ระบบบริหารจัดการคลาวด์ สำหรับภาควิชาวิศวกรรมคอมพิวเตอร์

COMPUTER ENGINEERING CLOUD PLATFORM

ซาฟีรีน โตะกานี

ปฏิยุทธ์ หวังทอง

เพิ่มพูน บุญฤทธิ์

สิระติ หิรัญธานี

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2567

ปริญญานิพนธ์ปีการศึกษา 2567

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบบริหารจัดการคลาวด์ สำหรับภาควิชาวิศวกรรมคอมพิวเตอร์

COMPUTER ENGINEERING CLOUD PLATFORM

ผู้จัดทำ

1. นายซาฟีรีน โตะกานี รหัสนักศึกษา 64010192
2. นายปฏิยุทธ์ หวังทอง รหัสนักศึกษา 64010479
3. นายเพิ่มพูน บุญฤทธิ์ รหัสนักศึกษา 64010633
4. นายสิระติ หิรัญธานี รหัสนักศึกษา 64010899

_____ อาจารย์ที่ปรึกษา

(อาจารย์จิระศักดิ์ สิทธิกร)

_____ อาจารย์ที่ปรึกษาร่วม

(อาจารย์ปริญญา เอกปริญญา)

ระบบบริหารจัดการคลาวด์ สำหรับภาควิชาวิศวกรรมคอมพิวเตอร์

นายซาฟีรีน	โตะกานี	64010192
นายปฏิยุทธ์	หวังทอง	64010479
นายเพิ่มพูน	บุญฤทธิ์	64010633
นายสิระติ	หิรัญธานี	64010899
อาจารย์จิระศักดิ์	สิทธิกร	อาจารย์ที่ปรึกษา
อาจารย์ปริญญา	เอกปริญญา	อาจารย์ที่ปรึกษาร่วม

ปีการศึกษา 2567

บทคัดย่อ

โครงงานนี้เป็นการพัฒนาระบบบริหารจัดการคลาวด์ ซึ่งมีลักษณะเป็นระบบคลาวด์ส่วนตัว (Private Cloud) สำหรับบริการเครื่องคอมพิวเตอร์แบบเสมือนให้กับนักศึกษา ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เพื่อส่งเสริมการเรียน การทำงาน และการวิจัย โดยโครงสร้างหลักของระบบคลาวด์ใช้ OpenStack สำหรับใช้บริหารจัดการทรัพยากรคอมพิวเตอร์แบบเสมือน อาทิเช่น การประมวลผล เครือข่าย และการจัดเก็บข้อมูล เพื่อให้ระบบสามารถใช้งานได้กับโครงสร้างและข้อจำกัดของสถาบันฯ จึงสร้างเว็บพร็อกซีเพื่อให้ผู้ใช้สามารถเข้าถึงและจัดการทรัพยากรในคลาวด์ได้อย่างสะดวก นอกจากนี้ยังนำ Microsoft Active Directory (AD) มาใช้ในการจัดเก็บและบริหารจัดการข้อมูลผู้ใช้งาน เพื่อให้มั่นใจว่ามีการยืนยันตัวตนและควบคุมสิทธิการเข้าถึงระบบอย่างเหมาะสม

COMPUTER ENGINEERING CLOUD PLATFORM

Mr. Safireen	Tokanee	64010192
Mr. Patiyut	Wangtong	64010479
Mr. Permpoon	Boonyarit	64010633
Mr. Sirati	Hiranthani	64010899
Mr. Jirasak	Sittikorn	Advisor
Mr. Parinya	Ekparinya	Co-Advisor

Academic Year 2024

ABSTRACT

This project aims to develop a private cloud system for students in the Faculty of Computer Engineering at King Mongkut's Institute of Technology Ladkrabang (KMITL), utilizing OpenStack as the core infrastructure. The private cloud provides virtualized computing resources and services tailored to the needs of students, including computing, networking, and storage capabilities. A custom-developed web proxy interface allows users to easily access and manage their cloud resources. Additionally, Microsoft Active Directory (AD) is integrated to securely store and manage user information, ensuring proper authentication and role-based access control within the system. This private cloud system is designed to enhance learning experiences by providing a scalable, flexible, and secure environment for practical coursework and research.

กิตติกรรมประกาศ

โครงงานในภาคการศึกษานี้สำเร็จลุล่วงไปได้ด้วยดีจากความอนุเคราะห์ช่วยเหลือจากหลากหลายบุคคล ดังนั้นจึงขอแสดงความขอบคุณทุกท่านที่อนุเคราะห์ช่วยเหลือดังรายนามต่อไปนี้

ขอขอบคุณอาจารย์จิระศักดิ์ สิทธิกร อาจารย์ที่ปรึกษาโครงงาน และ ดร. ปริญญา เอกปริญญา อาจารย์ที่ปรึกษาโครงงานร่วม ผู้ให้ทั้งองค์ความรู้ ความช่วยเหลือชี้แนะแนวทางต่างๆ การให้คำปรึกษาเพื่อหาทางออกของปัญหา รวมไปถึงสิทธิในการเข้าถึงเครื่องคอมพิวเตอร์แม่ข่ายและทรัพยากรระบบอันเป็นประโยชน์ต่อการพัฒนาโครงงานนี้เป็นอย่างยิ่ง

ขอขอบคุณอาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ได้ประสิทธิ์ประศาสน์วิชาองค์ความรู้มาตลอดระยะเวลา 4 ปี ซึ่งเป็นความรู้พื้นฐานและสามารถนำมาต่อยอด ทำให้โครงงานนี้สำเร็จได้

ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ให้การสนับสนุน อำนวยความสะดวก เอื้อเฟื้อสถานที่ในการปฏิบัติงาน และทรัพยากรต่างๆที่ใช้ในการพัฒนาโครงงาน

ขอขอบคุณเพื่อนๆ พี่ๆ วิศวกรรมคอมพิวเตอร์ ที่ให้คำปรึกษา และแลกเปลี่ยนองค์ความรู้ซึ่งกันและกัน

สุดท้ายนี้ขอขอบคุณบิดา มารดา ครอบครัวและผู้ช่วยเหลือที่เกี่ยวข้องทั้งหมด ที่เลี้ยงดูอบรมความรู้ คุณธรรม จริยธรรม และสนับสนุนด้านการศึกษามาโดยตลอดจนถึงโอกาสนี้

ซาฟีรีน โตะกานี
ปฏิยุทธ์ หวังทอง
เพิ่มพูน บุญฤทธิ์
สิระติ หิรัญธานี

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
สารบัญ IV
สารบัญตาราง VI
สารบัญรูป VIII

บทที่ 1 บทนำ 1
1.1 ความเป็นมาของปัญหา 1
1.2 วัตถุประสงค์ของโครงงาน 2
1.3 ประโยชน์ที่คาดว่าจะได้รับ 2
1.4 ขอบเขตของโครงงาน 3
1.5 แผนการดำเนินงาน 6

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 7
2.1 เทคโนโลยีจำลองเสมือนจริง (VIRTUALIZATION TECHNOLOGY) 7
2.2 เทคโนโลยีเครือข่ายที่กำหนดด้วยซอฟต์แวร์ (SOFTWARE-DEFINED NETWORKING) 10
2.3 เทคโนโลยีระบบจัดเก็บข้อมูลที่กำหนดด้วยซอฟต์แวร์ (SOFTWARE-DEFINED STORAGE) 11
2.4 เทคโนโลยีประมวลผลคลาวด์ (CLOUD COMPUTING) 12
2.5 เทคโนโลยีระบบยืนยันตัวตน 16
2.6 เทคโนโลยีการเข้าถึงเครือข่ายจากระยะไกล 17
2.7 เทคโนโลยีเครือข่ายส่วนตัวเสมือน 20
2.8 REVERSE PROXY 21
2.9 DOMAIN NAME SYSTEM (DNS) 22
2.10 REMOTE AUTHENTICATION DIAL-IN USER SERVICE (RADIUS) 23
2.11 LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL (LDAP) 24

บทที่ 3 การออกแบบและการพัฒนา 27

3.1 ภาพรวมของระบบ 27
3.2 ประเภทของผู้ใช้งานในระบบ 29
3.3 SYSTEM REQUIREMENTS SPECIFICATION 30
3.4 CENTRALIZED AUTHENTICATION SYSTEM 34
3.5 ระบบเครือข่ายของเครื่องคอมพิวเตอร์เสมือน (VIRTUAL MACHINE) ภายใน OPENSTACK 35
3.6 SOFTWARE DIAGRAM 35
3.7 การออกแบบส่วนติดต่อกับผู้ใช้งาน 46

บทที่ 4 การทดลองและผลการทดลอง 53
4.1 ทดสอบการทำงานของแพลตฟอร์ม 53
4.2 การทำงานร่วมกันของระบบ 75
4.2 CENTRALIZED AUTHENTICATION SYSTEM 81
4.3 VIRTUAL PRIVATE NETWORK (VPN) 83

บทที่ 5 สรุปผล และข้อเสนอแนะ 87
5.1 สรุปผลการพัฒนาโครงงาน 87
5.2 ปัญหาและอุปสรรคที่พบ 87
5.3 แนวทางในการพัฒนาต่อ 88

สารบัญตาราง

ตาราง **หน้า**

ตารางที่ 1.1 แผนการดำเนินงาน 6
ตารางที่ 3.1 ประเภทผู้ใช้งานในระบบ 29
ตารางที่ 3.2 SYSTEM REQUIREMENTS SPECIFICATION 30
ตารางที่ 3.3 รายละเอียดฐานข้อมูลตาราง USER 39
ตารางที่ 3.4 รายละเอียดฐานข้อมูลตาราง SUBJECT 40
ตารางที่ 3.5 รายละเอียดฐานข้อมูลตาราง GROUP 40
ตารางที่ 3.6 รายละเอียดฐานข้อมูลตาราง CREDIT 40
ตารางที่ 3.7 รายละเอียดฐานข้อมูลตาราง PUBLIC_KEY 41
ตารางที่ 3.8 รายละเอียดฐานข้อมูลตาราง INSTANCE_INFO 41
ตารางที่ 4.1 ผลการทดสอบของ USE CASE LOGIN (LOGIN สำเร็จ) 53
ตารางที่ 4.2 ผลการทดสอบของ USE CASE LOGIN (LOGIN ไม่สำเร็จ) 54
ตารางที่ 4.3 ผลการทดสอบของ USE CASE LOGOUT 55
ตารางที่ 4.4 ผลการทดสอบของ USE CASE CHANGE PASSWORD 55
ตารางที่ 4.5 ผลการทดสอบของ USE CASE FORGOT PASSWORD 56
ตารางที่ 4.6 ผลการทดสอบของ USE CASE CREATE INSTANCE 57
ตารางที่ 4.7 ผลการทดสอบของ USE CASE CREATE INSTANCE (ผู้ใช้งานไม่มีสิทธิ์ในการสร้าง) 57
ตารางที่ 4.8 ผลการทดสอบของ USE CASE CREATE INSTANCE (ทรัพยากรไม่เพียงพอ) 58
ตารางที่ 4.9 ผลการทดสอบของ USE CASE ACCESS INSTANCE (เข้าใช้งาน INSTANCE ผ่าน VNC) 59
ตารางที่ 4.10 ผลการทดสอบของ USE CASE MANAGE INSTANCE (ผู้ใช้งานแก้ไข INSTANCE ของตนเอง) 59
ตารางที่ 4.11 ผลการทดสอบของ USE CASE MANAGE INSTANCE (ผู้ดูแลรายวิชาแก้ไข INSTANCE ของผู้อื่น) 60
ตารางที่ 4.12 ผลการทดสอบของ USE CASE MANAGE INSTANCE (ผู้ใช้งานลบ INSTANCE ของตนเอง) 61
ตารางที่ 4.13 ผลการทดสอบของ USE CASE MANAGE INSTANCE (ผู้ดูแลรายวิชาลบ INSTANCE ของผู้อื่น) 62
ตารางที่ 4.14 ผลการทดสอบของ USE CASE ADD PUBLIC KEY 63

ตารางที่ 4.15 ผลการทดสอบของ USE CASE MANAGE PUBLIC KEY (ลบ) 63
ตารางที่ 4.16 ผลการทดสอบของ USE CASE DOWNLOAD VPN 64
ตารางที่ 4.17 ผลการทดสอบของ USE CASE REQUEST SUBJECT 64
ตารางที่ 4.18 ผลการทดสอบของ USE CASE MANAGE GROUP (สร้าง GROUP) 65
ตารางที่ 4.19 ผลการทดสอบของ USE CASE MANAGE GROUP (แก้ไข GROUP) 66
ตารางที่ 4.20 ผลการทดสอบของ USE CASE MANAGE GROUP (ลบ GROUP) 66
ตารางที่ 4.21 ผลการทดสอบของ USE CASE MANAGE USER (เพิ่ม USER เข้าสู่ SUBJECT) 67
ตารางที่ 4.22 ผลการทดสอบของ USE CASE MANAGE USER (เพิ่ม USER หลายคนเข้าสู่ SUBJECT) 68
ตารางที่ 4.23 ผลการทดสอบของ USE CASE MANAGE USER (ลบ USER ออกจาก SUBJECT) 68
ตารางที่ 4.24 ผลการทดสอบของ USE CASE MANAGE QUOTA (อนุมัติคำขอ QUOTA) 69
ตารางที่ 4.25 ผลการทดสอบของ USE CASE MANAGE QUOTA (ปฏิเสธคำขอ QUOTA) 70
ตารางที่ 4.26 ผลการทดสอบของ USE CASE CREATE SUBJECT (สร้าง SUBJECT) 70
ตารางที่ 4.27 ผลการทดสอบของ USE CASE SUBJECT (แก้ไข) 71
ตารางที่ 4.28 ผลการทดสอบของ USE CASE MANAGE SUBJECT (ลบ) 72
ตารางที่ 4.29 ผลการทดสอบของ USE CASE MANAGE CREDIT (อนุมัติ) 72
ตารางที่ 4.30 ผลการทดสอบของ USE CASE MANAGE CREDIT (ปฏิเสธ) 73
ตารางที่ 4.31 ผลการทดสอบของ USE CASE MANAGE CREDIT (เพิ่ม) 74
ตารางที่ 4.32 ผลการทดสอบของ USE CASE REQUEST CREDIT 74

สารบัญรูป

รูป **หน้า**

รูปที่ 2.1 KERNEL-BASED VIRTUAL MACHINE STRUCTURE 8
รูปที่ 2.2 KERNEL-BASED VIRTUAL MACHINE STRUCTURE 9
รูปที่ 2.3 แสดงการทำงานของ LIBVIRT 10
รูปที่ 2.4 ฟีเจอร์ของ OPEN VSWITCH 11
รูปที่ 2.5 OPENSTACK CLOUD LANDSCAPE MAP 14
รูปที่ 2.6 การนำ ACTIVE DIRECTORY มาใช้งาน 17
รูปที่ 2.7 อธิบายขั้นตอนการทำงานของ SECURE SHELL 18
รูปที่ 2.8 เปรียบเทียบการใช้งาน VPN และไม่ใช้งาน VPN 21
รูปที่ 2.9 อธิบายการทำงานของ REVERSE PROXY 21
รูปที่ 2.10 อธิบายการทำงานของ DOMAIN NAME SYSTEM 23
รูปที่ 2.11 ตัวอย่างโครงสร้างของ LDAP DIRECTORY 25
รูปที่ 3.1 ภาพรวมของระบบ 27
รูปที่ 3.2 ลำดับชั้นของคลาวด์แพลตฟอร์ม 28
รูปที่ 3.3 การเชื่อมต่อของระบบยืนยันตัวตน 34
รูปที่ 3.4 USE CASE DIAGRAM ของการจัดการ DOMAIN 35
รูปที่ 3.5 USE CASE DIAGRAM ของการจัดการรายวิชา 36
รูปที่ 3.6 USE CASE DIAGRAM ของการจัดการ INSTANCE 37
รูปที่ 3.7 USE CASE DIAGRAM ของการจัดการการเข้าถึง 37
รูปที่ 3.8 USE CASE DIAGRAM ของการยืนยันตัวตน 38
รูปที่ 3.9 การออกแบบฐานข้อมูล 39
รูปที่ 3.10 SEQUENCE DIAGRAM ของการเข้าสู่ระบบ 42
รูปที่ 3.11 SEQUENCE DIAGRAM ของการเพิ่ม PUBLIC KEY 42
รูปที่ 3.12 SEQUENCE DIAGRAM ของการดาวน์โหลด VPN ของผู้ใช้งาน 43
รูปที่ 3.13 SEQUENCE DIAGRAM ของการเพิ่มผู้ใช้งานด้วยรหัสนักศึกษา 43
รูปที่ 3.14 SEQUENCE DIAGRAM ของการเพิ่มนักศึกษาเข้าในกลุ่มย่อยของรายวิชา 44
รูปที่ 3.15 SEQUENCE DIAGRAM ของการเพิ่มกลุ่มของนักศึกษาเข้าในรายวิชาด้วยไฟล์ CSV 44

รูปที่ 3.16 SEQUENCE DIAGRAM ของการสร้าง INSTANCE 45
รูปที่ 3.17 SEQUENCE DIAGRAM ของการแก้ไข INSTANCE 45
รูปที่ 3.18 SEQUENCE DIAGRAM ของการเข้าถึง INSTANCE ด้วย VNC 46
รูปที่ 3.19 หน้าเข้าสู่ระบบ (LOGIN) 46
รูปที่ 3.20 หน้าแสดงรายวิชา (SUBJECT) 47
รูปที่ 3.21 หน้าการสร้างรายวิชา (SUBJECT) 47
รูปที่ 3.22 หน้าการขอโควตา (QUOTA) 48
รูปที่ 3.23 หน้าแสดงรายการ INSTANCE 48
รูปที่ 3.24 หน้าการสร้าง INSTANCE 49
รูปที่ 3.25 หน้าแสดงกลุ่ม (GROUP) ทั้งหมด 49
รูปที่ 3.26 หน้าการสร้างกลุ่ม (GROUP) 50
รูปที่ 3.27 หน้าแสดงรายชื่อนักศึกษา 50
รูปที่ 3.28 หน้าการเพิ่มนักศึกษาเข้าในรายวิชา 51
รูปที่ 3.29 หน้าการขอเครดิต 51
รูปที่ 3.30 หน้าการจัดการโควตา 52
รูปที่ 3.31 หน้าการจัดการเครดิต (CREDIT) 52
รูปที่ 4.1 ผลลัพธ์ของการสร้างคอมพิวเตอร์เสมือน 78
รูปที่ 4.2 ผลลัพธ์ของแสดงรายการของ PROXY HOST ทั้งหมด 80
รูปที่ 4.3 ผลลัพธ์ของการเพิ่ม PROXY HOST 81
รูปที่ 4.4 ผลลัพธ์ของการเชื่อมต่อ OPENSTACK กับ WINDOWS ACTIVE DIRECTORY 82
รูปที่ 4.5 ตัวอย่างการตั้งค่าเชื่อมต่อ KEYCLOAK กับ WINDOWS ACTIVE DIRECTORY 83
รูปที่ 4.6 ผลลัพธ์เชื่อมต่อ KEYCLOAK กับ WINDOWS ACTIVE DIRECTORY 83
รูปที่ 4.7 การตั้งค่าเชื่อมต่อ RADIUS SERVER บน VPN SERVER 84
รูปที่ 4.8 การตั้งค่า RADIUS SERVER บน WINDOWS ACTIVE DIRECTORY 85
รูปที่ 4.9 การตั้งค่า POLICY ของ RADIUS SERVER บน WINDOWS ACTIVE DIRECTORY 85
รูปที่ 4.10 การตั้งค่า ROUTING TABLE สำหรับผู้ดูแลระบบ 86
รูปที่ 4.11 การตั้งค่า ROUTING TABLE สำหรับผู้ใช้งานทั่วไป 86

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

การถือกำเนิดของอินเทอร์เน็ตทำให้ระบบสารสนเทศพัฒนาขึ้นอย่างก้าวกระโดด ระบบสารสนเทศมีความสำคัญมายิ่งขึ้นในทุกๆ ภาคส่วนของโลก การใช้งานคอมพิวเตอร์เป็นที่แพร่หลายมากยิ่งขึ้น เป็นสาเหตุที่ทำให้ความรู้และความเข้าใจระบบคอมพิวเตอร์เริ่มจำเป็นมากยิ่งขึ้นในปัจจุบัน การถือกำเนิดของบริการระดับโลกต่างๆ บนระบบสารสนเทศในปัจจุบันนั้น ล้วนมาจากโครงสร้างพื้นฐานของระบบคอมพิวเตอร์ทั่วโลกที่ทำให้ผู้ใช้สามารถเข้าถึงบริการที่อยู่ต่างประเทศหรือต่างทวีปได้ แต่หนึ่งในปัจจัยที่สำคัญที่ทำให้ระบบสารสนเทศในปัจจุบันมีความก้าวหน้าแบบทุกวันนี้ก็คงหนีไม่พ้นบุคลากรที่มีคุณภาพ

การเพิ่มบุคลากรที่เชี่ยวชาญระบบสารสนเทศนั้นถือเป็นสิ่งที่ภาควิชาวิศวกรรมคอมพิวเตอร์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังได้ดำเนินการมาอย่างต่อเนื่องตลอดหลายทศวรรษที่ผ่านมา ทางภาควิชาได้มีการเพิ่มความเข้มข้นในวิชาปฏิบัติเพื่อให้นักศึกษาของภาควิชาได้สัมผัสการทำงานจริงที่ไม่สามารถเรียนรู้ผ่านตำราได้ สิ่งสำคัญที่ทำให้การเรียนรู้การปฏิบัติจริงสัมฤทธิ์ผลก็คือทรัพยากรที่เหมาะสม ซึ่งภาควิชาฯก็ได้สนับสนุนมาตลอด ไม่ว่าจะเป็นอุปกรณ์ทางฮาร์ดแวร์ อุปกรณ์เครือข่ายหรือเครื่องคอมพิวเตอร์สำหรับการเขียนโปรแกรมก็มีให้นักศึกษาได้เรียนรู้ แต่ในปัจจุบัน นอกจากควาามสามารถในการเขียนโปรแกรมแล้ว ความสามารถอื่นๆ ในการพัฒนาซอฟต์แวร์ก็มีความจำเป็นเช่นกัน ไม่ว่าจะเป็นการฝึกใช้งานเครื่องแม่ข่าย การติดตั้งซอฟต์แวร์บนระบบปฏิบัติการต่างๆ การใช้งานลินุกซ์ขั้นพื้นฐานและอื่นๆ ล้วนเป็นความสามารถที่จำเป็นในการพัฒนาซอฟต์แวร์ที่จำเป็น หนึ่งในทางออกที่ทำให้นักศึกษาสามารถเรียนรู้สิ่งต่างๆข้างต้นได้ก็คือการจำลองเครื่องคอมพิวเตอร์เสมือนขึ้นมาเพื่อเรียนรู้และทดลองการติดตั้งและเผยแพร่ (Deploy) ซอฟต์แวร์

การจำลองคอมพิวเตอร์ (Virtualization) เป็นเทคนิคที่มีการใช้งานกันมาอย่างต่อเนื่อง เนื่องจากเป็นเทคนิคที่ทำให้เราสามารถใช้งานหลายๆเครื่องคอมพิวเตอร์เสมือนบนเครื่องคอมพิวเตอร์เครื่องเดียวได้ และการจัดการเครื่องคอมพิวเตอร์เสมือนนั้นง่ายกว่าการจัดการเครื่องคอมพิวเตอร์จริงๆ แต่ในปัจจุบันการใช้งานเพียงแค่เครื่องคอมพิวเตอร์เสมือนก็ยังไม่เพียงพอที่จะเผยแพร่ซอฟต์แวร์ได้ทันทีเพราะยังมีปัจจัยอื่นๆอีกมากมาย เช่น การเชื่อมต่อเครือข่ายภายนอก การเพิ่ม DNS Record หรือ การทำ Reverse Proxy ซึ่งปัจจัยเหล่านี้ ในปัจจุบันก็มีการให้บริการคลาวด์ (Cloud) ที่เราสามารถนำซอฟต์แวร์ของเราไปทำงานบนเครื่องคอมพิวเตอร์เสมือนของผู้ให้บริการคลาวด์โดยไม่ต้องคำนึงถึงปัจจัยเหล่านี้ได้

แต่เนื่องจากบริการคลาวด์ในปัจจุบันนั้นมีค่าใช้จ่ายในการใช้งาน ทำให้นักศึกษาหลายคนเลือกที่จะไม่ใช้งานคลาวด์เพราะเหตุผลดังกล่าว เนื่องจากทางภาควิชาฯก็มีทรัพยากรในส่วนนี้ที่เป็นเครื่องแม่ข่ายสำหรับการใช้งานจำลองคอมพิวเตอร์เสมือนอยู่แล้ว จึงสามารถให้บริการคลาวด์ส่วนตัวแก่นักศึกษาของภาควิชาได้ แต่เนื่องจากซอฟต์แวร์ในปัจจุบันยังมีบางส่วนที่ต้องให้ผู้ดูแลระบบเป็นคนจัดการด้วนตนเองจึงยังไม่สะดวกสำหรับการใช้งานเนื่องจากจะมีเวลาที่ต้องรอระหว่างผู้ดูแลระบบทำการเพิ่มลดการตั้งค่าต่างๆของระบบให้นักศึกษาสามารถใช้งานได้

เนื่องโครงงานวิจัยของภาควิชาฯที่ผ่านมาเคยออกแบบและทดสอบเกี่ยวกับแนวคิดการให้บริการคลาวด์ส่วนตัวให้ภาควิชาฯแล้ว แต่ส่วนมากก็ยังมีส่วนที่ผู้ดูแลระบบต้องจัดการด้วยตนเอง โครงงานชิ้นนี้จึงจัดทำขึ้นเพื่อพัฒนาและต่อยอดแนวคิดนี้ โดยส่วนที่ผู้ดูแลยังต้องจัดการอยู่ก็จะทำการพัฒนาซอฟต์แวร์เพื่อให้การทำงานในส่วนนี้เกิดขึ้นอัตโนมัติ พร้อมกับการออกแบบนโยบายการใช้งานคลาวด์ของภาควิชาให้รองรับการใช้งานในระดับรายวิชาไปจนถึงระดับหลักสูตรต่างๆในภาควิชาฯ เพื่อให้การเรียนวิชาปฏิบัติมีทรัพยากรที่หลากหลายมากขึ้น และทำให้นักศึกษาของภาควิชาฯมีประสบการณ์ทำงานที่มากขึ้นพร้อมที่จะทำงานในอุตสาหกรรมทั้งในและต่างประเทศ

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อพัฒนาระบบบริหารจัดการคลาวด์ บริการคณาจารย์และนักศึกษาภาควิชาวิศวกรรมคอมพิวเตอร์สำหรับการเรียนการสอน
2) เพื่อส่งเสริมการพัฒนาคุณภาพการศึกษาของนักศึกษาภาควิชาวิศวกรรมคอมพิวเตอร์ ด้วยบริการคอมพิวเตอร์แบบเสมือน
3) เพื่อให้คณาจารย์และนักศึกษาภาควิชาวิศวกรรมคอมพิวเตอร์สามารถทำการทดสอบ พัฒนาและเผยแพร่ซอฟต์แวร์ประเภทเว็บไซต์ เว็บแอพพลิเคชัน โดยไม่ต้องพึ่งพาผู้ให้บริการคลาวด์ภายนอก
4) เพื่ออำนวยความสะดวกให้ผู้ดูแลระบบสามารถบริหารจัดการทรัพยากรของระบบคลาวด์

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1) นักศึกษาภายในภาควิชาวิศวกรรมคอมพิวเตอร์สามารถใช้งานคลาวด์แพลตฟอร์มสำหรับสร้างคอมพิวเตอร์แบบเสมือน เพื่อใช้ในการศึกษาวิจัยและพัฒนาซอฟต์แวร์ หรือใช้ประกอบการเรียนในรายวิชา รวมถึงสามารถใช้ในการเผยแพร่ผลงานซอฟต์แวร์ของตนเองสู่สาธารณะได้

2) อาจารย์ภายในภาควิชาวิศวกรรมคอมพิวเตอร์สามารถใช้งานคลาวด์แพลตฟอร์มสำหรับสร้างคอมพิวเตอร์แบบเสมือนให้สำหรับนักศึกษาใช้งานในรายวิชาหรือใช้ประกอบการสอนในรายวิชาได้
3) ภาควิชาวิศวกรรมคอมพิวเตอร์จะมีระบบยืนยันตัวตนกลางเพื่อใช้ในการเข้าใช้งานบริการต่างๆของภาควิชา และสามารถใช้เป็นระบบยืนยันตัวตนของซอฟต์แวร์ต่างๆของภาควิชาที่จะมีการพัฒนาขึ้นในอนาคต

1.4 ขอบเขตของโครงงาน

ระบบแพลตฟอร์มคลาวด์ของภาควิชาฯนั้นถูกออกแบบไว้เพื่อให้บุคลากรและนักศึกษาภาควิชาวิศวกรรมคอมพิวเตอร์สามารถใช้งานได้ โดยผู้ใช้จะสามารถสร้างเครื่องคอมพิวเตอร์เสมือน (Virtual Machine) ที่มีการคิดเครดิตได้ด้วยตนเอง และคณาจารย์ที่ดูแลรายวิชาที่ต้องการใช้งานเครื่องคอมพิวเตอร์เสมือนสามารถให้นักศึกษาที่ลงทะเบียนรายวิชาที่รับผิดชอบสามารถเข้าใช้งานเครื่องคอมพิวเตอร์เสมือนได้ โดยที่ผู้ใช้ทุกคนสามารถเข้าใช้งานเครื่องเสมือนจริงได้ทั้งจากเครือข่ายภายในและภายนอกภาควิชาฯ ผ่านเครือข่ายส่วนตัวเสมือน (Virtual Private Network) โดยแบ่งขอบเขตของโครงงานเป็น 2 ประเภทได้แก่

1.4.1 ส่วนโครงสร้างพื้นฐานของระบบ (Infrastructure)

1.4.1.1 ส่วนของฮาร์ดแวร์

เป็นระบบเครื่องคอมพิวเตอร์แม่ข่ายและระบบเครือข่ายของภาควิชาที่ได้มีการใช้งานอยู่แล้วในภาควิชา โดยอุปกรณ์ที่โครงงานนี้จะใช้ในการพัฒนามีดังนี้

1) เครื่องแม่ข่ายรุ่น HP ProLiant DL380p Gen8 จำนวน 2 เครื่อง ใช้สำหรับการให้บริการหลักนอกเหนือจาก OpenStack เช่น CE Cloud Console รวมถึง Reverse Proxy และ DNS
2) เครื่องแม่ข่ายรุ่น Dell PowerEdge R6515 จำนวน 1 เครื่อง ใช้สำหรับการให้บริการ OpenStack ในส่วนของ Controller Node
3) เครื่องแม่ข่ายรุ่น Dell PowerEdge R6515 จำนวน 1 เครื่อง ใช้สำหรับให้การให้บริการ OpenStack ในส่วนของ Networking
4) เครื่องแม่ข่ายรุ่น Dell PowerEdge R7625 จำนวน 3 เครื่อง ใช้สำหรับให้การให้บริการ OpenStack ในส่วนของ Computing
5) อุปกรณ์กระจายสัญญาณเครือข่าย (Switch) CRS354-48G-4S+2Q+RM

6) อุปกรณ์กระจายสัญญาณเครือข่าย (Switch) Dell PowerSwitch S5248F-ON
7) อุปกรณ์จัดหาเส้นทาง (Router) Mikrotik CCR2004-1G-12S+2XS

1.4.1.2 ส่วนของซอฟต์แวร์

เป็นระบบซอฟต์แวร์ของภาควิชาที่ได้มีการใช้งานอยู่แล้วในปัจจุบัน โดยซอฟต์แวร์ที่จะมีการใช้งานร่วมกันกับโครงงานมีดังนี้

1) OpenStack เวอร์ชัน Caracal สำหรับการให้บริการระบบคลาวด์ เช่น การสร้างเครื่องคอมพิวเตอร์แบบเสมือน
2) VMWare ESXi สำหรับสร้างเครื่องคอมพิวเตอร์เสมือน ให้กับบริการอื่นนอกเหนือจาก OpenStack รวมถึง CE Cloud Console
3) Ceph สำหรับการให้บริการเป็นระบบจัดเก็บข้อมูลแบบกระจาย (Distributed Storage System) ให้แก่ระบบภายในของ OpenStack
4) PowerDNS สำหรับการให้บริการ Domain Name System ของระบบ
5) NGINX Proxy Manager สำหรับการให้บริการ Reverse Proxy
6) MariaDB Cluster สำหรับการให้บริการ Database ให้แก่ซอฟต์แวร์ของระบบอื่นๆ
7) HAProxy และ KeepAlived สำหรับการทำ High Availability ให้กับระบบ

