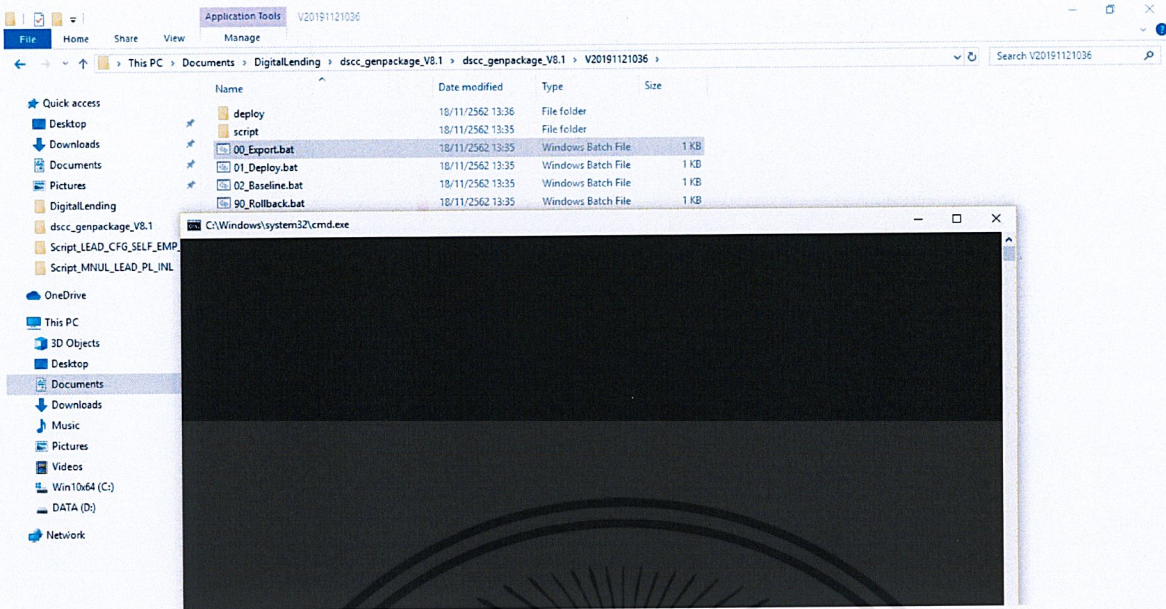


ภาพที่ 3.35 หน้าต่างของ Generate_dstage_deploy_package.bat



ภาพที่ 3.36 ไฟล์ใหม่

เมื่อกดเข้าไปในไฟล์ที่เกิดขึ้นมาใหม่ ให้ทำการกดที่ 00_Export.bat จากนั้นรอการประมวลผล ดังภาพที่ 3.37 เมื่อการประมวลผลเสร็จ จะมีชื่องาน Datastage ปรากฏในโฟลเดอร์ deploy ดังภาพที่ 3.38



ภาพที่ 3.37 รอกการประมวลผล



ภาพที่ 3.38 ชื่องาน Datastage ปรากฏในไฟล์เตอร์ deploy

เมื่อทำการ Pack Code ในส่วนของ Datastage เสร็จแล้ว ให้ทำการ Pack Code ในส่วนของ DDL ซึ่ง DDL ประกอบไปด้วยไฟล์ ดังภาพที่ 3.39



