

ANSPART: แอปพลิเคชันจัดการสินค้าคงคลังสำหรับธุรกิจซ่อมรถจักรยานยนต์ อนันต์สิทธิมอเตอร์

ANSPART: INVENTORY MANAGEMENT APPLICATION FOR BIKE REPAIR SHOPS ANANSIT MOTOR

จารุวัฒน์ สีนิน¹

ผศ. ดร. วิลาวัลย์ อินทร์ชำนาญ²

บทคัดย่อ

จากปัญหาการจัดการสินค้าคงคลังหรือสินค้าคงเหลือซึ่งสิ่งจำเป็นสำหรับธุรกิจซ่อมบำรุงและดูแลรักษาจักรยานยนต์เป็นอย่างมาก เพื่อให้ธุรกิจเป็นไปอย่างราบรื่นในการซ่อมรถจักรยานยนต์ที่ไม่ขาดตอน ธุรกิจก็ต้องจัดเก็บอะไหล่เพื่อใช้ในการซ่อมจักรยานยนต์ และยังคงเก็บอะไหล่ไว้เพื่อขายด้วย สินค้าสำเร็จรูปเหล่านี้ เราเรียกว่าสินค้าคงคลังหรือสินค้าคงเหลือนั่นเอง กิจกรรมซ่อมบำรุงและดูแลรักษาจักรยานยนต์ที่มีการจัดเก็บอะไหล่ จำนวนมากก็เป็นปัญหาทั้งสถานที่เก็บ และยังเป็นปัญหาที่ต้องใช้เงินทุนหมุนเวียนจำนวนมากอีกด้วยการบริหารสินค้าคงคลังให้เหมาะสมจึงเป็นสิ่งที่ธุรกิจต้องให้ความสำคัญจัดการคลังสินค้าให้ดีเพื่อให้มีสินค้าพอต่อการใช้งาน และมีต้นทุนการเงินที่ต่ำ ANSPART จึงถูกจัดทำขึ้นมา มีวัตถุประสงค์เพื่อพัฒนาแอปพลิเคชันจัดการสินค้าคงคลังสำหรับธุรกิจซ่อมบำรุงและรักษารถจักรยานยนต์บนระบบปฏิบัติการแอนดรอยด์ ให้สามารถบันทึกข้อมูล การซ่อมรถจักรยานยนต์ ข้อมูลการสั่งซื้ออะไหล่ และนำข้อมูลที่บันทึกมาวิเคราะห์ในการสั่งซื้ออะไหล่ในครั้งต่อไปได้อย่างมีประสิทธิภาพ การค้นหาอะไหล่ได้รวดเร็วและมีข้อมูลเป็นปัจจุบัน จากการวิจัยพบว่าผู้ใช้ระบบมีความพึงพอใจโดยมีค่าเฉลี่ยเท่ากับ 4.43 ค่าเบี่ยงเบนมาตรฐานเท่ากับ 0.64

คำสำคัญ: สินค้าคงคลัง, แอปพลิเคชันจัดการสินค้าคงคลัง, ระบบปฏิบัติการแอนดรอยด์, การบริหารสินค้าคงคลัง, แอปพลิเคชัน

ABSTRACT

From the issue of inventory management, which is crucial for motorcycle repair and maintenance businesses, ensuring the smooth operation of repairs requires the business to stock spare parts for use in

¹ นักศึกษาด้านวิศวกรรมเว็บฯ

² ที่ปรึกษาสารนิพนธ์หลัก

repairs and for potential sales. These finished goods are known as inventory or stock. Motorcycle repair and maintenance businesses that store large quantities of spare parts face challenges regarding storage space and require significant working capital. Proper inventory management is thus essential for businesses to focus on. Efficient warehouse management ensures that there are sufficient parts for use while maintaining low financial costs.

ANSPART was developed with the aim of creating an inventory management application specifically for motorcycle repair and maintenance businesses on the Android operating system. The application allows for the recording of motorcycle repair data, spare part orders, and the use of recorded data to efficiently analyze future spare part orders. It provides quick access to spare parts and ensures the information is up to date. Research shows that users of the system reported satisfaction with an average score of 4.43 and a standard deviation of 0.64.

Keywords: Inventory, Inventory Management Application, Android Operating System, Inventory Management, Application

1. บทนำ

1.1 ความเป็นมาและความสำคัญ

สินค้าคงคลังหรือสินค้าคงเหลือเป็นสิ่งจำเป็นสำหรับธุรกิจซ่อมบำรุงและดูแลรักษาจักรยานยนต์เป็นอย่างมาก เพื่อให้ธุรกิจเป็นไปอย่างราบรื่นในการซ่อมจักรยานยนต์ที่ไม่ขาดตอน ธุรกิจก็ต้องจัดเก็บอะไหล่ เพื่อใช้ในการซ่อมจักรยานยนต์และยังต้องเก็บอะไหล่ไว้เผื่อขายด้วย สินค้าสำเร็จรูปเหล่านี้เราวมเรียกว่าสินค้าคงคลังหรือสินค้าคงเหลือนั่นเอง

