



รายงาน

เรื่อง โครงการการออกแบบและพัฒนา Mobile Application สำหรับธุรกิจ

ระยะที่ 2 เสนอแผนโครงการ

แอปพลิเคชันติดตามสุขภาพ Healthacker

เสนอ

ผศ. ดร.	ศิริเพ็ญ	พงษ์ไพเชษฐ์
ดร.	สุรเดช	อินทกรณ์

จัดทำโดย

กลุ่ม 10

นางสาว ัญญารัตน์ วุฒิรุ่งเรืองสกุล 6687063

นางสาว เขมิสรา อนรรฆมงคล 6687074

โครงการนี้เป็นส่วนหนึ่งของรายวิชา

ITDS283 Mobile Application Development

ภาคการศึกษาที่ 2 ปีการศึกษา 2567

คณะเทคโนโลยีสารสนเทศและการสื่อสาร มหาวิทยาลัยมหิดล

คำนำ

รายงานฉบับนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของการศึกษารายวิชา ITDS283 Mobile Application Development โดยมีวัตถุประสงค์เพื่อศึกษาและทำความเข้าใจในเรื่องการออกแบบและพัฒนา Mobile Application สำหรับธุรกิจ

คณะผู้จัดทำหวังว่า รายงานนี้จะประโยชน์แก่ผู้อ่าน ในเรื่องการออกแบบและพัฒนา Mobile Application สำหรับธุรกิจ และสามารถนำความรู้ที่ได้ไปปรับใช้ในการศึกษาหรือการทำงานในอนาคต หากมีข้อเสนอแนะหรือข้อผิดพลาดประการใด คณะผู้จัดทำยินดีน้อมรับและขออภัยมา ณ ที่นี้

สารบัญ

หัวข้อ	หน้า
บทที่ 1 รายละเอียดของโครงการ	
1.1 ที่มาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์ของ Mobile Application	2
1.3 กลุ่มเป้าหมายของ Mobile Application	2
1.4 ศึกษาเปรียบเทียบกับ Mobile Application อื่นที่มีอยู่ในปัจจุบัน	10
บทที่ 2 User Journey Map	11
บทที่ 3 User Flow	13
บทที่ 4 แผนงานของโครงการ	25
บทที่ 5 การพัฒนา Mobile Application	
5.1 ผลการพัฒนาแอปพลิเคชัน	28
5.2 ตัวอย่าง Source Code ที่น่าสนใจ	37
บทที่ 6 การทดสอบ Mobile Application	
6.1 Test-Case	57
6.2 ผลการทดสอบจากกลุ่มเป้าหมาย	59
แหล่งอ้างอิง	63

บทที่ 1

รายละเอียดของโครงการ

1.1 ที่มาและความสำคัญของโครงการ

ในปัจจุบันการดูแลสุขภาพเป็นสิ่งสำคัญอย่างยิ่ง เนื่องจากความเร่งรีบและภาระงานที่มากมายในชีวิตประจำวัน ทำให้ผู้คนมักจะลืมใส่ใจและดูแลสุขภาพของตนเองอย่างเหมาะสม โดยเฉพาะอย่างยิ่งในเรื่องของการทานยาให้ตรงเวลา และการตรวจสุขภาพตามนัดหมายจากโรงพยาบาล ซึ่งหากไม่สามารถจัดการได้ดีอาจส่งผลกระทบต่อสุขภาพในระยะยาว และทำให้การรักษาหรือการฟื้นตัวจากอาการเจ็บป่วยไม่เป็นไปตามที่คาดหวัง จากบทความบนเว็บไซต์ [“หากลืมหินยาตามเวลา ควรทำอะไร”](#) ได้กล่าวไว้ว่า “การลืมหินยาเป็นปัญหาที่พบได้บ่อย โดยเฉพาะในผู้สูงอายุหรือผู้ที่ต้องรับประทานยาหลายชนิด ซึ่งอาจส่งผลกระทบต่อประสิทธิภาพในการรักษาและการฟื้นตัวจากอาการเจ็บป่วย บางครั้งการลืมหินยาเพียงมื้อเดียวอาจไม่ส่งผลร้ายแรง แต่หากลืมหินยาบ่อย ๆ อาจทำให้การรักษาไม่ได้ผลหรือเกิดภาวะแทรกซ้อนได้ ” ซึ่งสะท้อนให้เห็นถึงปัญหาที่เกิดขึ้นจากการลืมหินยา หรือการไม่ได้รับการรักษาอย่างต่อเนื่อง ส่งผลให้ผู้ป่วยอาจประสบปัญหาหรือภาวะแทรกซ้อนที่ไม่คาดคิด

จากปัญหาดังกล่าว คณะผู้จัดทำจึงได้ออกแบบและพัฒนาแอปพลิเคชัน Healthacker ซึ่งเป็นแอปพลิเคชันติดตามสุขภาพที่มีฟีเจอร์สำคัญ คือ การเตือนการกินยา เพิ่มนัดหมายโรงพยาบาล และนับรอบประจำเดือนสำหรับผู้หญิง เพื่อช่วยให้ผู้ใช้สามารถติดตามกิจกรรมสุขภาพที่สำคัญได้ง่ายและมีประสิทธิภาพมากขึ้น โดยเฉพาะการทานยาให้ตรงเวลา ซึ่งมีความสำคัญอย่างยิ่งในการรักษาโรคต่างๆ การลืมหินยาในเวลาที่กำหนดอาจทำให้การรักษาล่าช้าหรือไม่ได้ผล ดังนั้นการใช้แอปพลิเคชันเพื่อเตือนการทานยาจึงช่วยลดความเสี่ยงในจุดนี้ได้อย่างมาก นอกจากนี้ ฟีเจอร์การเพิ่มนัดหมายโรงพยาบาลยังช่วยให้ผู้ใช้ไม่พลาดการตรวจสุขภาพประจำหรือการพบแพทย์ตามที่นัดหมายไว้ ซึ่งการเข้ารับการตรวจสุขภาพเป็นการป้องกันและดูแลสุขภาพที่สำคัญอย่างยิ่ง ในส่วนของฟีเจอร์การนับรอบประจำเดือนสำหรับผู้หญิง แอปพลิเคชันยังมีประโยชน์ในการติดตามสุขภาพทางเพศ ช่วยให้ผู้ใช้สามารถเตรียมตัวสำหรับการมีประจำเดือนได้ดียิ่งขึ้น และลดปัญหาที่อาจเกิดจากการไม่รู้จักร่างกายหรือไม่ติดตามรอบประจำเดือนอย่างถูกต้อง

ทั้งนี้ แอปพลิเคชัน Healthacker ยังมีความเกี่ยวข้องกับเป้าหมายการพัฒนาที่ยั่งยืน (Sustainable Development Goals: SDGs) ในเป้าหมายที่ 3 การสร้างหลักประกันการมีสุขภาพที่ดี และส่งเสริมความเป็นอยู่ที่ดีสำหรับทุกคนในทุกช่วงวัย โดยการพัฒนาแอปพลิเคชัน Healthacker จะช่วยให้ผู้ใช้สามารถดูแลสุขภาพทั้งในผู้สูงอายุ ผู้ที่มีโรคประจำตัว หรือแม้แต่ในกลุ่มผู้หญิงที่ต้องการติดตามสุขภาพทางเพศได้อย่างมีประสิทธิภาพ ลด

ปัญหาการลืมทานยาและการตรวจสุขภาพ และยังมีผลในการลดภาระค่าใช้จ่ายทางการแพทย์จากการรักษาที่ล่าช้าหรือไม่เต็มประสิทธิภาพ จึงเป็นเครื่องมือที่สำคัญในการส่งเสริมสุขภาพที่ดีและยั่งยืนแก่ผู้ใช้ทุกคนในสังคม

1.2 วัตถุประสงค์ของ Mobile Application

- 1.2.1 เพื่อพัฒนาแอปพลิเคชันที่ช่วยให้ผู้ใช้สามารถติดตามและจัดการการทานยาได้อย่างมีประสิทธิภาพ
- 1.2.2 เพื่อสร้างเครื่องมือในการบันทึกและเตือนการนัดหมายของทางโรงพยาบาล
- 1.2.3 เพื่อช่วยติดตามและคำนวณรอบประจำเดือนสำหรับผู้หญิง

1.3 กลุ่มเป้าหมายของ Mobile Application

จากการศึกษาข้อมูลจากการตอบแบบสอบถามของผู้ใช้งานจำนวน 20 คน โดยมีคำถาม 3 ส่วน ดังนี้

ส่วนที่ 1 คำถามเกี่ยวกับข้อมูลพื้นฐานของผู้ใช้งาน ได้แก่

- อายุ
- และเพศ

ส่วนที่ 2 คำถามเกี่ยวกับข้อมูลสุขภาพเบื้องต้น ได้แก่

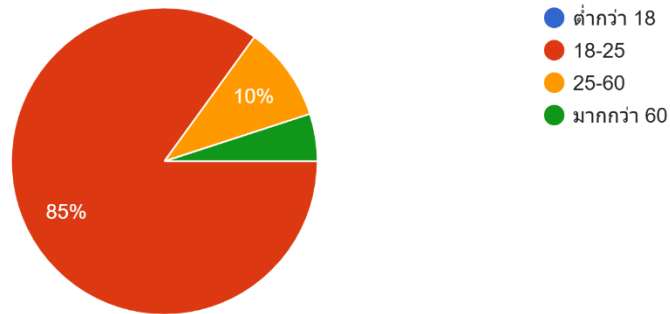
- ท่านมีโรคประจำตัวหรือไม่
 - ท่านมียาที่ต้องทานเป็นประจำหรือไม่
 - ท่านไปโรงพยาบาลบ่อยแค่ไหน
 - ท่านเคยลืมทานยาหรือลืมนัดหมายจากทางโรงพยาบาลหรือไม่
 - ท่านต้องการเครื่องมือช่วยเตือนการกินยาหรือนัดหมายจากโรงพยาบาลหรือไม่
- และคำถามเฉพาะเพศหญิงเพิ่มเติม ได้แก่
- ท่านมีรอบเดือนหรือไม่ และมาอย่างสม่ำเสมอหรือไม่
 - ท่านต้องการเครื่องมือเพื่อติดตามวันที่มีรอบเดือนหรือไม่

โดยมีผลจากแบบสำรวจข้างต้น ดังนี้

ผลสำรวจของคำถามเกี่ยวกับข้อมูลพื้นฐานของผู้ใช้งาน

อายุ

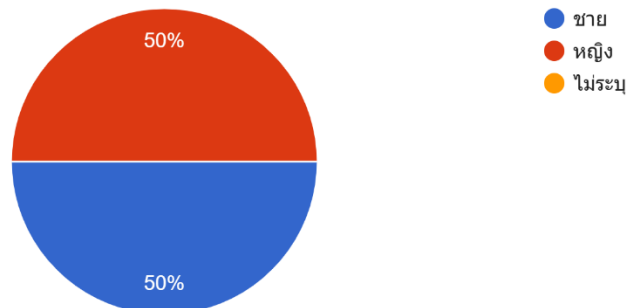
20 responses



(รูปภาพที่ 1 : ผลสำรวจของคำถาม อายุ)

เพศ

20 responses



(รูปภาพที่ 2 : ผลสำรวจของคำถาม เพศ)

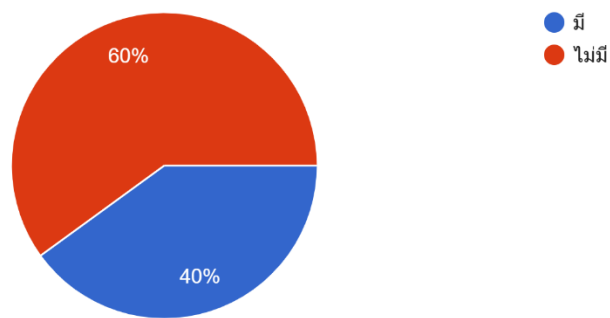
จากผลการสำรวจ กลุ่มผู้ใช้งานมีสัดส่วนอายุอยู่ในช่วง 18 ถึง 25 ปีมากที่สุดคิดเป็นร้อยละ 85 สัดส่วนรองลงมาคือ 25 ถึง 60 ปี และมากกว่า 60 ปี คิดเป็นร้อยละ 10 และ 5 ตามลำดับ และมีสัดส่วนของเพศเป็นเพศชายร้อยละ 50 และเพศหญิงร้อยละ 50

ผลสำรวจของคำถามเกี่ยวกับข้อมูลสุขภาพเบื้องต้น

เพศหญิง

ท่านมีโรคประจำตัวหรือไม่

10 responses

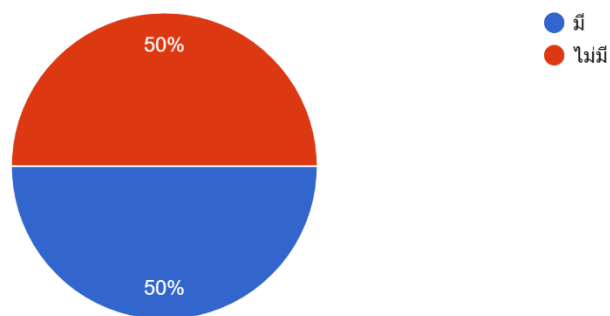


(รูปภาพที่ 3 : ผลสำรวจของคำถามโรคประจำตัวของเพศหญิง)

จากผลสำรวจพบว่า ผู้ใช้งานเพศหญิงมีโรคประจำตัวร้อยละ 40 และไม่มีโรคประจำตัวร้อยละ 60

ท่านมียาที่ต้องทานเป็นประจำหรือไม่

10 responses

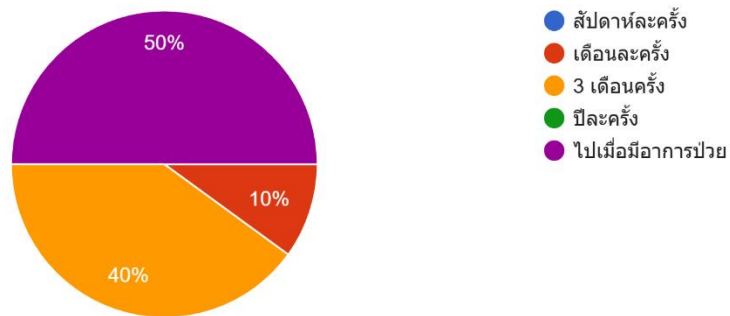


(รูปภาพที่ 4 : ผลสำรวจของคำถามยาที่ท่านเป็นประจำของเพศหญิง)

จากผลสำรวจพบว่า ผู้ใช้งานเพศหญิงมียาที่ต้องทานเป็นประจำร้อยละ 50 และไม่มียาที่ต้องทานเป็นประจำร้อยละ 50

ท่านไปโรงพยาบาลบ่อยแค่ไหน

10 responses

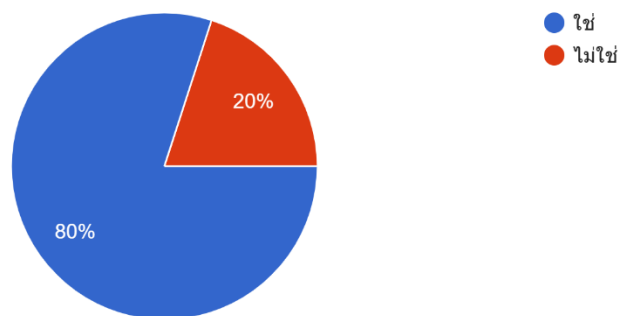


(รูปภาพที่ 5 : ผลสำรวจของคำถามการไปโรงพยาบาลของเพศหญิง)

จากผลสำรวจพบว่า ผู้ใช้งานเพศหญิงมีส่วนร่วมการไปโรงพยาบาลเมื่อมีอาการป่วยมากที่สุดคิดเป็นร้อยละ 50 สัดส่วนรองลงมาคือไปโรงพยาบาล 3 เดือนครั้งร้อยละ 40 และไปโรงพยาบาลเดือนละครั้งร้อยละ 10

ท่านเคยลิ้มทานยาหรือลิ้มน้ำดื่มจากทางโรงพยาบาลหรือไม่

10 responses

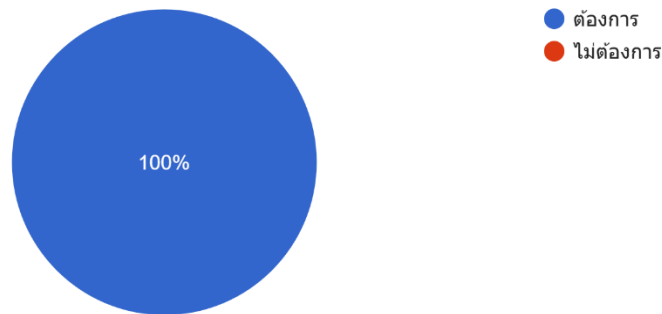


(รูปภาพที่ 6 : ผลสำรวจของคำถามการลิ้มทานยาหรือน้ำดื่มจากโรงพยาบาลของเพศหญิง)

