

แอปพลิเคชันสแกนยาและแจ้งเตือนการทานยา

โดย

นางสาว ปรารถนา สุภาวงศ์ 6587057
นาย ฤทธิชล พลราช 6587062
นางสาว จิรัชญา ราชพลแสน 6587074

อาจารย์ที่ปรึกษา
ดร.สนิท แสงเหลา

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตรบัณฑิต
(สาขาวิชาการและเทคโนโลยีดิจิทัล)

คณะเทคโนโลยีสารสนเทศและการสื่อสาร
มหาวิทยาลัยมหิดล
2025

ลิขสิทธิ์ของมหาวิทยาลัยมหิดล

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงได้ด้วยความร่วมมือและความกรุณาจากอาจารย์ที่ปรึกษา ดร.สันิท แสงเหลา ซึ่งได้ให้คำแนะนำและแนวทางในการพัฒนาโครงการอย่างใกล้ชิด ตลอดจนให้คำปรึกษาและแก้ไขข้อบกพร่องต่าง ๆ มาโดยตลอด จนทำให้โครงการเล่มนี้ เสร็จสมบูรณ์ ผู้จัดทำจึงขอรับขอบขอพระคุณเป็นอย่างสูง

ขอขอบคุณสมาชิกในกลุ่มทุกคน ได้แก่ นางสาวปราณนา สุภาวงศ์, นายฤทธิชล พลราช และ นางสาวจิรัชญา ราชพลแสน ที่ร่วมมือร่วมใจกันทำงานอย่างเต็มความสามารถ และเปลี่ยนความคิดเห็น และช่วยเหลือซึ่งกันและกันตลอดระยะเวลาการทำการทดลอง

ขอขอบคุณผู้ร่วมทดสอบระบบทุกท่านที่ได้สละเวลาอันมีค่าในการทดสอบและให้ข้อเสนอแนะที่เป็นประโยชน์ ซึ่งช่วยให้ระบบมีความสมบูรณ์และตอบสนองความต้องการของผู้ใช้งานได้ดียิ่งขึ้น

ขอขอบคุณสำนักพัฒนามาตรฐานระบบข้อมูลสุขภาพไทยที่ได้อธิบายข้อมูลจากระบบ Thai Medicine Terminology (TMT) ซึ่งเป็นแหล่งข้อมูลด้านยาที่มีความน่าเชื่อถือ เพื่อนำมาใช้ในการพัฒนาและทดสอบระบบในโครงการนี้

สุดท้ายนี้ ขอขอบคุณครอบครัวและผู้ปกครองที่ให้การสนับสนุน ให้กำลังใจ และเป็นแรงผลักดันให้ผู้จัดทำสามารถดำเนินโครงการนี้จนสำเร็จลุล่วงได้ด้วยดี

นางสาว ปราณนา สุภาวงศ์

นาย ฤทธิชล พลราช

นางสาว จิรัชญา ราชพลแสน

แอปพลิเคชันสแกนยาและแจ้งเตือนการทานยา

นางสาว ปรารถนา สุภาวงศ์ 6587057 ITDS/B

นาย ฤทธิชล พลราช 6587062 ITDS/B

นางสาว จิรัชญา ราชพลแสน 6587074 ITDS/B

วท.บ. (วิทยาการและเทคโนโลยีดิจิทัล)

อาจารย์ที่ปรึกษาโครงการ: ดร.สนิท แสงเหลา

บทคัดย่อ

โครงการนี้มีชื่อว่า "CapYaDoo" ซึ่งเป็นแอปพลิเคชันบนโทรศัพท์มือถือระบบปฏิบัติการ Android ที่ออกแบบมาเพื่อช่วยเหลือผู้สูงอายุในการจัดการยา โดยระบบสามารถ สแกนฉลากยา หรือ บาร์โค้ด เพื่อค้นหาข้อมูลยาได้อย่างสะดวกและรวดเร็ว ผู้ใช้งานสามารถ บันทึกยาที่กำลังใช้อยู่ จดบันทึกอาการหลังการรับประทานยา, และ ตั้งการแจ้งเตือนเพื่อเตือนเวลาทานยา ได้ภายในแอปเดียว

ระบบพัฒนาด้วย Flutter (Dart) สำหรับฝั่ง Mobile Application และ Spring Boot ร่วมกับ PostgreSQL สำหรับฝั่งเซิร์ฟเวอร์ มีการประมวลผลข้อมูลจากฐานข้อมูล TMT (Thai Medicine Terminology) โดยใช้ PySpark สำหรับจัดการและเตรียมข้อมูลก่อนนำเข้าฐานข้อมูล และทำการ Deploy ระบบโดยใช้ Kind เพื่อจัดการสภาพแวดล้อม Container

คำสำคัญ: Application /Framework /Cross-platform /API /Text-to-Speech /Notification /Optical Character Recognition - OCR /Frontend /Backend

สารบัญ

หน้า

กิตติกรรมประกาศ	ii
บทคัดย่อ	iii
LIST OF FIGURES	vii
LIST OF TABLES	viii
 1 บหนำ	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตงาน.....	2
1.3.1 ด้านฟังก์ชันการทำงาน.....	2
1.3.2 ด้านข้อมูล	2
1.3.3 ด้านเทคนิค	2
1.3.4 เครื่องมือเสริม	2
1.3.5 ด้านการศึกษา	3
1.4 ขั้นตอนการดำเนินงาน.....	3
1.4.1 ขั้นตอนที่ 1: การรวบรวมปัญหา วิเคราะห์ความต้องการของผู้ใช้งานและการออกแบบ	3
1.4.2 ขั้นตอนที่ 2: พัฒนาระบบในส่วนของหน้าบ้าน (FRONTEND).....	4
1.4.3 ขั้นตอนที่ 3: การพัฒนาระบบในส่วนของหลังบ้าน (BACKEND)	4
1.4.4 ขั้นตอนที่ 4: การศึกษาและพัฒนา DATA PIPELINE ด้วย PYSPARK และ KIND	4
1.4.5 ขั้นตอนที่ 5: การทดสอบและปรับปรุงระบบ	5
1.5 ประโยชน์ที่คาดว่าจะได้รับ	5
 2 ทฤษฎีที่เกี่ยวข้อง	6
2.1 สังคมสูงวัย (AGED SOCIETY และ SUPER-AGED SOCIETY)	6
2.2 โรคไม่ติดต่อเรื้อรัง (NON-COMMUNICABLE DISEASES: NCDS).....	6
2.3 การออกแบบระบบที่เหมาะสมกับผู้สูงอายุ (HUMAN-COMPUTER INTERACTION: HCI)	6
2.4 การจัดการการใช้ยา	6
2.5 พื้นฐานและการออกแบบฐานข้อมูล	7
2.6 OPTICAL CHARACTER RECOGNITION (OCR).....	7

2.7	POSTGRESQL	8
2.7.1	คุณสมบัติที่สำคัญ	8
2.7.2	ประโยชน์ของ POSTGRESQL	9
2.8	สถาปัตยกรรมแบบ CLIENT-SERVER (CLIENT-SERVER ARCHITECTURE)	9
2.8.1	องค์ประกอบหลัก	9
2.8.2	การทำงานของระบบ CLIENT-SERVER ในโครงงาน	10
2.8.3	ข้อดีของสถาปัตยกรรม CLIENT-SERVER	10
2.8.4	การประยุกต์ใช้ในโครงงาน	10
2.9	SPRING BOOT FRAMEWORK	11
2.9.1	คุณสมบัติหลัก	11
2.9.2	การใช้งานในโครงงาน	11
2.10	FLUTTER FRAMEWORK	11
2.10.1	เปรียบเทียบการพัฒนาแอปแบบเนทีฟ (NATIVE) และข้ามแพลตฟอร์ม (CROSS-PLATFORM)	11
2.10.2	ข้อดีของ FLUTTER	12
2.10.3	FLUTTER TEXT-TO-SPEECH (TTS) PLUGIN	12
2.10.4	FLUTTER_LOCAL_NOTIFICATIONS	13
2.10.5	GOOGLE_MLKIT_TEXT_RECOGNITION	13
2.10.6	MOBILE_SCANNER PLUGIN	13
2.11	DART	14
2.11.1	คุณสมบัติเด่นของ DART	14
2.12	คำศัพท์	14
3	ผลงานที่เกี่ยวข้อง	16
3.1	แอปพลิเคชันช่วยอ่านฉลากยาโดยเฉพาะฟังก์ชันที่ช่วยให้อ่านข้อมูลยาได้จ่ายขึ้นอย่างชัดเจน	16
3.2	แอปพลิเคชันบน LINE LIFF ร่วมกับ GOOGLE APPS SCRIPT สำหรับติดตามและประเมินผลความ สำมำเสมอในการรับประทานยาของผู้ป่วยรัตนโรค โรงพยาบาลชัยนาทurenthr.....	16
3.3	โครงการวิจัยเรื่อง เทคโนโลยีเภสัชสารสนเทศแสดงข้อมูลฉลากยาเอกสารกำกับยาแบบอัตโนมัติสำหรับ บริหารจัดการคลังยาปฏิชีวนะเพื่อความปลอดภัย	17
3.4	การเปรียบเทียบแอปพลิเคชันที่มีอยู่ในตลาด	18
3.4.1	ตารางเปรียบเทียบแอปพลิเคชันในตลาด	18
3.4.2	สรุปการเปรียบเทียบ	20
3.5	สรุปและการนำไปประยุกต์ใช้	20
3.5.1	แนวทางการพัฒนาที่ได้จากการวิจัย	20

3.5.2	ข้อจำกัดที่ต้องหลีกเลี่ยง	20
3.5.3	คุณสมบัติที่จะนำมายั่งยืนในโครงงาน	21
3.6	ประสบการณ์การพัฒนาที่ผ่านมา	21
3.6.1	APPLICATION CHAT MUSIC PLAYER	21
3.6.2	APPLICATION LOCATION DISCOVERY.....	22
3.6.3	บทเรียนที่ได้จากการพัฒนา.....	22
4	ระเบียบวิธีดำเนินโครงงาน	24
4.1	การวิเคราะห์และออกแบบระบบ	24
4.1.1	PERSONA.....	24
4.1.2	USER JOURNEY MAP.....	26
4.1.3	ปัญหาและแนวทางการแก้ไข	26
4.1.4	การวิเคราะห์ความต้องการ (REQUIREMENT ANALYSIS)	27
4.1.5	การออกแบบสถาปัตยกรรม (SYSTEM ARCHITECTURE DESIGN).....	28
4.1.6	การออกแบบฐานข้อมูล (DATABASE DESIGN)	29
4.1.7	ขั้นตอนการออกแบบระบบ	29
4.2	การพัฒนาและติดตั้ง (DEVELOPMENT AND IMPLEMENTATION).....	36
4.2.1	เครื่องมือและสภาพแวดล้อมในการพัฒนา	36
4.2.2	ขั้นตอนการพัฒนา	36
4.3	การทดสอบและการประเมินผล.....	38
4.3.1	การทดสอบระบบ (SYSTEM TESTING)	38
4.3.2	การทดสอบประสิทธิภาพ (PERFORMANCE TESTING)	38
4.3.3	การทดสอบการยอมรับจากผู้ใช้ (USER ACCEPTANCE TESTING - UAT)	38
5	การประเมินผล.....	39
6	สรุปและข้อเสนอแนะ.....	40
	เอกสารอ้างอิง	41
	ข้อความเปิดเผยการใช้เทคโนโลยีปัญญาประดิษฐ์	43
	ภาคผนวก ก	44
	ภาคผนวก ข	45

LIST OF FIGURES

	Page
Figure 1.1: แผนการดำเนินโครงการ	5
Figure 4.1: ตัวตนสมมติที่เป็นตัวแทนของกลุ่มผู้ใช้เป้าหมายคนที่ 1	24
Figure 4.2: ตัวตนสมมติที่เป็นตัวแทนของกลุ่มผู้ใช้เป้าหมายคนที่ 2	25
Figure 4.3: ตัวตนสมมติที่เป็นตัวแทนของกลุ่มผู้ใช้เป้าหมายคนที่ 3	25
Figure 4.4: ประสบการณ์ของผู้ใช้เมื่อมาปฏิสัมพันธ์กับแอปพลิเคชันของเรางานที่ 1	26
Figure 4.5: ประสบการณ์ของผู้ใช้เมื่อมาปฏิสัมพันธ์กับแอปพลิเคชันของเรางานที่ 2	26
Figure 4.6: ประสบการณ์ของผู้ใช้เมื่อมาปฏิสัมพันธ์กับแอปพลิเคชันของเรางานที่ 3	27
Figure 4.7: System Architecture Design.....	29
Figure 4.8: ออกแบบ UX/UI ของ Application	30
Figure 4.9: Entity Relationship Diagram (ERD).....	31
Figure 4.10: Functional Decomposition	32
Figure 4.11: Data Flow Diagram Level 0	32
Figure 4.12: Data Flow Diagram Level 1	33
Figure 4.13: Data Flow Diagram Level 2.1 - Manage User Information	34
Figure 4.14: Data Flow Diagram Level 2.2 - OCR/ Barcode/ Text-To-Speech	34
Figure 4.15: Data Flow Diagram Level 2.3 - Medication Management	35
Figure 4.16: Data Flow Diagram Level 2.4 - Notification Management	35
Figure 4.17: Data Flow Diagram Level 2.5 - History Logging	36

LIST OF TABLES

	Page
Table 3.1: เปรียบเทียบแอปพลิเคชันในตลาด	19

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ประเทศไทยกำลังเผชิญหน้ากับการเปลี่ยนแปลงทางประชากรครั้งใหญ่ โดยกำลังก้าวเข้าสู่ "สังคมสูงวัยระดับสุดยอด" (Super-Aged Society) ซึ่งเป็นผลมาจากการสัดส่วนประชากรผู้สูงอายุที่เพิ่มขึ้นอย่างรวดเร็ว ข้อมูลระบุว่าในปี พ.ศ. 2548 ไทยได้เข้าสู่ "สังคมสูงวัย" (Aged Society) ที่มีประชากรอายุเกิน 60 ปี มากกว่าร้อยละ 10 ของประชากรทั้งหมด และในปี พ.ศ. 2566 ได้ก้าวเข้าสู่ "สังคมสูงวัยอย่างสมบูรณ์" (Complete Aged Society) ด้วยสัดส่วนประชากรกลุ่มนี้ที่เพิ่มขึ้นมากกว่าร้อยละ 20 และมีแนวโน้มที่จะเข้าสู่ "สังคมสูงวัยระดับสุดยอด" (Super-Aged Society) ในปี พ.ศ. 2576 ซึ่งจะมีประชากรสูงอายุเกิน 60 ปี มากกว่าร้อยละ 28 หรือมีผู้สูงอายุเกิน 65 ปี มากกว่าร้อยละ 20 ของประชากรทั้งหมด [1]

การเข้าสู่สังคมสูงอายุอย่างรวดเร็วของประเทศไทยส่งผลให้เกิดความท้าทายด้านสุขภาพอย่างหลีกเลี่ยงไม่ได้ โดยเฉพาะอย่างยิ่งในเรื่องของ กลุ่มโรคไม่ติดต่อเรื้อรัง (Non-Communicable Diseases หรือ NCDs) ซึ่งเป็นสาเหตุการเสียชีวิตอันดับต้น ๆ ของประชากรทั่วโลก ปัจจัยเสี่ยงสำคัญ ได้แก่ การสูบบุหรี่ การไม่ออกกำลังกาย การรับประทานอาหารไม่เหมาะสม และการดื่มแอลกอฮอล์มากเกินไป ซึ่งนำไปสู่โรคหัวใจ มะเร็ง โรคระบบหัวใจ เบาหวาน และความดันโลหิตสูง [2]

ผู้สูงอายุที่ป่วยด้วยโรคไม่ติดต่อเรื้อรัง (NCDs) มักต้องจัดการการใช้ยาที่ซับซ้อน เช่น การจดจำชานิดของยา เวลารับประทานและผลข้างเคียง การลืมทานยาหรือทานผิดเวลาพابได้บ่อย และส่งผลต่อประสิทธิภาพการรักษา ขณะเดียวกัน แอปพลิเคชันสุขภาพที่ว่าไปยังไม่เหมาะสมกับการใช้งานของผู้สูงอายุ โดยเฉพาะในด้านการอ่านแบบหน้าจอ (UI) ที่ซับซ้อน

จากความท้าทายเหล่านี้ การพัฒนาแอปพลิเคชันที่สามารถแก้ไขปัญหาดังกล่าวได้จึงมีความสำคัญอย่างยิ่ง แอปพลิเคชันนี้จะทำหน้าที่เป็นเครื่องมือที่ใช้งานง่าย มีข้อมูลที่นาเข้าเชื่อมต่อ และมีฟังก์ชันที่ช่วยลดความผิดพลาดในการใช้ยา เพื่อช่วยให้ทั้งผู้ป่วยสามารถจัดการสุขภาพได้อย่างมีประสิทธิภาพและปลอดภัยสูงสุดในยุคที่ประเทศไทยกำลังก้าวเข้าสู่สังคมสูงอายุอย่างเต็มตัว

1.2 วัตถุประสงค์

- เพื่อช่วยให้ผู้ใช้สามารถจัดการตารางการทานยาได้อย่างมีประสิทธิภาพ
- เพื่อให้ผู้ใช้สามารถค้นหาข้อมูลยาได้อย่างรวดเร็วและถูกต้อง ผ่านการค้นหาด้วยข้อความ การสแกนباركโค้ด หรือการอ่านข้อความบนฉลากยา (OCR)
- เพื่อสร้างฐานข้อมูลยาที่น่าเชื่อถือ โดยจะรวบรวมข้อมูลยาที่สำคัญจากจากรหบบ Thai Medicine Terminology (TMT) [3]
- เพื่อให้ผู้ใช้สามารถติดตามอาการหลังการใช้ยาได้อย่างเป็นระบบ ผ่านฟังก์ชันการบันทึกอาการและผลข้างเคียงภายหลังการใช้ยา

- เพื่อส่งเสริมความเข้าใจด้านข้อมูลยาแก่ผู้ใช้ จากระบบเสียงอ่านข้อมูลยา (Text-to-Speech)
- เพื่อช่วยให้ผู้ใช้สามารถบันทึกและจัดเก็บประวัติการใช้ยาในเครื่องของตนเองได้

