



PROJECT ACTIWIZ  
APPLICATION FOR RECOMMENDED EVENT AND CLUB FOR KMUTT STUDENT

MR. KUNANON SUPMAMUL 63070501011

MR. NAPHATTARAK MUENTOEY 63070501018

MR. THANADOL THONGRIT 63070501029

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR  
THE DEGREE OF BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)  
FACULTY OF ENGINEERING  
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI  
2023

Project Activiz  
Application for recommended event and club for KMUTT student

Mr. Kunanon Supmamul 63070501011

Mr. Naphattarak Muentoey 63070501018

Mr. Thanadol Thongrit 63070501029

A Project Submitted in Partial Fulfillment  
of the Requirements for  
the Degree of Bachelor of Engineering (Computer Engineering)  
Faculty of Engineering  
King Mongkut's University of Technology Thonburi  
2023

Project Committee

.....  
(Asst.Prof. Rajchawit Sarochawikasit)

Project Advisor

.....  
(Dr. Kittipong Piyawanno)

Project Co-Advisor

.....  
(Asst.Prof.Dr. Nuttanart Muansuwan)

Committee Member

.....  
(Asst.Prof.Dr. Naruemon Wattanapongsakorn)

Committee Member

## กิตติกรรมประกาศ

โครงการนวัตกรรมคอมพิวเตอร์ฉบับนี้ สำเร็จลุล่วงได้อย่างสมบูรณ์ด้วยความกรุณาอย่างยิ่งจาก อาจารย์รำชวิชช์ สิริชวิกสิต และ อาจารย์กิตติพงษ์ ปิยะวรรณ์โนน ที่ได้สละเวลาอันมีค่าแก่คณะผู้จัดทำ เพื่อให้คำปรึกษาและแนะนำตลอดจนตรวจทานแก้ไขข้อบกพร่องต่างๆ ด้วยความเอาใจใส่เป็นอย่างยิ่ง จนโครงการนวัตกรรมคอมพิวเตอร์ฉบับนี้สำเร็จสมบูรณ์ลุล่วงได้ด้วยดี คณะผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี่จากใจจริง  
สุดท้ายนี้ ขออุทิศความดีที่มีในการศึกษา นี้แด่บิดา มารดา ครอบครัวของคณะผู้จัดทำและกำลังใจจากมิตรแท้ทุกท่าน

## สารบัญ

หน้า

กิตติกรรมประกาศ	iii
สารบัญ	iv
สารบัญตาราง	vii
สารบัญรูปภาพ	viii
<b>บทที่ 1 บทนำ</b>	<b>1</b>
1.1 คำสำคัญ	1
1.2 ที่มาและความสำคัญ	1
1.3 ประเภทของโครงงาน	1
1.4 วัตถุประสงค์	1
1.5 ตารางการดำเนินงาน	2
1.6 ขอบเขตของโครงงาน	3
1.7 ประโยชน์ที่คาดว่าจะได้รับ	3
<b>บทที่ 2 ทฤษฎีความรู้และงานที่เกี่ยวข้อง</b>	<b>4</b>
2.1 ทฤษฎีและความรู้ที่เกี่ยวข้อง	4
2.1.1 การเรียนรู้ของเครื่อง	4
2.1.1.1 Content-Based Filtering	4
2.1.1.2 Collaborative Filtering	4
2.1.1.3 การประมวลผลภาษาธรรมชาติ	4
2.1.1.4 Word Embedding	4
2.1.1.5 Transformer Model	5
2.1.1.6 โครงข่ายประสาทเทียม	5
2.1.2 สถาปัตยกรรมของแอปพลิเคชันในโทรศัพท์ [1]	6
2.1.3 HTTP Protocol	8
2.1.3.1 HTTP Protocol คืออะไร	8
2.1.3.2 HTTP Request-Response Cycle	9
2.1.3.3 HTTP Request Methods	9
2.1.4 REST API	10
2.1.4.1 REST คืออะไร	10
2.1.4.2 การออกแบบ REST API [2]	10
2.1.5 Graph Database	13
2.1.5.1 Graph Database คืออะไร	13
2.1.5.2 ทำไมต้องเป็น Graph Database	13
2.1.5.3 Property ของ Graph Model [3]	13
2.1.5.4 Graph Database vs Relational Database [4]	14
2.1.6 แผนภาพ UML	15
2.1.6.1 UML Diagram คืออะไร	15
2.1.6.2 Use case Diagram	15
2.1.6.3 Sequence Diagram	15
2.1.7 Principle Design	15
2.1.7.1 Principle Design คืออะไร	15
2.1.7.2 C.R.A.P. Principle Design	16

2.2 เทคโนโลยี	18
2.2.1 Integrated Development Environment (IDE)	18
2.2.2 Design	18
2.2.3 Frontend	18
2.2.4 Backend	18
2.2.5 Database	18
2.2.6 Machine Learning	19
2.2.7 Version Control	19
2.2.8 Testing	19
2.3 แบบสำรวจผลิตภัณฑ์	20
 บทที่ 3 วิธีการทำงาน กระบวนการและการออกแบบ	26
3.1 บทนำ	26
3.1.1 สำรวจความต้องการของผู้ใช้งานคุณภาพ	26
3.1.2 เส้นทางของผู้ใช้งาน (Journey Map)	26
3.1.3 ผู้ได้รับผลกระทบ (Stakeholder)	26
3.2 ความต้องการของผู้ใช้งาน (Requirement List)	27
3.3 รายการคุณลักษณะ (Feature List)	27
3.4 แผนภาพสถาปัตยกรรมของระบบ (System Architecture Diagram)	28
3.5 แผนภาพที่ใช้แสดงปฏิสัมพันธ์ระหว่างระบบงานและสิ่งที่อยู่นอกระบบงาน (Use Case Diagram)	29
3.6 Sequence diagram	33
3.6.1 การเข้าสู่ระบบ	33
3.6.2 การสมัครเข้าใช้งาน	34
3.6.3 การออกจากระบบ	35
3.6.4 การค้นหากิจกรรม	36
3.6.5 การอ่านรายละเอียดของกิจกรรมและเข้าร่วมกิจกรรม	37
3.6.6 การประเมินกิจกรรม	38
3.6.7 การแจ้งเตือนกิจกรรมชุมชน	39
3.6.8 การค้นหาชุมชน	40
3.6.9 อ่านรายละเอียดของชุมชนและการเข้าร่วมชุมชน	41
3.6.10 การถอดอกจากชุมชน	42
3.6.11 การแนะนำกิจกรรมให้ผู้ใช้งาน	43
3.6.12 การ Train Model การเรียนรู้ของเครื่อง	44
3.7 แบบจำลองข้อมูลแบบกราฟ (Graph Data Model)	45
3.7.1 โหนด	45
3.7.1.1 โหนด User: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดส่วนตัวของผู้ใช้งานภายในมหาวิทยาลัย	45
3.7.1.2 โหนด Department: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดของภาควิชาภายในมหาวิทยาลัย	46
3.7.1.3 โหนด Club: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดของชุมชนภายในมหาวิทยาลัย	46
3.7.1.4 โหนด Club Category: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดประเภทชุมชนภายในมหาวิทยาลัย	46
3.7.1.5 โหนด Activity: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดของชุมชนภายในมหาวิทยาลัย	47
3.7.1.6 โหนด Activity Category: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับประเภทของกิจกรรมภายในมหาวิทยาลัย	47
3.7.2 ความสัมพันธ์	
ความสัมพันธ์ภายในแบบจำลองข้อมูลแบบกราฟของแอพพลิเคชัน Actiwiz เป็นการสร้างความเชื่อมโยงและการมีปฏิสัมพันธ์ระหว่างโหนดต่าง ๆ ซึ่งมีส่วนช่วยในการนำเสนอเรื่องราวการเข้าร่วมกิจกรรมและชุมชนภายในมหาวิทยาลัยแบบบีดที่ญี่ปุ่นและเชื่อมโยงถึงกัน โดยความสัมพันธ์จะมีอยู่ 2 รูปแบบด้วยกัน ได้แก่ความสัมพันธ์ที่มีคุณสมบัติที่บอกข้อมูลเพิ่มเติมและความสัมพันธ์ที่ไม่มีคุณสมบัติเพิ่มเติม	47
3.7.2.1 ความสัมพันธ์ที่มีคุณสมบัติ	47
3.7.2.2 ความสัมพันธ์ที่ไม่มีคุณสมบัติ	48

<b>บทที่ 4 ผลการทดลองและอภิปรายผล</b>	<b>49</b>
4.1 การทดสอบการทำงาน (Functional Testing)	49
4.2 การทดสอบคุณภาพของระบบ (Non Functional Testing)	49
4.3 ผลการทดสอบ (Test Results)	49
<b>บทที่ 5 สรุปผลการดำเนินงาน</b>	<b>50</b>
5.1 กระบวนการทำงาน (Work Process)	50
5.1.1 การดำเนินงานในภาคการศึกษาที่ 1	50
5.1.2 การดำเนินงานในภาคการศึกษาที่ 2	50
5.2 ปัญหาและการแก้ไข (Problems and Solutions)	50
5.3 แนวทางการพัฒนา (Future Work)	50
5.4 บทสรุป (Conclusion)	50
<b>หนังสืออ้างอิง</b>	<b>51</b>

## สารบัญตาราง

ตารางที่	หน้า
2.1 HTTP Request Methods	9
2.2 Graph Database vs Relational Database	14
3.1 การสมัครเข้าใช้งาน	30
3.2 การเข้าสู่ระบบ	30
3.3 การออกจากระบบ	30
3.4 การค้นหากิจกรรม	31
3.5 การอ่านรายละเอียดกิจกรรม	31
3.6 การเข้าร่วมกิจกรรม	31
3.7 การประเมินกิจกรรม	31
3.8 การค้นหาชุมชน	31
3.9 การอ่านรายละเอียดชุมชน	32
3.10 การสมัครชุมชน	32
3.11 การลาออกจากชุมชน	32
3.12 ตารางเก็บข้อมูลของผู้ใช้งานที่ใช้งานแอปพลิเคชัน	45
3.13 ตารางเก็บข้อมูลของภาควิชาภายในมหาวิทยาลัย	46
3.14 ตารางเก็บข้อมูลของชุมชนภายในมหาวิทยาลัย	46
3.15 ตารางเก็บข้อมูลประเภทชุมชน	46
3.16 ตารางเก็บข้อมูลของชุมชนภายในมหาวิทยาลัย	47
3.17 ตารางเก็บข้อมูลประเภทของกิจกรรม	47
3.18 ตารางเก็บข้อมูลนักศึกษาภาควิชา	47
3.19 ตารางเก็บข้อมูลสมาชิกของชุมชน	47
3.20 ตารางเก็บข้อมูลสมาชิกเข้าร่วมกิจกรรม	48
3.21 ตารางเก็บข้อมูลกิจกรรมถูกจัดโดยชุมชน	48
4.1 ตารางการทดสอบการทำงานทดสอบการทำงาน	49
5.1 ตารางแสดงผลการดำเนินงานในภาคการศึกษาที่ 1	50
5.2 ตารางแสดงผลการดำเนินงานในภาคการศึกษาที่ 2	50

## สารบัญ

หัวที่	หน้า
1.1 ภาคการศึกษาที่ 1	2
1.2 ภาคการศึกษาที่ 2	2
2.1 Mobile App Architecture [5]	6
2.2 Presentation layer [6]	7
2.3 Business layer [7]	8
2.4 HTTP Request-Response Cycle [8]	9
2.5 tab เลือกหน้าของ Google	11
2.6 Handle Error1	12
2.7 Handle Error2	13
2.8 Property ของ Graph Model	14
2.9 contrast [9]	16
2.10 repetition [10]	16
2.11 alignment [11]	17
2.12 proximity [12]	17
2.13 หน้าแนะนำกิจกรรมของ ModLink และรายละเอียด	20
2.14 หน้าแนะนำกิจกรรมของ KMUTT Hatch	21
2.15 หน้า Mainpage ของ KMUTT Sinfo	21
2.16 หน้า Mainpage ของ Padlet	22
2.17 หน้าแนะนำกระซู่ของ Pantip	22
2.18 หน้า Homepage ของเพจชุมชนใน Facebook	23
2.19 หน้า Notifications ของ Instagram	23
2.20 ตารางความแตกต่างของ Feature	24
2.21 ระบบการแนะนำกิจกรรม	25
3.1 Architecture diagram	28
3.2 Use Case Diagram	29
3.3 Sequence diagram ของการเข้าสู่ระบบ	33
3.4 การสมัครเข้าใช้งาน	34
3.5 การออกจากระบบ	35
3.6 การค้นหา กิจกรรม	36
3.7 การอ่านรายละเอียดของกิจกรรมและเข้าร่วมกิจกรรม	37
3.8 การประเมินกิจกรรม	38
3.9 การแจ้งเตือนกิจกรรมชุมชน	39
3.10 การค้นหาชุมชน	40
3.11 อ่านรายละเอียดของชุมชนและการเข้าร่วมชุมชน	41
3.12 การลากອอกจากชุมชน	42
3.13 การแนะนำกิจกรรมให้ผู้ใช้งาน	43
3.14 การ Train Model การเรียนรู้ของเครื่อง	44
3.15 Graph Data Model	45

# บทที่ 1 บทนำ

## 1.1 คำสำคัญ

Native Mobile Application, Machine Learning, Recommendation System

## 1.2 ที่มาและความสำคัญ

เนื่องจากคณะผู้ศึกษาดูร่องรอยปัญหาที่เกี่ยวข้องกับการกระจายข่าวสารเกี่ยวกับกิจกรรมต่าง ๆ ภายในมหาวิทยาลัยที่มีการประชาสัมพันธ์ข้อมูลที่กระจัดกระจาดและมีจำนวนมาก ทำให้การที่นักศึกษาทุกคนสามารถทราบข่าวสารได้อย่างเท่าเทียมกันเป็นไปได้ยาก และนักศึกษาอาจพลาดข้อมูลเกี่ยวกับกิจกรรมที่สนใจเนื่องจากปัญหานี้ ด้วยความตั้งใจที่จะแก้ไขปัญหานี้ นักศึกษาได้มีแนวคิดในการพัฒนาแอปพลิเคชันเพื่อช่วยในการกระจายข่าวสารและกิจกรรมต่าง ๆ ที่เกิดขึ้นในมหาวิทยาลัย โดยใช้เทคโนโลยี Machine Learning เพื่อสนับสนุนในการจัดการข้อมูลเหล่านี้ แอปพลิเคชันจะมีหน้าที่ในการแนะนำกิจกรรม และชุมชนต่าง ๆ ให้แก่นักศึกษา โดยให้คำแนะนำที่เป็นไปตามความสนใจของแต่ละบุคคล เพื่อให้ทุกคนสามารถมีโอกาสเข้าถึงข้อมูลเกี่ยวกับกิจกรรมที่ตรงกับความสนใจส่วนตัวของตนได้อย่างง่ายดาย

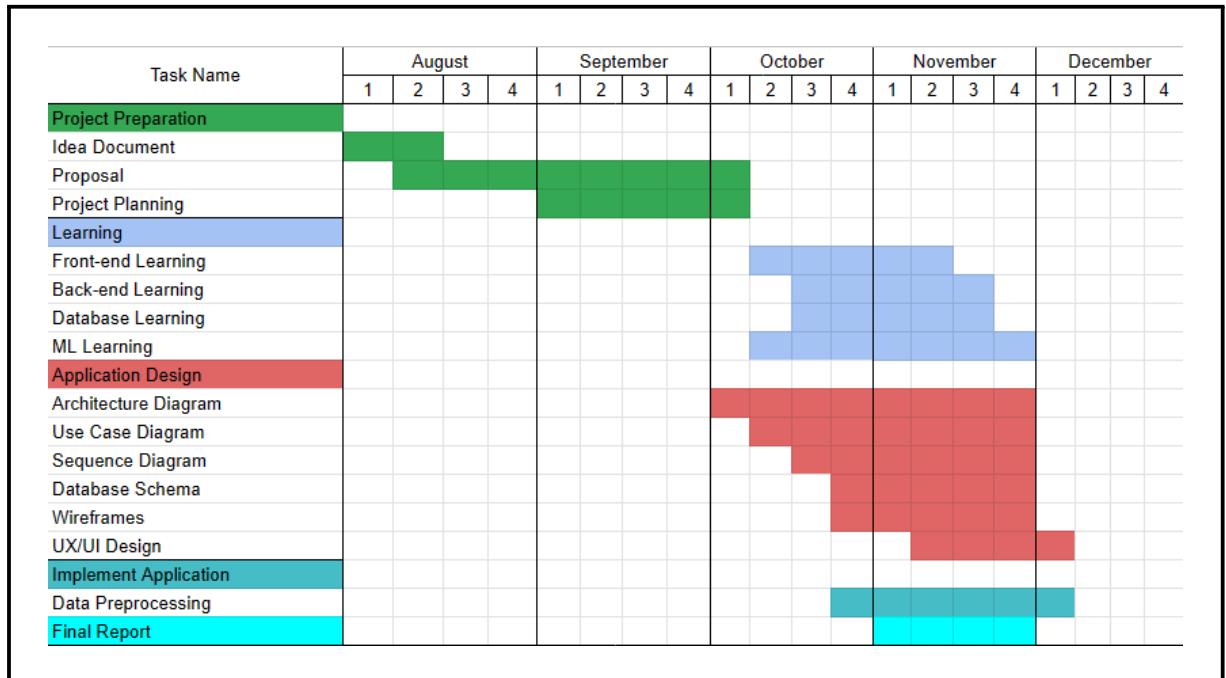
## 1.3 ประเภทของโครงการ

โครงการที่เป็นการประดิษฐ์ คิดค้น

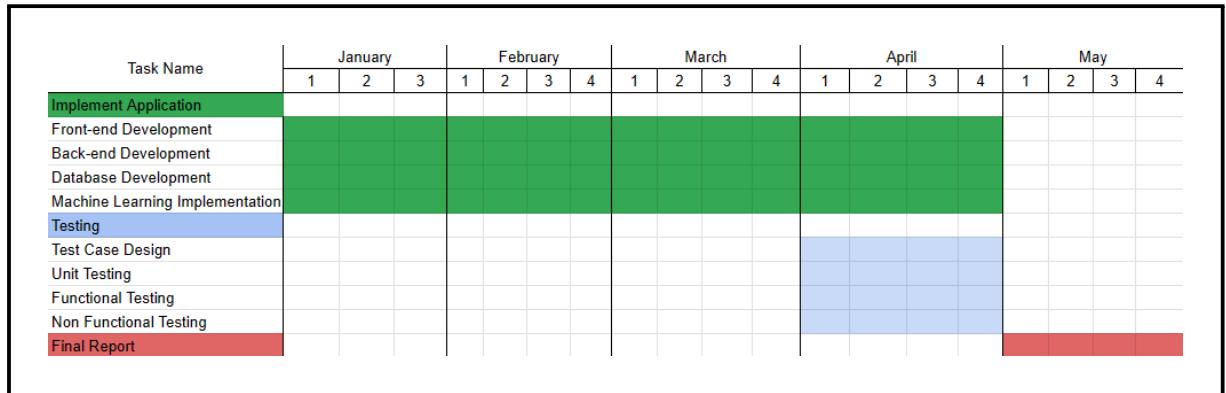
## 1.4 วัตถุประสงค์

- เพื่อพัฒนาแอปพลิเคชันสำหรับการกระจายข้อมูลข่าวสารและกิจกรรมต่าง ๆ ภายในมหาวิทยาลัย เพื่อให้ง่ายต่อการติดตามข่าวไว้แก่นักศึกษาภายในมหาวิทยาลัย
- เพื่อศึกษาพัฒนาและประเมินผลกิจกรรมของนักศึกษา เพื่อที่จะแนะนำกิจกรรมและชุมชนที่นักศึกษามีแนวโน้มให้ความสนใจ
- แอปพลิเคชันที่อำนวยความสะดวกต่อนักศึกษาในการเข้าร่วม กิจกรรมต่างๆ
- สามารถแนะนำแนวทางการจัดกิจกรรมที่มีนักศึกษาภายในมหาวิทยาลัยให้ความสนใจ ไปเสนอแก่ทางมหาวิทยาลัย เพื่อเพิ่มประสิทธิภาพในการจัดกิจกรรมต่อ ๆ ไป ที่จะเกิดขึ้นในอนาคต โดยวิเคราะห์จากเนื้อหา กิจกรรมที่ทางนักศึกษาให้ความสนใจ

### 1.5 ตารางการดำเนินงาน



รูปที่ 1.1: ภาคการศึกษาที่ 1



รูปที่ 1.2: ภาคการศึกษาที่ 2

## 1.6 ขอบเขตของโครงการ

แอปพลิเคชันสำหรับการแนะนำกิจกรรม และชุมชนให้แก่นักศึกษา โดยมีขีดความสามารถดังต่อไปนี้

1. ระบบ Log in ผ่านอีเมลของมหาวิทยาลัย
2. ระบบรวบรวมข้อมูลของชุมชนต่าง ๆ เอาไว้ โดยผู้ใช้สามารถค้นหาและติดตามข้อมูลของชุมชนที่ตนเองสนใจได้
3. ระบบแจ้งเตือนกิจกรรมที่เกี่ยวข้องกับชุมชนหรือความสนใจของนักศึกษา
4. ระบบแจ้งเตือนการประเมินผลกิจกรรม เมื่อฟอร์มการประเมินพร้อมใช้งาน
5. ระบบแนะนำกิจกรรมและชุมชน ตามความสนใจของผู้ใช้โดยอ้างอิงจาก tag ของกิจกรรม
6. ระบบวิเคราะห์ความสนใจของผู้ใช้ผ่านเนื้อหาของกิจกรรมที่ผู้ใช้เคยเข้าร่วม เข้าไปอ่านรายละเอียด หรือเกี่ยวข้องกับชุมชนที่สนใจ
7. ระบบแยกประเภทกิจกรรมอัตโนมัติโดยวิเคราะห์จากเนื้อหา ออกแบบเป็น tag ต่าง ๆ โดยใช้ Machine Learning

## 1.7 ประโยชน์ที่คาดว่าจะได้รับ

1. แอปพลิเคชันที่สนับสนุนการเข้าร่วมกิจกรรมของนักศึกษา และสามารถแนะนำกิจกรรมชุมชนที่นักศึกษาน่าจะสนใจได้
2. การจำแนกประเภทของกิจกรรมของนักศึกษา

## บทที่ 2 ทฤษฎีความรู้และงานที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงรายละเอียดของทฤษฎี ความรู้ และเทคโนโลยีที่นำมาใช้ในการสร้างและพัฒนาแอปพลิเคชัน Actiwiz โดยจะอธิบาย ความสามารถและการหยิบมาใช้งานในโครงการโดยละเอียด อีกทั้งยังมีการวิเคราะห์ผลิตภัณฑ์ในประเภทเดียวกันที่มีการใช้งานอยู่เพื่อนำ มาปรับใช้กับตัวโครงการอีกด้วย

### 2.1 ทฤษฎีและความรู้ที่เกี่ยวข้อง

#### 2.1.1 การเรียนรู้ของเครื่อง

##### 2.1.1.1 Content-Based Filtering

Content-Based Filtering [13] เป็นหนึ่งในแนวทางที่ได้รับความนิยมอย่างมากในระบบการแนะนำ เนื่องจากความสามารถในการให้ คำแนะนำที่มีความเป็นบุคคลและเกี่ยวข้องกับความสนใจของผู้ใช้งาน กระบวนการนี้เน้นการวิเคราะห์เนื้อหาที่ผู้ใช้งานมีความสนใจและ แนะนำสิ่งที่มีเนื้อหาที่คล้ายคลึงกันให้กับผู้ใช้งาน ตัวอย่างเช่น การแนะนำหนังสือที่มีเนื้อหาใกล้เคียงกับที่ผู้ใช้งานเคยซื้อมาก่อนหน้านี้ ดังนั้นเทคนิคนี้จึงเป็นการสร้างคำแนะนำที่เน้นความเป็นบุคคลและความสอดคล้องกับความต้องการของผู้ใช้งานอย่างมีความแม่นยำและ เหมาะสม อย่างไรก็ตามการทำ Content-Based Filtering นั้น จะยึดตามความสนใจของผู้ใช้งานที่มีประวัติเก็บเอาไว้ทำให้สามารถแนะนำ ได้ในวงแคบ ๆ เท่านั้น

##### 2.1.1.2 Collaborative Filtering

Collaborative Filtering [14], [15] เป็นหนึ่งในแนวทางที่ได้รับความนิยมอย่างมากในการสร้างคำแนะนำสำหรับผู้ใช้งาน แนวทาง นี้เสนอหลักการแนะนำที่มาจากการพูดคุยหรือการใช้งานที่เกิดขึ้นก่อนหน้านี้ของผู้ใช้งาน การ Collaborative Filtering มีวิธีการหลักที่สำคัญ สองวิธีคือ

User-Based Collaborative Filtering: แนวทางนี้จะทำการแนะนำสิ่งที่ผู้ใช้งานมีความสนใจอยู่ หากมีความคล้ายคลึงในพฤติกรรมการใช้ งานระหว่างผู้ใช้งานสองคน เช่น ถ้าผู้ใช้งานสองคนมีรูปแบบพฤติกรรมที่คล้ายกัน ระบบจะแนะนำสิ่งที่ผู้ใช้อีกคนเคยใช้งานให้กับผู้ใช้งาน คนอื่น ซึ่งเพิ่มโอกาสที่จะให้คำแนะนำที่หลากหลายขึ้นได้ อย่างไรก็ตามก็อาจมีโอกาสที่จะแนะนำสิ่งที่ผู้ใช้งานไม่สนใจเลยได้เช่นกัน และยัง มีปัญหาความซับซ้อนเกิดขึ้นได้หากมีผู้ใช้งานหรือสิ่งที่จะแนะนำเป็นจำนวนมาก

