

ปริญญา尼พนธ์

แอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว

Food portions and feline body mass index calculator application for cat

นางสาวณัฏฐณิชา เหลาเพ็ชร

นางสาวปานวด สاثสนิท

นายพงศ์ศิริ วิล่าวรรณ

ปริญญา尼พนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ครุศาสตร์อุตสาหกรรมบัณฑิต (ครุศาสตร์วิศวกรรม)

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรมและเทคโนโลยี

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2566

ปริมาณนิพนธ์

เรื่อง แอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว

Food portions and feline body mass index calculator application for cat

วัตถุประสงค์

- เพื่อศึกษาเครื่องมือในการสร้างแอปพลิเคชันคำนวณปริมาณอาหาร ดัชนีมวลกาย สำหรับแมว และการทำเครื่องซึ่งประเภทส่งข้อมูลผ่าน WiFi
- เพื่อออกแบบแอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว
- เพื่อสร้างแอปพลิเคชันคำนวณปริมาณอาหาร ดัชนีมวลกายสำหรับแมว และทำเครื่องซึ่งประเภทส่งข้อมูลผ่าน WiFi
- เพื่อทดลองใช้แอปพลิเคชันคำนวณปริมาณอาหาร ดัชนีมวลกายสำหรับแมว และเครื่องซึ่งประเภทส่งข้อมูลผ่าน WiFi ไปใช้กับกลุ่มตัวอย่าง
- เพื่อนำแอปพลิเคชันคำนวณปริมาณอาหาร ดัชนีมวลกายสำหรับแมว และเครื่องซึ่งประเภทส่งข้อมูลผ่าน WiFi ไปใช้งานจริง

ประโยชน์ที่คาดว่าจะได้รับ

- ได้ความรู้เกี่ยวกับเครื่องมือในการสร้างแอปพลิเคชันคำนวณปริมาณอาหาร ดัชนีมวลกาย สำหรับแมว และการทำเครื่องซึ่งประเภทส่งข้อมูลผ่าน WiFi
- ได้แบบจำลองของแอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว
- ได้แอปพลิเคชันคำนวณปริมาณอาหาร ดัชนีมวลกายสำหรับแมว และเครื่องซึ่งประเภทส่งข้อมูลผ่าน WiFi
- ได้ผลการทดลองใช้แอปพลิเคชันคำนวณปริมาณอาหาร ดัชนีมวลกายสำหรับแมวจาก และเครื่องซึ่งประเภทส่งข้อมูลผ่าน WiFi กลุ่มตัวอย่าง
- ได้นำแอปพลิเคชันคำนวณปริมาณอาหาร ดัชนีมวลกายสำหรับแมว และเครื่องซึ่งประเภทส่งข้อมูลผ่าน WiFi ไปใช้งานจริง

ชื่อหัวข้อ	แอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว	
ชื่อนักศึกษา	นางสาวณัฐชนิชา เหลาเพ็ชร	
	นางสาวปานวด	สาทสนิท
	นายพงศ์ศิริ	วิลาวรรณ
อาจารย์ที่ปรึกษา	ผศ. พงษ์เกียรติ จงไตรลักษณ์	
ปริญญา	ครุศาสตร์อุตสาหกรรมบัณฑิต	
หลักสูตร	ครุศาสตร์วิศวกรรม	
ปีการศึกษา	2566	

บทคัดย่อ

ปริญนานิพนธ์ฉบับนี้นำเสนอแอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว เพื่อแก้ไขปัญหาแมวที่มีดัชนีมวลกายเกินเกณฑ์มาตรฐาน เนื่องจากผู้เลี้ยงไม่สามารถคำนวณปริมาณอาหารที่เหมาะสมกับแมวได้ด้วยตนเอง ผู้จัดทำจึงได้พัฒนาแอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว โดยการทำางของแอปพลิเคชันมีการให้ผู้ใช้งานลงทะเบียนบัญชีแมว อีกทั้งยังสามารถคำนวณดัชนีมวลกายสำหรับแมว คำนวณโภชนาการของผลิตภัณฑ์ และบันทึกสูตรอาหารในรูปแบบการ์ด และทำการคำนวณปริมาณอาหารที่ควรได้รับต่อวัน

ผลการประเมินความพึงพอใจต่อแอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว โดยผู้ประเมินทั้งหมด 12 คน พบว่าความพึงพอใจต่อการออกแบบแอปพลิเคชัน มีค่าเฉลี่ยอยู่ที่ 3.85 ส่วนเบี่ยงเบนมาตรฐาน 0.37 อยู่ในระดับมาก ความพึงพอใจต่อการใช้งานแอปพลิเคชัน มีค่าเฉลี่ยอยู่ที่ 4.31 ส่วนเบี่ยงเบนมาตรฐาน 0.15 อยู่ในระดับมาก สรุปได้ว่าแอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว ผู้ใช้งานมีความพึงพอใจและแอปพลิเคชันสามารถนำไปใช้งานได้จริง

คำสำคัญ : ดัชนีมวลกายสำหรับแมว โภชนาการของผลิตภัณฑ์ ปริมาณอาหารที่ควรได้รับต่อวัน

II

Thesis Title	Food Portions and Feline Body Mass Index Calculator Application for Cat	
Student	Miss Nuttanicha	Loapet
	Miss Panwat	Satsanit
	Mr. Phongsiri	Wilawan
Advisor	Asst. Prof. Pongkiat	Jongtriluck
Degree	Bachelor of Science in Industrial Education	
Program in	Engineering Education	
Academic Year	2023	

ABSTRACT

This thesis presents a Food Portions and Feline Body Mass Index Calculator Application for Cat to address the issue of cats with overweight problems. This is because the cat owners cannot calculate an appropriate amount of food for their cats by themselves. So the developers have developed this application to help for calculation of the appropriate amount of food for their cats. The application works by allowing users to register cat accounts, calculate the feline body mass index for cats, calculate the nutrition of products, record food recipes in card format, and calculate a recommended daily intake.

The satisfaction of using this application was evaluated by 12 people. It was found that the satisfaction in application's design had an average score of 3.85 with SD 0.37 and application's usability had an average score of 4.31 with SD 0.15. Both categories of satisfaction show a high level of satisfaction. In conclusion, the Food Portions and Feline Body Mass Index Calculator Application for Cat was found to be satisfactory for users and can be used in practice.

Keyword : Feline body mass index, Nutrition of products, Recommended daily intake

กิตติกรรมประกาศ

ปริญญาอุดมศึกษาจุล่วงไปได้ด้วยดี เนื่องจากได้รับความกรุณาอย่างดียิ่งจาก
ผู้ช่วยศาสตราจารย์พงษ์เกียรติ จงไตรลักษณ์ อาจารย์ที่ปรึกษาปริญญาอุดมศึกษา ผู้ช่วยศาสตราจารย์สุระชัย
พิมพ์สาลี และผู้ช่วยศาสตราจารย์ ดร.เพบูลย์ พวงวงศ์ตระกูล ที่กรุณาให้คำปรึกษา เสนอแนะ
ตรวจสอบแก้ไข และให้ข้อคิดที่ดีจนสามารถจัดทำปริญญาอุดมศึกษาจุล่วงไปได้ดี

ขอขอบคุณผู้ที่ร่วมทดลองใช้งานและประเมินความพึงพอใจต่อแอปพลิเคชันคำนวณปริมาณ
อาหารและต้นน้ำมวลกายสำหรับแมว ทำให้ทราบข้อมูลร่องเพื่อนำไปปรับปรุงแก้ไขให้ปริญญาอุดมศึกษาจุล่วงไปได้ดี
นี้มีความสมบูรณ์มากขึ้น

ขอขอบพระคุณบิดา มารดา และผู้มีพระคุณสำหรับพากเราที่ได้ให้การสนับสนุนในการจัดทำ
ปริญญาอุดมศึกษาจุล่วงไปได้ดี และสุดท้ายต้องขอขอบคุณทุกท่านที่เป็นกำลังใจเสมอมา

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	I
ABSTRACT	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูปภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 จุดมุ่งหมายของโครงการ	2
1.3 สมมุติฐานของการจัดทำโครงการ	2
1.4 ขีดความสามารถของโครงการ	2
1.5 ขั้นตอนการทำโครงการ	3
1.6 เนื้อหาโดยสังเขป	4
บทที่ 2 ทรัพยากรถวายและหลักการ	5
2.1 กล่าวนำ	5
2.2 สุขภาพของแมว	5
2.2.1 การประเมินร่างกายของแมว	5
2.2.2 สาเหตุที่ทำให้แมวน้ำหนักเกินเกณฑ์มาตรฐาน	8
2.2.3 ความเสี่ยงด้านสุขภาพของแมวน้ำหนักเกินเกณฑ์มาตรฐาน	8
2.3 โภชนาการอาหารแมว	9
2.3.1 การคำนวณโภชนาการของผลิตภัณฑ์	9
2.3.2 การคำนวณพลังงานของแมว	12
2.3.3 การคำนวณปริมาณอาหารให้เหมาะสมกับแมว	13
2.4 หลักการเกี่ยวกับการออกแบบแบบแอปพลิเคชัน	14

สารบัญ (ต่อ)

เรื่อง	หน้า
2.4.1 การออกแบบประสบการณ์ของผู้ใช้งาน	14
2.4.2 การออกแบบส่วนติดต่อระหว่างผู้ใช้งานกับผลิตภัณฑ์	14
2.5 ทฤษฎีการเขียนแอปพลิเคชันด้วยภาษาดาร์ต	16
2.6 การศึกษาหลักการเข้มต่อเครื่องซึ่งน้ำหนักกับแอปพลิเคชัน	17
2.6.1 ประเภทของเครื่องซึ่ง	17
2.6.2 หลักการทำงานเครื่องซึ่ง	18
2.6.3 การเชื่อมต่อเครื่องซึ่งน้ำหนักกับแอปพลิเคชัน	22
2.7 งานวิจัยที่เกี่ยวข้อง	22
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	24
3.1 กล่าวนำ	24
3.2 การออกแบบแอปพลิเคชัน	24
3.2.1 การออกแบบในส่วนของ UX	24
3.2.2 การออกแบบในส่วนของ UI	26
3.3 การเขียนแอปพลิเคชัน	31
3.3.1 หน้าลงทะเบียนบัญชีเม瓦	31
3.3.2 หน้าคำนวนดัชนีมวลกาย	53
3.3.3 หน้าคำนวนปริมาณอาหารที่ควรได้รับต่อวัน	59
3.3.4 หน้าคำนวนโภชนาการของผลิตภัณฑ์ในสูตรอาหาร	74
3.3.5 หน้าบันทึกสูตรอาหาร	90
3.3.6 การเก็บข้อมูล	93
3.4 การเชื่อมต่อเครื่องซึ่งน้ำหนักกับแอปพลิเคชัน	97
3.5 การทำงานของแอปพลิเคชัน	98
บทที่ 4 การทดลองและผลการทดลอง	101
4.1 กล่าวนำ	101
4.2 การทดลองและผลการทดลองใช้งานแอปพลิเคชัน	101

สารบัญ (ต่อ)

เรื่อง	หน้า
4.2.1 การทดลองบันทึกข้อมูล	101
4.2.2 การทดลองคำนวณ	104
4.3 การประเมินผลการใช้งานของแอปพลิเคชัน	109
4.3.1 ข้อมูลทั่วไป	109
4.3.2 ระดับความพึงพอใจต่อแอปพลิเคชัน	112
บทที่ 5 บทสรุป	115
5.1 สรุป	115
5.2 ปัญหาและวิธีการแก้ไข	116
5.3 แนวทางการพัฒนา	117
บรรณานุกรม	118
ภาคผนวก ก ผังการทำงาน	120
ภาคผนวก ข แผนผังความสัมพันธ์ของฐานข้อมูล	127
ภาคผนวก ค แผนผังการออกแบบแอปพลิเคชัน	129
ภาคผนวก ง รหัสต้นฉบับของโปรแกรม	134
ภาคผนวก จ คู่มือการใช้งาน	176
ประวัติผู้แต่ง	188

สารบัญตาราง

ตารางที่	หน้า
2.1 แบบชนิดของข้อมูลพื้นฐานในภาษาดาร์ต	16
4.1 ผลจำนวนและร้อยละของกลุ่มผู้ทดลองใช้งาน โดยจำแนกตามเพศ	110
4.2 ผลจำนวนและร้อยละของกลุ่มผู้ทดลองใช้งาน โดยจำแนกตามอายุ	110
4.3 ผลจำนวนและร้อยละของกลุ่มผู้ทดลองใช้งาน โดยจำแนกตามอุปกรณ์ที่ใช้งาน	111
4.4 ผลจำนวนและร้อยละของกลุ่มผู้ทดลองใช้งาน โดยจำแนกตามจำนวนแมวที่ผู้ใช้เลี้ยง	111
4.5 ผลการประเมินความพึงพอใจต่อการออกแบบแอปพลิเคชัน	112
4.6 ผลการประเมินความพึงพอใจต่อการใช้งานแอปพลิเคชัน	113
4.7 ผลรวมการประเมินความพึงพอใจ	113

สารบัญรูปภาพ

รูปที่	หน้า
2.1 เกณฑ์การให้คะแนนร่างกายแมว 1-5 คะแนน	5
2.2 เกณฑ์การให้คะแนนร่างกายแมว 1-9 คะแนน	6
2.3 ตำแหน่งการวัดเพื่อคำนวณดัชนีมวลกายสำหรับแมว	7
2.4 สูตรคำนวณดัชนีมวลกายสำหรับแมว	7
2.5 รายละเอียดในผลิตภัณฑ์ของอาหารแมว	10
2.6 พลังงานที่สัตว์นำไปได้	10
2.7 สูตรการคำนวณโภชนาการของผลิตภัณฑ์	11
2.8 คุณภาพของอาหารสัตว์ทางเคมี	11
2.9 สูตรคำนวณร้อยละของการนำไปใช้เดรต	12
2.10 สูตรคำนวณโภชนาการของผลิตภัณฑ์	12
2.11 สูตรคำนวณปริมาณพลังงานที่แมวควรได้รับ	13
2.12 สูตรคำนวณปริมาณพลังงานที่แมวควรได้รับต่อวัน	13
2.13 สูตรคำนวณปริมาณของอาหารที่ควรได้รับต่อวัน	13
2.14 แนวคิดของ UX และ UI	15
2.15 การยึดหยดของสเตรนเกจ	19
2.16 วงจรปริ๊ดจ์บนโลดเซลล์	19
2.17 ส่วนประกอบของเครื่องซั่ง	20
2.18 HX711 Amplifier	20
2.19 ESP8266	21
2.20 การเชื่อมต่ออุปกรณ์การทำเครื่องซั่ง	21
2.21 การเชื่อมต่อเครื่องซั่งผ่านอินเทอร์เน็ต	22
3.1 หน้าลงทะเบียนบัญชีของแมว	24
3.2 หน้าคำนวณดัชนีมวลกาย	25
3.3 หน้าบันทึกสูตรอาหาร	25
3.4 สีของแอปพลิเคชัน	26

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
3.5 หน้าเข้าสู่ระบบผู้ใช้งาน	27
3.6 หน้าเลือกเครื่องซึ่งน้ำหนัก	27
3.7 หน้าคำนวนโภชนาการของผลิตภัณฑ์	28
3.8 หน้างลงทะเบียนบัญชีแมว	29
3.9 หน้าคำนวนดัชนีมวลกาย	29
3.10 หน้าแสดงและแก้ไขข้อมูลของแมว	30
3.11 หน้าคำนวนปริมาณอาหารที่ควรได้รับต่อวัน	30
3.12 หน้ากราฟน้ำหนัก ดัชนีมวลกาย และปริมาณอาหาร	31
3.13 คำสั่งสร้างปุ่มคำแนะนำ	31
3.14 ปุ่มคำแนะนำ	34
3.15 คำสั่งสร้างการ์ดบัญชีแมวและสูตรอาหาร	34
3.16 การ์ดบัญชีแมวและสูตรอาหาร	39
3.17 คำสั่งสร้างปุ่ลลงทะเบียนแมว	39
3.18 ปุ่มเข้าสู่หน้างลงทะเบียนแมว	40
3.19 คำสั่งสร้างปุ่มแสดงหน้า Home และหน้า Food	41
3.20 ปุ่มแสดงหน้าหลักและหน้าคำนวนปริมาณอาหารที่ควรได้รับต่อวัน	42
3.21 คำสั่งสร้างช่องรับข้อมูลชื่อแมว น้ำหนัก และเพศ	42
3.22 ช่องกรอกข้อมูลชื่อแมว น้ำหนัก และเพศ	44
3.23 คำสั่งสร้างปฏิทินวันเดือนปีเกิด	44
3.24 หน้าปฏิทินวันเดือนปีเกิด	46
3.25 คำสั่งสร้างปุ่มบันทึกข้อมูล	47
3.26 ปุ่มบันทึกข้อมูล	48
3.27 คำสั่งการกดเข้าสู่หน้าคำนวนดัชนีมวลกายผ่านการ์ดบัญชีแมว	49
3.28 การเข้าสู่หน้าคำนวนดัชนีมวลกายผ่านการ์ดบัญชีแมว	49
3.29 คำสั่งสร้างปุ่มแก้ไขข้อมูลแมว	50

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
3.30 คำสั่งแก้ไขข้อมูลเมือง	50
3.31 ปุ่มแก้ไขข้อมูลเมือง	53
3.32 คำสั่งแสดงรูปแบบหนึ่งของการวัดเพื่อคำนวณดัชนีมวลกายและช่องกรอกข้อมูลการวัด	53
3.33 การแสดงรูปแบบหนึ่งของการวัดเพื่อคำนวณดัชนีมวลกายและช่องกรอกข้อมูลการวัด	55
3.34 คำสั่งสร้างปุ่มคำนวณดัชนีมวลกาย	55
3.35 ปุ่มคำนวณดัชนีมวลกาย	56
3.36 คำสั่งคำนวณดัชนีมวลกาย	56
3.37 คำสั่งแสดงผลลัพธ์ดัชนีมวลกาย	57
3.38 การแสดงผลลัพธ์ดัชนีมวลกาย	58
3.39 คำสั่งสร้างช่องแสดงตัวเลือกสูตรอาหาร	59
3.40 ช่องแสดงตัวเลือกสูตรอาหาร	60
3.41 คำสั่งสร้างตัวเลือกสูตรอาหาร	60
3.42 การแสดงตัวเลือกสูตรอาหาร	61
3.43 คำสั่งสร้างช่องกรอกค่าโภชนาการของผลิตภัณฑ์และช่องกรอกค่าน้ำหนัก	62
3.44 ช่องกรอกค่าโภชนาการของผลิตภัณฑ์และช่องกรอกค่าน้ำหนัก	63
3.45 คำสั่งสร้างปุ่มเลือกดัชนีมวลกาย	63
3.46 ปุ่มเลือกดัชนีมวลกาย	65
3.47 คำสั่งสร้างผลลัพธ์การคำนวณปริมาณอาหารที่ควรได้รับต่อวัน	65
3.48 การแสดงข้อความและผลลัพธ์การคำนวณปริมาณอาหารที่ควรได้รับต่อวัน	66
3.49 คำสั่งการคำนวณสูตรอาหาร	67
3.50 คำสั่งสร้างข้อความเตือนหากผู้ใช้งานเลือกสูตรอาหารเกิน 1 รูปแบบ	68
3.51 ข้อความเตือนหากผู้ใช้งานเลือกสูตรอาหารเกิน 1 รูปแบบ	69
3.52 คำสั่งสร้างปุ่มเข้าสู่หน้าคำนวณโภชนาการในการสูตรอาหาร	70
3.53 เข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์	72
3.54 คำสั่งสร้างปุ่มเข้าสู่หน้าบันทึกสูตรอาหาร	72

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
3.55 ปุ่มเข้าสู่หน้าบันทึกสูตรอาหาร	74
3.56 คำสั่งสร้างปุ่มเข้าสู่หน้าคำนวนโภชนาการของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้	74
3.57 ปุ่มเข้าสู่หน้าคำนวนโภชนาการของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้	77
3.58 คำสั่งสร้างปุ่มเข้าสู่หน้าคำนวนโภชนาการของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้	77
3.59 ปุ่มเข้าสู่หน้าคำนวนโภชนาการของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้	79
3.60 คำสั่งสร้างช่องรับข้อมูลของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้	79
3.61 ช่องกรอกข้อมูลของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้	81
3.62 คำสั่งสร้างปุ่มคำนวนและการแสดงผลลัพธ์ของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้	81
3.63 ปุ่มคำนวนและการแสดงผลลัพธ์ของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้	84
3.64 คำสั่งสร้างช่องรับข้อมูลของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้	84
3.65 ช่องรับข้อมูลของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้	86
3.66 คำสั่งสร้างปุ่มคำนวนและการแสดงผลลัพธ์ของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้	87
3.67 ปุ่มคำนวนและการแสดงผลลัพธ์ของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้	89
3.68 คำสั่งสร้างช่องรับข้อมูลสูตรอาหาร	90
3.69 ช่องกรอกข้อมูลสูตรอาหาร	91
3.70 คำสั่งสร้างปุ่มบันทึกสูตรอาหาร	91
3.71 ปุ่มบันทึกสูตรอาหาร	93
3.72 คำสั่งจัดรูปแบบข้อมูล	93
3.73 คำสั่งการจัดเก็บข้อมูล	94
3.74 การเชื่อมต่อเครื่องซั่งน้ำหนักกับแอปพลิเคชัน	97
3.75 การทำงานของแอปพลิเคชันในการเชื่อมต่อ กับ เครื่องซั่งน้ำหนัก	98
3.76 การทำงานของเครื่องซั่งน้ำหนักในการเชื่อมต่อ กับ แอปพลิเคชัน	98
3.77 การทำงานของแอปพลิเคชัน	99
4.1 ตัวอย่างการทดลองเพิ่มบัญชีเม瓦	102
4.2 ตัวอย่างการทดลองแก้ไขข้อมูลของเม瓦	102

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
4.3 ตัวอย่างการทดลองกดปุ่มเข้าสู่หน้าบันทึกสูตรอาหาร	103
4.4 ตัวอย่างการทดลองบันทึกสูตรอาหาร	104
4.5 ตัวอย่างการทดลองการคำนวณดัชนีมวลกายสำหรับแมว	105
4.6 ตัวอย่างการทดลองแสดงผลลัพธ์ดัชนีมวลกายของแมว	105
4.7 ตัวอย่างการทดลองเข้าสู่หน้าคำนวณโภชนาการที่ผลิตภัณฑ์ระบุพลังงานที่นำไปใช้ได้	106
4.8 ตัวอย่างการทดลองเข้าสู่หน้าคำนวณโภชนาการที่ผลิตภัณฑ์ไม่ระบุพลังงานที่นำไปใช้ได้	107
4.9 ตัวอย่างการทดลองเลือกสูตรอาหารเพื่อคำนวณปริมาณของอาหารที่ควรได้รับต่อวัน	108
4.10 ตัวอย่างการทดลองกรอกโภชนาการด้วยตัวเอง	108
4.11 ตัวอย่างการทดลองแสดงข้อความเตือนให้เลือกใช้วิธีการคำนวณที่ลัษณะแบบ	109
ก.1 แผนผังการลงทะเบียนบัญชีแมว	121
ก.2 แผนผังการแก้ไขข้อมูลในบัญชีแมว	122
ก.3 แผนผังการบันทึกข้อมูลสูตรอาหาร	123
ก.4 แผนผังการคำนวณดัชนีมวลกาย	124
ก.5 แผนผังการคำนวณสูตรอาหาร	125
ก.6 แผนผังการคำนวณปริมาณอาหารที่ควรได้รับต่อวัน	126
ข.1 แผนผังความสัมพันธ์ของฐานข้อมูล	128
ค.1 หน้าลงทะเบียนแมว	130
ค.2 หน้าคำนวณดัชนีมวลกายสำหรับแมว	131
ค.3 หน้าคำนวณโภชนาการของผลิตภัณฑ์	132
ค.4 หน้าบันทึกสูตรอาหาร	132
ค.5 หน้าคำนวณปริมาณอาหารที่ควรได้รับต่อวัน	133
จ.1 การติดตั้งแอปพลิเคชัน	178
จ.2 หน้าแสดงคำแนะนำการใช้งาน	179
จ.3 ตัวอย่างการลงทะเบียนบัญชีแมว	180
จ.4 ตัวอย่างการแก้ไขข้อมูลในบัญชีแมว	181

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
จ.5 ตัวอย่างการคำนวณดัชนีมวลกายสำหรับแมว	182
จ.6 ปุ่มเข้าสู่หน้าเลือกสูตรในการคำนวณโภชนาการของผลิตภัณฑ์	182
จ.7 หน้าเลือกสูตรในการคำนวณโภชนาการของผลิตภัณฑ์	183
จ.8 ตัวอย่างคำนวณโภชนาการของผลิตภัณฑ์โดยระบุพลังงานที่สัตว์นำไปใช้ได้	184
จ.9 ตัวอย่างคำนวณโภชนาการของผลิตภัณฑ์โดยไม่ระบุพลังงานที่สัตว์นำไปใช้ได้	184
จ.10 ปุ่มเข้าสู่หน้าบันทึกสูตรอาหาร	185
จ.11 ตัวอย่างการบันทึกสูตรอาหาร	185
จ.12 ตัวอย่างการคำนวณปริมาณอาหารที่ควรได้รับต่อวันแบบใช้สูตรที่มีในแอปพลิเคชัน	186
จ.13 ตัวอย่างการคำนวณปริมาณอาหารที่ควรได้รับต่อวันแบบกรอกโภชนาการด้วยตัวเอง	187

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันพบเห็นแมวที่มีดัชนีมวลกายเกินเกณฑ์มาตรฐานได้มากขึ้น โดยแมวที่มีดัชนีมวลกายเกินเกณฑ์มาตรฐานนักจะส่งผลต่อรูปร่างน้ำหนักและการเคลื่อนไหวซึ่งส่งผลกระทบต่อสุขภาพเป็นวงกว้าง ดังนั้นจึงควรดูแลเอาใจใส่โดยการลดน้ำหนักแมวอย่างถูกต้องซึ่งการทำให้แมวมีดัชนีมวลกายตามเกณฑ์มาตรฐานจะต้องอาศัยความร่วมมือจากเจ้าของเป็นสำคัญ (Sidum, 2023)

จากการศึกษางานวิจัยทางด้านการควบคุมน้ำหนักและดัชนีมวลกายของแมวโดย Sidum (2023) ได้กล่าวว่า แมวที่มีดัชนีมวลกายเกินเกณฑ์มาตรฐานเกิดโอกาสเป็นไขมันสะสมตามร่างกาย และการลดน้ำหนักแมวโดยการอดอาหารจะทำให้แมวร่างกายทรุดโทรมลงจากการลดน้ำหนักที่หักโหมจึงควรมีการคำนวณปริมาณอาหารที่เหมาะสมกับแมวตัวนั้น ๆ

โดยข้อมูลจาก Meksophawankun (2023) ได้กล่าวว่า แมวที่มีดัชนีมวลกายเกินเกณฑ์มาตรฐานทำการต่างๆ ลำบาก เนื่องด้วยทั้งอุปนิสัยที่เปลี่ยนไปทำให้มีความกระตือรือร้นน้อยลงจึงทำให้เกิดปัญหาต่างๆ เช่น มีความเสี่ยงในการเป็นโรคเบาหวาน และโรคข้ออักเสบเนื่องจากน้ำหนักตัวที่มากเกินไปทำให้ข้อชาต่างๆ ต้องรับน้ำหนักมากขึ้น และไขมันมากอพกที่ตับอาจเกิดภาวะตับวายตามมา

นอกจากนี้ Royal Canin (2023) ได้อธิบายถึงสาเหตุที่ทำให้แมวมีดัชนีมวลกายเกินเกณฑ์มาตรฐานโดยกล่าวว่า แมวอ้วนเกิดได้จาก กินแบบไม่ชั่งตวง กินขนมไม่จำกัดจำนวน และไม่ค่อยเคลื่อนไหวร่างกาย การทำให้แมวมีดัชนีมวลกายตามเกณฑ์มาตรฐานต้องมีการรักษาสมดุลที่ถูกต้องระหว่างพลังงานที่แมวได้รับผ่านทางอาหารและพลังงานที่แมวใช้งาน

ดังนั้น เมื่อแมวมีน้ำหนักเกินกว่าเกณฑ์ปกติจึงมีความเสี่ยงที่จะเกิดโรคต่าง ๆ มากกว่าแมวที่มีน้ำหนักอยู่ในเกณฑ์ปกติ ขณะผู้จัดทำจึงมีความสนใจที่จะศึกษาหาแนวทางเพื่อทำให้แมวที่มีดัชนีมวลกายเกินเกณฑ์มาตรฐานให้เข้าสู่เกณฑ์ปกติได้อย่างเหมาะสม

1.2 จุดมุ่งหมายของโครงการ

1. เพื่อศึกษาเครื่องมือในการสร้างแอปพลิเคชันคำนวณปริมาณอาหาร ด้ัชนีมวลกาย สำหรับแมว และการทำเครื่องซึ่งประเภทส่งข้อมูลผ่าน WiFi
2. เพื่อออกแบบแอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว
3. เพื่อสร้างแอปพลิเคชันคำนวณปริมาณอาหาร ด้ัชนีมวลกายสำหรับแมว และทำเครื่องซึ่งประเภทส่งข้อมูลผ่าน WiFi
4. เพื่อทดลองใช้แอปพลิเคชันคำนวณปริมาณอาหาร ด้ัชนีมวลกายสำหรับแมว และเครื่องซึ่งประเภทส่งข้อมูลผ่าน WiFi ไปใช้กับกลุ่มตัวอย่าง
5. เพื่อนำแอปพลิเคชันคำนวณปริมาณอาหาร ด้ัชนีมวลกายสำหรับแมว และเครื่องซึ่งประเภทส่งข้อมูลผ่าน WiFi ไปใช้งานจริง

1.3 สมมุติฐานของการจัดทำโครงการ

แอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว สามารถทำให้ผู้ใช้งานคำนวณปริมาณอาหารต่อวันสำหรับแมวและคำนวณดัชนีมวลกายของแมวแต่ละตัวได้ โดยผู้ใช้งานจะมีความพึงพอใจในระดับดีขึ้นไป

1.4 ขั้นตอนการดำเนินการ

1. ลงทะเบียนผู้ใช้งานผ่านบัญชีอีเมล
2. ผู้ใช้เลือกรายการใช้งานเครื่องซึ่งน้ำหนักของแอปพลิเคชัน “Cat” ประเภทส่งข้อมูลผ่าน WiFi หรือผู้ใช้กรอกข้อมูลน้ำหนักด้วยตนเอง
3. เก็บประวัติข้อมูล ชื่อ อายุ เพศ น้ำหนักของแมว
4. แสดงข้อมูล ชื่อ อายุ เพศ น้ำหนักของแมว ในลักษณะข้อความ
5. แสดงรูปแนะนำการวัดร่างกายของแมว เพื่อคำนวณ Feline Body Mass Index
6. ใส่ข้อมูลความยาวเส้นรอบวงของกระดูกซี่โครง (ซม.) และความยาวของขาหลังตั้งแต่หัวเข่าถึงข้อเท้า (ซม.)
7. คำนวณ Feline Body Mass Index ของแมวได้ถูกต้อง

8. แสดง Feline Body Mass Index ของแมว ในลักษณะข้อความ
9. รับค่าข้อมูลน้ำหนักด้วยเครื่องชั่งน้ำหนักของแอปพลิเคชัน “Cat” ประเภทส่งข้อมูลผ่าน WiFi หรือผู้ใช้งานกรอกข้อมูลน้ำหนักด้วยตนเอง
10. แสดงรูปภาพประกอบการดู คุณภาพของอาหารสัตว์ทางเคมี (GA)
11. แสดงรูปภาพประกอบการดู พลังงานที่สัตว์นำไปใช้ได้ของผลิตภัณฑ์
12. รับข้อมูลคุณภาพของอาหารสัตว์ทางเคมี (GA) หรือข้อมูลพลังงานที่สัตว์นำไปใช้ได้ของผลิตภัณฑ์ (ME)
13. คำนวนโภชนาการในอาหารแมว
14. คำนวนปริมาณอาหารที่แมวควรได้รับต่อวัน
15. แสดงปริมาณอาหารที่แมวควรได้รับต่อวัน ในลักษณะข้อความ
16. แสดงสถิติ น้ำหนัก Feline Body Mass Index สูตรอาหารที่ใช้ และปริมาณอาหารที่แมวควรได้รับต่อวันในลักษณะกราฟ
17. คาดการณ์ระยะเวลาที่แมวจะมีค่า Feline Body Mass Index เกณฑ์น้ำหนักเหมาะสมกับร่างกายหรือผู้ใช้งานกรอกข้อมูลน้ำหนักแมวด้วยตัวเอง

1.5 ขั้นตอนการทำโครงการ

1. ศึกษาเกี่ยวกับโภชนาการในอาหารแมวและสุขภาพของแมว
2. ศึกษาการออกแบบ UI/UX ของแอปพลิเคชัน
3. ศึกษาการเขียนแอปพลิเคชันด้วย Flutter
4. ศึกษาการทำเครื่องชั่ง WiFi
5. ออกแบบ UI/UX ของแอปพลิเคชัน
6. เรียนแอปพลิเคชัน
7. ทำเครื่องชั่งน้ำหนัก
8. ประเมินความพึงพอใจต่อการใช้งานแอปพลิเคชัน

1.6 เนื้อหาโดยสังเขป

เนื้อหาภายในปริญญาบัตรนี้ จะแบ่งออกเป็นบทต่าง ๆ เพื่อสะดวกต่อการศึกษาและทำความเข้าใจ แต่ละบทประกอบด้วยเนื้อหา ดังต่อไปนี้

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปัญหา จุดมุ่งหมายของโครงงาน สมมุติฐานของ การจัดทำโครงงาน ขีดความสามารถของโครงงาน ขั้นตอนการทำโครงงาน และเนื้อหาโดยสังเขป

บทที่ 2 กล่าวถึงทฤษฎีและหลักการที่เกี่ยวข้องกับโครงงานแอปพลิเคชันคำนวณปริมาณอาหาร และดัชนีมวลกายสำหรับแมว ประกอบด้วย กล่าวนำ สุขภาพของแมว โภชนาการอาหารแมว หลักการ เกี่ยวกับการออกแบบแอปพลิเคชัน ทฤษฎีการเขียนด้วยภาษาดาร์ต การศึกษาหลักการเชื่อมต่อเครื่องซึ่ง น้ำหนักกับแอปพลิเคชัน งานวิจัยที่เกี่ยวข้อง

บทที่ 3 กล่าวถึงการออกแบบ การสร้าง และการทำงาน ประกอบด้วย กล่าวนำ การออกแบบ แอปพลิเคชัน การเขียนแอปพลิเคชัน การเชื่อมต่อเครื่องซึ่งน้ำหนักกับแอปพลิเคชัน และการทำงานของ แอปพลิเคชัน

บทที่ 4 กล่าวถึงการทดลองและผลการทดลอง ประกอบด้วย กล่าวนำ และการทดลองการใช้งาน ของแอปพลิเคชัน ซึ่งประกอบด้วย การทดลองการเข้าสู่ระบบ การเชื่อมต่อกับเครื่องซึ่ง การทดลอง คำนวณ ในส่วนของการทดลองคำนวณจะประกอบด้วย การทดลองคำนวณดัชนีมวลกายสำหรับแมว การ ทดลองคำนวณพลังงานที่แมวควรได้รับต่อวัน การทดลองคำนวณหาพลังงานที่สัตว์นำไปใช้ได้ของ ผลิตภัณฑ์ การทดลองคำนวณปริมาณของอาหารที่ควรได้รับต่อวัน

บทที่ 5 กล่าวถึง สรุป ปัญหาและวิธีการแก้ไข และแนวทางการพัฒนา

บรรณานุกรม

ภาคผนวก ก ผังการทำงาน

ภาคผนวก ข แผนผังความสัมพันธ์ของฐานข้อมูล

ภาคผนวก ค แผนผังการออกแบบแอปพลิเคชัน

ภาคผนวก ง รหัสต้นฉบับของโปรแกรม

ภาคผนวก จ คู่มือการใช้งาน

ประวัติผู้แต่ง

บทที่ 2

ทฤษฎีและหลักการ

2.1 กล่าวนำ

เนื้อหาของปริญญาในบทนี้เป็นทฤษฎีและหลักการที่จะนำมาใช้ประกอบการสร้างโครงงานโดยประกอบด้วย สุขภาพของแมว โภชนาการอาหารแมว หลักการออกแบบและผลิตเครื่องดื่มน้ำหนักกับแมว และงานวิจัยที่เกี่ยวข้อง

2.2 สุขภาพของแมว

แมวที่มีน้ำหนักเกินเกณฑ์มาตรฐานจะดำรงชีวิตและทำกิจกรรมได้ยาก เนื่องด้วยการกินที่มากเกินปริมาณที่ควรได้รับทำให้ความสามารถในการเคลื่อนไหวลดลง (Meksophawankun, 2022)

2.2.1 การประเมินร่างกายของแมว

2.2.1.1 คะแนนร่างกายของแมว

WSAVA Global Nutrition Committe (2022) ได้กล่าวว่า สัตวแพทย์จะประเมินคะแนนร่างกายของแมวโดยใช้การสังเกต และสัมผัสร่างกาย ปัจจุบันมีการให้คะแนนอยู่ 2 ประเภท คือ 1-5 คะแนน และ 1-9 คะแนน ซึ่งมีเกณฑ์ดังต่อไปนี้



รูปที่ 2.1 เกณฑ์การให้คะแนนร่างกายแมว 1-5 คะแนน

ที่มา : Beloved Animal Hospital (2015)

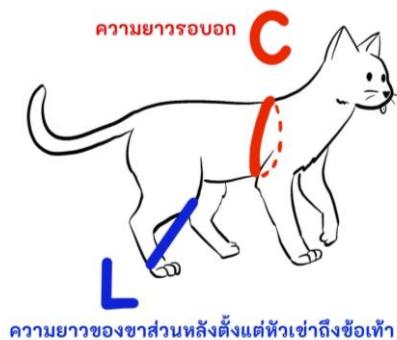


รูปที่ 2.2 เกณฑ์การให้คะแนนร่างกายแมว 1-9 คะแนน
ที่มา : WSAVA Global Nutrition Committee (2022)

- คะแนนร่างกาย 1 เต็ม 5 หรือคะแนน 1-2 เต็ม 9 โดยพิจารณาจากการไม่พบไขมันใต้ผิวนังเห็นกระดูกชี้โครงได้ง่าย และมีไขมัน <5% จัดอยู่ในเกณฑ์ พอสมควร
- คะแนนร่างกาย 2 เต็ม 5 หรือคะแนน 3 เต็ม 9 โดยพิจารณาจากการพบไขมันใต้ผิวนังเล็กน้อย สัมผัสถึงกระดูกชี้โครงได้ง่าย และมีไขมัน 10% จัดอยู่ในเกณฑ์ น้ำหนักต่ำกว่าที่ควร
- คะแนนร่างกาย 3 เต็ม 5 หรือคะแนน 4-5 เต็ม 9 โดยพิจารณาจากการพบไขมันใต้ผิวนังบ้าง สัมผัสถึงกระดูกชี้โครงได้โดยต้องออกแรงกด และมีไขมัน 20% จัดอยู่ในเกณฑ์ น้ำหนักตามที่ควรเป็น
- คะแนนร่างกาย 4 เต็ม 5 หรือคะแนน 6-7 เต็ม 9 โดยพิจารณาจากการพบไขมันใต้ผิวนังมาก สัมผัสถึงกระดูกชี้โครงได้แต่ต้องออกแรงกดมาก และมีไขมัน 30% จัดอยู่ในเกณฑ์ น้ำหนักเกินกว่าที่ควร
- คะแนนร่างกาย 5 เต็ม 5 หรือคะแนน 8-9 เต็ม 9 โดยพิจารณาจากการพบไขมันใต้ผิวนังจำนวนมาก สัมผัสไม่พบกระดูกชี้โครง และมีไขมัน ≥40% จัดอยู่ในเกณฑ์ อ้วน

2.2.1.2 ดัชนีมวลกายสำหรับแมว

การประเมินร่างกายแมวที่คำนวณได้จากการความยาวรอบอก (ซม.) และความยาวของขาส่วนหลังตั้งแต่หัวเข่าถึงข้อเท้า (ซม.) ซึ่งผลลัพธ์จากการคำนวณจะแสดงให้เห็นว่าแมวมีน้ำหนักเหมาะสมกับร่างกายหรือไม่ จากวิธีการประเมินร่างกายของแมวที่ได้กล่าวมาข้างต้น ทางผู้วิจัยได้เลือกใช้วิธีการคำนวณดัชนีมวลกายสำหรับแมว (Feline Body Mass Index : FBMI) เนื่องจากวิธีการประเมินคะแนนร่างกายของแมว (Body Condition Scores : BCS) จำเป็นจะต้องใช้ความรู้และความเชี่ยวชาญทางการแพทย์ในการประเมินคะแนนร่างกายของแมว



รูปที่ 2.3 ตำแหน่งการวัดเพื่อคำนวณดัชนีมวลกายสำหรับแมว

โดย C คือ ความยาวรอบอก (ซม.)

