



PROJECT ACTIWIZ

MR. KUNANON SUPMAMUL 63070501011

MR. NAPHATTARAK MUENTOEY 63070501018

MR. THANADOL THONGRIT 630705010

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

Mr. Kunanon Supmamul 63070501011

Mr. Naphattarak Muentoey 63070501018

Mr. Thanadol Thongrit 630705010

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

Project Advisor

(Assoc.Prof. My main advisor name , Ph.D.)

Project Co-Advisor

(Assoc.Prof. My Co-advisor name, Ph.D.)

Committee Member

(Asst.Prof. Committee2, Ph.D.)

Committee Member

(Asst.Prof. Committee3, Ph.D.)

Project Title	Project Actiwiz
Credits	3
Member(s)	Mr. Kunanon Supmamul 63070501011 Mr. Naphattarak Muentoey 63070501018 Mr. Thanadol Thongrit 630705010
Project Advisor	Assoc.Prof. My main advisor name , Ph.D.
Co-advisor	Assoc.Prof. My Co-advisor name, Ph.D.
Program	Bachelor of Engineering
Field of Study	Computer Engineering
Department	Computer Engineering
Faculty	Engineering
Academic Year	2023

### Abstract

--

**Keywords:** Multihop ad hoc networks / Topology control / Single-Hop Throughput

หัวข้อปริญญาในพนธ์	Actiwiz
หน่วยกิต	3
ผู้เขียน	นาย คุณานันต์ ทรัพย์มามูล นาย ณภัทร์ ก. เหมือนเดย นาย ธนาดล ทองฤทธิ์
อาจารย์ที่ปรึกษา	ผศ. ราชวิชช์ สโรชิวิกสิต ที่ปรึกษา วิทยานิพนธ์ ดร. กิตติพงษ์ ปิยะวรรรณโนน ที่ปรึกษา วิทยานิพนธ์ร่วม
หลักสูตร	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมคอมพิวเตอร์
คณะ	วิศวกรรมศาสตร์
ปีการศึกษา	2566

บทคัดย่อ

--

คำสำคัญ:

## กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์ฉบับนี้ สำเร็จลุล่วงได้อย่างสมบูรณ์ด้วยความกรุณาอย่างยิ่งจาก อาจารย์ราษฎร์ สโตร์วิกสิต และ อาจารย์ กิตติพงษ์ ปิยะวรรณ์โนน ที่ได้สละเวลาอันมีค่าแก่คณะผู้จัดทำ เพื่อให้คำปรึกษาและแนะนำตลอดจน ตรวจทานแก้ไขข้อบกพร่องต่างๆด้วย ความเอาใจใส่เป็นอย่างยิ่ง จนโครงงานวิศวกรรมคอมพิวเตอร์ฉบับนี้สำเร็จสมบูรณ์ลุล่วงได้ด้วยดี คณะผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้จากใจจริง

สุดท้ายนี้ ขออุทิศความดีให้มีในการศึกษา นี้แด่บิดา มารดา ครอบครัวของคณะผู้จัดทำและกำลังใจจากมิตรแท้ทุกท่าน

## สารบัญ

หน้า

ABSTRACT	ii
บทคัดย่อ	iii
กิตติกรรมประกาศ	iv
สารบัญ	v
สารบัญตาราง	vii
สารบัญรูปภาพ	viii
สารบัญสัญลักษณ์	ix
สารบัญคำศัพท์ทางเทคนิคและคำย่อ	x
 บทที่ 1 บทนำ	 1
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
 บทที่ 2 ทฤษฎีความรู้และงานที่เกี่ยวข้อง	 4
2.1 ทฤษฎีและความรู้ที่เกี่ยวข้อง	4
2.1.1 Machine Learning	4
2.1.1.1 Content-Based Filtering	4
2.1.1.2 Collaborative Filtering	4
2.1.1.3 Natural Language Processing	4
2.1.1.4 Word Embedding	4
2.1.1.5 Transformer Model	5
2.1.1.6 Artificial Neural Network	5
2.1.2 Mobile App Architecture	6
2.1.3 HTTP Protocol	9
2.1.3.1 HTTP Protocol คืออะไร	9
2.1.3.2 HTTP Request-Response Cycle	9
2.1.3.3 HTTP Request Methods	9
2.1.4 REST API	9
2.1.4.1 REST คืออะไร	9
2.1.4.2 การออกแบบ REST API	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	14
2.1.5.4 Graph Database vs Relational Database	15
2.1.6 UML Diagram	16
2.1.6.1 UML Diagram คืออะไร	16
2.1.6.2 Use case Diagram	16
2.1.6.3 Sequence Diagram	16

2.2 เทคโนโลยี	16
2.2.1 Integrated Development Environment (IDE)	16
2.2.2 Design	17
2.2.3 Frontend	17
2.2.4 Backend	17
2.2.5 Database	17
2.2.6 Machine Learning	17
2.2.7 Version Control	18
2.2.8 Testing	18
2.2.9 Product Survey	19
<b>บทที่ 3 วิธีการทำงาน กระบวนการและการออกแบบ</b>	<b>26</b>
3.1 บทนำ	26
3.1.1 สำรวจความต้องการของผู้ใช้เชิงคุณภาพ	26
3.1.2 Journey Map	26
3.1.3 Stakeholder	26
3.2 Requirement list	27
3.3 Feature list	27
3.4 Architecture diagram	28
3.5 Use case diagram and use case narratives	30
3.6 Sequence diagram	35
3.6.1 Login	35
3.6.2 Registration	36
3.6.3 Logout	37
3.6.4 Search Event	38
3.6.5 Select and join event	39
3.6.6 Event evaluation	40
3.6.7 Notification about event from club	41
3.6.8 Search club	42
3.6.9 Select and join club	43
3.6.10 Club resignation	44
3.6.11 Feed recommendation	45
3.6.12 Machine learning training	46
3.7 Database Model	47

## สารบัญตาราง

ตารางที่	หน้า
2.1 HTTP Request Methods	10
3.1 Registration scenario	31
3.2 Login scenario	31
3.3 Logout scenario	31
3.4 Search event scenario	32
3.5 Select event scenario	32
3.6 Join event scenario	32
3.7 Evaluate event scenario	33
3.8 Search club scenario	33
3.9 Select club scenario	33
3.10 Join club scenario	34
3.11 Resignation scenario	34

## สารบัญ

หัวข้อ	หน้า
1.1 ภาคการศึกษาที่ 1	2
1.2 ภาคการศึกษาที่ 2	2
2.1 Mobile App Architecture	6
2.2 Presentation layer	7
2.3 Business layer	8
2.4 HTTP Request-Response Cycle	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 Graph Database vs Relational Database	15
2.10 หน้าแนะนำกิจกรรมของ ModLink และรายละเอียด	19
2.11 หน้าแนะนำกิจกรรมของ KMUTT Hatch	19
2.12 หน้า Mainpage ของ KMUTT Sinfo	20
2.13 หน้า Mainpage ของ Padlet	21
2.14 หน้าแนะนำกระทุ้ของ Pantip	21
2.15 หน้า Homepage ของเพจชุมชนใน Facebook	22
2.16 หน้า Notifications ของ Instargram	23
2.17 ตารางความแตกต่างของ Feature	24
2.18 ระบบการแนะนำกิจกรรม	25
3.1 Architecture diagram	28
3.2 Use case diagram and use case narratives	30
3.3 Sequence diagram ของ login	35
3.4 Registration	36
3.5 Logout	37
3.6 Search Event	38
3.7 Select and join event	39
3.8 Event evaluation	40
3.9 Notification about event from club	41
3.10 Seacrh club	42
3.11 Select and join club	43
3.12 Club resignation	44
3.13 Feed recommendation	45
3.14 Machine learning training	46
3.15 Database Model	47

## สารบัญสัญลักษณ์

SYMBOL		UNIT
$\alpha$	Test variable	$m^2$
$\lambda$	Interarrival rate	jobs/ second
$\mu$	Service rate	jobs/ second

## สารบัญคำศัพท์ทางเทคนิคและคำย่อ

Test	=	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam non condimentum purus. Pellentesque sed augue sapien. In volutpat quis diam laoreet suscipit. Curabitur fringilla sem nisi, at condimentum lectus consequat vitae.
MANET	=	Mobile Ad Hoc Network

# บทที่ 1 บทนำ

## 1.1 คำสำคัญ

Native Mobile Application, Machine Learning, Recommendation System

## 1.2 ที่มาและความสำคัญ

เนื่องจากความผู้ศึกษาตระหนักถึงปัญหาที่เกี่ยวข้องกับการกระจายข่าวสารเกี่ยวกับกิจกรรมต่าง 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

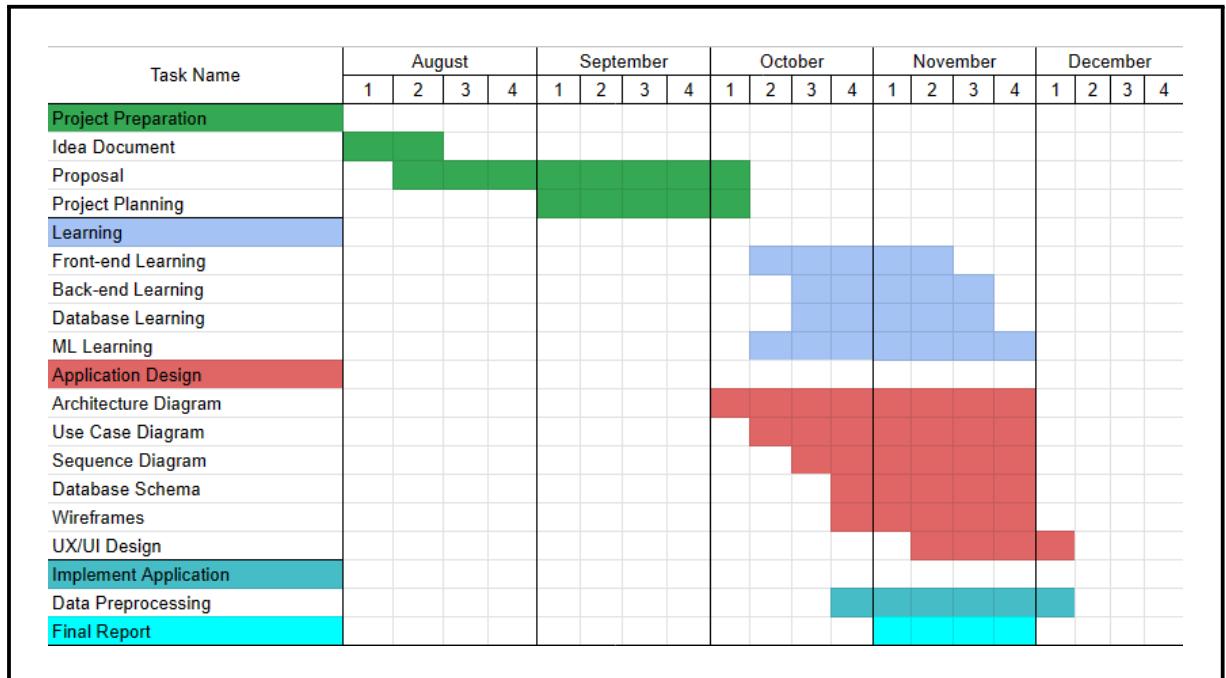
1.4.1 เพื่อพัฒนาแอปพลิเคชันสำหรับการกระจายข้อมูลข่าวสารและ กิจกรรมต่าง ๆ ภายในมหาวิทยาลัย เพื่อให้ง่ายต่อการติดตามข่าวไว้แก่นักศึกษาภายในมหาวิทยาลัย

1.4.2 เพื่อศึกษาพัฒนาระบบและความสามารถในการเข้าร่วมกิจกรรมของนักศึกษา เพื่อที่จะแนะนำกิจกรรมและชุมชนที่นักศึกษามีแนวโน้มให้ความสนใจ

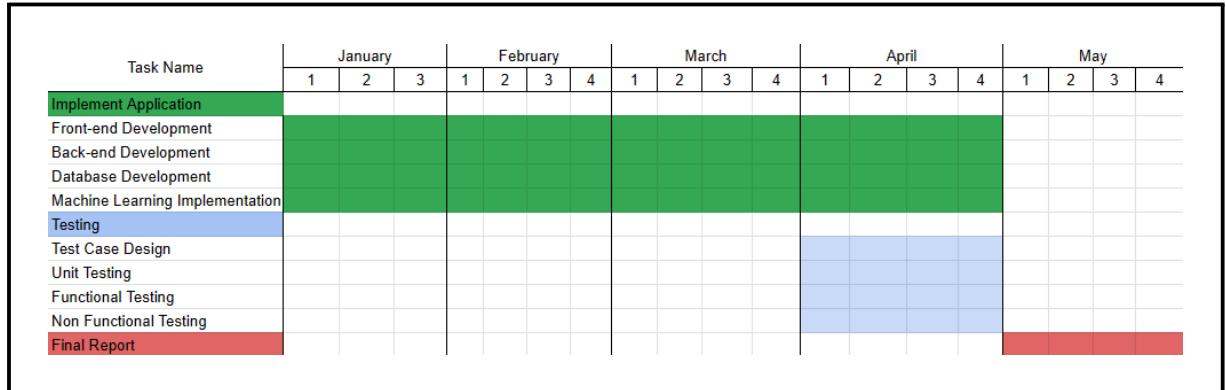
1.4.3 แอปพลิเคชันที่อำนวยความสะดวกในการเข้าร่วม กิจกรรมต่างๆ

1.4.4 สามารถแนะนำทางการจัดกิจกรรมที่มีนักศึกษาภายในมหาวิทยาลัยให้ความสนใจ ไปเสนอแก่ทางมหาวิทยาลัย เพื่อเพิ่มประสิทธิภาพในการจัดกิจกรรมต่อ ๆ ไป ที่จะเกิดขึ้นในอนาคต โดยวิเคราะห์จากเนื้อหากิจกรรมที่ทางนักศึกษาให้ความสนใจ

## 1.5 ตารางการดำเนินงาน



รูปที่ 1.1 ภาคการศึกษาที่ 1



รูปที่ 1.2 ภาคการศึกษาที่ 2

## 1.6 ขอบเขตของโครงการ

แอปพลิเคชันสำหรับการແນະນຳກิจกรรม ແລະ ຂໍມາຮັມໃຫ້ແກ່ນັກສຶກສາ ໂດຍມີເຊື້ອຄວາມສາມາດຕັ້ງຕ່ອໄປນີ້

1. ระบบ Log in ຜ່ານອື່ນເລຂອງມາວິທາລີ່ຍ
2. ຮະບບຮາບຮັມຂໍ້ອມລູກຂອງໝາຍຮັມຕ່າງ ๆ ເຄົາໄວ້ ໂດຍຜູ້ໃຊ້ສາມາດຄົນຫາແລະຕິດຕາມຂໍ້ອມລູກຂອງໝາຍຮັມທີ່ຕົນເອງສູນໄຈໄດ້
3. ຮະບບແຈ້ງເຕືອນກິຈกรรมທີ່ເກີ່ຽວຂ້ອງກັບໝາຍຮັມຫຼືຄວາມສູນໃຈຂອງນັກສຶກສາ
4. ຮະບບແຈ້ງເຕືອນກິຈกรรมປະເມີນພົກກິຈกรรม ເນື່ອຝ່ອມກິຈປະເມີນພົກກິຈການ
5. ຮະບບແນະນຳກິຈกรรมແລະ ຂໍມາຮັມ ຕາມຄວາມສູນໃຈຂອງຜູ້ໃຊ້ໂດຍອ້າງອີງຈາກ tag ຂອງກິຈกรรม
6. ຮະບບວິເຄຣະທີ່ຄວາມສູນໃຈຂອງຜູ້ໃຊ້ຜ່ານເນື້ອຫາຂອງກິຈกรรมທີ່ຜູ້ໃຊ້ເຄີຍເຂົ້າວ່ວມ ເຂົ້າໄປອ່ານຮາຍລະເອີຍດ ຫຼືເກີ່ຽວຂ້ອງກັບໝາຍຮັມທີ່ສູນໃຈ
7. ຮະບບແຍກປະເທດກິຈกรรมອັດໂນມັດໂດຍວິເຄຣະທີ່ຈາກເນື້ອຫາ ອອກມາເປັນ tag ຕ່າງ ๆ ໂດຍໃຊ້ Machine Learning

## 1.7 ປະໂຍື່ນທີ່ຄາດວ່າຈະໄດ້ຮັບ

1. ແອປັນທຶນທີ່ສັບສົນການເຂົ້າວ່ວມກິຈกรรมຂອງນັກສຶກສາ ແລະ ສາມາດແນະນຳ ກິຈกรรม ຂໍມາຮັມທີ່ນັກສຶກສານ່າຈະສູນໃຈໄດ້
2. ການຈຳແນກປະເທດຂອງກິຈกรรมຂອງນັກສຶກສາ

## บทที่ 2 ทฤษฎีความรู้และงานที่เกี่ยวข้อง

รายละเอียดของทฤษฎี ความรู้ และเทคโนโลยีที่นำมาใช้ในการสร้างและพัฒนาแอปพลิเคชัน

### 2.1 ทฤษฎีและความรู้ที่เกี่ยวข้อง

#### 2.1.1 Machine Learning

##### 2.1.1.1 Content-Based Filtering

Content-Based Filtering เป็นหนึ่งในแนวทางที่ได้รับความนิยมอย่างมากในระบบการแนะนำ เนื่องจากความสามารถในการให้คำแนะนำที่มีความเป็นบุคคลและเกี่ยวข้องกับความสนใจของผู้ใช้งาน กระบวนการนี้เน้นการวิเคราะห์เนื้อหาที่ผู้ใช้งานมีความสนใจและแนะนำสิ่งที่มีเนื้อหาที่คล้ายคลึงกันให้กับผู้ใช้งาน ตัวอย่างเช่น การแนะนำหนังสือที่มีเนื้อหาใกล้เคียงกับที่ผู้ใช้งานเคยซื้อมาก่อนหน้านี้ ดังนั้นเทคนิคนี้จึงเป็นการสร้างคำแนะนำที่เน้นความเป็นบุคคลและความสอดคล้องกับความต้องการของผู้ใช้งานอย่างมีความแม่นยำและเหมาะสม อย่างไรก็ตามการทำ Content-Based Filtering นั้น จะยึดตามความสนใจของผู้ใช้งานที่มีประวัติเก็บเอาไว้ทำให้สามารถแนะนำได้ในวงแคบ ๆ เท่านั้น

##### 2.1.1.2 Collaborative Filtering

Collaborative Filtering เป็นหนึ่งในแนวทางที่ได้รับความนิยมอย่างมากในการสร้างคำแนะนำสำหรับผู้ใช้งาน แนวทางนี้เสนอหลักการแนะนำที่มาจากการใช้งานที่เกิดขึ้นก่อนหน้านี้ของผู้ใช้งาน การ Collaborative Filtering มีวิธีการหลักที่สำคัญสองวิธีคือ User-Based Collaborative Filtering: แนวทางนี้จะทำการแนะนำสิ่งที่ผู้ใช้งานมีความสนใจอยู่ หากมีความคล้ายคลึงในพฤติกรรมการใช้งานระหว่างผู้ใช้งานสองคน เช่น ถ้าผู้ใช้งานสองคนมีรูปแบบพฤติกรรมที่คล้ายกัน ระบบจะแนะนำสิ่งที่ผู้ใช้อีกคนเคยใช้งานให้กับผู้ใช้งานคนอื่น ซึ่งเพิ่มโอกาสที่จะให้คำแนะนำที่หลากหลายขึ้นได้ อย่างไรก็ตามมีโอกาสที่จะแนะนำสิ่งที่ผู้ใช้งานไม่สนใจเลยได้ เช่นกัน และยังมีปัญหาความซับซ้อนเกิดขึ้นได้หากมีผู้ใช้งานหรือสิ่งที่จะแนะนำเป็นจำนวนมาก

Item-Based Collaborative Filtering: แนวทางนี้จะทำการแนะนำสิ่งที่ผู้ใช้งานมีความสนใจ และจะทำการแนะนำสิ่งที่มีเนื้อหาคล้ายกัน เมื่อพิจารณากระที่ทำให้เกิดขึ้นก่อนหน้านี้ ระบบจะแนะนำสิ่งที่มีความคล้ายคลึงให้กับผู้ใช้งาน อย่างไรก็ได้เนื่องจากเป็นการแนะนำจากสิ่งที่มีความใกล้เคียงกัน คำแนะนำที่ได้จึงอาจไม่ใช่สิ่งที่ผู้ใช้งานมองหา

##### 2.1.1.3 Natural Language Processing

Natural Language Processing หรือ NLP เป็นสาขานึงของปัญญาประดิษฐ์ ที่จะมุ่งเน้นการให้ความสามารถแก่คอมพิวเตอร์ที่เป็นการเข้าใจ ตีความ และปฏิสัมพันธ์กับภาษาของมนุษย์ แนวทางนี้นำเออกลไกอร์ทึมและเทคนิคในการประมวลผล วิเคราะห์ และสร้างข้อมูลข้อความและเสียง เพื่อให้เครื่องคอมพิวเตอร์สามารถสกัดความหมาย บริบท และความรู้จากภาษาที่มนุษย์สร้างขึ้นได้

##### 2.1.1.4 Word Embedding

Word Embedding ในการทำ Natural Language Processing คือการที่แปลงคำต่าง ๆ ออกมายเป็น vector เพื่อที่จะตรวจสอบความหมายของคำและความหมายในการสร้างประโยค ซึ่งเป็นพื้นฐานในการทำ Natural Language Processing Model ซึ่งสามารถนำมาประยุกต์ได้ทั้งการที่จะจัดประเภทกิจกรรมหรือการวิเคราะห์ความต้องการของผู้ใช้งานจากข้อความที่ถูกใช้

#### 2.1.1.5 Transformer Model

Transformer Model คือสถาปัตยกรรมการเรียนรู้ของเครื่อง สำหรับการประมวลผลและทำความเข้าใจภาษาของมนุษย์ ซึ่งทำงานได้ดีในการแปลภาษา การสรุป และการสร้างข้อความ โดยอาศัยหลักการของ self-attention ในการทำความเข้าความสัมพันธ์ระหว่างคำในประโยคพร้อมกัน ทำให้สามารถเข้าใจบริบทและความหมายได้มากขึ้นอย่างมีประสิทธิภาพ โดย Transformer-base model ที่ถูกใช้งานในโครงงานนี้ มี 2 รูปแบบ คือ

1. T5 (Text-to-Text Transfer Transformer) เป็นสถาปัตยกรรมที่ประยุกต์มาจาก unified text-to-text framework ซึ่งพัฒนาจาก NLP model หลากหลายงานให้ใช้งานง่ายขึ้น จึงสามารถนำมารันใน train ต่อเพื่อทำงานเกี่ยวกับ Natural language processing ได้อย่างเน้นประสิทธิภาพ โดยที่มีผลกระทบต่อประสิทธิภาพในการทำงานเพียงเล็กน้อย อย่างไรก็ตาม T5 มีข้อจำกัดในการเข้าใจบริบทของข้อความและต้องอาศัยการปรับแต่งในการทำงานที่เฉพาะด้าน
2. BERT (Bidirectional Encoder Representations from Transformers) สถาปัตยกรรมที่ถูก train โดยคลังข้อความขนาดใหญ่ จึงสามารถทำงานในการจับบริบทที่อยู่เบื้องหลังข้อความได้ 亥มากกับการทำงาน Natural language processing ที่หลากหลายอย่างไรก็ตาม BERT ต้องการหน่วยประมวลผลในการ train และใช้งาน รวมไปถึงไม่สามารถปรับแต่งได้เน้นประสิทธิภาพเท่า T5

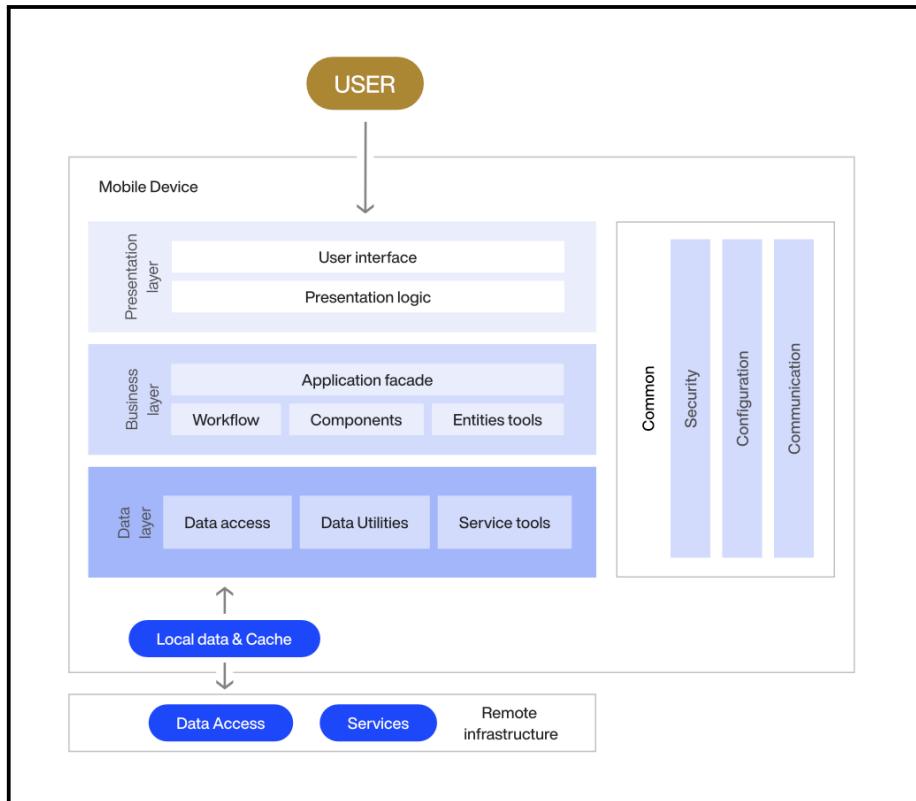
#### 2.1.1.6 Artificial Neural Network

Neural Network เป็นสถาปัตยกรรมการเรียนรู้ของเครื่องที่ได้รับแรงบันดาลใจมาจากการประมวลผลภาษาของมนุษย์ โดยมีหลักการคือการแบ่งการทำงานเป็น node ต่าง ๆ ที่รับค่าที่ประมวลผลมาจาก node ที่อยู่ใน input layer และส่งต่อให้ node ประมวลผลที่อยู่ใน hidden layer ไปเรื่อย ๆ จนถึง node ประมวลผลสุดท้ายใน output layer เพื่อตีความผลลัพธ์ ซึ่งสามารถทำงานได้หลากหลายรูปแบบ รวมไปถึงงาน Natural Language Processing ซึ่งในโครงงานนี้มี Artificial Neural Network 2 รูปแบบ คือ

1. Shallow Neural Network ซึ่งเป็น Neural Network ที่มีจำนวน Layer ใน การประมวลผลอยู่น้อย มีข้อดีคือการที่จะใช้หน่วยประมวลผลน้อยและสามารถ train ได้ไว แต่ในทางกลับกันมีข้อจำกัดในการตีความที่ซับซ้อนรวมถึงบริบทที่อยู่ในข้อความของงาน Natural Language Processing
2. Regularizing and Optimizing LSTM Language Model คือ Neural Network ที่ถูกพัฒนามาสำหรับการทำ Natural Language Processing ทำให้สามารถประมวลผลข้อมูลที่มีความต่อเนื่องอย่างเช่นประโยคได้ ตัวโมเดลมีความสามารถในการทำความเข้าใจบริบทในข้อความได้ โดยพิจารณาข้อมูลในหน่วยความจำของโมเดล ซึ่งสามารถจำหรือลบข้อมูลได้ตามความเหมาะสม และสามารถรับมือกับข้อความที่เมรู้จักได้ อย่างไรก็ตามคุณภาพของ model ขึ้นอยู่กับคุณภาพของข้อมูลที่ใช้ train และการปรับแต่งค่อนข้างส่งผลกระทบตัว model

### 2.1.2 Mobile App Architecture

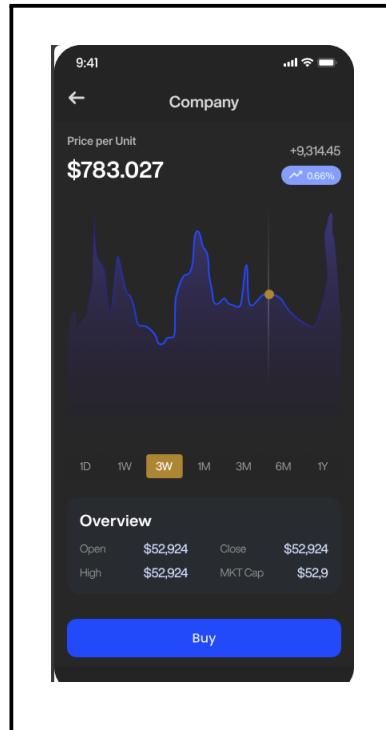
ในการพัฒนาแอปพลิเคชัน สถาปัตยกรรมหมายถึงกฎ กระบวนการ และโครงสร้างภายในของแอปพลิเคชัน หรือก็คือวิธีการสร้างแอปพลิเคชัน โดยจะเป็นเรื่องของการกำหนดรูปแบบที่ส่วนประกอบต่าง ๆ สื่อสารกันเพื่อประมวลผลข้อมูล input จากผู้ใช้และประมวลผลข้อมูล output ให้กับผู้ใช้



รูปที่ 2.1 Mobile App Architecture

สถาปัตยกรรมของแอปพลิเคชันส่วนใหญ่จะประกอบด้วยสามเลเยอร์หลัก ๆ ได้แก่ Presentation layer, Business layer และ Data layer

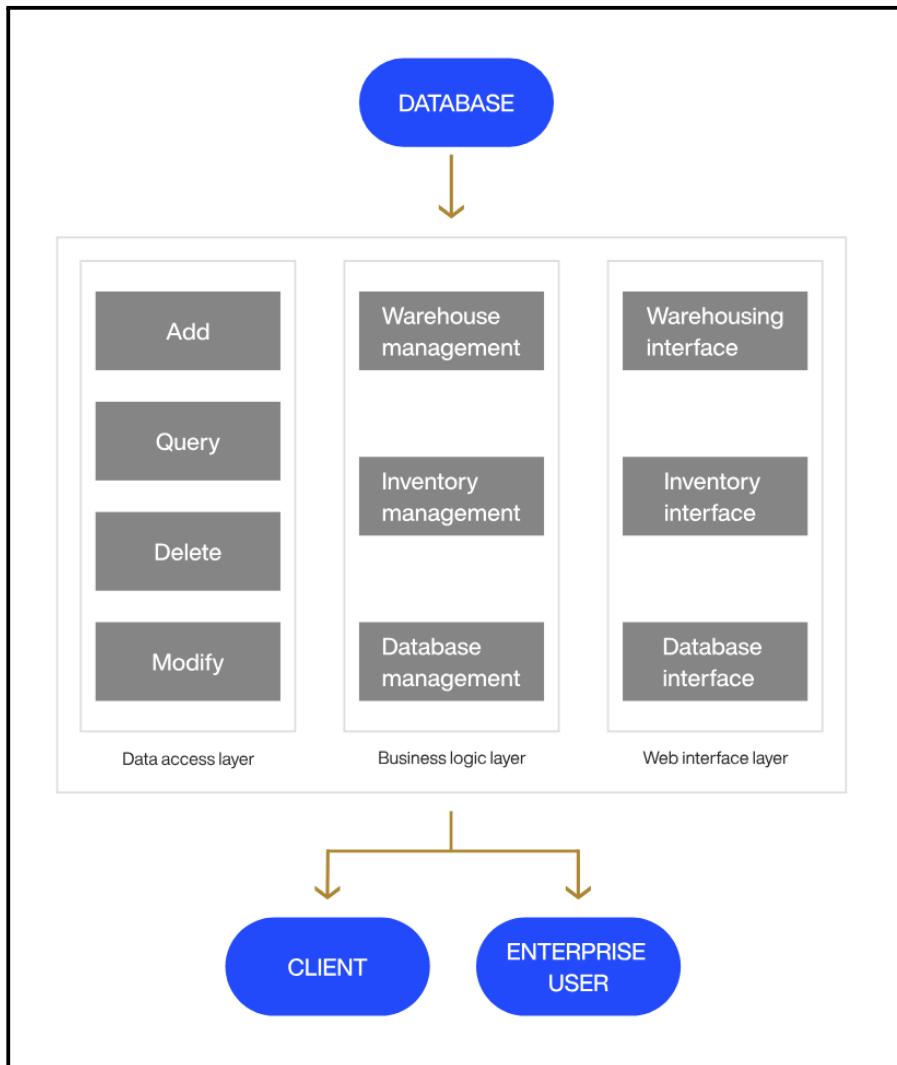
1. Presentation layer หรือ front end คือส่วนของแอปพลิเคชันที่ผู้ใช้มองเห็นและมีปฏิสัมพันธ์ได้ โดยมีส่วนติดต่อผู้ใช้ (user interface หรือ UI) ของแอปพลิเคชันเป็นส่วนสำคัญของเลเยอร์นี้ วัตถุประสงค์หลักของเลเยอร์นี้คือการนำข้อมูลที่ส่งมาจาก business layer มาแสดงผลในลักษณะที่ผู้ใช้สามารถเข้าใจได้ ไม่ว่าจะเป็น UI แบบพื้นฐาน เช่น UI แสดงที่อยู่อีเมลของผู้ใช้หรือ UI ที่ซับซ้อน เช่น แอปซื้อขายหุ้น ที่ใช้ข้อมูลแสดงเกี่ยวกับตลาดหลักทรัพย์และแสดงเป็นกราฟหรือแผนภูมิ แม้ว่าจะพัฒนาส่วนใหญ่จะรับผิดชอบด้านธุรกิจและข้อมูล



รูปที่ 2.2 Presentation layer

## 2. Business layer

เลเยอร์นี้จะประกอบด้วยตระกากของแอปพลิเคชัน หรือเก็อวิธีการทำงานของแอปพลิเคชัน โดยมักจะเป็นการนำข้อมูลที่ผู้ใช้ป้อนหรือข้อมูลจากขั้นข้อมูลมาประมวลผล จากนั้นจึงส่งไปยัง presentation layer ส่วนใหญ่ business layer จะเป็นส่วนที่ซับซ้อนที่สุดในแอปพลิเคชัน โดยปกติแล้วจะแบ่งออกเป็นเลเยอร์อยู่ห้องหรือส่วนประกอบหลายส่วน โดยแต่ละส่วนมีหน้าที่รับผิดชอบในการทำงานเฉพาะ ตัวอย่างเช่น หากคุณมีแอปการจัดการทรัพยากรองค์กร (ERP) business layer อาจมีส่วนประกอบสำหรับการจัดการคลังสินค้าและระบบจัดการสินค้าคงคลัง



รูปที่ 2.3 Business layer

### 3. Data layer

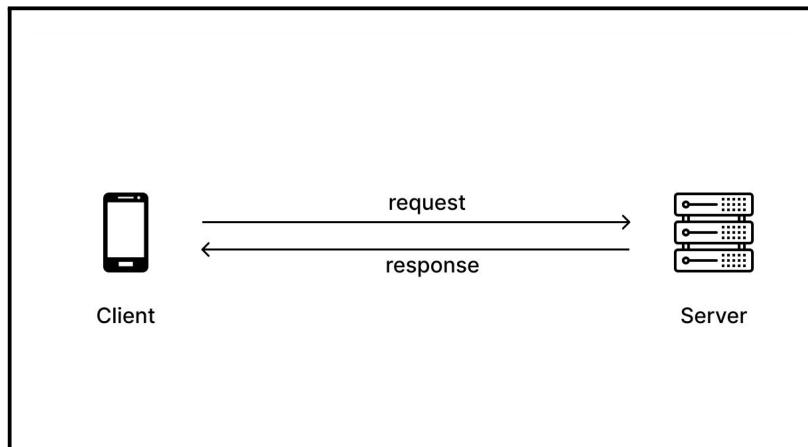
เลเยอร์นี้เป็นตัวกลางระหว่างเลเยอร์อื่นๆ กับทรัพยากรภายนอก วัตถุประสงค์หลักของเลเยอร์นี้คือการรวบรวมข้อมูลจากแหล่งต่างๆ (เช่นฐานข้อมูล เซิร์ฟเวอร์คลาวด์ หรือ API) และส่งไปยังเลเยอร์ด้านบน ตัวอย่างเช่น เมื่อผู้ใช้ขอให้แอปพลิเคชันแสดงໂປຣີຟ້າ ของตน data layer จะเชื่อมต่อกับฐานข้อมูลและขอข้อมูลที่เกี่ยวข้องทั้งหมด เช่น ชื่อ วันเกิด ໄຟຣີຟ້າປາກ ແລະ อื่นๆ อย่างไรก็ตาม ในเลเยอร์นี้ข้อมูลส่วนใหญ่ยังผ่านการประมวลผล จึงอาจจะมีข้อมูลบางอย่าง เช่น ແທັກຫຼືອໄວດີທີ່ຜູ້ໃໝ່ມ່ວຽນເຫັນ ໃນສ່ວນນີ້ຈິງຕ້ອງทำการส่งข้อมูลให้ business layer ประมวลผลเพิ่มเติมเป็นลำดับต่อไป

### 2.1.3 HTTP Protocol

#### 2.1.3.1 HTTP Protocol คืออะไร

HTTP (Hypertext Transfer Protocol) เป็นโปรโตคอลที่ใช้ในการแลกเปลี่ยนข้อมูลผ่านทางอินเทอร์เน็ต เปรียบเสมือนระบบส่งข้อมูลบนอินเทอร์เน็ตที่ช่วยให้มันใจได้ว่าข้อมูลจะถูกส่งจากที่หนึ่งไปยังอีกที่หนึ่งได้

#### 2.1.3.2 HTTP Request-Response Cycle



รูปที่ 2.4 HTTP Request-Response Cycle

การสื่อสารในโปรโตคอล HTTP มีศูนย์กลางอยู่ที่แนวคิดที่เรียกว่า Request-Response Cycle เป็นกระบวนการที่คลื่อนตัว (Client) เช่น เว็บเบราว์เซอร์หรือแอปพลิเคชันมือถือ สื่อสารกับเซิร์ฟเวอร์ (Server) เพื่อขอทรัพยากรที่ต้องการหรือเพื่อดำเนินการบางอย่าง โดยวงจรประกอบด้วยหลายขั้นตอนได้แก่

1. คลื่อนตัวเริ่มต้นการส่งคำขอไปยังเซิร์ฟเวอร์โดยการส่งข้อความร้องขอ (HTTP request message) ซึ่งประกอบด้วยข้อมูลอาทิเช่น ทรัพยากรที่ต้องการและพารามิเตอร์เพิ่มเติมอื่น ๆ
2. เซิร์ฟเวอร์ได้รับข้อความร้องขอและประมวลผลโดยใช้ทรัพยากรที่มีอยู่เพื่อสร้างข้อความตอบกลับ (HTTP response message)
3. เซิร์ฟเวอร์ส่งข้อความตอบกลับไปยังคลื่อนตัว ซึ่งโดยทั่วไปจะมีทรัพยากรที่ร้องขอ (เช่น หน้าเว็บ) และข้อมูลเพิ่มเติมหรือเมแทดา (ข้อมูลที่ใช้อธิบายชุดข้อมูลอื่นอีกที)
4. คลื่อนตัวได้รับข้อความตอบกลับและประมวลผล โดยปกติจะเป็นการแสดงเนื้อหาในเว็บเบราว์เซอร์หรือในแอปพลิเคชัน
5. คลื่อนตัวอาจเริ่มการร้องขอเพิ่มเติมไปยังเซิร์ฟเวอร์ โดยทำขั้นตอนเดิมแล้วแต่ความจำเป็น

#### 2.1.3.3 HTTP Request Methods

Request method จะเป็นการบอกเซิร์ฟเวอร์ว่าลูกค้าต้องการให้เซิร์ฟเวอร์ดำเนินการอะไร Request method ที่พบเจอบ่อยจะมีดังนี้

### 2.1.4 REST API

#### 2.1.4.1 REST คืออะไร

REST ย่อมาจาก Representational State Transfer เป็นรูปแบบการส่งข้อมูลระหว่าง Server-Client รูปแบบหนึ่งซึ่งอยู่บนพื้นฐานของ HTTP Protocol เป็นการสร้าง Web Service เพื่อแลกเปลี่ยนข้อมูลกันผ่าน Application วิธีหนึ่ง ซึ่งส่งข้อมูลได้หลายชนิด ไม่ว่าจะเป็น Text, XML, JSON หรือส่วนมาเป็นหน้า HTML เลย REST ทำงานอยู่บน HTTP Protocol ทำให้เวลาใช้งานจะต้องอยู่บนพื้นฐาน HTTP Method เช่น GET, POST, PUT, DELETE จะใช้ Method ไหนก็ขึ้นอยู่กับว่าจะทำอะไรกับข้อมูล แตกต่างกันไปตาม Operation CURD

HTTP METHODS	DEFINITION
HEAD	ถามเชิร์ฟเวอร์เกี่ยวกับสถานะ (ขนาด ความพร้อมใช้งาน) ของทรัพยากร
GET	ขอทรัพยากรจากเชิร์ฟเวอร์
POST	ขอให้เชิร์ฟเวอร์สร้างทรัพยากรใหม่
PUT	ขอให้เชิร์ฟเวอร์แก้ไข/อัปเดตทรัพยากรที่มีอยู่แล้ว
DELETE	ขอให้เชิร์ฟเวอร์ลบทรัพยากร

ตารางที่ 2.1 HTTP Request Methods

เช่น เมื่อต้องการจะเรียกดูข้อมูลทั้งหมด ก็ใช้ GET เมื่อต้องการเพิ่มข้อมูล ก็ใช้ POST เป็นต้น

#### 2.1.4.2 การออกแบบ REST API

- เลือกใช้ HTTP Method ให้เหมาะสมกับการใช้งาน ปกติในการสร้าง URL จะไม่ใส่ชื่อกิจกรรมของ API มาอยู่ใน path เช่น /createUsers, /getUserDetail แบบนี้คือผิดหลัก เพราะในการที่จะระบุว่าแต่ละ API จะใช้ในการทำอะไร จะมี HTTP Method ในการระบุอยู่แล้ว
- การสร้าง URL ของ API endpoint ให้ตรงตามมาตรฐาน ในการสร้าง URL ของ API นั้นมีทั้งหมดสามกฎที่สำคัญ ก็คือ
  - ควรจะมีแค่ชื่อ resource เท่านั้น (resource เป็นตัวแทนของ สิ่งของบางอย่าง ที่เข้ามายังกับข้อมูล เช่น Users, Customers, Orders)
  - ชื่อ path ควรจะเป็นรูปพหุพจน์ของ resource
  - ไม่ควรจะมีชื่อกิจกรรมที่บอกถึงวัตถุประสงค์ของ API (เช่น add, update, delete) ตามที่กล่าวไว้ในข้อแรก

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

- 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 ทั้งหมด ที่อยู่ใน customer id เป็น 2 สามารถเขียนเป็น GET /customers/2/users แต่มีบางกรณีที่ ข้อมูล Nested ที่เยอะมากๆ ถ้าเรา มันเป็น URL path ก็จะยาวมาก อาจจะใส่เป็น query string หรือ จับใส่ใน body แทนก็จะอ่านลำบากขึ้น ต้องพิจารณาถึง use case กันอีกที

- ควรมี API Versioning หาก API มีผู้ใช้งานมาใช้งานแล้ว ในการแก้ไขอย่างไรให้ถูกต้อง ก็จะทำได้ลำบากขึ้น เพราะถ้าแก้ไปแล้ว ทำให้ service ที่ใช้ API อยู่ใช้งานไม่ได้อาจจะทำให้เกิดปัญหาขึ้น เพราะอย่างนั้น ทุก ๆ API ควรทำ version ไว้ หากมีการเปลี่ยนแปลง ก็สามารถแยกออกมาระบุเป็นอีก version ได้เลย โดยทั้ง version เก่าและใหม่ต้องทำงานได้ทั้งคู่ ตัวอย่างเช่น POST v1/users และ POST v2/users สามารถนำเลข version มาต่อหน้า API ได้เลย
- 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 ผลลัพธ์หลัง filter คือข้อมูลของ order ที่มีชื่อว่า “MyOrders” และอยู่ใน customer id ที่ 2 จากหลักการข้อ 2. ที่กล่าวไปว่า ข้อมูล Nested ที่เยอะมาก ๆ หรือต้องการกรองข้อมูลเยอะมาก ถ้านำมาเป็น URL path ก็จะยาวเกิน ซึ่งสามารถนำมาใส่ใน filter ต่อท้าย URL path แทนได้
- Sorting (การจัดเรียงข้อมูล) สามารถเรียงลำดับข้อมูลที่เรียกมาแสดงผลได้ ซึ่งการออกแบบ sort ที่ดีจะต้องออกแบบให้ยืดหยุ่น สามารถเรียงจากน้อยไปมาก หรือมากไปน้อยก็ได้ โดยใส่ query เข้าไปต่อท้าย path (คล้ายกับ filter) ซึ่งสามารถนำ sort by ไปต่อท้ายได้ เช่น GET /users?sort by=+email หรือ GET /users?sort by=-email จากตัวอย่างด้านบน +email คือเรียงเป็นน้อยไปมาก และ -email เรียงเป็นมากไปน้อย หรือสามารถเขียนในรูปแบบอื่นๆ ได้ เช่น
- 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

โดยหลักสำคัญจะอยู่ที่ทุกรูปแบบการเขียนจะต้องหยิบหย่อน สามารถเปลี่ยนลำดับการเรียงข้อมูลได้ และผู้ใช้งานสามารถอ่านได้ อย่างเข้าใจว่าเป็นการเรียงลำดับแบบไหน

- 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) 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 แล้วฝั่ง Client จะรู้ด้วยว่า API ใช้งานได้จริงหรือเปล่า หรือส่งไปสำเร็จไหม จึงต้องส่ง response ที่มี HTTP Status code ระบุ กลับไปยัง client เพื่อบอกว่า request นั้น ๆ Pass, Fail หรือ request นั้นผิด กรณี Success จะมี HTTP status code ที่ใช้งานกันทั่วไปได้แก่

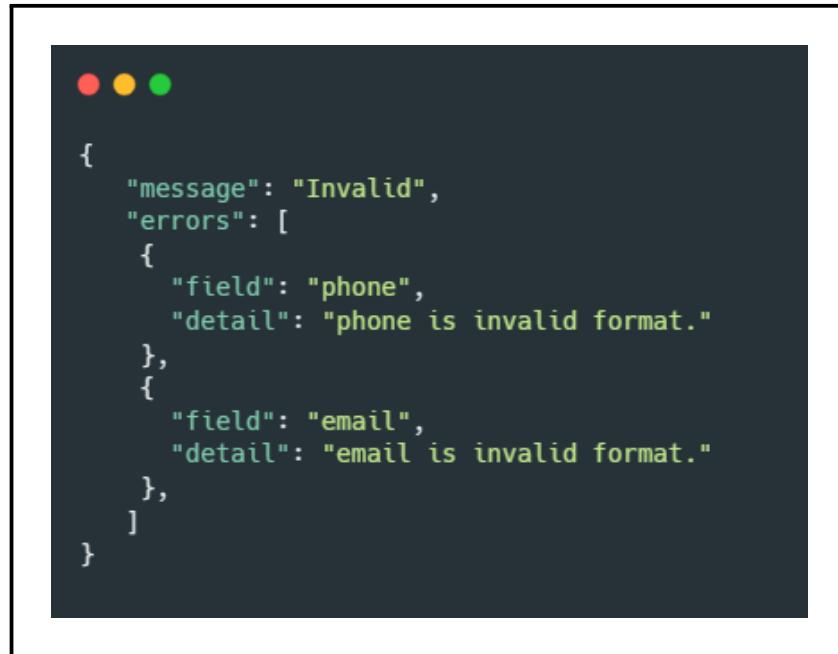
- 200 Ok: เป็นมาตรฐานของ HTTP response เพื่อบ่งบอกว่า request นั้นสำเร็จ ให้สำหรับ GET, PUT หรือ POST ก็ได้
- 201 Created: เป็น response เพื่อบ่งบอกว่าข้อมูลใหม่ได้ถูกสร้างขึ้นสำเร็จ ให้สำหรับ POST
- 204 No Content: เป็น response สำหรับบ่งบอกดำเนินการ Success แต่ไม่ได้ return ข้อมูลลับ ส่วนใหญ่จะใช้กรณีลบข้อมูล DELETE ที่ไม่ได้ส่ง response ที่เป็นข้อมูลลับไป 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 ไว้ไว้ API นั้น ๆ จะสามารถเรียกได้ครั้งเดียว
- 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 จะเห็นว่า HTTP Status code แต่ละตัวจะมีความหมายของมันชัดเจน เพราะฉะนั้นการออกแบบที่ดีจะต้องเลือกให้ HTTP Status code ให้ตรงตามวัตถุประสงค์เพื่อให้ผู้ใช้งานที่ได้รับ response กลับไปเข้าใจ response นั้นได้มากขึ้น

7. การ Handle Error ให้ user เข้าใจ นอกเหนือจาก HTTP Status code แล้ว ต้องออกแบบ response สำหรับ error กรณีต่าง ๆ ไว้ด้วย เพื่อให้ user เข้าใจ error ของ API หากขึ้น ยกตัวอย่างกรณีที่ request ส่งมา parameter ไม่ถูกต้อง แทนที่จะ response กลับไปว่า



รูปที่ 2.6 Handle Error1

สามารถระบุได้เลยว่า field ไหนที่ผิด format บ้าง



```

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

```

รูปที่ 2.7 Handle Error2

เท่านี้ user ก็เข้าใจแล้วว่ามี 2 field นี้ที่ผิด format

### 2.1.5 Graph Database

#### 2.1.5.1 Graph Database คืออะไร

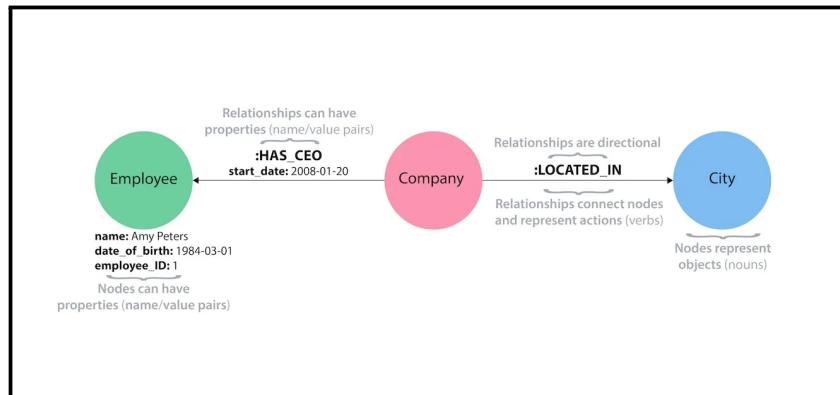
Graph database หรือฐานข้อมูลแบบกราฟ จัดเป็น NoSQL Database รูปแบบหนึ่ง ซึ่งนำมาใช้แก้ปัญหา database ที่มีข้อมูลขนาดใหญ่และไม่มีรูปแบบชัดเจน ฐานข้อมูลแบบ Graph ออกแบบมาเพื่อแสดงความสัมพันธ์ (Relationship) ระหว่างข้อมูลที่มีความเชื่อมโยงกับข้อมูลที่เราสนใจได้อย่างชัดเจน รวมถึงมีความสามารถในการเก็บข้อมูลที่ไม่ต้องกำหนดรูปแบบล่วงหน้า

#### 2.1.5.2 ทำไมต้องเป็น Graph Database

การเข้าถึง node และ relationships ในฐานข้อมูล Graph เป็นอิสระ มีประสิทธิภาพและใช้เวลาในการทำงานคงที่ และช่วยให้เราสำรวจ การเชื่อมต่อหลายล้านต่อวินาทีต่อเรคคอร์ดได้อย่างรวดเร็ว มีความเป็นอิสระจากขนาดรวมของชุดข้อมูลทั้งหมดของเรา ทำให้ฐานข้อมูลแบบ Graph มีความสามารถในการจัดการข้อมูลที่มีรูปแบบซับซ้อนและมีความเชื่อมต่อกันสูงได้มีประสิทธิภาพ

### 2.1.5.3 Property ของ Graph Model

เทคโนโลยีส่วนใหญ่มีวิธีการที่แตกต่างกันเล็กน้อยในการสร้าง องค์ประกอบที่สำคัญของฐานข้อมูล Graph วิธีหนึ่ง คือ Graph Model ซึ่งมุ่งจะถูกจัดระเบียบเป็น 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

	Relational Database	Graph Database
รูปแบบการเก็บข้อมูล	ตารางที่มีแคลและคอลัมน์	โนนดที่เชื่อมต่อถึงกันพร้อมข้อมูลที่แสดงเป็นเอกสาร JSON
การทำงาน	การทำงานของ SQL เช่น สร้าง อ่าน อัปเดต และลบ (CRUD)	การดำเนินการรวมถึง CRUD และการดำเนินการผ่านกราฟตามทฤษฎีกราฟทางคณิตศาสตร์
ความสามารถในการปรับขนาด	ฐานข้อมูลแบบเชิงสัมพันธ์แบบดั้งเดิมสามารถปรับขนาดในแนวตั้งได้แต่ไม่ค่อยเชี่ยวชาญกับการปรับขนาดในแนวนอน	ฐานข้อมูลแบบกราฟเชี่ยวชาญในการปรับขนาดตามแนวโน้ม สามารถใช้การแบ่งพาร์ติชันเพื่อกระจายข้อมูลไปยังโนนจำนวนมาก
ประสิทธิภาพ	ฐานข้อมูลแบบเชิงสัมพันธ์เพื่อซ้อนกับการสืบค้นที่ซับซ้อนเมื่อสำรวจความสัมพันธ์ที่อาจทำให้ประสิทธิภาพการทำงานช้าลง	ฐานข้อมูลแบบกราฟเชี่ยวชาญในการแสดงและสืบค้นความสัมพันธ์ระหว่างข้อมูล
ความสามารถในการใช้งาน	ฐานข้อมูลแบบเชิงสัมพันธ์ทำงานได้ดีกับข้อมูลขนาดใหญ่และข้อมูลที่มีโครงสร้างพวณนั้นไม่ค่อยเชี่ยวชาญเมื่อเป็นเรื่องการสืบค้นแบบหลายช่วง	ฐานข้อมูลแบบกราฟใช้งานง่ายเมื่อต้องจัดการกับข้อมูลที่เน้นความสัมพันธ์เป็นหลัก เมื่อใช้ภาษาสืบค้นแบบกราฟ คุณสามารถสืบค้นข้อมูลโดย平行รายการได้อย่างรวดเร็ว

รูปที่ 2.9 Graph Database vs Relational Database

### 2.1.6 UML Diagram

#### 2.1.6.1 UML Diagram คืออะไร

UML Diagram คือ แผนภาพที่ใช้ในการแสดงและอธิบายโครงสร้างและพฤติกรรมของ code เพื่อสื่อสารให้ Developer และผู้ที่เกี่ยวข้องทุกคนเข้าใจตรงกัน ซึ่งสามารถเอามาใช้อธิบายความสัมพันธ์ของสิ่งต่างๆที่อยู่ในชิ้นงานได้ โดย UML Diagram มีแผนภาพหลายรูปแบบ สำหรับใช้อธิบายโครงสร้างและความสัมพันธ์ในรูปแบบต่างๆที่ Developer ต้องทำความเข้าใจเพื่อใช้ในการปฏิบัติงาน

#### 2.1.6.2 Use case Diagram

Use Case Diagram เป็นหนึ่งใน UML Diagram ที่ใช้ในการแสดงภาพรวมของวิธีการใช้ระบบหรือซอฟต์แวร์ จากมุมมองของผู้ใช้หรือแต่ละกลุ่มผู้ใช้ โดยทำให้เจ้าต่อการเข้าใจและสื่อสารความต้องการของระบบกับผู้ใช้และทีมพัฒนา ลักษณะหลักของ Use Case Diagram ประกอบด้วย

1. Actor : แสดงตัวบุคคลหรือระบบที่มีส่วนร่วมในการใช้งานระบบ สามารถเป็นบุคคล, ระบบภายนอก, หรือภายในระบบได้
2. Use Case : แสดงกิจกรรมหรือฟังก์ชันที่ระบบหรือซอฟต์แวร์ให้บริการในแต่ละคำสั่งหรือเหตุการณ์ที่มีผู้ใช้ร้องขอ
3. Association : แสดงความสัมพันธ์ระหว่างผู้ที่เกี่ยวข้องกับ 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 เป็นหนึ่งใน UML Diagram ที่ใช้งานเพื่อแสดงลำดับขั้นตอนหรือการทำงานของวัตถุต่าง ๆ ภายในระบบหรือโปรแกรม 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, ควบคุมอยู่ที่วัตถุหรือส่วนชิวิตใด

## 2.2 เทคโนโลยี

### 2.2.1 Integrated Development Environment (IDE)

- Visual Studio Code

โปรแกรมสำหรับเขียนโค้ดที่ใช้ในการแก้ไขและปรับแต่งโค้ด จากค่ายไมโครซอฟท์ มีการพัฒนาออกแบบในรูปแบบของ OpenSource จึงสามารถนำไปใช้งานได้แบบฟรี ๆ สนับสนุนภาษาโปรแกรมมีจำนวนมากทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้เจ้ายิ่งขึ้น มีเครื่องมือส่วนขยายต่าง ๆ ให้เลือกใช้อย่างมากมาย

- Pycharm

โปรแกรมสำหรับเขียนโค้ดสำหรับภาษาไพทอน ติดตั้งบนเครื่องคอมพิวเตอร์ ได้ทั้งบนระบบปฏิบัติการ Windows, MacOS และ Linux

- Google Colab

โปรแกรมสำหรับเขียนโค้ดสำหรับภาษาไพทอนในเบราว์เซอร์ โดยไม่ต้องกำหนดค่าใดและสามารถเข้าถึง GPU โดยไม่มีค่าใช้จ่าย

### 2.2.2 Design

- Figma

เครื่องมือออกแบบ User interface โดยสามารถใช้ออกแบบได้ตั้งแต่เว็บไซต์ แอปพลิเคชัน หรือโลโก้ ในรูปแบบที่มีลูกเล่นมากกว่า ที่เคยเห็นในอดีต เช่น การออกแบบ Interactive component เป็นต้น

- LucidChart

เว็บแอปพลิเคชันสำหรับสร้างโดยรวม ผังงาน แผนภาพแบบจำลอง หรือแผนที่ความคิด สามารถแชร์แผนภาพให้ผู้อื่นเพื่อทำงาน ร่วมกันแบบเรียลไทม์ได้ มีเทมเพลตสำหรับรูปแบบ เช่น ผังงาน แบบโครงร่าง แผนภาพเครือข่าย และแผนผังเว็บไซต์ เป็นต้น นอกจากนี้ยังสามารถแสดงความคิดเห็น หรือสนทนารูปแบบกลุ่มได้ และยังดาวน์โหลดเป็นไฟล์รูปแบบต่าง ๆ ได้

### 2.2.3 Frontend

- React Native

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

FastAPI คือเว็บเฟรมเวิร์กที่มีความรวดเร็วและประสิทธิภาพสูง สำหรับการสร้าง API ด้วยภาษา Python ที่มีจุดเด่นได้แก่ 1. มีความรวดเร็ว ประสิทธิภาพเทียบเท่ากับ NodeJS และ Go 2. สร้างง่าย เพิ่มความเร็วในการพัฒนา 3. ลดข้อผิดพลาดที่เกิดจากมนุษย์ (นักพัฒนา)

### 2.2.5 Database

- Neo4j

เป็นระบบฐานข้อมูลที่ถูกออกแบบมาเพื่อจัดเก็บข้อมูลแบบกราฟ (Graph Database) ที่สามารถจัดเก็บแม่ปะและสืบทอดความสัมพันธ์ระหว่างข้อมูลได้อย่างมีประสิทธิภาพและมีความยืดหยุ่น ระบบฐานข้อมูลกราฟนี้ถูกออกแบบมาเพื่อเก็บข้อมูลในรูปแบบของ Node และ Edge ต่อระหว่างโนนенд้วยเส้นเชื่อมที่เรียกว่า Relationships ทำให้สามารถแสดงความสัมพันธ์และการเชื่อมโยงของข้อมูลได้อย่างชัดเจน

### 2.2.6 Machine Learning

- mT5

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

- BERT-th

เป็นโมเดลการเรียนรู้ของเครื่องสำหรับการประมวลภาษาธรรมชาติที่ถูกพัฒนาขึ้นเพื่อใช้ในภาษาไทย โดยเด่นที่มีความสามารถในการเข้าใจและประมวลผลข้อมูลที่เป็นภาษาไทยอย่างมีประสิทธิภาพ โดยใช้หลักการของ BERT (Bidirectional Encoder Representations

from Transformers) ซึ่งเป็นโมเดลการเรียนรู้ของเครื่องที่สามารถทำนายคำถัดไปในประโยคจากข้อมูลทั้งด้านซ้ายและด้านขวาของคำนั้น ๆ

- fastText

เป็น library สำหรับการทำโมเดลการเรียนรู้ของเครื่องสำหรับการประมวลภาษาธรรมชาติ พัฒนาโดย Facebook AI Research และเน้นการสร้างเวกเตอร์คำและการจัดกลุ่มคำศัพท์ โดยใช้หลักการของการแปลงคำเป็นเวกเตอร์ที่สามารถใช้ในการค้นหาความคล้ายคลึงระหว่างคำ

- thai2fit

เป็นโมเดลการเรียนรู้ของเครื่องสำหรับการประมวลภาษาธรรมชาติที่ได้รับการปรับปรุงและเพิ่มประสิทธิภาพสำหรับการทำงานกับข้อมูลภาษาไทย โดยใช้หลักการของ fastText และ Word Embedding ที่ถูกพัฒนามาเพื่อภาษาไทย

- BERT-Base-Multilingual-Case

เป็นโมเดลการเรียนรู้ของเครื่องสำหรับการประมวลภาษาธรรมชาติที่รองรับหลายภาษาและมีความสามารถในการประมวลผลข้อความในหลายภาษาที่มีตัวอักษรต่างกัน โดยเด่นอยู่ที่ Google และเป็นการปรับปรุงจาก BERT โดยรองรับการแปลงตัวอักษร เป็นตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก

- Thai nlp

เป็นแหล่งข้อมูลหลักสำหรับนักวิจัยและผู้พัฒนาที่กำลังทำงานด้าน Natural language processing ในภาษาไทย ซึ่งจะรวบรวมเครื่องมือการทำ Natural language processing โมเดลที่พร้อมใช้งานและ ข้อมูลที่เป็นประโยชน์สำหรับงานวิจัยและการพัฒนา ในด้าน NLP ในภาษาไทยเอาไว้ โดยมีความหลากหลายของเครื่องมือและข้อมูลที่มีคุณภาพสูง เช่น โมเดลการแปลงภาษา การจัดกลุ่มคำศัพท์ และวิธีการประมวลผลข้อมูลที่ถูกพัฒนาให้ใช้งานได้อย่างมีประสิทธิภาพ

### 2.2.7 Version Control

- Git

Version Control ตัวหนึ่งซึ่งจะเป็นระบบที่มีหน้าที่ทำการจัดเก็บการเปลี่ยนแปลงของไฟล์ใน Project และมีการ backup ให้สามารถที่จะเรียกดูหรือทำการย้อนกลับไปดูเวอร์ชันต่างๆของ Project ที่ได้เวลาใดก็ได้ ดังนั้น Version Control ก็หมายความว่า สำหรับนักพัฒนามาไม่ว่าจะเป็นทั้งในรูปแบบเดียวหรือสุ่มกึ่งตาม และนอกจากนั้นก็ยังสามารถเรียกดูได้ว่าใครเป็นคนเขียนหรือใครเป็นคนแก้ไข Project ในส่วนต่างๆ

- Github

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

### 2.2.8 Testing

- Jest

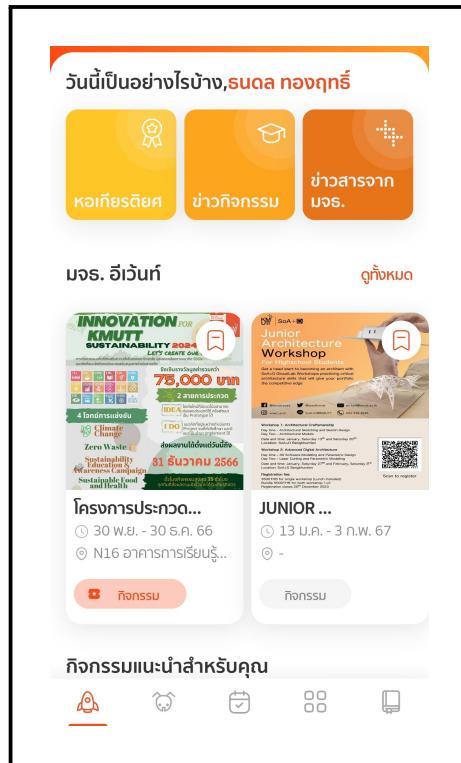
JavaScript Framework สำหรับเขียน Test เป็น Open Source ที่พัฒนาโดย Facebook ซึ่งมี helper มี function ต่างๆ ให้ใช้ ทำให้ง่ายต่อการเขียน Test สามารถเขียนเทสได้ทั้ง React, Vue, Angular หรือ JavaScript ทั่ว ๆ ไป

- PyTest

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

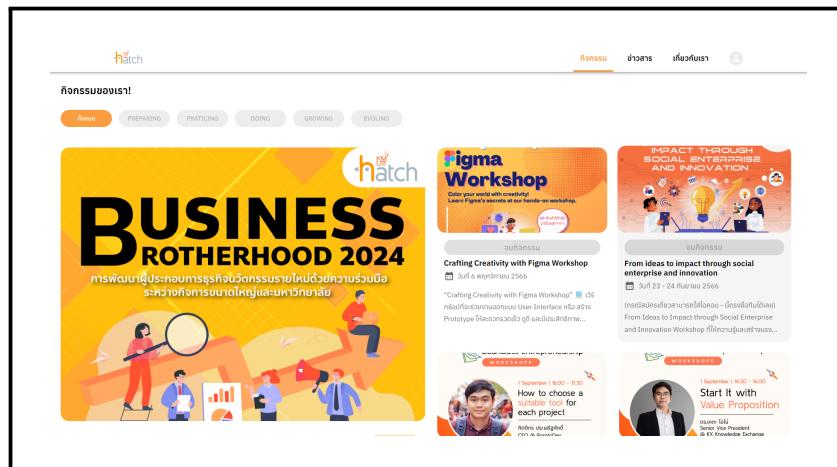
### 2.2.9 Product Survey

ModLink คือแอปพลิเคชันของทางมหาวิทยาลัยโดยข้อมูลภายใน แอปพลิเคชันนั้น จะเกี่ยวกับข้อมูลกิจกรรมต่าง ๆ ภายในมหาวิทยาลัย และข้อมูลส่วนตัวของนักศึกษา อย่างไรก็ตาม feature การแสดงข้อมูลเกี่ยวกิจกรรมของแอปพลิเคชันนั้นงดีกว่าทำได้ไม่ค่อยดีนัก เนื่องจากเป็นการกระจายข่าวสาร แบบทั่วไป ไม่ได้แบ่งแยกประเภทหรือแสดงตามที่ผู้ใช้ให้ความสนใจ



รูปที่ 2.10 หน้าแนะนำกิจกรรมของ ModLink และรายละเอียด

KMUTT Hatch คือ เว็บไซต์สำหรับนักศึกษาและศิษย์เก่าเพื่อประชาสัมพันธ์กิจกรรมของ Hatch ทางเว็บไซต์มีการประชาสัมพันธ์กิจกรรม ต่างๆ และข้อมูลที่จำเป็นต่อผู้ใช้งาน อย่างไรก็ตามเว็บไซต์นี้สามารถประชาสัมพันธ์ได้แค่กิจกรรมที่ทาง Hatch เป็นผู้จัดเท่านั้น



รูปที่ 2.11 หน้าแนะนำกิจกรรมของ KMUTT Hatch

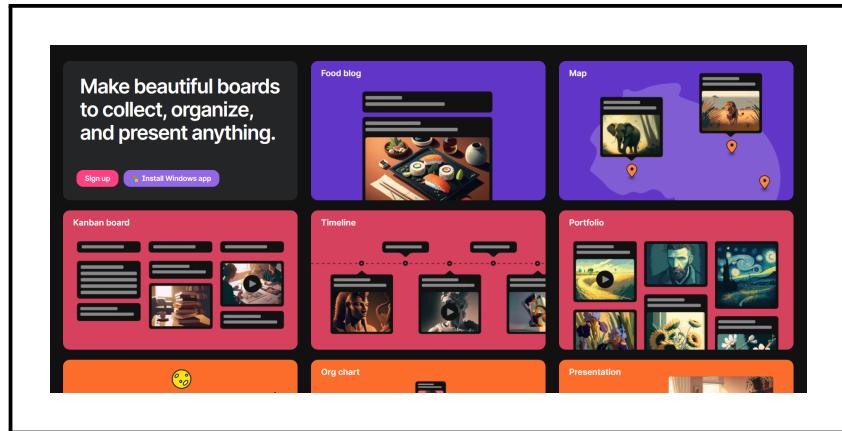
KMUTT Sinfo คือ เว็บไซต์สำหรับนักศึกษาที่ทำการรวบรวมระบบจัดการงานต่าง ๆ ของนักศึกษาไม่ว่าจะเป็นการลงทะเบียนเรียน ดูเกรด หรือประเมินกิจกรรม ซึ่งแม้ว่างานต่าง ๆ ของนักศึกษานั้นจะมีศูนย์กลางมาที่เว็บไซต์นี้ แต่ถึงกระนั้นก็เป็นเว็บที่ใช้งานไม่ค่อยสะดวก เนื่อง-

จากต้องเข้าผ่าน portal up ซึ่งต้องอาศัยการตั้งค่าและยังใช้งานไม่ได้ในบาง platform อีกทั้งยังจำกัดเวลาที่ใช้งานเอาไว้และแม้จะเป็นศูนย์รวมประวัติการทำกิจกรรมของนักศึกษา แต่กลับไม่สามารถແນະนำได้ว่านักศึกษานั้นให้ความสนใจกิจกรรมใด



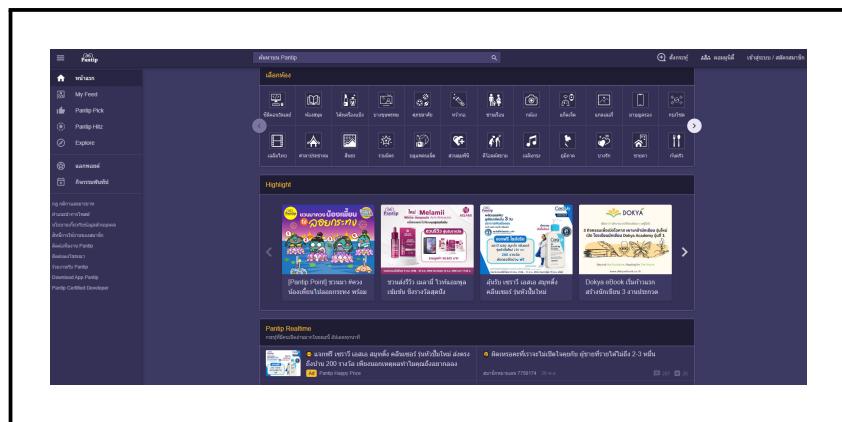
รูปที่ 2.12 หน้า Mainpage ของ KMUTT Sinfo

Padlet คือแอปพลิเคชันหรือเว็บไซต์ที่อยู่ในแพลตฟอร์มสำหรับ การระดมความคิด แสดงความคิดเห็น หรือแลกเปลี่ยนความรู้ร่วมกัน ผ่านกระดาษดิจิทัลในรูปแบบเสมือน Post it ที่ติดบนบอร์ด ซึ่งจะแสดงผลทุกอย่างแบบ Real-time สามารถโพสต์ทั้งในรูปแบบข้อความ รูปภาพ และวิดีโอของเว็บไซต์ได้ เร็บเพลที่จะให้ผู้ใช้งานแสดงความเห็น หรือโพสต์ข้อมูลงบนเว็บ ซึ่งจะต่างจากตรงที่ทางระบบของจะเป็น ตัวกลางในการคำนวนหาจากความชื่นชอบของผู้ใช้จากการรวมที่ผู้ใช้เคยได้เข้าร่วม หรือชุมชนที่ผู้ใช้สนใจอยู่



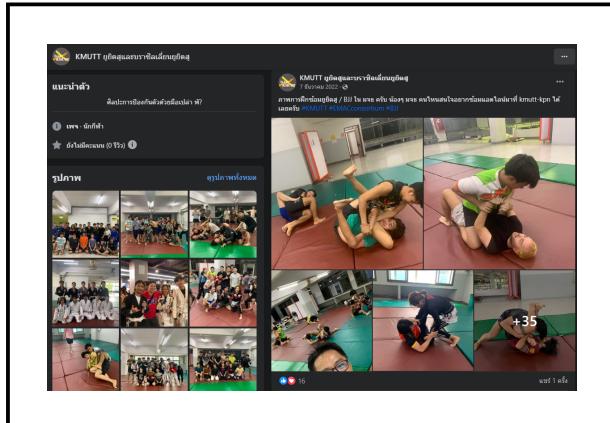
รูปที่ 2.13 หน้า Mainpage ของ Padlet

Pantip คือ พื้นที่สำหรับการแลกเปลี่ยนความคิดเห็นในหัวข้อ หรือประเด็นที่สนใจร่วมกัน สามารถสอบถาม บอกเล่าแบ่งปันประสบการณ์ ในเรื่องต่าง ๆ ในหน้ากระดาษสนทนadoโดยสมาชิกสามารถตั้ง หรือตอบกระทู้ต่าง ๆ ที่สมาชิกสนใจและสามารถเลือกหาอ่านข้อมูลได้จาก ป้ายหัวข้อในเรื่องต่าง ๆ ที่ทางเว็บไซต์ ได้สร้างขึ้นไว้ ซึ่งจะต่างจากตรงที่ทางระบบของ จะเป็นตัวกลางในการคำนวนหาจากความชื่นชอบ ของผู้ใช้จากการรวมที่ผู้ใช้เคยได้เข้าร่วม หรือชุมชนที่ผู้ใช้สนใจอยู่



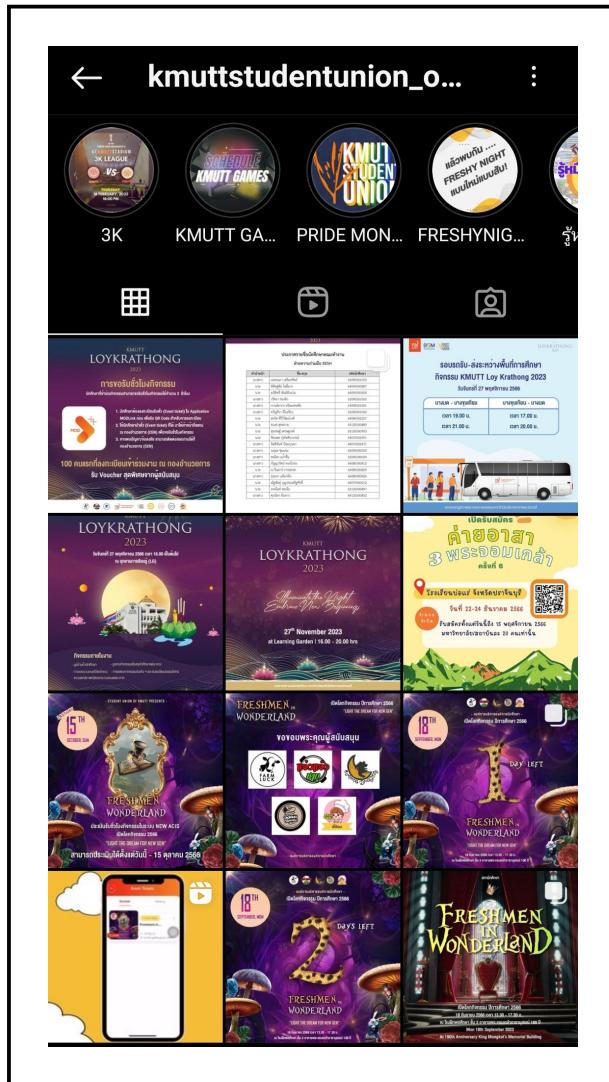
รูปที่ 2.14 หน้าแนะนำกระทุกของ Pantip

Facebook เป็น social media ที่ได้รับความนิยมที่สามารถใช้งาน ได้ในหลาย platform ซึ่งสามารถนำเสนอด้วยข้อมูลข่าวสารให้ผู้ใช้งานได้ มากมายและเป็น social media ที่มีคนใช้งานแบบตลอดทั้งวัน ทำให้บางชุมชนเลือก ที่จะทำหน้าเพจเพื่อกระจายข่าวสารเกี่ยวกับชุมชน ของตัวเอง อย่างไรก็ตามด้วยบริษัทของ facebook ทำให้ข่าวสารของชุมชนมักโดดเด่นด้วย ข่าวอื่น ๆ อยู่เสมอ ถึงแม้จะ สามารถเข้าไปสู่หน้าเพจเพื่อคุ้มครองเคลื่อนไหวได้แต่ก็ไม่สามารถแนะนำด้วยชุมชนหรือกิจกรรมที่ชุมชนจะจัดให้แก่นักศึกษาที่ไม่ติดตามเพจ ได้อยู่ดี



รูปที่ 2.15 หน้า Homepage ของเพจชุมชนใน Facebook

Instagram คือ แอปพลิเคชันบน smartphone และอุปกรณ์คอมพิวเตอร์ โดยแอปพลิเคชันนี้จะเน้นการแชร์รูปภาพ บน Social Network ซึ่งทำให้เพื่อน ของคุณสามารถเห็นภาพถ่ายของคุณได้และยังสามารถคอมเมนต์ภาพของคุณได้ ที่สำคัญ Instagram ยังสามารถแชร์ภาพ ของคุณไปยัง Twitter และ Facebook ได้อีกด้วย ยังสามารถกดติดตามบุคคลที่ชื่นชอบเพื่อที่จะได้เห็นโพสต์รูปภาพ วิดีโอ ของบุคคลนั้น ๆ ได้อีกด้วย และยังมีไฟล์เรอร์ story ที่ใช้ในการอัพรูปภาพ วิดีโอดอกลิปของคุณได้ด้วย



รูปที่ 2.16 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 2.17 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 2.18 ระบบการแนะนำกิจกรรม

## บทที่ 3 วิธีการทำงาน กระบวนการและการออกแบบ

### 3.1 บทนำ

#### 3.1.1 สำรวจความต้องการของผู้ใช้เชิงคุณภาพ

ในการทำระบบเพื่อแจ้งเตือนข่าวสารของกิจกรรมและชุมชน ทางผู้จัดทำระบุผู้ใช้งานและผู้ได้รับประโยชน์เป็น 3 กลุ่มใหญ่ๆด้วยกัน คือ ผู้เข้าร่วมกิจกรรม ผู้จัดกิจกรรม และประธานชุมชน โดยแต่ละกลุ่มมีความต้องการดังนี้ ผู้เข้าร่วมกิจกรรมซึ่งเป็นผู้ใช้งานหลัก คือ นักศึกษาทั่วไปในมหาวิทยาลัย โดยมีความต้องการพื้นฐานคือการที่อยากรู้จะทำกิจกรรมตามความสนใจ แต่ด้วยภาระการเรียนทำให้ส่วนใหญ่ไม่มีเวลาที่จะหา กิจกรรมหรือ ชุมชนที่ตนเองสนใจ สิ่งที่ต้องการจะเป็นแหล่งที่รวมรวม ข้อมูลข่าวสาร กิจกรรม และ รายละเอียดของชุมชนต่างๆรวมถึงสิ่งที่ชุมชนนั้นทำเอาไว้ใน ที่เดียวเพื่อที่จะหา กิจกรรมและชุมชนตามที่ตัวเองต้องการได้ ผู้ได้รับประโยชน์กลุ่มแรก คือ ผู้จัดกิจกรรมที่แทนผู้จัดกิจกรรมที่เป็นเจ้าหน้าที่ของมหาวิทยาลัย โดยมีความต้องการพื้นฐานคือการประชาสัมพันธ์ กิจกรรมที่จัด แต่ข่าวสารของกิจกรรมที่ถูกเผยแพร่ผ่าน platform ต่างๆ เช่น facebook หรือ instargram นั่นเอง ผู้ที่เห็นการประชาสัมพันธ์เพียงบางส่วนเท่านั้น ทำให้ผู้เข้าร่วมกิจกรรมมีน้อยกว่าที่คาดหวัง ผู้ได้รับประโยชน์กลุ่มที่ 2 คือ ประธานชุมชน ที่แทนผู้ที่ดำเนินงานชุมชน ซึ่งส่วนใหญ่เป็นนักศึกษาที่ใกล้จะจบการศึกษา หรือนักศึกษาชั้นปริญญาโท โดยมีความต้องการพื้นฐาน คือ การประชาสัมพันธ์ชุมชนที่ตัวเองจัดการอยู่ ถึงแม้ว่าจะเป็นงานชุมชนจะมี การนัดหมายกันแบบปากต่อปากอยู่แล้ว แต่การจัดการกิจกรรมชุมชน ก็ไม่เป็นระบบเท่าที่ควร อีกทั้งการหาสมาชิกชุมชนใหม่ หรือ การประชาสัมพันธ์กิจกรรมชุมชนให้คนภายนอกชุมชนจากการที่ สมาชิกชุมชนเป็นคนเชี่ยวชาญก็ยังมีโอกาสที่จะมีคนเห็นการประชาสัมพันธ์มีน้อย ซึ่งจากการสัมภาษณ์ทำให้พบว่าปัญหาใหญ่ที่เกิดขึ้นเกิดจากการที่ไม่มีแหล่งที่จะกระจายข้อมูลที่ผู้เข้าร่วมกิจกรรมต้องการในการตัดสินใจ เข้าร่วมกิจกรรม และ ชุมชนได้อย่างเหมาะสม ทำให้กิจกรรม และ ชุมชนไม่ได้รับความสนใจเท่าที่ควร

#### 3.1.2 Journey Map

จากกลุ่มผู้ใช้งานทั้งสามกลุ่มสามารถแบ่งพฤติกรรมการใช้งานได้เป็น 2 ประเภท คือ การหา กิจกรรมหรือชุมชน

ผู้กระทำ: ผู้เข้าร่วมกิจกรรม

Step 1 : ผู้เข้าร่วมกิจกรรมไปตามรายละเอียดจากผู้จัดกิจกรรมหรือประธานชุมชน  
ปัญหา - ผู้เข้าร่วมอาจไม่รู้ว่ามีกิจกรรมหรือชุมชนนี้อยู่

- บางที่ผู้เข้าร่วมกิจกรรมก็ไม่รู้จักผู้จัดกิจกรรมหรือประธานชุมชน

Step 2 : ผู้เข้าร่วมลงที่เบียนกิจกรรมหรือชุมชน

Step 3 : ผู้เข้าร่วมเข้าร่วมกิจกรรมหรือทำ กิจกรรมชุมชน

ปัญหา - ผู้เข้าร่วมอาจไม่รู้ตำแหน่งของสถานที่จัดกิจกรรมหรือชุมชน

Step 4 : ผู้เข้าร่วมประเมินกิจกรรม

ปัญหา - ผู้เข้าร่วมไม่รู้ว่าแบบประเมินกิจกรรมสามารถประเมินได้แล้ว

#### 3.1.3 Stakeholder

ผู้ที่คาดว่าจะได้รับประโยชน์จากการใช้งานแอปพลิเคชันของผู้จัดทำสร้างขึ้นนั้นแบ่งเป็น 2 กลุ่ม คือ ผู้เข้าร่วมกิจกรรม และ ผู้จัดกิจกรรมที่หมายถึงผู้จัดกิจกรรมและประธานชุมชน  
ผู้เข้าร่วมกิจกรรม คือ นักศึกษาชั้นปีที่ 1 เนื่องจากเป็นกลุ่มที่ยังไม่ข้อมูลเกี่ยวกับสิ่งต่าง ๆ ในมหาวิทยาลัยอยู่น้อย ทำให้ยากที่จะหาแหล่งข้อมูลของกิจกรรมหรือชุมชนที่สนใจ

ผู้จัดกิจกรรม คือ นักศึกษาชั้นปีที่ 4 ที่เป็นประธานชมรมและผู้จัดกิจกรรมที่เป็นเจ้าหน้าที่ของมหาวิทยาลัย เนื่องจากเป็นผู้ได้รับผลกระทบจากการประชาสัมพันธ์กิจกรรมและชมรมที่ไม่มีประสิทธิภาพเท่าที่ควร

### 3.2 Requirement list

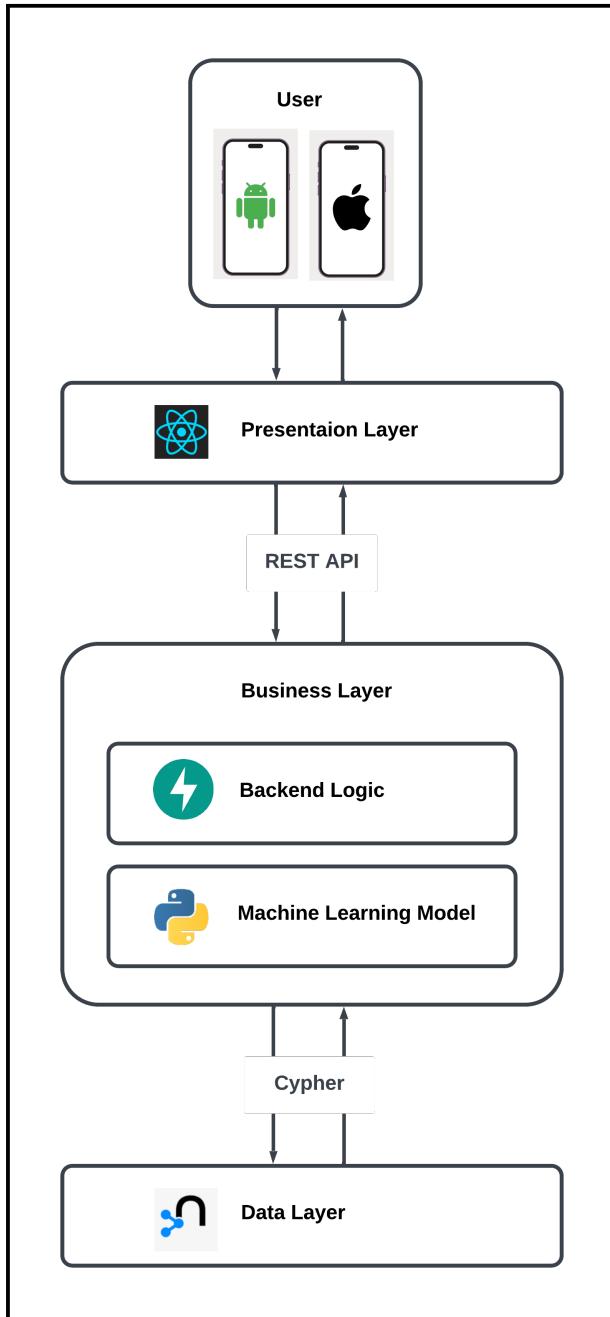
รายการข้อกำหนดหรือความต้องการที่จำเป็นต้องมีในโครงการหรือผลิตภัณฑ์ที่กำลังถูกพัฒนา ข้อกำหนดเหล่านี้ เป็นข้อมูลที่ถูกรวบรวมมาจากผู้ใช้, ลูกค้า, หรือผู้เกี่ยวข้องอื่น ๆ ซึ่งมีไว้เพื่อกำหนดขอบเขตและคุณลักษณะของ ผลิตภัณฑ์หรือโครงการ โดยในส่วนนี้ หลังจากที่ได้ทำการสำรวจความต้องการจาก Early Adopter แล้วทำให้เรา ได้ความต้องการมา ดังนี้

1. ข้อมูลของกิจกรรมและชมรมที่ชัดเจน
2. ระบบการค้นหา กิจกรรมและชมรม
3. การแนะนำชมรมและกิจกรรมที่น่าสนใจ
4. การแจ้งเตือนการประเมินกิจกรรม

### 3.3 Feature list

จากการวิเคราะห์ Requirement ทั้งหมด เราได้ทำการวิเคราะห์ feature มาเพื่อตอบโจทย์ความต้องการของผู้ใช้ ดังนี้

1. Login: ลงชื่อเพื่อเข้าใช้งานแอพพลิเคชัน
2. Registration: สมัครบัญชีของแอพพลิเคชันด้วยอีเมลมหาวิทยาลัย
3. Logout: ออกจากระบบ
4. Search: ค้นหา กิจกรรมหรือชมรมที่สนใจ
5. Select: เลือกอ่านรายละเอียด กิจกรรมหรือชมรมที่สนใจ
6. Join:
  - ลงชื่อเข้าร่วม กิจกรรมที่สนใจ
  - ลงชื่อเป็นสมาชิกของชมรมที่สนใจ
7. Resignation: ถอนชื่อจากการเป็นสมาชิกชมรม
8. Notification:
  - แจ้งเตือน กิจกรรมที่เกี่ยวข้องกับชมรมหรือความสนใจของนักศึกษา
  - แจ้งเตือน การประเมิน กิจกรรม
9. Event evaluation: ประเมิน กิจกรรม
10. Recommendation: แนะนำ กิจกรรมและชมรม ตามความสนใจของผู้ใช้โดยอ้างอิงจาก tag ของ กิจกรรม



รูปที่ 3.1 Architecture diagram

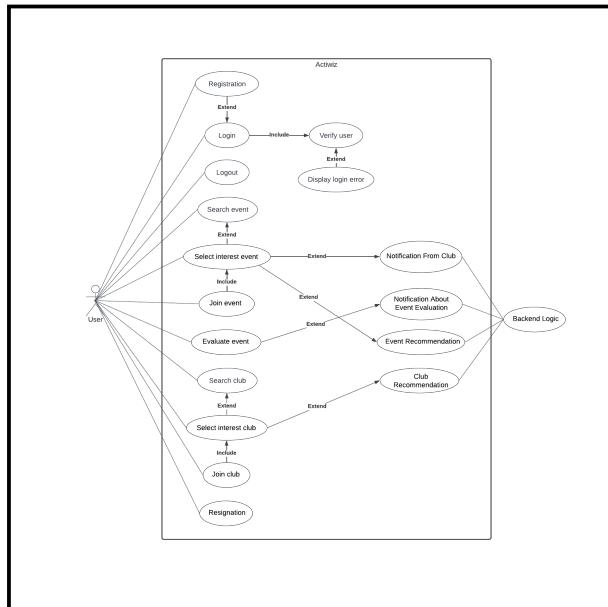
### 3.4 Architecture diagram

#### System Flow

1. User ใช้งาน React Native Application
2. React Native ส่ง HTTP request ตามรูปแบบของ Rest API ที่กำหนดไว้ ไปยัง FastAPI backend เพื่อเรียกใช้งานฟังก์ชันต่าง ๆ เช่น ค้นหา긱ิจกรรมหรือดูข้อมูลเกี่ยวกับชุมชน 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 Use case diagram and use case narratives



รูปที่ 3.2 Use case diagram and use case narratives

#### Use case narratives

use case diagram นี้เป็นการแสดงการทำงานของระบบและการใช้งานจากผู้ใช้งาน ใช้ case diagram ในการแสดงภาพรวมของระบบและลำดับการทำงานของฟังก์ชันใช้งานในระบบ โดยมีผู้กระทำหลักคือ User และผู้กระทำรองคือ Backed Logic

### Registration scenario

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

ตารางที่ 3.1 Registration scenario

### Login scenario

Actor	User
Actor	User
Goal	เข้าสู่ระบบและใช้งานฟังก์ชันต่างได้
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 Login scenario

### Logout scenario

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

ตารางที่ 3.3 Logout scenario

### Search event scenario

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

ตารางที่ 3.4 Search event scenario

### Select event scenario

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

ตารางที่ 3.5 Select event scenario

### Join event scenario

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

ตารางที่ 3.6 Join event scenario

### Evaluate event scenario

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

ตารางที่ 3.7 Evaluate event scenario

### Search club scenario

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

ตารางที่ 3.8 Search club scenario

### Select club scenario

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

ตารางที่ 3.9 Select club scenario

### Join club scenario

Actor	User
Goal	เข้าร่วมชมรม
Pre-conditions	<ul style="list-style-type: none"> <li>- User ต้องเข้าสู่ระบบก่อน</li> <li>- User ต้องเข้าสู่หน้าอ่านรายละเอียดของชมรมก่อน</li> </ul>
Main success scenario	1.User กดเข้าร่วมชมรม
Main success scenario	<ul style="list-style-type: none"> <li>1.User ได้รับแจ้งเตือนการประเมินกิจกรรม</li> <li>2.User กดเข้าประเมินกิจกรรม</li> <li>3.ระบบพาไปหน้าประเมินกิจกรรม</li> </ul>

ตารางที่ 3.10 Join club scenario

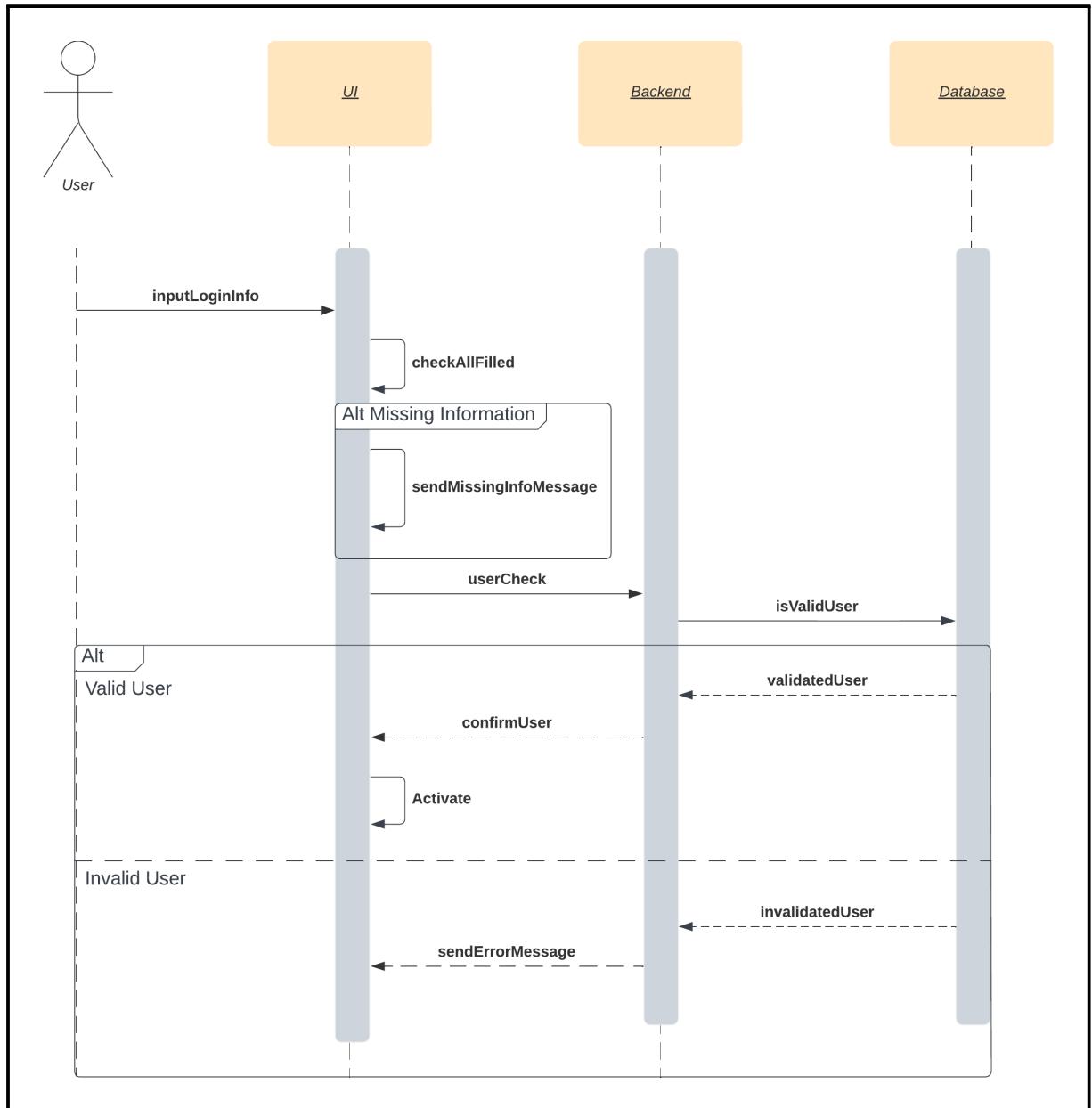
### Resignation scenario

Actor	User
Goal	เข้าร่วมชมรม
Pre-conditions	<ul style="list-style-type: none"> <li>- User ต้องเข้าสู่ระบบก่อน</li> <li>- User ต้องเข้าสู่หน้าอ่านรายละเอียดของชมรมก่อน</li> </ul>
Main success scenario	1.User กดเข้าร่วมชมรม
Main success scenario	<ul style="list-style-type: none"> <li>1.User ได้รับแจ้งเตือนการประเมินกิจกรรม</li> <li>2.User กดเข้าประเมินกิจกรรม</li> <li>3.ระบบพาไปหน้าประเมินกิจกรรม</li> </ul>

ตารางที่ 3.11 Resignation scenario

### 3.6 Sequence diagram

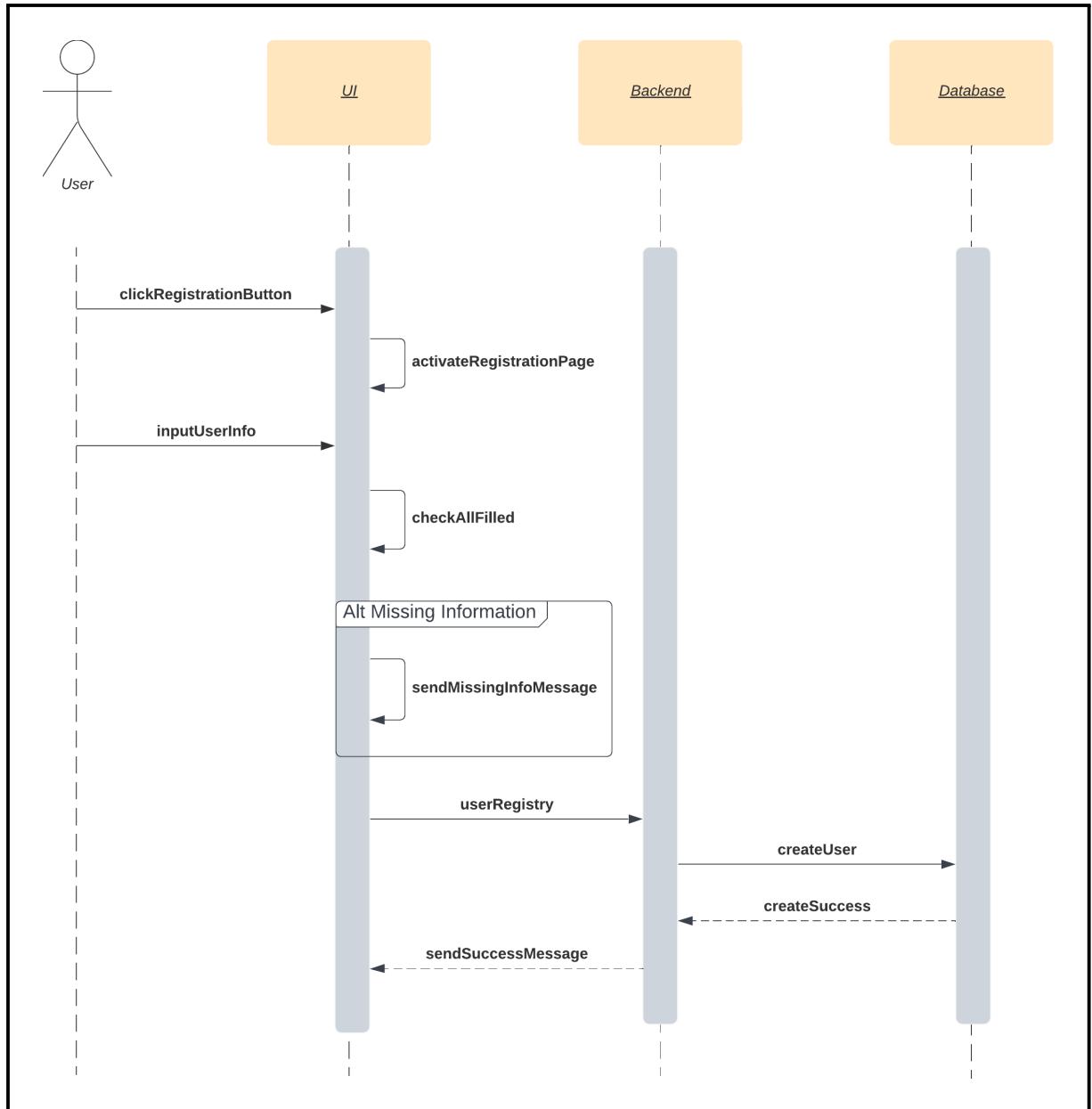
#### 3.6.1 Login



รูปที่ 3.3 Sequence diagram ของ login

Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานทำการเข้าสู่ระบบ โดยให้ผู้ใช้งานกรอกข้อมูลที่ใช้ในการเข้าสู่ระบบลงในหน้า log in ของ UI ตัว UI จะทำการตรวจสอบข้อมูลที่ต้องการใช้ในการเข้าสู่ระบบว่ามีส่วนใดที่ขาด จากนั้นจึงส่งต่อให้ระบบ Backend ที่จะเรียกข้อมูลจาก Data base เพื่อตรวจสอบผู้ใช้งาน และจึงเปิดหน้า UI ในกรณีที่ยืนยันได้ว่ามีผู้ใช้งานนั้นอยู่จริง

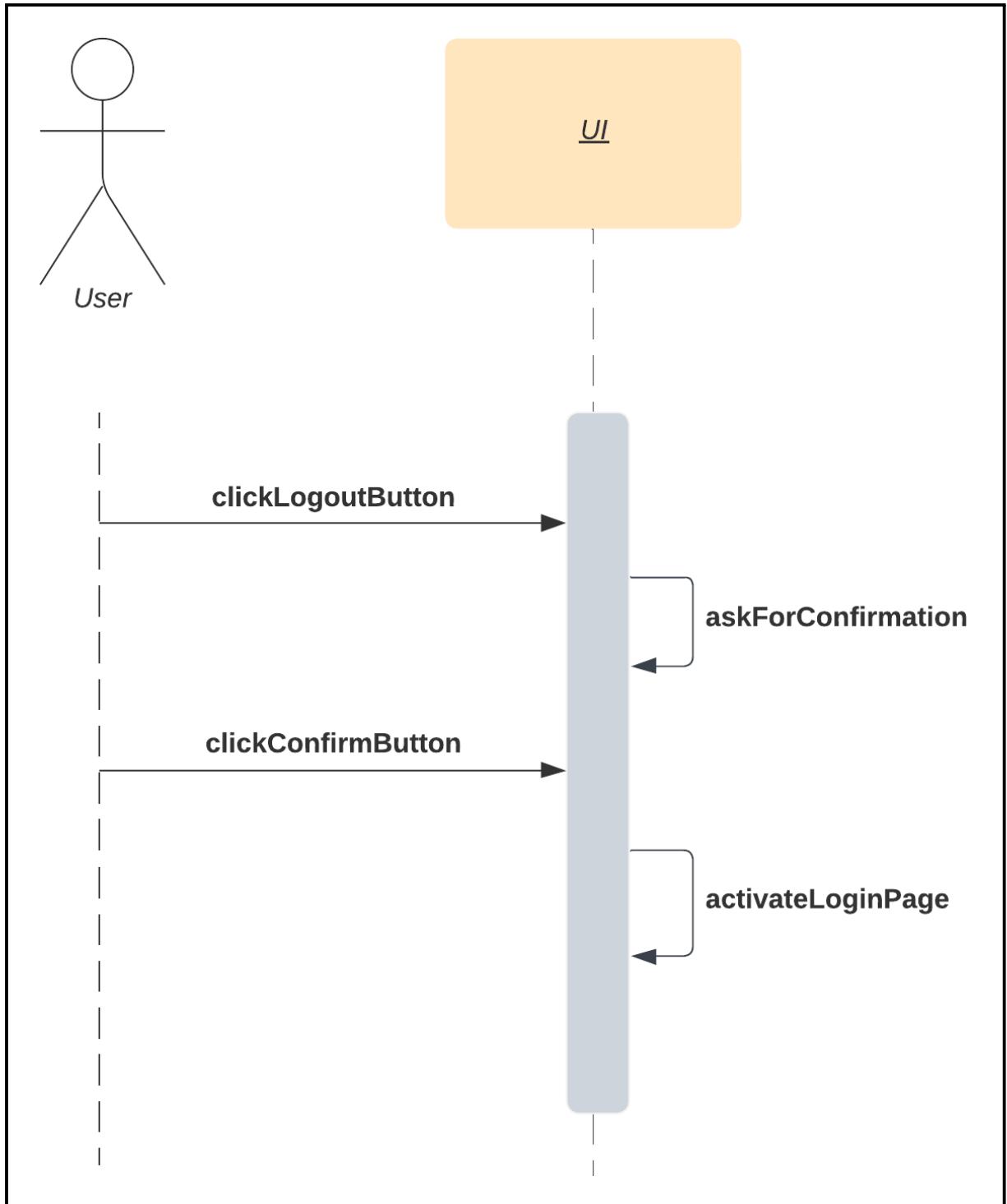
### 3.6.2 Registration



รูปที่ 3.4 Registration

Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานทำการสมัครเข้าใช้งาน โดยหลังจากผู้ใช้งาน กดตุ่มเพื่อสมัครใช้งานแล้วระบบ UI จะขึ้นหน้าให้กรอกข้อมูลที่ใช้ในการสมัครเข้าใช้งาน ตัว UI จะทำการตรวจสอบข้อมูลที่ต้องการใช้ในการเข้าสู่ระบบว่ามีส่วนใดที่ขาด จากนั้นจึงส่งต่อให้ระบบ Backend ซึ่งจะส่งข้อมูลไปบันทึกที่ Data base และระบบจะส่งข้อความยืนยันการสมัครเข้าใช้งานกลับขึ้นไปหาผู้ใช้งานที่สมัคร

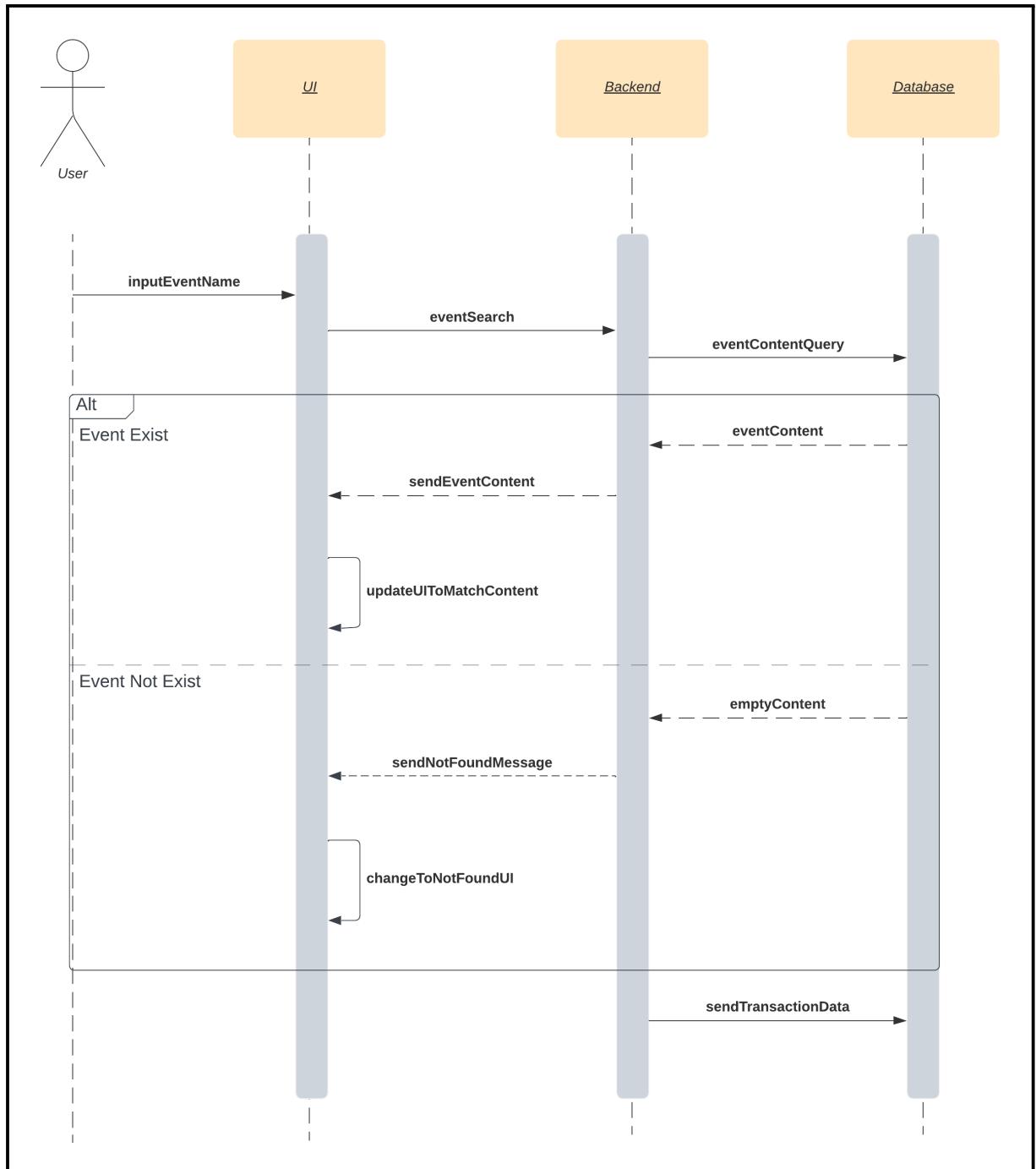
### 3.6.3 Logout



รูปที่ 3.5 Logout

Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานออกจากระบบ โดยหลังจากผู้ใช้งานกดปุ่มออกจากระบบ UI จะทำการยืนยันการออกจากระบบซึ่งหลังจากผู้ใช้งานกดยืนยันแล้วระบบจะพาผู้ใช้ไปยังหน้าเข้าสู่ระบบอีกครั้ง

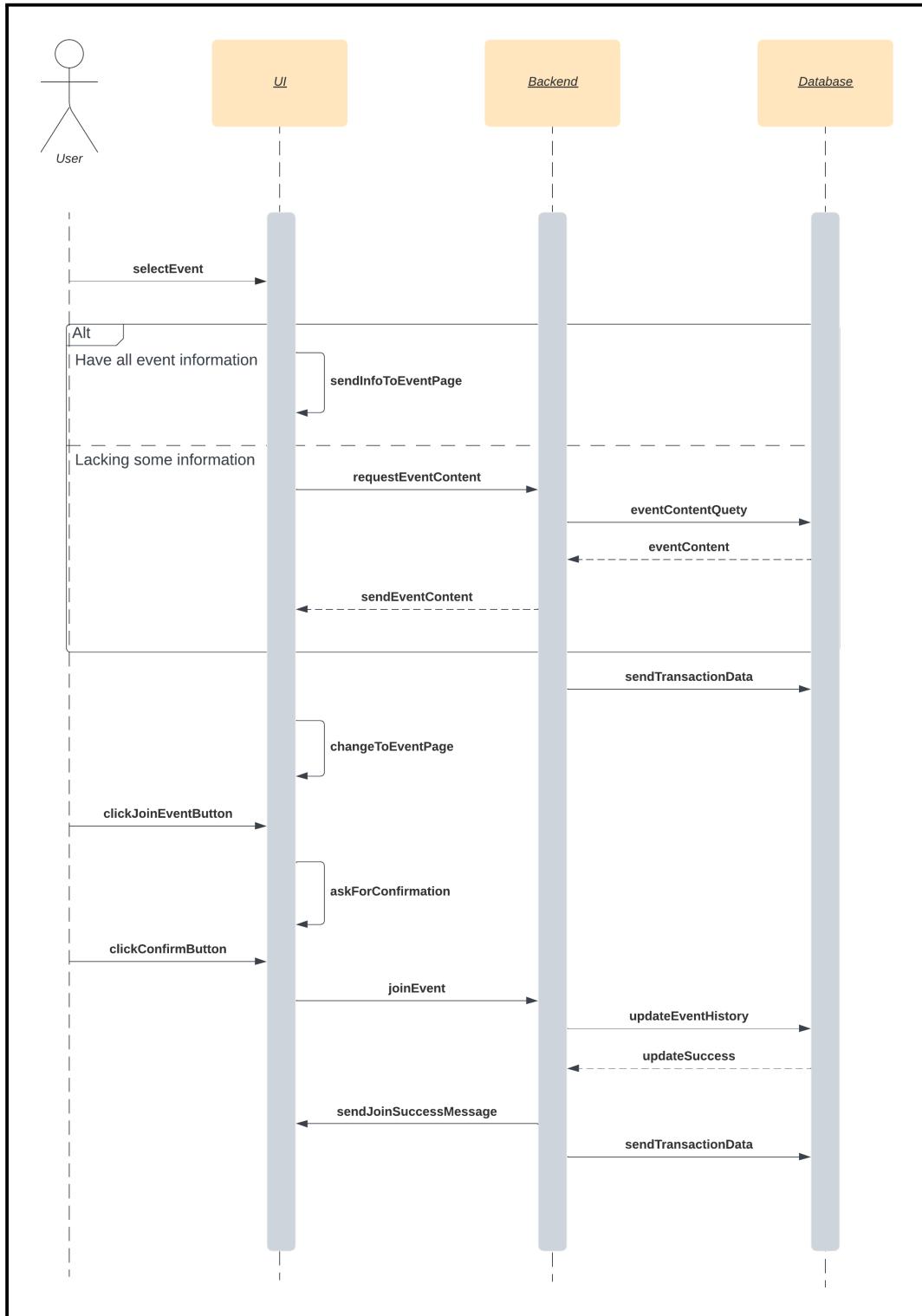
### 3.6.4 Search Event



รูปที่ 3.6 Search Event

Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานทำการค้นหาภารกิจกรรม โดย UI จะรับสิ่งที่ผู้ใช้งานต้องการค้นหาส่งไปหา Backend เพื่อให้ Backend ค้นหาข้อมูลที่เก็บไว้จาก Data base หากพบเจอจะส่งข้อมูลกลับไปหาผู้ใช้งาน หากไม่เจอจะแสดงข้อความว่าไม่พบภารกิจกรรม และส่งข้อมูลการใช้งานไปเก็บยัง Data base

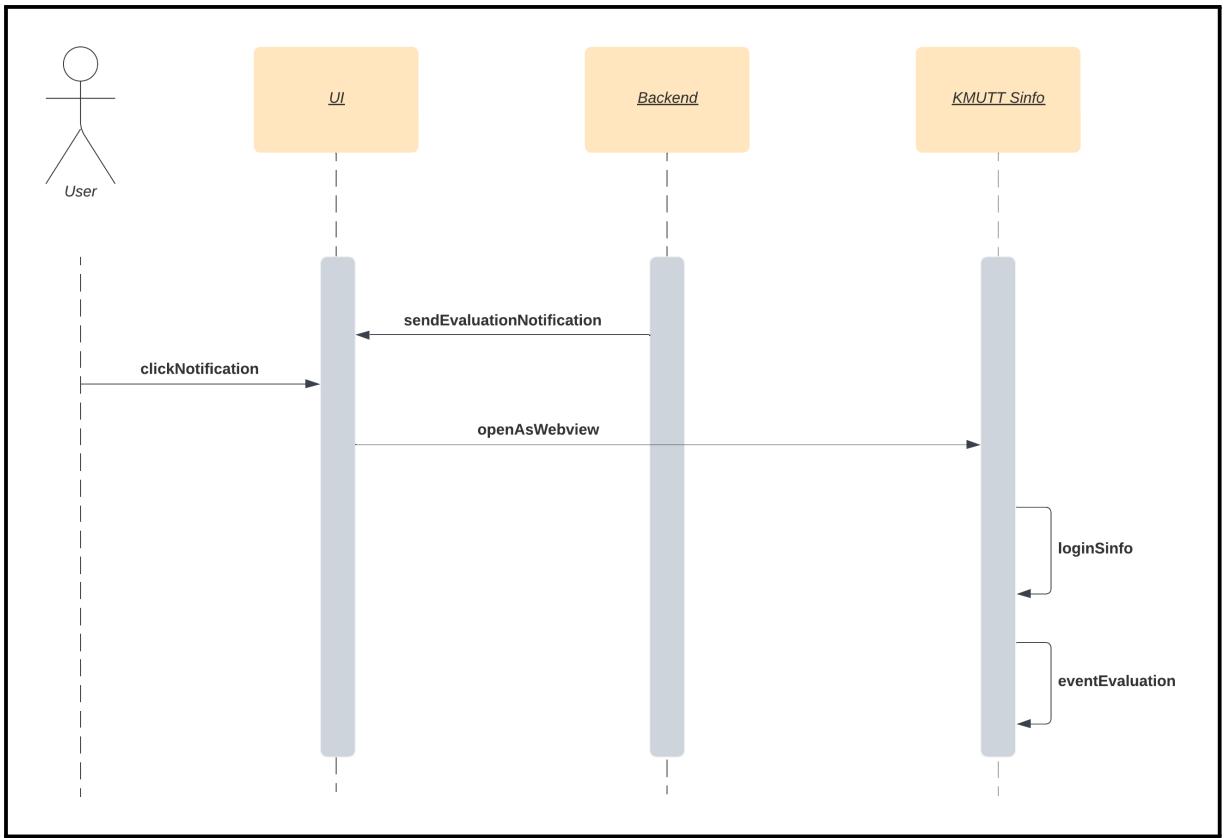
### 3.6.5 Select and join event



รูปที่ 3.7 Select and join event

Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานทำการเข้าไปอ่านรายละเอียดของกิจกรรม โดย UI จะแสดงข้อมูลหากมีข้อมูลของกิจกรรมนั้นอยู่แล้ว หากไม่มีจะส่งคำขอให้ Back end ดึงข้อมูลจาก Data base ให้ หากผู้ใช้งานทำการกดเข้าร่วมกิจกรรม UI จะทำการขอการยืนยันจากผู้ใช้งานอีกครั้ง ซึ่งหากกดยืนยันแล้ว ก็จะส่งข้อมูลการเข้าร่วมไปให้ Back end จัดเก็บใน Data base

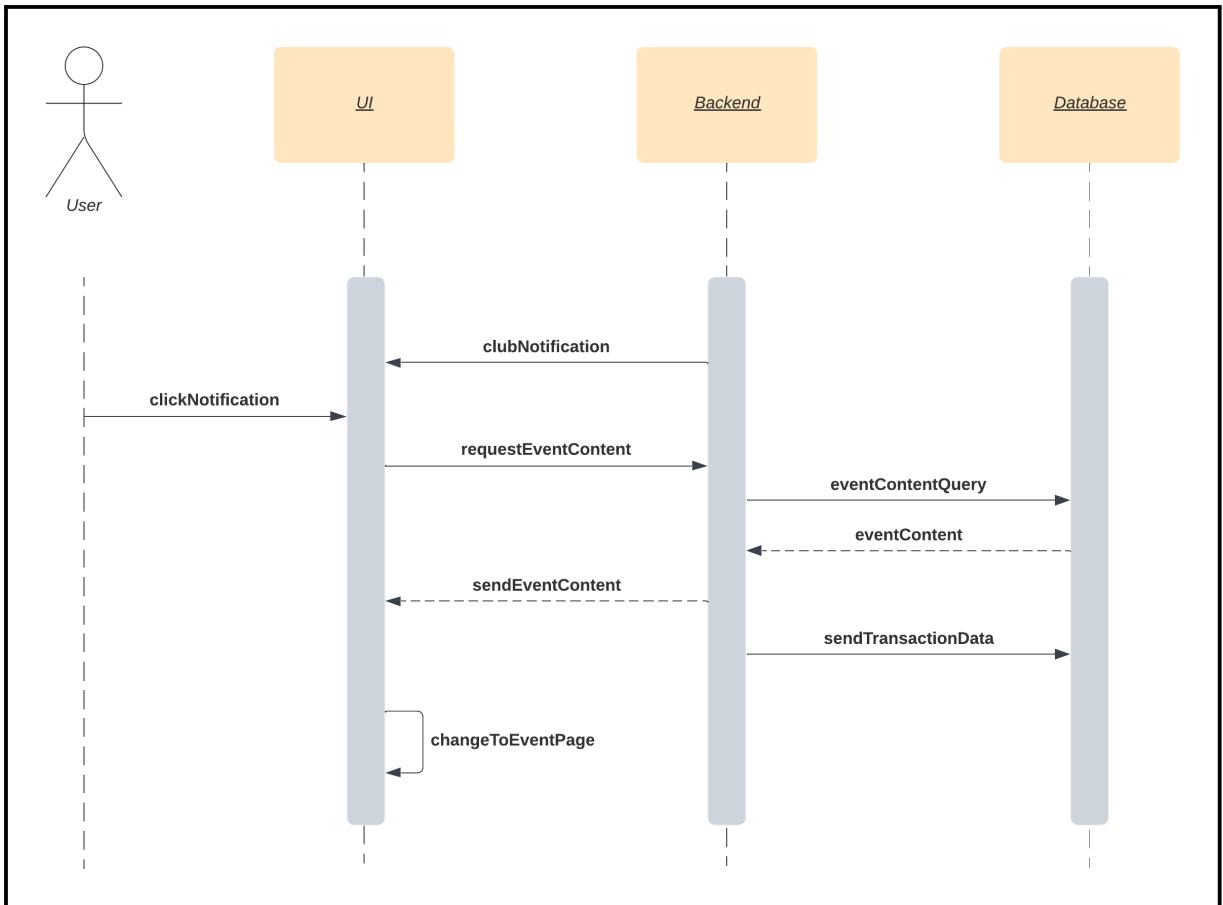
### 3.6.6 Event evaluation



รูปที่ 3.8 Event evaluation

Sequence diagram สำหรับแสดงการทำงานของระบบแจ้งเตือนกิจกรรมชุมชน โดย Backend จะส่งคำเตือนเข้ามาหา UI เพื่อที่จะแจ้งถึงกิจกรรมที่ชุมชนจัดขึ้น เมื่อผู้ใช้งานกดเข้าอ่านรายละเอียดกิจกรรมแล้ว UI จะขอให้ Backend หาข้อมูลกิจกรรมจาก Data base มาแสดงให้ผู้ใช้งาน และส่งข้อมูลการใช้งานไปเก็บยัง Data base

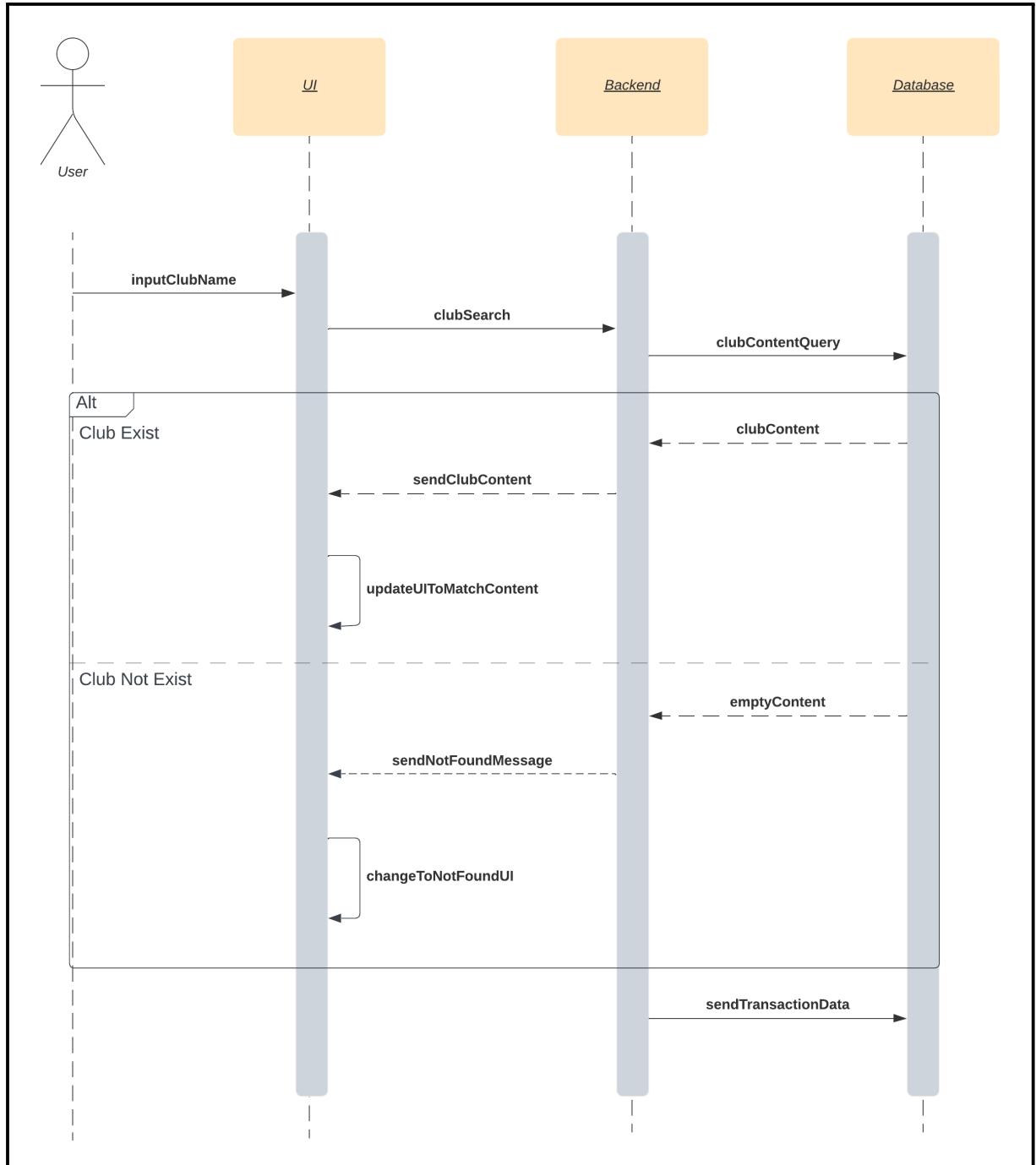
### 3.6.7 Notification about event from club



รูปที่ 3.9 Notification about event from club

Sequence diagram สำหรับแสดงการทำงานของระบบแจ้งเตือนการประเมินกิจกรรม โดย Backend จะส่งคำเตือนเข้ามาหา UI เพื่อที่จะแจ้งเตือนการประเมินกิจกรรม เมื่อผู้ใช้งานกดเข้าลิงค์ประเมินกิจกรรมแล้วจะเปิดหน้าต่างเพื่อประเมินกิจกรรมให้

