



CPE 334 Software Engineering Term Project

เรื่อง RABAI (ระบาย)

จัดทำโดย

นายนวิน	โตศิlananนท์	รหัสนักศึกษา 65070501033
นายปรัศกรณ	กิตติขญาพงค์	รหัสนักศึกษา 65070501036
นายรัฐธิพงษ์	สกุลจีน	รหัสนักศึกษา 65070501048
นายภูมิรพี	เมืองน้อย	รหัสนักศึกษา 65070501084
นายกันต์ดนัย	ศรีวัฒนะ	รหัสนักศึกษา 65070507203
นายฐานุวัชร	ธนโชติศิริรัตน์	รหัสนักศึกษา 65070507209
นางสาวฐิตาภรณ์	อันบุตร	รหัสนักศึกษา 65070507210
นายเดชพัฒน์	เอกจิตรระกุล	รหัสนักศึกษา 65070507211

เสนอ

รศ.ดร. ณัฐชา เดชดำรง

ผศ.ดร. ชัยยงค์ รักจิตเวชสกุล

รายงานนี้เป็นส่วนหนึ่งของรายวิชา Software Engineering (CPE 334)

คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ภาคเรียนที่ 1 ปีการศึกษา 2567

คำนำ

รายงานนี้เป็นส่วนหนึ่งของรายวิชา Software Engineering (CPE 334) ซึ่งเป็นการมุ่งเน้นในการพัฒนาทักษะด้านวิศวกรรมซอฟต์แวร์การทำงานนี้ไม่เพียงแต่เป็นการพัฒนาเว็บไซต์หรือแอปพลิเคชันตามข้อกำหนดที่ได้รับแต่ยังเป็นการเรียนรู้การประยุกต์ใช้หลักการวิศวกรรมซอฟต์แวร์การทำงานร่วมกันในทีม และการบริหารจัดการโครงการ โดยการเรียนรู้จากการวางแผน การออกแบบ การพัฒนาและการทดสอบแอปพลิเคชันต่าง ๆ ที่จะช่วยเสริมสร้างทักษะในการทำงานจริงในอุตสาหกรรมซอฟต์แวร์

ในรายงานฉบับนี้นำเสนอรายละเอียดของกระบวนการพัฒนาโครงการตั้งแต่การวางแผน การออกแบบ และการเลือก เทคโนโลยีที่ใช้ในการพัฒนาและการปรับปรุงแอปพลิเคชันให้ตอบสนองต่อความต้องการของผู้ใช้งานอย่างมีประสิทธิภาพ

สารบัญ

หน้า

คำนำ	ก
สารบัญ.....	ข
ที่มาและความสำคัญ	1
วัตถุประสงค์	2
ขอบเขตงาน.....	2
การพัฒนางานโดยใช้แนวทาง Agile ผ่าน Scrum Framework.....	3
Requirement.....	6
การออกแบบและพัฒนา	8
Data Flow diagram	8
Use case diagram	13
Class diagram.....	15
Sequence diagram.....	21
UX/UI Design และ การทำงานของระบบ.....	26
สรุปผลการดำเนินงาน.....	40

ที่มาและความสำคัญ

Rabai (ระบาย) คือ เว็บไซต์ที่เป็นพื้นที่สื่อสารสำหรับนักศึกษาในมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ซึ่งมีวัตถุประสงค์เพื่อให้ผู้ใช้สามารถระบาย แสดงความคิดเห็น แนะนำ พูดคุยปรึกษาแลกเปลี่ยนกัน ทั้งในด้านวิชาที่เรียน เหตุการณ์และกิจกรรมต่างๆ ที่เกิดขึ้นในมหาวิทยาลัย และเข้าร่วมกิจกรรมต่าง ๆ ภายในมหาวิทยาลัยได้อย่างอิสระ โดยฟีเจอร์หลักของเว็บไซต์ประกอบด้วย

- บอร์ดสนทนา (Discussion Board)

เป็นพื้นที่สำหรับผู้ใช้ที่ลงทะเบียนแล้วสามารถสร้างหัวข้อสนทนา แสดงความคิดเห็น และพูดคุยกันในรูปแบบ Post ผู้เยี่ยมชม (Guest) สามารถเข้ามาอ่านบอร์ดสนทนาได้แต่จะไม่สามารถมีส่วนร่วมในการสนทนาได้จนกว่าจะเข้าสู่ระบบ

- ห้องแชทแบบไม่ระบุตัวตน (Anonymous-chat)

ผู้ใช้สามารถสร้างห้องแชทแบบไม่ระบุตัวตนเพื่อพูดคุยกับคนอื่น ๆ โดยไม่ต้องเปิดเผยตัวตน ข้อความจะถูกลบเมื่อห้องแชทถูกปิดและห้องสามารถตั้งชื่อ และ กำหนดจำนวนคนได้ตามที่ต้องการ

- กิจกรรม (Activities)

ฟีเจอร์นี้เปิดโอกาสให้ผู้ใช้สามารถสร้างหรือเข้าร่วมกิจกรรมภายในมหาวิทยาลัย โดยสามารถกำหนดเวลาสถานที่และรายละเอียดต่าง ๆ ของกิจกรรมได้ ทั้งยังสามารถติดตามกิจกรรมยอดนิยมและสถิติการเข้าร่วมของผู้ใช้อื่น ๆ ได้อีกด้วย

วัตถุประสงค์

1. สร้างพื้นที่สำหรับการสื่อสารในมหาวิทยาลัย ทั้งแบบระบุตัวตนและไม่ระบุตัวตน
2. เพื่อศึกษากระบวนการทำงานทางวิศวกรรมซอฟต์แวร์

ขอบเขตงาน

1. การออกแบบการทำงานของเว็บไซต์ด้วย Class Diagrams , Usecase Diagram, Class Diagrams, Sequence Diagrams เป็นต้น
2. การพัฒนาฟีเจอร์หลักของเว็บไซต์
 - 2.1. ระบบสมัครสมาชิกและเข้าสู่ระบบสำหรับผู้ใช้ (User Registration and Login)
 - 2.2. ฟีเจอร์บอร์ดสนทนา (Discussion Board)
 - 2.3. ห้องแชทแบบไม่ระบุตัวตน (Anonymous Chat Room)
 - 2.4. ระบบกิจกรรม (Activities)
สำหรับการสร้างและเข้าร่วมกิจกรรมภายในมหาวิทยาลัย
 - 2.5. ระบบแจ้งเตือน (Notification System)
3. การออกแบบ User Interface และ User Experience ให้เหมาะสมกับนักศึกษา

การพัฒนางานโดยใช้แนวทาง Agile ผ่าน Scrum Framework

ในการพัฒนาเว็บไซต์ Rabai ได้นำ Scrum Agile มาใช้เป็นแนวทางในการบริหารงาน เพื่อให้สามารถวางแผน ควบคุม และสร้างฟีเจอร์ต่าง ๆ ได้อย่างมีประสิทธิภาพ โดย Scrum ช่วยให้ทีมสามารถปรับปรุงและพัฒนาโครงการได้อย่างต่อเนื่องผ่านการทำงานเป็นรอบระยะสั้นที่เรียกว่า Sprint แต่ละ Sprint จะมีการกำหนดเป้าหมายชัดเจน และทบทวนความคืบหน้าหลังสิ้นสุดแต่ละรอบ

ตารางด้านล่างแสดงรายละเอียดการทำงานในแต่ละ Sprint ซึ่งครอบคลุมฟีเจอร์หลักของเว็บไซต์ และการวางแผนการดำเนินงานในแต่ละช่วงเวลา

- เริ่มดำเนินงานเมื่อวันที่ 1 ตุลาคม 2567
- สิ้นสุดการดำเนินงานเมื่อวันที่ 26 พฤศจิกายน 2567

Sprint	ระยะเวลา	เป้าหมาย	งานที่ต้องทำ
1	Week 1-2	ออกแบบเว็บไซต์ สร้างระบบ User Registration and Login และเริ่มพัฒนา Discussion Boards	<ul style="list-style-type: none"> - ออกแบบเว็บไซต์เบื้องต้นทั้งหมด (UX/UI) - สร้างระบบ User Registration and Login (ลงทะเบียน, เข้าสู่ระบบ) - สร้าง Discussion Boards เบื้องต้นที่สามารถอ่านได้ (สำหรับ Guest) - ทดสอบการลงทะเบียนและเข้าสู่ระบบ
2	Week 3-4	พัฒนาฟีเจอร์การสนทนา (Discussion Boards, Anonymous-chat)	<ul style="list-style-type: none"> - เพิ่มฟีเจอร์การโพสต์และแสดงความคิดเห็นใน Discussion Boards

			<ul style="list-style-type: none"> - พัฒนาระบบการค้นหาและกรองหัวข้อสนทนา - สร้างฟีดเจอร์สำหรับผู้ใช้งานที่ลงทะเบียนให้สามารถติดตามกระทู้ที่สนใจ - เริ่มพัฒนาฟีดเจอร์ห้องแชทแบบไม่ระบุตัวตน (Anony-chat)
3	Week 5-6	พัฒนาฟีดเจอร์ห้องแชท (Anonymous-chat) และเพิ่มฟีดเจอร์เพิ่มเติม	<ul style="list-style-type: none"> - เพิ่มฟีดเจอร์การสร้างห้องแชทแบบไม่ระบุตัวตน - พัฒนาการแสดงผลข้อความในห้องแชท และระบบแท็กห้อง - เพิ่มระบบแจ้งเตือนสำหรับการตอบกลับในบอร์ดสนทนา

4	Week 7-8	<p>พัฒนาฟีเจอร์ระบบกิจกรรม (Activity)</p> <p>และปรับปรุงประสิทธิภาพ</p>	<ul style="list-style-type: none"> - พัฒนาฟีเจอร์กิจกรรม (สร้างกิจกรรม,รายละเอียดกิจกรรม) - เพิ่มการตั้งเวลาและสถานที่สำหรับกิจกรรม - เพิ่มระบบแจ้งเตือนสำหรับการเข้าร่วมกิจกรรม - ทดสอบการทำงานของห้องแชทและกิจกรรม - พัฒนาการจัดการโปรไฟล์ผู้ใช้ (แก้ไขข้อมูลส่วนตัว, การตั้งค่าการแจ้งเตือน) - ทดสอบและปรับปรุง UX/UI รวมถึงการทดสอบการใช้งานจริงของผู้ใช้
---	----------	---	---

Requirements

การเก็บ Requirements เพื่อรวบรวมข้อมูลความต้องการของผู้ใช้งานระบบและทำความเข้าใจปัญหาและความคาดหวังจากผู้ใช้งาน มีการใช้วิธีการสัมภาษณ์ (Interview) ซึ่งจะเป็นการเก็บข้อมูลเชิงลึกจากนักศึกษา มจร. ที่มีปัญหาร่วมกัน และแสดงความคิดเห็นต่อการแก้ปัญหาในมุมมองต่าง ซึ่งอาจจะมีการตั้งคำถามที่เน้นไปที่ปัญหาและวิธีการแก้ปัญหา เช่น “ในการลงวิชาเลือกตัดสินใจจากอะไร และคิดว่าจะมีวิธีการใดในการแก้ปัญหานี้” หลังจากได้ข้อมูลเหล่านี้แล้วสามารถนำไปเปลี่ยนเป็น Functional และ Non-Functional Requirements ได้ดังนี้

1. Functional Requirements

- 1.1. ระบบสมัครสมาชิกและเข้าสู่ระบบ (User Registration and Login System)
 - 1.1.1. ผู้ใช้สามารถสมัครสมาชิกและเข้าสู่ระบบเพื่อใช้งานฟีเจอร์หลัก
 - 1.1.2. Guest สามารถดูบอร์ดสนทนา แต่ไม่สามารถโพสต์หรือแสดงความคิดเห็นได้
- 1.2. บอร์ดสนทนา (Discussion Boards)
 - 1.2.1. ผู้ใช้ที่ลงทะเบียนสามารถสร้างโพสต์ แสดงความคิดเห็น
 - 1.2.2. ระบบรองรับการค้นหาและกรองโพสต์
 - 1.2.3. Guest สามารถดูโพสต์ได้แต่ไม่สามารถโต้ตอบ
- 1.3. แชทแบบไม่ระบุตัวตน (Anonymous Chat)
 - 1.3.1. ผู้ใช้สามารถสร้างห้องแชทและกำหนดจำนวนผู้เข้าร่วม
 - 1.3.2. ผู้เข้าร่วมแชททุกคนต้องเข้าร่วมในโปรไฟล์ Anonymous
 - 1.3.3. ห้องแชทจะถูกลบเมื่อไม่มีใครอยู่ในห้อง และข้อความจะไม่ถูกบันทึก
- 1.4. กิจกรรม (Activities)
 - 1.4.1. ผู้ใช้สามารถสร้างหรือเข้าร่วมกิจกรรม
 - 1.4.2. สามารถตั้งค่าเวลา สถานที่ และรายละเอียดกิจกรรม
 - 1.4.3. ผู้ใช้สามารถค้นหาและกรองกิจกรรมตามแท็ก
- 1.5. ระบบแจ้งเตือน (Notification System)
 - 1.5.1. แจ้งเตือนผู้ใช้เมื่อมีการตอบกลับใน Discussion Boards
 - 1.5.2. แจ้งเตือนผู้ใช้เมื่อมีการเข้าร่วมกิจกรรมที่สร้างขึ้น

1.6. การจัดการโปรไฟล์ผู้ใช้ (User Profile Management)

1.6.1. ผู้ใช้สามารถอัปเดตข้อมูลส่วนตัว เช่น ชื่อ อีเมล รูปโปรไฟล์

2. Non-Functional Requirements

2.1. Usability

2.1.1. อินเทอร์เฟซต้องเป็นมิตรกับผู้ใช้ และรองรับการใช้งานบนอุปกรณ์ต่างๆ (Responsive Design)

