



โครงการเรื่อง
เพื่อนใจ (MU Health & Friends Service)

6687033 พิชิตชัย แพเจริญชัย
6687077 ฉัตรดนัย เอมพันธุ์

โครงการนี้เป็นส่วนหนึ่งของรายวิชา
ITDS283 Mobile Application Development
คณะเทคโนโลยีสารสนเทศและการสื่อสาร มหาวิทยาลัยมหิดล
ปีการศึกษา 2567

คำนำ

โครงการนี้ จัดทำภายใต้รายวิชา ITDS283 Mobile Application Development เพื่อตอบสนองต่อเป้าหมายการพัฒนาที่ยั่งยืน (Sustainable Development Goals: SDGs) ใน Goal 3: Good Health and Well-being – สร้างหลักประกันการมีสุขภาวะที่ดี และ ส่งเสริมความเป็นอยู่ที่ดีสำหรับทุกคนในทุกช่วงวัย และ Goal 10: Reduced Inequality - ลดความไม่เสมอภาคภายในและระหว่างประเทศ ในเอกสารฉบับนี้ได้ให้คำอธิบายการพัฒนา และการออกแบบ เพื่อนใจ (MU Health & Friends Service) ซึ่งเป็นแอปพลิเคชันเพื่อ จัดการบริการ และช่วยเหลือและส่งเสริมสุขภาวะทางกายและใจของผู้ใช้ ผ่านเทคโนโลยี ปัญญาประดิษฐ์ (AI) โดยมุ่งเน้นการพัฒนาบนแพลตฟอร์ม Flutter เพื่อให้สามารถเข้าถึงและ ใช้งานได้อย่างสะดวกบนอุปกรณ์พกพา

เพื่อนใจ (MU Health & Friends Service) ออกแบบให้เป็น เครื่องมือในการจัดการ การนัดหมาย จองบริการ รวมถึงช่วยเหลือด้านสุขภาวะทางจิต ที่สามารถให้คำปรึกษา เป็นต้นแก่ผู้ใช้งานผ่าน Telemedicine พร้อมคำนึงถึงความเสมอภาคในการเข้าถึงบริการ ด้านสุขภาพกายและใจ โดยแอปพลิเคชันจะสามารถติดต่อกับผู้ใช้แบบอัตโนมัติ และให้ คำปรึกษาที่เหมาะสมและเข้มข้นไปยังแหล่งข้อมูล

ทีมผู้พัฒนา และคณะเทคโนโลยีสารสนเทศและการสื่อสาร มหาวิทยาลัยมหิดล หวัง เป็นอย่างยิ่งว่าเอกสารฉบับนี้ จะเป็นแนวทางสำคัญในการพัฒนาแอปพลิเคชันให้มี ประสิทธิภาพสูงสุด ทั้งในด้านการออกแบบ การใช้งาน และผลกระทบเชิงบวกต่อผู้ใช้

สารบัญ

	หน้า
บทที่ 1 รายละเอียดของโครงการ.....	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์ของ Mobile Application.....	2
1.3 กลุ่มเป้าหมายของ Mobile Application.....	4
1.4 ศึกษา porównเทียบกับ Mobile Application อื่นที่มีอยู่ในปัจจุบัน.....	5
บทที่ 2 User Journey Map.....	9
2.1 การวิเคราะห์ User Journey Map.....	10
2.2 ตารางการทำ User Journey Map ในรูปแบบตาราง	10
2.3 แนวทางแก้ไขปัญหาโดย Mobile Application.....	11
2.4 User Journey Map รูปแบบ Graphviz	12
บทที่ 3 User Flow.....	13
3.1 3.1 Login Page - User Flow ผู้ใช้.....	13
3.2 Homepage - User Flow ผู้ใช้	14
3.3 Booking page - User Flow ผู้ใช้.....	16
3.4 QR Scan page - User Flow ผู้ใช้.....	18
3.5 Notification page - User Flow ผู้ใช้.....	19
3.6 Health Tracking page - User Flow ผู้ใช้	21
3.7 Feedback page - User Flow ผู้ใช้.....	23
บทที่ 4 แผนงานของโครงการ.....	25
4.1 Phase 1.....	25
4.2 Phase 2.....	29
4.3 การดำเนินงานในระยะ phase 1.....	35
4.4 การดำเนินงานในระยะ phase 2.....	36
4.5 การดำเนินงานในระยะ phase 2 บน github.....	36

บทที่ 5 การพัฒนา Mobile Application	37
5.1 User.....	37
5.2 Staff.....	46
5.3 Backend.....	52
5.4 flow(relational) ของ user(student) and staff.....	59
5.5 ยกตัวอย่าง Source Code ที่นำเสนอ.....	63
บทที่ 6 การทดสอบ Mobile Application.....	69
6.1 หลักฐานการทดสอบ Mobile Application.....	70
บทที่ 7 สรุป.....	73
7.1 สรุปผลที่ได้รับจากการพัฒนา Mobile Application.....	73
7.2 สิ่งที่สามารถกลุ่มได้เรียนรู้.....	74
7.2.1 พังก์ชันหลักที่พัฒนา	75
7.2.2 เทคโนโลยีและฟังก์ชัน Firebase ที่ใช้เพิ่มเติม....	82
แหล่งอ้างอิง.....	86

บทที่ 1

รายละเอียดของโครงการ

1.1 ที่มาและความสำคัญของโครงการ

ในปัจจุบัน ปัญหาสุขภาพจิตเป็นประเด็นที่มีความสำคัญในระดับโลก องค์การอนามัยโลก (WHO) รายงานว่ามีประชากรจำนวนมากที่ประสบปัญหาด้านสุขภาพจิต และยังมีข้อจำกัดในการเข้าถึงบริการที่เหมาะสม ปัญหาดังกล่าวเกิดจากปัจจัยหลายประการ เช่น ทรัพยากรด้านสุขภาพจิตที่ไม่เพียงพอ ค่าใช้จ่ายที่สูง และการติด-trafficking ส่งผลให้หลายคนไม่ได้รับการดูแลที่เหมาะสมและนำไปสู่ปัญหาทางสุขภาพจิตที่รุนแรงขึ้น

ปัญหาสุขภาพจิตไม่ได้ส่งผลกระทบต่อบุคคลเท่านั้น แต่ยังส่งผลกระทบต่อเศรษฐกิจ และสังคมโดยรวม บุคคลที่เผชิญกับภาวะซึมเศร้าหรือความเครียดสะสมอาจมีประสิทธิภาพในการทำงานลดลง และอาจนำไปสู่ปัญหาทางสังคม เช่น ความรุนแรงในครอบครัว หรืออัตราการฆ่าตัวตายที่เพิ่มขึ้น ดังนั้น การพัฒนาระบบที่ช่วยสนับสนุนสุขภาพจิตจึงเป็นสิ่งจำเป็นเพื่อช่วยลดปัญหาเหล่านี้

โครงการการทำแอปพลิเคชัน "เพื่อนใจ (AI Mental Health Chatbot)" ได้รับการพัฒนาขึ้นภายใต้รายวิชา ITDS283 Mobile Application Development โดยมีเป้าหมายเพื่อสนับสนุนการพัฒนาที่ยั่งยืน (Sustainable Development Goals: SDGs) ดังต่อไปนี้

Goal 3: Good Health and Well-being – การสร้างหลักประกันให้ทุกคนมีสุขภาวะที่ดี และส่งเสริมความเป็นอยู่ที่ดีในทุกช่วงวัย โดยแอปพลิเคชันช่วยให้ผู้ใช้สามารถเข้าถึงบริการสุขภาพได้ง่ายขึ้น ลดระยะเวลาอุดหนู และได้รับการดูแลเบื้องต้นผ่าน Telemedicine

Goal 10: Reduced Inequality – ลดความไม่เสมอภาคในการเข้าถึงบริการสุขภาพ ระหว่างกลุ่มคนที่มีข้อจำกัดด้านเวลาและสถานที่ เช่น นักศึกษา ผู้สูงอายุ และผู้ที่อาศัยในพื้นที่ห่างไกล

ยังมีเทคโนโลยีปัญญาประดิษฐ์ (AI) ถูกนำมาใช้เพื่อช่วยให้คำปรึกษาด้านสุขภาพจิต เป็นต้นแก่ผู้ใช้งาน และช่วยให้สามารถเข้าถึงแหล่งข้อมูลที่เป็นประโยชน์ได้อย่างสะดวก เพื่อให้สามารถติดต่อกับผู้ใช้ได้แบบอัตโนมัติ ให้คำแนะนำที่เหมาะสม และช่วยลดอุปสรรค ในการเข้าถึงบริการสุขภาพจิต โดยการพัฒนาแอปพลิเคชันบนแพลตฟอร์ม Flutter ทำให้ สามารถใช้งานได้ง่ายบนอุปกรณ์พกพา เพิ่มโอกาสให้ประชาชนกลุ่มต่าง ๆ โดยเฉพาะผู้ที่ขาด แคลนทรัพยากรหรืออยู่ในพื้นที่ห่างไกล สามารถเข้าถึงการดูแลสุขภาพจิตได้อย่างมี ประสิทธิภาพ

1.2 วัตถุประสงค์ของ Mobile Application

แอปพลิเคชัน "เพื่อนใจ" มีวัตถุประสงค์หลักในการให้ความช่วยเหลือด้านสุขภาพจิต แก่ผู้ใช้ผ่านเทคโนโลยีปัญญาประดิษฐ์ โดยเน้นการให้คำปรึกษาเบื้องต้น ลดอุปสรรคในการ เข้าถึงบริการสุขภาพจิต และส่งเสริมการดูแลสุขภาพจิตในระยะยาว

หนึ่งในวัตถุประสงค์หลักคือการให้คำปรึกษาด้านสุขภาพจิตเบื้องต้นแก่ผู้ใช้ผ่าน AI Chatbot ที่สามารถติดต่อกับผู้ใช้ ให้คำแนะนำที่เหมาะสม และช่วยลดความเครียดหรือความวิตกกังวล ของผู้ใช้ เพื่อให้สามารถดูแลสุขภาพจิตของตนเองได้ในเบื้องต้น โดยไม่จำเป็นต้องพบแพทย์ หรือผู้เชี่ยวชาญทันที

อีกวัตถุประสงค์ที่สำคัญคือการเพิ่มการเข้าถึงบริการสุขภาพจิต โดยลดอุปสรรคด้าน เวลา สถานที่ และค่าใช้จ่าย ให้ผู้ใช้สามารถขอคำปรึกษาได้ทุกที่ทุกเวลา โดยเฉพาะกลุ่มที่มี ข้อจำกัดทางเศรษฐกิจหรือภูมิศาสตร์ นอกจากนี้ แอปพลิเคชันยังเชื่อมโยงผู้ใช้ไปยัง แหล่งข้อมูลและผู้เชี่ยวชาญ เช่น ศูนย์บริการสุขภาพจิต สายด่วนช่วยเหลือ หรือแพทย์ ผู้เชี่ยวชาญ เพื่อให้ได้รับการดูแลที่เหมาะสมจากแหล่งที่เชื่อถือได้

การออกแบบแอปพลิเคชันยังคำนึงถึงความปลอดภัยและความเป็นส่วนตัวของผู้ใช้ ระบบถูกออกแบบให้สามารถพูดคุยกับ Chatbot ได้อย่างมั่นใจ โดยไม่ต้องเปิดเผยตัวตน ซึ่ง ช่วยลดความกังวลเกี่ยวกับการถูกติดสินใจสังคมหรือคนรอบข้าง ส่งเสริมให้ผู้ใช้กล้าเข้ามา ปรึกษาและขอความช่วยเหลือมากขึ้น

แอปพลิเคชันได้รับการพัฒนาให้สามารถใช้งานได้ง่าย รองรับหลายภาษา และมีอินเตอร์เฟซที่เป็นมิตรกับผู้ใช้ทุกเพศทุกวัย เพื่อให้แน่ใจว่าทุกคนสามารถเข้าถึงและใช้บริการได้สะดวก อีกทั้งยังมีการพัฒนาเทคโนโลยี AI ให้มีประสิทธิภาพสูงสุด โดยปรับปรุงระบบให้สามารถประมวลผลและวิเคราะห์ข้อมูลได้อย่างแม่นยำ เพื่อให้ Chatbot ตอบสนองได้ถูกต้อง และเหมาะสมกับสถานการณ์ของผู้ใช้ และครอบคลุมประเด็นต่างๆ ต่อไปนี้

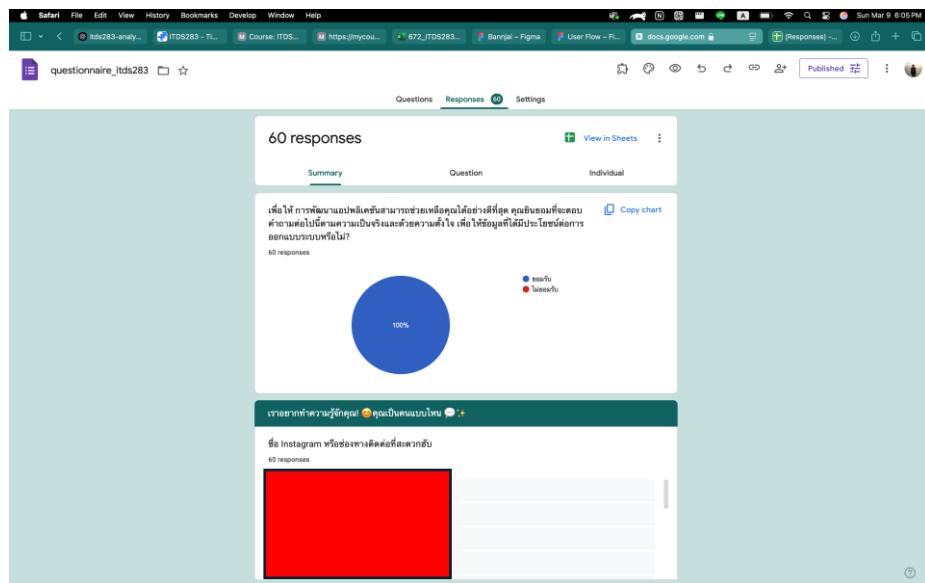
1. พัฒนาระบบ จัดการนัดหมายและจองบริการสุขภาพ ให้สะดวกและรวดเร็วขึ้น
2. สร้างระบบให้คำปรึกษาเบื้องต้นด้านสุขภาพจิตผ่าน AI Chatbot และ Teledicine เพื่อให้ผู้ใช้ได้รับข้อมูลที่เป็นประโยชน์ก่อนเข้ารับบริการจริง
3. ลดระยะเวลาการรอคิวและอำนวยความสะดวกในการเข้าถึงบริการสุขภาพ โดยเฉพาะกลุ่มที่มีข้อจำกัดในการเดินทาง
4. สนับสนุนการให้บริการด้านสุขภาพอย่างเท่าเทียมและเสมอภาค โดยไม่จำกัดสถานที่ และเวลาการเข้าถึง
5. สร้างช่องทางให้ผู้ใช้สามารถ ติดตามประวัติการรักษาและบันทึกสุขภาพส่วนตัว ได้ในแอปเดียว

1.3 กลุ่มเป้าหมายของ Mobile Application

แอปพลิเคชัน เพื่อนใจ (MU Health & Friends Service) ถูกออกแบบมาเพื่อรับผู้ใช้ หลัก 3 กลุ่ม ได้แก่

1. นักศึกษาและบุคลากรในมหาวิทยาลัย - มีข้อจำกัดด้านเวลา และต้องการความสะดวกในการจองคิวพบแพทย์หรือปรึกษาสุขภาพจิต
2. ผู้ที่ต้องการการดูแลสุขภาพจิตเบื้องต้น - อาจยังไม่พร้อมพบแพทย์ แต่ต้องการคำปรึกษาหรือข้อมูลเกี่ยวกับสุขภาพจิตผ่าน AI Chatbot
3. บุคคลทั่วไปที่ต้องการบริการทางการแพทย์แบบสะดวกและรวดเร็ว - เช่น ผู้ที่อาศัยในพื้นที่ห่างไกล หรือไม่สามารถเดินทางไปโรงพยาบาลได้สะดวก

เพื่อให้แอปพลิเคชันตอบโจทย์กลุ่มเป้าหมายอย่างแท้จริง โครงการนี้ได้ศึกษาข้อมูลจากกลุ่มผู้ใช้งานจริง ผ่านแบบสอบถามและการสัมภาษณ์ เพื่อรับรวมความคิดเห็นและปัญหาที่พบในการเข้าถึงบริการสุขภาพ



รูปที่ 1.1 การทำ UX Research – Questionnaire

1.4 ศึกษาเปรียบเทียบกับ Mobile Application อื่นที่มีอยู่ในปัจจุบัน

ปัจจุบันมีแอปพลิเคชันหลายตัวที่ให้บริการด้านสุขภาพและสุขภาพจิตผ่านแพลตฟอร์มดิจิทัล ซึ่งแต่ละแอปมีจุดเด่นและข้อจำกัดที่แตกต่างกัน การศึกษาข้อมูลของแอปพลิเคชันที่มีอยู่ในตลาดจะช่วยให้การพัฒนา เพื่อนใจ (MU Health & Friends Service) มีประสิทธิภาพและสามารถตอบโจทย์ผู้ใช้ได้ดียิ่งขึ้น

หนึ่งในแอปพลิเคชันที่ได้รับความนิยมคือ MorDee (หมอดี) ซึ่งเป็นแพลตฟอร์มที่ช่วยให้ผู้ใช้สามารถค้นหาแพทย์ตามอาการที่ต้องการ รวมถึงสามารถใช้บริการส่งยาถึงบ้านได้ นอกจากนี้ MorDee ยังรองรับการเคลมค่ารักษาพยาบาลกับบริษัทประกัน ทำให้ผู้ใช้สามารถเข้าถึงบริการสุขภาพได้ง่ายขึ้นโดยไม่ต้องสำรองจ่าย อย่างไรก็ตาม ข้อจำกัดของแอปพลิเคชันนี้คือ ค่าใช้จ่ายที่ค่อนข้างสูง และความจำเป็นในการเชื่อมต่ออินเทอร์เน็ตตลอดเวลา ซึ่งอาจเป็นอุปสรรคสำหรับผู้ใช้ที่มีข้อจำกัดด้านงบประมาณหรืออาศัยในพื้นที่ที่อินเทอร์เน็ตไม่เสถียร



รูปที่ 1.2 บริการ MorDee (หมอดี)

ในด้านสุขภาพจิต แอปพลิเคชัน Ooca (อูค้า) เป็นอีกหนึ่งแพลตฟอร์มที่เชื่อมต่อผู้ใช้กับจิตแพทย์และนักจิตวิทยาผ่านวิดีโอลิงค์ โดยให้บริการปรึกษาสุขภาพจิตในลักษณะที่สะดวกและเป็นส่วนตัว นอกเหนือนี้ยังมีบริการฟรีสำหรับเยาวชนในบางพื้นที่ เช่น กรุงเทพมหานคร อย่างไรก็ตาม ข้อเสียของ Ooca คือ บริการฟรีมีข้อจำกัด และค่าบริการปกติค่อนข้างสูง ทำให้ไม่สามารถเข้าถึงกลุ่มเป้าหมายที่มีรายได้น้อยได้ย่างทั่วถึง



รูปที่ 1.3 บริการ Ooca (อุ๊ก้า)

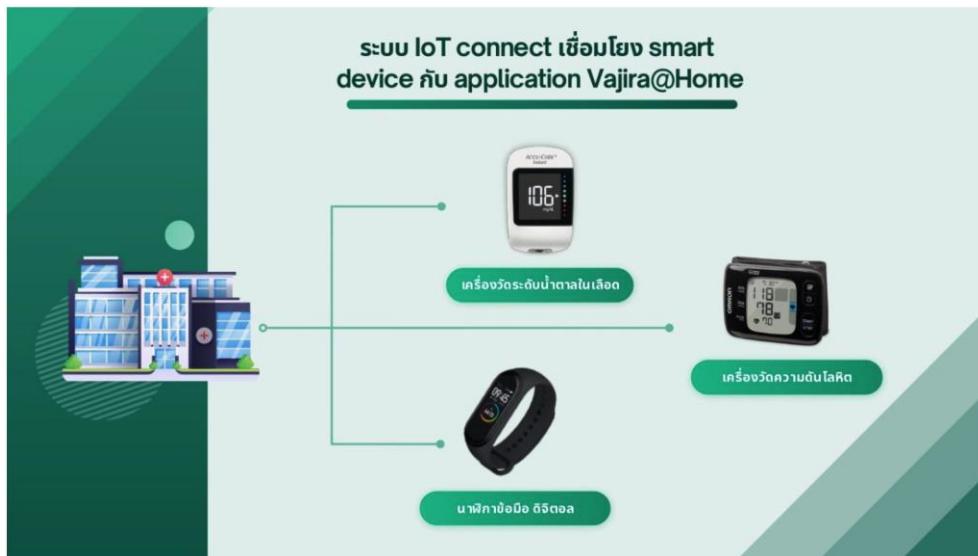
นอกจากนั้น แอปพลิเคชัน Clicknic มีแนวคิดที่คล้ายคลึงกัน โดยเปิดโอกาสให้ผู้ใช้ ปรึกษาแพทย์ผ่านแชทหรือวิดีโอกล้อ และสามารถปรึกษาเภสัชกรเพื่อสั่งซื้อยาผ่านระบบออนไลน์ได้ จุดแข็งของ Clicknic คือการมีฟังก์ชันคัดกรองผู้ติดเชื้อโควิด-19 โดยเจ้าหน้าที่ทางการแพทย์ฟรี อย่างไรก็ตาม แอปนี้ยังมีข้อจำกัดที่คล้ายกับ MorDee และ Ooca นั้น คือ การใช้งานที่ต้องพึ่งพาการเชื่อมต่ออินเทอร์เน็ต รวมถึง ค่าใช้จ่ายในการใช้บริการ ที่อาจไม่เหมาะสมสำหรับทุกกลุ่มเป้าหมาย



รูปที่ 1.4 บริการ คลิกนิก (Clicknic)

แอปพลิเคชัน Vajira@Home เป็นอีกตัวอย่างที่ถูกพัฒนาโดยโรงพยาบาลวิชรพยาบาลเพื่อให้บริการ Telemedicine สำหรับผู้ป่วยโรคไม่ติดต่อเรื้อรัง (NCDs) และผู้ป่วยที่

ต้องการติดตามอาการหลังผ่าตัด ข้อดีของแอปนี้คือช่วยลดภาระการเดินทางของผู้ป่วย ทำให้สามารถรับการดูแลจากแพทย์ได้จากที่บ้าน อย่างไรก็ตาม ข้อจำกัดสำคัญของ Vajira@Home คือ จำกัดเฉพาะผู้ป่วยของโรงพยาบาลวิชรพยาบาลเท่านั้น ซึ่งทำให้ไม่สามารถให้บริการได้ในวงกว้าง



รูปที่ 1.5 บริการ Vajira@Home

AI DMIND บนไลน์ สปสช. ซึ่งเป็นบริการที่ใช้ปัญญาประดิษฐ์ในการช่วยคัดกรองสุขภาพกายและสุขภาพจิต จุดแข็งของ AI DMIND คือ การเข้าถึงง่ายผ่านแอปพลิเคชัน LINE ทำให้ไม่จำเป็นต้องดาวน์โหลดแอปใหม่ อย่างไรก็ตาม ข้อจำกัดของแอปนี้คือ ความสามารถในการวิเคราะห์และให้คำแนะนำนัยยะจำกัดเมื่อเทียบกับแอปพลิเคชันเฉพาะทางด้านสุขภาพจิต



รูปที่ 1.6 บริการ AI DMIND บนไลน์ สปสช.

