



แอปพลิเคชันรวบรวมข้อมูลช่างเฉพาะทางด้านต่าง ๆ ในแต่ละพื้นที่

จัดทำโดย

นายเป็นหนึ่ง สายทรัพย์ 65021880

นายธนพล ยะใหม่วงศ์ 65021699

นายบุญชูธรรมี่ ชำนาญยา 65021879

รายงานนี้เป็นส่วนหนึ่งของรายวิชาการพัฒนาซอฟต์แวร์บนอุปกรณ์เคลื่อนที่
สาขาวิชา วิศวกรรมคอมพิวเตอร์ คณะเทคโนโลยีสารสนเทศและการสื่อสาร
มหาวิทยาลัยพะเยา ปีการศึกษา 2567

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

บ้านเรือน เครื่องใช้ไฟฟ้า และยานพาหนะของเราล้วนแต่มีอายุการใช้งานที่จำกัด ซึ่งย่อมมีการเสื่อมสภาพหรือเกิดความเสียหายได้จากการใช้งานในชีวิตประจำวัน ทั้งนี้ไม่ว่าจะเป็นปัญหาที่ดูเหมือนจะเล็กน้อยหรือใหญ่ เช่น ไฟฟ้าช็อต น้ำรั่ว หลอดคาร้ว หรือเครื่องยนตรถเสีย ล้วนเป็นปัญหาที่เราต้องเผชิญอยู่บ่อยครั้ง การซ่อมแซมปัญหาเหล่านี้จำเป็นต้องอาศัยความรู้และทักษะเฉพาะทางจากช่างฝีมือที่มีประสบการณ์และความเชี่ยวชาญในการแก้ไขปัญหาต่าง ๆ อย่างถูกต้องและปลอดภัย เพื่อให้เราสามารถกลับมาใช้ชีวิตได้ตามปกติการจ้างช่างผู้เชี่ยวชาญจึงเป็นทางเลือกที่ดีที่สุด เนื่องจากช่างที่มีความสามารถและประสบการณ์จะสามารถระบุปัญหาได้อย่างแม่นยำทำการซ่อมแซมได้อย่างถูกต้องรวดเร็ว ลดความเสี่ยงจากการทำงานผิดพลาดหรือการเกิดอันตราย เช่น อุบัติเหตุจากการซ่อมไฟฟ้าช็อต หรือการเกิดน้ำรั่วที่อาจทำให้เกิดความเสียหายเพิ่มเติม

ปัญหาที่ตามมาคือการหาข้อมูลการติดต่อช่างที่มีความสามารถและมีคุณภาพในงานที่ต้องการให้บริการ โดยปกติแล้วผู้ใช้บริการต้องพึ่งพาการแนะนำจากบุคคลใกล้ชิด เช่น เพื่อน ญาติ หรือคนรู้จัก ซึ่งบางครั้งอาจไม่ได้มีข้อมูลหรือประสบการณ์ที่เพียงพอในการคัดเลือกช่างที่เหมาะสมกับงานที่ต้องการการพึ่งพาคำแนะนำจากบุคคลใกล้ชิดอาจนำมาซึ่งข้อจำกัดในการเลือกช่างที่มีคุณสมบัติตรงกับความต้องการจริง เช่น ช่างที่มีความชำนาญในสาขานั้น ๆ หรือช่างที่สามารถรับงานในระยะเวลาอันสั้นซึ่งไม่สามารถรันทันทีได้ว่าผลลัพธ์ที่ได้จะมีคุณภาพหรือไม่ นอกจากนี้การค้นหาผ่านช่องทางต่าง ๆ เช่น โซเชียลมีเดีย เว็บไซต์ หรือแหล่งข้อมูลออนไลน์อื่น ๆ ก็ไม่ใช่ทางออกที่สมบูรณ์ เนื่องจากบางครั้งข้อมูลที่พบในช่องทางเหล่านี้ไม่ได้มีการตรวจสอบความถูกต้องอย่างชัดเจนอาจพบกับข้อมูลที่เก่าหรือไม่อัปเดตทำให้ผู้ใช้บริการไม่ได้รับข้อมูลที่เป็นปัจจุบันเกี่ยวกับช่างหรือบริการที่มีอยู่ในพื้นที่ ในบางกรณีอาจพบข้อมูลที่ไม่แม่นยำหรือไม่ตรงกับข้อเท็จจริง ซึ่งทำให้การเลือกช่างเป็นไปอย่างสุ่มเสี่ยงและบางครั้งช่างที่แสดงข้อมูลไว้บนเว็บไซต์หรือโซเชียลมีเดีย อาจไม่พร้อมให้บริการจริงหรือมีคุณภาพที่ไม่ได้มาตรฐานตามที่โฆษณาไว้การค้นหาช่างที่มีคุณภาพในพื้นที่ใกล้เคียงยังเป็นปัญหาที่สำคัญ เนื่องจากผู้ใช้บริการมักต้องการหาช่างที่สามารถเดินทางมาถึงได้ทันที เพื่อให้บริการในกรณีที่เกิดเหตุ ที่ต้องได้รับการแก้ไข

ในทันที การหาช่างที่อยู่ใกล้ตัวและพร้อมให้บริการอย่างรวดเร็วยังเป็นเรื่องยาก เนื่องจากการรวมข้อมูลช่างในแหล่งเดียวกันที่สามารถค้นหาได้ง่ายและสะดวก บางครั้งผู้ใช้บริการต้องใช้เวลานานในการติดต่อกับช่างหลายรายและยังไม่สามารถหาช่างที่ว่าง ตรงกับเวลาได้

แอปพลิเคชันนี้เกิดขึ้นจากแนวคิดที่จะช่วยให้การหาช่างมืออาชีพและบริการต่าง ๆ ในพื้นที่ใกล้เคียงเป็นเรื่องง่ายและรวดเร็ว โดยการรวบรวมข้อมูลของช่างที่มีความเชี่ยวชาญในหลาย ๆ ด้านไม่ว่าจะเป็นช่างไฟฟ้า ช่างประปา ช่างซ่อมรถยนต์ หรือช่างอื่นๆ ที่สามารถตอบโจทย์การซ่อมแซมและบริการต่างๆ ภายในบ้านหรือยานพาหนะของผู้ใช้ โดยไม่ต้องเสียเวลาในการติดต่อหลายแห่งหรือเสี่ยงกับบริการที่ไม่ตรงตามความคาดหวังของผู้ใช้

1.2 วัตถุประสงค์

1. เพื่อออกแบบแอปพลิเคชันรวบรวมข้อมูลช่างเฉพาะทางด้านต่าง ๆ ในแต่ละพื้นที่
2. เพื่อพัฒนาแอปพลิเคชันรวบรวมข้อมูลช่างเฉพาะทางด้านต่าง ๆ ในแต่ละพื้นที่
3. เพื่อทดสอบแอปพลิเคชันรวบรวมข้อมูลช่างเฉพาะทางด้านต่าง ๆ ในแต่ละพื้นที่

1.3 ขอบเขต

โครงการนี้มุ่งเน้นการออกแบบและพัฒนาแอปพลิเคชันรวบรวมข้อมูลช่างเฉพาะทางด้านต่าง ๆ ในแต่ละพื้นที่ ที่ครอบคลุมการคัดกรองข้อมูลหรือข้อมูลการติดต่อช่างในพื้นที่ที่ผู้ใช้งานเลือก สามารถตรวจสอบหรือประเมินคะแนนการให้บริการของช่างได้ สามารถแบ่งตามมุมมองออกเป็น 2 หัวข้อหลัก ๆ คือ ความสามารถของระบบและความสามารถของผู้ใช้งาน โดยรายละเอียดต่าง ๆ สามารถอธิบายได้ดังนี้

1.3.1 ความสามารถในการใช้งานของระบบ

- ระบบสามารถคัดกรองและแสดงข้อมูลช่างในพื้นที่ที่ค้นหาได้
- ระบบสามารถคัดกรองและแสดงข้อมูลช่างตามประเภทที่เลือกได้
- ระบบสามารถแสดงข้อมูลช่างที่เลือกได้
- ระบบสามารถแสดงความคิดเห็นของผู้ใช้งานได้
- ระบบสามารถดูช่างคนโปรดของผู้ใช้งานได้
- ระบบสามารถแสดงหน้าโปรไฟล์ของช่างได้
- ระบบสามารถแสดงแบบฟอร์มการสมัครช่างของผู้ใช้งานได้

1.3.2 ความสามารถของผู้ใช้งานระบบ

1.3.2.1 ผู้ใช้งานทั่วไป

- สามารถเลือกพื้นที่ที่ต้องการได้
- สามารถเลือกประเภทช่างที่ต้องการได้
- สามารถเรียกดูข้อมูลช่างที่ต้องการได้
- สามารถค้นหาช่างได้
- สามารถเขียนแสดงความคิดเห็นช่างแต่ละคนได้
- สามารถกดเพิ่มช่างที่ถูกใจได้
- สามารถทำการส่งสมัครช่างได้
- สามารถเปลี่ยนรหัสผ่านได้

1.3.2.2 ช่าง

- สามารถดูรายละเอียดข้อมูลส่วนตัวได้
- สามารถดูข้อความแสดงความคิดเห็นจากผู้ใช้งาน
- สามารถเลือกพื้นที่ที่ต้องการได้
- สามารถเลือกประเภทช่างที่ต้องการได้
- สามารถเรียกดูข้อมูลช่างที่ต้องการได้
- สามารถค้นหาช่างได้
- สามารถเปลี่ยนรหัสผ่านได้

1.4 แผนการดำเนินโครงการ

แผนการดำเนินโครงการและการจัดสร้างแอปพลิเคชันเริ่มจากกระบวนการ ศึกษาค้นคว้าข้อมูล และ ทฤษฎีแอปพลิเคชัน กำหนดวัตถุประสงค์ กำหนดขอบเขตการศึกษา ลงพื้นที่สำรวจความต้องการระบบ วิเคราะห์ปัญหาและการออกแบบ ดำเนินการขั้นตอนสร้างแอปพลิเคชัน CraftLocal ทดสอบแอปพลิเคชัน และแก้ไขข้อผิดพลาดของแอปพลิเคชัน ประเมินผล ข้อเสนอแนะ สรุปผล จัดทำรูปเล่ม นำเสนอ และ ส่งมอบงาน โดยช่วงเวลาของแผนการดำเนินงานแสดงดังตารางที่ 1.1

ตารางที่ 1.1 แผนการดำเนินโครงการงานแอปพลิเคชัน CraftLocal

<div>ระยะเวลา</div> <div>ขั้นตอน</div>	ปีการศึกษา 2567				
	สัปดาห์ที่ 1	สัปดาห์ที่ 2	สัปดาห์ที่ 3	สัปดาห์ที่ 4	สัปดาห์ที่ 5
1. ศึกษาค้นคว้าข้อมูลและ ทฤษฎีแอปพลิเคชัน					
2. ลงพื้นที่ และกำหนด ขอบเขตการศึกษา					
3. วิเคราะห์ปัญหาและการ ออกแบบ					
4. ดำเนินการขั้นตอนสร้าง โปรแกรม					
5. ทดสอบโปรแกรมและ แก้ไขข้อผิดพลาด					
6. ประเมินผล ข้อเสนอแนะ และสรุปผล					
7. จัดทำเอกสาร					
8. นำเสนอ					
9. ส่งมอบงาน					

