



แอปพลิเคชันนุมอร่า: ระบบจัดการการเงินส่วนบุคคล

โดย

นาย ปิณณทัต มณีวงษ์

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตร์บัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2568

ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

แอปพลิเคชันนอมอรา: ระบบจัดการการเงินส่วนบุคคล

โดย

นาย ปณณทัต มณีวงษ์

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตร์บัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

ปีการศึกษา 2568

ลิขสิทธิ์ของมหาวิทยาลัยธรรมศาสตร์

NUMORA APPLICATION: PERSONAL FINANCIAL MANAGEMENT
SYSTEM

BY

MR. PUNNATUT MANEEWONG

A FINAL-YEAR PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE

COMPUTER SCIENCE

FACULTY OF SCIENCE AND TECHNOLOGY

THAMMASAT UNIVERSITY

ACADEMIC YEAR 2025

COPYRIGHT OF THAMMASAT UNIVERSITY

มหาวิทยาลัยธรรมศาสตร์
คณะวิทยาศาสตร์และเทคโนโลยี

รายงานโครงการพิเศษ

ของ

นาย ปณิทัต มณีวงศ์

เรื่อง

แอปพลิเคชันนุมอร่า: ระบบจัดการการเงินส่วนบุคคล

ได้รับการตรวจสอบและอนุมัติ ให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
เมื่อ วันที่ 19 ธันวาคม พ.ศ. 2568

อาจารย์ที่ปรึกษา



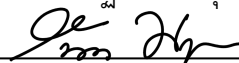
(ผศ.ดร. ทรงศักดิ์ ร่องวิริยะพานิช)

กรรมการสอบโครงการพิเศษ



(ผศ.ดร. ศาตนาฏ กิจศิริานุวัตร)

กรรมการสอบโครงการพิเศษ



(ผศ.ดร. วณิดา พงษ์วิทยา)

มหาวิทยาลัยธรรมศาสตร์
คณะวิทยาศาสตร์และเทคโนโลยี

รายงานโครงการพิเศษ

ของ

นาย ปณทัต มณีวงศ์

เรื่อง

แอปพลิเคชันนุมอรา: ระบบจัดการการเงินส่วนบุคคล

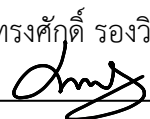
ได้รับการตรวจสอบและอนุมัติ ให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
เมื่อ วันที่ 19 ธันวาคม พ.ศ. 2568

อาจารย์ที่ปรึกษา



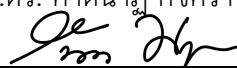
(ผศ.ดร. ทรงศักดิ์ ร่องวิริยะพานิช)

กรรมการสอบโครงการพิเศษ



(ผศ.ดร. ศาตนาฏ กิจศิริานุวัตร)

กรรมการสอบโครงการพิเศษ



(ผศ.ดร. วนิตา พงษ์อิทธิยา)

หัวข้อโครงการพิเศษ

ชื่อผู้เขียน

ชื่อปริญญา

สาขาวิชา/คณะ/มหาวิทยาลัย

อาจารย์ที่ปรึกษาโครงการพิเศษ

ปีการศึกษา

แอปพลิเคชันนุโมรา: ระบบจัดการการเงินส่วนบุคคล

นาย ปณณทัต มณีวงษ์

วิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยธรรมศาสตร์

ผศ.ดร. ทรงศักดิ์ ร่องวิริยะพานิช

2568

บทคัดย่อ

รายงานฉบับนี้นำเสนอ Numora แอปพลิเคชันบนระบบปฏิบัติการ Android สำหรับการจัดการการเงินส่วนบุคคล โดยออกแบบมาเพื่อให้เหมาะสมกับพฤติกรรมการเงินของผู้ใช้ชาวไทย มุ่งเน้นให้ผู้ใช้จ่ายรายรับ-รายจ่ายของตนเองได้ง่ายขึ้น ควบคุมพฤติกรรมการใช้เงิน และสร้างวินัยทางการเงินผ่านการแสดงผลเชิงภาพและคำแนะนำที่ชัดเจน

Numora รองรับการบันทึกรายรับ-รายจ่าย การจัดหมวดหมู่ การติดตามธุรกรรม การตั้งรายการซ้ำ (Recurring Items) การตั้งงบประมาณ การตั้งเป้าหมายทางการเงิน รวมถึงระบบ 50-30-20 แบบปรับอัตราส่วนตามพฤติกรรมจริงของผู้ใช้ (Dynamic 50-30-20 Rule) ซึ่งช่วยให้ระบบแนะนำการออม การใช้เงินจำเป็น และค่าใช้จ่ายฟุ่มเฟือยได้อย่างยืดหยุ่น

ระบบถูกพัฒนาด้วยภาษา Java ร่วมกับฐานข้อมูล SQLite และสถาปัตยกรรมแบบ MVC โดยออกแบบให้แอปพลิเคชันสามารถทำงานแบบออฟไลน์ทั้งหมด เพื่อลดการพึ่งพาการเชื่อมต่อกับฐานข้อมูลภายนอก ช่วยเพิ่มประสิทธิภาพด้านเวลาในการเข้าถึงข้อมูล และลดความซับซ้อนของระบบ สถาปัตยกรรม MVC ถูกนำมาใช้เพื่อแยกหน้าที่ของส่วนแสดงผล ส่วนควบคุม และส่วนจัดการข้อมูลอย่างชัดเจน ทำให้ระบบสามารถดูแลรักษา แก้ไข และพัฒนาเพิ่มเติมได้ง่ายขึ้น อีกทั้งยังช่วยลดการเข้าถึงข้อมูลโดยตรงจากส่วนแสดงผล เพิ่มความเป็นระเบียบและความปลอดภัยของโครงสร้างระบบโดยรวม

คำสำคัญ: บริหารการเงิน, แอปพลิเคชัน Android, การจัดการรายรับรายจ่าย, งบประมาณ, 50-30-20, ออฟไลน์

Thesis Title	NUMORA APPLICATION: PERSONAL FINANCIAL MANAGEMENT
Author	MR. Punnatut Maneewong
Degree	Bachelor of Science
Major Field/Faculty/University	Computer Science Faculty of Science and Technology Thammasat University
Project Advisor	Asst.Prof. Songsakdi Rongviriyanish, Ph.D.
Academic Years	2025

ABSTRACT

This report presents Numora, an Android application for personal financial management, designed to align with the financial behavior of Thai users. The application aims to help users better understand their income and expenses, control spending behavior, and build financial discipline through clear visualizations and straightforward recommendations.

Numora supports income and expense recording, transaction categorization, transaction tagging, recurring items, budget planning, and financial goal setting. It also incorporates a Dynamic 50-30-20 Rule, which adjusts allocation ratios based on users' actual spending behavior, allowing the system to provide flexible recommendations for savings, essential expenses, and discretionary spending.

The system is developed using the Java programming language in conjunction with an SQLite database and follows the Model–View–Controller (MVC) architectural pattern. The application is designed to operate entirely offline, reducing dependency on external database connections, improving data access performance, and minimizing system complexity. The MVC architecture is applied to clearly separate the presentation, control, and data management components, making the system easier to maintain, modify, and extend. This separation also limits direct access to the data layer from the presentation layer, contributing to a more structured and secure overall system design.

Keywords: Personal finance, Android application, budgeting, offline-first, financial behavior.

กิตติกรรมประกาศ

ข้าพเจ้าขอขอบพระคุณ ผศ.ดร.ทรงศักดิ์ รองวิริยะพานิช อาจารย์ที่ปรึกษาโครงการ
ที่ให้คำแนะนำและความช่วยเหลือตลอดระยะเวลาการทำโครงการ รวมถึงคณะกรรมการสอบทุกท่าน
ที่กรุณาให้ข้อเสนอแนะอย่างมีประโยชน์

นอกจากนี้ ขอขอบคุณเพื่อน ๆ และครอบครัวที่ให้กำลังใจและสนับสนุนอย่างเต็มที่
จนโครงการฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยความเคารพอย่างสูง

นาย ปณณทัต มณีวงษ์

สารบัญ

	หน้า
บทคัดย่อ	2
ABSTRACT	4
กิตติกรรมประกาศ	5
สารบัญ	6
สารบัญตาราง	9
สารบัญภาพ	10
รายการสัญลักษณ์และคำย่อ	11
บทที่ 1 บทนำ	12
1.1 ความเป็นมาและความสำคัญของโครงการ	12
1.1.1 สถานการณ์การเงินส่วนบุคคลในประเทศไทย	12
1.1.2 ข้อจำกัดของแอปพลิเคชันการเงินส่วนบุคคลที่มีอยู่ในปัจจุบัน	12
1.2 วัตถุประสงค์	13
1.3 ขอบเขตของโครงการ	13
1.4 ประโยชน์ของโครงการ	14
1.5 ข้อจำกัดของโครงการ	14
บทที่ 2 วรรณกรรมและงานวิจัยที่เกี่ยวข้อง	15
2.1 แนวคิดทฤษฎีที่เกี่ยวข้อง	15
2.1.1 การจัดการการเงินส่วนบุคคล (Personal Finance Management)	15

2.1.2	การออกแบบส่วนต่อประสานกับผู้ใช้สำหรับแอปพลิเคชันการเงิน (UI/UX Design for Financial Applications)	15
2.2	งานวิจัยที่เกี่ยวข้อง	16
2.2.1	พฤติกรรมการออมและการใช้จ่ายของคนไทย	16
2.2.2	การเปรียบเทียบแอปพลิเคชันจัดการการเงินส่วนบุคคล	17
บทที่ 3	วิธีการวิจัย	18
3.1	ภาพรวมของโครงการ	18
3.1.1	สถาปัตยกรรมระบบ	18
3.1.2	โมดูลหลักของระบบ	19
3.1.3	Use Case Specification	21
3.1.4	ตัวอย่างหน้าจอ UI	27
3.2	การวิเคราะห์ขอบเขตและความต้องการของระบบ	32
3.2.1	ความต้องการด้านฟังก์ชันการทำงาน (Functional Requirements)	32
3.2.2	ความต้องการที่ไม่ใช่ฟังก์ชันการทำงาน (Non-functional Requirements)	32
3.3	การดำเนินงาน	33
3.3.1	ระยะที่ 1: การวิเคราะห์ความต้องการ (Requirement Analysis)	33
3.3.2	ระยะที่ 2: การออกแบบระบบ (System Design)	33
3.3.3	ระยะที่ 3: การพัฒนาและเขียนโปรแกรม (Implementation)	34
3.3.4	ระยะที่ 4: การทดสอบระบบ (System Testing)	35
3.3.5	ระยะที่ 5: การประเมินผลและปรับปรุง (Evaluation & Refinement)	35
บทที่ 4	ผลการดำเนินงาน	37
4.1	ผลการพัฒนาระบบ	37
4.1.1	ระบบบันทึกรายรับ-รายจ่าย	37

