โครงงานเลขที่ วศ.คพ. **S007-1/2568**

เรื่อง

GymMate : เพื่อนรู้ใจเรื่องฟิตเนส

โดย

ภาณุเดช เสือเผือก รหัส **650610797**
ภูเบศร์ เรืองคุ้ม รหัส **650610798**
อนวัช รัตนกิจศร รหัส **650610818**

โครงงานนี้
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
ปีการศึกษา **2568**

PROJECT No. CPE S007-1/2568

GymMate : Your Fitness Buddy

Phanudet Sueaphueak 650610797

Pubest Ruengkum 650610798

Anawat Rattanakitsorn 650610818

A Project Submitted in Partial Fulfillment of Requirements

for the Degree of Bachelor of Engineering

Department of Computer Engineering

Faculty of Engineering

Chiang Mai University

2025

หัวข้อโครงงาน	: GymMate : เพื่อนรู้ใจเรื่องฟิตเนส
	: GymMate : Your Fitness Buddy
โดย	: ภาณุเดช เสือเผือก รหัส 650610797
	ภูเบศร์ เรืองคุ้ม รหัส 650610798
	อนวัช รัตนกิจศร รหัส 650610818
ภาควิชา	: วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	: ผศ.ดร. นวดนย์ คุณเลิศกิจ
ปริญญา	: วิศวกรรมศาสตรบัณฑิต
สาขา	: วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	: 2568

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงงานนี้เป็นส่วน-หนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(ผศ.ดร. นวดนย์ คุณเลิศกิจ)

คณะกรรมการสอบโครงงาน

..... ประธานกรรมการ
(ผศ.ดร. นวดนย์ คุณเลิศกิจ)

..... กรรมการ
(ผศ. โดม โพธิกานนท์)

..... กรรมการ
(ผศ.ดร. ลัชนา ระมิงค์วงศ์)

หัวข้อโครงงาน	: GymMate : เพื่อนรู้ใจเรื่องฟิตเนส
	: GymMate : Your Fitness Buddy
โดย	: ภาณุเดช เสือเผือก รหัส 650610797
	ภูเบศร์ เรืองคุ้ม รหัส 650610798
	อนวัช รัตนกิจศร รหัส 650610818
ภาควิชา	: วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	: ผศ.ดร. นวดนย์ คุณเลิศกิจ
ปริญญา	: วิศวกรรมศาสตรบัณฑิต
สาขา	: วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	: 2568

บทคัดย่อ

ปัจจุบันผู้ใช้บริการฟิตเนส โดยเฉพาะผู้เริ่มต้น มักประสบปัญหาขาดความรู้ในการออกกำลังกาย ขาดแรงจูงใจ และไม่มีระบบติดตามความก้าวหน้า ขณะเดียวกันผู้ประกอบการฟิตเนสยังขาดเครื่องมือดิจิทัลที่ช่วยบริหารจัดการเครื่องเล่นและสร้างการมีส่วนร่วมกับสมาชิก โครงงานนี้มีวัตถุประสงค์เพื่อสำรวจและออกแบบระบบ GymMate ซึ่งเป็นแพลตฟอร์มในรูปแบบ Web/Mobile Application ที่ผสานแนวคิด Gamification เข้ากับ Workout Recommendation และ Program Planner ระบบที่ออกแบบมีฟังก์ชันสำคัญ ได้แก่ โปรแกรมการฝึก (ทั้งแบบสำเร็จรูปและแบบปรับแต่งได้), การแสดงแผนผังยิม (Floorplan) พร้อมปุ่ม Flex Substitute เพื่อแนะนำเครื่องหรือท่าทางทดแทนเมื่ออุปกรณ์ไม่ว่าง, ระบบบันทึกและวิเคราะห์ความก้าวหน้า, การสะสมคะแนนและจัดอันดับเพื่อสร้างแรงจูงใจ รวมถึงแดชบอร์ดสำหรับผู้ดูแลในการจัดการข้อมูลเครื่องเล่นและพื้นที่ยิม ผลลัพธ์ของโครงงานนี้คือชุดคุณลักษณะและต้นแบบ UX/UI ที่ผ่านการเก็บและวิเคราะห์ความต้องการจากอาจารย์ที่ปรึกษา เจ้าของยิม และผู้จัดการฟิตเนส โดยอ้างอิงมาตรฐาน ACSM Guidelines และ Physical Activity Guidelines เพื่อนำไปสู่การพัฒนาระบบต้นแบบที่สามารถสนับสนุนการออกกำลังกายอย่างมีประสิทธิภาพและยั่งยืนได้ในอนาคต

Project Title	: GymMate : Your Fitness Buddy	
Name	: Phanudet Sueaphueak	650610797
	Pubest Ruengkum	650610798
	Anawat Rattanakitsorn	650610818
Department	: Computer Engineering	
Project Advisor	: Asst. Prof. Navadon Khunlertgit, Ph.D.	
Degree	: Bachelor of Engineering	
Program	: Computer Engineering	
Academic Year	: 2025	

ABSTRACT

Fitness users, especially beginners, often face challenges such as a lack of exercise knowledge, insufficient motivation, and the absence of effective progress-tracking tools. At the same time, gym owners lack digital solutions to manage equipment efficiently and engage members. This project aims to survey and design GymMate, a Web/Mobile application platform that integrates Gamification with Workout Recommendation and Program Planning. The proposed system includes key features such as preset and customizable workout programs, an interactive Gym Floorplan with a Flex Substitute function to suggest alternative machines or exercises when equipment is occupied, progress tracking and analytics, gamification elements (points, badges, streaks, and leaderboards), and an admin dashboard for managing gym equipment and information. The outcome of this project is a refined set of features and UX/UI prototypes, derived through iterative requirement gathering with advisors, gym owners, and fitness managers. The design also references the ACSM Guidelines and Physical Activity Guidelines, laying the foundation for a future prototype that supports effective and sustainable fitness training.

กิตติกรรมประกาศ

โครงงาน **GymMate : เพื่อนรู้ใจเรื่องฟิตเนส (GymMate : Your Fitness Buddy)** ไม่อาจสำเร็จลุล่วงได้ หากปราศจากแรงสนับสนุน คำแนะนำ และความช่วยเหลือจากหลายฝ่าย คณะผู้จัดทำขอถือโอกาสนี้แสดงความขอบคุณอย่างจริงใจต่อทุกท่านที่มีส่วนร่วมในการผลักดันให้โครงงานนี้สำเร็จลุล่วง

คณะผู้จัดทำขอขอบคุณ **ผู้ช่วยศาสตราจารย์ ดร. นวดนย์ คุณเลิศกิจ** อาจารย์ที่ปรึกษาโครงงาน ที่ได้ให้คำแนะนำทั้งด้านวิชาการและแนวทางการพัฒนาอย่างต่อเนื่องตลอดระยะเวลาของโครงงาน นอกจากนี้ขอขอบคุณ **ผู้ช่วยศาสตราจารย์ โดม โพธิกานนท์** และ **ผู้ช่วยศาสตราจารย์ ดร. ลัชนา ระมิงค์วงศ์** อาจารย์กรรมการสอบโครงงาน ที่ได้สละเวลาตรวจสอบและให้ข้อเสนอแนะอันมีค่า

คณะผู้จัดทำขอขอบคุณ **ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่** ที่ให้การสนับสนุนด้านทรัพยากรและองค์ความรู้ รวมถึงผู้เข้าร่วมทดสอบการใช้งานระบบ (Usability Testing) ทุกท่าน ที่ให้ข้อเสนอแนะอันเป็นประโยชน์ต่อการพัฒนาระบบ สุดท้ายขอขอบคุณครอบครัวทุกท่านที่ให้การสนับสนุนและเป็นกำลังใจตลอดมา

ภาณุเดช เสือเผือก
ภูเบศร์ เรืองคุ้ม
อนวัช รัตนกิจศร
24 มีนาคม 2569

สารบัญ

บทคัดย่อ . . . . . ข
Abstract . . . . . ค
กิตติกรรมประกาศ . . . . . ง
สารบัญ . . . . . จ
สารบัญรูป . . . . . ช
สารบัญตาราง . . . . . ซ

1 บทนำ **1**
1.1 ที่มาของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตของโครงงาน 1
1.3.1 กลุ่มเป้าหมาย 1
1.3.2 ขอบเขตฟังก์ชัน 2
1.3.3 ขอบเขตด้านฮาร์ดแวร์ 2
1.3.4 ขอบเขตด้านซอฟต์แวร์ 3
1.4 ข้อจำกัดของระบบ 3
1.5 ประโยชน์ที่ได้รับ 3
1.6 เทคโนโลยีและเครื่องมือที่ใช้ 3
1.7 แผนการดำเนินงาน 4
1.8 บทบาทและความรับผิดชอบ 4
1.9 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม 5

2 หลักการและทฤษฎีที่เกี่ยวข้อง **6**
2.1 หลักการออกแบบโปรแกรมออกกำลังกาย (FITT-VP Principle) 6
2.2 ทฤษฎีแรงจูงใจและความต่อเนื่องในการออกกำลังกาย (Self-Efficacy & Exercise Adherence) 6
2.3 Gamification และการออกแบบระบบแรงจูงใจดิจิทัล 6
2.4 การพัฒนา Mobile Application (React Native & Expo) 7
2.5 การพัฒนา Web Application (Vite + React SPA) 7
2.6 การออกแบบ Backend ด้วย Clean Architecture และ REST API 7
2.6.1 Clean Architecture 7
2.6.2 REST API 8
2.6.3 Stateless Application และ JWT Authentication 9
2.7 Cosine Similarity 9
2.8 งานวิจัยและระบบที่เกี่ยวข้อง 10
2.9 เครื่องมือที่ใช้ในการพัฒนา 10

3 การออกแบบและพัฒนาระบบ **11**
3.1 ภาพรวมสถาปัตยกรรมระบบ (3-Tier Architecture & Infrastructure) 11
3.2 ผู้ใช้งานและ Use Case ของระบบ 11
3.3 User Flow 12
3.3.1 Authentication Flow 12
3.3.2 Workout Plan Flow 12
3.3.3 Flex Substitute Flow 12
3.3.4 Progress และ History Flow 12
3.3.5 Floor Plan Flow 13