1.3 ขอบเขตงาน

1.3.1 ด้านฟังก์ชันการทำงาน

- ผู้ใช้งานสามารถเพิ่ม, แก้ไข และลบข้อมูลยาที่ทานได้
- ผู้ใช้งานสามารถค้นหาข้อมูลยาเพื่อดูรายละเอียดเพิ่มเติมได้ เช่น ชื่อยา, วิธีใช้, ผลข้างเคียง
- ผู้ใช้งานสามารถตั้งแจ้งเตือนการทานยาตามเวลาที่ต้องการได้
- ผู้ใช้งานสามารถเพิ่มรูปหลังทานเพื่อยืนยันการทานได้
- พัฒนาระบบแปลงข้อความเป็นเสียงสำหรับการอ่านข้อมูลยาให้ผู้ใช้ฟัง (Text-to-Speech)

1.3.2 ด้านข้อมูล

- รวบรวมข้อมูลยาจากระบบ Thai Medicine Terminology (TMT)
- จัดเก็บข้อมูลการทานยาและประวัติสุขภาพของผู้ใช้งานอย่างปลอดภัย
- จัดทำฐานข้อมูลที่มีความสอดคล้องกันของข้อมูลและมีความปลอดภัย

1.3.3 ด้านเทคนิค

- พัฒนาด้วยภาษา Dart และเฟรมเวิร์ก Flutter
- ใช้ฐานข้อมูล PostgreSQL สำหรับจัดเก็บข้อมูลยา และข้อมูลผู้ใช้
- ใช้ PySpark ในการจัดการข้อมูลจากระบบ Thai Medicine Terminology (TMT) ก่อนเก็บลงฐานข้อมูล
- ใช้ Kind เพื่อสร้าง Kubernetes cluster จำลองบนเครื่องพัฒนา (Local Environment) สำหรับทดสอบการทำงานของระบบ Spring Boot Application, PySpark และ ฐานข้อมูล PostgreSQL

1.3.4 เครื่องมือเสริม

- google_mlkit_text_recognition สำหรับอ่านข้อความบนฉลากยา

- mobile_scanner สำหรับสแกนบาร์โค้ด
- flutter_tts สำหรับทำ Text-to-Speech
- flutter_local_notifications สำหรับแจ้งเตือน

1.3.5 ต้านการศึกษา

- การศึกษาเกี่ยวกับการพัฒนาแอปพลิเคชันสำหรับผู้สูงอายุ
- ศึกษาการทำงานของ Flutter และ Dart
- ศึกษาการทำงานของ Spring Boot
- ศึกษาการทำงานของ PySpark
- ศึกษาการทำงานของ kind
- ศึกษาการทำงานของ package google_mlkit_text_recognition
- ศึกษาการทำงานของ package mobile_scanner
- ศึกษาการทำงานของ package flutter_tts
- ศึกษาการทำงานของ package flutter_local_notifications

1.4 ขั้นตอนการดำเนินงาน

โครงการนี้มีการดำเนินงานและการศึกษาที่เกี่ยวข้องแบ่งออกเป็น 6 ขั้นตอนหลัก เพื่อให้การพัฒนาเป็นไปอย่างเป็นระบบ และสามารถตอบโจทย์ความต้องการของผู้ใช้งานได้จริง โดยมีรายละเอียดดังนี้

1.4.1 ขั้นตอนที่ 1: การรวบรวมปัญหา วิเคราะห์ความต้องการของผู้ใช้งานและการออกแบบ

ในขั้นตอนแรกของโครงการ ได้ทำการศึกษาและวิเคราะห์ปัญหาของกลุ่มเป้าหมาย ซึ่งเป็นผู้สูงอายุที่ต้องรับประทานยาหลายชนิดในแต่ละวัน รวมถึงศึกษาข้อจำกัดทางด้านการมองเห็นและการได้ยิน จากนั้นจึงทำการเก็บรวบรวมความต้องการของผู้ใช้งาน (User Requirement) โดยการศึกษาแอปพลิเคชันที่มีอยู่ในตลาด เพื่อดูความคิดเห็นจากผู้ใช้ว่ามีปัญหาหรือข้อจำกัดในส่วนใด และยังขาดฟังก์ชันใดบ้าง เพื่อนำมาวิเคราะห์และกำหนดฟังก์ชันการทำงานของแอปพลิเคชันให้ตอบโจทย์ความต้องการได้อย่างแท้จริง

เมื่อได้ความต้องการของผู้ใช้งานแล้ว ขั้นตอนต่อมาคือการออกแบบระบบ (System Design) เพื่อให้เห็นภาพรวมและโครงสร้างของระบบก่อนการพัฒนา โดยประกอบด้วยส่วนสำคัญดังนี้

1. การออกแบบส่วนติดต่อผู้ใช้ (UI Design) ด้วย Figma

ใช้โปรแกรม Figma ในการออกแบบส่วนติดต่อผู้ใช้ (User Interface) เพื่อสร้างต้นแบบ (Prototype) ของหน้าจอแอปพลิเคชัน โดยเน้นการออกแบบให้ใช้งานง่าย เหมาะกับผู้สูงอายุ และมีสีสันที่มองเห็นได้ชัดเจน พร้อมจัดวางองค์ประกอบให้อยู่ในตำแหน่งที่เข้าถึงได้สะดวก

2. การออกแบบฐานข้อมูล (Database Design) ด้วย ERD (Entity Relationship Diagram)

ทำการออกแบบแผนภาพความสัมพันธ์ของข้อมูล (ERD) เพื่อแสดงความเชื่อมโยงระหว่างข้อมูลยา ผู้ใช้งาน และประวัติการทานยา ซึ่งช่วยให้การจัดเก็บและเรียกใช้งานข้อมูลภายในระบบมีความเป็นระเบียบและสัมพันธ์กันอย่างถูกต้อง

3. การแยกฟังก์ชันการทำงานของระบบ (Functional Decomposition)

ทำการแยกย่อยฟังก์ชันหลักของระบบออกเป็นส่วนย่อย เพื่อช่วยให้เห็นภาพรวมของระบบอย่างเป็นลำดับชั้น และง่ายต่อการแบ่งงานพัฒนาในแต่ละส่วน

4. การออกแบบกระบวนการทำงานของระบบ (Data Flow Diagram: DFD)

สร้างแผนภาพแสดงการไหลของข้อมูล (DFD) เพื่อแสดงการทำงานของระบบในแต่ละกระบวนการ โดยแสดงให้เห็นถึงการรับข้อมูลจากผู้ใช้ การประมวลผล และการจัดเก็บในฐานข้อมูลอย่างเป็นระบบ

1.4.2 ขั้นตอนที่ 2: พัฒนาระบบในส่วนของหน้าบ้าน (Frontend)

ในขั้นตอนนี้ได้ทำการออกแบบและพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android โดยใช้ Flutter Framework ร่วมกับภาษา Dart โดยพัฒนาหน้าจอต่าง ๆ เช่น หน้าลงทะเบียนและเข้าสู่ระบบ หน้าหลักสำหรับแสดงรายการและเวลาในการรับประทาน รวมถึงหน้าการแจ้งเตือนการรับประทานยา หลังจากนั้นทำการเชื่อมต่อกับระบบ API เพื่อให้สามารถดึงและบันทึกข้อมูลจากส่วน Backend ได้อย่างถูกต้อง

1.4.3 ขั้นตอนที่ 3: การพัฒนาระบบในส่วนของหลังบ้าน (Backend)

ในส่วนของหลังบ้าน ใช้ Spring Boot Framework ในการสร้าง REST API ที่เชื่อมต่อระหว่างแอปพลิเคชันกับฐานข้อมูล PostgreSQL ซึ่งทำหน้าที่เก็บข้อมูลยา ข้อมูลผู้ใช้ ตารางการทานยา และข้อมูลการบันทึกประวัติการรับประทานยา โดยออกแบบ API endpoints ให้รองรับการทำงานต่างๆ เช่น การเพิ่ม แก้ไข และลบข้อมูลยา การตั้งค่าการแจ้งเตือน และการดึงข้อมูลจากฐานข้อมูล นอกจากนี้ เพื่อให้ระบบสามารถทดสอบได้ในสภาพแวดล้อมเสมือนจริง ได้มีการใช้เครื่องมือ Kind (Kubernetes in Docker) ในการจำลองคลัสเตอร์ Kubernetes ภายในเครื่อง เพื่อทดสอบการทำงานของระบบในรูปแบบ Container

1.4.4 ขั้นตอนที่ 4: การศึกษาและพัฒนา Data Pipeline ด้วย PySpark และ Kind

ขั้นตอนนี้เป็นการออกแบบและพัฒนาระบวนการ Data Pipeline เพื่อจัดการข้อมูลยาจากแหล่งข้อมูลภายนอก โดยเฉพาะข้อมูลจากระบบ Thai Medicine Terminology (TMT) จากสำนักพัฒนามาตรฐานระบบข้อมูลสุขภาพไทย ซึ่งให้ข้อมูลภายในรูปแบบไฟล์ Excel โดยใช้ PySpark ในการประมวลผล ทำความสะอาดข้อมูล (Data Cleaning) และแปลงข้อมูลให้อยู่ในรูปแบบที่

เหมาะสมก่อนนำเข้าฐานข้อมูล PostgreSQL เพื่อให้ระบบ Data Pipeline สามารถทำงานร่วมกับส่วน Backend และ Database ได้อย่างราบรื่น ได้มีการนำ Kind มาใช้ควบคุมการทำงานของ Container ทั้งหมด โดยภายใน Cluster จะประกอบด้วย

1. Container ของ PySpark สำหรับประมวลผลและจัดการข้อมูลยา
2. Container ของ Spring Boot สำหรับให้บริการ API
3. Container ของ PostgreSQL สำหรับเก็บข้อมูลที่ผ่านการประมวลผลแล้ว

1.4.5 ขั้นตอนที่ 5: การทดสอบและปรับปรุงระบบ

ในขั้นตอนนี้ทำการทดสอบระบบทั้งหมด เพื่อตรวจสอบความถูกต้อง และประสิทธิภาพของแอปพลิเคชัน โดยแบ่งการทดสอบออกเป็น Unit Testing สำหรับทดสอบฟังก์ชันแต่ละส่วน Integration Testing สำหรับทดสอบการทำงานร่วมกันระหว่างส่วนต่างๆ และ User Acceptance Testing (UAT) โดยให้กลุ่มผู้สูงอายุทดลองใช้งานจริง 以便นั้นรวม Feedback และข้อเสนอแนะจากผู้ทดสอบเพื่อนำมาปรับปรุงแก้ไขระบบให้มีความสมบูรณ์และตอบสนองความต้องการของผู้ใช้งานได้ดียิ่งขึ้น

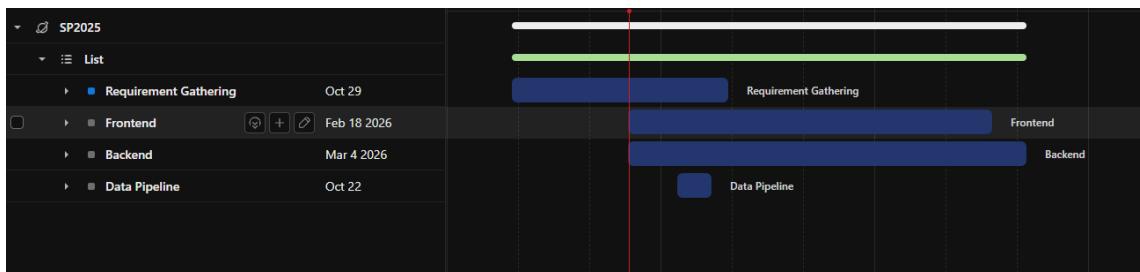


Figure 1.1: แผนการดำเนินโครงการ
[4]

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้ใช้สามารถค้นหาข้อมูลยาได้อย่างรวดเร็วผ่านการค้นหาด้วยข้อความ การสแกนباركโค้ด หรือการอ่านข้อความบนฉลากยา (OCR)
2. ระบบแจ้งเตือนการทานยาตามเวลาที่กำหนดอย่างแม่นยำ ช่วยลดปัญหาการลืมทานยา การทานยาช้า หรือทานยาไม่ตรงเวลา
3. ผู้ใช้สามารถบันทึกอาการและผลข้างเคียงที่เกิดขึ้นหลังการใช้ยา ซึ่งช่วยให้สามารถติดตามผลของยาและประเมินความปลอดภัยในการใช้ยาได้อย่างเป็นระบบ
4. การออกแบบ UI ที่ใช้งานง่าย เหมาะกับผู้สูงอายุ พร้อมระบบเสียงอ่านข้อมูลยา (Text-to-Speech)
5. แอปพลิเคชันสามารถเก็บประวัติการใช้ยาและการแจ้งเตือนต่าง ๆ ในเครื่อง ทำให้ผู้ใช้สามารถเรียกดูข้อมูล

ບທທີ່ 2 ທຖວາທີ່ເກີ່ວຂ້ອງ

2.1 ສັນຄມສູງວັຍ (Aged Society ແລະ Super-Aged Society)

ຄາດວ່າປະມານປີ พ.ສ. 2576 ປະເທດໄທຍະກ້າວເຂົ້າສູ່ ສັນຄມສູງວັຍຮະດັບສຸດຍອດ (Super-Aged Society) ຜຶ່ງມີປະຊາກອາຍຸ
ເກີນ 60 ປີ ມາກກວ່າຮ້ອຍລະ 28 ຂອງປະຊາກທັງໝົດ [1] ການເປົ້າມີແຜນນີ້ທີ່ໃຫ້ເກີດປັບປຸງຫາດ້ານສຸຂພາພແລະຄວາມຕ້ອງການເທິນໂລຍື
ເພື່ອຊ່າຍເຫຼືອກາຮຸແລະຜູ້ສູງອາຍຸ ໂດຍເພາະໃນເຮືອງ ການຈັດການການໃໝ່ຢາ ຜຶ່ງມີຄວາມໜັບໜັນແລະມີຄວາມເສື່ອງຕ່າງໆ

2.2 ໂຮມໄມ່ຕິດຕ່ອເຮືອຮັງ (Non-Communicable Diseases: NCDs)

ຜູ້ສູງອາຍຸຈຳນວນນັກປ່ວຍດ້ວຍໂຮມເຮືອຮັງ ເຊັ່ນ ຄວາມດັນໂລທິສູງ ເບາຫວານ ໂຮມທັງໃຈ ຜຶ່ງຕ້ອງໃໝ່ຢາຫລາຍໜິດຕ່ອນເນື່ອງເປັນເວລາ
ນານ [2] ການລື້ມທານຍາຫຼືອທານພິດເວລາອາຈສັງເປດຕ່ອປະສິດທິພາບການຮັກໝາ ແລະກ່ອໃຫ້ເກີດກວະແທຮກ້ອນ ການພັນນາແອປພັບເຄີ່ມ
ທີ່ຊ່າຍ ແຈ້ງເຕືອນແລະບັນທຶກປະວັດທິການໃໝ່ຢາ ຈຶ່ງຕອບໂຈທີການຈັດການໂຮມເຮືອຮັງອ່າຍ່າມໍຮັບປະ

2.3 ການອອກແບບຮະບບທີ່ເໝາະສົມກັບຜູ້ສູງອາຍຸ (Human-Computer Interaction: HCI)

ຜູ້ສູງອາຍຸມີຂໍອຈຳກັດທັງດ້ານ ສາຍຕາ ການໄດ້ຍືນ ຄວາມຈຳ ແລະ ຄວາມຄຸນເຄີຍກັບເທິນໂລຍື ການອອກແບບປົງສັນພັນຮະຫວ່າງ
ມຸນຸຍົກກັບຄອມພິວເຕັນ (HCI) ເປັນສາຂາວິชาທີ່ເນັນການທຳຄວາມເຂົ້າໃຈຜູ້ໃໝ່ ແລະ ນຳຄວາມເຂົ້າໃຈນັ້ນມາອອກແບບຮະບບໃຫ້ສອດຄລ້ອງກັບ
ພົກທິກຣມແລະຄວາມຕ້ອງການຂອງຜູ້ໃໝ່ຈຳນວນ ຮັງຈຸນີ້ກ່ອນວ່າຜູ້ໃໝ່ມີຄວາມຮັກໝາໃຫ້ຮັກໝາໄວ້ແລະມີພົກທິກຣມອ່າງໃຈ [5] ຜຶ່ງສໍາຫັບຜູ້ໃໝ່
ກຸລຸ່ມຜູ້ສູງອາຍຸ ຈະຕ້ອງຄຳນິ້ງຄືນິ້ງປັບປຸງໃຫ້ຜູ້ໃໝ່ໃຫ້ຮັກໝາໄວ້ແລະມີຄວາມຮັກໝາໃຫ້ຮັກໝາໄວ້ແລະມີພົກທິກຣມອ່າງໃຈ
ຈຳ ການອອກແບບແອປພັບເຄີ່ມເຈັດຕ້ອງຍືດຫຼັກ Accessible Design ໄດ້ແກ່

1. ຂາດຕ້ວອັກຊະຮີ່ຢ່າງເຈົ້າ
2. ສີສັນຕັດກັນຂັດເຈນ (Contrast)
3. ຮະບບແຈ້ງເຕືອນຫລາຍຮູປແບບ (ເສີ່ງ, ສັ່ນ, ໄຟແຈ້ງເຕືອນ)

2.4 ການຈັດການການໃໝ່ຢາ

[6] ການຈັດການຮະບບຍາເປັນສິ່ງສຳຄັນອ່າງຍິ່ງທີ່ເກີ່ວຂ້ອງກັບຄວາມປລອດກັຍຂອງຜູ້ປ່ວຍຫຼືອຜູ້ໃໝ່ບໍລິການໃນໂຮງພຍາບາລ ຮະບບ
ຍາທີ່ຄຣອບຄຸມສະຫຼອນໃຫ້ເຫັນຄືການທຳມານຮ່ວມກັນຂອງທີ່ມສຫສາວິຊາໜີພ ເພື່ອໃຫ້ການບໍລິການຈັດການຮະບບຍາໃນໂຮງພຍາບາລມີ
ປະສິດທິພາບແລະປລອດກັຍ

ເກີ່ວຂ້ອງກັບຄວາມປລອດກັຍ:

1. โรงงานผู้ผลิตที่ได้รับการรับรอง GMP (Good Manufacturing Practice)
2. ยามีคุณลักษณะเฉพาะตามมาตรฐานที่ได้รับการรับรองในประเทศไทย
3. มีเป็นเครื่องวัดที่สามารถวัดความถูกต้องและความสมบูรณ์ของข้อมูล และช่วยให้สามารถเรียกใช้หรือปรับปรุงข้อมูลได้อย่างมีประสิทธิภาพ โดยหลักการออกแบบฐานข้อมูลที่ดี มีดังนี้

2.5 พื้นฐานและการออกแบบฐานข้อมูล

[7] การออกแบบฐานข้อมูลที่ดีมีความสำคัญอย่างยิ่งต่อความถูกต้องและความสมบูรณ์ของข้อมูล และช่วยให้สามารถเรียกใช้หรือปรับปรุงข้อมูลได้อย่างมีประสิทธิภาพ โดยหลักการออกแบบฐานข้อมูลที่ดี มีดังนี้

1. ลดความซ้ำซ้อนของข้อมูล (Data Redundancy)
2. เพิ่มความถูกต้องสมบูรณ์ของข้อมูล (Data Integrity)
3. สามารถปรับปรุงหรือเปลี่ยนแปลงโครงสร้างได้ง่าย

2.6 Optical Character Recognition (OCR)

[8] หรือที่รู้จักกันในภาษาไทยคือ ”การรู้จำอักษรด้วยแสง” คือเทคโนโลยีที่ใช้ในการแปลงข้อความจากรูปแบบภาพ (เช่น รูปถ่าย, ภาพสแกน, ไฟล์ PDF ที่เป็นภาพ) ให้กลายเป็นข้อความดิจิทัลที่คอมพิวเตอร์สามารถอ่าน, แก้ไข, ค้นหา หรือนำไปใช้ประมวลผลต่อได้

ประเภทของ OCR

1. ซอฟต์แวร์รู้จำอักษรด้วยแสงอย่างง่าย : ใช้อัลกอริทึมการจับคู่รูปแบบเพื่อเบรย์เบที่บูรณาการข้อมูลภายในแบบอักษรที่ลงทะเบียน
2. ซอฟต์แวร์รู้จำอักษรแบบอัจฉริยะ (Intelligent Character Recognition: ICR) : ใช้ Machine Learning และ Neural Network เพื่ออ่านลายมือหรือตัวอักษรที่ซับซ้อน
3. การรู้จำคำแบบอัจฉริยะ (Intelligent Word Recognition) : วิเคราะห์คำทั้งคำแทนการประมวลผลที่ลงทะเบียนตัวอักษร
4. การรู้จำเครื่องหมายด้วยแสง (Optical Mark Recognition: OMR) : ใช้ระบุสัญลักษณ์หรือเครื่องหมาย เช่น โลโก้ หรือตัวเลือกในแบบฟอร์มสอบ

ข้อดีของ OCR

1. แปลงเอกสารกระดาษเป็นข้อมูลดิจิทัลที่สามารถค้นหาได้
2. สามารถประมวลผลข้อมูลตัวอักษรโดยอัตโนมัติ เพื่อใช้ในการวิเคราะห์และประมวลผลความรู้เพิ่มเติม
3. ลดเวลาการประมวลผลเอกสารและการป้อนข้อมูลด้วยตนเอง

4. OCR ມັກຄູຮ່ວມເຂົາກັບເຕົກໂນໂລຢີ AI ເພື່ອປະຍຸດໃຫ້ຈຳນວນສູງ ເຊັ່ນ ອຳນັດວຽກແບ່ນແບ່ນ ແລະ ປ້າຍຈາກຮູບໃຈ (OCR) (ການຮູ້ຈຳອັກຊະໜາດວິເສດງ) ສື່ບັນດາໂດຍ AWS

2.7 PostgreSQL

[7] PostgreSQL ເປັນຮະບບັດກາຮຽນຂໍ້ມູນເຊີງສັນພັນ (Relational Database Management System: RDBMS) ຮະດັບ ອົງຄ່ອງທີ່ມີຄວາມເສດຖຽນແລະ ເຊື່ອສື່ບັນດາໄດ້ສູງ ໂດຍຮອງຮັບການທຳມະນຸດແບບ ACID (Atomicity, Consistency, Isolation, Durability) ອິຍ່າງ ສມບູຽນ ທີ່ຈຶ່ງໝາຍເສີ່ງຄຸນສົມບັດພື້ນຮຽນທີ່ທໍາໃຫ້ການທຳມະນຸດ (Transaction) ໃນຮຽນຂໍ້ມູນມີຄວາມຖຸກຕ້ອງແລະປົດກັບ ໄດ້ແກ່:

1. Atomicity ສື່ບັນດາ ຖໍ່ການທຳມະນຸດ ຈະຕ້ອງສໍາເລົາຈັກທັງໝົດຫຼືໄມ່ສໍາເລົາເລີຍ ບໍ່ກີດຂໍ້ມູນພາດ ຮະບບຈະຍົກເລີກ (Rollback) ການ ທຳມະນຸດທັງໝົດເພື່ອໃຫ້ຂໍ້ມູນຄົກສະພາບເດີມ
2. Consistency ສື່ບັນດາ ທີ່ລັດການທຳມະນຸດ ຂໍຂໍ້ມູນໃນຮຽນຂໍ້ມູນຈະຕ້ອງຍັງຄົງຖຸກຕ້ອງຕາມກົງແລະ ຊົ່ວໂມງ (Constraints) ທີ່ກຳນົດໄວ້
3. Isolation ສື່ບັນດາ ທີ່ການທຳມະນຸດຫລາຍຮາຍການທີ່ເກີດຂຶ້ນພ້ອມກັນຈະໄໝສ່ວນຜົນກະທົບຕ່ອງກັນ ທຳໃຫ້ຜົນກະທົບເຖິງກົນທີ່ກຳນົດໄວ້
4. Durability ສື່ບັນດາ ເມື່ອທຳມະນຸດສົ່ງຈິງ ຂໍຂໍ້ມູນທີ່ເປີດແປງແລ້ວຈະຖຸກບັນທຶກການໃນຮະບບ ແມ່ຮະບບຈະປົດຕົວຫຼືໄດ້ກີດຄວາມ ຂັດຂ້ອງ

PostgreSQL ສາມາດຮອງຮັບການປະມາລຜົດໄດ້ທີ່ສອງຮູປແບບຫລັກຂອງຮະບບຮຽນຂໍ້ມູນ ສື່ບັນດາ:

1. OLTP (Online Transaction Processing): ໃໃໝ່ສໍາຮັບງານທີ່ຕ້ອງການການປະມາລຜົດຮູກຮົມຈຳນວນນາກຍ່າງຮວດເຮົວ ເຊັ່ນ ຮະບບຮານາຄາຮ່າງສິນຄ້າ ທີ່ຮັບການຈອງຕ່າງໆ ທີ່ມີການເພີ່ມ ລບ ທີ່ແກ້ໄຂຂໍ້ມູນລອ່ຽມຕົວເວລາ
2. OLAP (Online Analytical Processing): ໃໃໝ່ສໍາຮັບກາວິເຄຣະທີ່ຂໍ້ມູນເຊີງລຶກ ເຊັ່ນ ການສຽບປອດໄພ ກາວິເຄຣະທີ່ແນວໂນ້ມ ທີ່ຮັບການກົດລົງຂໍ້ມູນຈຳນວນນາກຈາກຫລາຍແລ້ວມາຄຳນວນຮົມກັນ

ອອກຈາກນີ້ PostgreSQL ຍັງຮອງຮັບການທຳມະນຸດຂໍ້ມູນແພລຕົວໂຮມໄດ້ ໂດຍສາມາດຕິດຕັ້ງແລະໃຫ້ຈຳນວນຫລາຍຮະບບປົກປົກຕິການ ເຊັ່ນ Windows, Linux, macOS, FreeBSD ແລະ OpenBSD ວິວທີ່ຍັງມີຄວາມສາມາດໃນການຂໍ້ມູນໄພ່ເພີ່ມເຕີມ (Extensibility) ດ້ວຍການ ສ້າງຟັງກົນຂໍ້ມູນໃໝ່ ທີ່ມີການເພີ່ມຂໍ້ມູນໃໝ່ (Extensions) ເພື່ອເພີ່ມຂໍ້ມູນສາມາດໃຫ້ຕຽບຕ່າງໆ ແລະ ຕ້ອງການຂອງຜູ້ໃຊ້ຈຳນວນອື່ນດ້ວຍ

2.7.1 ຄຸນສົມບັດທີ່ສຳຄັນ

ຮອງຮັບຂໍ້ມູນທີ່ທາງຫລາຍ

1. ຂໍຂໍ້ມູນເຊີງໂຄຮ່ວງສ້າງ (Structured Data) ເຊັ່ນ Integer, Text, Date
2. ຂໍຂໍ້ມູນເຊີງໄມ້ໂຄຮ່ວງສ້າງ ເຊັ່ນ JSON ແລະ JSONB
3. ຂໍຂໍ້ມູນເຊີງພື້ນທີ່ (Spatial Data) ຜ່ານສ່ວນເສີມ PostGIS

การทำงานแบบ Multi-Version Concurrency Control (MVCC)

ช่วยให้ผู้ใช้หลายคนเข้าถึงข้อมูลพร้อมกันโดยไม่เกิดการขัดแย้ง

2.7.2 ประโยชน์ของ PostgreSQL

- สามารถใช้งานได้โดยไม่จำเป็นต้องซ่ารัจเเจนและเป็น Open Source ไม่มีค่าใช้จ่ายด้านลิขสิทธิ์
- มีความน่าเชื่อถือสูง รองรับการทำงานต่อเนื่อง (Fault Tolerant)
- รองรับหลายภาษาโปรแกรม เช่น Python, Java, C/C++, PHP, Ruby, Perl, Erlang, Lua, Scala, Haskell, etc.
- รองรับการทำงานข้ามแพลตฟอร์ม

2.8 สถาปัตยกรรมแบบ Client-Server (Client-Server Architecture)

สถาปัตยกรรมแบบ Client-Server เป็นรูปแบบการออกแบบระบบคอมพิวเตอร์ที่แบ่งการทำงานออกเป็น 2 ส่วนหลัก คือ Client (ผู้ใช้) และ Server (เซิร์ฟเวอร์) โดยทั้งสองส่วนจะสื่อสารกันผ่านเครือข่าย (Network) เพื่อแลกเปลี่ยนข้อมูลและประมวลผลงานร่วมกัน

2.8.1 องค์ประกอบหลัก

Client (ผู้ใช้)

คือแอปพลิเคชันหรือโปรแกรมที่ทำงานบนอุปกรณ์ของผู้ใช้ เช่น มือถือ คอมพิวเตอร์ หรือแท็บเล็ต ในโครงงานนี้ แอปพลิเคชันที่พัฒนาด้วย Flutter และ Dart ทำหน้าที่เป็น Client ซึ่งทำงานบนอุปกรณ์ Android ของผู้ใช้ มีหน้าที่หลักดังนี้

- แสดงส่วนติดต่อผู้ใช้งาน (User Interface)
- รับข้อมูลจากผู้ใช้ เช่น การกรอกข้อมูล การคลิกปุ่ม
- ส่งคำขอ (Request) ไปยัง Server เพื่อขอข้อมูลหรือให้ประมวลผล
- รับและแสดงผลข้อมูลที่ได้จาก Server

Server (เซิร์ฟเวอร์)

คือระบบที่ทำงานบนเครื่องแม่ข่าย มีหน้าที่รับคำขอจาก Client และประมวลผลตามคำขอนั้น ในโครงงานนี้ Spring Boot Application ทำหน้าที่เป็น Server ที่ให้บริการ REST API และเชื่อมต่อกับฐานข้อมูล PostgreSQL ได้แก่

- รับและประมวลผลคำขอ (Request) จาก Client
- เข้าถึงและจัดการฐานข้อมูล (Database)
- ตรวจสอบความถูกต้องของข้อมูล (Validation)

4. ประมวลผลตรรกะทางธุรกิจ (Business Logic)

5. ส่งผลลัพธ์ (Response) กลับไปยัง Client

2.8.2 การทำงานของระบบ Client-Server ในโครงงาน

การสื่อสารระหว่าง Client และ Server ในโครงงานนี้เป็นไปตามขั้นตอนดังนี้

- ผู้ใช้ติดต่อกับแอปพลิเคชัน Flutter (Client) เช่น ค้นหาข้อมูลยา หรือบันทึกการทำยา
- Client ส่งคำขอ HTTP Request ไปยัง Server ผ่าน REST API
- Spring Boot Server รับคำขอและประมวลผล โดยอาจเข้าถึงฐานข้อมูล PostgreSQL เพื่อดึงหรือบันทึกข้อมูล
- Server ส่งผลลัพธ์กลับมาในรูปแบบ HTTP Response (มักเป็น JSON)
- Client รับข้อมูลและแสดงผลบนหน้าจอให้ผู้ใช้เห็น

2.8.3 ข้อดีของสถาปัตยกรรม Client-Server

- การแยกหน้าที่ชัดเจน: Client รับผิดชอบส่วนแสดงผล ขณะที่ Server จัดการข้อมูลและตรรกะทางธุรกิจ
- ความปลอดภัย: ข้อมูลสำคัญและตรรกะทางธุรกิจถูกเก็บไว้ที่ Server ไม่ถูกเปิดเผยบน Client
- ง่ายต่อการบำรุงรักษา: สามารถอัปเดต Server โดยไม่ต้องแก้ไข Client (และในทางกลับกัน) ตราบใดที่ API ยังคงเหมือนเดิม
- รองรับผู้ใช้หลายคน: Server สามารถให้บริการ Client หลายเครื่องพร้อมกันได้
- ความยืดหยุ่น: สามารถพัฒนา Client หลายแพลตฟอร์ม (iOS, Android, Web) ที่ใช้ Server เดียวกัน
- การจัดการข้อมูลแบบรวมศูนย์: ข้อมูลทั้งหมดถูกจัดเก็บและจัดการที่ Server ทำให้ง่ายต่อการสำรองข้อมูลและการรักษาความสอดคล้อง

2.8.4 การประยุกต์ใช้ในโครงงาน

โครงงานนี้ใช้สถาปัตยกรรม Client-Server โดยมีองค์ประกอบดังนี้

- Client: แอปพลิเคชัน Flutter บน Android ใช้สำหรับแสดงข้อมูลยา, แจ้งเตือนการทำยา, บันทึกประวัติ
- Server: Spring Boot Application ใช้สำหรับจัดการ REST API, ตรรกะทางธุรกิจ, การเข้าถึงฐานข้อมูล
- Database: PostgreSQL ใช้สำหรับจัดเก็บข้อมูลยา, ข้อมูลผู้ใช้, ประวัติการทำยา
- Data Pipeline: PySpark ใช้สำหรับประมวลผลข้อมูลจาก Thai Medicine Terminology (TMT) ก่อนนำเข้าฐานข้อมูล

2.9 Spring Boot Framework

[9] Spring Boot เป็น Framework ที่พัฒนาโดย Pivotal (ปัจจุบันเป็นส่วนหนึ่งของ VMware) สำหรับการสร้างแอปพลิเคชัน Java แบบ standalone ที่พร้อมใช้งานได้ทันที โดย Spring Boot ช่วยลดความซับซ้อนในการตั้งค่าและพัฒนาแอปพลิเคชัน Spring แบบดั้งเดิม ทำให้นักพัฒนาสามารถสร้างแอปพลิเคชันได้รวดเร็วขึ้น

2.9.1 คุณสมบัติหลัก

1. Auto-Configuration: ตั้งค่าอัตโนมัติตาม dependencies ที่มีในโปรเจกต์
2. Embedded Server: มี Web Server ในตัว เช่น Tomcat, Jetty ไม่ต้องติดตั้งแยก
3. Production-Ready: มีเครื่องมือสำหรับ monitoring, health checks, และ metrics
4. Microservices Support: รองรับการพัฒนา Microservices Architecture

2.9.2 การใช้งานในโครงงาน

ในโครงงานนี้ Spring Boot ถูกใช้เพื่อ

1. สร้าง REST API endpoints สำหรับการสื่อสารระหว่าง Flutter Client และฐานข้อมูล
2. จัดการ CRUD operations (Create, Read, Update, Delete) สำหรับข้อมูลรายละเอียดผู้ใช้
3. ตรวจสอบความถูกต้องของข้อมูล (Validation)
4. จัดการการเชื่อมต่อกับ PostgreSQL Database

2.10 Flutter Framework

[10] เป็น Framework Open Source เปิดตัวในปี 2018 ที่พัฒนาโดย Google ใช้สร้างอินเทอร์เฟซผู้ใช้ (UI) ของแอปพลิเคชันสำหรับแพลตฟอร์ม iOS, Android, Web, Windows, macOS และ Linux โดยใช้ Codebaseเดียวในการสร้างแอปมือถือ เว็บ และเดสก์ท็อป (Windows, macOS, Linux) ได้ ซึ่ง Flutter ช่วยลดความซับซ้อนของการสร้าง UI ที่สวยงามและสอดคล้องกัน บนหลายแพลตฟอร์ม

2.10.1 เปรียบเทียบการพัฒนาแอปแบบเนทีฟ (Native) และข้ามแพลตฟอร์ม (Cross-platform)

Native

1. เขียนแยกเฉพาะแต่ละแพลตฟอร์ม เช่น Swift สำหรับ iOS, Kotlin/Java สำหรับ Android
2. สามารถเข้าถึงฟังก์ชันของอุปกรณ์ได้อย่างเต็มประสิทธิภาพ
3. มีประสิทธิภาพสูง เนื่องจากพัฒนาแพลตฟอร์มอย่างเฉพาะเจาะจง แต่ต้องเขียนหลายโค้ดเบส และมีค่าใช้จ่ายที่สูง

Cross-platform

1. สามารถใช้โค้ดเบสเดียวกันสำหรับหลายแพลตฟอร์ม
2. ประหยัดเวลาและค่าใช้จ่าย
3. อาจเข้าถึงฟังก์ชันบางอย่างของอุปกรณ์ได้อย่างจำกัด
4. ประสบการณ์การใช้งานของผู้ใช้สอดคล้องกัน

2.10.2 ข้อดีของ Flutter

1. ประสิทธิภาพใกล้เคียงแบบ Native และมีค่าใช้จ่ายที่ต่ำกว่า
2. Render UI เร็วและสม่ำเสมอ เนื่องจากใช้ Engine Graphic ของตัวเอง แสดงผล UI สอดคล้องกันข้ามแพลตฟอร์ม
3. รองรับการทำงานขนาน (Parallel Processing)
4. มีการสนับสนุนการพัฒนาโดยทีมงานของ Google ที่สนับสนุนการพัฒนาและรองรับการใช้งาน