จากผลสำรวจพบว่า ผู้ใช้งานเพศหญิงมีส่วนร่วมการลิ้มทานยาหรือน้ำดื่มจากโรงพยาบาลคิดเป็นร้อยละ 80 และไม่เคยลิ้มร้อยละ 20

ท่านต้องการเครื่องมือช่วยเตือนการกินยาหรือนัดหมายจากโรงพยาบาลหรือไม่

10 responses

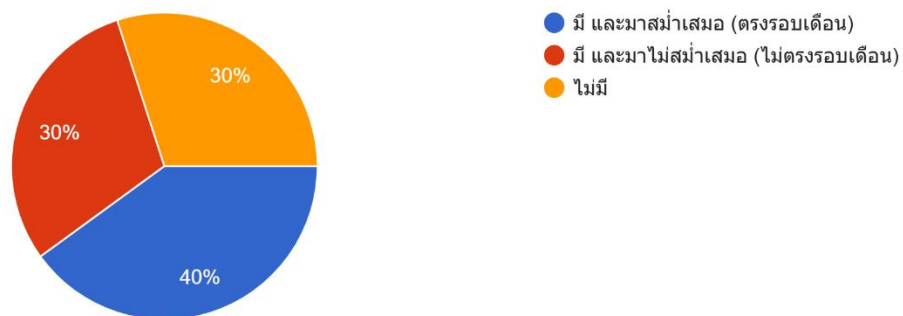


(รูปภาพที่ 7 : ผลสำรวจของคำถามความต้องการเครื่องมือช่วยเหลือการทานยาหรือนัดหมายของเพศหญิง)

จากผลสำรวจพบว่า ผู้ใช้งานเพศหญิงทั้งหมดต้องการเครื่องมือที่ช่วยเตือนการกินยาหรือนัดหมายจากทางโรงพยาบาล

ท่านมีรอบเดือนหรือไม่ และมาอย่างสม่ำเสมอหรือไม่

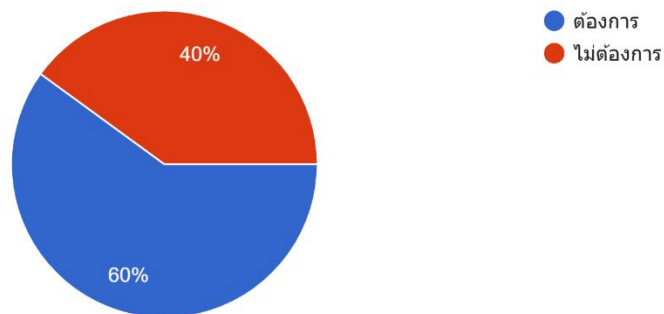
10 responses



(รูปภาพที่ 8 : ผลสำรวจของคำถามการมีรอบเดือนของเพศหญิง)

จากผลสำรวจพบว่า ผู้ใช้งานเพศหญิงมีส่วนการมีรอบเดือนและมาอย่างสม่ำเสมอมากที่สุดคิดเป็นร้อยละ 40 ส่วนรองลงมาคือ มีและไม่สม่ำเสมอคิดเป็นร้อยละ 30 และไม่มีการมีรอบเดือนคิดเป็นร้อยละ 30

ท่านต้องการเครื่องมือเพื่อติดตามวันที่มีรอบเดือนหรือไม่
10 responses

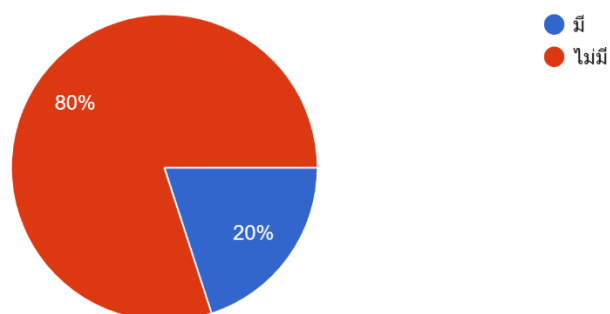


(รูปภาพที่ 9 : ผลสำรวจของคำถามความต้องการเครื่องมือช่วยเหลือการติดตามรอบเดือนของเพศหญิง)

จากผลสำรวจพบว่า ผู้ใช้งานเพศหญิงต้องการเครื่องมือที่เพื่อติดตามวันที่มีรอบเดือนร้อยละ 60 และไม่ต้องการร้อยละ 40

เพศชาย

ท่านมีโรคประจำตัวหรือไม่
10 responses

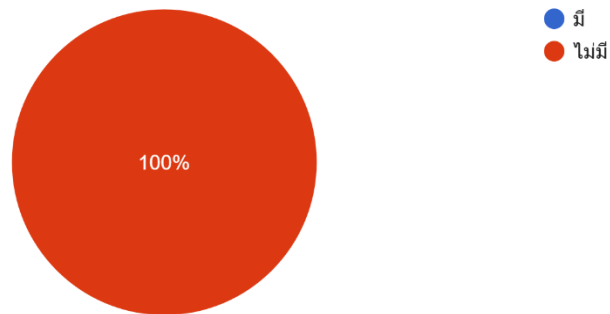


(รูปภาพที่ 10 : ผลสำรวจของคำถามโรคประจำตัวของเพศชาย)

จากผลสำรวจพบว่า ผู้ใช้งานเพศชายมีโรคประจำตัวร้อยละ 20 และไม่มีโรคประจำตัวร้อยละ 80

ท่านมียาที่ต้องทานเป็นประจำหรือไม่

10 responses

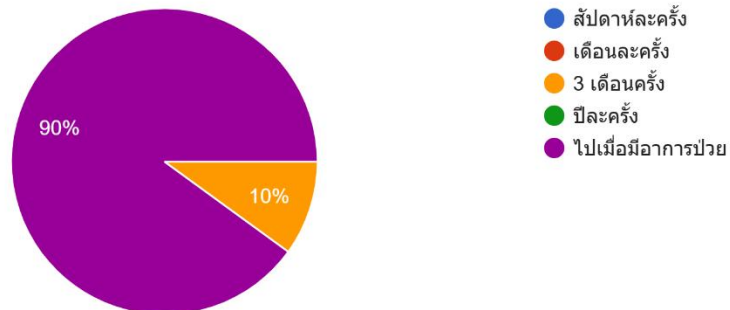


(รูปภาพที่ 11 : ผลสำรวจของคำถามยาที่ท่านเป็นประจำของเพศชาย)

จากผลสำรวจพบว่า ผู้ใช้งานเพศชายทั้งหมดไม่มียาที่ต้องทานเป็นประจำ

ท่านไปโรงพยาบาลบ่อยแค่ไหน

10 responses

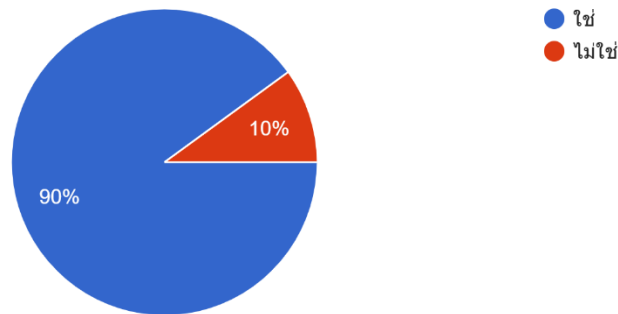


(รูปภาพที่ 12 : ผลสำรวจของคำถามการไปโรงพยาบาลของเพศชาย)

จากผลสำรวจพบว่า ผู้ใช้งานเพศชายมีส่วนการไปโรงพยาบาลเมื่อมีอาการป่วยมากที่สุดคิดเป็นร้อยละ 90 สัดส่วนรองลงมาคือไปโรงพยาบาล 3 เดือนครั้งร้อยละ 10

ท่านเคยลิ้มทานยาหรือลิ้มรสชาติของน้ำดื่มจากทางโรงพยาบาลหรือไม่

10 responses

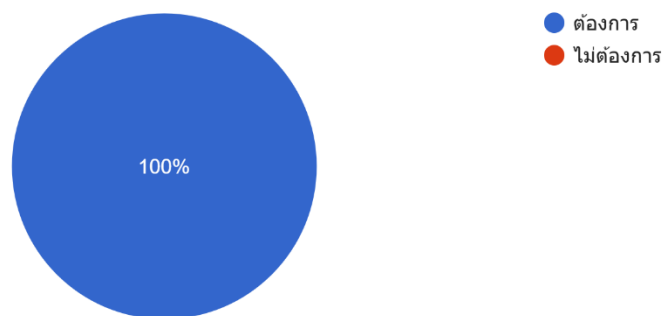


(รูปภาพที่ 13 : ผลสำรวจของคำถามการลิ้มทานยาหรือรสชาติของน้ำดื่มของเพศชาย)

จากผลสำรวจพบว่า ผู้ใช้งานเพศชายมีส่วนการลิ้มทานยาหรือรสชาติของน้ำดื่มจากโรงพยาบาลคิดเป็นร้อยละ 90 และไม่เคยลิ้มร้อยละ 10

ท่านต้องการเครื่องมือช่วยเตือนการกินยาหรือรสชาติของน้ำดื่มจากทางโรงพยาบาลหรือไม่

10 responses



(รูปภาพที่ 14 : ผลสำรวจของคำถามความต้องการเครื่องมือช่วยเตือนการทานยาหรือรสชาติของน้ำดื่มของเพศชาย)

จากผลสำรวจพบว่า ผู้ใช้งานเพศชายทั้งหมดต้องการเครื่องมือที่ช่วยเตือนการกินยาหรือรสชาติของน้ำดื่มจากทางโรงพยาบาล

จากผลการสำรวจข้างต้น กลุ่มเป้าหมายคือเพศหญิง อายุช่วง 18-25 ปี ซึ่งมีปัญหาเกี่ยวกับการทานยา และลิมนัดจากทางโรงพยาบาล และต้องการเครื่องมือสำหรับเตือนการทานยา นัดจากทางโรงพยาบาล และวันที่มีรอบเดือน และรองลงมาคือเพศชาย อายุช่วง 18-25 ปี ที่ต้องการเครื่องมือสำหรับเตือนการทานยาและนัดจากทางโรงพยาบาล

1.4 ศึกษาเปรียบเทียบกับ Mobile Application อื่นที่มีอยู่ในปัจจุบัน

คุณสมบัติของแอปพลิเคชัน	Max	Flo	Healthacker
1. สามารถใช้งานบนสมาร์ตโฟนได้	✓	✓	✓
2. ฟังก์ชันเตือนการกินยา	✓	✗	✓
3. ฟังก์ชันเตือนการมาของรอบเดือน	✗	✓	✓
4. ฟังก์ชันเตือนนัดของโรงพยาบาล	✗	✗	✓

(ตารางเปรียบเทียบกับแอปพลิเคชัน Max และ Flo)

จากการศึกษาแอปพลิเคชัน Max ซึ่งเป็นแอปพลิเคชันเกี่ยวกับเตือนการกินยา และแอปพลิเคชัน Flo ซึ่งเป็นแอปพลิเคชันสำหรับเพศหญิงที่มีฟังก์ชันการนับรอบเดือน พบว่าในแต่ละแอปพลิเคชันมีฟังก์ชันที่ต่างกันโดยสิ้นเชิง เมื่อผู้ใช้ต้องการใช้งานอาจทำให้ยากลำบาก แอปพลิเคชัน Healthacker จึงเป็นแอปพลิเคชันที่มีจุดแข็งคือการรวมฟังก์ชันที่จำเป็นต่อผู้ใช้งานไว้ในทีเดียว และมีฟังก์ชันเพิ่มเติมคือการเตือนการนัดหมายจากโรงพยาบาลทำให้ผู้ใช้งานมีความสะดวกสบายมากยิ่งขึ้น

บทที่ 2

User Journey Map



The User Journey Map for Healthacker is presented in a table with 6 rows and 5 columns. The columns represent stages: Awareness, Consideration, Contact, and Advocacy. The rows represent different aspects of the user experience: User Actions, Touchpoints, Sentiments, Emotions, and Possible Solutions. A legend at the bottom right shows five colored squares with corresponding smiley faces: red (sad), orange (neutral), yellow (happy), green (very happy), and blue (extremely happy).

	Awareness	Consideration	Contact	Advocacy
User Actions	ผู้ใช้สืบทายา หรือนัดหมายโรงพยาบาล หรือวันที่มีรอบเดือน	ผู้ใช้ค้นหาแอปพลิเคชันของเรา	ผู้ใช้เลือกฟีเจอร์ที่ต้องการใช้งาน	ผู้ใช้แนะนำแอปพลิเคชันให้คนรอบตัว หรือคนที่ปัญหา
Touchpoints	ผู้ใช้ค้นหาตัวช่วยจดบันทึก	ผู้ใช้โหลดและทดลองใช้แอปพลิเคชัน	ผู้ใช้กรอกข้อมูลของสิ่งที่ต้องการเตือน	ผู้ใช้รีวิวแอปพลิเคชัน
Sentiments	ผู้ใช้รู้สึกว่ามีความยุ่งยากในการจดบันทึก	ผู้ใช้กังวลว่าแอปพลิเคชันมีการใช้งานที่ยุ่งยาก	ผู้ใช้รู้สึกหายกังวลว่าสืบทายา หรือสืบนัดโรงพยาบาล หรือสืบนัดวันมีรอบเดือน	ผู้ใช้ไม่พบปัญหาเรื่องการสืบทายา นัดโรงพยาบาล รอบเดือน นกกระทบแผนอื่นๆในชีวิตประจำวัน
Emotions				
Possible Solutions	โปรโมตแอปพลิเคชันให้เป็น ที่รู้จักมากยิ่งขึ้น	มีคำอธิบายแต่ละฟีเจอร์อย่างละเอียด และตัวอย่างการใช้งาน	มีแจ้งเตือนกิจกรรมที่ใกล้ถึง	

(รูปภาพที่ 15 : User Journey Map)

จาก User Journey Map ข้างต้น แอปพลิเคชัน Healthacker สามารถลดอุปสรรคที่เกิดจากปัญหาที่ผู้ใช้พบเจอได้ ดังนี้

1. Awareness

- ปัญหา : ผู้ใช้สืบทายา หรือสืบนัดของทางโรงพยาบาล หรือสืบนัดวันที่มีรอบเดือนทำให้สิ่งเหล่านี้อาจเป็นปัญหาในการวางแผนการใช้ชีวิตได้ และเครื่องมือที่มีในปัจจุบันอาจมีความยุ่งยากในการใช้งาน
- แนวทางแก้ไขปัญหา : โปรโมตแอปพลิเคชันให้เป็นที่รู้จักในกลุ่มเป้าหมายมากขึ้น โดยการทำโฆษณาลงแพลตฟอร์มโซเชียลมีเดียต่างๆ เช่น Instagram Facebook ซึ่งเป็นแพลตฟอร์มที่นิยมในกลุ่มวัยรุ่น

2. Consideration

- ปัญหา : ผู้ใช้มีความกังวลว่าจะใช้แอปพลิเคชันไม่เป็น หรือกังวลว่าแอปพลิเคชันจะมีความซับซ้อนในการเลือกใช้งานฟีเจอร์ต่างๆ
- แนวทางแก้ไขปัญหา : มีคู่มือการใช้งานแอปพลิเคชันพื้นฐานตั้งแต่เริ่มต้นอย่างละเอียด

3. Contact

- ปัญหา : ผู้ใช้งานมีความกังวลในการกรอกข้อมูลของแต่ละพีเจอร์
- แนวทางแก้ไขปัญหา : มีคู่มือของแต่ละพีเจอร์อย่างละเอียด เป็นขั้นตอนต่างๆ และมีตัวอย่างหน้าต่างเมื่อมีการกรอกข้อมูล

4. Advocacy

- ปัญหา : ไม่มีช่องทางในการบอกต่อประสบการณ์และข้อดีข้อเสียของการใช้แอปพลิเคชัน
- แนวทางแก้ไขปัญหา : มีช่องทางให้ผู้ใช้สามารถรีวิวการใช้งานแอปพลิเคชันได้