2.2. Compatibility

2.2.1. ระบบต้องสามารถทำงานบนเบราว์เซอร์หลักๆ เช่น Chrome, Firefox, Safari

2.3. Security

2.3.1. ระบบต้องเข้ารหัส (encrypt) ข้อมูลผู้ใช้งานอย่างปลอดภัย

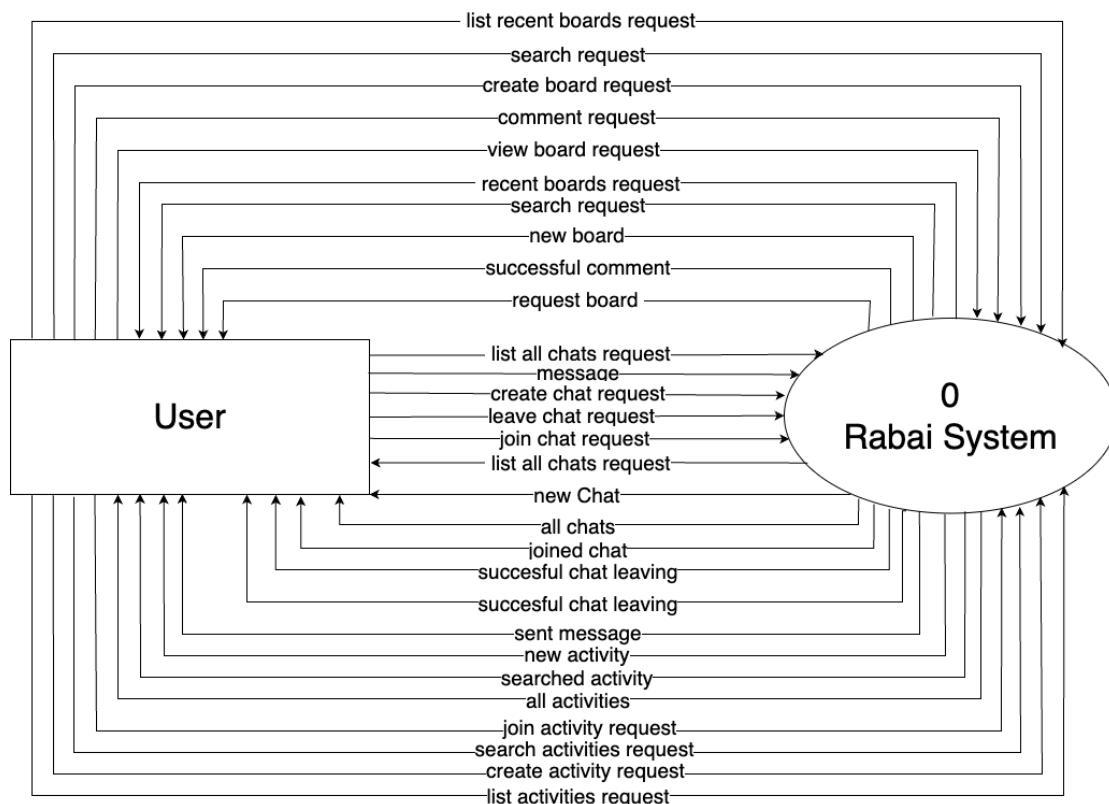
2.3.2. ห้องเซิร์ฟเวอร์ไม่ระบุตัวตนต้องไม่เปิดเผยข้อมูลผู้ใช้

การออกแบบและพัฒนาเว็บไซต์

ในการพัฒนา Rabai เว็บไซต์มีการวางโครงสร้างและออกแบบระบบต่างโดยใช้หลักและวิธีการทางวิศวกรรมซอฟต์แวร์โดยใช้เครื่องมือและวิธีการดังนี้

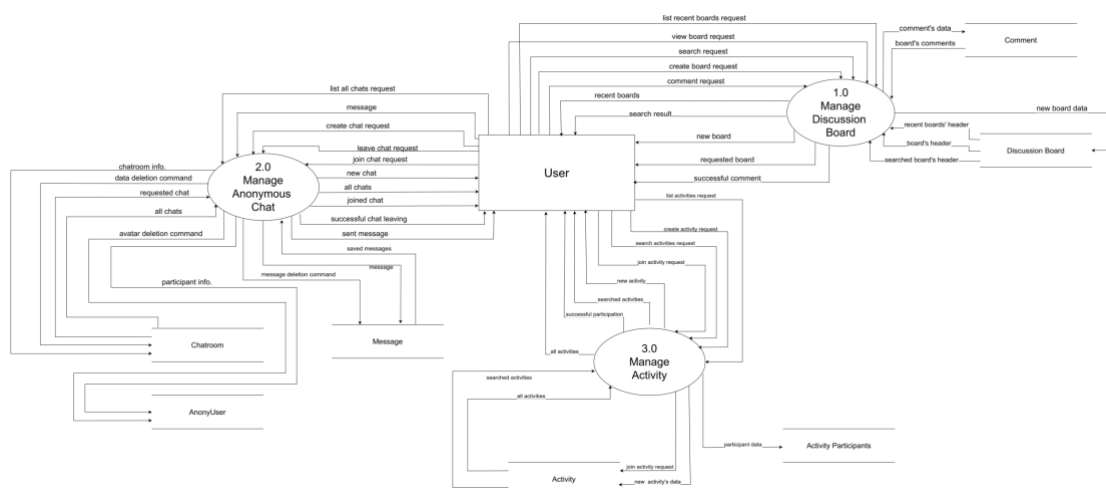
Data Flow diagram

- Context diagram



Context Diagram นี้แสดงภาพรวมของการไหลของข้อมูลในระบบ โดยที่ไม่เจาะลึกกระบวนการภายใน แต่จะแสดงให้เห็นการเชื่อมโยงระหว่าง External Entity นั่นคือ User และ Rabai System รวมถึง Data Flows ที่แสดงข้อมูลที่ผู้ใช้งาน (User) ส่งไปหา ระบบ (System) และข้อมูลที่ระบบส่งกลับไปหาผู้ใช้งาน

- Level 1 DFD



จาก Level 1 DFD เราแยก Rabai System ออกเป็น 3 process หลัก ประกอบไปด้วย

- 1.0 Manage Discussion board

แสดงการส่งข้อมูลโดยรวมของตัวระบบ Discussion Board โดยจะมีการส่ง Request ไปให้ระบบได้ทั้งหมด 5 Request ได้แก่ ขอดู Board ทั้งหมดในระบบ, ขอดู Board อย่างละเอียด, ทำการ Search หา Board ที่ตาม keyword ที่สนใจ, ขอสร้าง Board ใหม่ขึ้น และทำการแสดงความคิดเห็นลงใน Board จากนั้นระบบ Discussion Board ก็จะส่ง Board ทั้งหมด, Board โดยละเอียดที่แสดง comment ด้วย, Board ที่ได้จากการ Search, Board ที่ถูกสร้างขึ้นมาใหม่ และความคิดเห็นที่ส่งไป ขึ้นมาแสดง

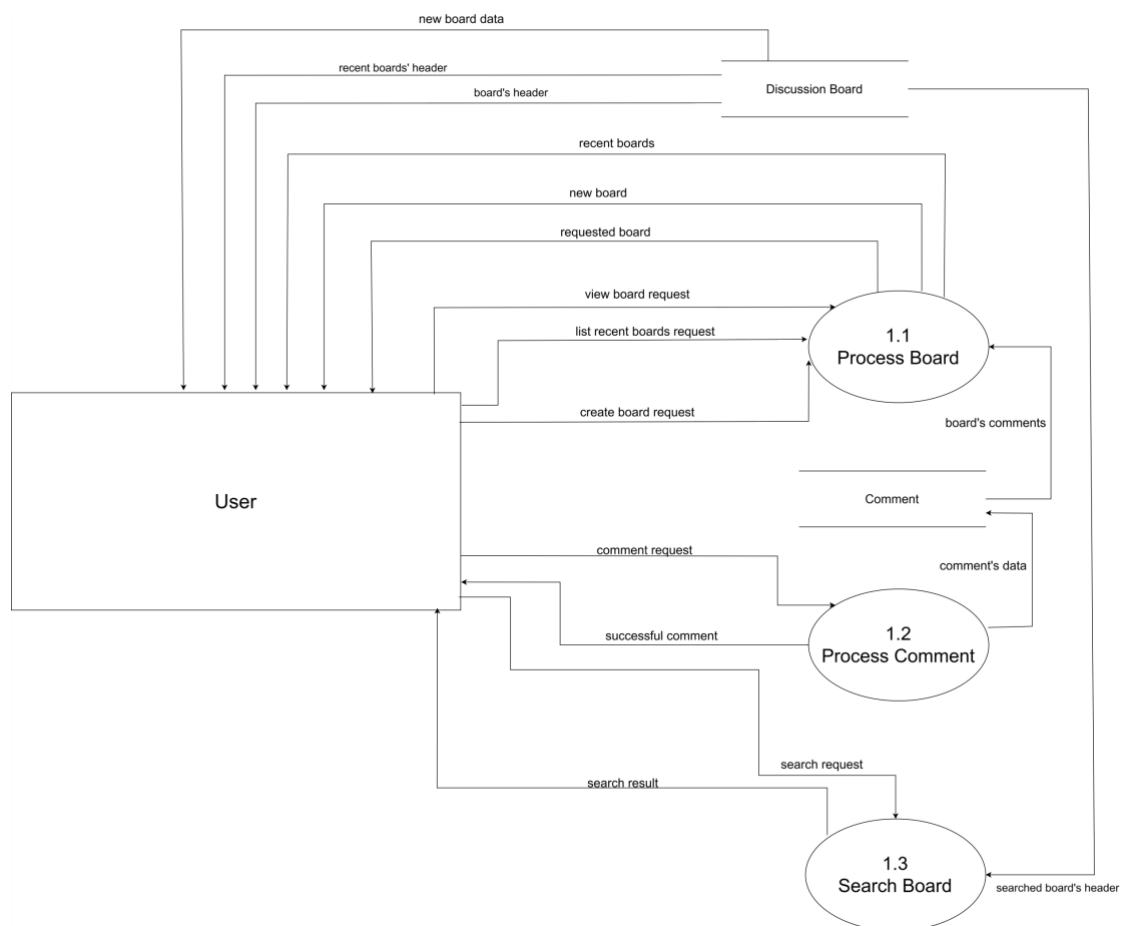
- 2.0 Manage Anonymous Chat

แสดงการส่งข้อมูลโดยรวมของตัวระบบ Anonymous Chat โดยจะมีการส่ง Request ไปให้ระบบได้ทั้งหมด 5 Request ได้แก่ ขอดูห้องแชททั้งหมดในระบบ, เข้าร่วมห้องแชท, ขอสร้างห้องแชทใหม่ และทำการส่งข้อความในห้องแชท จากนั้นระบบ Anonymous ก็จะส่งห้องแชททั้งหมด, ห้องแชทใหม่ และห้องแชทที่เข้าร่วมให้กับ User และจะส่งการออกจากห้องแชทและข้อความที่ต้องการส่ง ขึ้นมาแสดงให้กับ AnonyUser

- 3.0 Manage Activity

แสดงการส่งข้อมูลโดยรวมของตัวระบบ Activities โดยจะมีการส่ง Request ไปให้ระบบได้ทั้งหมด 4 Request ได้แก่ การสร้าง Activities, การค้นหา Activities, การเข้าร่วม Activities และการดูรายการ Activities จากนั้นระบบ Activities ก็จะตรวจสอบรายละเอียดของ User จากนั้นก็จะส่งคำยืนยันการสร้าง Activities สำเร็จ, ผลลัพธ์การค้นหา Activities, ยืนยันการเข้าร่วม Activities สำเร็จ และรายการทั้งหมดของ Activities ขึ้นมาแสดง

- Level 2 DFD

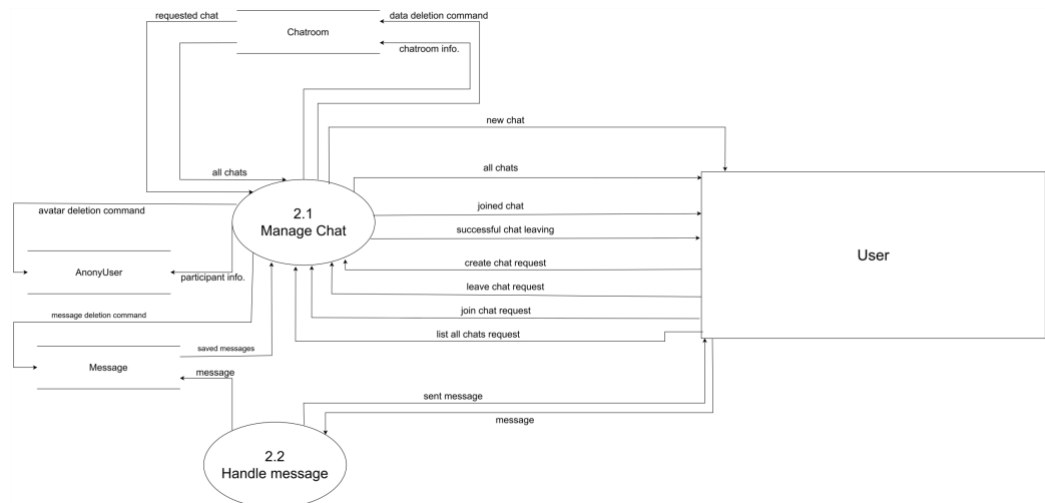


ใน Level 2 Diagram จะแสดง Process ต่างๆ ภายในระบบ Discussion Board ซึ่งจะประกอบไปด้วย

- 1.1 Process board

ทำหน้าที่จัดการทุกอย่างที่เกี่ยวกับ Board ได้แก่ การแสดง Board ทั้งหมด, การแสดง Board โดยละเอียด และการสร้าง Board ใหม่

- 1.2 Process comment : ทำหน้าที่รับ comment จาก User และบันทึก comment ลงใน database ที่เก็บข้อมูล comment
- 1.3 Search board : ทำหน้าที่เกี่ยวกับการ search และ filter ข้อมูล board และส่งกลับมาแสดงที่ฝั่ง User



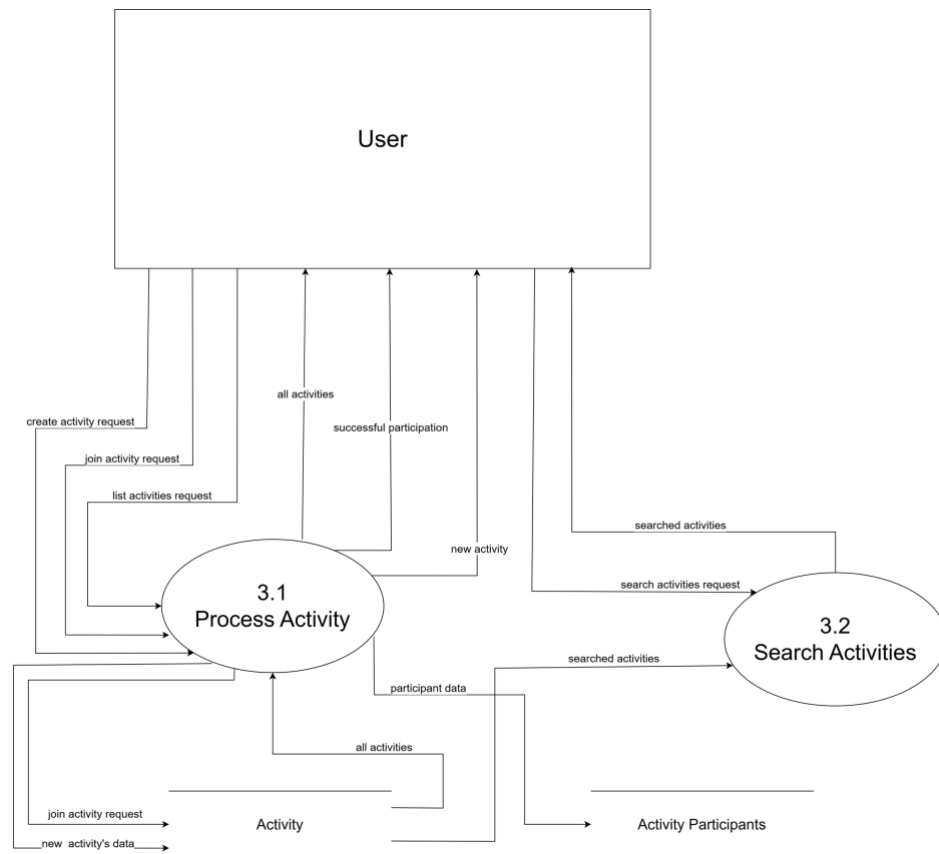
ใน Level 2 Diagram จะแสดง Process ต่างๆ ภายในระบบ Anonymous Chat ซึ่งจะประกอบไปด้วย