2.10.3 Flutter Text-to-Speech (TTS) Plugin

Flutter TTS (flutter_tts) เป็น public package ที่ช่วยให้นักพัฒนาสามารถนำความสามารถในการแปลงข้อความเป็นเสียงพูด (Text-to-Speech: TTS) มาใช้ภายในแอปพลิเคชันได้ โดยรองรับหลายแพลตฟอร์ม ได้แก่ Android, iOS, Web, Windows และ macOS ทำให้เหมาะสมกับการพัฒนาแอปข้ามแพลตฟอร์มด้วย Flutter

คุณสมบัติหลัก

1. พูดข้อความ (speak)
2. ดึงรายการภาษาที่รองรับ (getLanguages)
3. ตรวจสอบว่าภาษาหนึ่นใช้งานได้หรือไม่ (isLanguageAvailable)
4. กำหนดภาษา (setLanguage)
5. เลือกเสียง (voice) ที่เหมาะสมกับภาษาและสำเนียง
6. ความเร็วในการพูด (setSpeechRate)
7. ระดับเสียง (setVolume)
8. ระดับโทนเสียง (setPitch)

ข้อดี

1. ลดภาระการอ่านข้อความยาว ๆ โดยเฉพาะในเอกสารกำกับยา
2. ช่วยผู้ใช้ที่อ่านภาษาได้ไม่คล่อง แต่สามารถที่จะฟังและเข้าใจ

2.10.4 flutter_local_notifications

เป็นปลั๊กอินของ Flutter ที่ใช้สำหรับ การแสดงผลการแจ้งเตือน (Local Notifications) บนอุปกรณ์ต่าง ๆ โดยไม่ต้องมีการเชื่อมต่ออินเทอร์เน็ต เนื่องจากสามารถรับแจ้งเตือนที่เกิดขึ้นภายในเครื่อง และสามารถนำมายังแอปพลิเคชันที่ซับซ้อน

คุณสมบัติหลัก

1. แสดงการแจ้งเตือนพื้นฐาน (basic notifications)
2. การตั้งเวลาแจ้งเตือน (Scheduled Notification) เช่น แจ้งเตือนรายวัน รายสัปดาห์ หรือแจ้งเตือนช้าเป็นช่วงเวลา

2.10.5 google_mlkit_text_recognition

เหมาะสมสำหรับแอปที่ต้องการ สแกนและแปลงข้อความจากรูปภาพอย่างแม่นยำ, รองรับหลายภาษา, และ ทำงานออฟไลน์

ข้อดี

1. ใช้โมเดล Machine Learning ของ Google ทำให้การอ่านตัวอักษรจากภาพแม่นยำ แม้ภาพไม่ชัดเจนหรือมีเงา
2. มี API ที่เข้าใจง่าย สามารถรวมเข้ากับ Flutter ได้สะดวก
3. ทำงานแบบ On-device ไม่ต้องส่งรูปภาพขึ้นไปประมวลผลบนเซิร์ฟเวอร์ ทำให้มีความรวดเร็ว, ประหยัดข้อมูล, และปลอดภัย ต่อข้อมูลส่วนตัว
4. สามารถสแกนข้อความจากกล้องได้ทันที (Live Camera OCR)

คุณสมบัติเด่น

1. สามารถแปลงรูปภาพให้เป็นข้อความโดยไม่ต้องส่งข้อมูลไปยังเซิร์ฟเวอร์
2. สามารถแบ่งข้อความออกเป็นบล็อก, บรรทัด, หรือองค์ประกอบตัวอักษรย่อย ๆ เพื่อการประมวลผลที่ละเอียด
3. สแกนและแสดงข้อความจากกล้องแบบเรียลไทม์ได้

2.10.6 mobile_scanner plugin

เป็น Flutter plugin สำหรับสแกน Barcode ด้วยกล้องของอุปกรณ์ โดยรองรับทั้ง Android, iOS, macOS, Web (แต่ไม่รองรับ Linux และ Windows) จุดเด่นของปลั๊กอินนี้คือ ประสิทธิภาพสูง, น้ำหนักเบา, และปรับแต่งได้ง่าย

คุณสมบัติเด่น

- สแกนبارك็อคได้เร็ว รองรับการตรวจจับแบบ real-time
- รองรับหลายรูปแบบของبارك็อค เช่น QR Code, Code128, EAN-13 เป็นต้น
- ปรับแต่งกล้องและตัวสแกนได้ เช่น ความละเอียดของกล้อง, ความเร็วการตรวจจับ, เปิด/ปิดแฟลช, การกลับภาพ, การซูม อัตโนมัติ (Flutter คืออะไร โดย AWS, 2567)

2.11 Dart

[11] เป็นภาษาโปรแกรมที่ออกแบบและพัฒนาโดย Google โดยมีเป้าหมายเพื่อรองรับการพัฒนา Client Application บนหลายแพลตฟอร์ม ทั้ง มือถือ, เว็บ และ เดสก์ท็อป (Windows, macOS, Linux) โดยเน้นที่ความ รวดเร็ว, มีประสิทธิภาพ, และ ง่าย ต่อการพัฒนา และยังมีความสามารถในการ คอมไพล์ไปยังหลายแพลตฟอร์ม ได้แก่ Dart Native และ Dart Web

2.11.1 คุณสมบัติเด่นของ Dart

- Type Safety (ระบบความปลอดภัยของชนิดข้อมูล)

Dart ใช้ Static Type Checking เพื่อตรวจสอบความถูกต้องของชนิดข้อมูลตั้งแต่ตอนคอมไพล์ ทำให้ลดข้อผิดพลาดขณะรัน โปรแกรม

- Null Safety (ระบบป้องกันค่า Null)

Dart มีระบบป้องกัน Null ที่บังคับใช้ได้อย่างเคร่งครัด ตัวแปรที่ไม่ได้ระบุว่าสามารถเป็น Null ได้ จะไม่สามารถเก็บค่า Null ได้เลย ป้องกัน Null Reference Error ซึ่งเป็นปัญหาคลาสสิกในภาษาอื่น ๆ

- รองรับการเขียนโปรแกรมแบบ Asynchronous

โดยมีคำสำคัญ เช่น async, await, และชนิดข้อมูล Future รวมถึง Stream ที่ช่วยให้นักพัฒนาจัดการงานแบบไม่ประสาน เวลา (เช่น ดึงข้อมูลจาก API หรือทำงาน I/O) ได้อย่างสะดวก

2.12 คำศัพท์

- แอปพลิเคชัน (Application): โปรแกรมคอมพิวเตอร์ที่ถูกสร้างขึ้นมาเพื่อทำงานบางอย่างบนอุปกรณ์
- เฟรมเวิร์ก (Framework): ชุดเครื่องมือและโครงสร้างสำเร็จรูปที่ช่วยให้นักพัฒนาสามารถสร้างโปรแกรมได้รวดเร็วและเป็น ระบบมากขึ้น
- Codebase: ชุดไฟล์คัดหลักทั้งหมดที่ใช้ในการสร้างโปรแกรมหนึ่ง ๆ
- Native: การพัฒนาแอปพลิเคชันที่สามารถใช้โค้ดเดียวทั้งหมดบนระบบปฏิบัติการที่แตกต่างกันได้
- Cross-platform: การพัฒนาแอปพลิเคชันที่สามารถใช้โค้ดเดียวทั้งหมดบนระบบปฏิบัติการที่แตกต่างกันได้

6. Backend: ส่วนการทำงานเบื้องหลังของแอปพลิเคชัน ทำหน้าที่จัดการข้อมูล, ประมวลผลคำสั่ง, และสื่อสารกับฐานข้อมูล
7. Frontend: ส่วนที่ผู้ใช้มองเห็นและใช้งานโดยตรง เช่น หน้าจอแอปพลิเคชัน, ปุ่มกด, หรือรูปภาพต่าง ๆ
8. API (Application Programming Interface): ตัวกลางที่ช่วยให้โปรแกรมหรือระบบที่แตกต่างกันสามารถสื่อสารและแลกเปลี่ยนข้อมูลกันได้
9. RESTful API: รูปแบบมาตรฐานในการออกแบบ API ที่ใช้สื่อสารผ่านอินเทอร์เน็ต โดยมีคำสั่งหลักๆ เช่น GET (ขอข้อมูล), POST (เพิ่มข้อมูล), PUT (แก้ไขข้อมูล) และ DELETE (ลบข้อมูล)
10. Notification: การแจ้งเตือนที่ปรากฏบนหน้าจออุปกรณ์เพื่อบอกให้ผู้ใช้ทราบถึงข้อมูลหรือกิจกรรมบางอย่าง
11. Text-to-Speech (TTS): เทคโนโลยีที่แปลงข้อความตัวอักษรให้เป็นเสียงพูด
12. การรู้จำอักษรด้วยแสง (Optical Character Recognition - OCR): เทคโนโลยีที่ใช้ในการแปลงข้อความจากรูปภาพให้กลายเป็นข้อความที่คอมพิวเตอร์สามารถเข้าใจและแก้ไขได้

บทที่ 3 ผลงานที่เกี่ยวข้อง

3.1 แอปพลิเคชันช่วยอ่านฉลากยาโดยเฉพาะฟังก์ชันที่ช่วยให้อ่านข้อมูลยาได้ง่ายขึ้นอย่างชัดเจน

[12] ผู้วิจัย: ณัฐกรณ์ ศรีบูรณ์ (2567)

ที่มา: ผู้สูงอายุที่มีอายุมากกว่า 60 ปีขึ้นไป มีปัญหาด้านสายตา ความจำ และไม่คุ้นเคยกับเทคโนโลยี ซึ่งส่งผลให้เกิดความยากลำบากในการอ่านฉลากยาและปฏิบัติตามคำแนะนำในการใช้ยาได้อย่างถูกต้อง

วิธีการ: พัฒนาแอปพลิเคชันที่เป็นตัวช่วยในการอ่านฉลากยา โดยใช้เทคโนโลยีการรู้จำอักษรด้วยแสง (OCR) เพื่อแปลงข้อความบนฉลากยาให้เป็นข้อมูลดิจิทัล และนำเสนอข้อมูลในรูปแบบที่เข้าใจง่าย มีตัวอักษรขนาดใหญ่ เพื่อส่งเสริมพฤติกรรมการใช้ยาที่ถูกต้อง

ผลลัพธ์: จากการทดสอบกับกลุ่มอาสาสมัคร พบร่วมกับความสามารถตอบคำถามเกี่ยวกับวิธีการรับประทานยา จำนวนครั้งต่อวัน จำนวนเม็ด และต้องใช้ยา ก่อนหรือหลังอาหารหรือเวลา มีอาการ ได้ถูกต้องทั้งหมดทุกข้อและในทุกคลาส แสดงให้เห็นว่า แอปพลิเคชันสามารถช่วยให้ผู้สูงอายุเข้าใจข้อมูลยาได้ดีขึ้น

ข้อดี:

1. เป็นแอปพลิเคชันที่ตอบโจทย์ผู้สูงอายุได้อย่างตรงจุด
2. ช่วยลดความพิคพาดในการใช้ยาอย่างมีนัยสำคัญ
3. ใช้งานง่าย เหมาะสมกับผู้สูงอายุที่ไม่คุ้นเคยกับเทคโนโลยี

ข้อจำกัด:

1. การสแกนข้อมูลบางครั้งมีความหน่วง (delay) เพื่อให้การอ่านฉลากยาในครั้งก่อนหน้าเสร็จเรียบร้อยก่อนดำเนินการสแกนครั้งต่อไป
2. อาจมีข้อจำกัดในการรองรับฉลากยาที่มีรูปแบบแตกต่างกัน

3.2 แอปพลิเคชันบน LINE LIFF ร่วมกับ Google Apps Script สำหรับติดตามและประเมินผลความสมำเสมอในการรับประทานยาของผู้ป่วยวันโรค โรงพยาบาลชั้นนำทันเรนทร

[13] ผู้วิจัย: วิศวััสดุ์ ประยิษฐ์ (2567)

ที่มา: ปัญหาของผู้ป่วยที่ขาดความสมำเสมอในการทานยา โดยเฉพาะในกลุ่มโรคเรื้อรัง เช่น วันโรค และผู้สูงอายุที่มักจะลืมหรือสับสนเรื่องการทานยา ซึ่งส่งผลกระทบต่อประสิทธิภาพการรักษาและอาจทำให้เกิดภาวะแทรกซ้อนได้

วิธีการ: ใช้แอปพลิเคชันขนาดเล็กบนแพลตฟอร์ม LINE LIFF (LINE Front-end Framework) เพื่อติดตามการทำงานยาของผู้ป่วยในโรงพยาบาล โดยแอปพลิเคชันจะทำหน้าที่ดังนี้

1. ส่งข้อความแจ้งเตือนผู้ป่วยเมื่อถึงเวลาทานยา
2. ให้ผู้ป่วยส่งภาพถ่ายยืนยันการทำงานยาแต่ละครั้ง
3. บันทึกตำแหน่งที่ผู้ใช้ทานยาได้
4. จัดเก็บข้อมูลลงในฐานข้อมูลเพื่อการติดตามผลและวิเคราะห์

ผลลัพธ์: หลังจากผู้ป่วยใช้แอปพลิเคชันเป็นเวลา 4 เดือน พบร่วมกับผลลัพธ์ พบว่าค่าเฉลี่ยความสมำเสมอในการทานยาเพิ่มขึ้นจาก 90.66% เป็น 98.38% อย่างมีนัยสำคัญทางสถิติ ซึ่งแสดงให้เห็นว่าการแจ้งเตือนและการติดตามผลอย่างเป็นระบบช่วยให้ผู้ป่วยมีวินัยในการทานยามากขึ้น และสามารถลดโอกาสการลืมทานยาได้อย่างมีประสิทธิภาพ

ข้อดี:

1. ใช้แพลตฟอร์ม LINE ซึ่งเป็นแอปพลิเคชันที่ผู้ป่วยคุ้นเคยและใช้งานอยู่แล้ว ทำให้เรียนรู้และเข้าถึงได้ง่าย
2. ต้นทุนในการพัฒนาระบบไม่สูงมาก
3. ผลลัพธ์ที่ได้มีความชัดเจนและวัดผลได้
4. สามารถติดตามและวิเคราะห์พฤติกรรมการทำงานยาได้อย่างเป็นระบบ

ข้อเสนอแนะ:

1. เนื่องจากจะมีการยุติการให้บริการ LINE Notify ตั้งแต่วันที่ 31 มีนาคม พ.ศ. 2568 เป็นต้นไป จึงจำเป็นต้องใช้โปรแกรมอื่นมาใช้ทดแทน LINE Notify เช่น Google Chat หรือ Telegram สำหรับการแจ้งเตือนต่อไป
2. ควรมีการพัฒนาระบบให้รองรับการทำงานแบบ standalone หากว่าจะต้องใช้แพลตฟอร์มเฉพาะ

3.3 โครงการวิจัยเรื่อง เทคโนโลยีเภสัชสารสนเทศแสดงข้อมูลฉลากยาเอกสารกำกับยาแบบอัตโนมัติสำหรับ บริหารจัดการคลังยาปฏิชีวนะเพื่อความปลอดภัย

[14] ผู้จัด: วิรุฬห์ ศรีบริรักษ์ (2562)

ที่มา: ผู้บริโภคขาดความรู้ความเข้าใจถึงอันตรายที่เกี่ยวกับยา อิกทั้งยังไม่มีแหล่งข้อมูลที่เข้าถึงได้ง่าย ข้อมูลบนฉลากยาจากคลินิกหรือโรงพยาบาลมักไม่เพียงพอต่อการทำความเข้าใจ นอกจากนี้ยังมีปัญหาในการบริหารจัดการคลังยาที่ต้องอาศัยการบันทึกข้อมูลด้วยมือซึ่งใช้เวลานานและเสี่ยงต่อความผิดพลาด

วิธีการ: พัฒนาระบบที่ใช้เทคโนโลยี Optical Character Recognition (OCR) ในการแปลงไฟล์ภาพเอกสารที่ได้รับการสแกนให้กลายเป็นไฟล์ข้อมูลความต้องการ และจัดเก็บในฐานข้อมูลในเครื่องแม่ข่าย ระบบนี้ทำหน้าที่เป็นเครื่องมือที่ช่วยในการ

1. ลดความคลาดเคลื่อนทางยา (Medication Error)

2. ป้องกันความผิดพลาดในการจัดการคลังยา

3. ให้ข้อมูลยาที่ถูกต้องและครบถ้วนแก่ผู้ใช้

ผลลัพธ์: ระบบที่พัฒนาขึ้นมีความแม่นยำในการแปลงข้อมูลสูงถึงร้อยละ 96.61 และสามารถช่วยลดเวลาของการเก็บข้อมูลลงฐานข้อมูลยาได้อย่างมีนัยสำคัญ เมื่อเทียบกับการบันทึกข้อมูลด้วยมือ

ข้อดี:

1. ช่วยให้ผู้บริโภcm มีความเข้าใจในการใช้ยามากขึ้น

2. ช่วยให้การใช้ยา มีความปลอดภัยและลดปัญหาของการใช้ยาที่ผิดกับโรค

3. ลดเวลาและแรงงานในการบันทึกข้อมูลยา

4. มีความแม่นยำสูงในการอ่านข้อมูล

ข้อจำกัด:

1. ยังคงต้องพัฒนาความยืดหยุ่นของระบบให้รองรับรูปแบบเอกสารที่แตกต่างกัน เช่น ฉลากยาจากโรงพยาบาลต่างๆ ที่อาจมีรูปแบบไม่เหมือนกัน

2. ความแม่นยำอาจลดลงเมื่อฉลากยา มีคุณภาพพำเพ;t หรือข้อมูลไม่ชัดเจน

3. ต้องมีการปรับปรุงและฝึกฝนโมเดล OCR อย่างต่อเนื่องเพื่อรองรับฟอนต์และรูปแบบตัวอักษรที่หลากหลาย

3.4 การเปรียบเทียบแอปพลิเคชันที่มีอยู่ในตลาด

การวิเคราะห์แอปพลิเคชันที่อยู่ในตลาดช่วยให้เห็นภาพรวมของฟีเจอร์และแนวทางการออกแบบที่ได้รับความนิยม ซึ่งสามารถนำมาประยุกต์ใช้ในการพัฒนาแอปพลิเคชันของคุณได้

