

รายงานการปฏิบัติงานสหกิจศึกษา  
เรื่อง  
เว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้  
สมัครงาน  
Pre-Employment Testing

ปฏิบัติงาน ณ บริษัท ไชเงิน

โดย

นายศตวรรษ ธิติสุภกุล  
รหัสประจำตัว 60070093

รายงานนี้เป็นส่วนหนึ่งของการศึกษารายวิชา สหกิจศึกษา  
สาขาวิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ  
ภาคเรียนที่ 1 ปีการศึกษา 2563  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

รายงานการปฏิบัติงานสหกิจศึกษา  
เว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้  
สมัครงาน

**Pre-Employment Testing**

ปฏิบัติงาน ณ บริษัท ไซเงิน

โดย

นายศตวรรษ ชิติสกุล  
รหัสประจำตัว 60070093

ปฏิบัติงาน ณ บริษัท ไซเงิน  
65/60 ชั้น 6 อาคารชานาญเพ็ญชาติปิสน์เซ็นเตอร์ ถนนพระราม 9 เขตห้วยขวาง,  
กรุงเทพมหานคร 10310

Web site : <https://zygencenter.com/>

# **Pre-Employment Testing**

**Satawat Thtisupakul**

**A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF  
THE REQUIREMENT FOR COOPERATIVE EDUCATION PROGRAM  
THE DEGREE OF BACHELOR OF SCIENCE PROGRAM IN  
INFORMATION TECHNOLOGY  
FACULTY OF INFORMATION TECNOLOGY  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**1/ 2020**

**COPYRIGHT 2020**

**FACULTY OF INFORMATION TECHNOLOGY**

**KING MONGKUT'S INSTITUTE OF ECHNOLOGY LADKRABANG**

วันที่ xx พฤศจิกายน พ.ศ. 2563

เรื่อง ขอส่งรายงานการปฏิบัติงานสหกิจศึกษา

เรียน ดร.สุพัฒนดา โชติพันธ์

ที่ปรึกษาสหกิจศึกษาในสาขาเทคโนโลยีสารสนเทศ

ตามที่ข้าพเจ้า นายศตวรรษ ธิติสุภกุล นักศึกษาสาขาวิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ได้ปฏิบัติงานสหกิจศึกษาระหว่างวันที่ 3 สิงหาคม พ.ศ.2563 ถึงวันที่ 30 พฤศจิกายน พ.ศ.2563 ในตำแหน่ง Full Stack Developer ณ สถานประกอบการชื่อ บริษัท ไซเงิน และได้รับมอบหมายจากพนักงานที่ปรึกษาให้ศึกษาและจัดทำรายงานเรื่อง เว็บไซต์พลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน

บัดนี้ การปฏิบัติงานสหกิจศึกษาได้สิ้นสุดลงแล้ว จึงใคร่ขอส่งรายงานการปฏิบัติงาน สหกิจศึกษาดังกล่าวมาพร้อมนี้ จำนวน 1 เล่ม เพื่อขอรับคำปรึกษาต่อไป

จึงเรียนมาเพื่อโปรดพิจารณา

ขอแสดงความนับถือ

.....  
(นายศตวรรษ ธิติสุภกุล)

# กิตติกรรมประกาศ

ตามที่ข้าพเจ้า นายศตวรรษ ธิติสุภกุล ได้มาปฏิบัติงานสหกิจศึกษา ณ บริษัท ไซเงิน ตั้งแต่วันที่ 3 สิงหาคม พ.ศ.2563 ถึงวันที่ 30 พฤศจิกายน พ.ศ.2563 ทำให้ข้าพเจ้าได้รับความรู้และประสบการณ์ต่าง ๆ ที่มีคุณค่ามากมาย สำหรับรายงานสหกิจศึกษานี้สำเร็จลงได้ด้วยดี จากความช่วยเหลือและความร่วมมือสนับสนุนของหลายฝ่าย ดังนี้

1. คุณ วิวัฒน์ เสงี่ยมอนต์ ตำแหน่ง Web Developer (พนักงานที่ปรึกษา)

นอกจากนี้ยังมีบุคคลท่านอื่น ๆ อีกที่ไม่ได้กล่าวไว้ ณ ที่นี้ ซึ่งให้ความกรุณาแนะนำในจัดทำรายงานสหกิจศึกษานี้ ข้าพเจ้าจึงใคร่ขอขอบพระคุณทุกท่านที่ได้มีส่วนร่วมในการให้ข้อมูลและให้ความเข้าใจเกี่ยวกับชีวิตของการปฏิบัติงาน รวมถึงเป็นที่ปรึกษาในการจัดทำรายงานฉบับนี้จนเสร็จสมบูรณ์

นายศตวรรษ ธิติสุภกุล

ผู้จัดทำรายงาน

วันที่ xx พฤศจิกายน พ.ศ. 2563

ชื่อรายงานการปฏิบัติงานสหกิจศึกษา	เว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของ ผู้สมัครงาน
ผู้รายงาน	นายศตวรรษ ธิติศุกุล
คณะ	เทคโนโลยีสารสนเทศ
สาขาวิชา	เทคโนโลยีสารสนเทศ

(ดร.สุพัฒนดา โชติพันธ์)  
อาจารย์ที่ปรึกษาสหกิจศึกษา

(นายวิวัฒน์ เล็งอนันต์)  
พนักงานที่ปรึกษา

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
อนุมัติให้นับรายงานการปฏิบัติงานสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษา  
ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ชื่อรายงาน	เว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน
ชื่อนักศึกษา	นายศตวรรษ ธิติสุกุล
รหัสนักศึกษา	60070093
สาขาวิชา	เทคโนโลยีสารสนเทศ
อาจารย์ที่ปรึกษา	ดร.สุพัฒนดา โชติพันธ์
ปีการศึกษา	2563

## บทคัดย่อ

รายงานการปฏิบัติงานสหกิจศึกษานี้กล่าวถึงวัตถุประสงค์ของการปฏิบัติงาน และรายละเอียดของการปฏิบัติงาน ซึ่งประกอบด้วย ที่มา, ความสำคัญ, รายละเอียด, การออกแบบ ของกระบวนการทำงานของเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน โดยทางบริษัท ไซเงิน ได้มอบหมายในช่วงเวลาที่ปฏิบัติงานสหกิจศึกษา สร้างเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน เป็นเว็บแอปพลิเคชันที่สามารถให้พนักงาน มาออกแบบข้อสอบเพื่อทดสอบความสามารถของผู้ที่มาสมัครงาน โดยการออกข้อสอบนั้นสามารถออกแบบคำถามได้หลายรูปแบบ เช่น คำถามอัตนัย, ปรนัย หรือ ถามโดยการแนบไฟล์ต่างๆ ให้ทำเพื่อวัดความสามารถของผู้สมัครงาน ซึ่งทางบริษัทนั้นมีระบบดังกล่าวมาก่อนหน้านี้แล้ว แต่การส่งข้อสอบให้ผู้สมัครงานในแต่ละครั้งทางบริษัทต้องออกข้อสอบใหม่ทุกครั้ง เพื่อป้องกันข้อสอบเดิมที่อาจถูกเผยแพร่ซึ่งทำให้การทดสอบนั้นมีประสิทธิภาพลดลง พนักงานในบริษัทจึงต้องสละเวลาออกมาออกข้อสอบทุกครั้งที่มีการสมัครงานเพื่อให้ข้อสอบมีมาตรฐานสม่ำเสมอ เนื่องด้วยปัญหาดังกล่าวทางบริษัทจึงมีการมอบหมายให้นักศึกษาสร้างเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงานใหม่ โดยที่เว็บแอปพลิเคชันนี้สามารถกำหนดความยากของข้อสอบ เพื่อทำการสุ่มคำถามจากคลังข้อสอบที่สร้างไว้ ทำให้การทดสอบแต่ละครั้งผู้ทดสอบจะได้ข้อสอบที่แตกต่างกันออกไป อ้างอิงโดยระดับความยากที่ผู้ออกข้อสอบทำการกำหนดไว้ มีการกำหนดเวลาในการทำข้อสอบโดยให้ทำข้อสอบภายในเวลาที่กำหนด และมีการกำหนดวันสิ้นสุดของการทำข้อสอบซึ่งผู้สมัครงานจะไม่สามารถเริ่มข้อสอบทำได้หากเกินเวลาที่กำหนด โดยเว็บแอปพลิเคชันนี้จะทำการส่งข้อสอบให้ผู้สมัครงานผ่านทางอีเมลตามที่ผู้ออกข้อสอบกำหนด เมื่อผู้สมัครงานทำข้อสอบเรียบร้อยแล้วทางบริษัทจะได้รับผลคะแนนที่เป็นปรนัย ส่วนคำถามที่เป็นอัตนัย และคำถามที่มีไฟล์แนบนั้นผู้ที่ส่งข้อสอบให้ทำต้องเป็นผู้ทำการตรวจด้วยตนเอง แล้วระบบจะทำการสรุปคะแนนทั้งหมดให้โดยอัตโนมัติ เว็บแอปพลิเคชันนี้จะทำให้ทางบริษัท ไซเงิน ลดระยะเวลาในการสร้างข้อสอบเพื่อการคัดกรองผู้ที่มาสมัครงาน และได้รับผู้สมัครงานที่มีคุณภาพตรงตามความต้องการของบริษัท



<b>Project Title</b>	Pre-Employment Testing
<b>Name</b>	Satawat Thtisupakul
<b>Student ID</b>	60070093
<b>Department</b>	Information Technology
<b>Advisor</b>	AdvisorName AdvisorSurname
<b>Year</b>	2563

## **Abstract**

Abstract

# สารบัญ

	หน้า
กิตติกรรมประกาศ	vi
บทคัดย่อ	i
บทคัดย่อ ภาษาอังกฤษ	ii
สารบัญ	iii
สารบัญตาราง	iv
สารบัญภาพ	v
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์การปฏิบัติงาน	1
1.3 ประวัติ และรายละเอียดบริษัท	2
บทที่ 2 รายละเอียดการปฏิบัติงาน	4
2.1 ตำแหน่งและลักษณะงานที่นักศึกษาได้รับมอบหมาย	4
2.2 รายละเอียดของโครงการที่ได้รับผิดชอบ	4
2.3 แนวคิดและทฤษฎีที่เกี่ยวข้อง	4
2.4 เครื่องมือและเทคโนโลยีที่ใช้ในการปฏิบัติงาน	10
2.5 ลักษณะขั้นตอนการทำงาน	15
บทที่ 3 การออกแบบระบบและรายละเอียดการพัฒนา	16
3.1 ภาพรวมของเว็บแอปพลิเคชัน	16
3.2 วิเคราะห์ความต้องการ	17
3.3 การวิเคราะห์และออกแบบระบบ	19
3.4 รายละเอียดการพัฒนาระบบ	47
บรรณานุกรม	57
บทที่ ก เรื่องที่หนึ่ง	58

# สารบัญตาราง

	หน้า
ตารางที่ 3.1 รายละเอียดคุณสมบัติ ลงทะเบียน	20
ตารางที่ 3.2 รายละเอียดคุณสมบัติ เข้าสู่ระบบ	21
ตารางที่ 3.3 รายละเอียดคุณสมบัติ สร้างหมวดหมู่คำถาม	22
ตารางที่ 3.4 รายละเอียดคุณสมบัติ คู่มือหมวดหมู่คำถามของผู้ใช้	23
ตารางที่ 3.5 รายละเอียดคุณสมบัติ แก้ไขหมวดหมู่คำถามของผู้ใช้	24
ตารางที่ 3.6 รายละเอียดคุณสมบัติ คัดลอกหมวดหมู่คำถามของผู้ใช้	25
ตารางที่ 3.7 รายละเอียดคุณสมบัติ ลบหมวดหมู่คำถามของผู้ใช้	26
ตารางที่ 3.8 รายละเอียดคุณสมบัติ คู่มือหมวดหมู่คำถามสาธารณะ	27
ตารางที่ 3.9 รายละเอียดคุณสมบัติ คู่มือรายละเอียดของหมวดหมู่คำถามสาธารณะ	28
ตารางที่ 3.10 รายละเอียดคุณสมบัติ คัดลอกหมวดหมู่คำถามสาธารณะ	29
ตารางที่ 3.11 รายละเอียดคุณสมบัติ สร้างข้อสอบ	30
ตารางที่ 3.12 รายละเอียดคุณสมบัติ คู่มือข้อสอบทั้งหมดของตนเอง	31
ตารางที่ 3.13 รายละเอียดคุณสมบัติ แก้ไขข้อสอบของตนเอง	32
ตารางที่ 3.14 รายละเอียดคุณสมบัติ แก้ไขข้อสอบของตนเอง	33
ตารางที่ 3.15 รายละเอียดคุณสมบัติ ลบข้อสอบ	34
ตารางที่ 3.16 รายละเอียดคุณสมบัติ คู่มือข้อสอบสาธารณะ	35
ตารางที่ 3.17 รายละเอียดคุณสมบัติ คู่มือข้อสอบสาธารณะ	36
ตารางที่ 3.18 รายละเอียดคุณสมบัติ คัดลอกข้อสอบสาธารณะ	37
ตารางที่ 3.19 รายละเอียดคุณสมบัติ ส่งข้อสอบให้ผู้สมัครงาน	38
ตารางที่ 3.20 รายละเอียดคุณสมบัติ คู่มือวิธีการส่งข้อสอบทั้งหมด	39
ตารางที่ 3.21 รายละเอียดคุณสมบัติ คู่มือรายละเอียดการส่งข้อสอบ	40
ตารางที่ 3.22 รายละเอียดคุณสมบัติ รายละเอียดข้อสอบ	41
ตารางที่ 3.23 รายละเอียดคุณสมบัติ รายละเอียดข้อสอบ	42
ตารางที่ 3.24 รายละเอียดคุณสมบัติ ส่งข้อสอบให้ทางบริษัท	43
ตารางที่ 3.25 รายละเอียดคุณสมบัติ หมดเวลาทำข้อสอบ	44
ตารางที่ 3.26 รายละเอียดคุณสมบัติ คู่มือข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว	45
ตารางที่ 3.27 รายละเอียดคุณสมบัติ คู่มือข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว	46
ตารางที่ 3.28 ฐานข้อมูลทั้งหมดที่ใช้	47
ตารางที่ 3.29 รายละเอียดฟังก์ชันสร้างผู้ใช้งาน	48

# สารบัญภาพ

	หน้า
รูปที่ 1.1 ตราสัญลักษณ์ของ บริษัท ไซเงิน	2
รูปที่ 1.2 ตัวอย่างการให้บริการของ บริษัท ไซเงิน	3
รูปที่ 2.1 แสดง MVC architecture	8
รูปที่ 2.2 แสดงตาราง SQL เปรียบเทียบกับตาราง NoSQL	8
รูปที่ 2.3 แสดง MVVM architecture	9
รูปที่ 2.4 แสดงตัวอย่างการใช้งาน Trello	10
รูปที่ 2.5 แสดงตัวอย่างการใช้งาน Figma	10
รูปที่ 2.6 แสดงตัวอย่างการใช้งาน vscode	11
รูปที่ 2.7 แสดงตัวอย่างการทำงานของ docker	12
รูปที่ 2.8 แสดงตัวอย่างการใช้งาน postman	13
รูปที่ 2.9 แสดงตัวอย่าง jwt encoded	13
รูปที่ 2.10 แสดงตัวอย่างการใช้งาน Robo 3T	14
รูปที่ 2.11 แสดงตัวอย่างสถิติของอีเมลที่ทำการส่งด้วย sendgrid	15
รูปที่ 3.1 แสดงการทำงานของเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน	16
รูปที่ 3.2 แผนภาพยูสเคสของระบบ	19

# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญ

บริษัท ไซเงิน เป็นบริษัทที่บริการและให้คำปรึกษาและบริการโซลูชัน SAP ในองค์กรต่างๆมากมาย และพัฒนาคิดค้นนวัตกรรมใหม่ๆ เพื่อช่วยแก้ปัญหาในองค์กร ทำให้ธุรกิจสามารถดำเนินไปได้อย่างราบรื่นมากยิ่งขึ้น เช่น การทำ SAP BUSINESS ONE ช่วยทำให้เห็นภาพรวมของการทำธุรกิจได้อย่างชัดเจน, การทำ RPA (Robotic Process Automation) โรบอทที่จะมาช่วยทำงานออฟฟิศแทนมนุษย์ และ BUDDY RECRUIT เทคโนโลยีที่ช่วยให้หาคนเข้ามาทำงานได้ตรงตามความต้องการของบริษัทนั้นๆ

การรับสมัครงาน เป็นกระบวนการที่ทุกบริษัทต้องมีเพื่อที่จะรับพนักงาน ที่มีความสามารถตรงตามที่บริษัทนั้นๆต้องการ ในปัจจุบันกระบวนการรับสมัครงานของแต่ละบริษัทส่วนใหญ่ ก่อนที่จะมีการสัมภาษณ์งานเกิดขึ้น ทางบริษัทจะดูข้อมูลของผู้สมัครผ่านทางเรซูเม่ เพื่อดูประสิทธิภาพการทำงาน และทักษะต่างๆที่ผู้สมัครมี แต่ถ้าหากมีผู้สมัครเป็นจำนวนมาก อาจทำให้ใช้ระยะเวลาในการคัดกรองผู้สมัครงานที่มากขึ้นตามไปด้วย และผู้ที่มาสมัครนั้นอาจมีความสามารถไม่ตรงตามความต้องการของบริษัท จึงมีการสร้างเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน เว็บแอปพลิเคชันนี้จะมาช่วยในการคัดกรองผู้สมัครงาน โดยการให้ทางบริษัทสร้างข้อสอบในเรื่องที่ต้องการวัดความรู้ความสามารถ มาใช้ทดสอบผู้สมัครงาน เพื่อเป็นการลดระยะเวลาในการคัดเลือกพนักงาน และทดสอบว่าผู้สมัครงานนั้นมีความรู้ความสามารถตรงตามที่บริษัทกำหนด ซึ่งทางบริษัทได้มีระบบดังกล่าวมาก่อนหน้านี้แล้ว แต่การส่งข้อสอบในแต่ละครั้งทางบริษัทต้องออกข้อสอบใหม่ทุกครั้ง เพื่อป้องกันข้อสอบที่อาจถูกเผยแพร่ซึ่งทำให้การทดสอบนั้นมีประสิทธิภาพลดลง พนักงานในบริษัทจึงต้องสละเวลาออกข้อสอบทุกครั้ง

