

การพัฒนาแอปพลิเคชันติดตามแคลอรีการบริโภคอาหารบนอุปกรณ์พกพา

THE DEVELOPMENT OF CALORIES TRACKER APPLICATION ON MOBILE DEVICE

วีรวิทย์ ชัยยะสมบูรณ์¹

รองศาสตราจารย์ ดร.วราพร จิระพันธุ์ทอง²

บทคัดย่อ

งานวิจัยนี้เป็นการพัฒนาแอปพลิเคชันมือถือ เพื่อให้ผู้ใช้สามารถตรวจสอบรายการอาหารที่บริโภคเข้าไปในแต่ละมื้อ ว่ามีปริมาณแคลอรี ที่มากเกินไปจนเกินความจำเป็นจากปริมาณที่ผู้ใช้สามารถบริโภคในแต่ละวันเข้าไปหรือไม่? รวมทั้งในแอปพลิเคชันนี้ มีข้อมูลแนะนำอาหารสุขภาพ และวิธีการออกกำลังกายให้ผู้ใช้ได้นำไปใช้ได้ด้วย และผู้ใช้อย่างยังสามารถที่จะดูข้อมูลอาหารที่ผู้ใช้ท่านอื่นบริโภคเข้าไปเพื่อเป็นแนวทางของตนเองได้ การพัฒนาแอปพลิเคชันนี้ใช้ SwiftUI framework ต่อกับฐานข้อมูล Firebase (ใช้ Authentication, Cloud Firestore, และ Cloud Storage) มีการนำเสนอ Requirement Analysis ได้นำเสนอความต้องการของระบบทั้ง Functional Requirement และ Non-Functional Requirement แสดง Use Case Diagram/Use Case description, Activity Diagram, Class Diagram, ER Diagram เพื่ออธิบายระบบที่จะสร้างขึ้น รวมทั้งมีการออกแบบ UX/UI ของระบบโดยใช้ Figma ท้ายที่สุดมีการนำเสนอ UI ที่ทำสำเร็จ และ Test cases ต่างๆ

คำสำคัญ: แคลอรี, สุขภาพ, ออกกำลังกาย, แอปพลิเคชันมือถือ, ไอโฟน, iOS, Firebase

ABSTRACT

This research project involves developing a mobile application where users can check the food items they consume at each meal to determine if they exceed their necessary calorie intake for the day. Additionally, the app provides information on healthy food choices and exercise routines for users to incorporate into their lifestyle. Users can also view the food consumption data of other users as a reference for their own consumption habits. The development of this application utilizes the SwiftUI framework integrated with

¹ นักศึกษาหลักสูตรวิศวกรรมเว็บฯ มหาวิทยาลัยธุรกิจบัณฑิต

² อาจารย์ที่ปรึกษาสารนิพนธ์

Firebase database (utilizing Authentication, Cloud Firestore, and Cloud Storage). Requirement Analysis has been conducted, presenting both Functional and Non-Functional Requirements, along with Use Case Diagrams, Use Case descriptions, Activity Diagrams, Class Diagrams, and ER Diagrams to explain the system to be developed. Furthermore, UX/UI design has been crafted using Figma, culminating in the successful presentation of UI and various Test cases.

1. บทนำ

1.1 ที่มาและความสำคัญ

ทุกท่านคงเคยได้ยินประโยคที่กล่าวกันว่า “ความไม่มีโรค เป็นลาภอันประเสริฐ” ไม่ว่าจะยากดีมีเงิน ถ้าไม่ดูแลสุขภาพของตนเองแต่เนิ่น ๆ คนมั่งมีก็อาจจะกลายเป็นคนสิ้นเนื้อประดาตัวก็เป็นได้ การที่เราจะสามารถดูแลร่างกายของเรานั้น นอกจากจะต้องมีการออกกำลังกายอย่างสม่ำเสมอแล้วนั้น อาหารที่เร บริโภคเข้าไปก็มีส่วนไม่มากก็น้อย ดังประโยคที่กล่าวว่า “You are what you eat” นี่จึงเป็นที่มาของการริเริ่ม จัดทำและพัฒนาแอปพลิเคชัน “HealthMe” นี้ขึ้น เพื่อที่ผู้ใช้งานจะสามารถตรวจสอบรายการอาหารที่บริโภคเข้าไปในแต่ละมื้อ ว่ามีปริมาณแคลอรี ที่มากเกินไปจนเกินความจำเป็นจากปริมาณที่ผู้ใช้งานสามารถบริโภคในแต่ละวันเข้าไปหรือไม่? ซึ่งเป็นการตัดไฟตั้งแต่ต้นลม ซึ่งถ้าผู้ใช้งานแอปพลิเคชันนี้ ทำควบคู่กับการออกกำลังกายไปด้วย จึงจะเป็นประโยชน์ไม่มากก็น้อยต่อการดูแลสุขภาพของผู้ใช้เอง อนึ่งในแอปพลิเคชันนี้ มีข้อมูลแนะนำอาหาร สุขภาพ และวิธีการออกกำลังกายให้ผู้ใช้งานได้นำไปใช้ได้ด้วย รวมทั้งผู้ใช้งานสามารถที่จะดูข้อมูลอาหารที่ผู้ใช้งานอื่นบริโภคเข้าไปเพื่อเป็นแนวทางของตนเองได้

1.2 ปัญหา และวัตถุประสงค์

เนื่องด้วยมีแอปพลิเคชันน้อย ที่ใช้ในการติดตามการบริโภคอาหารเข้าไปในแต่ละมื้อว่ามีแคลอรีมากเกินไป ความจำเป็นของผู้ใช้งานในแต่ละวันหรือไม่? รวมทั้งเครื่องมือที่มี Features ต่างๆ ไม่ครบถ้วน ใช้งานยาก และ มีค่าใช้จ่าย ในการแก้ปัญหาดังกล่าว งานวิจัยนี้จึงมีวัตถุประสงค์ดังต่อไปนี้