เราได้ทำการเปรียบเทียบฟีเจอร์หลักของแอปพลิเคชันยอดนิยม 5 ตัว ได้แก่ CapYaDoo, PharmaSee [15], MyYaAndYou [16], Medisafe [17] และ RDU รูรี่องยา [18] ดังตารางต่อไปนี้

3.4.1 ตารางเปรียบเทียบแอปพลิเคชันในตลาด

หมวดฟีเจอร์	CapYaDoo	PharmaSee	MyYaAndYou	Medisafe	RDU รู้เรื่องยา
ค้นหาข้อมูลยา	ค้นหา ด้วย ข้อความ / สแกน / แปลภาษาได้	ค้นหา จาก ภาพ ยา (AI ตรวจรูป)	ค้นหาด้วยชื่อยา	ค้นหา ด้วย ชื่อยา เท่านั้น	ค้นหา จากชื่อยา ในฐานข้อมูล อย.
เสียง อ่าน ข้อมูล ยา (Text-to-Speech)	มีเสียงอ่านข้อมูล ยาให้ผู้ใช้ฟัง	ไม่มี	ไม่มี	ไม่มี	ไม่มี
บันทึกอาการหลังทานยา	มี ระบบ บันทึก อาการ และ ผล ข้างเคียง	ไม่มี	มีบันทึกยา แต่ไม่ บันทึกอาการ	ไม่มี	ไม่มี
แจ้งเตือนการทานยา	ตั้งเวลาเตือนแยก ตามยาได้	ไม่มี	มีระบบแจ้งเตือน	มีระบบแจ้งเตือน ครอบคลุม	ไม่มีระบบเตือน
สแกนบาร์โค้ด / ตัว หนังสือ	สแกน ได้ ทั้ง "บาร์ โค้ด" และ "ข้อความ บน ฉลากยา"	สแกน รูป ยา (AI Image Recognition)	ไม่มี สแกน บาร์ โค้ด	ไม่มีสแกน	ไม่มีสแกน
แปลภาษาข้อมูลยา	แปล ข้อความ ยา ได้ (ไทย ↔ อังกฤษ)	ไม่มี	ไม่มี	ไม่มี	ไม่มี
ตรวจสอบข้อมูลยา / อย.	ตรวจ ผ่าน ฐาน ข้อมูล ยา ใน ประเทศไทย	ขึ้นกับฐานข้อมูล ของ AI	อ้างอิง จาก ฐาน ข้อมูล อย.	ข้อมูล ทั่วไป ไม่ เผพะ อย.	ใช้ฐานข้อมูลจาก อย. โดยตรง
เหมาะสมกับผู้สูงอายุ / ใช้งานง่าย	UI เรียบง่าย / มีเสียง / แปล ภาษา	ต้องถ่ายรูป อาจ ใช้งานยาก	UI เชิงง่าย	ใช้งานง่าย และ ระบบ แจ้ง เตือน ชัดเจน	ข้อความ เยอะ ต้องอ่านเอง
บันทึก ประวัติ ยา ใน เครื่อง	มี ระบบ เก็บ ประวัติ ยา และ อาการย้อนหลัง	เก็บ เนพะ ภาพ ยา	บันทึก ข้อมูล ยา ได้	เก็บข้อมูลยาและ เวลาเตือน	ไม่มีระบบบันทึก
เทคโนโลยี AI / OCR	ใช้ OCR อ่าน ฉลาก ยา และ ตรวจบาร์โค้ด	ใช้ AI แยกภาษา	ไม่มี	ไม่มี	ไม่มี

ตาราง 3.1: เปรียบเทียบแอปพลิเคชันในตลาด

จากการเปรียบเทียบข้างต้น จะเห็นได้ว่า CapYaDoo มีข้อได้เปรียบในหลายด้าน ได้แก่

ข้อได้เปรียบของ CapYaDoo

- ความหลากหลายในการสแกน: รองรับทั้งการสแกนบาร์โค้ดและตัวหนังสือ (OCR)
- การจัดการข้อมูลยา: สามารถเพิ่ม แก้ไข และลบข้อมูลยาที่กำลังใช้อยู่ได้
- เสียงอ่าน: รองรับการอ่านข้อมูลให้ผู้ใช้ฟัง (Text-to-Speech)
- ระบบแจ้งเตือน: มีระบบเตือนทานยาที่สามารถปรับวัน เวลา ที่ต้องการให้แจ้งเตือนได้อย่างละเอียด
- การติดตามอาการ: สามารถบันทึกและติดตามอาการหลังทานยาได้

จุดเด่นที่เหมือนกัน

1. การค้นหาข้อมูลยา: แอปพลิเคชันส่วนใหญ่เน้นการค้นหาด้วยชื่อยาเป็นหลัก
2. การแจ้งเตือนการทานยา: มีแจ้งเตือนการทานยาตามกำหนด
3. การบันทึกประวัติยา: มีระบบบันทึก/เก็บประวัติข้อมูลยาที่เคยใช้งานในเครื่อง

3.4.2 สรุปการเปรียบเทียบ

จากการเปรียบเทียบที่เห็นได้ว่า CapYaDoo มีความครบถ้วนและครอบคลุมที่เจอร์มากที่สุด เมื่อเทียบกับแอปพลิเคชันอื่นๆ ในตลาด ซึ่งแต่ละแอปพลิเคชันนั้นมีฟีเจอร์ที่โดดเด่นแตกต่างกันไป เช่น PharmaSee โดยเด่นด้านการใช้ AI ตรวจจำภาพยา, MyYaAndYou มีจุดเด่นด้านการเข้าถึงข้อมูลยาที่อ้างอิงจากฐานข้อมูลของ อย. พร้อมทั้งมีระบบแจ้งเตือนและบันทึกข้อมูลยา เหมาะสำหรับผู้ใช้ทั่วไปที่ต้องการข้อมูลยาที่เชื่อถือได้, Medisafe มีระบบแจ้งเตือนที่ครบถ้วนและใช้งานง่าย และ RDU รู้เรื่องยา มีฐานข้อมูลตรงจาก อย. โดยตรง อย่างไรก็ตาม CapYaDoo สามารถรวมฟีเจอร์ที่โดดเด่นจากทุกแอปพลิเคชันไว้ในแอปเดียว ไม่ว่าจะเป็นการค้นหาข้อมูลยาแบบหากหลาย, ระบบแจ้งเตือนการทานยา, การบันทึกประวัติยา, เสียงอ่านข้อมูลยา, และการบันทึกอาการหลังทานยา ด้วยเหตุนี้ ผู้ใช้จึงสามารถใช้งานเพียงแอปพลิเคชันเดียว แต่ได้ครอบคลุมทุกความต้องการสำหรับการดูแลเรื่องยา ตั้งแต่การค้นหาข้อมูล การจัดการตารางยา การแจ้งเตือนทานยาตรงเวลา ไปจนถึงการติดตามและบันทึกอาการหลังทานยา ทำให้ผู้ใช้สามารถทานยาได้อย่างถูกต้อง ครบถ้วน และปลอดภัยมากที่สุด อีกทั้งยังช่วยให้สามารถติดตามผลของยาและอาการที่เกิดขึ้นได้อย่างเป็นระบบ

3.5 สรุปและการนำไปประยุกต์ใช้

จากการศึกษางานวิจัยและโครงการที่เกี่ยวข้องทั้ง 3 งาน รวมถึงการเปรียบเทียบแอปพลิเคชันที่มีอยู่ในตลาด สามารถสรุปประเด็นสำคัญที่จะนำมาประยุกต์ใช้ในโครงการนี้ได้ดังนี้

3.5.1 แนวทางการพัฒนาที่ได้จากการวิจัย

1. การออกแบบที่เหมาะสมกับผู้สูงอายุ จากการวิจัยที่ 1 แสดงให้เห็นถึงความสำคัญของการออกแบบ UI/UX ที่เรียบง่าย ใช้ตัวอักษรขนาดใหญ่ และมีความชัดเจน
2. ระบบแจ้งเตือนและติดตามผล จากการวิจัยที่ 2 แสดงให้เห็นว่าการมีระบบแจ้งเตือนและติดตามผลอย่างเป็นระบบสามารถเพิ่มความสนใจในการทานยาได้อย่างมีนัยสำคัญ
3. เทคโนโลยี OCR จากการวิจัยที่ 3 แสดงให้เห็นถึงประสิทธิภาพของ OCR ในการอ่านข้อมูลยา แต่ต้องคำนึงถึงความยืดหยุ่นในการรองรับรูปแบบฉลากที่แตกต่างกัน

3.5.2 ข้อจำกัดที่ต้องหลีกเลี่ยง

1. หลีกเลี่ยงการพึ่งพาแพลตฟอร์มเฉพาะที่อาจมีการเปลี่ยนแปลงหรือยุติบริการ

2. ต้องพัฒนาระบบ OCR ให้มีความแม่นยำและรองรับรูปแบบคลาสิกที่หลากหลาย
3. ต้องแก้ไขปัญหา delay ในการประมวลผลให้มีประสิทธิภาพมากขึ้น

3.5.3 คุณสมบัติที่จะนำมาพัฒนาในโครงการ

โครงการนี้จะนำแนวคิดและข้อดีจากการวิจัยที่เกี่ยวข้องมาประยุกต์ใช้ โดยมีจุดเด่นดังนี้

1. ใช้เทคโนโลยี OCR ใน การอ่านคลาสิกพร้อมกับฐานข้อมูลยาจาก Thai Medicine Terminology (TMT) ที่มีความน่าเชื่อถือ
2. พัฒนาระบบแจ้งเตือนที่หลากหลาย เช่น เสียง, การสั่น เพื่อรองรับผู้สูงอายุที่มีข้อจำกัดทางด้านการได้ยินและการมองเห็น
3. สร้างระบบบันทึกและติดตามประวัติการทานยาที่ครบถ้วน
4. ออกแบบ UI/UX ที่เหมาะสมกับผู้สูงอายุโดยเฉพาะ

3.6 ประสบการณ์การพัฒนาที่ผ่านมา

เพื่อให้เห็นถึงประสบการณ์และความสามารถของทีมผู้พัฒนาในการสร้างแอปพลิเคชัน จึงได้นำเสนอโครงการที่ผ่านมาที่เกี่ยวข้องกับการพัฒนาแอปพลิเคชันด้วย Flutter Framework

3.6.1 Application Chat Music Player

วัตถุประสงค์:

1. เพื่อให้ผู้ใช้งานเกิดความผ่อนคลายและสร้างความสุขระหว่างการใช้งานให้กับผู้ใช้ได้
2. เพื่อให้ผู้ใช้งานได้พูดคุยและแลกเปลี่ยนความคิดเห็นกับผู้ใช้คนอื่น ๆ ได้

บทบาทและสิ่งที่ทำ:

1. นาย ฤทธิชล พลราช: มีหน้าที่ในการออกแบบ UX/UI และทำการพัฒนาแอปพลิเคชัน
2. นางสาว จิรัชญา ราชพลแสน: มีหน้าที่ในการจัดทำรายงานและพัฒนาแอปพลิเคชัน

เทคโนโลยีที่ใช้:

1. ใช้ภาษา Dart และ Flutter framework
2. Backend ใช้ Firebase Firestore, Firebase Authentication
3. Package ที่ใช้คือ audioplayer

3.6.2 Application Location Discovery

วัตถุประสงค์หลัก:

- เพื่อให้ผู้ใช้งานเลือก/ไม่เลือก สถานที่พักผ่อน ตามหนังสือภายในมหาวิทยาลัย
- เพื่อให้ผู้ใช้งานใช้ข้อมูลในการตัดสินใจเลือก เข้า สถานที่ท่องเที่ยวใกล้ๆ เรา สิ่งอำนวยความสะดวกในสถานที่นั้น

บทบาทและสิ่งที่ทำ:

- นางสาว ปรารณา สุภารวงศ์: ทำหน้าที่ในการออกแบบแอปพลิเคชัน Logo, UI และพัฒนาแอปพลิเคชัน

เทคโนโลยีที่ใช้:

- ใช้ภาษา Dart และ Flutter framework
- Backend ใช้ Firebase Firestore, Firebase Authentication

3.6.3 บทเรียนที่ได้จากการพัฒนาแอปพลิเคชัน

จากการพัฒนาแอปพลิเคชันทั้งสองโครงการ ทีมผู้พัฒนาได้เรียนรู้และสะสมประสบการณ์ในด้านต่างๆ ดังนี้

ด้านการออกแบบ UI/UX

- การออกแบบส่วนติดต่อผู้ใช้ที่ใช้งานง่ายและสวยงาม
- การเลือกใช้สีและฟอนต์ที่เหมาะสมกับกลุ่มผู้ใช้
- การจัดวางองค์ประกอบบนหน้าจอให้เป็นระเบียบและเข้าถึงได้ง่าย

ด้านการพัฒนาแอปพลิเคชัน

- การใช้ Flutter Framework ใน การพัฒนาแอปพลิเคชันข้ามแพลตฟอร์ม
- การจัดการ State Management ในแอปพลิเคชัน
- การใช้งาน Package ต่างๆ เพื่อเพิ่มฟังก์ชันให้กับแอปพลิเคชัน

ด้านการจัดการฐานข้อมูล

- การใช้ Firebase Firestore ในการจัดเก็บข้อมูล
- การจัดการ Authentication และ User Management
- การออกแบบโครงสร้างข้อมูลให้เหมาะสมกับการใช้งาน

ด้านการทำงานเป็นทีม

1. การแบ่งหน้าที่และความรับผิดชอบในทีม
2. การสื่อสารและประสานงานระหว่างสมาชิกในทีม
3. การจัดการเวลาและการกำหนดการในการพัฒนา

ประสบการณ์เหล่านี้จะเป็นประโยชน์อย่างมากในการพัฒนาโครงการ CapYaDoo เนื่องจากทีมผู้พัฒนาได้มีความคุ้นเคยกับ Flutter Framework และการทำงานร่วมกันมาแล้ว

บทที่ 4

ระเบียบวิธีดำเนินโครงการ

4.1 การวิเคราะห์และออกแบบระบบ

4.1.1 Persona

[19] นายถึง ตัวตนสมมติที่เป็นตัวแทนกลุ่มลูกค้าเป้าหมายของธุรกิจ โดยอิงจากข้อมูลการวิจัยและวิเคราะห์พฤติกรรมความต้องการ และแรงจูงใจของกลุ่มลูกค้าจริง เพื่อให้ธุรกิจเข้าใจลูกค้าได้อย่างลึกซึ้ง

ความสำคัญของ Persona

1. ช่วยให้ธุรกิจเข้าใจลูกค้าได้อย่างลึกซึ้ง
2. ปรับกลยุทธ์ทางการตลาดให้เหมาะสมกับพฤติกรรมของกลุ่มเป้าหมาย
3. สื่อสารได้อย่างตรงใจลูกค้า

 <p>Key Attribute ต้องการการดูแลและความปลอดภัย: ให้ความสำคัญกับการปิดบานช่วย จัดการสุขภาพและยาให้เป็นไปอย่างปลอดภัย</p>	<p>คุณย่ามาลี (อายุ 75 ปี) อาชีพ: แม่บ้าน ผู้สูงอายุที่ต้องการควบคุมช่วย เหลือ</p> <p>Short description ผู้สูงอายุที่บีบปูนห้าด้านสุขภาพและการมองเห็น ไม่เคยการใช้ เทคโนโลยี กำลังต้องพึ่งพาลูกหลานในการดูแลและการเดินทางและ</p>
<p>Needs</p> <ul style="list-style-type: none"> ความบัน្តื่องว่าจะมีคนดูแลช่วยจัดการยาให้ถูกต้องและปลอดภัย วิธีการรับรู้ข้อมูลยาที่ง่ายและชัดเจน เช่น การพิมพ์เสียง การแจ้งเตือนที่ชัดเจนให้ในสิ่งของยา 	<p>Opportunities</p> <ul style="list-style-type: none"> พัฒนาฟังก์ชันแปลงข้อความเป็นเสียง (Text-to-Speech) สำหรับ ข้อมูลยา ใช้การแจ้งเตือนด้วยเสียงที่ดังและชัดเจน

รูป 4.1: ตัวตนสมมติที่เป็นตัวแทนของกลุ่มผู้ใช้เป้าหมายคนที่ 1

 <p>ชื่อ: คุณหมอนชัย (อายุ 58 ปี) อาชพ: แพทย์เกณฑ์ ผู้สูงอายุที่นับความเป็นอยู่ที่ดี และสุขภาพ</p>	<p>Short description ผู้สูงอายุที่เป็นแพทย์เกณฑ์ มีความรู้เรื่องสุขภาพเป็นอย่างดี และยังคงต้องการเครื่องมือที่ช่วยให้การติดตามสุขภาพเป็นเรื่องง่าย และครบวงจร</p>
<p>Key Attribute ต้องการความรู้และการบูรณาการ: ให้ความสำคัญกับความถูกต้องของข้อมูลและการจัดการสุขภาพแบบครบวงจร</p>	<p>Challenges</p> <ul style="list-style-type: none"> ข้อมูลในแอปพลิเคชันที่ไม่เข้ากันได้กัน แอปพลิเคชันที่ซับซ้อนและมีข้อมูลที่ไม่จำเป็นมากเกินไป ความต้องการว่าข้อมูลสุขภาพล่วงหน้าไม่ปลอดภัย
<p>Needs</p> <ul style="list-style-type: none"> เครื่องมือที่ช่วยจัดระเบียบข้อมูลสุขภาพและยาได้อย่างแม่นยำ ฟังก์ชันที่สามารถบันทึกข้อมูลสุขภาพอื่น ๆ ได้ เช่น การออกกำลังกาย หรืออาหาร ข้อมูลยาที่ละเอียดและบันทึกจากแหล่งอื่นที่เชื่อมต่อได้ 	<p>Opportunities</p> <ul style="list-style-type: none"> แอปพลิเคชันที่บันทึกข้อมูลสุขภาพได้ จัดทำฐานข้อมูลยาที่ละเอียดพร้อมแหล่งอ้างอิง