บทที่ 2

ทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้อง

สำหรับเนื้อหาภายในบทนี้จะกล่าวถึงทฤษฎีพื้นฐานและงานวิจัยที่มีส่วนเกี่ยวข้องกับการพัฒนาแอปพลิเคชัน CraftLocal ซึ่งจะแบ่งเนื้อหาออกเป็น 7 หัวข้อหลัก ๆ ได้แก่ การเลือกใช้การจัดการฐานข้อมูลงานวิจัยที่เกี่ยวข้อง และบทสรุป โดยหัวข้อที่กล่าวมาข้างต้นจะเป็นองค์ความรู้พื้นฐานที่ก่อให้เกิดความรู้ความเข้าใจเพื่อนำไปประยุกต์ใช้ในการพัฒนาระบบสารสนเทศสำหรับบริหารจัดการแผนกชีวภิบาลต่อไป รายละเอียดของเนื้อหาในแต่ละหัวข้อสามารถอธิบายได้ดังต่อไปนี้

2.1 การออกแบบหน้าบ้าน (Frontend)

ในการพัฒนาแอปพลิเคชัน หนึ่งในองค์ประกอบสำคัญคือ Frontend หรือส่วนที่ผู้ใช้งานมองเห็นและโต้ตอบ ซึ่งมีเทคโนโลยีหลายประเภทให้เลือกใช้ โดยแต่ละแบบมีจุดเด่นและข้อจำกัดที่แตกต่างกัน การเลือกเทคโนโลยีที่เหมาะสมจะช่วยให้แอปพลิเคชันทำงานได้อย่างราบรื่น ตอบสนองต่อผู้ใช้ และง่ายต่อการพัฒนา

ประเภทของ Frontend สำหรับแอปพลิเคชัน

2.1.1 Native Frontend

พัฒนาโดยใช้ภาษาพื้นฐานของแต่ละระบบปฏิบัติการ เช่น

- Android: Kotlin หรือ Java
- iOS: Swift หรือ Objective-C
- ข้อดี
 - ประสิทธิภาพสูง (Native Performance)
 - เข้าถึง API ของอุปกรณ์ได้เต็มที่
- ข้อเสีย
 - ต้องพัฒนาแยกกันระหว่าง Android และ iOS ทำให้ใช้เวลาและทรัพยากรมาก

2.1.2 Cross-Platform Frontend

ใช้เฟรมเวิร์กที่ช่วยให้สามารถพัฒนาแอปพลิเคชันบนทั้ง iOS และ Android ได้จากโค้ดชุดเดียว เช่น Flutter (Dart), React Native (JavaScript), Xamarin (C#)

- ข้อดี
 - พัฒนาโค้ดครั้งเดียว ใช้ได้ทั้ง Android และ iOS
 - ลดเวลาและค่าใช้จ่ายในการพัฒนา
- ข้อเสีย
 - ประสิทธิภาพอาจต่ำกว่า Native เล็กน้อย
 - ต้องใช้แพ็คเกจเสริมในการเข้าถึงฟีเจอร์เฉพาะของอุปกรณ์

2.1.3 Web-Based Frontend (Progressive Web App - PWA)

ใช้เทคโนโลยีเว็บ เช่น HTML, CSS, JavaScript เฟรมเวิร์ก เช่น Vue.js, React.js, Angular

- ข้อดี
 - ทำงานได้บนทุกอุปกรณ์ผ่านเบราว์เซอร์
 - ไม่ต้องติดตั้งแอปในเครื่อง
- ข้อเสีย
 - ไม่สามารถเข้าถึงฟีเจอร์ของอุปกรณ์ได้เต็มที่ เช่น GPS, Bluetooth
 - ประสิทธิภาพอาจต่ำกว่าการพัฒนาแบบ Native

ทำไมเลือกใช้ Flutter เป็น Frontend

- ความสามารถของ Flutter
 - พัฒนาแอปพลิเคชัน Cross-Platform ได้จากโค้ดเดียว
 - ใช้ภาษา Dart ที่มีประสิทธิภาพสูง
 - รองรับ Hot Reload ทำให้พัฒนาและแก้ไขโค้ดได้เร็วขึ้น
 - มี UI ที่สวยงามและปรับแต่งได้ง่ายผ่าน Widget



รูปที่ 2.1 สัญลักษณ์ของฟลัตเตอร์

ตารางที่ 2.1 ตารางเปรียบเทียบ Flutter กับเฟรมเวิร์กอื่น ๆ

	Flutter	React Native	Kotlin (Android)	Swift (iOS)
ใช้ได้ทุกแพลตฟอร์ม	✓	✓	✗	✗
ประสิทธิภาพสูง	✓	✗	✓	✓
เข้าถึงฟีเจอร์ของ อุปกรณ์	✓	✗	✓	✓
การพัฒนา UI ที่ ยืดหยุ่น	✓	✓	✗	✗
ชุมชนและเครื่องมือ สนับสนุน	✓	✓	✓	✓

2.2 การออกแบบหลังบ้าน (Backend)

การเลือกใช้ Backend API เป็นสิ่งสำคัญในการพัฒนาแอปพลิเคชัน เนื่องจากมันจะเป็นตัวกลางที่เชื่อมต่อและส่งข้อมูลระหว่าง Frontend (เช่น แอป Flutter) และฐานข้อมูล (เช่น PostgreSQL) นอกจากนี้ ยังต้องพิจารณาความต้องการเฉพาะของแอป และเลือกเทคโนโลยีที่เหมาะสมกับโครงการ

- ปัจจัยในการเลือกใช้ Backend API

1. ประสิทธิภาพและการรองรับการทำงานพร้อมกัน (Scalability)

หากแอปต้องรองรับการใช้งานจำนวนมาก หรือมีการทำงานแบบเรียลไทม์ เช่น การแสดงข้อมูลที่อัปเดตตลอดเวลา (Real-Time Data), การเลือก Backend API ที่รองรับ WebSocket หรือ gRPC จะเป็นทางเลือกที่ดี

Node.js เหมาะกับงานที่ต้องการประสิทธิภาพสูงเพราะใช้ Non-blocking I/O ทำให้รองรับคำขอหลาย ๆ คำขอพร้อมกันได้ดี

Spring Boot (Java) หรือ Django (Python) อาจเหมาะสำหรับระบบที่ต้องการ scalability สูง และรองรับการทำงานที่ซับซ้อน

2. ความง่ายในการพัฒนาและเรียนรู้ (Ease of Use and Learning Curve)

ถ้าโปรเจกต์ของคุณต้องการความรวดเร็วในการพัฒนาและมีทีมที่ถนัด JavaScript, การใช้ Node.js กับ Express.js หรือ NestJS จะเหมาะสมมาก

Django (Python) เป็นตัวเลือกที่ดีสำหรับทีมที่ถนัด Python เพราะมันมี convention over configuration ทำให้การพัฒนา API สะดวกและรวดเร็ว

Laravel (PHP) ก็เป็นทางเลือกที่ดีสำหรับทีมที่มีประสบการณ์ใน PHP และต้องการสร้างเว็บแอปหรือ API ที่รวดเร็ว

3. ความเหมาะสมกับฐานข้อมูล (Database Compatibility)

ทุก ๆ Backend API จะรองรับการเชื่อมต่อกับ ฐานข้อมูล ไม่ว่าจะเป็น PostgreSQL, MySQL, MongoDB หรือฐานข้อมูลอื่น ๆ

หากแอปต้องการการประมวลผลข้อมูลที่มีความซับซ้อน Django หรือ Spring Boot อาจเหมาะสมเพราะรองรับการทำงานร่วมกับฐานข้อมูลเชิงสัมพันธ์ได้ดี

Node.js ก็สามารถเชื่อมต่อกับฐานข้อมูลเชิงสัมพันธ์และ NoSQL ได้ดีเช่นกันผ่านไลบรารีต่าง ๆ เช่น pg สำหรับ PostgreSQL หรือ mongoose สำหรับ MongoDB

4. การรองรับ API ประเภทต่าง ๆ (API Type Support)

REST API: เป็น API ประเภทที่นิยมที่สุด ใช้งานง่ายและรองรับการสื่อสารผ่าน HTTP Methods เช่น GET, POST, PUT, DELETE

GraphQL: เหมาะสำหรับแอปที่ต้องการดึงข้อมูลในรูปแบบที่ยืดหยุ่นและสามารถดึงข้อมูลหลายชุดในคำขอเดียว

WebSocket: เหมาะสำหรับการพัฒนาแอปที่ต้องการทำงานแบบ Real-time เช่น แอปแชทหรือแอปที่แสดงข้อมูลที่อัปเดตต่อเนื่อง

gRPC: เหมาะกับการพัฒนา Microservices ที่ต้องการประสิทธิภาพสูงในการสื่อสารระหว่างบริการ

5. ความปลอดภัย (Security)

ทุก API ต้องมีการ ป้องกันข้อมูล และ ตรวจสอบสิทธิ์ ของผู้ใช้ เช่น การใช้งาน JWT (JSON Web Token) เพื่อจัดการการยืนยันตัวตน

Django มาพร้อมกับระบบ Authentication และ Authorization ในตัว ทำให้การพัฒนาแอปที่ต้องการความปลอดภัยสูงง่ายขึ้น

Node.js ก็สามารถใช้ JWT หรือ OAuth เพื่อตรวจสอบการเข้าถึงข้อมูลได้ง่ายเช่นกัน

● ทำไมเลือก Node.js

Node.js เป็นแพลตฟอร์มที่ใช้ JavaScript สำหรับพัฒนา Backend API ซึ่งมีข้อดีหลายอย่าง เช่น

ประสิทธิภาพสูง : ใช้ Non-blocking I/O ทำให้รองรับการทำงานพร้อมกันหลายคำขอได้ดี

รองรับ REST, GraphQL และ WebSocket : สามารถสร้าง API ได้หลายรูปแบบ

ชุมชนขนาดใหญ่ : มีไลบรารีและแพ็คเกจให้เลือกใช้งานมากมาย เช่น Express.js, NestJS

รองรับการทำงานร่วมกับ PostgreSQL ได้ดี : ใช้ไลบรารีอย่าง pg หรือ Prisma เพื่อเชื่อมต่อฐานข้อมูล



รูปที่ 2.2 สัญลักษณ์ของโนดเจเอส

- Node.js เทียบกับ Backend อื่น ๆ

ข้อดี

- เร็วและมีประสิทธิภาพสูง → ใช้ Non-blocking I/O ทำให้รองรับคำขอจำนวนมากได้ดี
- รองรับทั้ง REST, GraphQL และ WebSocket → ใช้พัฒนาได้หลากหลาย
- ใช้ JavaScript ทั้ง Frontend และ Backend → ลดปัญหาการเรียนรู้หลายภาษา
- มีชุมชนขนาดใหญ่และไลบรารีเยอะ → มีแพ็คเกจให้เลือกใช้มากมาย เช่น express, socket.io, nestjs
- เหมาะกับ Microservices และ Cloud-based API → ใช้ร่วมกับ Docker และ Kubernetes ได้ง่าย

ข้อเสีย

- ไม่เหมาะกับงานที่ต้องใช้ CPU สูง เช่น Machine Learning หรือ AI
- ต้องบริหารหน่วยความจำดี ๆ เพราะใช้ Single Threaded
- Callback Hell อาจเกิดขึ้นหากเขียนโค้ดแบบ Nested Callbacks (แต่สามารถแก้ได้โดยใช้ Async/Await หรือ Promise)

2.3 ฐานข้อมูลที่ใช้จัดการข้อมูล (Database Management Systems)

การเลือกใช้ Database เป็นปัจจัยสำคัญในการพัฒนาแอปพลิเคชัน เนื่องจากมันมีผลต่อประสิทธิภาพ, ความสามารถในการขยายตัว (scalability), และการจัดการข้อมูล การเลือก Database ที่

เหมาะสมกับลักษณะงานและความต้องการของแอปจะช่วยให้การพัฒนาและการบำรุงรักษาแอปเป็นไปได้อย่างราบรื่น

- ปัจจัยที่ต้องพิจารณาในการเลือกใช้ Database

1. ประเภทของข้อมูล (Data Type)

