เว็บไซต์เพื่อจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆ

GATHERCLOUD : WEB-BASED APPLICATION FOR MANAGING MULTIPLE CLOUD STORAGES

ธีรพรรณ วิเชียรภักดา

ชาวิตรี บุญชนะ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2563

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เว็บไซต์เพื่อจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆ

GATHERCLOUD: WEB-BASED APPLICATION FOR MANAGING MULTIPLE CLOUD STORAGES

ธีรพรรณ วิเชียรศักดา

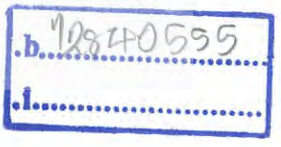
TEERAPAN VICHIANSAKDA

สาวิตรี บุญสอน

SAWITREE BOONSORN

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร. ธนิศา นุ่มนนท์



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เว็บไซต์เพื่อจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆ

GATHERCLOUD: WEB-BASED APPLICATION FOR MANAGING MULTIPLE CLOUD STORAGES

ธีรพรรณ วิเชียรศักดา

TEERAPAN VICHIANSAKDA

สาวิตรี บุญสอน

SAWITREE BOONSORN

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร. ธนิศา นุ่มนนท์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GATHERCLOUD: WEB-BASED APPLICATION FOR MANAGING MULTIPLE CLOUD STORAGES

TEERAPAN VICHIANSAKDA

SAWITREE BOONSORN

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/2015

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2016

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองปริญญานิพนธ์ ประจำปีการศึกษา 2558

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เว็บไซต์เพื่อจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆ

GATHERCLOUD: WEB-BASED APPLICAION FOR MANAGING MULTIPLE CLOUD STORAGES

ผู้จัดทำ

1. นายธีรพรรณ วิเชียรศักดา รหัสนักศึกษา 55070059
2. นางสาวสาวิตรี บุญสอน รหัสนักศึกษา 55070129

.....อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร. ธนิศา นุ่มนนท์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน	เว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆ
นักศึกษา	นายธีรพรรณ วิเชียรศักดา รหัสนักศึกษา 55070059
	นางสาวสาวิตรี บุญสอน รหัสนักศึกษา 55070129
ปริญญา	วิทยาศาสตรบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
ปีการศึกษา	2558
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. ธนิศา นุ่มนนท์

บทคัดย่อ

แหล่งเก็บข้อมูลบนกลุ่มเมฆได้เริ่มเป็นที่นิยมมากขึ้นอย่างเห็นได้ชัดในปัจจุบัน ผู้ให้บริการแหล่งเก็บข้อมูลบนก้อนเมฆก็ได้พยายามที่จะนำเสนอบริการแหล่งเก็บข้อมูลบนก้อนเมฆที่หลากหลาย ดังนั้นจึงทำให้ผู้ใช้มักที่จะสมัครหรือเลือกที่จะใช้บริการมากกว่าหนึ่งจากผู้ให้บริการแหล่งเก็บข้อมูลบนก้อนเมฆ อย่างไรก็ตามผู้ใช้อาจจะพบกับความลำบากในการเข้าถึงและการจัดการแหล่งเก็บข้อมูลบนก้อนเมฆจำนวนหลายบริการได้อย่างต่อเนื่อง

ด้วยเหตุที่กล่าวมาข้างต้น ทำให้ผู้พัฒนาได้คิดค้นเว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆ ซึ่งโครงงานนี้จะเป็นตัวช่วยให้สามารถจัดการข้อมูลบนกลุ่มเมฆได้ง่ายขึ้น โดยเฉพาะผู้ใช้ที่มีการใช้งานแหล่งเก็บข้อมูลบนกลุ่มเมฆมากกว่าหนึ่งแหล่งจะสามารถเข้าถึงแหล่งเก็บข้อมูลและจัดการข้อมูลบนกลุ่มเมฆได้พร้อมกัน ช่วยให้มีความสะดวกสบายในการเข้าถึงข้อมูลมากยิ่งขึ้น ลดความยุ่งยากและความซ้ำซ้อน โดยเริ่มแรกโครงงานนี้สามารถจัดการข้อมูลจากแหล่งเก็บข้อมูลบนกลุ่มเมฆได้ 5 แหล่งด้วยกัน ได้แก่ Google Drive, OneDrive, Box, Dropbox และ Copy โดยการเรียกใช้งานผ่าน Application Program Interface (API) ของแต่ละผู้ใช้บริการ ทำให้มีความปลอดภัยและความน่าเชื่อถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Title	GatherCloud: Web-based Application for Managing Multiple Cloud Storages
Student	Teerapan Vichiansakda Student ID 55070059
	Sawitree Boonsorn Student ID 55070129
Degree	Bachelor of Science
Program	Information Technology
Academic Year	2015
Advisor	Asst. Prof. Dr. Thanisa Numnonda

Abstract

Cloud storage has become more significantly popular. Many cloud storage providers try to introduce a variety of cloud storage services. Therefore, one user usually applies to more than one cloud storage provider. However, users may feel inconvenient to access and manage their multiple online cloud storage services simultaneously.

From the problem statement above, developers have an idea of inventing GatherCloud: Web-based Application for Managing Multiple Cloud Storages. This project will help in managing data in cloud storage especially user that use more than one cloud storage service. GatherCloud can help user to access cloud storage and manage data inside it simultaneously providing convenient in accessing multiple cloud storages, reduce complication and redundancy. At first, this project will manage data from these 5 according cloud storage service provider including Google Drive, OneDrive, Box, Dropbox and Copy through API of each provider to guarantee security and reliability.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่องเว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆฉบับนี้นั้นสามารถสำเร็จลุล่วงได้ด้วยความกรุณาและการสนับสนุนจาก ผู้ช่วยศาสตราจารย์ ดร.ธนิศา นุ่นนนท์ อาจารย์ที่ปรึกษาของโครงงานและคณะอาจารย์จากคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่านที่ได้ให้แรงสนับสนุนและความร่วมมือเป็นอย่างดีในทุกๆ ด้าน ไม่ว่าจะเป็น ทางด้านความรู้และข้อเสนอแนะต่างๆ การให้ข้อมูล การเสนอแนวคิด และแนวทางในการแก้ปัญหาหรืออุปสรรคที่พบเจอขณะที่ได้ศึกษาและพัฒนาโครงงานดังกล่าว จนสามารถเกิดเป็นปริญญานิพนธ์เล่มนี้ที่เสร็จสมบูรณ์ ทางผู้จัดทำจึงขอแสดงความขอบพระคุณเป็นอย่างยิ่งไว้ ณ ที่นี้

ผู้จัดทำขอแสดงความขอบพระคุณต่อเจ้าหน้าที่งานสนับสนุนเทคโนโลยีสารสนเทศ และเจ้าหน้าที่งานทะเบียน คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้ความร่วมมือและประสานงานด้วยดีตลอดการจัดทำจนสามารถก่อเกิดเป็นปริญญานิพนธ์ฉบับนี้ได้

นอกจากนี้ ผู้จัดทำยังขอขอบคุณนายสรวุฒิ กิตติศรีชัยกุล และนายสารัช ท้วมชมธรรม ซึ่งเป็นผู้ที่มีประสบการณ์ในการพัฒนาเว็บไซต์และเว็บแอพพลิเคชั่น ที่เสียสละเวลาส่วนตัวมาช่วยให้คำปรึกษาและให้คำแนะนำ รวมทั้งช่วยอธิบายและเสริมเกร็ดความรู้ต่างๆ ที่ผู้จัดทำไม่ทราบ ทำให้สามารถเรียนรู้เรื่องต่างๆ ได้อย่างชัดเจนและเข้าใจมากยิ่งขึ้น

ธีรพรรณ วิเชียรศักดา

สาวิตรี บุญสอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญรูป VII

สารบัญตาราง XI

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการพัฒนา 1

1.3 ขอบเขตโครงงาน 1

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในโครงงาน 2

1.5 ขั้นตอนออกแบบและพัฒนาระบบ 2

1.6 ประโยชน์ที่คาดว่าจะได้รับ 3

บทที่ 2 การทบทวนวรรณกรรมที่เกี่ยวข้อง 4

2.1 การประมวลผลบนกลุ่มเมฆ (Cloud Computing) 4

2.1.1 Infrastructure as a Service (IaaS) 4

2.1.2 Platform as a Service (PaaS) 5

2.1.3 Software as a Service (SaaS) 5

2.2 แหล่งเก็บข้อมูลบนกลุ่มเมฆ (Cloud Storage) 5

2.3 เว็บไซต์ 5

2.3.1 JavaScript 6

2.3.2 PHP 6

2.3.3 Composer 6

2.4 Application Program Interface (API) 6

2.5 JavaScript Object Notation (JSON) 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.6 Representational State Transfer (REST) 7

2.7 cURL 7

2.8 OAuth 7

2.9 MVC Architectural Pattern 8

2.10 Nested Set [1] 9

บทที่ 3 การวิเคราะห์และออกแบบระบบ 10

3.1 ศึกษาระบบงานเดิม 10

3.2 ปัญหาที่พบในระบบปัจจุบัน 11

3.3 การวิเคราะห์ความต้องการ 12

3.3.1 ความต้องการที่เป็นหน้าที่หลัก (Functional Requirement) 12

3.3.2 ความต้องการที่ไม่ใช่หน้าที่หลักของระบบ (Non-Functional Requirement) 13

3.4 การวิเคราะห์และวิจารณ์ระบบที่ต้องการออกแบบ 13

3.4.1 จุดประสงค์ของโครงงาน 13

3.4.2 เหตุผลในการเลือกใช้อุปกรณ์และเครื่องมือต่างๆในการพัฒนา 13

3.5 การออกแบบระบบใหม่ 15

3.5.1 บล็อกไดอะแกรม (Block Diagram) 15

3.5.2 ยูสเคสโมเดล 16

3.5.3 Activity Diagram 26

3.5.4 Website Structure 34

3.6 การออกแบบฐานข้อมูล 35

3.6.1 ER Diagram 35

3.6.2 รายละเอียดตาราง (Table) และ แอตทริบิวต์ (Attribute) 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 4 ระบบต้นแบบหรือผลการทดลอง 39

4.1 ผลการทดลองและการศึกษา 39

4.1.1 ผลการศึกษา API ของ Dropbox [3] 39

4.1.2 ผลการศึกษา API ของ Copy [4] 44

4.1.3 ผลการศึกษา API ของ Onedrive [5] 49

4.1.4 ผลการศึกษา API ของ Box [6] 54

4.1.5 ผลการศึกษา API ของ Google Drive [7] 57

4.1.6 บันทึกการเพิ่มการรองรับผู้ให้บริการของระบบ 61

4.1.7 บันทึกการลดการรองรับผู้ให้บริการของระบบ 63

4.2 ระบบต้นแบบ 64

4.2.1 Framework 64

4.2.2 Graphic User Interface (GUI) 67

4.3.3 ผลการทำงานของระบบต้นแบบ 69

บทที่ 5 สรุปผลและข้อเสนอแนะ 80

5.1 สรุปผลการดำเนินงาน 80

5.2 ปัญหาและอุปสรรคที่พบในการพัฒนาระบบ 80

5.3 ข้อจำกัดของระบบ 81

5.4 ข้อเสนอแนะและแนวทางในการพัฒนาระบบในอนาคต 81

บรรณานุกรม 83

ภาคผนวก ก. 84

ภาคผนวก ข. 100

ประวัติผู้เขียน 105

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 ภาพจำลองรูปแบบ MVC 8
2.2 ภาพจำลองการเก็บข้อมูลแบบ Nested Set 9
3.1 หน้าเว็บไซต์ MultCloud 10
3.2 หน้าต่างการจัดการข้อมูลในแต่ละแหล่งเก็บข้อมูลบนกลุ่มเมฆ 11
3.3 แผนภาพ Block Diagram ของโครงสร้างการทำงานของ GatherCloud 15
3.4 แผนภาพยูสเคสของ GatherCloud 17
3.5 แผนภาพกิจกรรมยูสเคสสมัครสมาชิก 26
3.6 แผนภาพกิจกรรมยูสเคสเข้าสู่ระบบ 27
3.7 แผนภาพกิจกรรมยูสเคสตั้งค่าข้อมูลส่วนตัว 28
3.8 แผนภาพกิจกรรมยูสเคสเพิ่มการเชื่อมต่อกับ Cloud Storage 29
3.9 แผนภาพกิจกรรมยูสเคสจัดการไฟล์และข้อมูล 30
3.10 แผนภาพกิจกรรมยูสเคสจัดการย้ายไฟล์หรือข้อมูลข้าม Cloud Storage 31
3.11 แผนภาพกิจกรรมยูสเคสค้นหาไฟล์หรือข้อมูล 32
3.12 แผนภาพกิจกรรมยูสเคสสร้างมุมมองการมองเห็นข้อมูล 33
3.12 โครงสร้างเว็บไซต์ Gathercloud 34
3.13 แผนภาพโครงสร้างฐานข้อมูล 35
4.1 หน้า Application Console ของ Dropbox 40
4.2 แผนภาพหลักการการเชื่อมต่อของกับ Dropbox API 42
4.3 หน้า Application Console ของ Copy 44
4.4 หน้ายินยอมสิทธิในการเข้าถึงบัญชีผู้ใช้จาก Copy Application 46
4.5 หน้าสร้าง Application ของ OneDrive 49
4.6 แผนภาพโฟลว์การทำงานของ Token Code 50
4.7 แผนภาพโฟลว์การทำงานของ Authorization Code 52
4.8 หน้าสร้าง Application ของ Box 55
4.9 ขั้นตอนการร้องขอ Token ในรูปแบบของ Web Aplication 58
4.10 หน้าเว็บไซต์รายละเอียดของ Google Client Library 62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.11 โครงสร้างของ Google Client Library 62
4.12 รายละเอียดของ ModelInterface ในรูปแบบของ UML 63
4.13 หน้าเลือกผู้ให้บริการแหล่งเก็บข้อมูลเพื่อเชื่อมต่อกับ GatherCloud 63
4.14 โครงสร้างไฟล์ของ Laravel 5.1 66
4.15 หน้าจัดการไฟล์ของ GatherCloud 67
4.16 หน้าจัดการไฟล์ของ GatherCloud พร้อมตัวบ่งชี้การใช้งาน 67
4.17 หน้าสมัครสมาชิก 69
4.18 หน้าล็อกอิน 69
4.19 เมนู Dropdown 70
4.20 หน้าจัดการข้อมูลผู้ใช้ 70
4.21 หน้าเลือกผู้ให้บริการสำหรับเพิ่มการเชื่อมต่อกับระบบ 71
4.22 หน้ายินยอมสิทธิการเข้าถึงของผู้ให้บริการ Box 71
4.23 หน้ายินยอมสิทธิการเข้าถึงของผู้ให้บริการ Dropbox 71
4.24 หน้ายินยอมสิทธิการเข้าถึงของผู้ให้บริการ Google Drive 72
4.25 หน้ายินยอมสิทธิการเข้าถึงของผู้ให้บริการ OneDrive 72
4.26 หน้าจัดการการเชื่อมต่อกับผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆ 73
4.27 หน้าจัดการไฟล์ในแหล่งเก็บข้อมูลแบบแยกของระบบ 73
4.28 หน้าจัดการไฟล์ในเว็บไซต์ Dropbox ที่ได้เพิ่มการเชื่อมต่อเข้าไป 74
4.29 หน้าเมนูในการจัดการไฟล์ 74
4.30 หน้าจัดการไฟล์ในแหล่งเก็บข้อมูลแบบรวมของระบบ 75
4.31 หน้าแรกของฟังก์ชั่น GatherLinks 76
4.32 หน้าสำหรับเลือกไฟล์เพื่อสร้างมุมมอง 76
4.33 หน้าสรุปผลการเลือกไฟล์เพื่อสร้างมุมมอง 76
4.34 หน้าแสดงมุมมองจากผู้ใช้ที่เป็นเจ้าของ 77
4.35 หน้าแสดงมุมมองจากผู้ใช้อื่น 77
4.36 หน้าแสดงผลลัพธ์การค้นหาไฟล์ 78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.37 หน้าแสดงผลลัพธ์การตรวจสอบความซ้ำซ้อนของไฟล์ 78
4.38 หน้าแสดงผลลัพธ์คำนวนพื้นที่คงเหลือของแหล่งเก็บข้อมูลบนกลุ่มเมฆ 79
4.39 หน้าการเลือกปลายทางการโอนย้ายไฟล์ 79
ก.1 การเข้าถึงหน้า Developers ของ Dropbox 84
ก.2 หน้า Developers ของ Dropbox 85
ก.3 หน้า Create New Application ของ Dropbox 86
ก.4 หน้าต่างรายละเอียดของ Application ที่สร้างบน Dropbox 87
ก.5 หน้าต่างลงทะเบียนเป็น Developers ของ Copy 88
ก.6 หน้าต่างการจัดการ Application ของ Copy 88
ก.7 หน้าต่างสร้าง Application ใหม่ของ Copy 89
ก.8 หน้าต่างรายละเอียดของ Application ของ Copy 89
ก.9 หน้าต่างการจัดการ Application ของ OneDrive 90
ก.10 หน้าต่างสร้าง Application ใหม่บน OneDrive 90
ก.11 หน้าต่างตั้งค่า Redirect URL ภายใน Application ของ OneDrive 91
ก.12 หน้าต่างแสดงรายละเอียด Application ของ OneDrive 91
ก.13 หน้าต่างเมนู Developers ของ Box 92
ก.14 หน้าต่างการสร้าง Application ใหม่ของ Box 93
ก.15 หน้าต่างตั้งค่า Application ของ Box 94
ก.16 หน้าต่างแสดงข้อมูลและตั้งค่าเกี่ยวกับ Application ของ Box 95
ก.17 หน้าต่างสร้าง Project สำหรับ Google 96
ก.18 หน้าต่างแสดงข้อมูลและตั้งค่าเกี่ยวกับ Project ของ Google 97
ก.19 หน้าต่างสำหรับเลือกประเภท Credentials ของ Google 97
ก.20 หน้าต่างสำหรับสร้าง Credentials ประเภท OAuth client ID ของ Google 98
ก.21 หน้าต่างแสดง client ID และ cilent secret 98
ก.22 หน้าต่างแสดง Overview แทบ Google APIs 99
ก.23 หน้าต่างแสดงรายละเอียดของ Google Dive API 99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

ข.1 หน้าต่าง XAMPP Control Panel 101
ข.2 ชุดคำสั่งเพื่อติดตั้ง Composer 102
ข.3 ชุดคำสั่งเพื่อติดตั้ง Composer 102
ข.4 ชุดคำสั่งเพื่อติดตั้ง Laravel Framework 102
ข.5 ชุดคำสั่งเพื่อสร้าง Project 102
ข.6 ชุดคำสั่งเพื่อสร้าง Project ผ่าน Composer 102
ข.7 ภาพการวางไฟล์ php_oauth.dll ไว้ใน path ที่ระบุ 103
ข.8 ภาพไลบรารี่ OAuth ที่ปรากฏใน phpinfo() 103
ข.9 ส่วนเชื่อมต่อกับฐานข้อมูลภายในไฟล์ .env 104

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

3.1 รายละเอียดของฟังก์ชั่นฟรีและฟังก์ชั่นที่ต้องเสียค่าใช้จ่ายของเว็บไซต์ MultCloud 12
3.2 รายละเอียดการทำงานของฟังก์ชั่นการสมัครสมาชิก 18
3.3 รายละเอียดการทำงานของฟังก์ชั่นการเข้าสู่ระบบ 19
3.4 รายละเอียดการทำงานของฟังก์ชั่นตั้งค่าข้อมูลส่วนตัว.....20
3.5 รายละเอียดการทำงานของฟังก์ชั่นเพิ่มการเชื่อมต่อกับ Cloud Storage 21
3.6 รายละเอียดการทำงานของฟังก์ชั่นจัดการไฟล์และข้อมูล 22
3.7 รายละเอียดการทำงานของฟังก์ชั่นย้ายไฟล์หรือข้อมูลข้าม Cloud Storage 23
3.8 รายละเอียดการทำงานของฟังก์ชั่นค้นหาไฟล์หรือข้อมูล 24
3.9 รายละเอียดการทำงานของฟังก์ชั่นสร้างมุมมองการมองเห็นข้อมูล 25
3.10 แอตทริบิวต์ตาราง users 35
3.11 แอตทริบิวต์ตาราง tokens 36
3.12 แอตทริบิวต์ตาราง providers 36
3.13 แอตทริบิวต์ตาราง links.....37
3.14 แอตทริบิวต์ตาราง dummy_files 37
3.15 แอตทริบิวต์ตาราง files.....38
4.1 รายละเอียด HTTP Request ในการเชื่อมต่อกับ Dropbox API 40
4.2 รายละเอียด HTTP Request ในการใช้ Code เพื่อดึง Access Token.....41
4.3 รายชื่อฟังก์ชั่นการทำงานของ Dropbox REST API 43
4.4 รายละเอียด HTTP Request ของ Copy API เพื่อขอ Request Token 45
4.5 รายละเอียด HTTP Request ของ Copy API เพื่อขอ Access Token 47
4.6 รายละเอียดเกี่ยวกับฟังก์ชั่นที่สนับสนุนของ Copy API 48
4.7 รายละเอียด HTTP Request ของ OneDrive API เพื่อขอ Access Token โดยตรง.....50
4.8 รายละเอียด HTTP Request ของ OneDrive API เพื่อขอ Authorization code.....51
4.9 รายละเอียด HTTP Request ของ OneDrive API เพื่อขอ Access Token ด้วย Code.....51
4.10 รายละเอียดเกี่ยวกับฟังก์ชั่นของ OneDrive API 53
4.11 รายละเอียด HTTP Request ของ Box API เพื่อขอ Authorization Code.....55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

4.12 รายละเอียด HTTP Request ไว้ร้องขอ Access Token จาก Box API ด้วย Code.....56
4.13 รายละเอียดฟังก์ชั่นการทำงานของ Box API.....56
4.14 รายละเอียด HTTP Request ของ Google Drive API เพื่อขอ Authorization Code.....59
4.15 รายละเอียด HTTP Request ของ Google Drive API เพื่อขอ Token จาก code.....59
4.16 รายละเอียดฟังก์ชั่นการทำงานของ Google Drive API.....60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันข้อมูลบนเครือข่ายนั้นได้มีบทบาทสำคัญมากยิ่งขึ้นในการใช้ชีวิตประจำวันของผู้คน จึงทำให้การจัดเก็บรักษาข้อมูลบนเครือข่ายมีความสำคัญไม่แพ้กัน ซึ่งในช่วงเวลาที่ผ่านมาได้เกิดเทคโนโลยีการเก็บข้อมูลออนไลน์ หรือการเก็บข้อมูลบนกลุ่มเมฆ หมายถึง การเก็บข้อมูลฝากไว้กับผู้ที่ให้บริการ โดยมีข้อดีได้แก่ ทำให้ผู้ใช้สามารถเข้าถึงข้อมูลได้ทุกที่จึงช่วยให้มีความสะดวกสบายมากยิ่งขึ้น และป้องกันการสูญหายของข้อมูล อีกทั้งยังช่วยสนับสนุนการแชร์ข้อมูลกับผู้อื่นอีกด้วย ทำให้มีผู้สนใจใช้งานเป็นจำนวนมาก รวมถึงทำให้เกิดผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆหลายรายด้วยกัน ซึ่งมีผู้ใช้จำนวนมากที่ใช้งานแหล่งเก็บข้อมูลบนกลุ่มเมฆมากกว่าหนึ่งแหล่ง ทั้งนี้การใช้งานแหล่งเก็บข้อมูลบนกลุ่มเมฆมากกว่าหนึ่งแหล่งจะส่งผลให้เกิดการกระจายกันของพื้นที่ที่ใช้จัดเก็บข้อมูล เกิดปัญหาความซ้ำซ้อนและความยุ่งยากในการจัดการข้อมูลทั้งหมดจากแหล่งเก็บข้อมูลบนกลุ่มเมฆที่แตกต่างกัน เช่น ไม่สามารถหาข้อมูลได้ตามที่ต้องการเพราะลืมว่าเก็บไว้ที่แหล่งเก็บข้อมูลใด เป็นต้น

ดังนั้นผู้พัฒนาจึงได้คิดเว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆ หรือ GatherCloud เพื่อมาแก้ไขปัญหาที่กล่าวมาเบื้องต้น โดยเชื่อว่าเว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆ จะทำให้ผู้ใช้มีความสะดวกสบายมากยิ่งขึ้น

1.2 ความมุ่งหมายและวัตถุประสงค์ของการพัฒนา

1. เพื่อให้ผู้ใช้งานแหล่งเก็บข้อมูลบนกลุ่มเมฆหลายแหล่งบริการ สามารถใช้งานและจัดการข้อมูลบนแหล่งเก็บข้อมูลบนกลุ่มเมฆดังกล่าวได้พร้อมกันในที่เดียว โดยมองเห็นแหล่งเก็บข้อมูลเป็นกลุ่มเดียวกันทั้งหมด

1.3 ขอบเขตโครงงาน

โครงงานเว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆนั้นจะเป็นการพัฒนาเว็บไซต์ด้วยภาษา PHP โดยจะมีการทำงานร่วมกับ Application Programmable Interface (API) ของ ผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆ ได้แก่ Google Drive, OneDrive, Box, Dropbox และ Copy เพื่อสามารถทำการจัดการข้อมูลได้ดังนี้ อัพโหลด, ดาวน์โหลด, อ่าน, แก้ไข และลบ ทั้งนี้ผู้ใช้งานต้องมีบัญชีของผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆที่กล่าวมาข้างต้นและอนุญาตสิทธิการเข้าถึงข้อมูล จึงจะสามารถใช้งานการจัดการข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในโครงงาน

แนวคิดหลักที่ใช้ในการพัฒนาและออกแบบโครงงานเว็บไซต์เพื่อจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆ คือ ต้องการให้ผู้ใช้สามารถใช้งานฟังก์ชั่นต่างๆ ของผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆหลายๆ ผู้ให้บริการไว้ในที่เดียวกันและสามารถทำงานรวมกันโดยไม่ต้องแบ่งแยก ซึ่งแนวทางและทฤษฎีในการพัฒนาและออกแบบนั้นจะใช้ภาษา PHP และ MySQL มาช่วยในการเก็บข้อมูลและประมวลผล และใช้งานฟังก์ชั่นต่างๆ ของ API ของเหล่าผู้ให้บริการเพื่อเข้าถึงข้อมูลของผู้ใช้ที่ผู้ให้บริการได้เก็บไว้ ซึ่งทางผู้พัฒนาจะพยายามออกแบบระบบให้มีการเรียกใช้ API ต่างๆ เท่าที่จำเป็นเท่านั้นเพื่อลดภาระทางเครือข่ายและหลีกเลี่ยงข้อจำกัดต่างๆ ที่ผู้ให้บริการได้กำหนดให้กับเหล่าผู้พัฒนา โดยมีการรวบรวม API หรือ Library ต่างๆ มาไว้ในที่เดียวและออกแบบให้สามารถทำงานร่วมกันได้ภายใต้ระบบเดียว

การรวบรวมข้อมูลต่างๆ ที่ได้มาจากผู้ให้บริการผ่าน API นั้นจะมีรูปแบบที่ไม่เหมือนกัน ดังนั้นจึงต้องมีการจัดการให้เหมาะสมเพื่อที่จะสามารถนำมาประมวลผลได้ นอกจากนี้ถ้ามีการพัฒนาระบบตามหลักการและแนวคิดที่กล่าวมาด้านบน จะพบว่าจะสามารถนำข้อมูลของผู้ใช้มาประมวลอีกต่อหนึ่งได้ ทำให้เกิดการประยุกต์ข้อมูลได้หลากหลายตามที่ต้องการ แต่ทว่าหากนำข้อมูลนั้นมาเก็บไว้ก่อนก็จะทำให้ข้อมูลไม่เป็นปัจจุบันหรือ Real-time เพราะตัวข้อมูลที่ต้องการนั้นระบบต้องร้องขอจากผู้ให้บริการเสมอซึ่งบางครั้งก็ต้องร้องขอจากหลายผู้ให้บริการพร้อมๆ กัน ผู้พัฒนาจึงไม่เก็บข้อมูลใดๆ ไว้จึงทำให้ต้องมีการร้องขอจากผู้ให้บริการทุกครั้งที่ต้องการข้อมูลเพื่อคงความปัจจุบันของข้อมูล

1.5 ขั้นตอนออกแบบและพัฒนาระบบ

1. ศึกษาหลักการทำงานพื้นฐานของแหล่งเก็บข้อมูลบนกลุ่มเมฆ
2. ศึกษาแหล่งเก็บข้อมูลบนกลุ่มเมฆและหาฟังก์ชั่นการทำงานที่เป็นเอกลักษณ์ของแต่ละผู้ให้บริการ
3. ค้นหา วิเคราะห์ และศึกษาระบบหรือเว็บไซต์ที่มีการทำงานใกล้เคียง
4. ศึกษาวิธีใช้งาน API ของ Google Drive, OneDrive, Box, Dropbox และ Copy
5. ศึกษาวิธีการพัฒนาเว็บไซต์โดยใช้ภาษา PHP, CSS, HTML5 และ JavaScript
6. รวบรวมความต้องการของผู้ใช้งาน
7. ออกแบบและเริ่มพัฒนาระบบที่สามารถบูรณาการ API เข้าด้วยกันได้
8. ทดสอบระบบและนำผลทดสอบมาวิเคราะห์และปรับปรุง แก้ไขข้อบกพร่องและเพิ่มเติมในส่วนที่ขาดหาย
9. ปรับปรุงระบบให้มีความเป็นมาตรฐานและยึดตามหลักการปฏิบัติที่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. สรุปผลการดำเนินการตั้งแต่ต้นจนจบ

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจหลักการการทำงานของแหล่งเก็บข้อมูลบนกลุ่มเมฆ

2. รู้จักการวางแผนและพัฒนาระบบให้มีความเป็นมาตรฐานและปฏิบัติตามหลักการที่ดีของการพัฒนาระบบ

3. รู้จักโครงสร้างและหลักการทำงานของ API

4. มีความเข้าใจในการพัฒนาเว็บไซต์ด้วยภาษา PHP, CSS, HTML5 และ JavaScript

5. ช่วยอำนวยความสะดวกในการเข้าถึงบริการแหล่งเก็บข้อมูลบนกลุ่มเมฆ ได้หลาย ๆ บริการพร้อมกัน

6. ตระหนักถึงความสำคัญของความปลอดภัยบนเครือข่ายและสามารถเข้าใจถึงหลักการในพัฒนาระบบที่มีความปลอดภัยได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
การทบทวนวรรณกรรมที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีบทต่างๆ ที่เกี่ยวข้องกับการพัฒนาระบบซึ่งเนื้อหาจะมีรายละเอียดดังนี้

2.1 การประมวลผลบนกลุ่มเมฆ (Cloud Computing)

เทคโนโลยีใหม่ที่เกิดขึ้นในยุคปัจจุบันมีมากมาย หนึ่งในนั้นคือ การประมวลผลบนกลุ่มเมฆ ซึ่งเป็นเทคโนโลยีที่มีหลักการคือ การแบ่งทรัพยากรจากคอมพิวเตอร์หลายๆ เครื่องเพื่อมาช่วยในการประมวลผลโดยเชื่อมต่อกันด้วยเครือข่ายคอมพิวเตอร์ ช่วยในการทำงานหรือแก้ปัญหาต่างๆ ที่มีความยุ่งยากและซับซ้อน

การเกิดเทคโนโลยีการประมวลผลบนกลุ่มเมฆส่งผลให้เกิดประโยชน์ทางด้านธุรกิจเป็นอย่างมาก สามารถทำให้ธุรกิจใช้ทรัพยากรน้อยลงอย่างเห็นได้ชัด ทั้งค่าใช้จ่ายและเวลาในการพัฒนาแอพพลิเคชั่น ที่ผ่านมาในการดำเนินธุรกิจหรือต้องการใช้แอพพลิเคชั่นใดๆ เข้ามาช่วย จะมีการใช้แอพพลิเคชั่นที่เป็นที่รู้จัก เช่น SaaP (Service as a Product) เป็นต้น ซึ่งต้องมีค่าใช้จ่ายจำนวนมาก ไม่ว่าจะเป็นค่าไฟฟ้า ค่าทรัพยากรต่างๆ ค่าบุคลากร ค่าเครือข่าย ค่าเซิร์ฟเวอร์ และต้องมีการวางแผนในการพัฒนา ดำเนินการและแก้ไขข้อผิดพลาดที่อาจจะเกิดขึ้นระหว่างกระบวนการ นอกจากนี้เมื่อแอพพลิเคชั่นต้องการการอัพเดทเพื่อรองรับการทำงานก็อาจจะเกิดปัญหาขึ้นได้ ปัจจุบันการประมวลผลบนกลุ่มเมฆจึงเข้ามามีบทบาทสำคัญเพราะนอกจากจะใช้ทรัพยากรและค่าใช้จ่ายน้อยกว่ากันแล้ว การประมวลผลบนกลุ่มเมฆยังมีความยืดหยุ่นมากกว่าอีกด้วย จากที่กล่าวมาเบื้องต้นเป็นเพียงตัวอย่างบริการที่เกิดขึ้นจากเทคโนโลยีการประมวลผลบนกลุ่มเมฆ ที่ชื่อว่า Enterprise Cloud Computing ซึ่งก็มีอีกหลายบริการ โดยบริการต่าง ๆ ที่เกิดขึ้นบนการประมวลผลบนกลุ่มเมฆ สามารถแบ่งออกได้เป็น 3 จำพวกใหญ่ ๆ ได้แก่

2.1.1 Infrastructure as a Service (IaaS)

รูปแบบของ IaaS คือ มีบุคคลที่สามให้บริการทรัพยากรต่างๆ เช่น Hardware, Software, Servers, แหล่งเก็บข้อมูล หรือส่วนประกอบอื่นๆ ในโครงสร้างพื้นฐานของระบบสารสนเทศ โดยจะอนุญาตให้ผู้ที่ต้องการใช้งานและเข้าถึงผ่านอินเตอร์เน็ต ซึ่งมีความสามารถในการปรับตัวให้มีความเหมาะสมกับงาน รวมถึงง่ายต่อการดูแลและสำรองข้อมูลอีกด้วย ผู้ให้บริการ IaaS ที่มีชื่อเสียงในปัจจุบัน ได้แก่ Amazon Web Services (AWS), Windows Azure, Google Compute Engine, Rackspace Open Cloud, และ IBM SmartCloud Enterprise

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 Platform as a Service (PaaS)

รูปแบบของการให้บริการแบบ PaaS คือการให้บริการ Platform ให้กับลูกค้าหรือผู้ที่ต้องการใช้บริการ พัฒนา จัดการ หรือรันแอพพลิเคชั่นต่างๆ โดยไม่ต้องผ่านความวุ่นวายในการเตรียมทรัพยากรหรือสภาพแวดล้อมที่จำเป็นในการพัฒนาซึ่งสามารถเปลี่ยนหรืออัพเกรดเองได้ อีกทั้งยังสนับสนุนการทำงานร่วมกันของทีมผู้พัฒนาแอพพลิเคชั่นอีกด้วย โดยทั่วไปจะสามารถเข้าถึงบริการ PaaS ผ่านหน้า Web Browser ซึ่ง Platform ที่กล่าวถึงก็จะทำงานบนโครงสร้าง (Infrastructure) ของผู้ให้บริการ เช่น หนึ่งในผู้ขาย PaaS คือ Salesforce.com ที่ให้บริการ Enterprise Customer Relationship Management (CRM) Platform, หรือ Platform อื่นๆที่ใช้ในการพัฒนาและจัดการซอฟต์แวร์ ได้แก่ Appear IQ, Mendix, Amazon Web Services (AWS) Elastic Beanstalk, Google App Engine, และ Heroku

2.1.3 Software as a Service (SaaS)

การให้บริการในรูปแบบของ SaaS ได้เป็นที่แพร่หลายมากขึ้นในปัจจุบัน โดยจะทำงานร่วมกับเทคโนโลยี Web Services และ Services-Oriented Architecture (SOA) ที่จะเน้นการพัฒนาแบบอ้างอิงสถาปัตยกรรมที่จะมีพื้นฐานมาจากการให้บริการต่อๆ ไป ซึ่งการใช้บริการแบบ SaaS จะเป็นการกระจายตัวซอฟต์แวร์ไปยังผู้ขายหรือผู้ให้บริการเพื่อให้ลูกค้าสามารถเข้าถึงซอฟต์แวร์ผ่านอินเตอร์เน็ตได้ ทำให้มีการดูแลได้ง่าย สามารถอัพเดทซอฟต์แวร์ได้อัตโนมัติ ทำให้ผู้ใช้ทุกคนเข้าถึงซอฟต์แวร์ที่เป็นเวอร์ชั่นเดียวกันได้ นอกจากนี้ยังสนับสนุนการทำงานร่วมกันเป็นไปอย่างง่ายดาย และสามารถเข้าถึงตัวซอฟต์แวร์ได้จากที่ใดก็ได้

2.2 แหล่งเก็บข้อมูลบนกลุ่มเมฆ (Cloud Storage)