	8
4.1.2 ระบบหมวดหมู่และแท็ก	38
4.1.3 ระบบวิเคราะห์ตามกฎหมาย Dynamic 50-30-20	38
4.2 ผลการทดสอบระบบ (System Testing)	38
4.3 ตัวอย่างหน้าจอที่พัฒนาเสร็จ	39
4.4 สรุปผลการดำเนินงาน	39
บทที่ 5 สรุป	40
5.1 สรุปผลการดำเนินงาน	40
5.2 อภิปรายผล	41
5.3 ข้อเสนอแนะสำหรับการพัฒนาต่อยอด	42
รายการอ้างอิง	43
ภาคผนวก	46
ภาคผนวก ก. แผนผังฐานข้อมูล (ER Diagram) และโครงสร้างตารางของระบบ Numora	46
ก.1 แผนภาพ ER Diagram ของระบบ	46
ก.2 รายละเอียดโครงสร้างตารางฐานข้อมูล	47
ก.3 สรุปภาพรวมฐานข้อมูล	50

สารบัญตาราง

	หน้า
ตารางที่ 2.1 การเปรียบเทียบแอปพลิเคชันจัดการการเงิน	6
ตารางที่ 3.1 Use Case Specification ของบันทึกรายรับ-รายจ่าย	10
ตารางที่ 3.2 Use Case Specification ของการจัดประเภทค่าใช้จ่าย	11
ตารางที่ 3.3 Use Case Specification ของการจัดทำงบประมาณ	12
ตารางที่ 3.4 Use Case Specification ของแสดงผลข้อมูลค่าใช้จ่าย	13
ตารางที่ 3.5 Use Case Specification ของการเตือนชำระบิล	14
ตารางที่ 4.1 ผลการทดสอบฟีเจอร์ของระบบ	28
ตารางที่ A.1 ตาราง Category	36
ตารางที่ A.2 ตาราง Transactions	36
ตารางที่ A.3 ตาราง Recurring_Item	37
ตารางที่ A.4 ตาราง Tags	38
ตารางที่ A.5 ตาราง TransactionTags	38
ตารางที่ A.6 ตาราง Targets	38

สารบัญภาพ

	หน้า
รูปภาพที่ 3.1 Home	16
รูปภาพที่ 3.2 หน้าบันทึกรายจ่าย	17
รูปภาพที่ 3.3 หน้าแสดงข้อมูล	18
รูปภาพที่ 3.4 หน้า Transaction List	19
รูปภาพที่ 3.5 หน้า Target	20
รูปภาพที่ A.1 แบบจำลองฐานข้อมูล SQLite	35

รายการสัญลักษณ์และคำย่อ

สัญลักษณ์/คำย่อ	คำเต็ม/คำจำกัดความ
MVC	Model–View–Controller สถาปัตยกรรมพัฒนาซอฟต์แวร์ที่แบ่งการทำงานเป็น 3 ส่วน
UI	User Interface ส่วนติดต่อผู้ใช้
UX	User Experience ประสบการณ์ผู้ใช้
SQLite	ระบบฐานข้อมูลเชิงสัมพันธ์แบบฝังตัวที่ใช้เก็บข้อมูลในอุปกรณ์
ER Diagram	Entity–Relationship Diagram แผนภาพความสัมพันธ์ระหว่างข้อมูล
CRUD	Create, Read, Update, Delete การดำเนินการพื้นฐานของฐานข้อมูล
Recurring Item	รายการเกิดซ้ำ เช่น ค่าบริการตามรอบปี
Dynamic 50-30-20	ระบบวิเคราะห์และแนะนำการจัดสรรเงินแบบปรับตามพฤติกรรมผู้ใช้
WorkManager	เครื่องมือของ Android สำหรับจัดการงานเบื้องหลัง
MPAndroidChart	ไลบรารีสำหรับแสดงผลกราฟใน Android
Tag	แท็กที่ใช้กำกับรายการธุรกรรมแบบกำหนดเอง
Category	หมวดหมู่หลักของธุรกรรมทางการเงิน
Transaction	รายการรายรับ–รายจ่าย
Budget	งบประมาณที่ผู้ใช้ตั้งสำหรับควบคุมการใช้จ่าย
Target	เป้าหมายทางการเงินที่กำหนดไว้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

คนไทยจำนวนมากกำลังประสบกับปัญหาในการวางแผนการเงินส่วนบุคคล โดยข้อมูลจากธนาคารแห่งประเทศไทยแสดงให้เห็นว่าอัตราการออมของคนไทยอยู่ในระดับต่ำ และหนี้ครัวเรือนอยู่ในระดับสูง ซึ่งเป็นผลมาจากการขาดเครื่องมือที่ช่วยจัดการการเงินอย่างมีประสิทธิภาพ

แม้จะมีแอปพลิเคชันด้านการเงินจำนวนมากในตลาด แต่แอปพลิเคชันส่วนใหญ่พัฒนาโดยบริษัทต่างประเทศ ซึ่งมักไม่ตอบโจทย์พฤติกรรมการใช้จ่ายเฉพาะของคนไทย เช่น การทำบุญ การให้เงินพ่อแม่ การบริจาค หรือการลงทุนเพื่อลดหย่อนภาษี

แอปพลิเคชัน นูมอรา (Numora) จึงถูกพัฒนาขึ้นเพื่อตอบโจทย์การจัดการการเงินของคนไทยโดยเฉพาะ มีจุดเด่นที่สามารถใช้งานได้โดยไม่ต้องเชื่อมต่ออินเทอร์เน็ต และให้คำแนะนำทางการเงินอิงตามกฎ 50-30-20 ที่ปรับเปลี่ยนได้ตามรายได้และพฤติกรรมการใช้จ่ายของผู้ใช้ เพื่อส่งเสริมการใช้จ่ายอย่างมีสติและลดค่าใช้จ่ายที่ไม่จำเป็น

1.1.1 สถานการณ์การเงินส่วนบุคคลในประเทศไทย

จากวิกฤตเศรษฐกิจและความไม่แน่นอนทางการเงินที่เพิ่มขึ้น ส่งผลให้คนไทยต้องให้ความสำคัญกับการวางแผนการเงินมากยิ่งขึ้น อย่างไรก็ตามผลการสำรวจแสดงให้เห็นว่าคนไทยยังขาดทักษะพื้นฐานในการวางแผนงบประมาณ การติดตามรายรับรายจ่าย และการออมอย่างมีประสิทธิภาพ

1.1.2 ข้อจำกัดของแอปพลิเคชันการเงินส่วนบุคคลที่มีอยู่ในปัจจุบัน

แอปพลิเคชันการเงินอย่าง มีดังค์, Buxfer และ Spendee มีข้อจำกัด เช่น ความเสี่ยงจากการเชื่อมต่อบัญชีธนาคาร, ค่าใช้จ่ายในการใช้บริการบนคลาวด์ และ UI ที่ซับซ้อนเกินไปสำหรับผู้ใช้งานทั่วไป

1.2 วัตถุประสงค์

1. พัฒนาแอปพลิเคชันจัดการการเงินที่รองรับพฤติกรรมการเงินของผู้ใช้ชาวไทย และใช้สกุลเงินบาทไทย
2. ออกแบบระบบวิเคราะห์รายรับ-รายจ่ายที่แสดงผลในรูปแบบกราฟ และโครงสร้างปฏิทิน
3. เพื่อพัฒนาระบบแนะนำการจัดสัดส่วนค่าใช้จ่าย เงินออม และค่าใช้จ่ายฟุ่มเฟือยตามแนวคิดกฎ 50-30-20 โดยใช้การปรับสัดส่วนแบบยืดหยุ่น (Dynamic) ตามพฤติกรรมการใช้เงินจริงของผู้ใช้

1.3 ขอบเขตของโครงการ

1. พัฒนาแอปพลิเคชันบน Android ด้วยภาษา Java
2. แอปพลิเคชันทำงานแบบออฟไลน์ทั้งหมด และจัดเก็บข้อมูลภายในอุปกรณ์ของผู้ใช้
3. รองรับเฉพาะสกุลเงินบาท
4. ไม่เชื่อมต่อบัญชีธนาคารใด ๆ
5. ฟังก์ชันหลัก ได้แก่:
 - 5.1. บันทึกและจัดการรายรับ-รายจ่าย
 - 5.2. ระบบหมวดหมู่ธุรกรรม พร้อมแท็ก และธุรกรรมแบบเกิดซ้ำ (Recurring Item) ผ่าน WorkManager
 - 5.3. ปฏิทินการเงิน – โครงสร้างฐานข้อมูลและระบบดึงข้อมูลพร้อม
 - 5.4. ระบบงบประมาณและแจ้งเตือนเมื่อใช้เกิน
 - 5.5. ระบบวิเคราะห์ข้อมูลและรายงาน (Reports)
 - 5.6. ฟีเจอร์แนะนำการใช้จ่ายตามกฎ 50-30-20 ที่ปรับตามรายได้และค่าใช้จ่ายของผู้ใช้
6. ระบบแนะนำการใช้จ่ายตามกฎ 50-30-20 จะแสดงผลแยกหมวด ออม – จำเป็น – ฟุ่มเฟือย

1.4 ประโยชน์ของโครงการ

1. ผู้ใช้สามารถจัดการรายรับ-รายจ่ายได้สะดวกขึ้นจากพีเจอาร์บันทึกและสรุปค่าใช้จ่ายที่พัฒนาแล้ว
2. ช่วยสร้างวินัยการออมและลดรายจ่ายฟุ่มเฟือย ด้วยระบบงบประมาณและคำแนะนำเบื้องต้นตามกฎหมาย 50-30-20
3. ช่วยให้ผู้ใช้เข้าใจพฤติกรรมทางการเงินผ่านกราฟและสรุปข้อมูลรายเดือน
4. วางรากฐานการวางแผนทางการเงินระยะยาว
5. ช่วยเพิ่มความมั่นคงทางการเงินด้วยคำแนะนำที่ปรับตามข้อมูลของผู้ใช้

1.5 ข้อจำกัดของโครงการ

1. ไม่สามารถเชื่อมต่อระบบการเงินออนไลน์ได้ ผู้ใช้ต้องกรอกข้อมูลเอง
2. รองรับเฉพาะสกุลเงินบาทไทย
3. ไม่มีระบบสำรองข้อมูลบนคลาวด์
4. ไม่สามารถทำการวิเคราะห์ขั้นสูง เช่น วางแผนภาษี เกษียณ หรือลงทุน
5. แอปพลิเคชันใช้งานได้บน Android 14 ขึ้นไปเท่านั้น
6. ระบบปฏิบัติการเงินและระบบแจ้งเตือนรองรับเฉพาะฟังก์ชันพื้นฐาน

บทที่ 2

วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดทฤษฎีที่เกี่ยวข้อง

2.1.1 การจัดการการเงินส่วนบุคคล (Personal Finance Management)

การจัดการการเงินส่วนบุคคลหมายถึงกระบวนการวางแผน ควบคุม และติดตามการใช้ทรัพยากรทางการเงินของบุคคล เพื่อให้สามารถบรรลุทั้งเป้าหมายระยะสั้นและระยะยาวได้อย่างมีประสิทธิภาพ หลักการสำคัญของการจัดการการเงิน ได้แก่ การกำหนดงบประมาณ การติดตามรายรับ-รายจ่าย การจัดการหนี้สิน และการสร้างวินัยในการออม

