

แอปพลิเคชันสแกนยาและแจ้งเตือนการทานยา

โดย

นางสาว	ปรารธนา	สุภาวงศ์	6587057
นาย	ฤทธิช	พลราช	6587062
นางสาว	จิรัชญา	ราชพลแสน	6587074

อาจารย์ที่ปรึกษา
ดร.สนธิ์ แสงเหลา

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

วิทยาศาสตร์บัณฑิต
(สาขาวิทยาการและเทคโนโลยีดิจิทัล)

คณะเทคโนโลยีสารสนเทศและการสื่อสาร
มหาวิทยาลัยมหิดล
2025

ลิขสิทธิ์ของมหาวิทยาลัยมหิดล

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงได้ด้วยความร่วมมือและความกรุณาจากอาจารย์ที่ปรึกษา ดร.สนธิ แสงเหลา ซึ่งได้ให้คำแนะนำและแนวทางในการพัฒนาโครงการอย่างใกล้ชิด ตลอดจนให้คำปรึกษาและแก้ไขข้อบกพร่องต่าง ๆ มาโดยตลอด จนทำให้โครงการเล่มนี้เสร็จสมบูรณ์ ผู้จัดทำจึงขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณสมาชิกในกลุ่มทุกคน ได้แก่ นางสาวปรารถนา สุภาวงศ์, นายฤทธิช พลราช และ นางสาวจิรัชญา ราชพลแสน ที่ร่วมมือร่วมใจกันทำงานอย่างเต็มความสามารถ แลกเปลี่ยนความคิดเห็น และช่วยเหลือซึ่งกันและกันตลอดระยะเวลาการทำโครงการ

ขอขอบคุณผู้ร่วมทดสอบระบบทุกท่านที่ได้สละเวลาอันมีค่าในการทดสอบและให้ข้อเสนอแนะที่เป็นประโยชน์ ซึ่งช่วยให้ระบบมีความสมบูรณ์และตอบสนองความต้องการของผู้ใช้งานได้ดียิ่งขึ้น

ขอขอบคุณสำนักพัฒนามาตรฐานระบบข้อมูลสุขภาพไทยที่ได้เอื้อเพื่อข้อมูลจากระบบ Thai Medicine Terminology (TMT) ซึ่งเป็นแหล่งข้อมูลด้านยาที่มีความน่าเชื่อถือ เพื่อนำมาใช้ในการพัฒนาและทดสอบระบบในโครงการนี้

สุดท้ายนี้ ขอขอบคุณครอบครัวและผู้ปกครองที่ให้การสนับสนุน ให้กำลังใจ และเป็นแรงผลักดันให้ผู้จัดทำสามารถดำเนินโครงการนี้จนสำเร็จลุล่วงได้ด้วยดี

นางสาว ปรารถนา สุภาวงศ์

นาย ฤทธิช พลราช

นางสาว จิรัชญา ราชพลแสน

แอปพลิเคชันสแกนยาและแจ้งเตือนการทานยา

นางสาว ปรรธนา สุภาวงศ์ 6587057 ITDS/B

นาย ฤทธิช พลราช 6587062 ITDS/B

นางสาว จิรัชญา ราชพลแสน 6587074 ITDS/B

วท.บ. (วิทยาการและเทคโนโลยีดิจิทัล)

อาจารย์ที่ปรึกษาโครงการ: ดร.สนธิ แสงเหลา

บทคัดย่อ

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

ระบบพัฒนาด้วย Flutter (Dart) สำหรับฝั่ง Mobile Application และ Spring Boot ร่วมกับ PostgreSQL สำหรับฝั่งเซิร์ฟเวอร์ มีการประมวลผลข้อมูลยาจากฐานข้อมูล 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

52 หน้า

สารบัญ

หน้า

กิตติกรรมประกาศ	ii
บทคัดย่อ	iii
1 บทนำ	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: การรวบรวมปัญหาและวิเคราะห์ความต้องการของผู้ใช้งาน	3
1.4.2 ขั้นตอนที่ 2: การออกแบบและพัฒนาระบบในส่วนของหน้าบ้าน (FRONTEND)	3
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	4
1.4.6 ขั้นตอนที่ 6: การจัดทำเอกสารและนำเสนอ	4
1.5 ประโยชน์ที่คาดว่าจะได้รับ	5
2 ทฤษฎีที่เกี่ยวข้อง	6
2.1 สังคมสูงวัย (AGED SOCIETY และ SUPER-AGED SOCIETY)	6
2.2 โรคติดต่อเรื้อรัง (NON-COMMUNICABLE DISEASES: NCDS)	6
2.3 การออกแบบระบบที่เหมาะสมกับผู้สูงอายุ (HUMAN-COMPUTER INTERACTION: HCI)	6
2.4 การจัดการการใช้จ่าย	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	14
2.11	DART	14
2.11.1	คุณสมบัติเด่นของ DART	14
2.12	คำศัพท์	15
3	ผลงานที่เกี่ยวข้อง	16
3.1	แอปพลิเคชันช่วยอ่านฉลากยาโดยเฉพาะฟังก์ชันที่ช่วยให้อ่านข้อมูลยาได้ง่ายขึ้นอย่างชัดเจน	16
3.2	แอปพลิเคชันบน LINE LIFF ร่วมกับ GOOGLE APPS SCRIPT สำหรับติดตามและประเมินผลความสำเร็จในการรับประทานยาของผู้ป่วยวัณโรค โรงพยาบาลชยันนาทนเรนทร	16
3.3	โครงการวิจัยเรื่อง เทคโนโลยีเภสัชสารสนเทศแสดงข้อมูลฉลากยาเอกสารกำกับยาแบบอัตโนมัติสำหรับบริหารจัดการคลังยาปฏิชีวนะเพื่อความปลอดภัย	17
3.4	การเปรียบเทียบแอปพลิเคชันที่มีอยู่ในตลาด	18
3.4.1	ตารางเปรียบเทียบแอปพลิเคชันในตลาด	18
3.4.2	สรุปการเปรียบเทียบ	19
3.4.3	การวิเคราะห์ช่องว่างในตลาด	20
3.5	ประสบการณ์การพัฒนาที่ผ่านมา	20
3.5.1	APPLICATION CHAT MUSIC PLAYER	20

3.5.2	APPLICATION LOCATION DISCOVERY	21
3.5.3	บทเรียนที่ได้จากประสบการณ์การพัฒนา.....	21
3.6	สรุปและการนำไปประยุกต์ใช้	22
3.6.1	แนวทางการพัฒนาที่ได้จากงานวิจัย	22
3.6.2	ข้อจำกัดที่ต้องหลีกเลี่ยง	22
3.6.3	คุณสมบัติที่จะนำมาพัฒนาในโครงการ	22
4	ระเบียบวิธีดำเนินโครงการ	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)	26
4.1.5	การออกแบบสถาปัตยกรรม (SYSTEM ARCHITECTURE DESIGN).....	28
4.1.6	การออกแบบฐานข้อมูล (DATABASE DESIGN)	31
4.1.7	ขั้นตอนการออกแบบระบบ	31
4.2	การพัฒนาและติดตั้ง (DEVELOPMENT AND IMPLEMENTATION).....	43
4.2.1	เครื่องมือและสภาพแวดล้อมในการพัฒนา	43
4.2.2	ขั้นตอนการพัฒนา	43
4.3	การทดสอบและการประเมินผล.....	44
4.3.1	การทดสอบระบบ (SYSTEM TESTING).....	44
4.3.2	การทดสอบประสิทธิภาพ (PERFORMANCE TESTING)	44
4.3.3	การทดสอบการยอมรับจากผู้ใช้งาน (USER ACCEPTANCE TESTING - UAT)	45
5	การประเมินผล.....	46
6	สรุปและข้อเสนอแนะ.....	47
	เอกสารอ้างอิง	48
	ข้อความเปิดเผยการใช้เทคโนโลยีปัญญาประดิษฐ์	50
	ภาคผนวก ก	51
	ภาคผนวก ข	52

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ประเทศไทยกำลังเผชิญหน้ากับการเปลี่ยนแปลงทางประชากรครั้งใหญ่ โดยกำลังก้าวเข้าสู่ "สังคมสูงวัยระดับสุดยอด" (Super-Aged Society) ซึ่งเป็นผลมาจากสัดส่วนประชากรผู้สูงอายุที่เพิ่มขึ้นอย่างรวดเร็ว 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 (กรมอนามัย, 2566)

การเข้าสู่สังคมสูงวัยอย่างรวดเร็วของประเทศไทยส่งผลให้เกิดความท้าทายด้านสุขภาพอย่างหลีกเลี่ยงไม่ได้ โดยเฉพาะอย่างยิ่งในเรื่องของ กลุ่มโรคไม่ติดต่อเรื้อรัง (Non-Communicable Diseases หรือ NCDs) ซึ่งเป็นสาเหตุการเสียชีวิตอันดับต้น ๆ ของประชากรทั่วโลก ปัจจัยเสี่ยงสำคัญ ได้แก่ การสูบบุหรี่ การไม่ออกกำลังกาย การรับประทานอาหารไม่เหมาะสม และการดื่มแอลกอฮอล์มากเกินไป ซึ่งนำไปสู่โรคหัวใจ มะเร็ง โรคระบบหายใจเรื้อรัง เบาหวาน และความดันโลหิตสูง ("NCD โรคไม่ติดต่อ" โดย ผศ.ดร.พญ.มยุรี หอมสนิท, 2557)

ผู้สูงอายุที่ป่วยด้วยโรคไม่ติดต่อเรื้อรัง (NCDs) มักต้องจัดการการใช้ยาที่ซับซ้อน เช่น การจดจำชนิดของยา เวลารับประทาน และผลข้างเคียง การลืมทานยาหรือทานผิดเวลาพบได้บ่อย และส่งผลต่อประสิทธิภาพการรักษา ขณะเดียวกัน แอปพลิเคชันสุขภาพทั่วไปยังไม่เหมาะสมกับการใช้งานของผู้สูงอายุ โดยเฉพาะในด้านการออกแบบหน้าจอ (UI) ที่ซับซ้อน

จากความท้าทายเหล่านี้ การพัฒนาแอปพลิเคชันที่สามารถแก้ไขปัญหาดังกล่าวได้จึงมีความสำคัญอย่างยิ่ง แอปพลิเคชันนี้จะทำหน้าที่เป็นเครื่องมือที่ใช้งานง่าย มีข้อมูลที่น่าเชื่อถือ และมีฟังก์ชันที่ช่วยลดความผิดพลาดในการใช้ยา เพื่อช่วยให้ทั้งผู้ป่วยสามารถจัดการสุขภาพได้อย่างมีประสิทธิภาพและปลอดภัยสูงสุดในยุคที่ประเทศไทยกำลังก้าวเข้าสู่สังคมสูงวัยอย่างเต็มตัว

1.2 วัตถุประสงค์

- เพื่อพัฒนาแอปพลิเคชันสำหรับแจ้งเตือน บันทึกการทานยา และบันทึกอาการหลังทานยา เพื่อช่วยให้ผู้ใช้งานจดจำและทานยาได้ถูกต้องสม่ำเสมอ พร้อมระบบเพิ่ม-แก้ไขข้อมูลยาได้ด้วยตนเอง
- เพื่อเพิ่มฟังก์ชันที่ตอบสนองความต้องการของผู้สูงอายุ ได้แก่
 - การแจ้งเตือนเพื่อช่วยเหลือผู้สูงอายุที่มีข้อจำกัดในการได้ยินและการมองเห็นจากการใช้เสียงแจ้งเตือนที่ดังและชัดเจน และยังมีแจ้งเตือนด้วยการสั่น

– การแปลงข้อความให้เป็นเสียงเพื่ออำนวยความสะดวกในการเข้าถึงข้อมูลสำหรับผู้ที่มีปัญหาด้านการมองเห็น

- เพื่อสร้างฐานข้อมูลยาที่น่าเชื่อถือ โดยจะรวบรวมข้อมูลยาที่สำคัญจากจากระบบ Thai Medicine Terminology (TMT)
- เพื่อให้ผู้ใช้งานสามารถเพิ่ม, แก้ไข, และลบข้อมูลยาที่กำลังทานอยู่หรือบันทึกประวัติการทานยาได้
- เพื่อช่วยให้ผู้ใช้งานสามารถบันทึกและติดตามประวัติการทานยาในแต่ละวัน เพื่อให้ผู้ใช้งานนำข้อมูลไปปรึกษาแพทย์ได้อย่างถูกต้อง และเพื่อให้ผู้ใช้งานสามารถตรวจสอบความสม่ำเสมอในการทานยาได้อย่างแม่นยำ

1.3 ขอบเขตงาน

1.3.1 ด้านฟังก์ชันการทำงาน

- ผู้ใช้งานสามารถเพิ่ม, แก้ไข, และลบข้อมูลยาที่กำลังทานอยู่หรือบันทึกประวัติการทานยาได้
- ผู้ใช้งานสามารถค้นหาข้อมูลยาเพื่อดูรายละเอียดเพิ่มเติมได้ เช่น ชื่อยา, วิธีใช้, ผลข้างเคียง
- พัฒนาระบบแจ้งเตือนที่สามารถแจ้งเตือนผู้ใช้งานได้ตามเวลาที่กำหนดไว้
- มีระบบแปลงข้อความเป็นเสียงสำหรับการอ่านข้อมูลยาให้ผู้ใช้งานฟัง

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) สำหรับการทดสอบการทำงานของระบบ 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 ด้านการศึกษา

- การศึกษาเกี่ยวกับการพัฒนาแอปพลิเคชันสำหรับผู้สูงอายุ
- ศึกษาการทำงานของ Flutter และ Dart
- ศึกษาการทำงานของ Spring Boot
- ศึกษาการทำงานของ PySpark
- ศึกษาการทำงานของ kind
- ศึกษาการทำงานของ package flutter_tesseract_ocr
- 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: การรวบรวมปัญหาและวิเคราะห์ความต้องการของผู้ใช้งาน

ในขั้นตอนแรกของโครงการ ได้ทำการศึกษาและวิเคราะห์ปัญหาของกลุ่มเป้าหมาย ซึ่งเป็นผู้สูงอายุที่ต้องรับประทานยาหลายชนิดในแต่ละวัน รวมถึงศึกษาข้อจำกัดทางด้านการมองเห็นและการได้ยิน จากนั้นจึงทำการเก็บรวบรวมความต้องการของผู้ใช้งาน (User Requirement) โดยการสัมภาษณ์และสำรวจความคิดเห็นจากกลุ่มผู้สูงอายุและผู้ดูแล เพื่อนำมาวิเคราะห์และกำหนดฟังก์ชันการทำงานของแอปพลิเคชันที่ตอบโจทย์ความต้องการได้อย่างแท้จริง