ภาพที่ 3.39 การ Pack Code ในส่วนของ DDL

การสร้างไฟล์ 01 จะคล้ายกับการสร้าง Script ในขั้นตอนของการจัดการข้อมูลโดย Datastage จะเป็นไฟล์ที่ใส่ข้อมูลทั้งขา IN ขา OUT และขา V ตัวอย่างดังภาพที่ 3.40 ทำการสร้างไฟล์ 02 เพื่อเป็นการยินยอมให้สิทธิ์ผู้ใช้งานในการเข้าถึง Path ของขา IN ดังภาพที่ 3.41

```

File Edit Selection View Go Debug Terminal Help
01_DDL_LEAD_CFG_SELF_EMP_INL_HIST.sql X
C: > Users > chonnikarn.w > Desktop > 01_DDL_LEAD_CFG_SELF_EMP_INL_HIST.sql
1 SET SYNC_DDL=1;
2
3 DROP TABLE if exists DATATANK.IN_LEAD_CFG_SELF_EMP_INL_HIST ;
4
5 CREATE EXTERNAL TABLE DATATANK.IN_LEAD_CFG_SELF_EMP_INL_HIST
6 (
7     ,POS_DT          TIMESTAMP
8     ,PD_TP_CD       STRING
9     ,SPCL_PRJ_CD    STRING
10    ,MN_FNL_PD       DECIMAL(10,6)
11    ,MX_FNL_PD       DECIMAL(10,6)
12    ,OL_SELLR        DECIMAL(9,5)
13    ,OL_SELLR_WEB    DECIMAL(9,5)
14    ,OL_SELLR_KHKT   DECIMAL(9,5)
15    ,OL_SELLR_EV     DECIMAL(9,5)
16    ,PTNL_OL_SELLR   DECIMAL(9,5)
17    ,OL_SELLR_INFLOW DECIMAL(9,5)
18    ,OL_SELLR_TAX    DECIMAL(9,5)
19    ,FRLN            DECIMAL(9,5)
20    ,ORG_SEG         DECIMAL(9,5)
21    ,DUAL_SEG        DECIMAL(9,5)
22    ,IDV_SME         DECIMAL(9,5)
23    ,BSN_OWN         DECIMAL(9,5)
24    ,EDC_PD_HLDG    DECIMAL(9,5)
25    ,CA_PD_HLDG     DECIMAL(9,5)
26    ,SPL_CHN         DECIMAL(9,5)
27    ,FAV_SHOP        DECIMAL(9,5)
28    ,LKE_ENTPNR     DECIMAL(9,5)
29    ,ADVG_BUYR       DECIMAL(9,5)
30    ,ADVG_BUYR_MEDIA DECIMAL(9,5)
31    ,ADVG_BUYR_TPRT DECIMAL(9,5)
32    ,MEMO            DECIMAL(9,5)
33    ,PPN_TMS         TIMESTAMP

```

ภาพที่ 3.40 การสร้างไฟล์ 01 สำหรับ Pack Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

File Edit Selection View Go Debug Terminal Help
02_GRANT_DDL.sh - Visual Studio Code
01_DDL_LEAD_CFG_SELF_EMP_INL_HIST.sql 02_GRANT_DDL.sh X
C: > Users > chonnikarn.w > Desktop > 02_GRANT_DDL.sh
1 hadoop fs -chown -R EBANdatastage /Shared/EBAN/DATA/LEAD/IN/CL/IN_LEAD_CFG_SELF_EMP_INL_HIST/

```

ภาพที่ 3.41 การสร้างไฟล์ 02 สำหรับ Pack Code

ทำการสร้างไฟล์ 03 เพื่อเป็นการ Refresh ข้อมูลต่าง ๆ ในฐานข้อมูล เพื่อไม่ให้เกิดปัญหาเรื่องข้อมูลไม่สามารถเข้าฐานข้อมูลได้ ดังภาพที่ 3.42 จากนั้นทำการสร้างไฟล์ 04 เพื่อเป็น Script ในการดำเนินการ Run แบบอัตโนมัติ ดังภาพที่ 3.43

```

File Edit Selection View Go Debug Terminal Help
03_REFRESH.sql - Visual
01_DDL_LEAD_CFG_SELF_EMP_INL_HIST.sql 02_GRANT_DDL.sh 03_REFRESH.sql X
C: > Users > chonnikarn.w > Desktop > 03_REFRESH.sql
1 REFRESH DATATANK.IN_LEAD_CFG_SELF_EMP_INL_HIST ;
2 REFRESH DATATANK.P_LEAD_CFG_SELF_EMP_INL_HIST ;
3 REFRESH DATATANK_VIEW.VP_LEAD_CFG_SELF_EMP_INL_HIST;
4
5 INVALIDATE METADATA DATATANK.IN_LEAD_CFG_SELF_EMP_INL_HIST ;
6 INVALIDATE METADATA DATATANK.P_LEAD_CFG_SELF_EMP_INL_HIST ;

```

ภาพที่ 3.42 การสร้างไฟล์ 03 สำหรับ Pack Code