- Structured Data: ถ้าข้อมูลที่ต้องการจัดเก็บเป็น ข้อมูลที่มีโครงสร้าง (structured data) เช่น ข้อมูลในรูปแบบ ตาราง (Table) หรือ ไฟล์ที่มีความสัมพันธ์ กัน, เช่น ระบบบัญชี หรือการจัดการผู้ใช้, Relational Database (RDBMS) เช่น PostgreSQL หรือ MySQL จะเหมาะสม
- Unstructured Data: หากข้อมูลที่ต้องการจัดเก็บเป็น ข้อมูลที่ไม่มีโครงสร้าง หรือ ข้อมูลที่มีการเปลี่ยนแปลงบ่อย เช่น ข้อความ, รูปภาพ, หรือ Log Data, NoSQL Database เช่น MongoDB หรือ Cassandra อาจเป็นตัวเลือกที่ดี

2. ความสามารถในการขยายตัว (Scalability)

- Horizontal Scaling: ถ้าคุณคาดว่าแอปจะมีปริมาณข้อมูลสูงมากหรือมีการเข้าถึงจากผู้ใช้จำนวนมาก, คุณอาจต้องการ NoSQL Database ที่รองรับการขยายตัวแบบ horizontal scaling เช่น MongoDB, Cassandra หรือ Couchbase
- Vertical Scaling: ถ้าคุณต้องการระบบที่สามารถขยายประสิทธิภาพได้ง่าย ๆ โดยเพิ่มทรัพยากรของเครื่อง (CPU, RAM) ระบบ Relational Database เช่น PostgreSQL หรือ MySQL ก็ยังสามารถรองรับการขยายตัวในลักษณะนี้ได้ดี

3. ความปลอดภัยและความเชื่อถือได้ (Security & Reliability)

- RDBMS เช่น PostgreSQL และ MySQL มักจะมีระบบ Transaction และ ACID properties ที่ช่วยให้การจัดการข้อมูลมีความถูกต้องและเชื่อถือได้
- NoSQL Database เช่น MongoDB อาจไม่มีการรองรับ ACID อย่างเต็มที่ แต่ก็มีการออกแบบให้รองรับการกระจายข้อมูลในระบบที่มีขนาดใหญ่ได้

4. ความง่ายในการตั้งค่าและการบำรุงรักษา (Ease of Setup and Maintenance)

- Relational Databases เช่น PostgreSQL และ MySQL มักจะมีการตั้งค่าและการจัดการที่เป็นมาตรฐานและเหมาะสมกับระบบที่ต้องการความเข้ากันได้สูง
- NoSQL เช่น MongoDB และ Cassandra จะมีการตั้งค่าที่ยืดหยุ่นและสามารถปรับตัวได้ง่ายเมื่อต้องการรองรับข้อมูลที่มีการเปลี่ยนแปลงบ่อย

5. การสนับสนุนและเครื่องมือ (Support & Tools)

- ระบบ Relational Database อย่าง PostgreSQL และ MySQL มีเครื่องมือมากมายที่สามารถใช้ในการบริหารจัดการฐานข้อมูล เช่น pgAdmin หรือ MySQL Workbench
- NoSQL Database เช่น MongoDB มีเครื่องมือเช่น MongoDB Atlas ที่ช่วยให้การจัดการและการทำงานกับฐานข้อมูลง่ายขึ้น

ตารางที่ 2.2 ตารางแสดงการเปรียบเทียบ PostgreSQL กับ Database อื่น ๆ

คุณสมบัติ	PostgreSQL	MySQL	MongoDB	Firebase (Firestore)	Cassandra
ประเภท	Relational (SQL)	Relational (SQL)	NoSQL (Document)	NoSQL (Document)	NoSQL (Column-based)
โครงสร้างข้อมูล	ตาราง (Table)	ตาราง (Table)	JSON/BSON	JSON	Wide-column Store
รองรับ ACID	ใช่	ใช่ (InnoDB)	ไม่เต็มรูปแบบ	ไม่รองรับ	ไม่รองรับ
การรองรับ Transaction	ดีมาก	ดีพอสมควร	จำกัด	ไม่รองรับ	ไม่รองรับ
Scalability (ขยายระบบ)	แนวดิ่งและแนวนอน	แนวดิ่งและแนวนอน	แนวนอน	แนวนอน	แนวนอน
ความเร็วในการอ่าน/เขียน	เร็ว	เร็วมาก	เร็วมาก	เร็วมาก	เร็ว

เหมาะกับ ข้อมูลประเภท	โครงสร้าง ชัดเจน, ข้อมูล ที่ต้องการ Integrity สูง	โครงสร้าง ชัดเจน, แอป ทั่วไป	ข้อมูลไม่มี โครงสร้าง เช่น JSON	แอป Real- time	Big Data, Distributed Systems
การรองรับ JSON	ดีมาก (Native JSONB)	รองรับ (แต่ไม่ดี เท่า PostgreSQL)	ดีมาก	รองรับ	ไม่รองรับ
เหมาะสำหรับ	ระบบองค์กร, แอปพลิเคชัน ขนาดใหญ่	แอปทั่วไป, เว็บ แอป	แอปที่ต้องการ ความยืดหยุ่น สูง เช่น Social Media	แอปที่ต้องการ Real-time เช่น Chat	Big Data, AI, ระบบ กระจาย ข้อมูล

2.4 งานวิจัยที่เกี่ยวข้อง

งานวิจัยด้าน Matching Algorithm

หัวข้อ: "Matching Service Providers with Consumers in Urban Areas Based on Proximity and Service Type"

- การจับคู่ผู้ใช้งานกับช่างซ่อมโดยพิจารณาจากที่อยู่และประเภทบริการ
- ประยุกต์ใช้ในแอป: สร้างระบบจับคู่ช่างกับผู้ใช้ โดยจัดลำดับจากระยะทางและความเชี่ยวชาญ

งานวิจัยด้านการพัฒนาระบบจัดการข้อมูล

หัวข้อ: "Database Optimization for Real-Time Applications"

- การปรับปรุงประสิทธิภาพฐานข้อมูลเพื่อรองรับการเข้าถึงข้อมูลแบบเรียลไทม์ในแอปพลิเคชัน
- ประยุกต์ใช้ในแอป: ใช้เทคนิคการสร้างดัชนี (Indexing) และการแคชข้อมูล (Caching) เพื่อลดเวลาในการดึงข้อมูล

งานวิจัยด้านความปลอดภัยข้อมูล

หัวข้อ: "Securing User Data in Mobile Applications: A Privacy-Centric Approach"

- การออกแบบระบบที่คำนึงถึงความปลอดภัยและความเป็นส่วนตัวของผู้ใช้งาน

- ประยุกต์ใช้ในแอป: เพิ่มการยืนยันตัวตนหลายขั้นตอน (Multi-Factor Authentication) และการแจ้งเตือนเมื่อมีการเข้าถึงข้อมูลส่วนตัว

งานวิจัยด้านการหาตำแหน่งผู้ใช้งาน (User Location Tracking or Positioning)

สาขาที่ได้รับความสนใจอย่างมากในหลายๆ ด้าน เช่น การติดตามพฤติกรรมผู้ใช้, การประยุกต์ใช้ในระบบนำทาง, การตลาดที่ขึ้นอยู่กับตำแหน่ง, การติดตามสุขภาพ, และการขนส่ง โดยเทคโนโลยีที่ใช้ในการหาตำแหน่งผู้ใช้งานมีหลายประเภท ซึ่งรวมถึง GPS (Global Positioning System), Wi-Fi positioning, Bluetooth Low Energy (BLE), RFID, Ultrasound, และ Sensor Fusion เป็นต้น

แนวทางการวิจัยในด้านการหาตำแหน่งผู้ใช้งาน

1. การใช้เทคโนโลยีต่างๆ สำหรับการหาตำแหน่ง

o GPS (Global Positioning System): ใช้สัญญาณดาวเทียมในการระบุตำแหน่งในพื้นที่กลางแจ้ง งานวิจัยในด้านนี้มักจะมุ่งเน้นการเพิ่มความแม่นยำในการระบุตำแหน่งที่เกิดจากสัญญาณดาวเทียมในพื้นที่ที่มีสิ่งกีดขวาง เช่น ในเมืองที่มีอาคารสูงหรือในอาคาร

o Wi-Fi Positioning: ใช้สัญญาณ Wi-Fi ในการหาตำแหน่งในพื้นที่ในร่ม ซึ่งเป็นการใช้ จุดเชื่อมต่อ Wi-Fi ที่รู้จักเพื่อระบุตำแหน่งของผู้ใช้ งานวิจัยมักจะมุ่งเน้นการปรับปรุง ความแม่นยำของระบบ Wi-Fi โดยการใช้เทคนิคการคำนวณระยะทางจากสัญญาณหรือ การปรับปรุงฐานข้อมูลที่เชื่อมโยงกับตำแหน่ง

o Bluetooth Low Energy (BLE) Beacons: ใช้บีคอน Bluetooth ในการติดตาม ตำแหน่งในระยะใกล้ เหมาะสำหรับใช้งานในสถานที่ต่างๆ เช่น ห้างสรรพสินค้า หรือพิพิธภัณฑ์

o Sensor Fusion: ใช้การรวมข้อมูลจากหลายๆ เซ็นเซอร์ เช่น GPS, accelerometer, gyroscope เพื่อเพิ่มความแม่นยำในการระบุตำแหน่งในสภาพแวดล้อมที่หลากหลาย โดยเฉพาะในสภาพแวดล้อมที่ GPS อาจจะไม่สามารถทำงานได้ดี (เช่น ในอาคาร)

2. การประยุกต์ใช้การหาตำแหน่งผู้ใช้งาน

o Location-Based Services (LBS): ใช้ข้อมูลตำแหน่งของผู้ใช้งานในการให้บริการที่เกี่ยวข้องกับสถานที่ เช่น การนำทาง, การส่งข้อความหรือโปรโมชั่นตามตำแหน่งที่ตั้ง

- o Indoor Positioning Systems (IPS): ระบบที่ใช้ในอาคารเพื่อระบุตำแหน่งของผู้ใช้งาน เช่น การใช้ Wi-Fi, BLE, หรือ UWB (Ultra-Wideband) ในการติดตามตำแหน่งใน สถานที่ในร่ม เช่น โรงพยาบาล ห้างสรรพสินค้า

- o การตลาดที่ใช้ตำแหน่ง (Location-based Marketing): ใช้ตำแหน่งของผู้ใช้งานในการ นำเสนอโฆษณาหรือโปรโมชั่นในเวลาที่เหมาะสมเมื่อผู้ใช้เข้ามาในพื้นที่ที่ต้องการ

- o การติดตามสุขภาพ: การใช้ข้อมูลตำแหน่งเพื่อวิเคราะห์พฤติกรรมการเคลื่อนไหวของ ผู้ใช้งานหรือในการติดตามสุขภาพผ่านอุปกรณ์ที่มีเซ็นเซอร์ต่างๆ

3. การเพิ่มประสิทธิภาพและความแม่นยำในการหาตำแหน่ง

- o การใช้การฟิวชั่นข้อมูล (Data Fusion): การผสานข้อมูลจากหลายแหล่ง เช่น GPS, Wi-Fi, Bluetooth, และเซ็นเซอร์อื่นๆ เพื่อลดข้อผิดพลาดและเพิ่มความแม่นยำในการหา ตำแหน่ง โดยงานวิจัยมักจะศึกษาการใช้ Kalman filter, Particle filter หรือ Extended Kalman filter เพื่อดำเนินตำแหน่งที่ถูกต้องมากขึ้น

- o การใช้ Machine Learning: การใช้เทคนิคการเรียนรู้ของเครื่องในการปรับปรุงความ แม่นยำของการระบุตำแหน่ง เช่น การใช้ Deep Learning ในการเรียนรู้รูปแบบจาก ข้อมูลตำแหน่งในอดีต