L คือ ความยาวของขาส่วนหลังตั้งแต่หัวเข่าถึงข้อเท้า (ซม.)

ชื่อสามารถคำนวณ FBMI ได้จาก

$$\text{FBMI} = \left(\frac{\left(\frac{C}{0.7062} \right)}{0.9156} \right) - L$$

รูปที่ 2.4 สูตรคำนวณดัชนีมวลกายสำหรับแมว

ที่มา : Butterwick (2000)

แปลความหมาย ค่า FBMI < 15	หมายถึง ดัชนีมวลกายต่ำกว่าเกณฑ์มาตรฐาน
ค่า FBMI 15 - 29.9	หมายถึง ดัชนีมวลกายอยู่ในเกณฑ์มาตรฐาน
ค่า FBMI 30 - 42	หมายถึง ดัชนีมวลกายเกินเกณฑ์มาตรฐาน
ค่า FBMI > 42	หมายถึง ดัชนีมวลกายอยู่ในเกณฑ์อ้วน

2.2.2 สาเหตุที่ทำให้แมวน้ำหนักเกินเกณฑ์มาตรฐาน

Arcadewipad (2005) กล่าวว่า ปัจจัยเสี่ยงที่มีส่งผลให้แมวน้ำหนักเกินเกณฑ์มาตรฐาน

1. แมวมีอายุเพิ่มขึ้น ปัญหาของน้ำหนักตัวก็เพิ่มขึ้นเช่นกัน
2. แมวเพศเมียจะมีน้ำหนักเกินเกณฑ์มาตรฐานได้ง่ายกว่าแมวเพศผู้และถ้าหากทำหมันแล้ว ความเสี่ยงต่อภาวะน้ำหนักเกินเกณฑ์มาตรฐานจะสูงกว่าแมวที่ไม่ได้ทำหมันถึง 2 เท่า
3. พัฒนาระบบ การเผาผลาญพลังงานของแมวแต่ละตัวจะแตกต่างกันออกไปและสามารถถ่ายทอดทางพัฒนาระบบ
4. สายพันธุ์ แมวที่มีความเสี่ยงน้ำหนักเกินเกณฑ์มาตรฐานมากพบในพันธุ์ผสมสูงกว่าแมวสายพันธุ์แท้
5. พฤติกรรม การร้องขออาหารของแมว ส่งผลให้ผู้เลี้ยงให้อาหารและขนมเกินความจำเป็นต่อร่างกายของแมว
6. ที่อยู่อาศัย โดยปัจจุบันคนส่วนใหญ่นิยมอยู่ในคอนโดหรือหอพักที่มีบริเวณจำกัด ทำให้แมวมีพื้นที่สำหรับการเคลื่อนไหวร่างกายลดลง

2.2.3 ความเสี่ยงด้านสุขภาพของแมวน้ำหนักเกินเกณฑ์มาตรฐาน

Royal Canin (2023) กล่าวว่า เมื่อแมวที่มีน้ำหนักเกินเกณฑ์มาตรฐาน ร่างกายแมวจะเริ่มเก็บพลังงานส่วนเกินกลายเป็นไขมัน ก่อให้เกิดโรคดังนี้

1. โรคเบาหวาน
แมวที่มีน้ำหนักเกินเกณฑ์มาตรฐานมีความเสี่ยงที่จะมีสภาวะโรคเบาหวานซึ่งต้องฉีดอินซูลินทุกวัน บ่อยครั้งแมวที่เป็นโรคเบาหวานมีอาการดีขึ้นเมื่อน้ำหนักลดลง เพราะไขมันสะสมที่เป็นตัวการทำให้การควบคุมกลูโคสล้มเหลวได้หายไป
2. ระบบภูมิคุ้มกันบกพร่อง
ระบบภูมิคุ้มกันของแมวอาจบกพร่องเนื่องการดีม์น้ำน้อย ทำให้ส่งผลกระทบต่อระบบทางเดินปัสสาวะที่อาจติดเชื้อได้ จึงทำให้มีปัญหารื่องระบบภูมิคุ้มกันบกพร่อง

3. ตัว่วย

ร่างกายของแมวที่มีดัชนีมวลกายเกินเกณฑ์มาตรฐาน กระบวนการทางร่างกายมีความต้องการสารอาหารเพิ่มขึ้นจนเกินความจำเป็น จึงมีความเสี่ยงที่ตัวจะทำงานบกพร่องจนก่อให้เกิดภาวะตัว่วย

4. โรคผิวนัง

เมื่อแมวมีน้ำหนักเกินเกณฑ์มาตรฐานจะส่งผลให้ขนาดตัวใหญ่ขึ้น จึงทำความสะอาดขนด้วยตัวเองได้ยาก ซึ่งอาจนำไปสู่โรคผิวนัง

2.3 โภชนาการอาหารแมว

Purina (2021) ตามธรรมชาติแมวเป็นสัตว์กินเนื้อ ซึ่งแปลว่าสารอาหารหลักที่แมวต้องการจะอยู่ในเนื้อสัตว์ และแมวแต่ละวัยมีความต้องการสารอาหารที่ต่างกัน ดังนั้นสารอาหารที่แมวต้องการมีดังต่อไปนี้

1. โปรตีน

เพื่อช่วยในการสร้างกล้ามเนื้อและให้พลังงานเพียงพอ กับพัฒนาระบบในแต่ละวันของแมว

2. แคลเซียมและฟอฟอรัส

เพื่อการเสริมสร้างกระดูกและข้อต่อ ช่วยให้เคลื่อนไหวคล่องแคล่ว

3. ดีเอชเอ

เพื่อเสริมสร้างพัฒนาการทางสมองและการมองเห็นของแมว

4. ไขอาหารสูงและแคลอรีต่ำ

ลดการเกิดก้อนขน ช่วยเรื่องระบบการขับถ่ายอุจจาระเป็นก้อน

5. โอมega 3 และ 6

ช่วยบำรุงผิวนังและขนให้สุขภาพดี เงางาม

6. สารต้านอนุมูลอิสระจากวิตามิน A E และซีลีเนียม

ช่วยเสริมสร้างภูมิคุ้มกัน

2.3.1 การคำนวณโภชนาการของผลิตภัณฑ์

Lekcharoensuk (2021) การคำนวณปริมาณอาหารให้กับแมวต้องคำนึงถึงความต้องการ พลังงานในแต่ละวันของแมว จึงสามารถคำนวณปริมาณอาหารต่อวันสำหรับแมว และส่วนของฉลากอาหารแมวจะประกอบไปด้วย น้ำหนักของอาหาร จำนวนที่กินได้ ชื่อสูตรอาหาร คุณสมบัติ วันหมดอายุ

สารอาหารและส่วนประกอบ ตารางปริมาณการให้อาหาร คุณสมบัติอาหารโดยละเอียด และเลขทะเบียนอาหารสัตว์ ดังรูปที่ 2.4



รูปที่ 2.5 รายละเอียดในผลิตภัณฑ์ของอาหารแมว

ที่มา : ผลิตภัณฑ์ Royal Canin (2021)

โดยดูที่ข้อมูลจากผลิตภัณฑ์สามารถคำนวณหาโภชนาการของผลิตภัณฑ์ (Nutrition) ซึ่งคำนวณได้ 2 วิธี

1. กรณีที่ผลิตภัณฑ์ระบุพลังงานที่นำไปใช้ได้ ดังรูปที่ 2.6



รูปที่ 2.6 พลังงานที่สัตว์นำไปใช้ได้

ที่มา : ผลิตภัณฑ์ Silver สูตร fit and firm (2023)

การคำนวณหาโภชนาการของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้ โดยใช้ข้อมูลแคลอรี่ (Calorie content) และปริมาณอาหารต่อการคำนวณ (Amount of food) สามารถคำนวณได้ดังรูปที่ 2.7

$$\text{Nutrition (\%)} = \frac{\text{Calorie content (Kcal.)}}{\text{Amount of food (g.)}}$$

รูปที่ 2.7 สูตรการคำนวณโภชนาการของผลิตภัณฑ์

ที่มา : เฉลิมพล เล็กเจริญสุข (2021)

2. กรณีที่ผลิตภัณฑ์ไม่ระบุพัลังงานที่นำไปใช้ได้

การคำนวณหาโภชนาการจากคุณภาพอาหารสัตว์ทางเคมี โดยมีข้อมูลสารอาหารโปรตีน (%) ไขมัน (%) ไฟเบอร์ (%) และความชื้น (%) ดังรูปที่ 2.8

คุณภาพของอาหารสัตว์ทางเคมี GUARANTEED ANALYSIS			
โปรตีน / Protein	30	(Min)	
ไขมัน / Fat	9	(Min)	
ไฟเบอร์ / Fiber	5	(Max)	
ความชื้น / Moisture	10	(Max)	

รูปที่ 2.8 คุณภาพของอาหารสัตว์ทางเคมี

ที่มา : ผลิตภัณฑ์ Silver สูตร fit and firm (2023)

ข้อมูลร้อยละของโปรตีน ไขมัน ไฟเบอร์ และความชื้นสามารถดูได้จากฉลากของผลิตภัณฑ์ แต่ข้อมูลร้อยละของคาร์บอไฮเดรตจะทราบเมื่อนำข้อมูลร้อยละของโปรตีน ไขมัน ไฟเบอร์ และความชื้น โดยใช้สูตรคำนวณดังรูปที่ 2.9

$$\text{Carbohydrate (\%)} = 100 - (\text{Protein (\%)} + \text{Fat (\%)} + \text{Fiber (\%)} + \text{Moisture (\%)})$$

รูปที่ 2.9 สูตรคำนวณร้อยละของคาร์โบไฮเดรต

ที่มา : เฉลิมพล เล็กเจริญสุข (2021)

จากนั้นนำข้อมูลร้อยละของ โปรตีน ไขมัน และคาร์โบไฮเดรตมาคำนวณโภชนาการของผลิตภัณฑ์ได้ดังรูปที่ 2.10

$$\text{Nutrition (\%)} = \frac{(\text{Protein (\%)} \times 3.5) + (\text{Fat (\%)} \times 8.5) + (\text{Carbohydrate (\%)} \times 3.5)}{100}$$

รูปที่ 2.10 สูตรคำนวณโภชนาการของผลิตภัณฑ์

ที่มา : เฉลิมพล เล็กเจริญสุข (2021)

2.3.2 การคำนวณพลังงานของแมว

Butterwick (2000) สาเหตุที่ปริมาณพลังงานในอาหารแมวมีความสำคัญ เพราะค่าโภชนาการของผลิตภัณฑ์และดัชนีมวลกายสามารถคำนวณได้ว่าแมวต้องได้รับปริมาณพลังงานเท่าไร จากนั้นสัตวแพทย์สามารถนำข้อมูลปริมาณพลังงานที่แมวควรได้รับ มาใช้ในการคำนวณปริมาณอาหารที่ควรได้รับต่อวัน ซึ่งการคำนวณพลังงานสามารถแบ่งได้เป็น 2 ส่วน คือ คำนวณปริมาณพลังงานที่แมวควรได้รับ (Resting Energy Requirement : RER) และคำนวณปริมาณพลังงานที่แมวควรได้รับต่อวัน (Daily Energy Requirement : DER) โดยเริ่มจากคำนวณปริมาณพลังงานที่แมวควรได้รับและนำไปแทนค่าในการคำนวณพลังงานที่แมวควรได้รับต่อวัน สามารถคำนวณได้ดังรูปที่ 2.11

$$\text{RER (calories/day)} = 70 \times \text{weight (kg.)}^{0.75}$$

รูปที่ 2.11 สูตรคำนวณปริมาณพลังงานที่แมวควรได้รับ
ที่มา : Butterwick (2000)

จากนั้นคำนวณพลังงานที่แมวควรได้รับต่อวัน จากสูตรดังรูปที่ 2.12

$$\text{DER (kcal/kg.)} = \text{Factor} \times \text{RER (calories/day)}$$

รูปที่ 2.12 สูตรคำนวณปริมาณพลังงานที่แมวควรได้รับต่อวัน
ที่มา : Butterwick (2000)

กำหนดค่า Factor ใช้ “1.2” เมื่อมีค่า FBMI ≥ 30

Factor ใช้ “1.4” เมื่อมีค่า FBMI < 30

2.3.3 การคำนวณปริมาณอาหารให้เหมาะสมกับแมว

การคำนวณปริมาณของอาหารที่ควรได้รับต่อวัน (Recommended Daily Intakes : RDI) โดยนำค่าจากการคำนวณโภชนาการของผลิตภัณฑ์ ในข้อที่ 2.3.1 และพลังงานที่แมวควรได้รับต่อวัน ในข้อที่ 2.3.2 มาคำนวณหาปริมาณของอาหารที่ควรได้รับต่อวัน จากสูตรดังรูปที่ 2.13

$$\text{RDI (g./day)} = \frac{\text{DER (kcal/kg.)}}{\text{Nutrition (\%)}}$$

รูปที่ 2.13 สูตรคำนวณปริมาณของอาหารที่ควรได้รับต่อวัน
ที่มา : Butterwick (2000)

2.4 หลักการเกี่ยวกับการออกแบบแอปพลิเคชัน

Prasertsak (2022) ในปัจจุบันแนวทางการออกแบบประสบการณ์ของผู้ใช้งาน (User Experience : UX) และการออกแบบส่วนติดต่อระหว่างผู้ใช้งานกับผลิตภัณฑ์ (User Interface : UI) มีความสำคัญมากขึ้น เนื่องจากในปัจจุบันมุ่งเน้นการที่จะสื่อสารกันมากขึ้น การออกแบบต้องทำอย่างไร เพื่อทำให้ใช้งานง่ายโดยไม่ต้องเปิดคู่มือการใช้งาน

2.4.1 การออกแบบประสบการณ์ของผู้ใช้งาน

การออกแบบประสบการณ์ของผู้ใช้งานเพื่อให้ผู้ใช้เกิดความพึงพอใจสูงสุด เช่น ใช้งานง่าย มีลำดับขั้นตอนชัดเจนเป็นการสร้างความรู้สึกใจจำให้ผู้ใช้งานอยากรีบกลับมาใช้งานอีก มีกระบวนการหรือแนวคิดในการออกแบบได้แก่

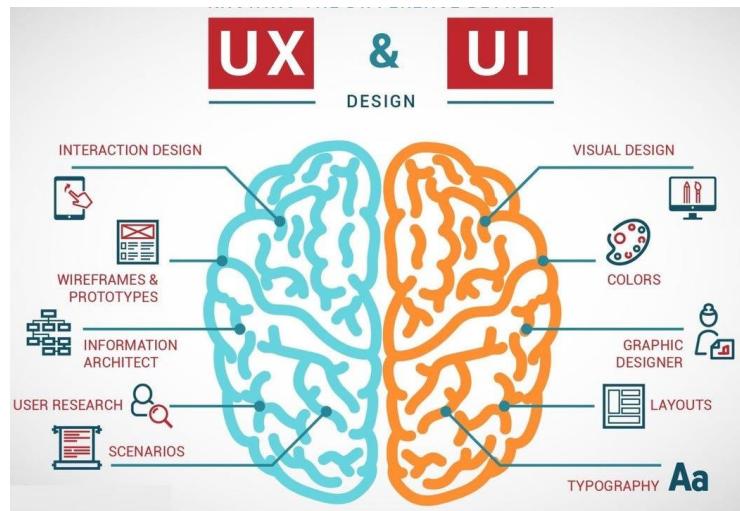
- 1) Design Thinking การคิดเชิงออกแบบ เป็นกระบวนการคิดที่ให้ความสำคัญกับการทำความเข้าใจปัญหาของผู้ใช้ เพื่อให้สิ่งที่ออกแบบมาตอบสนองต่อความต้องการของผู้ใช้งาน สูงสุด
- 2) Service Design การออกแบบบริการ จะเน้นการบริการโดยคำนึงถึงประสบการณ์ของผู้ที่เกี่ยวข้องในงานทั้งหมด เพื่อสร้างระบบที่มีประสิทธิภาพและมีความต่อเนื่องกันตั้งแต่ต้นจนจบกระบวนการ
- 3) Design Sprint กระบวนการเพื่อแก้ไขปัญหาที่เกิดขึ้น เช่น ผลิตภัณฑ์ที่ออกแบบสู่ตลาดอย่างเป็นระบบ ซึ่งจุดเด่นคือ ความรวดเร็วโดยจะใช้เวลาเพียงไม่กี่วันเพื่อแก้ไขปัญหาที่เกิดขึ้น
- 4) User Centered Design กระบวนการออกแบบที่ให้ความสำคัญกับผู้ใช้งานเป็นหลัก โดยผู้ใช้งานจะเข้ามามีส่วนร่วมในการประเมินผลการออกแบบ

2.4.2 การออกแบบส่วนติดต่อระหว่างผู้ใช้งานกับผลิตภัณฑ์

การออกแบบส่วนติดต่อระหว่างผู้ใช้งานกับผลิตภัณฑ์ ซึ่งมุ่งเน้นไปที่หน้าการออกแบบ เช่น การวางแผน ปุ่ม หรือขนาดตัวอักษร มีองค์ประกอบในการออกแบบ ได้แก่

- 1) Information Design กระบวนการออกแบบที่เน้นนำเสนอข้อมูลเพื่อสร้างความเข้าใจ และเกิดการเรียงลำดับข้อมูลได้อย่างครบทั่ว
- 2) Interaction Design กระบวนการออกแบบโดยคำนึงถึงปฏิสัมพันธ์ระหว่างผู้ใช้งานกับวัตถุที่เน้นการตอบสนองที่ถูกต้องและผู้ใช้งานสามารถเข้าถึงเป้าหมายได้อย่างรวดเร็ว

- 3) Information Architecture เป็นการจัดเรียงข้อมูลอย่างเป็นระบบและการจัดกลุ่มข้อมูลให้มีความถูกต้อง เพื่อให้ผู้ใช้งานรู้ลำดับการเข้าถึงข้อมูลได้อย่างรวดเร็ว เช่น การทำแผนผังเว็บไซต์ (Sitemap)
- 4) Visual Design เน้นการนำเสนอภาพเพื่อให้เกิดความสวยงาม เช่น การจัดวางตำแหน่ง การใช้สี และการใช้ตัวอักษร ในการนำเสนอให้ผู้อ่านเข้าใจได้ง่าย เช่น การทำอินโฟกราฟิก (Infographics)
- 5) Human Computer Interaction การออกแบบปฏิสัมพันธ์ระหว่างมนุษย์กับคอมพิวเตอร์ ซึ่งเป็นพื้นฐานในการเรียนรู้การออกแบบ UX UI ที่ดี เพราะการพัฒนานั้นเริ่มต้นตั้งแต่การใช้งานคอมพิวเตอร์ผ่านคอมมานด์ไลน์ (Command line) ที่มีความยุ่งยากซับซ้อนจนมาถึงการใช้ภาพเป็นตัวประสานกับผู้ใช้งาน (Graphical user interface : GUI) ที่มีสีสัน หน้าตาที่สวยงาม และใช้ง่ายมากยิ่งขึ้น โดยในปัจจุบันยังคงมีการพัฒนาอยู่อย่างต่อเนื่อง



รูปที่ 2.14 แนวคิดของ UX และ UI

ที่มา : Ratiheldina (2022)

2.5 ทฤษฎีการเขียนแอปพลิเคชันด้วยภาษาดาร์ต

ภาษาดาร์ตถูกพัฒนาขึ้นโดยมีวัตถุประสงค์เพื่อให้เป็นภาษาสำหรับโปรแกรมประยุกต์ ซึ่งเป็นภาษาที่ถูกพัฒนาโดยغوเกิล (Google) และได้รับความสนใจเป็นอย่างมากในวงการพัฒนาโปรแกรมประยุกต์ (Walrath & Ladd, 2012)

โครงสร้างของภาษาดาร์ต คล้ายกับภาษา C C++ และ Java โดยที่จะมีความเป็นภาษาแบบโครงสร้าง (Structure Programming) และเป็นรูปแบบภาษาโปรแกรมเชิงวัตถุ (Object Oriented Programming) ซึ่งจะประกอบไปด้วย

1. ชนิดของข้อมูล

ตารางที่ 2.1 แบบชนิดของข้อมูลพื้นฐานในภาษาดาร์ต

ชนิดข้อมูล	ชนิดข้อมูล	รูปแบบข้อมูล
bool	ค่าตรรกศาสตร์	true / false
double	ตัวเลขที่มีทศนิยม	2.88, 1.2, 3.999
int	ตัวเลขที่ไม่มีทศนิยม	128, 5, 100
String	ชนิดข้อมูลที่เป็นข้อความ	"ซัชชาติ", "แต่งโม", "Teerasak"
List<type>	ชนิดข้อมูลที่เป็น array โดยมี key เป็นตัวเลขเรียงลำดับเริ่มต้นที่ 0	["แอปเปิล","มะนาว","มันผั่ง"]
Map <index,value>	ชนิดข้อมูลที่เป็น object โดยมาเป็นชุดข้อมูลที่มี key และ value จับคู่กัน	{firstName:"bas", lastName:"suksai",age:28};
Dynamic	ตัวแปรที่สามารถเปลี่ยนค่าได้เปลี่ยนชนิดข้อมูลได้	55,35,"Teerasak",Fase,True

2. พัฟ์ชันพารามิเตอร์

- 1) พารามิเตอร์ที่จำเป็น (Required parameters) เวลาเรียกใช้ฟังก์ชัน (Function) ต้องแนบพารามิเตอร์ (Parameter)
- 2) พารามิเตอร์เพื่อเลือก (Optional parameters) เวลาเรียกใช้ฟังก์ชันจะแนบพารามิเตอร์ หรือไม่ก็ได้

3. อ็อปเจ็ค

อ็อปเจ็ค (Object) จะมี 2 ส่วน คือ ลักษณะของอ็อปเจ็คเรียกว่า แอตทริบิวต์ (Attribute) และสิ่งที่อ็อปเจ็คกระทำหรือดำเนินการเรียกว่า เมธอด (Method)

4. คอนสตรัคเตอร์

คอนสตรัคเตอร์ (Constructor) จะทำงานอัตโนมัติ เมื่อสร้างอ็อปเจ็คขึ้นมา

5. ภาษาดาร์ตบันฟลัตเตอร์

นักพัฒนานิยมใช้เพื่อสร้างหน้าการแสดงผลสำหรับผู้ใช้งานแอปพลิเคชัน โดยสามารถใช้งานได้ หลายแพลตฟอร์มด้วยฐานรหัสเดียว และข้อดีดเด่นของการใช้ฟลัตเตอร์ (Flutter) มีดังต่อไปนี้

- 1) ประสิทธิภาพในการเข้าถึงการใช้งานได้ง่ายและรวดเร็ว ผ่านอุปกรณ์โฮสต์ (Host)
- 2) การสร้างภาพแบบจำลอง (Render) ที่รวดเร็ว สม่ำเสมอ และปรับแต่งได้ ซึ่งใช้ไลบรารี (Library) ที่ไม่เสียเงินของกูเกิลในการแสดงผล ทำให้ผู้ใช้เห็นภาพที่สอดคล้องกันไม่ว่า จะใช้แพลตฟอร์มใดในการเข้าถึงแอปพลิเคชันก็ตาม
- 3) เครื่องมืออัตโนมัติ (Hot reload) นักพัฒนาสามารถดูตัวอย่างการเปลี่ยนแปลงโค้ด (Code) ได้ทันทีโดยไม่ต้องรีสตาร์ต (Restart)

2.6 การศึกษาหลักการเชื่อมต่อเครื่องชั้งน้ำหนักกับแอปพลิเคชัน

Chansuwan (2020) จากบทความเรื่อง เทคนิคการใช้เครื่องชั้งในห้องปฏิบัติการ ได้กล่าวไว้ว่า เครื่องชั้งน้ำหนักสามารถแบ่งออกเป็น 2 ประเภท ดังนี้

2.6.1 ประเภทของเครื่องชั้ง

2.6.1.1 เครื่องชั้งระบบกล

เครื่องชั้งจะใช้หลักการเปรียบเทียบน้ำหนักระหว่างสิ่งที่ต้องการทราบน้ำหนักกับน้ำหนักมาตรฐาน โดยอาศัยการสมดุลของคาน เครื่องชั้งประเภทนี้ได้แก่ เครื่องชั้งสองแขน เครื่องชั้งจานเดียว และเครื่องชั้งสปริง

2.6.1.2 เครื่องชั่งไฟฟ้า

1) เครื่องชั่งไฟฟาระบบอิเล็กทรอนิกส์

เครื่องชั่งอาศัยหลักการเปลี่ยนแปลงค่าความต้านทานของตัวนำไฟฟ้าที่เรียกว่า สเตรน เกจ (Strain Gauge) จำนวน 4 ชิ้น ติดอยู่กับส่วนรับน้ำหนักที่ฐานเครื่องชั่งซึ่งสามารถยืดหยุ่นได้ เมื่อไม่มีวัตถุบนajanชั่งสเตรนเกจทั้ง 4 ชิ้น จะมีค่าความต้านทานไฟฟ้าเท่ากันทุกตัวทำให้ค่าความต่างศักย์ไฟฟ้ามีค่าเป็น 0 เมื่อวางวัตถุบนajanชั่งสเตรนเกจ ทั้ง 4 ชิ้น จะถูกเชื่อมต่อกันเป็นวงจรไฟฟ้า แรงดึงของวัตถุบนajanชั่งจะทำให้ส่วนรับน้ำหนักที่ฐานเครื่องชั่งและสเตรนเกจทั้ง 4 ชิ้น มีการยืดหดไม่เท่ากันทำให้มีความต่างศักย์ไฟฟ้าเพิ่มขึ้นระหว่างสเตรนเกจทั้ง 4 ชิ้น โดยอุปกรณ์อิเล็กทรอนิกส์ภายในเครื่องชั่งจะทำการแปลงค่าความต่างศักย์ไฟฟ้าให้เป็นตัวเลขแสดงเป็นน้ำหนักกว่าตุบันหน้าจอเครื่องชั่ง

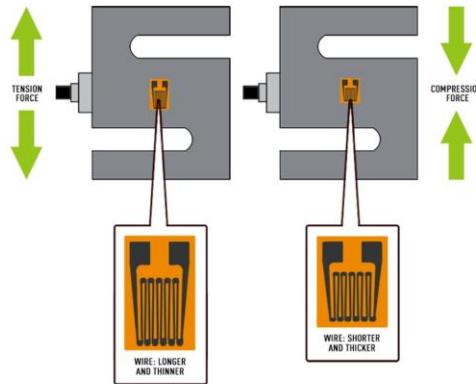
2) เครื่องชั่งไฟฟาระบบแม่เหล็กไฟฟ้า

หลักการทำงานของเครื่องชั่งประเภทนี้คือ ขดลวดตัวนำที่ติดอยู่ใต้ajanชั่งจะวางอยู่ในตำแหน่งที่มีสนามแม่เหล็กภายในเครื่องชั่ง เมื่อไม่มีวัตถุอยู่บนajanชั่ง ajanชั่งจะอยู่ในลักษณะสมดุล แต่เมื่อวางวัตถุบนajanชั่ง ระบบไฟฟ้าภายในเครื่องชั่งจะจ่ายกระแสไฟฟ้าผ่านขดลวดตัวนำใต้ajanชั่ง ทำให้เกิดแรงแม่เหล็กไฟฟ้าต้านการเลื่อนต่ำลงของajanชั่งเพื่อให้janชั่งอยู่ในลักษณะสมดุล ปริมาณกระแสไฟฟ้าที่ใช้ทำให้janชั่งอยู่ในลักษณะสมดุลแล้วถูกแปลงให้เป็นค่าความต่างศักย์ไฟฟ้าและแสดงค่าเป็นตัวเลขของน้ำหนักวัตถุบนหน้าจอเครื่องชั่ง

2.6.2 หลักการทำงานเครื่องชั่ง

2.6.2.1 หลักการทำงานของโหลดเซลล์

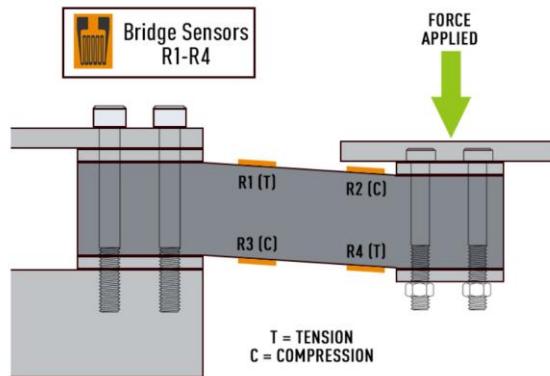
โหลดเซลล์ (Load Cell) เป็นอุปกรณ์ใช้วัดน้ำหนักโดยมีหลักการคือ นำสเตรนเกจไปแปะติดกับแท่งเหล็กแล้วยึดปลายด้านหนึ่งของแท่งเหล็กไว้กับพื้น จากนั้นที่ปลายอีกด้านใช้วางสิ่งของที่ต้องการชั่งน้ำหนัก เมื่อวางสิ่งของลงไปแล้ว จะทำให้แท่งเหล็กงอเล็กน้อย การงอจะไปทำให้สเตรนเกจถูกยืดออกและแผ่นทองแดงมีความยาวมากขึ้น เมื่อความยาวของทองแดงมากขึ้นจะส่งผลต่อความต้านทานของสเตรนเกจให้มีค่ามากขึ้น ดังรูปที่ 2.15



รูปที่ 2.15 การยืดหดของสเตรนเกจ

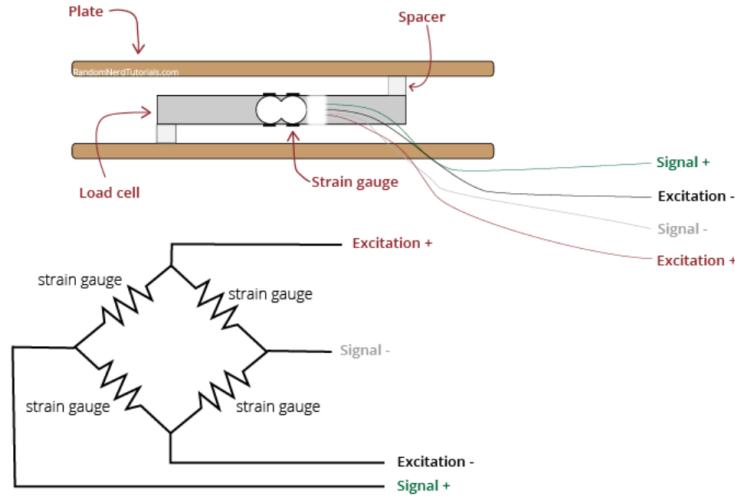
ที่มา : Artronshop (2021)

วงจรบริดจ์ คือ วงจรที่นำสเตรนเกจ 4 แผ่น มาต่อ กันเป็นวงจรแล้วและติดลิงไปกับแท่งเหล็ก ด้านบน 2 แผ่น และด้านล่าง 2 แผ่น เมื่อแท่งเหล็กจะส่งผลให้สเตรนเกจ 2 แผ่น ยืดออกในขณะที่ สเตรนเกจ อีก 2 แผ่นหดเข้า การเปลี่ยนแปลงค่าความต้านทานสเตรนเกจทั้ง 4 แผ่น ทำให้สามารถ ตรวจจับการเปลี่ยนแปลงค่าความต้านทานได้ง่ายขึ้น ดังรูปที่ 2.16



รูปที่ 2.16 วงจรบริดจ์บนโลดเซลล์

ที่มา : Artronshop (2021)

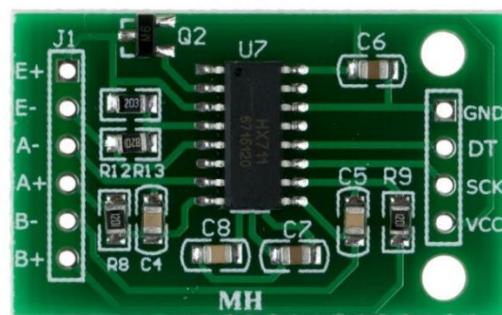


รูปที่ 2.17 ส่วนประกอบของเครื่องชั่ง

ที่มา : Artronshop (2021)

2.6.2.2 หลักการทำงานของ HX711

อุปกรณ์ที่ใช้สำหรับการขยายสัญญาณจากโหลดเซลล์ เพื่อให้สัญญาณที่รับมีความแม่นยำ และความเสถียรสูงพอที่จะนำไปใช้งานต่อได้ โดยส่วนใหญ่จะใช้ในการช่วยเพิ่มแรงสัญญาณในการวัดน้ำหนัก วัตถุที่สเตรนเกจของโหลดเซลล์ส่งออกมาก ซึ่งจะทำให้สามารถอ่านค่าน้ำหนักได้ง่ายขึ้น

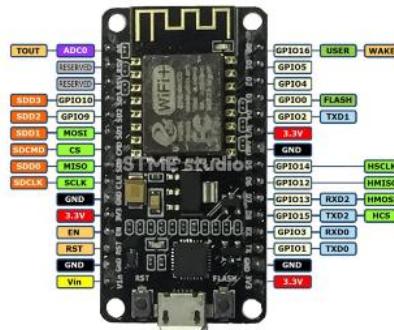


รูปที่ 2.18 HX711 Amplifier

ที่มา : Artronshop (2021)

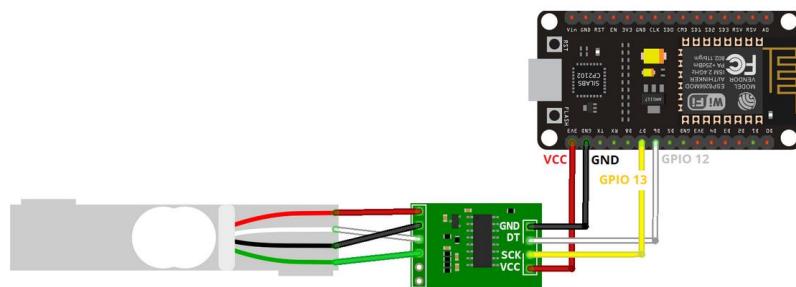
2.6.2.3 หลักการทำงานของ ESP8266

ไมโครชิปขนาดเล็ก (Microchip) หรือจะเรียกว่าเป็นสมองกลแบบฝังตัวชนิดต่อสัญญาณเชื่อมต่ออินเทอร์เน็ตแบบไร้สายได้ (WiFi) ซึ่งผลิตโดย Espressif Systems ถูกใช้อุปกรณ์ที่มีชื่อว่า IoT และอุปกรณ์สมาร์ทโฮม (Smart home) ต่างๆ ESP8266 ถูกนำไปใช้งานในหลายรูปแบบตั้งแต่การทำอุปกรณ์สมาร์ทโฮมจนถึงใช้งานในระดับอุตสาหกรรม จากการความสามารถในการใช้งานร่วมกับเซ็นเซอร์ (Senser) ต่าง ๆ เช่น วัดอุณหภูมิ วัดความชื้น และวัดกระแสไฟฟ้า



รูปที่ 2.19 ESP8266

ที่มา : Robotsiam (2021)



รูปที่ 2.20 การเชื่อมต่ออุปกรณ์การทำเครื่องซั่ง

ที่มา : Robotsiam (2021)

2.6.3 การเชื่อมต่อเครื่องชั้งน้ำหนักกับแอปพลิเคชัน

การเชื่อมต่อระหว่างเครื่องชั้งน้ำหนักและแอปพลิเคชันด้วยการใช้ฐานข้อมูลไฟล์เบส (Firebase) ช่วยให้นักพัฒนาสามารถรับส่งข้อมูลระหว่างแอปพลิเคชันและเครื่องชั้งแบบเรียลไทม์ (Real time) มีการรักษาความปลอดภัยในการสื่อสารระหว่างอุปกรณ์



รูปที่ 2.21 การเชื่อมต่อเครื่องชั้งผ่านอินเทอร์เน็ต

ที่มา : Hackanons (2018)

2.7 งานวิจัยที่เกี่ยวข้อง

Somboon (2023) ได้ศึกษาเรื่อง มุ่งมองของผู้เลี้ยงแมวในการเลือกซื้ออาหารแมวสำเร็จรูป ระดับพรีเมียม จากผลการวิจัยพบว่าผู้เลี้ยงแมวให้ความสำคัญในมุ่งมองของความพรีเมียมโดยคำนึงถึง ส่วนผสมของอาหาร วัตถุดิบ และคุณค่าของสารอาหารที่ครบถ้วนให้เหมาะสมกับแมวแต่ละช่วงวัย เพราะ แมวแต่ละสายพันธุ์มีการดูแลที่แตกต่างกันผู้ประกอบการอาจจะต้องทำสูตรอาหารที่เน้นอย่างโดยย่าง หนึ่งเป็นพิเศษ เช่น เน้นโปรตีนที่มีผลต่อการเจริญเติบโตสำหรับแมวเด็กรวมถึงต้องเป็นสารอาหารที่มี ประโยชน์ต่อแมว โดยมองจากส่วนประกอบของอาหารแมวเป็นหลัก

Supadanaisorn et al., (2019) ได้ศึกษาเรื่อง การออกแบบและสร้างระบบวัดดัชนีมวลกาย อัตโนมัติแสดงผลผ่านเครือข่ายอินเทอร์เน็ต ซึ่งใช้ไมโครคอนโทรลเลอร์ (Microcontroller) ที่สามารถ เชื่อมตอกับสัญญาณอินเทอร์เน็ตแบบไร้สายได้ (NodeMCU) ประกอบด้วยตัวบอร์ด (Development

Kit) และซอฟต์แวร์บนบอร์ด (Firmware) โมดูล (Module) ESP8266 เป็นส่วนสำคัญที่ใช้เชื่อมต่อ มือถือ ด้วยกันหลายรุ่น ซึ่งการใช้งานมีลักษณะคล้ายกันไม่แตกต่างกันมากนัก NodeMCU มีลักษณะคล้ายกับ Arduino คือ มีช่องรับส่งสามารถเขียนโปรแกรมควบคุมอุปกรณ์ได้ โดยไม่ต้องผ่านอุปกรณ์อื่น ๆ และใช้ โปรแกรม Arduino IDE ในการเขียนโปรแกรมได้

บทที่ 3

การออกแบบ การสร้าง และการทำงาน

3.1 กล่าวนำ

การออกแบบ การสร้าง และการทำงานของแอปพลิเคชันซึ่งเนื้อหาของปริญญา妮พนธ์ในบทนี้ได้กล่าวถึง การออกแบบแอปพลิเคชัน การเขียนแอปพลิเคชัน การเชื่อมต่อเครื่องซึ่งน้ำหนักกับแอปพลิเคชัน และการทำงานของแอปพลิเคชัน

3.2 การออกแบบแอปพลิเคชัน

3.2.1 การออกแบบในส่วนของ UX

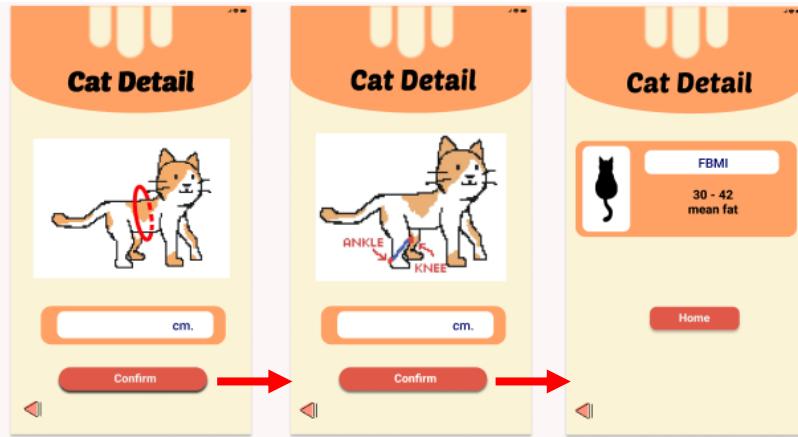
การออกแบบประสบการณ์ของผู้ใช้งานเพื่อให้ผู้ใช้เกิดความพึงพอใจจึงต้องออกแบบให้แอปพลิเคชันมีการเรียงขั้นตอนการทำงานที่ทำให้ผู้ใช้งานเข้าใจได้ง่าย (Prasertsak, 2022)