ซอฟต์แวร์ระบบของภาควิชาฯ ต่อไปนี้ยังไม่ได้มีการถูกติดตั้งและใช้งาน ทางผู้จัดทำโครงงานจะทำการติดตั้งและตั้งค่าซอฟต์แวร์ต่อไปนี้

1) OpenVPN Server สำหรับการให้บริการเครือข่ายส่วนตัวเสมือน
2) Microsoft Active Directory สำหรับจัดเก็บข้อมูลผู้ใช้ รวมถึงการแบ่งกลุ่มและกำหนดสิทธิ์

1.4.1.3 ส่วนเว็บแอปพลิเคชันของแพลตฟอร์มคลาวด์ (Cloud Platform Web Application)

1.4.1.3.1 พัฒนาระบบจัดการคอมพิวเตอร์เสมือน (Virtual Machine)

1.4.1.3.1.1 สำหรับผู้แลระบบคลาวด์ (Cloud Admin)

1) สามารถสร้างหรือแก้ไขคอมพิวเตอร์เสมือนในระบบทั้งหมดได้

2) สามารถบริหารจัดการทรัพยากรทั้งหมดในระบบได้
3) สามารถบริหารจัดการผู้ใช้ทั้งหมดในระบบได้
4) สามารถสร้างและบริหารจัดการกลุ่มของผู้ใช้งานได้
5) สามารถจัดสรรทรัพยากรให้แต่ละหลักสูตรได้

1.4.1.3.1.2 สำหรับผู้แลระดับหลักสูตร (Domain Admin)

1) สามารถสร้างหรือแก้ไขคอมพิวเตอร์เสมือนในหลักสูตรของตนเองได้
2) สามารถบริหารจัดการทรัพยากรในหลักสูตรได้
3) สามารถบริหารจัดการรายวิชาในหลักสูตรได้
4) สามารถบริหารจัดการผู้ใช้ในหลักสูตรได้
5) สามารถสร้างและบริหารจัดการกลุ่มของผู้ใช้งานในหลักสูตรได้

1.4.1.3.1.3 สำหรับผู้แลระดับรายวิชา (Subject Admin)

1) สามารถสร้างหรือแก้ไขคอมพิวเตอร์เสมือนในรายวิชาของตนเองได้
2) สามารถบริหารจัดการผู้ใช้ในรายวิชาที่ตนเองดูแลอยู่ได้
3) สามารถขอโควตาการใช้งานคอมพิวเตอร์เสมือนได้ผ่านระบบร้องขอโควตา

1.4.1.3.1.4 สำหรับผู้ใช้งานทั่วไป

1) สามารถสร้างและใช้งานคอมพิวเตอร์เสมือนในระบบตามสิทธิ์ที่ได้รับมาได้
2) สามารถขอสิทธิ์การใช้งานคอมพิวเตอร์เสมือนเพิ่มได้ผ่านระบบร้องขอเครดิตได้
3) สามารถดาวน์โหลดไฟล์สำหรับเข้าถึงเครือข่ายส่วนตัวเสมือน (Virtaul Private Network) ได้ เพื่อ

ใช้สำหรับการเข้าถึงเครือข่ายของคอมพิวเตอร์เสมือนของตัวเองทั้งหมดในระบบ

4) สามารถบันทึก SSH Public Key ของผู้ใช้งาน เพื่อใช้สำหรับการเข้าใช้งานคอมพิวเตอร์เสมือนของตัวเองทั้งหมดในระบบ

1.5 แผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

หัวข้อกิจกรรม	กรกฎาคม				สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน				ธันวาคม				มกราคม				กุมภาพันธ์				มีนาคม			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
ออกแบบภาพรวมของระบบ																																				
ศึกษาข้อมูลของส่วนต่างที่ได้ออกแบบไว้																																				
ปรับปรุงแก้ไขการออกแบบ																																				
พัฒนาระบบตามที่ออกแบบไว้																																				
จัดทำเล่มโครงงานและนำเสนอครั้งที่ 1																																				
พัฒนาระบบตามที่ออกแบบไว้(ต่อ)																																				
ทดสอบระบบ																																				
ปรับปรุงแก้ไขระบบ																																				
จัดทำเล่มโครงงานและนำเสนอครั้งที่ 2																																				

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

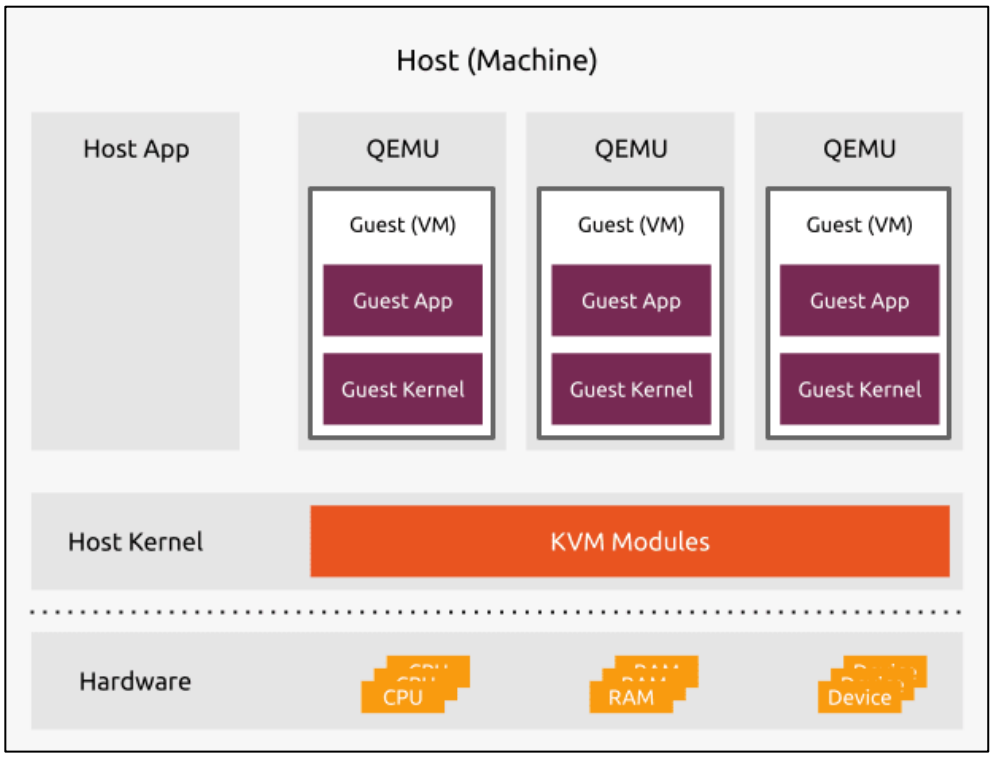
2.1 เทคโนโลยีจำลองเสมือนจริง (Virtualization Technology)

การจำลองเสมือนจริง (Virtualization) คือการจำลองทรัพยากรในระบบคอมพิวเตอร์ทางกายภาพที่มีอยู่จริงเป็นระบบคอมพิวเตอร์เสมือนแยกจากกัน ทำให้สามารถใช้งานระบบปฏิบัติการและแอพพลิเคชันได้หลายระบบพร้อมกันภายในทรัพยาการทางกายภาพจริงเพียงระบบเดียว โดยแต่ละระบบปฏิบัติการหรือแอปพลิเคชันจะมีสภาพแวดล้อมเสมือนเป็นของตัวเองโดยไม่ส่งผลกระทบต่อระบบหรือแอปพลิเคชันอื่น ที่กำลังทำงานบนทรัพยาการทางกายภาพจริงเดียวกัน

เทคโนโลยีจำลองเสมือนจริงช่วยให้การบริหารจัดการทรัพยากรเป็นไปอย่างคุ้มค่า เนื่องจากระบบคอมพิวเตอร์ไม่ว่าจะเป็นด้านการประมวลผล การจัดเก็บข้อมูล และการติดต่อสื่อสาร จะมีการทำงานหนักเฉพาะช่วงเวลา ซึ่งเป็นสูญเสียทรัพยากรในช่วงเวลาที่ไม่ทำงาน แต่เทคโนโลยีจำลองเสมือนจริงได้รวมคอมพิวเตอร์เสมือนหลายเครื่องมาที่ศูนย์กลางโดยใช้ทรัพยากรร่วมกัน ซึ่งทำให้จัดสรรการใช้ทรัพยากรได้เป็นประโยชน์ ตามความต้องการของแต่ละระบบเสมือนในช่วงเวลาต่างๆ โดยทั่วไปมีผู้ดูแลระบบคอยบริหารจัดการและ บำรุงรักษาทรัพยากรต่างๆ

2.1.1 KVM (Kernel-based Virtual Machine)

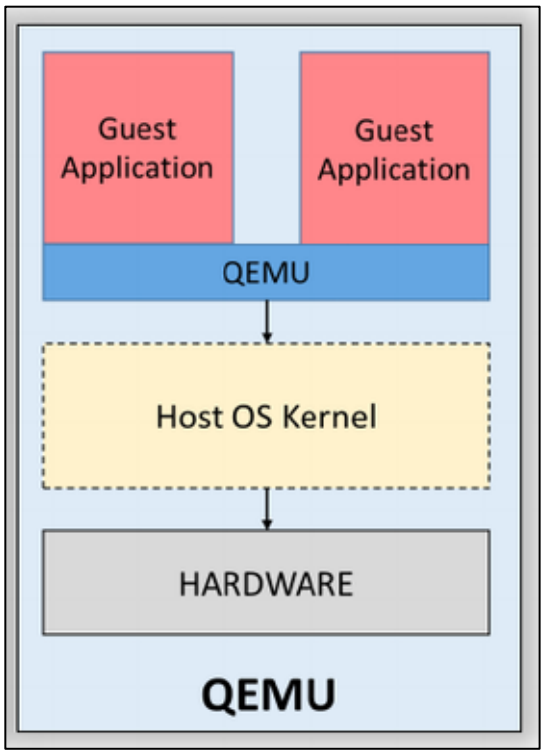
KVM (Kernel Based Virtual Machine) เป็นโมดูลฟังก์ชั่นการจำลองเสมือนจริง ที่ถูกฝังอยู่ใน Linux Kernel ซึ่งช่วยเพิ่มขีดความสามารถในการทำ Hardware Assisted Virtualization ให้กับระบบปฏิบัติการสามารถทำให้จำลองการทำงานของ CPU และ Memory ได้ เช่น การจัดการตารางงาน (Task Scheduling) การจัดการหน่วยความจำ (Memory Management) ตลอดจนการโต้ตอบกับอุปกรณ์ฮาร์ดแวร์ต่างๆ โดยทั่วไปแล้ว KVM สามารถให้บริการการจำลองเสมือน สำหรับ CPU และหน่วยความจำเท่านั้น KVM จึงต้องการส่วนประกอบเทคโนโลยีการจำลองเสมือน เพิ่มเติมเพื่อให้เครื่องเสมือน (Virtual Machine) มีการจำลองเสมือนสำหรับฮาร์ดแวร์ เช่น การ์ด เครือข่าย บัส IO การ์ดกราฟิก และอื่นๆ



รูปที่ 2.1 Kernel-based Virtual Machine Structure

2.1.2 QEMU (Quick Emulator)

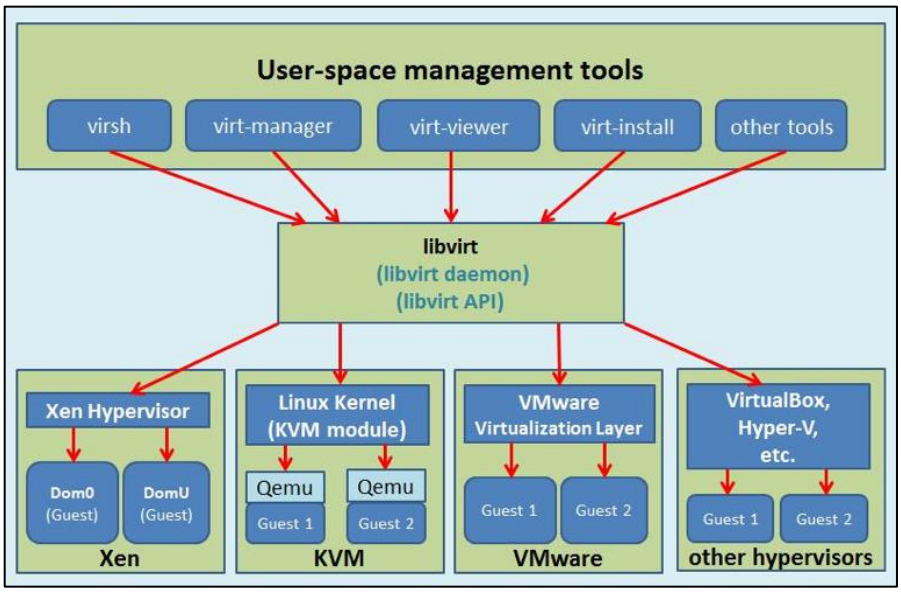
QEMU (Quick Emulator) เป็นซอฟต์แวร์ที่ใช้สำหรับการจำลอง (Emulator) ที่ทำการจำลองโปรเซสเซอร์ของเครื่องคอมพิวเตอร์ เพื่อให้สามารถใช้งานระบบปฏิบัติการหรือซอฟต์แวร์ที่ถูกพัฒนามาเพื่อสถาปัตยกรรมหนึ่ง บนเครื่องคอมพิวเตอร์ที่มีสถาปัตยกรรมแตกต่างกันสิ้นเชิงได้ เช่น x86, ARM, PowerPC, Sparc, MIPS1 โดยมีการใช้งานการแปลงแบบไดนามิก (Dynamic Translation) ทำให้ได้ประสิทธิภาพที่สูงเทียบเท่ากับการทำงานบนโปรเซสเซอร์สถาปัตยกรรมเดิม



รูปที่ 2.2 Kernel-based Virtual Machine Structure

2.1.3 Libvirt

Libvirt เป็นชุด (Collection) ของซอฟต์แวร์ที่อำนวยความสะดวกในการจัดการเครื่องคอมพิวเตอร์เสมือนและช่วยจัดการส่วนประกอบอื่นที่เกี่ยวข้องกับการจำลองเสมือนจริง เช่น การจัดการระบบเก็บข้อมูลและระบบเครือข่าย ซึ่งเป้าหมายหลักของ Libvirt คือรองรับการจัดการเครื่องคอมพิวเตอร์เสมือนจากหลายๆไฮเปอร์ไวเซอร์ (Hypervisor) ผ่าน Libvirt เพียงอย่างเดียว โดย Libvirt มีการรองรับไฮเปอร์ไวเซอร์ที่หลากหลาย อาทิเช่น KVM Xen และ VMWare ESX



รูปที่ 2.3 แสดงการทำงานของ Libvirt

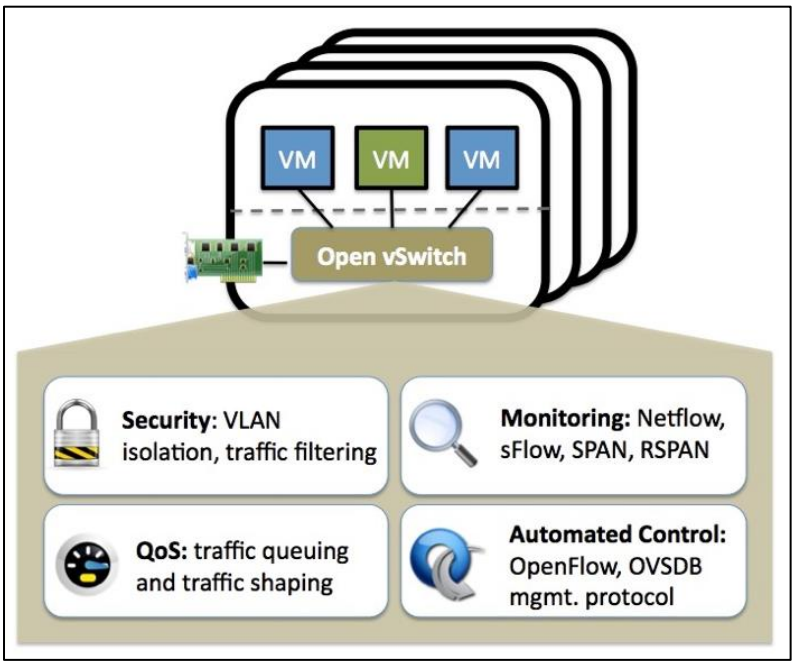
2.2 เทคโนโลยีเครือข่ายที่กำหนดด้วยซอฟต์แวร์ (Software-Defined Networking)

Software-Defined Networking (SDN) เป็นเทคโนโลยีที่มีไว้เพื่อช่วยในการจัดการและควบคุมเครือข่ายคอมพิวเตอร์โดยใช้ซอฟต์แวร์ในการควบคุม โดย SDN จะแยกการจัดการเครือข่ายออกเป็น 2 ส่วน คือส่วนของเครือข่าย (Data Plane) ซึ่งเป็นส่วนที่ทำหน้าที่ส่งข้อมูล และส่วนของการควบคุม (Control Plane) ซึ่งเป็นส่วนที่ควบคุมการเชื่อมต่อและส่งข้อมูลของเครือข่าย

ด้วย SDN ช่วยให้ผู้ดูแลระบบสามารถควบคุมเครือข่ายได้อย่างมีประสิทธิภาพและเป็นอัตโนมัติ ทำให้มีความยืดหยุ่นในการปรับเปลี่ยนและปรับแต่งเครือข่ายได้ตามต้องการ และยังช่วยลดความซับซ้อนของการจัดการเครือข่ายด้วยการใช้แนวคิดของ “Centralized Control” ที่ช่วยให้ผู้ดูแลระบบสามารถควบคุมและปรับแต่งเครือข่ายได้ด้วยระบบการควบคุมเดียวกันที่จัดตั้งอยู่ที่ศูนย์กลาง (SDN Controller) โดยอาจจะใช้โปรโตคอล OpenFlow ในการติดต่อสื่อสารระหว่าง SDN Controller และอุปกรณ์เครือข่ายที่ควบคุมได้โดยตรง (OpenFlow Switch)

2.2.1 Open vSwitch

Open vSwitch เป็นเทคโนโลยี Virtual Switch ที่รองรับทั้งความสามารถในการทำ VLAN, VLAN Isolation, Netflow, sFlow, SPAN, RSAPN, QoS, Traffic Queuing, Traffic Shaping, OpenFlow, OVSDB และ อีกหลากหลายเทคโนโลยี เพื่อรองรับการทำหน้าที่เป็น Virtual Switch บนระบบ Open Source Hypervisor หลากหลายอาทิเช่น Xen, KVM, Proxmox VE และ Virtual Box



รูปที่ 2.4 ฟีเจอร์ของ Open vSwitch

2.3 เทคโนโลยีระบบจัดเก็บข้อมูลที่กำหนดด้วยซอฟต์แวร์ (Software-Defined Storage)

เทคโนโลยีระบบจัดเก็บข้อมูลที่กำหนดด้วยซอฟต์แวร์ (Software-Defined Storage) หรือ SDS นั้นเป็นสถาปัตยกรรมของระบบจัดเก็บข้อมูลหรือสตอเรจที่แยกซอฟต์แวร์สำหรับการจัดเก็บข้อมูลออกจากตัวฮาร์ดแวร์ของสตอเรจ โดย SDS สามารถทำให้ใช้งานระบบสตอเรจได้โดยไม่ต้องสนใจว่าตัวฮาร์ดแวร์นั้นจะถูกผลิตขึ้นมาจากผู้ผลิตใด ซึ่งจะมีประโยชน์ในด้านต่างๆ เช่น สามารถใช้งานฮาร์ดแวร์จัดเก็บข้อมูลที่หลากหลายร่วมกันได้ และการเพิ่มปริมาณการเก็บข้อมูลสามารถทำได้ง่ายดาย

2.3.1 Ceph

Ceph คือ ซอฟต์แวร์โอเพนซอร์ซที่ได้รับการออกแบบมาเพื่อจัดการพื้นที่จัดเก็บข้อมูลแบบออบเจ็กต์ บล็อกและไฟล์ที่มีความยืดหยุ่นสูง Ceph ได้รับการออกแบบให้ทำงานบนฮาร์ดแวร์พื้นฐานโดยใช้อัลกอริทึมที่เรียกว่า CRUSH ทำหน้าที่ควบคุมการจำลองแบบภายใต้ Scalable Hashing เพื่อให้แน่ใจว่าข้อมูลมีการกระจายอย่างสม่ำเสมอทั่วทั้งคลัสเตอร์และโหนดคลัสเตอร์ทั้งหมดสามารถดึงข้อมูลได้อย่างรวดเร็ว

2.4 เทคโนโลยีประมวลผลคลาวด์ (Cloud Computing)

Cloud Computing คือ โครงสร้างในการให้บริการทรัพยากรทางด้านเทคโนโลยีเพื่อการคำนวณ ซึ่งผู้ใช้งานสามารถเข้าถึงได้ผ่านอินเทอร์เน็ต โดยผู้ให้บริการจะมีการพัฒนาโครงสร้างพื้นฐานเป็น Data Center ที่ประกอบด้วย Cloud Server ที่ซับซ้อนจำนวนมาก

การใช้งานจะเปรียบเสมือนการใช้งานคอมพิวเตอร์เครื่องหนึ่งที่ผู้ใช้สามารถทำงานได้จากทุกที่บนโลกเพียงแค่มีอินเทอร์เน็ต ซึ่งแตกต่างจากคอมพิวเตอร์ทั่วไปคือการใช้งานที่มีขีดจำกัดสูงกว่า ทำให้มีความเร็ว ปลอดภัย และสะดวกมากยิ่งขึ้น

2.4.1 คุณลักษณะของ Cloud Computing

2.4.1.1 On-demand Self-service

ผู้ใช้งานสามารถเข้าใช้งานทรัพยากรคอมพิวเตอร์ที่อยู่บนเครือข่ายคลาวด์ ด้วยตัวเองได้ตลอดเวลาโดยไม่ต้องอาศัยการติดต่อกับผู้ให้บริการ

2.4.1.2 Broad Network Access

ผู้ใช้งานสามารถเข้าถึงทรัพยากรคอมพิวเตอร์ที่อยู่บนเครือข่ายคลาวด์ได้ด้วยเครื่องมือ หรือแพลตฟอร์มหลากหลายชนิด อาทิเช่น โทรศัพท์มือถือ

2.4.1.3 Resource pooling

เครือข่ายคลาวด์เป็นการรวมกันของทรัพยากรคอมพิวเตอร์ของผู้ให้บริการหลายแหล่ง ที่เรียกว่า Multi-tenant Model โดยทรัพยากรคอมพิวเตอร์เหล่านี้มีทั้งแบบกายภาพ (Physical) และแบบเสมือน (Virtual) ที่ถูกจัดสรรมาไว้ในเครือข่าย Cloud ซึ่งอาจมาจากผู้ให้บริการหลายแห่งหรือหลายประเทศ โดยที่ผู้ใช้งานไม่สามารถรู้หรือควบคุมได้ว่ากำลังใช้บริการทรัพยากรคอมพิวเตอร์จากแหล่งใด

2.4.1.4 Rapid elasticity

ระบบบริการคลาวด์สามารถปรับเปลี่ยนตามรูปแบบที่ผู้ใช้งานได้อย่างรวดเร็ว

2.4.1.5 Measured service

ผู้ใช้งานและผู้ให้บริการสามารถที่จะวัดปริมาณการใช้งานระบบบริการคลาวด์ตามความเป็นจริงได้

2.4.2 รูปแบบการใช้งานของ Cloud

2.4.2.1 Public Cloud

เป็น Cloud ที่เปิดให้มีการใช้บริการแบบสาธารณะผ่านอินเทอร์เน็ต ทำให้ทั้งองค์กรและบุคคลทั่วไปสามารถเข้าถึงได้ ทั้งนี้ Public Cloud สามารถที่จะอยู่ใน

Abstraction Level ใดก็ได้ ตัวอย่างของ Public Cloud เช่น Google Print, Google Docs, Microsoft Office 365 และ Amazon EC2

2.4.2.2 Private Cloud

เป็นคลาวด์เฉพาะขององค์กร บริษัท หรือบุคคลหนึ่งๆ ที่มีการให้บริการผ่านอินเทอร์เน็ตในขอบเขตที่จำกัด หรือที่เรียกว่า Wide Area Network (WAN) โดย Private Cloud จะมีการตั้งระบบรักษาความปลอดภัยเพื่อจำกัดการใช้เฉพาะบุคคลภายในเท่านั้น ซึ่งหลักการจะคล้ายกับ Local Area Network (LAN) ที่มีการใช้กันอยู่ตามบ้านหรือสำนักงานทั่วไปแต่ WAN จะครอบคลุมบริเวณที่กว้างกว่า

2.4.2.3 Hybrid Cloud

เป็นรูปแบบผสมของ Cloud Deployment Model (Public Private Community) ตั้งแต่ 2 ประเภทขึ้นไป โดยผู้ให้บริการในคลาวด์แต่ละรายภายใต้ Hybrid Cloud นี้จะทำงานแบบเป็นอิสระต่อกัน แต่จะถูกเชื่อมกันด้วยเทคโนโลยีเฉพาะที่ทำให้ข้อมูลและ Application สามารถถูกนำมาใช้งานร่วมกันได้

2.4.3 รูปแบบการให้บริการคลาวด์

2.4.3.1 Infrastructure-as-a-Service (IaaS)

เป็นระบบที่มีโครงสร้างพื้นฐานด้านไอทีครบถ้วน คล้ายกับระบบคอมพิวเตอร์ขนาดใหญ่หรือเซิร์ฟเวอร์ โดยมีหน่วยการประมวลผล ระบบจัดเก็บข้อมูล และเครือข่ายเชื่อมต่อที่มีความปลอดภัย ทำให้ผู้ใช้งานสามารถต่อยอดการทำงานได้มากกว่าจากทรัพยากรด้านไอทีที่มี เช่น ระบบเว็บไซต์ แอปพลิเคชัน Big Data หรือ AI เป็นต้น

2.4.3.2 Platform-as-a-Service (PaaS)

เป็นระบบที่พัฒนาโครงสร้างพื้นฐานและมีการติดตั้งซอฟต์แวร์สำหรับการใช้งานที่สะดวกเพิ่มมากขึ้น โดยซอฟต์แวร์ในการใช้งานจะขึ้นอยู่กับผู้ให้บริการที่จะมีความแตกต่างกันออกไป ซึ่งทำให้ผู้ใช้งานไม่ต้องกังวลเรื่องฮาร์ดแวร์ และซอฟต์แวร์ในการใช้งาน เนื่องจากมีการจัดเตรียม Infrastructure และแอปพลิเคชันไว้แล้ว

2.4.3.3 Software-as-a-Service (SaaS)

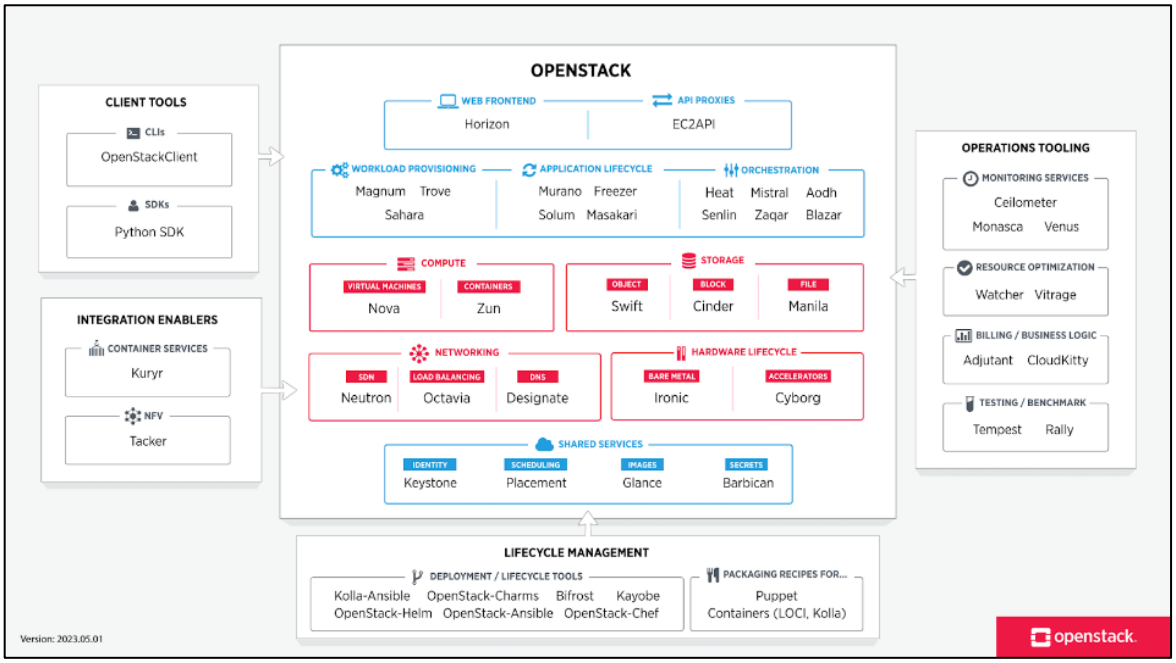
เป็นระบบที่พัฒนาโครงสร้างพื้นฐานและมีการติดตั้งซอฟต์แวร์สำหรับการใช้งานที่สะดวกเพิ่มมากขึ้น โดยซอฟต์แวร์ในการใช้งานจะขึ้นอยู่กับผู้ให้บริการที่จะมีความแตกต่างกันออกไป ซึ่งทำให้ผู้ใช้งานไม่ต้องกังวลเรื่องฮาร์ดแวร์

และซอฟต์แวร์ในการใช้งาน เนื่องจากมีการจัดเตรียม Infrastructure และแอปพลิเคชันไว้เรียบร้อยแล้ว

2.4.4 OpenStack

เป็นชุดระบบซอฟต์แวร์ Open Source ใช้ทำระบบ Private Cloud ซึ่งถูกออกแบบมาเพื่อให้บุคคลทั่วไปหรือองค์กรต่าง ๆ สามารถสร้างและจัดการโครงสร้างพื้นฐานระบบคลาวด์ในรูปแบบ Infrastructure as a Service (IaaS) โดยแพลตฟอร์มนี้ ประกอบด้วยหลากหลายโมดูลที่ทำงานร่วมกัน เพื่อให้บริการในส่วนของ การประมวลผล (compute), การเชื่อมต่อเครือข่าย (Networking), พื้นที่เก็บข้อมูล (Storage) รวมถึงทรัพยากรและเทคโนโลยีที่องค์กรต่าง ๆ ต้องเรียนรู้และศึกษาการใช้งาน เพื่อนำมาประยุกต์ใช้ให้สอดคล้องกับโครงสร้างระบบธุรกิจที่จะส่งผลให้การดำเนินงานมีประสิทธิภาพมากยิ่งขึ้น

2.4.4.1 โมดูลต่างๆใน OpenStack



รูปที่ 2.5 OpenStack Cloud landscape map

2.4.4.1.1 Keystone (Identity Service)

เป็นบริการจัดการตัวตนทำหน้าที่ตรวจสอบและอนุญาตให้ผู้ใช้เข้าถึงทรัพยากรต่างๆ

2.4.4.1.2 Nova (Compute Service)

ระบบ Computing ที่ให้บริการความสามารถในการจัดการอินสแตนซ์เสมือน (Virtual Instances)

2.4.4.1.3 Swift (Object Storage Service)

ระบบจัดเก็บข้อมูลแบบไม่มีโครงสร้าง (Unstructured Data) ที่รองรับการจัดเก็บข้อมูลขนาดใหญ่

2.4.4.1.4 Cinder (Block Storage Service)

บริการจัดเก็บข้อมูลแบบบล็อกที่ใช้สำหรับการจัดเก็บข้อมูลที่มีโครงสร้างและความต้องการสูง

2.4.4.1.5 Neutron (Networking Service)

ช่วยให้สามารถสร้างและจัดการเครือข่ายส่วนตัวภายใน OpenStack

2.4.4.1.6 Glance (Image Service)

บริการสำหรับการจัดเก็บและการเข้าถึงอิมเมจ (Images) ทีใช้สำหรับอินสแตนซ์เสมือน

2.4.4.1.7 Horizon (Dashboard)

หน้าเว็บอินเทอร์เฟซสำหรับการจัดการและการเข้าถึงทรัพยากรต่างๆ ใน OpenStack

2.4.4.2 การบริหารจัดการขอบเขต (Administrative Boundaries for Management)

2.4.4.2.1 โดเมน (Domain)