- o การประเมินความแม่นยำ (Accuracy Assessment): การทดสอบความแม่นยำของ ระบบการหาตำแหน่งในสถานการณ์ต่างๆ และการปรับปรุงอัลกอริธึมเพื่อให้ผลลัพธ์ที่ ดีกว่า

4. ปัญหาด้านความเป็นส่วนตัวและความปลอดภัย

- o Privacy Concerns: งานวิจัยในด้านนี้เน้นการพัฒนาเทคนิคการปกป้องข้อมูลตำแหน่ง ของผู้ใช้งาน โดยการใช้เทคนิคต่างๆ เช่น Differential Privacy หรือการเข้ารหัสข้อมูล ตำแหน่ง

- o Security Issues: การป้องกันการโจมตีและการติดตามข้อมูลโดยไม่ได้รับอนุญาต โดย การพัฒนา

ระบบที่สามารถตรวจจับการแอบติดตามหรือการรั่วไหลของข้อมูล

ตัวอย่างงานวิจัยที่เกี่ยวข้องกับการหาตำแหน่งผู้ใช้งาน

1. "A Survey of Indoor Positioning Systems for Wireless Personal Networks" (2012)

o งานวิจัยนี้ทำการสำรวจเทคนิคการหาตำแหน่งภายในอาคาร (Indoor Positioning Systems - IPS) โดยมีการทบทวนเทคโนโลยีต่างๆ เช่น Wi-Fi, Bluetooth, RFID, และ UWB พร้อมทั้งการประยุกต์ใช้งานในสถานที่ต่างๆ เช่น ห้างสรรพสินค้า, สถานีรถไฟ, หรือโรงพยาบาล

2. "Bluetooth Indoor Positioning System Based on RSSI Measurements" (2015)

o งานวิจัยนี้ศึกษาเกี่ยวกับการหาตำแหน่งภายในอาคารโดยใช้ Bluetooth Low Energy และการวัด RSSI (Received Signal Strength Indicator) เพื่อประเมินตำแหน่งของ ผู้ใช้งานในระยะใกล้

3. "Location-Based Services: A Survey and Future Directions" (2014)

o งานวิจัยนี้ทบทวนการประยุกต์ใช้งาน Location-based Services (LBS) รวมถึงการใช้ ตำแหน่ง ผู้ใช้งานในการนำทาง, การตลาดตามตำแหน่ง, การติดตามพฤติกรรมผู้ใช้งาน และปัญหาด้านความเป็นส่วนตัว

4. "An Overview of the Indoor Positioning Techniques Based on RSSI, Bluetooth, and UWB" (2018)

o การทบทวนเทคนิคการหาตำแหน่งภายในอาคาร โดยใช้ RSSI (Received Signal Strength Indicator) จาก Bluetooth และ UWB (Ultra-Wideband) รวมถึงข้อดี ข้อเสียของแต่ละเทคโนโลยี งานวิจัยด้านการออกแบบ UX/UI

หัวข้อ: "Improving User Engagement through Simplified Interface Design"

การสร้าง UI ที่ตอบสนองผู้ใช้ด้วยการออกแบบหน้าจอที่ชัดเจนและเรียบง่าย

ประยุกต์ใช้ในแอป: ออกแบบหน้าจอรายการช่างแบบเรียบง่ายพร้อมข้อมูลสำคัญ เช่น คะแนน รีวิวและเบอร์โทร

หัวข้อ: "Improving User Engagement through Simplified Interface Design"

การสร้าง UI ที่ตอบสนองผู้ใช้ด้วยปุ่มที่เรียบง่ายและการนำเสนอข้อมูลที่ชัดเจน ประยุกต์ใช้ในแอป:

ออกแบบหน้าจอแสดงรายชื่อช่างเป็นแบบรายการ (ListView) ที่มีปุ่ม "โทรหา" และ "รายละเอียด"

ระบบสารสนเทศแจ้งซ่อมส่วนงานช่างโดยใช้แอปพลิเคชันมือถือ

- งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาแอปพลิเคชันแจ้งซ่อมสำหรับส่วนงานช่างภายในมหาวิทยาลัยราชภัฏพระนครศรีอยุธยา เพื่อเพิ่มประสิทธิภาพในการจัดการงานซ่อมบำรุงและความพึงพอใจของผู้ใช้งาน

Link : ระบบสารสนเทศแจ้งซ่อมส่วนงานช่างโดยใช้แอปพล

แอปพลิเคชันค้นหาตำแหน่งและบริการอยู่ซ่อมรถ

- งานวิจัยนี้มีวัตถุประสงค์ในการพัฒนาแอปพลิเคชันสำหรับค้นหาตำแหน่งและบริการของอยู่ซ่อม รถ เพื่ออำนวยความสะดวกให้กับผู้ใช้ในการค้นหาอยู่ซ่อมที่ตรงกับความต้องการ

Link : แอปพลิเคชันค้นหาตำแหน่งและบริการอยู่ซ่อมรถ Loca

ระบบบริหารงานแจ้งซ่อมบำรุงออนไลน์ SC Maintenance Management System

- งานวิจัยนี้พัฒนาระบบออนไลน์สำหรับการแจ้งซ่อมบำรุง เพื่อเพิ่มประสิทธิภาพในการจัดการงานซ่อมบำรุงภายในคณะวิทยาศาสตร์ มหาวิทยาลัยมหิดล

Link : 36 ระบบบริหารงานแจ้งซ่อมบำรุงออนไลน์ SC Maintenance Management

การพัฒนาระบบแจ้งซ่อมคอมพิวเตอร์ออนไลน์ผ่านการแจ้งเตือนผ่านไลน์และเฟซบุ๊กแอสเซนเจอร์

- งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาระบบแจ้งซ่อมคอมพิวเตอร์ออนไลน์ โดยมีการแจ้งเตือนผ่านแอปพลิเคชันไลน์และเฟซบุ๊กแอสเซนเจอร์ เพื่อเพิ่มความสะดวกในการแจ้งซ่อมและติดตามสถานะการซ่อม

Link : การพัฒนาระบบแจ้งซ่อมคอมพิวเตอร์ออนไลน์ ผ่าน

บทที่ 3

การวิเคราะห์และออกแบบระบบ

ในบทนี้จะกล่าวถึงหลักการวิเคราะห์และออกแบบแอปพลิเคชัน CraftLocal อันประกอบด้วย ต้นแบบหน้าเว็บไซต์ โครงสร้างฐานข้อมูล คู่มือการติดตั้งแอปพลิเคชันและการทำงานของแอปพลิเคชัน รายละเอียดในหัวข้อต่าง ๆ สามารถอธิบายได้ดังต่อไปนี้

3.1 ต้นแบบหน้าเว็บไซต์ (Mockup)

3.1.1 หน้าลงชื่อเข้าใช้ของแอปพลิเคชัน

หน้าลงชื่อเข้าใช้ของแอปพลิเคชันได้รับการออกแบบให้ใช้งานง่าย โดยมีองค์ประกอบหลักคือปุ่ม "ลงชื่อ" ซึ่งเป็นปุ่มสำคัญที่ใช้สำหรับการเข้าถึงระบบ เพื่อให้ผู้ใช้งานสามารถใช้ฟังก์ชันต่างๆ ของแอปพลิเคชันได้อย่างสะดวกและมีประสิทธิภาพ โดยมีองค์ประกอบสำคัญดังต่อไปนี้

- ช่องกรอกข้อมูล: ประกอบด้วยช่องสำหรับป้อนอีเมลและช่องสำหรับกรอกรหัสผ่าน ซึ่งเป็นข้อมูลที่จำเป็นสำหรับการยืนยันตัวตนของผู้ใช้
- ปุ่ม "Login": ใช้สำหรับยืนยันข้อมูลที่กรอกและดำเนินการเข้าสู่ระบบ
- ปุ่ม "สมัครสมาชิก": ปุ่มสำหรับผู้ใช้งานใหม่ที่ต้องการลงทะเบียนเพื่อสร้างบัญชีผู้ใช้งาน



รูปที่ 3.1 หน้าลงชื่อเข้าใช้ของแอปพลิเคชันค้นหาช่างภายในพื้นที่

3.1.2 หน้าสมัครสมาชิก

หน้าสมัครสมาชิกของแอปพลิเคชัน Craftlocal ได้รับการออกแบบมาเพื่อให้ผู้ใช้งานสามารถสร้างบัญชีใหม่ เพื่อเข้าถึงฟังก์ชันต่างๆ ภายในแอปพลิเคชัน โดยมีองค์ประกอบหลักดังนี้

- ช่องกรอกข้อมูล: ประกอบด้วยช่องกรอกอีเมลและช่องกรอกรหัสผ่าน ซึ่งเป็นข้อมูลที่จำเป็นสำหรับการสร้างบัญชีใหม่
- ปุ่ม "สมัคร": ใช้สำหรับยืนยันข้อมูลที่กรอกและดำเนินการสร้างบัญชีผู้ใช้งาน



รูปที่ 3.2 หน้าสมัครสมาชิกของแอปจัดหารบริการช่างในพื้นที่

3.1.3 หน้าหลักสำหรับผู้ใช้งาน

หน้าหลักของแอปพลิเคชัน Craftlocal ถูกออกแบบเพื่อให้ผู้ใช้งานสามารถเข้าถึงฟังก์ชันต่างๆ ภายในแอปพลิเคชันได้อย่างรวดเร็วและสะดวก โดยมีองค์ประกอบหลักที่สำคัญดังนี้

- ปุ่มค้นหาหามุมบนขวา ปุ่มนี้ช่วยให้ผู้ใช้งานสามารถค้นหาช่างหรือบริการที่ต้องการได้อย่างรวดเร็ว โดยไม่จำเป็นต้องเลื่อนดูข้อมูลด้วยตนเอง
- ฟีเจอร์กรองช่าง ระบบกรองที่ช่วยให้ผู้ใช้งานสามารถเลือกประเภทของช่างที่ต้องการได้ เช่น ช่างซ่อมไฟฟ้า ช่างซ่อมแอร์ หรือช่างประเภทอื่น ๆ ตามความต้องการ
- หน้าต่างเลือกช่างบริการ ผู้ใช้งานสามารถเลือกช่างบริการตามประเภทที่ต้องการได้จากหน้าต่างที่มีการแสดงประเภทต่างๆ เช่น ช่างไฟฟ้า ช่างแอร์ หรือช่างประปา



รูปที่ 3.3 หน้าหลักของแอปบริการช่างในพื้นที่

3.1.4 หน้ารายละเอียดช่าง

หน้ารายละเอียดของช่างในแอปพลิเคชัน Craftlocal ถูกออกแบบมาเพื่อให้ผู้ใช้งานสามารถเข้าถึงข้อมูลที่สำคัญเกี่ยวกับช่างบริการได้อย่างครบถ้วน โดยประกอบด้วยองค์ประกอบหลักดังนี้

- ที่อยู่: แสดงที่อยู่ของช่างเพื่อให้ผู้ใช้งานสามารถทราบตำแหน่งที่ตั้งหรือบริการที่ช่างสามารถให้บริการได้
- เบอร์โทรศัพท์: แสดงหมายเลขโทรศัพท์ของช่างเพื่อให้ผู้ใช้งานสามารถติดต่อได้โดยตรง
- พื้นที่ให้บริการ: แสดงข้อมูลเกี่ยวกับพื้นที่หรือเขตที่ช่างให้บริการ เช่น จังหวัดหรืออำเภอ
- การรีวิวและคะแนน: แสดงความคิดเห็นจากผู้ใช้งานก่อนหน้าและคะแนนการบริการ เพื่อให้ผู้ใช้งานสามารถตัดสินใจได้ดีขึ้น
- รูปโปรไฟล์: แสดงรูปภาพของช่างเพื่อเพิ่มความน่าเชื่อถือและความคุ้นเคย