ธุรกิจซ่อมบำรุงและดูแลรักษาจักรยานยนต์ที่มีการจัดเก็บอะไหล่จำนวนมาก มักจะมีปัญหาทั้งสถานที่เก็บ การค้นหาอะไหล่ การจดจำข้อมูลอะไหล่ที่ใช้กับรถจักรยานยนต์ การใช้อะไหล่ที่ไม่มีการจดบันทึกเมื่ออะไหล่ถูกใช้จนหมดการซ่อมรถจักรยานยนต์ครั้งต่อไปจะไม่มีอะไหล่สำรองไว้ใช้งาน เนื่องจากอะไหล่ไม่ได้ถูกสั่งซื้อเข้ามา และไม่รู้ว่าจะอะไหล่ชิ้นไหนได้ถูกใช้ไปจนหมด และปัญหาที่ต้องใช้เงินทุนหมุนเวียนจำนวนมากอีกด้วย การบริหารสินค้าคงคลังให้เหมาะสมจึงเป็นสิ่งที่ธุรกิจต้องบริหารจัดการให้ดี เพื่อให้มีสินค้าขายและมีต้นทุนการเงินที่ต่ำ

จากปัญหาที่ได้กล่าวมา ผู้พัฒนาจึงพัฒนา ANSPart : แอปพลิเคชันจัดการสินค้าคงคลังสำหรับธุรกิจซ่อมรถจักรยานยนต์อัตโนมัติเพื่อเพิ่มความสามารถจัดการสินค้าคงคลัง ช่วยให้ธุรกิจซ่อมบำรุงและดูแลรักษาจักรยานยนต์ทำงานได้ง่ายขึ้นด้วยการช่วยค้นหาข้อมูลอะไหล่ได้เร็วขึ้น บันทึกข้อมูลการสั่งซื้ออะไหล่ นำระบบ Barcode มาใช้เพื่อเพิ่มประสิทธิภาพการทำงานในคลังสินค้าให้สะดวก ลดความ

ผิดพลาดจากการทำงาน และ บันทึกข้อมูลการซ่อมรถจักรยานยนต์ ส่วนข้อมูลที่ได้จากการบันทึกก็สามารถนำไปวิเคราะห์ เพื่อพัฒนาการดำเนินธุรกิจได้อีกด้วย

1.2 ปัญหา

- 1.2.1 ธุรกิจซ่อมบำรุงและดูแลรักษาจักรยานยนต์ที่มีการจัดเก็บอะไหล่จำนวนมาก มักจะมีปัญหาการค้นหอะไหล่
- 1.2.2 ข้อจำกัดในการจดจำข้อมูลอะไหล่ที่ใช้กับรถจักรยานยนต์ที่เข้ามาใช้บริการ
- 1.2.3 การใช้อะไหล่ที่ไม่มีการจดบันทึกเมื่ออะไหล่ถูกใช้จนหมด การซ่อมรถจักรยานยนต์ครั้งต่อไปจะไม่มีอะไหล่สำรองไว้ใช้งาน

1.3 วัตถุประสงค์

- 1.3.1 พัฒนาเครื่องมือเพื่อช่วยในการค้นหาข้อมูลอะไหล่ในธุรกิจซ่อมรถจักรยานยนต์
- 1.3.2 พัฒนาเครื่องมือเพื่อช่วยในการบันทึกข้อมูลการซ่อมในธุรกิจซ่อมรถจักรยานยนต์
- 1.3.3 พัฒนาเครื่องมือเพื่อช่วยในการบันทึกข้อมูลการสั่งซื้ออะไหล่ในธุรกิจซ่อมรถจักรยานยนต์

2. แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

การออกแบบและการพัฒนาแอปพลิเคชัน ผู้พัฒนาได้นำ ทฤษฎี และผลงานวิจัยที่เกี่ยวข้องกับการพัฒนาแอปพลิเคชัน เพื่อใช้เป็นแนวทางในการพัฒนา ดังนี้

2.1 ศึกษากระบวนการทำงานและปัญหาในการทำงาน

จากการศึกษากระบวนการทำงานของร้านซ่อมบำรุงและดูแลรักษารถจักรยานยนต์อันันต์สิทธิ์มอเตอร์ มีการทำงานทั้งหมด ดังนี้

2.1.1 การจัดเก็บอะไหล่ อะไหล่สำรองของร้านมีจำนวนมาก การจัดเก็บอะไหล่ใช้ประสิทธิภาพของการใช้ งานเพื่อเอาไปซ่อมบำรุงรักษารถจักรยานยนต์ เช่น อะไหล่ที่ใช้ด้วยกันบ่อยจะจัดเก็บใกล้กัน เพื่อต่อการนำไปใช้งาน แต่การนำอะไหล่ไปใช้งานแต่ละครั้งไม่มีการจดบันทึก ทำให้บางครั้งอะไหล่ที่ต้องใช้ซ่อมบำรุงรถ จักรยานยนต์ไม่เพียงพอและเสียโอกาสในการทำงาน