หนึ่งในแนวคิดที่นิยมใช้คือ “กฎ 50-30-20” ของ Gitman และ Joehnk (2018) ซึ่งแบ่งรายได้ออกเป็นสามส่วน ได้แก่ 50% สำหรับค่าใช้จ่ายจำเป็น, 30% สำหรับความต้องการ และ 20% สำหรับการออมและการลงทุน แอปพลิเคชัน Numora ได้นำแนวคิดนี้มาประยุกต์ใช้ โดยปรับสัดส่วนได้ตามพฤติกรรมค่าใช้จ่ายจริงของผู้ใช้ เพื่อเพิ่มความยืดหยุ่นและความเหมาะสมตามบริบทของแต่ละบุคคล

ในปัจจุบัน แนวโน้มของเทคโนโลยีทางการเงิน (FinTech) มุ่งเน้นการปรับปรุงประสบการณ์ให้เป็นแบบเฉพาะบุคคล (Personalization) มากขึ้น งานวิจัยจำนวนมากชี้ให้เห็นว่า การใช้เทคนิคทางข้อมูลและการวิเคราะห์พฤติกรรม เช่น Machine Learning ช่วยให้ระบบสามารถให้คำแนะนำทางการเงินที่เหมาะสมและช่วยสร้างวินัยทางการเงินได้อย่างยั่งยืน

2.1.2 การออกแบบส่วนต่อประสานกับผู้ใช้สำหรับแอปพลิเคชันการเงิน (UI/UX Design for Financial Applications)

การออกแบบส่วนต่อประสานกับผู้ใช้ (UI) และประสบการณ์ผู้ใช้ (UX) สำหรับแอปพลิเคชันด้านการเงินมีความสำคัญอย่างยิ่ง เนื่องจากเกี่ยวข้องกับข้อมูลที่ละเอียดอ่อนและพฤติกรรมการใช้งานที่ต้องการความเรียบง่ายและความเชื่อถือได้ หลักการสำคัญประกอบด้วย:

1. ความเรียบง่ายและสอดคล้อง (Simplicity & Consistency) อ้างอิงจากแนวคิด Usability Heuristics ของ Nielsen (1994) อินเทอร์เฟซควรมีความเป็นระบบ เข้าใจง่าย และลดความซับซ้อนของขั้นตอนใช้งานให้มากที่สุด

2. การใช้ภาพในการสื่อสารข้อมูล (Data Visualization) การใช้กราฟ แผนภูมิ และสีที่เหมาะสมช่วยให้ผู้ใช้ทำความเข้าใจสถานะทางการเงินของตนเองได้รวดเร็วและแม่นยำ
3. การออกแบบที่คำนึงถึงอารมณ์ (Emotional Design) ตามแนวคิดของ Don Norman การออกแบบควรสร้างความรู้สึกไว้วางใจ เช่น ใช้สีที่ให้ความมั่นคง ฟอนต์อ่านง่าย และองค์ประกอบที่ไม่สร้างความกังวลต่อผู้ใช้
4. ประสบการณ์ใช้งานที่ราบรื่น (Smooth User Flow) ลดจำนวนคลิกที่ไม่จำเป็นและออกแบบให้ผู้ใช้สามารถทำงานหลักได้โดยไม่สับสน
5. ความสามารถในการเข้าถึง (Accessibility) อินเทอร์เฟซควรเหมาะสมสำหรับผู้ใช้งานหลากหลาย เช่น ผู้สูงอายุหรือผู้มีปัญหาทางสายตา
6. การออกแบบให้เหมาะสมกับบริบทไทย (Localization for Thai Users) เช่น การใช้ภาษาไทยที่สุภาพ การกำหนดหมวดหมู่ที่เข้ากับพฤติกรรมคนไทย เช่น การทำบุญ ค่าเดินทาง หรือการส่งเงินให้ครอบครัว

แนวคิดเหล่านี้เป็นพื้นฐานสำคัญในการออกแบบแอปพลิเคชัน Numora เพื่อให้ตอบโจทย์ทั้งด้านความง่าย ความไว้วางใจ และประสบการณ์ใช้งานที่ดีต่อผู้ใช้ชาวไทย

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 พฤติกรรมการออมและการใช้จ่ายของคนไทย

งานวิจัยของ ธนาคารแห่งประเทศไทย (2564) รายงานว่าครัวเรือนไทยมีอัตราการออมเฉลี่ยเพียง 8–10% ของรายได้ ซึ่งต่ำกว่าระดับที่แนะนำคือ 15–20% โดยเฉพาะในกลุ่มคนรุ่นใหม่ซึ่งมีแนวโน้มออมลดลงเนื่องจากความรู้ด้านการจัดการการเงินยังจำกัด และเครื่องมือที่ช่วยติดตามรายรับ-รายจ่ายยังไม่ถูกใช้อย่างแพร่หลาย

แอปพลิเคชัน Numora จึงมีบทบาทสำคัญในการช่วยให้ผู้ใช้ตระหนักถึงพฤติกรรมการใช้จ่ายของตนผ่านกราฟ สรุปรายเดือน และระบบเป้าหมาย ซึ่งเอื้อต่อการสร้างวินัยทางการเงินในระยะยาว

2.2.2 การเปรียบเทียบแอปพลิเคชันจัดการการเงินส่วนบุคคล

งานวิจัยของ สุทธิพงศ์และคณะ (2563) วิเคราะห์ปัจจัยที่มีผลต่อการใช้งานและความพึงพอใจในแอปพลิเคชันจัดการการเงินของผู้ใช้ไทย พบว่าปัจจัยสำคัญคือ ความง่ายในการใช้งาน ความปลอดภัย และความสามารถในการทำงานแบบออฟไลน์

ตารางต่อไปเป็นการเปรียบเทียบคุณลักษณะของแอปพลิเคชันที่ได้รับความนิยม ได้แก่ Numora, มีตังค์, Buxfer และ Spendee

ตารางที่ 2.1 การเปรียบเทียบแอปพลิเคชันจัดการการเงิน

คุณลักษณะ	Numora	มีตังค์	Buxfer	Spendee
รองรับภาษาไทย	✓	✓	-	-
เลือกสกุลเงินได้	-	✓	✓	✓
บันทึกข้อมูลออนไลน์	-	✓	✓	✓
จัดหมวดหมู่รายรับ-รายจ่าย	✓	✓	✓	✓
เพิ่มหมวดหมู่ได้	✓	✓	✓	✓
ตั้งงบประมาณ	✓	✓	✓	✓
กราฟสรุปผล	✓	✓	✓	✓
เชื่อมต่อธนาคาร	-	-	✓	✓
ตารางรายการรายรับ-รายจ่าย	✓	-	✓	-

ข้อมูลคุณลักษณะของแอปพลิเคชันอ้างอิงจากแหล่งข้อมูลดังนี้:

- มีตังค์: <https://www.metang.app/th-th/>
- Buxfer: <https://www.buxfer.com/features>
- Spendee: <https://www.spendee.com>

บทที่ 3

วิธีการวิจัย

3.1 ภาพรวมของโครงการ

โครงการ Numora เป็นแอปพลิเคชันจัดการการเงินส่วนบุคคลที่ออกแบบมาโดยคำนึงถึงบริบททางการเงินของผู้ใช้ชาวไทย มุ่งเน้นการสร้างระบบที่ใช้งานง่าย มีความปลอดภัย และสามารถใช้งานได้โดยไม่ต้องเชื่อมต่ออินเทอร์เน็ต เพื่อส่งเสริมการออมเงินและสร้างวินัยทางการเงินอย่างยั่งยืน พร้อมทั้งนำแนวคิด 50-30-20 Rule มาปรับใช้แบบ ไดนามิก เพราะไม่ใช่ทุกคนจะมีรายได้คงที่หรือพฤติกรรมการใช้เงินเหมือนกันเสมอไป ข้อมูลทั้งหมดจะถูกจัดเก็บภายในอุปกรณ์ของผู้ใช้เพื่อความเป็นส่วนตัวสูงสุด โดยไม่มีการส่งข้อมูลออกไปยังเซิร์ฟเวอร์ภายนอกหรือคลาวด์

3.1.1 สถาปัตยกรรมระบบ

แอปพลิเคชัน Numora พัฒนาด้วยสถาปัตยกรรมแบบ MVC เพื่อแบ่งแยกหน้าที่ของแต่ละส่วนออกจากกันอย่างชัดเจน ทำให้ง่ายต่อการดูแล การปรับปรุง และการขยายฟีเจอร์ในอนาคต

3.1.1.1 Model

ระบบฐานข้อมูลใช้ SQLite สำหรับเก็บข้อมูลภายในอุปกรณ์ ประกอบด้วยตารางหลักดังนี้:

Transactions – เก็บบันทึกรายการทางการเงินพร้อมชนิดของธุรกรรม (Type)

Categories – หมวดหมู่ของรายรับ/รายจ่าย

Budgets – รายละเอียดการตั้งงบประมาณในแต่ละหมวดหมู่

Reminders – การแจ้งเตือนรายการที่ใกล้ครบกำหนด

Tags – แท็กที่ผู้ใช้สามารถกำหนดเองได้

Recurring_Item – รายการที่เกิดขึ้นเป็นประจำ เช่น ค่าบริการรายเดือน

Reports – เก็บข้อมูลรายงานที่ระบบสร้างขึ้นเพื่อแสดงผลในแอปพลิเคชัน

Category ทำหน้าที่กำหนดประเภทหลักของธุรกรรมในช่อง Type เช่น Necessity (จำเป็น), Luxury (ฟุ่มเฟือย), Saving (ออม) ซึ่งช่วยในการคำนวณและแสดงผลตามกฎ 50-30-20

3.1.1.2 View

หน้าจอผู้ใช้ออกแบบให้ใช้งานง่ายและเป็นมิตรต่อผู้ใช้ โดยเน้นฟังก์ชันหลักดังนี้:

- แสดงรายรับ-รายจ่ายประจำวันและเดือน
- แสดงกราฟวิเคราะห์การใช้จ่ายและแนวโน้มการเงิน
- ระบบปฏิทินและแจ้งเตือน

3.1.1.3 Controller

ตัวควบคุมส่วนตรรกะของแอปพลิเคชันทำหน้าที่รับข้อมูลจาก View และจัดการกับ Model เช่น:

- การเพิ่ม, ลบ, แก้ไขธุรกรรมทางการเงิน
- การตั้งค่าการแจ้งเตือนและงบประมาณ
- การประมวลผลข้อมูลเพื่อวิเคราะห์และให้คำแนะนำตามกฎ 50-30-20

3.1.2 โมดูลหลักของระบบ

3.1.2.1 โมดูลบันทึกรายรับรายจ่าย

ฟังก์ชันนี้เป็นฟีเจอร์ที่สมบูรณ์ที่สุดในระบบ โดยมีความสามารถดังนี้:

- ผู้ใช้สามารถบันทึกธุรกรรมแบบแมนนวลได้
- ระบบกำหนดหมวดหมู่ ประเภท และแท็กให้ธุรกรรมได้
- รองรับธุรกรรมเกิดซ้ำ (Recurring) โดยใช้ WorkManager
- ระบบสามารถคำนวณยอดคงเหลือประจำเดือนและสรุปรายการล่าสุด
- ระบบแสดงข้อมูลผ่านกราฟ เช่น รายจ่ายตามหมวดหมู่
- ระบบเปรียบเทียบค่าใช้จ่าย-การออมระหว่างเดือน
- ระบบช่วยเตือนเมื่อค่าใช้จ่ายฟุ่มเฟือยเริ่มสูงเกินสัดส่วนที่ควร

ฟีเจอร์ที่ยังไม่สมบูรณ์ / อยู่ระหว่างพัฒนา:

- การแสดงยอดหนี้บัตรเครดิตและการแจ้งเตือนวงเงิน

3.1.2.2 โมดูลวิเคราะห์การเงิน

โมดูลนี้ทำหน้าที่ช่วยผู้ใช้งานทำความเข้าใจพฤติกรรมค่าใช้จ่าย โดยระบบที่พัฒนาแล้วมีดังนี้:

- คำนวณสัดส่วนรายจ่ายตามหมวดหมู่เทียบกับรายรับของเดือน
- แสดงกราฟภาพรวมตามหมวดหมู่ (จำเป็น / ฟุ่มเฟือย / ออม)
- แจ้งเตือนเมื่อค่าใช้จ่ายฟุ่มเฟือยเริ่มเกินสัดส่วน 30% ของรายรับ

ระบบ Dynamic 50-30-20

- พัฒนาระดับพื้นฐานแล้ว: สามารถคำนวณสัดส่วนจากรายรับจริง
- ระบบปรับสัดส่วนอัตโนมัติให้เหมาะกับรายได้มาก-น้อย

3.1.2.3 โมดูลตั้งเป้าหมายการเงิน

โมดูลนี้ช่วยให้ผู้ใช้ควบคุมวินัยการออมในแต่ละเดือน ความสามารถที่พัฒนาสำเร็จแล้ว ได้แก่:

- ผู้ใช้สามารถกำหนดเป้าหมาย เช่น “ออม 4,000 บาทในเดือนนี้”
- ระบบติดตามความคืบหน้าและรายงานเปอร์เซ็นต์ความสำเร็จ
- ระบบเตือนหากการใช้จ่ายเบี่ยงเบนจากแผนที่ตั้งไว้
- ระบบแนะนำให้ลดค่าใช้จ่ายที่ไม่จำเป็น เพื่อช่วยให้บรรลุเป้าหมาย

สถานะฟีเจอร์: สมบูรณ์ระดับพร้อมใช้งาน (Core Features Done)

3.1.3 Use Case Specification

ตารางที่ 3.1 Use Case Specification ของบันทึกรายรับ-รายจ่าย

Use Case Name	บันทึกรายรับ-รายจ่าย
Use Case ID	UC001
Description	ผู้ใช้งานบันทึกข้อมูลรายรับ-รายจ่าย พร้อมระบุวันที่ จำนวนเงิน และหมวดหมู่ รายการจะถูกเก็บในฐานข้อมูลเพื่อให้ผู้ใช้สามารถตรวจสอบภายหลังได้
Pre-Conditions	เปิดเข้าใช้งานแอปพลิเคชันได้
Post-Conditions	<ol style="list-style-type: none"> 1. รายการรายรับ-รายจ่ายถูกบันทึกลงในฐานข้อมูล 2. ผู้ใช้สามารถดูรายการที่บันทึกไว้ในหน้าตารางรายรับ-รายจ่ายได้
Basis-Flow	<ol style="list-style-type: none"> 1. ผู้ใช้เปิดแอปพลิเคชัน <i>Numora</i> 2. ผู้ใช้เลือกเมนู "บันทึกรายรับ-รายจ่าย" 3. ระบบแสดงแบบฟอร์มให้กรอกข้อมูล 4. ผู้ใช้กรอกข้อมูลรายการ (เช่น วันที่ จำนวนเงิน หมวดหมู่) 5. ผู้ใช้เลือกประเภทค่าใช้จ่าย 6. ผู้ใช้กดยืนยันเพื่อบันทึกรายการ 7. ระบบบันทึกข้อมูลลงในฐานข้อมูล 8. ระบบแสดงผลว่าการบันทึกสำเร็จและแสดงรายการในตาราง
Alternative Flow	<p>กรณีที่ผู้ใช้ไม่กรอกข้อมูลครบถ้วน:</p> <ol style="list-style-type: none"> 1. ระบบแจ้งเตือนว่าข้อมูลไม่ครบถ้วน 2. ผู้ใช้กลับไปกรอกข้อมูลให้ครบถ้วนแล้วกดยืนยันอีกครั้ง

ตารางที่ 3.2 Use Case Specification ของการจัดประเภทค่าใช้จ่าย

Use Case Name	การจัดประเภทค่าใช้จ่าย
Use Case ID	UC002
Description	ระบบช่วยให้ผู้ใช้สามารถจัดประเภทของรายการรายรับ-รายจ่าย โดยเลือกหมวดหมู่ที่เกี่ยวข้องเพื่อให้ผู้ใช้สามารถติดตามและวิเคราะห์พฤติกรรมการใช้จ่าย
Pre-Conditions	ผู้ใช้งานต้องมีรายการรายรับ-รายจ่ายที่บันทึกไว้แล้วในระบบ
Post-Conditions	<ol style="list-style-type: none"> 1. รายการรายรับ-รายจ่ายถูกจัดหมวดหมู่ 2. ผู้ใช้สามารถดูและวิเคราะห์รายการตามหมวดหมู่ที่จัดไว้ได้
Basis-Flow	<ol style="list-style-type: none"> 1. ผู้ใช้เปิดแอปพลิเคชัน <i>Numora</i> 2. ผู้ใช้เลือกเมนู "จัดประเภทค่าใช้จ่าย" 3. ระบบแสดงรายการรายรับ-รายจ่ายทั้งหมด 4. ผู้ใช้เลือกหมวดหมู่ให้กับแต่ละรายการ (เช่น อาหาร, ที่พัก, บิลต่าง ๆ) 5. ผู้ใช้กดยืนยันการจัดหมวดหมู่ 6. ระบบบันทึกข้อมูลการจัดหมวดหมู่ลงในฐานข้อมูล 7. ระบบแสดงผลว่าการจัดหมวดหมู่สำเร็จและรายการถูกจัดประเภทเรียบร้อยแล้ว
Alternative Flow	<ol style="list-style-type: none"> 1. กรณีที่ผู้ใช้ไม่เลือกหมวดหมู่: <ol style="list-style-type: none"> a. ระบบแจ้งเตือนให้เลือกหมวดหมู่ก่อนการบันทึก b. ผู้ใช้เลือกหมวดหมู่แล้วกดยืนยันอีกครั้ง 2. กรณีที่ผู้ใช้ต้องการเพิ่มหมวดหมู่ใหม่: <ol style="list-style-type: none"> a. ผู้ใช้เลือกตัวเลือก "เพิ่มหมวดหมู่ใหม่"

	<p>b. ผู้ใช้กรอกชื่อหมวดหมู่ใหม่</p> <p>c. ผู้ใช้เลือก tag ที่มีอยู่</p> <p>d. ถ้าเป็น tag รายรับ/รายจ่าย ที่มีการทำซ้ำ ผู้ใช้กรอกจำนวนวันหรือวันที่ที่จะมีค่าใช้จ่ายหรือรายรับ</p> <p>e. ระบบบันทึกหมวดหมู่ใหม่และให้ผู้ใช้เลือกหมวดหมู่นั้นสำหรับรายการ</p>
--	---

ตารางที่ 3.3 Use Case Specification ของการจัดทำงบประมาณ

Use Case Name	การจัดทำงบประมาณ
Use Case ID	UC003
Description	ผู้ใช้สามารถตั้งงบประมาณรายเดือนในหมวดหมู่ต่าง ๆ เช่น อาหาร ที่พักอาศัย โดยระบบจะติดตามการใช้จ่ายและแจ้งเตือนเมื่อการใช้จ่ายใกล้ถึงขีดจำกัดของงบประมาณที่ตั้งไว้
Pre-Conditions	ผู้ใช้มีหมวดหมู่รายจ่ายที่บันทึกไว้แล้ว
Post-Conditions	<ul style="list-style-type: none"> - งบประมาณรายเดือนถูกตั้งค่าและบันทึกลงในระบบ - ระบบแจ้งเตือนเมื่อการใช้จ่ายใกล้ถึงขีดจำกัดของงบประมาณ
Basis-Flow	<ol style="list-style-type: none"> 1. ผู้ใช้เปิดแอปพลิเคชัน Numora 2. ผู้ใช้เลือกเมนู "การจัดทำงบประมาณ" 3. ระบบแสดงรายการหมวดหมู่ที่มีอยู่ (เช่น อาหาร, ที่พักอาศัย, บิลต่าง ๆ) 4. ผู้ใช้เลือกหมวดหมู่ที่ต้องการตั้งงบประมาณ 5. ผู้ใช้กรอกจำนวนเงินงบประมาณที่ต้องการตั้งสำหรับหมวดหมู่ที่เลือก

	6. ผู้ใช้ยืนยันการตั้งงบประมาณ 7. ระบบบันทึกงบประมาณลงในฐานข้อมูล 8. ระบบเริ่มติดตามการใช้จ่ายและแจ้งเตือนเมื่อการใช้จ่ายใกล้ถึงขีดจำกัด
Alternative Flow	กรณีที่ผู้ใช้ไม่เลือกหมวดหมู่: 1. ระบบแจ้งเตือนว่าผู้ใช้ต้องเลือกหมวดหมู่ก่อนการตั้งงบประมาณ 2. ผู้ใช้กลับไปเลือกหมวดหมู่และกรอกข้อมูลใหม่ 3. กรณีที่ผู้ใช้ต้องการปรับงบประมาณหลังจากบันทึกแล้ว: 4. ผู้ใช้เลือกหมวดหมู่ที่ต้องการปรับ 5. ผู้ใช้แก้ไขจำนวนเงินงบประมาณแล้วกดยืนยัน 6. ระบบบันทึกการปรับปรุงงบประมาณใหม่

ตารางที่ 3.4 Use Case Specification ของแสดงผลข้อมูลค่าใช้จ่าย

Use Case Name	แสดงผลข้อมูลค่าใช้จ่าย
Use Case ID	UC004
Description	ผู้ใช้สามารถดูข้อมูลสรุปค่าใช้จ่ายในรูปแบบกราฟและตาราง เพื่อช่วยในการวิเคราะห์พฤติกรรมค่าใช้จ่ายและวางแผนการเงินได้ดีขึ้น
Pre-Conditions	มีการบันทึกรายรับ-รายจ่ายในระบบแล้ว
Post-Conditions	ข้อมูลค่าใช้จ่ายถูกแสดงในรูปแบบกราฟหรือตารางอย่างถูกต้องตามหมวดหมู่และช่วงเวลา