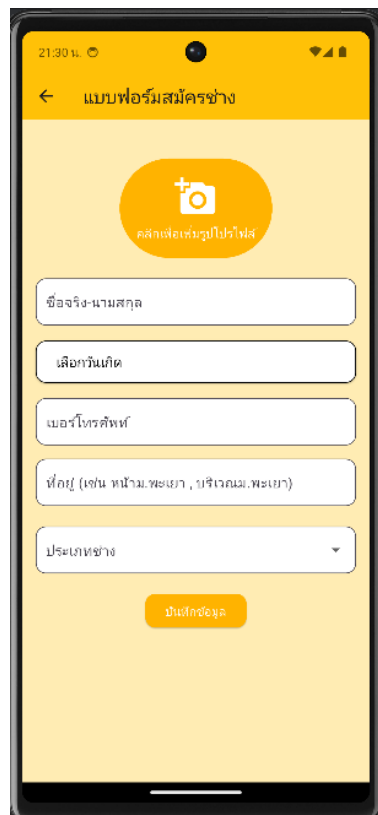


รูปที่ 3.4 หน้ารายละเอียดช่างแอปบริการช่างในพื้นที่

3.1.5 หน้าสมัครช่าง

หน้าสำหรับสมัครช่างในแอปพลิเคชัน Craftlocal ถูกออกแบบมาเพื่อให้ผู้ที่สนใจสมัครเป็นช่างสามารถลงทะเบียนและเริ่มให้บริการได้อย่างสะดวก โดยประกอบด้วยองค์ประกอบหลักดังนี้

- ช่องกรอกข้อมูลส่วนตัว: ประกอบด้วยช่องสำหรับกรอกข้อมูลที่จำเป็น เช่น ชื่อ-นามสกุล, อีเมล, เบอร์โทรศัพท์, และประเภทของช่างที่ต้องการสมัคร
- ช่องกรอกข้อมูลที่อยู่: เพื่อให้ผู้สมัครสามารถระบุที่ตั้งหรือพื้นที่ที่สามารถให้บริการได้
- ช่องอัปโหลดโปรไฟล์: ให้ช่างสามารถอัปโหลดรูปโปรไฟล์เพื่อแสดงตัวตนในการให้บริการ เพิ่มความน่าเชื่อถือ
- ปุ่ม "สมัคร": ปุ่มหลักที่ใช้สำหรับส่งข้อมูลที่กรอกและดำเนินการสมัครเข้าสู่ระบบ

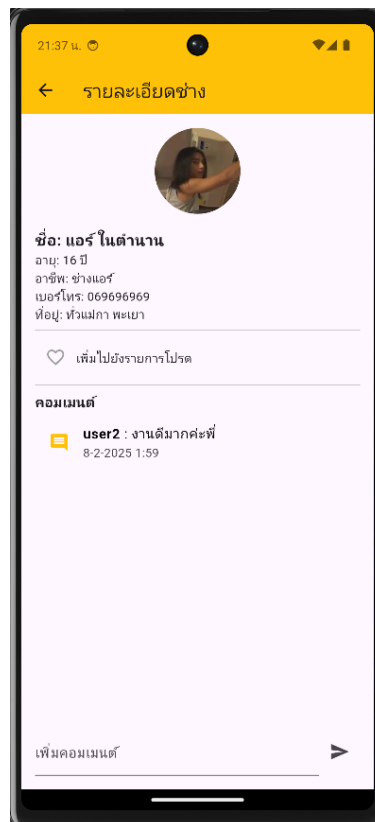


รูปที่ 3.5 หน้ารายละเอียดของการสมัครช่างแอปบริการช่างในพื้นที่

3.1.6 หน้าคอมเมนต์

หน้าคอมเมนต์ในแอปพลิเคชัน Craftlocal ถูกออกแบบมาเพื่อให้ผู้ใช้งานสามารถแสดงความคิดเห็นเกี่ยวกับบริการที่ได้รับจากช่าง รวมถึงให้แอดมินหรือผู้ใช้อื่น ๆ สามารถติดตามการรีวิวและความพึงพอใจได้ง่าย ๆ โดยมีองค์ประกอบหลักดังนี้

- ช่องกรอกคอมเมนต์: ให้ผู้ใช้งานสามารถกรอกความคิดเห็นเกี่ยวกับการบริการที่ได้รับ เช่น คุณภาพการบริการ, ความเร็วในการทำงาน, หรือประสบการณ์โดยรวม
- คะแนนรีวิว (Rating): ผู้ใช้งานสามารถให้คะแนนบริการจากช่างในรูปแบบดาว 1-5 ดาว เพื่อแสดงความพึงพอใจ
- แสดงคอมเมนต์จากผู้ใช้งาน: คอมเมนต์ที่ได้รับจากผู้ใช้งานจะถูกแสดงในรูปแบบที่อ่านง่าย พร้อมกับชื่อผู้ใช้งานและวันที่รีวิว
- ปุ่ม "ส่งคอมเมนต์": ปุ่มที่ผู้ใช้งานใช้สำหรับส่งความคิดเห็นและคะแนนรีวิวที่กรอกรระบบ

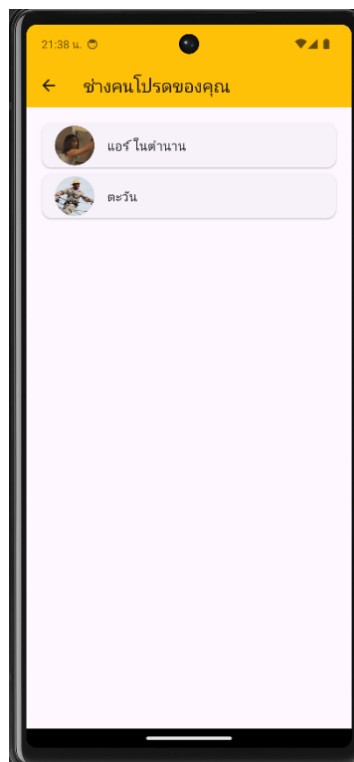


รูปที่ 3.6 หน้าคอมเมนต์ของแอปบริการช่างในพื้นที่

3.1.7 หน้ารายการโปรด

หน้ารายการโปรดในแอปพลิเคชัน Craftlocal ถูกออกแบบมาเพื่อให้ผู้ใช้งานสามารถบันทึกช่างหรือบริการที่ชื่นชอบไว้ในรายการโปรด เพื่อสะดวกในการค้นหาและจองบริการในครั้งถัดไป โดยมีองค์ประกอบหลักดังนี้

- รายการช่างที่บันทึกไว้: แสดงรายชื่อช่างที่ผู้ใช้งานได้ทำการบันทึกไว้ในรายการโปรด พร้อมกับข้อมูลพื้นฐาน เช่น ชื่อ, รูปโปรไฟล์, และประเภทของบริการ
- สามารถดูรายละเอียดของช่างได้โดยกดที่รูปโปรไฟล์ช่าง: สามารถคลิกเพื่อดูรายละเอียดเพิ่มเติมของช่างในรายการโปรด เช่น ข้อมูลการติดต่อ, รีวิว, ที่อยู่



รูปที่ 3.7 หน้ารายการโปรดแอปบริการช่างในพื้นที่

3.1.8 หน้าการตั้งค่า

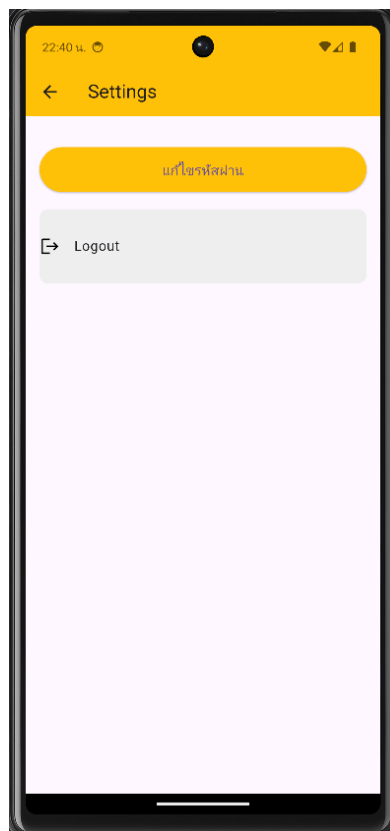
หน้าการตั้งค่าในแอปพลิเคชัน Craftlocal ถูกออกแบบมาเพื่อให้ผู้ใช้งานสามารถปรับเปลี่ยนการตั้งค่าบางประการ เช่น การเปลี่ยนรหัสผ่าน, การออกจากระบบ (ล็อกเอ้า) และการตั้งค่าอื่นๆ เพื่อความสะดวกในการใช้งาน โดยมีองค์ประกอบหลักดังนี้

เปลี่ยนรหัสผ่าน:

- ช่องกรอกรหัสผ่านใหม่: สำหรับกรอกรหัสผ่านใหม่ที่ต้องการตั้ง
- ช่องยืนยันรหัสผ่านใหม่: กรอกรหัสผ่านใหม่อีกครั้งเพื่อยืนยัน
- ปุ่ม "บันทึกการเปลี่ยนแปลง": ปุ่มที่ใช้ยืนยันการเปลี่ยนแปลงรหัสผ่าน

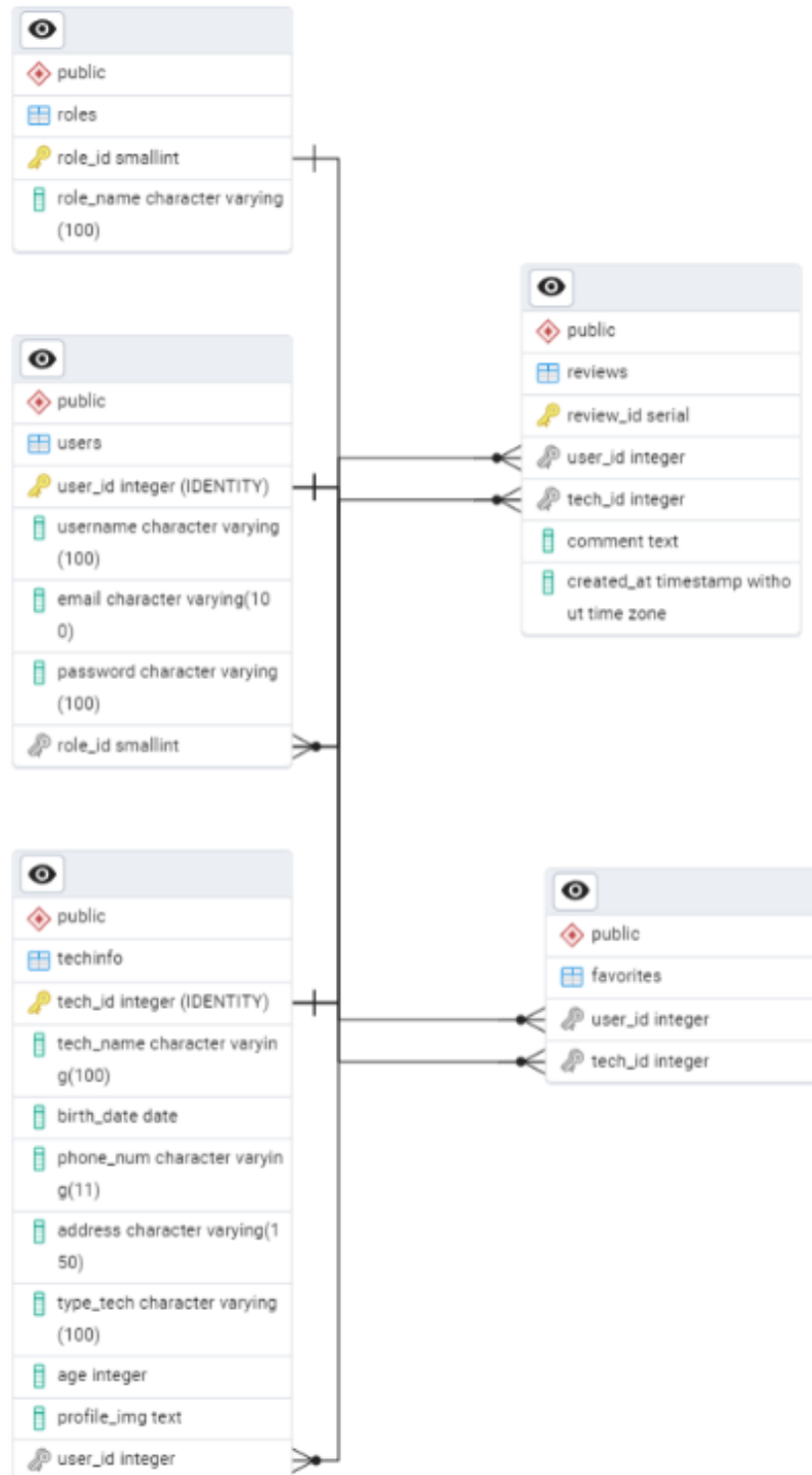
ล็อกเอ้า (ออกจากระบบ):