- 2.1. Manage Chat

ทำหน้าที่จัดการทุกอย่างที่เกี่ยวกับห้องแชท ได้แก่ การแสดงห้องแชททั้งหมด, การเพิ่มผู้เข้าร่วมแชท, การสร้างห้องแชทใหม่ และการลบผู้เข้าร่วมออกจากห้องแชท

- 2.2. Handle message

ทำหน้าที่รับข้อความที่ AnonyUser ต้องการส่ง และบันทึกข้อความลงใน Database จากนั้นจึงส่งข้อความกลับไปแสดงบนห้องแชท



ใน Level 2 Diagram จะแสดง Process ต่างๆ ภายในระบบ Activities ซึ่งจะประกอบไปด้วย

- 3.1. Create Activity

ทำหน้าที่สร้างกิจกรรมใหม่ จากนั้นทำการบันทึก Activity ลงใน database ที่เก็บข้อมูล Activity และแจ้งคำยืนยันสร้าง Activity สำเร็จให้ User

- 3.2. Search Activities

ทำหน้าที่ค้นหากิจกรรมตามความต้องการของ User โดยดึงข้อมูล Activity มาจาก database

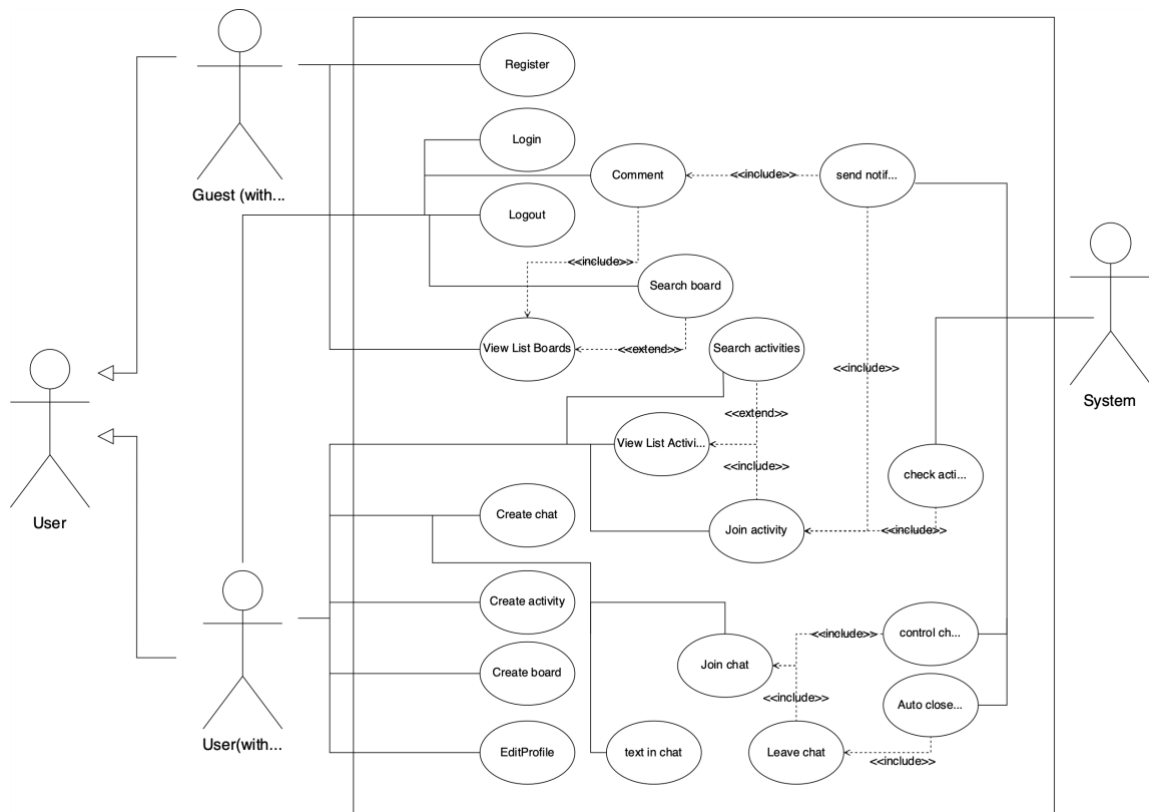
3. Join Activity

ทำหน้าที่ส่ง request ขอเข้าร่วมกิจกรรม จากนั้นทำการบันทึก participants data ลงใน database ที่เก็บข้อมูล activity participants และแจ้งคำยืนยันผู้เข้าร่วมสำเร็จไปยัง User

4. List Activities

ทำหน้าที่แสดง Activity ทั้งหมด โดยดึงข้อมูล Activity มาจาก

Use case diagram



สำหรับ Use case diagram มีการออกแบบโดยมี Guest User และ System ซึ่ง Guest และ User จะมีการทำงานร่วมกันกับ System ดังนี้

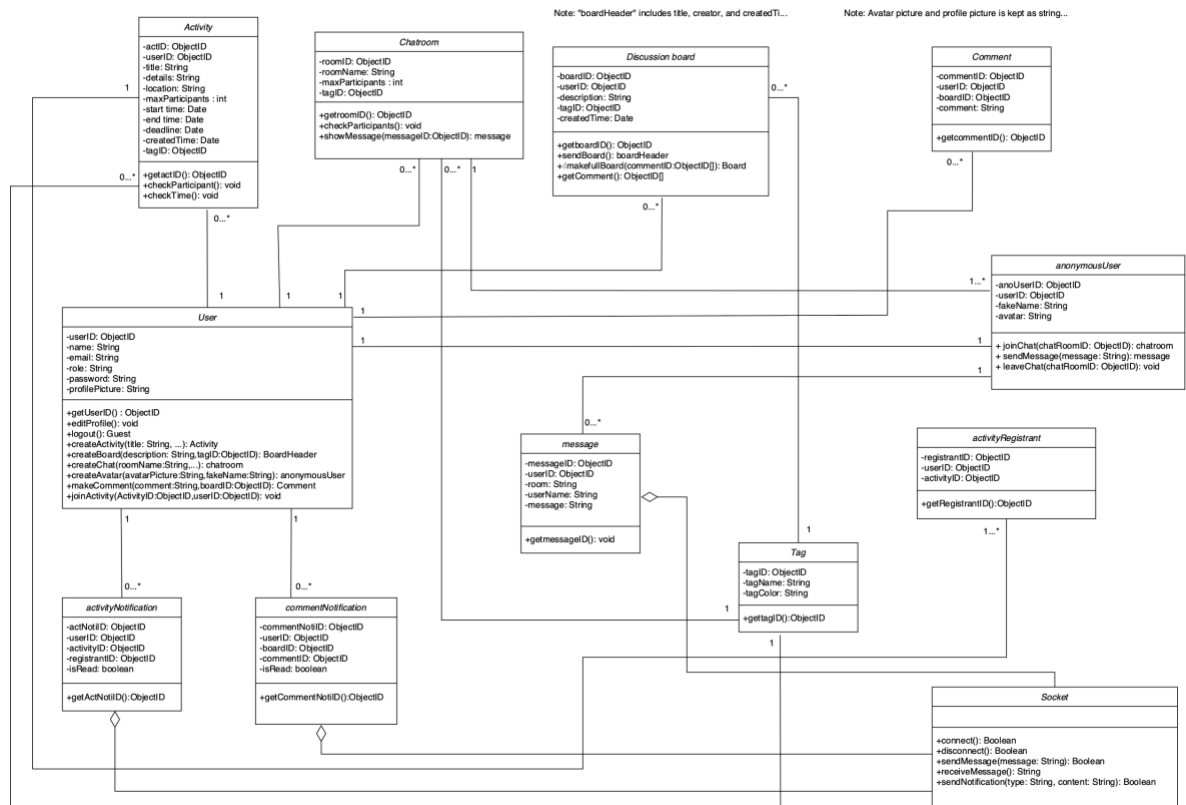
1. Guest (with out member) คือผู้ใช้ที่ไม่มีบัญชีในระบบ
 - 1.1. Register: ผู้ใช้ที่ไม่มีบัญชีในระบบสามารถลงทะเบียนได้
 - 1.2. View List Board: ผู้ใช้ที่ไม่มีบัญชีในระบบสามารถใช้ระบบนี้เพื่อเรียกดูโพสต์ Discussion Board ได้ (ไม่สามารถเรียกใช้ฟังก์ชันอื่น ๆ ที่เกี่ยวข้องได้เช่น Comment)
 - 1.3. Search Board: ผู้ใช้ที่ไม่มีบัญชีในระบบสามารถใช้ระบบ Serch Board ซึ่งระบบนี้ใช้สำหรับค้นหาและกรองโพสต์

2. User (with member) คือผู้ใช้ที่มีบัญชีในระบบ
 - 2.1. Login: ฟังก์ชันสำหรับผู้ใช้ที่มีบัญชีอยู่แล้วเพื่อเข้าสู่ระบบ
 - 2.2. Logout: ฟังก์ชันสำหรับผู้ใช้ออกจากระบบเมื่อใช้งานเสร็จ
 - 2.3. View List Board: ฟังก์ชันสำหรับเรียกดูโพสต์ Discussion Board
 - 2.4. Search Board: ฟังก์ชันสำหรับค้นหาและกรองโพสต์
 - 2.5. Comment:

ฟังก์ชันที่เปิดให้ผู้ใช้สามารถแสดงความคิดเห็นในกระทู้หรือเนื้อหาในระบบได้มีการเรียกใช้ฟังก์ชัน send notification เพื่อแจ้งเตือนเมื่อมีความคิดเห็นใหม่
 - 2.6. Create Board: ฟังก์ชันสำหรับผู้ใช้ในการสร้างกระทู้ใหม่
 - 2.7. Create Chat: ฟังก์ชันสำหรับสร้างห้องแชทใหม่
 - 2.8. Join Chat: ฟังก์ชันสำหรับเข้าห้องใช้ใหม่ มีการเรียกใช้ฟังก์ชัน Control chat participants และ LeaveChat
 - 2.9. Text in Chat: ฟังก์ชันสำหรับการส่งข้อความในห้องแชท
 - 2.10. Leave Chat: ฟังก์ชันสำหรับออกจากห้องแชท มีการเรียกใช้ฟังก์ชัน Auto close chat เพื่อตรวจสอบจำนวนผู้ใช้งานและปิดห้องแชทอัตโนมัติเมื่อไม่มีคนอยู่
 - 2.11. View List Activity: ฟังก์ชันสำหรับเรียกดูรายการ Activity
 - 2.12. Search Activity: ฟังก์ชันสำหรับค้นหากิจกรรมต่างๆ ในระบบ
 - 2.13. Create Activity: ฟังก์ชันสำหรับสร้างกิจกรรมใหม่ให้ผู้อื่นสามารถเข้าร่วมได้
 - 2.14. Join Activity: ฟังก์ชันที่ให้ผู้ใช้งานเข้าร่วมกิจกรรม มีการเรียกใช้ฟังก์ชัน check activity participants เพื่อเช็คจำนวนผู้เข้าร่วม
3. System ส่วนของระบบที่จะจัดการอัตโนมัติ
 - 3.1. Send Notification: ฟังก์ชันสำหรับส่งการแจ้งเตือนไปยังผู้อื่น

เมื่อมีกิจกรรมใหม่หรือความคิดเห็นใหม่
 - 3.2. Check activity participants: ฟังก์ชันสำหรับตรวจสอบจำนวนผู้เข้าร่วมกิจกรรม
 - 3.3. Control chat participants: ฟังก์ชันสำหรับควบคุมจำนวนผู้เข้าร่วมในห้องแชท และจัดการสิทธิ์ต่างๆ
 - 3.4. Auto close chat: ปิดห้องแชทอัตโนมัติเมื่อไม่มีผู้ใช้งาน หรือถึงเงื่อนไขที่กำหนด

Class diagram



Class diagram นี้แสดงให้เห็นถึงโครงสร้างของเว็บไซต์ RABAI ซึ่งประกอบไปด้วย

1. Class: User

1.1. Attribute:

- userID เก็บรหัสของผู้ใช้งาน
- name เก็บชื่อของผู้ใช้งาน
- email เก็บอีเมลของผู้ใช้งาน
- role เก็บประเภทของผู้ใช้งาน
- password เก็บรหัสผ่านของผู้ใช้งานที่ถูกเข้ารหัส (Hashed Password)
- profilePicture เก็บที่อยู่ของรูปภาพโปรไฟล์ของผู้ใช้งาน

1.2. Method:

- getUserID() ดึงข้อมูลของผู้ใช้งานจากรหัสผู้ใช้งาน
- editProfile() อัปเดตข้อมูลโปรไฟล์ของผู้ใช้งานใหม่
- logout() ยกเลิกสถานะการเข้าสู่ระบบของผู้ใช้

- createActivity() สร้าง Activity
- createBoard() สร้าง Discussion board
- createChat() สร้างห้องแชท Anonymous-chat
- createAvatar() สร้างอวตารของผู้ใช้งานเพื่อใช้กับ Anonymous-chat
- makeComment() เพิ่มความคิดเห็นใหม่ใน Discussion board
- joinActivity() เข้าร่วมกิจกรรมที่มีอยู่

2. Class: Activity

2.1. Attribute:

- actID เก็บรหัสของกิจกรรม
- userID เก็บรหัสของผู้ใช้งานที่สร้างกิจกรรม
- title เก็บชื่อของกิจกรรม
- details เก็บรายละเอียดของกิจกรรม
- location เก็บสถานที่ที่จัดกิจกรรม
- maxParticipant เก็บจำนวนสูงสุดของผู้เข้าร่วมกิจกรรม
- start time เก็บเวลาที่เริ่มกิจกรรม
- end time เก็บเวลาที่สิ้นสุดกิจกรรม
- deadline เก็บเวลาสิ้นสุดการเข้าร่วมกิจกรรม
- createTime เก็บเวลาที่สร้างกิจกรรม
- tagID เก็บรหัสแท็กของกิจกรรม

2.2. Method:

- getactID() ดึงข้อมูลกิจกรรมจากรหัสของกิจกรรม
- checkParticipant() ตรวจสอบจำนวนผู้เข้าร่วมกิจกรรม
- checkTime() ตรวจสอบเวลาเริ่มต้นและเวลาสิ้นสุดกิจกรรม

3. Class: activityRegistrant

3.1. Attribute:

- registrantID เก็บรหัสของการระบุผู้เข้าร่วมกิจกรรม
- userID เก็บรหัสของผู้ใช้งานที่ลงทะเบียนเข้าร่วมกิจกรรม
- activityID เก็บรหัสของกิจกรรมที่ผู้ใช้ได้ลงทะเบียนเข้าร่วม

3.2. Method:

- getRegistrantID() ดึงข้อมูลผู้ลงทะเบียนจากรหัสของการลงทะเบียน

4. Class: ActivityNotification

4.1. Attribute:

- actNotiID เก็บรหัสของการแจ้งเตือนการเข้าร่วมกิจกรรม
- userID เก็บรหัสของผู้ใช้งานที่ได้รับการแจ้งเตือน
- activityID เก็บรหัสของกิจกรรมที่ผู้ใช้งานได้สร้างไว้
- registrantID เก็บรหัสของการระบุผู้เข้าร่วมกิจกรรม
- isRead เก็บสถานะการอ่าน