2.1.2 การสั่งซื้ออะไหล่ เมื่อพบว่าอะไหล่ที่เก็บไว้ใกล้จะหมด จะจัดห่ออะไหล่ไว้ในสมุด โดยสมุดจะแยก ไปตามยี่ห้อของอะไหล่ หรือบริษัทที่จำหน่ายอะไหล่ ต่อมาก็จะส่งข้อมูลโดยการถ่ายรูปของรายการอะไหล่ที่จดบันทึกไว้แล้วส่งไปให้ร้านขายอะไหล่ โดยช่องทางติดต่อสื่อสารที่ทางร้าน ขายอะไหล่ กำหนด เช่น Line หรือ Facebook

2.1.3 การซื้ออะไหล่เข้ามาจัดเก็บ หลังจากซื้ออะไหล่หรือว่าร้านขายอะไหล่ นำอะไหล่ที่สั่งซื้อไว้ มาส่ง จะได้รับใบเสร็จ และรายการอะไหล่ที่เข้ามา ดังภาพที่ 2.3 ทางร้านซ่อมบำรุงรักษารถจักรยานยนต์ ก็จะตรวจสอบความถูกต้องของข้อมูลในใบเสร็จกับอะไหล่ที่เข้ามา หลังจากข้อมูลในใบเสร็จและอะไหล่ที่เข้าตรงกัน ต่อไปก็จะไปตรวจสอบข้อมูลอะไหล่ที่เข้ามากับรายการอะไหล่ที่สั่งและส่งไปให้ร้านขาย

อะไหล่เพื่อตรวจสอบความถูกต้องว่า อะไหล่ที่เข้ามามีจำนวนตรงกับที่ส่งไปหรือไม่หลังจากนั้นก็จะเข้าสู่กระบวนการจัดเก็บอะไหล่

2.2 ศึกษาเทคโนโลยีที่ใช้ในการพัฒนาแอปพลิเคชัน

การพัฒนาแอปพลิเคชัน ผู้พัฒนาได้วิเคราะห์และตรวจสอบว่ามีเทคโนโลยีที่มาช่วยในการพัฒนาแอปพลิเคชันทำให้แอปพลิเคชันสนับสนุนให้ผู้ใช้งานได้ทำงานได้สะดวกมากขึ้น โดยมีเทคโนโลยีที่ศึกษาเพื่อใช้ในการพัฒนา ดังนี้

2.2.1 ภาษาในการพัฒนาโปรแกรม

1. React Native [1] เป็นเฟรมเวิร์กสำหรับการพัฒนาแอปพลิเคชันโมบาย ซึ่งถูกพัฒนาโดย Facebook และเปิดตัวครั้งแรกในปี 2015 ซึ่งเป็นเครื่องมือที่ช่วยในการสร้างแอปพลิเคชันโมบายที่สามารถทำงานได้ทั้งบนระบบปฏิบัติการ Android และ iOS โดยใช้ภาษา JavaScript ร่วมกับ React ซึ่งเป็นเฟรมเวิร์กสำหรับสร้าง UI (User Interface) โดยจะใช้โครงสร้างของ React ในการสร้างส่วนต่าง ๆ ของแอปพลิเคชัน ทำให้สามารถสร้าง UI ที่สวยงามและใช้งานได้ง่ายขึ้น

2. Swift [2] เป็นภาษาโปรแกรมมิ่งที่พัฒนาโดย Apple Inc. และถูกออกแบบมาเพื่อใช้ในการพัฒนาแอปพลิเคชันบนแพลตฟอร์ม iOS, macOS, watchOS, และ tvOS. Swift ถูกเปิดเผยเป็นครั้งแรกในงาน Apple's Worldwide Developers Conference (WWDC) ในปี 2014 และเป็นภาษาที่มีการออกแบบให้มีประสิทธิภาพสูง มีความปลอดภัย และใช้งานง่ายต่อการเขียนและอ่านโค้ด

3. Kotlin [3] เป็นภาษาโปรแกรมมิ่งที่มีการพัฒนาโดย JetBrains ซึ่งเปิดตัวครั้งแรกในปี 2011 และได้รับการเปิดเผยให้เป็นภาษาโอเพนซอร์สในปี 2012 ภายหลัง Google ได้เลือก Kotlin เป็นภาษาที่สนับสนุนในการพัฒนาแอปพลิเคชันบนแพลตฟอร์ม Android ในปี 2017

4. Flutter [4] เป็นเฟรมเวิร์กสำหรับการพัฒนาแอปพลิเคชัน โมบายที่ถูกพัฒนาโดย Google ซึ่งเป็นเครื่องมือที่มุ่งเน้นไปที่การสร้าง UI (User Interface) ที่สวยงามและเร็ว โดยใช้งานได้ทั้งบนแพลตฟอร์ม Android และ iOS ด้วยโค้ดเดียวกัน โดยใช้ Dart เป็นภาษาหลักในการเขียนโค้ด