Item-Based Collaborative Filtering: แนวทางนี้จะทำการแนะนำสิ่งที่ผู้ใช้งานมีความสนใจ และจะทำการแนะนำสิ่งที่มีเนื้อหาคล้ายกัน เมื่อพิจารณาการกระทำที่เกิดขึ้นก่อนหน้านี้ ระบบจะแนะนำสิ่งที่มีความคล้ายคลึงให้กับผู้ใช้งาน อย่างไรก็ได้เนื่องจากเป็นการแนะนำจาก สิ่งที่มีความใกล้เคียงกัน คำแนะนำที่ได้จึงอาจไม่ใช่สิ่งที่ผู้ใช้งานมองหา

##### 2.1.1.3 การประมวลผลภาษาธรรมชาติ

การประมวลผลภาษาธรรมชาติ (Natural Language Processing หรือ NLP) [16], [17] เป็นสาขานึงของปัญญาประดิษฐ์ ที่จะมุ่งเน้น การให้ความสามารถแก่คอมพิวเตอร์ที่เป็นการเข้าใจ ตีความ และปฏิสัมพันธ์กับภาษาของมนุษย์ แนวทางนี้นำเออัลกอริทึมและเทคนิคใน การประมวลผล วิเคราะห์ และสร้างข้อมูลข้อความและเสียง เพื่อให้เครื่องคอมพิวเตอร์สามารถสกัดความหมาย บริบท และความรู้จาก ภาษาที่มนุษย์สร้างขึ้นได้

##### 2.1.1.4 Word Embedding

Word Embedding [18], [19] ในการทำการประมวลผลภาษาธรรมชาติคือการที่แปลงคำต่าง ๆ ออกมายieldเป็นเวกเตอร์เพื่อที่จะตรวจ สอบความหมายของคำและความหมายในการสร้างประโยค ซึ่งเป็นพื้นฐานในการทำงานเดลการประมวลผลภาษาธรรมชาติซึ่งสามารถนำ มาประยุกต์ได้ทั้งการที่จะจัดประเภทกิจกรรมหรือการวิเคราะห์ความต้องการของผู้ใช้งานจากข้อความที่ถูกใช้

#### 2.1.1.5 Transformer Model

Transformer Model [20] คือสถาปัตยกรรมการเรียนรู้ของเครื่องสำหรับการประมวลผลและทำความเข้าใจภาษาของมนุษย์ ซึ่งทำงานได้ด้วยการแปลภาษา การสรุป และการสร้างข้อความ โดยอาศัยหลักการของ self-attention ในการทำความเข้าความสัมพันธ์ระหว่างคำในประโยคพร้อมกัน ทำให้สามารถเข้าใจบริบทและความหมายได้มากขึ้นอย่างมีประสิทธิภาพ โดย Transformer-base model ที่ถูกใช้งานในโครงงานนี้ มี 2 รูปแบบ คือ

1. mT5 (Text-to-Text Transfer Transformer) เป็นสถาปัตยกรรมที่ประยุกต์มาจาก unified text-to-text framework ซึ่งพัฒนาจาก NLP model หลากหลายงานให้ใช้งานง่ายขึ้น จึงสามารถนำมา train ต่อเพื่อทำงานเกี่ยวกับ Natural language processing ได้อย่างเน้นประสิทธิภาพ โดยที่มีผลกรอบต่อประสิทธิภาพในการทำงานเพียงเล็กน้อย อย่างไรก็ตาม T5 มีข้อจำกัดในการเข้าใจบริบทของข้อความและต้องอาศัยการปรับแต่งในการทำงานที่เฉพาะด้าน
2. BERT (Bidirectional Encoder Representations from Transformers) [21] สถาปัตยกรรมที่ถูก train โดยคลังข้อความขนาดใหญ่ จึงสามารถทำงานในการจับบริบทที่อยู่เบื้องหลังข้อความได้ เหมาะกับการทำงาน Natural language processing ที่หลากหลาย อย่างไรก็ตาม BERT ต้องการหน่วยประมวลผลในการ train และใช้งาน รวมไปถึงไม่สามารถปรับแต่งได้บนกิริยาเดียว

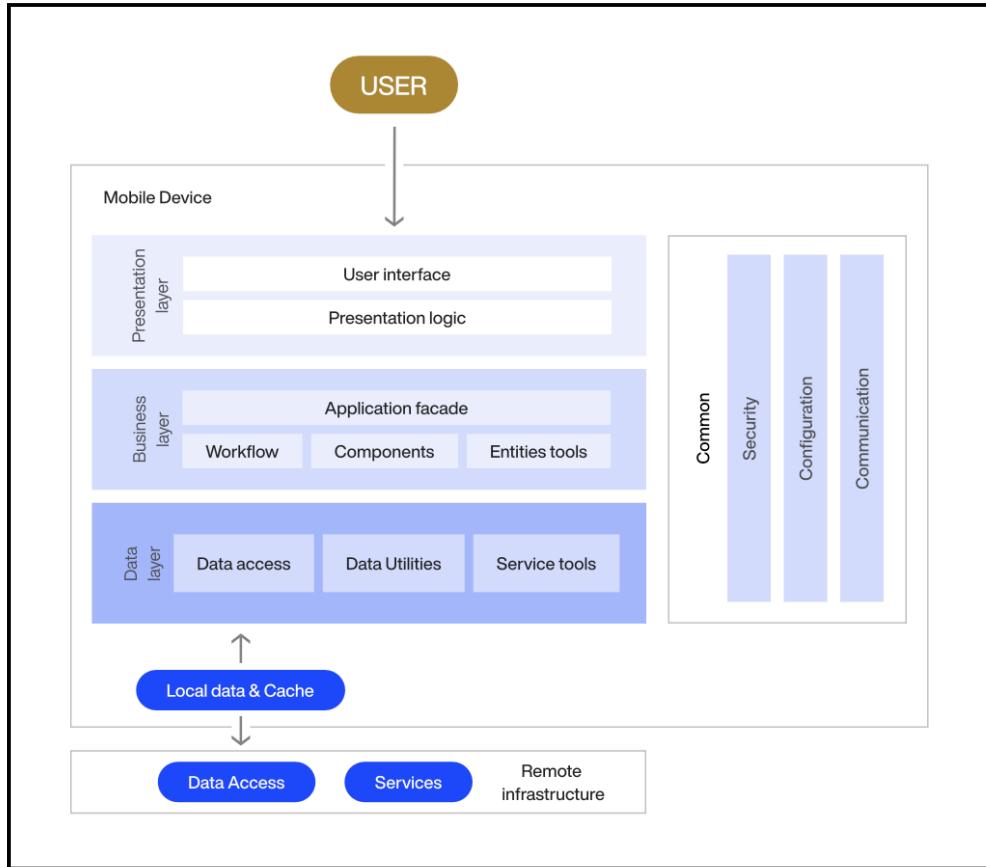
#### 2.1.1.6 โครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม (Artificial neural network) เป็นสถาปัตยกรรมการเรียนรู้ของเครื่องที่ได้รับแรงบันดาลใจจากระบบประสาทของมนุษย์ โดยมีหลักการคือการแบ่งการทำงานเป็น node ต่าง ๆ ที่รับค่าที่ประมวลผลมาจาก node ที่อยู่ใน input layer และส่งต่อให้ node ประมวลผลที่อยู่ใน hidden layer ไปเรื่อย ๆ จนถึง node ประมวลผลสุดท้ายใน output layer เพื่อตีความผลลัพธ์ ซึ่งสามารถทำงานได้หลากหลายรวมไปถึงงานการประมวลผลภาษาธรรมชาติซึ่งในโครงงานนี้มี โครงข่ายประสาทเทียม 2 รูปแบบ คือ

1. Shallow Neural Network ซึ่งเป็น Neural Network ที่มีจำนวน Layer ใน การประมวลผลอยู่น้อย มีข้อดีคือการที่จะใช้หน่วยประมวลผลน้อยและสามารถ train ได้ไว แต่ในทางกลับกันก็มีข้อจำกัดในการตีความที่ซับซ้อนรวมถึงบริบทที่อยู่ในข้อความของงาน Natural Language Processing
2. Regularizing and Optimizing LSTM Language Model [22] คือ Neural Network ที่ถูกพัฒนามาสำหรับการทำ Natural Language Processing ทำให้สามารถประมวลผลข้อมูลที่มีความต่อเนื่องอย่างเช่นประโยคได้ ตัวโมเดลมีความสามารถในการทำความเข้าใจบริบทในข้อความได้ โดยพิจารณาข้อมูลในหน่วยความจำของโมเดล ซึ่งสามารถจดจำหรือลับข้อมูลได้ตามความเหมาะสม และสามารถรับมือกับข้อความที่ไม่รู้จักได้ อย่างไรก็ตามคุณภาพของ model ขึ้นอยู่กับคุณภาพของข้อมูลที่ใช้ train และการปรับแต่งค่อนข้างส่งผลกระทบตัว model

### 2.1.2 สถาปัตยกรรมของแอปพลิเคชันในโทรศัพท์ [1]

ในการพัฒนาแอปพลิเคชัน สถาปัตยกรรมหมายถึงกฎ กระบวนการ และโครงสร้างภายในของแอปพลิเคชัน หรือก็คือวิธีการสร้างแอปพลิเคชัน โดยจะเป็นการกำหนดรูปแบบที่ส่วนประกอบต่าง ๆ สื่อสารกันเพื่อประมวลผลข้อมูล input จากผู้ใช้และประมวลผลข้อมูล output ให้กับผู้ใช้ โดยมีตัวอย่างสถาปัตยกรรมดังรูปที่ 2.1

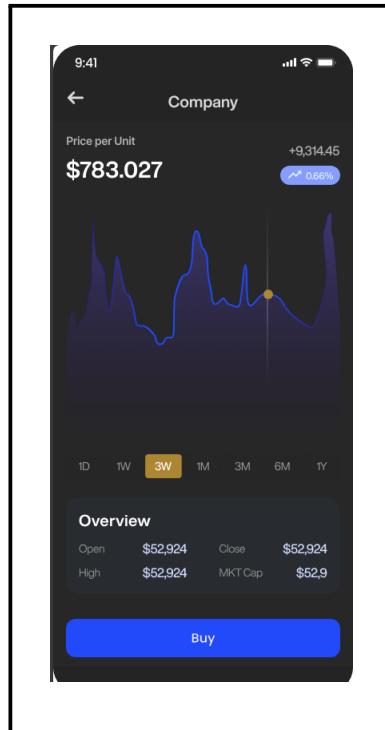


รูปที่ 2.1: Mobile App Architecture [5]

สถาปัตยกรรมของแอปพลิเคชันส่วนใหญ่จะประกอบด้วยสามเลเยอร์หลัก ๆ ได้แก่ Presentation layer, Business layer และ Data layer

#### 1. Presentation layer[1]

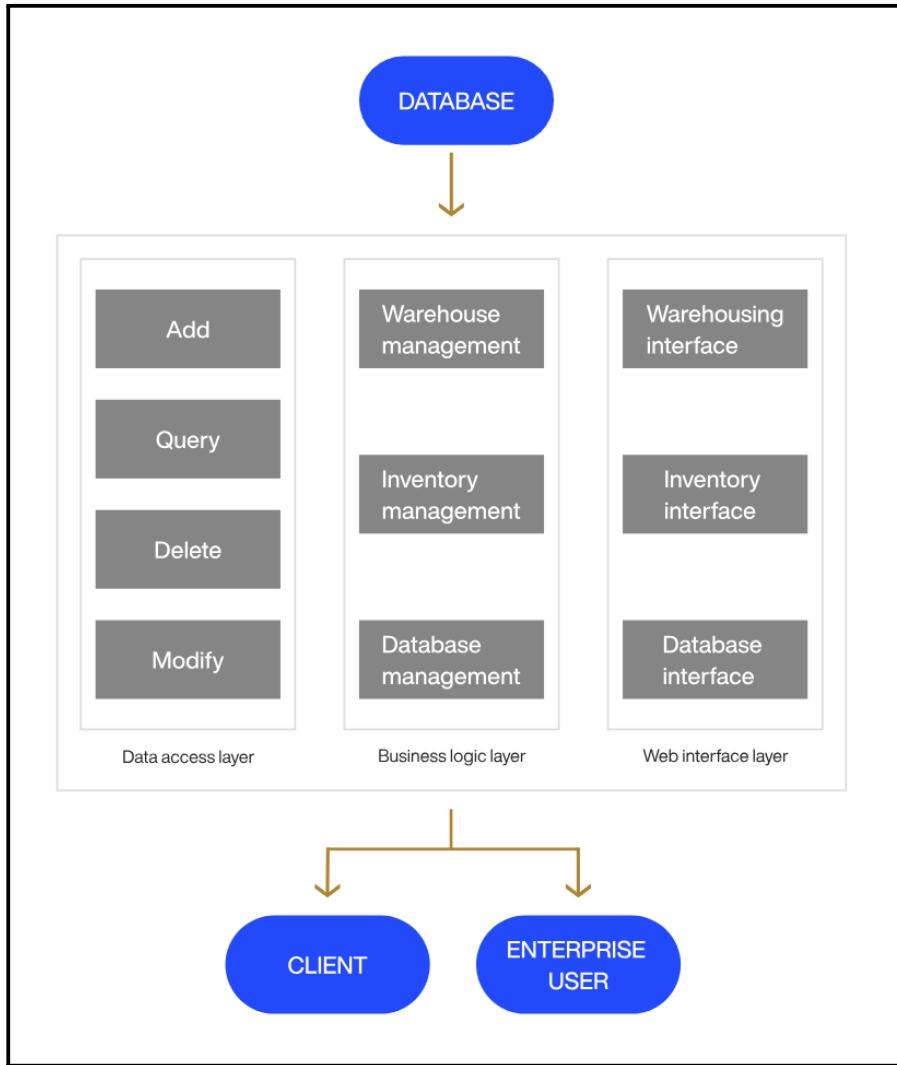
หรือก็คือ front end เป็นส่วนของแอปพลิเคชันที่ผู้ใช้มองเห็นและมีปฏิสัมพันธ์ได้ โดยมีส่วนติดต่อผู้ใช้ (user interface หรือ UI) ของแอปพลิเคชันเป็นส่วนสำคัญของเลเยอร์นี้ วัตถุประสงค์หลักของเลเยอร์นี้คือการนำข้อมูลที่ส่งมาจาก business layer มาแสดงผลในลักษณะที่ผู้ใช้สามารถเข้าใจได้ ไม่ว่าจะเป็น UI แบบพื้นฐาน เช่น UI แสดงที่อยู่อีเมลของผู้ใช้หรือ UI ที่ซับซ้อน เช่น แอปซื้อขายหุ้นซึ่งแสดงข้อมูลสดเกี่ยวกับตลาดหลักทรัพย์อุปกรณ์แสดงเป็นกราฟหรือแผนภูมิ แม้ว่าจะพัฒนาส่วนใหญ่จะรับผิดชอบ Business layer และ Data layer



รูปที่ 2.2: Presentation layer [6]

## 2. Business layer [1]

เลเยอร์นี้จะประกอบด้วยตระกูลของแอปพลิเคชัน หรือคือวิธีการทำงานของแอปพลิเคชัน โดยมักจะเป็นการนำข้อมูลที่ผู้ใช้ป้อนหรือข้อมูลจาก Data layer มาประมวลผล จากนั้นจึงส่งไปยัง presentation layer ส่วนใหญ่ business layer จะเป็นส่วนที่ซับซ้อนที่สุดในแอปพลิเคชัน โดยปกติแล้วจะแบ่งออกเป็น Layer อย่างๆหรือส่วนประกอบหลายส่วน โดยแต่ละส่วนมีหน้าที่รับผิดชอบในการทำงานเฉพาะ ตัวอย่างเช่น หากคุณมีแอปการจัดการทรัพยากรองค์กร (ERP) business layer อาจมีส่วนประกอบสำหรับการจัดการคลังสินค้าและระบบจัดการสินค้าคงคลัง



รูปที่ 2.3: Business layer [7]

### 3. Data layer

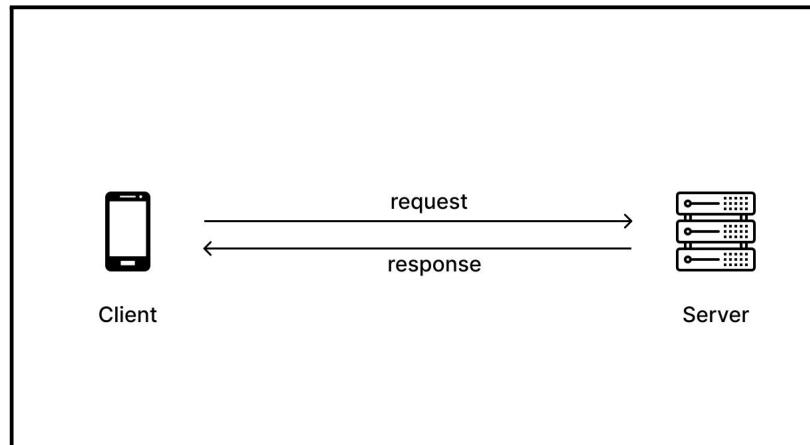
Data Layer [1] นี้เป็นตัวกลางระหว่าง Layer อื่นๆ กับทรัพยากรภายนอก วัตถุประสงค์หลักของ Layer นี้คือการรวบรวมข้อมูลจากแหล่งต่างๆ เช่น ฐานข้อมูล, เซิร์ฟเวอร์คลาวด์ หรือ API และส่งไปยัง Layer ด้านบน ด้วยวิธีการเชื่อมต่อ กับ Database และขอข้อมูลที่เกี่ยวข้องทั้งหมด เช่น ข้อมูลลูกค้า ข้อมูลสินค้า ข้อมูลขาย เป็นต้น ข้อมูลที่ได้มาจะถูกจัดรูปแบบให้เข้ากับมาตรฐานเดียวกัน แล้วส่งกลับไปยัง Layer ด้านบน ตัวอย่างเช่น เมื่อผู้ใช้งานต้องการดูรายละเอียดของสินค้า ข้อมูลจะถูกจัดรูปแบบเป็น JSON แล้วส่งไปยังหน้าจอแสดงผล หรือ API ที่เชื่อมต่ออยู่กับ Database แล้วส่งข้อมูลที่ต้องการไปยังหน้าจอแสดงผล จึงสามารถแสดงผลลัพธ์ที่ต้องการได้โดยตรง

#### 2.1.3 HTTP Protocol

##### 2.1.3.1 HTTP Protocol คืออะไร

HTTP [23] (Hypertext Transfer Protocol) เป็นโปรโตคอลที่ใช้ในการแลกเปลี่ยนข้อมูลผ่านทางอินเทอร์เน็ต เปรียบเสมือนระบบส่งข้อมูลบนอินเทอร์เน็ตที่ช่วยให้มั่นใจได้ว่าข้อมูลจะถูกส่งจากที่หนึ่งไปยังอีกที่หนึ่งได้

### 2.1.3.2 HTTP Request-Response Cycle



รูปที่ 2.4: HTTP Request-Response Cycle [8]

การสื่อสารในโปรโตคอล HTTP มีศูนย์กลางอยู่ที่แนวคิดที่เรียกว่า Request-Response Cycle เป็นกระบวนการที่client เตือน (Client) เช่น เว็บเบราว์เซอร์หรือแอปพลิเคชันมือถือ สื่อสารกับเซิร์ฟเวอร์ (Server) เพื่อขอทรัพยากรที่ต้องการหรือเพื่อดำเนินการบางอย่าง โดยจะประกอบด้วยหลักขั้นตอนได้แก่

1. โคลอนต์เริ่มต้นการส่งคำขอไปยังเซิร์ฟเวอร์โดยการส่งข้อความร้องขอ (HTTP request message) ซึ่งประกอบด้วยข้อมูลอาทิเช่น ทรัพยากรที่ต้องการและพารามิเตอร์เพิ่มเติมอื่น ๆ
2. เซิร์ฟเวอร์ได้รับข้อความร้องขอและประมวลผลโดยใช้ทรัพยากรที่มีอยู่เพื่อสร้างข้อความตอบกลับ (HTTP response message)
3. เซิร์ฟเวอร์ส่งข้อความตอบกลับไปยังโคลอนต์ ซึ่งโดยทั่วไปจะมีทรัพยากรที่ร้องขอ (เช่น หน้าเว็บ) และข้อมูลเพิ่มเติมหรือเมแทดาตา (ข้อมูลที่ใช้อธิบายชุดข้อมูลอื่นอีกที)
4. โคลอนต์ได้รับข้อความตอบกลับและประมวลผล โดยปกติจะเป็นการแสดงเนื้อหาในเว็บเบราว์เซอร์หรือในแอปพลิเคชัน
5. โคลอนต์อาจเริ่มการร้องขอเพิ่มเติมไปยังเซิร์ฟเวอร์ โดยทำซ้ำขั้นตอนเดิมแล้วแต่ความจำเป็น

### 2.1.3.3 HTTP Request Methods

Request method จะเป็นการบอกเซิร์ฟเวอร์ว่าลูกค้าต้องการให้เซิร์ฟเวอร์ดำเนินการอะไร Request method ที่พบเจอบ่อยจะมีดังนี้

HTTP METHODS	DEFINITION
HEAD	ถามเซิร์ฟเวอร์เกี่ยวกับสถานะ (ขนาด ความพร้อมใช้งาน) ของทรัพยากร
GET	ขอทรัพยากรจากเซิร์ฟเวอร์
POST	ขอให้เซิร์ฟเวอร์สร้างทรัพยากรใหม่
PUT	ขอให้เซิร์ฟเวอร์แก้ไข/อัปเดตทรัพยากรที่มีอยู่แล้ว
DELETE	ขอให้เซิร์ฟเวอร์ลบทรัพยากร

ตารางที่ 2.1 HTTP Request Methods

## 2.1.4 REST API

### 2.1.4.1 REST คืออะไร

REST [24] ย่อมาจาก Representational State Transfer เป็นรูปแบบการส่งข้อมูลระหว่าง Server-Client รูปแบบหนึ่งซึ่งอยู่บนพื้นฐานของ HTTP Protocol เป็นการสร้าง Web Service เพื่อแลกเปลี่ยนข้อมูลกันผ่าน Application วิธีหนึ่ง ซึ่งส่งข้อมูลได้หลายชนิด ไม่ว่าจะเป็น Text, XML, JSON หรือส่งมาเป็นหน้า HTML เลย REST ทำงานอยู่บน HTTP Protocol ทำให้เวลาใช้งานจะต้องอยู่บนพื้นฐาน HTTP Method เช่น GET, POST, PUT, DELETE จะใช้ Method ไหนก็ขึ้นอยู่กับว่าจะทำอะไรกับข้อมูล แต่ก็ต้องควรใช้คู่กับ Operation CURD เช่น เมื่อต้องการจะเรียกดูข้อมูลทั้งหมดก็ใช้ GET เมื่อต้องการเพิ่มข้อมูลก็ใช้ POST เป็นต้น

### 2.1.4.2 การออกแบบ REST API [2]

1. เลือกใช้ HTTP Method ให้เหมาะสมกับการใช้งาน ในกรณีปกติ การสร้าง URL จะไม่ใส่ชื่อกิริยาของ API มาอยู่ใน path เช่น /createUsers, /getUserDetail นั้นผิดหลักในการสร้าง เนื่องจากในการที่จะระบุว่าแต่ละ API จะถูกใช้ทำหน้าที่อะไรนั้นจะมี HTTP Method ในการระบุอยู่แล้ว
2. การสร้าง URL ของ API endpoint ให้ตรงตามมาตรฐานในการสร้าง URL ของ API นั้นมีทั้งหมดสามกฎที่สำคัญ ก็คือ
  - ควรจะมีแค่ชื่อ resource เท่านั้น เนื่องจาก resource เป็นตัวแทนของ สิ่งของบางอย่าง ที่เข้มโยงกับข้อมูล เช่น Users, Customers, Orders
  - ชื่อ path ควรจะเป็นรูปพหุจังหวะของ resource
  - ไม่ควรจะมีชื่อกิริยาที่บอกรถวัตถุประสงค์ของ API (เช่น add, update, delete) ตามที่กล่าวไว้ในข้อแรก

สมมติว่าต้องการ API ที่เกี่ยวข้องกับ Users โดยสามารถ สร้างข้อมูล user, แก้ไขข้อมูล user, และลบข้อมูล user สามารถเขียนออกมารูปดังนี้

- method: POST path: /users สร้างข้อมูล user ใหม่
- method: PUT path: /users/99 จะแก้ไขข้อมูล user ที่ id 99
- method: GET path: /users จะได้ข้อมูลของ user ทั้งหมด
- method: GET path: /users/99 จะได้รายละเอียดของ user ที่ id 99
- method: DELETE path: /users/99 จะเป็นการลบข้อมูล user ที่ id 99

กรณีที่ข้อมูลความเกี่ยวข้องกัน ส่วนใหญ่จะใช้เป็น Nested endpoint แทน query string เช่น ต้องการข้อมูลของ user ทั้งหมดที่อยู่ใน customer id เป็น 2 สามารถเขียนเป็น GET /customers/2/users แต่มีบางกรณีที่ ข้อมูล Nested ที่เยอะมากๆ จนหากใช้งานเป็น URL path จะมียาวมากเกินไป อาจจะใส่เป็น query string หรือ ใส่ใน body แทน เพื่อให้อ่านได้ง่ายขึ้น ซึ่งต้องพิจารณา ถึง use case ด้วย

