



โครงการวิทยาการคอมพิวเตอร์

แอปพลิเคชันออกกำลังกาย

Exercise Application

นายวรเมธ กาญจนกวีกุล รหัสประจำตัวนิสิต 631450689

เสนอ

ผศ.ศิริกร จันท์นวล

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยเกษตรศาสตร์

ภาคปลาย ปีการศึกษา 2566

โครงการวิทยาการคอมพิวเตอร์

ชื่อเรื่อง แอปพลิเคชันออกกำลังกาย
(Exercise Application)

ผู้จัดทำ นายวรเมธ กาญจนกวีกุล รหัสประจำตัวนิสิต 6310450689

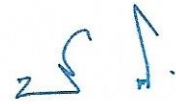
อาจารย์ที่ปรึกษาหลัก



(ผศ.ศิริกร จันทร์นวล)

8 / 4 / 2567

อาจารย์ที่ปรึกษาร่วม



(ผศ.พบสิทธิ์ กมลเวช)

8 / 4 / 2567

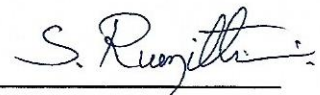
อาจารย์ที่ปรึกษาร่วม



(ผศ. ดร.อุษา สัมมาพันธ์)

8 / 4 / 67

หัวหน้าภาควิชาวิทยาการคอมพิวเตอร์



(อ.สมโชค เรืองอิทธินันท์)

9 / 4 / 67

นายวรเมธ กาญจนวิกุล. 2566. แอปพลิเคชันออกกำลังกาย.โครงการวิทยการคอมพิวเตอร์ ปริญญา
วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
มหาวิทยาลัยเกษตรศาสตร์.

อาจารย์ที่ปรึกษา: ผ.ศ.ศิริกร จันทร์นวล

บทคัดย่อ

การออกกำลังกายเป็นกิจกรรมที่สำคัญสำหรับการรักษาและเพิ่มพูนสุขภาพทั้งทางกายและใจ มีหลายเป้าหมายที่คน
อาจจะมีเมื่อต้องการออกกำลังกายรวมถึง การสร้างกล้ามเนื้อ การควบคุมน้ำหนัก และ สุขภาพที่ดีของผู้ออกกำลังกาย ใน
ปัจจุบันผู้ออกกำลังส่วนใหญ่ที่เป็นมือใหม่ที่ไม่มีความรู้ในการออกกำลังกายที่ถูกต้องอาจทำให้ร่างกายได้รับผลกระทบจากการ
ออกกำลังกาย เช่น การออกกำลังกายด้วยการยกน้ำหนัก หากยกผิดท่าหรือน้ำหนักเกินกำลัง อาจทำให้กล้ามเนื้ออักเสบ
ผู้จัดทำมองเห็นปัญหาในการออกกำลังกายให้ถูกวิธี จึงได้จัดทำแอปพลิเคชันออกกำลังกายโดยพัฒนาด้วย Flutter Framework
และภาษา Dart เพื่อให้แอปพลิเคชันสมบูรณ์ และตอบสนองความต้องการของผู้ใช้งานให้ได้มากที่สุด

คำสำคัญ: Flutter Framework, Dart

Worramet Kanjanakaweeikul. **Exercise Application.** Computer Science Project. Bachelor of Science in Computer Science, Department of Computer Science, Faculty of Science, Kasetsart University.

Project Advisor: Asst. Prof. Sirikorn Channual

ABSTRACT

Exercise is a critical activity for maintaining and enhancing both physical and mental health. People may have various goals when they decide to exercise, including muscle building, weight management, and overall health improvement. Nowadays, many beginners who lack proper exercise knowledge might suffer from the adverse effects of incorrect workout practices. For instance, lifting weights improperly or with too much weight can lead to muscle inflammation. Realizing the problem of exercising correctly, an exercise application has been developed using the Flutter Framework and Dart language to make the application comprehensive and meet the users' needs as much as possible.

Keywords: Flutter Framework, Dart

กิตติกรรมประกาศ

การจัดทำโครงการวิทยการคอมพิวเตอร์หัวข้อแอปพลิเคชันการออกกำลังกายในครั้งนี้ ผู้จัดทำได้รับความอนุเคราะห์จากอาจารย์ที่ปรึกษา ทำให้โครงการสำเร็จลุล่วงไปด้วยดี โดยได้รับคำแนะนำ การช่วยเหลือ และการสนับสนุนจากบุคคลดังนี้

ขอขอบคุณอาจารย์ที่ปรึกษาโครงการ ผศ.ศิริกร จันทร์นวล และอาจารย์ที่ปรึกษาร่วมได้แก่ ผศ.พสิษฐ์ กมลเวช และ ผศ. ดร.อุษา สัมมาพันธ์ ที่ได้สละเวลาให้ข้อเสนอแนะ แนวคิด รวมถึงเสนอแนวทางการแก้ไขปัญหาต่างๆ ที่เกิดขึ้นระหว่างการดำเนินงาน ตลอดจนการดำเนินโครงการ และให้คำปรึกษาในการเขียนรายงานอย่างดี

คณะผู้จัดทำตระหนักถึงความกรุณาจากทุกท่านที่มีส่วนช่วยให้โครงการนี้สำเร็จลุล่วงไปด้วยดี และขอขอบพระคุณมา ณ โอกาสนี้

ผู้จัดทำ

วรเมธ กาญจนกวีกุล

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ	1
1.4 ขอบเขตและข้อจำกัดของระบบ	2
1.5 ผู้ใช้ประโยชน์จากระบบ	2
1.6 เครื่องมือที่ใช้ในการพัฒนาระบบ	3
1.7 แผนการดำเนินงาน.....	3
บทที่ 2 ทฤษฎี หลักการ และเทคโนโลยีที่เกี่ยวข้อง	4
2.1 เครื่องมือที่ใช้ในการพัฒนา.....	4
2.2 ภาษาที่ใช้ในการพัฒนา.....	5
บทที่ 3 การวิเคราะห์และการออกแบบระบบ	6
3.1 การวิเคราะห์และออกแบบระบบ.....	6
3.2 Input/Output Specification.....	7
3.3 Use Case Diagram	7
3.4 Use Case Description	8
3.5 ซีเควนไดอะแกรม	11
3.6 การออกแบบฐานข้อมูล	15
3.7 การออกแบบส่วนต่อประสานกับผู้ใช้.....	16
บทที่ 4 การพัฒนาระบบ	20
4.1 การเชื่อมต่อระบบฐานข้อมูล.....	20
4.2 หน้า Log In	25
4.3 หน้า Profile	28
4.4 หน้า Exercise	31
4.5 หน้าสถิติการออกกำลังกาย	39
4.6 บันทึกข้อมูลออกกำลังกาย	40

บทที่ 5 การทดสอบระบบ	43
5.1 การทดสอบสมัครบัญชี	43
5.2 การทดสอบเข้าสู่ระบบ	44
5.3 การทดสอบเข้าสู่ระบบ	45
5.4 การทดสอบดูวิธีออกกำลังกาย	46
5.5 การทดสอบดูวิธีออกกำลังกาย	47
5.6 การทดสอบบันทึกข้อมูลออกกำลังกาย	48
บทที่ 6 บทสรุป	49
6.1 บทสรุป	49
6.2 ข้อจำกัดของระบบ	49
6.3 ปัญหาและข้อเสนอแนะ	49
6.4 แนวทางการพัฒนาต่อยอด	49
บรรณานุกรม	50
ภาคผนวก ก	51
1. การติดตั้ง Flutter SDK	51
2. การติดตั้ง Android Studio	51
3. การติดตั้ง Visual Studio Code	52
ภาคผนวก ข	53
1. หน้า Sign Up	53
2. หน้า Log in	54
3. วิธีออกกำลังกาย	55
4. วิธีบันทึกข้อมูลออกกำลังกาย	58
5. หน้า Profile	60
6. หน้า Statistic	61

สารบัญรูปภาพ

รูปที่	หน้า
รูปภาพที่ 3.1 Use Case Diagram	7
รูปภาพที่ 3.2 Sequence diagram การสร้างบัญชีผู้ใช้งาน	11
รูปภาพที่ 3.3 Sequence diagram การสร้างบัญชีผู้ใช้งาน	11
รูปภาพที่ 3.4 Sequence diagram โปรไฟล์	12
รูปภาพที่ 3.5 Sequence diagram หน้าออกกำลังกาย	13
รูปภาพที่ 3.6 Sequence diagram หน้าสถิติข้อมูลออกกำลังกาย	14
รูปภาพที่ 3.7 Sequence diagram หน้าสถิติข้อมูลออกกำลังกาย	14
รูปภาพที่ 3.8 Layout Sign Up	16
รูปภาพที่ 3.9 Layout Log In	16
รูปภาพที่ 3.10 Layout Profile	17
รูปภาพที่ 3.11 Layout Exercise	17
รูปภาพที่ 3.12 Layout Posture	18
รูปภาพที่ 3.13 Layout Info Posture	18
รูปภาพที่ 3.14 Layout Calendar	19
รูปภาพที่ 3.15 Layout Detail Calendar	19
รูปภาพที่ 4.1 ตัวอย่างการลง Package เชื่อมฐานข้อมูล	20
รูปภาพที่ 4.2 ตัวอย่างการติดตั้ง Firebase และการ Initialize	21
รูปภาพที่ 4.3 ตัวอย่างการส่งข้อมูลสู่ฐานข้อมูลบน Firebase	21
รูปภาพที่ 4.4 ตัวอย่างโค้ด Sign up	22

รูปภาพที่ 4.5 ตัวอย่างโค้ด Sign up ตอนที่ 2	23
รูปภาพที่ 4.6 ตัวอย่างโค้ด Sign up ตอนที่ 3	24
รูปภาพที่ 4.7 ตัวอย่างโค้ด Log In	25
รูปภาพที่ 4.8 ตัวอย่างโค้ด Log In ตอนที่ 2	26
รูปภาพที่ 4.9 ตัวอย่างโค้ด Log In ตอนที่ 3	27
รูปภาพที่ 4.10 ตัวอย่างโค้ด Profile	28
รูปภาพที่ 4.11 ตัวอย่างโค้ด Profile ตอนที่ 2	29
รูปภาพที่ 4.12 ตัวอย่างโค้ด Profile ตอนที่ 3	30
รูปภาพที่ 4.13 ตัวอย่างโค้ด สัดส่วนร่างกาย	31
รูปภาพที่ 4.14 ตัวอย่างโค้ด สัดส่วนร่างกาย ตอนที่ 2	32
รูปภาพที่ 4.15 ตัวอย่างโค้ด ทำออกกำลังกาย	33
รูปภาพที่ 4.16 ตัวอย่างโค้ด ทำออกกำลังกาย ตอนที่ 2	34
รูปภาพที่ 4.17 ตัวอย่างโค้ด ทำออกกำลังกาย ตอนที่ 3	35
รูปภาพที่ 4.18 ตัวอย่างโค้ด หน้าวิดีโอออกกำลังกาย	36
รูปภาพที่ 4.19 ตัวอย่างโค้ด หน้าวิดีโอออกกำลังกาย ตอนที่ 2	37
รูปภาพที่ 4.20 ตัวอย่างโค้ด หน้าวิดีโอออกกำลังกาย ตอนที่ 3	38
รูปภาพที่ 4.21 ตัวอย่างโค้ด สถิติการออกกำลังกาย	39
รูปภาพที่ 4.22 ตัวอย่างโค้ด Calendar	40
รูปภาพที่ 4.23 ตัวอย่างโค้ด Detail Calendar	41
รูปภาพที่ 4.24 ตัวอย่างโค้ด Detail Calendar ตอนที่ 2	42
รูปภาพที่ 5.1 แสดงการทดสอบหน้า Sign Up	43
รูปภาพที่ 5.2 แสดงการทดสอบหน้า Log In	44
รูปภาพที่ 5.3 แสดงการทดสอบหน้า Profile	45
รูปภาพที่ 5.4 แสดงการทดสอบหน้า ออกกำลังกาย	46

รูปภาพที่ 5.5 แสดงการทดสอบหน้า สถิติการออกกำลังกาย	47
รูปภาพที่ 5.6 แสดงการทดสอบหน้า บันทึกข้อมูลออกกำลังกาย	48
รูปภาพที่ ข.1 ภาพการใช้งานระบบ Sign Up	53
รูปภาพที่ ข.2 ภาพการใช้งานระบบ Log In	54
รูปภาพที่ ข.3 ภาพการใช้งานระบบ Exercise	55
รูปภาพที่ ข.4 ภาพการใช้งานระบบ body Screen	56
รูปภาพที่ ข.5 ภาพการใช้งานระบบ Info Posture Screen	57
รูปภาพที่ ข.6 ภาพการใช้งานระบบ Calendar Screen	58
รูปภาพที่ ข.7 ภาพการใช้งานระบบ Detail Calendar Screen	59
รูปภาพที่ ข.9 ภาพการใช้งานระบบ Profile	60
รูปภาพที่ ข.9 ภาพการใช้งานระบบ Statistic	61

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 1.1 ตารางแผนการดำเนินงาน	3
ตารางที่ 3.1 หน้า Register	7
ตารางที่ 3.2 Use case description หน้า Register	8
ตารางที่ 3.4 Use case description หน้า Profile	9
ตารางที่ 3.5 Use case description ดูท่าออกกำลังกาย	9
ตารางที่ 3.6 Use case description สถิติการออกกำลังกาย	10
ตารางที่ 3.7 Use case description บันทึกข้อมูลออกกำลังกาย	10
ตารางที่ 3.8 ฐานข้อมูล User	15
ตารางที่ 3.9 ฐานข้อมูล Workouts	15
ตารางที่ 5.1 แสดงการสมัครบัญชีของผู้ใช้งาน	43
ตารางที่ 5.2 แสดงการเข้าสู่ระบบบัญชีผู้ใช้	44
ตารางที่ 5.3 แสดงการข้อมูลโปรไฟล์	45
ตารางที่ 5.4 แสดงการเล่นออกกำลังกาย	46
ตารางที่ 5.5 แสดงการดูสถิติการออกกำลังกาย	46
ตารางที่ 5.6 แสดงการบันทึกออกกำลังกาย	47

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

เนื่องจากในปัจจุบันคนเริ่มหันมาออกกำลังกายกันมากขึ้น เพราะ การออกกำลังกายช่วยเสริมสร้างสมรรถภาพทางกาย ช่วยรักษาน้ำหนักให้ถูกสุขภาพ ช่วยย่อยอาหาร ช่วยสร้างและรักษาความหนาแน่นของกระดูก ความแข็งแรงของกล้ามเนื้อ และความยืดหยุ่นได้ของข้อต่อ ลดความเสี่ยงเมื่อผ่าตัด และเสริมความแข็งแรงของระบบภูมิคุ้มกัน

ในปัจจุบันผู้ออกกำลังกายส่วนใหญ่ที่เป็นมือใหม่ที่ไม่มีความรู้ในการออกกำลังกายที่ถูกต้องอาจทำให้ร่างกายได้รับผลกระทบจากการออกกำลังกาย เช่น การออกกำลังกายด้วยการยกน้ำหนัก หากยกผิดท่าหรือน้ำหนักเกินกำลัง อาจทำให้กล้ามเนื้ออักเสบ เป็นต้น เพื่อที่จะทำให้ออกกำลังกายถูกวิธีมีการจ้างครูฝึกเพื่อช่วยแนะนำออกกำลังกายทำให้มีค่าใช้จ่ายที่สูง

ผู้จัดทำมองเห็นปัญหาในการออกกำลังกายให้ถูกวิธี จึงได้จัดทำแอปพลิเคชันออกกำลังกายโดยเป็นการออกกำลังกายพื้นฐานที่สามารถออกกำลังกายได้โดยใช้แค่อุปกรณ์ดัมเบลอย่างถูกวิธี เพื่อให้ผู้ใช้งานสามารถออกกำลังกายสร้างหุ่น สร้างสุขภาพที่ดีได้ตัวเอง