เมื่อเปรียบเทียบกับแอปพลิเคชันข้างต้น เพื่อนใจ (MU Health & Friends Service) ได้รับการออกแบบให้เป็นแพลตฟอร์มที่รวมเอาจุดแข็งของแต่ละแอปเข้าด้วยกัน ในขณะที่ลดข้อจำกัดที่พัฒนาในแอปพลิเคชันอื่น ๆ แอปนี้มุ่งเน้น การรวมบริการด้านสุขภาพกายและสุขภาพจิตในแอปเดียว พร้อมใช้ AI Chatbot เพื่อช่วยให้คำปรึกษาเบื้องต้นด้านสุขภาพจิต และรองรับ การจองคิวพบแพทย์และ Telemedicine เพื่อช่วยให้ผู้ใช้สามารถเข้าถึงบริการสุขภาพได้ง่ายขึ้น นอกจากนี้ แอปยังถูกพัฒนาโดยใช้ Flutter ซึ่งช่วยให้สามารถใช้งานได้บนทั้ง iOS และ Android อย่างมีประสิทธิภาพ

โดยสรุป แอปพลิเคชัน เพื่อนใจ ถูกพัฒนาขึ้นเพื่อตอบโจทย์ผู้ใช้ในแง่ของ การเข้าถึงที่สะดวก รวดเร็ว และครอบคลุมทั้งสุขภาพกายและสุขภาพจิต ซึ่งทำให้แตกต่างจากแอปพลิเคชันอื่นที่มีอยู่ในตลาดปัจจุบัน

บทที่ 2

User Journey Map

ในการพัฒนา Mobile Application เพื่อให้บริการคำปรึกษาด้านสุขภาพกายและจิตใจ จำเป็นต้องเข้าใจ User Journey ของผู้ใช้ตลอดกระบวนการใช้งาน เพื่อรับปัญหาที่พบและหาแนวทางปรับปรุงให้เหมาะสม โดย User Journey Map แสดงให้เห็นถึงเส้นทางที่ผู้ใช้จะต้องผ่าน ตั้งแต่เริ่มต้นตระหนักถึงปัญหาของตนเอง ค้นหาข้อมูล สมัครใช้งาน ไปจนถึงการให้คำปรึกษาและการกลับมาใช้ซ้ำ โดยสามารถสรุปเป็นตารางได้ดังนี้

User Journey Map:

Phase	Action	Touchpoint	Thought	Feeling	Pain Point	Opportunity
1. การรับรู้ปัญหา	ผู้ใช้รู้สึกไม่สบายใจหรือต้องการคำปรึกษา	ความรู้สึกส่วนตัว, โซเชียลมีเดีย, การบอกร่อง	รู้สึกไม่ดีเลย ควรทำอย่างไรดี	วิตกกังวล, สับสน	ไม่รู้จะปรึกษาใคร, กลัวคนอื่นรู้	สร้างช่องทางให้ผู้ใช้เข้าถึงข้อมูลและคำปรึกษาได้ง่ายและเป็นส่วนตัว
2. การค้นหาข้อมูล	ผู้ใช้ค้นหาข้อมูลเกี่ยวกับแอป "เพื่อนใจ"	App Store, Google Play Store, เว็บไซต์, โซเชียลมีเดีย	แอปนี้ช่วยอะไรได้บ้าง, มีคนใช้แล้วเป็นอย่างไร	สนใจ, อยากรู้	ข้อมูลไม่ชัดเจน, รีวิวไม่น่าเชื่อถือ	ปรับปรุงข้อมูลแอปให้ชัดเจนและน่าเชื่อถือ, แสดงรีวิวจากผู้ใช้จริง
3. การดาวน์โหลดและลงที่เมียน	ผู้ใช้ดาวน์โหลดและลงที่เมียนแอป	App Store, Google Play Store, และ	ลงที่เมียนง่าย, ไม่มีข้อมูลส่วนตัวจะปลดล็อกภายใน	คาดหวัง, กังวล	ขั้นตอนลงที่เมียนชันช์อ่อน, กังวลเรื่องความปลอดภัยของข้อมูล	ออกแบบขั้นตอนลงที่เมียนให้ง่ายและรวดเร็ว, สร้างความมั่นใจเรื่องความปลอดภัยของข้อมูล
4. การใช้งานครั้งแรก	ผู้ใช้สำรวจฟีเจอร์ต่างๆ ของแอป	แอป	มีฟีเจอร์อะไรบ้าง, ใช้งานยากไหม	ตื่นเต้น, อยากลอง	ฟีเจอร์ใช้งานยาก, ไม่เข้าใจวิธีการใช้งาน	ออกแบบ UI/UX ให้ใช้งานง่ายและเป็น user friendly, มีคู่มือการใช้งานหรือวิดีโอแนะนำ
5. การขอคำปรึกษา	ผู้ใช้ขอคำปรึกษาเมื่อต้น	แอป (แท็บ, วิดีโอดอคอล)	จะได้รับคำปรึกษาที่ดีในมี, จะได้รับคำแนะนำที่เป็นประโยชน์ในมี	คาดหวัง, กังวล	รอความนาน, ไม่ได้รับคำปรึกษาที่ตรงจุด	เพิ่มจำนวนผู้ให้คำปรึกษา, พัฒนา AI ในสามารถให้คำปรึกษาเมื่อต้นได้

6. การจองคิว และเรียกคิว	ผู้ใช้จองคิว และเรียกคิว รักษา	แอป	จองคิวง่าย ในเมือง จะรู้ได้ อย่างไรว่าถึง คิวแล้ว	สะดวก, มั่นใจ	ระบบจองคิว ขั้นช่อน, ไม่มี การแจ้งเตือน เมื่อถึงคิว	ออกแบบระบบ จองคิวให้ง่าย และรวดเร็ว, มี ระบบแจ้งเตือน เมื่อถึงคิว
7. การให้ คะแนนและรีวิว และรีวิวแอป	ผู้ใช้ให้คะแนน และรีวิวแอป	แอป	อยากร่วมเป็น ประสบการณ์ การใช้งาน	พึงพอใจ, อยากรอต่อ	ไม่มีช่องทาง ให้รีวิว, ขั้น ตอนรีวิว ขั้นช่อน	สร้างช่องทาง ให้ผู้ใช้รีวิวได้ ง่าย, นำรีวิวไป ปรับปรุงแอป
8. การใช้งาน ต่อเนื่อง	ผู้ใช้กลับมาใช้งานและอีก ครั้ง	แอป	แอปนี้ช่วยจัน ได้จริงๆ	พึงพอใจ, เชื่อ ถูก	ไม่มีที่เจอ รีบมาก, เนื้อหา ไม่เก่าสนใจ	อัปเดตที่เจอ รีบมาก, เนื้อหา อายุมาก

Phase	Action	Touchpoint	Thought	Feeling	Pain Point	Opportunity
						สามารถ ประเมิน ความ ต้องการ ของ ผู้ใช้ ที่ ต้อง ปรับปรุง ให้ ดี ขึ้น

รูปที่ 2.1 ตารางการทำ user journey map ในรูปแบบตาราง

2.1 การวิเคราะห์ User Journey Map สามารถแบ่งออกเป็น 8 Phase หลัก ได้แก่

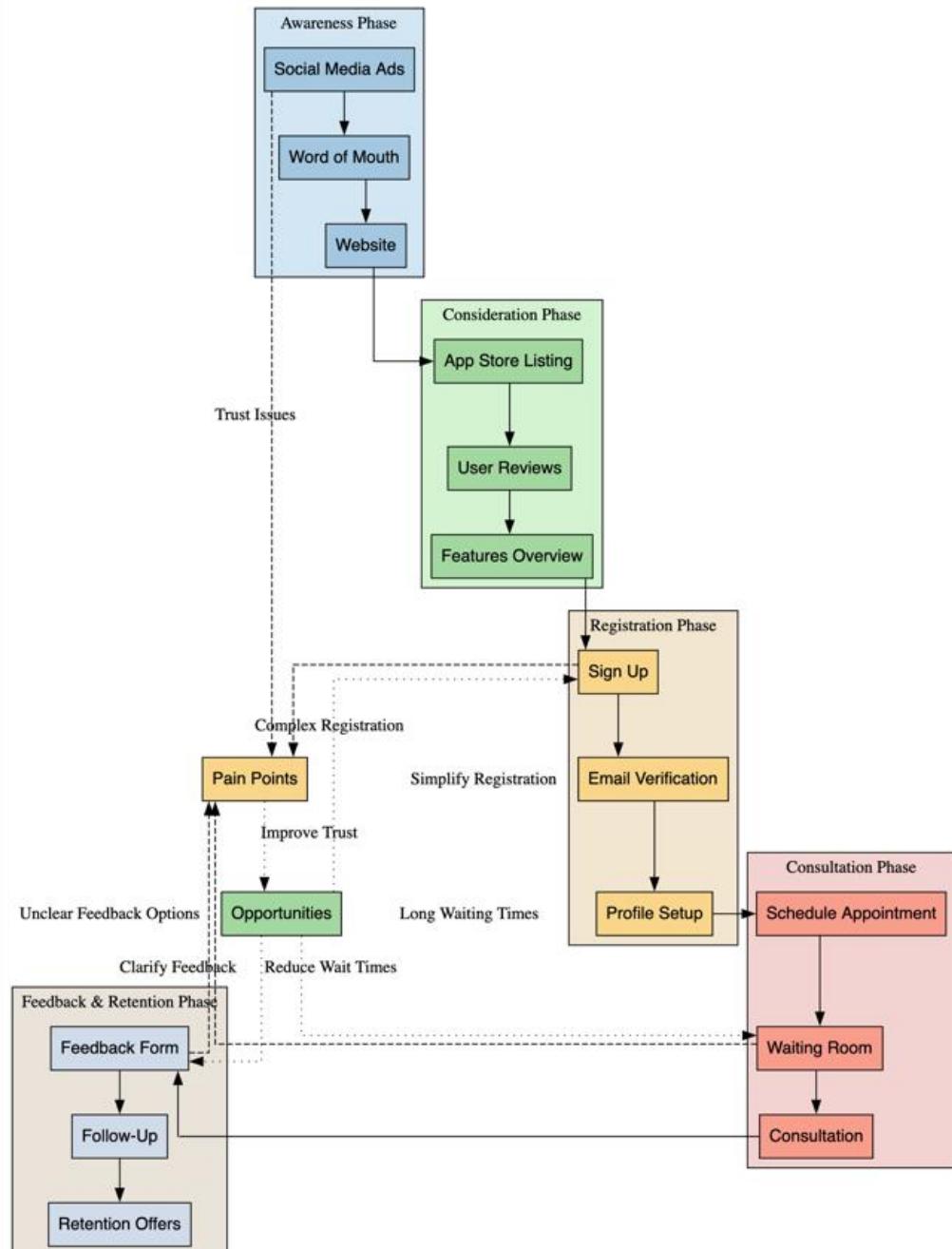
- การรับรู้ปัญหา – ผู้ใช้เริ่มรู้สึกว่าตนเองมีปัญหาทางสุขภาพจิตและต้องการคำปรึกษา แต่ยังไม่แน่ใจว่าควรขอความช่วยเหลือจากใคร จุดอ่อนของเฟสนี้คือ ผู้ใช้ขาดช่องทาง ข้อมูลที่น่าเชื่อถือ ซึ่งอาจทำให้รู้สึกสับสนและไม่กล้าขอความช่วยเหลือ
- การค้นหาข้อมูล – ผู้ใช้ค้นหาแอปพลิเคชันที่สามารถช่วยแก้ปัญหาได้ ผ่าน App Store, Google Play Store, เว็บไซต์ และโซเชียลมีเดีย อย่างไรก็ตาม ผู้ใช้อาจเผชิญ กับปัญหา ข้อมูลไม่ครบถ้วน หรือรีวิวไม่น่าเชื่อถือ ทำให้ลังเลในการดาวน์โหลด
- การดาวน์โหลดและลงทะเบียน – เมื่อผู้ใช้ตัดสินใจดาวน์โหลดแอป อาจพบปัญหานี้ใน การ ลงทะเบียนที่ชับช้อน หรือเกิดความกังวลเรื่อง ความปลอดภัยของข้อมูล ส่วนตัว ทำให้บางคนเลือกที่จะไม่สมัครใช้งาน
- การใช้งานครั้งแรก – ผู้ใช้เริ่มสำรวจฟีเจอร์ต่าง ๆ ของแอป แต่หาก UI/UX ไม่เป็น มิตรหรือไม่มีคำแนะนำในการใช้งานอาจทำให้รู้สึกสับสนและไม่ต้องการใช้งานต่อ
- การขอคำปรึกษา – เมื่อผู้ใช้ต้องการขอคำปรึกษา อาจพบว่า ไม่มีที่ปรึกษาพร้อม ให้บริการ หรือระบบจองคิวล่าช้า ส่งผลให้เกิดความคาดหวังที่ไม่เป็นจริงและเกิด ความรู้สึกกังวล

6. การลองคิดและเรียกคืนคำแนะนำ – ผู้ใช้ต้องการกลับมาดูคำแนะนำที่ได้รับเพื่อประเมินว่าควรทำอย่างไรต่อไป หากแอปไม่มี ฟังก์ชันเก็บบันทึกคำแนะนำที่เข้าถึงได้ง่าย อาจทำให้ผู้ใช้รู้สึกไม่สะดวก
7. การให้คะแนนและรีวิว – ผู้ใช้ต้องการแบ่งปันประสบการณ์ของตนเอง แต่หากแอปไม่มีช่องทางรีวิวที่ชัดเจน หรือขั้นตอนการให้คะแนนซับซ้อน อาจทำให้ไม่สามารถรับฟังความคิดเห็นของผู้ใช้ได้อย่างมีประสิทธิภาพ
8. การใช้งานต่อเนื่อง – หากแอปไม่มีฟีเจอร์ที่ช่วยกระตุนให้ผู้ใช้กลับมาใช้งาน อาจทำให้อัตราการใช้งานช้าลงเนื่องจากผู้ใช้รู้สึกว่าแอปไม่มีคุณค่าในระยะยาว

2.2 แนวทางแก้ไขปัญหาโดย Mobile Application

เพื่อแก้ไขปัญหาที่พบใน User Journey Map แอปพลิเคชันสามารถนำเสนอแนวทางปรับปรุงดังนี้

- ให้ข้อมูลที่น่าเชื่อถือและเข้าถึงได้ง่าย โดยสร้างแพลตฟอร์มที่มีข้อมูลเกี่ยวกับสุขภาพจิต พร้อมรีวิวจากผู้ใช้จริง
- ปรับปรุงระบบลงทะเบียน ให้ใช้งานง่ายขึ้น และสร้างความมั่นใจในด้านความปลอดภัยของข้อมูล
- ออกแบบ UI/UX ที่ใช้งานง่าย และมีคู่มือแนะนำการใช้งานแบบอินเทอร์แอคทีฟ
- พัฒนาระบบ AI หรือที่ปรึกษาอัตโนมัติ เพื่อลดระยะเวลาอคิวและให้ผู้ใช้ได้รับคำแนะนำเบื้องต้น
- จัดทำระบบบันทึกคำแนะนำ เพื่อให้ผู้ใช้สามารถเรียกคืนข้อมูลได้สะดวก
- เพิ่มช่องทางให้คะแนนและรีวิว ที่เข้าใจง่ายและรวดเร็ว
- กระตุนให้เกิดการใช้งานซ้ำ โดยเพิ่มฟีเจอร์ใหม่ ๆ ที่ตอบโจทย์ผู้ใช้ เช่น ระบบติดตามผลหรือกิจกรรมที่ช่วยพัฒนาสุขภาพจิต

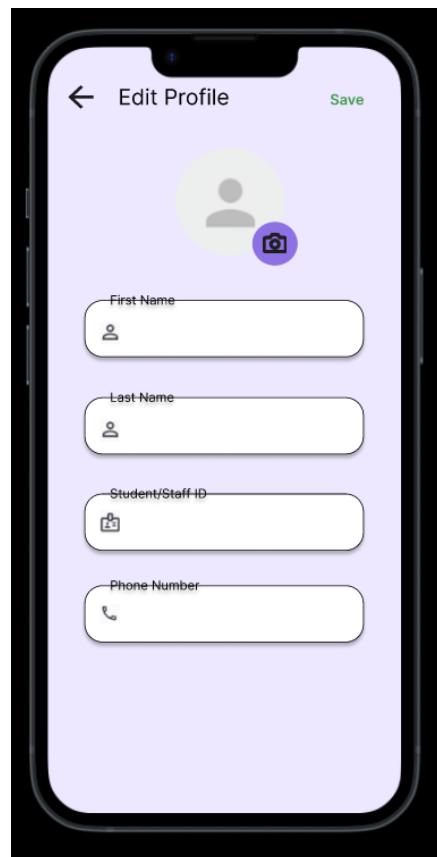


រូបថត 2.2 User Journey Map រូបແບບ Graphviz

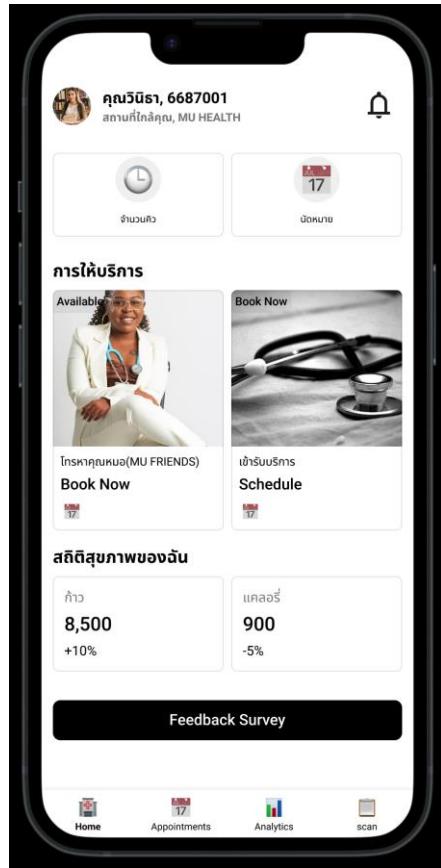
บทที่ 3 User Flow

- ฝ่าย User

หน้าจอ login สำหรับผู้ใช้ (User)