3. รวมมี API Versioning หาก API มีผู้ใช้งานมาใช้งานแล้ว ในการแก้ไขสิ่งที่มีขนาดใหญ่ ก็จะทำได้ลำบากขึ้น เพราะถ้าแก้ไปแล้ว ทำให้ service ที่ใช้ API อยู่ใช้งานไม่ได้อาจจะทำให้เกิดปัญหาขึ้น เพราะอย่างนั้น ทุก ๆ API ควรทำ version ไว้ หากมีการเปลี่ยนแปลง ก็สามารถแยกออกมารูปแบบอีก version ได้เลย โดยทั้ง version เก่าและใหม่ต้องทำงานได้ทั้งคู่ ตัวอย่างเช่น POST v1/users และ POST v2/users สามารถนำเลข version มาต่อหน้า API ได้เลย
4. การตั้งชื่อ (Naming Conventions) ให้สัมพันธ์กันทั้งระบบ การตั้งชื่อตัวแปร (ของ body และ response) ที่พบเจอบ่อยที่สุดจะ เป็น camel case, snake case เป็น key ซึ่งสามารถเลือกใช้ได้ตามใจชอบ แต่ควรตั้งชื่อให้เหมือนกันทั้งระบบ
5. ใช้งาน parameters ให้เหมาะสม parameters คือ query ที่ต่อท้าย URL path ซึ่งจะมี action ต่างๆ ดังนี้
  - Filtering (การกรองข้อมูล) สามารถกรองข้อมูลแบบมีเงื่อนไขได้ โดย จะส่งผ่านมาทาง query ที่ต่อท้าย URL path เช่น GET /orders?name=MyOrders&customerId=2 ผลลัพธ์ทั้ง filter คือข้อมูลของ order ที่มีชื่อว่า "MyOrders" และ อยู่ใน customer id ที่ 2 จากหลักการข้อ 2. ที่กล่าวไปว่า ข้อมูล Nested ที่เยอะมาก ๆ หรือต้องการกรองข้อมูลจำนวนมาก ถ้านำมาเป็น URL path จะมีความยาวเกินไป สามารถนำมายัง filter ต่อท้าย URL path แทนได้

- Sorting (การจัดเรียงข้อมูล) สามารถเรียงลำดับข้อมูลที่เรียกมาแสดงผลได้ ซึ่งการอوكแบบ sort ที่ต้องออกแบบให้ยืดหยุ่น สามารถเรียงจากน้อยไปมาก หรือมากไปน้อยได้ โดยใส่ query เช้าไปต่อท้าย path คล้ายกับ filter ซึ่งสามารถนำ sort by ไปต่อท้ายได้ เช่น GET /users?sort by=+email หรือ GET /users?sort by=-email จากตัวอย่างด้านบน +email คือการเรียงจากน้อยไปมาก และ -email เรียงจากมากไปน้อย หรือสามารถเขียนในรูปแบบอื่นๆ ได้เช่น
  - GET /users?sort by=asc(email) หรือ GET /users?sort by=desc(email)
  - GET /users?sort by=email.asc หรือ GET /users?sort by=email.desc
  - GET /users?sort by=email&order by=asc หรือ GET /users?sort by=email&order by=desc

โดยหลักสำคัญจะอยู่ที่ทุกรูปแบบการเขียนจะต้องหยิบหยุ่น สามารถเปลี่ยนลำดับการเรียงข้อมูลได้ และผู้ใช้งานสามารถอ่านได้อย่างเข้าใจว่าเป็นการเรียงลำดับแบบไหน

- Searching (การค้นหาข้อมูล) หลักการจะค่อนข้างคล้าย filter คือการค้นหาข้อมูลแบบมีเงื่อนไข เมื่อต้องการค้นหาข้อมูล จะส่งผ่านมาทาง query ต่อท้าย String ตัวอย่างเช่น เมื่อต้องการค้นหาชื่อของ order ที่ชื่อว่า “THAIPOST1234” ในระบบ จะส่งผ่านทาง query params ตามตัวอย่างด้านล่าง GET /orders?search='THAIPOST1234'
- Pagination (การจัดแบ่งหน้า) [25] สามารถจัดหน้าของข้อมูลได้ในกรณีที่ข้อมูลมีจำนวนมาก



รูปที่ 2.5: tab เลือกหน้าของ Google

ตัว paginate จะช่วยย่อยข้อมูลออกมาเป็นก้อนเล็ก ๆ โดยสามารถโอน query เพื่อรับหน้า จำนวนข้อมูลที่ต้องการแสดงได้ เช่น GET /orders?page=2&limit=50 ผลลัพธ์ที่ได้จะเป็นข้อมูลตัวที่ 51–100 นั้นเอง เพราะเป็น page ที่ 2 ข้อมูลจำนวน 50 ตัว

6. ใช้ HTTP Status code ให้ตรงตามความหมาย หลังจากที่ผู้ใช้งาน API (Client) ส่ง request ไปหา server ผ่าน API แล้วฝ่าย Client จะต้องทำการยืนยันให้ได้ว่า API ใช้งานได้จริงหรือไม่ หรือส่งไปสำเร็จใหม่ จึงมีการต้องส่ง response ที่มี HTTP Status code ระบุ กลับไปยัง client เพื่อบอกว่า request นั้น ๆ Pass, Fail หรือ request นั้นผิด กรณี Success จะมี HTTP status code ที่ใช้งานกันทั่วไปได้แก่

- 200 Ok: เป็นมาตรฐานของ HTTP response เพื่อบ่งบอกว่า request นั้นสำเร็จ ใช้สำหรับ GET, PUT หรือ POST ก็ได้
- 201 Created: เป็น response เพื่อบ่งบอกว่าข้อมูลใหม่ได้ถูกสร้างขึ้นสำเร็จ ใช้สำหรับ POST
- 204 No Content: เป็น response สำหรับบ่งบอกถัดมีการ Success แต่ไม่ได้ return ข้อมูลกลับ ส่วนใหญ่จะใช้กรณีลบข้อมูล DELETE ที่ไม่ได้ส่ง response ที่เป็นข้อมูลกลับไป กรณี Error จะมี HTTP status code ที่ใช้งานกันทั่วไปได้แก่
- 400 Bad Request: status นี้จะบ่งบอกว่า request ที่ส่งมาโดย client นั้นไม่มี action ใดๆ และ Server ไม่เข้าใจ เช่น JSON ผิด หรือ parameters ไม่ถูกต้อง
- 401 Unauthorized: เป็น response ที่บ่งบอกว่า client ไม่ได้รับอนุญาตในการเข้าถึง อาจจะเป็นกรณีที่ใส่ token ผิด หมดอายุ หรือไม่ได้แนบ token มา
- 403 Forbidden: เป็น response ที่บ่งบอกว่า client ได้รับการอนุญาตในการเข้าถึงระบบ (login ผ่าน) แต่จะมีข้อมูลบางหน้า ที่ไม่มีสิทธิ์ในการเข้าถึง
- 404 Not Found: เป็น response ที่บ่งบอกว่า request นั้นไม่ว่าจะใช้งานตอนนี้ หรือ request ที่เรียกนั้นไม่มีอยู่ในระบบ

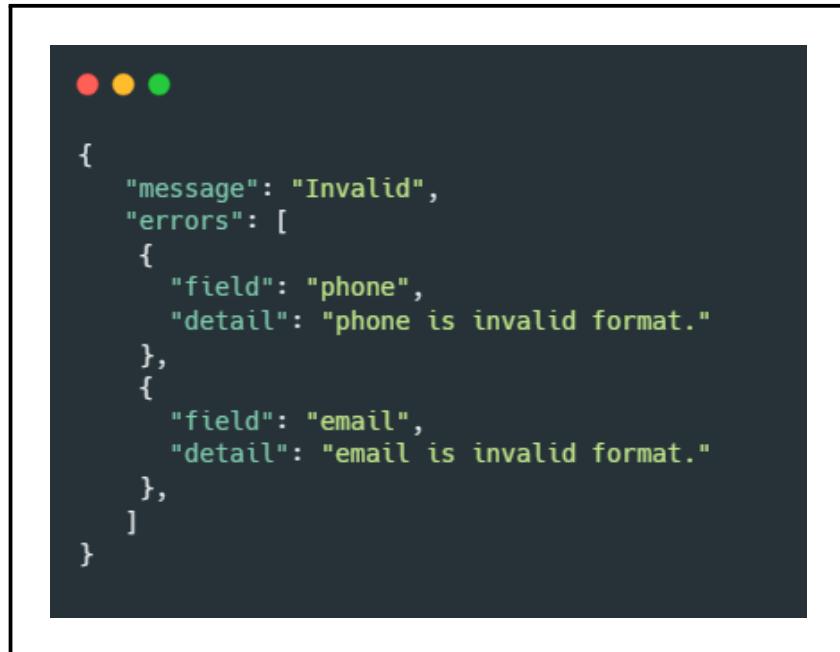
- 405 Gone: เป็น response ที่บ่งบอกว่า resource ที่ต้องการนั้นไม่มี หรือถูกย้ายไป
- 429 Too many Request: เป็น response ที่บ่งบอกว่า request นั้นติด limit ใช้กรณีที่กำหนด rate limit ไว้กับ API นั้น ๆ จะสามารถเรียกได้ครั้ง
- 500 Internal Server Error: เป็น response ที่บ่งบอกว่าการ request นั้นถูกต้องแล้ว แต่ server พังเอง ซึ่งอาจจะพังที่ตัวโค้ดของระบบเอง
- 503 Service Unavailable: เป็น response ที่บ่งบอกว่า server ใช้การไม่ได้ (ระบบพัง) โดย Server จะไม่สามารถรับ request ที่ส่งเข้ามาได้
- 504 Bad Gateway Gateway Timeout: เป็น response ที่บ่งบอกว่า web server อย่างพวก nginx หรือ apache พัง จนเห็นว่า HTTP Status code แต่ละตัวจะมีความหมายของของตัวเองขัดเจน เพราะฉะนั้นการออกแบบที่ดีจะต้องเลือกให้ HTTP Status code ให้ตรงตามวัตถุประสงค์เพื่อให้ผู้ใช้งานที่ได้รับ response กลับไป เข้าใจ response นั้นได้มากขึ้น

7. การ Handle Error ให้ user เข้าใจ นอกเหนือจาก HTTP Status code และ ต้องออกแบบ response สำหรับ error กรณีต่าง ๆ ไว้ด้วย เพื่อให้ user เข้าใจ error ของ API มากขึ้น ยกตัวอย่างกรณีที่ request ส่งบาง parameter มาไม่ถูกต้อง แทนที่จะ response กลับไปว่า



รูปที่ 2.6: Handle Error1

ให้นดเป็นเหมือน entity ของกราฟสามารถที่จะเก็บ attribute จำนวนมากได้ สำหรับกราฟเราจะเรียก attribute ว่าคุณสมบัติ ความสัมพันธ์หมายถึงความสัมพันธ์ที่เชื่อมระหว่าง 2 โนด และเหมือนกับโนดซึ่งสามารถเก็บคุณสมบัติได้



```

{
  "message": "Invalid",
  "errors": [
    {
      "field": "phone",
      "detail": "phone is invalid format."
    },
    {
      "field": "email",
      "detail": "email is invalid format."
    }
  ]
}

```

รูปที่ 2.7: Handle Error2

การส่งไปในรูปแบบดังกล่าวจะทำให้ user เข้าใจได้เล่าย่าว่ามี field ใดที่ผิด format

### 2.1.5 Graph Database

#### 2.1.5.1 Graph Database คืออะไร

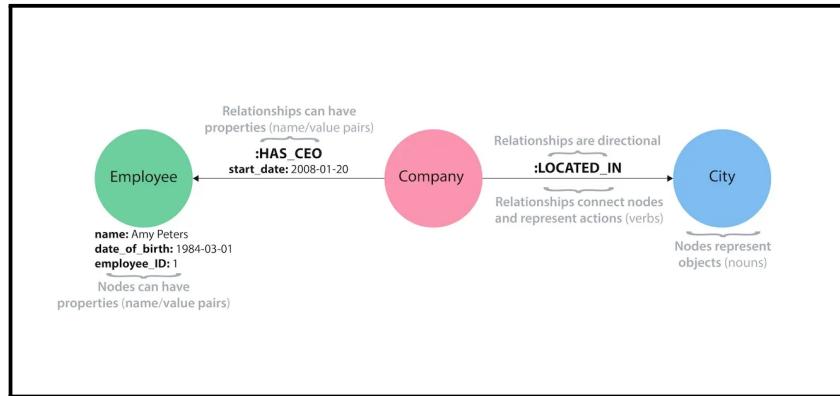
Graph database [26] หรือฐานข้อมูลแบบกราฟ จัดเป็น NoSQL Database รูปแบบหนึ่ง ซึ่งนำมาใช้แก้ปัญหา database ที่มีข้อมูลขนาดใหญ่และไม่มีรูปแบบชัดเจน ฐานข้อมูลแบบ Graph ออกแบบมาเพื่อแสดงความสัมพันธ์ (Relationship) ระหว่างข้อมูลที่มีความเชื่อมโยงกับข้อมูลที่เราสนใจได้อย่างชัดเจน รวมถึงมีความสามารถในการเก็บข้อมูลที่ไม่ต้องกำหนดรูปแบบล่วงหน้า

#### 2.1.5.2 ทำไมต้องเป็น Graph Database

การเข้าถึง node และ relationships ในฐานข้อมูล Graph เป็นวิธีที่มีประสิทธิภาพและใช้เวลาในการทำงานคงที่ และช่วยให้เราสำรวจการเชื่อมต่อหอยล้านต่อวินาทีต่อเร็วๆ ไม่ว่าจะมาจากขนาดรวมของข้อมูลทั้งหมดของเรา ทำให้ฐานข้อมูลแบบ Graph มีความสามารถในการจัดการข้อมูลที่มีรูปแบบซับซ้อนและมีความเชื่อมต่อ กันสูงได้ประสิทธิภาพ

#### 2.1.5.3 Property ของ Graph Model [3]

เทคโนโลยีส่วนใหญ่มีวิธีการที่แตกต่างกันเล็กน้อยในการสร้าง องค์ประกอบที่สำคัญของฐานข้อมูล Graph วิธีหนึ่ง คือ Graph Model ข้อมูลจะถูกจัดระเบียบเป็น node, relationship และ properties(ข้อมูลที่อยู่บน node หรือ relationship)



รูปที่ 2.8: Property ของ Graph Model

Nodes เป็นเหมือน entity ของ Graph สามารถที่จะเก็บ attribute จำนวนมากได้ สำหรับ Graph เราจะเรียก attribute ว่า properties Relationships เป็นความสัมพันธ์ที่เชื่อมระหว่าง 2 node และเหมือนกับ node มันสามารถเก็บ properties ได้

#### 2.1.5.4 Graph Database vs Relational Database [4]

	Relational Database	Graph Database
รูปแบบการเก็บข้อมูล	ตารางที่มีແറາและຄอลัมน์	โหนดที่เชื่อมต่อถึงกันพร้อมข้อมูลที่แสดงเป็นเอกสาร JSON
การทำงาน	การทำงานของ SQL เช่น สร้าง อ่าน อัปเดต และลบ (CRUD)	การดำเนินการรวมถึง CRUD และการดำเนินการผ่านกราฟตามทฤษฎีกราฟทางคณิตศาสตร์
ความสามารถในการปรับขนาด	ฐานข้อมูลแบบเชิงสัมพันธ์แบบตั้งเดิม สามารถปรับขนาดในแนวตั้งได้แต่ไม่ค่อยเชี่ยวชาญกับการปรับขนาดในแนวโน้ม	ฐานข้อมูลแบบกราฟเชี่ยวชาญในการปรับขนาดตามแนวโน้ม สามารถใช้การแบ่งพาร์ติชันเพื่อกระจายข้อมูลไปยังโหนดจำนวนมาก
ประสิทธิภาพ	ฐานข้อมูลแบบเชิงสัมพันธ์เชิงลึก การสืบคันที่ซับซ้อนเมื่อสำรวจความสัมพันธ์ที่อาจทำให้ประสิทธิภาพการทำงานช้าลง	ฐานข้อมูลแบบกราฟเชี่ยวชาญในการแสดงและสืบค้นความสัมพันธ์ระหว่างข้อมูล
ความสามารถในการใช้งาน	ฐานข้อมูลแบบเชิงสัมพันธ์ทำงานได้กับข้อมูลขนาดใหญ่และข้อมูลที่มีโครงสร้างพวกลมไม่ค่อยเชี่ยวชาญเมื่อเป็นเรื่องการสืบค้นแบบหลายช่วง	ฐานข้อมูลแบบกราฟใช้งานง่าย เมื่อต้องจัดการกับข้อมูลที่เน้นความสัมพันธ์เป็นหลัก เมื่อใช้ภาษาสืบค้นแบบกราฟ คุณสามารถสืบค้นข้อมูลของหลายรายการได้อย่างรวดเร็ว

ตารางที่ 2.2 Graph Database vs Relational Database

## 2.1.6 แผนภาพ UML

### 2.1.6.1 UML Diagram คืออะไร

แผนภาพ UML [27] (UML Diagram) คือ แผนภาพที่ใช้ในการแสดงและอธิบายโครงสร้างและพฤติกรรมของ code เพื่อสื่อสารให้ผู้ที่เกี่ยวข้องทุกคนเข้าใจตรงกัน ซึ่งสามารถนำมาใช้อธิบายความสัมพันธ์ของสิ่งต่างๆ ที่อยู่ในชั้นงานได้โดยแผนภาพ UML มีแผนภาพหลายรูปแบบสำหรับใช้อธิบายโครงสร้างและความสัมพันธ์ในรูปแบบต่างๆ ที่นักพัฒนาต้องทำความเข้าใจเพื่อใช้ในการปฏิบัติงาน

### 2.1.6.2 Use case Diagram

Use Case Diagram เป็นหนึ่งในแผนภาพ UML ที่ใช้ในการแสดงภาพรวมของวิธีการใช้ระบบหรือซอฟต์แวร์ จากมุมมองของผู้ใช้หรือแต่ละกลุ่มผู้ใช้ โดยทำให้ย่าต่อการเข้าใจและสื่อสารความต้องการของระบบกับผู้ใช้และทีมพัฒนา ลักษณะหลักของ Use Case Diagram ประกอบด้วย

1. Actor : แสดงตัวบุคคลหรือระบบที่มีส่วนร่วมในการใช้งานระบบ สามารถเป็นบุคคล, ระบบภายนอก, หรือภายในระบบได้
2. Use Case : แสดงกิจกรรมหรือฟังก์ชันที่ระบบหรือซอฟต์แวร์ให้บริการในแต่ละคำสั่งหรือเหตุการณ์ที่มีผู้ใช้ร้องขอ
3. Association : แสดงความสัมพันธ์ระหว่างผู้ใช้กับ Use Case
4. System Boundary : แสดงขอบเขตของระบบที่กำหนดไว้ใน Use Case Diagram
5. Include Relationship : แสดงว่า Use Case หนึ่งสามารถเรียกใช้ (include) Use Case อื่น ๆ ในทำนองของการนำเข้า (include)
6. Extend Relationship : แสดงว่า Use Case นึงสามารถขยาย (extend) ไปยัง Use Case อื่น ๆ ในกรณีที่มีเหตุการณ์เฉพาะที่เกิดขึ้น

Use Case Diagram มีประโยชน์มากในการทำความเข้าใจและกำหนดความต้องการของระบบจากมุมมองของผู้ใช้ และช่วยให้ทีมพัฒนามีภาพรวมของฟังก์ชันและการทำงานของระบบที่ชัดเจน

### 2.1.6.3 Sequence Diagram

Sequence Diagram เป็นหนึ่งในแผนภาพ UML ที่ใช้งานเพื่อแสดงลำดับขั้นตอนหรือการทำงานของวัตถุต่าง ๆ ภายใต้ระบบหรือโปรแกรม ในแต่ละขั้นตอนของการทำงานนั้นๆ ลักษณะหลักของ Sequence Diagram ประกอบด้วย

1. Lifeline : แสดงสิ่งต่าง ๆ ที่มีบทบาทในกระบวนการ สามารถเป็นวัตถุ, คลาส, หรือนักพัฒนา
2. Message : แสดงการสื่อสารระหว่าง Lifeline สามารถแบ่งเป็น Synchronous (ทำงานพร้อมกัน) หรือ Asynchronous (ทำงานไม่พร้อมกัน)
3. Activation Box : แสดงช่วงเวลาที่วัตถุทำงาน หรือทำการเรียกใช้งาน
4. Return Message : แสดงการส่งคืนจากการทำงานหรือเรียกใช้งาน
5. Focus of Control : แสดงว่าในขณะที่โปรแกรมทำงาน, ควบคุมอยู่ที่วัตถุหรือสั่นชีวิตใด

## 2.1.7 Principle Design

### 2.1.7.1 Principle Design คืออะไร

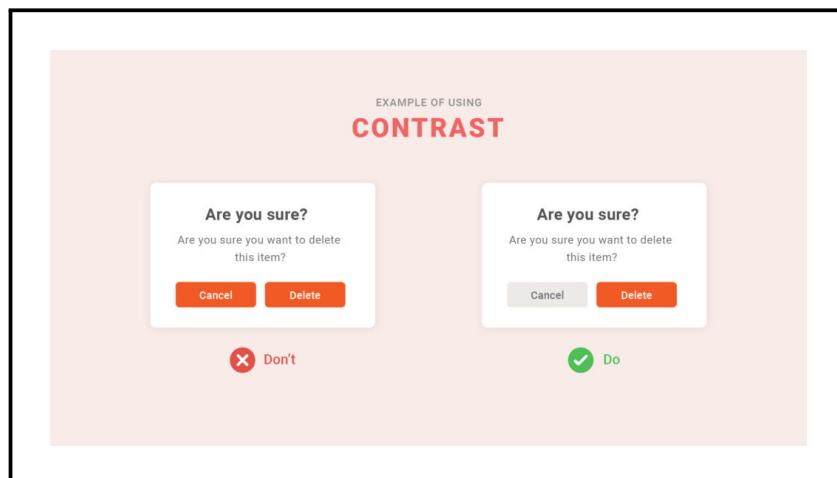
Principle Design [28] คือหลักการในการออกแบบที่ช่วยให้เข้าใจธรรมชาติในการรับรู้ข้อมูลและจำแนกประเภทของสมองมนุษย์ ยกตัวอย่างง่าย ๆ เช่นการที่เราสามารถที่จะนำสิ่งของที่หน้าตาคล้าย ๆ กันนำมาร่วมไว้ในกลุ่ม

### 2.1.7.2 C.R.A.P. Principle Design

โดยหลัก Principle design แบ่งเป็น 4 ข้อสำคัญคือ Contrast, Repetition, Alignment และ Proximity หรือตัวย่อ คือ C.R.A.P.

#### 1. ความแตกต่าง (Contrast)

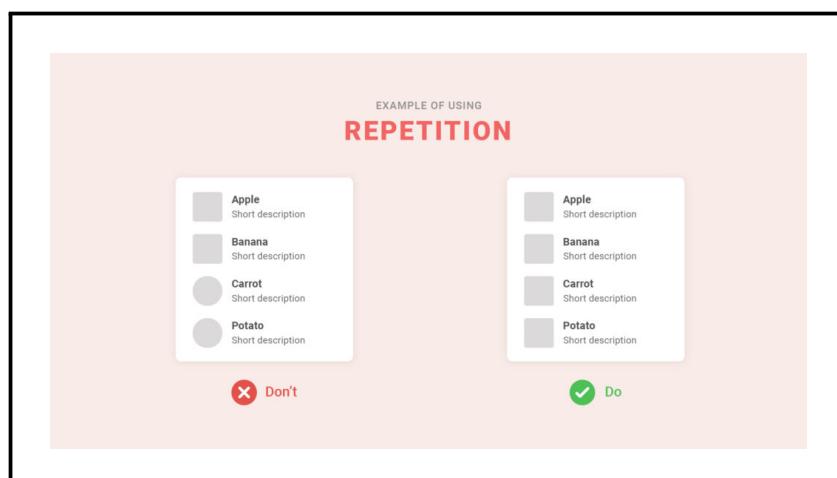
คือการแบ่งแยกข้อมูลด้วยความแตกต่าง โดยความแตกต่างในที่นี้ไม่ใช่แค่สีเพียงอย่างเดียว แต่รวมถึง ขนาดที่แตกต่างกัน รูปทรงที่แตกต่างกัน เป็นต้น เมื่อใช้ความแตกต่างในการแบ่งแยกข้อมูลแล้ว จะส่งผลต่อความสำคัญในการอ่าน ทำให้เราเลือกได้ว่าจะให้ผู้ใช้งานเห็นสิ่งไหนเป็นอย่างแรก และเห็นสิ่งไหนเป็นอย่างถัดมา ซึ่งสามารถดูวิธีแสดงความแตกต่างตามรูปที่ 2.9



รูปที่ 2.9: contrast [9]

#### 2. การทำซ้ำ (Repetition)

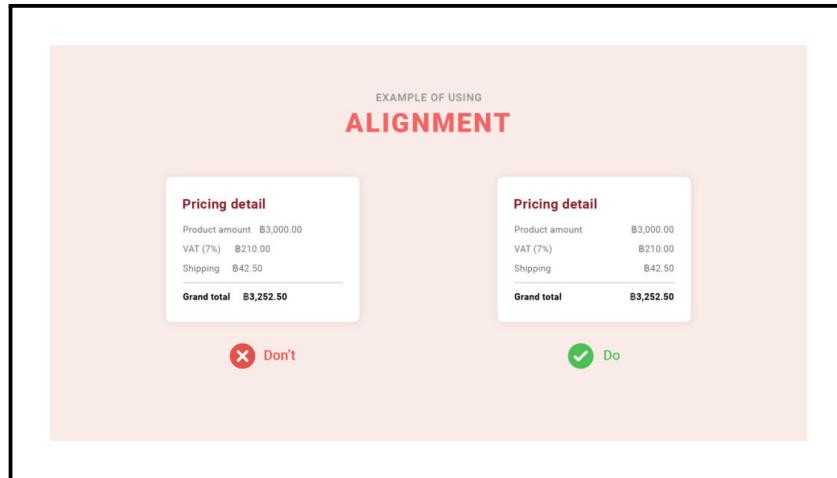
การทำซ้ำในที่นี้คือการทำซ้ำของเนื้อหา ทำให้ผู้ใช้งานไม่ต้องเรียนรู้ใหม่เมื่อเจอรูปแบบการจัดวางที่คล้าย ๆ กัน ก็จะเข้าใจได้ว่ามันมีการใช้งานที่เหมือนกัน หรือจะนำ Repetition มาใช้ในการแบ่งกลุ่ม เช่นของที่มีหน้าตาเหมือนกัน เมื่อนำมาเรียงไว้ด้วยกัน ผู้ใช้จะเข้าใจได้ว่ามันคือหมวดหมู่เดียวกัน ซึ่งสามารถดูตัวอย่างการทำซ้ำได้ดังรูปที่ 2.10