การออกแบบหน้าลงทะเบียนบัญชีของแมวนั้นได้ให้ความสำคัญในเรื่องการใช้งานที่สะดวกและง่ายต่อความเข้าใจ โดยเมื่อผู้ใช้งานได้ทำการกรอกข้อมูลเพื่อสร้างบัญชีของแมวเรียบร้อยแล้ว แอปพลิเคชันจะให้ผู้ใช้งานสามารถทำการตรวจสอบความถูกต้องของข้อมูลอีกรอบ ดังรูปที่ 3.1



รูปที่ 3.1 หน้าลงทะเบียนบัญชีของแมว

การลงทะเบียนบัญชีของแมวเสร็จสมบูรณ์แล้วแอปพลิเคชันจะให้ผู้ใช้งานกรอกข้อมูลในการคำนวณดัชนีมวลกายก่อน เพราะหากไม่ทำเช่นนี้แล้วผู้ใช้งานอาจจะข้ามไปคำนวณปริมาณอาหารที่ควรได้รับต่อวันเลย ซึ่งจะทำให้แอปพลิเคชันไม่สามารถคำนวณได้ เพราะขาดข้อมูลดัชนีมวลกายซึ่งเป็นส่วนหนึ่งของการคำนวณปริมาณอาหารที่ควรได้รับต่อวัน ดังรูปที่ 3.2



รูปที่ 3.2 หน้าคำนวณดัชนีมวลกาย

หน้าเก็บสูตรอาหารนั้นได้ถูกออกแบบมาในลักษณะเช่นเดียวกันกับหน้าคำนวณดัชนีมวลกาย ที่ต้องการทราบข้อมูลสูตรอาหารก่อนนำไปคำนวณปริมาณอาหารที่ควรได้รับต่อวัน ดังรูปที่ 3.3



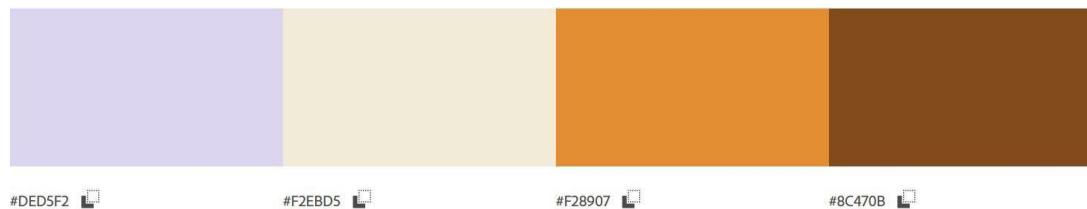
รูปที่ 3.3 หน้าบันทึกสูตรอาหาร

แอปพลิเคชันนั้นจะคำนวณว่าแม่ต้องการรับพลังงานเท่าไรได้หลังจากผู้ใช้งานได้ทำการกรอกค่าโภชนาการของผลิตภัณฑ์และดัชนีมวลกายเรียบร้อยแล้ว เมื่อแอปพลิเคชันทราบแล้วว่าแม่ต้องการรับพลังงานเท่าไรก็จะนำไปคำนวณปริมาณอาหารที่ควรได้รับต่อวันในขั้นตอนต่อไป (Butterwick, 2022)

3.2.2 การออกแบบในส่วนของ UI

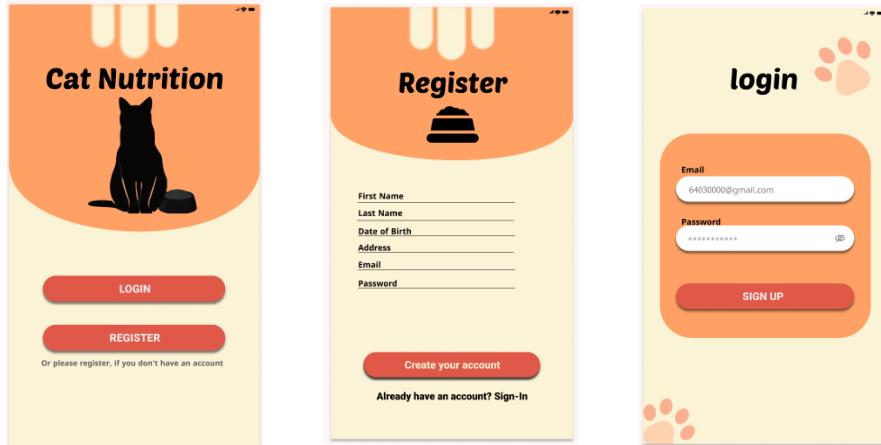
การนำปฏิสัมพันธ์ระหว่างผู้ใช้กับผลิตภัณฑ์และสุนทรียภาพทางสายตา มาใช้ในการออกแบบช่องกรอกข้อมูล ปุ่มกด การวางแผนหน่วงข้อความในแต่ละหน้า และการทำภาพกราฟิก (Prasertsak, 2022)

แอปพลิเคชันได้ถูกออกแบบโดยใช้การผสมผสานกันระหว่างสีโทนร้อน อย่าง สีส้ม สีครีม สีน้ำตาล และสีโทนเย็นอย่างสีม่วง โดยสัดส่วนของสีที่ใช้ในแอปพลิเคชันส่วนใหญ่นั้นจะเป็นสีโทนร้อนอย่าง สีส้ม สีครีม และสีน้ำตาลเป็นหลัก เนื่องจากทางผู้วิจัยได้รับแรงบัลดาลใจจากลักษณะสีของแมวบางชนิดที่มีสีส้มทั้งตัวเป็นเอกลักษณ์ และวิจัยได้เลือกสีโทนเย็นอย่างสีม่วงที่เป็นสีตรงข้ามเพื่อเน้นความสำคัญของผลลัพธ์การคำนวณ และง่ายต่อการสังเกต



รูปที่ 3.4 สีของแอปพลิเคชัน

หน้าตาของแอปพลิเคชันได้ถูกออกแบบโดยใช้ลักษณะเฉพาะตัวของแมวสอดแทรกเข้าไป เช่น การใช้รอยเท้าแมวในพื้นหลังของแอปพลิเคชัน ลายทางสีของขนแมวบนแถบแอปบาร์ของแอปพลิเคชัน ความโค้งมนของกรอบข้อความและปุ่มกดต่าง ๆ ที่นำมาจากลักษณะพฤติกรรมของแมวที่มีความซื่อสัตย์ เป็นมิตร ซึ่งลักษณะของการแสดงผลที่มีความโค้งมนนี้ทำให้ดูสวยงามและเข้าถึงง่ายขึ้น อีกทั้งการแสดงข้อความหัวข้อด้านบนแอปบาร์ช่วยทำให้ผู้ใช้งานเข้าใจโดยง่ายว่ากำลังใช้งานแอปพลิเคชันหน้าอะไรอยู่ ดังรูปที่ 3.5



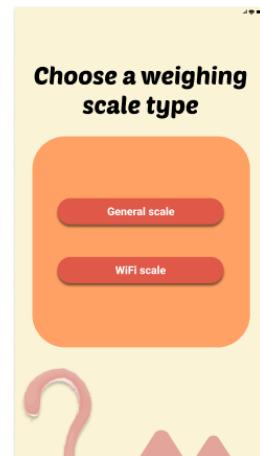
ก. หน้าเข้าสู่ระบบ

ข. หน้าลงทะเบียนผู้ใช้งาน

ค. หน้าเข้าสู่ระบบผู้ใช้งาน

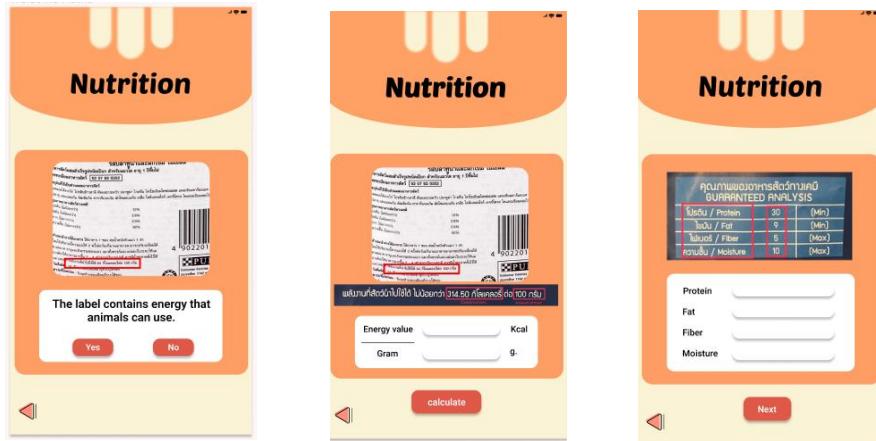
รูปที่ 3.5 หน้าเข้าสู่ระบบผู้ใช้งาน

หน้าเลือกเครื่องชั่งน้ำหนัก ในหน้านี้ประกอบด้วย การเลือกเครื่องชั่งทั่วไปและเครื่องชั่งเชื่อมต่อไร้สาย ดังรูปที่ 3.6



รูปที่ 3.6 หน้าเลือกเครื่องชั่งน้ำหนัก

เครื่องชั่งน้ำหนักหลังจากที่ถูกเลือกเรียบร้อยแล้ว ผู้ใช้งานจะสามารถคำนวณโภชนาการของผลิตภัณฑ์ได้ ซึ่งมีรูปแบบการคำนวณ 2 รูปแบบด้วยกันโดยมีเงื่อนไขกำหนดไว้ คือ ผลิตภัณฑ์ที่ระบุพลังงานที่สัตว์นำไปใช้ได้และผลิตภัณฑ์ที่ไม่ระบุพลังงานที่สัตว์นำไปใช้ได้ ดังรูปที่ 3.7



ก. หน้าเลือกการ
คำนวณโภชนาการ

ข. หน้าคำนวณโดยฉลาก
ระบุพลังงานที่สัตว์
นำไปใช้ได้

ค. หน้าคำนวณโดยฉลาก
ไม่ระบุพลังงานที่สัตว์
นำไปใช้ได้

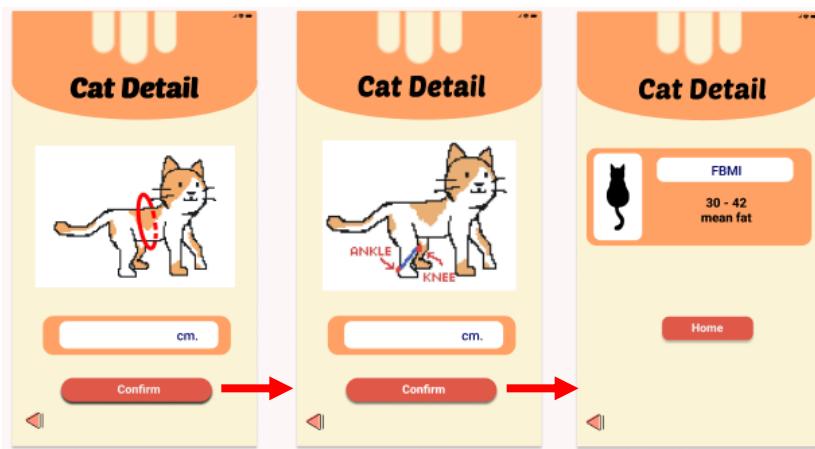
รูปที่ 3.7 หน้าคำนวณโภชนาการของผลิตภัณฑ์

หน้าการใช้งานหลักของแอปพลิเคชัน เมื่อผู้ใช้งานทำการคำนวณโภชนาการของผลิตภัณฑ์เสร็จเรียบร้อยแล้วจะถูกแอปพลิเคชันพามาหน้านี้ เพื่อเพิ่มบัญชีแมวโดยกดที่ปุ่ม Add Profile Cat ดังรูปที่ 3.8



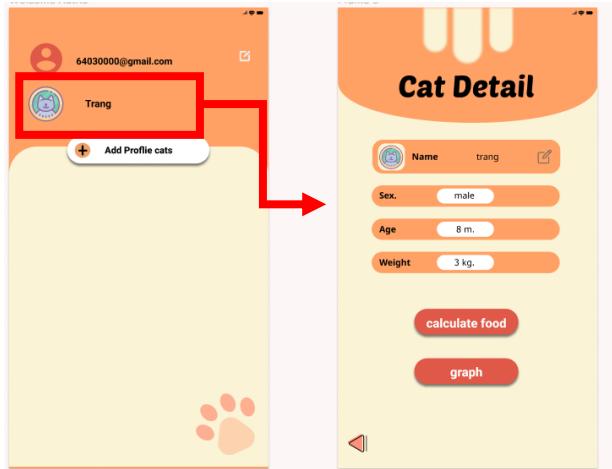
รูปที่ 3.8 หน้าลงทะเบียนบัญชีแมว

แอปพลิเคชันจะให้ผู้ใช้งานทำการวัดร่างกายของแมวเพื่อคำนวณดัชนีมวลกาย เป็นลำดับต่อมา
ดังรูปที่ 3.9



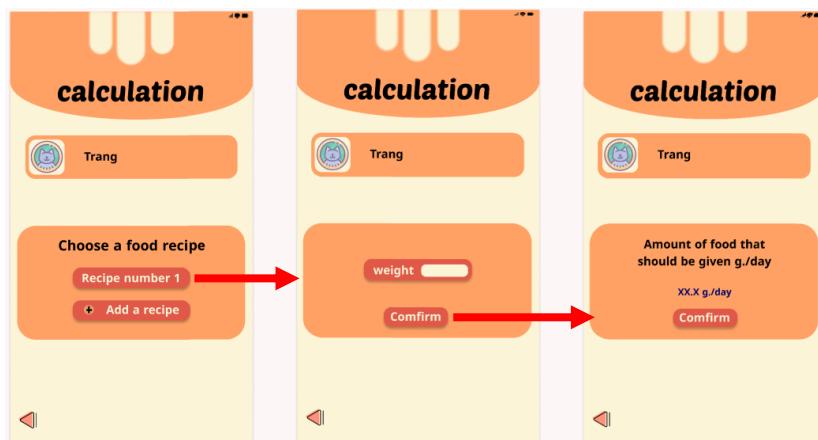
รูปที่ 3.9 หน้าคำนวณดัชนีมวลกาย

หน้าหลักของแอปพลิเคชัน ผู้ใช้งานจะเห็นบัญชีของแมวแสดงอยู่ ซึ่งผู้ใช้งานสามารถทำการเข้าไปแก้ไขข้อมูลส่วนตัวของแมว และคำนวณปริมาณอาหารที่ควรได้รับต่อวันใหม่ได้โดยกดปุ่มคำนวณปริมาณอาหาร อีกทั้งยังมีปุ่มแสดงกราฟแนวโน้มดัชนีมวลกาย น้ำหนัก และปริมาณอาหาร ดังรูปที่ 3.10



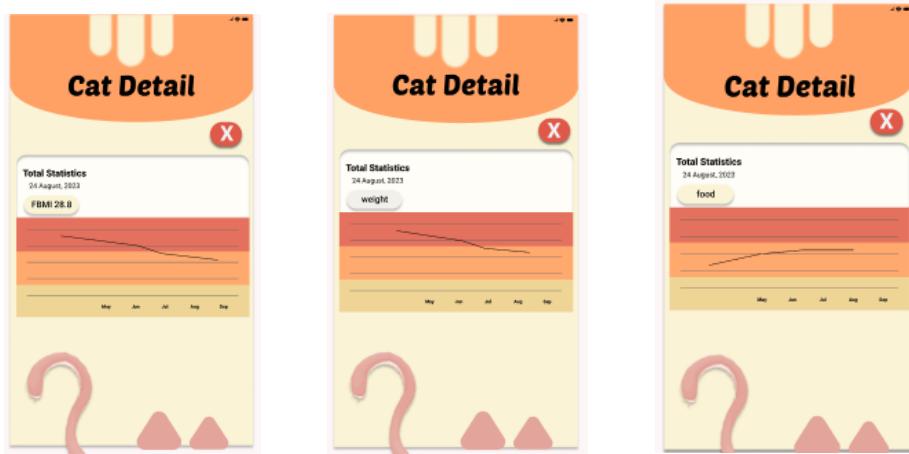
รูปที่ 3.10 หน้าแสดงและแก้ไขข้อมูลของแมว

หน้าคำนวณปริมาณอาหารที่ควรได้รับต่อวันได้ถูกออกแบบให้เลือกและใส่ข้อมูลทีละหน้าจากนั้นจะแสดงผลลัพธ์การคำนวณ ดังรูปที่ 3.11



รูปที่ 3.11 หน้าคำนวณปริมาณอาหารที่ควรได้รับต่อวัน

หน้าแสดงกราฟแนวโน้มดัชนีมวลกาย น้ำหนัก และปริมาณอาหารได้ถูกออกแบบให้แยกเป็น 3 หน้าเพื่อจ่ายต่อการเข้าใจ ดังรูปที่ 3.12



ก. หน้ากราฟแนวโน้ม
ดัชนีมวลกาย

ข. หน้ากราฟแนวโน้ม
น้ำหนัก

ค. หน้ากราฟแนวโน้ม
ปริมาณอาหาร

รูปที่ 3.12 หน้ากราฟน้ำหนัก ดัชนีมวลกาย และปริมาณอาหาร

3.3 การเขียนแอปพลิเคชัน

แอปพลิเคชันที่เขียนโดยภาษา dart ซึ่งสามารถเขียนได้ทั้ง Mobile App และอุปกรณ์ อิเล็กทรอนิกส์ เช่น ไมโครคอนโทรลเลอร์ ซึ่งประมวลผลได้รวดเร็ว (Walrath & Ladd, 2011)

3.3.1 หน้างลงทะเบียนบัญชีแมว

3.3.1.1 ปุ่มคำแนะนำ

การทำปุ่มคำแนะนำจะเป็นไอคอนชื่อ info ซึ่งปุ่มจะทำหน้าที่แสดงคำแนะนำในรูปแบบพ้อปอัป (pop up) โดยเขียนคำสั่งดังรูปที่ 3.13

```
@override
Widget build(BuildContext context) {
  return DefaultTabController(length: 2,
```

รูปที่ 3.13 คำสั่งสร้างปุ่มคำแนะนำ

```
        child: Scaffold(  
  
            appBar: AppBar(  
  
@override  
Widget build(BuildContext context) {  
  
    return DefaultTabController(  
  
        length: 2,  
  
        child: Scaffold(  
  
            appBar: AppBar(  
  
                backgroundColor: Colors.orange,  
  
                automaticallyImplyLeading: false,  
  
                actions: [  
  
                    IconButton(  
  
                        icon: Icon(Icons.info),  
  
                        onPressed: _showGreeting,  
  
                        iconSize: 35,  
  
                    ),  
  
                ],  
  
@override  
_HomeScreenState createState() => _HomeScreenState();  
}  
  
class _HomeScreenState extends State<HomeScreen> {  
  
    void _showGreeting() {  
  
        showDialog(  
  
            context: context,  
  
            builder: (BuildContext context) {
```

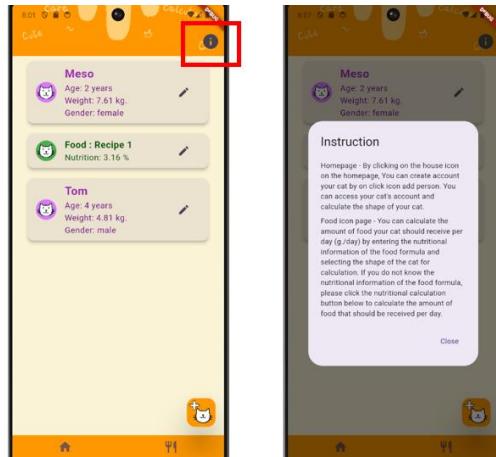
```

return AlertDialog(
    title: Text("Instruction"),
    contentPadding:
        EdgeInsets.fromLTRB(24, 20, 24, 20),
    content: Column(
        mainAxisAlignment: MainAxisAlignment.min,
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [Text(
            "Homepage - By clicking on the house icon on the
homepage, You can create account your cat by on click icon add person.
You can access your cat's account and calculate the shape of your
cat."), SizedBox(height: 8),
        Text(
            "Food icon page - You can calculate the amount of
food your cat should receive per day (g./day) by entering the
nutritional information of the food formula and selecting the shape of
the cat for calculation. If you do not know the nutritional information
of the food formula, please click the nutritional calculation button
below to calculate the amount of food that should be received per
day."), ],
        actions: <Widget>[
        TextButton(
            onPressed: () {
                Navigator.of(context).pop();
                child: Text("Close"),
            ), ],
        );
    )
)

```

รูปที่ 3.13 (ต่อ) คำสั่งสร้างปุ่มคำแนะนำ

ผู้ใช้งานกดปุ่มคำแนะนำนำที่มุมขวาบนของหน้าการใช้งานหลักจะมีหน้าต่างคำแนะนำพื้นอัปขึ้นมา จากนั้นหากต้องการจะปิดหน้าต่างให้กดที่ปุ่ม close เพื่อทำการปิดหน้าต่าง ดังรูปที่ 3.14



ก. ปุ่มคำแนะนำนำ ข. หน้าต่างคำแนะนำนำ

รูปที่ 3.14 ปุ่มคำแนะนำนำ

3.3.1.2 หน้าแสดงบัญชีแมวและสูตรอาหาร

หน้าการใช้งานนี้ต้องการแสดง 2 ข้อมูล คือ บัญชีแมวและสูตรอาหาร ซึ่งบัญชีแมวจะต้องแสดงข้อมูลเช่น อายุ น้ำหนัก และเพศ ส่วนสูตรอาหารจะแสดงข้อมูลสูตรอาหารและโภชนาการของผลิตภัณฑ์ โดยเขียนคำสั่งดังรูปที่ 3.15

```
body: Container(
  color: Color(0xFFFFAF3D6),
  child:
    TabBarView(
      children:
        [

```

รูปที่ 3.15 คำสั่งสร้างการ์ดบัญชีแมวและสูตรอาหาร

```
Consumer<TransactionProvider>(){

    builder: (context, provider, child) {

        var transactions = provider.transactions;

        var count = transactions.length;

        if (count == 0) {

            return Center(
                child: Text("Please create a cat account."),
            );
        } else {

            return ListView.builder(
                itemCount: count,
                itemBuilder: (context, int index) {
                    Transactions data = transactions[index];

                    return Card(
                        elevation: 5,
                        margin: const EdgeInsets.symmetric(
                            vertical: 10,
                            horizontal: 30,
                        ),
                        color: Color(0xFFFFAF3D6),
                        child: ListTile(
                            title: Column(
                                crossAxisAlignment:
                                    CrossAxisAlignment.start,
                            children: [

```

รูปที่ 3.15 (ต่อ) คำสั่งสร้างการ์ดบัญชีแมวและสูตรอาหาร

```
        if (data.title != null &&
data.title.isNotEmpty)

            Text(
                '${data.title.toString()}',
                style: TextStyle(
                    color: Colors.purple,
                    fontSize: 22,
                    fontWeight: FontWeight.bold,
                ),),
            if (data.age != null && data.age != 0)

                Text(
                    'Age: ${data.age.toString()} years',
                    style: TextStyle(
                        color: Color.fromARGB(255, 111,
9, 130),
                    ),,
                ),
            if (data.weight != null && data.weight
!= 0.0)

                Text(
                    'Weight: ${data.weight.toString()} kg.',
                    style: TextStyle(
                        color: Color.fromARGB(255, 111,
9, 130),),
                ),
        ),
    ),
}
```

รูปที่ 3.15 (ต่อ) คำสั่งสร้างการ์ดบัญชีแมวและสุตรอาหาร

```

        ) ,
        if (data.gender != null &&
            data.gender.isNotEmpty)
            Text(
                'Gender: ${data.gender}' ,
                style: TextStyle(
                    color: Color.fromARGB(255, 111,
9, 130),
                ) ,
            ) ,
        if (data.foodName != null &&
            data.foodName.isNotEmpty)
            Text(
                'Food : ${data.foodName}' ,
                style: TextStyle(
                    color: Color.fromARGB(255, 9, 88,
12),
                    fontSize: 18,
                    fontWeight: FontWeight.bold,
                ) ,
            ) ,
        if (data.nutritionValue != null &&
            data.nutritionValue != 0.0)
            Text(
                'Nutrition:
${data.nutritionValue.toString()} %',

```

รูปที่ 3.15 (ต่อ) คำสั่งสร้างการ์ดบัญชีเมวและสูตรอาหาร

```

        style: TextStyle(
            color: Color.fromARGB(255, 9, 88,
12),
        ) ,
        ) ,
    ] ,
),
leading: CircleAvatar(
backgroundColor: (data.title != null &&
data.age != null &&
data.age != 0 &&
data.weight != null &&
data.weight != 0.0)
? Color.fromARGB(255, 216, 122, 250)
: Colors.green,
child: Image.network(
'https://blogger.googleusercontent.com/
img/b/R29vZ2x1/AVvXsEiqGdgHQfHnx_4jqHgtwIHVgTHWeR4W-9-
AwgOuwDuaGGVZaMVmWmwCal9S8zYZPBF_vHkwVn7lD6jrvnF20tfpsYeYGNU_IMHBmC9R9d
CEgf3v9mDUO-YLd_u9PZAotF2biyM1SqKLesLLSrqmhGtqmYO2Vn7z2LxVMf-H5CNrkdP-
N75hcIwuKtLriaxK/w1684-h1069-p-k-no-nu/2350965C-B25F-4DEB-8FC3-
8F634D5E52CD.png',
width: 40,
height: 40,
),
),

```

รูปที่ 3.15 (ต่อ) คำสั่งสร้างการ์ดบัญชีแมวและสูตรอาหาร

การ์ดบัญชี หากเป็นบัญชีแมว การ์ดจะเป็นสีชมพู แต่ถ้าหากเป็นโภชนาการของผลิตภัณฑ์ การ์ดจะเป็นสีเขียว ดังรูปที่ 3.16



รูปที่ 3.16 การ์ดบัญชีแมวและสูตรอาหาร

3.3.1.3 ปุ่มลงทะเบียนแมว

การสร้างปุ่มให้กดเข้าไปสู่หน้าลงทะเบียนแมว เมื่อผู้ใช้งานกดปุ่มนี้จะถูกนำพาไปหน้าลงทะเบียนแมว โดยเขียนคำสั่งดังรูปที่ 3.17

```
floatingActionButton: FloatingActionButton(
    onPressed: () {
        Navigator.push(context, MaterialPageRoute(builder:
            (context) {
                return FormScreen();
            }
        ));
    },
    backgroundColor: Colors.orange,
    elevation: 50,
```

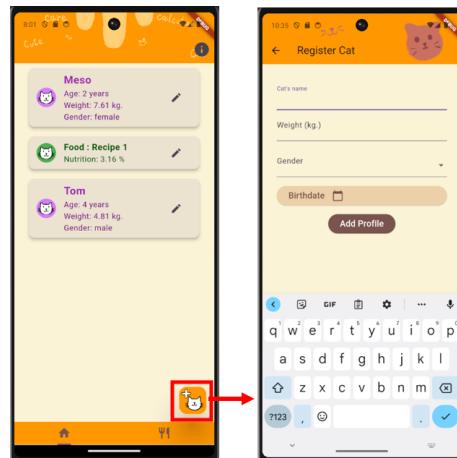
รูปที่ 3.17 คำสั่งสร้างปุ่มลงทะเบียนแมว

```

child:
  Image.network(
    'https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXs
Eh1YloZbvZi_bcS9V_AAEIHddrJdYRDa5tZlmXKbwJQadvAkwCeEbfNrKrd64C8_8qNXyD7
x_VElNYUPQBhyphenhyphen0nPsf8qyU8qR5mSyczhlnVrIC-
UZg9kxqdr_6znHyyh_S_ZuKx3tXnRsSF6v1qQ4FnAe3cjN0
1N39coevp8RVI3YqxpvjhGSv0qIu_HHTsL/s320/872B90B1-6954-4944-ACFE-
296A6F2F7A1D.png',
    width: 50,
    height: 50,
  ),
),
)
,
```

รูปที่ 3.17 (ต่อ) คำสั่งสร้างปุ่มลงทะเบียนแมว

โดยที่ปุ่มจะแสดงรูปแมวและเครื่องหมายบวกอยู่ด้วยกัน ดังรูปที่ 3.18



รูปที่ 3.18 ปุ่มเข้าสู่หน้าลงทะเบียนแมว

3.3.1.4 ปุ่มแสดงหน้าหลักและหน้าคำนวณปริมาณอาหารที่ควรได้รับต่อวัน

ปุ่มแสดงหน้าหลักและหน้าคำนวณปริมาณอาหารที่ควรได้รับต่อวัน ซึ่งเมื่อผู้ใช้งานกดปุ่ม แอปพลิเคชันจะพาไปยังหน้าต่างของปุ่มนั้น ๆ โดยเขียนคำสั่งดังรูปที่ 3.19

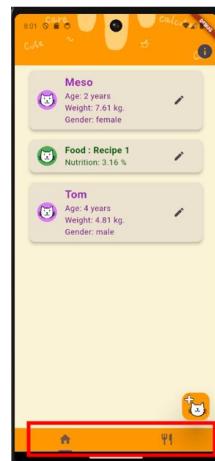
```

floatingActionButtonLocation: FloatingActionButtonLocation.endFloat,
bottomNavigationBar: PreferredSize(
  preferredSize: Size.fromHeight(90),
  child: Container(
    padding: EdgeInsets.only(bottom: 0),
    decoration: BoxDecoration(
      color: Colors.orange,
    ),
    child: TabBar(
      tabs: [
        Tab(
          icon: Icon(Icons.home, size: 28, color:
Colors.brown),
        ),
        Tab(
          icon: Icon(Icons.restaurant, size: 28, color:
Colors.brown),
        ),
      ],
    ),
  ),
)

```

รูปที่ 3.19 คำสั่งสร้างปุ่มแสดงหน้า Home และหน้า Food

ปุ่มแสดงหน้าหลักและหน้าคำนวนปริมาณอาหารที่ควรได้รับต่อวันนั้นจะแสดงอยู่ในแถบด้านล่างของแอปพลิเคชัน โดยปุ่มแสดงหน้าหลักจะเป็นสัญลักษณ์รูปบ้านและปุ่มหน้าคำนวนปริมาณอาหารที่ควรได้รับต่อวันจะเป็นสัญลักษณ์ส้อมและมีด ดังรูปที่ 3.20



รูปที่ 3.20 ปุ่มแสดงหน้าหลักและหน้าคำนวนปริมาณอาหารที่ควรได้รับต่อวัน

3.3.1.5 ช่องกรอกข้อมูล ชื่อแมว น้ำหนัก และเพศ

ช่องกรอกข้อมูล ชื่อและเพศ รับข้อมูลในรูปแบบข้อความ ส่วนข้อมูลน้ำหนัก เก็บข้อมูลในรูปแบบตัวเลขทศนิยม โดยเขียนคำสั่งดังรูปที่ 3.21

```
TextField(
    decoration: InputDecoration(
        labelText: "Cat's name"),
    autofocus: true, controller: titleController,
    validator: (String? str) {
        if (str == null || str.isEmpty) {
            return "Please enter your cat's name.";
        }
    }
)
```

รูปที่ 3.21 คำสั่งสร้างช่องรับข้อมูลชื่อแมว น้ำหนัก และเพศ

```
        } return null; }, ),  
  
        TextFormField(  
  
            decoration: InputDecoration(labelText: "Weight (kg.)"),  
            controller: weightController,  
            keyboardType: TextInputType.number,  
            validator: (String? str) {  
  
                if (str == null || str.isEmpty) {  
  
                    return "Please enter your cat's weight."; }  
  
                return null; }, ),  
  
        SizedBox(height: 10),  
  
        DropdownButtonFormField<String>(  
  
            decoration: InputDecoration(labelText: 'Gender'),  
            value: selectedGender,  
            items: ['male', 'female', 'other']  
  
.map((gender) => DropdownMenuItem(  
  
                child: Text(gender),  
                value: gender, ))  
  
.toList(),  
  
onChanged: (String? value) {  
  
    setState(() {  
  
        selectedGender = value; });},  
  
validator: (value) {  
  
    if (value == null || value.isEmpty) {  
  
        return 'Please select your cat\'s gender.'; }  
  
    return null; }, ),
```

รูปที่ 3.21 (ต่อ) คำสั่งสร้างช่องรับข้อมูลชื่อแมว น้ำหนัก และเพศ

ช่องกรอกข้อมูลนั้น หากผู้ใช้งานไม่ดำเนินการกรอกข้อมูลในช่องได้ก็ตาม และปุลิเคชันจะแสดงข้อความเตือนให้ช่องใส่ข้อมูลที่ผู้ใช้งานไม่ได้ใส่ข้อมูลนั้น ๆ ดังรูปที่ 3.22



รูปที่ 3.22 ช่องกรอกข้อมูลซึ่งเม瓦 นำหนัก และเพศ

3.3.1.6 ปฏิทินเลือกวันเดือนปีเกิด

ปฏิทินวันเดือนปีเกิดของเม瓦 โดยเขียนคำสั่งดังรูปที่ 3.23

```
SizedBox(height: 15),  
ElevatedButton(  
    style: ButtonStyle(  
        backgroundColor: MaterialStateProperty.all(  
            Color.fromARGB(255, 244, 215, 171)),  
    ),  
    onPressed: () {  
        showDatePicker(  
    },
```

รูปที่ 3.23 คำสั่งสร้างปฏิทินวันเดือนปีเกิด

```
context: context,
        initialDate: DateTime.now(),
        firstDate: DateTime(2000),
        lastDate: DateTime.now(),
        builder: (BuildContext context, Widget? child) {
            return Theme(
                data: ThemeData.light().copyWith(
                    colorScheme: ColorScheme.light(
                        primary: Colors.orange,
                    ),
                ),
            ),
            child: child!,
        );
    },
).then((pickedDate) {
    if (pickedDate != null) {
        setState(() {
            selectedDate = pickedDate;
        });
    }
});
},
child: Row(
    children: [
        Text(

```

```

selectedDate == null

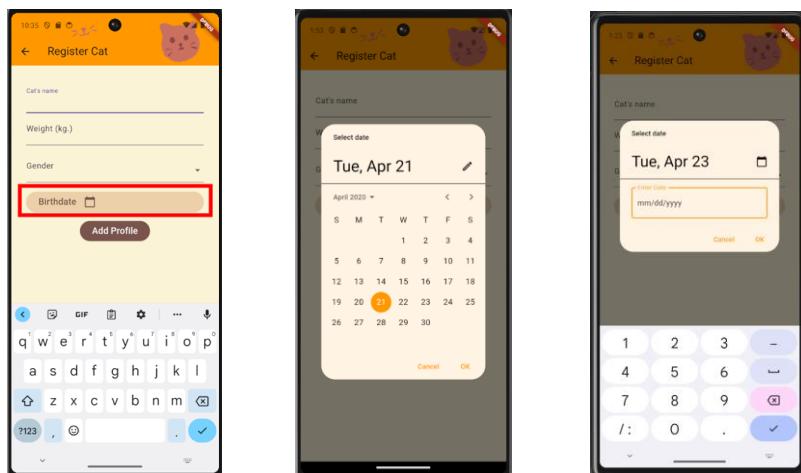
        ? 'Birthdate': 'Birthdate:
${selectedDate.toString().substring(0, 10)}',

        style: TextStyle(
            fontSize: 18,
            color: Colors.brown),),
SizedBox(width: 15),
Icon(Icons.calendar_today,
color: Colors.brown),],),),

```

รูปที่ 3.23 (ต่อ) คำสั่งสร้างปฏิทินวันเดือนปีเกิด

ปฏิทินวันเดือนปีเกิด สามารถเลือกวันเดือนปีเกิดจากตารางปฏิทิน หรือกรอกวันเดือนปีเกิดได้ ดังรูปที่ 3.24



ก. หน้าปุ่มปฏิทิน

วันเดือนปีเกิด

ข. หน้าตารางปฏิทิน

วันเดือนปีเกิด

ค. หน้ากรอกปฏิทิน

วันเดือนปีเกิด

รูปที่ 3.24 หน้าปฏิทินวันเดือนปีเกิด

3.3.1.7 ปุ่มบันทึกข้อมูล

ปุ่มบันทึกข้อมูลจะทำการบันทึกชื่อ น้ำหนัก เพศ และอายุของแมว ซึ่งอายุแสดงผลนั้นจะได้จากการเอาปีปัจจุบันมาลบด้วยปีเกิดของแมว ซึ่งวิธีการนี้จะเก็บค่าอายุได้เป็นหน่วยปี ซึ่งข้อมูลทั้งหมดจะถูกจัดเป็นรูปแบบของ Transactions statement ก่อนส่งไปที่ฐานข้อมูลต่อไป ดังรูปที่ 3.25

```
SizedBox(
    height: 10),
ElevatedButton(
    onPressed: () {
        if (formKey.currentState?.validate() ?? false) {
            var title = titleController.text;
            var weight = double.parse(weightController.text);
            var age = DateTime.now().year -
                selectedDate!.year;
            Transactions statement = Transactions(
                title: title,
                weight: weight,
                age: age,
                gender: selectedGender!,
            );
            var provider =
Provider.of<TransactionProvider>(context,
                listen: false);
            provider.addTransaction(statement);
            Navigator.pop(context);
        }
    }
);
```

รูปที่ 3.25 คำสั่งสร้างปุ่มบันทึกข้อมูล

```

} } ,
style: ButtonStyle(
backgroundColor: MaterialStateProperty.all(Color(
0xFF795548)) ,
),
child: Text(
"Add Profile",
style: TextStyle(
color: Colors.white,
fontSize: 18,
),),),),

```

รูปที่ 3.25 (ต่อ) คำสั่งสร้างปุ่มบันทึกข้อมูล

ปุ่มบันทึกข้อมูล จะมีตัวอักษรสีครีมและพื้นหลังสีน้ำตาล ดังรูปที่ 3.26



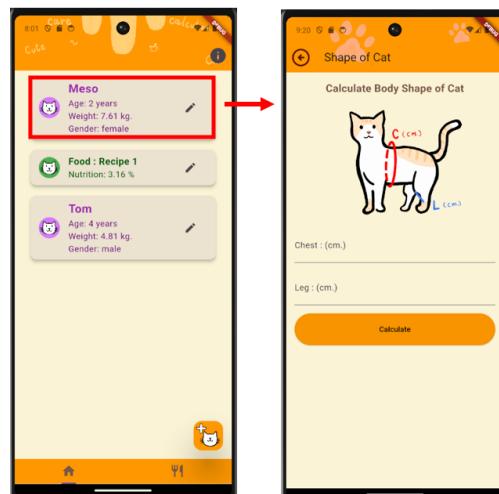
รูปที่ 3.26 ปุ่มบันทึกข้อมูล

3.3.1.8 การเข้าสู่หน้าคำนวณดัชนีมวลกายผ่านการ์ดบัญชีแมว

การเข้าสู่หน้าคำนวณดัชนีมวลกายผ่านการ์ดบัญชีแมว เมื่อผู้ใช้งานกดที่การ์ดบัญชีแมวที่เป็นสีชมพู และคลิกเข้าไปจะเข้าสู่หน้าคำนวณดัชนีมวลกาย โดยเปลี่ยนคำสั่งดังรูปที่ 3.27

```
onTap: () {
    Navigator.push(context,
        MaterialPageRoute(builder: (context) {
            return ProfileCard();
        })
    );
},
```

รูปที่ 3.27 คำสั่งการกดเข้าสู่หน้าคำนวณดัชนีมวลกายผ่านการ์ดบัญชีแมว



รูปที่ 3.28 การเข้าสู่หน้าคำนวณดัชนีมวลกายผ่านการ์ดบัญชีแมว

3.3.1.9 ปุ่มแก้ไขข้อมูลของแมว

ปุ่มการแก้ไขข้อมูลของแมวเมื่อผู้ใช้งานกดปุ่มแอปพลิเคชันจะพาผู้ใช้งานไปยังหน้าข้อมูลของแมวเพื่อทำการแก้ไข ข้อมูลชื่อ อายุ น้ำหนัก และเพศ โดยเขียนคำสั่งปุ่มแก้ไขข้อมูลแมวดังรูป 3.29

```
trailing: IconButton(
    icon: Icon(Icons.edit),
    onPressed: () async {
        final updatedData = await Navigator.push(
            context,
            MaterialPageRoute(
                builder: (context) =>
                    EditTransactionScreen(data: data),));
        if (updatedData != null) {
            provider.updateTransaction(updatedData); }},
```