1.4.2 ขั้นตอนที่ 2: การออกแบบและพัฒนาระบบในส่วนของหน้าบ้าน (Frontend)

ในขั้นตอนนี้ได้ทำการออกแบบและพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android โดยใช้ Flutter Framework ร่วมกับภาษา Dart เพื่อสร้างส่วนติดต่อผู้ใช้งาน (User Interface) ที่เหมาะสมกับผู้สูงอายุ โดยเน้นความเรียบง่าย ใช้งานง่าย มีตัวอักษรขนาด

ใหญ่ ปุ่มที่มีขนาดใหญ่และเห็นได้ชัดเจน รวมถึงการเลือกใช้สีที่มีความคมชัดเพื่อให้ผู้สูงอายุมองเห็นได้ง่าย นอกจากนี้ยังได้พัฒนาฟีเจอร์การแจ้งเตือนด้วยเสียง การสั่น และการแปลงข้อความเป็นเสียง (Text-to-Speech) เพื่ออำนวยความสะดวกให้กับผู้ใช้งาน

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 และการดึงข้อมูลยาจากฐานข้อมูล นอกจากนี้ เพื่อให้ระบบสามารถทดสอบได้ในสภาพแวดล้อมเสมือนจริง ได้มีการใช้เครื่องมือ Kind (Kubernetes in Docker) ในการจำลองคลัสเตอร์ Kubernetes ภายในเครื่อง เพื่อทดสอบการทำงานของระบบในรูปแบบ Container

1.4.4 ขั้นตอนที่ 4: การศึกษาและพัฒนา Data Pipeline ด้วย PySpark และ Kind

ขั้นตอนนี้เป็นการออกแบบและพัฒนาระบบการ Data Pipeline เพื่อจัดการข้อมูลยาจากแหล่งข้อมูลภายนอก โดยเฉพาะข้อมูลจากระบบ Thai Medicine Terminology (TMT) จากสำนักพัฒนามาตรฐานระบบข้อมูลสุขภาพไทย ซึ่งให้ข้อมูลยาในรูปแบบไฟล์ Excel โดยใช้ PySpark ในการประมวลผล ทำความสะอาดข้อมูล (Data Cleaning) และแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสมก่อนนำเข้าฐานข้อมูล PostgreSQL เพื่อให้ระบบ Data Pipeline สามารถทำงานร่วมกับส่วน Backend และ Database ได้อย่างราบรื่น ได้มีการนำ Kind มาใช้ควบคุมการทำงานของ Container ทั้งหมด โดยภายใน Cluster จะประกอบด้วย

- Container ของ PySpark สำหรับประมวลผลและจัดการข้อมูลยา
- Container ของ Spring Boot สำหรับให้บริการ API
- Container ของ PostgreSQL สำหรับเก็บข้อมูลที่ผ่านการประมวลผลแล้ว

1.4.5 ขั้นตอนที่ 5: การทดสอบและปรับปรุงระบบ

ในขั้นตอนนี้ทำการทดสอบ ระบบ ทั้งหมด เพื่อตรวจสอบ ความถูกต้อง และ ประสิทธิภาพของแอปพลิเคชัน โดยแบ่งการทดสอบออกเป็น Unit Testing สำหรับทดสอบฟังก์ชันแต่ละส่วน Integration Testing สำหรับทดสอบการทำงานร่วมกันระหว่างส่วนต่างๆ และ User Acceptance Testing (UAT) โดยให้กลุ่มผู้สูงอายุทดลองใช้งานจริง จากนั้นรวบรวม Feedback และข้อเสนอแนะจากผู้ทดสอบเพื่อนำมาปรับปรุงแก้ไขระบบให้มีความสมบูรณ์และตอบสนองความต้องการของผู้ใช้งานได้ดียิ่งขึ้น

1.4.6 ขั้นตอนที่ 6: การจัดทำเอกสารและนำเสนอ

ขั้นตอนสุดท้ายเป็นการจัดทำเอกสารรายงานโครงการ โดยรวบรวมข้อมูลทั้งหมดตั้งแต่การศึกษาปัญหา การวิเคราะห์ความต้องการ กระบวนการพัฒนา ผลการทดสอบ และข้อสรุป พร้อมทั้งเตรียมเอกสารคู่มือการใช้งานสำหรับผู้ใช้ (User Manual) และเอกสารทางเทคนิคสำหรับผู้พัฒนา (Technical Documentation) นอกจากนี้ยังได้เตรียมการนำเสนอผลงานโครงการต่ออาจารย์ที่ปรึกษาและคณะกรรมการ พร้อมทั้งสาธิตการทำงานของแอปพลิเคชันให้เห็นถึงฟังก์ชันและความสามารถของระบบอย่างชัดเจน

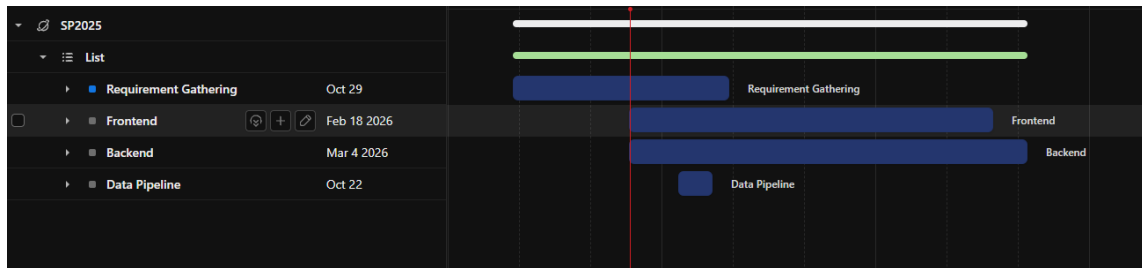


Figure 1.1: แผนการดำเนินโครงการ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- ผู้ใช้งานจะสามารถทานยาได้อย่างถูกต้องและสม่ำเสมอตามคำสั่งแพทย์
- ผู้สูงอายุและผู้ที่มีปัญหาด้านการมองเห็นและการได้ยินจะสามารถเข้าถึงข้อมูลยาได้อย่างปลอดภัยและง่ายมากยิ่งขึ้น
- ผู้ใช้งานจะได้รับข้อมูลที่ถูกต้องและมีแหล่งอ้างอิงที่ชัดเจน
- ช่วยให้ผู้ใช้งานลดความกังวลในการจัดการยาด้วยตัวเองและไม่ลืมที่จะทานยา
- ช่วยให้การให้ข้อมูลการทานยาแก่แพทย์เป็นไปอย่างแม่นยำและถูกต้อง ส่งผลให้การวินิจฉัยและการรักษามีประสิทธิภาพยิ่งขึ้น

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 สังคมสูงวัย (Aged Society และ Super-Aged Society)

คาดว่าประมาณปี พ.ศ. 2576 ประเทศไทยจะก้าวเข้าสู่ สังคมสูงวัยระดับสุดยอด (Super-Aged Society) ซึ่งมีประชากรอายุเกิน 60 ปี มากกว่าร้อยละ 28 ของประชากรทั้งหมด [?] การเปลี่ยนแปลงนี้ทำให้เกิดปัญหาด้านสุขภาพและความต้องการเทคโนโลยีเพื่อช่วยเหลือการดูแลผู้สูงอายุ โดยเฉพาะในเรื่อง การจัดการการใช้ยา ซึ่งมีความซับซ้อนและมีความเสี่ยงต่อการใช้ยาผิดพลาด

2.2 โรคไม่ติดต่อเรื้อรัง (Non-Communicable Diseases: NCDs)

ผู้สูงอายุจำนวนมากป่วยด้วยโรคเรื้อรัง เช่น ความดันโลหิตสูง เบาหวาน โรคหัวใจ ซึ่งต้องใช้ยาหลายชนิดต่อเนื่องเป็นเวลานาน [1] การลืมหยาหรือทานผิดเวลาอาจส่งผลกระทบต่อประสิทธิภาพการรักษา และก่อให้เกิดภาวะแทรกซ้อน [2] การพัฒนาแอปพลิเคชันที่ช่วย แจ้งเตือนและบันทึกประวัติการใช้ยา จึงตอบโจทย์การจัดการโรคเรื้อรังอย่างมีระบบ [3]

2.3 การออกแบบระบบที่เหมาะสมกับผู้สูงอายุ (Human-Computer Interaction: HCI)

ผู้สูงอายุมีข้อจำกัดทั้งด้าน สายตา การได้ยิน ความจำ และความคุ้นเคยกับเทคโนโลยี การออกแบบปฏิสัมพันธ์ระหว่างมนุษย์กับคอมพิวเตอร์ (HCI) เป็นสาขาวิชาที่เน้นการทำความเข้าใจผู้ใช้และนำความเข้าใจนั้นมาออกแบบระบบให้สอดคล้องกับพฤติกรรมและความต้องการของผู้ใช้งาน [4] หัวใจหลักของ HCI คือการศึกษาว่าผู้ใช้คิดอย่างไรและมีพฤติกรรมอย่างไร ซึ่งสำหรับผู้สูงอายุ จะต้องคำนึงถึงปัจจัยเฉพาะทางกายภาพและสมอง เช่น ปัญหาด้านสายตา, ปัญหาด้านการได้ยิน และปัญหาด้านความจำ การออกแบบแอปพลิเคชันจึงต้องยึดหลัก Accessible Design ได้แก่

- ขนาดตัวอักษรใหญ่ อ่านง่าย
- สีตัดกันชัดเจน (Contrast)
- ระบบแจ้งเตือนหลายรูปแบบ (เสียง, สั่น, ไฟแจ้งเตือน)

2.4 การจัดการการใช้ยา

การจัดการระบบยาเป็นสิ่งสำคัญอย่างยิ่งที่เกี่ยวกับความปลอดภัยของผู้ป่วยหรือผู้ใช้บริการในโรงพยาบาล ระบบยาที่ครอบคลุมสะท้อนให้เห็นถึงการทำงานร่วมกันของทีมสหสาขาวิชาชีพ เพื่อให้การบริหารจัดการระบบยาในโรงพยาบาลมีประสิทธิภาพและปลอดภัย

เกณฑ์การพิจารณาคุณภาพยา:

- โรงงานผู้ผลิตที่ได้รับการรับรอง GMP (Good Manufacturing Practice)
- ยามีคุณลักษณะเฉพาะตามมาตรฐานที่ได้รับการรับรองในประเทศไทย
- มีใบตรวจวิเคราะห์คุณภาพยาจากบริษัทหรือจากกรมวิทยาศาสตร์การแพทย์ (ภญ.วณินฯ วราชน, 2561)

2.5 พื้นฐานและการออกแบบฐานข้อมูล

การออกแบบฐานข้อมูลที่ดีมีความสำคัญอย่างยิ่งต่อความถูกต้องและความสมบูรณ์ของข้อมูล และช่วยให้สามารถเรียกใช้หรือปรับปรุงข้อมูลได้อย่างมีประสิทธิภาพ [5] โดยหลักการออกแบบฐานข้อมูลที่ดี มีดังนี้

- ลดความซ้ำซ้อนของข้อมูล (Data Redundancy)
- เพิ่มความถูกต้องสมบูรณ์ของข้อมูล (Data Integrity)
- สามารถปรับปรุงหรือเปลี่ยนแปลงโครงสร้างได้ง่าย

2.6 Optical Character Recognition (OCR)

หรือที่รู้จักกันในภาษาไทยคือ "การรู้จำอักขระด้วยแสง" คือเทคโนโลยีที่ใช้ในการแปลงข้อความจากรูปแบบภาพ (เช่น รูปถ่าย, ภาพสแกน, ไฟล์ PDF ที่เป็นภาพ) ให้กลายเป็นข้อความดิจิทัลที่คอมพิวเตอร์สามารถอ่าน, แก้ไข, ค้นหา หรือนำไปใช้ประมวลผลต่อได้ [2]

ประเภทของ OCR

- ซอฟต์แวร์รู้จำอักขระด้วยแสงอย่างง่าย : ใช้อัลกอริทึมการจับคู่รูปแบบเพื่อเปรียบเทียบรูปภาพข้อความกับฐานข้อมูลภายในแบบอักขระทีละตัว
- ซอฟต์แวร์รู้จำอักขระแบบอัจฉริยะ (Intelligent Character Recognition: ICR) : ใช้ Machine Learning และ Neural Network เพื่ออ่านลายมือหรือตัวอักษรที่ซับซ้อน
- การรู้จำคำแบบอัจฉริยะ (Intelligent Word Recognition) : วิเคราะห์คำทั้งคำแทนการประมวลผลทีละตัวอักษร
- การรู้จำเครื่องหมายด้วยแสง (Optical Mark Recognition: OMR) : ใช้ระบุสัญลักษณ์หรือเครื่องหมาย เช่น โลโก้ หรือตัวเลือกในแบบฟอร์มสอบ

ข้อดีของ OCR

- แปลงเอกสารกระดาษเป็นข้อมูลดิจิทัลที่สามารถค้นหาได้
- สามารถประมวลผลข้อมูลตัวอักษรโดยอัตโนมัติ เพื่อใช้ในการวิเคราะห์และประมวลผลความรู้เพิ่มเติม
- ลดเวลาการประมวลผลเอกสารและการป้อนข้อมูลด้วยตนเอง

- OCR มักถูกรวมเข้ากับเทคโนโลยี AI เพื่อประยุกต์ใช้งานขั้นสูง เช่น อ่านป้ายทะเบียนและป้ายจราจรในรถยนต์ไร้คนขับ (OCR (การรู้จำอักขระด้วยแสง) คืออะไร โดย AWS)

2.7 PostgreSQL

PostgreSQL เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System: RDBMS) ระดับองค์กรที่มีความเสถียรและเชื่อถือได้สูง โดยรองรับการทำงานแบบ ACID (Atomicity, Consistency, Isolation, Durability) อย่างสมบูรณ์ ซึ่งหมายถึงคุณสมบัติพื้นฐานที่ทำให้การทำธุรกรรม (Transaction) ในฐานข้อมูลมีความถูกต้องและปลอดภัย ได้แก่:

- Atomicity คือ การทำธุรกรรม จะต้องสำเร็จทั้งหมดหรือไม่สำเร็จเลย หากเกิดข้อผิดพลาด ระบบจะยกเลิก (Rollback) การทำงานทั้งหมดเพื่อให้ข้อมูลคงสภาพเดิม
- Consistency คือ หลังการทำธุรกรรม ข้อมูลในฐานข้อมูลจะต้องยังคงถูกต้องตามกฎและข้อจำกัด (Constraints) ที่กำหนดไว้เสมอ
- Isolation คือ การทำธุรกรรมหลายรายการที่เกิดขึ้นพร้อมกันจะไม่ส่งผลกระทบต่อกัน ทำให้ผลลัพธ์เทียบเท่ากับการทำงานแบบลำดับ (Serial Execution)
- Durability คือ เมื่อทำธุรกรรมเสร็จสิ้น ข้อมูลที่เปลี่ยนแปลงแล้วจะถูกบันทึกถาวรในระบบ แม้ระบบจะปิดตัวหรือเกิดความขัดข้อง