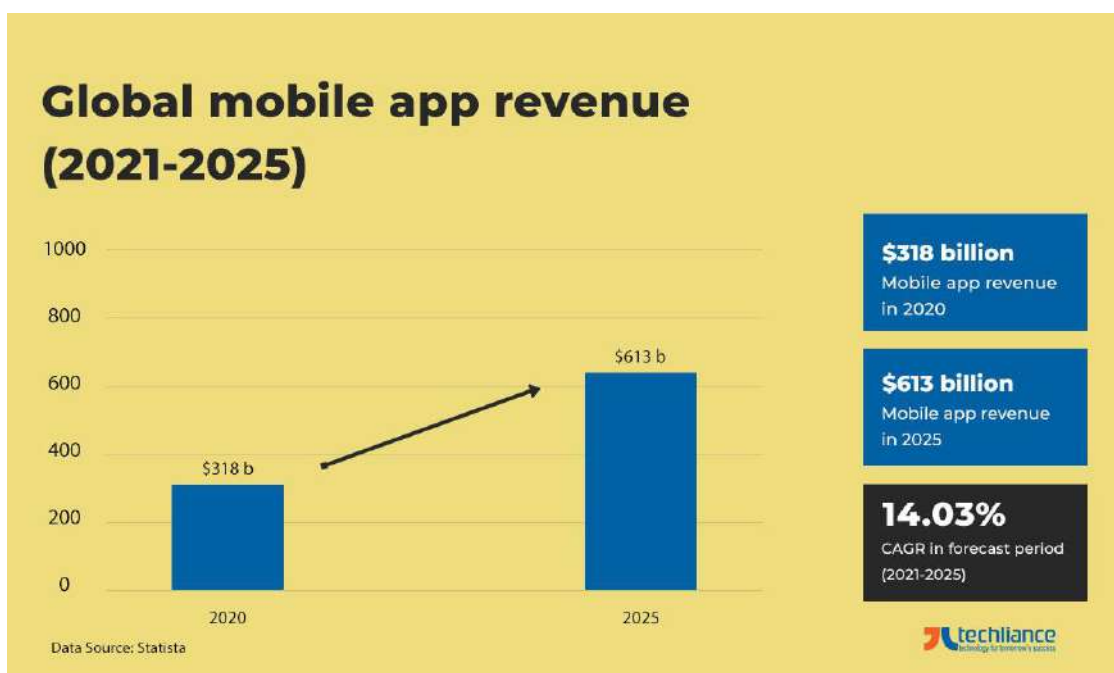
- 1.2.1 พัฒนาเครื่องมือในการติดตามการบริโภคอาหารเข้าไปในแต่ละมื้อว่ามีแคลอรี มากเกินความจำเป็นของผู้ใช้งานในแต่ละวันหรือไม่? อย่างไร?
- 1.2.2 พัฒนาเครื่องมือในรูปแบบแอปพลิเคชันมือถือ โดยต้องสามารถใช้งานง่ายได้กับผู้ทุพพลภาพ และเป็นแอปพลิเคชันที่ใช้งานได้แบบไม่มีค่าใช้จ่าย
- 1.2.3 แอปพลิเคชันนี้สามารถแนะนำรายการอาหารสุขภาพ และแนะนำวิธีในการออกกำลังกายแบบต่างๆ ให้ผู้ใช้งานได้
- 1.2.4 ผู้ใช้งานสามารถดูข้อมูลอาหารของผู้ใช้งานอื่นที่บริโภคไปได้
- 1.2.5 แอปพลิเคชันสามารถเก็บข้อมูลผู้ใช้งานในฐานข้อมูลกลาง ซึ่งผู้ใช้งานสามารถเข้าถึงข้อมูลของตนเองได้โดยข้อมูลไม่มีการสูญหาย ถ้ามีการเปลี่ยนอุปกรณ์มือถือหรือเข้าใช้ในอุปกรณ์เครื่อง

2. แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

การพัฒนาแอปพลิเคชันในงานวิจัยนี้ จะพัฒนาสำหรับอุปกรณ์มือถือ iPhone (iOS platform) เป็นหลัก (โดยใช้ SwiftUI framework ในการพัฒนา) โดยจะมีการต่อกับฐานข้อมูล Firebase ซึ่งในบทนี้ ผู้เขียนจะแสดงเหตุผลว่าทำไมต้องพัฒนาใน Platform นี้

2.1 การพัฒนาแอปพลิเคชันบนอุปกรณ์มือถือในปัจจุบัน

ในปัจจุบันมี Mobile Operating System ในตลาด ใหญ่ ๆ อยู่ 2 ค่าย คือ Andriod Platform และ iOS Platform และถ้าพิจารณาจากภาพที่ 2.1 ด้านล่าง จะเห็นว่ารายรับของ Mobile App จะมีมูลค่า \$613 billion ในปี 2025 นี่จึงเป็นอีกเหตุผลที่ผู้เขียนเลือกที่จะพัฒนาแอปพลิเคชันนี้ในอุปกรณ์พกพาหรือมือถือนั่นเอง (นอกจาก Use cases ที่มีแสดงว่าควรพัฒนาเป็นแอปพลิเคชันมือถือ ซึ่งจะได้กล่าวถึงต่อไป)



รูปที่ 2.1 Global mobile app revenue (2021-2025) [1]

สำหรับเครื่องมือและภาษาที่นิยมใช้ในปัจจุบันมีดังต่อไปนี้

- (1) Android Native platform พัฒนาโดยใช้ภาษา Kotlin/Java
- (2) iOS Native platform พัฒนาโดยใช้ภาษา Swift/Objective-c
- (3) Cross platform พัฒนาโดยใช้ภาษา React-Native
- (4) Cross platform พัฒนาโดยใช้ภาษา Flutter(Dart)
- (5) Cross platform อื่นๆ เช่น Xamarin, Iconic, etc.

แต่สำหรับงานวิจัยนี้จะทำการพัฒนาบน iOS platform โดยใช้ SwiftUI framework ดังที่กล่าวไปข้างต้น โดยจะแสดงผลผลสนับสนุนในหัวข้อต่อ ๆ ไป

2.2 iOS platform

Apple เป็นผู้นำในเรื่องลูกค้าที่ยอมจ่ายซื้อ App มากกว่า Android platform จึงทำให้ผลตอบแทนการลงทุน (ROI – Return On Investment) สำหรับ iOS platform ดีกว่า รวมทั้งในการพัฒนา App สำหรับ Apple platform ผู้พัฒนาเพียงพัฒนาสำหรับ iPhone และ iPad เท่านั้น ซึ่งตั้งแต่ปี 2020, Apple ผลิต Mac ที่เป็น Apple Silicon chip ซึ่งทำให้ App native iOS สามารถทำงานบน Mac ได้ด้วย ซึ่งทำให้การพัฒนา App ใน Apple platform มีต้นทุนต่ำและใช้เวลาน้อยกว่าถ้าเทียบกับ Android platform ที่มีความหลากหลายของอุปกรณ์^[2]

ส่วนเรื่องความปลอดภัย - มีการพยากรณ์ว่าภายในปี 2021 ภัยทางไซเบอร์ (Cybercrime) จะมีมูลค่าถึง 6 พันล้านดอลลาร์ ต่อปี ซึ่งทำให้ผู้ใช้ระวังและกังวลเรื่องความปลอดภัยใน App ที่เรียกใช้ ข้อมูลต่างๆเกี่ยวกับการจ่ายเงินและข้อมูลส่วนบุคคล สำหรับเรื่องเหล่านี้ Apple ขึ้นชื่อมาแต่เนิ่นนานเรื่องความปลอดภัยในอุปกรณ์ต่างๆของตน รวมทั้งผู้พัฒนา App ใน Apple platform ต้องมีการส่ง App ที่ตนพัฒนาเพื่อ Review ก่อน ส่งถึงมือผู้ใช้ ทำให้ผู้ใช้เชื่อมั่นใน App ต่างๆของ Apple platform มากกว่า^[2]