รูปที่ 2.10: repetition [10]

### 3. ตำแหน่งการจัดวาง (Alignment)

ส่งผลต่อความเป็นระเบียบ ความสวยงาม และความเชื่อมโยงกันของเนื้อหาโดย Alignment จะเป็น สิ่งแรก ๆ ที่ผู้ใช้งานรู้สึกได้ เมื่อมีการเปลี่ยนแปลง เมื่อจัดวางเนื้อหาในระบบเดียวกัน ผู้ใช้งานเข้าใจได้ว่าเนื้อหานี้มีความเชื่อมโยงกัน ซึ่งสามารถดูตัวอย่างการจัดวางได้ตามรูปที่ [2.11](#)



รูปที่ 2.11: alignment [[11](#)]

### 4. การจัดกลุ่มข้อมูล (Proximity)

คือการจัดวางองค์ประกอบของข้อมูล โดยข้อมูลที่มีความเชื่อมโยงกันควรจัดให้เป็นกลุ่มก้อนเดียวกัน จะช่วยให้ข้อมูลในหน้าเว็บใช้งานได้สะดวกขึ้นที่เรารอออกแบบแบบมีความซับซ้อนน้อยลง และมีการแบ่งหมวดหมู่ชัดเจนขึ้น ซึ่งจะเห็นได้ดังตัวอย่างในรูปที่ [2.12](#)



รูปที่ 2.12: proximity [[12](#)]

## 2.2 เทคโนโลยี

### 2.2.1 Integrated Development Environment (IDE)

- Visual Studio Code [29]

โปรแกรมสำหรับเขียนโค้ดที่ใช้ในการแก้ไขและปรับแต่งโค้ด จากค่ายไมโครซอฟท์ มีการพัฒนาออกแบบในรูปแบบของ OpenSource จึงสามารถนำมายังงานได้แบบฟรี ๆ สนับสนุนภาษาโปรแกรมมีมากมายทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่าง ๆ ให้เลือกใช้อย่างมากมาย

- Pycharm [30]

โปรแกรมสำหรับเขียนโค้ดสำหรับภาษาไฟทอน ติดตั้งบนเครื่องคอมพิวเตอร์ ได้ทั้งบนระบบปฏิบัติการ Windows, MacOS และ Linux

- Google Colab [31]

โปรแกรมสำหรับเขียนโค้ดสำหรับภาษาไฟทอนในเบราว์เซอร์ โดยไม่ต้องกำหนดค่าใดและสามารถเข้าถึง GPU โดยไม่มีค่าใช้จ่าย

### 2.2.2 Design

- Figma [32]

เครื่องมือออกแบบ User interface โดยสามารถใช้ออกแบบได้ตั้งแต่เว็บไซต์ แอปพลิเคชัน หรือโลโก้ ในรูปแบบที่มีลูกเล่นมากกว่าที่เคยเห็นในอดีต เช่น การออกแบบ Interactive component เป็นต้น

- LucidChart [33]

เว็บแอปพลิเคชันสำหรับสร้างโดยแกรม ผังงาน แผนภาพแบบจำลอง หรือแผนที่ความคิด สามารถแชร์แผนภาพให้ผู้อื่นเพื่อทำงานร่วมกันแบบเรียลไทม์ได้ มีเทมเพลตสำหรับรูปแบบที่เลือกใช้งานได้หลากหลายรูปแบบ เช่น ผังงาน แบบโครงร่าง แผนภาพเครือข่าย และแผนผังเว็บไซต์ เป็นต้น นอกจากนี้ยังสามารถแสดงความคิดเห็น หรือสนทนารูปแบบกลุ่มได้ และยังดาวน์โหลดเป็นไฟล์รูปแบบต่าง ๆ ได้

### 2.2.3 Frontend

- React Native [34]

Cross-Platform Framework ที่ใช้ในการพัฒนา Native Mobile Application สำหรับ Android และ iOS ที่พัฒนาโดยบริษัท Facebook Inc. React Native มีหลักการคล้ายกับ Xamarin คือสามารถ Reuse Code ได้มากกว่า 70%

### 2.2.4 Backend

- FastAPI [35]

FastAPI คือเว็บเฟรมเวิร์กที่มีความรวดเร็วและประสิทธิภาพสูง สำหรับการสร้าง API ด้วยภาษา Python ที่มีจุดเด่นได้แก่

1. มีความรวดเร็ว ประสิทธิภาพเทียบเท่ากับ NodeJS และ Go
2. สร้างง่าย เพิ่มความเร็วในการพัฒนา
3. ลดข้อผิดพลาดที่เกิดจากมนุษย์ (นักพัฒนา)

### 2.2.5 Database

- Neo4j [36]

เป็นระบบฐานข้อมูลที่ถูกออกแบบมาเพื่อจัดเก็บข้อมูลแบบกราฟ (Graph Database) ที่สามารถจัดเก็บแบบและสอบถามความสัมพันธ์ระหว่างข้อมูลได้อย่างมีประสิทธิภาพและมีความยืดหยุ่น ระบบฐานข้อมูลกราฟนี้ถูกออกแบบมาเพื่อเก็บข้อมูลในรูปแบบของ Node และ Edge ที่จะต่อระหว่างโหนดด้วยเส้นเชื่อมที่เรียกว่า Relationships ทำให้สามารถแสดงความสัมพันธ์และการเชื่อมโยงของข้อมูลได้อย่างชัดเจน

### 2.2.6 Machine Learning

- mT5

เป็นโมเดลการเรียนรู้ของเครื่องสำหรับการประมวลภาษาธรรมชาติที่มีความหลากหลายในการรับรองภาษาต่าง ๆ ซึ่งถูกพัฒนาโดย Google Research และเป็นการปรับปรุงจากโมเดล T5 (Text-to-Text Transfer Transformer) ซึ่งเป็นโมเดลที่มีความสามารถในการเรียนรู้จากข้อมูลข้อความและการประมวลผลข้อความอย่างมีประสิทธิภาพ

- BERT-th

เป็นโมเดลการเรียนรู้ของเครื่องสำหรับการประมวลภาษาธรรมชาติที่ถูกพัฒนาขึ้นเพื่อใช้ในภาษาไทย โดยเดลนี้มีความสามารถในการเข้าใจและประมวลผลข้อมูลที่เป็นภาษาไทยอย่างมีประสิทธิภาพ โดยใช้หลักการของ BERT (Bidirectional Encoder Representations from Transformers) ซึ่งเป็นโมเดลการเรียนรู้ของเครื่องที่สามารถทำงานคำถัดไปในประโยคจากข้อมูลทั้งด้านซ้ายและด้านขวาของคำนั้น ๆ

- fastText

เป็น library สำหรับการทำโมเดลการเรียนรู้ของเครื่องสำหรับการประมวลภาษาธรรมชาติ พัฒนาโดย Facebook AI Research และเน้นการสร้างเวกเตอร์คำและการจัดกลุ่มคำพิเศษ โดยใช้หลักการของการแปลงคำเป็นเวกเตอร์ที่สามารถใช้ในการค้นหาความคล้ายคลึงระหว่างคำ

- thai2fit

เป็นโมเดลการเรียนรู้ของเครื่องสำหรับการประมวลภาษาธรรมชาติที่ได้รับการปรับปรุงและเพิ่มประสิทธิภาพสำหรับการทำงานกับข้อมูลภาษาไทย โดยใช้หลักการของ fastText และ Word Embedding ที่ถูกพัฒนามาเพื่อภาษาไทย

- BERT-Base-Multilingual-Case

เป็นโมเดลการเรียนรู้ของเครื่องสำหรับการประมวลภาษาธรรมชาติที่รองรับหลายภาษาและมีความสามารถในการประมวลผลข้อความในหลายภาษาที่มีตัวอักษรต่างกัน โดยเดลนี้ถูกพัฒนาโดย Google และเป็นการปรับปรุงจาก BERT โดยรองรับการแปลงตัวอักษรเป็นตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก

- Thai nlp

เป็นแหล่งข้อมูลหลักสำหรับนักวิจัยและผู้พัฒนาที่กำลังทำงานด้าน Natural language processing ในภาษาไทย ซึ่งจะรวบรวมเครื่องมือการทำ Natural language processing โดยมีความหลากหลายของเครื่องมือและข้อมูลที่เป็นประโยชน์สำหรับงานวิจัยและการพัฒนาในด้าน NLP ในภาษาไทยเช่นเดียว โดยมีความหลากหลายของเครื่องมือและข้อมูลที่มีคุณภาพสูง เช่น โมเดลการแปลงภาษา การจัดกลุ่มคำพิเศษ และวิธีการประมวลผลข้อมูล ที่ถูกพัฒนาให้ใช้งานได้อย่างมีประสิทธิภาพ

### 2.2.7 Version Control

- Git [37]

Version Control ตัวหนึ่งซึ่งจะเป็นระบบที่มีหน้าที่ทำการจัดเก็บการเปลี่ยนแปลงของไฟล์ใน Project และมีการ backup ให้สามารถที่จะเรียกดูหรือทำการย้อนกลับไปดูเวอร์ชันต่างๆของ Project ที่ได้เวลาใดก็ได้ ดังนั้น Version Control ก็หมายความว่าสามารถเชื่อมโยงกับคนที่ทำงานใน Project ได้โดยตรง ไม่ว่าจะเป็นผู้ที่เขียนโค้ด ผู้ที่ออกแบบ UI หรือผู้ที่ทดสอบ

- Github

เว็บไซต์เว็บที่ให้บริการในการฝากไฟล์ Git หรือพูดง่าย ๆ ก็คือ Git ที่อยู่บนเว็บไซต์นั้นเอง ซึ่งจะทำให้สามารถใช้ Git ร่วมกับคนอื่นได้โดยผ่านเว็บไซต์ซึ่งจะมักนิยมใช้กันมาก ในการเก็บ Project Open Source ต่างๆ

### 2.2.8 Testing

- Jest [38]

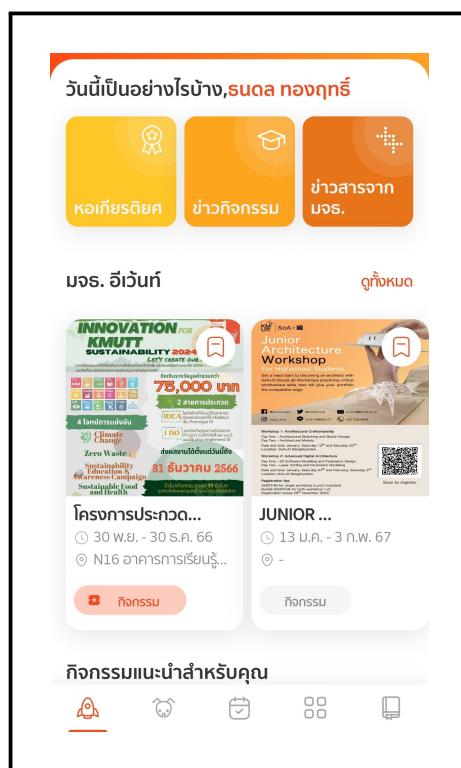
JavaScript Framework สำหรับเขียน Test เป็น Open Source ที่พัฒนาโดย Facebook ซึ่งมี helper มี function ต่างๆ ให้ใช้ทำให้ง่ายต่อการเขียน Test สามารถเขียนテストได้ทั้ง React, Vue, Angular หรือ JavaScript ทั่ว ๆ ไป

- PyTest

เป็นหนึ่งในเครื่องมือทดสอบโค้ดโปรแกรมภาษาไพทอนยอดนิยม โดย รองรับทั้ง Python 2 , Python 3 มี auto-discovery และอื่น ๆ ใช้ License: MIT license

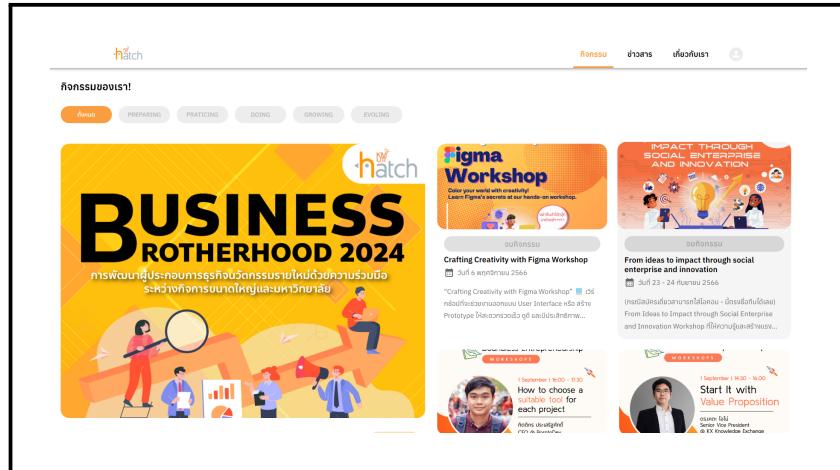
### 2.3 แบบสำรวจผลิตภัณฑ์

ModLink คือแอปพลิเคชันของมหาวิทยาลัยโดยข้อมูลภายใน แอปพลิเคชันนี้ จะเกี่ยวกับข้อมูลกิจกรรมต่าง ๆ ภายในมหาวิทยาลัย และข้อมูลส่วนตัวของนักศึกษา อย่างไรก็ตาม feature การแสดงข้อมูลเกี่ยวกิจกรรมของแอปพลิเคชันนี้ยังถือว่าทำได้ไม่ค่อยดีนัก เนื่องจากเป็นการกระจายข่าวสาร แบบทั่วไป ไม่ได้แบ่งแยกประเภทหรือแสดงตามที่ผู้ใช้ให้ความสนใจ ตัวอย่างหน้า UI ของ ModLink อยู่ในรูปที่ 2.13



รูปที่ 2.13: หน้าแนะนำกิจกรรมของ ModLink และรายละเอียด

KMUTT Hatch คือ เว็บไซต์สำหรับนักศึกษาและศิษย์เก่าเพื่อประชาสัมพันธ์กิจกรรมของ Hatch ทางเว็บไซต์มีการประชาสัมพันธ์ กิจกรรมต่างๆและข้อมูลที่จำเป็นต่อผู้ใช้งาน อย่างไรก็ตามเว็บไซต์นี้สามารถประชาสัมพันธ์ได้แค่กิจกรรมที่ทาง Hatch เป็นผู้จัดเท่านั้น



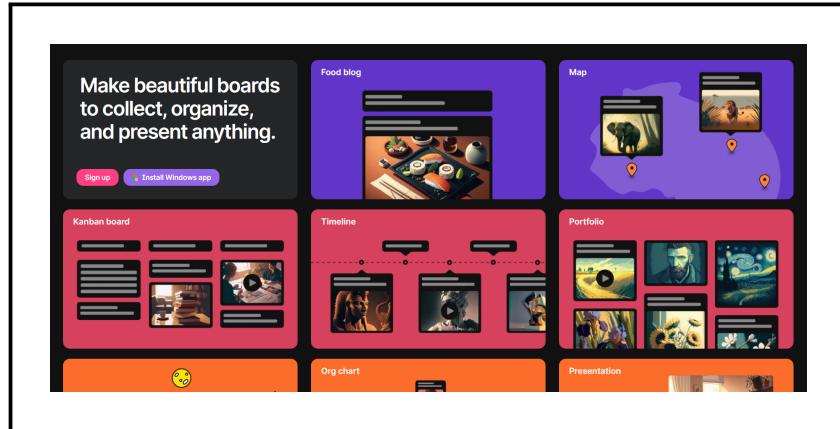
รูปที่ 2.14: หน้าแนะนำกิจกรรมของ KMUTT Hatch

KMUTT Sinfo คือ เว็บไซต์สำหรับนักศึกษาที่ทำการรวบรวมระบบจัดการงานต่าง ๆ ของนักศึกษาไม่ว่าจะเป็นการลงทะเบียนเรียน ดูเกรด หรือประเมินกิจกรรม ซึ่งแม้ว่างานต่าง ๆ ของนักศึกษานั้นจะมีศูนย์กลางมาที่เว็บไซต์นี้ แต่ถึงกระนั้นก็เป็นเว็บที่ใช้งานไม่ค่อยสะดวกเนื่องจากต้องเข้าผ่าน pop up ซึ่งต้องอาศัยการตั้งค่าและบังใช้งานไม่ได้ในบาง platform อีกทั้งยังจำกัดเวลาที่ใช้งานเอาไว้แล้วและแม้จะเป็นศูนย์รวมประวัติการทำกิจกรรมของนักศึกษา แต่กลับไม่สามารถแนะนำได้ว่านักศึกษานั้นให้ความสนใจในกิจกรรมใด ตัวอย่างหน้าใช้งานของ KMUTT sinfo อยู่ในรูปที่ 2.15



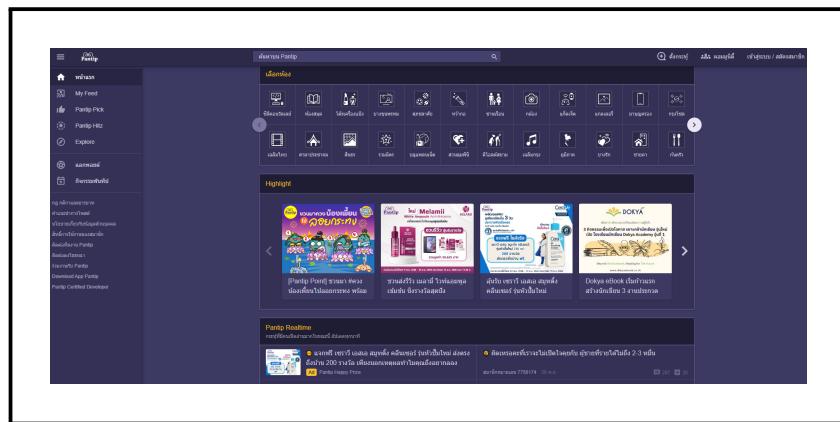
รูปที่ 2.15: หน้า Mainpage ของ KMUTT Sinfo

Padlet คือแอปพลิเคชันหรือเว็บไซต์ที่อยู่ในแพลตฟอร์มสำหรับ การระดมความคิด แสดงความคิดเห็น หรือแลกเปลี่ยนความรู้ร่วมกัน ผ่านกระดานดิจิทัลในรูปแบบเสมือน Post it ที่ติดบนบอร์ด ซึ่งจะแสดงผลทุกอย่างแบบ Real-time สามารถโพสต์ทั้งในรูปแบบข้อความ รูปภาพ และลิงก์ของเว็บไซต์ได้ เร็บเพลที่จะให้ผู้ใช้มาแสดงความเห็น หรือโพสต์ข้อมูลงบนเว็บ ซึ่งจะต่างจากตรงที่ทางระบบของเป็น ตัวกลางในการคำนวนหาจากความซื่นชอบของผู้ใช้จากการที่ผู้ใช้เคยได้เข้าร่วม หรือชุมรมที่ผู้ใช้สนใจอยู่ หน้าที่เข้าไปยังบอร์ดต่างๆ ของ Padlet อยู่ในรูปที่ 2.16



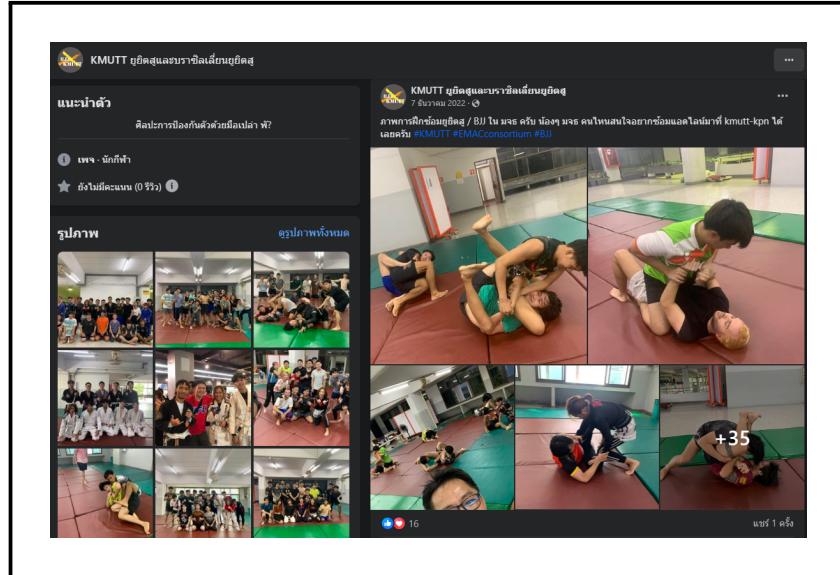
รูปที่ 2.16: หน้า Mainpage ของ Padlet

Pantip คือ พื้นที่สำหรับการแลกเปลี่ยนความคิดเห็นในหัวข้อ หรือประเด็นที่สนใจร่วมกัน สามารถสอบถาม บอกเล่าแบ่งปันประสบการณ์ ในเรื่องต่าง ๆ ในหน้ากระดานสนทนาก็สามารถถด้วยความสามารถด้วย หรือตอบกระทู้ต่าง ๆ ที่สมาชิกสนใจและสามารถเลือกหาอ่านข้อมูลได้จาก ป้ายหัวข้อในเรื่องต่าง ๆ ที่ทางเว็บไซต์ ได้สร้างขึ้นไว้ ซึ่งจะต่างจากตรงที่ทางระบบของ จะเป็นตัวกลางในการคำนวนหาจากความซื่นชอบ ของผู้ใช้จากการที่ผู้ใช้เคยได้เข้าร่วม หรือชุมรมที่ผู้ใช้สนใจอยู่ โดยรูปที่ 2.17 คือหน้าที่แสดงเนื้อหาต่างๆ ของ Pantip



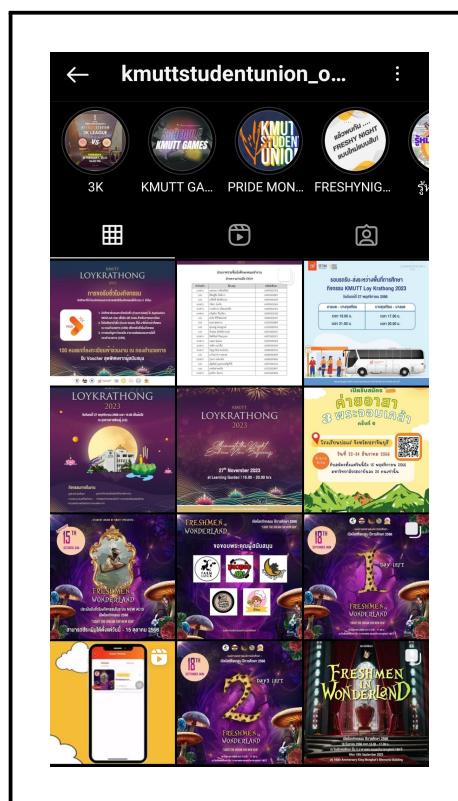
รูปที่ 2.17: หน้าแนะนำกระทุกของ Pantip

Facebook เป็น social media ที่ได้รับความนิยมที่สามารถใช้งาน ได้ในหลาย platform ซึ่งสามารถนำเสนอข้อมูลข่าวสารให้ผู้ใช้งาน ได้มากมายและเป็น social media ที่มีคนใช้งานแทบทลอดทั้งวัน ทำให้บางชุมรมเลือก ที่จะทำหน้าเพจเพื่อกระจายข่าวสารเกี่ยวกับชุมรม ของตัวเอง อย่างไรก็ตามด้วยปริมาณข่าวสารมากมายของ facebook ทำให้ข่าวของชุมรมมักโดดเด่นด้วย ข่าวอื่น ๆ อยู่เสมอ ถึงแม้จะ สามารถเข้าไปปล่อยหน้าเพจเพื่อประชาสัมพันธ์ใน Facebook ดูได้ในรูปที่ 2.18



รูปที่ 2.18: หน้า Homepage ของเพจชุมชนใน Facebook

Instagram คือ แอปพลิเคชันบน smartphone และอุปกรณ์คอมพิวเตอร์ โดยแอปพลิเคชันนี้จะเน้นการแชร์รูปภาพ บน Social Network ซึ่งทำให้เพื่อน ของคุณสามารถเห็นภาพถ่ายของคุณได้และยังสามารถคอมเม้นต์ภาพของคุณได้ ที่สำคัญ Instagram ยังสามารถ แชร์ภาพของคุณไปยัง Twitter และ Facebook ได้อีกด้วย ยังสามารถกดติดตามบุคคลที่ชื่นชอบเพื่อที่จะได้เห็นโพสต์รูปภาพ วิดีโอ ของ บุคคลนั้น ๆ ได้อีกด้วย และยังมีเฟิร์ม story ที่ใช้ในการอัพรูปภาพ วิดีโอลิปของคุณได้ด้วย โดยตัวอย่างของ instagram ของทางชุมชน สามารถดูได้จากรูปที่ 2.19



รูปที่ 2.19: หน้า Notifications ของ Instagram

ตารางความแตกต่างของ Feature

คู่แข่ง	แนะนำกิจกรรม	แนะนำชุมชน	เก็บประวัติ กิจกรรม	ประเมิน กิจกรรม	แจ้งเตือนกิจกรรม
mod link	!!	X	X	!!	O
kmutt sinfo	X	X	!!	O	X
kmutt hatch	!!	X	X	X	!!
padlet	!!	!!	X	X	X
pantip	!!	!!	X	X	X
facebook	!!	!!	X	X	X
ig	!!	!!	X	X	X

O คือ ใช้งานได้มีประสิทธิภาพ  
!! คือ ใช้งานได้แต่ไม่มีประสิทธิภาพ  
X คือ ไม่มี Feature