PostgreSQL สามารถรองรับการประมวลผลได้ทั้งสองรูปแบบหลักของระบบฐานข้อมูล คือ:

- OLTP (Online Transaction Processing): ใช้สำหรับงานที่ต้องการการประมวลผลธุรกรรมจำนวนมากอย่างรวดเร็ว เช่น ระบบธนาคาร ระบบขายสินค้า หรือระบบการจองต่าง ๆ ที่มีการเพิ่ม ลบ หรือแก้ไขข้อมูลอยู่ตลอดเวลา
- OLAP (Online Analytical Processing): ใช้สำหรับการวิเคราะห์ข้อมูลเชิงลึก เช่น การสรุปยอดขาย การวิเคราะห์แนวโน้ม หรือการทำรายงานทางธุรกิจ ที่ต้องใช้ข้อมูลจำนวนมากจากหลายแหล่งมาคำนวณรวมกัน

นอกจากนี้ PostgreSQL ยังรองรับการทำงานข้ามแพลตฟอร์มได้ โดยสามารถติดตั้งและใช้งานได้บนหลายระบบปฏิบัติการ เช่น Windows, Linux, macOS, FreeBSD และ OpenBSD รวมถึงยังมีความสามารถในการขยายเพิ่มเติม (Extensibility) ด้วยการสร้างฟังก์ชัน ข้อมูลชนิดใหม่ หรือโมดูลเสริม (Extensions) เพื่อเพิ่มขีดความสามารถให้ตรงตามความต้องการของผู้ใช้งานอีกด้วย

2.7.1 คุณสมบัติที่สำคัญ

รองรับข้อมูลที่หลากหลาย

- ข้อมูลเชิงโครงสร้าง (Structured Data) เช่น Integer, Text, Date
- ข้อมูลเชิงไม่โครงสร้าง เช่น JSON และ JSONB
- ข้อมูลเชิงพื้นที่ (Spatial Data) ผ่านส่วนเสริม PostGIS

การทำงานแบบ Multi-Version Concurrency Control (MVCC)

ช่วยให้ผู้ใช้หลายคนเข้าถึงข้อมูลพร้อมกันโดยไม่เกิดการขัดแย้ง

2.7.2 ประโยชน์ของ PostgreSQL

- สามารถใช้งานได้โดยไม่ต้องชำระเงินและเป็น Open Source ไม่มีค่าใช้จ่ายด้านลิขสิทธิ์
- มีความน่าเชื่อถือสูง รองรับการทำงานต่อเนื่อง (Fault Tolerant)
- รองรับหลายภาษาโปรแกรม เช่น Python, Java, C/C++, PHP, Ruby, Perl, Erlang, Lua, Scala, Haskell, etc.
- รองรับการทำงานข้ามแพลตฟอร์ม [5]

2.8 สถาปัตยกรรมแบบ Client-Server (Client-Server Architecture)

สถาปัตยกรรมแบบ Client-Server เป็นรูปแบบการออกแบบระบบคอมพิวเตอร์ที่แบ่งการทำงานออกเป็น 2 ส่วนหลัก คือ Client (ฝั่งผู้ใช้) และ Server (ฝั่งเซิร์ฟเวอร์) โดยทั้งสองส่วนจะสื่อสารกันผ่านเครือข่าย (Network) เพื่อแลกเปลี่ยนข้อมูลและประมวลผลงานร่วมกัน

2.8.1 องค์ประกอบหลัก

Client (ฝั่งผู้ใช้)

Client คือแอปพลิเคชันหรือโปรแกรมที่ทำงานบนอุปกรณ์ของผู้ใช้ เช่น มือถือ คอมพิวเตอร์ หรือแท็บเล็ต มีหน้าที่หลักดังนี้

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

ในโครงงานนี้ แอปพลิเคชันที่พัฒนาด้วย Flutter [6] และ Dart [7] ทำหน้าที่เป็น Client ซึ่งทำงานบนอุปกรณ์ Android ของผู้ใช้

Server (ฝั่งเซิร์ฟเวอร์)

Server คือระบบที่ทำงานบนเครื่องแม่ข่าย มีหน้าที่รับคำขอจาก Client และประมวลผลตามคำขอนั้น ได้แก่

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

- ประมวลผลตรรกะทางธุรกิจ (Business Logic)
- ส่งผลลัพธ์ (Response) กลับไปยัง Client

ในโครงงานนี้ Spring Boot Application [8] ทำหน้าที่เป็น Server ที่ให้บริการ REST API และเชื่อมต่อกับฐานข้อมูล PostgreSQL

2.8.2 การทำงานของระบบ Client-Server ในโครงงาน

การสื่อสารระหว่าง Client และ Server ในโครงงานนี้เป็นไปตามขั้นตอนดังนี้

1. ผู้ใช้ได้ตอบกับแอปพลิเคชัน Flutter (Client) เช่น ค้นหาข้อมูลยา หรือบันทึกการทานยา
2. Client ส่งคำขอ HTTP Request ไปยัง Server ผ่าน REST API
3. Spring Boot Server รับคำขอและประมวลผล โดยอาจเข้าถึงฐานข้อมูล PostgreSQL เพื่อดึงหรือบันทึกข้อมูล
4. Server ส่งผลลัพธ์กลับมาในรูปแบบ HTTP Response (มักเป็น JSON)
5. Client รับข้อมูลและแสดงผลบนหน้าจอให้ผู้ใช้เห็น

2.8.3 ข้อดีของสถาปัตยกรรม Client-Server

- การแยกหน้าที่ชัดเจน (Separation of Concerns) — 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 ทำให้ง่ายต่อการสำรองข้อมูลและการรักษาความปลอดภัย

2.8.4 การประยุกต์ใช้ในโครงงาน

โครงงานนี้ใช้สถาปัตยกรรม Client-Server โดยมีองค์ประกอบดังนี้

- Client: แอปพลิเคชัน Flutter บน Android ใช้สำหรับแสดงข้อมูลยา, แจ้งเตือนการทานยา, บันทึกประวัติ

- Server: Spring Boot Application ใช้สำหรับการจัดการ REST API, ตรรกะทางธุรกิจ, การเข้าถึงฐานข้อมูล
- Database: PostgreSQL ใช้สำหรับจัดเก็บข้อมูลยา, ข้อมูลผู้ใช้, ประวัติการทานยา
- Data Pipeline: PySpark ใช้สำหรับประมวลผลข้อมูลยาจาก Thai Medicine Terminology (TMT) ก่อนนำเข้าฐานข้อมูล

การใช้สถาปัตยกรรมนี้ทำให้ระบบมีความยืดหยุ่น ปลอดภัย และสามารถขยายการทำงานได้ในอนาคต เช่น การเพิ่มแพลตฟอร์ม iOS หรือ Web โดยยังคงใช้ Backend เดียวกัน

2.9 Spring Boot Framework

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

2.9.1 คุณสมบัติหลัก

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

2.9.2 การใช้งานในโครงการ

ในโครงการนี้ Spring Boot ถูกใช้เพื่อ

- สร้าง REST API endpoints สำหรับการสื่อสารระหว่าง Flutter Client และฐานข้อมูล
- จัดการ CRUD operations (Create, Read, Update, Delete) สำหรับข้อมูลยาและผู้ใช้
- ตรวจสอบความถูกต้องของข้อมูล (Validation)
- จัดการการเชื่อมต่อกับ PostgreSQL Database

2.10 Flutter Framework

เป็น Framework Open Source เปิดตัวในปี 2018 ที่พัฒนาโดย Google ใช้สร้างอินเทอร์เฟซผู้ใช้ (UI) ของแอปพลิเคชันสำหรับแพลตฟอร์ม iOS, Android, Web, Windows, macOS และ Linux โดยใช้ Codebase เดียวในการสร้างแอปมือถือ เว็บ และเดสก์ท็อป (Windows, macOS, Linux) ได้ ซึ่ง Flutter ช่วยลดความซับซ้อนของการสร้าง UI ที่สวยงามและสอดคล้องกัน บนหลายแพลตฟอร์ม (AWS, 2024)

2.10.1 เปรียบเทียบการพัฒนาแอปแบบเนทีฟ (Native) และข้ามแพลตฟอร์ม (Cross-platform)

Native

- เขียนแยกเฉพาะแต่ละแพลตฟอร์ม เช่น Swift สำหรับ iOS, Kotlin/Java สำหรับ Android
- สามารถเข้าถึงฟังก์ชันของอุปกรณ์ได้อย่างเต็มประสิทธิภาพ
- มีประสิทธิภาพสูง เนื่องจากพัฒนาแพลตฟอร์มอย่างเฉพาะเจาะจง แต่ต้องเขียนหลายโค้ดเบส และมีค่าใช้จ่ายที่สูง

Cross-platform

- สามารถใช้โค้ดเบสเดียวกันสำหรับหลายแพลตฟอร์ม
- ประหยัดเวลาและค่าใช้จ่าย
- อาจเข้าถึงฟังก์ชันบางอย่างของอุปกรณ์ได้อย่างจำกัด
- ประสบการณ์การใช้งานของผู้ใช้สอดคล้องกัน

2.10.2 ข้อดีของ Flutter

- ประสิทธิภาพใกล้เคียงแบบ Native แต่มีค่าใช้จ่ายที่ต่ำกว่า
- Render UI เร็วและสม่ำเสมอ เนื่องจากใช้ Engine Graphic ของตัวเอง แสดงผล UI สอดคล้องกันข้ามแพลตฟอร์ม
- รองรับการทำงานขนาน (Parallel Processing)
- 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 ทำให้เหมาะกับการพัฒนาแอปข้ามแพลตฟอร์มด้วย Flutter

คุณสมบัติหลัก

- พูดข้อความ (speak)
- ดึงรายการภาษาที่รองรับ (getLanguages)
- ตรวจสอบว่าภาษานั้นใช้งานได้หรือไม่ (isLanguageAvailable)
- กำหนดภาษา (setLanguage)

- เลือกเสียง (voice) ที่เหมาะสมกับภาษาและสำเนียง
- ความเร็วในการพูด (setSpeechRate)
- ระดับเสียง (setVolume)
- ระดับโทนเสียง (setPitch)

ข้อดี

- ลดภาระการอ่านข้อความยาว ๆ โดยเฉพาะในเอกสารกำกับยา
- ช่วยผู้ใช้ที่อ่านภาษาได้ไม่คล่อง แต่สามารถที่จะฟังและเข้าใจ

2.10.4 flutter_local_notifications

เป็นปลั๊กอินของ Flutter ที่ใช้สำหรับ การแสดงผลการแจ้งเตือน (Local Notifications) บนอุปกรณ์ต่าง ๆ โดยไม่ต้องมีการเชื่อมต่ออินเทอร์เน็ต เหมาะสำหรับการแจ้งเตือนที่เกิดขึ้นภายในเครื่อง และสามารถนำมาใช้เพื่อสร้างการแจ้งเตือนที่ซับซ้อน

คุณสมบัติหลัก

- แสดงการแจ้งเตือนพื้นฐาน (basic notifications)
- การตั้งเวลาแจ้งเตือน (Scheduled Notification) เช่น แจ้งเตือนรายวัน รายสัปดาห์ หรือแจ้งเตือนซ้ำเป็นช่วงเวลา

google_mlkit_text_recognition

2.10.5 google_mlkit_text_recognition

เหมาะสำหรับแอปที่ต้องการ สแกนและแปลงข้อความจากรูปภาพอย่างแม่นยำ, รองรับหลายภาษา, และ ทำงานออฟไลน์

ข้อดี

- ใช้โมเดล Machine Learning ของ Google ทำให้การอ่านตัวอักษรจากภาพแม่นยำ แม้อาจไม่ชัดเจนหรือมีเงา
- มี API ที่เข้าใจง่าย สามารถรวมเข้ากับ Flutter ได้สะดวก
- ทำงานแบบ On-device ไม่ต้องส่งรูปภาพขึ้นไปประมวลผลบนเซิร์ฟเวอร์ ทำให้มีความรวดเร็ว, ประหยัดข้อมูล, และปลอดภัยต่อข้อมูลส่วนตัว
- สามารถสแกนข้อความจากกล้องได้ทันที (Live Camera OCR)

คุณสมบัติเด่น

- สามารถแปลงรูปภาพให้เป็นข้อความโดยไม่ต้องส่งข้อมูลไปยังเซิร์ฟเวอร์
- สามารถแบ่งข้อความออกเป็นบล็อก, บรรทัด, หรือองค์ประกอบตัวอักษรย่อย ๆ เพื่อการประมวลผลที่ละเอียด
- 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

- สแกนบาร์โค้ดได้เร็ว รองรับการตรวจจับแบบ real-time
- รองรับหลายรูปแบบของบาร์โค้ด เช่น QR Code, Code128, EAN-13 เป็นต้น
- ปรับแต่งกล้องและตัวสแกนได้ เช่น ความละเอียดของกล้อง, ความเร็วการตรวจจับ, เปิด/ปิดแฟลช, การกลับภาพ, การซูมอัตโนมัติ (Flutter คืออะไร โดย AWS, 2567)

2.11 Dart

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

2.11.1 คุณสมบัติเด่นของ Dart

- Type Safety (ระบบความปลอดภัยของชนิดข้อมูล)
Dart ใช้ Static Type Checking เพื่อตรวจสอบความถูกต้องของชนิดข้อมูลตั้งแต่ตอนคอมไพล์ ทำให้ลดข้อผิดพลาดขณะรันโปรแกรม
- Null Safety (ระบบป้องกันค่า Null)
Dart มีระบบป้องกัน Null ที่บังคับใช้ได้อย่างเคร่งครัด ตัวแปรที่ไม่ได้ระบุว่าเป็น Null ได้ จะไม่สามารถเก็บค่า Null ได้เลย ป้องกัน Null Reference Error ซึ่งเป็นปัญหาคลาสสิกในภาษาอื่น ๆ
- รองรับการเขียนโปรแกรมแบบ Asynchronous
โดยมีคำสำคัญ เช่น async, await, และชนิดข้อมูล Future รวมถึง Stream ที่ช่วยให้นักพัฒนาจัดการงานแบบไม่ประสานเวลา (เช่น ดึงข้อมูลจาก API หรือทำงาน I/O) ได้อย่างสะดวก (Dart overview โดย Dart, 2568)