สำหรับการใช้งานและประสบการณ์ผู้ใช้ – Apple มีข้อกำหนดด้านรูปแบบของ App มากกว่า Android ซึ่งทำให้รูปแบบส่วนต่อกับผู้ใช้มีคุณภาพมากกว่า ซึ่ง iOS platform ทำให้ผู้ใช้งานมั่นใจได้ว่า App จะใช้งานได้ง่ายและสร้างประสบการณ์ให้ผู้ใช้ได้เป็นอย่างดี ซึ่งทำให้บริษัทหลายแห่งทำ App ใน Apple platform เพื่อเป็นหน้าเป็นตาให้กับบริษัท^[2]

สุดท้าย Apple เป็นบริษัทที่มีชื่อเสียงระดับโลก ทำให้ Apps ใน App Store มีชื่อเสียงอยู่ในระดับสูงเมื่อเทียบกับ Android apps, iPhone รุ่นแรกทำให้เกิดอุตสาหกรรมใหม่สำหรับ Smartphone รวมทั้งตอนที่ Apple นำเสนอ iPad ทำให้มีการผลิต Tablets ต่างๆออกมาแข่ง จึงกล่าวได้ว่า Apple มีชื่อเสียงในด้านนวัตกรรม ซึ่งผู้ใช้งานทั้งหมดทราบดีสำหรับสิ่งเหล่านี้^[2]

2.3 Native (SwiftUI) เปรียบเทียบ Cross platform

เนื่องด้วยเราพัฒนาแอปพลิเคชันในงานวิจัยนี้ ด้วย Native language (Swift) และ Native framework (SwiftUI) ทำให้ App มีสมรรถนะดีกว่า เร็วกว่า และตอบสนองต่อผู้ใช้ได้ดีกว่าการพัฒนาด้วย Cross platform ที่ต้องแปลง codes เป็น Native รวมทั้งมีความปลอดภัยดีกว่า เพราะ Native app เข้าถึง Built-in security features^[3] รวมทั้ง Native app มี UX มีคุณภาพ ให้ประสบการณ์กับผู้ใช้ได้ดีกว่า เพราะ UI/UX ทำตาม Design guidelines จาก Native OS platform ซึ่งทำให้ App มีความเป็นธรรมชาติในการใช้งานได้มากกว่า และยังสามารถเข้าถึง Hardware/Services ทุกส่วนของ Device ทำให้ App มี Features ทุกอย่างที่มี ซึ่งเพิ่มประสบการณ์ของผู้ใช้^[3] สุดท้าย Native app สร้างด้วย Native SDKs ทำให้การพัฒนามีข้อผิดพลาดที่น้อยกว่าผู้พัฒนาจะได้รับ SDKs ที่มีการแก้ไขและ/หรือเพิ่มคุณสมบัติได้ทันที^[3]

2.4 SwiftUI เทียบกับ UIKit

ก่อนหน้านี้ SwiftUI framework, การพัฒนาแอปพลิเคชันบน iOS Platform ใช้ UIKit framework ในการพัฒนา งานวิจัยนี้เลือกที่จะพัฒนาด้วย SwiftUI เพราะสามารถพัฒนาแอปพลิเคชันได้อย่างง่ายและรวดเร็วมากกว่า UIKit เนื่องด้วย SwiftUI เป็นแบบ Declarative syntax ผู้พัฒนาสามารถกำหนด UI ได้เลยโดยไม่ต้องเขียน Code แบบเป็นลำดับขั้นเหมือน UIKit ^[4]

อีกทั้ง SwiftUI เป็น Apple Cross Platform Devices ผู้พัฒนาสามารถพัฒนาแอปพลิเคชันเพียงครั้งเดียวโดยสามารถ Run ได้ทุก Apple Devices ทำให้ลดเวลาและค่าใช้จ่ายในการพัฒนา ^[4]

อย่างไรก็ตาม ข้อจำกัดของ SwiftUI คือสนับสนุนได้ตั้งแต่ iOS 13 ขึ้นไป และรวมทั้ง SwiftUI เป็น framework ที่ยังคงใหม่อู่ยังไม่ Maturity เหมือน UIKit แต่อย่างไรก็ตามเราสามารถ Integrate UIKit view to SwiftUI ได้ ^[4]

2.5 Firebase platform

Firebase มีคุณสมบัติของ Backend อยู่ในตัว ทำให้เราสามารถเชื่อมต่อ Frontend ไปที่ Firebase ได้โดยตรง ทำให้ลดเวลาในการพัฒนาลงอย่างมาก รวมทั้งไม่ต้องมีการบำรุงรักษาทางด้าน Backend อีกทั้ง Firebase ยังสามารถเพิ่ม Google Analytics ซึ่งสามารถวิเคราะห์และ เก็บข้อมูลของผู้ใช้ ทำให้สามารถสร้าง App ที่ตอบสนองความต้องการของผู้ใช้ได้ดีขึ้น (Personalization) ^[5]

Firebase ทำให้มีการบำรุงรักษาด้าน Frontend ไม่ต้องมีการพัฒนาและบำรุงรักษาด้าน Backend ทำให้ App สามารถออกสู่ตลาดได้เร็ว นอกจากจะลดบุคคลากรด้าน Backend แล้วนั้น ยังลดข้อผิดพลาดต่างๆ (Bugs) ในการพัฒนาด้วย ทำให้มีความคุ้มค่า (Cost-Effectiveness) ^[5]

สำหรับ Firestore database ที่ใช้ในงานวิจัยนี้ เป็น NoSQL Database ซึ่งมีคุณสมบัติที่สามารถเก็บข้อมูลได้จำนวนมากและยังคงมีสมรรถนะที่ดี รวมทั้งมีคุณสมบัติยืดหยุ่นและสามารถขยายฐานข้อมูลได้ (Flexible and Scalable) ^[5]

Firebase มีชุมชนผู้ใช้นขนาดใหญ่ และมี Technical documents ที่เพียงพอ รวมทั้งง่ายต่อการบูรณาการ (Ease of integration) โดยสามารถสร้าง App ได้โดยง่าย มีการ configuration/setup ที่ง่าย ^[5]

สุดท้าย Firebase มี Cloud messaging ทำให้ผู้พัฒนาสามารถทำ Push notification ส่งให้ผู้ใช้ทราบถึงสิทธิประโยชน์/Promotions ต่างๆ ที่มีในเวลานั้นๆ ^[5]

2.6 สรุปเครื่องมือที่ใช้ในการพัฒนา

- (1) ใช้ SwiftUI framework ในการพัฒนา
- (2) ใช้ภาษา Swift ในการพัฒนา
- (3) ใช้ Xcode เป็น IDE ในการพัฒนา
- (4) ใช้ Firebase เป็นระบบฐานข้อมูล (Cloud Firestore/Cloud Storage) รวมทั้งระบบ Firebase Authentication (โดยใช้ Email/Password)

3. ระเบียบวิธีวิจัย

ในบทนี้จะเป็นการหาความต้องการของระบบ (Requirement analysis) โดยเราจะใช้ UML diagrams (Object-oriented modeling) เป็นหลักในการอธิบายระบบที่ผู้เขียนพัฒนา (ระบบที่ผู้เขียนพัฒนาได้ถูกตั้งชื่อว่า HealthMe App)