3.3.6 Leaderboard Flow . . . 14
3.3.7 Trainer Portal Flow . . . 14
3.4 การออกแบบฐานข้อมูล (ERD) . . . 14
3.4.1 Users & Authentication . . . 14
3.4.2 Exercise Library . . . 14
3.4.3 Programs & Sessions . . . 15
3.4.4 User Programs & Scheduling (Snapshot Pattern) . . . 15
3.4.5 Workout Logging . . . 15
3.4.6 Floorplan . . . 15
3.4.7 Gamification . . . 16
3.4.8 Tracking . . . 16
3.5 การออกแบบ API และระบบ Authentication . . . 16
3.5.1 การออกแบบ RESTful API . . . 16
3.5.2 ระบบ Authentication . . . 17
3.5.3 ระบบ Authorization (RBAC) . . . 17
3.6 การออกแบบและพัฒนาฟีเจอร์หลัก . . . 17
3.6.1 Member UI (Mobile App) . . . 17
3.6.2 Admin UI (Web Application) . . . 18
3.6.3 Trainer UI (Web Application) . . . 19
3.7 การออกแบบ UI/UX . . . 19

4 การทดสอบระบบ 21
4.1 Unit Testing . . . 21
4.1.1 โครงสร้างของ Test . . . 21
4.1.2 หลักการทดสอบ (Arrange-Act-Assert) . . . 21
4.1.3 การรัน Test และ CI/CD . . . 21
4.2 การทดสอบการใช้งานจริง (Usability Testing / UAT) . . . 22
4.2.1 กลุ่มผู้ทดสอบและรูปแบบการเก็บข้อมูล . . . 22
4.2.2 หัวข้อการประเมิน . . . 22
4.2.3 ผลการทดสอบ . . . 22
4.3 การประเมินผลตามวัตถุประสงค์ของโครงการ . . . 22

5 สรุปและข้อเสนอแนะ 24
5.1 สรุปผลการดำเนินงาน . . . 24
5.2 ปัญหาและอุปสรรคที่พบ . . . 24
5.3 ข้อเสนอแนะและแนวทางพัฒนาต่อ . . . 25

บรรณานุกรม 26

สารบัญรูป

2.1 แผนภาพ Clean Architecture ที่ใช้ในการออกแบบ Backend ของ GymMate 8

3.1 ภาพรวม User Flow ของระบบ GymMate ฝั่งผู้ใช้งาน (Member) 13
3.2 ภาพรวม User Flow ของระบบ GymMate ฝั่งผู้ดูแลระบบ (Admin/Trainer) 14
3.3 ตัวอย่างหน้า Floor Plan ของ GymMate แสดงตำแหน่งอุปกรณ์ภายในยิม 18
3.4 Problem-to-Feature Mapping แสดงความสัมพันธ์ระหว่างปัญหาและฟีเจอร์ของ GymMate 20

สารบัญตาราง

2.1 การตีความค่า Cosine Similarity 9

3.1 HTTP Methods ที่ใช้ใน GymMate API 17
3.2 RBAC Role ในระบบ GymMate 17

4.1 ตัวอย่าง Test Scenarios ในระบบ GymMate 22
4.2 ผลการประเมินตามวัตถุประสงค์ของโครงงาน GymMate 23

บทที่ 1
บทนำ

1.1 ที่มาของโครงงาน

ปัจจุบันมีการส่งเสริมให้นักศึกษามหาวิทยาลัยเชียงใหม่หันมาใส่ใจสุขภาพและการออกกำลังกายมากขึ้น โดยมีบริการฟิตเนสภายในมหาวิทยาลัย เช่น CMU Fitness by Playground อย่างไรก็ตาม พบว่าผู้ใช้งานจำนวนไม่น้อยยังประสบปัญหาขาดแรงจูงใจ ความต่อเนื่องในการออกกำลังกาย รวมถึงความยุ่งยากในการวางแผนการฝึกและการเลือกใช้อุปกรณ์อย่างเหมาะสม

จากสภาพปัญหาดังกล่าว จึงเกิดแนวคิดในการพัฒนาโครงงาน GymMate ซึ่งมีจุดมุ่งหมายเพื่อเป็นต้นแบบระบบที่ช่วยสนับสนุนการออกกำลังกายของนักศึกษา โดยเน้นการสร้างแรงจูงใจและการจัดการการใช้งานฟิตเนสให้มีประสิทธิภาพมากยิ่งขึ้น

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 พัฒนาระบบการวางแผนการออกกำลังกาย (Workout Program Planner)
GymMate มุ่งเน้นการพัฒนาโปรแกรมฝึกที่สามารถเลือกได้ทั้งรูปแบบสำเร็จรูป (preset) และการปรับแต่งตามความต้องการ (custom) เพื่อช่วยให้นักศึกษาสามารถวางแผนการออกกำลังกายได้สะดวกและเหมาะสมกับเป้าหมายของตนเอง

1.2.2 เพิ่มประสิทธิภาพการออกกำลังกายด้วยระบบ Flex Substitute
ด้วยระบบแนะนำท่าและเครื่องเล่นทดแทนอัตโนมัติ เมื่ออุปกรณ์ที่ต้องการไม่พร้อมใช้งาน ผู้ใช้สามารถเลือกทางเลือกที่เหมาะสมกับกล้ามเนื้อที่ต้องการฝึกได้ โดยระบบจะช่วยแนะนำผ่านอัลกอริทึมที่ออกแบบมาโดยเฉพาะ

1.2.3 จัดการฟิตเนสอย่างเป็นระบบผ่าน CMS Dashboard
ฝั่งผู้ดูแลฟิตเนสสามารถจัดการตำแหน่งและข้อมูลเครื่องเล่นได้อย่างมีประสิทธิภาพ ลดความซับซ้อนในการบริหารจัดการ และเพิ่มความสะดวกให้กับทั้งผู้ดูแลและผู้ใช้งาน

1.2.4 สร้างแรงจูงใจและความต่อเนื่องในการออกกำลังกาย
ด้วยระบบ Gamification และ Leaderboard ผู้ใช้งานสามารถสะสมคะแนน แข่งขันกับผู้ใช้คนอื่น ติดตามความก้าวหน้าของตนเอง ทำให้การออกกำลังกายน่าสนใจ สนุก และมีแรงกระตุ้นมากขึ้น

1.3 ขอบเขตของโครงงาน

โครงงาน GymMate มีขอบเขตการดำเนินงานที่เน้นการ สำรวจและออกแบบระบบต้นแบบ โดยมีรายละเอียดดังนี้

1.3.1 กลุ่มเป้าหมาย

- **ผู้ใช้บริการฟิตเนส (Fitness Members):**
  โดยเฉพาะกลุ่มผู้เริ่มต้น (Beginners) และนักศึกษามหาวิทยาลัยเชียงใหม่ ที่ต้องการโปรแกรมฝึก การติดตามความก้าวหน้า และแรงจูงใจในการออกกำลังกาย

- **ผู้จัดการและเจ้าของฟิตเนส (Gym Managers & Owners):**
  ต้องการเครื่องมือในการบริหารจัดการเครื่องเล่นและสื่อสารกับสมาชิก เพื่อเพิ่มประสิทธิภาพการใช้งานพื้นที่และเสริมความพึงพอใจของลูกค้า

1.3.2 ขอบเขตฟังก์ชัน

1. **Workout Program Planner** ระบบวางแผนโปรแกรมการออกกำลังกาย
   ผู้เริ่มต้นสามารถเลือกใช้ Preset Plan เพื่อสร้างพื้นฐานการฝึกอย่างเป็นระบบ หรือปรับเป็น Custom Plan ให้สอดคล้องกับเป้าหมายเฉพาะบุคคล พร้อมแนวทาง Goal & Approach เช่น Push–Pull–Legs, Upper–Lower หรือ Total Body เพื่อเพิ่มประสิทธิภาพ
2. **Floorplan UI & Flex Substitute** ส่วนติดต่อผู้ใช้สำหรับผังพื้นที่และการปรับเปลี่ยนแบบยืดหยุ่น
   ระบบมีการจัดทำ แผนผังยิม เพื่อแสดงตำแหน่งและการจัดวางของเครื่องเล่นอย่างชัดเจน อีกทั้งมีฟังก์ชัน ปุ่ม Flex Substitute ที่ช่วยแนะนำเครื่องหรือท่าทางทดแทนเมื่ออุปกรณ์ที่ต้องการใช้งานเต็ม โดยยังคงมุ่งเน้นการฝึกที่กล้ามเนื้อเป้าหมายเดิมเพื่อไม่ให้ผลลัพธ์การออกกำลังกายคลาดเคลื่อน.
3. **User Management & Tracking** ระบบจัดการและติดตามผู้ใช้งาน
   ระบบมี โปรไฟล์ผู้ใช้ สำหรับจัดเก็บข้อมูลการฝึก พร้อมการบันทึกสถิติ เช่น จำนวนเซต, ครั้งทำซ้ำ และน้ำหนักที่ใช้ นอกจากนี้ยังมี กราฟความก้าวหน้า เพื่อสะท้อนพัฒนาการอย่างต่อเนื่อง รวมถึงรายงานผลการฝึกในรูปแบบรายสัปดาห์และรายเดือน เพื่อช่วยประเมินและปรับแผนการออกกำลังกายได้อย่างเป็นระบบ
4. **Gamification Layer** ชั้นการใช้องค์ประกอบเชิงเกม
   ระบบมีการสะสมคะแนนจากความสม่ำเสมอในการฝึก พร้อมกลไกสร้างแรงจูงใจผ่าน Badge, Level, Streak และ Leaderboard ซึ่งช่วยเสริมทั้งความต่อเนื่องและการแข่งขันเชิงบวกระหว่างผู้ใช้
5. **Content & Notification** ระบบจัดการเนื้อหาและการแจ้งเตือน
   ระบบรองรับ วิดีโอและภาพประกอบท่าทาง ผ่าน QR code ที่ติดไว้บนเครื่องออกกำลังกาย เพื่อให้ผู้ใช้เข้าถึงวิธีการใช้งานได้อย่างถูกต้อง นอกจากนี้ยังมี การแจ้งเตือน (Notification) ตามตารางโปรแกรม รวมถึงการเปลี่ยนท่าอัตโนมัติเมื่อเครื่องที่ต้องการใช้งานเต็ม เพื่อคงประสิทธิภาพของการฝึก
6. **Admin Dashboard (CMS)** แดชบอร์ดผู้ดูแลระบบ
   ระบบสามารถ จัดการข้อมูลเครื่องเล่น ได้อย่างยืดหยุ่น ทั้งการเพิ่ม ย้าย หรือแก้ไขสถานะของเครื่องออกกำลังกาย พร้อมทั้งแสดง ข้อมูลพื้นฐานของฟิตเนส เช่น เวลาเปิด–ปิด และข่าวสารที่เกี่ยวข้อง เพื่ออำนวยความสะดวกแก่ผู้ใช้