รูปที่ 2.20: ตารางความแตกต่างของ Feature

### ระบบการแนะนำกิจกรรม

คู่แข่ง	แนะนำกิจกรรม
mod link	แนะนำแต่กิจกรรมใหญ่ของมหาวิทยาลัย จึงไม่ค่อยตรงกับความต้องการของผู้ใช้
kmutt hatch	แนะนำแต่กิจกรรมที่ kmutt hatch เป็นผู้จัด จึงแนะนำได้แค่กิจกรรมที่เฉพาะทาง
kmutt sinfo	ไม่มีการแนะนำเกี่ยวกับกิจกรรม
padlet	ไม่มีการแนะนำจากทางระบบ แต่จะเป็นการที่ผู้ใช้ต้องเข้าไปค้นหาเกี่ยวกับกิจกรรมนั้น ๆ เอง
pantip	ไม่มีการแนะนำจากทางระบบ แต่จะเป็นการที่ผู้ใช้ต้องเข้าไปค้นหาเกี่ยวกับกิจกรรมนั้น ๆ เอง
facebook	จะมีการแนะนำกิจกรรมตามที่ผู้ใช้ได้ทำการปักหมุด
ig	ไม่มีการแนะนำจากทางระบบ แต่จะเป็นการที่ผู้ใช้ต้องเข้าไปค้นหาเกี่ยวกับกิจกรรมนั้น ๆ เอง

รูปที่ 2.21: ระบบการแนะนำกิจกรรม

### บทที่ 3 วิธีการทำงาน กระบวนการและการออกแบบ

ในบทนี้จะกล่าวถึงการสำรวจความต้องการของผู้ใช้งานเพื่อนำมาวิเคราะห์คุณลักษณะที่โครงการควรมี และนำสิ่งที่วิเคราะห์ได้มาใช้ออกแบบโครงสร้างและองค์ประกอบต่างๆของตัวผลิตภัณฑ์ โดยจะมีการอธิบายโครงสร้าง และองค์ประกอบต่างๆที่อยู่ในตัวผลิตภัณฑ์รวมถึงวิธีการทำงานโดยละเอียด

### 3.1 บหนា

### 3.1.1 สำรวจความต้องการของผู้ใช้เชิงคุณภาพ

ในการทำระบบเพื่อแจ้งเตือนข่าวสารของกิจกรรมและชุมชน ทางผู้จัดทำระบบทุกผู้ใช้งานและผู้ได้รับประโยชน์เป็น 3 กลุ่มใหญ่ๆด้วยกัน คือ ผู้เข้าร่วมกิจกรรม ผู้จัดกิจกรรม และประธานชุมชน โดยแต่ละกลุ่มนี้มีความต้องการดังนี้ ผู้เข้าร่วมกิจกรรมซึ่งเป็นผู้ใช้งานหลัก คือ นักศึกษาทั่วไปในมหาวิทยาลัย โดยมีความต้องการพื้นฐานคือการที่อย่างจะทำกิจกรรมตามความสนใจ แต่ด้วยภาระการเรียนทำให้ส่วนใหญ่ไม่มีเวลาที่จะหา กิจกรรมหรือ ชุมชนที่ตนเองสนใจ สิ่งที่ต้องการจะเป็นแหล่งที่รวมรวม ข้อมูลข่าวสาร กิจกรรม และ รายละเอียดของชุมชนต่างรวมถึงสิ่งที่ชุมชนนั้นทำเอาไว้ใน ที่เดียวเพื่อที่จะหา กิจกรรมและชุมชนตามที่ตัวเองต้องการได้ ผู้ได้รับประโยชน์กลุ่มแรก คือ ผู้จัดกิจกรรมที่แทนผู้จัดกิจกรรมที่เป็นเจ้าหน้าที่ของมหาวิทยาลัย โดยมีความต้องการพื้นฐานคือการประชาสัมพันธ์ กิจกรรมที่จัด แต่ข่าวสารของกิจกรรมที่ถูกเผยแพร่ผ่าน platform ต่างๆ เช่น facebook หรือ instagram นั้นมีผู้ที่เห็นการประชาสัมพันธ์เพียงบางส่วนเท่านั้น ทำให้ผู้เข้าร่วมกิจกรรมมีน้อยกว่าที่คาดหวัง ผู้ได้รับประโยชน์กลุ่มที่ 2 คือ ประธานชุมชน ที่แทนผู้ที่ดำเนินงานชุมชน ซึ่งส่วนใหญ่เป็นนักศึกษาที่ใกล้จะจบการศึกษา หรือนักศึกษาชั้นปริญญาโท โดยมีความต้องการพื้นฐาน คือ การประชาสัมพันธ์ชุมชนที่ตัวเองจัดการอยู่ ถึงแม้จะเป็นงานชุมชนจะ มี การนัดหมายกันแบบปากต่อปากอยู่แล้ว แต่การจัดการกิจกรรมชุมชน ก็ไม่เป็นระบบเท่าที่ควร อีกทั้งการหาสมาชิกชุมชนใหม่ หรือ การประชาสัมพันธ์กิจกรรมชุมชนให้คนภายนอกชุมชนจากการที่ สมาชิกชุมชนเป็นคนเชื่อมโยงกัน คือ ผู้เข้าร่วมกิจกรรมที่มีความสนใจในกิจกรรมนั้นอยู่ ซึ่งจากการสัมภาษณ์ทำให้พบว่าปัญหาใหญ่ที่เกิดขึ้นเกิดจากการที่ไม่มีแหล่งที่จะกระจายข้อมูลที่ผู้เข้าร่วมกิจกรรมต้องการในการตัดสินใจ เข้าร่วมกิจกรรม และ ชุมนุมได้อย่างเหมาะสม ทำให้กิจกรรม และ ชุมนุมไม่ได้รับความสนใจเท่าที่ควร

### 3.1.2 เส้นทางของผู้ใช้งาน (Journey Map)

จากกลุ่มผู้ใช้งานทั้งสามกลุ่มสามารถแบ่งพฤติกรรมการใช้งานได้เป็น 2 ประเภท คือ การหากิจกรรมหรือชุมชน

## ຜົກຮະທຳ: ຜົເຂ້າຮ່ວມກິຈກຣມ

Step 1 : ผู้เข้าร่วมกิจกรรมไปตามรายละเอียดจากผู้จัดกิจกรรมหรือประธานชุมชน

ปัญหา - ผู้เข้าร่วมอาจไม่รู้ว่ามีกิจกรรมหรือชุมชนนี้อยู่ หรือไม่รู้จักผู้จัดกิจกรรม

Step 2 : ដៃខោរវំលងទៅបើយនកិច្ចរមអីអមរម

Step 3 : ผู้เข้าร่วมเข้าร่วมกิจกรรมหรือทำกิจกรรมตามรูป

ป้าเหา - ผู้เข้าร่วมอาชีวศึกษาที่จัดกิจกรรมเรียนรู้

Step 4 : ผู้เข้าร่วมฯ ระบุเบิกกิจกรรมฯ

ីកុង - ីកុងរៀង ីកុងរៀងបោនិយោន ីកុងរៀងបោនិយោន

### 3.1.3 ຜົກຄ້າຮ່ວມຍົດງານ (Stakeholder)

ผู้ที่คาดว่าจะได้รับประโยชน์จากการใช้งานแอปพลิเคชันของผู้จัดทำสร้างขึ้นนั้นแบ่งเป็น 2 กลุ่ม คือ ผู้เข้าร่วมกิจกรรม และ ผู้ผู้จัดกิจกรรมที่หมายถึงผู้จัดกิจกรรมและประธานชุมชน ผู้ผู้เข้าร่วมกิจกรรม คือ นักศึกษาชั้นปีที่

1. เนื่องจากเป็นกลุ่มที่ยังไม่มีข้อมูลเกี่ยวกับสิ่งต่าง ๆ ในมหาวิทยาลัยอยู่น้อยทำให้ยากที่จะหาแหล่งข้อมูลของกิจกรรมหรือชุมชนที่สนใจ ฝ่ายผู้จัดกิจกรรม คือ นักศึกษาชั้นปีที่ 4 ที่เป็นประธานชุมชนและผู้จัดกิจกรรมที่เป็นเจ้าหน้าที่ของมหาวิทยาลัย เนื่องจากเป็นผู้ได้รับผลกระทบจากการประชาสัมพันธ์กิจกรรมและชุมชนที่ไม่มีประสิทธิภาพเท่าที่ควร

### 3.2 ความต้องการของผู้ใช้งาน (Requirement List)

รายการข้อกำหนดหรือความต้องการที่จำเป็นต้องมีในโครงการหรือผลิตภัณฑ์ที่กำลังถูกพัฒนา ข้อกำหนดเหล่านี้เป็นข้อมูลที่ถูกรวบรวมมาจากผู้ใช้, ลูกค้า, หรือผู้เกี่ยวข้องอื่น ๆ ซึ่งมีไว้เพื่อกำหนดขอบเขตและคุณลักษณะของผลิตภัณฑ์หรือโครงการ โดยในส่วนนี้ หลังจากที่ได้ทำการสำรวจความต้องการจากผู้ได้รับผลกระทบโดยชั้นแล้วทำให้เราได้ความต้องการมา ดังนี้

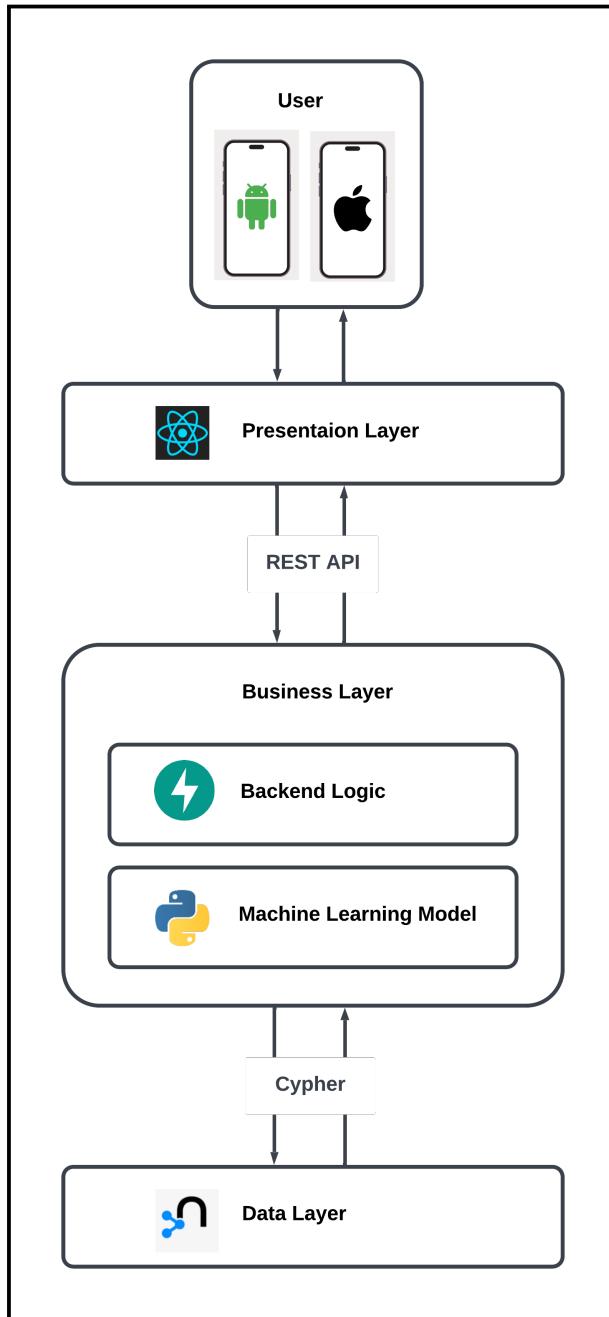
1. ข้อมูลของกิจกรรมและชุมชนที่ชัดเจน
2. ระบบการค้นหา กิจกรรมและชุมชน
3. การแนะนำชุมชนและกิจกรรมที่น่าสนใจ
4. การแจ้งเตือนการประเมินกิจกรรม

### 3.3 รายการคุณลักษณะ (Feature List)

จากการวิเคราะห์ความต้องการทั้งหมด เราได้ทำการวิเคราะห์คุณลักษณะมาเพื่อตอบโจทย์ความต้องการของผู้ใช้ดังนี้

1. Login: ลงชื่อเพื่อเข้าใช้งานแอพพลิเคชัน
2. Registration: สมัครบัญชีของแอพพลิเคชันด้วยอีเมลมหาวิทยาลัย
3. Logout: ออกจากระบบ
4. Search: ค้นหา กิจกรรมหรือชุมชนที่สนใจ
5. Select: เลือกอ่านรายละเอียด กิจกรรมหรือชุมชนที่สนใจ
6. Join:
  - ลงชื่อเข้าร่วม กิจกรรมที่สนใจ
  - ลงชื่อเป็นสมาชิกของชุมชนที่สนใจ
7. Resignation: ถอนชื่อจากการเป็นสมาชิกชุมชน
8. Notification:
  - แจ้งเตือน กิจกรรมที่เกี่ยวข้องกับชุมชนหรือความสนใจของนักศึกษา
  - แจ้งเตือน การประเมิน กิจกรรม
9. Event evaluation: ประเมิน กิจกรรม
10. Recommendation: แนะนำ กิจกรรมและชุมชน ตามความสนใจของผู้ใช้โดยอ้างอิงจาก tag ของกิจกรรม

### 3.4 แผนภาพสถาปัตยกรรมของระบบ (System Architecture Diagram)



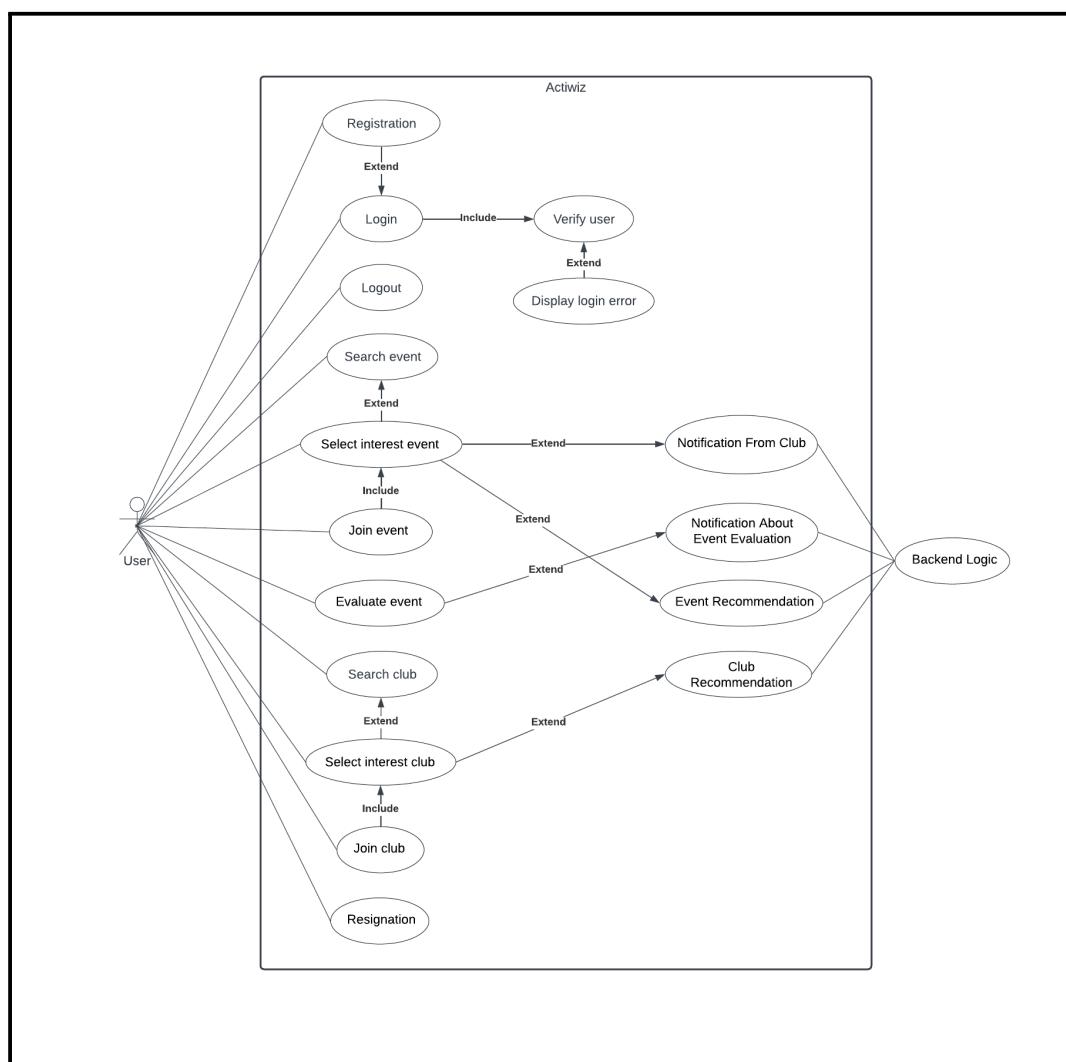
รูปที่ 3.1: Architecture diagram

จากรูปที่ 3.1 จะเป็นแผนภาพแสดงรายละเอียดโครงสร้างของระบบ เพื่อแสดงให้เห็นภาพรวมขององค์ประกอบ และเครื่องมือต่าง ๆ ที่ใช้ในการพัฒนาแอปพลิเคชัน Actiwiz โดยตัวแอปพลิเคชันนี้รองรับการทำงานทั้งในระบบปฏิบัติการทั้ง IOS และ Android โดยใน Presentation layer ถูกสร้างโดยใช้ react native ซึ่งสื่อสารกับ Business layer โดยอาศัย REST API Backend Logic ใช้ fast api ใน การพัฒนา และทำการสร้าง Model การเรียนรู้ของเครื่องโดยใช้ Python ซึ่งจะใช้ Cypher ในการสื่อสารกับ Data layer ที่ถูกสร้างขึ้นด้วย Neo4j database

### ผังงานระบบ (System Flow)

1. User ใช้งาน React Native Application
2. React Native ส่ง HTTP request ตามรูปแบบของ Rest API ที่กำหนดไว้ ไปยัง FastAPI backend เพื่อเรียกใช้งานฟังก์ชันต่าง ๆ เช่น ค้นหากิจกรรมหรือดูข้อมูลเกี่ยวกับชุมชน เป็นต้น
3. FastAPI ประมวลผล request และส่ง request เพิ่มเติมด้วย Cypher ไปยัง Neo4j database เพื่อขอข้อมูล หรือเปลี่ยนแปลงข้อมูล สำหรับการแนะนำและการประมวลผลข้อมูลกิจกรรมหรือชุมชน FastAPI จะติดต่อกับ Machine Learning Model เพิ่มเติม
4. Neo4j ประมวลผลข้อมูลตาม request ของ FastAPI
5. FastAPI ส่ง HTTP response กลับไปยัง React Native และ React Native ปรับเปลี่ยน User Interface ตาม response ที่ได้รับ

### 3.5 แผนภาพที่ใช้แสดงปฏิสัมพันธ์ระหว่างระบบงานและสิ่งที่อยู่นอกระบบงาน (Use Case Diagram)



รูปที่ 3.2: Use Case Diagram

จากการที่ 3.2 เป็นแผนภาพที่แสดงให้เห็นความสัมพันธ์ว่าแต่ละระบบมีความเกี่ยวข้องกันอย่างไรและผู้มีส่วนเกี่ยวข้องในระบบสามารถทำอะไรได้บ้าง โดยในฝั่งของผู้ใช้งานนั้นจะเริ่มตั้งแต่สามารถสมัครบัญชีการใช้งานได้ การที่เข้าสู่ระบบเพื่อใช้งานคุณลักษณะต่างๆ เช่น การค้นหา, อ่านรายละเอียดและเข้าร่วมกิจกรรม การประเมินกิจกรรม การค้นหา, เข้าร่วม และลากอกรายการ รวมถึงงานที่ระบบ Backend กระทำการต่อระบบ ทั้งการแนะนำ商品หรือกิจกรรม แจ้งเตือนกิจกรรมที่ชุมชนจัดหรือการประเมินกิจกรรมที่เกิดขึ้น

**Use Case Narratives** ส่วนนี้จะเป็นการบรรยายให้รายละเอียดเกี่ยวกับกรณีการใช้งานระบบต่าง ๆ ในตัวแอปพลิเคชันที่ถูกกำหนดไว้ในแผนภาพที่ใช้แสดงปฏิสัมพันธ์ระหว่างระบบงานและสิ่งที่อยู่นอกระบบงาน ทั้งเป้าหมายของผู้ใช้งาน, เงื่อนไขการใช้งาน และขั้นตอนต่างๆ ในกรณีใช้งานนั้นโดยละเอียด โดยจะแบ่งเป็น 3 ส่วนหลัก ได้แก่

1. Goal หรือเป้าหมายของระบบ
2. Pre-conditions หรือเงื่อนไขที่ต้องเกิดขึ้นก่อนที่จะใช้งานระบบในกรณีใช้งานนั้นได้
3. Main Success Scenario หรือขั้นตอนการใช้งานระบบในสถานการณ์ปกติ

ซึ่งแอปพลิเคชันที่เป็นผลิตภัณฑ์ของโครงการมี นั้นมีผู้กระทำหลักคือ User และผู้กระทำการคือ Backed Logic ซึ่งขั้นตอนที่เกิดขึ้นและเงื่อนไขในการเกิดขึ้นของแต่ละกรณีใช้งานมีรายละเอียดดังที่ระบุไว้ในตารางดังต่อไปนี้ การสมัครเข้าใช้งาน

Actor	User
Goal	ลงทะเบียนสร้าง account
Pre-conditions	-
Main success scenario	1.User ทำการกดเข้าหน้าลงทะเบียน 2.User กรอกแบบฟอร์ม 3.User กดยืนยันการสร้าง account

ตารางที่ 3.1 การสมัครเข้าใช้งาน

### การเข้าสู่ระบบ

Actor	User
Goal	เข้าสู่ระบบและใช้งานฟังก์ชันต่างๆ
Pre-conditions	-User จำเป็นต้องลงทะเบียน account ก่อน -User จำเป็นต้องยืนยัน account ก่อน
Main success scenario	1.User กรอกอีเมลและรหัสผ่าน 2.User เข้าสู่ homepage ของแอปพลิเคชัน 3.ระบบแสดงฟังก์ชันที่ใช้งานได้ทั้งหมด 4.User ใช้งานฟังก์ชันต่างๆ ในแอปพลิเคชัน

ตารางที่ 3.2 การเข้าสู่ระบบ

### การออกจากระบบ

Actor	User
Goal	ออกจากระบบ
Pre-conditions	-User ต้องเข้าสู่ระบบก่อน
Main success scenario	User ทำการกดออกจากระบบ

ตารางที่ 3.3 การออกจากระบบ

### การค้นหากิจกรรม

Actor	User
Goal	ค้นหากิจกรรม
Pre-conditions	- User ต้องเข้าสู่ระบบก่อน
Main success scenario	1.User ค้นหากิจกรรมที่ต้องการ 2.ระบบแสดงกิจกรรมที่ค้นหา

ตารางที่ 3.4 การค้นหากิจกรรม

### การอ่านรายละเอียดกิจกรรม

Actor	User
Goal	อ่านรายละเอียดกิจกรรม
Pre-conditions	- User ต้องเข้าสู่ระบบก่อน
Main success scenario	1.User กดไปที่กิจกรรมที่สนใจ 2.ระบบแสดงรายละเอียดกิจกรรมที่ค้นหา

ตารางที่ 3.5 การอ่านรายละเอียดกิจกรรม

### การเข้าร่วมกิจกรรม

Actor	User
Goal	เข้าร่วมกิจกรรม
Pre-conditions	- User ต้องเข้าสู่ระบบก่อน - User ต้องเข้าสู่หน้าอ่านรายละเอียดกิจกรรมก่อน
Main success scenario	1.User กดเข้าร่วมกิจกรรม 2.ระบบพาไปยังหน้ากรอกแบบฟอร์มของกิจกรรม

ตารางที่ 3.6 การเข้าร่วมกิจกรรม

### การประเมินกิจกรรม

Actor	User
Goal	ประเมินกิจกรรม
Pre-conditions	- User ต้องเข้าสู่ระบบก่อน
Main success scenario	1.User ได้รับแจ้งเตือนการประเมินกิจกรรม 2.User กดเข้าประเมินกิจกรรม 3.ระบบพาไปหน้าประเมินกิจกรรม

ตารางที่ 3.7 การประเมินกิจกรรม

### การค้นหาชุมชน

Actor	User
Goal	ค้นหาชุมชน
Pre-conditions	- User ต้องเข้าสู่ระบบก่อน
Main success scenario	1.User ค้นหาชุมชนที่ต้องการ 2.ระบบแสดงชุมชนที่ค้นหา

ตารางที่ 3.8 การค้นหาชุมชน

### การอ่านรายละเอียดชุมชน

Actor	User
Goal	อ่านรายละเอียดชุมชน
Pre-conditions	- User ต้องเข้าสู่ระบบก่อน
Main success scenario	1. User กดไปที่ชุมชนที่สนใจ 2. ระบบแสดงรายละเอียดชุมชนที่ค้นหา

ตารางที่ 3.9 การอ่านรายละเอียดชุมชน

### การสมัครชุมชน

Actor	User
Goal	เข้าร่วมชุมชน
Pre-conditions	- User ต้องเข้าสู่ระบบก่อน - User ต้องเข้าสู่หน้าอ่านรายละเอียดชุมชนก่อน
Main success scenario	User กดเข้าร่วมชุมชน

