

รายงานการปฏิบัติงานสหกิจศึกษา
เรื่อง
เว็บแอพพลิเคชันประเมินความสามารถเบื้องต้นของผู้
สมัครงาน
Pre-Employment Testing

ปฏิบัติงาน ณ บริษัท ไซเจ็น

โดย

นายศตวรรษ พิติศุภกุล
รหัสประจำตัว 60070093

รายงานนี้เป็นส่วนหนึ่งของการศึกษารายวิชา สหกิจศึกษา
สาขาวิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 1 ปีการศึกษา 2563
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

รายงานการปฏิบัติงานสหกิจศึกษา
เว็บแอพพลิเคชันประเมินความสามารถเบื้องต้นของผู้
สมัครงาน

Pre-Employment Testing

ปฏิบัติงาน ณ บริษัท ไซเจ็น

โดย

นายศตวรรษ พิชิตศุภกุล
รหัสประจำตัว 60070093

ปฏิบัติงาน ณ บริษัท ไซเจ็น
65/60 ชั้น 6 อาคารชานาญเพ็ญชาติบิสเนสเซ็นเตอร์ 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**

**COPYRIGHT 2020
FACULTY OF INFORMATION TECHNOLOGY
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

วันที่ xx พฤษภาคม พ.ศ. 2563

เรื่อง ขอส่งรายงานการปฏิบัติงานสหกิจศึกษา
เรียน ดร.สุพัฒนา ใจดีพันธ์
ที่ปรึกษาสหกิจศึกษาในสาขาวิชาเทคโนโลยีสารสนเทศ

ตามที่ข้าพเจ้า นายศตวรรษ ชิติศุภกุล นักศึกษาสาขาวิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ได้ปฏิบัติงานสหกิจศึกษาระหว่างวันที่ 3 สิงหาคม พ.ศ.2563 ถึงวันที่ 30 พฤษภาคม พ.ศ.2563 ในตำแหน่ง Full Stack Developer ณ สถานประกอบการชื่อ บริษัท ไซเจ็น และได้รับมอบหมายจากผู้จัดการที่ปรึกษาให้ศึกษาและจัดทำรายงาน เรื่อง เว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน

บัดนี้ การปฏิบัติงานสหกิจศึกษาได้สิ้นสุดลงแล้ว จึงได้ขอส่งรายงานการปฏิบัติงาน สหกิจศึกษา ดังกล่าวมาพร้อมนี้ จำนวน 1 เล่ม เพื่อรับคำปรึกษาต่อไป

จึงเรียนมาเพื่อโปรดพิจารณา

ขอแสดงความนับถือ

(นายศตวรรษ ชิติศุภกุล)

กิตติกรรมประกาศ

ตามที่ข้าพเจ้า นายศตวรรษ พิติสุภากุล ได้มาปฏิบัติงานทางศึกษา ณ บริษัท ไซเจ็น ตั้งแต่วันที่ 3 สิงหาคม พ.ศ.2563 ถึงวันที่ 30 พฤษภาคม พ.ศ.2563 ทำให้ข้าพเจ้าได้รับความรู้และประสบการณ์ต่าง ๆ ที่มีคุณค่ามากมาย สำหรับรายงานทางศึกษาฉบับนี้สำเร็จลงได้ด้วยดี จากความช่วยเหลือและความร่วมมือสนับสนุนของหลายฝ่าย ดังนี้

- คุณ วิวัฒน์ เสียงอนันต์ ตำแหน่ง Web Developer (พนักงานที่ปรึกษา)

นอกจากนี้ยังมีบุคคลท่านอื่น ๆ อีกที่ไม่ได้กล่าวไว้ ณ ที่นี่ ซึ่งให้ความกรุณาแนะนำในจัดทำรายงานทางศึกษาฉบับนี้ ข้าพเจ้าจึงขอขอบพระคุณทุกท่านที่ได้มีส่วนร่วมในการให้ข้อมูลและให้ความเข้าใจเกี่ยวกับชีวิตของการปฏิบัติงาน รวมถึงเป็นที่ปรึกษาในการจัดทำรายงานฉบับนี้จนเสร็จสมบูรณ์

นายศตวรรษ พิติสุภากุล

ผู้จัดทำรายงาน

วันที่ xx พฤษภาคม พ.ศ. 2563

ชื่อรายงานการปฏิบัติงานสหกิจศึกษา เว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของ
ผู้สมัครงาน
ผู้รายงาน นายศตวรรษ ชิติสุภกุล
คณะ เทคโนโลยีสารสนเทศ
สาขาวิชา เทคโนโลยีสารสนเทศ

.....
(คร.สุพัฒดา โชคพันธ์)
อาจารย์ที่ปรึกษาสหกิจศึกษา

.....
(นายวิวัฒน์ เสี้งอนันต์)
พนักงานที่ปรึกษา

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
อนุมัติให้นับรายงานการปฏิบัติงานสหกิจศึกษาฉบับนี้ เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ชื่อรายงาน	เว็บแอพพลิเคชั่นประเมินความสามารถเบื้องต้นของผู้สมัครงาน
ชื่อนักศึกษา	นายศตวรรษ ชิตศุภกุล
รหัสนักศึกษา	60070093
สาขาวิชา	เทคโนโลยีสารสนเทศ
อาจารย์ที่ปรึกษา	ดร.สุพันดา โชคพันธ์
ปีการศึกษา	2563

บทคัดย่อ

รายงานการปฏิบัติงานสหกิจศึกษาฉบับนี้ กล่าวถึงวัตถุประสงค์ของการปฏิบัติงาน และรายละเอียดของการปฏิบัติงาน ซึ่งประกอบด้วย ที่มา, ความสำคัญ, รายละเอียด, การออกแบบ ของกระบวนการทำงานของเว็บแอพพลิเคชั่นประเมินความสามารถเบื้องต้นของผู้สมัครงาน โดยทางบริษัท ใช้เงิน ได้มอบหมายให้ช่วงเวลาที่ปฏิบัติงานสหกิจศึกษา สร้างนวัตกรรมใหม่ๆ ขึ้นมาออกแบบ ข้อสอบเพื่อทดสอบความสามารถของผู้สมัครงาน เป็นเว็บแอพพลิเคชั่นที่ สามารถให้พนักงานมาออกแบบข้อสอบเพื่อทดสอบความสามารถของผู้ที่มาสมัครงาน โดยการออกแบบข้อสอบนั้นสามารถตอบโจทย์ความสามารถได้หลายรูปแบบ เช่น ความสามารถอัตนัย, ปรนัย หรือ ตามโดยการแนบไฟล์ต่างๆ ให้ทำเพื่อวัดความสามารถของผู้สมัครงาน ซึ่งทางบริษัทนั้นมีระบบดังกล่าวมาก่อนหน้านี้แล้ว แต่การส่งข้อสอบให้ผู้สมัครงานในแต่ละครั้งทางบริษัทต้องออกแบบข้อสอบใหม่ทุกครั้ง เพื่อป้องกันข้อสอบเดิมที่อาจถูกเผยแพร่ซึ่งทำให้การทดสอบนั้นไม่ประสิทธิภาพลง พนักงานในบริษัทจึงต้องสละเวลาในการออกแบบข้อสอบทุกครั้งที่มีการสมัครงานเพื่อให้ข้อสอบมีมาตรฐานสม่ำเสมอ เนื่องด้วยปัญหาดังกล่าวทางบริษัทจึงมีการมอบหมายให้นักศึกษาสร้างเว็บแอพพลิเคชั่นประเมินความสามารถเบื้องต้นของผู้สมัครงานใหม่ โดยที่เว็บแอพพลิเคชั่นนี้สามารถกำหนดความยากของข้อสอบ เพื่อทำการสุ่มคำถามจากคลังข้อสอบที่สร้างไว้ ทำให้การทดสอบแต่ละครั้งผู้ทดสอบจะได้ข้อสอบที่แตกต่างกันออกไป อ้างอิงโดยระดับความยากที่ผู้ออกข้อสอบทำการทำหน้าที่ มีการกำหนดเวลาในการทำข้อสอบโดยให้ทำข้อสอบภายในเวลาที่กำหนด และมีการกำหนดวันสื้นสุดของการทำข้อสอบซึ่งผู้สมัครงานจะไม่สามารถเริ่มข้อสอบทำได้หากเกินเวลาที่กำหนด โดยเว็บแอพพลิเคชั่นนี้จะทำการส่งข้อสอบให้ผู้สมัครงานผ่านทางอีเมลตามที่ผู้ออกข้อสอบกำหนด เมื่อผู้สมัครงานทำข้อสอบเรียบร้อยแล้วทางบริษัทจะได้รับผลคะแนนที่เป็นปัจจัย ส่วนคำถามที่เป็นอัตนัย และคำถามที่มีไฟล์แนบนั้นผู้ที่ส่งข้อสอบให้ทำต้องเป็นผู้ทำการตรวจตัวยตโนเอง และระบบจะทำการสรุปคะแนนทั้งหมดให้โดยอัตโนมัติ เว็บแอพพลิเคชั่นนี้จะทำให้ทางบริษัท ใช้เงิน ลดระยะเวลาในสร้างข้อสอบเพื่อการคัดกรองผู้ที่มาสมัครงาน และได้รับผู้สมัครงานที่มีคุณภาพตรงตามความต้องการของบริษัท

Project Title Pre-Employment Testing
Name Satawat Thitisupakul
Student ID 60070093
Department Information Technology
Advisor AdvisorName AdvisorSurname
Year 2563

Abstract

Abstract

สารบัญ

	หน้า
กิตติกรรมประกาศ	vi
บทคัดย่อ	i
บทคัดย่อ ภาษาอังกฤษ	ii
สารบัญ	iii
สารบัญตาราง	iv
สารบัญภาพ	vii
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	16
บทที่ 3 การออกแบบระบบและรายละเอียดการพัฒนา	18
3.1 ภาพรวมของเว็บแอปพลิเคชัน	18
3.2 วิเคราะห์ความต้องการ	19
3.3 การวิเคราะห์และออกแบบระบบ	21
3.4 รายละเอียดการพัฒนาระบบ	49
บทที่ 4 ผลการปฏิบัติงาน	64
4.1 ผลการปฏิบัติงาน	64
บทที่ 5 สรุปผล	82
5.1 ประโยชน์ที่ได้รับจากการปฏิบัติงาน	82
5.2 วิเคราะห์จุดเด่น จุดด้อย โอกาส อุปสรรค(Swot Analysis)	82
5.3 ประสบการณ์ที่ประทับใจ	82
บรรณานุกรม	83
บทที่ ก เรื่องที่หนึ่ง	84

สารบัญตาราง

หน้า

ตารางที่ 3.1 รายละเอียดคุณภาพ ลงทะเบียน	22
ตารางที่ 3.2 รายละเอียดคุณภาพ เข้าสู่ระบบ	23
ตารางที่ 3.3 รายละเอียดคุณภาพ สร้างหมวดหมู่คำถาม	24
ตารางที่ 3.4 รายละเอียดคุณภาพ คุณภาพหมวดหมู่คำถามของผู้ใช้	25
ตารางที่ 3.5 รายละเอียดคุณภาพ แก้ไขหมวดหมู่คำถามของผู้ใช้	26
ตารางที่ 3.6 รายละเอียดคุณภาพ คัดลอกหมวดหมู่คำถามของผู้ใช้	27
ตารางที่ 3.7 รายละเอียดคุณภาพ ลบหมวดหมู่คำถามของผู้ใช้	28
ตารางที่ 3.8 รายละเอียดคุณภาพ คุณภาพหมวดหมู่คำถามสาธารณะ	29
ตารางที่ 3.9 รายละเอียดคุณภาพ คุ้มครองหมวดหมู่คำถามสาธารณะ	30
ตารางที่ 3.10 รายละเอียดคุณภาพ คัดลอกหมวดหมู่คำถามสาธารณะ	31
ตารางที่ 3.11 รายละเอียดคุณภาพ สร้างข้อสอบ	32
ตารางที่ 3.12 รายละเอียดคุณภาพ คูชี้อสอบทั้งหมดของตนเอง	33
ตารางที่ 3.13 รายละเอียดคุณภาพ แก้ไขข้อสอบของตนเอง	34
ตารางที่ 3.14 รายละเอียดคุณภาพ แก้ไขข้อสอบของตนเอง	35
ตารางที่ 3.15 รายละเอียดคุณภาพ ลบข้อสอบ	36
ตารางที่ 3.16 รายละเอียดคุณภาพ คูชี้อสอบสาธารณะ	37
ตารางที่ 3.17 รายละเอียดคุณภาพ คูชี้อสอบสาธารณะ	38
ตารางที่ 3.18 รายละเอียดคุณภาพ คัดลอกข้อสอบสาธารณะ	39
ตารางที่ 3.19 รายละเอียดคุณภาพ ส่งข้อสอบให้ผู้สมัครงาน	40
ตารางที่ 3.20 รายละเอียดคุณภาพ คุ้มครองสิทธิ์การส่งข้อสอบทั้งหมด	41
ตารางที่ 3.21 รายละเอียดคุณภาพ คุ้มครองสิทธิ์การส่งข้อสอบ	42
ตารางที่ 3.22 รายละเอียดคุณภาพ รายละเอียดข้อสอบ	43
ตารางที่ 3.23 รายละเอียดคุณภาพ รายละเอียดข้อสอบ	44
ตารางที่ 3.24 รายละเอียดคุณภาพ ส่งข้อสอบให้ทางบริษัท	45
ตารางที่ 3.25 รายละเอียดคุณภาพ หมวดเวลาทำข้อสอบ	46
ตารางที่ 3.26 รายละเอียดคุณภาพ คูชี้อสอบที่ผู้สมัครงานทำเสร็จแล้ว	47
ตารางที่ 3.27 รายละเอียดคุณภาพ คูชี้อสอบที่ผู้สมัครงานทำเสร็จแล้ว	48
ตารางที่ 3.28 ฐานข้อมูลทั้งหมดที่ใช้	49
ตารางที่ 3.29 รายละเอียดฟังก์ชันสร้างผู้ใช้งาน	50

สารบัญตาราง (ต่อ)

	หน้า
ตารางที่ 3.30 รายละเอียดฟังก์ชันยืนยันตัวตนเพื่อเข้าใช้งานระบบ	50
ตารางที่ 3.31 รายละเอียดฟังก์ชันการเข้าสู่ระบบ	50
ตารางที่ 3.32 รายละเอียดฟังก์ชันแก้ไขข้อมูลผู้ใช้งาน	51
ตารางที่ 3.33 รายละเอียดฟังก์ชันการออกจากระบบ	51
ตารางที่ 3.34 รายละเอียดฟังก์ชันการออกจากระบบทุกเครื่องที่เข้าใช้งาน	51
ตารางที่ 3.35 รายละเอียดฟังก์ชันการลบผู้ใช้งานออกจากระบบ	52
ตารางที่ 3.36 รายละเอียดฟังก์ชันการสร้างหมวดหมู่คำถาม	52
ตารางที่ 3.37 รายละเอียดฟังก์ชันสำหรับเพิ่มรูปภาพกับไฟล์ต่างๆ	52
ตารางที่ 3.38 รายละเอียดฟังก์ชันสำหรับลบรูปภาพกับไฟล์ต่างๆ	53
ตารางที่ 3.39 รายละเอียดฟังก์ชันการแก้ไขหมวดหมู่คำถาม	53
ตารางที่ 3.40 รายละเอียดฟังก์ชันการคุณภาพที่อยู่ภายใต้หมวดหมู่คำถาม	53
ตารางที่ 3.41 รายละเอียดฟังก์ชันการคุณภาพหมวดหมู่คำถามทั้งหมดของตนเอง	54
ตารางที่ 3.42 รายละเอียดฟังก์ชันการคุณภาพหมวดหมู่คำถามทั้งหมดที่เป็นสาธารณะ	54
ตารางที่ 3.43 รายละเอียดฟังก์ชันการคุณภาพหมวดหมู่คำถามทั้งหมดที่ใช้สร้างข้อสอบสาธารณะ	54
ตารางที่ 3.44 รายละเอียดฟังก์ชันการคุณภาพหมวดหมู่คำถามทั้งหมดที่ใช้สร้างข้อสอบที่ไม่เปิดเป็นสาธารณะ	55
ตารางที่ 3.45 รายละเอียดฟังก์ชันการลบหมวดหมู่คำถาม	55
ตารางที่ 3.46 รายละเอียดฟังก์ชันการคัดลอกหมวดหมู่คำถาม	55
ตารางที่ 3.47 รายละเอียดฟังก์ชันการสร้างข้อสอบ	56
ตารางที่ 3.48 รายละเอียดฟังก์ชันการคูรายละเอียดข้อสอบ	56
ตารางที่ 3.49 รายละเอียดฟังก์ชันการแก้ไขรายละเอียดต่างๆภายในข้อสอบ	57
ตารางที่ 3.50 รายละเอียดฟังก์ชันการคูข้อสอบทั้งหมดของตนเอง	57
ตารางที่ 3.51 รายละเอียดฟังก์ชันการคูข้อสอบทั้งหมดที่เป็นสาธารณะ	57
ตารางที่ 3.52 รายละเอียดฟังก์ชันการลบข้อสอบของตนเอง	58
ตารางที่ 3.53 รายละเอียดฟังก์ชันการคัดลอกข้อสอบ	58
ตารางที่ 3.54 รายละเอียดฟังก์ชันการส่งข้อสอบให้ผู้สมัครงานทำ	59
ตารางที่ 3.55 รายละเอียดฟังก์ชันการคูประวัติการส่งข้อสอบให้ผู้สมัครงาน	59

สารบัญตาราง (ต่อ)

	หน้า
ตารางที่ 3.56 รายละเอียดฟังก์ชันการลบประวัติการส่งข้อสอบ	59
ตารางที่ 3.57 รายละเอียดฟังก์ชันการคูณรายละเอียดของข้อสอบก่อนที่ผู้สมัครจะเริ่มทำข้อสอบ	60
ตารางที่ 3.58 รายละเอียดฟังก์ชันการเริ่มทำข้อสอบ	60
ตารางที่ 3.59 รายละเอียดฟังก์ชันการส่งข้อสอบของผู้สมัครงาน	60
ตารางที่ 3.60 รายละเอียดฟังก์ชันการเพิ่มไฟล์	61
ตารางที่ 3.61 รายละเอียดฟังก์ชันการลบไฟล์	61
ตารางที่ 3.62 รายละเอียดฟังก์ชันการคูนข้อสอบทั้งหมด	61
ตารางที่ 3.63 รายละเอียดฟังก์ชันการคูนข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว	62
ตารางที่ 3.64 รายละเอียดฟังก์ชันการยืนยันการตรวจสอบข้อสอบ	62
ตารางที่ 3.65 รายละเอียดฟังก์ชันการลบข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว	62

สารบัญภาพ

หน้า

รูปที่ 1.1 ตราสัญลักษณ์ของ บริษัท ไซเจ็น	2
รูปที่ 1.2 ตัวอย่างการให้บริการของ บริษัท ไซเจ็น	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 Kubernetes	13
รูปที่ 2.9 แสดงตัวอย่างการใช้งาน postman	14
รูปที่ 2.10 แสดงตัวอย่าง jwt endcoded	14
รูปที่ 2.11 แสดงตัวอย่างการใช้งาน Robo 3T	15
รูปที่ 2.12 แสดงตัวอย่างสถิติของอีเมลที่ทำการส่งด้วย sendgrid	16
รูปที่ 2.13 แสดงการออกแบบภาพรวมของระบบด้วย Figma ของนักศึกษา	17
รูปที่ 2.14 แสดง Trello board ของนักศึกษา	17
รูปที่ 3.1 แสดงการทำงานของเว็บแอพพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน	18
รูปที่ 3.2 แผนภาพยุทธศาสตร์ของระบบ	21
รูปที่ 3.3 แสดงตัวอย่างการใช้งาน Figma	63
รูปที่ 3.4 แสดงการเขียน Dockerfile ด้วยการทำ multi-stage build docker	63
รูปที่ 4.1 แสดงหน้าเข้าสู่ระบบของเว็บแอพพลิเคชัน	64
รูปที่ 4.2 แสดงหน้าสมัครสมาชิกสู่ระบบของเว็บแอพพลิเคชัน	65
รูปที่ 4.3 แสดงหน้าหลักของเว็บแอพพลิเคชัน	66
รูปที่ 4.4 แสดงหน้าสร้างหมวดหมู่คำถามของเว็บแอพพลิเคชัน	67
รูปที่ 4.5 แสดงหน้าคูหมวดหมู่คำถามทั้งหมดของเว็บแอพพลิเคชัน	68
รูปที่ 4.6 แสดงหน้าแก้ไขหมวดหมู่คำถามของเว็บแอพพลิเคชัน	69
รูปที่ 4.7 แสดงหน้าคูหมวดหมู่คำถามที่เปิดเป็นสาธารณะของเว็บแอพพลิเคชัน	70
รูปที่ 4.8 แสดงหน้าคูคำถามในหมวดหมู่คำถามที่เปิดเป็นสาธารณะของเว็บแอพพลิเคชัน	71
รูปที่ 4.9 แสดงหน้าสร้างข้อสอบเพื่อใช้สำหรับผู้สมัครงานของเว็บแอพพลิเคชัน	71

สารบัญภาพ (ต่อ)

	หน้า
รูปที่ 4.10 แสดงหน้าดูข้อสอบตอนของเว็บแอพพลิเคชัน	72
รูปที่ 4.11 แสดงหน้าดูข้อสอบตอนในขั้นตอนส่งข้อสอบให้ผู้สมัครงานของเว็บแอพพลิเคชัน	73
รูปที่ 4.12 แสดงคุณรายละเอียดก่อนเริ่มทำข้อสอบของเว็บแอพพลิเคชัน	74
รูปที่ 4.13 แสดงหน้าทำข้อสอบของเว็บแอพพลิเคชัน	75
รูปที่ 4.14 แสดงหน้าแก้ไขข้อสอบเพื่อใช้สำหรับส่งให้ผู้สมัครงานของเว็บแอพพลิเคชัน	76
รูปที่ 4.15 แสดงหน้าดูข้อสอบเพื่อใช้สำหรับส่งให้ผู้สมัครงานของเว็บแอพพลิเคชัน	76
รูปที่ 4.16 แสดงหน้าดูข้อสอบที่เป็นข้อสอบสาธารณะของเว็บแอพพลิเคชัน	77
รูปที่ 4.17 แสดงหน้าดูรายละเอียดข้อสอบที่เป็นข้อสอบสาธารณะเว็บแอพพลิเคชัน	77
รูปที่ 4.18 แสดงหน้าดูประวัติการส่งข้อสอบของเว็บแอพพลิเคชัน	78
รูปที่ 4.19 แสดงหน้าดูข้อสอบที่ผู้สมัครงานทำเสร็จแล้วของเว็บแอพพลิเคชัน	79
รูปที่ 4.20 แสดงหน้าตรวจสอบของผู้สมัครงานของเว็บแอพพลิเคชัน	80
รูปที่ 4.21 แสดงหน้าตรวจข้อสอบอัตโนมัติของผู้สมัครงานของเว็บแอพพลิเคชัน	80

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

บริษัท ไซเจ้น เป็นบริษัทที่บริการและให้คำปรึกษาและบริการโซลูชัน SAP ในองค์กรต่างๆ มากมาย และพัฒนาคิดค้นนวัตกรรมใหม่ๆ เพื่อช่วยแก้ปัญหาในองค์กร ทำให้ธุรกิจสามารถดำเนินไปได้อย่าง ราบรื่นมากยิ่งขึ้น เช่น การทำ SAP BUSINESS ONE ช่วยทำให้เห็นภาพรวมของการทำธุรกิจได้อย่าง ชัดเจน, การทำ RPA (Robotic Process Automation) โronoth ที่จะมาช่วยทำงานออฟฟิศแทนมนุษย์ และ BUDDY RECRUIT เทคโนโลยีที่ช่วยให้หางานเข้ามายังงานได้ตรงตามความต้องการของบริษัทนั้นๆ

การรับสมัครงาน เป็นกระบวนการที่ทุกๆ บริษัทต้องมีเพื่อที่จะรับพนักงาน ที่มีความสามารถตรง ตามที่บริษัทนั้นๆ ต้องการ ในปัจจุบันกระบวนการรับสมัครงานของแต่ละบริษัทส่วนใหญ่ ก่อนที่จะ มีการสัมภาษณ์งานเกิดขึ้น ทางบริษัทจะคุยกับผู้สมัครผ่านทางเรซูเม่ เพื่อคุ้มครองการ ทำงาน และทักษะต่างๆ ที่ผู้สมัครมี แต่ถ้าหากมีผู้สมัครเป็นจำนวนมาก อาจทำให้ใช้ระยะเวลาในการคัด กรองผู้สมัครงานที่มากขึ้นตามไปด้วย และผู้ที่มาสมัครนั้นอาจมีความสามารถไม่ตรงตามความต้องการ ของบริษัท จึงมีการสร้างเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน เว็บแอปพลิ เคชันนี้จะมาช่วยในการคัดกรองผู้สมัครงาน โดยการให้ทางบริษัทสร้างข้อสอบในเรื่องที่ต้องการ วัด ความรู้ความสามารถ มาใช้ทดสอบผู้สมัครงาน เพื่อเป็นการลดระยะเวลาในการคัดเลือกพนักงาน และ ทดสอบว่าผู้สมัครงานนั้นมีความรู้ความสามารถตรงตามที่บริษัทกำหนด ซึ่งทางบริษัทได้มีระบบดัง กล่าวมา ก่อนหน้านี้แล้ว แต่การส่งข้อสอบในแต่ละครั้งทางบริษัทต้องออกข้อสอบใหม่ทุกครั้ง เพื่อ ป้องกันข้อสอบที่อาจถูกเผยแพร่ ซึ่งทำให้การทดสอบนั้นมีประสิทธิภาพลง พนักงานในบริษัทจึง ต้องสละเวลาามาออกข้อสอบทุกครั้ง

ในการปฏิบัติงานครั้นนี้ นักศึกษาได้เป็นในสมาชิกของทีม BUDDY RECRUIT ซึ่งเป็นทีมที่สร้าง เทคโนโลยีที่ช่วยให้บริษัทหางานเข้ามายังงานได้ตรงตามความต้องการของบริษัทนั้นๆ โดยได้รับมอบ หมายงานให้ทำเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงานรูปแบบใหม่ ที่สามารถ สุ่มคำถามจากคลังข้อสอบได้ โดยสุ่มตามระดับความยากที่ผู้ออกข้อสอบเป็นคนกำหนด ด้วยวิธีนี้จะ ช่วยให้ลดระยะเวลาในการให้พนักงานมาออกข้อสอบใหม่ทุกครั้งที่มีผู้สมัครงาน

1.2 วัตถุประสงค์การปฏิบัติงาน

- ศึกษาเทคโนโลยีใหม่ๆ ที่ใช้ในการทำงาน
- พัฒนาระบวนการคิด และการลงมือทำในสายงาน Full Stack Developer
- พัฒนาทักษะการทำงานภายใต้แรงกดดัน และระยะเวลาที่จำกัด
- พัฒนาทักษะการแก้ไขปัญหาเฉพาะหน้า

5. ศึกษากระบวนการทำงานภายในทีม
6. พัฒนาทักษะการสื่อสาร
7. นำความรู้ที่ได้ไปต่อยอดกับงานในอนาคต

1.3 ประวัติ และรายละเอียดบริษัท

1.3.1 ชื่อสถานประกอบการ

ชื่อบริษัท(ภาษาไทย): บริษัท ไซเจ็น จำกัด
 ชื่อบริษัท(ภาษาอังกฤษ): ZyGen Company Limited

1.3.2 สถานที่ตั้ง

65/60 ชั้น 6 อาคาร ชานาณเพลู ชาติบิสเนสเซ็นเตอร์ ถนนพระราม 9 เขตห้วยขวาง, กรุงเทพ -
 ขนาดคร 10310