1.3.3 ขอบเขตด้านฮาร์ดแวร์

โครงงานนี้ไม่ได้พัฒนาฮาร์ดแวร์ใหม่ แต่ใช้โครงสร้างพื้นฐานและบริการที่มีอยู่ ได้แก่

- สมาร์ทโฟนของผู้ใช้งานสำหรับติดตั้งและใช้งานแอปพลิเคชัน

- สถานที่และเครื่องเล่นออกกำลังกายของยิม

1.3.4 ขอบเขตด้านซอฟต์แวร์

ขอบเขตด้านซอฟต์แวร์ของโครงงานประกอบด้วย

- ต้องรองรับการใช้งานและติดตั้งแอปพลิเคชันบนสมาร์ทโฟนของผู้ใช้
- ต้องรองรับการใช้งานผ่านคอมพิวเตอร์สำหรับการจัดการ CMS Dashboard ของผู้ดูแลระบบ

1.4 ข้อจำกัดของระบบ

1.4.1 ระบบเป็นเพียงต้นแบบ (Prototype) ที่ยังไม่ได้นำไปใช้งานจริงในสภาพแวดล้อมฟิตเนสจริง

1.4.2 ระบบยังไม่มีการเชื่อมต่อกับอุปกรณ์หรือเซนเซอร์ (IoT) ทำให้ไม่สามารถตรวจสอบสถานะการใช้งานเครื่องออกกำลังกายแบบเรียลไทม์ได้

1.4.3 ข้อมูลการออกกำลังกายที่บันทึกขึ้นอยู่กับการกรอกของผู้ใช้ ซึ่งอาจมีความคลาดเคลื่อน

1.4.4 ระบบ Flex Substitute เป็นการแนะนำโดยใช้หลักการที่กำหนดไว้ (rule-based) ยังไม่ได้ใช้ปัญญาประดิษฐ์ขั้นสูง

1.4.5 ยังไม่มีการทดสอบระบบกับผู้ใช้งานจริงในวงกว้าง ทำให้ผลลัพธ์ด้านประสิทธิภาพและความพึงพอใจยังไม่สามารถสรุปได้อย่างครอบคลุม

1.5 ประโยชน์ที่ได้รับ

1.4.1 ได้ต้นแบบแอปพลิเคชัน GymMate ที่ช่วยสนับสนุนการออกกำลังกายและเพิ่มแรงจูงใจสำหรับนักศึกษามหาวิทยาลัยเชียงใหม่

1.4.2 ผู้ใช้งานสามารถวางแผนโปรแกรมการออกกำลังกายได้สะดวกและเหมาะสมกับเป้าหมายของตนเอง ทั้งแบบสำเร็จรูปและแบบปรับแต่ง

1.4.3 ผู้ดูแลฟิตเนสสามารถบริหารจัดการตำแหน่งและข้อมูลเครื่องเล่นได้ง่ายขึ้น ผ่าน CMS Dashboard

1.4.4 ระบบช่วยเพิ่มความต่อเนื่อง สนุก และประสิทธิภาพในการออกกำลังกายของนักศึกษา

1.4.5 โครงงานสามารถต่อยอดไปสู่การใช้งานจริง และขยายผลสำหรับฟิตเนสหรือมหาวิทยาลัยอื่นได้

1.6 เทคโนโลยีและเครื่องมือที่ใช้

เพื่อพัฒนาและนำเสนอระบบ GymMate ได้ตามเป้าหมาย โครงงานนี้เลือกใช้เทคโนโลยีหลัก 3 ส่วน ดังนี้

1.5.1 **Web Application (CMS Dashboard)**

ระบบสำหรับ ผู้ดูแล (Admin/Gym Manager) ทำหน้าที่บริหารจัดการข้อมูลเครื่องเล่น รวมถึงการย้ายหรือแก้ไขสถานะใน Floorplan และจัดการข้อมูลฟิตเนส เช่น ข่าวสารและเวลาเปิด–ปิด เพื่อให้การดำเนินงานเป็นระเบียบและมีประสิทธิภาพ

1.5.2 **Mobile Application (User App)**

ระบบสำหรับ ผู้ใช้ฟิตเนส (Members) รองรับการเลือกโปรแกรมฝึกที่เหมาะสม พร้อมฟังก์ชัน Calendar/Day Plan เพื่อดูตารางการฝึก และ Floorplan Highlight ช่วยระบุตำแหน่งเครื่องเล่น นอกจากนี้ยังมี Flex Substitute สำหรับแนะนำท่าทางทดแทน, Log Progress เพื่อบันทึกผลการฝึก, ระบบ Gamification เพื่อสร้างแรงจูงใจ และ Notification สำหรับแจ้งเตือนตามโปรแกรมอย่างต่อเนื่อง

1.5.3 **Server & Backend Services**

ระบบนี้ใช้สำหรับ ประมวลผลและจัดเก็บข้อมูล ที่เกี่ยวข้องกับโปรแกรมการฝึก การใช้งาน และการจัดการเครื่อง โดยมีฟังก์ชันหลัก ได้แก่ Workout Planner สำหรับวางแผนการฝึก, Flex Substitute Algorithm เพื่อแนะนำทางเลือกเมื่อเครื่องเต็ม, ระบบ Authentication/Authorization เพื่อควบคุมสิทธิ์การใช้งาน และ Data Logging สำหรับเก็บบันทึกข้อมูลอย่างเป็นระบบ

1.7 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	มิ.ย. 2568	ก.ค. 2568	ส.ค. 2568	ก.ย. 2568	ต.ค. 2568	พ.ย. 2568	ธ.ค. 2568	ม.ค. 2569	ก.พ. 2569	มี.ค. 2569
ศึกษาค้นคว้าและหาหัวข้อสำหรับโครงงาน	■	■								
เก็บและวิเคราะห์ความต้องการ		■	■	■						
ออกแบบฟีเจอร์และปรับขอบเขต			■	■						
ออกแบบ UX/UI และต้นแบบ				■	■	■				
ออกแบบเชิงเทคนิค: Database, API, System Architecture						■	■			
พัฒนา Backend							■	■	■	
พัฒนา Mobile App								■	■	
พัฒนา Web CMS (Admin Dashboard, Floorplan Editor)								■	■	
ทดลองใช้งานจริงกับผู้ใช้									■	■

1.8 บทบาทและความรับผิดชอบ

1. นายภาณุเดช เสือเผือก รับบทบาท Product Owner, Frontend
   - **หน้าที่:** เสนอไอเดียจากประสบการณ์การเข้ายิม, พัฒนา Mobile/Web Frontend, ตรวจสอบความถูกต้องของข้อมูลการออกกำลังกาย
   - **ความรู้ที่ใช้:** ความรู้ด้านการออกกำลังกาย และการพัฒนาส่วนติดต่อผู้ใช้ (Frontend Development)
2. นายภูเบศร์ เรืองคุ้ม รับบทบาท Backend & System

- **หน้าที่:** ออกแบบสถาปัตยกรรมระบบและฐานข้อมูล, พัฒนา API และ Flex Substitute และความปลอดภัยระบบ
- **ความรู้ที่ใช้:** การเขียนโปรแกรมฝั่งเซิร์ฟเวอร์ (Backend Programming), ฐานข้อมูล และความปลอดภัยของระบบ

3. นายอนวัช รัตนกิจศร รับบทบาท UX/UI & Documentation

   - **หน้าที่:** ออกแบบ UX/UI Mockup, จัดทำเอกสารรายงาน, ประสานงานกับอาจารย์และ Stakeholder, สนับสนุนงาน Frontend
   - **ความรู้ที่ใช้:** การออกแบบประสบการณ์ผู้ใช้และส่วนติดต่อผู้ใช้ (UX/UI Design), การเขียนเชิงเทคนิค (Technical Writing), การสื่อสาร (Communication)

1.9 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม

โครงงาน GymMate อาจมีผลกระทบและการประยุกต์ใช้งานในหลายมิติ ดังนี้

1. **ด้านสังคม:** สนับสนุนให้นักศึกษาหันมาออกกำลังกายมากขึ้น ช่วยสร้างแรงจูงใจผ่านระบบ Gamification และ Leaderboard
2. **ด้านสุขภาพ:** ช่วยส่งเสริมสุขภาพกายและจิตใจของผู้ใช้ ลดความเสี่ยงจากโรคที่เกิดจากการขาดการออกกำลังกาย
3. **ด้านความปลอดภัย:** การแนะนำการใช้อุปกรณ์และท่าทางอย่างถูกต้อง ลดความเสี่ยงในการบาดเจ็บจากการออกกำลังกาย
4. **ด้านกฎหมาย:** ต้องคำนึงถึงลิขสิทธิ์ของข้อมูล รูปภาพ และเพลง รวมถึงการจัดเก็บข้อมูลผู้ใช้ให้เป็นไปตามกฎหมายคุ้มครองข้อมูลส่วนบุคคล (PDPA/GDPR)
5. **ด้านวัฒนธรรม:** ปรับการออกแบบแอปให้เหมาะกับวิถีชีวิตและพฤติกรรมของนักศึกษา ส่งเสริมการสร้างวัฒนธรรมการดูแลสุขภาพในมหาวิทยาลัย

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 หลักการออกแบบโปรแกรมออกกำลังกาย (FITT-VP Principle)

หลักการ FITT-VP เป็นแนวทางมาตรฐานในการออกแบบโปรแกรมการออกกำลังกายให้เหมาะสมกับแต่ละบุคคล [1] โดยประกอบด้วยองค์ประกอบสำคัญ 6 ประการ ได้แก่

- **Frequency (ความถี่)**: จำนวนครั้งในการออกกำลังกายต่อสัปดาห์
- **Intensity (ความหนัก)**: ระดับความเข้มข้น เช่น เบา ปานกลาง หรือหนัก
- **Time (ระยะเวลา)**: ระยะเวลาในแต่ละครั้งที่ออกกำลังกาย
- **Type (ประเภท)**: รูปแบบของการออกกำลังกาย เช่น คาร์ดิโอ เวทเทรนนิ่ง
- **Volume (ปริมาณรวม)**: ปริมาณการออกกำลังกายรวม เช่น จำนวนเซตหรือเวลารวมต่อสัปดาห์
- **Progression (การเพิ่มระดับ)**: การปรับเพิ่มความยากหรือปริมาณอย่างเหมาะสม

ในการพัฒนา GymMate หลักการนี้ถูกนำมาใช้ในการแนะนำโปรแกรมการออกกำลังกายที่เหมาะสมกับผู้ใช้แต่ละคน เพื่อเพิ่มประสิทธิภาพและลดความเสี่ยงในการบาดเจ็บ

2.2 ทฤษฎีแรงจูงใจและความต่อเนื่องในการออกกำลังกาย (Self-Efficacy & Exercise Adherence)