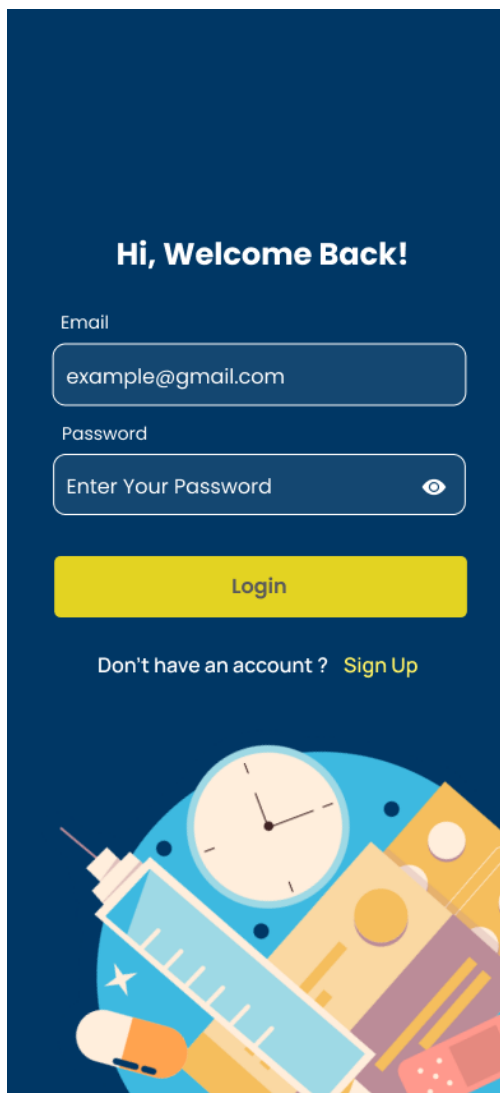
บทที่ 3

User Flow

แผนภาพ user flow แสดงถึงแผนภาพเมื่อผู้ใช้ใช้งานแอปพลิเคชัน โดยสามารถแบ่งหน้าการใช้งานได้ ดังนี้

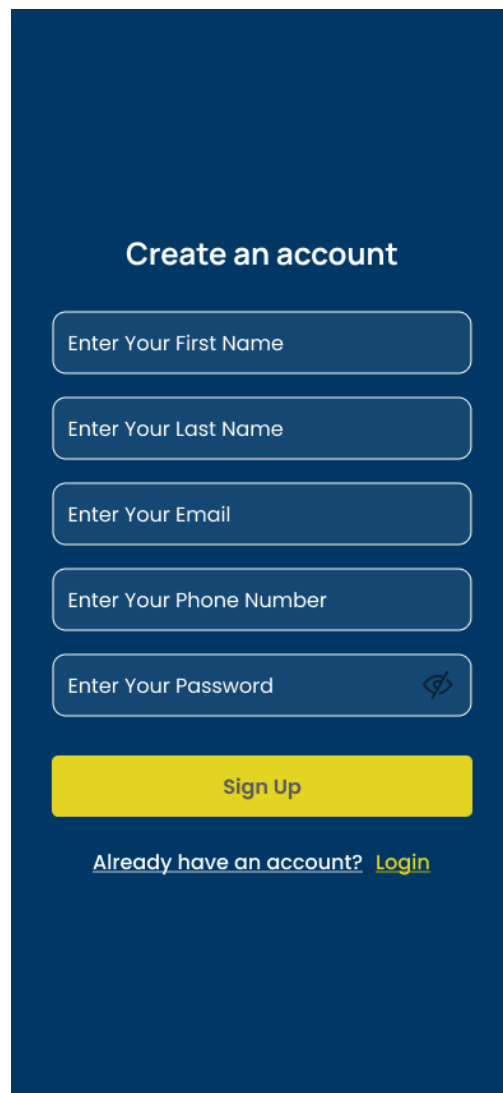
1. Login page and Create account

เมื่อเข้าตัวแอปพลิเคชันครั้งแรก ผู้ใช้จะต้องทำการเข้าสู่ระบบก่อน โดยใช้ Email และ Password ที่ลงทะเบียนไว้ หากยังไม่มีบัญชีสามารถสร้างบัญชีใหม่ได้ โดยกด Sign up ซึ่งจะเชื่อมไปยัง Create account page ผู้ใช้จะต้องใส่ข้อมูลชื่อ-นามสกุล Email เบอร์โทรศัพท์ และตั้งรหัสผ่าน



The login page features a dark blue background. At the top, it says "Hi, Welcome Back!". Below this are two input fields: "Email" with the placeholder "example@gmail.com" and "Password" with the placeholder "Enter Your Password" and an eye icon. A yellow "Login" button is positioned below the password field. At the bottom, there is a link "Don't have an account ? Sign Up" and a colorful illustration of a clock, a syringe, and a pill.

(รูปภาพที่ 16 : login page)

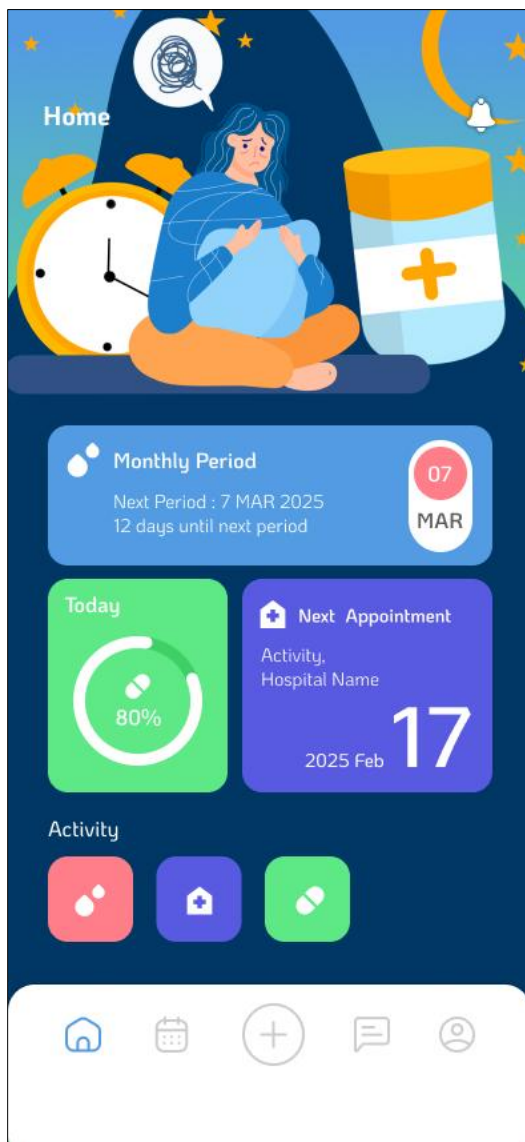


The create account page has a dark blue background. It features five input fields: "Enter Your First Name", "Enter Your Last Name", "Enter Your Email", "Enter Your Phone Number", and "Enter Your Password" with an eye icon. A yellow "Sign Up" button is located below the password field. At the bottom, there is a link "Already have an account? Login".

(รูปภาพที่ 17 : Create account)

2. Home page

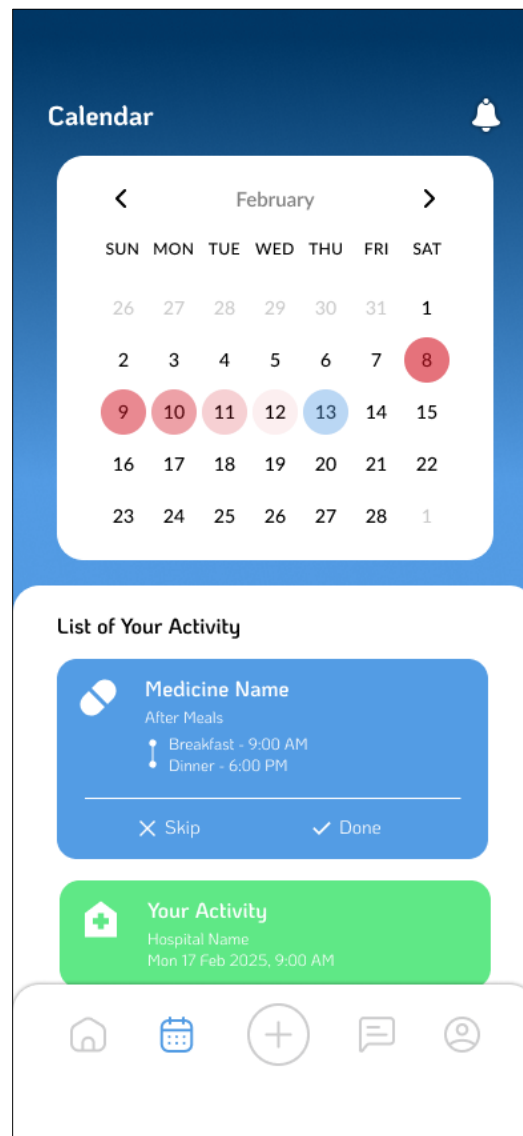
Home page จะแสดงข้อมูล ได้แก่ ข้อมูลสรุปผลการติดตามประจำเดือนโดยจะแสดงผลรอบเดือนครั้งถัดไป ข้อมูลการทานยาในวันนี้ และข้อมูลนัดหมายจากทางโรงพยาบาลในครั้งถัดไป อีกทั้งยังมีช่องทางลัดที่เชื่อมไปยัง Monthly Period page, Doctor's Appointment page และ Take Medicine อีกด้วย



(รูปภาพที่ 18 : Home page)

3. Calendar page

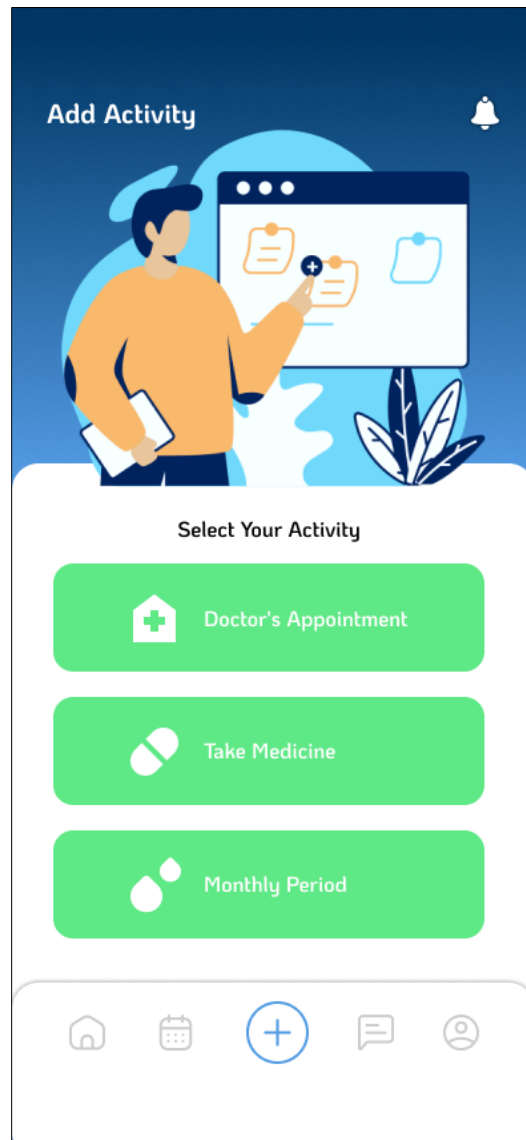
Calendar page ประกอบไปด้วยปฏิทินที่จะสรุปการติดตามและนัดหมายทั้งหมดของผู้ใช้ และแสดง List of Your Activity โดยในส่วนของการติดตามการทานยา จะสามารถกดยืนยันได้ว่าทานยาไปแล้วหรือยังต่อมาส่วนของนัดหมายจากโรงพยาบาล จะแสดงผลชื่อกิจกรรม โรงพยาบาล และวันที่ที่นัดหมาย สุดท้ายในส่วนของการติดตามประจำเดือน จะแสดงผลการบันทึกประจำเดือนของผู้ใช้ อีกทั้งจะแสดงผลการคาดคะเนการมีประจำเดือนครั้งถัดไปอีกด้วย



(รูปภาพที่ 19 : Calendar)

4. Select Activity page

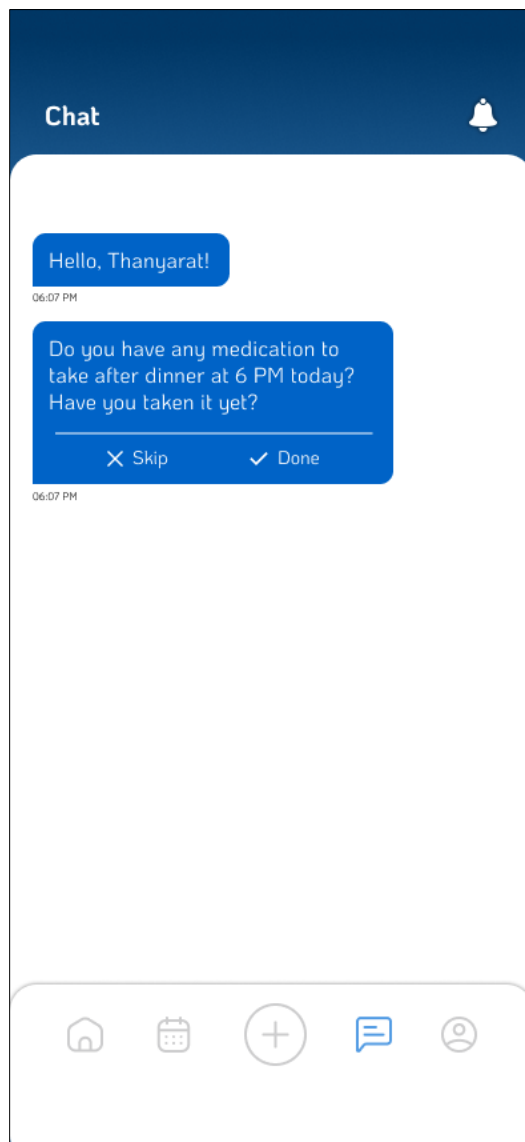
Select activity page ประกอบด้วยตัวเลือกกิจกรรมทั้ง 3 อย่าง คือ 1) Doctor's Appointment ซึ่งเชื่อมไปยัง Doctor's Appointment page 2) Take Medical ซึ่งเชื่อมไปยัง Take Medical page และ 3) Monthly Period ซึ่งเชื่อมไปยัง Monthly Period page



(รูปภาพที่ 20 : Select activity page)

5. Chat

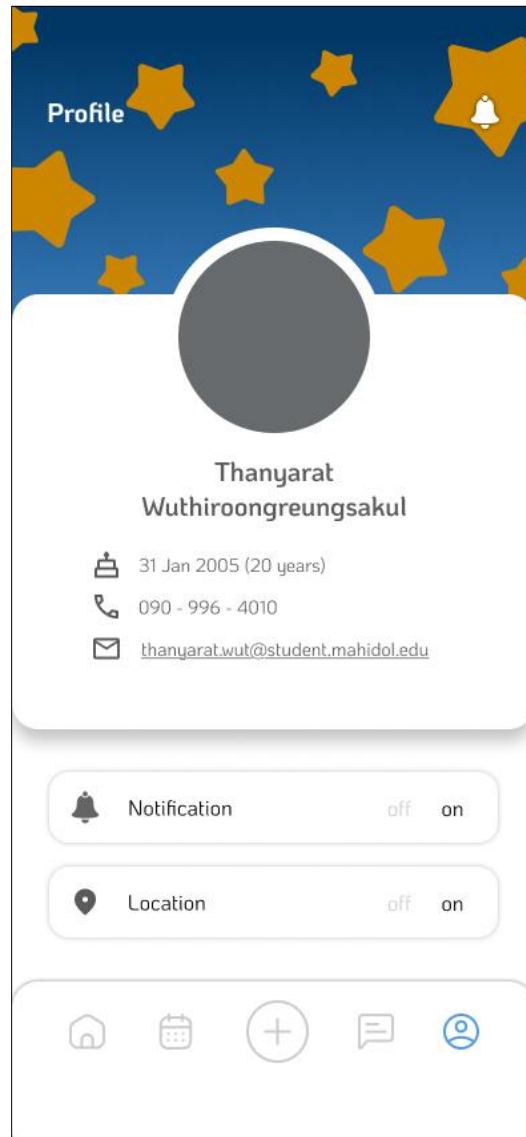
Chat จะแสดงข้อความเตือนจากระบบไปยัง notifications ของอุปกรณ์ โดยข้อความการแจ้งเตือนกินยาจะสามารถกดยืนยันการทานยาได้ และเมื่อยืนยันก็จะมีข้อความตอบกลับอัตโนมัติ



(รูปภาพที่ 21 : Chat)

6. Profile page

Profile page ประกอบด้วยรูป profile ชื่อ-นามสกุล วันเกิด เบอร์โทรศัพท์ Email ส่วนการเปิด-ปิดการแจ้งเตือนและการเข้าถึง location



(รูปภาพที่ 22 : Profile page)

7. Doctor's Appointment page

Doctor's Appointment page จะรับข้อมูลชื่อของกิจกรรม นัศหมายจากโรงพยาบาล วันและเวลาที่นัดหมาย สถานที่ เวลาการแจ้งเตือน โน้ตรายละเอียดต่างๆ ในแต่ละนัดหมายสามารถเลือกสีได้ตามต้องการของผู้ใช้ ส่วนสุดท้ายคือปุ่ม submit โดยจะเชื่อมไปยัง Calendar และ cancel โดยจะเชื่อมไปยัง Select activity page

The image displays two versions of the 'Doctor's Appointment' mobile app interface. Both screens feature a dark blue header with the title 'Doctor's Appointment' and a notification bell icon. The left screen shows a form with the following fields: 'Title' with a blue dot indicator, an 'All day' toggle switch, a date and time range for 'Mon, 17 Feb 2025' from '9:00 AM' to '10:00 AM', a 'Location' field with a location pin icon, a '1 hour before' notification setting, and a 'Note' field with a list icon. At the bottom of the form are 'Cancel' and 'Save' buttons. The right screen shows the same form but with a calendar view for February 2025, where the 17th is selected. The calendar shows days of the week (SU, M, TU, W, TH, F, SA) and dates. The bottom navigation bar on both screens includes icons for Home, Calendar, Add (a blue circle with a plus sign), Messages, and Profile.