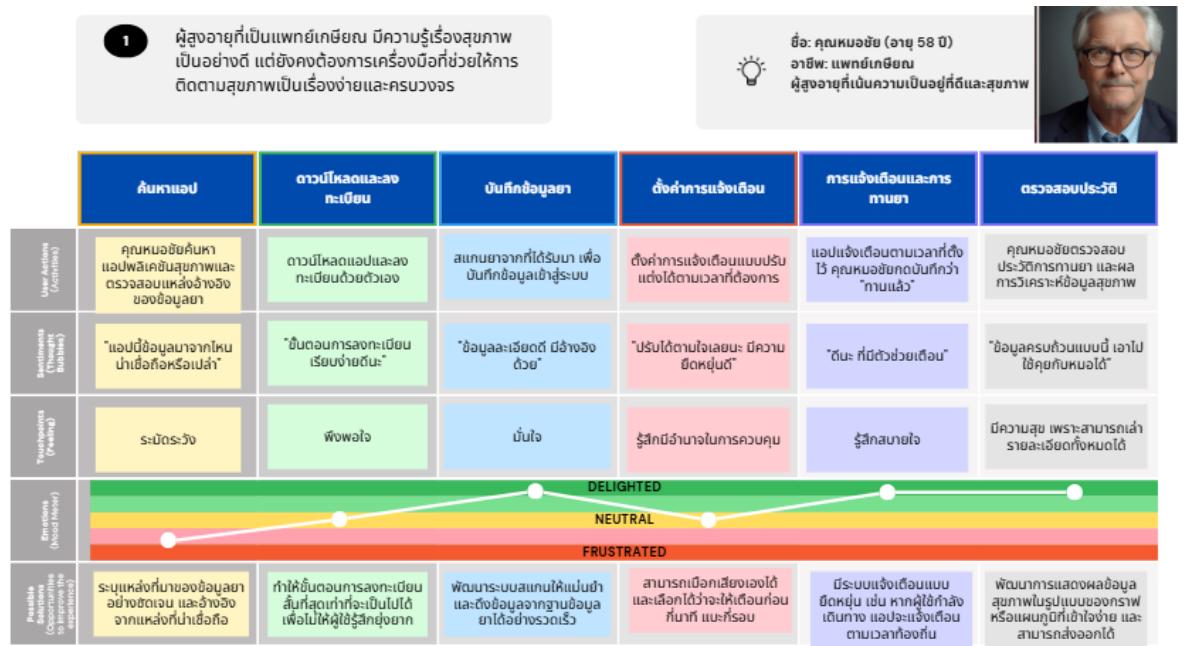
รูป 4.2: ตัวตนสมมติที่เป็นตัวแทนของกลุ่มผู้ใช้เป้าหมายคนที่ 2

 <p>ชื่อ: คุณนารี (อายุ 58 ปี) อาชพ: พนักงานบริษัท ผู้ป่วยเรื้อรัง (Chronic Patient)</p>	<p>Short description พนักงานบริษัทที่ป่วยเป็นโรคความดันโลหิตสูงและเบาหวาน ต้องการยาหลายชนิดอย่างเป็นระบบเพื่อให้สับสนและลืมหายา</p>
<p>Key Attribute ต้องการความแม่นยำและเป็นระบบ: ต้องการเครื่องมือที่ช่วยจัดการซึ่งกันและกัน</p>	<p>Challenges</p> <ul style="list-style-type: none"> ลืมหายาบ่อยครั้ง เพราะต้องการยาหลายเวลาในแต่ละวัน สับสนเรื่องขบวนของยาและเวลาที่ต้องการ ไม่แน่ใจว่ายาตัวไหนมีผลข้างเคียงอย่างไร
<p>Needs</p> <ul style="list-style-type: none"> เครื่องมือที่ช่วยจัดการยาที่ต้องการยาหลายชนิดในแต่ละวันอย่างเป็นระบบ ข้อมูลที่ถูกต้องเที่ยวกันยาที่กำลังจะกิน เช่น ผลข้างเคียง หรือคำแนะนำพิเศษ ระบบแจ้งเตือนที่ช่วยให้ไม่ลืมหายา 	<p>Opportunities</p> <ul style="list-style-type: none"> พัฒนาระบบแจ้งเตือนที่ใช้งานง่ายและปรับแต่งได้ จัดทำฐานข้อมูลยาที่ให้ข้อมูลครบถ้วนและเข้าใจง่าย ออกแบบหน้าจอให้แสดงตารางการกินยาแต่ละวันอย่างชัดเจน

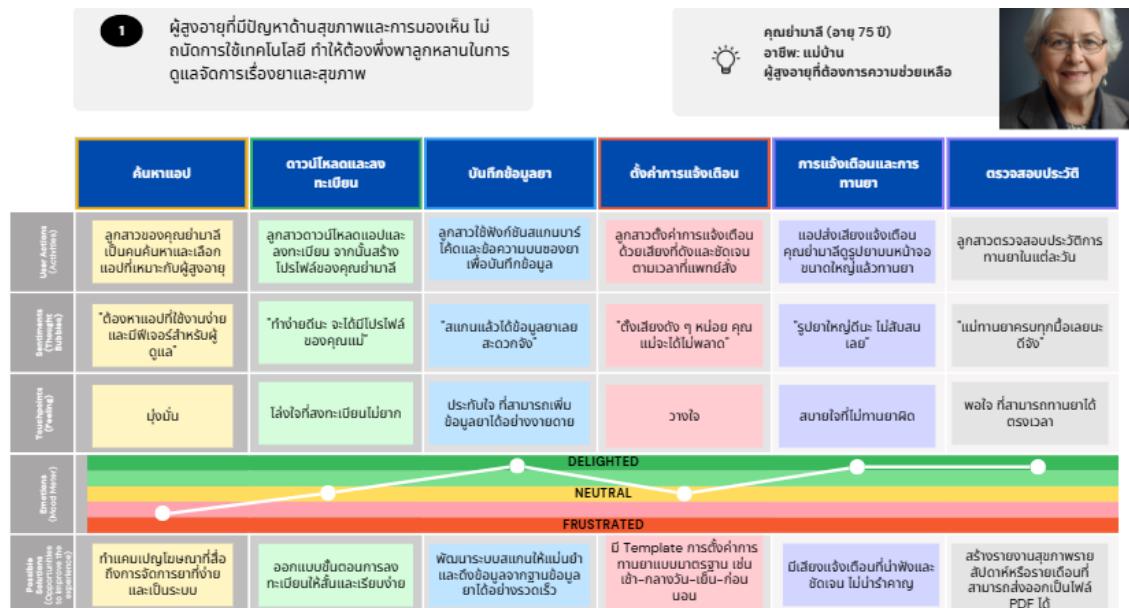
รูป 4.3: ตัวตนสมมติที่เป็นตัวแทนของกลุ่มผู้ใช้เป้าหมายคนที่ 3

4.1.2 User Journey Map

[20] หรือ แผนภาพเส้นทางผู้ใช้ เป็นเครื่องมือที่ใช้อธิบายขั้นตอนและประสบการณ์ของผู้ใช้เมื่อมีปฏิสัมพันธ์กับแอปพลิเคชันของเรา ว่าผู้ใช้ต้องทำอย่างไร พบรัญหาตรงไหน และจะปรับปรุงจุดใดได้บ้าง เพื่อให้ผู้ใช้ได้รับประสบการณ์ที่ดีขึ้น ซึ่งจะแสดงภาพรวมตั้งแต่จุดเริ่มต้นที่ผู้ใช้เริ่มสนใจไปจนถึงการบรรลุเป้าหมายโดยแบ่งเป็นช่วงเวลาและเหตุการณ์สำคัญต่าง ๆ



รูป 4.4: ประสบการณ์ของผู้ใช้เมื่อมีปฏิสัมพันธ์กับแอปพลิเคชันของเราคนที่ 1



รูป 4.5: ประสบการณ์ของผู้ใช้เมื่อมีปฏิสัมพันธ์กับแอปพลิเคชันของเราคนที่ 2



รูป 4.6: ประสบการณ์ของผู้ใช้เมื่อมีปฏิสัมพันธ์กับแอปพลิเคชันของเรางานที่ 3

4.1.3 ปัญหาและแนวทางการแก้ไข

ผู้สูงอายุมักลืมทานยาหรือสับสนเรื่องชนิดและเวลาทานยา ปัญหานี้แก้ได้ด้วย:

1. ระบบแจ้งเตือนที่ตั้งเวลาได้ตามแพทย์กำหนด
2. ฐานข้อมูลยาที่ครบถ้วน เชื่อถือได้ และเข้าใจง่าย
3. UI ที่ชัดเจนและใช้งานง่ายสำหรับผู้สูงอายุ
4. พิมพ์ชั้น Text-to-Speech และการแจ้งเตือนด้วยเสียง/สั่น/แสง
5. ความปลอดภัยของข้อมูลสุขภาพและการเชื่อมต่อกับอุปกรณ์สุขภาพอื่น ๆ

4.1.4 การวิเคราะห์ความต้องการ (Requirement Analysis)

ความต้องการของผู้ใช้ (User Requirements)

1. ระบบจัดการข้อมูลยา (เพิ่ม/แก้ไข/ลบ)
2. ระบบแจ้งเตือนการทานยาที่ตั้งเวลาได้
3. ระบบการสแกนข้อมูลยาจากฉลาก (Barcode & OCR)
4. ระบบบันทึกประวัติการทานยาและการ
5. ระบบคำนวณความสะอาดสำหรับผู้สูงอายุ (Text-to-Speech)

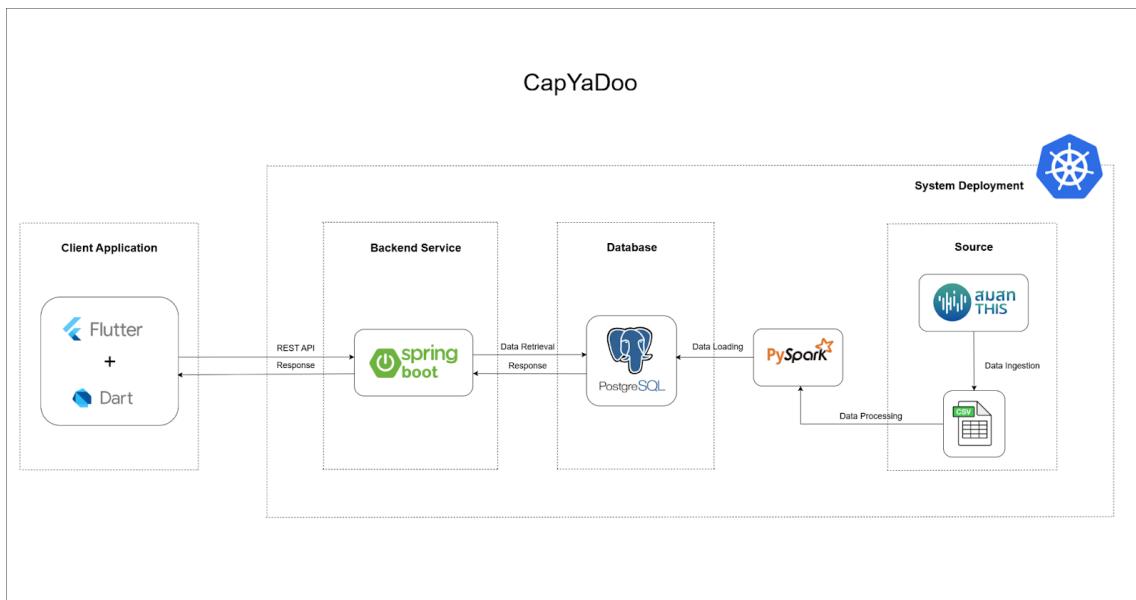
ความต้องการของระบบ (System Requirements)

1. สามารถทำงานได้บน Android
2. มีความเร็วและความแม่นยำในการสแกนข้อมูล
3. มีการจัดเก็บข้อมูลผู้ใช้และข้อมูลยาอย่างปลอดภัย

4.1.5 การออกแบบสถาปัตยกรรม (System Architecture Design)

1. Source: ข้อมูลต้นทางมาจาก สำนักพัฒนามาตรฐานระบบข้อมูลสุขภาพไทย ซึ่งเผยแพร่ข้อมูลในรูปแบบไฟล์ CSV ข้อมูลดังกล่าวจะถูกนำมาใช้เป็นแหล่งข้อมูลหลักของระบบ
2. Data Processing: ข้อมูลจากไฟล์ CSV จะถูกนำเข้าสู่กระบวนการประมวลผลด้วย PySpark ซึ่งเป็นเครื่องมือที่เหมาะสมสำหรับการจัดการและประมวลผลข้อมูลขนาดใหญ่ (Big Data Processing) เพื่อจัดรูปแบบข้อมูลให้อยู่ในโครงสร้างที่เหมาะสมต่อการจัดเก็บและการนำไปใช้งานต่อไป
3. Database: หลังจากการประมวลผลแล้ว ข้อมูลจะถูกโหลดเข้าสู่ฐานข้อมูล PostgreSQL เพื่อจัดเก็บอย่างเป็นระบบและสามารถเข้าถึงได้อย่างมีประสิทธิภาพ
4. Backend Service: ใช้ Spring Boot Framework ในการพัฒนาเป็นบริการฝั่งเซิร์ฟเวอร์ (Backend Service) ทำหน้าที่เป็นตัวกลางระหว่างฐานข้อมูลและผู้ใช้งาน โดยจัดการคำขอ (Request) จากฝั่ง Client รวมถึงประมวลผลและส่งข้อมูลกลับไปยังผู้ใช้งานในรูปแบบ API
5. Client Application: ส่วนของผู้ใช้งาน (Frontend) ถูกพัฒนาด้วย Dart และ Flutter ซึ่งเป็นเครื่องมือสำหรับพัฒนาแอปพลิเคชันแบบ Cross-platform เพื่อให้สามารถใช้งานได้ทั้งบนระบบปฏิบัติการ Android, iOS และ Web
6. System Deployment (kind Cluster): สำหรับการทดสอบและจำลองสภาพแวดล้อมการทำงานของระบบบน Kubernetes ได้มีการใช้เครื่องมือ kind (Kubernetes IN Docker) เพื่อสร้าง Kubernetes cluster จำลองบนเครื่องพัฒนา (Local Environment) โดย cluster ที่สร้างขึ้นจะประกอบด้วย container หลัก ได้แก่ Spring Boot Application Container และ PostgreSQL Database Container

การใช้ kind ช่วยให้สามารถจำลองขั้นตอนการ deploy ระบบจริง เช่น การสร้าง Pod, Service, และการ expose port ให้สามารถเข้าถึงได้จากภายนอก เมื่อการใช้งานบนสภาพแวดล้อม Cloud จริง ทั้งยังช่วยลดค่าใช้จ่ายและเวลาในการเตรียมระบบทดสอบอีกด้วย



รูป 4.7: System Architecture Design

4.1.6 การออกแบบฐานข้อมูล (Database Design)

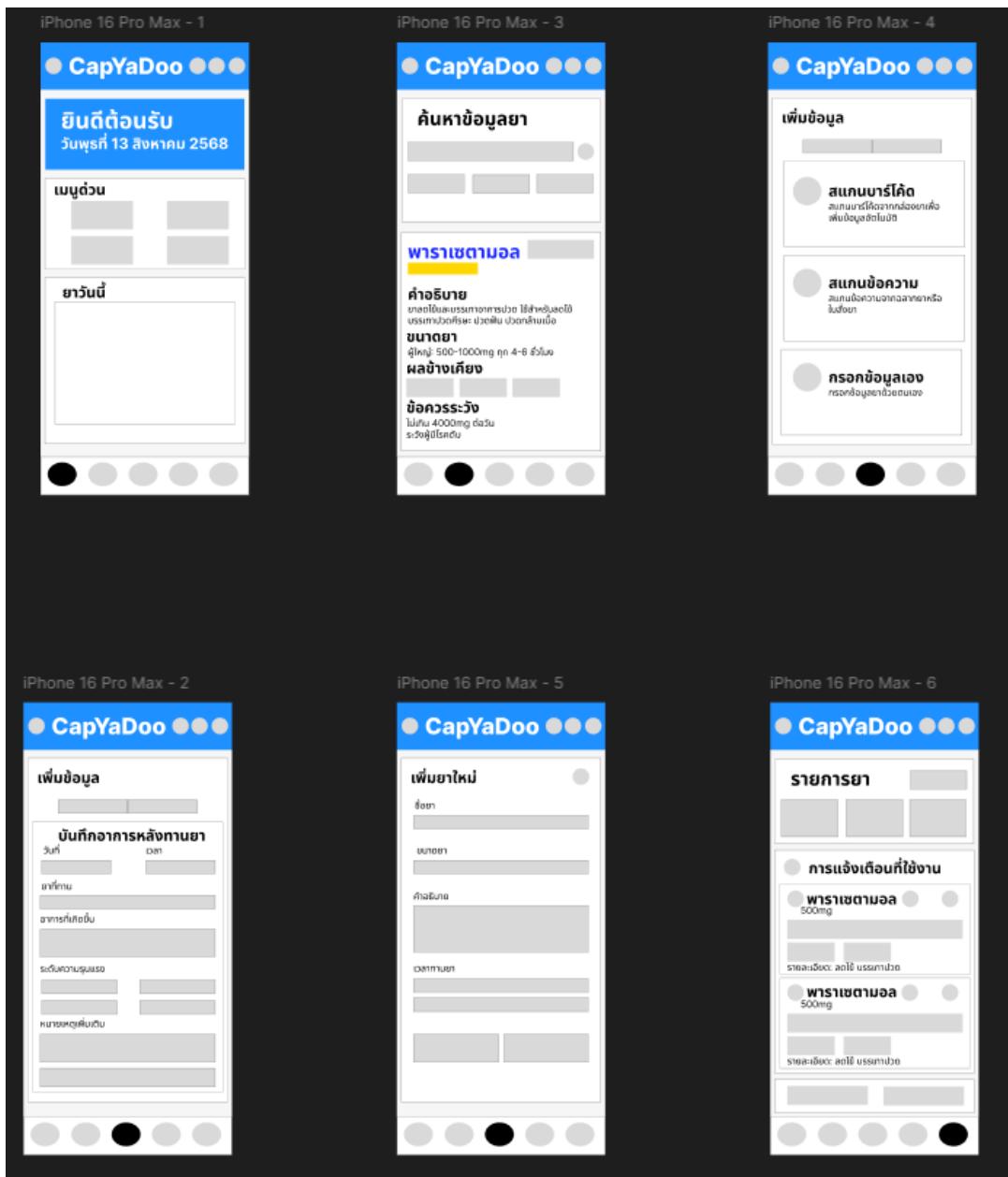
ฐานข้อมูลหลักออกแบบ PostgreSQL ซึ่งเป็นฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ที่มีความน่าเชื่อถือและเหมาะสมกับการจัดการข้อมูลที่มีโครงสร้างชัดเจน โครงสร้างหลัก (Tables) ที่ออกแบบไว้ประกอบด้วย:

1. users: จัดเก็บข้อมูลของผู้ใช้
2. master_drugs: ฐานข้อมูลยาหลัก
3. medications: จัดเก็บข้อมูลยาที่ผู้ใช้ทำการบันทึกไว้
4. schedules: จัดเก็บตารางเวลาการแจ้งเตือนการทานยา
5. histories: จัดเก็บประวัติการทานยาและอาการของผู้ใช้แต่ละคน

โดยแต่ละตารางจะมีการกำหนด Primary Key และ Foreign Key เพื่อเชื่อมโยงข้อมูลเข้าด้วยกันอย่างเป็นระบบ และป้องกันข้อมูลชำรุด

4.1.7 ขั้นตอนการออกแบบระบบ

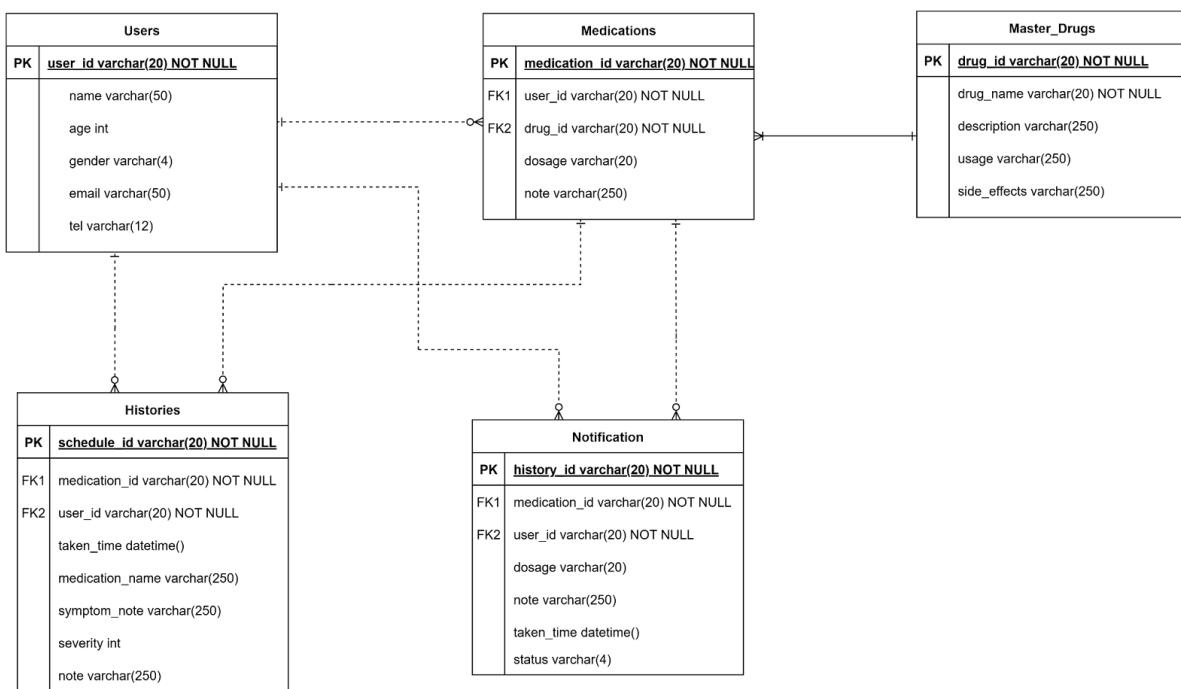
1. การพัฒนา UX/UI: ออกแบบและสร้างหน้าจอการใช้งานตามหลัก Human-Computer Interaction (HCI) ที่เหมาะสมกับผู้สูงอายุ เช่น ใช้ขนาดตัวอักษรที่ใหญ่ สีสันที่ตัดกันชัดเจน และมีปุ่มกดที่ชัดเจน โดยใช้ [21] Figma เป็นเครื่องมือออกแบบ UI/UX แบบออนไลน์ (Cloud-based)



รูป 4.8: ออกแบบ UX/UI ของ Application

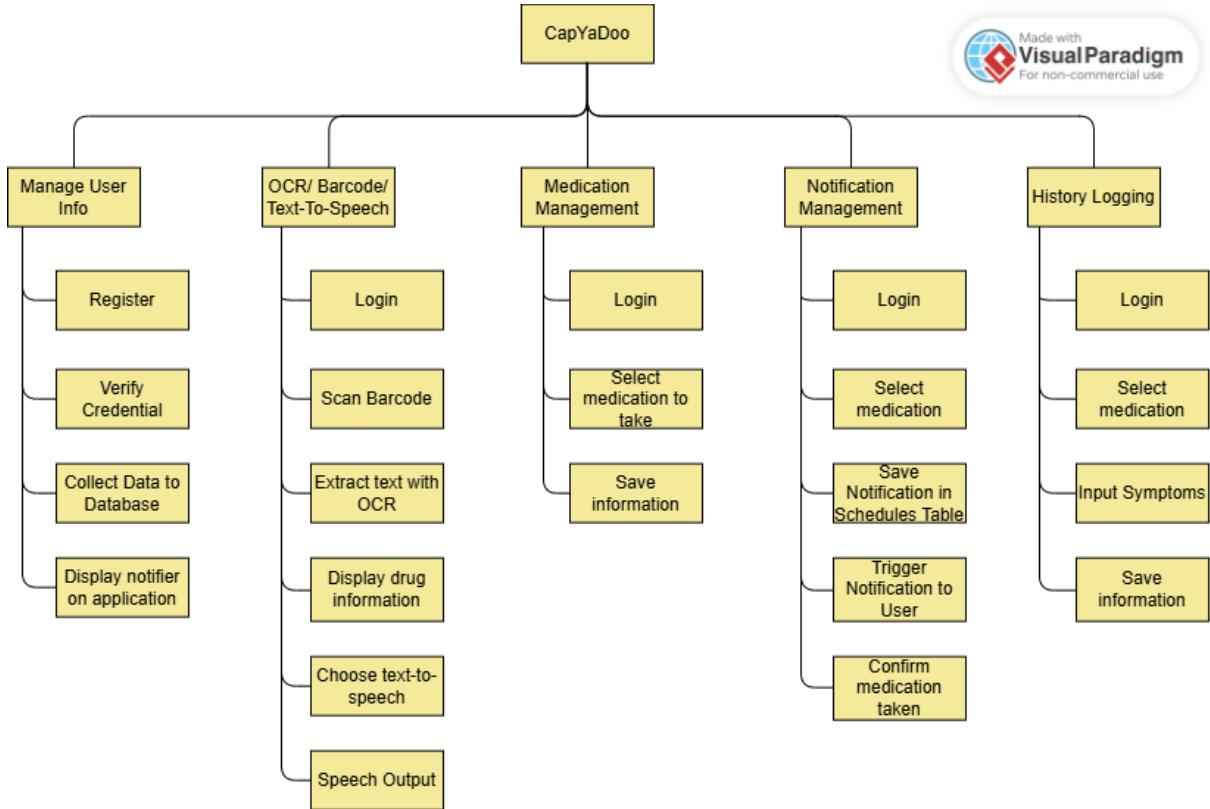
2. Entity Relationship Diagram (ERD) [22] : อธิบายโครงสร้างระหว่างฐานข้อมูลในรูปแบบแผนภาพ โดยในโครงงานนี้ จะใช้ในแนวคิด ERD แบบ Crow's Foot Notation ซึ่งจะประกอบไปด้วย

- (a) Entity: สิ่งของ บุคคล เทศกรณ์ หรือแนวคิดที่สามารถระบุได้ชัดเจน
- (b) Relationship: ความสัมพันธ์ระหว่างเอนทิตี้
- (c) Attributes: คุณลักษณะของเอนทิตี้หรือความสัมพันธ์
- (d) Primary Key: ใช้เพื่อระบุเอนทิตี้อย่างไม่ซ้ำกัน
- (e) Cardinality: จำนวนขั้นต่ำและสูงสุดของความสัมพันธ์ระหว่างเอนทิตี้ เช่น คุณครู 1 คน มีสอนได้หลายวิชา



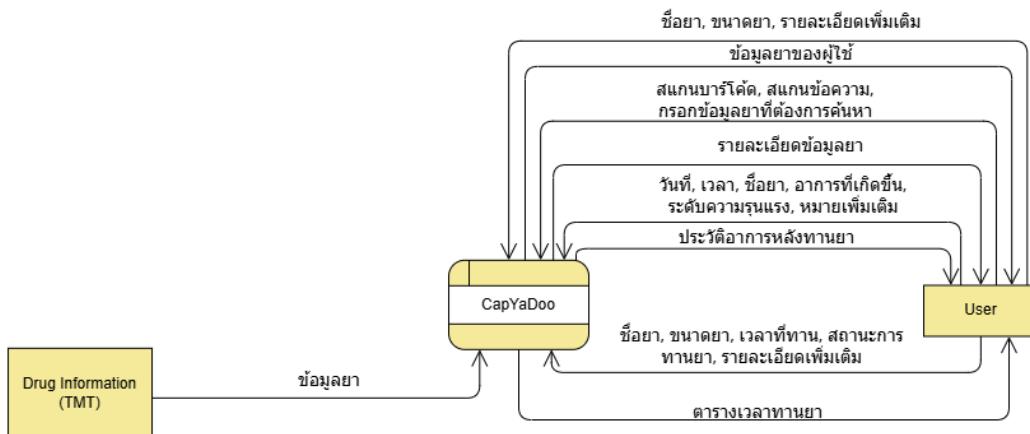
รูป 4.9: Entity Relationship Diagram (ERD)

3. Functional Decomposition: ขั้นตอนการแบ่งระบบออกเป็นส่วนย่อย ๆ เพื่อให้ง่ายต่อการทำความเข้าใจ การวิเคราะห์ การพัฒนา และการนำไปใช้งาน



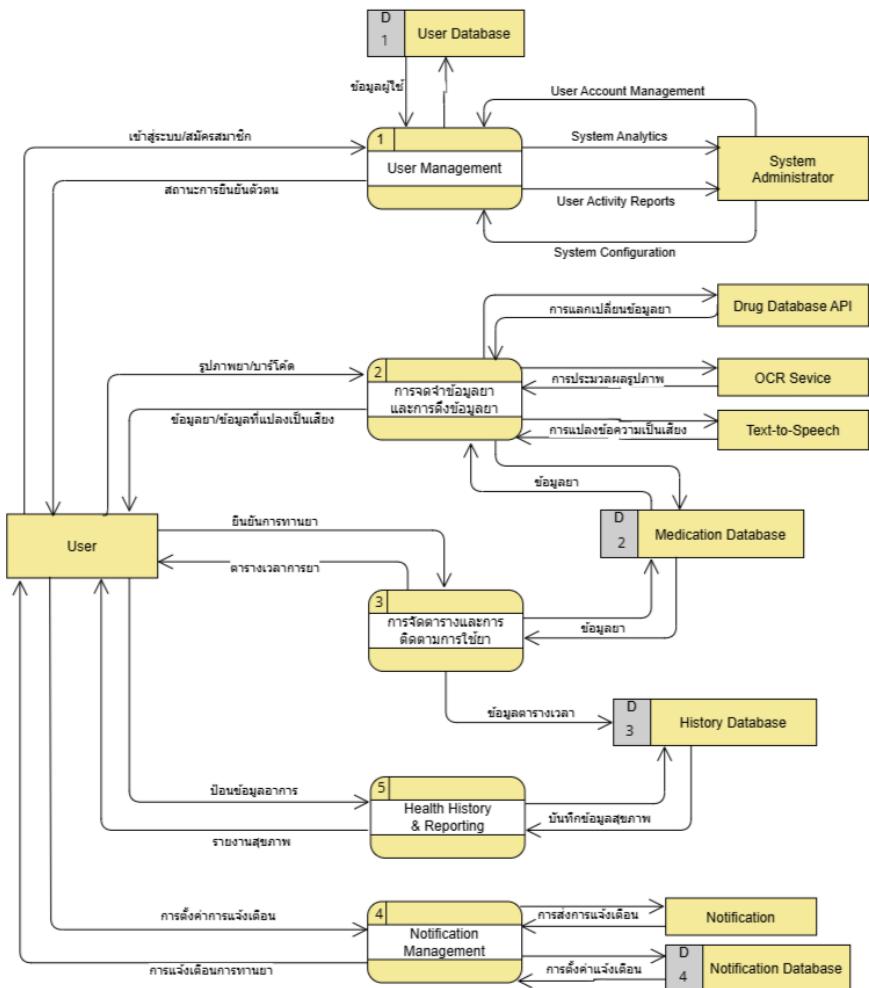
รูป 4.10: Functional Decomposition

4. Data Flow Diagram Level 0: หรือ Context Diagram ใช้แสดงภาพรวมของระบบทั้งหมด



รูป 4.11: Data Flow Diagram Level 0

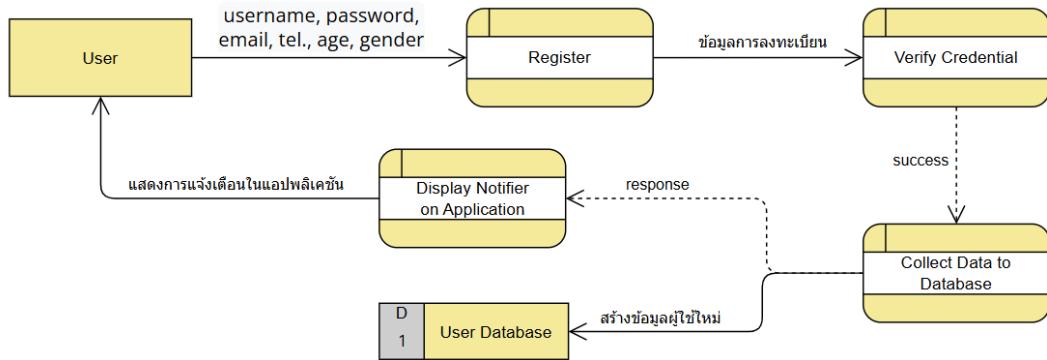
5. Data Flow Diagram Level 1 [23] : ใช้แสดงกระบวนการย่อย ๆ ภายในระบบ



รูป 4.12: Data Flow Diagram Level 1

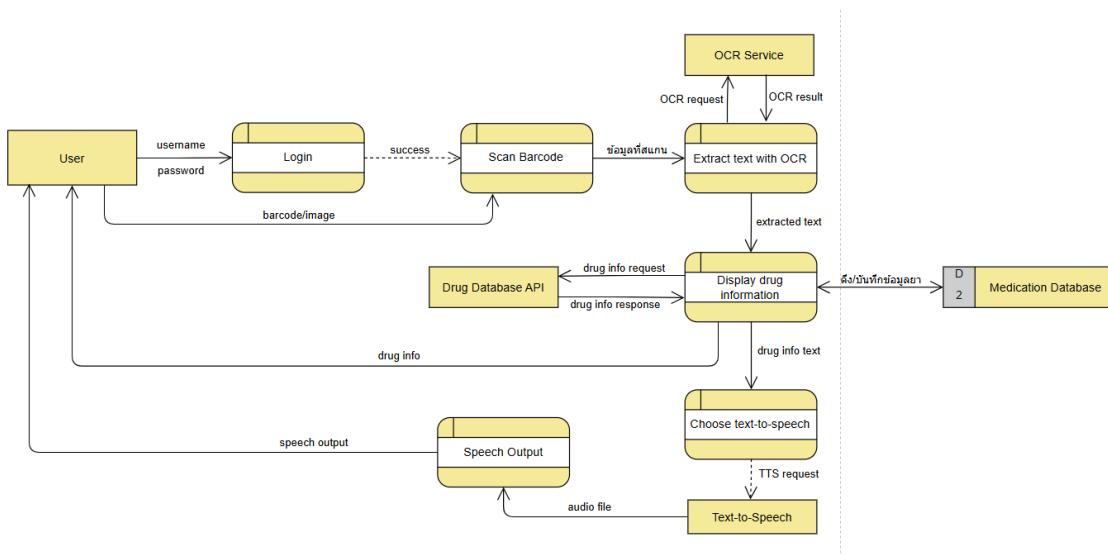
6. Data Flow Diagram Level 2 [23] : ใช้ในการขยายข้อมูลจาก Data Flow Diagram ระดับ 1 ให้ลักษณะมากยิ่งขึ้น

(a) Manage User Information



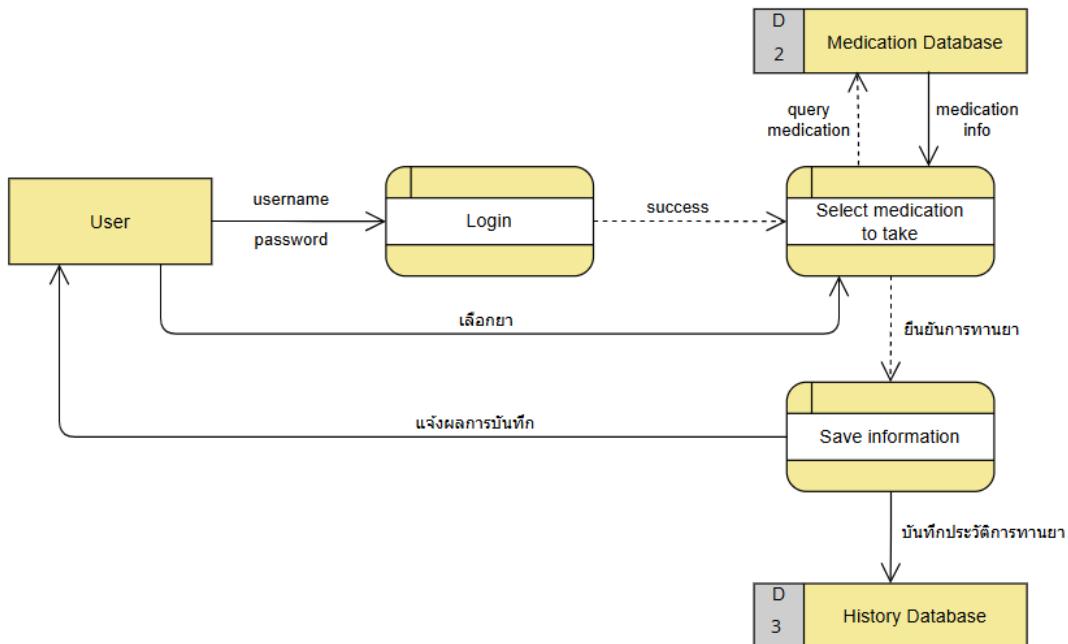
รูป 4.13: Data Flow Diagram Level 2.1 - Manage User Information