ทฤษฎี Self-Efficacy ของ Bandura กล่าวถึงความเชื่อมั่นของบุคคลในความสามารถของตนเอง ซึ่งมีผลโดยตรงต่อพฤติกรรมและความพยายามในการทำกิจกรรมต่างๆ [2] ในบริบทของการออกกำลังกาย

- ผู้ที่มี self-efficacy สูง จะมีแนวโน้มออกกำลังกายสม่ำเสมอ
- ผู้ที่มี self-efficacy ต่ำ มักเลิกกลางคัน

Exercise Adherence คือความสามารถในการรักษาพฤติกรรมการออกกำลังกายอย่างต่อเนื่องในระยะยาว [13] ระบบ GymMate จึงออกแบบให้มีระบบติดตามความคืบหน้า การตั้งเป้าหมาย และ feedback แบบ real-time เพื่อเสริมสร้างแรงจูงใจ

2.3 Gamification และการออกแบบระบบแรงจูงใจดิจิทัล

Gamification คือการนำองค์ประกอบของเกม เช่น คะแนน (points), เหรียญรางวัล (badges) และอันดับ (leaderboard) มาใช้ในระบบที่ไม่ใช่เกม เพื่อเพิ่มแรงจูงใจของผู้ใช้ [3]

ในระบบ GymMate มีการประยุกต์ใช้ Gamification ได้แก่

- ระบบ **Leaderboard** จัดอันดับผู้ใช้งาน
- ระบบ **Achievement / Badge** มอบรางวัลจากการทำสำเร็จ
- ระบบ **Streak** วัดความต่อเนื่องของการออกกำลังกาย
- ระบบ **XP และ Level** สะสมประสบการณ์และพัฒนาระดับ

แนวคิดนี้ช่วยให้ผู้ใช้รู้สึกสนุก ท้าทาย และมีเป้าหมาย ส่งผลให้เกิดการใช้งานอย่างต่อเนื่อง

2.4 การพัฒนา Mobile Application (React Native & Expo)

React Native เป็น Framework สำหรับพัฒนา Mobile Application แบบ Cross-platform [5] ที่สามารถใช้ JavaScript/TypeScript ในการพัฒนาได้ทั้ง iOS และ Android

Expo [4] เป็นเครื่องมือที่ช่วยลดความซับซ้อนในการพัฒนา เช่น

- ไม่ต้องตั้งค่า environment ที่ซับซ้อน
- สามารถทดสอบแอปผ่านมือถือได้ทันที
- รองรับ API ต่างๆ เช่น กล้อง เซนเซอร์

การใช้ React Native ร่วมกับ Expo ทำให้สามารถพัฒนาแอป GymMate ได้รวดเร็ว ลดเวลาในการพัฒนา และดูแลโค้ดได้ง่าย

2.5 การพัฒนา Web Application (Vite + React SPA)

Vite [14] เป็นเครื่องมือสำหรับพัฒนา Frontend ที่มีความเร็วสูง โดยใช้หลักการของ modern bundler

React [12] เป็น JavaScript Library สำหรับสร้าง UI แบบ Component-based

SPA (Single Page Application) คือแอปที่โหลดหน้าเพียงครั้งเดียว และใช้ JavaScript ในการเปลี่ยนเนื้อหา ทำให้โหลดเร็ว UX ลื่นไหล และลดการ reload หน้า

GymMate Web ใช้ Vite + React SPA เพื่อให้ได้ประสบการณ์ผู้ใช้ที่รวดเร็ว และเหมาะกับระบบ dashboard เช่น Admin และ Trainer Portal

2.6 การออกแบบ Backend ด้วย Clean Architecture และ REST API

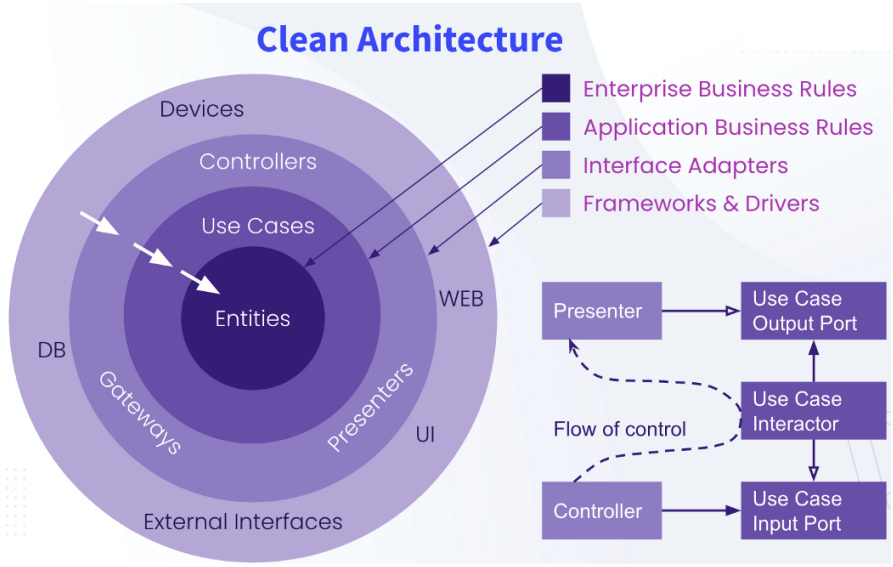
2.6.1 Clean Architecture

Clean Architecture ถูกเสนอโดย Robert C. Martin [8] โดยเป็น software architecture pattern ที่มุ่งเน้นการแยก business logic ออกจากรายละเอียดเชิงเทคนิคภายนอก ทั้ง framework, database และ UI โดยมีหลักการสำคัญ 2 ประการ ได้แก่

1. business rules สามารถทดสอบได้โดยไม่ต้องพึ่งพา UI, database หรือ web server ใดๆ
2. ส่วนประกอบภายนอกสามารถเปลี่ยนแทนได้โดยไม่กระทบต่อ business logic กลาง

องค์ประกอบหลักของ Clean Architecture ที่นำมาใช้ใน GymMate แบ่งออกเป็น 4 layer ดังนี้

1. **Models Layer** — ชั้นในสุด เก็บ core entity เช่น `User`, `Exercise`, `Program`, `WorkoutLog`, `GamificationProfile` ไม่มี dependency กับ layer ใดๆ
2. **Repository Layer** — กำหนด interface สำหรับการเข้าถึงข้อมูล โดย Usecase Layer รู้จักเพียง interface ไม่รู้จัก implementation จริง (เช่น GORM สำหรับ PostgreSQL, MinIO สำหรับ object storage)



รูปที่ 2.1: แผนภาพ Clean Architecture ที่ใช้ในการออกแบบ Backend ของ GymMate

3. **Usecase Layer** — ชั้น business logic หลัก ไม่มี dependency กับ Gin framework หรือ HTTP เก็บ logic เช่น การคำนวณ Flex Substitute ด้วย Cosine Similarity, กฎการให้ XP และ Badge
4. **Delivery Layer** — ชั้นนอกสุด รับ HTTP request ผ่าน Gin framework แปลงเป็น input ส่งต่อให้ Usecase และแปลง output กลับเป็น JSON response

กฎสำคัญของ Clean Architecture คือ dependency จะไหลเข้าด้านในเท่านั้น ทำให้ GymMate สามารถพัฒนาและทดสอบ business logic ของแต่ละฟีเจอร์ได้อิสระจากการเปลี่ยนแปลงของ infrastructure

2.6.2 REST API

REST (Representational State Transfer) เป็น software architecture style สำหรับการออกแบบระบบที่สื่อสารผ่าน HTTP โดย GymMate ใช้ RESTful API เป็นช่องทางสื่อสารหลักระหว่าง Backend กับ Mobile App และ Web CMS Dashboard

Constraints หลักของ REST ที่นำมาใช้ใน GymMate ได้แก่

1. **Client-Server** — แยก UI ออกจาก data storage อย่างชัดเจน ทำให้ Mobile App และ Web CMS พัฒนาได้อิสระจาก Backend
2. **Stateless** — server ไม่เก็บ session state ของ client แต่ละ request ต้องบรรจุข้อมูลที่จำเป็นทั้งหมดในตัวเอง
3. **Uniform Interface** — ใช้ HTTP Methods ตามความหมายมาตรฐาน และ versioning ที่ `/api/v1/`
4. **Layered System** — client ไม่จำเป็นต้องรู้ว่าติดต่อกับ server จริงโดยตรงหรือผ่าน intermediary เช่น Cloudflare Tunnel

2.6.3 Stateless Application และ JWT Authentication

Stateless หมายถึงการออกแบบให้ server ไม่เก็บ state ของ session ผู้ใช้ไว้ในหน่วยความจำของตัวเอง ข้อดีได้แก่

- **Scalability** — สามารถเพิ่ม server instance ได้โดยไม่ต้องกังวลเรื่อง session synchronization
- **Reliability** — หาก server instance หนึ่งล้มเหลว request ถัดไปสามารถส่งไปยัง instance อื่นได้ทันที

GymMate implement Stateless ผ่านกลไก **JWT (JSON Web Token) HS256** โดยมีรายละเอียดดังนี้

- เมื่อผู้ใช้ login สำเร็จ server จะออก JWT ที่ลงนามด้วย secret key และส่งกลับในรูป HTTP-only cookie
- ใน request ถัดๆ ไป client แนบ JWT มาด้วยทุกครั้ง server จะตรวจสอบลายเซ็นและดึงข้อมูล `user_id` และ `role` ออกจาก token โดยตรง
- HTTP-only cookie ป้องกัน JavaScript ฝั่ง client อ่าน token ได้โดยตรง ลดความเสี่ยงจาก XSS attack

2.7 Cosine Similarity

Cosine Similarity เป็น metric ทางคณิตศาสตร์ที่ใช้วัดความคล้ายคลึงระหว่าง vector สองตัว โดยคำนวณจากมุมระหว่าง vector ทั้งสองในปริภูมิหลายมิติ (multi-dimensional space) นิยามดังสมการที่ 2.1

$$\text{cosine similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \, \|\mathbf{B}\|} \tag{2.1}$$

โดยที่ $\mathbf{A} \cdot \mathbf{B}$ คือ dot product ระหว่าง vector และ $\|\mathbf{A}\|$, $\|\mathbf{B}\|$ คือ Euclidean norm ของ vector แต่ละตัว ผลลัพธ์มีค่าอยู่ในช่วง $[0, 1]$ สำหรับ vector ที่มีค่าไม่ติดลบ