(รูปภาพที่ 23 : Doctor's Appointment page)

8. Take Medicine page

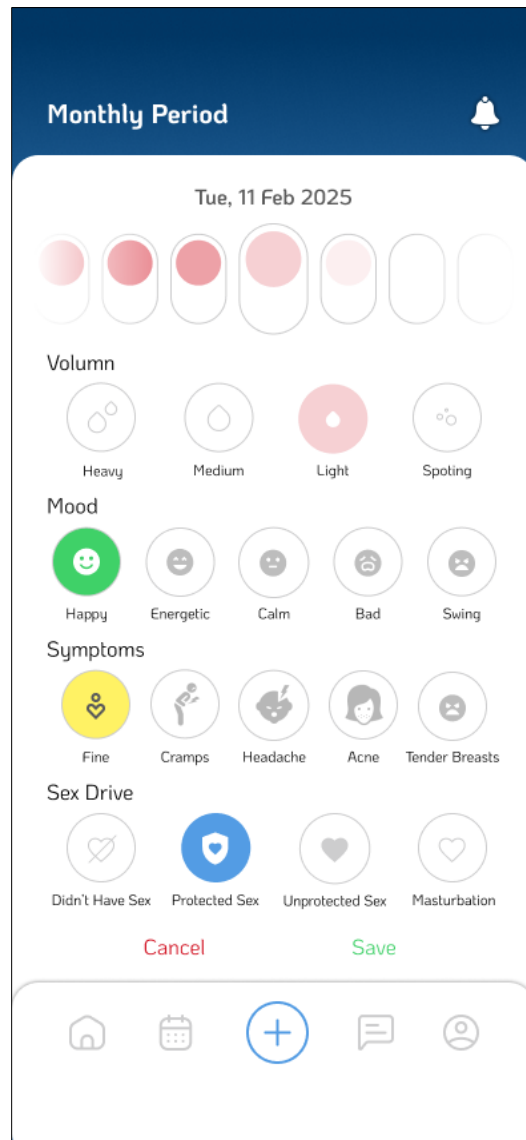
Take Medicine page จะรับข้อมูลชื่อของยา จำนวนยาและการทานต่อวัน การเลือกช่วงเวลา เช้า กลาง เย็น และก่อนนอน การเลือกหลังหรือก่อนมื้ออาหารเมื่อกดเลือกมื้ออาหารเสร็จแล้ว จึงจะแสดงผลให้เลือกเวลาที่ ต้องการทานยา เวลาการแจ้งเตือน โน้ตรายละเอียดต่างๆ ในแต่ละนัดหมายสามารถเลือกสีได้ตามต้องการของผู้ใช้ ส่วนสุดท้ายคือปุ่ม submit โดยจะเชื่อมโยงไปยัง Calendar และ cancel โดยจะเชื่อมโยงไปยัง Select activity page

The image displays two versions of the 'Take Medicine' app interface. Both screens feature a dark blue header with the title 'Take Medicine' and a notification bell icon. Below the header is a section titled 'Add Your Medicine' with a blue circular button. The first screen shows the 'Amount (Tablet)' set to 10 and 'Times Per Day' set to 2. It includes four meal timing icons: Breakfast, Lunch, Dinner, and Bedtime. The 'Before meal' option is selected. Below these are a notification bell icon with '1 hour before' and a 'Note' field. At the bottom are 'Cancel' and 'Save' buttons. The second screen shows the 'Take (Tablet)' set to 10 and 'Times Per Day' set to 2. It includes the same four meal timing icons. The 'After meal' option is selected. Below these are a list of meal times: Breakfast 09:00 AM, Lunch 00:00 AM, Dinner 06:00 PM, and Bedtime 00:00 PM. Below the list is a time picker showing 09:00 AM. Below the time picker is a notification bell icon with '1 hours before' and a 'Note' field. At the bottom are 'Cancel' and 'Save' buttons. Both screens have a bottom navigation bar with icons for Home, Calendar, Add, Chat, and Profile.

(รูปภาพที่ 24 : Take Medicine page)

9. Monthly Period page

Monthly Period page ประกอบไปด้วยปฏิทินขนาดย่อสำหรับติดตามรอบเดือน การเลือกปริมาณของประจำเดือนที่มี อารมณ์ อาการ และพฤติกรรมการมีเพศสัมพันธ์ ส่วนสุดท้ายคือปุ่ม submit โดยจะเชื่อมโยงไปยัง Calendar และ cancel โดยจะเชื่อมโยงไปยัง Select activity page

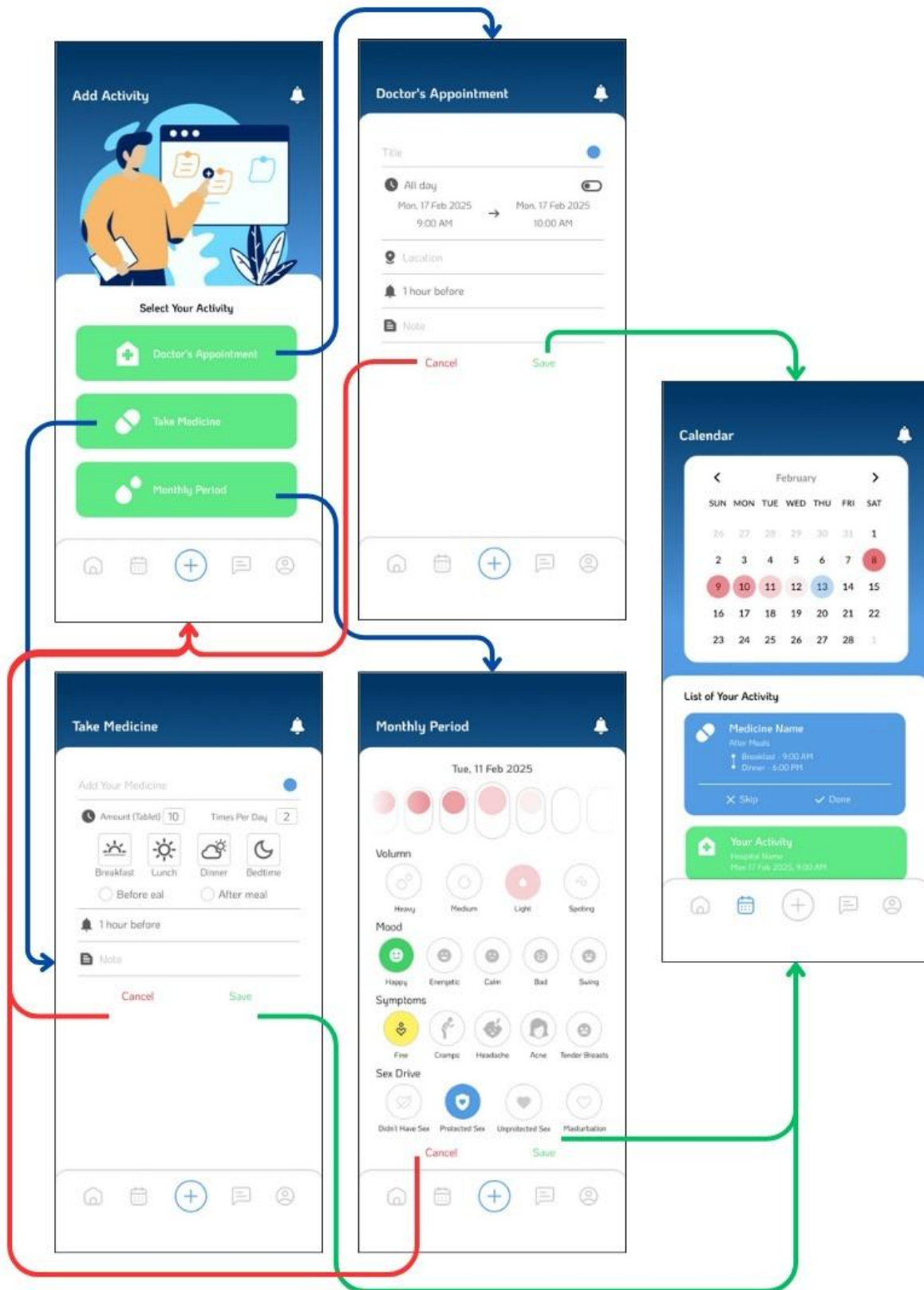


(รูปภาพที่ 25 : Take Medicine page)

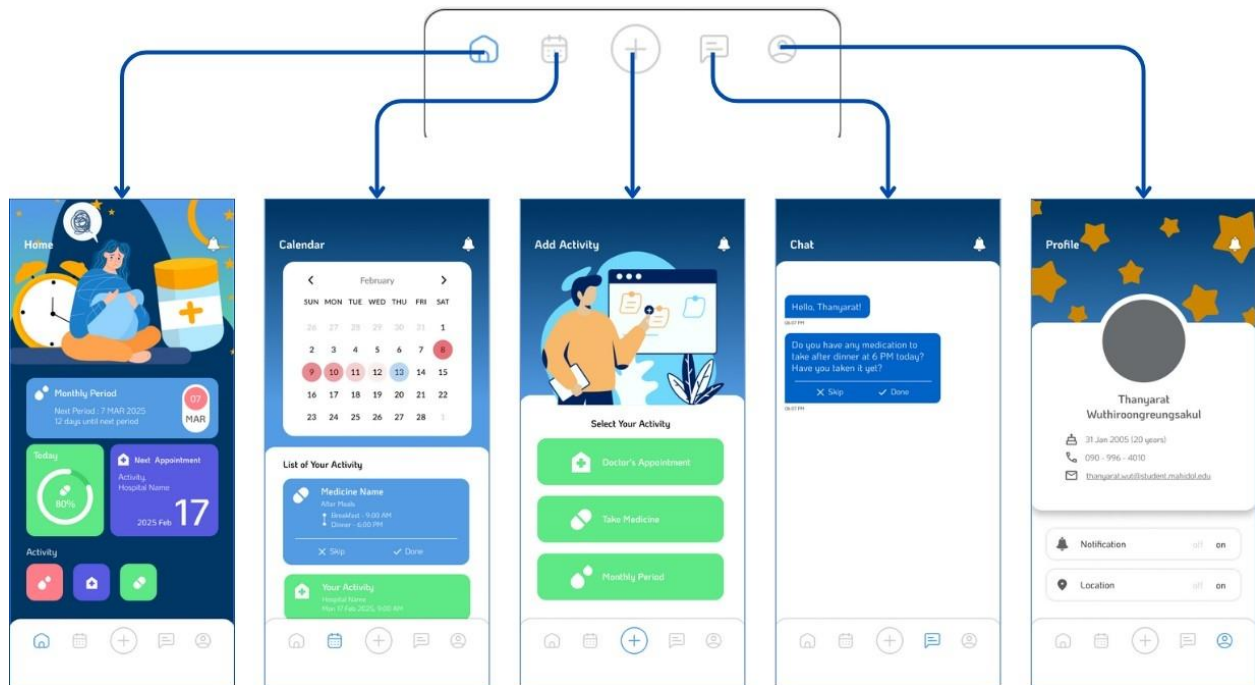
10. แผนภาพ Wireframe



(รูปภาพที่ 26 : Wireframe ของแอปพลิเคชัน)



(รูปภาพที่ 27 : Wireframe ของหน้า Add Activity)



(รูปภาพที่ 28 : Wireframe ของ Navigation bar)

11. Figma

Project Mobile App – Figma

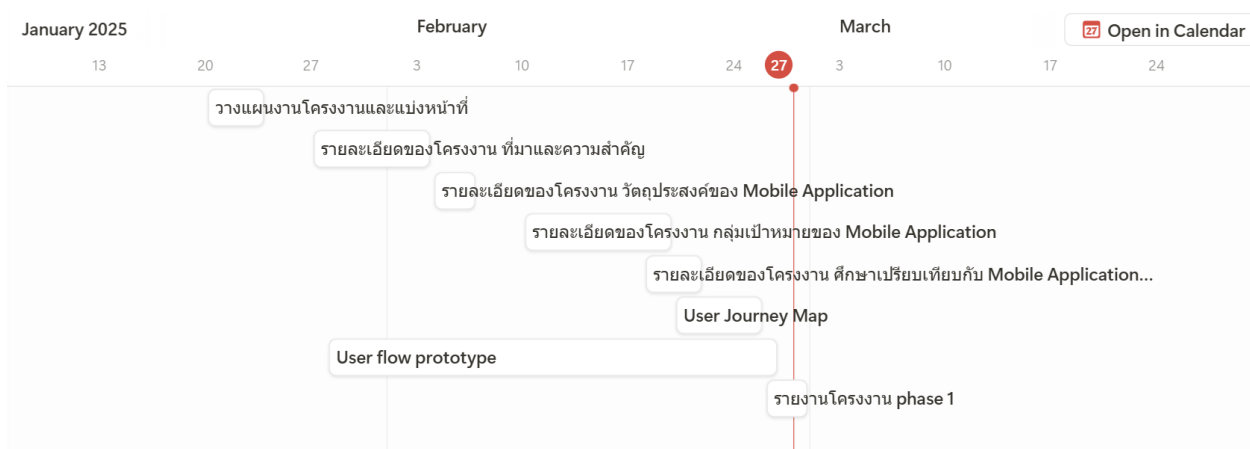
บทที่ 4

แผนงานของโครงการ

ทางคณะผู้จัดทำมีแผนการดำเนินงานโครงการในระยะที่ 1 ดังนี้

📅 แผนงานโครงการ ระยะที่ 1	👤 Assign	📅 Date
วางแผนงานโครงการและแบ่งหน้าที่	K KEMISARA A. 🌐 THANYARAT WUTHIROONGREUNGSAKI	January 20, 2025 → January 23, 2025
รายละเอียดของโครงการ ที่มาและความสำคัญ	K KEMISARA A.	January 27, 2025 → February 3, 2025
User flow prototype	🌐 THANYARAT WUTHIROONGREUNGSAKUL	January 28, 2025 → February 26, 2025
รายละเอียดของโครงการ วัตถุประสงค์ของ Mobile Application	K KEMISARA A.	February 4, 2025 → February 6, 2025
รายละเอียดของโครงการ กลุ่มเป้าหมายของ Mobile Application	K KEMISARA A. 🌐 THANYARAT WUTHIROONGREUNGSAKI	February 10, 2025 → February 19, 2025
รายละเอียดของโครงการ ศึกษาเปรียบเทียบกับ Mobile Application อื่นที่มีอยู่ในปัจจุบัน	K KEMISARA A.	February 18, 2025 → February 21, 2025
User Journey Map	K KEMISARA A. 🌐 THANYARAT WUTHIROONGREUNGSAKI	February 20, 2025 → February 25, 2025
รายงานโครงการ phase 1	K KEMISARA A. 🌐 THANYARAT WUTHIROONGREUNGSAKI	February 26, 2025 → February 28, 2025

(รูปภาพที่ 29 : แผนงานระยะที่ 1 ของโครงการในรูปแบบ Table)



(รูปภาพที่ 30 : แผนงานระยะที่ 1 ของโครงการในรูปแบบ Timeline)