```

File Edit Selection View Go Debug Terminal Help
deploy.sh - Visual Studio Code
01_DDL_LEAD_CFG_SELF_EMP_INL_HIST.sql 02_GRANT_DDL.sh 03_REFRESH.sql deploy.sh X
C: > Users > chonnikarn.w > Desktop > PackCode_OD+SME(NEW) > V2019115036_MNUL_LEAD_SME_INL_DDL > deploy.sh
1 #!/bin/ksh
2 #####
3 # Script Objective: For automation deploy SQL
4 # Prerequisite:
5 # Usage: sh deploy.sh
6 # Example: sh deploy.sh
7 # Script by: Naiyaporn Smanote
8 # Script Version: 2.1
9 # Edit by: Jatuporn Somsamai
10 # Create Date: 7 Aug 2018
11 # Last Modified: 11 Feb 2019
12 #####
13
14 #Variable
15 SCRIPTPATH=$(pwd)
16 CONFIGPATH=$(cd ${SCRIPTPATH};cd ../../config;pwd;)
17 #Check config file
18 if [ ! -f ${CONFIGPATH}/deploy.cfg ];
19 then
20     echo "Error: Input file does not exist."
21     echo ""
22     echo "Solution:"
23     echo " 1. Check for file "${CONFIGPATH}"/deploy.cfg"
24     echo ""
25     exit 1
26 else
27     . ${CONFIGPATH}/deploy.cfg
28
29 #Clear old file (in case rerun).
30 >${SCRIPTPATH}/run.sh
31 rm -f ${SCRIPTPATH}/ERROR.ERR
32

```

ภาพที่ 3.43 การสร้างไฟล์ 04 สำหรับ Pack Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผ่านขั้นตอนทั้งหมดที่ได้กล่าวข้างต้นแล้ว ก็สามารถนำงานไปใช้จริงบน Server หรือเรียกว่า การ Deploy บน Production โดยการ Deploy งานได้นั้น จะต้องมีพนักงานท่านอื่นที่ทำหน้าที่ Deploy งานในบริษัท มาทำการ Deploy งานของ Developer นั่นคือ Developer ไม่สามารถ Deploy งานของตนเองขึ้น Production ได้เอง หากทำการ Deploy แล้วไม่เกิดข้อผิดพลาดอะไร ถือว่างานสามารถนำไปใช้จริงได้สำเร็จ