การเก็บข้อมูลในปัจจุบันจะมีการเก็บไว้ในแหล่งเก็บข้อมูลที่เป็นในส่วนท้องถิ่น เช่น เก็บข้อมูลภายในฮาร์ดดิสก์ เป็นต้น มีข้อดีคือสามารถเข้าถึงได้ง่ายและรวดเร็ว แต่ก็สามารถชำรุดหรือสูญหายได้ง่ายเช่นกัน ซึ่งหลังจากที่ได้รู้จักกับเทคโนโลยีการประมวลผลบนกลุ่มเมฆ จึงได้เกิดการบริการแหล่งเก็บข้อมูลบนกลุ่มเมฆที่จะเป็นโมเดลในการเก็บรักษา จัดการและสำรองข้อมูลผ่านเครือข่าย ซึ่งส่วนมากจะทำงานผ่านอินเตอร์เน็ต และมีข้อดีในการเข้าถึงข้อมูลได้ง่ายจากทุกที่ ทุกเวลา และจากทุกอุปกรณ์

2.3 เว็บไซต์

เว็บไซต์ คือ ส่วนที่รวบรวมและแสดงเว็บเพจ (Web pages) โดยต้องเรียกผ่าน URL (Uniform Resource Locator) และมีความนิยมเป็นอย่างมาก ซึ่งในการพัฒนาเว็บไซต์ได้จำเป็นต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีความรู้พื้นฐานเกี่ยวกับ HTML (Hyper Text Markup Language), CSS (Cascading Style Sheet) และเพื่อการพัฒนาเว็บไซต์ที่สามารถทำได้ซับซ้อนมากขึ้น จึงจำเป็นต้องมีความรู้เกี่ยวกับเรื่องดังต่อไปนี้

2.3.1 JavaScript

ในการสร้าง Web Page ที่มีการประมวลผลที่ซับซ้อน จะมีการใช้ภาษา JavaScript ประกอบในการพัฒนา โดยจะใช้ไปพร้อมกับ HTML และ CSS จะทำให้มีการประมวลผลในฝั่งของผู้ใช้งาน (Client-Side) โดยจะเป็นหน้าที่ของ Web Browser ที่ทำการประมวลผล เพื่อเพิ่มการตอบสนองต่อเหตุการณ์ต่างๆได้ เว็บไซต์จะทำงานได้หลากหลายมากยิ่งขึ้น

2.3.2 PHP

PHP คือ ภาษาที่ใช้ในการพัฒนาเว็บไซต์ เป็นภาษาที่เป็น Script และเขียนในรูปแบบของ Interpreter โดยจะทำหน้าที่ประมวลผลและทำหน้าที่ในฝั่งของเซิร์ฟเวอร์ (Server-Side) และเป็นภาษาที่มีความนิยม สามารถทำงานร่วมกับ HTML ได้ ช่วยทำให้เว็บไซต์รับรองการเปลี่ยนแปลงของข้อมูลตามที่ต้องการได้

2.3.3 Composer

ในการพัฒนาเว็บไซต์ที่เป็นภาษา PHP นั้น บางครั้งจำเป็นต้องใช้ ส่วนเสริมหรือ Library ที่เราอาจจะนำมาใช้ต่อในการพัฒนา และด้วยความยุ่งยากของการเรียกใช้และติดตั้ง จึงมีเครื่องมือที่ชื่อ Composer มาช่วยอำนวยความสะดวกในการเรียกใช้และจัดการกับความซ้ำซ้อน อีกทั้งยังเป็นแหล่งรวบรวม Library ให้กับเหล่าผู้พัฒนาอีกด้วย

2.4 Application Program Interface (API)

ในการพัฒนาโปรแกรมหรือเว็บไซต์อาจจะต้องมีการติดต่อกันระหว่างตัว Application ซึ่งจะมี API จะทำหน้าที่ในการระบุขั้นตอนและวิธีในการติดต่อกันระหว่างทั้งสอง Application ทำให้มีการ บูรณาการและทำงานร่วมกัน ซึ่งส่วนมากจะมาในรูปแบบของ Library ถ้ามีการพัฒนา API ที่ดี จะทำให้มีการนำไปใช้ต่อได้ง่ายและไม่ซับซ้อน

2.5 JavaScript Object Notation (JSON)

JSON เป็นอีกหนึ่งรูปแบบใช้ในการแลกเปลี่ยนข้อมูล มีขนาดเล็ก ทำงานได้รวดเร็ว สามารถอ่านได้เข้าใจทั้งจากมนุษย์และคอมพิวเตอร์ สามารถสร้างและส่งต่อได้ง่ายทำให้ทำงานได้เร็ว สามารถเก็บข้อมูลที่เป็น String, Number, Boolean, Array หรือแม้แต่ Object ก็สามารถเก็บได้ โดยรูปแบบในการเก็บจะเป็นการเก็บที่มี Key และ Value ที่สัมพันธ์กัน ทำให้สามารถเข้าถึงได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 Representational State Transfer (REST)

REST คือ รูปแบบของสถาปัตยกรรม ที่ได้ออกแบบมาเพื่อแอพพลิเคชั่นที่ต้องการทำงานบนเครือข่ายคอมพิวเตอร์ เพื่อที่จะหาวิธีทำงานที่ง่ายกว่ากลไกอื่นๆ เช่น Common Object Request Broker Architecture (CORBA) หรือ Remote Procedure Call (RPC) เป็นต้น ใช้ในติดต่อสื่อสารแลกเปลี่ยนข้อมูลภายในเครือข่ายคอมพิวเตอร์ ตัวอย่างของการพัฒนาตามหลักสถาปัตยกรรมแบบ REST ที่เห็นได้ชัดที่สุดคือ

- HTTP Protocol ที่มีการทำงานในรูปแบบของ Client-Server มีการ Request-Respond ข้อมูลกัน

- Statelessในปัจจุบันหรือก็คือไม่พึ่งพากับ Request ใดๆ ก่อนหน้า

- Cacheable หมายถึงการเก็บข้อมูลสำรองไว้ได้ตามทางที่ผ่าน

ซึ่งสถาปัตยกรรมแบบ REST จะมีการทำงานที่ง่าย ไม่ซับซ้อนและตรงไปตรงมา เป็นอิสระต่อ Platform และภาษาในการพัฒนา และนอกจากนี้การพัฒนาวิธีการติดต่อสื่อสารแบบ REST ยังทำให้เห็นว่าสามารถทำงานได้ครบถ้วนเช่นเดียวกับวิธีอื่นๆ อีกด้วย

2.7 cURL

cURL คือชื่อโครงงานซอฟต์แวร์หนึ่ง ที่มีลักษณะเป็น Library โดยสามารถอ่านออกเสียงเป็น “see URL” ได้ แต่ว่าจริงๆแล้วเป็นตัวย่อที่มาจาก “Client URL Request Library” ซึ่ง cURL นั้นประกอบด้วยสองอย่างคือ libcurl และ curl โดยจะทำงานร่วมกัน ซึ่งเป็น Library ที่สามารถเรียกใช้ในภาษา PHP เช่นกัน ใช้เพื่อสร้าง HTTP Request และอ่าน HTTP Response ที่ได้กลับมาแต่ไม่ได้นำมาประมวลแสดงเหมือนกับ Web-Browser จึงใช้เขียนในการทำงานในส่วนของ Back-End มากกว่า

2.8 OAuth

OAuth หรือก็คือมาตรฐานสำหรับมอบสิทธิหรือทำให้ Application ที่เป็นบุคคลที่สามสามารถได้รับสิทธิหรือการเข้าถึงข้อมูลได้อย่างปลอดภัยและยังสามารถกระทำการแทนผู้ที่อนุญาตสิทธิให้กับ Application โดยที่ไม่เปิดเผยข้อมูลประจำตัวอีกด้วย ซึ่งจะมีการแบ่งปันสัญลักษณ์หรือเครื่องหมายในการเข้าถึงได้ (access token) เพื่อใช้ในการยืนยันตัวตน โดยสามารถเรียกใช้ผ่าน HTTP/HTTPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 MVC Architectural Pattern

รูปแบบสถาปัตยกรรมแบบ MVC หรือ Model-View-Controller เป็นรูปแบบในการออกแบบและพัฒนาซอฟต์แวร์แบบหนึ่ง โดนจะแบ่งการทำงานของซอฟต์แวร์ออกเป็น 3 ส่วน ได้แก่ Model, View และ Controller ซึ่งรูปแบบการออกแบบแบบ MVC ได้มีความนิยมในการออกแบบเว็บแอพลิเคชั่นเป็นอย่างมาก โดย MVC นั้นมีจุดเริ่มต้นมาจากการออกแบบระบบที่ต้องมีการทำงานร่วมกับ Graphical User Interface (GUI)

ส่วนประกอบหลักที่สำคัญของหลักการการออกแบบในรูปแบบของ MVC คือ Model ที่จะทำหน้าที่จำลองในส่วนของปัญหาขึ้นมา ทำหน้าที่จัดการข้อมูล หลักตรรกะและกฎเกณฑ์ต่างๆ โดยตรง มีการกระทำที่ชัดเจนโดยแก้ไขและเปลี่ยนแปลงข้อมูลโดยตรง ส่วนถัดมาคือ View ที่ทำหน้าที่แสดงผลข้อมูลที่ได้รับมา อาจจะแสดงในรูปแบบใดๆ ก็ได้ และส่วนที่คอยทำหน้าที่สั่งให้ Model หรือ View ทำงาน หรือทำหน้าที่ควบคุม สั่งการ คือ Controller ซึ่งจะทำการประสานงานและข้อมูลระหว่าง Model และ View ดังรูปที่ 2.1

รูปที่ 2.1 ภาพจำลองรูปแบบ MVC

(ที่มา: http://perl-diving.blogspot.com/p/blog-page.html)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 Nested Set [1]

Nested Set คือ เทคนิคหนึ่งในการเก็บข้อมูลประเภท Trees หรือ Hierarchies ไว้ใน relational database มีหลักการการทำงานโดยจะเก็บ node ต่างๆ ไว้และเก็บจำนวนขนาดของฝั่งซ้ายและขวาเอาไว้ เพื่อเปรียบเทียบ โดยแสดงได้ดังรูปที่ 2.2 ซึ่งมีประสิทธิภาพในการดึงข้อมูลออกมาได้ค่อนข้างเร็ว ถือได้ว่าการเก็บข้อมูลแบบ Nested Set จะสามารถ query ข้อมูลออกมาได้เร็วกว่า Store procedure แต่ทว่าก็มีข้อเสียเช่นกัน เนื่องจากต้องมีการแก้ไขเลขด้านซ้ายและขวาอย่างสม่ำเสมอ ทำให้การเพิ่มข้อมูลใหม่ ส่งผลกระทบต่อข้อมูลเก่าและต้องมีการแก้ไข จึงถือได้ว่าช้าและเป็นข้อเสียอย่างหนึ่งของ Nested Set

ระหว่างการพัฒนา ผู้จัดทำได้นำเทคนิคมาประยุกต์ใช้ในการเก็บข้อมูลเนื่องจากลักษณะของข้อมูลเป็นลำดับชั้นและตรงตามคุณลักษณะที่จะสามารถแก้ไขได้ด้วยเทคนิค Nested Set เช่น รายชื่อไฟล์และโฟลเดอร์ รวมถึงไฟล์ภายในโฟลเดอร์ต่างๆ เป็นลำดับชั้น เป็นต้น และการเก็บข้อมูลด้วย Nested Set นั้นส่งผลให้เราสามารถนำข้อมูลมาประมวลผลได้สะดวกมากยิ่งขึ้น อีกทั้งยังสามารถรักษาความถูกต้องของข้อมูลภายในระบบไว้ได้เป็นอย่างดี ด้วยการออกแบบ Relational Database ที่ไม่ส่งผลให้เกิดความซ้ำซ้อนหรือความผิดพลาดของข้อมูลอีกด้วย

รูปที่ 2.2 ภาพจำลองการเก็บข้อมูลแบบ Nested Set

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การวิเคราะห์และออกแบบระบบ

3.1 ศึกษาระบบงานเดิม

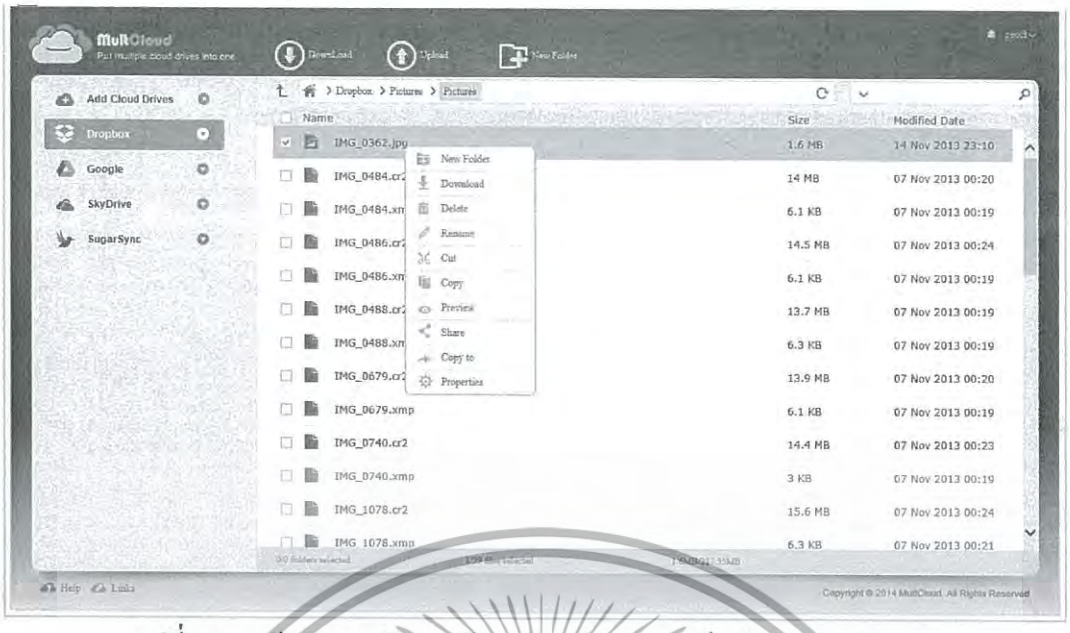
รูปที่ 3.1 หน้าเว็บไซต์ MultCloud

(ที่มา: https://multcloud.com)

จากที่ได้ศึกษาเบื้องต้นเกี่ยวกับปัญหาความซ้ำซ้อนและยุ่งยากในการจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆหลายๆ บริการพร้อมกัน จึงได้พบกับระบบงานที่มีอยู่แล้วอยู่ในรูปแบบของเว็บไซต์ https://www.multcloud.com/ ชื่อว่า Multcloud [2] ตามรูปที่ 3.1 ซึ่งมี Feature ที่สำคัญได้แก่

- สามารถจัดการข้อมูลในแหล่งเก็บข้อมูลบนกลุ่มเมฆได้พร้อมกัน
- ย้ายไฟล์และข้อมูลข้าม Cloud Storage ได้
- กำหนดช่วงเวลาในการย้ายไฟล์ต่างๆ ข้าม Cloud Storage ได้
- สามารถการันตีเรื่องความปลอดภัยของข้อมูลได้
- เว็บไซต์ MultCloud รับรอง Cloud Storage จำนวนมาก ได้แก่ Dropbox, Google Drive, Amazon Drive, OneDrive, Amazon S3, Box, MediaFire, OwnCloud, SugarSync, Copy, Alfresco, WebDav, CloudMe, Cubby, MyDrive, WEB.DE, Yandex, HiDrive, ADrive และ Baidu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 หน้าต่างการจัดการข้อมูลในแต่ละแหล่งเก็บข้อมูลบนกลุ่มเมฆ

(ที่มา: https://multcloud.com)

จากรูปที่ 3.2 จะเห็นได้ว่า ในเว็บไซต์ MultCloud จะมีหน้าต่าง Explorer ด้านซ้าย เพื่อเป็นตัวจัดการไฟล์ที่เก็บไว้ในแต่ละ Cloud Storage ซึ่งมีลักษณะการวางที่เป็นระเบียบและแยกจากกันอย่างชัดเจน รวมถึงมีการออกแบบที่ดูสะอาดตา ใช้งานง่ายและมีความใกล้เคียงกับ Explorer ที่เราได้พบเจอบ่อยๆ เช่น Windows Explorer เป็นต้น

3.2 ปัญหาที่พบในระบบปัจจุบัน

จากปัญหาที่พบได้ในเว็บไซต์ MultCloud คือ จะมีการแยกผู้ให้บริการออกจากกันอย่างชัดเจน ไม่ได้มองแหล่งเก็บข้อมูลบนกลุ่มเมฆทั้งหมดรวมกันเป็นแหล่งเดียว แต่ด้วยไอเดียของเว็บไซต์ GatherCloud จะมีใช้แนวคิดในการมองแหล่งเก็บข้อมูลหลายๆบริการเป็นเพียงบริการเดียวและสามารถทำงานด้วยกันได้โดยใช้รูปแบบการจัดการที่ดีกว่าเพื่ออำนวยความสะดวกให้กับผู้ใช้ให้มากที่สุด

นอกจากนี้ MultCloud ยังมีบางฟังก์ชั่นที่ต้องเสียค่าบริการ จึงจะใช้งานได้ ซึ่งอาจจะทำให้ผู้ใช้บางคนที่จ่ายเงินค่าบริการให้กับผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆไปแล้ว ต้องเสียเงินเพิ่มให้กับฟังก์ชั่นการใช้งานเพิ่มเติมของ MultCloud อีก ซึ่งมีรายละเอียดตามตารางต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 รายละเอียดของฟังก์ชั่นฟรีและฟังก์ชั่นที่ต้องเสียค่าใช้จ่ายของเว็บไซต์ MultCloud

Features	**Free Account**	**Premium Account**
File Manager	Yes	Yes
Cross-Cloud File Transfer	Yes	Yes
Cloud Drive Quantity	Countless	Countless
Data Traffic	Less than 10 TB	Unlimited Traffic
Scheduled File Transfer	No	Yes
File Transfer Filter	No	Yes
Price	$ 0.00	Monthly : $4.99 Yearly : $42.00

3.3 การวิเคราะห์ความต้องการ

3.3.1 ความต้องการที่เป็นหน้าที่หลัก (Functional Requirement)

- ผู้ใช้สามารถสมัครสมาชิกกับเว็บไซต์ GatherCloud
- ผู้ใช้สามารถแก้ไขข้อมูลส่วนตัวที่ลงทะเบียนไว้กับ GatherCloud ได้
- ผู้ใช้สามารถเลือกการเชื่อมต่อกับแหล่งเก็บข้อมูลบนกลุ่มเมฆได้มากกว่าหนึ่งการเชื่อมต่อ โดยจะเชื่อมต่อเพียงครั้งเดียวและไม่จำเป็นต้องเชื่อมต่ออีก
- เว็บไซต์ GatherCloud จะรองรับการเชื่อมต่อกับแหล่งเก็บข้อมูลบนกลุ่มเมฆดังนี้ Google Drive, OneDrive, Box, Dropbox และ Copy
- ผู้ใช้สามารถอัพโหลด ดาวน์โหลด ลบ แก้ไข และแชร์ไฟล์หรือข้อมูลจากแหล่งเก็บข้อมูลบนกลุ่มเมฆได้
- ผู้ใช้สามารถค้นหาไฟล์หรือข้อมูลจากแหล่งเก็บข้อมูลบนกลุ่มเมฆได้
- เว็บไซต์ GatherCloud จะต้องมีความสามารถแจ้งเตือน (Notification) ภายในหน้าเว็บไซต์ได้
- เว็บไซต์ GatherCloud จะสามารถแสดงรายละเอียดเกี่ยวกับแหล่งเก็บข้อมูลบนกลุ่มเมฆ และรายละเอียดเกี่ยวกับไฟล์ที่ถูกเก็บไว้ได้
- ผู้ใช้สามารถย้ายไฟล์ข้ามแหล่งเก็บข้อมูลบนกลุ่มเมฆได้
- ผู้ใช้สามารถแชร์สิทธิในการมองเห็นหรือเข้าถึงไฟล์และข้อมูลตามที่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 ความต้องการที่ไม่ใช่หน้าที่หลักของระบบ (Non-Functional Requirement)

- เว็บไซต์ GatherCloud จะต้องมีความปลอดภัยและรักษาความเป็นส่วนตัวของข้อมูลที่ผู้ใช้เก็บไว้ได้

3.4 การวิเคราะห์และวิจารณ์ระบบที่ต้องการออกแบบ

3.4.1 จุดประสงค์ของโครงงาน

โครงงานตั้งใจจะทำขึ้นมาเพื่อตอบสนองปัญหาในการใช้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆหลายแห่ง ทำให้มีความยุ่งยากเมื่อต้องการจะค้นหาไฟล์บางไฟล์ หรือจะจัดการไฟล์ข้ามบริการแหล่งเก็บข้อมูลบนกลุ่มเมฆ โดยจะมีแนวทางในพัฒนา คือ พัฒนาเว็บไซต์ที่แสดงให้ผู้ใช้เห็นว่า ผู้ใช้มีผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆเพียงอันเดียว แต่โดยแท้จริงแล้วเกิดจากการรวบรวมแหล่งเก็บข้อมูลบนกลุ่มเมฆหลายบริการเข้าไว้ด้วยกันและอาศัยเทคนิคและการจัดการที่ดีเพื่อให้เกิดความสะดวกสบายแก่ผู้ใช้งาน ทำให้จัดการได้ง่าย ใช้งานง่าย และทำให้การใช้พื้นที่เก็บข้อมูลบนแหล่งเก็บข้อมูลบนกลุ่มเมฆเป็นไปอย่างคุ้มค่ามากที่สุด เพื่อที่จะช่วยประหยัดค่าใช้จ่ายของผู้ใช้ และลดความแปลืองพื้นที่ของการเก็บข้อมูลที่เกิดขึ้นกับผู้ให้บริการ นอกจากนี้ยังมุ่งหมายให้ผู้คนทั่วไปหันมาสนใจและใช้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆมากขึ้นอีกด้วย

3.4.2 เหตุผลในการเลือกใช้อุปกรณ์และเครื่องมือต่างๆในการพัฒนา

3.4.2.1 ภาษา PHP

ภาษา PHP เป็นภาษาที่เป็นนิยมมากในการเขียนเว็บไซต์ เพราะเป็นภาษาที่เขียนและเข้าใจได้ง่าย ทำงานได้เร็ว นอกจากนี้เนื่องจาก GatherCloud เป็นเว็บไซต์ที่ต้องการทำงานร่วมกับ Library อื่นๆเพื่อจะเข้าถึงแหล่งเก็บข้อมูลบนกลุ่มเมฆจึงเลือกที่จะพัฒนาด้วยภาษา PHP ที่เป็นนิยมและมี Library สนับสนุนจำนวนมาก อีกทั้งผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆจะสนับสนุนภาษา PHP เป็นหลักอีกด้วย

3.4.2.2 API

1) Copy API สาเหตุที่เลือกพัฒนาให้เว็บไซต์สามารถรองรับ Copy API ได้ ก็เพราะ Copy เป็นหนึ่งในผู้ให้บริการ Cloud Storage ที่มีชื่อเสียงและเป็นที่รู้จัก และบริการที่ Copy ได้นำเสนอก็ถือว่าเป็นการบริการที่มีคุณภาพ เชื่อถือได้ และใช้งานได้สะดวก

2) OneDrive API OneDrive นั้นเป็นการเปลี่ยนชื่อมาจาก SkyDrive ซึ่งถูกพัฒนาโดย Microsoft และปัจจุบันในสมัยนี้ ได้มีการผูกข้อมูลต่างๆใน Platform วินโดวส์กับ Cloud Storage ทำให้มีผู้คนเลือกที่จะใช้ OneDrive ที่มาพร้อมกับระบบปฏิบัติการวินโดวส์ไปในตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยเฉพาะ Windows 10 อีกด้วย และด้วยจากการพิจารณาคาดได้ว่า OneDrive เป็นผู้ให้บริการที่มีความยั่งยืนจึงได้คัดเลือกมาพัฒนา

3) Google Drive API ในปัจจุบัน Google ได้เป็นหนึ่งในองค์กรทางเทคโนโลยีที่เป็นที่รู้จักและเป็นผู้นำทางด้านเทคโนโลยีใหม่ๆ โดยเป็นที่รู้จักมาจาก Google Search Engine ต่อมาได้พัฒนาแหล่งเก็บข้อมูลบนกลุ่มเมฆชื่อว่า Google Drive และเป็นที่นิยมเป็นอย่างมาก เพราะมีการผูกกับบริการอื่นๆ และเป็นที่ฝากเก็บไฟล์ที่ดี สนับสนุนให้มีการทำงานร่วมกัน สามารถแก้ไขไฟล์ได้ร่วมกัน ทำให้ Google Drive เป็นทางเลือกที่ดีในการพัฒนาให้สามารถทำงานร่วมกับ GatherCloud ได้ นอกจากนี้ยังมีความสัมพันธ์ผูกกับระบบปฏิบัติการ Android อีกด้วย

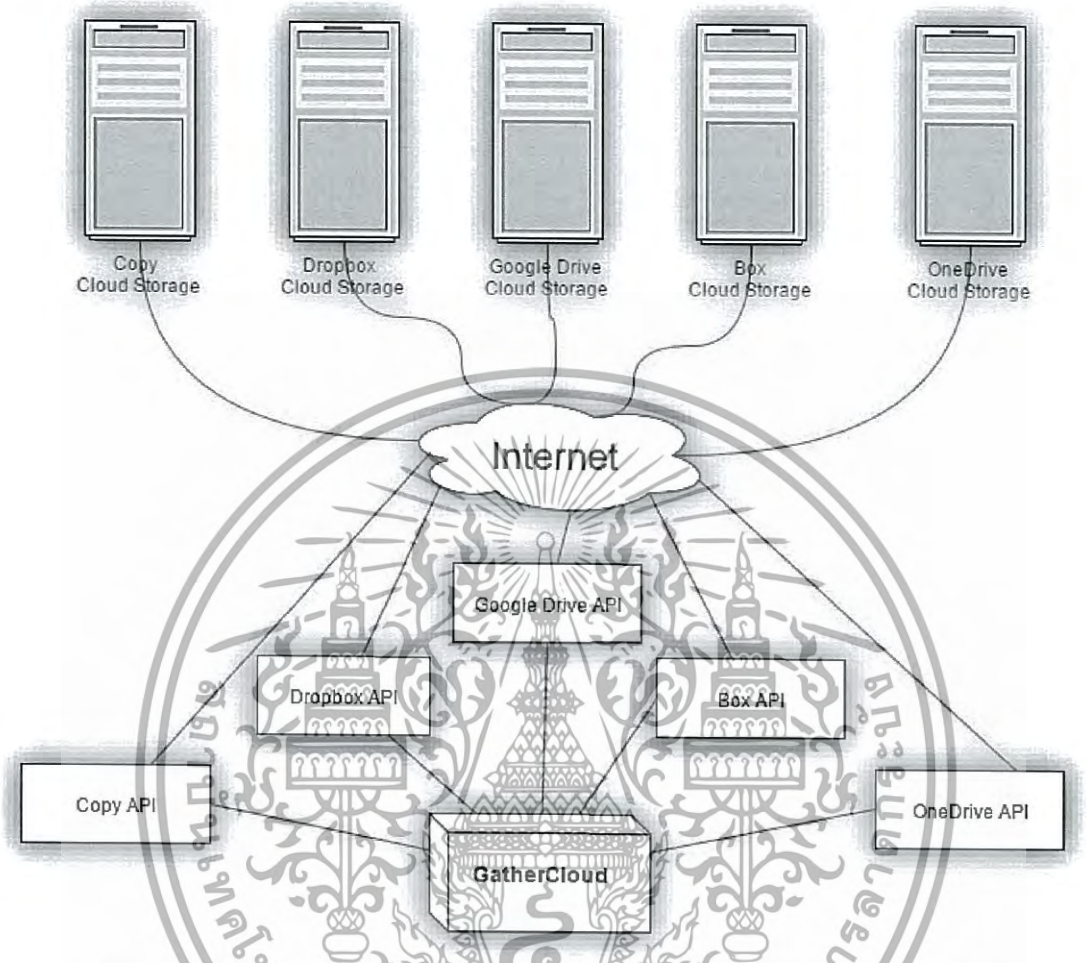
4) Box API ผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆที่ชื่อว่า Box นั้นเป็นผู้ให้บริการที่มีการบูรณาการร่วมกับแอพพลิเคชั่นอื่นๆเป็นจำนวนมาก จึงเลือกที่จะนำมาศึกษาเพราะง่ายต่อการพัฒนาให้สามารถทำงานร่วมกันกับแหล่งเก็บข้อมูลบนกลุ่มเมฆ

5) Dropbox API Dropbox เป็นหนึ่งในผู้ที่ให้บริการแหล่งเก็บข้อมูลบนก้อนเมฆแรกๆ จึงควรแก่การนำมาพัฒนาเพราะฐานผู้ใช้มีจำนวนมากและมีนักพัฒนาอื่นๆ นำไปพัฒนาร่วมกับอีกหลายๆแอพพลิเคชั่นอีกด้วย Dropbox จึงเป็นทางเลือกที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การออกแบบระบบใหม่

3.5.1 บล็อกไดอะแกรม (Block Diagram)



รูปที่ 3.3 แผนภาพ Block Diagram ของโครงสร้างการทำงานของ GatherCloud

GatherCloud ทำหน้าที่เป็นตัวกลางสำหรับติดต่อ API ทั้ง 5 ซึ่งส่งคำสั่งการทำงานผ่านอินเตอร์เน็ตไปยังเซิร์ฟเวอร์ของแหล่งเก็บข้อมูลบนกลุ่มเมฆเพื่อดึงข้อมูลที่ต้องการมาใช้งาน ดังแผนภาพที่แสดงในรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2 ยูสเคสโมเดล

3.5.2.1 ผู้ที่เกี่ยวข้องในระบบ (Actor)

- ผู้ใช้งาน

3.5.2.2 ฟังก์ชั่นการทำงานหลัก

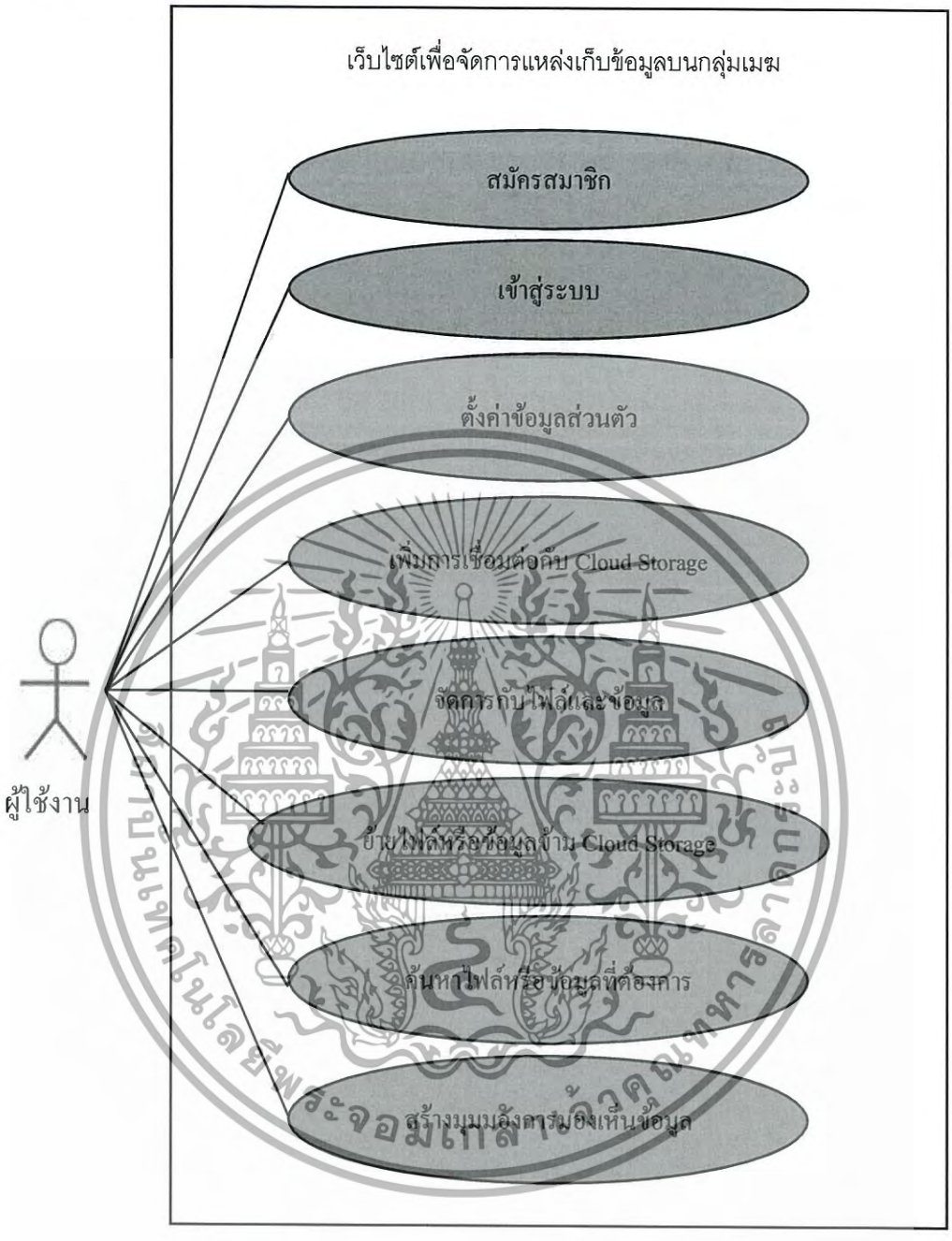
- สมัครสมาชิก
- เข้าสู่ระบบ
- ตั้งค่าข้อมูลส่วนตัว
- เพิ่มการเชื่อมต่อกับ Cloud Storage
- จัดการกับไฟล์และข้อมูล
- ย้ายไฟล์หรือข้อมูลข้าม Cloud Storage
- ค้นหาไฟล์หรือข้อมูลที่ต้องการ
- สร้างมุมมองการมองเห็นข้อมูล

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.5.2.3 แผนภาพยูสเคส (Use Case Diagram)



รูปที่ 3.4 แผนภาพยูสเคสของ GatherCloud

146232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2.4 รายละเอียดของยูสเคส (Use Case Description)

a) รายละเอียดการทำงานของฟังก์ชั่นการสมัครสมาชิก

ตารางที่ 3.2 รายละเอียดการทำงานของฟังก์ชั่นการสมัครสมาชิก

Use Case Name:	สมัครสมาชิก	**ID:** 1
Triggering Event:	-	
Brief Description:	ผู้ใช้งานสมัครสมาชิกเพื่อจะเป็นสมาชิกเว็บไซต์	
Actors:	ผู้ใช้งาน	
Related Use Case:	-	
Stakeholder:	-	
Pre-Condition:	-	
Post-Condition:	-	
Flow of Events	1. ผู้ใช้กดปุ่มสมัครสมาชิก 2. ผู้ใช้กรอกข้อมูล 3. ผู้ใช้ยืนยันการสมัครสมาชิก	
Alternative Flows	-	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b) รายละเอียดการทำงานของฟังก์ชั่นการเข้าสู่ระบบ