ตารางที่ 2.1 ตารางเปรียบเทียบภาษาที่ใช้ในการพัฒนาโปรแกรม

ลักษณะ	React Native	Swift	Kotlin	Dart
ประสิทธิภาพ	ปานกลาง	สูง	สูง	สูง
การรองรับอุปกรณ์และระบบปฏิบัติการ	Cross-platform	iOS เท่านั้น	Android เท่านั้น	Cross-platform
ความสามารถในการใช้โค้ดที่มีอยู่แล้ว	มีปัญหาเล็กน้อย	ไม่มีปัญหา	ต้องตั้งค่าเพื่อใช้งาน	ไม่มีปัญหา

การจัดการกับหน่วยความจำและแบตเตอรี่	ปานกลาง	สูง	สูง	ปานกลางถึงสูง
การพัฒนา Cross-platform	ทำได้	ทำไม่ได้	ทำไม่ได้	ทำได้
ความนิยมในชุมชนนักพัฒนา	สูง	สูง	สูง	กำลังเพิ่มขึ้น
ความเร็วการประมวลผล	ปานกลางถึงช้า	สูง	สูง	สูง
การบำรุงรักษา	ค่อนข้างง่าย	ง่าย	ง่าย	ง่าย

ผู้จัดทำได้เลือก Flutter ซึ่งมีประสิทธิภาพในการพัฒนาแอปพลิเคชันแบบ Cross-platform ที่สามารถพัฒนาใช้ได้ทั้งระบบปฏิบัติการ iOS และ Android ได้ในเพียงโค้ดเพียงชุดเดียว ซึ่งตอบโจทย์กับการพัฒนาแอปพลิเคชันที่มีผู้พัฒนาน้อยและสามารถขยายกลุ่มผู้ใช้งานได้หลายปฏิบัติการ และยังคงความเร็วในการประมวลผลมากกว่า React Native ที่มีการพัฒนาแอปพลิเคชันแบบ Cross-platform คล้ายกัน

2.2.2 เครื่องมือที่ใช้ในการบันทึกข้อมูล

1. Firebase: Firebase เป็นแพลตฟอร์มของ Google ที่มีการให้บริการฐานข้อมูลแบบ Real-time Database และ Cloud Firestore ซึ่งมีประสิทธิภาพสูงและสามารถใช้งานได้ง่าย มีความยืดหยุ่นในการปรับปรุงและปรับใช้ระบบ รวมถึงมีการรองรับการทำงานแบบ Offline ที่ดี

2. MySQL: MySQL เป็นระบบฐานข้อมูลแบบ Relational Database Management System (RDBMS) ที่มีความเสถียรและมีประสิทธิภาพสูง มีความยืดหยุ่นในการจัดการข้อมูลและสามารถใช้งานร่วมกับแอปพลิเคชันที่มีข้อมูลขนาดใหญ่ได้

3. MongoDB: MongoDB เป็นฐานข้อมูลแบบ NoSQL ที่เหมาะสำหรับการพัฒนาแอปพลิเคชันที่มีโครงสร้างข้อมูลที่ผิดเพี้ยนหรือเปลี่ยนแปลงบ่อยครั้ง มีความยืดหยุ่นสูงและสามารถเพิ่มข้อมูลได้

4. SQLite: SQLite เป็นฐานข้อมูลแบบฝังตัวที่เหมาะสมสำหรับแอปพลิเคชันที่มีขนาดเล็กหรือต้องการความเร็วในการเข้าถึงข้อมูล มีความเสถียรและง่ายต่อการจัดการ แต่มีข้อจำกัดในการใช้กับแอปพลิเคชันที่มีข้อมูลมากหรือมีการใช้งานพร้อมกันจำนวนมาก

ตารางที่ 2.2 ตารางเปรียบเทียบเครื่องมือในการบันทึกข้อมูลของระบบ

ลักษณะ	Firebase	MySQL	MongoDB	SQLite
ความยืดหยุ่น	มีความยืดหยุ่นในการปรับปรุง	มีความยืดหยุ่นในการจัดการข้อมูล	มีความยืดหยุ่นสูง	มีความยืดหยุ่นในการใช้งาน

	และปรับใช้ระบบ			และการจัดการข้อมูล
ประสิทธิภาพ	ประสิทธิภาพสูงและสามารถใช้งานได้ง่าย	ประสิทธิภาพสูง	ประสิทธิภาพสูง	ประสิทธิภาพในการใช้งานกับข้อมูลขนาดเล็กหรือกลาง
ความปลอดภัย	มีความปลอดภัยในระดับสูง	มีระบบการรักษาความปลอดภัยที่ดี	ระบบการรักษาความปลอดภัยที่มีความเข้มงวด	มีความปลอดภัยในการใช้งาน
ความเหมาะสมในการใช้งาน	เหมาะสำหรับแอปพลิเคชันที่ทำงานแบบ Real-time	เหมาะสำหรับแอปพลิเคชันที่มีโครงสร้างข้อมูลแบบเชิงสัมพันธ์	เหมาะสำหรับแอปพลิเคชันที่มีเปลี่ยนแปลงข้อมูลบ่อยครั้ง	เหมาะสำหรับแอปพลิเคชันที่มีข้อมูลขนาดเล็กหรือกลาง
ประเภทของฐานข้อมูล	NoSQL (Real-time Database)	Relational Database	NoSQL	SQL