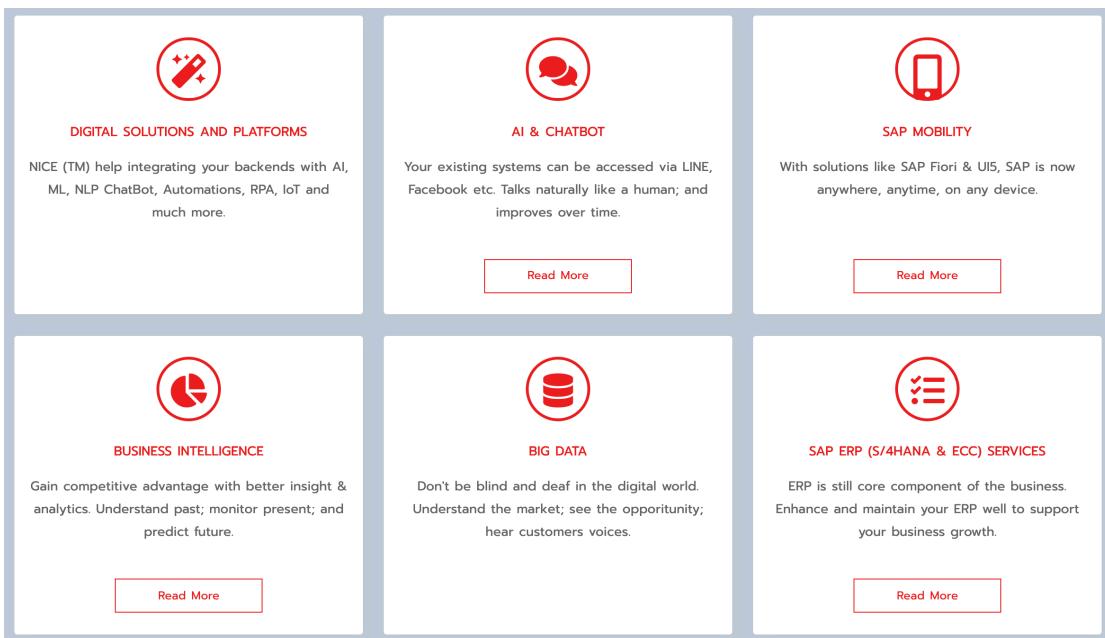
1.3.3 ลักษณะสถานประกอบการ

บริษัท ไซเจ็น เป็นบริษัทที่บริการและให้คำปรึกษาและบริการโซลูชัน SAP ในองค์กรต่างๆ มากมาย และพัฒนาคิดค้นนวัตกรรมใหม่ๆ เพื่อช่วยแก้ปัญหาในองค์กร ทำให้ธุรกิจสามารถดำเนินไปได้อย่าง ราบรื่นมากยิ่งขึ้น ไซเจ็น เป็นบริษัทที่มีฐานลูกค้าที่มั่นคง และยังเพิ่มฐานลูกค้าต่อไป ด้วยการให้ บริการของ ไซเจ็น เช่น การทำ SAP BUSINESS ONE ช่วยทำให้เห็นภาพรวมของการทำธุรกิจได้อย่าง ชัดเจน, การทำ RPA (Robotic Process Automation) โронอที่จะมาช่วยทำงานออฟฟิศแทนมนุษย์ และ BUDDY RECRUIT เทคโนโลยีที่ช่วยให้บริษัทหางานเข้ามาทำงานได้ตรงตามความต้องการของบริษัท นั่นๆ



ZYGEN

รูปที่ 1.1: ตราสัญลักษณ์ของ บริษัท ไซเจ็น



รูปที่ 1.2: ตัวอย่างการให้บริการของ บริษัท ไซเจ็น

1.3.4 ตำแหน่งและลักษณะงานที่นักศึกษาได้รับมอบหมายให้รับผิดชอบ

ตำแหน่ง: Full Stack Developer

หน้าที่: พัฒนาเว็บแอปพลิเคชันตามความต้องการของลูกค้า

1.3.5 ชื่อและตำแหน่งของพนักงานที่ปรึกษา

ชื่อ-นามสกุล: นายวิวัฒน์ เสิงอนันต์

ตำแหน่ง: Web Developer

แผนก: Development Team

1.3.6 ระยะเวลาปฏิบัติงาน

ช่วงเวลาปฏิบัติงาน: 1 มิถุนายน 2563 - 30 กันยายน 2563

ช่วงเวลาปฏิบัติงาน: จันทร์ - สุกรี เวลา 09:00 น. - 18:00 น.

รวมระยะเวลา: 4 เดือน

บทที่ 2

รายละเอียดการปฏิบัติงาน

2.1 ตำแหน่งและลักษณะงานที่นักศึกษาได้รับมอบหมาย

ตำแหน่ง: Full Stack Developer

2.1.1 งานที่ได้รับผิดชอบ

- ศึกษาเทคโนโลยีที่มีคุณภาพเพื่อประยุกต์ใช้กับการทำงาน
- พัฒนาเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน
- ควบคุมคุณภาพของโค้ดให้มีคุณภาพที่ดี

2.2 รายละเอียดของโครงการที่ได้รับผิดชอบ

เนื่องจากนักศึกษาได้รับหน้าที่ในตำแหน่ง Developer ในทีม BUDDY RECRUIT ซึ่งทำเกี่ยวกับการช่วยให้บริษัทหางานเข้ามาทำงาน ได้ตรงตามความต้องการของบริษัท จึงได้รับมอบหมายให้ทำเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน โดยเว็บแอปพลิเคชันนี้ช่วยให้คัดกรองผู้คนได้มีประสิทธิภาพมากยิ่งขึ้น ผ่านการทำข้อสอบที่สามารถกำหนดระดับความยากของข้อสอบได้ และดึงคลังคำามนາแบบสุ่ม ส่งให้ผู้ทำข้อสอบโดยอัตโนมัติผ่านทางอีเมล เมื่อผู้สมัครงานทำข้อสอบเสร็จแล้ว ข้อสอบจะถูกส่งไปที่ผู้ออกข้อสอบ พร้อมตรวจข้อที่เป็นคำถามปนัยให้อัตโนมัติ โดยระบบนี้แบ่งเป็น 2 ส่วนหลักๆ ได้แก่

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

อนึ่ง ข้อมูลข้างต้นเป็นเพียงภาพรวมของเว็บแอปพลิเคชันโดยย่อ ซึ่งรายละเอียดของเว็บแอปพลิเคชันนี้จะกล่าวโดยละเอียดในบทต่อไป

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 ทำให้สามารถนำรูปภาพมาจัดเรียงได้ [1]

- Client-server architecture: Client ผู้ที่เข้ามายื่นคำขอ resources ไม่ต้องรู้ Business logic ภายใน ส่วน Server มีหน้าที่เก็บข้อมูล ไม่จำเป็นต้องรู้เกี่ยวกับ UI หรือสถานะของผู้เรียก
- Stateless: ส่ง request ให้เซิร์ฟเวอร์แล้วรับกลับมาเป็น response เมื่อรับ response มาแล้วจะทำการทำงาน
- Cache: สามารถกำหนดได้ว่าจะเก็บ cache ของ response นั้นหรือไม่
- Layered system: สามารถปรับปรุงความสามารถในการขยายระบบได้ โดยการใช้งานการทำ Load balance
- Interface/Uniform Contract: วิธีการที่จะคุยกับเซิร์ฟเวอร์โดยไม่คำนึงถึงประเภทของอุปกรณ์ หรือประเภทของแอพพลิเคชัน

2.3.2 HTTP Request

HTTP คือ protocol ที่อนุญาติให้โคลอนต์ดึงข้อมูลจากเซิร์ฟเวอร์ [1]

- Request-Line คือส่วนที่ระบุ HTTP Method, Request-URI และ version ของ protocol เช่น HTTP/1.0, HTTP/1.1, HTTP/2.0
- Headers คือส่วนที่อนุญาติให้ใส่ข้อมูลเพิ่มเติม 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 ตามที่คุณต้องการ โดยกำหนดขอบเขตที่ต้องการผ่าน headers บน HTTP Request

- 3xx (Redirection)

request ที่ไคลเอนต์ส่งไปหาเซิร์ฟเวอร์ แล้วถูก redirect ส่งไปประมวลผลที่อื่น เพื่อทำให้กระบวนการทำงานสำเร็จ

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

301 : (Moved Permanently) URL ที่ทำการ request ขอข้อมูลถูกคุปลี่ยนไปจาว จึง response ออกมายัง 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 หรือไม่ถูกรูปแบบ ทำให้เซิร์ฟเวอร์ไม่เข้าใจ

401 : (Unauthorized) ไคลเอนต์ต้องทำการยืนยันตัวตนก่อนที่จะได้รับ response

403 : (Forbidden) ไคลเอนต์ทำการยืนยันตัวตนแล้ว แต่ไม่มีสิทธิ์ในการเข้าถึงข้อมูลนี้

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

405 : (Method Not Allowed) method ที่เรียกใช้ไม่ถูกต้อง

406 : (Not Acceptable) header ที่โคลอนต์ request ไม่สัมพันธ์กับเซิร์ฟเวอร์

413 : (Payload Too Large) request ที่ขอใหญ่กว่า limit ที่เซิร์ฟเวอร์กำหนดไว้

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

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

- 5XX (Server Error)

เซิร์ฟเวอร์มีปัญหา

500 : (Internal Server Error) เซิร์ฟเวอร์เจอกับสถานการณ์ที่ไม่สามารถจัดการได้

501 : (Not Implemented) โคลอนต์เรียก request method ที่เซิร์ฟเวอร์ไม่รองรับ และเซิร์ฟเวอร์ ไม่สามารถจัดการได้

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

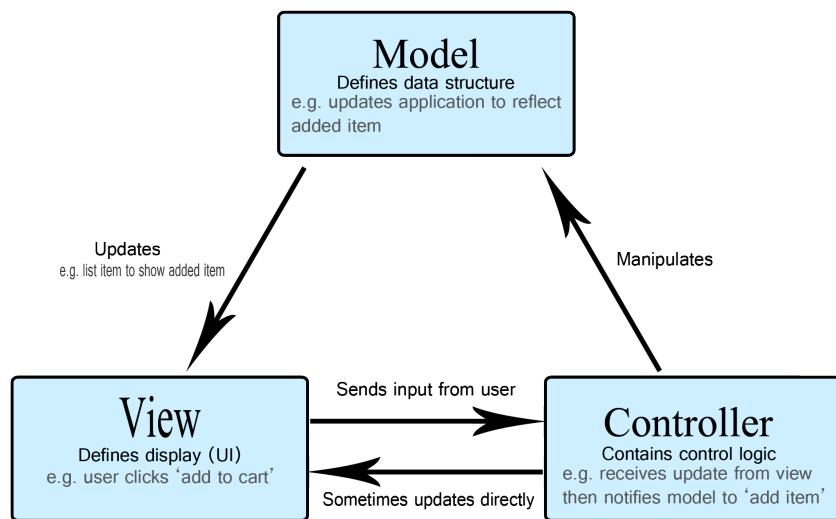
503 : (Service Unavailable) เซิร์ฟเวอร์อยู่ระหว่างการปรับปรุง หรือยังไม่พร้อมที่จะจัดการ request

504 : (Gateway Timeout) เซิร์ฟเวอร์เป็น gateway และไม่สามารถ response ข้อมูลในเวลาที่กำหนดได้

2.3.5 MVC (Model View Controller)

คือ software design pattern ที่แยกการทำงานของโปรแกรมออกเป็น 3 ส่วนเพื่อแยกข้อมูลภายใน โปรแกรมกับข้อมูลที่แสดงให้ผู้ใช้เห็น [3]

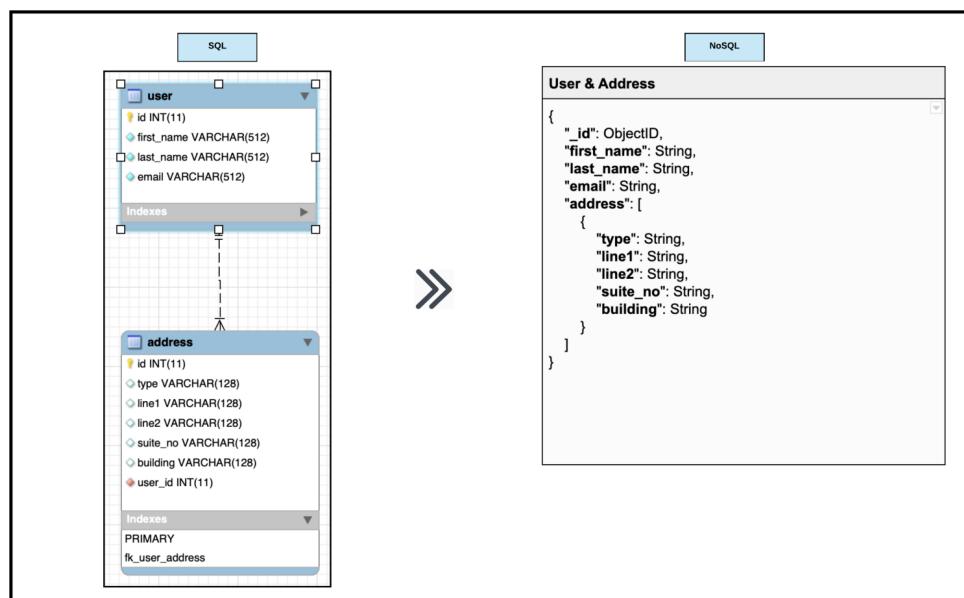
- Model คือส่วนที่เป็นโครงสร้างของข้อมูล กำหนดกฎเกณฑ์ของการเก็บข้อมูล และเป็นส่วนที่ ไว้สำหรับการจัดการข้อมูลโดยตรง
- View คือส่วนที่ไว้แสดงผลตามที่ผู้ใช้ต้องการ ผู้ใช้สามารถเปลี่ยนแปลงข้อมูลที่ผู้ใช้เห็นได้
- Controller คือส่วนที่ไว้จัดการกับ Model โดยขึ้นอยู่กับการกระทำของ View ที่กำหนดโดยผู้ใช้ และสร้างข้อมูลจาก Model เพื่อไปแสดงใน View



รูปที่ 2.1: แสดง MVC architecture

2.3.6 NoSQL Database

คือ ฐานข้อมูลที่สร้างมาเพื่อให้จัดการข้อมูลที่มีความซับซ้อนได้ง่ายขึ้นเมื่อเทียบกับ SQL databases ที่มีการเก็บข้อมูลในที่รูปแบบแน่นอน (structured data) โดยเพิ่มความสามารถในการจัดเก็บข้อมูลในรูปแบบที่ไม่แน่นอน (unstructured data) ทำให้สามารถเก็บข้อมูลที่ซับซ้อนได้, เพิ่มความสามารถในการขยายระบบในรูปแบบแนวนอน (Horizontal Scalability) เพื่อรับปริมาณข้อมูลในปัจจุบัน [4]



รูปที่ 2.2: แสดงตาราง SQL เปรียบเทียบกับตาราง NoSQL

2.3.7 Token

เป็นชุดรหัสเอาไว้ระบุตัวตนของผู้ใช้ว่าผู้ใช้นั้นเป็นใคร ไม่มีรูปแบบที่ตายตัว

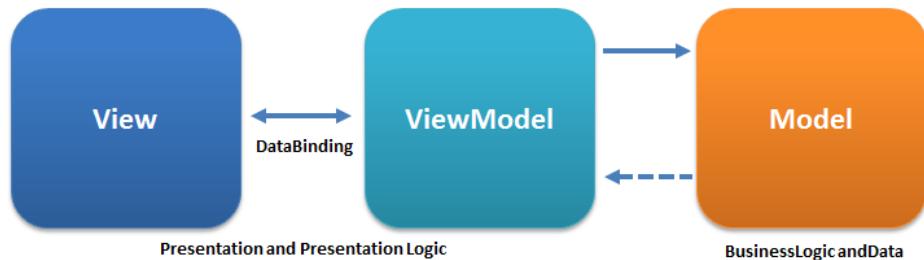
2.3.8 Container

เป็นหน่วยของซอฟแวร์ที่ทำการลงทะเบียนทุกอย่างที่ต้องใช้ในแอพ และตัวโค้ดของแอพ ดังนั้นแอปพลิเคชันจะสามารถถูกเรียกใช้ได้อย่างรวดเร็ว และใช้ในสภาพแวดล้อมใดก็สามารถทำงานได้โดยมีความเป็นมาตรฐาน และประยุคทรัพยากรที่ใช้งานตัวแอปพลิเคชัน [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

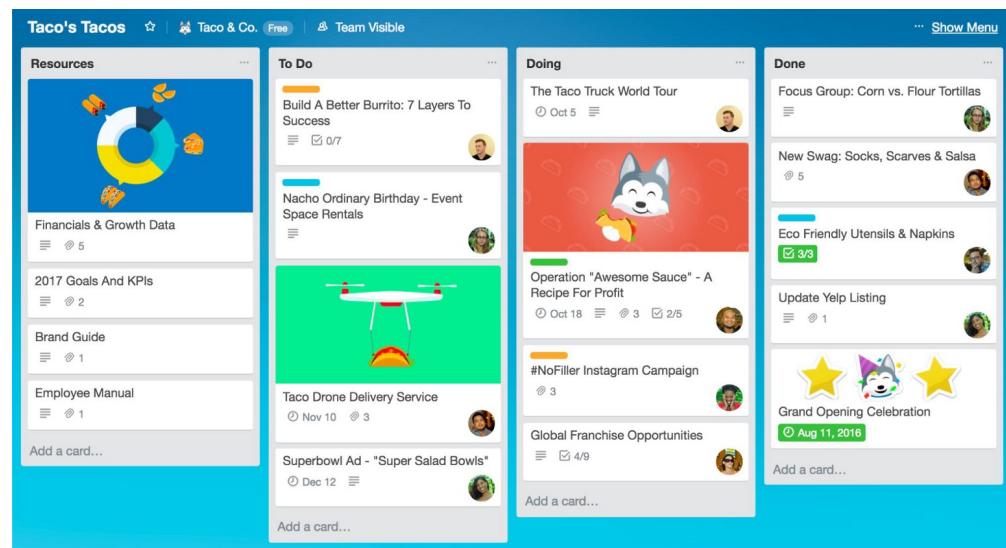


รูปที่ 2.3: แสดง MVVM architecture

2.4 เครื่องมือและเทคโนโลยีที่ใช้ในการปฏิบัติงาน

2.4.1 Trello

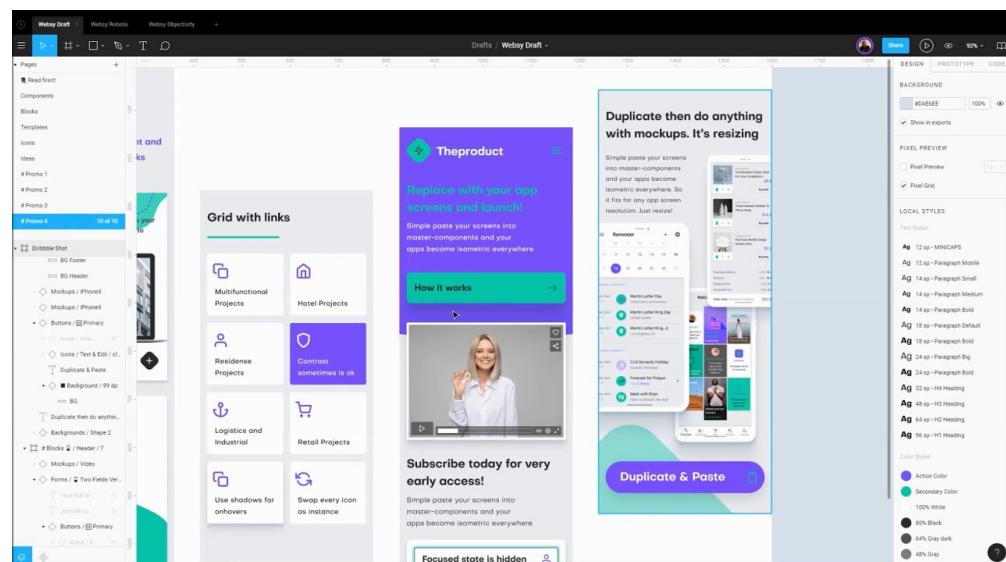
เป็นเว็บแอปพลิเคชันที่ช่วยจัดการบริหารโครงการ ให้เป็นระเบียบได้อย่างเรียบง่าย เปรียบเสมือนกระดานไว้วางแผนการทำงาน ระบุรายละเอียดของงานแต่ละงาน โดยเว็บแอปพลิเคชันนี้สามารถใช้งานร่วมกันภายใต้ทีมได้ ทำให้การทำงานต่างๆในโครงการมีความคล่องตัวมากยิ่งขึ้น



รูปที่ 2.4: แสดงตัวอย่างการใช้งาน Trello

2.4.2 Figma

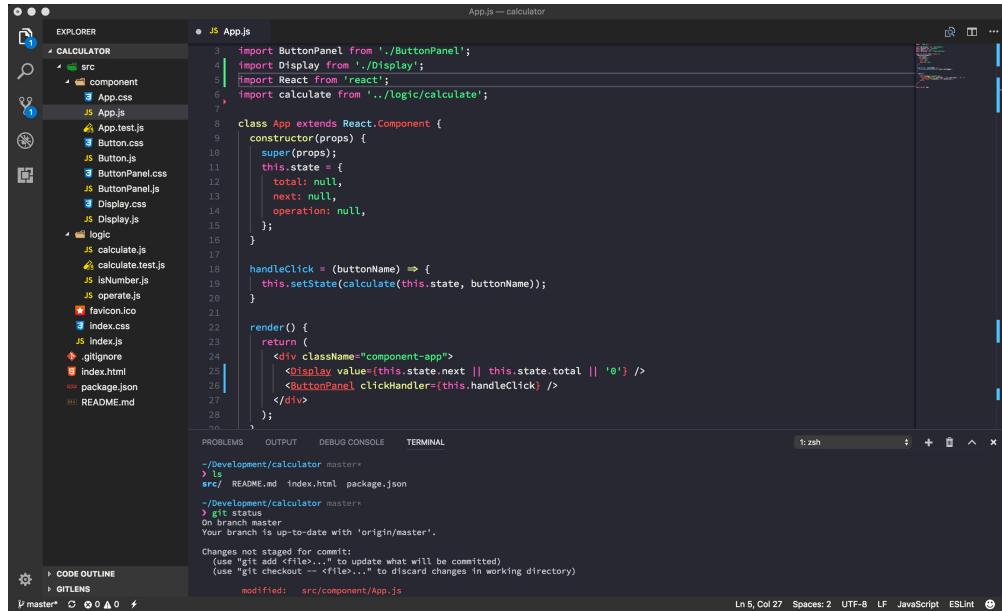
เป็นเครื่องมือที่นำมาใช้ในการออกแบบ prototype ของงานที่ทำ โดยสามารถใช้งานแอปพลิเคชัน Figma ได้ในทุกแพลตฟอร์ม



รูปที่ 2.5: แสดงตัวอย่างการใช้งาน Figma

2.4.3 Visual Studio Code

เป็นโปรแกรม code editor ที่ใช้ในการเขียน แก้ไข และปรับแต่งโค้ด โดยพัฒนาออกแบบในรูปแบบ OpenSource โดยมีความสามารถในการเปลี่ยนสี Syntax ของโค้ด ตามแต่ละภาษา เพื่อให้ง่ายต่อการมอง และการเขียนโค้ด ซึ่งสนับสนุนภาษาหลาย ยกตัวอย่าง เช่น ภาษา C++, Java, JavaScript, PHP, Python เป็นต้น และยังสามารถติดต่อส่วนขยายที่เพิ่มความสามารถด้านภาษาในการเขียนโค้ด 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 เพียงครั้งเดียวแต่สามารถเรียกข้อมูลอื่นๆ แบบ dynamic ได้ ทำให้ประสบการณ์การใช้งานเว็บใกล้เคียง native app มากยิ่งขึ้น

2.4.5 Vuex

เป็น state management pattern ที่ผสมผสานกับ library เพื่อใช้สำหรับจัดการ state เพื่อให้ data flow ของโปรเจคไปในทิศทางเดียวกัน code จึงมีความเป็นระบบมากยิ่งขึ้น และลดการเขียน code ที่ซับซ้อน 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 ซึ่งมีฟิลเจอร์ต่างๆ ที่ช่วยให้พัฒนาเชิร์ฟเวอร์ด้วย Node.js ได้สะดวกมากยิ่งขึ้น เช่นการจัดการ request และ response

2.4.8 Git

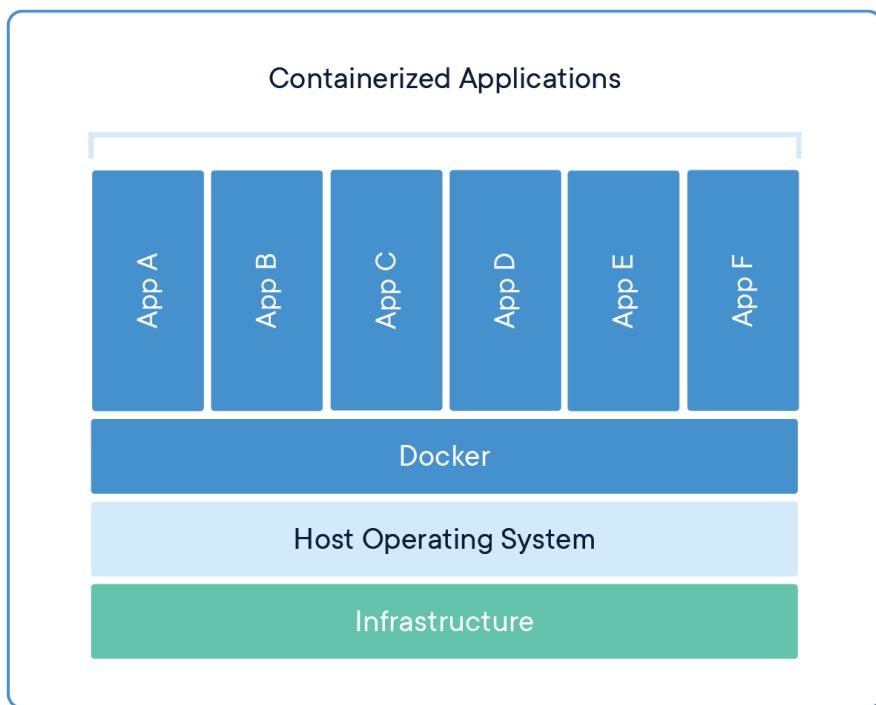
เป็น Version control ที่เอาไว้ติดตาม และควบคุมการเปลี่ยนแปลงของโค้ดเพื่อให้นักพัฒนา สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพ [8]

2.4.9 Github

เป็น website Git (version control repository) สำหรับนักพัฒนาซอฟแวร์ ใช้หลักการทำงานของ git แต่สามารถใช้งานร่วมกับผู้อื่นได้ผ่านอินเตอร์เน็ต

2.4.10 Docker

เป็น engine ที่มีการจำลองสภาพแวดล้อมขึ้นมาบนเครื่องเซิร์ฟเวอร์ เพื่อทำการนำ service ที่ต้องการมาทำงานอยู่บนเซิร์ฟเวอร์ โดยใช้หลักการ container ใน การจำลองสภาพแวดล้อมขึ้นมาทำงาน service ของเราโดยที่ไม่ต้องมีการนำ os เข้ามาใช้ ทำให้ service ทำงานในสภาพแวดล้อมได้ก็ได้ [5]