ตารางที่ 3.10 การสมัครชุมชน

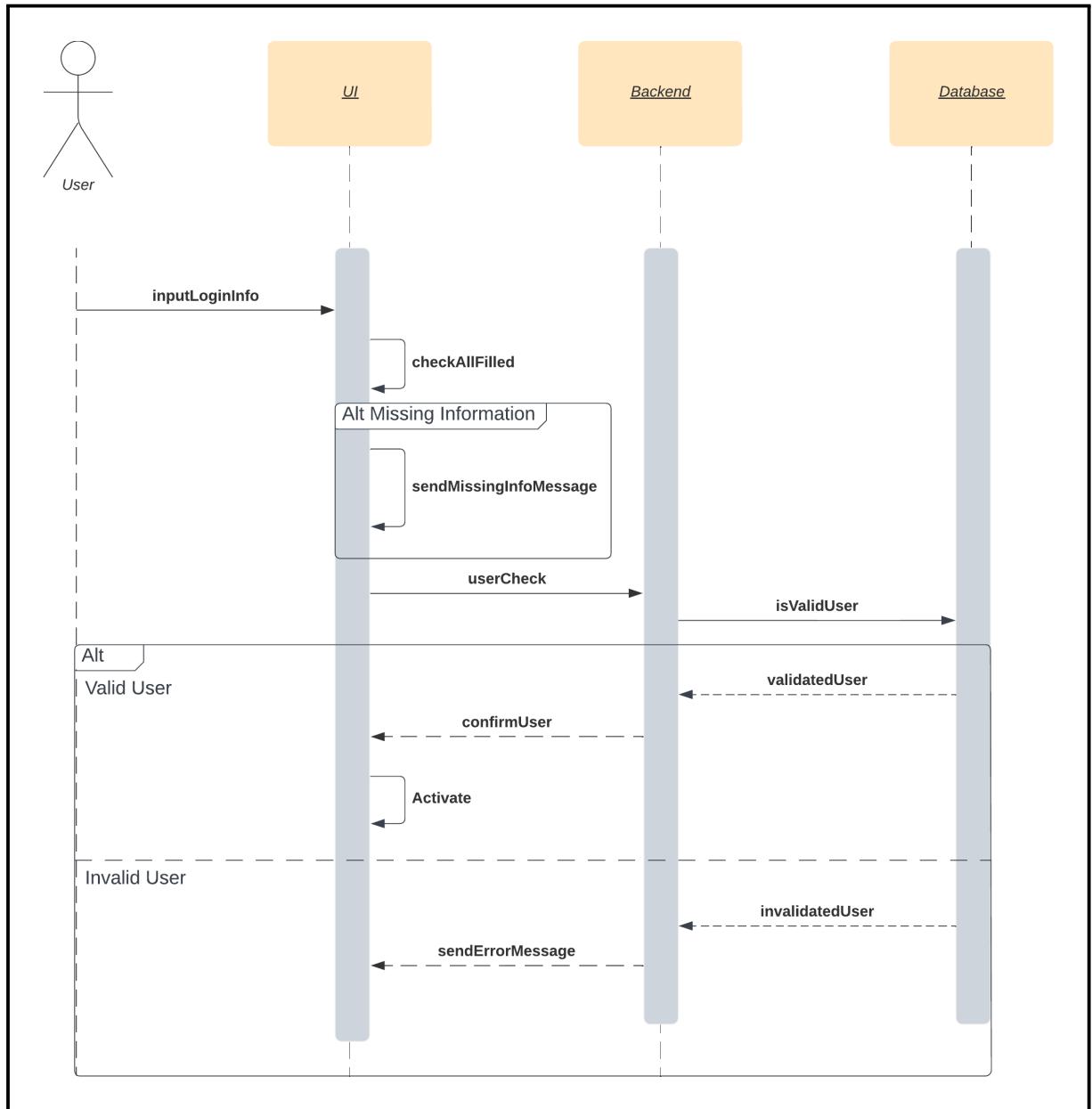
### การลากออกจากรายละเอียดชุมชน

Actor	User
Goal	เข้าร่วมชุมชน
Pre-conditions	- User ต้องเข้าสู่ระบบก่อน - User ต้องเข้าสู่หน้าอ่านรายละเอียดชุมชนก่อน
Main success scenario	User กดเข้าร่วมชุมชน

ตารางที่ 3.11 การลากออกจากรายละเอียดชุมชน

### 3.6 Sequence diagram

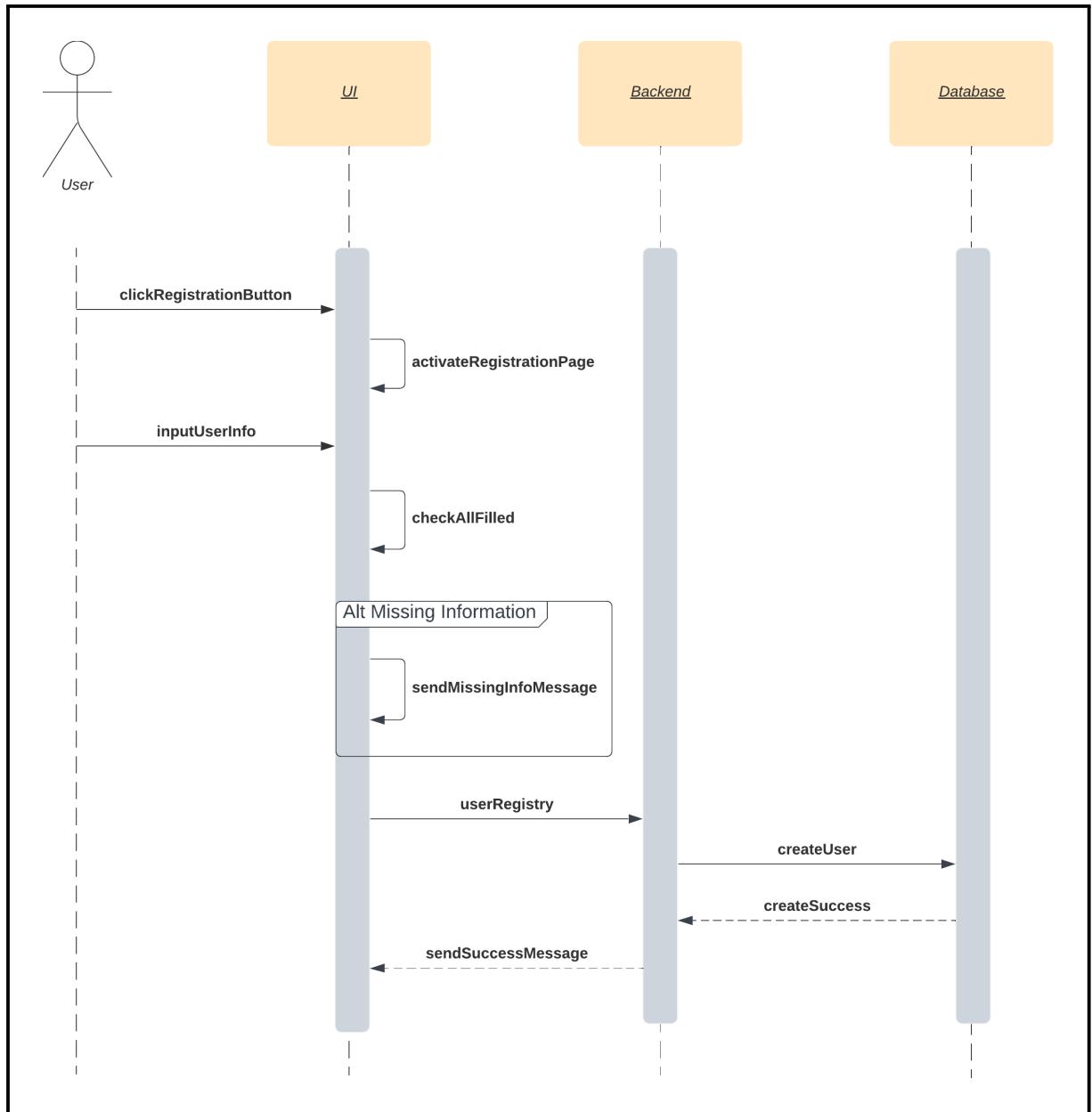
#### 3.6.1 การเข้าสู่ระบบ



รูปที่ 3.3: Sequence diagram ของการเข้าสู่ระบบ

จาก Sequence diagram ในรูปที่ 3.3 เป็น Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานทำการเข้าสู่ระบบ โดยให้ผู้ใช้งานกรอกข้อมูลที่ใช้ในการเข้าสู่ระบบลงในหน้า log in ของ UI ตัว UI จะทำการตรวจสอบข้อมูลที่ต้องการใช้ในการเข้าสู่ระบบว่ามีส่วนใดที่ขาด จากนั้นจึงส่งต่อให้ระบบ Backend ที่จะเรียกข้อมูลจาก Data base เพื่อตรวจสอบผู้ใช้งาน แล้วจึงเปิดหน้า UI ในกรณีที่ยืนยันได้ว่ามีผู้ใช้งานนั้นอยู่จริง

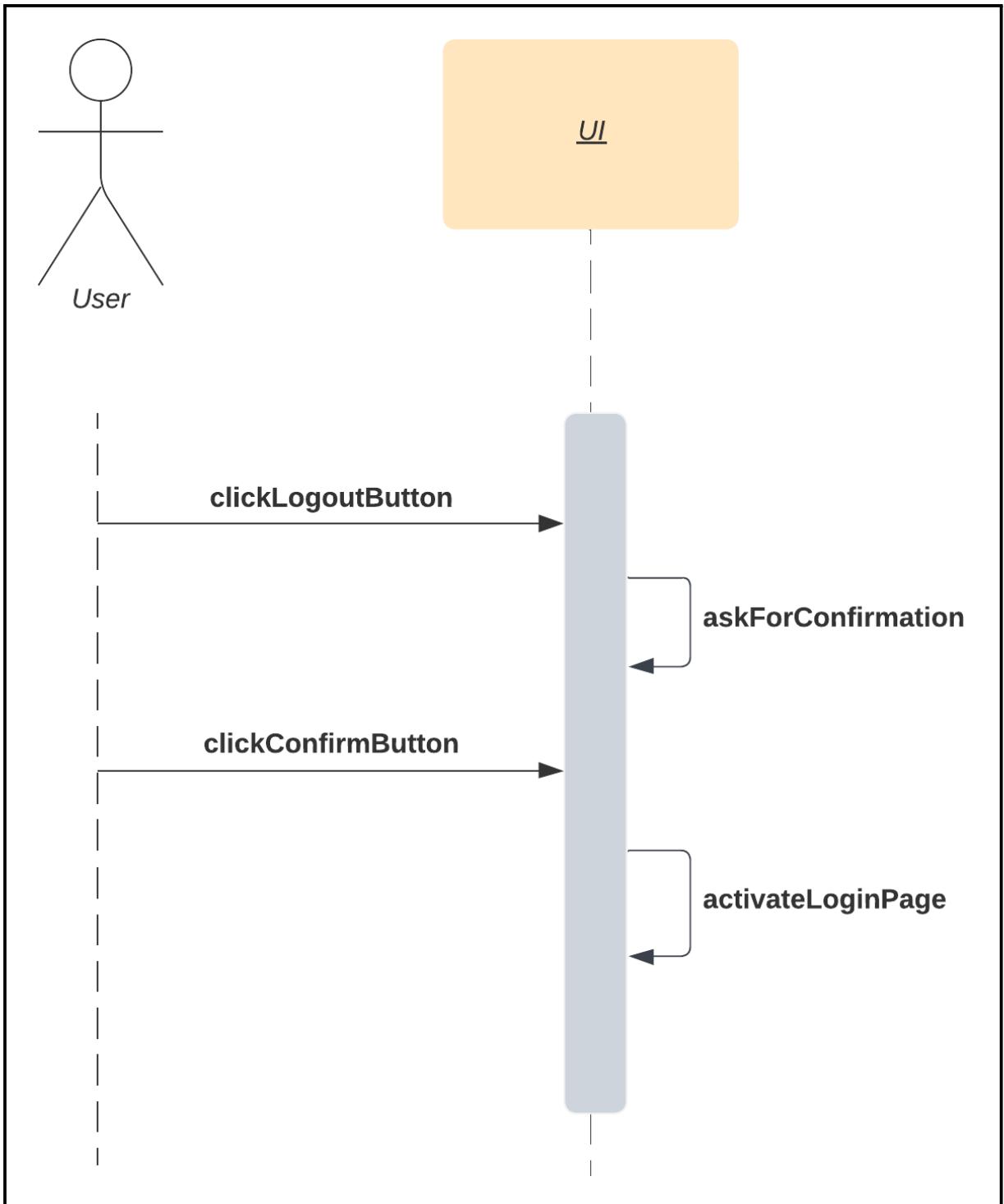
### 3.6.2 การสมัครเข้าใช้งาน



รูปที่ 3.4: การสมัครเข้าใช้งาน

จาก Sequence diagram ในรูปที่ 3.4 เป็น Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานทำการสมัครเข้าใช้งาน โดยหลังจากผู้ใช้งานกดตุ่มเพื่อสมัครใช้งานแล้วระบบ UI จะขึ้นหน้าให้กรอกข้อมูลที่ใช้ในการสมัครเข้าใช้งาน ตัว UI จะทำการตรวจข้อมูลที่ต้องการใช้ในการเข้าสู่ระบบว่ามีส่วนใดที่ขาด จากนั้นจึงส่งต่อให้ระบบ Backend ซึ่งจะส่งข้อมูลไปบันทึกที่ Data base และระบบจะส่งข้อความยืนยันการสมัครเข้าใช้งานกลับขึ้นไปหาผู้ใช้งานที่สมัคร

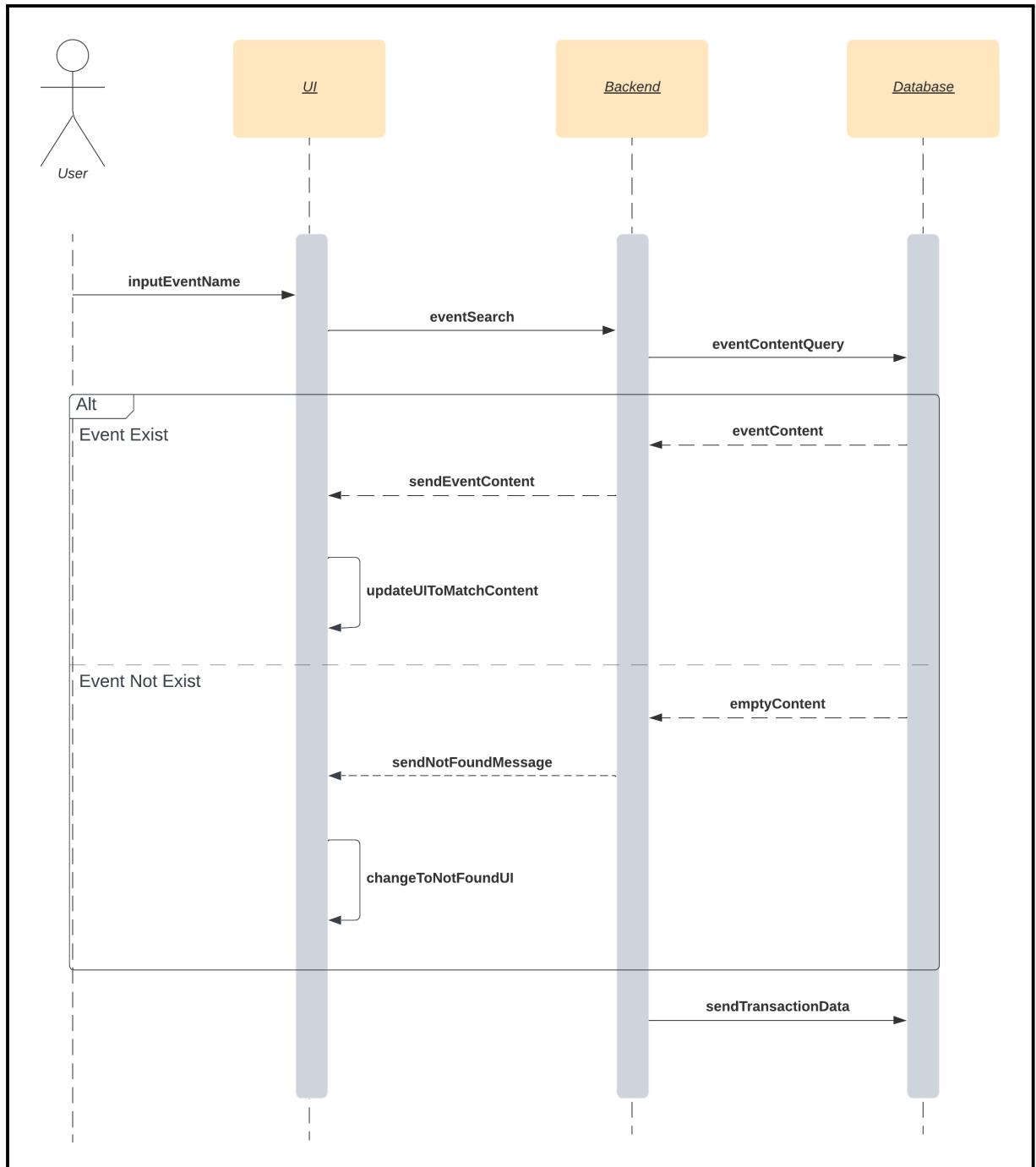
### 3.6.3 การออกจากระบบ



รูปที่ 3.5: การออกจากระบบ

จาก Sequence diagram ในรูปที่ 3.5 เป็น Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานออกจากระบบ โดยหลังจากผู้ใช้งานกดปุ่มออกจากระบบ UI จะทำการยืนยันการออกจากระบบซึ่งหลังจากผู้ใช้งานกดยืนยันแล้วระบบจะพาผู้ใช้ไปยังหน้าเข้าสู่ระบบอีกครั้ง

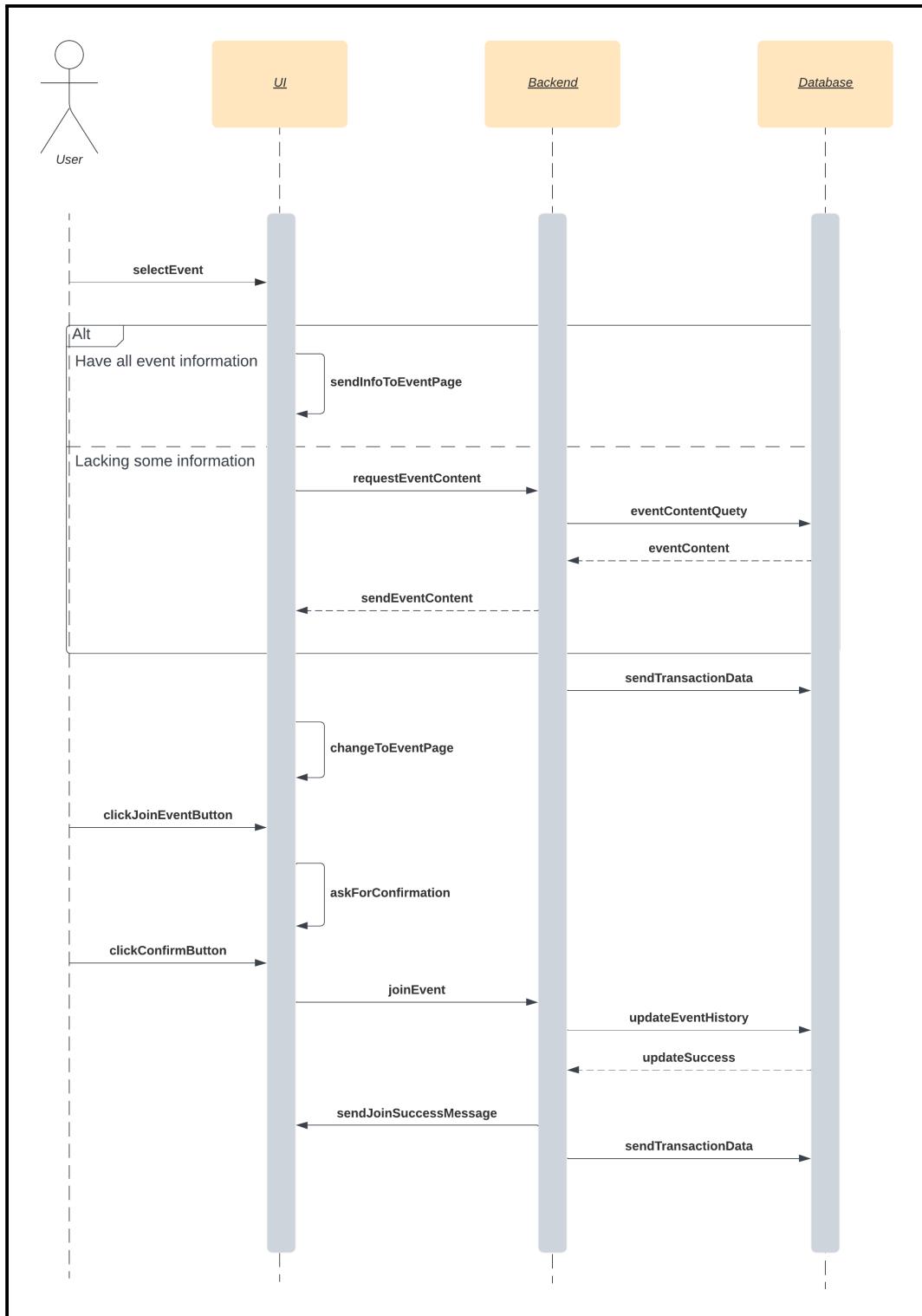
### 3.6.4 การค้นหาภารกิจกรรม



รูปที่ 3.6: การค้นหาภารกิจกรรม

จาก Sequence diagram ในรูปที่ 3.6 เป็น Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานทำการค้นหาภารกิจกรรม โดย UI จะรับสิ่งที่ผู้ใช้งานต้องการค้นหาส่งไปหา Backend เพื่อให้ Backend ค้นหาข้อมูลที่เก็บไว้จาก Data base หากพบเจอจะส่งข้อมูลกลับไปหาผู้ใช้งาน หากไม่เจอจะแสดงข้อความว่าไม่พบภารกิจกรรม และส่งข้อมูลการใช้งานไปเก็บยัง Data base

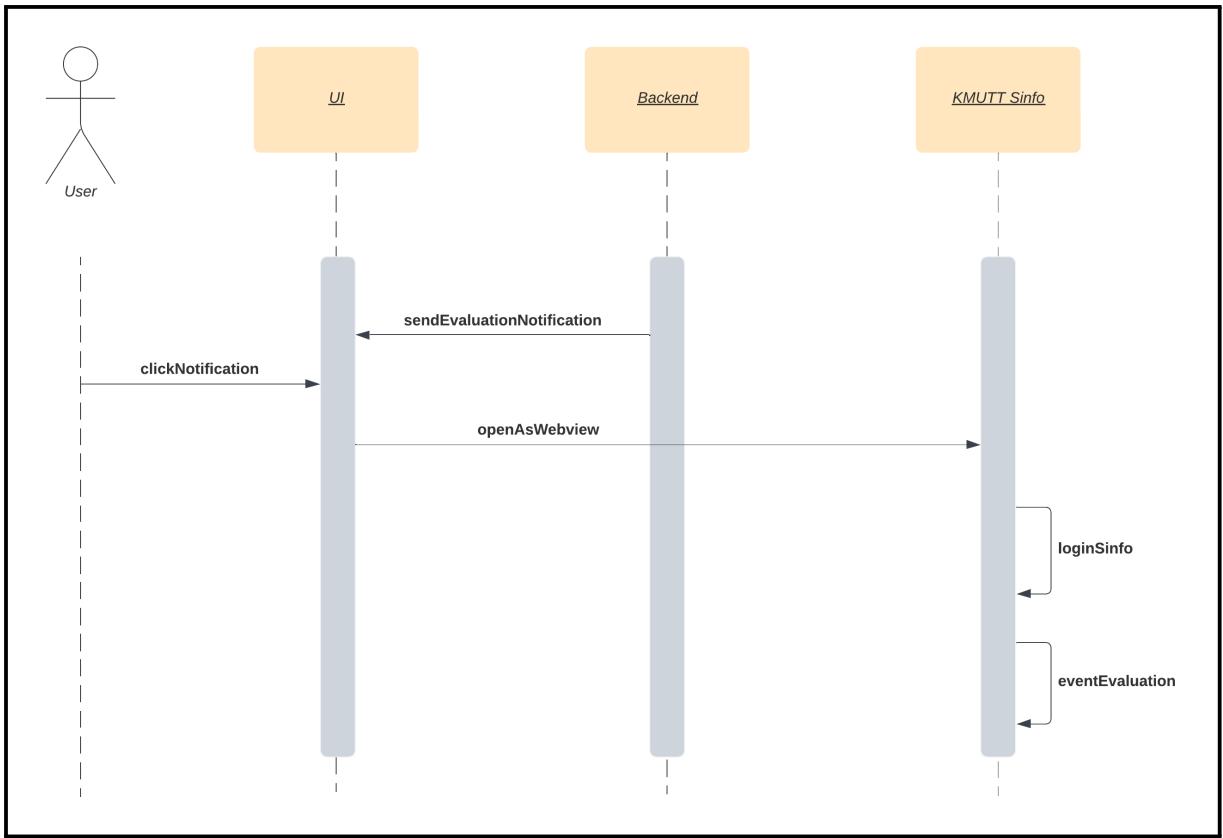
### 3.6.5 การอ่านรายละเอียดของกิจกรรมและเข้าร่วมกิจกรรม



รูปที่ 3.7: การอ่านรายละเอียดของกิจกรรมและเข้าร่วมกิจกรรม

จาก Sequence diagram ในรูปที่ 3.7 เป็น Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานทำการเข้าไปอ่านรายละเอียดของกิจกรรม โดย UI จะแสดงข้อมูลหากว่ามีข้อมูลของกิจกรรมนั้นอยู่แล้ว หากไม่มี จะส่งคำขอให้ Back end ดึงข้อมูลจาก Data base ให้ หากผู้ใช้งานทำการกดเข้าร่วมกิจกรรม UI จะทำการขอการยืนยันจากผู้ใช้งานอีกครั้ง ซึ่งหากกดยืนยันแล้วก็จะส่งข้อมูลการเข้าร่วมไปให้ Backend จัดเก็บใน Data base