ในการปฏิบัติงานครั้งนี้ นักศึกษาได้เป็นในสมาชิกของทีม BUDDY RECRUIT ซึ่งเป็นทีมที่สร้างเทคโนโลยีที่ช่วยให้บริษัทหาคนเข้ามาทำงานได้ตรงตามความต้องการของบริษัทนั้นๆ โดยได้รับมอบหมายงานให้ทำเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงานรูปแบบใหม่ ที่สามารถสุ่มคำถามจากคลังข้อสอบได้ โดยสุ่มตามระดับความยากที่ผู้ออกข้อสอบเป็นคนกำหนด ด้วยวิธีนี้จะช่วยให้ลดระยะเวลาในการให้พนักงานมาออกข้อสอบใหม่ทุกครั้งที่มีผู้สมัครงาน

### 1.2 วัตถุประสงค์การปฏิบัติงาน

1. ศึกษาเทคโนโลยีใหม่ๆ ที่ใช้ในการทำงาน
2. พัฒนาระบบการคิด และการลงมือทำในสายงาน Full Stack Developer
3. พัฒนาทักษะการทำงานภายใต้แรงกดดัน และระยะเวลาที่จำกัด
4. พัฒนาทักษะการแก้ไขปัญหาเฉพาะหน้า

5. ศึกษากระบวนการทำงานภายในทีม
6. พัฒนาทักษะการสื่อสาร
7. นำความรู้ที่ได้ไปต่อยอดกับงานในอนาคต

### 1.3 ประวัติ และรายละเอียดบริษัท

#### 1.3.1 ชื่อสถานประกอบการ

ชื่อบริษัท(ภาษาไทย): บริษัท ไชเจน จำกัด

ชื่อบริษัท(ภาษาอังกฤษ): ZyGen Company Limited

#### 1.3.2 สถานที่ตั้ง

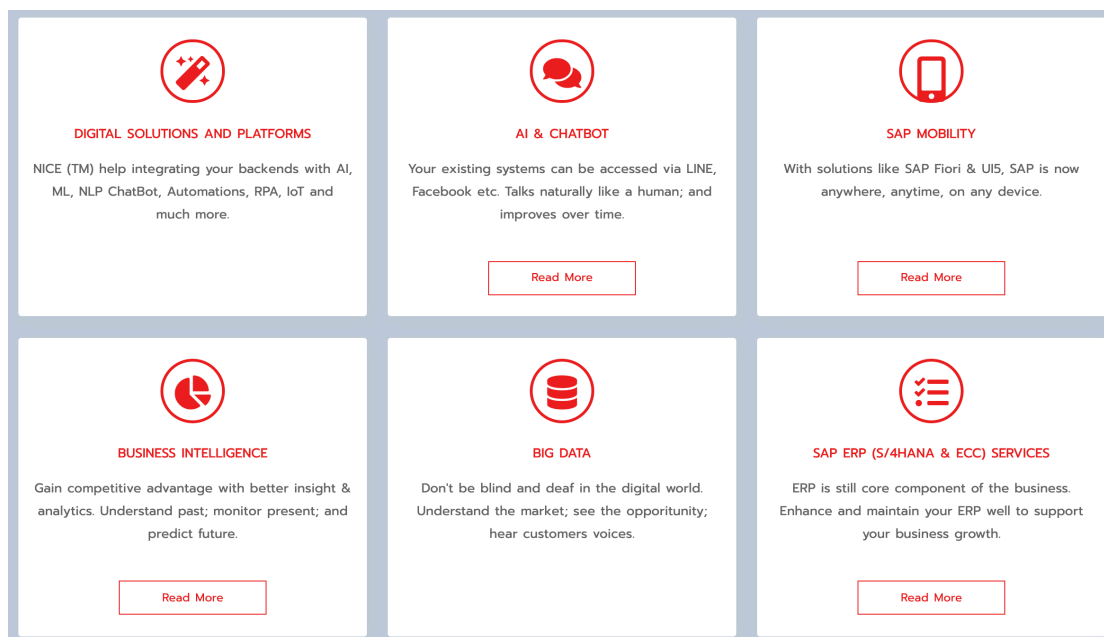
65/60 ชั้น 6 อาคาร ชำนาญเพ็ญชาติบิสเนสเซ็นเตอร์ ถนนพระราม 9 เขตห้วยขวาง, กรุงเทพฯ -  
มหานคร 10310

#### 1.3.3 ลักษณะสถานประกอบการ

บริษัท ไชเจน เป็นบริษัทที่บริการและให้คำปรึกษาและบริการโซลูชัน SAP ในองค์กรต่างๆมากมาย และพัฒนาคิดค้นนวัตกรรมใหม่ๆ เพื่อช่วยแก้ปัญหาในองค์กร ทำให้ธุรกิจสามารถดำเนินไปได้อย่างราบรื่นมากยิ่งขึ้น ไชเจน เป็นบริษัทที่มีฐานลูกค้าที่มั่นคง และยังเพิ่มฐานลูกค้าต่อไป ตัวอย่างการให้บริการของ ไชเจน เช่น การทำ SAP BUSINESS ONE ช่วยทำให้เห็นภาพรวมของการทำธุรกิจได้อย่างชัดเจน, การทำ RPA (Robotic Process Automation) โรบอทที่จะมาช่วยทำงานออฟฟิศแทนมนุษย์ และ BUDDY RECRUIT เทคโนโลยีที่ช่วยให้บริษัทหาคนเข้ามาทำงานได้ตรงตามความต้องการของบริษัทนั้นๆ



รูปที่ 1.1: ตราสัญลักษณ์ของ บริษัท ไชเจน



รูปที่ 1.2: ตัวอย่างการให้บริการของ บริษัท ไซเงิน

#### 1.3.4 ตำแหน่งและลักษณะงานที่นักศึกษาได้รับมอบหมายให้รับผิดชอบ

ตำแหน่ง: Full Stack Developer

หน้าที่: พัฒนาเว็บแอปพลิเคชันตามความต้องการของลูกค้า

#### 1.3.5 ชื่อและตำแหน่งของพนักงานที่ปรึกษา

ชื่อ-นามสกุล: นายวิวัฒน์ เสงี่ยมอนันต์

ตำแหน่ง: Web Developer

แผนก: Development Team

#### 1.3.6 ระยะเวลาปฏิบัติงาน

ช่วงเวลาดำเนินงาน: 1 มิถุนายน 2563 - 30 กันยายน 2563

ช่วงเวลาดำเนินงาน: จันทร์ - ศุกร์ เวลา 09:00 น. - 18:00 น.

รวมระยะเวลา: 4 เดือน

## บทที่ 2

# รายละเอียดการปฏิบัติงาน

### 2.1 ตำแหน่งและลักษณะงานที่นักศึกษาได้รับมอบหมาย

ตำแหน่ง: Full Stack Developer

#### 2.1.1 งานที่ได้รับผิดชอบ

1. ศึกษาเทคโนโลยีที่มีคุณภาพเพื่อประยุกต์ใช้กับการทำงาน
2. พัฒนาเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน โดยเว็บแอปพลิเคชันนี้ช่วยให้คัดกรองผู้คนได้มีประสิทธิภาพมากยิ่งขึ้น ผ่านการทำข้อสอบที่สามารถกำหนดระดับความยากของข้อสอบได้ และดึงคลังคำถามมาแบบสุ่ม ส่งให้ผู้ทำข้อสอบโดยอัตโนมัติผ่านทางอีเมล เมื่อผู้สมัครงานทำข้อสอบเสร็จแล้ว ข้อสอบจะถูกส่งไปที่ผู้ออกข้อสอบ พร้อมตรวจข้อที่เป็นคำถามปรนัยให้อัตโนมัติ โดยระบบนี้แบ่งเป็น 2 ส่วนหลักๆ ได้แก่

### 2.2 รายละเอียดของโครงการที่ได้รับผิดชอบ

เนื่องจากนักศึกษาได้รับหน้าที่ในตำแหน่ง Developer ในทีม BUDDY RECRUIT ซึ่งทำเกี่ยวกับการช่วยให้บริษัทหาคนเข้ามาทำงาน ได้ตรงตามความต้องการของบริษัท จึงได้รับมอบหมายให้ทำเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน โดยเว็บแอปพลิเคชันนี้ช่วยให้คัดกรองผู้คนได้มีประสิทธิภาพมากยิ่งขึ้น ผ่านการทำข้อสอบที่สามารถกำหนดระดับความยากของข้อสอบได้ และดึงคลังคำถามมาแบบสุ่ม ส่งให้ผู้ทำข้อสอบโดยอัตโนมัติผ่านทางอีเมล เมื่อผู้สมัครงานทำข้อสอบเสร็จแล้ว ข้อสอบจะถูกส่งไปที่ผู้ออกข้อสอบ พร้อมตรวจข้อที่เป็นคำถามปรนัยให้อัตโนมัติ โดยระบบนี้แบ่งเป็น 2 ส่วนหลักๆ ได้แก่

1. ส่วนต่อประสานกับผู้ใช้(Front-end) เป็นส่วนที่ผู้ใช้งานสามารถมีปฏิสัมพันธ์กับเว็บแอปพลิเคชันได้ ผ่าน UI(User Interface) ที่หน้าเว็บแอปพลิเคชัน
2. ส่วนที่จัดการกับฐานข้อมูล(Back-end) เป็นส่วนที่ผู้ใช้งานไม่สามารถมองเห็นได้จากภายนอก เป็นการทำงานหลักๆของเว็บแอปพลิเคชัน เช่น การเก็บข้อมูล การเรียกใช้ข้อมูล และการเชื่อมต่อการทำงานร่วมกับระบบอื่นๆ

อนึ่ง ข้อมูลข้างต้นเป็นเพียงภาพรวมของเว็บแอปพลิเคชันโดยย่อ ซึ่งรายละเอียดของเว็บแอปพลิเคชันนี้จะกล่าวโดยละเอียดในบทถัดไป

### 2.3 แนวคิดและทฤษฎีที่เกี่ยวข้อง

#### 2.3.1 RESTful Web Services (RWS)

เป็น web service ที่ใช้ REST architectural style โดยจะอนุญาตให้ระบบ Request และเข้าถึง Resource บนเว็บโดยใช้ชุดคำสั่งที่กำหนดเอาไว้ การตอบโต้ของ REST อยู่บนพื้นฐานของ Hypertext Transfer



Protocol (HTTP) โดย Request จะส่งคำขอไปยัง URI ที่กำหนด และส่งข้อมูลกลับมาในรูปแบบ HTML, XML, JSON หรือ format อื่นๆ ทำให้สามารถบำรุงรักษาง่าย และสามารถ scale service ได้ [1]

- Client-server architecture: Client ผู้ที่เข้ามาขอ resources ไม่ต้องรู้ Business logic ภายใน ส่วน Server มีหน้าที่เก็บข้อมูล ไม่จำเป็นต้องรู้เกี่ยวกับ UI หรือสถานะของผู้เรียก
- Stateless: ส่ง request ให้เซิร์ฟเวอร์แล้วรับกลับมาเป็น response เมื่อรับ response มาแล้วจบการทำงาน
- Cache: สามารถกำหนดได้ว่าเก็บ cache ของ response นั้นหรือไม่
- Layered system: สามารถปรับปรุงความสามารถในการขยายระบบได้ โดยการใช้งานการ ทำ Load balance
- Interface/Uniform Contract: วิธีการที่จะคุยกับเซิร์ฟเวอร์โดยไม่คำนึงถึงประเภทของอุปกรณ์ หรือประเภทของแอปพลิเคชัน

### 2.3.2 HTTP Request

HTTP คือ protocol ที่อนุญาตให้ไคลเอนต์ดึงข้อมูลจากเซิร์ฟเวอร์ [1]

- Request-Line คือส่วนที่ระบุ HTTP Method, Request-URI และ version ของ protocol เช่น HTTP/1.0, HTTP/1.1, HTTP/2.0
- Headers คือส่วนที่อนุญาตให้ใส่ข้อมูลเพิ่มเติม หรือกฎเกณฑ์ต่างๆเกี่ยวกับการ request เช่น รูปแบบของข้อมูล, การเข้ารหัส
- Body คือส่วนที่ระบุข้อมูลที่ต้องการจะส่งให้ปลายทาง สามารถส่ง parameter ต่างๆ ไปใน Body เพื่อเพิ่ม, ลบ หรือแก้ไขข้อมูลบนเซิร์ฟเวอร์ได้

### 2.3.3 HTTP Request Methods

คือส่วนที่ใช้ในการกำหนดประเภทของคำร้องขอต่างๆ บน HTTP Request [1] โดยมี 4 Methods หลักคือ

- POST สำหรับใช้เพื่อสร้างค่าใหม่ เช่น สร้างรายชื่อพนักงานใหม่
- GET สำหรับขอข้อมูลจากเซิร์ฟเวอร์ เช่น ขอข้อมูลพนักงานทั้งหมด
- PUT สำหรับแก้ไขค่าต่างๆบนเซิร์ฟเวอร์ โดยส่งมาใน body ของ HTTP Request เช่น แก้ไขข้อมูลของพนักงาน
- DELETE สำหรับลบค่าบนเซิร์ฟเวอร์ เช่น ลบข้อมูลของพนักงาน

### 2.3.4 HTTP Response Status Code

คือมาตรฐานสถานะที่เซิร์ฟเวอร์ตอบสนองกับเว็บไซต์ต่างๆ [2]

- 2xx (Successful)

Request ที่ไคลเอนต์ส่งไปยังเซิร์ฟเวอร์ถูกประมวลผลเรียบร้อย และไม่มี error ใดๆ ประกอบด้วย

200 : (OK) ส่ง request สำเร็จ

201 : (Created) ผู้ใช้สร้างข้อมูลลง database สำเร็จ response นี้จะได้รับหลัง POST หรือ PUT requests

202 : (Accepted) request สำเร็จแล้วแต่ เซิร์ฟเวอร์ยังประมวลผลไม่เสร็จ

203 : (Non-Authoritative Information) เซิร์ฟเวอร์ประมวลผลสำเร็จแล้ว แต่ทำการส่งข้อมูลมาจากแหล่งอื่น

204 : (No Content) เซิร์ฟเวอร์ประมวลผลสำเร็จแล้ว แต่ไม่มีข้อมูลที่ต้องส่งคืนไป

206 : (Partial Content) เซิร์ฟเวอร์ส่งข้อมูลบางส่วน ตามที่ไคลเอนต์ต้องการ โดยกำหนดขอบเขตที่ต้องการผ่าน headers บน HTTP Request

- 3xx (Redirection)

request ที่ไคลเอนต์ส่งไปหาเซิร์ฟเวอร์ แล้วถูก redirect ส่งไปประมวลผลที่อื่น เพื่อให้กระบวนการทำงานสำเร็จ

300 : (Multiple Choices) request ที่ไคลเอนต์ส่งไปมี response มากกว่า 1 ตัว ไคลเอนต์สามารถเลือกสิ่งที่จะ redirect ไปได้

301 : (Moved Permanently) URL ที่ทำการ request ของข้อมูลถูกเปลี่ยนไปถาวร จึง response ออกมาเป็น URL ใหม่

302 : (Found) URL ที่ทำการ request มีการเปลี่ยนชั่วคราว

303 : (See Other) request ที่เรียกอยู่ภายใต้ URL อื่น

304 : (Not Modified) response นี้ยังไม่ถูกแก้ไข ดังนั้น ไคลเอนต์จะได้รับ response ที่เป็น cached version

- 4XX (Client Error)

เกิด error มาจาก request ของไคลเอนต์ที่ผิดพลาด เช่น ผิด URL หรือผิด syntax

400 : (Bad Request) ไคลเอนต์เขียน syntax ผิด หรือ ไม่ถูกรูปแบบ ทำให้เซิร์ฟเวอร์ไม่เข้าใจ

401 : (Unauthorized) ไคลเอนต์ต้องทำการยืนยันตัวตนก่อนที่จะได้รับ response

403 : (Forbidden) ไคลเอนต์ทำการยืนยันตัวตนแล้ว แต่ไม่มีสิทธิ์ในการเข้าถึงข้อมูลนี้

404 : (Not Found) ถ้าเกิดบน browser คือ URL ไม่ถูกจดจำบนเซิร์ฟเวอร์ แต่ถ้าเกิดบน API คือมีการขอข้อมูลที่ต้องการ แต่ไม่มีข้อมูลนี้อยู่

405 : (Method Not Allowed) method ที่เรียกใช้ไม่ถูกต้อง

406 : (Not Acceptable) header ที่ไคลเอนต์ request ไม่สัมพันธ์กับเซิร์ฟเวอร์

413 : (Payload Too Large) request ที่ขอใหญ่กว่า limit ที่เซิร์ฟเวอร์กำหนดไว้

414 : (URI Too Long) URL ที่ทำการ request โดยไคลเอนต์ ยาวกว่าที่เซิร์ฟเวอร์จะยอมรับได้

415 : (Unsupported Media Type) เซิร์ฟเวอร์ไม่รองรับ media (รูป หรือ สื่อต่างๆ) ดังนั้นเซิร์ฟเวอร์จึงปฏิเสธการ request

- 5XX (Server Error)

เซิร์ฟเวอร์มีปัญหา

500 : (Internal Server Error) เซิร์ฟเวอร์เจอกับสถานการณ์ที่ไม่สามารถจัดการได้

501 : (Not Implemented) ไคลเอนต์เรียก request method ที่เซิร์ฟเวอร์ไม่รองรับ และเซิร์ฟเวอร์ไม่สามารถจัดการได้

502 : (Bad Gateway) เซิร์ฟเวอร์เป็น gateway หรือ proxy ได้รับ response ที่ผิดพลาดจากเซิร์ฟเวอร์อื่น