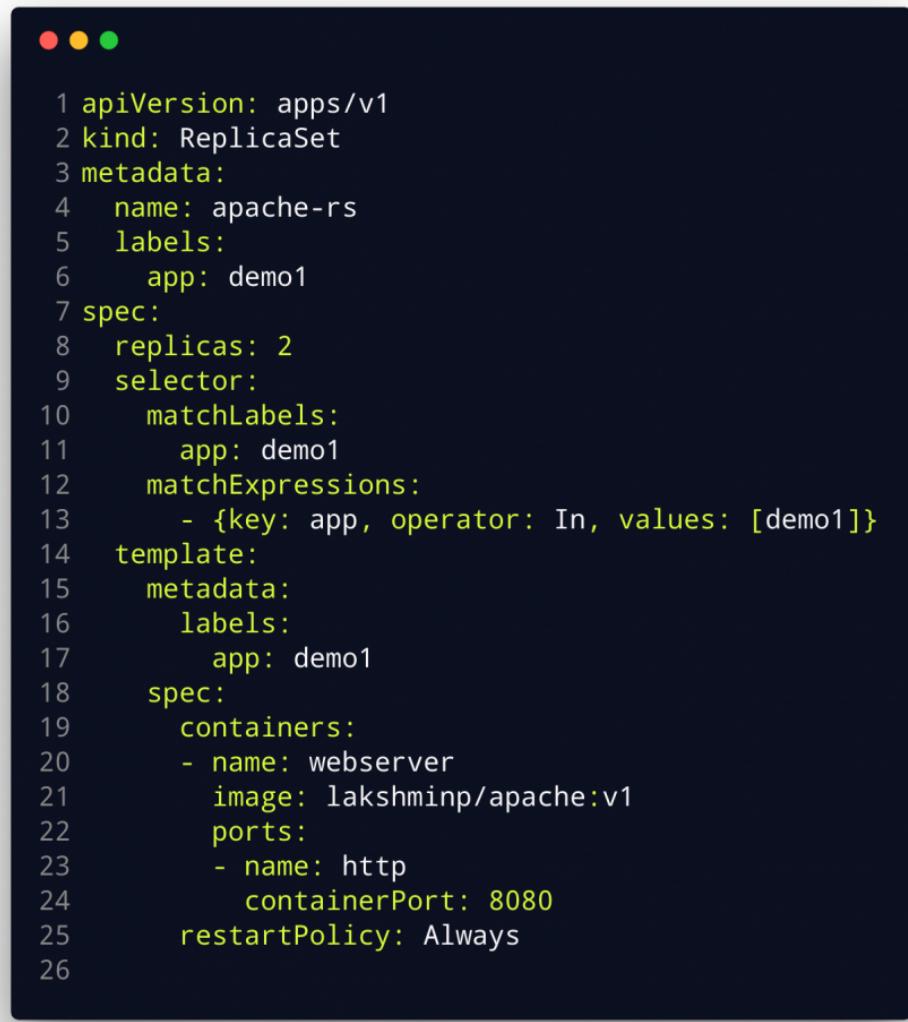


รูปที่ 2.7: แสดงตัวอย่างการทำงานของ docker

2.4.11 Kubernetes

Kubernetes เป็น Orchestration ที่มาช่วยในการ จัดการคอนเทนเนอร์ ซึ่งหน้าที่หลักๆ ก็คือช่วยให้สามารถติดตั้ง(Deployment) จัดสรรทรัพยากรหรือเพิ่มลดทรัพยากรแบบอัตโนมัติได้(Managing & Scaling)

และสามารถใช้งานได้อย่างต่อเนื่องตามที่เราต้องการ ด้วยระบบที่พร้อมใช้งานตลอดเวลา(Auto Self-Healing) โดยในการตั้งค่าต่างๆ ให้กับ Kubernetes สามารถทำได้ด้วยการสร้างไฟล์ .yaml หรือ .yml โดยรูปที่ 2.8 จะเป็นตัวอย่างของไฟล์ .yaml ที่เอาไว้ตั้งค่า Kubernetes [9]



```

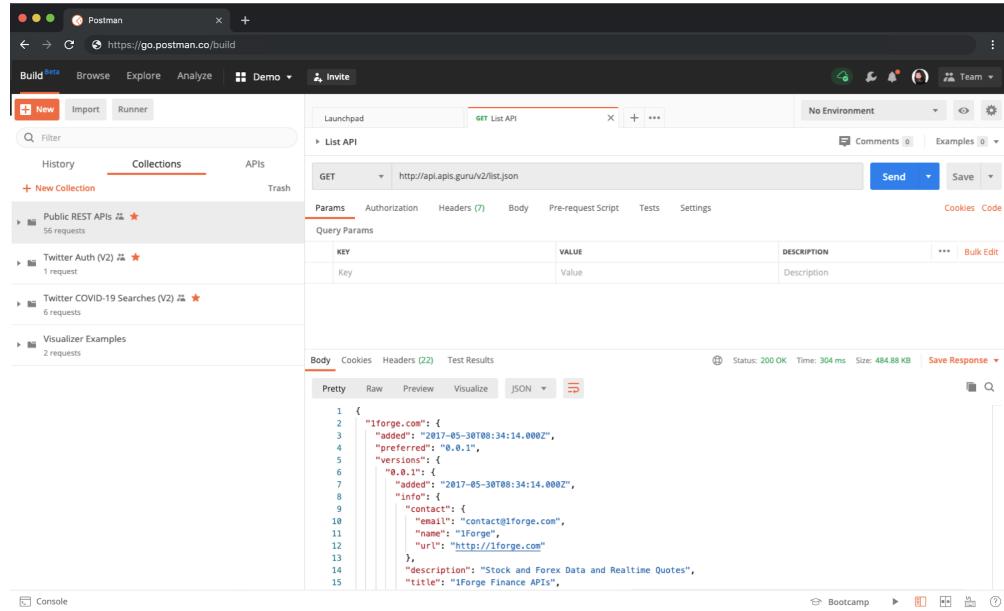
1 apiVersion: apps/v1
2 kind: ReplicaSet
3 metadata:
4   name: apache-rs
5   labels:
6     app: demo1
7 spec:
8   replicas: 2
9   selector:
10    matchLabels:
11      app: demo1
12    matchExpressions:
13      - {key: app, operator: In, values: [demo1]}
14   template:
15     metadata:
16       labels:
17         app: demo1
18     spec:
19       containers:
20         - name: webserver
21           image: lakshminp/apache:v1
22         ports:
23           - name: http
24             containerPort: 8080
25   restartPolicy: Always
26

```

รูปที่ 2.8: แสดงตัวอย่างการใช้งาน Kubernetes

2.4.12 Postman

เป็นเครื่องมือที่มาช่วยในการ API เพื่อทดสอบการทำงานของ Service โดยมีหน้าตาส่วนต่อประสานกับผู้ใช้(UI) ที่สวยงามและใช้งานง่าย



รูปที่ 2.9: แสดงตัวอย่างการใช้งาน postman

2.4.13 JWT (Json Web Token)

เป็นมาตรฐาน RFC 7519 ในการยืนยันตัวตน (Authentication) ที่เข้ามาแก่ปัญหาการส่งข้อมูลระหว่างกันในวิธีแบบดั้งเดิมคือ Server Based Authentication ที่เปลืองทรัพยากรในการเก็บ Session ID และไม่รองรับการขยายตัว (Scalability) โดย JWT นั้นสามารถเก็บข้อมูลภายในตัวได้ และมีขนาดที่กระหัดรัด เพื่อนำมาใช้กับ Single Page Web Application (SPA) [10] โดย JWT แบ่งโครงสร้างออกเป็น 3 ส่วน

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



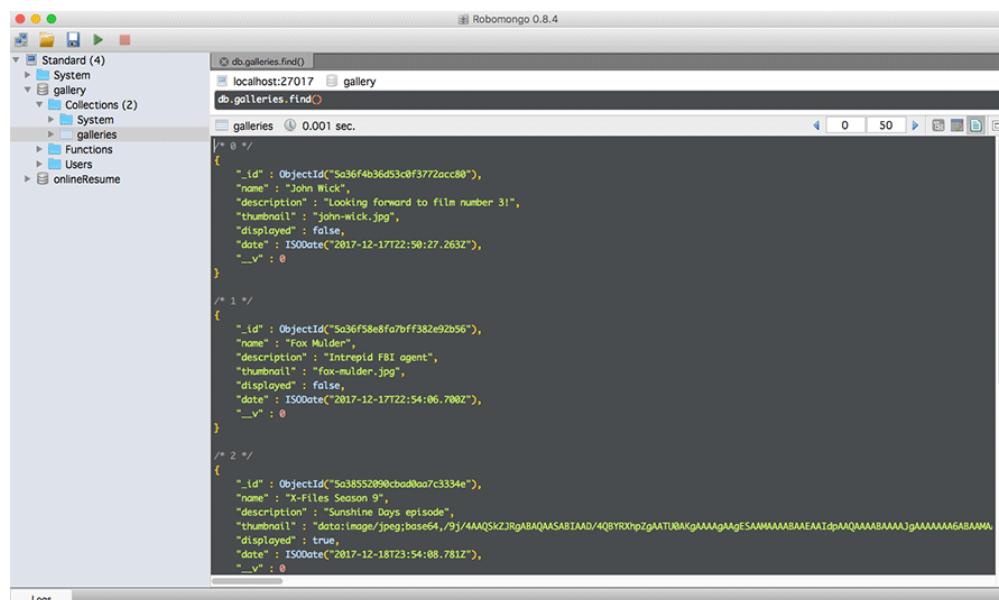
รูปที่ 2.10: แสดงตัวอย่าง jwt encoded

2.4.14 MongoDB

เป็น open-source document database ประเภทหนึ่ง โดยเก็บข้อมูลแบบ NoSQL Database และเก็บข้อมูลในรูปแบบของ JSON (JavaScript Object Notation) ซึ่งเก็บเป็น key และ value ซึ่งมีสมรรถภาพสูงกว่าการเก็บด้วยโครงสร้างแบบเรขาและหลัก(row/column) แบบดั้งเดิม [11]

2.4.15 Robo 3T

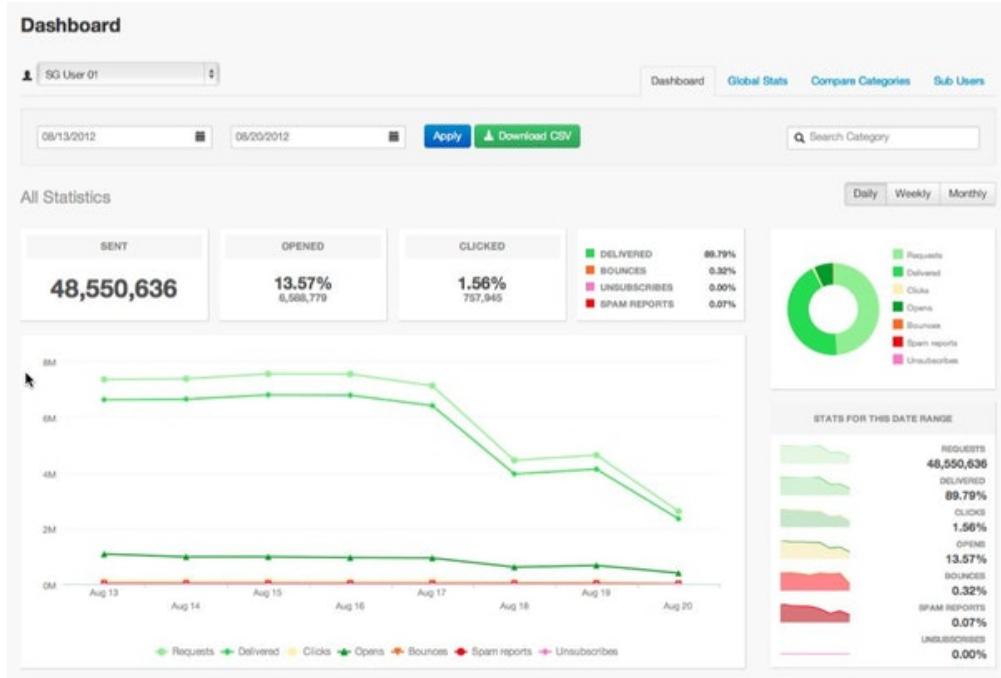
เป็นการใช้ภาพเป็นตัวประสานกับผู้ใช้งาน(GUI) ของ MongoDB ช่วยทำให้ใช้งาน MongoDB ได้สะดวกมากยิ่งขึ้น เช่น เขียน SQL เพื่อ query ข้อมูลใน MongoDB หรือ Import และ export ไฟล์ข้อมูลเป็น CSV, JSON, SQL and BSON/mongodump ได้ [?]



รูปที่ 2.11: แสดงตัวอย่างการใช้งาน Robo 3T

2.4.16 SendGrid

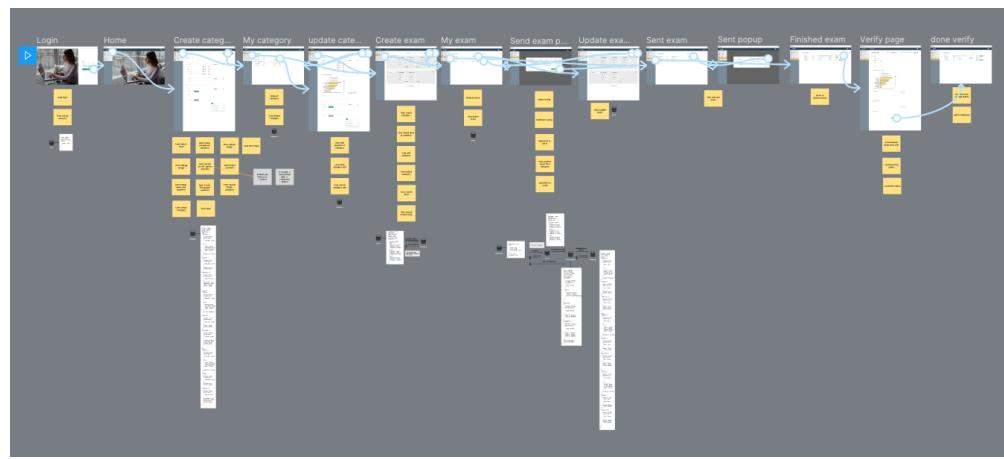
เป็น API ช่วยในการส่งอีเมลให้ผู้อื่น สามารถตรวจสอบอีเมลที่ส่งว่าส่งไปถึงหรือไม่ มีปัญหาอะไรเกินขีนหรือเปล่า และมีการเก็บสถิติข้อมูลอีเมลที่ส่ง สรุปอุปกรณ์แสดงให้วิเคราะห์ เช่น มีการเปิดเมลอ่านกี่ครั้ง มีการรายงานว่าเป็นสแปมกี่ครั้ง



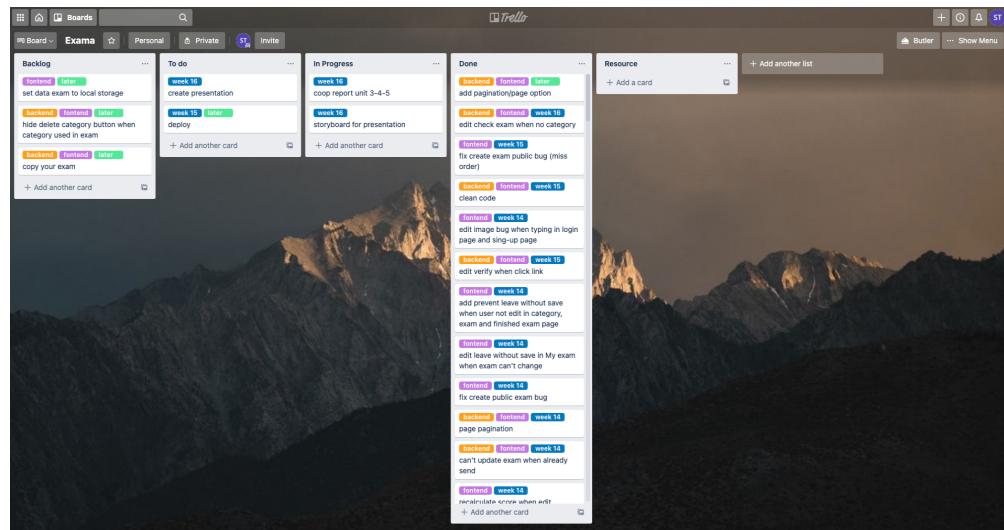
รูปที่ 2.12: แสดงตัวอย่างสถิติของอีเมลที่ทำการส่งด้วย sendgrid

2.5 ลักษณะขั้นตอนการทำงาน

ในขั้นตอนแรก จะมีการรับ Requirement จากพี่ๆ ในทีม ที่ต้องการสร้างเว็บแอปพลิเคชันประเมิน ความสามารถเบื้องต้นของผู้สมัครงาน เพื่อมาทดสอบระบบเดิม เมื่อรับ Requirement มาแล้ว นักศึกษา ก็ทำการวิเคราะห์และออกแบบภาพรวมของระบบด้วยโปรแกรม Figma ดังรูป 2.13 เพื่อนำเสนอว่าเว็บ แอปพลิเคชันจะออกแบบมาหน้าตาประมาณไหน มีฟังก์ชันการทำงานอย่างไรบ้าง ซึ่งนักศึกษาได้ออกแบบ ตามการวิเคราะห์จาก Requirement ที่ได้รับ และให้พี่ๆ ในทีมแสดงความคิดเห็น เพื่อปรับแต่งแก้ไข เมื่อการออกแบบลงตัว และ นักศึกษาทำการแบ่งงานออกแบบเป็นหลายงานย่อย โดยใช้โปรแกรม Trello ดังรูป 2.14 ในการแทก Tasks ต่างๆ โดยนำ Tasks ทั้งหมดไปเก็บไว้ที่ Backlogs เมื่อแทก Tasks เสร็จ และ นักศึกษาจึงทำการวางแผนการทำงานในสัปดาห์นั้นๆ ว่าต้องการทำอะไรบ้าง โดยนำ Tasks จาก Backlogs มาใส่ลงในช่อง To do และถ่ายงานใหม่อยู่ในขั้นตอนกำลังดำเนินงาน งานนั้น จะถูกใส่ลงใน ช่อง In Progress เมื่อทำงานตามที่วางแผนในแต่ละสัปดาห์แล้วลิงจะขยายนไปที่ช่อง Done ในทุกวัน ศุกร์ จะมีการรายงานผลการทำงานให้ พี่ๆ ในทีม เพื่อรายงานความคืบหน้า และ รับฟังความคิดเห็นเพื่อ นำมาปรับแก้ไขกับเว็บแอปพลิเคชัน



รูปที่ 2.13: แสดงการออกแบบภาพรวมของระบบด้วย Figma ของนักศึกษา



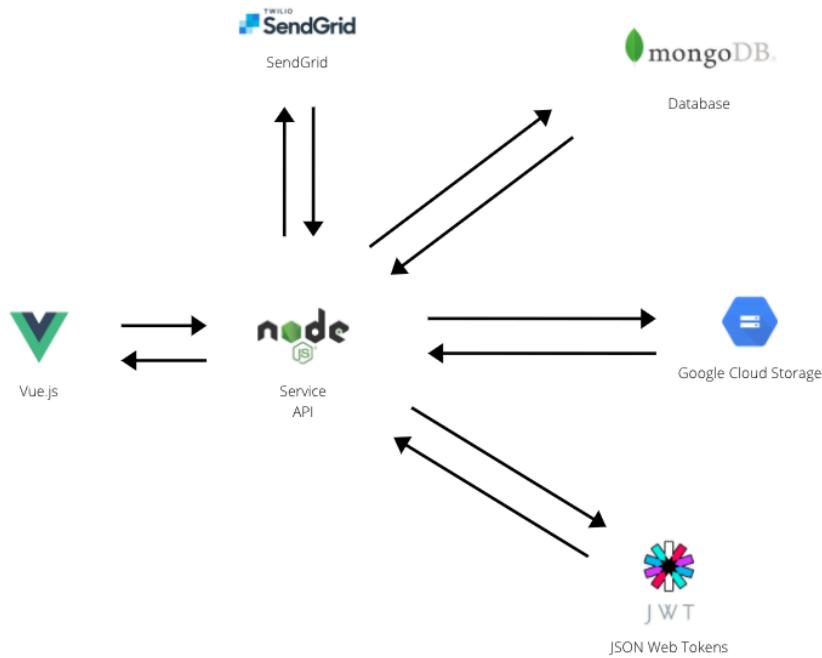
รูปที่ 2.14: แสดง Trello board ของนักศึกษา

บทที่ 3

การออกแบบระบบและรายละเอียดการพัฒนา

3.1 ภาพรวมของเว็บแอปพลิเคชัน

เว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน คือระบบที่จะช่วยให้คัดกรองผู้คนได้มีประสิทธิภาพมากยิ่งขึ้น ผ่านการทำข้อสอบที่สามารถกำหนดระดับความยากของข้อสอบได้ และดึงคลังคำามาแบบสุ่ม เพื่อส่งให้ผู้สมัครงาน โดยอัตโนมัติผ่านทางอีเมล เมื่อผู้สมัครงานทำข้อสอบเสร็จแล้ว ข้อสอบจะถูกส่งไปที่ผู้ออก พร้อมตรวจข้อที่เป็นคำามปรนัยให้ชัดโน้มติ 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 JSON Web Token ภาพรวมการทำงานของระบบจะเป็นดังรูป 3.1



รูปที่ 3.1: แสดงการทำงานของเว็บแอปพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน

3.2 วิเคราะห์ความต้องการ

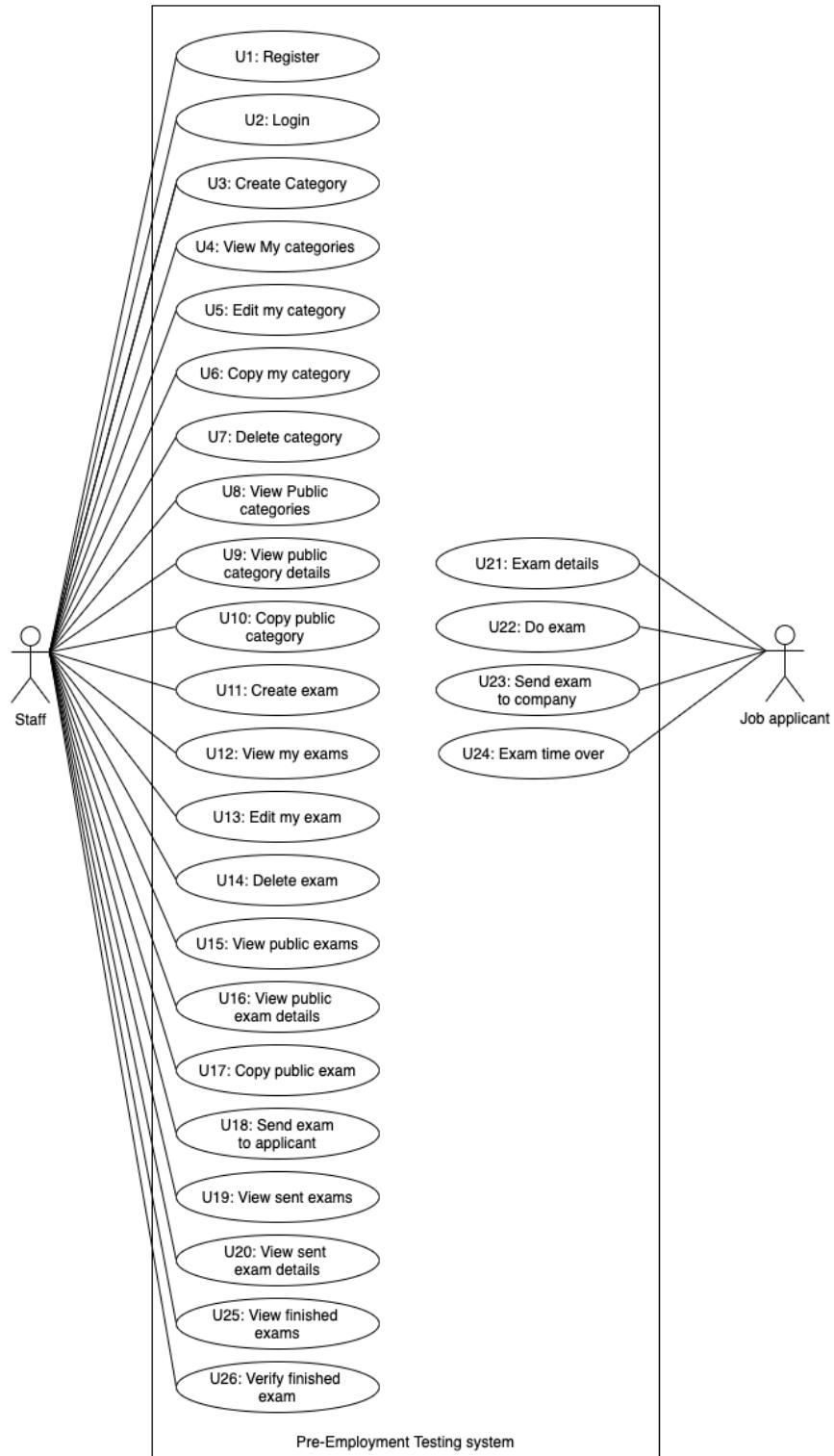
3.2.1 ความต้องการที่เป็นหน้าที่หลักของระบบ (Functional Requirement)

1. ผู้ใช้งานสามารถสามารถลงทะเบียนเป็นสมาชิกเพื่อใช้งานเว็บแอปพลิเคชันได้
2. ผู้ใช้ต้องยืนยันตัวตนผ่านอีเมลหลังจากทำการลงทะเบียน ถึงจะเข้าใช้งานระบบได้
3. ผู้ใช้งานสามารถสร้าง, แก้ไข และลบคำาณในหมวดหมู่คำาณได้
4. ระบบสามารถป้องกันการรีเฟรชเพจ หรือการเปลี่ยนหน้าได้ หากผู้ใช้กำลังสร้างหรือแก้ไขคำาณ ในหมวดหมู่คำาณ
5. ผู้ใช้งานสามารถยกเลิกการสร้างหมวดหมู่คำาณได้
6. ผู้ใช้งานไม่สามารถลบหมวดหมู่คำาณได้ หากหมวดหมู่นั้นถูกใช้งานอยู่ในข้อสอบ
7. ผู้ใช้งานสามารถสร้างคำาณประเภท ปรนัย, อัตโนมัติ และคำาณที่มีไฟล์แนบได้
8. คำาณที่เป็นปรนัย สามารถเลือกให้มีข้อถูกมากกว่า 1 ข้อได้
9. สามารถใส่รูปภาพเพิ่มในตัวเลือกคำาณของหมวดคำาณประเภทปรนัยได้
10. หากผู้ใช้งานส่งข้อสอบให้ผู้สมัคร งานทำแล้ว ถ้าหมวดหมู่คำาณนั้นถูกใช้งานในข้อสอบ คำาณในหมวดหมู่นั้นจะสามารถแก้ไขได้เฉพาะเฉลยเท่านั้น ไม่สามารถแก้ไขคำาณได้ โดยหากต้องการแก้ไขให้ทำการคัดลอกหมวดหมู่คำาณของตนเองมาแก้ไขเป็นหมวดหมู่ใหม่
11. ผู้ใช้งานสามารถคัดลอกหมวดหมู่คำาณของตนเองได้
12. ผู้ใช้งานสามารถเปิดหมวดหมู่คำาณให้เป็นสาธารณะได้
13. หมวดหมู่คำาณที่เป็นสาธารณะ ผู้ที่ไม่ใช้เจ้าของไม่สามารถแก้ไขได้ โดยผู้ใช้สามารถคัดลอก หมวดหมู่คำาณที่เป็นสาธารณะไปเป็นเป็นของตนเองได้ หากผู้ใช้ไม่ใช้เจ้าของแต่ต้องการแก้ไข
14. ผู้ใช้งานสามารถค้นหาหมวดหมู่คำาณได้
15. ผู้ใช้งานสามารถสร้างข้อสอบ ที่กำหนดหมวดหมู่คำาณที่ต้องการ, จำนวนข้อ และความยากได้ โดยข้อสอบจะทำการสุ่มเมื่อผู้ใช้ส่งให้ผู้ทำข้อสอบ
16. ระบบสามารถป้องกันการรีเฟรชเพจ หรือการเปลี่ยนหน้าได้ หากผู้ใช้กำลังสร้างหรือแก้ไขข้อสอบ
17. ผู้ใช้สามารถลบแก้ไขหมวดหมู่คำาณที่ใช้, จำนวนข้อ และระดับความยากของข้อสอบได้
18. ผู้ใช้สามารถค้นหาข้อสอบได้
19. ผู้ใช้สามารถเปิดข้อสอบให้เป็นสาธารณะได้