ซึ่งเราจะแสดงความต้องการของระบบ, Use case diagram, Use case description, Activity diagram, Class diagram, และ ER diagram. รวมทั้งจะแสดงการออกแบบ User Experience/User Interface (UX/UI) อีกด้วย

3.1 ความต้องการของระบบ

ความต้องการมี 2 แบบ คือ ความต้องการตามการใช้งาน (Functional Requirement) ซึ่งก็คือความต้องการที่ระบบที่เราพัฒนาต้องทำได้ มีทั้งหมด 9 ข้อ (ตามตารางที่ 3.1) และความต้องการที่ไม่ใช่ตามการใช้งาน (Non-Functional Requirement) ซึ่งเป็นความต้องการที่สนับสนุนการทำงานของระบบที่เราพัฒนา มีทั้งหมด 2 ข้อ

3.1.1 ความต้องการตามการใช้งาน (Functional Requirement)

ตารางที่ 3.1 ความต้องการตามการใช้งาน (Functional Requirement)

Requirement ID	Description
1	สามารถสมัครสมาชิกได้
2	สามารถคำนวณ BMI/BMR ได้
3	นำเข้ารูปอาหาร และใส่ข้อมูลที่เกี่ยวข้อง (สามารถแก้ไขได้ภายหลัง หรือลบออกได้)
4	สามารถติดตาม (Track) ปริมาณแคลอรีที่บริโภคไป เทียบกับที่แนะนำได้
5	แสดงกราฟปริมาณแคลอรีที่บริโภคไป และกราฟปริมาณแคลอรีที่บริโภคไป 7 วันย้อนหลัง
6	แสดงรายการอาหารสุขภาพที่แนะนำพร้อมรายละเอียด
7	แสดงรายการการออกกำลังกายที่แนะนำพร้อมรายละเอียด
8	แสดงรายละเอียดของผู้ใช้/แก้ไขได้
9	แสดงการสะสมแต้มของผู้ใช้ (Leaderboard)
10	แสดงรายการอาหารของผู้ใช้ทุกคนที่ใส่เข้ามาในระบบ

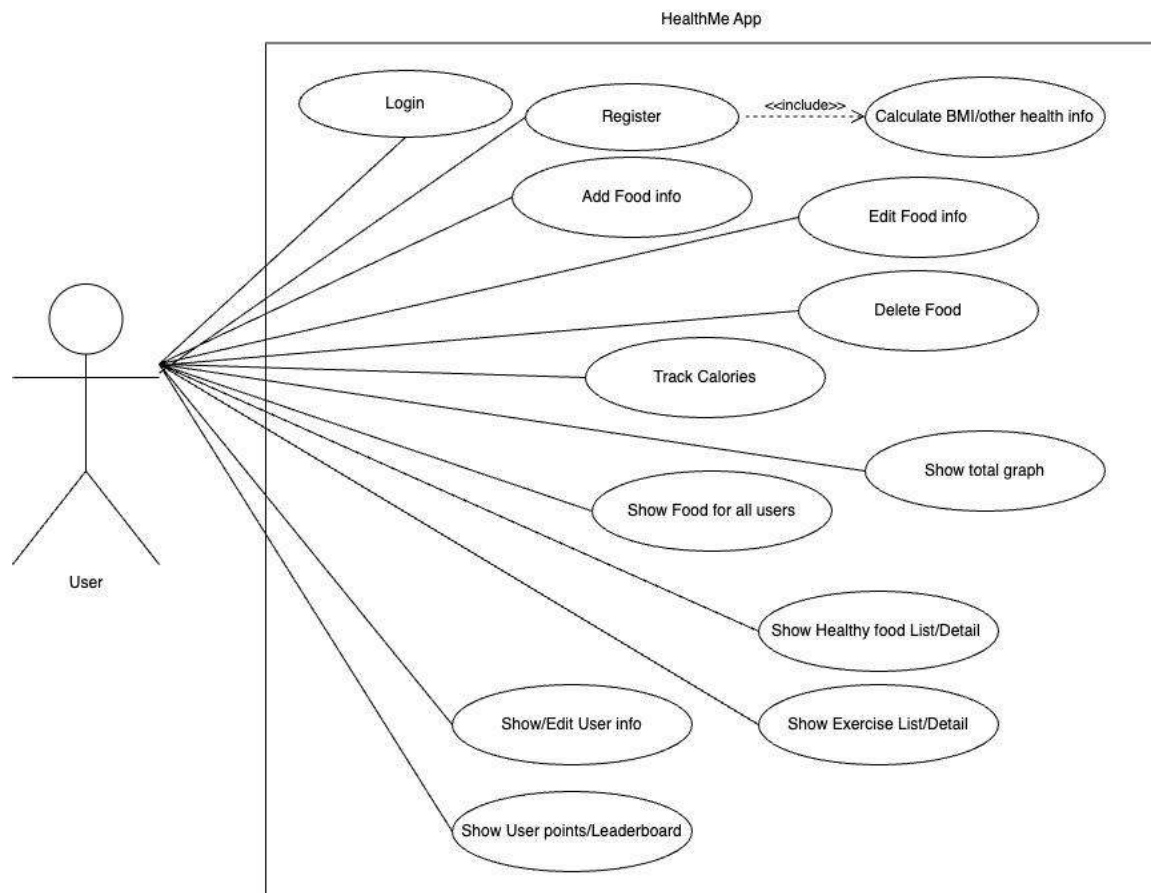
3.1.2 ความต้องการที่ไม่ใช่ตามการใช้งาน (Non-Functional Requirement)

(1) แอปทำงานในระบบ iOS platform (support iPhone only)

(2) เชื่อมต่อระบบ Firebase (Authentication/Cloud Firestore/Storage)

3.2 Use cases

3.2.1 Use case diagram คือ แผนภาพแสดงว่าผู้ใช้สามารถทำอะไรได้บ้างกับระบบที่เราพัฒนา ซึ่งในกรณีนี้มีทั้งหมด 12 Use cases ดังภาพที่ 3.1 ด้านล่าง



รูปที่ 3.1 Use case diagram

3.2.2 Use case description จะเป็นการบรรยายรายละเอียดของแต่ละ Use case ที่มี ซึ่งในกรณีนี้มีทั้งหมด 12 Use cases (แต่ในที่นี้จะยกตัวอย่างเพียง 1 Use case ตามตาราง 3.2)