3.7 การทดสอบความถูกต้อง

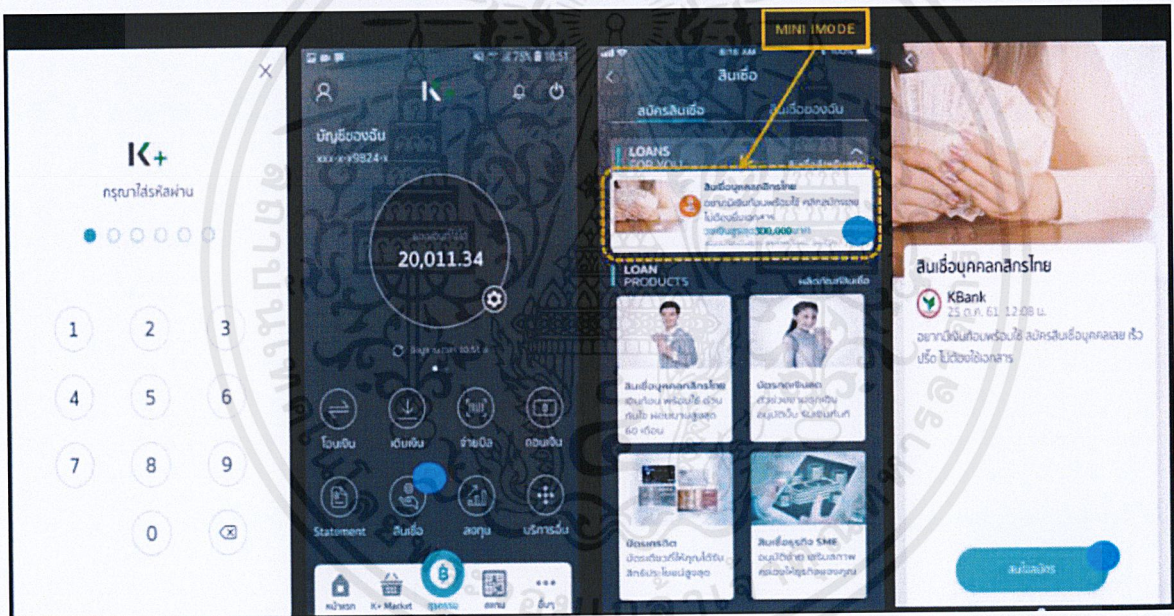
เมื่อทำการ Deploy งานได้สำเร็จแล้ว จะต้องทำการทดสอบความถูกต้องอีกครั้ง โดยการทดสอบความถูกต้องมี 2 ประเภท คือ

- 1) Performance Verification Test (PVT) เป็นการทดสอบระบบอีกครั้งกับ User ว่าตรงตามความต้องการของ User หรือไม่ เช่น หาก User ต้องการทราบว่า มีข้อมูลจำนวนลูกค้าที่ยังไม่เสียชีวิตกี่คน Developer ต้องทำการ Query ข้อมูลให้ได้ เป็นต้น
- 2) Integration Verification Test (IVT) เป็นการทดสอบระบบอีกครั้ง โดย Developer หรือ Business Analyst เป็นผู้ทดสอบระบบด้วยตนเอง เพื่อตรวจดูว่าระบบถูกต้องหรือไม่ โดยไม่ต้องผ่านความเห็นของ User

บทที่ 4 ผลการวิจัย

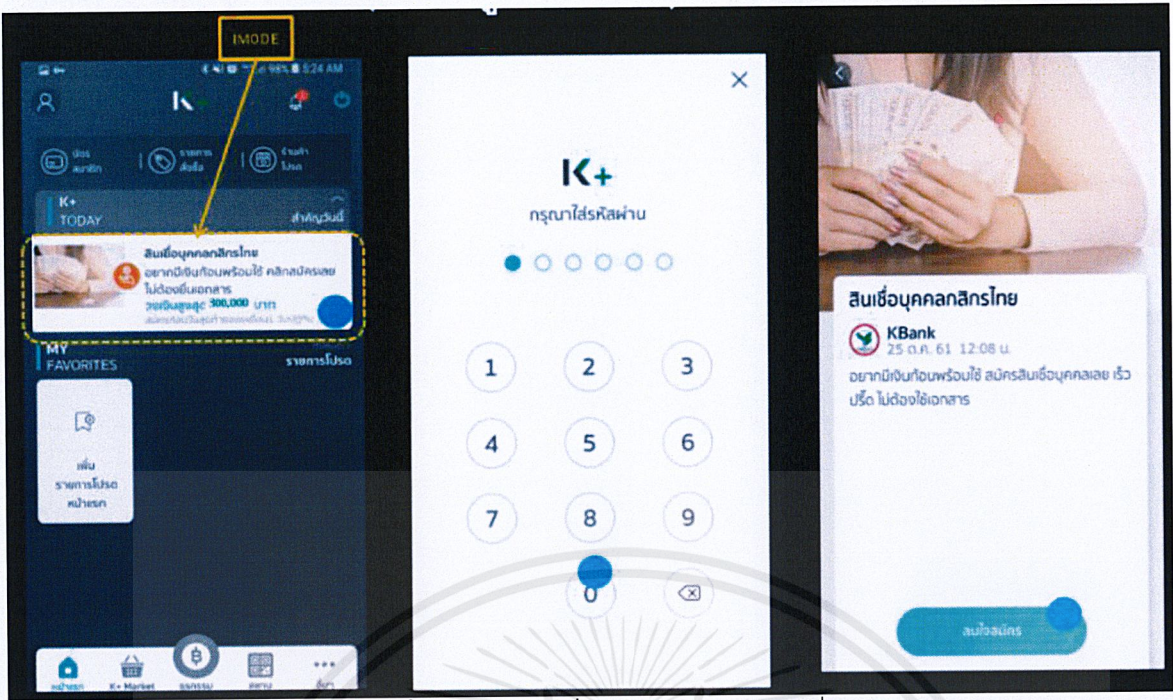
ตลอดโครงการสหกิจศึกษานี้ ได้มีการค้นคว้าและพัฒนาโครงการการกู้ยืมเงินผ่านระบบดิจิทัล ในส่วนของระบบการอนุมัติสินเชื่อเบื้องต้น (Digital Lending : Lead Generation) โดยมีผลจากการพัฒนาระบบดังนี้

ในส่วนแรกนี้จะเริ่มต้นจากการเข้าใช้แอปพลิเคชัน K-PLUS ในโทรศัพท์มือถือเพื่อเข้าไปสมัครสินเชื่อ หรือต้องการทราบข้อมูลได้ 3 ช่องทางหลัก ช่องทางแรกคือ เมื่อเข้าสู่แอปพลิเคชัน ทำธุรกรรม ใสรหัสผ่านเข้าไปยังบัญชี จากนั้นทำการกดสินเชื่อ เมื่อกดเข้าไปจะเจอกับ MINI IMODE จะเป็นโฆษณาชักชวนสมัครสินเชื่อ ลูกค้าทั้งที่ต้องการสมัครและไม่ต้องการสมัครสินเชื่อ จะสามารถเห็นหน้านี้ได้ทั้งหมด หากลูกค้าสนใจสมัครสินเชื่อสามารถกดสนใจสมัครได้ ดังภาพที่ 4.1



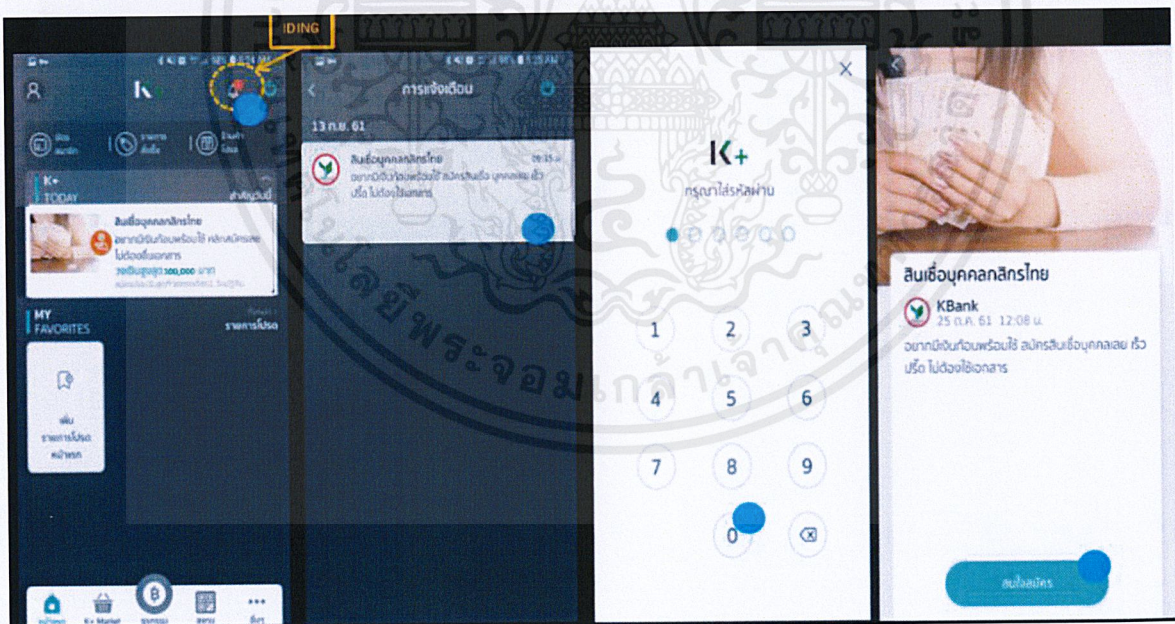
ภาพที่ 4.1 ช่องทางแรกของการสมัครสินเชื่อ

ช่องทางที่ 2 จะเหมาะสำหรับลูกค้าที่ต้องการสมัครสินเชื่อ การโฆษณาสมัครสินเชื่อจะปรากฏบนหน้า KPLUS Today ในส่วนของ IMODE นั่นเอง เมื่อกดเข้าไปทำการกรอกรหัสผ่าน ก็สามารถสมัครสินเชื่อได้ ดังภาพที่ 4.2