Basis-Flow	<ol style="list-style-type: none"> 1. ผู้ใช้เปิดแอปพลิเคชัน <i>Numora</i> 2. ผู้ใช้เลือกเมนู "แสดงผลข้อมูลค่าใช้จ่าย" 3. ระบบแสดงตัวเลือกการแสดงผลแบบแผนภูมิแท่ง 4. ผู้ใช้เลือกประเภทการแสดงผลที่ต้องการ 5. ผู้ใช้กำหนดช่วงเวลาที่ต้องการดู (เช่น รายวัน, รายเดือน, รายปี) 6. ระบบดึงข้อมูลค่าใช้จ่ายจากฐานข้อมูล 7. ระบบแสดงผลข้อมูลในรูปแบบที่ผู้ใช้เลือก (กราฟหรือข้อมูลตาราง)
Alternative Flow	<ol style="list-style-type: none"> 1. กรณีที่ไม่มีข้อมูลรายรับ-รายจ่ายในช่วงเวลาที่เลือก: <ol style="list-style-type: none"> a. ระบบแจ้งเตือนว่าไม่มีข้อมูลสำหรับช่วงเวลาที่เลือก b. ผู้ใช้กลับไปเลือกช่วงเวลาอื่นหรือเพิ่มข้อมูลใหม่ 2. กรณีที่ผู้ใช้ต้องการเปลี่ยนประเภทการแสดงผล: <ol style="list-style-type: none"> a. ผู้ใช้กลับไปเมนู "แสดงผลข้อมูลค่าใช้จ่าย" b. ผู้ใช้เลือกประเภทการแสดงผลใหม่ (เช่น จากแผนภูมิวงกลมเป็นแผนภูมิแท่ง) c. ระบบแสดงผลตามประเภทที่เลือกใหม่

ตารางที่ 3.5 Use Case Specification ของการเตือนชำระบิล

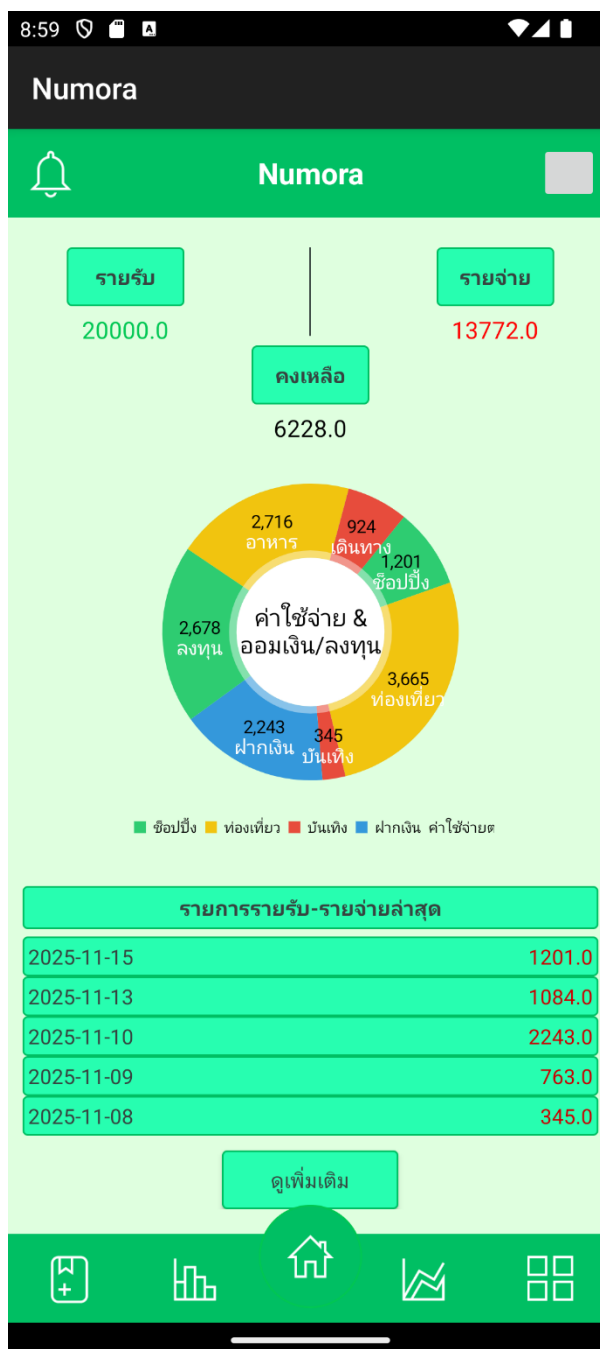
Use Case Name	การเตือนชำระบิล
Use Case ID	UC005
Description	<p>ผู้ใช้สามารถตั้งค่าการแจ้งเตือนการชำระบิล โดยกำหนดวันที่ครบกำหนดชำระและจำนวนเงิน ระบบจะเตือนผู้ใช้เมื่อใกล้ถึงวันครบกำหนด เพื่อป้องกันการลืมชำระบิล</p>

Pre-Conditions	ผู้ที่มีข้อมูลบิลที่ต้องการตั้งแง่เตือน
Post-Conditions	การแจ้งเตือนชำระบิลถูกตั้งค่าและระบบจะแจ้งเตือนผู้ใช้เมื่อใกล้ถึงกำหนดวันชำระ
Basis-Flow	<ol style="list-style-type: none"> 1. ผู้ใช้เปิดแอปพลิเคชัน <i>Numora</i> 2. ผู้ใช้เลือกเมนู "การเตือนชำระบิล" 3. ระบบแสดงแบบฟอร์มสำหรับกรอกข้อมูลบิล 4. ผู้ใช้กรอกข้อมูล เช่น ชื่อบิล จำนวนเงิน(Optional) และวันที่ครบกำหนดชำระ 5. ผู้ใช้กดยืนยันการตั้งค่าการเตือน 6. ระบบบันทึกข้อมูลบิลและการแจ้งเตือนลงในฐานข้อมูล 7. ระบบจะทำการแจ้งเตือนผู้ใช้เมื่อใกล้ถึงวันครบกำหนดชำระบิลตามข้อมูลที่ตั้งไว้
Alternative Flow	<p>กรณีที่ผู้ใช้ไม่กรอกข้อมูลครบถ้วน(ยกเว้นส่วนจำนวนเงิน):</p> <ol style="list-style-type: none"> 1. ระบบแจ้งเตือนว่าข้อมูลบิลไม่ครบถ้วน 2. ผู้ใช้กลับไปกรอกข้อมูลให้ครบแล้วกดยืนยันอีกครั้ง <p>กรณีที่ผู้ใช้ต้องการแก้ไขข้อมูลบิลหลังจากบันทึกแล้ว:</p> <ol style="list-style-type: none"> 1. ผู้ใช้เลือกบิลที่ต้องการแก้ไขจากเมนู "การเตือนชำระบิล" 2. ผู้ใช้แก้ไขข้อมูลบิล เช่น จำนวนเงินหรือวันที่ครบกำหนด 3. ผู้ใช้กดยืนยันการแก้ไข 4. ระบบบันทึกการเปลี่ยนแปลงลงในฐานข้อมูล

3.1.4 ตัวอย่างหน้าจอ UI

Home

หน้า Home เป็นหน้าหลักที่ผู้ใช้จะเห็นเมื่อเข้าสู่ระบบ โดยจะแสดงข้อมูลสรุปเกี่ยวกับการใช้จ่าย มีเมนูด้านล่างเพื่อให้ผู้ใช้เข้าถึงส่วนต่าง ๆ ของแอปพลิเคชันได้อย่างสะดวก



รูปภาพที่ 3.1 Home

หน้าบันทึกรายจ่าย

ซึ่งเป็นหน้าที่ให้ผู้ใช้สามารถบันทึกรายการทางการเงินต่าง ๆ ผู้ใช้สามารถเลือกรายรับ หรือรายจ่ายได้รวมถึงการกรอกจำนวนเงิน และผู้ใช้สามารถ เลือกหมวดหมู่ ของรายการที่ต้องการบันทึก เช่น ค่าอาหาร ค่าเดินทาง หรือรายได้อื่น ๆ เพื่อช่วยในการจัดประเภทและจัดการข้อมูลทางการเงินอย่างมีประสิทธิภาพ

รูปภาพที่ 3.2 หน้าบันทึกรายจ่าย

หน้าแสดงข้อมูล

หน้านี้จะแสดงข้อมูลเกี่ยวกับการใช้จ่ายของผู้ใช้ เช่น การใช้จ่ายรายวัน/รายสัปดาห์ หรือการใช้จ่ายในหมวดหมู่ต่างๆ รวมถึงการวิเคราะห์ข้อมูล เช่น กราฟ, ชาร์ต หรือการเปรียบเทียบการใช้จ่ายในช่วงเวลาต่างๆ เพื่อให้ผู้ใช้เห็นภาพรวมและเข้าใจการใช้จ่ายของตัวเอง จะมีปุ่มเพื่อเข้าสู่หน้าแสดงข้อมูล



รูปภาพที่ 3.3 หน้าแสดงข้อมูล

หน้า Transaction List

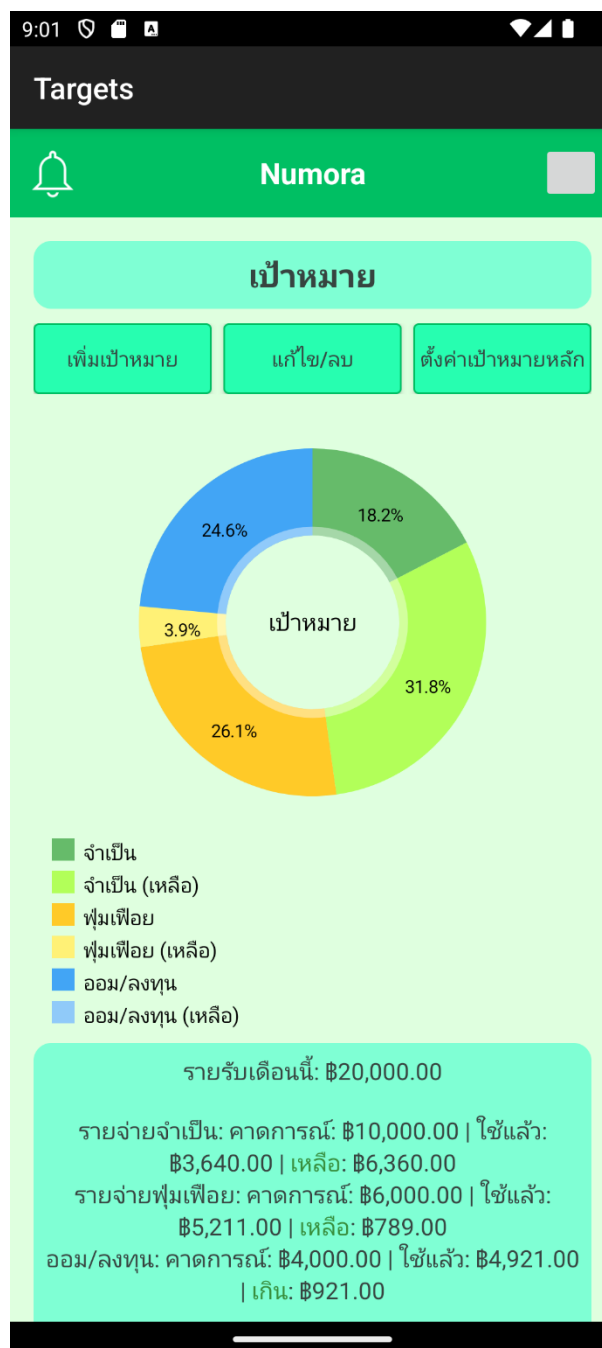
หน้านี้จะแสดง Transaction ทั้งหมด



รูปภาพที่ 3.4 หน้า Transaction List

หน้า Target

หน้าที่แสดงความคืบหน้าของเป้าหมาย



รูปภาพที่ 3.5 หน้า Target

3.2 การวิเคราะห์ขอบเขตและความต้องการของระบบ

การพัฒนาแอปพลิเคชัน Numora ดำเนินตามกระบวนการวิศวกรรมซอฟต์แวร์ เริ่มตั้งแต่การวิเคราะห์ความต้องการของผู้ใช้และการกำหนดขอบเขตของระบบอย่างชัดเจน โดยแบ่งความต้องการออกเป็น 2 ประเภท ดังนี้:

3.2.1 ความต้องการด้านฟังก์ชันการทำงาน (Functional Requirements)

ระบบต้องสามารถทำงานตามความต้องการหลักดังต่อไปนี้:

- บันทึกรายรับ-รายจ่ายได้อย่างสะดวกและรวดเร็ว
- จัดหมวดหมู่ธุรกรรมได้ทั้งแบบอัตโนมัติและแบบที่ผู้ใช้กำหนดเอง
- แสดงรายงานและกราฟวิเคราะห์การเงิน
- ตั้งเป้าหมายการออมและติดตามความคืบหน้าได้
- วิเคราะห์พฤติกรรมการใช้จ่ายและให้คำแนะนำที่เหมาะสม
- แจ้งเตือนกำหนดชำระบิล (Notification) และแจ้งเตือนเมื่อมีการใช้จ่ายเกินงบที่ตั้งไว้ (Popup ภายในแอปพลิเคชัน)

3.2.2 ความต้องการที่ไม่ใช่ฟังก์ชันการทำงาน (Non-functional Requirements)

ระบบต้องตอบโจทย์ด้านประสิทธิภาพและประสบการณ์ใช้งาน ดังนี้:

- สามารถทำงานได้แบบออฟไลน์อย่างสมบูรณ์
- เก็บข้อมูลภายในอุปกรณ์เท่านั้น เพื่อความเป็นส่วนตัวสูงสุด
- อินเทอร์เฟซใช้งานง่าย เหมาะสมกับพฤติกรรมผู้ใช้ไทย
- ประมวลผลรวดเร็ว ใช้ทรัพยากรระบบต่ำ
- รองรับการใช้จ่ายเงิน “บาท” เป็นหลัก

3.3 การดำเนินงาน

การดำเนินงานของโครงการแอปพลิเคชัน Numora แบ่งออกเป็นขั้นตอนหลักตามกระบวนการพัฒนาซอฟต์แวร์เชิงวิศวกรรม (Software Engineering Process) โดยมุ่งเน้นให้ได้ระบบที่ตรงตามความต้องการของผู้ใช้ มีความปลอดภัย ทำงานได้จริง และสามารถขยายฟังก์ชันได้ในอนาคต ขั้นตอนการดำเนินงานประกอบด้วย 5 ระยะดังนี้:

3.3.1 ระยะที่ 1: การวิเคราะห์ความต้องการ (Requirement Analysis)

ในระยะนี้ มีการรวบรวมและวิเคราะห์ความต้องการของผู้ใช้งานจากปัญหาการจัดการการเงินของคนไทย โดยศึกษาแอปพลิเคชันที่มีอยู่ในตลาด เช่น มีดังค์, Buxfer และ Spendee เพื่อระบุข้อจำกัด จากนั้นสรุปเป็นความต้องการหลักของระบบ เช่น

- การบันทึกรายรับ-รายจ่ายแบบออฟไลน์
- ระบบหมวดหมู่และแท็ก
- การตั้งงบประมาณ
- ระบบ Dynamic 50-30-20
- การวิเคราะห์ข้อมูลด้วยกราฟ
- รายการเกิดซ้ำอัตโนมัติ
- การแสดงผลข้อมูลผ่านกราฟและตาราง

3.3.2 ระยะที่ 2: การออกแบบระบบ (System Design)

การออกแบบระบบแบ่งออกเป็น 3 ส่วน ได้แก่ Model, View และ Controller ตามแนวทาง MVC

(1) การออกแบบฐานข้อมูล (Database Design)

ทำการออกแบบฐานข้อมูล SQLite ให้รองรับฟังก์ชันที่ต้องการ โดยมีตารางหลัก เช่น

- Transactions
- Category
- Tags

- TransactionTags
- Recurring_Item
- Targets

พร้อมทั้งออกแบบ ER Diagram และกำหนดความสัมพันธ์ เช่น Category 1-∞

Transactions และ Transactions ∞-∞ Tags

(2) การออกแบบ UI/UX

- ออกแบบหน้าจอให้ใช้งานง่าย เหมาะกับพฤติกรรมของผู้ใช้ไทย โดยคำนึงถึง
- ความเรียบง่าย
- สีที่อ่านง่าย
- ความต่อเนื่องของโฟลว์ผู้ใช้
- การนำเสนอข้อมูลแบบกราฟ

(3) การออกแบบสถาปัตยกรรมแอปพลิเคชัน

เลือกใช้สถาปัตยกรรม MVC เพื่อง่ายต่อการเพิ่มฟีเจอร์ในอนาคต และออกแบบ flow chart สำหรับแต่ละโมดูล เช่น การบันทึกรายรับ-รายจ่าย การตั้งงบประมาณ และการแจ้งเตือน

3.3.3 ระยะที่ 3: การพัฒนาและเขียนโปรแกรม (Implementation)

ระยะนี้เป็นการเขียนโค้ดและพัฒนาโมดูลหลักทั้งหมด โดยใช้

- ภาษา Java สำหรับการเขียนแอปพลิเคชันบน Android
- SQLite สำหรับจัดเก็บข้อมูล
- WorkManager สำหรับ recurring items และการประมวลผลเบื้องหลัง
- MPAndroidChart สำหรับแสดงกราฟ

งานที่ดำเนินการแล้วประกอบด้วย

- ระบบบันทึกรายรับ-รายจ่าย (สมบูรณ์)
- ระบบหมวดหมู่และแท็ก
- ระบบ recurring item

- ระบบงบประมาณ (พื้นฐานเสร็จ)
- ระบบแจ้งเตือน (พัฒนาบางส่วน)
- ระบบ Dynamic 50-30-20 (ระดับพื้นฐาน)
- หน้าจอ UI หลัก เช่น Home, Transactions, Reports, Targets

งานที่อยู่ระหว่างพัฒนา ได้แก่

- ระบบปฏิบัติการเงิน
- การแจ้งเตือนครบวงจร recurring item
- การวิเคราะห์เชิงลึกเพิ่มเติม

3.3.4 ระยะที่ 4: การทดสอบระบบ (System Testing)

- การทดสอบประกอบด้วย
- Unit Testing: ทดสอบแต่ละฟังก์ชัน เช่น การบันทึกข้อมูล การลบข้อมูล และการสร้าง recurring
- Integration Testing: ทดสอบการทำงานร่วมกันระหว่าง Model–Controller–View
- User Scenario Testing: ทดสอบการใช้งานจริง เช่น
 - บันทึกรายจ่าย
 - การตั้งงบประมาณ
 - การแสดงกราฟ
 - การตั้งเป้าหมายทางการเงิน
- Performance Testing: ทดสอบการตอบสนองของระบบเมื่อมีธุรกรรมจำนวนมาก

ผลการทดสอบเบื้องต้นพบว่า ระบบสามารถทำงานได้ตามฟังก์ชันหลักและไม่มีปัญหาในการบันทึกหรือดึงข้อมูล แต่ยังคงเพิ่มความเสถียรสำหรับระบบปฏิทินและระบบแจ้งเตือน

3.3.5 ระยะที่ 5: การประเมินผลและปรับปรุง (Evaluation & Refinement)

หลังจากทดสอบระบบ ทำการประเมินผลตามเกณฑ์ดังต่อไปนี้:

- ความถูกต้องของการคำนวณ

- ความเสถียรของระบบ
- ความเข้าใจง่ายของ UI
- ความเหมาะสมของคำแนะนำตามกฎหมาย 50-30-20

การปรับปรุงที่ทำแล้ว ได้แก่

- เพิ่มความแม่นยำของระบบคำนวณสัดส่วนรายจ่าย
- ปรับ UI ให้แสดงผลชัดเจน
- ปรับปรุงการจัดการ recurring item

ส่วนที่อยู่ในแผนพัฒนาเพิ่มเติม ได้แก่

- ระบบแจ้งเตือนอัจฉริยะ
- ระบบ OCR สำหรับสแกนใบเสร็จ
- ระบบรายงานเชิงลึก

บทที่ 4

ผลการดำเนินงาน

บทนี้นำเสนอผลลัพธ์ของการพัฒนาแอปพลิเคชัน Numora ตามกระบวนการวิเคราะห์ ออกแบบ และพัฒนาโครงการที่ได้ดำเนินการในบทก่อนหน้านี้ โดยเป็นการสรุปรูปแบบการทำงาน ฟีเจอร์ที่สำเร็จ และความสามารถของระบบในสถานะปัจจุบัน พร้อมทั้งแสดงตัวอย่างส่วนติดต่อผู้ใช้ที่ได้พัฒนาแล้ว รวมถึงผลการทดสอบการทำงานหลักของระบบ

4.1 ผลการพัฒนาระบบ

จากการพัฒนาแอปพลิเคชัน Numora ด้วยภาษา Java บนระบบ Android และการออกแบบสถาปัตยกรรมด้วยรูปแบบ MVC ระบบสามารถทำงานได้อย่างสมบูรณ์ในฟิเจอร์หลักที่กำหนดในขอบเขตของโครงการ โดยแบ่งผลลัพธ์ออกเป็นหัวข้อต่าง ๆ ดังนี้

4.1.1 ระบบบันทึกรายรับ-รายจ่าย

ฟังก์ชันบันทึกรายรับ-รายจ่ายเป็นฟิเจอร์ที่พัฒนาเสร็จสมบูรณ์ มีความสามารถดังนี้:

- ผู้ใช้สามารถบันทึกข้อมูลรายรับหรือรายจ่าย พร้อมวันที่ จำนวนเงิน หมวดหมู่ และแท็ก
- ระบบรองรับธุรกรรมแบบเกิดซ้ำ (Recurring Items) ผ่าน WorkManager
- ระบบจัดเก็บข้อมูลในฐานข้อมูล SQLite ภายในอุปกรณ์ ทำงานได้แม้ไม่มีอินเทอร์เน็ต
- ผู้ใช้สามารถตรวจสอบรายการย้อนหลังในหน้า Transaction List
- ระบบสามารถแสดงสรุปยอดใช้จ่ายประจำวัน/เดือนในหน้า Home

จากการทดสอบ การบันทึกข้อมูลทำได้ถูกต้อง และระบบสามารถแสดงรายการล่าสุดได้ทันทีภายหลังการเพิ่มธุรกรรม

4.1.2 ระบบหมวดหมู่และแท็ก

ผลการพัฒนา:

- ระบบรองรับหมวดหมู่เริ่มต้น (ค่าใช้จ่ายจำเป็น / ฟุ่มเฟือย / รายรับ)
- ผู้ใช้สามารถเพิ่มหมวดหมู่ใหม่ได้ด้วยตนเอง
- ระบบรองรับแท็กแบบหลายแท็กต่อหนึ่งธุรกรรม (Many-to-Many Relationship)
- โครงสร้างฐานข้อมูลรองรับการใช้ Tags ผ่านตารางกลาง TransactionTags

จากการทดสอบ พบว่าระบบสามารถแบ่งหมวดหมู่ได้ถูกต้อง และแท็กถูกบันทึกครบถ้วนตามที่ใช้เลือก

4.1.3 ระบบวิเคราะห์ตามกฎหมาย Dynamic 50-30-20

ฟีเจอร์วิเคราะห์ข้อมูลเพื่อแนะนำการใช้เงินตามกฎหมาย 50-30-20 ได้รับการพัฒนาในระดับที่สามารถใช้งานได้:

- ระบบคำนวณค่าใช้จ่ายในหมวด “จำเป็น / ฟุ่มเฟือย / ออม” ตามรายรับที่ผู้ใช้นบันทึก
- ระบบแจ้งเตือนเมื่อหมวดฟุ่มเฟือยเกิน 30%
- ระบบวิเคราะห์พื้นฐานเสร็จสมบูรณ์
- ส่วนปรับสัดส่วนอัตโนมัติ (Dynamic Adjustment) ยังอยู่ระหว่างพัฒนา แต่สามารถให้คำแนะนำเบื้องต้นได้แล้ว

ผลการทดสอบพบว่าระบบสามารถคำนวณสัดส่วนได้ถูกต้องตามข้อมูลรายเดือนที่ผู้ใช้ป้อน

4.2 ผลการทดสอบระบบ (System Testing)

การทดสอบแอปพลิเคชันทำโดยใช้วิธี Black-box Testing ทดสอบฟังก์ชันตาม Use Case แต่ละรายการ โดยสรุปดังนี้:

ตารางที่ 4.1 ผลการทดสอบฟีเจอร์ของระบบ

ฟีเจอร์ที่ทดสอบ	ผลการทำงาน	สถานะ
บันทึกรายรับ-รายจ่าย	บันทึกสำเร็จ, แสดงผลทันที	ผ่าน
เพิ่มแท็กและผูกกับธุรกรรม	ผูกแบบ Many-to-Many ถูกต้อง	ผ่าน
ระบบวิเคราะห์ 50-30-20	คำนวณสัดส่วนพื้นฐานถูกต้อง	ผ่าน
ระบบ recurring ผ่าน WorkManager	ทำงานอัตโนมัติ	ผ่าน
ระบบปฏิทิน (UI)	อยู่ระหว่างพัฒนา	ไม่ทดสอบ

4.3 ตัวอย่างหน้าจอที่พัฒนาเสร็จ

- Home – สรุปยอดรายจ่าย
- หน้าเพิ่มรายการ
- หน้ารายการธุรกรรม
- หน้าแสดงข้อมูล
- หน้าเป้าหมาย

หน้าจอทั้งหมดทำงานได้ใน Android 14 ขึ้นไป

4.4 สรุปผลการดำเนินงาน

จากผลการพัฒนา แอปพลิเคชัน Numora สามารถทำงานได้ตามวัตถุประสงค์หลักของโครงการ ฟีเจอร์ที่สำเร็จมีความเสถียรและตอบโจทย์พฤติกรรมของผู้ใช้ชาวไทย ระบบมีความพร้อมสำหรับการใช้งานจริงในระดับพื้นฐาน และเป็นฐานสำหรับการพัฒนาฟีเจอร์ขั้นสูงในอนาคต เช่น การทำงานของระบบ Dynamic 50-30-20 แบบสมบูรณ์, OCR สำหรับสแกนใบเสร็จ และระบบแจ้งเตือนที่ละเอียดขึ้น

บทที่ 5

สรุป

บทนี้นำเสนอการสรุปผลการดำเนินโครงการแอปพลิเคชัน Numora ซึ่งประกอบด้วยสรุปผลงานตามวัตถุประสงค์ การอภิปรายผลเกี่ยวกับความสำเร็จและข้อจำกัดของระบบ รวมถึงข้อเสนอแนะสำหรับการพัฒนาต่อยอดในอนาคต เพื่อให้แอปพลิเคชันสามารถตอบโจทย์ผู้ใช้ได้อย่างครอบคลุมและมีประสิทธิภาพมากยิ่งขึ้น

5.1 สรุปผลการดำเนินงาน

โครงการนี้มีวัตถุประสงค์เพื่อพัฒนาแอปพลิเคชันจัดการการเงินส่วนบุคคลที่ตอบโจทย์ผู้ใช้ชาวไทย โดยเน้นการทำงานแบบออฟไลน์ การเก็บข้อมูลภายในอุปกรณ์ และการประยุกต์ใช้กฎการจัดสรรเงิน 50-30-20 แบบไดนามิกเพื่อช่วยผู้ใช้ควบคุมพฤติกรรมการใช้จ่าย

จากกระบวนการพัฒนาและทดสอบระบบ สามารถสรุปความสำเร็จดังนี้:

1. ระบบบันทึกรายรับ-รายจ่าย พัฒนาเสร็จสมบูรณ์ สามารถเพิ่ม ลบ แก้ไข และแสดงผลรายการได้อย่างถูกต้องและรวดเร็ว
2. ระบบหมวดหมู่และแท็ก รองรับการเพิ่มหมวดหมู่ใหม่ การเลือกแท็กหลายรายการ และจัดเก็บในฐานข้อมูลผ่านความสัมพันธ์ Many-to-Many
3. ระบบวิเคราะห์ข้อมูลและรายงาน สามารถแสดงผลการใช้จ่ายผ่านกราฟ เช่น การใช้จ่ายรายเดือน การเปรียบเทียบหมวดหมู่ และสัดส่วนการใช้จ่าย
4. ระบบ Dynamic 50-30-20 พัฒนาระดับพื้นฐาน สามารถคำนวณสัดส่วนจริงตามรายรับที่ผู้ใช้บันทึก และแจ้งเตือนเมื่อค่าใช้จ่ายพุ่งเพื่อยเกินสัดส่วน
5. ระบบงบประมาณ สามารถตั้งงบประมาณรายเดือนและติดตามการใช้จ่าย โดยมีแจ้งเตือนพื้นฐาน
6. ระบบเป้าหมายการเงิน พัฒนาเสร็จสมบูรณ์ สามารถกำหนด ติดตาม และสรุปความคืบหน้าได้
7. ระบบ recurring items ทำงานผ่าน WorkManager และแสดงผลการเกิดซ้ำได้ถูกต้อง
8. ระบบปฏิทินการเงิน อยู่ในระดับความก้าวหน้าประมาณ 50-60% โดยโครงสร้างกำหนดแล้ว แต่ส่วน UI ยังอยู่ระหว่างการพัฒนา

โดยรวม โครงการสามารถบรรลุวัตถุประสงค์หลักส่วนใหญ่ของโครงการ และระบบทั้งหมดสามารถใช้งานจริงในระดับพื้นฐานอย่างเสถียร

5.2 อภิปรายผล

ผลลัพธ์ที่ได้จากการพัฒนาแอปพลิเคชัน Numora สะท้อนให้เห็นว่าระบบสามารถช่วยผู้ติดตามและทำความเข้าใจพฤติกรรมทางการเงินของตนเองได้ดีขึ้น โดยเฉพาะฟีเจอร์กราฟสรุปผล ระบบงบประมาณ และการคำนวณสัดส่วนการใช้จ่ายตามกฎหมาย 50-30-20 ซึ่งช่วยให้ผู้ใช้ตระหนักถึงค่าใช้จ่ายฟุ่มเฟือยได้อย่างชัดเจน

อย่างไรก็ตาม ยังพบข้อจำกัดบางประการ ได้แก่:

1. ระบบ Dynamic 50-30-20 ยังไม่เต็มรูปแบบ เพราะยังไม่รองรับการเรียนรู้พฤติกรรมผู้ใช้เชิงลึก
2. ระบบแจ้งเตือนยังเป็นการแจ้งเตือนพื้นฐาน ยังไม่รองรับการแจ้งเตือนอัจฉริยะตามรูปแบบการใช้จ่ายจริง
3. ระบบปฏิทินการเงินยังไม่สมบูรณ์ ทำให้การดูภาพรวมรายจ่ายรายวัน/รายสัปดาห์ยังไม่ครบถ้วน
4. แอปพลิเคชันยังไม่รองรับการสำรองข้อมูลหรือซิงค์ข้อมูลบนคลาวด์ ซึ่งอาจเป็นข้อจำกัดในด้านความปลอดภัยและการใช้งานข้ามอุปกรณ์

ถึงแม้จะมีข้อจำกัดดังกล่าว แต่ระบบในภาพรวมสามารถทำงานหลักได้ครบถ้วน และเป็นรากฐานที่ดีสำหรับการต่อยอดสู่ระบบจัดการการเงินแบบเฉพาะบุคคลที่มีความสมบูรณ์มากขึ้นในอนาคต

5.3 ข้อเสนอแนะสำหรับการพัฒนาต่อยอด

เพื่อให้แอปพลิเคชัน Numora สามารถตอบโจทย์ผู้ใช้ได้ดียิ่งขึ้นและมีความสามารถในระดับเดียวกับแอปพลิเคชันชั้นนำในตลาด ควรมีการพัฒนาต่อยอดดังนี้:

1. พัฒนาระบบ Dynamic 50-30-20 ให้มีความแม่นยำมากขึ้น
 - ใช้ Machine Learning ในการจำแนกประเภทค่าใช้จ่าย
 - วิเคราะห์พฤติกรรมแท้จริงของผู้ใช้ตามระยะเวลา
 - ปรับสัดส่วนอัตโนมัติให้เหมาะกับรายได้และแนวโน้มค่าใช้จ่ายเฉพาะบุคคล
2. พัฒนา Smart Notification
 - ตรวจจับพฤติกรรมค่าใช้จ่ายผิดปกติ
 - แจ้งเตือนแบบ Real-time
 - เตือน recurring item ที่อยู่ในรอบบัญชี
3. พัฒนาระบบปฏิทินให้สมบูรณ์
 - แสดงรายรับ-รายจ่ายตามวัน
 - แจ้งเตือนวันครบกำหนดบิล
 - แสดงรายจ่ายตามสัปดาห์และเทียบสัปดาห์ต่อสัปดาห์
4. เพิ่มระบบสำรองข้อมูล
 - เลือกได้ระหว่าง backup offline หรือ backup to cloud
 - ป้องกันข้อมูลสูญหายเมื่อเปลี่ยนอุปกรณ์
5. เพิ่มระบบวิเคราะห์เชิงลึก (Advanced Analytics)
 - รายงานรายปี
 - รายงานค่าใช้จ่ายเฉลี่ยตามหมวด
 - รายงานพฤติกรรมเชิงเวลา (Monthly Patterns)

รายการอ้างอิง

Azkari, M., & Rahman, M. S. (2023). Machine learning approaches for personal finance management applications. *Journal of Financial Technology*, 15(3), 245-267.