รูปที่ 3.1 Login Page - User Flow ฝั่ง User



รูปที่ 3.2 Homepage - User Flow ผู้ใช้ User

หน้าจอหลักสำหรับผู้ใช้ (User) ซึ่งมีฟังก์ชันหลักๆ ดังนี้:

1. ข้อมูลส่วนตัวผู้ใช้:

- แสดงชื่อผู้ใช้ "คุณวินิชา, 6687001"
- แสดงข้อมูลโรงพยาบาลหรือหน่วยงาน "คลินิกเพื่อนใจ, MU HEALTH"
- แสดงรูปโปรไฟล์ผู้ใช้
- มีโหมดแจ้งเตือน (Notification)

2. ข้อมูลการนัดหมาย:

- แสดงเวลาทำงานพิเศษ (Hours)
- แสดงวันที่นัดหมาย (Date)

3. การให้บริการ (Services Offered):

- แสดงบริการที่มีให้เลือก 2 บริการหลัก:
 - บริการแรกเป็นรูปภาพบุคคล (คาดว่าเป็นบริการปรึกษา) พร้อมข้อความ "คลินิกเพื่อนใจ(MU FRIENDS)" และปุ่ม "Book Now"
 - บริการที่สองเป็นรูปเครื่องมือแพทย์ (คาดว่าเป็นบริการนัดหมาย) พร้อมข้อความ "เจ้ายังเทพ" และปุ่ม "Schedule"
- มีอุปกรณ์แสดงสถานะบริการ (คาดว่าเป็นสถานะ "Available")

4. สถิติสุขภาพของฉัน (Your Health Statistics):

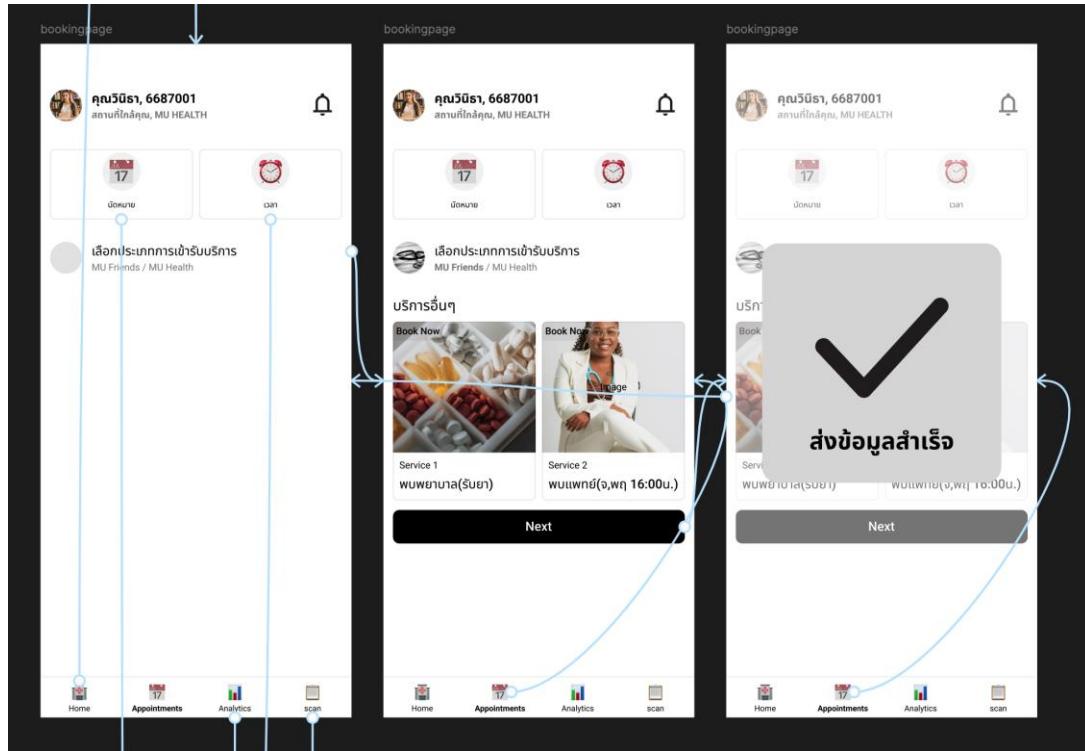
- แสดงข้อมูลสุขภาพ 2 รายการ:
 - รายการแรกแสดง "ก้าว" พร้อมตัวเลข "8,500" และเปอร์เซ็นต์การเปลี่ยนแปลง "+10%"
 - รายการที่สองแสดง "แคลอรี่" พร้อมตัวเลข "900" และเปอร์เซ็นต์การเปลี่ยนแปลง "-5%"

5. ปุ่มตอบแบบสำรวจความคิดเห็น (Feedback Survey):

- มีปุ่ม "Feedback Survey" เพื่อให้ผู้ใช้แสดงความคิดเห็น

6. แถบนำทางด้านล่าง (Bottom Navigation Bar):

- มีอุปกรณ์ "Home" (หน้าหลัก)
- มีอุปกรณ์ "Appointments" (นัดหมาย)
- มีอุปกรณ์ "Analytics" (สถิติ)
- มีอุปกรณ์ "Chat" (สนทนา)



รูปที่ 3.3 Booking page - User Flow ผู้ใช้ User

แสดงขั้นตอนการจองบริการสำหรับผู้ใช้ (User) โดยมีพังก์ชันหลักๆ ดังนี้:

1. หน้าจอแรก: เลือกประเภทการเข้ารับบริการ

- **ข้อมูลส่วนตัวผู้ใช้:**
 - แสดงชื่อผู้ใช้ "คุณวิชิต, 6687001"
 - แสดงข้อมูลโรงพยาบาลหรือหน่วยงาน "คลินิกเพื่อนใจ, MU HEALTH"
 - แสดงรูปโปรไฟล์ผู้ใช้
 - มีไอคอนแจ้งเตือน (Notification)
- **ข้อมูลการนัดหมาย:**
 - แสดงไอคอนปฏิทินพร้อมวันที่ (17)
 - แสดงไอคอนนาฬิกา (Time)
- **เลือกประเภทการเข้ารับบริการ:**
 - มีข้อความ "เลือกประเภทการเข้ารับบริการ"

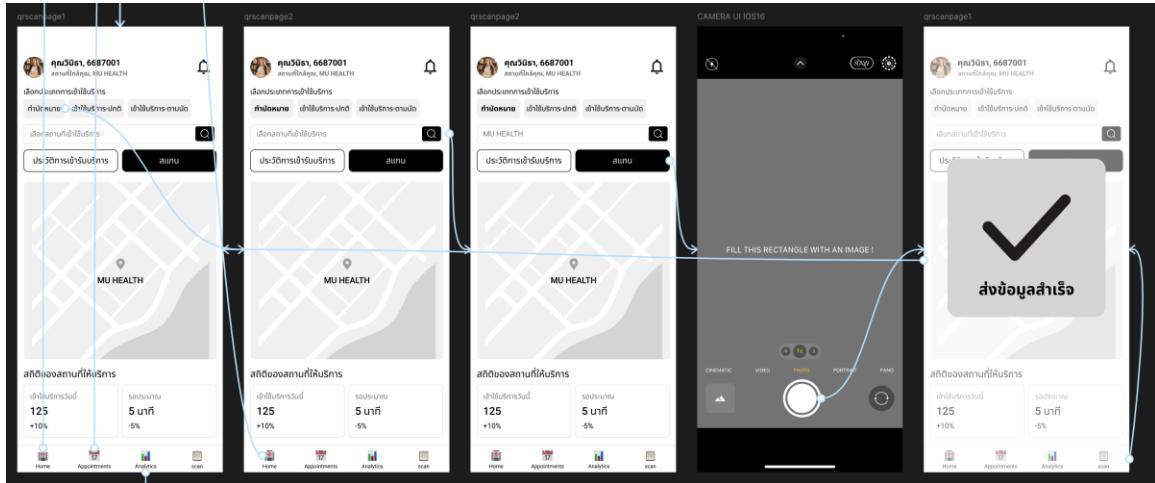
- มีข้อความ "MU Friends / MU Health" (คาดว่าเป็นชื่อคลินิกหรือบริการ)
- มีวงกลมให้เลือก (คาดว่าเป็นการเลือกประเภทบริการหลัก)
- ปุ่มกดไป (Next):
 - มีปุ่ม "Next" เพื่อไปยังขั้นตอนถัดไป

2. หน้าจอที่สอง: เลือกบริการอื่นๆ

- ข้อมูลส่วนตัวผู้ใช้: (เหมือนกับหน้าจอแรก)
- ข้อมูลการนัดหมาย: (เหมือนกับหน้าจอแรก)
- เลือกประเภทการเข้ารับบริการ: (เหมือนกับหน้าจอแรก)
- บริการอื่นๆ (Other Services):
 - แสดงรายการบริการอื่นๆ ที่มีให้เลือก 2 บริการ (Service 1 และ Service 2)
 - บริการแรก "พบพยาบาล (รับยา)" มีรูปภาพและรายละเอียด
 - บริการที่สอง "พบแพทย์ (จันทร์ 16:00 น.)" มีรูปภาพและรายละเอียด
 - มีปุ่ม "Book Now" สำหรับจองบริการแต่ละรายการ
- ปุ่มกดไป (Next):
 - มีปุ่ม "Next" เพื่อไปยังขั้นตอนถัดไป

3. หน้าจอที่สาม: ส่งข้อมูลสำเร็จ

- ข้อมูลส่วนตัวผู้ใช้: (เหมือนกับหน้าจอแรก)
- ข้อมูลการนัดหมาย: (เหมือนกับหน้าจอแรก)
- เลือกประเภทการเข้ารับบริการ: (เหมือนกับหน้าจอแรก)
- บริการอื่นๆ (Other Services): (เหมือนกับหน้าจอที่สอง)
- ส่งข้อมูลสำเร็จ (Data Sent Successfully):
 - มีไอคอนเครื่องหมายถูกสีเขียวขนาดใหญ่
 - มีข้อความ "ส่งข้อมูลสำเร็จ"
- ปุ่มกดไป (Next):
 - มีปุ่ม "Next" เพื่อไปยังขั้นตอนถัดไป (คาดว่าเป็นการกลับสู่หน้าหลัก)



รูปที่ 3.4 QR Scan page - User Flow ผู้ใช้

User Flow: การสแกน QR Code เพื่อเข้าใช้บริการ

1. หน้าหลักของผู้ใช้ (User Main Page)

- ผู้ใช้เข้าถึงหน้าหลักที่แสดงข้อมูลของผู้ใช้ รวมถึงปุ่มสำหรับ "ประวัติการรับบริการ" และ "สแกน"
- มีแผนที่แสดงตำแหน่ง NU HEALTH พร้อมรายละเอียดสถิติของการเข้ารับบริการ เช่น จำนวนครั้งและเวลาที่ใช้

2. การกดปุ่ม "สแกน" (QR Scan Trigger)

- เมื่อผู้ใช้กดปุ่ม "สแกน" ระบบจะนำไปยังหน้าสแกน QR Code

3. หน้าสแกน QR Code (QR Code Scanner)

- ระบบเปิด UI ของกล้องสำหรับสแกน QR Code
- มีคำแนะนำให้ผู้ใช้จัดตำแหน่ง QR Code ให้อยู่ภายในการอบที่กำหนด

4. การยืนยันข้อมูล (Confirmation Page)

- หากสแกนสำเร็จ ระบบจะนำผู้ใช้กลับไปยังหน้าหลัก พร้อมแสดงข้อความยืนยันว่า "ส่งข้อมูลสำเร็จ" พร้อมเครื่องหมายถูก
- ข้อมูลที่สแกนได้จะถูกบันทึกไว้ และอาจอัปเดตสถิติการใช้บริการ

5. กลับสู่หน้าหลัก

- ผู้ใช้สามารถดูข้อมูลอัปเดตเกี่ยวกับการใช้บริการได้ หรือทำการอื่นต่อไป



รูปที่ 3.5 Notification page - User Flow ผู้ใช้

User Flow: หน้าการแจ้งเตือนของผู้ใช้

1. ผู้ใช้เข้าสู่หน้าการแจ้งเตือน

- ผู้ใช้เข้าสู่หน้าจอจากแถบ Navigation Bar ด้านล่าง
- แสดงข้อมูลส่วนตัว เช่น ชื่อและสถานที่ปัจจุบัน (MU HEALTH)
- แสดงจำนวนครั้งที่เข้ารับบริการและจำนวนการนัดหมาย

2. การดูสถานะคิว (Queue Status)

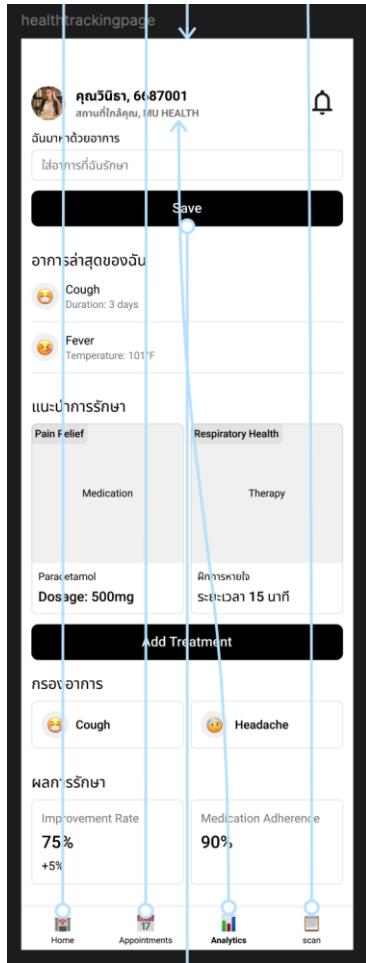
- มีการแสดงสถานะของคิวที่ต้องรับบริการ
- ตัวอย่างเช่น:
 - คิวที่ 1: อยู่ในสถานะ "รอรับบริการ"
 - คิวที่ 2: อยู่ในสถานะ "กำลังรับบริการ"
- ผู้ใช้สามารถดูรายละเอียดเพิ่มเติมหรือยกเลิกคิวที่ยังไม่ได้รับบริการ

3. การดูบัตรนัดหมายล่วงหน้า (Upcoming Appointments)

- แสดงรายการนัดหมายที่กำลังจะถึง เช่น
 - Dentist Appointment กับ Dr. Smith (12th June)
 - Medical Checkup ที่โรงพยาบาล (15th June)
- ผู้ใช้สามารถจัดการหรือดูรายละเอียดเพิ่มเติมเกี่ยวกับนัดหมาย

4. การนำทางไปยังหน้าอื่น (Navigation Bar)

- ด้านล่างมี Navigation Bar ซึ่งมี 4 เมนูหลัก ได้แก่:
 - Home - กลับไปยังหน้าหลัก
 - Appointments - ดูและจัดการนัดหมาย
 - Analytics - ดูสถิติและประวัติการใช้บริการ
 - Scan - ใช้กล้องสแกน QR Code



รูปที่ 3.6 Health Tracking page - User Flow ฝั่ง User

User Flow: หน้าติดตามสุขภาพของผู้ใช้

1. ผู้ใช้เข้าสู่หน้าติดตามสุขภาพ (Health Tracking Page)

- แสดงข้อมูลส่วนตัว เช่น ชื่อและสถานที่ปัจจุบัน (MU HEALTH)
- มีช่องค้นหาข้อมูลที่ต้องการดูย้อนหลัง
- ปุ่ม "Save" สำหรับบันทึกข้อมูลสุขภาพ

2. แสดงอาการล่าสุดของผู้ใช้ (Recent Symptoms)

- รายงานอาการล่าสุด เช่น

- Cough (ไอ) เป็นเวลา 3 วัน
- Fever (มีไข้) อุณหภูมิ 101°F
- ใช้อุปกรณ์ไม่มีจิ่งหองให้เข้าใจสถานะสุขภาพได้ง่ายขึ้น

3. แนะนำการรักษา (Treatment Recommendations)

- ระบบแนะนำวิธีการรักษาแบ่งเป็น 2 หมวดหมู่:
 - Pain Relief (บรรเทาปวด): เช่น การใช้ยา Paracetamol ขนาด 500mg
 - Respiratory Health (สุขภาพทางเดินหายใจ): เช่น การทำ Breathing Therapy เป็นเวลา 15 นาที
- มีปุ่ม "Add Treatment" เพื่อเพิ่มแนวทางการรักษาใหม่

4. กรองอาการที่เกี่ยวข้อง (Filter Symptoms)

- ระบบสามารถกรองอาการ เช่น
 - Cough
 - Headache

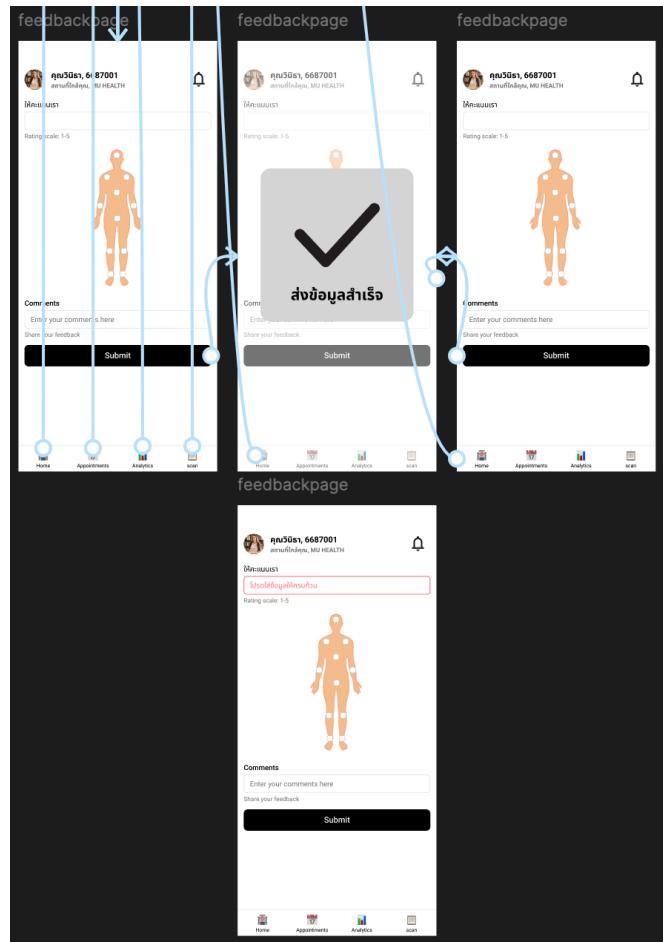
5. แสดงผลลัพธ์ของการรักษา (Treatment Results)

- มีสถิติเกี่ยวกับผลลัพธ์การรักษา เช่น
 - Improvement Rate (อัตราการดีขึ้น): 75% (+5% จากเดิม)
 - Medication Adherence (ความสม่ำเสมอในการใช้ยา): 90%

6. การนำทางไปยังหน้าอื่น (Navigation Bar)

- แถบด้านล่างประกอบด้วย 4 เมนูหลัก:
 1. Home - กลับไปยังหน้าหลัก
 2. Appointments - ดูและจัดการนัดหมาย
 3. Analytics - ดูสถิติสุขภาพโดยละเอียด

4. Scan - ใช้กล้องสแกน QR Code



รูปที่ 3.7 Feedback page - User Flow ผู้ใช้

User Flow: หน้าสำหรับให้ความคิดเห็นเกี่ยวกับบริการด้านสุขภาพ

1. ผู้ใช้เข้าสู่หน้า Feedback Page

- แสดงข้อมูลส่วนตัว เช่น ชื่อและสถานที่ปัจจุบัน (MU HEALTH)
- มีช่องให้ผู้ใช้กรอกคะแนน (Rating scale: 1-5 หรือ 1-10)

- แสดงไม่เดลร่างกายของมนุษย์ ซึ่งอาจใช้ระบุดที่มีอาการหรือความพึงพอใจเกี่ยวกับการรักษา

2. ผู้ใช้กรอกข้อมูลฟีดแบ็ค

- ผู้ใช้สามารถใส่คะแนน (Rating score)
- มีช่อง "Comments" ให้กรอกความคิดเห็นเพิ่มเติม
- ปุ่ม "Submit" เพื่อส่งฟีดแบ็ค

3. การส่งข้อมูลสำเร็จ

- หากผู้ใช้กรอกข้อมูลครบและกด "Submit" ระบบจะแสดงหน้าแจ้งเตือนว่า "ส่งข้อมูลสำเร็จ" พร้อมเครื่องหมายถูก
- ผู้ใช้สามารถกลับไปยังหน้าหลักหรือเมนูอื่น ๆ ผ่าน Navigation Bar