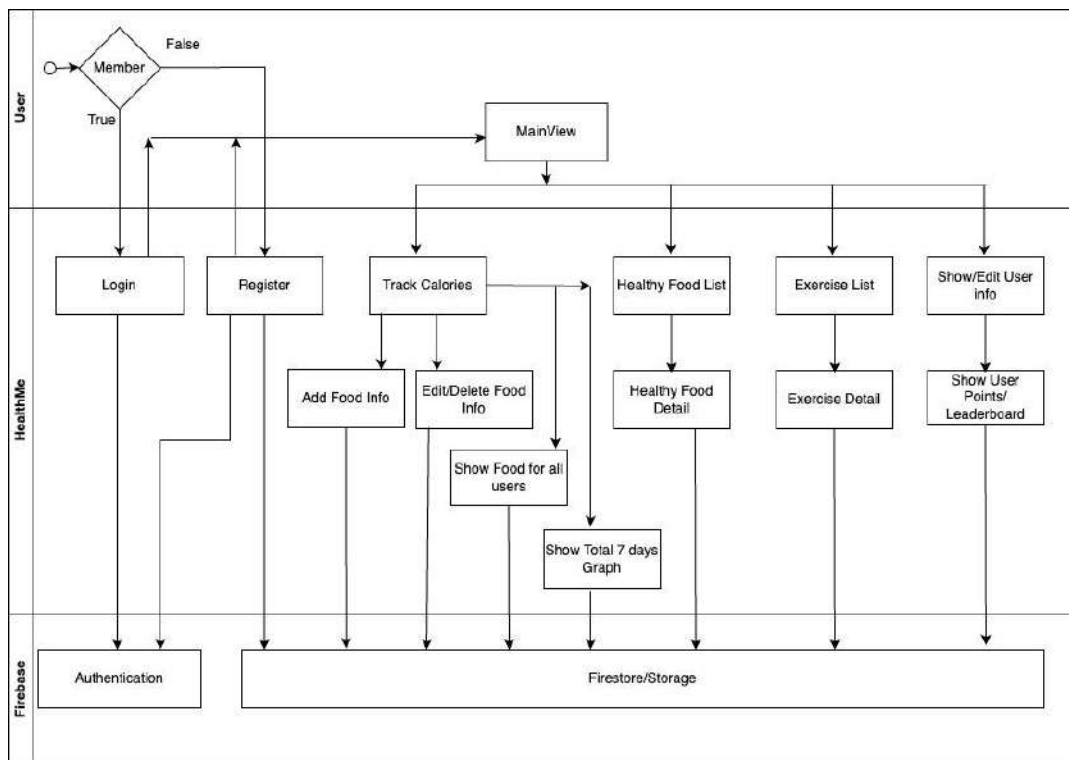
ตารางที่ 3.2 Use case description UC01

Use case number	UC01
Use case name	ลงทะเบียน (Register)
Pre-condition	ผู้ใช้ที่ยังไม่เคยลงทะเบียน
Primary Actor	ผู้ใช้งานแอป
Scenario	1. ผู้ใช้งาน Download App 2. App แสดงหน้า Login และผู้ใช้ยังไม่เคยลงทะเบียนมาก่อน ให้กดปุ่ม Sign Up

	3. App แสดงหน้า Create Account 4. ผู้ใช้กรอกข้อมูลตาม Fields ต่างๆ (App มีการ Validating fields ต่างๆ จะแสดงให้ผู้ใช้นทราบ) 5. ผู้ใช้กดปุ่ม Sign Up ระบบจะสร้างข้อมูลของผู้ใช้ ซึ่งรวมทั้งระบบ Authentication (ใช้ Email/Password ของผู้ใช้) และข้อมูลของผู้ใช้ โดยระบบจะมีการคำนวณค่า BMI/BMR แล้วเก็บในฐานข้อมูล Firebase 6. App จะนำผู้ใช้เข้าสู่หน้าหลักของ App โดยอัตโนมัติ (หน้า Calories Tracker) 7. เมื่อผู้ใช้กลับมาใช้ App อีกครั้ง ระบบจะยังคงให้ผู้ใช้เข้าใช้งานหน้าหลักได้เลย (ระบบมีการเก็บ user session ไว้) นอกจากผู้ใช้ได้ทำการ Sign Out ออกจากระบบ (ปุ่ม Sign Out อยู่ในหน้า Profile) กรณีนี้ App จะนำผู้ใช้ไปสู่หน้า Login ซึ่งผู้ใช้จะต้องทำการ Login ด้วย Email/Password ที่ลงทะเบียนไว้
--	--

3.3 Activity Diagram

เป็นการแสดง Work flow ของระบบที่ผู้ใช้มีการตอบสนองด้วย ซึ่งภาพที่ 3.2 แสดงองค์รวมทั้งหมดของระบบ



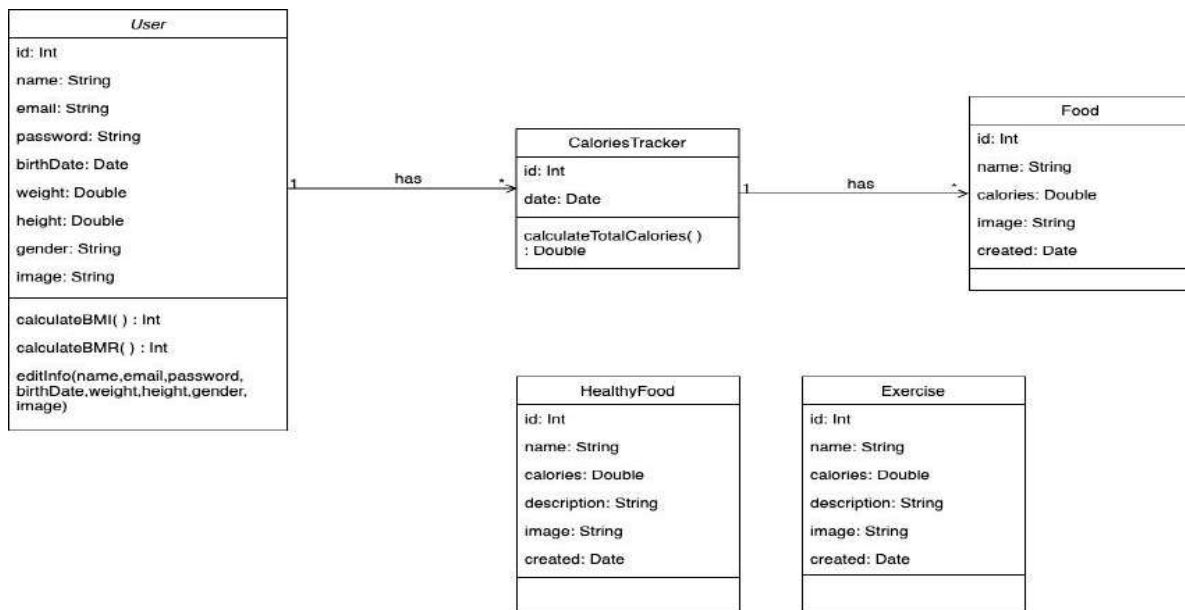
รูปที่ 3.2 Activity Diagram

3.4 Class Diagram

เป็นการแสดงโครงสร้างของระบบ ตามแผนภาพที่ 3.3 ประกอบไปด้วย

- (1) User class
- (2) CaloriesTracker class
- (3) Food class
- (4) HealthyFood class
- (5) Exercise class