4.2. Method:

- getActNotiID()

ดึงข้อมูลการแจ้งเตือนการเข้าร่วมกิจกรรมจากรหัสของการแจ้งเตือนการเข้าร่วมกิจกรรม

5. Class: Discussion board

5.1. Attribute:

- boardID เก็บรหัสของกระดานสนทนา
- userID เก็บรหัสของผู้ใช้งานที่สร้างกระดานสนทนา
- description เก็บเนื้อหาในกระดานสนทนา
- tagID เก็บรหัสของแท็กของกระดานสนทนา
- createTime เก็บเวลาที่สร้างกระดานสนทนา

5.2. Method:

- getboardID() ดึงข้อมูลกระดานสนทนาจากรหัสของกระดานสนทนา
- sendBoard() ส่งข้อมูลของการดานสนทนา
- makefullBoard() ส่งข้อมูลของการดานสนทณารวมกับความคิดเห็น
- getComment() ดึงข้อมูลรหัสความคิดเห็นที่เกี่ยวข้องกับกระดานสนทนา

6. Class: Comment

6.1. Attribute:

- commentID เก็บรหัสของความคิดเห็น
- userID เก็บรหัสของผู้ใช้งานที่แสดงความคิดเห็น

- boardID เก็บรหัสของกระดานสนทนาที่ถูกแสดงความคิดเห็น
- comment เก็บข้อความความคิดเห็น

6.2. Method:

- getcommentID() ดึงข้อมูลการแสดงความคิดเห็นจากรหัสของความคิดเห็น

7. Class: CommentNotification

7.1. Attribute:

- commentNotiID เก็บรหัสของการแจ้งเตือนความคิดเห็น
- userID เก็บรหัสของผู้ใช้งานที่ได้รับการแจ้งเตือน
- boardID เก็บรหัสของกระดานสนทนาที่ผู้ใช้งานได้สร้างไว้
- commentID เก็บข้อความความคิดเห็น
- isRead เก็บสถานะการอ่าน

7.2. Method:

- getCommentNotiID()
ดึงข้อมูลการแจ้งเตือนความคิดเห็นจากรหัสของการแจ้งเตือนความคิดเห็น

8. Class: Chatroom

8.1. Attribute:

- roomID เก็บรหัสของห้องสนทนา
- roomName เก็บชื่อของห้องสนทนา
- maxParticipants เก็บจำนวนผู้เข้าร่วมสูงสุดของห้องสนทนา
- tagID เก็บรหัสแท็กของห้องสนทนา

8.2. Method:

- getroomId() ดึงข้อมูลของห้องสนทนาจากรหัสของห้องสนทนา
- checkParticipants() ตรวจสอบจำนวนคนในห้องสนทนา
- showMessage() แสดงข้อความในห้องสนทนา

9. Class: AnonymousUser

9.1. Attribute:

- anoUserID เก็บรหัสของผู้ใช้ที่ไม่ระบุตัวตน
- userID เก็บรหัสของผู้ใช้ที่สร้างผู้ใช้ที่ไม่ระบุตัวตน

- fakeName เก็บชื่อผู้ใช้ที่ไม่ระบุตัวตน
- avatar เก็บที่อยู่ของรูปภาพโปรไฟล์ของผู้ใช้ที่ไม่ระบุตัวตน

9.2. Method:

- joinChat() เข้าร่วมห้องแชท
- sendMessage() ส่งข้อความไปยังห้องแชท
- leaveChat() ออกจากห้องแชท

10. Class: Message

10.1. Attribute:

- messageId เก็บรหัสของข้อความสนทนา
- userID เก็บรหัสของผู้ใช้ที่ส่งข้อความสนทนา
- room เก็บชื่อห้องสนทนาที่ผู้ใช้ส่งข้อความ
- userName เก็บชื่อของผู้ใช้ที่ไม่ระบุตัวตนที่ผู้ใช้เป็นคนสร้าง
- message เก็บข้อความสนทนา

10.2. Method:

- getMessageID() ดึงข้อมูลข้อความสนทนาจากรหัสของข้อความสนทนา

11. Class: Message

11.1. Attribute:

- tagID เก็บรหัสของแท็ก
- tagName เก็บชื่อของแท็ก
- tagColor: เก็บรหัสสีของแท็ก (Hex color code)

11.2. Method:

- getTagID() ดึงข้อมูลของแท็กจากรหัสของแท็ก

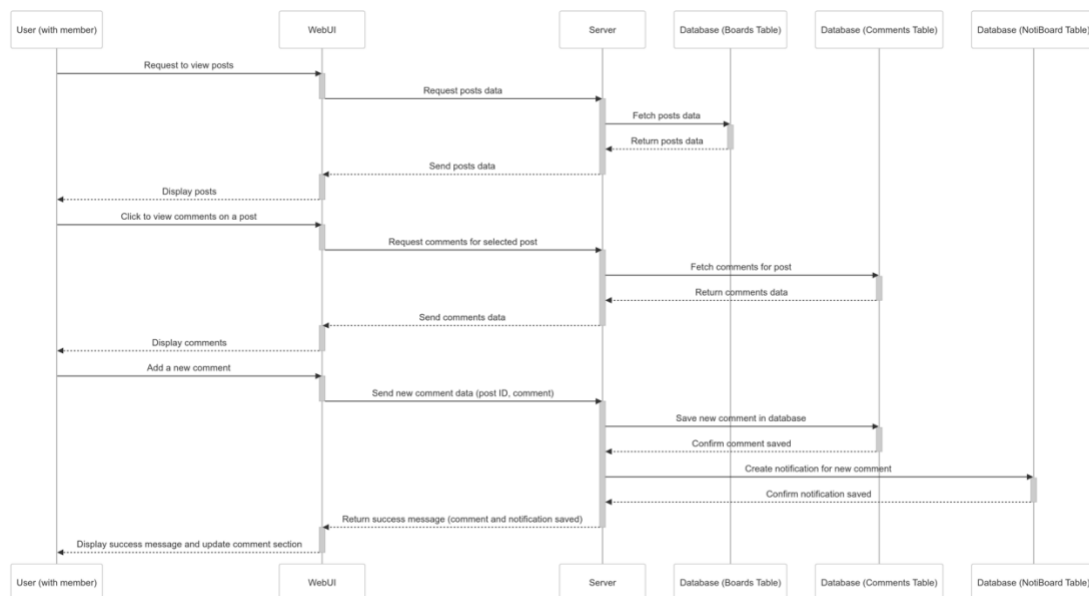
12. Class: Socker

12.1. Method:

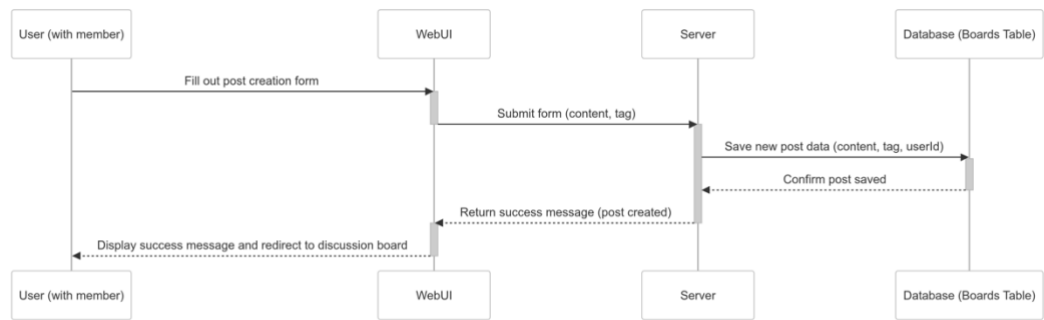
- connect ใช้สำหรับการเชื่อมต่อ socket
- disconnect ใช้สำหรับเมื่อเลิกการเชื่อมต่อ
- sendMessage ใช้สำหรับการส่งข้อความเข้า socket
- receiveMessage ใช้สำหรับการรอรับข้อความจาก socket
- sendNotification ใช้สำหรับการส่งข้อความเข้า socket

Sequence diagram

1. ระบบ DiscussionBoard

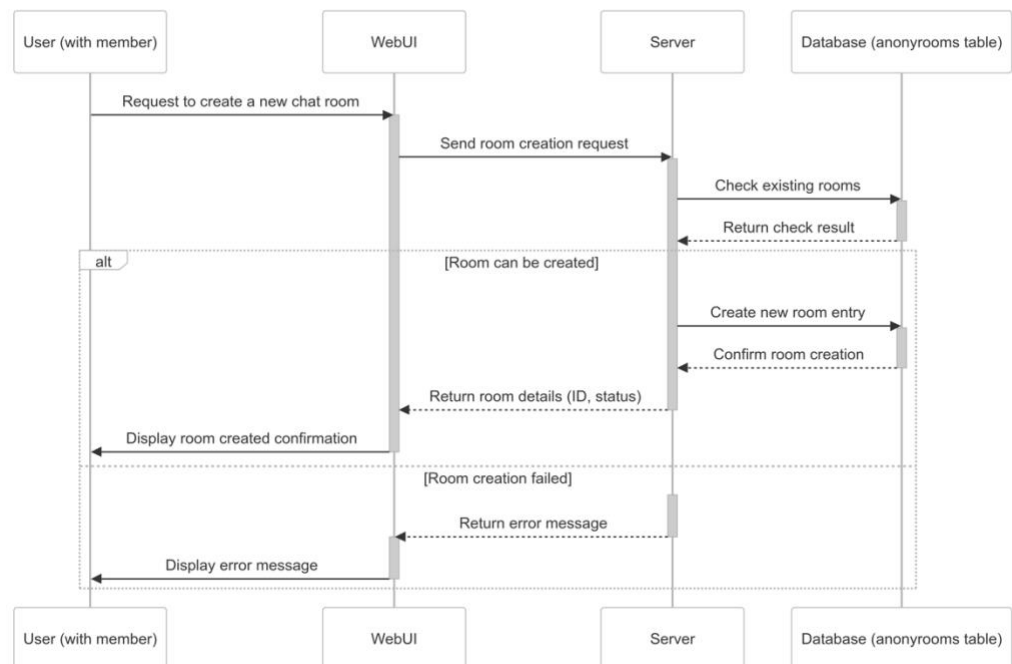


เมื่อผู้ใช้เปิดหน้า Discussion Board ขึ้นมาจะเป็นการส่งคำขอ ดู post ไปให้ server และ server จะทำการดึงข้อมูลมาจาก Database และส่งคืนกลับไปให้ WebUI แสดงผล และหากผู้ใช้กดโชว์ความคิดเห็นขึ้นมา นั้นจะเป็นการส่งคำขอ ดูความคิดเห็นไปให้ server และ server จะทำการดึงข้อมูลมาจาก Database และส่งคืนกลับไปให้ WebUI แสดงผลเช่นเดียวกัน เมื่อผู้ใช้ทำการแสดงความคิดเห็นลงใน ข้อมูลจะถูกส่งให้ server จัดการเก็บลงใน database ในตาราง comment และ notification และทำการส่งผลลัพธ์เพื่อนำไปแสดงผลต่อไป

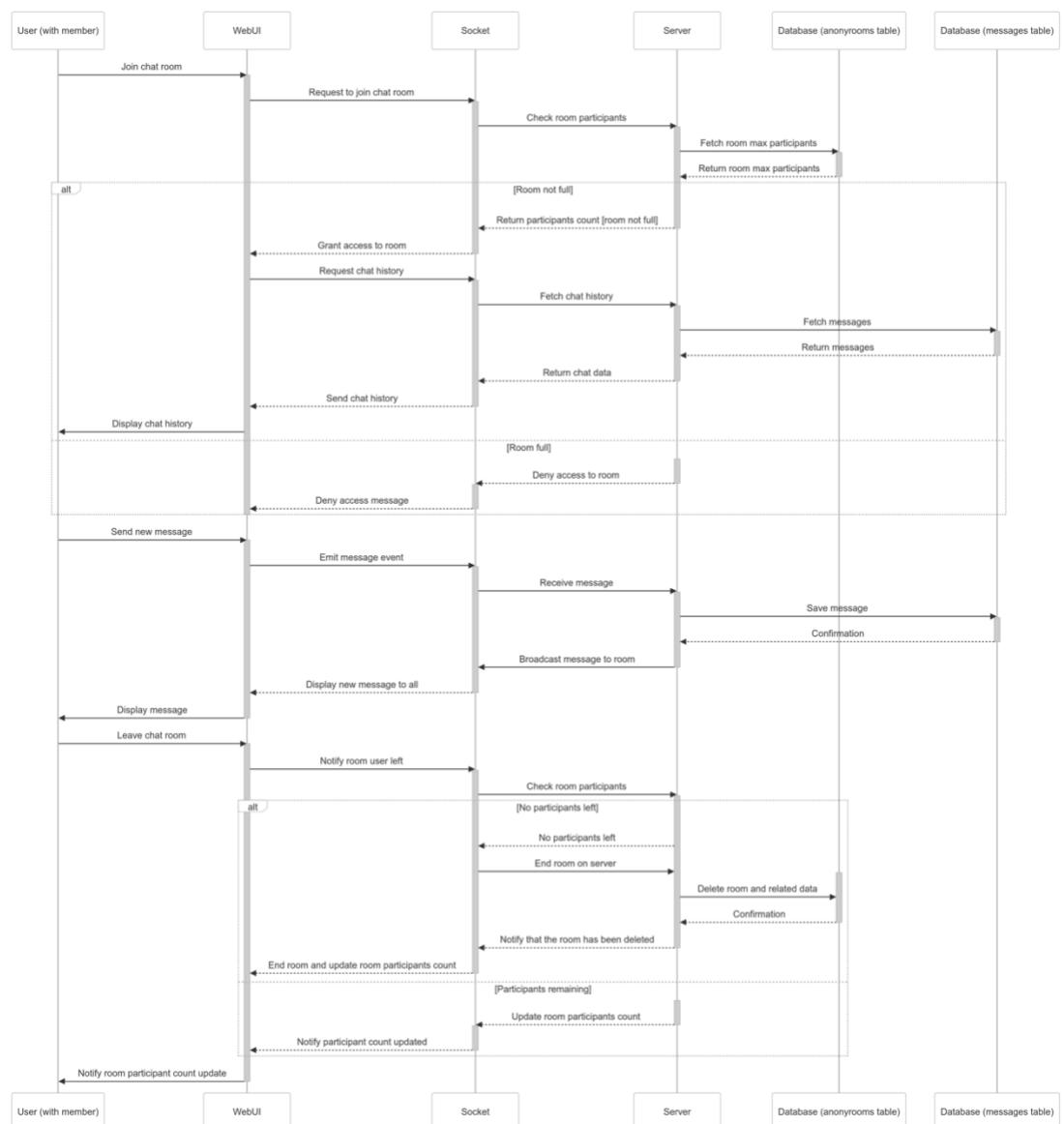


สำหรับส่วนของการสร้าง post ใหม่ ผู้ใช้จะทำการกรอกข้อมูล จากนั้น WebUI จะส่งข้อมูลไปให้ server เพื่อนำไปเก็บใน database ต่อไป หากบันทึกสำเร็จจะแจ้งกลับมาที่ WebUI และทำการโหลดหน้าใหม่อีกครั้งเพื่อแสดงข้อมูลล่าสุด