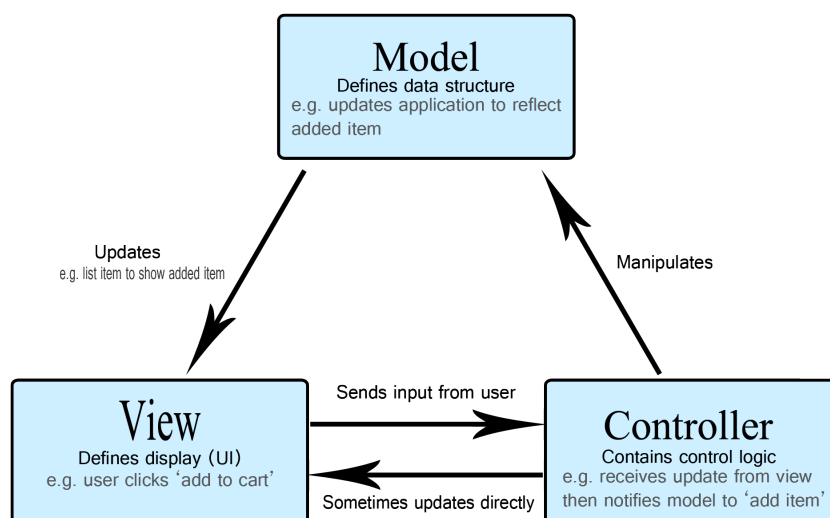
503 : (Service Unavailable) เซิร์ฟเวอร์อยู่ระหว่างการปรับปรุง หรือยังไม่พร้อมที่จะจัดการ request

504 : (Gateway Timeout) เซิร์ฟเวอร์เป็น gateway และไม่สามารถ response ข้อมูลในเวลาที่กำหนดได้

### 2.3.5 MVC (Model View Controller)

คือ software design pattern ที่แยกการทำงานของโปรแกรมออกเป็น 3 ส่วนเพื่อแยกข้อมูลภายในโปรแกรมกับข้อมูลที่แสดงให้ผู้ใช้เห็น [3]

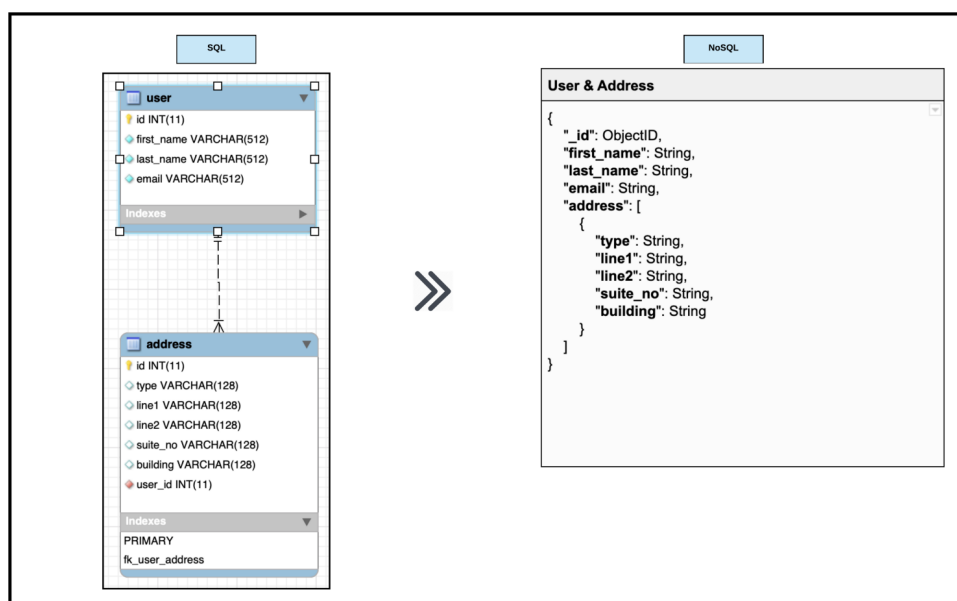
- Model คือส่วนที่เป็นโครงสร้างของข้อมูล กำหนดกฎเกณฑ์ของการเก็บข้อมูล และเป็นส่วนที่ใช้สำหรับการจัดการข้อมูลโดยตรง
- View คือส่วนที่ไว้แสดงผลตามที่ผู้ใช้ต้องการ ผู้ใช้สามารถเปลี่ยนแปลงข้อมูลที่ผู้ใช้เห็นได้
- Controller คือส่วนที่ไว้จัดการกับ Model โดยขึ้นอยู่กับการทำงานของ View ที่กำหนดโดยผู้ใช้ และสรรหาข้อมูลจาก Model เพื่อไปแสดงใน View



รูปที่ 2.1: แสดง MVC architecture

### 2.3.6 NoSQL Database

คือฐานข้อมูลที่สร้างมาเพื่อให้จัดการข้อมูลที่มีความซับซ้อนได้ง่ายขึ้นเมื่อเทียบกับ SQL databases ที่มีการเก็บข้อมูลในที่รูปแบบแน่นอน (structured data) โดยเพิ่มความสามารถในการจัดเก็บข้อมูลในรูปแบบที่ไม่แน่นอน (unstructured data) ทำให้สามารถเก็บข้อมูลที่ซับซ้อนได้, เพิ่มความสามารถในการขยายระบบในรูปแบบแนวนอน (Horizontal Scalability) เพื่อรองรับปริมาณข้อมูลในปัจจุบัน [4]



รูปที่ 2.2: แสดงตาราง SQL เปรียบเทียบกับตาราง NoSQL

### 2.3.7 Token

เป็นชุดรหัสเอาไว้ระบุตัวตนของผู้ใช้ว่าผู้ใช้นั้นเป็นใคร ไม่มีรูปแบบที่ตายตัว

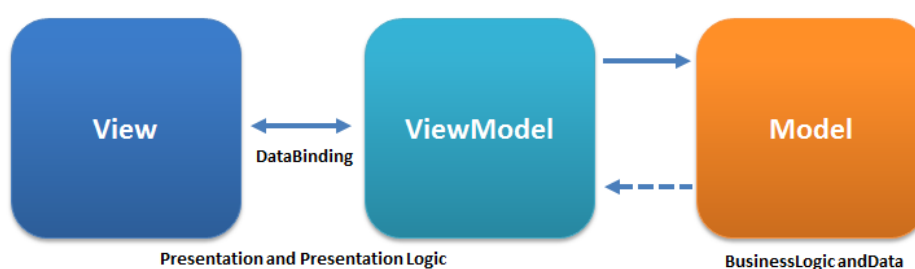
### 2.3.8 Container

เป็นหน่วยของซอฟต์แวร์ที่ทำการลงทรัพยากรทุกอย่างที่ต้องใช้ในแอป และตัวโค้ดของแอป ดังนั้นแอปพลิเคชันจะสามารถถูกเรียกใช้ได้อย่างรวดเร็ว และใช้ในสภาพแวดล้อมใดก็สามารถทำงานได้โดยมีความเป็นมาตรฐาน และประหยัดทรัพยากรที่ใช้ทำงานตัวแอปพลิเคชัน [5]

### 2.3.9 Model-view-viewmodel

เป็น software architectural pattern รูปแบบที่ช่วยแยกการพัฒนาแอปพลิเคชันออกเป็น 3 ส่วน โดยแบ่งออกจากกันอย่างชัดเจนเพื่อให้ง่ายต่อการจัดการและแสดงกระบวนการทำงานในแต่ละส่วน [6]

- Model คือส่วนที่เป็นโครงสร้างของข้อมูล กำหนดกฎเกณฑ์ของการเก็บข้อมูล และเป็นส่วนที่ไว้สำหรับการจัดการข้อมูลโดยตรง
- View คือส่วนที่ไว้แสดงผลตามที่ผู้ใช้ต้องการ ผู้ใช้สามารถเปลี่ยนแปลงข้อมูลได้
- View model มีหน้าที่เก็บข้อมูลทั้งหมดที่ View ต้องการ โดย View และ View model มีการใช้ Data-binding ซึ่งกันและกัน กล่าวคือถ้าข้อมูลของ View มีการแก้ไข ข้อมูลของ View model จะได้รับการแก้ไขไปด้วย หรือข้อมูลของ View model มีการแก้ไข ข้อมูลของ View จะได้รับการแก้ไขไปด้วย

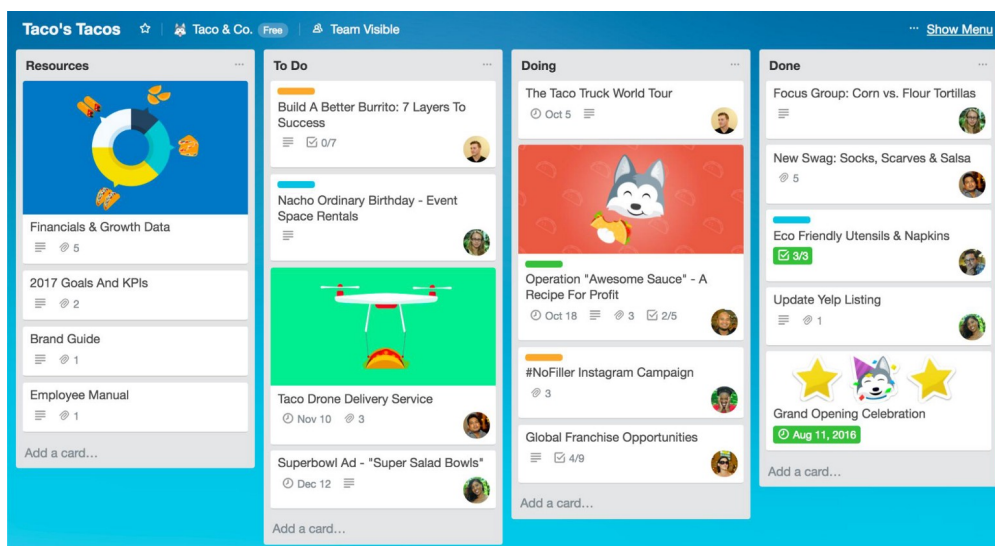


รูปที่ 2.3: แสดง MVVM architecture

## 2.4 เครื่องมือและเทคโนโลยีที่ใช้ในการปฏิบัติงาน

### 2.4.1 Trello

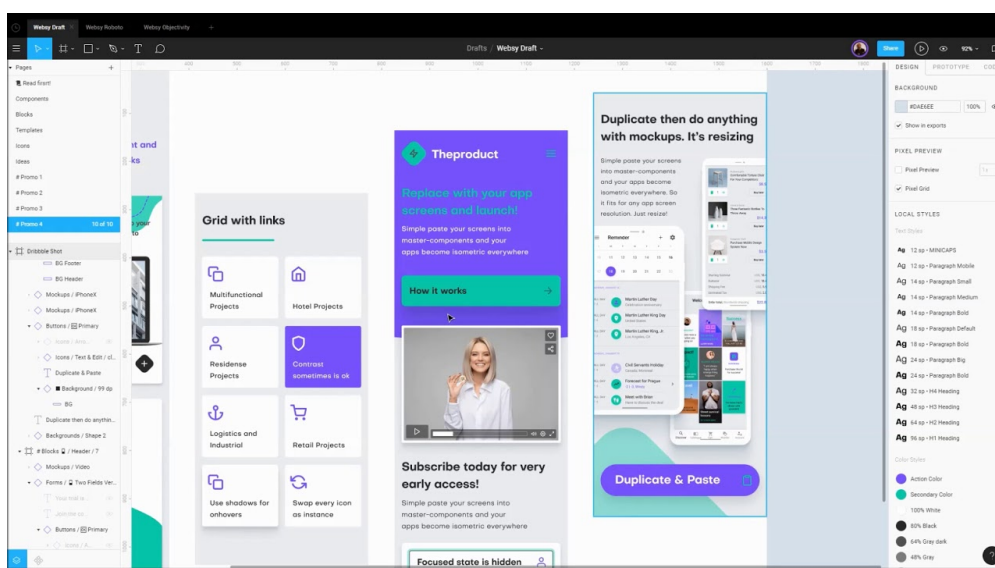
เป็นเว็บแอปพลิเคชันที่ช่วยจัดการบริหารโปรเจกต์ ให้เป็นระเบียบได้อย่างเรียบง่ายเปรียบเสมือนกระดานไว้วางแผนการทำงาน ระบุรายละเอียดของงานแต่ละงาน โดยเว็บแอปพลิเคชันนี้สามารถใช้งานร่วมกันภายในทีมได้ ทำให้การทำงานต่างๆในโปรเจกต์มีความคล่องตัวมากยิ่งขึ้น



รูปที่ 2.4: แสดงตัวอย่างการใช้งาน Trello

### 2.4.2 Figma

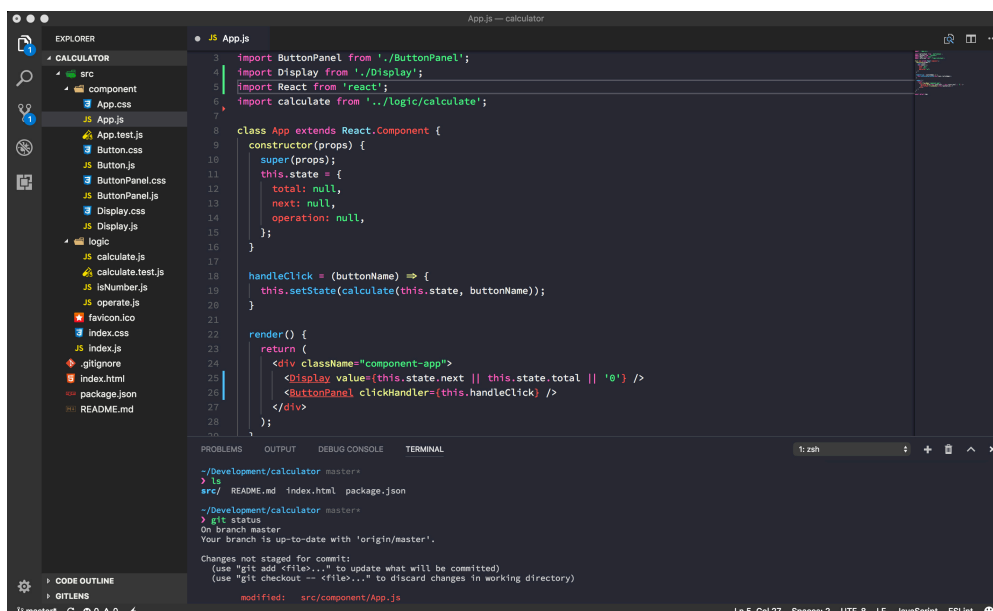
เป็นเครื่องมือที่นำมาใช้ในการออกแบบ prototype ของงานที่ทำ โดยสามารถใช้งานแอปพลิเคชัน Figma ได้ในทุกแพลตฟอร์ม



รูปที่ 2.5: แสดงตัวอย่างการใช้งาน Figma

### 2.4.3 Visual Studio Code

เป็นโปรแกรม code editor ที่ใช้ในการเขียน แก้ไข และปรับแต่งโค้ด โดยพัฒนาออกมาในรูปแบบ OpenSource โดยมีความสามารถในการเปลี่ยนสี Syntax ของโค้ด ตามแต่ละภาษาเพื่อให้ง่ายต่อการมอง และการเขียนโค้ด ซึ่งสนับสนุนภาษามากมาย ยกตัวอย่างเช่น ภาษา C++, Java, JavaScript, PHP, Python เป็นต้น และยังสามารถติดตั้งส่วนขยายที่เพิ่มความสะดวกสบายในการเขียนโค้ด ตามที่เราต้องการได้



รูปที่ 2.6: แสดงตัวอย่างการใช้งาน vscode

### 2.4.4 Vue.js

เป็น JavaScript Framework ที่สร้างส่วนต่อประสานกับผู้ใช้ (UI) และเป็น single-page applications คือ web application ที่ทำการโหลด page เพียงครั้งเดียวแต่สามารถเรียกข้อมูลอื่นๆแบบ dynamic ได้ ทำให้ประสบการณ์การใช้งานเว็บใกล้เคียง native app มากยิ่งขึ้น

### 2.4.5 Vuex

เป็น state management pattern ที่ผสมผสานกับ library เพื่อใช้สำหรับจัดการ state เพื่อให้ data flow ของโปรเจกต์ไปในทิศทางเดียวกัน code จึงมีความเป็นระบบมากยิ่งขึ้น และลดการเขียน code ที่ซ้ำซ้อน ทำให้การทำงานร่วมกันภายในทีมสะดวกมากยิ่งขึ้น เช่นการให้คนในทีมรับผิดชอบแต่ละ module [7]

### 2.4.6 Node.js

เป็น JavaScript runtime กล่าวคือเป็นตัวที่ทำให้ JavaScript สามารถใช้งานในส่วนของ backend หรือเซิร์ฟเวอร์ได้

### 2.4.7 Express.js

เป็น Node.js web application framework ซึ่งมีฟีเจอร์ต่างๆที่ช่วยให้พัฒนาเซิร์ฟเวอร์ด้วย Node.js ได้สะดวกมากยิ่งขึ้น เช่นการจัดการ request และ response