2.12 คำศัพท์

- แอปพลิเคชัน (Application): โปรแกรมคอมพิวเตอร์ที่ถูกสร้างขึ้นมาเพื่อทำงานบางอย่างบนอุปกรณ์
- เฟรมเวิร์ก (Framework): ชุดเครื่องมือและโครงสร้างสำเร็จรูปที่ช่วยให้นักพัฒนาสามารถสร้างโปรแกรมได้รวดเร็วและเป็นระบบมากขึ้น
- Codebase: ชุดไฟล์โค้ดทั้งหมดที่ใช้ในการสร้างโปรแกรมหนึ่ง ๆ
- Native: การพัฒนาแอปพลิเคชันที่เขียนโค้ดแยกกันเฉพาะสำหรับระบบปฏิบัติการนั้น ๆ โดยเฉพาะ
- Cross-platform: การพัฒนาแอปพลิเคชันที่สามารถใช้ได้ชุดเดียวกันบนระบบปฏิบัติการที่แตกต่างกันได้
- Backend: ส่วนการทำงานเบื้องหลังของแอปพลิเคชัน ทำหน้าที่จัดการข้อมูล, ประมวลผลคำสั่ง, และสื่อสารกับฐานข้อมูล
- Frontend: ส่วนที่ผู้ใช้มองเห็นและใช้งานโดยตรง เช่น หน้าจอแอปพลิเคชัน, ปุ่มกด, หรือรูปภาพต่าง ๆ
- API (Application Programming Interface): ตัวกลางที่ช่วยให้โปรแกรมหรือระบบที่แตกต่างกันสามารถสื่อสารและแลกเปลี่ยนข้อมูลกันได้
- RESTful API: รูปแบบมาตรฐานในการออกแบบ API ที่ใช้สื่อสารผ่านอินเทอร์เน็ต โดยมีคำสั่งหลักๆ เช่น GET (ขอข้อมูล), POST (เพิ่มข้อมูล), PUT (แก้ไขข้อมูล) และ DELETE (ลบข้อมูล)
- Notification: การแจ้งเตือนที่ปรากฏบนหน้าจออุปกรณ์เพื่อบอกให้ผู้ใช้ทราบถึงข้อมูลหรือกิจกรรมบางอย่าง
- Text-to-Speech (TTS): เทคโนโลยีที่แปลงข้อความตัวอักษรให้เป็นเสียงพูด
- การรู้จำอักขระด้วยแสง (Optical Character Recognition - OCR): เทคโนโลยีที่ใช้ในการแปลงข้อความจากรูปภาพให้กลายเป็นข้อความที่คอมพิวเตอร์สามารถเข้าใจและแก้ไขได้

บทที่ 3

ผลงานที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงงานวิจัยและโครงการที่เกี่ยวข้องกับการพัฒนาแอปพลิเคชันสำหรับช่วยเหลือผู้สูงอายุในการจัดการการใช้ยา โดยจะทำการวิเคราะห์และเปรียบเทียบแนวทางการพัฒนา ข้อดี และข้อจำกัดของแต่ละงานวิจัย เพื่อนำมาเป็นแนวทางในการพัฒนาโครงการนี้ให้มีประสิทธิภาพและตอบสนองความต้องการของผู้ใช้งานได้อย่างเหมาะสม

3.1 แอปพลิเคชันช่วยอ่านฉลากยาโดยเฉพาะฟังก์ชันที่ช่วยให้อ่านข้อมูลยาได้ง่ายขึ้นอย่างชัดเจน

ผู้วิจัย: ณัฐกรณ์ ศรีบุรมย์ (2567)

ที่มา: ผู้สูงอายุที่มีอายุมากกว่า 60 ปีขึ้นไป มีปัญหาด้านสายตา ความจำ และไม่คุ้นเคยกับเทคโนโลยี ซึ่งส่งผลให้เกิดความยากลำบากในการอ่านฉลากยาและปฏิบัติตามคำแนะนำในการใช้ยาได้อย่างถูกต้อง

วิธีการ: พัฒนาแอปพลิเคชันที่เป็นตัวช่วยในการอ่านฉลากยา โดยใช้เทคโนโลยีการรู้จำอักขระด้วยแสง (OCR) เพื่อแปลงข้อความบนฉลากยาให้เป็นข้อมูลดิจิทัล และนำเสนอข้อมูลในรูปแบบที่เข้าใจง่าย มีตัวอักษรขนาดใหญ่ เพื่อส่งเสริมพฤติกรรมการใช้ยาที่ถูกต้อง

ผลลัพธ์: จากการทดสอบกับกลุ่มอาสาสมัคร พบว่าอาสาสมัครสามารถตอบคำถามเกี่ยวกับวิธีการรับประทานยา จำนวนครั้งต่อวัน จำนวนเม็ด และต้องใช้อย่างก่อนหรือหลังมีอาหารหรือเวลามีอาการ ได้ถูกต้องทั้งหมดทุกข้อและในทุกฉลาก แสดงให้เห็นว่าแอปพลิเคชันสามารถช่วยให้ผู้สูงอายุเข้าใจข้อมูลยาได้ดีขึ้น

ข้อดี:

- เป็นแอปพลิเคชันที่ตอบโจทย์ผู้สูงอายุได้อย่างตรงจุด
- ช่วยลดความผิดพลาดในการใช้ยาอย่างมีนัยสำคัญ
- ใช้งานง่าย เหมาะสมกับผู้สูงอายุที่ไม่คุ้นเคยกับเทคโนโลยี

ข้อจำกัด:

- การสแกนข้อมูลบางครั้งมีความหน่วง (delay) เพื่อให้การอ่านฉลากยาในครั้งก่อนหน้าเสร็จเรียบร้อยก่อนดำเนินการสแกนครั้งต่อไป
- อาจมีข้อจำกัดในการรองรับฉลากยาที่มีรูปแบบแตกต่างกัน

3.2 แอปพลิเคชันบน LINE LIFF ร่วมกับ Google Apps Script สำหรับติดตามและประเมินผลความสม่ำเสมอในการรับประทานยาของผู้ป่วยโรค โรงพยาบาลชัยนาทนเรนทร

ผู้วิจัย: วิศว์สดี ปาริยะประเสริฐ (2567)

ที่มา: ปัญหาของผู้ป่วยที่ขาดความสม่ำเสมอในการทานยา โดยเฉพาะในกลุ่มโรคเรื้อรังเช่นวัณโรค และผู้สูงอายุที่มักจะลืมหรือสับสนเรื่องการทานยา ซึ่งส่งผลกระทบต่อประสิทธิภาพการรักษาและอาจทำให้เกิดภาวะแทรกซ้อนได้

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

- ส่งข้อความแจ้งเตือนผู้ป่วยเมื่อถึงเวลาทานยา
- ให้ผู้ป่วยส่งภาพถ่ายยืนยันการทานยาแต่ละครั้ง
- บันทึกตำแหน่งที่ผู้ใช้ทานยาได้
- จัดเก็บข้อมูลลงในฐานข้อมูลเพื่อการติดตามผลและวิเคราะห์

ผลลัพธ์: หลังจากผู้ป่วยใช้แอปพลิเคชันเป็นเวลา 4 เดือน พบว่าค่าเฉลี่ยความสม่ำเสมอในการทานยาเพิ่มขึ้นจาก 90.66% เป็น 98.38% อย่างมีนัยสำคัญทางสถิติ ซึ่งแสดงให้เห็นว่าการแจ้งเตือนและการติดตามผลอย่างเป็นระบบช่วยให้ผู้ป่วยมีวินัยในการทานยามากขึ้น และสามารถลดโอกาสการล้มเหลวในการรักษาได้อย่างมีประสิทธิภาพ

ข้อดี:

- ใช้แพลตฟอร์ม LINE ซึ่งเป็นแอปพลิเคชันที่ผู้ป่วยคุ้นเคยและใช้งานอยู่แล้ว ทำให้เรียนรู้และเข้าถึงได้ง่าย
- ต้นทุนในการพัฒนาระบบไม่สูงมาก
- ผลลัพธ์ที่ได้มีความชัดเจนและวัดผลได้
- สามารถติดตามและวิเคราะห์พฤติกรรมการทานยาได้อย่างเป็นระบบ

ข้อเสนอแนะ:

- เนื่องจากจะมีการยุติการให้บริการ LINE Notify ตั้งแต่วันที่ 31 มีนาคม พ.ศ. 2568 เป็นต้นไป จึงจำเป็นต้องใช้โปรแกรมอื่นมาใช้ทดแทน LINE Notify เช่น Google Chat หรือ Telegram สำหรับการแจ้งเตือนต่อไป
- ควรมีการพัฒนาระบบให้รองรับการทำงานแบบ standalone มากขึ้น เพื่อลดการพึ่งพาแพลตฟอร์มเฉพาะ

3.3 โครงการวิจัยเรื่อง เทคโนโลยีสารสนเทศแสดงข้อมูลผลยาเอกสารกำกับยาแบบอัตโนมัติสำหรับ บริหารจัดการคลังยาปฏิชีวนะเพื่อความปลอดภัย

ผู้วิจัย: วิรุฬห์ ศรีบริรักษ์ (2562)

ที่มา: ผู้บริโภคขาดความรู้ความเข้าใจถึงอันตรายที่เกี่ยวกับยา อีกทั้งยังไม่มีแหล่งข้อมูลที่เข้าถึงได้ง่าย ข้อมูลบนฉลากยาจากคลินิกหรือโรงพยาบาลมักไม่เพียงพอต่อการทำความเข้าใจ นอกจากนี้ยังมีปัญหาในการบริหารจัดการคลังยาที่ต้องอาศัยการบันทึกข้อมูลด้วยมือซึ่งใช้เวลานานและเสี่ยงต่อความผิดพลาด

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

- ลดความคลาดเคลื่อนทางยา (Medication Error)
- ป้องกันความผิดพลาดในการจัดการคลังยา
- ให้ข้อมูลยาที่ถูกต้องและครบถ้วนแก่ผู้ใช้

ผลลัพธ์: ระบบที่พัฒนาขึ้นมีความแม่นยำในการแปลงข้อมูลสูงถึงร้อยละ 96.61 และสามารถช่วยลดเวลาของการเก็บข้อมูล
ลงฐานข้อมูลได้อย่างมีนัยสำคัญ เมื่อเทียบกับการบันทึกข้อมูลด้วยมือ

ข้อดี:

- ช่วยให้ผู้ใช้โรคมีความเข้าใจในการใช้ยามากขึ้น
- ช่วยให้การใช้ยามีความปลอดภัยและลดปัญหาของการใช้ยาที่ผิดกับโรค
- ลดเวลาและแรงงานในการบันทึกข้อมูลยา
- มีความแม่นยำสูงในการอ่านข้อมูล

ข้อจำกัด:

- ยังคงต้องพัฒนาความยืดหยุ่นของระบบให้รองรับรูปแบบเอกสารที่แตกต่างกัน เช่น ฉลากยาจากโรงพยาบาลต่างๆ ที่อาจมีรูปแบบไม่เหมือนกัน
- ความแม่นยำอาจลดลงเมื่อฉลากยามีคุณภาพภาพต่ำหรือข้อมูลไม่ชัดเจน
- ต้องมีการปรับปรุงและฝึกฝนโมเดล OCR อย่างต่อเนื่องเพื่อรองรับฟอนต์และรูปแบบตัวอักษรที่หลากหลาย

3.4 การเปรียบเทียบแอปพลิเคชันที่มีอยู่ในตลาด

การวิเคราะห์แอปพลิเคชันที่อยู่ในตลาดช่วยให้เห็นภาพรวมของฟีเจอร์และแนวทางการออกแบบที่ได้รับความนิยม ซึ่งสามารถนำมาประยุกต์ใช้ในการพัฒนาแอปพลิเคชันของคุณได้

เราได้ทำการเปรียบเทียบฟีเจอร์หลักของแอปพลิเคชันยอดนิยม 5 ตัว ได้แก่ CapYaDoo, PharmaSee, MyYaAndYou, Medisafe และ RDU รู้เรื่องยา ดังตารางต่อไปนี้

3.4.1 ตารางเปรียบเทียบแอปพลิเคชันในตลาด

จากการเปรียบเทียบข้างต้น จะเห็นได้ว่า CapYaDoo มีข้อได้เปรียบเหนือ RDU รู้เรื่องยา ในหลายด้าน ได้แก่

ข้อได้เปรียบของ CapYaDoo

- ความหลากหลายในการสแกน: รองรับทั้งการสแกนบาร์โค้ดและตัวหนังสือ (OCR) ขณะที่ RDU รองรับเฉพาะ QR Code
- การจัดการข้อมูลยา: สามารถเพิ่มข้อมูลยาจากการสแกนได้ ขณะที่ RDU ไม่มีฟีเจอร์นี้

หมวดฟีเจอร์	CapYaDoo	PharmaSee	MyYaAndYou	Medisafe	RDU รู้เรื่องยา
ค้นหาข้อมูลยา	ค้นหา ด้วยข้อความ / สแกน / แปลภาษาได้	ค้นหา จาก ภาพยา (AI ตรวจรูป)	ค้นหาด้วยชื่อยา	ค้นหา ด้วยชื่อยาเท่านั้น	ค้นหา จาก ชื่อยา ในฐานข้อมูล ยย.
เสียง อ่าน ข้อมูล ยา (Text-to-Speech)	มีเสียงอ่านข้อมูลยาให้ผู้ใช้ฟัง	ไม่มี	ไม่มี	ไม่มี	ไม่มี
บันทึกอาการหลังทานยา	มี ระบบ บันทึกอาการ และ ผลข้างเคียง	ไม่มี	มีบันทึกยา แต่ไม่มีบันทึกอาการ	ไม่มี	ไม่มี
แจ้งเตือนการทานยา	ตั้งเวลาเตือนแยกตามยาได้	ไม่มี	มีระบบแจ้งเตือน	มีระบบแจ้งเตือนครบถ้วน	ไม่มีระบบเตือน
สแกนบาร์โค้ด / ตัวหนังสือ	สแกน ได้ ทั้ง "บาร์ โค้ด" และ "ข้อความ บนฉลากยา"	สแกน รูป ยา (AI Image Recognition)	ไม่มี สแกน บาร์โค้ด	ไม่มีสแกน	ไม่มีสแกน
แปลภาษาข้อมูลยา	แปล ข้อมูลยาได้ (ไทย ↔ อังกฤษ)	ไม่มี	ไม่มี	ไม่มี	ไม่มี
ตรวจสอบข้อมูลยา / ยย.	ตรวจ ผ่าน ฐานข้อมูล ยา ในประเทศ	ขึ้นกับฐานข้อมูลของ AI	อ้างอิง จาก ฐานข้อมูล ยย.	ข้อมูล ทั่วไป ไม่เฉพาะ ยย.	ใช้ฐานข้อมูลจาก ยย. โดยตรง
เหมาะกับ ผู้สูงอายุ / ใช้งานง่าย	UI เรียบง่าย / มีเสียง / แปลภาษา	ต้องถ่ายรูป อาจใช้งานยาก	UI ใช้งานง่าย	ใช้งานง่าย และระบบ แจ้งเตือนชัดเจน	ข้อความ เยอะ ต้องอ่านเอง
บันทึก ประวัติ ยา ในเครื่อง	มี ระบบ เก็บประวัติ ยา และอาการย้อนหลัง	เก็บ เฉพาะ ภาพยา	บันทึก ข้อมูล ยาได้	เก็บข้อมูลยาและเวลาเตือน	ไม่มีระบบบันทึก
เทคโนโลยี AI / OCR	ใช้ OCR อ่านฉลาก ยา และตรวจบาร์โค้ด	ใช้ AI แยกภาพยา	ไม่มี	ไม่มี	ไม่มี