ในส่วนของการบันทึกข้อมูลผู้จัดทำได้เลือก Firebase เป็นฐานข้อมูลในการบันทึกข้อมูล Firebase เป็นบริการฐานข้อมูลแบบ NoSQL บนคลาวด์ ซึ่งความสามารถในการจัดเก็บข้อมูล และจัดการข้อมูลแบบ Real-Time ซึ่งเหมาะกับการทำงานกับภาษาที่พัฒนาแอปพลิเคชันด้วย Flutter

3. ระเบียบวิธีวิจัย

การดำเนินการพัฒนา ANSpart : แอปพลิเคชันจัดการสินค้าคงคลังสำหรับธุรกิจซ่อมรถจักรยานยนต์อัตโนมัติ เพื่อให้การจัดการสินค้าคงคลัง มั่นใจได้ว่ามีอะไหล่ที่จะพร้อมใช้งานในปริมาณที่เหมาะสมในเวลาที่เหมาะสม มีวิธีการดำเนินการดังนี้

3.1 วิเคราะห์และออกแบบระบบ

จากขั้นตอนการศึกษาปัญหาและความต้องการของระบบ ผู้พัฒนาได้นำข้อมูลที่รับมาเข้าสู่ขั้นตอนการวิเคราะห์และออกแบบระบบ ดังนี้

3.1.1 บุคคลที่เกี่ยวข้องกับระบบ จากการวิเคราะห์ปัญหาของสำหรับธุรกิจซ่อมบำรุงและดูแลรักษาจักรยานยนต์พบว่า มีผู้เกี่ยวข้องกับการทำงานดังนี้

1. ผู้จัดการร้าน เป็นตำแหน่งที่จัดการสินค้าคงคลัง เพื่อให้มั่นใจได้ว่ามีอะไหล่ที่เหมาะสมจะพร้อมใช้งานในปริมาณที่เหมาะสม ปัญหาที่พบคือ การนำอะไหล่ออกไปซ่อมแต่ละครั้งไม่ได้มีการ

บันทึก ทำให้มีการตรวจสอบจำนวนคงเหลืออะไหล่ต้องทำบ่อย เพื่อให้อะไหล่ทุกชิ้นมีจำนวนเพียงพอในการซ่อมบำรุงรถจักรยานยนต์ ถ้าหากละเลยการตรวจสอบนี้จะเกิดปัญหาอะไหล่ที่จะเก็บจะไม่เพียงพอต่อการใช้งาน และการสั่งซื้ออะไหล่แต่ละครั้งได้มีการบันทึกข้อมูลลงในสมุด ทำให้ค้นหาข้อมูลการสั่งอะไหล่ใช้เวลานาน

2. พนักงานอะไหล่ เป็นตำแหน่งที่ส่งมอบอะไหล่ให้กับช่างซ่อมบำรุงรถจักรยานยนต์ และจัดเก็บอะไหล่ที่สั่งซื้อไว้ในคลังสินค้า ปัญหาที่พบคือ เมื่อช่างซ่อมบำรุงต้องการอะไหล่เพื่อไปซ่อมรถจักรยานยนต์ พนักงานอะไหล่จะนำอะไหล่ไปให้ การใช้เวลาในการค้นหาอะไหล่จะใช้เวลานาน และการจัดเก็บอะไหล่ที่ซื้อมาใช้เวลานาน เนื่องจากต้องไปตรวจสอบกับการสั่งอะไหล่ ที่ผู้จัดการร้านสั่งและบันทึกลงในสมุด และต้องหาสถานที่จัดเก็บของอะไหล่ชิ้นนั้นด้วย

3.2 การออกแบบเอกสารการพัฒนา

3.2.1 Use Case Diagram การออกแบบ Use Case Diagram ช่วยให้ผู้พัฒนาเข้าใจความต้องการของระบบอย่างชัดเจน โดย Use Case Diagram จะมองเป็นแผนผังที่แสดงฟังก์ชันหลักของระบบ การใช้งานของผู้ใช้ และการสื่อสารระหว่างผู้ใช้กับระบบที่กำลังพัฒนา.



ภาพที่ 3.1 ภาพ Use Case Diagram

3.2.2 Use Case Description จาก Use Case Diagram (ภาพที่ 3.1) สามารถสรุปและอธิบายเป็น Use case Description ที่สำคัญโดยทางผู้จัดทำยกตัวอย่าง 1 Use case จากทั้งหมด 9 Use case ดังตารางที่ 3.1 ได้ดังนี้