### 2.4.8 Git

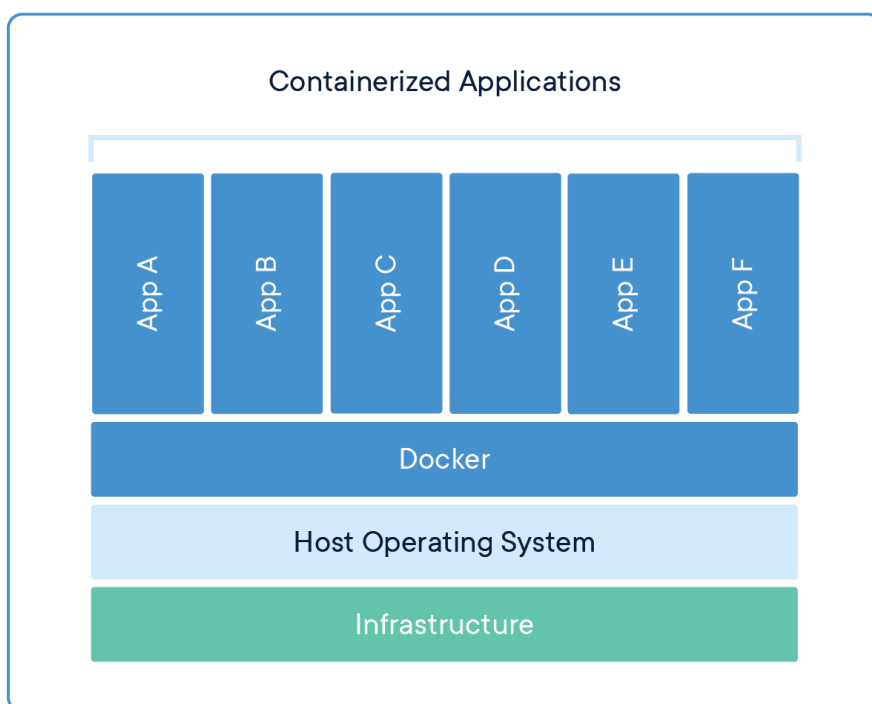
เป็น Version control ที่เอาไว้ติดตาม และควบคุมการเปลี่ยนแปลงของโค้ดเพื่อให้นักพัฒนา สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพ [8]

### 2.4.9 Github

เป็น website Git (version control repository) สำหรับนักพัฒนาซอฟต์แวร์ ใช้หลักการทำงานของ git แต่สามารถใช้งานร่วมกับผู้อื่นได้ผ่านอินเทอร์เน็ต

### 2.4.10 Docker

เป็น engine ที่มีการจำลองสภาพแวดล้อมขึ้นมาบนเครื่องเซิร์ฟเวอร์เพื่อทำการนำ service ที่ต้องการมาทำงานอยู่บนเซิร์ฟเวอร์ โดยใช้หลักการ container ในการจำลองสภาพแวดล้อมขึ้นมาทำงาน service ของเราโดยที่ไม่ต้องมีการนำ os เข้ามาใช้ ทำให้ service ทำงานในสภาพแวดล้อมใดก็ได้ [5]

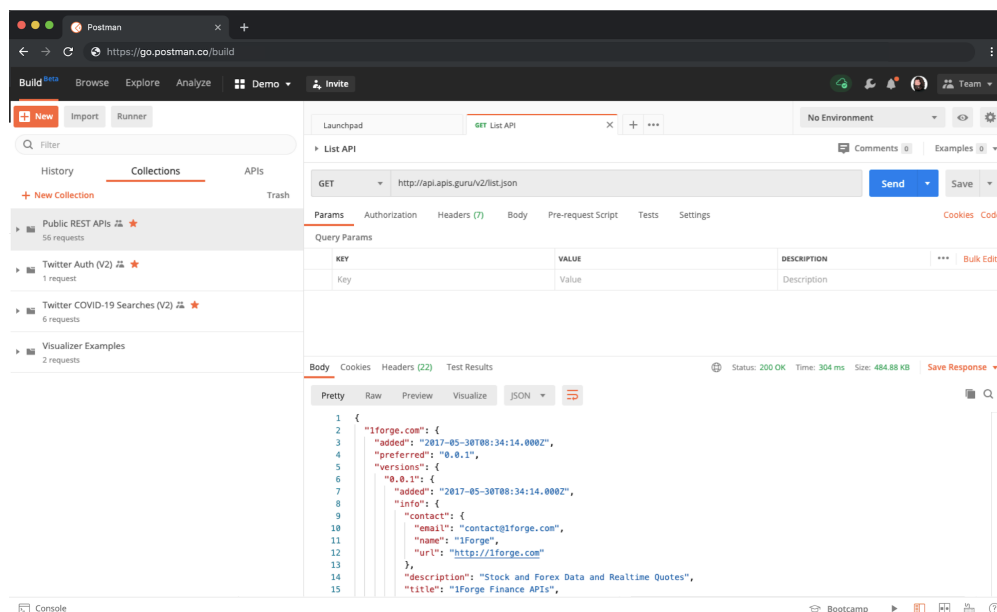


รูปที่ 2.7: แสดงตัวอย่างการทำงานของ docker

### 2.4.11 Postman

เป็นเครื่องมือที่มาช่วยในการ API เพื่อทดสอบการทำงานของ Service โดยมีหน้าต่างส่วนต่อประสานกับผู้ใช้(UI) ที่สวยและใช้งานง่าย





รูปที่ 2.8: แสดงตัวอย่างการใช้งาน postman

## 2.4.12 JWT (Json Web Token)

เป็นมาตรฐาน RFC 7519 ในการยืนยันตัวตน (Authentication) ที่เข้ามาแก้ปัญหาการส่งข้อมูลระหว่างกันในรูปแบบดั้งเดิมคือ Server Based Authentication ที่เปลืองทรัพยากรในการเก็บ Session ID และไม่รองรับการขยายตัว (Scalability) โดย JWT นั้นสามารถเก็บข้อมูลภายในตัวได้ และมีขนาดที่กระทัดรัด เพื่อนำมาใช้กับ Single Page Web Application (SPA) [9] โดย JWT แบ่งโครงสร้างออกเป็น 3 ส่วน

1. Header เก็บประเภทของ token
2. Payload เก็บข้อมูล
3. Signature เป็นลายเซ็นที่อยู่ในรูปแบบของอิเล็กทรอนิกส์ (Digital Signed) เพื่อเช็คว่าเป็น token ที่ถูกสร้างอย่างถูกต้องหรือไม่

# JWT TOKEN



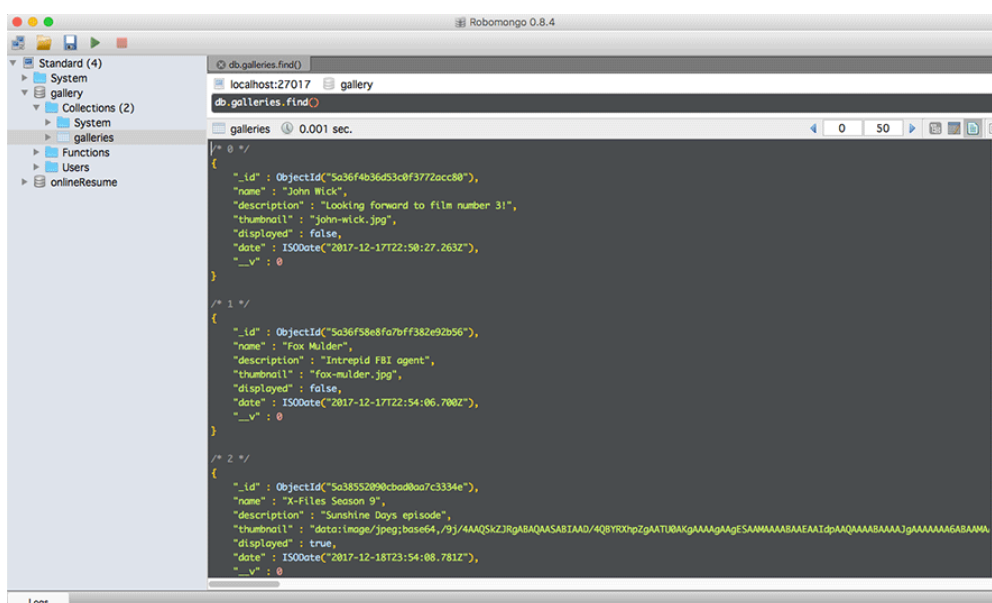
รูปที่ 2.9: แสดงตัวอย่าง jwt encoded

### 2.4.13 MongoDB

เป็น open-source document database ประเภทหนึ่ง โดยเก็บข้อมูลแบบ NoSQL Database และเก็บข้อมูลในรูปแบบของ JSON (JavaScript Object Notation) ซึ่งเก็บเป็น key และ value ซึ่งมีสมรรถภาพสูงกว่าการเก็บด้วยโครงสร้างแบบแถวและหลัก(row/column) แบบดั้งเดิม [10]

### 2.4.14 Robo 3T

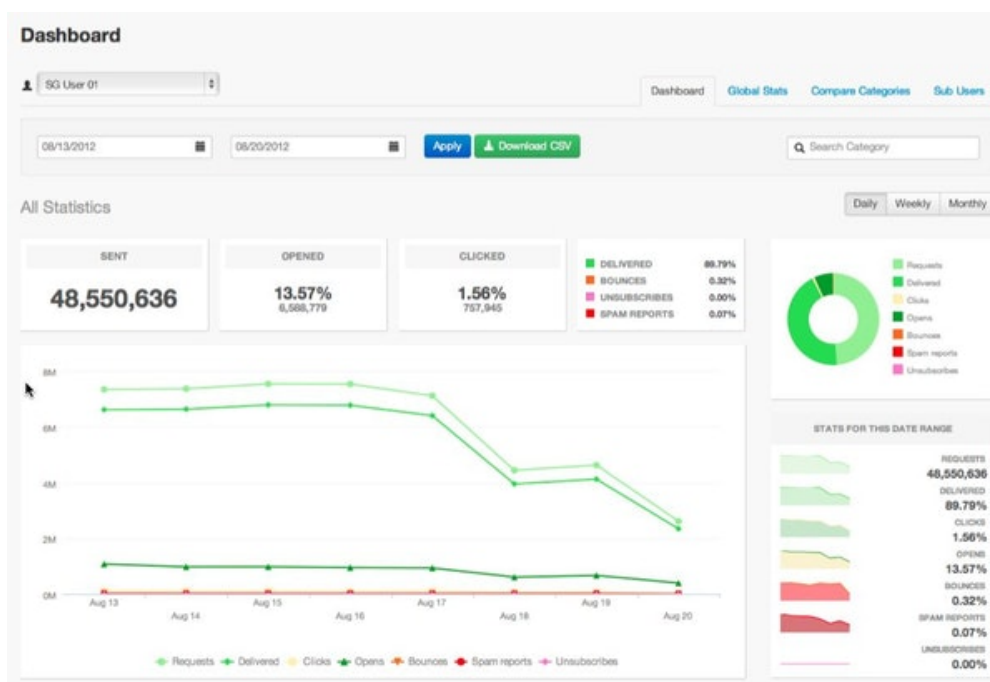
เป็นการใช้ภาพเป็นตัวประสานกับผู้ใช้ของ(GUI) ของ MongoDB ช่วยทำให้ใช้งาน MongoDB ได้สะดวกมากยิ่งขึ้น เช่น เขียน SQL เพื่อ query ข้อมูลใน MongoDB หรือ Import และ export ไฟล์ข้อมูลเป็น CSV, JSON, SQL and BSON/mongodump ได้ [?]



รูปที่ 2.10: แสดงตัวอย่างการใช้งาน Robo 3T

### 2.4.15 SendGrid

เป็น API ช่วยในการส่งอีเมลให้ผู้อื่น สามารถตรวจสอบอีเมลที่ส่งว่าส่งไปถึงหรือไม่ มีปัญหาอะไรเกิดขึ้นหรือเปล่า และมีการเก็บสถิติข้อมูลอีเมลที่ส่ง สรุปออกมาแสดงให้วิเคราะห์เช่น มีการเปิดเมลอ่านกี่ครั้ง มีการรายงานว่าเป็นสแปมกี่ครั้ง



รูปที่ 2.11: แสดงตัวอย่างสถิติของอีเมลที่ทำการส่งด้วย sendgrid

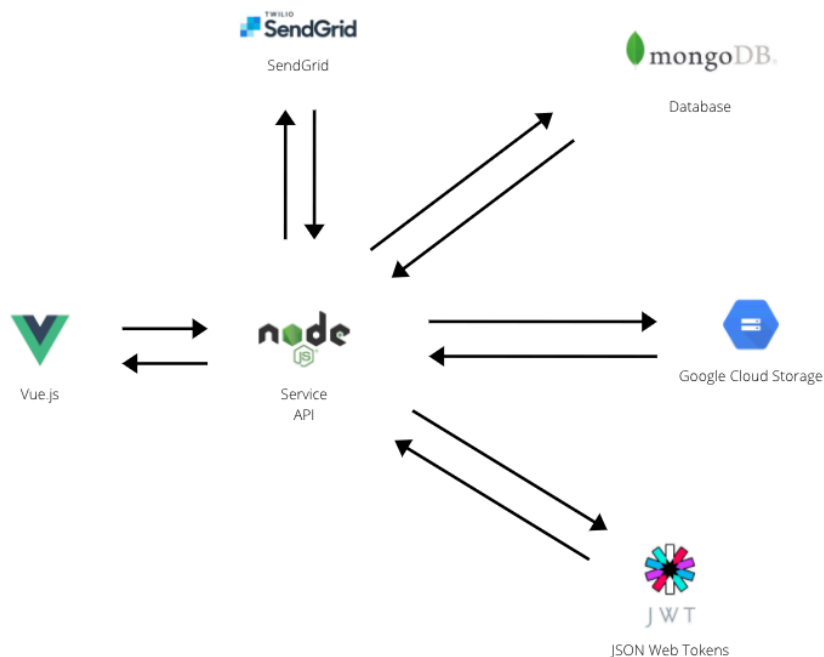
## 2.5 ลักษณะขั้นตอนการทำงาน

## บทที่ 3

# การออกแบบระบบและรายละเอียดการพัฒนา

### 3.1 ภาพรวมของเว็บแอปพลิเคชัน

เว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน คือระบบที่จะช่วยให้คัดกรองผู้คนได้มีประสิทธิภาพมากยิ่งขึ้น ผ่านการทำข้อสอบที่สามารถกำหนดระดับความยากของข้อสอบได้ และดึงคลังคำถามมาแบบสุ่ม เพื่อส่งให้ผู้สมัครงานโดยอัตโนมัติผ่านทางอีเมล เมื่อผู้สมัครงานทำข้อสอบเสร็จแล้ว ข้อสอบจะถูกส่งไปที่ผู้ออก พร้อมตรวจข้อที่เป็นคำถามปรนัยให้อัตโนมัติ โดยระบบนี้แบ่งเป็น 2 ส่วนหลักๆ คือส่วนต่อประสานกับผู้ใช้ที่พัฒนาด้วย Vue.js framework และส่วนที่จัดการกับฐานข้อมูล พัฒนาด้วย Node.js framework โดยการเก็บข้อมูลทั่วไปถูกเก็บอยู่ใน mongoDB และการเก็บไฟล์ เช่น ไฟล์รูปภาพ จะถูกเก็บอยู่บน Google Cloud Storage การจัดการอีเมลใช้บริการของ SendGrid เข้ามาช่วยจัดการซึ่งสามารถวิเคราะห์การส่งอีเมลในแต่ละครั้งได้ และการยืนยันตัวตนจะถูกเข้ารหัสด้วย JWSN Web Token ภาพรวมการทำงานของระบบจะเป็นดังรูป 3.1



รูปที่ 3.1: แสดงการทำงานของเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน

## 3.2 วิเคราะห์ความต้องการ

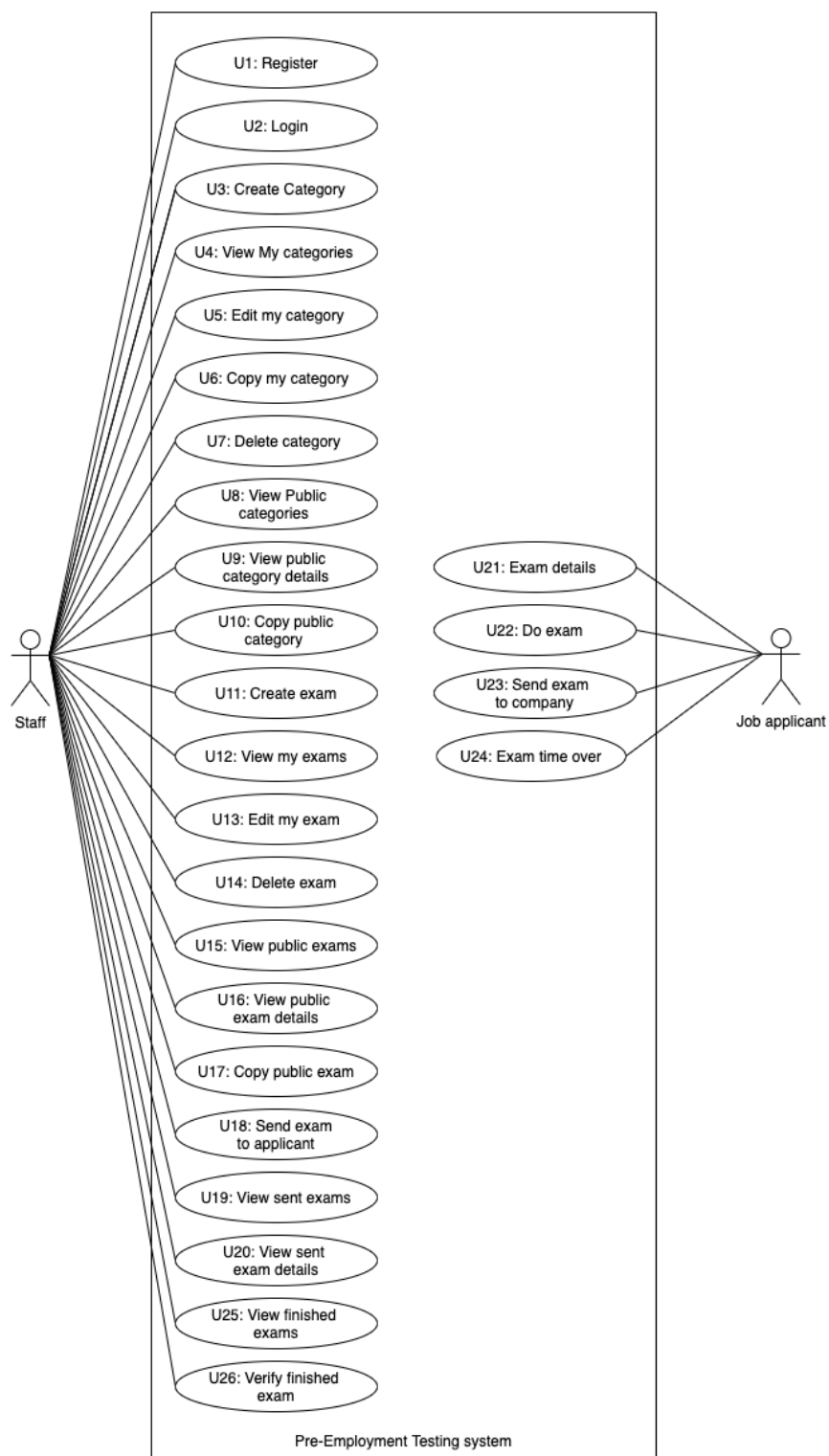
### 3.2.1 ความต้องการที่เป็นหน้าที่หลักของระบบ (Functional Requirement)