- ปุ่ม "ออกจากระบบ" ให้ผู้ใช้งานออกจากบัญชีของตัวเองและกลับไปยังหน้าหลักการเข้าสู่ระบบ



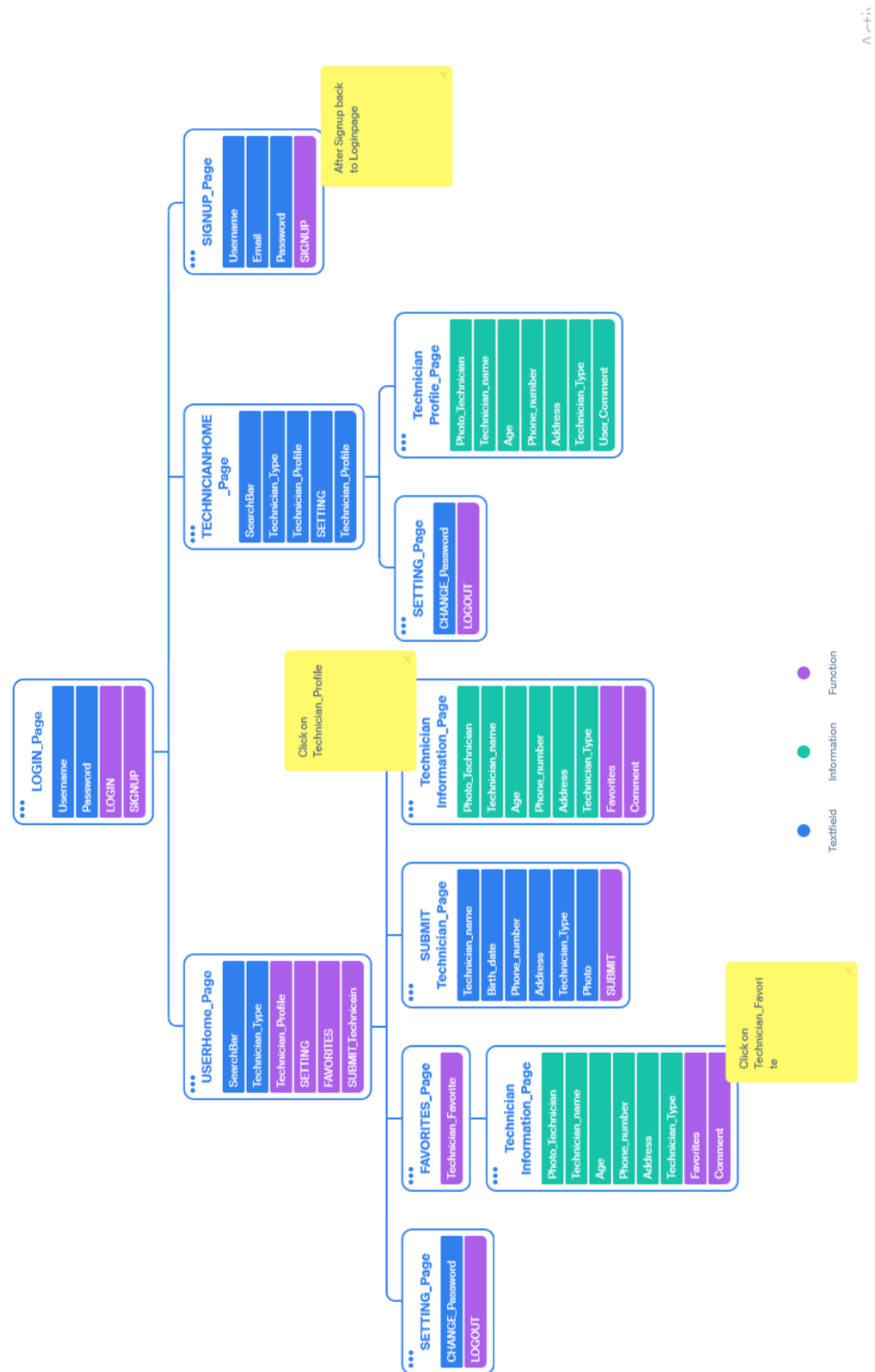
รูปที่ 3.8 หน้าการตั้งค่าของแอปบริการช่างในพื้นที่

3.2 Database Schema



รูปภาพที่ 3.9 ฐานข้อมูลของแอปบริการช่างในพื้นที่

3.3 Site Map Mobile Application



รูปที่ 3.10 แผนผังแอปพลิเคชัน

3.3 คู่มือการติดตั้ง APK 1.9.1

3.3.1 รันไฟล์ server.js เพื่อสร้างการเชื่อมต่อระหว่างแอปพลิเคชันกับฐานข้อมูล

3.3.2 เปิดการอนุญาตให้ติดตั้งแอปพลิเคชันที่ไม่รู้จัก ในการเปิดการอนุญาตให้ติดตั้งแอปพลิเคชันที่ไม่รู้จักมีขั้นตอนดังนี้ settings > Security and privacy > More security settings > Install unknown apps > กดเปิดการ ติดตั้งจาก Chrome