20. ข้อสอบที่เป็นสาระณะ ผู้ที่ไม่ใช้เจ้าของไม่สามารถแก้ไขได้ โดยผู้ใช้สามารถคัดลอกข้อสอบสาระณะไปเป็นเป็นของตนเองได้ หากผู้ใช้ไม่ใช้เจ้าของแต่ต้องการแก้ไข
21. หากผู้ใช้งานส่งข้อสอบให้ผู้สมัครงานทำแล้ว ข้อสอบที่ผู้ใช้งานสร้างขึ้นนั้นจะไม่สามารถแก้ไขได้ แต่สามารถส่งให้ผู้สมัครงานคนอื่นทำต่อไปได้ โดยระบบจะทำการสุ่มคำตามที่ผู้ออกข้อสอบกำหนด จำนวนข้อประเภทคำตาม และระดับความยาก
22. ผู้ใช้งานสามารถดูประวัติที่ผู้ใช้ทำการส่งข้อสอบให้ผู้สมัครงานได้
23. ผู้ใช้สามารถกำหนดระยะเวลาการทำข้อสอบได้
24. ผู้ใช้สามารถกำหนดวันหมดอายุของข้อสอบได้
25. ผู้ใช้สามารถส่งข้อสอบให้ผู้สมัครงานผ่านอีเมลได้
26. ผู้ทำข้อสอบสามารถเข้ามาทำข้อสอบในเว็บแอพพลิเคชันได้ตลอด หากเวลาในการทำข้อสอบยังไม่หมด
27. เมื่อผู้ทำข้อสอบทำการส่งข้อสอบแล้ว ระบบสามารถตรวจข้อสอบที่เป็นปัจจัยได้
28. มีการแสดงสถานะบอกว่าผู้ใช้งาน ตรวจข้อสอบนั้นหรือยัง
29. มีการสรุปคะแนนรวมของผู้สมัครงาน

3.3 การวิเคราะห์และออกแบบระบบ

3.3.1 แผนภาพพยัญชนะ (Use Case Diagram)



รูปที่ 3.2: แผนภาพพยัญชนะของระบบ

3.3.2 รายละเอียดการทำงานในแต่ละยูสเคส (Use Case Description)

3.3.2.1 รายละเอียดยูสเคส ลงทะเบียน

Use Case No:	1									
Use Case Name:	ลงทะเบียน									
Use Case Scenario:	ผู้ใช้งานลงทะเบียนกับระบบ									
Triggering Event:	เมื่อผู้ใช้เข้าใช้งานเป็นครั้งแรก									
Brief Description:	สำหรับให้ผู้ใช้งานเข้าสู่ระบบ									
Actors:	พนักงาน									
Related Use Cases:	-									
Stakeholders:	-									
Pre - Conditions:	-									
Post - Conditions:	ผู้ใช้สามารถเข้าใช้งานระบบได้									
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1. ผู้ใช้เลือกลงทะเบียน</td> <td>2. ระบบแสดงแบบฟอร์มให้กรอกข้อมูล</td> </tr> <tr> <td>3. ผู้ใช้ทำการกรอกชื่อ-นามสกุล, อีเมล, รหัสผ่าน และยืนยันรหัสผ่าน แล้วทำการคลิกลงทะเบียน</td> <td>4. ระบบทำการส่งอีเมลให้ผู้ใช้เพื่อทำการยืนยันอีเมล</td> </tr> <tr> <td>5. ผู้ใช้งานได้รับอีเมล แล้วทำการคลิกลิงค์เว็บไซต์</td> <td>6. ระบบทำการยืนยันอีเมล แล้วแสดงหน้าจอหลักของผู้ใช้งาน</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. ผู้ใช้เลือกลงทะเบียน	2. ระบบแสดงแบบฟอร์มให้กรอกข้อมูล	3. ผู้ใช้ทำการกรอกชื่อ-นามสกุล, อีเมล, รหัสผ่าน และยืนยันรหัสผ่าน แล้วทำการคลิกลงทะเบียน	4. ระบบทำการส่งอีเมลให้ผู้ใช้เพื่อทำการยืนยันอีเมล	5. ผู้ใช้งานได้รับอีเมล แล้วทำการคลิกลิงค์เว็บไซต์	6. ระบบทำการยืนยันอีเมล แล้วแสดงหน้าจอหลักของผู้ใช้งาน	
ผู้ใช้	ระบบ									
1. ผู้ใช้เลือกลงทะเบียน	2. ระบบแสดงแบบฟอร์มให้กรอกข้อมูล									
3. ผู้ใช้ทำการกรอกชื่อ-นามสกุล, อีเมล, รหัสผ่าน และยืนยันรหัสผ่าน แล้วทำการคลิกลงทะเบียน	4. ระบบทำการส่งอีเมลให้ผู้ใช้เพื่อทำการยืนยันอีเมล									
5. ผู้ใช้งานได้รับอีเมล แล้วทำการคลิกลิงค์เว็บไซต์	6. ระบบทำการยืนยันอีเมล แล้วแสดงหน้าจอหลักของผู้ใช้งาน									
Exception Conditions:	<ul style="list-style-type: none"> - ผู้ใช้กรอกรายละเอียดไม่ครบถ้วน หรือไม่ตรงเจื่อนใจ - ผู้ใช้ไม่ได้ทำการยืนยันอีเมล - ผู้ใช้กรอกอีเมลที่ไม่มีอยู่จริง - ผู้ใช้กรอกอีเมลที่ผู้ใช้งานไม่สามารถเข้าใช้อีเมลนั้นได้ 									

ตารางที่ 3.1: รายละเอียดยูสเคส ลงทะเบียน

3.3.2.2 รายละเอียดьюสเคส เข้าสู่ระบบ

Use Case No:	2							
Use Case Name:	เข้าสู่ระบบ							
Use Case Scenario:	ผู้ใช้ต้องการเข้าสู่ระบบ							
Triggering Event:	<ul style="list-style-type: none"> - เมื่อผู้ใช้ต้องการเข้าใช้งานระบบ - ผู้ใช้เข้าเว็บแอปพลิเคชันโดยที่ยังไม่เข้าใช้งานระบบ 							
Brief Description:	สำหรับให้ผู้ใช้เข้าสู่ระบบเพื่อเข้าใช้งานเว็บแอปพลิเคชัน							
Actors:	พนักงาน							
Related Use Cases:	ลงทะเบียน							
Stakeholders:	-							
Pre - Conditions:	ผู้ใช้ต้องทำการลงทะเบียนและยืนยันตัวตนผ่านอีเมลเรียบร้อยแล้ว							
Post - Conditions:	ระบบแสดงหน้าจอสำหรับผู้ใช้งาน							
Flow of Events:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">ผู้ใช้</th> <th style="text-align: center;">ระบบ</th> </tr> </thead> <tbody> <tr> <td>1. ผู้ใช้เข้าหน้าเว็บโดยที่ยังไม่ได้ทำการเข้าสู่ระบบ</td> <td>2. ระบบแสดงแบบฟอร์มให้กรอกข้อมูล</td> </tr> <tr> <td>3. ผู้ใช้ทำการกรอกอีเมล และรหัสผ่านแล้วทำการคลิกเข้าสู่ระบบ หรือกด Enter บนแป้นพิมพ์</td> <td>4. ระบบ แสดง หน้า จอ หลัก ของผู้ใช้งาน</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. ผู้ใช้เข้าหน้าเว็บโดยที่ยังไม่ได้ทำการเข้าสู่ระบบ	2. ระบบแสดงแบบฟอร์มให้กรอกข้อมูล	3. ผู้ใช้ทำการกรอกอีเมล และรหัสผ่านแล้วทำการคลิกเข้าสู่ระบบ หรือกด Enter บนแป้นพิมพ์	4. ระบบ แสดง หน้า จอ หลัก ของผู้ใช้งาน	
ผู้ใช้	ระบบ							
1. ผู้ใช้เข้าหน้าเว็บโดยที่ยังไม่ได้ทำการเข้าสู่ระบบ	2. ระบบแสดงแบบฟอร์มให้กรอกข้อมูล							
3. ผู้ใช้ทำการกรอกอีเมล และรหัสผ่านแล้วทำการคลิกเข้าสู่ระบบ หรือกด Enter บนแป้นพิมพ์	4. ระบบ แสดง หน้า จอ หลัก ของผู้ใช้งาน							
Exception Conditions:	ผู้ใช้กรอกอีเมล หรือรหัสผ่านผิด							

ตารางที่ 3.2: รายละเอียดьюสเคส เข้าสู่ระบบ

3.3.2.3 รายละเอียดยูสเคส สร้างหมวดหมู่คำถาน

Use Case No:	3											
Use Case Name:	สร้างหมวดหมู่คำถาน											
Use Case Scenario:	ผู้ใช้ต้องการสร้างหมวดหมู่คำถาน											
Triggering Event:	เมื่อผู้ใช้ต้องการสร้างหมวดหมู่คำถาน ไว้ใช้ในข้อสอบของตนเอง หรือให้พนักงานคนอื่นใช้											
Brief Description:	สำหรับให้ผู้ใช้ ออกข้อสอบโดยแบ่งแยกตามแต่ละหมวดหมู่ เช่น คณิตศาสตร์, โปรแกรมมิ่ง หรือ แบบทดสอบสติปัญญา โดยมีชนิดคำถานแบ่งออกเป็น อัตนัย, ปรนัย และคำถานที่มีไฟล์แนบ											
Actors:	พนักงาน											
Related Use Cases:	เข้าสู่ระบบ											
Stakeholders:	-											
Pre - Conditions:	ผู้ใช้ต้องทำการเข้าสู่ระบบ											
Post - Conditions:	ผู้ใช้มีหมวดหมู่คำถานที่ตนเองสร้างขึ้น และแสดงหน้าหมวดหมู่ทั้งหมดของผู้ใช้											
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1. ไปที่หน้าสร้างหมวดหมู่</td> <td>2. ระบบแสดงหน้าสร้างหมวดหมู่คำถาน</td> </tr> <tr> <td>3. ผู้ใช้กรอกชื่อหมวดหมู่ และกรอกหัวเรื่อง</td> <td></td> </tr> <tr> <td>4. สร้างคำถานภาษาไทย หมวดหมู่นั้นโดยสามารถเลือกระดับความยาก และประเภทของคำถาน แล้วทำการคลิกสร้างหมวดหมู่</td> <td></td> </tr> <tr> <td></td> <td>5. ระบบทำการสร้างหมวดหมู่คำถาน และแสดงหน้าหมวดหมู่คำถานทั้งหมดของผู้ใช้</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. ไปที่หน้าสร้างหมวดหมู่	2. ระบบแสดงหน้าสร้างหมวดหมู่คำถาน	3. ผู้ใช้กรอกชื่อหมวดหมู่ และกรอกหัวเรื่อง		4. สร้างคำถานภาษาไทย หมวดหมู่นั้นโดยสามารถเลือกระดับความยาก และประเภทของคำถาน แล้วทำการคลิกสร้างหมวดหมู่			5. ระบบทำการสร้างหมวดหมู่คำถาน และแสดงหน้าหมวดหมู่คำถานทั้งหมดของผู้ใช้	
ผู้ใช้	ระบบ											
1. ไปที่หน้าสร้างหมวดหมู่	2. ระบบแสดงหน้าสร้างหมวดหมู่คำถาน											
3. ผู้ใช้กรอกชื่อหมวดหมู่ และกรอกหัวเรื่อง												
4. สร้างคำถานภาษาไทย หมวดหมู่นั้นโดยสามารถเลือกระดับความยาก และประเภทของคำถาน แล้วทำการคลิกสร้างหมวดหมู่												
	5. ระบบทำการสร้างหมวดหมู่คำถาน และแสดงหน้าหมวดหมู่คำถานทั้งหมดของผู้ใช้											
Exception Conditions:	<ul style="list-style-type: none"> - ผู้ใช้กรอกชื่อหมวดหมู่ซ้ำกับหมวดหมู่ที่ตนเองมีอยู่แล้ว - ผู้ใช้กรอกชื่อ nulla ไม่ครบ ตามฟอร์มที่ให้กรอก - ไฟล์ที่แนบนามีขนาดใหญ่เกินที่กำหนด หรือประเภทของไฟล์ไม่ถูกต้อง 											

ตารางที่ 3.3: รายละเอียดยูสเคส สร้างหมวดหมู่คำถาน

3.3.2.4 รายละเอียดยูสเคส คุณวัดหมู่คำาณของผู้ใช้

Use Case No:	4					
Use Case Name:	คุณวัดหมู่คำาณของผู้ใช้					
Use Case Scenario:	ผู้ใช้ต้องการคุณวัดหมู่คำาณของตนเอง					
Triggering Event:	<ul style="list-style-type: none"> - เมื่อผู้ใช้ต้องการคุณวัดหมู่คำาณของตนเอง - เมื่อผู้ใช้สร้างประเภทของข้อสอบสำเร็จ หน้าเว็บจะทำการไปแสดงผลที่หน้าคุณวัดหมู่คำาณทั้งหมดของผู้ใช้ 					
Brief Description:	สำหรับให้ผู้ใช้งานคุณวัดหมู่คำาณที่ตนเองสร้างขึ้น หรือ หมู่คำาณที่คัดลอกมาจากหมู่คำาณสาธารณะ					
Actors:	พนักงาน					
Related Use Cases:	เข้าสู่ระบบ, สร้างหมู่คำาณ, คัดลอกหมู่คำาณ จากหมู่คำาณสาธารณะ					
Stakeholders:	-					
Pre - Conditions:	<ul style="list-style-type: none"> - ผู้ใช้ต้องทำการเข้าสู่ระบบ - สร้างหมู่คำาณ หรือคัดลอกหมู่คำาณจาก หมู่คำาณสาธารณะ 					
Post - Conditions:	แสดงหน้าคุณวัดหมู่ทั้งหมดของผู้ใช้					
Flow of Events:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">ผู้ใช้</th> <th style="text-align: center;">ระบบ</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1. ไปที่หน้าคุณวัดหมู่คำาณของผู้ใช้</td> <td style="text-align: center;">2. ระบบแสดงหมู่ทั้งหมดของผู้ใช้</td> </tr> </tbody> </table>		ผู้ใช้	ระบบ	1. ไปที่หน้าคุณวัดหมู่คำาณของผู้ใช้	2. ระบบแสดงหมู่ทั้งหมดของผู้ใช้
ผู้ใช้	ระบบ					
1. ไปที่หน้าคุณวัดหมู่คำาณของผู้ใช้	2. ระบบแสดงหมู่ทั้งหมดของผู้ใช้					
Exception Conditions:	ผู้ใช้ไม่มีหมู่คำาณ					

ตารางที่ 3.4: รายละเอียดยูสเคส คุณวัดหมู่คำาณของผู้ใช้

3.3.2.5 รายละเอียดยูสเคส แก้ไขหมวดหมู่คำาณของผู้ใช้

Use Case No:	5											
Use Case Name:	แก้ไขหมวดหมู่คำาณของผู้ใช้											
Use Case Scenario:	ผู้ใช้ต้องการแก้ไขหมวดหมู่คำาณของตนเอง											
Triggering Event:	เมื่อผู้ใช้ต้องการแก้ไขรายละเอียดในหมวดหมู่คำาณ เช่นเพิ่มคำาณ, แก้ไขคำาณ หรือแก้ไขชื่อหมวดหมู่											
Brief Description:	สำหรับให้ผู้ใช้งานแก้ไขรายละเอียดต่างๆในหมวดหมู่คำาณ											
Actors:	พนักงาน											
Related Use Cases:	เข้าสู่ระบบ, สร้างหมวดหมู่คำาณ, คัดลอกหมวดหมู่คำาณ จากหมวดหมู่คำาณสาธารณะ											
Stakeholders:	-											
Pre - Conditions:	<ul style="list-style-type: none"> - ผู้ใช้ต้องทำการเข้าสู่ระบบ - สร้างหมวดหมู่คำาณ หรือคัดลอกหมวดหมู่คำาณจากหมวดหมู่คำาณสาธารณะ 											
Post - Conditions:	แสดงหน้าหมวดหมู่ทั้งหมดของผู้ใช้											
Flow of Events:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">ผู้ใช้</th> <th style="text-align: center;">ระบบ</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1.ไปที่หน้าคุ้มหมวดหมู่คำาณของผู้ใช้</td> <td style="text-align: center;">2.ระบบแสดงหมวดหมู่ทั้งหมดของผู้ใช้</td> </tr> <tr> <td style="text-align: center;">3.คลิกที่หมวดหมู่คำาณที่ต้องการแก้ไข</td> <td style="text-align: center;">4.ระบบแสดงหน้าแก้ไขหมวดหมู่นั้นๆ</td> </tr> <tr> <td style="text-align: center;">5.ผู้ใช้ทำการแก้ไขข้อมูลต่างๆในประเภทที่ต้องการแล้วทำการคลิกยืนยันการแก้ไข</td> <td style="text-align: center;">6.ระบบทำการแก้ไขข้อมูล</td> </tr> <tr> <td></td> <td style="text-align: center;">7.ระบบแสดงหน้าหมวดหมู่ทั้งหมดของผู้ใช้</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1.ไปที่หน้าคุ้มหมวดหมู่คำาณของผู้ใช้	2.ระบบแสดงหมวดหมู่ทั้งหมดของผู้ใช้	3.คลิกที่หมวดหมู่คำาณที่ต้องการแก้ไข	4.ระบบแสดงหน้าแก้ไขหมวดหมู่นั้นๆ	5.ผู้ใช้ทำการแก้ไขข้อมูลต่างๆในประเภทที่ต้องการแล้วทำการคลิกยืนยันการแก้ไข	6.ระบบทำการแก้ไขข้อมูล		7.ระบบแสดงหน้าหมวดหมู่ทั้งหมดของผู้ใช้	
ผู้ใช้	ระบบ											
1.ไปที่หน้าคุ้มหมวดหมู่คำาณของผู้ใช้	2.ระบบแสดงหมวดหมู่ทั้งหมดของผู้ใช้											
3.คลิกที่หมวดหมู่คำาณที่ต้องการแก้ไข	4.ระบบแสดงหน้าแก้ไขหมวดหมู่นั้นๆ											
5.ผู้ใช้ทำการแก้ไขข้อมูลต่างๆในประเภทที่ต้องการแล้วทำการคลิกยืนยันการแก้ไข	6.ระบบทำการแก้ไขข้อมูล											
	7.ระบบแสดงหน้าหมวดหมู่ทั้งหมดของผู้ใช้											
Exception Conditions:	<ul style="list-style-type: none"> - ผู้ใช้กรอกชื่อหมวดหมู่ซ้ำกับหมวดหมู่ที่ตนเองมีอยู่แล้ว - ผู้ใช้กรอกรายละเอียดไม่ครบถ้วน หรือไม่ตรงเจื่อนใจ - ไฟล์ที่แนบมาไม่ขนาดใหญ่เกินที่กำหนด หรือประเภทของไฟล์ไม่ถูกต้อง - สามารถแก้ไขได้เฉพาะคำาณ หากข้อสอบเคยถูกใช้ไปแล้ว 											

ตารางที่ 3.5: รายละเอียดยูสเคส แก้ไขหมวดหมู่คำาณของผู้ใช้

3.3.2.6 รายละเอียดยูสเคส คัดลอกหมวดหมู่คำาณของผู้ใช้

Use Case No:	6									
Use Case Name:	คัดลอกหมวดหมู่คำาณของผู้ใช้									
Use Case Scenario:	ผู้ใช้ต้องการคัดลอกหมวดหมู่คำาณของตนเอง									
Triggering Event:	เมื่อผู้ใช้ต้องการแก้ไขหมวดหมู่คำาณที่ถูกส่งให้ผู้สมัครงาน แล้ว ทำให้มีความสามารถแก้ไขหมวดหมู่เดิมได้ หรือต้องการคัดลอก หมวดหมู่มาแก้ไข โดยที่ไม่ต้องการให้ระบบหมวดหมู่เดิม									
Brief Description:	สำหรับให้ผู้ใช้งานคัดลอกหมวดหมู่คำาณของตนเอง									
Actors:	พนักงาน									
Related Use Cases:	เข้าสู่ระบบ, สร้างหมวดหมู่คำาณ, ส่งข้อสอบให้ผู้สมัคร									
Stakeholders:	-									
Pre - Conditions:	<ul style="list-style-type: none"> - ผู้ใช้ต้องทำการเข้าสู่ระบบ - สร้างหมวดหมู่คำาณ หรือคัดลอกหมวดหมู่คำาณจาก หมวดหมู่คำาณสาธารณะ 									
Post - Conditions:	ระบบทำการคัดลอกหมวดหมู่คำาณที่ผู้ใช้ต้องการ									
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1. ไปที่หน้าจอหมวดหมู่ คำาณของผู้ใช้</td> <td>2. ระบบแสดงหมวดหมู่ ทั้งหมดของผู้ใช้</td> </tr> <tr> <td>3. คลิกปุ่มคัดลอกที่หมวด หมู่ที่ต้องการคัด ลอก</td> <td>4. ระบบแสดงหน้ายืนยันการ คัดลอก</td> </tr> <tr> <td>5. ผู้ใช้งานกดยืนยันการคัด ลอก</td> <td>6. ระบบทำการคัดลอกหมวด หมู่คำาณนั้น</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. ไปที่หน้าจอหมวดหมู่ คำาณของผู้ใช้	2. ระบบแสดงหมวดหมู่ ทั้งหมดของผู้ใช้	3. คลิกปุ่มคัดลอกที่หมวด หมู่ที่ต้องการคัด ลอก	4. ระบบแสดงหน้ายืนยันการ คัดลอก	5. ผู้ใช้งานกดยืนยันการคัด ลอก	6. ระบบทำการคัดลอกหมวด หมู่คำาณนั้น	
ผู้ใช้	ระบบ									
1. ไปที่หน้าจอหมวดหมู่ คำาณของผู้ใช้	2. ระบบแสดงหมวดหมู่ ทั้งหมดของผู้ใช้									
3. คลิกปุ่มคัดลอกที่หมวด หมู่ที่ต้องการคัด ลอก	4. ระบบแสดงหน้ายืนยันการ คัดลอก									
5. ผู้ใช้งานกดยืนยันการคัด ลอก	6. ระบบทำการคัดลอกหมวด หมู่คำาณนั้น									
Exception Conditions:	ผู้ใช้ไม่มีหมวดหมู่คำาณ									

ตารางที่ 3.6: รายละเอียดยูสเคส คัดลอกหมวดหมู่คำาณของผู้ใช้

3.3.2.7 รายละเอียดยูสเคส ลบหมวดหมู่คำาณของผู้ใช้

Use Case No:	7									
Use Case Name:	ลบหมวดหมู่คำาณของผู้ใช้									
Use Case Scenario:	ผู้ใช้ต้องการลบหมวดหมู่คำาณของตนเอง									
Triggering Event:	เมื่อผู้ใช้ไม่ต้องการหมวดหมู่คำาณนั้นๆแล้ว									
Brief Description:	สำหรับให้ผู้ใช้งานลบหมวดหมู่คำาณออกจากหมวดหมู่คำาณของตนเอง									
Actors:	พนักงาน									
Related Use Cases:	เข้าสู่ระบบ, สร้างหมวดหมู่คำาณ, คัดลอกหมวดหมู่คำาณจากหมวดหมู่คำาณสาธารณะ, ข้อสอบของผู้ใช้									
Stakeholders:	-									
Pre - Conditions:	<ul style="list-style-type: none"> - ผู้ใช้ต้องทำการเข้าสู่ระบบ - สร้างหมวดหมู่คำาณ หรือคัดลอกหมวดหมู่คำาณจากหมวดหมู่สาธารณะ 									
Post - Conditions:	ระบบลบหมวดหมู่คำาณของผู้ใช้									
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1. ไปที่หน้าจอหมวดหมู่คำาณของผู้ใช้</td> <td>2. ระบบแสดงหมวดหมู่ทั้งหมดของผู้ใช้</td> </tr> <tr> <td>3. คลิกที่รูปถึงขยะของหมวดหมู่คำาณที่ต้องการลบ</td> <td>4. ระบบทำการแสดงหน้าต่างข้อความยืนยันการลบ</td> </tr> <tr> <td>5. ผู้ใช้ทำการยืนยันการลบ</td> <td>6. ระบบทำการลบหมวดหมู่คำาณที่ผู้ใช้ต้องการลบ</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. ไปที่หน้าจอหมวดหมู่คำาณของผู้ใช้	2. ระบบแสดงหมวดหมู่ทั้งหมดของผู้ใช้	3. คลิกที่รูปถึงขยะของหมวดหมู่คำาณที่ต้องการลบ	4. ระบบทำการแสดงหน้าต่างข้อความยืนยันการลบ	5. ผู้ใช้ทำการยืนยันการลบ	6. ระบบทำการลบหมวดหมู่คำาณที่ผู้ใช้ต้องการลบ	
ผู้ใช้	ระบบ									
1. ไปที่หน้าจอหมวดหมู่คำาณของผู้ใช้	2. ระบบแสดงหมวดหมู่ทั้งหมดของผู้ใช้									
3. คลิกที่รูปถึงขยะของหมวดหมู่คำาณที่ต้องการลบ	4. ระบบทำการแสดงหน้าต่างข้อความยืนยันการลบ									
5. ผู้ใช้ทำการยืนยันการลบ	6. ระบบทำการลบหมวดหมู่คำาณที่ผู้ใช้ต้องการลบ									
Exception Conditions:	<ul style="list-style-type: none"> - ผู้ใช้ไม่ทำการยืนยันการลบ - หมวดหมู่คำาณที่ต้องการลบ ถูกใช้งานอยู่ในข้อสอบ ทำให้ไม่สามารถลบได้ 									

ตารางที่ 3.7: รายละเอียดยูสเคส ลบหมวดหมู่คำาณของผู้ใช้

3.3.2.8 รายละเอียดยูสเคส คูหมวดหมู่คำาณสารณะ

Use Case No:	8					
Use Case Name:	คูหมวดหมู่คำาณสารณะ					
Use Case Scenario:	ผู้ใช้ต้องการคูหมวดหมู่คำาณสารณะ					
Triggering Event:	<ul style="list-style-type: none"> - เมื่อผู้ใช้ต้องการคูหมวดหมู่คำาณสารณะ - เมื่อผู้ใช้ต้องการนำหมวดหมู่คำาณสารณะไปใช้ในข้อสอบของตนเอง 					
Brief Description:	สำหรับให้ผู้ใช้คูว่ามีหมวดหมู่คำาณสารณะใดบ้างที่อยู่ในระบบและสามารถกดลอกไปใช้ได้					
Actors:	พนักงาน					
Related Use Cases:	เข้าสู่ระบบ					
Stakeholders:	-					
Pre - Conditions:	ผู้ใช้ต้องทำการเข้าสู่ระบบ					
Post - Conditions:	ระบบแสดงหมวดหมุข้อสอบทั้งหมด ที่เป็นสารณะ					
Flow of Events:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">ผู้ใช้</th> <th style="text-align: center;">ระบบ</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1.ไปที่หน้าคูหมวดหมู่ คำาณสารณะ</td> <td style="text-align: center;">2.ระบบแสดงหมวดหมุ ทั้งหมดที่เป็นสารณะ</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1.ไปที่หน้าคูหมวดหมู่ คำาณสารณะ	2.ระบบแสดงหมวดหมุ ทั้งหมดที่เป็นสารณะ	
ผู้ใช้	ระบบ					
1.ไปที่หน้าคูหมวดหมู่ คำาณสารณะ	2.ระบบแสดงหมวดหมุ ทั้งหมดที่เป็นสารณะ					
Exception Conditions:	ไม่มีหมวดหมู่คำาณสารณะที่เป็นสารณะอยู่ในระบบ					