2. ระบบ Anonymous-Chat



สำหรับ AnonymousChat เริ่มจากการสร้างห้องแชท เมื่อผู้ใช้ทำการกรอกข้อมูลสำหรับสร้างห้องเรียบร้อยแล้ว WebUI จะส่งข้อมูลไปให้ Server เพื่อให้ Server นำไปตรวจสอบว่ามีห้องชื่อเดียวกันอยู่ใน Database หรือไม่ หากไม่มีจึงจะทำการบันทึกลง Database และ redirect ไปหน้าห้องแชทที่สร้าง แต่หากสร้างห้องแชทไม่ได้จะทำการแจ้งกลับไปให้ผู้ใช้ทราบ

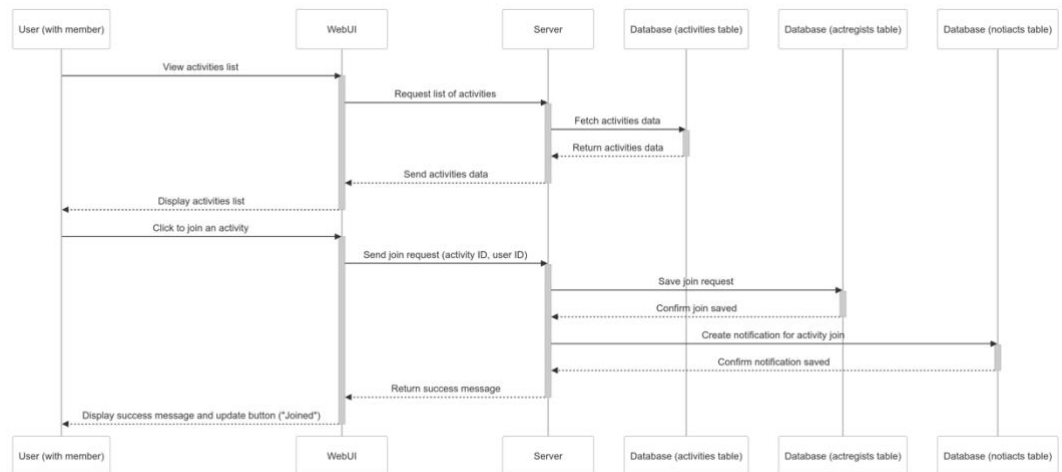


เมื่อเข้ามาในห้องแชท WebUI จะส่งคำขอในการเข้าห้องไปให้ Server และ Server จะทำการตรวจสอบจาก Socket ว่ามีจำนวนคนในห้องเกินกว่าที่กำหนดหรือไม่ หากไม่เกิน ก็จะอนุญาตให้เข้ามาในห้องนี้ แต่ถ้าเกินจำนวนที่กำหนดจะแสดงข้อความกลับไปเมื่อเข้าห้องมาได้แล้ว Client จะส่งคำขอในการเรียกดูประวัติแชทจาก Server และ Server จะดึงข้อมูลจาก Database มาให้อีกที

ในส่วนของการส่งข้อความ เมื่อผู้ใช้ส่งข้อความเข้ามา WebUI จะส่งต่อข้อความไปที่ Socket และ Socket จะทำหน้าที่ 2 อย่างคือ ส่งข้อความไปให้ Server เพื่อเก็บใน Database และส่งข้อความคืนกลับมาให้ผู้ใช้เห็นแบบ Realtime

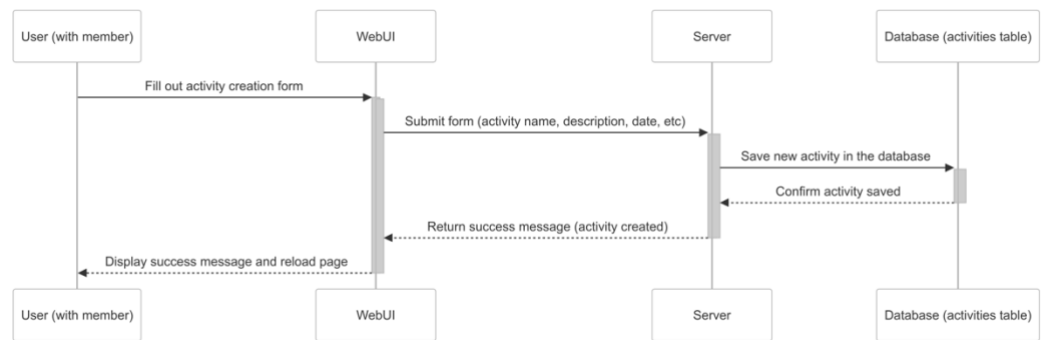
กรณีที่ออกจากห้อง WebUI จะส่งคำขอในการเข้าห้องไปให้ server และ server จะทำการตรวจสอบจาก socket ว่ามีจำนวนคนอยู่ในห้องหรือไม่ หากไม่มี Server จะทำการลบข้อมูลห้องและข้อมูล Anonymous ออกจาก Database และส่งค่าคืนกลับมาแสดงให้ผู้ใช้ แต่ถ้าหายังมีผู้ใช้อื่นอยู่ในห้องจะเพียงแค่ทำการลบข้อมูล Anonymous ของตัวเองและทำการอัปเดตค่าใน Socket เท่านั้น

3. ระบบ Activities



เมื่อผู้ใช้เปิดหน้า Activity ขึ้นมา WebUI จะส่งคำขอตู้ activities จาก Server และทำการดึงข้อมูลจาก Database ส่งกลับคืนไปยัง WebUI เพื่อนำมาแสดงผล

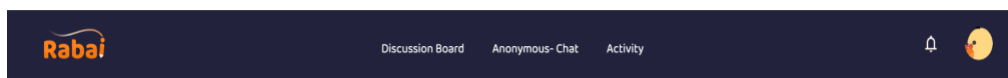
หลังจากผู้ใช้ได้ทำการกดปุ่ม join activity หลังจากนั้น WebUI ก็ทำการส่งข้อมูลคำร้องขอไปยัง Server และทำการบันทึกข้อมูลคำร้องขอนี้ไว้ใน Database และทำการสร้างแจ้งเตือนสำหรับการเข้าร่วม activity นี้ หากมีการบันทึกคำร้องขอสำเร็จ ก็ทำการแจ้งกลับไปยัง WebUI เพื่อแสดงผลการขอเข้าร่วมสำเร็จ



ในส่วนของการสร้าง activity ใหม่ นั้นผู้ใช้จะทำการกรอกข้อมูล จากนั้น WebUI จะส่งข้อมูลไปยัง Server เพื่อนำไปเก็บใน Database หากมีการบันทึกสำเร็จจะแจ้งกลับมาที่ WebUI และทำการโหลดหน้าใหม่อีกครั้งเพื่อแสดงข้อมูลล่าสุด

UX/UI Design และ การทำงานของระบบ

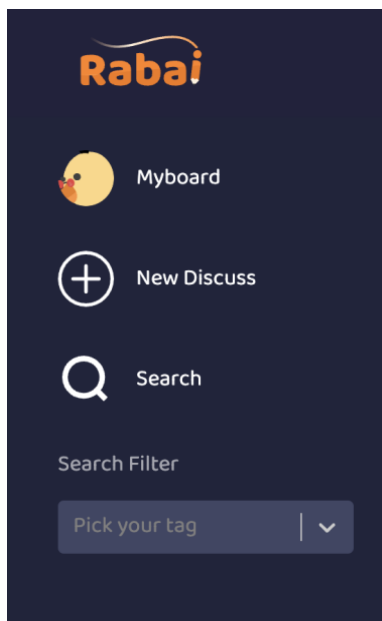
1. Navbar



เป็นแถบนำทางด้านบนของ page ซึ่งจะรวบรวมปุ่มสำหรับนำทางผู้ใช้ไปยังหน้าที่ต้องการหรือเพื่อใช้ฟีเจอร์ต่าง ๆ ของระบบ โดยการดีไซน์จะเน้นไปที่ความเรียบง่าย ทำให้ผู้ใช้สามารถอ่านและเลือกใช้ฟีเจอร์ต่างๆ ได้อย่างง่ายดาย ซึ่งประกอบไปด้วย

- 1.1. Discussion Board
- 1.2. Anonymous- Chat, Activity
- 1.3. Notification(Bell Icon) : จะเป็นส่วน pop up แจ้งเตือนเมื่อมีคน comment หรือมีคนกด join activity
- 1.4. Profile

2. Side bar

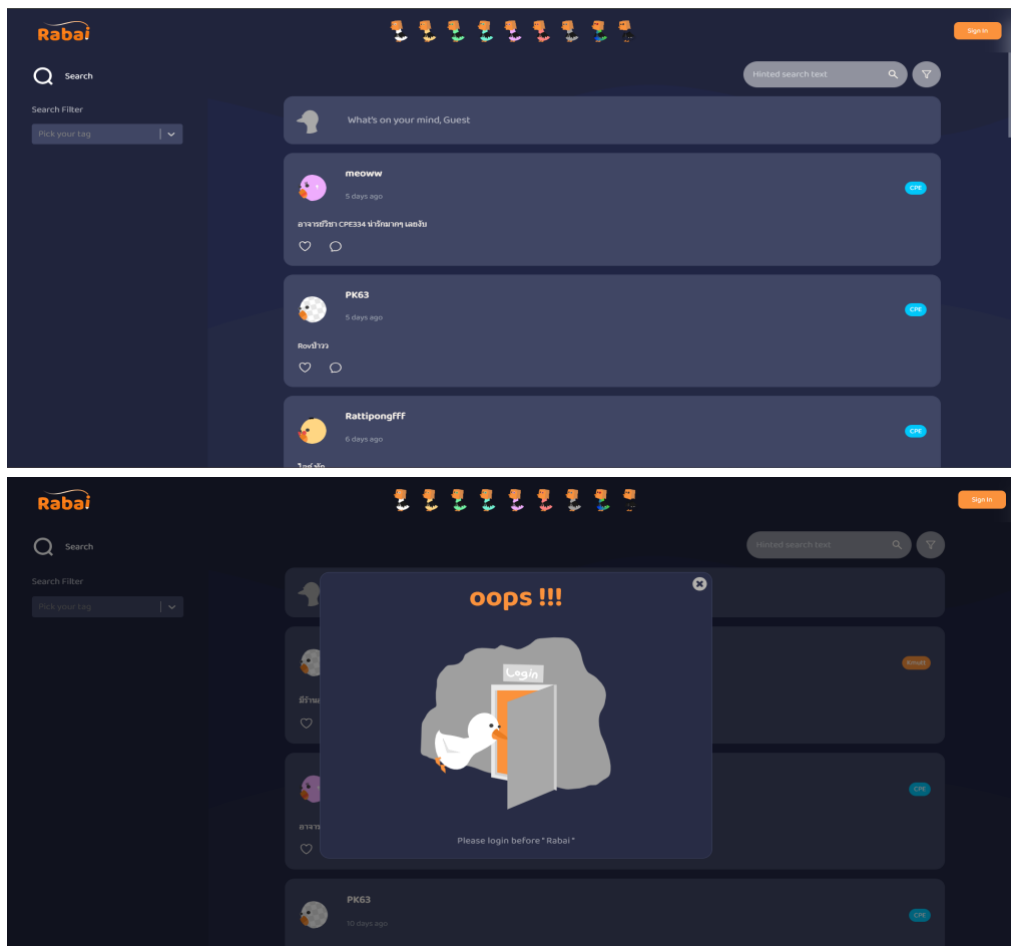


หลังจากที่ได้มีการสำรวจความพึงพอใจทั้งความง่ายในการใช้งานและความสวยงามในการใช้งานแล้วพบว่า จะให้แถบ menu อยู่ด้านซ้ายของ page เป็นที่รวบรวมฟีเจอร์เสริมของระบบ ซึ่งจะประกอบไปด้วย

- 2.1. Myboard : แสดงเฉพาะ topic ที่ตัวเองสร้างขึ้น

- 2.2. New Discuss : สร้าง topic ใหม่
- 2.3. Search : ค้นหา keyword ที่ต้องการ
- 2.4. Filter : กรอง topic โดยใช้ tag

3. Guest View



สำหรับ Guest view หน้านี้จะมีการจำกัดการใช้งานฟีเจอร์ของระบบเอาไว้ โดยผู้ที่ไม่ได้ทำการลงชื่อเข้าสู่ระบบจะสามารถทำได้แค่อ่านฟีดและข่าวสารต่างๆ พร้อมทั้ง search and filter เพื่อค้นหา topic ที่สนใจหรือเกี่ยวข้องกับคณะสาขาของตัวเองเท่านั้น สำหรับดีไซน์จะมีการแบ่งเป็น 3 ส่วนหลักๆคือ Sidebar, Navbar และ Board ถ้าผู้ใช้มีการใช้งานส่วนของ create discuss หรือ comment ระบบจะแสดง pop-up เพื่อแจ้งให้ผู้ใช้ทราบว่า ต้องทำการ Sign In ก่อนจึงจะใช้ได้

4. Sign up

The top screenshot shows the 'Create Your Profile' sign-up page. At the top, there is a validation message: 'Email must be a valid @mail.kmutt.ac.th address'. The main heading is 'Create Your Profile'. Below the heading is a circular profile picture placeholder showing a white bird with a checkered pattern. To the right of the placeholder are four input fields: 'Username' (containing 'PK63'), 'E-mail' (containing 'fu555@gmail.com'), 'Password', and 'Confirm password'. Each password field has a 'Show' link. Below the input fields is a 'Submit' button. At the bottom left, there is a link: 'Already a member? Sign in here'.

The bottom screenshot shows the 'Select Your Profile' modal. It features a 3x3 grid of nine different bird avatars. The avatars are: top row (purple, red, grey), middle row (yellow, green, blue), and bottom row (white, rainbow, checkered). The modal is centered over the sign-up form.