1. ผู้ใช้งานสามารถสามารถลงทะเบียนเป็นสมาชิกเพื่อใช้งานเว็บแอปพลิเคชันได้
2. ผู้ใช้ต้องยืนยันตัวตนผ่านอีเมลหลังจากทำการลงทะเบียน ถึงจะเข้าใช้งานระบบได้
3. ผู้ใช้งานสามารถสร้าง, แก้ไข และลบคำถามในหมวดหมู่คำถามได้
4. ระบบสามารถป้องกันการรีเฟรชเพจ หรือการเปลี่ยนหน้าได้ หากผู้ใช้กำลังสร้างหรือแก้ไขคำถามในหมวดหมู่คำถาม
5. ผู้ใช้งานสามารถยกเลิกการสร้างหมวดหมู่คำถามได้
6. ผู้ใช้งานไม่สามารถลบหมวดหมู่คำถามได้ หากหมวดหมู่นั้นถูกใช้งานอยู่ในข้อสอบ
7. ผู้ใช้งานสามารถสร้างคำถามประเภท ปรนัย, อัตนัย และคำถามที่มีไฟล์แนบได้
8. คำถามที่เป็นปรนัย สามารถเลือกให้มีข้อถูกมากกว่า 1 ข้อได้
9. สามารถใส่รูปภาพเพิ่มในตัวเลือกคำตอบของคำถามประเภทปรนัยได้
10. หากผู้ใช้งาน ส่งข้อสอบให้ผู้สมัครงาน ทำแล้ว ถ้าหมวดหมู่คำถามนั้น ถูกใช้งานในข้อสอบ คำถามในหมวดหมู่นั้นจะสามารถแก้ไขได้เฉพาะเฉลยเท่านั้น ไม่สามารถแก้ไขคำถามได้ โดยหากต้องการแก้ไขให้ทำการคัดลอกหมวดหมู่คำถามของตนเองมาแก้ไขเป็นหมวดหมู่ใหม่
11. ผู้ใช้งานสามารถคัดลอกหมวดหมู่คำถามของตนเองได้
12. ผู้ใช้งานสามารถเปิดหมวดหมู่คำถามให้เป็นสาธารณะได้
13. หมวดหมู่คำถามที่เป็นสาธารณะ ผู้ที่ไม่ใช่เจ้าของไม่สามารถแก้ไขได้ โดยผู้ใช้งานสามารถคัดลอกหมวดหมู่คำถามที่เป็นสาธารณะไปเป็นเป็นของตนเองได้ หากผู้ใช้ไม่ใช่เจ้าของแต่ต้องการแก้ไข
14. ผู้ใช้งานสามารถค้นหาหมวดหมู่คำถามได้
15. ผู้ใช้งานสามารถสร้างข้อสอบ ที่กำหนดหมวดหมู่คำถามที่ต้องการ, จำนวนข้อ และความยากได้ โดยข้อสอบจะทำการสุ่มเมื่อผู้ใช้ส่งให้ผู้ทำข้อสอบ
16. ระบบสามารถป้องกันการรีเฟรชเพจ หรือการเปลี่ยนหน้าได้ หากผู้ใช้กำลังสร้างหรือแก้ไขข้อสอบ
17. ผู้ใช้สามารถลบแก้ไขหมวดหมู่คำถามที่ใช้, จำนวนข้อ และระดับความยากของข้อสอบได้
18. ผู้ใช้สามารถค้นหาข้อสอบได้
19. ผู้ใช้สามารถเปิดข้อสอบให้เป็นสาธารณะได้

20. ข้อสอบที่เป็นสาธารณะ ผู้ที่ไม่ใช่เจ้าของไม่สามารถแก้ไขได้ โดยผู้ใช้สามารถคัดลอกข้อสอบสาธารณะไปเป็นเป็นของตนเองได้ หากผู้ใช้ไม่ใช่เจ้าของแต่ต้องการแก้ไข
21. หากผู้ใช้งานส่งข้อสอบให้ผู้สมัครงานทำแล้ว ข้อสอบที่ผู้ใช้งานสร้างขึ้นนั้นจะไม่สามารถแก้ไขได้ แต่สามารถส่งให้ผู้สมัครงานคนอื่นทำต่อไปได้ โดยระบบจะทำการสุ่มคำถามที่ผู้ออกข้อสอบกำหนด จำนวนข้อประเภทคำถาม และระดับความยาก
22. ผู้ใช้งานสามารถดูประวัติที่ผู้ใช้ทำการส่งข้อสอบให้ผู้สมัครงานได้
23. ผู้ใช้สามารถกำหนดระยะเวลาการทำข้อสอบได้
24. ผู้ใช้สามารถกำหนดวันหมดอายุของข้อสอบได้
25. ผู้ใช้สามารถส่งข้อสอบให้ผู้สมัครงานผ่านอีเมลได้
26. ผู้ทำข้อสอบสามารถเข้ามาทำข้อสอบในเว็บแอปพลิเคชันได้ตลอด หากเวลาในการทำข้อสอบยังไม่หมด
27. เมื่อผู้ทำข้อสอบทำการส่งข้อสอบแล้ว ระบบสามารถตรวจข้อสอบที่เป็นปรนัยได้
28. มีการแสดงสถานะบอกว่าผู้ใช้งาน ตรวจข้อสอบนั้นหรือยัง
29. มีการสรุปคะแนนรวมข้อผู้ทำข้อสอบ

### 3.3 การวิเคราะห์และออกแบบระบบ

#### 3.3.1 แผนภาพยูสเคส (Use Case Diagram)



รูปที่ 3.2: แผนภาพยูสเคสของระบบ

### 3.3.2 รายละเอียดการทำงานในแต่ละยูสเคส (Use Case Description)

#### 3.3.2.1 รายละเอียดยูสเคส ลงทะเบียน

Use Case No:	1	
Use Case Name:	ลงทะเบียน	
Use Case Scenario:	ผู้ลงทะเบียนกับระบบ	
Triggering Event:	เมื่อผู้ใช้เข้าใช้งานเป็นครั้งแรก	
Brief Description:	สำหรับให้ผู้ใช้ลงทะเบียนเข้าสู่ระบบ	
Actors:	พนักงาน	
Related Use Cases:	-	
Stakeholders:	-	
Pre - Conditions:	-	
Post - Conditions:	ผู้ใช้สามารถเข้าใช้งานระบบได้	
Flow of Events:	ผู้ใช้	ระบบ
	1. ผู้ใช้เลือกลงทะเบียน  3. ผู้ใช้ทำการกรอกชื่อ-นามสกุล, อีเมล, รหัสผ่าน และยืนยันรหัสผ่าน แล้วทำการคลิกลงทะเบียน  5. ผู้ใช้งานได้รับอีเมล แล้วทำการคลิกลิงค์เว็บไซต์	2. ระบบแสดงแบบฟอร์มให้กรอกข้อมูล  4. ระบบทำการส่งอีเมลให้ผู้ใช้เพื่อทำการยืนยันอีเมล  6. ระบบทำการยืนยันอีเมลแล้วแสดงหน้าจอหลักของผู้ใช้งาน
Exception Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้กรอกรายละเอียดไม่ครบถ้วน หรือไม่ตรงเงื่อนไข</li> <li>- ผู้ใช้ไม่ได้ทำการยืนยันอีเมล</li> <li>- ผู้ใช้กรอกอีเมลที่ไม่มีอยู่จริง</li> <li>- ผู้ใช้กรอกอีเมลที่ผู้ใช้งานไม่สามารถเข้าใช้อีเมลนั้นได้</li> </ul>	

ตารางที่ 3.1: รายละเอียดยูสเคส ลงทะเบียน



### 3.3.2.2 รายละเอียดยูสเคส เข้าสู่ระบบ

Use Case No:	2	
Use Case Name:	เข้าสู่ระบบ	
Use Case Scenario:	ผู้ใช้งานต้องการเข้าสู่ระบบ	
Triggering Event:	<ul style="list-style-type: none"> <li>- เมื่อผู้ใช้งานต้องการเข้าใช้งานระบบ</li> <li>- ผู้ใช้เข้าเว็บแอปพลิเคชันโดยที่ยังไม่เข้าใช้งานระบบ</li> </ul>	
Brief Description:	สำหรับให้ผู้ใช้งานเข้าสู่ระบบเพื่อเข้าใช้งานเว็บแอปพลิเคชัน	
Actors:	พนักงาน	
Related Use Cases:	ลงทะเบียน	
Stakeholders:	-	
Pre - Conditions:	ผู้ใช้งานต้องการลงทะเบียนและยืนยันตัวตนผ่านอีเมลเรียบร้อยแล้ว	
Post - Conditions:	ระบบแสดงหน้าจอสำหรับผู้ใช้งาน	
Flow of Events:	ผู้ใช้	ระบบ
	1. ผู้ใช้เข้าหน้าเว็บโดยที่ยังไม่ได้ทำการเข้าสู่ระบบ  3. ผู้ใช้ทำการกรอกอีเมล และรหัสผ่านแล้วทำการคลิกเข้าสู่ระบบ หรือ กด Enter บนแป้นพิมพ์	2. ระบบแสดงแบบฟอร์มให้กรอกข้อมูล  4. ระบบ แสดง หน้า จอ หลักของผู้ใช้งาน
Exception Conditions:	ผู้ใช้งานกรอกอีเมล หรือรหัสผ่านผิด	

ตารางที่ 3.2: รายละเอียดยูสเคส เข้าสู่ระบบ

### 3.3.2.3 รายละเอียดยูสเคส สร้างหมวดหมู่คำถาม

Use Case No:	3	
Use Case Name:	สร้างหมวดหมู่คำถาม	
Use Case Scenario:	ผู้ใช้งานต้องการสร้างหมวดหมู่คำถาม	
Triggering Event:	เมื่อผู้ใช้งานต้องการสร้างหมวดหมู่คำถาม ไว้ใช้ในข้อสอบของตนเอง หรือให้พนักงานคนอื่นใช้	
Brief Description:	สำหรับให้ผู้ใช้งาน ออกข้อสอบโดยแบ่งแยกตามแต่ละหมวดหมู่ เช่น คณิตศาสตร์, โปรแกรมมิ่ง หรือ แบบทดสอบสติปัญญา โดยมีชนิดคำถามแบ่งออกเป็น อัตนัย, ปรนัย และคำถามที่มีไฟล์แนบ	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ	
Stakeholders:	-	
Pre - Conditions:	ผู้ใช้งานต้องการเข้าสู่ระบบ	
Post - Conditions:	ผู้ใช้งานมีหมวดหมู่คำถามที่ตนเองสร้างขึ้น และแสดงหน้าหมวดหมู่ทั้งหมดของผู้ใช้	
Flow of Events:	ผู้ใช้งาน	ระบบ
	1.ไปที่หน้าสร้างหมวดหมู่  3.ผู้ใช้งานกรอกชื่อหมวดหมู่ และกรอกหัวเรื่อง  4.สร้างคำถามภายใต้ หมวดหมู่นั้นโดยสามารถเลือกระดับความยาก และประเภทของคำถาม แล้วทำการคลิกสร้างหมวดหมู่	2.ระบบแสดงหน้าสร้างหมวดหมู่คำถาม          5.ระบบทำการสร้างหมวดหมู่คำถาม แล้วแสดงหน้าหมวดหมู่คำถามทั้งหมดของผู้ใช้
Exception Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้งานกรอกชื่อหมวดหมู่ซ้ำกับ หมวดหมู่ที่ตนเองมีอยู่แล้ว</li> <li>- ผู้ใช้งานกรอกข้อมูลไม่ครบ ตามฟอร์มที่ให้กรอก</li> <li>- ไฟล์ที่แนบมามีขนาดใหญ่เกินที่กำหนด หรือประเภทของไฟล์ไม่ถูกต้อง</li> </ul>	

ตารางที่ 3.3: รายละเอียดยูสเคส สร้างหมวดหมู่คำถาม

### 3.3.2.4 รายละเอียดยูสเคส ดูหมวดหมู่คำถามของผู้ใช้

Use Case No:	4	
Use Case Name:	ดูหมวดหมู่คำถามของผู้ใช้	
Use Case Scenario:	ผู้ใช้งานต้องการดูหมวดหมู่คำถามของตนเอง	
Triggering Event:	<ul style="list-style-type: none"> <li>- เมื่อผู้ใช้งานต้องการดูหมวดหมู่คำถามของตนเอง</li> <li>- เมื่อผู้ใช้งานสร้างประเภทของข้อสอบสำเร็จ หน้าเว็บจะทำการไปแสดงผลที่หน้าหมวดหมู่คำถามทั้งหมดของผู้ใช้</li> </ul>	
Brief Description:	สำหรับให้ผู้ใช้งานดูหมวดหมู่คำถามที่ตนเองสร้างขึ้น หรือหมวดหมู่คำถามที่คัดลอกมาจากหมวดหมู่สาธารณะ	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, สร้างหมวดหมู่คำถาม, คัดลอกหมวดหมู่คำถามจากหมวดหมู่คำถามสาธารณะ	
Stakeholders:	-	
Pre - Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้ต้องทำการเข้าสู่ระบบ</li> <li>- สร้างหมวดหมู่คำถาม หรือคัดลอกหมวดหมู่คำถามจากหมวดหมู่คำถามสาธารณะ</li> </ul>	
Post - Conditions:	แสดงหน้าหมวดหมู่ทั้งหมดของผู้ใช้	
Flow of Events:	ผู้ใช้งาน	ระบบ
	1.ไปที่หน้าดูหมวดหมู่คำถามของผู้ใช้	2.ระบบแสดงหมวดหมู่ทั้งหมดของผู้ใช้
Exception Conditions:	ผู้ใช้งานไม่มีหมวดหมู่คำถาม	

ตารางที่ 3.4: รายละเอียดยูสเคส ดูหมวดหมู่คำถามของผู้ใช้

### 3.3.2.5 รายละเอียดยูสเคส แก้ไขหมวดหมู่คำถามของผู้ใช้

Use Case No:	5	
Use Case Name:	แก้ไขหมวดหมู่คำถามของผู้ใช้	
Use Case Scenario:	ผู้ใช้งานต้องการแก้ไขหมวดหมู่คำถามของตนเอง	
Triggering Event:	เมื่อผู้ใช้งานต้องการแก้ไขรายละเอียดในหมวดหมู่คำถาม เช่น เพิ่มคำถาม, แก้ไขคำถาม หรือแก้ไขชื่อหมวดหมู่	
Brief Description:	สำหรับให้ผู้ใช้งานแก้ไขรายละเอียดต่างๆในหมวดหมู่คำถาม	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, สร้างหมวดหมู่คำถาม, คัดลอกหมวดหมู่คำถามจากหมวดหมู่คำถามสาธารณะ	
Stakeholders:	-	
Pre - Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้ต้องทำการเข้าสู่ระบบ</li> <li>- สร้างหมวดหมู่คำถาม หรือคัดลอกหมวดหมู่คำถามจากหมวดหมู่คำถามสาธารณะ</li> </ul>	
Post - Conditions:	แสดงหน้าหมวดหมู่ทั้งหมดของผู้ใช้	
Flow of Events:	<div>ผู้ใช้งาน</div> <div> 1.ไปที่หน้าหมวดหมู่คำถามของผู้ใช้   3.คลิกที่หมวดหมู่คำถามที่ต้องการแก้ไข   5.ผู้ใช้งานทำการแก้ไขข้อมูลต่างๆในประเภทที่ต้องการแล้วทำการคลิกยืนยันการแก้ไข </div>	<div>ระบบ</div> <div> 2.ระบบแสดงหมวดหมู่ทั้งหมดของผู้ใช้   4.ระบบแสดงหน้าแก้ไขหมวดหมู่นั้นๆ   6.ระบบทำการแก้ไขข้อมูล  7.ระบบแสดงหน้าหมวดหมู่ทั้งหมดของผู้ใช้ </div>
Exception Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้กรอกชื่อหมวดหมู่ซ้ำกับ หมวดหมู่ที่ตนเองมีอยู่แล้ว</li> <li>- ผู้ใช้กรอกรายละเอียดไม่ครบถ้วน หรือไม่ตรงเงื่อนไข</li> <li>- ไฟล์ที่แนบมามีขนาดใหญ่เกินที่กำหนด หรือประเภทของไฟล์ไม่ถูกต้อง</li> <li>- สามารถแก้ไขได้เฉพาะคำตอบ หากข้อสอบเคยถูกใช้ไปแล้ว</li> </ul>	

ตารางที่ 3.5: รายละเอียดยูสเคส แก้ไขหมวดหมู่คำถามของผู้ใช้

### 3.3.2.6 รายละเอียดยูสเคส คัดลอกหมวดหมู่คำถามของผู้ใช้

Use Case No:	6	
Use Case Name:	คัดลอกหมวดหมู่คำถามของผู้ใช้	
Use Case Scenario:	ผู้ใช้งานต้องการคัดลอกหมวดหมู่คำถามของตนเอง	
Triggering Event:	เมื่อผู้ใช้งานต้องการแก้ไขหมวดหมู่คำถามที่ถูกส่งให้ผู้สมัครงานแล้ว ทำให้ไม่สามารถแก้ไขหมวดหมู่เดิมได้ หรือต้องการคัดลอกหมวดหมู่มาแก้ไข โดยที่ไม่ต้องการให้กระทบหมวดหมู่เดิม	
Brief Description:	สำหรับให้ผู้ใช้งานคัดลอกหมวดหมู่คำถามของตนเอง	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, สร้างหมวดหมู่คำถาม, ส่งข้อสอบให้ผู้สมัคร	
Stakeholders:	-	
Pre - Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้ต้องทำการเข้าสู่ระบบ</li> <li>- สร้างหมวดหมู่คำถาม หรือคัดลอกหมวดหมู่คำถามจากหมวดหมู่คำถามสาธารณะ</li> </ul>	
Post - Conditions:	ระบบทำการคัดลอกหมวดหมู่คำถามที่ผู้ใช้งานต้องการ	
Flow of Events:	ผู้ใช้	ระบบ
	1.ไปที่หน้าคู่มือหมวดหมู่คำถามของผู้ใช้  3.คลิกปุ่มคัดลอกที่หมวดหมู่ที่ต้องการคัดลอก  5.ผู้ใช้งานกดยืนยันการคัดลอก	2.ระบบแสดงหมวดหมู่ทั้งหมดของผู้ใช้  4.ระบบแสดงหน้ายืนยันการคัดลอก  6.ระบบทำการคัดลอกหมวดหมู่คำถามนั้น
Exception Conditions:	ผู้ใช้งานไม่มีหมวดหมู่คำถาม	