รูปที่ 3.29 คำสั่งสร้างปุ่มแก้ไขข้อมูลแมว

ข้อมูลชื่อและเพศจะถูกเก็บในรูปแบบข้อความ ข้อมูลอายุจะถูกเก็บเป็นเลขจำนวนเต็ม และข้อมูลน้ำหนักจะถูกเก็บเป็นเลขทศนิยม โดยเขียนคำสั่งดังรูปที่ 3.30

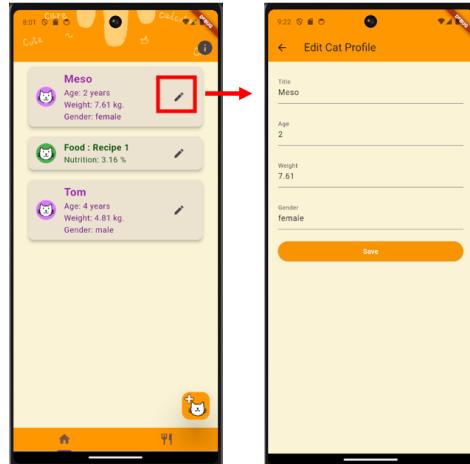
```
void initState() {super.initState();
    _titleController = TextEditingController(text:
    widget.data.title);
    _ageController = TextEditingController(text:
    widget.data.age.toString());
    _weightController = TextEditingController(text:
```

รูปที่ 3.30 คำสั่งแก้ไขข้อมูลแมว

```
widget.data.weight.toString());  
  
widget.data.nutritionValue.toString());  
  
_genderController = TextEditingController(text:  
widget.data.gender);  
  
}  
  
@override  
  
Widget build(BuildContext context) {  
  
    return Scaffold(  
  
        appBar: AppBar(  
  
            title: Text('Edit Cat Profile'),  
  
            backgroundColor: Colors.orange,  
  
) ,  
  
        body: Container(  
  
            color: Color(0xFFFFAF3D6),  
  
            padding: EdgeInsets.all(20.0),  
  
            child: Column(  
  
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
  
                children: [  
  
                    TextFormField(  
  
                        controller: _titleController,  
  
                        decoration: InputDecoration(labelText: 'Title'), ),  
  
                    SizedBox(height: 20.0),  
  
                    TextFormField(  
  
                        controller: _ageController,  
  
                        decoration: InputDecoration(labelText: 'Age'),  
                ]  
            )  
        )  
    );  
}
```

```
keyboardType: TextInputType.number,) ,  
SizedBox(height: 20.0) ,  
TextFormField(  
    controller: _weightController,  
    decoration: InputDecoration(labelText: 'Weight') ,  
    keyboardType: TextInputType.number, ) ,  
SizedBox(height: 20.0) ,  
TextFormField(  
    controller: _genderController,  
    decoration: InputDecoration(labelText: 'Gender') , ,  
SizedBox(height: 20.0) ,  
ElevatedButton(  
    onPressed: () async {  
        var updatedData = Transactions(  
            id: widget.data.id,  
            title: _titleController.text,  
            age: int.parse(_ageController.text) ,  
            weight: double.parse(_weightController.text) ,  
            gender: _genderController.text,) ;
```

รูปที่ 3.30 (ต่อ) คำสั่งแก้ไขข้อมูลเมื่อ



รูปที่ 3.31 ปุ่มแก้ไขข้อมูลแมว

3.3.2 หน้าคำนวนดัชนีมวลกาย

3.3.2.1 การแสดงรูปแบบหน้าจอการวัดเพื่อคำนวนดัชนีมวลกายและช่องกรอกข้อมูลการวัด

การแสดงรูปภาพแบบหน้าจอการวัดเพื่อคำนวนดัชนีมวลกาย ซึ่งมีช่องกรอกข้อมูลความยาวรอบอก (ซม.) และความยาวของขาส่วนหลังตั้งแต่หัวเข้าถึงข้อเท้า (ซม.) ซึ่งอยู่ภายใต้รูปภาพแบบหน้าจอการวัดเพื่อคำนวนดัชนีมวลกาย โดยเขียนคำสั่งดังรูปที่ 3.32

```
class _ProfileCardState extends State<ProfileCard> {
  TextEditingController cController = TextEditingController();
  TextEditingController lController = TextEditingController();
  String bodyShapeResult = '';
  String foodPerDayResult = '';
  String imageUrl = '';
  double nutrition = 0.0;
  double weight = 0.0;
```

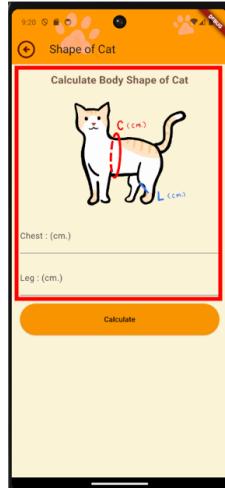
รูปที่ 3.32 คำสั่งแสดงรูปตำแหน่งการวัดเพื่อคำนวนดัชนีมวลกายและช่องกรอกข้อมูลการวัด

```

body: SingleChildScrollView(
    child: Container(
        padding: EdgeInsets.all(16.0),
        child: Column(
            mainAxisAlignment: MainAxisAlignment.spaceEvenly,
            children: [
                Text(
                    'Calculate Body Shape of Cat',
                    style: TextStyle(
                        fontSize: 20,
                        fontWeight: FontWeight.bold,
                        color: Colors.brown,
                    ),
                    textAlign: TextAlign.center,
                ),
                Image.network(
                    'https://blogger.googleusercontent.com/img/b/R29vZ2xl
/AVvXsEg8T-0piFs8_mOXVPwr0_7rmfS-1rknE3XBDq8yNvmNlKI7UHgu2kUJ-7v-
w5lSq2LssAi6RnABGhcmSrFa78TCxAxaHpQ3RBHcNTqDj0Jz5fMACrhnm8sczEWEjQ3fR7z
_7g4_Zf4AQWu-vHfbUjwn8-7tWd1vhfJXs-Dm-
minh6knwhdSEwKmad8iRcr7/s3508/8556CEC9-4735-40F5-9355-
0CDDF02D2CD2.gif',
                    height: 250, width: 500,
                ),
                buildTextField(controller: cController, label: 'Chest : (cm.)'),
                SizedBox(height: 16),
                buildTextField(controller: lController, label: 'Leg : (cm.)'), SizedBox(height: 16),
            ],
        ),
    ),
);

```

รูปที่ 3.32 (ต่อ) คำสั่งแสดงรูปตำแหน่งการวัดเพื่อคำนวนดัชนีมวลกายและช่องกรอกข้อมูลการวัด



รูปที่ 3.33 การแสดงรูปแบบหน้าจอการวัดเพื่อคำนวณตัวนิ่มมวลกายและช่องกรอกข้อมูลการวัด

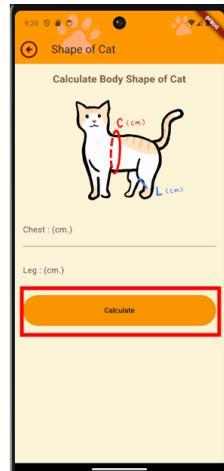
3.3.2.2 ปุ่มคำนวณตัวนิ่มมวลกาย

ปุ่มคำนวณตัวนิ่มมวลกาย เขียนคำสั่งได้ดังรูปที่ 3.34

```
ElevatedButton(onPressed: () {calculate();},  
style: ButtonStyle(  
backgroundColor: MaterialStateProperty.all(Colors.orange),  
foregroundColor: MaterialStateProperty.all(Colors.black),  
padding:  
MaterialStateProperty.all(EdgeInsets.symmetric(horizontal: 100,  
vertical: 20)),),  
child: Text('Calculate'),),
```

รูปที่ 3.34 คำสั่งสร้างปุ่มคำนวณตัวนิ่มมวลกาย

ปุ่มคำนวณตัวนิ่มมวลกายนั้นจะ มีลักษณะเป็นรูปทรงสี่เหลี่ยมผืนผ้าขอบมน ซึ่งแนวอนมีขนาด 100 พิกเซลและแนวตั้ง 20 พิกเซล สีส้ม และมีตัวอักษรสีดำ ดังรูปที่ 3.35



รูปที่ 3.35 ปุ่มคำนวณด้ชนีมวลกาย

3.3.2.3 สูตรการคำนวณด้ชนีมวลกาย

การคำนวณด้ชนีมวลกายนั้น จะนำข้อมูลที่ผู้ใช้งานกรอกในช่องกรอกข้อมูลมาคำนวณ โดยเขียนคำสั่งดังรูปที่ 3.36

```
Widget buildTextField({
    required TextEditingController controller,
    required String label, }) {
    return TextField(
        controller: controller,
        keyboardType: TextInputType.number,
        decoration: InputDecoration
        (labelText: label),
    );
}

void calculate() {
    try {
```

รูปที่ 3.36 คำสั่งคำนวณด้ชนีมวลกาย

```

double c = double.parseDouble(cController.text);

double l = double.parseDouble(lController.text);

double calculatedResult = ((c / 0.7062) - 1) / 0.9156 - 1;

setState(() {

    bodyShapeResult = calculatedResult.toString();

    setProfileImage(calculatedResult);

} catch (e) {

    setState(() {

        imageUrl = '';
    });
}}
```

รูปที่ 3.36 (ต่อ) คำสั่งคำนวณดัชนีมวลกาย

3.3.2.4 การแสดงผลลัพธ์การคำนวณดัชนีมวลกาย

ผลลัพธ์จากการคำนวณจะแสดงรูปภาพตามเงื่อนไข โดยผลลัพธ์จะแสดงออกมา 3 ผลลัพธ์ ด้วยกัน โดยสามารถเขียนคำสั่งได้ดังรูปที่ 3.37

```

void setProfileImage(double calculatedResult) {

    if (calculatedResult < 15) { imageUrl =

        'https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEh
jOrp_BLVIhLMH9TPWEAqqUagJjPDAFeDyE6Q7jad-
ONzgvlgHXb01tXiwoj_d4sQugeNzyxirDlRzA_OIgKrY_IpjeNPB-
i18cUgaZuJzar_oEdqsXMYbga23u_KNZmj3R4sdRDbG2UtzT9dlUi6z1UvFX4ussGe1ZCx
MLZkb1-JWaztta2N5QPAnB/s320/2DEE3F7F-E831-4A5B-8206-
AF123CCEADF4.png'; } else if (calculatedResult >= 15 && calculatedResult
< 29.9) { imageUrl =

        'https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEg
CL0gROofHUJjM-H4eh-
```

รูปที่ 3.37 คำสั่งแสดงผลลัพธ์ดัชนีมวลกาย

```

8JD4rBy0SwXE8S5bMfCYuZm0Tfu9f25QhzixSdpDhJa_o2gmabwqIdVTNi9-
N6zHAnd7R62Kpo6HUqqqtYF0ujZMA6yYzjnz3i8qo2XBVNe3CgvWVsfs23uEvCdMqcDvB5aoZ
hu8RIYUwmqI7KYINwuepuOsZCj-jD3xsuYVmYp1/s320/14176E22-0B03-449A-A434-
030A8DA7A060.png';

} else if (calculatedResult >= 30) { imageUrl =

    'https://blogger.googleusercontent.com/img/b/R29vZ2x1/AVvXsEh
VLbs19049imHbvXUt1XfNeLNv9hHB_QMyK1d51ATxbvBmkh-
dwqXZivY1qe9tUFsQhs1TGo7j2G09w7P6ptCL1fHIemqkenCNozPMsOzx76YIyd6V7NIBC-
v_r9HGNWmXz1VcBhJn8VN3GGVHsazqketkga0PI50_9RUSHbZYqoGsEg96UiQmT1YUGP9X/
s320/2786AE1D-0A3F-4F6D-A8DB-FDCCCB7C47B7.png';

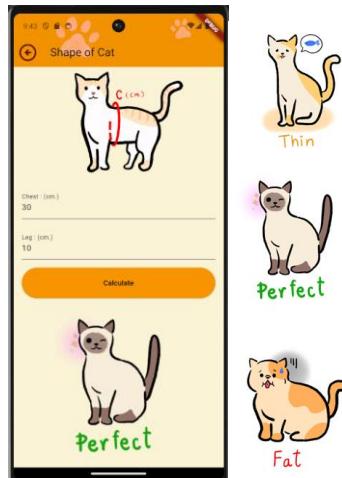
} else {

    imageUrl = '';
}

```

รูปที่ 3.37 (ต่อ) คำสั่งแสดงผลลัพธ์ดัชนีมวลกาย

ผลลัพธ์ดัชนีมวลการที่แสดงจะมี 3 ผลลัพธ์ด้วยกัน คือ ผลลัพธ์ดัชนีมวลกายเกินเกณฑ์มาตรฐาน ผลลัพธ์ดัชนีมวลกายอยู่ในเกณฑ์มาตรฐาน และผลลัพธ์ดัชนีมวลกายต่ำกว่าเกณฑ์มาตรฐาน ดังรูปที่ 3.38



รูปที่ 3.38 การแสดงผลลัพธ์ดัชนีมวลกาย

3.3.3 หน้าคำนวณปริมาณอาหารที่ควรได้รับต่อวัน

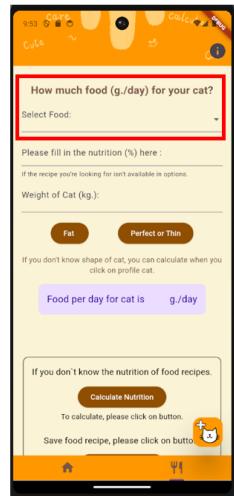
3.3.3.1 ช่องแสดงตัวเลือกสูตรอาหาร

ช่องเลือกสูตรอาหารเพื่อนำมาใช้ในการคำนวณ โดยเขียนคำสั่งดังรูปที่ 3.39

```
@override
Widget build(BuildContext context) {
  return SingleChildScrollView(
    child: Container(
      padding: EdgeInsets.all(16),
      color: Color(0xFFFAF3D6),
      child: Column(
        children: [
          SizedBox(height: 16),
          Text('How much food (g./day) for your cat?',
            style: TextStyle(fontSize: 20,
              fontWeight: FontWeight.bold,
              color: Colors.brown,),
            textAlign: TextAlign.center,),

          DropdownButtonFormField<Map<String, dynamic>>(
            value: selectedFood,
            onChanged: (value) {
              if (customNutritionController.text.trim().isEmpty) {
                setState(() {
                  selectedFood = value!;
                });
              }
            }
          )
        ],
      ),
    ),
  );
}
```

รูปที่ 3.39 คำสั่งสร้างช่องแสดงตัวเลือกสูตรอาหาร



รูปที่ 3.40 ช่องแสดงตัวเลือกสูตรอาหาร

3.3.3.2 ข้อมูลตัวเลือกสูตรอาหาร

ตัวเลือกสูตรอาหารจะมีชื่อสูตรอาหารและโภชนาการของผลิตภัณฑ์ และจะแสดงเห็นเป็นชื่อสูตรอาหารเท่านั้น โดยเขียนคำสั่งดังรูปที่ 3.41

```
String foodPerDayResult = '';
Map<String, dynamic>? selectedFood;
List<Map<String, dynamic>> foodList = [
    {"name": "PURINA ONE : 1+ years", "nutritionValue": 3.61},
    {"name": "Silver : Indoor", "nutritionValue": 3.0925},
    {"name": "Cat's taste : in gravy", "nutritionValue": 0.45},
    {"name": "Cat's taste : Senior 7+", "nutritionValue": 0.39}, ];
items: foodList.map(
    (food) {
        return DropdownMenuItem<Map<String, dynamic>>(

```

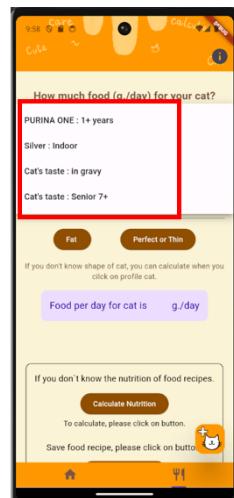
รูปที่ 3.41 คำสั่งสร้างตัวเลือกสูตรอาหาร

```

value: food,
    child: Text(food['name'] as String),);
) .toList(),
decoration: InputDecoration(labelText: 'Select Food:',
),
),
)
,
```

รูปที่ 3.41 (ต่อ) คำสั่งสร้างตัวเลือกสูตรอาหาร

คำสั่งตัวเลือกสูตรอาหารนี้ เมื่อกดปุ่มเลือกสูตรอาหาร ตัวเลือกจะแสดงออก ดังรูปที่ 3.42



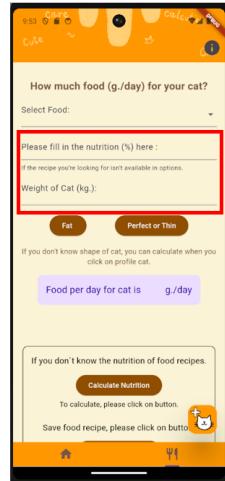
รูปที่ 3.42 การแสดงตัวเลือกสูตรอาหาร

3.3.3.3 ช่องกรอกค่าโภชนาการของผลิตภัณฑ์และช่องกรอกค่าน้ำหนัก

ช่องกรอกข้อมูลโภชนาการด้วยตัวเองหากตัวเลือกสูตรอาหารไม่มีสูตรอาหารที่ผู้ใช้งานต้องการ และช่องกรอกข้อมูลน้ำหนัก โดยเขียนคำสั่งดังรูปที่ 3.43

```
SizedBox(height: 16),  
    TextField(  
        controller: customNutritionController,  
        keyboardType: TextInputType.number,  
        enabled: selectedFood ==  
            null,  
        decoration: InputDecoration(  
            labelText: "Please fill in the nutrition (%) here :",  
            helperText:  
                "If the recipe you're looking for isn't available  
in options.",  
            contentPadding:  
                EdgeInsets.only(bottom: 8),  
        ),  
    ),  
    buildTextField(  
        controller: weightController,  
        label: 'Weight of Cat (kg.):',  
    ),
```

รูปที่ 3.43 คำสั่งสร้างช่องกรอกค่าโภชนาการของผลิตภัณฑ์และช่องกรอกค่าน้ำหนัก



รูปที่ 3.44 ช่องกรอกค่าโภชนาการของผลิตภัณฑ์และช่องกรอกค่าน้ำหนัก

3.3.3.4 ปุ่มเลือกดัชนีมวลกายในการคำนวณปริมาณอาหารที่ควรได้รับต่อวัน

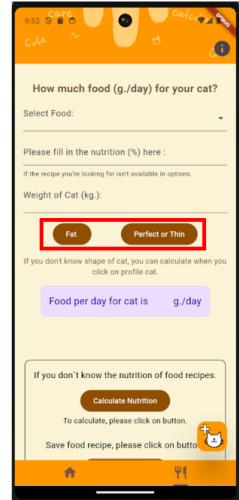
ปุ่มเลือกดัชนีมวลกายจะมีทั้งหมด 2 ปุ่ม คือ ปุ่มดัชนีมวลกายอยู่ในเกณฑ์มาตรฐานและปุ่มดัชนีมวลกายอยู่ในเกณฑ์มาตรฐานหรือต่ำกว่าเกณฑ์มาตรฐานซึ่ง 2 ปุ่มนี้จะใช้สูตรคำนวณแตกต่างกันโดยเยี่ยนคำสั่งดังรูปที่ 3.45

```
SizedBox(height: 16),  
Row(mainAxisAlignment:  
    MainAxisAlignment.spaceEvenly,  
    children: [  
        ElevatedButton(  
            onPressed: () {  
                calculateFoodPerDay(isFat: true);  
            },  
            style: ElevatedButton.styleFrom().copyWith(  
                backgroundColor: MaterialStateProperty.all(
```

รูปที่ 3.45 คำสั่งสร้างปุ่มเลือกดัชนีมวลกาย

```
Color.fromARGB(255, 147, 79, 2)),  
),  
child: Text(  
'Fat ',  
style: TextStyle(  
color: Color.fromARGB(255, 252, 252, 252),  
),  
),  
),  
ElevatedButton(  
onPressed: () {  
calculateFoodPerDay(isFat: false);  
},  
style: ElevatedButton.styleFrom().copyWith(  
backgroundColor: MaterialStateProperty.all(  
Color.fromARGB(255, 147, 79, 2)),  
),  
child: Text(  
'Perfect or Thin',  
style: TextStyle(  
color: const Color.fromARGB(255, 254, 254, 254),  
),),  
),],),
```

รูปที่ 3.45 (ต่อ) คำสั่งสร้างปุ่มเลือกดัชนีมวลกาย



รูปที่ 3.46 ปุ่มเลือกดัชนีมวลกาย

3.3.3.5 การแสดงผลลัพธ์การคำนวณปริมาณอาหารที่ควรได้รับต่อวัน

การแสดงผลลัพธ์การคำนวณปริมาณอาหารที่ควรได้รับต่อวัน โดยเขียนคำสั่งดังรูปที่ 3.47

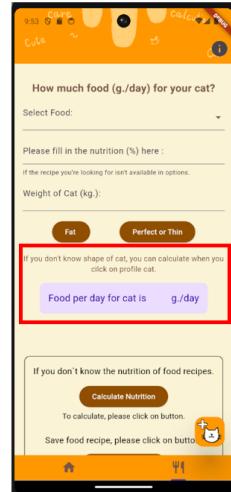
```
SizedBox(height: 16),
Text(
  "If you don't know shape of cat, you can calculate when
you cilck on profile cat.",
  style: TextStyle(
    fontSize: 14,
    color: Colors.brown,
  ),
  textAlign: TextAlign.center,
),
SizedBox(height: 16),
```

รูปที่ 3.47 คำสั่งสร้างผลลัพธ์การคำนวณปริมาณอาหารที่ควรได้รับต่อวัน

```

Container(
    decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(10),
        color: Color.fromARGB(255, 235, 221, 255),
    ),
    padding: EdgeInsets.all(16),
    child: Text(
        'Food per day for cat  
is $foodPerDayResult g./day',
        style: TextStyle(
            color: Color.fromARGB(255, 41, 2, 159), fontSize: 18),
    ),
),
)
,
```

รูปที่ 3.47 (ต่อ) คำสั่งสร้างผลลัพธ์การคำนวณปริมาณอาหารที่ควรได้รับต่อวัน



รูปที่ 3.48 การแสดงข้อความและผลลัพธ์การคำนวณปริมาณอาหารที่ควรได้รับต่อวัน

3.3.3.6 การเลือกสูตรในการคำนวณ

ปุ่มด้านล่างเกินเกณฑ์มาตรฐานและปุ่มด้านบนอยู่ในเกณฑ์มาตรฐานหรือต่ำกว่าเกณฑ์มาตรฐานซึ่งใช้สูตรต่างกันตามเงื่อนไข ดังรูปที่ 3.49

```
Widget buildTextField({
    required TextEditingController controller,
    required String label,
}) {
    return TextField(
        controller: controller,
        keyboardType: TextInputType.number,
        decoration: InputDecoration(labelText: label),
    );
}

void calculateFoodPerDay({required bool isFat}) {
    try {
        double weight = double.parse(weightController.text);
        double nutritionValue = 0.0;
        if (selectedFood != null && selectedFood!['nutritionValue'] != null) {
            nutritionValue = selectedFood!['nutritionValue'] as double;
        } else {
            nutritionValue = double.parse(customNutritionController.text);
        }
        double factor = isFat ? 1.2 : 1.4;
    }
}
```

รูปที่ 3.49 คำสั่งการคำนวณสูตรอาหาร

```

    double foodPerDay = factor * (70 * pow(weight, 0.75)) /
nutritionValue;

    setState(() {
        foodPerDayResult = foodPerDay.toStringAsFixed(2);
    });
} catch (e) {
    setState(() {
        foodPerDayResult = 'Invalid input';
    });
}
}

```

รูปที่ 3.49 (ต่อ) คำสั่งการคำนวนสูตรอาหาร

3.3.3.7 การป้องกันการเลือกสูตรอาหารพร้อมกัน 2 รูปแบบ

การแสดงข้อความเตือนเมื่อผู้ใช้งานกรอกข้อมูลในช่องโภชนาการและเลือกสูตรอาหารพร้อมกัน โดยเขียนคำสั่งดังรูปที่ 3.50

```

DropdownButtonFormField<Map<String, dynamic>>(
    value: selectedFood,
    onChanged: (value) {
        if (customNutritionController.text.trim().isEmpty) {
            setState(() {
                selectedFood = value!;
            });
        }
    }
)

```

รูปที่ 3.50 คำสั่งสร้างข้อความเตือนหากผู้ใช้งานเลือกสูตรอาหารเกิน 1 รูปแบบ

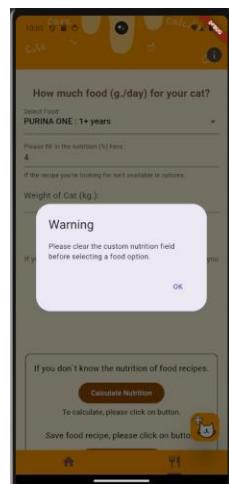
```

} else {

    showDialog(
        context: context,
        builder: (context) => AlertDialog(
            title: Text('Warning'),
            content: Text('Please clear the custom nutrition field before selecting a food option.'),
            actions: [
                TextButton(
                    onPressed: () {
                        Navigator.of(context).pop();
                    },
                    child: Text('OK'),
                ),
            ],
        ),
    );
}

```

รูปที่ 3.50 (ต่อ) คำสั่งสร้างข้อความเตือนหากผู้ใช้งานเลือกสูตรอาหารเกิน 1 รูปแบบ



รูปที่ 3.51 ข้อความเตือนหากผู้ใช้งานเลือกสูตรอาหารเกิน 1 รูปแบบ

3.3.3.8 ปุ่มเข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์

การแสดงปุ่มสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์ ซึ่งแอปพลิเคชันจะนำพาไปที่หน้าเลือกสูตรคำนวณโภชนาการของผลิตภัณฑ์ โดยเขียนคำสั่งดังรูปที่ 3.52

```
SizedBox(height: 80),
    Container(
        decoration: BoxDecoration(
            border: Border.all(color: Colors.black),
            borderRadius: BorderRadius.circular(10),),
        padding: EdgeInsets.all(16),
        child: Column(
            children: [
                Text(
                    'If you don`t know the nutrition of food recipes.',
                    style: TextStyle(
                        fontSize: 16,
                        color: Colors.black,),),
                SizedBox(height: 12),
                ElevatedButton(
                    onPressed: () {
                        Navigator.push(
                            context,
                            MaterialPageRoute(builder: (context) =>
CalFood()),);
                },),
```

รูปที่ 3.52 คำสั่งสร้างปุ่มเข้าสู่หน้าคำนวณโภชนาการในการสูตรอาหาร

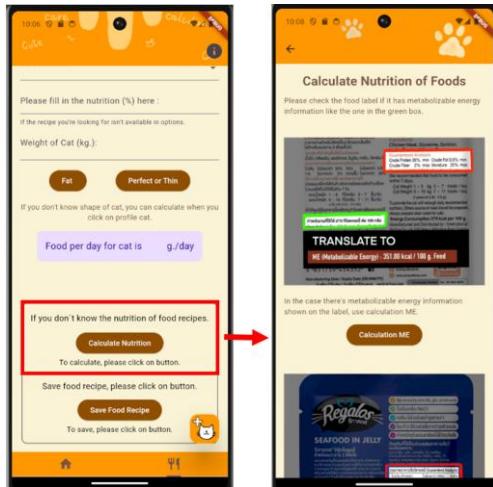
```

child: Text(
    'Calculate Nutrition',
    style: TextStyle(
        color: const Color.fromARGB(255, 255, 255,
255),
    ),
),
style: ElevatedButton.styleFrom().copyWith(
    backgroundColor: MaterialStateProperty.all(
        Color.fromARGB(255, 147, 79, 2)),
),
),
SizedBox(height: 3),
Text(
    'To calculate, please click on button.',
    style: TextStyle(
        fontSize: 14,
        color: Colors.black,
),
),

```

รูปที่ 3.52 (ต่อ) คำสั่งสร้างปุ่มเข้าสู่หน้าคำนวณโภชนาการในการสูตรอาหาร

ปุ่มเข้าสู่หน้าการเลือกสูตรคำนวนจะมีข้อความอธิบายว่า ปุ่มดังกล่าวใช้งานภายใต้เงื่อนไขการเลือกคำนวนโภชนาการผลิตภัณฑ์ ว่าสามารถใช้สูตรไหนในการคำนวนได้ ดังรูปที่ 3.53



รูปที่ 3.53 เข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์

3.3.3.9 ปุ่มเข้าสู่หน้าบันทึกสูตรอาหาร

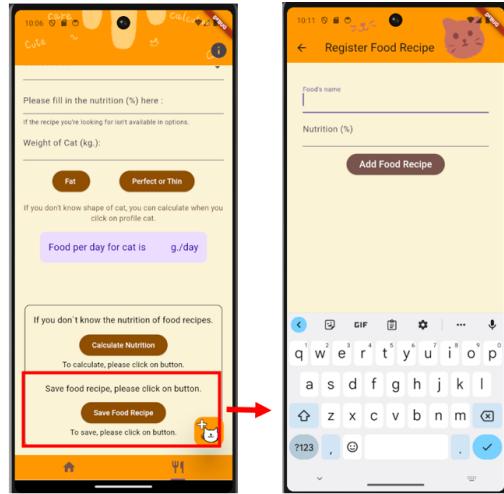
การแสดงปุ่มเข้าสู่หน้าบันทึกสูตรอาหาร ซึ่งแอปพลิเคชันจะนำพาไปที่หน้าบันทึกสูตรอาหาร โดยเขียนคำสั่งดังรูปที่ 3.54

```
SizedBox(height: 24),  
Text(  
  'Save food recipe, please click on button.',  
  style: TextStyle(  
    fontSize: 16,  
    color: Colors.black,  
  ),),  
SizedBox(height: 10),  
ElevatedButton(
```

รูปที่ 3.54 คำสั่งสร้างปุ่มเข้าสู่หน้าบันทึกสูตรอาหาร

```
onPressed: () {  
    Navigator.push(  
        context,  
        MaterialPageRoute(builder: (context) =>  
FormFood(),  
    );  
},  
child: Text(  
    'Save Food Recipe',  
    style: TextStyle(  
        color: Colors.white,  
    ),  
),  
style: ElevatedButton.styleFrom().copyWith(  
    backgroundColor: MaterialStateProperty.all(  
        Color.fromARGB(255, 147, 79, 2)),  
    ),  
,  
SizedBox(height: 3),  
Text(  
    'To save, please click on button.',  
    style: TextStyle(  
        fontSize: 14,  
        color: Colors.black,)),,
```

รูปที่ 3.54 (ต่อ) คำสั่งสร้างปุ่มเข้าสู่หน้าบันทึกสูตรอาหาร



รูปที่ 3.55 ปุ่มเข้าสู่หน้าบันทึกสูตรอาหาร

3.3.4 หน้าคำนวณโภชนาการของผลิตภัณฑ์ในสูตรอาหาร

3.3.4.1 ปุ่มเข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้

ปุ่มเข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้ ซึ่งจะมีข้อความอธิบาย เพื่อทำให้ผู้ใช้งานทราบว่าควรเลือกใช้สูตรใดในการคำนวณ จึงจะเหมาะสมกับข้อมูลบนฉลากของ ผลิตภัณฑ์ที่ผู้ใช้งานต้องการคำนวณโภชนาการ โดยเขียนคำสั่งดังรูปที่ 3.56

```
body: SingleChildScrollView(
    child: Container(
        padding: EdgeInsets.all(16),
        color: Color(0xFFFFAF3D6),
        child: Column(
            children: [
                Text('Calculate Nutrition of Foods',
```

รูปที่ 3.56 คำสั่งสร้างปุ่มเข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้

```

style: TextStyle(
    fontSize: 24,
    fontWeight: FontWeight.bold,
    color: Colors.brown,
),
),
SizedBox(height: 10),
Text(
    "Please check the food label if it has metabolizable
energy information like the one in the green box.",
style: TextStyle(
    color: Colors.brown,
),
),
SizedBox(height: 30),
Image.network(
    'https://blogger.googleusercontent.com/img/b/R29vZ2xl/A
VvXsEggx8OorqUsZrh-G4cwJ-mc19_Hzv381ECp7cwuy-qZWP-
hgvv1nq8wooGCbGwYRQhzGOT1bTjZ7lhotnV7yPAvKnMMKm_P1YRogu-omOkrNj2_O6Cr-
TqLgwdnpXqibrcrz_1IwaSA_V_02-8D5ZiCX65p18hncnwGVRsH0NP5-
CfCn83IkOSL72aY-ntY/w1684-h1069-p-k-no-
nu/%E0%B8%A3%E0%B8%B9%E0%B8%9B1.jpg',
height: 300,
width: 500,
),
SizedBox(height: 10),
Text(

```

รูปที่ 3.56 (ต่อ) คำสั่งสร้างปุ่มเข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้

```

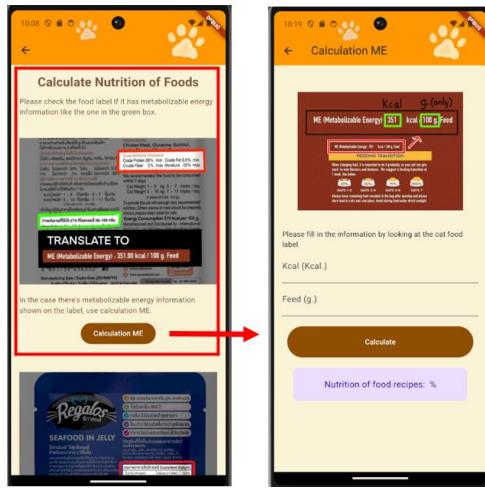
"In the case there's metabolizable energy information shown on the
label, use calculation ME.",

    style: TextStyle(
        color: Colors.brown,
    ),
    SizedBox(height: 10),
    ElevatedButton(
        onPressed: () {
            Navigator.push(
                context,
                MaterialPageRoute(builder: (context) => MEfoods()),
            );
        },
        child: Text(
            'Calculation ME',
            style: TextStyle(
                color: const Color.fromARGB(255, 255, 254, 254),
            ),
        ),
    ),
}

```

รูปที่ 3.56 (ต่อ) คำสั่งสร้างปุ่มเข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้

ปุ่มเข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้ จะมีการอธิบายว่า ผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้นั้นเป็นอย่างไร เพื่อทำให้ผู้ใช้งานทราบว่าฉลากของผลิตภัณฑ์ที่ ผู้ใช้งานต้องการคำนวณสามารถคำนวณแบบผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้ ได้หรือไม่ ดังรูปที่



รูปที่ 3.57 ปุ่มเข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์ที่ระบุพลังงานที่นำໄไปใช้ได้

3.3.4.2 ปุ่มเข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำໄไปใช้ได้
ปุ่มเข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำໄไปใช้ได้ ซึ่งจะมีข้อความ
อธิบายเพื่อทำให้ผู้ใช้งานทราบว่าควรเลือกใช้สูตรใดในการคำนวณ จึงจะเหมาะสมกับข้อมูลบนฉลากของ
ผลิตภัณฑ์ที่ผู้ใช้งานต้องการคำนวณโภชนาการ ดังรูปที่ 3.58

```

style: ButtonStyle(
    backgroundColor: MaterialStateProperty.all<Color>(
        Color.fromARGB(255, 147, 79, 2), ), ), ),
SizedBox(height: 40),
Image.network(
    'https://blogger.googleusercontent.com/img/b/R29vZ2xl/A
VvXsEhQnJsQ01IuPE1TWVFwziV7AjTXw8bvMamwr6NTMCXsXW816GD5Rpr9RdvLtwleurTt
VMTyR4i7-qbEMf88R669BcMkr9JjKFKRppV0yfe_3SWM4xYcO80yG2Mex8g_y4h1nOj-
VbC_Re3Lt-sUV0stcY_1JYt9sOYdm7UaNmMvF8rTTSzFlRLfNmRx20Nt/w1684-h1069-p-
k-no-nu/%E0%B8%A3%E0%B8%B9%E0%B8%9B2.jpg',
)

```

รูปที่ 3.58 คำสั่งสร้างปุ่มเข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำໄไปใช้ได้

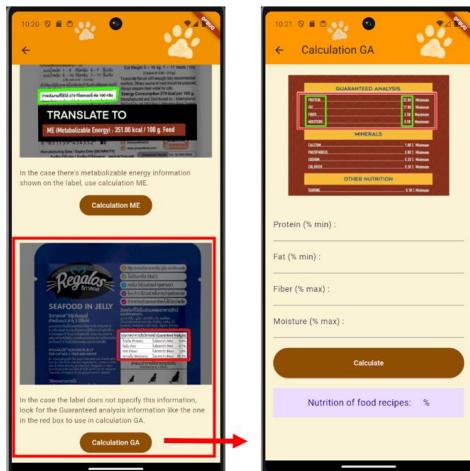
```

        height: 300,
        width: 500,),
SizedBox(height: 10),
Text(
    "In the case the label does not specify this information,
look for the Guaranteed analysis information like the one in the red box
to use in calculation GA.",
style: TextStyle(
    color: Colors.brown,,),
SizedBox(height: 10),
ElevatedButton(
onPressed: () {
    Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => GAfoods())),
    );
},
child: Text(
    'Calculation GA',
style: TextStyle(
    color: const Color.fromARGB(255, 255, 254, 254),
),),
style: ButtonStyle(
backgroundColor: MaterialStateProperty.all<Color>(
    Color.fromARGB(255, 147, 79, 2),
),),

```

รูปที่ 3.58 (ต่อ) คำสั่งสร้างปุ่มเข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้

ปุ่มเข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้ จะมีการอธิบายว่า ผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้นั้นเป็นอย่างไร เพื่อทำให้ผู้ใช้งานทราบว่าฉลากของผลิตภัณฑ์ที่ผู้ใช้งานต้องการคำนวณสามารถคำนวณแบบผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้ ได้หรือไม่ ดังรูปที่ 3.59



รูปที่ 3.59 ปุ่มเข้าสู่หน้าคำนวณโภชนาการของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้

3.3.4.3 ช่องกรอกข้อมูลของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้

การแสดงรูปภาพตัวอย่างเป็นคู่มือในการกรอกข้อมูลและการทำซองกรอกข้อมูลในการคำนวณ ต่อ ดังรูปที่ 3.60

```
body: SingleChildScrollView(
  child: Container(
    padding: EdgeInsets.all(16),
    child:
      Column (
```

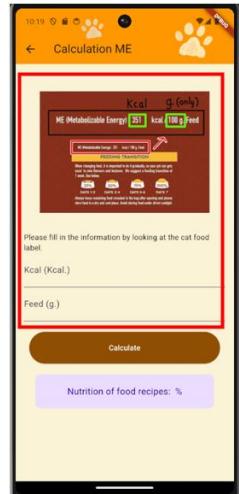
รูปที่ 3.60 คำสั่งสร้างช่องรับข้อมูลของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้

```

children: [
    Image.network(
        'https://blogger.googleusercontent.com/img/b/R29vZ2x1/A
VvXsEgMJppDD_eJQJlTWvx1qVhDQ_S-
byE_6xXVMiSvAU6kctv7s1JHqAkzKTxSy1Eg0DVIABXPWTJ1O12v9nAhKxfBdQTeYXCEnHe
pAhRe7UG8ylBMcDICRsc6pEba09SERCLpOxa4LCWXf-
xwIolXhQP2uqFB3NmH_U1FUC4O8zCw-
72fOAv4v9ZwjZKzLft/s320/%E0%B8%A3%E0%B8%B9%E0%B8%9B4.jpg',
        height: 300,
        width: 500,
    ),
    Text(
        "Please fill in the information by looking at the cat
food label."),
    buildNumberTextField(
        labelText: 'Kcal (Kcal.)',
        controller: kcalController,
    ),
    buildNumberTextField(
        labelText: 'Feed (g.)',
        controller: feedController,
    ),
],