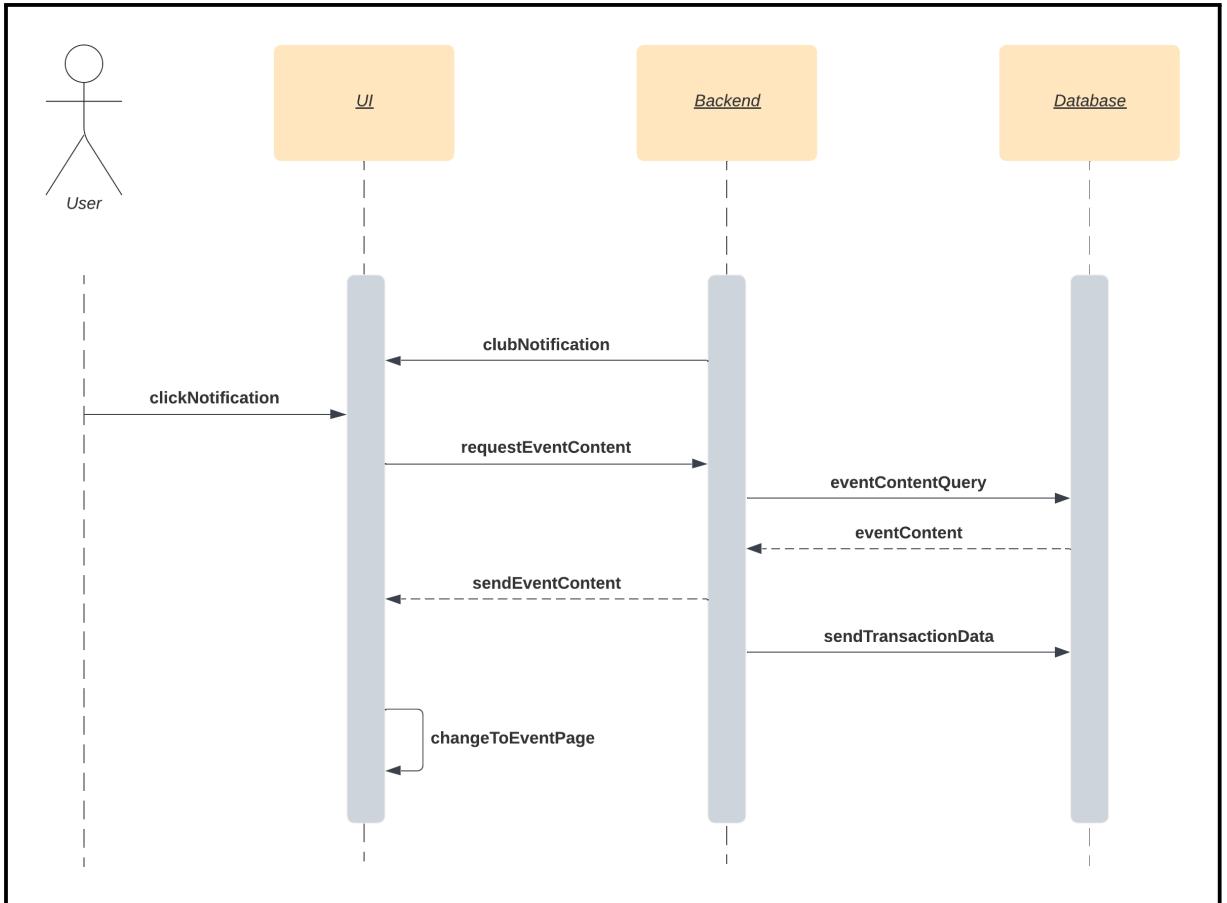
### 3.6.6 การประเมินกิจกรรม



รูปที่ 3.8: การประเมินกิจกรรม

จาก Sequence diagram ในรูปที่ 3.8 เป็น Sequence diagram สำหรับแสดงการทำงานของระบบแจ้งเตือนการประเมินกิจกรรม โดย Backend จะส่งคำเตือนเข้ามาหา UI เพื่อที่จะแจ้งเตือนการประเมินกิจกรรม เมื่อผู้ใช้งานกดเข้าลิงค์ประเมินกิจกรรมแล้วจะเปิดหน้าต่างเพื่อประเมินกิจกรรมให้

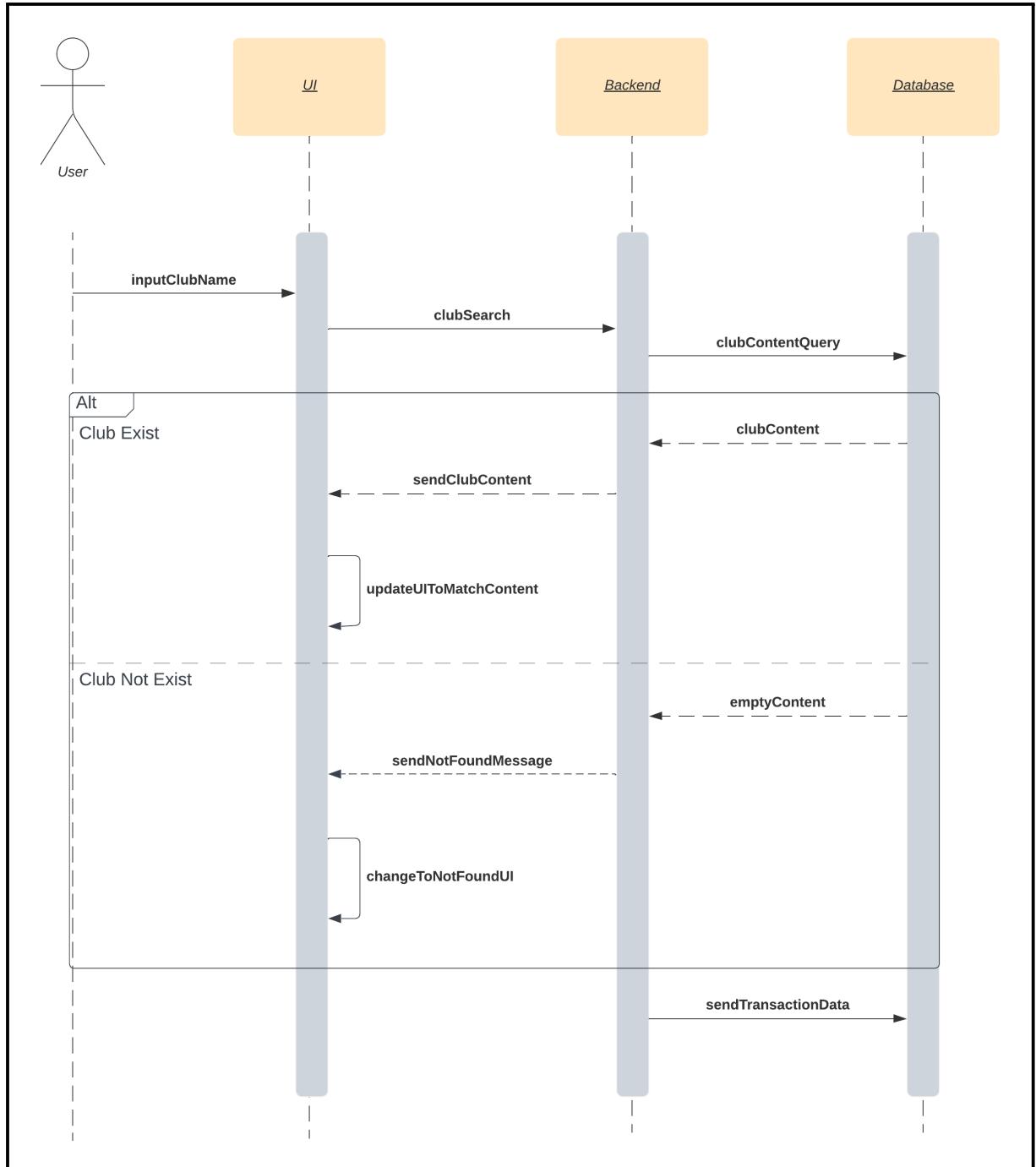
### 3.6.7 การแจ้งเตือนกิจกรรมชุมชน



รูปที่ 3.9: การแจ้งเตือนกิจกรรมชุมชน

จาก Sequence diagram ในรูปที่ 3.9 เป็น Sequence diagram สำหรับแสดงการทำงานของระบบแจ้งเตือนกิจกรรมชุมชน โดย Backend จะส่งคำเตือนเข้ามาหา UI เพื่อที่จะแจ้งถึงกิจกรรมที่ชุมชนจัดขึ้น เมื่อผู้ใช้งานกดเข้าอ่านรายละเอียดกิจกรรมแล้ว UI จะขอให้ Backend หาข้อมูลกิจกรรมจาก Data base มาแสดงให้ผู้ใช้งาน และส่งข้อมูลการใช้งานไปเก็บยัง Data base

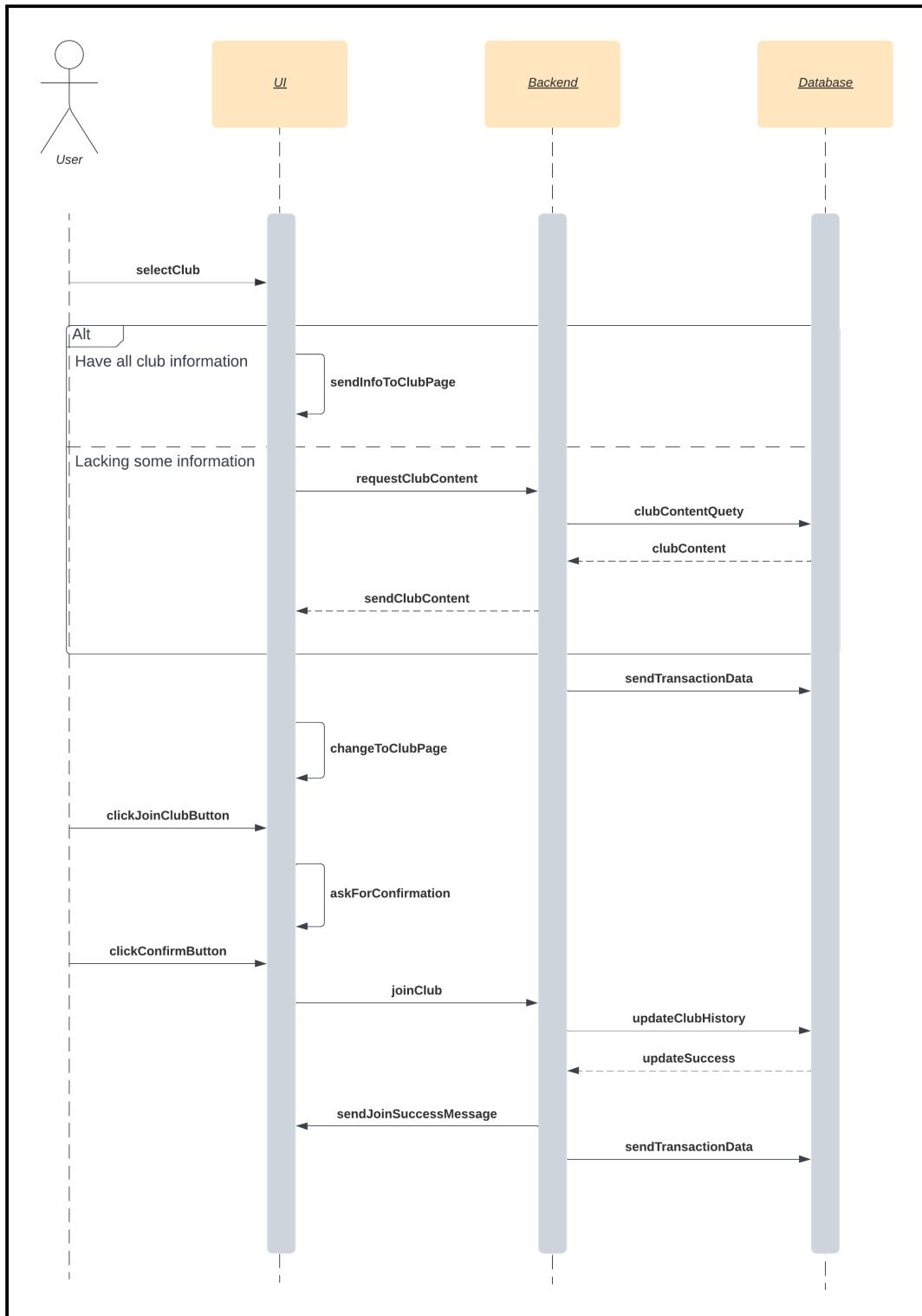
### 3.6.8 การค้นหาชุมชน



รูปที่ 3.10: การค้นหาชุมชน

จาก Sequence diagram ในรูปที่ 3.10 เป็น Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานทำการค้นหาชุมชนโดย UI จะรับสิ่งที่ผู้ใช้งานต้องการค้นหาส่งไปหา Backend เพื่อให้ Backend ค้นหาข้อมูลที่เก็บไว้จาก Data base หากพบเจอจะส่งข้อมูลกลับไปหาผู้ใช้งาน หากไม่เจอจะแสดงข้อความว่าไม่พบกิจกรรม และส่งข้อมูลการใช้งานไปเก็บยัง Data base

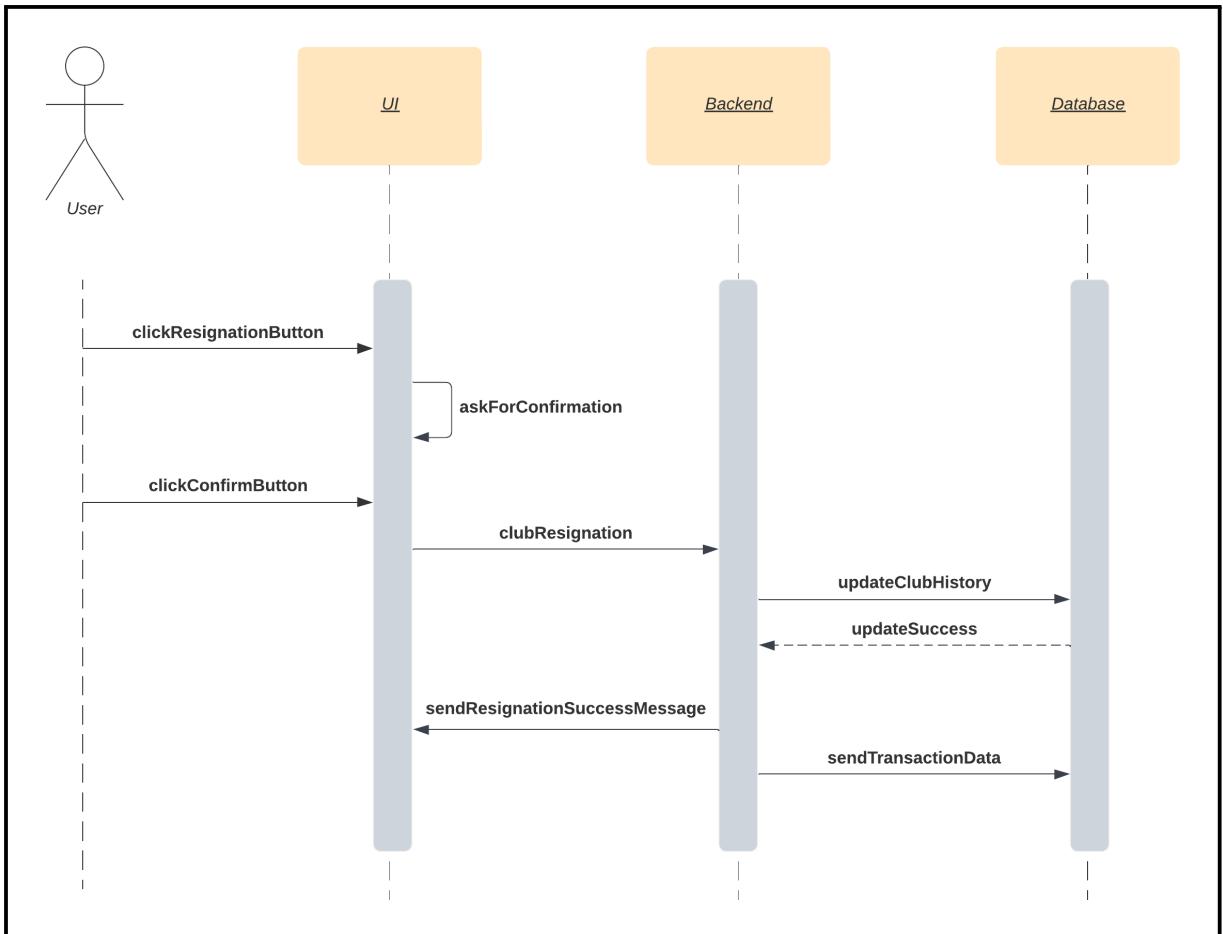
### 3.6.9 อ่านรายละเอียดของชุมชนและการเข้าร่วมชุมชน



รูปที่ 3.11: อ่านรายละเอียดของชุมชนและการเข้าร่วมชุมชน

จาก Sequence diagram ในรูปที่ 3.11 เป็น Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานทำการเข้าไปอ่านรายละเอียดของชุมชน โดย UI จะแสดงข้อมูลหากมีข้อมูลของกิจกรรมนั้นอยู่แล้ว หากไม่มีจะส่งคำขอให้ Back end ดึงข้อมูลจาก Data base ให้ หากผู้ใช้งานทำการกดเข้าร่วมกิจกรรม UI จะทำการขอการยืนยันจากผู้ใช้งานอีกครั้ง ซึ่งหากกดยืนยันแล้วก็จะส่งข้อมูลการเข้าร่วมไปให้ Backend จัดเก็บใน Data base

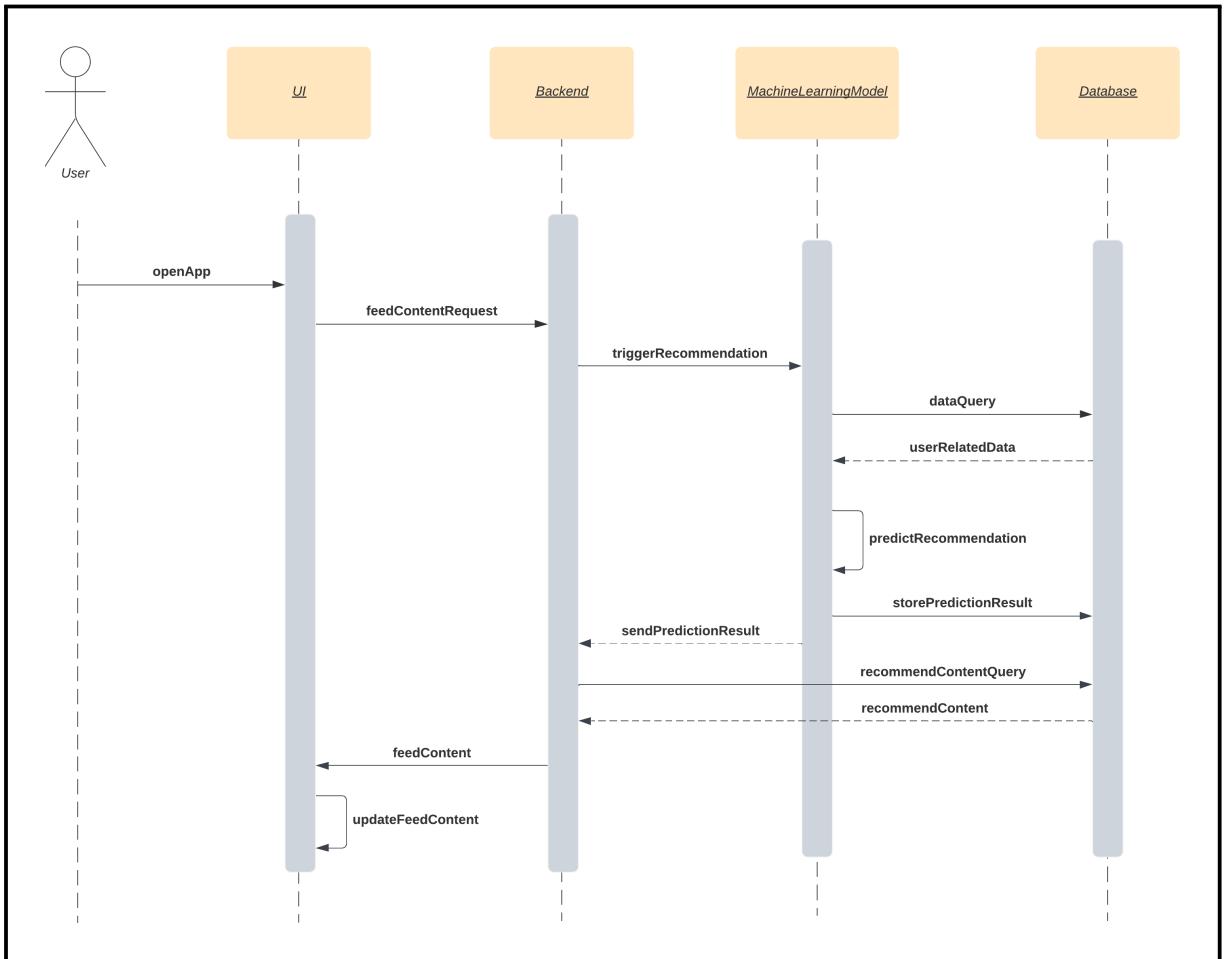
### 3.6.10 การลาออกจากชุมชน



รูปที่ 3.12: การลาออกจากชุมชน

จาก Sequence diagram ในรูปที่ 3.12 เป็น Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานทำการลาออกจากชุมชน โดยหลังจากกดปุ่มสมัคร UI จะถามยืนยันการลาออกจาก หากผู้ใช้งานกดยืนยัน UI จะทำการส่งข้อมูลการใช้งานไปให้ Backend จัดเก็บใน Data base และส่งข้อความยืนยันการลาออกจาก UI และกับผู้ใช้งาน

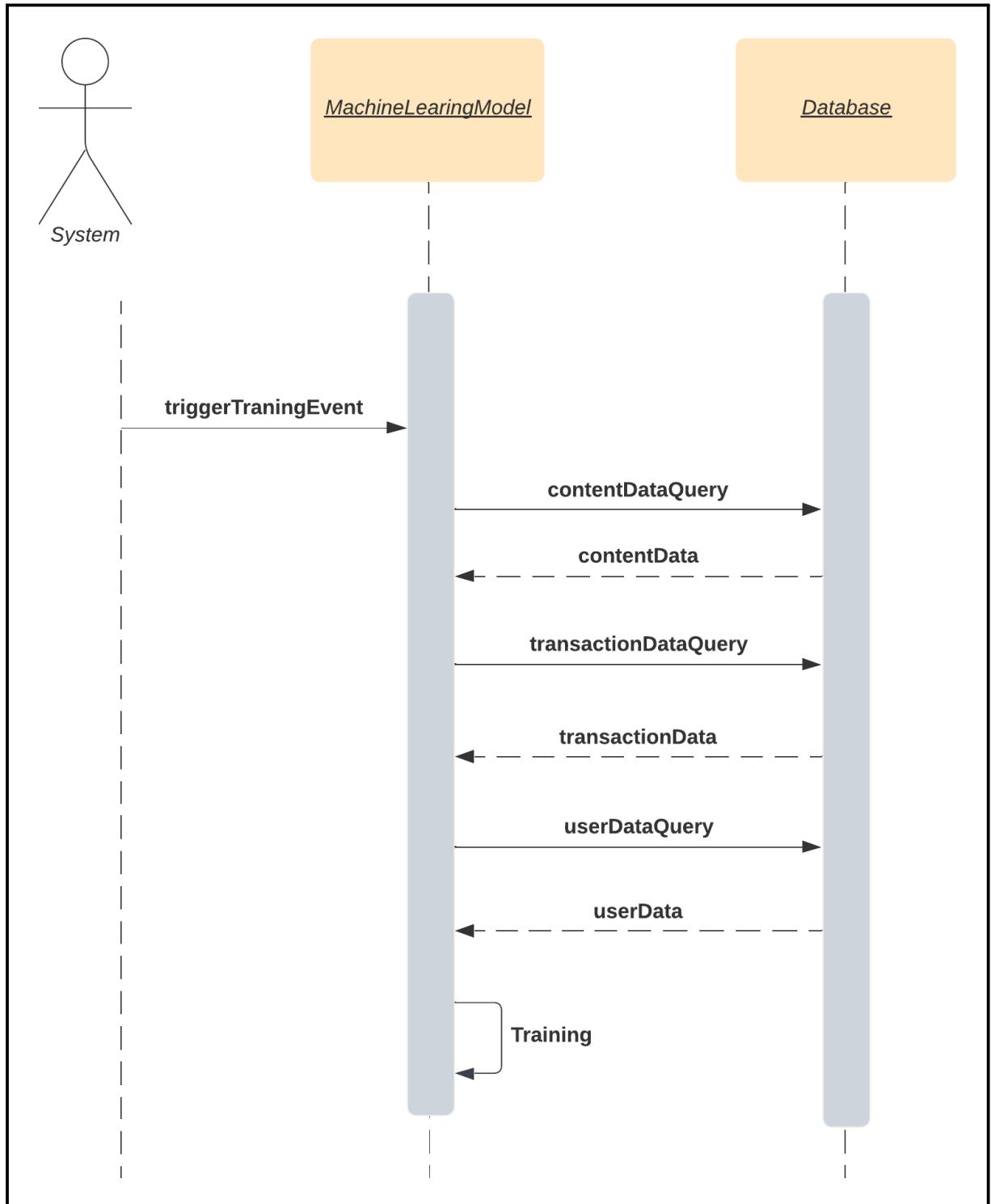
### 3.6.11 การแนะนำกิจกรรมให้ผู้ใช้งาน



รูปที่ 3.13: การแนะนำกิจกรรมให้ผู้ใช้งาน

จาก Sequence diagram ในรูปที่ 3.13 เป็น Sequence diagram สำหรับแสดงการทำงานของระบบเพื่อแนะนำกิจกรรมให้ผู้ใช้งาน โดยหลังจากเข้าแอปพลิเคชัน UI จะร้องขอเนื้อหาที่จะแนะนำไปยัง Backend เพื่อไปสั่งระบบ Machine learning Model เพื่อประมวลผลคำแนะนำซึ่งดึงข้อมูลที่เก็บไว้จาก Database มาประมวลผล ซึ่งจะส่งทั้งคำแนะนำกลับไปหา ทั้ง Backend เพื่อส่งคำแนะนำไปหาผู้ใช้งาน และ Data base เพื่อเก็บผลคำแนะนำที่ประมวลมาได้ Backend สามารถเรียกคำแนะนำที่ถูกเก็บเอาไว้มาเพื่อแสดงแก่ผู้ใช้งานได้