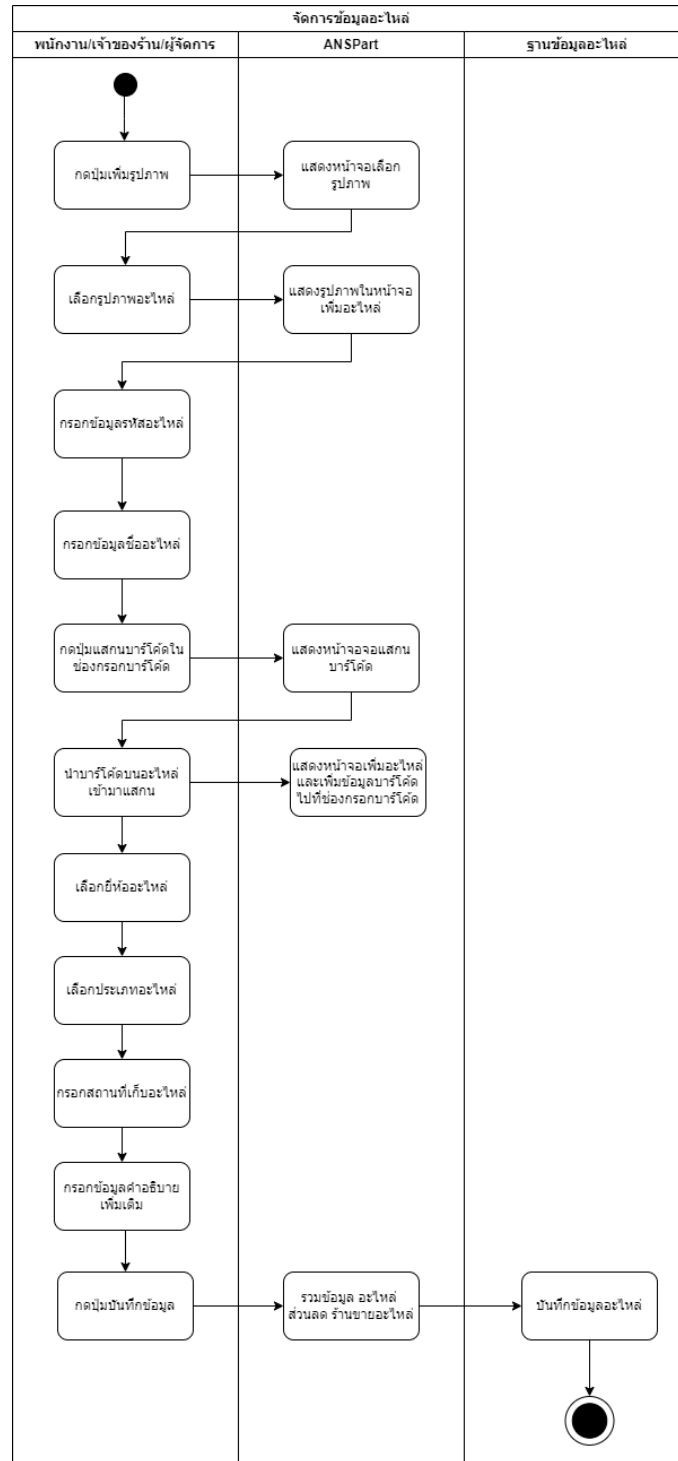
ตารางที่ 3.1 Use Case Description จัดการข้อมูลอะไหล่

Use Case ID	UC01
Use Case Name	จัดการข้อมูลอะไหล่
Primary Actor	เจ้าของร้าน/ผู้จัดการ/พนักงาน
Stakeholders and Interests	1. ผู้จัดการและเจ้าของร้าน: มีข้อมูลอะไหล่ใช้ในฟังก์ชันการสั่งอะไหล่ 2. พนักงาน: มีข้อมูลอะไหล่ในเปิดบิลซ่อม และการนำเข้าอะไหล่
Preconditions	เข้าสู่ระบบด้วยบัญชีที่มีสถานะเป็นเจ้าของร้านหรือผู้จัดการ
Postconditions	มีข้อมูลอะไหล่ที่ข้อมูลถูกต้องเพื่อใช้งานในฟังก์ชันอื่นในระบบ
Main Success Scenario (Basic Flow)	- กดไปที่เมนูข้อมูลอะไหล่ - กดปุ่มเครื่องหมาย + มุมล่างขวาเพื่อเพิ่มข้อมูลอะไหล่ - ระบบจําหน่ายไปที่หน้าจอเพิ่มอะไหล่ - เพิ่มรูปภาพอะไหล่จากคลังรูปภาพของคุณ - กรอกข้อมูลรหัสสินค้า - กรอกชื่อสินค้า - สแกนบาร์โค้ดอะไหล่ (ถ้ามี) - เลือกยี่ห้อของอะไหล่ - เลือกประเภทอะไหล่ - กรอกสถานที่เก็บ (ตัวอักษรสองตัวแรกจะกรอกอัตโนมัติเมื่อเลือกประเภทอะไหล่) - กรอกคำอธิบายเพิ่มเติม - กดที่ปุ่ม บันทึก ด้านบนมุมขวา - ระบบจะแสดงรายละเอียดของอะไหล่ที่สร้าง
Alternative Flows (Extensions)	1. ไม่กรอกข้อมูลช่อง ชื่ออะไหล่