เป็นส่วนที่ห่อหุ้ม โปรเจ็กต์ (Projects) กลุ่ม (Groups) บทบาท (Roles) และ ผู้ใช้ (Users) เปรียบเสมือน "เขตพื้นที่" ที่แยกผู้ใช้ โปรเจ็กต์ และกลุ่ม ออกจากกัน ช่วยให้การจัดการระบบคลาวด์มีประสิทธิภาพและปลอดภัยมากขึ้น

2.4.4.2.2 โปรเจ็กต์ (Project)

ส่วนย่อยรองจาก โดเมน (Domain) ซึ่งภายในจะมีเครื่องคอมพิวเตอร์เสมือน, ต้นแบบของเครื่องคอมพิวเตอร์เสมือน (Images) และผู้ใช้ 1 คนสามารถอยู่ได้มากกว่า 1 โปรเจ็กต์ โดยมี บทบาท (Roles) มาคอยกำหนดสิทธิ์ (Permission)

2.4.4.2.3 กลุ่ม (Group)

ถูกสร้างขึ้นเพื่อให้ง่ายต่อการจัดการสิทธิ์การเข้าถึงทรัพยากรภายในระบบของผู้ใช้ กลุ่มช่วยให้ผู้ดูแลระบบสามารถกำหนดสิทธิ์การเข้าถึงให้กับกลุ่มผู้ใช้ทั้งหมดแทนที่จะต้องกำหนดสิทธิ์ให้กับผู้ใช้ทีละคน

2.4.4.2.4 บทบาท (Roles)

ชุดของสิทธิ์การเข้าถึง

2.4.5 ระบบเครดิต

เป็นระบบที่ผู้ใช้งานทุกคนจะได้รับเครดิต และจะหักเครดิตของผู้ใช้งานเมื่อมีการสร้างคอมพิวเตอร์เสมือนและเปิดใช้งานเป็นระยะเวลาหนึ่ง โดยเครื่องคอมพิวเตอร์เสมือนแต่ละประเภทจะมีราคา ในหน่วยเครดิตเพื่อใช้ในการหักตามรอบเวลาที่เปิดใช้งาน มีไว้เพื่อจำกัดการใช้ทรัพยากรของผู้ใช้งาน ไม่ให้ใช้ทรัพยากรของระบบเพียงผู้เดียวเป็นระยะเวลานาน

2.5 เทคโนโลยีระบบยืนยันตัวตน

2.5.1 Microsoft Active Directory

เป็นเครื่องมือที่มีมากับ Windows Server Operating System โดยทำหน้าที่ช่วยจัดการทรัพยากรในระบบ ในกรณีที่องค์กรที่มีผู้ใช้จำนวนมาก ๆ การนำ Active Directory มาใช้งานจะช่วยลดภาระค่าใช้จ่ายในการบริหารจัดการทรัพยากรของผู้ใช้ อีกทั้งยังเพิ่มความปลอดภัยให้กับระบบโดยรวม นอกจากจะช่วยจัดการทรัพยากรในระบบแล้ว Active Directory ทำหน้าที่จัดเก็บข้อมูลเกี่ยวกับ Object ต่างๆ เช่น ผู้ใช้, กลุ่ม, คอมพิวเตอร์ หรือนโยบายรักษาความปลอดภัย เป็นต้น

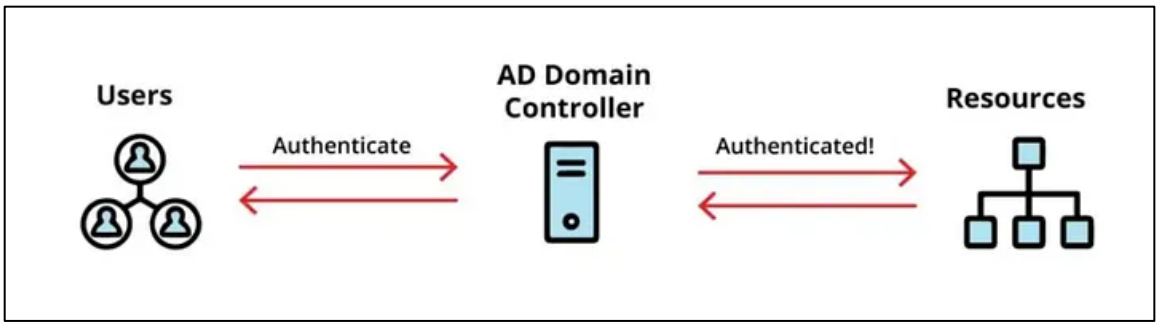
ส่วนประกอบของ Active Directory นั้น จะมีส่วนประกอบอยู่ 2 ส่วนด้วยกัน คือ

2.5.1.1 Active Directory Service

เป็นส่วนประกอบที่ทำหน้าที่ให้บริการแก่ ผู้ใช้งาน และผู้ดูแลระบบ

2.5.1.2 Active Directory Database

เป็นฐานข้อมูลสำหรับใช้ในการเก็บ Directory Object ต่างๆ เช่น บัญชีผู้ใช้, บัญชีกลุ่ม, โฟลเดอร์ที่ใช้ร่วมกัน เป็นต้น



รูปที่ 2.6 การนำ Active Directory มาใช้งาน

2.6 เทคโนโลยีการเข้าถึงเครือข่ายจากระยะไกล

2.6.1 Secure Shell (SSH)

เป็น Network Protocol ที่สามารถแลกเปลี่ยนข้อมูลโดยช่องทางที่ปลอยภัย (Secure Channel) ระหว่างอุปกรณ์เครือข่ายสองตัว ใช้ Linux หรือ Unix เป็นระบบปฏิบัติการพื้นฐานในการเข้าถึงบัญชีผู้ใช้ (Shell Accounts) ซึ่ง SSH ได้รับการออกแบบให้มาแทนการ Telnet, Rlogin, RSH (The remote shell) ด้วยเหตุผลทางด้านความปลอดภัย การส่งข้อมูลจะอยู่ในรูปแบบตัวอักษร (Plaintext) ที่มีการเข้ารหัสข้อมูล (Encryption) เพื่อให้ข้อมูลเป็นความลับและให้สามารถส่งข้อมูลผ่านเครือข่าย Internet ได้อย่างสมบูรณ์ โดย SSH มีหลักการทำงานดังนี้

2.6.1.1 การเข้ารหัสข้อมูล (Encryption)

SSH ใช้การเข้ารหัสเพื่อป้องกันการดักฟังข้อมูลระหว่างเครื่องไคลเอนต์ (Client) และเครื่องเซิร์ฟเวอร์ (Server) เมื่อมีการเชื่อมต่อผ่านเครือข่าย ซึ่งจะเริ่มต้นด้วยการแลกเปลี่ยนคีย์สาธารณะ (Public Key) เพื่อสร้างการเข้ารหัสแบบสมมาตรที่ใช้ปกป้องข้อมูล

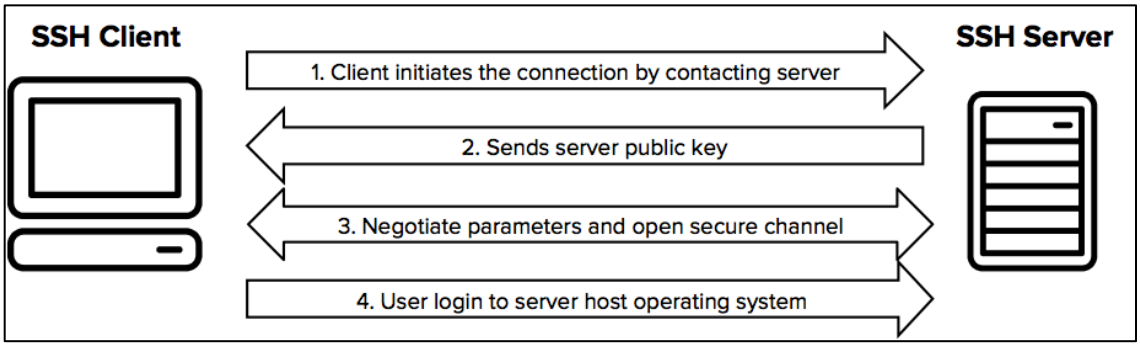
2.6.1.2 การรับรองความถูกต้อง (Authentication)

หลังจากสร้างการเชื่อมต่อที่ปลอดภัยแล้ว ไคลเอนต์ต้องยืนยันตัวตนกับเซิร์ฟเวอร์ การรับรองความถูกต้องนี้สามารถทำได้หลายวิธี

1) **Password-based authentication:** ไคลเอนต์ส่งรหัสผ่านไปยังเซิร์ฟเวอร์เพื่อยืนยันตัวตน
2) **Public key authentication:** ไคลเอนต์ใช้คู่คีย์สาธารณะและคีย์ส่วนตัวในการยืนยันตัวตน โดยคีย์สาธารณะถูกจัดเก็บไว้ในเซิร์ฟเวอร์ และคีย์ส่วนตัวถูกเก็บไว้ในเครื่องไคลเอนต์

2.6.1.3 การสร้างช่องทางการเชื่อมต่อที่ปลอดภัย (Secure Tunneling)

SSH สามารถสร้างการเชื่อมต่อที่ปลอดภัยระหว่างเครื่องไคลเอนต์และเซิร์ฟเวอร์ ซึ่งป้องกันไม่ให้ข้อมูลที่ส่งผ่านถูกดักฟังหรือแก้ไขได้



รูปที่ 2.7 อธิบายขั้นตอนการทำงานของ Secure Shell

2.6.2 Virtual Network Computing (VNC)

เป็นเทคโนโลยีที่ใช้สำหรับการเข้าถึงและควบคุมเครื่องคอมพิวเตอร์ระยะไกลผ่านเครือข่าย โดยแสดงหน้าจอของเครื่องปลายทาง (Remote) บนหน้าจอของเครื่องไคลเอนต์ (Client) ทำให้ผู้ใช้สามารถควบคุมการทำงานทั้งหมดได้เหมือนกับว่ากำลังใช้งานเครื่องนั้นโดยตรง โดยมีหลักการทำงานดังนี้

2.6.2.1 การแชร์หน้าจอ (Screen Sharing)

VNC จะแชร์หน้าจอของเครื่องเซิร์ฟเวอร์ (เครื่องปลายทาง) และส่งภาพของหน้าจอนั้นไปยังเครื่องไคลเอนต์ที่ทำการเชื่อมต่อ โดยแสดงผลเหมือนกับหน้าจอจริงบนเครื่องไคลเอนต์

2.6.2.2 การเชื่อมต่อแบบ Client-Server

VNC ประกอบด้วยสองส่วนคือ

1) **VNC Server** โปรแกรมที่ติดตั้งบนเครื่องปลายทางและทำหน้าที่แชร์หน้าจอ รวมถึงรับคำสั่งจากไคลเอนต์
2) **VNC Viewer (Client)** โปรแกรมที่ติดตั้งบนเครื่องผู้ใช้ ซึ่งใช้ในการเชื่อมต่อกับเครื่องปลายทางผ่านเซิร์ฟเวอร์ ไคลเอนต์ส่งคำสั่งควบคุมต่าง ๆ เช่น การคลิกเมาส์หรือการพิมพ์ผ่านเครือข่ายไปยังเซิร์ฟเวอร์ที่เครื่องปลายทาง

2.6.2.2 การส่งข้อมูลผ่านโปรโตคอล RFB (Remote Framebuffer Protocol)

VNC ใช้โปรโตคอลที่เรียกว่า RFB (Remote Framebuffer Protocol) ซึ่งทำหน้าที่ส่งข้อมูลภาพของหน้าจอและคำสั่งการควบคุมระหว่างเครื่องไคลเอนต์และเซิร์ฟเวอร์

2.6.2.3 การเข้ารหัสและความปลอดภัย

โดยพื้นฐานแล้ว VNC ไม่มีการเข้ารหัสข้อมูลที่ส่งไปมา ซึ่งหมายความว่าการใช้ VNC บนเครือข่ายที่ไม่ปลอดภัยอาจเสี่ยงต่อการถูกดักฟังข้อมูล เพื่อเพิ่มความปลอดภัย สามารถใช้ SSH หรือ VPN ควบคู่กับ VNC เพื่อเข้ารหัสข้อมูลและเพิ่มความปลอดภัยในการเชื่อมต่อ

2.6.3 Remote Desktop Protocol (RDP)

เป็นโปรโตคอลสำหรับการเข้าใช้งานเครื่องคอมพิวเตอร์เดส์กท็อป (Desktop Computer) จากระยะไกล โดย RDP ถูกพัฒนาโดยบริษัทไมโครซอฟต์ (Microsoft) และสามารถใช้งานร่วมกับคอมพิวเตอร์ที่มีระบบปฏิบัติการวินโดวส์ส่วนใหญ่ได้

โดยโปรโตคอลนี้ถูกออกแบบมาเพื่ออำนวยความสะดวกให้สามารถใช้งานเครื่องคอมพิวเตอร์จากระยะไกลได้ โดยการนำการแสดงผลทางหน้าจอของเครื่องคอมพิวเตอร์ระยะไกลมาแสดงผลที่เครื่องคอมพิวเตอร์ของผู้ใช้งาน และส่งอินพุตจากเครื่องคอมพิวเตอร์ของผู้ใช้งาน กลับไปยังเครื่องคอมพิวเตอร์จากระยะไกล โดยมีคุณสมบัติหลักๆดังนี้

2.6.3.1 Multi-Channel Communication

RDP รองรับช่องสื่อสารหลายช่องพร้อมกันสำหรับการถ่ายโอนข้อมูล ซึ่งช่วยให้สามารถส่งข้อมูลประเภทต่าง ๆ เช่น การเปลี่ยนเส้นทางอุปกรณ์ การเข้ารหัสลับและข้อมูลใบอนุญาตได้

2.6.3.2 การบีบอัดและการแคช

RDP บีบอัดข้อมูลเพื่อลดการใช้แบนด์วิดธ์ และใช้กลไกแคชเพื่อปรับปรุงประสิทธิภาพ

2.6.3.3 ความปลอดภัย

RDP รวมคุณสมบัติเช่นการเข้ารหัสและการรับรองระดับเครือข่าย (NLA) เพื่อเสริมความปลอดภัย

2.7 เทคโนโลยีเครือข่ายส่วนตัวเสมือน

2.7.1 Virtual Private Network (VPN)

คือเทคโนโลยีที่ใช้สร้างการเชื่อมต่อเครือข่ายส่วนตัวผ่านเครือข่ายสาธารณะ (เช่น อินเทอร์เน็ต) โดยมีการเข้ารหัสข้อมูล เพื่อให้มั่นใจว่าข้อมูลที่ส่งผ่านนั้นมีความปลอดภัยและเป็นส่วนตัว แม้จะเชื่อมต่อผ่านเครือข่ายที่ไม่ปลอดภัย โดยมีหลักการทำงานดังนี้

2.7.1.1 การเข้ารหัสข้อมูล (Encryption)

VPN ใช้การเข้ารหัสเพื่อปกป้องข้อมูลขณะเดินทางผ่านอินเทอร์เน็ต ข้อมูลที่ส่งจากอุปกรณ์ต้นทางจะถูกเข้ารหัสก่อนที่จะถูกส่งผ่านเครือข่าย เพื่อป้องกันไม่ให้บุคคลที่สามดักฟังหรือเข้าถึงข้อมูลได้

2.7.1.2 การสร้าง Tunneling (การสร้างช่องทางส่วนตัว)

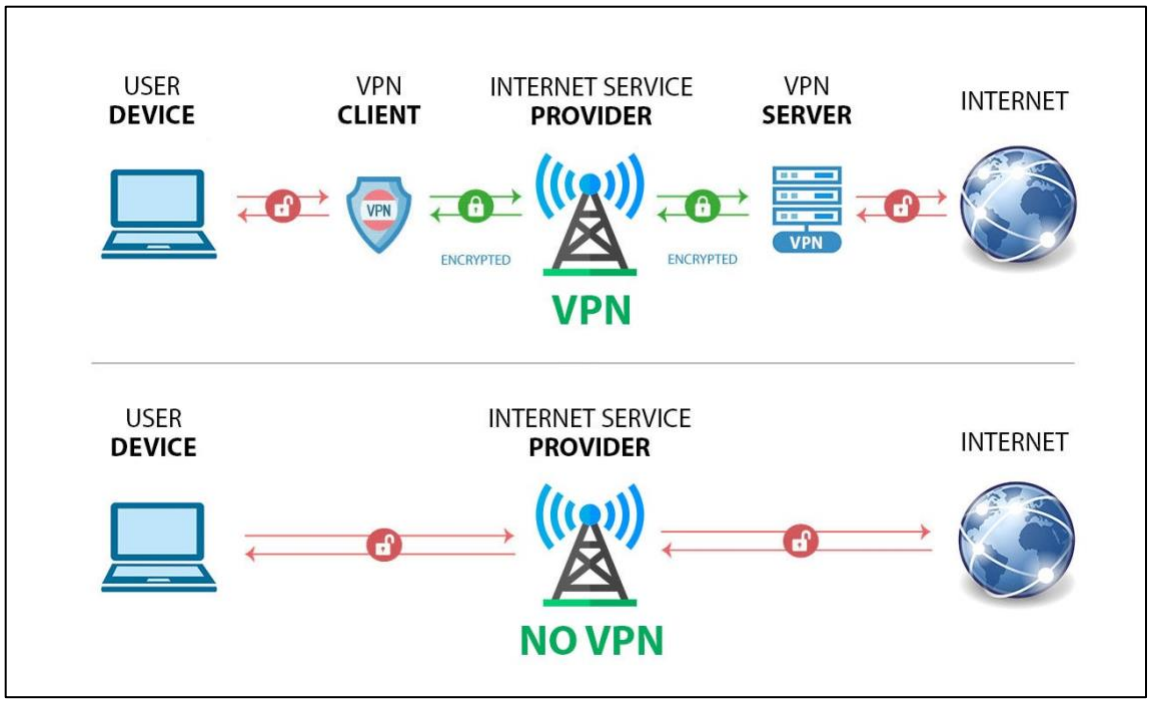
VPN จะสร้าง "อุโมงค์" ส่วนตัวที่ปลอดภัยสำหรับการส่งข้อมูล โดยข้อมูลจะถูกส่งผ่านช่องทางนี้ในรูปแบบที่เข้ารหัส ทำให้ไม่มีใครสามารถเห็นหรือดักจับข้อมูลขณะเคลื่อนที่ได้ ซึ่ง Protocols ที่ใช้กันบ่อยได้แก่ IPsec (Internet Protocol Security), OpenVPN, L2TP (Layer 2 Tunneling Protocol), WireGuard

2.7.1.3 การปกปิด IP Address

เมื่อใช้งาน VPN อุปกรณ์ของจะเชื่อมต่อกับเซิร์ฟเวอร์ของผู้ให้บริการ VPN จากนั้นเซิร์ฟเวอร์ VPN จะเชื่อมต่อไปยังเว็บไซต์หรือบริการอื่น ๆ แทนที่ IP Address ของคุณด้วย IP Address ของเซิร์ฟเวอร์ VPN ซึ่งช่วยปกปิดตัวตนและตำแหน่งที่แท้จริง

2.7.1.4 การยืนยันตัวตน

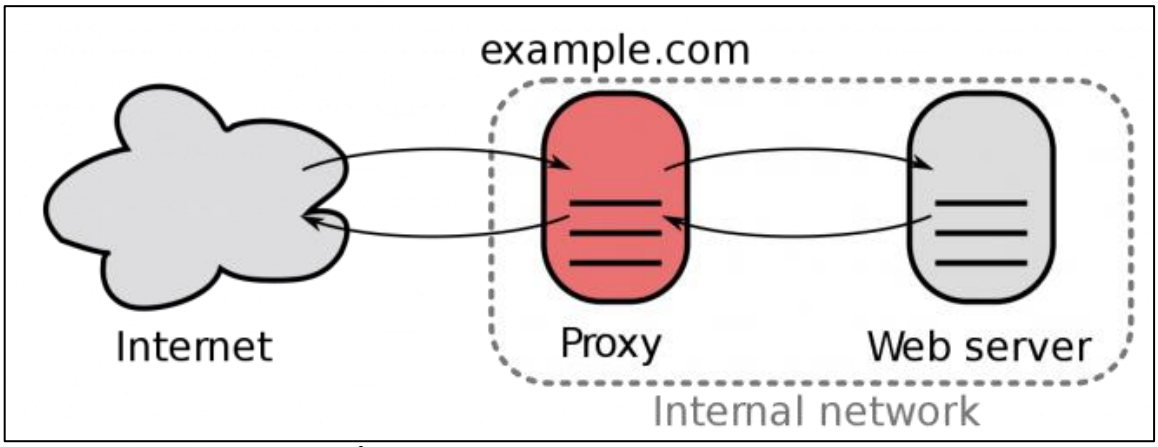
VPN ใช้การรับรองความถูกต้องเพื่อให้แน่ใจว่าผู้ใช้ที่กำลังเข้าถึงเครือข่ายเป็นผู้ใช้งานที่ได้รับอนุญาต การใช้วิธีการรับรองความถูกต้อง เช่น รหัสผ่าน, ใบรับรองดิจิทัล, หรือ การตรวจสอบหลายขั้นตอน (2FA) ช่วยป้องกันไม่ให้บุคคลที่ไม่ได้รับอนุญาตเข้าถึงระบบ



รูปที่ 2.8 เปรียบเทียบการใช้งาน VPN และไม่ใช้งาน VPN

2.8 Reverse Proxy

เซิร์ฟเวอร์ที่ทำหน้าที่เป็นตัวกลางระหว่างผู้ใช้ และเซิร์ฟเวอร์ที่ให้บริการแอพพลิเคชัน โดยที่ผู้ใช้จะส่งคำขอ มายัง Reverse Proxy แทนที่จะไปที่เซิร์ฟเวอร์โดยตรง จากนั้น Reverse Proxy จะส่งคำขอไปยังเซิร์ฟเวอร์ที่อยู่ภายใน Internal Network และนำผลลัพธ์กลับมาให้ผู้ใช้



รูปที่ 2.9 อธิบายการทำงานของ Reverse Proxy

2.9 Domain Name System (DNS)

ระบบที่มีไว้สำหรับบริหารจัดการข้อมูลของชื่อโดเมนเนม (Domain Name) และ ทำหน้าที่ในการแปลงชื่อโดเมนเนมดังกล่าวเป็นหมายเลขไอพีแอดเดรส (IP Address) เพื่อนำหมายเลขไอพีดังกล่าวไปติดต่อยัง Sever อื่น ๆ ที่ต้องการต่างๆ เช่น Sever Email Hosting , Server Web Hosting เป็นต้น

2.9.1 ส่วนประกอบของ DNS

2.9.1.1 Name Resolvers

มีหน้าที่หลักในการแปลงชื่อคอมพิวเตอร์ ให้เป็นหมายเลข IP เครื่องลูกข่ายที่ต้องการทราบหมายเลข IP หรือเรียกว่า Resolver เป็นซอฟท์แวร์ที่สร้างมากับเครื่องลูกข่าย แอปพลิเคชัน หรือไลบรารีที่อยู่ในเครื่องนั้นจะเป็นส่วนที่รับข้อมูลไว้

2.9.1.2 Domain Name Space

เป็นฐานข้อมูลระบบ DNS มีโครงสร้างเป็นต้นไม้ แต่ละ Domain Name Space จะมีชื่อเรียกและมีโดเมนย่อย หรือที่เรียกว่า Subdomain จะใช้จุด (.) เป็นเครื่องหมายแบ่งระหว่างโดเมนหลัก และโดเมนย่อย

2.9.1.3 Name Servers

เครื่องคอมพิวเตอร์แม่ข่าย ที่มีหน้าที่จัดการฐานข้อมูลในระบบ DNS โดยใช้โปรแกรมตอบกลับการร้องขอที่ได้รับมา ด้วยการค้นหาข้อมูลในฐานข้อมูลตัวเอง หรือสำหรับบางที่สามารถเขียนโปรแกรมให้สามารถค้นหาข้อมูลที่ฐานข้อมูลอื่นใน Name Server อื่นได้ ถ้าพบข้อมูลที่ได้รับร้องขอ จะเรียกว่า Authoritative ถ้าไม่พบข้อมูลจะเรียกว่า Non-Authoritative

2.9.2 ประเภท และวิธีการทำงาน

ในการทำงานปกติของ DNS Server เมื่อ User พิมพ์ชื่อเว็บไซต์ ส่งมา URL จะถูกนำส่งผ่าน Server ทั้งหมด 4 ชนิด เพื่อหา IP Address ซึ่ง Server เหล่านี้ทำงานร่วมกันเพื่อหาเลข IP Address ที่ถูกต้องให้เครื่อง Client โดยเซิร์ฟเวอร์ทั้ง 4 ตัวนี้ ประกอบด้วย

2.9.2.1 DNS Recursor

ทำหน้าที่รับคำร้องจาก DNS Client จากนั้นจึงสื่อสารไปยังตัวเซิรฟ์เวอร์ ให้หาเลข IP Address ที่ถูกต้อง เมื่อเจ้า DNS Recursor นี้ได้รับข้อมูลคำร้องขอจาก Client มันจะทำหน้าที่เสมือนเป็นเครื่อง Client เอง เพื่อกระจายคำสั่งไปยัง DNS เซิร์ฟเวอร์อีก 3 ชนิดที่เหลือ

2.9.2.2 Root Nameservers

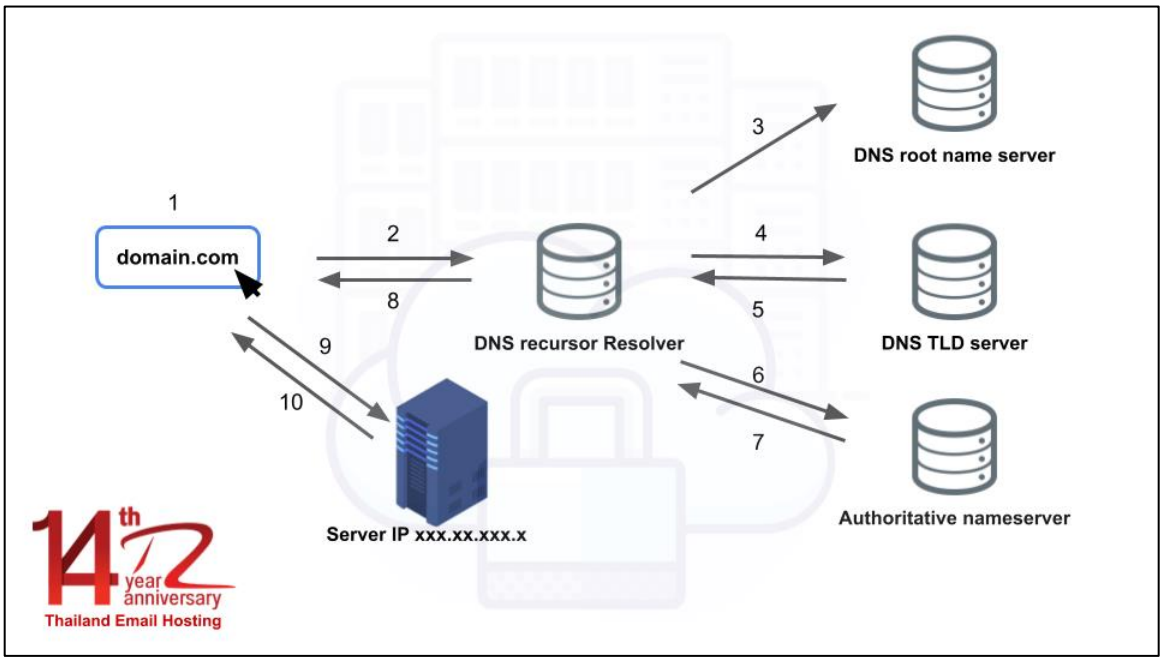
ออกแบบมาสำหรับใช้งานใน DNS Root Zone ของ Internet หน้าที่หลักคือตอบสนองต่อคำขอที่ถูกส่งมา และบันทึกลงใน Root Zone วิธีการตอบสนองคือการส่งกลับ List ของ Authoritative Nameserver พร้อม TLD ที่ถูกต้อง

2.9.2.3 TLD Nameservers

ทำหน้าที่เก็บเลข IP Address ของ Second-Level Domain และ TLD Name ทำหน้าที่ในการส่งผ่าน IP Address และส่งไปยัง Domain's Nameserver

2.9.2.4 Authoritative Nameservers

เป็นเซิร์ฟเวอร์ที่ทำหน้าที่ส่งคำตอบที่แท้จริงให้แก้คำร้องขอ Domain Name โดย Server ชนิดนี้แบ่งออกอีกเป็น 2 ประเภทย่อย ได้แก่ Master Server (Primary) และ Slave Server (Secondary) โดยตัว Master Server นั้นทำหน้าที่เก็บต้นฉบับของ Zone Record Slave Server ทำหน้าที่คัดลอกและ Backup ให้กับ Master Server เพื่อทำหน้าที่แทนในกรณีที่ตัว Master Server เกิดล่ม หรือใช้การไม่ได้ขึ้นมา



รูปที่ 2.10 อธิบายการทำงานของ Domain Name System

2.10 Remote Authentication Dial-In User Service (RADIUS)

โปรโตคอล RADIUS (Remote Authentication Dial-In User Service) เป็นโปรโตคอลที่ใช้ในระบบเครือข่ายสำหรับการตรวจสอบสิทธิ์ (Authentication) การกำหนดสิทธิ์การเข้าถึง (Authorization)

และการเก็บบันทึกข้อมูลการใช้งาน (Accounting) หรือที่เรียกรวมกันว่า กระบวนการ AAA (Authentication, Authorization, and Accounting) ซึ่งมีบทบาทสำคัญในการรักษาความปลอดภัยและบริหารจัดการการเข้าถึงเครือข่ายในองค์กร

RADIUS ทำงานบน สถาปัตยกรรมแบบไคลเอนต์-เซิร์ฟเวอร์ (Client-Server Architecture) โดยที่อุปกรณ์ต่าง ๆ ในเครือข่าย เช่น จุดเชื่อมต่อไร้สาย (Access Point), สวิตช์ (Switch), เซิร์ฟเวอร์เครือข่ายส่วนตัวเสมือน (VPN Server) หรืออุปกรณ์ไฟร์วอลล์ (Firewall) ทำหน้าที่เป็น RADIUS Client ซึ่งจะส่งคำขอตรวจสอบสิทธิ์ไปยัง RADIUS Server ที่ทำหน้าที่ประมวลผลข้อมูลรับรองตัวตนของผู้ใช้ โดยอ้างอิงจากฐานข้อมูลที่กำหนด เช่น Active Directory (AD), LDAP (Lightweight Directory Access Protocol) หรือฐานข้อมูล SQL

2.10.1 กระบวนการทำงานของ RADIUS

2.10.1.1 Authentication (การตรวจสอบสิทธิ์ผู้ใช้)

1) RADIUS Client ส่งข้อมูลรับรองตัวตนของผู้ใช้ (เช่น ชื่อผู้ใช้และรหัสผ่าน) ไปยัง RADIUS Server
2) RADIUS Server ตรวจสอบข้อมูลดังกล่าวกับฐานข้อมูล และตัดสินใจอนุมัติหรือปฏิเสธการเข้าถึง

2.10.1.2 Authorization (การกำหนดสิทธิ์การเข้าถึง)

1) หากการตรวจสอบสิทธิ์สำเร็จ RADIUS Server จะกำหนดสิทธิ์ของผู้ใช้ เช่น ระดับการเข้าถึงเครือข่าย การกำหนด VLAN หรือข้อจำกัดในการใช้งานทรัพยากรเครือข่าย

2.10.1.3 Accounting (การเก็บบันทึกการใช้งานเครือข่าย)

1) ระบบ RADIUS สามารถบันทึกข้อมูลการเข้าใช้งานของผู้ใช้ เช่น เวลาที่เริ่มและสิ้นสุดการเชื่อมต่อ ที่อยู่ IP ที่ได้รับ และปริมาณข้อมูลที่ใช้

2.11 Lightweight Directory Access Protocol (LDAP)

LDAP (Lightweight Directory Access Protocol) เป็นโปรโตคอลที่ใช้สำหรับการเข้าถึงและจัดการข้อมูลในระบบ Directory Service ซึ่งเป็นฐานข้อมูลที่มีโครงสร้างลำดับชั้น ใช้สำหรับเก็บข้อมูลเกี่ยวกับบัญชีผู้ใช้ อุปกรณ์ และทรัพยากรเครือข่าย เช่น รายชื่อผู้ใช้ กลุ่มผู้ใช้ หมายเลขโทรศัพท์ และสิทธิ์การเข้าถึงเครือข่าย LDAP ได้รับการออกแบบมาเพื่อให้สามารถค้นหาและดึงข้อมูลได้อย่างรวดเร็วและมีประสิทธิภาพ โดยเฉพาะในสภาพแวดล้อมขององค์กรที่มีผู้ใช้จำนวนมาก