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อสร้างแอปพลิเคชันออกกำลังกายให้ผู้ใช้งานได้ออกกำลังกายอย่างถูกต้องด้วยตัวเอง
- 1.2.2 เพื่อจัดทำแผนการออกกำลังกายสำหรับผู้ใช้เพื่อให้การฝึกฝนมีประสิทธิภาพมากยิ่งขึ้น
- 1.2.3 เพื่อสนับสนุนการวัดและติดตามความก้าวหน้าทางสุขภาพของผู้ใช้
- 1.2.4 ให้ข้อมูลและคำแนะนำที่เกี่ยวกับการดูแลสุขภาพในขณะที่ออกกำลังกาย

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1.3.1 เพื่ออำนวยความสะดวกในการออกกำลังกายให้กับผู้ใช้งาน ให้ออกกำลังกายได้อย่างถูกต้อง
- 1.3.2 เพื่อให้ผู้ใช้งานสามารถเปลี่ยนแปลงพฤติกรรมทางกายของตนได้จากการใช้แอปพลิเคชัน
- 1.3.3 การใช้เทคโนโลยีและการออกแบบที่น่าสนใจทำให้กระบวนการออกกำลังกายเป็นสนุกและน่าตื่นเต้น

1.4 ขอบเขตและข้อจำกัดของระบบ

1.4.1. การออกกำลังกายพื้นฐานโดยมีอุปกรณ์ดัมเบล ดังนี้

- 1) เสริมสร้างกล้ามเนื้อหน้าแขน
- 2) เสริมสร้างกล้ามเนื้อส่วนอก
- 3) เสริมสร้างกล้ามเนื้อส่วนขา
- 4) เสริมสร้างกล้ามเนื้อส่วนไหล่

1.4.2. โดยมุ่งเน้นไปที่กลุ่มมือใหม่โดยผู้ใช้งานสามารถใช้ระบบได้ ดังนี้

- 1) ผู้ใช้งานสามารถสมัครสมาชิกเพื่อจัดเก็บข้อมูลของผู้ใช้งาน
- 2) ผู้ใช้งานสามารถจับเวลาขณะออกกำลังกาย
- 3) ผู้ใช้งานสามารถออกแบบตารางออกกำลังกายเองได้
- 4) ผู้ใช้งานสามารถเพิ่มโปรไฟล์ตนเองเพื่อให้ระบบคำนวณสุขภาพของผู้ใช้งาน
- 5) ผู้ใช้งานสามารถเพิ่มเวลาออกกำลังกายในแต่ละเซต

1.5 ผู้ใช้ประโยชน์จากระบบ

ระบบสามารถพัฒนาต่อยอด เพื่อนำไปเป็นเว็บแอปพลิเคชันที่ใช้งานได้จริง และสามารถนำฟังก์ชันต่าง ๆ ไปพัฒนาเพื่อให้เกิดความน่าสนใจ และสะดวกต่อการใช้งานของผู้ใช้งานต่อไป

บทที่ 2

ทฤษฎี หลักการ และเทคโนโลยีที่เกี่ยวข้อง

ในบทที่ 2 จะเป็นการอธิบายถึงหลักการ ทฤษฎีที่เกี่ยวข้อง และเทคโนโลยีที่ผู้จัดทำเลือกใช้ในการ ทำโครงการนี้ เพื่อให้มีความเข้าใจเกี่ยวข้องกับโครงงานมากยิ่งขึ้น

ในปัจจุบันผู้ออกกำลังส่วนใหญ่ที่เป็นมือใหม่ที่ไม่มีความรู้ในการออกกำลังกายที่ถูกต้องอาจทำให้ร่างกายได้รับผลกระทบจากการออกกำลังกาย เช่น การออกกำลังกายด้วยการยกน้ำหนัก หากยกผิดท่าหรือน้ำหนักเกินกำลัง อาจทำให้กล้ามเนื้ออักเสบ เป็นต้น

ในโครงงานนี้เป็นการพัฒนาแอปพลิเคชัน โดยพัฒนาด้วย Flutter Framework และเขียนด้วยภาษา dart ใช้งานฐานข้อมูลจาก Google Firebase

2.1 เครื่องมือที่ใช้ในการพัฒนา

2.1.1 Visual Studio Code

คือโปรแกรมแก้ไข Source Code ที่พัฒนามาจาก Microsoft รองรับการทำงานบนเครื่องคอมพิวเตอร์หลายระบบ ทั้ง Windows, MacOS และ Linux ซึ่งรองรับหลักหลายภาษาให้ใช้งาน เช่น dart ที่นำมาใช้พัฒนาแอปพลิเคชัน ซึ่งเป็นโปรแกรมที่ใช้งานได้ง่าย ในการแก้ไขปรับแต่ง Code อีกทั้งยังเชื่อมต่อกับ Git ได้ง่ายเป็นประโยชน์ในการอัปเดตโปรเจกต์ [1]

2.1.2 Flutter Framework

Flutter คือ framework แบบ open-source สำหรับการสร้างแอปพลิเคชันมือถือ, แอปพลิเคชันเดสก์ท็อป, และแอปพลิเคชันเว็บไซต์ที่มีคุณภาพสูง โดยทำงานบนฐานของภาษาโปรแกรม Dart ซึ่งพัฒนาและรักษาโดย Google. Flutter ให้ความสามารถในการพัฒนา UI (User Interface) ที่มีประสิทธิภาพสูง, การปรับแต่งที่คล่องตัว, และการใช้งานได้หลากหลายพื้นที่ประยุกต์, จากแอปเดียวกันสามารถรันได้ทั้งบน iOS, Android, Windows, Mac, Linux, และเว็บ [2]

2.1.3 Firebase

Firebase คือ ฐานข้อมูล NoSQL Cloud Database ที่เก็บข้อมูลเป็นโครงสร้าง Json Tree และ สามารถชิงค์ข้อมูลได้อย่างเรียลไทม์ ใช้งานทุกสถานการณ์ต่อเนื่องได้ โดยไม่จำกัดว่าต้องเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต ประสิทธิภาพสูง ความเร็วในการอ่านเขียนข้อมูลรวดเร็วมากๆ ถึงแม้จะมีขนาดข้อมูลที่ใหญ่ ซึ่งเหมาะ และตอบสนองการใช้งานที่หลากหลาย อีกทั้งยังสามารถทำงานแม้อยู่ในสถานะ Offline ได้ [3]

2.1.4 Android Studio

Android Studio คือ สภาพแวดล้อมการพัฒนาแบบรวม (Integrated Development Environment, IDE) สำหรับการพัฒนาแอปพลิเคชันสำหรับแพลตฟอร์ม Android ถูกพัฒนาและซัพพอร์ตโดย Google และเป็นทางการ IDE สำหรับการพัฒนาแอป Android [4]

2.2 ภาษาที่ใช้ในการพัฒนา

2.2.1 Dart

Dart เป็นภาษาโปรแกรมมิ่งที่ถูกออกแบบและพัฒนาโดย Google. มันถูกสร้างขึ้นมาเพื่อการพัฒนาแอปพลิเคชัน สำหรับเว็บ และมือถือและเป็นภาษาที่ใช้ในการพัฒนาแอปพลิเคชันด้วย Flutter [5]

บทที่ 3

การวิเคราะห์และออกแบบระบบ

ในบทที่ 3 นี้จะกล่าวถึงการวิเคราะห์และการออกแบบการทำงานของแอปพลิเคชันในโครงงาน วิทยาการคอมพิวเตอร์ฉบับนี้เพื่อเพิ่มความเข้าใจในหลักการของแอปพลิเคชัน จึงทำการนำเสนอผลการ วิเคราะห์และออกแบบเว็บไซต์เป็นแผนภาพและตารางด้วย Use Case Diagram, ตาราง Use Case Description และ User interface

3.1 การวิเคราะห์และออกแบบระบบ

จากการวิเคราะห์ผ่านระบบผ่านแอปพลิเคชันออกกำลังกาย หรือแอปพลิเคชันต่างๆที่ใกล้เคียง และความคิดเห็นส่วนตัวของผู้ดูแลระบบ วิเคราะห์ได้ว่าแอปพลิเคชันออกกำลังกายควรมีระบบต่างๆ ดังนี้

3.1.1 ระบบสมัครสมาชิก

เป็นระบบเพิ่มบัญชีผู้ใช้ใหม่เพื่อใช้งานแอปพลิเคชันในฐานะผู้ใช้งาน โดยให้ใส่ข้อมูลที่จำเป็นแก่แอปพลิเคชัน ประกอบด้วย ชื่อผู้ใช้, อีเมล และ รหัสผ่าน

3.2.2 ระบบเข้าสู่ระบบ

เป็นระบบที่ตรวจสอบข้อมูลที่ผู้ใช้งานนำเข้า เพื่อสำหรับบันทึกข้อมูลการใช้งานของผู้ใช้

3.2.3 ระบบออกจากระบบ

เป็นระบบสำหรับออกจากบัญชีผู้ใช้ เพื่อให้ผู้ใช้งานสามารถเข้าสู่ระบบด้วยบัญชีผู้ใช้งานอื่นได้

3.2.4 ระบบโปรไฟล์

เป็นระบบสำหรับการบันทึกข้อมูลส่วนตัวผู้ใช้งาน เพื่อให้ผู้ใช้งานบันทึก ชื่อ, น้ำหนัก และ ส่วนสูง โดยสามารถสังเกตความพัฒนาของผู้ใช้งานเองได้ และ มีฟังก์ชัน คำนวณ BMI ให้สำหรับผู้ใช้งาน

3.2.5 ระบบดูวิธีออกกำลังกาย

เป็นระบบสำหรับการออกกำลังกายของผู้ใช้งาน เพื่อให้ผู้ใช้งานได้ออกกำลังกายให้ถูกวิธี โดยผู้ใช้งานสามารถเลือกจำนวนเซตออกกำลังกายได้ เล่นเมื่อผู้ใช้งานกดนับถอยหลังระบบจะนับเวลาถอยหลัง

3.2.6 ระบบสถิติการออกกำลังกาย

เป็นระบบสำหรับให้ผู้ใช้งานได้ดูสรุปรายละเอียดการออกกำลังกายของผู้ใช้งานของตัวเองได้

3.2.7 ระบบบันทึกข้อมูลออกกำลังกาย

เป็นระบบสำหรับบันทึกออกกำลังกายของผู้ใช้งานเมื่อผู้ใช้งานกดเข้าไปที่ปฏิทิน ให้ผู้ใช้งานเลือกวันที่และเลือกวางแผนการออกกำลังกายของแต่ละวันได้

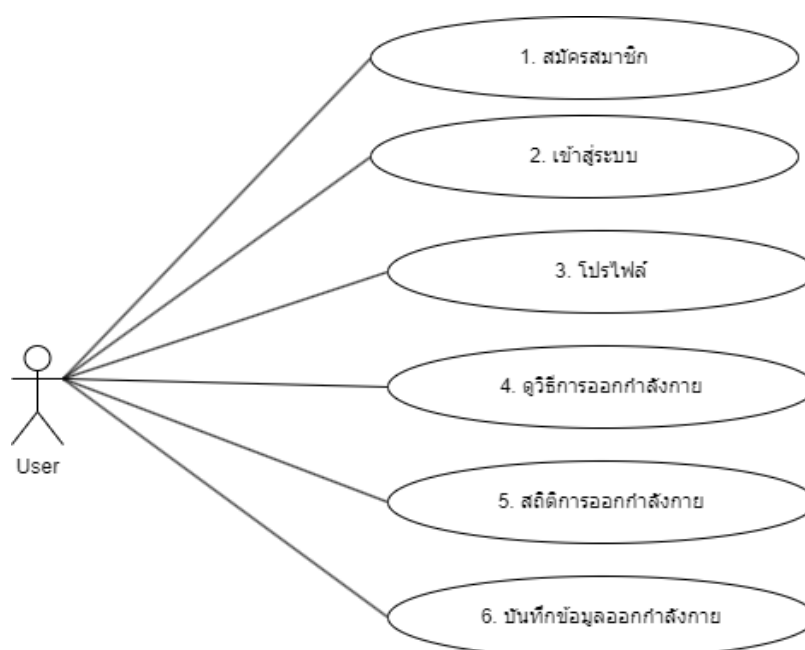
3.2 Input/Output Specification

ตารางที่ 3.1 หน้า Register

Function	Input	Process	Output
1.สมัครสมาชิก	<ul style="list-style-type: none"> - Email - Password - Confirm password 	ระบบตรวจสอบและบันทึกข้อมูลเข้าสู่ระบบ	- ผลการสมัคร
2.เข้าสู่ระบบ	<ul style="list-style-type: none"> - Email - Password 	ตรวจสอบ user ว่ามีบัญชีหรือไม่	- เข้าสู่ระบบสำเร็จ
3.โปรไฟล์ คำนวณ BMI	<ul style="list-style-type: none"> - Height - Weight 	ระบบคำนวณสูตร	- แสดงผล BMI
4.เลือกท่าที่ต้องการ	<ul style="list-style-type: none"> - video - detail text 	ระบบดึงข้อมูลที่ผู้ใช้งานได้เลือก	- แสดงท่าออกกำลังกายที่ผู้ใช้งานเลือก
5.บันทึกข้อมูลออกกำลังกาย	<ul style="list-style-type: none"> - select Body part - select Posture 	ระบบบันทึกข้อมูลที่ใช้เลือก	- แสดงผลการบันทึก
6.นับเวลาถอยหลัง	-number	คำนวณเวลา	- แสดงผลเวลา

3.3 Use Case Diagram

Use Case Diagram เป็นแผนภาพที่ใช้แสดงให้เห็นว่าระบบมีการทำงานหรือหน้าที่ใดบ้าง โดยจะใช้ Use Case จากรูปที่ 3.1 อธิบายฟังก์ชันการทำงานของเว็บไซต์



รูปภาพที่ 3.1 Use Case Diagram

3.4 Use Case Description

Use Case Description เป็นตารางที่ใช้อธิบาย functional requirement ของระบบอย่างละเอียด

1. สมัครสมาชิก

Use Case ID	1	
Name	สมัครสมาชิก	
Actor	User	
Description	สมัครสมาชิกเพื่อใช้งาน	
Pre-Condition	เปิดแอปพลิเคชัน	
Normal Process	<u>Actor</u> 1. กรอกข้อมูล 1.1 กรอก Email 1.2 กรอก Password 1.3 กรอก Confirm Password 2. กด Sign Up	<u>System</u> 3. แสดงหน้า Login
Post-Condition	แสดงหน้า Log In	

ตารางที่ 3.2 Use case description หน้า Register

2. เข้าสู่ระบบ

Use Case ID	2	
Name	เข้าสู่ระบบ	
Actor	User	
Description	ล็อกอินเข้าสู่ระบบเพื่อใช้งาน	
Pre-Condition	ต้องมีบัญชีอยู่แล้ว	
Normal Process	<u>Actor</u> 1. กรอกข้อมูล 1.1 กรอก Email 1.2 กรอก Password 2. กด Login	<u>System</u> 3. แสดงหน้าออกกำลังกาย
Post-Condition	แสดงหน้า Exercise	

ตารางที่ 3.3 หน้า Login

3. โปรไฟล์

Use Case ID	3	
Name	โปรไฟล์	
Actor	User	
Description	หน้าโปรไฟล์	
Pre-Condition	คลิกปุ่ม icon Profile	
Normal Process	<u>Actor</u> 1. กดปุ่ม 1.1 ชื่อ 1.2 กรอกข้อมูลน้ำหนัก 1.3 กรอกข้อมูลส่วนสูง 2. กดคำนวณ BMI	<u>System</u> 3. ระบบแสดงค่า BMI
Post-Condition	-	

ตารางที่ 3.4 Use case description หน้า Profile

4. ดูท่าออกกำลังกาย

Use Case ID	4	
Name	ดูท่าออกกำลังกาย	
Actor	User	
Description	ดูรายละเอียดท่าออกกำลังกาย	
Pre-Condition	คลิกปุ่ม Icon ออกกำลังกาย	
Normal Process	<u>Actor</u> 1. กดปุ่มออกกำลังกาย 2. เลือกสัดส่วนออกกำลังกาย 3. เลือกท่าที่ต้องการออกกำลังกาย 5. เลือกจำนวนเซต 6. กด Play	<u>System</u> 4. ระบบแสดงข้อมูลรายละเอียดพร้อมวิดีโอของท่าออกกำลังกาย และรายละเอียดการออกกำลังกาย 7. เวลานั้นบทยหลัง
Post-Condition	-	