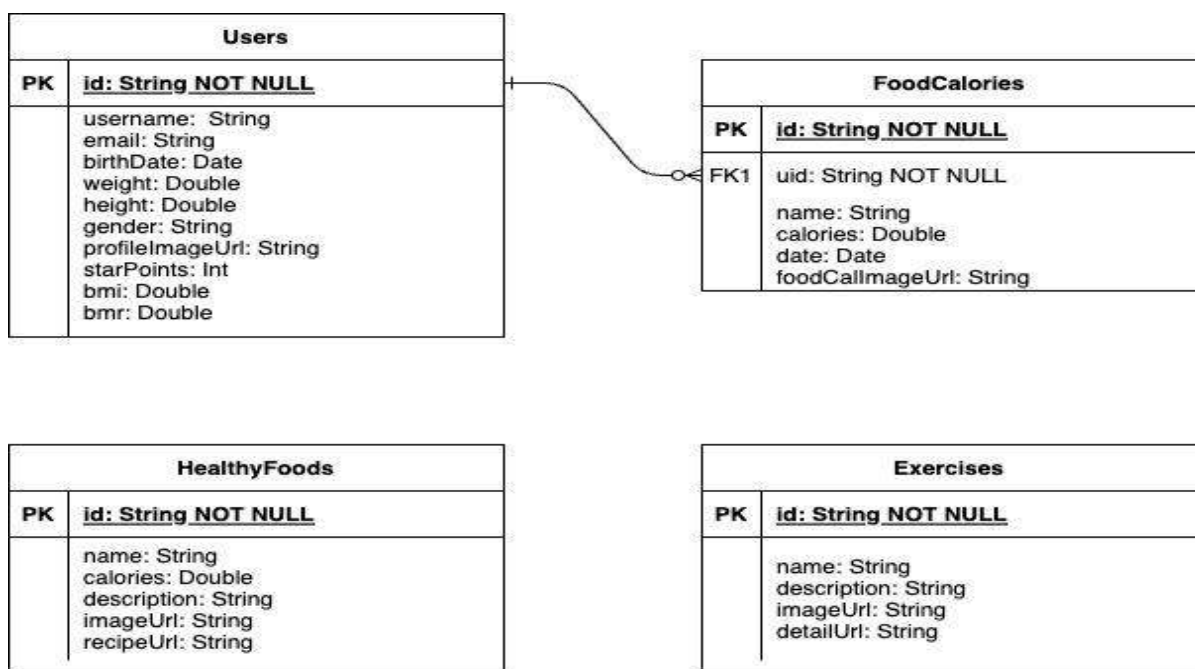
โดย User class มีความสัมพันธ์แบบ One-to-Many กับ CaloriesTracker class ซึ่งมีความสัมพันธ์แบบ One-to-Many กับ Food class ส่วน HealthyFood class กับ Exercise class ไม่มีความสัมพันธ์กับ class ใด ๆ เพราะเป็นเพียงแสดงข้อมูลในระบบเท่านั้น



รูปที่ 3.3 Class Diagram

3.5 ER Diagram

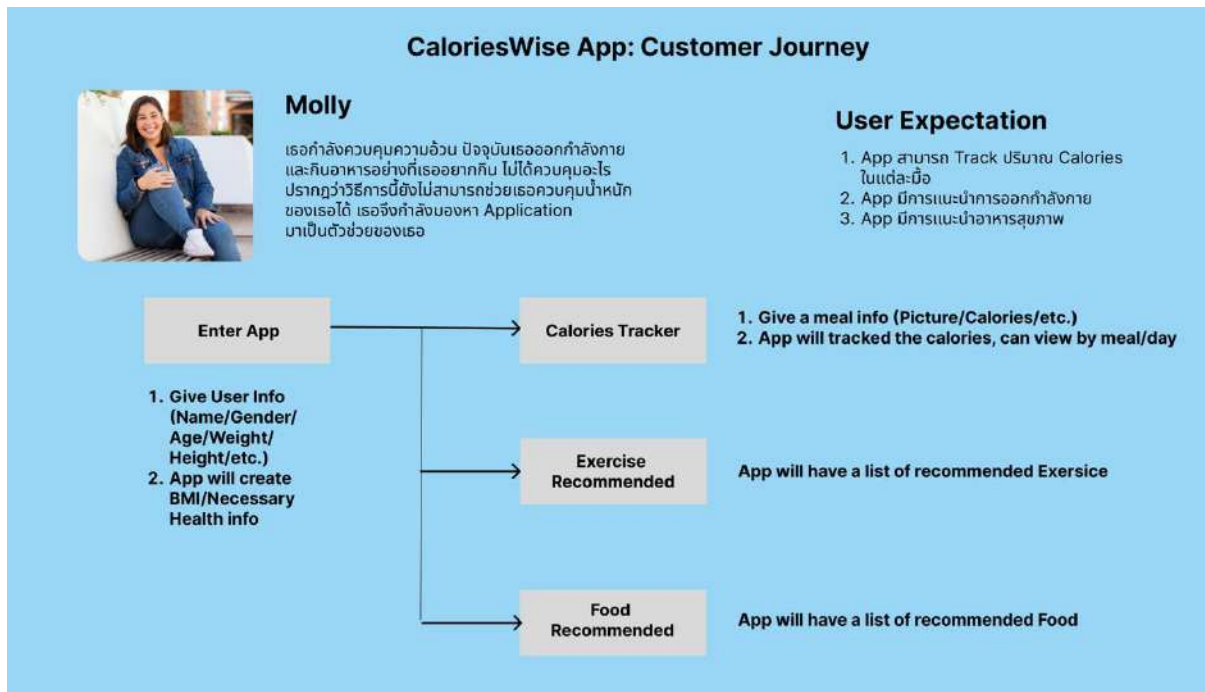
ER diagram หรือ Entity Relationship diagram แสดงโครงสร้างของ Database โดยแสดงเป็น Diagram ดังภาพที่ 3.4 ซึ่งจะเห็นได้ว่า Users Entity มีความสัมพันธ์แบบ One-to-Many กับ FoodCalories Entity สำหรับ HealthyFoods Entity กับ Exercises Entity ไม่มีความสัมพันธ์กับ Entity ใด เนื่องจากเป็นการแสดงข้อมูลในระบบเท่านั้น



รูปที่ 3.4 ER Diagram

3.6 UX/UI design

ในเริ่มแรกผู้เขียน มีการทำ Customer Journey ของ App มีการ Setup Persona ของผู้ใช้ App และ Expectation ของผู้ใช้ ดังแสดงรายละเอียด ตามภาพที่ 3.5 ด้านล่าง (ในขณะนั้นตั้งชื่อว่า CaloriesWise App และมาเปลี่ยนชื่อในภายหลังเป็น HealthMe App)