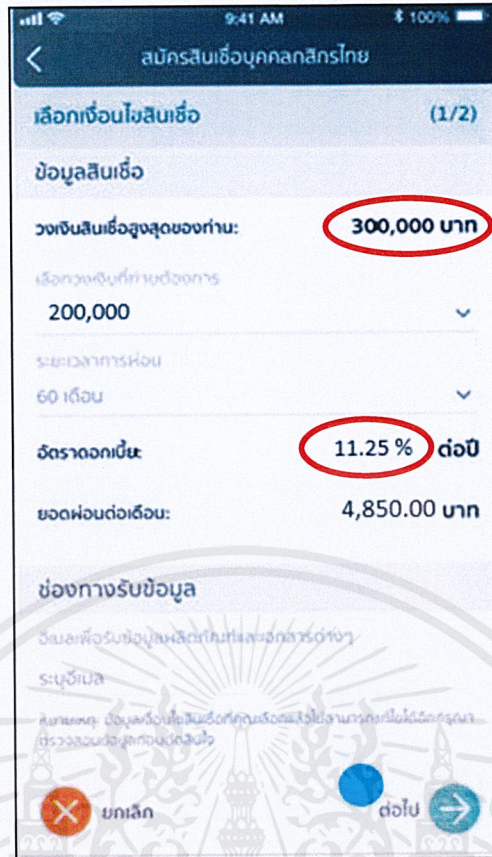
ภาพที่ 4.2 ช่องทางที่ 2 ของการสมัครสินเชื่อ

ช่องทางที่ 3 จะมีการโฆษณาสมัครสินเชื่อผ่านทาง Notification หรือเรียกว่า IDING เมื่อกดเข้าไป จะเจอกับโฆษณา หากลูกค้าสนใจก็สามารถกดเข้าไป ทำการกรอกรหัส และสมัครสินเชื่อได้ ดังภาพที่ 4.3



ภาพที่ 4.3 ช่องทางที่ 3 ของการสมัครสินเชื่อ

เมื่อเข้าสู่ขั้นตอนการสมัครสินเชื่อแล้วนั้น จะมีในส่วนของวงเงิน ที่ทางแอปพลิเคชันได้คำนวณมาให้แล้ว ว่าวงเงินสินเชื่อสูงสุดของลูกค้าคือกี่บาท และอัตราดอกเบี้ยต่อปีกี่บาท ดังภาพที่ 4.4



ภาพที่ 4.4 วงเงินสินเชื่อและอัตราดอกเบี้ย

ซึ่งวงเงินสินเชื่อสูงสุด และอัตราดอกเบี้ยที่ทางแอปพลิเคชันได้คำนวณมาให้นั้น เกิดจากระบบการอนุมัติสินเชื่อเบื้องต้น ในการคำนวณวงเงินและอัตราดอกเบี้ยให้กับลูกค้าเป็นรายบุคคล ดังนั้นลูกค้าแต่ละท่านจะมีวงเงินและอัตราดอกเบี้ยไม่เท่ากัน อีกทั้งยังมีการกรองลูกค้าที่มีรายได้น้อยกว่า 30,000 บาท จะต้องกรอกข้อมูลเพิ่มเติม ทั้งนี้ การที่แอปพลิเคชันสามารถคำนวณวงเงินสินเชื่อ อัตราดอกเบี้ย และรายได้ลูกค้า แสดงในแอปพลิเคชัน K-PLUS ได้นั้น เกิดจากระบบอนุมัติสินเชื่อเบื้องต้นที่ทางวิศวกรข้อมูลเป็นผู้จัดทำขึ้น เพื่อช่วยให้ผลของการคำนวณวงเงินสินเชื่อและอัตราดอกเบี้ยเป็นไปอย่างถูกต้อง ไม่เกิดข้อผิดพลาด

เมื่อทำการกรอกข้อมูลครบถ้วน จะมีหน้ายินยอมต่าง ๆ ปรากฏขึ้นมาให้ลูกค้ากดเลือกยินยอมหรือไม่ยินยอม หากลูกค้าให้ความยินยอม ลูกค้าจะได้รับหลักฐานการให้ความยินยอมทางอีเมลล์ และหากได้รับอนุมัติหรือผ่านกฏสินเชื่อของระบบอนุมัติสินเชื่อเบื้องต้น ลูกค้าจะได้รับสัญญาสินเชื่อ จากนั้นกดตกลง ลูกค้าจะได้รับสำเนาสัญญาผ่านทางอีเมล กระบวนการดังกล่าวสามารถเห็นได้ดังภาพที่ 4.5