ตารางที่ 3.3 รายละเอียดการทำงานของฟังก์ชั่นการเข้าสู่ระบบ

Use Case Name:	เข้าสู่ระบบ	**ID:** 2
Triggering Event:	-	
Brief Description:	ผู้ใช้งานเข้าสู่ระบบเพื่อสามารถเรียกใช้ฟังก์ชั่นอื่นๆได้	
Actors:	ผู้ใช้งาน	
Related Use Case:	-	
Stakeholder:	-	
Pre-Condition:	-	
Post-Condition:	-	
Flow of Events	4. ผู้ใช้งานต้องการเข้าสู่ระบบ 4.1. ใส่ E-mail และรหัสผ่าน 4.2. กด "เข้าสู่ระบบ" 5. ระบบตรวจสอบการเข้าสู่ระบบ 6. ระบบแสดงผลการเข้าสู่ระบบสำเร็จ	
Alternative Flows	A1 ผู้ใช้งานใส่ E-mail หรือรหัสผ่านไม่ถูกต้อง 1. ระบบแจ้งเตือนว่ากรอกข้อมูลไม่ถูกต้อง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

c) รายละเอียดการทำงานของฟังก์ชั่นตั้งค่าข้อมูลส่วนตัว

ตารางที่ 3.4 รายละเอียดการทำงานของฟังก์ชั่นตั้งค่าข้อมูลส่วนตัว

Use Case Name:	ตั้งค่าข้อมูลส่วนตัว	**ID:** 3
Triggering Event:	-	
Brief Description:	ผู้ใช้งานตั้งค่าข้อมูลส่วนตัว	
Actors:	ผู้ใช้งาน	
Related Use Case:	-	
Stakeholder:	-	
Pre-Condition:	ผู้ใช้งานต้องเข้าสู่ระบบสำเร็จแล้ว	
Post-Condition:	-	
Flow of Events	1. ผู้ใช้งานเลือก "ตั้งค่า" 1.1. กรอกข้อมูล 1.2. กด "บันทึก" 2. ระบบบันทึกข้อมูลที่กรอกเข้ามาใหม่ 3. ระบบแสดงผลว่าการบันทึกสำเร็จเรียบร้อย	
Alternative Flows	-	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

d) รายละเอียดการทำงานของฟังก์ชั่นเพิ่มการเชื่อมต่อกับ Cloud Storage

ตารางที่ 3.5 รายละเอียดการทำงานของฟังก์ชั่นเพิ่มการเชื่อมต่อกับ Cloud Storage

Use Case Name:	เพิ่มการเชื่อมต่อกับ Cloud Storage	**ID:** 4
Triggering Event:	-	
Brief Description:	ผู้ใช้งานสร้างการเชื่อมต่อและอนุญาติให้ระบบเข้าถึง Cloud Storage ของผู้ใช้งานได้	
Actors:	ผู้ใช้งาน	
Related Use Case:	-	
Stakeholder:	-	
Pre-Condition:	ผู้ใช้งานต้องเข้าสู่ระบบสำเร็จแล้ว	
Post-Condition:	ผู้ใช้งานจะต้องยินยอมให้ระบบสามารถเข้าถึงข้อมูลใน Cloud Storage ได้	
Flow of Events	1. ผู้ใช้เลือก Cloud Storage ที่ต้องการจะเชื่อมต่อ 2. กำหนดชื่อของการเชื่อมต่อ 3. ผู้ใช้ยินยอมให้ระบบเข้าถึงข้อมูลใน Cloud Storage 4. ระบบอัพเดทการแสดงผลบนหน้าจอหลัก 5. ระบบเก็บข้อมูลเกี่ยวกับการเชื่อมต่อไว้ในฐานข้อมูล 6. ระบบแสดงรายชื่อไฟล์และข้อมูลที่อยู่ใน Cloud Storage	
Alternative Flows	A1 ผู้ใช้งานไม่ได้เข้าสู่ระบบของหน้าเว็บผู้ให้บริการ Cloud Storage 1. เข้าสู่ระบบด้วย E-mail และรหัสผ่านที่มี 2. ผู้ใช้ยินยอมให้ระบบเข้าถึงข้อมูลใน Cloud Storage 3. ระบบอัพเดทการแสดงผลบนหน้าจอหลัก 4. ระบบแสดงรายชื่อไฟล์และข้อมูลที่อยู่ใน Cloud Storage	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

e) รายละเอียดการทำงานของฟังก์ชั่นจัดการไฟล์และข้อมูล

ตารางที่ 3.6 รายละเอียดการทำงานของฟังก์ชั่นจัดการไฟล์และข้อมูล

Use Case Name:	จัดการไฟล์และข้อมูล	**ID:** 5
Triggering Event:	-	
Brief Description:	ผู้ใช้งานจัดการไฟล์และข้อมูลใน Cloud Storage	
Actors:	ผู้ใช้งาน	
Related Use Case:	-	
Stakeholder:	-	
Pre-Condition:	ผู้ใช้งานต้องเข้าสู่ระบบสำเร็จแล้ว	
Post-Condition:	-	
Flow of Events	1. ผู้ใช้เลือกฟังก์ชั่นการใช้งาน ได้แก่ 1.1. อัพโหลด 1.2. ดาวน์โหลด 1.3. แชร์ 1.4. แก้ไขชื่อ 2. ผู้ใช้ใส่ข้อมูลเพิ่มเติมที่จำเป็นให้แก่ระบบ 3. ระบบจัดการไฟล์และข้อมูล 4. ระบบแสดงผลว่าสำเร็จ	
Alternative Flows	A1 เกิดการทำงานผิดพลาดขณะระบบกำลังจัดการไฟล์และข้อมูล 1. ระบบแสดงผลว่าไม่สำเร็จ 2. ยกเลิกการกระทำทั้งหมด	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

f) รายละเอียดการทำงานของฟังก์ชั่นย้ายไฟล์หรือข้อมูลข้าม Cloud Storage

ตารางที่ 3.7 รายละเอียดการทำงานของฟังก์ชั่นย้ายไฟล์หรือข้อมูลข้าม Cloud Storage

Use Case Name:	ย้ายไฟล์หรือข้อมูลข้าม Cloud Storage	**ID:** 6
Triggering Event:	-	
Brief Description:	ผู้ใช้งานย้ายไฟล์หรือข้อมูลข้าม Cloud Storage	
Actors:	ผู้ใช้งาน	
Related Use Case:	-	
Stakeholder:	-	
Pre-Condition:	ผู้ใช้งานต้องเข้าสู่ระบบสำเร็จแล้ว	
Post-Condition:	-	
Flow of Events	1. ผู้ใช้เลือกฟังก์ชั่น "ย้ายโอน" 2. ผู้ใช้เลือกข้อมูลที่ต้องการจะย้าย 3. ผู้ใช้เลือก Cloud Storage ปลายทาง 4. ผู้ใช้กดยืนยัน 5. ระบบแสดงผลว่าสำเร็จ	
Alternative Flows	A1 เกิดการทำงานผิดพลาดขณะทำงาน 1. ระบบแสดงผลว่าไม่สำเร็จ 2. ยกเลิกการกระทำทั้งหมด A2 Cloud Storage ปลายทางพื้นที่ไม่พอ 1. ระบบแสดงผลแจ้งเตือนผู้ใช้ 2. ผู้ใช้เลือกข้อมูลที่ต้องการจะย้ายใหม่	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

g) รายละเอียดการทำงานของฟังก์ชั่นค้นหาไฟล์หรือข้อมูล

ตารางที่ 3.8 รายละเอียดการทำงานของฟังก์ชั่นค้นหาไฟล์หรือข้อมูล

Use Case Name:	ค้นหาไฟล์หรือข้อมูล	**ID:** 7
Triggering Event:	-	
Brief Description:	ผู้ใช้งานค้นหาไฟล์หรือข้อมูลที่มีในระบบ	
Actors:	ผู้ใช้งาน	
Related Use Case:	-	
Stakeholder:	-	
Pre-Condition:	ผู้ใช้งานต้องเข้าสู่ระบบสำเร็จแล้ว	
Post-Condition:	-	
Flow of Events	1. ผู้ใช้เลือกฟังก์ชั่น "ค้นหา" 2. ผู้ใช้กรอกข้อมูลที่ต้องการจะค้นหา 3. ระบบแสดงผลที่ได้จากการค้นหา	
Alternative Flows	A1 ระบบค้นหาไฟล์หรือข้อมูลไม่พบ 1. ระบบแสดงผลว่า หาไฟล์หรือข้อมูลที่ต้องการ ไม่พบ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

h) รายละเอียดการทำงานของฟังก์ชั่นสร้างมุมมองการมองเห็นข้อมูล

ตารางที่ 3.9 รายละเอียดการทำงานของฟังก์ชั่นสร้างมุมมองการมองเห็นข้อมูล

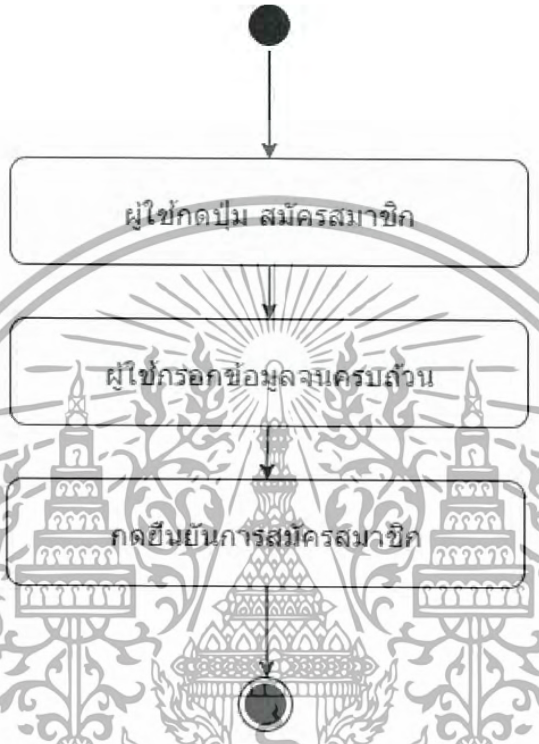
Use Case Name:	สร้างมุมมองการมองเห็นข้อมูล	**ID:** 8
Triggering Event:	-	
Brief Description:	ผู้ใช้งานสร้างมุมมองการมองเห็นข้อมูล	
Actors:	ผู้ใช้งาน	
Related Use Case:	-	
Stakeholder:	-	
Pre-Condition:	ผู้ใช้งานต้องเข้าสู่ระบบสำเร็จแล้ว	
Post-Condition:	-	
Flow of Events	1. ผู้ใช้เลือกฟังก์ชั่น "สร้างมุมมองของข้อมูล" 2. ผู้ใช้เลือกไฟล์ที่ต้องการอยู่ในมุมมอง 3. ผู้ใช้กรอกข้อมูลของ "มุมมอง" 4. ผู้ใช้กด ยืนยัน 5. ระบบแสดงผลว่าดำเนินการสำเร็จ	
Alternative Flows	-	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 Activity Diagram

แผนภาพที่แสดงลำดับขั้นตอนการทำงานของฟังก์ชั่นการทำงานภายในระบบ

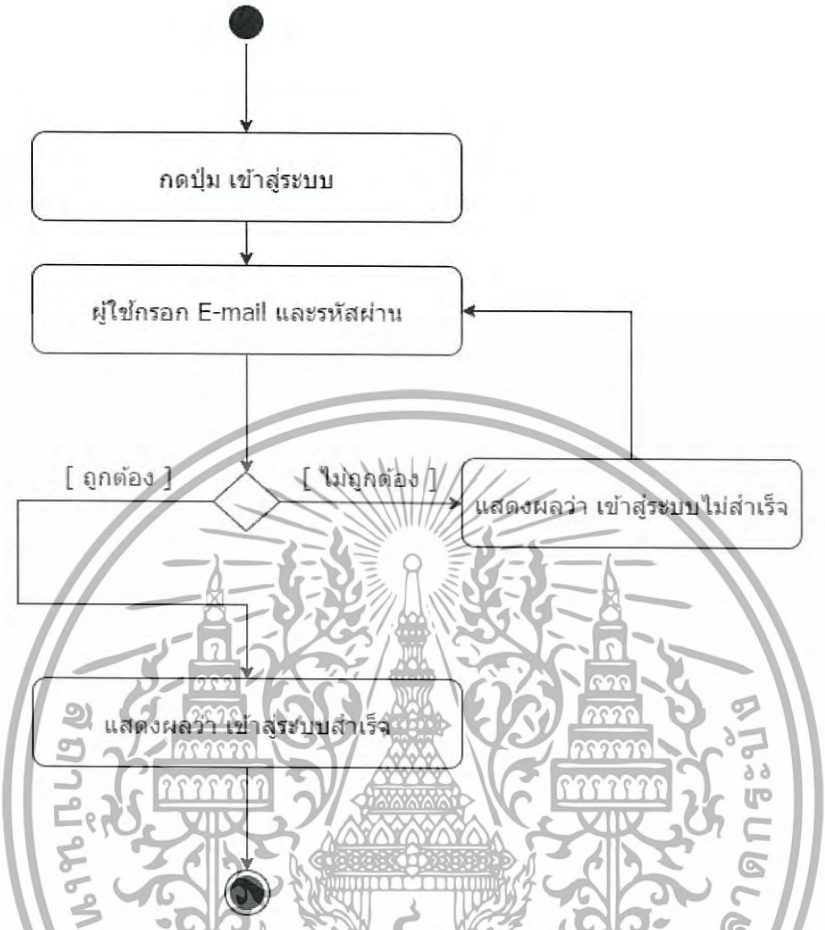
a) แผนภาพกิจกรรมยูสเคส สมัครสมาชิก



รูปที่ 3.5 แผนภาพกิจกรรมยูสเคสสมัครสมาชิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

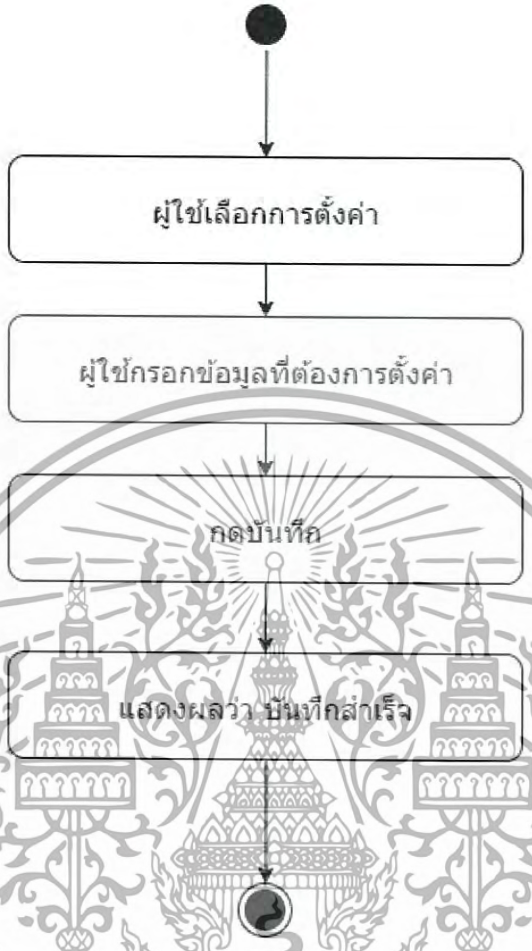
b) แผนภาพกิจกรรมยูสเคส เข้าสู่ระบบ



รูปที่ 3.6 แผนภาพกิจกรรมยูสเคสเข้าสู่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

c) แผนภาพกิจกรรมยูสเคส ตั้งค่าข้อมูลส่วนตัว



รูปที่ 3.7 แผนภาพกิจกรรมยูสเคสตั้งค่าข้อมูลส่วนตัว

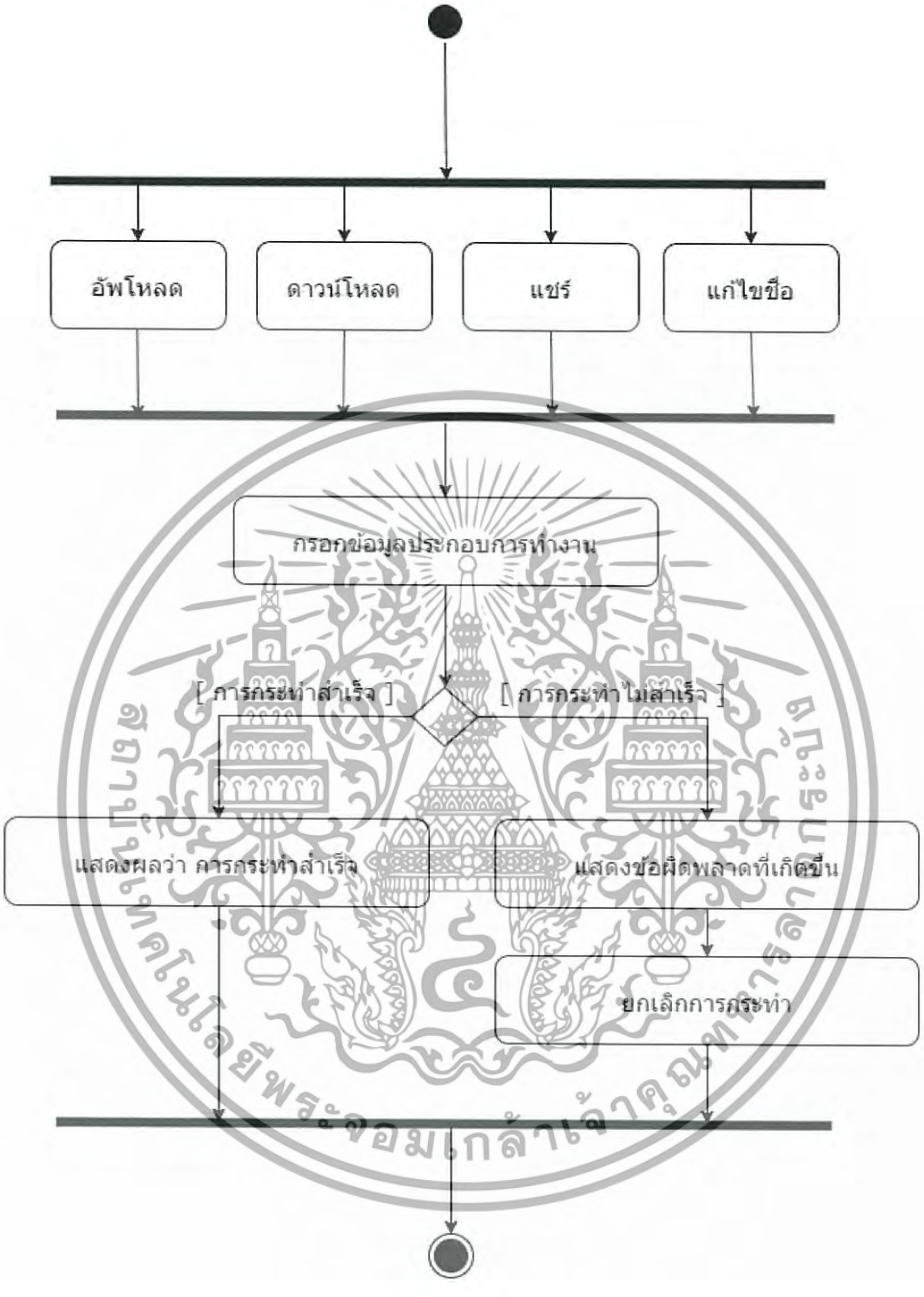
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

d) แผนภาพกิจกรรมยูสเคส เพิ่มการเชื่อมต่อกับ Cloud Storage

รูปที่ 3.8 แผนภาพกิจกรรมยูสเคสเพิ่มการเชื่อมต่อกับ Cloud Storage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

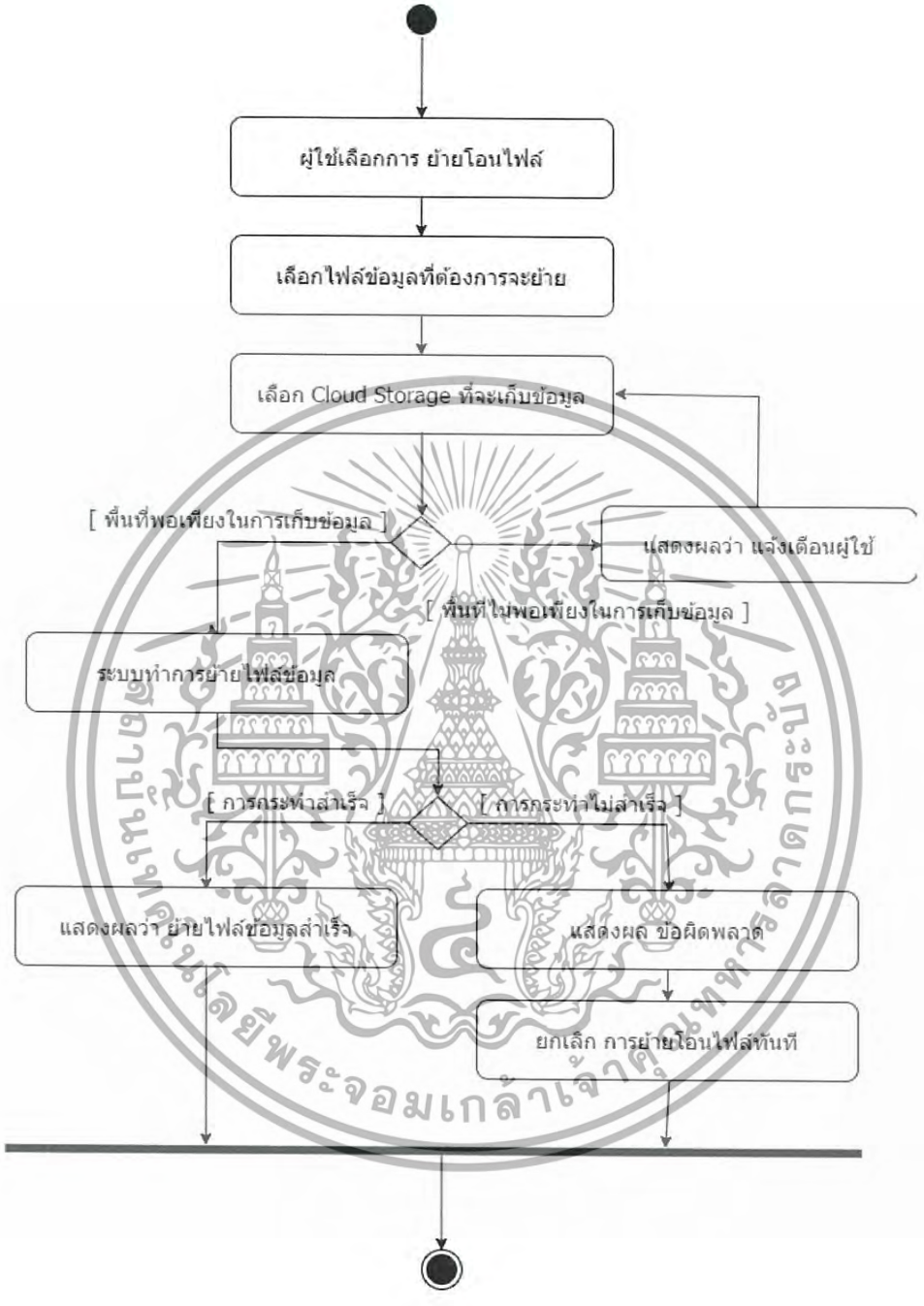
e) แผนภาพกิจกรรมยูสเคส จัดการไฟล์และข้อมูล



รูปที่ 3.9 แผนภาพกิจกรรมยูสเคสจัดการไฟล์และข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

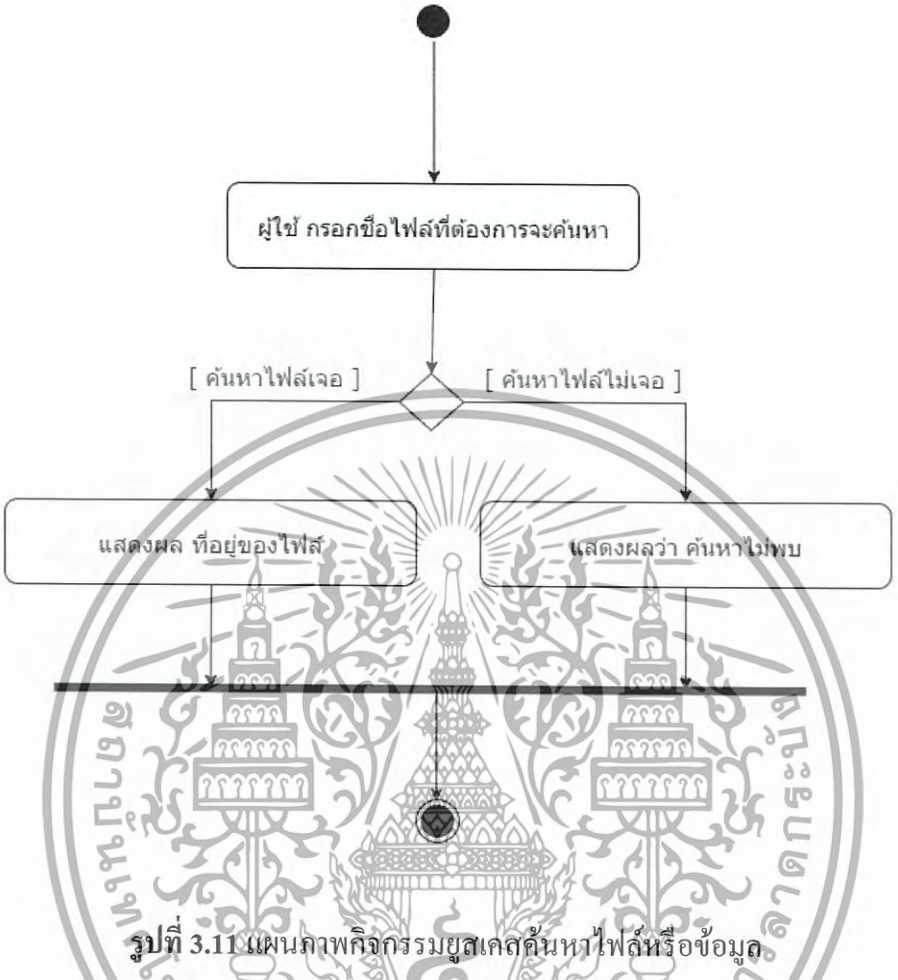
f) แผนภาพกิจกรรมยูสเคส ย้ายไฟล์หรือข้อมูลข้าม Cloud Storage



รูปที่ 3.10 แผนภาพกิจกรรมยูสเคสจัดการย้ายไฟล์หรือข้อมูลข้าม Cloud Storage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

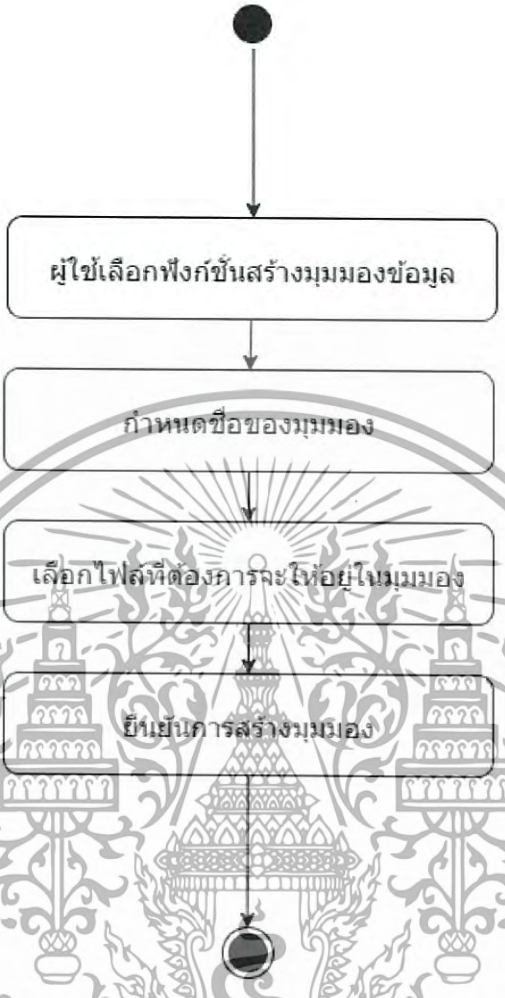
g) แผนภาพกิจกรรมยูสเคส ค้นหาไฟล์หรือข้อมูล



รูปที่ 3.11 แผนภาพกิจกรรมยูสเคสค้นหาไฟล์หรือข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

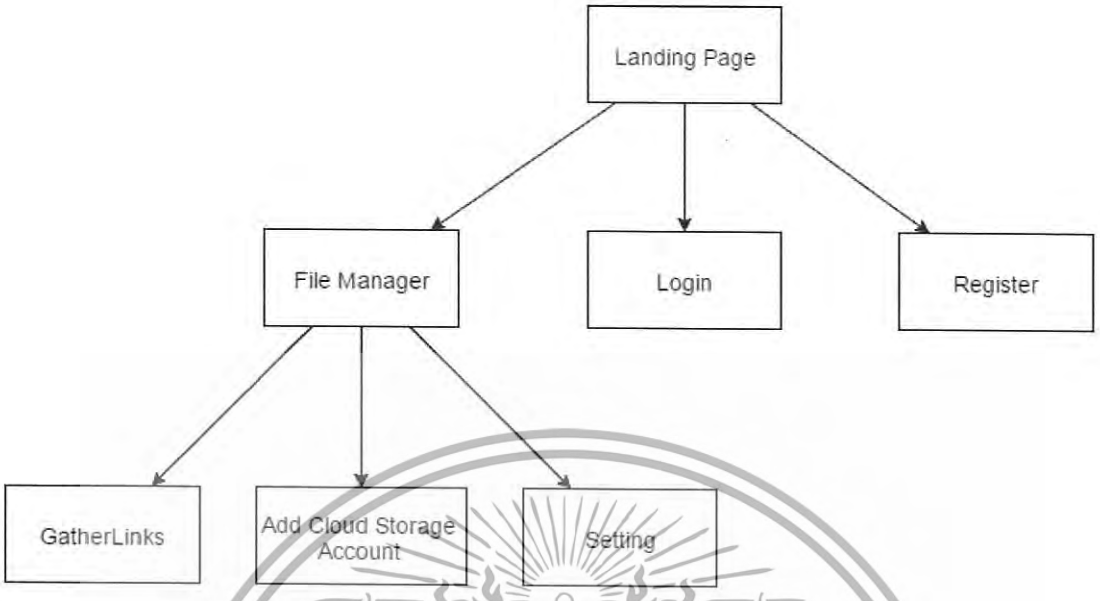
h) แผนภาพกิจกรรมยูสเคส สร้างมุมมองการมองเห็นข้อมูล



รูปที่ 3.12 แผนภาพกิจกรรมยูสเคสสร้างมุมมองการมองเห็นข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.4 Website Structure



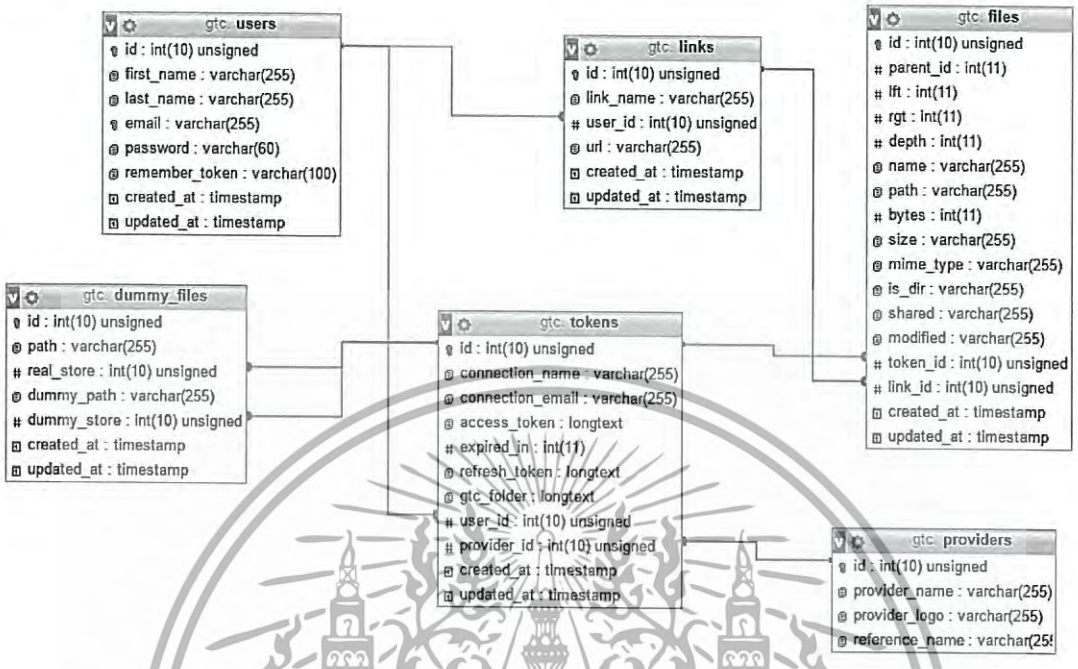
รูปที่ 3.12 โครงสร้างเว็บไซต์ Gathercloud

โครงสร้างของเว็บไซต์ จะมีหน้าแสดงผลแรกที่พบเมื่อเริ่มใช้งานคือ หน้า Landing Page ที่จะแสดงรายละเอียดการคร่าวๆ เพื่อให้ผู้ใช้เข้าใจในระบบ และสามารถเลือกที่จะ Login หรือ Register เพื่อให้เข้าถึงหน้า File Manager ที่จะเป็นหน้าเว็บไซต์ที่เกี่ยวข้องกับการทำงานหลักทั้งหมดของระบบ ไม่ว่าจะเป็นการจัดการไฟล์ข้อมูล อัพโหลด ดาวน์โหลด และหลังจากนั้น จะสามารถเข้าถึงหน้าที่ Add Cloud Storage Account ที่จะสามารถเพิ่มบัญชีของแต่ละบริการ Cloud Storage ได้ รวมถึงหน้า Setting ที่ไว้ใช้ในการตั้งค่า ข้อมูลต่างๆ เกี่ยวกับบัญชีผู้ใช้ของระบบ GatherCloud และนอกจากนี้ยังมีอีกฟังก์ชั่นการทำงานของระบบ ได้แก่ GatherLinks ที่จะเป็นหน้าเว็บไซต์ที่ช่วยในการสร้างมุมมองข้อมูลจากหลายบริการ Cloud Storage ผ่านระบบ GatherCloud ได้ ตามแผนภาพในรูปที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การออกแบบฐานข้อมูล

3.6.1 ER Diagram



รูปที่ 3.13 แผนภาพโครงสร้างฐานข้อมูล

3.6.2 รายละเอียดตาราง (Table) และ แอตทริบิวต์ (Attribute)

1. ตาราง users ซึ่งจะใช้ในการจัดเก็บข้อมูลการระบุตัวตนของผู้ใช้งาน ประกอบด้วยแอตทริบิวต์ดังนี้

ตารางที่ 3.10 แอตทริบิวต์ตาราง users

แอตทริบิวต์	ประเภท	คำอธิบาย
id(PK)	Int	ไอดีของผู้ใช้งาน
first_name	Varchar(255)	ชื่อจริง
last_name	Varchar(255)	นามสกุล
email	Varchar(255)	อีเมล
password	Varchar(60)	พาสเวิร์ดของผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ตาราง tokens ซึ่งจะใช้ในการจัดเก็บการจับคู่ผู้ใช้งานกับ Token ประกอบด้วยแอตทริบิวต์ดังนี้

ตารางที่ 3.11 แอตทริบิวต์ตาราง tokens

แอตทริบิวต์	ประเภท	คำอธิบาย
id(PK)	Int	ไอดีของ Token
connection_name	Varchar(255)	ชื่อการเชื่อมต่อกับบัญชีแหล่งเก็บข้อมูลบนกลุ่มเมฆ
connection_email	Varchar(255)	อีเมลของบัญชีแหล่งเก็บข้อมูลบนกลุ่มเมฆ
access_token	Longtext	Access Token
expired_in	Int	จำนวนวินาทีที่ Access Token จะหมดอายุ
refresh_token	Longtext	Refresh Token
gtc_folder	Longtext	โฟลเดอร์ที่รองรับการเก็บไฟล์จำลอง
provider_id(FK)	Int	ไอดีของผู้ให้บริการ
users_id(FK)	Int	ไอดีของผู้ใช้งาน

3. ตาราง providers ที่ใช้เก็บข้อมูลที่เกี่ยวข้องกับผู้ให้บริการที่ระบบรองรับ ประกอบด้วยแอตทริบิวต์ดังนี้

ตารางที่ 3.12 แอตทริบิวต์ตาราง providers

แอตทริบิวต์	ประเภท	คำอธิบาย
id(PK)	Int	ไอดีของผู้ให้บริการ
provider_name	Varchar(255)	ชื่อของผู้ให้บริการ
provider_logo	Varchar(255)	ชื่อไฟล์ภาพที่จะใช้เป็นโลโก้ของผู้ให้บริการ
reference_name	Varchar(255)	ค่าที่จะใช้ในการประมวลภายในระบบของผู้ให้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตาราง links ซึ่งจะใช้ในการจัดเก็บ URL ของมุมมองการมองเห็นข้อมูลที่ผู้ใช้สร้างขึ้น ประกอบด้วยแอตทริบิวต์ดังนี้

ตารางที่ 3.13 แอตทริบิวต์ตาราง links

แอตทริบิวต์	ประเภท	คำอธิบาย
id(PK)	Int	ไอดีของมุมมองการมองเห็นข้อมูล
link_name	Varchar(255)	ชื่อของมุมมองการมองเห็นข้อมูล
url	Varchar(255)	จะเก็บเฉพาะ Token ที่เป็นเอกลักษณ์ของ View
users_id(FK)	Int	ไอดีของผู้ใช้งาน

5. ตาราง dummy_files ที่ใช้ในการเก็บข้อมูลเกี่ยวกับไฟล์ที่ถูกเก็บไว้ในผู้ให้บริการอื่นๆ โดยลักษณะการเก็บจะเป็นการเก็บแบบจำลองเพื่อสนับสนุนการทำงานในแนวคิดที่ระบบจะจัดการข้อมูลเหมือนกับมีแหล่งเก็บข้อมูลบนกลุ่มเมฆพียงที่เดียว ประกอบด้วยแอตทริบิวต์ดังนี้

ตารางที่ 3.14 แอตทริบิวต์ตาราง dummy_files

แอตทริบิวต์	ประเภท	คำอธิบาย
id(PK)	Int	ไอดีของไฟล์จำลอง
path	Varchar(255)	path ของจริงที่ไฟล์ถูกเก็บไว้
real_store (FK)	Int	ไอดีของ Token ที่ไฟล์ตัวจริงถูกเก็บไว้อยู่
dummy_path	Varchar(255)	path จำลองที่เก็บไฟล์ไว้
dummy_store (FK)	Int	ไอดีของ Token ที่ไฟล์ถูกเก็บไว้ในที่จำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ตาราง files ซึ่งจะใช้ในการจัดเก็บรายละเอียดของ files ต่างๆ ที่ผู้ใช้เป็นเจ้าของ โดยใน Table นี้ เราได้ใช้ Library Baum เข้ามาช่วย เพื่อจะแปลงข้อมูลภายใน Table ให้กลายเป็นข้อมูลโครงสร้างแบบ Tree หรือ Nested Set ได้ เพื่อเก็บข้อมูลไฟล์และโฟลเดอร์ต่างๆ ประกอบด้วยแอตทริบิวต์ดังนี้

ตารางที่ 3.15 แอตทริบิวต์ตาราง files

แอตทริบิวต์	ประเภท	คำอธิบาย
id(PK)	Int	ไอดีรายละเอียดของมุมมองการมองเห็นข้อมูล
parent_id	Int	บอก id ของ node ที่เป็น parent
lft	Int	บ่งบอกขอบเขตของ Child node
rgt	Int	บ่งบอกขอบเขตของ Child node
depth	Int	บ่งบอกความลึกของ Trees
name	Varchar(255)	ชื่อของไฟล์
path	Varchar(255)	Path ของไฟล์ที่อยู่ในแหล่งเก็บข้อมูล
bytes	Int	ขนาดของข้อมูล
size	Varchar(255)	ขนาดของข้อมูลแบบอ่านได้
mime_type	Varchar(255)	ชนิดของไฟล์
is_dir	Varchar(255)	บ่งบอกสถานะว่าเป็นโฟลเดอร์หรือไม่
shared	Varchar(255)	ไฟล์ดังกล่าวอยู่ในสถานะที่แบ่งปันกับผู้อื่นหรือไม่
modified	Varchar(255)	วันที่ที่แก้ไขไฟล์ล่าสุด
token_id(FK)	Int	Token ที่เป็นที่มาของไฟล์
link_id	Int	ไฟล์ดังกล่าวเป็นสมาชิกของ Link ใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ระบบต้นแบบหรือผลการทดลอง