ตาราง 3.1: เปรียบเทียบแอปพลิเคชันในตลาด

- การแปลภาษา: รองรับการแปลภาษาระหว่างไทย-อังกฤษ และมี Text-to-Speech ขณะที่ RDU ไม่มี
- ระบบแจ้งเตือน: มีระบบเตือนทานยาที่ครบถ้วน ขณะที่ RDU ไม่มีฟีเจอร์นี้
- การติดตามอาการ: สามารถบันทึกและติดตามอาการหลังทานยาได้ ขณะที่ RDU ไม่มีฟีเจอร์นี้

จุดเด่นที่เหมือนกัน

- การค้นหาข้อมูลยา: ทั้งสองแอปสามารถค้นหาและแสดงข้อมูลยาได้
- Text-to-Speech: ทั้งสองแอปรองรับการฟังเสียงจากข้อความ

3.4.2 สรุปการเปรียบเทียบ

จากการเปรียบเทียบจะเห็นได้ว่า CapYaDoo มีความครบถ้วนและครอบคลุมฟีเจอร์มากกว่า RDU รู้เรื่องยา โดยเฉพาะในด้านการจัดการข้อมูลยา การแจ้งเตือน และการติดตามอาการ ซึ่งเป็นฟีเจอร์ที่สำคัญสำหรับผู้สูงอายุในการจัดการการใช้ยา

อย่างไรก็ตาม RDU รู้เรื่องยากก็มีจุดเด่นในด้านความเรียบง่ายและเน้นการค้นหาข้อมูลยาเป็นหลัก ซึ่งอาจเหมาะกับผู้ใช้ที่ต้องการแค่ดูข้อมูลยาเท่านั้น

3.4.3 การวิเคราะห์ช่องว่างในตลาด

จากตารางเปรียบเทียบจะเห็นว่า แอปพลิเคชันในตลาดส่วนใหญ่เน้นฟังก์ชันพื้นฐานอย่างการแจ้งเตือนและการสแกน Barcode แต่ยังขาดฟังก์ชันสำคัญ เช่น

- การสแกนตัวอักษร (OCR) สำหรับยาที่ไม่มี Barcode
- การเพิ่มยาที่กำลังทาน และติดตามประวัติการใช้ยา
- การติดตามอาการ หลังจากการทานยา
- การแปลภาษา และ Text-to-Speech สำหรับผู้สูงอายุ
- การ Export รายงาน ประวัติยาและอาการ

เนื่องจากฟังก์ชันเหล่านี้ไม่ได้รวมอยู่ในแอปพลิเคชันเดียว แอปของเราจึงรวบรวมฟังก์ชันพื้นฐานและเพิ่มเติมฟังก์ชันที่จำเป็นเพื่อความสะดวกในการใช้งาน โดยเฉพาะการตอบสนองความต้องการของผู้สูงอายุที่มีข้อจำกัดทางด้านการมองเห็นและการได้ยิน

3.5 ประสบการณ์การพัฒนาที่ผ่านมา

เพื่อให้เห็นถึงประสบการณ์และความสามารถของทีมผู้พัฒนาในการสร้างแอปพลิเคชัน จึงได้นำเสนอโครงการที่ผ่านมาที่เกี่ยวข้องกับการพัฒนาแอปพลิเคชันด้วย Flutter Framework

3.5.1 Application Chat Music Player

วัตถุประสงค์:

1. เพื่อให้ผู้ใช้งานเกิดความผ่อนคลายและสร้างความสุขระหว่างการใช้งานให้กับผู้ใช้ได้
2. เพื่อให้ผู้ใช้งานได้พูดคุยและแลกเปลี่ยนความคิดเห็นกับผู้ใช้คนอื่น ๆ ได้

บทบาทและสิ่งที่ทำ:

- นาย ฤทธิช พลราช: มีหน้าที่ในการออกแบบ UX/UI และทำการพัฒนาแอปพลิเคชัน
- นางสาว จิรัชญา ราชพลแสน: มีหน้าที่ในการจัดทำรายงานและพัฒนาแอปพลิเคชัน

เทคโนโลยีที่ใช้:

- ใช้ภาษา Dart และ Flutter framework

- Backend ใช้ Firebase Firestore, Firebase Authentication
- Package ที่ใช้คือ audioplayer

3.5.2 Application Location Discovery

วัตถุประสงค์หลัก:

1. เพื่อให้ผู้ใช้งานเลือก/ไม่เลือก สถานที่พักผ่อน อ่านหนังสือภายในมหาวิทยาลัย
2. เพื่อให้ผู้ใช้งานใช้ข้อมูลในการตัดสินใจเลือก เช่น สถานที่ที่อยู่ใกล้เรา สิ่งอำนวยความสะดวกในสถานที่นั้น

บทบาทและสิ่งที่ทำ:

- นางสาว ปรรณนา สุภาวงศ์: ทำหน้าที่ในการออกแบบแอปพลิเคชัน Logo, UI และพัฒนาแอปพลิเคชัน

เทคโนโลยีที่ใช้:

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

3.5.3 บทเรียนที่ได้จากประสบการณ์การพัฒนา

จากการพัฒนาแอปพลิเคชันทั้งสองโครงการ ทีมผู้พัฒนาได้เรียนรู้และสะสมประสบการณ์ในด้านต่างๆ ดังนี้

ด้านการออกแบบ UI/UX

- การออกแบบส่วนติดต่อผู้ใช้ที่ใช้งานง่ายและสวยงาม
- การเลือกใช้สีและฟอนต์ที่เหมาะสมกับกลุ่มผู้ใช้
- การจัดวางองค์ประกอบบนหน้าจอให้เป็นระเบียบและเข้าถึงได้ง่าย

ด้านการพัฒนาแอปพลิเคชัน

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

ด้านการจัดการฐานข้อมูล

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

ด้านการทำงานเป็นทีม

- การแบ่งหน้าที่และความรับผิดชอบในทีม
- การสื่อสารและประสานงานระหว่างสมาชิกในทีม
- การจัดการเวลาและกำหนดการในการพัฒนา

ประสบการณ์เหล่านี้จะเป็นประโยชน์อย่างมากในการพัฒนาโครงงาน CapYaDoo เนื่องจากทีมผู้พัฒนาได้มีความคุ้นเคยกับ Flutter Framework และการทำงานร่วมกันมาแล้ว

3.6 สรุปและการนำไปประยุกต์ใช้

จากการศึกษางานวิจัยและโครงการที่เกี่ยวข้องทั้ง 3 งาน รวมถึงการเปรียบเทียบแอปพลิเคชันที่มีอยู่ในตลาด สามารถสรุปประเด็นสำคัญที่จะนำมาประยุกต์ใช้ในโครงงานนี้ได้ดังนี้

3.6.1 แนวทางการพัฒนาที่ได้จากงานวิจัย

- การออกแบบที่เหมาะสมกับผู้สูงอายุ จากงานวิจัยที่ 1 แสดงให้เห็นถึงความสำคัญของการออกแบบ UI/UX ที่เรียบง่าย ใช้ตัวอักษรขนาดใหญ่ และมีความชัดเจน
- ระบบแจ้งเตือนและติดตามผล จากงานวิจัยที่ 2 แสดงให้เห็นว่าการมีระบบแจ้งเตือนและติดตามผลอย่างเป็นระบบสามารถเพิ่มความสม่ำเสมอในการทานยาได้อย่างมีนัยสำคัญ
- เทคโนโลยี OCR จากงานวิจัยที่ 3 แสดงให้เห็นถึงประสิทธิภาพของ OCR ในการอ่านข้อมูลยา แต่ต้องคำนึงถึงความยืดหยุ่นในการรองรับรูปแบบฉลากที่แตกต่างกัน

3.6.2 ข้อจำกัดที่ต้องหลีกเลี่ยง

- หลีกเลี่ยงการพึ่งพาแพลตฟอร์มเฉพาะที่อาจมีการเปลี่ยนแปลงหรือยุติบริการ
- ต้องพัฒนาระบบ OCR ให้มีความแม่นยำและรองรับรูปแบบฉลากที่หลากหลาย
- ต้องแก้ไขปัญหา delay ในการประมวลผลให้มีประสิทธิภาพมากขึ้น

3.6.3 คุณสมบัติที่จะนำมาพัฒนาในโครงงาน

โครงงานนี้จะนำแนวคิดและข้อดีจากงานวิจัยที่เกี่ยวข้องมาประยุกต์ใช้ โดยมีจุดเด่นดังนี้

- ใช้เทคโนโลยี OCR ในการอ่านฉลากยาพร้อมทั้งฐานข้อมูลยาจาก Thai Medicine Terminology (TMT) ที่มีความน่าเชื่อถือ
- พัฒนาระบบแจ้งเตือนที่หลากหลาย เช่น เสียง, การสั่น เพื่อรองรับผู้สูงอายุที่มีข้อจำกัดทางการได้ยินและการมองเห็น

- สร้างระบบบันทึกและติดตามประวัติการทานยาที่ครบถ้วน
- ออกแบบ UI/UX ที่เหมาะสมกับผู้สูงอายุโดยเฉพาะ

บทที่ 4

ระเบียบวิธีดำเนินโครงการ

4.1 การวิเคราะห์และออกแบบระบบ

4.1.1 Persona

หมายถึง ตัวตนสมมติที่เป็นตัวแทนกลุ่มลูกค้าเป้าหมายของธุรกิจ โดยอิงจากข้อมูลการวิจัยและวิเคราะห์พฤติกรรม ความต้องการ และแรงจูงใจของกลุ่มลูกค้าจริง เพื่อให้ธุรกิจเข้าใจลูกค้าได้อย่างลึกซึ้ง [9]

ความสำคัญของ Persona

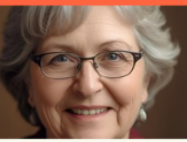
- ช่วยให้ธุรกิจเข้าใจลูกค้าได้อย่างลึกซึ้ง
- ปรับกลยุทธ์ทางการตลาดให้เหมาะสมกับพฤติกรรมของกลุ่มเป้าหมาย
- สื่อสารได้อย่างตรงใจลูกค้า

	คุณยามาลิ (อายุ 75 ปี) อาชีพ: แม่บ้าน ผู้สูงอายุที่ต้องการความช่วยเหลือ	Short description ผู้สูงอายุที่มีปัญหาด้านสุขภาพและการมองเห็น ไม่ถนัดการใช้เทคโนโลยี ทำให้ต้องพึ่งพาลูกหลานในการดูแลจัดการเรื่องยาและสุขภาพ
Key Attribute ต้องการการดูแลและความปลอดภัย: ให้ความสำคัญกับการมีคนช่วยจัดการสุขภาพและยาให้เป็นไปอย่างปลอดภัย		Challenges <ul style="list-style-type: none">• ไม่สามารถอ่านฉลากยาได้ด้วยตัวเอง• หลงลืมเรื่องการทานยาบ่อยครั้ง• ไม่คุ้นเคยและกลัวการใช้เทคโนโลยี
Needs <ul style="list-style-type: none">• ความมั่นใจว่าจะมีคนคอยช่วยจัดการยาให้ถูกต้องและปลอดภัย• วิธีการรับรู้ข้อมูลยาที่ง่ายและชัดเจน เช่น การฟังเสียง• การแจ้งเตือนที่ช่วยให้ไม่ลืมทานยา		Opportunities <ul style="list-style-type: none">• พัฒนาฟังก์ชันแปลง ข้อความเป็นเสียง (Text-to-Speech) สำหรับข้อมูลยา• ใช้การแจ้งเตือนด้วย เสียงที่ดังและชัดเจน

รูป 4.1: ตัวตนสมมติที่เป็นตัวแทนของกลุ่มผู้ใช้เป้าหมายคนที่ 1

	ชื่อ: คุณหมอชัย (อายุ 58 ปี) อาชีพ: แพทย์เกษียณ ผู้สูงอายุที่เน้นความเป็นอยู่ที่ดีและสุขภาพ	Short description ผู้สูงอายุที่เป็นแพทย์เกษียณ มีความรู้เรื่องสุขภาพเป็นอย่างดี แต่ยังคงต้องการเครื่องมือที่ช่วยให้การติดตามสุขภาพเป็นเรื่องง่ายและครบวงจร
Key Attribute ต้องการความรู้และการบูรณาการ: ให้ความสำคัญกับความถูกต้องของข้อมูลและการจัดการสุขภาพแบบครบวงจร		Challenges <ul style="list-style-type: none"> ข้อมูลในแอปพลิเคชันที่น่าเชื่อถือ หรือไม่เป็นไปตามหลักการแพทย์ แอปพลิเคชันที่ซับซ้อนและมีข้อมูลที่ไม่จำเป็นมากเกินไป ความกังวลว่าข้อมูลสุขภาพส่วนตัวจะไม่ปลอดภัย
Needs <ul style="list-style-type: none"> เครื่องมือที่ช่วยจัดระเบียบข้อมูลสุขภาพและยาได้อย่างแม่นยำ ฟังก์ชันที่สามารถบันทึกข้อมูลสุขภาพอื่น ๆ ได้ เช่น การออกกำลังกาย หรืออาหาร ข้อมูลที่ละเอียดและน่าเชื่อถือจากแหล่งอ้างอิงที่เชื่อถือได้ 		Opportunities <ul style="list-style-type: none"> แอปพลิเคชันที่เน้นการบันทึกข้อมูลสุขภาพได้ จัดทำฐานข้อมูลยาที่ละเอียดพร้อมแหล่งอ้างอิง