ภาพที่ 4.5 กระบวนการยินยอมสมัครสินเชื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์ผลการดำเนินงาน

5.1 บทสรุปปริญญานิพนธ์

การพัฒนาโครงการการกู้ยืมเงินผ่านระบบดิจิทัล ในส่วนระบบการอนุมัติสินเชื่อเบื้องต้น เพื่อทำการตรวจสอบ และกรองลูกค้าที่สามารถได้รับการอนุมัติจากผลิตภัณฑ์ต่าง ๆ เช่น สินเชื่อบ้าน วงเงินกู้เบิกเกินบัญชี บัตรเครดิต และบัตรกดเงินสด เป็นต้น ซึ่งลูกค้าที่มีศักยภาพที่ทางธนาคารจะให้เงินกู้ ลูกค้าที่ได้แสดงความสนใจในสิ่งที่ทางแอปพลิเคชันนำเสนอ และมีข้อมูลช่วยให้ระบุตัวตนได้แน่นอน ลูกค้าเหล่านี้จะเข้ามาเห็นการเสนอการอนุมัติผลิตภัณฑ์ต่าง ๆ ในแอปพลิเคชัน K-PLUS เมื่อลูกค้าสนใจที่จะทำสินเชื่อต่าง ๆ ผ่านระบบดิจิทัล ลูกค้าจะสามารถทำได้อย่างง่ายดาย สะดวก รวดเร็ว และปลอดภัย จากผลการประเมินความพึงพอใจของบริษัทในด้านการคำนวณวงเงิน อัตราดอกเบี้ย และการกรองลูกค้าเพื่อสินเชื่อต่าง ๆ อยู่ในระดับดีถึงดีมาก แต่การดำเนินงานมีความยุ่งยาก เนื่องจากต้องมีการติดต่อประสานงานกับฝ่ายต่าง ๆ มากมาย จึงทำให้งานสำเร็จล่าช้า

5.2 ปัญหาที่พบในระหว่างการทำงาน

- 1) ขาดประสบการณ์ทางด้าน Big Data คำศัพท์ทางเทคนิค และเครื่องมือต่าง ๆ จึงทำให้การทำงานในช่วงแรกเป็นไปด้วยความล่าช้า
- 2) การทำงานเข้าใช้เครื่องมือต่าง ๆ ต้องเชื่อมต่อกับระบบอินเทอร์เน็ตในองค์กรเท่านั้น
- 3) การติดตั้งโปรแกรมเกือบทุกโปรแกรม ต้องทำการติดตั้งกับทาง IT Clinic ของบริษัทเท่านั้น ทำให้การทำงานล่าช้า
- 4) คำศัพท์ทางเทคนิคต่าง ๆ ที่ใช้ในการประชุมสำหรับภักุสินเชื่อที่ใช้กรองลูกค้า เป็นคำศัพท์ที่ไม่เคยใช้งานมาก่อน หรือได้ยินมาก่อน จึงต้องใช้เวลาในการเรียนรู้เบื้องต้น