	ระบบทำการแจ้งเตือน “กรณารอกชื่ออะไหล่” 2. ไม่กรอกข้อมูลช่อง รหัสอะไหล่ ระบบทำการแจ้งเตือน “กรณารอกรหัสอะไหล่” 3. ไม่กรอกข้อมูลช่อง ตำแหน่งจัดเก็บ ระบบทำการแจ้งเตือน “กรณารอกตำแหน่งจัดเก็บ” 4. ไม่เลือกประเภทอะไหล่ 5. ระบบทำการแจ้งเตือน “กรณาเลือกประเภทอะไหล่” 6. ไม่เลือกยี่ห้ออะไหล่ ระบบทำการแจ้งเตือน “กรณาเลือกยี่ห้อ”
Special Requirements	-
Technology and Data Variations List	ระบบแอสกนบาร์โค้ด
Frequency of Occurrence	1. ช่วงแรก: เกิดขึ้นบ่อยเพราะต้องบันทึกข้อมูลอะไหล่ลงระบบ 2. ช่วงหลัง: จะเกิดขึ้นตอนมีอะไหล่ใหม่เข้ามา
Open Issues	-

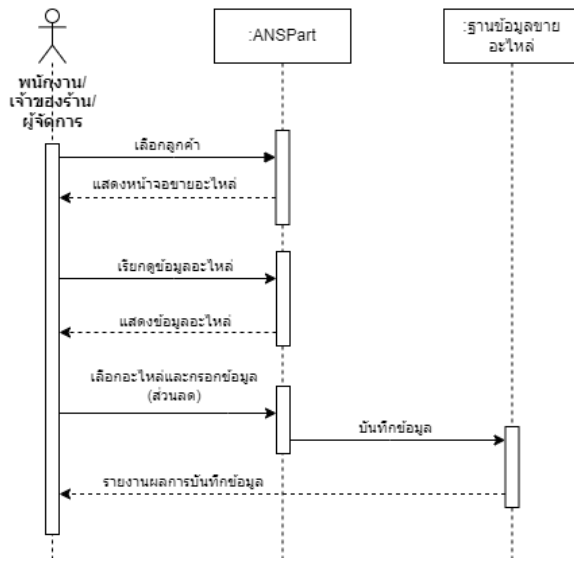
3.2.3 Activity Diagram

นำ Use Case Diagram ไปสร้างเป็น Activity Diagram จะช่วยให้เข้าใจการทำงานของระบบได้อย่างละเอียดและชัดเจนมากยิ่งขึ้น โดยจะช่วยให้เห็นได้ชัดเจนว่ากระบวนการทำงานมีขั้นตอนอย่างไรบ้าง



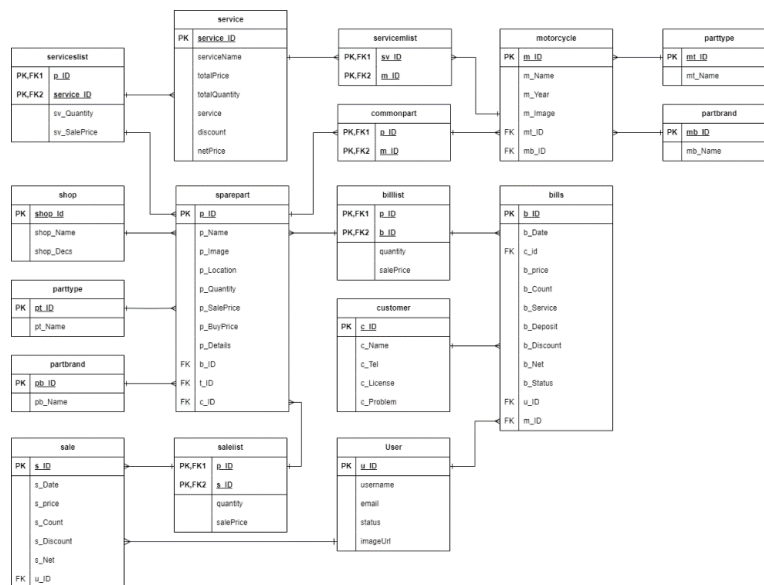
ภาพที่ 3.2 ภาพ Activity Diagram จัดการข้อมูลอะไหล่

3.2.4 Sequence Diagram เป็นแผนภาพที่แสดงลำดับขั้นตอนของการโต้ตอบระหว่างอ็อบเจกต์ (Objects) ในระบบ หรือการสื่อสารระหว่างคอมพิวเตอร์ในแอปพลิเคชัน โดยเฉพาะอย่างยิ่งในระหว่างการเรียกใช้งานเมธอด (Method) ของอ็อบเจกต์ต่าง ๆ ที่เกี่ยวข้องกัน โดยจะช่วยให้เห็นการปฏิสัมพันธ์กันแต่ละขั้นตอน ว่ามีการส่งข้อมูลอะไรถึงกัน โดยแสดงเรียงลำดับตามลำดับเวลา