3.3.3 การดาวน์โหลดแอปพลิเคชัน กดที่ลิงค์นี้เพื่อดาวน์โหลดไฟล์.apk ของแอปพลิเคชัน

3.4 คู่มือการใช้งานแอปพลิเคชัน

3.4.1 หน้าแรกของแอปพลิเคชันการเข้าสู่ระบบและการสมัครสมาชิก

หมายเลข 1 กด “เข้าสู่ระบบ” เพื่อเข้าสู่ระบบแอปพลิเคชัน

หมายเลข 2 กด “สมัคร” เพื่อสมัครใช้งานแอปพลิเคชัน

หมายเลข 3 กรอกชื่อผู้ใช้งาน

หมายเลข 4 กรอกรหัสผ่าน



3.4.2 หน้าสมัครสมาชิก

หมายเลข 5 กดไอคอน <- เพื่อกลับไปยังหน้าจอหน้าก่อนหน้า

หมายเลข 6 กรอกชื่อ

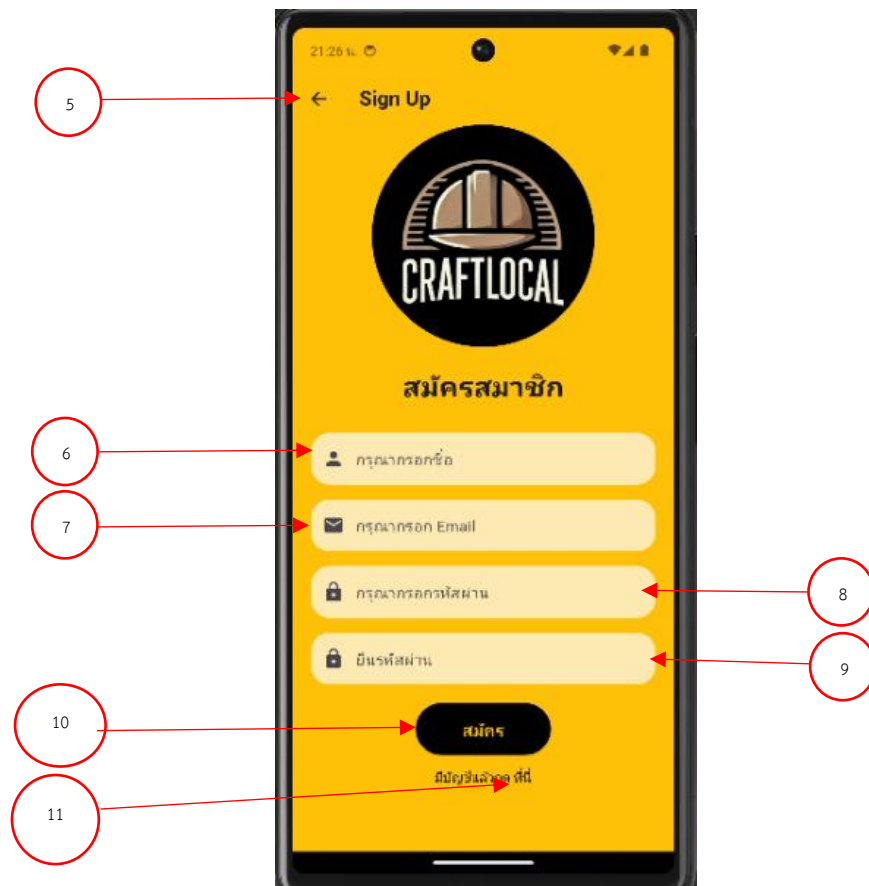
หมายเลข 7 กรอกอีเมล

หมายเลข 8 กรอกรหัสผ่าน

หมายเลข 9 กรอกยืนยันรหัสผ่านใหม่อีกครั้ง

หมายเลข 10 กรอกยืนยันสมัครสมาชิก

หมายเลข 11 หากมีบัญชีผู้ใช้งานอยู่แล้ว



3.4.3 หน้าหลักสำหรับผู้ใช้งาน

หมายเลข 12 กดรูปโปรไฟล์ช่าง เพื่อเข้าสู่ข้อมูลช่าง

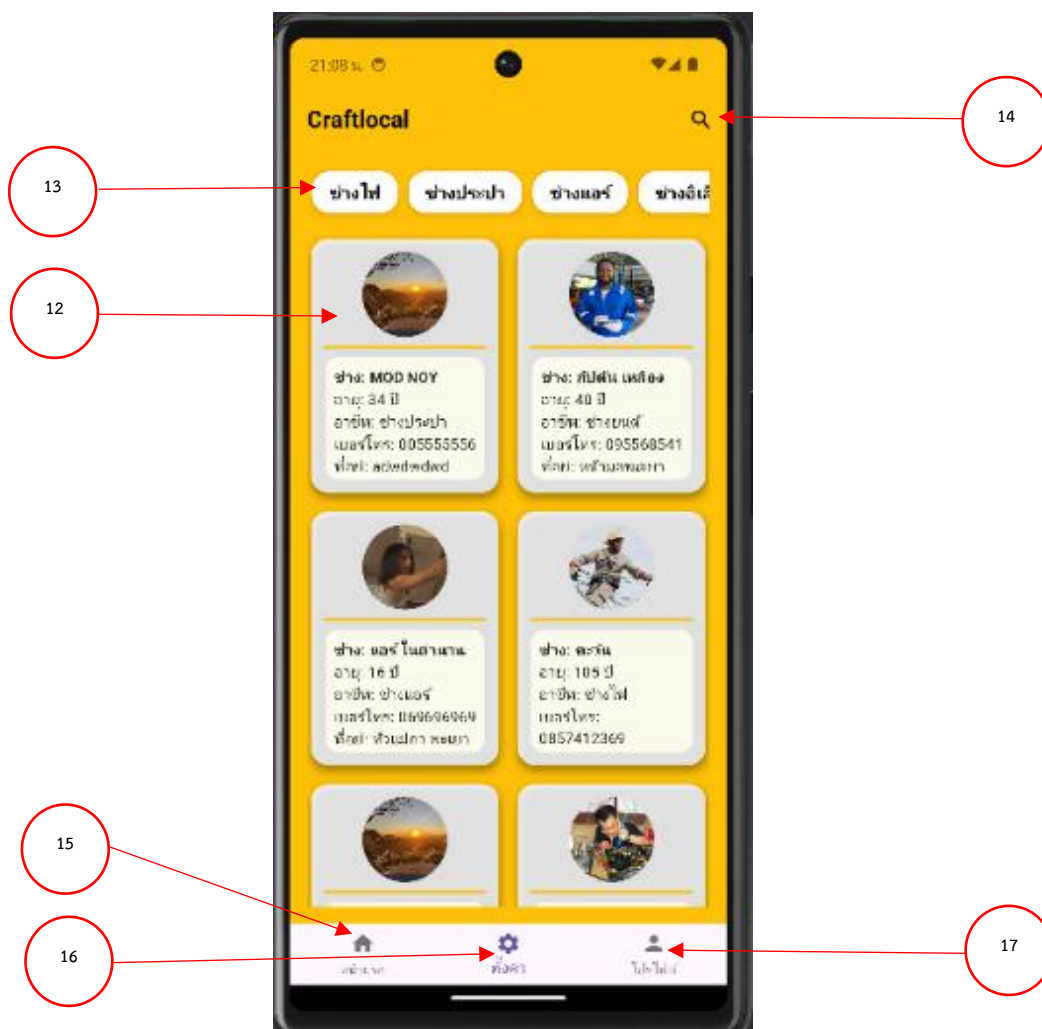
หมายเลข 13 กดช่องฟิลเตอร์ “ช่างไฟ” เพื่อแสดงเฉพาะช่างไฟเท่านั้น

หมายเลข 14 กดปุ่มค้นหา เพื่อค้นหาช่างในพื้นที่ หรือชื่อช่าง

หมายเลข 15 กดปุ่มโฮม เพื่อกลับมาหน้าแรก

หมายเลข 16 กดปุ่มตั้งค่า เพื่อตั้งค่าแอปพลิเคชัน

หมายเลข 17 กดปุ่มสมัครช่าง เพื่อสมัครการใช้งานเป็นช่าง

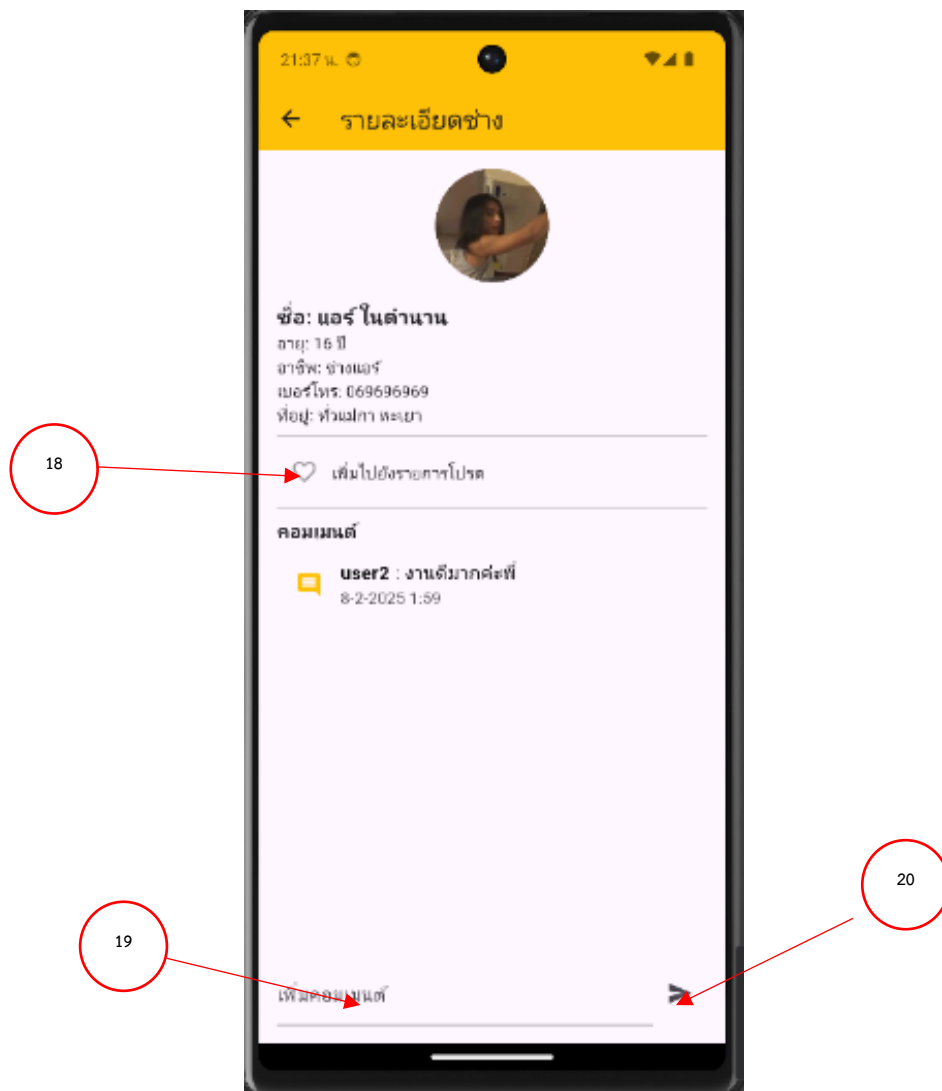


3.4.4 หน้ารายละเอียดช่างและคอมเมนต์รีวิวช่าง

หมายเลข 18 ปุ่มกดหัวใจ เพื่อเพิ่มไปยังรายการโปรด

หมายเลข 19 กดเพิ่มคอมเมนต์ เพื่อแสดงความคิดเห็นกับช่างคนนี้

หมายเลข 20 กดปุ่มเพื่อส่งคอมเมนต์



3.4.5 แบบฟอร์มสมัครช่าง

หมายเลข 21 กดปุ่มกล้อง เพื่ออัปโหลดโปรไฟล์ของช่าง

หมายเลข 22 กดปุ่มกรอกชื่อ เพื่อลงชื่อ

หมายเลข 23 กดปุ่มเลือกวันเกิด เพื่อกรอกวันเกิด

หมายเลข 24 กดปุ่มเลือกเบอร์ เพื่อกรอกเบอร์

หมายเลข 25 กดปุ่มกรอกที่อยู่

หมายเลข 26 กดปุ่มประเภทช่าง เพื่อกรอกประเภทของช่าง

21

22

23

24

25

26

21.30 น.

← แบบฟอร์มสมัครช่าง

คลิกเพื่อเพิ่มรูปโปรไฟล์

ชื่อจริง-นามสกุล

เลือกวันเกิด

เบอร์โทรศัพท์

ที่อยู่ (เช่น ถนน, หมู่บ้าน, ตำบล, อำเภอ, จังหวัด)