- 5) ต้องทำงานร่วมกันเป็นกลุ่ม บางครั้งจำนวนงานมากเกินไปจนไม่พอกับกำลังคน ทำให้เกิดความล่าช้าในการสื่อสาร และประสานงานภายในกลุ่ม จึงทำงานได้ไม่รวดเร็วเท่าที่ควร

5.3 แนวทางการแก้ปัญหา

- 1) พยายามศึกษาพื้นฐานทางด้าน Big Data คำศัพท์ทางเทคนิคต่าง ๆ และเครื่องมือในการทำงานเป็นประจำ จนรู้สึกคุ้นเคย ทำให้สามารถทำงานได้อย่างราบรื่น และรวดเร็วยิ่งขึ้น
- 2) ศึกษาหาความรู้จากอินเทอร์เน็ตเพิ่มเติม เมื่อเกิดความไม่เข้าใจหรือการทำงานไม่เป็นไปตามความต้องการ

- 3) มีการรายงานความคืบหน้าของงานที่ได้รับมอบหมายให้หัวหน้าวิศวกรในที่อย่างสม่ำเสมอ
- 4) ปรึกษาขอคำแนะนำจากพี่วิศวกรในที่

5.4 ข้อเสนอแนะและแนวทางในการพัฒนา

โครงการการกู้ยืมเงินผ่านระบบดิจิทัลโดยระบบการอนุมัติสินเชื่อเบื้องต้น ในการที่จะกรองลูกค้าที่สามารถได้รับการอนุมัติจากผลิตภัณฑ์ต่าง ๆ ได้นั้น จะต้องมีความรู้ของการกรองในแต่ละสินเชื่อไม่เหมือนกัน ทำให้ใช้เวลานานในการตัดสินใจตั้งกฎการกรองต่าง ๆ จึงมีแนวทางการพัฒนาคือ หากโครงการใดมีลักษณะคล้ายกัน สามารถใช้กฎการกรองเหมือนกัน หรือทำการปรับเปลี่ยนเพิ่มเติมบางกฎได้เมื่อเห็นสมควร เพื่อเพิ่มประสิทธิภาพให้มากขึ้น



เอกสารอ้างอิง

- [1] บริษัท กลสิกร เทคโนโลยี กรุ๊ป เซครเทเรียด จำกัด. [ออนไลน์]. เข้าถึงได้จาก : <http://www.kbtg.tech/th>. (วันที่ค้นข้อมูล: 13 พฤศจิกายน 2562).
- [2] Lead. [ออนไลน์]. เข้าถึงได้จาก : <https://contentshifu.com/clarify-marketing-sales-qualified-lead/>. (วันที่ค้นข้อมูล: 20 สิงหาคม 2562).
- [3] Database. [ออนไลน์]. เข้าถึงได้จาก : <https://saixiii.com/what-is-database/>. (วันที่ค้นข้อมูล: 20 สิงหาคม 2562).
- [4] Database server. [ออนไลน์]. เข้าถึงได้จาก : <https://maae2517.wordpress.com/2011/10/26/database-server-สำหรับการใช้งานในองค์กร/>. (วันที่ค้นข้อมูล: 20 สิงหาคม 2562).