4.1 ผลการทดลองและการศึกษา

ในการระหว่างการพัฒนา GatherCloud ได้มีการทดลองและศึกษาเนื้อหาที่เกี่ยวข้องกับ API ของผู้ให้บริการแหล่งเก็บข้อมูลแต่ละเจ้า ซึ่งจะมีรายละเอียดดังต่อไปนี้

4.1.1 ผลการศึกษา API ของ Dropbox [3]

4.1.1.1 สภาพแวดล้อมในการพัฒนา

ในการใช้งาน API v1 ที่ทาง Dropbox ให้บริการนั้น ทาง Dropbox ได้มีการให้บริการโดยสนับสนุนวิธีการพัฒนาอย่างหลากหลาย ได้แก่ Python, Ruby, PHP, Java, Android, iOS, OS X, และ Pure HTTP ซึ่งวิธีการพัฒนาที่เราเลือกใช้คือ Pure HTTP จะเป็นการทำงานในรูปแบบของ REST หรือก็คือการเรียกฟังก์ชั่นต่างๆที่ API สนับสนุน ผ่าน URI ที่ทาง Dropbox กำหนดด้วย HTTP Verbs เช่น GET, PUT, POST เป็นต้น สิ่งที่ควรรู้เบื้องต้นนั้น ตัว Core API ของ Dropbox นั้นจะต้องอาศัยการ HTTP Request บน SSL และต้องใช้สิทธิในการเข้าถึงและแก้ไขเมื่อได้มีการทำ Authentication นอกจากนี้ยังมีรูปแบบของวันที่ที่ถูกกำหนดไว้อย่างชัดเจน และจำเป็นต้องใช้การติดต่อโดยใช้ String ที่ผ่านการ Encoding แบบ UTF-8 เรียบร้อยแล้วเพื่อให้เกิดการทำงานอย่างเต็มประสิทธิภาพ

Hostname ของ API ที่ทาง Dropbox ได้จัดเตรียมไว้ให้ ได้แก่

- https://api.dropboxapi.com
- https://content.dropboxapi.com
- https://notify.dropboxapi.com

4.1.1.2 ขั้นตอนการทำงาน

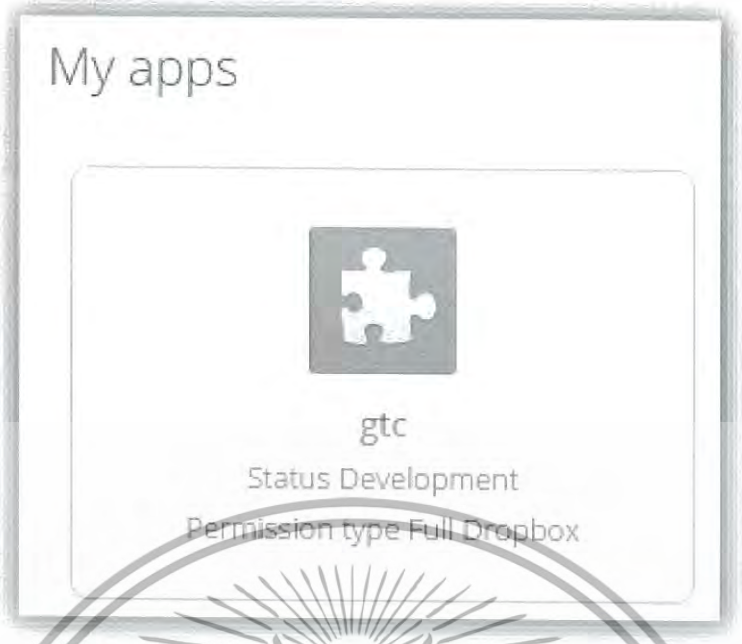
1. เริ่มจากการสร้าง Application บนบริการของ Dropbox

1.1 สมัคร Account ของ Dropbox

1.2 Create Application และกรอกข้อมูลให้ครบถ้วน รวมถึงกำหนดระดับของสิทธิที่ Application ของเราจะสามารถเข้าถึงข้อมูลของผู้ใช้ได้

1.3 กำหนด redirect_uri ที่จำเป็นต้องใช้ต่อในขั้นตอนถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 หน้า Application Console ของ Dropbox

2. User กระทำปุ่มที่ต้องการเชื่อมต่อกับ Application โดยจะเชื่อมต่อกับ Dropbox ดังนี้

ตารางที่ 4.1 รายละเอียด HTTP Request ในการเชื่อมต่อกับ Dropbox API

URL	https://www.dropbox.com/oauth2/authorize
Method	GET
Parameter	- response_type (required) : จะสามารถใส่ได้เพียง token หรือ code เท่านั้น - client_id (required) : Application Consumer Key ที่ทาง Dropbox ระบุไว้ - redirect_uri : จำเป็นต้องตรงกับที่กรอกไว้เมื่อสร้าง Application - state : ใช้เพื่อป้องกันการโจมตีแบบ cross-site request forgery (CSRF) - require_role : ระบุประเภทของ Account ที่สามารถอนุญาตให้ Application เข้าถึงได้ โดยจะระบุเป็น work หรือ personal - force_reaaprove : มีค่าตั้งต้นเป็น false อยู่แล้ว แต่ถ้าเป็น true จะทำให้ User ต้องอนุญาตให้ Application เข้าถึงได้อีกครั้งแม้ว่าจะอนุญาตไปแล้วก็ตาม - disable_signup : มีค่าตั้งต้นเป็น false แต่ถ้าเป็น true จะทำให้ User ไม่สามารถลงทะเบียนกับ Dropbox ในหน้า Authorization ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

a) RESPONSE_TYPE = code

การตอบรับแบบ code นั้นจะทำให้ได้ code ที่จะต้องนำไปเรียกใช้งานต่ออีกครั้งก่อนจะได้ Access Token มาใช้งาน โดยจะได้รับมาอยู่ในรูปแบบดังตัวอย่าง

[REDIRECT_URI]?code={Code}&state=[STATE]

และหลังจากนั้นต้องนำ Code ไปเรียกใช้งาน

ตารางที่ 4.2 รายละเอียด HTTP Request ในการใช้ Code เพื่อดึง Access Token

URL	https://www.dropbox.com/oauth2/token
Method	POST
Parameter	- code (required) : code ที่ได้จากการเรียก Response_type=code - grant_type (required) : จะต้องมีค่าเป็น authorization_code เท่านั้น - redirect_uri : จำเป็นต้องตรงกับที่กรอกไว้เมื่อสร้าง Application - client_id : แสดงตัวตนของ Application หรือ Application Keys นั้นเอง - client_secret : ไว้ใช้เพื่อยืนยันตัวตนของ Application หรือ Application Secret

b) RESPONSE_TYPE = token

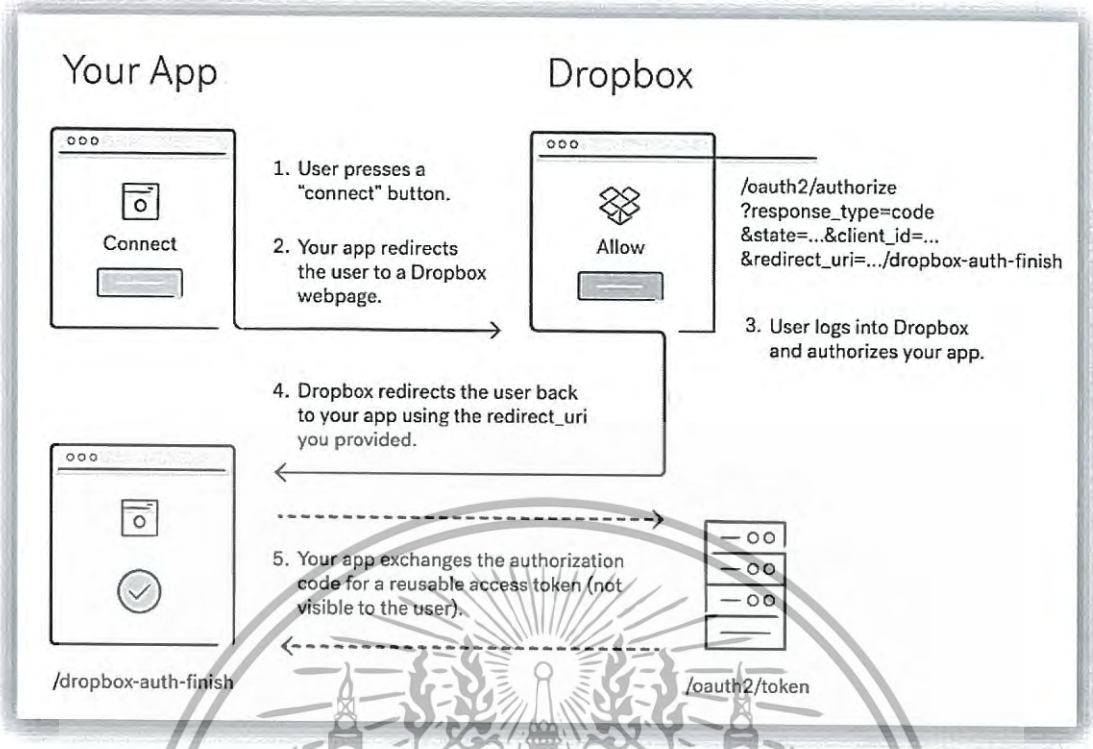
จะได้รับ Access Token มาทันที โดยจะส่งมาผ่านทาง URL Fragments และจะสามารถนำไปเรียกใช้ฟังก์ชั่นการทำงานของ Dropbox API ได้ทันที ดังตัวอย่าง

[REDIRECT_URI]#access_token={Access_token}&token_type=bearer&

uid={Dropbox User ID}&state=[STATE]

3. นำ Access Token ที่ได้มา ไปใช้เพื่อเรียกใช้งานฟังก์ชั่นต่างๆ ของ API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แผนภาพหลักการการเชื่อมต่อของกับ Dropbox API

(ที่มา: https://www.dropbox.com/developers/reference/oauth-guide)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.3 รายชื่อฟังก์ชั่นการทำงานที่สนับสนุน

ตารางที่ 4.3 รายชื่อฟังก์ชั่นการทำงานของ Dropbox REST API

URI	Function
/accounts/info	ดึงข้อมูลเบื้องต้นเกี่ยวกับ User's Account นั้น
/files (GET)	ใช้ในการดาวน์โหลดข้อมูล
/files_put	ใช้ในการอัพโหลดไฟล์ขึ้นไปบนแหล่งเก็บข้อมูลบนก้อนเมฆ
/metadata	รับข้อมูลเกี่ยวกับไฟล์หรือโฟลเดอร์ที่กำหนด
/search	ค้นหาไฟล์หรือโฟลเดอร์ที่กำหนด โดยจะให้ข้อมูลที่เกี่ยวกับไฟล์หรือโฟลเดอร์นั้น
/shares	สร้าง link ที่สามารถแชร์ให้ผู้ใช้คนอื่นๆ เข้าถึงได้
/chunked_upload	ทยอยอัพโหลดไฟล์ที่มีขนาดใหญ่ ทีละชิ้นๆ เพราะฟังก์ชั่น /files_put มีข้อจำกัด สามารถอัพโหลดไฟล์ได้ไม่เกิน 150 MB
/commit_chunked_upload	ช่วยในการปิดการอัพโหลดแบบ /chunked_upload ให้เสร็จสมบูรณ์
/filesops/copy	คัดลอกไฟล์ภายใน Dropbox ได้
/filesops/create_folder	สร้างโฟลเดอร์ใหม่ภายใน Dropbox ได้
/filesops/delete	ลบไฟล์ภายใน Dropbox ได้
/filesops/move	ย้ายไฟล์ไปไว้อีกตำแหน่งหนึ่งภายใน Dropbox

4.1.1.4 Library

ในการพัฒนาโปรแกรม เราเลือกที่จะใช้ Library ของ f4bsch/DropPHP อยู่บน GitHub ซึ่งเป็นเพียง Library ขนาดเล็ก DropPHP มีการทำงานเป็น Interface อย่างง่ายๆ เพื่อเชื่อมต่อกับ API มีเพียงแค่ Class DropboxClient ไว้เพื่อเรียกใช้งานฟังก์ชั่นพื้นฐานเท่านั้น เช่น อัพโหลด ดาวน์โหลด ดึงข้อมูลผู้ใช้ ดึงข้อมูลของไฟล์หรือโฟลเดอร์ เป็นต้น ทำให้เข้าใจได้ง่าย สามารถนำมาใช้งานได้ทันที สามารถปรับแต่งเพิ่มเติมได้ และคุณสมบัติของ Library ดังกล่าวยังทำงานด้วย REST ซึ่งไม่มีความซับซ้อน ส่งผลให้สามารถศึกษาแนวทางในการพัฒนา Library เผื่อนำไปประยุกต์กับการเรียกใช้ API ตัวอื่นๆ ได้อีกด้วย นอกจากนี้ DropPHP ยังสามารถทำงานได้ด้วยตัวเอง ไม่จำเป็นต้องพึ่ง Library ตัวอื่นๆ ของ PHP ซึ่งลดปัญหาในการจัดเตรียม Environment ได้เป็นอย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ผลการศึกษา API ของ Copy [4]

4.1.2.1 สภาพแวดล้อมในการพัฒนา

จากการศึกษา API ของ Copy ก็ได้พบว่ามีการใช้งานในลักษณะ REST จึงทำให้สามารถเข้าถึงได้ง่าย และใช้หลักการของ HTTP Request และ Response เป็นหลัก ซึ่งสิ่งที่จะได้กลับในจะอยู่ในรูปแบบข้อมูลของ JSON และ Copy API นั้น จะทำงานบน HTTPS เท่านั้น เพื่อป้องกันความเป็นส่วนตัวของข้อมูล การ Request หรือ Response HTTP ที่ไม่ได้ถูกเข้ารหัสจะไม่สามารถใช้งาน API ดังกล่าวได้ และนอกจากนี้ในการเรียกใช้งาน API ของ Copy จำเป็นที่จะต้องเพิ่ม HTTP Request Header ไปยัง Base URL ดังนี้ https://api.copy.com/rest/ โดยต้องเพิ่ม

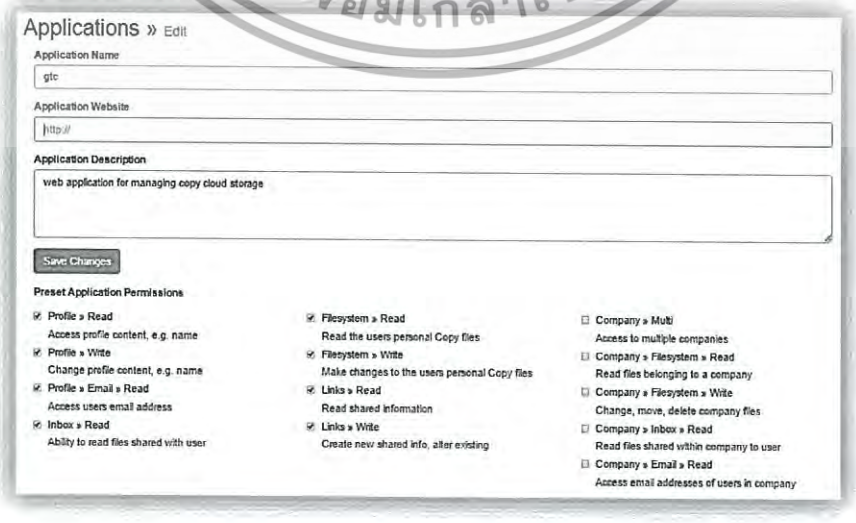
- **X-Api-Version**: 1
- **Accept**: application/json

และในการเรียกใช้ฟังก์ชั่นการใช้งาน API ของ Copy จะอาศัย HTTP Verbs มาช่วยในการแยกการทำงาน ทำให้ URI ที่ใช้มีจำนวนน้อยและเข้าใจได้ง่าย

- GET : ใช้เพื่อรับค่าจากทรัพยากรที่ URL กำหนด
- POST : สร้างข้อมูลใหม่ ไปยังที่ที่ URL กำหนด
- PUT : แก้ไขหรืออัพเดทข้อมูลที่ URL กำหนด
- PATCH : ใช้แก้ไขข้อมูลบางส่วนตามที่ URL กำหนด
- DELETE : ใช้เพื่อลบข้อมูลตามที่ URL กำหนด

4.1.2.2 ขั้นตอนการทำงาน

1. สร้าง Application บน Application Console ของ Copy รวมถึงกำหนดสิทธิการเข้าถึงของ Application ให้เรียบร้อย โดยสร้างผ่าน https://developers.copy.com



รูปที่ 4.3 หน้า Application Console ของ Copy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Application ที่เป็น Third-Party เรียกขอ Request Token จาก Copy API โดยมีรายละเอียดดังนี้

ตารางที่ 4.4 รายละเอียด HTTP Request ของ Copy API เพื่อขอ Request Token

HTTPS Request	
URL	https://api.copy.com/oauth/request
Method	GET
Parameter	?scope={Application Permission}
GET or Authorization Header	- oauth_callback - oauth_consumer_key - oauth_signature_method - oauth_nonce - oauth_timestamp - oauth_version - oauth_signature
HTTP Response Body	- oauth_token - oauth_token_secret - oauth_callback_confirmed
HTTP Request Example	GET /oauth/request?scope=%7B%22profile%22%3A%7B%22read%22%3A+true%2C%22write%22%3A+true%7D%7D HTTP/1.1 User-Agent: PECL-OAuth/1.2.3 Host: api.copy.com Accept: */* Authorization: OAuth oauth_callback="{ Callback URL }" oauth_consumer_key="{ Your Application Consumer Key }", oauth_signature_method="HMAC-SHA1", oauth_nonce="147648019951a8be1b1b7a37.94665308", oauth_timestamp="1370013211", oauth_version="1.0", oauth_signature="aAar8Fy1%2FgIh7xLhGIn7nDTPtYQ%3D"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. User อนุญาตให้ Third Party Application เข้าถึง และ Copy API จะมอบ oauth_token และ oauth_verifier สามารถระบุ Callback URL ใน oauth_callback ที่ต้องการจะให้ Application กลับไป และนำ oauth_token และ oauth_verifierไปใช้ต่อ ในขั้นตอนถัดไป

รูปที่ 4.4 หน้ายินยอมสิทธิในการเข้าถึงบัญชีผู้ใช้จาก Copy Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Third-Party Application ร้องขอ Access Token จาก Copy API โดยมีรายละเอียดดังนี้

ตารางที่ 4.5 รายละเอียด HTTP Request ของ Copy API เพื่อขอ Access Token

HTTP Request	
URL	https://api.copy.com/oauth/access
Method	GET
Parameter	None
GET or Authorization Header	- oauth_verifier - oauth_consumer_key - oauth_signature_method - oauth_nonce - oauth_timestamp - oauth_version - oauth_token - oauth_signature
HTTP Response Body	- oauth_token - oauth_token_secret
HTTP Request Example	GET /oauth/access HTTP/1.1 User-Agent: PECL-OAuth/1.2.3 Host: api.copy.co Accept: */* Authorization: OAuth oauth_verifier="66aac376b9eeaa01f17448c7923bd3c9", oauth_consumer_key="{ Your Application Consumer Key}", oauth_signature_method="HMAC-SHA1", oauth_nonce="146033336251a8be1e459232.95602358", oauth_timestamp="1370013214", oauth_version="1.0", oauth_token="2NXCMeG771vhoWY5vhGIm4xbGuxqUvbH", oauth_signature="1Pp0VQNIXiSNZGer%2B3m0SlDfm4s%3D"

5. นำ Access Token ที่ได้ไปใช้เพื่อเรียกฟังก์ชั่นการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2.3 รายชื่อฟังก์ชั่นการทำงานที่สนับสนุน

ตารางที่ 4.6 รายละเอียดเกี่ยวกับฟังก์ชั่นที่สนับสนุนของ Copy API

HTTP Verb	URI	Function
GET	/user	ดึงข้อมูลเบื้องต้นของผู้ใช้งาน
PUT	/user	แก้ไขข้อมูลเบื้องต้นของผู้ใช้งาน
GET	/meta	อ่านข้อมูลเบื้องต้นของไฟล์จาก root
GET	/meta/{param}	อ่านข้อมูลเบื้องต้นในระดับที่ลึกขึ้นได้ตาม {param}
DELETE	/meta/{param}/path/to/file	ลบไฟล์หรือโฟลเดอร์
PUT	/meta/{param}/path/to/FILE.txt?name=NEW.txt	แก้ไขชื่อไฟล์ FILE.txt ด้วย NEW.txt
PUT	/meta/{param}/path/to/FILE.txt?name=/NEW/PATH/FILE.txt	ทำการย้ายไฟล์ไปยัง Directory ที่กำหนด
POST	/meta/{param}/path/to/OBJECTNAME	สร้างโฟลเดอร์ หรืออัพโหลดไฟล์
GET	/links	อ่านค่า Link ทั้งหมดที่เคยสร้าง
GET	/links/TOKEN	อ่านค่า Link ที่กำหนด
POST	/links	สร้าง Link ใหม่เพื่อแชร์ให้ผู้อื่นได้
PUT	/links/TOKEN	แก้ไขข้อมูลหรือไฟล์ที่ให้คนอื่นใน Link ที่กำหนด
PATCH	/links/TOKEN	แก้ไขผู้ที่จะมีส่วนร่วมใน Link รวมถึงสิทธิการกระทำด้วย
DELETE	/links/TOKEN	ลบ Link ตามที่กำหนด
GET	/meta/links/TOKEN	ดูไฟล์หรือโฟลเดอร์ที่คนอื่นสามารถมองเห็นใน Link นั้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2.4 Library

ในส่วนของ Library ที่เราเลือกใช้ คือ Library ที่ได้พัฒนามาจากทีมงานที่พัฒนา Copy Application หรือ Barracuda โดย copy-app/php-client-library ดังกล่าวก็เป็น Library ที่สนับสนุนการทำงานร่วมกับภาษา PHP ทำให้สามารถเข้าใจได้ง่ายและตรงกับความต้องการของ GatherCloud อีกทั้งตัว Library นั้นค่อนข้างศึกษาและใช้งานง่าย มีการทำงานร่วมกับ OAuth ซึ่งเป็น Library เสริมที่ช่วยในการยืนยันตัวตนหรือ Authentication ก่อนที่จะสามารถใช้งาน copy-app/php-client-library ได้จริงๆ

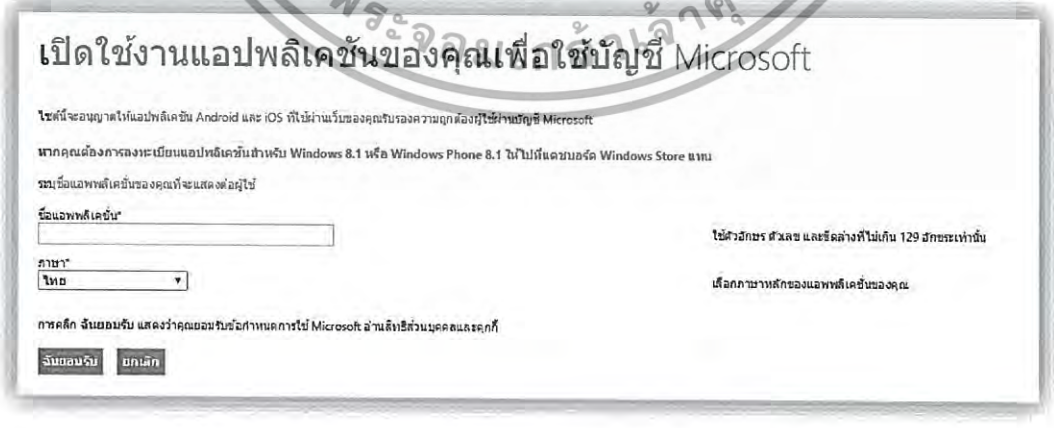
4.1.3 ผลการศึกษา API ของ Onedrive [5]

4.1.3.1 สภาพแวดล้อมในการพัฒนา

จากการศึกษาการใช้งาน API ของ OneDrive พบว่ามีลักษณะการทำงานแบบ REST ซึ่งใช้ HTTP Request ส่งผ่าน URL ที่กำหนดไว้เพื่อขอใช้ฟังก์ชั่นต่าง ๆ และรับ Response กลับมา ซึ่งส่วนใหญ่จะอยู่ในรูปแบบของ JSON ที่มีขนาดเล็กและสามารถเข้าใจได้ง่าย ซึ่งการเรียกใช้ฟังก์ชั่นทั้งหมดจะสามารถทำได้หลังจากผ่านการ Authentication แล้วเท่านั้น โดย API ของ OneDrive ใช้ OAuth 2.0 ในการรับรองสิทธิการเข้าถึงการใช้งาน ทำให้มีความปลอดภัยและความน่าเชื่อถือ

4.1.3.2 ขั้นตอนการทำงาน

1. สมัครสมาชิกกับทาง Microsoft และสร้างแอพพลิเคชั่นผ่านเว็บไซต์สำหรับนักพัฒนาที่ https://account.live.com/developers/applications หลังจากนั้นทำการตั้งค่าเพื่อกำหนดระดับสิทธิเข้าถึงข้อมูลผู้ใช้งานของแอพพลิเคชั่น และตั้งค่า Redirect URL



รูปที่ 4.5 หน้าสร้าง Application ของ OneDrive

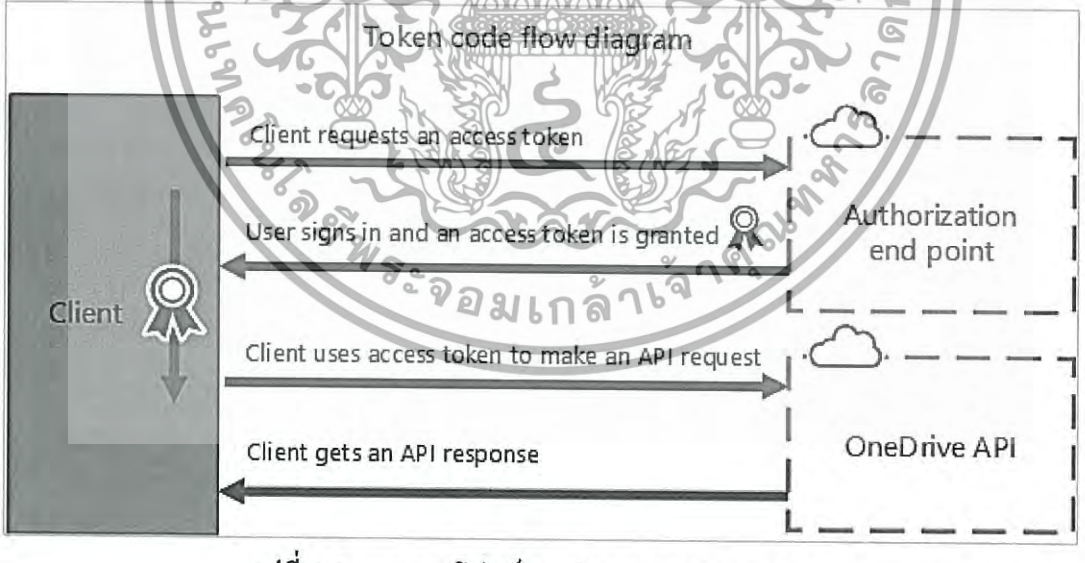
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เชื่อมต่อกับ OneDrive API โดยร้องขอการรับรองสิทธิเข้าถึงการใช้งานจากผู้ใช้งานแอพพลิเคชั่นผ่าน Authorization end point มีด้วยกันสองวิธี ดังนี้

2.1 ร้องขอ Access Token โดยตรง

ตารางที่ 4.7 รายละเอียด HTTP Request ของ OneDrive API เพื่อขอ Access Token โดยตรง

HTTP Request	
URL	https://login.live.com/oauth20_authorize.srf
Method	GET
Query String Parameter	- response_type (required) : จะสามารถใส่ได้เพียง token หรือ code เท่านั้น ในที่นี้ใส่ token - client_id (required) : Application Consumer Key ที่ทาง OneDrive ระบุไว้ - redirect_uri (required) : จำเป็นต้องตรงกับ Redirect URL ที่ตั้งค่าไว้ - scope (required) : wl.signin
HTTP Response	
[Redirect URL]#access_token={Access Token}&authentication_token={Authentication Token }&token_type=bearer &expires_in=3600&scope={scope}&user_id={User ID}	



รูปที่ 4.6 แผนภาพโฟลว์การทำงานของ Token Code

(ที่มา: https://dev.onedrive.com/auth/msa_oauth.htm)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ร้องขอ Access Token ผ่าน Authorization code

2.2.1 ร้องขอ Authorization code

ตารางที่ 4.8 รายละเอียด HTTP Request ของ OneDrive API เพื่อขอ Authorization code

HTTP Request	
URL	https://login.live.com/oauth20_authorize.srf
Method	GET
Query String Parameter	- response_type (required) : จะสามารถใส่ได้เพียง token หรือ code เท่านั้น ในที่นี้ใส่ code - client_id (required) : Application Consumer Key ที่ทาง OneDrive ระบุไว้ - redirect_uri (required) : จำเป็นต้องตรงกับ Redirect URL ที่ตั้งค่าไว้ - scope (required) : wl.signin
HTTP Response	
[Redirect URL]?code={Authorization code}	

2.2.2. เมื่อได้รับ Response กลับมาที่หน้า Redirect URL ที่เราตั้งค่าไว้ ให้นำ Authorization code ที่แนบมาในรูปของพารามิเตอร์ชื่อว่า code มาใช้เพื่อร้องขอ Access Token ผ่าน Token end point

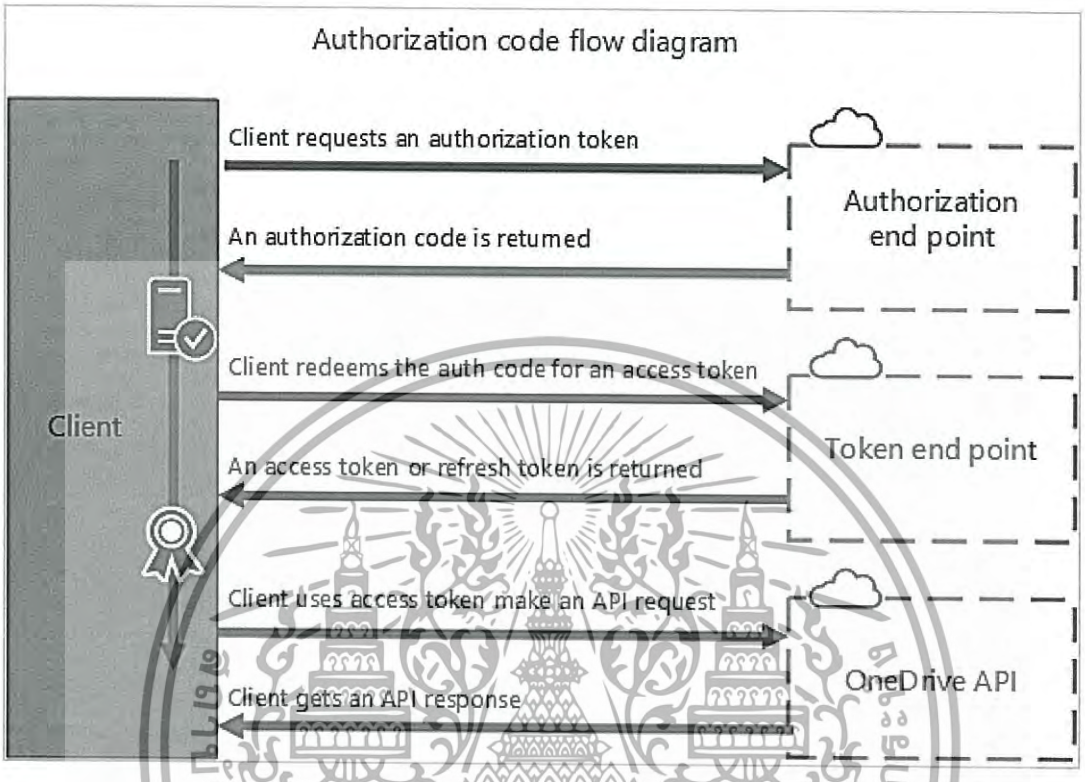
ตารางที่ 4.9 รายละเอียด HTTP Request ของ OneDrive API เพื่อขอ Access Token ด้วย Code

HTTP Request	
URL	https://login.live.com/oauth20_token.srf
Method	POST
Header	Content-Type: application/x-www-form-urlencoded
Request Body Parameter	- client_id (required) : Application Consumer Key ที่ทาง OneDrive ระบุไว้ - client_secret (required) : Application Secret Key ที่ทาง OneDrive ระบุไว้ - redirect_uri (required) : จำเป็นต้องตรงกับ Redirect URL ที่ตั้งค่าไว้ - code (required) : {Authorization code}
HTTP Response	
JSON	- token_type - expires_in - scope - access_token

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.9 (ต่อ) รายละเอียด HTTP Request ของ OneDrive API เพื่อขอ Access Token ด้วย Code

	- refresh_token (หากกำหนด scope เป็น wl.offline_access)



รูปที่ 4.7 แผนภาพโฟลว์การทำงานของ Authorization Code

(ที่มา: https://dev.onedrive.com/auth/msa_oauth.htm)

3. นำ Access Token ที่ได้มา ไปใช้เพื่อเรียกใช้งานฟังก์ชั่นต่างๆ ของ API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3.3 รายชื่อฟังก์ชั่นการทำงานที่สนับสนุน

ตารางที่ 4.10 รายละเอียดเกี่ยวกับฟังก์ชั่นของ OneDrive API

Base URL	https://api.onedrive.com/v1.0	
HTTP Verb	**URI**	**Function**
GET	/drive	ดึงข้อมูลเบื้องต้นของผู้ใช้งาน
GET	/drives/{drive-id}	ดึงข้อมูลเบื้องต้นของผู้ใช้งานที่ระบุโดย drive-id
GET	/drive/items/root?expand=children	ดึงข้อมูลโฟลเดอร์และไฟล์ทั้งหมดโดยมองจาก Root
GET	/drive/items/{item-id}	ดึงข้อมูลโฟลเดอร์หรือไฟล์ที่ระบุโดย item-id
GET	/drive/items/{item-id}/content	ดาวน์โหลดโฟลเดอร์หรือไฟล์ที่ระบุโดย item-id
PUT	/drive/items/{parent-id}/children/{name}	สร้างโฟลเดอร์ใหม่ ระบุตำแหน่งโดย parent-id และชื่อโฟลเดอร์โดย name
PUT	/drive/items/{parent-id}/children/{filename}/content *แนบรายละเอียดที่ Body Parameter*	อัพโหลดไฟล์ ระบุตำแหน่งโดย parent-id และชื่อโฟลเดอร์โดย filename
DELETE	/drive/items/{id}	ลบโฟลเดอร์หรือไฟล์ที่ระบุโดย id
PATCH	/drive/items/{id} *แนบรายละเอียดที่ Body Parameter*	แก้ไขชื่อหรือตำแหน่งที่อยู่ของโฟลเดอร์หรือไฟล์ที่ระบุโดย id
POST	/drive/items/{id}/action.copy *แนบรายละเอียดที่ Body Parameter*	คัดลอกโฟลเดอร์หรือไฟล์ที่ระบุโดย id ไปยังอีกที่หนึ่ง
GET	/drive/items/{id}/view.search	ค้นหาโฟลเดอร์หรือไฟล์ที่ระบุโดย id
POST	/drive/items/{id}/action.createLink	สร้าง Link ใหม่ สำหรับโฟลเดอร์หรือไฟล์ที่ระบุโดย id เพื่อแชร์ให้ผู้อื่นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3.4 Library

ทางผู้ให้บริการ OneDrive มี Official Library ให้เลือกใช้ทั้งภาษา .NET, C#, Xamarin, iOS, Python และ Android แต่ไม่มี ภาษา PHP ที่ใช้พัฒนา GatherCloud ดังนั้นจึงทำการค้นหา Library ภาษา PHP ทำให้พบ krizalys/onedrive-php-sdk ที่อยู่บน Github จึงได้ทำการทดลองใช้งาน Library ดังกล่าว มีตัวอย่างการใช้งานทำให้เข้าใจการใช้งานได้รวดเร็ว แต่ยังมีฟังก์ชั่นการทำงานที่ไม่ครบถ้วนเช่นการสร้าง Link สำหรับแชร์โฟลเดอร์หรือไฟล์ จึงทำให้ต้องเขียนฟังก์ชั่นนี้ขึ้นมาเองแต่ด้วยตัว Library ที่เขียนแบ่งคลาสและฟังก์ชั่นการทำงานที่ชัดเจน จึงทำให้สะดวกต่อการแก้ไขและเพิ่มเติม

4.1.4 ผลการศึกษา API ของ Box [6]