ตารางที่ 3.5 Use case description ดูท่าออกกำลังกาย

5. สถิติการออกกำลังกาย

Use Case ID	5	
Name	สถิติการออกกำลังกาย	
Actor	User	
Description	สามารถดูสถิติการออกกำลังกายได้	
Pre-Condition	คลิกปุ่ม Icon สถิติ	
Normal Process	<u>Actor</u> 1. กดปุ่ม statistic	<u>System</u> 2. ระบบแสดงสถิติการออกกำลังกายของผู้ใช้
Post-Condition	-	

ตารางที่ 3.6 Use case description สถิติการออกกำลังกาย

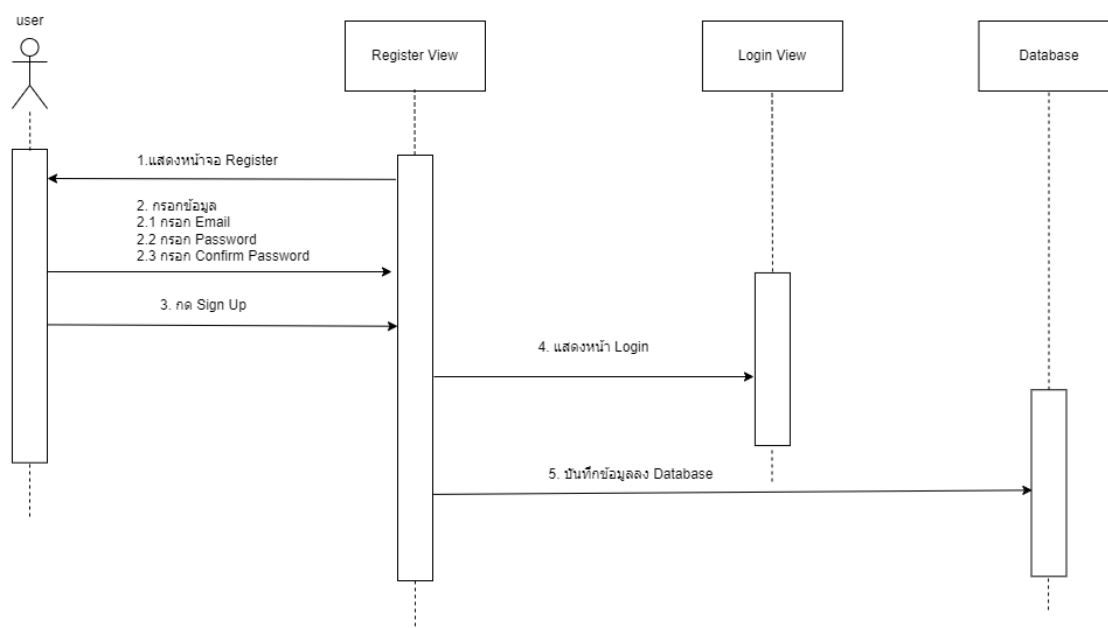
6. บันทึกข้อมูลออกกำลังกาย

Use Case ID	6	
Name	บันทึกข้อมูลออกกำลังกาย	
Actor	User	
Description	วางแผนออกแบบกำลังกายที่ปฏิบัติ	
Pre-Condition	คลิกปุ่ม Icon ปฏิบัติ	
Normal Process	<u>Actor</u> 1. กดวันที่ที่ต้องการบันทึก 2. เลือกสัดส่วนออกกำลังกาย 3. เลือกท่าออกกำลังกาย	<u>System</u> 4. ระบบบันทึกข้อมูล
Post-Condition	แจ้งการบันทึกข้อมูล	

ตารางที่ 3.7 Use case description บันทึกข้อมูลออกกำลังกาย

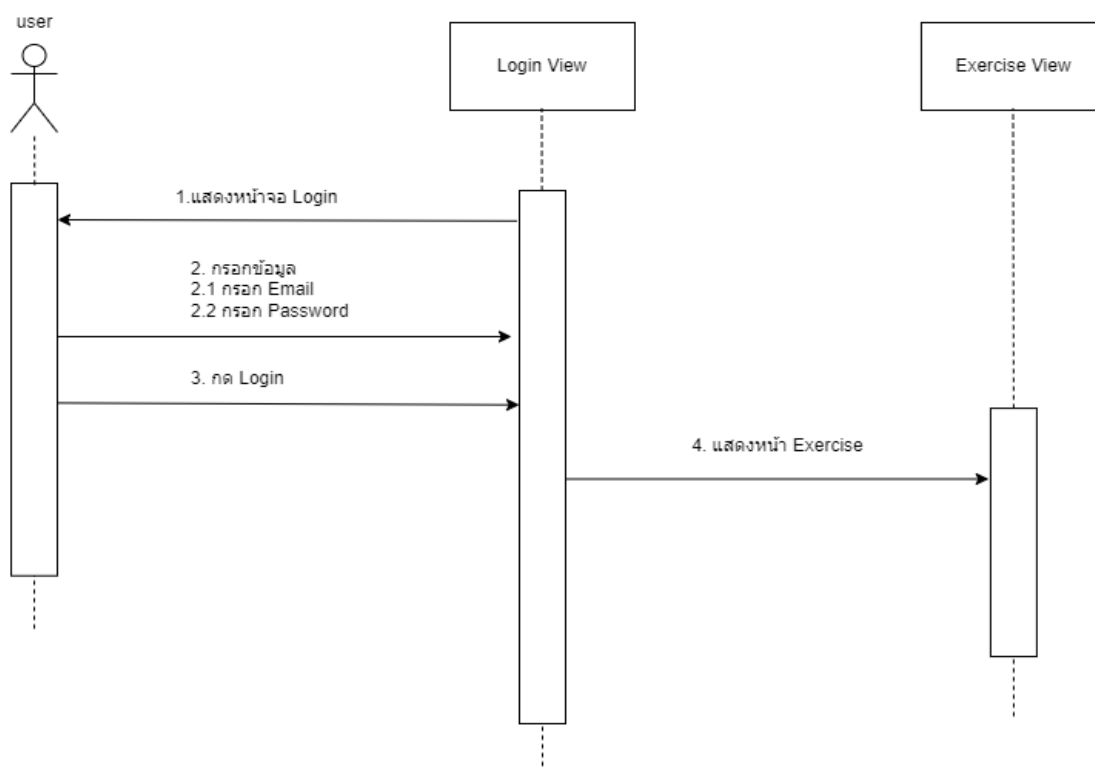
3.5 ซีควอนไดอะแกรม

1. Sequence diagram การสร้างบัญชีผู้ใช้งาน



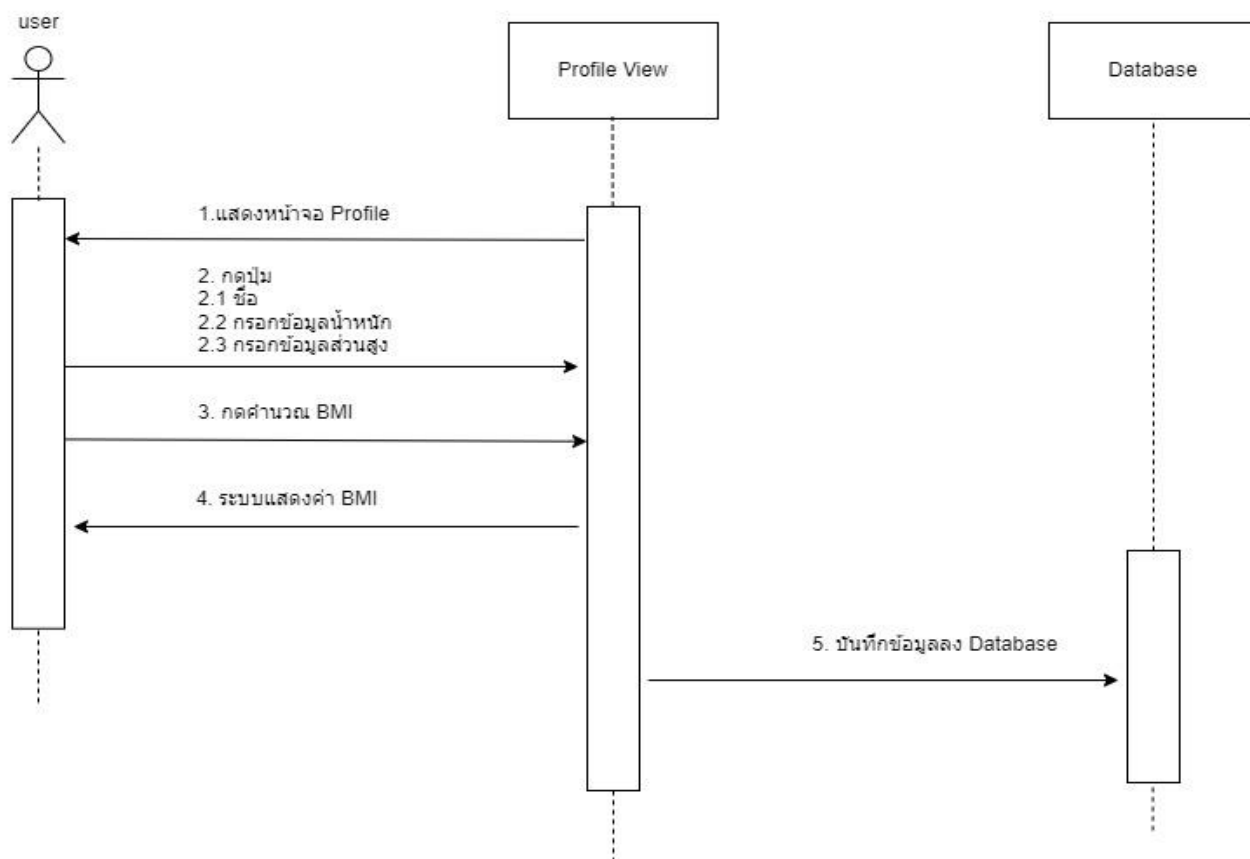
รูปภาพที่ 3.2 Sequence diagram การสร้างบัญชีผู้ใช้งาน

2. Sequence diagram การเข้าบัญชีผู้ใช้งาน



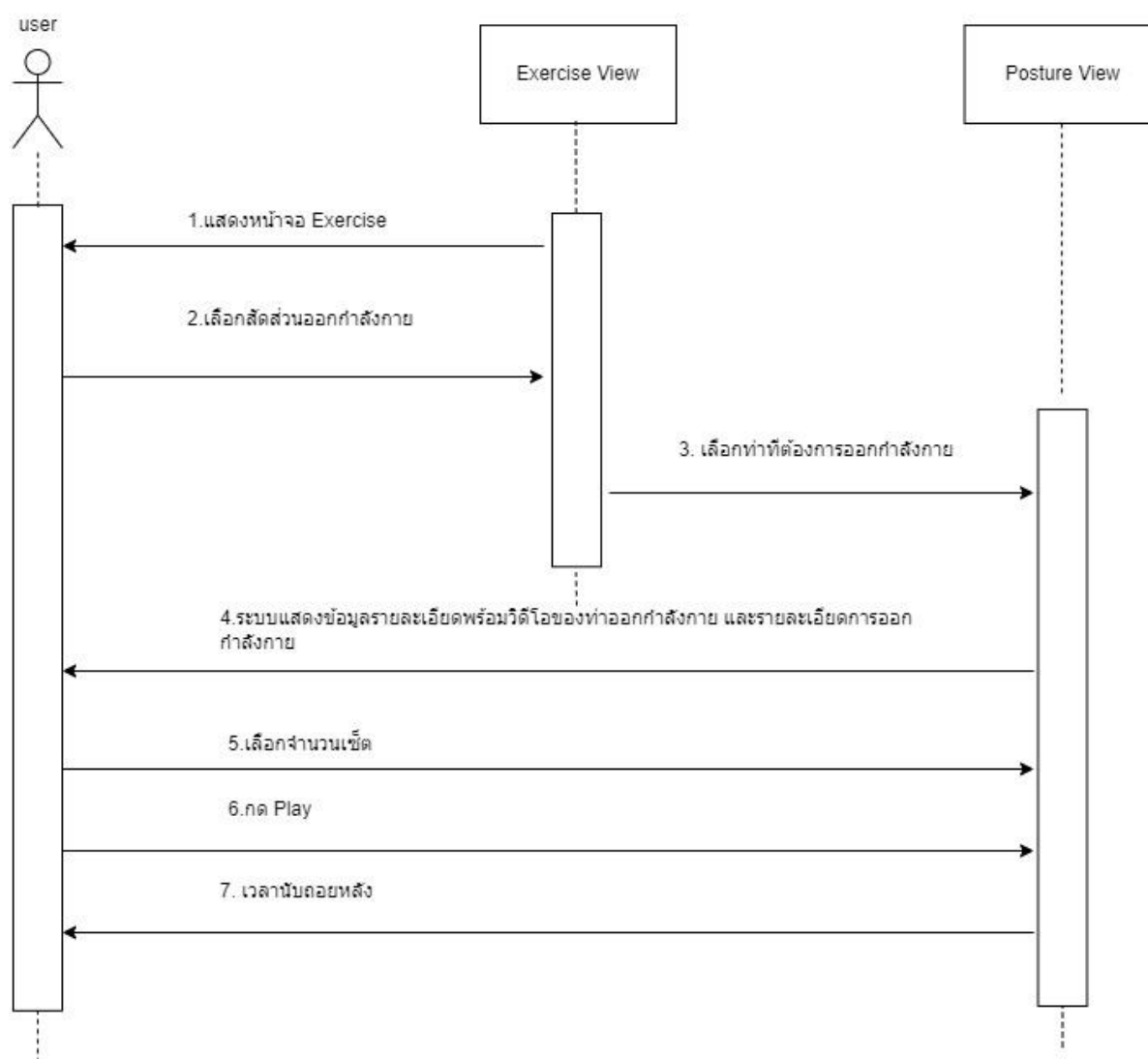
รูปภาพที่ 3.3 Sequence diagram การสร้างบัญชีผู้ใช้งาน