รูปที่ 3.5 Customer Journey

Persona สำหรับ App นี้ที่วิเคราะห์ไว้นั้นคือ เป็นผู้หญิงวัยทำงาน ที่มีรูปร่างเจ้าเนื้อ แต่ไม่อ้วน ซึ่งต้องการควบคุมน้ำหนัก ซึ่งเธอมีการออกกำลังกายเป็นประจำ แต่ไม่ได้มีการควบคุมอาหารแต่อย่างใด ซึ่งวิธีการออกกำลังกายอย่างเดียวของเธอไม่สามารถทำให้เธอควบคุมน้ำหนักได้ เธอจึงต้องการหาตัวช่วยโดยการหาเครื่องมือให้เธอช่วยในการควบคุมปริมาณอาหารที่เธอบริโภคเข้าไป จึงเกิดเป็นแนวคิดของ App นี้

นั่นเอง

โดย Expectation ของ App นี้คือ

- (1) App สามารถ Track ปริมาณแคลอรี ที่บริโภคอาหารเข้าไปในแต่ละมื้อ
- (2) App มีการแนะนำการออกกำลังกาย
- (3) App มีการแนะนำอาหารสุขภาพ

ส่วน UI design ในเบื้องต้น ได้ทำการออกแบบโดยใช้ Figma ซึ่งมีแนวคิดคร่าวๆ ของ App คือ เป็นแอปที่เป็นกันเองไม่เป็นทางการมากนัก ผู้ใช้ใช้แล้วควรจะรู้สึกสนุกกับการใช้ และอยากกลับมาใช้แอปอยู่ตลอด จึงออกแบบแอปโดยใช้ 3 สี คือ สีเหลือง สีขาว และสีดำ เป็นโทนสีหลักของแอป ยกตัวอย่างตามภาพที่ 3.6 ที่ออกแบบไว้เบื้องต้นใน Figma

Login Screen

Register Screen

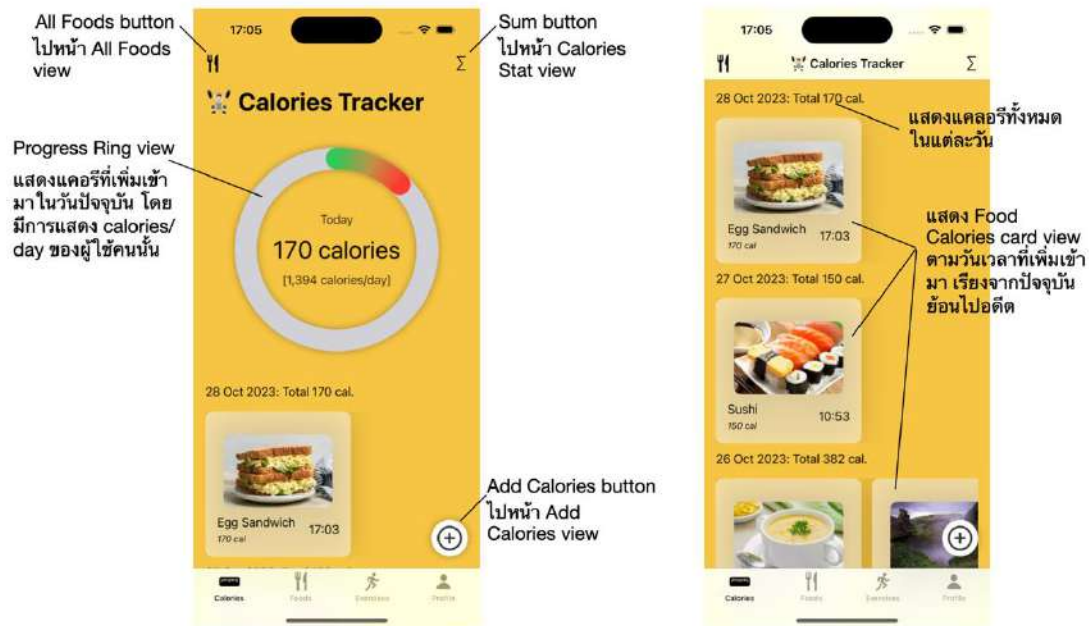
รูปที่ 3.6 Login screen and Register screen

4. ผลการวิจัย

ในบทนี้จะแสดงผลของการพัฒนาแอปพลิเคชัน โดยแสดง User Interface ของแอปพลิเคชันที่พัฒนา แสดงฐานข้อมูล Firebase ที่สร้างขึ้น และแสดงผลการทดสอบระบบที่พัฒนาขึ้น

4.1 User Interface สำหรับแอปพลิเคชันที่พัฒนาสำเร็จ

แสดง User Interface ของแอปพลิเคชันที่พัฒนาสำเร็จ ยกตัวอย่างตามภาพที่ 4.1



รูปที่ 4.1 Calories Tracker view

4.2 แสดงฐานข้อมูล Firebase สำหรับแอปพลิเคชันที่พัฒนาสำเร็จ

สำหรับ Firebase ผู้พัฒนาใช้งานในแอปพลิเคชันอยู่ 3 ส่วนคือ

- (1) Authentication with Email/Password
- (2) Cloud Firestore เก็บข้อมูลเป็น Collections
- (3) Cloud Storage ใช้เก็บข้อมูลรูปภาพต่างๆ แล้วเก็บ Link ของรูปภาพนั้นไว้ที่ Cloud

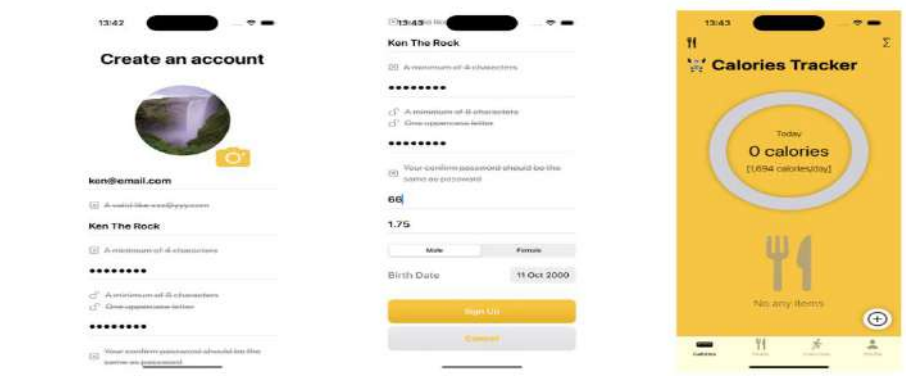
Firestore

4.3 แสดงการทดสอบระบบของแอปพลิเคชันที่พัฒนาสำเร็จ

ในการทดสอบ ผู้เขียนได้ทำการแยกเป็น 33 Test cases โดยจะยกตัวอย่าง 1 Test case ตามตารางที่ 4.1 ด้านล่าง

ตารางที่ 4.1 TC01 การลงทะเบียนผู้ใช้ใหม่ (Create new user account) - ใส่ข้อมูลครบและถูกต้อง