ค่า Cosine Similarity	ความหมาย
1.0	คล้ายกันสมบูรณ์ (ชี้ทิศทางเดียวกัน)
0.5 – 1.0	คล้ายกันในระดับสูง
0.0	ไม่มีความคล้ายคลึงกันเลย
ต่ำกว่า 0.0	ชี้ทิศทางตรงกันข้าม

ตารางที่ 2.1: การตีความค่า Cosine Similarity

ใน GymMate Cosine Similarity ถูกนำมาใช้ในระบบ Flex Substitute โดยแปลงข้อมูลสัดส่วนการใช้งานกล้ามเนื้อ (muscle activation percentage) ของแต่ละท่าออกกำลังกายเป็น vector แล้วคำนวณความคล้ายคลึงเพื่อแนะนำท่าทดแทนที่เน้นกล้ามเนื้อกลุ่มเดียวกันมากที่สุด

2.8 งานวิจัยและระบบที่เกี่ยวข้อง

มีงานวิจัยจำนวนมากที่ศึกษาการใช้เทคโนโลยีเพื่อส่งเสริมการออกกำลังกาย เช่น แอปติดตามสุขภาพ (Fitness Tracking Apps), ระบบแนะนำโปรแกรมออกกำลังกาย และระบบ Gamification เพื่อเพิ่ม motivation [13]

ผลการศึกษาพบว่า การใช้ Gamification และระบบติดตามผลสามารถเพิ่มอัตราการออกกำลังกายอย่างต่อเนื่องได้ นอกจากนี้ยังมีแอปพลิเคชันที่คล้ายคลึง เช่น MyFitnessPal, Nike Training Club และ Strava ซึ่งเป็นแนวทางในการพัฒนา GymMate ให้มีฟีเจอร์ที่ตอบโจทย์ผู้ใช้มากขึ้น

2.9 เครื่องมือที่ใช้ในการพัฒนา

ทีมพัฒนาได้เลือกใช้เครื่องมือที่ครอบคลุมทุกขั้นตอนของ software development lifecycle ตั้งแต่การวางแผน การออกแบบ การพัฒนา ไปจนถึงการ deploy และ monitoring โดยแบ่งตามหน้าที่ได้ดังนี้

- **Project Management** — Jira สำหรับวางแผนและติดตามงานในรูปแบบ Agile โดยแบ่งงานออกเป็น Sprint จัดการ Backlog และกำหนด task ให้สมาชิก
- **UI/UX Design** — Figma สำหรับออกแบบ Wireframe, Prototype และ UI ของทั้ง Mobile App และ Web CMS Dashboard รองรับการทำงานร่วมกันแบบ real-time
- **Version Control** — Git และ GitHub โดยแบ่ง repository ออกเป็น 3 ส่วนตามชั้น architecture ได้แก่ BACKEND, FRONTEND และ MOBILE
- **CI/CD** — GitHub Actions สำหรับ pipeline อัตโนมัติ ได้แก่ CI pipeline (lint + unit test) และ CD pipeline (build Docker image □ push □ deploy อัตโนมัติ) พร้อม Discord Webhook แจ้งเตือนผลการ deploy
- **API Documentation** — Swagger UI สร้างจากไฟล์ `openapi.yaml` deploy ที่ port 8081 ใช้เป็นสื่อกลาง API specification ระหว่าง Backend กับทีม Frontend/Mobile
- **Infrastructure** — Docker + Docker Compose สำหรับ containerize application ทุกส่วน; Proxmox VE (Type 1 Hypervisor) สำหรับแบ่ง Virtual Machine; Tailscale (WireGuard-based VPN) สำหรับเชื่อมต่อ dev machine ทั้ง 3 เครื่องกับ server; Cloudflare Tunnel สำหรับ expose ระบบสู่ internet ที่ domain `gymmate.site` และ `api.gymmate.site`; Dozzle สำหรับ real-time container log viewer

บทที่ 3
การออกแบบและพัฒนาระบบ

3.1 ภาพรวมสถาปัตยกรรมระบบ (3-Tier Architecture & Infrastructure)

ระบบ GymMate ถูกออกแบบโดยใช้แนวคิด **3-Tier Architecture** เพื่อแยกหน้าที่ของระบบออกเป็นส่วนต่างๆ อย่างชัดเจน ได้แก่

- **Presentation Layer (Client Layer):** ประกอบด้วย Mobile Application (React Native + Expo) สำหรับผู้ใช้บริการฟิตเนส และ Web Application (Vite + React) สำหรับผู้ดูแลระบบ (Admin/Trainer) ทำหน้าที่เป็นส่วนติดต่อผู้ใช้และสื่อสารกับ Backend ผ่าน REST API
- **Application Layer (Backend):** พัฒนาด้วยภาษา **Go 1.24.2** + Gin Framework ออกแบบตามหลัก Clean Architecture ทำหน้าที่ประมวลผล business logic เช่น Workout Planner, Flex Substitute Algorithm และ Gamification
- **Data Layer:** ใช้ **PostgreSQL 18.2** เป็นฐานข้อมูลหลัก, **MinIO** สำหรับ object storage (รูปภาพ/วิดีโอ) และ **Redis 7** สำหรับ caching layer

ในส่วนของ Infrastructure ระบบถูก containerize ด้วย Docker และ deploy บน Proxmox VE พร้อม Cloudflare Tunnel เป็น layer กลางระหว่าง internet กับ backend server รองรับการขยายตัวเป็น Microservices ในอนาคต

3.2 ผู้ใช้งานและ Use Case ของระบบ

ระบบ GymMate มีผู้ใช้งานหลัก 3 กลุ่ม ได้แก่ Member, Trainer และ Admin

Use Case หลักของ Member

- สมัครสมาชิก / เข้าสู่ระบบ
- เลือกและดำเนินการโปรแกรมออกกำลังกาย
- บันทึกผลการออกกำลังกาย (Workout Log)
- ใช้ฟังก์ชัน Flex Substitute เพื่อเปลี่ยนท่า
- ดูแผนผังฟิตเนส (Floorplan) และค้นหาเครื่องออกกำลังกาย
- ติดตามความก้าวหน้า (Progress & Analytics)
- ดูอันดับ Leaderboard

Use Case หลักของ Admin/Trainer

- จัดการข้อมูลเครื่องออกกำลังกายและตำแหน่งใน Floorplan
- สร้างและแก้ไขโปรแกรมการออกกำลังกาย (Program Template)
- มอบหมายโปรแกรมให้ผู้ใช้ (Assign Program)
- จัดการข้อมูลท่าออกกำลังกาย (Exercise Library)
- ติดตาม Progress ของ Member ที่อยู่ในความดูแล
- จัดการสิทธิ์ผู้ใช้งาน (User Management)

3.3 User Flow

3.3.1 Authentication Flow

ผู้ใช้ทุกกลุ่มต้องผ่านขั้นตอน login โดยกรอก email และ password ระบบตรวจสอบด้วย bcrypt และออก JWT ผ่าน HTTP-only cookie เมื่อเข้าสู่ระบบสำเร็จ ระบบจะ redirect ไปยังหน้าหลักตาม role (Member □ Home, Admin/Trainer □ Dashboard)

3.3.2 Workout Plan Flow

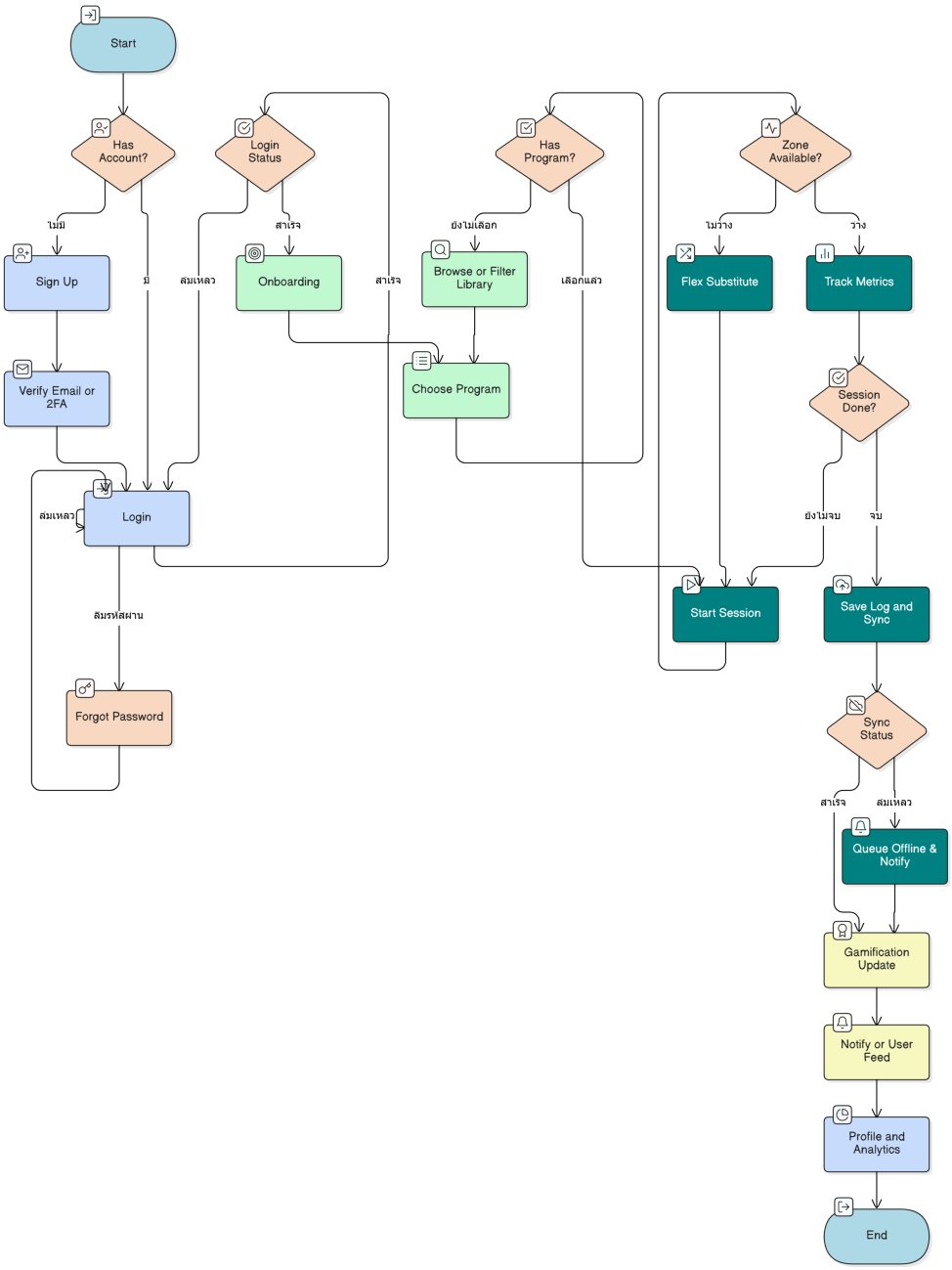
1. ผู้ใช้เลือกวันออกกำลังกายผ่าน Calendar ระบบตรวจสอบว่ามี Workout ที่ถูก schedule ไว้หรือไม่
2. ระหว่างการออกกำลังกาย ผู้ใช้ปรับน้ำหนักผ่าน Weight Picker และสามารถเปลี่ยนท่าผ่าน Flex Substitute Flow
3. เมื่อทำครบทุกท่า Slide to Confirm เพื่อยืนยัน ระบบบันทึก Workout Log และมอบ XP หากยกเลิกระบบบันทึกเป็น Missed Log