4.1.4.1 สภาพแวดล้อมในการพัฒนา

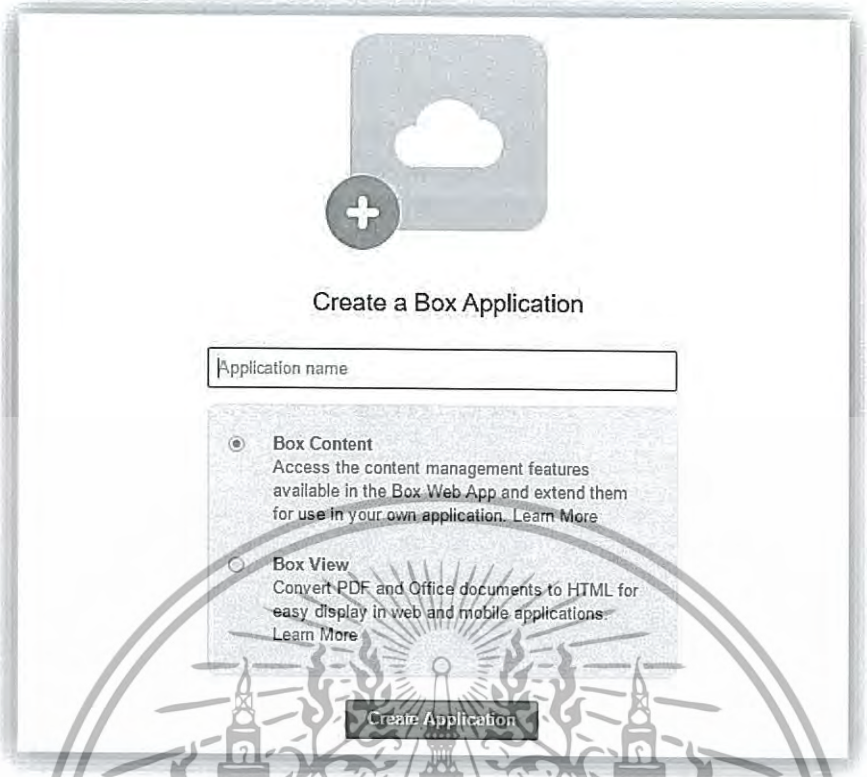
จากการศึกษาวิธีการใช้งาน Box API พบว่ามีการแบ่ง API เป็นสองส่วนหลักได้แก่ Box Content API Docs เป็นส่วนของการจัดการข้อมูลของผู้ใช้งานทั้งหมด และ Box View API Docs เป็นส่วนสำหรับช่วยให้ไฟล์จำพวก Office documents สามารถแสดงผลบน HTML ได้ ซึ่งส่วนที่เราศึกษามุ่งเน้นไปที่ Box Content API Docs

Box Content API Docs มีการเรียกใช้งานในลักษณะ REST คือการเรียกใช้ผ่าน HTTP Request ตาม URL ที่ระบุไว้ และใช้ OAuth 2.0 ในการรับรองสิทธิการเข้าถึงการใช้งาน ส่วนของ Response จะได้รับกลับมาในรูปแบบของ JSON ที่มีขนาดเล็กและเข้าใจได้ง่าย

4.1.4.2 ขั้นตอนการทำงาน

1. สมัครสมาชิกกับทาง Box และสร้างแอพพลิเคชั่นผ่านเว็บไซต์สำหรับนักพัฒนาที่ https://app.box.com/developers/services เลือกประเภท Box Content หลังจากนั้นทำการตั้งค่าเพื่อกำหนดระดับสิทธิเข้าถึงข้อมูลผู้ใช้งานของแอพพลิเคชั่น และตั้งค่า Redirect URL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 หน้าสร้าง Application ของ Box

2. เชื่อมต่อกับ Box API โดยร้องขอการรับรองสิทธิเข้าถึงการใช้งานจากผู้ใช้งานแอพพลิเคชั่น ผ่าน Authorization endpoint ดังนี้

2.1. ร้องขอ Access Token ผ่าน Authorization code

ตารางที่ 4.11 รายละเอียด HTTP Request ของ Box API เพื่อขอ Authorization Code

HTTP Request	
URL	https://app.box.com/api/oauth2/authorize
Method	GET
Query String Parameter	- response_type (required) : code - client_id (required) : Application Consumer Key ที่ทาง Box ระบุไว้ - redirect_uri (required) : จำเป็นต้องตรงกับ Redirect URL ที่ตั้งค่าไว้
HTTP Response	
[Redirect URL]?code={Authorization code}	

2.2 เมื่อได้รับ Response กลับมาที่หน้า Redirect URL ที่เราตั้งค่าไว้ ให้นำ Authorization code ที่แนบมาในรูปของพารามิเตอร์ชื่อว่า code มาใช้เพื่อร้องขอ Access Token ผ่าน Token end point

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.12 รายละเอียด HTTP Request ไว้ร้องขอ Access Token จาก Box API ด้วย Code

HTTP Request	
URL	https://api.box.com/oauth2/token
Method	POST
Request Body Parameter	- client_id (required) : Application Consumer Key ที่ทาง OneDrive ระบุไว้ - client_secret (required) : Application Secret Key ที่ทาง OneDrive ระบุไว้ - grant_type (required) : authorization_code - code (required) : {Authorization code}
HTTP Response	
JSON	- access_token - expires_in - token_type - refresh_token

2.3 นำ Access Token ที่ได้มา ไปใช้เพื่อเรียกใช้งานฟังก์ชั่นต่างๆ ของ API

4.1.4.3 รายชื่อฟังก์ชันการทำงานที่สนับสนุน

ตารางที่ 4.13 รายละเอียดฟังก์ชั่นการทำงานของ Box API

Base URL	https://api.box.com/2.0	
HTTP Verb	**URI**	**Function**
GET	/users/me	ดึงข้อมูลเบื้องต้นของผู้ใช้งาน
GET	/users/{USER_ID}	ดึงข้อมูลเบื้องต้นของผู้ใช้งานที่ระบุโดย USER_ID
GET	/folders/{0}	ดึงข้อมูลโฟลเดอร์ Root
GET	/folders/{FOLDER_ID}	ดึงข้อมูลโฟลเดอร์ที่ระบุโดย FOLDER_ID
GET	/folders/{FOLDER_ID}/items	ลิสต์รายละเอียดคร่าวๆ ของโฟลเดอร์หรือไฟล์ที่อยู่ในโฟลเดอร์ที่ระบุโดย FOLDER_ID
POST	/folders *แนบรายละเอียดที่ Body Parameter*	สร้างโฟลเดอร์ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.13 (ต่อ) รายละเอียดฟังก์ชั่นการทำงานของ Box API

DELETE	/folders/{FOLDER_ID}	ลบโฟลเดอร์ที่ระบุโดย FOLDER_ID
POST	/folders/{FOLDER_ID}/copy *แนบรายละเอียดที่ Body Parameter*	คัดลอกโฟลเดอร์ที่ระบุโดย FOLDER_ID ไปยังอีกที่หนึ่ง
PUT	/folders/{FOLDER_ID} *แนบรายละเอียดที่ Body Parameter*	แก้ไขข้อมูลต่างๆ เช่น ชื่อ, ตำแหน่งที่อยู่ หรือสร้าง Link สำหรับแชร์ของโฟลเดอร์ที่ระบุโดย FOLDER_ID
GET	/files/{FILE_ID}	ดึงข้อมูลไฟล์ที่ระบุโดย FILE_ID
POST	Base URL: https://upload.box.com/api/2.0/files/content *แนบรายละเอียดที่ Body Parameter*	อัพโหลดไฟล์
DELETE	/files/{FILE_ID}	ลบไฟล์ที่ระบุโดย FILE_ID
POST	/files/{FILE_ID}/copy *แนบรายละเอียดที่ Body Parameter*	คัดลอกไฟล์ที่ระบุโดย FILE_ID ไปยังอีกที่หนึ่ง
PUT	/files/{FILE_ID} *แนบรายละเอียดที่ Body Parameter*	แก้ไขข้อมูลต่างๆ เช่น ชื่อ, ตำแหน่งที่อยู่ หรือสร้าง Link สำหรับแชร์ของไฟล์ที่ระบุโดย FILE_ID

4.1.4.4 Library

ในส่วน Official Library ของ Box API นั้นมีให้เลือกหลายภาษา ทั้ง Java, Python, Ruby, .NET, Mobile, OSX, Chrome และ Salesforce แต่ไม่มีภาษา PHP ที่ GatherCloud เลือกใช้พัฒนา จึงได้ค้นหาและพบ adammbalogh/box-php-sdk ซึ่งเป็น Unofficial Library ที่อยู่บน Github มีข้อมูลการใช้งานคร่าว ๆ ไม่ลงรายละเอียดลึกแต่มีฟังก์ชั่นการทำงานครอบคลุมตามที่ต้องการ ตัว Library นี้ใช้ Library ตัวอื่นควบคู่ด้วย แต่สามารถจัดการได้ด้วย Composer ทำให้มีความสะดวกมากขึ้น

4.1.5 ผลการศึกษา API ของ Google Drive [7]

4.1.5.1 สภาพแวดล้อมในการทำงาน

จากการศึกษา API ของ Google Drive พบว่ามีลักษณะการทำงานแบบ REST ที่ส่ง HTTP Request ไปเรียกใช้งานคำสั่งต่าง ๆ ของ API และส่ง Respone กลับมาในรูปแบบของ JSON แต่ละ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

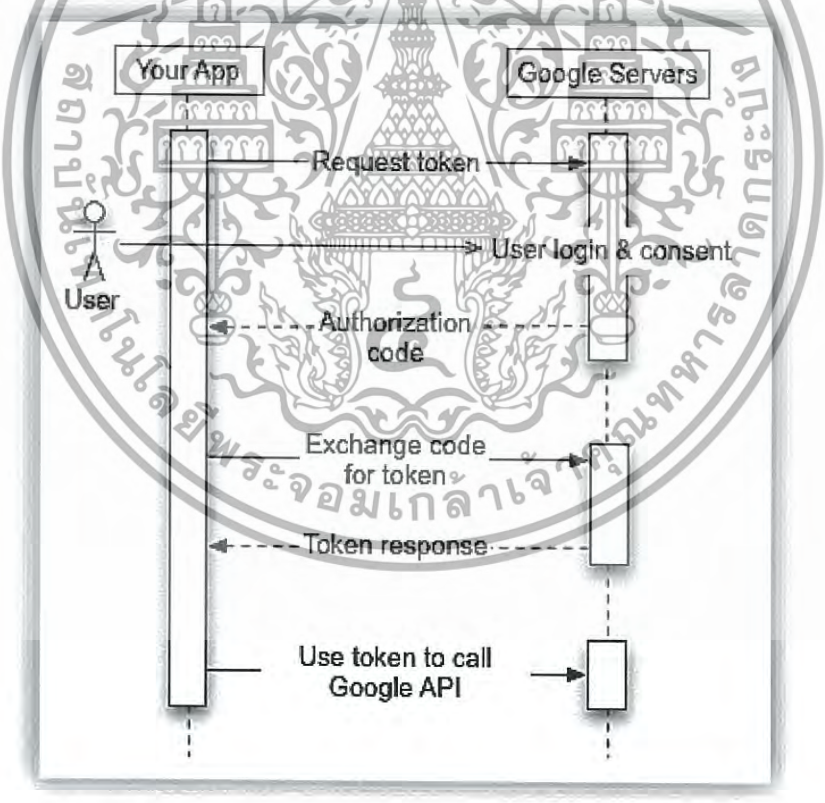
Respone จะมี Property ของ JSON ที่แตกต่างกันขึ้นอยู่กับคำสั่งที่เรียกใช้ไป ในส่วนของการรับรองสิทธิการเข้าถึงการใช้งานนั้น Google Drive ใช้ OAuth 2.0 ที่เป็นที่นิยมกันมากที่สุด

ซึ่ง Google Drive ได้พัฒนา Library สำหรับส่งคำสั่งผ่าน REST ขึ้นมารับรองภาษาในการพัฒนาต่าง ๆ ไว้ได้แก่ .NET, Android, Go, iOS, Java, JavaScript, Node.js, PHP, Python และ Ruby นอกจากนี้ยังมีเอกสารการใช้งาน Library เฉพาะของแต่ภาษาอธิบายไว้อย่างชัดเจนอีกด้วย

4.1.5.2 ขั้นตอนการทำงาน

1. สมัครสมาชิกกับทาง Google และเข้าไปในเว็บไซต์สำหรับนักพัฒนาที่ URL https://console.developers.google.com เพื่อสร้าง Project และ Credentials สำหรับรับ client ID และ Client secret เพื่อใช้ในการรับรองสิทธิการเข้าถึงการใช้งาน

2. เชื่อมต่อกับ Google Drive API โดยร้องขอการรับรองสิทธิเข้าถึงการใช้งานจากผู้ใช้งานแอพพลิเคชั่น ผ่าน Authorization endpoint ดังนี้



รูปที่ 4.9 ขั้นตอนการร้องขอ Token ในรูปแบบของ Web Aplication

(ที่มา: https://developers.google.com/identity/protocols/OAuth2#installed)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 ร้องขอ Access Token ผ่าน Authorization code

ตารางที่ 4.14 รายละเอียด HTTP Request ของ Google Drive API เพื่อขอ Authorization Code

HTTP Request	
URL	https://accounts.google.com/o/oauth2/auth
Method	GET
Query String Parameter	- response_type (required) : code - client_id (required) : client ID จาก Credentials ที่สร้างไว้ - redirect_uri (required) : หน้าที่จะรับ code มาใช้ต่อ - scope(required) : https://www.googleapis.com/auth/drive - access_type : offline - approval_prompt : force
HTTP Response	
[Redirect URL]?code={Authorization code}	

2.2 เมื่อได้รับ Response กลับมาที่หน้า Redirect URL ที่เราตั้งค่าไว้ให้นำ Authorization code ที่แนบมาในรูปของพารามิเตอร์ชื่อว่า code มาใช้เพื่อร้องขอ Access Token ผ่าน Token end point

ตารางที่ 4.15 รายละเอียด HTTP Request ของ Google Drive API เพื่อขอ Token จาก code

HTTP Request	
URL	https://www.googleapis.com/oauth2/v4/token
Method	POST
Request Body Parameter	- client_id (required) : client ID จาก Credentials ที่สร้างไว้ - client_secret (required) : client secret จาก Credentials ที่สร้างไว้ - grant_type (required) : authorization_code - redirect_uri(required) : หน้าที่ต้องการให้ย้อนกลับมา - code (required) : {Authorization code}
HTTP Response	
JSON	- access_token - expires_in - token_type - refresh_token

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 นำ Access Token ที่ได้มา ไปใช้เพื่อเรียกใช้งานฟังก์ชั่นต่างๆ ของ API

4.1.5.3 รายชื่อฟังก์ชั่นการทำงานที่สนับสนุน

ตารางที่ 4.16 รายละเอียดฟังก์ชั่นการทำงานของ Google Drive API

Base URL	https://www.googleapis.com/drive/v3	
HTTP Verb	**URI**	**Function**
GET	/about	ดึงข้อมูลเบื้องต้นของผู้ใช้งาน
GET	/files	ดึงข้อมูลโฟลเดอร์และไฟล์ทั้งหมด
GET	/files/{item-id}	ดึงข้อมูลโฟลเดอร์หรือไฟล์ที่ระบุโดย item-id
GET	/files/{item-id}?alt-media	ดาวน์โหลดโฟลเดอร์หรือไฟล์ที่ระบุโดย item-id
POST	/files *แนบรายละเอียดที่ Body Parameter*	สร้างโฟลเดอร์ใหม่หรืออัพโหลดไฟล์
DELETE	/files/{item-id}	ลบโฟลเดอร์หรือไฟล์ที่ระบุโดย item-id
PATCH	/files/{item-id} *แนบรายละเอียดที่ Body Parameter*	แก้ไขชื่อหรือตำแหน่งที่อยู่ของโฟลเดอร์หรือไฟล์ที่ระบุโดย item-id
POST	/files/{item-id}/copy *แนบรายละเอียดที่ Body Parameter*	คัดลอกโฟลเดอร์หรือไฟล์ที่ระบุโดย item-id ไปยังอีกที่หนึ่ง
GET	/files?q={keyword}	ค้นหาโฟลเดอร์หรือไฟล์ที่ระบุโดย keyword
POST	/files/{item-id}/permissions *แนบรายละเอียดที่ Body Parameter*	สร้าง Link ใหม่ สำหรับโฟลเดอร์หรือไฟล์ที่ระบุโดย item-id เพื่อแชร์ให้ผู้อื่นได้

4.1.5.4 Library

เนื่องจากทาง Google มีการเตรียม Library ไว้สำหรับพัฒนาไว้หลายภาษา ได้แก่ .NET, Android, Go, iOS, Java, JavaScript, Node.js, PHP, Python และ Ruby พร้อมทั้งคู่มือที่อธิบายวิธีการใช้งาน Library ไว้อย่างชัดเจน อีกทั้ง Library นั้นยังสามารถเข้าถึง API ตัวอื่น ๆ ที่เป็นของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Google ได้อีกด้วย อาทิเช่น Google Calendar หรือ Gmail ซึ่งตัว Library ภาษา PHP ที่ทาง Google พัฒนามาสามารถใช้งานร่วมกับ Composer ที่เป็นตัวจัดการ Library ได้อีกด้วย

4.1.6 บันทึกการเพิ่มการรองรับผู้ให้บริการของระบบ

ภายหลังของการพัฒนาระบบนั้น ผู้พัฒนาได้ตระหนักถึงความสำคัญของการออกแบบระบบให้สามารถแก้ไขได้ง่ายเพื่อที่จะรองรับการเพิ่มการเชื่อมต่อและรองรับกับผู้ให้บริการที่ผู้พัฒนาตัดสินใจที่จะเพิ่มเข้ามาในภายหลัง จึงได้พยายามแก้ไขและปรับปรุงให้ระบบมีความยืดหยุ่นและได้ทำการทดสอบด้วยการเพิ่มการเชื่อมต่อกับผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆ ซึ่งก็คือ Google Drive และได้ทำการบันทึกขั้นตอนในการแก้ไขตัวระบบ เพื่อสังเกตความยุ่งยากในการแก้ไขและเพื่อเป็นกรณีศึกษาในอนาคตอีกด้วย

ขั้นตอนในการเพิ่มการเชื่อมต่อผู้ให้บริการกับระบบโดยสังเขปมีดังนี้

1. เพิ่มข้อมูลของผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆไว้ในฐานข้อมูลโดยมีชื่อตารางว่า Provider ด้วยคำสั่งของ MySQL

2. ศึกษาขั้นตอนในการใช้งาน API ของผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆ หรือนำ Library ที่ศึกษามาใช้งานในระบบ ผ่าน Composer หรือนำ Source Code มาไว้ในระบบ

3. สร้างคลาส Interface สำหรับเรียกใช้งาน Sub-system ที่เป็น API หรือ Library ของผู้ให้บริการ โดยทำการ Implement มาจาก App\Library\ModelInterface.class และทำการ Implement ทุก methods ในคลาสดังกล่าวให้ครบถ้วนและทำงานได้ และตั้งชื่อคลาสโดยเป็นชื่อผู้ให้บริการตามด้วยคำว่า Interface เช่น GoogleDriveInterface เป็นต้น

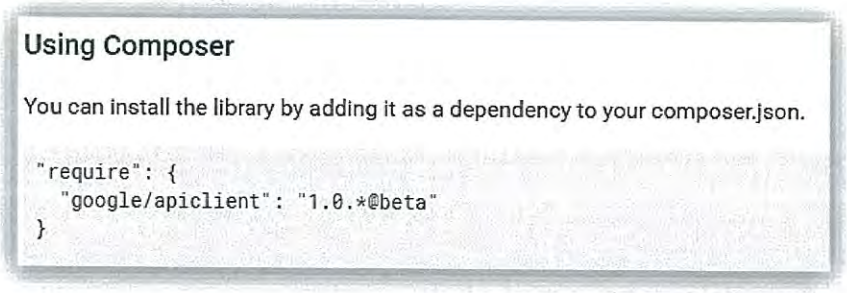
4. สามารถนำภาพ logo ของผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆมาไว้ในระบบได้ด้วยเช่นกัน โดยนำไฟล์ภาพมาใส่ไว้ใน public\images\logo-provider และนำชื่อไฟล์เก็บไว้ในฐานข้อมูล

หลังจากที่ได้แก้ไขตามวิธีที่ได้ระบุมาด้านบนจะทำให้ระบบสามารถรองรับผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆ ระบบก็รองรับการทำงานกับ API ของผู้ให้บริการนั้น

ตัวอย่างถัดไปคือบันทึกที่ผู้พัฒนาได้ทำการเพิ่มการรองรับผู้ให้บริการของระบบ ซึ่งผู้บริการที่จะเชื่อมต่อกับระบบคือ Google Drive โดยมีรายละเอียดดังนี้

1. ในขั้นตอนแรก ผู้พัฒนาเริ่มจากการศึกษา API และ Library ของผู้ให้บริการ Google Drive และได้นำ client library ของ Google มาใช้งานในระบบผ่าน Composer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 หน้าเว็บไซต์รายละเอียดของ Google Client Library

(ที่มา: https://goo.gl/z4mhIY)

2. ใช้คำสั่ง `composer install` ผ่านหน้า Command Line เพื่อให้ Composer ทำการดาวน์โหลด client library มาไว้ใน vendor ดังรูปภาพที่ 4.10

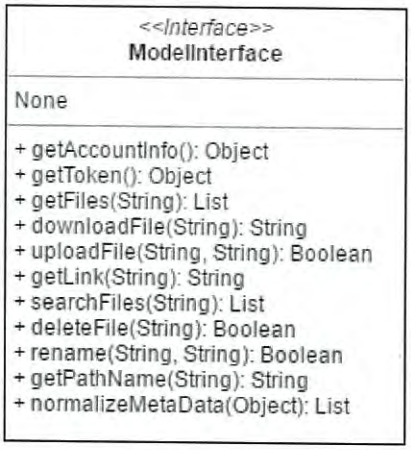
รูปที่ 4.11 โครงสร้างของ Google Client Library

2. ใส่ข้อมูลของ Provider ลงในฐานข้อมูลระบบผ่านคำสั่ง MySQL และนำ logo-googledrive.png วางไว้ใน public\images\logo-provider เรียบร้อย

```
INSERT INTO `providers`(`provider_name`, `provider_logo`, `value`)
VALUES ('Google Drive','logo-googledrive.png','GoogleDrive');
```

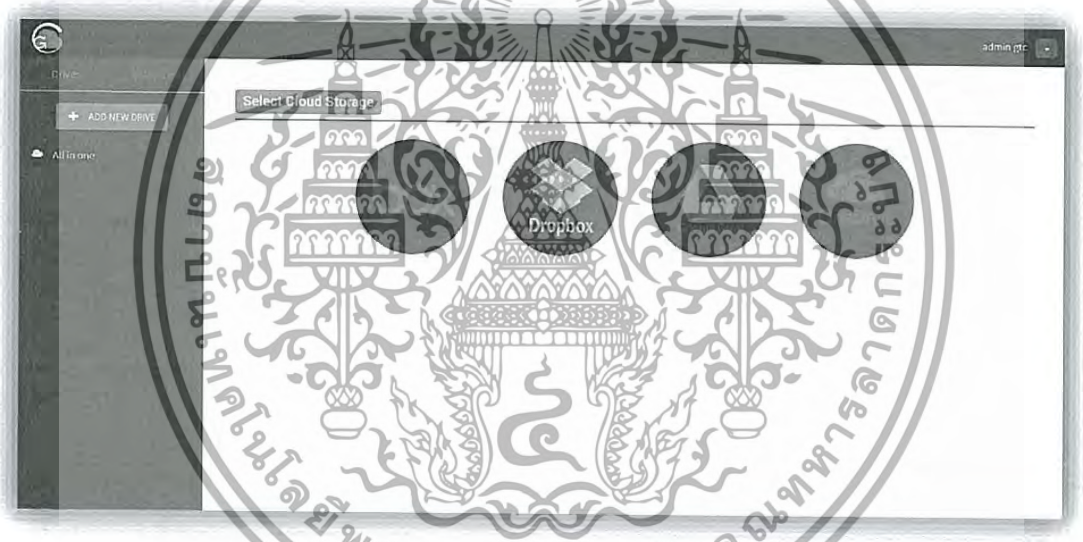
4. อ่านและทำความเข้าใจกับ ModelInterface.php เพื่อที่จะ Implement method ได้อย่างครบถ้วนและถูกต้อง และทำการ Implement method ให้ครบถ้วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 รายละเอียดของ ModelInterface ในรูปแบบของ UML

5. ทำการทดสอบกับระบบ โดยสังเกตุได้ว่าการเพิ่มการเชื่อมต่อกับผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆได้รองรับ Google Drive สำเร็จ



รูปที่ 4.13 หน้าเลือกผู้ให้บริการแหล่งเก็บข้อมูลเพื่อเชื่อมต่อกับ GatherCloud

4.1.7 บันทึกการลดการรองรับผู้ให้บริการของระบบ

ในการพัฒนาระบบนั้น ผู้พัฒนาได้พบกับสถานการณ์ที่กระตุ้นให้เห็นถึงปัญหาของระบบที่ไม่ยืดหยุ่นเพียงพอที่จะรองรับในสถานการณ์ที่ต้องยกเลิกการรองรับการเชื่อมต่อกับผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆ จึงทำให้ตระหนักถึงปัญหาและต้องการให้ความสำคัญกับการออกแบบระบบมากยิ่งขึ้น โดยสถานการณ์ที่ได้พบเจอนั้นคือ การประกาศข่าวจากผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆ Copy จะปิดให้บริการลงในวันที่ 1 พฤษภาคม พ.ศ.2559 จึงต้องมีการปรับปรุงระบบให้เข้ากับสถานการณ์ หรือก็คือปรับให้ระบบลดการเชื่อมต่อกับผู้ให้บริการ และเพื่อที่จะป้องกันปัญหาที่จะอาจจะเกิดขึ้นในอนาคตโดยเกี่ยวกับการลดการเชื่อมต่อกับผู้ให้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้น ผู้พัฒนาจึงได้ทดลองและบันทึกขั้นตอนในการแก้ไขระบบ เพื่อสังเกตถึงความยุ่งยากและซับซ้อนของระบบอีกด้วย

ขั้นตอนในการลดการรองรับผู้ให้บริการของระบบโดยสังเขปมีดังนี้

1. ลบคลาส Interface ที่สร้างขึ้นมา

2. นำ Library หรือ API ที่ได้นำเข้ามาสู่ระบบออก

3. ลบข้อมูลเกี่ยวกับผู้ให้บริการออกจากฐานข้อมูลตาราง Provider และฐานข้อมูลจะทำการลบข้อมูลที่เกี่ยวข้องกับผู้ให้บริการนั้นทั้งหมดรวมถึงข้อมูลของผู้ใช้ทั้งหมดโดยอัตโนมัติ

เมื่อผู้พัฒนาได้ทำการตามวิธีการด้านบน ทำให้ระบบไม่รองรับต่อการทำงานของผู้ให้บริการ Copy อีกต่อไป อีกทั้งยังทำข้อมูลของผู้ใช้ที่เกี่ยวข้องกับผู้ให้บริการหายไปจากระบบทั้งหมดอีกด้วย ดังนั้นจึงควรพิจารณาให้ดีก่อนที่จะทำการลบข้อมูลหรือไฟล์ใดๆ ออกจากระบบ

4.2 ระบบต้นแบบ

4.2.1 Framework

ในส่วนการทำงานของ GatherCloud ที่มีการพัฒนาจนออกมาเป็นระบบต้นแบบนั้น ได้เลือกใช้ PHP Framework เข้ามาช่วยในการพัฒนา ซึ่งในจำนวน Framework มากมายที่มีการพัฒนาออกมานั้น ได้พิจารณาและเลือกที่จะใช้ PHP Framework ที่ชื่อว่า Laravel 5.1 [8]

Laravel 5.1 เป็น PHP Framework ที่ทำงานในรูปแบบของ MVC (Model-View-Controller) ซึ่งเป็นส่วนสำคัญประกอบการพิจารณา ที่ถือได้ว่าตรงกับสถาปัตยกรรมที่เราเลือกจะใช้ไว้แต่แรก อีกทั้งยังเป็น Framework ที่สนับสนุนพัฒนาเว็บไซต์ด้วยภาษา PHP อีกด้วย รองรับทั้งตัวภาษา PHP และส่วนขยายอื่นๆ ที่จำเป็นเพื่อที่จะติดต่อและใช้งาน API หรือ Library ในอนาคต จุดเด่นสำคัญของ Laravel 5.1 คือ โค้ดจะมีลักษณะที่อ่านได้ง่ายและเป็นสัดเป็นส่วน มีการแบ่งโครงสร้างของเว็บไซต์ไว้อย่างชัดเจน อีกทั้งยังมีฟังก์ชั่นที่เกี่ยวข้องกับ Database ให้ใช้ได้อย่างสะดวก มีการใช้หลักการการทำงานตามสถาปัตยกรรมของ MVC อีกด้วย อีกทั้ง Laravel 5.1 ยังคงคำนึงความสะดวกในการใช้งานและวิธีการตั้งค่าในตอนเริ่มต้นอีกด้วย มีทั้งเอกสารและคำแนะนำที่ง่ายต่อการศึกษา จึงทำให้เราเชื่อว่า Laravel 5.1 เป็น PHP Framework ที่เหมาะสมในการพัฒนา GatherCloud เป็นอย่างมาก

ในส่วนของ Front-End Development หรือ การพัฒนาในส่วนของ GUI นั้น Laravel 5.1 ก็ได้มีรูปแบบการทำงานที่ช่วยแบ่งเบาภาระการทำงานได้อย่างดีเยี่ยม หรือก็คือ Blade Template ที่ถือได้ว่าช่วยในการออกแบบและจัดการข้อมูลบนหน้าเว็บไซต์ได้อย่างดี มีหลักการในการทำงานคือช่วยอำนวยความสะดวกในการส่งและรับข้อมูลระหว่าง View และ Controller และยังช่วยให้การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขียนและออกแบบเป็นไม่อย่างไม่ซ้ำซ้อนและเป็นระบบด้วยการนำหลาย Template มาประกอบกันได้

ฟังก์ชั่นที่เกี่ยวข้องกับ Database ที่ได้เกริ่นมาเบื้องต้นนั้น Laravel 5.1 ก็ได้มีส่วนเสริมอีกเช่นกัน โดยมีชื่อฟังก์ชั่นว่า Migrations โดยผู้พัฒนาจะสามารถสร้าง Table ต่างๆ ภายใน Database ได้อย่างสมบูรณ์ รวมถึง Relationship ระหว่าง Table ได้เช่นกัน ซึ่งเป็นฟังก์ชั่นที่สะดวกและง่ายต่อการนำไปใช้งานเป็นอย่างมาก หลังจากที่สร้างไฟล์ Migration เรียบร้อยแล้ว เราจะสามารถใช้คำสั่งใน Command Line และใส่ชุดคำสั่งเข้าไปก็จะทำให้สามารถสร้าง Database ที่มีความสัมพันธ์และข้อมูลต่างๆ เช่นเดียวกันกับตอนที่พัฒนา ถือว่าเป็น Version Control สำหรับ Database Schema

อีกหนึ่งสาเหตุที่ทำให้ Laravel 5.1 เป็น Framework ที่สนับสนุนการทำงานของสถาปัตยกรรมแบบ MVC ก็เพราะใช้ Eloquent ORM ซึ่งเป็นเทคนิคสำหรับการเขียนโปรแกรมเป็นการจำลองเชิงวัตถุเพื่อใช้ร่วมกับฐานข้อมูล โดยตัวโมเดลจะสามารถทำเข้าถึงข้อมูลตรงตามตารางฐานข้อมูล รวมถึงการเพิ่มหรือลบข้อมูลลงฐานข้อมูลจริงๆ อีกด้วย

ในการใช้งาน Laravel 5.1 นั้นจะมีการเตรียมสภาพแวดล้อมตามความต้องการของ Laravel 5.1 ได้แก่

- PHP Version 5.5.9 or More
- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension

นอกจากที่กล่าวมาเบื้องต้น Laravel 5.1 ยังมีฟังก์ชั่นและ Features อีกมากมายที่ช่วยอำนวยความสะดวกที่ยังไม่ได้กล่าวถึงในการพัฒนาเว็บไซต์ด้วยภาษา PHP โดยการใช้ Laravel 5.1 จะส่งผลให้สามารถพัฒนา GatherCloud ได้ตรงตามรูปแบบของ MVC อีกทั้งยังวางรากฐานของระบบให้สามารถดูแลและจัดการได้ง่ายภายในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ทำการสร้างโปรเจคของ Laravel 5.1 จะได้โครงสร้างไฟล์ดังรูปที่ 4.14 [9]

รูปที่ 4.14 โครงสร้างไฟล์ของ Laravel 5.1

ซึ่งจะมีรายละเอียดคร่าวๆ ได้แก่

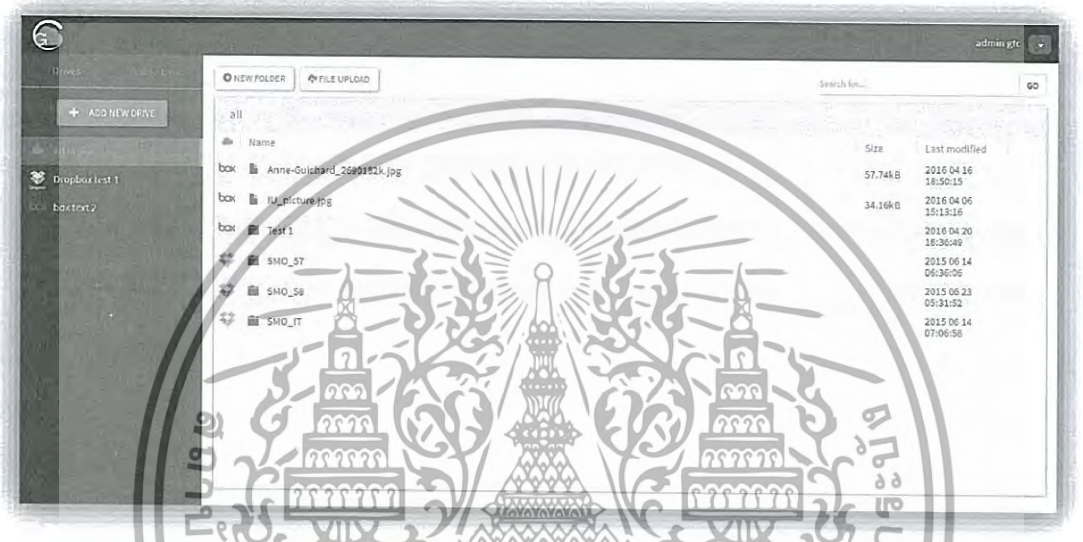
- app : จะเป็นโฟลเดอร์ที่เก็บการทำงานที่เป็นแกนหลักของ Application
- bootstrap : เก็บไฟล์ที่ทำหน้าที่เริ่มต้นการทำงานของ Framework
- config : เก็บข้อมูลการตั้งค่าและไฟล์ที่กำหนดค่าต่างๆ ของ Application
- database : เก็บฟังก์ชั่นการทำงานของ Framework ที่เรียกว่า Migration และ Seed ที่ช่วยอำนวยความสะดวกในด้าน Database
- public : เก็บไฟล์ที่เป็นไฟล์ที่สามารถเข้าถึงได้จาก URL และข้อมูลที่เป็นส่วนประกอบของเว็บไซต์ เช่น CSS, JavaScript, รูปภาพ และอื่นๆ
- resources : ใช้เก็บไฟล์ที่เป็นส่วนของ Views และส่วนประกอบอื่นๆ
- storage : เป็นโฟลเดอร์ที่ใช้เก็บไฟล์ที่ใช้ในการประมวลผล Blade Template และส่วนประกอบอื่นๆ ที่เกี่ยวข้องกับการใช้งาน Framework
- tests : ใช้เก็บไฟล์ที่เกี่ยวข้องกับการทดสอบ Application
- vendor : เป็นไฟล์ที่ไว้เก็บ Library อื่นๆ ที่มีการทำงานร่วมกับ composer

หลักการทำงานของ Laravel 5.1 คร่าวๆ หรือวงจรชีวิตของ Request ที่ผ่าน Framework นั้นจะต้องมีเข้าถึงไฟล์ public/index.php ซึ่งจะถูกนำทางมาโดย Server ของเราเอง และภายในไฟล์นั้น

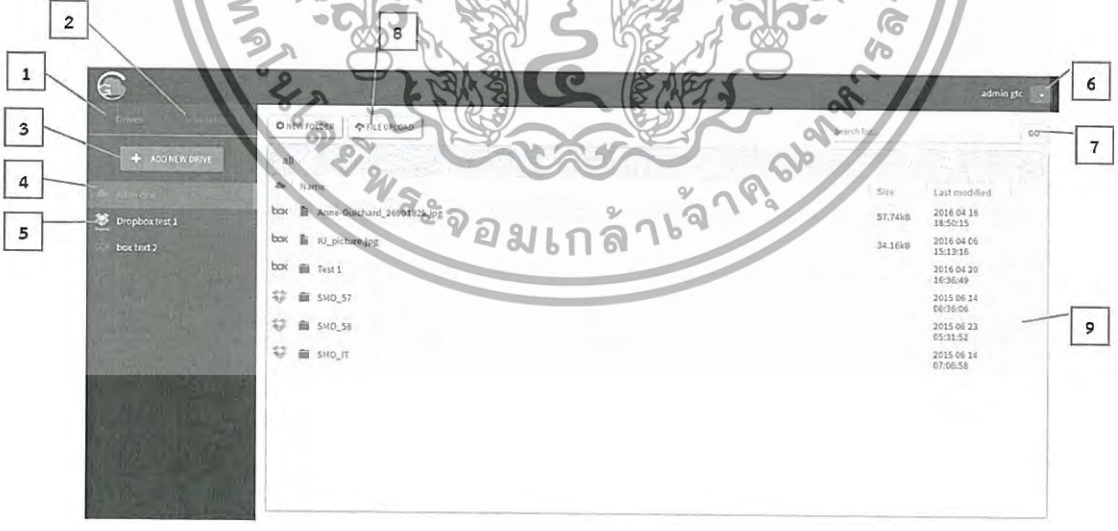
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเป็นจุดเริ่มต้นในการนำทางไปสู่ส่วนอื่นๆ ของ Framework โดยจะถูกโยนเข้าไปหา app/Http/Kernel.php ที่ทำหน้าที่ตรวจสอบสิ่งแวดล้อมที่เกี่ยวข้องของ Application และยังทำหน้าที่ติดต่อกับ Middleware ต่างๆ ที่ทำหน้าที่ใน Application เช่น ยืนยัน CSRF Token เพื่อป้องกันการโจมตีแบบ Cross-site Request Forgery เป็นต้น ก่อนที่จะส่ง Request ไปต่อยังไฟล์ handle ที่ทำหน้าที่รับ Request และตอบกลับด้วย Response