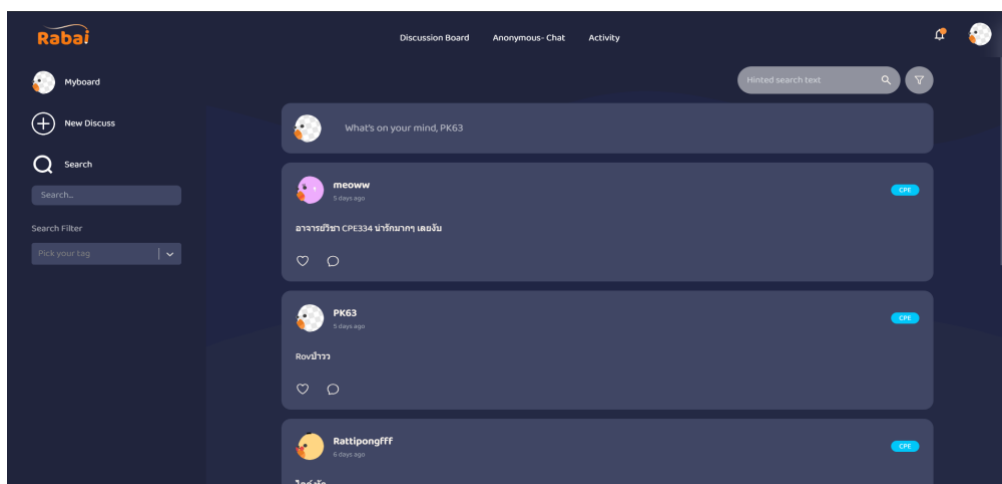
สำหรับ Sign Up Page หน้านี้จะมีการดีไซน์ให้ผู้ใช้กรอกข้อมูลเพื่อทำการลงทะเบียนสมัครเพื่อสร้างบัญชีของตัวเองในกรณีที่ยังไม่มีบัญชีหรือต้องการจะสร้างบัญชีใหม่ โดยผู้ใช้จะต้องกรอกรายละเอียดดังนี้ Username, Email, Password, Confirm-password และทำการเลือกรูป profile ตามที่ต้องการ สำหรับ Email address ผู้ใช้จะต้องใช้ address ของมหาวิทยาลัยเท่านั้น ถ้าผู้ใช้กรอก Email address ไม่ถูกต้องจะมี alert เพื่อแจ้งเตือนว่าผู้ใช้กรอก Email address ไม่ถูกต้อง นอกจากนี้หากผู้ใช้เคยลงทะเบียนไปแล้วจะสามารถคลิกที่ sign in here เพื่อไปสู่หน้า sign in แทนได้

5. Sign in

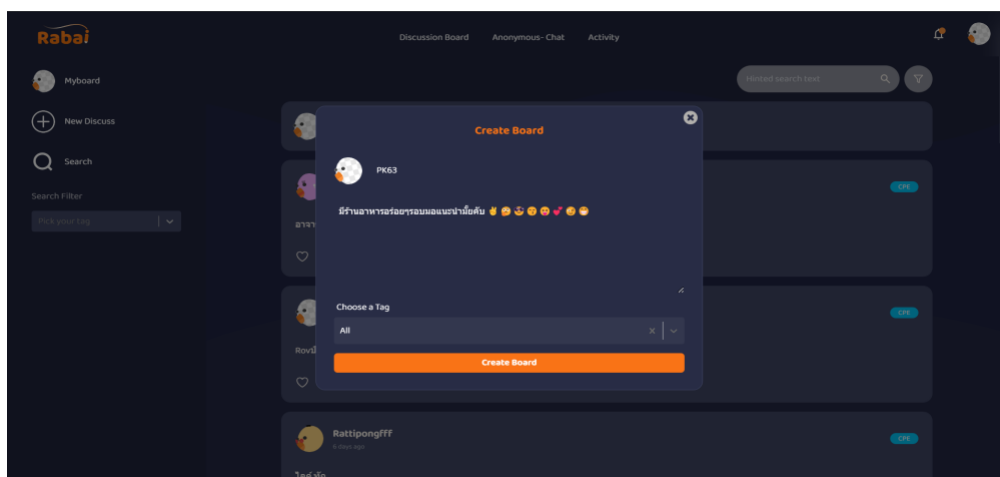
The image shows a sign-in page for 'Just Rabai!'. The page has a dark blue background with a subtle mountain silhouette. At the top center is the 'Just Rabai!' logo, with 'Just' in white and 'Rabai!' in orange. Below the logo is a sign-in form. It has two input fields: 'Email' and 'Password'. The 'Email' field contains the text 'fu555@mail.kmutt.ac.th'. The 'Password' field contains several dots and a 'Show' link to its right. Below the password field is an orange 'Sign in' button. At the bottom of the form, there is a link that says 'Don't have an account? Register here'.

สำหรับ Sign In Page หน้านี้จะมีการดีไซน์ให้ผู้กรอกข้อมูลอีเมล และ Password เพื่อเข้าสู่ระบบ โดยที่อีเมลจะต้องเป็นอีเมลของมหาวิทยาลัยเท่านั้น (ลงท้ายด้วย @mail.kmutt.ac.th) และในช่องกรอก Password สามารถกดปุ่ม Show เพื่อเช็ค Password ที่กรอกลงไปได้ว่ากรอกลงไปถูกต้องหรือไม่ เมื่อกรอกข้อมูลครบแล้วก็กดปุ่ม Sign in เพื่อล็อกอินเข้าสู่ระบบ แต่ถ้ากรอกข้อมูลผิด การล็อกอินก็จะไม่สำเร็จ โดยจะมีการเตือนว่ามีการกรอกข้อมูลช่องไหนผิด และถ้าหากผู้ใช้ยังไม่มีบัญชี ก็สามารถกดปุ่ม Register here เพื่อ redirect ไปยังหน้าลงทะเบียนบัญชีใหม่ได้

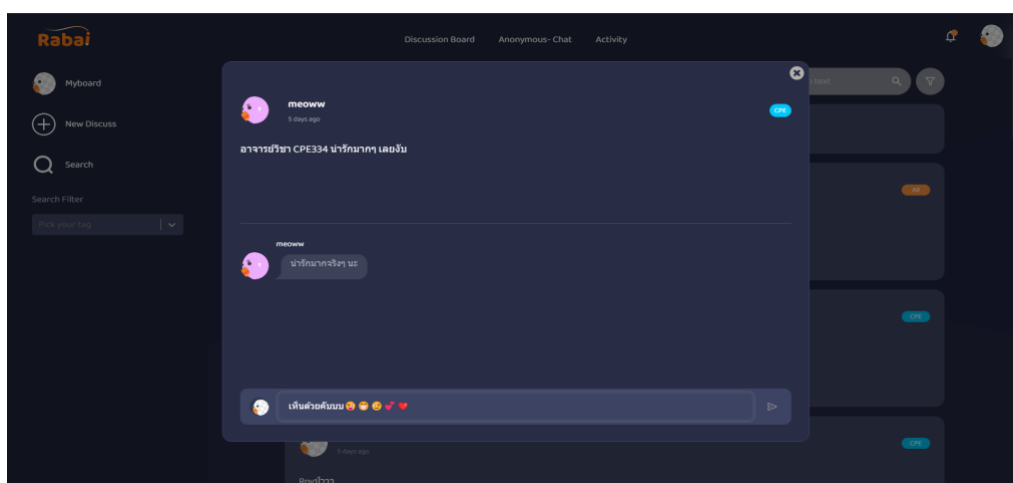
6. Discussion Board



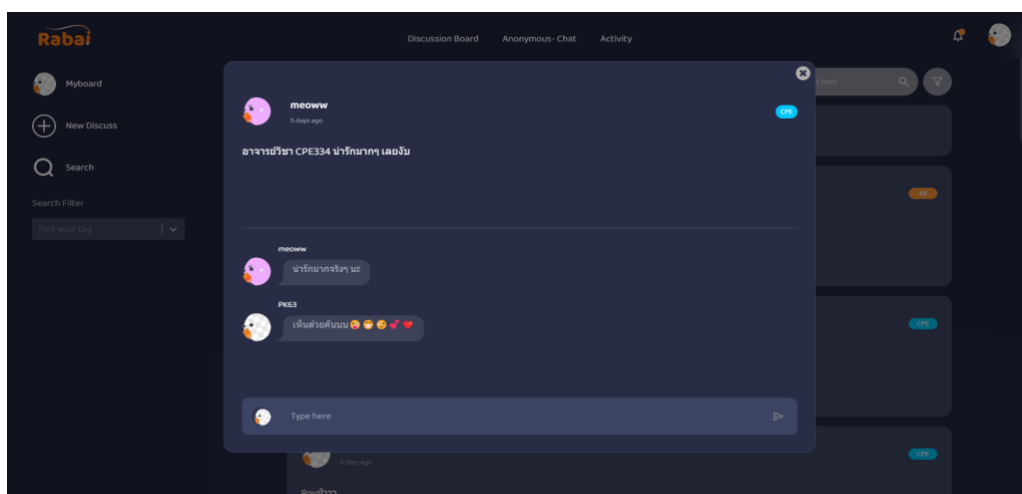
สำหรับหน้า Discussion Board จะมีการแสดงบอร์ดทั้งหมดขึ้นมาโดยเรียงตามเวลาโพสต์ล่าสุด หากเป็นผู้ใช้ที่ล็อกอินเข้ามาแล้ว จะสามารถแสดงความคิดเห็นได้ โดยกดที่ไอคอนกล่องคำพูดใต้โพสต์และจะมีหน้าต่างลอยปรากฏขึ้นมา ซึ่งจะแสดงข้อมูลของบอร์ดและคอมเมนต์ทั้งหมดของโพสต์ และจะมีช่องสำหรับกรอกคอมเมนต์ หากต้องการแสดงความคิดเห็นก็พิมพ์ข้อความลงในช่องและกดปุ่มส่ง (ไอคอนเครื่องบินกระดาษ) เป็นอันเสร็จสิ้น การแสดงความคิดเห็น และสามารถสร้างบอร์ดใหม่ได้โดยกด ที่ช่องด้านบนที่มีข้อความ “What’s in your mind” แต่หากเป็นผู้ใช้ระดับ Guest ที่ไม่ได้ล็อกอิน ก็จะไม่สามารถแสดงความคิดเห็นได้และหากพยายามกดปุ่มไอคอนกล่องคำพูด ก็จะมีหน้าต่างเตือนให้ล็อกอินก่อนปรากฏขึ้นมา จึงจะสามารถคอมเมนต์ได้ และก็จะไม่สามารถสร้างบอร์ดได้เช่นกัน โดยหากพยายามกดช่องบนสุด ก็จะมีหน้าต่างเตือนให้ล็อกอินก่อนปรากฏขึ้นมา



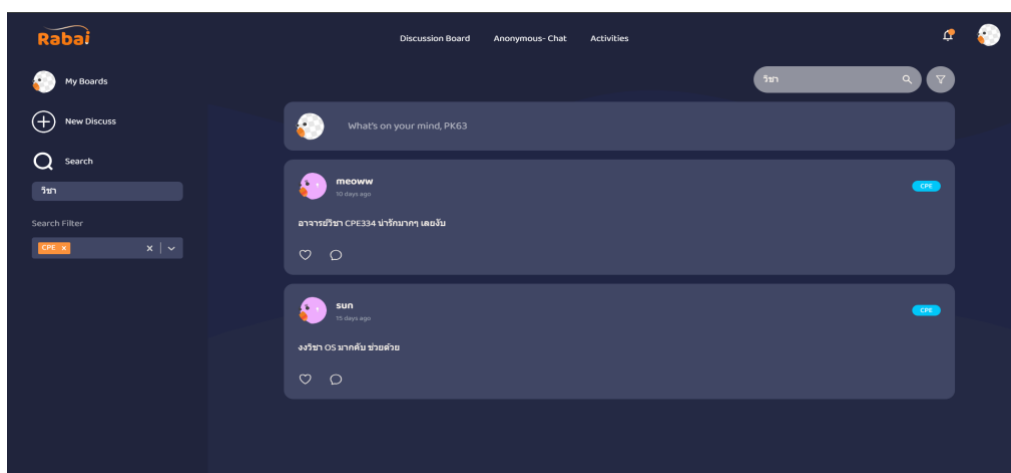
สำหรับการสร้างบอร์ดใหม่สามารถทำได้ด้วยการกดที่ปุ่ม New Discuss ที่แถบ Sidebar หรือกดที่ช่องที่มีข้อความ “What’s in your mind” เมื่อทำการกดแล้ว จะมีหน้าต่าง Create Board ขึ้นมาหลังจากนั้นก็เพิ่มพื้ที่ต้องการจะโพสต์ลงไปได้ และยังสามารถเลือกแท็กที่ต้องการได้อีกด้วยหลังจากนั้นเมื่อทำการเลือกทุกอย่างเสร็จเรียบร้อยแล้วก็สามารถกดปุ่ม Create Board ที่อยู่ด้านล่างเพื่งเท่านั้นการสร้างโพสดีใหม่ก็จะเสร็จสมบูรณ์



ในการแสดงความคิดเห็น (คอมเมนต์) เมื่อปรากฏหน้าต่างลอยดังกล่าวก็สามารถแสดงความคิดเห็นได้ โดยพิมพ์ความคิดเห็นที่ต้องการแสดง ลงไปในช่องกรอกคอมเมนต์ที่มีคำว่า “Type here” เมื่อกรอกคอมเมนต์เสร็จแล้ว ก็สามารถส่งคอมเมนต์ด้วยการกดปุ่มรูปเครื่องบินกระดาษเพื่อยืนยันการส่งคอมเมนต์

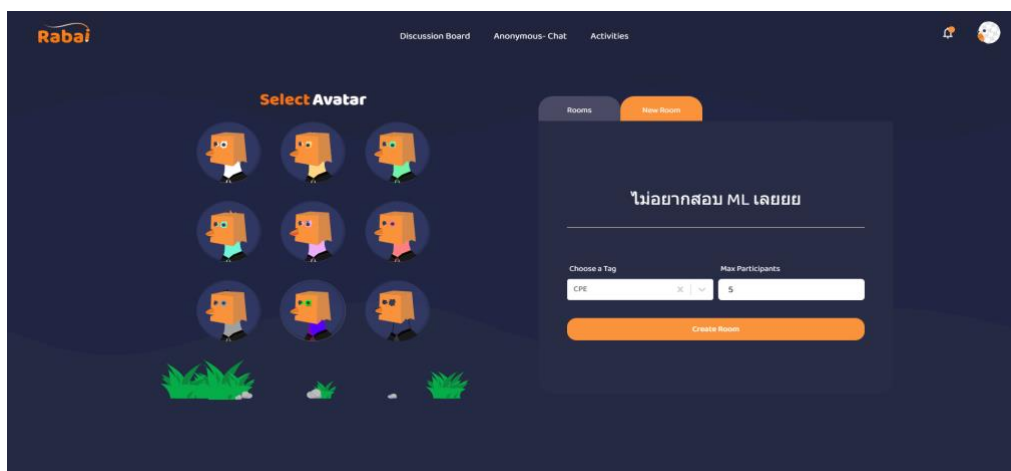


สำหรับการแสดงความคิดเห็น สามารถทำได้โดยการกดสัญลักษณ์พูดคุย หลังจากนั้นจะมีหน้าโพสต์ที่ต้องการแสดงความคิดเห็นขึ้นมา สามารถอ่านและพิมพ์ความคิดเห็นลงไปได้ เมื่อทำการแสดงความคิดเห็นสำเร็จข้อความ ก็จะขึ้นมาแสดงบนหน้าของโพสต์นั้นด้วยเช่นเดียวกัน