3. Sequence diagram โปรไฟล์



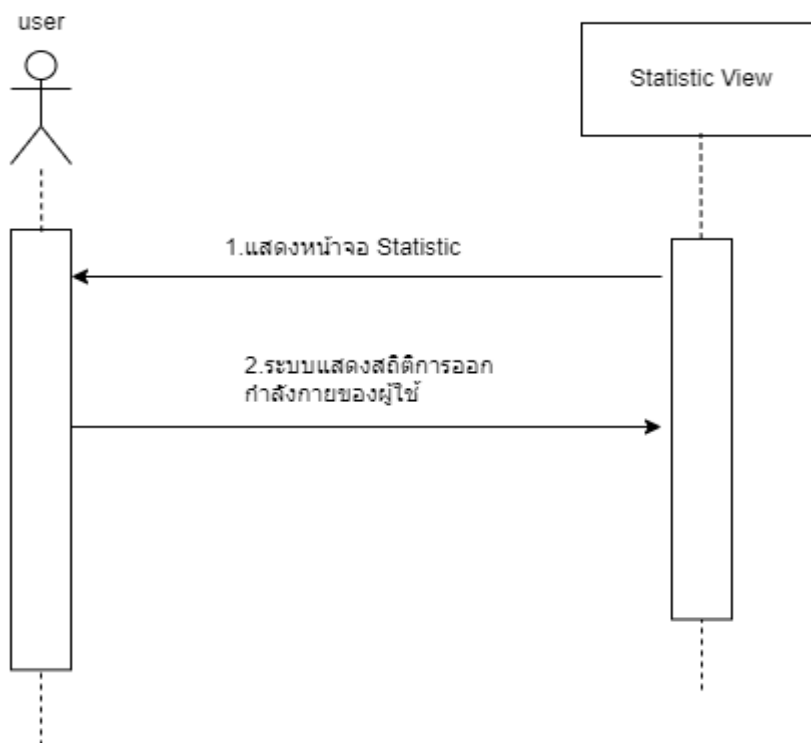
รูปภาพที่ 3.4 Sequence diagram โปรไฟล์

4. Sequence diagram หน้าออกกำลังกาย



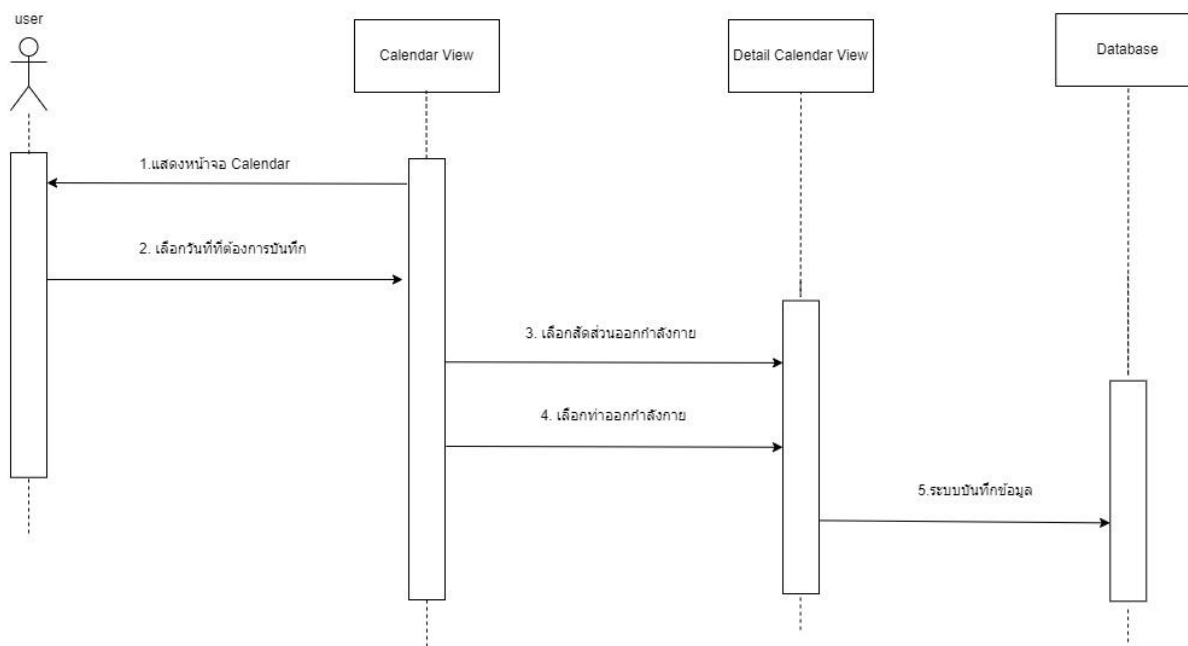
รูปภาพที่ 3.5 Sequence diagram หน้าออกกำลังกาย

5. Sequence diagram หน้าสถิติข้อมูลออกกำลังกาย



รูปภาพที่ 3.6 Sequence diagram หน้าสถิติข้อมูลออกกำลังกาย

6. Sequence diagram หน้าบันทึกการออกกำลังกาย



รูปภาพที่ 3.7 Sequence diagram หน้าสถิติข้อมูลออกกำลังกาย

3.6 การออกแบบฐานข้อมูล

การออกแบบฐานข้อมูลของโครงการนี้เป็นการออกแบบฐานข้อมูลโดยใช้ NoSQL

ชื่อตาราง: User

คำอธิบาย: ข้อมูลผู้ใช้งาน

collection	Field	Data type
User	bmi	number
	Email	string
	Height	number
	Weight	number
	Name	string
	Uid	string
	weight	number

ตารางที่ 3.8 ฐานข้อมูล User

ชื่อตาราง: Workouts

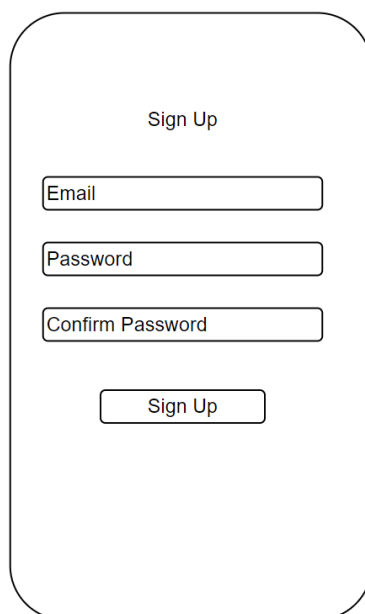
คำอธิบาย: ข้อมูลออกแบบการออกกำลังกายในแต่ละวัน

collection	Field	Data type
Workouts	date	number
	bodyPart	string
	Posture	string

ตารางที่ 3.9 ฐานข้อมูล Workouts

3.7 การออกแบบส่วนต่อประสานกับผู้ใช้

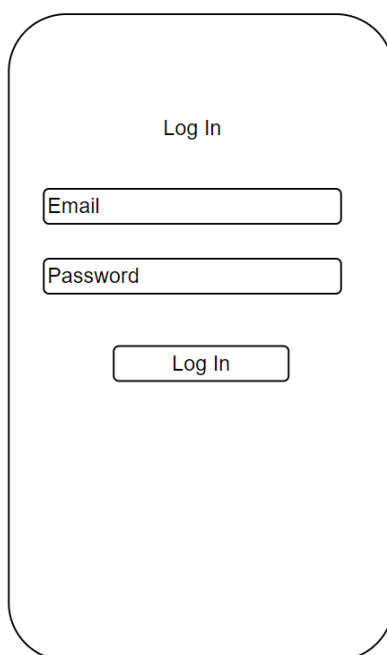
1. แสดงการออกแบบ Layout หน้า Sign up ซึ่งเป็นหน้าแรกที่จะแสดงเมื่อผู้ใช้งานเปิดแอปพลิเคชัน



A wireframe of a 'Sign Up' form. The form is contained within a rounded rectangle. At the top, the text 'Sign Up' is centered. Below it are three input fields, each with a label on the left: 'Email', 'Password', and 'Confirm Password'. At the bottom of the form is a button labeled 'Sign Up'.

รูปภาพที่ 3.8 Layout Sign Up

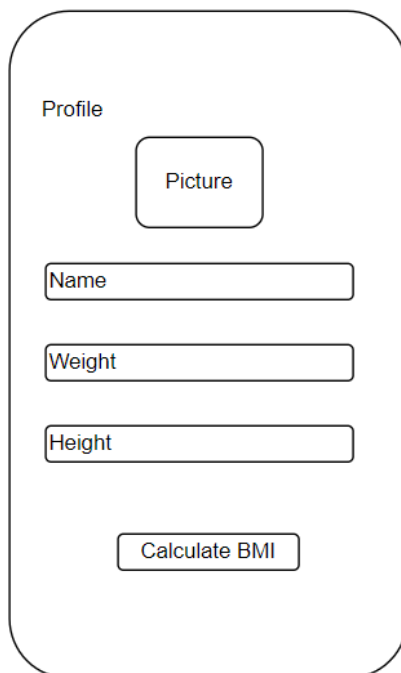
2. แสดงการออกแบบ Layout หน้า Log in ให้สำหรับเข้าสู่ระบบ



A wireframe of a 'Log In' form. The form is contained within a rounded rectangle. At the top, the text 'Log In' is centered. Below it are two input fields, each with a label on the left: 'Email' and 'Password'. At the bottom of the form is a button labeled 'Log In'.

รูปภาพที่ 3.9 Layout Log In

3. แสดงการออกแบบหน้า Profile



Profile

Picture

Name

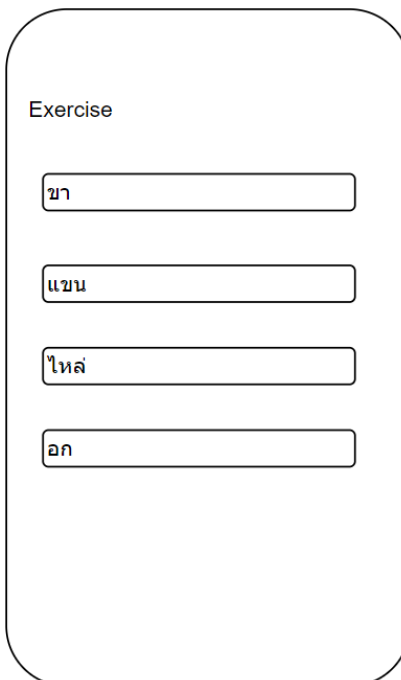
Weight

Height

Calculate BMI

รูปภาพที่ 3.10 Layout Profile

4. แสดงการออกแบบหน้า Exercise ให้ผู้ใช้งานเลือกสัดส่วนร่างกายที่ต้องการจะออกกำลังกาย



Exercise

ขา

แขน

ไหล่

อก

รูปภาพที่ 3.11 Layout Exercise

5. แสดงการออกแบบหน้า Posture ให้ผู้ใช้งานเลือกท่าร่างกายที่ต้องการจะออกกำลังกาย

ข่า

ท่าออกกำลังกาย 1

ท่าออกกำลังกาย 2

ท่าออกกำลังกาย 3

ท่าออกกำลังกาย 4

รูปภาพที่ 3.12 Layout Posture

6. แสดงการออกแบบหน้า Info Posture เพื่อให้ผู้ใช้งานได้ทำความเข้าใจของท่าออกกำลังกายและออกกำลังกายด้วยตนเอง

วิดีโอ

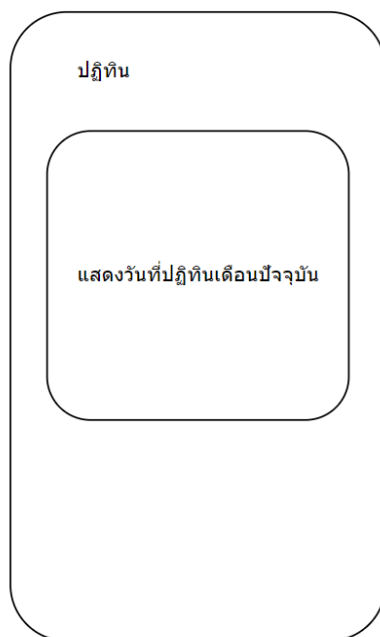
คำอธิบายท่าออกกำลังกาย

จำนวนเซต

เริ่มนับถอยหลัง

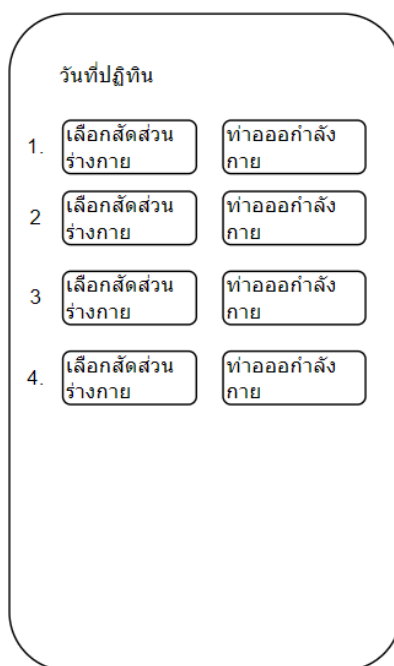
รูปภาพที่ 3.13 Layout Info Posture

7. แสดงการออกแบบหน้า Calendar



รูปภาพที่ 3.14 Layout Calendar

8. แสดงการออกแบบหน้า Detail Calendar เพื่อให้ผู้ใช้งานได้วางแผนการออกกำลังกาย



รูปภาพที่ 3.15 Layout Detail Calendar

บทที่ 4

การพัฒนาระบบ

ในบทที่ 4 ได้กล่าวถึงรายละเอียดการพัฒนาระบบ เครื่องมือที่ใช้ในการ รวมถึง Code บางส่วนเพื่อให้เกิดความเข้าใจในการทำงานของระบบมากยิ่งขึ้น และแนวทางการทดสอบระบบเพื่อค้นหาข้อผิดพลาดต่าง ๆ ของการทำงาน เพื่อนำไปแก้ไขและปรับปรุงให้ดียิ่งขึ้น

ในการพัฒนาโครงงานนี้พัฒนาโดย Flutter Framework เขียนด้วยภาษา Dart ใช้งานฐานข้อมูลจาก Firebase ใช้โปรแกรม Visual studio code และ Android Studio code เป็นเครื่องมือที่ใช้ในการเขียนโปรแกรม

เครื่องมือที่ใช้ในการพัฒนา

ซอฟต์แวร์ที่ใช้ในการพัฒนา

1. Visual Studio Code
2. Android Studio Code

ฮาร์ดแวร์ที่ใช้ในการพัฒนา

1. Notebook Lenovo

4.1 การเชื่อมต่อระบบฐานข้อมูล

4.1.1 Firebase

เชื่อมต่อฐานข้อมูลสำหรับการพัฒนาแอปพลิเคชันออกกำลังกาย โดยใช้ Firebase ในการเก็บข้อมูลบน Cloud Firestore โดยลง Package ที่ Folder pubspec.yaml ดังนี้

```
dependencies:  
  flutter:  
    sdk: flutter  
  firebase_core: ^2.24.2  
  firebase_auth: ^4.15.3  
  cloud_firestore: ^4.14.0
```

รูปภาพที่ 4.1 ตัวอย่างการลง Package เชื่อมฐานข้อมูล

จากนั้นเพิ่มการเข้าถึงฐานข้อมูลที่ไฟล์ต้องการเชื่อมต่อกับฐานข้อมูล โดย Initialize เพื่อระบุการเชื่อมต่อกับฐานข้อมูลบน Firebase ดังตัวอย่างนี้

```
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_auth/firebase_auth.dart';
static Future<FirebaseApp> initializeApp({
  String? name,
  FirebaseOptions? options,
}) async {
  FirebaseAppPlatform app = await _delegate.initializeApp(
    name: name,
    options: options,
  );

  return FirebaseApp._(app);
}
```

รูปภาพที่ 4.2 ตัวอย่างการติดตั้ง Firebase และการ Initialize

ในการส่งข้อมูลหรือสร้างฟังก์ชันระบุ Path References เพื่อบอกชื่อ collection หรือชื่อ document พร้อมข้อมูลที่ต้องการจะเก็บ

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_storage/firebase_storage.dart';

class UpdateUser {
  final firestore = FirebaseFirestore.instance;
  final FirebaseStorage storage = FirebaseStorage.instance;
  Future<void> updateBmi(String uid, double bmi) async {
    await firestore.collection('user').doc(uid).update({
      'bmi': bmi,
    });
  }
}
```

รูปภาพที่ 4.3 ตัวอย่างการส่งข้อมูลสู่ฐานข้อมูลบน Firebase

4.2 หน้า Sign Up

สมัครบัญชีผู้ใช้โดยให้ผู้ใช้กรอก email, password และ Confirm password

```
class SignupScreen extends StatefulWidget {
  const SignupScreen({Key? key}) : super(key: key);
  @override
  _SignupScreenState createState() => _SignupScreenState();
}
class _SignupScreenState extends State<SignupScreen> {
  final emailController = TextEditingController();
  final passwordController = TextEditingController();
  final confirmPasswordController = TextEditingController();
  final formKey = GlobalKey<FormState>();
  Profile profile = Profile(email: "", password: "", uid: "");
  final Future<FirebaseApp> firebase = Firebase.initializeApp();
  final firestore = FirebaseFirestore.instance;
  @override
  void initState() {
    super.initState();
    emailController.addListener(() {
      setState(() {});
    });
    passwordController.addListener(() {
      setState(() {});
    });
    confirmPasswordController.addListener(() {
      setState(() {});
    });
  }
}
```

รูปภาพที่ 4.4 ตัวอย่างโค้ด Sign up

```

@override
Widget build(BuildContext context) {
  return FutureBuilder(
    future: firebase,
    builder: (context, snapshot) {
      if (snapshot.hasError) {
return Scaffold(
  appBar: AppBar(
    title: const Text("Error"),
  ),
  body: Center(
    child: Text("${snapshot.error}"),
  ),
);
      } else if (snapshot.connectionState == ConnectionState.done) {
return Scaffold(
  resizeToAvoidBottomInset: false,
  body: Container(
    color: Colors.black87,
    child: Padding(
      padding: const EdgeInsets.all(30.0),
      child: Form(
        key: formKey,
        child: SingleChildScrollView(
          child: Column(
            children: [
              SizedBox(
                height: MediaQuery.of(context).size.height *
0.2,
                child: const Row(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: [
                    Text("Sign Up",
                      style: TextStyle(
                        fontSize: 50,
                        fontWeight: FontWeight.bold,
                        color: Colors.orange)
                    ],
                  ),
                ),
              ],
            ),
          ),
        ),
      ),
    ),
  ),
);

```

รูปภาพที่ 4.5 ตัวอย่างโค้ด Sign up ต่อ 2