4.2.2 Graphic User Interface (GUI)



รูปที่ 4.15 หน้าจัดการไฟล์ของ GatherCloud



รูปที่ 4.16 หน้าจัดการไฟล์ของ GatherCloud พร้อมตัวบ่งชี้การใช้งาน

จากการออกแบบส่วนเชื่อมต่อกับผู้ใช้ (GUI) นั้นจะพบว่า หลังจากผู้ใช้ได้ทำการล็อกอินเข้าสู่ระบบเรียบร้อยแล้วจะพบกับหน้าหลักสำหรับการใช้งานเพื่อจัดการไฟล์และเข้าถึงฟังก์ชั่นต่างๆ ของระบบตามรูปที่ 4.16 ซึ่ง จะอธิบายการทำงานของส่วนเชื่อมต่อกับผู้ใช้โดยสังเขปดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. แท็บของส่วนที่จัดการการเชื่อมต่อกับแหล่งเก็บข้อมูลบนกลุ่มเมฆ โดยจะแสดงข้อมูลที่เกี่ยวข้องด้านล่าง เช่น ปุ่ม ADD NEW DRIVE และรายชื่อการเชื่อมต่อที่เราได้เพิ่มมาก่อนหน้า

2. แท็บของส่วนที่จัดการฟังก์ชั่นมุมมองของข้อมูลที่เกิดจากการเลือกไฟล์หรือข้อมูลที่แตกต่างจากแหล่งเก็บข้อมูลบนกลุ่มเมฆที่หลากหลายเพื่อใช้ในการแบ่งปันให้ผู้ใช้คนอื่นสามารถเข้าถึงได้

3. ปุ่ม ADD NEW DRIVE ที่ใช้เพิ่มการเชื่อมต่อกับแหล่งเก็บข้อมูลบนกลุ่มเมฆผ่านระบบซึ่งต้องผ่านการ Authorization หรือการได้รับอนุญาตจากผู้ใช้ก่อน

4. หน้า All in One ที่จะแสดงข้อมูลไฟล์จากทุกการเชื่อมต่อที่ผู้ใช้ได้เชื่อมต่อกับระบบภายในหน้าเดียว

5. ปุ่มเพื่อเข้าถึงการเชื่อมต่อกับผู้ให้บริการที่ผู้ใช้ได้เพิ่มเข้ามาสู่ระบบ เพื่อแสดงข้อมูลภายในการเชื่อมต่อนั้นๆ โดยจะแสดงตามชื่อที่ผู้ใช้ตั้งไว้ในตอนแรก

6. ปุ่ม Dropdown Menu โดยมีฟังก์ชั่นให้เลือก 3 อย่าง ได้แก่ ตั้งค่าผู้ใช้ (Profile Setting), ตั้งค่าและจัดการเกี่ยวกับการเชื่อมต่อกับผู้ให้บริการ (Cloud Connection), และลงชื่อออกจากระบบ (Log Out)

7. ปุ่ม Search File ใช้ในการค้นหาไฟล์จากแหล่งเก็บข้อมูลบนกลุ่มเมฆในทุกๆ การเชื่อมต่อที่ผู้ใช้ได้เพิ่มกับระบบ และแสดงผลออกมาในตารางหลัก

8. ปุ่ม File Upload ที่ใช้การอัพโหลดไฟล์ผ่านระบบ โดยไฟล์นั้นจะเก็บไว้ในแหล่งเก็บข้อมูลบนกลุ่มเมฆที่เราเลือกไว้ นอกจากนี้ยังเป็นมีการทำงานร่วมกับฟังก์ชั่นที่ช่วยในการตรวจสอบความซ้ำซ้อนของไฟล์อีกด้วย

9. ส่วนตารางหลักที่ใช้แสดงรายละเอียดของไฟล์และโฟลเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

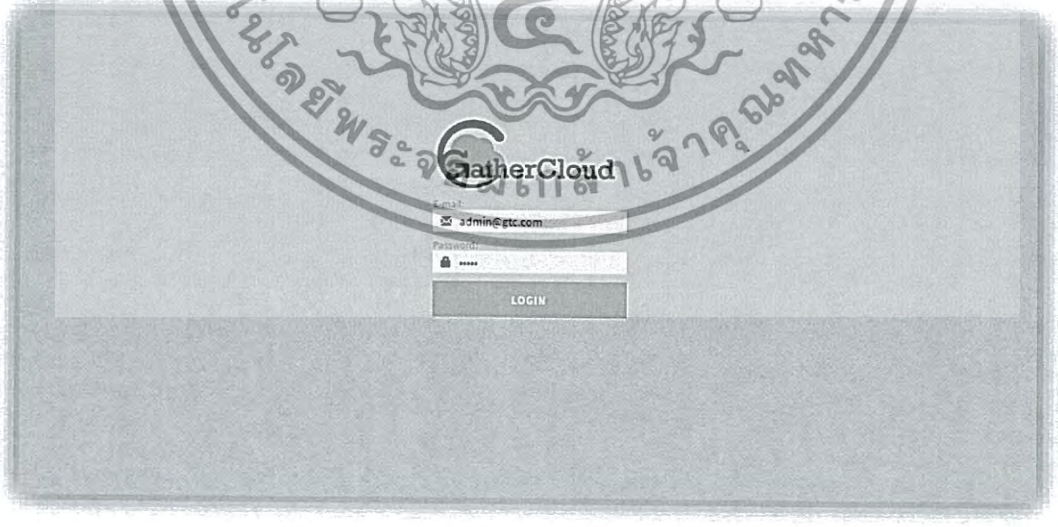
4.3.3 ผลการทำงานของระบบต้นแบบ

4.3.3.1 ระบบ Authentication

หลังจากที่พัฒนาระบบต้นแบบขึ้นมา ฟังก์ชั่นการทำงานอย่างแรกที่ผู้ใช้จะได้พบคือ ระบบการยืนยันตัวตนหรือ Authentication หรือก็คือ การล็อกอิน โดยจะสามารถสมัครสมาชิกได้ดังรูปที่ 4.17

รูปที่ 4.17 หน้าสมัครสมาชิก

เมื่อได้ทำการสมัครสมาชิกเสร็จเรียบร้อยผู้ใช้ก็จะมีบัญชีผู้ใช้ของระบบ ซึ่งสามารถนำไปล็อกอินได้และสามารถเข้าถึงฟังก์ชั่นอื่นๆ ของระบบดังรูปที่ 4.18



รูปที่ 4.18 หน้าล็อกอิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3.2 ระบบจัดการข้อมูลผู้ใช้

หลังจากที่ผู้ใช้ได้เข้าสู่ระบบแล้ว ผู้ใช้สามารถเลือกเครื่องหมายลูกศรด้านบนขวามือเพื่อเข้าถึงฟังก์ชั่นในการจัดการข้อมูลส่วนตัวของผู้ใช้ได้ โดยเลือก Profile Setting ตามรูปที่ 4.19

รูปที่ 4.19 เมนู Dropdown

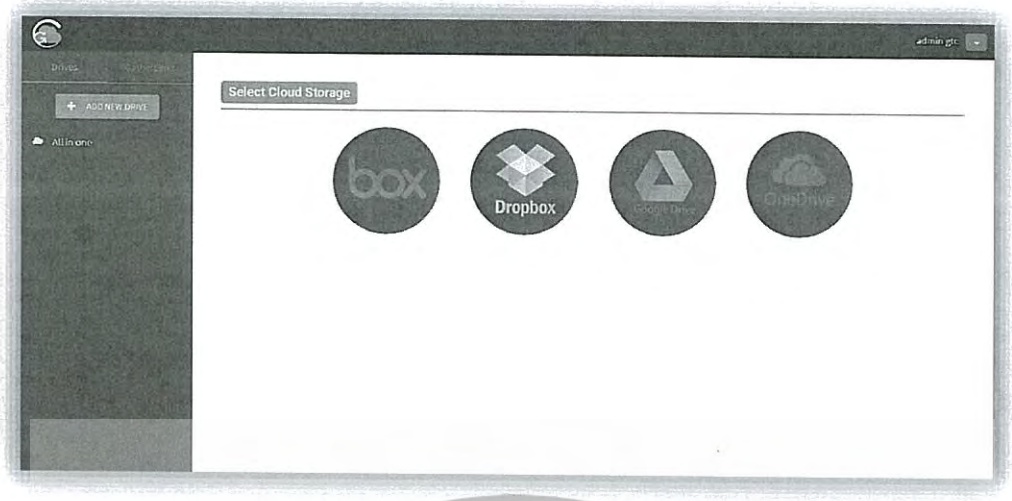
และแก้ไขข้อมูลของผู้ใช้ที่มีในระบบได้ โดยมีลักษณะตามรูปที่ 4.20

รูปที่ 4.20 หน้าจัดการข้อมูลผู้ใช้

4.3.3.3 ระบบจัดการการเชื่อมต่อกับผู้ให้บริการแหล่งเก็บข้อมูล

ในการใช้งานตัวระบบนั้น ผู้ใช้จำเป็นต้องมีการเพิ่มการเชื่อมต่อกับแหล่งเก็บข้อมูลบนกลุ่มเมฆ เพื่อที่จะให้ระบบสามารถเข้าถึงได้ ซึ่งผู้ใช้จำเป็นต้องกรอกรายละเอียดที่เกี่ยวข้องและอนุญาตให้ระบบสามารถเข้าถึงข้อมูลส่วนตัวภายในบัญชีผู้ใช้ได้ โดยเลือกปุ่ม ADD NEW DRIVE ซึ่งระบบจะเก็บข้อมูลที่เกี่ยวข้องกับการเชื่อมต่อไว้ในฐานข้อมูลของระบบเพื่อสามารถเรียกใช้ได้ โดยเลือกได้ในลักษณะตามรูปที่ 4.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 หน้าเลือกผู้ให้บริการสำหรับเพิ่มการเชื่อมต่อกับระบบ

รูปที่ 4.22 หน้ายินยอมสิทธิการเข้าถึงของผู้ให้บริการ Box

รูปที่ 4.23 หน้ายินยอมสิทธิการเข้าถึงของผู้ให้บริการ Dropbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.24 หน้ายินยอมสิทธิการเข้าถึงของผู้ให้บริการ Google Drive

รูปที่ 4.25 หน้ายินยอมสิทธิการเข้าถึงของผู้ให้บริการ OneDrive

และเมื่อใช้งานตัวระบบไปเรื่อยๆ ผู้ใช้จะสามารถจัดการกับการเชื่อมต่อที่ผู้ใช้เคยเพิ่มขึ้น โดยเลือก Cloud Connection ใน Dropdown Menu ตามรูปที่ 4.19 และจะเข้าสู่หน้าสำหรับจัดการการเชื่อมต่อกับแหล่งเก็บข้อมูลบนกลุ่มเมฆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.26 หน้าจัดการการเชื่อมต่อกับผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆ

4.3.3.4 ระบบจัดการไฟล์ในแหล่งเก็บข้อมูลแบบแยก

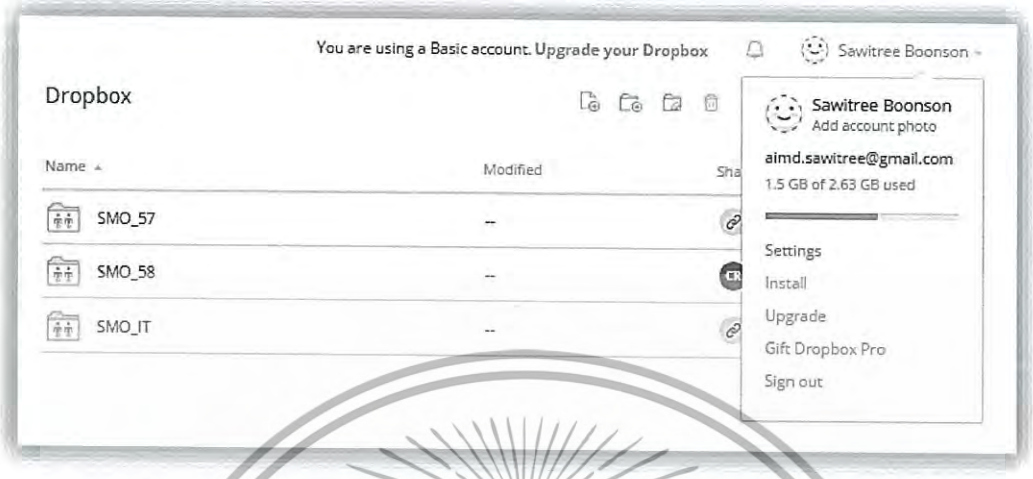
หลังจากที่ผู้ใช้ได้เพิ่มการเชื่อมต่อกับแหล่งเก็บข้อมูลบนกลุ่มเมฆเรียบร้อยแล้ว จะทำให้ระบบสามารถเข้าถึงข้อมูลที่อยู่บนแหล่งเก็บข้อมูลของผู้ใช้ได้ และนำมาแสดงผลบน GUI ของระบบได้ ซึ่งผู้ใช้จำเป็นต้องเลือกแหล่งเก็บข้อมูลที่ต้องการจะแสดงผลจากรายชื่อด้านข้าง ตามรูปที่ 4.27

รูปที่ 4.27 หน้าจัดการไฟล์ในแหล่งเก็บข้อมูลแบบแยกของระบบ

รายชื่อของข้อมูลที่แสดงผลอยู่นั้น เป็นข้อมูลที่ดึงมาจาก API ของผู้ให้บริการ โดยใช้ Access Token ที่ผู้ใช้ได้อนุญาตให้เราเข้าถึงข้อมูลภายในบัญชีผู้ใช้ โดยการแสดงข้อมูลทุกครั้งจะมีการร้องขอไปยังผู้ให้บริการเพื่อให้ได้ข้อมูลที่เป็นปัจจุบันและถูกต้องอีกด้วย ซึ่งจะผ่านการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมวลผลเบื้องต้นภายในระบบและนำมาแสดงภายในหน้าจัดการไฟล์ในแหล่งเก็บข้อมูลแบบแยก โดยข้อมูลนั้นไม่ได้มีอะไรสูญหายและมีความถูกต้อง



รูปที่ 4.28 หน้าจัดการไฟล์ในเว็บไซต์ Dropbox ที่ได้เพิ่มการเชื่อมต่อเข้าไป

ในการจัดการไฟล์นั้นผู้ใช้จะสามารถคลิกขวาที่ไฟล์ เพื่อแสดงหน้าเมนูที่ใช้ในการเลือกฟังก์ชั่นการทำงานที่จะสามารถจัดการไฟล์ภายในแหล่งเก็บข้อมูลบนกลุ่มเมฆได้ โดยสามารถดาวน์โหลดไฟล์ แชร์ไฟล์ไปให้คนอื่น ลบ แก้ไข ย้าย และคัดลอกไฟล์ได้

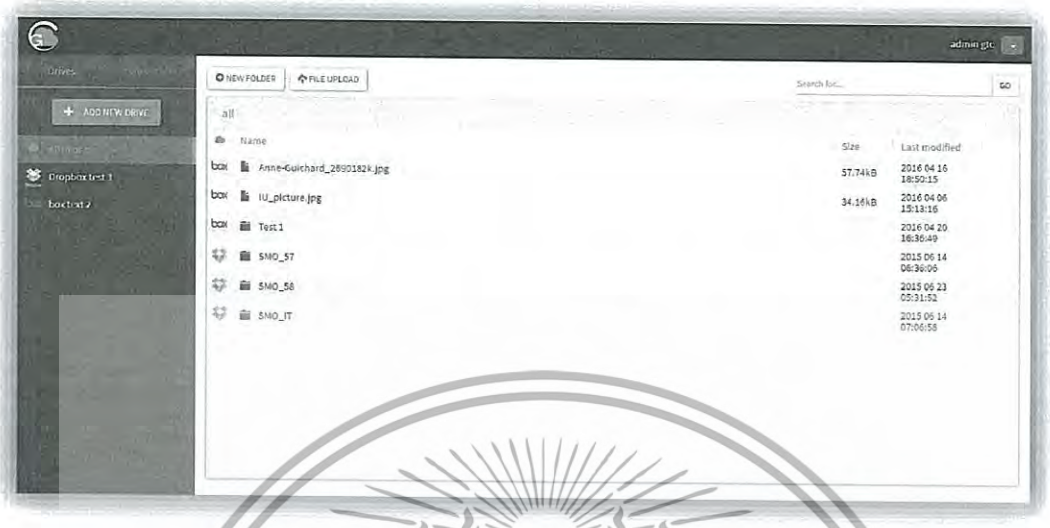
รูปที่ 4.29 หน้าเมนูในการจัดการไฟล์

4.3.3.5 ระบบจัดการไฟล์ในแหล่งเก็บข้อมูลแบบรวม

ฟังก์ชั่นการใช้งานหลักของระบบตามแนวคิดหลักในการพัฒนาหรือก็คือ การมองให้แหล่งเก็บข้อมูลบนกลุ่มเมฆหลายๆ แห่งทำงานเป็นหนึ่งเดียวกันภายใต้การดูแลของระบบ ซึ่งระบบมีการจัดการและมีการแสดงผลให้ข้อมูลภายในแหล่งเก็บข้อมูลบนกลุ่มเมฆที่ผู้ใช้ได้ลงทะเบียนไว้กับระบบได้ถูกแสดงไว้ในหน้าจัดการไฟล์แบบรวม ซึ่งทำให้ผู้ใช้ไม่จำเป็นต้องคำนึงถึงที่มาของไฟล์และสามารถจัดการไฟล์ได้อย่างสะดวก โดยหลักการในดึงข้อมูลนั้น ระบบได้ทำการดึงข้อมูลมาจาก API ของผู้ให้บริการและนำข้อมูลที่ได้มาประมวลผลให้ข้อมูลที่ได้อยู่ในลักษณะเดียวกันและเป็นมาตรฐานของระบบ เพื่อนำมาแสดงผล และมีฟังก์ชั่นในการจัดการไฟล์เช่นเดียวกับการจัดการ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์แบบแยก เช่น เปลี่ยนชื่อไฟล์ ลบ อัพโหลด หรือดาวน์โหลดก็ตาม เป็นต้น โดยจะแบ่งแยกที่มาของไฟล์ด้วย icon ที่เป็น logo ของผู้ให้บริการอยู่ด้านหน้าไฟล์

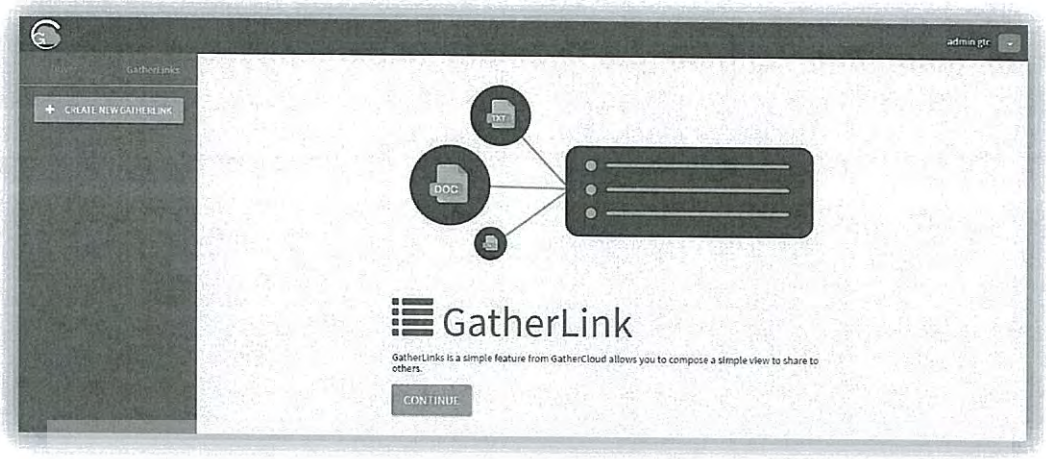


รูปที่ 4.30 หน้าจัดการไฟล์ในแหล่งเก็บข้อมูลแบบรวมของระบบ

4.3.3.6 ระบบจัดการมุมมองของไฟล์ (GatherLinks)

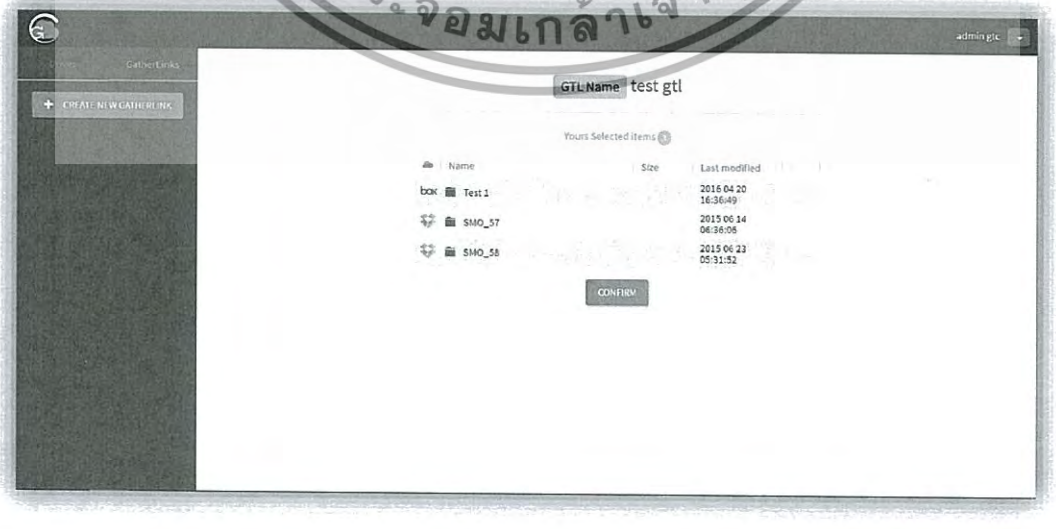
ฟังก์ชั่น GatherLink เป็นฟังก์ชั่นที่ใช้ในการสร้างมุมมองของข้อมูลซึ่งมีจุดประสงค์คือการแชร์ให้คนอื่นได้เห็นมุมมองของข้อมูลแบบที่เราต้องการ ซึ่งในการสร้างนั้นจะไม่จำกัดว่าข้อมูลต้องอยู่ในแหล่งเก็บข้อมูลเดียวกัน จึงเป็นการแชร์มุมมองของข้อมูลจากหลายๆ แหล่งเก็บข้อมูลไว้ในที่เดียวได้ ช่วยอำนวยความสะดวกในการแบ่งปันข้อมูล ผู้ที่สามารถเข้าถึงได้นั้นจะต้องเข้าผ่าน URL ที่ทางระบบได้สร้างขึ้นและผู้ที่สามารถเข้าถึงทำได้เพียงมองและดาวน์โหลดไฟล์ จะไม่สามารถแก้ไขหรือเปลี่ยนแปลงใดๆ ได้ ในแง่ของการพัฒนานั้น ระบบจะทำการเก็บข้อมูลที่ใช้อธิบายไฟล์ที่ถูกเลือกไว้ในฐานข้อมูล และสร้างขึ้นมาเป็นมุมมองโดยมี Token ที่เกิดจากการเข้ารหัส String เพื่อกำหนดให้ผู้ใช้ที่มี Token นี้สามารถเข้าถึงได้ และให้เจ้าของมุมมองสามารถแก้ไขได้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.31 หน้าแรกของฟังก์ชั่น GatherLinks

รูปที่ 4.32 หน้าสำหรับเลือกไฟล์เพื่อสร้างมุมมอง



รูปที่ 4.33 หน้าสรุปผลการเลือกไฟล์เพื่อสร้างมุมมอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.34 หน้าแสดงมุมมองจากผู้ใช้ที่เป็นเจ้าของ

รูปที่ 4.35 หน้าแสดงมุมมองจากผู้ใช้อื่น

4.3.3.7 ระบบค้นหาไฟล์

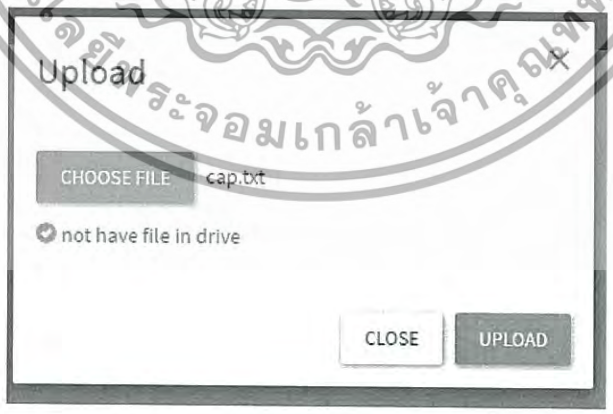
ระบบได้พัฒนาฟังก์ชั่นการค้นหาไฟล์ที่ต้องการจากทุกแหล่งเก็บข้อมูลบนกลุ่มเมฆ เพื่อลดปัญหาในการเข้าถึงแหล่งเก็บข้อมูลบนกลุ่มเมฆจากผู้ให้บริการที่ต่างกัน ทำให้เกิดความสะดวกสบายและช่วยทำให้ผู้ใช้เจอไฟล์ที่ต้องการได้เร็วมากยิ่งขึ้น โดยวิธีในการค้นหา คือ ระบบจะรับ Keyword จากผู้ใช้ และค้นหาผ่าน API โดยจะค้นหาจากทุกแหล่งเก็บข้อมูลที่ผู้ใช้ได้ลงทะเบียนไว้กับระบบ และนำผลลัพธ์ที่ได้มาประมวลผลและแสดงผลออกมาเป็นผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.36 หน้าแสดงผลลัพธ์การค้นหาไฟล์

4.3.3.8 ระบบการอัพโหลดไฟล์

ในการอัพโหลดไฟล์เพื่อเก็บไว้บนแหล่งเก็บข้อมูลบนกลุ่มเมฆนั้นปกติผู้ใช้จะทำการอัพโหลดโดยตรงผ่านหน้าเว็บที่ผู้ให้บริการได้เตรียมไว้ให้ใช้หรือซิงค์ผ่าน Desktop Application ที่ได้ติดตั้งไว้บนคอมพิวเตอร์ของผู้ใช้ แต่ถ้าหากเลือกใช้เว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆเป็นตัวกลางในการอัพโหลดไฟล์เพื่อเก็บข้อมูลไว้บนแหล่งเก็บข้อมูลบนกลุ่มเมฆ ระบบจะมีฟังก์ชั่นในการตรวจสอบความซ้ำซ้อนของไฟล์ที่ผู้ใช้เลือกอัพโหลด อีกทั้งยังสามารถเรียงลำดับในการเลือกผู้ให้บริการเพื่ออัพโหลดโดยอ้างอิงจากพื้นที่คงเหลือที่เก็บได้ของแต่ละแหล่งเก็บข้อมูลอีกด้วย



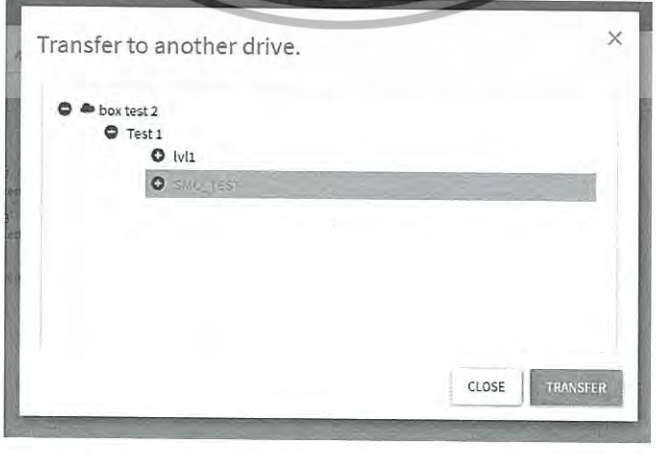
รูปที่ 4.37 หน้าแสดงผลลัพธ์การตรวจสอบความซ้ำซ้อนของไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.38 หน้าแสดงผลลัพธ์คำนวนพื้นที่คงเหลือของแหล่งเก็บข้อมูลบนกลุ่มเมฆ

4.3.3.9 ระบบการโอนย้ายไฟล์

ในการโอนย้ายไฟล์ข้ามแหล่งเก็บข้อมูลนั้น พัฒนาขึ้นมาเพื่ออำนวยความสะดวกให้กับผู้ใช้สามารถทำได้อย่างสะดวก โดยผู้ใช้จะสามารถโอนไฟล์หรือข้อมูลได้จากการคลิ๊กขวาที่ไฟล์และเลือก Transferและหลังจากนั้นผู้ใช้จะต้องเลือกที่อยู่บนแหล่งเก็บข้อมูลอื่น เพื่อที่จะทำการย้ายไฟล์ ตัวระบบนั้นจะทำหน้าที่ในการดาวน์โหลดไฟล์มาเก็บไว้ที่เครื่องเซิร์ฟเวอร์ชั่วคราวและหลังจากนั้นจะทำการอัพโหลดไฟล์นั้นขึ้นไปยังที่อยู่ที่ผู้ใช้ระบุบนแหล่งเก็บข้อมูลที่ต้องการ และเมื่อระบบทำงานเสร็จจะทำการลบไฟล์ชั่วคราวนั้นออกจากเซิร์ฟเวอร์ ซึ่งฟังก์ชั่นดังกล่าวเป็นฟังก์ชั่นที่ต้องใช้เวลาในการทำงานระดับหนึ่งขึ้นอยู่ขนาดของไฟล์ที่ต้องการจะโอนย้าย ซึ่งอาจจะส่งผลให้ระบบตอนสนองช้ากว่าปกติ



รูปที่ 4.39 หน้าการเลือกปลายทางการโอนย้ายไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

โครงงานนี้จัดทำขึ้นเพื่อนำเสนอเว็บไซต์ที่สามารถจัดการแหล่งเก็บข้อมูล 5 แหล่งบนกลุ่มเมฆ ได้โดยในการทำงานของระบบจะเป็นการทำงานบนเว็บไซต์ พัฒนาด้วยภาษา PHP และใช้ PHP Framework ชื่อ Laravel 5.1 ช่วยในการพัฒนาและทำงานตามรูปแบบสถาปัตยกรรมแบบ MVC (Model-View-Controller)

การทำงานหลักของระบบจะมีการทำงานเพื่อตอบสนองปัญหาความยุ่งยากวุ่นวายในการเข้าถึงและจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆของผู้ใช้ อีกทั้งยังช่วยแก้ปัญหาความซ้ำซ้อนของข้อมูลที่อยู่บนแหล่งเก็บข้อมูลบนกลุ่มเมฆ โดยหลักการทำงานของระบบจะเป็นการใช้หลักการ Single Sign-on ให้กับผู้ใช้เพื่อลงทะเบียนแหล่งเก็บข้อมูลบนกลุ่มเมฆต่างๆ กับระบบ และหลังจากนั้นตัวระบบจะทำงานเชื่อมต่อกับ API ต่างๆ ของแหล่งเก็บข้อมูลบนกลุ่มเมฆ ได้แก่ Dropbox, Copy, Onedrive, Box และ Google Drive เพื่อรวบรวมและจัดการไฟล์ข้อมูลของผู้ใช้ผ่านระบบ GatherCloud ในที่เดียว อีกทั้งแนวคิดหลักของระบบจะพยายามมองแหล่งเก็บข้อมูลบนกลุ่มเมฆหลายๆ บริการเป็นเพียงแหล่งเก็บข้อมูลเดียว เพื่ออำนวยความสะดวกในการเข้าถึงและจัดการไฟล์ข้อมูลให้กับผู้ใช้มากยิ่งขึ้น

ปัจจุบันได้พัฒนาระบบจากการศึกษา API และออกแบบระบบให้สอดคล้องกับการเรียกใช้ API ต่างๆ ได้แก่ Dropbox, Copy, Onedrive, Box และ Google Drive รวมถึงออกแบบส่วนเชื่อมต่อกับผู้ใช้ที่เหมาะสม (User Interface) กับฟังก์ชั่นการทำงานของระบบ ทั้งนี้ระหว่างการพัฒนาก็ได้พบกับปัญหาต่างๆ และได้เรียนรู้อะไรใหม่ๆ มากมาย ไม่ว่าจะเป็นทักษะในการออกแบบระบบที่ดี การแก้ไขปัญหาและทำให้ระบบสามารถทำงานได้ตามความต้องการ เรียนรู้เทคโนโลยีใหม่ๆ นอกจากนี้ยังได้วางแผนที่จะพัฒนาระบบโดยเลือกที่จะบันทึกขั้นตอนในการเพิ่มและลดจำนวนผู้ให้บริการที่ระบบรองรับอีกด้วย เนื่องจากในระหว่างเวลาที่ได้พัฒนา ผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆ Copy ได้หยุดให้บริการไปในวันที่ 1 พฤษภาคม พ.ศ. 2559

5.2 ปัญหาและอุปสรรคที่พบในการพัฒนาระบบ

1. ในการพัฒนาระบบ จะต้องมีการศึกษาเทคโนโลยีใหม่หลายๆ ด้าน ทั้ง Library ที่ต้องใช้และ API ที่แต่ละ Cloud Storage ได้จัดเตรียมไว้ให้ ซึ่งเนื้อหาที่ต้องทำการศึกษามีความหลากหลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และแตกต่างกัน จึงต้องใช้เวลาในการศึกษานานกว่าที่คาดการณ์ไว้ จึงส่งผลกระทบต่อแผนในการพัฒนาระบบที่วางไว้ในตอนแรก

2. ในการพัฒนาระบบ GatherCloud ให้ครบถ้วนสมบูรณ์จำเป็นต้องมีความรู้เรื่องเครือข่ายคอมพิวเตอร์และความปลอดภัยบนเครือข่าย ซึ่งผู้พัฒนายังไม่มีความรู้และความเชี่ยวชาญมากพอในด้านนี้ จึงทำให้ต้องใช้เวลาในการศึกษาและพัฒนาเพิ่มเติมมากยิ่งขึ้น

3. เนื่องจากผู้ให้บริการแหล่งเก็บข้อมูลมีจำนวนมาก จึงทำให้ต้องมีการออกแบบระบบให้สามารถรองรับการพัฒนาต่อในอนาคตและง่ายต่อการดูแลและการต่อเติมที่อาจจะเกิดขึ้น จึงทำให้ทีมผู้พัฒนาต้องทำการศึกษาถึงหลักการในการออกแบบให้เข้าใจมากยิ่งขึ้น เพื่อนำความรู้ที่ได้มาปรับให้การออกแบบระบบสอดคล้องกับหลักการดังกล่าว

5.3 ข้อจำกัดของระบบ

1. เนื่องจากต้องมีการออกแบบระบบสามารถเข้าถึงและทำงานร่วมกับแหล่งเก็บข้อมูลบนก้อนเมฆได้หลายที่ พร้อมกัน รวมถึงข้อจำกัดของ API ที่ผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆแต่ละเจ้าได้จัดเตรียมจึงทำให้ไม่สามารถใช้ฟังก์ชั่นหรือฟีเจอร์ที่เป็นจุดเด่นของแต่ละเจ้าได้อย่างเต็มที่

2. หากมีการเปลี่ยนแปลงเวอร์ชั่นของ API ของผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆอาจจะทำให้ระบบทำงานบกพร่อง จึงต้องมีการดูแลและอัพเดทอย่างสม่ำเสมอ

3. ระบบจะต้องอาศัยการประมวลผลข้อมูลที่ร้องขอมาจากผู้ให้บริการอีกต่อหนึ่ง ดังนั้นรูปแบบการทำงานจึงไม่สามารถอยู่ในลักษณะ Real-time กับผู้ใช้ได้อย่างเต็มที่

5.4 ข้อเสนอแนะและแนวทางในการพัฒนาระบบในอนาคต

1. สามารถพัฒนาระบบให้สนับสนุนและคำนึงถึงเรื่องความปลอดภัยทางเครือข่ายคอมพิวเตอร์มากยิ่งขึ้น

2. สามารถศึกษา API ให้ลึกซึ้งมากยิ่งขึ้น และพยายามดึงฟังก์ชั่นการทำงานของแต่ละ API ให้ออกมาได้ประสิทธิภาพสูงสุด

3. สามารถที่จะพัฒนาต่อในรูปแบบของ Desktop Application และสนับสนุนการซิงค์กับข้อมูลที่อยู่เครื่อง Local ที่ทำงานได้อย่างรวดเร็วและถูกต้อง

4. ในการพัฒนาระบบให้รองรับผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆเพิ่ม ควรจะศึกษา API ให้ชัดเจน และพัฒนา Model ของการเชื่อมต่อ โดยทำการ Implement Interface Class ที่ชื่อว่า ModelInterface ซึ่งทำงานภายใต้ Model ที่ชื่อว่า Provider