4. กรณีที่ผู้ใช้กรอกข้อมูลไม่ครบ

- หากผู้ใช้ไม่ได้ให้คะแนน ระบบจะแสดงแจ้งเตือนเป็น แถบสีแดง ระบุว่า "โปรดให้คะแนนก่อนส่งฟีดแบ็ค"
- ผู้ใช้ต้องกลับไปแก้ไขก่อนกด "Submit" ได้สำเร็จ

5. การนำทางไปยังหน้าอื่น (Navigation Bar)

- ด้านล่างมี Navigation Bar ซึ่งมี 4 เมนูหลัก ได้แก่:
 - Home - กลับไปยังหน้าหลัก
 - Appointments - ดูและจัดการนัดหมาย
 - Analytics - ดูสถิติและข้อมูลสุขภาพ
 - Scan - ใช้กล้องสแกน QR Code

บทที่ 4

แผนงานของโครงการ

เพื่อให้การพัฒนา Mobile Application เป็นไปอย่างมีประสิทธิภาพและสามารถจัดการงานได้อย่างเป็นระบบ โครงการนี้ใช้แนวทาง Scrum Framework ในการบริหารโครงการ โดยแบ่งการดำเนินงานออกเป็น 5 รอบ (Sprints) แต่ละรอบมีขอบเขตงานที่ชัดเจน พร้อมทั้งกำหนดระยะเวลาและผู้รับผิดชอบที่เหมาะสม นอกเหนือจากนี้ ระบบ Jira ถูกนำมาใช้เป็นเครื่องมือในการติดตามความคืบหน้าของงาน รวมถึงการแบ่งงานและการประเมินผลในแต่ละ Sprint และแบ่งเป็น 2 Phrase ได้แก่

Phrase 1

Sprint 1: การเตรียมความพร้อมงาน UX Research (21-23 กุมภาพันธ์ 2025)

วัตถุประสงค์ของ Sprint นี้คือการจัดเตรียมเครื่องมือและเอกสารที่จำเป็นสำหรับการวิจัย UX เพื่อให้มั่นใจว่าการเก็บข้อมูลในขั้นตอนต่อไปจะดำเนินไปอย่างราบรื่น งานที่ต้องดำเนินการประกอบด้วย:

- จัดทำ UX Research Questionnaire ฉบับที่ 1
- จัดเตรียม ใบขอความอนุเคราะห์ เพื่อใช้ในการขออนุญาตเก็บข้อมูล
- จัดทำ ใบยินยอม (Consent Form) สำหรับผู้เข้าร่วมวิจัย
- ออกแบบแบบสอบถามให้มีความครบถ้วนและสอดคล้องกับเป้าหมายของโครงการ
- ทดสอบแบบสอบถามกับกลุ่มตัวอย่างขนาดเล็กเพื่อตรวจสอบความเข้าใจ (**ดำเนินการช่วง เม.ย.**)
- ปรับปรุงแบบสอบถามตามผลการทดสอบ
- จัดเตรียมระบบสำหรับจัดเก็บและประมวลผลข้อมูลที่ได้รับจากแบบสอบถาม

Sprint 2: การเก็บข้อมูล UX Research Questionnaire และ Competition Analysis (24-28 กุมภาพันธ์ 2025)

หลังจากเตรียมแบบสอบถามเรียบร้อยแล้ว จะเข้าสู่กระบวนการเก็บข้อมูลจากกลุ่มเป้าหมายจริง โดยมีกระบวนการดังต่อไปนี้:

- ส่งใบขอความอนุเคราะห์ไปยังหน่วยงานหรือองค์กรที่เกี่ยวข้อง
- ติดตามผลการขออนุเคราะห์ และดำเนินการตามข้อกำหนดของแต่ละหน่วยงาน
- ดำเนินการเก็บข้อมูล จากกลุ่มเป้าหมายที่กำหนดไว้
- ตรวจสอบความถูกต้องของข้อมูลที่ได้รับ เพื่อให้มั่นใจว่าข้อมูลที่ได้ไม่มีข้อผิดพลาด
- วิเคราะห์ข้อมูลที่เก็บได้ เพื่อนำไปใช้ในการออกแบบ User Flow และ User Journey Map
- **Competition Analysis** ศึกษาและวิเคราะห์แอปพลิเคชันคู่แข่ง เพื่อให้เข้าใจแนวทางการออกแบบ UX/UI ที่เหมาะสม

Sprint 3: การออกแบบ User Flow และ User Journey Map (1-2 มีนาคม 2025)

ข้อมูลที่ได้จาก UX Research จะถูกนำมาใช้ในการออกแบบ User Flow และ User Journey Map ซึ่งเป็นขั้นตอนสำคัญที่ช่วยให้เข้าใจประสบการณ์ของผู้ใช้งานอย่างเป็นระบบ โดยมีกระบวนการดังต่อไปนี้:

- ออกแบบ User Flow ที่แสดงลำดับการใช้งานของผู้ใช้ในแอปพลิเคชัน
- จัดทำ User Journey Map เพื่อนำเสนอประสบการณ์ของผู้ใช้ในแต่ละขั้นตอน
- ทบทวนและปรับปรุงการออกแบบให้เหมาะสมกับข้อมูลที่ได้จาก UX Research

Sprint 4: การออกแบบ Clickable Prototype ด้วย Figma (3-8 มีนาคม 2025)

หลังจากกำหนด User Flow และ User Journey Map แล้ว ขั้นตอนถัดไปคือการออกแบบต้นแบบของแอปพลิเคชันโดยใช้ Figma เพื่อสร้าง Clickable Prototype ซึ่งจะช่วยให้สามารถลองใช้งานแอปได้ก่อนการพัฒนา โดยมีงานที่ต้องดำเนินการดังนี้:

- ออกแบบ Wireframe สำหรับแต่ละหน้าจอของแอปพลิเคชัน

- พัฒนา UI Design ให้มีความสวยงามและใช้งานง่าย
- สร้าง Interactive Prototype บน Figma เพื่อทดสอบประสบการณ์ของผู้ใช้
- ทดสอบต้นแบบกับผู้ใช้กลุ่มตัวอย่าง และปรับปรุงตามผลตอบรับ

Sprint 5: จัดทำรายงานและนำเสนอผลลัพธ์ (9 มีนาคม 2025)

ในช่วงสุดท้ายของโครงการ จะมีการจัดทำรายงานสรุปผลการดำเนินงานของโครงการ โดยครอบคลุมถึง:

- บทสรุปของกระบวนการพัฒนา Mobile Application
- ข้อมูลเชิงลึกจาก UX Research และการออกแบบ
- การนำเสนอ Clickable Prototype
- รายงานผลการศึกษาและข้อเสนอแนะสำหรับการพัฒนาเพิ่มเติม

โดยผู้รับผิดชอบในงานเอกสารดำเนินงานรายงาน, งาน Project Management, งาน UX Research, งานสร้าง User Flow Diagram (User & Staff) & Wireframe(User) และงาน Prototype Clickable User Flow (User), พัฒนา

ดำเนินการโดย พิชิตชัย แพเจริญชัย

ในงาน User Journey Map, จัดทำเอกสารขอความอนุเคราะห์ และใบแสดงความยินยอม (Consent Form), งานวิเคราะห์ Competition Analysis, งาน วิเคราะห์ User Journey, งาน User Wireframe (Staff) และงาน Prototype Clickable User Flow (Staff)

ดำเนินการโดย ฉัตรดันนัย เอมพันธ์

การใช้ Jira ในการบริหารโครงการ

Jira ถูกนำมาใช้เป็นเครื่องมือหลักในการบริหารจัดการโครงการ โดยใช้แนวทาง Scrum ซึ่งประกอบด้วย Backlog, Sprint Planning, Daily Standups, และ Sprint Review เพื่อให้สามารถติดตามความคืบหน้าของงานได้อย่างมีประสิทธิภาพ ฟีเจอร์หลักที่ใช้ได้แก่:

- **Backlog Management:** จัดลำดับความสำคัญของงาน
- **Sprint Boards:** ใช้ Kanban Board เพื่อติดตามสถานะของงานแต่ละส่วน
- **Task Assignment:** กำหนดผู้รับผิดชอบของแต่ละ Task อย่างชัดเจน
- **Reports & Analytics:** ใช้ Burndown Chart และ Velocity Chart เพื่อตรวจสอบความคืบหน้า

Phrase 2

Sprint 1: การเตรียมความพร้อมโดยการสร้างและออกแบบ Repository บน Github (20 มีนาคม 2025)

วัตถุประสงค์ของ Sprint นี้คือการจัดเตรียมเครื่องมือที่จำเป็นสำหรับการดำเนินการโครงการ จัดทำแอปพลิเคชัน เพื่อให้มั่นใจว่าขั้นตอนต่อไปจะดำเนินไปอย่างราบรื่น งานที่ต้องดำเนินการประกอบด้วย:

<https://github.com/Chatra-Danai/ICT-Mahidol-672-ITDS283-Group8-MU-Health-Friends-Service.git>

- การสร้าง Repository
- จัดเตรียมชื่อสอดคล้องกับงานที่ได้รับมอบหมายให้เป็นไป ตามมาตรฐานการใช้งาน
- เพิ่มชื่อสมาชิกในกลุ่ม
- สมาชิกในกลุ่มเข้าร่วม (25 มีนาคม 2568)

ดำเนินการโดย ฉัตรดนัย เอมพันธุ์

Sprint 2: การพัฒนา (20 มีนาคม – 14 เมษายน 2025)

หลังจากเตรียมทุกอย่างเรียบร้อยแล้ว จึงเริ่มดำเนินการพัฒนาแอปพลิเคชัน :

- Home page
- Booking Page with Map Navigation
- Feedback Page
- Qr page

พัฒนาโดย ฉัตรดนัย เอมพันธุ์

- Log in Page
- การติดตามผลสุขภาพผู้ใช้
- Notification Page

พัฒนาโดย พิชิตชัย แพเจริญชัย

Sprint 3: การเชื่อมต่อระบบฐานข้อมูล (10 – 17 เมษายน 2025)

โดยเลือกเป็นการใช้ firebase

Sprint 4: การทดสอบแอปพลิเคชัน (18 เมษายน 2025)

โดยดำเนินการให้กลุ่มเป้าหมายได้แก่ เจ้าหน้าที่ MU Health, นักศึกษาชั้นปีที่ 3 หลักสูตร ICT, นักศึกษาผู้เข้ารับบริการที่ MU Health:



รูปที่ 4.1 ภาพกลุ่มเป้าหมายทดลองระบบจริง

Sprint 5: จัดทำรายงานและนำเสนอผลลัพธ์ (19 มีนาคม 2025)

ในช่วงสุดท้ายของการ จะมีการจัดทำรายงานสรุปผลการดำเนินงานของโครงการ โดย

โดยผู้รับผิดชอบใน งาน Project Management Phase 1, งาน UX Research, งานสร้าง User Flow Diagram (User & Staff) & Wireframe(User) และงาน Prototype Clickable User Flow (User), การเขียนต่อระบบฐานข้อมูล, พัฒนาหน้า Log in Page, การติดตามผล สุขภาพผู้ใช้, Notification Page

ดำเนินการโดย พิชิตชัย แพเจริญชัย

โดยผู้รับผิดชอบใน งาน Project Management Phase 2, งานเอกสารดำเนินงานรายงาน, งาน User Journey Map, จัดทำเอกสารขอความอนุเคราะห์ และใบแสดงความยินยอม (Consent Form) , งานวิเคราะห์ Competition Analysis, งานสร้าง User Flow Diagram งาน User Wireframe (Staff) และงาน Prototype Clickable User Flow (Staff), พัฒนา Home page, Booking Page with Map Navigation, Feedback Page, Qr page

ดำเนินการโดย ฉัตรดันัย เออมพันธุ์

การใช้ Jira ในการบริหารโครงการ ใน Phase 1

Jira ถูกนำมาใช้เป็นเครื่องมือหลักในการบริหารจัดการโครงการ โดยใช้แนวทาง Scrum ซึ่งประกอบด้วย Backlog, Sprint Planning, Daily Standups, และ Sprint Review เพื่อให้สามารถติดตามความคืบหน้าของงานได้อย่างมีประสิทธิภาพ ฟีเจอร์หลักที่ใช้ได้แก่:

- **Backlog Management:** จัดลำดับความสำคัญของงาน
- **Sprint Boards:** ใช้ Kanban Board เพื่อติดตามสถานะของงานแต่ละส่วน
- **Task Assignment:** กำหนดผู้รับผิดชอบของแต่ละ Task อย่างชัดเจน
- **Reports & Analytics:** ใช้ Burndown Chart และ Velocity Chart เพื่อตรวจสอบความคืบหน้า

การใช้ instagantt และ github ในการบริหารโครงการ ใน Phase 2

การผสานพลังของ Instagantt และ GitHub ในการบริหารโครงการระยะที่สองนั้น นำมาซึ่ง ข้อได้เปรียบที่หลากหลายและส่งเสริมประสิทธิภาพในทุกมิติของการทำงาน ประการแรกคือ ความชัดเจนของภาพรวมโครงการที่ได้รับการยกระดับขึ้นอย่างมาก Instagantt ทำหน้าที่เสน่ห์อน แผนที่นำทางที่มองเห็นได้ ช่วยให้ทีมและผู้มีส่วนได้ส่วนเสียเข้าใจ ลำดับการทำงาน ความเชื่อมโยงของ Task และความคืบหน้าโดยรวมได้อย่างง่ายดาย ในขณะเดียวกัน GitHub จะเป็น ศูนย์กลางบัญชาการสำหรับการพัฒนา ที่บันทึกทุกการเปลี่ยนแปลงของโค้ด และสถานะของงานทางเทคนิคอย่างละเอียด

ข้อดีที่สำคัญอีกประการคือ การทำงานร่วมกันที่มีประสิทธิภาพมากยิ่งขึ้น Instagantt ช่วยให้การ มอบหมายงาน กำหนดผู้รับผิดชอบ และติดตามความคืบหน้าเป็นไปอย่างโปร่งใส ทุกคนในทีมสามารถเห็น สิ่งที่ต้องทำ ใครกำลังทำ และสถานะปัจจุบันของงาน ส่วน GitHub ส่งเสริมการทำงานร่วมกันของนักพัฒนา ผ่านระบบ Branching, Pull Requests และ Code Review ทำให้การ พัฒนาโค้ดเป็นไปอย่างมีคุณภาพและลดข้อผิดพลาด

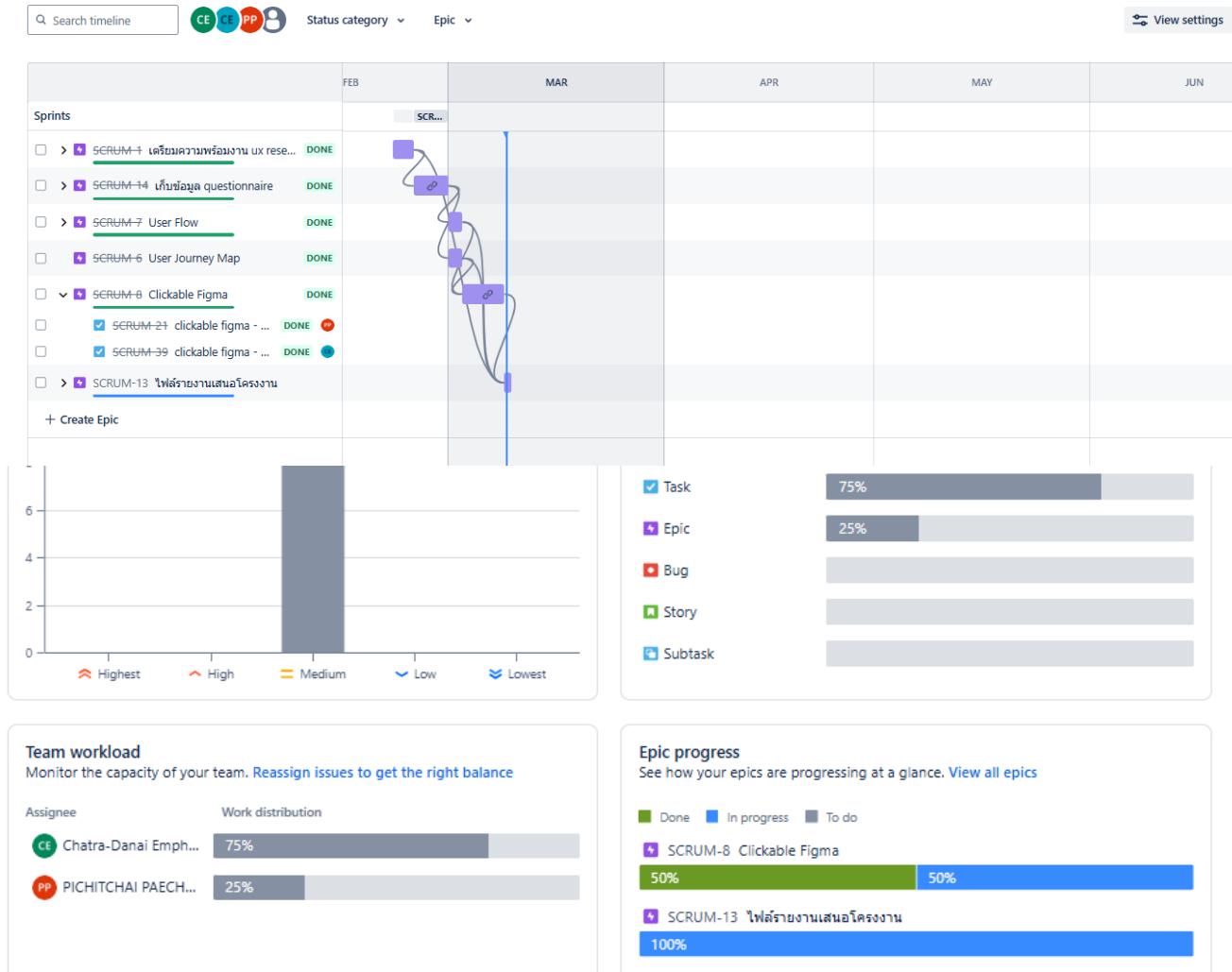
นอกจากนี้ การใช้เครื่องมือทั้งสองร่วมกันยังนำไปสู่ การติดตามความคืบหน้าที่แม่นยำ และครอบคลุม Instagantt ช่วยให้ ติดตามความคืบหน้าตามแผนเวลา ในขณะที่ GitHub สะท้อนความคืบหน้าของการพัฒนาโค้ดจริง การเชื่อมโยงข้อมูลจากทั้งสองแหล่งช่วยให้ประเมินสถานะโครงการได้อย่างรอบด้าน และสามารถ ระบุความเสี่ยงหรือปัญหาที่อาจเกิดขึ้นได้ล่วงหน้า