ตารางที่ 3.8: รายละเอียดยูสเคส คูหมวดหมู่คำาณสารณะ

3.3.2.9 รายละเอียดьюสเคส คุร้ายละเอียดของหมวดหมู่คำาณสารณะ

Use Case No:	9							
Use Case Name:	คุร้ายละเอียดของหมวดหมู่คำาณสารณะ							
Use Case Scenario:	ผู้ใช้ต้องการคุร้ายละเอียดในหมวดหมู่คำาณสารณะที่ต้องการ							
Triggering Event:	<ul style="list-style-type: none"> - เมื่อผู้ใช้ต้องการคุร้ายละเอียดของหมวดหมู่คำาณสารณะ - เมื่อผู้ใช้ต้องการนำหมวดหมู่คำาณสารณะนั้นๆไปใช้ในข้อสอบของตนเอง 							
Brief Description:	สำหรับให้ผู้ใช้คุร้ายละเอียดของหมวดหมู่คำาณสารณะ ว่าในหมวดหมู่นั้นมีคำาณอะไรบ้าง							
Actors:	พนักงาน							
Related Use Cases:	เข้าสู่ระบบ, ดูหมวดหมู่คำาณสารณะ							
Stakeholders:	-							
Pre - Conditions:	ผู้ใช้ต้องทำการเข้าสู่ระบบ							
Post - Conditions:	ระบบรายละเอียดของหมวดหมู่ข้อสอบนั้นๆ							
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1. ไปที่หน้าคุหมวดหมู่ คำาณสารณะ</td> <td>2. ระบบแสดงหมวดหมู่ ทั้งหมดที่เป็นสารณะ</td> </tr> <tr> <td>3. คลิกหมวดหมู่ที่ต้องการ คุร้ายละเอียด</td> <td>4. ระบบแสดงรายละเอียด ของหมวดหมู่คำาณนั้น</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. ไปที่หน้าคุหมวดหมู่ คำาณสารณะ	2. ระบบแสดงหมวดหมู่ ทั้งหมดที่เป็นสารณะ	3. คลิกหมวดหมู่ที่ต้องการ คุร้ายละเอียด	4. ระบบแสดงรายละเอียด ของหมวดหมู่คำาณนั้น	
ผู้ใช้	ระบบ							
1. ไปที่หน้าคุหมวดหมู่ คำาณสารณะ	2. ระบบแสดงหมวดหมู่ ทั้งหมดที่เป็นสารณะ							
3. คลิกหมวดหมู่ที่ต้องการ คุร้ายละเอียด	4. ระบบแสดงรายละเอียด ของหมวดหมู่คำาณนั้น							
Exception Conditions:	ไม่มีหมวดหมู่คำาณที่เป็นสารณะอยู่ในระบบ							

ตารางที่ 3.9: รายละเอียดьюสเคส คุร้ายละเอียดของหมวดหมู่คำาณสารณะ

3.3.2.10 รายละเอียดมูสกेट คัดลอกหมวดหมู่คำาณสารณะ

Use Case No:	10							
Use Case Name:	คัดลอกหมวดหมู่คำาณสารณะ							
Use Case Scenario:	ผู้ใช้ต้องการนำหมวดหมู่คำาณสารณะ							
Triggering Event:	เมื่อผู้ใช้ต้องการนำหมวดหมู่คำาณสารณะนั้นๆ ไปใช้ในข้อสอบของตนเอง							
Brief Description:	สำหรับให้ผู้ใช้ทำการคัดลอกหมวดหมู่คำาณสารณะ ไปใช้ในข้อสอบของตนเอง							
Actors:	พนักงาน							
Related Use Cases:	เข้าสู่ระบบ, ดูหมวดหมู่คำาณหมวดหมู่คำาณสารณะ							
Stakeholders:	-							
Pre - Conditions:	ผู้ใช้ต้องทำการเข้าสู่ระบบ							
Post - Conditions:	ระบบทำการคัดลอกหมวดหมู่สารณะที่ผู้ใช้ต้องการ ไปเก็บในหมวดหมู่คำาณของตนเอง							
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1.ไปที่หน้าคุ้มหมวดหมู่ คำาณสารณะ</td> <td>2.ระบบแสดงหมวดหมู่ ทั้งหมดที่เป็นสารณะ</td> </tr> <tr> <td>3.คลิกปุ่มคัดลอกที่หมวด หมู่ที่ต้องการ</td> <td>4.ระบบทำการคัดลอกข้อมูล สารณะที่ต้องการ ไปเก็บ ในหมวดหมู่คำาณของตน เอง</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1.ไปที่หน้าคุ้มหมวดหมู่ คำาณสารณะ	2.ระบบแสดงหมวดหมู่ ทั้งหมดที่เป็นสารณะ	3.คลิกปุ่มคัดลอกที่หมวด หมู่ที่ต้องการ	4.ระบบทำการคัดลอกข้อมูล สารณะที่ต้องการ ไปเก็บ ในหมวดหมู่คำาณของตน เอง	
ผู้ใช้	ระบบ							
1.ไปที่หน้าคุ้มหมวดหมู่ คำาณสารณะ	2.ระบบแสดงหมวดหมู่ ทั้งหมดที่เป็นสารณะ							
3.คลิกปุ่มคัดลอกที่หมวด หมู่ที่ต้องการ	4.ระบบทำการคัดลอกข้อมูล สารณะที่ต้องการ ไปเก็บ ในหมวดหมู่คำาณของตน เอง							
Exception Conditions:	ไม่มีหมวดหมู่คำาณที่เป็นสารณะอยู่ในระบบ							

ตารางที่ 3.10: รายละเอียดมูสกेट คัดลอกหมวดหมู่คำาณสารณะ

3.3.2.11 รายละเอียดยูสเคส สร้างข้อสอบ

Use Case No:	11							
Use Case Name:	สร้างข้อสอบ							
Use Case Scenario:	ผู้ใช้ต้องการสร้างข้อสอบ							
Triggering Event:	เมื่อผู้ใช้ต้องการสร้างข้อสอบไปของตนเอง เพื่อนำไปทดสอบกับผู้สมัครงาน							
Brief Description:	สำหรับผู้ใช้ที่ต้องการทำการทำออกข้อสอบ โดยการคึ่งคำตามจากประเภทของข้อสอบ ตามจำนวนข้อและความยากที่ต้องการ เช่น ต้องการคำตามหมวดหมู่ คณิตศาสตร์ ข้อระดับง่าย 2 ข้อ ข้อระดับยาก 1 ข้อ และคำตามหมวดหมู่ ภาษาอังกฤษ ข้อระดับปานกลาง 3 ข้อ สามารถกำหนดระยะเวลาในการทำข้อสอบ และวันหมดอายุของข้อสอบได้							
Actors:	พนักงาน							
Related Use Cases:	เข้าสู่ระบบ, สร้างหมวดหมู่คำตาม, กัดลอกหมวดหมู่คำตาม สาระณะ							
Stakeholders:	-							
Pre - Conditions:	<ul style="list-style-type: none"> - ผู้ใช้ต้องทำการเข้าสู่ระบบ - ผู้ใช้งานต้องสร้างหมวดหมู่คำตามของตนเอง หรือกัดลอกหมวดหมู่คำตามสาระณะ 							
Post - Conditions:	ระบบทำการสร้างข้อสอบตามที่ผู้ใช้กำหนด							
Flow of Events:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">ผู้ใช้</th> <th style="text-align: center;">ระบบ</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1.ไปที่หน้าสร้างข้อสอบ</td> <td style="text-align: center;">2.ระบบแสดงหมวดหมู่ของตนเองสำหรับเลือกใช้ในข้อสอบ และแสดงรายละเอียดให้กรอก</td> </tr> <tr> <td style="text-align: center;">3.เลือกหมวดหมู่คำตามที่ต้องการ และเลือกระดับความยากและจำนวนข้อที่ต้องการในหมวดหมู่นั้นๆ และคลิกสร้างข้อสอบ</td> <td style="text-align: center;">4.ระบบทำการสร้างข้อสอบตามที่ผู้ใช้งานกำหนด</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1.ไปที่หน้าสร้างข้อสอบ	2.ระบบแสดงหมวดหมู่ของตนเองสำหรับเลือกใช้ในข้อสอบ และแสดงรายละเอียดให้กรอก	3.เลือกหมวดหมู่คำตามที่ต้องการ และเลือกระดับความยากและจำนวนข้อที่ต้องการในหมวดหมู่นั้นๆ และคลิกสร้างข้อสอบ	4.ระบบทำการสร้างข้อสอบตามที่ผู้ใช้งานกำหนด	
ผู้ใช้	ระบบ							
1.ไปที่หน้าสร้างข้อสอบ	2.ระบบแสดงหมวดหมู่ของตนเองสำหรับเลือกใช้ในข้อสอบ และแสดงรายละเอียดให้กรอก							
3.เลือกหมวดหมู่คำตามที่ต้องการ และเลือกระดับความยากและจำนวนข้อที่ต้องการในหมวดหมู่นั้นๆ และคลิกสร้างข้อสอบ	4.ระบบทำการสร้างข้อสอบตามที่ผู้ใช้งานกำหนด							
Exception Conditions:	<ul style="list-style-type: none"> - ไม่มีหมวดหมู่คำตามของตนเอง - ผู้ใช้กรอกรายละเอียดไม่ครบถ้วน หรือไม่ตรงเจื่อนใจ 							

ตารางที่ 3.11: รายละเอียดยูสเคส สร้างข้อสอบ

3.3.2.12 รายละเอียดัญญาสเคส คุ้มข้อสอบทั้งหมดของตนเอง

Use Case No:	12					
Use Case Name:	คุ้มข้อสอบทั้งหมดของตนเอง					
Use Case Scenario:	ผู้ใช้ต้องการคุ้มข้อสอบทั้งหมดของตนเอง					
Triggering Event:	เมื่อผู้ใช้ต้องการคุ้มว่าตนเองมีข้อสอบอะไรบ้าง					
Brief Description:	สำหรับผู้ใช้ที่คุ้มข้อสอบที่ตนเองสร้างขึ้น หรือคุ้มข้อสอบที่ทำการคัดลอกมาจากข้อสอบสาธารณะ					
Actors:	พนักงาน					
Related Use Cases:	เข้าสู่ระบบ, สร้างข้อสอบ, คัดลอกข้อสอบจากสาธารณะ					
Stakeholders:	-					
Pre - Conditions:	ผู้ใช้ต้องทำการเข้าสู่ระบบ					
Post - Conditions:	ระบบทำการแสดงข้อสอบทั้งหมดของตนเอง					
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1. ไปที่หน้าคุ้มข้อสอบทั้งหมดของตนเอง</td> <td>2. ระบบแสดงข้อสอบทั้งหมดของผู้ใช้</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. ไปที่หน้าคุ้มข้อสอบทั้งหมดของตนเอง	2. ระบบแสดงข้อสอบทั้งหมดของผู้ใช้	
ผู้ใช้	ระบบ					
1. ไปที่หน้าคุ้มข้อสอบทั้งหมดของตนเอง	2. ระบบแสดงข้อสอบทั้งหมดของผู้ใช้					
Exception Conditions:	ไม่มีข้อสอบของตนเอง					

ตารางที่ 3.12: รายละเอียดัญญาสเคส คุ้มข้อสอบทั้งหมดของตนเอง

3.3.2.13 รายละเอียดยูสเคส แก้ไขข้อสอบของตนเอง

Use Case No:	13									
Use Case Name:	แก้ไขข้อสอบของตนเอง									
Use Case Scenario:	ผู้ใช้ต้องการแก้ไขข้อสอบของตนเอง									
Triggering Event:	เมื่อผู้ใช้ต้องการแก้ไขข้อสอบที่ตนเองสร้างขึ้น หรือข้อสอบที่ทำการคัดสอบมาจากข้อสอบสาธารณะ									
Brief Description:	สำหรับผู้ใช้ที่ต้องการแก้ไขข้อสอบ เช่น เพิ่มหรือลดจำนวนข้อและระดับความยากในหมวดหมู่คำถามนั้นๆ									
Actors:	พนักงาน									
Related Use Cases:	เข้าสู่ระบบ, สร้างข้อสอบ, คัดลอกข้อสอบจากสาธารณะ									
Stakeholders:	-									
Pre - Conditions:	<ul style="list-style-type: none"> - ผู้ใช้ต้องทำการเข้าสู่ระบบ - ผู้ใช้ต้องสร้างข้อสอบ หรือคัดลอกข้อสอบจากข้อสอบสาธารณะ 									
Post - Conditions:	ระบบทำการแก้ไขข้อสอบตามที่ผู้ใช้กำหนด									
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1.ไปที่หน้าดูข้อสอบทั้งหมดของตนเอง</td> <td>2.ระบบแสดงข้อสอบทั้งหมดของผู้ใช้</td> </tr> <tr> <td>3.เลือกข้อสอบที่ต้องการแก้ไข</td> <td>4.ระบบทำการแสดงรายละเอียดของข้อสอบนั้น</td> </tr> <tr> <td>5.ทำการแก้ไขข้อสอบตามที่ต้องการ แล้วคลิกยืนยัน</td> <td>6.ระบบทำการแก้ไขตามที่ผู้ใช้กำหนด</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1.ไปที่หน้าดูข้อสอบทั้งหมดของตนเอง	2.ระบบแสดงข้อสอบทั้งหมดของผู้ใช้	3.เลือกข้อสอบที่ต้องการแก้ไข	4.ระบบทำการแสดงรายละเอียดของข้อสอบนั้น	5.ทำการแก้ไขข้อสอบตามที่ต้องการ แล้วคลิกยืนยัน	6.ระบบทำการแก้ไขตามที่ผู้ใช้กำหนด	
ผู้ใช้	ระบบ									
1.ไปที่หน้าดูข้อสอบทั้งหมดของตนเอง	2.ระบบแสดงข้อสอบทั้งหมดของผู้ใช้									
3.เลือกข้อสอบที่ต้องการแก้ไข	4.ระบบทำการแสดงรายละเอียดของข้อสอบนั้น									
5.ทำการแก้ไขข้อสอบตามที่ต้องการ แล้วคลิกยืนยัน	6.ระบบทำการแก้ไขตามที่ผู้ใช้กำหนด									
Exception Conditions:	<ul style="list-style-type: none"> - ไม่มีข้อสอบของตนเอง - ผู้ใช้มีการใช้ชื่อซ้ำกับข้อสอบของตนเองที่มีอยู่ - ผู้ใช้กรอกรายละเอียดไม่ครบถ้วน หรือไม่ตรงเจื่อนใจ 									

ตารางที่ 3.13: รายละเอียดยูสเคส แก้ไขข้อสอบของตนเอง

3.3.2.14 รายละเอียดยูสเคส แก้ไขข้อสอบของตนเอง

Use Case No:	13									
Use Case Name:	แก้ไขข้อสอบของตนเอง									
Use Case Scenario:	ผู้ใช้ต้องการแก้ไขรายละเอียดข้อสอบของตนเอง									
Triggering Event:	เมื่อผู้ใช้ต้องการแก้ไขรายละเอียดข้อสอบที่ตนเองสร้างขึ้น หรือ ข้อสอบที่ทำการคัดสอบมาจากข้อสอบสาธารณะ									
Brief Description:	สำหรับผู้ใช้ที่ต้องการแก้ไขรายละเอียดข้อสอบ เช่น เพิ่มหรือลดจำนวนข้อและระดับความยากในหมวดหมู่คำถามนั้นๆ									
Actors:	พนักงาน									
Related Use Cases:	เข้าสู่ระบบ, สร้างข้อสอบ, คัดลอกข้อสอบจากสาธารณะ									
Stakeholders:	-									
Pre - Conditions:	<ul style="list-style-type: none"> - ผู้ใช้ต้องทำการเข้าสู่ระบบ - ผู้ใช้ต้องสร้างข้อสอบ หรือคัดลอกข้อสอบจากข้อสอบสาธารณะ 									
Post - Conditions:	ระบบทำการแก้ไขข้อสอบตามที่ผู้ใช้กำหนด									
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1. ไปที่หน้าดูข้อสอบทั้งหมดของตนเอง</td> <td>2. ระบบแสดงข้อสอบทั้งหมดของผู้ใช้</td> </tr> <tr> <td>3. เลือกข้อสอบที่ต้องการแก้ไข</td> <td>4. ระบบทำการแสดงรายละเอียดของข้อสอบนั้น</td> </tr> <tr> <td>5. ทำการแก้ไขข้อสอบตามที่ต้องการ แล้วคลิกยืนยัน</td> <td>6. ระบบทำการแก้ไขตามที่ผู้ใช้กำหนด</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. ไปที่หน้าดูข้อสอบทั้งหมดของตนเอง	2. ระบบแสดงข้อสอบทั้งหมดของผู้ใช้	3. เลือกข้อสอบที่ต้องการแก้ไข	4. ระบบทำการแสดงรายละเอียดของข้อสอบนั้น	5. ทำการแก้ไขข้อสอบตามที่ต้องการ แล้วคลิกยืนยัน	6. ระบบทำการแก้ไขตามที่ผู้ใช้กำหนด	
ผู้ใช้	ระบบ									
1. ไปที่หน้าดูข้อสอบทั้งหมดของตนเอง	2. ระบบแสดงข้อสอบทั้งหมดของผู้ใช้									
3. เลือกข้อสอบที่ต้องการแก้ไข	4. ระบบทำการแสดงรายละเอียดของข้อสอบนั้น									
5. ทำการแก้ไขข้อสอบตามที่ต้องการ แล้วคลิกยืนยัน	6. ระบบทำการแก้ไขตามที่ผู้ใช้กำหนด									
Exception Conditions:	<ul style="list-style-type: none"> - ไม่มีข้อสอบของตนเอง - ผู้ใช้มีการใช้ชื่อซ้ำกับข้อสอบของตนเองที่มีอยู่ - ผู้ใช้กรอกรายละเอียดไม่ครบถ้วน หรือไม่ตรงเจื่อนใจ 									

ตารางที่ 3.14: รายละเอียดยูสเคส แก้ไขข้อสอบของตนเอง

3.3.2.15 รายละเอียดมูสคेस ลบข้อสอบ

Use Case No:	14									
Use Case Name:	ลบข้อสอบ									
Use Case Scenario:	ผู้ใช้ต้องการลบข้อสอบ									
Triggering Event:	เมื่อผู้ใช้ไม่ต้องการข้อสอบนั้นแล้ว									
Brief Description:	สำหรับผู้ใช้ที่ต้องลบข้อสอบที่ตนเองสร้างขึ้น หรือลบข้อสอบที่ทำการคัดลอกมาจากข้อสอบสาธารณะ									
Actors:	พนักงาน									
Related Use Cases:	เข้าสู่ระบบ, สร้างข้อสอบ, คัดลอกข้อสอบจากสาธารณะ									
Stakeholders:	-									
Pre - Conditions:	<ul style="list-style-type: none"> - ผู้ใช้ต้องทำการเข้าสู่ระบบ - ผู้ใช้ต้องสร้างข้อสอบ หรือคัดลอกข้อสอบจากข้อสอบสาธารณะ 									
Post - Conditions:	ระบบทำการคัดลอกข้อสอบที่ผู้ใช้ต้องการ									
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1. ไปที่หน้าดูข้อสอบทั้งหมดของตนเอง</td> <td>2. ระบบแสดงข้อสอบทั้งหมดของผู้ใช้</td> </tr> <tr> <td>3. เลือกข้อสอบที่ต้องการลบ</td> <td>4. ระบบทำการแสดงหน้าต่างข้อความยืนยันการลบ</td> </tr> <tr> <td>5. ทำการยืนยันการลบข้อสอบ</td> <td>6. ระบบทำการคัดลอกหมวดหมู่คำถามนั้น</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. ไปที่หน้าดูข้อสอบทั้งหมดของตนเอง	2. ระบบแสดงข้อสอบทั้งหมดของผู้ใช้	3. เลือกข้อสอบที่ต้องการลบ	4. ระบบทำการแสดงหน้าต่างข้อความยืนยันการลบ	5. ทำการยืนยันการลบข้อสอบ	6. ระบบทำการคัดลอกหมวดหมู่คำถามนั้น	
ผู้ใช้	ระบบ									
1. ไปที่หน้าดูข้อสอบทั้งหมดของตนเอง	2. ระบบแสดงข้อสอบทั้งหมดของผู้ใช้									
3. เลือกข้อสอบที่ต้องการลบ	4. ระบบทำการแสดงหน้าต่างข้อความยืนยันการลบ									
5. ทำการยืนยันการลบข้อสอบ	6. ระบบทำการคัดลอกหมวดหมู่คำถามนั้น									
Exception Conditions:	<ul style="list-style-type: none"> - ไม่มีข้อสอบของตนเอง - ผู้ใช้ไม่ทำการยืนยันการลบข้อสอบ 									

ตารางที่ 3.15: รายละเอียดมูสคेस ลบข้อสอบ

3.3.2.16 รายละเอียดยูสเคส คุ้มครองข้อมูลส่วนบุคคล

Use Case No:	15					
Use Case Name:	คุ้มครองข้อมูลส่วนบุคคล					
Use Case Scenario:	ผู้ใช้ต้องการคุ้มครองข้อมูลส่วนบุคคล					
Triggering Event:	เมื่อผู้ใช้มีความต้องการที่จะคุ้มครองข้อมูลส่วนบุคคล					
Brief Description:	สำหรับผู้ใช้ที่ต้องการคุ้มครองข้อมูลส่วนบุคคลที่เป็นส่วนบุคคล					
Actors:	พนักงาน					
Related Use Cases:	เข้าสู่ระบบ					
Stakeholders:	-					
Pre - Conditions:	ผู้ใช้ต้องทำการเข้าสู่ระบบ					
Post - Conditions:	ระบบทำการแสดงข้อมูลที่เป็นส่วนบุคคลทั้งหมด					
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1. ไปที่หน้าคุ้มครอง ส่วนบุคคล</td> <td>2. ระบบแสดงข้อมูลทั้งหมดที่เป็นส่วนบุคคล</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. ไปที่หน้าคุ้มครอง ส่วนบุคคล	2. ระบบแสดงข้อมูลทั้งหมดที่เป็นส่วนบุคคล	
ผู้ใช้	ระบบ					
1. ไปที่หน้าคุ้มครอง ส่วนบุคคล	2. ระบบแสดงข้อมูลทั้งหมดที่เป็นส่วนบุคคล					
Exception Conditions:	ไม่มีข้อมูลส่วนบุคคลอยู่ในระบบ					

ตารางที่ 3.16: รายละเอียดยูสเคส คุ้มครองข้อมูลส่วนบุคคล

3.3.2.17 รายละเอียดมูสคेस ดูรายละเอียดของข้อสอบสารณะ

Use Case No:	16							
Use Case Name:	ดูรายละเอียดของข้อสอบสารณะ							
Use Case Scenario:	ผู้ใช้ต้องการดูรายละเอียดของข้อสอบสารณะ							
Triggering Event:	<ul style="list-style-type: none"> - เมื่อผู้ใช้มีความต้องการที่จะดูรายละเอียดของข้อสอบสารณะ เพื่อที่จะนำไปเป็นแนวทางในข้อสอบของตนเอง - ผู้ใช้ต้องการทำการทำคัดลอกข้อสอบสารณะไปปรับแต่งหรือนำไปใช้กับผู้สมัครสอบ 							
Brief Description:	สำหรับให้ผู้ใช้คุ่าว่ามีข้อสอบสารณะใดบ้างที่อยู่ในระบบ และสามารถคัดลอกไปใช้ได้							
Actors:	พนักงาน							
Related Use Cases:	เข้าสู่ระบบ							
Stakeholders:	-							
Pre - Conditions:	ผู้ใช้ต้องทำการเข้าสู่ระบบ							
Post - Conditions:	ระบบทำการแสดงรายละเอียดของข้อสอบ							
Flow of Events:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">ผู้ใช้</th> <th style="text-align: center;">ระบบ</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1. ไปที่หน้าข้อสอบ สารณะ</td> <td style="text-align: center;">2. ระบบแสดงข้อสอบ ทั้งหมดที่เป็นสารณะ</td> </tr> <tr> <td style="text-align: center;">3. คลิกคูข้อสอบที่ต้องการดู รายละเอียด</td> <td style="text-align: center;">4. ระบบแสดงรายละเอียด ข้อสอบ</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. ไปที่หน้าข้อสอบ สารณะ	2. ระบบแสดงข้อสอบ ทั้งหมดที่เป็นสารณะ	3. คลิกคูข้อสอบที่ต้องการดู รายละเอียด	4. ระบบแสดงรายละเอียด ข้อสอบ	
ผู้ใช้	ระบบ							
1. ไปที่หน้าข้อสอบ สารณะ	2. ระบบแสดงข้อสอบ ทั้งหมดที่เป็นสารณะ							
3. คลิกคูข้อสอบที่ต้องการดู รายละเอียด	4. ระบบแสดงรายละเอียด ข้อสอบ							
Exception Conditions:	ไม่มีข้อสอบสารณะอยู่ในระบบ							

ตารางที่ 3.17: รายละเอียดมูสคेस ดูข้อสอบสารณะ

3.3.2.18 รายละเอียดยูสเคส กัดลอกข้อสอบสารณะ

Use Case No:	17									
Use Case Name:	กัดลอกข้อสอบสารณะ									
Use Case Scenario:	ผู้ใช้ต้องการกัดลอกข้อสอบสารณะ									
Triggering Event:	เมื่อผู้ใช้มีความต้องการที่จะนำข้อสอบสารณะไปใช้ หรือนำไปปรับแก้ เพื่อนำไปใช้กับผู้สมัครงาน									
Brief Description:	สำหรับให้ผู้ใช้กัดลอกข้อสอบไปใช้ หรือนำไปปรับแก้ตามที่ผู้ใช้ต้องการ									
Actors:	พนักงาน									
Related Use Cases:	เข้าสู่ระบบ, ดูข้อสอบสารณะ									
Stakeholders:	-									
Pre - Conditions:	ผู้ใช้ต้องทำการเข้าสู่ระบบ									
Post - Conditions:	ระบบทำการกัดลอกข้อสอบสารณะที่ผู้ใช้ต้องการ									
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1. ไปที่หน้าข้อสอบสารณะ</td> <td>2. ระบบแสดงข้อสอบทั้งหมดที่เป็นสารณะ</td> </tr> <tr> <td>3. คลิกดูข้อสอบที่ต้องการดูรายละเอียด</td> <td>4. ระบบแสดงหน้าค้างยืนยันการกัดลอก</td> </tr> <tr> <td>5. ผู้ใช้ทำการยืนยันการกัดลอกข้อสอบ</td> <td>6. ระบบทำการกัดลอกข้อสอบที่ผู้ใช้ต้องการ และหมวดหมู่คำถานทั้งหมดที่ใช้ในข้อสอบนั้น ไปเก็บไว้ในข้อสอบของตนเองและหมวดหมู่คำถานของตนเอง</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. ไปที่หน้าข้อสอบสารณะ	2. ระบบแสดงข้อสอบทั้งหมดที่เป็นสารณะ	3. คลิกดูข้อสอบที่ต้องการดูรายละเอียด	4. ระบบแสดงหน้าค้างยืนยันการกัดลอก	5. ผู้ใช้ทำการยืนยันการกัดลอกข้อสอบ	6. ระบบทำการกัดลอกข้อสอบที่ผู้ใช้ต้องการ และหมวดหมู่คำถานทั้งหมดที่ใช้ในข้อสอบนั้น ไปเก็บไว้ในข้อสอบของตนเองและหมวดหมู่คำถานของตนเอง	
ผู้ใช้	ระบบ									
1. ไปที่หน้าข้อสอบสารณะ	2. ระบบแสดงข้อสอบทั้งหมดที่เป็นสารณะ									
3. คลิกดูข้อสอบที่ต้องการดูรายละเอียด	4. ระบบแสดงหน้าค้างยืนยันการกัดลอก									
5. ผู้ใช้ทำการยืนยันการกัดลอกข้อสอบ	6. ระบบทำการกัดลอกข้อสอบที่ผู้ใช้ต้องการ และหมวดหมู่คำถานทั้งหมดที่ใช้ในข้อสอบนั้น ไปเก็บไว้ในข้อสอบของตนเองและหมวดหมู่คำถานของตนเอง									
Exception Conditions:	ไม่มีข้อสอบสารณะอยู่ในระบบ									