3.3.3 Flex Substitute Flow

1. ผู้ใช้กดไอคอน Substitute บน Exercise Card ระบบแสดง FlexSub Modal พร้อมดึงรายการท่าทางเลือกจากเซิร์ฟเวอร์ โดยคำนวณด้วย Cosine Similarity
2. เมื่อผู้ใช้เลือกท่าทางเลือก ระบบแทนที่ท่าเดิมใน Exercise List ทันทีโดยไม่กระทบ flow หลัก

3.3.4 Progress และ History Flow

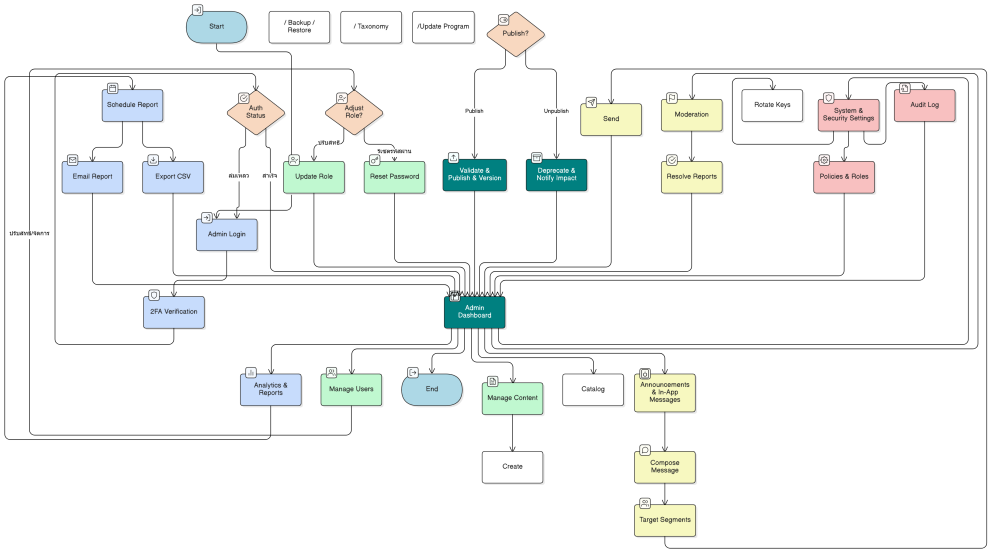
ระบบบันทึกผลการออกกำลังกายอัตโนมัติลงใน Progress/History โดยแสดงในรูปแบบ Bar Chart (ความถี่) และ Donut Chart (สัดส่วนกล้ามเนื้อ) พร้อมรายละเอียดของแต่ละ Session



รูปที่ 3.1: ภาพรวม User Flow ของระบบ GymMate ฝั่งผู้ใช้งาน (Member)

3.3.5 Floor Plan Flow

1. ระบบดึงข้อมูลอุปกรณ์ทั้งหมดและแปลงตำแหน่งจาก cm เป็น pixel แล้ว render บน Interactive Canvas
2. ผู้ใช้สำรวจผ่าน Pinch Zoom และ Pan กรองอุปกรณ์ด้วย Search Bar หรือ Type Chip
3. กดเลือกอุปกรณ์เพื่อดูรายละเอียดและสถานะ (Available / Out-of-Order)



รูปที่ 3.2: ภาพรวม User Flow ของระบบ GymMate ฝั่งผู้ดูแลระบบ (Admin/Trainer)

3.3.6 Leaderboard Flow

ผู้ใช้เลือกประเภทการจัดอันดับ (Volume, Streak, Program) และช่วงเวลา (Weekly, Monthly, All Time) ระบบแสดงอันดับ 1-3 ในรูปแบบ Podium และอันดับถัดไปในรูปแบบ List

3.3.7 Trainer Portal Flow

Trainer login เข้าสู่ Trainer Portal จัดการข้อมูลหลักของระบบ ได้แก่ Program Template, Exercise Library และ Equipment พร้อมกำหนด Program ให้ผู้ใช้ ระบบสร้างตารางออกกำลังกายใน Calendar อัตโนมัติ และ Trainer ติดตาม Progress ของ Member แบบ real-time

3.4 การออกแบบฐานข้อมูล (ERD)

ระบบ GymMate ใช้ PostgreSQL เป็น relational database หลัก [10] โดยออกแบบ schema แบ่งออกเป็น 8 domain ตามหน้าที่ของข้อมูล

3.4.1 Users & Authentication

ตาราง `users` เป็นศูนย์กลางของระบบ เก็บข้อมูลผู้ใช้ทุกประเภทในตารางเดียว (Member, Trainer, Admin) แยกบทบาทผ่าน `role` field และกำหนดสถานะบัญชีผ่าน `status` field โดยมีการอ้างอิงตัวเองผ่าน `trainer_id` เพื่อเชื่อมความสัมพันธ์ Trainer-Member นอกจากนี้เก็บข้อมูลสุขภาพพื้นฐาน เช่น `height_cm`, `fitness_level` และ `fitness_goal`

3.4.2 Exercise Library

Domain นี้ประกอบด้วย 6 ตารางที่ทำงานร่วมกัน

- `exercises` — เก็บข้อมูล exercise แต่ละท่า เช่น movement_type, movement_pattern และระดับความยาก
- `exercise_muscles` — junction table เชื่อม exercise กับกล้ามเนื้อ มี `involvement_type` (primary/secondary/stabilizer) และ `activation_percentage` ใช้สร้าง muscle activation vector สำหรับ Flex Substitute
- `computed_exercise_substitutes` — เก็บผลลัพธ์ Cosine Similarity ที่คำนวณไว้ล่วงหน้าแบบ bidirectional พร้อม UNIQUE constraint บน `(original_exercise_id, substitute_exerc`

3.4.3 Programs & Sessions

- `programs` — มี flag `is_template` แยก Preset Program จาก Assigned Program
- `program_sessions` — กำหนด session รายวัน ระบุด้วย `day_number` และ `workout_split` (Push/Pull/Legs)
- `session_exercises` — junction table เชื่อม session กับ exercise พร้อม `sets`, `reps`, `weight` และ `order_sequence`

3.4.4 User Programs & Scheduling (Snapshot Pattern)

Domain นี้ใช้ **Snapshot Pattern** เป็น design หลัก เพื่อให้การแก้ไข Program template ไม่กระทบต่อ workout ที่ถูก schedule ไปแล้ว

- `user_programs` — instance ของ program ที่ assign ให้ผู้ใช้แต่ละคน ติดตาม `current_week`, `current_day` และ `status`
- `scheduled_workouts` — สร้างอัตโนมัติเมื่อ assign program แต่ละแถวแทน 1 session ในปฏิทิน
- `scheduled_workout_exercises` — snapshot copy ของ `session_exercises` ณ เวลาสร้าง ทำให้แผนการออกกำลังกายไม่เปลี่ยนแม้ Trainer แก้ไข template

3.4.5 Workout Logging

- `workout_logs` — บันทึกผลการออกกำลังกายจริง เชื่อมกับ `scheduled_workouts` แบบ optional รองรับทั้ง planned และ free workout
- `log_exercises` — เก็บผลจริงในแต่ละท่า เช่น `sets_completed`, `reps_completed`, `weight_used` และ `rpe_rating`

3.4.6 Floorplan

- `floorplans` — กำหนด canvas ของแผนผังยิม มี `canvas_width`, `canvas_height` และ `grid_size`

- `equipment_instances` — เชื่อม equipment กับ floorplan พร้อมตำแหน่งจริง (`position_x`, `position_y`, `rotation`) และขนาด
- `floorplan_walls` — เก็บข้อมูลผนังและโครงสร้างยิมผ่าน `start_x`, `start_y`, `end_x`, `end_y`

3.4.7 Gamification

- `user_gamification_profiles` — เก็บสถานะ gamification ปัจจุบัน เช่น `total_xp`, `current_level`, `current_streak`, `longest_streak`
- `badges` — กำหนด badge แต่ละประเภทพร้อม `trigger_type`, `trigger_value` และ `xp_reward`
- `xp_events` — บันทึกประวัติการรับ XP ทุกรายการ

3.4.8 Tracking

- `muscle_trackings` — บันทึก `weekly_volume` และ `recovery_status` รายสัปดาห์
- `user_weight_logs` — บันทึกน้ำหนักตัวผู้ใช้พร้อม timestamp

Key Design Decisions:

1. **Snapshot Pattern** — ป้องกันการแก้ไข template กระทบ scheduled workout
2. **Template vs Instance** — `programs.is_template` แยก Preset Program จาก Assigned Program
3. **Bidirectional Substitutes** — query หา substitute โดยใช้เพียง `WHERE original_exercise_id = ?`
4. **Soft Delete** — ทุกตารางใช้ `deleted_at` (GORM convention) แทนการลบจริง

3.5 การออกแบบ API และระบบ Authentication

3.5.1 การออกแบบ RESTful API

GymMate ออกแบบ API ในรูปแบบ RESTful โดยกำหนดหลักการดังนี้

- **Versioning** — endpoint ทุกตัวขึ้นต้นด้วย `/api/v1/`
- **Resource Naming** — ใช้ noun พหูพจน์ เช่น `/exercises`, `/programs`, `/users`
- **API Documentation** — Swagger UI ที่ port 8081 สร้างจาก `openapi.yaml`

Method	การใช้งาน	ตัวอย่าง
GET	ดึงข้อมูล	`GET /api/v1/exercises`
POST	สร้างข้อมูลใหม่	`POST /api/v1/exercises`
PUT	แก้ไขข้อมูลทั้งหมด	`PUT /api/v1/programs/{id}`
PATCH	แก้ไขข้อมูลบางส่วน	`PATCH /api/v1/users/{id}`
DELETE	ลบข้อมูล	`DELETE /api/v1/equipment/{id}`