2.11.1 คุณลักษณะของ LDAP

1) ใช้โครงสร้างแบบต้นไม้ (Tree Structure) ในการจัดเก็บข้อมูล
2) LDAP ใช้โปรโตคอล TCP/IP เป็นพื้นฐานในการรับส่งข้อมูล
3) รองรับการทำงานแบบกระจายศูนย์ (Distributed Directory Service) สามารถใช้งานร่วมกับเซิร์ฟเวอร์หลายเครื่องเพื่อกระจายภาระการทำงาน
4) รองรับการรับรองความถูกต้อง (Authentication) และการเข้ารหัสข้อมูลเพื่อเพิ่มความปลอดภัย

2.11.2 โครงสร้างของ LDAP

LDAP ใช้ โครงสร้างลำดับชั้น (Hierarchical Directory Structure) ซึ่งกำหนดตำแหน่งของข้อมูลแต่ละรายการในรูปแบบของ DN (Distinguished Name) โดยแต่ละโหนดภายใน LDAP Directory ถูกกำหนดด้วย Attribute (คุณลักษณะของข้อมูล) ที่ใช้ระบุตัวตน

ตัวอย่างโครงสร้างของ LDAP Directory:

```
dc=example,dc=com
│
├── ou=Users
│   ├── cn=John Doe
│   ├── cn=Jane Smith
│
├── ou=Groups
│   ├── cn=Admin
│   ├── cn=Developers
```

รูปที่ 2.11 ตัวอย่างโครงสร้างของ LDAP Directory

2.11.2.1 คำอธิบายโครงสร้าง

1) dc (Domain Component) ระบุโดเมนขององค์กร เช่น dc=example, dc=com
2) ou (Organizational Unit) ใช้ในการจัดกลุ่ม เช่น ou=Users สำหรับเก็บข้อมูลบัญชีผู้ใช้ และ ou=Groups สำหรับกลุ่มผู้ใช้
3) cn (Common Name) ใช้ระบุชื่อของแต่ละบัญชีผู้ใช้หรือกลุ่ม เช่น cn=John Doe หรือ cn=Admin

2.11.3 กระบวนการทำงานของ LDAP

2.11.3. การตรวจสอบสิทธิ์ (Authentication)

ผู้ใช้ต้องผ่านกระบวนการรับรองความถูกต้องก่อนเข้าถึงข้อมูล โดยสามารถใช้การตรวจสอบแบบ Simple Authentication (การใช้ชื่อผู้ใช้และรหัสผ่าน) หรือการเข้ารหัสผ่าน SASL (Simple Authentication and Security Layer)

2.11.3.2 การค้นหาข้อมูล (Querying & Searching)

LDAP ช่วยให้สามารถค้นหาข้อมูลจากฐานข้อมูลไดเรกทอรีได้อย่างรวดเร็ว โดยใช้ LDAP Query ซึ่งสามารถระบุเงื่อนไขในการค้นหาได้ เช่น การค้นหารายชื่อผู้ใช้ที่อยู่ในหน่วยงานเฉพาะ

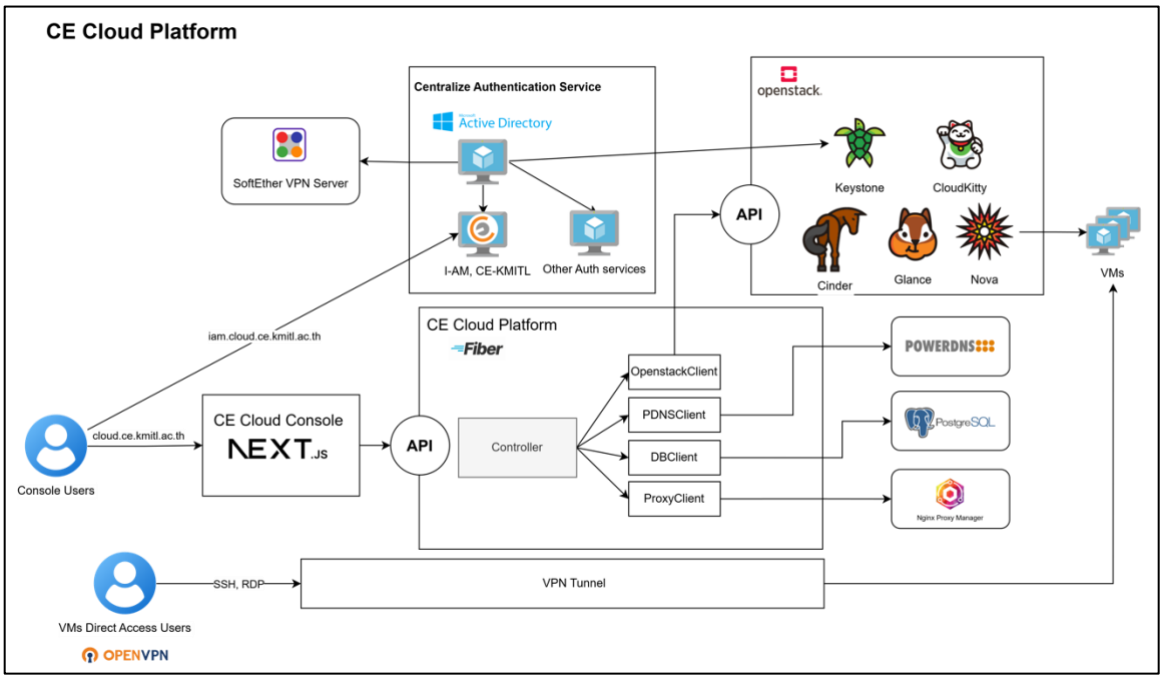
2.11.3.3 การบริหารจัดการบัญชีผู้ใช้

ผู้ดูแลระบบสามารถเพิ่ม แก้ไข หรือลบข้อมูลผู้ใช้และกลุ่มได้อย่างเป็นระบบ

บทที่ 3

การออกแบบและการพัฒนา

3.1 ภาพรวมของระบบ



รูปที่ 3.1 ภาพรวมของระบบ

3.1.1 CE Cloud Console

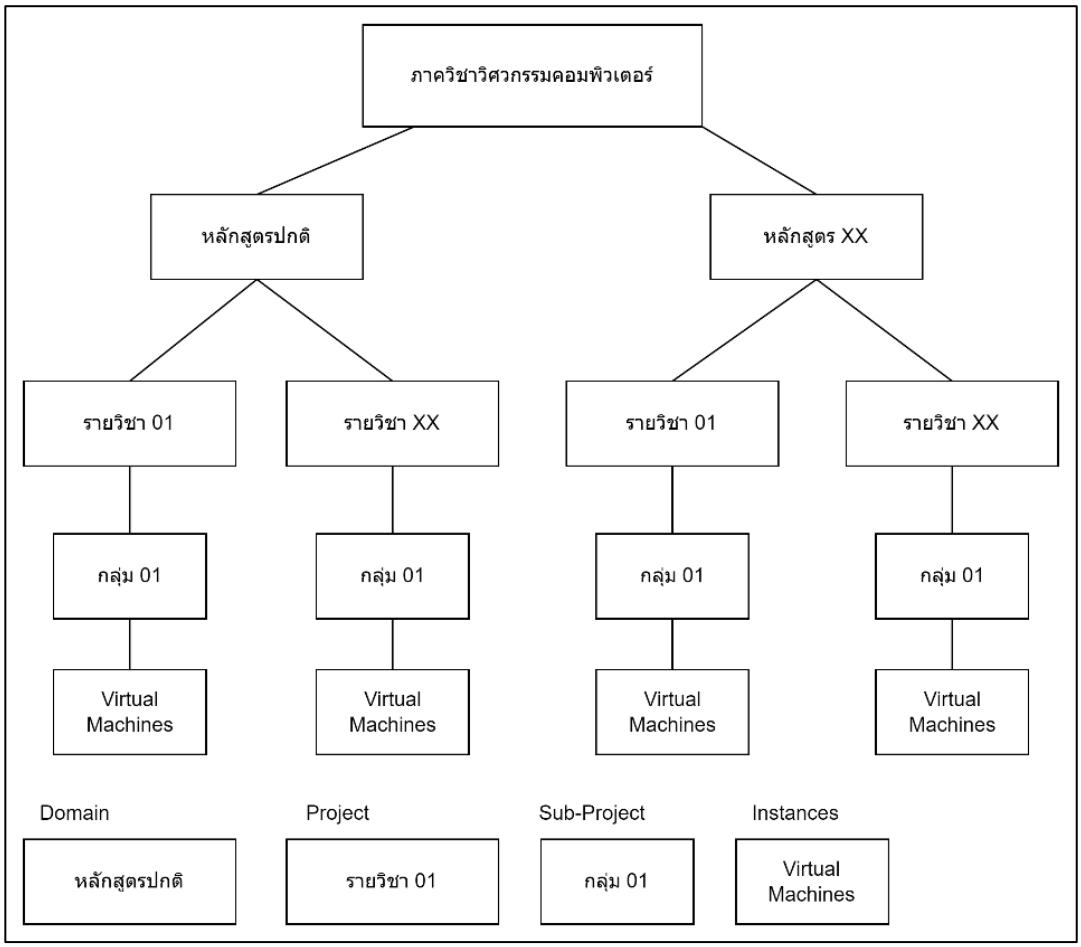
เป็น Web Application ที่พัฒนาด้วย Next.js Framework สำหรับให้ผู้ใช้สามารถเข้าถึงบริการต่างๆของ CE Cloud โดยพัฒนาขึ้นเพื่อลดความซับซ้อนในการบริหารจัดการ VM และทรัพยากรแทนระบบเดิม

3.1.2 CE Cloud Platform

เป็นระบบหลังบ้านสำหรับติดต่อกับ API ของระบบอื่นๆในภาควิชา โดยใช้ Go Fiber เป็น Framework สำหรับการพัฒนา และมีการใช้ Library และ SDK อื่นๆ สำหรับการใช้งานร่วมกับ OpenStack และ CE Infrastructure Software โดยแพลตฟอร์มได้มีการพัฒนาต่อจาก OpenStack โดยมีลำดับชั้นดังนี้

1) Domain หรือ หลักสูตร เช่น หลักสูตรปกติ, หลักสูตร International เป็นต้น

2) Project หรือ รายวิชา ซึ่งก็คือวิชาต่างๆ ที่เปิดสอนภายในภาควิชาวิศวกรรมคอมพิวเตอร์
3) Sub Project หรือ กลุ่มของนักศึกษาภายในรายวิชานั้นๆ
4) Instance หรือ เครื่องคอมพิวเตอร์เสมือนที่ถูกสร้างขึ้นมาภายใต้ Sub Project



รูปที่ 3.2 ลำดับชั้นของคลาวด์แพลตฟอร์ม

3.1.3 Identity and Access Management System (IAM)

เนื่องจากการยืนยันตัวตนต่างๆของระบบคลาวด์ จะทำการยืนยันตัวตนกับ Windows Active Directory ทำให้จำเป็นต้องมีบริการสำหรับการเปลี่ยนหรือกู้คืนรหัสผ่านสำหรับผู้ใช้งาน โดยที่ระบบนี้จะแยกออกมาเป็นอีกระบบหนึ่ง เพื่อรองรับกับในอนาคตที่อาจจะมีซอฟแวร์อื่นๆ ที่จะมายืนยันตัวผ่าน Windows Active Directory

3.1.4 Centralize Authentication Service

เป็นส่วนที่ใช้สำหรับเก็บข้อมูลผู้ใช้งาน ที่ใช้สำหรับการยืนยันตัวตน(Authentication) ซึ่งถูกจัดเก็บอยู่ใน Microsoft Active Directory โดยที่ภายในจะมีการแบ่ง Organization Unit (OU) เป็นหลักสูตรต่างๆในภาควิชา และมีการนำซอฟแวร์สำหรับการยืนยันตัวตนอื่นๆ (ในที่นี้คือ Keycloak) มาใช้เป็นระบบยืนยันตัวตนกลางของภาควิชา

3.1.5 OpenStack

เป็นระบบหลักที่ใช้สำหรับการสร้าง Private Cloud โดยมีหน้าที่จัดการโครงสร้างพื้นฐานของระบบคลาวด์ในรูปแบบของ Infrastructure as a Service (IaaS) ซึ่งช่วยให้สามารถบริหารจัดการทรัพยากรเซิร์ฟเวอร์ เครือข่าย และพื้นที่จัดเก็บข้อมูล ที่ใช้ในการสร้างเครื่องคอมพิวเตอร์เสมือน ได้อย่างมีประสิทธิภาพ

3.1.6 CE Infrastructure Software

ซอฟต์แวร์อื่นๆที่มีการติดตั้งและใช้งานอยู่แล้วในภาควิชา โดยจะมีการใช้งานดังนี้

3.1.5.1 PowerDNS

ใช้สำหรับการเพิ่ม DNS Record ให้กับเครื่องคอมพิวเตอร์เสมือน(Virtual Machine) ที่ถูกสร้างขึ้นจะอยู่ภายใต้ Subdomain *.cloud.ce.kmitl.ac.th โดยที่ Cloud Platform จะทำการเพิ่ม DNS Record ผ่าน API Server ของ PowerDNS

3.1.5.2 Nginx Proxy Manager

ใช้สำหรับการจัดการ Reverse Proxy โดยจะทำการจับคู่ Domain Name เข้ากับ Private IP และ Port ของเครื่องคอมพิวเตอร์เสมือน(Virtual Machine) เพื่อให้สามารถเข้าถึงบริการหรือแอพลิเคชันจากภายนอกได้ที่

3.1.5.3 SoftEther VPN Server

ใช้สำหรับให้ผู้ใช้งานระบบสามารถเข้าถึงเครื่องคอมพิวเตอร์เสมือน (Virtual Machine) ผ่าน Secure Shell (SSH) และ Remote Desktop Protocol (RDP) จากเครือข่ายภายนอกสถาบันได้

3.2 ประเภทของผู้ใช้งานในระบบ

ตารางที่ 3.1 ประเภทผู้ใช้งานในระบบ

ประเภทผู้ใช้งาน	รายละเอียด

ผู้ดูแลระดับหลักสูตร	1) สามารถสร้างหรือแก้ไขคอมพิวเตอร์เสมือนในหลักสูตรของตนเองได้ 2) สามารถบริหารจัดการทรัพยากรในหลักสูตรได้ 3) สามารถบริหารจัดการรายวิชาในหลักสูตรได้ 4) สามารถบริหารจัดการผู้ใช้ในหลักสูตรได้ 5) สามารถสร้างและบริหารจัดการกลุ่มของผู้ใช้งานในหลักสูตรได้
ผู้ดูแลระดับรายวิชา	1) สามารถสร้างหรือแก้ไขคอมพิวเตอร์เสมือนในรายวิชาของตนเองได้ 2) สามารถขอโควตาการใช้งานรายวิชาได้ผ่านระบบร้องขอโควตา 3) สามารถบริหารจัดการผู้ใช้ในรายวิชาที่ตนเองดูแลอยู่ได้ 4) สามารถสร้างกลุ่มย่อยของรายวิชาเพื่อแบ่งให้กับกลุ่มของผู้ใช้ในรายวิชา
ผู้ใช้งานทั่วไป	1) สามารถสร้างและใช้งานคอมพิวเตอร์เสมือนในระบบตามสิทธิ์ที่ได้รับมาได้ 2) สามารถขอสิทธิ์การใช้งานคอมพิวเตอร์เสมือนเพิ่มได้ผ่านระบบร้องขอเครดิตได้ 3) สามารถดาวน์โหลดไฟล์สำหรับเข้าถึงเครือข่ายส่วนตัวเสมือน (Virtual Private Network) ได้ เพื่อใช้สำหรับการเข้าถึงเครือข่ายของคอมพิวเตอร์เสมือนของตัวเองทั้งหมดในระบบ 4) สามารถบันทึก SSH Public Key ของผู้ใช้งาน เพื่อใช้สำหรับการเข้าใช้งานคอมพิวเตอร์เสมือนของตัวเองทั้งหมดในระบบ

3.3 System Requirements Specification

ตารางที่ 3.2 System Requirements Specification

ลำดับที่	Requirements	Priority	Type
ส่วนการใช้งานสำหรับผู้ใช้งานทุกประเภท			

R01	นักศึกษาและบุคลากรภาควิชาวิศวกรรมคอมพิวเตอร์ทุกคนสามารถยืนยันตัวตนเพื่อใช้งานระบบได้	Must Have	- Authentication - Functional
R02	ผู้ใช้งานทุกคนที่มีสิทธิ์สามารถสร้างคอมพิวเตอร์เสมือนได้	Must Have	- Instance System - Functional
R03	ผู้ใช้งานทุกคนสามารถจัดการคอมพิวเตอร์เสมือนจริงที่ตนเองมีสิทธิ์บริหารจัดการได้	Must Have	- Instance System - Functional
R04	ผู้ใช้งานทุกคนสามารถเข้าถึงคอมพิวเตอร์เสมือนได้จากเครือข่ายทั้งภายในและภายนอกสถาบัน	Must Have	- Instance System - Functional
R05	ผู้ใช้งานทุกคนสามารถดาวน์โหลดไฟล์สำหรับการเข้าใช้งาน VPN ผ่านระบบได้	Must Have	- Access Management System - Functional
R06	ผู้ใช้งานทุกคนสามารถใช้งานคอมพิวเตอร์เสมือนผ่าน SSH ได้	Must Have	- Instance System - Functional
R07	ผู้ใช้งานทุกคนสามารถใช้งานคอมพิวเตอร์เสมือนผ่าน VNC ได้	Must Have	- Instance System - Functional
R08	ผู้ใช้งานทุกคนสามารถใช้งานคอมพิวเตอร์เสมือนที่มีสิทธิ์ผ่าน RDP ได้	Must Have	- Instance System - Functional
R09	ผู้ใช้งานทุกคนสามารถเพิ่ม SSH Public Key เข้าสู่ระบบเพื่อใช้ในการเชื่อมต่อกับคอมพิวเตอร์เสมือนผ่าน SSH ได้	Must Have	- Access Management System - Functional

R10	ผู้ใช้งานทุกคนสามารถร้องขอเครดิตเพื่อใช้สำหรับการสร้างคอมพิวเตอร์เสมือนจริงในระบบเครดิต	Must Have	- Resource System - Functional
R11	ระบบสามารถเพิ่ม DNS Record และทำ Reverse Proxy ให้แก่คอมพิวเตอร์เสมือนได้โดยไม่ต้องผ่านผู้ดูแลระบบ	Must Have	- Instance System - Functional
R12	ผู้ใช้งานทุกคนสามารถแก้ไข SSH Public Key ที่ใช้งานอยู่ในคอมพิวเตอร์เสมือนได้	Should Have	- Instance System - Functional
R13	รองรับการใช้งานระบบโควตา ทำให้ผู้ใช้งานสามารถสร้างคอมพิวเตอร์เสมือนได้ตามสิทธิ์ที่ได้รับมาเท่านั้น	Should Have	- Instance System - Functional
R14	รองรับการใช้งานระบบเครดิต ทำให้ผู้ใช้งานสามารถสร้างคอมพิวเตอร์เสมือนได้ตามต้องการขึ้นอยู่กับเครดิตคงเหลือ	Should Have	- Instance System - Functional
R15	รองรับการร้องขอเครดิตของผู้ใช้งาน	Should Have	- Instance System - Functional
ส่วนการใช้งานสำหรับผู้ดูแลรายวิชา			
R16	ผู้ดูแลรายวิชาสามารถบริหารจัดการผู้ใช้งานในขอบเขตการดูแลของตัวเองได้	Must Have	- Subject Management System - Functional
R17	ผู้ดูแลรายวิชาสามารถสร้างกลุ่มย่อยภายในรายวิชาที่ตนดูแลได้	Must Have	- Subject Management System - Functional

R18	ผู้ดูแลรายวิชาสามารถจัดการคอมพิวเตอร์เสมือนภายในวิชาตนเองได้	Must Have	- Subject Management System - Functional
R19	ผู้ดูแลรายวิชาสามารถร้องขอโควตาการใช้งานรายวิชาได้	Should Have	- Subject Management System - Functional
ส่วนการใช้งานสำหรับผู้ดูแลหลักสูตร			
R20	ผู้ดูแลหลักสูตรสามารถสร้างรายวิชาให้แก่ผู้ดูแลรายวิชาได้	Must Have	- Subject Management System - Functional
R21	ผู้ดูแลสามารถกำหนดการใช้ทรัพยากรของผู้ใช้งานหรือกลุ่มผู้ใช้งานในขอบเขตการดูแลของตัวเองได้	Must Have	- Subject Management System - Functional
R22	ผู้ดูแลสามารถดูแลจัดการคอมพิวเตอร์เสมือนของผู้ใช้งานในขอบเขตการดูแลของตัวเองได้	Must Have	- Instance System - Functional
R23	ผู้ดูแลหลักสูตรสามารถจัดการระบบเครือข่ายเสมือนของระบบได้	Must Have	- Infrastructure - Functional
R24	ผู้ดูแลหลักสูตรสามารถจัดการระบบเครดิตให้แก่ผู้ใช้งานได้	Should Have	- Subject Management System - Functional
ส่วนที่เกี่ยวกับระบบ Cloud			
R25	สามารถรองรับการเพิ่มจำนวนเครื่องแม่ข่าย	Should Have	- Infrastructure - Non Functional

R26	ระบบสามารถรองรับการเชื่อมต่อกับบริการอื่นๆของภาควิชาในอนาคต	Should Have	- Infrastructure - Non Functional
R27	ระบบโดยรวมมีความปลอดภัย	Should Have	- Infrastructure - Non Functional
R28	ระบบมีเสถียรภาพ	Should Have	- Infrastructure - Non Functional

3.4 Centralized Authentication System

3.4.1 Microsoft Active Directory (Parent Domain)

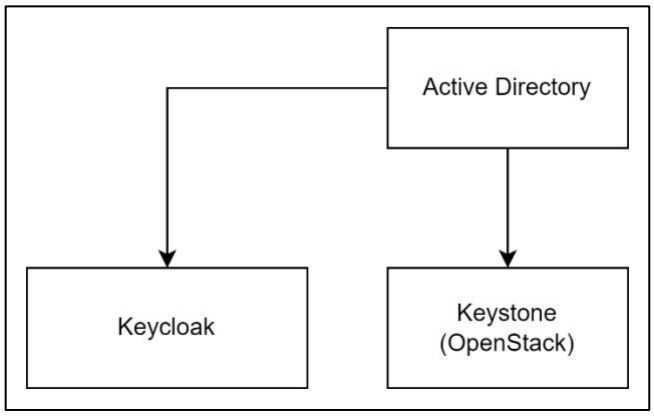
ทำหน้าที่ในการจัดเก็บรายชื่อผู้ใช้งานทั้งหมด โดยภายใน Microsoft Active Directory จะมีการแบ่งรายชื่อผู้ใช้งานในแต่ละหลักสูตรของภาควิชาโดยใช้ Organization Unit (OU)

3.4.2 Keystone Service

เป็นบริการของ OpenStack ที่ใช้สำหรับการยืนยันตัวตน (Authentication) โดยจะทำการร้องขอชื่อผู้ใช้งานแต่ละหลักสูตรมาจาก Microsoft Active Directory

3.4.3 Keycloack

เป็นระบบยืนยันตัวตนที่เชื่อมต่อกับ Windows Active Directory Parent Domain เพื่อเป็นระบบยืนยันตัวตนกลางสำหรับซอฟแวร์ภายในภาควิชาที่จะพัฒนาขึ้นมาในอนาคต



รูปที่ 3.3 การเชื่อมต่อของระบบยืนยันตัวตน

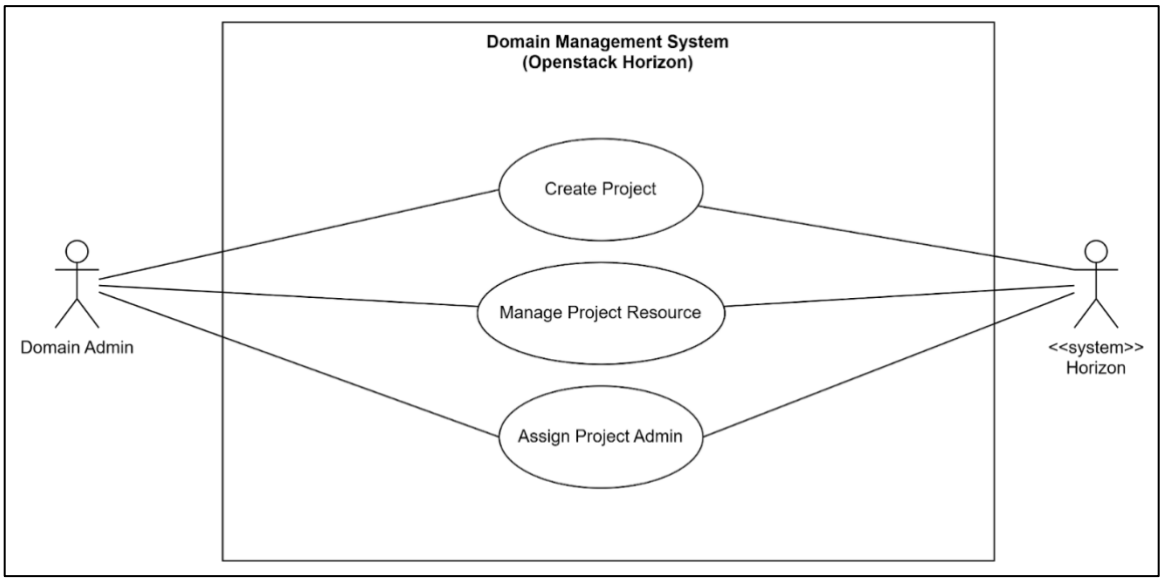
3.5 ระบบเครือข่ายของเครื่องคอมพิวเตอร์เสมือน (Virtual Machine) ภายใน OpenStack

ใน OpenStack ระบบเครือข่ายเสมือนถูกออกแบบให้แยกออกจากเครือข่ายภายนอก โดยมีการจัดการผ่าน Virtual Routers ภายในสภาพแวดล้อมเสมือน ซึ่งส่งผลให้การเข้าถึงจากภายนอกไปยังเครื่องคอมพิวเตอร์เสมือน (VM) จำเป็นต้องมีการกำหนดค่าพิเศษ สำหรับการเข้าถึง VM ผ่าน Secure Shell (SSH) หรือ Remote Desktop Protocol (RDP) ผู้ใช้งานต้องเพิ่ม Network Interface ให้กับ VM เพื่อให้สามารถเชื่อมโยงกับ IP Address ที่ได้จาก Physical Router ของเครือข่ายภายนอก

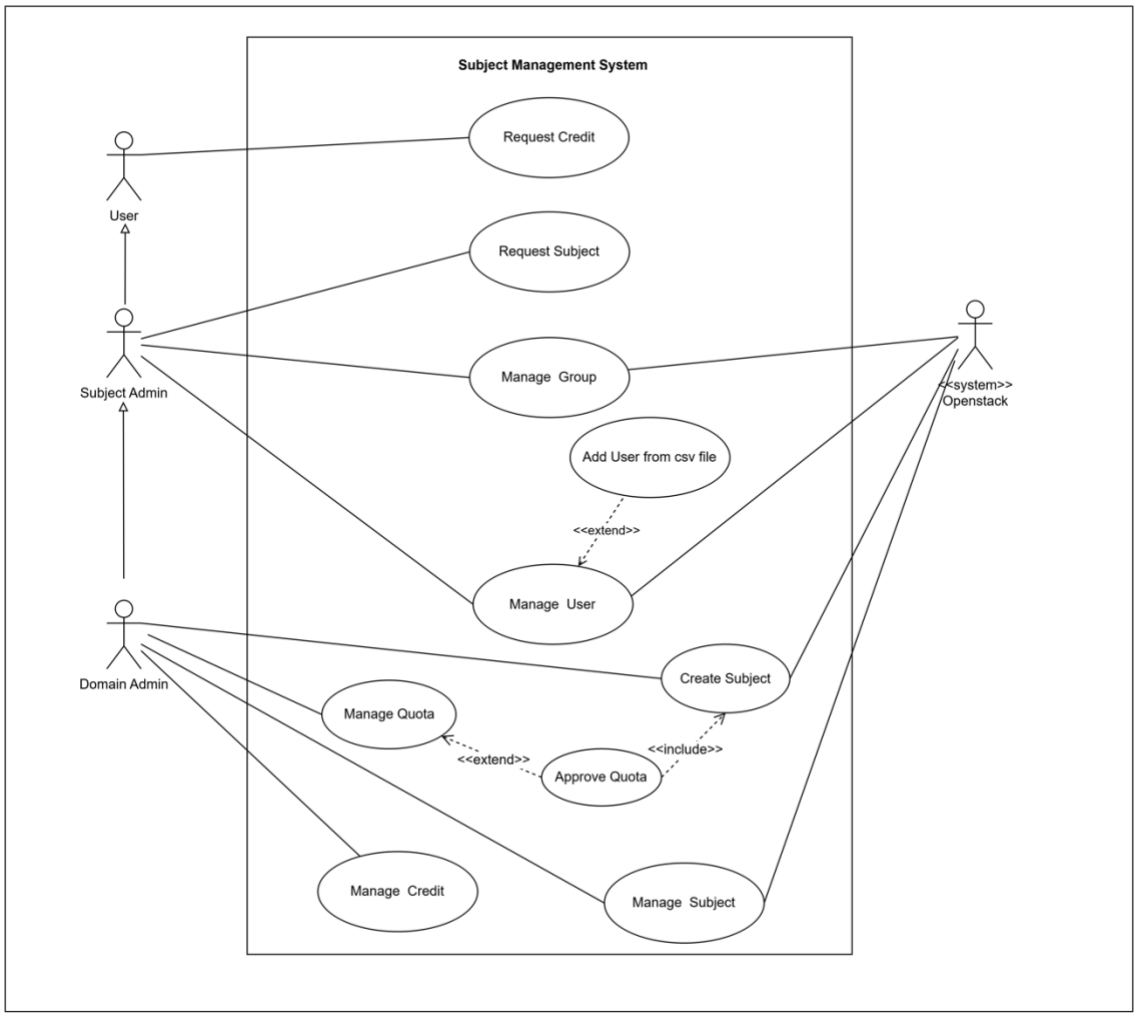
นอกจากนี้ NGINX Proxy Manager จะทำ Reverse Proxy เพื่อเข้าใช้งาน VM ผ่าน DNS Records ซึ่งถูกสร้างและจัดการผ่าน PowerDNS ทั้งนี้ เครื่อง VM จะสามารถเข้าถึงได้ผ่าน VPN ของทางภาควิชา เพื่อเพิ่มความปลอดภัยในการเข้าถึงผ่านเครือข่ายภายนอก จึงทำให้ VM เหล่านี้สามารถถูกเข้าถึงได้จากภายนอกอย่างปลอดภัยผ่านระบบเครือข่ายหลัก