รูป 4.2: ตัวตนสมมติที่เป็นตัวแทนของกลุ่มผู้ใช้เป้าหมายคนที่ 2

	ชื่อ: คุณนารี (อายุ 58 ปี) อาชีพ: พนักงานบริษัท ผู้ป่วยเรื้อรัง (Chronic Patient)	Short description พนักงานบริษัทที่ป่วยเป็นโรคความดันโลหิตสูงและเบาหวาน ต้องการจัดการยาหลายชนิดอย่างเป็นระบบเพื่อให้สับสนและลืมทานยา
Key Attribute ต้องการความแม่นยำและเป็นระบบ: ต้องการเครื่องมือที่ช่วยจัดการชีวิตประจำวันให้ง่ายขึ้น		Challenges <ul style="list-style-type: none"> ลืมทานยาบ่อยครั้งเพราะต้องทานหลายเวลาในแต่ละวัน สับสนเรื่องชนิดของยาและเวลาที่ต้องทาน ไม่แน่ใจว่ายาตัวไหนมีผลข้างเคียงอย่างไร
Needs <ul style="list-style-type: none"> เครื่องมือที่ช่วยจัดการยาที่ต้องทานหลายชนิดในแต่ละวันอย่างเป็นระบบ ข้อมูลที่ถูกต้องเกี่ยวกับยาที่กำลังจะทาน เช่น ผลข้างเคียง หรือคำแนะนำพิเศษ ระบบแจ้งเตือนที่ช่วยให้ไม่ลืมทานยา 		Opportunities <ul style="list-style-type: none"> พัฒนาระบบแจ้งเตือนที่ใช้งานง่ายและปรับแต่งได้ จัดทำฐานข้อมูลที่ให้ข้อมูลครบถ้วนและเข้าใจง่าย ออกแบบหน้าจอให้แสดงตารางการทานยาแต่ละวันอย่างชัดเจน

รูป 4.3: ตัวตนสมมติที่เป็นตัวแทนของกลุ่มผู้ใช้เป้าหมายคนที่ 3

4.1.2 User Journey Map

หรือ แผนภาพเส้นทางผู้ใช้ เป็นเครื่องมือที่ใช้อธิบายขั้นตอนและประสบการณ์ของผู้ใช้เมื่อมีปฏิสัมพันธ์กับแอปพลิเคชันของเรา ว่าผู้ใช้ได้ตอบอย่างไร พบปัญหาตรงไหน และจะปรับปรุงจุดใดได้บ้าง เพื่อให้ผู้ใช้ได้รับประสบการณ์ที่ดีขึ้น ซึ่งจะแสดงภาพรวมตั้งแต่จุดเริ่มต้นที่ผู้ใช้เริ่มสนใจไปจนถึงการบรรลุเป้าหมายโดยแบ่งเป็นช่วงเวลาและเหตุการณ์สำคัญต่าง ๆ โดยมีองค์ประกอบดังนี้[10]

- The User: ผู้ใช้เป้าหมาย แรงจูงใจ และปัญหาที่พบ
- The Scenario Objective: สถานการณ์และวัตถุประสงค์เพื่ออธิบายว่าผู้กำลังพยายามทำอะไร
- Journey Phases: ช่วงเส้นทางของผู้ใช้ โดยจะแบ่งออกเป็น 5 ช่วงหลัก ได้แก่
 - Awareness: เริ่มรู้จักผลิตภัณฑ์
 - Consideration: พิจารณา
 - Decision: ตัดสินใจ
 - Purchase: ซื้อหรือใช้งาน
 - Retention: กลับมาใช้อีก
- Actions, Attitudes, and Emotions: พฤติกรรมและอารมณ์ของผู้ใช้ในแต่ละช่วง
- Opportunities: โอกาสในการปรับปรุง วิเคราะห์ข้อมูลที่ได้เพื่อหาทางปรับปรุง

4.1.3 ปัญหาและแนวทางการแก้ไข

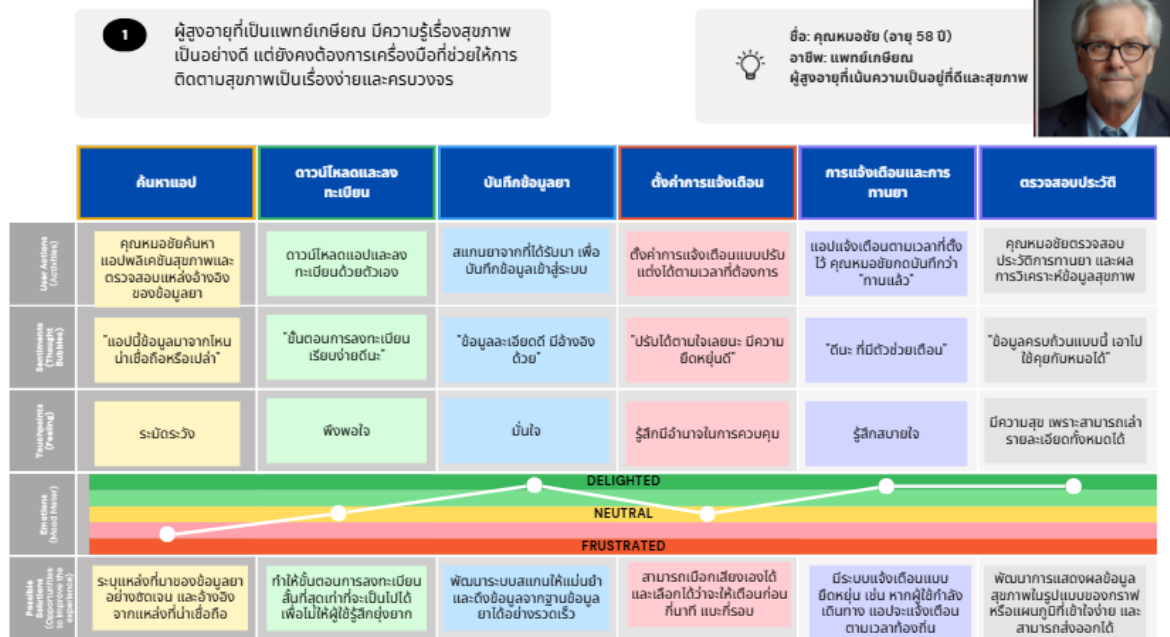
ผู้สูงอายุมักลืมทานยาหรือสับสนเรื่องชนิดและเวลาทานยา ปัญหานี้แก้ได้ด้วย:

- ระบบแจ้งเตือนที่ตั้งเวลาได้ตามแพทย์กำหนด
- ฐานข้อมูลยาที่ครบถ้วน เชื่อถือได้ และเข้าใจง่าย
- UI ที่ชัดเจนและใช้งานง่ายสำหรับผู้สูงอายุ
- ฟังก์ชัน Text-to-Speech และการแจ้งเตือนด้วยเสียง/สั่น/แสง
- ความปลอดภัยของข้อมูลสุขภาพและการเชื่อมต่อกับอุปกรณ์สุขภาพอื่น ๆ

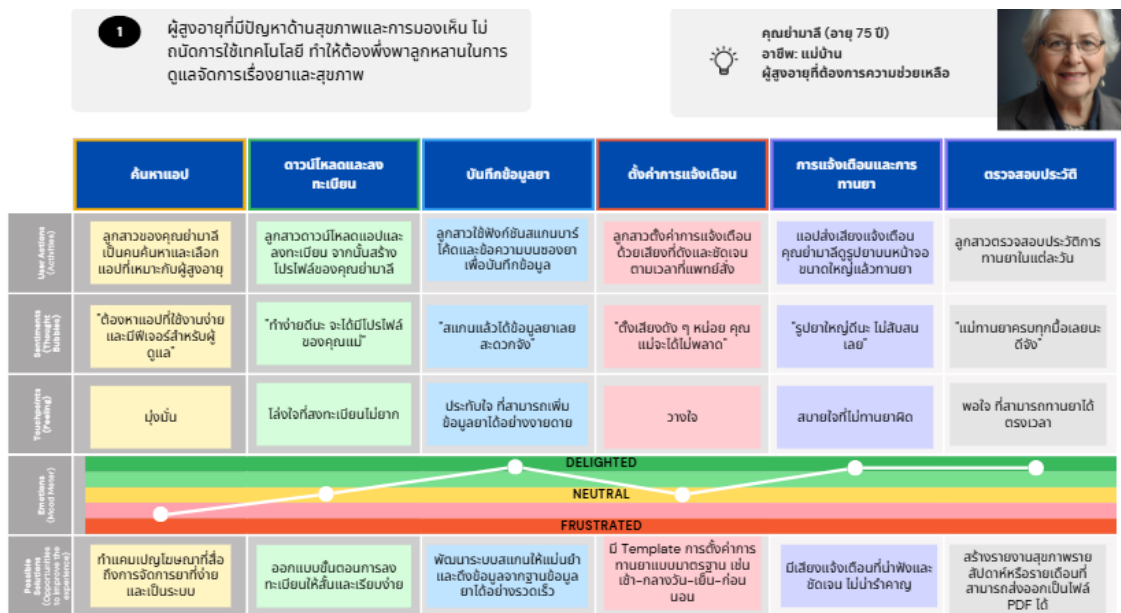
4.1.4 การวิเคราะห์ความต้องการ (Requirement Analysis)

ความต้องการของผู้ใช้ (User Requirements)

รวบรวมฟังก์ชันที่ผู้ใช้ต้องการ ได้แก่



รูป 4.4: ประสบการณ์ของผู้ใช้เมื่อมีปฏิสัมพันธ์กับแอปพลิเคชันของเราคนที่ 1



รูป 4.5: ประสบการณ์ของผู้ใช้เมื่อมีปฏิสัมพันธ์กับแอปพลิเคชันของเราคนที่ 2

- ระบบจัดการข้อมูลยา (เพิ่ม/แก้ไข/ลบ)
- ระบบแจ้งเตือนการทานยาที่ตั้งเวลาได้
- ระบบการสแกนข้อมูลยาจากฉลาก (Barcode & OCR)
- ระบบบันทึกประวัติการทานยาและอาการ



รูป 4.6: ประสบการณ์ของผู้ใช้เมื่อมีปฏิสัมพันธ์กับแอปพลิเคชันของเราคนที่ 3

- ระบบอำนวยความสะดวกสำหรับผู้สูงอายุ (Text-to-Speech)

ความต้องการของระบบ (System Requirements)

กำหนดคุณสมบัติที่ระบบต้องมี ได้แก่

- สามารถทำงานได้บน Android
- มีความเร็วและความแม่นยำในการสแกนข้อมูล
- มีการจัดเก็บข้อมูลผู้ใช้และข้อมูลยาอย่างปลอดภัย

4.1.5 การออกแบบสถาปัตยกรรม (System Architecture Design)

Source

ข้อมูลต้นทางมาจาก สำนักพัฒนามาตรฐานระบบข้อมูลสุขภาพไทย [11] ซึ่งเผยแพร่ข้อมูลในรูปแบบไฟล์ CSV ข้อมูลดังกล่าวจะถูกนำมาใช้เป็นแหล่งข้อมูลหลักของระบบ

Data Processing

ข้อมูลจากไฟล์ CSV จะถูกนำเข้าสู่กระบวนการประมวลผลด้วย PySpark [12] ซึ่งเป็นเครื่องมือที่เหมาะสมสำหรับการจัดการและประมวลผลข้อมูลขนาดใหญ่ (Big Data Processing) เพื่อจัดรูปแบบข้อมูลให้อยู่ในโครงสร้างที่เหมาะสมต่อการจัดเก็บและการนำไปใช้งานต่อไป

Database

หลังจากผ่านการประมวลผลแล้ว ข้อมูลจะถูกโหลตเข้าสู่ฐานข้อมูล PostgreSQL เพื่อจัดเก็บอย่างเป็นระบบและสามารถเข้าถึงได้อย่างมีประสิทธิภาพ

Backend Service

ใช้ Spring Boot Framework ในการพัฒนาเป็นบริการฝั่งเซิร์ฟเวอร์ (Backend Service) ทำหน้าที่เป็นตัวกลางระหว่างฐานข้อมูลและฝั่งผู้ใช้งาน โดยจัดการคำขอ (Request) จากฝั่ง Client รวมถึงประมวลผลและส่งข้อมูลกลับไปยังผู้ใช้งานในรูปแบบ API

Client Application

ส่วนของผู้ใช้งาน (Frontend) ถูกพัฒนาด้วย Dart และ Flutter ซึ่งเป็นเครื่องมือสำหรับพัฒนาแอปพลิเคชันแบบ Cross-platform เพื่อให้สามารถใช้งานได้ทั้งบนระบบปฏิบัติการ Android, iOS และ Web

System Deployment (kind Cluster)

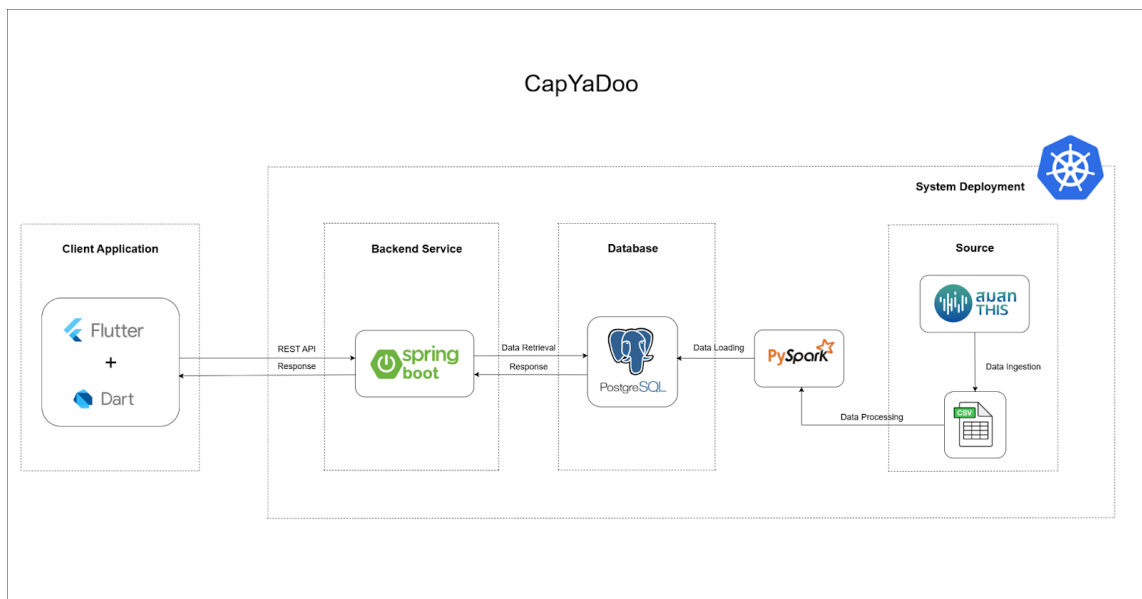
สำหรับการทดสอบและจำลองสภาพแวดล้อมการทำงานของระบบบน Kubernetes ได้มีการใช้เครื่องมือ kind [13] (Kubernetes IN Docker) เพื่อสร้าง Kubernetes cluster จำลองบนเครื่องพัฒนา (Local Environment)

โดย cluster ที่สร้างขึ้นจะประกอบด้วย container หลัก ได้แก่

- Spring Boot Application Container
- PostgreSQL Database Container

ซึ่งเชื่อมต่อกันผ่าน Kubernetes service ภายใน cluster

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



รูป 4.7: System Architecture Design

4.1.6 การออกแบบฐานข้อมูล (Database Design)