ตารางที่ 3.18: รายละเอียดยูสเคส กัดลอกข้อสอบสารณะ

3.3.2.19 รายละเอียดยูสเคส ส่งข้อสอบให้ผู้สมัครงาน

Use Case No:	18									
Use Case Name:	ส่งข้อสอบให้ผู้สมัครงาน									
Use Case Scenario:	ผู้ใช้งานต้องการส่งข้อสอบให้ผู้สมัครทำ									
Triggering Event:	เมื่อผู้ใช้สร้างข้อสอบเรียบร้อยแล้ว หรือไปทำการคัดลอกข้อสอบมาจากสารณะ และต้องการส่งข้อสอบให้ผู้สมัครงานทำ									
Brief Description:	สำหรับให้ผู้ใช้ส่งข้อสอบให้ผู้สมัครงานผ่านทางอีเมล									
Actors:	พนักงาน									
Related Use Cases:	เข้าสู่ระบบ, สร้างข้อสอบ, คัดลอกข้อสอบสารณะ									
Stakeholders:	-									
Pre - Conditions:	<ul style="list-style-type: none"> - ผู้ใช้ต้องทำการเข้าสู่ระบบ - ผู้ใช้ต้องทำการสร้างข้อสอบ หรือคัดลอกข้อสอบจากข้อสอบสารณะ 									
Post - Conditions:	ระบบทำการส่งข้อสอบให้ผู้สมัครงานผ่านอีเมล									
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1. ไปที่หน้าข้อสอบของตนเอง</td> <td>2. ระบบแสดงข้อสอบทั้งหมดของผู้ใช้</td> </tr> <tr> <td>3. คลิกปุ่มส่งข้อสอบบริเวณข้อสอบที่ต้องการส่ง</td> <td>4. ระบบแสดงกรอบหน้าต่างให้ใส่รายละเอียดผู้สมัครงาน เช่น อีเมลผู้สมัคร, ระยะเวลาในการทำข้อสอบ และวันหมดอายุของข้อสอบ</td> </tr> <tr> <td>5. กรอกรายละเอียดแล้วคลิกส่งข้อสอบให้ผู้สมัคร</td> <td>6. ระบบทำการสั่นคำตามจากหมวดหมู่คำตามที่กำหนดจำนวนข้อและความยากโดยรายละเอียดของข้อสอบ แล้วส่งให้ผู้สมัครงานผ่านอีเมล</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. ไปที่หน้าข้อสอบของตนเอง	2. ระบบแสดงข้อสอบทั้งหมดของผู้ใช้	3. คลิกปุ่มส่งข้อสอบบริเวณข้อสอบที่ต้องการส่ง	4. ระบบแสดงกรอบหน้าต่างให้ใส่รายละเอียดผู้สมัครงาน เช่น อีเมลผู้สมัคร, ระยะเวลาในการทำข้อสอบ และวันหมดอายุของข้อสอบ	5. กรอกรายละเอียดแล้วคลิกส่งข้อสอบให้ผู้สมัคร	6. ระบบทำการสั่นคำตามจากหมวดหมู่คำตามที่กำหนดจำนวนข้อและความยากโดยรายละเอียดของข้อสอบ แล้วส่งให้ผู้สมัครงานผ่านอีเมล	
ผู้ใช้	ระบบ									
1. ไปที่หน้าข้อสอบของตนเอง	2. ระบบแสดงข้อสอบทั้งหมดของผู้ใช้									
3. คลิกปุ่มส่งข้อสอบบริเวณข้อสอบที่ต้องการส่ง	4. ระบบแสดงกรอบหน้าต่างให้ใส่รายละเอียดผู้สมัครงาน เช่น อีเมลผู้สมัคร, ระยะเวลาในการทำข้อสอบ และวันหมดอายุของข้อสอบ									
5. กรอกรายละเอียดแล้วคลิกส่งข้อสอบให้ผู้สมัคร	6. ระบบทำการสั่นคำตามจากหมวดหมู่คำตามที่กำหนดจำนวนข้อและความยากโดยรายละเอียดของข้อสอบ แล้วส่งให้ผู้สมัครงานผ่านอีเมล									
Exception Conditions:	กรอกข้อมูลไม่ครบ หรือไม่ตรงตามเงื่อนไข									

ตารางที่ 3.19: รายละเอียดยูสเคส ส่งข้อสอบให้ผู้สมัครงาน

3.3.2.20 รายละเอียดมูสคेस คุปะร่วติการส่งข้อสอบทั้งหมด

Use Case No:	19					
Use Case Name:	คุปะร่วติการส่งข้อสอบทั้งหมด					
Use Case Scenario:	ผู้ใช้ต้องการคุปะร่วติการส่งข้อสอบ					
Triggering Event:	เมื่อผู้ใช้ส่งข้อสอบ แล้วต้องการคุปะร่วติการส่งข้อสอบที่ผู้ใช้ได้ส่งไปให้พนักงานทำ					
Brief Description:	สำหรับให้ผู้ใช้ที่ต้องการทราบคุปะร่วติการส่งข้อสอบของตนเอง					
Actors:	พนักงาน					
Related Use Cases:	เข้าสู่ระบบ, ส่งข้อสอบให้ผู้สมัคร					
Stakeholders:	-					
Pre - Conditions:	<ul style="list-style-type: none"> - ผู้ใช้ต้องทำการเข้าสู่ระบบ - ผู้ใช้ต้องทำการส่งข้อสอบให้ผู้สมัคร สาธารณ 					
Post - Conditions:	ระบบทำการแสดงคุปะร่วติการส่งข้อสอบที่ผู้ใช้ทำการส่งให้ผู้สมัคร					
Flow of Events:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">ผู้ใช้</th> <th style="text-align: center;">ระบบ</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1.ไปที่หน้าของคุปะร่วติการ ส่งข้อสอบ</td> <td style="text-align: center;">2.ระบบแสดงคุปะร่วติข้อ สอบทั้งหมดที่ผู้ใช้ทำการ ส่งให้ผู้สมัครงาน</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1.ไปที่หน้าของคุปะร่วติการ ส่งข้อสอบ	2.ระบบแสดงคุปะร่วติข้อ สอบทั้งหมดที่ผู้ใช้ทำการ ส่งให้ผู้สมัครงาน	
ผู้ใช้	ระบบ					
1.ไปที่หน้าของคุปะร่วติการ ส่งข้อสอบ	2.ระบบแสดงคุปะร่วติข้อ สอบทั้งหมดที่ผู้ใช้ทำการ ส่งให้ผู้สมัครงาน					
Exception Conditions:	ไม่มีการส่งข้อสอบให้ผู้สมัครงาน					

ตารางที่ 3.20: รายละเอียดมูสคेस คุปะร่วติการส่งข้อสอบทั้งหมด

3.3.2.21 รายละเอียดยูสเคส คุรายละเอียดประวัติการส่งข้อสอบ

Use Case No:	20							
Use Case Name:	คุรายละเอียดประวัติการส่งข้อสอบ							
Use Case Scenario:	เมื่อผู้ใช้ส่งข้อสอบให้ผู้สมัครงานทำ แล้วต้องการคุรายละเอียดว่า ส่งให้ผู้สมัครคนไหน และส่งไปเมื่อไหร่							
Triggering Event:	เมื่อผู้ใช้งานส่งข้อสอบให้ผู้สมัครทำ แล้วต้องการคุรายละเอียด ของการส่งข้อสอบให้ผู้สมัครงาน							
Brief Description:	สำหรับให้ผู้ใช้ที่ต้องการทราบประวัติการส่งข้อสอบให้ผู้สมัคร เช่น ส่งให้ผู้สมัครคนไหนบ้าง, ให้เวลาในการทำเท่าไหร่ และกำหนดวันหมดอายุไว้วันไหน							
Actors:	พนักงาน							
Related Use Cases:	เข้าสู่ระบบ, ส่งข้อสอบให้ผู้สมัคร							
Stakeholders:	-							
Pre - Conditions:	<ul style="list-style-type: none"> - ผู้ใช้ต้องทำการเข้าสู่ระบบ - ผู้ใช้ต้องทำการส่งข้อสอบให้ผู้สมัคร 							
Post - Conditions:	ระบบทำการแสดงรายละเอียดของประวัติข้อสอบที่มีการส่งให้ผู้สมัคร							
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1.ไปที่หน้าของประวัติการส่งข้อสอบ</td> <td>2.ระบบแสดงประวัติข้อสอบทั้งหมดที่ผู้ใช้ทำการส่งให้ผู้สมัครงาน</td> </tr> <tr> <td>3.คลิกประวัติการส่งข้อสอบที่ต้องการคุรายละเอียด</td> <td>4.ระบบทำการแสดงรายละเอียดประวัติข้อสอบที่ต้องการ</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1.ไปที่หน้าของประวัติการส่งข้อสอบ	2.ระบบแสดงประวัติข้อสอบทั้งหมดที่ผู้ใช้ทำการส่งให้ผู้สมัครงาน	3.คลิกประวัติการส่งข้อสอบที่ต้องการคุรายละเอียด	4.ระบบทำการแสดงรายละเอียดประวัติข้อสอบที่ต้องการ	
ผู้ใช้	ระบบ							
1.ไปที่หน้าของประวัติการส่งข้อสอบ	2.ระบบแสดงประวัติข้อสอบทั้งหมดที่ผู้ใช้ทำการส่งให้ผู้สมัครงาน							
3.คลิกประวัติการส่งข้อสอบที่ต้องการคุรายละเอียด	4.ระบบทำการแสดงรายละเอียดประวัติข้อสอบที่ต้องการ							
Exception Conditions:	ไม่มีการส่งข้อสอบให้ผู้สมัครงาน							

ตารางที่ 3.21: รายละเอียดยูสเคส คุรายละเอียดประวัติการส่งข้อสอบ

3.3.2.22 รายละเอียดัญญาสเคส 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:	<table border="1"> <tr> <td style="text-align: center;">ผู้ใช้</td> <td style="text-align: center;">ระบบ</td> </tr> <tr> <td>1. คลิกที่ลิงค์ในอีเมลที่ได้รับจากทางบริษัท</td> <td>2. ระบบทำการแสดงรายละเอียดข้อสอบ</td> </tr> </table>	ผู้ใช้	ระบบ	1. คลิกที่ลิงค์ในอีเมลที่ได้รับจากทางบริษัท	2. ระบบทำการแสดงรายละเอียดข้อสอบ	
ผู้ใช้	ระบบ					
1. คลิกที่ลิงค์ในอีเมลที่ได้รับจากทางบริษัท	2. ระบบทำการแสดงรายละเอียดข้อสอบ					
Exception Conditions:						

ตารางที่ 3.22: รายละเอียดัญญาสเคส รายละเอียดข้อสอบ

3.3.2.23 รายละเอียดยูสเคส เริ่มทำข้อสอบ

Use Case No:	22					
Use Case Name:	เริ่มทำข้อสอบ					
Use Case Scenario:	ผู้สมัครกดเริ่มทำข้อสอบ					
Triggering Event:	เมื่อสมัครงานต้องการเริ่มทำข้อสอบ หลังจากที่คุรายละเอียดของข้อสอบแล้ว					
Brief Description:	สำหรับให้ผู้สมัครงานทำข้อสอบที่ได้รับมาจากทางบริษัท					
Actors:	ผู้สมัครงาน					
Related Use Cases:	รายละเอียดข้อสอบ					
Stakeholders:	-					
Pre - Conditions:	ผู้สมัครต้องทำการเข้าอีเมลเพื่อเปิดลิงค์ที่ทางบริษัทได้ทำการส่งมา					
Post - Conditions:	ระบบแสดงข้อสอบให้ผู้สมัครและเริ่มจับเวลาในการทำข้อสอบ					
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1. คลิกเริ่มทำข้อสอบ</td> <td>2. ระบบ แสดง ข้อสอบ ให้ ผู้ สมัคร และเริ่มจับเวลาในการ ทำข้อสอบ</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. คลิกเริ่มทำข้อสอบ	2. ระบบ แสดง ข้อสอบ ให้ ผู้ สมัคร และเริ่มจับเวลาในการ ทำข้อสอบ	
ผู้ใช้	ระบบ					
1. คลิกเริ่มทำข้อสอบ	2. ระบบ แสดง ข้อสอบ ให้ ผู้ สมัคร และเริ่มจับเวลาในการ ทำข้อสอบ					
Exception Conditions:	<ul style="list-style-type: none"> - ผู้สมัครส่งข้อสอบไปแล้ว จึงไม่สามารถทำใหม่อีกครั้งได้ - ข้อสอบหมดอายุการใช้งานแล้ว 					

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:	เมื่อผู้สมัครงานทำการส่งข้อสอบเสร็จแล้ว และต้องการส่งข้อสอบให้ทางบริษัท					
Brief Description:	สำหรับให้ผู้สมัครงานทำการส่งข้อสอบ					
Actors:	ผู้สมัครงาน					
Related Use Cases:	เริ่มทำการส่ง					
Stakeholders:	-					
Pre - Conditions:	ผู้สมัครต้องทำการกดเริ่มทำการส่ง ที่ทางบริษัทส่งมา					
Post - Conditions:	ระบบทำการส่งข้อสอบให้ทางบริษัท					
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1. คลิกส่งข้อสอบ</td> <td>2. ระบบทำการส่งข้อสอบให้ทางบริษัท</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. คลิกส่งข้อสอบ	2. ระบบทำการส่งข้อสอบให้ทางบริษัท	
ผู้ใช้	ระบบ					
1. คลิกส่งข้อสอบ	2. ระบบทำการส่งข้อสอบให้ทางบริษัท					
Exception Conditions:	<ul style="list-style-type: none"> - ผู้สมัครส่งข้อสอบไปแล้ว จึงไม่สามารถทำการส่งใหม่อีกครั้งได้ - ข้อสอบหมดอายุการใช้งานแล้ว 					

ตารางที่ 3.24: รายละเอียดยูสเคส ส่งข้อสอบให้ทางบริษัท

3.3.2.25 รายละเอียดยูสเคส หมวดเวลาทำข้อสอบ

Use Case No:	24					
Use Case Name:	หมวดเวลาทำข้อสอบ					
Use Case Scenario:	ผู้สมัครงานทำข้อสอบเกินเวลาที่กำหนด					
Triggering Event:	เมื่อผู้สมัครงานทำข้อสอบอยู่แล้วหมวดเวลาทำข้อสอบ โดยที่ยังไม่ได้ทำการส่งข้อสอบ					
Brief Description:	สำหรับกรณีที่ผู้สมัครทำข้อสอบไม่ทันในเวลาที่กำหนด					
Actors:	ผู้สมัครงาน					
Related Use Cases:	เริ่มทำข้อสอบ					
Stakeholders:	-					
Pre - Conditions:	ผู้ใช้ต้องคลิกเริ่มทำข้อสอบ และทำงานหมวดเวลาโดยที่ยังไม่ได้ส่ง					
Post - Conditions:	ระบบทำการส่งข้อสอบให้ทางบริษัท					
Flow of Events:	<table border="1"> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> <tr> <td></td> <td>1. ระบบตรวจสอบว่าหมวดเวลาในการทำข้อสอบ แล้วทำการส่งข้อสอบให้ทางบริษัท</td> </tr> </table>	ผู้ใช้	ระบบ		1. ระบบตรวจสอบว่าหมวดเวลาในการทำข้อสอบ แล้วทำการส่งข้อสอบให้ทางบริษัท	
ผู้ใช้	ระบบ					
	1. ระบบตรวจสอบว่าหมวดเวลาในการทำข้อสอบ แล้วทำการส่งข้อสอบให้ทางบริษัท					
Exception Conditions:	-					

ตารางที่ 3.25: รายละเอียดยูสเคส หมวดเวลาทำข้อสอบ

3.3.2.26 รายละเอียดยูสเคส คุ้มข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว

Use Case No:	25					
Use Case Name:	คุ้มข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว					
Use Case Scenario:	ผู้ใช้คุ้มข้อสอบที่ผู้สมัครงานคนใหม่ทำข้อสอบเสร็จแล้วบ้าง					
Triggering Event:	เมื่อผู้ใช้ต้องการคุ้มข้อสอบที่ผู้สมัครงานคนใหม่ทำข้อสอบเสร็จแล้วบ้าง					
Brief Description:	สำหรับกรณีที่ผู้ใช้ต้องการคุ้มข้อสอบที่ผู้สมัครงานคนใหม่ทำข้อสอบเสร็จแล้วบ้าง					
Actors:	พนักงาน					
Related Use Cases:	เข้าสู่ระบบ, ส่งข้อสอบให้ทางบริษัท					
Stakeholders:	-					
Pre - Conditions:	<ul style="list-style-type: none"> - ผู้ใช้ต้องทำการเข้าสู่ระบบ - ผู้ใช้ต้องทำการส่งข้อสอบให้ผู้สมัครงาน - ผู้สมัครงานต้องส่งข้อสอบให้บริษัทเรียบร้อยแล้ว 					
Post - Conditions:	ระบบทำการแสดงข้อสอบทั้งหมดที่ผู้สมัครงานทำเสร็จแล้ว					
Flow of Events:	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">ผู้ใช้</th> <th style="text-align: center;">ระบบ</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1. ไปที่หน้าข้อสอบที่ทำเสร็จแล้ว</td> <td style="text-align: center;">2. ระบบแสดงข้อสอบทั้งหมดที่ผู้สมัครทำเสร็จแล้ว</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1. ไปที่หน้าข้อสอบที่ทำเสร็จแล้ว	2. ระบบแสดงข้อสอบทั้งหมดที่ผู้สมัครทำเสร็จแล้ว	
ผู้ใช้	ระบบ					
1. ไปที่หน้าข้อสอบที่ทำเสร็จแล้ว	2. ระบบแสดงข้อสอบทั้งหมดที่ผู้สมัครทำเสร็จแล้ว					
Exception Conditions:	ไม่มีข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว					

ตารางที่ 3.26: รายละเอียดยูสเคส คุ้มข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว

3.3.2.27 รายละเอียดยูสเคส ตรวจสอบข้อสอบที่ผู้สมัครทำเสร็จแล้ว

Use Case No:	26									
Use Case Name:	ตรวจสอบข้อสอบที่ผู้สมัครทำเสร็จแล้ว									
Use Case Scenario:	ผู้ใช้ตรวจสอบข้อสอบที่ผู้สมัครเสร็จแล้ว									
Triggering Event:	เมื่อผู้ใช้ต้องการตรวจสอบข้อสอบที่มีคำ답ນประกอบอัตนัย และคำ답น์ที่มีไฟล์แนบ ซึ่งไม่มีคำตอบที่ถูกต้อง									
Brief Description:	สำหรับให้ผู้ใช้ตรวจสอบข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว									
Actors:	พนักงาน									
Related Use Cases:	เข้าสู่ระบบ, ส่งข้อสอบให้ทางบริษัท									
Stakeholders:	-									
Pre - Conditions:	<ul style="list-style-type: none"> - ผู้ใช้ต้องทำการเข้าสู่ระบบ - ผู้ใช้ต้องทำการส่งข้อสอบให้ผู้สมัครงาน - ผู้สมัครงานต้องส่งข้อสอบให้บริษัทเรียบร้อยแล้ว 									
Post - Conditions:	ระบบทำการบันทึกคะแนน									
Flow of Events:	<table border="1"> <thead> <tr> <th>ผู้ใช้</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1.ไปที่หน้าข้อสอบทั้งหมดที่ผู้สมัครงานทำเสร็จแล้ว</td> <td>2.ระบบแสดงข้อสอบทั้งหมดที่ผู้สมัครทำเสร็จแล้ว</td> </tr> <tr> <td>3.คลิกข้อสอบที่ต้องการทำการตรวจ</td> <td>4.ระบบแสดงหน้าตรวจข้อสอบ</td> </tr> <tr> <td>5.ผู้ใช้ทำการตรวจข้อสอบแล้วคลิกยืนยัน</td> <td>6.ระบบทำการบันทึกการตรวจข้อสอบแล้วสรุปผลออกมายเป็นคะแนน</td> </tr> </tbody> </table>	ผู้ใช้	ระบบ	1.ไปที่หน้าข้อสอบทั้งหมดที่ผู้สมัครงานทำเสร็จแล้ว	2.ระบบแสดงข้อสอบทั้งหมดที่ผู้สมัครทำเสร็จแล้ว	3.คลิกข้อสอบที่ต้องการทำการตรวจ	4.ระบบแสดงหน้าตรวจข้อสอบ	5.ผู้ใช้ทำการตรวจข้อสอบแล้วคลิกยืนยัน	6.ระบบทำการบันทึกการตรวจข้อสอบแล้วสรุปผลออกมายเป็นคะแนน	
ผู้ใช้	ระบบ									
1.ไปที่หน้าข้อสอบทั้งหมดที่ผู้สมัครงานทำเสร็จแล้ว	2.ระบบแสดงข้อสอบทั้งหมดที่ผู้สมัครทำเสร็จแล้ว									
3.คลิกข้อสอบที่ต้องการทำการตรวจ	4.ระบบแสดงหน้าตรวจข้อสอบ									
5.ผู้ใช้ทำการตรวจข้อสอบแล้วคลิกยืนยัน	6.ระบบทำการบันทึกการตรวจข้อสอบแล้วสรุปผลออกมายเป็นคะแนน									
Exception Conditions:	ไม่มีข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว									

ตารางที่ 3.27: รายละเอียดยูสเคส ดูข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว

3.3.3 การออกแบบฐานข้อมูล

ทางผู้จัดทำได้ใช้ฐานข้อมูลแบบไม่มีโครงสร้าง ในเว็บแอ�플ิเคชันนี้ ซึ่งเป็นฐานข้อมูลที่สามารถเก็บข้อมูลในปริมาณที่มากได้ สามารถอ่านเขียนข้อมูลได้อย่างรวดเร็ว และสามารถเก็บข้อมูลที่ซับซ้อนได้ซึ่งเว็บแอฟพลิกेशันประเมินความสามารถเบื้องต้นของผู้สมัครงาน ได้แบ่งประเภทของการเก็บข้อมูลออกเป็น 5 ตาราง ดังนี้

ชื่อตาราง	รายละเอียด
CATEGORIES	สำหรับจัดเก็บคำาณตามหมวดหมู่
EXAMS	สำหรับจัดเก็บรายละเอียดการของสุ่มคำาณในข้อสอบตามที่ผู้ออกข้อสอบกำหนด
EXAM_RANDOMS	จัดเก็บข้อสอบที่ถูกสุ่มคำาณมาจาก หมวดหมู่ต่างๆ ตามที่กำหนดอยู่ในรายละเอียดข้อสอบ
SEND_EXAMS	สำหรับจัดเก็บประวัติการส่งข้อสอบ
USERS	สำหรับจัดเก็บข้อมูลผู้ใช้งาน

ตารางที่ 3.28: ฐานข้อมูลทั้งหมดที่ใช้

3.4 รายละเอียดการพัฒนาระบบ

เว็บแอฟพลิกेशันประเมินความสามารถเบื้องต้นของผู้สมัครงาน ได้แบ่งการพัฒนาออกเป็น 3 ส่วนหลักๆ คือส่วนที่จัดการกับฐานข้อมูล, ส่วนต่อประสานกับผู้ใช้ และส่วนที่สร้างคอนเทนเนอร์เพื่อนำไปใช้ในการเดployment ด้วย Kubernetes

3.4.1 ส่วนที่จัดการกับฐานข้อมูล (back-end)

รายละเอียดของแต่ละเซอร์วิสของเว็บแอฟพลิกेशันประเมินความสามารถเบื้องต้นของผู้สมัครงาน ที่ทางผู้จัดทำได้พัฒนาขึ้น มีด้วยกันทั้งหมด 6 เซอร์วิสดังต่อไปนี้

1. **user service** เป็นเซอร์วิสที่จัดการกับข้อมูลผู้ใช้งาน เพื่อเข้าใช้งานระบบ และเป็นข้อมูลอ้างอิงในการสร้างส่วนต่างๆ ในระบบ โดยมีฟังก์ชันต่างๆ ดังนี้

- **create user** ฟังก์ชันสร้างผู้ใช้งาน เพื่อเข้าใช้งานระบบ

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

ตารางที่ 3.29: รายละเอียดฟังก์ชันสร้างผู้ใช้งาน

- **activate user** ฟังก์ชันยืนยันตัวตนเพื่อเข้าใช้งานระบบ หลังจากที่ทำการสมัครสมาชิก

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

ตารางที่ 3.30: รายละเอียดฟังก์ชันยืนยันตัวตนเพื่อเข้าใช้งานระบบ

- **login user** ฟังก์ชันการเข้าสู่ระบบ

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

ตารางที่ 3.31: รายละเอียดฟังก์ชันการเข้าสู่ระบบ

- **update user** ฟังก์ชันแก้ไขข้อมูลผู้ใช้งาน

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

ตารางที่ 3.32: รายละเอียดฟังก์ชันแก้ไขข้อมูลผู้ใช้งาน

- **logout user** ฟังก์ชันการออกจากระบบ

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

ตารางที่ 3.33: รายละเอียดฟังก์ชันการออกจากระบบ

- **logout all user** ฟังก์ชันการออกจากระบบ ทุกเครื่องที่มีการเข้าสู่ระบบด้วยผู้ใช้เดียวกัน

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

ตารางที่ 3.34: รายละเอียดฟังก์ชันการออกจากระบบทุกเครื่องที่เข้าใช้งาน

- **delete user** ฟังก์ชันการลบผู้ใช้งานออกจากระบบ

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

ตารางที่ 3.35: รายละเอียดฟังก์ชันการลบผู้ใช้งานออกจากระบบ

2. **category service** เป็นเซอร์วิสที่จัดการกับหมวดหมู่คำถาม โดยสร้างคำถามที่แบ่งตามเป็นหมวดหมู่ต่างๆ ตามที่ผู้ใช้ต้องการ พร้อมกำหนดระดับความยากของคำถามในแต่ละชื่อ

- **create category** ฟังก์ชันการสร้างหมวดหมู่คำถาม

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

ตารางที่ 3.36: รายละเอียดฟังก์ชันการสร้างหมวดหมู่คำถาม

- **upload file** ฟังก์ชันสำหรับเพิ่มรูปภาพกับไฟล์ต่างๆ ที่อยู่ในคำถาม หรือคำตอบ

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

ตารางที่ 3.37: รายละเอียดฟังก์ชันสำหรับเพิ่มรูปภาพกับไฟล์ต่างๆ

- **delete file** ฟังก์ชันการลบรูปภาพกับไฟล์ต่างๆ ที่อยู่ในคำาณ หรือคำตอบ

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

ตารางที่ 3.38: รายละเอียดฟังก์ชันสำหรับลบรูปภาพกับไฟล์ต่างๆ

- **update category** ฟังก์ชันการแก้ไขหมวดหมู่คำาณ หากมีการส่งข้อสอบให้ผู้สมัครงานทำแล้ว จะสามารถแก้ไขได้เฉพาะคำตอบ