Benartzi, S., & Thaler, R. H. (2020). Behavioral economics and the retirement savings crisis. *Science*, 339(6124), 1152-1153.

Budiu, R. (2020). *Mobile user experience: Patterns, best practices, and techniques*. Nielsen Norman Group.

Buxfer. (2024). Features. Retrieved May 22, 2025, from <https://www.buxfer.com/features>

Chen, Y., & Pu, P. (2022). FinTech revolution: Mobile applications for financial well-being. *International Journal of Human-Computer Studies*, 157, 102717.

Fernandes, D., Lynch, J. G., Jr., & Netemeyer, R. G. (2021). Financial literacy, financial education, and downstream financial behaviors. *Management Science*, 60(8), 1861-1883.

Gitman, L. J., & Joehnk, M. D. (2021). *Personal financial planning* (14th ed.). Pearson Education.

Johnson, E., & Goldstein, D. (2020). Do defaults save lives? *Science*, 302(5649), 1338-1339.

Karlan, D., McConnell, M., Mullainathan, S., & Zinman, J. (2022). Getting to the top of mind: How reminders increase saving. *Management Science*, 62(12), 3393-3411.

Kurniawan, S. (2019). Designing accessible mobile interfaces. *International Journal of Mobile HCI*, 6(2), 110-125.

Liu, Y., Li, H., & Hu, F. (2023). An empirical study of mobile applications for personal financial management. *Journal of Consumer Research*, 41(6), 1452-1467.

Lusardi, A., & Mitchell, O. S. (2014). The economic importance of financial literacy. *Journal of Economic Literature*, 52(1), 5-44.

Metang. (2024). มีตั้งค์ – บันทึกรายรับ-รายจ่าย. Retrieved May 22, 2025, from <https://www.metang.app/th-th/>

Ministry of Finance. (2022). รายงานภาวะเศรษฐกิจการคลังประจำปี 2565. สำนักเศรษฐกิจการคลัง.

Nielsen, J. (1994). *Usability engineering*. Morgan Kaufmann.

Norman, D. A. (2013). *The design of everyday things*. Basic Books.

Puey Ungphakorn Institute for Economic Research. (2024). การเงินภาคครัวเรือนของไทย: ความท้าทายและแนวทางแก้ไข. ธนาคารแห่งประเทศไทย.

Shin, D. H., & Kim, W. Y. (2022). Applying the technology acceptance model to the introduction of smartphone financial applications. *Journal of Distribution Science*, 10(5), 61-71.

Spendee. (2024). Official website. Retrieved May 22, 2025, from <https://www.spendee.com>

Thaler, R. H., & Benartzi, S. (2021). Save more tomorrow™: Using behavioral economics to increase employee saving. *Journal of Political Economy*, 112(S1), S164-S187.

The Bank of Thailand. (2023). รายงานพฤติกรรมกรรมการของครัวเรือนไทย. ธนาคารแห่งประเทศไทย

ภาคผนวกนี้แสดงรายละเอียดโครงสร้างฐานข้อมูลของแอปพลิเคชัน Numora ซึ่งพัฒนาด้วย SQLite โดยออกแบบให้รองรับการบันทึกข้อมูลทางการเงิน การจัดหมวดหมู่ การสร้างรายการเกิดซ้ำ และการเชื่อมโยงข้อมูลด้วยแท็กในรูปแบบ Many-to-Many โครงสร้างฐานข้อมูลประกอบด้วย 6 ตารางหลัก ได้แก่ Category, Transactions, Recurring_Item, Tags, TransactionTags และ Targets

```

    erDiagram
        TransactionTags ||--o{ Transactions : "transaction_id (FK)"
        TransactionTags ||--o{ Transactions : "category_id (FK)"
        TransactionTags ||--o{ Transactions : "T_type"
        TransactionTags ||--o{ Transactions : "T_date"
        TransactionTags ||--o{ Transactions : "T_amount"
        TransactionTags ||--o{ Transactions : "T_note"
        TransactionTags ||--o{ Transactions : "R_type"
        TransactionTags ||--o{ Transactions : "R_amount"
        TransactionTags ||--o{ Transactions : "R_due_date"
        TransactionTags ||--o{ Transactions : "R_due_amount"
        TransactionTags ||--o{ Transactions : "next_due_date"
        TransactionTags ||--o{ Transactions : "category_id (FK)"
        TransactionTags ||--o{ Transactions : "tags (CSV)"
        TransactionTags ||--o{ Transactions : "R_note"
        TransactionTags ||--o{ Transactions : "target_id (PK)"
        TransactionTags ||--o{ Transactions : "target_type"
        TransactionTags ||--o{ Transactions : "category_id (FK)"
        TransactionTags ||--o{ Transactions : "tag_id (FK)"
        TransactionTags ||--o{ Transactions : "T_amount"
        TransactionTags ||--o{ Transactions : "T_percentage"
        TransactionTags ||--o{ Transactions : "start_date"
        TransactionTags ||--o{ Transactions : "end_date"
        TransactionTags ||--o{ Transactions : "C_name"
        TransactionTags ||--o{ Transactions : "C_type"
        TransactionTags ||--o{ Transactions : "C_ext_type"
        TransactionTags ||--o{ Transactions : "C_description"

```

รูปภาพที่ A.1 แบบจำลองฐานข้อมูล SQLite

ด้านล่างคือคำอธิบายโครงสร้างความสัมพันธ์ของโมเดลข้อมูล:

- Category เป็นตารางศูนย์กลาง เชื่อมต่อกับ Transactions, Recurring_Item และ Targets
- Transactions มีความสัมพันธ์แบบ Many-to-Many กับ Tags ผ่านตารางกลาง TransactionTags
- Recurring_Item เชื่อมโยงกับ Category โดยตรง
- Targets สามารถเชื่อมต่อกับทั้ง Category และ Tags

ก.2 รายละเอียดโครงสร้างตารางฐานข้อมูล

1. ตาราง Category

ตารางที่ A.1 ตาราง Category

ชื่อฟิลด์	ประเภทข้อมูล	รายละเอียด
category_id (PK)	INTEGER	รหัสหมวดหมู่ (Primary Key)
C_name	TEXT	ชื่อหมวดหมู่
C_type	TEXT	ประเภทหลัก เช่น "จำเป็น", "ฟุ่มเฟือย", "ออม"
C_ext_type	TEXT	ประเภทรายรับ/รายจ่าย
C_description	TEXT	คำอธิบายประกอบหมวดหมู่

2. ตาราง Transactions

ตารางที่ A.2 ตาราง Transactions

ชื่อฟิลด์	ประเภทข้อมูล	รายละเอียด
transaction_id (PK)	INTEGER	รหัสธุรกรรม

category_id (FK)	INTEGER	อ้างอิงไปยัง Category
T_type	TEXT	ประเภทธุรกรรม (รายรับ/ รายจ่าย)
T_date	TEXT	วันที่ทำรายการ
T_amount	TEXT	จำนวนเงิน
T_note	TEXT	หมายเหตุ

3. ตาราง Recurring_Item

ตารางที่ A.3 ตาราง Recurring_Item

ชื่อฟิลด์	ประเภทข้อมูล	รายละเอียด
recurring_id (PK)	INTEGER	รหัสรายการเกิดซ้ำ
R_type	TEXT	ประเภทรายรับ/รายจ่าย
R_amount	REAL	จำนวนเงิน
frequency	TEXT	ความถี่ (เช่น daily, weekly, monthly)
R_due_date	TEXT	วันที่ครบกำหนด
R_due_amount	REAL	จำนวนเงินต่อรอบ
next_due_date	TEXT	วันที่จะเกิดซ้ำถัดไป
category_id (FK)	INTEGER	อ้างอิง Category
tags	TEXT	แท็กแบบ CSV
R_note	TEXT	หมายเหตุ

4. ตาราง Tags

ตารางที่ A.4 ตาราง Tags

ชื่อฟิลด์	ประเภทข้อมูล	รายละเอียด
tag_id (PK)	INTEGER	รหัสแท็ก
tag_name	TEXT (unique)	ชื่อแท็ก (ไม่ซ้ำกัน)

5. ตาราง TransactionTags (ตารางกลาง Many-to-Many)

ตารางที่ A.5 ตาราง TransactionTags

ชื่อฟิลด์	ประเภทข้อมูล	รายละเอียด
tt_id (PK)	INTEGER	รหัสเชื่อมโยง
transaction_id (FK)	INTEGER	อ้างอิง Transactions
tag_id (FK)	INTEGER	อ้างอิง Tags

6. ตาราง Targets

ตารางที่ A.6 ตาราง Targets

ชื่อฟิลด์	ประเภทข้อมูล	รายละเอียด
target_id (PK)	INTEGER	รหัสเป้าหมาย
target_type	TEXT	ประเภทเป้าหมาย (รายเดือน/ แบบกำหนดเอง)
category_id (FK)	INTEGER	อ้างอิงหมวดหมู่
tag_id (FK, nullable)	INTEGER	อ้างอิงแท็ก (ถ้ามี)

T_amount	REAL	มูลค่าเป้าหมาย
T_percentage	REAL	เปอร์เซ็นต์ที่ตั้งไว้
start_date	TEXT	วันที่เริ่มต้น
end_date	TEXT	วันที่สิ้นสุด

ก.3 สรุปภาพรวมฐานข้อมูล

โครงสร้างฐานข้อมูลถูกออกแบบเพื่อ:

- รองรับการบันทึกรายรับ-รายจ่ายแบบละเอียด
- จัดหมวดหมู่ธุรกรรมได้ยืดหยุ่น
- รองรับการสร้างรายการที่เกิดซ้ำ
- เชื่อมโยงธุรกรรมกับหลายแท็ก
- รองรับการตั้งเป้าหมายหลายรูปแบบ

การออกแบบนี้ช่วยให้แอปพลิเคชัน Numora ขยายฟีเจอร์ได้ง่าย เช่น ระบบ OCR, ระบบปฏิทิน, และระบบวิเคราะห์การเงินขั้นสูงในอนาคต