ตารางที่ 3.1: HTTP Methods ที่ใช้ใน GymMate API

3.5.2 ระบบ Authentication

GymMate ใช้ JWT HS256 เป็น mechanism หลักสำหรับยืนยันตัวตน มีขั้นตอนดังนี้

1. ผู้ใช้ส่ง `email` และ `password` ผ่าน `POST /api/v1/auth/login`
2. Server ตรวจสอบ email และเปรียบเทียบ password ด้วย bcrypt
3. Server สร้าง JWT ลงนามด้วย HS256 โดย payload บรรจุ `user_id`, `role` และ expiration time
4. Server ส่ง JWT กลับในรูป HTTP-only cookie ป้องกัน XSS
5. ใน request ถัดไป client แนบ cookie อัตโนมัติ server validate ลายเซ็นและดึง `user_id` และ `role` โดยไม่ต้อง query database

3.5.3 ระบบ Authorization (RBAC)

GymMate ใช้ **Role-Based Access Control (RBAC)** โดยแบ่ง role ออกเป็น 3 ระดับ ตรวจสอบใน Middleware layer ของ Gin ก่อนที่ request จะถึง handler จริง

Role	สิทธิ์การเข้าถึง
Member	ข้อมูลและฟีเจอร์ของตัวเองเท่านั้น
Trainer	จัดการ program และติดตาม member ในความดูแล, แก้ไข exercise library
Admin	เข้าถึงได้ทุกส่วนของระบบ รวมถึง CMS Dashboard และ user management

ตารางที่ 3.2: RBAC Role ในระบบ GymMate

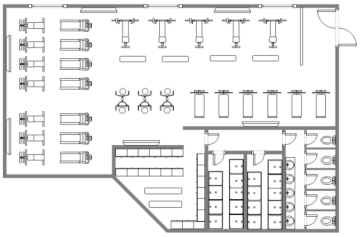
3.6 การออกแบบและพัฒนาฟีเจอร์หลัก

3.6.1 Member UI (Mobile App)

Mobile Application พัฒนาด้วย React Native + Expo ออกแบบสำหรับผู้ใช้บริการฟิตเนส ประกอบด้วยหน้าจอหลักดังนี้

- **Home** — แสดงสถิติ Level, XP, Streak และ Badge System กระตุ้นแรงจูงใจ

- **Search Exercises** — ค้นหาท่าออกกำลังกายด้วย Search Bar และ Filter Chip (กล้ามเนื้อ) พร้อม Debounce 300 ms แสดงรายละเอียดผ่าน Detail Modal (Muscle Activation, อุปกรณ์ที่ใช้)
- **Workout Plan** — Calendar เลือกวัน, Exercise List พร้อม Weight Picker, ฟีเจอร์ Flex Substitute และ Slide to Confirm
- **Flex Substitute** — แสดงท่าทดแทนพร้อมค่า Similarity Score เมื่อเลือกแล้วอัปเดต Exercise List ทันที
- **Floor Plan** — Interactive Canvas แสดงตำแหน่งเครื่อง รองรับ Pinch Zoom และ Pan กรองด้วย Search Bar/Type Chip แสดงสถานะ Available/Out-of-Order
- **Progress** — Bar Chart (ความถี่) และ Donut Chart (สัดส่วนกล้ามเนื้อ) พร้อมรายละเอียด Session
- **Leaderboard** — จัดอันดับ Volume/Streak/Program แบบ Weekly/Monthly/All Time แสดงผล Podium อันดับ 1-3



รูปที่ 3.3: ตัวอย่างหน้า Floor Plan ของ GymMate แสดงตำแหน่งอุปกรณ์ภายในยิม

3.6.2 Admin UI (Web Application)

Web CMS Dashboard พัฒนาด้วย Vite + React + Tailwind CSS + Framer Motion ออกแบบสำหรับ desktop เน้น information density รองรับ workflow ที่ซับซ้อน ประกอบด้วยหน้าจอหลักดังนี้

- **Dashboard Overview** — สรุปสถิติการใช้งาน จำนวนผู้ใช้ โปรแกรม และสถานะระบบโดยรวม
- **Floorplan Editor** — จัดการผังยิมแบบ Interactive Admin ลาก-วาง เพิ่ม แก้ไข หรือลบตำแหน่งเครื่องออกกำลังกาย พร้อมโหมดแก้ไข (Edit Mode) และการสร้าง Equipment Instance
- **Program Builder** — สร้าง Program Template กำหนดท่า จำนวนเซต จำนวนครั้ง และ Assign Program ให้ผู้ใช้
- **Exercise Management** — เพิ่ม/แก้ไข exercise พร้อม upload รูปภาพ/วิดีโอ และแสดงผลการคำนวณ Cosine Similarity
- **Muscle Atlas** — แสดงข้อมูลกล้ามเนื้อในร่างกาย ช่วยให้เข้าใจผลต่อกล้ามเนื้อแต่ละส่วน
- **User Management** — เพิ่ม/แก้ไข/ลบผู้ใช้งาน กำหนด Role (Admin, Trainer, Member)

3.6.3 Trainer UI (Web Application)

- **Trainer Dashboard** — แสดงภาพรวม Member ที่อยู่ในความดูแล รายชื่อสมาชิก จำนวนครั้งออกกำลังกายล่าสุด
- **Member Detail** — ข้อมูลเฉพาะบุคคล เช่น ระดับการฝึก โปรแกรมที่ assign และสถิติล่าสุด
- **Daily Program Edit** — แก้ไขโปรแกรมรายวันให้ลูกเทรน
- **Workout History & Progress** — ติดตาม log และ progress ของ Member แต่ละคน

3.7 การออกแบบ UI/UX

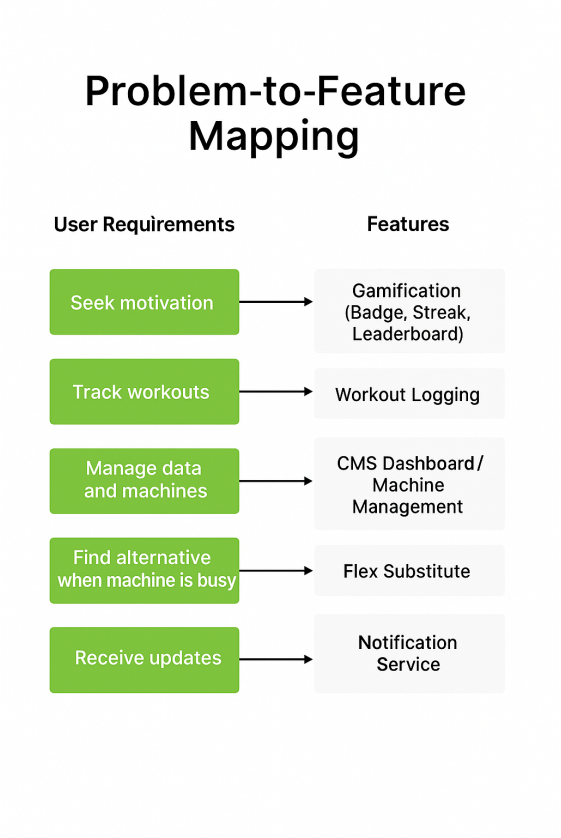
การออกแบบ UI/UX ของ GymMate ยึดหลักการดังนี้

Mobile App (Member)

ออกแบบตาม **Mobile-first** เน้น simplicity และ intuitive interaction รองรับการใช้งานด้วยมือเดียว ใช้สี primary และ accent สอดคล้องกับธีม fitness/energy พร้อม animation เพื่อ feedback ที่ชัดเจน

Web CMS Dashboard (Admin & Trainer)

ออกแบบสำหรับ desktop browser เน้น **information density** แสดงข้อมูลได้มากในหน้าเดียว ใช้ Tailwind CSS สำหรับ utility-first styling และ Framer Motion สำหรับ transition เพื่อประสบการณ์ที่ลื่นไหล



รูปที่ 3.4: Problem-to-Feature Mapping แสดงความสัมพันธ์ระหว่างปัญหาและฟีเจอร์ของ GymMate

บทที่ 4
การทดสอบระบบ

4.1 Unit Testing

Unit Testing คือการทดสอบหน่วยย่อยที่สุดของโปรแกรมแบบแยกส่วน โดยแต่ละ test case ตรวจสอบว่า function หรือ method หนึ่งทำงานถูกต้องตาม specification ที่กำหนดไว้ภายใต้เงื่อนไขต่างๆ โดยไม่พึ่งพา component ภายนอก เช่น database หรือ network [11]

GymMate เขียน unit test สำหรับ **Usecase Layer** ซึ่งเป็นชั้นที่บรรจุ business logic หลักของระบบ การที่ทดสอบได้โดยไม่ต้องพึ่ง database จริงเป็นผลโดยตรงจากการออกแบบตาม Clean Architecture ที่ Usecase Layer เข้าถึงข้อมูลผ่าน Repository interface เท่านั้น ทำให้สามารถแทนที่ด้วย mock object ระหว่างการทดสอบได้

4.1.1 โครงสร้างของ Test

แต่ละ feature module ภายใต้ `feature/{feature_name}/usecase/` ประกอบด้วยไฟล์ 3 ไฟล์

- **`usecase.go`** — implement business logic จริงของ feature รับ Repository interface ผ่าน constructor (Dependency Injection)
- **`mock_test.go`** — สร้าง mock struct ที่ implement Repository interface เดียวกัน แต่ไม่ติดต่อ database จริง กำหนดพฤติกรรมได้ในแต่ละ test case
- **`usecase_test.go`** — เขียน test case โดยสร้าง usecase instance ที่รับ mock repository แทน repository จริง แล้วตรวจสอบผลลัพธ์ที่ได้

4.1.2 หลักการทดสอบ (Arrange-Act-Assert)

test case แต่ละตัวออกแบบตามโครงสร้าง **Arrange-Act-Assert**

1. **Arrange** — กำหนดพฤติกรรมของ mock และสร้าง input data
2. **Act** — เรียก function ที่ต้องการทดสอบด้วย input ที่เตรียมไว้
3. **Assert** — ตรวจสอบว่า output และ error ที่ได้ตรงตาม expected result

แต่ละ function ทดสอบหลาย scenario ครอบคลุมทั้ง happy path และ error path

4.1.3 การรัน Test และ CI/CD

test ทั้งหมดถูกรันอัตโนมัติใน **CI pipeline ผ่าน GitHub Actions** ทุกครั้งที่มีการ push code ไปยัง repository โดยใช้คำสั่ง `go test ./...`