ตารางที่ 3.6: รายละเอียดยูสเคส คัดลอกหมวดหมู่คำถามของผู้ใช้

### 3.3.2.7 รายละเอียดยูสเคส ลบหมวดหมู่คำถามของผู้ใช้

Use Case No:	7	
Use Case Name:	ลบหมวดหมู่คำถามของผู้ใช้	
Use Case Scenario:	ผู้ใช้งานต้องการลบหมวดหมู่คำถามของตนเอง	
Triggering Event:	เมื่อผู้ใช้งานไม่ต้องการหมวดหมู่คำถามนั้นๆแล้ว	
Brief Description:	สำหรับให้ผู้ใช้งานลบหมวดหมู่คำถามออกจากหมวดหมู่คำถามของตนเอง	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, สร้างหมวดหมู่คำถาม, คัดลอกหมวดหมู่คำถามจากหมวดหมู่คำถามสาธารณะ, ข้อสอบของผู้ใช้	
Stakeholders:	-	
Pre - Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้ต้องทำการเข้าสู่ระบบ</li> <li>- สร้างหมวดหมู่คำถาม หรือคัดลอกหมวดหมู่คำถามจากหมวดหมู่สาธารณะ</li> </ul>	
Post - Conditions:	ระบบลบหมวดหมู่คำถามของผู้ใช้	
Flow of Events:	ผู้ใช้งาน	ระบบ
	1.ไปที่หน้าดูหมวดหมู่คำถามของผู้ใช้  3.คลิกที่รูปถึงขยะของหมวดหมู่คำถามที่ต้องการลบ  5.ผู้ใช้งานทำการยืนยันการลบ	2.ระบบแสดงหมวดหมู่ทั้งหมดของผู้ใช้  4.ระบบทำการแสดงหน้าต่างข้อความยืนยันการลบ  6.ระบบทำการลบหมวดหมู่คำถามที่ผู้ใช้งานต้องการลบ
Exception Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้ไม่ทำการยืนยันการลบ</li> <li>- หมวดหมู่คำถามที่ต้องการลบ ถูกใช้งานอยู่ในข้อสอบ ทำให้ไม่สามารถลบได้</li> </ul>	

ตารางที่ 3.7: รายละเอียดยูสเคส ลบหมวดหมู่คำถามของผู้ใช้

### 3.3.2.8 รายละเอียดยูสเคส ดูหมวดหมู่คำถามสาธารณะ

Use Case No:	8	
Use Case Name:	ดูหมวดหมู่คำถามสาธารณะ	
Use Case Scenario:	ผู้ใช้งานต้องการดูหมวดหมู่คำถามสาธารณะ	
Triggering Event:	<ul style="list-style-type: none"> <li>- เมื่อผู้ใช้งานต้องการดูหมวดหมู่คำถามสาธารณะ</li> <li>- เมื่อผู้ใช้งานต้องการนำหมวดหมู่คำถามสาธารณะไปใช้ในข้อสอบของตนเอง</li> </ul>	
Brief Description:	สำหรับให้ผู้ใช้งานดูว่ามีหมวดหมู่คำถามสาธารณะใดบ้างที่อยู่ในระบบและสามารถคัดลอกไปใช้ได้	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ	
Stakeholders:	-	
Pre - Conditions:	ผู้ใช้งานต้องการเข้าสู่ระบบ	
Post - Conditions:	ระบบแสดงหมวดหมู่ข้อสอบทั้งหมด ที่เป็นสาธารณะ	
Flow of Events:	ผู้ใช้งาน	ระบบ
	1.ไปที่หน้าดูหมวดหมู่คำถามสาธารณะ	2.ระบบแสดงหมวดหมู่ทั้งหมดที่เป็นสาธารณะ
Exception Conditions:	ไม่มีหมวดหมู่คำถามที่เป็นสาธารณะอยู่ในระบบ	

ตารางที่ 3.8: รายละเอียดยูสเคส ดูหมวดหมู่คำถามสาธารณะ

### 3.3.2.9 รายละเอียดยูสเคส ดูรายละเอียดของหมวดหมู่คำถามสาธารณะ

Use Case No:	9	
Use Case Name:	ดูรายละเอียดของหมวดหมู่คำถามสาธารณะ	
Use Case Scenario:	ผู้ใช้ต้องการดูรายละเอียดในหมวดหมู่คำถามสาธารณะที่ต้องการ	
Triggering Event:	<ul style="list-style-type: none"> <li>- เมื่อผู้ใช้ต้องการดูรายละเอียดของหมวดหมู่คำถามสาธารณะ</li> <li>- เมื่อผู้ใช้ต้องการนำหมวดหมู่คำถามสาธารณะนั้นๆไปใช้ในข้อสอบของตนเอง</li> </ul>	
Brief Description:	สำหรับให้ผู้ใช้ดูรายละเอียดของหมวดหมู่คำถามสาธารณะ ว่าในหมวดหมู่นั้นมีคำถามอะไรบ้าง	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, ดูหมวดหมู่คำถามสาธารณะ	
Stakeholders:	-	
Pre - Conditions:	ผู้ใช้ต้องทำการเข้าสู่ระบบ	
Post - Conditions:	ระบบรายละเอียดของหมวดหมู่ข้อสอบนั้นๆ	
Flow of Events:	ผู้ใช้	ระบบ
	1.ไปที่หน้าดูหมวดหมู่คำถามสาธารณะ  3.คลิกหมวดหมู่ที่ต้องการดูรายละเอียด	2.ระบบแสดงหมวดหมู่ทั้งหมดที่เป็นสาธารณะ  4.ระบบแสดงรายละเอียดของหมวดหมู่คำถามนั้น
Exception Conditions:	ไม่มีหมวดหมู่คำถามที่เป็นสาธารณะอยู่ในระบบ	

ตารางที่ 3.9: รายละเอียดยูสเคส ดูรายละเอียดของหมวดหมู่คำถามสาธารณะ



### 3.3.2.10 รายละเอียดยูสเคส คัดลอกหมวดหมู่คำถามสาธารณะ

Use Case No:	10	
Use Case Name:	คัดลอกหมวดหมู่คำถามสาธารณะ	
Use Case Scenario:	ผู้ใช้งานต้องการนำหมวดหมู่คำถามสาธารณะ	
Triggering Event:	เมื่อผู้ใช้งานต้องการนำหมวดหมู่คำถามสาธารณะนั้นๆ ไปใช้ในข้อสอบของตนเอง	
Brief Description:	สำหรับให้ผู้ใช้งานทำการคัดลอกหมวดหมู่คำถามสาธารณะ ไปใช้ในข้อสอบของตนเอง	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, คู่มือหมวดหมู่คำถามหมวดหมู่คำถามสาธารณะ	
Stakeholders:	-	
Pre - Conditions:	ผู้ใช้งานต้องการเข้าสู่ระบบ	
Post - Conditions:	ระบบทำการคัดลอกหมวดหมู่คำถามสาธารณะที่ผู้ใช้งานต้องการ ไปเก็บในหมวดหมู่คำถามของตนเอง	
Flow of Events:	ผู้ใช้งาน	ระบบ
	1.ไปที่หน้าคู่มือหมวดหมู่คำถามสาธารณะ  3.คลิกปุ่มคัดลอกที่หมวดหมู่ที่ต้องการ	2.ระบบแสดงหมวดหมู่ทั้งหมดที่เป็นสาธารณะ  4.ระบบทำการคัดลอกข้อมูลสาธารณะ ที่ต้องการ ไปเก็บในหมวดหมู่คำถามของตนเอง
Exception Conditions:	ไม่มีหมวดหมู่คำถามที่เป็นสาธารณะอยู่ในระบบ	

ตารางที่ 3.10: รายละเอียดยูสเคส คัดลอกหมวดหมู่คำถามสาธารณะ

### 3.3.2.11 รายละเอียดยูสเคส สร้างข้อสอบ

Use Case No:	11						
Use Case Name:	สร้างข้อสอบ						
Use Case Scenario:	ผู้ใช้งานต้องการสร้างข้อสอบ						
Triggering Event:	เมื่อผู้ใช้งานต้องการสร้างข้อสอบไปของตนเอง เพื่อนำไปทดสอบกับผู้สมัครงาน						
Brief Description:	สำหรับผู้ใช้งานที่ต้องการทำการออกข้อสอบ โดยการดึงคำถามจากประเภทของข้อสอบ ตามจำนวนข้อและความยากที่ต้องการ เช่น ต้องการคำถามหมวดหมู่ คณิตศาสตร์ ข้อระดับง่าย 2 ข้อ ข้อระดับยาก 1 ข้อ และคำถามหมวดหมู่ ภาษาอังกฤษ ข้อระดับปานกลาง 3 ข้อ สามารถกำหนดระยะเวลาในการทำข้อสอบ และวันหมดอายุของข้อสอบได้						
Actors:	พนักงาน						
Related Use Cases:	เข้าสู่ระบบ, สร้างหมวดหมู่คำถาม, คัดลอกหมวดหมู่คำถามสาธารณะ						
Stakeholders:	-						
Pre - Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้ต้องทำการเข้าสู่ระบบ</li> <li>- ผู้ใช้งานต้องสร้างหมวดหมู่คำถามของตนเอง หรือคัดลอกหมวดหมู่คำถามสาธารณะ</li> </ul>						
Post - Conditions:	ระบบทำการสร้างข้อสอบตามที่ผู้ใช้งานกำหนด						
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th><th>ระบบ</th></tr> </thead> <tbody> <tr> <td>1.ไปที่หน้าสร้างข้อสอบ</td><td>2.ระบบแสดงหมวดหมู่ของตนเองสำหรับเลือกใช้ในข้อสอบ และแสดงรายละเอียดให้กรอก</td></tr> <tr> <td>3.เลือกหมวดหมู่คำถามที่ต้องการ แล้วเลือกระดับความยากและจำนวนข้อที่ต้องการในหมวดหมู่นั้นๆ แล้วคลิกสร้างข้อสอบ</td><td>4.ระบบทำการสร้างข้อสอบตามที่ผู้ใช้งานกำหนด</td></tr> </tbody> </table>	ผู้ใช้	ระบบ	1.ไปที่หน้าสร้างข้อสอบ	2.ระบบแสดงหมวดหมู่ของตนเองสำหรับเลือกใช้ในข้อสอบ และแสดงรายละเอียดให้กรอก	3.เลือกหมวดหมู่คำถามที่ต้องการ แล้วเลือกระดับความยากและจำนวนข้อที่ต้องการในหมวดหมู่นั้นๆ แล้วคลิกสร้างข้อสอบ	4.ระบบทำการสร้างข้อสอบตามที่ผู้ใช้งานกำหนด
ผู้ใช้	ระบบ						
1.ไปที่หน้าสร้างข้อสอบ	2.ระบบแสดงหมวดหมู่ของตนเองสำหรับเลือกใช้ในข้อสอบ และแสดงรายละเอียดให้กรอก						
3.เลือกหมวดหมู่คำถามที่ต้องการ แล้วเลือกระดับความยากและจำนวนข้อที่ต้องการในหมวดหมู่นั้นๆ แล้วคลิกสร้างข้อสอบ	4.ระบบทำการสร้างข้อสอบตามที่ผู้ใช้งานกำหนด						
Exception Conditions:	<ul style="list-style-type: none"> <li>- ไม่มีหมวดหมู่คำถามของตนเอง</li> <li>- ผู้ใช้กรอกรายละเอียดไม่ครบถ้วน หรือไม่ตรงเงื่อนไข</li> </ul>						

ตารางที่ 3.11: รายละเอียดยูสเคส สร้างข้อสอบ

### 3.3.2.12 รายละเอียดยูสเคส ดูข้อสอบทั้งหมดของตนเอง

Use Case No:	12	
Use Case Name:	ดูข้อสอบทั้งหมดของตนเอง	
Use Case Scenario:	ผู้ใช้ต้องการดูข้อสอบทั้งหมดของตนเอง	
Triggering Event:	เมื่อผู้ใช้ต้องการดูว่าตนเองมีข้อสอบอะไรบ้าง	
Brief Description:	สำหรับผู้ใช้ที่ดูข้อสอบที่ตนเองสร้างขึ้น หรือดูข้อสอบที่ทำการคัดลอกมาจากข้อสอบสาธารณะ	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, สร้างข้อสอบ, คัดลอกข้อสอบจากสาธารณะ	
Stakeholders:	-	
Pre - Conditions:	ผู้ใช้ต้องทำการเข้าสู่ระบบ	
Post - Conditions:	ระบบทำการแสดงข้อสอบทั้งหมดของตนเอง	
Flow of Events:	ผู้ใช้	ระบบ
	1.ไปที่หน้าดูข้อสอบทั้งหมดของตนเอง	2.ระบบแสดงข้อสอบทั้งหมดของผู้ใช้
Exception Conditions:	ไม่มีข้อสอบของตนเอง	

ตารางที่ 3.12: รายละเอียดยูสเคส ดูข้อสอบทั้งหมดของตนเอง

### 3.3.2.13 รายละเอียดยูสเคส แก้ไขข้อสอบของตนเอง

Use Case No:	13	
Use Case Name:	แก้ไขข้อสอบของตนเอง	
Use Case Scenario:	ผู้ใช้งานต้องการแก้ไขข้อสอบของตนเอง	
Triggering Event:	เมื่อผู้ใช้งานต้องการแก้ไขข้อสอบที่ตนเองสร้างขึ้น หรือข้อสอบที่ทำการคัดลอกมาจากข้อสอบสาธารณะ	
Brief Description:	สำหรับผู้ใช้งานที่ต้องการแก้ไขข้อสอบ เช่น เพิ่มหรือลดจำนวนข้อและระดับความยากในหมวดหมู่คำถามนั้นๆ	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, สร้างข้อสอบ, คัดลอกข้อสอบจากสาธารณะ	
Stakeholders:	-	
Pre - Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้ต้องทำการเข้าสู่ระบบ</li> <li>- ผู้ใช้ต้องสร้างข้อสอบ หรือคัดลอกข้อสอบจากข้อสอบสาธารณะ</li> </ul>	
Post - Conditions:	ระบบทำการแก้ไขข้อสอบตามที่ผู้ใช้งานกำหนด	
Flow of Events:	ผู้ใช้งาน	ระบบ
	1.ไปที่หน้าดูข้อสอบทั้งหมดของตนเอง  3.เลือกข้อสอบที่ต้องการแก้ไข  5.ทำการแก้ไขข้อสอบตามที่ต้องการ แล้วคลิกยืนยัน	2.ระบบแสดงข้อสอบทั้งหมดของผู้ใช้  4.ระบบทำการแสดงรายละเอียดของข้อสอบนั้น  6.ระบบทำการแก้ไขตามที่ผู้ใช้งานกำหนด
Exception Conditions:	<ul style="list-style-type: none"> <li>- ไม่มีข้อสอบของตนเอง</li> <li>- ผู้ใช้มีการใช้ข้อซ้ำกับข้อสอบของตนเองที่มีอยู่</li> <li>- ผู้ใช้กรอกรายละเอียดไม่ครบถ้วน หรือไม่ตรงเงื่อนไข</li> </ul>	

ตารางที่ 3.13: รายละเอียดยูสเคส แก้ไขข้อสอบของตนเอง

### 3.3.2.14 รายละเอียดยูสเคส แก้ไขข้อสอบของตนเอง

Use Case No:	13	
Use Case Name:	แก้ไขข้อสอบของตนเอง	
Use Case Scenario:	ผู้ใช้งานต้องการแก้ไขรายละเอียดข้อสอบของตนเอง	
Triggering Event:	เมื่อผู้ใช้งานต้องการแก้ไขรายละเอียดข้อสอบที่ตนเองสร้างขึ้น หรือ ข้อสอบที่ทำการคัดลอกมาจากข้อสอบสาธารณะ	
Brief Description:	สำหรับผู้ใช้งานที่ต้องการแก้ไขรายละเอียดข้อสอบ เช่น เพิ่มหรือลด จำนวนข้อและระดับความยากในหมวดหมู่คำถามนั้นๆ	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, สร้างข้อสอบ, คัดลอกข้อสอบจากสาธารณะ	
Stakeholders:	-	
Pre - Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้ต้องทำการเข้าสู่ระบบ</li> <li>- ผู้ใช้ต้องสร้างข้อสอบ หรือคัดลอกข้อสอบจากข้อสอบสาธารณะ</li> </ul>	
Post - Conditions:	ระบบทำการแก้ไขข้อสอบตามที่ผู้ใช้งานกำหนด	
Flow of Events:	ผู้ใช้	ระบบ
	1.ไปที่หน้าดูข้อสอบทั้งหมด ของตนเอง  3.เลือกข้อสอบที่ต้องการ แก้ไข  5.ทำการแก้ไขข้อสอบตามที่ ต้องการ แล้วคลิกยืนยัน	2.ระบบแสดงข้อสอบ ทั้งหมดของผู้ใช้  4.ระบบทำการแสดงราย ละเอียดของข้อสอบนั้น  6.ระบบทำการแก้ไขตามที่ผู้ ใช้กำหนด
Exception Conditions:	<ul style="list-style-type: none"> <li>- ไม่มีข้อสอบของตนเอง</li> <li>- ผู้ใช้มีการใช้ข้อจำกัดข้อสอบของตนเองที่มีอยู่</li> <li>- ผู้ใช้กรอกรายละเอียดไม่ครบถ้วน หรือไม่ตรงเงื่อนไข</li> </ul>	