3.6 Software Diagram

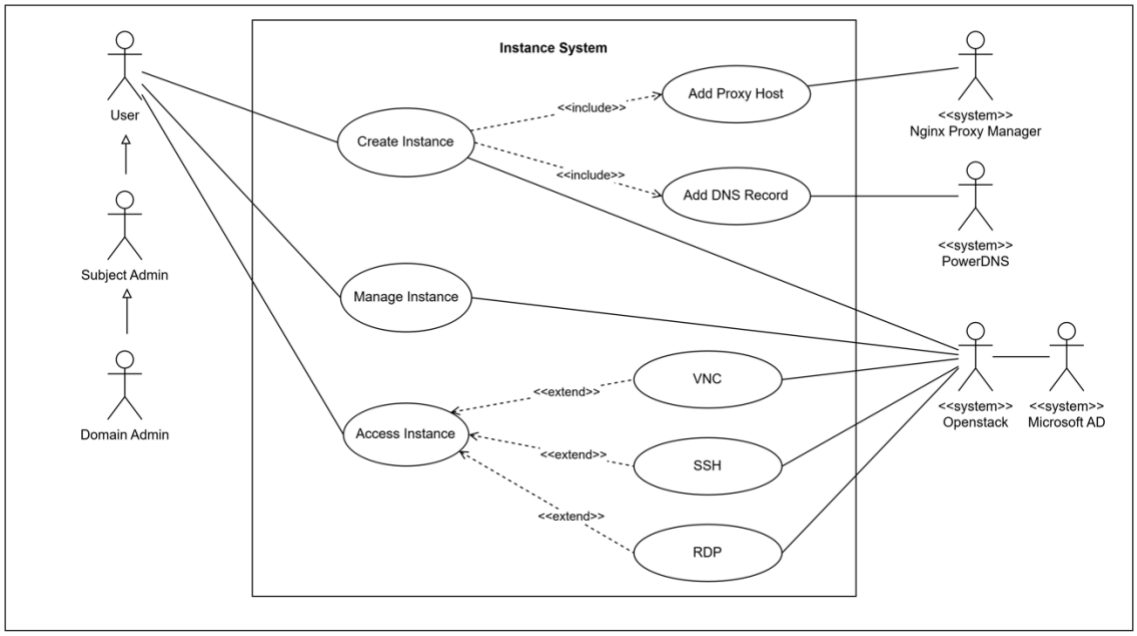
3.6.1 Use Case Diagram



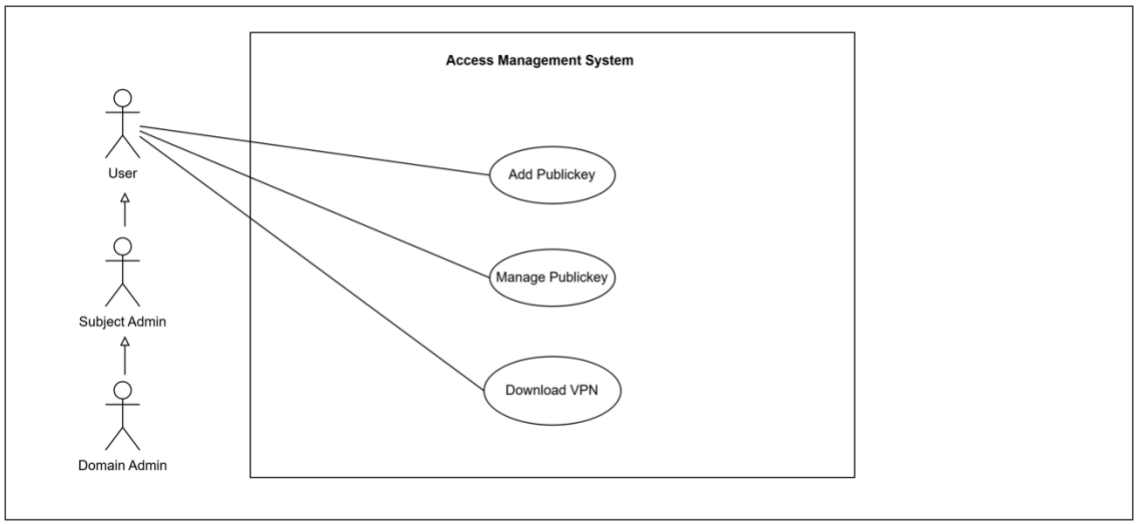
รูปที่ 3.4 Use Case Diagram ของการจัดการ Domain



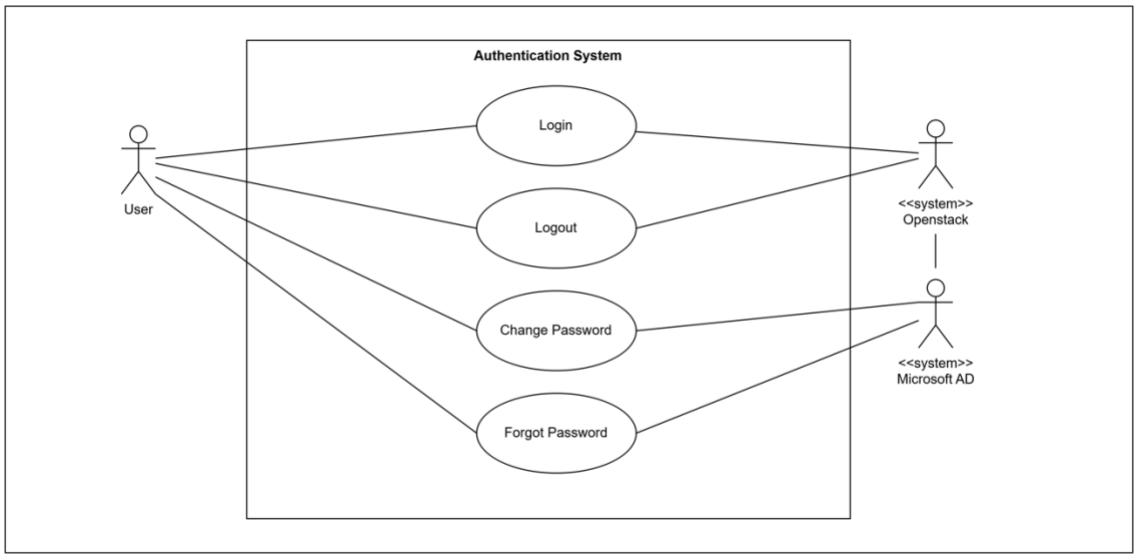
รูปที่ 3.5 Use Case Diagram ของการจัดการรายวิชา



รูปที่ 3.6 Use Case Diagram ของการจัดการ Instance

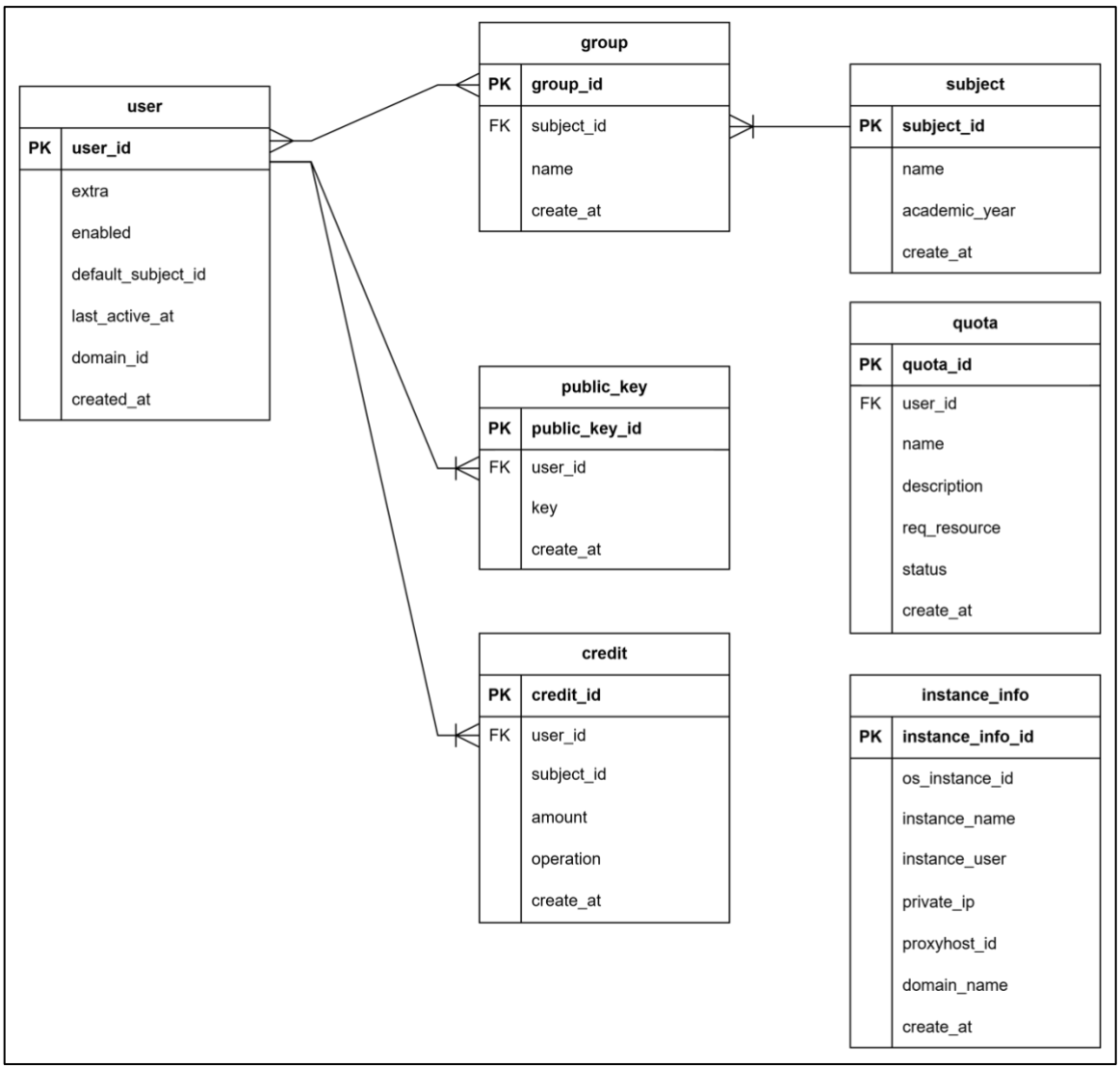


รูปที่ 3.7 Use Case Diagram ของการจัดการการเข้าถึง



รูปที่ 3.8 Use Case Diagram ของการยืนยันตัวตน

3.6.2 การออกแบบฐานข้อมูล



รูปที่ 3.9 การออกแบบฐานข้อมูล

การออกแบบฐานข้อมูลและความสัมพันธ์ของข้อมูล มีรายละเอียดดังต่อไปนี้

ตารางที่ 3.3 รายละเอียดฐานข้อมูลตาราง user

คีย์	ชื่อแอตทริบิวต์	ชนิดตัวแปร	รายละเอียด
PK	id	uuid	ไอดีของผู้ใช้งาน
	extra	text	รายละเอียดเพิ่มเติมของผู้ใช้งาน

	enabled	tinyint	สถานะของผู้ใช้งาน
FK	default_project_id	uuid	โปรเจกต์หลัก
	create_at	datetime	วันที่สร้าง
	last_active_at	datetime	วันที่ใช้งานล่าสุด
FK	domain_id	uuid	ไอดีของโดเมน

ตารางที่ 3.4 รายละเอียดฐานข้อมูลตาราง subject

คีย์	**ชื่อแอตทริบิวต์**	**ชนิดตัวแปร**	**รายละเอียด**
PK	subject_id	uuid	ไอดีของวิชา
	name	varchar(50)	ชื่อของวิชา
	create_at	datetime	วันที่สร้าง
	academic_year	int	ปีการศึกษา

ตารางที่ 3.5 รายละเอียดฐานข้อมูลตาราง group

คีย์	**ชื่อแอตทริบิวต์**	**ชนิดตัวแปร**	**รายละเอียด**
PK	group_id	uuid	ไอดีของกลุ่มผู้ใช้
FK	subject_id	uuid	ไอดีของวิชาที่กลุ่มอยู่
	create_at	datetime	เวลาที่สร้าง
	name	varchar(50)	ชื่อของกลุ่ม

ตารางที่ 3.6 รายละเอียดฐานข้อมูลตาราง credit

คีย์	**ชื่อแอตทริบิวต์**	**ชนิดตัวแปร**	**รายละเอียด**
PK	credit_id	uuid	ไอดีของเครดิตของผู้ใช้
FK	user_id	uuid	ไอดีของผู้ใช้

FK	subject_id	uuid	ไอดีของวิชาที่ให้เครดิต
	create_at	datetime	เวลาที่สร้าง
	amount	int	จำนวนเครดิตที่ได้/เสีย
	operation	varchar(20)	ตัวดำเนินการเพื่อบอกว่าได้รับหรือใช้ credit

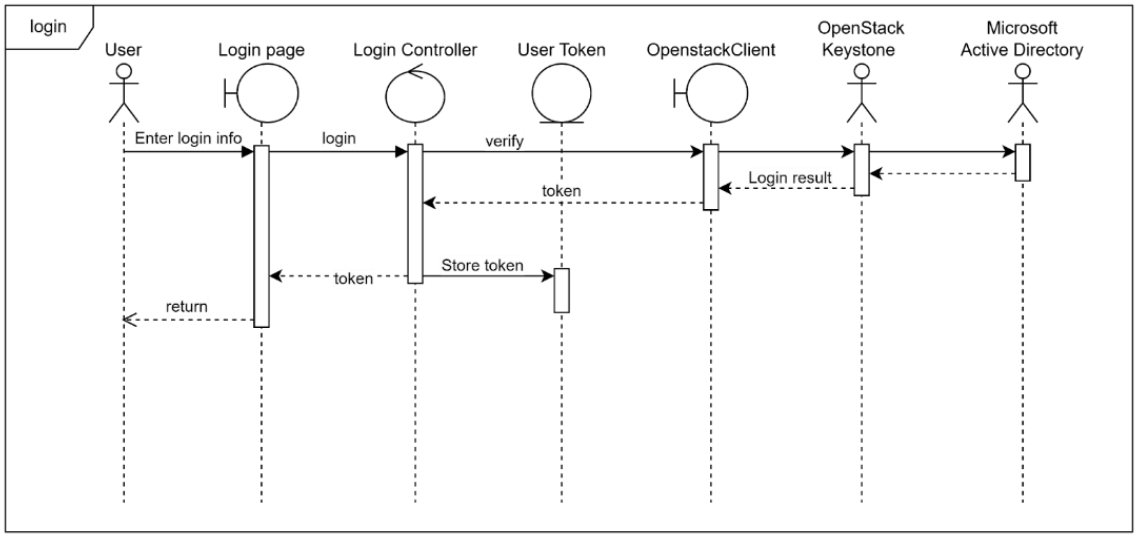
ตารางที่ 3.7 รายละเอียดฐานข้อมูลตาราง public_key

คีย์	**ชื่อแอตทริบิวต์**	**ชนิดตัวแปร**	**รายละเอียด**
PK	public_key_id	uuid	ไอดีของคีย์ของผู้ใช้
FK	user_id	uuid	ไอดีของผู้ใช้
	create_at	datetime	เวลาที่สร้าง
	key	text	คีย์ของผู้ใช้

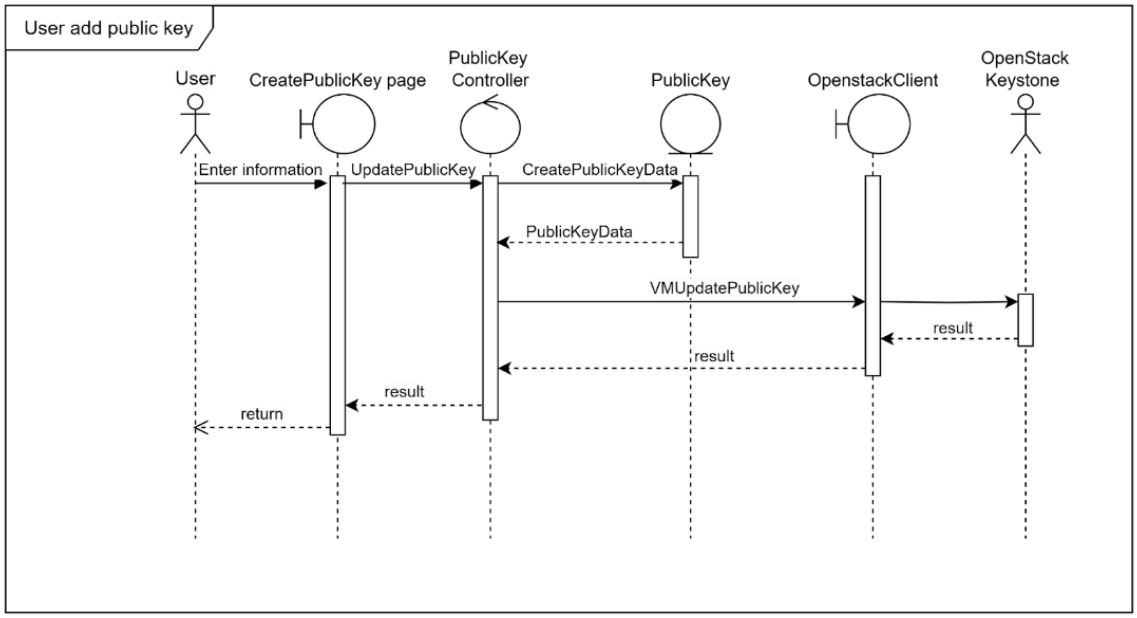
ตารางที่ 3.8 รายละเอียดฐานข้อมูลตาราง instance_info

คีย์	**ชื่อแอตทริบิวต์**	**ชนิดตัวแปร**	**รายละเอียด**
PK	instance_info_id	uuid	ไอดีของคอมพิวเตอร์เสมือน
	os_instance_id	uuid	ไอดีของระบบปฏิบัติการ
	domain_name	uuid	ชื่อโดเมนของคอมพิวเตอร์เสมือน
	subject_id	uuid	ไอดีวิชาของคอมพิวเตอร์เสมือน
	create_at	datetime	เวลาที่สร้าง
	proxyhost_id	bigint	ไอดีของ Proxy
	instance_user	varchar(120)	ไอดีผู้ใช้ที่สร้างคอมพิวเตอร์เสมือน
	instance_id	varchar	ไอดีของคอมพิวเตอร์เสมือน

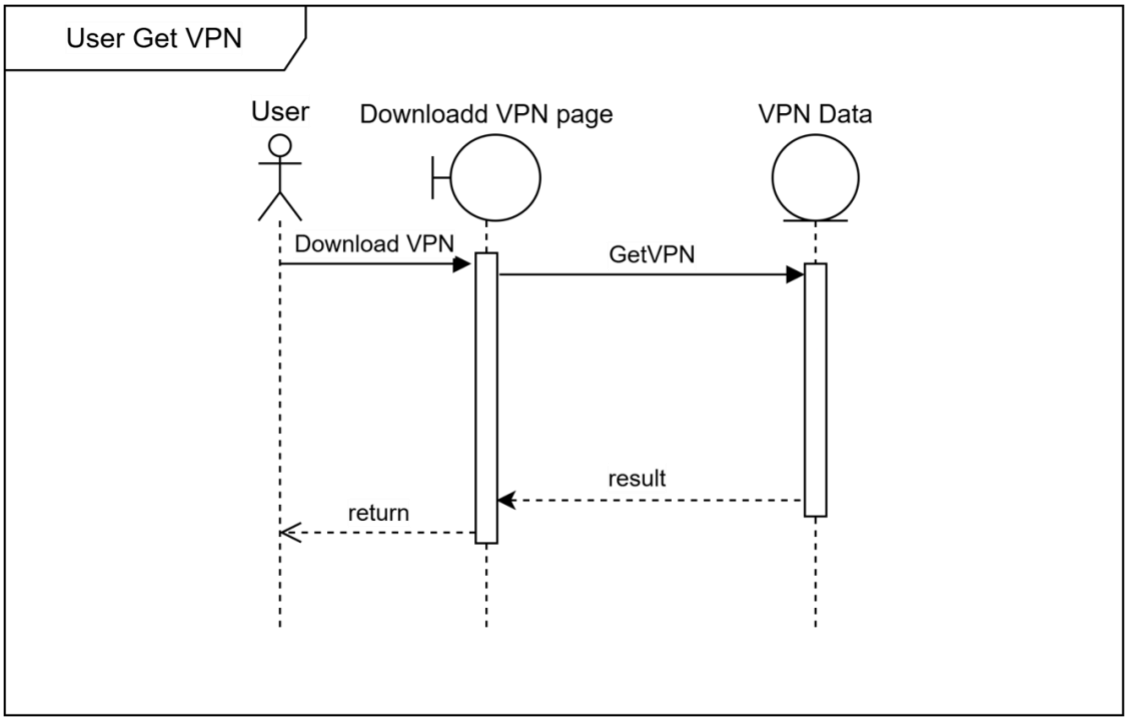
3.6.3 Sequence Diagram



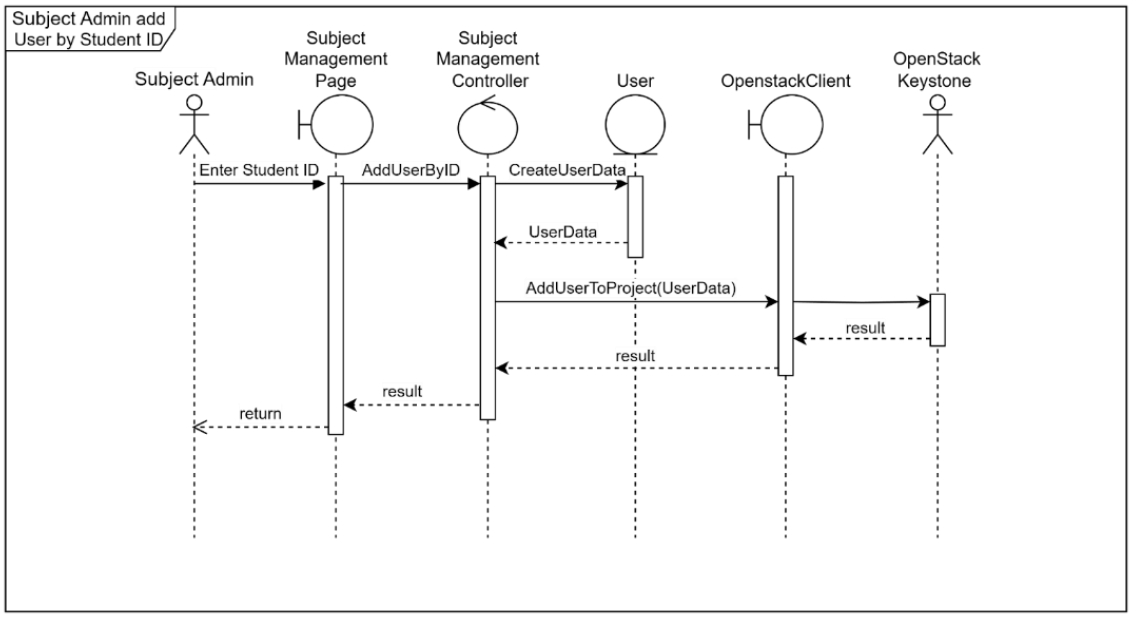
รูปที่ 3.10 Sequence Diagram ของการเข้าสู่ระบบ



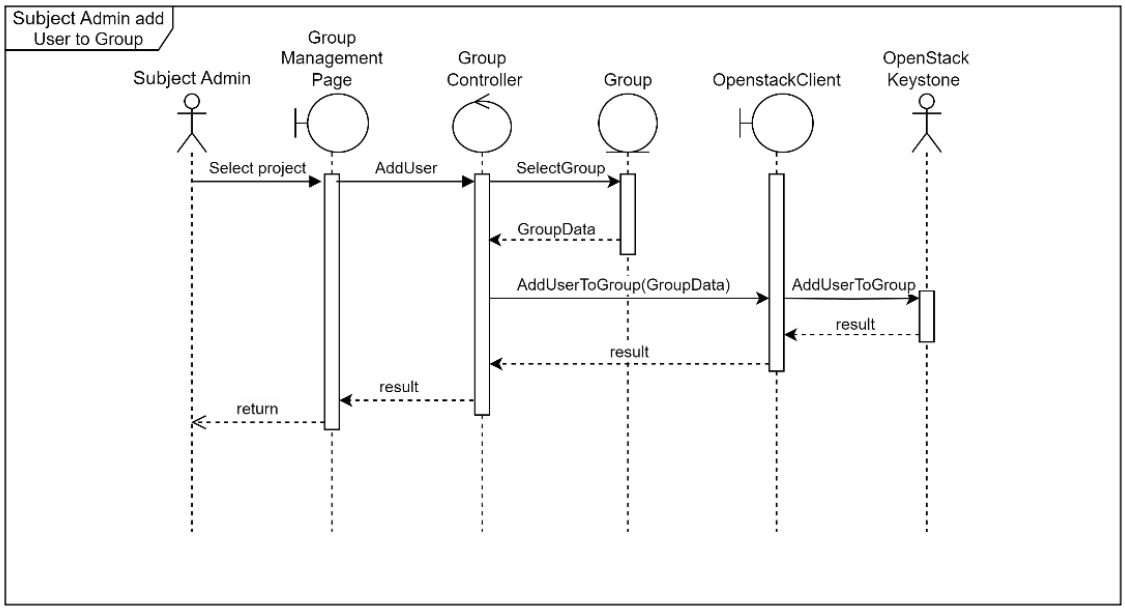
รูปที่ 3.11 Sequence Diagram ของการเพิ่ม Public Key



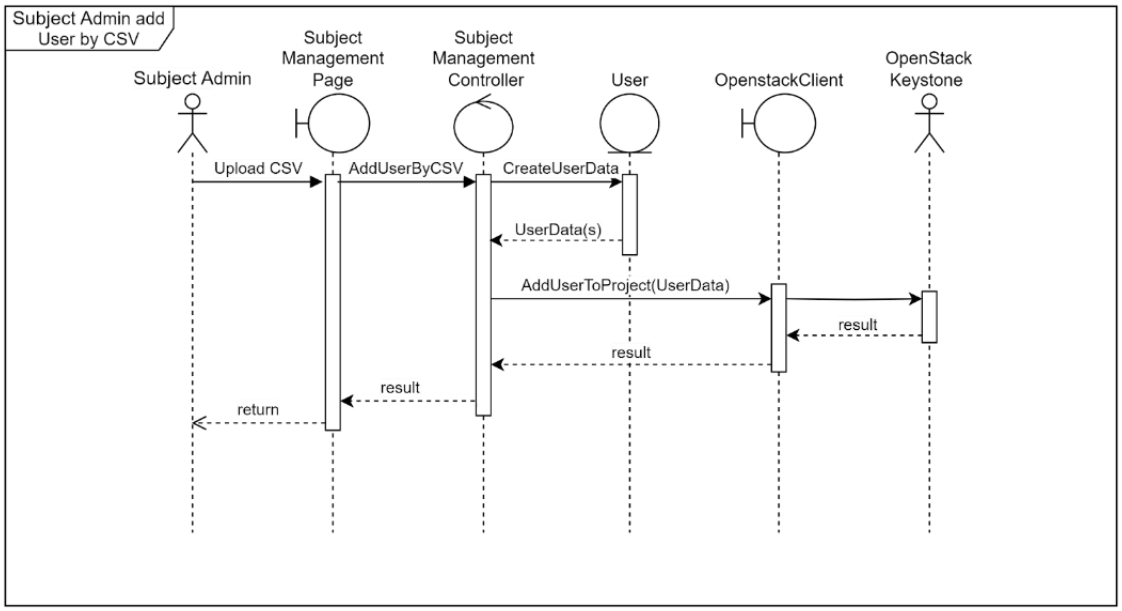
รูปที่ 3.12 Sequence Diagram ของการดาวน์โหลด VPN ของผู้ใช้งาน



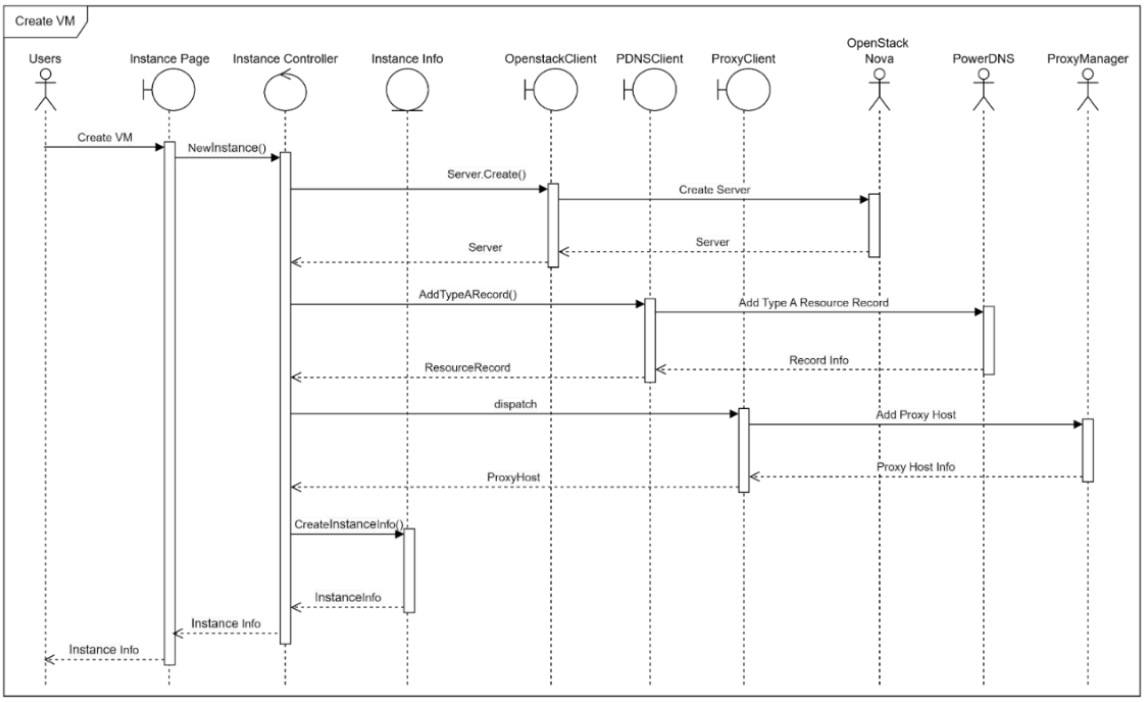
รูปที่ 3.13 Sequence Diagram ของการเพิ่มผู้ใช้งานด้วยรหัสนักศึกษา



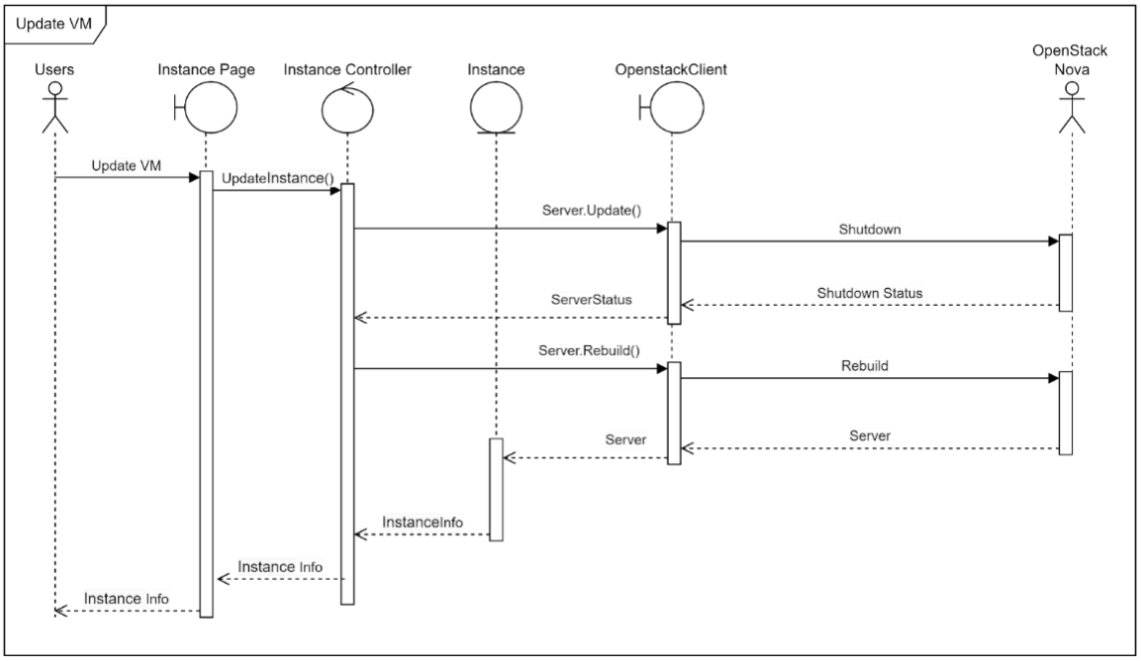
รูปที่ 3.14 Sequence Diagram ของการเพิ่มนักศึกษาเข้าในกลุ่มย่อยของรายวิชา



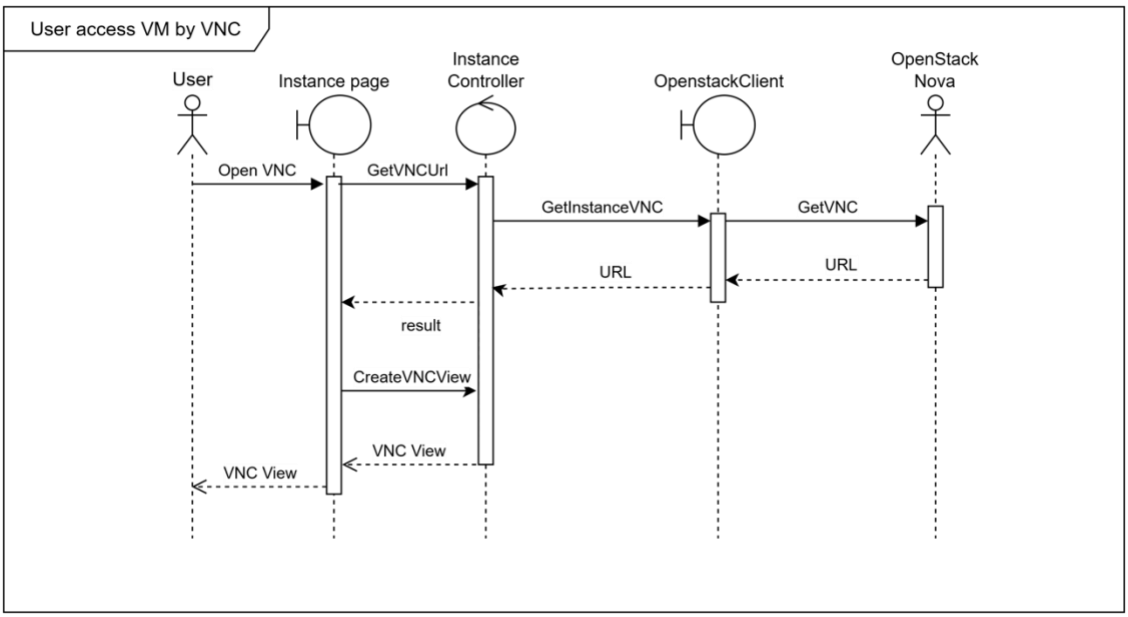
รูปที่ 3.15 Sequence Diagram ของการเพิ่มกลุ่มของนักศึกษาเข้าในรายวิชาด้วยไฟล์ CSV