Projects / ITDS283

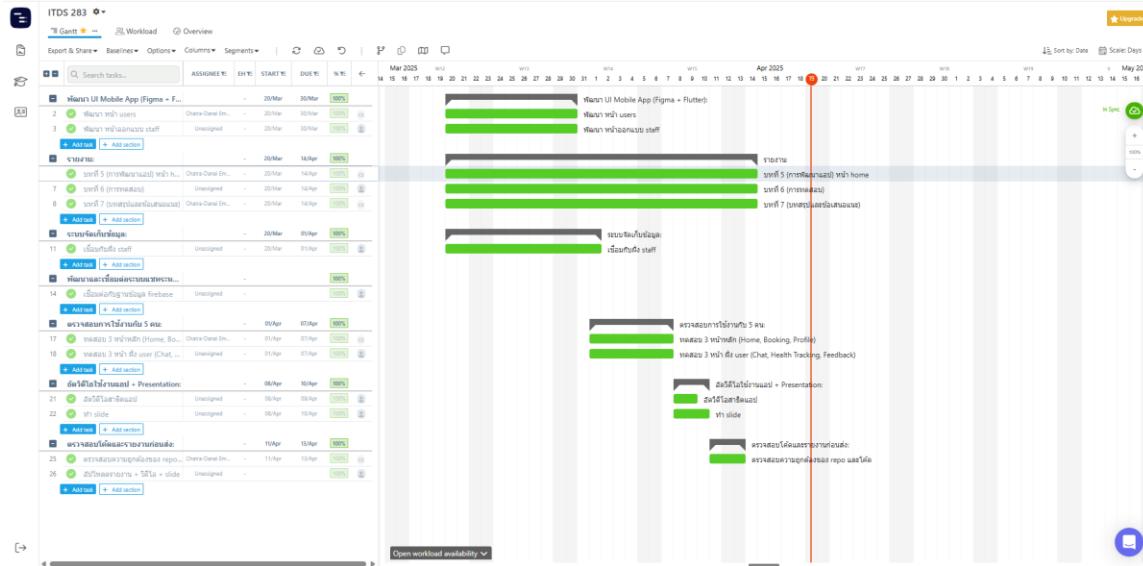
Summary

Projects / ITDS283

Timeline



รูปที่ 4.1 การดำเนินงานในระยะ phase 1



รูปที่ 4.2 การดำเนินการใน phase 2

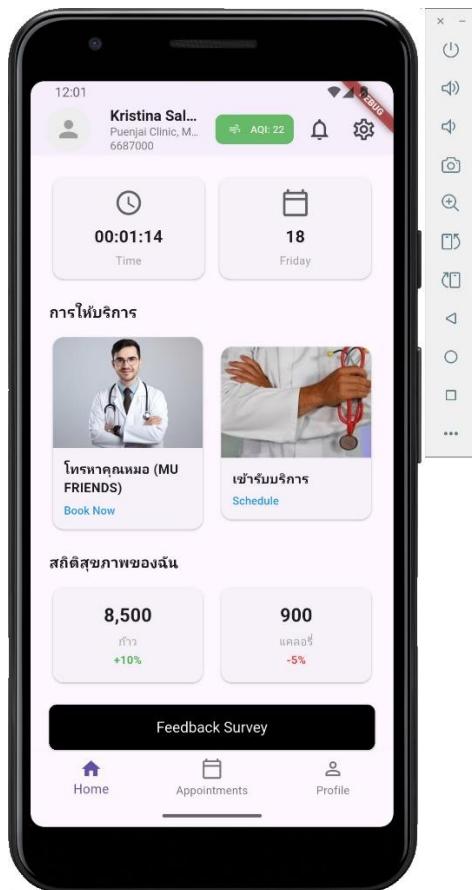
The screenshot shows the GitHub commit history for the repository `ICT-Mahidol-672-ITDS283-Group8-MU-Health-Friend...`. The commits are listed as follows:

- Commits on Apr 17, 2025:**
 - Add TestProject** by pichitchaipae committed 2 days ago. Commit hash: 58a9909.
- Commits on Apr 16, 2025:**
 - emm_user_app** by Chatra-Danai committed 4 days ago. Commit hash: a52b97f.
- End of commit history for this file**

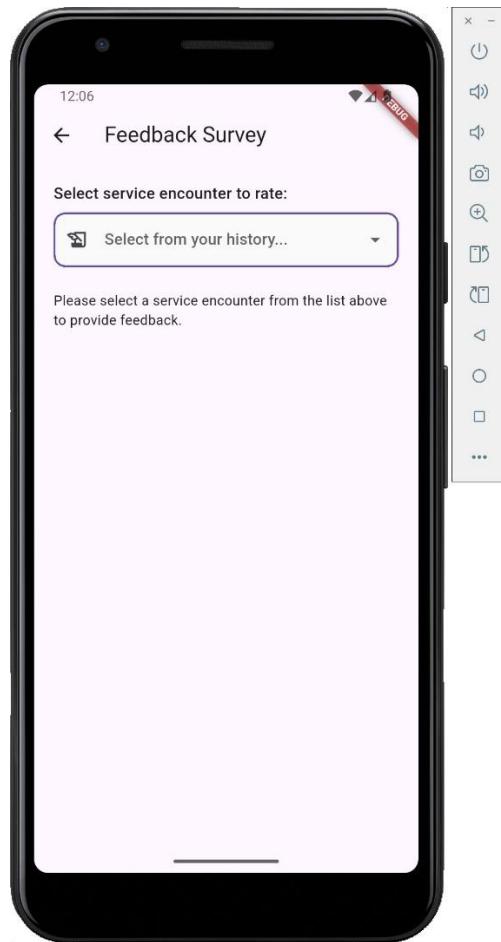
รูปที่ 4.3 การดำเนินการใน phase 2 บน github

บทที่ 5 การพัฒนา Mobile Application

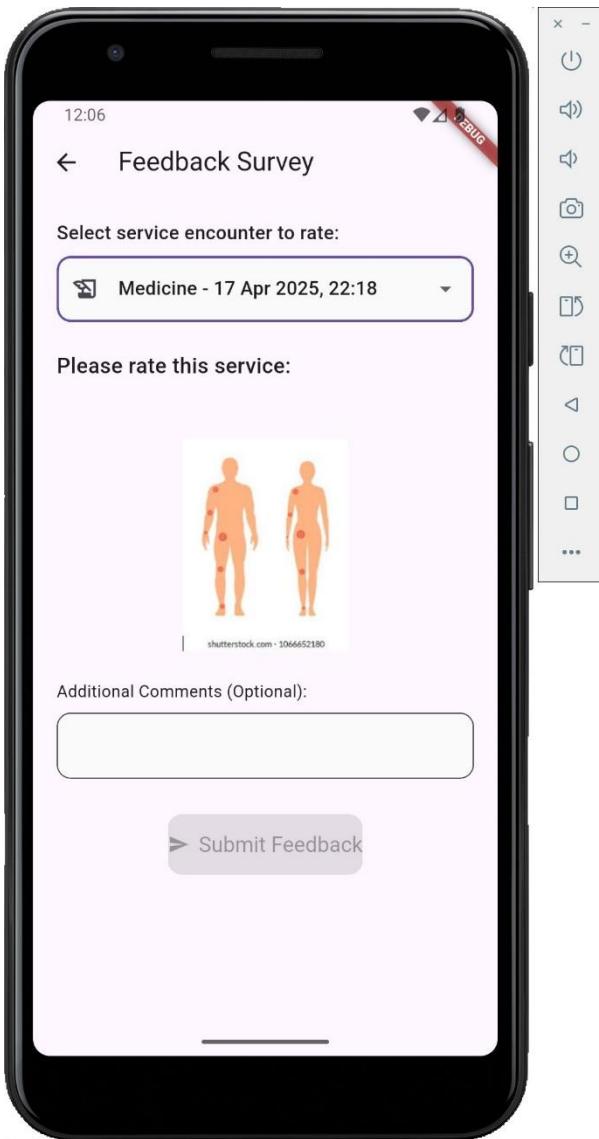
1. User



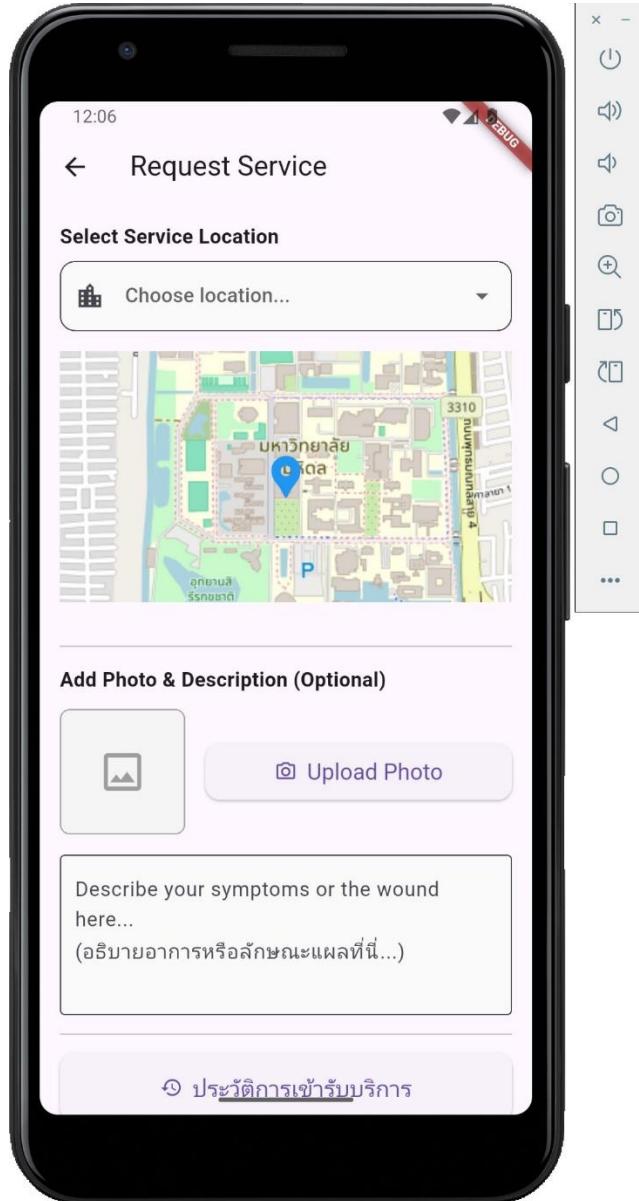
คำอธิบาย: หน้า home เป็นหน้าที่รวมหน้าต่างๆไว้ และเป็นหน้าจอหลัก



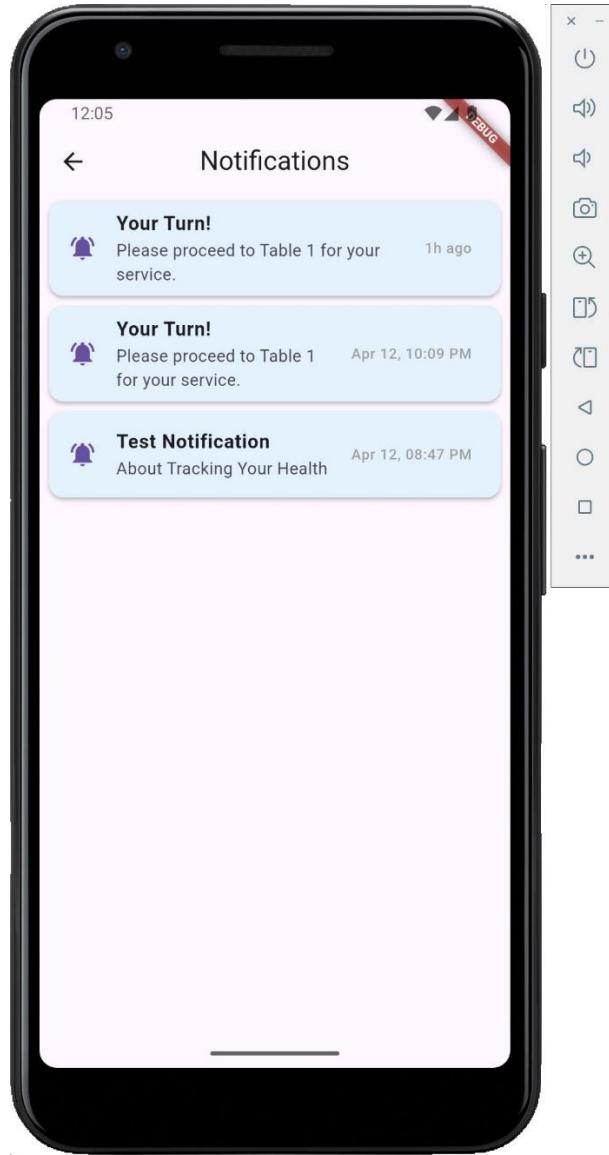
คำอธิบาย: หน้า feedback เมื่อ user พิมพ์ข้อความ ข้อความจะถูกส่งไปที่ database



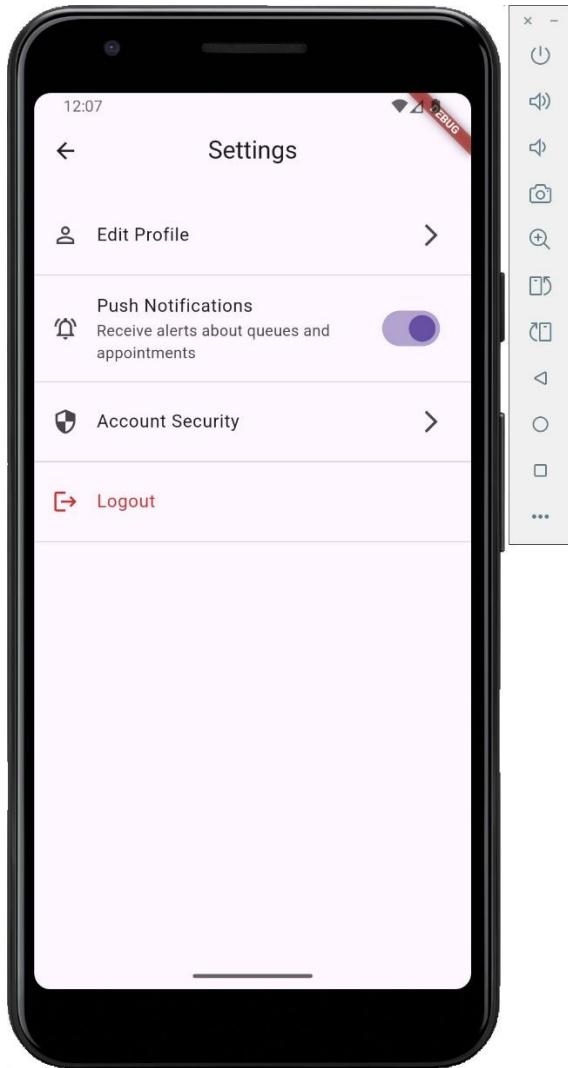
คำอธิบาย: หน้า feedback เมื่อ user พิมพ์ข้อความ ข้อความจะถูกส่งไปที่ database มีการเลือกวันที่เพื่อ rate คะแนน



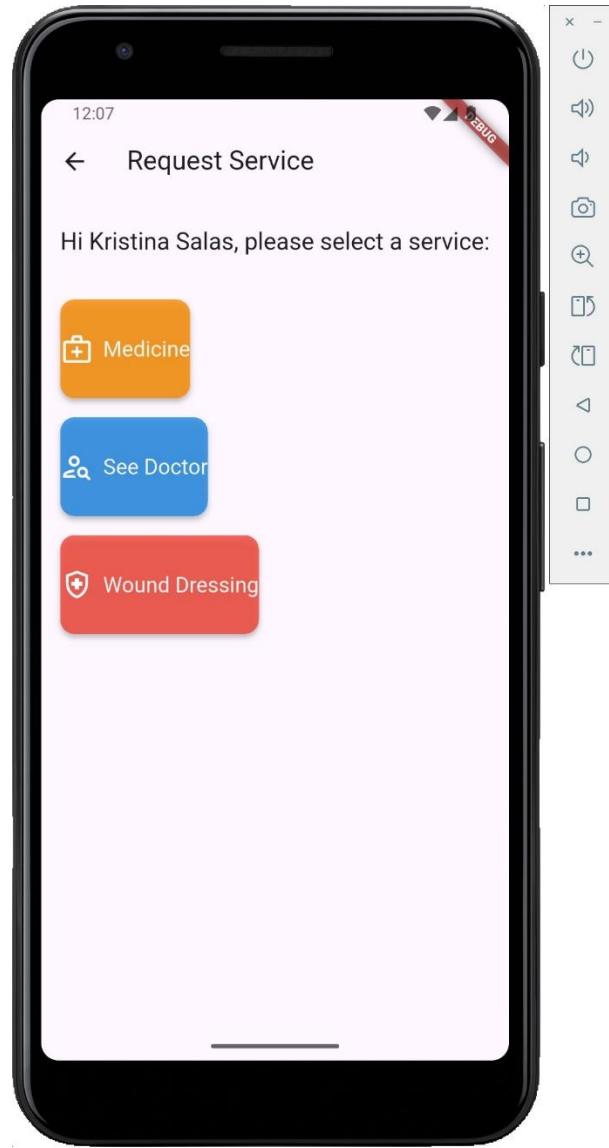
คำอธิบาย: หน้า Request Service สามารถเลือกจุดรับบริการ และจะแสดงผลบนแผนที่ นอกเหนือนี้ยังสามารถอัปโหลดรูปถ่ายได้อีกด้วย เพื่อให้แพทย์ทราบอาการเบื้องต้น



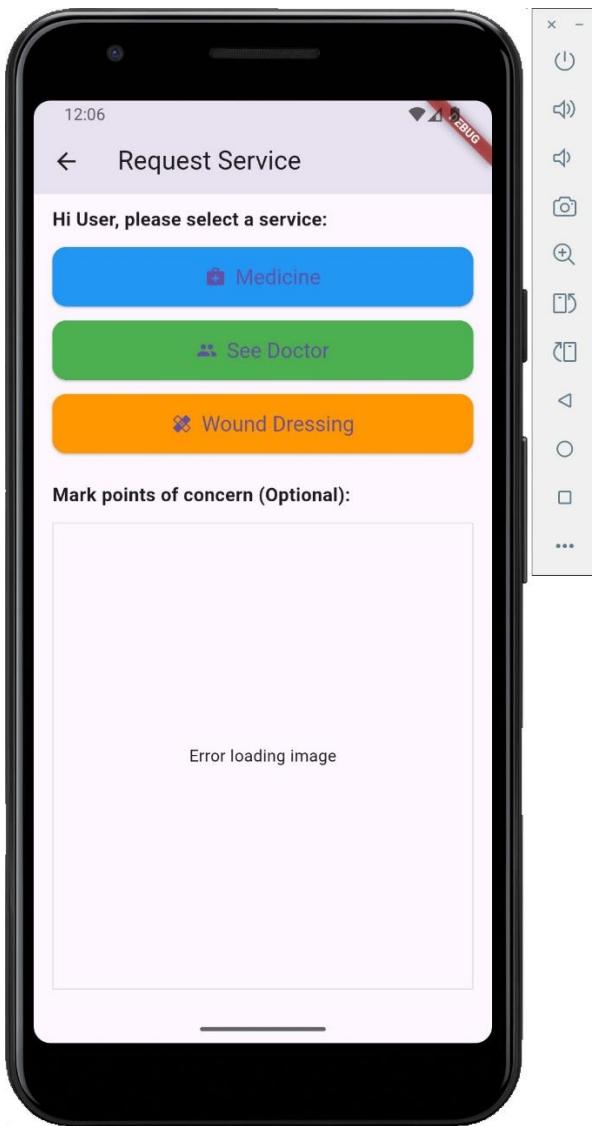
คำอธิบาย: หน้า Notification จะมีการแจ้งเตือนแบบ real-time เมื่อถึงคิวบริการ



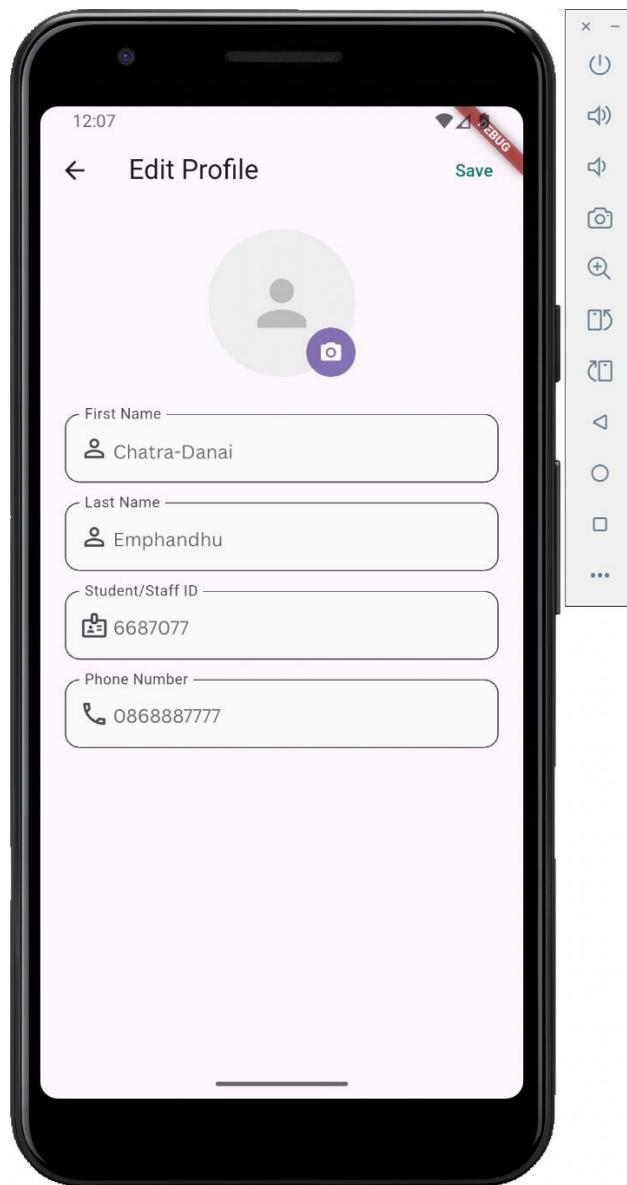
คำอธิบาย: หน้า Setting สามารถแก้ไข profile, เปิดปิดการแจ้งเตือน, ดูระบบ
รักษาความปลอดภัย และออกจากระบบ