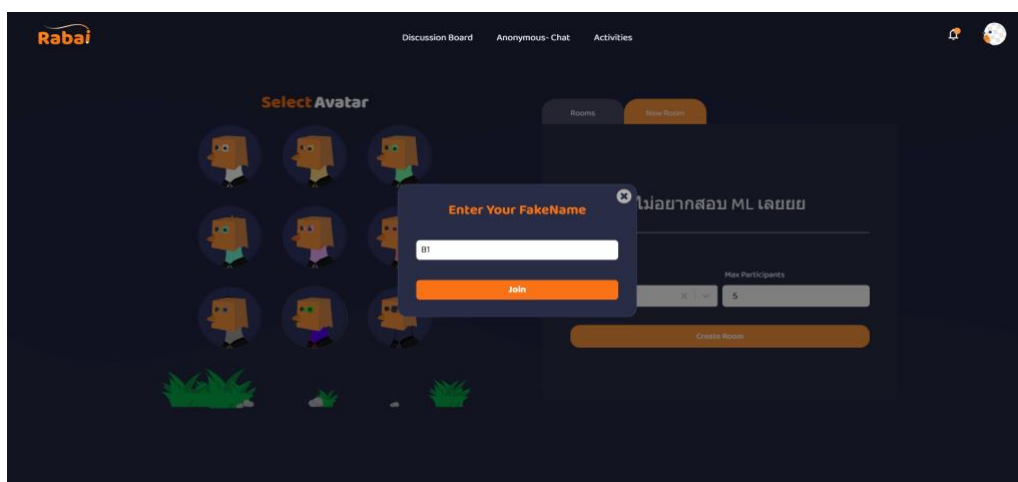


สำหรับการค้นหาบอร์ด สามารถเสิร์ชด้วยแท็กและ keyword โดยหากต้องการเสิร์ชแท็ก สามารถทำได้โดยการกดปุ่มรูปกรวย จากนั้นจะมี Drop down ของแท็กต่างๆให้ผู้ใช้เลือกโดยที่สามารถเลือกได้เพียง 1 แท็กเท่านั้นและหากต้องการเสิร์ชด้วย keyword ก็ทำได้โดยการพิมพ์ keyword ที่ต้องการหา ลงไปในช่องที่มีไอคอนแว่นขยาย โดยที่ผู้ใช้ไม่จำเป็นต้องกด Enter เพื่อเสิร์ช แต่ระบบจะเสิร์ชด้วย keyword ให้อย่างอัตโนมัติตลอดเวลา โดยบอร์ดที่ปรากฏจะเปลี่ยนไปเรื่อยๆ ตามเกณฑ์ที่ใช้ในการเสิร์ช

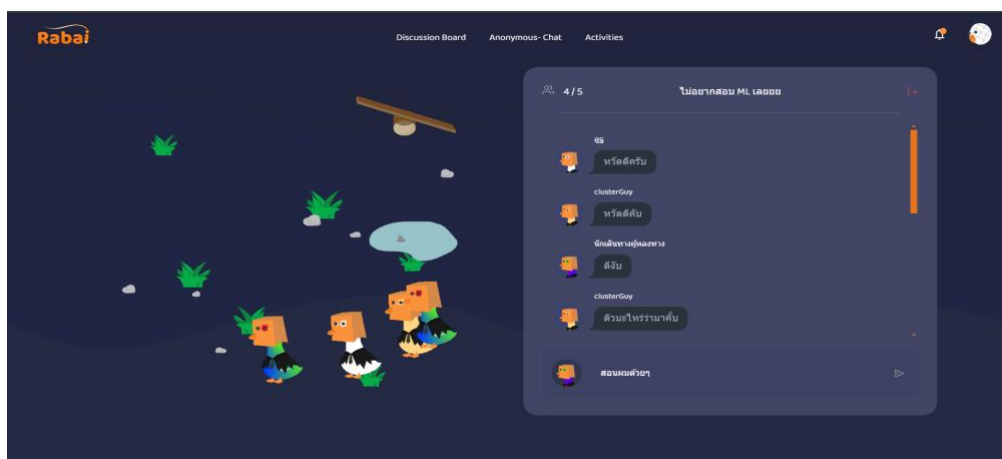
7. Anonymous chat



ในส่วนของการสร้างห้องแชทใหม่ เมื่อเข้าหน้า Anonymous-Chat ให้กดเข้าไปที่หน้า New Room และจะมีช่องต่างๆ ให้กรอกข้อมูลของห้องแชท ได้แก่ ชื่อห้อง แท็ก จำนวนผู้เข้าร่วมแชทสูงสุด (รวมผู้สร้างห้องแชทด้วย) และเลือกรูปโปรไฟล์ เมื่อกรอกข้อมูลครบทั้งหมด ก็กดปุ่ม “Create Room”



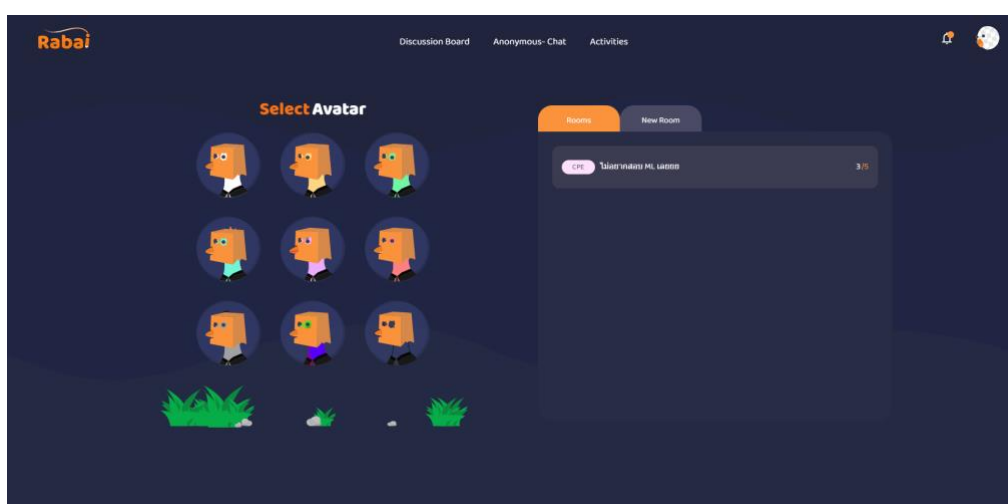
หลังจากกดปุ่ม “Create Room” จะมีหน้าต่างลอยที่มีช่องสำหรับกรอกชื่อสำหรับใช้ในห้องแชทเมื่อกรอกเรียบร้อยแล้วก็กดปุ่ม “Join” เพื่อเข้าสู่ห้องแชท



ภายในห้องแชทจะแบ่งเป็นสองส่วนหลัก ส่วนของแชททางด้านขวาจะเป็นส่วนที่แสดงข้อความซึ่งจะมี 3 ส่วนย่อย คือ

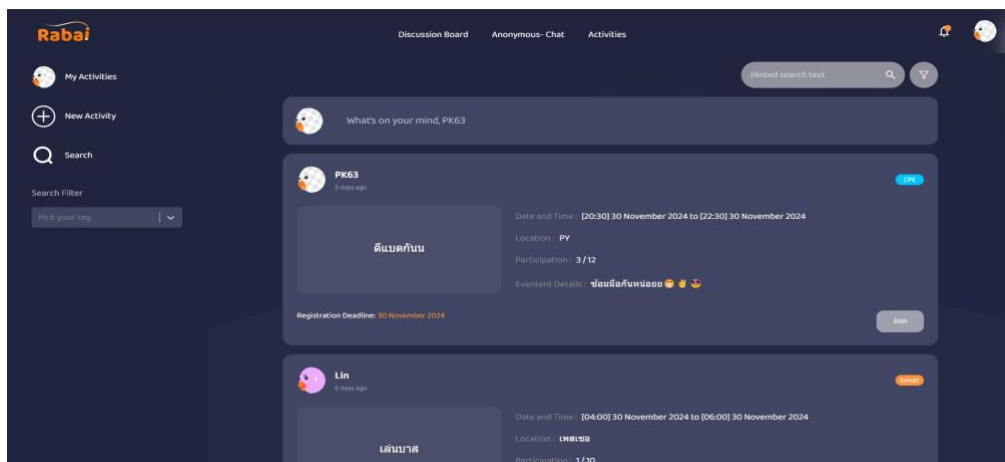
- ส่วนหัว ซึ่งแสดงชื่อแชท จำนวนคนในห้อง และปุ่มออกจากห้องแชท
- ส่วนของแชท ที่จะแสดงข้อความ ชื่อ และรูปโปรไฟล์ของคนที่ส่งข้อความ
- ส่วนของการส่งข้อความ ซึ่งจะมีรูปโปรไฟล์ของผู้ใช้ ช่องกรอกข้อความ และปุ่มสำหรับกดส่งข้อความ

และในหน้า Chat Room ด้านซ้ายจะมีการแสดง avatar ที่ผู้ใช้ภายในห้องนี้แต่ละคนได้ทำการเลือกโดยมีการเพิ่มรายละเอียดเล็กน้อยเพื่อเพิ่มความน่าสนใจและน่าใช้งาน เช่น animation ของ avatar เป็นต้น

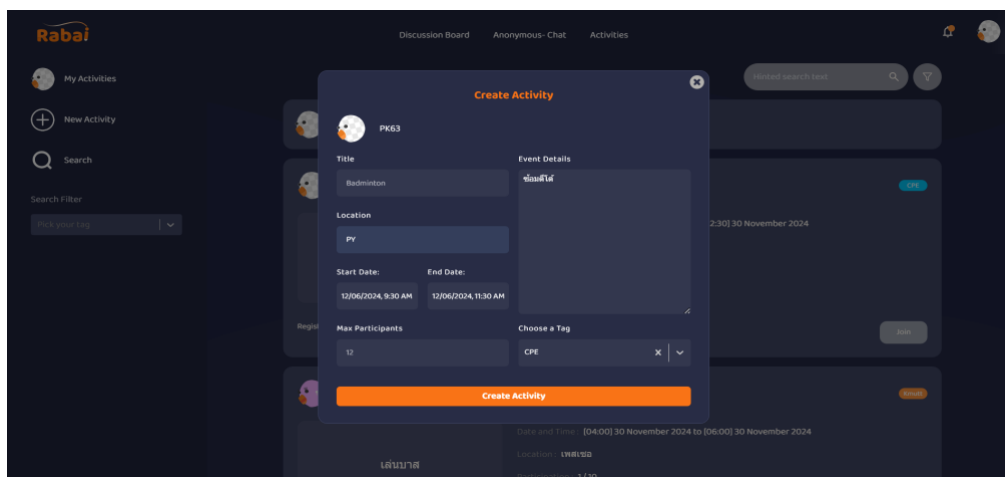


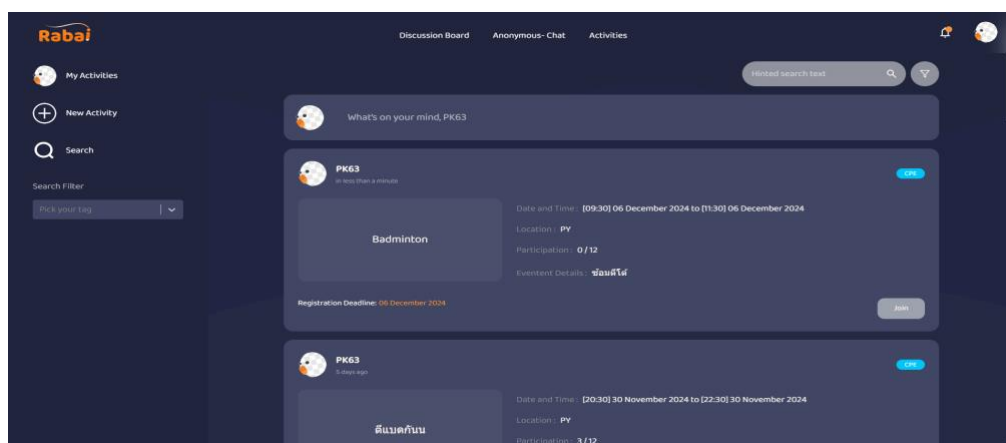
ซึ่งในหน้า Anonymous-Chat จะมีการแสดงรายการของห้องแชททั้งหมดที่สามารถเข้าร่วมได้ โดยจะมีแท็ก ชื่อห้อง และจำนวนคนที่อยู่ในห้องแชทแสดงบอกผู้ใช้งาน โดยการเข้าห้องแชทจะต้องมีการเลือกโปรไฟล์ และตั้งชื่อเช่นเดียวกัน

8. Activity

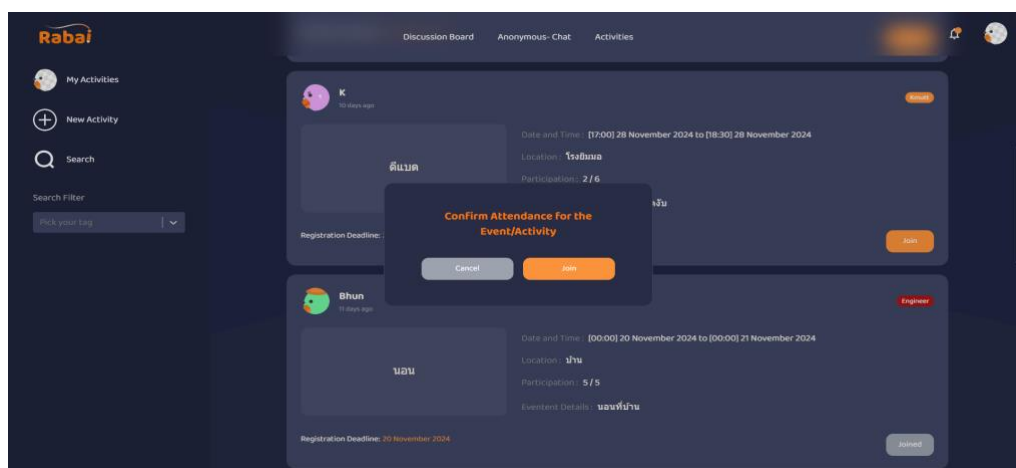


สำหรับหน้า Activity มีการดีไซน์ให้สามารถเข้าได้จาก Navbar ซึ่งจะเป็นหน้าฟีดที่รวบรวมกิจกรรมต่างๆ ไว้ โดยการดีไซน์จะแบ่งเป็น 3 ส่วนเพื่อแยกเป็นสัดส่วนให้เข้าใจง่ายคือ Sidebar, Navbar, Feed ผู้ใช้สามารถเลือกดูหรือค้นหา activity ได้โดยใช้ search and filter เพื่อค้นหา activity ที่สนใจ และกด join เพื่อเข้าร่วม activity ที่สนใจ ซึ่งจะมีการดีไซน์ปุ่มโดยใช้สีที่แตกต่างจากพื้นหลังเพื่อให้เห็นและสังเกตได้ง่าย และผู้ใช้สามารถสร้าง activity ใหม่ได้โดยใช้ฟีเจอร์ New Activity ที่อยู่ใน sidebar