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

ตารางที่ 3.39: รายละเอียดฟังก์ชันการแก้ไขหมวดหมู่คำาณ

- **get category details** ฟังก์ชันการดูคำาณที่อยู่ภายใต้หมวดหมู่คำาณ

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

ตารางที่ 3.40: รายละเอียดฟังก์ชันการดูคำาณที่อยู่ภายใต้หมวดหมู่คำาณ

- **get my categories** ฟังก์ชันการคูหมวดหมู่คำตามทั้งหมดของตนเอง

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

ตารางที่ 3.41: รายละเอียดฟังก์ชันการคูหมวดหมู่คำตามทั้งหมดของตนเอง

- **get public categories** ฟังก์ชันการคูหมวดหมู่คำตามทั้งหมดที่เป็นสาธารณะ และไม่ใช่หมวดหมู่คำตามของตนเอง

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

ตารางที่ 3.42: รายละเอียดฟังก์ชันการคูหมวดหมู่คำตามทั้งหมดที่เป็นสาธารณะ

- **get all categories for public exam** ฟังก์ชันการคูหมวดหมู่คำตามทั้งหมดที่ใช้สำหรับสร้างข้อสอบที่เป็นสาธารณะ

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

ตารางที่ 3.43: รายละเอียดฟังก์ชันการคูหมวดหมู่คำตามทั้งหมดที่ใช้สร้างข้อสอบสาธารณะ

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

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

ตารางที่ 3.44: รายละเอียดฟังก์ชันการคุ้มครองหมวดหมู่คำตามทั้งหมดที่ใช้สร้างข้อสอบที่ไม่เปิดเป็นสาธารณะ

- **delete category** พิ้งกี้ชั้นการลบหมวดหมู่คำตาม หากมีการใช้หมวดหมู่คำตามที่อยู่ในข้อสอบ จะไม่สามารถทำการลบได้

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

ตารางที่ 3.45: รายละเอียดฟังก์ชันการลบหมวดหมู่คำตาม

- **copy category** พิ้งกี้ชั้นการคัดลอกหมวดหมู่คำตาม ที่ต้องการคัดลอก เช่น คัดลอกหมวดหมู่คำตามสาธารณะ หรือคัดลอกหมวดหมู่คำตามของตนเอง

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

ตารางที่ 3.46: รายละเอียดฟังก์ชันการคัดลอกหมวดหมู่คำตาม

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

- **create exam** ฟังก์ชันการสร้างข้อสอบ โดยกำหนดรายละเอียดต่างๆ ของข้อสอบ เช่น ต้องการหมวดหมู่คำถามใดบ้าง, ต้องการกี่ข้อและระดับความยากเท่าใด, ให้เวลาในการทำข้อสอบเท่าใด และวันสุดท้ายที่สามารถทำข้อสอบได้ เมื่อกำหนดรายละเอียดต่างๆ ของข้อสอบ เรียบร้อยแล้ว ระบบจะสุ่มคำถามเมื่อส่งข้อสอบให้ผู้สมัครงานทำ

URL:	https://url/exam
Method:	POST
Auth require:	True
Format:	JSON
Parameters:	-
Body:	exam data
Response:	exam data

ตารางที่ 3.47: รายละเอียดฟังก์ชันการสร้างข้อสอบ

- **get exam details** ฟังก์ชันการดูรายละเอียดข้อสอบ เพื่อดูรายละเอียดภายในข้อสอบนั้นๆ

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

ตารางที่ 3.48: รายละเอียดฟังก์ชันการดูรายละเอียดข้อสอบ

- **update exam** ฟังก์ชันการแก้ไขรายละเอียดต่างๆภายในข้อสอบ

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

ตารางที่ 3.49: รายละเอียดฟังก์ชันการแก้ไขรายละเอียดต่างๆภายในข้อสอบ

- **get my exams** ฟังก์ชันการคูณข้อสอบทั้งหมดของตนเอง

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

ตารางที่ 3.50: รายละเอียดฟังก์ชันการคูณข้อสอบทั้งหมดของตนเอง

- **get public exams** ฟังก์ชันการคูณข้อสอบทั้งหมดที่เป็นสาธารณะ และไม่ใช่ข้อสอบของตนเอง

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

ตารางที่ 3.51: รายละเอียดฟังก์ชันการคูณข้อสอบทั้งหมดที่เป็นสาธารณะ

- **delete exam** ฟังก์ชันการลบข้อสอบของตนเอง

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

ตารางที่ 3.52: รายละเอียดฟังก์ชันการลบข้อสอบของตนเอง

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

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

ตารางที่ 3.53: รายละเอียดฟังก์ชันการคัดลอกข้อสอบ

4. **send exam service** เป็นเซอร์วิสที่ไว้ใช้ในการส่งข้อสอบให้ผู้สมัครงานทำ ผ่านทางอีเมล และจัดเก็บประวัติการส่งเอาไว้ โดยมีฟังก์ชันต่างๆดังนี้

- **send exam** ฟังก์ชันการส่งข้อสอบให้ผู้สมัครงานทำ โดยเมื่อผู้ใช้กดส่งข้อสอบ ระบบจะทำการสุ่มคำถามที่ข้อสอบนั้นได้กำหนดหมวดหมู่คำถามที่ต้องการ จำนวนข้อ และระดับความยาก แล้วส่งให้ผู้สมัครงานตามอีเมลที่กำหนด

URL:	https://url}/send-exam
Method:	POST
Auth require:	True
Format:	JSON
Parameters:	-
Body:	send exam data
Response:	send exam data

ตารางที่ 3.54: รายละเอียดฟังก์ชันการส่งข้อสอบให้ผู้สมัครงานทำ

- **get sent exams** ฟังก์ชันการคูประวัติการส่งข้อสอบให้ผู้สมัครงาน

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

ตารางที่ 3.55: รายละเอียดฟังก์ชันการคูประวัติการส่งข้อสอบให้ผู้สมัครงาน

- **delete send exams** ฟังก์ชันการลบประวัติการส่งข้อสอบ

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

ตารางที่ 3.56: รายละเอียดฟังก์ชันการลบประวัติการส่งข้อสอบ

5. **do exam service** เป็นเซอร์วิสที่ใช้สำหรับทำข้อสอบที่พนักงานส่งมาให้ โดยมีระยะเวลาในการทำข้อสอบที่จำกัด และมีวันหมดอายุหากผู้สมัครงานไม่ทำข้อสอบในเวลาที่กำหนด

- **get exam by id** ฟังก์ชันการคูรายละเอียดของข้อสอบก่อนที่ผู้สมัครจะเริ่มทำข้อสอบ โดยจะกล่าวถึงชื่อผู้สมัคร และรายละเอียดต่างๆของข้อสอบ ก่อนที่ผู้สมัครจะเริ่มทำข้อสอบ

URL:	https://{url}/do-exam/{:id}
Method:	GET
Auth require:	False
Format:	JSON
Parameters:	id(ID)
Body:	-
Response:	do exam data

ตารางที่ 3.57: รายละเอียดฟังก์ชันการดูรายละเอียดของข้อสอบก่อนที่ผู้สมัครจะเริ่มทำข้อสอบ

- **start do exam** ฟังก์ชันการเริ่มทำข้อสอบ โดยระบบจะเริ่มจับเวลาในการทำข้อสอบ และเปิดให้ผู้ใช้งานเมื่อทำข้อสอบได้

URL:	https://{url}/do-exam/start/{:id}
Method:	PATCH
Auth require:	False
Format:	JSON
Parameters:	id(ID)
Body:	-
Response:	do exam data

ตารางที่ 3.58: รายละเอียดฟังก์ชันการเริ่มทำข้อสอบ

- **send do exam** ฟังก์ชันการส่งข้อสอบเมื่อผู้สมัครงานทำข้อสอบเสร็จแล้วกดส่ง หรือหมดเวลาในการทำข้อสอบ ระบบจะทำการตรวจข้อสอบที่เป็นปนัยแล้วส่งข้อสอบให้ทางทางบริษัท พร้อมสรุปคะแนนในส่วนที่เป็นข้อสอบปนัย

URL:	https://{url}/do-exam/done/{:id}
Method:	PATCH
Auth require:	False
Format:	JSON
Parameters:	id(ID)
Body:	do exam data
Response:	do exam data

ตารางที่ 3.59: รายละเอียดฟังก์ชันการส่งข้อสอบของผู้สมัครงาน

- **upload file** ฟังก์ชันการเพิ่มไฟล์ หากทางบริษัทให้โจทย์ที่ต้องตอบกลับเป็นไฟล์ต่างๆ เช่น ไฟล์ที่เป็น Excel

URL:	https://url}/upload
Method:	PATCH
Auth require:	False
Format:	Format-data
Parameters:	-
Body:	(file)
Response:	URL to file path(STRING), filename(STRING)

ตารางที่ 3.60: รายละเอียดฟังก์ชันการเพิ่มไฟล์

- **delete file** ฟังก์ชันการลบไฟล์ ในกรณีที่ผู้สมัครงานได้ทำการเพิ่มไฟล์ไป แล้วต้องการแก้ไขไฟล์นั้น

URL:	https://url}/file
Method:	DELETE
Auth require:	False
Format:	JSON
Parameters:	-
Body:	filename(STRING)
Response:	delete status(STRING)

ตารางที่ 3.61: รายละเอียดฟังก์ชันการลบไฟล์

6. **finished exam service** เป็นเซอร์วิสที่ไว้สำหรับให้ทางบริษัทตรวจข้อสอบที่เป็นอัตโนมัติและข้อสอบที่มีการอัปโหลดไฟล์ แล้วสรุปผลออกมาเป็นคะแนน โดยมีฟังก์ชันต่างๆดังนี้

- **get finished exams** ฟังก์ชันการคูณข้อสอบทั้งหมด ที่ผู้สมัครทำเสร็จแล้ว

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

ตารางที่ 3.62: รายละเอียดฟังก์ชันการคูณข้อสอบทั้งหมด

- **get finished exam details** ฟังก์ชันการดูข้อมูลที่ผู้สมัครงานทำเสร็จแล้ว เพื่อทำการตรวจสอบข้อมูล

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

ตารางที่ 3.63: รายละเอียดฟังก์ชันการดูข้อมูลที่ผู้สมัครงานทำเสร็จแล้ว

- **verify finished exam** ฟังก์ชันการยืนยันการตรวจสอบ เพื่อให้ระบบทำการบันทึกคะแนนที่ทางบริษัททำการตรวจสอบข้อมูล

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

ตารางที่ 3.64: รายละเอียดฟังก์ชันการยืนยันการตรวจสอบข้อมูล

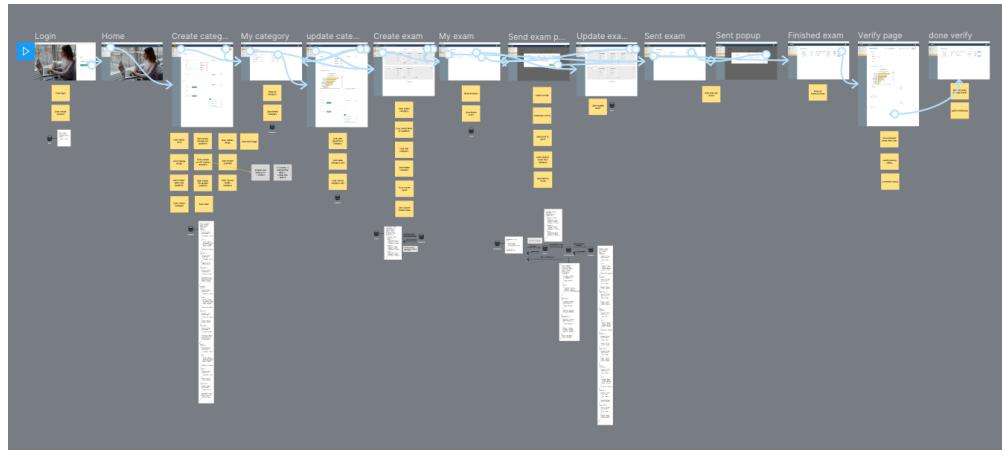
- **delete finished exam** ฟังก์ชันการลบข้อมูลที่ผู้สมัครงานทำเสร็จแล้ว

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

ตารางที่ 3.65: รายละเอียดฟังก์ชันการลบข้อมูลที่ผู้สมัครงานทำเสร็จแล้ว

3.4.2 ส่วนต่อประสานกับผู้ใช้ (front-end)

ในส่วนนี้ได้มีการออกแบบด้วย Figma ซึ่งเป็นเครื่องมือที่ไว้ใช้ในการออกแบบproto型ของงานก่อนที่จะเริ่มพัฒนา ด้วยการนำความต้องการต่างๆที่ได้รับมาออกแบบส่วนประสานกับผู้ใช้ โดยในแต่ละหน้าจะบันทึกว่ามีการใช้เซอร์วิสไหนในหน้านั้นบ้าง และแต่ละปุ่มของหน้านั้นกดแล้วปลายทางไปที่หน้าไหนดังรูปที่ 3.3 หลังจากนั้นทำการพัฒนาส่วนต่อประสานกับผู้ใช้ด้วย Vue.js



รูปที่ 3.3: แสดงตัวอย่างการใช้งาน Figma

3.4.3 สร้างคอนเทนเนอร์

ทั้งในส่วนต่อประสานกับผู้ใช้(front-end) และส่วนที่จัดการกับฐานข้อมูล(back-end) แต่ละส่วนจะถูกทำให้เป็นคอนเทนเนอร์โดยใช้ Docker เพื่อให้สะดวกต่อการ Deploy ด้วย Kubernetes โดยแต่ละส่วนจะมี Dockerfile ไว้สำหรับสร้างคอนเทนเนอร์อีกเมื่อของส่วนนั้นๆ โดยเราจะสร้างด้วยวิธี multi-stage build docker ที่จะมาช่วยในการลดขนาดของอิมเมจ ด้วยการนำของจากอิมเมจหนึ่งมาใส่ในอีกอิมเมจหนึ่ง เช่นดังรูป 3.4

```
Dockerfile ×
frontend > Dockerfile > ...
You, a week ago | 1 author (You)
1 # build stage
2 FROM node:14.7-slim as build-stage
3 WORKDIR /app
4 COPY package*.json .
5 RUN npm install
6 COPY . .
7 RUN npm run build
8
9 # production stage
10 FROM nginx:stable-alpine
11 COPY --from=build-stage /app/public /usr/share/nginx/html
12 EXPOSE 8080
13 CMD ["nginx", "-g", "daemon off;"] You, 3 months ago .
```

รูปที่ 3.4: แสดงการเขียน Dockerfile ด้วยการทำ multi-stage build docker

บทที่ 4

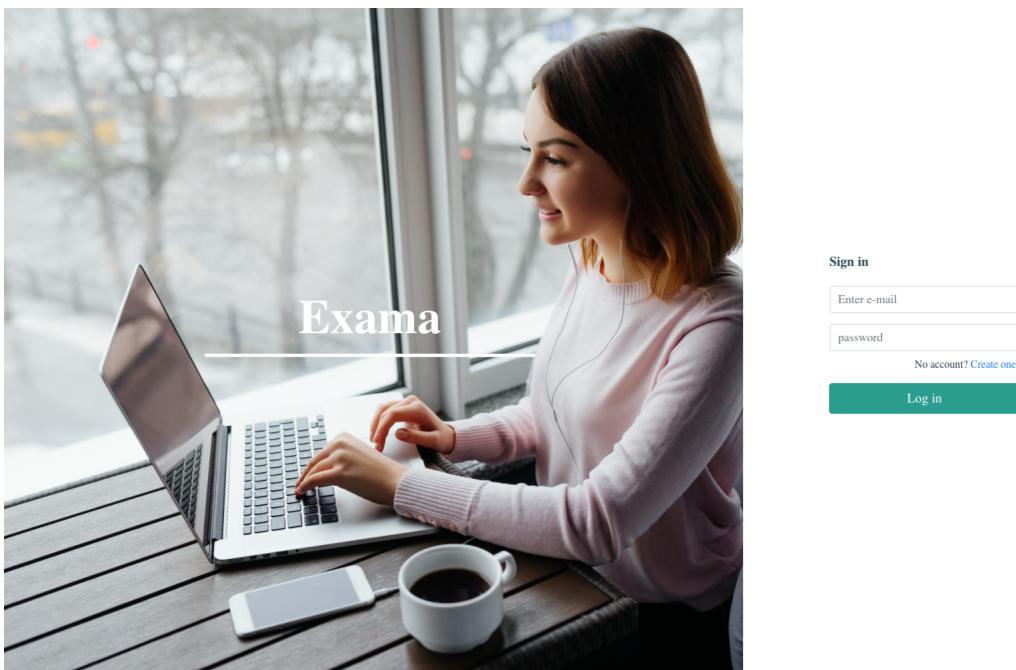
ผลการปฏิบัติงาน

การการปฏิบัติงานสหกิจศึกษาที่ บริษัท ไซเจ็น ด้วยตำแหน่ง Full Stack Developer เป็นระยะเวลา 4 เดือน ตั้งแต่วันที่ 3 สิงหาคม พ.ศ.2563 จนถึงวันที่ 30 พฤษภาคม พ.ศ.2563 สามารถสรุปผลการปฏิบัติงานได้ดังนี้

4.1 ผลการปฏิบัติงาน

ฟังก์ชันหลักของเว็บแอพพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน สามารถทำงานได้ตามที่ได้ทำการออกแบบไว้ โดยสามารถส่งข้อสอบให้ผู้สมัครงานทำได้ผ่านอีเมล โดยอัตโนมัติ และสามารถตรวจข้อสอบแล้วสรุปผลออกมาเป็นคะแนนได้ โดยในฟังก์ชันหลักที่สร้างขึ้น จะถูกใช้ในส่วนต่อประสานกับผู้ใช้งานแต่ละหน้าต่อไปนี้

4.1.1 หน้าเข้าสู่ระบบ

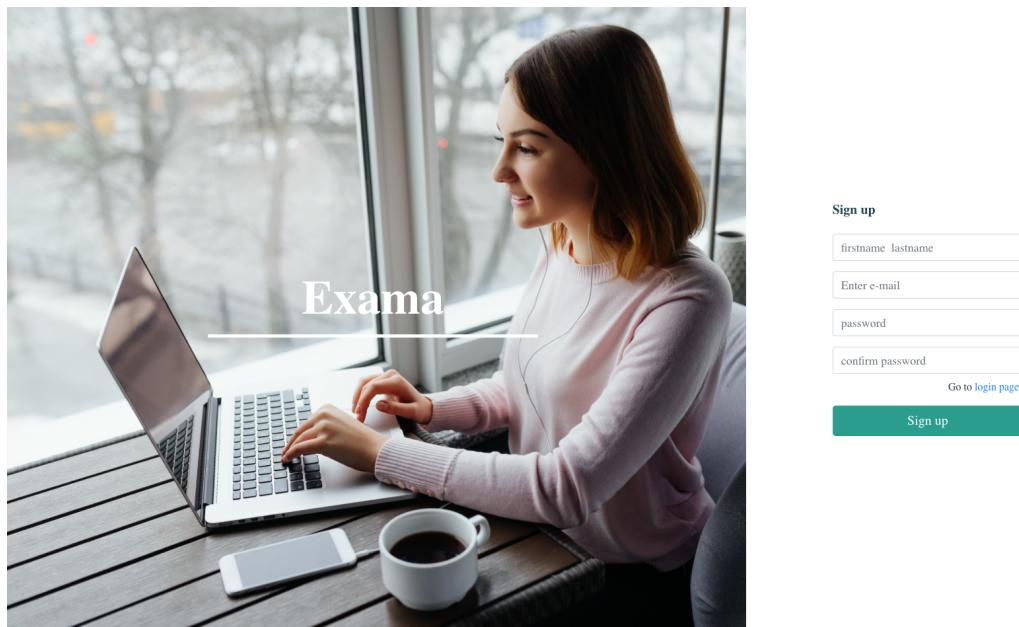


รูปที่ 4.1: แสดงหน้าเข้าสู่ระบบของเว็บแอพพลิเคชัน

ในหน้าเข้าสู่ระบบประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- Log in เพื่อเข้าสู่ระบบ
- Create one เพื่อสมัครสมาชิก

4.1.2 หน้าสมัครสมาชิก

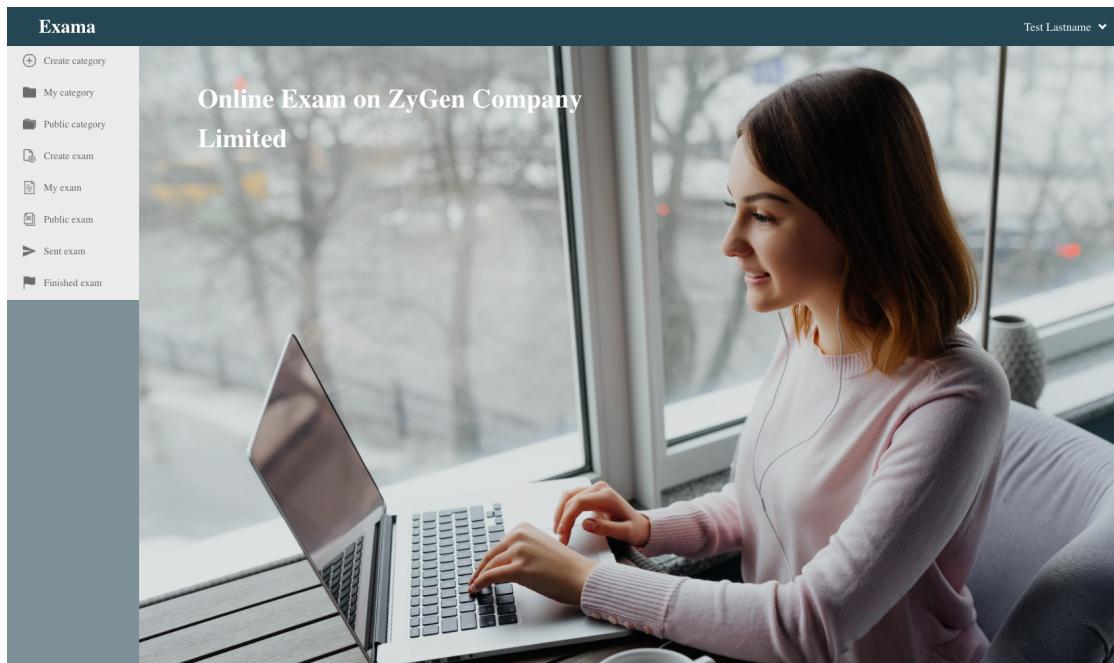


รูปที่ 4.2: แสดงหน้าสมัครสมาชิกสู่ระบบของเว็บแอพพลิเคชัน

ในหน้าสมัครสมาชิกประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- Sign up เพื่อสมัครสมาชิก ซึ่งจะมีอีเมลส่งไปที่ผู้สมัครสมาชิกเพื่อทำการยืนยันอีเมล หลังจากยืนยันอีเมลแล้ว ผู้ใช้งานถึงจะสามารถเข้าสู่ระบบได้
- Go to login page หากต้องการยกเลิกการสมัครสมาชิก

4.1.3 หน้าหลัก



รูปที่ 4.3: แสดงหน้าหลักของเว็บแอพพลิเคชัน

ในหน้าหลักประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- Side Navigation Menu เมนูเพื่อการเข้าถึงการทำงานต่างๆ
- logout เพื่อทำการออกสู่ระบบ โดยจะอยู่มุมบนซ้ายของหน้า หากนำเม้าส์ไปวางไว้ที่ซื่อของคันเรื่อง

4.1.4 หน้าสร้างหมวดหมู่คำถาม

The screenshot shows the 'Create Exam' interface. On the left is a sidebar with navigation links: 'Create category', 'My category', 'Public category', 'Create exam', 'My exam', 'Public exam', 'Sent exam', and 'Finished exam'. The main area has tabs for 'Easy', 'Medium', and 'Hard'. It displays three types of questions:

- Closed-ended question:** A section titled 'Closed-ended question' contains a question input field '1. Question ?' and four answer input fields labeled 'answer'. Each answer field has a small image icon and a red delete button.
- Open-ended question:** A section titled 'Open-ended question' contains a question input field '1. Question ?' and a text input field 'Answer...'. Both have image icons and red delete buttons.
- File question:** A section titled 'File question' contains a question input field '1. Question ?' and a file attachment input field 'attach file'. Both have image icons and red delete buttons.

At the top right, there are buttons for 'Category name', 'Create', and 'Cancel'. A status bar at the bottom says 'Test Lastname'.

รูปที่ 4.4: แสดงหน้าสร้างหมวดหมู่คำถามของเว็บแอปพลิเคชัน

ในหน้าสร้างหมวดหมู่คำถามประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้⁹

- ใส่รายละเอียดต่างๆ ของหมวดหมู่คำถาม
- สามารถกำหนดได้ว่าหมวดหมู่คำถามนี้ ต้องการเปิดเป็นสาธารณะให้ผู้อื่นใช้งานหรือไม่
- สามารถมีคำตอบที่ถูกมากกว่า 1 ข้อได้
- สามารถเพิ่มตัวเลือกคำตอบของคำ답นั้นได้
- สามารถเพิ่ม ลบ หรือแก้ไขรูปภาพได้
- สามารถเพิ่มไฟล์ต่างๆ ได้
- สามารถลบคำ답นั้นออกได้
- สามารถกำหนดความยากง่ายของคำ답นั้นได้

- สามารถแนบไฟล์ในคำถามได้
- สามารถกำหนดเฉลยของคำถามได้
- สามารถเพิ่มคำถามได้
- สามารถป้องกันการกดเปลี่ยนหน้า, การปิดหน้า หรือการรีเฟรชหน้าได้ หากผู้ใช้งานทำการป้อนข้อมูลลงไปแล้ว
- สามารถยกเลิกการสร้างได้

4.1.5 หน้าดูหมวดหมู่คำถามทั้งหมด

Name	Title	Copy	Copy from	Created by	Created at	Edit by	Edit at
TOEIC	reading test		-	Test Lastname	Nov 3, 2020	Test Lastname	Nov 3, 2020
Math105	математика		-	Test Lastname	Nov 3, 2020	Test Lastname	Nov 3, 2020
math103 (1)	aa		Test Lastname	Test Lastname	Nov 3, 2020	Test Lastname	Nov 3, 2020
math103	aa		-	Test Lastname	Nov 3, 2020	Test Lastname	Nov 3, 2020
english (6)	aa		Satwat Thitisupakul	Test Lastname	Oct 29, 2020	Test Lastname	Oct 29, 2020

รูปที่ 4.5: แสดงหน้าดูหมวดหมู่คำถามทั้งหมดของเว็บแอปพลิเคชัน

ในหน้าดูหมวดหมู่คำถามทั้งหมดประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- สามารถค้นหา หมวดหมู่ที่ต้องการได้
- สามารถลบได้ หากไม่ถูกใช้ในข้อสอบ
- สามารถคัดลอกหมวดหมู่คำถามของตนเองได้

4.1.6 หน้าแก้ไขหมวดหมู่คำถาม

รูปที่ 4.6: แสดงหน้าแก้ไขหมวดหมู่คำถามของเว็บแอพพลิเคชัน

ในหน้าแก้ไขหมวดหมู่คำถามประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- สามารถแก้ไขได้
- ถ้าหมวดหมู่คำถามนั้นนั้นมีการใช้ในข้อสอบ และถูกส่งให้ผู้สมัครงานแล้ว คำถามจะไม่สามารถแก้ไขได้ สามารถแก้ไขได้เพียงคำตอบดังรูป 4.6 เพื่อให้ผู้สมัครงานได้รับข้อสอบที่มีมาตรฐานเดียวกัน ถ้าต้องการแก้ไขส่วนอื่นๆ ให้ทำการคัดลอก เพื่อสร้างหมวดหมู่คำถามใหม่
- สามารถยกเลิกการแก้ไขได้

4.1.7 หน้าดูหมวดหมู่คำถามที่เปิดเป็นสาธารณะ

Name	Title	Copy	Created by	Created at	Edit by	Edit at
english	aa		Satwat Thitisupakul	Oct 12, 2020	Satwat Thitisupakul	Oct 12, 2020

รูปที่ 4.7: แสดงหน้าดูหมวดหมู่คำถามที่เปิดเป็นสาธารณะของเว็บแอพพลิเคชัน

ในหน้าดูหมวดหมู่คำถามที่เปิดเป็นสาธารณะประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- สามารถค้นหาได้
- สามารถคัดลอกหมวดหมู่คำถามเพื่อนำไปแก้ไขได้
- สามารถคลิกเข้าไปดูคำถามที่อยู่ในหมวดหมู่นั้นๆได้

4.1.8 หน้าตู้คำ答ในหมวดหมู่คำ答ที่เปิดเป็นสาธารณะ

The screenshot shows the Exama application interface. On the left, there's a sidebar with navigation links: Create category, My category, Public category (which is selected and highlighted in yellow), Create exam, My exam, Public exam, Sent exam, and Finished exam. The main area has a title 'category title : reading test' and a dropdown 'category name : TOEIC'. Below this, there are three tabs: Easy, Medium, and Hard. Under the 'Easy' tab, there's a section for 'Closed-ended question'. The first question is: '1. The report concerning the sales figures that you requested _____ on your desk.' with a note 'Multiple choice: off'. There are four radio buttons: 'is' (selected), 'am', 'are', and 'were'. The second question is: '2. The _____ company owners always dress appropriately when they need to attend business events and meetings.' with a note 'Multiple choice: off'. There are four radio buttons: 'succession', 'success', 'succeed', and 'successful'. The third question is: '3. Before _____ us your resume, be sure to check that it meets our requirements precisely.' with a note 'Multiple choice: off'. There is one radio button 'sends'.