รูปที่ 3.16 Sequence Diagram ของการสร้าง Instance



รูปที่ 3.17 Sequence Diagram ของการแก้ไข Instance

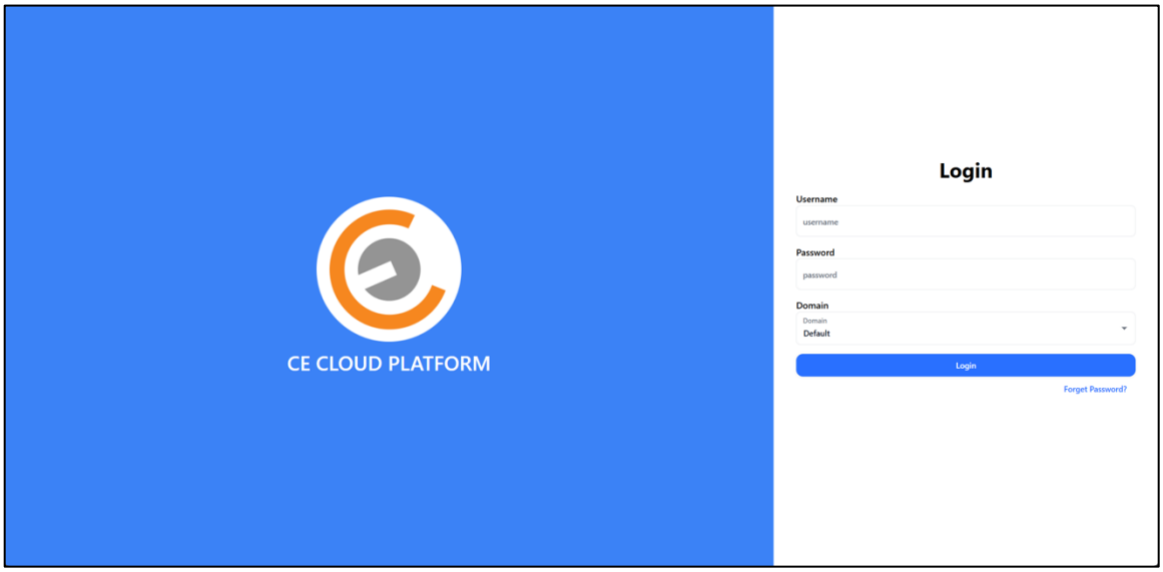


รูปที่ 3.18 Sequence Diagram ของการเข้าถึง Instance ด้วย VNC

3.7 การออกแบบส่วนติดต่อกับผู้ใช้งาน

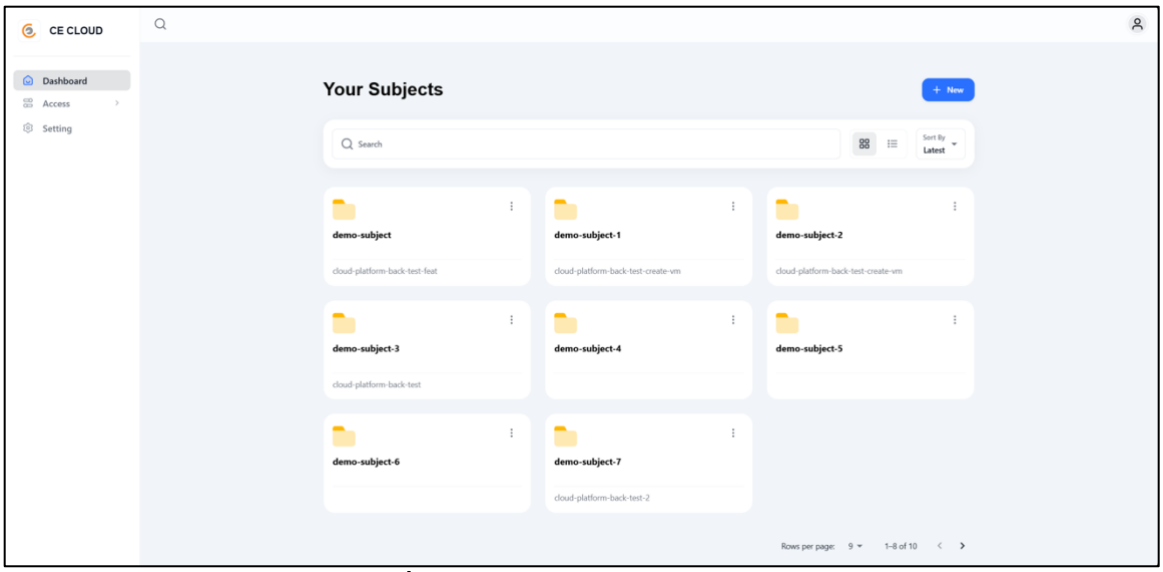
การออกแบบ User interface ของแพลตฟอร์ม มีดังนี้

1) หน้าเข้าสู่ระบบ (Login) สำหรับการยืนยันตัวตนผู้ใช้งาน



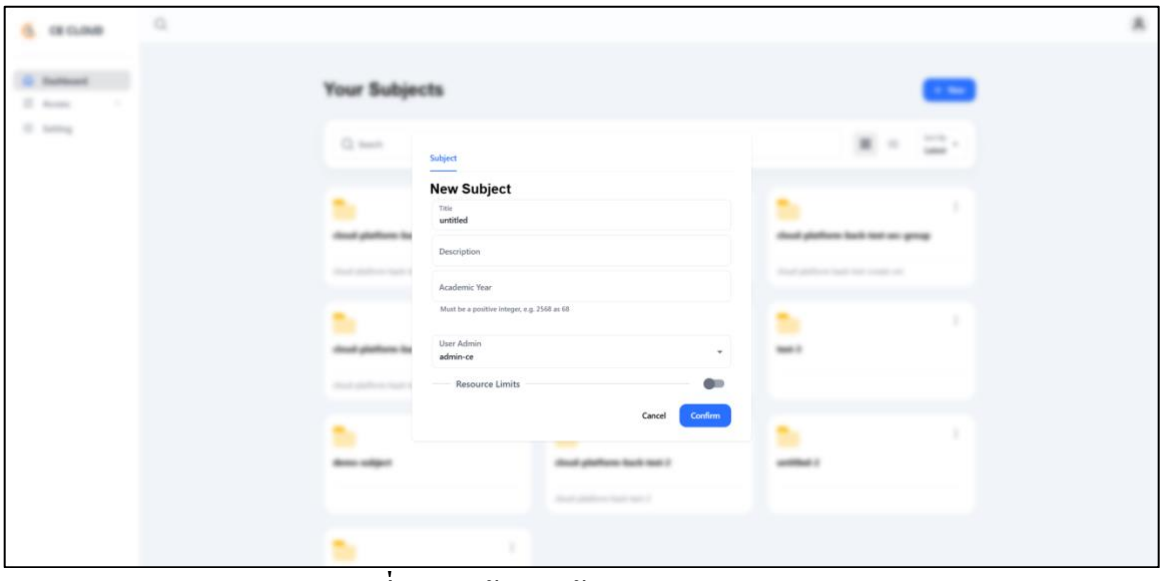
รูปที่ 3.19 หน้าเข้าสู่ระบบ (Login)

2) หน้าแสดงรายวิชา (Subject) ทั้งหมดที่ผู้ใช้งานได้ทำการลงทะเบียนเรียนไว้



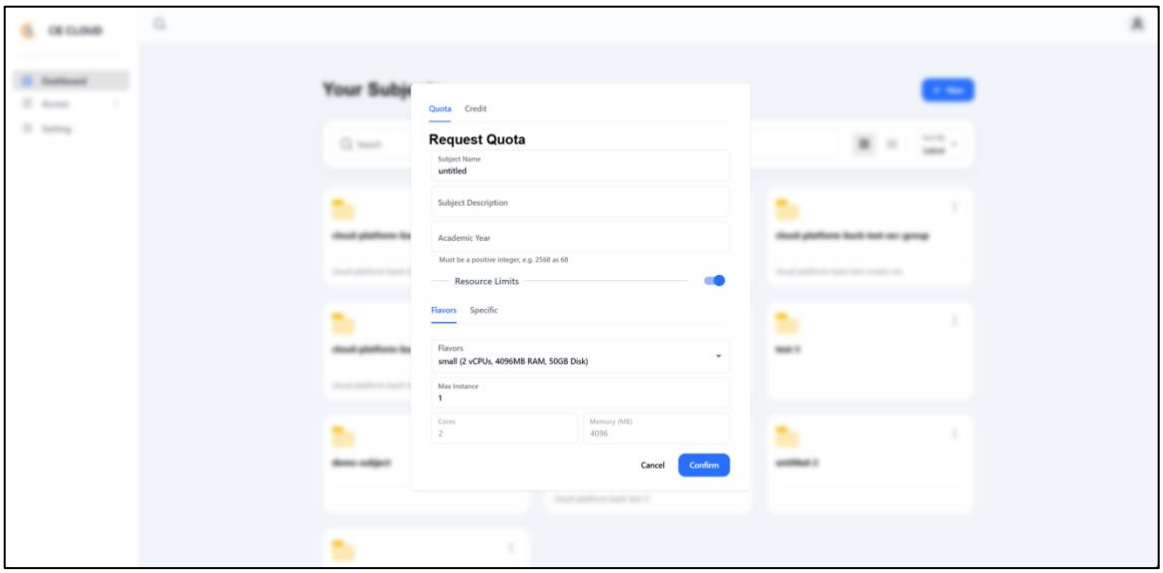
รูปที่ 3.20 หน้าแสดงรายวิชา (Subject)

3) หน้าการสร้างรายวิชา (Subject) สำหรับให้ผู้ดูแลใช้ในการสร้างรายวิชาใหม่ขึ้นมา



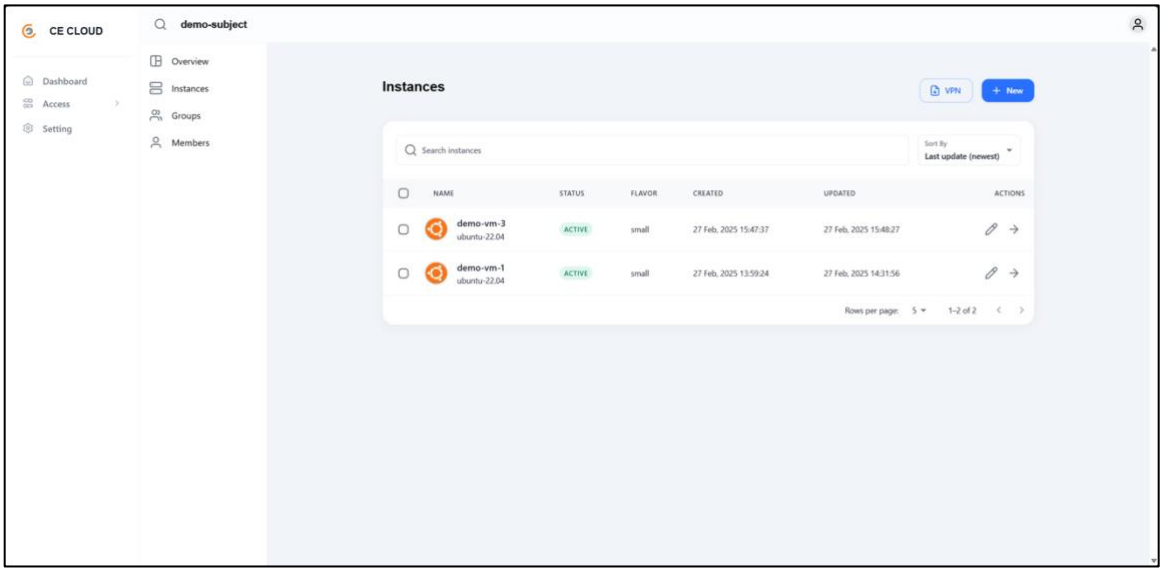
รูปที่ 3.21 หน้าการสร้างรายวิชา (Subject)

4) หน้าการขอโควตา (Quota) สำหรับอาจารย์ประจำวิชาที่ต้องการขอทรัพยากรจำนวนหนึ่งให้นักศึกษาในรายวิชาของตนเองใช้งาน



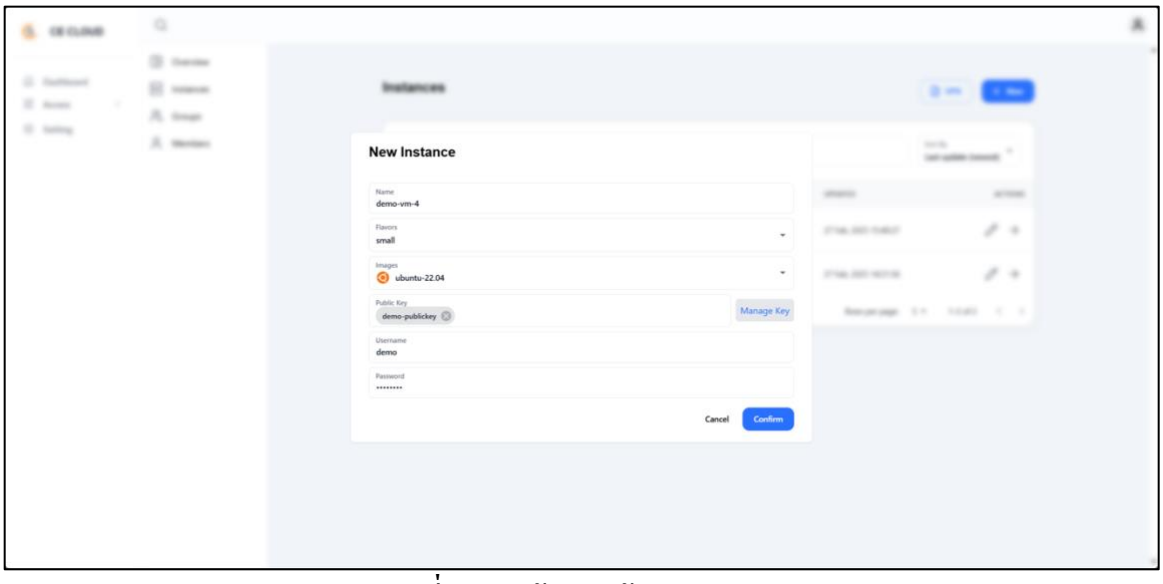
รูปที่ 3.22 หน้าการขอโควตา (Quota)

5) หน้าแสดงรายการ Instance ทั้งหมดของผู้ใช้งานแต่ละคน



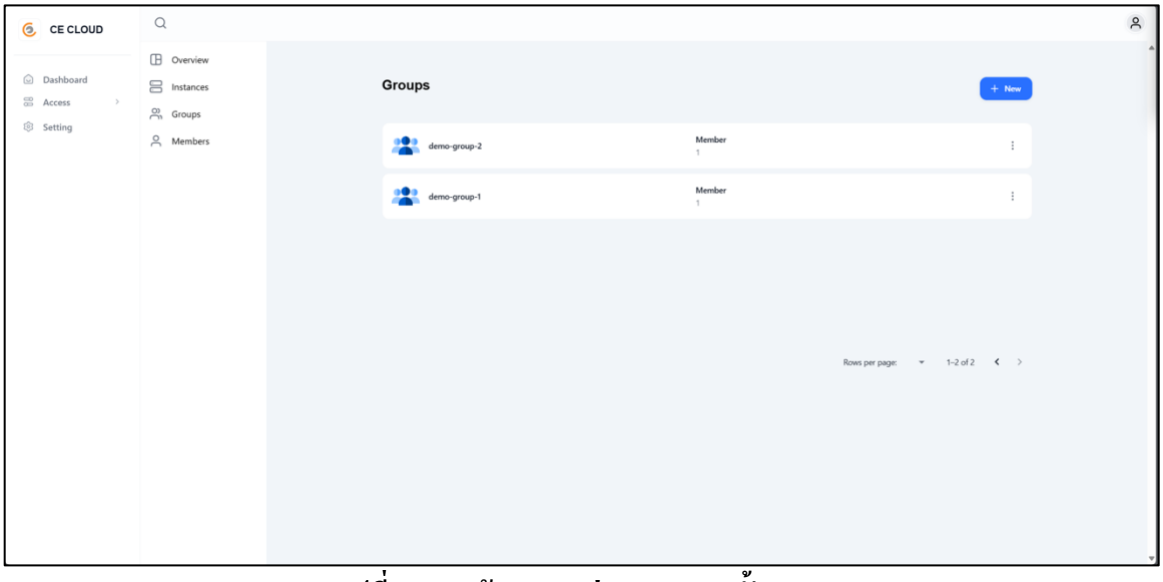
รูปที่ 3.23 หน้าแสดงรายการ Instance

6) หน้าการสร้าง Instance



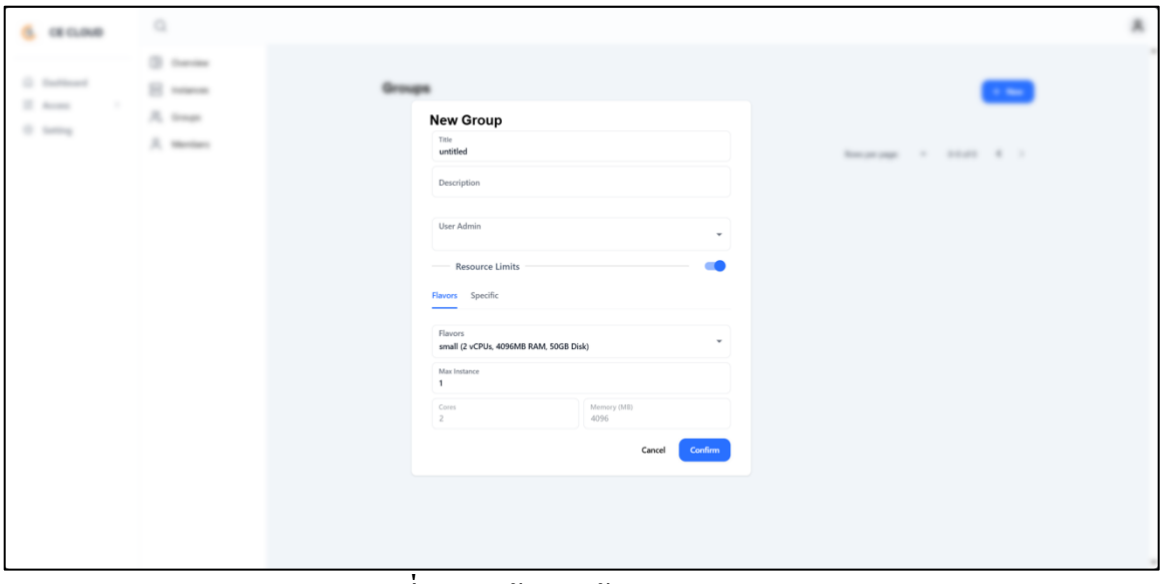
รูปที่ 3.24 หน้าการสร้าง Instance

7) หน้าแสดงกลุ่ม (Group) ทั้งหมด ซึ่งจะแสดงกลุ่มทั้งหมดในรายวิชานั้นๆ



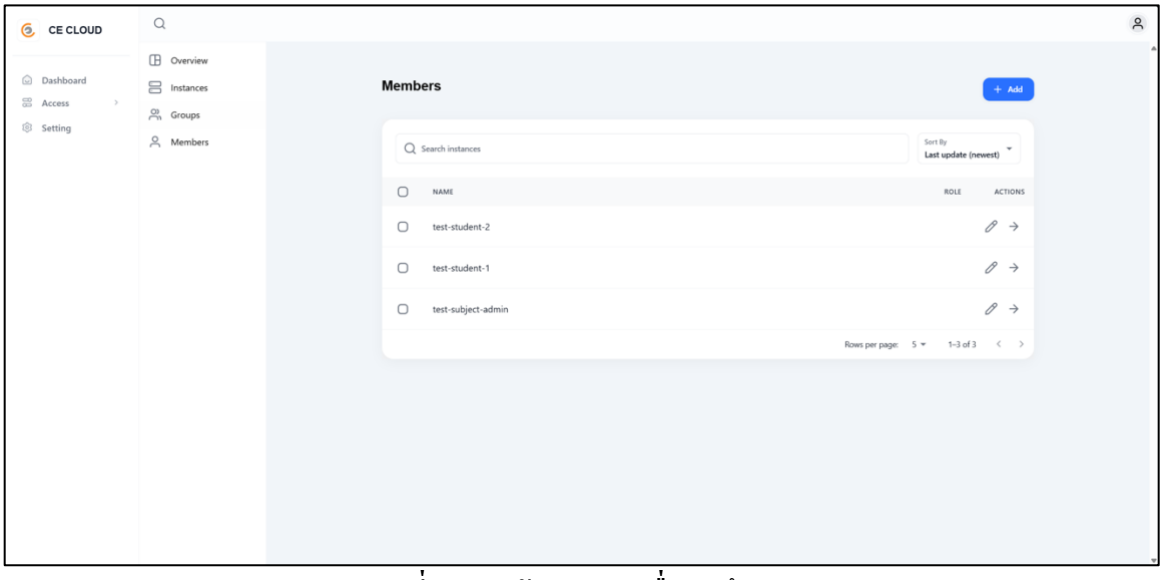
รูปที่ 3.25 หน้าแสดงกลุ่ม (Group) ทั้งหมด

8) หน้าการสร้างกลุ่ม (Group)



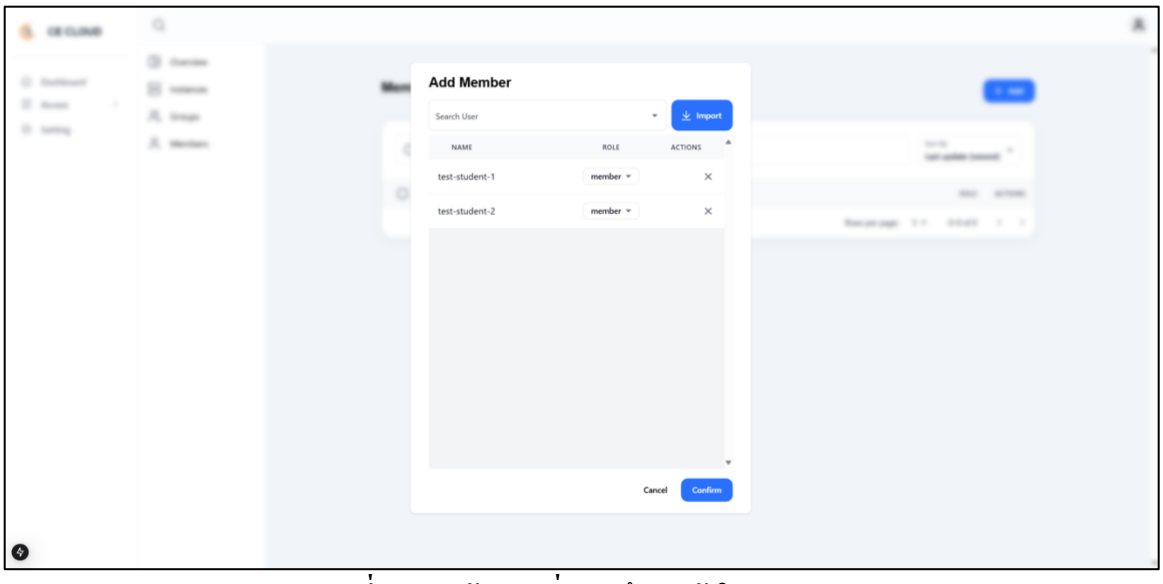
รูปที่ 3.26 หน้าการสร้างกลุ่ม (Group)

9) หน้าแสดงรายชื่อนักศึกษาในแต่ละวิชา



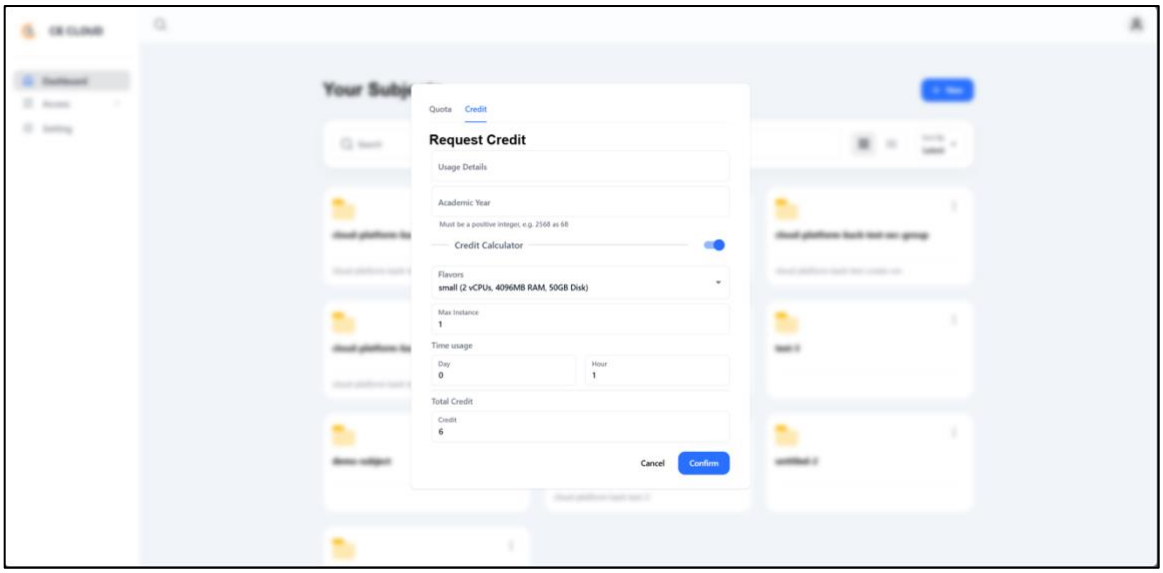
รูปที่ 3.27 หน้าแสดงรายชื่อนักศึกษา

10) หน้าการเพิ่มนักศึกษาเข้าในรายวิชา ซึ่งสามารถเพิ่มเป็นรายคน หรือหลายคนพร้อมกันก็ได้



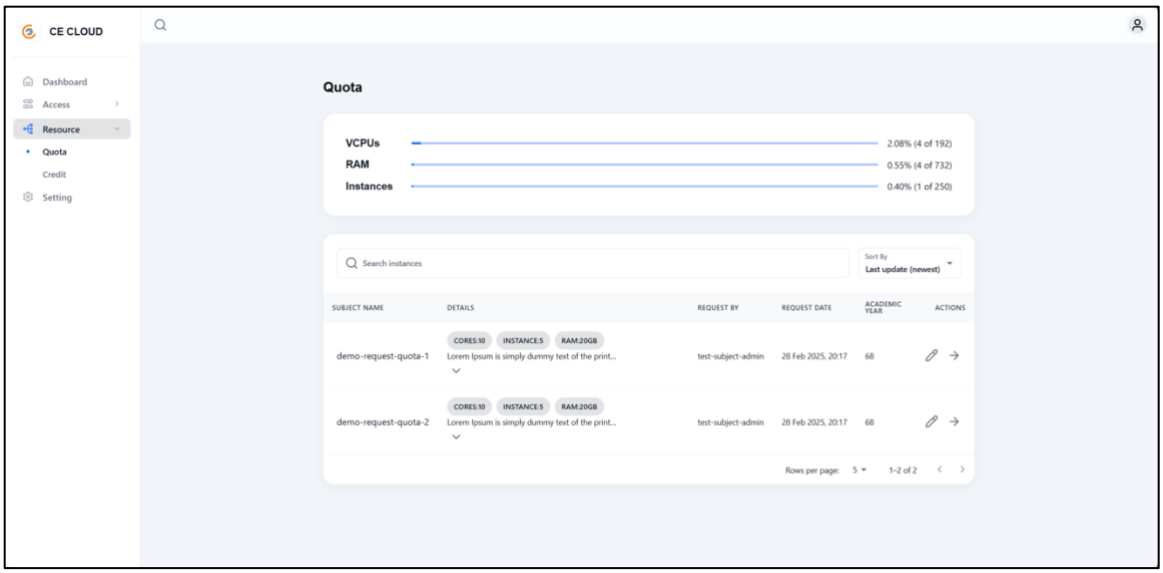
รูปที่ 3.28 หน้าการเพิ่มนักศึกษาเข้าในรายวิชา

11) หน้าการขอเครดิต สำหรับผู้ใช้งานที่ต้องการจะใช้งาน Instance โดยไม่ได้อยู่ภายใต้รายวิชาใดๆ



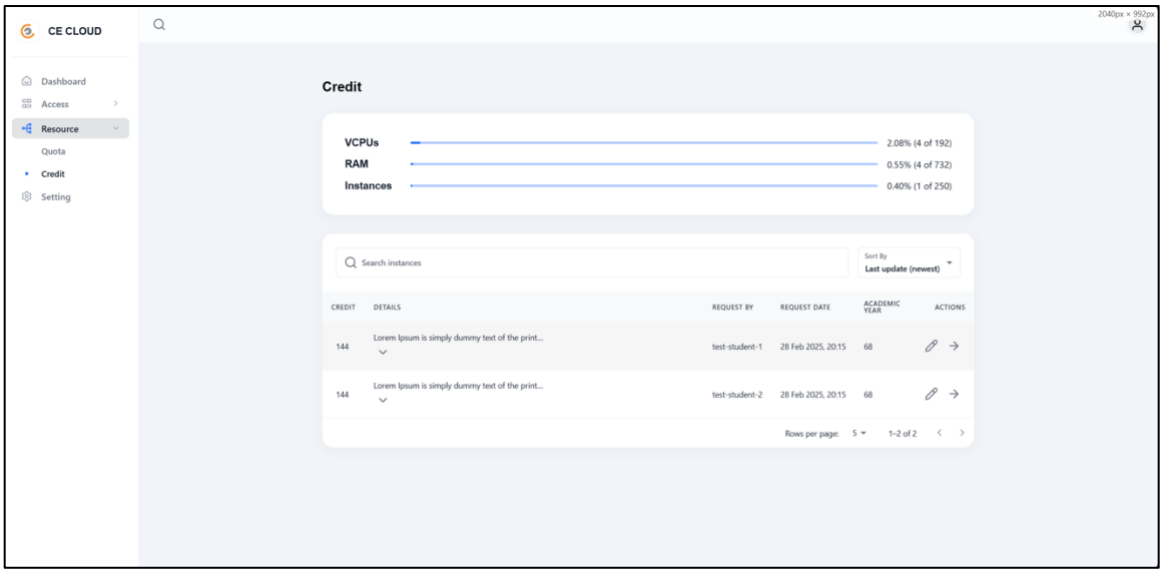
รูปที่ 3.29 หน้าการขอเครดิต

12) หน้าการจัดการโควตา สำหรับให้ผู้ดูแลทำการอนุมัติ หรือปฏิเสธที่จะให้โควตา (Quota) กับอาจารย์ประจำวิชาที่ทำการขอโควตา (Quota)



รูปที่ 3.30 หน้าการจัดการโควตา

13) หน้าการจัดการเครดิต (Credit) สำหรับให้ผู้ดูแลทำการอนุมัติ หรือปฏิเสธที่จะให้เครดิตกับผู้ใช้งานที่ทำการขอเครดิต (Credit)



รูปที่ 3.31 หน้าการจัดการเครดิต (Credit)

บทที่ 4

การทดลองและผลการทดลอง

4.1 ทดสอบการทำงานของแพลตฟอร์ม

4.1.1 วิธีการทดลอง

1) ผู้พัฒนาเข้าใช้งานเว็บแอพลิเคชันของบริการแพลตฟอร์มคลาวด์ที่มีการแบ่งผู้ใช้งานออกเป็น 3 ประเภท ตามการออกแบบที่อธิบายเอาไว้ในหัวข้อ 3.2
2) ทดสอบการทำงานของระบบ Logic, ความถูกต้องของปุ่ม, Flow การทำงานของระบบว่าถูกต้องเป็นไปตามที่ออกแบบเอาไว้ในหัวข้อ 3.3 หรือไม่
3) ทดสอบการทำงานฟังก์ชันทุกฟังก์ชันที่ได้ทำการออกแบบและ Implement ว่า ถูกต้องตาม Use Case ของระบบที่ได้ออกแบบไว้ในหัวข้อ 3.6.1 หรือไม่
4) ทดสอบการทำงานร่วมกันระหว่างส่วนติดต่อผู้ใช้งาน (Frontend) และส่วนระบบจัดการเซอร์วิสต่าง ๆ (Backend) ว่าสามารถทำงานร่วมกันได้อย่างถูกต้องหรือไม่
5) บันทึกผลการทดสอบ