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. หลังจากที่ปล่อยให้ระบบทำงานและทำการเก็บข้อมูลเชิงสถิติเกี่ยวกับการทำงานของ API ของผู้ให้บริการแหล่งเก็บข้อมูลต่างๆ จะสามารถนำข้อมูลมาพัฒนาระบบให้สามารถทำงานได้ดียิ่งขึ้นอีกทั้งยังสามารถนำไปสู่งานวิจัยอื่นๆ ที่เกี่ยวข้องได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Estanislau Trepat (etrepat). **"Baum: Nested Set pattern for Laravel 5's Eloquent ORM."** [Online]. Available: https://github.com/etrepat/baum. 2016

[2] MultCloud Developer Team. **"MultCloud."** [Online]. Available: https://multcloud.com. 2016

[3] Dropbox Development Team. **"Dropbox Core API v1 Documentation."** [Online]. Available: https://www.dropbox.com/developers-v1/core/docs. 2016

[4] Barracuda Technical Team. **"Barracuda Copy API Documentation."** [Online]. Available: https://techlib.barracuda.com/Copy/CopyAPIGuide. 2016

[5] Microsoft. **"Develop with the OneDrive API."** [Online]. Available: https://dev.onedrive.com/README.htm. 2016

[6] Box Developer. **"Box Content API Docs."** [Online]. Available: https://developers.box.com/docs-overview. 2016

[7] Google Developers. **"API Client Library for PHP (Beta)."** [Online]. Available: https://developers.google.com/api-client-library/php/. 2016

[8] Taylor Otwell. **"Laravel 5.1 Documentation."** [Online]. Available: http://laravel.com/docs/5.1. 2015

[9] Jeffrey Way. **"Laracast: Laravel 5 Fundamentals."** [Online]. Available: https://laracasts.com/series/laravel-5-fundamentals. 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ขั้นตอนการสร้างแอพพลิเคชั่นของแต่ละผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆ

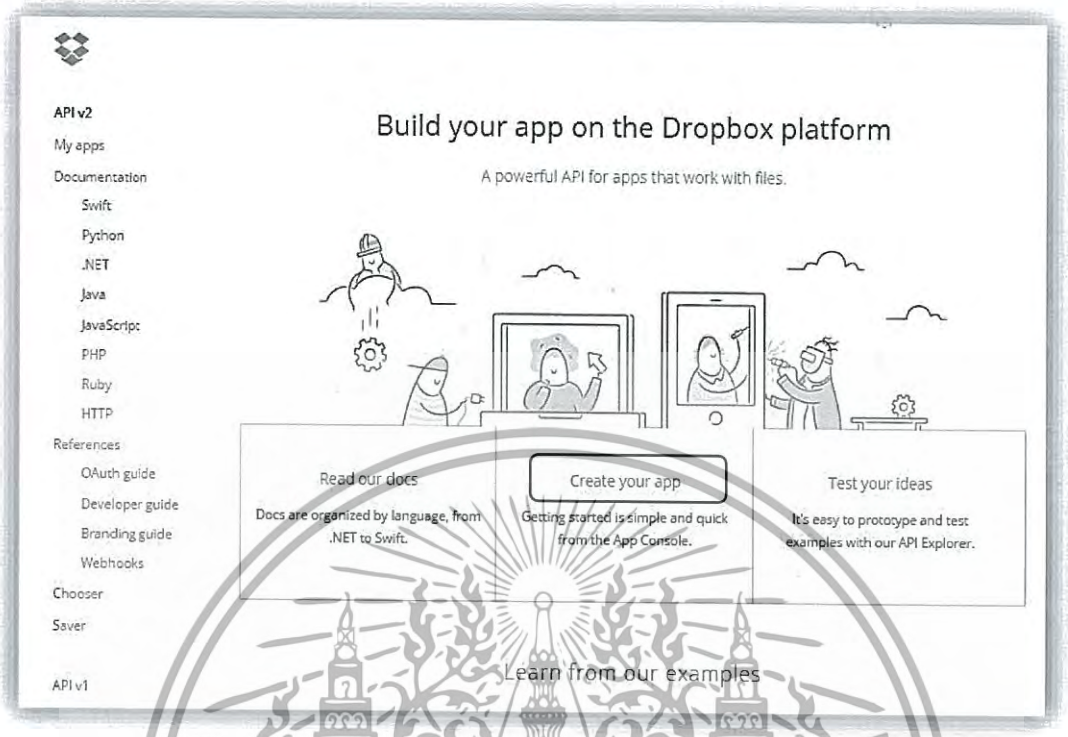
1. Dropbox

1.1 เข้าสู่ระบบที่เว็บ https://www.dropbox.com แล้วเข้าไปในส่วนของ Developers หรือ https://www.dropbox.com/developers

รูปที่ ก.1 การเข้าถึงหน้า Developers ของ Dropbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 เลือก Create your app

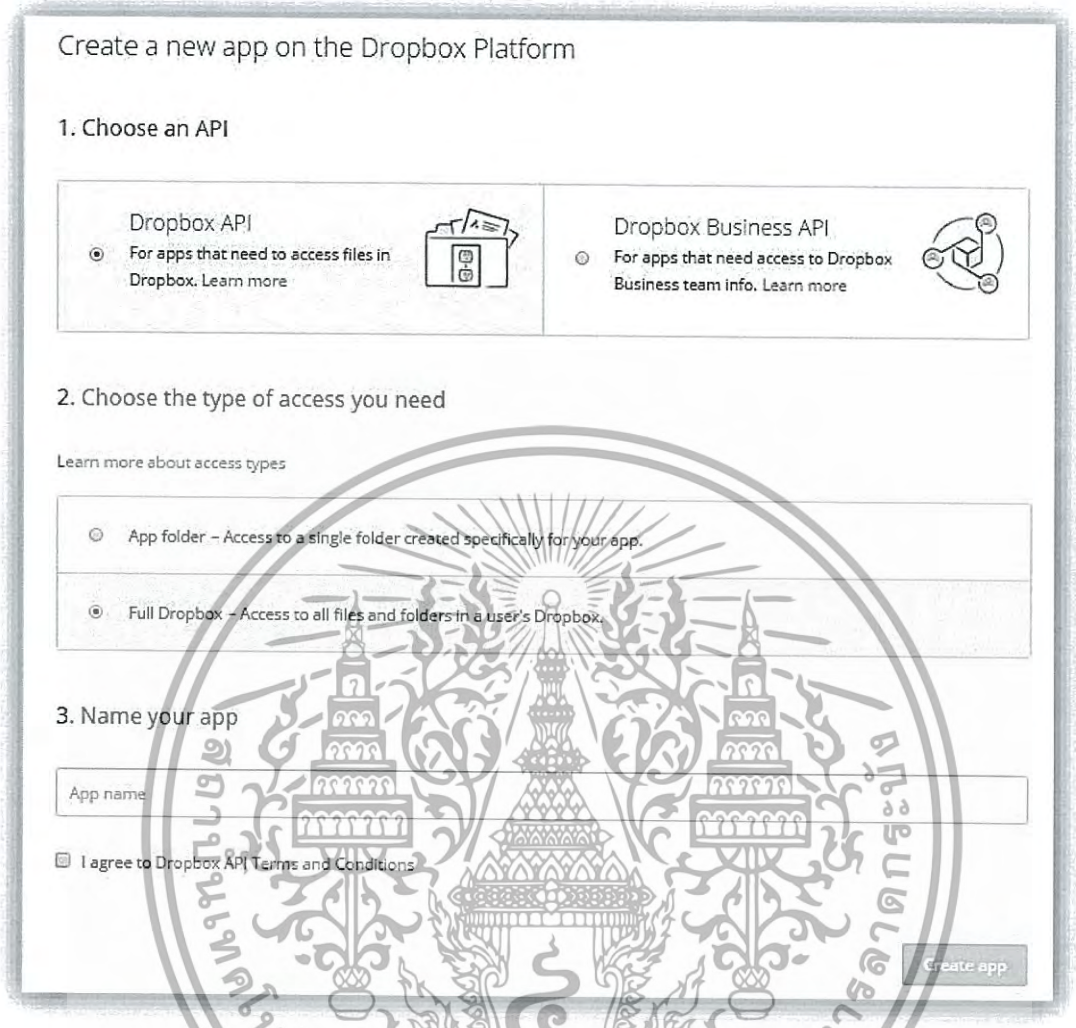


รูปที่ ก.2 หน้า Developers ของ Dropbox

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 กรอกรายละเอียดที่ต้องการแล้วกด Create app



รูปที่ ก.3 หน้า Create New Application ของ Dropbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

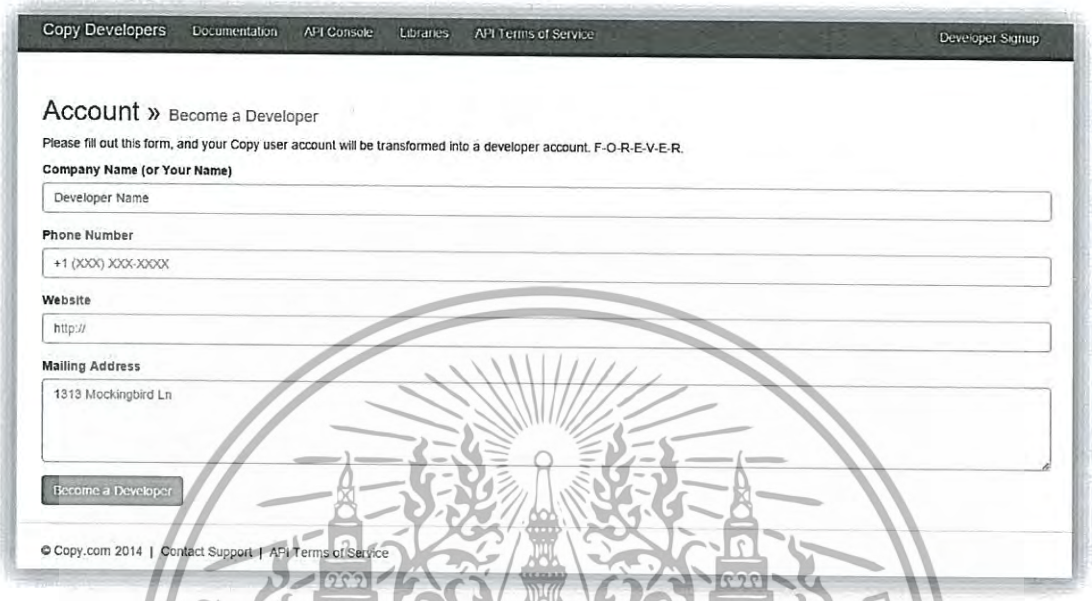
1.4 กรอก Redirect URIs และตั้งค่าส่วนอื่น ๆ ตามต้องการ สามารถนำ App Key และ App secret ไปใช้งานได้เลย

รูปที่ ก.4 หน้าต่างรายละเอียดของ Application ที่สร้างบน Dropbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Copy

2.1 เข้าสู่ระบบที่เว็บ https://www.copy.com และลงทะเบียนเป็นผู้พัฒนาระบบที่ https://developers.copy.com/account/signup



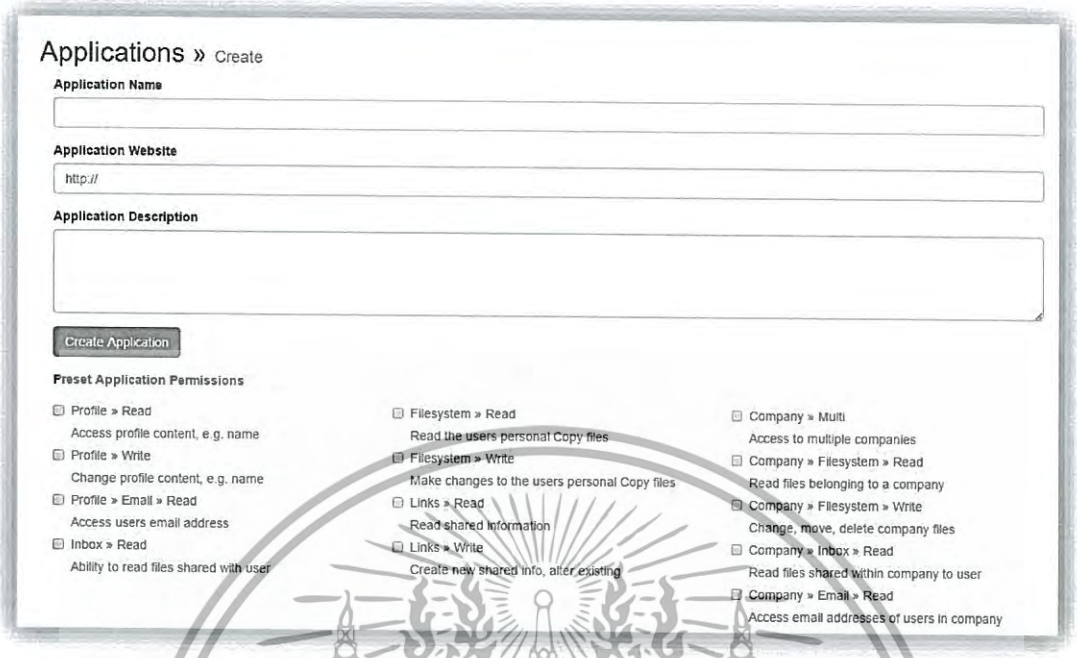
รูปที่ ก.5 หน้าต่างลงทะเบียนเป็น Developers ของ Copy

2.2 เข้าไปที่หน้า Applications เลือก Create One (กรณีที่ยังไม่เคยสร้างแอพพลิเคชั่นมาก่อน)

รูปที่ ก.6 หน้าต่างการจัดการ Application ของ Copy

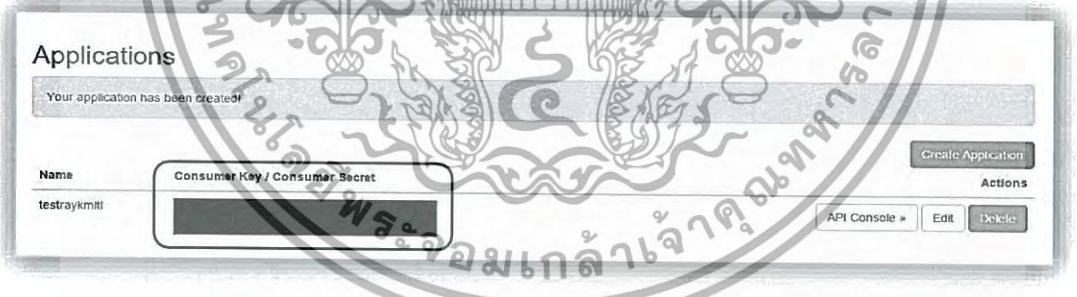
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 กรอกรายละเอียดและเลือก Permission ที่ต้องการ



รูปที่ ก.7 หน้าต่างสร้าง Application ใหม่ของ Copy

2.4 เมื่อสร้างแอพพลิเคชั่นเสร็จเรียบร้อย จะได้ Consumer Key และ Consumer Secret สามารถนำไปใช้งานได้เลย



รูปที่ ก.8 หน้าต่างรายละเอียดของ Application ของ Copy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. OneDrive

3.1 เข้าสู่ระบบที่เว็บไซต์ของ OneDrive ที่ https://login.live.com/login.srf และไปที่หน้า https://account.live.com/developers/applications เลือกสร้างแอพพลิเคชั่น

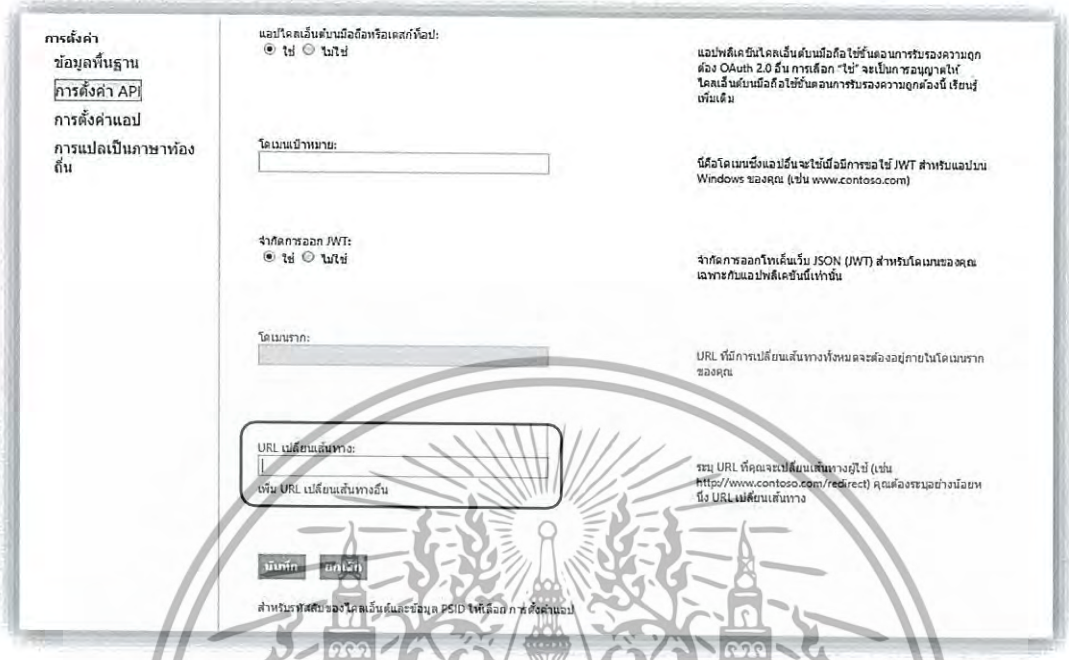
รูปที่ ก.9 หน้าต่างการจัดการ Application ของ OneDrive

3.2 ตั้งชื่อแอพพลิเคชั่นและเลือกภาษา จากนั้นกด ฉันยอมรับ

รูปที่ ก.10 หน้าต่างสร้าง Application ใหม่บน OneDrive

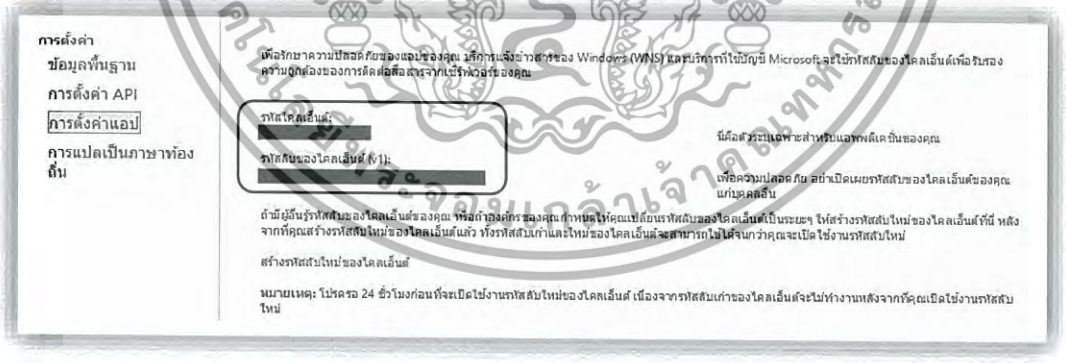
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 เมื่อสร้างแอพพลิเคชั่นเสร็จเรียบร้อย ให้ไปที่ การตั้งค่า API กรอก URL เปลี่ยนเส้นทาง (Redirect URL) แล้วบันทึก



รูปที่ ก.11 หน้าต่างตั้งค่า Redirect URL ภายใน Application ของ OneDrive

3.4 ไปที่การตั้งค่าแอป สามารถนำ รหัสไคลเอ็นต์ และ รหัสลับของไคลเอ็นต์ ไปใช้งานได้เลย



รูปที่ ก.12 หน้าต่างแสดงรายละเอียด Application ของ OneDrive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Box

4.1 เข้าสู่ระบบที่เว็บไซต์ของ Box ที่ https://app.box.com แล้วไปที่หน้า นักพัฒนาที่ https://app.box.com/developers/services เลือก Create a Box Application

รูปที่ ก.13 หน้าต่างเมนู Developers ของ Box

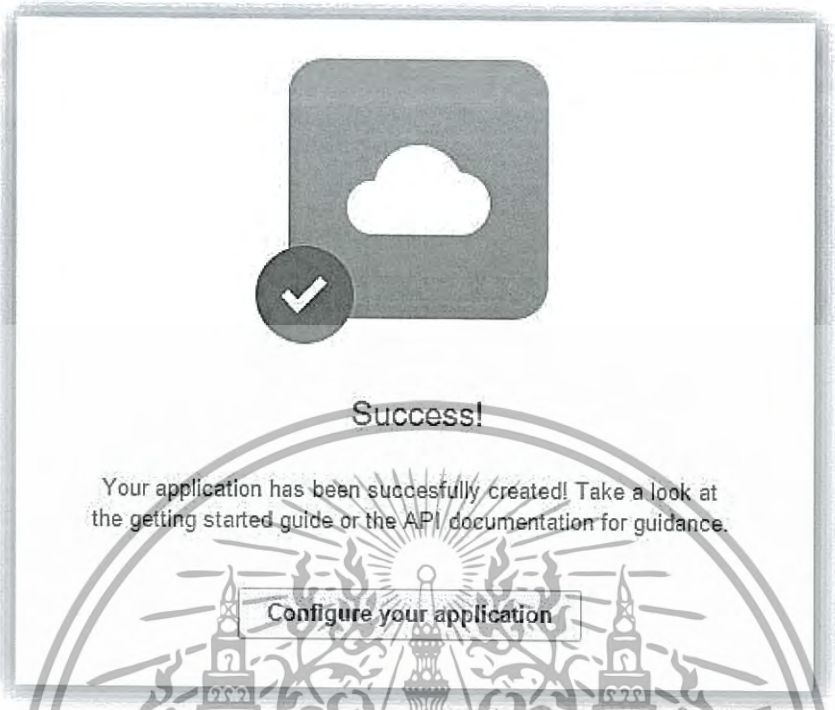
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ตั้งชื่อแอพพลิเคชั่นแล้วเลือก Box Content กด Create Application

รูปที่ ก.14 หน้าต่างการสร้าง Application ใหม่ของ Box

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 เมื่อสร้างแอพพลิเคชั่นเสร็จเรียบร้อย กด Configure your application

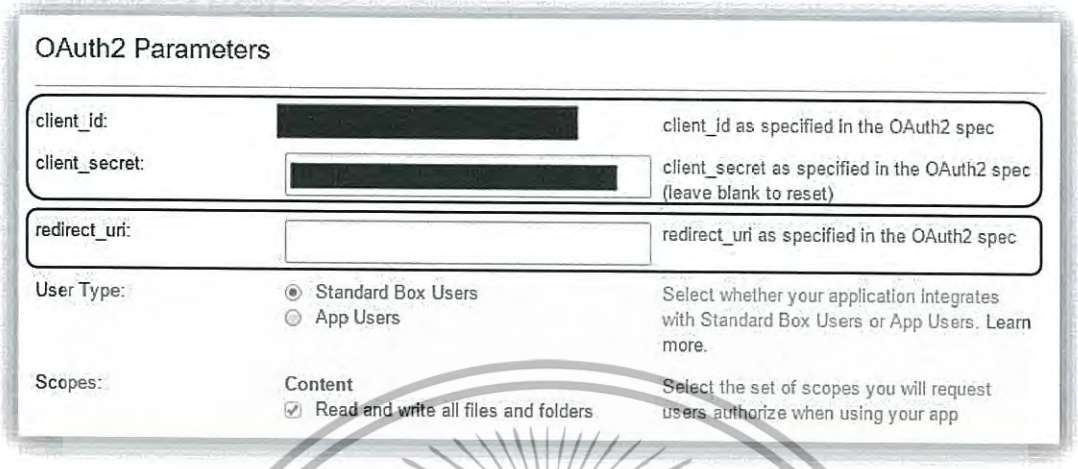


รูปที่ ก.15 หน้าต่างตั้งค่า Application ของ Box

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ในส่วนของ Create a Box Application ให้กรอก Redirect URL และ Save Application ที่ด้านล่างสุด สามารถนำ client_id และ client_secret ไปใช้งานได้เลย



รูปที่ ก.16 หน้าต่างแสดงข้อมูลและตั้งค่าเกี่ยวกับ Application ของ Box

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Google Drive

5.1 เข้าสู่ระบบที่เว็บไซต์ https://console.developers.google.com แล้วทำการสร้าง Project

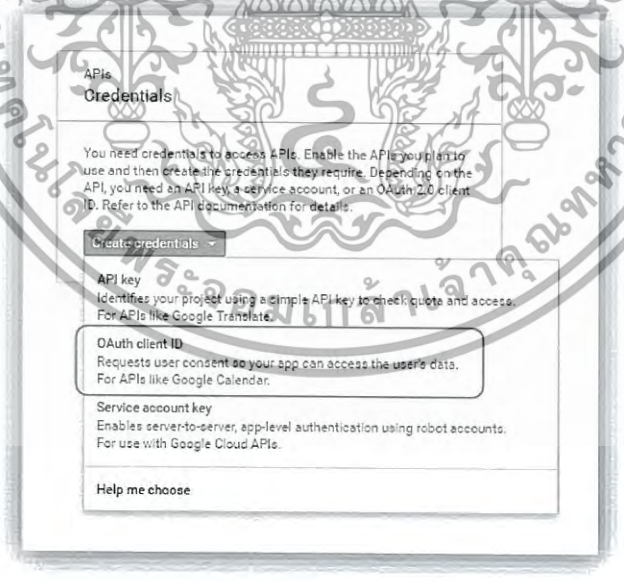
รูปที่ ก.17 หน้าต่างสร้าง Project สำหรับ Google

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 เมื่อสร้าง Project เรียบร้อยแล้ว ให้ไปที่แถบ Credentials และตั้งค่าที่ OAuth consent screen เสร็จแล้วทำการบันทึก

รูปที่ ก.18 หน้าต่างแสดงข้อมูลและตั้งค่าเกี่ยวกับ Project ของ Google

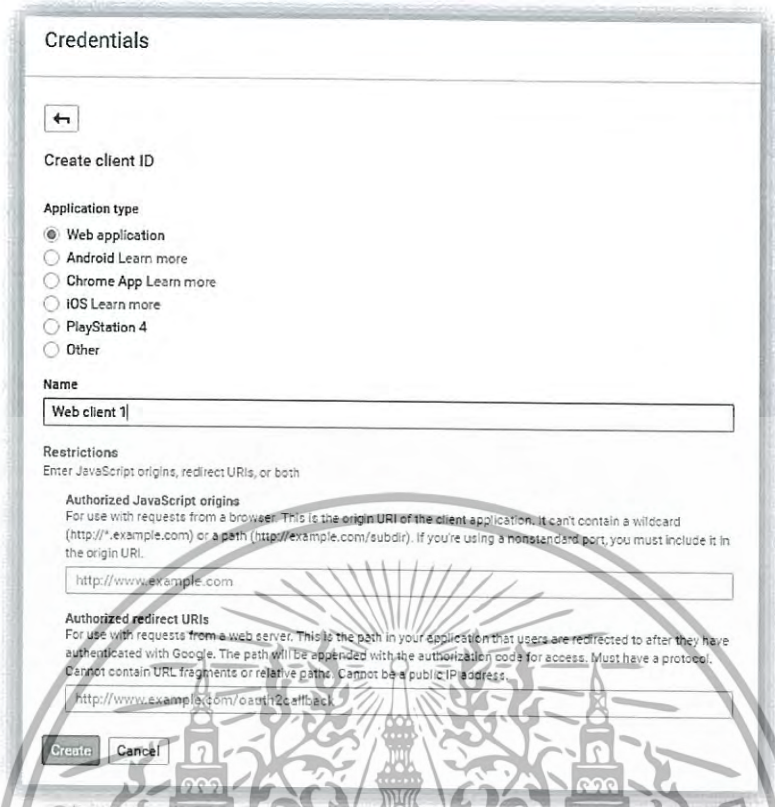
5.3 จากนั้นไปที่ Credentials เพื่อสร้าง OAuth client ID สำหรับรับ client ID และ cilent secret เพื่อใช้ในการขอความยินยอมการเข้าถึงข้อมูลของผู้ที่ใช้แอพพลิเคชั่นของเรา



รูปที่ ก.19 หน้าต่างสำหรับเลือกประเภท Credentials ของ Google

5.4 เรียกประเภทของแอพพลิเคชั่น แล้วตั้งค่า จากนั้นกด Create

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



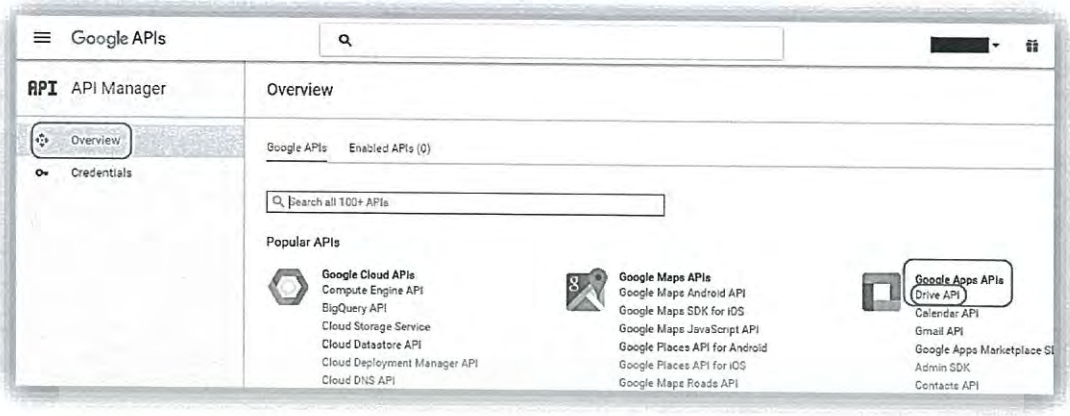
รูปที่ ก.20 หน้าต่างสำหรับสร้าง Credentials ประเภท OAuth client ID ของ Google

5.5 หลังจากสร้าง OAuth client ID เรียบร้อยแล้วจะ ได้รับ client ID และ cilent secret

รูปที่ ก.21 หน้าต่างแสดง client ID และ cilent secret

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.6 ไปที่แท็บ Overview ในหมวด Google Apps APIs เลือก Drive API



รูปที่ ก.22 หน้าต่างแสดง Overview แท็บ Google APIs

5.7 เมื่อเปิดการใช้งาน Google Drive API ก็สามารถเชื่อมต่อได้เลย

รูปที่ ก.23 หน้าต่างแสดงรายละเอียดของ Google Dive API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

คู่มือการติดตั้งเว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆ

1. Checklist

- Web Server ที่รองรับ PHP 5.5
- MySQL Relational Database
- Composer
- Laravel
- Oauth.dll (optional for Windows OS)
- Project Source Code

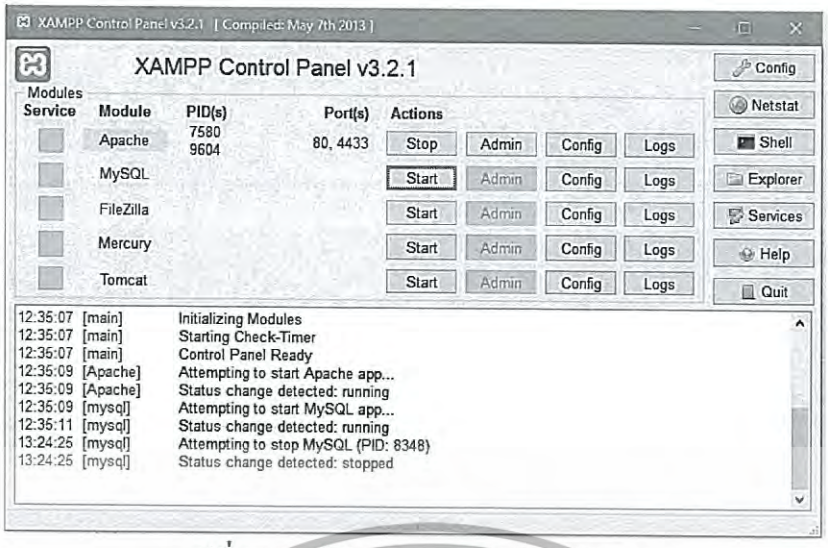
2. Installation

2.1 Web Server

ในกรณีของผู้พัฒนาได้เริ่มพัฒนาจากเครื่อง Localhost จึงเลือกใช้ XAMPP for Windows 1.8.3 ที่รองรับ PHP 5.5.15 และทำหน้าที่เป็นเว็บเซิร์ฟเวอร์จำลองบนเครื่องที่พัฒนา แต่ถ้าต้องการจะทำการ Deploy ลงบนเว็บเซิร์ฟเวอร์อื่นก็ทำได้ ขอเพียงแค่รองรับ PHP 5.5 ขึ้นไปและตั้งค่าให้เหมือนกับที่กำหนดไว้ ซึ่งขั้นตอนในการติดตั้งและตั้งค่า XAMPP มีดังนี้

1. ดาวน์โหลดตัวติดตั้งของ XAMPP for Windows หรือ XAMPP for Linux จาก https://www.apachefriends.org/download.html
2. ทำการติดตั้ง
3. หลังจากที่ทำการติดตั้งเรียบร้อยให้เปิด XAMPP Control Panel ขึ้นมาและ Start Module ของ Apache ดังรูปที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.1 หน้าต่าง XAMPP Control Panel

4. ทำการทดสอบเว็บเซิร์ฟเวอร์ว่าสามารถใช้งานได้หรือไม่ ถ้าเป็นเว็บเซิร์ฟเวอร์จำลองสามารถทำงานได้หรือไม่นั้น สามารถทดสอบได้โดยเข้า URL http://localhost/xampp/ บน Browser ถ้าขึ้นหน้าเว็บแสดงว่าสามารถใช้งานได้

2.2 MySQL

ฐานข้อมูลที่ระบบใช้นั้น เลือกใช้เป็น Relational Database ที่จะสามารถแสดงความสัมพันธ์ของข้อมูลได้ดี วิธีการในการติดตั้งนั้นบางครั้งก็ไม่จำเป็น เพราะเว็บเซิร์ฟเวอร์จำลองบางตัวนั้นจะทำการติดตั้ง MySQL ให้โดยอัตโนมัติ แต่ถ้าไม่มีการลง MySQL สามารถลงได้เอง ส่วนมาก Linux จะลง Service ให้อยู่แล้ว แต่ถ้าเป็น Windows OS สามารถดาวน์โหลดตัวติดตั้งได้ที่ URL http://dev.mysql.com/downloads/installer/ เพื่อมาติดตั้งเองได้

2.3 Composer

- Windows OS นั้นสามารถดาวน์โหลดตัวติดตั้งได้จาก https://getcomposer.org/download/ และหลังจากติดตั้งเสร็จ จะสามารถเรียกใช้งานคำสั่ง composer จาก Command Line ได้

- Linux/Unix นั้นจะมีการลงที่ซับซ้อนกว่า โดยต้องโหลดไฟล์ install จาก https://getcomposer.org/doc/00-intro.md#installation-linux-unix-osx ซึ่งเป็นไฟล์ .php ธรรมดาไฟล์นึง ทำหน้าที่ในการเช็คการตั้งค่าบางอย่างของ PHP และทำการโหลด composer.phar ให้โดยอัตโนมัติ และใช้คำสั่งที่สองทำการตั้งค่าให้สามารถใช้คำสั่ง composer แบบ Global ได้ ตามรูปที่ 2 และ รูปที่ 3 ซึ่งสามารถศึกษาเพิ่มเติมได้จาก https://getcomposer.org/doc/00-intro.md#installation-linux-unix-osx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
php composer-setup.php --install-dir=bin --filename=composer
```

รูปที่ ข.2 ชุดคำสั่งเพื่อติดตั้ง Composer

```
mv composer.phar /usr/local/bin/composer
```

รูปที่ ข.3 ชุดคำสั่งเพื่อติดตั้ง Composer

2.4 Laravel

หลังจากที่ได้ติดตั้งตัวจัดการไลบรารี่ต่างๆ (Composer) แล้ว ก็ต้องทำการติดตั้ง Laravel Framework โดยต้องใช้ extension ต่างๆ ดังนี้

- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension

ซึ่งสามารถจัดการได้ภายในไฟล์ php.ini และหลังจากที่ตรวจสอบ extension ดังกล่าวครบถ้วนแล้วก็จะทำการติดตั้ง Laravel Framework โดยพิมพ์ชุดคำสั่งตามรูปภาพถัดไป

```
composer global require "laravel/installer"
```

รูปที่ ข.4 ชุดคำสั่งเพื่อติดตั้ง Laravel Framework

เมื่อทำการติดตั้งเรียบร้อยแล้ว สามารถทดสอบได้โดยสร้าง Project ด้วยชุดคำสั่งใด คำสั่งหนึ่ง ดังนี้