```

Column(
    children: [
        Padding(
            padding: const EdgeInsets.fromLTRB(0, 0, 0,
25),
            child: buildTextFormField(
                controller: emailController,
                icon: Icons.email_rounded,
                hint: "Email",
                validators: ffv.MultiValidator([
                    ffv.RequiredValidator(
                        errorText: "Please enter your
email"),
                    ffv.EmailValidator(
                        errorText:
                            "Please enter a valid email
address"),
                ]),
            ),
        ),
        Padding(
            padding: const EdgeInsets.fromLTRB(0, 0, 0,
25),
            child: buildTextFormField(
                controller: passwordController,
                icon: Icons.lock,
                hint: "Password",
                validators: ffv.RequiredValidator(
                    errorText: "Please enter your
password"),
                obscureText: true,
            ),
        ),
    ],
)

```

รูปภาพที่ 4.6 ตัวอย่างโค้ด Sign up ต่อ 3

4.2 หน้า Log In

เข้าสู่ระบบบัญชีผู้ใช้โดยให้ผู้ใช้กรอก email และ password เพื่อเข้าสู่ระบบ

```
class LoginScreen extends StatefulWidget {
  const LoginScreen({super.key});

  @override
  State<LoginScreen> createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  TextEditingController? emailcontroller;
  TextEditingController? passwordcontroller;
  final formKey = GlobalKey<FormState>();
  Profile profile = Profile(email: "", password: "", uid: "");
  final Future<FirebaseApp> firebase = Firebase.initializeApp();
  @override
  void initState() {
    super.initState();
    emailcontroller = TextEditingController()
      ..addListener(() {
        setState(() {});
      });
    passwordcontroller = TextEditingController()
      ..addListener(() {
        setState(() {});
      });
  }
}
```

รูปภาพที่ 4.7 ตัวอย่างโค้ด Log In

```

@override
Widget build(BuildContext context) {
  return FutureBuilder(
    future: firebase,
    builder: (context, snapshot) {
      if (snapshot.hasError) {
        return Scaffold(
          appBar: AppBar(
            title: const Text("Error"),
          ),
          body: Center(
            child: Text("${snapshot.error}"),
          ),
        );
      } else if (snapshot.connectionState == ConnectionState.done) {
        return Scaffold(
          resizeToAvoidBottomInset: false,
          body: Container(
            color: Colors.black87,
            child: Column(
              children: [
                SizedBox(
                  height: MediaQuery.of(context).size.height * 0.2,
                  child: const Row(
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: [
                      Text("Log In",
                        style: TextStyle(
                          fontSize: 50,
                          fontWeight: FontWeight.bold,
                          color: Colors.orange))
                    ],
                  ),
                ],
              ),
            ),
          ),
        );
      }
    },
  );
}

```

รูปภาพที่ 4.8 ตัวอย่างโค้ด Log In ต่อที่ 2

```

Column(
  children: [
    Padding(
      padding: const EdgeInsets.fromLTRB(0, 0, 0,
25),
      child: Container(
        width: MediaQuery.of(context).size.width *
0.8,
        height: MediaQuery.of(context).size.height *
0.055,
        decoration: BoxDecoration(
          color: Colors.white,
          borderRadius: BorderRadius.circular(15),
        ),
        child: Row(
          children: [
            const Padding(
              padding:
EdgeInsetsDirectional.symmetric(
                horizontal: 10),
              child: Icon(
                Icons.email_rounded,
                color: Colors.black87,
              ),
            ),
            Expanded(
              child: TextFormField(
                validator: MultiValidator([
                  RequiredValidator(
                    errorText: "Please enter your
email"),
                  EmailValidator(
                    errorText:
"please enter an email
address")
                ]),

```

รูปภาพที่ 4.9 ตัวอย่างโค้ด Log In ส่วนที่ 3

4.3 หน้า Profile

ในส่วนของโปรไฟล์ให้ผู้ใช้สามารถกรอก รูป, ชื่อ, น้ำหนัก และส่วนสูง โดยระบบจะมีฟังก์ชัน คำนวณ BMI ให้กับผู้ใช้

```
void loadUserProfile() async {
    String userId = ReadUserData().getCurrentUserId();

    var userData =
        await
        FirebaseFirestore.instance.collection('user').doc(userId).get();
    var user = userData.data();
    if (user != null) {
        setState(() {
            nameController.text = user['name'] ?? '';
            weightController.text = user['weight']?.toString() ?? '';
            heightController.text = user['height']?.toString() ?? '';
        });
    }
}

void selectImage() async {
    final ImagePicker picker = ImagePicker();
    final XFile? image = await picker.pickImage(source:
    ImageSource.gallery);
    if (image != null) {
        Uint8List imageBytes = await image.readAsBytes();
        setState(() {
            this.image = imageBytes;
        });
    }
}
```

รูปภาพที่ 4.10 ตัวอย่างโค้ด Profile

```

@override
Widget build(BuildContext context) {
  return FutureBuilder(
    future: Firebase.initializeApp(),
    builder: (context, snapshot) {
      if (snapshot.connectionState == ConnectionState.done) {
        return SingleChildScrollView(
          padding: const EdgeInsets.all(16.0),
          child: Form(
            key: formKey,
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                if (image != null)
                  CircleAvatar(
                    radius: 64,
                    backgroundImage: MemoryImage(image!),
                  )
                else
                  const CircleAvatar(
                    radius: 64,
                    backgroundImage:
                      AssetImage('path/to/default/avatar/image'),
                  ),
                IconButton(
                  onPressed: selectImage,
                  icon: const Icon(Icons.add_a_photo),
                ),
              ],
            ),
          ),
        );
      }
    },
  );
}

```

รูปภาพที่ 4.11 ตัวอย่างโค้ด Profile ต่อที่ 2


```

Row (
  children: [
    Expanded(
      child: TextFormField(
        controller: heightController,
        keyboardType: const
TextInputType.numberWithOptions(
          decimal: true),
        decoration: InputDecoration(
          labelText: 'Height (cm)',
          suffixIcon: IconButton(
            icon: const Icon(Icons.edit),
            onPressed: () {

          },
        ),
      ),
    ),
    onFieldSubmitted: (value) {

      if (heightController.text.isNotEmpty &&
        double.tryParse(heightController.text) !=
          null) {
        updateUserData('height',
          double.parse(heightController.text));

        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            content: Text(
              'Height updated to
${heightController.text} cm')),
          );
      }
    },
  ],

```

รูปภาพที่ 4.12 ตัวอย่างโค้ด Profile ต่อที่ 3

4.4 หน้า Exercise

4.4.1 เลือกสัดส่วนร่างกาย

ผู้ใช้งานกดเลือกสัดส่วนร่างกายที่ต้องการ โดยแบ่งเป็น ชาย, แขน, ไหล่ และ ออก

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text("Exercise"),
      automaticallyImplyLeading: false,
    ),
    body: Padding(
      padding: const EdgeInsets.all(10.0),
      child: Center(
        child: Column(
          children: [
            ElevatedButton(
              style: ElevatedButton.styleFrom(
                foregroundColor: Colors.black,
                backgroundColor: Colors.red,
                padding: const EdgeInsets.symmetric(
                  vertical: 20), // Button padding
                textStyle: const TextStyle(fontSize: 20),
                minimumSize: const Size(double.infinity, 80),
              ),
              child: const Text('๓'),
              onPressed: () {
                Navigator.push(context,
                  MaterialPageRoute(builder: (context) =>
                    LegScreen()));
              },
            ),
          ],
        ),
      ),
    ),
  );
}
```

รูปภาพที่ 4.13 ตัวอย่างโค้ด สัดส่วนร่างกาย

```

const SizedBox(height: 20),
  ElevatedButton(
    style: ElevatedButton.styleFrom(
      foregroundColor: Colors.black,
      backgroundColor: Colors.red, // Text color
      padding: const EdgeInsets.symmetric(
        vertical: 20),
      textStyle: const TextStyle(fontSize: 20),
      minimumSize: const Size(double.infinity, 80),
    ),
    child: const Text('๓'),
    onPressed: () {
      Navigator.push(context,
        MaterialPageRoute(builder: (context) =>
BreastScreen())));
    },
  ),
const SizedBox(height: 20),
ElevatedButton(
  style: ElevatedButton.styleFrom(
    foregroundColor: Colors.black,
    backgroundColor: Colors.red,
    padding: const EdgeInsets.symmetric(
      vertical: 20),
    textStyle: const TextStyle(fontSize: 20),
    minimumSize: const Size(double.infinity, 80),
  ),
  child: const Text('ไหล่'),
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => ShoulderScreen()));
  },
),

```

รูปภาพที่ 4.14 ตัวอย่างโค้ด สัดส่วนร่างกาย ตอนที่ 2

4.4.2 เลือกท่าออกกำลังกาย

ผู้ใช้งานกดเลือกสัดส่วนร่างกายที่ต้องการ โดยแบ่งเป็น ขา, แขน, ไหล่ และ ออก และ สร้างตัวแปร ทำ reference path ตัววิดีโอ และ ข้อความอธิบายท่าเล่น ไปในหน้า posture screen

```

    SizedBox(
      width: double.infinity,
      child: Material(
        elevation: 5,
        child: ElevatedButton(
          style: ElevatedButton.styleFrom(
            backgroundColor: Colors.red,
            minimumSize:
              const Size(150, 80),
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(5.0),
            ),
          ),
          child: const Text(
            "Dumbbell Biceps Curl",
            style: TextStyle(
              fontSize: 20,
              color: Colors.black),
          ),
          onPressed: () {
            // การกระทำเมื่อกดปุ่ม
            _navigateToInfoPosture(
              context,
              'assets/video/Dumbbell_Biceps_Curl.mp4',
              'ท่าเล่นแขน ดัมเบล Dumbbell Bicep Curl ที่ช่วยเสริมส่วนกล้ามเนื้อหน้าแขน เก็บศอกให้ชิดข้างลำตัว ไม่แกว่งเวลาผ่อนและออกแรง จะทำให้โดนกล้ามเนื้อหน้าแขนมากขึ้น',
            );
          },
        ),
      ),
    ),
  ),
),

```

รูปภาพที่ 4.15 ตัวอย่างโค้ด ท่าออกกำลังกาย

```

SizedBox(
  width: double.infinity,
  child: Material(
    elevation: 5,
    child: ElevatedButton(
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.red,
        minimumSize: const Size(double.infinity,
          80),
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(5.0),
        ),
      ),
      child: const Text(
        "Dumbbell Hammer Curl",
        style: TextStyle(
          fontSize: 20,
          color: Colors.black),
      ),
      onPressed: () {
        _navigateToInfoPosture(
          context,
          'assets/video/Dumbbell_hammer_Curl.mp4',
          'จัดตำแหน่งดัมเบลทั้งสองไว้ข้างตัว หันฝ่ามือทั้งสองข้างเข้าด้านใน แขนทั้งสองข้างอยู่ในลักษณะตรงจัดข้อศอกให้อยู่ด้านข้างลำตัว ยกดัมเบลด้านหนึ่งจนกระทั่งปลายแขนเป็นแนวตั้ง และนิ้วหัวแม่มืออยู่หน้าไหล่ และลดระดับไปที่ตำแหน่งเดิม และทำซ้ำ ',
        );
      },
    ),
  ),
),

```

รูปภาพที่ 4.16 ตัวอย่างโค้ด ทำออกกำลังกาย ต่อที่ 2

```

SizedBox(
  width: double.infinity,
  child: Material(
    elevation: 5,
    child: ElevatedButton(
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.red,
        minimumSize: const Size(double.infinity,
          80),
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(5.0),
        ),
      ),
      child: const Text(
        "Dumbbell Overhead Extension",
        style: TextStyle(
          fontSize: 20,
          color: Colors.black),
      ),
      onPressed: () {
        _navigateToInfoPosture(
          context,
          'assets/video/dumbbell_overhead_extension.mp4',
          'ยืนกางขาเล็กน้อย ยึดตัวตรง ไม่ห่อไหล่ และถือดัมเบลด้วยมือทั้งสองข้าง
โดยยกขึ้นมาเหนือหลังศีรษะ และเหยียดแขนให้ตรงจนรู้สึกว่ายึดแน่น หลังจากนั้นค่อยๆย่อแขนบริเวณข้อศอกให้ดัมเบลไป
ด้านหลัง และเหยียดแขนกลับมาในท่าเริ่มต้น ซึ่งหายใจเข้าเมื่อยกขึ้นเหนือศีรษะ และหายใจออกเมื่อลง',
        );
      },
    ),
  ),
),

```

รูปภาพที่ 4.17 ตัวอย่างโค้ด ทำออกกำลังกาย ตอนที่ 3

4.4.3 หน้าคลิกวิดีโอและข้อความ

ผู้ใช้งานสามารถกดเล่นวิดีโอ และ ข้อความที่อธิบายท่าเล่นออกกำลังกาย รวมถึงเลือกจำนวนเซตออกกำลังกายและนับเวลาถอยหลังได้ด้วย

```
body: SingleChildScrollView(
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.stretch,
    children: [
      SizedBox(
        height: MediaQuery.of(context).size.height / 3,
        child: _controller.value.isInitialized
          ? Chewie(
              controller: _chewieController,
            )
          : const Center(
              child: CircularProgressIndicator(),
            ),
      ),
      Padding(
        padding: const EdgeInsets.all(8.0),
        child: SizedBox(
          height: 50,
          width: 50,
          child: FittedBox(
            child: FloatingActionButton(
              onPressed: () {
                setState(() {
                  if (_controller.value.isPlaying) {
                    _controller.pause();
                  } else {
                    _controller.play();
                  }
                });
              },
            ),
          ),
        ),
      ),
    ],
  ),
)
```

รูปภาพที่ 4.18 ตัวอย่างโค้ด หน้าวิดีโอออกกำลังกาย

```

Padding(
  padding: const EdgeInsets.all(15.0),
  child: Text(
    widget.detailText,
    style: const TextStyle(
      fontSize: 16.0,
      color: Colors.black,
    ),
    textAlign: TextAlign.center,
  ),
),
),
DropdownButton<int>(
  value: selectedTime,
  items: [1, 2, 3, 4].map((int value) {
    return DropdownMenuItem<int>(
      value: value,
      child: Text(value.toString()),
    );
  }).toList(),
  onChanged: (int? newValue) {
    if (newValue != null) {
      setState(() {
        selectedTime = newValue;
        updateCountdown();
      });
    }
  },
),

```

รูปภาพที่ 4.19 ตัวอย่างโค้ด หน้าวิดีโอออกกำลังกาย ตอนที่ 2


```

isExpanded:
    true,
    underline: Container(
        height: 2,
        color: Colors.red,
    ),
),
Center(
    child: Text(
        "${(countdown ~/ 60).toString().padLeft(2,
'0')}:${(countdown % 60).toString().padLeft(2, '0')}",
        style:
            const TextStyle(fontSize: 48, fontWeight:
FontWeight.bold),
        textAlign: TextAlign.center,
    ),
),
Padding(
    padding:
        const EdgeInsets.symmetric(vertical: 20.0, horizontal:
60.0),
    child: ElevatedButton(
        onPressed: startCountdown,
        style: ElevatedButton.styleFrom(
            foregroundColor: Colors.white,
            backgroundColor: Colors.red,
            shape: const StadiumBorder(),
            padding: const EdgeInsets.symmetric(
                vertical: 14.0),
        ),
        child: const Text("เริ่มนับถอยหลัง"),
    ),
),
],
),
),