```

รูปที่ 3.60 (ต่อ) คำสั่งสร้างช่องรับข้อมูลของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้ ช่องกรอกข้อมูลของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้ จะให้ผู้ใช้งานกรอกข้อมูล Kcal และ น้ำหนักอาหาร (กรัม) ดังรูปที่ 3.61



รูปที่ 3.61 ช่องกรอกข้อมูลของผลิตภัณฑ์ที่ระบุพลังงานที่นำมาใช้ได้

3.3.4.4 ปุ่มคำนวณและการแสดงผลลัพธ์ของผลิตภัณฑ์ที่ระบุพลังงานที่นำมาใช้ได้
การคำนวณโภชนาการของผลิตภัณฑ์ที่ระบุพลังงานที่นำมาใช้ได้ เมื่อกรอกข้อมูลแล้ว ทำการกดปุ่มคำนวณจะแสดงผลลัพธ์ในรูปแบบข้อความ ดังรูปที่ 3.62

```
SizedBox(height: 20),  
  
ElevatedButton(  
    onPressed: () {  
        if (kcalController.text.isEmpty ||  
            feedController.text.isEmpty) {  
            ScaffoldMessenger.of(context).showSnackBar(  
                SnackBar(  
                    content: Text('Please enter the amount of nutrients.'),  
                    duration: Duration(seconds: 2),),);  
    }  
},
```

รูปที่ 3.62 คำสั่งสร้างปุ่มคำนวณและการแสดงผลลัพธ์ของผลิตภัณฑ์ที่ระบุพลังงานที่นำมาใช้ได้

```

} else {

    setState(() {
        double kcal = double.parse(kcalController.text);
        double feed = double.parse(feedController.text);

        if (feed == 0.0) {
            ScaffoldMessenger.of(context).showSnackBar(
                SnackBar(
                    content:
                        Text('Feed value must be greater than zero.'),
                    duration: Duration(seconds: 2),
                ),
            );
        } else {
            result = (kcal / feed).toStringAsFixed(2);
        }
    });
}

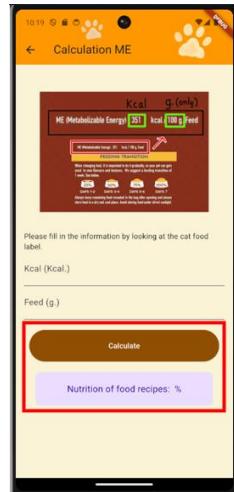
child: Text(
    'Calculate',
    style: TextStyle(
        color: Colors.white,
    ),
    style: ButtonStyle(
        backgroundColor:

```

รูปที่ 3.62 (ต่อ) คำสั่งสร้างปุ่มคำนวณและการแสดงผลลัพธ์ของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้

```
MaterialStateProperty.all(Color.fromARGB(255, 147, 79, 2)),  
    padding:  
MaterialStateProperty.all(EdgeInsets.symmetric(  
    horizontal: 150, vertical: 20)),  
,  
,  
  
SizedBox(height: 20),  
Container(  
decoration: BoxDecoration(  
    borderRadius: BorderRadius.circular(10),  
    color: Color.fromARGB(255, 235, 221, 255),  
,  
padding: EdgeInsets.symmetric(  
    horizontal: 60,  
    vertical:  
    20),  
child: Text(  
    'Nutrition of food recipes: \$result%',  
    style: TextStyle(fontSize: 17 ,color:  
Color.fromARGB(255, 41, 2, 159),),  
,),
```

รูปที่ 3.62 (ต่อ) คำสั่งสร้างปุ่มคำนวณและการแสดงผลลัพธ์ของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้



รูปที่ 3.63 บุ่มคำนวณและการแสดงผลลัพธ์ของผลิตภัณฑ์ที่ระบุพลังงานที่นำไปใช้ได้

3.3.4.5 ช่องรับข้อมูลของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้
การแสดงรูปภาพตัวอย่างเป็นคู่มือในการกรอกข้อมูลและการทำซองกรอกข้อมูลในการคำนวณ
ต่อ ดังรูปที่ 3.64

```
body: SingleChildScrollView(
  child: Padding(
    padding: const EdgeInsets.all(16.0),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.stretch,
      children: [
        Image.network('https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEjvpTEGt9a-UHSWhtPqbZDrf7qA_HHVJxOV8p74hND-Djo0D4fgWkP6k6fI3CU3q1Za1bOJAC_IxOPv1AhpJNA7XxYV_2VOuKRcnDk01qyFotFSfxWSBvfOAdf7i3A9f4hyB5RxqYJgCf2LVpDj_OqhuJsUSi9IPcZSL47XQzDsbJBYaUuc8c1aNI
DjWTB/s320/%E0%B8%A3%E0%B8%B9%E0%B8%9B3.jpg',
```

รูปที่ 3.64 คำสั่งสร้างช่องรับข้อมูลของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้

```
        height: 250,  
        width: 500,  
    ),  
    SizedBox(height: 20),  
    buildNumberTextField(  
        labelText: 'Protein (% min) :',  
        onChanged: (value) {  
            setState(() {  
                proteinValue = double.tryParse(value) ?? 0.0;  
            });  
        },  
    ),  
    buildNumberTextField(  
        labelText: 'Fat (% min) :',  
        onChanged: (value) {  
            setState(() {  
                fatValue = double.tryParse(value) ?? 0.0;  
            });  
        },  
    ),  
    buildNumberTextField(  
        labelText: 'Fiber (% max) :',  
        onChanged: (value) {  
            setState(() {  
                fiberValue = double.tryParse(value) ?? 0.0;});  
        },  
    ),
```

รูปที่ 3.64 (ต่อ) คำสั่งสร้างช่องรับข้อมูลของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้

```

) ,
buildNumberTextField(
    labelText: 'Moisture (% max) :',
    onChanged: (value) {
        setState(() {
            moistureValue = double.tryParse(value) ?? 0.0;
        });
    },
),
)
,
```

รูปที่ 3.64 (ต่อ) คำสั่งสร้างช่องรับข้อมูลของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้

ช่องกรอกข้อมูลของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้ จะให้ผู้ใช้งานกรอกข้อมูลโปรตีน (%) ไขมัน (%) ไฟเบอร์ (%) และความชื้น (%) ดังรูปที่ 3.65



รูปที่ 3.65 ช่องรับข้อมูลของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้

3.3.4.6 ปุ่มคำนวณและการแสดงผลลัพธ์ของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้ การคำนวณโดยใช้การของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้ เมื่อกรอกค่ามาแล้วและกดปุ่มคำนวณจะแสดงผลลัพธ์ในรูปแบบข้อความ ดังรูปที่ 3.66

```
SizedBox(height: 20),  
  
    ElevatedButton(  
  
        onPressed: () {  
  
            if (proteinValue == 0.0 ||  
  
                fatValue == 0.0 ||  
  
                fiberValue == 0.0 ||  
  
                moistureValue == 0.0) {  
  
                ScaffoldMessenger.of(context).showSnackBar(  
  
                    SnackBar(  
  
                        content: Text('Please enter the amount of  
nutrients.'),  
  
                        duration: Duration(seconds: 2),  
  
                    ),);  
  
            } else {  
  
                setState(() {  
  
                    result = ((proteinValue * 3.5) +  
  
                        (fatValue * 8.5) +  
  
                        ((100 -  
  
                            proteinValue -  
  
                            fatValue -
```

รูปที่ 3.66 คำสั่งสร้างปุ่มคำนวนและแสดงผลลัพธ์ของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้

```

        fiberValue -
        moistureValue) *
        3.5)) /
        100;
    });
}
},
child: Text(
    'Calculate',
    style: TextStyle(
        color: Colors.white,
    ),
),
style: ButtonStyle(
    backgroundColor:
        MaterialStateProperty.all(Color.fromRGBO(255,
147, 79, 2)),
    padding:
MaterialStateProperty.all(EdgeInsets.symmetric(
        horizontal: 10, vertical: 20)),
),
),
SizedBox(height: 20),
Container(
color:
    Color.fromRGBO(255, 235, 221, 255),

```

รูปที่ 3.66 (ต่อ) คำสั่งสร้างปุ่มคำนวณและแสดงผลลัพธ์ของผลิตภัณฑ์ที่ไม่ระบุพลังงานที่นำไปใช้ได้

```

padding: EdgeInsets.all(16) ,
        child: Center(
            child: Text(
                proteinValue == 0.0 ||
                fatValue == 0.0 ||
                fiberValue == 0.0 ||
                moistureValue == 0.0
                ? 'Nutrition of food recipes:      %'
                : 'Nutrition of food recipes:      $result %',
            style: TextStyle(fontSize: 18 ,color:
Color.fromARGB(255, 41, 2, 159),),
        ),
    
```

รูปที่ 3.66 (ต่อ) คำสั่งสร้างปุ่มคำนวณและการแสดงผลลัพธ์ของผลิตภัณฑ์ที่มีระบุพลังงานที่นำไปใช้ได้



รูปที่ 3.67 ปุ่มคำนวณและการแสดงผลลัพธ์ของผลิตภัณฑ์ที่มีระบุพลังงานที่นำไปใช้ได้

3.3.5 หน้าบันทึกสูตรอาหาร

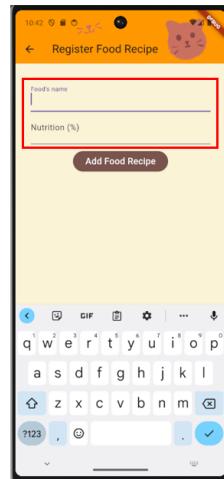
3.3.5.1 ช่องกรอกข้อมูลสูตรอาหาร

ช่องกรอกข้อมูลสูตรอาหารจะทำการเก็บชื่อสูตรอาหารและโภชนาการของผลิตภัณฑ์ โดยเขียนคำสั่งดังรูปที่ 3.68

```
children: [
    TextFormField(
        decoration: InputDecoration(labelText: "Food's name"),
        autofocus: true,
        controller: foodNameController,
        validator: (String? str) {
            if (str == null || str.isEmpty) {
                return "Please enter your Food's name.";
            }
            return null;
        },
    ),
    TextFormField(
        decoration: InputDecoration(labelText: "Nutrition (%)"),
        controller: nutritionValueController,
        keyboardType: TextInputType.number,
        validator: (String? str) {
            if (str == null || str.isEmpty) {
                return "Please enter Nutrition.";
            }
            return null;
        },
    ),
]
```

รูปที่ 3.68 คำสั่งสร้างช่องรับข้อมูลสูตรอาหาร

ช่องกรอกข้อมูลจะให้ผู้ใช้งานกรอกข้อมูลสูตรอาหารและโภชนาการของผลิตภัณฑ์ที่ต้องการเก็บข้อมูล ดังรูปที่ 3.69



รูปที่ 3.69 ช่องกรอกข้อมูลสูตรอาหาร

3.3.5.2 ปุ่มบันทึกสูตรอาหาร

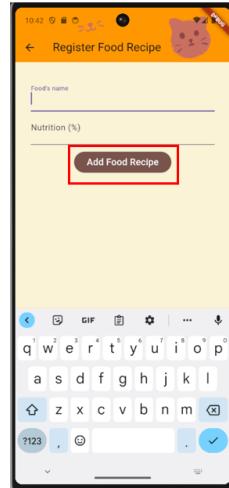
ปุ่มบันทึกสูตรอาหาร เมื่อผู้ใช้งานกดที่ปุ่มบันทึกข้อมูล ข้อมูลจะถูกนำไปเก็บในส่วนของ Transactions statement และค่อยส่งไปยังฐานข้อมูลต่อไป โดยเขียนคำสั่งดังรูปที่ 3.70

```
ElevatedButton(
    onPressed: () {
        if (formKey.currentState?.validate() ?? false) {
            var foodName = foodNameController.text;
            var nutritionValue =
                double.parse(nutritionValueController.text);
            Transactions statement = Transactions(
```

รูปที่ 3.70 คำสั่งสร้างปุ่มบันทึกสูตรอาหาร

```
foodName: foodName,  
nutritionValue: nutritionValue,  
);  
  
var provider =  
Provider.of<TransactionProvider>(context,  
listen: false);  
  
provider.addTransaction(statement);  
  
Navigator.pop(context);  
}  
},  
style: ButtonStyle(  
backgroundColor: MaterialStateProperty.all(  
Color(0xFF795548)),  
),  
child: Text(  
"Add Food Recipe",  
style: TextStyle(  
color: Colors.white,  
fontSize: 18,  
), ), ), ),
```

รูปที่ 3.70 (ต่อ) คำสั่งสร้างปุ่มบันทึกสูตรอาหาร



รูปที่ 3.71 ปุ่มบันทึกสูตรอาหาร

3.3.6 การเก็บข้อมูล

การเก็บข้อมูลจะมีส่วนของรูปแบบการจัดข้อมูล โดยเขียนคำสั่งดังรูปที่ 3.72

```
class Transactions {
    late int id;
    String title;
    String gender;
    String foodName;
    double nutritionValue;
    int age;
    double weight;
    Transactions(){
        this.id = 0,
```

รูปที่ 3.72 คำสั่งจัดรูปแบบข้อมูล

```

    this.title = '',
    this.gender = '',
    this.foodName = '',
    this.age = 0,
    this.nutritionValue = 0.0,
    double? weight,
}) : weight = weight != null ? weight : 0.0;
Map<String, dynamic> toMap() { return {
    'id': id,
    'title': title,
    'gender': gender,
    'foodName': foodName,
    'nutritionValue': nutritionValue,
    'age': age,
    'weight': weight, }; }

```

รูปที่ 3.72 (ต่อ) คำสั่งจัดรูปแบบข้อมูล

การเก็บข้อมูลในฐานข้อมูล โดยใช้คำสั่งดังรูปที่ 3.73

```

Future<int> insertData(Txns statement) async {
    var db = await this.openDatabase();
    var store = intMapStoreFactory.store("expense");
    var keyID = store.add(db, {

```

รูปที่ 3.73 คำสั่งการจัดเก็บข้อมูล

```

"title": statement.title,
"weight": statement.weight,
"age": statement.age,
"foodName": statement.foodName,
"nutritionValue": statement.nutritionValue,
"gender": statement.gender,
});

await db.close();

return keyID; }

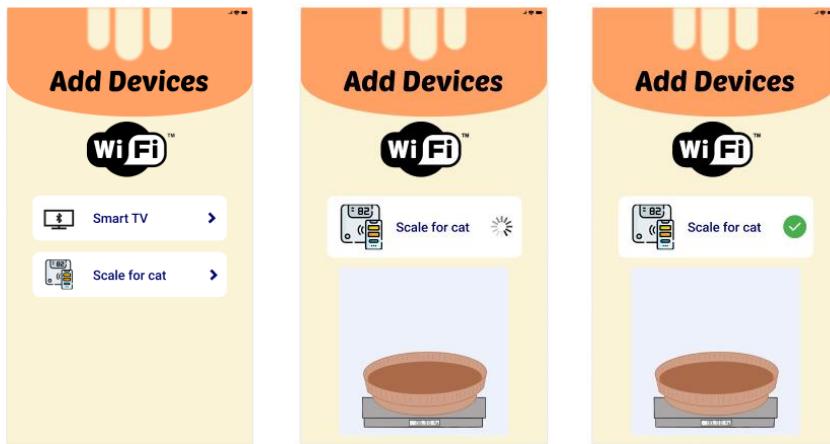
Future<List<Transactions>> loadAllData() async {
var db = await this.openDatabase();
var store = intMapStoreFactory.store("expense");
var snapshot = await store.find(db);
List<Transactions> transactionList = [];
for (var record in snapshot) {
if (record["weight"] != null &&
record["nutritionValue"] != null &&
record["gender"] != null &&
record["age"] != null &&
record["foodName"] != null &&
) {
int? age = record["age"] as int?;
transactionList.add(
Transactions(
id: record.key,

```

```
title: record["title"] as String,  
        weight: record["weight"] as double,  
        age: age ?? 0,  
        foodName: record["foodName"] as String,  
        nutritionValue: record["nutritionValue"] as double,  
        gender: record["gender"] as String,  
    );  
}  
}  
  
await db.close();  
  
return transactionList; }  
  
Future<void> updateTransaction(Transactions updatedData) async {  
    var db = await this.openDatabase();  
    var store = intMapStoreFactory.store("expense");  
    await store.record(updatedData.id).update(db, {  
        "title": updatedData.title,  
        "age": updatedData.age,  
        "weight": updatedData.weight,  
        "foodName": updatedData.foodName,  
        "nutritionValue": updatedData.nutritionValue,  
        "gender": updatedData.gender,  
    });  
    await db.close();  
}
```

3.4 การเชื่อมต่อเครื่องซึ่งน้ำหนักกับแอปพลิเคชัน

การใช้งานเครื่องซึ่งเลือกเครื่องซึ่งแบบไร้สายแอปพลิเคชันจะแสดงว่ามีอุปกรณ์อะไรบ้างที่พร้อมใช้งานจากนั้นทำการเชื่อมต่อกับเครื่องซึ่งน้ำหนัก โดยแสดงผลทางหน้าจอ ดังรูปที่ 3.74



ก. หน้าแสดงอุปกรณ์ที่สามารถเชื่อมต่อได้

ข. หน้าบอกรสถานะกำลังเชื่อมต่อ

ค. หน้าบอกรสถานะเชื่อมต่อสำเร็จ

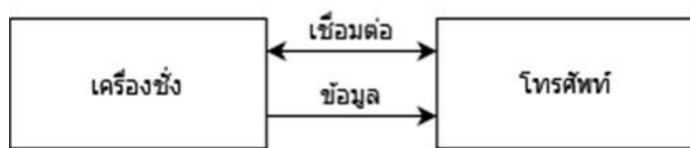
รูปที่ 3.74 การเชื่อมต่อเครื่องซึ่งน้ำหนักกับแอปพลิเคชัน

แอปพลิเคชันจะให้ผู้ใช้งานทำการเลือกเครื่องซึ่งแล้วเชื่อมต่อ โดยเมื่อเชื่อมต่อสำเร็จ แอปพลิเคชันจะแสดงให้ผู้ใช้งานทราบว่าเชื่อมต่อสำเร็จและพร้อมใช้งาน ดังรูปที่ 3.75 โดยใช้ไมโครคอนโทรลเลอร์ที่สามารถเชื่อมต่อกับสัญญาณอินเทอร์เน็ตแบบไร้สายได้ (NodeMcu) ซึ่งเป็นแพลตฟอร์มหนึ่งที่ใช้ช่วยในการสร้างงานทางด้านเครือข่ายรวมของอุปกรณ์ที่เชื่อมต่อถึงกัน (Internet of Things) (Supadanaisorn & Klang, 2019)



รูปที่ 3.75 การทำงานของแอปพลิเคชันในการเชื่อมต่อกับเครื่องชั่งน้ำหนัก

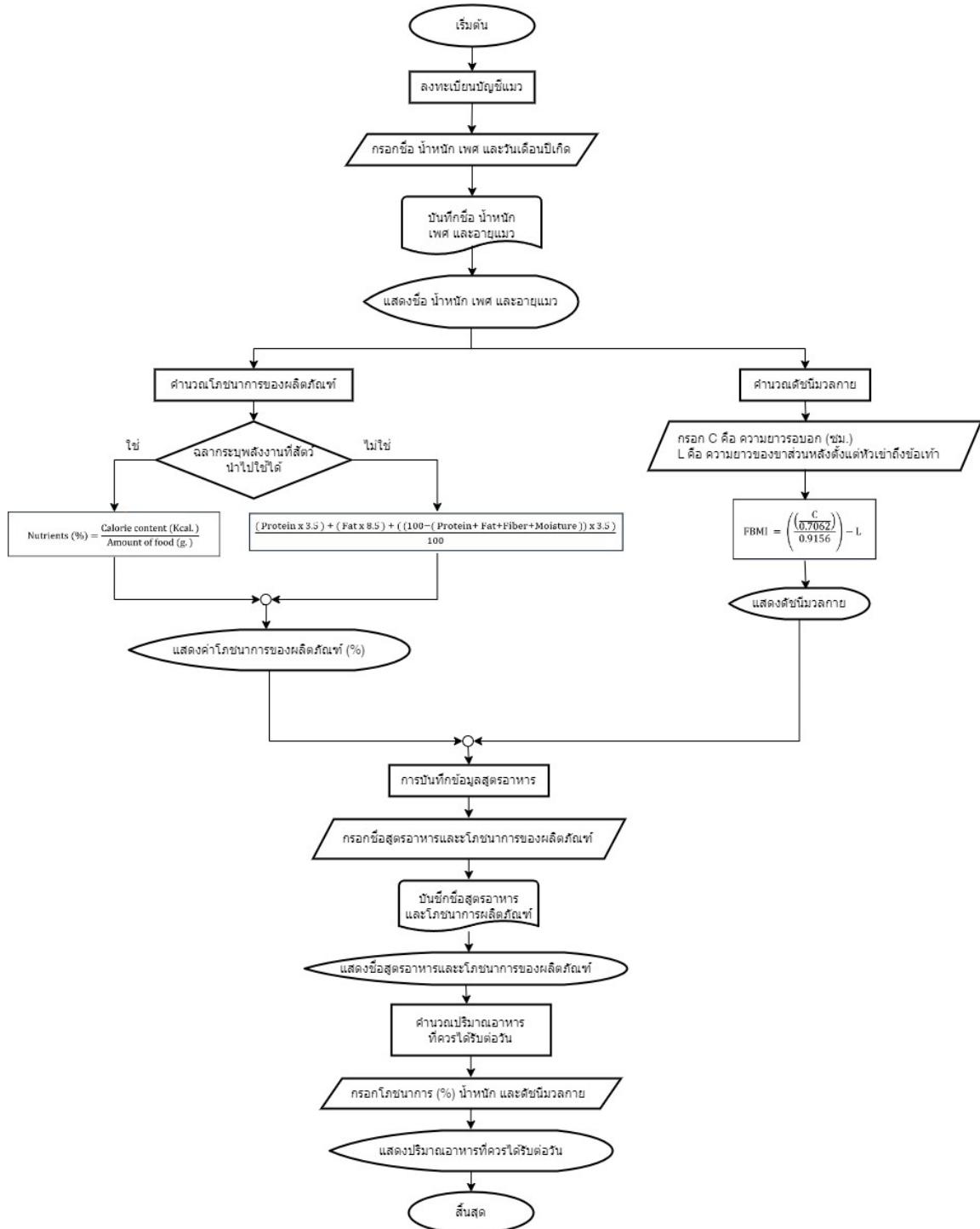
การทำงานของเครื่องชั่งน้ำหนักจะทำการเชื่อมต่อโดยส่งสัญญาณตอบรับกับแอปพลิเคชัน และเมื่อเชื่อมต่อสำเร็จเครื่องชั่งจะส่งข้อมูลน้ำหนักไปที่แอปพลิเคชัน ดังรูปที่ 3.76



รูปที่ 3.76 การทำงานของเครื่องชั่งน้ำหนักในการเชื่อมต่อกับแอปพลิเคชัน

3.5 การทำงานของแอปพลิเคชัน

การทำงานเริ่มจาก การลงทะเบียนบัญชีใหม่ การคำนวณโภชนาการของผลิตภัณฑ์ การคำนวณดัชนีมวลกาย การบันทึกสูตรอาหาร และการคำนวณปริมาณอาหารที่ควรได้รับต่อวัน ดังรูปที่ 3.77



รูปที่ 3.77 การทำงานของแอปพลิเคชัน

การลงทะเบียนข้อมูลต่าง ๆ ของแมว เมื่อกรอกข้อมูลครบถ้วน ผู้ใช้งานสามารถคำนวณโภชนาการของผลิตภัณฑ์หรือคำนวณดัชนีมวลกาย ซึ่งเลือกคำนวณสิ่งใดก่อนก็ได้ การคำนวณโภชนาการของผลิตภัณฑ์นั้นผู้ใช้งานต้องดูว่าฉลากมีข้อมูลอะไรบ้างแล้วเลือกใช้สูตรคำนวณที่เหมาะสม จากนั้นคำนวณโภชนาการของผลิตภัณฑ์และบันทึกสูตรอาหารในรูปแบบของกราฟ ส่วนการคำนวณดัชนีมวลกายจะให้ผู้ใช้งานกรอกข้อมูลแล้วคำนวณแล้วผลลัพธ์จะแสดงผลออกทางรูปและข้อความ ส่วนสุดท้ายการคำนวณปริมาณอาหารที่ควรได้รับต่อวัน (g./day) ดำเนินการโดยกรอกน้ำหนักของโภชนาการ (%) และเลือกดัชนีมวลกาย เมื่อดำเนินการทุกอย่างเรียบร้อยเป็นอันเสร็จสิ้น

บทที่ 4

การทดลองและผลการทดลอง

4.1 กล่าวนำ

ในส่วนของบทนี้จะกล่าวถึงการทดลองและผลการทดลองของแอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว ซึ่งผลการทดลองได้รับเคราะห์ตามสมมุติฐานผ่านแบบประเมินความพึงพอใจต่อการใช้งานแอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว

4.2 การทดลองและผลการทดลองใช้งานแอปพลิเคชัน

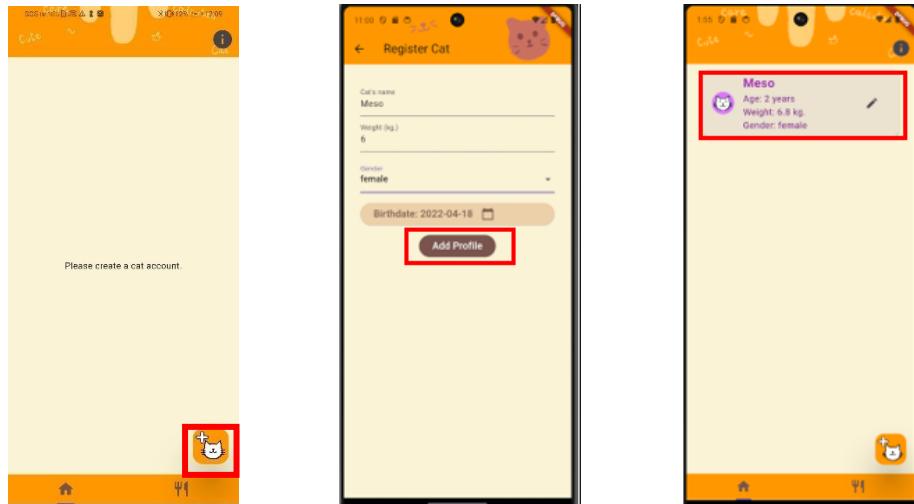
4.2.1 การทดลองบันทึกข้อมูล

4.2.1.1 การทดลองบันทึกและแก้ไขบัญชีแมว

- 1) การทดลองบันทึกบัญชีแมว
- 2) การทดลองแก้ไขข้อมูลของแมว

4.2.1.2 ผลการทดลองเพิ่มและแก้ไขบัญชีแมว

หน้าแรกของแอปพลิเคชันจะเป็นหน้าหลักที่ใช้แสดงผลการ์ดบัญชีแมวและสูตรอาหาร ซึ่งการเพิ่มบัญชีแมวสามารถทำได้โดยการกดปุ่มรูปแมวและเครื่องหมายบวก จากนั้นจะแสดงหน้ากรอกข้อมูลของแมว เมื่อทำการกรอกข้อมูลพร้อมกดปุ่ม Add Profile และแอปพลิเคชันจะแสดงการ์ดอยู่ที่หน้าหลัก ดังรูปที่ 4.1



ก. หน้าหลัก

ข. หน้ากรอกข้อมูลเม瓦

ค. หน้าแสดงการ์ดบัญชีเม瓦

รูปที่ 4.1 ตัวอย่างการทดลองเพิ่มบัญชีเม瓦

การแก้ไขข้อมูลเม瓦สามารถกรอกข้อมูลใหม่แทนข้อมูลเดิม จากนั้นกดปุ่มบันทึก ดังรูปที่ 4.2



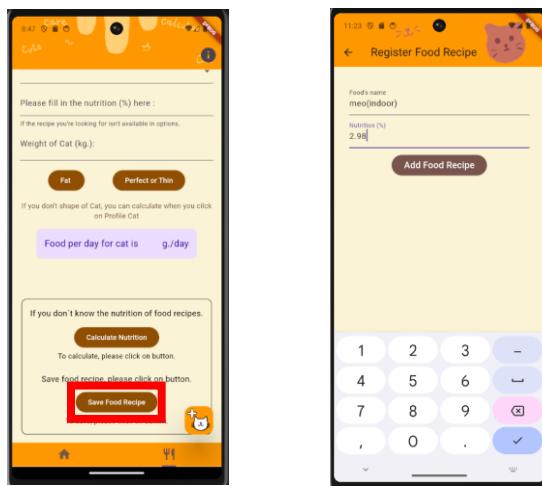
รูปที่ 4.2 ตัวอย่างการทดลองแก้ไขข้อมูลของเม瓦

4.2.1.3 การทดลองเข้าหน้าสูตรอาหารและบันทึกข้อมูลสูตรอาหาร

- 1) การทดลองกดปุ่มเข้าสู่หน้าบันทึกข้อมูลสูตรอาหาร
- 2) การทดลองบันทึกข้อมูลสูตรอาหาร

4.2.1.4 ผลการทดลองเข้าหน้าสูตรอาหารและบันทึกข้อมูลสูตรอาหาร

การบันทึกสูตรอาหารโดยการกดที่ปุ่ม Save Food Recipe จากนั้นจะแสดงหน้าบันทึกสูตรอาหาร ซึ่งจะให้กรอกชื่อสูตรอาหารและโภชนาการ (%) ดังรูปที่ 4.3



ก. ปุ่มเข้าสู่หน้าบันทึกสูตรอาหาร ข. หน้าบันทึกสูตรอาหาร

รูปที่ 4.3 ตัวอย่างการทดลองกดปุ่มเข้าสู่หน้าบันทึกสูตรอาหาร

การบันทึกข้อมูลเมื่อเสร็จสมบูรณ์แล้ว แอปพลิเคชันจะแสดงการ์ดสูตรอาหารที่หน้าหลัก ดังรูปที่ 4.4



รูปที่ 4.4 ตัวอย่างการทดลองบันทึกสูตรอาหาร

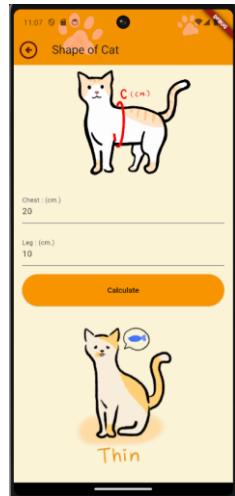
4.2.2 การทดลองคำนวน

4.2.2.1 การทดลองคำนวนดัชนีมวลกายสำหรับแมวและการแสดงผลลัพธ์

- 1) การทดลองคำนวนดัชนีมวลกายสำหรับแมว
- 2) การทดลองการแสดงผลลัพธ์ทั้ง 3 รูปแบบ

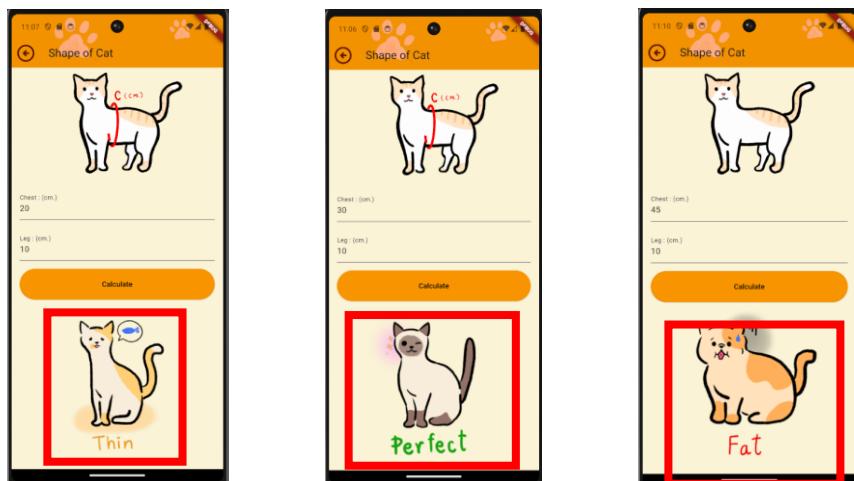
4.2.2.2 ผลการทดลองคำนวนดัชนีมวลกายสำหรับแมวและการแสดงผลลัพธ์

การคำนวนดัชนีมวลกายสำหรับแมวมีความจำเป็นที่ต้องทราบข้อมูลความยาวรอบอก (ซม.) และความยาวของขาส่วนหลังตั้งแต่หัวเข่าถึงข้อเท้า (ซม.) และนำมากรอกข้อมูลในช่องใส่ข้อมูล จากนั้นทำการกดปุ่มเพื่อคำนวนหาผลลัพธ์ และแอปพลิเคชันจะแสดงเป็นรูปภาพและข้อความ ดังรูปที่ 4.5



รูปที่ 4.5 ตัวอย่างการทดลองการคำนวณดัชนีมวลกายสำหรับแมว

ผลลัพธ์มีทั้งหมด 3 รูปแบบ คือ ผลลัพธ์แมวที่มีดัชนีมวลกายต่ำกว่าเกณฑ์มาตรฐาน ผลลัพธ์แมวที่มีดัชนีมวลกายอยู่ในเกณฑ์มาตรฐาน และผลลัพธ์แมวที่มีดัชนีมวลกายเกินเกณฑ์มาตรฐาน ดังรูปที่ 4.6



ก. ผลลัพธ์ดัชนีมวลกาย
ที่ต่ำกว่าเกณฑ์

ข. ผลลัพธ์ดัชนีมวลกาย
ที่อยู่ในเกณฑ์

ค. ผลลัพธ์ดัชนีมวลกาย
ที่เกินเกณฑ์

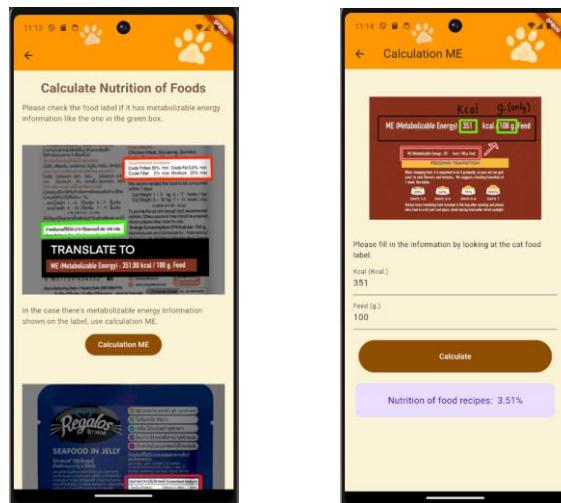
รูปที่ 4.6 ตัวอย่างการทดลองแสดงผลลัพธ์ดัชนีมวลกายของแมว

4.2.2.3 การทดลองคำนวณโภชนาการของผลิตภัณฑ์

- 1) การทดลองกดปุ่ม Calculation ME และปุ่ม Calculation GA
- 2) การทดลองคำนวณหาโภชนาการโดยที่ฉลากระบุพลังงานที่นำไปใช้ได้
- 3) การทดลองคำนวณหาโภชนาการโดยที่ฉลากไม่ระบุพลังงานที่นำไปใช้ได้

4.2.2.4 ผลการทดลองคำนวณโภชนาการของผลิตภัณฑ์

การทดลองกดปุ่ม Calculation ME จะแสดงผลหน้าคำนวณโภชนาการโดยที่ฉลากระบุพลังงานที่นำไปใช้ได้ ดังรูปที่ 4.7



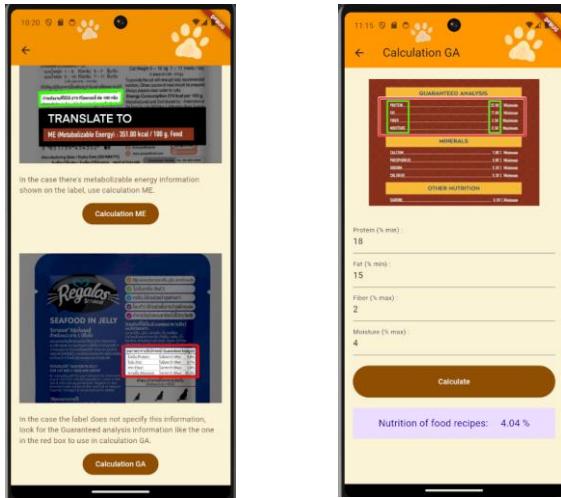
ก. ปุ่มเข้าสู่หน้า

Calculation ME

ข. หน้า Calculation ME

รูปที่ 4.7 ตัวอย่างการทดลองเข้าสู่หน้าคำนวณโภชนาการที่ผลิตภัณฑ์ระบุพลังงานที่นำไปใช้ได้

การทดลองกดปุ่ม Calculation GA จะแสดงผลหน้าคำนวณโภชนาการได้โดยที่ฉลากไม่ระบุพลังงานที่นำไปใช้ได้ ดังรูปที่ 4.8



ก. ปุ่มเข้าสู่หน้า

Calculation GA

ข. หน้า Calculation GA

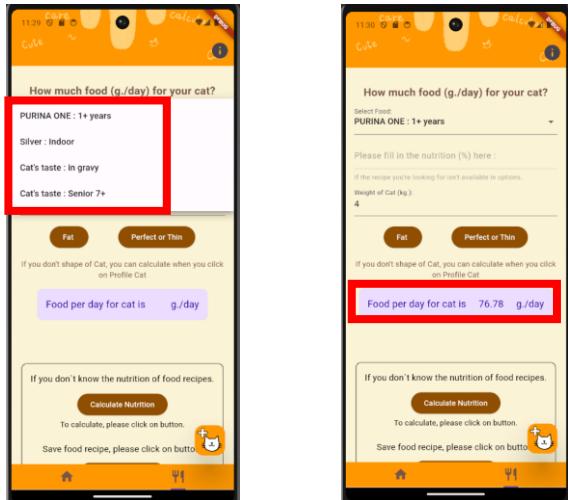
รูปที่ 4.8 ตัวอย่างการทดลองเข้าสู่หน้าคำนวณโภชนาการที่ผลิตภัณฑ์เมรับพลังงานที่นำไปใช้ได้

4.2.2.5 การทดลองคำนวณปริมาณของอาหารที่ควรได้รับต่อวัน

- 1) การทดลองเลือกสูตรอาหารในตัวเลือก Select Food
- 2) การทดลองกรอกโภชนาการด้วยตัวเอง
- 3) การทดลองเลือกสูตรอาหารในตัวเลือก Select Food พร้อมกรอกโภชนาการด้วยตัวเอง

4.2.2.6 ผลการทดลองคำนวณปริมาณของอาหารที่ควรได้รับต่อวัน

การคำนวณโดยเลือกใช้สูตรอาหารภายในแอปพลิเคชัน ด้วยการกดที่ Select Food และผลลัพธ์แสดงในรูปแบบข้อความ ดังรูปที่ 4.9



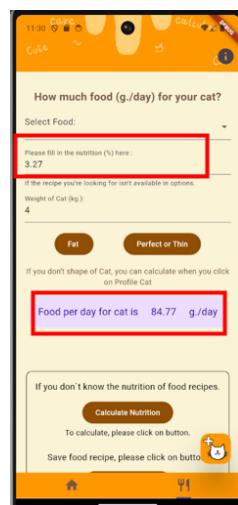
ก. ตัวเลือกใน Select Food

ข. ผลลัพธ์โดยใช้

ตัวเลือกในการคำนวณ

รูปที่ 4.9 ตัวอย่างการทดลองเลือกสูตรอาหารเพื่อคำนวณปริมาณของอาหารที่ควรได้รับต่อวัน

การกรอกค่าโภชนาการ (%) ด้วยตัวเองหากไม่มีสูตรอาหารที่ต้องการใน Select Food และผลลัพธ์แสดงในรูปแบบข้อความ ดังรูปที่ 4.10



รูปที่ 4.10 ตัวอย่างการทดลองกรอกโภชนาการด้วยตัวเอง

การป้องกันไม่ให้ใช้วิธีการคำนวณทั้ง 2 รูปแบบพร้อมกัน โดยแสดงในรูปแบบข้อความเตือน ดังรูปที่ 4.11



รูปที่ 4.11 ตัวอย่างการทดลองแสดงข้อความเตือนให้เลือกใช้วิธีการคำนวณทีละรูปแบบ

4.3 การประเมินผลการใช้งานของแอปพลิเคชัน

การประเมินความพึงพอใจของแอปพลิเคชันคำนวณปริมาณอาหารและต้นน้ำมูลกายสำหรับแมวโดยผู้ใช้งานมีจำนวน 12 คน ได้ใช้แบบสอบถามในการเก็บข้อมูล