```
laravel new blog
```

รูปที่ ข.5 ชุดคำสั่งเพื่อสร้าง Project

```
composer create-project --prefer-dist laravel/laravel blog
```

รูปที่ ข.6 ชุดคำสั่งเพื่อสร้าง Project ผ่าน Composer

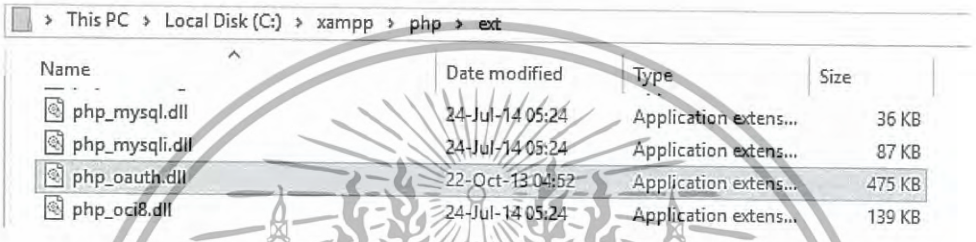
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 Oauth.dll

ในบางไลบรารี่ที่ระบบได้นำมาใช้นั้นบางครั้งต้องใช้ Oauth ซึ่งเป็น extension เสริม เราได้พบระหว่างการพัฒนาว่า เว็บเซิร์ฟเวอร์จำลองที่ติดตั้งบน Windows นั้นไม่มี extension เสริมดังกล่าวจึงต้องมีการติดตั้งเอง โดยวิธีการติดตั้งคือ

1. ดาวน์โหลด extension เสริม https://pecl.php.net/package/oauth โดยเลือกเวอร์ชั่น 1.2.3 ซึ่งบน Windows นั้นจะใช้ไฟล์ .dll

2. นำไฟล์ .dll ที่ดาวน์โหลดมาไปวางไว้ในโฟลเดอร์ที่ติดตั้งเว็บเซิร์ฟเวอร์ โดยมี path: {webserv}\php\ext โดยตั้งชื่อไฟล์ว่า php_oauth.dll ดังภาพ



รูปที่ ข.7 ภาพการวางไฟล์ php_oauth.dll ไว้ใน path ที่ระบุ

3. เปิดการใช้งาน extension ที่ได้ใส่เข้าไป โดยแก้ไขในไฟล์ php.ini เพิ่มคำสั่ง `extension=php_oauth.dll` และไม่มีเครื่องหมาย colon ไว้ด้านหน้า ซึ่งจะเพิ่มในส่วนของ Dynamic Extensions ของไฟล์

4. ทดสอบโดยการรัน phpinfo() เพื่อเช็คว่าเว็บเซิร์ฟเวอร์ได้รองรับ extension Oauth แล้ว

OAuth

OAuth support	enabled
PLAINTEXT support	enabled
RSA-SHA1 support	not supported
HMAC-SHA1 support	enabled
Request engine support	php_streams, curl
source version	$Id: oauth.c 325799 2012-05-24 21:07:51Z jawed $
version	1.2.3

รูปที่ ข.8 ภาพไลบรารี่ OAuth ที่ปรากฏใน phpinfo()

2.8 Project Source Code

ขั้นตอนในการติดตั้งเว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆมีดังนี้

1. ดาวน์โหลด Source Code จาก URL: https://github.com/noxfate/gathercloud

2. นำ Source Code มาวางไว้ในเว็บเซิร์ฟเวอร์

3. พิมพ์คำสั่ง `composer install` ภายใน Command Line เพื่อติดตั้ง Library ที่ระบบต้องใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. แก้ไขไฟล์ .env โดยแก้ไขตรงส่วนที่ระบุการเชื่อมต่อกับฐานข้อมูลให้เหมาะสมตามที่ต้องการ

```
DB_HOST=localhost
DB_DATABASE=homestead
DB_USERNAME=homestead
DB_PASSWORD=secret
```

รูปที่ ข.9 ส่วนเชื่อมต่อกับฐานข้อมูลภายในไฟล์ .env

5. หลังจากที่แก้ไขไฟล์ .env ได้ถูกต้อง พิมพ์คำสั่ง `php artisan migrate` ภายใน Command Line เพื่อสร้างสกีม่าของฐานข้อมูลในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ	นายธีรพรรณ วิเชียรศักดา
วัน เดือน ปี เกิด	4 ธันวาคม 2537
ที่อยู่	เลขที่ 19/93 หมู่บ้าน เดอะคอนเน็ค ถนน เพิ่มสิน-สายไหม แขวงคลองถนน เขต สายไหม จังหวัดกรุงเทพมหานครฯ 10220
ประวัติการศึกษา	วิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ชื่อ	นางสาวสาวิตรี บุญสอน
วัน เดือน ปี เกิด	17 มิถุนายน 2536
ที่อยู่	เลขที่ 75/61 หมู่ 6 ซอยสุขสวัสดิ์ 76 ถนนสุขสวัสดิ์ ตำบลบางจาก อำเภอ พระประประแดง จังหวัดสมุทรปราการ 10130
ประวัติการศึกษา	วิทยาศาตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆ

ธีรพรรณ วิเชียรศักดา และ สาวิตรี บุญสอน

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ

Emails: teerapan.vi@gmail.com, aimd.sawitree@gmail.com

บทคัดย่อ

บทความฉบับนี้ได้นำเสนอข้อมูลที่เกี่ยวข้องกับเว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆโดยมีจุดประสงค์ในการพัฒนาเพื่อช่วยอำนวยความสะดวกสบาย ลดความวุ่นวายและความลำบากในการเข้าถึงและจัดการข้อมูลของผู้ใช้ที่เลือกใช้บริการแหล่งเก็บข้อมูลมากกว่าหนึ่งบริการ อีกทั้งยังช่วยลดความซ้ำซ้อนของข้อมูลอีกด้วย เริ่มแรกนั้นระบบจะรองรับผู้ให้บริการ 5 บริการ ได้แก่ Google Drive, OneDrive, Box, Dropbox และ Copy โดยการทำงานของระบบนั้นจะเป็นทำหน้าที่เป็นศูนย์กลางการทำงานของ Application Program Interface (API) ของแต่ละผู้ให้บริการและทำการประมวลผลข้อมูลที่ได้มาจากการเรียกใช้ฟังก์ชั่นใน API ของผู้ให้บริการที่ต่างกันเพื่อนำไปใช้ต่อในเว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆ

คำสำคัญ – แหล่งเก็บข้อมูลบนกลุ่มเมฆ; Application Program Interface(API); เว็บแอพพลิเคชั่น;

1. บทนำ

ปัจจุบันข้อมูลบนเครือข่ายนั้นได้มีบทบาทสำคัญมากยิ่งขึ้นในชีวิตประจำวันของผู้คน จึงส่งผลให้เทคโนโลยีที่เกี่ยวข้องก็มีความสำคัญไม่แพ้กัน ในช่วงเวลาที่ผ่านมาได้เกิดเทคโนโลยีการเก็บข้อมูลออนไลน์ หรือการเก็บข้อมูลบนกลุ่มเมฆ หมายถึง การเก็บข้อมูลโดยฝากไว้กับผู้ที่ให้บริการ ซึ่งมีข้อดีมากมายได้แก่ ทำให้ผู้ใช้สามารถเข้าถึงข้อมูลได้จากทุกที่ ทุกเวลาและจากทุกอุปกรณ์ผ่านอินเตอร์เน็ต ป้องกันการสูญหายของข้อมูล และช่วยสนับสนุนการแชร์ข้อมูลกับผู้อื่นได้อย่างสะดวกสบาย ความนิยมในการใช้แหล่งเก็บข้อมูลบนกลุ่มเมฆเริ่มมีการแพร่หลาย ทั้งผู้ใช้งานและผู้ให้บริการก็มีเพิ่มมากขึ้น จนเกิดความหลากหลายทางการใช้งาน ตัวผู้ใช้เองจึงพบกับปัญหาเรื่องการกระจายกันของพื้นที่ที่ใช้จัดเก็บข้อมูล เกิดปัญหาความซ้ำซ้อนและความยุ่งยากในการจัดการข้อมูลทั้งหมดจากแหล่งเก็บข้อมูลบนกลุ่มเมฆที่แตกต่างกัน เช่น ไม่สามารถหาข้อมูลได้ตามที่ต้องการเพราะลืมว่าเก็บไว้ที่แหล่งเก็บข้อมูลใด เป็นต้น จึงเป็นที่มาของเว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆ

แนวคิดหลักที่ใช้การพัฒนาและออกแบบ คือ การออกแบบระบบให้ผู้ใช้สามารถใช้งานฟังก์ชั่นต่างๆ ของผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆหลายๆ ผู้ให้บริการไว้ในที่เดียวและสามารถทำงานร่วมกันได้โดยไม่แบ่งแยกโดยวิธีการที่ผู้พัฒนาเลือกใช้คือการใช้ API หรือ Library ต่างๆ ของผู้ให้บริการหรือผู้พัฒนาคนอื่นๆ โดยนำมาใช้งานรวมกันภายใต้ระบบเดียวและออกแบบให้มีการทำงานร่วมกันได้ ในปัจจุบันรองรับผู้ให้บริการเหล่านี้ได้แก่ Google Drive, OneDrive, Box, Dropbox และ Copy ซึ่งในการรวมรวมข้อมูลต่างๆ ที่ได้มาจากผู้ให้บริการนั้นจะมีรูปแบบที่ไม่เหมือนกันอีกทั้งยังมีรายละเอียดในการใช้งานที่ต่างกัน ระบบจึงต้องรองรับการทำงานจาก API หรือ Library ตัวอื่นๆ อีกด้วย ซึ่งแน่นอนว่าต้องมีการประมวลผลข้อมูลที่ได้มาเพื่อทำให้ข้อมูลที่มาจากหลายผู้ให้บริการอยู่ในรูปแบบที่เป็นมาตรฐานของระบบ จึงจะสามารถนำไปใช้งานต่อและแสดงผลออกมาให้ผู้ใช้เห็นได้

2. ทบทวนวรรณกรรมที่เกี่ยวข้อง

ส่วนนี้จะกล่าวถึงทฤษฎีบทต่างๆ ที่เกี่ยวข้องกับการพัฒนาระบบ

2.1. การประมวลผลบนกลุ่มเมฆ

เทคโนโลยีในการประมวลผลบนกลุ่มเมฆ (Cloud Computing) คือเทคโนโลยีในการประมวลผลโดยมีหลักการในการแบ่งทรัพยากรจากคอมพิวเตอร์หลายๆ เครื่องมาช่วยในการประมวลผลโดยเชื่อมต่อกันด้วยเครือข่ายคอมพิวเตอร์ และการเกิดเทคโนโลยีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมวลผลบรกลุ่มเมฆส่งผลให้เกิดประโยชน์มากมาย ไม่ว่าจะเป็นด้านธุรกิจ หรือด้านการสร้างสรรค์นวัตกรรมใหม่ๆ ตัวอย่างเช่น ในการพัฒนาแอพพลิเคชั่นเพื่อมาใช้ในการดำเนินธุรกิจจะเสียค่าใช้จ่ายเป็นอย่างมาก แต่ด้วยเทคโนโลยีการประมวลผลบนกลุ่มเมฆทำให้ค่าใช้จ่ายลดน้อยลงและมีความยืดหยุ่นมาก ไม่ว่าจะเป็นด้านการพัฒนา การติดตั้ง การอัพเดท การดูแล และอื่นๆ อีกมากมาย โดยบริการต่างๆ เหล่านี้สามารถแบ่งออกเป็น 3 จำพวกใหญ่ๆ ได้แก่

2.1.1. Infrastructure as a Service (IaaS)

รูปแบบของ IaaS คือ มีบุคคลที่สามให้บริการทรัพยากรต่างๆ ในโครงสร้างพื้นฐานของระบบทำให้ง่ายต่อการจัดการและปรับแต่งให้มีความเหมาะสมกับงาน รวมถึงง่ายต่อการดูแลและสำรองข้อมูลที่เก็บไว้ได้อีกด้วย โดยผู้ให้บริการที่มีชื่อเสียงในปัจจุบัน ได้แก่ Amazon Web Services (AWS), Google Compute Engine, Windows Azure

2.1.2. Platform as a Service (PaaS)

การให้บริการแบบ PaaS คือ การให้บริการ Platform ให้กับลูกค้าเพื่อพัฒนา จัดการหรือรันแอพพลิเคชั่นบน Platform ที่จัดเตรียมไว้ให้ ทำให้ง่ายต่อการเตรียมสภาพแวดล้อมและการจัดการหรือแม้กระทั่งการอัพเกรด และยังสนับสนุนการทำงานร่วมกันของผู้ใช้อีกด้วย ตัวอย่างที่เห็นภาพได้ชัดเจนคือ Salesforce.com ที่ให้บริการ Enterprise Customer Relationship Management (CRM) Platform และอื่นๆ เช่น Appear IQ, Mendix, Google App Engine, AWS Elastic Beanstalk, Heroku เป็นต้น

2.1.3. Software as a Service (SaaS)

การบริการในรูปแบบนี้เริ่มแพร่หลายในปัจจุบัน โดยจะทำงานร่วมกับ Web Services และ Services-Oriented Architecture (SOA) และบริการแบบ SaaS นั้นจะเป็นการกระจายซอฟต์แวร์ไปยังลูกค้าทำให้ลูกค้าสามารถเข้าถึงซอฟต์แวร์ผ่านอินเตอร์เน็ตได้ จึงทำให้ผู้ใช้เข้าถึงซอฟต์แวร์ที่เป็นเวอร์ชั่นเดียวกันอีกด้วย

2.2. แหล่งเก็บข้อมูลบนกลุ่มเมฆ

การเก็บข้อมูลในปัจจุบันจะมีการเก็บข้อมูลไว้ในแหล่งเก็บข้อมูลภายในตัวเครื่อง เช่น ฮาร์ดดิสก์ โดยมีข้อดีคือเข้าถึงข้อมูลที่เก็บไว้ได้ง่ายและรวดเร็ว แต่แหล่งเก็บข้อมูลเหล่านั้นก็สามารถชำรุดและสูญหายได้ง่าย ซึ่งหลังจากที่ได้รู้จักกับเทคโนโลยีการประมวลผลบนกลุ่มเมฆจึงได้เกิดการบริการแหล่งเก็บข้อมูลบนกลุ่มเมฆ (Cloud Storage) ที่จะเป็นโมเดลในการเก็บรักษาจัดการและสำรองข้อมูลผ่านเครือข่าย ซึ่งส่วนมากจะทำงานโดยให้ผู้ใช้เข้าถึงได้ผ่านอินเตอร์เน็ต ข้อดีของแหล่งเก็บข้อมูลบนกลุ่มเมฆคือ สามารถเข้าถึงข้อมูลได้ง่ายจากทุกที่ ทุกเวลา และจากทุกอุปกรณ์ แต่ทั้งนี้ก็มีข้อจำกัดคือจำเป็นต้องใช้อินเตอร์เน็ตในการเข้าถึง

2.3. Application Program Interface (API)

ในการพัฒนาโปรแกรมหรือเว็บไซต์อาจจะต้องมีการติดต่อกันระหว่าง Application ซึ่ง API จะทำหน้าที่ในการระบุขั้นตอนและวิธีในการติดต่อกันระหว่างทั้งสอง Application ทำให้มีการบูรณาการและทำงานร่วมกัน ทั้งนี้ถ้ามีการพัฒนา API ที่ดี จะทำให้การนำไปประยุกต์ใช้ง่ายและไม่ซับซ้อน

2.4. Representational State Transfer

REST คือรูปแบบของสถาปัตยกรรมอย่างหนึ่งที่ออกแบบมาเพื่อแอพพลิเคชั่นที่ต้องการทำงานบนเครือข่ายคอมพิวเตอร์ คุณลักษณะสำคัญได้แก่ ใช้ HTTP Protocol, ทำงานแบบ Stateless, และสามารถเก็บข้อมูลสำรองได้ Cacheable ซึ่งสถาปัตยกรรมแบบ REST จะทำงานง่าย ไม่ซับซ้อนและตรงไปตรงมา เป็นอิสระต่อ Platform และภาษาที่เลือกใช้พัฒนา

2.5. cURL

cURL หรือ "Client URL Request Library" เป็น Library หนึ่งที่ใช้ในการสร้าง HTTP Request และ อ่าน HTTP Response ที่ได้กลับมาแต่ไม่ได้นำไปประมวลผลแสดงเหมือนกับ Web Browser ซึ่งมักจะถูกนำไปใช้ในการเขียนประมวลผลฝั่ง Back-end

2.6. OAuth

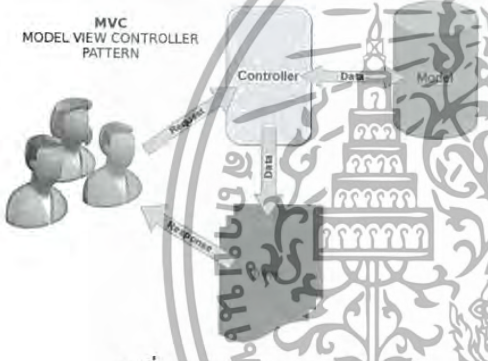
OAuth คือ มาตรฐานสำหรับการมอบสิทธิให้กับ Application ที่เป็นบุคคลที่สามสามารถเข้าถึงข้อมูลได้อย่างปลอดภัยและยังสามารถกระทำการแทนผู้ที่อนุญาต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิทธิให้กับ Application ได้ ซึ่งจะมีเครื่องหมายในการเข้าถึงได้เพื่อใช้ในการยืนยันตัวตน (Access Token) โดยสามารถเรียกใช้งานผ่าน HTTP/HTTPS ได้

2.7. MVC Architectural Pattern

สถาปัตยกรรม MVC หรือ Model-View-Controller คือรูปแบบในการออกแบบและพัฒนาซอฟต์แวร์แบบหนึ่ง โดยจะแบ่งการทำงานออกเป็น 3 ส่วน ส่วนประกอบที่สำคัญคือ Model ที่คอยจำลองในส่วนที่ทำหน้าที่จัดการข้อมูลขึ้นมา และมีการกระทำกับข้อมูลที่ชัดเจน ซึ่งจะคอยมี View ที่ใช้ในการติดต่อกับผู้ใช้ หรือ Controller ที่ทำหน้าที่ประสานงานระหว่าง View และ Model คอยช่วย รูปแบบ MVC จะทำให้การพัฒนาแยกออกเป็นสัดส่วนชัดเจนทำให้ง่ายต่อการดูแล พัฒนาต่อและแก้ไข



รูปที่ 1 ภาพจำลองรูปแบบ MVC

2.8. Nested Set [1]

Nested Set คือเทคนิคในการเก็บข้อมูลประเภท Trees หรือ Hierarchies ลงใน relational database โดยมีหลักการเก็บตัวแปรต่างๆ ที่บ่งบอกถึงความสัมพันธ์ของแต่ละ Node ซึ่งเป็นวิธีที่มีประสิทธิภาพในการ Query ที่เร็ว แต่ทำให้การเพิ่มหรือแก้ไขข้อมูลทำได้ช้า ซึ่งผู้พัฒนาได้นำเทคนิคดังกล่าวมาใช้ในการเก็บข้อมูลเรื่องจากลักษณะของข้อมูลที่ต้องการจะเก็บนั้นมีลักษณะและรูปแบบที่ตรงกับ Nested Set ส่งผลให้มีการดูแลข้อมูลได้อย่างสะดวกสบายและสามารถรักษาความถูกต้องของข้อมูลได้ด้วย

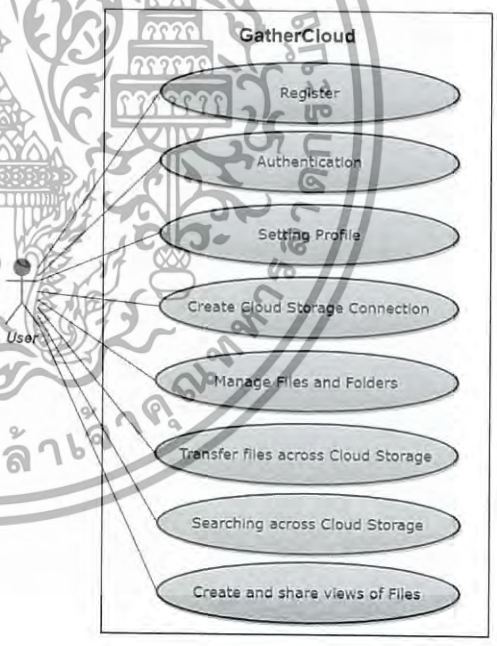
3. การวิเคราะห์และออกแบบระบบ

ส่วนนี้จะพูดถึงการวิเคราะห์และออกแบบระบบเว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆโดยได้ทำการวิเคราะห์จากปัญหาจนออกมาเป็นความต้องการของระบบ

3.1. ความต้องการของระบบ

จากการศึกษาปัญหาได้พบปัญหาในเรื่องของการทำงานที่ซ้ำซ้อนและความยุ่งยากที่เกิดขึ้นเมื่อต้องการจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆและได้พบกับระบบงานที่มีอยู่แล้ว คือ Multcloud [2] ซึ่งหลังจากที่ได้วิเคราะห์ระบบงานเดิมก็ได้พบว่ายังไม่สามารถแก้ปัญหาได้ตรงจุด จึงเลือกที่จะนำระบบงานเดิมมาวิเคราะห์และพัฒนาเพิ่มเติมจากเดิม โดยมีแนวคิดหลักในการพัฒนาคือ การมองแหล่งเก็บข้อมูลหลายๆ บริการเป็นเพียงบริการเดียวและสามารถทำงานด้วยกันโดยใช้รูปแบบในการจัดการที่เป็นหนึ่งเดียวเพื่อให้ผู้ใช้สามารถใช้งานได้สะดวกที่สุด จึงได้ความต้องการที่เป็นหน้าที่หลักของระบบมาดังนี้

3.1.1. Use Case Diagram



รูปที่ 2 แผนภาพยูสเคสของระบบ

- ผู้ใช้สามารถสมัครสมาชิกกับเว็บไซต์ได้
- ผู้ใช้สามารถแก้ไขข้อมูลส่วนตัวที่ลงทะเบียนได้
- ผู้ใช้สามารถเลือกการเชื่อมต่อกับแหล่งเก็บข้อมูลบนกลุ่มเมฆได้มากกว่าหนึ่งการเชื่อมต่อ โดยจะเชื่อมต่อเพียงครั้งเดียวและไม่จำเป็นต้องเชื่อมต่ออีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เว็บไซต์จะรองรับการเชื่อมต่อกับแหล่งเก็บข้อมูลบนกลุ่มเมฆดังนี้ Google Drive, OneDrive, Box, Dropbox และ Copy
- ผู้ใช้สามารถอัพโหลด ดาวน์โหลด ลบ แก้ไข และแชร์ไฟล์หรือข้อมูลจากแหล่งเก็บข้อมูลบนกลุ่มเมฆได้
- ผู้ใช้สามารถค้นหาไฟล์หรือข้อมูลจากแหล่งเก็บข้อมูลบนกลุ่มเมฆได้
- เว็บไซต์จะสามารถแสดงรายละเอียดเกี่ยวกับแหล่งเก็บข้อมูลบนกลุ่มเมฆ และรายละเอียดเกี่ยวกับไฟล์ที่ถูกเก็บไว้ได้
- ผู้ใช้สามารถย้ายไฟล์ข้ามแหล่ง เก็บข้อมูลบนกลุ่มเมฆได้
- ผู้ใช้สามารถแชร์สิทธิในการมองเห็นหรือเข้าถึงไฟล์และข้อมูลตามที่ต้องการได้

และในส่วนของความต้องการที่ไม่ใช่หน้าที่หลักของระบบนั้น เว็บไซต์จะต้องมีความปลอดภัยและรักษาความเป็นส่วนตัวที่เกี่ยวข้องกับข้อมูลของผู้ใช้ได้

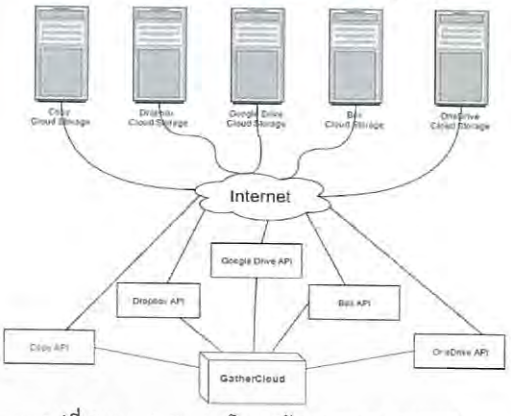
3.2. การวิเคราะห์และวิจารณ์ระบบ

โครงงานนี้มีจุดประสงค์เพื่อตอบสนองปัญหาและช่วยอำนวยความสะดวกในการใช้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆหลายๆ แห่งพร้อมกัน ด้วยการพัฒนาเว็บไซต์ที่แสดงให้ผู้ใช้เห็นว่า ผู้ใช้ได้ใช้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆเพียงอันเดียวแต่โดยแท้จริงแล้วเกิดจากการรวมแหล่งเก็บข้อมูลบนกลุ่มเมฆหลายบริการไว้ด้วยกันและอาศัยเทคนิคการออกแบบและการจัดการที่ดี ทั้งนี้ยังเพื่อให้ผู้ใช้สามารถใช้พื้นที่การเก็บข้อมูลได้อย่างคุ้มค่ามากที่สุด และลดความสิ้นเปลืองให้กับผู้ให้บริการอีกด้วย

ในการพัฒนาระบบดังกล่าว ผู้พัฒนาได้เลือกใช้ภาษา PHP ในการพัฒนาเพราะเป็นภาษาที่นิยมมากในการเขียนเว็บไซต์ และเนื่องจากเว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆนั้นต้องมีการทำงานร่วมกับ Library และ API เพื่อที่จะเข้าถึงแหล่งเก็บข้อมูลบนกลุ่มเมฆ จึงเลือกที่จะใช้ภาษา PHP ที่นิยมและมี Library ที่สนับสนุนจำนวนมาก

3.3. การออกแบบระบบใหม่

3.3.1. บล็อคไดอะแกรม (Block Diagram)



รูปที่ 3 แผนภาพแสดงโครงสร้างภาพรวมของระบบ

จากแผนภาพบล็อคไดอะแกรมจะเห็นได้ว่าเว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลจะทำหน้าที่เป็นตัวกลางสำหรับติดต่อกับ API ของผู้ให้บริการทั้ง 5 ซึ่งส่งคำสั่งการทำงานผ่านอินเตอร์เน็ตไปยังเซิร์ฟเวอร์หลักของผู้ให้บริการเจ้าต่างๆ เพื่อดึงข้อมูลที่ต้องการมาใช้งาน

4. ระบบต้นแบบและผลการศึกษา

ในส่วนนี้จะเป็นข้อมูลของผลที่ได้จากการศึกษา API ของผู้ให้บริการแต่ละอัน โดยจะกล่าวถึงส่วนประกอบและวิธีการใช้งานโดยสรุป ซึ่งจะมีรายชื่อผู้ให้บริการดังนี้ ได้แก่ Dropbox, Copy, OneDrive, Google Drive และ Box และกล่าวถึงการพัฒนาระบบต้นแบบโดยสรุป

4.1. ผลการศึกษา API ของผู้ให้บริการโดยสรุป

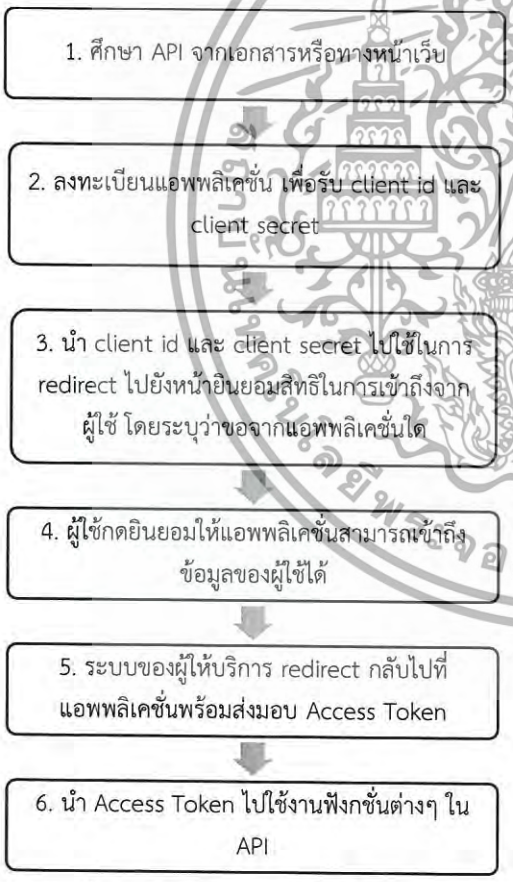
จากการศึกษาการทำงานของ API ทั้ง 5 ของผู้ให้บริการแล้ว จึงได้องค์ความรู้ที่สำคัญมามากมาย และสามารถสรุปหาจุดร่วมของ API ได้พอสังเขปอีกด้วย

ลักษณะการทำงานของ API ที่ได้ศึกษามานั้นส่วนมากจะทำงานด้วยรูปแบบของ REST ซึ่งใช้ HTTP/HTTPS ในการทำงาน ซึ่งในการเข้าถึงแต่ละฟังก์ชั่นของผู้ให้บริการที่ API สนับสนุนนั้น จะใช้กำหนดจาก URL ซึ่งผู้ให้บริการแต่ละเจ้า จะมี Hostname ของเซิร์ฟเวอร์ที่เป็นตัวให้บริการ API ที่แตกต่างกัน และยังกำหนด URL Endpoint ที่ต่างกันอีกด้วย ซึ่งด้วยการออกแบบให้ทำงานด้วย REST ทำให้ HTTP Verbs ต่างๆ เช่น GET, POST, PUT, PATCH, DELETE ทำให้ URL Endpoint ทำงานได้หลากหลายมากยิ่งขึ้น และหลังจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่เรียกใช้ฟังก์ชั่นการทำงานของผู้ให้บริการ ผลลัพธ์ที่ได้มานั้นส่วนมากจะได้รับข้อมูลกลับมาในรูปแบบของ JSON เพราะเป็นวิธีการส่งข้อมูลที่มีขนาดเล็กและรวดเร็ว

ทั้งนี้ช่องทางในการใช้ API ของผู้ให้บริการนั้นมีมากมาย ไม่ว่าจะเป็น Library ที่ผู้ให้บริการจัดเตรียมไว้ให้นำไปใช้อยู่แล้ว ซึ่งรองรับในหลายภาษาหรือ Library ที่มีผู้พัฒนาคนอื่นๆ ที่ได้เขียนขึ้นมาเองโดยเรียกใช้ API และแบ่งปันให้คนอื่นนำไปใช้ต่อได้ หลักการในการเลือกใช้ Library นั้น ควรจะเลือกใช้ Library ที่มีเอกสารหรือการเขียนบรรยายที่เข้าใจง่ายและสามารถนำมาประยุกต์ใช้กับแอพพลิเคชั่นของตนได้ ทั้งนี้ควรจะดูกระแสตอบรับจากผู้พัฒนาคนอื่นๆ ที่มีต่อ Library นั้นด้วย



รูปที่ 4 แผนผังหลักการการใช้ API โดยสังเขป

รูปที่ 5 หน้ายินยอมสิทธิในการเข้าถึงบัญชีผู้ใช้จาก Dropbox Application

ระบบของผู้ให้บริการอาจจะส่งมอบ Access Token หรือ Authorization Code โดยที่ผู้ใช้จะไม่เห็นกระบวนการดังกล่าว ซึ่งบางผู้ให้บริการก็อาจจะมีการส่งมอบสองขั้นตอนโดยใช้ Authorization code ยืนยันก่อนเพื่อรับ Access Token อีกครั้ง หรือบางผู้ให้บริการก็จะส่งมอบ Access Token มาให้เลย ซึ่งอาจจะมีการใช้ OAuth เพื่อส่งเสริมความปลอดภัยบนเครือข่าย และเมื่อแอพพลิเคชั่นได้รับ Access Token มาแล้ว ก็จะสามารถเก็บไว้ได้ ซึ่งในบางกรณีนั้น Access Token ที่ผู้ให้บริการให้มาจะมีวันหมดอายุ จึงต้องใช้ Refresh Token ในการร้องขอต่ออายุของ Access Token นั้นๆ อีกด้วย

4.2. ระบบต้นแบบ

ระบบต้นแบบที่ทางผู้พัฒนาได้จัดทำขึ้นนั้น ได้พัฒนาขึ้นมาด้วยภาษา PHP โดยพัฒนาตามหลักของ MVC ซึ่งได้ใช้ Framework ที่ชื่อว่า Laravel 5.1 [3] เข้ามาช่วยในการพัฒนา และใช้ Relational Database คือ MySQL ในการเก็บข้อมูลของระบบ นอกจากนี้ยังใช้ Library ที่ช่วยในการเก็บข้อมูลแบบ Nested Set ที่มีชื่อว่า etrepat/baum [8] อีกด้วย

ระบบต้นแบบที่เราได้พัฒนานั้นพยายามที่จะออกแบบให้มีการทำงานที่เป็นสัดเป็นส่วนชัดเจน และพยายามพัฒนาให้เข้าใจง่าย เพื่อที่จะสามารถนำไปพัฒนาประยุกต์ใช้กับผู้ให้บริการอื่นๆ ได้ง่ายมากยิ่งขึ้น นอกจากนี้ยังออกแบบระบบให้รองรับต่อการเพิ่มหรือลดผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆอีกด้วย เผื่อในกรณีที่ต้องการให้ระบบรองรับผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆมากขึ้นกว่าเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6 หน้าจัดการไฟล์ของระบบต้นแบบ

รูปที่ 7 หน้าเพิ่มการเชื่อมต่อกับผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆของระบบต้นแบบ

5. สรุปผล

ในการพัฒนาโครงงานเว็บไซต์สำหรับการจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆ มีวัตถุประสงค์หลักเพื่อที่จะแก้ไขปัญหาความยุ่งยากและความวุ่นวายในการจัดการและเข้าถึงข้อมูลภายในแหล่งเก็บข้อมูลบนกลุ่มเมฆ อีกทั้งยังช่วยแก้ไขปัญหาความซ้ำซ้อนของข้อมูลอีกด้วย โดยหลักการทำงานหลักของระบบจะเป็นการใช้ Single Sign-on ให้กับผู้ใช้เพื่อลงทะเบียนเชื่อมต่อแหล่งเก็บข้อมูลบนกลุ่มเมฆต่างๆ กับระบบ และระบบจะทำงานโดยเชื่อมต่อกับ API ของผู้ให้บริการ ได้แก่ Dropbox, Copy, Onedrive, Box, และ Google Drive ภายใต้แนวคิดหลักของระบบที่จะพยายามทำให้ผู้ใช้มองแหล่งเก็บข้อมูลบนกลุ่มเมฆหลายๆ บริการเป็นบริการเดียว

ผู้พัฒนาได้พัฒนาระบบมาจากการศึกษา API และพยายามออกแบบระบบให้สามารถทำงานร่วมกันได้ รวมถึงออกแบบและพัฒนาระบบในส่วนต่างๆ ไม่ว่าจะเป็นการทำงาน หรือส่วนติดต่อกับผู้ใช้ แม้กระทั่งโครงสร้างของระบบที่ง่ายต่อการแก้ไขและดูแล จึงถือว่าการศึกษาค้นคว้าและพัฒนาโครงงานดังกล่าวขึ้นมาทำให้ผู้พัฒนาได้องค์ความรู้ในด้านต่างๆ มากมาย ไม่ว่าจะเป็นความรู้ทางด้านทฤษฎีต่างๆ หรือจะเป็นทักษะที่ได้จากการทดลองและลงมือทำ นอกจากนี้ระหว่างการพัฒนานั้นก็ได้พบกับปัญหาและอุปสรรคต่างๆ มากมาย ทำให้ผู้พัฒนาได้ฝึกทักษะในการแก้ไขปัญหาและเรียนรู้อะไรใหม่ๆ เสมอ ไม่ว่าจะเป็นการแบ่งเวลาในการศึกษาและพัฒนาให้เหมาะสม หรือปัญหาจากการออกแบบที่ต้องรองรับการทำงานในอนาคตอีกด้วย

ระบบต้นแบบที่ได้พัฒนาขึ้นมานั้นยังมีข้อจำกัดอยู่บางส่วน ได้แก่ ไม่สามารถใช้งานฟังก์ชั่นที่เป็นจุดเด่นของแต่ละผู้ให้บริการแหล่งเก็บข้อมูลบนกลุ่มเมฆได้อย่างเต็มที่ ข้อจำกัดที่เกี่ยวข้องกับการเปลี่ยนแปลงที่เกี่ยวข้องกับ API ไม่ว่าจะเป็นการอัพเกรดเวอร์ชั่น ก็อาจจะส่งผลกับระบบทำให้ต้องมีการดูแลและอัพเดทระบบอย่างสม่ำเสมอ

ในอนาคตการเก็บข้อมูลบนแหล่งเก็บข้อมูลบนกลุ่มเมฆจะเป็นที่นิยมมากยิ่งขึ้น โครงงานเว็บไซต์สำหรับจัดการแหล่งเก็บข้อมูลบนกลุ่มเมฆจึงควรจะมีการพัฒนาระบบอย่างสม่ำเสมอ โดยทิศทางรวมถึงฟังก์ชั่นและฟีเจอร์ในการพัฒนานั้นมีมากมายที่จะทำให้ระบบกลายเป็นระบบที่มีเอกลักษณ์และมีความแตกต่างมากยิ่งขึ้น และด้วยองค์ความรู้ที่ได้จากการศึกษา พัฒนาโครงงานยังสามารถนำไปสู่งานวิจัยด้านต่างๆ ที่มีเนื้อหาในเชิงลึกมากยิ่งขึ้นอีกด้วย

เอกสารอ้างอิง

[1] Estanislau Trepat (etrepat). **"Baum: Nested Set pattern for Laravel 5's Eloquent ORM."** [Online]. Available: https://github.com/etrepat/baum. 2016

[2] MultCloud Developer Team. **"MultCloud."** [Online]. Available: https://multcloud.com. 2016

[3] Taylor Otwell. **"Laravel 5.1 Documentation."** [Online]. Available: http://laravel.com/docs/5.1. 2015

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้