```

รูปภาพที่ 4.20 ตัวอย่างโค้ด หน้าวิดีโอออกกำลังกาย ตอนที่ 3

4.5 หน้าสถิติการออกกำลังกาย

ในส่วนของสถิติการออกกำลังกายระบบจะดึงข้อมูลจากการที่ผู้ใช้งานกดนับถอยหลัง และจะแสดงสถิติการออกกำลังกายของผู้ใช้งาน

```
class PieChartWidget extends StatelessWidget {
  final int legCount;
  final int armCount;
  final int shoulderCount;
  final int breastCount;

  const PieChartWidget({
    Key? key,
    required this.legCount,
    required this.armCount,
    required this.shoulderCount,
    required this.breastCount,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return SizedBox(
      height: 450,
      child: PieChart(
        PieChartData(
          sections: [
            PieChartSectionData(
              color: Colors.blue,
              value: legCount.toDouble(),
              title: 'Leg',
              radius: 70,
            ),
            PieChartSectionData(
              color: Colors.green,
              value: armCount.toDouble(),
              title: 'Arm',
              radius: 70,
            ),
            PieChartSectionData(
              color: Colors.yellow,
              value: shoulderCount.toDouble(),
              title: 'Shoulder',
              radius: 70,
            ),
            PieChartSectionData(
              color: Colors.pink,
              value: breastCount.toDouble(),
              title: 'Breast',
              radius: 70,
            ),
          ],
        ),
      ),
    );
  }
}
```

รูปภาพที่ 4.21 ตัวอย่างโค้ด สถิติการออกกำลังกาย

4.6 บันทึกข้อมูลออกกำลังกาย

4.6.1 Calendar

ในส่วนนี้จะหน้าปฏิทินให้ผู้ใช้งานได้เลือกวันที่ที่ต้องการไปวางแผนการออกกำลังกาย

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('ปฏิทิน'),
      automaticallyImplyLeading: false, // Add this line
    ),
    body: TableCalendar(
      firstDay: DateTime.utc(2010, 10, 16),
      lastDay: DateTime.utc(2030, 3, 14),
      focusedDay: _focusedDay,
      selectedDayPredicate: (day) {
        return isSameDay(_selectedDay, day);
      },
      onDaySelected: (selectedDay, focusedDay) {
        setState(() {
          _selectedDay = selectedDay;
          _focusedDay = focusedDay;
        });

        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => DetailScreen(selectedDay:
selectedDay),
          ),
        );
      },
    ),
  );
}
```

รูปภาพที่ 4.22 ตัวอย่างโค้ด Calendar

4.6.2 Calendar detail

ในส่วนนี้จะป็นหน้าบันทึกตารางออกกำล้งกายโดยให้เลือก สัดส่วนร่างกาย และ ทำออออกำล้งกายโดยใช้การจัดหมวดหมู่ของสัดส่วนและทำออออกำล้งกาย

```
List<String> getPostures(String bodyPart) {
    if (bodyPart == 'แขน') {
        return [
            'Dumbbell Biceps Curl',
            'Dumbbell Hammer Curl',
            'Dumbbell Overhead Extension',
            'Dips',
            'Dumbbell Tricep Kick Back'
        ];
    } else if (bodyPart == 'ขา') {
        return [
            'Forward lunge',
            'Back Lunge',
            'Side Leg Lifts',
            'Squat',
            'Donkey Kicks'
        ];
    } else if (bodyPart == 'ไหล่') {
        return [
            'Dumbbell Overhead Press',
            'Dumbbell Lateral Raise',
            'Dumbbell Front Raise',
            'Dumbbell High Pull'
        ];
    }
}
```

รูปภาพที่ 4.23 ตัวอย่างโค้ด Detail Calendar