4.3.1 ข้อมูลทั่วไป

แบบประเมินความพึงพอใจที่ได้จากการถุ่มผู้ทดลองใช้งานคำนวณหาค่าระดับเกณฑ์ความพึงพอใจโดยใช้เกณฑ์การประเมินระดับความคิดเห็นเพื่อหาค่ามาตรฐานส่วนประมาณค่า 5 ระดับ ตามวิธีของลิโคอร์ท ซึ่งมีดังนี้ (Srisaard, 2560)

ค่าเฉลี่ยเท่ากับ 4.51 – 5.00	หมายความว่า ระดับมากที่สุด
ค่าเฉลี่ยเท่ากับ 3.51 – 4.50	หมายความว่า ระดับมาก
ค่าเฉลี่ยเท่ากับ 2.51 – 3.50	หมายความว่า ระดับปานกลาง
ค่าเฉลี่ยเท่ากับ 1.51 – 2.50	หมายความว่า ระดับน้อย
ค่าเฉลี่ยเท่ากับ 1.00 – 1.50	หมายความว่า ระดับน้อยที่สุด

การวิเคราะห์ข้อมูลพื้นฐานส่วนบุคคลของกลุ่มผู้ทดลองใช้งานได้แก่ เพศ อายุ อุปกรณ์ที่ใช้งาน และจำนวนเมมที่ผู้ใช้งานเลี้ยง นำเสนอในรูปแบบตารางแสดงจำนวนคนและร้อยละ ดังตารางที่ 4.1 ถึง ตารางที่ 4.4

ตารางที่ 4.1 ผลจำนวนและร้อยละของกลุ่มผู้ทดลองใช้งาน โดยจำแนกตามเพศ

เพศ	จำนวน (คน)	ร้อยละ
ชาย	5	41.7
หญิง	7	58.3
รวม	12	100.00

จากตารางที่ 4.1 จะพบว่ากลุ่มผู้ทดลองใช้งานเป็นเพศชายจำนวน 5 คน คิดเป็นร้อยละ 41.7 และเป็นเพศหญิงจำนวน 7 คน คิดเป็นร้อยละ 58.3

ตารางที่ 4.2 ผลจำนวนและร้อยละของกลุ่มผู้ทดลองใช้งาน โดยจำแนกตามอายุ

อายุ (ปี)	จำนวน (คน)	ร้อยละ
ต่ำกว่า 20	3	25
20 - 29	6	50
30 - 50	3	25
มากกว่า 50	-	-
รวม	12	100.00

จากตารางที่ 4.2 จะพบว่ากลุ่มผู้ทดลองใช้งานอายุต่ำกว่า 20 ปี จำนวน 3 คน คิดเป็นร้อยละ 25 อายุ 20-29 ปี จำนวน 6 คน คิดเป็นร้อยละ 50 อายุ 30-50 ปี จำนวน 3 คน คิดเป็นร้อยละ 25 และไม่มีผู้ทดลองใช้งานที่อายุมากกว่า 50 ปี

ตารางที่ 4.3 ผลจำนวนและร้อยละของกลุ่มผู้ทดลองใช้งาน โดยจำแนกตามอุปกรณ์ที่ใช้งาน

อุปกรณ์ที่ใช้งาน	จำนวน (คน)	ร้อยละ
สมาร์ทโฟนแอนดรอยด์	9	75
แท็บเล็ตแอนดรอยด์	3	25
รวม	12	100.00

จากตารางที่ 4.3 จะพบว่ากลุ่มผู้ทดลองใช้งานใช้สมาร์ทโฟนแอนดรอยด์จำนวน 9 คน เป็นร้อยละ 75 และใช้งานแท็บเล็ตแอนดรอยด์จำนวน 3 คน เป็นร้อยละ 25

ตารางที่ 4.4 ผลจำนวนและร้อยละของกลุ่มผู้ทดลองใช้งาน โดยจำแนกตามจำนวนแมวที่ผู้ใช้เลี้ยง

จำนวนแมวที่ผู้ใช้เลี้ยง (ตัว)	จำนวน (คน)	ร้อยละ
1	6	50
2-3	4	33.3
4-10	2	16.7
มากกว่า 10	-	-
รวม	12	100.00

จากตารางที่ 4.4 จะพบว่ากลุ่มผู้ทดลองใช้งานเลี้ยงแมวจำนวน 1 ตัว เป็นร้อยละ 50 ผู้ทดลองใช้งานที่เลี้ยงแมวจำนวน 2-3 ตัว เป็นร้อยละ 33.3 ผู้ทดลองใช้งานที่เลี้ยงแมวจำนวน 4-10 ตัว เป็นร้อยละ 16.7 และไม่มีผู้ทดลองใช้งานที่เลี้ยงแมวมากกว่า 10 ตัว

4.3.2 ระดับความพึงพอใจต่อแอปพลิเคชัน

ผลการประเมินความพึงพอใจที่ได้จากการกลุ่มผู้ทดลองใช้งานมาประเมินหาค่าระดับเกณฑ์ โดยใช้เกณฑ์การประเมินระดับความคิดเห็น เพื่อหาค่ามาตราส่วนประมาณค่า 5 ระดับ ตามวิธีของลิเครอร์ท โดยมีการประเมินความพึงพอใจต่อการออกแบบแอปพลิเคชันและการใช้งานนำเสนอด้วยรูปแบบตารางแสดงค่าเฉลี่ย (\bar{x}) และส่วนเบี่ยงเบนมาตรฐาน (S.D.) ดังตารางที่ 4.5 ถึงตารางที่ 4.7

ตารางที่ 4.5 ผลการประเมินความพึงพอใจต่อการออกแบบแอปพลิเคชัน

หัวข้อการประเมิน	\bar{x}	S.D.	ระดับ
ขนาดตัวอักษรเมื่อความเหมาะสม	4.08	0.66	มาก
รูปแบบตัวอักษรอ่านง่าย	4.16	0.57	มาก
โทนสีของแอปพลิเคชันมีความสวยงาม	3.83	0.57	มาก
เมนูเข้าใจง่าย	3.33	0.65	ปานกลาง
เฉลี่ยรวม	3.85	0.37	มาก

จากตารางที่ 4.5 ผลการประเมินความพึงพอใจต่อการออกแบบแอปพลิเคชัน มีค่าเฉลี่ยรวมเท่ากับ 3.85 และมีส่วนเบี่ยงเบนมาตรฐาน 0.37 จัดอยู่ในระดับมาก และพบว่ามี 3 รายการส่วนใหญ่ที่ผู้ประเมินพึงพอใจอยู่ในระดับมาก คือ อันดับแรก รูปแบบตัวอักษรอ่านง่าย มีค่าเฉลี่ยอยู่ที่ 4.16 ผลการประเมินความพึงพอใจอยู่ในระดับมาก อันดับที่สอง ขนาดตัวอักษรเมื่อความเหมาะสม มีค่าเฉลี่ยอยู่ที่ 4.08 ผลการประเมินความพึงพอใจอยู่ในระดับมาก และอันดับที่สาม โทนสีของแอปพลิเคชันมีความสวยงาม มีค่าเฉลี่ยอยู่ที่ 3.83 ผลการประเมินความพึงพอใจอยู่ในระดับมาก

ตารางที่ 4.6 ผลการประเมินความพึงพอใจต่อการใช้งานแอปพลิเคชัน

หัวข้อการประเมิน	\bar{x}	S.D.	ระดับ
ความเร็วในการแสดงผลลัพธ์	4.16	0.71	มาก
แอปพลิเคชันใช้งานได้ลื่นไหลไม่สะดุด	4.16	0.83	มาก
ช่วยคำนวณรูปร่างของแมว	4.5	0.52	มาก
ช่วยคำนวณปริมาณอาหารต่อวัน (g./day) ของแมว	4.41	0.51	มาก
เฉลี่ยรวม	4.31	0.15	มาก

จากตารางที่ 4.6 ผลการประเมินความพึงพอใจต่อการใช้งานแอปพลิเคชัน มีค่าเฉลี่ยรวมเท่ากับ 4.31 และมีส่วนเบี่ยงเบนมาตรฐาน 0.15 จัดอยู่ในระดับ มาก และพบว่าทั้ง 4 รายการผู้ประเมินพึงพอใจอยู่ในระดับ มาก คือ อันดับแรก ช่วยคำนวณรูปร่างของแมว มีค่าเฉลี่ยอยู่ที่ 4.5 ผลการประเมินความพึงพอใจอยู่ในระดับ มาก อันดับที่สอง ช่วยคำนวณปริมาณอาหารต่อวัน (g./day) ของแมว มีค่าเฉลี่ยอยู่ที่ 4.41 ผลการประเมินความพึงพอใจอยู่ในระดับ มาก และอันดับที่สาม มี 2 รายการ ได้แก่ ความเร็วในการแสดงผลลัพธ์ มีค่าเฉลี่ยอยู่ที่ 4.16 ผลการประเมินความพึงพอใจอยู่ในระดับ มาก และแอปพลิเคชันใช้งานได้ลื่นไหลไม่สะดุด มีค่าเฉลี่ยอยู่ที่ 4.16 ผลการประเมินความพึงพอใจอยู่ในระดับ มาก

ตารางที่ 4.7 ผลรวมการประเมินความพึงพอใจ

หัวข้อการประเมิน	\bar{x}	S.D.	ระดับ
ความพึงพอใจต่อการออกแบบแอปพลิเคชัน	3.85	0.37	มาก
ความพึงพอใจต่อการใช้งานแอปพลิเคชัน	4.31	0.15	มาก
เฉลี่ยรวม	4.08	0.15	มาก

จากตารางที่ 4.7 ผลการประเมินความพึงพอใจต่อการออกแบบแอปพลิเคชันมีค่าเฉลี่ยเท่ากับ 3.85 ส่วนเบี่ยงเบนมาตรฐาน 0.37 จัดอยู่ในระดับ มาก และความพึงพอใจต่อการใช้งานแอปพลิเคชัน

มีค่าเฉลี่ยเท่ากับ 4.31 ส่วนเบี่ยงเบนมาตรฐาน 0.15 จัดอยู่ในระดับมาก และพบว่าการประเมินทั้ง 2 รายการผู้ประเมินพึงพอใจอยู่ในระดับมาก คือ อันดับแรก ความพึงพอใจต่อการออกแบบแอปพลิเคชัน มีค่าเฉลี่ยอยู่ที่ 3.85 ผลการประเมินความพึงพอใจอยู่ในระดับมาก และอันดับที่สอง ความพึงพอใจต่อ การใช้งานแอปพลิเคชัน มีค่าเฉลี่ยอยู่ที่ 4.31 ผลการประเมินความพึงพอใจอยู่ในระดับมาก ทำให้ผล การประเมินความพึงพอใจโดยกลุ่มผู้ทดลองใช้งานมีค่าเฉลี่ยรวมเท่ากับ 4.08 ส่วนเบี่ยงเบนมาตรฐาน 0.15 จัดอยู่ในระดับมาก

บทที่ 5

บทสรุป

5.1 สรุป

แอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว จัดทำขึ้นเพื่อแก้ปัญหาแมวที่มีดัชนีมวลกายเกินเกณฑ์มาตรฐานโดยปัจจัยจากการกินอาหารในปริมาณที่ไม่เหมาะสมกับร่างกายด้วยการใช้งานแอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว ซึ่งการใช้งานที่คาดหวังสามารถใช้งานได้ ดังต่อไปนี้

1. การเข้าสู่ระบบผู้ใช้งาน
 - 1.1 การลงทะเบียนบัญชีผู้ใช้งาน
 - 1.2 การเข้าสู่ระบบ
2. การเชื่อมต่อกับเครื่องซึ่งน้ำหนักผ่านการเชื่อมต่ออินเทอร์เน็ตแบบไร้สาย
3. การบันทึกข้อมูล
 - 3.1 การบันทึกและแก้ไขบัญชีของแมว
 - 3.2 การบันทึกสูตรอาหารและนำไปใช้ในการคำนวณปริมาณอาหารที่ควรได้รับต่อวันได้
4. การคำนวณต่าง ๆ
 - 4.1 การคำนวณดัชนีมวลกายสำหรับแมว
 - 4.2 การคำนวณพลังงานที่แมวควรได้รับ
 - 4.3 การคำนวณพลังงานที่สัตว์นำໄไปใช้ได้ของผลิตภัณฑ์
 - 4.4 การคำนวณปริมาณอาหารที่ควรได้รับต่อวัน
5. การแสดงกราฟแนวโน้มน้ำหนัก ดัชนีมวลกาย และปริมาณอาหาร

แอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมวเขียนด้วยภาษา Dart บนโปรแกรม Flutter ภายในแอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมวจะอ้างอิงทฤษฎีสุขภาพของแมวและโภชนาการของผลิตภัณฑ์ ปัจจุบันแอปพลิเคชันสามารถใช้งานได้ 2 ส่วน คือ ส่วนการบันทึกข้อมูลและส่วนการคำนวณต่าง ๆ

1. การบันทึกข้อมูล
 - 1.1 การบันทึกและแก้ไขบัญชีของแมว

- 1.2 การบันทึกสูตรอาหาร
2. การคำนวณต่าง ๆ
 - 2.1 การคำนวณดัชนีมวลกายสำหรับแมว
 - 2.2 การคำนวณพลังงานที่แมวควรได้รับ
 - 2.3 การคำนวณพลังงานที่สัตว์นำไปใช้ได้ของผลิตภัณฑ์
 - 2.4 การคำนวณปริมาณอาหารที่ควรได้รับต่อวัน

จากที่กล่าวมานั้นแอปพลิเคชันไม่สามารถทำงานได้ครบตามที่คาดหวัง ซึ่งสิ่งที่ทำไม่ได้นั้นประกอบด้วย การเข้าสู่ระบบผู้ใช้งาน การเชื่อมต่อกับเครื่องซั่งน้ำหนักผ่านการเชื่อมต่ออินเทอร์เน็ตแบบไร้สาย การนำสูตรอาหารไปใช้ในการคำนวณปริมาณอาหารที่ควรได้รับต่อวันได้ และการแสดงกราฟแนวโน้มน้ำหนัก ดัชนีมวลกาย และปริมาณอาหาร แต่การทำงานที่เป็นส่วนสำคัญ คือ การคำนวณต่าง ๆ ใช้งานได้ครบถ้วน จึงสามารถนำแอปพลิเคชันไปใช้งานได้

ผลการนำแบบประเมินความพึงพอใจต่อการใช้งานแอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว โดยผู้ประเมินทั้งหมด 12 คน พบร่วมความพึงพอใจต่อการออกแบบแอปพลิเคชัน มีค่าเฉลี่ย 3.85 อยู่ในระดับมาก และความพึงพอใจต่อการใช้งานแอปพลิเคชัน มีค่าเฉลี่ย 4.31 อยู่ในระดับมาก ทำให้ผลรวมความพึงพอใจต่อแอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว มีค่าเฉลี่ยอยู่ที่ 4.08 อยู่ในระดับมาก

5.2 ปัญหาและวิธีการแก้ไข

1. การแก้ไขข้อมูลในบัญชีแมวไม่สามารถแก้ไขได้ เมื่อลองทดสอบพบว่ามีการรับข้อมูลแล้ว
วิธีการแก้ไข ข้อมูลที่แก้ไขใหม่ไม่สามารถแสดงแทนที่ข้อมูลเก่าได้จึงส่งผลให้ต้องแก้ไขข้อมูล โดยลบข้อมูลเดิมออกก่อน และแทนด้วยข้อมูลที่แก้ไข
2. การเพิ่มการดับบล์ชีแมวสามารถเพิ่มได้ แต่เพิ่มสูตรอาหารไม่ได้
วิธีการแก้ไข ให้แสดงเพียงข้อมูลที่รับล่าสุด เพราะการแสดงผลใช้ฟอร์มเดียวกัน จึงต้องสร้างเงื่อนไขให้แสดงเฉพาะข้อมูลที่รับล่าสุด เช่น บันทึกสูตรอาหาร ก็จะแสดงแค่ชื่อสูตรอาหารและโภชนาการของผลิตภัณฑ์

5.3 แนวทางการพัฒนา

1. พัฒนาการเข้าสู่ระบบผู้ใช้งานให้สามารถลงทะเบียนบัญชีผู้ใช้งานและเข้าสู่ระบบได้
2. พัฒนาเครื่องซึ่งน้ำหนักให้สามารถใช้งานและเชื่อมต่อกับแอปพลิเคชันได้
3. พัฒนาให้สามารถนำสูตรอาหารที่บันทึกไปใช้ในการคำนวณปริมาณอาหารที่ควรได้รับต่อวันได้
4. พัฒนาให้มีการแสดงกราฟแนวโน้มน้ำหนัก ดัชนีมวลกาย และปริมาณอาหาร
5. พัฒนาความละเอียดในการคำนวณให้สามารถคำนวณได้แม่นยำมากขึ้น เช่น เกณฑ์การคำนวณดัชนีมวลกายและการคำนวณปริมาณอาหารที่ควรได้รับต่อวัน โดยมีการแทนค่าตัวแปรในการคำนวณให้ละเอียดมากขึ้น
6. พัฒนาการใช้งานให้เข้าใจง่ายและลดจำนวนหน้าเพื่อให้สะดวกในการใช้งานมากขึ้น
7. พัฒนาให้สามารถวัดดัชนีมวลกายของแมวโดยใช้เพียงรูปถ่าย
8. พัฒนาแอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว ให้มีข้อมูลโรงพยาบาลสัตว์และสามารถติดต่อกับโรงพยาบาลผ่านแอปพลิเคชันได้โดยตรง
9. พัฒนาให้มีการแจ้งเตือนต่าง ๆ เช่น ครบกำหนดเวลาที่ต้องคำนวณดัชนีมวลกายหรือปริมาณอาหารที่ควรได้รับต่อวันอีกครั้ง

บรรณานุกรม

- Butterwick, R. (2000). How Fat is That Cat? *Journal of Feline Medicine and Surgery*, 2(2), 91–94. <https://doi.org/10.1053/jfms.2000.0078>
- Kathy Walrath & Seth Ladd. (2012). *Dart: Up and Running [Book]*.
<https://www.oreilly.com/library/view/dart-up-and/9781449330880/>
- Monteiro, B. P., Lascelles, B. D. X., Murrell, J., Robertson, S., Steagall, P. V. M., & Wright, B. (2023). 2022 WSAVA guidelines for the recognition, assessment and treatment of pain. *Journal of Small Animal Practice*, 64(4), 177–254.
<https://doi.org/10.1111/jsap.13566>
- Purina. (2021). แนะนำข้อมูลอาหารสำหรับลูกแมว แต่ละชนิด / Purina.
<https://www.purina.co.th/articles/cats/feeding/what-cats-eat/cats-food-tips>
- Royal Canin. (2023). / Royal Canin TH. <https://www.royalcanin.com/th/cats/health-and-wellbeing/health-risks-of-overweight-and-obese-cats>
- กิตติกร ประเสริฐศักดิ์. (2022). *Complete UX/UI Design* / borntoDev School.
<https://school.borntodev.com/course/complete-uxui-design>
- เฉลิมพล เล็กเจริญสุข. (2021). VPN Magazine. https://readvpn.com/Topic/Info/a1a40224-1929-4360-b801-5e8a4bdbfb9?fbclid=IwZXh0bgNhZW0CMTAAAR3MnhXbjlAC5g2YAZspHNVdsfTGnulxBsll8S1HrUgbEMMcUrDJdRKueY_aem_ATgq6ynCsq4cAjjF0niO0ZOaGnRq4w3naqaqOW0kuC0yN8aRpK7EGpftReqYvCJgydjcyiuwT6WMTWNojbr9Xv-l
- กานิดา เมฆโสภาวรรณ. (2023). ความเสี่ยงจากโรคอ้วนในแมว.
https://www.vet4hospital.com/index.php?route=content/content&path=7&content_id=528
- วรวิทย์ จันทร์สุวรรณ. (2020). เทคนิคการใช้เครื่องชั่งในห้องปฏิบัติการ. *Active Learning : Learning for All*. https://web.rmutp.ac.th/woravith/?page_id=3337
- ศันสนีย์ สมบูรณ์. (2023). การศึกษาความต้องของผู้เลี้ยงแมวในการเลือกซื้ออาหารแมวสำหรับปะดับพรีเมียม.

บรรณานุกรม (ต่อ)

ศุภกิตติ์ สีดำ. (2023, January 27). แมวอ้วน ต้นตอโรคร้ายแรง! มาลดน้ำหนักแมวกันดีกว่า!

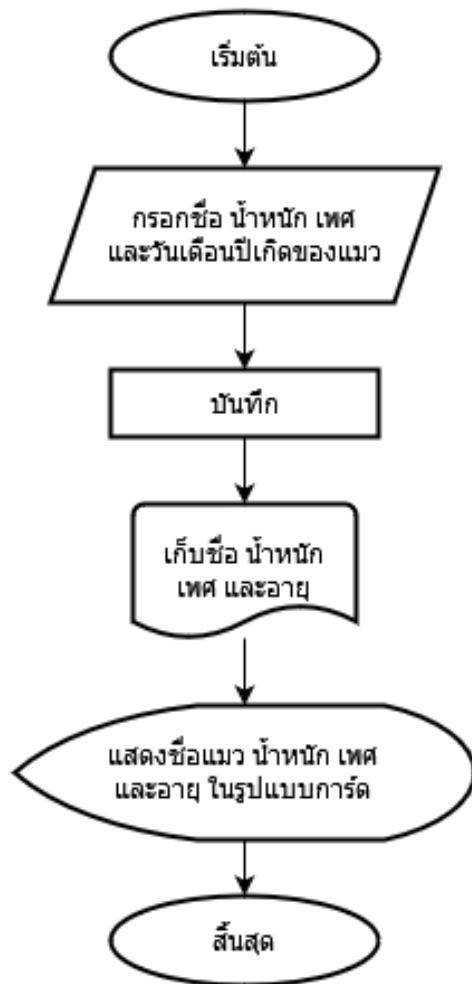
<https://ultravet-pet.com/cat-obesity/>

สุกดนัยสร, ร., ปานกลาง, ร., & ศรีชุมแสง, จ. (2019). การออกแบบและสร้างระบบวัดตัวน้ำหนัก (BMI) อัตโนมัติแสดงผลผ่านเครือข่ายอินเทอร์เน็ต. วารสารวิชาการคณะเทคโนโลยีอุตสาหกรรม
มหาวิทยาลัยราชภัฏเทพศรี
ISSN XXXX-XXXX, 14(2), Article 2.

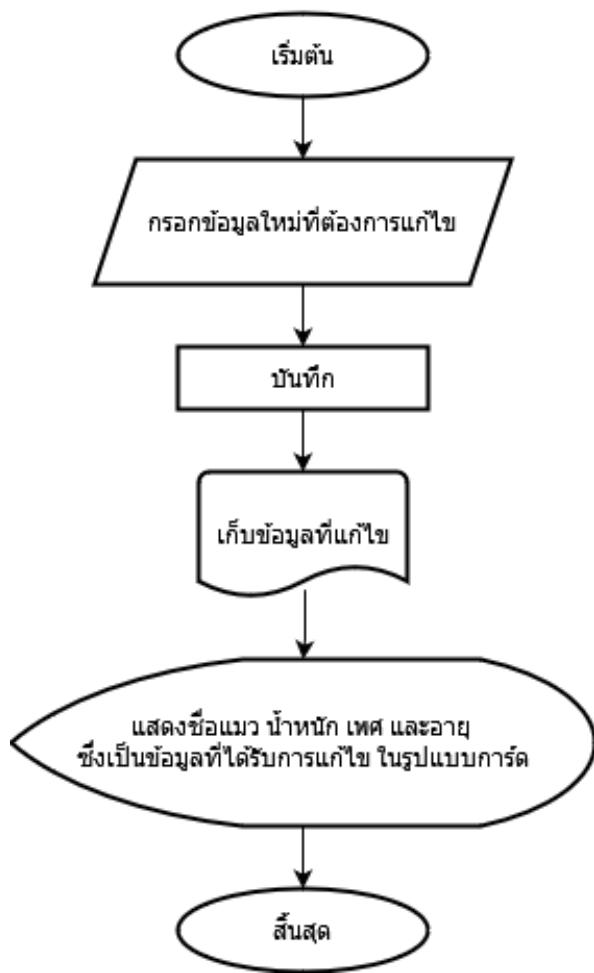
อารีรัตน์ อาการวิตาต. (2005). ภาวะอ้วนกับหมาและแมว.

<https://www.vet.cmu.ac.th/cmvj/document/journal/08%20Areerat.pdf>

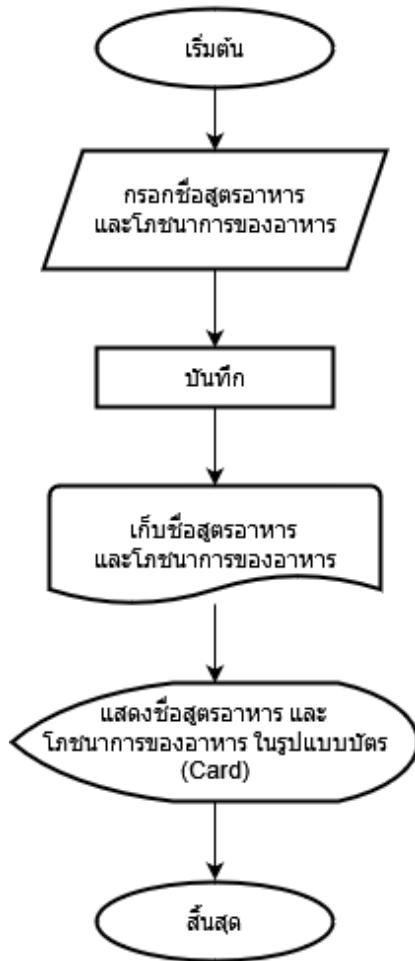
ภาคผนวก ก
ผังการทำงาน



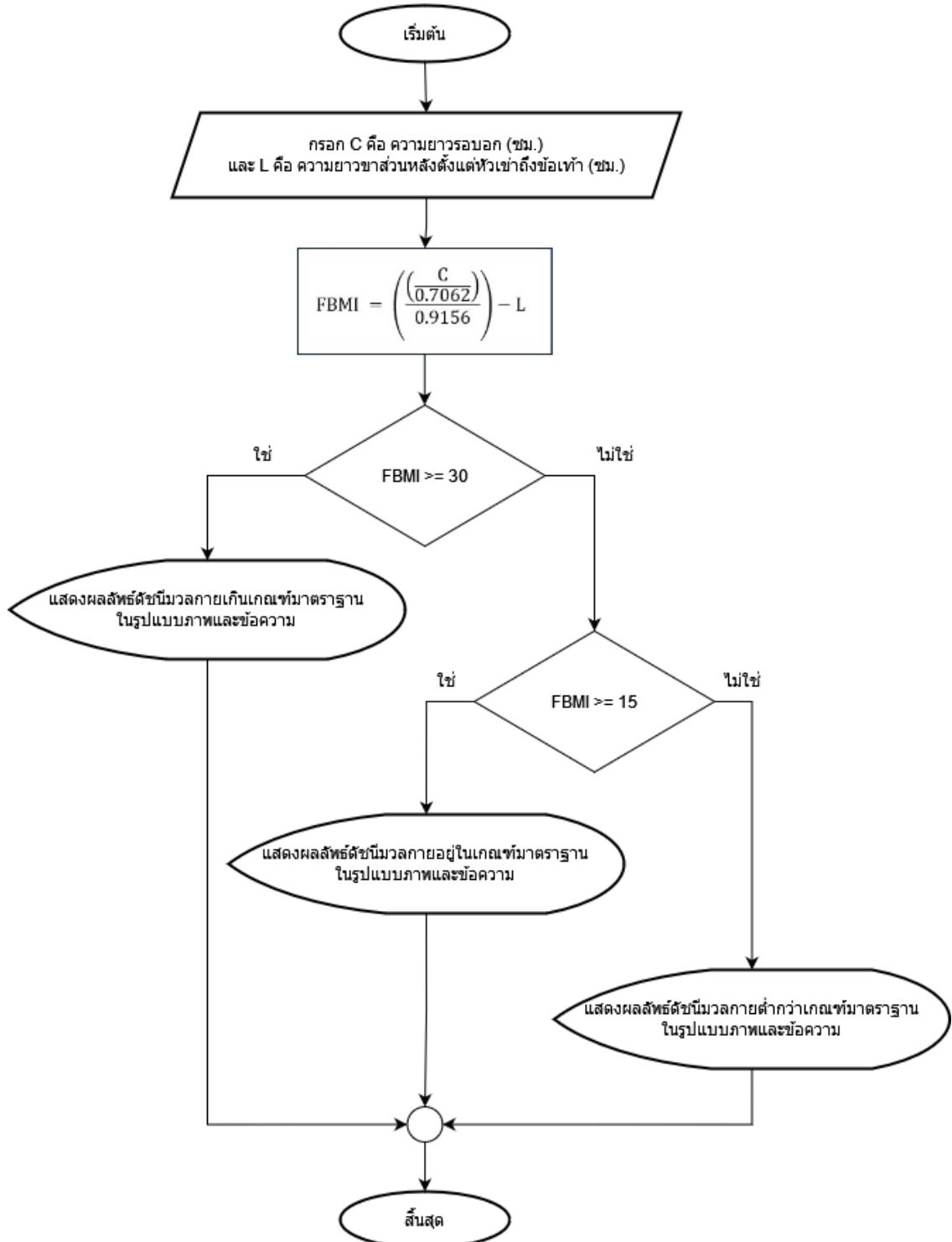
รูปที่ ก.1 แผนผังการลงท่าเปลี่ยนบัญชีแมว



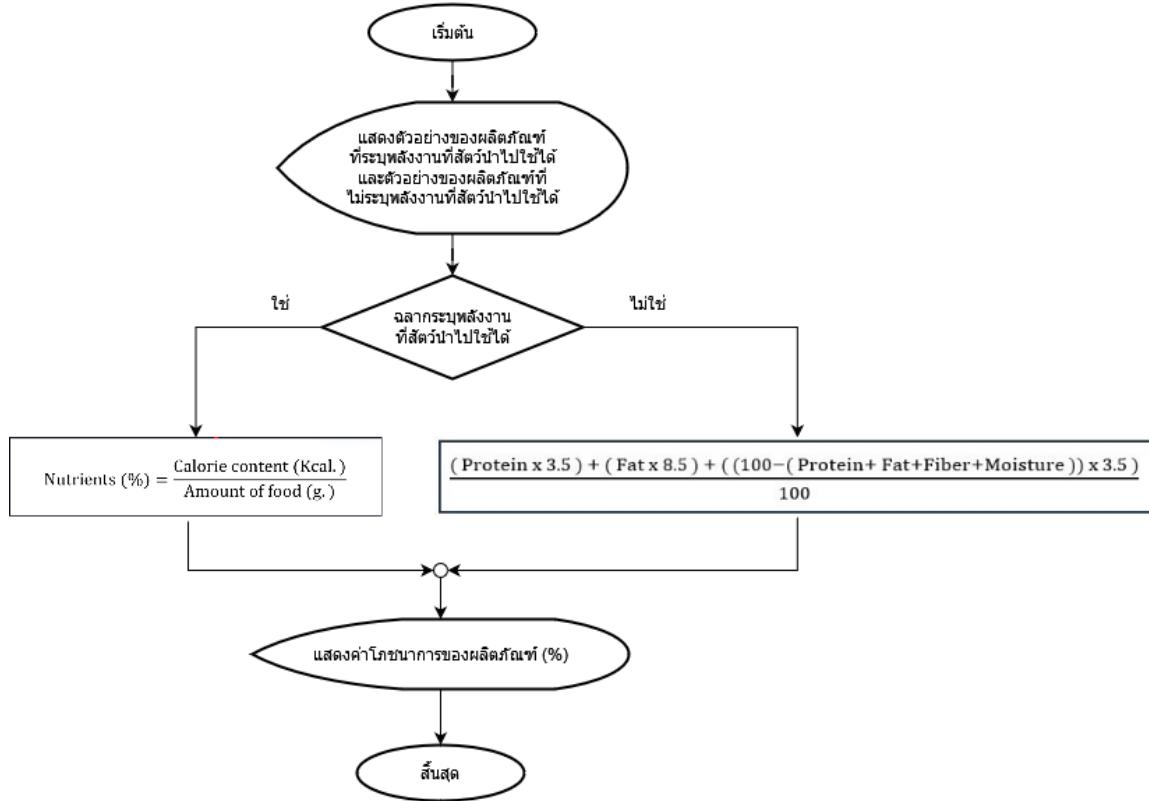
รูปที่ ก.2 แผนผังการแก้ไขข้อมูลในบัญชีแนว



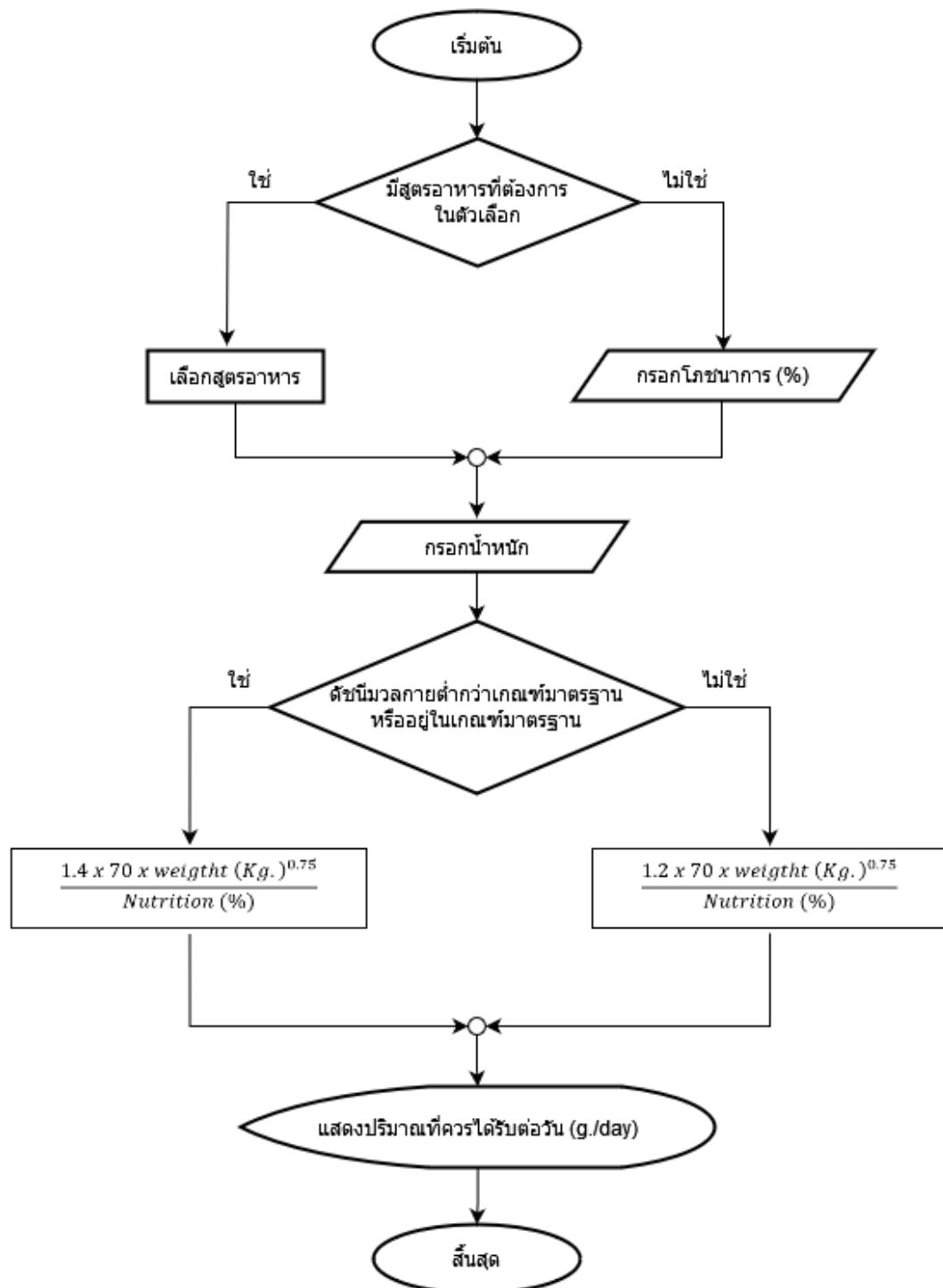
รูปที่ ก.3 แผนผังการบันทึกข้อมูลสูตรอาหาร



รูปที่ ก.4 แผนผังการคำนวณดัชนีมวลกาย



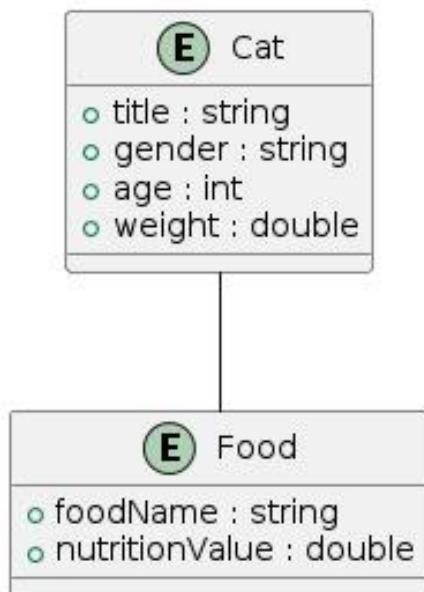
รูปที่ ก.5 แผนผังการคำนวณสูตรอาหาร



รูปที่ ก.6 แผนผังการคำนวณปริมาณอาหารที่ควรได้รับต่อวัน

ภาคผนวก ข

แผนผังความลึกพื้นที่ของฐานข้อมูล

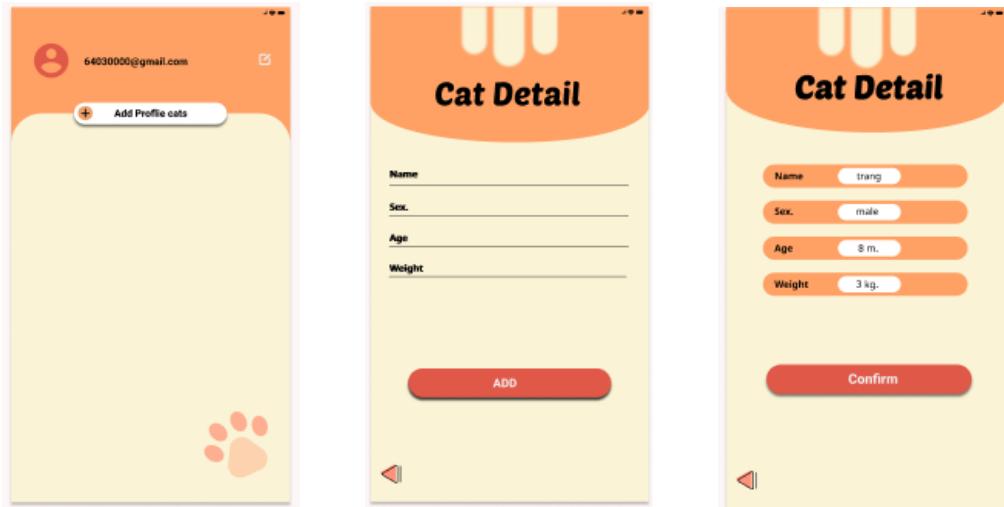


รูปที่ ข.1 แผนผังความสัมพันธ์ของฐานข้อมูล

ภาคผนวก ค
แผนผังการออกแบบแอปพลิเคชัน

1. การลงทะเบียนแมว

การลงทะเบียนแมวโดยรับข้อมูลชื่อ เพศ อายุ และน้ำหนัก จากนั้นบันทึกข้อมูล ดังรูปที่ ค.1