คำอธิบาย: หน้า Request Service คือการเลือกรับประเภทบริการ

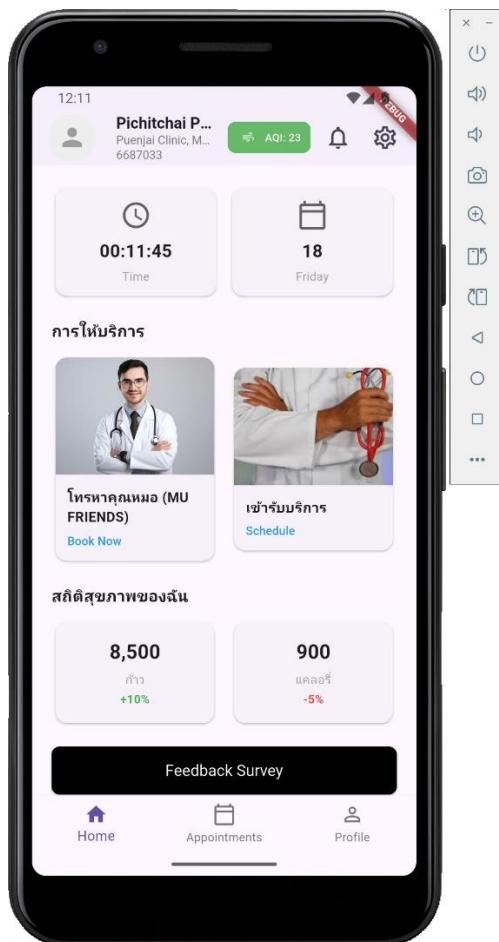


คำอธิบาย: หน้านี้จะสามารถอัปโหลดรูปภาพได้โดยไม่บังคับ

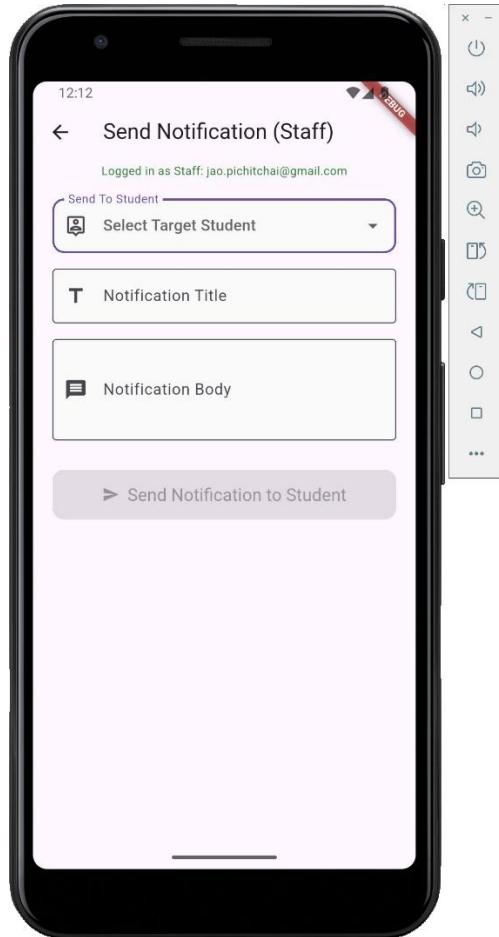


คำอธิบาย: หน้า Edit Profile สามารถแก้ไขข้อมูลส่วนตัวได้

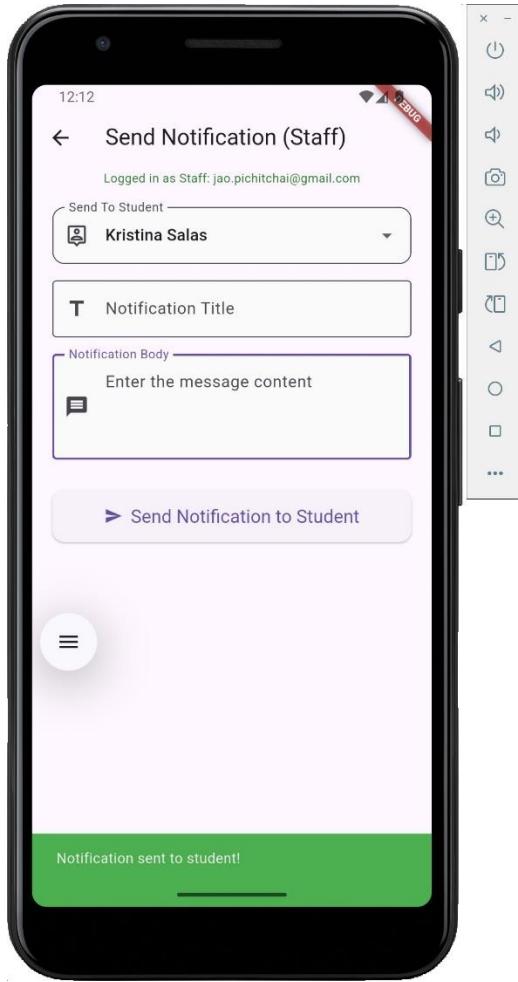
2. Staff



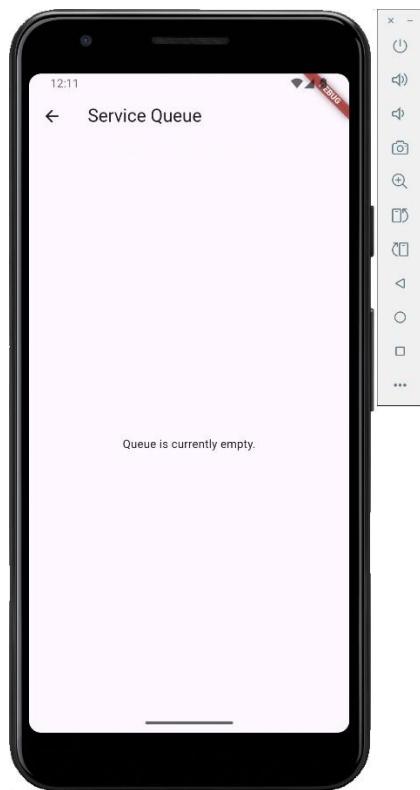
คำอธิบาย: หน้าจอหลักของ staff มีเวลาแบบ realtime และคิวที่จองของ user



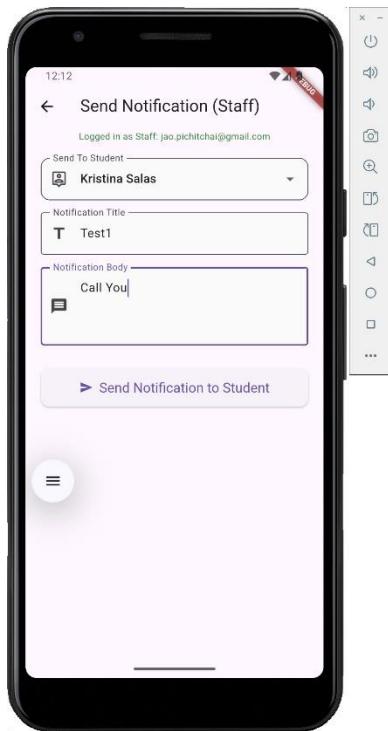
คำอธิบาย: สามารถเลือกส่งการแจ้งเตือนแบบเจาะจง account user ได้



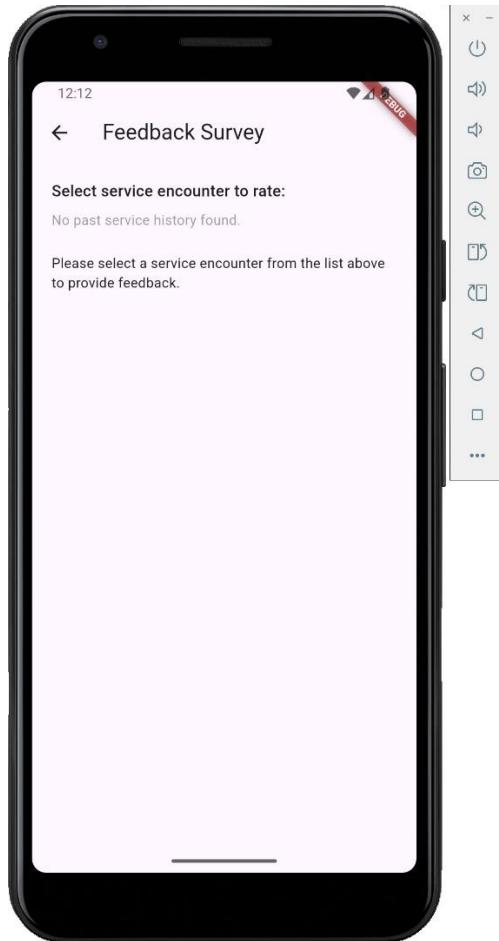
คำอธิบาย: เมื่อส่งการแจ้งเตือนสำเร็จ



คำอธิบาย: หน้าจอแสดงรายการคิวการจอง

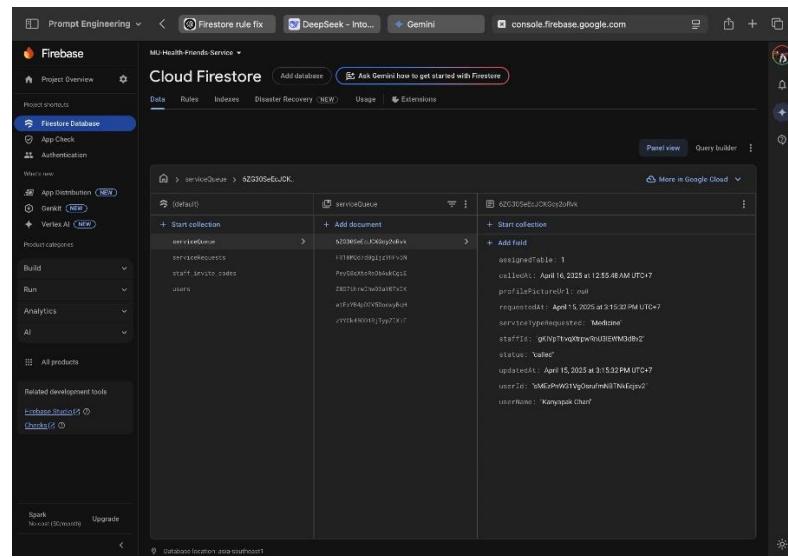


คำอธิบาย: การพิมพ์ข้อความเพื่อส่งการแจ้งเตือนโดย staff



คำอธิบาย: การแสดงประวัติ feedback ที่ได้รับ

3. Backend



The screenshot shows the Firebase Cloud Firestore interface. On the left, the navigation sidebar includes Project Overview, App Check, Authentication, Vertex AI, Product categories, Build, Run, Analytics, AI, and All products. Under Related development tools, it lists Firebase Status and Checks. At the bottom, it shows a database location: Asia/Southeast1.

The main area displays a document in the serviceQueue collection. The document ID is 6ZC30SeSuOKgSy2fRvK. It contains the following fields:

- serviceQueue**: A subcollection containing documents with IDs N203M6elXx0sg2hAvx and f3118Pcc09j1zHnV>N.
- serviceRequests**: A subcollection containing documents with IDs 287121rcw20304t7cX and e1FvYq4D9XWawgEyu4.
- staff_invite_codes**: A subcollection containing documents with IDs vYV1k4H0HfJ3yq*YJ7.
- user**: A subcollection containing documents with IDs 6ZC30SeSuOKgSy2fRvK.

Each document in the serviceQueue collection has the following fields:

- assignedTable**: 1
- calledAt**: April 16, 2025 at 12:55:48 AM UTC+7
- profilePictureUrl**: 1.0sf
- requestedAt**: April 1, 2025 at 8:19:37 PM UTC+7
- serviceTypeRequested**: Medicine
- staffId**: gH9NgTwpkpwRnUEUEM3BzC
- status**: Valid
- updatedAt**: April 15, 2025 at 3:15:32 PM UTC+7
- userId**: 6MOpnW3TygOmfuhhTNkEjn2
- userName**: "KanyaPak Chat"

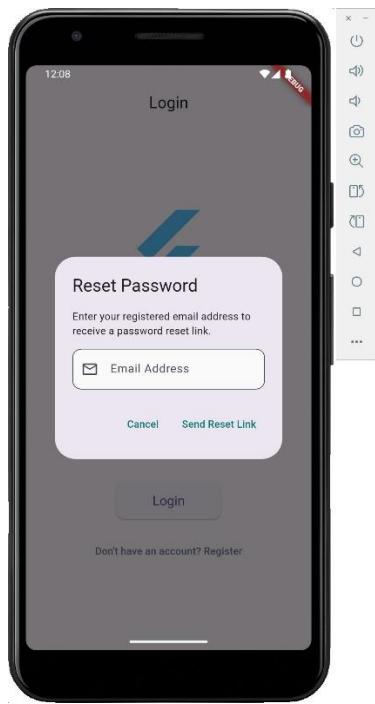
คำอธิบาย: การแสดง การเข้าฐานข้อมูล firebase

```

1  // Match TWO FIRESTORE RULES EDITION:
2  rules_version = '2';
3
4  service cloud.firestore {
5     match /databases/{database}/documents {
6
7         // --- ■ Helper Functions (Firestore Only) ---
8         function isSignedIn() {
9             return request.auth != null;
10        }
11
12         function isUser(userId) {
13             return isSignedIn() && request.auth.uid == userId;
14        }
15
16         function isAdminStaff() {
17             return isUser(userId) && get(/databases/$(database)/documents/users/$(request.auth.uid)).data.isStaff;
18        }
19
20         // --- ■ Staff Invite Order ---
21         match /staff_invite_codes/{codeId} {
22             allow read, update if isSignedIn();
23             allow create, delete if isAdminStaff();
24             allow update, delete if false;
25         }
26
27         // --- ■ User Profiles & Subscriptions ---
28         match /users/{userId} {
29             allow create, if isSignedIn();
30             allow read, if isSignedIn();
31             allow update, if isAdminStaff();
32             allow delete, if isUser(userId);
33         }
34
35         // ■ Notifications Subcollection
36         match /notifications/{notificationId} {
37             allow read, if isSignedIn();
38             allow update, if isSignedIn();
39             allow create, if isAdminStaff();
40             allow delete, if false;
41         }
42
43         // ■ Health History Subcollection
44         match /healthhistory/{historyId} {
45             allow create, if isAdminStaff();
46             allow read, if isSignedIn();
47             allow update, if isSignedIn();
48             allow delete, if false;
49         }
50
51     } // End of /users/{userId} match
52
53     // --- ■ Service Quest (Top Level Collection) ---
54     match /service_quests/{questId} {
55         allow read, if isSignedIn() &&
56             exists(/databases/$(database)/documents/users/$(request.auth.uid));
57         allow create, if isSignedIn() &&
58             (exists(/databases/$(database)/documents/users/$(request.auth.uid)) && !isUser(request.auth.uid));
59     }
60
61 } // End of cloud.firestore

```

คำอธิบาย: การแสดง Code ที่ใช้ในการทำระบบ database



คำอธิบาย: การ reset password โดยการพิมพ์ e-mail เพื่อรับรหัสการแก้ไข

Reset your password for mu-health-friends-service Inbox x 🖨️ 🔗

◀ Summarize this email

 noreply@mu-health-friends-service.firebaseioapp.co... Sat, Apr 12, 8:30 PM (6 days ago) ☆ 😊 ↪ ⋮

to me ▾

Hello,

Follow this link to reset your mu-health-friends-service password for your jao.pichitchai@gmail.com account.

https://mu-health-friends-service.firebaseioapp.com/_/auth/action?mode=resetPassword&oobCode=f1pK202yXwuOk-Biufmr8I3joC0ZxXqSPcTJQxWvwL4AAAGWKjFvA&apiKey=AlzaSyA5AIJDtsnxoNGeOUoyjLPeCKOmFC7G97I&lang=en

If you didn't ask to reset your password, you can ignore this email.

Thanks,

Your mu-health-friends-service team

↪ Reply ↪ Forward 😊

คำอธิบาย: link เพื่อ reset password

The image displays a mobile application registration screen and two screenshots of the Firebase Cloud Firestore console.

Mobile Application Registration Screen:

- The screen shows a title "Register" at the top left.
- A placeholder icon for a profile picture is centered.
- The text "Register as:" is followed by two radio button options: "Student" and "Staff".
- Form fields include: First Name, Last Name, Student ID, Faculty / Unit (with a dropdown menu), Phone Number, Email (used for login), and Password.

Firebase Cloud Firestore Screenshot 1:

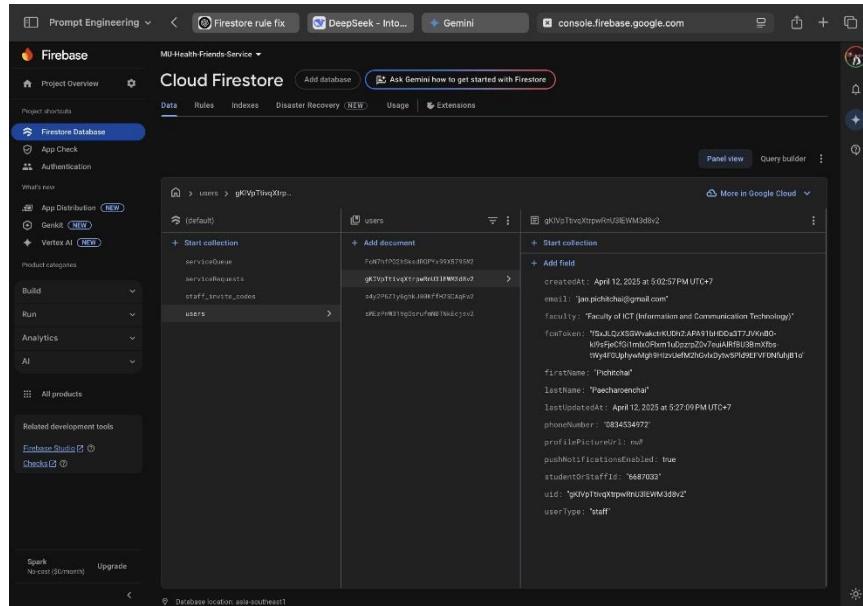
- The interface shows the "Cloud Firestore" section of the Firebase console.
- The "staff_invite_codes" collection is selected.
- A single document named "auto-ID" is shown with the following data:
 - code: "STAFF01"
 - createdAt: April 12, 2025 at 6:00:00 PM UTC+7
 - expiredAt: April 12, 2025 at 7:00:00 PM UTC+7
 - isUsed: true

Firebase Cloud Firestore Screenshot 2:

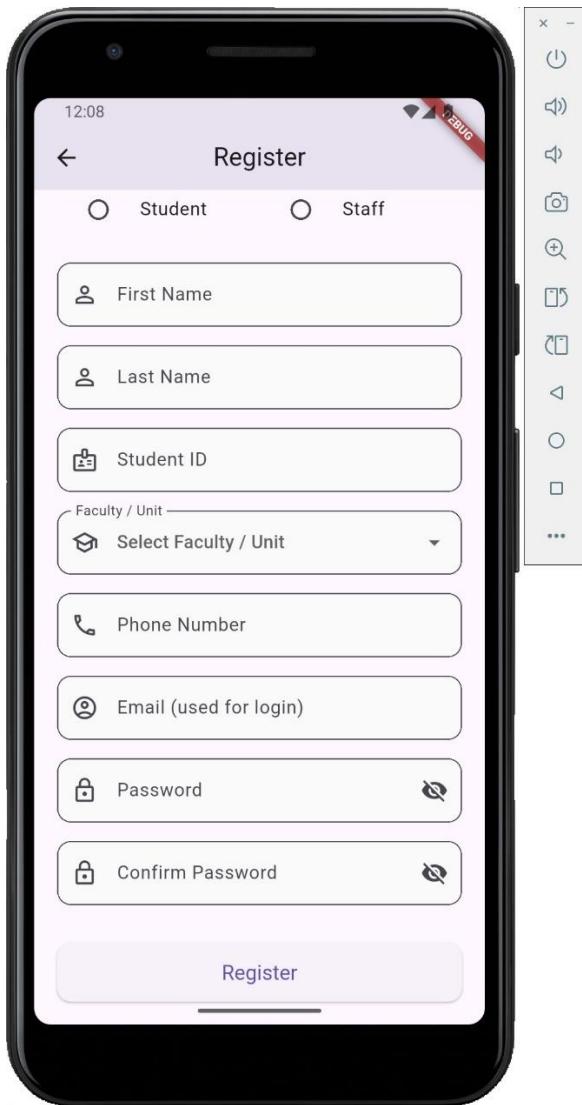
- The interface shows the "Indexes" tab.
- Four indexes are listed:

Collection ID	Fields indexed	Query scope	Status
serviceRequests	userId Ascending timestamp Descending __name__ Descending	Collection	Enabled
staff_invite_codes	code Ascending isUsed Ascending expiresAt Ascending __name__ Ascending	Collection	Enabled
serviceQueue	status Ascending requestedAt Ascending __name__ Ascending	Collection	Enabled
users	userType Ascending firstName Ascending __name__ Ascending	Collection	Enabled

คำอธิบาย: เมื่อกรอกข้อมูล data จะถูกส่ง
เข้าไป firebase

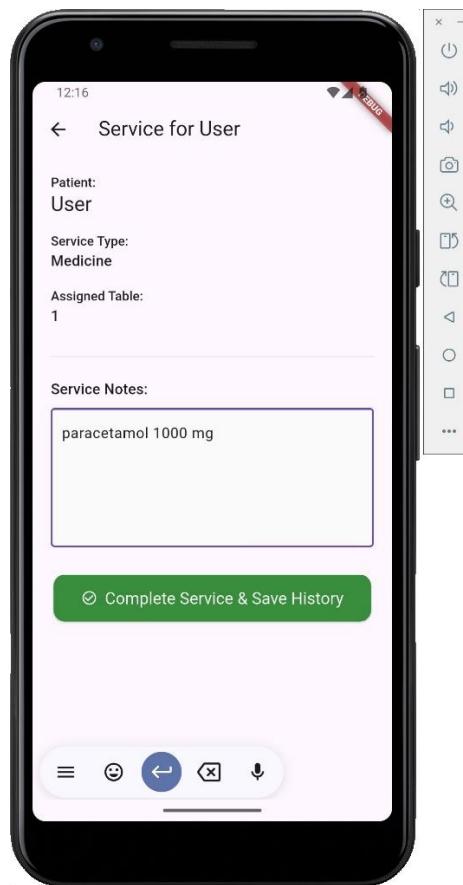


คำอธิบาย: ข้อมูลที่ได้รับจาก user บน firebase

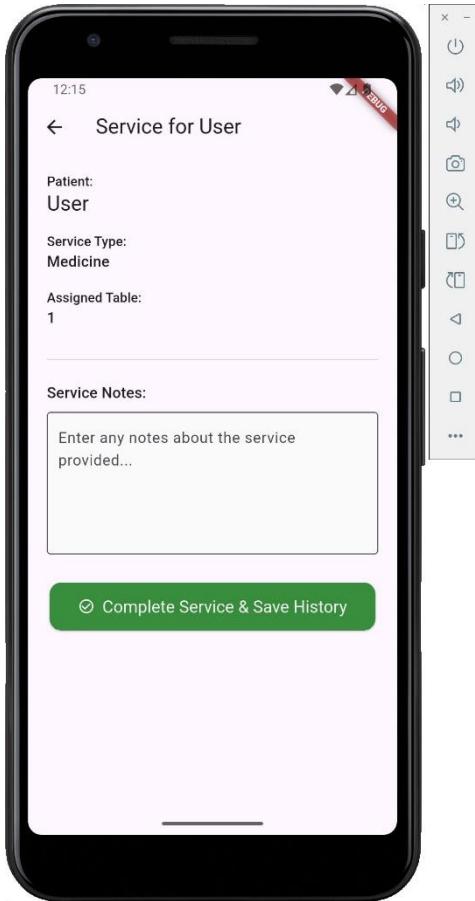


คำอธิบาย: การ register จะมีให้เลือกสถานะว่าเป็น student หรือ staff

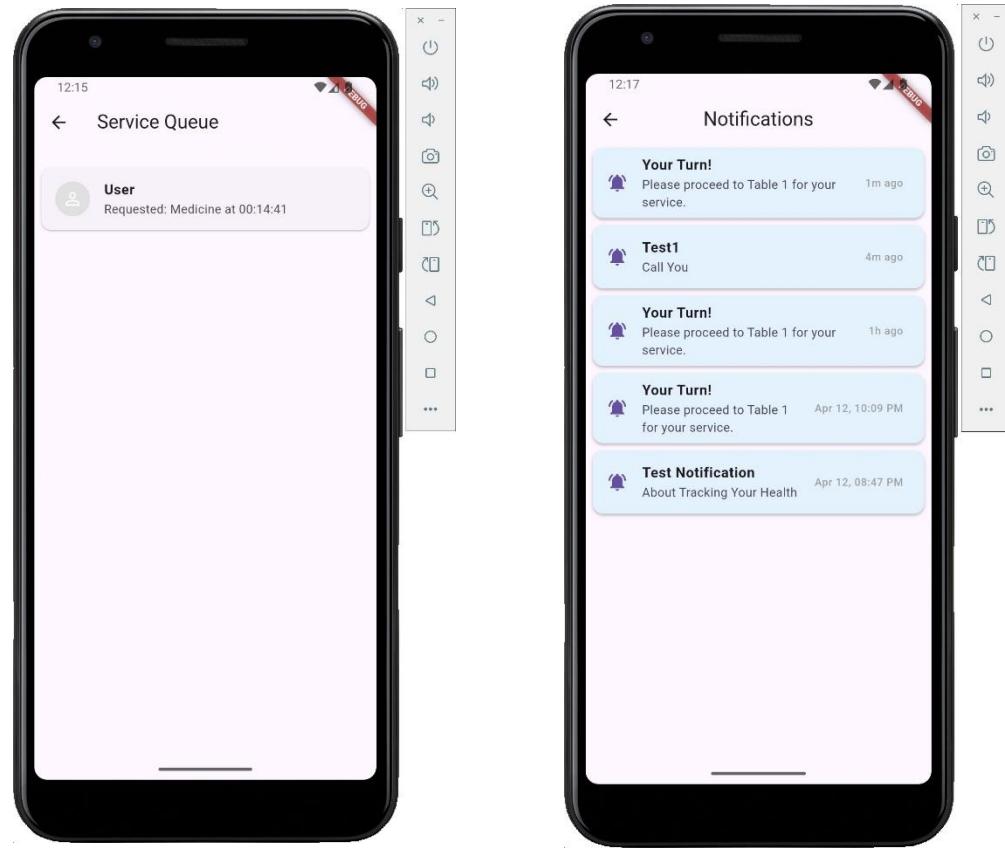
4. flow(relational) ของ user(student) and staff



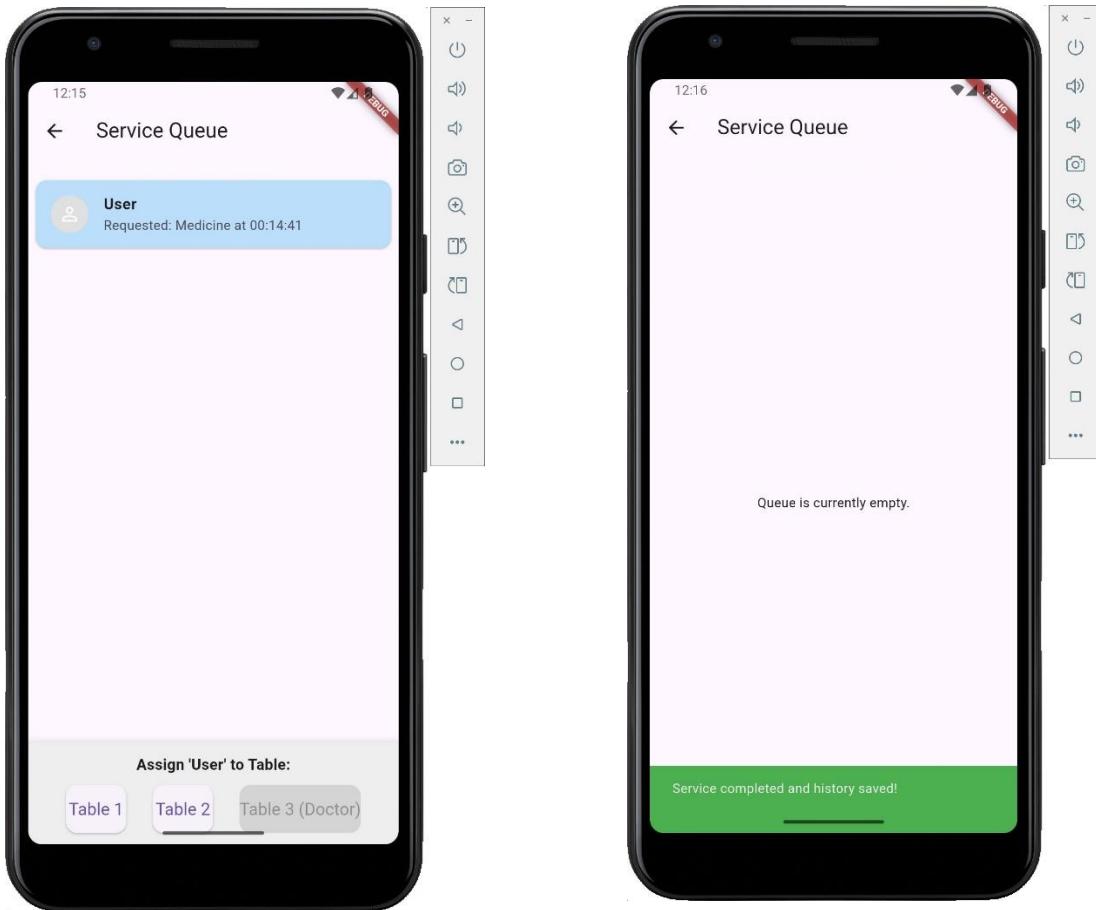
คำอธิบาย: การเพิ่ม data สำหรับ user ที่มีผลต่อ service เช่น การให้ยา



คำอธิบาย: เมื่อพิมพ์ข้อความแล้ว ข้อมูลจะถูกส่งไปที่ระบบ
ฐานข้อมูล



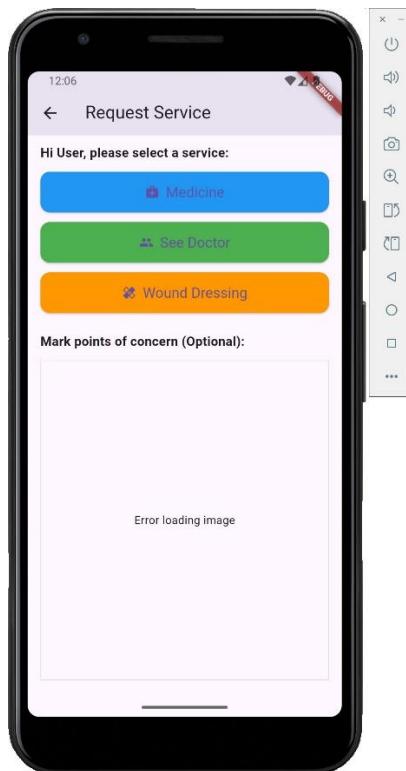
คำอธิบาย: แสดงลำดับการรับบริการ และจะส่งการแจ้งเตือนให้ user



คำอธิบาย: Staff สามารถ assign ให้ผู้ใช้บริการไปรับบริการที่ต้อง^{จะ} และจะแสดงลำดับการรับบริการ

5. ยกตัวอย่าง Source Code ที่น่าสนใจที่นักศึกษาได้พัฒนา อย่างน้อย 3 ตัวอย่าง โดยพยายามอธิบาย และมีตัวอย่างของหน้าจอประกอบการบรรยาย

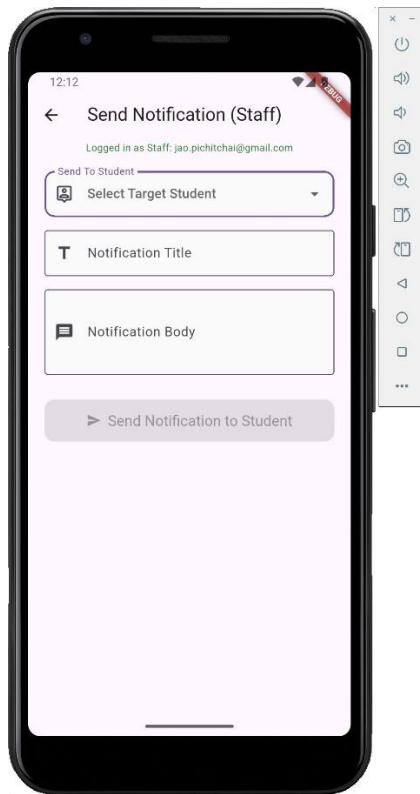
5.1 หน้า Request Service



```
80 def binary_search(arr, target):
81
82     low = 0
83     high = len(arr) - 1
84
85     while low <= high:
86         mid = (low + high) // 2
87         if arr[mid] == target:
88             return mid
89         elif arr[mid] < target:
90             low = mid + 1
91         else:
92             high = mid - 1
93
94     return -1
95
96
97 sorted_list = [2, 5, 7, 8, 11, 12]
98 search_value = 11
99 result = binary_search(sorted_list, search_value)
100
101 if result != -1:
102     print(f"Found {search_value} at {result}")
103 else:
104     print(f"{search_value} Not found")
```

การอธิบาย: ฟังก์ชัน `binary_search` เป็นอัลกอริธึมที่มีประสิทธิภาพในการค้นหาค่า `target` ในรายการ `arr` ที่เรียงลำดับแล้ว โดยการแบ่งครึ่งรายการและเปรียบเทียบค่าตรงกลางซ้ำๆ จนกว่าจะพบ `target` หรือไม่พบ (คืนค่า `-1`)

5.2 Send Notification (Staff)



```

import 'package:flutter/material.dart';
class SendNotificationScreen extends StatefulWidget {
  @override
  _SendNotificationScreenState createState() => _SendNotificationScreenState();
}

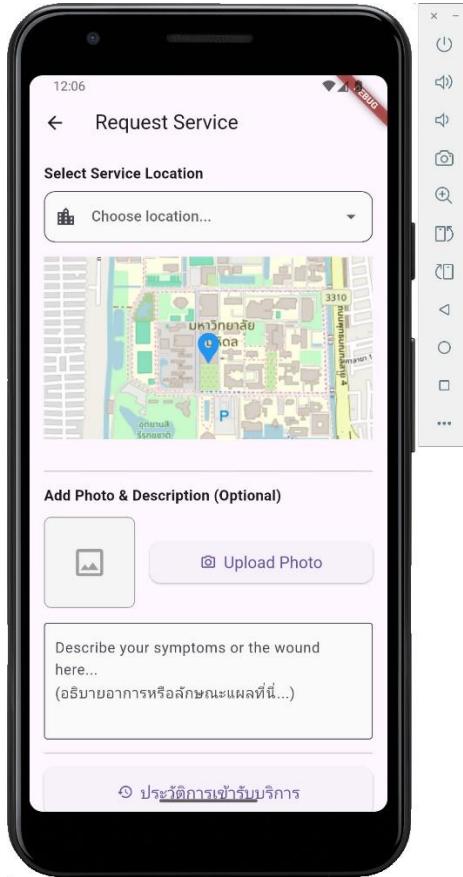
class _SendNotificationScreenState extends State<SendNotificationScreen> {
  String? _selectedStudent;
  TextEditingController _titleController = TextEditingController();
  TextEditingController _bodyController = TextEditingController();
  List<String> _studentList = ['Student A', 'Student B', 'Student C']; // Replace with actual student data

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Send Notification (Staff)'),
        leading: IconButton(
          icon: Icon(Icons.arrow_back),
          onPressed: () {
            Navigator.pop(context);
          },
        ),
      ),
      body: SingleChildScrollView(
        padding: EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: <Widget>[
            Text(
              'Logged in as Staff: jao.pichitchai@gmail.com',
              style: TextStyle(fontSize: 12.0, color: Colors.grey),
            ),
            SizedBox(height: 20.0),
            Text('Send To Student', style: TextStyle(fontWeight: FontWeight.bold)),
            SizedBox(height: 8.0),
            DropdownButtonFormField<String>(
              decoration: InputDecoration(
                border: OutlineInputBorder(),
                prefixIcon: Icon(Icons.person),
              ),
            ),
          ],
        ),
      ),
    );
  }
}

```

การอธิบาย: การสร้างหน้าจอ "Send Notification (Staff)" ที่ช่วยให้เจ้าหน้าที่สามารถส่งข้อความแจ้งเตือนไปยังนักเรียนที่เลือกได้ โดยประกอบด้วยส่วนหัว, ข้อมูลผู้ใช้ที่ล็อกอิน, ตัวเลือก Dropdown สำหรับเลือกนักเรียนเป้าหมาย, ช่อง TextField สำหรับป้อนหัวข้อและเนื้อหาการแจ้งเตือน, และปุ่ม "Send Notification to Student" สำหรับดำเนินการส่งข้อความ ซึ่งมีการจัดการสถานะของข้อมูลที่ป้อนและการแสดงข้อความแจ้งเตือนเมื่อมีการดำเนินการส่งหรือเมื่อข้อมูลไม่ครบถ้วน

5.3 Request Service (User)



```
196 import 'package:flutter/material.dart';
197
198 class RequestServiceScreen extends StatefulWidget {
199   @override
200   _RequestServiceScreenState createState() => _RequestServiceScreenState();
201 }
202
203 class _RequestServiceScreenState extends State<RequestServiceScreen> {
204   String? _selectedLocation;
205   TextEditingController _descriptionController = TextEditingController();
206
207
208
209   List<String> _locations = ['MU Health'];
210
211   Map<String, String> _locationDetails = {
212     'MU Health': 'Mahidol Learning Center',
213   };
214 }
```

การอธิบาย: ช่วยให้ผู้ใช้สามารถเลือกสถานที่รับบริการจาก Dropdown, เพิ่มรูปภาพและคำอธิบายเพิ่มเติมเกี่ยวกับอาการหรือบาดแผล (optional) ผ่านปุ่มสำหรับถ่ายรูปหรืออัปโหลด, และมีปุ่ม "ประวัติการเข้ารับบริการ" สำหรับไปยังหน้าดูประวัติ โดยมีการจัดการสถานะของสถานที่ที่เลือกและข้อความที่ป้อน และแสดงรายละเอียดเพิ่มเติมของสถานที่ที่เลือก

บทที่ 6

การทดสอบ Mobile Application

จากผลการทดสอบชุดทดสอบสำหรับแอปพลิเคชันมือถือ MU Health Friends Service จำนวน 10 ท่าน โดยรวมแล้วให้ผลลัพธ์ในทางบวก โดยเฉพาะอย่างยิ่งในส่วนหลักของการยืนยันตัวตนและการดำเนินงานของเจ้าหน้าที่ เมื่อเปิดแอปพลิเคชัน หน้าจอเข้าสู่ระบบจะปรากฏขึ้นอย่างสม่ำเสมอ ซึ่งเป็นจุดเริ่มต้นที่จำเป็นสำหรับผู้ใช้ในการเข้าใช้งาน. ระบบการยืนยันตัวตนผู้ใช้มีความน่าเชื่อถือ โดยทั้งนักเรียนและเจ้าหน้าที่สามารถเข้าสู่ระบบได้สำเร็จโดยใช้ข้อมูลรับรองที่ถูกต้อง และระบบจะนำทางไปยังแดชบอร์ดที่เกี่ยวข้องของแต่ละกลุ่ม