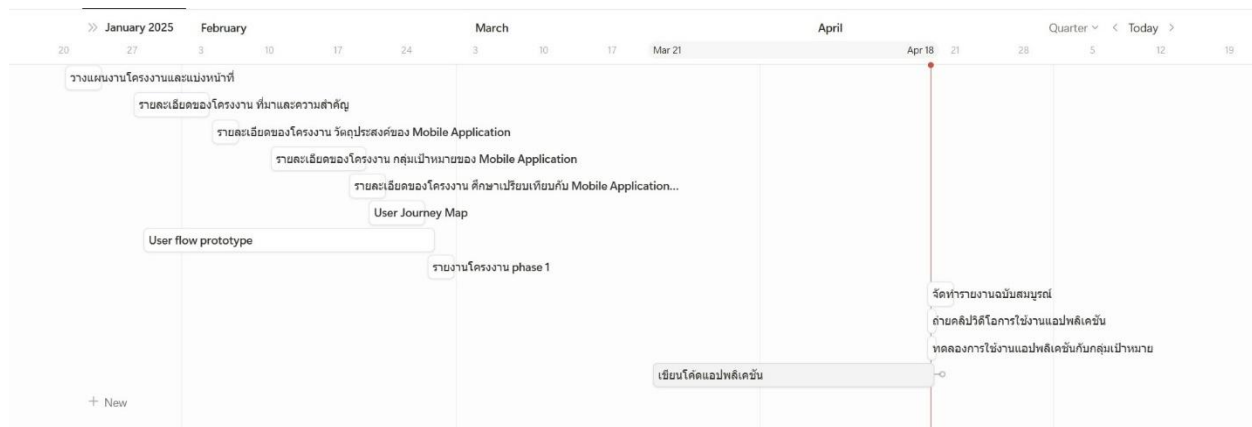
- วางแผนโครงการและแบ่งหน้าที่ของโครงการระยะที่ 1 ตั้งแต่วันที่ 20 มกราคม ถึง 23 มกราคม เป็นเวลา 4 วัน ผู้รับผิดชอบคือ นางสาวธัญญารัตน์และนางสาวเคมีสรา
- จัดทำรายละเอียดของโครงการ หัวข้อที่มาและความสำคัญ ตั้งแต่วันที่ 27 มกราคม ถึง 3 กุมภาพันธ์ เป็นเวลา 8 วัน ผู้รับผิดชอบคือ นางสาวเคมีสรา
- จัดทำรายละเอียดของโครงการ หัวข้อวัตถุประสงค์ของ Mobile Application ตั้งแต่วันที่ 4 กุมภาพันธ์ ถึง 6 กุมภาพันธ์ เป็นเวลา 3 วัน ผู้รับผิดชอบคือ นางสาวเคมีสรา

- จัดทำรายละเอียดของโครงการ หัวข้อกลุ่มเป้าหมายของ Mobile Application ตั้งแต่วันที่ 10 กุมภาพันธ์ ถึง 19 กุมภาพันธ์ เป็นเวลา 10 วัน ผู้รับผิดชอบคือ นางสาวธัญญารัตน์และนางสาวเขมิสรา
- จัดทำรายละเอียดของโครงการ หัวข้อศึกษาเปรียบเทียบกับ Mobile Application อื่นที่มีอยู่ในปัจจุบัน ตั้งแต่วันที่ 18 กุมภาพันธ์ ถึง 21 กุมภาพันธ์ เป็นเวลา 4 วัน ผู้รับผิดชอบคือ นางสาวเขมิสรา
- จัดทำ User Journey Map ตั้งแต่วันที่ 20 กุมภาพันธ์ ถึง 25 กุมภาพันธ์ เป็นเวลา 6 วัน ผู้รับผิดชอบคือ นางสาวธัญญารัตน์และนางสาวเขมิสรา
- จัดทำ User flow prototype ตั้งแต่วันที่ 28 มกราคม ถึง 26 กุมภาพันธ์ เป็นเวลา 30 วัน ผู้รับผิดชอบคือ นางสาวธัญญารัตน์
- จัดทำรายงานโครงการระยะที่ 1 ตั้งแต่วันที่ 26 กุมภาพันธ์ ถึง 28 กุมภาพันธ์ เป็นเวลา 3 วัน ผู้รับผิดชอบคือ นางสาวธัญญารัตน์และนางสาวเขมิสรา

และมีแผนการดำเนินงานโครงการในระยะที่ 2 เพิ่มเติม ดังนี้

📅 แผนงานโครงการ	👤 Assign	📅 Date	📊 Status
วางแผนงานโครงการและแบ่งหน้าที่	K KEMISARA A. 🧑 THANYARAT WUTHIROONGREUNGSAKI	January 20, 2025 → January 23, 2025	🟢 Done
รายละเอียดของโครงการ ที่มาและความสำคัญ	K KEMISARA A.	January 27, 2025 → February 3, 2025	🟢 Done
User flow prototype	🧑 THANYARAT WUTHIROONGREUNGSAKUL	January 28, 2025 → February 26, 2025	🟢 Done
รายละเอียดของโครงการ วัตถุประสงค์ของ Mobile Application	K KEMISARA A.	February 4, 2025 → February 6, 2025	🟢 Done
รายละเอียดของโครงการ กลุ่มเป้าหมายของ Mobile Application	K KEMISARA A. 🧑 THANYARAT WUTHIROONGREUNGSAKI	February 10, 2025 → February 19, 2025	🟢 Done
รายละเอียดของโครงการ ศึกษาเปรียบเทียบกับ Mobile Application อื่นที่มีอยู่ในปัจจุบัน	K KEMISARA A.	February 18, 2025 → February 21, 2025	🟢 Done
User Journey Map	K KEMISARA A. 🧑 THANYARAT WUTHIROONGREUNGSAKI	February 20, 2025 → February 25, 2025	🟢 Done
รายงานโครงการ phase 1	K KEMISARA A. 🧑 THANYARAT WUTHIROONGREUNGSAKI	February 26, 2025 → February 28, 2025	🟢 Done
เขียนโค้ดแอปพลิเคชัน	K KEMISARA A. 🧑 THANYARAT WUTHIROONGREUNGSAKI	March 21, 2025 → April 18, 2025	🟢 Done
ถ่ายคลิปวิดีโอการใช้งานแอปพลิเคชัน <input type="button" value="📄 OPEN"/>	🧑 THANYARAT WUTHIROONGREUNGSAKUL	April 18, 2025	🟢 Done
ทดลองการใช้งานแอปพลิเคชันกับกลุ่มเป้าหมาย	🧑 THANYARAT WUTHIROONGREUNGSAKUL	April 18, 2025	🟢 Done
จัดทำรายงานฉบับสมบูรณ์	K KEMISARA A. 🧑 THANYARAT WUTHIROONGREUNGSAKI	April 18, 2025 → April 20, 2025	🟢 Done

(รูปภาพที่ 31 : แผนงานระยะที่ 2 ของโครงการในรูปแบบ Table)



(รูปภาพที่ 32 : แผนงานระยะที่ 2 ของโครงการในรูปแบบ Timeline)

- เขียน code แอปพลิเคชัน ตั้งแต่วันที่ 21 มีนาคม ถึง 18 เมษายน เป็นเวลา 29 วัน ผู้รับผิดชอบ คือ นางสาวธัญญารัตน์และนางสาวเขมสิรา
- ถ่ายคลิปวิดีโอการใช้งานแอปพลิเคชัน ในวันที่ 18 เมษายน เป็นเวลา 1 วัน ผู้รับผิดชอบ คือ นางสาวธัญญารัตน์
- ทดลองการใช้งานแอปพลิเคชันกับกลุ่มเป้าหมาย ในวันที่ 18 เมษายน เป็นเวลา 1 วัน ผู้รับผิดชอบคือ นางสาวธัญญารัตน์
- จัดทำรายงานฉบับสมบูรณ์ ตั้งแต่วันที่ 18 เมษายน ถึง 20 เมษายน เป็นเวลา 2 วัน ผู้รับผิดชอบ คือ นางสาวธัญญารัตน์และนางสาวเขมสิรา

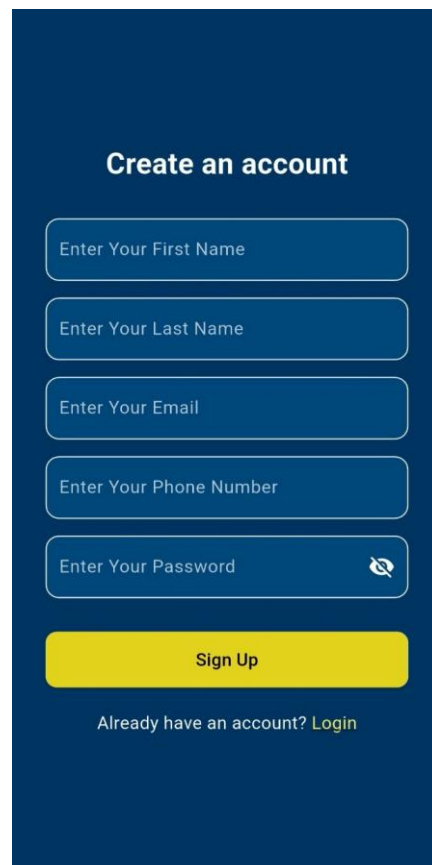
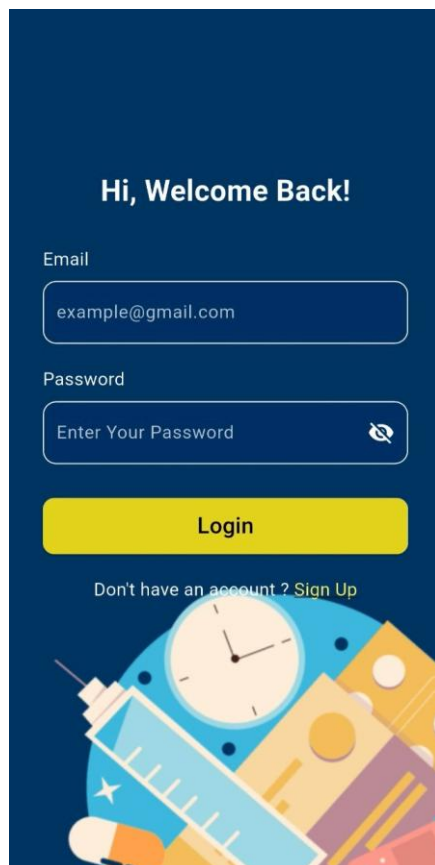
บทที่ 5

การพัฒนา Mobile Application

จากการศึกษาและพัฒนาแอปพลิเคชัน ผู้จัดทำขอนำเสนอผลการพัฒนาและตัวอย่าง Source Code ที่น่าสนใจ ดังนี้

1. Login page and Create account

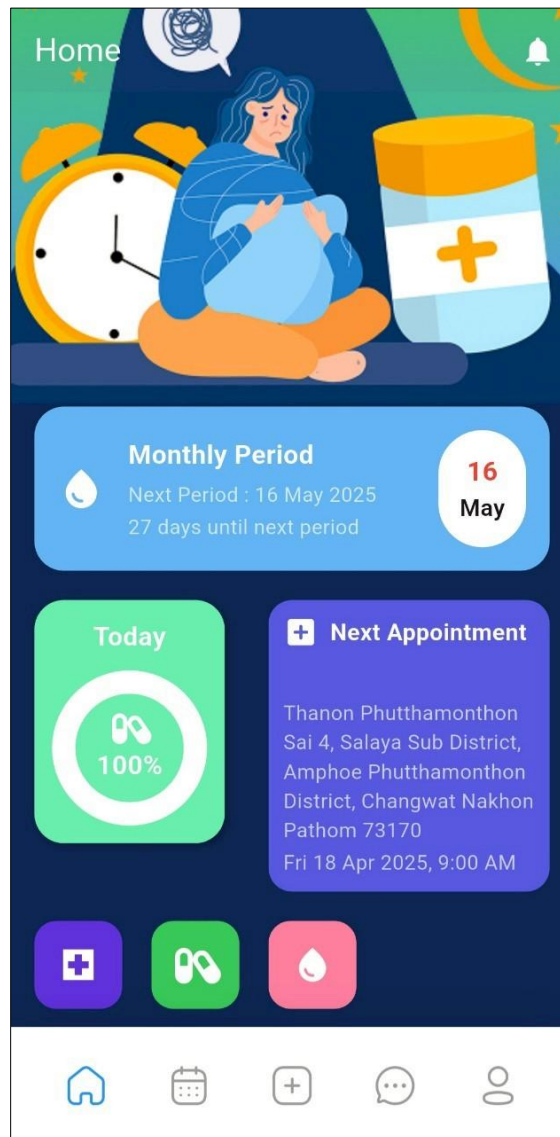
ในกรณีที่ผู้ใช้มีบัญชีอยู่แล้วสามารถ login ด้วย email และรหัสผ่าน หาก email และรหัสผ่านถูกต้องตรงตามฐานข้อมูล จะสามารถเข้าสู่ระบบได้และแสดงผลหน้า Home แต่หากมี email หรือรหัสผิด จะไม่สามารถ login ได้ ในกรณีที่ผู้ใช้ยังไม่มีบัญชีสามารถกด sign-up ด้านล่างกล่องข้อความรหัสผ่าน เพื่อทำการสร้างบัญชีและบันทึกข้อมูลลงฐานข้อมูลได้ โดยหน้า Sign- up จะเก็บข้อมูล ชื่อ นามสกุล email เบอร์โทรศัพท์ และรหัสผ่าน หากข้อมูลครบถ้วนและกดปุ่ม sign- up ระบบจะบันทึกข้อมูลลงฐานข้อมูล และกลับไปยังหน้า login เพื่อให้ผู้ใช้สามารถ login ได้ทันที



(รูปภาพที่ 33 : login page จากการพัฒนา) (รูปภาพที่ 34 : Create account จากการพัฒนา)

2. Home page

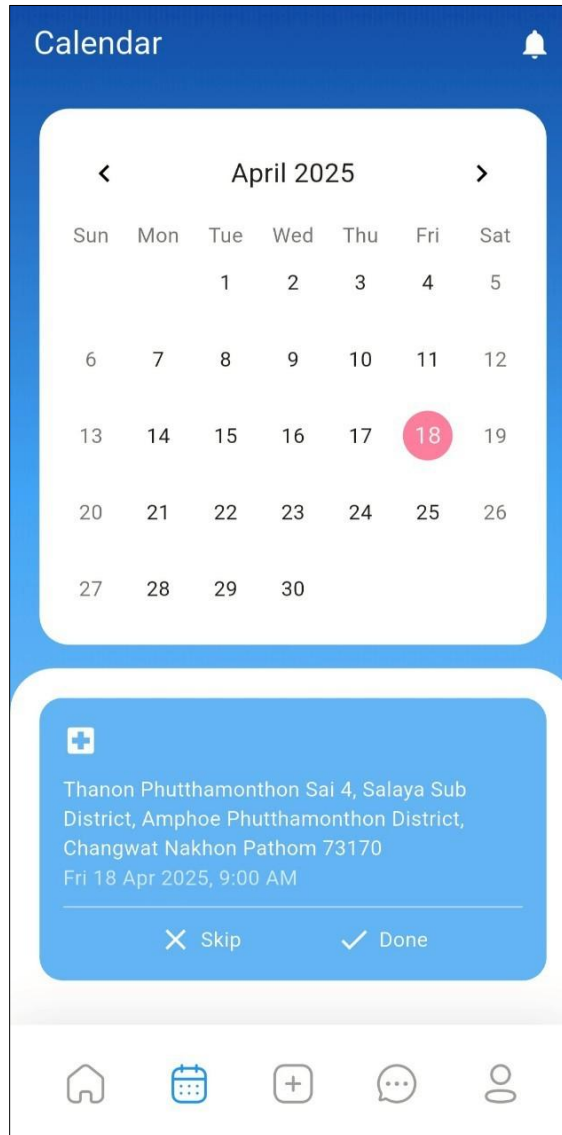
แสดงผลการติดตามประจำเดือนซึ่งจะแสดงวันที่ประจำเดือนมาในรอบถัดไป การกินยา ประจำวันในรูปแบบ piechart และนัดหมายจากโรงพยาบาลโดยแสดง ชื่อการนัดหมาย สถานที่ วันและ เวลานั้น โดยดึงข้อมูลจากฐานข้อมูล และส่วนด้านล่างสุดเป็นช่องทางลัดที่เชื่อมไปยังหน้าเพิ่มนัดหมาย จากโรงพยาบาล หน้าเพิ่มการกินยา และหน้าติดตามประจำเดือน ตามลำดับ



(รูปภาพที่ 35 : Home page จากการพัฒนา)