Test case number	TC01
Test case title	การลงทะเบียนผู้ใช้ใหม่ (Create new user account) - ใส่ข้อมูลครบและถูกต้อง
Precondition	Download App
Test steps	(4) กดปุ่ม Sign up ในหน้า Login (5) ระบบนำสู่หน้า Create user account (6) ผู้ใช้กรอกข้อมูลครบถ้วน และถูกต้อง (7) ระบบ Create user account ใน Firebase Authentication และ user ใน users collection ใน Cloud Firestore (8) App นำผู้ใช้ไปหน้า Main view (Calories Tracker view)
Test data	1. Profile image 2. Email (A valid like xxx@yyy.com): ken@email.com 3. Username (Minimum 4 characters): Ken The Rock 4. Password (Minimum 8 characters/One uppercase letter): 1234567A 5. Confirm Password (Same as Password): 1234567A 6. Weight(kg) (if not fill, default 50 kg): 66 7. Height(m) (if not fill, default 1.65 m): 1.75 8. Choose between Male or Female: Male 9. Birthdate (default is Today): Oct 11, 2000
Expected result	ระบบสร้าง User ใน Firebase Authentication/Cloud Firestore และนำผู้ใช้เข้าสู่หน้า Main view (Calories Tracker View)
Actual result	ระบบสร้าง User ใน Firebase Authentication/Cloud Firestore และนำผู้ใช้เข้าสู่หน้า Main view (Calories Tracker View)
Status	Passed

Result Screens	
----------------	--

4.4 แสดงผลการสำรวจความพึงพอใจของผู้ใช้

ผู้พัฒนามีการอัด VDO Demo ส่งให้ผู้ใช่ โดยมีการส่งแบบสำรวจให้ผู้ใช่แสดงความเห็นใน Google Form โดยมีคำถามดังต่อไปนี้

- (1) เพศของผู้ตอบแบบสอบถาม - มีตัวเลือก (ชาย, หญิง, ไม่ระบุ)
- (2) อายุของผู้ตอบแบบสอบถาม - เป็น Field ให้เติม
- (3) ความพึงพอใจต่อ User Interface - มีตัวเลือก (น้อยที่สุด, น้อย, ปานกลาง, มาก, มากที่สุด)
- (4) ความพึงพอใจต่อ Functions - มีตัวเลือก (น้อยที่สุด, น้อย, ปานกลาง, มาก, มากที่สุด)
- (5) ข้อเสนอแนะ หรือความเห็นอื่นๆ - เป็น Field ให้เติม

โดยมีผลจากแบบสอบถาม (ล่าสุดวันที่ 11/11/2566) ดังต่อไปนี้

1. เพศของผู้ตอบแบบสอบถาม: ชาย 66.7%, หญิง 33.3%
2. อายุของผู้ตอบแบบสอบถาม: 23 ถึง 53
3. ความพึงพอใจต่อ User Interface: 16.7% มากที่สุด, 75% มาก, 8.3% ปานกลาง
4. ความพึงพอใจต่อ Functions: 33.3% มากที่สุด, 50% มาก, 16.7% ปานกลาง
5. ข้อเสนอแนะ หรือความเห็นอื่น ๆ โดยมีการเสนอแนะ หรือความเห็นอื่นๆ(จะไม่นำเสนอในที่นี้)

5. สรุปผลการวิจัย อภิปรายผล

ผู้เขียนได้ทำการพัฒนาแอปพลิเคชันติดตามแคลอรีการบริโภคอาหารบนอุปกรณ์พกพา โดยใช้ SwiftUI framework ต่อกับฐานข้อมูล Firebase (ใช้ในส่วนของการ Authentication ด้วย Email/Password, Cloud Firestore, และ Cloud Storage) ซึ่งแอปพลิเคชันนี้รองรับบนอุปกรณ์ iPhone iOS 16.4 ขึ้นไป ซึ่งหลังจากพัฒนาแล้วเสร็จ ก็ตรวจสอบกลับได้ว่าตรงกับวัตถุประสงค์ทั้งหมดในข้อ 1.3 และมี Function ครบถ้วนตาม ความต้องการของระบบในข้อ 3.1 ซึ่งแอปพลิเคชันนี้มีทั้งหมด 12 Use cases ตามหัวข้อ 3.2 มีการแสดง Flow ของแอปในหัวข้อ 3.3 Activity Diagram รวมทั้ง Class Diagram หัวข้อ 3.4 และ ER Diagram หัวข้อ 3.5 และมีการแสดง UX/UI ที่มีการออกแบบไว้ด้วยเครื่องมือ Figma ในหัวข้อ 3.6

สำหรับแอปที่พัฒนาสำเร็จ มีการแสดง UI ที่พัฒนาขึ้นในหัวข้อ 4.1 ซึ่งมีการอธิบายว่ามีส่วนประกอบอะไรบ้างในแต่ละหน้าจอของ UI รวมทั้งในหัวข้อ 4.2 มีการแสดงฐานข้อมูล Firebase ที่พัฒนาขึ้นตามที่มีการออกแบบไว้ (Authentication ด้วย Email/Password, เก็บรูปภาพใน Cloud Storage, และเก็บฐานข้อมูลแบบ Document เป็น Collection ใน Cloud Firestore) ส่วนในหัวข้อ 4.3 มีการแสดง Test cases ทั้งหมด 24 Test cases โดยมีการเทียบว่าตรงกับ Use case ไດ และตัวอย่างผลการทดสอบตามตาราง และในหัวข้อ 4.4 มีการสรุปผลการสำรวจกับผู้ใช้ โดยการบันทึก VDO Demo ให้ผู้ใช้ดู และให้ผู้ใช้บอกความเห็นในแบบสำรวจที่ทำขึ้นด้วย Google Form โดยความพึงพอใจต่อ User Interface - 20% พอใจมากที่สุด 80% พอใจมาก 10% พอใจปานกลาง และความพึงพอใจต่อ Function ที่มี - 40% พอใจมากที่สุด 40% พอใจมาก 20% พอใจปานกลาง

บรรณานุกรม

- [1] “Best Mobile App Development Tools, Frameworks & Platforms,” Techliance Blog [Online]. <https://blog.techliance.com/mobile-app-development-tools/>. (Accessed: Oct 27, 2023).
- [2] “Why Your Business Should Develop An iOS App,” Quantilus innovation [Online]. <https://quantilus.com/article/why-your-business-should-develop-an-ios-app/>. (Accessed: Oct 27, 2023).
- [3] J. Schmitt, “Native vs cross-platform mobile app development,” Circleci Blog [Online]. <https://circleci.com/blog/native-vs-cross-platform-mobile-dev/>. (Accessed: Oct 27, 2023).
- [4] M. Pratap, “SwiftUI vs UIKit: Which one to choose for your iOS app,” Supersourcing [Online]. <https://supersourcing.com/blog/?s=SwiftUI+vs+UIKit%3A+Which+one+to+choose+for+your+iOS+app>. (Accessed: Oct 27, 2023).
- [5] B. Guriev, “Firebase Pros and Cons: When you should and shouldn’t use Firebase,” OSDB [Online]. <https://osdb.io/firebase-pros-and-cons-when-you-should-and-shouldnt-use-firebase-osdb/>. (Accessed: Oct 27, 2023).