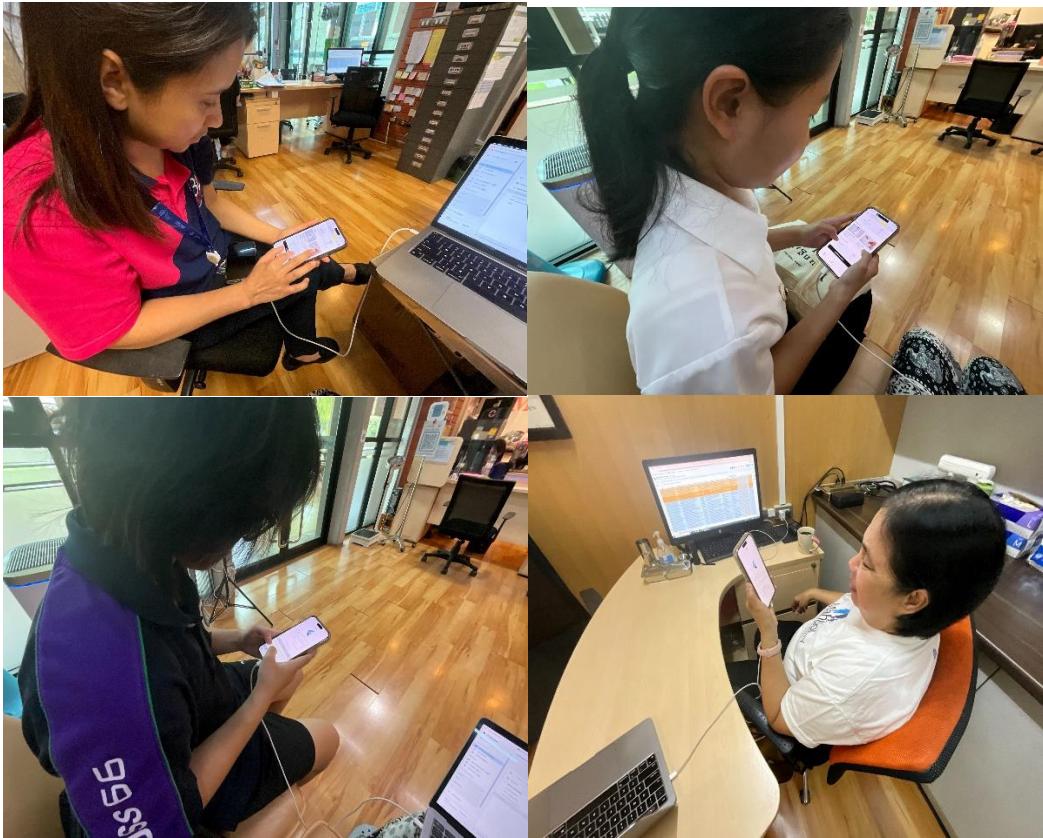
นอกจากนี้ แอปพลิเคชันยังจัดการความพยายามในการเข้าสู่ระบบที่ไม่ถูกต้องได้อย่างมีประสิทธิภาพเมื่อผู้ใช้ป้อนรหัสผ่านที่ไม่ถูกต้อง ระบบจะแสดงข้อความผิดพลาดที่เหมาะสม. ในทำนองเดียวกัน ความพยายามในการเข้าสู่ระบบด้วยที่อยู่อีเมลที่ไม่ได้ลงทะเบียนในระบบ หรือการเว้นว่างช่องอีเมลหรือรหัสผ่าน จะส่งผลให้ระบบแสดงข้อความแจ้งเตือนการตรวจสอบที่ให้ข้อมูลอย่างชัดเจน พงก์ชันการออกจากระบบก็ทำงานตามที่ออกแบบไว้ เช่นกัน โดยนำผู้ใช้กลับไปยังหน้าจอเข้าสู่ระบบได้สำเร็จ สำหรับผู้ใช้ที่เป็นเจ้าหน้าที่ คุณสมบัติการจัดการคิวของแอปพลิเคชันทำงานได้อย่างถูกต้อง หน้าจอคิวของเจ้าหน้าที่แสดงรายชื่อนักเรียนที่รอรับบริการอย่างแม่นยำ

ยิ่งไปกว่านั้น ส่วนติดต่อผู้ใช้ยังช่วยให้สามารถโต้ตอบกับคิวได้อย่างเหมาะสม: การเลือกนักเรียนจะเน้นรายการของนักเรียน และปุ่ม "Assign Table" จะถูกเปิดหรือปิดใช้งานอย่างถูกต้องตามความต้องการของนักเรียนที่เลือก กระบวนการกำหนดนักเรียนไปยังพื้นที่บริการ การอัปเดตข้อมูลคิว การสร้างการแจ้งเตือน และการนำทางไปยังหน้าจอ Active Service ดำเนินไปโดยไม่มีข้อผิดพลาด

สุดท้ายนี้ การดำเนินการบริการให้เสร็จสิ้น การอัปเดตประวัติบริการ และการกลับไปยังหน้าจอคิวภายหลัง ล้วนประสบความสำเร็จ อย่างไรก็ตาม ในส่วนเจ้าหน้าที่ของคณะ ICT ในฐานะผู้ใช้บริการจะต้องลงทะเบียนเป็น student เพราะไม่สามารถเข้าถึงข้อมูลต่างๆ ของ staff ซึ่งเป็นบุคลิกการทางการแพทย์ได้ ตามนโยบาย PDPA

6.1 หลักฐานการทดสอบ Mobile Application:







บทที่ 7

สรุป

7.1 สรุปผลที่ได้รับจากการพัฒนา Mobile Application ความสามารถบรรลุวัตถุประสงค์หรือไม่

จากการทำโครงการ พัฒนา Mobile Application สามารถสรุปผลได้ว่า ได้ดังนี้:

การยืนยันตัวตน (Authentication): การทดสอบส่วนใหญ่ในส่วนนี้ถือว่าประสบความสำเร็จ. แอปพลิเคชันสามารถแสดงหน้าจอเข้าสู่ระบบได้ถูกต้อง, ผู้ใช้ที่เป็นนักเรียนและเจ้าหน้าที่สามารถเข้าสู่ระบบได้เมื่อกรอกข้อมูลถูกต้อง, และระบบมีการจัดการข้อผิดพลาดเมื่อผู้ใช้กรอกรหัสผ่านผิด, อีเมลที่ไม่ได้ลงทะเบียน, หรือเว้นว่างช่องอีเมล/รหัสผ่าน การออกจากระบบ (Logout): พังก์ชันการออกจากระบบทำงานได้ตามที่คาดไว้

ขั้นตอนการทำงานของเจ้าหน้าที่ (Staff Flow): การทดสอบในส่วนนี้ก็ให้ผลลัพธ์ที่ดี. เจ้าหน้าที่สามารถเข้าสู่ระบบและเห็นหน้าจอคิวได้, การแสดงคิวและการต่อตัวกับคิวทำงานถูกต้อง, และการกำหนดบริการ/การจัดการบริการดำเนินการได้ตามที่ออกแบบไว้.

อย่างไรก็ตาม มีข้อยกเว้นคือการลืมรหัสผ่าน (Forgot Password): แม้ว่าระบบจะแสดงข้อความว่าได้ส่งลิงก์รีเซ็ตรหัสผ่านไปแล้ว, แต่ยังไม่มีการยืนยันว่าอีเมลถูกส่งถึงผู้ใช้จริงและลิงก์ใช้งานได้หรือไม่ ส่วนนี้ยังคงต้องมีการตรวจสอบเพิ่มเติม

โดยสรุป Mobile Application บริการ MU Health Friends โดยรวมสามารถบรรลุวัตถุประสงค์ในการพัฒนาในด้านการยืนยันตัวตน, การออกจากระบบ, และขั้นตอนการทำงานของเจ้าหน้าที่ได้เป็นอย่างดี. แต่ในส่วนของฟังก์ชัน "ลืมรหัสผ่าน" ยังคงต้องมีการตรวจสอบเพิ่มเติมเพื่อยืนยันว่าสามารถทำงานได้อย่างสมบูรณ์ ในส่วนเจ้าหน้าที่ของมหาวิทยาลัยในฐานะผู้ใช้บริการจะต้องลงทะเบียนเป็น student เพราะไม่สามารถเข้าถึงข้อมูลต่างๆ ของ staff ซึ่งเป็นบุคลากรทางการแพทย์ได้ ตามนโยบาย PDPA

7.2 สิ่งที่สำคัญๆ ได้เรียนรู้จากการทำโครงการ และสิ่งที่อยากรู้เพิ่มเติมใน Mobile Application นี้

จากการทำโครงการ ได้เรียนรู้ถึงความสำคัญของการทดสอบอย่างละเอียดถี่ถ้วน เพื่อให้สามารถบูรณาหารที่อาจเกิดขึ้นได้ สามารถเห็นว่าการทดสอบระบบการยืนยันตัวตนผู้ใช้ (Authentication) เป็นเรื่องที่สำคัญมาก และได้รับประสบการณ์ในการสร้างและตรวจสอบความถูกต้องของกระบวนการเข้าสู่ระบบ, ออกจากระบบ, และการกู้คืนรหัสผ่าน นอกจากนี้ การทดสอบขั้นตอนการทำงานของเจ้าหน้าที่ (Staff Flow) ยังแสดงให้เห็นถึงความซับซ้อนในการจัดการคิว, การมอบหมายบริการ, และการจัดการการแจ้งเตือน ซึ่งทำให้พวากษาได้เรียนรู้วิธีการจำลองและนำกระบวนการทำงานจริงเหล่านี้มาสร้างเป็นแอปพลิเคชัน การทดสอบการจัดการข้อผิดพลาด (Error Handling) ยังทำให้เห็นถึงความสำคัญของการแสดงข้อความผิดพลาดที่ชัดเจนและเป็นประโยชน์ เพื่อแนะนำผู้ใช้ในการแก้ไขปัญหา และสุดท้ายนี้ ข้อสังเกตเกี่ยวกับฟังก์ชัน "ลืมรหัสผ่าน" แสดงให้เห็นว่าการพัฒนาเป็นกระบวนการที่ต้องทำขั้นตอนๆ และปรับปรุง ซึ่งได้เรียนรู้ถึงความสำคัญของการแก้ไขจุดที่ต้องปรับปรุงจากการทดสอบ

จากการพัฒนาแอปพลิเคชัน ต้องการที่จะพัฒนาฟังก์ชัน "ลืมรหัสผ่าน" ให้สมบูรณ์ และตรวจสอบได้ทั้งหมด ซึ่งรวมถึงการตรวจสอบว่าอีเมลถูกส่งถึงผู้ใช้จริงและลิงก์ที่ส่งไปใช้งานได้จริง นอกจากนี้ ต้องการปรับปรุงส่วนติดต่อผู้ใช้ (User Interface) และประสบการณ์ผู้ใช้ (User Experience) เพื่อให้แอปพลิเคชันใช้งานง่ายและน่าดึงดูดยิ่งขึ้น ทั้งนี้ขึ้นอยู่กับขอบเขตของแอปพลิเคชัน เพิ่มคุณสมบัติเพิ่มเติม เช่น การนัดหมายบริการ, ประวัติสุขภาพผู้ใช้, การเชื่อมต่อกับระบบอื่น ๆ, หรือระบบการแจ้งเตือนที่ละเอียดกว่าเดิม การปรับปรุงประสิทธิภาพของแอปพลิเคชันในด้านความเร็ว, การตอบสนอง, และการใช้พลังงานแบตเตอรี่ ก็อาจเป็นสิ่งที่พวากษาให้ความสำคัญ รวมถึงในการลงทุนเพิ่ม staff ของ ICT และสุดท้ายนี้ เพื่อความปลอดภัยของข้อมูลผู้ใช้ ต้องการเพิ่มมาตรการรักษาความปลอดภัยที่นอกเหนือไปจากการยืนยันตัวตนขั้นพื้นฐาน

สามารถสรุปผลการดำเนินงาน ปัญหา และ Feedback

7.2.1 ฟังก์ชันหลักที่พัฒนา

7.2.1.1 Login/Register ('authen.dart', 'register.dart'):

* Login: ใช้ `await FirebaseAuth.instance.signInWithEmailAndPassword(email: email, password: password);`

7.2.1.2 Register: ใช้ `await

FirebaseAuth.instance.createUserWithEmailAndPassword(email: email, password: password);`

7.2.1.3 Forgot Password: ใช้ `await

FirebaseAuth.instance.sendPasswordResetEmail(email: email);`

7.2.1.4 User Profile (Firestore Save)

บันทึกข้อมูลตอน Register ด้วย

```dart

```
await FirebaseFirestore.instance.collection('users').doc(user.uid).set({
 'uid': user.uid,
 'firstName': _firstNameController.text.trim(),
 'lastName': _lastNameController.text.trim(),
 // ... other fields ...
 'userType': _selectedUserType.toString().split('.').last,
 'createdAt': FieldValue.serverTimestamp(),
});
```

### 7.1.1.5 Staff Invite Code Check (ใน `register.dart`):\*\* ตรวจสอบโค้ดก่อนสร้าง User:

```
```dart
final querySnapshot = await FirebaseFirestore.instance
    .collection('staff_invite_codes')
    .where('code', isEqualTo: enteredCode)
    .where('expiresAt', isGreaterThanOrEqualTo: Timestamp.now())
    // .where('isUsed', isEqualTo: false) // Optional
    .limit(1)
    .get();
if (querySnapshot.docs.isEmpty) {
  throw Exception("Invalid or expired Staff Invite Code.");
}
// ... proceed with registration ...

// Optional: Mark code as used later using querySnapshot.docs.first.id
```

```

### 7.1.1.6 User Dashboard (`user\_dashboard\_screen.dart`):\*\*

\* Fetch Data: ดึงข้อมูลด้วย `await`

FirebaseFirestore.instance.collection('users').doc(uid).get();` และใช้ `setState` อัปเดต `'\_userName'`, `'\_profilePicUrl'`, `'\_userType'` ฯลฯ

\* API: ใช้ `http.get(Uri.parse(apiUrl))`

\* Role-Based Nav: ใน `BottomNavigationBar.onTap`:

```
```dart
if (index == 1) { // Appointments Tab
  if (_userType == 'student') {
    Navigator.push(context, MaterialPageRoute(builder: (_) => const
      BookingPage()));
  }
}
```

```

```

} else if (_userType == 'staff') {
 Navigator.push(context, MaterialPageRoute(builder: (_) => const
 StaffQueueScreen()));
}
```
```

```

#### 7.1.1.7 Appointment Booking ('booking\_page.dart'):\*\*

\* Add to Queue: សំចាប់ផ្តើមបណ្តុះបណ្តាល:

```

```dart
await FirebaseFirestore.instance.collection('serviceQueue').add({
  'userId': _currentUser!.uid,
  'userName': _userName, // Fetched full name
  'profilePictureUrl': _profilePicUrl,
  'requestedAt': FieldValue.serverTimestamp(),
  'status': 'waiting',
  'serviceTypeRequested': serviceType,
  'requestedLocation': _selectedServiceLocation,
  'painPoints': pointsToSave.isEmpty ? null : pointsToSave,
  'woundImageUrl': woundImageUrl,
  'description': description.isEmpty ? null : description,
});
```
```

```

7.1.1.8 Staff Queue ('staff_queue_screen.dart'):**

* Read Queue: ใช้ `StreamBuilder` กับ:

```
```dart
stream: FirebaseFirestore.instance
 .collection('serviceQueue')
 .where('status', isEqualTo: 'waiting')
 .orderBy('requestedAt', descending: false)
 .snapshots(),
````
```

* Assign Table: อัปเดตข้อมูลด้วย:

```
```dart
await
FirebaseFirestore.instance.collection('serviceQueue').doc(queueDocId).update({
 'status': 'called',
 'calledAt': FieldValue.serverTimestamp(),
 'staffId': staffId,
 'assignedTable': tableNumber,
});
````
```

7.1.1.9 Active Service ('active_service_screen.dart'):**

* Save History: บันทึกประวัติและอัปเดตคิวด้วย 'WriteBatch':

```dart

```
WriteBatch batch = firestore.batch();
DocumentReference historyRef =
firestore.collection('users').doc(widget.userId).collection('healthHistory').doc();
batch.set(historyRef, historyData); // historyData contains notes, type, staff etc.
DocumentReference queueRef =
firestore.collection('serviceQueue').doc(widget.queueDocId);
batch.update(queueRef, {'status': 'completed', 'completedAt':
FieldValue.serverTimestamp()});
await batch.commit();
````
```

7.1.1.10 Notifications ('notification.dart'):**

* User Viewing List: ใช้ `StreamBuilder` กับ:

```
```dart
stream: FirebaseFirestore.instance
 .collection('users')
 .doc(_currentUser!.uid)
 .collection('notifications')
 .orderBy('timestamp', descending: true)
 .snapshots(),
```

```

* Staff Sending (Trigger): เขียน Notification ตอน Assign Table:

```
```dart
// (Inside _assignToTable or utility function)
await _sendNotificationToUser(
 targetUserId: targetUserId,
 title: "Your Turn!",
 body: "Please proceed to Table $tableNumber..."
)
// This function internally does:
//
FirebaseFirestore.instance.collection('users').doc(targetUserId).collection('notifications').add({...});
```

```

* **Settings/Edit Profile ('settings_screen.dart', 'edit_profile_screen.dart'):**

* Notification Toggle: อัปเดต Firestore ด้วย:

```
```dart
await FirebaseFirestore.instance
 .collection('users')
 .doc(_currentUser!.uid)
 .update({'pushNotificationsEnabled': newValue});
```

```

...

* Edit Profile Save: อัปเดตข้อมูล User ด้วย:

```dart

```
// If image changed: newProfilePictureUrl = await _uploadProfileImage(...);
Map<String, dynamic> updateData = { /* firstName, lastName, etc. */ };
if (newProfilePictureUrl != null) { updateData['profilePictureUrl'] =
 newProfilePictureUrl; }
await FirebaseFirestore.instance.collection('users').doc(uid).update(updateData);
```

```

* Image Upload: ใช้ `FirebaseStorage.instance.ref().child(path).putFile(file)` และ `getDownloadURL()`

* **Feedback (`feedback_page.dart`):**

* Save Feedback: อัปเดต History document ด้วย:

```dart

```
await FirebaseFirestore.instance
 .collection('users').doc(uid)
 .collection('healthHistory').doc(selectedHistoryId)
 .update({'feedbackRating': rating, 'feedbackComment': comment, ...});
```

```

* **QR Scan (`scan_page.dart`):**

* รับค่า: ใช้ `controller.scannedDataStream.listen(...)` และคืนค่าด้วย

`Navigator.pop(context, scanData.code)` (จาก package `qr_code_scanner`)

7.2.2 เทคโนโลยีและฟังก์ชัน Firebase ที่ใช้เพิ่มเติม: **

7.2.2.1 Firestore Rules Helpers:**

```javascript

```
function isUser(userId) { return request.auth != null && request.auth.uid ==
userId; }

function isStaff() { return request.auth != null &&
get(/databases/${database}/documents/users/${request.auth.uid}).data.userTyp
e == 'staff'; }
...`
```

### 7.2.2.2 Real-time Read (StreamBuilder):\*\*

```dart

```
StreamBuilder<QuerySnapshot>(  
  stream: _queueStream, // From .snapshots()  
  builder: (context, snapshot) {  
    if (snapshot.hasData) {  
      final docs = snapshot.data!.docs;  
      return ListView.builder(...);  
    } // ... handle loading/error ...  
  }  
)`
```

7.2.2.3 Storage Upload (Conceptual):**

```
```dart
Future<String?> _uploadImage(XFile imageFile, String path) async {
try {
final storageRef = FirebaseStorage.instance.ref().child(path);
await storageRef.putFile(File(imageFile.path));
return await storageRef.getDownloadURL();
} catch (e) { return null; }
}
```
```

```

#### 7.2.2.4 FCM Token Handling:\*\*

```
```dart
// Get Token
String? fcmToken = await FirebaseMessaging.instance.getToken();
// Save Token
await FirebaseFirestore.instance.collection('users').doc(uid).set({'fcmToken':
fcmToken}, SetOptions(merge: true));
// Listen for Refresh
FirebaseMessaging.instance.onTokenRefresh.listen((newToken) =>
_saveFcmToken(newToken));
```
```

```

7.2.2.5 Cloud Function Trigger (Conceptual `index.js`):**

```
```javascript
const functions = require("firebase-functions");
const admin = require("firebase-admin");
admin.initializeApp();

exports.sendPushOnNewNotification = functions.firestore
 .document("/users/{userId}/notifications/{notificationId}")
 .onCreate(async (snap, context) => {
 const userId = context.params.userId;
 const notificationPayload = snap.data();
 // TODO: Get user's FCM token from /users/{userId}
 // TODO: Check user's notification preference
 // TODO: Construct FCM message
 // TODO: Send message using admin.messaging().sendToDevice(token,
 message);
 return null;
});
```

```

7.2.3 การใช้ Packages และความสามารถอื่นๆ

- * `http`: `final response = await http.get(Uri.parse(apiUrl));`
- * `image_picker`: `final XFile? image = await ImagePicker().pickImage(source: ...);`
- * `geolocator`: `Position? position = await Geolocator.getCurrentPosition();`
- * `flutter_map`: `FlutterMap(options: MapOptions(...), children: [TileLayer(...), MarkerLayer(...)])`
- * `qr_code_scanner`: `QRView(key: qrKey, onQRViewCreated: _onQRViewCreated, ...)`

7.2.4 ปัญหาที่พบและการแก้ไข

- * **Gradle:** `flutter clean`, เช็ค `JAVA_HOME`, แก้ `gradle-wrapper.properties`
- * **iOS Build:** `cd ios && rm -rf Pods Podfile.lock && pod install --repo-update && cd ..`
- * **Firestore Rules ('permission-denied'):** แก้ไข `allow read/write/create` ให้ตรง เนื่องจาก เช่น:


```
``javascript
// Example fix for reading history
match /healthHistory/{historyId} {
  allow read: if isUser(userId) || isStaff(); // Correct condition
}
...``
```

แหล่งอ้างอิง

Diagramming Ai เครื่องมือสร้าง User Journey Map ในรูปแบบ Graphviz

<https://diagrammingai.com>

Google Form เครื่องมือเพื่อทำ User Research ในรูปแบบ Questionnaire

<https://docs.google.com/forms/u/0/>

Jira เครื่องมือในการทำ Project Management

<https://www.atlassian.com/software/jira>

Figma เครื่องมือเพื่อทำงาน UX Design และ Prototype Clickable User Flow

<https://www.figma.com/>