3. Calendar page

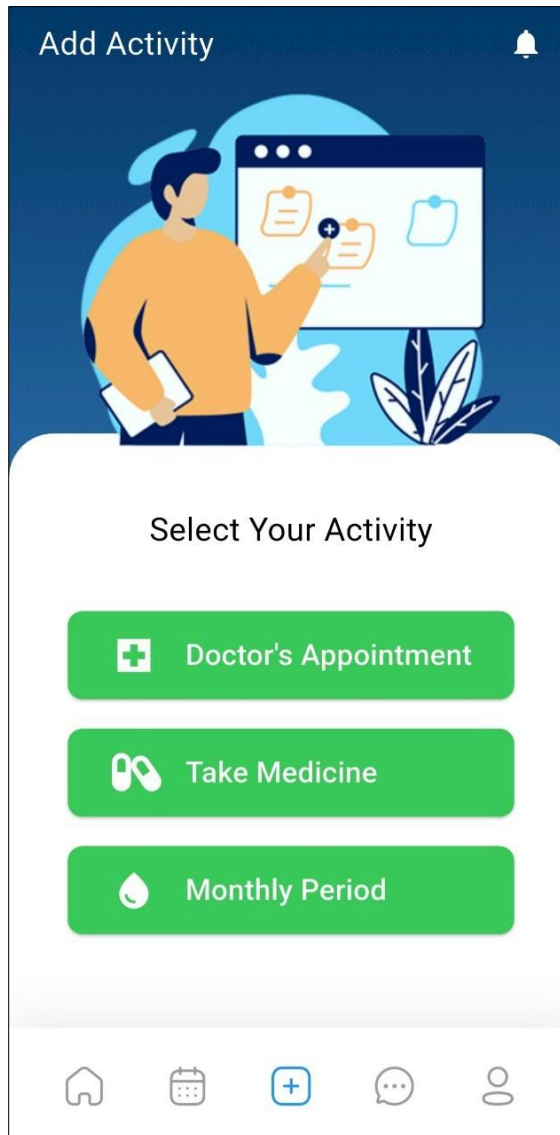
ปฏิทินสำหรับแสดงภาพรวมในเดือนนั้นๆ โดยจะแสดงนัดหมายและการกินยาในแต่ละวัน ในส่วนการกินยาสามารถกด Done หรือ Skip ได้ในหน้านี้ซึ่งจะไปแสดงผลที่ piechart ในหน้า Home



(รูปภาพที่ 36 : Calendar page จากการพัฒนา)

4. Select Activity page

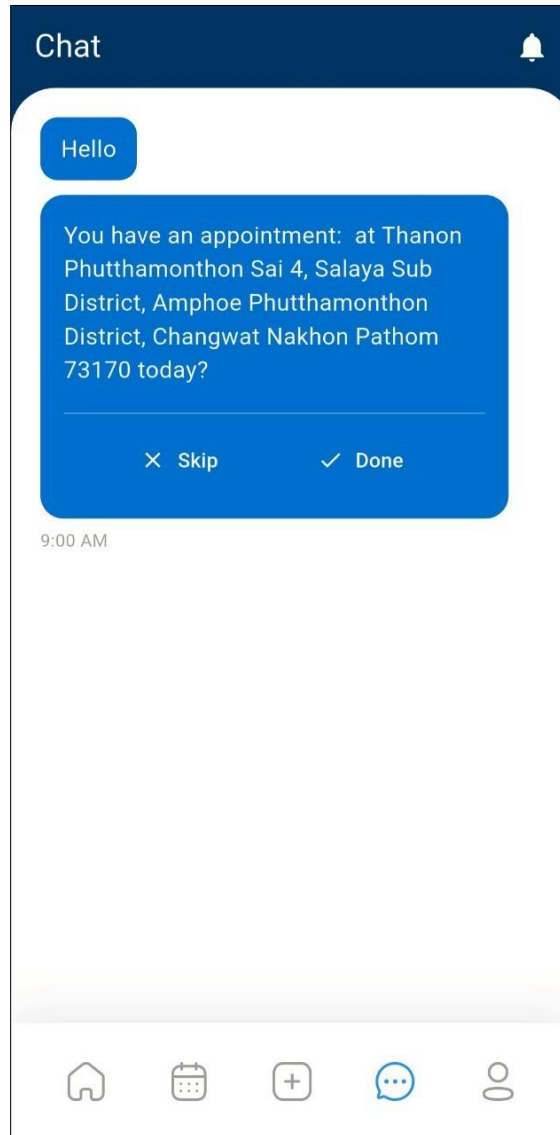
หน้าสำหรับเลือกสิ่งที่ต้องการจะติดตาม ได้แก่ การเพิ่มนัดหมายจากโรงพยาบาล การเพิ่มการกินยา และการติดตามประจำเดือน ซึ่งจะนำไปยังหน้าเพิ่มนัดหมายจากโรงพยาบาล หน้าเพิ่มการกินยา และหน้าติดตามประจำเดือน ตามลำดับ



(รูปภาพที่ 37 : Select Activity page จากการพัฒนา)

5. Chat

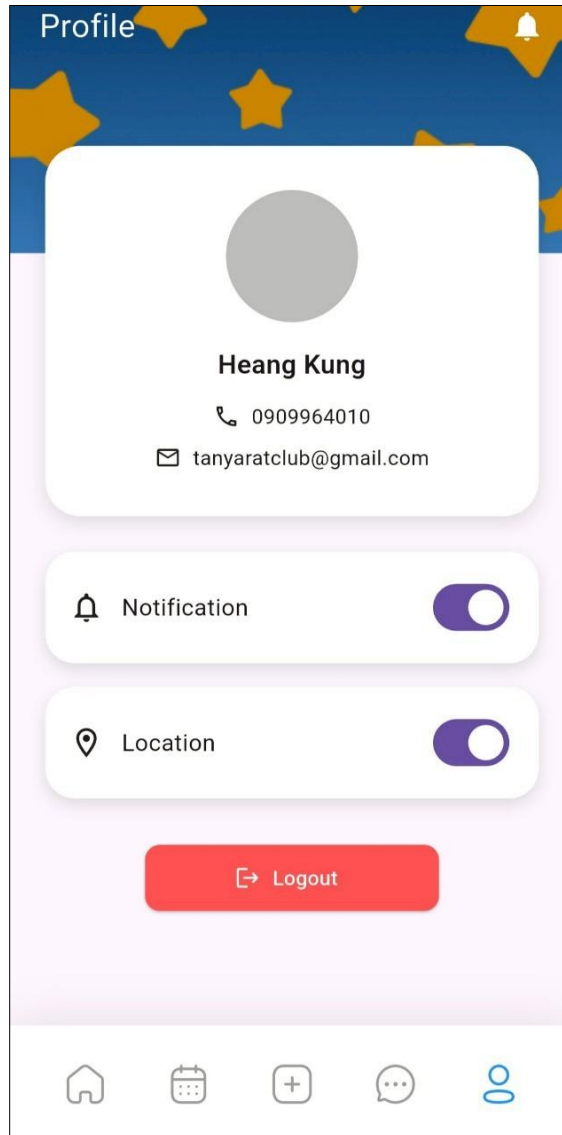
หน้าการแจ้งเตือนอัตโนมัติ โดยถ้าหากเรามีนัดหมายหรือต้องกินยา Chat จะทำการแจ้งเตือนอัตโนมัติเพื่อเตือนความจำ



(รูปภาพที่ 38 : Chat จากการพัฒนา)

6. Profile page

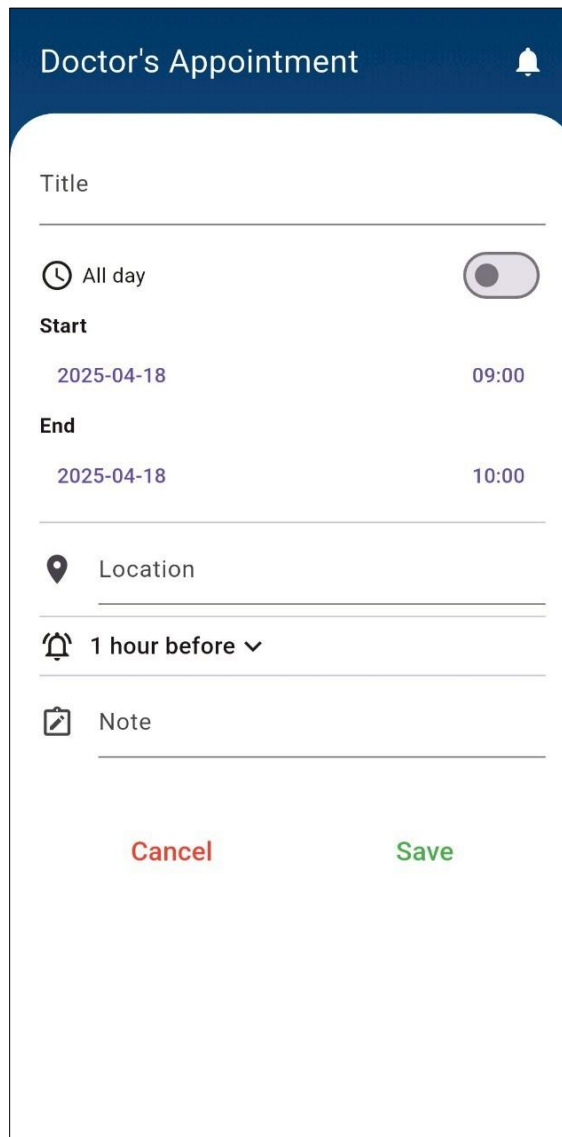
แสดงข้อมูลของ user ได้แก่ ชื่อ นามสกุล เบอร์โทรศัพท์ email มีส่วนในการเปิด-ปิด notification และการเข้าถึง location และส่วนสุดท้ายคือ ปุ่ม log-out โดยเมื่อกดจะนำไปสู่หน้า login



(รูปภาพที่ 39 : Select Activity page จากการพัฒนา)

7. Doctor's Appointment page

สำหรับกรอกข้อมูลการนัดหมายจากโรงพยาบาล ผู้ใช้สามารถใส่ข้อมูลการนัดหมายจากโรงพยาบาลได้ในหน้านี้ โดยต้องใส่ข้อมูลได้แก่ ชื่อการนัดหมาย (Title) วันและเวลาดนัดหมาย โดยสามารถเลือกเป็นทั้งวันได้ สถานที่ การแจ้งเตือน และโน้ตซึ่งจะใส่หรือไม่ก็ได้ เมื่อผู้ใช้ใส่ข้อมูลครบถ้วนและกดปุ่ม Save ระบบจะทำการบันทึกข้อมูลลงฐานข้อมูล และกลับไปยังหน้าก่อนหน้า



The screenshot shows a mobile application interface for creating a doctor's appointment. The title bar is dark blue with the text "Doctor's Appointment" and a bell icon. The form is white with rounded corners and contains the following fields and controls:

- Title:** A text input field.
- All day:** A toggle switch, currently turned off.
- Start:** A date and time selection area showing "2025-04-18" and "09:00".
- End:** A date and time selection area showing "2025-04-18" and "10:00".
- Location:** A text input field with a location pin icon.
- Notification:** A bell icon followed by the text "1 hour before" and a dropdown arrow.
- Note:** A text input field with a notepad icon.
- Buttons:** At the bottom, there are two buttons: "Cancel" in red and "Save" in green.

(รูปภาพที่ 40 : Doctor's Appointment page จากการพัฒนา)

8. Take Medicine page

สำหรับกรอกข้อมูลการกินยา ผู้ใช้สามารถใส่ข้อมูลยาที่ต้องรับประทานได้ในหน้านี้ โดยต้องใส่ข้อมูลได้แก่ ชื่อยา (Title) จำนวนยาและจำนวนยาที่ต้องทานต่อวัน ช่วงเวลาที่ต้องทานยา มีอาหารเวลาการแจ้งเตือน และโน้ตซึ่งจะใส่หรือไม่ก็ได้ เมื่อผู้ใช้ใส่ข้อมูลครบถ้วน และกดปุ่ม Save ระบบจะทำการบันทึกข้อมูลลงฐานข้อมูล และกลับไปยังหน้าก่อนหน้า

Take Medicine

Add Your Medicine

Amount (Tablet)
10

Times Per Day
2

Breakfast Lunch Dinner Bedtime

☐ Before meal ☐ After meal

1 hour before ▾

Note

Cancel Save

(รูปภาพที่ 41 : Take Medicine page จากการพัฒนา)

9. Monthly Period page

สำหรับติดตามประจำเดือน ผู้ใช้สามารถใส่ข้อมูลเพื่อติดตามปริมาณประจำเดือน อารมณ์ อาการอื่นๆ และรูปแบบการมีเพศสัมพันธ์ ได้ในหน้านี้ โดยจะเป็นตัวเลือกในแต่ละหัวข้อ เมื่อผู้ใช้เลือกข้อมูลครบถ้วน และกดปุ่ม Save ระบบจะทำการบันทึกข้อมูลลงฐานข้อมูล และกลับไปยังหน้าก่อนหน้า

Add Period

Friday, 18 Apr 2025

15 16 17 **18** 19 20 21

Volume

Heavy Medium Light Spotting

Mood

Happy Energetic Calm Swing Bad

Symptoms

Fine Cramps Headache Acne

Sex Drive

Didn't Have Sex Masturbation

Protected Sex Unprotected Sex

Cancel Save

(รูปภาพที่ 42 : Monthly Period page จากการพัฒนา)

ตัวอย่าง Source Code ที่น่าสนใจ

- ตัวอย่างที่ 1 : การเชื่อมต่อและเรียกใช้งาน API Location (TomTom)
 - การเชื่อมต่อ API TomTom (lib/service/api_map.dart)

Code:
<pre>//----- Import -----// import 'dart:convert'; import 'package:http/http.dart' as http; //----- การเชื่อม API TomTom -----// class TomTomService { static const String _apiKey = 'SqRm4x62Fb7oJ7l8Wj0dJfdbRgAAOu5a'; // API Key static const String _baseUrl = 'https://api.tomtom.com/search/2/search'; // URL API static Future<List<String>> searchLocations(String query) async { final url = Uri.parse('\${_baseUrl}/\${query}.json?key=\$_apiKey&limit=5'); final response = await http.get(url); if (response.statusCode == 200) { final data = json.decode(response.body); final results = data['results'] as List<dynamic>; return results.map((e) => e['address']['freeformAddress'] as String).toList(); } else { throw Exception('Failed to load locations from TomTom'); } } }</pre>
Description:
สำหรับการเชื่อมต่อ API TomTom ขั้นแรกจะต้อง import 2 อย่าง คือ 1) dart:convert ใช้สำหรับแปลงข้อมูล JSON จาก API ให้เป็น Map หรือ List ที่ Dart ใช้งานได้ 2) ใช้ในการส่ง request ไปยัง API ภายนอก (เช่น GET, POST)

ต่อมาใน class TomTomService ต้องทำการใส่ Token ที่ได้จากเว็บไซต์ของ TomTom โดยเก็บไว้ในตัวแปร `_apiKey` และใส่ URL สำหรับการค้นหาสถานที่ของ TomTom โดยเก็บไว้ในตัวแปร `_baseUrl`

ใน function หลัก `searchLocations()` ทำการสร้าง URL โดยใช้ตัวแปร `url` และใช้ `http.get()` เพื่อส่ง request ไปยัง URL และสร้างเงื่อนไขเพื่อตรวจสอบสถานะการตอบกลับ โดยถ้าหาก API ตอบกลับสำเร็จ จะทำการแปลงผลลัพธ์ JSON เป็น Dart Object เก็บไว้ในตัวแปร `data` และดึงค่าจาก key "results" ของ `data` ซึ่งเป็น list ของผลลัพธ์สถานที่ วนค่าใน `results` เพื่อหาและดึงค่า `address` ในรูปแบบที่จัดมาแล้วและแปลงทั้งหมดให้เป็นลิสต์ของ String แต่ถ้าหาก API ตอบกลับไม่สำเร็จ จะแสดงผล 'Failed to load locations from TomTom'

— การเรียกใช้ใน Doctor's Appointment page (lib/pages/doctor.dart)

Code:

```
//----- Import -----//
. . .

import '../service/api_map.dart';
//----- ส่วนกำหนดตัวแปรของ Doctor's Appointment page -----//
. . .

//----- API Location -----//

@override
void initState() {
  super.initState();
  locationController.addListener() async {
    final query = locationController.text;
    if (query.length >= 3) {
      try {
        final results = await TomTomService.searchLocations(query);
        setState(() {
          locationSuggestions = results;
        });
      } catch (e) {
        print("Error fetching locations: \$e");
      }
    }
  }
}
```

<pre> } }); } . . . </pre>
<p>Description:</p> <p>สำหรับการเรียกใช้ API TomTom ใน Doctor's Appointment page ขั้นแรกจะต้อง import => <code>'../service/api_map.dart';</code> ก่อน</p> <p>Function <code>initState()</code> เมื่อหน้าเริ่มโหลด จะตั้งค่า listener ให้ <code>locationController</code> เพื่อเรียก API หา location suggestions เมื่อพิมพ์ใน location text field 3 ตัวอักษรขึ้นไป และมีการใช้ <code>try/catch</code> เพื่อตรวจจับการเชื่อมต่อ API โดยถ้าหากเชื่อมต่อ API ได้ จะเรียก API เก็บไว้ในตัวแปร <code>results</code> และจะ set state ให้ <code>locationSuggestions = results</code> แต่หากเชื่อมต่อไม่ได้จะแสดงผล "Error fetching locations: <code>\\$e</code> (รายละเอียดของ Error)"</p>