หาก test ใดล้มเหลว CI pipeline จะหยุดทำงานและ block การ merge code เข้า main branch ทำให้มั่นใจได้ว่า business logic ที่ deploy ขึ้น production ผ่านการตรวจสอบแล้วทุกครั้ง

Scenario	ตัวอย่าง (Equipment Usecase)
Happy path	GetEquipment คืน equipment ที่ถูกต้องเมื่อ ID มีอยู่
Not found	GetEquipment คืน error เมื่อ ID ไม่มีในระบบ
Duplicate	CreateEquipment คืน error เมื่อชื่อซ้ำ
Invalid input	CreateEquipment คืน error เมื่อ input ไม่ผ่าน validation
Repository error	คืน error ที่เหมาะสมเมื่อ database มีปัญหา

ตารางที่ 4.1: ตัวอย่าง Test Scenarios ในระบบ GymMate

4.2 การทดสอบการใช้งานจริง (Usability Testing / UAT)

การทดสอบนี้มีวัตถุประสงค์เพื่อประเมินประสบการณ์ผู้ใช้งาน (User Experience) และความง่ายในการใช้งานของระบบ

4.2.1 กลุ่มผู้ทดสอบและรูปแบบการเก็บข้อมูล

- กลุ่มผู้ทดสอบ: นักศึกษาที่ใช้งานฟิตเนสและผู้ใช้งานทั่วไป จำนวน 11 คน
- รูปแบบการเก็บข้อมูล: แบบสอบถาม (Google Form) และการสังเกตพฤติกรรมผู้ใช้งาน

4.2.2 หัวข้อการประเมิน

- ความง่ายในการใช้งาน (Ease of Use)
- ความเข้าใจของ UI
- ความสะดวกในการวางแผนการออกกำลังกาย
- ความพึงพอใจโดยรวม
- ความน่าจะเป็นในการใช้งานต่อ
- ความรู้สึกต่อดีไซน์ของแอป

4.2.3 ผลการทดสอบ

จากการเก็บข้อมูลผู้ทดสอบ 11 คน ผลการประเมินในภาพรวมแสดงให้เห็นว่า ผู้ทดสอบส่วนใหญ่พึงพอใจกับการออกแบบ UI/UX และมองว่าแอปใช้งานง่าย ฟีเจอร์ตอบโจทย์ความต้องการ และมีแนวโน้มจะใช้งานต่อในอนาคต

นอกจากนี้ในระหว่างการทดสอบ พบ bug 1 รายการ ได้แก่ การลืมใส่ Role middleware ที่ endpoint หนึ่ง ซึ่งได้รับการแก้ไขเรียบร้อยแล้ว

4.3 การประเมินผลตามวัตถุประสงค์ของโครงการ

จากวัตถุประสงค์ของโครงงานที่ตั้งไว้ในบทที่ 1 สามารถประเมินผลได้ดังตาราง 4.2

วัตถุประสงค์	ผลลัพธ์
พัฒนาระบบวางแผนการออกกำลังกาย (Workout Program Planner)	สามารถสร้างและใช้งานโปรแกรมทั้ง Preset Plan และ Custom Plan ได้จริง ผู้ใช้สามารถเลือก Goal & Approach และดำเนินการออกกำลังกายตามปฏิทินได้
เพิ่มประสิทธิภาพด้วยระบบ Flex Substitute	ระบบสามารถแนะนำท่าทดแทนโดยใช้ Cosine Similarity เมื่ออุปกรณ์ไม่ว่างหรือผู้ใช้ต้องการเปลี่ยนท่า
จัดการฟิตเนสผ่าน CMS Dashboard	Admin สามารถจัดการข้อมูลเครื่อง Floorplan Program และผู้ใช้ได้ผ่าน Web CMS
สร้างแรงจูงใจในการออกกำลังกาย	ระบบ Gamification (XP, Badge, Streak, Leaderboard) ช่วยเพิ่ม engagement และการใช้งานอย่างต่อเนื่อง

ตารางที่ 4.2: ผลการประเมินตามวัตถุประสงค์ของโครงงาน GymMate

บทที่ 5
สรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

โครงงาน GymMate ได้ออกแบบและพัฒนาระบบต้นแบบสำหรับสนับสนุนการออกกำลังกายของนักศึกษามหาวิทยาลัยเชียงใหม่ในหลายมิติ โดยมีผลลัพธ์สำคัญดังนี้

- **ระบบ Workout Program Planner** — รองรับทั้ง Preset Plan และ Custom Plan พร้อมแนวทาง Goal & Approach เช่น Push–Pull–Legs, Upper–Lower และ Total Body ผู้ใช้สามารถดำเนินการออกกำลังกายตามปฏิทินและบันทึกผลลัพธ์ได้จริง
- **ระบบ Flex Substitute** — ใช้อัลกอริทึม Cosine Similarity เพื่อแนะนำท่าทดแทนที่เน้นกล้ามเนื้อกลุ่มเดิม โดยผลลัพธ์ถูกคำนวณและเก็บไว้ล่วงหน้า (pre-computed) เพื่อประสิทธิภาพสูงสุด
- **Interactive Floorplan** — แผนผังยิมแบบ Interactive ให้ผู้ใช้สำรวจตำแหน่งเครื่องและสถานะได้ พร้อมระบบ Editor ฝั่ง Admin
- **Gamification Layer** — ระบบ XP, Level, Badge, Streak และ Leaderboard ช่วยเพิ่มแรงจูงใจและความต่อเนื่องในการออกกำลังกาย
- **Admin & Trainer Dashboard** — Web CMS ครบถ้วนสำหรับจัดการ exercise, program, floorplan และผู้ใช้งาน
- **Clean Architecture** — Backend ออกแบบตาม Clean Architecture ด้วย Go + Gin + GORM + PostgreSQL ทำให้ unit test ได้โดยไม่ต้องพึ่ง database จริง และผ่าน CI pipeline อัตโนมัติ

5.2 ปัญหาและอุปสรรคที่พบ

- **ข้อจำกัดด้านเวลา** — ระยะเวลาในการพัฒนาไม่เพียงพอสำหรับการพัฒนาระบบให้สมบูรณ์และทดสอบกับผู้ใช้งานจริงในวงกว้าง
- **ขาดข้อมูลจากการใช้งานจริง (Real-world Data)** — ระบบบางส่วน เช่น Flex Substitute และ Gamification ยังไม่สามารถปรับให้เหมาะสมกับผู้ใช้จริงได้อย่างเต็มที่ เนื่องจากไม่มีข้อมูลพฤติกรรมผู้ใช้จริง
- **ข้อจำกัดด้านฮาร์ดแวร์** — ระบบไม่ได้เชื่อมต่อกับอุปกรณ์ฟิตเนสจริง ทำให้ไม่สามารถเก็บข้อมูลแบบ real-time เช่น การใช้งานเครื่องหรือ biometric data
- **ความซับซ้อนของการออกแบบระบบ** — การออกแบบระบบที่มีหลายฟีเจอร์ เช่น Planner, Gamification และ Floorplan ทำให้ต้องใช้เวลาในการออกแบบโครงสร้างระบบและฐานข้อมูลค่อนข้างมาก
- **ข้อจำกัดด้านการทดสอบ** — ยังไม่มีการทดสอบในระดับ production หรือ load test อย่างเต็มรูปแบบ

5.3 ข้อเสนอแนะและแนวทางพัฒนาต่อ

เพื่อพัฒนา GymMate ให้สามารถนำไปใช้งานจริงได้อย่างมีประสิทธิภาพ ควรมีการปรับปรุงในด้านต่างๆ ดังนี้

- **พัฒนา Personalization และ AI Recommendation** — ใช้ Machine Learning เพื่อแนะนำโปรแกรมการออกกำลังกายที่เหมาะสมกับผู้ใช้แต่ละคนจากพฤติกรรมการใช้งานจริง
- **เชื่อมต่อกับอุปกรณ์ IoT หรือ Fitness Tracker** — เช่น Smartwatch หรือเครื่องออกกำลังกาย เพื่อเก็บข้อมูลแบบ real-time และเพิ่มความแม่นยำของระบบ
- **ปรับปรุงระบบ Flex Substitute** — ให้สามารถแนะนำท่าทางได้แม่นยำมากขึ้น โดยอิงจากข้อมูลผู้ใช้จริงและบริบทของยิม เช่น ความหนาแน่นของผู้ใช้งาน
- **พัฒนา Gamification ให้หลากหลายขึ้น** — เช่น ระบบ Challenge, Group Competition หรือ Seasonal Event เพื่อเพิ่ม engagement
- **รองรับระบบแบบ Real-time** — เช่น การแสดงสถานะเครื่องว่าง/ไม่ว่างแบบ real-time ผ่าน WebSocket หรือ Server-Sent Events
- **ขยายระบบเป็น Microservices Architecture** — เพื่อรองรับการใช้งานในระดับใหญ่และเพิ่ม scalability
- **ปรับปรุงด้าน UI/UX อย่างต่อเนื่อง** — โดยอิงจาก feedback ของผู้ใช้งานจริง เพื่อให้ใช้งานง่ายและตอบโจทย์มากขึ้น

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่าโครงงาน GymMate จะเป็นเครื่องมือที่ช่วยส่งเสริมการออกกำลังกายให้มีประสิทธิภาพมากยิ่งขึ้น และสามารถนำไปต่อยอดให้เกิดประโยชน์ต่อผู้ใช้งานในอนาคต

บรรณานุกรม

[1] American College of Sports Medicine. *ACSM's Guidelines for Exercise Testing and Prescription*. Wolters Kluwer, Philadelphia, PA, 11 edition, 2021.

[2] Albert Bandura. Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, 84(2):191–215, 1977.

[3] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: Defining "gamification". In *Proceedings of the 15th International Academic MindTrek Conference*, pages 9–15, 2011.

[4] Expo. Expo documentation. https://docs.expo.dev/, 2024.

[5] Facebook Inc. React native documentation. https://reactnative.dev/, 2024.

[6] Golang. The go programming language documentation. https://go.dev/, 2024.

[7] GORM. Gorm documentation. https://gorm.io/, 2024.

[8] Robert C. Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall, Boston, MA, 2017.

[9] OWASP Foundation. Owasp top ten. https://owasp.org/www-project-top-ten/, 2021.

[10] PostgreSQL Global Development Group. Postgresql documentation. https://www.postgresql.org/docs/, 2024.

[11] Roger S. Pressman and Bruce R. Maxim. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, New York, NY, 9 edition, 2019.

[12] React. React documentation. https://react.dev/, 2024.

[13] U.S. Department of Health and Human Services. Physical activity guidelines for americans. https://health.gov/, 2018.

[14] Vite. Vite documentation. https://vitejs.dev/, 2024.