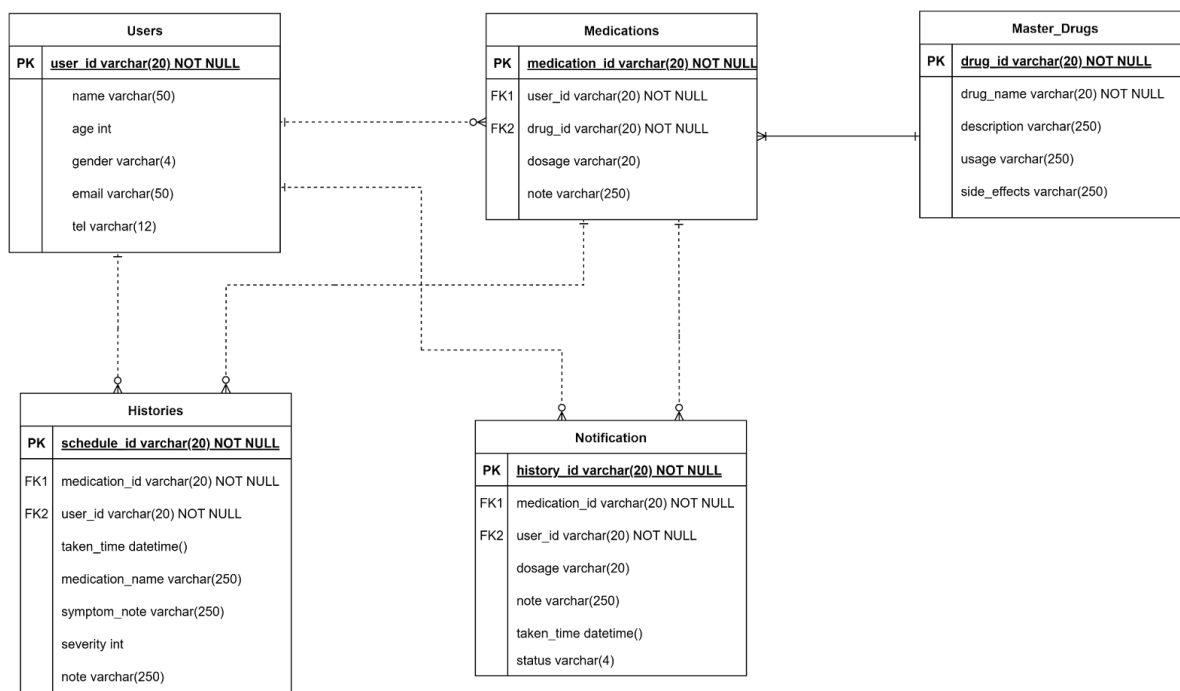
ฐานข้อมูลถูกออกแบบบน PostgreSQL ซึ่งเป็นฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ที่มีความน่าเชื่อถือและเหมาะสมกับการจัดการข้อมูลที่มีโครงสร้างชัดเจน โครงสร้างหลัก (Tables) ที่ออกแบบไว้ประกอบด้วย:

- users: จัดเก็บข้อมูลของผู้ใช้
- master_drugs: ฐานข้อมูลยาหลัก
- medications: จัดเก็บข้อมูลยาที่ผู้ใช้ทำการบันทึกไว้
- schedules: จัดเก็บตารางเวลาการแจ้งเตือนการทานยา
- histories: จัดเก็บประวัติการทานยาและอาการของผู้ใช้แต่ละคน

โดยแต่ละตารางจะมีการกำหนด Primary Key และ Foreign Key เพื่อเชื่อมโยงข้อมูลเข้าด้วยกันอย่างเป็นระบบและป้องกันข้อมูลซ้ำซ้อน

4.1.7 ขั้นตอนการออกแบบระบบ

Entity Relationship Diagram (ERD)



รูป 4.8: Entity Relationship Diagram (ERD)

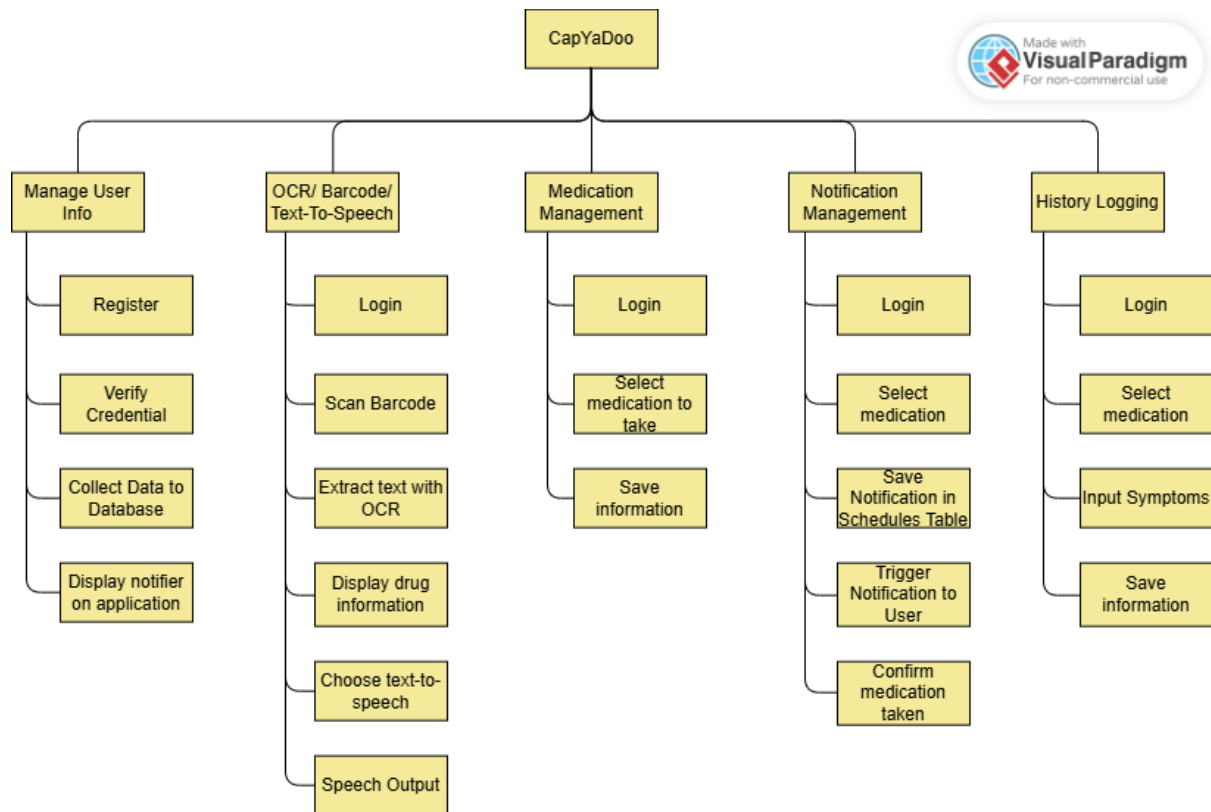
อธิบายโครงสร้างระหว่างฐานข้อมูลในรูปแบบแผนภาพ โดยในโครงงานนี้ จะใช้ในแนวคิด ERD แบบ Crow's Foot Notation ซึ่งจะประกอบไปด้วย [14]

- Entity: สิ่งของ บุคคล เหตุการณ์ หรือแนวคิดที่สามารถระบุได้ชัดเจน
- Relationship: ความสัมพันธ์ระหว่างเอนทิตี
- Attributes: คุณลักษณะของเอนทิตีหรือความสัมพันธ์
- Primary Key: ใช้เพื่อระบุเอนทิตีอย่างไม่ซ้ำกัน
- Cardinality: จำนวนขั้นต่ำและสูงสุดของความสัมพันธ์ระหว่างเอนทิตี เช่น คุณครู 1 คน มีสอนได้หลายวิชา

สัญลักษณ์

- □ (วงกลม): Zero ซึ่งสามารถไม่มีค่าได้
- < (ขาแฉกสามง่าม): Many หมายถึงมีได้หลายรายการ
- □<: Zero or Many หมายถึงอาจไม่มีค่าได้หรือมีได้หลายรายการ
- 1 (ขาแฉกเดียว): One หมายถึงมีได้หนึ่งรายการเท่านั้น
- □1: Zero or One หมายถึงอาจไม่มีค่าได้หรือมีได้หนึ่งรายการเท่านั้น

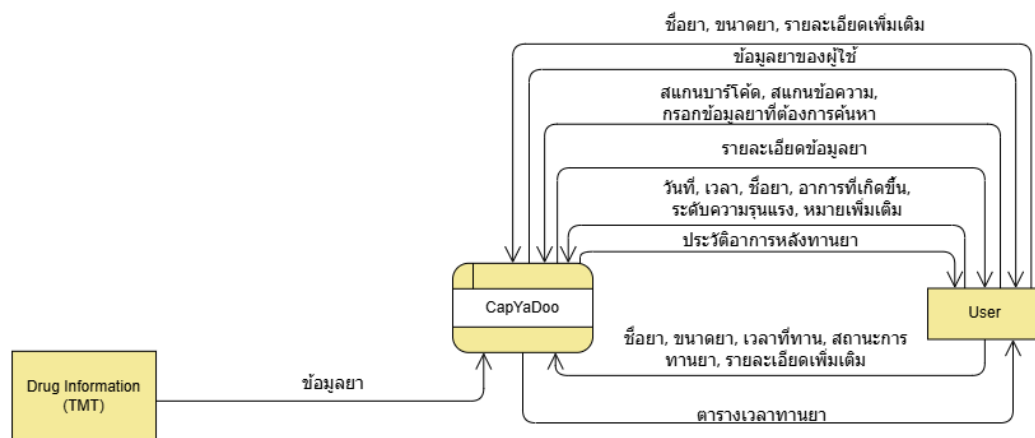
Functional Decomposition



รูป 4.9: Functional Decomposition

ขั้นตอนการแบ่งระบบออกเป็นส่วนย่อย ๆ เพื่อให้ง่ายต่อการทำความเข้าใจ การวิเคราะห์ การพัฒนา และการนำไปใช้งาน