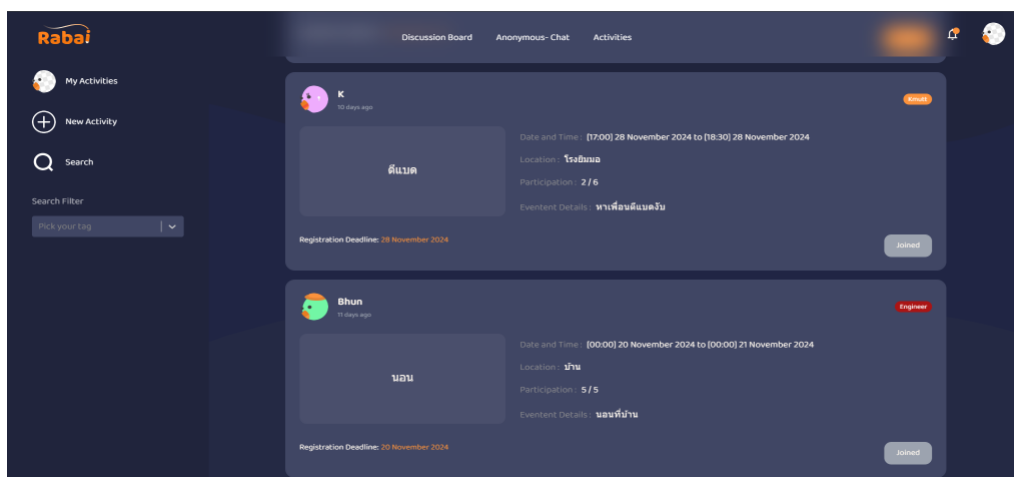




เมื่อกดที่ New Activity ที่แถบ Side Bar ผู้ใช้จะเข้าสู่หน้า Create Activity ซึ่งผู้ใช้สามารถสร้างกิจกรรมที่ต้องการจะทำได้ โดยแจ้งสิ่งที่จะทำ วันที่ เวลา สถานที่ และผู้เข้าร่วมเมื่อกรอกข้อมูลข้างต้นเสร็จสิ้นให้กดที่ปุ่ม Create Board

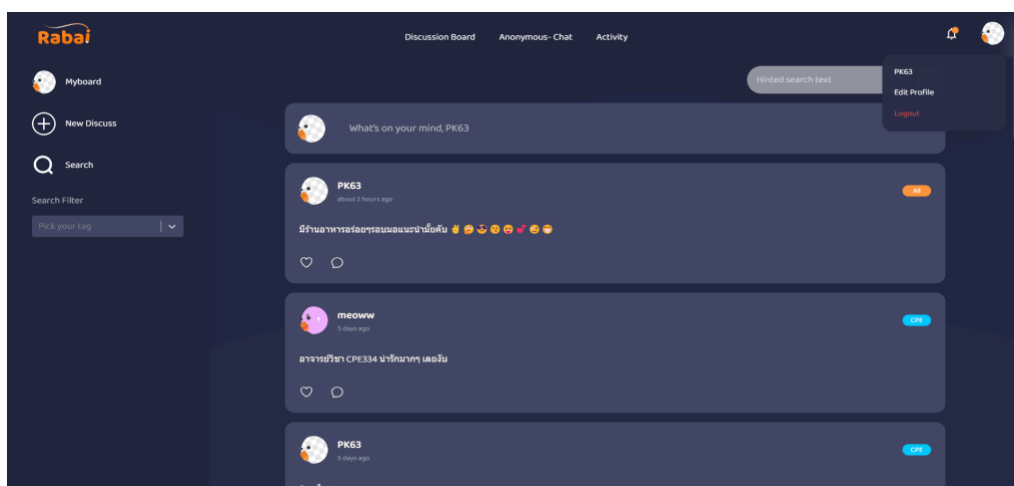


ผู้ใช้สามารถเลือกเข้าร่วม Activity ที่ต้องการได้โดยการกดปุ่ม “Join” ที่มุมขวาล่างของ Activity ที่สนใจเข้าร่วม โดยหลังจากกดปุ่มแล้ว จะมีหน้าต่างลอยให้กดยืนยันการเข้าร่วมอีกครั้ง

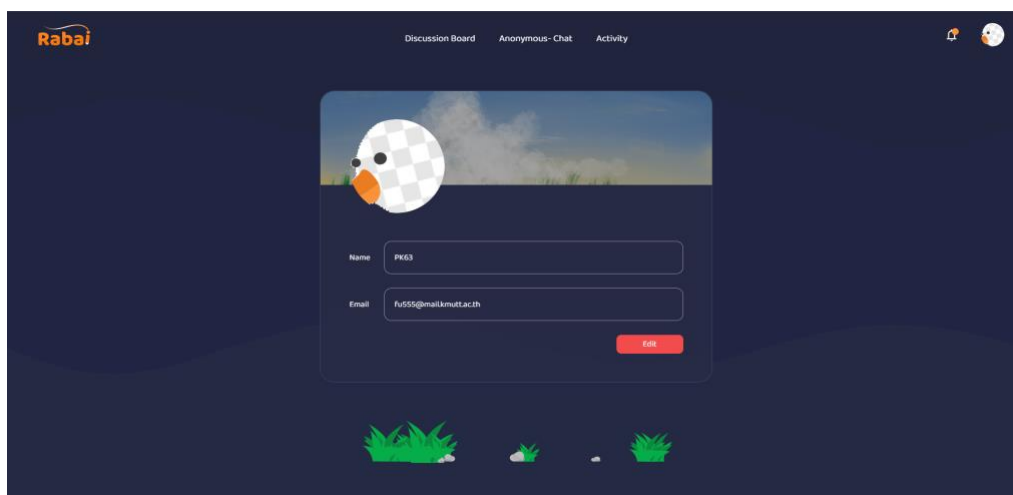


Activity ที่ได้เข้าร่วมแล้วจะไม่สามารถกดเข้าร่วมซ้ำได้อีก โดยมีการดีไซน์ให้ปุ่ม “Join” สีส้ม เปลี่ยนเป็นปุ่ม “Joined” สีเทา เพื่อให้ผู้ใช้รับทราบว่า join สำเร็จแล้ว

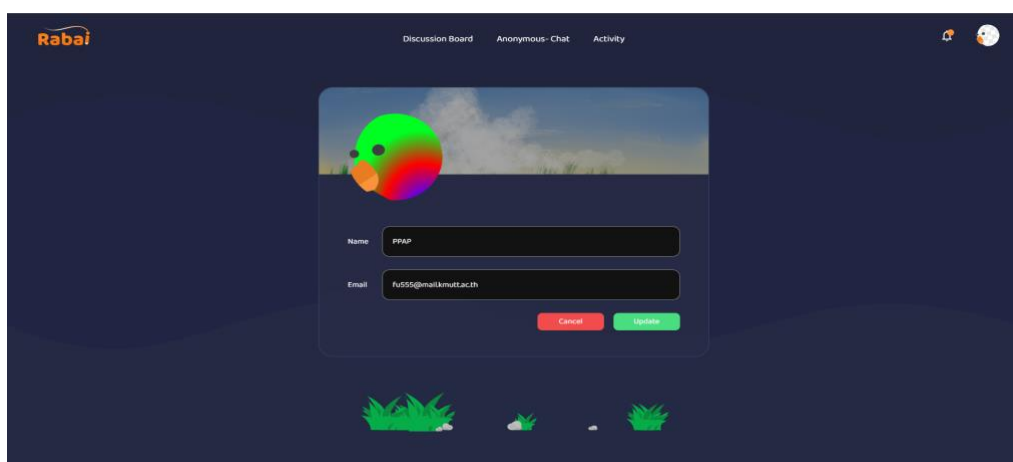
9. Profile : ผู้ใช้สามารถเปลี่ยนรูปโปรไฟล์รวมถึงชื่อได้



สำหรับ Profile and Edit Profile page จะมีการดีไซน์ให้คงความเรียบง่าย และมีรายละเอียดที่ไม่เยอะจนเกินไป โดยผู้ใช้งานจะสามารถเข้าถึงหน้านี้ได้ด้วยการคลิกที่รูปโปรไฟล์ แล้วจะมี dropdown ให้เลือกเพื่อไปสู่หน้านั้นๆ

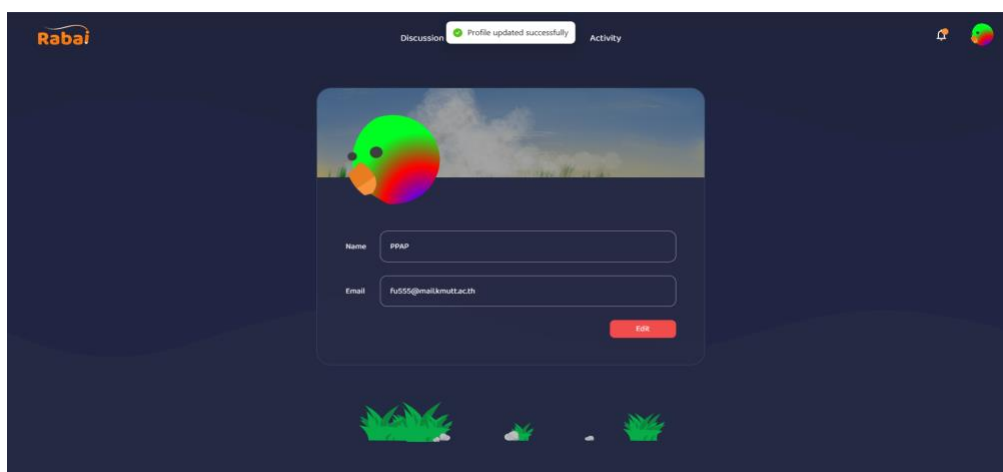


เมื่อผู้ใช้เข้ามาในหน้า Edit Profile จะมีการแสดงรายละเอียดข้อมูลของผู้ใช้ โดยจะแสดงชื่อและอีเมล ผู้ใช้สามารถทำการแก้ไขชื่อ Email และรูปโปรไฟล์ได้ สำหรับการออกแบบในหน้า Edit Profile เป็นการออกแบบให้มี ความเรียบง่ายที่สุดเพื่อให้สามารถอ่านรายละเอียด และใช้งานได้อย่างง่าย และมีการใช้สีแดงเพื่อเป็นจุดสังเกตให้ผู้ใช้งานมองเห็นแล้วสะดุดตาน่าสนใจ



ซึ่งเมื่อกดปุ่มแก้ไขข้อมูลจะมีการเปลี่ยนโหมดให้สามารถแก้ไขข้อมูลในช่องได้ นอกจากนั้นยังสามารถกดที่รูปเพื่อแก้ไขรูปโปรไฟล์ได้ซึ่งการแก้ไขจะยังไม่บันทึก

จนกว่าจะมีการ กดปุ่ม Update ข้อมูลเพื่อเป็นการบันทึกข้อมูลใหม่ไปยังระบบ



และหาก Update ข้อมูลผู้ใช้เสร็จเรียบร้อยแล้วและระบบ จะมีการแจ้งว่าการบันทึกสำเร็จและจะเห็นได้ว่าข้อมูลของผู้ใช้มีการเปลี่ยนแปลงทันที ซึ่งในส่วนของการแก้ไขข้อมูลผู้ใช้ ผู้ใช้สามารถทำการแก้ไขข้อมูลตนเองได้ทุกเมื่อ เว้นแต่การแก้ไขรหัสผ่านในการเข้าสู่ระบบจะไม่สามารถทำได้

สรุปผลการดำเนินงาน

การพัฒนาเว็บไซต์ Rabai (ระบาย) นี้ได้พัฒนาเว็บไซต์ที่เป็นพื้นที่สื่อสารสำหรับนักศึกษาของมหาวิทยาลัยเทคโนโลยี-พระจอมเกล้าธนบุรี โดยมีฟีเจอร์หลัก เช่น บอร์ดสนทนา (Discussion Board), ห้องแชทแบบไม่ระบุตัวตน (Anonymous Chat), และระบบกิจกรรม (Activities) เพื่อสนับสนุนการแลกเปลี่ยนความคิดเห็นและการเข้าร่วมกิจกรรมต่าง ๆ ภายในมหาวิทยาลัย ผลลัพธ์ที่ได้คือเว็บไซต์ที่มีฟีเจอร์ครบถ้วน และใช้งานได้จริงตามความต้องการของผู้ใช้ โดยมีการพัฒนาให้เหมาะสมกับนักศึกษาและการทำงานที่สะดวกในทุกอุปกรณ์

ในการดำเนินงานได้มีการเรียนรู้และประยุกต์ใช้เครื่องมือ Diagram หลายประเภทที่สำคัญในการออกแบบระบบซอฟต์แวร์ซึ่งเป็นกระบวนการที่ช่วยให้สามารถเข้าใจและจัดการกับความซับซ้อนของระบบได้อย่างมีประสิทธิภาพ หนึ่งในเครื่องมือที่ใช้ในการออกแบบคือ Use Case Diagram ซึ่งช่วยให้ทีมพัฒนาเห็นภาพรวมของฟีเจอร์หลักในระบบ เช่น ระบบการสมัครสมาชิกและเข้าสู่ระบบ ฟีเจอร์การโพสต์ในบอร์ดสนทนา หรือการสร้างห้องแชทแบบไม่ระบุตัวตน โดยการใช้ Use Case Diagram ทำให้สามารถ เข้าใจความต้องการของผู้ใช้งานในแต่ละขั้นตอนและเชื่อมโยงกับฟังก์ชันที่ระบบต้องการได้อย่างชัดเจน

นอกจากนี้ การใช้ Class Diagram เป็นเครื่องมือที่ช่วยในการออกแบบโครงสร้างข้อมูลของระบบ โดยแสดงความสัมพันธ์ระหว่างคลาสต่างๆ และวิธีการที่ระบบจะจัดการกับข้อมูล ซึ่งช่วยให้การพัฒนาเป็นไปตามหลักการของการจัดการข้อมูลอย่างมีระเบียบและสามารถขยายระบบได้ในอนาคต การแสดงความสัมพันธ์ของคลาสต่างๆ ทำให้มองเห็นภาพรวมของการทำงานในระบบและสามารถวางแผนการพัฒนาได้ดีขึ้น

อีกหนึ่งเครื่องมือที่มีความสำคัญคือ Sequence Diagram ซึ่งช่วยในการออกแบบลำดับขั้นตอนการทำงานของระบบในแต่ละฟีเจอร์ โดยแสดงการโต้ตอบระหว่างผู้ใช้และระบบ ในลำดับขั้นตอนที่เหมาะสม เช่น การลงทะเบียนผู้ใช้ การโพสต์ข้อความในบอร์ดสนทนา หรือการแชทแบบไม่ระบุตัวตน การใช้ Sequence Diagram ทำให้สามารถจัดระเบียบกระบวนการทำงานในระบบได้อย่างชัดเจนและสามารถติดตามได้ง่ายในการพัฒนา

การพัฒนาระบบ Rabai ยังมีการนำแนวทาง Agile ผ่าน Scrum Framework มาใช้ในการบริหารจัดการโครงการ ซึ่งช่วยให้ทีมสามารถพัฒนาอย่างต่อเนื่องและปรับปรุงได้อย่างรวดเร็วผ่านการทำงานเป็นรอบๆ (Sprint) โดยในแต่ละรอบจะมีการ กำหนดเป้าหมายที่

ชัดเจนและมีการทบทวนความคืบหน้าของงานซึ่งช่วยให้โครงการสามารถปรับเปลี่ยนหรือพัฒนาไปตามความคิดเห็นของผู้ใช้งานได้อย่างมีประสิทธิภาพ

จากการเรียนรู้และประยุกต์ใช้เครื่องมือเหล่านี้ทำให้เข้าใจถึงกระบวนการและหลักการออกแบบที่จำเป็นในการพัฒนาซอฟต์แวร์ที่มีประสิทธิภาพและสามารถตอบสนองความต้องการของผู้ใช้ได้เหมาะสม รวมถึงการออกแบบระบบที่ปรับเปลี่ยนและพัฒนาได้ในอนาคต