- ตัวอย่างที่ 2 : การบันทึกข้อมูลของ Doctor's Appointment page ลงในฐานข้อมูล
(lib/pages/doctor.dart)

<p>Code:</p> <pre> //----- Import -----// . . . //----- Stateful Widget -----// class DoctorAppointmentPage extends StatefulWidget { @override _DoctorAppointmentPageState createState() => _DoctorAppointmentPageState(); } //----- ส่วนที่ 1 กำหนดตัวแปร -----// class _DoctorAppointmentPageState extends State<DoctorAppointmentPage> { TextEditingController titleController = TextEditingController(); TextEditingController locationController = TextEditingController(); TextEditingController noteController = TextEditingController(); List<String> locationSuggestions = []; </pre>
--

```

bool isAllDay = false;
TimeOfDay startTime = TimeOfDay(hour: 9, minute: 0);
TimeOfDay endTime = TimeOfDay(hour: 10, minute: 0);
DateTime startDate = DateTime.now();
DateTime endDate = DateTime.now();
String reminder = "1 hour before";

//----- API Location -----//

. . .

//----- ส่วนที่ 2 function เลือกวันและเวลา -----//

void _pickTime({required bool isStart}) async {
  final picked = await showTimePicker(
    context: context,
    initialTime: isStart ? startTime : endTime,
  );
  if (picked != null) {
    setState() {
      if (isStart) {
        startTime = picked;
      } else {
        endTime = picked;
      }
    };
  }
}

void _pickDate({required bool isStart}) async {
  final picked = await showDatePicker(
    context: context,
    initialDate: isStart ? startDate : endDate,
    firstDate: DateTime(2020),
    lastDate: DateTime(2100),
  );
}

```

```

);
if (picked != null) {
    setState(() {
        if (isStart) {
            startDate = picked;
        } else {
            endDate = picked;
        }
    });
}
}

//----- ส่วนที่ 3 function บันทึกข้อมูล -----//
void _onSave() async {
    final prefs = await SharedPreferences.getInstance();
    final userId = prefs.getInt('userId');
    if (userId == null) return;

    final startDT = DateTime(
        startDate.year,
        startDate.month,
        startDate.day,
        startTime.hour,
        startTime.minute,
    );
    final endDT = DateTime(
        endDate.year,
        endDate.month,
        endDate.day,
        endTime.hour,
        endTime.minute,

```

```

);
final appointment = AppointmentModel(
  title: titleController.text,
  location: locationController.text,
  note: noteController.text,
  startTime: startDT,
  endTime: endDT,
  isAllDay: isAllDay,
  reminder: reminder,
);
await DatabaseHelper().insertAppointment(appointment, userId);
ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(content: Text("Appointment saved")),
);
Navigator.pop(context);
}
//----- ส่วนที่ 4 ยกเลิกการใส่ข้อมูล -----//
void _onCancel() {
  Navigator.pop(context);
}
//----- UI -----//
@override
Widget build(BuildContext context) {
  . . .
//----- Widget ที่จำเป็นเพิ่มเติม -----//
  Widget _buildDateTimePicker(String label, bool isStart) {
    final date = isStart ? startDate : endDate;
    final time = isStart ? startTime : endTime;
    return Column(
      crossAxisAlignment: CrossAxisAlignment.start,

```

```

        children: [
          Align(
            alignment: Alignment.centerLeft,
            child: Text(label, style: TextStyle(fontWeight: FontWeight.bold)),
          ),
          Row(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [
              TextButton(
                onPressed: () => _pickDate(isStart: isStart),
                child: Text(DateFormat('y-MM-dd').format(date)),
              ),
              TextButton(
                onPressed: () => _pickTime(isStart: isStart),
                child: Text(time.format(context)),
              ),
            ],
          ),
        ],
      );
    }

    Widget _buildReminderPicker() {
      return Row(
        children: [
          Icon(Icons.notifications_active_outlined),
          SizedBox(width: 10),
          GestureDetector(
            onTap: () => _showReminderOptions(),
            child: Row(
              children: [

```

```

        Text(reminder, style: TextStyle(fontSize: 16, fontWeight: FontWeight.w500)),
        Icon(Icons.keyboard_arrow_down),
    ],
),
),
],
);
}

void _showReminderOptions() {
    showModalBottomSheet(
        context: context,
        shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.vertical(top: Radius.circular(20)),
        ),
        builder: (context) {
            final options = [
                "30 minutes before",
                "1 hour before",
                "2 hours before",
                "5 hours before",
                "12 hours before",
                "1 day before",
                "3 days before",
                "1 week before",
            ];
            return ListView(
                shrinkWrap: true,
                children: options.map((option) => ListTile(
                    title: Text(option),
                    trailing: reminder == option ? Icon(Icons.check, color: Colors.blue) : null,
                )),
            );
        },
    );
}

```

```

        onTap: () {
          setState(() => reminder = option);
          Navigator.pop(context);
        },
      ),
    ).toList(),
  );
  . . .
}

```

Description:

ภายใน class DoctorAppointmentPage เป็น Stateful Widget ซึ่งใช้ในการสร้างหน้า UI สำหรับการจองนัดหมายแพทย์ โดยมีการจัดการสถานะต่าง ๆ เช่น วัน เวลา สถานที่ ชื่อเรื่อง การเตือนความจำ ฯลฯ ผ่าน State ของ Widget แบ่งเป็น 5 ส่วนหลักๆ ได้แก่

ส่วนที่ 1 กำหนดตัวแปรต่างๆที่จำเป็นในการสร้างนัดหมาย : ภายใน class `_DoctorAppointmentPageState` จะมีการกำหนดตัวแปรควบคุม input ได้แก่ `titleController`, `locationController`, และ `noteController` เพื่อรับค่าจากผู้ใช้ รวมทั้งมีตัวแปร `locationSuggestions` สำหรับแนะนำตำแหน่ง, `isAllDay` สำหรับเก็บสถานะของการนัดหมายแบบทั้งวัน, `startTime` และ `endTime` สำหรับเวลาเริ่มต้น-สิ้นสุด, `startDate` และ `endDate` สำหรับวันเริ่มต้น-สิ้นสุด และ `reminder` เพื่อกำหนดการแจ้งเตือนก่อนถึงเวลานัดหมาย

ส่วนที่ 2 ฟังก์ชันเลือกวันและเวลา : ในส่วนนี้มีฟังก์ชัน `_pickTime()` และ `_pickDate()` สำหรับให้ผู้ใช้เลือกเวลาและวันของการนัดหมาย โดยใช้ `showTimePicker` และ `showDatePicker` ที่เป็น built-in widget ของ Flutter หากผู้ใช้เลือกค่าแล้วจะมีการอัปเดตค่าลงใน State ด้วย `setState()`

- o `_pickTime` ตรวจสอบว่าเป็นเวลาเริ่มต้นหรือสิ้นสุด และกำหนดค่า `startTime` หรือ `endTime` ตามนั้น
- o `_pickDate` ใช้หลักการเดียวกันกับ `_pickTime` ในการกำหนดค่า `startDate` หรือ `endDate`

ส่วนที่ 3 ฟังก์ชันบันทึกข้อมูลนัดหมาย : ในฟังก์ชัน `_onSave()` จะทำการดึง `userId` จาก `SharedPreferences` ซึ่งเป็นที่เก็บข้อมูลแบบ local บนอุปกรณ์ จากนั้นจะรวมค่าของวัน (`startDate`, `endDate`) และเวลา (`startTime`, `endTime`) เข้าด้วยกันเพื่อสร้าง `DateTime` ต่อมานำข้อมูลที่ผู้ใช้กรอก ได้แก่ ชื่อนัดหมาย สถานที่ หมายเหตุ และตัวเลือกการแจ้งเตือน มาสร้างเป็น `AppointmentModel` แล้วทำการบันทึกลงฐานข้อมูลด้วย `DatabaseHelper().insertAppointment()` หากบันทึกสำเร็จ จะมีข้อความ `SnackBar` แสดงว่า "Appointment saved" และกลับไปยังหน้าก่อนหน้าโดยใช้ `Navigator.pop(context)`

ส่วนที่ 4 ฟังก์ชันยกเลิกการกรอกข้อมูล : ในกรณีที่ผู้ใช้ต้องการยกเลิกการนัดหมายหรือไม่ต้องการบันทึกข้อมูล สามารถกดปุ่มยกเลิก ซึ่งจะเรียกใช้ _onCancel() เพื่อ pop() ออกจากหน้าจอปัจจุบัน

ส่วนที่ 5 การสร้าง UI ที่เกี่ยวข้อง : ภายใน build() จะมีการสร้าง UI ต่างๆ โดยใช้ฟังก์ชันย่อย ได้แก่

- o _buildDateTimePicker() : ฟังก์ชันนี้ใช้สร้าง UI สำหรับเลือกวันและเวลา โดยแสดงปุ่ม 2 ปุ่ม (วัน และ เวลา) ให้ผู้ใช้สามารถกดเพื่อเปลี่ยนค่า โดยจะดึงค่าจากตัวแปร startDate, endDate, startTime, endTime แล้วแสดงผลแบบฟอร์แมต เช่น 2025-04-19
- o _buildReminderPicker() : ใช้แสดงตัวเลือกการแจ้งเตือนก่อนนัดหมาย เช่น "1 hour before", "1 day before" เป็นต้น โดยใช้ GestureDetector คลิกเพื่อเปิด BottomSheet
- o _showReminderOptions() : เมื่อคลิกตัวเลือกการแจ้งเตือน จะเรียกใช้ฟังก์ชันนี้เพื่อแสดง BottomSheet ที่มีรายการตัวเลือกการเตือนล่วงหน้า เช่น 30 นาที, 1 ชั่วโมง, 1 วัน ฯลฯ เมื่อผู้ใช้เลือกค่าหนึ่ง ระบบจะอัปเดตค่า reminder และปิด BottomSheet โดยใช้ Navigator.pop(context)

- ตัวอย่างที่ 3 : การดึงข้อมูลจากฐานข้อมูลไปแสดงผลในหน้า Home page
(lib/widgets/homecontent.dart)

Code:

```
//----- Import -----//
. . .

import 'package:pie_chart/pie_chart.dart';
import 'package:intl/intl.dart';
import 'package:shared_preferences/shared_preferences.dart';
import '../models/appointment.dart';
import '../models/period.dart';
import '../db/database.dart';

//----- Stateful Widget -----//
class HomeContent extends StatefulWidget {
  @override
  _HomeContentState createState() => _HomeContentState();
}

//----- ส่วนที่ 1 กำหนดตัวแปรและโหลดข้อมูลเริ่มต้น -----//
```

```

class _HomeContentState extends State<HomeContent> {
  AppointmentModel? nextAppointment;
  DateTime? _lastPeriodDate;

  double progress = 1.0;
  int doneCount = 0;
  int totalCount = 0;

  int? userId;

  @override
  void initState() {
    super.initState();
    _loadUserData();
  }
  //----- ส่วนที่ 2 โหลดข้อมูลจาก Database -----//
  Future<void> _loadUserData() async {
    final prefs = await SharedPreferences.getInstance();
    final id = prefs.getInt('userId');
    final dateStr = prefs.getString('last_period_date');

    if (id == null) return;

    setState(() {
      userId = id;
      if (dateStr != null) {
        _lastPeriodDate = DateTime.parse(dateStr);
      }
    });
  }
}

```

```

    await _loadNextAppointment();
    await _loadTodayMedicineProgress();
}

Future<void> _loadNextAppointment() async {
    if (userId == null) return;
    final all = await DatabaseHelper().getAppointmentsByUser(userId!);
    final now = DateTime.now();
    final upcoming = all.where((a) => a.startTime.isAfter(now)).toList();
    upcoming.sort((a, b) => a.startTime.compareTo(b.startTime));
    setState() {
        nextAppointment = upcoming.isNotEmpty ? upcoming.first : null;
    });
}

Future<void> _loadTodayMedicineProgress() async {
    if (userId == null) return;

    final today = DateTime.now();
    final meds = await DatabaseHelper().getAllMedicines(userId!);
    final todayMeds = meds
        .where((m) =>
            m.date.year == today.year &&
            m.date.month == today.month &&
            m.date.day == today.day)
        .toList();

    totalCount = todayMeds.length;
    doneCount = todayMeds.where((m) => m.status == 'done').length;

    setState() {

```

```

        progress = (totalCount == 0) ? 1.0 : doneCount / totalCount;
    });
}

//----- UI -----//

@override
Widget build(BuildContext context) {
    Map<String, double> dataMap = {
        "Completed": (progress * 100).roundToDouble(),
        "Remaining": 100 - (progress * 100).roundToDouble(),
    };
    return Scaffold(
        . . .
    );
}

//----- Widget ที่จำเป็นเพิ่มเติม -----//

Widget _buildIconButton({
    required IconData icon,
    required Color color,
    required VoidCallback onTap,
}) {
    return GestureDetector(
        onTap: onTap,
        child: Container(
            width: 60,
            height: 60,
            decoration: BoxDecoration(
                color: color,
                borderRadius: BorderRadius.circular(15),
            ),
            child: Center(child: Icon(icon, color: Colors.white, size: 28)),
        ),
    );
}

```

```

    ),
  );
}

Widget _buildPeriodCard() {
  final next = _lastPeriodDate!.add(Duration(days: 28));
  return Container(
    padding: EdgeInsets.all(16),
    decoration: BoxDecoration(
      color: Color(0xFF64B5F6),
      borderRadius: BorderRadius.circular(20),
    ),
    child: Row(
      children: [
        Icon(Icons.water_drop, color: Colors.white, size: 32),
        SizedBox(width: 16),
        Expanded(
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                "Monthly Period",
                style: TextStyle(
                  fontSize: 18,
                  fontWeight: FontWeight.bold,
                  color: Colors.white,
                ),
              ),
              SizedBox(height: 4),
              Text(
                "Next Period : ${DateFormat('d MMM yyyy').format(next)}",

```

```

        style: TextStyle(color: Colors.white70),
      ),
      SizedBox(height: 2),
      Text(
        "${_daysUntilNextPeriod()} days until next period",
        style: TextStyle(color: Colors.white70),
      ),
    ],
  ),
),
Container(
  width: 60,
  height: 80,
  decoration: BoxDecoration(
    color: Colors.white,
    borderRadius: BorderRadius.circular(30),
  ),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Text(
        DateFormat('dd').format(next),
        style: TextStyle(
          fontSize: 18,
          color: Colors.red,
          fontWeight: FontWeight.bold,
        ),
      ),
      Text(
        DateFormat('MMM').format(next),

```

```

        style: TextStyle(fontSize: 16, fontWeight: FontWeight.w600),
      ),
    ],
  ),
),
],
),
);
}

Widget _buildProgressCard(Map<String, double> dataMap) {
  return Container(
    width: 130,
    padding: EdgeInsets.all(12),
    decoration: BoxDecoration(
      color: Colors.greenAccent,
      borderRadius: BorderRadius.circular(20),
      boxShadow: [
        BoxShadow(color: Colors.black26, blurRadius: 4, offset: Offset(2, 2)),
      ],
    ),
    child: Column(
      children: [
        Text(
          "Today",
          style: TextStyle(
            color: Colors.white,
            fontSize: 18,
            fontWeight: FontWeight.bold,
          ),
        ),
      ],
    ),
  );
}

```

```

    SizedBox(height: 10),
    Stack(
      alignment: Alignment.center,
      children: [
        PieChart(
          dataMap: dataMap,
          chartType: ChartType.ring,
          ringStrokeWidth: 16,
          colorList: [Colors.white, Colors.white.withOpacity(0.3)],
          legendOptions: LegendOptions(showLegends: false),
          chartValuesOptions: ChartValuesOptions(showChartValues: false),
        ),
        Column(
          children: [
            Icon(
              FontAwesomeIcons.capsules,
              color: Colors.white,
              size: 24,
            ),
            Text(
              '${(progress * 100).round()}%',
              style: TextStyle(
                fontSize: 18,
                color: Colors.white,
                fontWeight: FontWeight.bold,
              ),
            ),
          ],
        ),
      ],
    ),
  ],