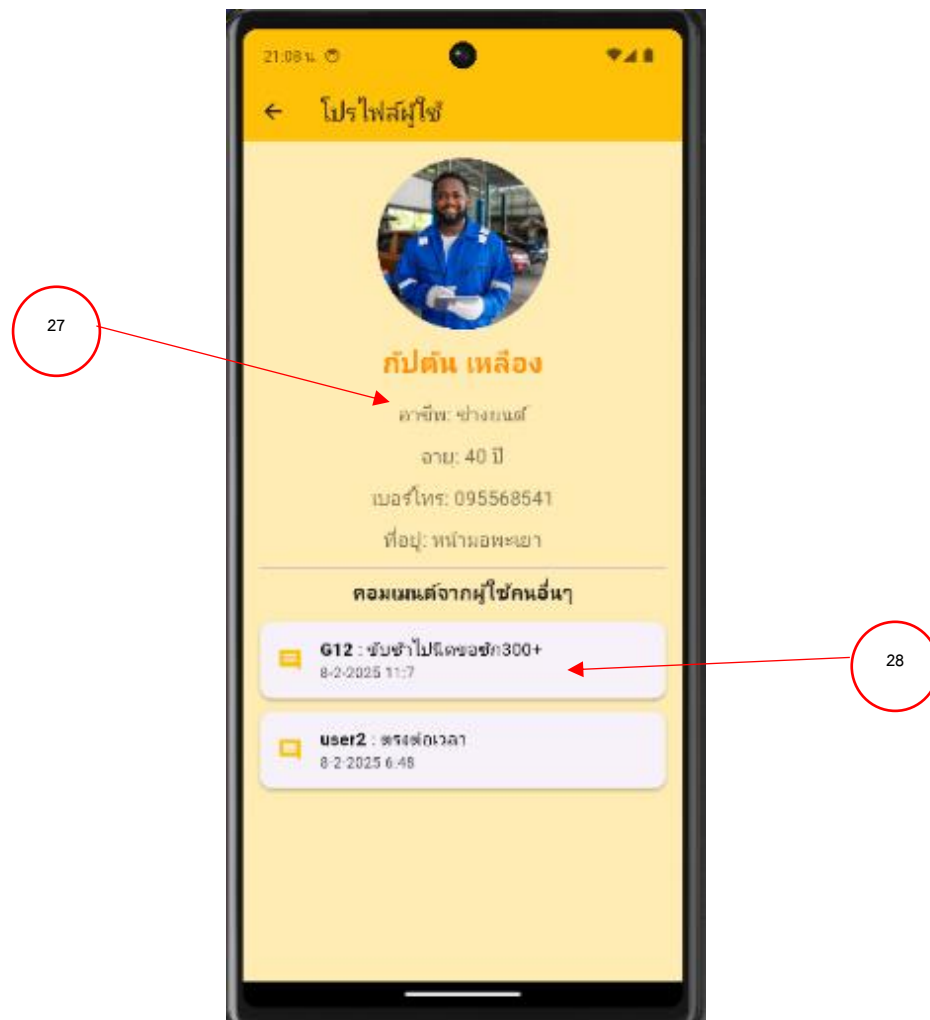
ประเภทช่าง

บันทึกข้อมูล

3.4.6 หน้าโปรไฟล์ช่าง

หมายเลข 27 แสดงชื่อ และข้อมูลช่าง

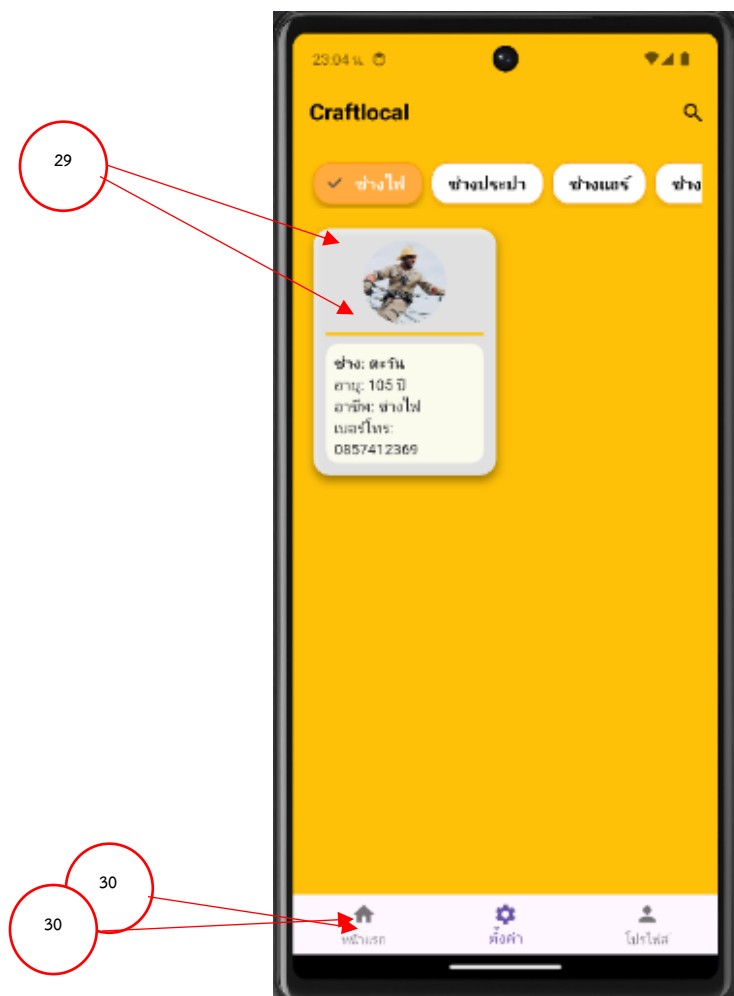
หมายเลข 28 แสดงคอมเมนต์



3.4.7 แสดงแค่ข้อมูลช่างไฟ

หมายเลข 29 กดช่องช่าง ตะวัน เพื่อดูรายละเอียด

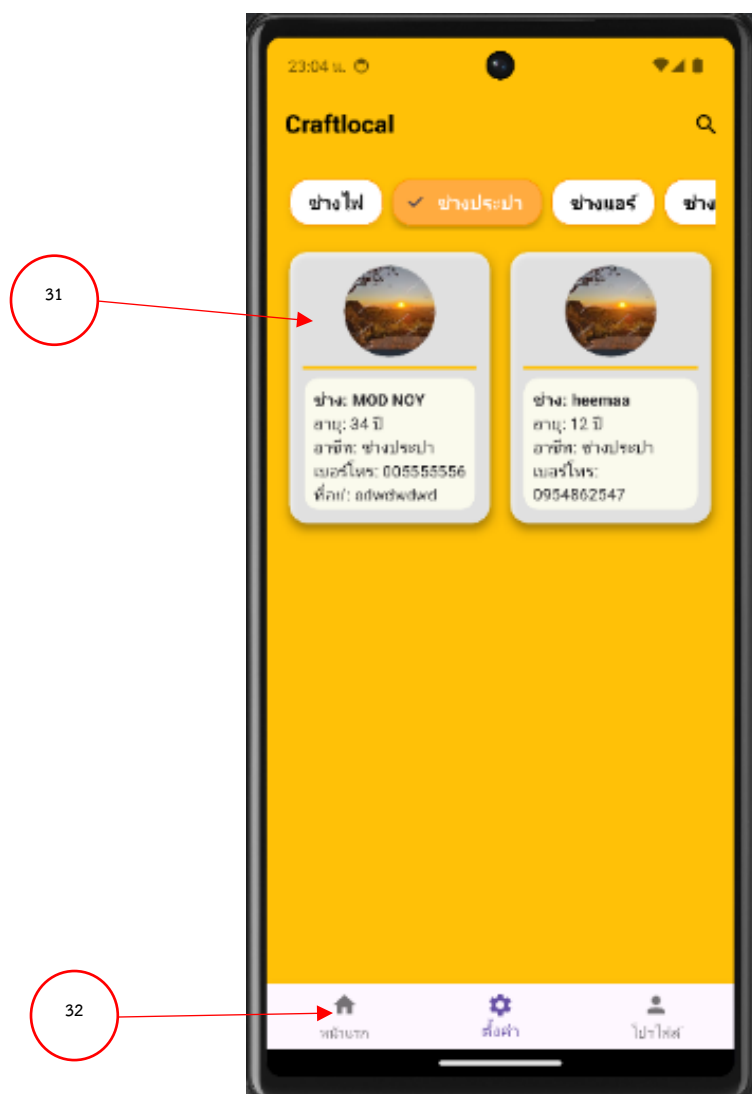
หมายเลข 30 กดปุ่มโฮม เพื่อกลับมาหน้าแรก



3.4.8 แสดงแค่ข้อมูลช่างประปา

หมายเลข 31 กดช่องช่าง MOD NOY เพื่อดูรายละเอียด

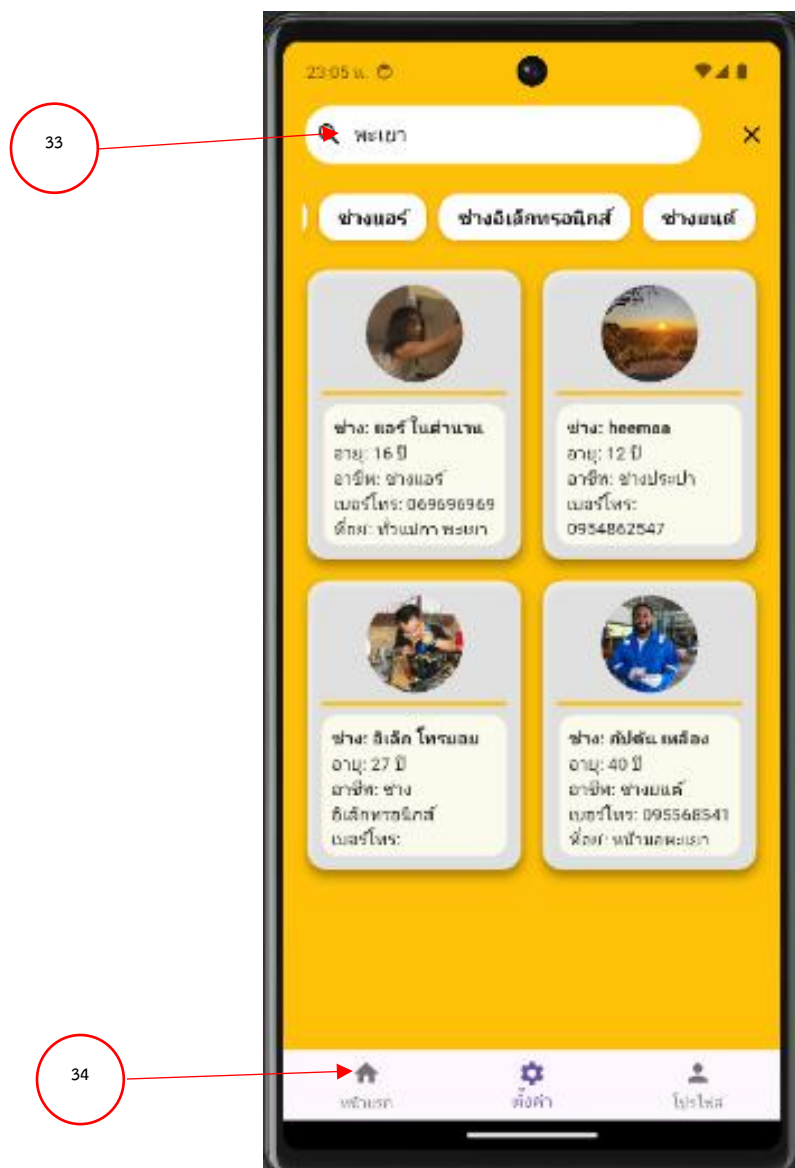
หมายเลข 32 กดปุ่มโฮม เพื่อกลับมาหน้าแรก



3.4.9 แสดงแค่ข้อมูลช่างในพื้นที่ที่ค้นหา

หมายเลข 33 กดช่องค้นหา เพื่อค้นหาช่างในพื้นที่ หรือชื่อช่าง

หมายเลข 34 กดปุ่มโฮม เพื่อกลับมาหน้าแรก



3.5 สรุปผลโครงการ

แอปพลิเคชันรวบรวมข้อมูลช่างในพื้นที่ (CraftLocal) ได้รับการพัฒนาเพื่อตอบสนองความต้องการของผู้ใช้งานในการค้นหาข้อมูลช่างในพื้นที่ที่ต้องการอย่างเป็นระบบ ซึ่งแอปพลิเคชันนี้ได้รับการออกแบบให้สามารถบันทึกข้อมูลช่าง ข้อมูลการรีวิว ข้อมูลผู้ใช้งาน และค้นหาข้อมูลช่างได้

3.5.1. วัตถุประสงค์ของโครงการ

- เพื่อสร้างแอปพลิเคชันที่สามารถรวบรวมข้อมูลของช่างในพื้นที่ต่างๆ โดยให้ผู้ใช้งานสามารถสมัครสมาชิกและค้นหาข้อมูลช่างตามประเภทที่ต้องการ
- ช่วยให้ผู้ใช้สามารถเข้าถึงข้อมูลช่างที่มีประสิทธิภาพและคุณสมบัติตามต้องการได้สะดวกยิ่งขึ้น

3.5.2. ฟีเจอร์หลัก

- การสมัครสมาชิก : ผู้ใช้สามารถเลือกสมัครเป็นสมาชิกทั่วไปเพื่อใช้งาน หรือสมัครเป็นช่างโดยกรอกข้อมูลเช่น ชื่อ วันเกิด เบอร์มือถือ ที่อยู่ ประเภทช่าง และอัปโหลดรูปภาพ
- หน้าเมนูการค้นหาช่าง : ผู้ใช้งานสามารถค้นหาข้อมูลช่างจากฟอร์มที่ผู้ใช้กรอก เช่น ประเภทช่าง ที่อยู่ หรือคำค้นหาต่างๆ
- การจัดเก็บข้อมูล : ข้อมูลทั้งหมดจะถูกจัดเก็บในฐานข้อมูล PostgreSQL ผ่าน Backend API ที่พัฒนาโดยใช้ Node.js

3.5.3. เทคโนโลยีที่ใช้

- Frontend: Flutter สำหรับสร้าง UI ที่รองรับทั้ง Android และ iOS
- Backend: Node.js ใช้สำหรับเชื่อมต่อกับฐานข้อมูล PostgreSQL
- Database: PostgreSQL สำหรับเก็บข้อมูลผู้ใช้งานและข้อมูลของช่าง

3.5.4. ผลลัพธ์ที่ได้

- แอปพลิเคชันสามารถรองรับการสมัครสมาชิกและการค้นหาช่างในพื้นที่ได้อย่างมีประสิทธิภาพ
- ผู้ใช้สามารถเข้าถึงข้อมูลช่างในพื้นที่ใกล้เคียงได้ง่าย
- ข้อมูลถูกจัดเก็บอย่างปลอดภัยและสามารถจัดการได้ผ่าน Backend API
-

3.5.5 สรุปภาพรวมของโครงการ

จากการดำเนินโครงการพบว่าแอปพลิเคชัน CraftLocal สามารถตอบโจทย์ความต้องการของผู้ใช้งานในการบันทึกข้อมูลช่างและช่วยส่งเสริมอาชีพช่าง ทำให้สามารถนำข้อมูลไปใช้ประกอบในการแก้ปัญหาด้านการซ่อมแซมได้

3.6 รายละเอียด ความรับผิดชอบแต่ละบุคคลในทีม

รหัสனிสิต	ชื่อ - สกุล	รายละเอียดความรับผิดชอบ
65021699	นายธนพล ยะใหม่วงศ์	1. ออกแบบ UX/UI 2. back-end 3. Fronend 4. เขียน Report
65021879	นายปัญชรัสมิ ชำนาญยา	1. ออกแบบ UX/UI 2. Fronend 3. tester 4. เขียน Report
65021880	นายเป็นหนึ่ง สายทรัพย์	1. ออกแบบ UX/UI 2. back-end 3. tester 4. เขียน Report