4.1.2 ผลการทดลอง

เว็ปแอพลิเคชันสามารถแสดงผล และใช้งานได้ตามที่ออกแบบไว้ โดยมีรายละเอียดดังนี้

ตารางที่ 4.1 ผลการทดสอบของ Use Case Login (Login สำเร็จ)

Test Case ID	T-001
Use Case	Login
Actor	User
Pre-Condition	ใส่ Username และ Password ถูกต้อง
Main Flow	1) User ใส่ Username และ Password 2) User กดปุ่ม Login
ผลการทดสอบที่คาดหวัง	User สามารถเข้าสู่ระบบได้

Test Case ID	T-001
Use Case	Login
Actor	User
Pre-Condition	ใส่ Username และ Password ถูกต้อง
ผลการทดสอบจริง	User สามารถเข้าสู่ระบบได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.2 ผลการทดสอบของ Use Case Login (Login ไม่สำเร็จ)

Test Case ID	T-002
Use Case	Login
Actor	User
Pre-Condition	ใส่ Username และ Password ไม่ถูกต้อง
Main Flow	1) User ใส่ Username และ Password ที่ไม่ถูกต้อง 2) User กดปุ่ม Login
ผลการทดสอบที่คาดหวัง	User ไม่สามารถเข้าสู่ระบบได้ และระบบขึ้นแจ้งเตือน
ผลการทดสอบจริง	User ไม่สามารถเข้าสู่ระบบได้ และระบบขึ้นแจ้งเตือน
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.3 ผลการทดสอบของ Use Case Logout

Test Case ID	T-003
Use Case	Logout
Actor	User
Pre-Condition	User มี Account ที่สามารถใช้งานได้
Main Flow	User กดปุ่ม Logout
ผลการทดสอบที่คาดหวัง	User ทำการออกจากระบบ และกลับสู่หน้า Login
ผลการทดสอบจริง	User ทำการออกจากระบบ และกลับสู่หน้า Login
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.4 ผลการทดสอบของ Use Case Change Password

Test Case ID	T-004
Use Case	Change Password
Actor	User
Pre-Condition	User มี Account ที่สามารถใช้งานได้
Main Flow	1) User เข้าสู่หน้า IAM ของภาควิชา 2) User เข้าสู่ระบบ 3) User กดเปลี่ยนรหัสผ่าน
ผลการทดสอบที่คาดหวัง	รหัสผ่านของ User จะถูกเปลี่ยน และสามารถใช้ Password ใหม่ในการเข้าสู่ระบบได้

Test Case ID	T-004
Use Case	Change Password
Actor	User
Pre-Condition	User มี Account ที่สามารถใช้งานได้
ผลการทดสอบจริง	รหัสผ่านของ User ถูกเปลี่ยน และสามารถใช้ Password ใหม่ในการเข้าสู่ระบบได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.5 ผลการทดสอบของ Use Case Forgot Password

Test Case ID	T-005
Use Case	Forgot Password
Actor	User
Pre-Condition	User มี Account ที่สามารถใช้งานได้
Main Flow	1) User กดปุ่ม Forgot Password 2) User กรอกอีเมล
ผลการทดสอบที่คาดหวัง	ระบบจะทำการส่งอีเมลการ Reset รหัสผ่านไปยังอีเมลที่กรอก และ User สามารถตั้งรหัสผ่านใหม่ได้
ผลการทดสอบจริง	ระบบทำการส่งอีเมลการ Reset รหัสผ่านไปยังอีเมลที่กรอก และ User สามารถตั้งรหัสผ่านใหม่ได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.6 ผลการทดสอบของ Use Case Create Instance

Test Case ID	T-006
Use Case	Create Instance
Actor	User
Pre-Condition	1) User ต้องทำการเข้าสู่ระบบก่อน 2) User มีสิทธิ์ในการสร้าง Instance ในรายวิชา 3) ระบบมีทรัพยากรเพียงพอสำหรับการสร้าง Instance
Main Flow	1) User กดปุ่มสร้าง Instance 2) User กรอกข้อมูลสำหรับการใช้สร้าง Instance 3) ระบบจะทำการแสดงผลสถานะของ Instance ที่สร้าง
ผลการทดสอบที่คาดหวัง	User สามารถสร้าง Instance ได้
ผลการทดสอบจริง	User สามารถสร้าง Instance ได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.7 ผลการทดสอบของ Use Case Create Instance (ผู้ใช้งานไม่มีสิทธิ์ในการสร้าง)

Test Case ID	T-007
Use Case	Create Instance
Actor	User
Pre-Condition	1) User ต้องทำการเข้าสู่ระบบก่อน 2) User ไม่มีสิทธิ์ในการสร้าง Instance ในรายวิชา 3) ระบบมีทรัพยากรเพียงพอสำหรับการสร้าง Instance
Main Flow	User กดปุ่มสร้าง Instance

Test Case ID	T-007
Use Case	Create Instance
Actor	User
Pre-Condition	1) User ต้องทำการเข้าสู่ระบบก่อน 2) User ไม่มีสิทธิ์ในการสร้าง Instance ในรายวิชา 3) ระบบมีทรัพยากรเพียงพอสำหรับการสร้าง Instance
ผลการทดสอบที่คาดหวัง	User ไม่สามารถกดสร้าง Instance ได้
ผลการทดสอบจริง	User ไม่สามารถกดสร้าง Instance ได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.8 ผลการทดสอบของ Use Case Create Instance (ทรัพยากรไม่เพียงพอ)

Test Case ID	T-008
Use Case	Create Instance
Actor	User
Pre-Condition	1) User ต้องทำการเข้าสู่ระบบก่อน 2) User มีสิทธิ์ในการสร้าง Instance ในรายวิชา 3) ระบบมีทรัพยากรไม่เพียงพอสำหรับการสร้าง Instance
Main Flow	1) User กดปุ่มสร้าง Instance 2) User กรอกข้อมูลสำหรับการใช้สร้าง Instance
ผลการทดสอบที่คาดหวัง	ระบบจะขึ้นสถานะ Pending
ผลการทดสอบจริง	ระบบขึ้นสถานะ Pending
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.9 ผลการทดสอบของ Use Case Access Instance (เข้าใช้งาน Instance ผ่าน VNC)

Test Case ID	T-009
Use Case	Access Instance
Actor	User
Pre-Condition	1) User ทำการเข้าสู่ระบบ 2) User มี Instance ที่ตนมีสิทธิ์ใช้งาน **3)** User ต้องเชื่อมต่ออยู่กับเครือข่ายสถาบัน
Main Flow	1) User ทำการเลือก Instance ที่ต้องการเข้าถึง 2) User กดปุ่ม Console 3) ระบบจะ Redirect ไปที่ VNC Session ของ Instance
ผลการทดสอบที่คาดหวัง	User สามารถเข้าถึง Instance ผ่าน VNC ได้
ผลการทดสอบจริง	User สามารถเข้าถึง Instance ผ่าน VNC ได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.10 ผลการทดสอบของ Use Case Manage Instance (ผู้ใช้งานแก้ไข Instance ของตนเอง)

Test Case ID	T-010
Use Case	Manage Instance
Actor	User
Pre-Condition	1) User ทำการเข้าสู่ระบบ **2)** User มี Instance ที่ตนเองเป็นผู้สร้าง
Main Flow	1) User ทำการเลือก Instance ที่ต้องการแก้ไข 2) User กดปุ่ม Edit

Test Case ID	T-010
Use Case	Manage Instance
Actor	User
Pre-Condition	1) User ทำการเข้าสู่ระบบ **2)** User มี Instance ที่ตนเองเป็นผู้สร้าง
	3) User กรอกข้อมูลที่ต้องการแก้ไข 4) ระบบทำการแสดงผลสถานะการแก้ไข
ผลการทดสอบที่คาดหวัง	User สามารถแก้ไข Instance ได้
ผลการทดสอบจริง	User สามารถแก้ไข Instance ได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.11 ผลการทดสอบของ Use Case Manage Instance (ผู้ดูแลรายวิชาแก้ไข Instance ของผู้อื่น)

Test Case ID	T-011
Use Case	Manage Instance
Actor	Subject Admin
Pre-Condition	1) Subject Admin ทำการเข้าสู่ระบบ **2)** ภายในรายวิชามี Instance ที่ถูกสร้างโดยผู้ใช้คนอื่น
Main Flow	Subject Admin ทำการเลือก Instance ที่ต้องการแก้ไข 1) Subject Admin กดปุ่ม Edit 2) Subject Admin กรอกข้อมูลที่ต้องการแก้ไข 3) ระบบทำการแสดงผลสถานะการแก้ไขUser กดปุ่ม Edit 4) User กรอกข้อมูลที่ต้องการแก้ไข

Test Case ID	T-011
Use Case	Manage Instance
Actor	Subject Admin
Pre-Condition	1) Subject Admin ทำการเข้าสู่ระบบ **2)** ภายในรายวิชามี Instance ที่ถูกสร้างโดยผู้ใช้คนอื่น
	5) ระบบทำการแสดงผลสถานะการแก้ไข
ผลการทดสอบที่คาดหวัง	Subject Admin สามารถแก้ไข Instance ที่ถูกสร้างโดยผู้อื่นได้
ผลการทดสอบจริง	Subject Admin สามารถแก้ไข Instance ที่ถูกสร้างโดยผู้อื่นได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.12 ผลการทดสอบของ Use Case Manage Instance (ผู้ใช้งานลบ Instance ของตนเอง)

Test Case ID	T-012
Use Case	Manage Instance
Actor	User
Pre-Condition	1) User ทำการเข้าสู่ระบบ 2) User มี Instance ที่ตนเองเป็นผู้สร้าง
Main Flow	1) User ทำการเลือก Instance ที่ต้องการลบ 2) User กดปุ่ม Delete 3) ระบบทำการแสดงผลสถานะการลบ Instance
ผลการทดสอบที่คาดหวัง	Instance ถูกลบ
ผลการทดสอบจริง	Instance ถูกลบ

Test Case ID	T-012
Use Case	Manage Instance
Actor	User
Pre-Condition	1) User ทำการเข้าสู่ระบบ 2) User มี Instance ที่ตนเองเป็นผู้สร้าง
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.13 ผลการทดสอบของ Use Case Manage Instance (ผู้ดูแลรายวิชาลบ Instance ของผู้อื่น)

Test Case ID	T-013
Use Case	Manage Instance
Actor	Subject Admin
Pre-Condition	1) Subject Admin ทำการเข้าสู่ระบบ 2) ภายในรายวิชามี Instance ที่ถูกสร้างโดยผู้ใช้คนอื่น
Main Flow	1) Subject Admin ทำการเลือก Instance ที่ต้องการลบ 2) Subject Admin กดปุ่ม Delete 3) ระบบทำการแสดงผลสถานะการลบ Instance
ผลการทดสอบที่คาดหวัง	Instance ถูกลบ
ผลการทดสอบจริง	Instance ถูกลบ
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.14 ผลการทดสอบของ Use Case Add Public Key

Test Case ID	T-014
Use Case	Add Public key
Actor	User
Pre-Condition	User ทำการเข้าสู่ระบบ
Main Flow	1) User กดเมนูจัดการ Public Key 2) User กดเมนู Add Public Key 3) User กรอกข้อมูล Public Key 4) User กดยืนยัน 5) ระบบแสดงผลสถานะการเพิ่ม Public Key
ผลการทดสอบที่คาดหวัง	User สามารถเพิ่ม Public Key ของตนเองได้
ผลการทดสอบจริง	User สามารถเพิ่ม Public Key ของตนเองได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.15 ผลการทดสอบของ Use Case Manage Public key (ลบ)

Test Case ID	T-015
Use Case	Manage Public key
Actor	User
Pre-Condition	1) User ทำการเข้าสู่ระบบ 2) มี Public Key ที่เคยเพิ่มเข้าสู่ระบบอยู่แล้ว
Main Flow	1) User กดเมนูจัดการ Public Key 2) User กดเมนู Delete Public Key 3) User กดยืนยัน

	4) ระบบแสดงผลสถานะการลบ Public Key
ผลการทดสอบที่คาดหวัง	User สามารถกดลบ Public Key ที่เคยมีอยู่ได้
ผลการทดสอบจริง	User สามารถกดลบ Public Key ที่เคยมีอยู่ได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.16 ผลการทดสอบของ Use Case Download VPN

Test Case ID	T-016
Use Case	Download VPN
Actor	User
Pre-Condition	User ทำการเข้าสู่ระบบ
Main Flow	User กดเมนูดาวน์โหลด VPN
ผลการทดสอบที่คาดหวัง	User สามารถดาวน์โหลดข้อมูลสำหรับใช้งาน VPN ได้
ผลการทดสอบจริง	User สามารถดาวน์โหลดข้อมูลสำหรับใช้งาน VPN ได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.17 ผลการทดสอบของ Use Case Request Subject

Test Case ID	T-017
Use Case	Request Subject
Actor	User
Pre-Condition	User ทำการเข้าสู่ระบบ
Main Flow	1) User กดเลือกเมนู Request Subject

	2) User กรอกข้อมูลการ Request 3) User กดปุ่มยืนยันการ Request 4) ระบบแสดงสถานะการ Request
ผลการทดสอบที่คาดหวัง	User สามารถกรอกข้อมูลเพื่อทำการ Request Subject ได้
ผลการทดสอบจริง	User สามารถกรอกข้อมูลเพื่อทำการ Request Subject ได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.18 ผลการทดสอบของ Use Case Manage Group (สร้าง Group)

Test Case ID	T-018
Use Case	Manage Group
Actor	Subject Admin
Pre-Condition	1) Subject Admin ทำการเข้าสู่ระบบ 2) มี Subject ที่จะใช้สำหรับสร้าง Group ภายในนั้น
Main Flow	1) Subject Admin เลือก Subject 2) Subject Admin กดเมนู Create Group 3) Subject Admin กรอกข้อมูลของ Group 4) Subject Admin กดยืนยัน 5) ระบบแสดงสถานะการสร้าง Group
ผลการทดสอบที่คาดหวัง	Subject Admin สามารถสร้าง Group ได้
ผลการทดสอบจริง	Subject Admin สามารถสร้าง Group ได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.19 ผลการทดสอบของ Use Case Manage Group (แก้ไข Group)

Test Case ID	T-019
Use Case	Manage Group
Actor	Subject Admin
Pre-Condition	1) Subject Admin ทำการเข้าสู่ระบบ 2) มี Group ที่สร้างไว้แล้วและต้องการแก้ไข
Main Flow	1) Subject Admin เลือก Group ที่ต้องการแก้ไข 2) Subject Admin กดปุ่ม Edit 3) Subject Admin กรอกรายละเอียดที่ต้องการแก้ไข 4) Subject Admin กดยืนยัน 5) ระบบแสดงสถานะการแก้ไข Group
ผลการทดสอบที่คาดหวัง	Subject Admin สามารถแก้ไข Group ได้
ผลการทดสอบจริง	Subject Admin สามารถแก้ไข Group ได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.20 ผลการทดสอบของ Use Case Manage Group (ลบ Group)

Test Case ID	T-020
Use Case	Manage Group
Actor	Subject Admin
Pre-Condition	1) Subject Admin ทำการเข้าสู่ระบบ 2) มี Group ที่สร้างไว้แล้วและต้องการลบ
Main Flow	1) Subject Admin เลือก Group ที่ต้องการลบ 2) Subject Admin กดปุ่ม Delete

	3) Subject Admin กดยืนยัน 4) ระบบแสดงสถานะการลบ Group
ผลการทดสอบที่คาดหวัง	Subject Admin สามารถลบ Group ได้
ผลการทดสอบจริง	Subject Admin สามารถลบ Group ได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.21 ผลการทดสอบของ Use Case Manage User (เพิ่ม User เข้าสู่ Subject)

Test Case ID	T-021
Use Case	Manage User
Actor	Subject Admin (หรือ Domain Admin)
Pre-Condition	1) Subject Admin ทำการเข้าสู่ระบบ 2) มี Subject ที่สร้างไว้แล้ว 3) Subject Admin มี User ID ที่ต้องการเพิ่มเข้าสู่ Subject
Main Flow	1) Subject Admin เลือก Subject ที่ต้องการเพิ่ม User 2) Subject Admin กดเมนู Add User 3) Subject Admin กรอก User ID ที่ต้องการเพิ่ม 4) Subject Admin กดยืนยัน 5) ระบบแสดงสถานะการเพิ่ม User เข้าสู่ Subject
ผลการทดสอบที่คาดหวัง	User ที่ถูกเพิ่มเข้า Subject สามารถเข้าใช้งาน Subject นั้นได้
ผลการทดสอบจริง	User ที่ถูกเพิ่มเข้า Subject สามารถเข้าใช้งาน Subject นั้นได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.22 ผลการทดสอบของ Use Case Manage User (เพิ่ม User หลายคนเข้าสู่ Subject)

Test Case ID	T-022
Use Case	Manage User
Actor	Subject Admin (หรือ Domain Admin)
Pre-Condition	1) Subject Admin ทำการเข้าสู่ระบบ 2) มี Subject ที่สร้างไว้แล้ว 3) Subject Admin มีไฟล์ CSV ที่เก็บข้อมูล User ID ทั้งหมดที่ต้องการเพิ่มเข้าสู่ Subject
Main Flow	1) Subject Admin เลือก Subject ที่ต้องการเพิ่ม User 2) Subject Admin กดเมนู Add User 3) Subject Admin กดเลือกเมนูอัพโหลด CSV 4) Subject Admin กดยืนยัน 5) ระบบแสดงสถานะการเพิ่ม User เข้าสู่ Subject
ผลการทดสอบที่คาดหวัง	User ทั้งหมดที่อยู่ในไฟล์ CSV จะถูกเพิ่มเข้า Subject และสามารถเข้าใช้งาน Subject นั้นได้
ผลการทดสอบจริง	User ทั้งหมดที่อยู่ในไฟล์ CSV ถูกเพิ่มเข้า Subject และสามารถเข้าใช้งาน Subject นั้นได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.23 ผลการทดสอบของ Use Case Manage User (ลบ User ออกจาก Subject)

Test Case ID	T-023
Use Case	Manage User
Actor	Subject Admin (หรือ Domain Admin)

Pre-Condition	Subject Admin ทำการเข้าสู่ระบบ มี Subject ที่สร้างไว้แล้ว Subject Admin มี User ID ที่ต้องการลบออกจาก Subject
Main Flow	1) Subject Admin เลือก Subject ที่ต้องการลบ User 2) Subject Admin กดเมนู Users 3) Subject Admin เลือก User ที่ต้องการลบออกจาก Subject 4) Subject Admin กดปุ่ม Delete 5) ระบบแสดงสถานะการลบ User ออกจาก Subject
ผลการทดสอบที่คาดหวัง	User ที่ถูกลบออกจาก Subject จะไม่สามารถเข้าใช้งาน Subject นั้นได้
ผลการทดสอบจริง	User ที่ถูกลบออกจาก Subject ไม่สามารถเข้าใช้งาน Subject นั้นได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.24 ผลการทดสอบของ Use Case Manage Quota (อนุมัติคำขอ Quota)

Test Case ID	T-024
Use Case	Manage Quota
Actor	Domain Admin
Pre-Condition	1) มีผู้ใช้งานทำการร้องขอ Quota มาแล้ว 2) Domain Admin เข้าสู่ระบบ
Main Flow	1) Domain Admin เข้าหน้า Quota Request 2) Domain Admin เลือกคำขอที่ต้องการอนุมัติ 3) Domain Admin กดปุ่ม Approve 4) ระบบแสดงสถานะการอนุมัติคำขอ

ผลการทดสอบที่คาดหวัง	ผู้ร้องขอสามารถเข้าใช้งาน Subject ที่ระบบสร้างขึ้นได้
ผลการทดสอบจริง	ผู้ร้องขอสามารถเข้าใช้งาน Subject ที่ระบบสร้างขึ้นได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.25 ผลการทดสอบของ Use Case Manage Quota (ปฏิเสธคำขอ Quota)

Test Case ID	T-025
Use Case	Manage Quota
Actor	Domain Admin
Pre-Condition	1) มีผู้ใช้งานทำการร้องขอ Quota มาแล้ว 2) Domain Admin เข้าสู่ระบบ
Main Flow	1) Domain Admin เข้าหน้า Quota Request 2) Domain Admin เลือกคำขอที่ต้องการปฏิเสธ 3) Domain Admin กดปุ่ม Reject 4) ระบบแสดงสถานะการปฏิเสธคำขอ
ผลการทดสอบที่คาดหวัง	ระบบจะไม่ทำการสร้าง Subject ที่ร้องขอ
ผลการทดสอบจริง	ระบบไม่ทำการสร้าง Subject ที่ร้องขอ
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.26 ผลการทดสอบของ Use Case Create Subject (สร้าง Subject)

Test Case ID	T-026
Use Case	Create Subject
Actor	Domain Admin

Pre-Condition	Domain Admin เข้าสู่ระบบ
Main Flow	1) Domain Admin กดปุ่ม Create ในหน้า Subject 2) Domain Admin กรอกรายละเอียดของ Subject ที่ต้องการสร้าง 3) Domain Admin กดยืนยัน 4) ระบบแสดงสถานะการสร้าง Subject
ผลการทดสอบที่คาดหวัง	Domain Admin สามารถใช้งาน Subject นั้นได้
ผลการทดสอบจริง	Domain Admin สามารถใช้งาน Subject นั้นได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.27 ผลการทดสอบของ Use Case Subject (แก้ไข)

Test Case ID	T-027
Use Case	Manage Subject
Actor	Domain Admin
Pre-Condition	1) Domain Admin เข้าสู่ระบบ 2) มี Subject ที่สร้างไว้แล้วและต้องการแก้ไข
Main Flow	1) Domain Admin เลือก Subject ที่ต้องการแก้ไข 2) Domain Admin กดปุ่ม Edit 3) Domain Admin กรอกรายละเอียดที่ต้องการแก้ไข 4) Domain Admin กดยืนยัน 5) ระบบแสดงสถานะการแก้ไข Subject
ผลการทดสอบที่คาดหวัง	Domain Admin สามารถแก้ไข Subject ได้
ผลการทดสอบจริง	Domain Admin สามารถแก้ไข Subject ได้

สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.28 ผลการทดสอบของ Use Case Manage Subject (ลบ)

Test Case ID	T-028
Use Case	Manage Subject
Actor	Domain Admin
Pre-Condition	1) Domain Admin เข้าสู่ระบบ 2) มี Subject ที่สร้างไว้แล้วและต้องการลบ
Main Flow	1) Domain Admin เลือก Subject ที่ต้องการลบ 2) Domain Admin กดปุ่ม Delete 3) Domain Admin กดยืนยัน 4) ระบบแสดงสถานะการลบ Subject
ผลการทดสอบที่คาดหวัง	Domain Admin สามารถลบ Subject ได้
ผลการทดสอบจริง	Domain Admin สามารถลบ Subject ได้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.29 ผลการทดสอบของ Use Case Manage Credit (อนุมัติ)

Test Case ID	T-029
Use Case	Manage Credit (Approve)
Actor	Domain Admin
Pre-Condition	1) Domain Admin เข้าสู่ระบบ 2) ระบบมีทรัพยากรเพียงพอ

	3) มี User ที่ทำการร้องขอ Credit ไว้แล้ว
Main Flow	1) Domain Admin ทำการเลือกการร้องขอ Credit 2) Domain Admin กดปุ่ม Approve 3) ระบบแสดงสถานะอนุมัติคำร้องขอ Credit
ผลการทดสอบที่คาดหวัง	Domain Admin สามารถอนุมัติคำขอ Credit และ User ที่ร้องขอสามารถนำ Credit ที่ได้รับไปใช้งานต่อได้
ผลการทดสอบจริง	Domain Admin สามารถอนุมัติคำขอ และ User ที่ร้องขอมียอด Credit คงเหลือเพิ่มขึ้น
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.30 ผลการทดสอบของ Use Case Manage Credit (ปฏิเสธ)

Test Case ID	T-030
Use Case	Manage Credit (Reject)
Actor	Domain Admin
Pre-Condition	1) Domain Admin เข้าสู่ระบบ 2) ระบบมีทรัพยากรเพียงพอ 3) มี User ที่ทำการร้องขอ Credit ไว้แล้ว
Main Flow	1) Domain Admin ทำการเลือกการร้องขอ Credit 2) Domain Admin กดปุ่ม Reject 3) ระบบแสดงสถานะการปฏิเสธคำร้องขอ Credit
ผลการทดสอบที่คาดหวัง	Domain Admin สามารถปฏิเสธคำขอ Credit และ User ที่ร้องขอไม่ได้รับ Credit

ผลการทดสอบจริง	Domain Admin สามารถปฏิเสธคำขอ และ User ที่ร้องขอมียอด Credit คงเหลือเท่าเดิม
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.31 ผลการทดสอบของ Use Case Manage Credit (เพิ่ม)

Test Case ID	T-031
Use Case	Manage Credit (Add)
Actor	Domain Admin
Pre-Condition	1) Domain Admin เข้าสู่ระบบ 2) ระบบมีทรัพยากรเพียงพอ 3) มี User ที่ต้องการเพิ่ม Credit ให้
Main Flow	1) Domain Admin ทำการเลือก User ที่ต้องการเพิ่ม Credit ให้ 2) Domain Admin กดปุ่ม Add 3) Domain Admin กรอกจำนวน Credit ที่ต้องการเพิ่มให้ 4) ระบบแสดงสถานะเพิ่ม Credit ให้ User ที่เลือก
ผลการทดสอบที่คาดหวัง	Domain Admin สามารถเพิ่ม Credit ให้แก่ User ได้โดยที่ User ยังไม่จำเป็นต้องส่งคำขอ
ผลการทดสอบจริง	Domain Admin สามารถเพิ่ม Credit ให้แก่ User ได้ และยอด Credit คงเหลือของ User เพิ่มขึ้นตามจำนวนที่เพิ่มให้
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

ตารางที่ 4.32 ผลการทดสอบของ Use Case Request Credit

Test Case ID	T-032

Use Case	Request Credit
Actor	User
Pre-Condition	User เข้าสู่ระบบ
Main Flow	1) User เลือกเมนู Request Credit 2) User กรอกข้อมูลที่จำเป็น 3) User ตรวจสอบข้อมูลและกดปุ่มยืนยัน 4) ระบบแสดงสถานะการยื่นคำขอ
ผลการทดสอบที่คาดหวัง	User สามารถร้องขอ Credit พร้อมแนบเหตุผลได้ และ Domain Admin สามารถเห็นคำร้องขอของ User ได้
ผลการทดสอบจริง	User สามารถร้องขอ Credit และคำร้องขอได้แสดงผลบนหน้า Manage Credit ของ Domain Admin
สรุปผลการทดสอบ	☒ ผ่าน ☐ ไม่ผ่าน

4.2 การทำงานร่วมกันของระบบ

4.2.1 การทำงานร่วมกับ OpenStack

โครงงานนี้ได้ทำการรวบรวม API ของ OpenStack ที่จำเป็นสำหรับแพลตฟอร์มและใช้งาน OpenStack SDK (Software Development Kit) ที่พัฒนาด้วยภาษา Go มีชื่อว่า GopherCloud เพื่อใช้งานร่วมกับแพลตฟอร์มที่พัฒนาด้วยภาษา Go โดยได้ทำการทดลองใช้งาน GopherCloud ตาม Requirements แล้วดังนี้

1) สามารถใช้ชื่อผู้ใช้และรหัสผ่านในการยืนยันตัวตนกับ OpenStack
2) สามารถใช้โทเค็น (Token) ที่ได้จากการยืนยันตัวตนไปใช้งานต่อได้
3) สามารถแสดงผลรายวิชาทั้งหมดของผู้ใช้งานได้
4) สามารถแสดงผลกลุ่มย่อยของรายวิชาทั้งหมดของผู้ใช้งานได้
5) สามารถสร้างรายวิชาได้
6) สามารถแสดงรายการของเครื่องคอมพิวเตอร์เสมือนของผู้ใช้งานได้
7) สามารถสร้างเครื่องคอมพิวเตอร์เสมือนได้

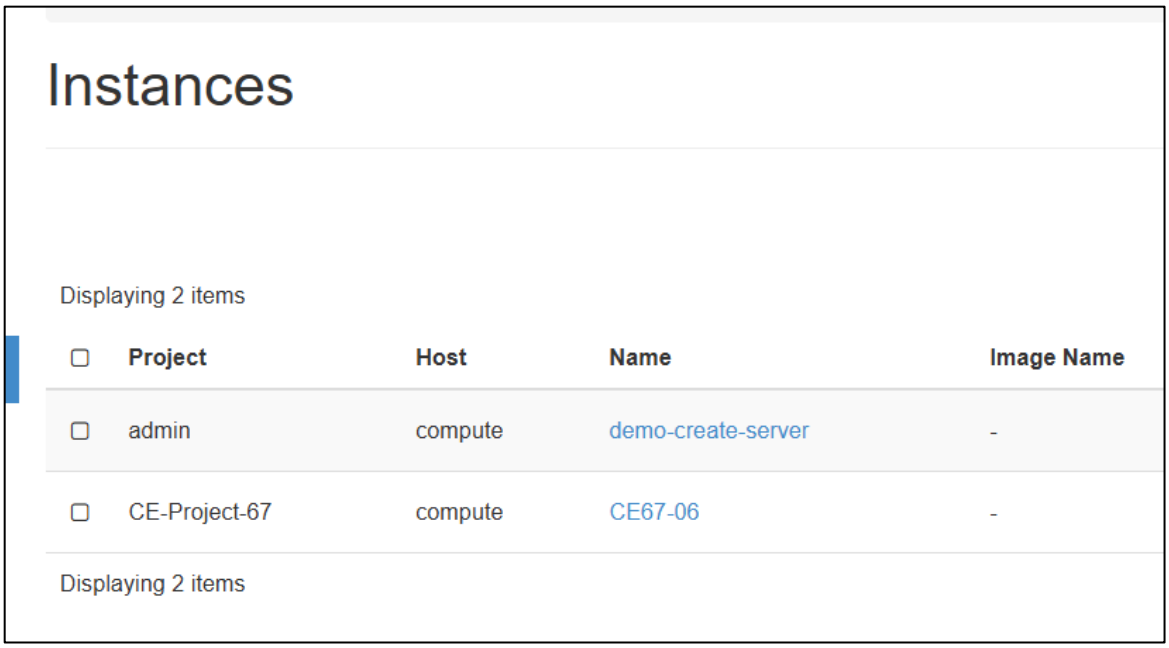
8) สามารถเพิ่ม Floating IP Address ให้กับเครื่องคอมพิวเตอร์เสมือนได้

9) ผู้ใช้สามารถตั้งรหัสผ่านของเครื่องคอมพิวเตอร์เสมือนผ่าน Cloud-Init ได้

โปรแกรม 4.1 ฟังก์ชันสร้างคอมพิวเตอร์เสมือน

```
func (sv *Server) Create(name string, flavorID string,
imageID string, volumeSize int, netUUID string, sshkey
[]string, username string, password string, ip string)
(*servers.Server, error) {
    iTrue := true
    blockDevices := []servers.BlockDevice{
        {
            DeleteOnTermination: true,
            DestinationType:
servers.DestinationVolume,
            SourceType:          servers.SourceImage,
            UUID:                imageID,
            VolumeSize:          volumeSize,
        },
    }

    secGroups := []string{
        "default",
    }

    networks := []servers.Network{
        {
            UUID:    netUUID,
            FixedIP: ip,
        },
    }
```

```
    encodedCloudInit :=
CloudInitBase64(CloudInitConfig{
        User:           username,
        Password:       password,
        PasswordExpire: true,
        SSHPwAuth:      true,
        SSHKeys:        sshkey,
    })

    created, err := servers.Create(context.TODO(),
sv.ServiceClient, servers.CreateOpts{
        Name:             name,
        FlavorRef:        flavorID,
        BlockDevice:      blockDevices,
        SecurityGroups:   secGroups,
        DiskConfig:       servers.Auto,
        AvailabilityZone: "nova",
        Networks:         networks,
        UserData:         encodedCloudInit,
        ConfigDrive:      &iTrue,
    }, servers.SchedulerHintOpts{}).Extract()
    if err != nil {
        return nil, err
    }

    err = servers.WaitForStatus(context.Background(),
sv.ServiceClient, created.ID, "ACTIVE")
    if err != nil {
        return nil, err
    }
    return created, nil
}
```