```



```

    ),
  ],
),
);
}
Widget _buildAppointmentCard() {
  return Container(
    padding: EdgeInsets.all(10),
    decoration: BoxDecoration(
      color: Color(0xFF585AE2),
      borderRadius: BorderRadius.circular(16),
    ),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Row(
          children: [
            Icon(Icons.add_box_rounded, color: Colors.white),
            SizedBox(width: 8),
            Text(
              "Next Appointment",
              style: TextStyle(
                color: Colors.white,
                fontSize: 16,
                fontWeight: FontWeight.bold,
              ),
            ),
          ],
        ),
        SizedBox(height: 10),
      ],
    ),
  );
}

```

```

        if (nextAppointment != null) ...[
            Text(
                nextAppointment!.title,
                style: TextStyle(color: Colors.white, fontSize: 14),
            ),
            SizedBox(height: 2),
            Text(
                nextAppointment!.location,
                style: TextStyle(color: Colors.white70),
            ),
            SizedBox(height: 2),
            Text(
                DateFormat('EEE d MMM yyyy, h:mm a')
                    .format(nextAppointment!.startTime),
                style: TextStyle(color: Colors.white70),
            ),
        ] else
            Text(
                "No upcoming appointments",
                style: TextStyle(color: Colors.white70),
            ),
    ],
),
);
}

int _daysUntilNextPeriod() {
    final next = _lastPeriodDate!.add(Duration(days: 28));
    return next.difference(DateTime.now()).inDays;
}
}

```

Description:

ในส่วนของการ import และ create state มีการเรียกใช้งาน Package และกำหนดคลาสหลักของหน้า Home ในส่วนนี้มีการ import ไฟล์ต่างๆ ที่จำเป็น เช่น ไฟล์หน้าอื่นๆ (doctor.dart, AddMonthlyPeriod.dart), model (appointment.dart, period.dart), และ database (database.dart) รวมถึง package พื้นฐานของ Flutter เช่น Material, SharedPreferences, และ intl สำหรับการจัดการวันที่ จากนั้นสร้าง StatefulWidget ชื่อ HomeContent ซึ่งเป็นหน้าหลักที่แสดงข้อมูลสุขภาพ

ส่วนที่ 1 กำหนดตัวแปรและโหลดข้อมูลเริ่มต้น : ในคลาส _HomeContentState มีการกำหนดตัวแปรหลัก เช่น nextAppointment เก็บนัดหมายถัดไป, _lastPeriodDate วันที่มีประจำเดือนครั้งล่าสุด, progress, doneCount, totalCount ใช้คำนวณเปอร์เซ็นต์การกินยาของวันนี้, userId รหัสผู้ใช้ที่โหลดจาก SharedPreferences และใน initState() มีการเรียก loadUserData() เพื่อโหลดข้อมูลผู้ใช้, ข้อมูลนัดหมายถัดไป และสถานะการกินยารวันนี้

ส่วนที่ 2 โหลดข้อมูลจาก Database : มี 3 ฟังก์ชันที่ใช้สำหรับโหลดข้อมูลจาก Database ได้แก่

- ฟังก์ชัน _loadUserData() โหลด userId และ last_period_date จาก SharedPreferences และแปลง last_period_date เป็น DateTime เพื่อใช้แสดงผล เมื่อโหลดเสร็จจะเรียกโหลดนัดหมายและข้อมูลยาเพิ่มเติม
- ฟังก์ชัน _loadNextAppointment() โหลดนัดหมายทั้งหมดของผู้ใช้จากฐานข้อมูล และหาวันที่อยู่หลังเวลาปัจจุบัน แล้วเรียงลำดับเวลา เก็บนัดหมายถัดไปใน nextAppointment
- ฟังก์ชัน _loadTodayMedicineProgress() โหลดรายการยาทั้งหมดของผู้ใช้ แล้วกรองเฉพาะยาที่ตรงกับวันที่ปัจจุบัน คำนวณจำนวนยาทั้งหมดและจำนวนยาที่กินแล้ว เพื่อแสดงผลใน Pie Chart

ส่วนที่ 3 ฟังก์ชันย่อยที่ใช้สร้าง Widget ต่างๆ ได้แก่

- _buildIconButton() : สร้างปุ่มไอคอนที่ใช้ใน Row ล่างสุด สำหรับลัดไปยังหน้าต่าง ๆ
- _buildPeriodCard() : สร้างกล่องสี่เหลี่ยมแสดงวันมีประจำเดือนรอบถัดไป และจำนวนวันที่เหลือ
- _buildProgressCard() : แสดง Pie Chart สำหรับเปอร์เซ็นต์การกินยาของวันนี้
- _buildAppointmentCard() : แสดงข้อมูลนัดหมายล่าสุดที่กำลังจะถึง
- _daysUntilNextPeriod() : คำนวณจำนวนวันจนถึงรอบประจำเดือนถัดไป โดยใช้ Duration(days: 28) จากวันล่าสุด

บทที่ 6

การทดสอบ Mobile Application

จากที่ผู้จัดทำได้ทำการศึกษาและพัฒนาแอปพลิเคชัน Healthacker สำหรับติดตามสุขภาพขึ้น ผู้จัดทำได้มีการทดสอบและสร้าง Test-Case ขึ้น อีกทั้งยังนำแอปพลิเคชันที่พัฒนาขึ้นไปทดสอบกับกลุ่มเป้าหมายอีกด้วย โดยมีการทดสอบจริงจากกลุ่มเป้าหมายจำนวน 5 คน ผู้จัดทำจึงขอนำเสนอ Test-Case ที่สร้างขึ้นและผลการทดสอบจากกลุ่มเป้าหมาย ดังนี้

1. Test-Case

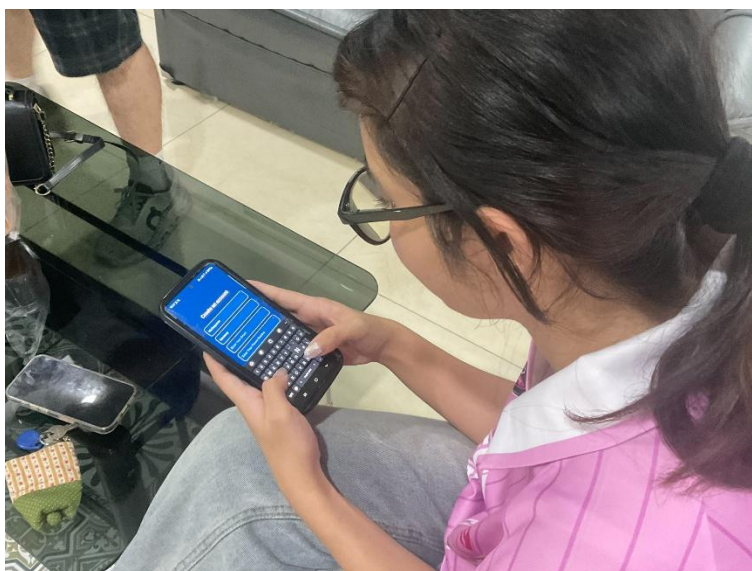
No.	Text Procedure	Expected Result	Text Result (Pass/Fail)	Remask
1.	กรอก username & password ที่ถูกต้อง	แสดงหน้า home	Pass	
2.	กรอก username & password ที่ไม่ถูกต้อง	แสดงข้อความว่า username or password is incorrect ในหน้า login	Pass	
3.	กรอกข้อมูล sign up ใหม่โดยที่ถูกต้องทั้งหมด	สามารถบันทึกข้อมูลลงฐานข้อมูล และกลับไปหน้า login	Pass	
4.	กรอกข้อมูล sign up ใหม่โดยที่ email ซ้ำกับในฐานข้อมูล	ไม่สามารถ sign up ได้ และแสดงข้อความ Email already exists ในหน้า sign up	Pass	
5.	กรอกข้อมูล Title วันเวลานัดสถานที่ หน้า Doctor ครบถ้วน และกดบันทึกข้อมูล	บันทึกข้อมูลสำเร็จและกลับมายังหน้าก่อนหน้า	Pass	
6.	กดยกเลิกในหน้า Doctor	ไม่บันทึกข้อมูลและกลับมายังหน้าก่อนหน้า	Pass	

7.	กรอกข้อมูล Title จำนวนยา เวลาที่ต้องทานยา ในหน้า Medicine ครบถ้วนและกด บันทึกข้อมูล	บันทึกข้อมูลสำเร็จและกลับมา ยังหน้าก่อนหน้า	Pass	
8.	กดยกเลิกในหน้า Medicine	ไม่บันทึกข้อมูลและกลับมา ยังหน้าก่อนหน้า	Pass	
9.	กรอกข้อมูลในหน้า Period ครบถ้วนและกดบันทึกข้อมูล	บันทึกข้อมูลสำเร็จและกลับมา ยังหน้าก่อนหน้า	Pass	
10.	กดยกเลิกในหน้า Period	ไม่บันทึกข้อมูลและกลับมา ยังหน้าก่อนหน้า	Pass	
11.	เลือกวันในหน้า Calendar	แสดงข้อมูลที่เคยบันทึกไว้จาก หน้า Doctor, Medicine	Pass	
12.	กดเปลี่ยนเดือน	เปลี่ยนเดือนของปฏิทิน	Pass	
13.	กดปุ่ม Done ในการแจ้งเตือน การกินยา ในหน้า Calendar	Piechart การกินยาประจำวัน ในหน้า Home เพิ่มเปอร์เซ็นต์	Pass	
14.	กดปุ่ม Skip ในการแจ้งเตือน การกินยา ในหน้า Calendar	Piechart การกินยาประจำวัน ในหน้า Home ไม่เพิ่ม เปอร์เซ็นต์	Pass	
15.	กดปุ่มไปยังหน้า home	แสดงข้อมูลประจำเดือนครั้ง ถัดไป การกินยาประจำวัน นัด หมายครั้งถัดไป	Pass	
16.	กดปุ่มทางลัดไปยังหน้า Doctor	แสดงหน้า Doctor	Pass	
17.	กดปุ่มทางลัดไปยังหน้า Medicine	แสดงหน้า Medicine	Pass	
18.	กดปุ่มทางลัดไปยังหน้า Period	แสดงหน้า Period	Pass	
19.	กดปุ่ม Done ในการแจ้งเตือน การกินยา ในหน้า Chat	Piechart การกินยาประจำวัน ในหน้า Home เพิ่มเปอร์เซ็นต์	Pass	
20.	กดปุ่ม Skip ในการแจ้งเตือน การกินยา ในหน้า Chat	Piechart การกินยาประจำวัน ในหน้า Home ไม่เพิ่ม เปอร์เซ็นต์	Pass	

21.	เปิด-ปิด Notification ในหน้า profile	ปิด-เปิดการแจ้งเตือนของแอปพลิเคชัน	Fail	ไม่สามารถปิด-เปิดการแจ้งเตือนของแอปพลิเคชัน
22.	เปิด-ปิด location ในหน้า profile	ปิด-เปิดการเข้าถึงตำแหน่งของแอปพลิเคชัน	Fail	ไม่สามารถปิด-เปิดการเข้าถึงตำแหน่งของแอปพลิเคชัน
23.	กด logout ในหน้า profile	ออกจากระบบและกลับไปยังหน้า login	Pass	

2. ผลการทดสอบจากกลุ่มเป้าหมาย

- ผู้ทดสอบคนที่ 1 : ผลการทดสอบผ่านตามตาราง Test-Case



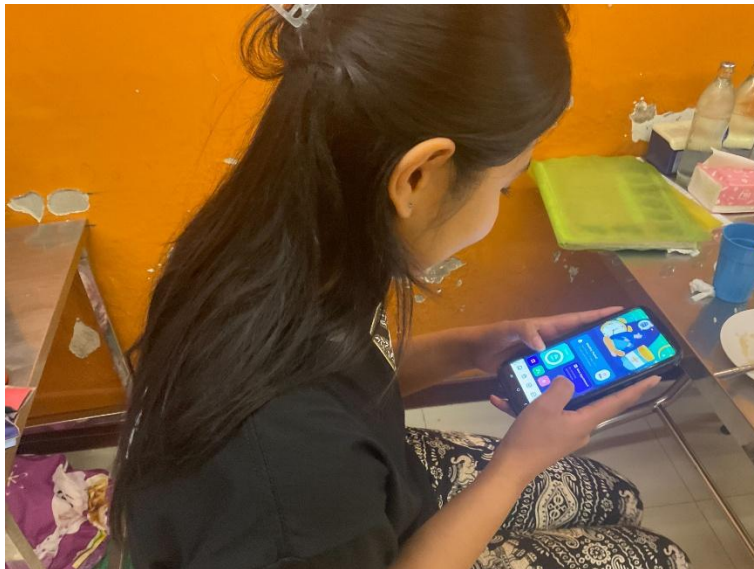
(รูปภาพที่ 43 : ผู้ทดสอบคนที่ 1)

- ผู้ทดสอบคนที่ 2 : ผลการทดสอบผ่านตามตาราง Test-Case



(รูปภาพที่ 44 : ผู้ทดสอบคนที่ 2)

- ผู้ทดสอบคนที่ 3 : ผลการทดสอบผ่านตามตาราง Test-Case



(รูปภาพที่ 45 : ผู้ทดสอบคนที่ 3)

- ผู้ทดสอบคนที่ 4 : ผลการทดสอบผ่านตามตาราง Test-Case



(รูปภาพที่ 46 : ผู้ทดสอบคนที่ 4)

- ผู้ทดสอบคนที่ 5 : ผลการทดสอบผ่านตามตาราง Test-Case



(รูปภาพที่ 47 : ผู้ทดสอบคนที่ 5)

บทที่ 7

บทสรุป

สรุปผลการดำเนินงาน

สรุปผลการดำเนินงานโครงการพัฒนา Mobile application สำหรับธุรกิจโดยทางผู้จัดทำทำการพัฒนาแอปพลิเคชันเพื่อการดูแลสุขภาพส่วนบุคคล ได้แก่ เตือนนัดหมายจากโรงพยาบาล เตือนการทานยา และเตือนการมีประจำเดือน

จากการดำเนินโครงการพัฒนาแอปพลิเคชัน แอปพลิเคชันที่พัฒนาขึ้นสามารถตอบสนองต่อวัตถุประสงค์หลักของโครงการได้อย่างครบถ้วน โดยมีความสามารถในการแจ้งเตือนการนัดหมายจากโรงพยาบาล เตือนการรับประทานยา และติดตามรอบประจำเดือนของผู้หญิง ซึ่งช่วยเพิ่มประสิทธิภาพในการดูแลสุขภาพของผู้ใช้งานได้อย่างชัดเจน

แอปพลิเคชันนี้สามารถช่วยให้ผู้ใช้งานสามารถติดตามและจัดการการทานยาได้อย่างมีประสิทธิภาพ ลดความเสี่ยงจากการลืมรับประทานยา พร้อมทั้งมีระบบบันทึกข้อมูลการนัดหมายของโรงพยาบาล และระบบแจ้งเตือนล่วงหน้าเพื่อให้ผู้ใช้งานสามารถวางแผนและเตรียมตัวได้อย่างเหมาะสม นอกจากนี้ยังมีฟีเจอร์สำหรับติดตามและคำนวณรอบประจำเดือน ซึ่งออกแบบมาเพื่อรองรับผู้ใช้งานเพศหญิงโดยเฉพาะ ทำให้สามารถติดตามรอบเดือนของตนเองได้อย่างสะดวกและแม่นยำ

ดังนั้น แอปพลิเคชันที่พัฒนาขึ้นสามารถตอบโจทย์ความต้องการของกลุ่มเป้าหมาย และมีศักยภาพในการต่อยอดและพัฒนาเพิ่มเติมในอนาคตเพื่อรองรับการใช้งานที่หลากหลายยิ่งขึ้น

สิ่งที่สมาชิกกลุ่มได้เรียนรู้

- ได้พัฒนาทักษะในการวางแผนและออกแบบระบบแอปพลิเคชันให้ตอบโจทย์ผู้ใช้งานจริง
- ฝึกการแก้ไขปัญหา และการพัฒนาแอปพลิเคชันที่ยืดการใช้งานของผู้ใช้เป็นหลัก

สิ่งที่อยากพัฒนาต่อใน Mobile Application นี้

- เพิ่มระบบผู้ช่วยสุขภาพ (Health Assistant) ที่ให้คำแนะนำการดูแลสุขภาพรายวัน
- เชื่อมต่อกับอุปกรณ์ IoT เช่น สมาร์ทวอตช์ เพื่อเก็บข้อมูลสุขภาพแบบเรียลไทม์

- พัฒนา UI/UX ให้ใช้งานง่ายและเข้าถึงได้กับผู้ใช้ทุกช่วงวัย
- เพิ่มระบบแชตกับเภสัชกรหรือแพทย์เพื่อให้คำปรึกษาเรื่องการชื้อยา

แหล่งอ้างอิง

- [หากลืมกินยาตามเวลา ควรทำอย่างไร](#)
- <https://www.sdgmovement.com/intro-to-sdgs/>
- <https://www.pinterest.com/>