(b) OCR/ Barcode/ Text-To-Speech



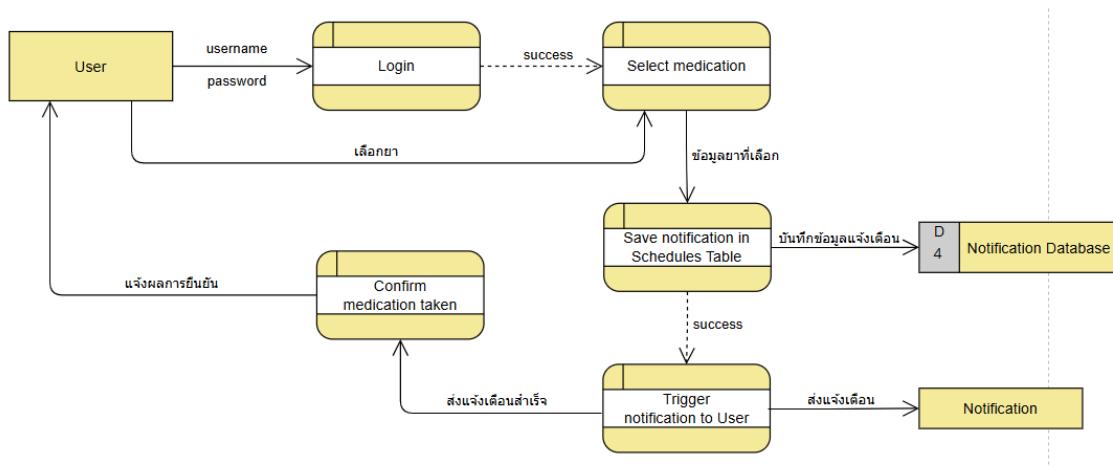
รูป 4.14: Data Flow Diagram Level 2.2 - OCR/ Barcode/ Text-To-Speech

(c) Medication Management



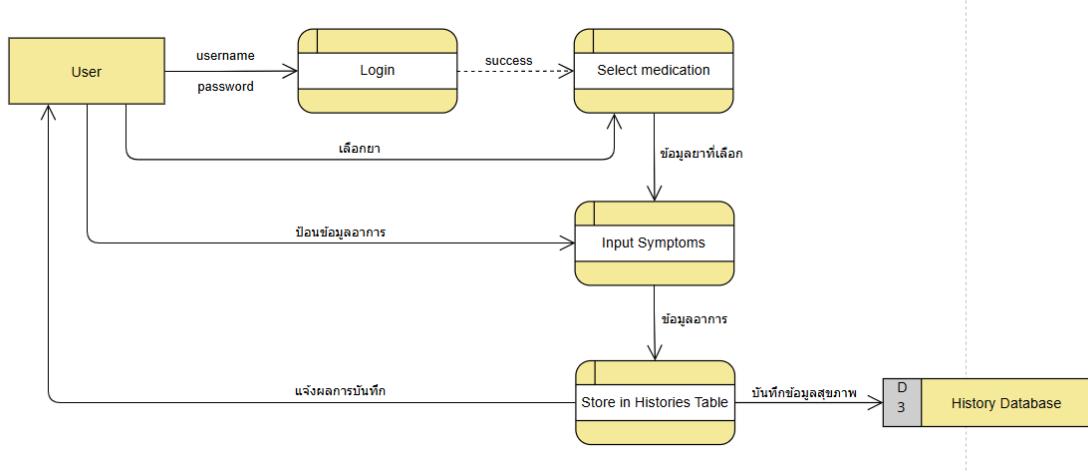
รูป 4.15: Data Flow Diagram Level 2.3 - Medication Management

(d) Notification Management



รูป 4.16: Data Flow Diagram Level 2.4 - Notification Management

(e) History Logging



รูป 4.17: Data Flow Diagram Level 2.5 - History Logging

4.2 การพัฒนาและติดตั้ง (Development and Implementation)

ขั้นตอนนี้จะเปลี่ยนการออกแบบให้กลายเป็นแอปพลิเคชันที่สามารถใช้งานได้จริง

4.2.1 เครื่องมือและสภาพแวดล้อมในการพัฒนา

1. ภาษาโปรแกรม: Dart (สำหรับ Frontend), Java (สำหรับ Backend)
2. Framework: Flutter, Spring Boot
3. IDE (Integrated Development Environment): Android Studio และ Visual Studio Code สำหรับ Flutter, IntelliJ IDEA สำหรับ Spring Boot
4. Kubernetes in Docker (kind): จำลองบนเครื่องพัฒนา (Local Environment)
5. ฐานข้อมูล: PostgreSQL
6. Plugins/Libraries ที่ใช้:
 - flutter_local_notifications: สำหรับระบบแจ้งเตือน
 - google_mlkit_text_recognition: สำหรับฟังก์ชัน OCR
 - mobile_scanner: สำหรับการสแกน Barcode
 - flutter_tts: สำหรับฟังก์ชัน Text-to-Speech

4.2.2 ขั้นตอนการพัฒนา

การพัฒนาในส่วนของหน้าบ้าน (Frontend Development)

พัฒนาในส่วนของ แอปพลิเคชันบนระบบ Android โดยใช้ Flutter (ภาษา Dart) ซึ่งเป็น Framework ที่ช่วยให้การพัฒนา Mobile Application ภายในแอปพลิเคชันมีฟังก์ชันหลักดังนี้

1. สแกนฉลากยา (OCR) ด้วยไลบรารี flutter_tesseract_cr

1. สแกนبارك์โค้ด ด้วย mobile_scanner

1. Text-to-Speech (flutter_tts) เพื่ออ่านข้อความยาให้ผู้ใช้งานฟัง

2. flutter_local_notifications สำหรับตั้งเวลาและแจ้งเตือนการรับประทานยา

ระบบฝั่ง Frontend จะทำการเชื่อมต่อ กับ API ของ Backend ผ่าน HTTP Request เพื่อดึงข้อมูลยา เพิ่มข้อมูลการใช้งาน และบันทึกอาการหลังรับประทานยา

การพัฒนาในส่วนของหลังบ้าน (Backend Development)

ส่วนของหลังบ้านพัฒนาด้วย Spring Boot Framework ซึ่งทำหน้าที่เป็นตัวกลางในการเชื่อมต่อระหว่างแอปพลิเคชันกับฐานข้อมูล PostgreSQL โดยมีการออกแบบ RESTful API สำหรับให้บริการข้อมูล เช่น

1. API สำหรับค้นหาข้อมูลยา

2. API สำหรับเพิ่ม ลบ และแก้ไขข้อมูลยา

3. API สำหรับเพิ่ม ลบ และแก้ไขอาการหลังการรับประทานยา

4. API สำหรับจัดการข้อมูลผู้ใช้งาน

การสร้างและประมวลผลข้อมูลยา (Data Pipeline Development)

ได้ทำการสร้าง Data Pipeline เพื่อจัดการข้อมูลยา โดยใช้เครื่องมือ PySpark สำหรับประมวลผลข้อมูลขนาดใหญ่จากแหล่งข้อมูลภายนอก

ข้อมูลยาหลักที่ใช้คือจาก Thai Medicine Terminology (TMT) ซึ่งให้ข้อมูลในรูปแบบไฟล์ .xlsx โดยมีการดำเนินการดังนี้

1. นำเข้าข้อมูล (Data Ingestion) จากไฟล์ TMT

2. ทำความสะอาดข้อมูล (Data Cleaning) จัดรูปแบบข้อความให้เป็นมาตรฐาน

3. แปลงรูปแบบข้อมูล (Data Transformation) ให้อยู่ในโครงสร้างที่เหมาะสมกับฐานข้อมูล

4. ตรวจสอบความถูกต้อง (Validation) เพื่อให้แน่ใจว่าข้อมูลสมบูรณ์

5. บันทึกข้อมูล (Data Loading) ลงในฐานข้อมูล PostgreSQL

การจำลองและติดตั้งระบบด้วย Kind (Kubernetes in Docker)

เพื่อให้สามารถทดสอบระบบในสภาพแวดล้อมจริงแบบ Containerized ได้อย่างมีประสิทธิภาพ โดยการใช้เครื่องมือ Kind (Kubernetes in Docker) ซึ่งเป็นแพลตฟอร์มสำหรับจำลองการทำงานของ Kubernetes บนเครื่องพัฒนา โดยมีโครงสร้างดังนี้

1. Container ของ PySpark สำหรับประมวลผลข้อมูลยา
2. Container ของ Spring Boot (Backend) สำหรับให้บริการ API
3. Container ของ PostgreSQL (Database) สำหรับเก็บข้อมูลยาและข้อมูลผู้ใช้งาน

4.3 การทดสอบและการประเมินผล

การทดสอบมีความสำคัญเพื่อรับประกันว่าแอปพลิเคชันทำงานได้อย่างถูกต้องและมีคุณภาพ

4.3.1 การทดสอบระบบ (System Testing)

ทดสอบการทำงานของฟังก์ชันหลักทั้งหมด ตั้งแต่การเข้าสู่ระบบไปจนถึงการใช้งานฟีเจอร์ต่างๆ เพื่อหาข้อผิดพลาด

4.3.2 การทดสอบประสิทธิภาพ (Performance Testing)

วัดความเร็วและความแม่นยำของฟังก์ชันสำคัญ เช่น ความเร็วในการสแกน Barcode และ OCR

4.3.3 การทดสอบการยอมรับจากผู้ใช้ (User Acceptance Testing - UAT)

นำแอปพลิเคชันไปให้กลุ่มผู้สูงอายุทดลองใช้จริง เพื่อรับรวมข้อเสนอแนะและปัญหาที่พบจากการใช้งาน ซึ่งจะนำไปสู่การปรับปรุงในเวอร์ชันต่อไป

บทที่ 5 การประเมินผล

บทที่ 6
สรุปและข้อเสนอแนะ

เอกสารอ้างอิง

- [1] กรมอนามัย. “สถานการณ์ สภาพภัยไม่ติดต่อเรื้อรัง (Non-Communicable diseases)”; 2022, สืบค้นเมื่อ: 17 สิงหาคม 2568, [ออนไลน์], [Online]. Available: <https://share.google/Dsz4woehlYvHsfYik>.
- [2] ผศ. ดร. พญ. มยุรี หอมสนิท. “NCD โรคไม่ติดต่อ”; 2557, สืบค้นเมื่อ 29 สิงหาคม 2568, [ออนไลน์], [Online]. Available: <https://share.google/fLVqWdcNTidgS22mZ>.
- [3] สำนักพัฒนามาตรฐานระบบข้อมูลสุขภาพไทย. “บัญชีข้อมูลยาและรหัสยามาตรฐานไทย”; 1 2013, สืบค้นเมื่อ: 15 สิงหาคม 2568, [ออนไลน์], [Online]. Available: <https://this.or.th>.
- [4] ClickUp. “Hundreds of powerful tools, one platform.”;, [ออนไลน์], [Online]. Available: <https://clickup.com/features>.
- [5] Alan Dix. “Human-Computer Interaction”;, [ออนไลน์], [Online]. Available: <https://dl.acm.org/doi/10.1145/3544548.3581518>.
- [6] ภญ. วนิช วรachaun. “ระบบการจัดการด้านยาในโรงพยาบาลคุณภาพ”; 2018, สืบค้นเมื่อ 2 กันยายน 2568, [ออนไลน์], [Online]. Available: <https://share.google/ZwsMuslBHQop6T20O>.
- [7] Appmaster. “Fundamentals of Database Systems”; 10 2022, สืบค้นเมื่อ: 1 กันยายน 2568, [ออนไลน์], [Online]. Available: <https://appmaster.io/th/blog/postgresql-khuue-aair>.
- [8] AWS. “OCR (การรู้จำอักษรด้วยแสง) คืออะไร”; 2025, สืบค้นเมื่อ: 20 กันยายน 2568, [ออนไลน์], [Online]. Available: <https://share.google/brT7qg1wnMVyyZ9S9>.
- [9] VMware, Inc. . “Spring Boot: Build anything with Spring Boot”; 2024, สืบค้นเมื่อ: 20 กันยายน 2568, [ออนไลน์], [Online]. Available: <https://spring.io/projects/spring-boot>.
- [10] Google LLC. “Flutter: Build apps for any screen”; 2024, สืบค้นเมื่อ: 20 กันยายน 2568, [ออนไลน์], [Online]. Available: <https://flutter.dev>.
- [11] Dart. “Dart”; 9 2025, สืบค้นเมื่อ: 20 กันยายน 2568, [ออนไลน์], [Online]. Available: <https://dart.dev/overview>.
- [12] ณัฐกรน์ ศรีบุรุษย์. “การพัฒนาและการประเมินระบบโปรแกรมประยุกต์บนโทรศัพท์มือถือ เพื่อช่วยในการอ่านฉลากยาสำหรับผู้สูงอายุ”; มีนาคม 2024, สืบค้นเมื่อ: 20 กันยายน 2568, [ออนไลน์], [Online]. Available: <https://he01.tci-thaijo.org/index.php/TJPP/article/view/260634>.

- [13] วิศวัสดิ์ ประยิษฐ์เสรีสู. “การพัฒนาแอปพลิเคชันบน LINE LIFF ร่วมกับ Google Apps Script สำหรับ ติดตามและประเมินผล ความสม่ำเสมอในการรับประทานยา ของผู้ป่วยวัณโรค โรงพยาบาลชัยนาทเรนทร”; 2024, [ออนไลน์], [Online]. Available: <https://share.google/DS1rM9eBJ3zmpBa8P>.
- [14] วิรุฬห์ ศรีบริรักษ์. “โครงการวิจัยเรื่อง เทคโนโลยีเกสัชสารสนเทศแสดงข้อมูลภารกิยาเอกสารกำกับยาแบบอัตโนมัติสำหรับ บริหารจัดการคลังยาปฏิชีวนะเพื่อความปลอดภัย”; 2018, [ออนไลน์], [Online]. Available: <https://share.google/JaCyThQY9ISZ5uNNf>.
- [15] สมาคมการค้าເຊີ້ຫວີ່ເກົ່າໄທ. “PharmaSee”;, [ออนไลน์], [Online]. Available: <https://www.thaihealthtech.com/thta-applications-and-services/pharmasee-app>.
- [16] NECTEC. “MyYaAndYou”;, [ออนไลน์], [Online]. Available: <https://software.thaiware.com/4396-MyYaAndYou-App.html>.
- [17] Medisafe. “Medisafe”; 2025, [ออนไลน์], [Online]. Available: https://play.google.com/store/apps/details/Pill_Reminder_Medication_Tracker_Medisafe?id=com.medisafe.android.client&hl=th.
- [18] ผศ. ดร. จริยา วิทยาศุกร์ และ ผศ. ดร. จุฬารักษ์ กวีวิธชัย. “RDU”;, [ออนไลน์], [Online]. Available: <https://www.rama.mahidol.ac.th/nursing/sites/default/files/public/knowledge/doc/1.pdf>.
- [19] DIGITORY CONTENT. “PERSONA คืออะไร สำคัญอย่างไรต่อการสร้างแบรนด์”; 5 2023, สืบค้นเมื่อ: 10 กันยายน 2568, [ออนไลน์], [Online]. Available: <https://digitorystyle.com/what-is-persona/>.
- [20] Figma. “User journey mapping: What it is + how to do it”;, [ออนไลน์], [Online]. Available: <https://www.figma.com/resource-library/user-journey-map>.
- [21] Figma. “Figma Design”;, [ออนไลน์], [Online]. Available: <https://www.figma.com/design/>.
- [22] Ihechikara Abba. “Crow’s Foot Notation – Relationship Symbols And How to Read Diagrams”; 6 2022, สืบค้นเมื่อ 29 สิงหาคม 2568, [ออนไลน์], [Online]. Available: <https://www.freecodecamp.org/news/crows-foot-notation-relationship-symbols-and-how-to-read-diagrams>.
- [23] Napat Rammanu. “Data Flow Diagrams จัดการเส้นทางการไหลของข้อมูล”; 6 2025, สืบค้นเมื่อ 29 สิงหาคม 2568, [ออนไลน์], [Online]. Available:

ข้อความเปิดเผยการใช้เทคโนโลยีปัญญาประดิษฐ์

ข้าพเจ้าขอรับรองว่าได้มีการใช้เทคโนโลยี Generative AI ในการจัดทำกิจกรรมหรือชิ้นงานนี้ โดยนำเครื่องมือ Generative AI (....., ,) มาใช้เพื่อวัตถุประสงค์ดังต่อไปนี้:

- ระดมความคิด
- ทบทวนวรรณกรรมและผลงานที่เกี่ยวข้อง
- สรุปและถอดความเนื้อหา
- ร่างชิ้นงานเบื้องต้น
- ตรวจสอบและแก้ไขภาษา ไวยากรณ์หรือการแปล
- แก้ไขและตรวจสอบโค้ด
- เขียนโค้ด
- ใช้ AI เป็นส่วนหนึ่งในผลงาน (เช่น การใช้ generative AI API)
- อื่น ๆ (โปรดระบุ):

นางสาว ปราณนา สุవวงศ์

นาย ฤทธิชล พลราช

นางสาว จิรัชญา ราชพลเสน

ภาคผนวก ก ชื่อของภาคผนวก

ส่วนเพิ่มเติมในเอกสาร นำเสนอด้วยข้อมูลเพิ่มเติมที่ไม่สามารถใส่ไว้ในเนื้อหาหลักของเอกสารได้ เพราะอาจทำให้เนื้อหาหลัก ยาวเกินไป ทำหน้าที่ให้ข้อมูลเสริมที่ช่วยสนับสนุนความเข้าใจในเนื้อหาหลักของรายงาน ตัวอย่างข้อมูลที่ใส่ในภาคผนวก เช่น แบบสอบถาม ข้อมูลดิบ ผลการประเมินแบบละเอียด

ภาคผนวก ข
ชื่อของภาคผนวก