รูปที่ 4.1 ผลลัพธ์ของการสร้างคอมพิวเตอร์เสมือน

4.2.2 การทำงานร่วมกับ PowerDNS

การพัฒนาในส่วนของ Backend ใช้แพ็คเกจ (Package) ภาษา Go ที่ชื่อว่า Go-PowerDNSซึ่งใช้สำหรับการติดต่อกับ API ของ PowerDNS โดยได้ทำการทดลองใช้งานโดยได้มีการพัฒนาแล้วดังนี้

1) สามารถเพิ่ม DNS Record ผ่าน Backend ของแพลตฟอร์มได้

โปรแกรม 4.2 ฟังก์ชันเพิ่ม DNS Record

```
func (r *Record) Add(ctx context.Context, zoneName,
recordName, content string) error {
    err := r.ServiceClient.Records.Add(ctx, zoneName,
recordName, powerdns.RRTypeA, 3600, []string{content})
    if err != nil {
        return fmt.Errorf("failed to add record: %w",
err)
    }
    fmt.Printf("Record added successfully")
    return nil
}
```

2) สามารถลบ DNS Record ผ่าน Backend ของแพลตฟอร์มได้

โปรแกรม 4.3 ฟังก์ชั่นลบ DNS Record

```
func (r *Record) Remove(ctx context.Context, zoneName,
recordName string) error {
    err := r.ServiceClient.Records.Delete(ctx,
zoneName, recordName, powerdns.RRTypeA)
    if err != nil {
        return fmt.Errorf("failed to remove record:
%w", err)
    }
    fmt.Printf("Record removed successfully")
    return nil
}
```

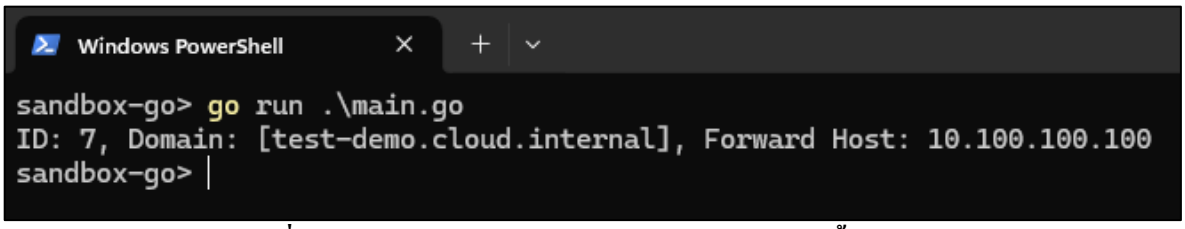
4.2.3 การทำงานร่วมกับ Nginx Proxy Manager

เนื่องจาก Nginx Proxy Manager ไม่ได้มีการเผยแพร่ส่วนที่เป็นอินเทอร์เฟซ (API) ให้ใช้งาน จึงจำเป็นที่จะต้องมีการพัฒนาส่วนที่ใช้เรียก (Client) ขึ้นมาเองโดยเริ่มจากการรวบรวม HTTP Request ที่จำเป็นต้องใช้ในการทำงานร่วมกับโครงงาน และทำการพัฒนาต่อโดยใช้ภาษา Go โดยการทำงานที่ได้ทำการพัฒนาไปแล้วมีดังนี้

1) การ Login เพื่อใช้ในการยืนยันตัวตน
2) การแสดงรายการ ของ Proxy Host ที่มีอยู่ใน Nginx Proxy Manager

โปรแกรม 4.4 ฟังก์ชันแสดงรายการของ Proxy Host ทั้งหมดในระบบ

```
proxclient, err := NewProxyManagerClient(identity,
secret)
if err != nil{panic(err)}
result, err := proxclient.ProxyHost.List()
     if err != nil {
          log.Fatalf("Error list proxy host: %v", err)
     }
     for _, host := range result {
          fmt.Printf("ID: %d, Domain: %v, Forward Host:
     %s\n", host.ID, host.DomainNames,
     host.ForwardHost)
     }
```

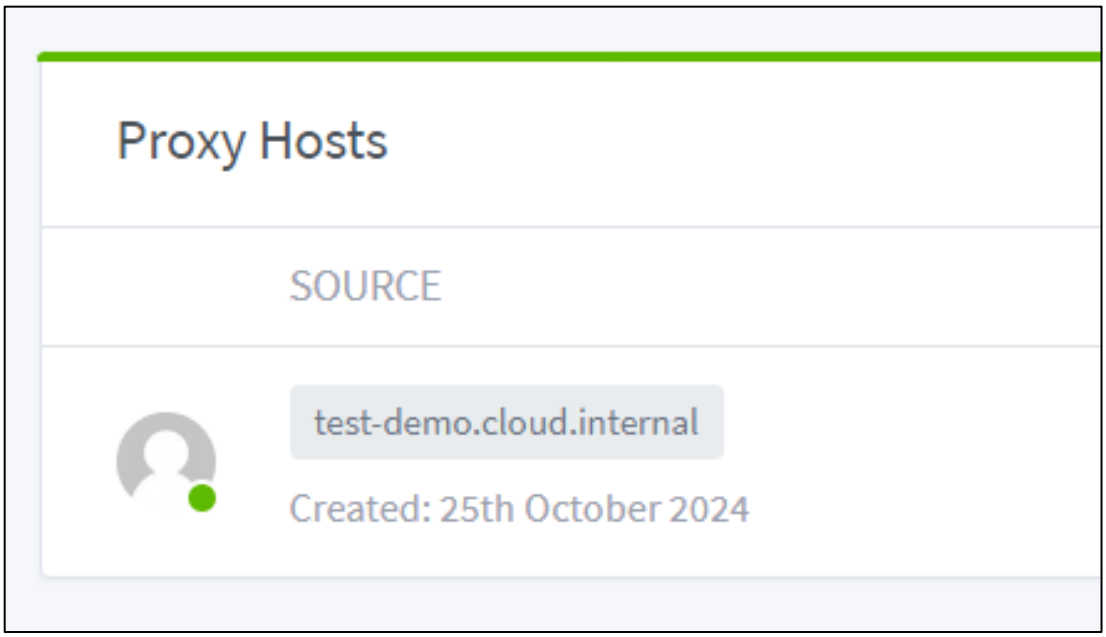


รูปที่ 4.2 ผลลัพธ์ของแสดงรายการของ Proxy Host ทั้งหมด

3) การเพิ่ม Proxy Host

โปรแกรม 4.5 เพิ่ม Proxy Host

```
    proxclient, err := NewProxyManagerClient(identity,
secret)
    if err != nil{panic(err}
    createOpts := CreateOpts{
        DomainName:    "test-demo.cloud.internal",
        ForwardHost:   "10.100.100.100",
        ForwardPort:   80,
        AccessListId:  "0",
        CertificateId: "0",
    }
    result, err :=
proxclient.ProxyHost.Create(createOpts)
    if err != nil{panic(err)}
    fmt.Println(result.DomainNames)
```



รูปที่ 4.3 ผลลัพธ์ของการเพิ่ม Proxy Host

4.2 Centralized Authentication System

เนื่องจากระบบถูกออกแบบมาเพื่อให้เป็น Centralized Authentication จึงมีการติดตั้ง Windows Active Directory เพื่อใช้ในการเก็บชื่อผู้ใช้งาน (นักศึกษาในคณะวิศวกรรมคอมพิวเตอร์) แล้วจึงให้ Openstack (Keystone Service) มาทำการอ่านข้อมูลผู้ใช้งานจาก Windows Active Directory ทั้งนี้การออกแบบนั้นรองรับซอฟแวร์อื่นๆในภาควิชาที่ต้องการยืนยันตัวตนผู้ใช้งาน ก็สามารถมาทำการยืนยันตัวตนกับ Windows Active Directory โดยผ่าน KeyCloak ได้เช่นกัน โดยงานที่ได้พัฒนาไปแล้วมีดังนี้

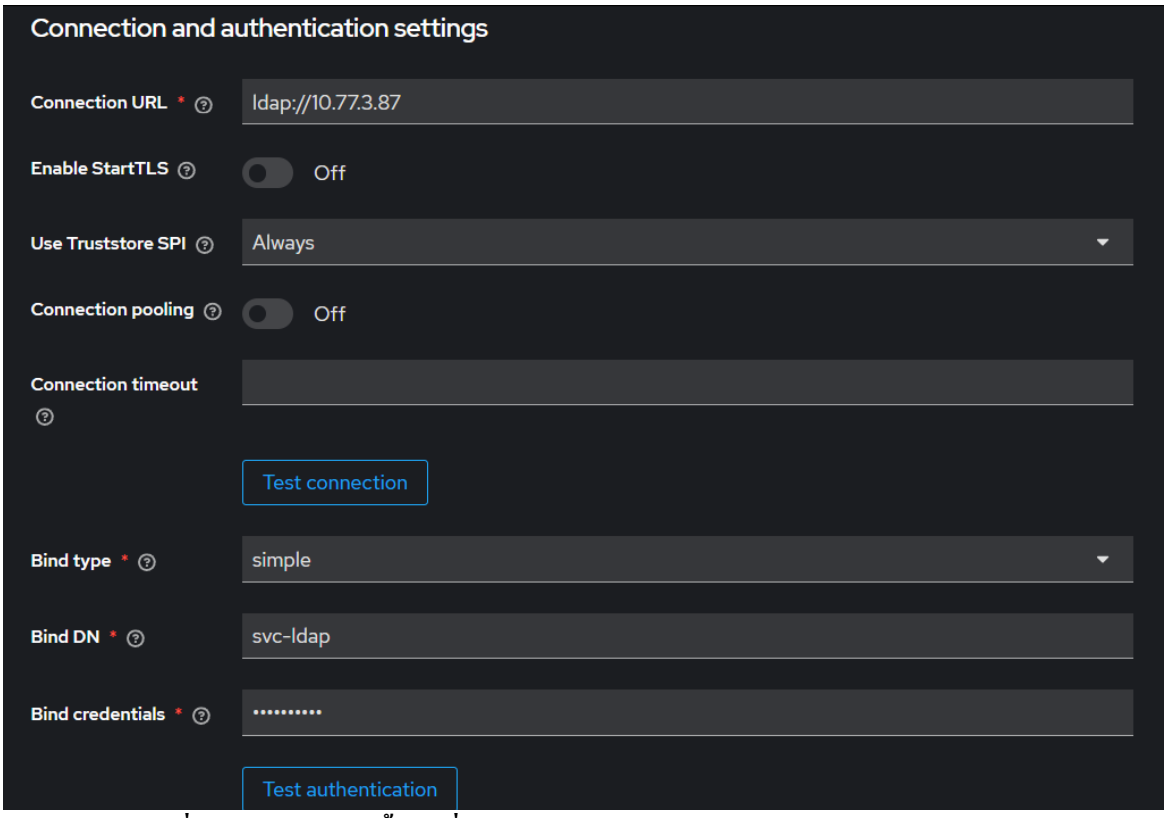
1) เชื่อมต่อ OpenStack กับ Windows Active Directory

โปรแกรม 4.6 ตัวอย่างการตั้งค่าการเชื่อมต่อ OpenStack กับ Windows Active Directory

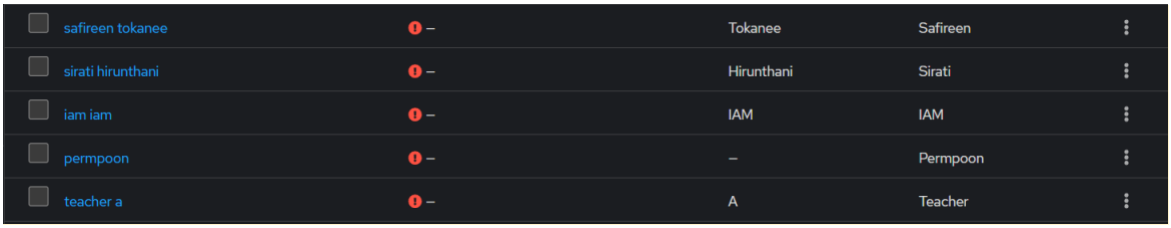
```
# /etc/keystone/domains/keystone.CE.conf
[ldap]
url = ldap://ce-dc.ce.kmitl.internal:389
user =
CN=svc-
ldap,OU=ServiceAccounts,DC=ce,DC=kmitl,DC=internal
password = ********
suffix = DC=ce,DC=kmitl,DC=internal
user_tree_dn = DC=ce,DC=kmitl,DC=internal
user_objectclass = person
user_filter =
(&(objectClass=user)(memberOf=CN=grp-openstack-
ce,OU=CE,DC=ce,DC=kmitl,DC=i
nternal))
user_id_attribute = sAMAccountName
user_name_attribute = sAMAccountName
user_mail_attribute = mail
user_pass_attribute =
user_enabled_attribute = userAccountControl
user_enabled_mask = 2
user_enabled_default = 512
user_attribute_ignore = password,tenant_id,tenants
group_objectclass = group
group_tree_dn = OU=CE,DC=ce,DC=kmitl,DC=internal
group_filter = (CN=grp-openstack-ce)
group_id_attribute = cn
group_name_attribute = name
use_tls = False
query_scope = sub
chase_referrals = false
[identity]
driver = ldap
```

รูปที่ 4.4 ผลลัพธ์ของการเชื่อมต่อ OpenStack กับ Windows Active Directory

2) เชื่อมต่อ Keycloak กับ Windows Active Directory



รูปที่ 4.5 ตัวอย่างการตั้งค่าเชื่อมต่อ Keycloak กับ Windows Active Directory



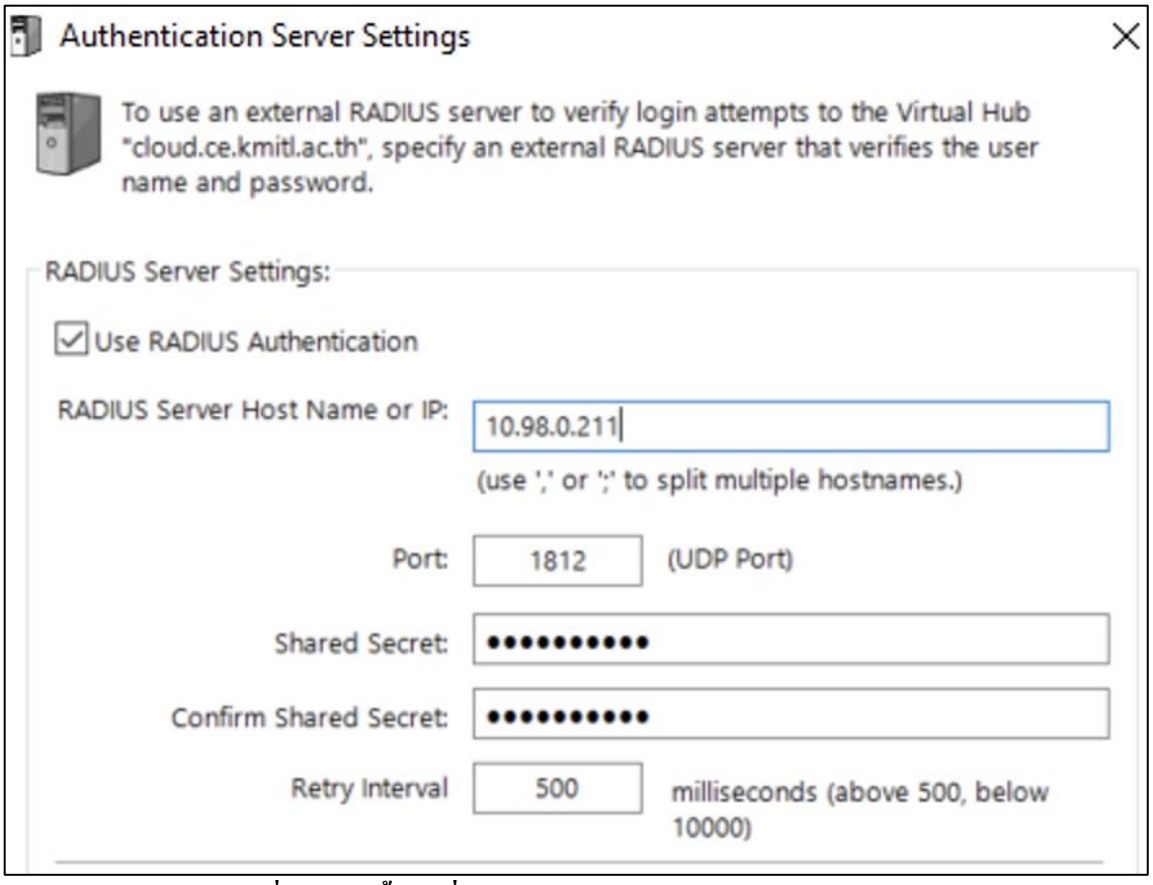
รูปที่ 4.6 ผลลัพธ์เชื่อมต่อ Keycloak กับ Windows Active Directory

4.3 Virtual Private Network (VPN)

เนื่องจากระบบถูกออกแบบมาให้เป็นระบบคลาวด์แบบส่วนตัว (Private Cloud) ผู้ใช้งานจึงต้องทำการเชื่อมต่อ VPN เพื่อให้สามารถเข้าถึงเครือข่ายภายในสถาบันได้ โดยมีรายละเอียดการตั้งค่าหลักๆ ดังนี้

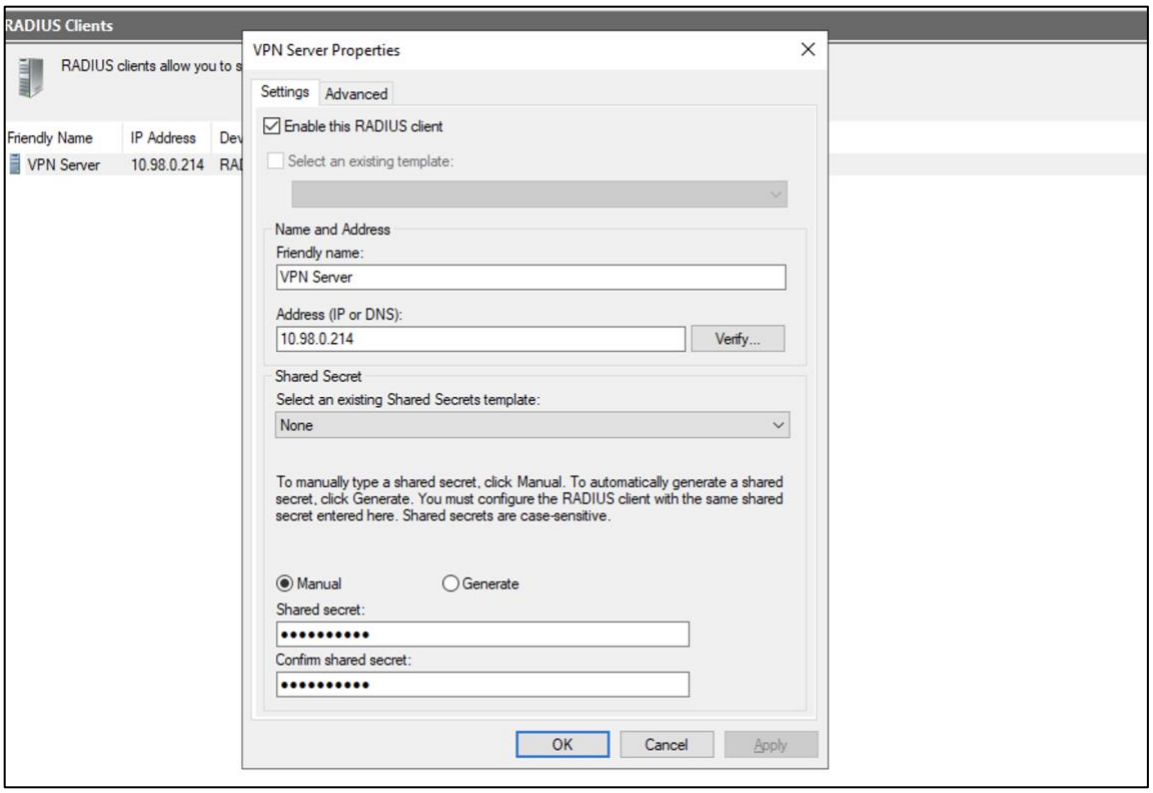
4.3.1 การยืนยันตัวตน

การยืนยันตัวตนในการเชื่อมต่อ VPN นั้นจะทำการยืนยันตัวตนกับ Windows Active Directory ผ่าน RADIUS Protocol โดยการตั้งค่าให้ SofterEther VPN Server ส่งต่อ Packet การยืนยันตัวตนไปที่ RADIUS Server บน Windows Active Directory

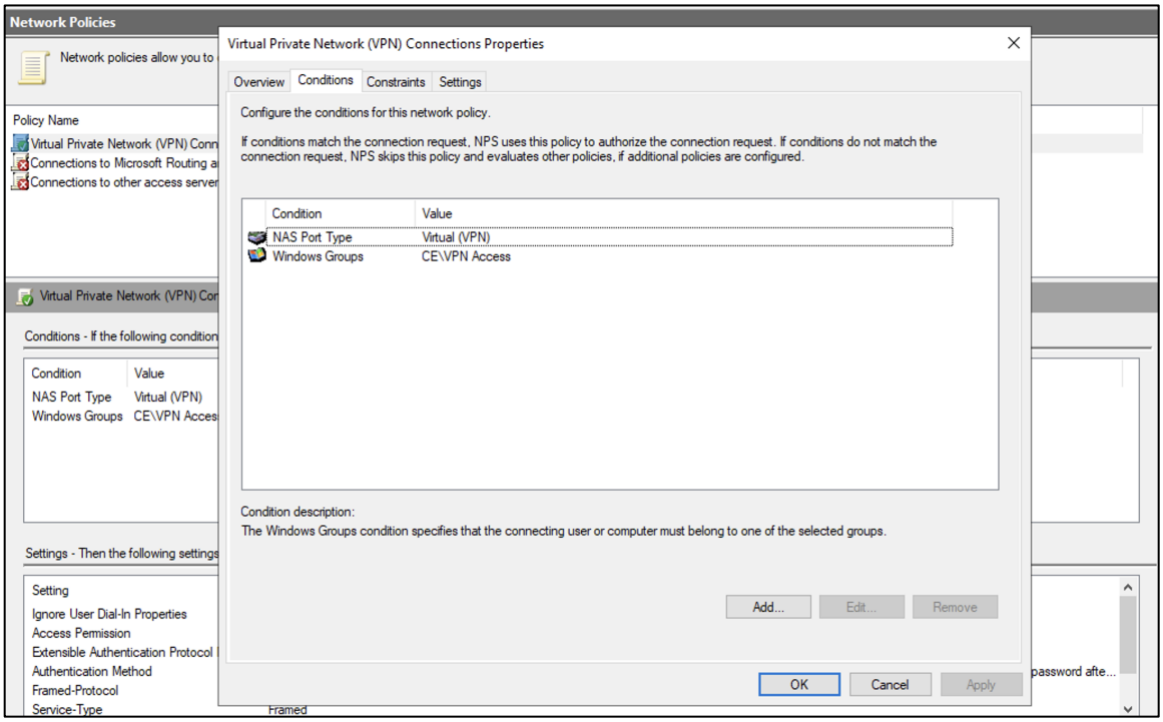


รูปที่ 4.7 การตั้งค่าเชื่อมต่อ RADIUS Server บน VPN Server

และทำการกำหนด IP Address ของเครื่องที่ทำหน้าที่เป็น VPN Server บน RADIUS Server รวมถึงการกำหนด Policy เพื่ออนุญาตให้ชื่อผู้ใช้งานที่อยู่ในกลุ่ม VPN Access เท่านั้นที่สามารถเชื่อมต่อ VPN ได้



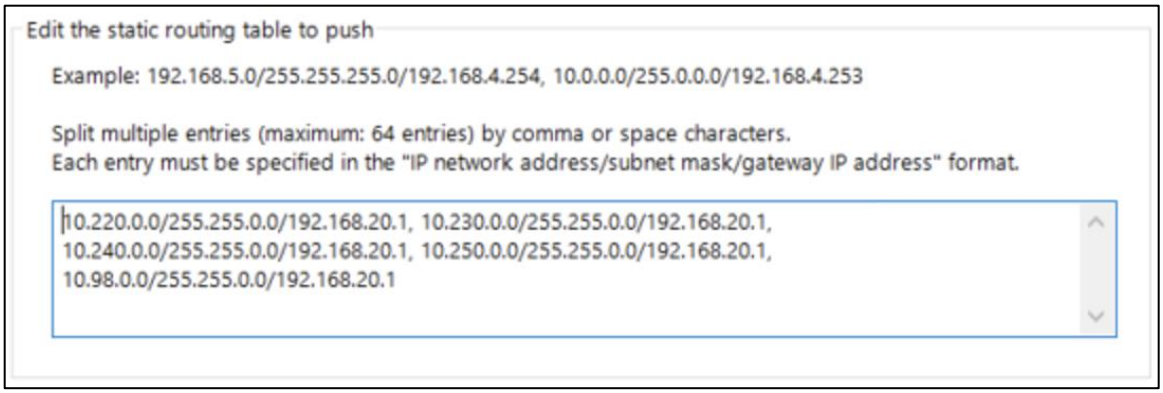
รูปที่ 4.8 การตั้งค่า RADIUS Server บน Windows Active Directory



รูปที่ 4.9 การตั้งค่า Policy ของ RADIUS Server บน Windows Active Directory

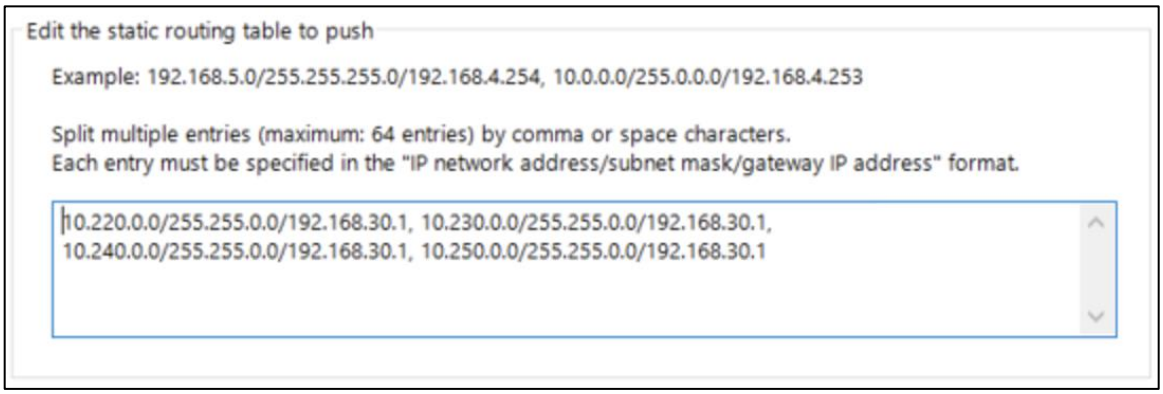
4.3.2 ขอบเขตการเข้าถึงเครือข่ายภายในสถาบัน

การเชื่อมต่อ VPN สำหรับผู้ดูแลระบบและผู้ใช้งานทั่วไปนั้นจะสามารถเข้าถึงเครือข่ายภายในได้แตกต่างกัน โดยที่สำหรับผู้ดูแลระบบจะสามารถเข้าถึงเครือข่าย 10.98.0.1/16, 10.220.0.1/16, 10.230.0.1/16, 10.240.0.1/16, 10.250.0.1/16



รูปที่ 4.10 การตั้งค่า Routing Table สำหรับผู้ดูแลระบบ

และสำหรับผู้ใช้งานปกติจะสามารถเข้าถึงเครือข่าย 10.220.0.1/16, 10.230.0.1/16, 10.240.0.1/16, 10.250.0.1/16



รูปที่ 4.11 การตั้งค่า Routing Table สำหรับผู้ใช้งานทั่วไป

บทที่ 5
สรุปผล และข้อเสนอแนะ

5.1 สรุปผลการพัฒนาโครงงาน

5.1.1 การพัฒนาแพลตฟอร์ม

1) พัฒนาระบบเชื่อมต่อกับ OpenStack
2) พัฒนาระบบเชื่อมต่อกับ PowerDNS
3) พัฒนาระบบเชื่อมต่อกับ Nginx Proxy Manager
4) พัฒนาระบบยืนยันตัวตนและสิทธิ์การใช้งานของผู้ใช้
5) พัฒนาระบบการสร้างคอมพิวเตอร์เสมือน
6) พัฒนาระบบบริหารจัดการวิชาและกลุ่ม
7) พัฒนาระบบจัดการ Public keys ของผู้ใช้
8) พัฒนาระบบจัดการเครือข่ายส่วนตัวเสมือน (Virtual Private Network)
9) พัฒนา User Interface ของ Web Application

5.1.2 การออกแบบและติดตั้งระบบยืนยันตัวตนกลาง (Centralized Authentication System)

1) เชื่อมต่อ Openstack กับ Windows Active Directory
2) เชื่อมต่อ Keycloak กับ Windows Active Directory

5.1.3 การพัฒนาระบบ Identity and Access Management (IAM)

1) เชื่อมต่อ Web Application กับ Windows Active Directory
2) พัฒนาระบบการเปลี่ยนรหัสผ่านและกู้คืนรหัสผ่าน

5.1.4 การติดตั้งระบบเครือข่ายส่วนตัวเสมือน (Virtual Private Network)

1) เชื่อมต่อ SoftEther VPN Server กับ Windows Active Directory

5.2 ปัญหาและอุปสรรคที่พบ

1) เนื่องจากสถาบันมีการจำกัดไม่ให้ใช้งาน Protocol UDP ในการสื่อสารจากภายนอกสถาบัน จึงทำให้ผู้ใช้งานที่ต้องการจะใช้งานคอมพิวเตอร์เสมือนผ่าน VNC จะต้องทำการเชื่อมต่อ VPN จึงจะสามารถใช้งานได้
2) เนื่องจากที่ภาควิชาวิศวกรรมคอมพิวเตอร์เกิดปัญหาไฟฟ้าขัดข้องบ่อยครั้ง ซึ่งอาจทำให้เกิดปัญหาไม่สามารถใช้งานแอพลิเคชันได้เป็นบางเวลา

5.3 แนวทางในการพัฒนาต่อ

1) เพิ่ม Feature ของระบบมากยิ่งขึ้นเพื่อความสะดวกในการใช้งาน เช่น การเพิ่มระบบการสร้าง Container, การเพิ่มระบบการทำ Load Balance หรือ Redundant เป็นต้น
2) เพิ่มประสิทธิภาพในการทำงานของแอพลิเคชันทั้งในเชิง Performance และ Availability

บรรณานุกรม

QEMU. 2020. **Main Page**. [Online]. Available: https://wiki.qemu.org/Main_Page

Libvirt. **Libvirt FAQ**. [Online]. Available: https://wiki.libvirt.org/FAQ.html#what-is-libvirt

KVM. **Main Page**. [Online]. Available: https://linux-kvm.org/page/Main_Page

ธนเดช ธรรมณวโสฬส. 2023. **Software-Defined Networking**. [Online]. Available: https://totdatacom.medium.com/software-defined-networking-sdn-dc27468c9191

Best IDC. 2018. **Ceph storage คืออะไร**. [Online]. Available: https://www.bestinternet.co.th/single_blog.php?id=47&ceph%20storage%20คืออะไร

Kenneth Hess and Amy Newman. 2009. **Virtualization**. [Online]. Available: https://www.computerworld.com/article/2528781/virtualization-technologies-compared.html

Redhat. 2018. **What is software-defined storage**. [Online]. Available: https://www.redhat.com/en/topics/data-storage/software-defined-storage

GopherCloud. 2024. **Gophercloud: an OpenStack SDK for Go**. [Online]. Available: https://github.com/gophercloud/gophercloud/tree/master

Go-PowerDNS. 2024. **PowerDNS 4.x API bindings for Golang**. [Online]. Available: https://github.com/joeig/go-powerdns

Nginx Proxy Manager. 2024. **Nginx Proxy Manager**. [Online]. Available: https://nginxproxymanager.com/

Cloudflare. **What is the Remote Desktop Protocol (RDP)?**. [Online]. Available: https://www.cloudflare.com/learning/access-management/what-is-the-remote-desktop-protocol/

Wikipedia. **Reverse proxy**. [Online]. Available: https://en.wikipedia.org/wiki/Reverse_proxy

Thailand Email Hosting. 2021. **DNS คืออะไร ทำงานอย่างไร**. [Online]. Available: https://thailandemailhosting.com/what-dns-work/

ANET Internet. 2019. **Domain Name Server (DNS) คืออะไร?** . [Online]. Available: https://www.anet.net.th/a/44884/Domain_Name_Server_DNS_คืออะไร_

Microsoft. 2022. **Active Directory Domain Services Overview** . [Online]. Available: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview