



ACTIWIZ: APPLICATION FOR RECOMMENDED EVENT AND CLUB FOR KMUTT STUDENT

ACTIWIZ: แอปพลิเคชันสำหรับการแนะนำกิจกรรมและชมรมสำหรับนักศึกษามหาวิทยาลัยเทคโนโลยีราชมงคลกรุงเทพ

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

Actiwiz: Application for recommended event and club for KMUTT student
Actiwiz: แอปพลิเคชันสำหรับการแนะนำกิจกรรมและชมรมสำหรับนักศึกษามหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

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

Project Title	Actiwiz: Application for recommended event and club for KMUTT student Actiwiz: แอปพลิเคชันสำหรับการแนะนำกิจกรรมและชมรมสำหรับนักศึกษามหาวิทยาลัย เทคโนโลยีพระจอมเกล้าธนบุรี
Credits	3
Member(s)	Mr. Kunanon Supmamul 63070501011 Mr. Naphattarak Muentoey 63070501018 Mr. Thanadol Thongrit 63070501029
Project Advisor	Asst.Prof. Rajchawit Sarochawikasit
Co-advisor	Dr. Kittipong Piyawanno
Program	Bachelor of Engineering
Field of Study	Computer Engineering
Department	Computer Engineering
Faculty	Engineering
Academic Year	2023

Abstract

King Mongkut's University of Technology Thonburi (KMUTT) requires undergraduate students to accumulate no less than 100 hours of extracurricular activities to graduate. Participation in these activities is crucial for degree completion. The university organizes at least 150 events each academic year. However, being a large institution with over 10,000 students, KMUTT faces challenges in effectively disseminating information about these activities. As a result, students often miss opportunities to participate in events aligning with their interests, and some activities receive less engagement than deserve. Currently, KMUTT has the Modlink application for sharing university announcements and updates. Aside from tracking academic progress and announcements, Modlink's ability to recommend activities to students is relatively ineffective. Examples include suggesting expired events, prioritizing staff-led activities over student-centric ones, providing limited event details, or recommending activities inaccessible to students from certain departments without considering their majors. Through a survey of event organizers within the university, it was found that despite their efforts to plan beneficial activities for students, the lack of a centralized information hub hinders the dissemination of event details to interested students. To address these issues, the project team aims to develop an AI-powered application to facilitate student participation in extracurricular activities and serve as a centralized platform for event information. The application will leverage artificial intelligence to recommend activities tailored to each student's interests, considering factors such as academic department, previously attended events, and preferences of students in similar majors. The goal is to enable students to easily access and engage with university-organized activities, streamlining the process of accumulating the required activity hours. The application is expected to enhance the efficiency of event communication within the university, ultimately aligning activities more closely with students' interests and needs.

Keywords: Native Mobile Application, Machine Learning, Recommendation System

หัวข้อปริญญาในพนธ์	Actiwiz
หน่วยกิต	3
ผู้เขียน	นาย คุณานนต์ ทรัพย์มามูล
	นาย ณภัทรัก เหมือนเตย
	นาย อนดล ทองฤทธิ์
อาจารย์ที่ปรึกษา	ผศ. ราชวิชช์ สโตร์วิกสิติ ที่ปรึกษา วิทยานินพนธ์ ดร. กิตติพงษ์ ปิยะวรรัตน์โนน ที่ปรึกษา วิทยานินพนธ์ร่วม
หลักสูตร	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมคอมพิวเตอร์
คณะ	วิศวกรรมศาสตร์
ปีการศึกษา	2566

บทคัดย่อ

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรีนั้นกำหนดหลักสูตรให้ผู้ที่จะสำเร็จการศึกษาชั้นปริญญาตรีได้นั้นต้องมีช่วงโมงกิจกรรมสะสมไม่น้อยกว่า 100 ชั่วโมง ทำให้การเข้าร่วมกิจกรรมต่าง ๆ นั้นมีความสำคัญกับการสำเร็จการศึกษา ซึ่งทางมหาวิทยาลัยนั้นได้มีการจัดกิจกรรมขึ้นมาในแต่ละปีการศึกษาไม่น้อยกว่า 150 รายการ อย่างไรก็ตามเนื่องจากมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรีเป็นสถาบันการศึกษาขนาดใหญ่ซึ่งมีจำนวนนักศึกษาไม่น้อยกว่า 10000 คน ทำให้การกระจายข่าวสารของกิจกรรมต่าง ๆ ที่ทางมหาวิทยาลัยจัดขึ้นไม่มีประสิทธิภาพ ซึ่งส่งผลให้นักศึกษาเองก็มีพัสดุโอกาสที่จะเข้าร่วมกิจกรรมตามความสนใจของตนเอง และบางกิจกรรมไม่ได้รับการตอบรับเท่าที่ควร ในปัจจุบันนี้ทางมหาวิทยาลัยแอปพลิเคชันสำหรับการกระจายข่าวสารต่าง ๆ ของทางมหาวิทยาลัยแล้ว Modlink ให้นักศึกษาใช้งาน อย่างไรก็ตามนอกจากการใช้งานในการติดตามผลการเรียนและประกาศต่าง ๆ ของทางมหาวิทยาลัยแล้ว Modlink นั้นค่อนข้างที่จะมีความสามารถในการแนะนำกิจกรรมให้แก่นักศึกษาที่ค่อนข้างจะไม่มีประสิทธิภาพ ยกตัวอย่างเช่น การที่เลือกที่จะแนะนำกิจกรรมที่หมดเวลาเข้าร่วมไปแล้ว การที่ตัวแอปเลือกแนะนำกิจกรรมโดยเน้นไปที่บุคลากรของมหาวิทยาลัยมากกว่าที่จะเป็นกิจกรรมที่นักศึกษาสามารถเข้าร่วมและได้ชื่โน้มกิจกรรม การที่ระบุรายละเอียดของกิจกรรมเอาไว้อย่างชัดเจน หรือการที่แนะนำกิจกรรมที่นักศึกษาจากต่างภาควิชาไม่สามารถเข้าร่วมได้โดยไม่คำนึงว่าผู้ใช้งานศึกษาในภาควิชานี้ใด อีกทั้งจำในการสำรวจผู้จัดงานกิจกรรมต่าง ๆ ในมหาวิทยาลัยซึ่งแม้จะพยายามจัดกิจกรรมให้อีกประยุกต์แก่นักศึกษา แต่ด้วยความไม่เป็นศูนย์กลางในการกระจายข่าวสารก็ทำให้กิจกรรมที่ถูกจัดขึ้นไม่สามารถแพร่กระจายไปไม่นักศึกษาที่สนใจได้เท่าที่ควร จากปัญหาเหล่านี้ทำให้คณผู้จัดทำจึงมีแนวคิดที่จะพัฒนาแอปพลิเคชันที่จะอำนวยความสะดวกในการเข้าร่วมกิจกรรมต่าง ๆ โดยมีการนำเทคโนโลยีปัญญาประดิษฐ์มาใช้ในการแนะนำกิจกรรมต่าง ๆ ของนักศึกษาและเป็นศูนย์กลางในการกระจายข่าวสารของกิจกรรมต่าง ๆ โดยมีการนำเทคโนโลยีปัญญาประดิษฐ์มาใช้ในการแนะนำกิจกรรมต่าง ๆ ให้แก่นักศึกษาตามความสนใจของนักศึกษาคนนั้น ๆ โดยพิจารณาจากปัจจัยต่าง ๆ เช่น ภาควิชาที่เข้าศึกษา กิจกรรมที่นักศึกษาเคยเข้าร่วม หรือ กิจกรรมที่นักศึกษาในภาควิชาเดียวกันเคยเข้าร่วมเป็นต้น ซึ่งจะทำให้นักศึกษาสามารถเข้าถึงกิจกรรมต่าง ๆ ที่ทางมหาวิทยาลัยเป็นผู้จัดได้่ายิ่งขึ้น และอำนวยความสะดวกในการทำกิจกรรมต่าง ๆ กับทางมหาวิทยาลัย เพื่อแลกเปลี่ยนความรู้และการทำกิจกรรม ผู้จัดทำนั้นคาดหวังที่จะพัฒนาแอปพลิเคชันที่จะอำนวยความสะดวกในการทำกิจกรรมของนักศึกษา และช่วยกระจายข่าวสารของกิจกรรมต่าง ๆ ที่ถูกจัดขึ้นภายในมหาวิทยาลัยให้มีประสิทธิภาพยิ่งขึ้น โดยมุ่งไปที่การแนะนำกิจกรรมที่ตรงกับความสนใจและความต้องการของนักศึกษาให้มากที่สุด

คำสำคัญ: Native Mobile Application, Machine Learning, Recommendation System

กิตติกรรมประกาศ

โครงการนวัตกรรมคอมพิวเตอร์ฉบับนี้ สำเร็จลุล่วงได้อย่างสมบูรณ์ด้วยความกรุณาอย่างยิ่งจาก อาจารย์รำชวิชช์ สิริชวิกสิต และ อาจารย์กิตติพงษ์ ปิยะวรรณ์โนน ที่ได้สละเวลาอันมีค่าแก่คณะกรรมการผู้จัดทำ เพื่อให้คำปรึกษาและแนะนำตลอดจนตรวจทานแก้ไขข้อบกพร่องต่างๆ ด้วยความเอาใจใส่เป็นอย่างยิ่ง จนโครงการนวัตกรรมคอมพิวเตอร์ฉบับนี้สำเร็จสมบูรณ์ลุล่วงได้ด้วยดี คณะกรรมการผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี่จากใจจริง
สุดท้ายนี้ ขออุทิศความดีที่มีในการศึกษา นี้แด่บิดา มารดา ครอบครัวของคณะกรรมการผู้จัดทำและกำลังใจจากมิตรแท้ทุกท่าน

สารบัญ

หน้า

ABSTRACT	ii
บทคัดย่อ	iii
กิตติกรรมประกาศ	iv
สารบัญ	v
สารบัญตาราง	ix
สารบัญรูปภาพ	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 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 ทำไงด้วยเป็น 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

2.2 Principle Design	15
2.2.1 Principle Design คืออะไร	15
2.2.2 C.R.A.P. Principle Design	16
2.3 F-Shape Pattern	17
2.4 เทคโนโลยี	18
2.4.1 Integrated Development Environment (IDE)	18
2.4.2 Design	18
2.4.3 Frontend	18
2.4.4 Backend	18
2.4.5 Database	18
2.4.6 Machine Learning	19
2.4.7 Version Control	19
2.4.8 Testing	19
2.5 แบบสำรวจผลิตภัณฑ์	20
บทที่ 3 วิธีการทำงาน กระบวนการและการออกแบบ	26
3.1 บทนำ	26
3.1.1 สำรวจความต้องการของผู้ใช้เชิงคุณภาพ	26
3.1.2 เสน่ห์ทางของผู้ใช้ (Journey Map)	26
3.1.3 ผู้ได้รับผลกระทบ (Stakeholder)	26
3.2 ความต้องการของผู้ใช้ (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	32
3.6.1 การเข้าสู่ระบบ	32
3.6.2 การสมัครเข้าใช้งาน	33
3.6.3 การออกแบบระบบ	34
3.6.4 การค้นหากิจกรรม	35
3.6.5 การอ่านรายละเอียดของกิจกรรมและเข้าร่วมกิจกรรม	36
3.6.6 การประเมินกิจกรรม	37
3.6.7 การแจ้งเตือนกิจกรรมชั้นรุ่ม	38
3.6.8 การค้นหาชั้นรุ่ม	39
3.6.9 อ่านรายละเอียดของชั้นรุ่มและการเข้าร่วมชั้นรุ่ม	40
3.6.10 การลาออกจากชั้นรุ่ม	40
3.6.11 การแนะนำกิจกรรมให้ผู้ใช้	42
3.6.12 การ Train Model การเรียนรู้ของเครื่อง	43
3.7 แบบจำลองข้อมูลแบบกราฟ (Graph Data Model)	44
3.7.1 โหนด	45
3.7.1.1 โหนด User: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดส่วนตัวของผู้ใช้ภายในมหาวิทยาลัย	45
3.7.1.2 โหนด Department: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดของภาควิชาภายในมหาวิทยาลัย	45
3.7.1.3 โหนด Club: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดของชมรมภายในมหาวิทยาลัย	45
3.7.1.4 โหนด Club Category: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดประเภทชมรมภายในมหาวิทยาลัย	46
3.7.1.5 โหนด Activity: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดของกิจกรรมภายในมหาวิทยาลัย	46
3.7.1.6 โหนด Activity Category: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับประเภทของกิจกรรมภายในมหาวิทยาลัย	47
3.7.2 ความสัมพันธ์	47
3.7.2.1 ความสัมพันธ์ที่มีคุณสมบัติ	47
3.7.2.2 ความสัมพันธ์ที่ไม่มีคุณสมบัติ	48

3.8 การแบ่งประเภทของกิจกรรมและชั้นรุ่ม	49
3.9 หลักการพิจารณาสิ่งที่จะแนะนำให้ผู้ใช้	50
3.9.1 Content-Based Filtering	50
3.9.2 Collaborative Filtering	50
3.10 ส่วนติดต่อผู้ใช้ (User Interface)	51
3.10.1 หน้าลงชื่อเข้าใช้ระบบ	51
3.10.2 หน้าแนะนำกิจกรรมและชั้นรุ่ม	52
3.10.3 หน้าแสดงการແຈ້ງເຕືອນ	53
3.10.4 หน้าแสดงຂໍ້ມູນຜູ້ໃຊ້	54
3.10.5 หน้ารายละเอียดໝາຍ	55
3.10.6 หน้าลงทะเบียนຜູ້ໃຊ້	56
3.10.7 หน้ารายละเอียດກິຈกรรม	57
3.10.8 หน้าປະເມີນກິຈกรรม	58
บทที่ 4 ผลการทดลองและอภิปรายผล	59
4.1 การเปลี่ยนแปลงรายละเอียดของโครงการ	59
4.1.1 รายการคุณลักษณะ (Feature List)	59
4.1.2 แผนภาพที่ใช้แสดงปฏิสัมพันธ์ระหว่างระบบงานและสิ่งที่อยู่นอกระบบงาน (Use Case Diagram)	60
4.1.3 แผนภาพที่ใช้แสดงการทำงานของระบบ (Sequence Diagram)	62
4.1.3.1 การเข้าสู่ระบบ	62
4.1.3.2 การแจ้งเตือนการປະເມີນກິຈกรรม	64
4.1.4 ส่วนติดต่อผู้ใช้ (User Interface)	64
4.2 การพัฒนาของโครงการ	65
4.2.1 ระบบแนะนำกิจกรรมและชั้นรุ่ม	65
4.2.1.1 การแบ่งประเภทของกิจกรรมและชั้นรุ่ม	65
4.2.1.2 หลักการพิจารณาสิ่งที่จะแนะนำให้ผู้ใช้งาน	65
4.2.2 การพัฒนาแอปพลิเคชัน	66
4.2.2.1 ลงชื่อเพื่อเข้าใช้งานแอปพลิเคชัน	66
4.2.2.2 กรณีที่เคยลงชื่อเข้าใช้ในแอปพลิเคชัน	68
4.2.2.3 ลงทะเบียนຜູ້ໃຊ້ກັບแอปพลิเคชัน	69
4.2.2.4 แนะนำกิจกรรมตามความสนใจของຜູ້ໃຊ້โดยอ้างอิงจาก tag ของກິຈกรรม	70
4.2.2.5 แนะนำໝາຍ ตามความสนใจของຜູ້ໃຊ້โดยอ้างอิงจาก tag ของກິຈกรรม	71
4.2.2.6 ค้นหาກິຈกรรมທີ່ສົນໃຈ	72
4.2.2.7 ค้นຫາໝາຍທີ່ສົນໃຈ	73
4.2.2.8 เลือกອ່ານรายละเอียดກິຈกรรมທີ່ສົນໃຈ	75
4.2.2.9 เลือกອ່ານรายละเอียດໝາຍທີ່ສົນໃຈ	76
4.2.2.10 ลงชื่อເຂົ້າຮ່ວມກິຈกรรมທີ່ສົນໃຈ	77
4.2.2.11 ลงชื่อເຂົ້າຮ່ວມໝາຍທີ່ສົນໃຈ	79
4.2.2.12 ถอนชื่อจากการປັບສາມາຊີກໝາຍ	80
4.2.2.13 ແຈ້ງເຕືອນກິຈกรรมທີ່ເກີ່ວຂ້ອງກັບໝາຍຫຼືຄວາມສົນໃຈຂອງນັກສຶກຫາ	80
4.2.2.14 ແຈ້ງເຕືອນກາປະເມີນກິຈกรรม	81
4.2.2.15 ອອກຈາກຮບບ	81
4.3 การทดสอบการทำงาน (Functional Testing)	83
4.4 การทดสอบคุณภาพของระบบ (Non Functional Testing)	84
บทที่ 5 สรุปผลการดำเนินงาน	87

5.1 กระบวนการทำงาน (Work Process)	87
5.1.1 การดำเนินงานในภาคการศึกษาที่ 1	87
5.1.1.1 การเตรียมการเริ่มต้นโครงการ	87
5.1.1.2 ศึกษาวิธีการพัฒนาแอปพลิเคชัน	87
5.1.1.3 ออกแบบแอปพลิเคชัน	87
5.1.1.4 จัดเตรียมข้อมูล	87
5.1.2 การดำเนินงานในภาคการศึกษาที่ 2	88
5.1.2.1 การพัฒนาแอปพลิเคชัน	88
5.1.2.2 การทดสอบแอปพลิเคชัน	88
5.2 ปัญหาที่พบในโครงการและการแก้ไข (Problems and Solutions)	88
5.2.1 ความไม่คุ้นชินกับเทคโนโลยีที่ใช้ในการพัฒนา	88
5.2.2 จำนวนข้อมูลที่สามารถใช้ได้มีน้อย	88
5.2.3 ความลำบากในการขอข้อมูลจากมหาวิทยาลัย	88
5.2.4 หน่วยประมวลผลที่ใช้ในการพัฒนา Machine Learning มีจำนวนน้อย	88
5.3 แนวทางการพัฒนาในอนาคต (Future Work)	89
หนังสืออ้างอิง	90
APPENDIX	92
A หลักฐานการพัฒนาโครงการ	93
B Functional Test	98

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	30
3.5 การอ่านรายละเอียดกิจกรรม	31
3.6 การเข้าร่วมกิจกรรม	31
3.7 การประเมินกิจกรรม	31
3.8 การค้นหาชุมชน	31
3.9 การอ่านรายละเอียดชุมชน	31
3.10 การสมัครชุมชน	32
3.11 การลาออกจากชุมชน	32
3.12 ตารางเก็บข้อมูลของผู้ใช้ที่ใช้งานแอปพลิเคชัน	45
3.13 ตารางเก็บข้อมูลของภาควิชาภายในมหาวิทยาลัย	45
3.14 ตารางเก็บข้อมูลของชุมชนภายในมหาวิทยาลัย	45
3.15 ตารางเก็บข้อมูลประเภทชุมชน	46
3.16 ตารางเก็บข้อมูลของชุมชนภายในมหาวิทยาลัย	46
3.17 ตารางเก็บข้อมูลประเภทของกิจกรรม	47
3.18 ตารางเก็บข้อมูลนักศึกษาภาควิชา	47
3.19 ตารางเก็บข้อมูลสมาชิกของชุมชน	47
3.20 ตารางเก็บข้อมูลสมาชิกเข้าร่วมกิจกรรม	48
3.21 ตารางเก็บข้อมูลกิจกรรมถูกจัดโดยชุมชน	48
4.1 การสมัครเข้าใช้งานแบบใหม่	60
4.2 การเข้าสู่ระบบแบบใหม่	61
4.3 การประเมินกิจกรรม	61
4.4	83
4.5 ตารางความพึงพอใจในการแก้ปัญหา	84
4.6 ตารางความสะดวกในการใช้งานแอปพลิเคชัน	85
4.7 ตารางความพึงพอใจโดยรวมของแอปพลิเคชัน	86
5.1 ตารางรายการการเตรียมการเริ่มต้นโครงการ	87
5.2 ตารางรายการการศึกษาวิธีการพัฒนาแอปพลิเคชัน	87
5.3 ตารางรายการการออกแบบแอปพลิเคชัน	87
5.4 ตารางรายการการจัดเตรียมข้อมูล	87
5.5 ตารางรายการการพัฒนาแอปพลิเคชัน	88
5.6 ตารางรายการการทดสอบแอปพลิเคชัน	88

สารบัญ

รูปที่	หน้า
1.1 ภาคการศึกษาที่ 1	2
1.2 ภาคการศึกษาที่ 2	2
2.1 Mobile Application 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 การเข้าสู่ระบบ	32
3.4 การสมัครเข้าใช้งาน	33
3.5 การออกจากระบบ	34
3.6 การค้นหากิจกรรม	35
3.7 การอ่านรายละเอียดของกิจกรรมและเข้าร่วมกิจกรรม	36
3.8 การประเมินกิจกรรม	37
3.9 การแจ้งเตือนกิจกรรมชุมชน	38
3.10 การค้นหาชุมชน	39
3.11 อ่านรายละเอียดของชุมชนและการเข้าร่วมชุมชน	40
3.12 การลากออกจากรายชื่อ	41
3.13 การแนะนำกิจกรรมให้ผู้ใช้	42
3.14 การ Train Model การเรียนรู้ของเครื่อง	43
3.15 Graph Data Model	44
3.16 ตัวอย่าง Cluster	49
3.17 หน้าลงชื่อเข้าใช้ระบบ	51
3.18 หน้าแนะนำกิจกรรมและชุมชน	52
3.19 หน้าแสดงการแจ้งเตือน	53
3.20 หน้าแสดงข้อมูลผู้ใช้	54
3.21 หน้ารายละเอียดชุมชน	55
3.22 หน้าลงทะเบียนผู้ใช้	56

3.23	หน้ารายละเอียดกิจกรรม	57
3.24	หน้าประเมินกิจกรรม	58
4.1	Use Case Diagram แบบใหม่	60
4.2	แผนภาพการเข้าสู่ระบบแบบใหม่	62
4.3	แผนภาพการแจ้งเตือนการประเมินกิจกรรมแบบใหม่	64
4.4	หน้าแรกของแอปพลิเคชัน	66
4.5	หน้าระบบลงชื่อเข้าใช้งาน Microsoft	67
4.6	หน้าแนะนำกิจกรรม	68
4.7	หน้าลงทะเบียนผู้ใช้	69
4.8	เลือกห้องมุ่ลคุรบถawan	70
4.9	ขอข้อมูลกิจกรรมแนะนำเพิ่ม	71
4.10	ขอข้อมูลชุมชนแนะนำเพิ่ม	72
4.11	หน้าค้นหากิจกรรม	73
4.12	หน้าแสดงชุมชนที่แนะนำ	74
4.13	หน้าค้นหาชุมชน	75
4.14	ข้อมูลกิจกรรม	76
4.15	รายละเอียดของชุมชน	77
4.16	ปุ่มเข้าร่วมกิจกรรม	78
4.17	รูปเมื่อเข้าร่วมกิจกรรมแล้ว	79
4.18	ปุ่มลาออกจากชุมชน	80
4.19	การแจ้งเตือนกิจกรรมที่เกี่ยวข้องกับชุมชนหรือความสนใจของนักศึกษา	81
4.20	หน้าข้อมูลผู้ใช้	82
4.21	การยืนยันการออกจากระบบ	82
4.22	แผนภาพคะแนนความพึงพอใจในการแก้ปัญหาของแอปพลิเคชัน	84
4.23	แผนภาพคะแนนความสะดวกในการใช้งานแอปพลิเคชัน	85
4.24	แผนภาพคะแนนความพึงพอใจรวมของแอปพลิเคชัน	86
A.1	หน้าเข้าสู่ระบบของแอปพลิเคชัน Actiwiz	93
A.2	ฐานข้อมูลของชุมชนในมหาวิทยาลัย	94
A.3	ระบบแจ้งเตือนของแอปพลิเคชัน Actiwiz	94
A.4	ความเชื่อมโยงของผู้ใช้กับ tag ต่างๆ ของกิจกรรม	95
A.5	การวิเคราะห์ความสนใจจากเนื้อหาที่ผู้ใช้เข้าไปอ่าน	95
A.6	ตัวอย่างของกิจกรรมที่ถูกแบ่งโดย tag ต่างๆ	96
B.1	กรณีผู้ใช้ไม่กรอกอีเมล	98
B.2	กรณีผู้ใช้ไม่กรอกอีเมลไม่ถูกต้อง	99
B.3	กรณีผู้ใช้ไม่กรอกรหัสผ่าน	100
B.4	กรณีผู้ใช้กรอกรหัสผ่านผิด	101
B.5	กรณีที่ไม่มีบัญชีผู้ใช้ในฐานข้อมูล	102
B.6	กรณีที่ผู้ใช้กรอกข้อมูล	103
B.7	กรณีที่ผู้ใช้ต้องการออกจากระบบ	104
B.8	กรณีผู้ใช้กดปุ่ม No หรือ กด nok ครอบแจ้งเตือน	105
B.9	กรณีที่ผู้ใช้ต้องการค้นหา กิจกรรม	106
B.10	กรณีผู้ใช้คลิกองค์กรเข้าไปดูรายละเอียดของกิจกรรม	107
B.11	กรณีผู้ใช้คลิกองค์กรเข้าไปดูรายละเอียดของชุมชน	108
B.12	กรณีผู้ใช้กดเข้าร่วมกิจกรรม	109
B.13	กรณีผู้ใช้กดเข้าร่วมชุมชน	110
B.14	กรณีผู้ใช้ยกเลิกการเข้าร่วมชุมชน	111
B.15	กรณีผู้ใช้ตรวจสอบการแจ้งเตือนของตัวแอปพลิเคชัน	112
B.16	กรณีผู้ใช้ตรวจสอบกิจกรรมที่ทางระบบแนะนำมาให้	113

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

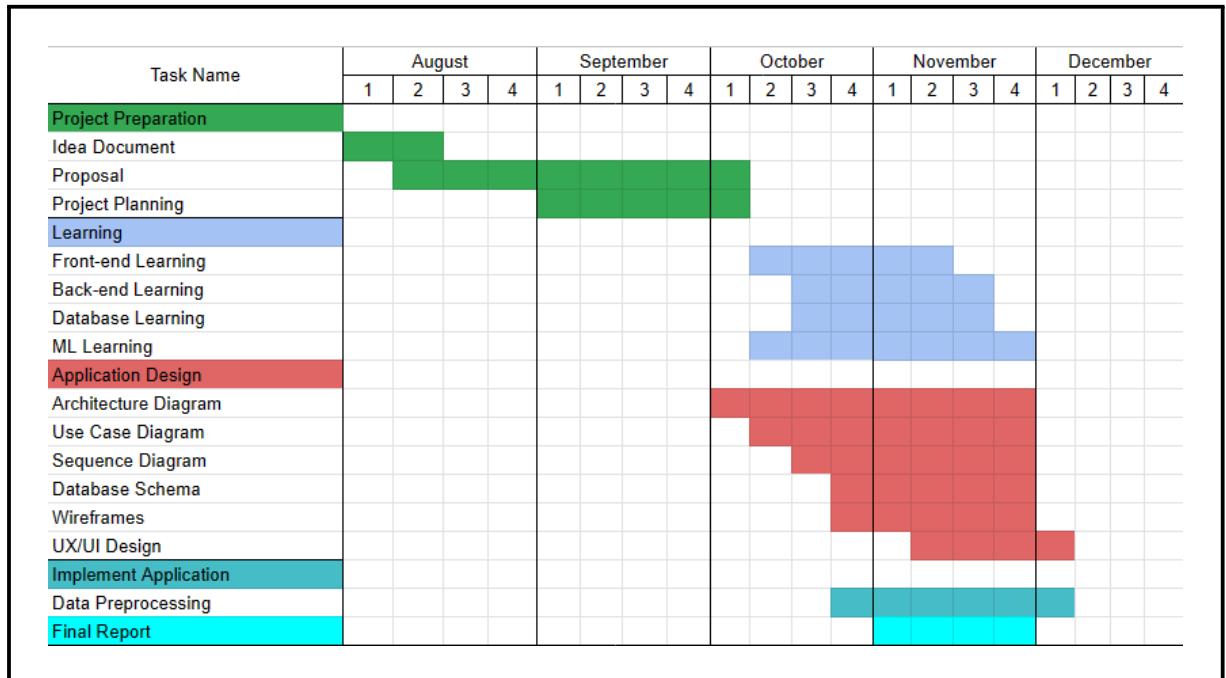
1.3 ประเภทของโครงการ

โครงการที่เป็นการประดิษฐ์ คิดค้น

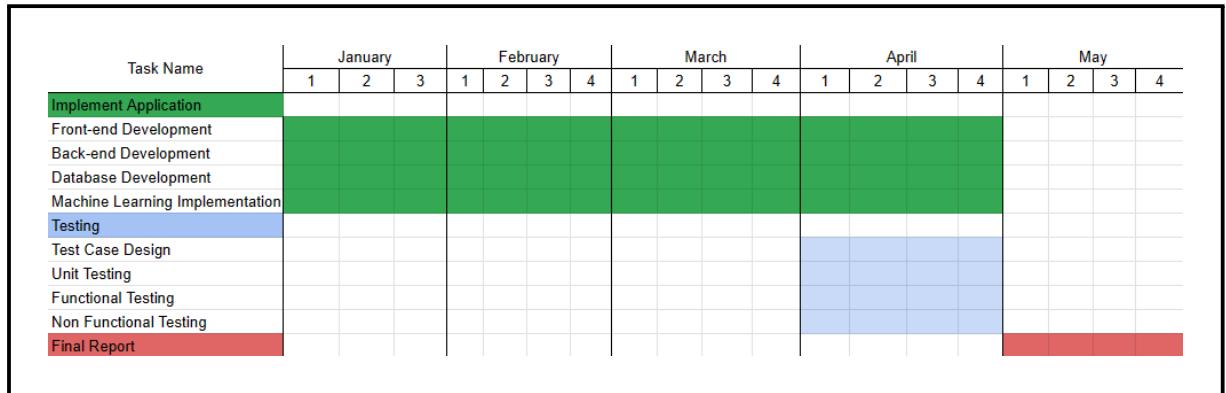
1.4 วัตถุประสงค์

- เพื่อพัฒนาแอปพลิเคชันสำหรับการกระจายข้อมูลข่าวสารและกิจกรรมต่าง ๆ ภายในมหาวิทยาลัย เพื่อให้ง่ายต่อการติดตามข่าวไว้แก่นักศึกษาภายในมหาวิทยาลัย
- เพื่อศึกษาพัฒนาและประเมินผลกิจกรรมของนักศึกษา เพื่อที่จะแนะนำกิจกรรมและชุมชนที่นักศึกษามีแนวโน้มให้ความสนใจ
- แอปพลิเคชันที่อำนวยความสะดวกต่อนักศึกษาในการเข้าร่วม กิจกรรมต่าง ๆ
- 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 อกมาเป็น 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 เนื่องจากความสามารถในการให้คำแนะนำที่มีความเป็นบุคคลและเกี่ยวข้องกับความสนใจของผู้ใช้ กระบวนการนี้เน้นการวิเคราะห์เนื้อหาที่ผู้ใช้มีความสนใจและแนะนำสิ่งที่มีเนื้อหาที่คล้ายคลึงกันให้กับผู้ใช้ ตัวอย่างเช่น การแนะนำหนังสือที่มีเนื้อหาใกล้เคียงกับที่ผู้ใช้เคยซื้อมาอ่อนหนานี้ ดังนั้นเทคนิคนี้จึงเป็นการสร้างคำแนะนำที่เน้นความเป็นบุคคลและความสอดคล้องกับความต้องการของผู้ใช้อย่างมีความแม่นยำและเหมาะสม อย่างไรก็ตามการทำ Content-Based Filtering นั้น จะยึดตามความสนใจของผู้ใช้ที่มีประวัติเก็บเอาไว้ทำให้สามารถแนะนำได้ในวงแคบ ๆ เท่านั้น

2.1.1.2 Collaborative Filtering

Collaborative Filtering [14], [15] เป็นหนึ่งในแนวทางที่ได้รับความนิยมอย่างมากในการสร้างคำแนะนำสำหรับผู้ใช้ แนวทางนี้เสนอหลักการแนะนำที่มาจากพฤติกรรมการใช้งานที่เกิดขึ้นก่อนหน้านี้ของผู้ใช้ การ Collaborative Filtering มีวิธีการหลักที่สำคัญสองวิธีคือ User-Based Collaborative Filtering: แนวทางนี้จะทำการแนะนำสิ่งที่ผู้ใช้มีความสนใจอยู่ หากมีความคล้ายคลึงในพฤติกรรมการใช้งานระหว่างผู้ใช้สองคน เช่น ถ้าผู้ใช้สองคนมีรูปแบบพฤติกรรมที่คล้ายกัน ระบบจะแนะนำสิ่งที่ผู้ใช้อีกคนเคยใช้งานให้กับผู้ใช้คนอื่น ซึ่งเพิ่มโอกาสที่จะให้คำแนะนำที่หลากหลายขึ้นได้ อย่างไรก็ตามก็อาจมีโอกาสที่จะแนะนำสิ่งที่ผู้ใช้ไม่สนใจเลยได้เช่นกัน และยังมีปัญหาความซ้ำซ้อนเกิดขึ้นได้หากมีผู้ใช้หรือสิ่งที่จะแนะนำเป็นจำนวนมากมาก

Item-Based Collaborative Filtering: แนวทางนี้จะทำการแนะนำสิ่งที่ผู้ใช้มีความสนใจ แล้วจะทำการแนะนำสิ่งที่มีเนื้อหาคล้ายกัน เมื่อพิจารณาการกระทำที่เกิดขึ้นก่อนหน้านี้ ระบบจะแนะนำสิ่งที่มีความคล้ายคลึงให้กับผู้ใช้ อย่างไรก็ได้เนื่องจากเป็นการแนะนำจากสิ่งที่มีความใกล้เคียงกัน คำแนะนำที่ได้จึงอาจไม่ใช่สิ่งที่ผู้ใช้มองหา

2.1.1.3 การประมวลผลภาษาธรรมชาติ

การประมวลผลภาษาธรรมชาติ (Natural Language Processing หรือ NLP) [16], [17] เป็นสาขานึงของปัญญาประดิษฐ์ ที่จะมุ่งเน้นการให้ความสามารถแก่คอมพิวเตอร์ที่เป็นการเข้าใจ ตีความ และปฏิสัมพันธ์กับภาษาของมนุษย์ แนวทางนี้นำเอาระบบคอมพิวเตอร์และเทคโนโลยีในการประมวลผล วิเคราะห์ และสร้างข้อมูลข้อความและเสียง เพื่อให้เครื่องคอมพิวเตอร์สามารถสกัดความหมาย บริบท และความรู้จากภาษาที่มนุษย์สร้างขึ้นได้

2.1.1.4 Word Embedding

Word Embedding [18], [19] ในการทำการประมวลผลภาษาธรรมชาติคือการที่แปลงคำต่าง ๆ ออกมายเป็นเวกเตอร์เพื่อที่จะตรวจสอบความหมายของคำและความหมายในการสร้างประโยค ซึ่งเป็นพื้นฐานในการทำงานเดลการประมวลผลภาษาธรรมชาติซึ่งสามารถนำมาประยุกต์ได้ทั้งการที่จะจัดประเภทกิจกรรมหรือการวิเคราะห์ความต้องการของผู้ใช้จากข้อความที่ถูกใช้

2.1.1.5 Transformer Model

Transformer Model [20] คือสถาปัตยกรรมการเรียนรู้ของเครื่องสำหรับการประมวลผลและทำความเข้าใจภาษาของมนุษย์ ซึ่งทำงานได้ดีในการแปลภาษา การสรุป และการสร้างข้อความ โดยอาศัยหลักการของ self-attention ในการทำความเข้าความสัมพันธ์ระหว่างคำในประโยคพร้อมกัน ทำให้สามารถเข้าใจบริบทและความหมายได้มากขึ้นอย่างมีประสิทธิภาพ โดย Transformer-base model ที่ถูกใช้งานในโครงการนี้ 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 รวมไปถึงไม่สามารถปรับแต่งได้บนประสิทธิภาพเท่า T5

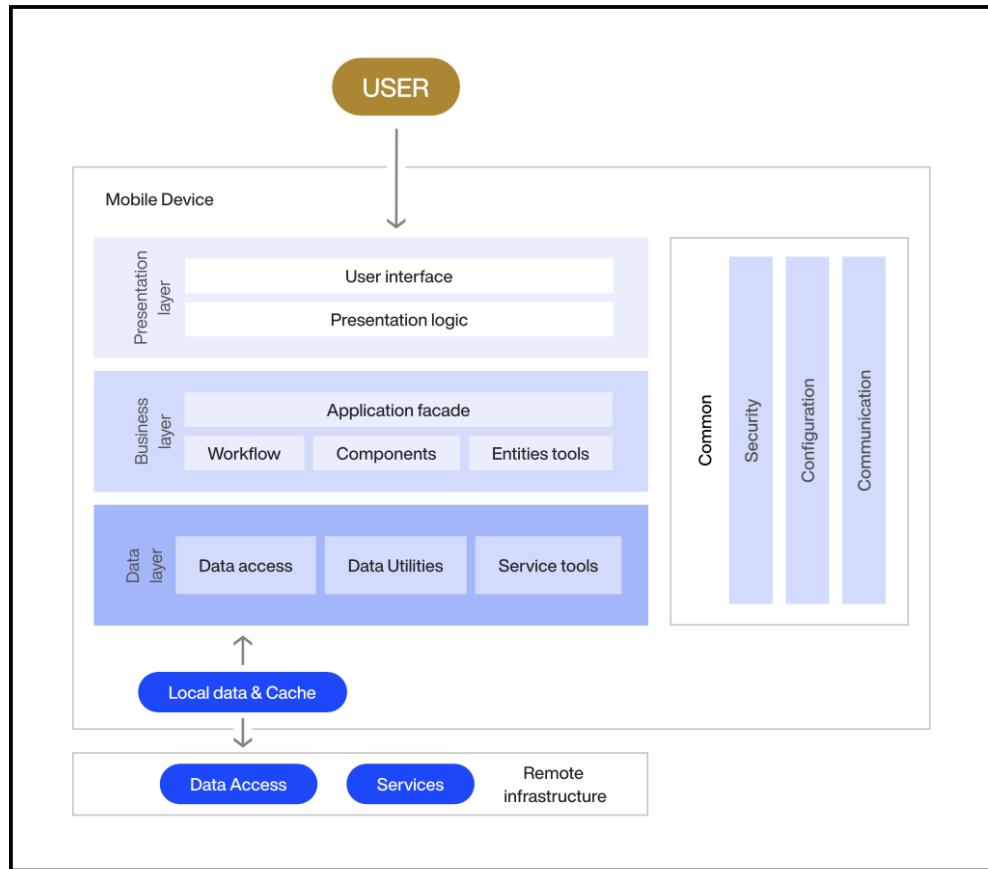
2.1.1.6 โครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม (Artificial neural network) เป็นสถาปัตยกรรมการเรียนรู้ของเครื่องที่ได้รับแรงบันดาลใจจากระบบประสาทของมนุษย์ โดยมีหลักการคือการแบ่งการทำงานเป็นหนندต่าง ๆ ที่รับค่าที่ประมวลผลมาจากหนึ่งใน input layer และส่งต่อให้หนندประมวลผลที่อยู่ใน hidden layer ไปเรื่อยๆ จนถึงหนندประมวลผลสุดท้ายใน output layer เพื่อตีความผลลัพธ์ ซึ่งสามารถทำงานได้หลากหลายรวมไปถึงงานการประมวลผลภาษาธรรมชาติซึ่งในโครงการนี้มี 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 ซึ่งสามารถจำหรือลบข้อมูลได้ตามความเหมาะสม 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

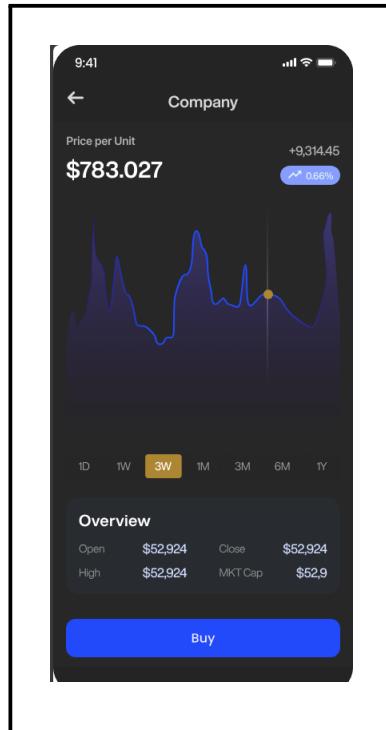


รูปที่ 2.1: Mobile Application Architecture [5]

สถาปัตยกรรมของแอปพลิเคชันส่วนใหญ่จะประกอบด้วยสามเลเยอร์หลัก ๆ ได้แก่ Presentation layer, Business layer และ Data layer

1. Presentation layer[1]

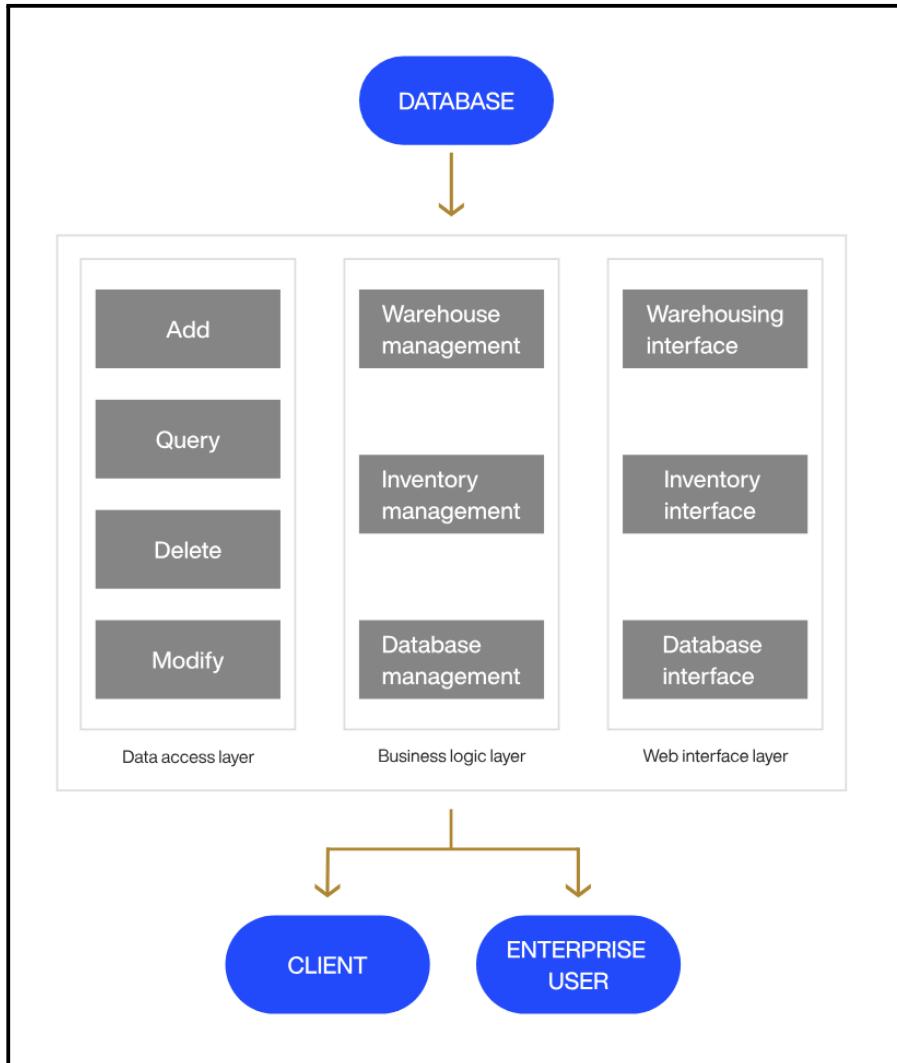
หรือก็คือ front end เป็นส่วนของแอปพลิเคชันที่ผู้ใช้มองเห็นและมีปฏิสัมพันธ์ได้ โดยมีส่วนติดต่อผู้ใช้ (user interface หรือ UI) ของแอปพลิเคชันเป็นส่วนสำคัญของเลเยอร์นี้ วัตถุประสงค์หลักของเลเยอร์นี้คือการนำข้อมูลที่ส่งมาจาก business layer มาแสดงผลในลักษณะที่ผู้ใช้สามารถเข้าใจได้ ไม่ว่าจะเป็น UI แบบพื้นฐาน 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 ย่อยๆหรือส่วนประกอบหลายส่วน โดยแต่ละส่วนมีหน้าที่รับผิดชอบในการทำงานเฉพาะ ตัวอย่างเช่น หากคุณมีแอปการจัดการทรัพยากรองค์กร (ERP) business layer อาจมีส่วนประกอบสำหรับการจัดการคลังสินค้าและระบบจัดการสินค้าคงคลัง



รูปที่ 2.3: Business layer [7]

3. Data layer

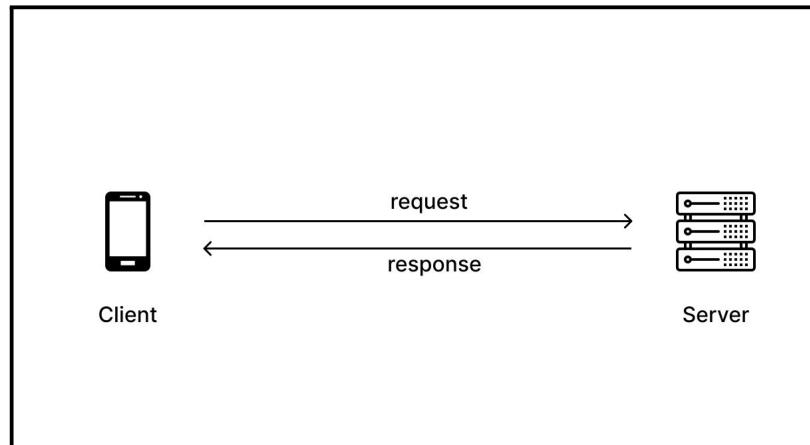
Data Layer [1] นี้เป็นตัวกลางระหว่าง Layer อื่นๆ กับทรัพยากรภายนอก วัตถุประสงค์หลักของ Layer นี้คือการรวบรวมข้อมูลจากแหล่งต่าง ๆ เช่น ฐานข้อมูล, เซิร์ฟเวอร์คลาวด์ หรือ API และส่งไปยัง Layer ด้านบน ตัวอย่างเช่น เมื่อผู้ใช้ขอให้แอปพลิเคชันแสดงไฟล์ของตน data layer จะเข้ามายื่นกับ Database และขอข้อมูลที่เกี่ยวข้องทั้งหมด เช่น ชื่อ วันเกิด ไฟล์รูปภาพ และอื่นๆ อย่างไรก็ตามใน layer นี้ข้อมูลส่วนใหญ่ยังไม่ผ่านการประมวลผล จึงอาจจะมีข้อมูลบางอย่าง เช่น แท็กหรืออีดีที่ผู้ใช้ไม่ควรเห็น ในส่วนนี้จึงต้องทำการส่งข้อมูลให้ business layer ประมวลผลเพิ่มเติมเป็นลำดับต่อไป

2.1.3 HTTP Protocol

2.1.3.1 HTTP Protocol คืออะไร

HTTP [23] (Hypertext Transfer Protocol) เป็นโปรดักโคลที่ใช้ในการแลกเปลี่ยนข้อมูลผ่านทางอินเทอร์เน็ต 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 ต่าง ๆ 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 และผู้ใช้สามารถอ่านได้อย่างเข้าใจว่าเป็นการเรียงลำดับแบบไหน

- 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 โดยสามารถใส่ query เพื่อระบุหน้า 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 หลังจากที่ผู้ใช้ API (Client) ส่ง request ไปหา server ผ่าน API แล้วผู้ใช้ Client จะต้องทำการยืนยันให้ได้ว่า API ใช้งานได้จริงหรือไม่ หรือส่งไปสำเร็จไหม จึงมีการต้องส่ง response ที่มี HTTP Status code ระบุกลับไปยัง 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 (เข้าสู่ระบบผ่าน) แต่จะมีข้อมูลบางหน้า ที่ไม่มีสิทธิ์ในการเข้าถึง
- 404 Not Found: เป็น response ที่บ่งบอกว่า request นั้นไม่ว่าจะใช้งานตอนนี้ หรือ request ที่เรียกนั้นไม่มีอยู่ในระบบ

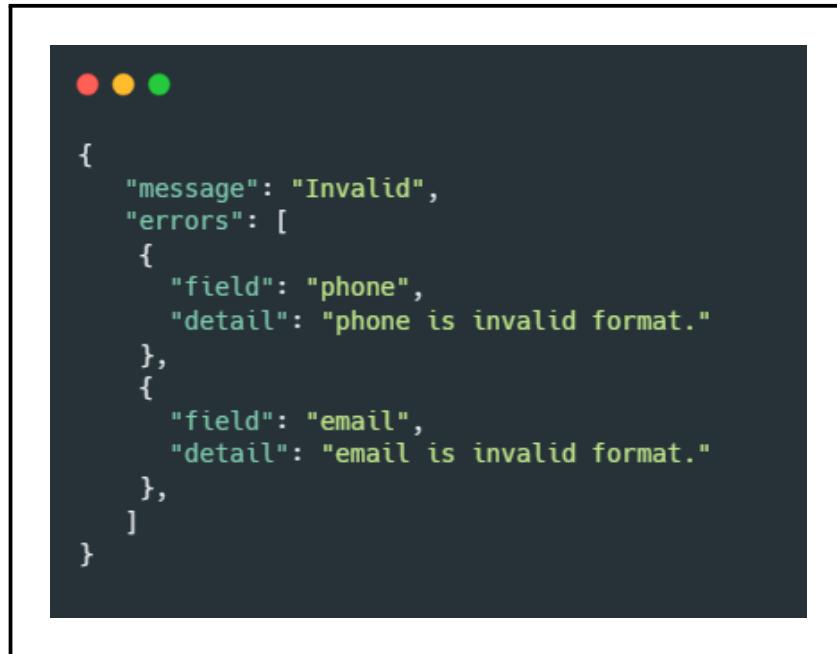
- 405 Gone: เป็น response ที่บ่งบอกว่า resource ที่ต้องการนั้นไม่มี หรือถูกย้ายไป
- 429 Too many Request: เป็น response ที่บ่งบอกว่า request นั้นติด limit ใช้กรณีที่กำหนด rate limit ไว้กับ API นั้น ๆ จะสามารถเรียกได้ครั้ง
- 500 Internal Server Error: เป็น response ที่บ่งบอกว่าการ request นั้นถูกต้องแล้ว 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 ให้ตรงตามวัตถุประสงค์เพื่อให้ผู้ใช้ที่ได้รับ response กลับไปเข้าใจ response นั้นได้มากขึ้น

7. การ Handle Error ให้ user เข้าใจ nok เนื่องจาก 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

ให้นำเป็นเหมือน entity ของกราฟสามารถที่จะเก็บ attribute จำนวนมากได้ สำหรับกราฟเราจะจึงเรียก attribute ว่าคุณสมบัติ ความสัมพันธ์หมายถึงความสัมพันธ์ที่เชื่อมระหว่าง 2 โหนด และเหมือนกับโหนดซึ่งสามารถเก็บคุณสมบัติได้



```

{
  "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

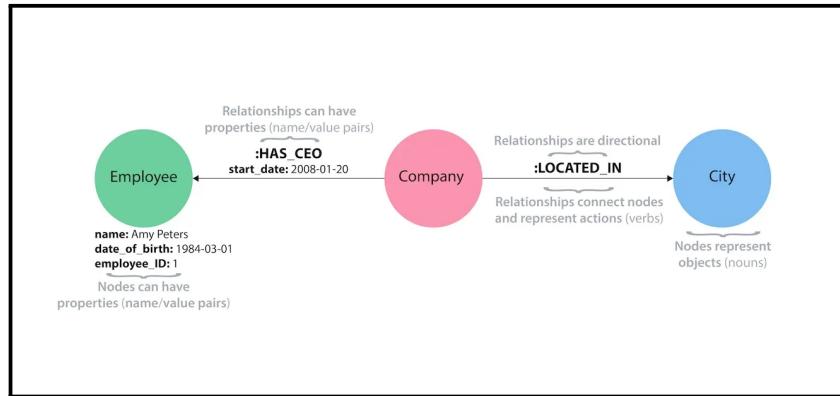
Graph database [26] หรือฐานข้อมูลแบบกราฟ จัดเป็น NoSQL Database รูปแบบหนึ่ง ซึ่งนำมาใช้แก้ปัญหา database ที่มีข้อมูลขนาดใหญ่และไม่มีรูปแบบชัดเจน ฐานข้อมูลแบบ Graph ออกแบบมาเพื่อแสดงความสัมพันธ์ (Relationship) ระหว่างข้อมูลที่มีความเชื่อมโยงกับข้อมูลที่เราสนใจได้อย่างชัดเจน รวมถึงมีความสามารถในการเก็บข้อมูลที่ไม่ต้องกำหนดรูปแบบล่วงหน้า

2.1.5.2 ทำไมต้องเป็น Graph Database

การเข้าถึงโนดและ relationships ในฐานข้อมูล Graph เป็นวิธีที่มีประสิทธิภาพและใช้เวลาในการทำงานคงที่ และช่วยให้เราสำรวจการเชื่อมต่อหลายล้านต่อวินาทีต่อเรคคอร์ดได้อย่างรวดเร็ว มีความเป็นอิสระจากขนาดรวมของข้อมูลทั้งหมดของเรา ทำให้ฐานข้อมูลแบบ Graph มีความสามารถในการจัดการข้อมูลที่มีรูปแบบซับซ้อนและมีความเชื่อมต่อ กันสูงได้มีประสิทธิภาพ

2.1.5.3 Property ของ Graph Model [3]

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



รูปที่ 2.8: Property ของ Graph Model

โหนดเป็นเหมือน entity ของ Graph สามารถที่จะเก็บ attribute จำนวนมากได้ สำหรับ Graph เราจะเรียก attribute ว่า properties Relationships เป็นความสัมพันธ์ที่เชื่อมระหว่าง 2 โหนดและเหมือนกับโหนดมันสามารถเก็บ 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	ฐานข้อมูลแบบเชิงสัมพันธ์เชิงๆ กับการสืบค้นที่ซับซ้อนเมื่อสำรวจความสัมพันธ์ที่อาจทำให้ประสิทธิภาพการทำงานช้าลง	ฐานข้อมูลแบบกราฟเชี่ยวชาญในการแสดงและสืบค้นความสัมพันธ์ระหว่างข้อมูล
ความสามารถในการใช้งาน	ฐานข้อมูลแบบเชิงสัมพันธ์ทำงานได้ดีกับข้อมูลขนาดใหญ่และข้อมูลที่มีโครงสร้างพวกลมไม่ค่อยเชี่ยวชาญเมื่อเป็นเรื่องการสืบค้นแบบหลายช่วง	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 เพื่อสื่อสารให้นักพัฒนาและผู้ที่เกี่ยวข้องทุกคนเข้าใจตรงกัน ซึ่งสามารถนำมาใช้อธิบายความสัมพันธ์ของสิ่งต่าง ๆ ที่อยู่ในชั้นงานได้ โดยแผนภาพ UML มีแผนภาพหลายรูปแบบสำหรับใช้อธิบายโครงสร้างและความสัมพันธ์ในรูปแบบต่าง ๆ ที่นักพัฒนาต้องทำความเข้าใจเพื่อใช้ในการปฏิบัติงาน

2.1.6.2 Use case Diagram

Use Case Diagram เป็นหนึ่งในแผนภาพ UML ที่ใช้ในการแสดงภาพรวมของวิธีการใช้ระบบหรือซอฟต์แวร์ จากมุมมองของผู้ใช้หรือแต่ละกลุ่มผู้ใช้ โดยทำให้่ายต่อการเข้าใจและสื่อสารความต้องการของระบบกับผู้ใช้และทีมพัฒนา 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 หนึ่งสามารถขยาย (extend) ไปยัง Use Case อื่น ๆ ในกรณีที่มีเหตุการณ์เฉพาะที่เกิดขึ้น

Use Case Diagram มีประโยชน์มากในการทำความเข้าใจและกำหนดความต้องการของระบบจากมุมมองของผู้ใช้ และช่วยให้ทีมพัฒนามีภาพรวมของฟังก์ชันและการทำงานของระบบที่ชัดเจน

2.1.6.3 Sequence Diagram

Sequence Diagram เป็นหนึ่งในแผนภาพ UML ที่ใช้งานเพื่อแสดงลำดับขั้นตอนหรือการทำงานของวัตถุต่าง ๆ ภายในระบบหรือโปรแกรม ในแต่ละขั้นตอนของการทำงานนั้น ๆ 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 Principle Design

2.2.1 Principle Design คืออะไร

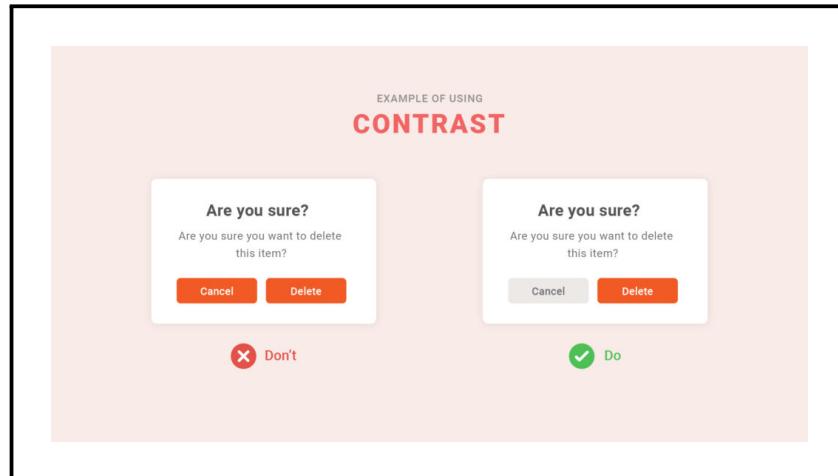
Principle Design [28] คือหลักการในการออกแบบที่ช่วยให้เข้าใจธรรมชาติในการรับรู้ข้อมูลและจำแนกประเภทของสมองมนุษย์ ยกตัวอย่างง่าย ๆ เช่นการที่เราสามารถสื่อสารกันได้ทางภาษาไทย ภาษาอังกฤษ ฯ ฯ กันได้โดยไม่ต้องรู้ภาษาเดียวกัน

2.2.2 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)

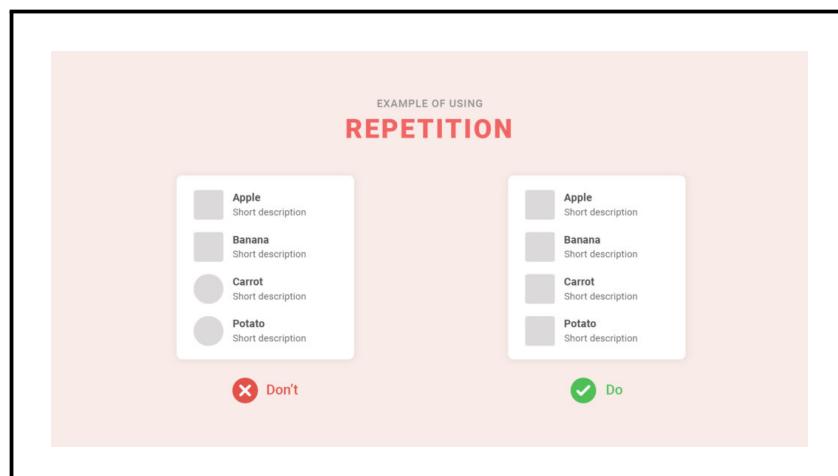
คือการแบ่งแยกข้อมูลด้วยความแตกต่าง โดยความแตกต่างในที่นี้ไม่ใช่แค่สีเพียงอย่างเดียว แต่รวมถึง ขนาดที่แตกต่างกัน รูปทรงที่แตกต่างกัน เป็นต้น เมื่อใช้ความแตกต่างในการแบ่งแยกข้อมูลแล้ว จะส่งผลต่อความสำคัญในการอ่าน ทำให้เราเลือกได้ว่าจะให้ผู้ใช้เห็นสิ่งไหนเป็นอย่างแรก และเห็นสิ่งไหนเป็นอย่างถัดมา ซึ่งสามารถดูวิธีแสดงความแตกต่างตามรูปที่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



รูปที่ 2.9: contrast [9]

2. การทำซ้ำ (Repetition)

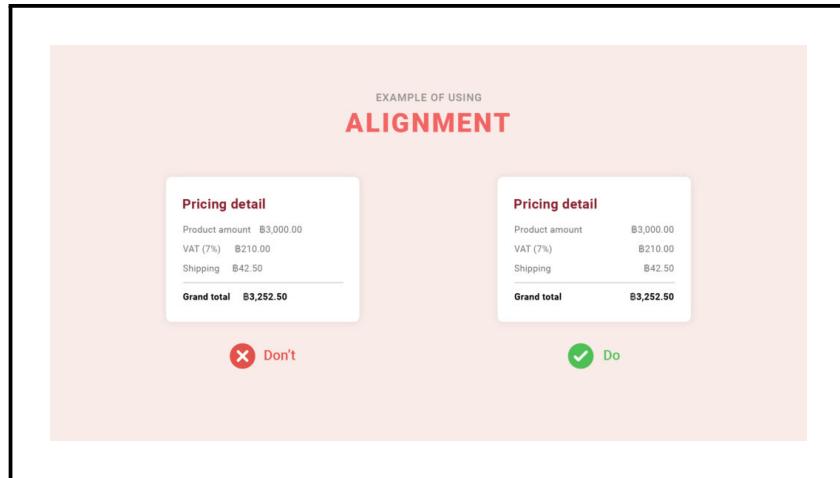
การทำซ้ำในที่นี้คือการทำซ้ำของเนื้อหา ทำให้ผู้ใช้มีต้องเรียนรู้ใหม่เมื่อเจออรูปแบบการจัดวางที่คล้าย ๆ กัน ก็จะเข้าใจได้ว่ามันมีการใช้งานที่เหมือนกัน หรือจะนำ Repetition มาใช้ในการแบ่งกลุ่ม เช่นของที่มีหน้าตาเหมือนกัน เมื่อนำมาเรียงไว้ด้วยกัน ผู้ใช้จะเข้าใจได้ว่ามันคือหมวดหมู่เดียวกัน ซึ่งสามารถดูตัวอย่างการทำซ้ำได้ดังรูปที่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



รูปที่ 2.10: repetition [10]

3. ตำแหน่งการจัดวาง (Alignment)

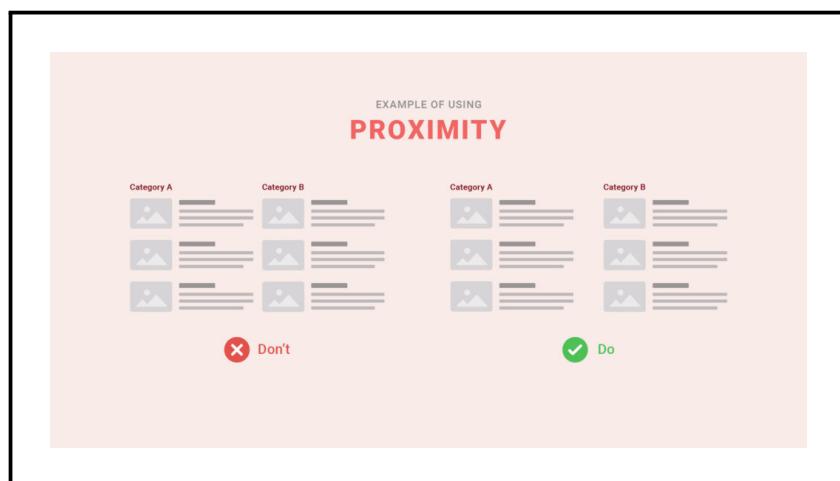
ส่งผลต่อความเป็นระเบียบ ความสวยงาม และความเชื่อมโยงกันของเนื้อหาโดย Alignment จะเป็น สิ่งแรก ๆ ที่ผู้ใช้รู้สึกได้ เมื่อ มีการเปลี่ยนแปลง เมื่อรักษาเนื้อหาในระบบเดียวกัน ผู้ใช้จะเข้าใจได้ว่าเนื้อหานี้มีความเชื่อมโยงกัน ซึ่งสามารถดูตัวอย่างการจัดวางได้ตามรูปที่ 2.11



รูปที่ 2.11: alignment [11]

4. การจัดกลุ่มข้อมูล (Proximity)

คือการจัดวางองค์ประกอบของข้อมูล โดยข้อมูลที่มีความเชื่อมโยงกันควรจัดให้เป็นกลุ่มก้อนเดียวกัน จะช่วยให้ข้อมูลในหน้าเว็บไซต์ หรือแอปพลิเคชันที่เราออกแบบมีความซับซ้อนน้อยลง และมีการแบ่งหมวดหมู่ชัดเจนขึ้น ซึ่งจะเห็นได้ดังตัวอย่างในรูปที่ 2.12



รูปที่ 2.12: proximity [12]

2.3 F-Shape Pattern

F-Shape Pattern [29] หรือ การออกแบบในรูปแบบตัวอักษร F คือการออกแบบที่มักจะพับได้บ่อที่สุดเมื่ออ่านบล็อกของเนื้อหา โดย การที่ผู้ใช้จะอ่านเนื้อหาโดยที่จะเริ่มจากด้านบนซ้ายไปขวา และจะลดลงไปเรื่อย ๆ จนจบที่ด้านล่างขวา โดยจะมีการสร้างเส้นที่เรียงตัวเป็น F ซึ่งจะช่วยให้ผู้ใช้สามารถอ่านเนื้อหาได้เร็วขึ้น

2.4 เทคโนโลยี

2.4.1 Integrated Development Environment (IDE)

- Visual Studio Code [30]

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

- Pycharm [31]

โปรแกรมสำหรับเขียนโค้ดสำหรับภาษาไฟทอน ติดตั้งบนเครื่องคอมพิวเตอร์ ได้ทั้งบนระบบปฏิบัติการ Windows, MacOS และ Linux

- Google Colab [32]

โปรแกรมสำหรับเขียนโค้ดสำหรับภาษาไฟทอนในเบราว์เซอร์ โดยไม่ต้องกำหนดค่าใดและสามารถเข้าถึง GPU โดยไม่มีค่าใช้จ่าย

2.4.2 Design

- Figma [33]

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

- LucidChart [34]

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

2.4.3 Frontend

- React Native [35]

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

2.4.4 Backend

- FastAPI [36]

FastAPI คือเว็บเฟรมเวิร์กที่มีความรวดเร็วและประสิทธิภาพสูง สำหรับการสร้าง API ด้วยภาษา Python ที่มีจุดเด่นได้แก่

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

2.4.5 Database

- Neo4j [37]

เป็นระบบฐานข้อมูลที่ถูกออกแบบมาเพื่อจัดเก็บข้อมูลแบบกราฟ (Graph Database) ที่สามารถจัดเก็บแบบและสอบถามความสัมพันธ์ระหว่างข้อมูลได้อย่างมีประสิทธิภาพและมีความยืดหยุ่น ระบบฐานข้อมูลกราฟนี้ถูกออกแบบมาเพื่อเก็บข้อมูลในรูปแบบของโหนดและ เชื่อมต่อระหว่างโหนดด้วยเส้นเชื่อมที่เรียกว่า Relationships ทำให้สามารถแสดงความสัมพันธ์และการเชื่อมโยงของข้อมูลได้อย่างชัดเจน

2.4.6 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 และเน้นการสร้างเวกเตอร์คำและการจัดกลุ่มคำศัพท์ 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 การจัดกลุ่มคำศัพท์ และวิธีการประมวลผลข้อมูล ที่ถูกพัฒนาให้ใช้งานได้อย่างมีประสิทธิภาพ

2.4.7 Version Control

- Git [38]

Version Control ตัวหนึ่งซึ่งจะเป็นระบบที่มีหน้าที่ทำการจัดเก็บการเปลี่ยนแปลงของไฟล์ใน Project และมีการ backup ให้สามารถที่จะเรียกดูหรือทำการย้อนกลับไปดูเวอร์ชันต่าง ๆ ของ Project ที่ได้เวลาใดก็ได้ ดังนั้น Version Control ก็หมายความว่าสามารถดูได้ว่าใครเป็นคนที่เปลี่ยนหรือแก้ไข Project ในส่วนต่างๆ

- Github

เว็บไซต์เว็บที่ให้บริการในการฝากไฟล์ Git หรือพูดง่าย ๆ ก็คือ Git ที่อยู่บนเว็บไซต์นั่นเอง ซึ่งจะทำให้สามารถใช้ Git ร่วมกับคนอื่นได้โดยผ่านเว็บไซต์ซึ่งจะมักนิยมใช้กันมาก ในการเก็บ Project Open Source ต่าง ๆ

2.4.8 Testing

- Jest [39]

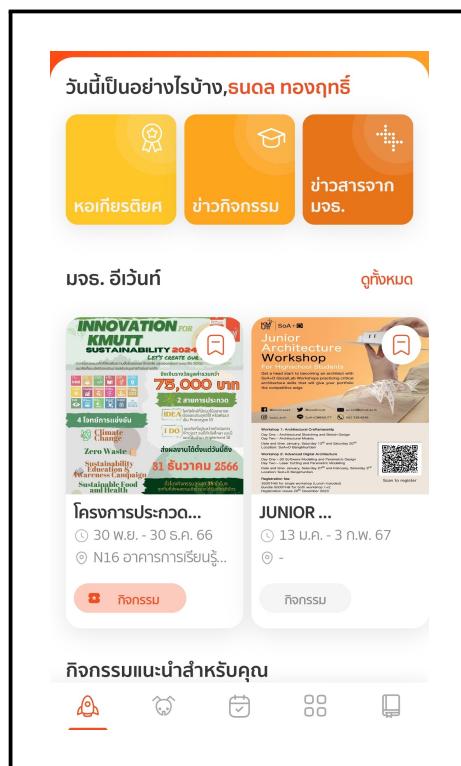
JavaScript Framework สำหรับเขียน Test เป็น Open Source ที่พัฒนาโดย Facebook ซึ่งมี helper มี function ต่าง ๆ ให้ใช้ทำให้ง่ายต่อการเขียน Test สามารถเขียนทดสอบได้ทั้ง React, Vue, Angular หรือ JavaScript ทั่ว ๆ ไป

- PyTest

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

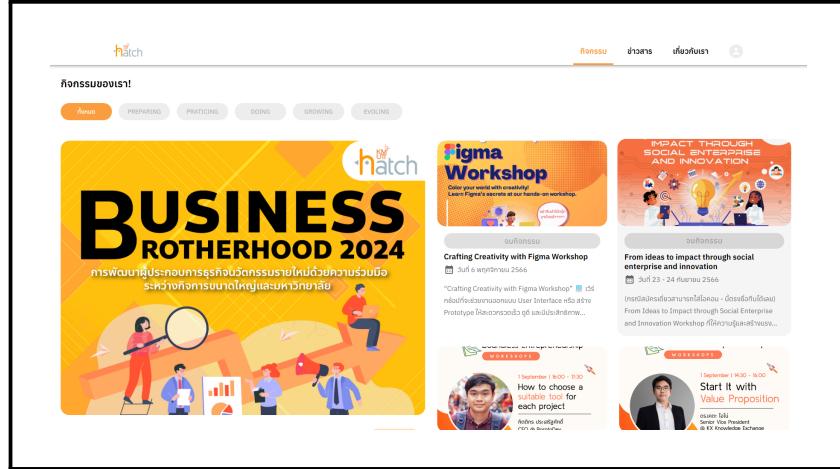
2.5 แบบสำรวจผลิตภัณฑ์

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 กิจกรรมต่าง ๆ และข้อมูลที่จำเป็นต่อผู้ใช้ อย่างไรก็ตามเว็บไซต์สามารถประชาสัมพันธ์ได้แค่กิจกรรมที่ทาง Hatch เป็นผู้จัดทำนั้น



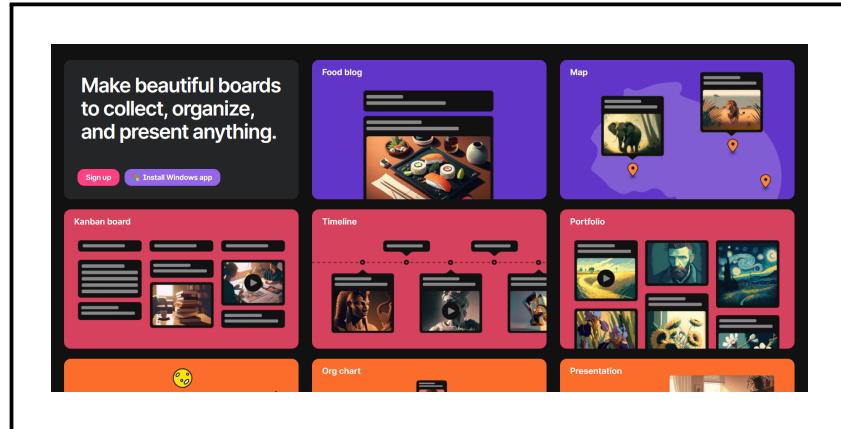
รูปที่ 2.14: หน้าแนะนำกิจกรรมของ KMUTT Hatch

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



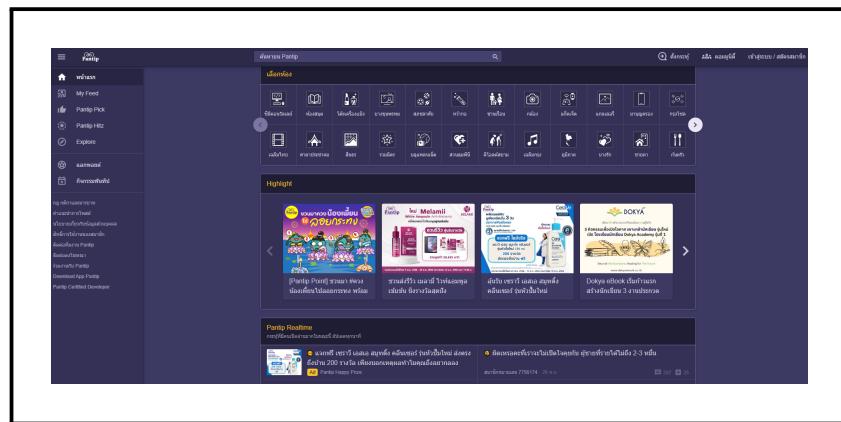
รูปที่ 2.15: หน้า Mainpage ของ KMUTT Sinfo

Padlet คือแอปพลิเคชันหรือเว็บไซต์ที่อยู่ในแพลตฟอร์มสำหรับ การระดมความคิด แสดงความคิดเห็น หรือแลกเปลี่ยนความรู้ร่วมกัน ผ่านกระดานดิจิทัลในรูปแบบเสมือน Post it ที่ติดบนบอร์ด ซึ่งจะแสดงผลทุกอย่างแบบ Real-time สามารถโพสต์ทั้งในรูปแบบข้อความ รูปภาพ และลิงก์ของเว็บไซต์ได้ เร็บเพลที่จะให้ผู้ใช้มาแสดงความเห็น หรือโพสต์ข้อมูลงบนเว็บ ซึ่งจะต่างจากตรงที่ทางระบบของจะเป็น ตัวกลางในการคำนวนหาจากความซื่นชอบของผู้ใช้จากการที่ผู้ใช้เคยได้เข้าร่วม หรือชั้นรมที่ผู้ใช้สนใจอยู่ หน้าที่เข้าไปยังบอร์ดต่าง ๆ ของ Padlet อยู่ในรูปที่ 2.16



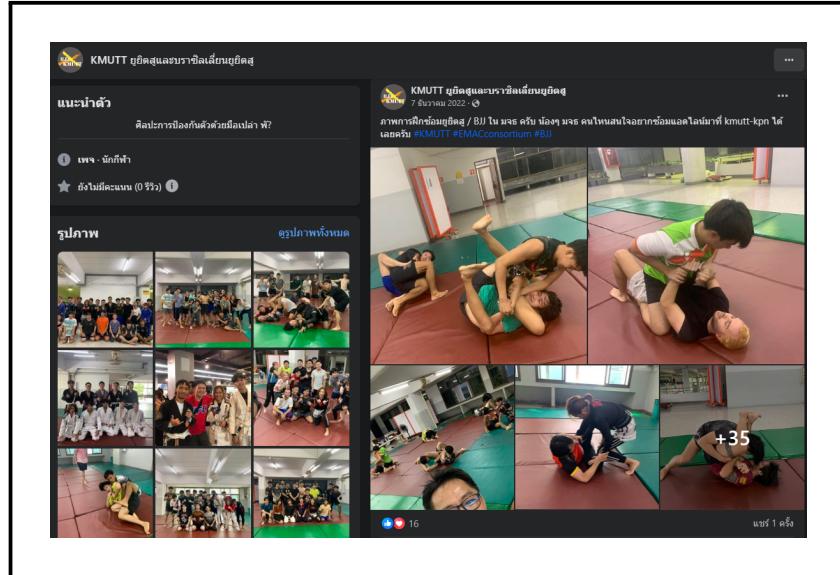
รูปที่ 2.16: หน้า Mainpage ของ Padlet

Pantip คือ พื้นที่สำหรับการแลกเปลี่ยนความคิดเห็นในหัวข้อ หรือประเด็นที่สนใจร่วมกัน สามารถสอบถาม บอกเล่าแบ่งปันประสบการณ์ ในเรื่องต่าง ๆ ในหน้ากระดานสนทนากโดยสมาชิกสามารถถั่ง หรือตอบกระทู้ต่าง ๆ ที่สมาชิกสนใจและสามารถเลือกหาอ่านข้อมูลได้จาก ป้ายหัวข้อในเรื่องต่าง ๆ ที่ทางเว็บไซต์ ได้สร้างขึ้นไว้ ซึ่งจะต่างจากตรงที่ทางระบบของ จะเป็นตัวกลางในการคำนวนหาจากความซื่นชอบ ของผู้ใช้จากการที่ผู้ใช้เคยได้เข้าร่วม หรือชั้นรมที่ผู้ใช้สนใจอยู่ โดยรูปที่ 2.17 คือหน้าที่แสดงเนื้อหาต่าง ๆ ของ Pantip



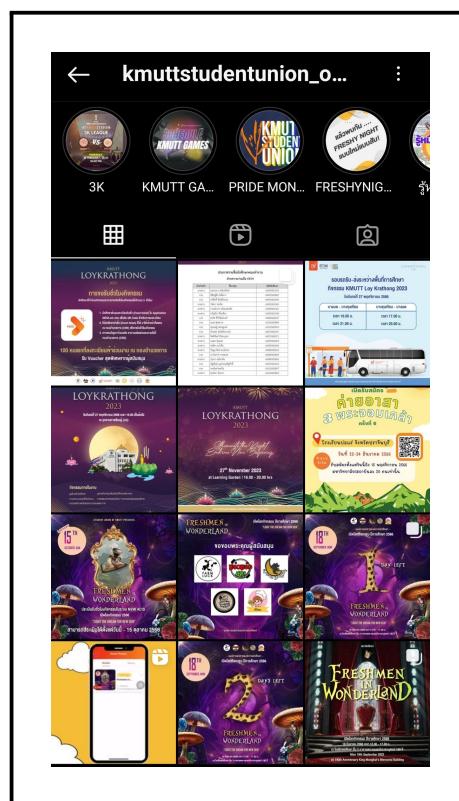
รูปที่ 2.17: หน้าแนะนำกระทู้ของ Pantip

Facebook เป็น social media ที่ได้รับความนิยมที่สามารถใช้งาน ได้ในหลาย platform ซึ่งสามารถนำเสนอข้อมูลข่าวสารให้ผู้ใช้ได้ มากมายและเป็น social media ที่มีคนใช้งานแบบตลอดทั้งวัน ทำให้บางชั้นรมเลือก ที่จะทำหน้าเพจเพื่อกระจายข่าวสารเกี่ยวกับชั้นรม ของตัวเอง อย่างไรก็ตามด้วยปริมาณข่าวสารมากมายของ facebook ทำให้ข่าวของชั้นรมมักโดดเด่นด้วย ข่าวอื่น ๆ อยู่เสมอ ถึงแม้จะ สามารถเข้าไปสู่หน้าเพจเพื่อดูความเคลื่อนไหวได้แต่ก็ไม่สามารถแนะนำตัวชั้นรมหรือกิจกรรมที่ชั้นรมจะจัดให้แก่นักศึกษาที่ไม่ติดตามเพจ ได้อยู่ดี โดยตัวอย่างเพจเพื่อประชาสัมพันธ์ใน 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故事 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

ในบทนี้จะกล่าวถึงการสำรวจความต้องการของผู้ใช้เพื่อนำมาวิเคราะห์คุณลักษณะที่โครงการควรมี และนำเสนอที่วิเคราะห์ได้มาใช้ออกแบบ โครงสร้างและองค์ประกอบต่าง ๆ ของตัวผลิตภัณฑ์ โดยจะมีการอธิบายโครงสร้างและองค์ประกอบต่าง ๆ ที่อยู่ในตัวผลิตภัณฑ์รวมถึงวิธีการทำงานโดยละเอียด

3.1 บทนำ

3.1.1 สำรวจความต้องการของผู้ใช้เชิงคุณภาพ

ในการทำระบบเพื่อแจ้งเตือนข่าวสารของกิจกรรมและชุมชน ทางผู้จัดทำระบุผู้ใช้และผู้ได้รับประโยชน์เป็น 3 กลุ่มใหญ่ๆ ด้วยกัน คือ ผู้เข้าร่วมกิจกรรม ผู้จัดกิจกรรม และประธานชุมชน โดยแต่ละกลุ่มมีความต้องการต่างนี้ ผู้เข้าร่วมกิจกรรมซึ่งเป็นผู้ใช้หลัก คือ นักศึกษาที่วิ่งในมหาวิทยาลัย โดยมีความต้องการพื้นฐานคือการที่อย่างที่ทำกิจกรรมตามความสนใจ แต่ด้วยภาระการเรียนทำให้ส่วนใหญ่ไม่มีโอกาสที่จะหา กิจกรรมหรือ ชุมชนที่ตนสนใจ สิ่งที่ต้องการจึงเป็นแหล่งที่รวบรวม ข้อมูลข่าวสาร กิจกรรม และ รายละเอียดของชุมชนต่างรวมถึงสิ่งที่ชุมชนนั้นทำเอาไว้ ที่เดียวเพื่อที่จะหา กิจกรรมและชุมชนตามที่ต้องการได้ ผู้ได้รับประโยชน์กลุ่มแรก คือ ผู้จัดกิจกรรมที่แทนผู้จัดกิจกรรมที่เป็นเจ้าหน้าที่ของมหาวิทยาลัย โดยมีความต้องการพื้นฐานคือการประชาสัมพันธ์กิจกรรมที่จัด แต่ข่าวสารของกิจกรรมที่ถูกเผยแพร่ผ่าน platform ต่าง ๆ เช่น facebook หรือ instargram นั้นมีผู้ที่เห็นการประชาสัมพันธ์เพียงบางส่วนเท่านั้น ทำให้ผู้เข้าร่วมกิจกรรมมีน้อยกว่าที่คาดหวัง ผู้ได้รับประโยชน์กลุ่มที่ 2 คือ ประธานชุมชน ที่แทนผู้ที่ดำเนินงานชุมชน ซึ่งส่วนใหญ่เป็นนักศึกษาที่ใกล้จะจบการศึกษา หรือนักศึกษาชั้นปริญญาโท โดยมีความต้องการพื้นฐาน คือ การประชาสัมพันธ์ชุมชนที่ต้องการอยู่ ถึงแม้จะเป็นงานชุมชนจะมี การนัดหมายกันแบบปากต่อปากอยู่แล้ว แต่การจัดการกิจกรรมชุมชน ก็ไม่เป็นระบบเท่าที่ควร อีกทั้งการหาสมาชิกชุมชนใหม่ หรือ การประชาสัมพันธ์กิจกรรมชุมชนให้คนภายนอกชุมชนอกรายการที่สามารถเป็นคนเชื่อใจได้มากที่สุด จึงจำเป็นต้องการในการตัดสินใจ เข้าร่วมกิจกรรม และ ชุมชนได้อย่างเหมาะสม ทำให้กิจกรรม และ ชุมชนไม่ได้รับความสนใจเท่าที่ควร

3.1.2 เส้นทางของผู้ใช้ (Journey Map)

จากกลุ่มผู้ใช้ทั้งสามกลุ่มสามารถแบ่งพฤติกรรมการใช้งานได้เป็น 2 ประเภท คือ การหา กิจกรรมหรือชุมชน

ผู้กระทำ: ผู้เข้าร่วมกิจกรรม

Step 1 : ผู้เข้าร่วมกิจกรรมไปตามรายละเอียดจากผู้จัดกิจกรรมหรือประธานชุมชน
ปัญหา - ผู้เข้าร่วมอาจไม่รู้ว่ามีกิจกรรมหรือชุมชนน้อย หรือไม่รู้จักผู้จัดกิจกรรม

Step 2 : ผู้เข้าร่วมลงทะเบียนกิจกรรมหรือชุมชน

Step 3 : ผู้เข้าร่วมเข้าร่วมกิจกรรมหรือทำกิจกรรมชุมชน

ปัญหา - ผู้เข้าร่วมอาจไม่รู้ตำแหน่งของสถานที่จัดกิจกรรมหรือชุมชน

Step 4 : ผู้เข้าร่วมประเมินกิจกรรม

ปัญหา - ผู้เข้าร่วมไม่รู้ว่าแบบประเมินกิจกรรมสามารถประเมินได้แล้ว

3.1.3 ผู้ได้รับผลกระทบ (Stakeholder)

ผู้ที่คาดว่าจะได้รับประโยชน์จากการใช้งานแอปพลิเคชันของผู้จัดทำสร้างขึ้นแบ่งเป็น 2 กลุ่ม คือ ผู้เข้าร่วมกิจกรรม และ ผู้จัดกิจกรรมที่หมายถึงผู้จัดกิจกรรมและประธานชุมชน ผู้เข้าร่วมกิจกรรม คือ นักศึกษาชั้นปีที่ 1 เป็นต้นไปเป็นกลุ่มที่ยังไม่ข้อมูลเกี่ยวกับสิ่งต่าง ๆ ในมหาวิทยาลัยอยู่น้อยที่หากาที่จะหาแหล่งข้อมูลของกิจกรรมหรือชุมชนที่สนใจ ผู้จัดกิจกรรม คือ นักศึกษาชั้นปีที่ 4 ที่เป็นประธานชุมชนและผู้จัดกิจกรรมที่เป็นเจ้าหน้าที่ของมหาวิทยาลัย เป็นต้นไปเป็นผู้ได้รับผลกระทบจากการประชาสัมพันธ์กิจกรรมและชุมชนที่ไม่ประสิทธิภาพเท่าที่ควร

3.2 ความต้องการของผู้ใช้ (Requirement List)

รายการข้อกำหนดหรือความต้องการที่จำเป็นต้องมีในโครงการหรือผลิตภัณฑ์ที่กำลังถูกพัฒนา ข้อกำหนดเหล่านี้เป็นข้อมูลที่ถูกรวบรวมมาจากผู้ใช้ ลูกค้า หรือผู้เกี่ยวข้องอื่นๆ ซึ่งมีไว้เพื่อกำหนดขอบเขตและคุณลักษณะของผลิตภัณฑ์หรือโครงการ โดยในส่วนนี้ หลังจากที่ได้ทำการสำรวจความต้องการจากผู้ใช้ได้รับผลประโยชน์แล้วทำให้เราได้ความต้องการมา ดังนี้

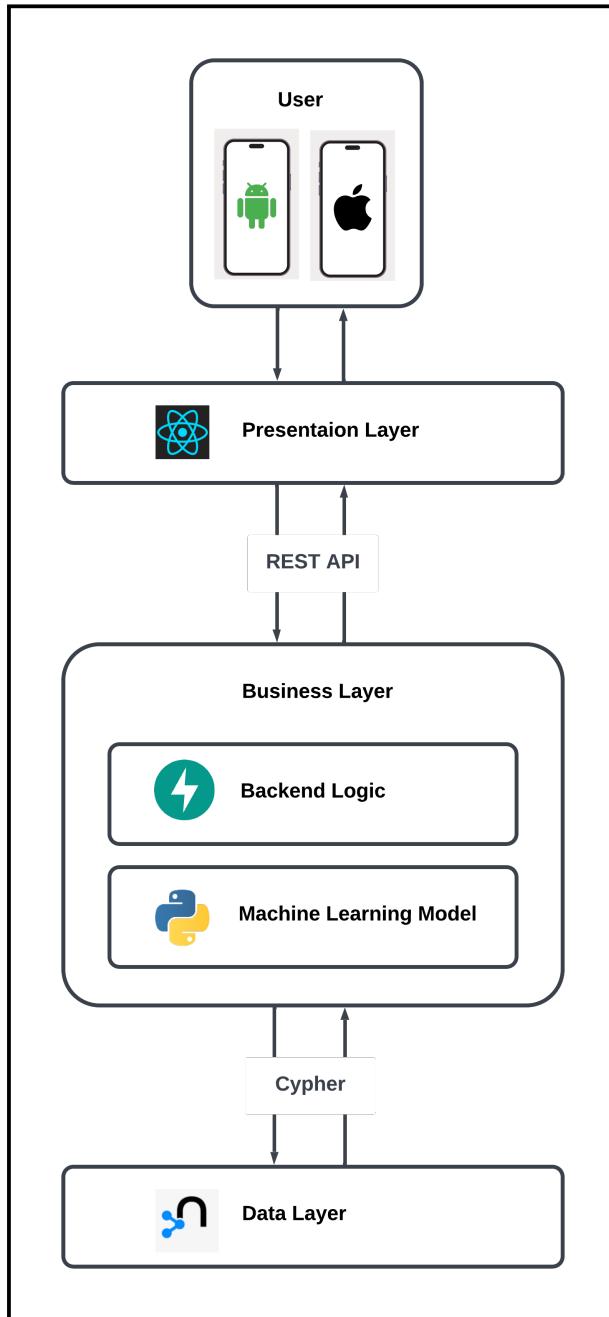
1. ข้อมูลของกิจกรรมและชุมชนที่ชัดเจน
2. ระบบการค้นหากิจกรรมและชุมชน
3. การแนะนำชุมชนและกิจกรรมที่น่าสนใจ
4. การแจ้งเตือนการประเมินกิจกรรม

3.3 รายการคุณลักษณะ (Feature List)

จากการวิเคราะห์ความต้องการทั้งหมด เราได้ทำการวิเคราะห์คุณลักษณะมาเพื่อตอบโจทย์ความต้องการของผู้ใช้ดังนี้

1. Login: ลงชื่อเพื่อเข้าใช้งานแอพพลิเคชัน
2. Registration: สมัครบัญชีของแอพพลิเคชันด้วยอีเมลหรือหมายเลขโทรศัพท์
3. Logout: ออกจากระบบ
4. Search: ค้นหากิจกรรมหรือชุมชนที่สนใจ
5. Select: เลือกอ่านรายละเอียดกิจกรรมหรือชุมชนที่สนใจ
6. Join:
 - ลงชื่อเข้าร่วมกิจกรรมที่สนใจ
 - ลงชื่อเป็นสมาชิกของชุมชนที่สนใจ
7. Resignation: ถอนชื่อจากการเป็นสมาชิกชุมชน
8. Notification:
 - แจ้งเตือนกิจกรรมที่เกี่ยวข้องกับชุมชนหรือความสนใจของนักศึกษา
 - แจ้งเตือนการประเมินกิจกรรม
9. Event evaluation: ประเมินกิจกรรม
10. Recommendation: แนะนำกิจกรรมและชุมชน ตามความสนใจของผู้ใช้โดยอ้างอิงจาก tag ของกิจกรรม

3.4 แผนภาพสถาปัตยกรรมของระบบ (System Architecture Diagram)



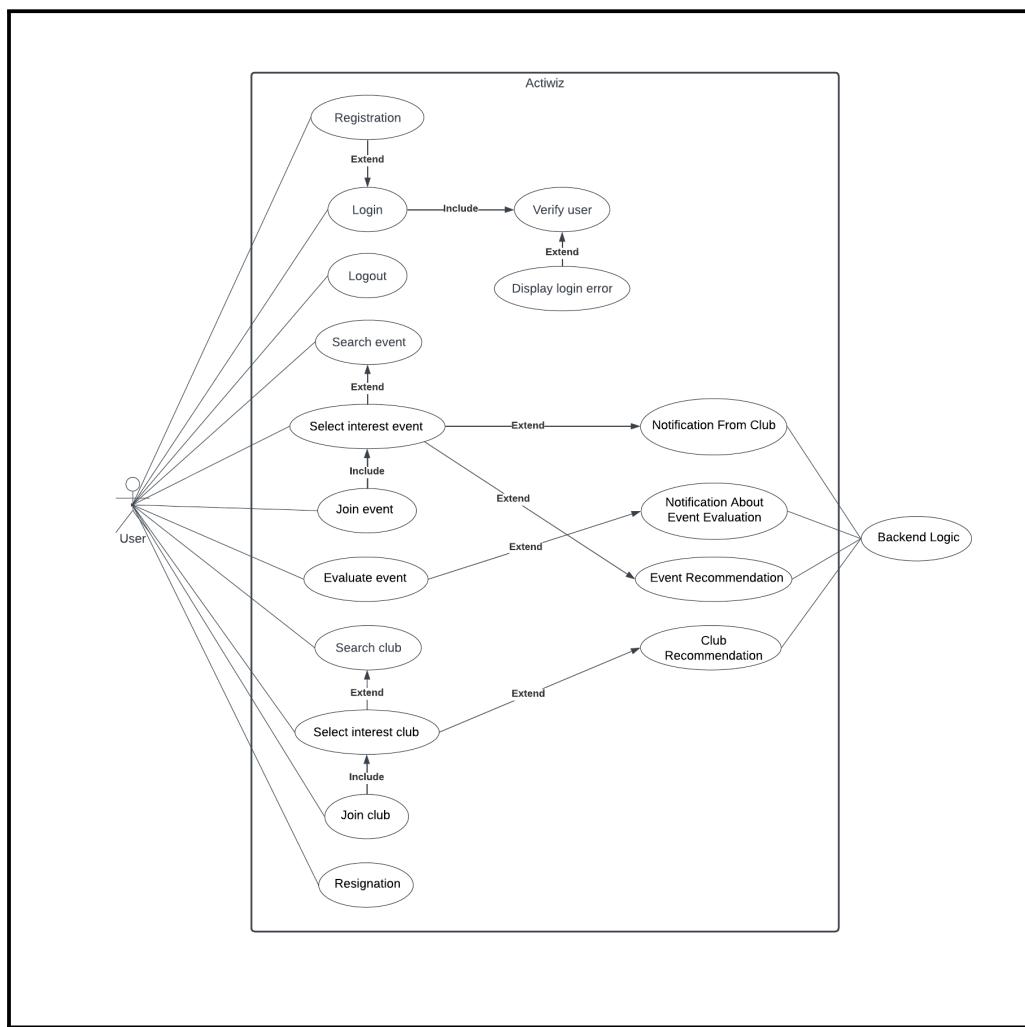
รูปที่ 3.1: Architecture diagram

จากรูปที่ 3.1 จะเป็นแผนภาพแสดงรายละเอียดโครงสร้างของระบบ เพื่อแสดงให้เห็นภาพรวมขององค์ประกอบและเครื่องมือต่าง ๆ ที่ใช้ในการพัฒนาแอปพลิเคชัน Activiz โดยตัวแอปพลิเคชันนั้นรองรับการทำงานทั้งในระบบปฏิบัติการทั้ง 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 ที่ถูกสร้างขึ้น ด้วย Neo4j database

ผังงานระบบ (System Flow)

1. User ใช้งาน React Native Application
2. React Native ส่ง HTTP request ตามรูปแบบของ Rest API ที่กำหนดไว้ไปยัง 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

จากรูปที่ 3.2 เป็นแผนภาพที่แสดงให้เห็นความสัมพันธ์ว่าแต่ละระบบมีความเกี่ยวข้องกันอย่างไรและผู้มีส่วนเกี่ยวข้องในระบบสามารถทำอะไรได้บ้าง โดยในฝั่งของผู้ใช้งานจะเริ่มตั้งแต่สามารถสมัครบัญชีการใช้งานได้ การที่เข้าสู่ระบบเพื่อใช้งานคุณลักษณะต่าง ๆ เช่น การค้นหา, อ่านรายละเอียดและเข้าร่วมกิจกรรม การประมวลผลกิจกรรม การค้นหา, เข้าร่วม และลาออกจากชุมชน รวมถึงงานที่ระบบ Backend กระทำต่อระบบ ทั้งการแนะนำชุมชนหรือกิจกรรม แจ้งเตือนกิจกรรมที่ซึ่งมีจัดหรือการประมวลผลกิจกรรมที่เกิดขึ้น

Use Case Narratives ส่วนนี้จะเป็นการบรรยายให้รายละเอียดเกี่ยวกับกรณีการใช้งานระบบต่าง ๆ ในตัวแอปพลิเคชันที่ถูกเขียนไว้

ในแผนภาพที่ใช้แสดงปฏิสัมพันธ์ระหว่างระบบงานและสิ่งที่อยู่นอกระบบงาน ทั้งเป้าหมายของผู้ใช้ เงื่อนไขการใช้งาน และขั้นตอนต่าง ๆ ในกรณีใช้งานนั้นโดยละเอียด โดยจะแบ่งเป็น 3 ส่วนหลัก ได้แก่

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

ซึ่งแอปพลิเคชันที่เป็นผลิตภัณฑ์ของโครงการมี นั้นมีผู้กระทำหลักคือ User และผู้กระทำการรองคือ Backend Logic ซึ่งขั้นตอนที่เกิดขึ้นและเงื่อนไขในการเกิดขึ้นของแต่ละกรณีใช้งานมีรายละเอียดดังที่ระบุไว้ในตารางดังที่ไปเบนี้

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 ใช้งานฟังก์ชันต่าง ๆ 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

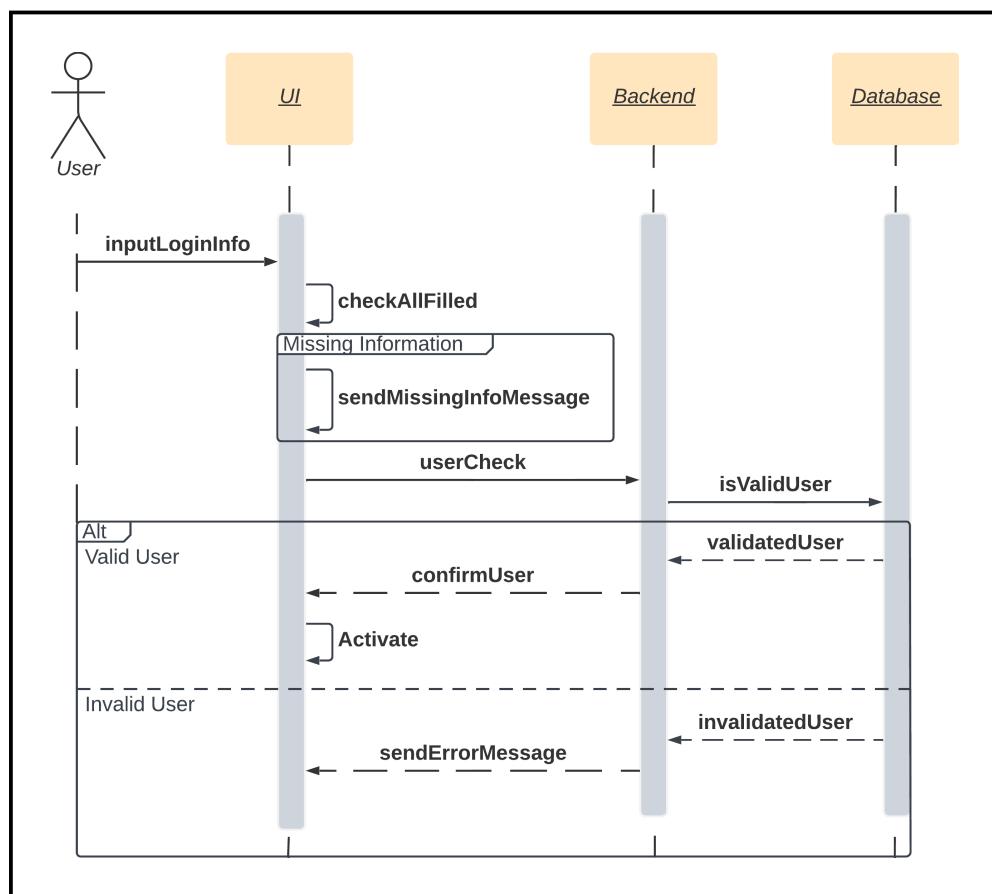
การลากออกจากการชุมชน

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

ตารางที่ 3.11 การลากออกจากการชุมชน

3.6 Sequence diagram

3.6.1 การเข้าสู่ระบบ



รูปที่ 3.3: การเข้าสู่ระบบ

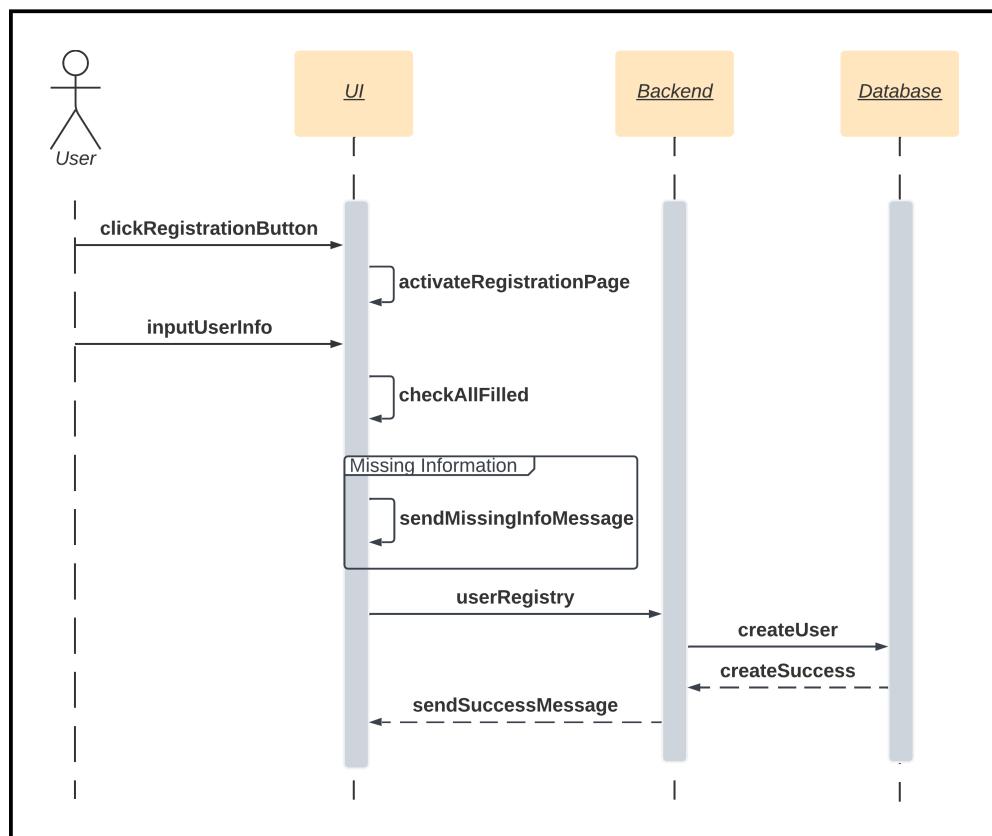
จากภาพ Sequence Diagram ในรูปที่ 3.3 แสดงขั้นตอนการทำงานของระบบเมื่อผู้ใช้ต้องการเข้าสู่ระบบ โดยมีรายละเอียดดังนี้

- ผู้ใช้เริ่มต้นโดยการกรอกข้อมูลที่ใช้ในการเข้าสู่ระบบ (ชื่อผู้ใช้และรหัสผ่าน) ลงในหน้า Log in ของ User Interface (UI)

2. เมื่อผู้ใช้กรอกข้อมูลเรียบร้อยแล้ว UI จะทำการตรวจสอบความถูกต้องและความครบถ้วนของข้อมูลที่ผู้ใช้กรอก หากพบว่าขาดข้อมูลบางส่วน UI จะแจ้งให้ผู้ใช้ทราบและให้กรอกข้อมูลให้ครบถ้วน
3. เมื่อข้อมูลที่จำเป็นสำหรับการเข้าสู่ระบบครบถ้วนแล้ว UI จะส่งข้อมูลดังกล่าวไปยัง Backend System
4. Backend System จะเรียกข้อมูลจาก Database เพื่อตรวจสอบว่าข้อมูลที่ผู้ใช้กรอกมานั้นถูกต้องและตรงกับข้อมูลผู้ใช้ที่มีอยู่ในระบบหรือไม่
5. หากการตรวจสอบผ่าน Backend System จะแจ้งกลับไปยัง UI ว่าผู้ใช้สามารถเข้าสู่ระบบได้
6. UI จะแสดงหน้าจอหลักของระบบให้กับผู้ใช้ เพื่อให้ผู้ใช้สามารถใช้งานฟังก์ชันต่าง ๆ ของระบบได้ต่อไป

โดยสรุป Sequence Diagram นี้แสดงให้เห็นถึงขั้นตอนการทำงานร่วมกันระหว่าง User Interface, Backend System และ Database ในการตรวจสอบและยืนยันตัวตนของผู้ใช้ เพื่อให้ผู้ใช้สามารถเข้าสู่ระบบและใช้งานฟังก์ชันต่าง ๆ ของระบบได้อย่างถูกต้อง

3.6.2 การสมัครเข้าใช้งาน



รูปที่ 3.4: การสมัครเข้าใช้งาน

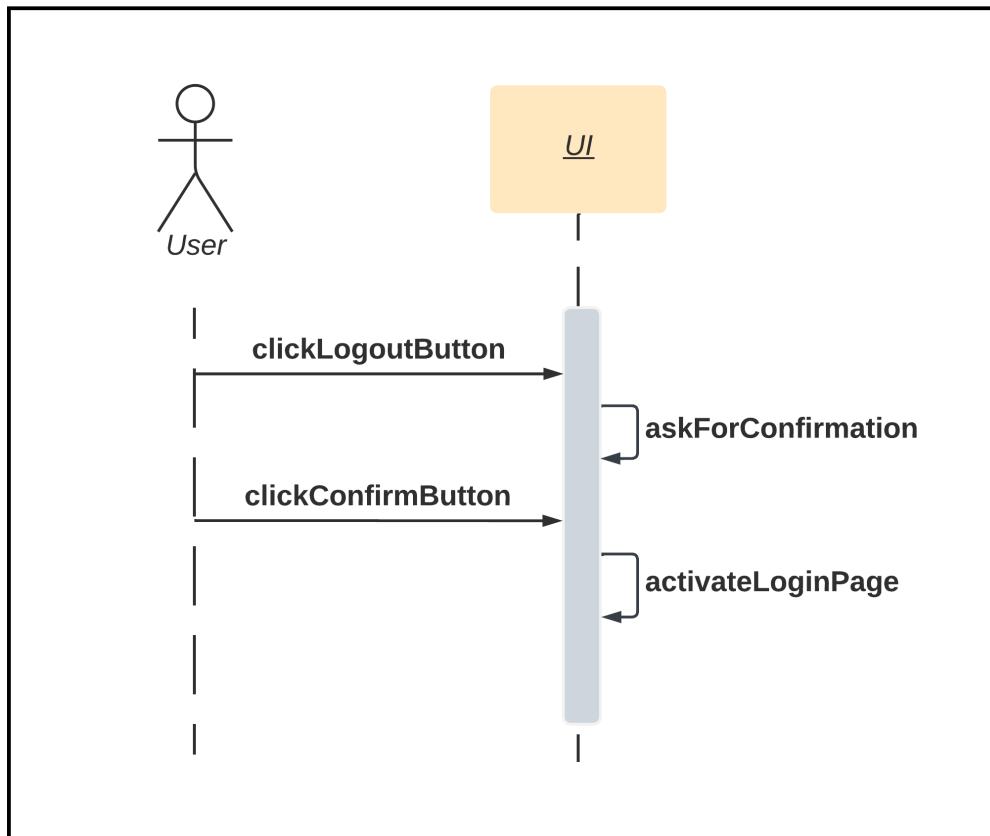
จากภาพ Sequence Diagram ในรูปที่ 3.4 แสดงขั้นตอนการทำงานของระบบเมื่อผู้ใช้ต้องการสมัครเข้าใช้งานระบบ โดยมีรายละเอียดดังนี้

1. ผู้ใช้เริ่มต้นโดยการกดปุ่มสมัครเข้าใช้งานบนหน้า User Interface (UI) ของระบบ
2. หลังจากผู้ใช้กดปุ่มสมัครเข้าใช้งาน UI จะแสดงหน้าจอสำหรับให้ผู้ใช้กรอกข้อมูลที่จำเป็นในการสมัครสมาชิก เช่น ชื่อผู้ใช้ รหัสผ่าน อีเมล ฯลฯ
3. เมื่อผู้ใช้กรอกข้อมูลเรียบร้อยแล้ว UI จะทำการตรวจสอบความถูกต้องและความครบถ้วนของข้อมูลที่ผู้ใช้กรอก หากพบว่าขาดข้อมูลบางส่วน UI จะแจ้งให้ผู้ใช้ทราบและให้กรอกข้อมูลให้ครบถ้วน
4. เมื่อข้อมูลที่จำเป็นสำหรับการสมัครสมาชิกครบถ้วนแล้ว UI จะส่งข้อมูลดังกล่าวไปยัง Backend System

5. Backend System จะบันทึกข้อมูลผู้ใช้รายใหม่ลงใน Database
6. หลังจากบันทึกข้อมูลเสร็จสิ้น Backend System จะส่งข้อความยืนยันการสมัครสมาชิกสำหรับลับไปยัง UI
7. UI จะแสดงข้อความยืนยันการสมัครสมาชิกสำหรับผู้ใช้ท่าน

โดยสรุป Sequence Diagram นี้แสดงให้เห็นถึงขั้นตอนการทำงานร่วมกันระหว่าง User Interface, Backend System และ Database ในการรับข้อมูลการสมัครสมาชิกจากผู้ใช้ ตรวจสอบความถูกต้องและความครบถ้วนของข้อมูล บันทึกข้อมูลลงใน Database และแจ้งผลการสมัครสมาชิกกลับไปยังผู้ใช้

3.6.3 การออกจากระบบ

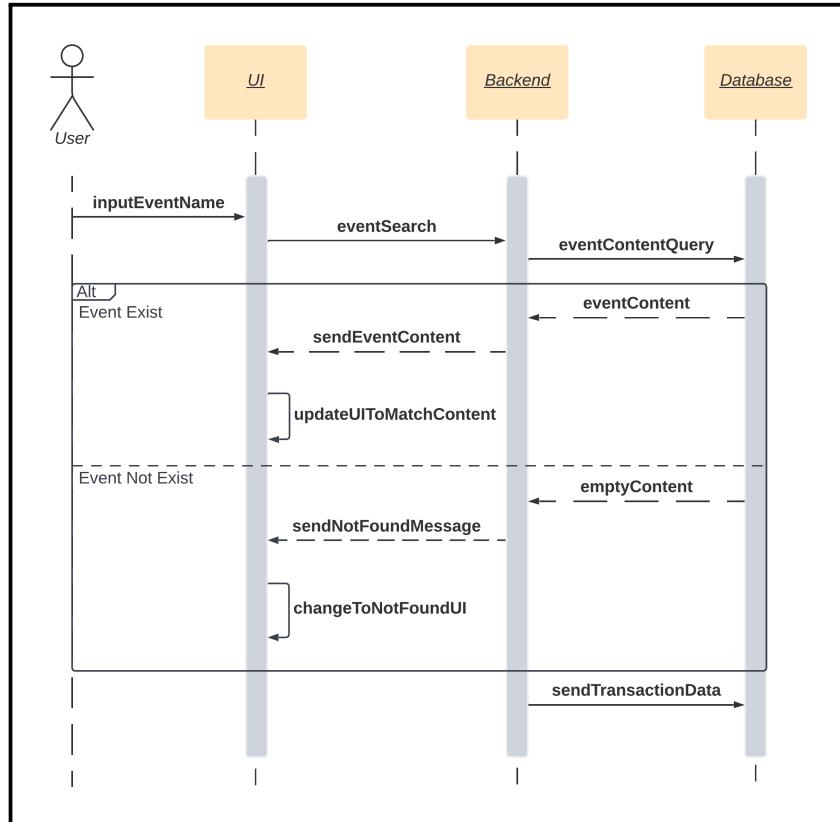


รูปที่ 3.5: การออกจากระบบ

- จากภาพ Sequence Diagram ในรูปที่ 3.5 แสดงขั้นตอนการทำงานของระบบเมื่อผู้ใช้ต้องการออกจากระบบ โดยมีรายละเอียดดังนี้
1. ผู้ใช้เริ่มต้นโดยการกดปุ่มออกจากระบบ (Log Out) บนหน้า User Interface (UI) ของระบบ
 2. หลังจากผู้ใช้กดปุ่มออกจากระบบ UI จะแสดงหน้าจอยืนยันการออกจากระบบ (Confirm Log Out) ให้ผู้ใช้อ่านเพื่อป้องกันการออกจากระบบโดยไม่ได้ตั้งใจ
 3. หากผู้ใช้ยืนยันการออกจากระบบ UI จะส่งคำสั่งออกจากระบบไปยัง Backend System
 4. Backend System จะทำการปิดการเชื่อมต่อ (Session) ของผู้ใช้ร้านค้าและลบข้อมูลการเข้าสู่ระบบ (Log Out) ของผู้ใช้ออกจากระบบ
 5. หลังจากออกจากระบบเสร็จสิ้น Backend System จะแจ้งผลการออกจากระบบกลับไปยัง UI
 6. UI จะนำพาผู้ใช้กลับไปยังหน้าจอเข้าสู่ระบบ (Log In) เพื่อให้ผู้ใช้สามารถเข้าสู่ระบบใหม่ได้อีกรอบ

โดยสรุป Sequence Diagram นี้แสดงให้เห็นถึงขั้นตอนการทำงานร่วมกันระหว่าง User Interface และ Backend System ในการดำเนินการออกจากระบบของผู้ใช้ โดยมีการยืนยันการออกจากระบบจากผู้ใช้ก่อน เพื่อป้องกันการออกจากระบบโดยไม่ได้ตั้งใจ หลังจากนั้น Backend System จะปิดการเชื่อมต่อและลบข้อมูลการเข้าสู่ระบบของผู้ใช้ออกจากระบบ และนำผู้ใช้กลับไปยังหน้าจอเข้าสู่ระบบ

3.6.4 การค้นหาภารกิจกรรม



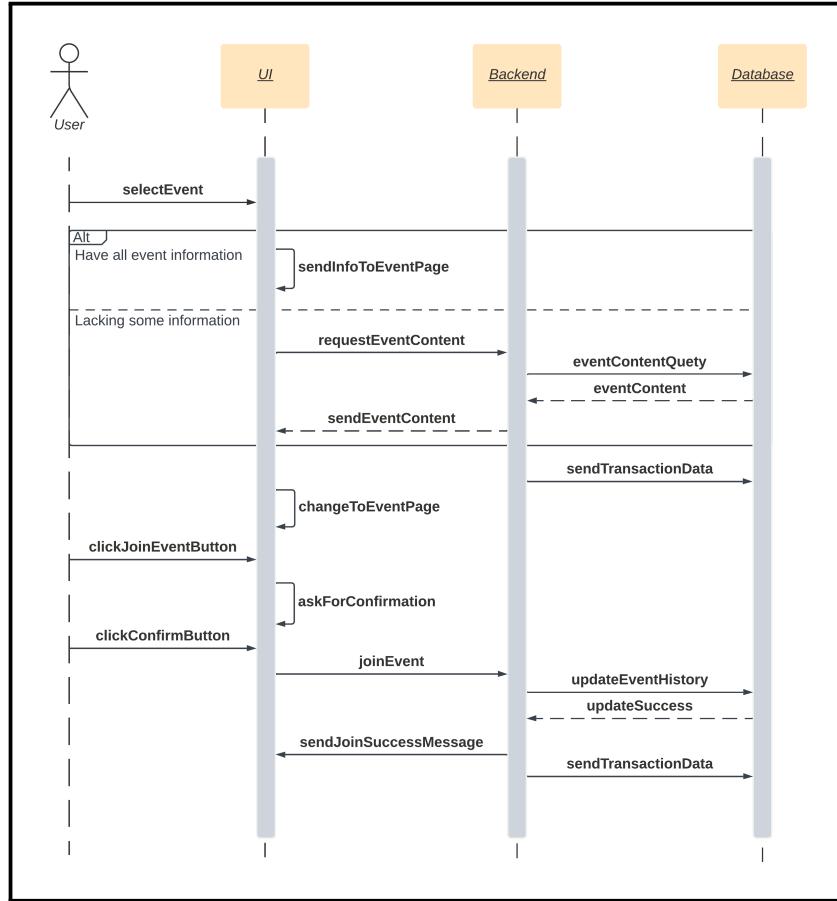
รูปที่ 3.6: การค้นหาภารกิจกรรม

จากภาพ Sequence Diagram ในรูปที่ 3.6 แสดงขั้นตอนการทำงานของระบบเมื่อผู้ใช้ต้องการค้นหาภารกิจกรรม โดยมีรายละเอียดดังนี้

1. ผู้ใช้เริ่มต้นโดยการป้อนคำค้นหาหรือเงื่อนไขที่ต้องการค้นหาภารกิจกรรมลงในช่องค้นหาบนหน้า User Interface (UI) ของระบบ
2. หลังจากผู้ใช้ป้อนคำค้นหาเรียบร้อยแล้ว UI จะส่งคำค้นหาหรือเงื่อนไขดังกล่าวไปยัง Backend System
3. Backend System จะทำการค้นหาข้อมูลภารกิจกรรมที่ตรงกับคำค้นหาหรือเงื่อนไขที่ได้รับมาจาก Database
4. หากพบข้อมูลภารกิจกรรมที่ตรงกับคำค้นหาหรือเงื่อนไข Backend System จะส่งข้อมูลภารกิจกรรมเหล่านั้นกลับไปยัง UI
5. UI จะแสดงข้อมูลภารกิจกรรมที่ค้นหาได้ให้ผู้ใช้เห็น
6. หากไม่พบข้อมูลภารกิจกรรมที่ตรงกับคำค้นหาหรือเงื่อนไข Backend System จะแจ้งกลับไปยัง UI ว่าไม่พบข้อมูลภารกิจกรรมที่ค้นหา
7. UI จะแสดงข้อความ ”ไม่พบภารกิจกรรมที่ค้นหา” ให้ผู้ใช้ทราบ
8. นอกจากนี้ Backend System จะบันทึกข้อมูลการค้นหาของผู้ใช้ลงใน Database ด้วย ไม่ว่าการค้นหานั้นจะประสบความสำเร็จ หรือไม่ก็ตาม

โดยสรุป Sequence Diagram นี้แสดงให้เห็นถึงขั้นตอนการทำงานร่วมกันระหว่าง User Interface, Backend System และ Database ในการค้นหาข้อมูลภารกิจกรรมตามคำค้นหาหรือเงื่อนไขที่ผู้ใช้ต้องการ โดยจะแสดงผลการค้นหาให้ผู้ใช้ทราบ และบันทึกข้อมูลการค้นหาลงใน Database ไม่ว่าผลการค้นหาจะสำเร็จหรือไม่ก็ตาม

3.6.5 การอ่านรายละเอียดของกิจกรรมและเข้าร่วมกิจกรรม



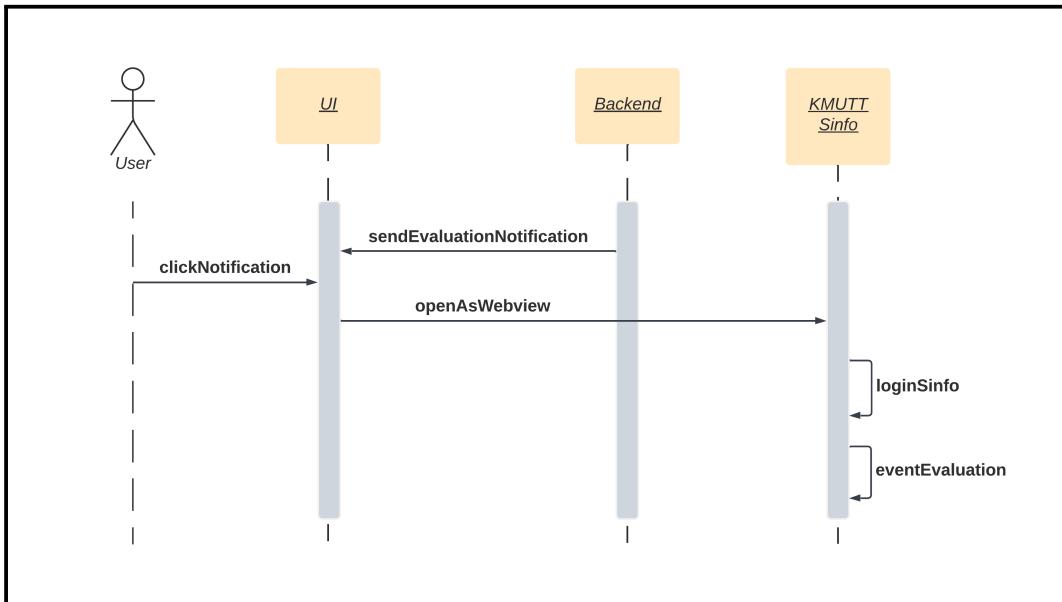
รูปที่ 3.7: การอ่านรายละเอียดของกิจกรรมและเข้าร่วมกิจกรรม

จากภาพ Sequence Diagram ในรูปที่ 3.7 แสดงขั้นตอนการทำงานของระบบเมื่อผู้ใช้ต้องการอ่านรายละเอียดของกิจกรรม และเข้าร่วมกิจกรรมนั้น โดยมีรายละเอียดดังนี้

1. ผู้ใช้เลือกกิจกรรมที่ต้องการอ่านรายละเอียดจากหน้า User Interface (UI) ของระบบ
2. UI จะตรวจสอบว่ามีข้อมูลรายละเอียดของกิจกรรมดังกล่าวอยู่ในระบบหรือไม่
3. หากมีข้อมูลรายละเอียดของกิจกรรมอยู่แล้ว UI จะแสดงข้อมูลรายละเอียดนั้นให้ผู้ใช้เห็น
4. หากไม่มีข้อมูลรายละเอียดของกิจกรรม UI จะส่งคำขอไปยัง Backend System เพื่อให้ได้ข้อมูลรายละเอียดของกิจกรรมมาจาก Database
5. Backend System จะค้นหาและดึงข้อมูลรายละเอียดของกิจกรรมที่ต้องการจาก Database และส่งข้อมูลรายละเอียดกิจกรรมกลับไปยัง UI
6. UI จะแสดงข้อมูลรายละเอียดของกิจกรรมให้ผู้ใช้เห็น
7. หากผู้ใช้ต้องการเข้าร่วมกิจกรรมนั้น ผู้ใช้จะกดปุ่มเข้าร่วมกิจกรรมบนหน้า UI และจะแสดงข้อความยืนยันการเข้าร่วมชุมชนให้ผู้ใช้
8. หากผู้ใช้ยืนยันการเข้าร่วมกิจกรรม UI จะส่งข้อมูลการเข้าร่วมกิจกรรมของผู้ใช้ไปยัง Backend System
9. Backend System จะบันทึกข้อมูลการเข้าร่วมกิจกรรมของผู้ใช้ลงใน Database

โดยรูป Sequence Diagram นี้แสดงให้เห็นถึงขั้นตอนการทำงานร่วมกันระหว่าง User Interface, Backend System และ Database ในการแสดงรายละเอียดของกิจกรรมให้ผู้ใช้ และรับข้อมูลการเข้าร่วมกิจกรรมจากผู้ใช้ โดยมีการยืนยันการเข้าร่วมกิจกรรมจากผู้ใช้ก่อนเพื่อป้องกันการเข้าร่วมกิจกรรมโดยไม่ได้ตั้งใจ

3.6.6 การประเมินกิจกรรม



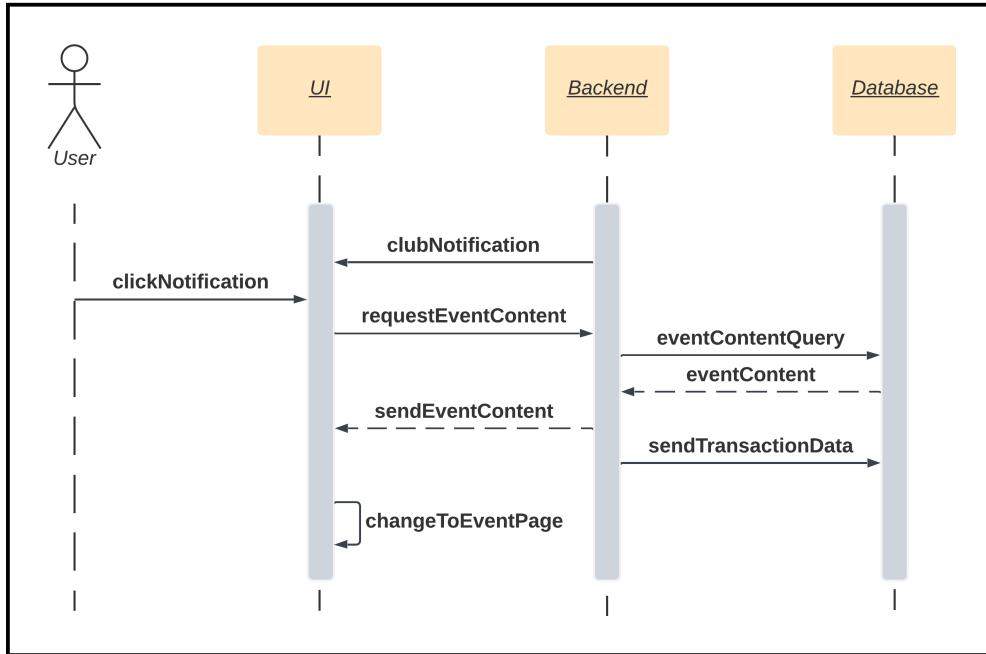
รูปที่ 3.8: การประเมินกิจกรรม

จากภาพ Sequence Diagram ในรูปที่ 3.8 แสดงขั้นตอนการทำงานของระบบแจ้งเตือนการประเมินกิจกรรม โดยมีรายละเอียดดังนี้

1. Backend System จะตรวจสอบกำหนดการหรือเงื่อนไขที่ต้องแจ้งเตือนการประเมินกิจกรรมให้กับผู้ใช้ เช่น หลังจากกิจกรรมสิ้นสุดลงภายในระยะเวลาที่กำหนด
2. เมื่อถึงกำหนดการหรือเงื่อนไขที่ต้องแจ้งเตือน Backend System จะส่งคำเตือนการประเมินกิจกรรมไปยัง User Interface (UI) ของผู้ใช้
3. UI จะแสดงข้อความแจ้งเตือนการประเมินกิจกรรมให้ผู้ใช้ทราบ พร้อมกับลิงก์หรือปุ่มสำหรับเข้าไปประเมินกิจกรรม
4. เมื่อผู้ใช้กดเข้าลิงก์หรือปุ่มประเมินกิจกรรม UI จะเปิดหน้าต่างหรือหน้าจอสำหรับให้ผู้ใช้ประเมินกิจกรรมนั้น เพื่อให้ผู้กรอกข้อมูลและประเมินกิจกรรมลงในหน้าประเมินกิจกรรม
5. หลังจากผู้ใช้กรอกข้อมูลการประเมินกิจกรรมเสร็จแล้ว UI จะส่งข้อมูลการประเมินดังกล่าวไปยัง Backend System และจะบันทึกข้อมูลการประเมินกิจกรรมของผู้ใช้ลงใน Database

โดยสรุป Sequence Diagram นี้แสดงให้เห็นถึงขั้นตอนการทำงานร่วมกันระหว่าง Backend System, User Interface และ Database ในการแจ้งเตือนให้ผู้ใช้ประเมินกิจกรรม โดย Backend System จะเป็นผู้ส่งคำเตือนไปยัง UI เพื่อให้ UI แสดงข้อความแจ้งเตือนและลิงก์สำหรับเข้าประเมินกิจกรรมให้ผู้ใช้ หลังจากผู้ใช้ประเมินกิจกรรมเสร็จแล้ว ข้อมูลการประเมินจะถูกส่งกลับไปบันทึกยัง Backend System และ Database

3.6.7 การแจ้งเตือนกิจกรรมชุมชน



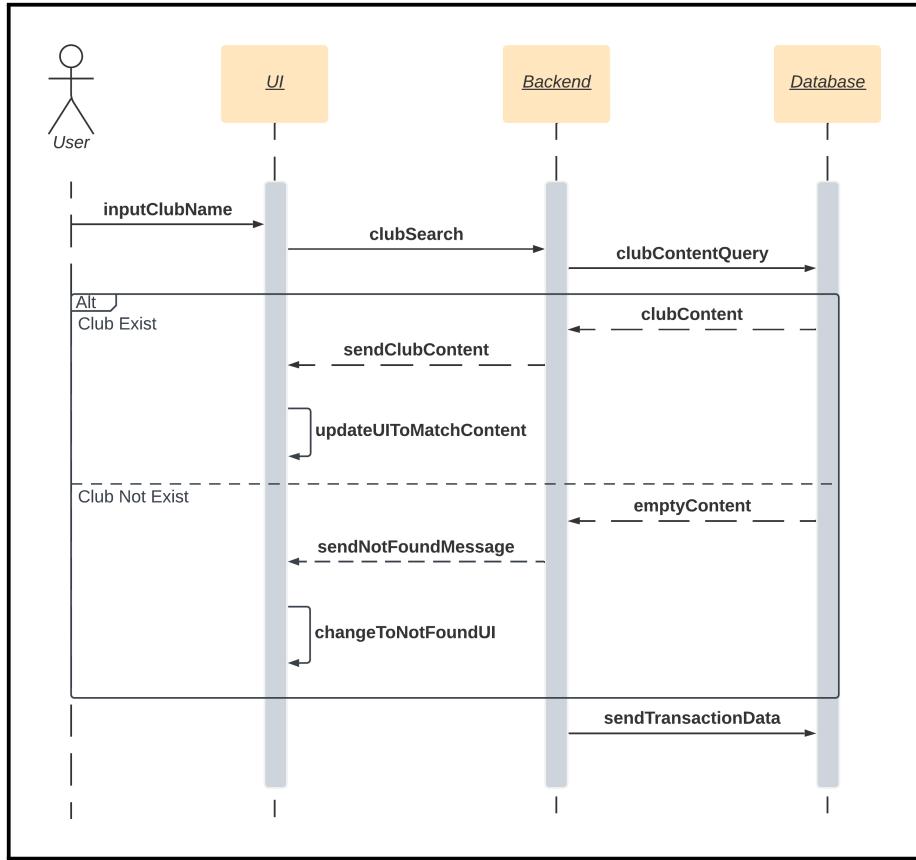
รูปที่ 3.9: การแจ้งเตือนกิจกรรมชุมชน

จากภาพ Sequence Diagram ในรูปที่ 3.9 แสดงขั้นตอนการทำงานของระบบแจ้งเตือนกิจกรรมชุมชน โดยมีรายละเอียดดังนี้

- Backend System จะตรวจสอบกำหนดการหรือเงื่อนไขที่ต้องแจ้งเตือนกิจกรรมชุมชนให้กับผู้ใช้ เช่น ก่อนถึงวันที่มีกิจกรรมชุมชน ตามระยะเวลาที่กำหนด
- เมื่อถึงกำหนดการหรือเงื่อนไขที่ต้องแจ้งเตือน Backend System จะส่งข้อความแจ้งเตือนกิจกรรมชุมชนไปยัง User Interface (UI) ของผู้ใช้
- UI จะแสดงข้อความแจ้งเตือนกิจกรรมชุมชนให้ผู้ใช้ทราบ พร้อมกับลิงก์หรือปุ่มสำหรับเข้าไปอ่านรายละเอียดของกิจกรรม
- เมื่อผู้ใช้กดลิงก์หรือปุ่มอ่านรายละเอียดกิจกรรม UI จะส่งคำขอไปยัง Backend System เพื่อให้ดึงข้อมูลรายละเอียดของกิจกรรมมาจาก Database
- Backend System จะค้นหาและดึงข้อมูลรายละเอียดของกิจกรรมที่ต้องการจาก Database
- Backend System จะส่งข้อมูลรายละเอียดของกิจกรรมกลับไปยัง UI
- UI จะแสดงข้อมูลรายละเอียดของกิจกรรมให้ผู้ใช้เห็น
- นอกจากนี้ Backend System จะบันทึกข้อมูลการใช้งานของผู้ใช้ เช่น การเข้าอ่านรายละเอียดกิจกรรม ลงใน Database ด้วย

โดยสรุป Sequence Diagram นี้แสดงให้เห็นถึงขั้นตอนการทำงานร่วมกันระหว่าง Backend System, User Interface และ Database ในการแจ้งเตือนกิจกรรมชุมชนให้ผู้ใช้ โดย Backend System จะเป็นผู้ส่งข้อความแจ้งเตือนไปยัง UI เพื่อให้ UI แสดงข้อความแจ้งเตือนและลิงก์สำหรับเข้าอ่านรายละเอียดกิจกรรมให้ผู้ใช้ เมื่อผู้ใช้เข้าอ่านรายละเอียดกิจกรรม UI จะขอข้อมูลรายละเอียดจาก Backend System ซึ่งจะดึงมาจาก Database และข้อมูลการใช้งานของผู้ใช้จะบันทึกลงใน Database ด้วย

3.6.8 การค้นหาชุมชน



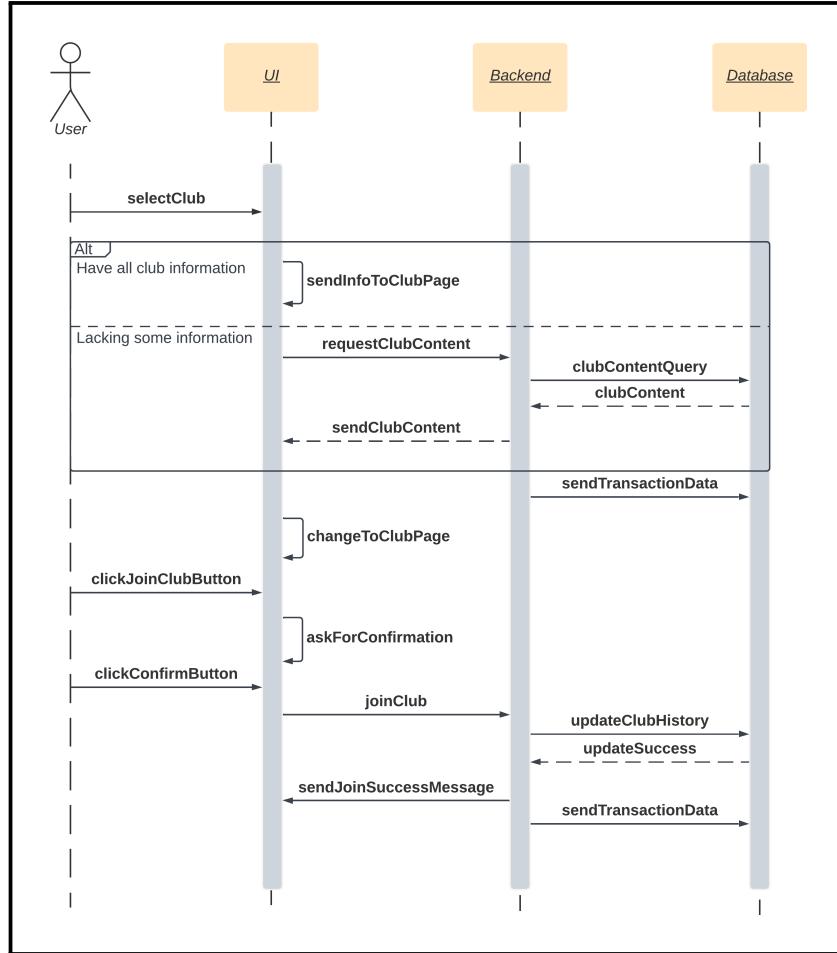
รูปที่ 3.10: การค้นหาชุมชน

จากภาพ Sequence Diagram ในรูปที่ 3.10 แสดงขั้นตอนการทำงานของระบบเมื่อผู้ใช้ต้องการค้นหาชุมชน โดยมีรายละเอียดดังนี้

1. ผู้ใช้เริ่มต้นโดยการป้อนคำค้นหาหรือเงื่อนไขที่ต้องการค้นหาชุมชนลงในช่องค้นหานั้น User Interface (UI) ของระบบ
2. หลังจากผู้ใช้ป้อนคำค้นหาเรียบร้อยแล้ว UI จะส่งคำค้นหาหรือเงื่อนไขดังกล่าวไปยัง Backend System
3. Backend System จะทำการค้นหาข้อมูลชุมชนที่ตรงกับคำค้นหาหรือเงื่อนไขที่ได้รับมาจาก Database
4. หากพบข้อมูลชุมชนที่ตรงกับคำค้นหาหรือเงื่อนไข Backend System จะส่งข้อมูลชุมชนเหล่านั้นกลับไปยัง UI
5. UI จะแสดงข้อมูลชุมชนที่ค้นหาได้ให้ผู้ใช้เห็น
6. หากไม่พบข้อมูลชุมชนที่ตรงกับคำค้นหาหรือเงื่อนไข Backend System จะแจ้งกลับไปยัง UI ว่าไม่พบข้อมูลชุมชนที่ค้นหา
7. UI จะแสดงข้อความ "ไม่พบชุมชนที่ค้นหา" ให้ผู้ใช้ทราบ
8. นอกเหนือจาก Backend System จะบันทึกข้อมูลการค้นหาชุมชนของผู้ใช้ลงใน Database ด้วย ไม่ว่าการค้นหานั้นจะประสบความสำเร็จหรือไม่ก็ตาม

โดยสรุป Sequence Diagram นี้แสดงให้เห็นถึงขั้นตอนการทำงานร่วมกันระหว่าง User Interface, Backend System และ Database ในการค้นหาข้อมูลชุมชนตามคำค้นหาหรือเงื่อนไขที่ผู้ใช้ต้องการ โดยจะแสดงผลการค้นหาให้ผู้ใช้ทราบ และบันทึกข้อมูลการค้นหาลงใน Database ไม่ว่าผลการค้นหาจะสำเร็จหรือไม่ก็ตาม

3.6.9 อ่านรายละเอียดของชมรมและการเข้าร่วมชมรม



รูปที่ 3.11: อ่านรายละเอียดของชมรมและการเข้าร่วมชมรม

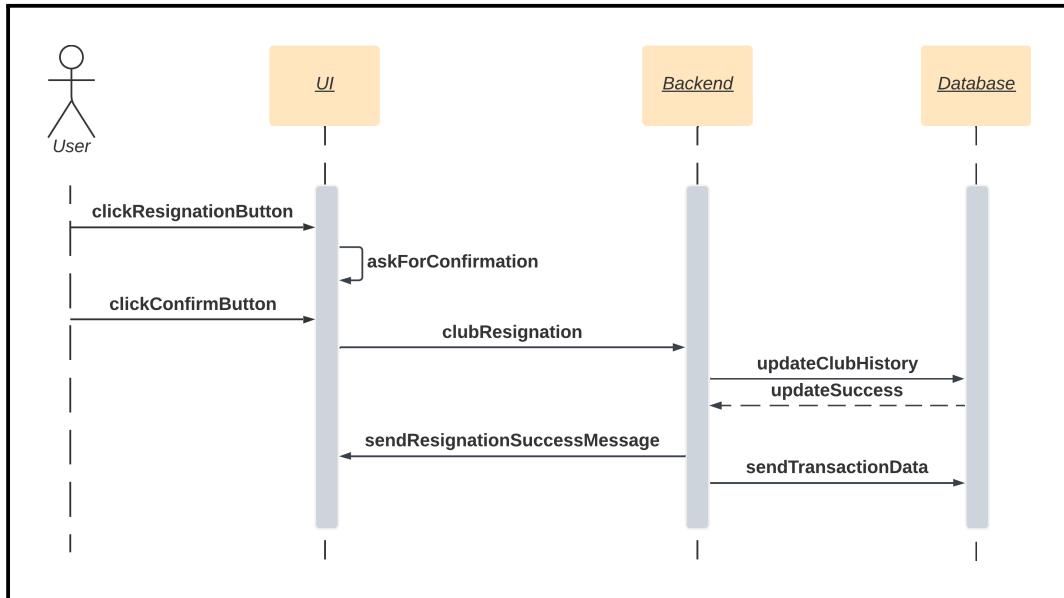
จากภาพ Sequence Diagram ในรูปที่ 3.11 แสดงขั้นตอนการทำงานของระบบเมื่อผู้ใช้ต้องการอ่านรายละเอียดของชมรม และเข้าร่วมชมรมนั้น โดยมีรายละเอียดดังนี้

1. ผู้ใช้เลือกชมรมที่ต้องการอ่านรายละเอียดจากหน้า User Interface (UI) ของระบบ
2. UI จะตรวจสอบว่ามีข้อมูลรายละเอียดของชมรมดังกล่าวอยู่ในระบบหรือไม่ และจะแสดงข้อมูลรายละเอียดนั้นให้ผู้ใช้เห็น
3. หากไม่มีข้อมูลรายละเอียดของชมรม UI จะส่งคำขอไปยัง Backend System เพื่อให้ดึงข้อมูลรายละเอียดของชมรมมาจาก Database
4. Backend System จะค้นหาและดึงข้อมูลรายละเอียดของชมรมที่ต้องการจาก Database และส่งไปยัง UI และแสดงให้ผู้ใช้เห็น
5. หากผู้ใช้ต้องการเข้าร่วมชมรมนั้น ผู้ใช้จะกดปุ่มเข้าร่วมชมรมบนหน้า UI และจะแสดงข้อความยืนยันการเข้าร่วมชมรมให้ผู้ใช้เห็น
6. หากผู้ใช้ยืนยันการเข้าร่วมชมรม UI จะส่งข้อมูลการเข้าร่วมชมรมของผู้ใช้ไปยัง Backend System
7. Backend System จะบันทึกข้อมูลการเข้าร่วมชมรมของผู้ใช้ลงใน Database

โดยสรุป Sequence Diagram นี้แสดงให้เห็นถึงขั้นตอนการทำงานร่วมกันระหว่าง User Interface, Backend System และ Database ในการแสดงรายละเอียดของชมรมให้ผู้ใช้ และรับข้อมูลการเข้าร่วมชมรมจากผู้ใช้ โดยมีการยืนยันการเข้าร่วมชมรมจากผู้ใช้ก่อน เพื่อป้องกันการเข้าร่วมชมรมโดยไม่ได้ตั้งใจ

3.6.10 การลาออกจากชมรม

จากภาพ Sequence diagram ในรูปที่ 3.12 ซึ่งแสดงการทำงานของระบบขณะที่ผู้ใช้ทำการลาออกจากชมรม ดังนี้

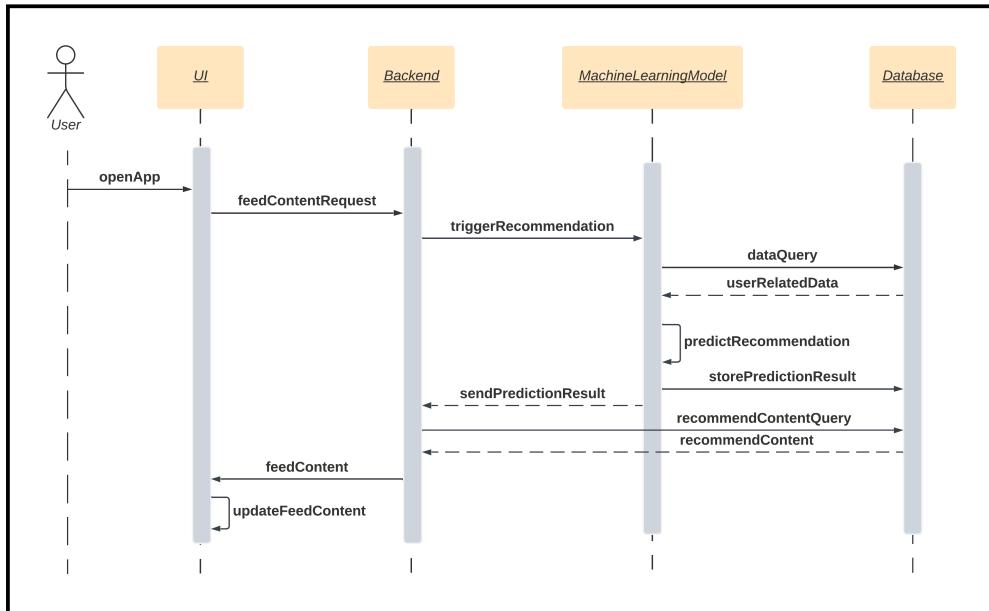


รูปที่ 3.12: การลาออกจากชุมชน

1. เริ่มต้นเมื่อผู้ใช้กดปุ่มลาออกจากส่วนติดต่อผู้ใช้ (UI)
2. UI จะแสดงข้อความยืนยันการลาออกจากชุมชนให้ผู้ใช้
3. หากผู้ใช้ยืนยันการลาออกจาก UI จะส่งข้อมูลการร้องขอลาออกจาก Backend
4. Backend จะประมวลผลข้อมูลการลาออกจากแล็บบันทึกลงในฐานข้อมูล (Database)
5. หลังจากบันทึกข้อมูลเรียบร้อยแล้ว Backend จะส่งข้อความยืนยันการลาออกจากลับไปที่ UI
6. UI จะแสดงข้อความยืนยันการลาออกให้ผู้ใช้ทราบ

โดยสรุป Sequence diagram นี้แสดงถึงขั้นตอนการทำงานระหว่างส่วนติดต่อผู้ใช้ (UI) กับระบบหลังบ้าน (Backend) และฐานข้อมูล (Database) ในกระบวนการลาออกจากชุมชนของผู้ใช้ ซึ่งประกอบด้วยการยืนยันจากผู้ใช้ การประมวลผลและบันทึกข้อมูลจาก Backend และการแสดงผลข้อความยืนยันจาก UI

3.6.11 การแนะนำกิจกรรมให้ผู้ใช้



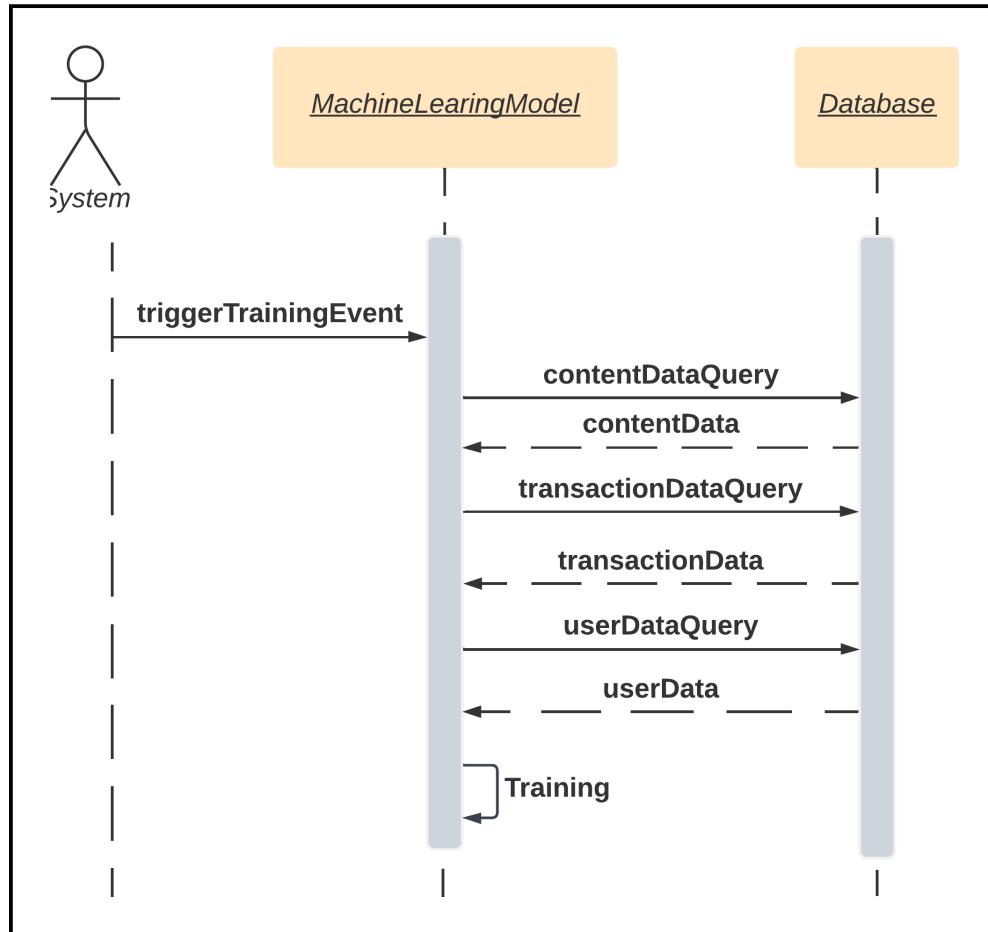
รูปที่ 3.13: การแนะนำกิจกรรมให้ผู้ใช้

จากภาพ Sequence diagram ในรูปที่ 3.13 ชี้แสดงการทำงานของระบบในการแนะนำกิจกรรมให้แก่ผู้ใช้ ดังนี้

1. เริ่มต้นเมื่อผู้ใช้เปิดแอปพลิเคชันบนส่วนติดต่อผู้ใช้ (UI)
2. UI จะส่งคำร้องขอเนื้อหาเพื่อแนะนำกิจกรรมไปยังระบบหลังบ้าน (Backend)
3. Backend จะส่งคำร้องขอไปยังระบบ Machine Learning Model เพื่อประมวลผลคำแนะนำกิจกรรม
4. Machine Learning Model จะดึงข้อมูลประวัติและพฤติกรรมของผู้ใช้จากฐานข้อมูล (Database) มาประมวลผล
5. หลังประมวลผลเสร็จ Machine Learning Model จะส่งคำแนะนำกิจกรรมกลับไปที่ Backend
6. Backend จะส่งคำแนะนำกิจกรรมไปแสดงบนส่วนติดต่อผู้ใช้ (UI) ให้ผู้ใช้เห็น
7. พร้อมกันนี้ Backend จะบันทึกคำแนะนำกิจกรรมในฐานข้อมูล (Database) เพื่อใช้ในการประมวลผลครั้งต่อไป นอกจากนี้ เมื่อผู้ใช้ต้องการคุ้มครองข้อมูลความเป็นส่วนตัว สามารถทำได้โดย
8. UI ส่งคำร้องขอคุ้มครองข้อมูลความเป็นส่วนตัวไปยัง Backend
9. Backend ดึงข้อมูลคำแนะนำกิจกรรมย้อนหลังจากฐานข้อมูล (Database)
10. Backend ส่งคำแนะนำกิจกรรมย้อนหลังไปแสดงบน UI ให้ผู้ใช้เห็น

โดยสรุป Sequence diagram นี้แสดงถึงการทำงานร่วมกันของ UI, Backend, Machine Learning Model และ Database ในการแนะนำกิจกรรมให้แก่ผู้ใช้ตามประวัติและพฤติกรรม รวมถึงการบันทึกและแสดงประวัติคำแนะนำย้อนหลัง

3.6.12 การ Train Model การเรียนรู้ของเครื่อง



รูปที่ 3.14: การ Train Model การเรียนรู้ของเครื่อง

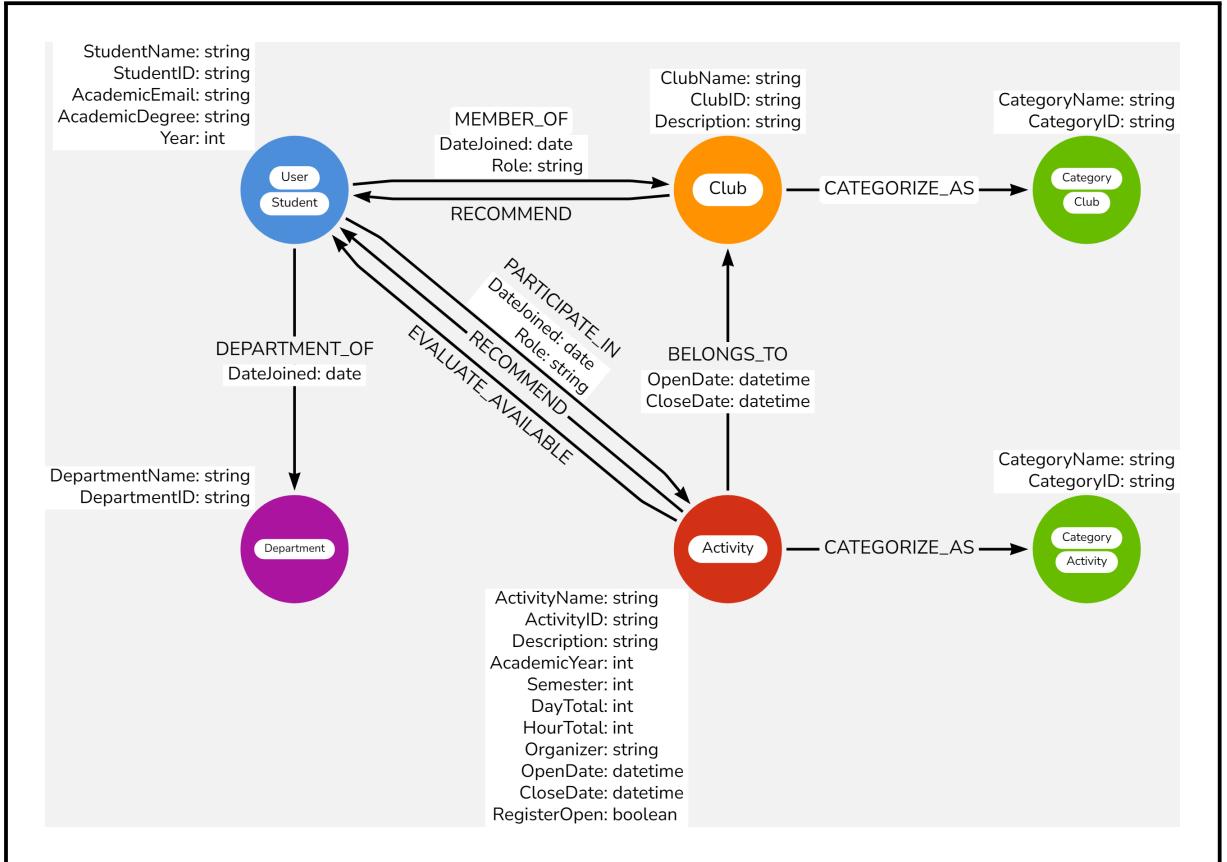
จากภาพ Sequence Diagram ในรูปที่ 3.14 ซึ่งแสดงการทำงานของระบบในขณะที่มีการ Train Model การเรียนรู้ของเครื่องใหม่ สามารถขยายความได้ดังนี้

1. เริ่มต้นเมื่อระบบต้องการทำการ Train หรือปรับปรุงประสิทธิภาพของ Machine Learning Model
2. Machine Learning Model จะส่งคำร้องขอข้อมูลกิจกรรมและการใช้งานของผู้ใช้ไปยังฐานข้อมูล (Database)
3. ฐานข้อมูลจะรวบรวมและส่งข้อมูลที่เกี่ยวข้อง ได้แก่ ข้อมูลกิจกรรม พฤติกรรมการใช้งาน ฯลฯ ไปให้ Machine Learning Model
4. Machine Learning Model จะนำข้อมูลเหล่านี้มาใช้ในการ Train ตัวเองให้มีประสิทธิภาพในการคาดการณ์และแนะนำที่ดียิ่งขึ้น โดยอาจใช้เทคนิคต่าง ๆ เช่น Supervised Learning, Unsupervised Learning, Deep Learning เป็นต้น
5. หลังจากการ Train เสร็จสิ้น Machine Learning Model จะมีโมเดลที่ถูกปรับปรุงให้ดียิ่งขึ้น

การ Train Model นั้นก็จะเกิดขึ้นเป็นระยะๆ เพื่อให้ได้โมเดลที่มีประสิทธิภาพสูงสุดในการทำนายและแนะนำสำหรับผู้ใช้ โดยจะนำข้อมูลการใช้งานจริงของผู้ใช้ที่เก็บสะสมไว้ในฐานข้อมูลมาใช้ในการปรับปรุงโมเดล กระบวนการ Train Model นี้เป็นสิ่งสำคัญสำหรับระบบที่ใช้ Machine Learning หรือ AI ในการให้บริการ เพื่อเพิ่มความถูกต้องและความแม่นยำในการทำนายและแนะนำให้ตรงตามความต้องการของผู้ใช้มากที่สุด

3.7 แบบจำลองข้อมูลแบบกราฟ (Graph Data Model)

แบบจำลองข้อมูลแบบกราฟเป็นแบบจำลองเพื่อแสดงความเชื่อมโยงของโหนดที่เก็บข้อมูลของระบบ ว่าแต่ละโหนดมีความสัมพันธ์กันอย่างไรภายในแอปพลิเคชัน Activiz ของเรา แบบจำลองข้อมูลแบบกราฟจะตอบโจทย์การนำไปประยุกต์ในระบบฐานข้อมูลแบบกราฟ เช่น Neo4j ช่วยให้สามารถสำรวจและเรียกค้นข้อมูลที่เกี่ยวข้องได้อย่างมีประสิทธิภาพ



รูปที่ 3.15: Graph Data Model

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

3.7.1 โหนด

จากที่กล่าวมาโหนดจะทำหน้าที่ในการเก็บข้อมูลซึ่งจะแบ่งข้อมูลในแต่ละโหนดออกมาให้เป็นตารางได้ตามนี้

3.7.1.1 โหนด User: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดส่วนตัวของผู้ใช้ภายในมหาวิทยาลัย

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 ตารางเก็บข้อมูลของผู้ใช้ที่ใช้งานแอปพลิเคชัน

จากตารางที่ 3.12 แสดงข้อมูลเฉพาะบุคคลของผู้ใช้แต่ละคน โดย StudentName นั้นเป็นการจัดเก็บข้อมูลชื่อของนักศึกษาผู้ใช้ โดยเก็บข้อมูลเป็นสตริง และยังมีข้อมูลส่วนบุคคลต่าง ๆ ของนักศึกษา เช่น StudentID ที่จัดเก็บรหัสนักศึกษา AcademicEmail จัดเก็บที่อยู่อีเมลของทางมหาลัยที่นักศึกษาคนนั้นใช้งาน และ AcademicDegree ที่จัดเก็บระดับการศึกษาของผู้ใช้ ข้อมูลเหล่านี้จะถูกจัดเก็บไว้เป็นสตริงและข้อมูลสุดท้ายที่ถูกจัดเก็บในโหนดนี้คือ Year ซึ่งจะบันทึกข้อมูลชั้นปีของเจ้าของโหนดไว้เป็นจำนวนเต็ม โดยทุกข้อมูลที่เก็บไว้ในโหนดนี้ไม่สามารถถูกแก้ไขได้

3.7.1.2 โหนด Department: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดของภาควิชาภายในมหาวิทยาลัย

Property	DataType	Constrain	Definition
DepartmentName	STRING	Not NULL	ชื่อภาควิชา
DepartmentID	STRING	Not NULL	ชื่อของภาษาอังกฤษของภาควิชา
Description	STRING	NULL	คำอธิบายชุมชน

ตารางที่ 3.13 ตารางเก็บข้อมูลของภาควิชาภายในมหาวิทยาลัย

จากตารางที่ 3.13 แสดงข้อมูลเกี่ยวกับรายละเอียดของภาควิชาภายในมหาวิทยาลัยที่ความเชื่อมโยงกับนักศึกษาและกิจกรรมที่จัดขึ้นภายในมหาลัย มีข้อมูลถูกเก็บอยู่ในโหนดนี้ คือ DepartmentName ซึ่งเก็บชื่อของภาควิชา DepartmentID คือชื่อในภาษาอังกฤษของภาควิชานั้น ๆ โดยสองข้อมูลนี้จะถูกจัดเก็บไว้เป็นสตริงโดยจะไม่สามารถถูกแก้ไขได้ และข้อมูลสุดท้ายที่ถูกจัดเก็บไว้ในโหนดนี้คือ Description ซึ่งจะเก็บคำอธิบายภาควิชาเอาไว้เป็นสตริง อย่างไรก็ตามข้อมูลนี้สามารถแก้ไขได้

3.7.1.3 โหนด Club: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดของชุมชนภายในมหาวิทยาลัย

Property	DataType	Constrain	Definition
ClubName	STRING	Not NULL	ชื่อชุมชน
ClubID	STRING	Not NULL	รหัสชุมชน
Description	STRING	NULL	คำอธิบายชุมชน

ตารางที่ 3.14 ตารางเก็บข้อมูลของชุมชนภายในมหาวิทยาลัย

จากตารางที่ 3.14 แสดงข้อมูลเกี่ยวกับรายละเอียดของชุมชนภายในมหาวิทยาลัยซึ่งจะเชื่อมโยงไปยังกิจกรรมต่าง ๆ ที่ชุมชนเหล่านี้จัด และประเภทของชุมชนที่ทางผู้จัดทำได้แบ่งประเภทเอาไว้ ข้อมูลที่เก็บไว้ในโหนดเหล่านี้ ได้แก่ ClubName บันทึกข้อมูลของชื่อชุมชนที่มีเอาไว้ ClubID จะเก็บข้อมูลรหัสของชุมชนเอาไว้ โดยข้อมูลทั้งสองส่วนนี้จัดเก็บไว้เป็นสตริงแต่ไม่สามารถแก้ไขได้ และข้อมูลสุดท้ายที่ถูกจัดเก็บไว้ในโหนดนี้คือ Description หรือคำอธิบายชุมชนที่จะบันทึกเป็นสตริง แต่สามารถแก้ไขได้

3.7.1.4 โหนด Club Category: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดประเภทนิรภัยในมหาวิทยาลัย

Property	DataType	Constrain	Definition
CategoryName	STRING	Not NULL	ชื่อประเภท
CategoryID	STRING	Not NULL	รหัสประเภท
Description	STRING	NULL	คำอธิบายประเภทนิรภัย

ตารางที่ 3.15 ตารางเก็บข้อมูลประเภทนิรภัย

จากตารางที่ 3.15 แสดงข้อมูลเกี่ยวกับรายละเอียดประเภทนิรภัยในมหาวิทยาลัยซึ่งทางผู้จัดทำพยายามจัดแบ่งโดยอาศัยจากห้องชื่อโครงการและคำอธิบายโครงการใช้เพื่อสื่อสารกับผู้ที่เข้าร่วมกิจกรรมที่ชื่อมรมจัดหรือสมาชิกชุมชน โดยจะมีข้อมูลที่ถูกจัดเก็บไว้ ได้แก่ CategoryID หรือรหัสของประเภทนิรภัยที่จะถูกจัดเก็บไว้เป็นสตริง และอีกส่วนคือ Description คำอธิบายของประเภทนิรภัยจะถูกบันทึกเป็นสตริงแต่อนุญาตให้วางว่างข้อมูลในส่วนนี้เอาไว้ได้

3.7.1.5 โหนด Activity: เป็นโหนดที่จะเก็บข้อมูลเกี่ยวกับรายละเอียดของกิจกรรมภายในมหาวิทยาลัย

Property	DataType	Constrain	Definition
ActivityName	STRING	Not NULL	ชื่อกิจกรรม
ActivityID	STRING	Not NULL	รหัสกิจกรรม
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 ตารางเก็บข้อมูลของกิจกรรมภายในมหาวิทยาลัย

จากตารางที่ 3.16 แสดงข้อมูลที่ถูกจัดเก็บไว้ในโหนดนี้คือข้อมูลเกี่ยวกับรายละเอียดของกิจกรรมต่าง ๆ ที่ถูกจัดขึ้นภายในมหาวิทยาลัย มีข้อมูลรายละเอียดต่าง ๆ แจ้งลงออกแบบเป็น ActivityName จะจัดเก็บชื่อกิจกรรมเอาไว้เป็นสตริง โดยไม่สามารถละข้อมูลเอาไว้ได้ ActivityID เก็บ ID ของกิจกรรมเอาไว้เป็นสตริงซึ่งไม่สามารถคละข้อมูลเอาไว้ได้ เช่นกัน Description จะจัดเก็บคำอธิบายของโครง การซึ่งเป็นข้อมูลเดียวในโหนดนี้ที่สามารถเว้นว่างไว้ได้ AcademicYear จะบันทึกปีการศึกษาที่กิจกรรมที่จัดกิจกรรมนั้น ๆ ขึ้นมาเป็นจำนวนเต็ม Semester จะบันทึกภาคการศึกษาที่จัดกิจกรรมนั้น ๆ ขึ้นเป็นจำนวนเต็ม DayTotal จะบันทึกจำนวนวันที่จัดกิจกรรมจะถูกบันทึกเป็นจำนวนเต็ม HourTotal จะบันทึกจำนวนชั่วโมงที่จัดกิจกรรมจะถูกบันทึกเป็นจำนวนเต็ม Organizer จะบันทึกชื่อขององค์กรผู้ที่จัดกิจกรรมไว้เป็นสตริง OpenDate จะบันทึกผู้มุ่งเวลาของวันที่เริ่มกิจกรรมเอาไว้ CloseDate จะบันทึกผู้มุ่งเวลาของวันที่กิจกรรมจบลงเอาไว้ RegisterOpen จะบันทึกวันที่เปิดให้ลงทะเบียนเข้าร่วมกิจกรรมเอาไว้

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

จากตารางที่ 3.17 แสดงข้อมูลที่ถูกจัดเก็บไว้ในโหนดนี้คือข้อมูลเกี่ยวกับรายละเอียดประเภทกิจกรรมต่าง ๆ ที่ถูกจัดขึ้นภายในมหาวิทยาลัย ซึ่งทางผู้จัดทำพยายามจัดแบ่งโดยอาศัยจากทั้งชื่อโครงการและคำอธิบายโครงการเพื่อสื่อสารกับผู้ที่เข้าร่วมกิจกรรม โดยจะมีข้อมูลที่ถูกจัดเก็บไว้ได้แก่ CategoryID หรือรหัสของประเภทกิจกรรมที่จะถูกจัดเก็บไว้เป็นสตริง และอีกวันคือ Description คำอธิบายของประเภทกิจกรรมจะถูกบันทึกเป็นสตริงแต่อ่อนญาติให้เว้นว่างข้อมูลในส่วนนี้เอาไว้ได้

3.7.2 ความสัมพันธ์

ความสัมพันธ์ภายในแบบจำลองข้อมูลแบบกราฟของแอปพลิเคชัน Actiwiz เป็นการสร้างความเชื่อมโยงและการมีปฏิสัมพันธ์ระหว่างโหนดต่าง ๆ ซึ่งมีส่วนช่วยในการนำเสนอเรื่องราวการเข้าร่วมกิจกรรมและชุมชนภายในมหาวิทยาลัยแบบยืดหยุ่นและเชื่อมโยงถึงกัน โดยความสัมพันธ์จะมีอยู่ 2 รูปแบบด้วยกัน ได้แก่ ความสัมพันธ์ที่มีคุณสมบัติที่บันทึกข้อมูลเพิ่มเติมและความสัมพันธ์ที่ไม่มีคุณสมบัติเพิ่มเติม

3.7.2.1 ความสัมพันธ์ที่มีคุณสมบัติ

- User is in DEPARTMENT_OF Department: ความสัมพันธ์ DEPARTMENT_OF แสดงความสัมพันธ์ทางวิชาการของผู้ใช้กับภาควิชาของตน ความสัมพันธ์นี้ช่วยเชื่อมโยงผู้คนจากภาควิชาเดียวกันภายในมหาวิทยาลัย โดยมีคุณสมบัติที่บันทึกไว้ในความสัมพันธ์นี้คือ DateJoined ที่บันทึกข้อมูลวันเวลาที่ผู้ใช้เข้าร่วมภาควิชานั้น ๆ ซึ่งข้อมูลในส่วนนี้ไม่สามารถเปลี่ยนแปลงได้

Property	DataType	Constraint	Definition
DateJoined	DATE	Not NULL	วันที่เข้าเป็นนักศึกษาของภาควิชา

ตารางที่ 3.18: ตารางเก็บข้อมูลนักศึกษาภาควิชา

จากตารางที่ 3.18 แสดงข้อมูลที่ถูกจัดเก็บไว้แสดงความสัมพันธ์ทางวิชาการของผู้ใช้กับภาควิชาของตน ความสัมพันธ์นี้ช่วยเชื่อมโยงผู้คนจากภาควิชาเดียวกันภายในมหาวิทยาลัย โดยมีคุณสมบัติที่บันทึกไว้ในความสัมพันธ์นี้คือ DateJoined ที่บันทึกข้อมูลวันเวลาที่ผู้ใช้เข้าร่วมภาควิชานั้น ๆ ซึ่งข้อมูลในส่วนนี้ไม่สามารถเปลี่ยนแปลงได้

- User is a MEMBER_OF Club: ความสัมพันธ์ MEMBER_OF แสดงความสัมพันธ์ของผู้ใช้ว่าเป็นหนึ่งในสมาชิกของชมรมใดชมรมหนึ่ง

Property	DataType	Constraint	Definition
DateJoined	DATE	Not NULL	วันที่เริ่มเป็นสมาชิกชมรม
Role	STRING	Not NULL	ตำแหน่งในชมรม

ตารางที่ 3.19: ตารางเก็บข้อมูลสมาชิกของชมรม

จากตารางที่ 3.19 แสดงข้อมูลที่ถูกจัดเก็บไว้แสดงความสัมพันธ์ของผู้ใช้ว่าเป็นหนึ่งในสมาชิกของชมรมใดชมรมหนึ่ง ความสัมพันธ์นี้ช่วยเชื่อมโยงผู้คนที่ชมรมเดียวกัน โดยมีคุณสมบัติที่บันทึกไว้ในความสัมพันธ์นี้คือ DATE ที่บันทึกข้อมูลวันเวลาที่ผู้ใช้เข้าร่วมชมรมนั้น ๆ และ Role จะบันทึกตำแหน่งของผู้ใช้ในชมรมนั้นเอาไว้เป็นสตริง โดยทั้งสองข้อมูลนี้ไม่สามารถเปลี่ยนแปลงได้ทั้งคู่

- User PARTICIPATE_IN Activity: ความสัมพันธ์ PARTICIPATES_IN แสดงถึงการเข้าร่วมกิจกรรมของผู้ใช้ในกิจกรรมต่าง ๆ ภายในมหาวิทยาลัย

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

ตารางที่ 3.20: ตารางเก็บข้อมูลสมาชิกเข้าร่วมกิจกรรม

จากตารางที่ 3.20 แสดงข้อมูลที่ถูกจัดเก็บไว้แสดงความสัมพันธ์ของผู้ใช้เข้าร่วมกิจกรรมใดบ้าง ความสัมพันธ์นี้ช่วยเชื่อมโยงผู้คนที่เคยเข้าร่วมกิจกรรมเดียวกัน โดยมีคุณสมบัติที่บันทึกไว้ในความสัมพันธ์นี้คือ DATE ที่บันทึกข้อมูลวันเวลาที่ผู้ใช้เข้าร่วมกิจกรรมนั้น และ Role จะบันทึกบทบาทของผู้ใช้ในกิจกรรมนั้น ๆ เอาไว้เป็นสตริง โดยทั้งสองข้อมูลนี้สามารถเร้นว่างเอาไว้ได้ทั้งคู่

- 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

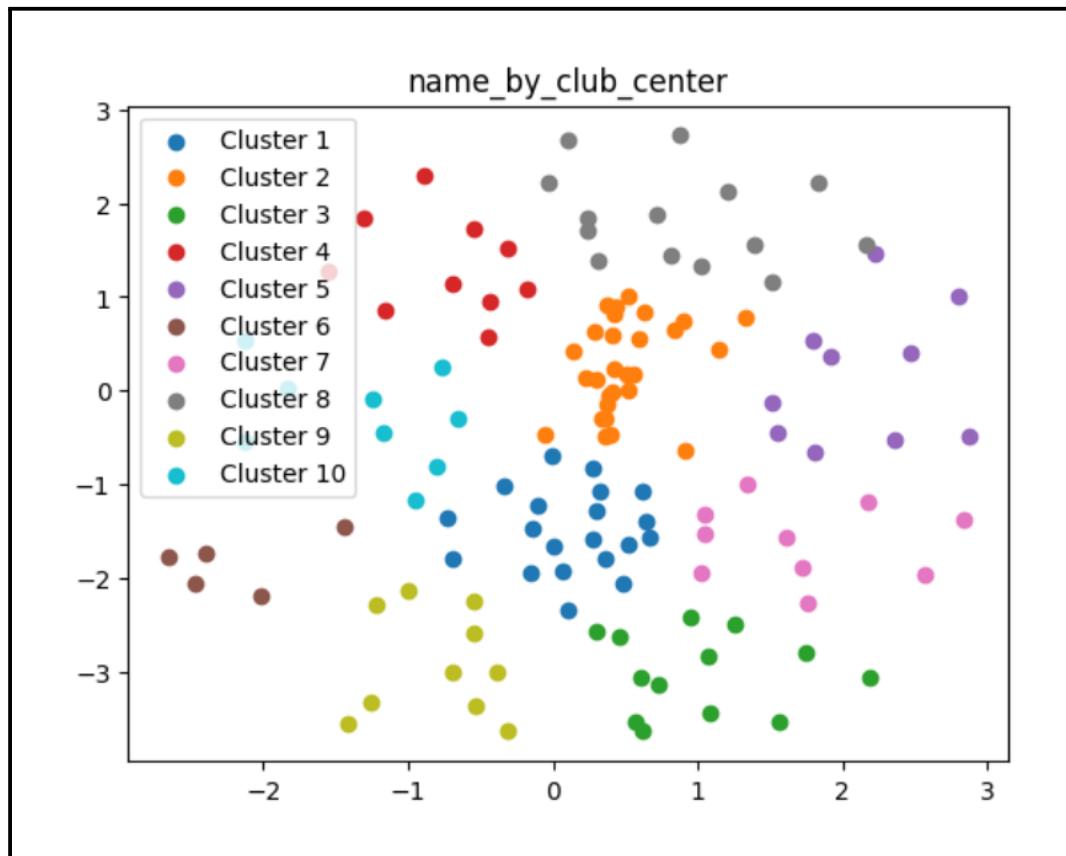
จากตารางที่ 3.21 แสดงข้อมูลที่ถูกจัดเก็บไว้แสดงความสัมพันธ์ระหว่างผู้จัดกิจกรรมกับกิจกรรมต่าง ๆ ที่ถูกจัดขึ้น ความสัมพันธ์นี้บอกถึงความเกี่ยวข้องกันของกิจกรรมและชมรมต่าง ๆ ในมหาวิทยาลัย โดยมีคุณสมบัติที่บันทึกไว้ในความสัมพันธ์นี้คือ OpenDate ที่บันทึกข้อมูลวันเวลาที่เริ่มต้นกิจกรรมนั้น และ CloseDate จะบันทึกวันเวลาที่กิจกรรมนั้นสิ้นสุดลง โดยทั้งสองข้อมูลนี้ไม่สามารถเร้นว่างเอาไว้ได้ทั้งคู่

3.7.2.2 ความสัมพันธ์ที่ไม่มีคุณสมบัติ

- Activity CATEGORIZE_AS Activity Category: ความสัมพันธ์ CATEGORIZE_AS จัดกลุ่มกิจกรรมด้วยการจัดประเภทของกิจกรรม ตามหมวดหมู่เนื้อหา
- Activity RECOMMEND to User: ความสัมพันธ์ RECOMMEND แสดงถึงการแนะนำกิจกรรมที่สร้างโดยระบบของแอปพลิเคชันโดยเกิดจากการประมวลผลของ machine learning ที่ปรับให้เหมาะสมตามผู้ใช้แต่ละคน
- Activity EVALUATE_AVAILABLE to User: ความสัมพันธ์แบบ EVALUATE_AVAILABLE จะเชื่อมโยงผู้ใช้เข้ากับกิจกรรมต่าง ๆ เมื่อกิจกรรมเปิดให้ผู้ใช้เข้าไปประเมินกิจกรรมได้
- Club CATEGORIZE_AS Club Category: ความสัมพันธ์ CATEGORIZE_AS จัดกลุ่มชมรมด้วยการจัดประเภทของชมรม ตามหมวดหมู่เนื้อหา
- Club RECOMMEND to User: ความสัมพันธ์ RECOMMEND แสดงถึงการแนะนำชมรมที่สร้างโดยระบบของแอปพลิเคชันโดยเกิดจากการประมวลผลของ machine learning ที่ปรับให้เหมาะสมตามผู้ใช้แต่ละคน

3.8 การแบ่งประเภทของกิจกรรมและชุมชน

เนื่องจากโครงการ Actiwiz เป็นโครงการที่จัดทำมาเพื่อที่จะอำนวยความสะดวกในการเข้าร่วมกิจกรรมของนักศึกษา ดังนั้นจึงพิจารณา วิธีการแบ่งประเภทของกิจกรรมและชุมชนจากเนื้อหาของกิจกรรมและชุมชน โดยอาศัยหลักการของ WordEmbedding ในการแปลงสิ่งที่เกี่ยวข้องกับกิจกรรมและชุมชนนั้น ไม่ว่าจะเป็นคำอธิบายโครงการ ชื่อชุมชน และเนื้อหาโดยรวมของกิจกรรมที่ชุมชนจัด เป็น Vector และจึงทำการ Clustering เพื่อจัดเนื้อหาที่ลักษณะใกล้เคียงกันไปอยู่ใน Cluster เดียวกัน เพื่อที่จะใช้ Cluster เหล่านี้ในการพิจารณาถึงสิ่งที่ควรจะแนะนำให้ผู้ใช้



รูปที่ 3.16: ตัวอย่าง Cluster

3.9 หลักการพิจารณาสิ่งที่จะแนะนำให้ผู้ใช้

ทางโครงการมีความต้องการที่จะแนะนำของข้อมูลของผู้ใช้จากเนื้อหาที่ผู้ใช้น่าจะให้ความสนใจ แต่ในทางกลับกันก็มีความหลากหลาย เพื่อให้ผู้ใช้มีโอกาสได้เรียนรู้กับประสบการณ์ใหม่ๆ จึงได้มีหลักเกณฑ์นำมารับประทานการแนะนำดังต่อไปนี้

3.9.1 Content-Based Filtering

Content-Based Filtering เป็นการที่พิจารณาความสนใจของผู้ใช้จากเนื้อหาของกิจกรรมและชุมชน โดยจะพิจารณาจากข้อมูลของผู้ใช้ ดังต่อไปนี้

1. ชั้นปีของนักศึกษา
2. คณะที่เข้าศึกษา
3. ภาควิชาที่เข้าศึกษา
4. กิจกรรมที่นักศึกษาเคยเข้าร่วม
5. ประวัติการใช้งานของนักศึกษา

โดยทางโครงการจะนำข้อมูลเหล่านี้เป็นสิ่งที่สันฐานถึงความสนใจของผู้ใช้ และอาศัยหลักการของ Word Embedding ในการแปลงข้อมูลเหล่านี้ออกมายเป็น Vector และอาศัย Vector เหล่านี้ในการหากิจกรรมหรือชุมรมที่มีเนื้อหาใกล้เคียงกับสนใจของผู้ใช้ โดยสิ่งที่ใช้เปรียบเทียบมี 2 วิธีการหลักๆดังต่อไปนี้

1. การเทียบ Distance Metric โดยตรงกับข้อมูลของกิจกรรม ชุมรม และ Cluster ของกิจกรรมและชุมรมที่แบ่งประเภทเอาไว้
2. การอาศัยความน่าจะเป็นที่คำนวนจาก Model Neural Network ต่างๆที่ Finetune ขึ้นมาจากข้อมูลของกิจกรรมและชุมรม

ค่าที่ได้จากการเปรียบเทียบเหล่านี้จะถูกนำมาคำนวณค่าน้ำหนักกับเวลาที่สามารถเข้าร่วมกิจกรรมได้เพื่อที่จะแสดงในหน้า Feed ของผู้ใช้ ซึ่งการพิจารณาสิ่งที่จะแนะนำด้วยวิธีการนี้จะทำให้ผู้ใช้ได้รับกิจกรรมที่น่าจะตรงกับความสนใจของตนเอง

3.9.2 Collaborative Filtering

Collaborative Filtering เป็นการที่พิจารณาถึงสิ่งที่จะแนะนำให้ผู้ใช้โดยพิจารณาจากผู้ใช้ที่มีพฤติกรรมใกล้เคียงกัน โดยอาศัยข้อมูลเบื้องต้นเหล่านี้ในการพิจารณา

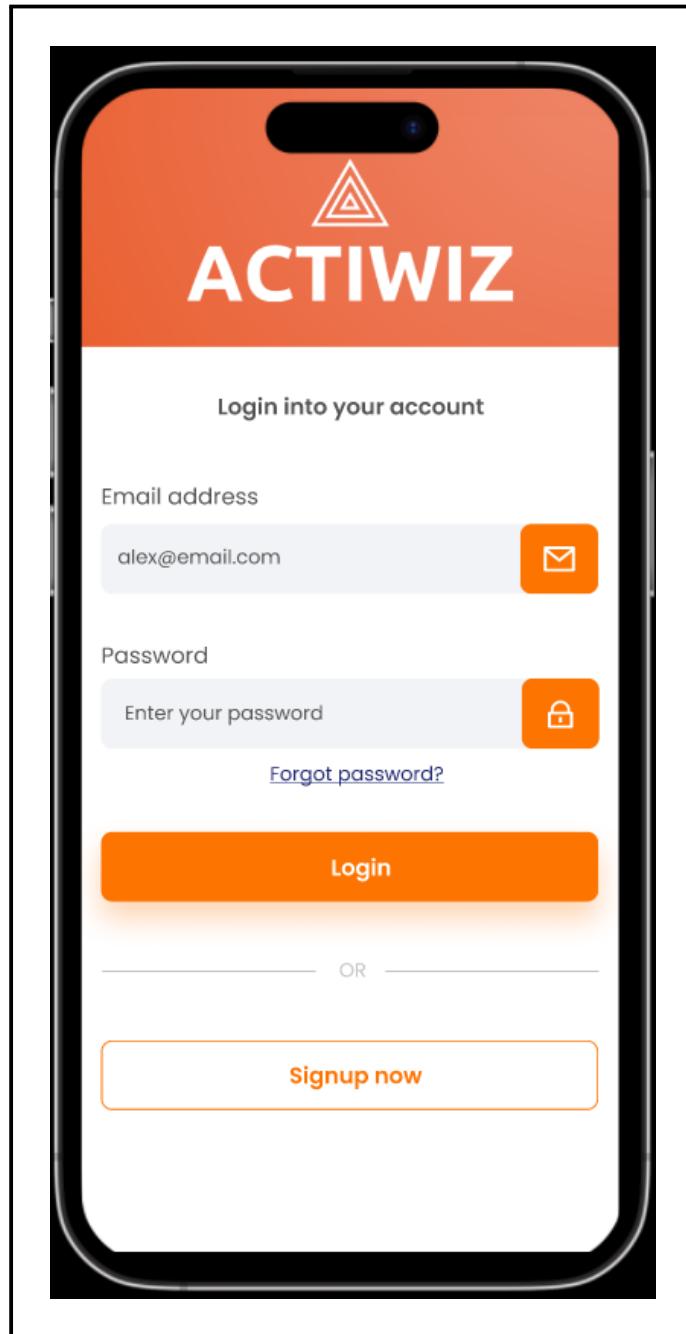
1. ชั้นปีของนักศึกษา
2. คณะที่เข้าศึกษา
3. ภาควิชาที่เข้าศึกษา
4. กิจกรรมที่นักศึกษาเคยเข้าร่วม
5. ประวัติการใช้งานของนักศึกษา

โดยกิจกรรมและชุมรมของผู้ใช้ที่มีลักษณะเหล่านี้ใกล้เคียงกันจะถูกแนะนำให้ผู้ใช้ยิ่งคน ซึ่งการพิจารณาสิ่งที่จะแนะนำด้วยวิธีการนี้จะทำให้ผู้ใช้ได้รับกิจกรรมที่หลากหลายมากขึ้น

3.10 ส่วนติดต่อผู้ใช้ (User Interface)

ส่วนติดต่อผู้ใช้หรือหน้าตาของแอปพลิเคชัน Actiwiz จะมีการออกแบบด้วยหลักการของ F-Shape เพื่อความสะดวกในการอ่าน และติดตามเนื้อหาของแอปพลิเคชัน Actiwiz โดยจะมีไฟเจ้าร์สำคัญ ๆ ดังนี้

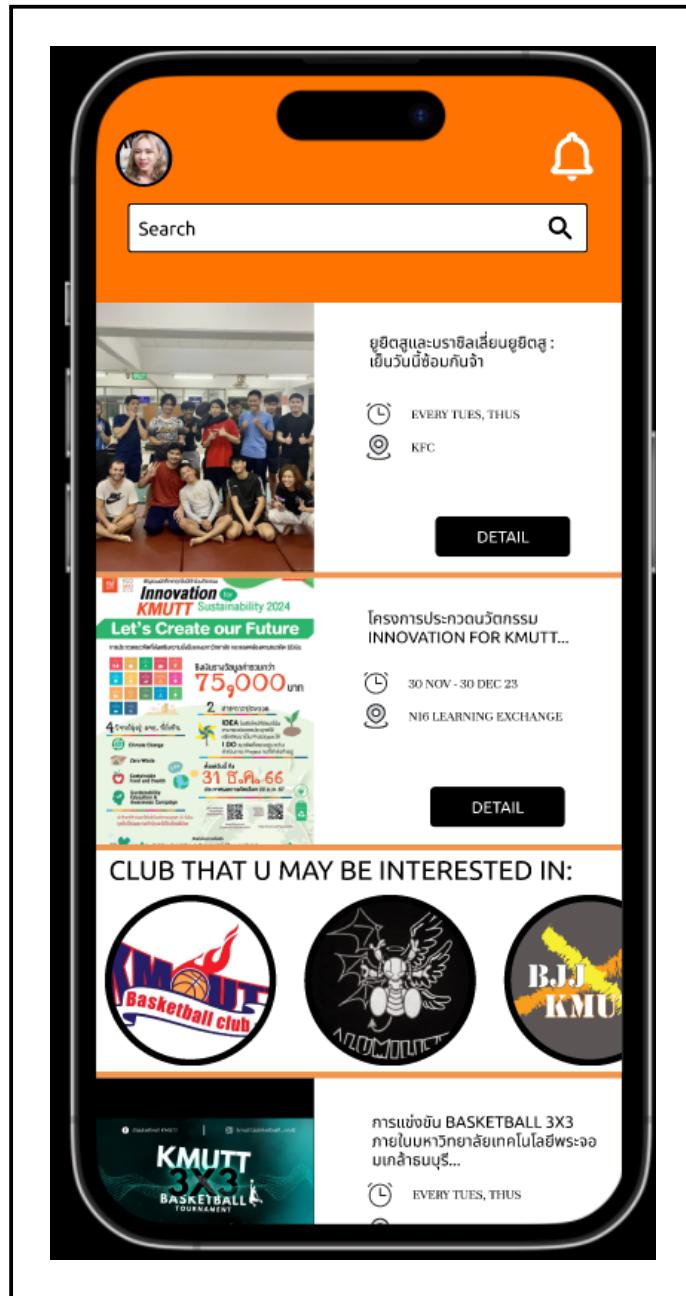
3.10.1 หน้าลงชื่อเข้าใช้ระบบ



รูปที่ 3.17: หน้าลงชื่อเข้าใช้ระบบ

หน้าลงชื่อเข้าใช้ระบบ จะเป็นหน้าแรกที่ผู้ใช้งานเห็นเมื่อเข้าใช้งานแอปพลิเคชัน โดยจะมีช่องให้กรอก Username และ Password และปุ่ม Login เพื่อเข้าสู่ระบบ หากผู้ใช้งานยังไม่มีบัญชีผู้ใช้งานสามารถกดปุ่ม Register เพื่อสมัครสมาชิกใหม่ หรือหากลืมรหัสผ่านสามารถกดปุ่ม Forgot Password เพื่อถูกดึงรหัสผ่าน

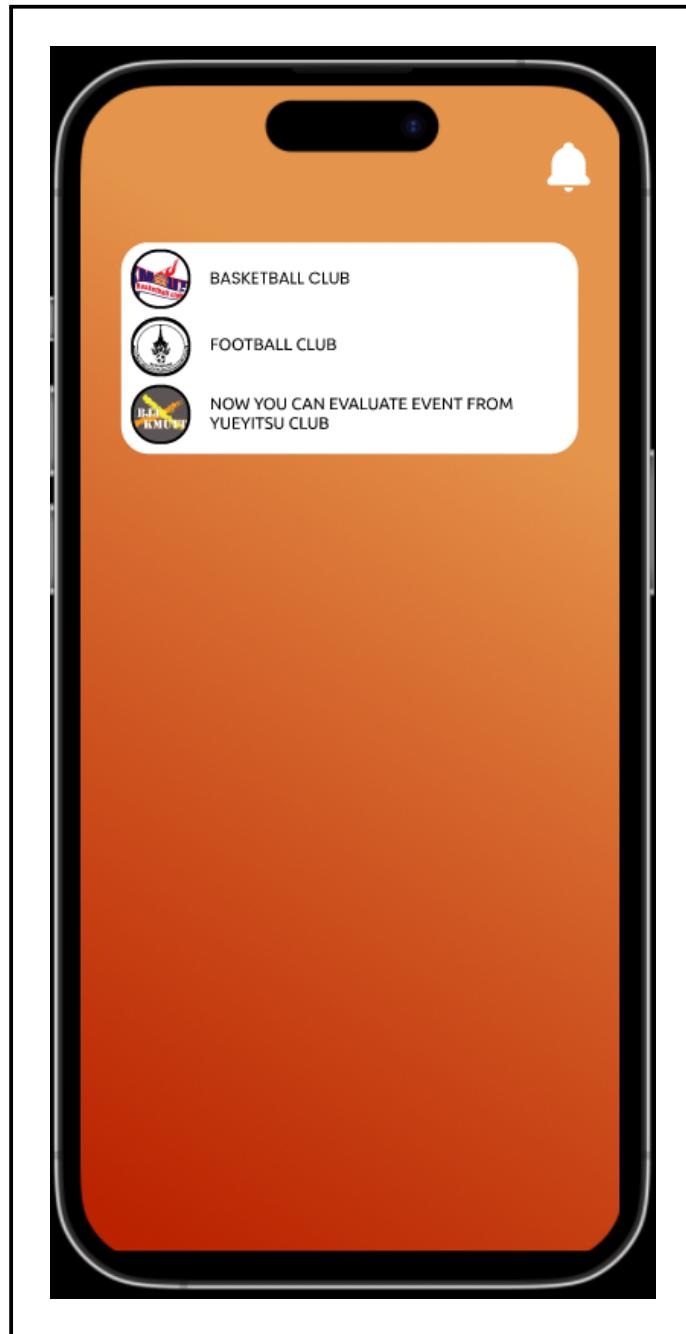
3.10.2 หน้าแนะนำกิจกรรมและชุมชน



รูปที่ 3.18: หน้าแนะนำกิจกรรมและชุมชน

หน้าแนะนำกิจกรรมและชุมชน จะเป็นหน้าที่แสดงกิจกรรมที่เกี่ยวข้องกับผู้ใช้งาน โดยจะแสดงกิจกรรม และชุมชน โดยอ้างอิงจากข้อมูลของผู้ใช้ในการแนะนำ

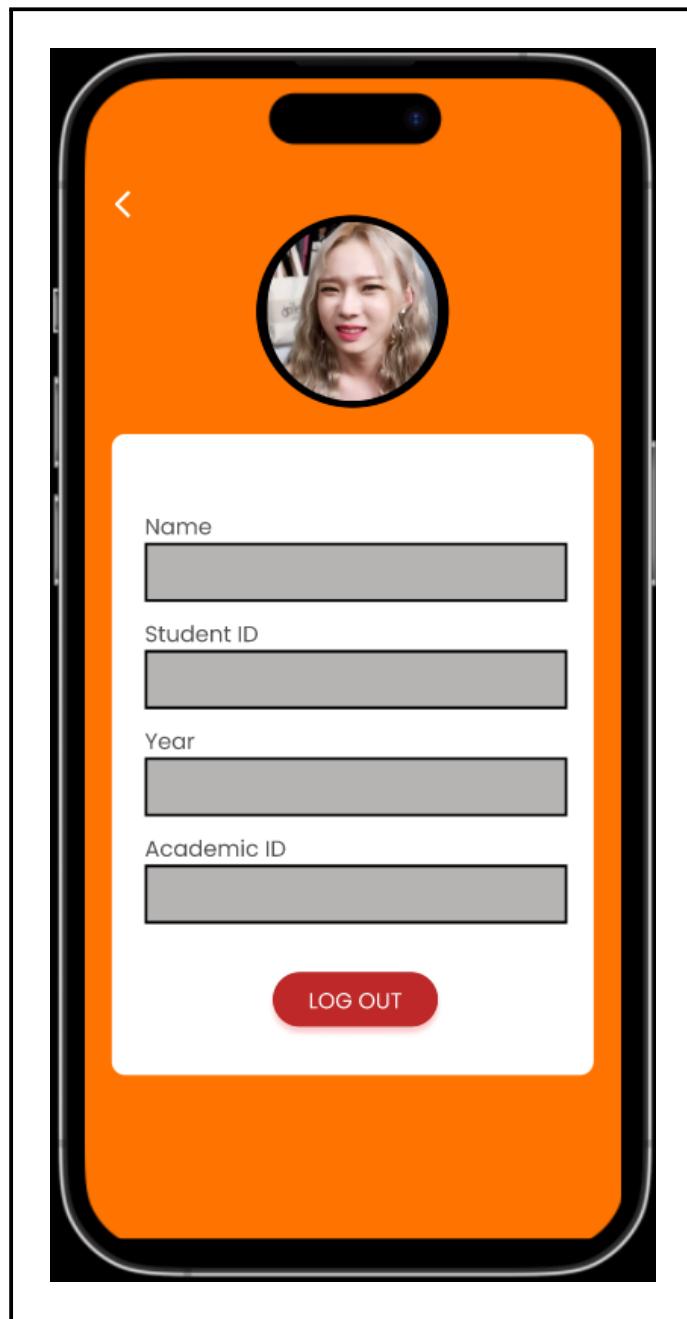
3.10.3 หน้าแสดงการแจ้งเตือน



รูปที่ 3.19: หน้าแสดงการแจ้งเตือน

หน้าแสดงการแจ้งเตือน จะเป็นหน้าที่แสดงการแจ้งเตือนต่างๆ ที่เกี่ยวข้องกับผู้ใช้ โดยจะแสดงการแจ้งเตือนเกี่ยวกับกิจกรรม ชุมชน และ การประเมินกิจกรรม

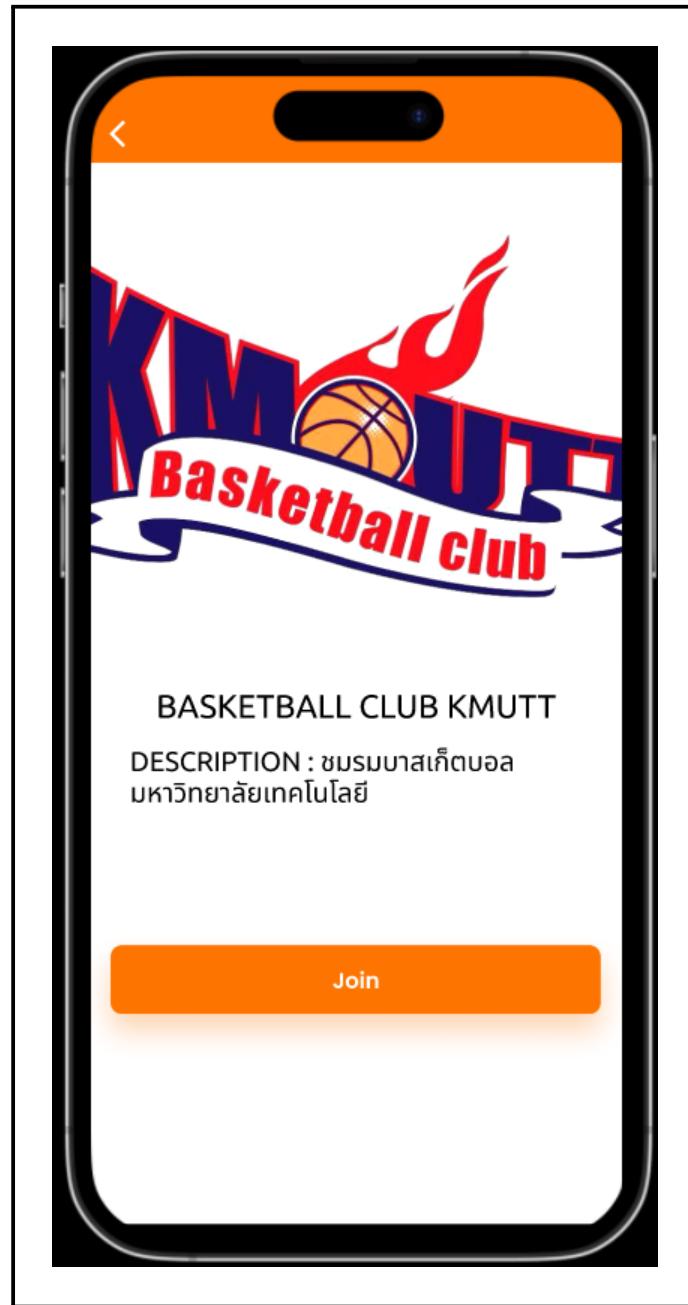
3.10.4 หน้าแสดงข้อมูลผู้ใช้



รูปที่ 3.20: หน้าแสดงข้อมูลผู้ใช้

หน้าแสดงข้อมูลผู้ใช้ จะเป็นหน้าที่แสดงข้อมูลของผู้ใช้ โดยจะแสดงข้อมูลของผู้ใช้ รวมไปถึงการออกจากระบบ

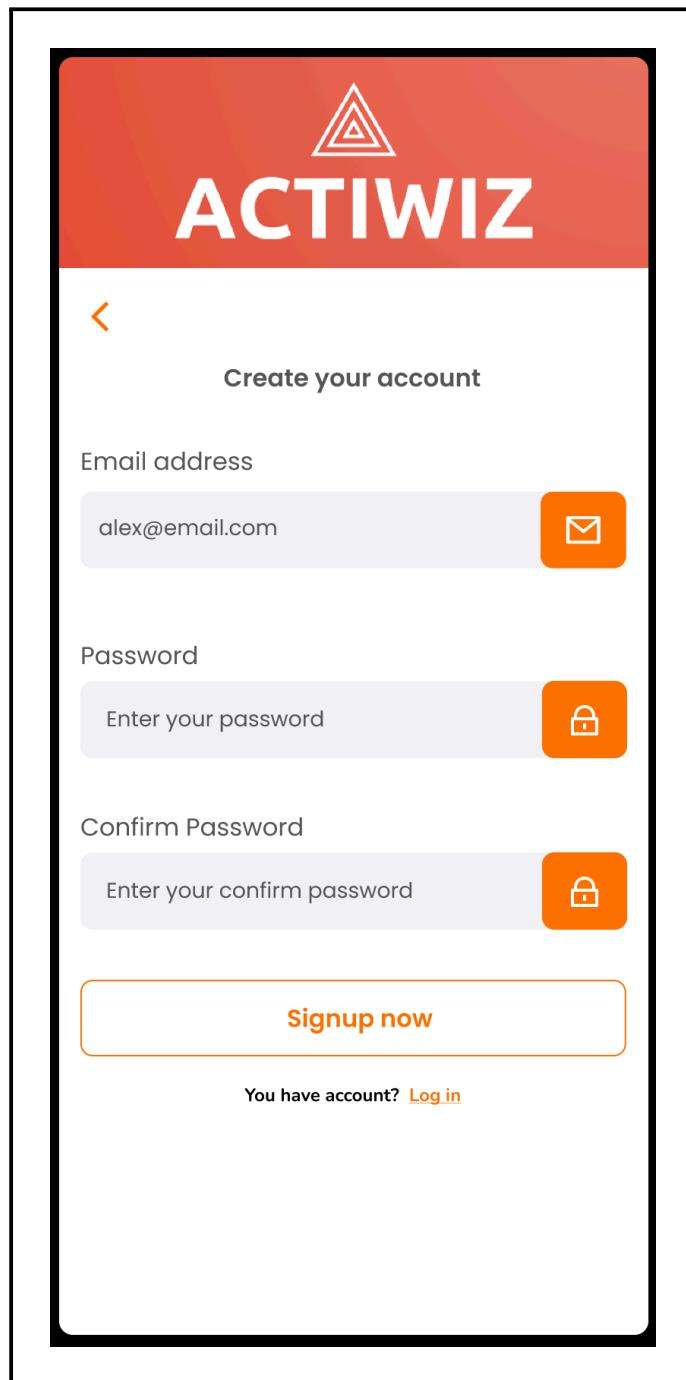
3.10.5 หน้ารายละเอียดชุมชน



รูปที่ 3.21: หน้ารายละเอียดชุมชน

หน้ารายละเอียดชุมชน จะเป็นหน้าที่แสดงข้อมูลของชุมชน โดยจะแสดงข้อมูลของชุมชน รวมไปถึงการเข้าร่วมชุมชน

3.10.6 หน้าลงทะเบียนผู้ใช้



รูปที่ 3.22: หน้าลงทะเบียนผู้ใช้

หน้าลงทะเบียนผู้ใช้ จะเป็นหน้าที่แสดงการสมัครสมาชิกใหม่ โดยจะมีช่องให้กรอก Email, Password และปุ่ม Register เพื่อสมัครสมาชิกใหม่ หรือหากมีบัญชีอยู่แล้วสามารถกดปุ่ม Login เพื่อเข้าสู่ระบบ

3.10.7 หน้ารายละเอียดกิจกรรม



รูปที่ 3.23: หน้ารายละเอียดกิจกรรม

หน้ารายละเอียดกิจกรรม จะเป็นหน้าที่แสดงข้อมูลของกิจกรรม โดยจะแสดงข้อมูลของกิจกรรม รวมไปถึงการเข้าร่วม กิจกรรม

3.10.8 หน้าประเมินกิจกรรม



รูปที่ 3.24: หน้าประเมินกิจกรรม

หน้าประเมินกิจกรรม จะเป็นหน้าที่แสดงการประเมินกิจกรรมผ่านระบบ Sinfo

บทที่ 4 ผลการทดลองและอภิปรายผล

ในบทนี้จะกล่าวถึงผลลัพธ์จากการทดสอบฟังก์ชันหลักของระบบในแอปพลิเคชัน รวมถึงความสามารถในการทำงาน ความหมาย สมใน การใช้งาน การตอบสนองต่อผู้ใช้ และความพึงพอใจของผู้ใช้

4.1 การเปลี่ยนแปลงรายละเอียดของโครงการ

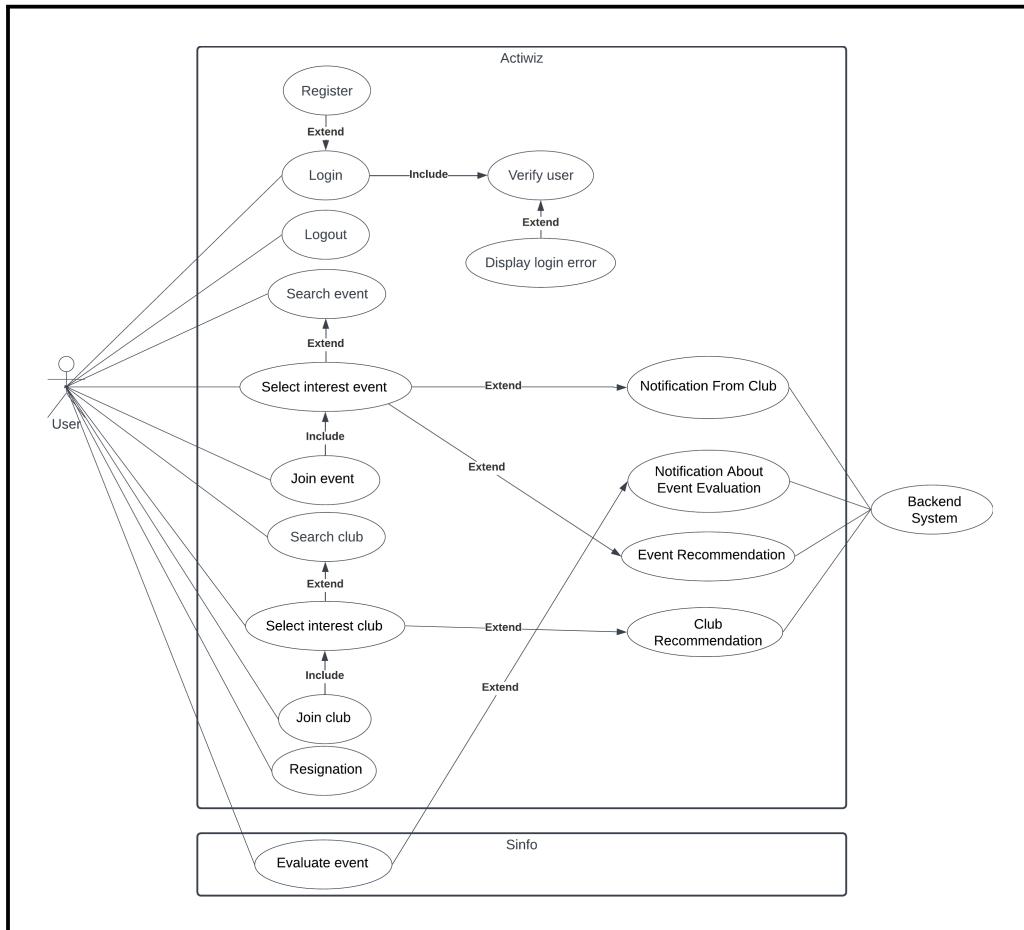
รายละเอียดของโครงการที่เปลี่ยนแปลงไปจากภาคการศึกษาที่ 1 เป็นการแก้ไขเปลี่ยนแปลงรายละเอียดเพื่อให้สอดคล้องกับแนวทาง การแก้ปัญหาที่ได้นำมาปรับใช้เพื่อ แก้ไขปัญหาที่พบเจอระหว่างการพัฒนาโครงการ โดยจะประกอบไปด้วยส่วนของการออกแบบแอปพลิเคชัน ดัง ๆ ตามหัวข้อดังต่อไปนี้

1. รายการคุณลักษณะ (Feature List)
2. แผนภาพที่ใช้แสดงงบประมาณพื้นที่ที่ต้องการของระบบ (Use Case Diagram)
3. แผนภาพที่ใช้แสดงการทำงานของระบบ (Sequence Diagram)
4. ส่วนติดต่อผู้ใช้ (User Interface)

4.1.1 รายการคุณลักษณะ (Feature List)

1. Login: ด้วยวัตถุประสงค์เพื่อเพิ่มประสิทธิภาพและความสะดวกสบายแก่ผู้ใช้แอปพลิเคชัน ทางผู้จัดทำได้ตัดสินใจปรับเปลี่ยนกระบวนการ การเข้าสู่ระบบ (Login) และการพิสูจน์ยืนยันตัวตน (Authenticate) ของผู้ใช้ ให้เป็นการดำเนินการผ่านระบบ Microsoft Entra ID อย่างไรก็ตาม ผู้ใช้ยังจำเป็นต้องให้ข้อมูลบางอย่างและลงทะเบียนกับทางแอปพลิเคชันเพื่อให้สามารถใช้งานฟังก์ชันต่าง ๆ ของแอปพลิเคชันได้อยู่ แต่ผู้ใช้ไม่จำเป็นต้องจำ username หรือรหัสผ่านใหม่เพื่อเข้าใช้งานระบบ เพราะผู้ใช้สามารถใช้บัญชี Kmutt Internet Account ซึ่งเป็นบัญชีที่ผู้ใช้ได้ลงทะเบียนไว้กับมหาวิทยาลัยเป็นที่เรียบร้อยแล้วในการเข้าสู่ระบบและพิสูจน์ยืนยันตัวตน
2. Event evaluation: ทางทีมของเราระบุต้องความเห็นและทำการลง Feature นี้ออกจากระบบของเรา เนื่องจาก戢ึงเห็นว่าการพัฒนา Feature นี้ให้สามารถทำได้ในแอปพลิเคชันของเราว่าจะต้องใช้เวลานานและมีความซับซ้อนในการพัฒนา อีกทั้งยังเป็น Feature ที่อยู่นอกเหนือขอบเขตของโครงการที่เราได้กำหนดไว้ ดังนั้นจึงไม่มีความจำเป็นที่จะต้องมี Feature นี้ในระบบของเรา ทั้งนี้แอปพลิเคชันของเรายังมีการรองรับเรื่องของการแจ้งเตือนการประเมินกิจกรรมอยู่ โดยผู้ใช้สามารถทำการประเมินกิจกรรมที่เข้าร่วมได้โดยตรงผ่านระบบ Sinfo ซึ่งเป็นระบบที่มหาวิทยาลัยของเราระบุต้องใช้ในการประเมินกิจกรรมต่าง ๆ ของนักศึกษา

4.1.2 แผนภาพที่ใช้แสดงปฏิสัมพันธ์ระหว่างระบบงานและสิ่งที่อยู่นอกระบบงาน (Use Case Diagram)



รูปที่ 4.1: Use Case Diagram แบบใหม่

ทางผู้จัดทำได้มีการปรับปรุงแผนภาพที่ใช้แสดงปฏิสัมพันธ์ระหว่างระบบงานและสิ่งที่อยู่นอกระบบงาน (Use Case Diagram) จากรูปที่ 3.2 มาเป็นรูปที่ 4.1 แทน เนื่องด้วยเหตุผลสำคัญตามการเปลี่ยนแปลงที่ได้กล่าวไปในหัวข้อที่ 4.1.1 เนื่องจากเราได้ทำการลบ Feature Event evaluation ออกจากระบบของเรา และให้ผู้ใช้งานเข้าประเมินกิจกรรมผ่านระบบของ Sinfo แทน ทำให้ห้องย้าย Use case Event evaluation ไปยัง Sinfo System แทน

Use Case Narratives

การสมัครเข้าใช้งาน

Actor	User
Goal	ลงทะเบียนสร้าง account
Pre-conditions	Login ผ่านระบบของ Microsoft สำเร็จ
Main success scenario	1. User เข้าสู่หน้าลงทะเบียนเพื่อขอข้อมูลเพิ่มเติม 2. User กรอกแบบฟอร์ม 3. User กดยืนยันการสร้าง Account

ตารางที่ 4.1 การสมัครเข้าใช้งานแบบใหม่

เนื่องจากมีการเปลี่ยนแปลงรูปแบบการให้เข้าสู่ระบบเป็นการใช้งานระบบของ Microsoft แทน ส่งผลให้รายละเอียดของตารางที่ 3.1 ซึ่งเป็นตารางสำหรับการอธิบายรายละเอียดของการสมัครเข้าใช้งานของระบบเปลี่ยนแปลงไปเล็กน้อย โดยในขั้นตอนแรกจะเริ่มต้นหลังจากผู้ใช้ Login ผ่านระบบของ Microsoft สำเร็จแทนที่จะเป็นการกดปุ่มเพื่อลงทะเบียน หลังจากนั้นจึงทำการขอข้อมูลเพิ่มเติมเพื่อลบทะเบียน account กับแอปพลิเคชัน ดังปรากฏในรูปแบบตารางที่ 4.1

การเข้าสู่ระบบ

Actor	User
Goal	เข้าสู่ระบบและใช้งานฟังก์ชันต่าง ๆ ได้
Pre-conditions	-
Main success scenario	1.User กดปุ่มเพื่อไปยังหน้าเข้าสู่ระบบของ Microsoft 2.User กรอกอีเมลและรหัสผ่าน 3.User เข้าสู่ homepage ของแอปพลิเคชัน 4.ระบบแสดงฟังก์ชันที่ใช้งานได้ทั้งหมด 5.User ใช้งานฟังก์ชันต่าง ๆ ในแอปพลิเคชัน

ตารางที่ 4.2 การเข้าสู่ระบบแบบใหม่

เนื่องจากมีการเปลี่ยนแปลงรูปแบบการให้เข้าสู่ระบบส่งผลให้รายละเอียดของตารางที่ [3.2](#) ซึ่งเป็นตารางสำหรับการอธิบายรายละเอียดของการเข้าสู่ระบบมีการเปลี่ยนแปลงไป โดยจะมีการเพิ่มขั้นตอนของการเข้าสู่ระบบ (Login) ผ่านระบบของ Microsoft เป็นอันดับแรก ก่อนการกรอกรายละเอียดของผู้ใช้เพื่อทำการเข้าสู่ระบบ ดังปรากฏในรูปแบบตามตารางที่ [4.2](#)

การประเมินกิจกรรม

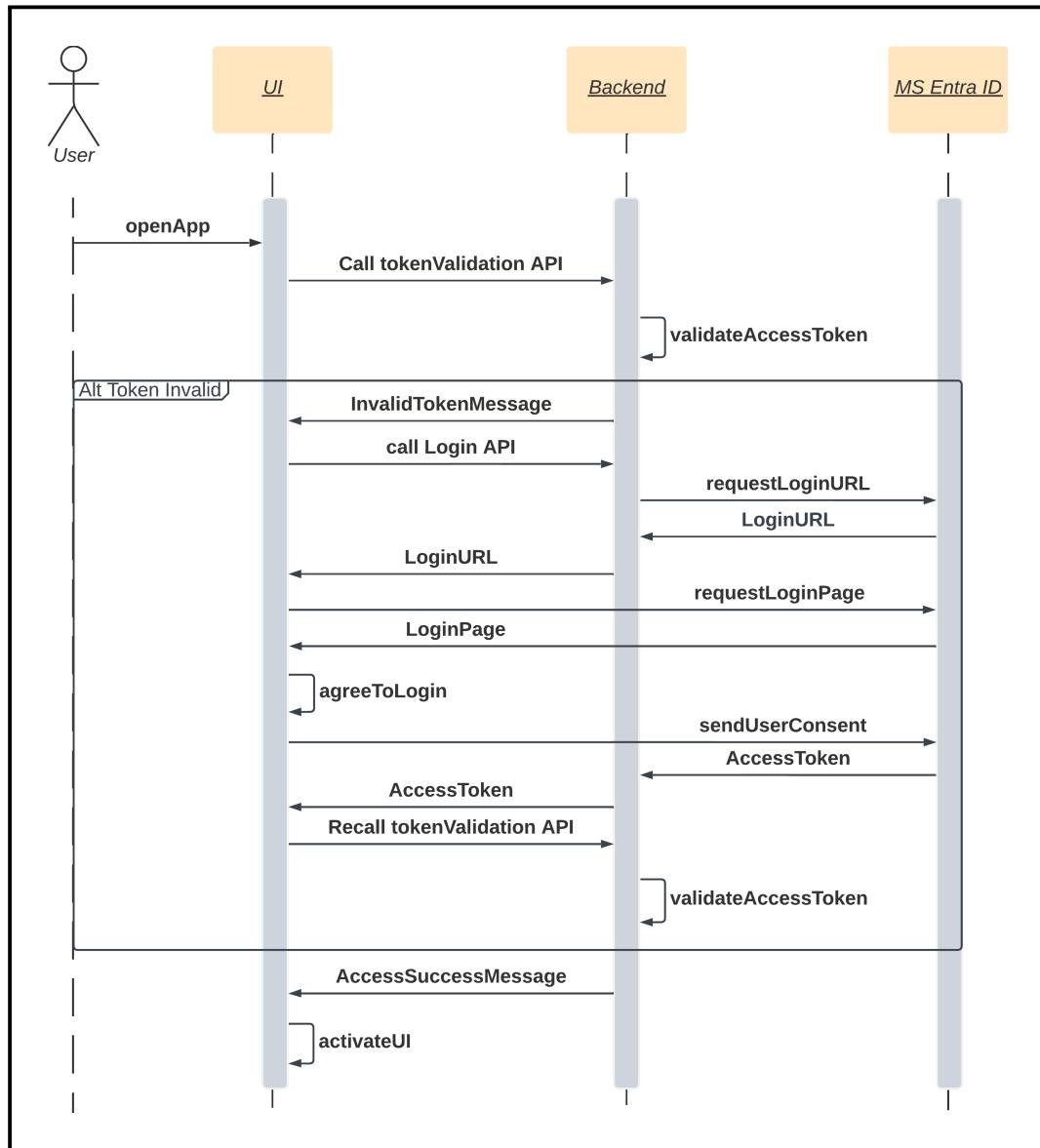
Actor	User
Goal	ประเมินกิจกรรม
Pre-conditions	- User ต้องเคยเข้าร่วมกิจกรรม
Main success scenario	1.User ได้รับแจ้งเตือนการประเมินกิจกรรม 2.User เข้าประเมินกิจกรรมในระบบ Sinfo

ตารางที่ 4.3 การประเมินกิจกรรม

เนื่องจากมีการเปลี่ยนแปลงรูปแบบการประเมินกิจกรรมเป็นการเข้าใช้งานระบบของ Sinfo ทำให้ตารางที่ [3.7](#) ซึ่งเป็นตารางสำหรับการอธิบายรายละเอียดของการประเมินกิจกรรมมีการเปลี่ยนแปลงไป โดยจะมีเงื่อนไขว่าผู้ใช้ต้องเคยเข้าร่วมกิจกรรมก่อนที่จะสามารถทำการประเมินกิจกรรมได้ และลบขั้นตอนการประเมินกิจกรรมในแอปพลิเคชันออก และให้ผู้ใช้งานเข้าประเมินกิจกรรมผ่านระบบของ Sinfo แทน ดังปรากฏในรูปแบบตามตารางที่ [4.3](#)

4.1.3 แผนภาพที่ใช้แสดงการทำงานของระบบ (Sequence Diagram)

4.1.3.1 การเข้าสู่ระบบ

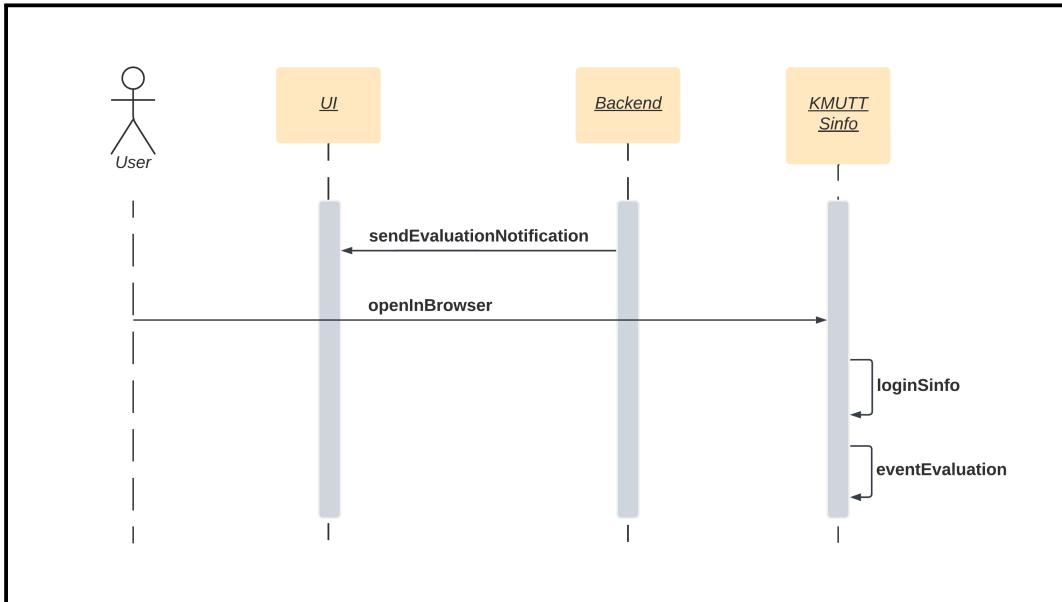


รูปที่ 4.2: แผนภาพการเข้าสู่ระบบแบบใหม่

ทางผู้จัดทำได้มีการปรับปรุงแผนภาพที่ใช้แสดงการทำงานของระบบ (Sequence Diagram) สำหรับกระบวนการเข้าสู่ระบบ (Login) จากรูปที่ 3.4 เป็นรูปที่ 4.2 เนื่องด้วยเหตุผลสำคัญตามการเปลี่ยนแปลงที่ได้กล่าวไปในหัวข้อที่ 4.1.1 ทำให้กระบวนการเข้าสู่ระบบ (Login) และการพิสูจน์ยืนยันตัวตน (Authenticate) ของผู้ใช้ จะเป็นการดำเนินการผ่านระบบ Microsoft Entra ID โดยแผนภาพ Sequence Diagram สำหรับกระบวนการเข้าสู่ระบบ (Login) ที่ได้ทำการแก้ไขไปหรือรูปที่ 4.2 นั้นแสดงลำดับการทำงานระหว่างองค์ประกอบต่าง ๆ ในระบบระหว่างกระบวนการเข้าสู่ระบบ (Login) และการพิสูจน์ยืนยันตัวตน (Authenticate) ของผู้ใช้ ซึ่งขั้นตอนการทำงานมีดังนี้:

1. ผู้ใช้เริ่มกระบวนการล็อกอินโดยเรียกใช้งานเบราว์เซอร์และติดต่อ Backend System ผ่าน API สำหรับการตรวจสอบ Access Token
2. แอปพลิเคชันส่ง Request ที่ประกอบด้วย Access Token ไปยัง Backend System ผ่านการเรียกใช้งาน API สำหรับการตรวจสอบ Access Token
3. Backend System ดำเนินการตรวจสอบความถูกต้องของ Access Token
4. ดำเนินการตามสถานะจากการตรวจสอบ Access Token
 - (a) หาก Access Token ไม่ถูกต้อง จะทำการขอ Access Token ใหม่จาก Microsoft Entra ID โดยจะเป็นขั้นตอนต่อไป
 - (b) หาก Access Token ถูกต้องจะดำเนินการตามเข้าใช้งานแอปพลิเคชันตามปกติตามขั้นตอนที่ 18 เป็นต้นไป
5. Backend System ส่ง Response กลับไปยังแอปพลิเคชันเพื่อแจ้งให้แอปพลิเคชันทราบว่า Access Token ไม่ถูกต้อง
6. แอปพลิเคชันส่ง Request ไปยัง Backend System เพื่อขอ Login URL ผ่านการเรียกใช้งาน API สำหรับการขอ Login URL
7. Backend System ส่ง Request ไปยัง Microsoft Entra ID เพื่อขอ Login URL
8. Microsoft Entra ID ส่ง Login URL กลับไปยัง Backend System
9. Backend System ส่ง Login URL กลับไปยังแอปพลิเคชัน
10. แอปพลิเคชันเรียกใช้ Login URL เพื่อร้องขอหน้าเข้าสู่ระบบของ Microsoft
11. Microsoft Entra ID ส่งหน้าเข้าสู่ระบบกลับไปยังแอปพลิเคชัน
12. ผู้ใช้กรอกข้อมูลเข้าสู่ระบบในหน้าเข้าสู่ระบบของ Microsoft
13. ผู้ใช้กดปุ่มเข้าสู่ระบบ
14. Microsoft Entra ID สร้าง Access Token ใหม่และส่งกลับไปยัง Backend System
15. Backend System ส่ง Access Token ไปยังแอปพลิเคชันเพื่อให้แอปพลิเคชันเรียกใช้งาน API ต่าง ๆ ในระบบ
16. แอปพลิเคชันส่ง Request ที่ประกอบด้วย Access Token ไปยัง Backend System ผ่านการเรียกใช้งาน API สำหรับการตรวจสอบ Access Token อีกครั้ง
17. Backend System ดำเนินการตรวจสอบความถูกต้องของ Access Token
18. Backend System ส่ง Response กลับไปยังแอปพลิเคชันเพื่อแจ้งให้แอปพลิเคชันทราบว่า Access Token ถูกต้อง
19. แอปพลิเคชันเริ่มใช้งานฟังก์ชันต่าง ๆ ในระบบได้

4.1.3.2 การแจ้งเตือนการประเมินกิจกรรม



รูปที่ 4.3: แผนภาพการแจ้งเตือนการประเมินกิจกรรมแบบใหม่

ทางผู้จัดทำได้มีการปรับปรุงแผนภาพที่ใช้แสดงการทำงานของระบบ (Sequence Diagram) สำหรับกระบวนการการแจ้งเตือนการประเมินกิจกรรมจากรูปที่ 3.4 เป็นรูปที่ 4.3 เนื่องด้วยเหตุผลสำคัญตามการเปลี่ยนแปลงที่ได้กล่าวไปในหัวข้อที่ 4.1.1 ทำให้ผู้ใช้จะต้องทำการประเมินกิจกรรมจากระบบของ Sinfo แทน ซึ่งขั้นตอนการทำงานมีดังนี้:

1. Backend System จะตรวจสอบกำหนดการหรือเงื่อนไขที่ต้องแจ้งเตือนการประเมินกิจกรรมให้กับผู้ใช้ เช่น หลังจากกิจกรรมสิ้นสุดลงภายในระยะเวลาที่กำหนด
2. เมื่อถึงกำหนดการหรือเงื่อนไขที่ต้องแจ้งเตือน Backend System จะส่งคำเตือนการประเมินกิจกรรมไปยัง User Interface (UI) ของผู้ใช้
3. UI จะแสดงข้อความแจ้งเตือนการประเมินกิจกรรมให้ผู้ใช้ทราบ
4. ผู้ใช้สามารถเข้าสู่ระบบของ Sinfo เพื่อให้ผู้ใช้กรอกข้อมูลและประเมินกิจกรรมได้

4.1.4 ส่วนติดต่อผู้ใช้ (User Interface)

1. Notification Page: เนื่องจากในขั้นตอนพัฒนาได้มีการเปลี่ยนรูปแบบ Notification เป็นในลักษณะของ Push Notification ทำให้ User Interface รูปที่ 3.19 ไม่ได้ถูกใช้งานอีกต่อไป
2. Evaluate Page: เนื่องจากมีการเปลี่ยนแปลงรูปแบบการประเมินกิจกรรมเป็นการเข้าใช้งานระบบของ Sinfo โดยตรง ทำให้ User Interface ของแอปพลิเคชันที่เกี่ยวข้องกับการประเมินกิจกรรมได้ถูกลบออก และให้ผู้ใช้งานเข้าไปประเมินกิจกรรมผ่านระบบของ Sinfo แทน ดังนั้น User Interface รูปที่ 3.24 ไม่ได้ถูกใช้งานอีกต่อไป

4.2 การพัฒนาของโครงการ

4.2.1 ระบบแนะนำกิจกรรมและชุมชน

4.2.1.1 การแบ่งประเภทของกิจกรรมและชุมชน

นำข้อมูลของชื่อและคำอธิบายโครงการของกิจกรรมและชุมชนต่างๆที่ได้มาจากการนักศึกษา มจร. มาทำการลับสิ่งที่ไม่เกี่ยวข้องกับเนื้อหา เช่น เครื่องหมายพิเศษต่างๆ ตัวอักษรหรือสระที่เกินมา หรือคำที่เขียนผิดต่างๆ จากนั้นจึงทำการสำรวจความเหมาะสมในการแปลงเนื้อหาเหล่านี้เป็น Vector ที่แสดงถึงเนื้อหาของกิจกรรมนั้นๆ แล้วจึงนำ Vector เหล่านี้มาทำการแบ่ง Cluster เพื่อจัดกลุ่มให้กิจกรรมและชุมชนที่มีเนื้อหาใกล้เคียงกันไปอยู่ใน Cluster เดียวกัน และใช้ cluster เหล่านี้เป็น Tag เพื่อแยกประเภทกิจกรรมและชุมชนเพิ่มเติมจากการแบ่งประเภทกิจกรรมที่มีอยู่เดิม

4.2.1.2 หลักการพิจารณาสิ่งที่จะแนะนำให้ผู้ใช้งาน

เนื่องจากข้อจำกัดของทางโครงการที่ไม่มีข้อมูลการใช้งานของผู้ใช้งานอยู่เลย ทางผู้จัดทำจึงสันนิฐานถึงความสนใจของผู้ใช้งานจากเนื้อหาของภาควิชา และคณะที่เข้าเรียนเป็นข้อมูลพื้นฐาน และอาศัยประวัติการใช้งานของผู้ใช้เป็นสิ่งที่เพิ่มข้อมูลในการพิจารณาความสนใจของผู้ใช้งาน ซึ่งในการหาความเชื่อมโยงระหว่างเนื้อหาเหล่านี้จำเป็นต้องใช้ Machine Learning ในการพิจารณาถึงสิ่งที่ควรแนะนำให้ผู้ใช้งาน ซึ่งจะมีการใช้งานต่างกันไปตามหลักการพิจารณาการแนะนำดังต่อไปนี้

- Content-Based Filtering เป็นการที่พิจารณาความสนใจของผู้ใช้งานจากเนื้อหาของกิจกรรมและชุมชน โดยจะพิจารณาจากข้อมูลของผู้ใช้งาน ดังต่อไปนี้

- ขั้นปีของนักศึกษา
- คณะที่เข้าศึกษา
- ภาควิชาที่เข้าศึกษา
- กิจกรรมที่นักศึกษาเคยเข้าร่วม
- ประวัติการใช้งานของนักศึกษา

โดยทางโครงการจะนำข้อมูลเหล่านี้เป็นสิ่งที่สันนิฐานถึงความสนใจของผู้ใช้งาน และอาศัยหลักการของ WordEmbedding ในการแปลงข้อมูลเหล่านี้เป็น Vector และอาศัย Vector เหล่านี้ในการหากิจกรรมหรือชุมชนที่มีเนื้อหาใกล้เคียงกับสนใจของผู้ใช้งาน โดยสิ่งที่ใช้เปรียบเทียบมี 2 วิธีการหลักๆดังต่อไปนี้

- การเทียบ Distance Metric โดยตรงกับข้อมูลของกิจกรรม ชุมชน และ Cluster ของกิจกรรมและชุมชนที่แบ่งประเภทเอาไว้ โดยการนำข้อมูลของผู้ใช้งานไปเทียบกับ Vector ของข้อมูลของกิจกรรม ชุมชน และจุดศูนย์กลางมวลของ Cluster ต่างๆ โดยอาศัยหลักการของ K-Nearest Neighbors ตามค่าวัดระยะห่างต่างๆ เช่น Euclidian Distance, Cosine Similarity หรือ Dot product เป็นต้น
- การอาศัยความน่าจะเป็นที่คำนวนจาก Model Neural Network ต่างๆที่ Finetune ขึ้นมาจากข้อมูลของกิจกรรมและชุมชนซึ่งทำความสะอาดดิสตัฟฟ์ไม่เกี่ยวข้องกับเนื้อหาใกล้กันหน้าแล้วตัดคำโดยอาศัย Tokenizer ของตัว Model ที่นำมา Finetune

ค่าที่ได้จากการเปรียบเทียบทะล่า�ี่จะถูกนำมาคำนวนค่าน้ำหนักกับเวลาที่สามารถเข้าร่วมกิจกรรมได้แล้วเก็บเอาไว้ในความสัมพันธ์ระหว่างหนอนเพื่อที่เมื่อวิเคราะห์การเปิดใช้งานแอปพลิเคชันความสัมพันธ์เหล่านี้จะถูกเรียกมาเพื่อพิจารณาว่ากิจกรรมใดที่จะถูกแสดงในหน้า Feed ของผู้ใช้งาน ซึ่งการพิจารณาสิ่งที่จะแนะนำด้วยวิธีการนี้จะทำให้ผู้ใช้งานได้รับกิจกรรมที่น่าจะตรงกับความสนใจของตนเอง

- Collaborative Filtering เป็นการที่พิจารณาถึงสิ่งที่จะแนะนำให้ผู้ใช้งานโดยพิจารณาจากผู้ใช้งานที่มีพฤติกรรมใกล้เคียงกัน โดยอาศัยข้อมูลเบื้องต้นเหล่านี้ในการพิจารณา

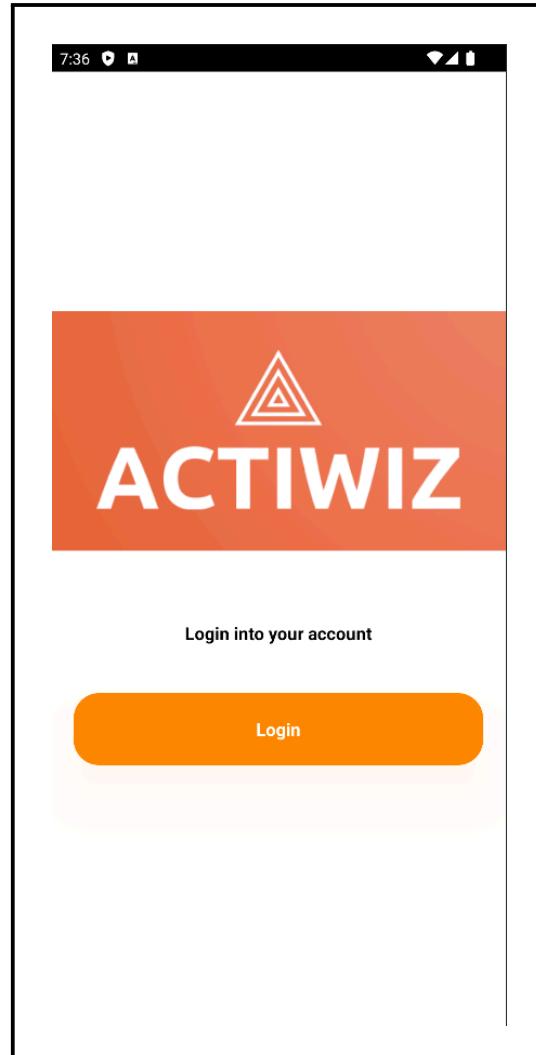
- ขั้นปีของนักศึกษา
- คณะที่เข้าศึกษา
- ภาควิชาที่เข้าศึกษา
- กิจกรรมที่นักศึกษาเคยเข้าร่วม
- ประวัติการใช้งานของนักศึกษา

โดยกิจกรรมและชุมชนของผู้ใช้งานที่มีลักษณะเหล่านี้ใกล้เคียงกันจะถูกแนะนำให้ผู้ใช้งานอีกคน ซึ่งการพิจารณาสิ่งที่จะแนะนำด้วยวิธีการนี้จะทำให้ผู้ใช้งานได้รับกิจกรรมที่หลากหลายมากขึ้น ซึ่งจะพิจารณาข้อมูลเหล่านี้จากความสัมพันธ์ที่เก็บเอาไว้ในฐานข้อมูล

4.2.2 การพัฒนาแอปพลิเคชัน

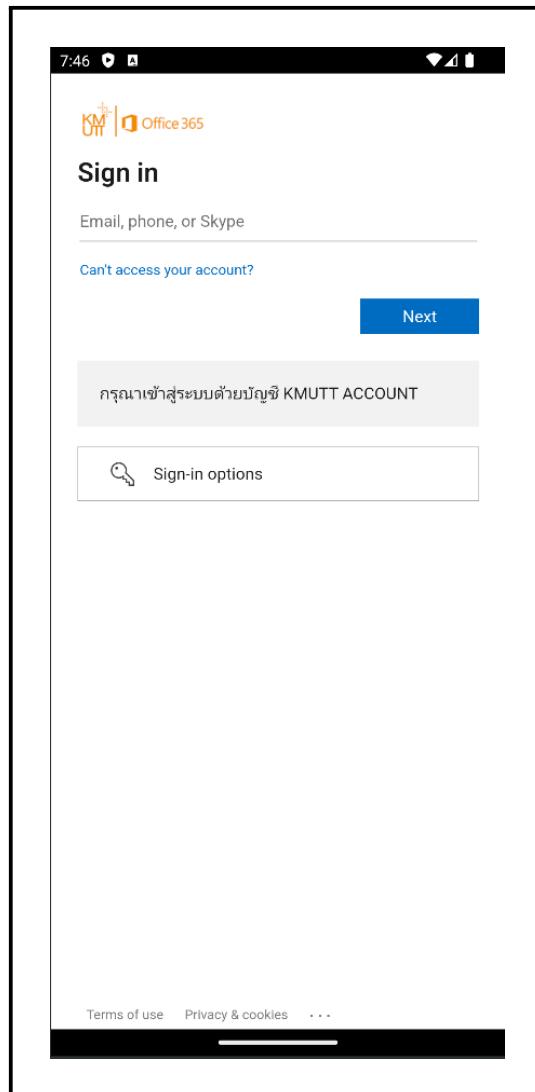
4.2.2.1 ลงชื่อเพื่อเข้าใช้งานแอปพลิเคชัน

กรณีที่ยังไม่เคยลงชื่อเข้าใช้ในแอปพลิเคชันผู้ใช้สามารถเริ่มกระบวนการล็อกอินโดยการกดปุ่ม Login ที่หน้าแรกของ Application ตามรูปที่ 4.4



รูปที่ 4.4: หน้าแรกของแอปพลิเคชัน

หลังจากผู้ใช้กดปุ่มแล้วแอปพลิเคชันจะส่ง Request ไปยัง Backend System เพื่อขอ Login URL และใช้ Login URL ตั้งกล่าวผ่าน Webview ของแอปพลิเคชันเพื่อร้องขอหน้าเข้าสู่ระบบของ Microsoft ตาม รูปที่ 4.5



รูปที่ 4.5: หน้าระบบลงชื่อเข้าใช้ของ Microsoft

หลังจากผู้ใช้ทำการลงชื่อเข้าใช้เรียบร้อย ระบบของ Microsoft จะส่ง Token กลับมา 3 token ได้แก่

1. API Token สำหรับการเข้าใช้งานระบบของแอปพลิเคชัน Actiwiz
2. API Token สำหรับการขอข้อมูลจาก Microsoft Graph Service และ
3. Refresh Token สำหรับการขอ API token 2 token ข้างต้นใหม่

4.2.2.2 กรณีที่เคลื่อนข้อเข้าใช้ในแอปพลิเคชัน

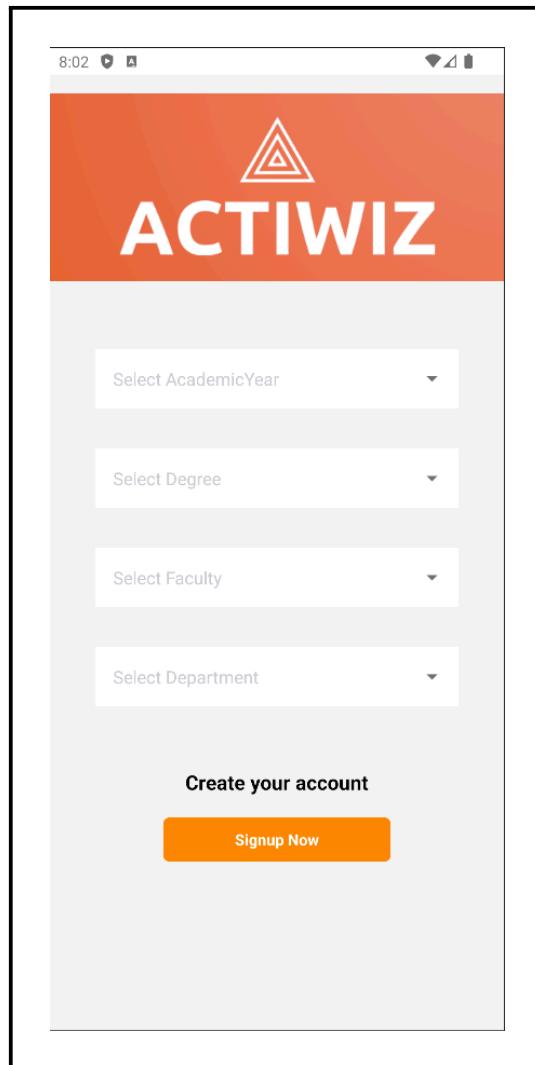
หาก API token ของผู้ใช้หมดอายุ และ Refresh Token ยังไม่หมดอายุ แอปพลิเคชันจะทำการส่ง Request ที่ประกอบด้วย Refresh Token ไปยัง Backend System เพื่อขอ API Token ทั้ง 2 token ใหม่ แต่หาก Refresh Token หมดอายุ ผู้ใช้จะต้องทำการเข้าสู่ระบบผ่านระบบของ Microsoft อีกครั้ง

เมื่อแอปพลิเคชันมี API Token ทั้ง 2 token แล้ว ระบบของแอปพลิเคชันจะทำการส่ง Request ที่ประกอบด้วย API Token ทั้ง 2 token ไปยัง Backend System ผ่านการเรียกใช้งาน API สำหรับการตรวจสอบข้อมูลผู้ใช้รวมถึงค่าตอบแทนที่ได้รับ แอปพลิเคชัน Actiwiz หรือไม่ หากมีข้อมูลผู้ใช้ดังกล่าวอยู่แล้ว Backend System จะส่ง user id มาให้กับทางแอปพลิเคชันเพื่อใช้ในการเข้าใช้ระบบอื่น ๆ ของทางแอปพลิเคชันต่อไป และเปลี่ยนหน้าของแอปพลิเคชันเป็นหน้าสำหรับการແນະนำกิจกรรมตาม รูปที่ 4.6



รูปที่ 4.6: หน้าແນະนำกิจกรรม

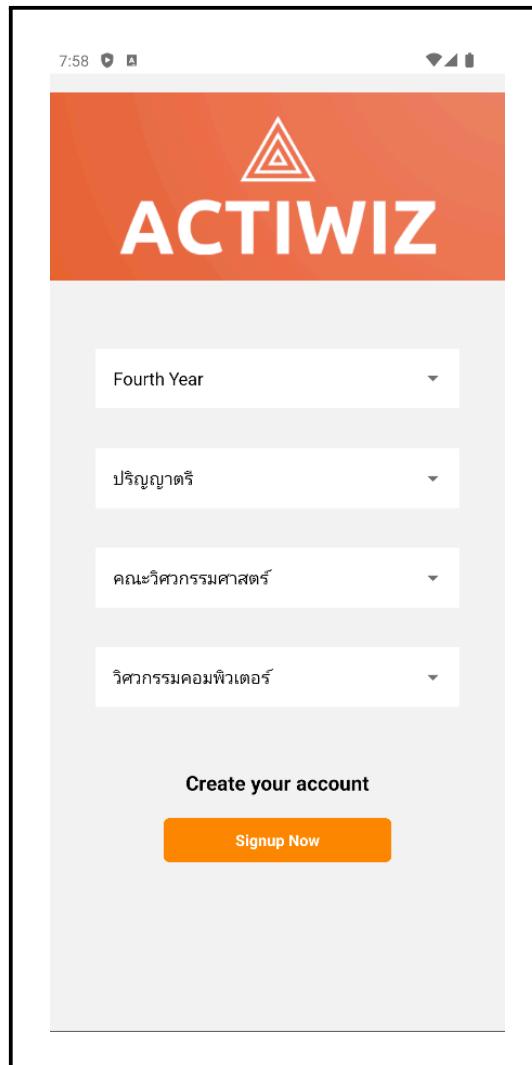
แต่หากไม่มีข้อมูลผู้ใช้ดังกล่าวอยู่ในฐานข้อมูล แอปพลิเคชันจะเปลี่ยนเป็นหน้าสำหรับลงทะเบียนผู้ใช้กับแอปพลิเคชันดัง รูปที่ 4.7



รูปที่ 4.7: หน้าลงทะเบียนผู้ใช้

4.2.2.3 ลงทะเบียนผู้ใช้กับแอปพลิเคชัน

หลังจากเข้าสู่ระบบ หากไม่มีข้อมูลผู้ใช้ดังกล่าวอยู่ในฐานข้อมูล ผู้ใช้จำเป็นต้องให้ข้อมูลเพิ่มเติมเพื่อเข้าใช้งานระบบ เนื่องจากการขอข้อมูลจาก Microsoft Graph Service สามารถขอข้อมูลได้แค่ชื่อและอีเมล ทำให้ต้องมีการขอข้อมูลที่จำเป็นเพิ่มเติมจากหน้าลงทะเบียนตามที่ระบุไว้ใน รูปที่ 4.7 โดยหลังจากผู้ใช้เลือกข้อมูลทุกอย่างเสร็จเรียบร้อยตาม รูปที่ 4.8



รูปที่ 4.8: เลือกข้อมูลครบถ้วน

ผู้ใช้สามารถกดปุ่ม Signup Now หลังจากนั้นแอปพลิเคชันจะส่ง Request ไปยัง Backend System เพื่อเพิ่มข้อมูลผู้ใช้ใหม่ในฐานข้อมูล หลังจากนั้น Backend System จะส่ง User Id มาให้บันทึกในแอปพลิเคชันเพื่อใช้ในการเข้าใช้ระบบอีกครั้งหนึ่ง ของทางแอปพลิเคชันต่อไป และเปลี่ยนหน้าของแอปพลิเคชันเป็นหน้าสำหรับการแนะนำกิจกรรมตาม รูปที่ 4.6

4.2.2.4 แนะนำกิจกรรมตามความสนใจของผู้ใช้โดยอ้างอิงจาก tag ของกิจกรรม

หลังจากทำการเข้าสู่ระบบ ผู้ใช้จะเข้ามาสู่หน้าแนะนำกิจกรรมตาม รูปที่ 4.6 โดยในการแนะนำกิจกรรม แอปพลิเคชันจะส่ง Request ไปยัง Backend System เพื่อขอรายการกิจกรรมแนะนำ โดย Backend System จะใช้หลักการเรียงลำดับกิจกรรมที่ควรจะแนะนำตามหัวข้อ 4.2.1 และเมื่อผู้ใช้เลื่อนลงมาหากข้อมูลของกิจกรรมที่แนะนำยังมีให้โหลดเพิ่มได้ผู้ใช้สามารถกดปุ่ม Load More ตาม รูปที่ 4.9



รูปที่ 4.9: ขอข้อมูลกิจกรรมแนะนำเพิ่ม

4.2.2.5 แนะนำชุมชน ตามความสนใจของผู้ใช้โดยอ้างอิงจาก tag ของกิจกรรม

หลังจากทำการเข้าสู่ระบบ ผู้ใช้สามารถกดไปที่ปุ่ม Club ที่อยู่ด้านล่างเพื่อเปลี่ยนรูปแบบของแอปพลิเคชันเป็นโหมดสำหรับชุมชนตามรูปที่ 4.6 โดยในการแนะนำชุมชน และแอปพลิเคชันจะส่ง Request ไปยัง Backend System เพื่อขอรายการชุมชนแนะนำ โดย Backend System จะใช้หลักการเรียกลำดับชุมชนที่ควรจะแนะนำตาม หัวข้อ 4.2.1 และเมื่อผู้ใช้เลือกลงมาหากข้อมูลของชุมชนที่แนะนำยังคงให้โหลดเพิ่มได้ผู้ใช้สามารถกดปุ่ม Load More ตาม รูปที่ 4.10

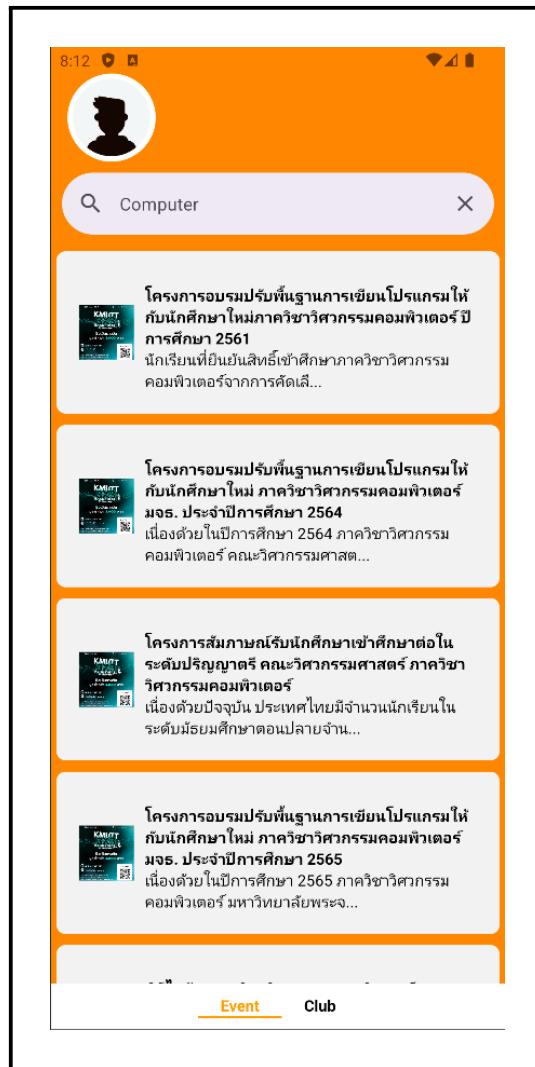


รูปที่ 4.10: ขอข้อมูลชมรมแนะนำเพื่อน

เพื่อโหลดข้อมูลกิจกรรมเพิ่มเติมได้

4.2.2.6 ค้นหากิจกรรมที่สนใจ

จากหน้าแนะนำกิจกรรม รูปที่ 4.6 ผู้ใช้สามารถค้นหากิจกรรมที่ต้องการได้ด้วยการกดปุ่ม Search และพิมพ์คำตามที่ต้องการ หลังจากนั้นแอปพลิเคชันจะส่ง Request ไปยัง Backend System เพื่อขอรายการกิจกรรมที่มีคำตั้งกล่าวอยู่ในชื่อ หลังจากแอปพลิเคชันได้รับการตั้งกล่าวมาแล้วจึงนำมาทำการแสดงผลเป็นการ์ดของกิจกรรมดัง รูปที่ 4.11



รูปที่ 4.11: หน้าค้นหากิจกรรม

และหากข้อมูลของกิจกรรมที่ค้นหายังไม่โหลดเพิ่มได้ผู้ใช้สามารถกดปุ่ม Load More ตาม รูปที่ 4.9 เพื่อโหลดข้อมูลกิจกรรมเพิ่มเติมได้

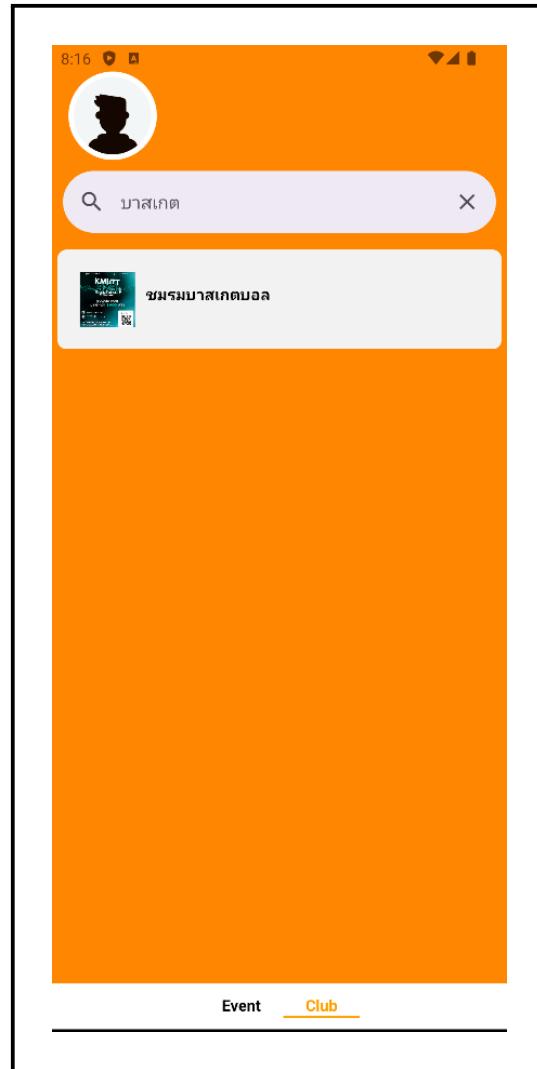
4.2.2.7 ค้นหาชุมชนที่สนใจ

จากหน้าแนะนำกิจกรรม รูปที่ 4.6 ผู้ใช้สามารถกดไปที่ปุ่ม Club ที่อยู่ด้านล่างเพื่อเปลี่ยนรูปแบบของแอปพลิเคชันเป็นโหมดสำหรับชมรูปตาม รูปที่ 4.12



รูปที่ 4.12: หน้าแสดงชมรมที่แน่นำ

และผู้ใช้สามารถค้นหาชมรมที่ต้องการได้ด้วยการกดปุ่ม และพิมพ์คำตามที่ต้องการ หลังจากนั้นแอปพลิเคชันจะส่ง Request ไปยัง Backend System เพื่อขอรายการชมรมที่มีคำดังกล่าวอยู่ในชื่อ หลังจากแอปพลิเคชันได้รายการดังกล่าวมาแล้วจึงนำมาทำการแสดงผล เป็นการ์ดของชมรมดัง รูปที่ 4.13



รูปที่ 4.13: หน้าค้นหาชุมชน

และหากข้อมูลของชุมชนที่ค้นหายังไม่โหลดเพิ่มได้ผู้ใช้สามารถกดปุ่ม Load More ตาม รูปที่ 4.10 เพื่อโหลดข้อมูลชุมชนเพิ่มเติมได้

4.2.2.8 เลือกอ่านรายละเอียดกิจกรรมที่สนใจ

จากหน้าแนะนำกิจกรรม รูปที่ 4.6 หรือหน้าค้นหากิจกรรม รูปที่ 4.11 ผู้ใช้สามารถกดเข้าไปในการดูของกิจกรรมที่ตัวเองสนใจได้ จากนั้นแอปพลิเคชันจะเปลี่ยนเป็นหน้าที่แสดงรายละเอียดของกิจกรรมนั้น โดยเป็นการนำข้อมูลจากการดึงมาจากแอปฯ ตาม รูปที่ 4.14



รูปที่ 4.14: ข้อมูลกิจกรรม

และแอปพลิเคชันจะส่ง Request ไปหา Backend System เพื่อดำเนินการสองอย่างได้แก่ เก็บข้อมูลจำนวนครั้งที่ผู้ใช้เข้ามาอ่านข้อมูล กิจกรรมดังกล่าว และตรวจสอบว่าผู้ใช้เข้าร่วมกิจกรรมดังกล่าวแล้วหรือไม่

4.2.2.9 เลือกอ่านรายละเอียดข้อมูลที่สนใจ

จากหน้าแนะนำข้อมูล รูปที่ 4.12 หรือหน้าค้นหาข้อมูล รูปที่ 4.13 ผู้ใช้สามารถกดเข้าไปในรายละเอียดของข้อมูลที่ตัวเองสนใจได้ จากนั้น แอปพลิเคชันจะเปลี่ยนเป็นหน้าที่แสดงรายละเอียดของข้อมูลนั้น โดยเป็นการนำข้อมูลจากการดึงข้อมูลมาแสดง ตาม รูปที่ 4.15



รูปที่ 4.15: รายละเอียดของชมรม

และแอปพลิเคชันจะส่ง Request ไปท่า Backend System เพื่อดำเนินการสองอย่างได้แก่ เก็บข้อมูลจำนวนครั้งที่ผู้ใช้เข้ามาอ่านข้อมูล ชมรมดังกล่าว และตรวจสอบว่าผู้ใช้เข้าร่วมชมรมดังกล่าวแล้วหรือไม่

4.2.2.10 ลงชื่อเข้าร่วมกิจกรรมที่สนใจ

จากหน้าแสดงรายละเอียดกิจกรรม รูปที่ 4.14 หากผู้ใช้เลื่อนลงมาจนถึงด้านล่างสุดของหน้าตาม รูปที่ 4.16



รูปที่ 4.16: ปุ่มเข้าร่วมกิจกรรม

ผู้ใช้สามารถกดปุ่มเพื่อเข้าร่วมกิจกรรมดังกล่าวได้ และหลังจากกดเข้าร่วมแล้วจะมีการส่ง Request ไปท่า Backend System เพื่อเก็บ
ข้อมูลการเข้าร่วมของผู้ใช้และเปลี่ยนรูปแบบปุ่มเข้าร่วมกิจกรรมเป็นแบบ รูปที่ 4.17



ຮູບທີ່ 4.17: ຮູບເນື້ອເຂົ້າຮ່ວມກິຈການແລ້ວ

ໂດຍນອກຈາກການຮັບທີ່ມີຮັບຮັບສົມມົນສຶກຂາ ໃຫ້ກັບນັກສຶກຂາ ເພື່ອເຂົ້າຮ່ວມກິຈການແລ້ວ ຮູ່ປູ້ມີເຂົ້າຮ່ວມກິຈການກົ່ຈະເປີ່ຍນ
ເປັນແບບ ຮູບທີ່ [4.17](#) ເຊັ່ນກັນ

4.2.2.11 ລົງທຶນເຂົ້າຮ່ວມຂໍມຽນທີ່ສົນໃຈ

ຈາກໜ້າແສດງຮາຍລະເລື່ອດໍາມຽນ ຮູບທີ່ [4.15](#) ຜູ້ໃຊ້ສາມາດຮັບທີ່ມີຮັບຮັບສົມມົນສຶກຂາໃໝ່ ແລະ ລັບກັດເຂົ້າຮ່ວມແລ້ວຈະມີການສ່າງ Request ໄປຫາ Backend System ເພື່ອເກັບຂໍ້ມູນການເຂົ້າຮ່ວມຂໍມຽນຂອງຜູ້ໃຊ້ ແລະ ເປີ່ຍນຮູ່ປູ້ມີເຂົ້າຮ່ວມຂໍມຽນເປັນປູ້ມີອອກຈາກຂໍມຽນຕາມ ຮູບທີ່ [4.18](#)



รูปที่ 4.18: ปุ่มลาออกจากชมรม

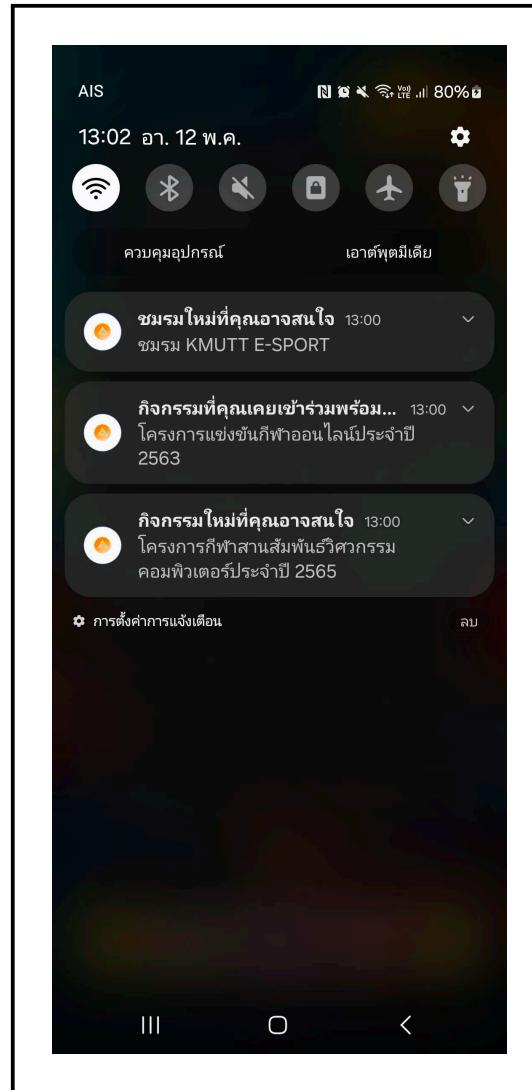
โดยนอกจากการกดเข้าร่วมแล้วหากตรวจสอบว่าผู้ใช้เข้าร่วมชมรมดังกล่าวแล้วตั้งแต่แรก รูปแบบปุ่มเข้าร่วมชมรมก็จะเปลี่ยนเป็นปุ่มออกจากชมรมตาม รูปที่ 4.18 เช่นกัน

4.2.2.12 ถอนชื่อจากการเป็นสมาชิกชมรม

จากหน้าแสดงรายละเอียดชมรม รูปที่ 4.15 หากผู้ใช้เข้าร่วมชมรมนั้นแล้วรูปแบบปุ่มจะกลายเป็นปุ่มออกจากชมรมตาม รูปที่ 4.18 และผู้ใช้สามารถกดออกจากชมรมได้ หลังจากกดออกจากชมรมแล้วจะมีการส่ง Request ไปหา Backend System เพื่อลบข้อมูลการเข้าร่วมชมรมดังกล่าวของผู้ใช้ และเปลี่ยนรูปแบบปุ่มออกจากชมรมเป็นปุ่มเข้าร่วมชมรมตาม รูปที่ 4.15

4.2.2.13 แจ้งเตือนกิจกรรมที่เกี่ยวข้องกับชมรมหรือความสนใจของนักศึกษา

เมื่อมีกิจกรรมใหม่ จะมีการส่งให้ผู้ใช้กดสำหรับการคำนวณว่ากิจกรรมดังกล่าวเกี่ยวข้องกับชมรมหรือความสนใจของผู้ใช้หรือไม่ ถ้าเกี่ยวข้องให้เก็บความสัมพันธ์ระหว่างกิจกรรมดังกล่าวกับผู้ใช้ไว้ ควรจะแจ้งเตือนกิจกรรมดังกล่าวแก่ผู้ใช้ และจะมีโค้ดของ Backend System ที่ทำงานทุก ๆ 6 ชั่วโมง เพื่อตรวจสอบความสัมพันธ์ดังกล่าวและเมื่อตรวจสอบพบว่ามีความสัมพันธ์ดังกล่าวที่เชื่อมโยงผู้ใช้กับกิจกรรมให้ทำการแจ้งเตือนไปยังผู้ใช้ผ่านระบบของ Firebase Cloud Messaging โดยเป็นรูปแบบการแจ้งเตือนแบบ Push Notification ตาม รูปที่ 4.19



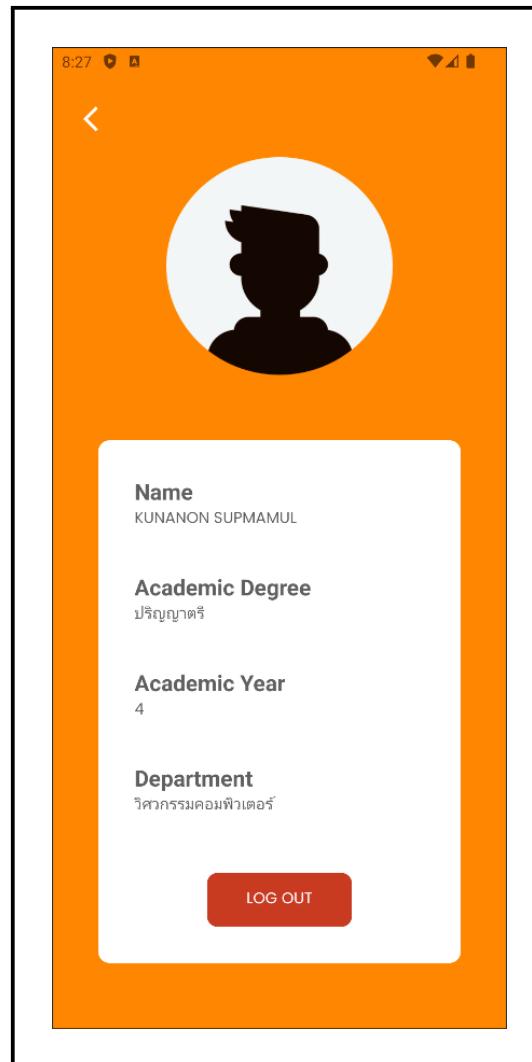
รูปที่ 4.19: การแจ้งเตือนกิจกรรมที่เกี่ยวข้องกับชัมรมหรือความสนใจของนักศึกษา

4.2.2.14 แจ้งเตือนการประเมินกิจกรรม

เมื่อมีกิจกรรมที่ผู้ใช้เคยเข้าร่วมพร้อมให้ประเมิน จะมีการสั่งให้เด็กสำหรับการเก็บความสัมพันธ์ระหว่างกิจกรรมดังกล่าวกับผู้ใช้ไว้ว่า ควรจะแจ้งเตือนการประเมินกิจกรรมดังกล่าวแก่ผู้ใช้ และจะมีโค้ดของ Backend System ที่ทำงานทุก ๆ 6 ชั่วโมง เพื่อตรวจสอบหากความสัมพันธ์ดังกล่าวและเมื่อเจอกิจกรรมสัมพันธ์ดังกล่าวที่เชื่อมโยงผู้ใช้กับกิจกรรมให้ทำการแจ้งเตือนไปยังผู้ใช้ผ่านระบบของ Firebase Cloud Messaging โดยเป็นรูปแบบการแจ้งเตือนแบบ Push Notification ตาม รูปที่ 4.19

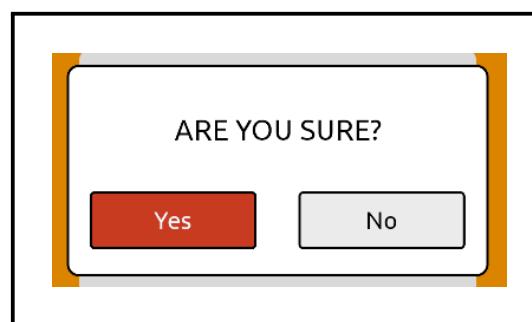
4.2.2.15 ออกรายงาน

จากหน้าแนะนำกิจกรรมและชัมรมตาม รูปที่ 4.6 ผู้ใช้สามารถออกจากรายงานได้ด้วยการกดปุ่ม Profile ทางด้านบนซ้ายของ รูปที่ 4.6 ได้เพื่อเข้าไปที่หน้า User Profile ตาม รูปที่ 4.20



รูปที่ 4.20: หน้าข้อมูลผู้ใช้

หลังจากนั้นผู้ใช้สามารถกดปุ่ม LOG OUT และระบบจะแจ้งเตือนขึ้นว่า ARE YOU SURE? ตาม รูปที่ 4.21



รูปที่ 4.21: การยืนยันการออกจากระบบ

โดยผู้ใช้สามารถกดปุ่ม Yes เพื่ออกจากระบบได้ หลังจากนั้นแอปพลิเคชันจะลบ Token ทั้งหมดและนำผู้ใช้กลับมายังหน้าแรกสุดของแอปพลิเคชันตาม รูปที่ 4.4 หรือผู้ใช้สามารถเลือกที่จะกดปุ่ม No เพื่อยกเลิกการออกจากระบบได้

4.3 การทดสอบการทำงาน (Functional Testing)

พิจาร์	สถานการณ์ในการทดสอบระบบ	ผลลัพธ์ที่คาดว่าจะได้รับ	ผลลัพธ์ที่ได้รับ
Login	กรณีผู้ใช้ไม่กรอกอีเมล	กรณีผู้ใช้ไม่กรอกอีเมล จะมีการแจ้งเตือนให้ผู้ใช้ กรอกอีเมลก่อนที่จะกดเข้าสู่ระบบ	เป็นไปตามที่คาดหวัง
Login	กรณีผู้ใช้ไม่กรอกอีเมลไม่ถูกต้อง	กรณีผู้ใช้ไม่กรอกอีเมลไม่ถูกต้อง จะมีการแจ้งเตือนให้ผู้ใช้กรอกอีเมลใหม่ก่อนที่จะกดเข้าสู่ระบบ	เป็นไปตามที่คาดหวัง
Login	กรณีผู้ใช้ไม่กรอกรหัสผ่าน	กรณีผู้ใช้ไม่กรอกรหัสผ่าน จะมีการแจ้งเตือนให้ผู้ใช้กรอกรหัสผ่านก่อนที่จะกดเข้าสู่ระบบ	เป็นไปตามที่คาดหวัง
Login	กรณีผู้ใช้กรอกรหัสผ่านผิด	กรณีผู้ใช้กรอกรหัสผ่านผิด จะมีการแจ้งเตือนให้ผู้ใช้กรอกรหัสผ่านใหม่ก่อนที่จะกดเข้าสู่ระบบ	เป็นไปตามที่คาดหวัง
Create Account	กรณีที่ไม่มีบัญชีผู้ใช้ในฐานข้อมูล	กรณีที่ไม่มีบัญชีผู้ใช้ในฐานข้อมูล ระบบจะให้ผู้ใช้กรอกข้อมูลต่อไปก่อนที่จะเข้าสู่ระบบ	เป็นไปตามที่คาดหวัง
Create Account	กรณีผู้ใช้กรอกข้อมูล	กรณีผู้ใช้กรอกข้อมูลไม่ครบ ระบบจะไม่สามารถสร้างบัญชีได้จากว่าผู้ใช้จะกรอกข้อมูลให้ครบถ้วน	เป็นไปตามที่คาดหวัง
Logout	กรณีที่ผู้ใช้ต้องการออกจากระบบ	กรณีผู้ใช้ต้องการออกจากระบบจะมีการแจ้งเตือน ตามขั้นตอนการออกจากระบบก่อนที่จะออกจากระบบ	เป็นไปตามที่คาดหวัง
Logout	กรณีผู้ใช้กดปุ่ม No หรือ กดนอกกรอบแจ้งเตือน	กรณีผู้ใช้กดปุ่ม No หรือ กดนอกกรอบแจ้งเตือนจะถือว่า เป็นการยกเลิกการออกจากระบบทั้งสองอย่าง	เป็นไปตามที่คาดหวัง
Search	กรณีที่ผู้ใช้ต้องการค้นหากิจกรรม	กรณีที่ผู้ใช้ต้องการค้นหากิจกรรม จะสามารถหาได้จากແບບค้นหาของแอปพลิเคชัน	เป็นไปตามที่คาดหวัง
FeedPage (Event)	กรณีผู้ใช้ลองกดเข้าไปดูรายละเอียดของกิจกรรมที่ไม่มีข้อมูล	กรณีผู้ใช้ลองกดเข้าไปดูรายละเอียดของกิจกรรมที่ไม่มีข้อมูล จะไม่พบข้อมูลใดๆ	เป็นไปตามที่คาดหวัง
FeedPage (Club)	กรณีผู้ใช้ลองกดเข้าไปดูรายละเอียดของชุมชนที่ไม่มีข้อมูล	กรณีผู้ใช้ลองกดเข้าไปดูรายละเอียดของชุมชนที่ไม่มีข้อมูล จะไม่พบข้อมูลใดๆ	เป็นไปตามที่คาดหวัง
Notification	กรณีผู้ใช้ตรวจสอบการแจ้งเตือนของตัวแอปพลิเคชัน	กรณีผู้ใช้ตรวจสอบการแจ้งเตือนของตัวแอปพลิเคชัน จะมีการแสดงการแจ้งเตือนนั้นหมดที่เกิดขึ้นในแอปพลิเคชัน	เป็นไปตามที่คาดหวัง
Recommendation	กรณีผู้ใช้ตรวจสอบกิจกรรมที่ทางระบบแนะนำมาให้โดยในเดือนยังนี้จะเป็นการอ้างอิงจากคนที่ผู้ใช้อ่าน	กรณีผู้ใช้ตรวจสอบกิจกรรมที่ทางระบบแนะนำมาให้โดยในเดือนยังนี้จะเป็นการอ้างอิงจากคนที่ผู้ใช้อ่าน	เป็นไปตามที่คาดหวัง

ตารางที่ 4.4

การทดสอบการทำงาน

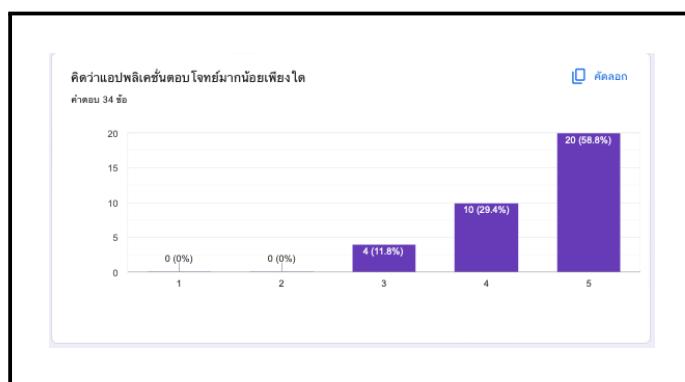
4.4 การทดสอบคุณภาพของระบบ (Non Functional Testing)

จากการทำแอปพลิเคชันให้นักศึกษาในมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรีที่ศึกษาอยู่ในระดับปริญญาตรีตั้งแต่ชั้นปีที่ 1 ถึงชั้นปีที่ 4 จำนวนทั้งสิ้น 34 คน ทำการทดสอบการใช้งานแอปพลิเคชัน Activiz ได้ผลลัพธ์จากการประเมินแยกเป็นหัวข้อต่าง ๆ โดยนำคะแนนในแต่ละหัวข้อของแต่ละผู้ทดสอบมาหาค่าเฉลี่ยโดยนำผลสุดท้าย ที่ได้มาเบริร์บเทียบกับเกณฑ์การประเมินเว็บแอปพลิเคชันโดยมีค่าคะแนนเดิมอยู่ที่ 5 คะแนน ดังนี้ 1. ควรปรับปรุง: 0 - 1.5 คะแนน 2. พอดี: 1.5 - 2.5 คะแนน 3. ดี: 2.5 - 3.5 คะแนน 4. ดีมาก: 3.5 - 4.5 คะแนน 5. ดีเยี่ยม: 4.5 - 5 คะแนน

- คิดว่าแอปพลิเคชันนี้ตอบโจทย์กับปัญหาหรือไม่ ?

ระดับความพึงพอใจ	จำนวนผู้ให้คะแนน
1	0
2	0
3	4
4	10
5	20

ตารางที่ 4.5 ตารางความพึงพอใจในการแก้ปัญหา



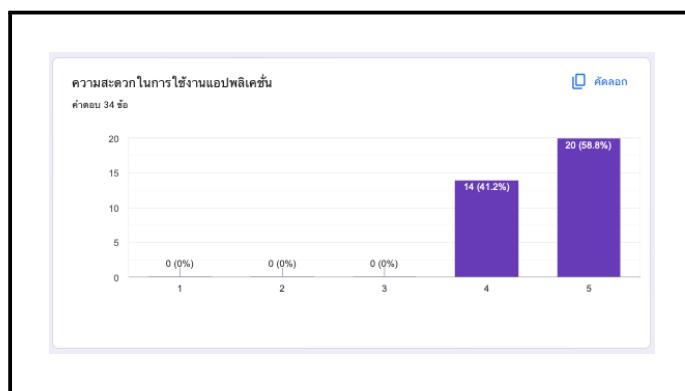
รูปที่ 4.22: แผนภาระคะแนนความพึงพอใจในการแก้ปัญหาของแอปพลิเคชัน

จากผลลัพธ์การประเมินในส่วนของการตอบโจทย์ของปัญหาตามตารางที่ 4.5 และรูปที่ 4.22 พบว่าค่าเฉลี่ยของความพึงพอใจจากคะแนนจาก 5 คะแนนมีค่าเฉลี่ยอยู่ที่ 4.47 ซึ่งนับว่าค่าเฉลี่ยของผลการประเมินอยู่ในเกณฑ์ดี

- ความสะดวกในการใช้งานแอปพลิเคชัน

ระดับความพึงพอใจ	จำนวนผู้ให้คะแนน
1	0
2	0
3	0
4	14
5	20

ตารางที่ 4.6 ตารางความสะดวกในการใช้งานแอปพลิเคชัน



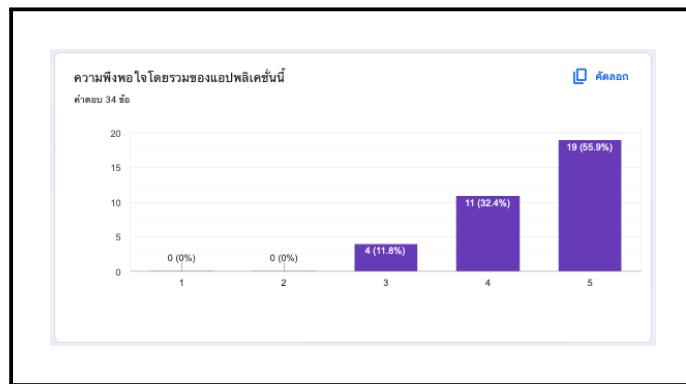
รูปที่ 4.23: แผนภูมิแท่งแสดงความสะดวกในการใช้งานแอปพลิเคชัน

จากผลลัพธ์การประเมินในส่วนของความสะดวกในการใช้งานแอปพลิเคชันตามตารางที่ 4.6 และรูปที่ 4.23 พบว่าค่าเฉลี่ยของความพึงพอใจจากคะแนนจาก 5 คะแนนมีคะแนนเฉลี่ยอยู่ที่ 4.58 ซึ่งนับว่าค่าเฉลี่ยของผลการประเมินอยู่ในเกณฑ์ดีเยี่ยม

- ความพึงพอใจโดยรวมของแอปพลิเคชันนี้

ระดับความพึงพอใจ	จำนวนผู้ให้คะแนน
1	0
2	0
3	4
4	11
5	19

ตารางที่ 4.7 ตารางความพึงพอใจโดยรวมของแอปพลิเคชัน



รูปที่ 4.24: แผนภูมิ柱状圖 แสดงความพึงพอใจโดยรวมของแอปพลิเคชัน

จากผลลัพธ์การประเมินในความพึงพอใจโดยรวมของแอปพลิเคชันตามตารางที่ 4.7 และรูปที่ 4.24 พบร่วมกันว่าค่าเฉลี่ยของความพึงพอใจจากคะแนนจาก 5 คะแนนมีคะแนนเฉลี่ยอยู่ที่ 4.44 ซึ่งนับว่าค่าเฉลี่ยของผลการประเมินอยู่ในเกณฑ์ดีมาก

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

5.1.1.1 การเตรียมการเริ่มต้นโครงการ

ภาระงาน	สถานะงาน
จัดทำ Idea Document	เสร็จสิ้น
จัดทำ Project Proposal	เสร็จสิ้น
จัดทำแผนการดำเนินงานของโครงการ	เสร็จสิ้น

ตารางที่ 5.1 ตารางรายการการเตรียมการเริ่มต้นโครงการ

5.1.1.2 ศึกษาวิธีการพัฒนาแอปพลิเคชัน

ภาระงาน	สถานะงาน
ศึกษาวิธีการพัฒนาแอปพลิเคชันส่วน Front-end	เสร็จสิ้น
ศึกษาวิธีการพัฒนาแอปพลิเคชันส่วน Back-end	เสร็จสิ้น
ศึกษาวิธีการพัฒนาฐานข้อมูล	เสร็จสิ้น
ศึกษาวิธีการพัฒนา Machine Learning	เสร็จสิ้น

ตารางที่ 5.2 ตารางรายการการศึกษาวิธีการพัฒนาแอปพลิเคชัน

5.1.1.3 ออกแบบแอปพลิเคชัน

ภาระงาน	สถานะงาน
จัดทำ Architecture Diagram	เสร็จสิ้น
จัดทำ Use Case Diagram	เสร็จสิ้น
จัดทำ Sequence Diagram	เสร็จสิ้น
จัดทำ Database Schema	เสร็จสิ้น
จัดทำ Application Wireframe	เสร็จสิ้น
จัดทำ Application UX/UI	เสร็จสิ้น

ตารางที่ 5.3 ตารางรายการการออกแบบแอปพลิเคชัน

5.1.1.4 จัดเตรียมข้อมูล

ภาระงาน	สถานะงาน
Data Preprocessing	เสร็จสิ้น

ตารางที่ 5.4 ตารางรายการการจัดเตรียมข้อมูล

จากตาราง 5.1 - 5.4 เป็นตารางสรุปภาระงานและสถานะงานที่ทางทีมพัฒนาได้ดำเนินการไปในภาคการศึกษาที่ 1 โดยภาระงานที่ทางทีมพัฒนาได้วางแผนไว้ในภาคการศึกษานี้จะเป็นการเตรียมความพร้อมสำหรับการพัฒนาแอปพลิเคชันในภาคการศึกษาที่ 2 และ

จากข้อมูลภายในตาราง สามารถสรุปได้ว่าทางผู้จัดทำได้ดำเนินการจัดการทุกภาระงานจนเสร็จสิ้นตามแผนการดำเนินงานที่กำหนดไว้เป็นที่เรียบร้อย

5.1.2 การดำเนินงานในภาคการศึกษาที่ 2

5.1.2.1 การพัฒนาแอปพลิเคชัน

ภาระงาน	สถานะงาน
พัฒนาแอปพลิเคชันส่วน Front-end	เสร็จสิ้น
พัฒนาแอปพลิเคชันส่วน Back-end	เสร็จสิ้น
พัฒนาฐานข้อมูล	เสร็จสิ้น
พัฒนา Machine Learning	เสร็จสิ้น

ตารางที่ 5.5 ตารางรายการการพัฒนาแอปพลิเคชัน

5.1.2.2 การทดสอบแอปพลิเคชัน

ภาระงาน	สถานะงาน
ออกแบบ Test Cases	เสร็จสิ้น
ทดสอบการทำงานของแอปพลิเคชัน (Functional Testing)	เสร็จสิ้น
ทดสอบคุณภาพของระบบ (Non Functional Testing)	เสร็จสิ้น

ตารางที่ 5.6 ตารางรายการการทดสอบแอปพลิเคชัน

จากตาราง 5.5 - 5.6 เป็นตารางสรุปภาระงานและสถานะงานที่ทางทีมพัฒนาวางแผนจะดำเนินการในภาคการศึกษาที่ 2 โดยภาระงานที่ทางทีมพัฒนาได้วางแผนไว้ในภาคการศึกษานี้จะเป็นการพัฒนาแอปพลิเคชันตามแผนที่ได้เตรียมไว้ในภาคการศึกษาที่ 1 โดยสถานะของภาระงานส่วนใหญ่ยังอยู่ในขั้นตอนการดำเนินการ

5.2 ปัญหาที่พบในโครงการและการแก้ไข (Problems and Solutions)

5.2.1 ความไม่คุ้นชินกับเทคโนโลยีที่ใช้ในการพัฒนา

เนื่องจากคณะกรรมการผู้จัดทำแต่ละคนมีความถันดัดในเทคโนโลยีที่แตกต่างกัน และในแต่ละเทคโนโลยีที่ใช้ในการพัฒนาโครงการนั้นผู้จัดทำแต่ละคนก็มีความเชี่ยวชาญที่ไม่เท่ากัน จึงเกิดเป็นอุปสรรคในการทำงานเนื่องจากความไม่คุ้นเคยกับเทคโนโลยีที่ถูกหยิบมาใช้งานในการพัฒนาโครงการ

5.2.2 จำนวนข้อมูลที่สามารถใช้ได้มีน้อย

ข้อมูลกิจกรรมนักศึกษาของทางมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรีที่ถูกบันทึกเข้าไว้ในระบบนั้นมีข้อมูลเพียงปีการศึกษาที่ 2558 เท่านั้น รวมจำนวนกิจกรรมที่บันทึกได้อよถึง 2565 รายการ ซึ่งเป็นจำนวนที่น้อยเมื่อต้องใช้ในการเทรนโมเดลการเรียนรู้ของเครื่องซึ่งจำเป็นต้องใช้ Transfer Learning เพื่อตึงค่าน้ำหนักของโมเดลที่มีการเทรนไว้ก่อนแล้วมาพัฒนาต่อเพื่อให้ได้โมเดลที่สามารถให้ผลลัพธ์ที่เหมาะสมกันมากได้

5.2.3 ความลำบากในการขอข้อมูลจากมหาวิทยาลัย

เนื่องจากหน่วยงานที่เป็นเจ้าของข้อมูลที่ใช้ในการพัฒนาโครงการนั้นเป็นหน่วยงานที่มีงานในการดำเนินงานตลอดเวลา จึงมีความลำบากในการติดต่อเพื่อเข้าไปขอใช้ข้อมูล

5.2.4 หน่วยประมวลผลที่ใช้ในการพัฒนา Machine Learning มีจำนวนน้อย

ในการเทรนโมเดลการเรียนรู้ของเครื่องนั้นต้องอาศัยหน่วยประมวลผลจำนวนมากในการเทรน ซึ่งด้วยหน่วยประมวลผลที่เข้าถึงได้แบบไม่มีค่าใช้จ่ายนั้นมีไม่เพียงพอต่อการพัฒนามोเดลอย่างมีประสิทธิภาพ

5.3 แนวทางการพัฒนาในอนาคต (Future Work)

- หากมีข้อมูลการใช้งานของระบบเพิ่ม จะสามารถสร้างระบบแนะนำที่พิจารณาการแบ่งประเภทผู้ใช้ กิจกรรม และชุมชน จากข้อมูลการใช้งานเพื่อเพิ่มความสามารถในการแนะนำได้
- เนื่องจากการขอข้อมูลของกิจกรรมที่มีรูปภาพอยู่น้อย การทำ Clustering โดยอาศัย Vector จาก WordEmbedding ถึงแม้จะสามารถจัดกลุ่มนี้อีกหลายกลุ่มได้ แต่ก็ยังไม่ชัดเจนพอที่จะตีความว่าเนื้อหาเหล่านี้สื่อถึงอะไร หากมีการเก็บข้อมูลเพิ่มเรื่อยๆ เช่นว่าพฤติกรรมเหล่านี้จะชัดเจนขึ้นจนสามารถอธิบายให้มนุษย์เข้าใจได้ටล่ะ Cluster สื่อถึงเนื้อหาแบบใด
- หากได้รับการสนับสนุนด้านข้อมูลจากทางมหาวิทยาลัย จะสามารถพัฒนาความสามารถของแอปพลิเคชันของเราให้มีความสม่ำเสมอ เป็นปัจจุบัน และมีประสิทธิภาพมากยิ่งขึ้น
- หากได้พัฒนาต่ออย่างเพิ่มเติมเพื่ออำนวยความสะดวกให้แก่นักศึกษามากยิ่งขึ้น เช่น สามารถลงทะเบียนการเรียนได้ผ่านแอปพลิเคชันของเราได้โดยตรง
- หากได้พัฒนาแอปพลิเคชันต่อในอนาคต อย่างการทำothสำหรับคัดลอกไฟล์การลงทะเบียนกิจกรรม เพื่อลดความยุ่งยากในการลงทะเบียนกิจกรรม
- หากได้พัฒนาแอปพลิเคชันต่ออย่างการทำระบบสำหรับคัดกรองกิจกรรมที่เหมาะสมกับนักศึกษา และนักศึกษาสามารถเข้าร่วมได้จริงๆ

หนังสืออ้างอิง

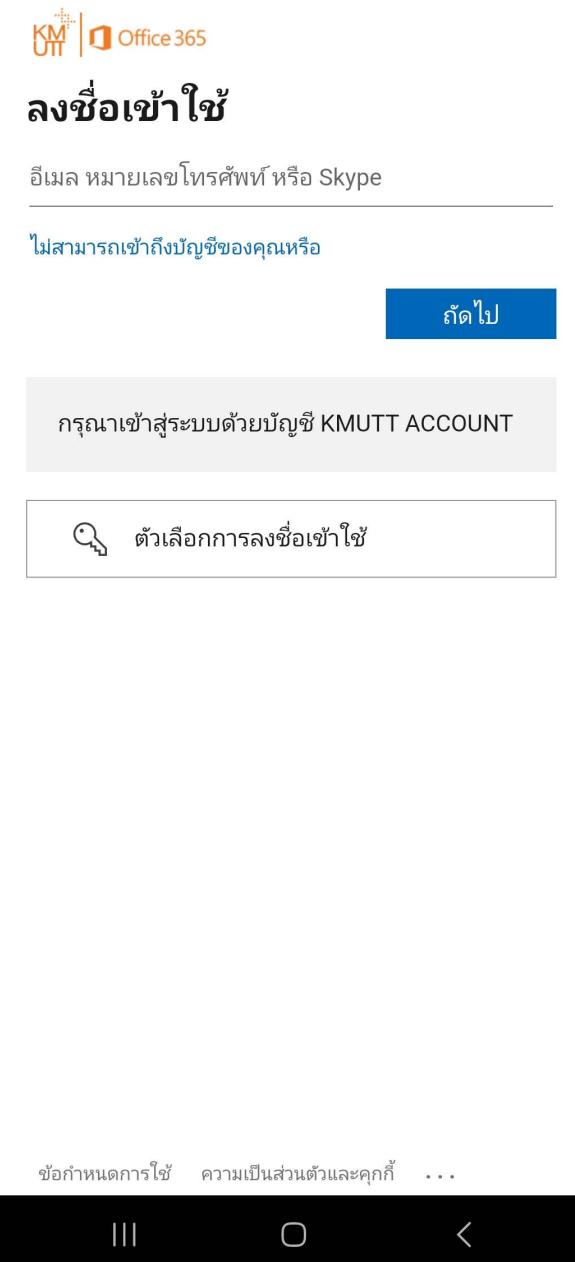
1. Toni Vujevic, 2566, “The Mobile App Architecture Guide,” จากเว็บไซต์ <https://decode.agency/article/mobile-app-architecture/>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
2. Jerapa Beamgo Soonsongthanee, 2563, “ออกแบบ RESTful API ยังไงให้ปัง,” จากเว็บไซต์ <https://medium.com/@jerapabeamgosongsongthanee/%E0%B8%9A%E0%B8%A1%E0%B8%AA%E0%B8%97%E0%B8%9B%E0%B8%99-%E0%B8%9A%E0%B8%A1%E0%B8%AA-%E0%B8%9A%E0%B8%A1%E0%B8%AA-%E0%B8%9A%E0%B8%A1%E0%B8%AA-%E0%B8%9A%E0%B8%A1%E0%B8%AA-%E0%B8%9A%E0%B8%A1-E6-69d2015bcc31>, [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, [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, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
10. Tao, “Repetition,” จากเว็บไซต์ 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, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
12. Tao, “Proximity,” จากเว็บไซต์ 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, [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, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].

20. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin, “Attention Is All You Need,” จากเว็บไซต์ <https://arxiv.org/abs/1706.03762>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
21. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” จากเว็บไซต์ <https://arxiv.org/abs/1810.04805>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
22. Stephen Merity, Nitish Shirish Keskar, and Richard Socher, “Regularizing and Optimizing LSTM Language Models,” จากเว็บไซต์ <https://arxiv.org/abs/1708.02182>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
23. Rufai Mustapha, 2566, “What is HTTP? Protocol Overview for Beginners,” จากเว็บไซต์ <https://www.freecodecamp.org/news/what-is-http/#introduction-to-http>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
24. Settawat Janpuk, 2562, “มาทำความรู้จักกับ REST API กันดีกว่า,” จากเว็บไซต์ <https://medium.com/@settawatjanpuk/beginner-มาทำความรู้จักกับ-rest-api-กันดีกว่า-มีตัวอย่าง-c855d93beaf3>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
25. Jerapa Beamgo Soonsongthanee, 2563, “Pagination,” จากเว็บไซต์ https://miro.medium.com/v2/resize:fit:1400/format:webp/1*nBaoHvynrhkWFWifMW3avA.png, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
26. Tanabodin Kamol, 2562, “Graph database มา กว่า ที่คุณคิด,” จากเว็บไซต์ <https://medium.com/icreativesystems/what-is-graph-database-d0206466f5ca>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
27. Saladpuk, “UML พื้นฐาน - Saladpuk.com,” จากเว็บไซต์ <https://www.saladpuk.com/basic/uml>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
28. Tao, “Principle Design คือ อะไร ทำไง UX/UI Designer ต้องทำตาม,” จากเว็บไซต์ <https://blog.clicknext.com/principle-design/>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
29. Nick Babich, 2560, “F-Shaped Pattern for Reading Content,” จากเว็บไซต์ <https://uxplanet.org/f-shaped-pattern-for-reading-content-80af79cd3394>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
30. mindphp, 2560, “รู้จักกับ Visual Studio Code (วิชาล สตูดิโอ โคด) โปรแกรมพิเศษจากค่ายไมโครซอฟท์,” จากเว็บไซต์ <https://www.mindphp.com/บทความ/microsoft/4829-visual-studio-code.html>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
31. thaiall, “ไฟชัม (pycharm) ใช้เขียนไพทอน (python),” จากเว็บไซต์ <https://www.thaiall.com/pycharm/>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
32. google research, “ยินดีต้อนรับสู่ Colaboratory,” จากเว็บไซต์ <https://colab.research.google.com/?hl=th>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
33. skooldio, 2565, “Figma คืออะไร ทำไงถึงเป็น Tool มาแรงที่สุดในวงการ Design,” จากเว็บไซต์ <https://blog.skooldio.com/figma-ui-design-tool/>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
34. ets kmutt, “Lucidchart,” จากเว็บไซต์ <https://techintegration.ets.kmutt.ac.th/content/tech-review/lucidchart>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
35. codemobiles, “React-Native Programming,” จากเว็บไซต์ https://www.codemobiles.co.th/online/course.php?id=react_native, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
36. Sirasit Boonklang, 2565, “มาทำความรู้จักกับ Fast API,” จากเว็บไซต์ <https://www.borntodev.com/2021/10/08/มาทำความรู้จักกับ-fast-api/>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
37. AppMaster, 2565, “ฐานข้อมูลกราฟ Neo4j: ความสามารถและผลกำไรสำหรับธุรกิจของคุณ,” จากเว็บไซต์ <https://appmaster.io/th/blog/thaankh-muulkraaf-neo4j>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
38. stackpython, 2563, “ทำความรู้จัก Git & GitHub พื้นและการใช้งานร่วมกับ VS Code เปื้องตัน,” จากเว็บไซต์ <https://stackpython.medium.com/ทำความรู้จัก-git-github-พื้นและการใช้งานร่วมกับ-vs-code-เบื้องต้น-f848f41a39e9>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].
39. stackpython, 2562, “Jest คืออะไร เริ่มต้นเขียน Test ด้วย Jest กันดีกว่า,” จากเว็บไซต์ <https://blog.devahoy.com/blog/2019/08/getting-started-with-testing-and-jest>, [ออนไลน์; วันที่สืบค้น 28 พฤศจิกายน 2566].

ภาคผนวก A

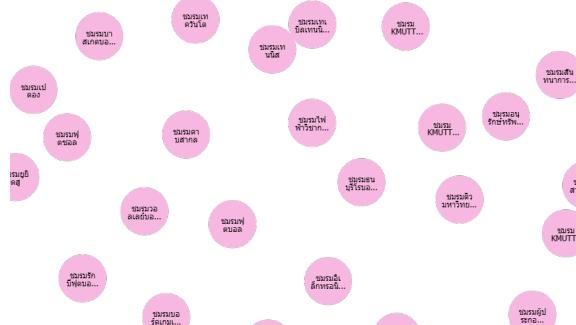
หลักฐานการพัฒนาโครงการ

ทางคณะผู้จัดทำได้จัดทำแอปพลิเคชัน Activiz โดยมีหน้าตาของฟีเจอร์ภายในแอปพลิเคชัน ดังรูปที่ A.1-A.6



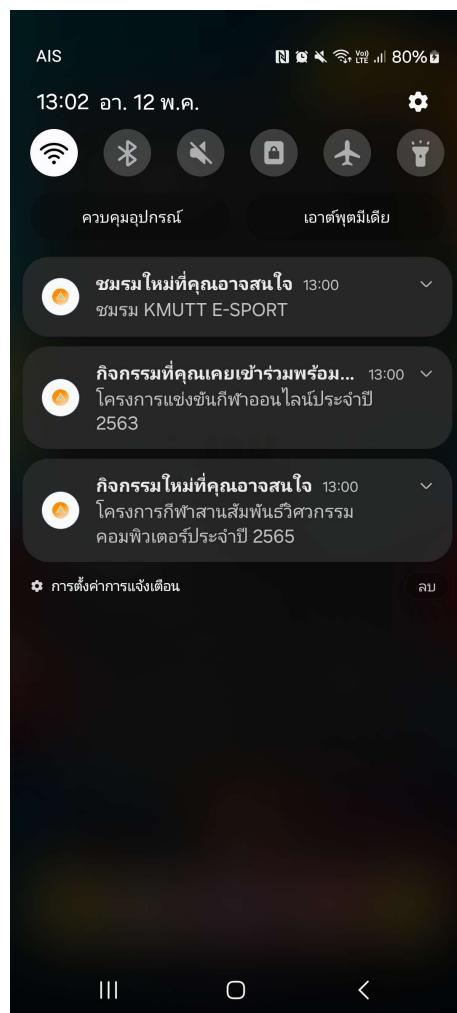
รูปที่ A.1: หน้าเข้าสู่ระบบของแอปพลิเคชัน Activiz

หน้าเข้าสู่ระบบของแอปพลิเคชัน Activiz นั้นเป็นหน้าที่ใช้ในการเข้าสู่ระบบของแอปพลิเคชัน โดยผู้ใช้จะต้องกรอกอีเมล และรหัสผ่านของตนเองเพื่อเข้าสู่ระบบผ่าน Microsoft Login



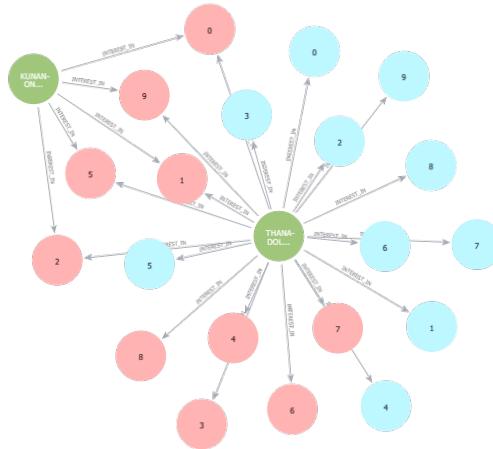
รูปที่ A.2: ฐานข้อมูลของชุมชนในมหาวิทยาลัย

ฐานข้อมูลของชุมชนในมหาวิทยาลัย นั้นเป็นฐานข้อมูลที่เก็บข้อมูลของชุมชนที่มีอยู่ในมหาวิทยาลัย



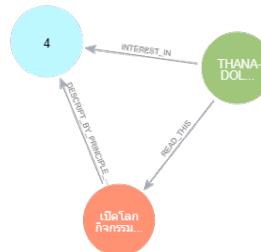
รูปที่ A.3: ระบบแจ้งเตือนของแอปพลิเคชัน Activiz

ระบบ Notification ของแอปพลิเคชัน Activiz นั้นเป็นระบบที่ใช้ในการแจ้งเตือนข้อมูลต่างๆของแอปพลิเคชัน โดยจะแจ้งเตือนเมื่อมีกิจกรรมใหม่ มีชุมชนใหม่ หรือกิจกรรมที่ผู้ใช้เข้าร่วมนั้นสามารถประเมินกิจกรรมได้แล้ว



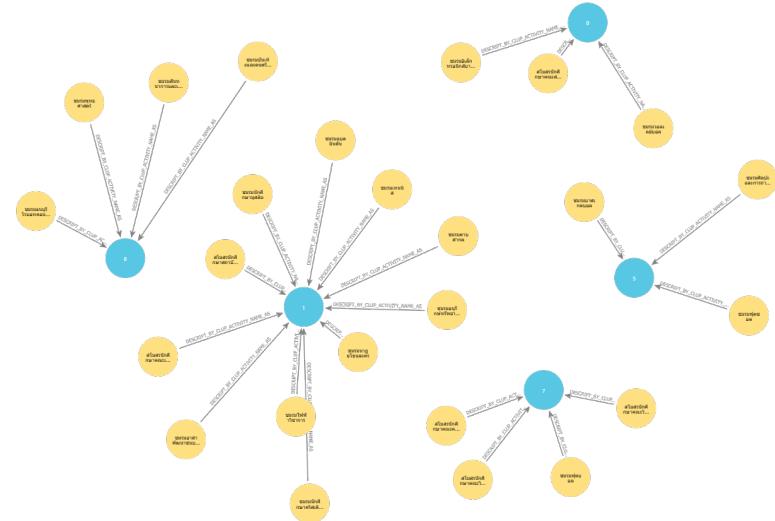
รูปที่ A.4: ความเชื่อมโยงของผู้ใช้กับ tag ต่างๆของกิจกรรม

Model Machine Learning จะเขื่อมโยงผู้ใช้เข้ากับ Tag ของกิจกรรมและชั้นเรียนที่แบ่งประเภทเอาไว้โดยอาศัยข้อมูลของผู้ใช้ งาน เช่น ชั้นปี คณะที่เข้าศึกษา หรือภาควิชาที่เข้าศึกษา ซึ่งความเชื่อมโยงเหล่านี้จะถูกเก็บไว้ในรูปแบบความสัมพันธ์ที่เก็บไว้ในฐานข้อมูล



รูปที่ A.5: การวิเคราะห์ความสนใจจากเนื้อหาที่ผู้ใช้เข้าไปอ่าน

Model Machine Learning คาดเดาความสนใจใน Tag ของกิจกรรมและชั้นเรียนที่แบ่งประเภทเอาไว้ จากข้อมูลการใช้งานของ Activiz เช่น การค้นหา หรือการเข้าไปอ่าน ซึ่งความเชื่อมโยงเหล่านี้จะถูกเก็บไว้ในรูปแบบความสัมพันธ์ที่เก็บไว้ในฐานข้อมูล



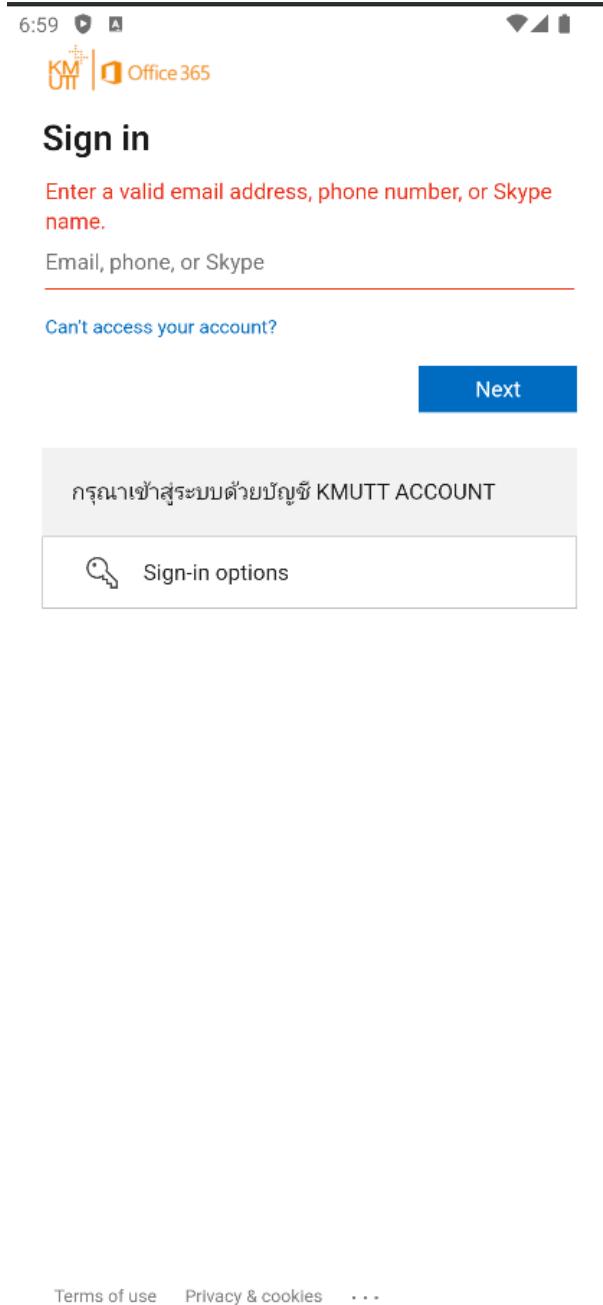
รูปที่ A.6: ตัวอย่างของกิจกรรมที่ถูกแบ่งโดย tag ต่างๆ

กิจกรรมและชุมชนต่างๆจะถูกแบ่งประเภทเป็น Tag ต่างๆจากการทำ Clustering กับ Vector ที่ได้มาจากการทำ Word Embedding ซึ่งแสดงถึงเนื้อหาของกิจกรรมและชุมชนต่างๆเอาไว้ โดย Tag เหล่านี้จะถูกเก็บไว้เป็นโหนดในฐานข้อมูล

ภาคผนวก B

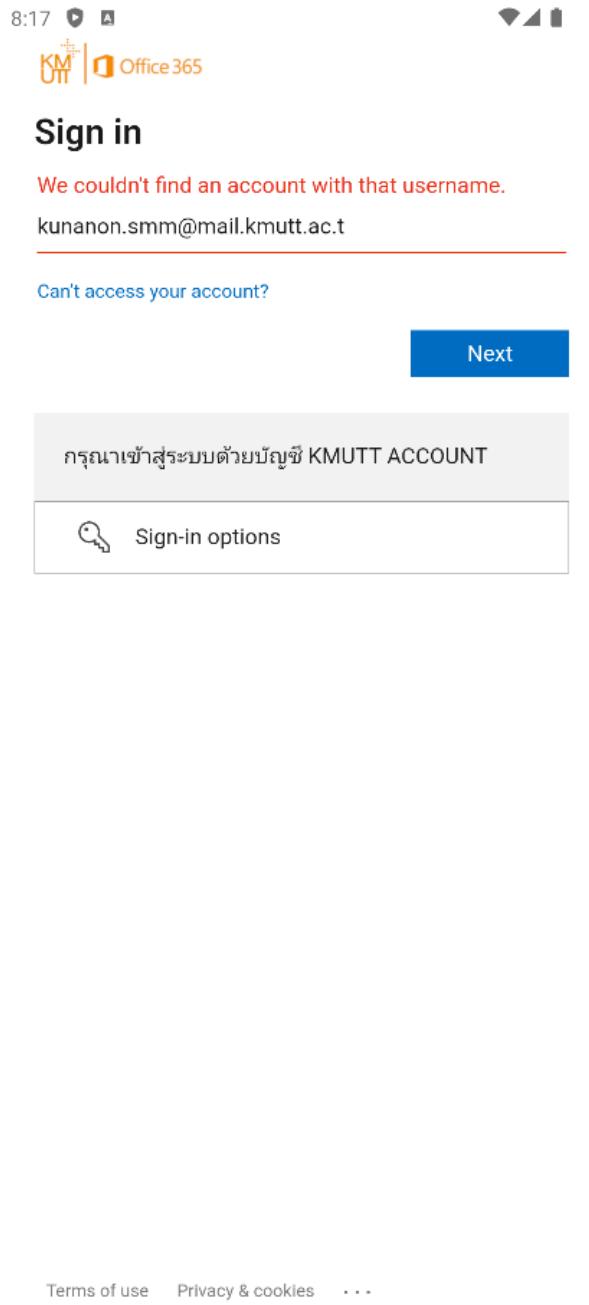
FUNCTIONAL TEST

ทางคณะผู้จัดทำได้จัดทำแอปพลิเคชัน Activiz โดยมีหน้าตาของฟีเจอร์ภายในแอปพลิเคชัน ดังรูปที่ B.1-B.15



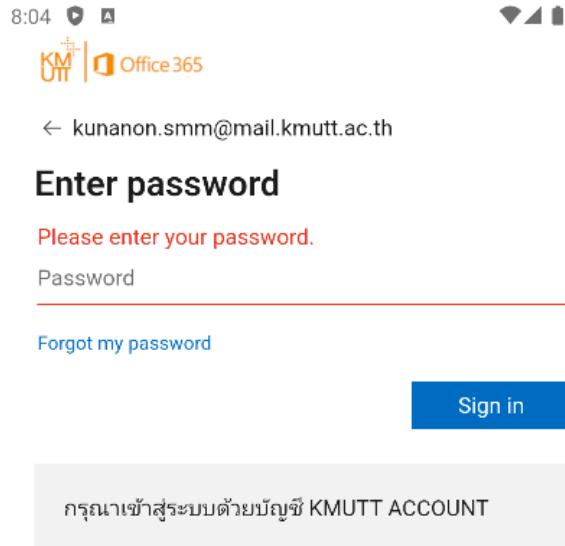
รูปที่ B.1: กรอบผู้ใช้ไม่กรอกอีเมล

กรณีผู้ใช้ไม่กรอกอีเมล จะมีการแจ้งเตือนให้ผู้ใช้กรอกอีเมลก่อนที่จะกดเข้าสู่ระบบ



รูปที่ B.2: กรณีผู้ใช้ไม่กรอกอีเมลไม่ถูกต้อง

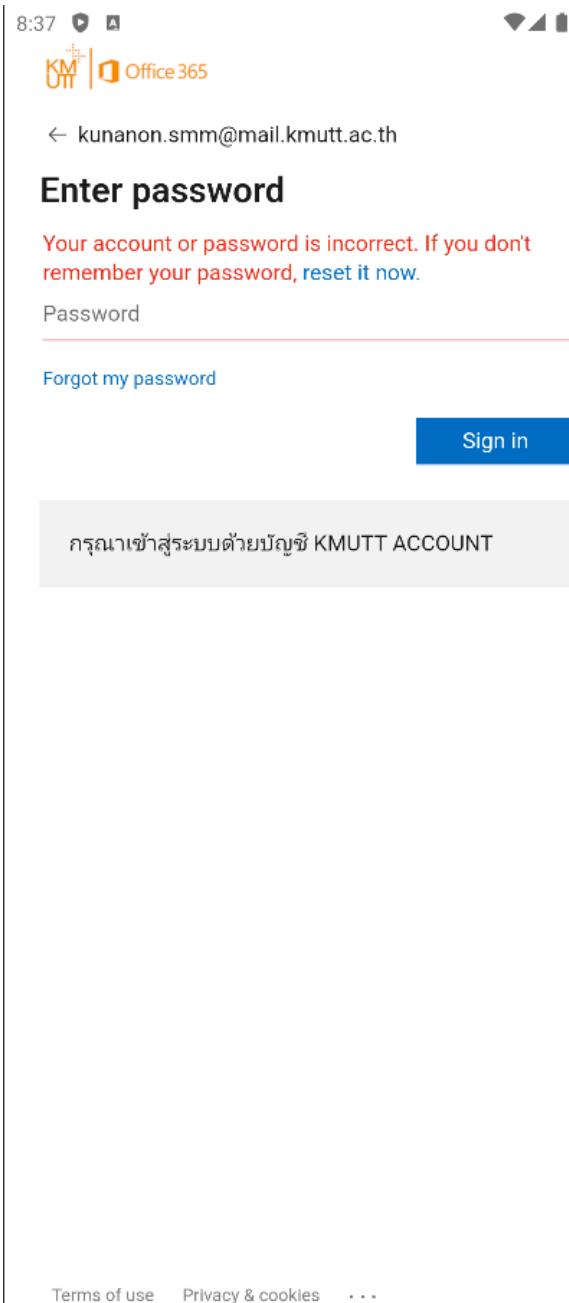
กรณีผู้ใช้ไม่กรอกอีเมลไม่ถูกต้อง จะมีการแจ้งเตือนให้ผู้ใช้กรอกอีเมลใหม่ก่อนที่จะกดเข้าสู่ระบบ



[Terms of use](#) [Privacy & cookies](#) [...](#)

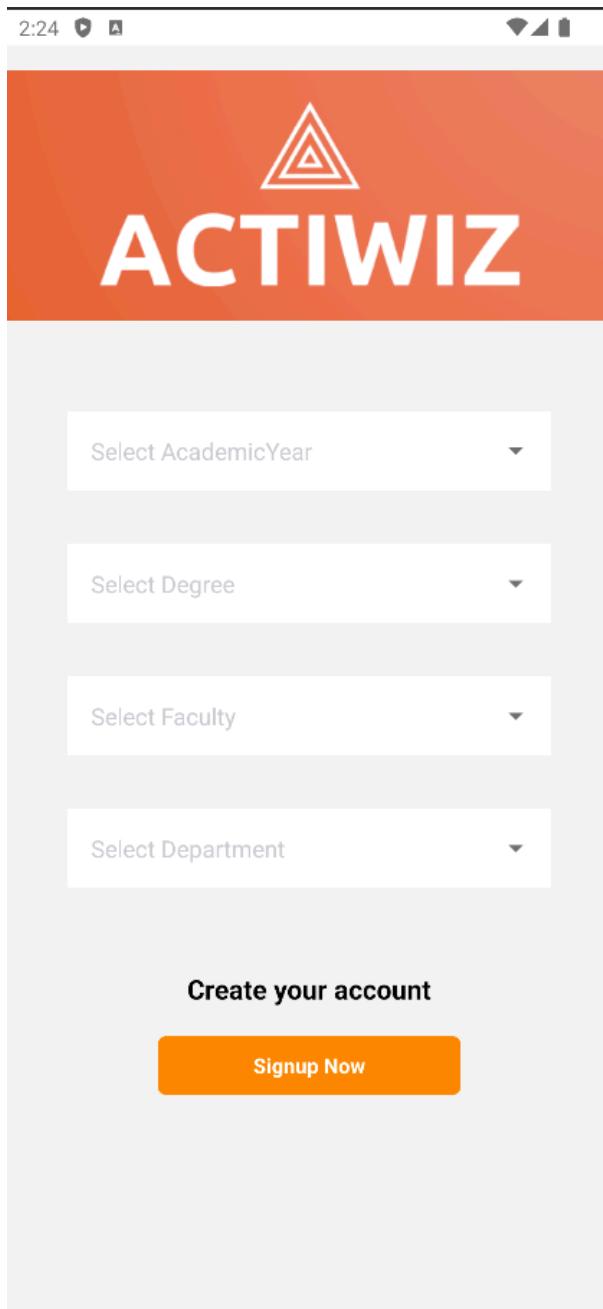
รูปที่ B.3: กรณีผู้ใช้ไม่กรอกรหัสผ่าน

กรณีผู้ใช้ไม่กรอกรหัสผ่าน จะมีการแจ้งเตือนให้ผู้ใช้กรอกรหัสผ่านก่อนที่จะกดเข้าสู่ระบบ



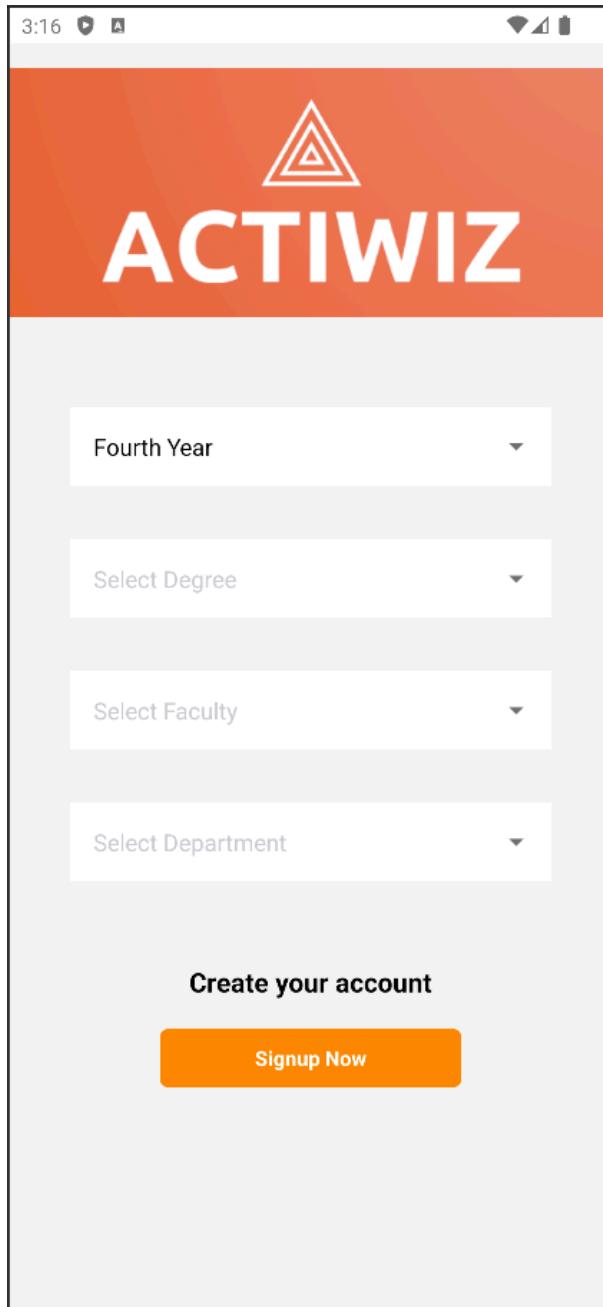
รูปที่ B.4: กรณีผู้ใช้กรอกรหัสผ่านผิด

กรณีผู้ใช้กรอกรหัสผ่านผิด จะมีการแจ้งเตือนให้ผู้ใช้กรอกรหัสผ่านใหม่ก่อนที่จะกดเข้าสู่ระบบ



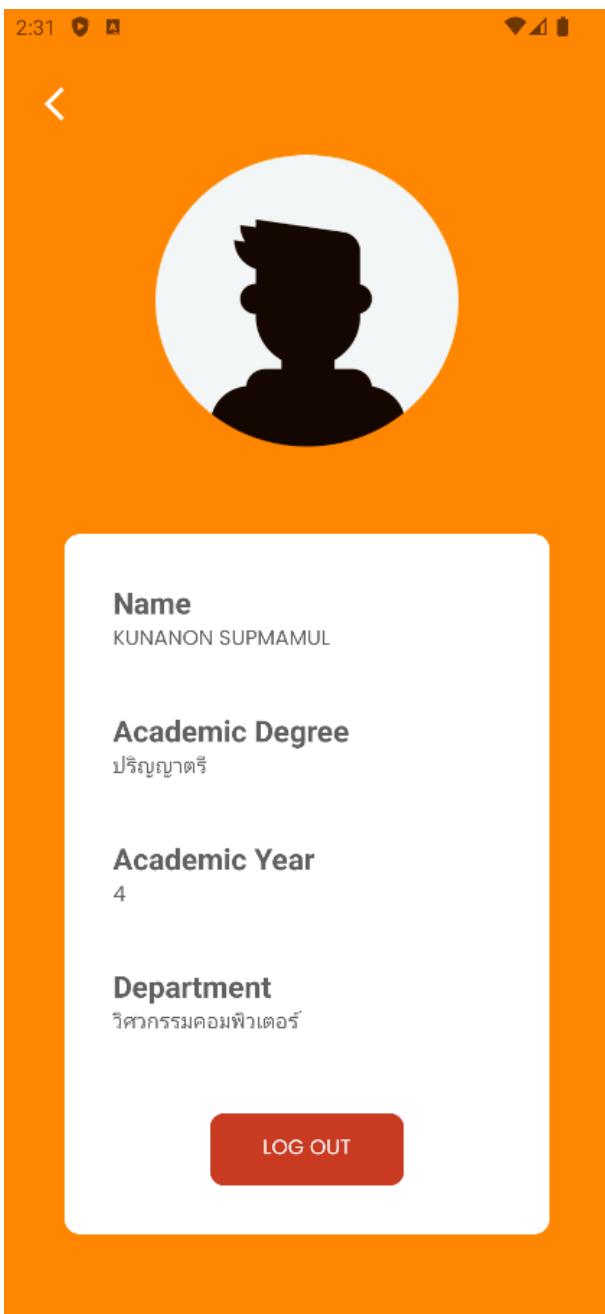
รูปที่ B.5: กรณีที่ไม่มีบัญชีผู้ใช้ในฐานข้อมูล

กรณีที่ไม่มีบัญชีผู้ใช้ในฐานข้อมูล ระบบจะให้ผู้ใช้กรอกข้อมูลส่วนตัวก่อนที่จะเข้าสู่ระบบ



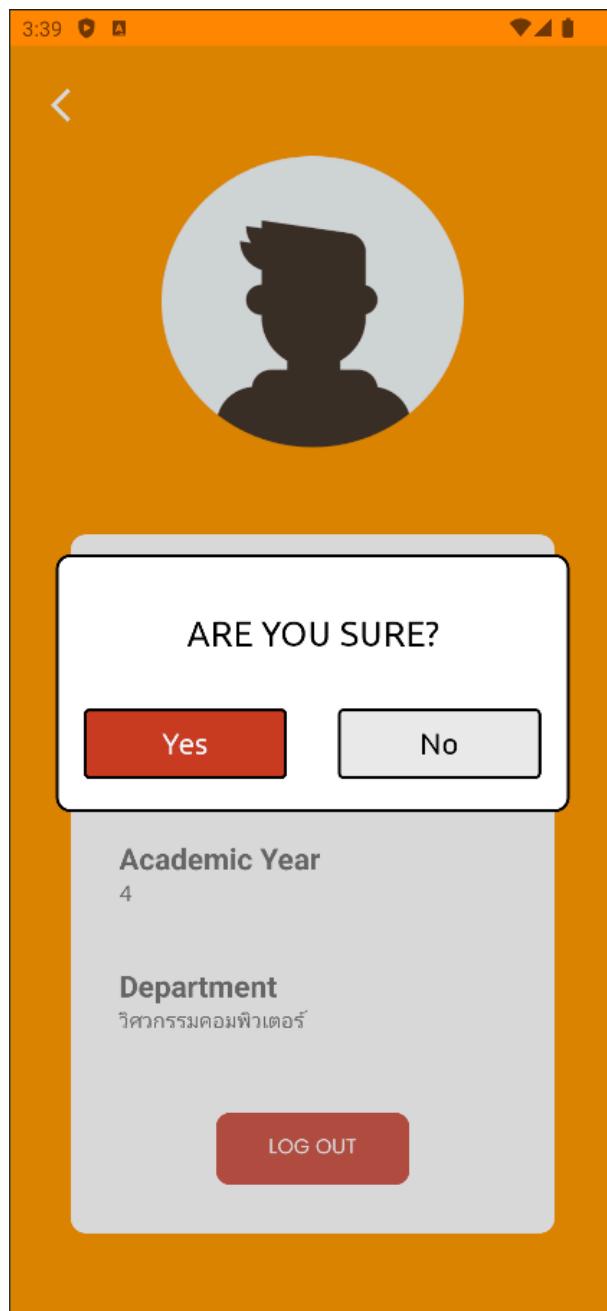
รูปที่ B.6: กรณีผู้ใช้กรอกข้อมูล

กรณีผู้ใช้กรอกข้อมูลไม่ครบ ระบบจะไม่สามารถสร้างบัญชีได้จนกว่าผู้ใช้จะกรอกข้อมูลให้ครบถ้วน



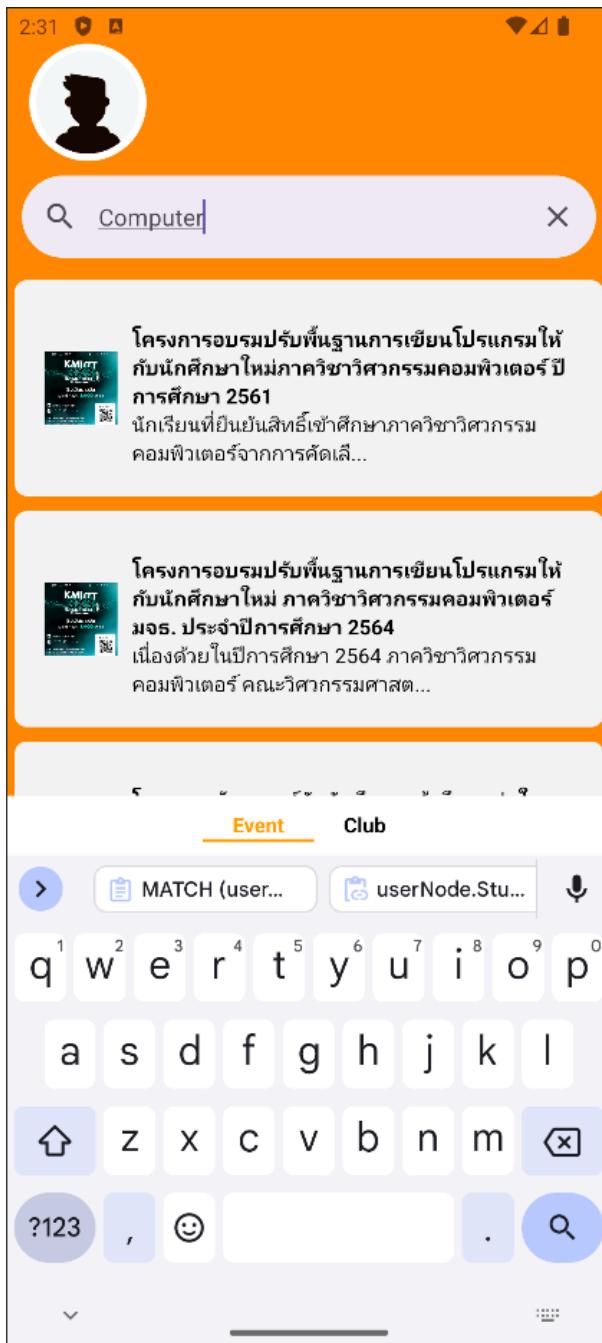
รูปที่ B.7: กรณีผู้ใช้ต้องการออกจากระบบ

กรณีผู้ใช้ต้องการออกจากระบบจะมีการแจ้งเตือนสถานะเมื่นย้อนการออกจากระบบก่อนที่จะออกจากระบบ



รูปที่ B.8: กรณีผู้ใช้กดปุ่ม No หรือ กดนอกกรอบแจ้งเตือน

กรณีผู้ใช้กดปุ่ม No หรือ กดนอกกรอบแจ้งเตือนจะถือว่าเป็นการยกเลิกการออกจากระบบทั้งสองอย่าง

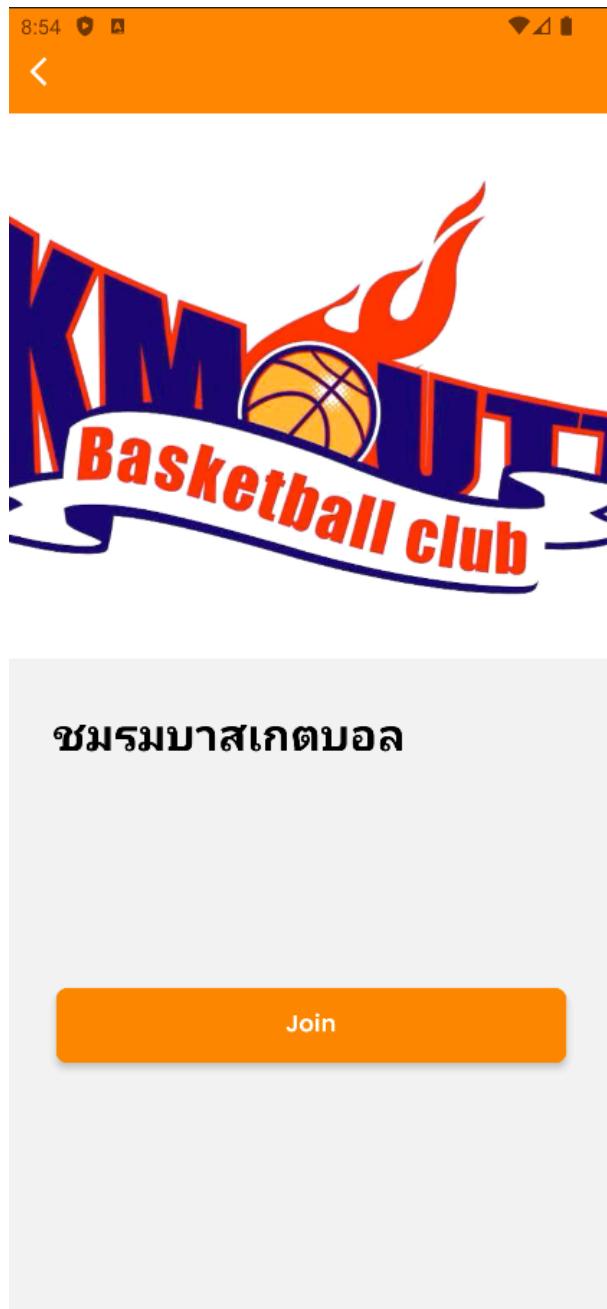


รูปที่ B.9: กรณีที่ผู้ใช้ต้องการค้นหากิจกรรม

กรณีที่ผู้ใช้ต้องการค้นหากิจกรรม จะสามารถหาได้จากແນບค้นหาของแอปพลิเคชัน



รูปที่ B.10: กรณีผู้ใช้ล่องกดเข้าไปดูรายละเอียดของกิจกรรม จะมีการแสดงรายละเอียดของกิจกรรมนั้นๆ



รูปที่ B.11: กรณีผู้ใช้คลองกดเข้าไปปดูรายละเอียดของชมรม

กรณีผู้ใช้คลองกดเข้าไปปดูรายละเอียดของชมรม จะมีการแสดงรายละเอียดของชมรมนั้นๆ

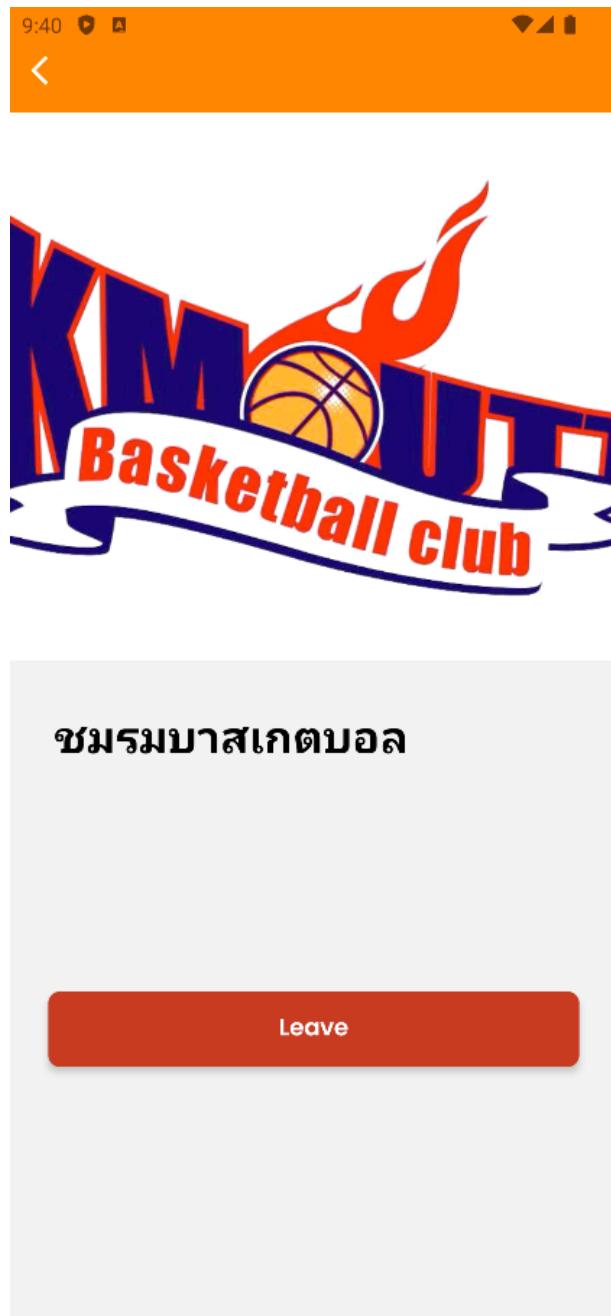
โครงการฝึกอบรมเชิงปฏิบัติการ คอมพิวเตอร์เมืองตัน ครั้งที่ 29

เนื่องจากการศึกษาเป็นปัจจัยสำคัญต่อการประสบความสำเร็จในชีวิต โดยเฉพาะการศึกษาในระดับอุดมศึกษา อย่างไรก็ตาม ในปัจจุบันยังคงมีนักเรียนระดับชั้นมัธยมศึกษาจำนวนมากที่ประสบปัญหาการเข้าศึกษาต่อ ในระดับอุดมศึกษา เพราะนักเรียนขาดช่องมูลและประสบการณ์ที่จำเป็นต่อการตัดสินใจเพื่อเข้าศึกษาต่อในสาขาวิชาที่ตนเองต้องการ เพื่อเป็นแนวทางของการแก้ไขปัญหาดังกล่าว ภาควิชา วิศวกรรมคอมพิวเตอร์จึงมีความต้องการจัดโครงการฝึกอบรมเชิงปฏิบัติการคอมพิวเตอร์เมืองตันครั้งที่ 29 เพื่อเป็นการแนะนำเนื้อหาเรียนระดับชั้นมัธยมศึกษาตอนปลายให้เข้าใจถึงรูปแบบการศึกษาของนักศึกษาภาค วิชาชีวกรรมคอมพิวเตอร์โดยถ่ายทอดประสบการณ์ จากนักศึกษาปัจจุบัน ผ่านกระบวนการบรรยายและเปิดโอกาสให้นักเรียนผู้เข้าร่วมได้ลังมือปฏิบัติจริง ในบทเรียนซึ่ง เป็นพื้นฐานของหลักสูตรภาควิชาชีวกรรมคอมพิวเตอร์ นอกจากนี้โครงการนี้ยังเป็นการประชาสัมพันธ์ภาค วิชาชีวกรรมคอมพิวเตอร์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ด้วยการแสดงศักยภาพของนักศึกษาปัจจุบัน รวมทั้งความพร้อมของอุปกรณ์ เครื่องมือ และห้องปฏิบัติการ เพื่อเป็นตัวเลือกสำหรับนักเรียน ระดับชั้นมัธยมศึกษาตอนปลาย ซึ่งมีความสนใจทางด้าน วิศวกรรมคอมพิวเตอร์



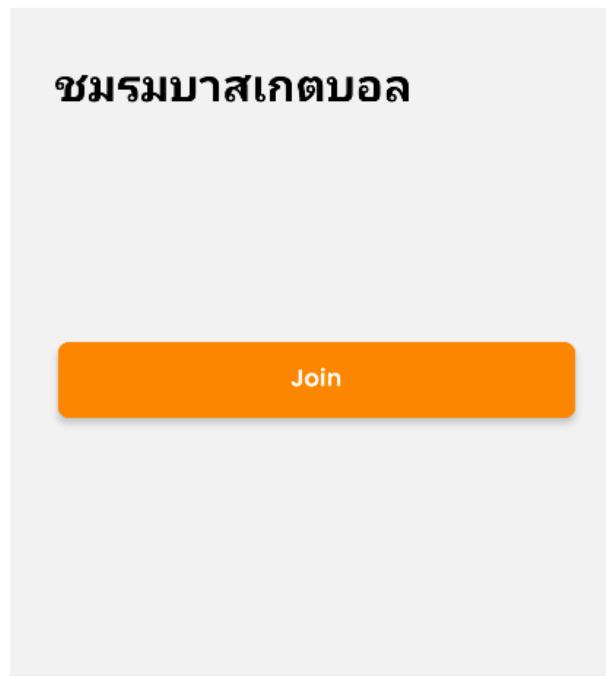
รูปที่ B.12: กรณีผู้ใช้กดเข้าร่วมกิจกรรม

กรณีผู้ใช้กดเข้าร่วมกิจกรรม จะมีการแจ้งเตือนตามยืนยันการเข้าร่วมแก่ผู้ใช้ก่อนเข้าร่วมกิจกรรม



รูปที่ B.13: กรณีผู้ใช้กดเข้าร่วมชัมรม

กรณีผู้ใช้กดเข้าร่วมชัมรม จะมีการแจ้งเตือนถอนยืนยันการเข้าร่วมแก่ผู้ใช้ก่อนเข้าร่วมชัมรม



รูปที่ B.14: กรณีผู้ใช้ยกเลิกการเข้าร่วมชมรม

กรณีผู้ใช้ยกเลิกการเข้าร่วมชมรม จะมีการแจ้งเตือนตามยืนยันการยกเลิกการเข้าร่วมแก่ผู้ใช้ก่อนยกเลิกการเข้าร่วมชมรม



รูปที่ B.15: กรณีผู้ใช้ตรวจสอบการแจ้งเตือนของตัวแอปพลิเคชัน

กรณีผู้ใช้ตรวจสอบการแจ้งเตือนของตัวแอปพลิเคชัน จะมีการแสดงการแจ้งเตือนทั้งหมดที่เกิดขึ้นในแอปพลิเคชัน กรณีผู้ใช้ตรวจสอบกิจกรรมที่ทางระบบแนะนำมาให้ โดยในตัวอย่างนี้จะเป็นการอ้างอิงจากคณะที่ผู้ใช้อยู่



รูปที่ B.16: กรณีผู้ใช้ตรวจสอบกิจกรรมที่ทางระบบแนะนำให้