ก. หน้าหลัก

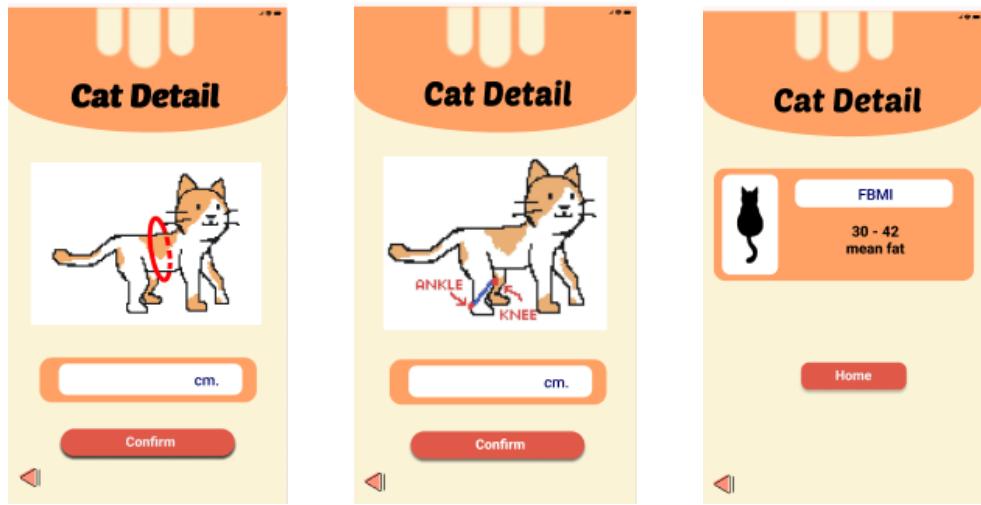
ข. หน้ากรอกข้อมูลแมว

ค. หน้ายืนยันข้อมูลแมว

รูปที่ ค.1 หน้าลงทะเบียนแมว

2. การคำนวณดัชนีมวลกายสำหรับแมว

การคำนวณดัชนีมวลกายสำหรับแมวจะมีการรับข้อมูลความยาวรอบอก (ซม.) และความยาวส่วนขาหลังตั้งแต่หัวเข่าถึงข้อเท้า (ซม.) จากนั้นการผลลัพธ์จะแสดงเป็นรูปภาพและข้อความ ดังรูปที่ ค.2



ก. หน้ากรอกความยาว
รอบอก

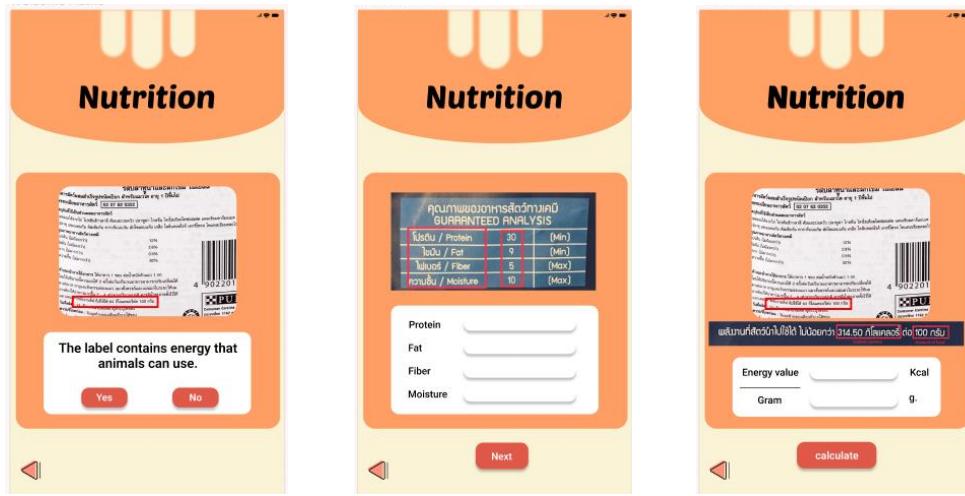
ข. หน้ากรอกความยาวขา
ส่วนหลังตั้งแต่หัวเข่า
ถึงข้อเท้า

ค. ผลลัพธ์ดัชนีมวลกาย
ของแมว

รูปที่ ค.2 หน้าคำนวณดัชนีมวลกายสำหรับแมว

3. การคำนวณโภชนาการของผลิตภัณฑ์

การคำนวณโภชนาการของผลิตภัณฑ์ สามารถเลือกสูตรคำนวณได้ 2 วิธี คือ การคำนวณโดยระบบ
พลังงานที่สัตว์นำ้ไปใช้ได้และการคำนวณโดยไม่ระบุพลังงานที่สัตว์นำ้ไปใช้ได้ ดังรูปที่ ค.3



ก. หน้าเลือกการ
คำนวณโภชนาการ

ข. หน้าคำนวณโดยฉลาก
ระบุพลังงานที่สัตว์
นำไปใช้ได้

ค. หน้าคำนวณโดยฉลาก
ไม่ระบุพลังงานที่สัตว์
นำไปใช้ได้

รูปที่ ค.3 หน้าคำนวณโภชนาการของผลิตภัณฑ์

4. การบันทึกสูตรอาหาร

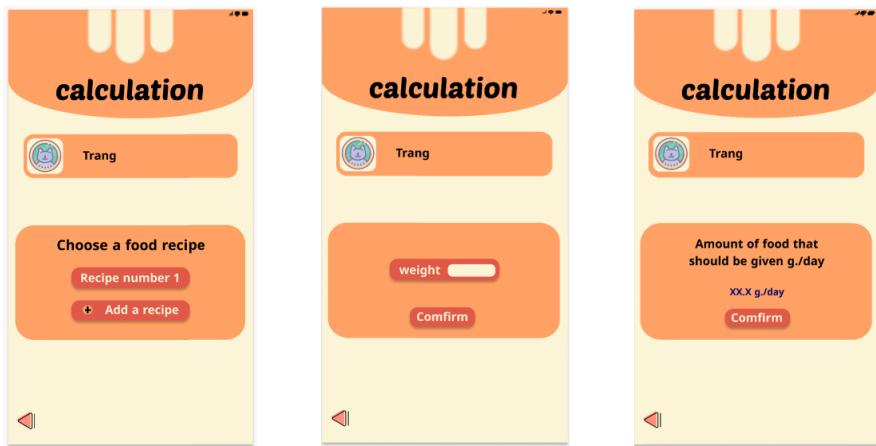
บันทึกเมื่อคำนวณโภชนาการของผลิตภัณฑ์ ดังรูปที่ ค.3 เรียบร้อยแล้ว โดยบันทึกเป็นชื่อสูตรอาหารและโภชนาการ (%) ดังรูปที่ ค.4



รูปที่ ค.4 หน้าบันทึกสูตรอาหาร

5. การคำนวณปริมาณอาหารที่ควรได้รับต่อวัน

การคำนวณปริมาณอาหารที่ควรได้รับต่อวัน (g./day) โดยเลือกสูตรอาหารที่มีอยู่ และกรอกน้ำหนัก จากนั้นคำนวณปริมาณอาหารที่ควรได้รับต่อวัน ดังรูปที่ ค.5



ก. หน้าเลือกสูตรอาหาร

ข. หน้ากรอกน้ำหนัก

ค. ผลลัพธ์ปริมาณอาหาร
ที่ควรได้รับต่อวัน

รูปที่ ค.5 หน้าคำนวณปริมาณอาหารที่ควรได้รับต่อวัน

ភាគធនវក ៤
រដ្ឋសត្ថនជប៊បខែងព្រៃករម

รหัสต้นฉบับหน้าแรกของแอปพลิเคชัน

```

import 'package:cat/Cats/Profile_card.dart';
import 'package:cat/models/Transactions.dart';
import 'package:cat/provider/Transaction_provider.dart';
import 'package:cat/screens/EditTransactionScreen.dart';
import 'package:cat/screens/Food.dart';
import 'package:cat/screens/form_screen.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';

void main() {
    runApp(MyApp());
}

class MyApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return MultiProvider(
            providers: [
                ChangeNotifierProvider(create: (context) =>
                    TransactionProvider(),
            ],
            child: MaterialApp(
                title: 'Cat Nutrition',
                theme: ThemeData(),
                home: HomeScreen(),
            ),
        );
    }
}

class HomeScreen extends StatefulWidget {
    @override
    _HomeScreenState createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
    // สร้างฟังก์ชัน _showGreeting เพื่อแสดง Dialog ของ Instruction
    void _showGreeting() {
}

```

```

showDialog(
    context: context,
    builder: (BuildContext context) {
        return AlertDialog(
            title: Text("Instruction"),
            contentPadding: EdgeInsets.fromLTRB(24, 20, 24, 20), // Add
padding around content
            content: Column(
                mainAxisSize: MainAxisSize.min,
                mainAxisAlignment: MainAxisAlignment.start,
                children: [
                    Text(
                        "Homepage - By clicking on the house icon on the
homepage, You can create account your cat by on click icon add person.
You can access your cat's account and calculate the shape of your
cat."),
                    SizedBox(height: 8),
                    Text(
                        "Food icon page - You can calculate the amount of
food your cat should receive per day (g./day) by entering the
nutritional information of the food formula and selecting the shape of
the cat for calculation. If you do not know the nutritional information
of the food formula, please click the nutritional calculation button
below to calculate the amount of food that should be received per
day."),
                ],
            ),
            actions: <Widget>[
                TextButton(
                    onPressed: () {
                        Navigator.of(context).pop();
                    },
                    child: Text("Close"),
                ),
            ],
        );
    },
);
}

@Override
Widget build(BuildContext context) {

```

```

return DefaultTabController(
    length: 2,
    child: Scaffold(
        appBar: AppBar(
            backgroundColor: Colors.orange,
            automaticallyImpliesLeading: false,
            actions: [
                IconButton(
                    icon: Icon(Icons.info),
                    onPressed: _showGreeting, // ရှိယုံခြုံဖောက်အား _showGreeting မှုပုံမှု
info အောင်
                    iconSize: 35,
                ),
            ],
            flexibleSpace: Container(
                decoration: BoxDecoration(
                    image: DecorationImage(
                        image: NetworkImage(
                            'https://blogger.googleusercontent.com/img/b/R29vZ2
x1/AVvXsEj8oITxuOHh6y2B3yHeqDcS7q7zkDjAFNX8ZGRrwhYhwi7DsGhTMb3lD06a9QxV
Z7LWHXh5uy1V0Nlm3Y9p2WuN1aWCFW787xE8yA3zbdwNZIZqRX7q2I1LDChqTgCx3b1-
DWaKWFH8eb4asGIsCfDOmf12Aonj17GwjpfSgWW65YFtLZdLJRQfxaVfYdgb/s3508/468C
279B-6EDD-4914-97FA-A8E286F73166.png'),
                            fit: BoxFit.cover,
                        ),
                    ),
                ),
            ),
        ),
        body: Container(
            color: Color(0xFFFFAF3D6),
            child: TabBarView(
                children: [
                    Consumer<TransactionProvider>(
                        builder: (context, provider, child) {
                            var transactions = provider.transactions;
                            var count = transactions.length;
                            if (count == 0) {
                                return Center(
                                    child: Text("Please create a cat account."),
                                );
                            } else {

```

```

        return ListView.builder(
            itemCount: count,
            itemBuilder: (context, int index) {
                Transactions data = transactions[index];
                return Card(
                    elevation: 5,
                    margin: const EdgeInsets.symmetric(
                        vertical: 10,
                        horizontal: 30,
                    ),
                    color: Color(0xFFFFAF3D6),
                    child: ListTile(
                        title: Column(
                            crossAxisAlignment:
CrossAxisAlignment.start,
                            children: [
                                if (data.title != null &&
data.title.isNotEmpty)
                                    Text(
                                        '${data.title.toString()}',
                                        style: TextStyle(
                                            color: Colors.purple,
                                            fontSize: 22,
                                            fontWeight: FontWeight.bold,
                                        ),
                                    ),
                                if (data.age != null && data.age != 0)
                                    Text(
                                        'Age: ${data.age.toString()} years',
                                        style: TextStyle(
                                            color: Color.fromARGB(255, 111, 9, 130),
                                        ),
                                    ),
                                if (data.weight != null && data.weight
!= 0.0)
                                    Text(
                                        'Weight: ${data.weight.toString()} kg.',
                                        style: TextStyle(
                                            color: Color.fromARGB(255, 111, 9, 130),
                                        ),
                                    ),
                            ],
                        ),
                    ),
                );
            },
        );
    }
}

```

```

),
if (data.gender != null &&
    data.gender.isNotEmpty)
Text(
    'Gender: ${data.gender}',
    style: TextStyle(
        color: Color.fromARGB(255, 111, 9, 130),
        ),
),
if (data.foodName != null &&
    data.foodName.isNotEmpty)
Text(
    'Food : ${data.foodName}',
    style: TextStyle(
        color: Color.fromARGB(255, 9, 88, 12),
        fontSize: 18,
        fontWeight: FontWeight.bold,
        ),
),
if (data.nutritionValue != null &&
    data.nutritionValue != 0.0)
Text(
    'Nutrition:
${data.nutritionValue.toString()} %',
    style: TextStyle(
        color: Color.fromARGB(255, 9, 88, 12),
        ),
),
],
),
leading: CircleAvatar(
    backgroundColor: (data.title != null &&
        data.age != null &&
        data.age != 0 &&
        data.weight != null &&
        data.weight != 0.0)
    ? Color.fromARGB(255, 216, 122, 250)
    : Colors.green,
    child: Image.network(
        'https://blogger.googleusercontent.com/
img/b/R29vZ2xl/AVvXsEiqGdgHQfHnx_4jqHgtwIHVgTHWeR4W-9-
```

```

AwgOuwDuaGGVZaMVmWmwCa19S8zYZPbf_vHkwVn7lD6jrvnF20fpsYeYGNU_IMHBmC9R9d
CEgf3v9mDUO-YLd_u9PZAotF2biyM1SqKLesLLSrqmhGtqmqYO2Vn7z2LxVMf-H5CNrkdP-
N75hcIwuKtLriaxK/w1684-h1069-p-k-no-nu/2350965C-B25F-4DEB-8FC3-
8F634D5E52CD.png',
width: 40,
height: 40,
),
),
onTap: () {
Navigator.push(context,
MaterialPageRoute(builder: (context)
{
return ProfileCard();
})));
},
trailing: IconButton(
icon: Icon(Icons.edit),
onPressed: () async {
final updatedData = await
Navigator.push(
context,
MaterialPageRoute(
builder: (context) =>
EditTransactionScreen(data: data),
),
);
if (updatedData != null) {
provider.updateTransaction(updatedData);
}
},
),
),
),
);
},
),
);
},
),
),
FoodCat(),
],
),

```

```
),
floatingActionButton: FloatingActionButton(
    onPressed: () {
        Navigator.push(context, MaterialPageRoute(builder:
(context) {
    return FormScreen();
})),
},
backgroundColor: Colors.orange,
elevation: 50,
child: Image.network(
    'https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXs
Eh1YloZbvZi_bcS9V_AAEIHddrJdYRDa5tZlmXKbwJQadvAkwCeEbfNrKrd64C8_8qNXyD7
x_VE1NYUPQBhyphenhyphen0nPsf8qyU8qR5mSyczhlnVrIC-
UZg9kxqdr_6znHyyh_S_ZuKx3tXnRsSF6v1qQ4FnAe3cjN01N39coevp8RVI3YqxpvjhGSv
0qIu_HHTsL/s320/872B90B1-6954-4944-ACFE-296A6F2F7A1D.png',
    width: 50,
    height: 50,
),
),
floatingActionButtonLocation:
FloatingActionButtonLocation.endFloat,
bottomNavigationBar: PreferredSize(
    preferredSize: Size.fromHeight(90),
    child: Container(
        padding: EdgeInsets.only(bottom: 0),
        decoration: BoxDecoration(
            color: Colors.orange,
),
        child: TabBar(
            tabs: [
                Tab(
                    icon: Icon(Icons.home, size: 28, color:
Colors.brown),
),
                Tab(
                    icon: Icon(Icons.restaurant, size: 28, color:
Colors.brown),
),
            ],
),
),
```

) ,
) ,
) ,
) ;
 }

รหัสตั้งฉบับหน้าคำนวณด้ชนีมวลกายสำหรับแมว

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

class _ProfileCardState extends State<ProfileCard> {
  TextEditingController rController = TextEditingController();
  TextEditingController lController = TextEditingController();
  TextEditingController nutritionController = TextEditingController();
  TextEditingController weightController = TextEditingController();
  String bodyShapeResult = '';
  String foodPerDayResult = '';
  String imageUrl = '';
  double nutrition = 0.0;
  double weight = 0.0;

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        backgroundColor: Color(0xFFFFAF3D6),
        appBar: AppBar(
          backgroundColor: Colors.orange,
          title: Text('Shape of Cat'),
          flexibleSpace: Container(
            decoration: BoxDecoration(
              image: DecorationImage(
                image: NetworkImage(

```

```

        'https://blogger.googleusercontent.com/img/b/R29vZ2
x1/AVvXsEgrrNU90ezusab2ylf2CTjVbdlhWoA0UUjXHG5R_M9n8icTyIYoL7ugMx_2DnDp
gZ0JYRk8bwj45n2G4c7mj4DSdEa-
TJORRRVE2anlSoL_oxPXiyllQkelsBrGrHNsE_RytGv5NMyRBz-hKaOI-
CpPsN6MQN9yfWgU5ENLr-Z15MdwkBJUmFE4O7oceiH/w1684-h1069-p-k-no-
nu/A8107C2F-495D-4408-B110-8B4195289C77.png'),
        fit: BoxFit.cover,
    ),
),
),
),
leading: IconButton(
    icon: Icon(Icons.arrow_circle_left_outlined),
    onPressed: () {
        Navigator.pop(context);
    },
    iconSize: 40,
    color: const Color.fromARGB(231, 77, 11, 11),
),
),
body: SingleChildScrollView(
    child: Container(
        padding: EdgeInsets.all(16.0),
        child: Column(
            mainAxisAlignment: MainAxisAlignment.spaceEvenly,
            children: [
                Text(
                    'Calculate Body Shape of Cat',
                    style: TextStyle(
                        fontSize: 20,
                        fontWeight: FontWeight.bold,
                        color: Colors.brown,
                    ),
                    textAlign: TextAlign.center,
                ),
                Image.network(
                    'https://blogger.googleusercontent.com/img/b/R29vZ2x1
/AVvXsEg8T-0piFs8_mOXVPwr0_7rmfS-1rkxE3XBDq8yNvmNlKI7UHgu2kUJ-7v-
w51Sq2LssAi6RnABGhcmSrFa78TCxAxaHpQ3RBHcNTqDj0Jz5fMACrhnm8sczEWEjQ3fR7z
_7g4_Zf4AQWu-vHfbUjwn8-7tWd1vhfJXs-Dm-
minh6knwhdSEwKmad8iRcr7/s3508/8556CEC9-4735-40F5-9355-
0CDDF02D2CD2.gif',

```

```
        height: 250,
        width: 500,
    ),
    buildTextField(controller: rController, label: 'Chest :
(cm.)'),
    SizedBox(height: 16),
    buildTextField(controller: lController, label: 'Leg :
(cm.)'),
    SizedBox(height: 16),
    ElevatedButton(
        onPressed: () {
            calculate();
        },
        style: ButtonStyle(
            backgroundColor:
MaterialStateProperty.all(Colors.orange),
            foregroundColor:
MaterialStateProperty.all(Colors.black),
            padding: MaterialStateProperty.all(
                EdgeInsets.symmetric(horizontal: 100, vertical:
20)),
        ),
        child: Text('Calculate'),
    ),
    SizedBox(height: 16),
    if (imageUrl.isNotEmpty)
        Image.network(
            imageUrl,
            height: 300,
            width: 300,
        ),
    ],
),
),
),
),
),
);
}

Widget buildTextField({
    required TextEditingController controller,
```

```

        required String label,
    }) {
    return TextField(
        controller: controller,
        keyboardType: TextInputType.number,
        decoration: InputDecoration(labelText: label),
    );
}

void calculate() {
try {
    double r = double.parse(rController.text);
    double l = double.parse(lController.text);

    double calculatedResult = ((r / 0.7062) - 1) / 0.9156 - 1;

    setState(() {
        bodyShapeResult = calculatedResult.toString();
        setProfileImage(calculatedResult);
    });
} catch (e) {
    setState(() {
        imageUrl = '';
    });
}
}

void setProfileImage(double calculatedResult) {
if (calculatedResult < 15) {
    imageUrl =
        'https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEh
jOrp_BLVIhLMH9TPWEAqqUagJjPDAFeDyE6Q7jad-
ONzgvlgHXb01tXiwwoj_d4sQugeNzyxirDlRzA_OIgKrY_IpjeNPB-
i18cUgaZuJzar_oEdqsXMYbgA23u_KNZmj3R4sdRDbG2Utzt9d1Ui6z1UvFX4ussGe1ZCx
MLZkb1-JWaztta2N5QPAnB/s320/2DEE3F7F-E831-4A5B-8206-AF123CCEADF4.png';
} else if (calculatedResult >= 15 && calculatedResult < 29.9) {
    imageUrl =
        'https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEg
CL0gROofHkJM-H4eh-
8JD4rBy0SwXE8S5bMfCYuZm0Tfu9f25QhzixSdpDhJa_o2gmabwqIdVTNi9-
N6zHAnd7R62Kpo6HUqqtYFOujZMA6yYzjnZ3i8qo2XBVNe3CgvWVs
f23uEvCdMqcDvB5aoZ

```

```
hu8RIYUwmqI7KYINwuepuOsZCj-jD3xsuYVmYp1/s320/14176E22-0B03-449A-A434-  
030A8DA7A060.png';  
    } else if (calculatedResult >= 30) {  
        imageUrl =  
            'https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEh  
VLbS19049imHbvXUtlXfNeLNv9hHB_QMyK1d51ATxbvBmkh-  
dwqXZivY1qe9tUFsQhs1TGo7j2G09w7P6ptCL1fHIemqkenCNozPMsOzx76YIyd6V7NIBC-  
v_r9HGNWmXz1VcBhJn8VN3GGVHsazqketkga0PI50_9RUSHbZYqoGsEg96UiQmT1YUGP9X/  
s320/2786AE1D-0A3F-4F6D-A8DB-FDCCCB7C47B7.png';  
    } else {  
        imageUrl = '';  
    }  
}
```

รหัสต้นฉบับลงทะเบียนบัญชีแมว

```
import 'package:flutter/material.dart';
import 'package:cat/models/Transactions.dart';
import 'package:cat/provider/Transaction_provider.dart';
import 'package:provider/provider.dart';

class FormScreen extends StatefulWidget {
  @override
  _FormScreenState createState() => _FormScreenState();
}

class _FormScreenState extends State<FormScreen> {
  final GlobalKey<FormState> formKey = GlobalKey<FormState>();

  final TextEditingController titleController = TextEditingController();
  final TextEditingController weightController = TextEditingController();
  DateTime? selectedDate;
  String? selectedGender;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
```

```

    backgroundColor: Colors.orange,
    title: Text('Register Cat'),
    flexibleSpace: Container(
      decoration: BoxDecoration(
        image: DecorationImage(
          image: NetworkImage(
            'https://blogger.googleusercontent.com/img/b/R29vZ2xl
/AVvXsEhvmTybHR9ftRzBGk4U_7p0ZH6Nx384pSukrGk2V_9kTMxigTfT0mvXU9vlBsjuCY
6T6aQadWPfzSVevflkLAuaHh0JxgO9sNSAV19exEzp4znMMBAwmVsVu_RhMiVWTkUXWsjhy
_rJDDKCAmgAI3iFQitzkiwIstJ0cDSz7OFFXSWTOAZQrKr24_kwIsJk/w1684-h1069-p-
k-no-nu/5807C3FC-07B2-4005-A56F-5D2A9B8986EB.png'),
            fit: BoxFit.cover,
          ),
        ),
      ),
    ),
  ),
  body: Padding(
    padding: const EdgeInsets.all(30.0),
    child: Form(
      key: formKey,
      child: Column(
        children: [
          TextFormField(
            decoration: InputDecoration(labelText: "Cat's name"),
            autofocus: true,
            controller: titleController,
            validator: (String? str) {
              if (str == null || str.isEmpty) {
                return "Please enter your cat's name.";
              }
              return null;
            },
          ),
          TextFormField(
            decoration: InputDecoration(labelText: "Weight (kg.)"),
            controller: weightController,
            keyboardType: TextInputType.number,
            validator: (String? str) {
              if (str == null || str.isEmpty) {
                return "Please enter your cat's weight.";
              }
            }
          )
        ],
      ),
    ),
  ),

```

```
        return null;
    },
),
SizedBox(
    height: 10),
DropdownButtonFormField<String>(
    decoration: InputDecoration(labelText: 'Gender'),
    value: selectedGender,
    items: ['male', 'female', 'other']
        .map((gender) => DropdownMenuItem(
            child: Text(gender),
            value: gender,
        ))
        .toList(),
onChanged: (String? value) {
    setState(() {
        selectedGender = value;
    });
},
validator: (value) {
    if (value == null || value.isEmpty) {
        return 'Please select your cat\'s gender.';
    }
    return null;
},
),
SizedBox(height: 15),
ElevatedButton(
    style: ButtonStyle(
        backgroundColor: MaterialStateProperty.all(
            Color.fromARGB(255, 244, 215, 171)),
    ),
    onPressed: () {
        showDatePicker(
            context: context,
            initialDate: DateTime.now(),
            firstDate: DateTime(2000),
            lastDate: DateTime.now(),
            builder: (BuildContext context, Widget? child) {
                return Theme(
                    data: ThemeData.light().copyWith(

```

```

        colorScheme: ColorScheme.light(
            primary: Colors.orange,
        ),
        ),
        child: child!,
    );
},
).then((pickedDate) {
    if (pickedDate != null) {
        setState(() {
            selectedDate = pickedDate;
        });
    }
});
},
child: Row(
    children: [
        Text(
            selectedDate == null
                ? 'Birthdate'
                : 'Birthdate:
${selectedDate.toString().substring(0, 10)}',
            style: TextStyle(
                fontSize: 18,
                color: Colors.brown),
        ),
        SizedBox(width: 15),
        Icon(Icons.calendar_today,
            color: Colors.brown),
    ],
),
),
),
),

SizedBox(
    height: 10),
ElevatedButton(
    onPressed: () {
        if (formKey.currentState?.validate() ?? false) {
            var title = titleController.text;
            var weight = double.parse(weightController.text);
            var age = DateTime.now().year -

```


รหัสต้นฉบับหน้าบันทึกสูตรอาหาร

```

import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:cat/provider/Transaction_provider.dart';
import 'package:cat/models/Transactions.dart';

class FormFood extends StatefulWidget {
  @override
  _FormFoodState createState() => _FormFoodState();
}

class _FormFoodState extends State<FormFood> {
  final GlobalKey<FormState> formKey = GlobalKey<FormState>();

  final foodNameController = TextEditingController();
  final nutritionValueController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.orange,
        title: Text('Register Food Recipe'),
        flexibleSpace: Container(
          decoration: BoxDecoration(
            image: DecorationImage(
              image: NetworkImage(
                'https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEhvmTybHR9ftRzBGk4U_7p0ZH6Nx384pSukrGk2V_9kTMxigTft0mvXU9v1BsjuCY6T6aQadWPfzSVevflkLAuaHh0JxgO9sNSAV19exEzp4znMMBAwmVsVu_RhMiVWTkUXWsjh_yrJDDKCAmgAI3iFQitzkiwIstJ0cDSz7OFFXSWTOAZQrKr24_kwIsJk/w1684-h1069-p-k-no-nu/5807C3FC-07B2-4005-A56F-5D2A9B8986EB.png'),
              fit: BoxFit.cover,
            ),
          ),
        ),
      ),
      body: Padding(

```

```
padding: const EdgeInsets.all(30.0),
child: Form(
  key: formKey,
  child: Column(
    children: [
      TextFormField(
        decoration: InputDecoration(labelText: "Food's name"),
        autofocus: true,
        controller: foodNameController,
        validator: (String? str) {
          if (str == null || str.isEmpty) {
            return "Please enter your Food's name.";
          }
          return null;
        },
      ),
      TextFormField(
        decoration: InputDecoration(labelText: "Nutrition (%)"),
        controller: nutritionValueController,
        keyboardType: TextInputType.number,
        validator: (String? str) {
          if (str == null || str.isEmpty) {
            return "Please enter Nutrition.";
          }
          return null;
        },
      ),
      SizedBox(
        height: 10,
      ),
      ElevatedButton(
        onPressed: () {
          if (formKey.currentState?.validate() ?? false) {
            var foodName = foodNameController.text;
            var nutritionValue =
                double.parse(nutritionValueController.text);

            Transactions statement = Transactions(
              foodName: foodName,
              nutritionValue: nutritionValue,
            );
          }
        },
      ),
    ],
  ),
);
```

ຮັບສ້າງລະບົບໜ້າຄໍານວນສູຕາອາຫານ

```
import 'package:cat/Foods/GA_foods.dart';
import 'package:cat/Foods/ME_foods.dart';
import 'package:flutter/material.dart';

class CalFood extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
```

```

appBar: AppBar(
  backgroundColor: Colors.orange,
  title: Text(
    '',
    style: TextStyle(fontSize: 20),
  ),
  flexibleSpace: Container(
    decoration: BoxDecoration(
      image: DecorationImage(
        image: NetworkImage(
          'https://blogger.googleusercontent.com/img/b/R29vZ2xl
/AVvXsEjgUWDrFQjC7zyEl2D0q-Ng-
RDYhbhjRPN8PYogvNpzermj3JsCzTczFryvgx022boce8G1E7XVPXZlFLPIqdXPPSqxU_pt
DX8RWFSSwiMaOo_77jByMRYIwi44ofA2g1f17p08Gd6sw8p6pQnrSVxore3q37q3Zoz2_51
IjumOZCCyrrnYN5J2zM8uI-uYQ/w1684-h1069-p-k-no-nu/75131769-6C7A-4E00-
96B7-A489E87A33B5.png'),
        fit: BoxFit.cover,
      ),
    ),
  ),
),
body: SingleChildScrollView(
  child: Container(
    padding: EdgeInsets.all(16),
    color: Color(0xFFFFAF3D6), // Set background color to orange
    child: Column(
      children: [
        Text(
          'Calculate Nutrition of Foods',
          style: TextStyle(
            fontSize: 24,
            fontWeight: FontWeight.bold,
            color: Colors.brown, // Set text color to dark brown
          ),
        ),
        SizedBox(height: 10),
        Text(
          "Please check the food label if it has metabolizable
energy information like the one in the green box.",
          style: TextStyle(
            color: Colors.brown, // Set text color to dark brown
          ),
        )
      ],
    ),
  ),
)

```

```

),
),
SizedBox(height: 30),
Image.network(
  'https://blogger.googleusercontent.com/img/b/R29vZ2xl/A
VvXsEggx8OorqUsZrH-G4cwJ-mc19_Hzv381EcP7cwuy-qZWP-
hgvv1nq8wooGCbGwYRQhzGOT1bTjZ7lhotnV7yPAvKnMMKm_P1YRogu-omOkrNj2_O6Cr-
TqLgwdnpXqibrcrz_lIwaSA_V_02-8D5ZiCX65p18hncnwGVRsH0NP5-
CfCn83IkOSL72aY-ntY/w1684-h1069-p-k-no-
nu/%E0%B8%A3%E0%B8%B9%E0%B8%9B1.jpg',
  height: 300,
  width: 500,
),
SizedBox(height: 10),
Text(
  "In the case there's metabolizable energy information
shown on the label, use calculation ME.",
  style: TextStyle(
    color: Colors.brown, // Set text color to dark brown
  ),
),
SizedBox(height: 10),
ElevatedButton(
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => MEfoods()),
    );
  },
  child: Text(
    'Calculation ME',
    style: TextStyle(
      color: const Color.fromARGB(255, 255, 254, 254), // Set text color to black
    ),
  ),
  style: ButtonStyle(
    backgroundColor: MaterialStateProperty.all<Color>(
      Color.fromARGB(255, 147, 79, 2), // Set background
color to orange
    ),
  ),
)
),

```

```

),
),
SizedBox(height: 40),
Image.network(
  'https://blogger.googleusercontent.com/img/b/R29vZ2xl/A
VvXsEhQnJsQ01IuPElTWVFwziV7AjTXw8bvMamwr6NTMCxsXW816GD5Rpr9RdvLtw1eurTt
VMTyR4i7-qbEMf88R669BcMkr9JjKFKRPpV0yfE_3SWM4xYcO80yG2Mex8g_y4h1nOj-
VbC_Re3Lt-sUV0stcY_1JYt9sOYdm7UaNaMvF8rTTSzFlRLfNmRx20Nt/w1684-h1069-p-
k-no-nu/%E0%B8%A3%E0%B8%B9%E0%B8%9B2.jpg',
  height: 300,
  width: 500,
),
SizedBox(height: 10),
Text(
  "In the case the label does not specify this
information, look for the Guaranteed analysis information like the one
in the red box to use in calculation GA.",
  style: TextStyle(
    color: Colors.brown, // Set text color to dark brown
  ),
),
SizedBox(height: 10),
ElevatedButton(
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => GAfoods()),
    );
  },
  child: Text(
    'Calculation GA',
    style: TextStyle(
      color: const Color.fromARGB(255, 255, 254, 254), // Set text color to black
    ),
  ),
  style: ButtonStyle(
    backgroundColor: MaterialStateProperty.all<Color>(
      Color.fromARGB(255, 147, 79, 2), // Set background
color to orange
    ),
  ),
)
),

```

```
        ),
        ],
),
),
),
),
);
}
}

import 'package:flutter/material.dart';

class MEfoods extends StatefulWidget {
@override
_MEfoodsState createState() => _MEfoodsState();
}

class _MEfoodsState extends State<MEfoods> {
TextEditingController kcalController = TextEditingController();
TextEditingController feedController = TextEditingController();
String result = '';

@Override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
backgroundColor: Colors.orange,
title: Text('Calculation ME'),
// Place the Container with DecorationImage here
flexibleSpace: Container(
decoration: BoxDecoration(
image: DecorationImage(
image: NetworkImage(
'https://blogger.googleusercontent.com/img/b/R29vZ2xl
/AVvXsEjgUWDrFQjC7zyE12D0q-Ng-
RDYbhbjRPN8PYogvNppermj3JsCzTczFryvgx022boce8G1E7XVPXZlFLPIqdXPPSqxU_pt
DX8RWFSswiMaOo_77jByMRYIwi44ofA2g1f17p08Gd6sw8p6pQnrSVxore3q37q3Zoz2_51
IjumOZCCyrrnYN5J2zM8uI-uYQ/w1684-h1069-p-k-no-nu/75131769-6C7A-4E00-
96B7-A489E87A33B5.png'),
fit: BoxFit.cover,
),

```

```

        ),
        ),
        ),
        body: SingleChildScrollView(
            child: Container(
                padding: EdgeInsets.all(16),
                child: Column(
                    children: [
                        Image.network(
                            'https://blogger.googleusercontent.com/img/b/R29vZ2x1/A
VvXsEgMJppDD_eJQJlTWvx1qVhDQ_S-
byE_6xXVMiSvAU6kctv7s1JHqAkzKTxSy1Eg0DVIABXPWTJ1O12v9nAhKxfBdQTeYXCEnHe
pAhRe7UG8ylBMcDICRsc6pEba09SERCLpOxa4LCWXf-
xwIolXhQP2uqFB3NmH_U1FUC4O8zCw-
72fOAv4v9ZwjZKzLft/s320/%E0%B8%A3%E0%B8%B9%E0%B8%9B4.jpg',
                            height: 300,
                            width: 500,
                        ),
                        Text(
                            "Please fill in the information by looking at the cat
food label."),
                    ],
                ),
                // Text inputs and calculation button
                buildNumberTextField(
                    labelText: 'Kcal (Kcal.)',
                    controller: kcalController,
                ),
                buildNumberTextField(
                    labelText: 'Feed (g.)',
                    controller: feedController,
                ),
                SizedBox(height: 20),
                ElevatedButton(
                    onPressed: () {
                        if (kcalController.text.isEmpty ||
                            feedController.text.isEmpty) {
                            // Show error message if any field is empty
                            ScaffoldMessenger.of(context).showSnackBar(
                                SnackBar(
                                    content: Text('Please enter the amount of

```

```

nutrients.'),
            duration: Duration(seconds: 2),
        ),
    );
} else {
    // Perform calculation if all fields are filled
    setState(() {
        double kcal = double.parse(kcalController.text);
        double feed = double.parse(feedController.text);

        if (feed == 0.0) {
            // Show error message if feed is zero
            ScaffoldMessenger.of(context).showSnackBar(
                SnackBar(
                    content:
                        Text('Feed value must be greater than
zero.'),
                    duration: Duration(seconds: 2),
                ),
            );
        } else {
            result = (kcal / feed).toStringAsFixed(2);
        }
    });
}
},
child: Text(
    'Calculate',
    style: TextStyle(
        color: Colors.white, // กำหนดสีของตัวอักษรเป็นสีขาว
    ),
),
style: ButtonStyle(
    backgroundColor:
        MaterialStateProperty.all(Color.fromARGB(255,
147, 79, 2)),
    padding:
MaterialStateProperty.all(EdgeInsets.symmetric(
        horizontal: 150, vertical: 20)), // ปรับขนาดของปุ่ม
),
),

```



```

@Override
_GAfoodsState createState() => _GAfoodsState();
}

class _GAfoodsState extends State<GAfoods> {
double proteinValue = 0.0;
double fatValue = 0.0;
double fiberValue = 0.0;
double moistureValue = 0.0;
double result = 0.0;

@Override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
backgroundColor: Colors.orange,
title: Text('Calculation GA'),
// Place the Container with DecorationImage here
flexibleSpace: Container(
decoration: BoxDecoration(
image: DecorationImage(
image: NetworkImage(
'https://blogger.googleusercontent.com/img/b/R29vZ2xl
/AVvXsEjgUWDrFQjC7zyEl2D0q-Ng-
RDYbhbjRPN8PYogvNpzermj3JsCzTczFryvgx022boce8G1E7XVPXZlFLPIqdXPPSqxU_pt
DX8RWFSSwiMaOo_77jByMRYIwi44ofA2g1f17p08Gd6sw8p6pQnrSVxore3q37q3Zoz2_51
IjumOZCCyrrnYN5J2zM8uI-uYQ/w1684-h1069-p-k-no-nu/75131769-6C7A-4E00-
96B7-A489E87A33B5.png'),
fit: BoxFit.cover,
),
),
),
),
),
body: SingleChildScrollView(
child: Padding(
padding: const EdgeInsets.all(16.0),
child: Column(
crossAxisAlignment: CrossAxisAlignment.stretch,
children: [
Image.network(
'https://blogger.googleusercontent.com/img/b/R29vZ2xl/A

```

```
VvXsEjvptEGt9a-UHSWhtpPqbZDrf7qA_HHVJxOV8p74hND-
Djo0D4fgWkp6k6fI3CU3q1Za1bOJAC_IxOPv1AhpJNA7XxYV_2VOuKRcnDk01qyFotFSfxW
sBvfOAdf7i3A9f4hyB5RxqYJgCf2LVpDj_OqhuJsUSi9IPcZSL47XQzDsbJBYaUuc8c1aNI
DjWTB/s320/%E0%B8%A3%E0%B8%B9%E0%B8%9B3.jpg',
    height: 250,
    width: 500,
),
SizedBox(height: 20),
buildNumberTextField(
    labelText: 'Protein (% min) :',
    onChanged: (value) {
        setState(() {
            proteinValue = double.tryParse(value) ?? 0.0;
        });
    },
),
buildNumberTextField(
    labelText: 'Fat (% min) :',
    onChanged: (value) {
        setState(() {
            fatValue = double.tryParse(value) ?? 0.0;
        });
    },
),
buildNumberTextField(
    labelText: 'Fiber (% max) :',
    onChanged: (value) {
        setState(() {
            fiberValue = double.tryParse(value) ?? 0.0;
        });
    },
),
buildNumberTextField(
    labelText: 'Moisture (% max) :',
    onChanged: (value) {
        setState(() {
            moistureValue = double.tryParse(value) ?? 0.0;
        });
    },
),
SizedBox(height: 20),
```

```

ElevatedButton(
    onPressed: () {
        if (proteinValue == 0.0 ||
            fatValue == 0.0 ||
            fiberValue == 0.0 ||
            moistureValue == 0.0) {
            // Show error message if any field is empty
            ScaffoldMessenger.of(context).showSnackBar(
                SnackBar(
                    content: Text('Please enter the amount of
nutrients.'),
                    duration: Duration(seconds: 2),
                ),
            );
        } else {
            // Perform calculation if all fields are filled
            setState(() {
                result = ((proteinValue * 3.5) +
                    (fatValue * 8.5) +
                    ((100 -
                        proteinValue -
                        fatValue -
                        fiberValue -
                        moistureValue) *
                    3.5)) /
                100;
            });
        }
    },
    child: Text(
        'Calculate',
        style: TextStyle(
            color: Colors.white, // ការណែនាំខ្លួនគឺមិនតើម្ខាន
        ),
    ),
    style: ButtonStyle(
        backgroundColor:
            MaterialStateProperty.all(Color.fromARGB(255,
147, 79, 2)),
        padding:
            MaterialStateProperty.all(EdgeInsets.symmetric(

```