```

body: Padding(
  padding: const EdgeInsets.all(8.0),
  child: ListView(
    children: [
      buildDropDownRow(
        '1:',
        selectedBodyPart1,
        selectedPosture1,
        (newValue) {
          setState(() {
            selectedBodyPart1 = newValue ?? '';
            selectedPosture1 =
              '';
          });
        },
        (newValue) {
          setState(() {
            selectedPosture1 = newValue ?? '';
          });
        },
      ),
      buildDropDownRow(
        '2:',
        selectedBodyPart2,
        selectedPosture2,
        (newValue) {
          setState(() {
            selectedBodyPart2 = newValue ?? '';
            selectedPosture2 =
              '';
          });
        },
      ),
    ],
  ),
)

```

รูปภาพที่ 4.24 ตัวอย่างโค้ด Detail Calendar ต่อที่ 2

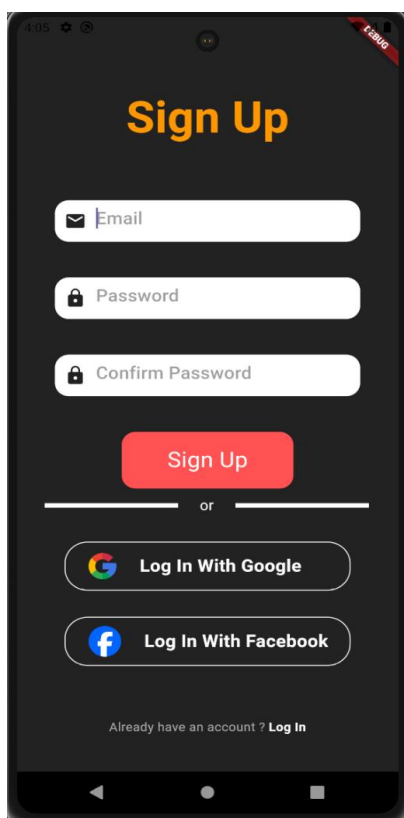
บทที่ 5

การทดสอบระบบ

5.1 การทดสอบสมรรถนะ

ตารางที่ 5.1 แสดงการสมรรถนะของผู้ใช้งาน

กรณีทดสอบ	เงื่อนไขทดสอบ	ผลที่คาดหวัง	ผลการทดสอบ
กรอกข้อมูลไม่ครบ	ผู้ใช้งานกรอกข้อมูลไม่ครบถ้วน	ระบบไม่ทำการบันทึกข้อมูลและแสดงการแจ้งเตือน “please enter your email”	ผ่าน
กรอกรหัสผ่านไม่ตรงกัน	ผู้ใช้งานกรอกยืนยันรหัสผ่านไม่ตรงกับรหัสผ่าน	ระบบไม่ทำการบันทึกข้อมูลและแสดงการแจ้งเตือน “passwords do not match”	ผ่าน
กรอกข้อมูลครบถ้วน	ผู้ใช้งานกรอกข้อมูลครบถ้วน	ระบบทำการบันทึกข้อมูล	ผ่าน

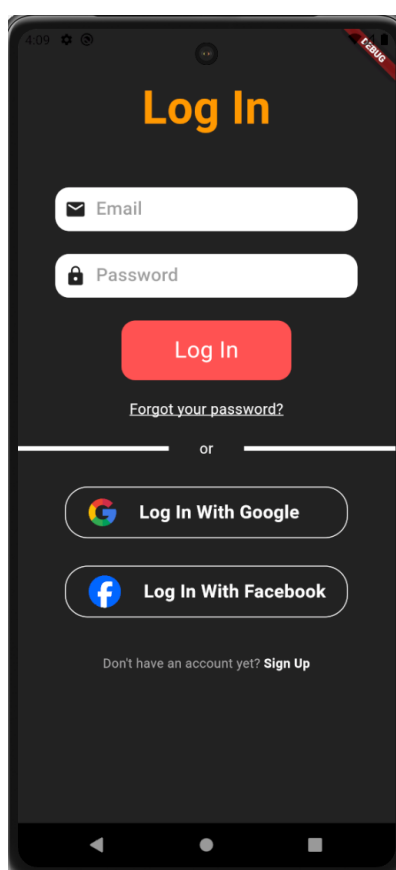


รูปภาพที่ 5.1 แสดงการทดสอบหน้า Sign Up

5.2 การทดสอบเข้าสู่ระบบ

ตารางที่ 5.2 แสดงการเข้าสู่ระบบบัญชีผู้ใช้

กรณีทดสอบ	เงื่อนไขทดสอบ	ผลที่คาดหวัง	ผลการทดสอบ
กรอกข้อมูลบัญชีผู้ใช้ ถูกต้อง	ผู้ใช้งานกรอกข้อมูลไม่ครบถ้วน	ระบบไม่ทำการบันทึกข้อมูลและแสดงการแจ้งเตือน “please enter your email”	ผ่าน
กรอกรหัสผ่านไม่ตรงกัน	ผู้ใช้งานกรอกอีเมล ไม่ตรงกับรหัสผ่าน	ระบบไม่ทำการบันทึกข้อมูลและแสดงการแจ้งเตือน “passwords do not match”	ผ่าน
กรอกข้อมูลครบถ้วน	ผู้ใช้งานกรอกข้อมูลครบถ้วน	แสดงหน้า Exercise	ผ่าน

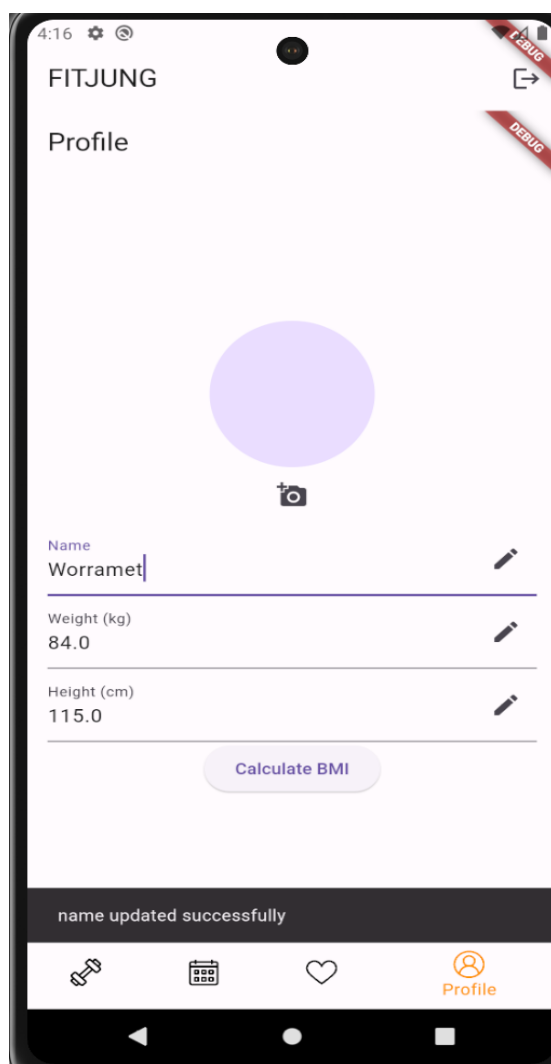


รูปภาพที่ 5.2 แสดงการทดสอบหน้า Log In

5.3 การทดสอบเข้าสู่ระบบ

ตารางที่ 5.3 แสดงการข้อมูลโปรไฟล์

กรณีทดสอบ	เงื่อนไขทดสอบ	ผลที่คาดหวัง	ผลการทดสอบ
กรอกข้อมูลโปรไฟล์	ผู้ใช้งานกรอกข้อมูลข้อมูลส่วนตัวของผู้ใช้	ระบบบันทึกข้อมูล	ผ่าน
กคคำนวณค่า BMI	หลังจากกรอกข้อมูลส่วนสูงและ น้ำหนักให้ผู้ใช้งานกดปุ่ม Calculate BMI	ระบบคำนวณแสดงค่า BMI	ผ่าน

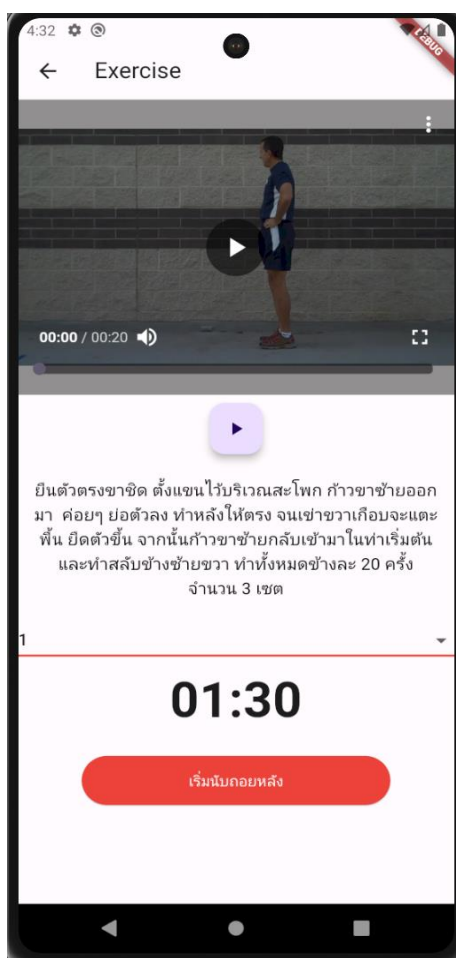


รูปภาพที่ 5.3 แสดงการทดสอบหน้า Profile

5.4 การทดสอบดูวิธีออกกำลังกาย

ตารางที่ 5.4 แสดงการเล่นออกกำลังกาย

กรณีทดสอบ	เงื่อนไขทดสอบ	ผลที่คาดหวัง	ผลการทดสอบ
เลือกสัดส่วนออกกำลังกาย	กดปุ่มสัดส่วนออกกำลังกาย	แสดงท่าออกกำลังกาย	ผ่าน
เลือกท่าออกกำลังกาย	กดปุ่มท่าออกกำลังกาย	แสดงหน้าออกกำลังกาย	ผ่าน
กดเล่นวิดีโอ	กดปุ่มเล่นวิดีโอ	วิดีโอทำงาน	ผ่าน
เลือกจำนวนเซต	กดเลือกเซตจาก drop down list	แสดงเวลาที่ถูกต้อง	ผ่าน
จับเวลานับถอยหลัง	กดปุ่มนับเวลาถอยหลัง	เวลานับถอยหลัง	ผ่าน

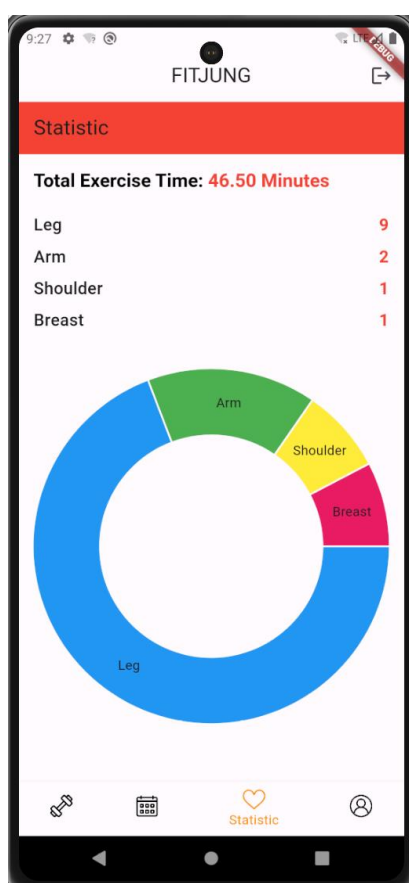


รูปภาพที่ 5.4 แสดงการทดสอบหน้า ออกกำลังกาย

5.5 การทดสอบการทดสอบการออกกำลังกาย

ตารางที่ 5.5 แสดงการทดสอบการออกกำลังกาย

กรณีทดสอบ	เงื่อนไขทดสอบ	ผลที่คาดหวัง	ผลการทดสอบ
การทดสอบจำนวนเวลาการเล่น	กดปุ่มนับถอยหลังในหน้าจอออกกำลังกาย	ผลรวมของเวลาการเล่นออกกำลังกายเพิ่มขึ้นตามจำนวนเวลาที่กดเล่น	ผ่าน
การทดสอบการเพิ่มจำนวนการเล่นของสัดส่วนร่างกาย	กดปุ่มทำออกกำลังกายท่า Dips เป็นร่างกายส่วนแขน	สถิติแขนเพิ่มขึ้น 1	ผ่าน
การทดสอบของ Pie chart	การทดสอบการออกกำลังกายของสัดส่วนร่างกาย	Pie Chart แสดงข้อมูลเป็นสัดส่วนที่ถูกต้อง	ผ่าน



รูปภาพที่ 5.5 แสดงการทดสอบหน้า สถิติการออกกำลังกาย

5.6 การทดสอบบันทึกข้อมูลออกกำลังกาย

ตารางที่ 5.6 แสดงการบันทึกออกกำลังกาย

กรณีทดสอบ	เงื่อนไขทดสอบ	ผลที่คาดหวัง	ผลการทดสอบ
เลือกวันที่ที่ต้องการ	คลิกวันที่ที่เลือก	แสดงหน้า Detail calendar	ผ่าน
กดเลือกสัดส่วนร่างกาย	คลิกสัดส่วนที่ต้องการจาก drop down	ระบบบันทึกข้อมูล	ผ่าน
กดเลือกท่าออกกำลังกาย	คลิกท่าออกกำลังกายที่ต้องการจาก drop down	ระบบบันทึกข้อมูล	ผ่าน



รูปภาพที่ 5.6 แสดงการทดสอบหน้า บันทึกออกกำลังกาย

บทที่ 6

บทสรุป

6.1 บทสรุป

ผู้จัดทำได้บรรลุตามวัตถุประสงค์ในการพัฒนาแอปพลิเคชันออกกำลังกายที่มีการทำงานครบตามขอบเขตของงานที่กำหนดไว้ โดยแอปพลิเคชันออกกำลังกาย สามารถกดดูท่าออกกำลังกายในแต่ละท่าได้ ในหน้าแสดงหน้าจอของหน้าออกกำลังกาย จะมีการอธิบายวิธีการออกกำลังกายรวมถึงวิดีโอให้ผู้ใช้งานได้ศึกษาทำตามได้ รวมถึงฟังก์ชันการบันทึกตารางการออกกำลังกายล่วงหน้าในแต่ละวัน การดูสถิติการออกกำลังกายของผู้ใช้งาน และการกรอกข้อมูลส่วนตัวในหน้าของโปรไฟล์ โดยจะมีฟังก์ชันคำนวณ BMI ของผู้ใช้งาน

นอกจากนี้การพัฒนาแอปพลิเคชันออกกำลังกายทำให้ผู้จัดทำมีความรู้ความเชี่ยวชาญมากขึ้นเกี่ยวกับการพัฒนาแอปพลิเคชันโดยใช้ Flutter, Dart และการใช้ Firebase เป็นฐานข้อมูล ทำให้ผู้จัดทำมีความรับผิดชอบต่องานที่ทำมากขึ้น รวมถึงการออกแบบวางแผนการทำงาน และการแก้ไขปัญหา ในระหว่างการทำงานของผู้จัดทำเอง ซึ่งเป็นการเพิ่มทักษะหลายด้านทำให้ผู้จัดทำดำเนินงานจนได้อย่างราบรื่น และทำให้งานออกมาสมบูรณ์

6.2 ข้อจำกัดของระบบ

แอปพลิเคชันออกกำลังกายที่ผู้จัดทำได้พัฒนาขึ้นมาในตัวคลิพวิดีโอ และคำอธิบาย ยังเป็นตัวอย่างการออกกำลังกายเบื้องต้น ซึ่งยังไม่สามารถให้ผู้ใช้งานใช้งานได้อย่างมีประสิทธิภาพ

6.3 ปัญหาและข้อเสนอแนะ

เนื่องจากเครื่องมือและเทคโนโลยีที่ใช้ ได้แก่ Flutter, Dart และ Firebase นั้นเป็นเครื่องมือ และเทคโนโลยีค่อนข้างใหม่ และยังมีการพัฒนาอยู่เสมอ ทำให้ผู้จัดทำยังไม่มีประสบการณ์พอในการใช้เครื่องมือ และเทคโนโลยีดังกล่าว ซึ่งต้องใช้ระยะเวลานานในการศึกษา และทำความเข้าใจ

ในระหว่างกระบวนการพัฒนาแอปพลิเคชันมักพบปัญหาในการเขียนโปรแกรมเนื่องจากไม่เข้าใจไวยากรณ์ของภาษา และการทำงานของเครื่องมือมากพอ ทำให้พบข้อผิดพลาดในการทำงานอยู่บ่อยครั้ง และเนื่องจากผู้จัดทำไม่มีความรู้ความเชี่ยวชาญเกี่ยวกับ Firebase ที่ใช้เก็บข้อมูลของระบบ ทำให้เกิดปัญหาในการจัดเก็บข้อมูลจากระบบลงในฐานข้อมูล และการดึงข้อมูลจากฐานข้อมูลมาแสดงในระบบ

6.4 แนวทางการพัฒนาต่อยอด

จากการวิเคราะห์ระบบแอปพลิเคชันที่ผู้จัดทำได้พัฒนาขึ้น ทำให้พบข้อจำกัดในการทำงานหลาย ๆ อย่างในแอปพลิเคชัน ผู้จัดทำจึงมีความคิดในการพัฒนาต่อยอดแอปพลิเคชัน เพื่อให้รับรองความต้องการของผู้ใช้ได้มากขึ้น ดังนี้

1. ฟังก์ชันเลือกจำนวนเซต ทางคณะผู้จัดทำได้ทำการเซตค่าเวลา เป็นเวลาเดียวกันกับทุกท่าออกกำลังกาย โดยส่วนนี้ยังสามารถพัฒนาให้ ท่าออกกำลังกายแต่ละท่ามีเวลาที่ต่างกันได้เพื่อให้ใกล้เคียงกับความเป็นจริงมากที่สุด
2. การบันทึกข้อมูลออกกำลังกาย ยังสามารถแสดงข้อมูลที่ได้ทำการบันทึกไว้ด้านล่างของหน้าปฏิทิน ณ วันปัจจุบันได้
3. สามารถเพิ่มฟังก์ชันแนะนำการออกกำลังกาย และแจ้งเตือนเมื่อผู้ใช้ออกกำลังกายหนักมากเกินไปให้ผู้ใช้งานได้

บรรณานุกรม

- [1] (2560). รู้จักกับ Visual Studio Code (วิชวล สตูดิโอ โค้ด) โปรแกรมฟรีจากค่ายไมโครซอฟท์. [ออนไลน์]. สืบค้นจาก: <https://www.mindphp.com/บทความ/microsoft/4829-visual-studio-code.html>. [29 มีนาคม 2567]
- [2] Amazon Web Services. Flutter คืออะไร. [ออนไลน์]. สืบค้นจาก: <https://www.aws.amazon.com/th/what-is-flutter/>. [29 มีนาคม 2567]
- [3] Sirawit. (2560). ทำความรู้จัก Firebase และ ผลิตภัณฑ์ต่าง ๆ ในช่วงต้นปี 2019 กัน. [ออนไลน์]. สืบค้นจาก: <https://www.mindphp.com/@sirawat/firebase-คืออะไร-ทำความรู้จัก-firebase-ในช่วงต้นปี-2019-กัน-473a8e8699fb>. [29 มีนาคม 2567]
- [4] (2563). Android Studio แอนดรอยด์ สตูดิโอ คืออะไร. [ออนไลน์]. สืบค้นจาก: <https://www.mindphp.com/คู่มือ/73-คืออะไร/3505-android-studio.html>. [29 มีนาคม 2567]
- [5] (2563). Dart 101: ทำความรู้จักภาษา Dart ฉบับโปรแกรมเมอร์. [ออนไลน์]. สืบค้นจาก: <https://www.tamemo.com/post/172/dart-101-imtro/>. [29 มีนาคม 2567]

ภาคผนวก ก

การติดตั้งระบบ

1. การติดตั้ง Flutter SDK

1.1 เข้าเว็บไซต์ <https://flutter.dev/>

1.2 ทำตาม Get started ส่วนของเอกสารเบื้องต้น เพื่อดาวน์โหลด Flutter SDK ตามประจำปฏิบัติการ

1.3 หลังจากดาวน์โหลด Flutter SDK แล้ว ให้สร้างโฟลเดอร์ src ขึ้นมาและแยกไฟล์ flutter ที่

ดาวน์โหลดมา ใส่ในโฟลเดอร์ src

1.4 ค้นหา Edit environment variables for your account ที่ตัวเครื่อง แล้วสังเกตคำว่า Path ที่ช่อง User variables โดยกด Edit ที่ช่อง Path แล้วนำ Path `%SystemRoot%\flutter\bin` ของ Flutter SDK ที่ได้ติดตั้ง ไปเพิ่มใน User variables

1.5 ตรวจสอบสถานะของการติดตั้ง Flutter SDK โดยเข้าไปที่ Command prompt และพิมพ์คำสั่ง `flutter doctor`

2. การติดตั้ง Android Studio

2.1 เข้าเว็บไซต์ <https://developer.android.com/studio>

2.2 กดปุ่ม Download Android Studio

2.3 ติดตั้ง Android Studio

2.4 ติดตั้ง Android SDK โดยเข้าเมนู SDK Manager และเลือกเวอร์ชันของ Android SDK ที่ต้องการติดตั้ง

2.5 ติดตั้ง Flutter plugin โดยเข้าเมนู Setting และกดคำว่า Plugin ที่อยู่บนแถบซ้ายของหน้าจอ แล้วทำการค้นหา Flutter plugin และคลิก Download

2.6 ตั้งค่า Path ของ Android SDK โดยเข้าเมนู Edit environment variables for your account ที่ตัวเครื่อง

1. กดปุ่ม New ที่ช่อง User variables

2. ใส่คำว่า `ANDROID_SDK_ROOT` ที่ช่อง Variable name

3. ใส่ Path ของ Android SDK ข้างต้น Variable value เช่น

`C:\Users\Atawee\AppData\Local\Android\Sdk`

4. กด Edit ข้างต้น Path และ New Path ของ Android SDK เช่น

`C:\Users\Worramet\AppData\Local\Android\Sdk\tools` และ

`C:\Users\Worramet\AppData\Local\Android\Sdk\tools\bin` และ

`C:\Users\Worramet\AppData\Local\Android\Sdk\platform-tools`

2.7 ตั้งค่า Android license โดยเข้าไปที่ Command prompt และพิมพ์คำสั่ง `flutter doctor --android-licenses`

2.8 ตรวจสอบสถานะของการติดตั้ง Android Studio, Android SDK และ Flutter plugin โดยเข้าไปที่ Command prompt และพิมพ์คำสั่ง `flutter doctor`

2.9 สร้าง Virtual devices

1. เข้าไปที่โปรแกรม Android Studio
2. เข้ามายัง AVD Manager
3. กด Create Virtual Device โดยสร้างตามขั้นตอนของ Android Studio
4. ตรวจสอบการเชื่อมต่อ Virtual device โดยเข้าไปที่ Command prompt และพิมพ์คำสั่ง

flutter doctor

3. การติดตั้ง Visual Studio Code

3.1 เข้าเว็บไซต์ <https://code.visualstudio.com/>

3.2 กดปุ่ม Download

3.3 ติดตั้ง Visual Studio Code

3.4 เข้าไปโปรแกรม Visual Studio Code และค้นหา Dart Extension และติดตั้ง Extension

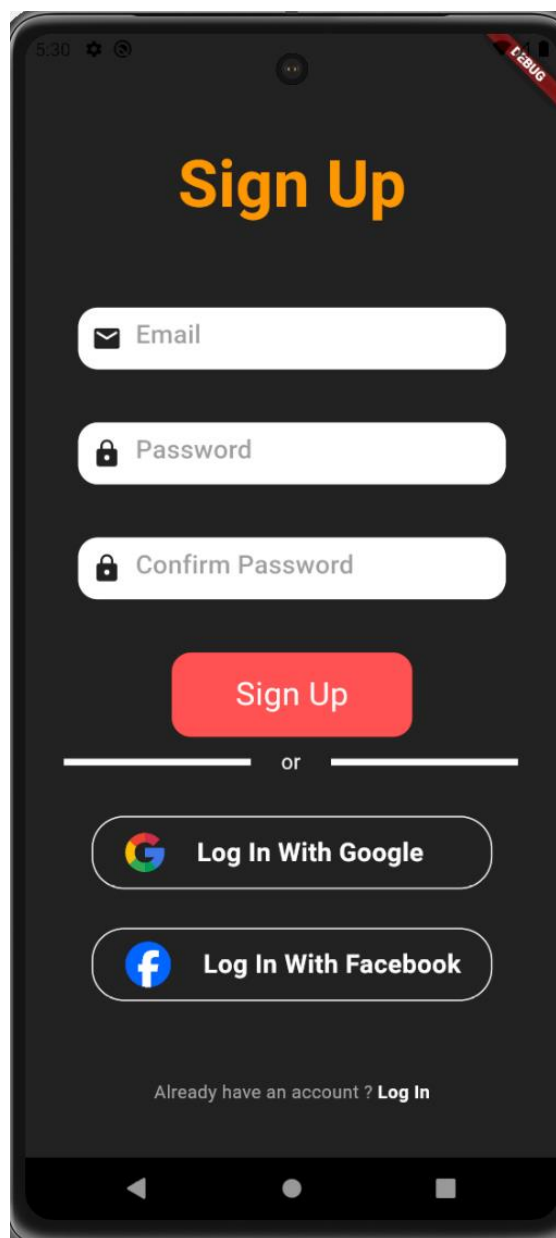
3.5 ตรวจสอบสถานะของการติดตั้ง Visual Studio Code และ Dart Extension โดยเข้าไปที่ Command prompt และพิมพ์คำสั่ง flutter doctor

ภาคผนวก ข

การใช้งานระบบ

1. หน้า Sign Up

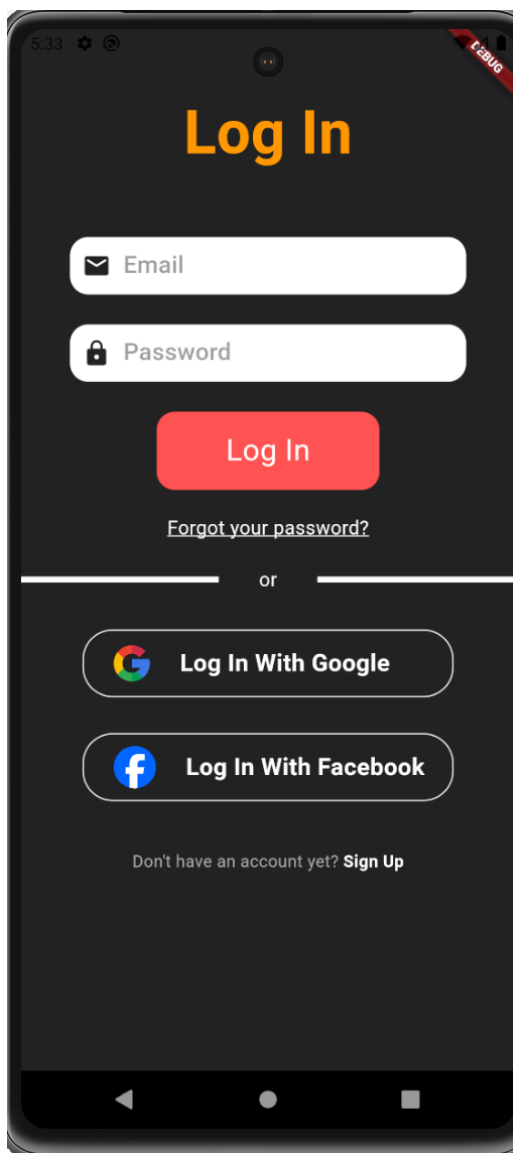
เมื่อผู้ใช้เปิดแอปพลิเคชัน จะพบหน้าจอ Sign Up ให้ผู้ใช้งานได้สมัครบัญชีผู้ใช้ จากนั้นให้ผู้ใช้งานกรอกข้อมูลให้ครบถ้วน หรือถ้าหากผู้ใช้งานต้องการ Log In ให้ผู้ใช้งานกด Log In ด้านล่าง



รูปภาพที่ ข.1 ภาพการใช้งานระบบ Sign Up

2. หน้า Log in

ให้ผู้ใช้งานกรอกข้อมูลบัญชีของผู้ใช้งานที่ได้ลงสมัคร หรือถ้าหากผู้ใช้งานยังไม่มีบัญชีให้ผู้ใช้งานกด Sign Up เพื่อเข้าสู่หน้าสมัคร

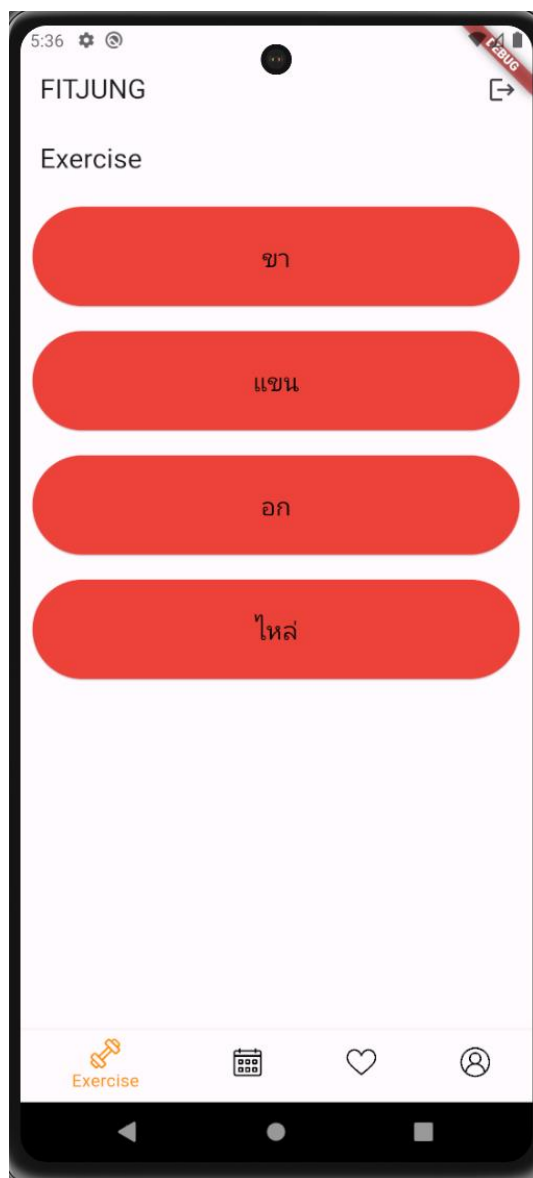


รูปภาพที่ ข.2 ภาพการใช้งานระบบ Log In

3. วิธีออกกำลังกาย

3.1 Exercise Screen

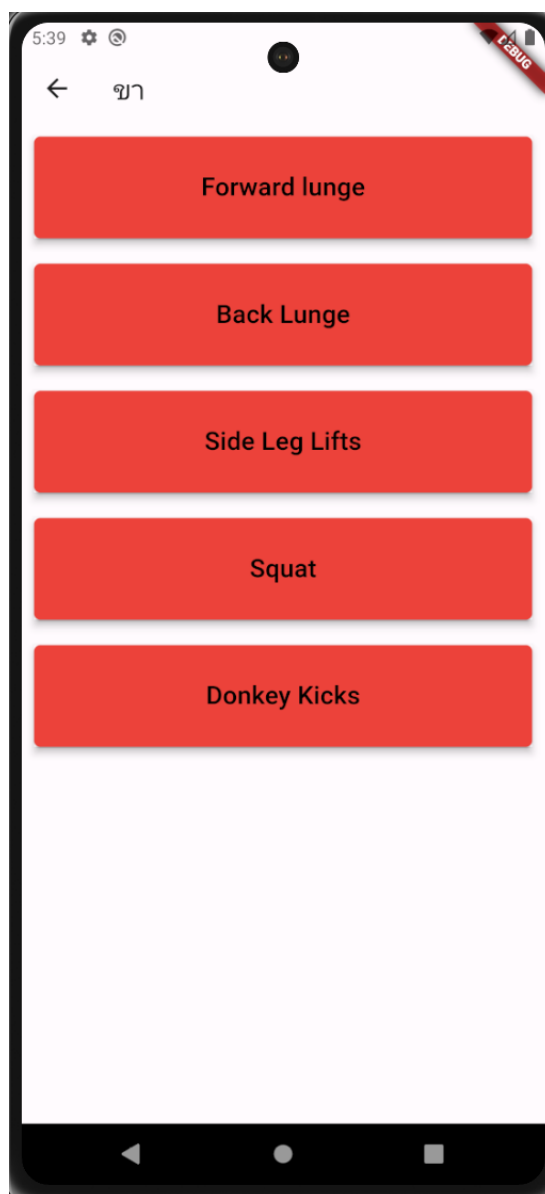
เมื่อผู้ใช้งานเข้าสู่ระบบสำเร็จระบบจะแสดงหน้า Exercise ผู้ใช้งานสามารถกดเลือกสัดส่วนท่าออกกำลังกายที่ผู้ใช้งานต้องการ



รูปภาพที่ ข.3 ภาพการใช้งานระบบ Exercise

3.2 body Screen

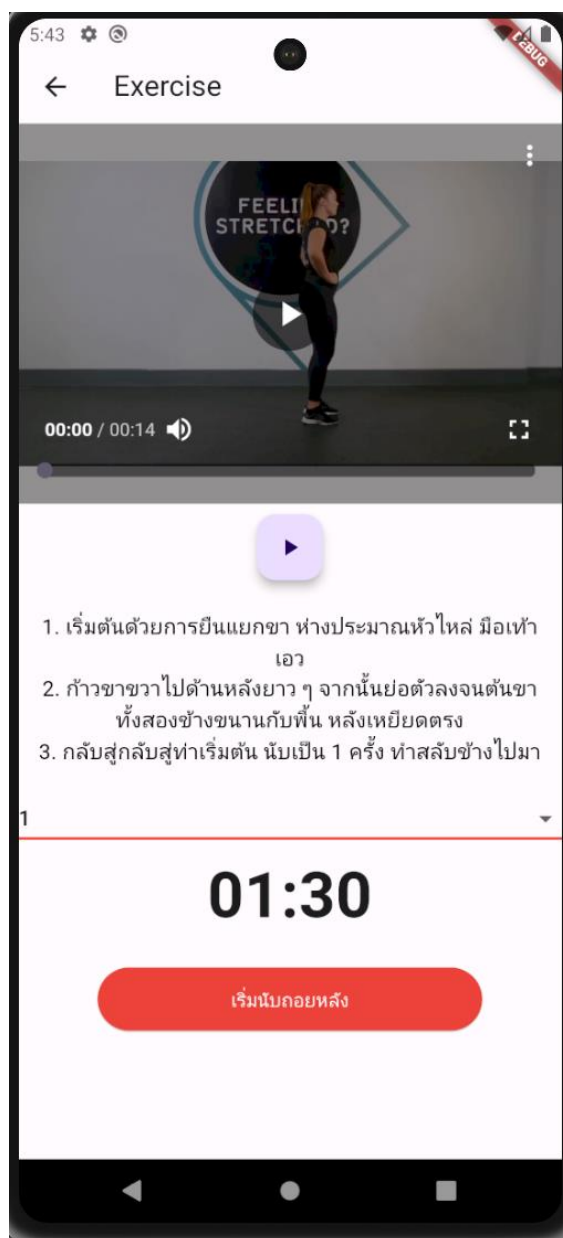
เมื่อผู้ใช้งานเลือกสัดส่วนร่างกายที่ต้องการเรียบร้อย ระบบจะแสดงหน้า body ผู้ใช้งานสามารถกดเลือกท่าออกกำลังกายที่ผู้ใช้งานต้องการ



รูปภาพที่ ข.4 ภาพการใช้งานระบบ body Screen

3.3 Info Posture Screen

เมื่อผู้ใช้งานเลือกท่าออกกำลังกาย ระบบจะแสดงวิดีโอ และ คำอธิบายท่าออกกำลังกาย รวมถึง การปรับจำนวนเซต เมื่อผู้ใช้งานเลือกจำนวนเซตเรียบร้อยแล้ว ให้ผู้ใช้งานกดนับถอยหลัง และออกกำลังกาย ระบบจะนับเวลาถอยหลังอัตโนมัติ

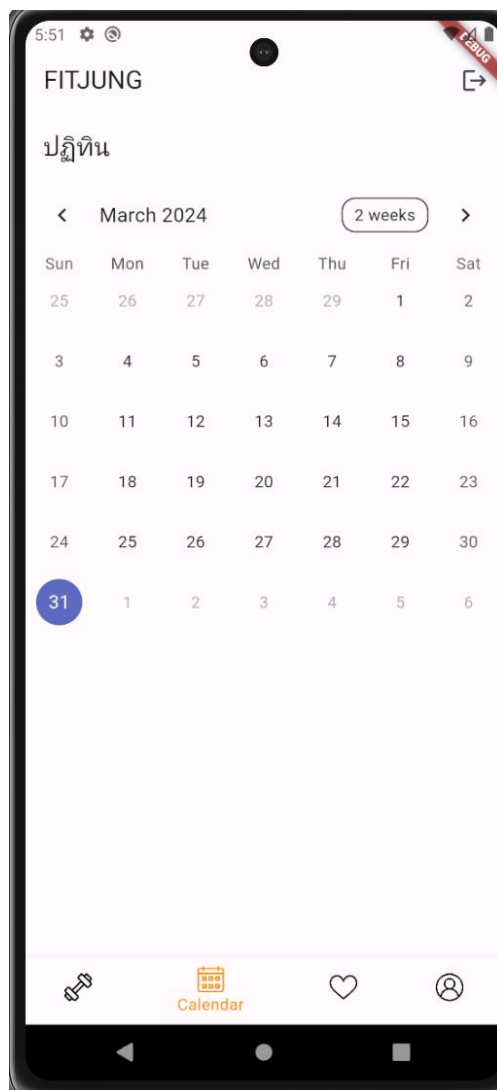


รูปภาพที่ ข.5 ภาพการใช้งานระบบ Info Posture Screen

4. วิธีบันทึกข้อมูลออกกำลังกาย

4.1 Calendar Screen

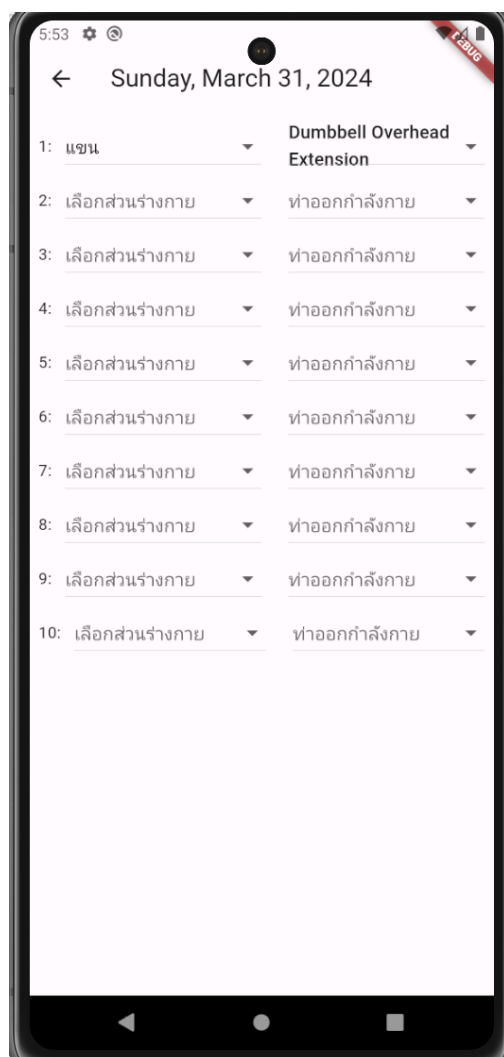
ให้ผู้ใช้งานกดปุ่ม  ด้านล่างเพื่อเข้าสู่หน้าปฏิทินจากนั้นให้ผู้ใช้งานเลือกวันที่ที่ต้องการ



รูปภาพที่ ข.6 ภาพการใช้งานระบบ Calendar Screen


4.2 Calendar Screen

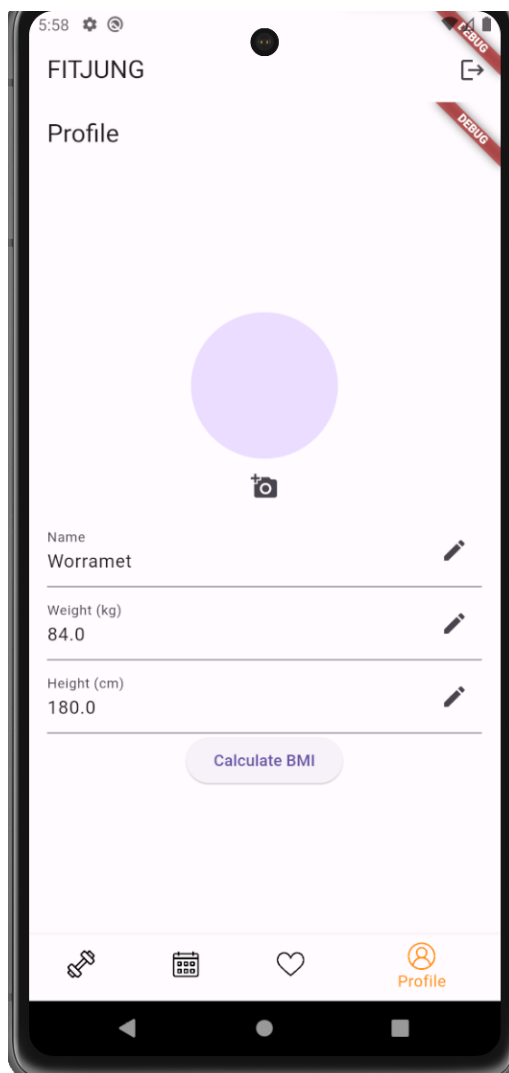
ให้ผู้ใช้งานเลือกบันทึกสัดส่วนร่างกาย และ ทำออกกำลังกายตามความต้องการของผู้ใช้งานระบบจะบันทึกข้อมูลทันที



รูปภาพที่ ข.7 ภาพการใช้งานระบบ Detail Calendar Screen


5. หน้า Profile

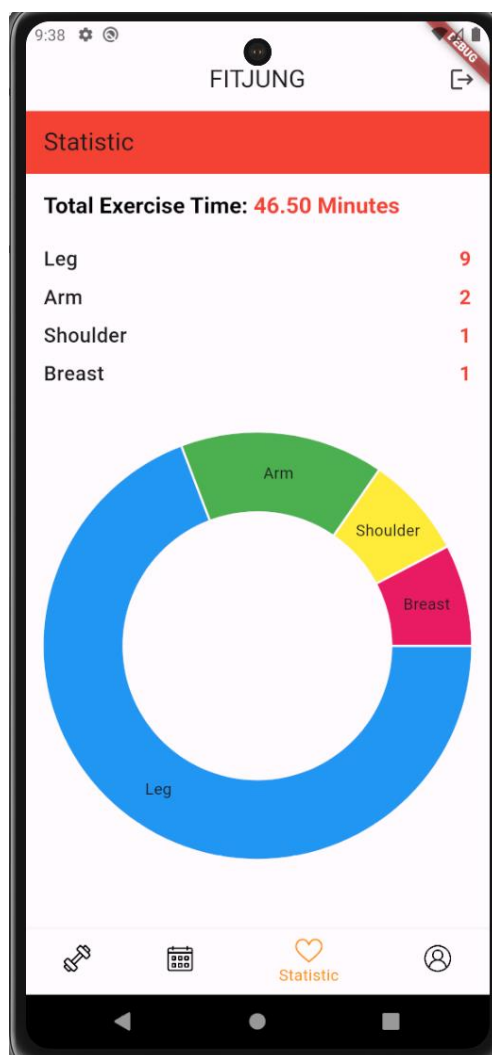
ให้ผู้ใช้งานกดปุ่ม  เพื่อเข้าสู่หน้า Profile ผู้ใช้งานสามารถเพิ่มข้อมูลส่วนตัวของผู้ใช้งานได้ ระบบจะบันทึกข้อมูลอัตโนมัติ



รูปภาพที่ ข.8 ภาพการใช้งานระบบ Profile

6. หน้า Statistic

ให้ผู้ใช้งานกดปุ่ม  เพื่อเข้าสู่หน้า Statistic ผู้ใช้งานสามารถดูสถิติการออกกำลังกายจากหน้า Statistic



รูปภาพที่ ข.9 ภาพการใช้งานระบบ Statistic