- [5] SQL. [ออนไลน์]. เข้าถึงได้จาก : <https://www.mindphp.com/คู่มือ/73-คืออะไร/2088-sql-คืออะไร.html>. (วันที่ค้นข้อมูล: 20 สิงหาคม 2562).
- [6] ประโยชน์ของภาษา SQL. [ออนไลน์]. เข้าถึงได้จาก : <https://www.9experttraining.com/articles/ภาษา-sql-คืออะไร>. (วันที่ค้นข้อมูล: 20 สิงหาคม 2562).
- [7] การนำภาษา SQL ไปใช้งาน. [ออนไลน์]. เข้าถึงได้จาก : <https://www.9experttraining.com/articles/ภาษา-sql-คืออะไร>. (วันที่ค้นข้อมูล: 20 สิงหาคม 2562).

เอกสารอ้างอิง (ต่อ)

- [8] Hadoop. [ออนไลน์]. เข้าถึงได้จาก :
<http://bigdataexperience.org/what-to-know-about-hadoop/>. (วันที่ค้นข้อมูล: 26 สิงหาคม 2562).
- [9] ETL Tool. [ออนไลน์]. เข้าถึงได้จาก :
<http://www.ltotal.com/2014/05/etl-tool.html>. (วันที่ค้นข้อมูล: 3 ตุลาคม 2562).
- [10] DbVisualizer. [ออนไลน์]. เข้าถึงได้จาก :
<http://th.softwared.com/developer-tools/download-dbvisualizer-342540-for-windows.html>. (วันที่ค้นข้อมูล: 3 ตุลาคม 2562).

ประวัติผู้เขียน



หัวข้อโครงการ การกู้ยืมเงินผ่านระบบดิจิทัลโดยระบบการอนุมัติสินเชื่อเบื้องต้น
ชื่อ-สกุล นางสาวชนนิกานต์ วรรณศักดิ์
รหัสนักศึกษา 59010267
คณะ วิศวกรรมศาสตร์
ภาควิชา วิศวกรรมคอมพิวเตอร์
สาขาวิชา วิศวกรรมสารสนเทศ

ประวัติส่วนตัว

วันเดือนปีเกิด 5 กันยายน 2540
ที่อยู่ 69 ซอย 2 หมู่ 5 บ้านไร่ ถนนเชียงใหม่-ฮอด ตำบลทุ่งต้อม อำเภอสันป่าตอง
จังหวัดเชียงใหม่ 50120

ประวัติการศึกษา

2559-ปัจจุบัน สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (ระดับปริญญาตรี)
2553-2558 โรงเรียนพระหฤทัย เชียงใหม่ (ระดับมัธยมศึกษา)
2546-2552 โรงเรียนพระหฤทัย เชียงใหม่ (ระดับประถมศึกษา)