รูปที่ 4.8: แสดงหน้าตู้คำ答ในหมวดหมู่คำ答ที่เปิดเป็นสาธารณะของเว็บแอพพลิเคชัน

ในหน้าตู้คำ答ในหมวดหมู่คำ答ที่เปิดเป็นสาธารณะประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- คุณภาพเชิงคุณภาพ และคำ答ในหมวดหมู่นั้น

4.1.9 หน้าสร้างข้อสอบเพื่อใช้สำหรับผู้สมัครงาน

The screenshot shows the Exama application interface for creating a new exam. The sidebar includes: Create category, My category, Public category, Create exam (selected and highlighted in yellow), My exam, Public exam, Sent exam, and Finished exam. The main area has a title 'Title' with 'TOEIC' entered. There are two sections: 'Closed-ended' and 'Open-ended'. In the 'Closed-ended' section, there are three rows for 'Easy', 'Medium', and 'Hard' levels, each with a dropdown menu containing '3', '0', and '0' respectively. In the 'Open-ended' section, there are three rows for 'Easy', 'Medium', and 'Hard' levels, each with a dropdown menu containing '0', '0', and '0' respectively. Both sections have a 'File upload' button with a trash icon. A dropdown menu at the bottom left lists 'math101', 'english', 'TOEIC (1)', 'Math103', 'TOEIC (2)', and 'Math103 (1)'. A green '+' button is located at the bottom center.

รูปที่ 4.9: แสดงหน้าสร้างข้อสอบเพื่อใช้สำหรับผู้สมัครงานของเว็บแอพพลิเคชัน

ในหน้าสร้างข้อสอบเพื่อใช้สำหรับผู้สมัครงานประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- ใส่รายละเอียดต่างๆของข้อสอบ
- สามารถกำหนดได้ว่าข้อสอบนี้ ต้องการเปิดเป็นสาธารณะให้ผู้อื่นใช้งานหรือไม่ ซึ่งถ้าเปิดข้อสอบให้เป็นสาธารณะ หมวดหมู่คำถามที่ใช้ในข้อสอบ จะสามารถใช้ได้เฉพาะหมวดหมู่คำถามที่เป็นสาธารณะด้วยเท่านั้น
- สามารถพิมพ์ค้นหาหมวดหมู่ของคำถามที่ต้องการได้
- สามารถเลือกจำนวนข้อของแต่ละระดับความยาก และประเภทของข้อสอบได้
- สามารถเพิ่มหมวดหมู่ได้ ตามหมวดหมู่ที่มีหัวข้อ
- สามารถยกเลิกการสร้างได้
- สามารถป้องกันการกดเปลี่ยนหน้า, การปิดหน้าหรือการรีเฟรชหน้าได้หากผู้ใช้งานทำการป้อนข้อมูลลงไปแล้ว

4.1.10 หน้าคุณข้อสอบของตนเอง

Name	Title	Copy from	Created by	Created at	Edit by	Edit at	
Fundamentals Exam	exam for everyone	<button>Send</button>	Satwat Thitisupakul	Nov 3, 2020	Satwat Thitisupakul	Now 3, 2020	
exammm3	this is exam	<button>Send</button>	-	Satwat Thitisupakul	Oct 28, 2020	Satwat Thitisupakul	Oct 28, 2020

รูปที่ 4.10: แสดงหน้าคุณข้อสอบตนเองของเว็บแอพพลิเคชัน

Name	Title	Copy from	Created by	Created at	Edit by	Edit at
Fundamentals Exam	exam for everyone	send	Test Lastname	Satwat Thitisupakul Nov 3, 2020	Satwat Thitisupakul	Nov 3, 2020
exammm3		send	Satwat Thitisupakul	Satwat Thitisupakul Oct 28, 2020		

The 'send-exam' modal window contains the following fields:

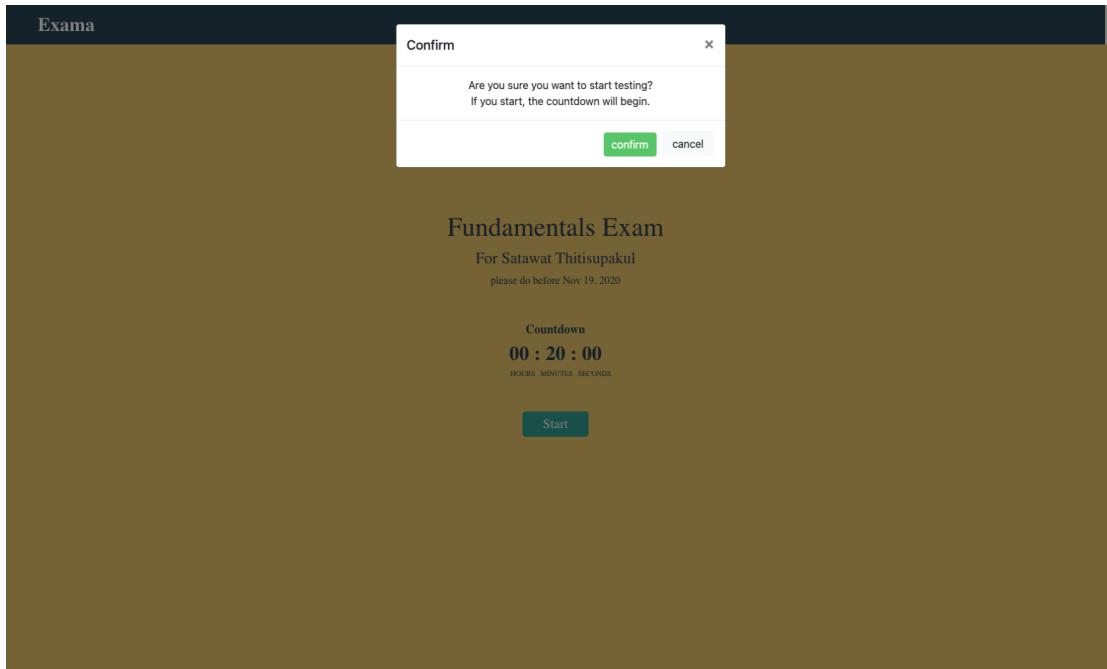
- From: exammm3
- Expire: No date selected
- Countdown: 00:20
- To (2 entries): Enter e-mail, Name (both empty)
- To (2 entries): Enter e-mail, Name (both empty)
- Buttons: Send, cancel

รูปที่ 4.11: แสดงหน้าดูข้อสอบตอนเองในขั้นตอนส่งข้อสอบให้ผู้สมัครงานของเว็บแอพพลิเคชัน

ในหน้าดูข้อสอบตอนขององค์ประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- สามารถดูข้อสอบทั้งหมดของตนเองได้
- สามารถค้นหาข้อสอบได้
- สามารถส่งข้อสอบให้ผู้สมัครกี่คนก็ได้ โดยระบบจะทำการสุ่มข้อสอบก่อนที่จะส่งให้ผู้สมัครงาน ดังรูป 4.11
- สามารถกำหนดวันที่หมดอายุของข้อสอบได้
- สามารถกำหนดระยะเวลาที่ให้ทำข้อสอบได้

4.1.11 หน้าดูรายละเอียดก่อนเริ่มทำข้อสอบ



รูปที่ 4.12: แสดงดูรายละเอียดก่อนเริ่มทำข้อสอบของเว็บแอพพลิเคชัน

ในหน้าดูรายละเอียดของข้อสอบสำหรับผู้สมัครงานประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- บอกรายละเอียดต่างๆ ก่อนเริ่มทำข้อสอบ
- มีการยืนยันการเริ่มทำข้อสอบ ก่อนที่ระบบจะเริ่มจัดเวลา ดังรูป 4.12
- หากข้อสอบหมดอายุ หรือผู้สมัครงานได้ทำข้อสอบนี้ไปแล้ว จะไม่สามารถทำข้อสอบได้

4.1.12 หน้าทำข้อสอบ

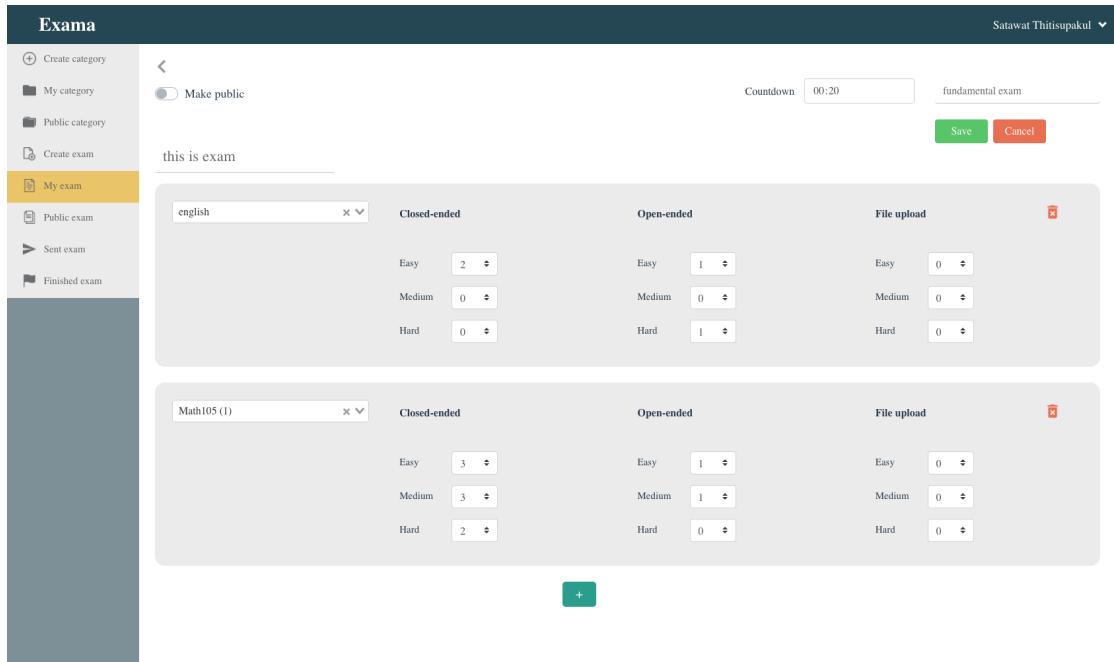
Exam	
Countdown 00 : 19 : 00 <small>HOURS MINUTES SECONDS</small>	<p>Fundamentals Exam exam for everyone</p> <p>1. กำหนดตัวมูล ตัวที่ 3 7 2 - 5 1 และค่าเฉลี่ยของตัวนี้คือเท่ากับเท่าใด (choose only one answer) <input type="radio"/> 1.6 <input type="radio"/> 2.5 <input type="radio"/> 3.6 <input type="radio"/> 4.8</p> <p>2. กำหนดตัวมูล ตัวที่ - 8 7 3 2 4 10 และค่าเฉลี่ยของตัวนี้คือเท่ากับเท่าใด (choose only one answer) <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 5 <input type="radio"/> 6</p> <p>3. ส่วนสูง (เซนติเมตร) ของหน้าร้านก่ออุบัติ เป็นตัวที่ 151 149 165 168 154 และมีน้ำหนักของตัวนี้คือเท่าไร (choose only one answer) <input type="radio"/> 154 เซนติเมตร <input type="radio"/> 157 เซนติเมตร <input type="radio"/> 165 เซนติเมตร <input type="radio"/> 168 เซนติเมตร</p>

รูปที่ 4.13: แสดงหน้าทำข้อสอบของเว็บแอพพลิเคชัน

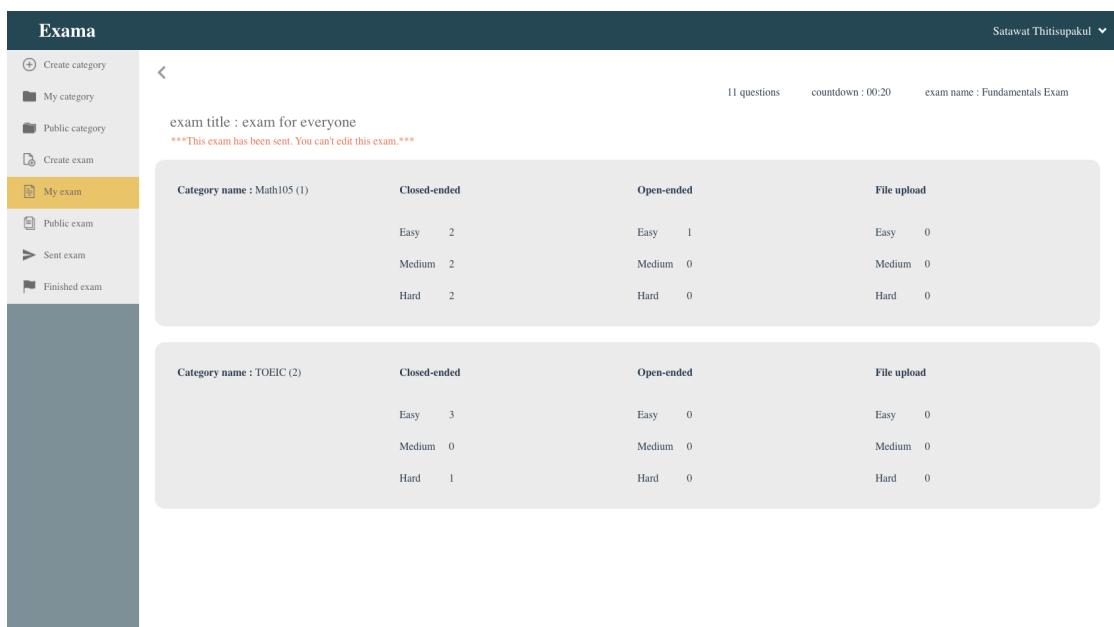
ในหน้าทำข้อสอบประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- ให้ผู้สมัครทำข้อสอบ
- มีการแสดงเวลาที่เหลืออยู่
- ผู้ใช้สามารถเข้ามาราทำได้ตลอด หากเวลาสิ้นไม่หมด
- สามารถส่งข้อสอบให้ทางบริษัทได้
- หากหมดเวลาสอบแล้วผู้ใช้งานไม่ได้ส่งข้อสอบให้ทางบริษัท ข้อสอบจะไม่สามารถทำต่อได้และระบบจะทำการส่งข้อสอบให้ทางบริษัททันที

4.1.13 หน้าแก้ไขข้อสอบเพื่อใช้สำหรับผู้สมัครงาน



รูปที่ 4.14: แสดงหน้าแก้ไขข้อสอบเพื่อใช้สำหรับส่งให้ผู้สมัครงานของเว็บแอปพลิเคชัน



รูปที่ 4.15: แสดงหน้าดูข้อสอบเพื่อใช้สำหรับส่งให้ผู้สมัครงานของเว็บแอปพลิเคชัน

ในหน้าแก้ไขข้อสอบเพื่อใช้สำหรับผู้สมัครงานประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- สามารถแก้ไขรายละเอียดต่างๆได้ ดังรูป 4.14
- หากข้อสอบถูกส่งไปแล้วจะไม่สามารถแก้ไขรายละเอียดต่างๆได้ ดังรูป 4.15

- สามารถป้องกันการกดเปลี่ยนหน้า, การปิดหน้าหรือการรีเฟรชหน้าได้หากผู้ใช้งานทำการป้อนข้อมูลลงไปแล้ว

4.1.14 หน้าดูข้อสอบที่เป็นข้อสอบสาธารณะ

The screenshot shows the Exama application interface. On the left, there is a sidebar with various options: Create category, My category, Public category, Create exam, My exam, **Public exam** (which is highlighted in yellow), Sent exam, and Finished exam. The main area is titled "Public exam" and displays a table with one row. The row contains: Name (Fundamentals Exam), Title (exam for everyone), Copy (button), Created by (Test Lastname), Created at (Nov 3, 2020), Edit by (Test Lastname), and Edit at (Nov 3, 2020). Below the table is a navigation bar with icons for back, forward, and search. At the top right, it says "Satwat Thitisupakul".

รูปที่ 4.16: แสดงหน้าดูข้อสอบที่เป็นข้อสอบสาธารณะของเว็บแอพพลิเคชัน

This screenshot shows a detailed view of a public exam. The sidebar on the left is identical to the previous one. The main area has a header with a back arrow, the exam title "exam for everyone", and status information: 11 questions, countdown: 00:20, and exam name: Fundamentals Exam. Below this, there are two sections. The first section is for "Category name : Math105" and the second for "Category name : TOEIC". Each section has tables for "Closed-ended" and "Open-ended" questions, and a "File upload" section. The Math105 section shows the following data:

Closed-ended		Open-ended		File upload	
Easy	2	Easy	1	Easy	0
Medium	2	Medium	0	Medium	0
Hard	2	Hard	0	Hard	0

The TOEIC section shows similar data:

Closed-ended		Open-ended		File upload	
Easy	3	Easy	0	Easy	0
Medium	0	Medium	0	Medium	0
Hard	1	Hard	0	Hard	0

รูปที่ 4.17: แสดงหน้าดูรายละเอียดข้อสอบที่เป็นข้อสอบสาธารณะเว็บแอพพลิเคชัน

ในหน้าดูข้อสอบที่เป็นข้อสอบสาธารณะประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- สามารถดูข้อสอบสาธารณะทั้งหมดได้

- สามารถค้นหาข้อสอบที่เป็นสาธารณะได้
- สามารถคัดลอกข้อสอบสาธารณะ เพื่อนำมาใช้ หรือแก้ไขได้
- สามารถดูรายละเอียดของข้อสอบสาธารณะได้

4.1.15 หน้าคุประวัติการส่งข้อสอบ

Name	Title	To	Created at
Fundamentals Exam	exam for everyone	Satwat Thitisupakul	Nov 3, 2020
Fundamentals Exam	exam for everyone	Satwat Thitisupakul	Nov 3, 2020

รูปที่ 4.18: แสดงหน้าคุประวัติการส่งข้อสอบของเว็บแอพพลิเคชัน

ในหน้าสร้างข้อสอบเพื่อใช้สำหรับผู้สมัครงานประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- สามารถดูรายละเอียดของการส่งข้อสอบได้
- สามารถค้นหาประวัติการส่งข้อสอบได้

4.1.16 หน้าดูข้อสอบที่ผู้สมัครงานทำเสร็จแล้ว

The screenshot shows a user interface for managing exams. On the left, there's a sidebar with various options: Create category, My category, Public category, Create exam, My exam, Public exam, Sent exam, and Finished exam (which is highlighted). The main area is titled "Finished exam" and displays a single row of results:

Exam name	Title	Name	Finished at	Status	Score
Fundamentals Exam	exam for everyone	Satwat Thitisupakul	Nov 3, 2020	verify	Correct 36% [progress bar]

Below the table is a navigation bar with icons for back, forward, and search.

รูปที่ 4.19: แสดงหน้าดูข้อสอบที่ผู้สมัครงานทำเสร็จแล้วของเว็บแอปพลิเคชัน

ในหน้าดูข้อสอบที่ผู้สมัครงานทำเสร็จแล้วประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- สามารถค้นหารายชื่อสอบที่ผู้สมัครงานทำเสร็จแล้วได้
- สามารถออกสถานะปัจจุบันของข้อสอบได้ว่าถูก ตรวจโดยพนักงานหรือยัง
- สามารถบอกรายละเอียดของผู้สมัครแต่ละคนได้
- สามารถลบได้

4.1.17 หน้าตรวจสอบข้อสอบของผู้สมัครงาน

Satawat Thitisupakul

Fundamentals Exam
exam for everyone

Email s.nack1919@gmail.com Name Satawat Thitisupakul Correct 36%

1. ก้ามันซึ่มูล ตัวนี้ 11 2 3 - 6 - 6 4 และค่าเฉลี่ยเลขคณิตคือค่า
เบ็ด一根只选一个 (choose only one answer)
from Math105 (1) easy closeEnd ✓
No. 3
○ 3/4
● 4/3 (correct answer)
○ 3
○ 4

2. ก้ามันซึ่มูล ตัวนี้ - 8 7 3 2 4 10 และค่าเฉลี่ยเลขคณิตคือค่า
เบ็ด一根只选一个 (choose only one answer)
from Math105 (1) easy closeEnd ✓
No. 2
○ 2
● 3 (correct answer)
○ 5
○ 6

3. ส่วนสูง (เซนติเมตร) ของพนักงานก่อสร้างเป็นตัวที่ 151 149
165 168 154 แล้วมีข้อมูลของส่วนสูงของพนักงานก่อสร้างนี้
เบ็ด一根只选一个 (choose only one answer)
from Math105 (1) medium
closeEnd ✗
No. 3
○ 154 เซนติเมตร (correct answer)
● 157 เซนติเมตร
○ 165 เซนติเมตร

รูปที่ 4.20: แสดงหน้าตรวจสอบข้อสอบของผู้สมัครงานของเว็บแอปพลิเคชัน

Satawat Thitisupakul

Exama

○ attain
○ observe
○ interfere (correct answer)

11. ตารางแสดงอายุ (ปี) ของผู้บุคคลในเข้าร่วมใหญ่องร โรง
พยาบาลเพื่อเงินเดือนอยู่ต้องของผู้บุคคลเป็นเข้าร่วมใหญ่อง
โรงพยาบาลเพื่อเบ็ด一根只选一个 (choose only one answer)
from Math105 (1) easy openEnd
No. 1

อายุ (ปี)	จำนวนผู้บุคคล
6 – 10	5
11 – 15	2
16 – 20	1
21 – 25	5
26 – 30	2

Corrected answer :
17.0 ✗

Answer :
14 ✗ ○ Correct
● Incorrect

Verify Cancel

รูปที่ 4.21: แสดงหน้าตรวจสอบข้อสอบอัตนัยของผู้สมัครงานของเว็บแอปพลิเคชัน

ในหน้าตรวจสอบข้อสอบของผู้สมัครงานประกอบด้วยฟังก์ชันการทำงานต่างๆ ดังนี้

- ระบบทำการตรวจสอบข้อสอบที่เป็นปรนัยแล้วทำการสรุปคะแนนให้โดยอัตโนมัติ
- ระบบทำการแสดงคำตอบที่ถูกต้องของคำ답ตามแต่ละข้อ

- ระบบสามารถบอกได้ว่าคำตามแต่ละข้อ ถูกสุ่มนจากหมวดหมู่ไหนและข้อที่เท่าไหร่
- ระบบแสดงช่องให้ผู้ใช้ทำการตรวจสอบข้อสอบที่เป็นอัตนัย และข้อสอบที่มีการอัปโหลดไฟล์
- สามารถยืนยัน หรือยกเลิกการตรวจได้

บทที่ 5

สรุปผล

ตลอดระยะเวลาการที่ปฏิบัติงานสหกิจศึกษา ณ บริษัท ไซเจ็น จำกัด นักศึกษาได้รับหน้าที่ให้พัฒนาเว็บแอ��พพลิเคชันประเมินความสามารถเบื้องต้นของผู้สมัครงาน โดยเว็บแอ��พพลิเคชันนี้ได้ถูกพัฒนาด้วยการเน้นส่วนที่จัดการกับฐานข้อมูล(back-end) เป็นหลัก โดยในส่วนนี้นักศึกษาสามารถทำสำเร็จลุล่วงไปด้วยดี สามารถใช้งานได้ทุกฟังก์ชันตามที่ได้ทำการออกแบบไว้ และ ส่วนต่อประสานกับผู้ใช้(front-end) ณ ตอนนี้ เป็นการออกแบบมาเพื่อให้ระบบใช้งานได้โดยที่ไม่เน้นเรื่องความสวยงามเป็นหลัก เนื่องจากมีระยะเวลาที่จำกัด ในส่วนนี้จึงได้ส่งมอบให้พี่ในทีม ได้นำเว็บแอ��พพลิเคชันไปพัฒนาส่วนต่อประสานกับผู้ใช้ต่อไป เพื่อให้มีการใช้งานที่สะดวกและสวยงามมากยิ่งขึ้น จากการที่ได้พัฒนาระบบ ทำให้นักศึกษาได้เรียนรู้ กระบวนการทำงานในชีวิตจริง ได้รู้วิธีการจัดสรรเวลาและวางแผนการทำงานต่างๆ ให้มีประสิทธิภาพ รวมถึงทักษะในการสื่อสาร เพราะเป็นการทำงานร่วมกับผู้คนหลายชั่วโมง ที่สุดท้ายนี้นักศึกษาได้เห็นว่าการเข้าร่วมสหกิจศึกษา ณ บริษัท ไซเจ็น จำกัด ทำให้นักศึกษาได้รับประสบการณ์ในการทำงาน ที่ไม่สามารถหาได้จากห้องเรียน ซึ่งนักศึกษาจะนำความรู้ความสามารถที่ได้รับนี้ไปปรับใช้เพื่อให้เกิดประโยชน์ต่อตัวนักศึกษาเอง และผู้อื่นด้วยเช่นกัน

5.1 ประโยชน์ที่ได้รับจากการปฏิบัติงาน

5.1.1 ประโยชน์ต่อตนเอง

จากการที่ได้เข้าร่วมสหกิจศึกษาเป็นระยะเวลา 4 เดือน ทำให้นักศึกษาได้รับความรู้ความสามารถ เสมือนการเรียนรู้นอกห้องเรียน

5.1.2 ประโยชน์ต่อสถานประกอบการ

5.1.3 ประโยชน์ต่อมหาวิทยาลัย

5.2 วิเคราะห์จุดเด่น จุดด้อย โอกาส อุปสรรค(Swot Analysis)

5.2.1 จุดเด่น

5.2.2 จุดด้อย

5.2.3 โอกาส

5.2.4 อุปสรรค

5.3 ประสบการณ์ที่ประทับใจ

បរចាំនាក្រម

- [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] T. K. Authors, “Kubernetes.” [Online]. Available: <https://kubernetes.io/>
- [10] Auth0, “Introduction to json web tokens.” [Online]. Available: <https://jwt.io/introduction>
- [11] I. MongoDB, “mongodb.” [Online]. Available: <https://www.mongodb.com>

ภาคผนวก ก
เรื่องที่หนึ่ง