### 3.6.12 การ Train Model การเรียนรู้ของเครื่อง

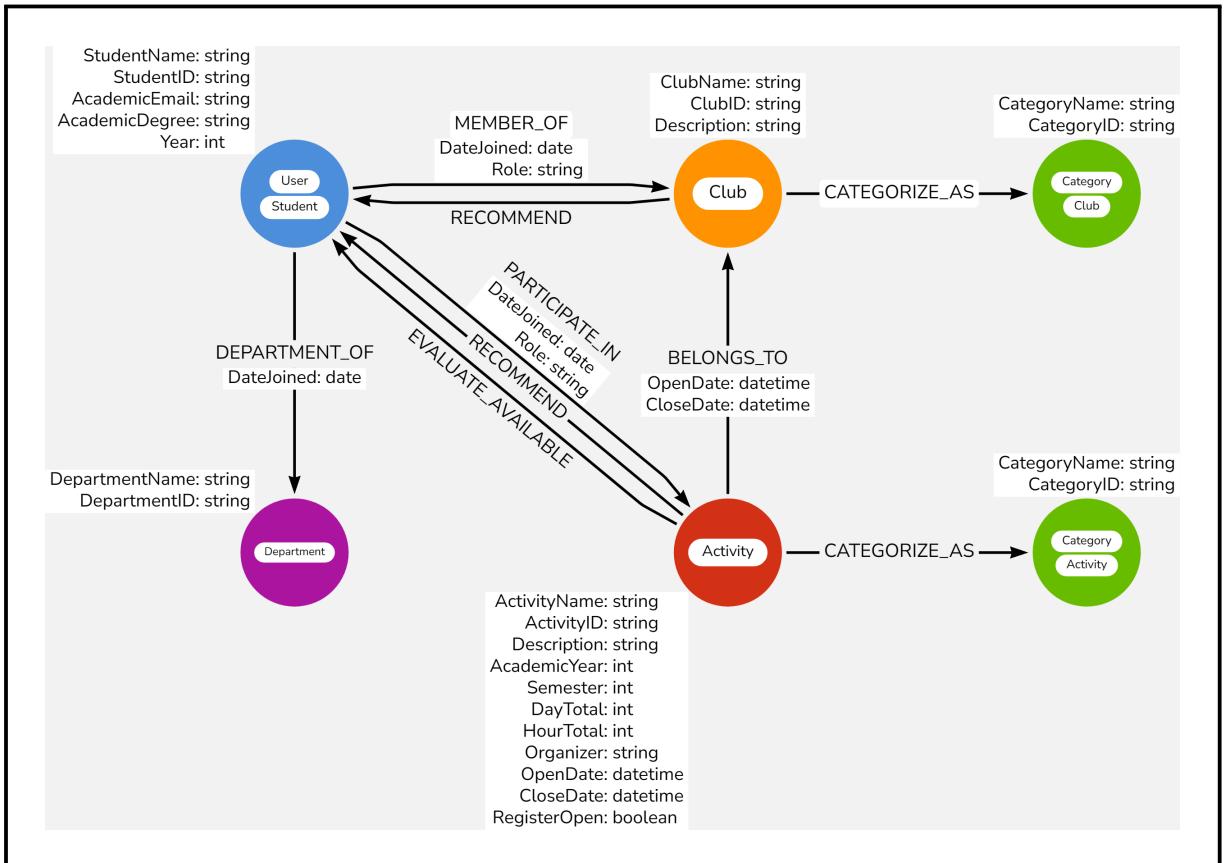


รูปที่ 3.14: การ Train Model การเรียนรู้ของเครื่อง

จาก Sequence diagram ในรูปที่ 3.14 เป็น Sequence diagram สำหรับแสดงการทำงานของระบบในขณะที่ Train Model การเรียนรู้ของเครื่องใหม่ โดย Model การเรียนรู้ของเครื่องจะดึงข้อมูลกิจกรรมและการใช้งานมาจาก Data base แล้วทำการ Train ตัวเอง

### 3.7 แบบจำลองข้อมูลแบบกราฟ (Graph Data Model)

แบบจำลองข้อมูลแบบกราฟเป็นแบบจำลองเพื่อแสดงความเชื่อมโยงของโหนดที่เก็บข้อมูลของระบบ ว่าแต่ละโหนดมีความสัมพันธ์กันอย่างไรภายในแอปพลิเคชัน Actiwiz ของเรา แบบจำลองข้อมูลแบบกราฟจะตอบโจทย์การนำไปประยุกต์ในระบบฐานข้อมูลแบบกราฟ เช่น Neo4j ช่วยให้สามารถสำรวจและเรียกคืนข้อมูลที่เกี่ยวข้องได้อย่างมีประสิทธิภาพ



รูปที่ 3.15: Graph Data Model

จากรูปที่ 3.15 จะเป็นแบบจำลองข้อมูลแบบกราฟของระบบ Actiwiz ซึ่งประกอบไปด้วยโหนด (Node) และความสัมพันธ์ (Relationship) ต่าง ๆ ดังนี้

#### 3.7.1 โหนด

จากที่กล่าวมาโหนดจะทำหน้าที่ในการเก็บข้อมูลซึ่งจะแบ่งข้อมูลในแต่ละโหนดออกมาให้เป็นตารางได้ตามนี้

##### 3.7.1.1 โหนด User: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดส่วนตัวของผู้ใช้งานภายในมหาวิทยาลัย

Property	DataType	Constrain	Definition
StudentName	STRING	Not NULL	ชื่อเต็มของผู้ใช้
StudentID	STRING	Not NULL	รหัสนักศึกษาของผู้ใช้
AcademicEmail	STRING	Not NULL	อีเมลมหาวิทยาลัย
AcademicDegree	STRING	Not NULL	ระดับการศึกษา
Year	INTEGER	Not NULL	ชั้นปีของนักศึกษา

ตารางที่ 3.12 ตารางเก็บข้อมูลของผู้ใช้งานที่ใช้งานแอปพลิเคชัน

3.7.1.2 โหนด Department: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดของภาควิชาภายในมหาวิทยาลัย

Property	DataType	Constrain	Definition
DepartmentName	STRING	Not NULL	ชื่อภาควิชา
DepartmentID	STRING	Not NULL	ชื่อย่อภาษาอังกฤษของภาควิชา
Description	STRING	NULL	คำอธิบายชั้นเรียน

ตารางที่ 3.13 ตารางเก็บข้อมูลของภาควิชาภายในมหาวิทยาลัย

3.7.1.3 โหนด Club: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดของชมรมภายในมหาวิทยาลัย

Property	DataType	Constrain	Definition
ClubName	STRING	Not NULL	ชื่อชมรม
ClubID	STRING	Not NULL	รหัสชมรม
Description	STRING	NULL	คำอธิบายชมรม

ตารางที่ 3.14 ตารางเก็บข้อมูลของชมรมภายในมหาวิทยาลัย

3.7.1.4 โหนด Club Category: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดประเภทชมรมภายในมหาวิทยาลัย

Property	DataType	Constrain	Definition
CategoryName	STRING	Not NULL	ชื่อประเภท
CategoryID	STRING	Not NULL	รหัสประเภท
Description	STRING	NULL	คำอธิบายชมรม
AcademicDegree	STRING	Not NULL	ระดับการศึกษา
Year	INTEGER	Not NULL	ชั้นปีของนักศึกษา

ตารางที่ 3.15 ตารางเก็บข้อมูลประเภทชมรม

### 3.7.1.5 โหนด Activity: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดของชั้นเรียนภายในมหาวิทยาลัย

Property	DataType	Constraint	Definition
ActivityName	STRING	Not NULL	ชื่อประเภท
ActivityID	STRING	Not NULL	รหัสประเภท
Description	STRING	NULL	คำอธิบายกิจกรรม
AcademicYear	INTEGER	Not NULL	ปีการศึกษา
Semester	INTEGER	Not NULL	ภาคการศึกษา
DayTotal	INTEGER	Not NULL	จำนวนวันของกิจกรรม
HourTotal	INTEGER	Not NULL	จำนวนชั่วโมงกิจกรรม
Organizer	STRING	Not NULL	ชื่อผู้จัดกิจกรรม
OpenDate	DATETIME	Not NULL	วันแรกของกิจกรรม
CloseDate	DATETIME	Not NULL	วันสุดท้ายของกิจกรรม
RegisterOpen	BOOLEAN	Not NULL	สถานะของการเปิดลงทะเบียนเข้าร่วมกิจกรรม

ตารางที่ 3.16 ตารางเก็บข้อมูลของชั้นเรียนภายในมหาวิทยาลัย

### 3.7.1.6 โหนด Activity Category: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับประเภทของกิจกรรมภายในมหาวิทยาลัย

Property	DataType	Constraint	Definition
CategoryName	STRING	Not NULL	ชื่อประเภท
CategoryID	STRING	Not NULL	รหัสประเภท

ตารางที่ 3.17 ตารางเก็บข้อมูลประเภทของกิจกรรม

## 3.7.2 ความสัมพันธ์

ความสัมพันธ์ภายในแบบจำลองข้อมูลแบบกราฟของแอปพลิเคชัน Actiwiz เป็นการสร้างความเชื่อมโยงและการมีปฏิสัมพันธ์ระหว่างโหนดต่าง ๆ ซึ่งมีส่วนช่วยในการนำเสนอเรื่องราวการเข้าร่วมกิจกรรมและชั้นเรียนภายในมหาวิทยาลัยแบบบีดหยุ่นและเชื่อมโยงถึงกัน โดยความสัมพันธ์จะมีอยู่ 2 รูปแบบด้วยกัน ได้แก่ ความสัมพันธ์ที่มีคุณสมบัติที่บอกข้อมูลเพิ่มเติมและความสัมพันธ์ที่ไม่มีคุณสมบัติเพิ่มเติม

### 3.7.2.1 ความสัมพันธ์ที่มีคุณสมบัติ

- User is in DEPARTMENT\_OF Department: ความสัมพันธ์ DEPARTMENT\_OF แสดงความสัมพันธ์ทางวิชาการของผู้ใช้งานกับภาควิชาของตน ความสัมพันธ์นี้ถ่ายทอดโดยผู้คนจากภาควิชาเดียวกันภายในมหาวิทยาลัย

Property	DataType	Constraint	Definition
DateJoined	DATE	Not NULL	วันที่เข้าเป็นนักศึกษาของภาควิชา

ตารางที่ 3.18: ตารางเก็บข้อมูลนักศึกษาภาควิชา

- User is a MEMBER\_OF Club: ความสัมพันธ์ MEMBER\_OF แสดงความสัมพันธ์ของผู้ใช้งานว่าเป็นหนึ่งในสมาชิกของชั้นเรียนได้ชั้นเรียนหนึ่ง

Property	DataType	Constraint	Definition
DateJoined	DATE	Not NULL	วันที่เริ่มเป็นสมาชิกของชั้นเรียน
Role	STRING	Not NULL	ตำแหน่งในชั้นเรียน

ตารางที่ 3.19: ตารางเก็บข้อมูลสมาชิกของชั้นเรียน

- User PARTICIPATE\_IN Activity: ความสัมพันธ์ PARTICIPATES\_IN แสดงถึงการเข้าร่วมกิจกรรมของผู้ใช้งานในกิจกรรมต่าง ๆ ภายในมหาวิทยาลัย

DateJoined	DataType	Constraint	Definition
DateJoined	DATE	Not NULL	วันที่เข้าร่วมกิจกรรม
Role	STRING	Not NULL	ตำแหน่งในกิจกรรม

ตารางที่ 3.20: ตารางเก็บข้อมูลสมาชิกเข้าร่วมกิจกรรม

- Activity BELONGS\_TO Club: ความสัมพันธ์ BELONGS\_TO แสดงความสัมพันธ์ระหว่างกิจกรรมกับชมรม โดยระบุว่าชมรมได้รับผิดชอบในการจัดกิจกรรมใดกิจกรรมหนึ่ง

Property	DataType	Constraint	Definition
OpenDate	DATETIME	Not NULL	วันแรกของกิจกรรม
CloseDate	DATETIME	Not NULL	วันสุดท้ายของกิจกรรม

ตารางที่ 3.21: ตารางเก็บข้อมูลกิจกรรมถูกจัดโดยชมรม

### 3.7.2.2 ความสัมพันธ์ที่ไม่มีคุณสมบัติ

- Activity CATEGORIZE\_AS Activity Category: ความสัมพันธ์ CATEGORIZE\_AS จัดกลุ่มกิจกรรมด้วยการจัดประเภทของกิจกรรม ตามหมวดหมู่เดียวกัน
- Activity RECOMMEND to User: ความสัมพันธ์ RECOMMEND แสดงถึงการแนะนำกิจกรรมที่สร้างโดยระบบของแอปพลิเคชันโดยเกิดจากการประมวลผลของ machine learning ที่ปรับให้เหมาะสมตามผู้ใช้งานแต่ละคน
- Activity EVALUATE\_AVAILABLE to User: ความสัมพันธ์แบบ EVALUATE\_AVAILABLE จะเชื่อมโยงผู้ใช้งานกับกิจกรรมต่าง ๆ เมื่อกิจกรรมเปิดให้ผู้ใช้งานเข้าไปประเมินกิจกรรมได้
- Club CATEGORIZE\_AS Club Category: ความสัมพันธ์ CATEGORIZE\_AS จัดกลุ่มชมรมด้วยการจัดประเภทของชมรม ตามหมวดหมู่เดียวกัน
- Club RECOMMEND to User: ความสัมพันธ์ RECOMMEND แสดงถึงการแนะนำชมรมที่สร้างโดยระบบของแอปพลิเคชัน โดยเกิดจากการประมวลผลของ machine learning ที่ปรับให้เหมาะสมตามผู้ใช้งานแต่ละคน

## บทที่ 4 ผลการทดลองและอภิปรายผล

ในบทนี้จะกล่าวถึงผลลัพธ์จากการทดสอบฟังก์ชันหลักของระบบในแอปพลิเคชัน รวมถึงความสามารถในการทำงาน ความหมาย สมในการใช้งาน การตอบสนองต่อผู้ใช้ และความพึงพอใจของผู้ใช้

### 4.1 การทดสอบการทำงาน (Functional Testing)

Feature	Scenario	ผลที่คาดว่าจะได้รับ	ผลที่ได้รับ
Sign up	ผู้ใช้งานไม่กรอกอีเมล	ระบบแจ้งเตือน "กรุณากรอกอีเมล"	
	ผู้ใช้งานไม่กรอกรหัสผ่าน	ระบบแจ้งเตือน "กรุณากรอกรหัสผ่าน"	
	ผู้ใช้งานไม่กรอกยืนยันรหัสผ่าน	ระบบแจ้งเตือน "กรุณากรอกยืนยันรหัสผ่าน"	
Sign In	ผู้ใช้งานไม่กรอกอีเมล	ระบบแจ้งเตือน "กรุณากรอกอีเมล"	
	ผู้ใช้งานไม่กรอกรหัสผ่าน	ระบบแจ้งเตือน "กรุณากรอกรหัสผ่าน"	
	ผู้ใช้งานกรอกอีเมลผิด	ระบบแจ้งเตือน "อีเมลไม่ถูกต้อง"	
	ผู้ใช้งานกรอกรหัสผ่านผิด	ระบบแจ้งเตือน "รหัสผ่านไม่ถูกต้อง"	
Feed page	ผู้ใช้งานกดดูรายละเอียดกิจกรรม	ระบบแสดงหน้ารายละเอียดของกิจกรรม	
	ผู้ใช้งานกดดูรายละเอียดของข้อมูล	ระบบแสดงหน้ารายละเอียดของข้อมูล	
Join Event	ผู้ใช้งานกดเข้าร่วมกิจกรรม	ระบบแสดงหน้าจอว่าผู้ใช้ได้เข้าร่วมกิจกรรมแล้ว	
	ผู้ใช้งานกดยกเลิกการเข้าร่วมกิจกรรม	ระบบแสดงหน้าจอว่าผู้ใช้ได้ยกเลิกการเข้าร่วมกิจกรรมแล้ว	
Join Club	ผู้ใช้งานกดเข้าร่วมชุมชน	ระบบแสดงหน้าจอว่าผู้ใช้ได้เข้าร่วมชุมชนแล้ว	
	ผู้ใช้งานกดยกเลิกการเข้าร่วมชุมชน	ระบบแสดงหน้าจอว่าผู้ใช้ได้ยกเลิกการเข้าร่วมชุมชนแล้ว	
Notification	ผู้ใช้กดเข้าไปดูแจ้งเตือนจากทางระบบ	ระบบแสดงหน้าจอเกี่ยวกับการแจ้งเตือนนั้นๆ เช่น ระบบแสดงหน้าจอการประเมินกิจกรรม	
Evaluate Page	ผู้ใช้กดเพื่อเข้าไปประเมินกิจกรรม	ระบบพาผู้ใช้เข้าสู่หน้า Log-in เพื่อประเมินกิจกรรม	

ตารางที่ 4.1 ตารางการทดสอบการทำงาน

จากตารางที่ 4.1 เป็นรายการสำหรับทดสอบการทำงานของฟังก์ชันต่างๆ ในแอปพลิเคชัน เพื่อทดสอบว่าการตอบสนองต่อการใช้งานต่างๆ เป็นไปตามที่ออกแบบไว้หรือไม่

### 4.2 การทดสอบคุณภาพของระบบ (Non Functional Testing)

### 4.3 ผลการทดสอบ (Test Results)

## บทที่ 5 สรุปผลการดำเนินงาน

หลังจากการออกแบบ พัฒนา และทดสอบการทำงานของแอปพลิเคชันในบทที่ 4 ในบทที่ 5 นี้จะเป็นการกล่าวถึง ผลสรุปการดำเนินงานของโครงการทั้งหมดดังปีการศึกษา 2023 ที่ผ่านมา โดยจะเป็นการสรุปผลกระบวนการทั้งที่ทางผู้จัดทำได้ทำสำเร็จไปแล้ว และกระบวนการที่ไม่สำเร็จ โดยรายละเอียดจะกล่าวตามแต่ละหัวข้อต่อไปนี้

### 5.1 กระบวนการทำงาน (Work Process)

#### 5.1.1 การดำเนินงานในภาคการศึกษาที่ 1

Tasks	Process
จัดทำ Idea Document	เสร็จ
จัดทำ Project Proposal	เสร็จ
จัดทำแผนการดำเนินงานของโครงการ	เสร็จ
ศึกษาวิธีการพัฒนาแอปพลิเคชันส่วน Front-end	เสร็จ
ศึกษาวิธีการพัฒนาแอปพลิเคชันส่วน Back-end	เสร็จ
ศึกษาวิธีการพัฒนาฐานข้อมูล	เสร็จ
ศึกษาวิธีการพัฒนา Machine Learning	เสร็จ
จัดทำ Architecture Diagram	เสร็จ
จัดทำ Use Case Diagram	เสร็จ
จัดทำ Sequence Diagram	เสร็จ
จัดทำ Database Schema	เสร็จ
จัดทำ Application Wireframe	เสร็จ
จัดทำ Application UX/UI	เสร็จ
จัดเตรียมข้อมูล (Data Preprocessing)	เสร็จ

ตารางที่ 5.1 ตารางแสดงผลการดำเนินงานในภาคการศึกษาที่ 1

จากตารางที่ 5.1 เป็นการแสดงรายการของการดำเนินงานที่วางแผนไว้ในภาคการศึกษาที่ 1 และผลของการดำเนินการ

#### 5.1.2 การดำเนินงานในภาคการศึกษาที่ 2

Tasks	Process
พัฒนาแอปพลิเคชันส่วน Front-end	-
พัฒนาแอปพลิเคชันส่วน Back-end	-
พัฒนาฐานข้อมูล	-
พัฒนา Machine Learning	-
ออกแบบ Test Cases	-
ทดสอบการทำงานของแอปพลิเคชัน (Functional Testing)	-
ทดสอบคุณภาพของระบบ (Non Functional Testing)	-

ตารางที่ 5.2 ตารางแสดงผลการดำเนินงานในภาคการศึกษาที่ 2

จากตารางที่ 5.2 เป็นการแสดงรายการของการดำเนินงานที่วางแผนไว้ในภาคการศึกษาที่ 2 และผลของการดำเนินการ

### 5.2 ปัญหาและการแก้ไข (Problems and Solutions)

#### 5.3 แนวทางการพัฒนา (Future Work)

#### 5.4 บทสรุป (Conclusion)

## หนังสืออ้างอิง

1. Toni Vujevic, 2566, “The Mobile App Architecture Guide,” จากเว็บไซต์ <https://decode.agency/article/mobile-app-architecture/>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
2. Jerapa Beamgo Soonsongthanee, 2563, “ออกแบบ RESTful API ยังไงให้ปัง,” จากเว็บไซต์ <https://medium.com/@jerapabeamgosoonsoongthanee/%E0%B8%AD%E0%B8%81%E0%B9%81%E0%B9%81%E0%B8%9A%E0%B8%9A-restful-api-%E0%B8%A2%E0%B8%B1%E0%B8%87%E0%B9%84%E0%B8%87%E0%B9%83%E0%B8%AB%E0%B9%89%E0%B8%9B%E0%B8%B1%E0%B8%87-69d2015bcc31>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
3. Tanabodin Kamol, 2562, “Graph database property,” จากเว็บไซต์ [https://miro.medium.com/v2/resize:fit:4800/format:webp/0\\*lsbatEPo6529Qcxw.jpg](https://miro.medium.com/v2/resize:fit:4800/format:webp/0*lsbatEPo6529Qcxw.jpg), [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
4. Amazon Web Services, “ฐานข้อมูลแบบกราฟและฐานข้อมูลแบบเชิงสัมพันธ์แตกต่างกันอย่างไร,” จากเว็บไซต์ <https://aws.amazon.com/th/compare/the-difference-between-graph-and-relational-database/>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
5. Toni Vujevic, 2566, “Mobile App Architecture Overview,” จากเว็บไซต์ <https://decode.agency/wp-content/uploads/2023/05/Frame-6435.svg>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
6. Toni Vujevic, 2566, “Presentation Layer,” จากเว็บไซต์ <https://decode.agency/wp-content/uploads/2023/05/1.30-Investment-stock-Detail.svg>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
7. Toni Vujevic, 2566, “Business Layer,” จากเว็บไซต์ <https://decode.agency/wp-content/uploads/2023/05/Group-6376.svg>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
8. Rufai Mustapha, 2566, “HTTP Request-Response Cycle,” จากเว็บไซต์ <https://www.freecodecamp.org/news/content/images/2023/04/http-client-server-RR.jpg>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
9. Tao, “Contrast,” จากเว็บไซต์ [https://blog.clicknext.com/wp-content/uploads/2020/12/cn\\_crap-01-1024x535.jpg](https://blog.clicknext.com/wp-content/uploads/2020/12/cn_crap-01-1024x535.jpg), [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
10. Tao, “Repetition,” จากเว็บไซต์ [https://blog.clicknext.com/wp-content/uploads/2020/12/cn\\_crap-02jpg-1024x535.jpg](https://blog.clicknext.com/wp-content/uploads/2020/12/cn_crap-02jpg-1024x535.jpg), [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
11. Tao, “Alignment,” จากเว็บไซต์ [https://blog.clicknext.com/wp-content/uploads/2020/12/cn\\_crap-03-1024x535.jpg](https://blog.clicknext.com/wp-content/uploads/2020/12/cn_crap-03-1024x535.jpg), [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
12. Tao, “Proximity,” จากเว็บไซต์ [https://blog.clicknext.com/wp-content/uploads/2020/12/cn\\_crap-04-1024x535.jpg](https://blog.clicknext.com/wp-content/uploads/2020/12/cn_crap-04-1024x535.jpg), [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
13. Michael J. Pazzani and Daniel Billsus, “Content-Based Recommendation Systems,” จากเว็บไซต์ [https://link.springer.com/chapter/10.1007/978-3-540-72079-9\\_10](https://link.springer.com/chapter/10.1007/978-3-540-72079-9_10), [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
14. Badrul Sarwar, George Karypis, Joseph Konstan, , and John Ried, “Item-Based Collaborative Filtering Recommendation Algorithms,” จากเว็บไซต์ <https://www.ra.ethz.ch/cdstore/www10/papers/pdf/p519.pdf>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
15. Prem Melville, Raymond J. Mooney, and Ramadass Nagarajan, “Content-Boosted Collaborative Filtering for Improved Recommendations,” จากเว็บไซต์ <https://www.cs.utexas.edu/~ml/papers/cbcf-aaai-02.pdf>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
16. Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, “Recommender Systems Handbook,” จากเว็บไซต์ <https://link.springer.com/book/10.1007/978-0-387-85820-3>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
17. Melania Berbatova, “Overview on NLP Techniques for Content-Based Recommender Systems for Books,” จากเว็บไซต์ <https://aclanthology.org/R19-2009.pdf>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
18. Oren Barkan and Noam Koenigstein, “ITEM2VEC: Neural item embedding for collaborative filtering,” จากเว็บไซต์ <https://ieeexplore.ieee.org/abstract/document/7738886>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
19. Daniel Valcarcea, Alfonso Landina, and Javier Parapara andAlvaro Barreiroa, “Collaborative Filtering Embeddingsfor Memory-based Recommender Systems,” จากเว็บไซต์ [https://www.researchgate.net/publication/334697349\\_Collaborative\\_filtering\\_embeddings\\_for\\_memory-based\\_recommender\\_systems](https://www.researchgate.net/publication/334697349_Collaborative_filtering_embeddings_for_memory-based_recommender_systems), [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].



- B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B8%A3%E0%B9%88%E0%B8%A7%E0%B8%  
A1%E0%B8%81%E0%B8%B1%E0%B8%9A-vs-code-%E0%B9%80%E0%B8%9A%E0%B8%B7%E0%B9%89%E0%  
B8%AD%E0%B8%87%E0%B8%95%E0%B9%89%E0%B8%99-f848f41a39e9, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน  
2566].
38. stackpython, 2562, “Jest คืออะไร เริ่มต้นเขียน Test ด้วย Jest กันตีก่าว,” จากเว็บไซต์ <https://blog.devahoy.com/blog/2019/08/getting-started-with-testing-and-jest>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].