ตารางที่ 3.14: รายละเอียดยูสเคส แก้ไขข้อสอบของตนเอง

### 3.3.2.15 รายละเอียดยูสเคส ลบข้อสอบ

Use Case No:	14	
Use Case Name:	ลบข้อสอบ	
Use Case Scenario:	ผู้ใช้งานต้องการลบข้อสอบ	
Triggering Event:	เมื่อผู้ใช้งานไม่ต้องการข้อสอบนั้นแล้ว	
Brief Description:	สำหรับผู้ใช้งานที่ต้องการลบข้อสอบที่ตนเองสร้างขึ้น หรือลบข้อสอบที่ทำการคัดลอกมาจากข้อสอบสาธารณะ	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, สร้างข้อสอบ, คัดลอกข้อสอบจากสาธารณะ	
Stakeholders:	-	
Pre - Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้ต้องทำการเข้าสู่ระบบ</li> <li>- ผู้ใช้ต้องสร้างข้อสอบ หรือคัดลอกข้อสอบจากข้อสอบสาธารณะ</li> </ul>	
Post - Conditions:	ระบบทำการคัดลอกข้อสอบที่ผู้ใช้งานต้องการ	
Flow of Events:	ผู้ใช้งาน	ระบบ
	1.ไปที่หน้าดูข้อสอบทั้งหมดของตนเอง  3.เลือกข้อสอบที่ต้องการลบ  5.ทำการยืนยันการลบข้อสอบ	2.ระบบแสดงข้อสอบทั้งหมดของผู้ใช้  4.ระบบทำการแสดงหน้าต่างข้อความยืนยันการลบ  6.ระบบทำการคัดลอกหมวดหมู่คำถามนั้น
Exception Conditions:	<ul style="list-style-type: none"> <li>- ไม่มีข้อสอบของตนเอง</li> <li>- ผู้ใช้ไม่ทำการยืนยันการลบข้อสอบ</li> </ul>	

ตารางที่ 3.15: รายละเอียดยูสเคส ลบข้อสอบ

### 3.3.2.16 รายละเอียดยูสเคส คู่มือสอบสาธารณะ

Use Case No:	15	
Use Case Name:	คู่มือสอบสาธารณะ	
Use Case Scenario:	ผู้ใช้งานต้องการคู่มือสอบสาธารณะ	
Triggering Event:	เมื่อผู้ใช้งานมีความต้องการที่จะคู่มือสอบสาธารณะ	
Brief Description:	สำหรับผู้ใช้งานที่ต้องการคู่มือสอบอะไรบางอย่างที่เปิดเป็นสาธารณะ	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ	
Stakeholders:	-	
Pre - Conditions:	ผู้ใช้งานต้องทำการเข้าสู่ระบบ	
Post - Conditions:	ระบบทำการแสดงข้อสอบที่เป็นสาธารณะทั้งหมด	
Flow of Events:	ผู้ใช้งาน	ระบบ
	1.ไปที่หน้าข้อสอบ สาธารณะ	2.ระบบแสดงข้อสอบ ทั้งหมดที่เป็นสาธารณะ
Exception Conditions:	ไม่มีข้อสอบสาธารณะอยู่ในระบบ	

ตารางที่ 3.16: รายละเอียดยูสเคส คู่มือสอบสาธารณะ

### 3.3.2.17 รายละเอียดยูสเคส ดูรายละเอียดของข้อสอบสาธารณะ

Use Case No:	16	
Use Case Name:	ดูรายละเอียดของข้อสอบสาธารณะ	
Use Case Scenario:	ผู้ใช้งานต้องการดูรายละเอียดของข้อสอบสาธารณะ	
Triggering Event:	<ul style="list-style-type: none"> <li>- เมื่อผู้ใช้งานมีความต้องการที่จะดูรายละเอียดของข้อสอบสาธารณะ เพื่อที่จะนำไปเป็นแนวทางในข้อสอบของตนเอง</li> <li>- ผู้ใช้งานต้องการทำการคัดลอกข้อสอบสาธารณะ ไปปรับแต่งหรือนำไปใช้กับผู้อื่น</li> </ul>	
Brief Description:	สำหรับให้ผู้ใช้งานดูว่ามีข้อสอบสาธารณะใดบ้างที่อยู่ในระบบ และสามารถคัดลอกไปใช้ได้	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ	
Stakeholders:	-	
Pre - Conditions:	ผู้ใช้งานต้องการเข้าสู่ระบบ	
Post - Conditions:	ระบบทำการแสดงรายละเอียดของข้อสอบ	
Flow of Events:	ผู้ใช้งาน	ระบบ
	1.ไปที่หน้าข้อสอบสาธารณะ  3.คลิกดูข้อสอบที่ต้องการดูรายละเอียด	2.ระบบแสดงข้อสอบทั้งหมดที่เป็นสาธารณะ  4.ระบบแสดงรายละเอียดข้อสอบ
Exception Conditions:	ไม่มีข้อสอบสาธารณะอยู่ในระบบ	

ตารางที่ 3.17: รายละเอียดยูสเคส ดูข้อสอบสาธารณะ



### 3.3.2.18 รายละเอียดยูสเคส คัดลอกข้อสอบสาธารณะ

Use Case No:	17	
Use Case Name:	คัดลอกข้อสอบสาธารณะ	
Use Case Scenario:	ผู้ใช้ต้องการคัดลอกข้อสอบสาธารณะ	
Triggering Event:	เมื่อผู้ใช้มีความต้องการที่จะนำข้อสอบสาธารณะไปใช้ หรือนำไปปรับแก้ เพื่อนำไปใช้กับผู้สมัครงาน	
Brief Description:	สำหรับให้ผู้ใช้คัดลอกข้อสอบไปใช้ หรือนำไปปรับแก้ไขตามที่ผู้ใช้ต้องการ	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, คู่มือข้อสอบสาธารณะ	
Stakeholders:	-	
Pre - Conditions:	ผู้ใช้ต้องทำการเข้าสู่ระบบ	
Post - Conditions:	ระบบทำการคัดลอกข้อสอบสาธารณะที่ผู้ใช้ต้องการ	
Flow of Events:	ผู้ใช้	ระบบ
	1.ไปที่หน้าข้อสอบสาธารณะ  3.คลิกคู่มือข้อสอบที่ต้องการดูรายละเอียด  5.ผู้ใช้ทำการยืนยันการคัดลอกข้อสอบ	2.ระบบแสดงข้อสอบทั้งหมดที่เป็นสาธารณะ  4.ระบบแสดงหน้าต่างยืนยันการคัดลอก  6.ระบบทำการคัดลอกข้อสอบที่ผู้ใช้ต้องการ และหมวดหมู่คำถามทั้งหมดที่ใช้ในข้อสอบนั้น ไปเก็บไว้ในข้อสอบของตนเองและหมวดหมู่คำถามของตนเอง
Exception Conditions:	ไม่มีข้อสอบสาธารณะอยู่ในระบบ	

ตารางที่ 3.18: รายละเอียดยูสเคส คัดลอกข้อสอบสาธารณะ

### 3.3.2.19 รายละเอียดยูสเคส ส่งข้อสอบให้ผู้สมัครงาน

Use Case No:	18	
Use Case Name:	ส่งข้อสอบให้ผู้สมัครงาน	
Use Case Scenario:	ผู้ใช้งานต้องการส่งข้อสอบให้ผู้สมัครทำ	
Triggering Event:	เมื่อผู้สร้างข้อสอบเรียบร้อยแล้ว หรือไปทำการคัดลอกข้อสอบมาจากสาธารณะ แล้วต้องการส่งข้อสอบให้ผู้สมัครงานทำ	
Brief Description:	สำหรับให้ผู้สร้างส่งข้อสอบให้ผู้สมัครงานผ่านทางอีเมล	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, สร้างข้อสอบ, คัดลอกข้อสอบสาธารณะ	
Stakeholders:	-	
Pre - Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้ต้องทำการเข้าสู่ระบบ</li> <li>- ผู้ใช้ต้องทำการสร้างข้อสอบ หรือคัดลอกข้อสอบจากข้อสอบสาธารณะ</li> </ul>	
Post - Conditions:	ระบบทำการส่งข้อสอบให้ผู้สมัครงานผ่านอีเมล	
Flow of Events:	<p>ผู้ใช้</p> <ol style="list-style-type: none"> <li>1.ไปที่หน้าข้อสอบของตนเอง</li> <li>3.คลิกปุ่มส่งข้อสอบบริเวณข้อสอบที่ต้องการส่ง</li> <li>5.กรอกรายละเอียดแล้วคลิกส่งข้อสอบให้ผู้สมัคร</li> </ol>	<p>ระบบ</p> <ol style="list-style-type: none"> <li>2.ระบบแสดงข้อสอบทั้งหมดของผู้ใช้</li> <li>4.ระบบแสดงกรอบหน้าต่างให้ใส่รายละเอียดผู้สมัครงาน เช่น อีเมลผู้สมัคร, ระยะเวลาในการทำข้อสอบ และวันหมดอายุของข้อสอบ</li> <li>6.ระบบทำการสุ่มคำถามจากหมวดหมู่คำถามที่กำหนดจำนวนข้อและความยากโดยรายละเอียดของข้อสอบ แล้วส่งให้ผู้สมัครงานผ่านอีเมล</li> </ol>
Exception Conditions:	กรอกข้อมูลไม่ครบ หรือไม่ตรงตามเงื่อนไข	

ตารางที่ 3.19: รายละเอียดยูสเคส ส่งข้อสอบให้ผู้สมัครงาน

### 3.3.2.20 รายละเอียดยูสเคส ดูประวัติการส่งข้อสอบทั้งหมด

Use Case No:	19	
Use Case Name:	ดูประวัติการส่งข้อสอบทั้งหมด	
Use Case Scenario:	ผู้ใช้งานต้องการดูประวัติการส่งข้อสอบ	
Triggering Event:	เมื่อผู้ใช้งานส่งข้อสอบ แล้วต้องการดูประวัติการส่งข้อสอบที่ผู้ใช้งานได้ส่งไปให้พนักงานทำ	
Brief Description:	สำหรับให้ผู้ใช้งานที่ต้องการทราบประวัติการส่งข้อสอบของตนเอง	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, ส่งข้อสอบให้ผู้สมัคร	
Stakeholders:	-	
Pre - Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้ต้องทำการเข้าสู่ระบบ</li> <li>- ผู้ใช้ต้องทำการส่งข้อสอบให้ผู้สมัคร สาธารณะ</li> </ul>	
Post - Conditions:	ระบบทำการแสดงประวัติการส่งข้อสอบที่ผู้ใช้งานทำการส่งให้ผู้สมัคร	
Flow of Events:	ผู้ใช้งาน	ระบบ
	1.ไปที่หน้าของประวัติการส่งข้อสอบ	2.ระบบแสดงประวัติข้อสอบทั้งหมดที่ผู้ใช้งานทำการส่งให้ผู้สมัครงาน
Exception Conditions:	ไม่มีการส่งข้อสอบให้ผู้สมัครงาน	

ตารางที่ 3.20: รายละเอียดยูสเคส ดูประวัติการส่งข้อสอบทั้งหมด

### 3.3.2.21 รายละเอียดยูสเคส ดูรายละเอียดประวัติการส่งข้อสอบ

Use Case No:	20	
Use Case Name:	ดูรายละเอียดประวัติการส่งข้อสอบ	
Use Case Scenario:	เมื่อผู้ใช้ส่งข้อสอบให้ผู้สมัครงานทำ แล้วต้องการดูรายละเอียดว่า ส่งให้ผู้สมัครงานคนไหน และส่งไปเมื่อไหร่	
Triggering Event:	เมื่อผู้ใช้งานส่งข้อสอบให้ผู้สมัครทำ แล้วต้องการดูรายละเอียด ของการส่งข้อสอบให้ผู้สมัครงาน	
Brief Description:	สำหรับให้ผู้ใช้ที่ต้องการทราบประวัติการส่งข้อสอบให้ผู้สมัคร เช่น ส่งให้ผู้สมัครคนไหนบ้าง, ใช้เวลาในการทำเท่าไร และกำ หนดวัน หมดอายุไว้วันไหน	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, ส่งข้อสอบให้ผู้สมัคร	
Stakeholders:	-	
Pre - Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้ต้องทำการเข้าสู่ระบบ</li> <li>- ผู้ใช้ต้องทำการส่งข้อสอบให้ผู้สมัคร</li> </ul>	
Post - Conditions:	ระบบทำการแสดงรายละเอียดของประวัติข้อสอบที่มีการส่งให้ ผู้สมัคร	
Flow of Events:	ผู้ใช้	ระบบ
	1.ไปที่หน้าของประวัติการ ส่งข้อสอบ  3.คลิกประวัติการส่งข้อสอบ ที่ต้องการดูรายละเอียด	2.ระบบแสดงประวัติข้อ สอบทั้งหมดที่ผู้ใช้ทำการ ส่งให้ผู้สมัครงาน  4.ระบบทำการแสดงรายละเอียดประวัติข้อสอบที่ต้อง การ
Exception Conditions:	ไม่มีการส่งข้อสอบให้ผู้สมัครงาน	

ตารางที่ 3.21: รายละเอียดยูสเคส ดูรายละเอียดประวัติการส่งข้อสอบ

### 3.3.2.22 รายละเอียดยูสเคส รายละเอียดข้อสอบ

Use Case No:	21	
Use Case Name:	รายละเอียดข้อสอบ	
Use Case Scenario:	ผู้สมัครงานอ่านรายละเอียดก่อนที่จะทำข้อสอบ	
Triggering Event:	เมื่อผู้สมัครงานได้รับลิงค์สำหรับทำข้อสอบผ่านทางอีเมล	
Brief Description:	เพื่อให้ผู้ใช้ดูรายละเอียดก่อนที่จะเริ่มทำข้อสอบ	
Actors:	ผู้สมัครงาน	
Related Use Cases:	ส่งข้อสอบให้ผู้สมัคร	
Stakeholders:	-	
Pre - Conditions:	ผู้สมัครต้องทำการเข้าอีเมลเพื่อเปิดลิงค์ที่ทางบริษัทได้ทำการส่งมา	
Post - Conditions:	ระบบทำการแสดงรายละเอียดข้อสอบ	
Flow of Events:	ผู้ใช้	ระบบ
	1.คลิกที่ลิงค์ในอีเมลที่ได้รับจากทางบริษัท	2.ระบบทำการแสดงรายละเอียดข้อสอบ
Exception Conditions:		

ตารางที่ 3.22: รายละเอียดยูสเคส รายละเอียดข้อสอบ

### 3.3.2.23 รายละเอียดยูสเคส เริ่มทำข้อสอบ

Use Case No:	22	
Use Case Name:	เริ่มทำข้อสอบ	
Use Case Scenario:	ผู้สมัครกดเริ่มทำข้อสอบ	
Triggering Event:	เมื่อสมัครงานต้องการเริ่มทำข้อสอบ หลังจากที่คุณรายละเอียดของข้อสอบแล้ว	
Brief Description:	สำหรับให้ผู้สมัครงานทำข้อสอบที่ได้รับมาจากทางบริษัท	
Actors:	ผู้สมัครงาน	
Related Use Cases:	รายละเอียดข้อสอบ	
Stakeholders:	-	
Pre - Conditions:	ผู้สมัครต้องทำการเข้าอีเมลเพื่อเปิดลิงค์ที่ทางบริษัทได้ทำการส่งมา	
Post - Conditions:	ระบบแสดงข้อสอบให้ผู้สมัครและเริ่มจับเวลาในการทำข้อสอบ	
Flow of Events:	ผู้ใช้	ระบบ
	1.คลิกเริ่มทำข้อสอบ	2.ระบบ แสดง ข้อสอบ ให้ ผู้สมัครและเริ่มจับเวลาในการทำข้อสอบ
Exception Conditions:	<ul style="list-style-type: none"> <li>- ผู้สมัครส่งข้อสอบไปแล้ว จึงไม่สามารถทำใหม่อีกครั้งได้</li> <li>- ข้อสอบหมดอายุการใช้งานแล้ว</li> </ul>	

ตารางที่ 3.23: รายละเอียดยูสเคส รายละเอียดข้อสอบ

### 3.3.2.24 รายละเอียดยูสเคส ส่งข้อสอบให้ทางบริษัท

Use Case No:	23	
Use Case Name:	ส่งข้อสอบให้ทางบริษัท	
Use Case Scenario:	ผู้สมัครงานทำการส่งข้อสอบกลับไปให้ทางบริษัท	
Triggering Event:	เมื่อผู้สมัครงานทำข้อสอบเสร็จแล้ว และต้องการส่งข้อสอบให้ทางบริษัท	
Brief Description:	สำหรับให้ผู้สมัครงานทำการส่งข้อสอบ	
Actors:	ผู้สมัครงาน	
Related Use Cases:	เริ่มทำข้อสอบ	
Stakeholders:	-	
Pre - Conditions:	ผู้สมัครต้องทำการกดเริ่มทำข้อสอบ ที่ทางบริษัทส่งมา	
Post - Conditions:	ระบบทำการส่งข้อสอบให้ทางบริษัท	
Flow of Events:	ผู้ใช้	ระบบ
	1.คลิกส่งข้อสอบ	2.ระบบทำการส่งข้อสอบให้ทางบริษัท
Exception Conditions:	<ul style="list-style-type: none"> <li>- ผู้สมัครส่งข้อสอบไปแล้ว จึงไม่สามารถทำใหม่อีกครั้งได้</li> <li>- ข้อสอบหมดอายุการใช้งานแล้ว</li> </ul>	