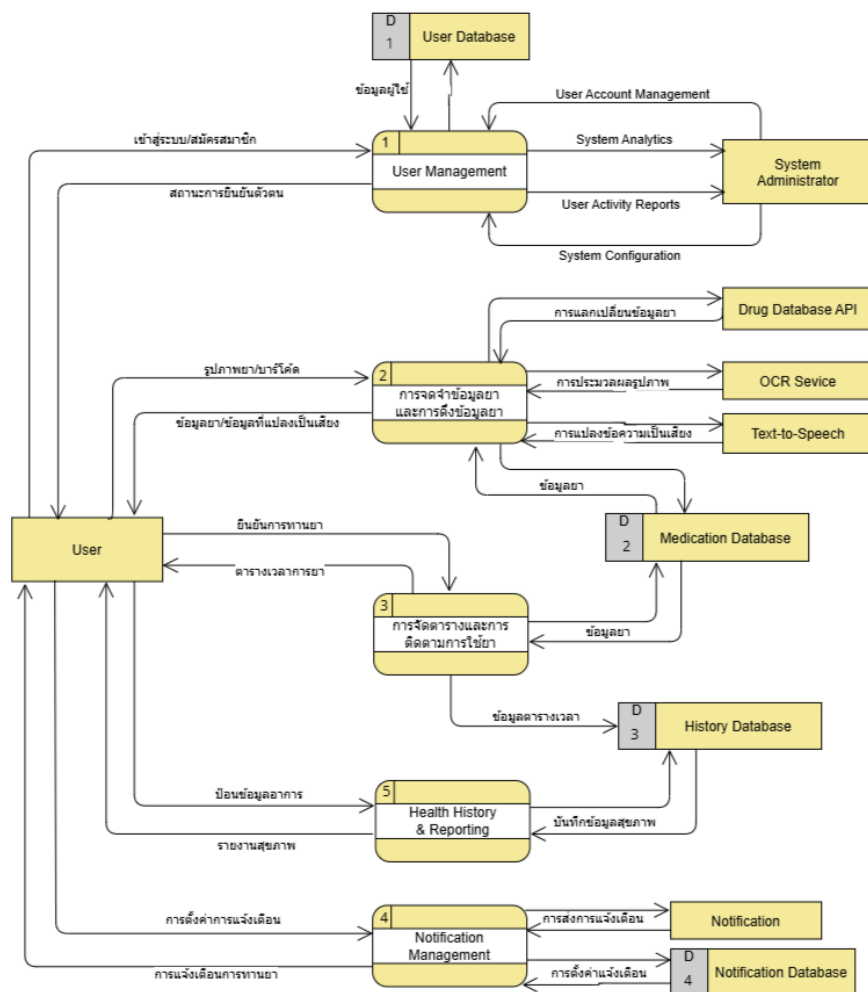
Data Flow Diagram Level 0



รูป 4.10: Data Flow Diagram Level 0

หรือ Context Diagram ใช้แสดงภาพรวมของระบบทั้งหมด

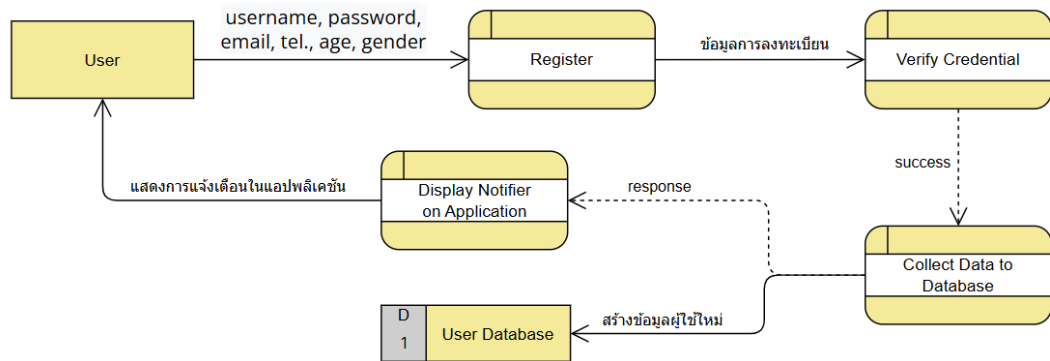
Data Flow Diagram Level 1



รูป 4.11: Data Flow Diagram Level 1

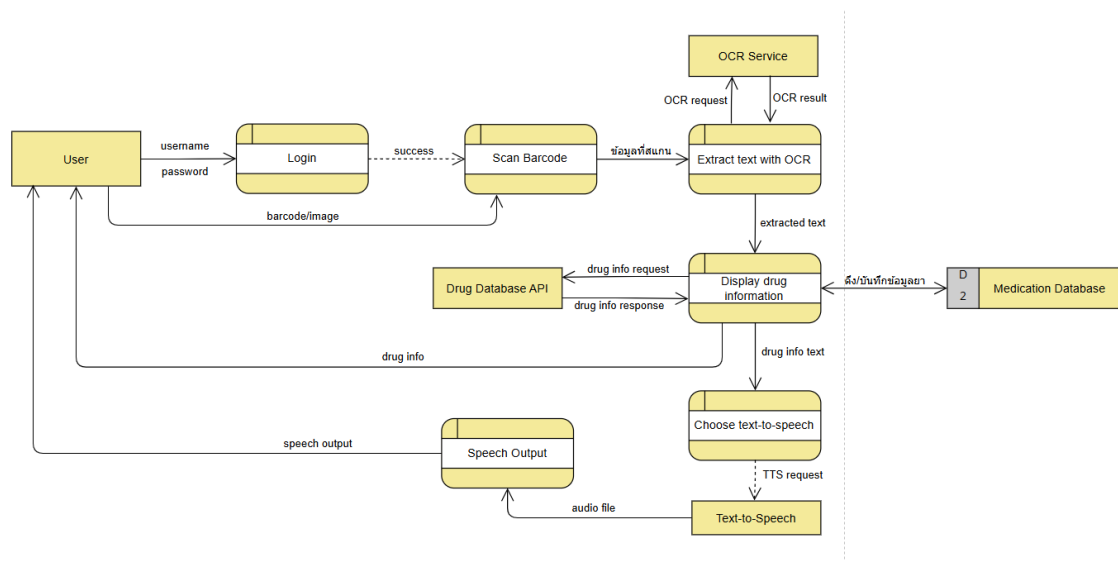
ใช้แสดงกระบวนการย่อย ๆ ภายในระบบ

Data Flow Diagram Level 2



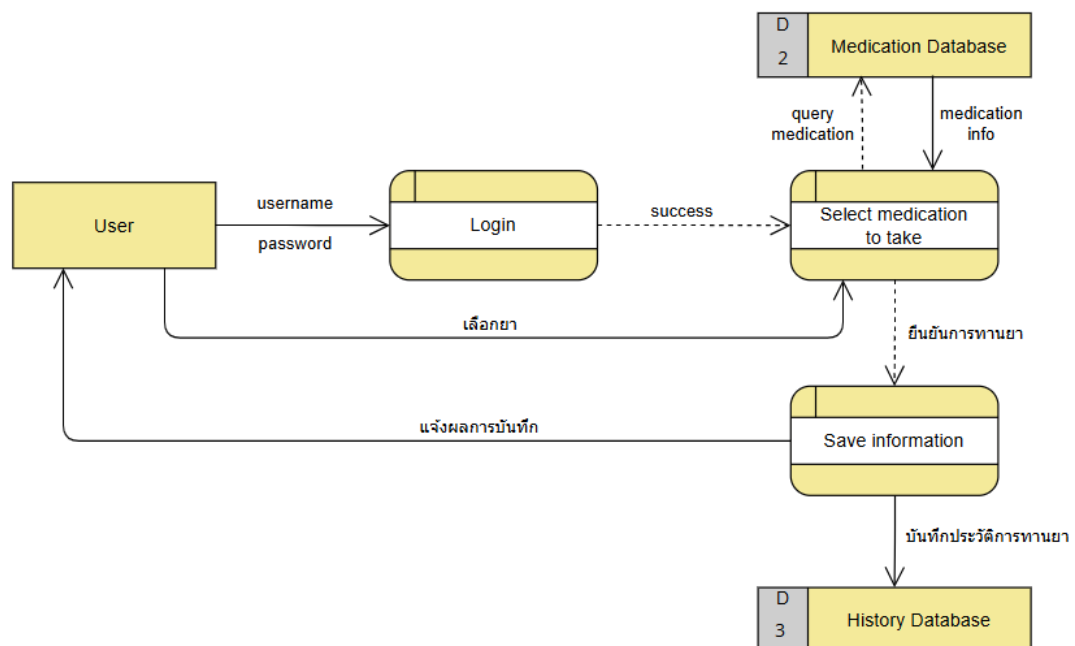
รูป 4.12: Data Flow Diagram Level 2.1 - Manage User Information

Manage User Information ใช้ในการขยายข้อมูลจาก Data Flow Diagram ระดับ 1 ให้ละเอียดมากยิ่งขึ้น



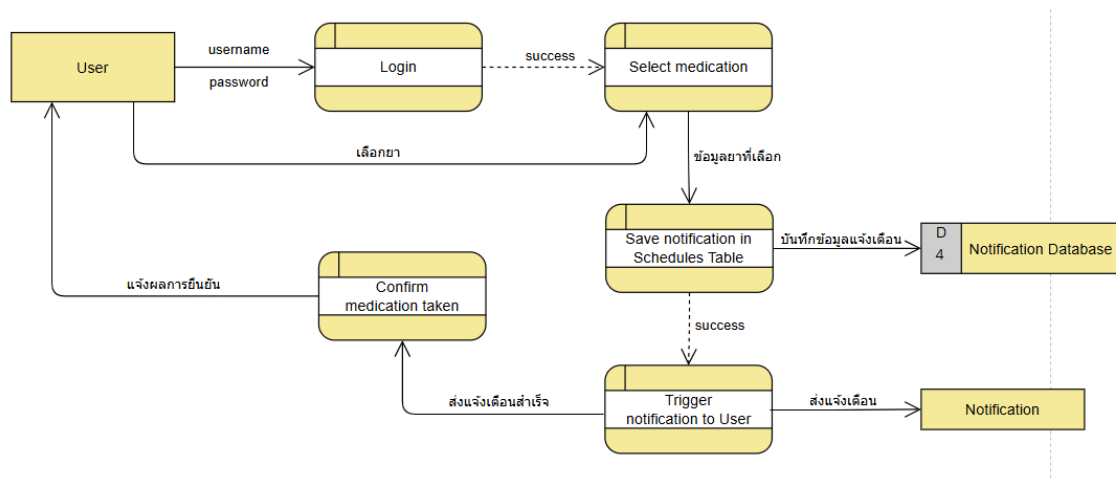
รูป 4.13: Data Flow Diagram Level 2.2 - OCR/ Barcode/ Text-To-Speech

OCR/ Barcode/ Text-To-Speech



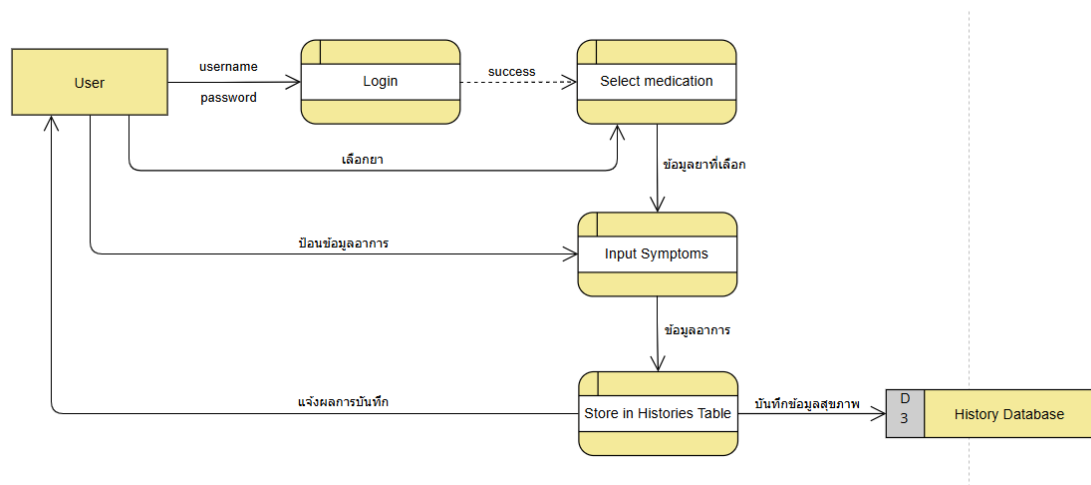
รูป 4.14: Data Flow Diagram Level 2.3 - Medication Management

Medication Management



รูป 4.15: Data Flow Diagram Level 2.4 - Notification Management

Notification Management



รูป 4.16: Data Flow Diagram Level 2.5 - History Logging

History Logging

4.2 การพัฒนาและติดตั้ง (Development and Implementation)

ขั้นตอนนี้จะเปลี่ยนการออกแบบให้กลายเป็นแอปพลิเคชันที่สามารถใช้งานได้จริง

4.2.1 เครื่องมือและสภาพแวดล้อมในการพัฒนา

- ภาษาโปรแกรม: Dart (สำหรับ Frontend), Java (สำหรับ Backend)
- Framework: Flutter, Spring Boot
- IDE (Integrated Development Environment): Android Studio และ Visual Studio Code สำหรับ Flutter, IntelliJ IDEA สำหรับ Spring Boot
- ฐานข้อมูล: PostgreSQL
- Plugins/Libraries ที่ใช้:
 - flutter_local_notifications: สำหรับระบบแจ้งเตือน
 - flutter_tesseract_ocr: สำหรับฟังก์ชัน OCR
 - mobile_scanner: สำหรับการสแกน Barcode
 - flutter_tts: สำหรับฟังก์ชัน Text-to-Speech

4.2.2 ขั้นตอนการพัฒนา

การพัฒนา UI/UX

ออกแบบและสร้างหน้าจอการใช้งานตามหลัก Human-Computer Interaction (HCI) ที่เหมาะสมกับผู้สูงอายุ เช่น ใช้ขนาดตัวอักษรที่ใหญ่, สีเส้นที่ตัดกันชัดเจน และมีปุ่มกดที่ชัดเจน

การพัฒนา API Backend

สร้าง RESTful API ด้วย Spring Boot เพื่อจัดการคำขอจากแอปพลิเคชัน (Client) และเชื่อมต่อกับฐานข้อมูล PostgreSQL



รูป 4.17: ออกแบบ UI/UX ของ Application

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 การทดสอบการยอมรับจากผู้ใช้งาน (User Acceptance Testing - UAT)

นำแอปพลิเคชันไปให้กลุ่มผู้สูงอายุทดลองใช้จริง เพื่อรวบรวมข้อเสนอแนะและปัญหาที่พบจากการใช้งาน ซึ่งจะนำไปสู่การปรับปรุงในเวอร์ชันต่อไป

บทที่ 5

การประเมินผล

บทที่ 6

สรุปและข้อเสนอแนะ

เอกสารอ้างอิง

- [1] หอมสนิท □. “NCD โรคไม่ติดต่อ”; 11 2557, สืบค้นเมื่อ 29 สิงหาคม 2568, [ออนไลน์], [Online]. Available: <https://share.google/fLVqWdcNTidgS22mZ>.
- [2] กรมอนามัย. “สถานการณ์ สถิติกลุ่มโรคไม่ติดต่อเรื้อรัง (Non-Communicable diseases)”; 2022, สืบค้นเมื่อ: 17 สิงหาคม 2568, [ออนไลน์], [Online]. Available: <https://share.google/Dsz4woehLYvHsfYik>.
- [3] ศรีบุรมย์ □. “การพัฒนาและการประเมินระบบโปรแกรมประยุกต์บนโทรศัพท์มือถือ เพื่อช่วยในการอ่านฉลากยาสำหรับผู้สูงอายุ”; มีนาคม 2024, สืบค้นเมื่อ: 20 กันยายน 2568, [ออนไลน์], [Online]. Available: <https://he01.tci-thaijo.org/index.php/TJPP/article/view/260634>.
- [4] Dix A.. “Human-Computer Interaction”;, [ออนไลน์], [Online]. Available: <https://dl.acm.org/doi/10.1145/3544548.3581518>.
- [5] Appmaster. “Fundamentals of Database Systems”; 10 2022, สืบค้นเมื่อ: 1 กันยายน 2568, [ออนไลน์], [Online]. Available: <https://appmaster.io/th/blog/postgresql-khuue-aair>.
- [6] Google LLC. “Flutter: Build apps for any screen”; 2024, สืบค้นเมื่อ: 20 กันยายน 2568, [ออนไลน์], [Online]. Available: <https://flutter.dev>.
- [7] Dart. “Dart”; 9 2025, สืบค้นเมื่อ: 20 กันยายน 2568, [ออนไลน์], [Online]. Available: <https://dart.dev/overview>.
- [8] VMware, Inc. . “Spring Boot: Build anything with Spring Boot”; 2024, สืบค้นเมื่อ: 20 กันยายน 2568, [ออนไลน์], [Online]. Available: <https://spring.io/projects/spring-boot>.
- [9] CONTENT D.. “PERSONA คืออะไร สำคัญอย่างไรต่อการสร้างแบรนด์”; 5 2023, สืบค้นเมื่อ: 10 กันยายน 2568, [ออนไลน์], [Online]. Available: <https://digitorystyle.com/what-is-persona/>.
- [10] Figma. “User journey mapping: What it is + how to do it”;, [ออนไลน์], [Online]. Available: <https://www.figma.com/resource-library/user-journey-map>.
- [11] สำนักพัฒนามาตรฐานระบบข้อมูลสุขภาพไทย. “บัญชีข้อมูลยาและรหัสยามาตรฐานไทย”; 1 2013, สืบค้นเมื่อ: 15 สิงหาคม 2568, [ออนไลน์], [Online]. Available: <https://this.or.th>.
- [12] Sirimarnkit N.. “เทคนิคการใช้ PySpark กับ Big Data”; 2022, สืบค้นเมื่อ: 30 ตุลาคม 2568, [ออนไลน์], [Online]. Available: <https://share.google/rRh7ia1zkUHB5IB4U>.

- [13] Kubernetes T.. “kind”; 3 2025, สืบค้นเมื่อ: 8 ตุลาคม 2568, [ออนไลน์], [Online]. Available: <https://kind.sigs.k8s.io/>.
- [14] Abba I.. “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>.

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 ปรรธนา สุภาวงศ์

นาย ฤทธิช พลราช

นางสาว จิรัชญา ราชพลแสน

ภาคผนวก ก

ชื่อของภาคผนวก

ส่วนเพิ่มเติมในเอกสาร นำเสนอข้อมูลเพิ่มเติมที่ไม่สามารถใส่ไว้ในเนื้อหาหลักของเอกสารได้ เพราะอาจทำให้เนื้อหาหลักยาวเกินไป ทำหน้าที่ให้ข้อมูลเสริมที่ช่วยสนับสนุนความเข้าใจในเนื้อหาหลักของรายงาน ตัวอย่างข้อมูลที่ใส่ในภาคผนวก เช่น แบบสอบถาม ข้อมูลดิบ ผลการประเมินแบบละเอียด

ภาคผนวก ข
ชื่อของภาคผนวก