```

        horizontal: 10, vertical: 20)), // ปรับขนาดของปุ่ม
    ),
),
SizedBox(height: 20),
Container(
color:
Color.fromARGB(255, 235, 221, 255), // กำหนดสีพื้นหลังให้กับ
Container เมื่อคลิก
padding: EdgeInsets.all(16), // เพิ่มระยะห่างภายใน Container
child: Center(
child: Text(
proteinValue == 0.0 ||
fatValue == 0.0 ||
fiberValue == 0.0 ||
moistureValue == 0.0
? 'Nutrition of food recipes:      %'
: 'Nutrition of food recipes:      $result %',
style: TextStyle(fontSize: 18 ,color:
Color.fromARGB(255, 41, 2, 159),),
),
),
),
],
),
),
),
// Set background color of the screen
),
backgroundColor: Color(0xFFFFAF3D6),
);
}

Widget buildNumberTextField({
required String labelText,
required void Function(String) onChanged,
}) {
return TextField(
keyboardType: TextInputType.number,
decoration: InputDecoration(labelText: labelText),
onChanged: onChanged,
);
}
}

```

รหัสต้นฉบับหน้าคำนวณปริมาณอาหารที่ควรได้รับต่อวัน

```

import 'package:cat/screens/cal_food.dart';
import 'package:cat/screens/form_food.dart';
import 'package:flutter/material.dart';
import 'dart:math';
void main() {
  runApp(MaterialApp(
    home: FoodCat(),
  ));
}
class FoodCat extends StatefulWidget {
  @override
  _FoodCatState createState() => _FoodCatState();
}
class _FoodCatState extends State<FoodCat> {
  TextEditingController weightController = TextEditingController();
  TextEditingController customNutritionController =
  TextEditingController();

  String foodPerDayResult = '';
  Map<String, dynamic>? selectedFood;
  List<Map<String, dynamic>> foodList = [
    {"name": "PURINA ONE : 1+ years", "nutritionValue": 3.61},
    {"name": "Silver : Indoor", "nutritionValue": 3.0925},
    {"name": "Cat's taste : in gravy", "nutritionValue": 0.45},
    {"name": "Cat's taste : Senior 7+", "nutritionValue": 0.39},
    // Add more food items as needed
  ];
  @override
  Widget build(BuildContext context) {
    return SingleChildScrollView(
      child: Container(
        padding: EdgeInsets.all(16),
        color: Color(0xFFFAF3D6),
        child: Column(
          children: [
            SizedBox(height: 16),
            Text(
              'How much food (g./day) for your cat?',
            ),

```

```

        style: TextStyle(
            fontSize: 20,
            fontWeight: FontWeight.bold,
            color: Colors.brown,
        ),
        textAlign: TextAlign.center,
    ),
    DropdownButtonFormField<Map<String, dynamic>>(
value: selectedFood,
onChanged: (value) {
    if (customNutritionController.text.trim().isEmpty) {
        setState(() {
            selectedFood = value!;
        });
    } else {
        showDialog(
            context: context,
            builder: (context) => AlertDialog(
                title: Text('Warning'),
                content: Text('Please clear the custom nutrition field before
selecting a food option.'),
                actions: [
                    TextButton(
                        onPressed: () {
                            Navigator.of(context).pop();
                        },
                        child: Text('OK'),
                    ),
                ],
            ),
        );
    }
},
items: foodList.map((food) {
    return DropdownMenuItem<Map<String, dynamic>>(
        value: food,
        child: Text(food['name'] as String),
    );
}).toList(),
decoration: InputDecoration(labelText: 'Select Food:'),
),

```

```

SizedBox(height: 16),
TextField(
    controller: customNutritionController,
    keyboardType: TextInputType.number,
    enabled: selectedFood ==
        null, // ត្រឡប់ពី selectedFood មិន null នៅពេល
    decoration: InputDecoration(
        labelText: "Please fill in the nutrition (%) here :",
        helperText:
            "If the recipe you're looking for isn't available
in options.",
        contentPadding:
            EdgeInsets.only(bottom: 8), // Adjust padding here
    ),
),

buildTextField(
    controller: weightController,
    label: 'Weight of Cat (kg.):',
),
SizedBox(height: 16),
Row(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: [
        ElevatedButton(
            onPressed: () {
                calculateFoodPerDay(isFat: true);
            },
            style: ElevatedButton.styleFrom().copyWith(
                backgroundColor: MaterialStateProperty.all(
                    Color.fromARGB(255, 147, 79, 2)),
            ),
            child: Text(
                'Fat ',
                style: TextStyle(
                    color: Color.fromARGB(255, 252, 252, 252),
                ),
            ),
        ),
        ElevatedButton(

```

```

        onPressed: () {
            calculateFoodPerDay(isFat: false);
        },
        style: ElevatedButton.styleFrom().copyWith(
            backgroundColor: MaterialStateProperty.all(
                Color.fromARGB(255, 147, 79, 2)),
        ),
        child: Text(
            'Perfect or Thin',
            style: TextStyle(
                color: const Color.fromARGB(255, 254, 254, 254),
            ),
        ),
    ),
],
),
SizedBox(height: 16),
Text(
    "If you don't know shape of cat, you can calculate when
you cilck on profile cat.",
    style: TextStyle(
        fontSize: 14,
        color: Colors.brown,
    ),
    textAlign: TextAlign.center,
),
SizedBox(height: 16),
Container(
    decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(10),
        color: Color.fromARGB(255, 235, 221, 255),
    ),
    padding: EdgeInsets.all(16),
    child: Text(
        'Food per day for cat
is      $foodPerDayResult      g./day',
        style: TextStyle(
            color: Color.fromARGB(255, 41, 2, 159), fontSize:
18),
    ),
),

```

```
SizedBox(height: 80), // Added space

Container(
  decoration: BoxDecoration(
    border: Border.all(color: Colors.black),
    borderRadius: BorderRadius.circular(10),
  ),
  padding: EdgeInsets.all(16),
  child: Column(
    children: [
      Text(
        'If you don`t know the nutrition of food recipes.',
        style: TextStyle(
          fontSize: 16,
          color: Colors.black,
        ),
      ),
      SizedBox(height: 12),
      ElevatedButton(
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) =>
CalFood()),
        );
      },
      child: Text(
        'Calculate Nutrition',
        style: TextStyle(
          color: const Color.fromARGB(255, 255, 255,
255),
        ),
      ),
      style: ElevatedButton.styleFrom().copyWith(
        backgroundColor: MaterialStateProperty.all(
          Color.fromARGB(255, 147, 79, 2)),
      ),
    ),
    SizedBox(height: 3),
    Text(
      'To calculate, please click on button.',
    ),
  ],
)
```

```
        style: TextStyle(
            fontSize: 14,
            color: Colors.black,
        ),
    ),
    SizedBox(height: 24),
    Text(
        'Save food recipe, please click on button.',
        style: TextStyle(
            fontSize: 16,
            color: Colors.black,
        ),
    ),
    SizedBox(height: 10),
    ElevatedButton(
        onPressed: () {
            Navigator.push(
                context,
                MaterialPageRoute(builder: (context) =>
FormFood(),
            );
        },
        child: Text(
            'Save Food Recipe',
            style: TextStyle(
                color: Colors.white,
            ),
        ),
        style: ElevatedButton.styleFrom().copyWith(
            backgroundColor: MaterialStateProperty.all(
                Color.fromARGB(255, 147, 79, 2)),
        ),
    ),
    SizedBox(height: 3),
    Text(
        'To save, please click on button.',
        style: TextStyle(
            fontSize: 14,
            color: Colors.black,
        ),
    ),
),
```

```

        ],
        ),
        ),
        ],
        ),
        ),
        );
    }
}

Widget buildTextField({
    required TextEditingController controller,
    required String label,
}) {
    return TextField(
        controller: controller,
        keyboardType: TextInputType.number,
        decoration: InputDecoration(labelText: label),
    );
}

void calculateFoodPerDay({required bool isFat}) {
    try {
        double weight = double.parse(weightController.text);
        double nutritionValue = 0.0;

        if (selectedFood != null && selectedFood!['nutritionValue'] != null) {
            nutritionValue = selectedFood!['nutritionValue'] as double;
        } else {
            nutritionValue = double.parse(customNutritionController.text);
        }

        double factor = isFat ? 1.2 : 1.4;
        double foodPerDay = factor * (70 * pow(weight, 0.75)) /
nutritionValue;
        setState(() {
            foodPerDayResult = foodPerDay.toStringAsFixed(2);
        });
    } catch (e) {
        setState(() {
            foodPerDayResult = 'Invalid input';
        });
    }
}
}

```

รหัสต้นฉบับการจัดเก็บฐานข้อมูล

```
class Transactions {  
    late int id;  
    String title;  
    String gender; // Changed from late to normal declaration  
    String foodName;  
    double nutritionValue;  
    int age;  
    double weight;  
    String fd;  
    double nutri;  
  
    Transactions(){  
        this.id = 0,  
        this.title = '',  
        this.gender = '', // Set default value to an empty string  
        this.foodName = '',  
        this.age = 0,  
        this.nutritionValue = 0.0,  
        this.fd = '',  
        this.nutri = 0.0,  
        double? weight,  
    }) : weight = weight != null ? weight : 0.0;  
  
    Map<String, dynamic> toMap() {  
        return {  
            'id': id,  
            'title': title,  
            'gender': gender,  
            'foodName': foodName,  
            'nutritionValue': nutritionValue,  
            'age': age,  
            'weight': weight,  
            'fd': fd,  
            'nutri': nutri,  
        };  
    }  
}
```

```

class TransactionProvider with ChangeNotifier {
  List<Transactions> transactions = [];

  List<Transactions> getTransactions() {
    return transactions;
  }

  void initData() async {
    var db = TransactionDB(dbName: "transactions.db");
    transactions = await db.loadAllData();
    notifyListeners();
  }

  Future<void> addTransaction(Transactions statement) async {
    var db = TransactionDB(dbName: "transactions.db");
    await db.insertData(statement); // սնված insertData սու InsertData
    transactions = await db.loadAllData();
    notifyListeners();
  }

  void updateTransaction(Transactions updatedData) async {
    var db = TransactionDB(dbName: "transactions.db");
    await db.updateTransaction(updatedData);
    transactions = await db.loadAllData();
    notifyListeners();
  }

  Future<void> removeTransaction(String id) async {
    var db = TransactionDB(dbName: "transactions.db");
    await db.deleteData(id); // սնված deleteData սու deleteData
    transactions.removeWhere((transaction) => transaction.id == id);
    notifyListeners();
  }
}

import 'dart:io';
import 'package:cat/models/Transactions.dart';
import 'package:path/path.dart';
import 'package:path_provider/path_provider.dart';
import 'package:sembast/sembast.dart';
import 'package:sembast/sembast_io.dart';
class TransactionDB {
  late String dbName;
}

```

```

    TransactionDB({required this.dbName});
    Future<Database> openDatabase() async {
        Directory appDirectory = await getApplicationDocumentsDirectory();
        String dbLocation = join(appDirectory.path, dbName);
        DatabaseFactory dbFactory = await databaseFactoryIo;
        Database db = await dbFactory.openDatabase(dbLocation);
        return db;
    }

    Future<int> insertData(Transactions statement) async {
        var db = await this.openDatabase();
        var store = intMapStoreFactory.store("expense");

        var keyID = store.add(db, {
            "title": statement.title,
            "weight": statement.weight,
            "age": statement.age,
            "foodName": statement.foodName,
            "nutritionValue": statement.nutritionValue,
            "gender": statement.gender,
            "fd": statement.fd, // Use "food" field
            "nutri": statement.nutri, // Use "nutrition" field
        });
        await db.close();
        return keyID;
    }

    Future<List<Transactions>> loadAllData() async {
        var db = await this.openDatabase();
        var store = intMapStoreFactory.store("expense");
        var snapshot = await store.find(db);
        List<Transactions> transactionList = [];
        for (var record in snapshot) {
            if (record["weight"] != null &&
                record["nutritionValue"] != null &&
                record["gender"] != null &&
                record["age"] != null &&
                record["foodName"] != null &&
                record["fd"] != null &&
                record["nutri"] != null
            ) {

```

```

        int? age = record["age"] as int?;
        transactionList.add(
            Transactions(
                id: record.key,
                title: record["title"] as String,
                weight: record["weight"] as double,
                age: age ?? 0,
                foodName: record["foodName"] as String,
                nutritionValue: record["nutritionValue"] as double,
                gender: record["gender"] as String,
                fd: record["fd"] as String,
                nutri: record["nutri"] as double,
            ),
        );
    }
}

await db.close();
return transactionList;
}

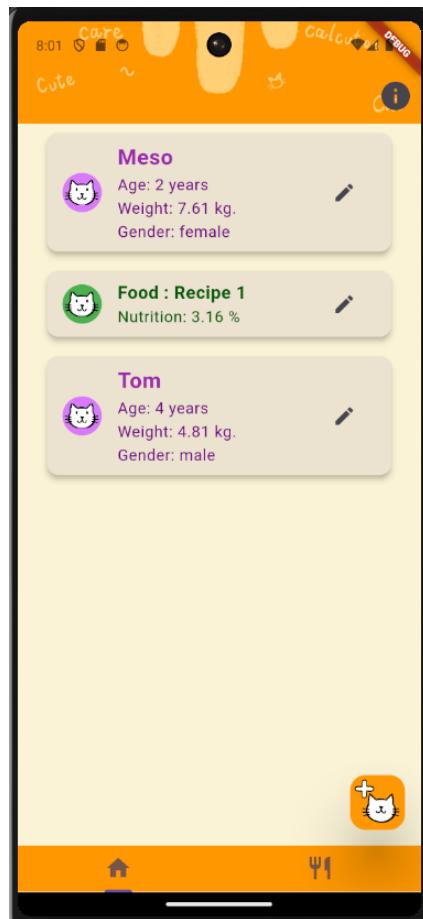
Future<void> updateTransaction(Transactions updatedData) async {
    var db = await this.openDatabase();
    var store = intMapStoreFactory.store("expense");
    await store.record(updatedData.id).update(db, {
        "title": updatedData.title,
        "age": updatedData.age,
        "weight": updatedData.weight,
        "foodName": updatedData.foodName,
        "nutritionValue": updatedData.nutritionValue,
        "gender": updatedData.gender,
        "fd": updatedData.fd, // Use "food" field
        "nutri": updatedData.nutri, // Use "nutrition" field
    });
    await db.close();
}

Future<void> deleteData(String id) async {
    var db = await this.openDatabase();
    var store = intMapStoreFactory.store("expense");
    await store.record(int.parse(id)).delete(db); // Parse id to int
    await db.close();
}
}

```

ภาคผนวก จ
คู่มือการใช้งาน

คู่มือการใช้งาน แอปพลิเคชันคำนวณปริมาณอาหารและดัชนีมวลกายสำหรับแมว

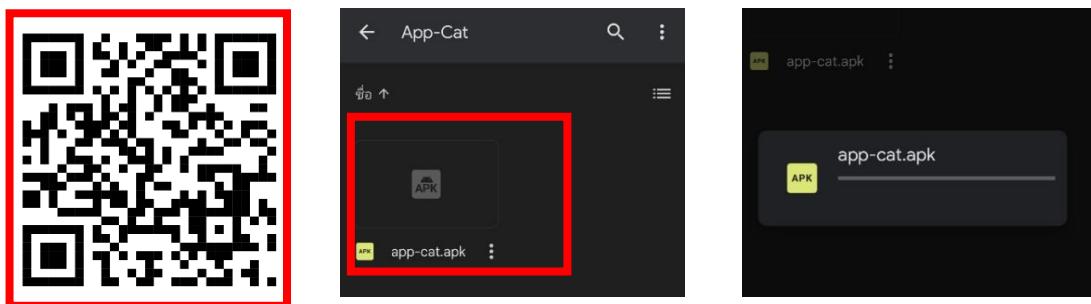


ภาควิชาครุศาสตร์วิศวกรรม
คณะครุศาสตร์อุตสาหกรรมและเทคโนโลยี
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2566

1. คำแนะนำเบื้องต้น

ก่อนที่จะใช้งานแอปพลิเคชัน Cat ควรศึกษาการใช้งานจากคู่มือให้เข้าใจ เพื่อการใช้บริการที่ถูกต้อง

2. วิธีการติดตั้งแอปพลิเคชัน

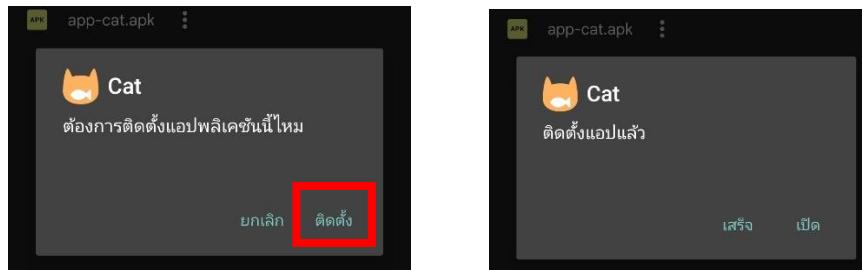


ก. คิวอาร์โค้ดดาวน์โหลด

แอปพลิเคชัน

ข. ไฟล์ดาวน์โหลดแอปพลิเคชัน

ค. การดำเนินการดาวน์โหลด



ก. ขั้นตอนติดตั้งแอปพลิเคชัน

จ. การติดตั้งเสร็จสมบูรณ์

รูปที่ จ.1 การติดตั้งแอปพลิเคชัน

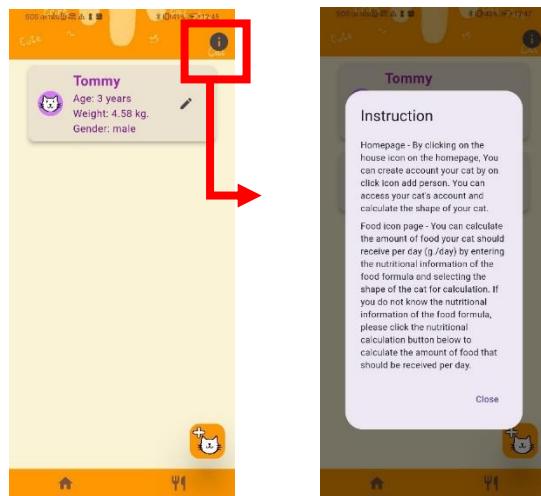
จากรูปที่ จ.1 สามารถติดตั้งแอปพลิเคชัน Cat ได้ดังนี้

1) ขั้นตอนแรกสแกนคิวอาร์โค้ด เพื่อทำการดาวน์โหลดแอปพลิเคชัน

- 2) หลังจากกดดาวน์โหลดจะมีข้อความสอบถาม ต้องการติดตั้งแอปพลิเคชันหรือไม่ ให้ทำ การเลือก ติดตั้ง
- 3) สามารถเข้าใช้งานแอปพลิเคชัน Cat ได้

3. การแสดงคำแนะนำการใช้งานแอปพลิเคชัน

คำแนะนำการใช้งานแอปพลิเคชันช่วยให้ผู้ใช้งานเข้าใจว่า ควรใช้งานแอปพลิเคชันอย่างไร โดยคำแนะนำจะแสดงในรูปแบบพ้อปอปป์ ซึ่งสามารถกดที่ปุ่มมุมขวาบนในหน้าหลัก จากนั้นแอปพลิเคชันจะแสดงพ้อปอปคำแนะนำขึ้นมา ดังรูปที่ จ.2

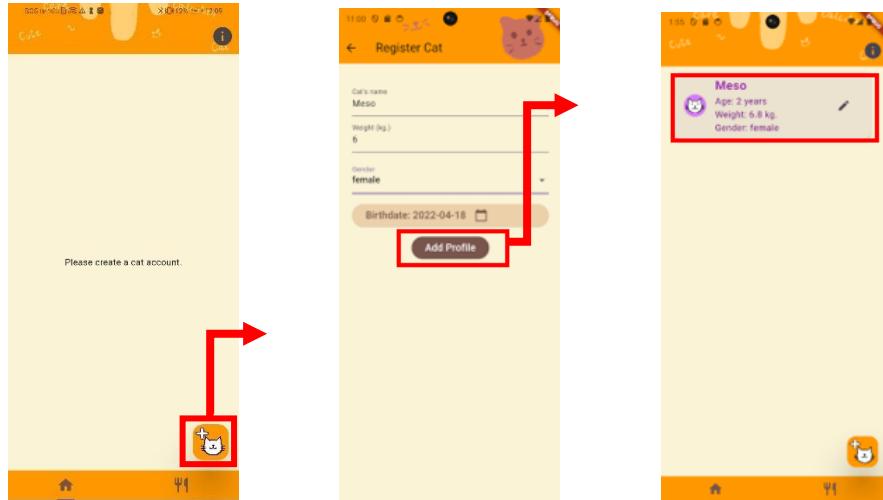


ก. ปุ่มคำแนะนำ
ในหน้าหลัก ข. คำแนะนำการใช้งาน

รูปที่ จ.2 หน้าแสดงคำแนะนำการใช้งาน

4. การลงทะเบียนบัญชีแมว

การเข้าใช้งานแอปพลิเคชันจะเริ่มต้นด้วยการลงทะเบียนบัญชีแมว เพื่อให้ผู้ใช้งานเห็นข้อมูลของแมวที่เลี้ยงและแสดงในรูปแบบการ์ด โดยวิธีการลงทะเบียนบัญชีแมวผู้ใช้งานจะต้องกดที่ปุ่มเพิ่มบัญชีแมว ที่มุมขวาล่างในหน้าหลักเพื่อเข้าสู่หน้าลงทะเบียนแมว จากนั้นทำการกรอกข้อมูลให้ครบถ้วนแล้วทำการบันทึก เมื่อกลับมาที่หน้าหลักจะพบว่ามีการแสดงบัญชีแมวเรียบร้อยแล้ว ดังรูปที่ จ.3



ก. ปุ่มลงทะเบียน

บัญชีแมวใน

หน้าหลัก

ข. หน้าลงทะเบียนแมว

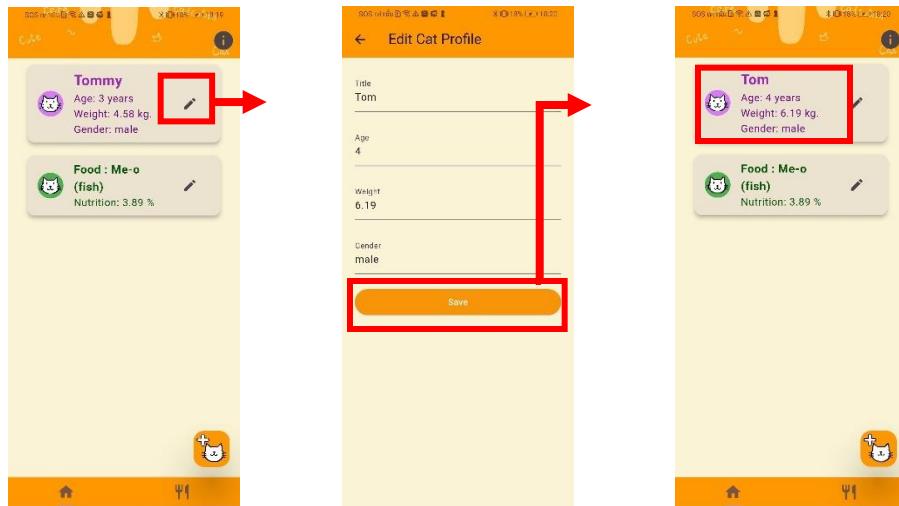
บัญชีแมว

ค. หน้าหลักแสดงการ์ด

รูปที่ จ.3 ตัวอย่างการลงทะเบียนบัญชีแมว

5. การแก้ไขข้อมูลในบัญชีแมว

การแก้ไขข้อมูลในบัญชีแมวเมื่อพบรูปเบื้องต้นของผู้ใช้สามารถเข้าสู่หน้าการแก้ไขข้อมูลได้ด้วยการกดที่ปุ่มรูปดินสอในรายละเอียดบัญชีแมวตัวที่ต้องการแก้ไข จากนั้นกรอกข้อมูลใหม่แทนที่ข้อมูลเดิมให้ครบถ้วนแล้วกดบันทึก เมื่อกลับมาที่หน้าหลักจะแสดงบัญชีแมวที่ได้รับแก้ไขข้อมูลเรียบร้อย ดังรูปที่ จ.4



ก. ปุ่มแก้ไขข้อมูล
ในบัญชีแมว

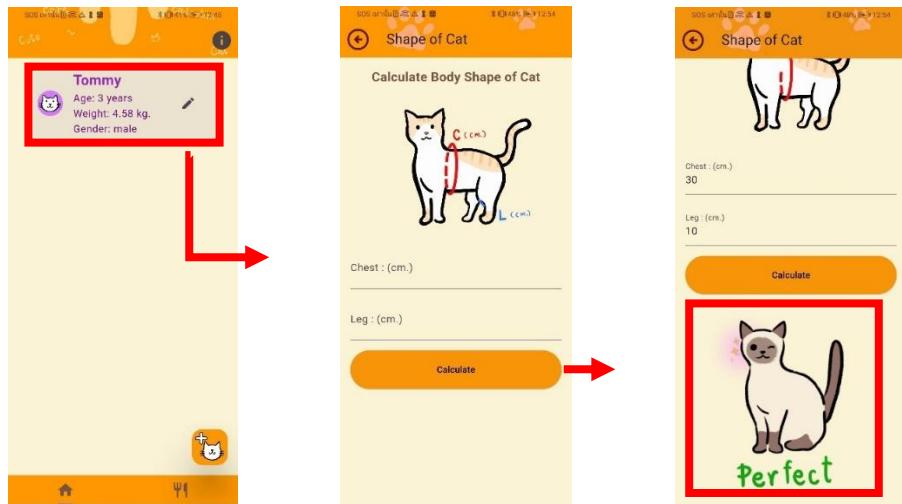
ข. หน้าแก้ไขข้อมูลแมว

ค. หน้าหลักแสดงการ์ด
บัญชีแมวที่แก้ไขแล้ว

รูปที่ จ.4 ตัวอย่างการแก้ไขข้อมูลในบัญชีแมว

6. การคำนวณดัชนีมวลกายสำหรับแมว

การคำนวณดัชนีมวลกายสำหรับแมวจะทำให้ทราบว่า แมวที่เลี้ยงมีดัชนีมวลกายอยู่ในเกณฑ์มาตรฐานหรือไม่ โดยแอปพลิเคชันจะแสดงผลลัพธ์ในรูปแบบภาพและข้อความ เมื่อกดที่การ์ดบัญชีแมว แอปพลิเคชันจะนำไปหน้าคำนวณดัชนีมวลกายสำหรับแมว จากนั้นทำการกรอกข้อมูลความยาวรอบอก (ซม.) และความยาวของขาส่วนหลังตั้งแต่หัวเข่าถึงข้อเท้า (ซม.) เมื่อกรอกข้อมูลครบถ้วนให้ทำการกดปุ่มคำนวณเพื่อแสดงผลลัพธ์ดัชนีมวลกายของแมว ดังรูปที่ จ.5



ก. การ์ดบัญชีแมว
ในหน้าหลัก

ข. หน้ากรอกข้อมูลเพื่อ^{คำนวณดัชนีมวลกาย}

ค. ผลลัพธ์ดัชนีมวลกาย

รูปที่ จ.5 ตัวอย่างการคำนวณดัชนีมวลกายสำหรับแมว

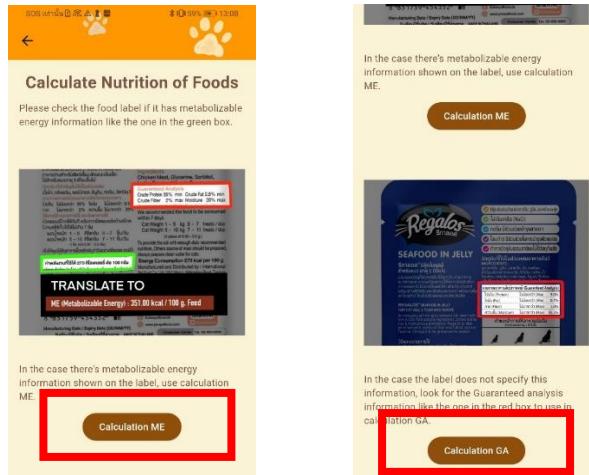
7. การคำนวณโภชนาการของผลิตภัณฑ์

โภชนาการของผลิตภัณฑ์ คือ ข้อมูลหนึ่งที่สำคัญต่อการคำนวณปริมาณอาหารที่ควรได้รับต่อวัน ซึ่งการไปสู่หน้าเลือกสูตรการคำนวณปริมาณอาหารที่ควรได้รับต่อวันจะต้องกดที่ปุ่ม Calculate Nutrition ดังรูปที่ จ.6



รูปที่ จ.6 ปุ่มเข้าสู่หน้าเลือกสูตรในการคำนวณโภชนาการของผลิตภัณฑ์

หน้าการเลือกสูตรคำนวณจะประกอบด้วย ปุ่มเข้าสู่หน้าการคำนวณโดยระบุพลังงานที่นำไปใช้ได้ และปุ่มเข้าสู่หน้าการคำนวณโดยไม่มีระบุพลังงานที่นำไปใช้ได้ ดังรูปที่ จ.7



ก. ปุ่มเข้าสู่หน้า

Calculation ME

ข. ปุ่มเข้าสู่หน้า

Calculation GA

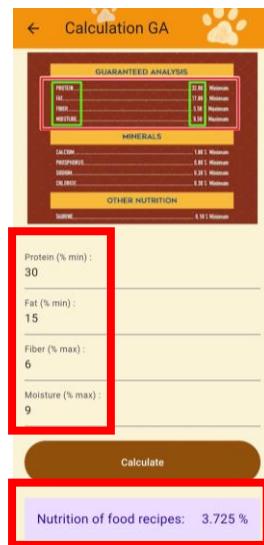
รูปที่ จ.7 หน้าเลือกสูตรในการคำนวณโภชนาการของผลิตภัณฑ์

หน้าคำนวณพลังงานที่นำไปใช้ได้โดยที่ฉลากระบุพลังงานที่นำไปใช้ได้จะให้ผู้ใช้งานกรอกข้อมูล
แคลอรี่ (Kcal) และปริมาณอาหารต่อการคำนวณ (g.) จากนั้นกดคำนวณ ดังรูปที่ จ.8



รูปที่ จ.8 ตัวอย่างคำนวณโภชนาการของผลิตภัณฑ์โดยระบุพลังงานที่สัตว์นำไปใช้ได้

หน้าคำนวณพลังงานที่นำไปใช้ได้โดยที่ฉลากไม่ระบุพลังงานที่นำไปใช้ได้จะให้ผู้ใช้งานกรอกข้อมูลแคลอรี่ (Kcal) และปริมาณอาหารต่อการคำนวณ (g.) จากนั้นกดคำนวณ ดังรูปที่ จ.9



รูปที่ จ.9 ตัวอย่างคำนวณโภชนาการของผลิตภัณฑ์โดยไม่ระบุพลังงานที่สัตว์นำไปใช้ได้

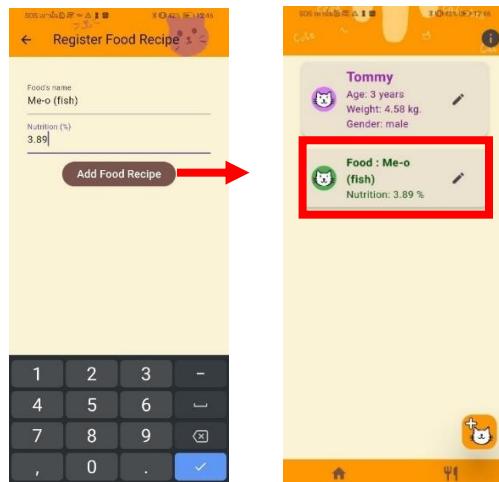
8. การบันทึกสูตรอาหารในรูปแบบการ์ด

การบันทึกสูตรอาหารในรูปแบบการ์ด ช่วยให้ผู้ใช้งานสามารถย้อนกลับมาดูสูตรอาหารที่เคยบันทึกไว้ได้ทุกเมื่อ โดยการเข้าสู่หน้าบันทึกสูตรอาหารให้ทำการกดที่ปุ่ม Save Food Recipe ดังรูปที่ จ.10



รูปที่ จ.10 ปุ่มเข้าสู่หน้าบันทึกสูตรอาหาร

หน้าบันทึกสูตรอาหารในรูปแบบการ์ด ให้ทำการกรอกข้อมูลชื่อสูตรอาหารและโภชนาการของผลิตภัณฑ์ และทำการบันทึกข้อมูล จากนั้นจะแสดงสูตรอาหารในรูปแบบการ์ด ดังรูปที่ จ.11



ก. หน้ากรอกสูตรอาหาร ข. หน้าหลักแสดง

สูตรอาหาร

รูปที่ จ.11 ตัวอย่างการบันทึกสูตรอาหาร

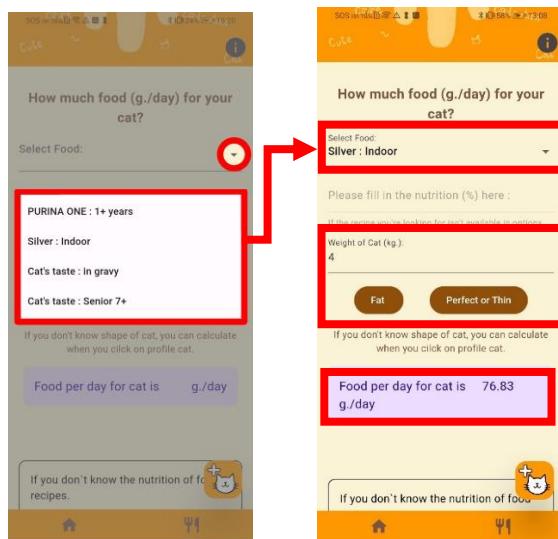
9. การคำนวณปริมาณอาหารที่ควรได้รับต่อวัน

การคำนวณปริมาณอาหารที่ควรได้รับต่อวัน จะทำให้ผู้ใช้งานทราบปริมาณอาหารที่เหมาะสมกับแมวแต่ละตัว ซึ่งการคำนวณสามารถใช้ได้ 2 วิธีการดังนี้

1. การใช้งานแบบเลือกสูตรอาหารที่มีในแอปพลิเคชันจัดเตรียมไว้ ซึ่งมีสูตรอาหารดังนี้

- 1) Purina One : 1+ years
- 2) Silver : indoor
- 3) Cat's taste : in gravy
- 4) Cat's taste : senior 7+

การใช้งานแบบเลือกสูตรอาหารที่มีในแอปพลิเคชัน จะให้ผู้ใช้งานเลือกสูตรอาหารในช่อง Select Food จากนั้นใส่น้ำหนักแล้วเลือกปุ่มดังนี้ตามความต้องการ เช่น น้ำหนัก 4 กิโลกรัม เพื่อคำนวณปริมาณอาหารที่ควรได้รับต่อวัน ดังรูปที่ จ.12

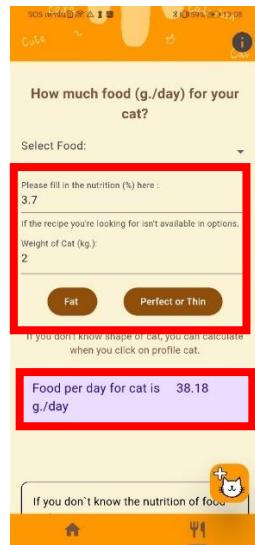


รูปที่ จ.12 ตัวอย่างการคำนวณปริมาณอาหารที่ควรได้รับต่อวันแบบใช้สูตรที่มีในแอปพลิเคชัน

2. การใช้งานแบบกรอกข้อมูลโภชนาการของผลิตภัณฑ์ด้วยตัวเอง

การใช้งานแบบกรอกข้อมูลโภชนาการของผลิตภัณฑ์ด้วยตัวเอง เมื่อตัวเลือก Select Food ไม่มีสูตรอาหารที่ต้องการนำมารามคำนวณปริมาณอาหารที่ควรได้รับต่อวัน ให้ผู้ใช้งานกรอกข้อมูลโภชนาการของ

ผลิตภัณฑ์ในช่อง Please fill in the nutrition here จากนั้นใส่น้ำหนักแล้วเลือกปุ่มดังนี้มีวิธีการของ
แมวเพื่อคำนวณปริมาณอาหารที่ควรได้รับต่อวัน ดังรูปที่ จ.13



รูปที่ จ.13 ตัวอย่างการคำนวณปริมาณอาหารที่ควรได้รับต่อวันแบบกรอกโภชนาการด้วยตัวเอง

ประวัติผู้แต่ง



ชื่อ-สกุล

วัน เดือน ปีเกิด

ภูมิลำเนา

นางสาวณัฏฐณิชา เหลาเพ็ชร

19 มิถุนายน พ.ศ. 2546

47/9 หมู่ 12 ต.ท่าจาม อ.เมืองปราจีนบุรี
จ. ปราจีนบุรี 25000

ประวัติการศึกษา

ประถมศึกษา

โรงเรียนมารีวิทยา ปราจีนบุรี

มัธยมศึกษาตอนต้น

โรงเรียนปราจินราษฎร์อัจฉรุง

มัธยมศึกษาตอนปลาย

โรงเรียนปราจินราษฎร์อัจฉรุง

ปริญญาตรี

ครุศาสตร์อุตสาหกรรมบัณฑิต (ครุศาสตร์วิศวกรรม)

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรมและเทคโนโลยี สจล.

ความสนใจพิเศษ

การทำแอนิเมชั่น และเกมดีไซน์

คติพจน์

เมื่อพบໄอเดีย จงลงมือทำจนกว่ามันจะสำเร็จ

ประวัติผู้แต่ง



ชื่อ-สกุล

วัน เดือน ปีเกิด

ภูมิลำเนา

นางสาวปานวด สاثสนิท

8 มกราคม พ.ศ. 2546

273/1 หมู่ 2 ตำบลเขาหินซ้อน อำเภอ พนมสารคาม

จังหวัด ฉะเชิงเทรา รหัสไปรษณีย์ 24120

ประวัติการศึกษา

ประถมศึกษา

โรงเรียนจุฬาทิพย์ จังหวัด ฉะเชิงเทรา

มัธยมศึกษาตอนต้น

โรงเรียนกาญจนากิจेकวิทยาลัย ฉะเชิงเทรา

มัธยมศึกษาตอนปลาย

โรงเรียนกาญจนากิจेकวิทยาลัย ฉะเชิงเทรา

ปริญญาตรี

ครุศาสตร์อุตสาหกรรมบัณฑิต (ครุศาสตร์วิศวกรรม)

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรมและเทคโนโลยี สจล.

ความสนใจพิเศษ

สนใจในงาน Graphic design

คติพจน์

อย่าดูถูกผู้ด้อยกว่าเรา เพราะเขาก็เป็นผู้ที่ช่วยเหลือเราได้

ประวัติผู้แต่ง



ชื่อ-สกุล

นายพงศ์ศิริ วิลารบรรณ

วัน เดือน ปีเกิด

16 มกราคม พ.ศ. 2546

ภูมิลำเนา

หมู่ 10 ตำบลบึงพระ อำเภอเมืองพิษณุโลก
จังหวัดพิษณุโลก รหัสไปรษณีย์ 65000

ประวัติการศึกษา

ประถมศึกษา

โรงเรียนโรงเรียนเซนต์ นิโกลาส

มัธยมศึกษาตอนต้น

โรงเรียนพิษณุโลกพิทยาคม

มัธยมศึกษาตอนปลาย

โรงเรียนเตรียมอุดมศึกษา ภาคเหนือ

ปริญญาตรี

ครุศาสตร์อุตสาหกรรมบัณฑิต (ครุศาสตร์วิศวกรรม)

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรมและเทคโนโลยี สจล.

ความสนใจพิเศษ

Web design

คติพจน์

อย่าเปลี่ยนแปลงตัวเองให้เหมือนคนอื่นเพียงคนอื่นได้ดี
จะเป็นเป็นตัวเองในรูปแบบที่คนอื่นไม่มีแล้วได้ดีกว่า