ตารางที่ 3.24: รายละเอียดยูสเคส ส่งข้อสอบให้ทางบริษัท

### 3.3.2.25 รายละเอียดยูสเคส หมดเวลาทำข้อสอบ

Use Case No:	24	
Use Case Name:	หมดเวลาทำข้อสอบ	
Use Case Scenario:	ผู้สมัครงานทำข้อสอบเกินเวลาที่กำหนด	
Triggering Event:	เมื่อผู้สมัครงานทำข้อสอบอยู่แล้วหมดเวลาทำข้อสอบ โดยที่ยังไม่ได้ทำการส่งข้อสอบ	
Brief Description:	สำหรับกรณี que ผู้สมัครทำข้อสอบไม่ทันในเวลาที่กำหนด	
Actors:	ผู้สมัครงาน	
Related Use Cases:	เริ่มทำข้อสอบ	
Stakeholders:	-	
Pre - Conditions:	ผู้ใช้ต้องคลิกเริ่มทำข้อสอบ และทำงานหมดเวลาโดยที่ยังไม่ได้ส่ง	
Post - Conditions:	ระบบทำการส่งข้อสอบให้ทางบริษัท	
Flow of Events:	ผู้ใช้	ระบบ
		1.ระบบตรวจพบว่าหมดเวลาในการทำข้อสอบ แล้วทำการส่งข้อสอบให้ทางบริษัท
Exception Conditions:	-	

ตารางที่ 3.25: รายละเอียดยูสเคส หมดเวลาทำข้อสอบ



### 3.3.2.26 รายละเอียดยูสเคส ดูข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว

Use Case No:	25	
Use Case Name:	ดูข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว	
Use Case Scenario:	ผู้ใช้ดูข้อว่ามีผู้สมัครงานคนไหนทำข้อสอบเสร็จแล้วบ้าง	
Triggering Event:	เมื่อผู้ใช้ต้องการดูว่าผู้สมัครงานคนไหนทำข้อสอบเสร็จแล้วบ้าง	
Brief Description:	สำหรับกรณีที่ผู้ใช้ต้องการดูว่าผู้สมัครงานคนไหนทำข้อสอบเสร็จแล้วบ้าง	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, ส่งข้อสอบให้ทางบริษัท	
Stakeholders:	-	
Pre - Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้ต้องทำการเข้าสู่ระบบ</li> <li>- ผู้ใช้ต้องทำการส่งข้อสอบให้ผู้สมัครงาน</li> <li>- ผู้สมัครงานต้องส่งข้อสอบให้บริษัทเรียบร้อยแล้ว</li> </ul>	
Post - Conditions:	ระบบทำการแสดงข้อสอบทั้งหมดที่ผู้สมัครงานทำเสร็จแล้ว	
Flow of Events:	ผู้ใช้	ระบบ
	1.ไปที่หน้าข้อสอบที่ทำเสร็จแล้ว	2.ระบบแสดงข้อสอบทั้งหมดที่ผู้สมัครทำเสร็จแล้ว
Exception Conditions:	ไม่มีข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว	

ตารางที่ 3.26: รายละเอียดยูสเคส ดูข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว

### 3.3.2.27 รายละเอียดยูสเคส ตรวจสอบข้อสอบที่ผู้สมัครทำเสร็จแล้ว

Use Case No:	26	
Use Case Name:	ตรวจสอบข้อสอบที่ผู้สมัครทำเสร็จแล้ว	
Use Case Scenario:	ผู้ใช้ตรวจสอบข้อสอบที่ผู้สมัครเสร็จแล้ว	
Triggering Event:	เมื่อผู้ใช้ต้องการตรวจสอบข้อสอบที่มีคำถามประเภทอัตนัย และคำถามที่มีไฟล์แนบ ซึ่งไม่มีคำตอบที่ตายตัว	
Brief Description:	สำหรับให้ผู้ใช้ตรวจสอบข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว	
Actors:	พนักงาน	
Related Use Cases:	เข้าสู่ระบบ, ส่งข้อสอบให้ทางบริษัท	
Stakeholders:	-	
Pre - Conditions:	<ul style="list-style-type: none"> <li>- ผู้ใช้ต้องทำการเข้าสู่ระบบ</li> <li>- ผู้ใช้ต้องทำการส่งข้อสอบให้ผู้สมัครงาน</li> <li>- ผู้สมัครงานต้องส่งข้อสอบให้บริษัทเรียบร้อยแล้ว</li> </ul>	
Post - Conditions:	ระบบทำการบันทึกคะแนน	
Flow of Events:	<div>ผู้ใช้</div> <div> <p>1.ไปที่หน้าข้อสอบทั้งหมดที่ผู้สมัครงานทำเสร็จแล้ว</p> <p>3.คลิกข้อสอบที่ต้องการทำการตรวจ</p> <p>5.ผู้ใช้ทำการตรวจข้อสอบแล้วคลิกยืนยัน</p> </div>	<div>ระบบ</div> <div> <p>2.ระบบแสดงข้อสอบทั้งหมดที่ผู้สมัครทำเสร็จแล้ว</p> <p>4.ระบบแสดงหน้าตรวจข้อสอบ</p> <p>6.ระบบทำการบันทึกการตรวจข้อสอบแล้วสรุปผลออกมาเป็นคะแนน</p> </div>
Exception Conditions:	ไม่มีข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว	

ตารางที่ 3.27: รายละเอียดยูสเคส ดูข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว

### 3.3.3 การออกแบบฐานข้อมูล

ทางผู้จัดทำได้ใช้ฐานข้อมูลแบบไม่มีโครงสร้าง ในเว็บแอปพลิเคชันนี้ ซึ่งเป็นฐานข้อมูลที่สามารถเก็บข้อมูลในปริมาณที่มากได้ สามารถอ่านเขียนข้อมูลได้อย่างรวดเร็ว และสามารถเก็บข้อมูลที่ซับซ้อนได้ ซึ่งเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน ได้แบ่งประเภทของการเก็บข้อมูลออกเป็น 5 ตาราง ดังนี้

ชื่อตาราง	รายละเอียด
CATEGORIES	สำหรับจัดเก็บคำถามตามหมวดหมู่
EXAMS	สำหรับจัดเก็บรายละเอียดการของสุ่มคำถามในข้อสอบตามที่ผู้ออกข้อสอบกำหนด
EXAM_RANDOMS	จัดเก็บข้อสอบที่ถูกสุ่มคำถามมาจาก หมวดหมู่ต่างๆ ตามที่กำหนดอยู่ในรายละเอียดข้อสอบ
SEND_EXAMS	สำหรับจัดเก็บประวัติการส่งข้อสอบ
USERS	สำหรับจัดเก็บข้อมูลผู้ใช้งาน

ตารางที่ 3.28: ฐานข้อมูลทั้งหมดที่ใช้

## 3.4 รายละเอียดการพัฒนาระบบ

เว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน ได้แบ่งการพัฒนาก่อเป็น 2 ส่วนหลักๆ คือส่วนต่อประสานกับผู้ใช้ และส่วนที่จัดการกับฐานข้อมูล

### 3.4.1 ส่วนต่อประสานกับผู้ใช้ (back-end)

รายละเอียดของแต่ละเซอร์วิสของเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน ที่ทางผู้จัดทำได้พัฒนาขึ้น มีด้วยกันทั้งหมด 6 เซอร์วิสดังต่อไปนี้

1. **user service** เป็นเซอร์วิสที่จัดการกับข้อมูลผู้ใช้งาน เพื่อเข้าใช้งานระบบ และเป็นข้อมูลอ้างอิงในการสร้างส่วนต่างๆในระบบ โดยมีฟังก์ชันต่างๆดังนี้
  - **create user** ฟังก์ชันสร้างผู้ใช้งาน เพื่อเข้าใช้งานระบบ

<b>URL:</b>	https://{url}/users
<b>Method:</b>	POST
<b>Auth require:</b>	False
<b>Format:</b>	JSON
<b>Parameters:</b>	-
<b>Body:</b>	user data
<b>Response:</b>	user data

ตารางที่ 3.29: รายละเอียดฟังก์ชันสร้างผู้ใช้งาน

- **activate user** ฟังก์ชันยืนยันตัวตนเพื่อเข้าใช้งานระบบ หลังจากที่ทำการสมัครสมาชิก

<b>URL:</b>	https://{url}/users/activate/{:activateKey}
<b>Method:</b>	PATCH
<b>Auth require:</b>	False
<b>Format:</b>	JSON
<b>Parameters:</b>	activateKey(String)
<b>Body:</b>	-
<b>Response:</b>	user data with token

- **login user** ฟังก์ชันการเข้าสู่ระบบ

<b>URL:</b>	https://{url}/users/login
<b>Method:</b>	POST
<b>Auth require:</b>	False
<b>Format:</b>	JSON
<b>Parameters:</b>	-
<b>Body:</b>	email(String), password(String)
<b>Response:</b>	user data with token

- **update user** ฟังก์ชันแก้ไขข้อมูลผู้ใช้งาน

<b>URL:</b>	https://{url}/users
<b>Method:</b>	PATCH
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	-
<b>Body:</b>	user data
<b>Response:</b>	user data

- **logout user** ฟังก์ชันการออกจากระบบ

<b>URL:</b>	https://{url}/users/logout
<b>Method:</b>	POST
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	-
<b>Body:</b>	-
<b>Response:</b>	user data

- **logout all user** ฟังก์ชันการออกจากระบบ ทุกเครื่องที่มีการเข้าสู่ระบบด้วยผู้ใช้เดียวกัน

<b>URL:</b>	https://{url}/users/logout-all
<b>Method:</b>	POST
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	-
<b>Body:</b>	-
<b>Response:</b>	user data

- **delete user** ฟังก์ชันการลบผู้ใช้งานออกจากระบบ

<b>URL:</b>	https://{url}/users/me
<b>Method:</b>	DELETE
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	-
<b>Body:</b>	-
<b>Response:</b>	user data

2. **category service** เป็นเซอร์วิสที่จัดการกับหมวดหมู่คำถาม โดยสร้างคำถามที่แบ่งตามเป็นหมวดหมู่ต่างๆ ตามที่ผู้ใช้งานต้องการ พร้อมกำหนดระดับความยากของคำถามในแต่ละข้อ

- **create category** ฟังก์ชันการสร้างหมวดหมู่คำถาม

<b>URL:</b>	https://{url}/category
<b>Method:</b>	POST
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	-
<b>Body:</b>	category data
<b>Response:</b>	category data

- **upload file** ฟังก์ชันสำหรับเพิ่มรูปภาพกับไฟล์ต่างๆ ที่อยู่ในคำถาม หรือคำตอบ

<b>URL:</b>	https://{url}/upload
<b>Method:</b>	POST
<b>Auth require:</b>	True
<b>Format:</b>	Form-data
<b>Parameters:</b>	-
<b>Body:</b>	(file)
<b>Response:</b>	URL to file path(String), filename(String)

- **delete file** ฟังก์ชันการลบรูปภาพกับไฟล์ต่างๆ ที่อยู่ในคำถาม หรือคำตอบ

<b>URL:</b>	https://{url}/file
<b>Method:</b>	DELETE
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	-
<b>Body:</b>	filename(STRING)
<b>Response:</b>	delete status(STRING)

- **update category** ฟังก์ชันการแก้ไขหมวดหมู่คำถาม

<b>URL:</b>	https://{url}/category/{:id}
<b>Method:</b>	PATCH
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	id(ID)
<b>Body:</b>	category data
<b>Response:</b>	category data

- **get category details** ฟังก์ชันการดูคำถามที่อยู่ภายในหมวดหมู่คำถาม

<b>URL:</b>	https://{url}/category/{:id}
<b>Method:</b>	POST
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	id(ID)
<b>Body:</b>	-
<b>Response:</b>	category data

- **get my categories** ฟังก์ชันการดูหมวดหมู่คำถามทั้งหมดของตนเอง

<b>URL:</b>	https://{url}/category/by/my-cat?page={:page}&search={:search}
<b>Method:</b>	GET
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	page(NUMBER), search(STRING)
<b>Body:</b>	-
<b>Response:</b>	categories data

- **get public categories** ฟังก์ชันการดูหมวดหมู่คำถามทั้งหมดที่เป็นสาธารณะ และไม่ใช้หมวดหมู่คำถามของตนเอง

<b>URL:</b>	https://{url}/category/by/public-cat?page={:page}&search={:search}
<b>Method:</b>	GET
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	page(NUMBER), search(STRING)
<b>Body:</b>	-
<b>Response:</b>	categories data

- **get all categories for public exam** ฟังก์ชันการดูหมวดหมู่คำถามทั้งหมด ที่ใช้สำหรับสร้างข้อสอบที่เป็นสาธารณะ

<b>URL:</b>	https://{url}/category/by/public-exam
<b>Method:</b>	GET
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	-
<b>Body:</b>	-
<b>Response:</b>	categories data



- **get all categories for private exam** ฟังก์ชันการดูหมวดหมู่คำถามทั้งหมด ที่ใช้สำหรับสร้างข้อสอบที่ไม่เปิดเป็นสาธารณะ

<b>URL:</b>	https://{url}/category/by/private-exam
<b>Method:</b>	GET
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	-
<b>Body:</b>	-
<b>Response:</b>	categories data

- **delete category** ฟังก์ชันการลบหมวดหมู่คำถาม

<b>URL:</b>	https://{url}/category/{:id}
<b>Method:</b>	DELETE
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	id(ID)
<b>Body:</b>	-
<b>Response:</b>	categories data

- **copy category** ฟังก์ชันการคัดลอกหมวดหมู่คำถาม ที่ต้องการคัดลอก เช่น คัดลอกหมวดหมู่คำถามสาธารณะ หรือคัดลอกหมวดหมู่คำถามของตนเอง

<b>URL:</b>	https://{url}/category/fork/{:id}
<b>Method:</b>	POST
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	id(ID)
<b>Body:</b>	-
<b>Response:</b>	categories data

3. **exam service** เป็นเซอร์วิสที่จัดการกับรายละเอียดต่างๆ ในข้อสอบ ข้อสอบนั้นสามารถกำหนดได้ว่า ต้องการนำคำถามจากหมวดหมู่คำถามไหนมาใช้ในข้อสอบ ใช้จำนวนกี่ข้อ และต้องการระดับความยากเท่าใด โดยมีฟังก์ชันต่างๆ ดังนี้

- **create exam** ฟังก์ชันการสร้างข้อสอบ โดยกำหนดรายละเอียดต่างๆ ของข้อสอบ เช่น ต้องการหมวดหมู่คำถามใดบ้าง, ต้องการกี่ข้อและระดับความยากเท่าใด, ให้อเวลาในการทำข้อสอบเท่าใด และวันสุดท้ายที่สามารถทำข้อสอบได้ เมื่อกำหนดรายละเอียดต่างๆ ของข้อสอบเรียบร้อยแล้ว ระบบจะสุ่มคำถามเมื่อส่งข้อสอบให้ผู้สมัครงานทำ

<b>URL:</b>	https://{url}/exam
<b>Method:</b>	POST
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	-
<b>Body:</b>	exam data
<b>Response:</b>	exam data

- **get exam details** ฟังก์ชันการดูรายละเอียดข้อสอบ เพื่อดูรายละเอียดภายในข้อสอบนั้นๆ

<b>URL:</b>	https://{url}/exam/{:id}
<b>Method:</b>	GET
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	id(ID)
<b>Body:</b>	-
<b>Response:</b>	exam data

- **update exam** ฟังก์ชันการแก้ไขรายละเอียดต่างๆภายในข้อสอบ

<b>URL:</b>	https://{url}/exam/{:id}
<b>Method:</b>	PATCH
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	id(ID)
<b>Body:</b>	exam data
<b>Response:</b>	exam data

- **get my exams** ฟังก์ชันการดูข้อสอบทั้งหมดของตนเอง

<b>URL:</b>	https://{url}/exam/by/my-exam?page={:page}&search={:search}
<b>Method:</b>	GET
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	page(NUMBER), search(STRING)
<b>Body:</b>	-
<b>Response:</b>	exams data

- **get public exams** ฟังก์ชันการดูข้อสอบทั้งหมดที่เป็นสาธารณะ และไม่ใช่วข้อสอบของตนเอง

<b>URL:</b>	https://{url}/exam/by/public?page={:page}&search={:search}
<b>Method:</b>	GET
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	page(NUMBER), search(STRING)
<b>Body:</b>	-
<b>Response:</b>	exams data

- **delete exam** ฟังก์ชันการลบข้อสอบของตนเอง

<b>URL:</b>	https://{url}/exam/{:id}
<b>Method:</b>	DELETE
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	id(ID)
<b>Body:</b>	-
<b>Response:</b>	exams data

- **copy exam** ฟังก์ชันการคัดลอกข้อสอบ จากข้อสอบสาธารณะมาเก็บไว้ในข้อสอบของตนเอง เพื่อนำมาใช้ หรือนำมาแก้ไข โดยการคัดลอกนี้จะทำการคัดลอก หมวดหมู่คำถามที่ใช้ในข้อสอบนั้น มาเก็บไว้ในหมวดหมู่คำถามของตนเองด้วย

<b>URL:</b>	https://{url}/exam/fork/{:id}
<b>Method:</b>	POST
<b>Auth require:</b>	True
<b>Format:</b>	JSON
<b>Parameters:</b>	id(ID)
<b>Body:</b>	-
<b>Response:</b>	exams data

## บรรณานุกรม

- [1] Guru99, “Restful web services tutorial with example,” 2020. [Online]. Available: <https://www.guru99.com/restful-web-services.html>
- [2] M. contributors, “Http response status codes.” [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>
- [3] W. contributors, “Model–view–controller.” [Online]. Available: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
- [4] M. Inc., “What is nosql?” [Online]. Available: <https://www.mongodb.com/nosql-explained>
- [5] D. Inc., “What is a container?” [Online]. Available: <https://www.docker.com/resources/what-container>
- [6] W. contributors, “Model–view–viewmodel.” [Online]. Available: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel>
- [7] Vue.js, “What is vuex?” [Online]. Available: <https://vuex.vuejs.org>
- [8] L. Torvalds, “Git.” [Online]. Available: <https://git-scm.com>
- [9] Auth0, “Introduction to json web tokens.” [Online]. Available: <https://jwt.io/introduction>
- [10] I. MongoDB, “mongodb.” [Online]. Available: <https://www.mongodb.com>

ภาคผนวก ก

เรื่องที่หนึ่ง