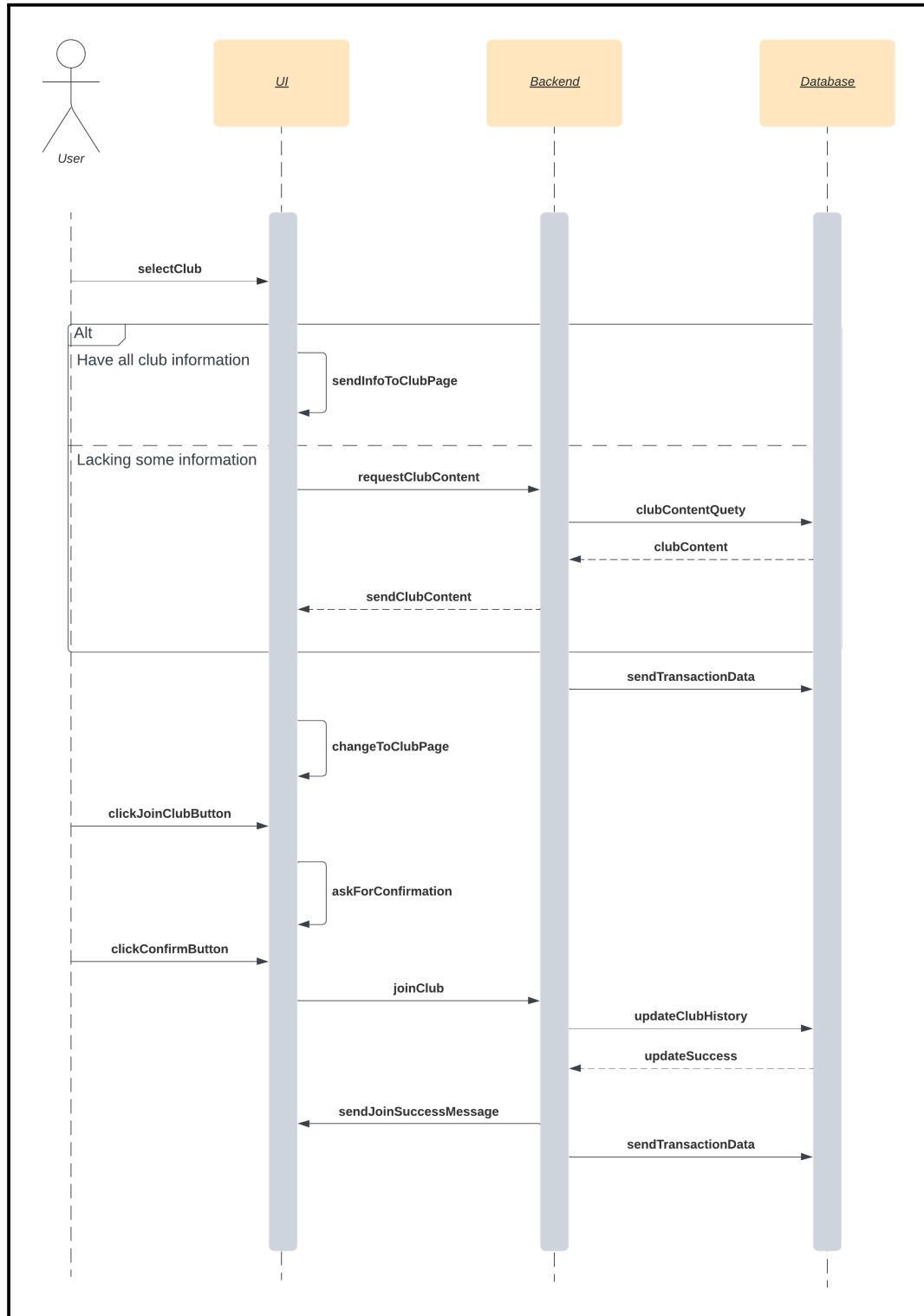
### 3.6.8 Seacrh club



รูปที่ 3.10 Seacrh club

Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานทำการค้นหาชื่อร่มโดย UI จะรับสิ่งที่ผู้ใช้งานต้องการค้นหาส่งไปหา Backend เพื่อให้ Backend ค้นหาข้อมูลที่เก็บไว้จาก Data base หากพบเจอจะส่งข้อมูลกลับไปหาผู้ใช้งาน หากไม่เจอจะแสดงข้อความว่าไม่พบกิจกรรม และส่งข้อมูลการใช้งานไปเก็บยัง Data base

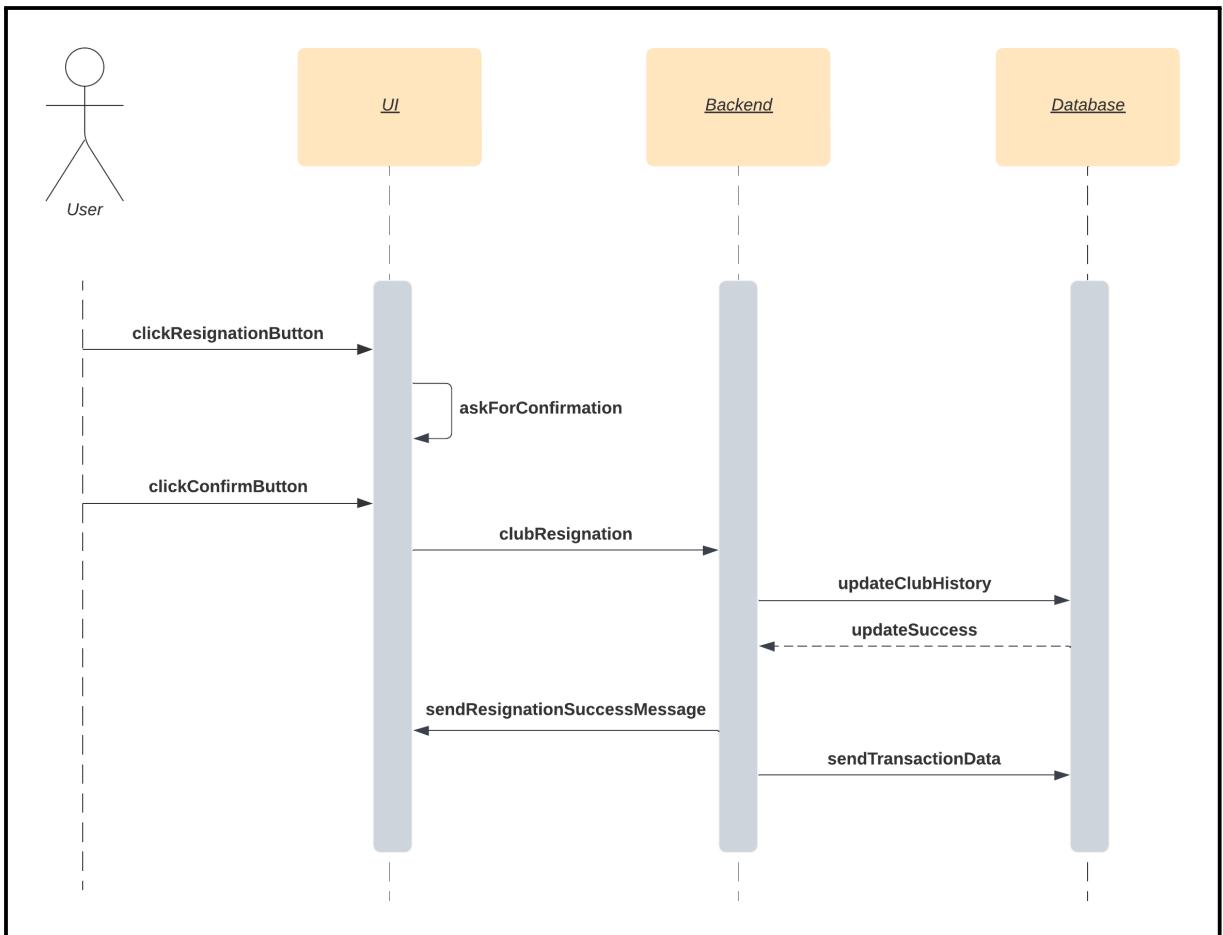
### 3.6.9 Select and join club



รูปที่ 3.11 Select and join club

Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานทำการเข้าไปอ่านรายละเอียดของชมรม โดย UI จะแสดงข้อมูลหากมีข้อมูลของกิจกรรมนั้นอยู่แล้ว หากไม่มีจะส่งคำขอให้ Back end ดึงข้อมูลจาก Data base ให้ หากผู้ใช้งานทำการกดเข้าร่วมกิจกรรม UI จะทำการขอการยืนยันจากผู้ใช้งานอีกครั้ง ซึ่งหากกดยืนยันแล้วก็จะ ส่งข้อมูลการเข้าร่วมไปให้ Back end จัดเก็บใน Data base

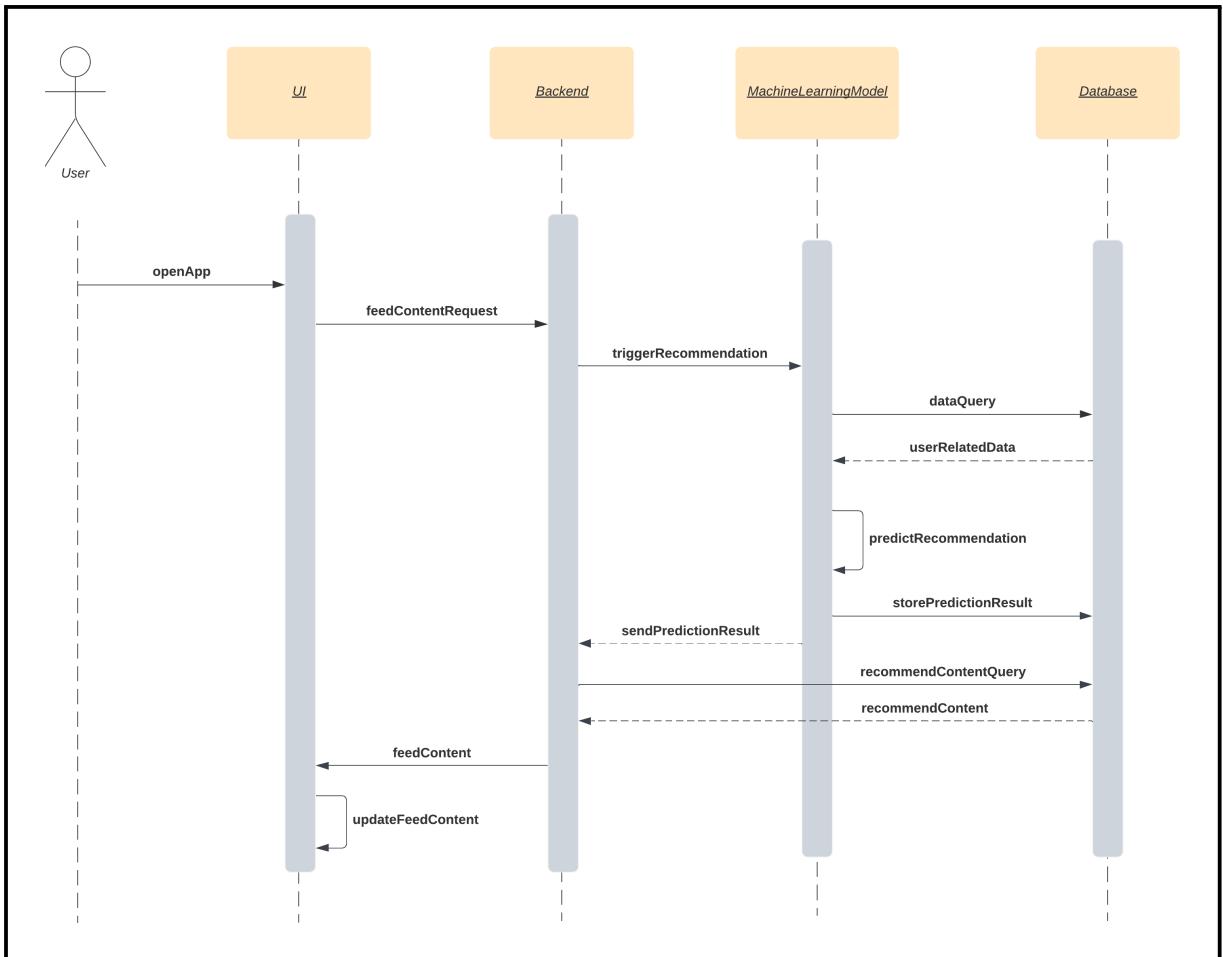
### 3.6.10 Club resignation



รูปที่ 3.12 Club resignation

Sequence diagram สำหรับแสดงการทำงานของระบบขณะที่ผู้ใช้งานทำการสมัครเข้าชมรม โดยหลังจากกดปุ่มสมัคร UI จะถามยืนยันการสมัคร หากผู้ใช้งานกดยืนยัน UI จะทำการส่งข้อมูลการใช้งานไปให้ Backend จัดเก็บใน Data base และส่งข้อความยืนยันการสมัครไปให้ UI และแสดงกับผู้ใช้งาน

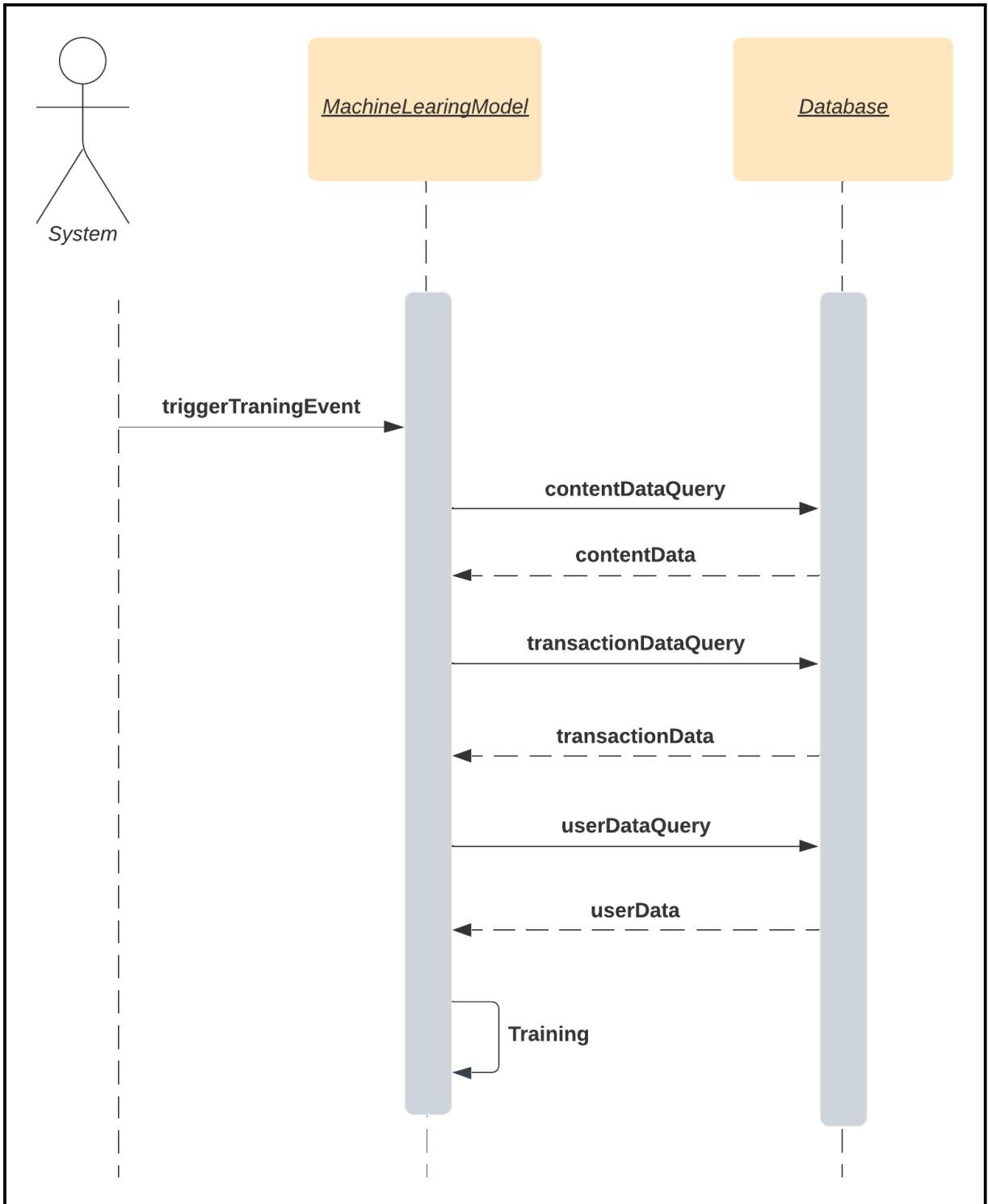
### 3.6.11 Feed recommendation



รูปที่ 3.13 Feed recommendation

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

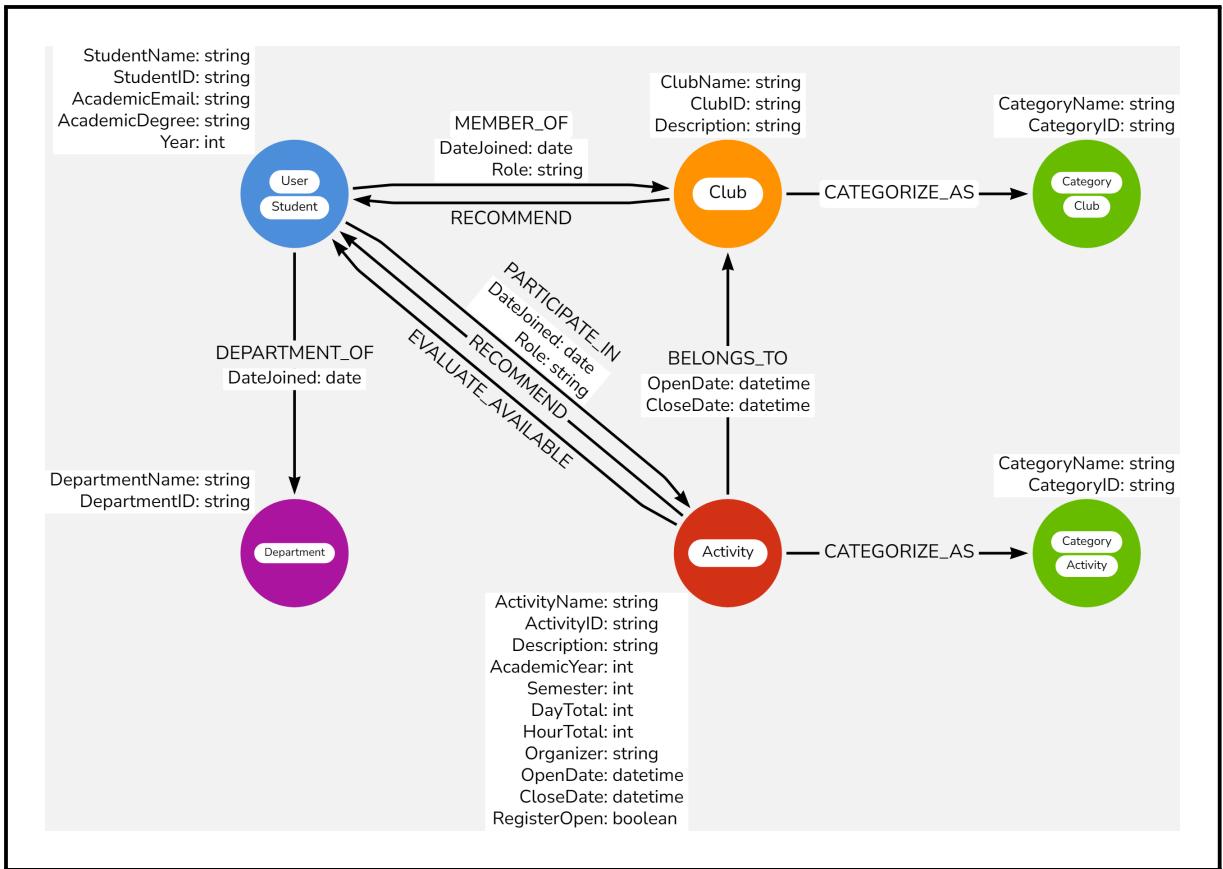
### 3.6.12 Machine learning training



รูปที่ 3.14 Machine learning training

Sequence diagram สำหรับแสดงการทำงานของระบบในขณะที่ Train Machine Learning Model ใหม่ โดย Machine Learning Model จะดึงข้อมูลกิจกรรมและการใช้งานมาจาก Data base แล้วทำการ Train ตัวเอง

### 3.7 Database Model



รูปที่ 3.15 Database Model