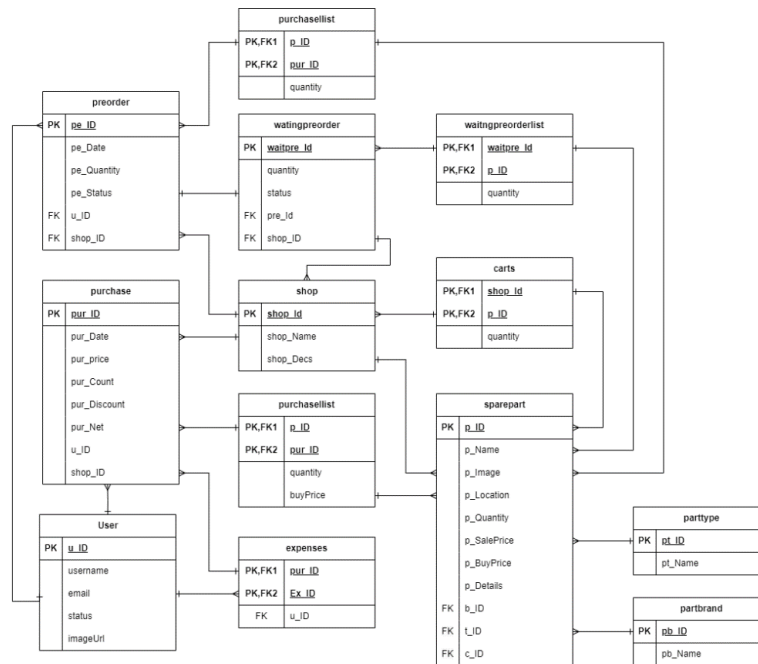


ภาพที่ 3.3 ภาพ Sequence Diagram จัดการข้อมูลอะไร่

3.2.5 Entity Relationship Diagram ในการจัดทำระบบฐานข้อมูลสามารถช่วยให้ผู้ใช้เข้าใจและวางแผนการจัดเก็บข้อมูลได้อย่างมีประสิทธิภาพ โดยให้เข้าใจความสัมพันธ์ระหว่าง Entity ต่าง ๆ ในระบบ

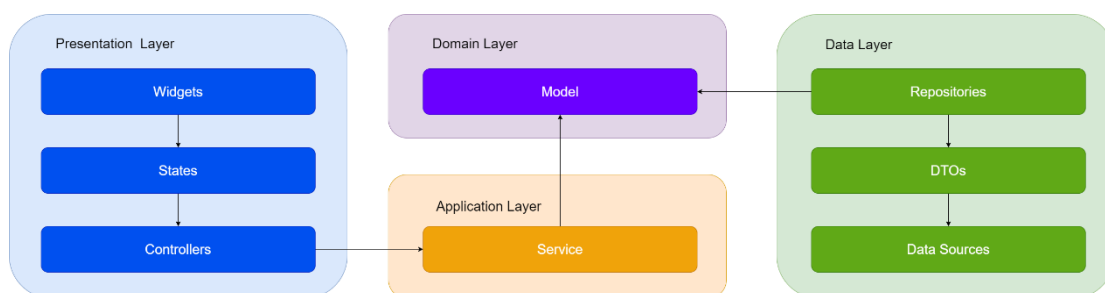


ภาพที่ 3.4 ภาพ ER Diagram การขายอะไหล่และการเปิดบิลซ่อม



ภาพที่ 3.5 ภาพ ER Diagram การนำเข้าอะไหล่,การสั่งอะไหล่และการชำระเงิน

3.2.6 สถาปัตยกรรมของระบบ (System Architecture) สถาปัตยกรรมระบบของแอปพลิเคชันจัดการสินค้าคงคลังสำหรับธุรกิจซ่อมบำรุงและดูแลรักษาจักรยานยนต์ แสดงให้เห็นการทำงานภายในแอปพลิเคชัน มีลักษณะการทำงานดังนี้



ภาพที่ 3.6 ภาพสถาปัตยกรรมของระบบ (System Architecture)

1. Widget เป็นองค์ประกอบพื้นฐานที่ใช้ในการสร้าง UI (User Interface) ของแอปพลิเคชัน ซึ่งมีหน้าที่แสดงข้อมูลและปรับเปลี่ยนส่วนต่าง ๆ ของ UI ตามโครงสร้างและการตั้งค่าที่กำหนดไว้
2. Widget States สถานะของ Widget สามารถเปลี่ยนแปลงสถานะได้ ซึ่งสามารถมีสถานะที่เปลี่ยนแปลงตามเหตุการณ์หรือการกระทำของผู้ใช้งาน เช่น การเลื่อนหน้าจอ เป็นต้น

3. Controller ใน Flutter คือองค์ประกอบหนึ่งในการจัดการข้อมูลและสถานะของแอปพลิเคชัน โดยทั่วไปมักใช้ในการควบคุมสิ่งต่าง ๆ เช่น การควบคุมการแสดงผลขององค์ประกอบต่าง ๆ

4. Service มีหน้าที่ประมวลผลทางธุรกิจและ การเชื่อมต่อกับแหล่งข้อมูลภายนอก เช่น การสื่อสารกับเซิร์ฟเวอร์ผ่าน API, การจัดการฐานข้อมูล, การกระทำทางธุรกิจ เป็นต้น

5. Model คือส่วนของโครงสร้างที่ใช้ในการเก็บข้อมูลและประมวลผลข้อมูล และการจัดรูปแบบข้อมูลสำหรับการแสดงผลในหน้าจอ

6. Repository คือชั้นของความทรงจำที่ใช้ในการจัดการการเข้าถึงและประมวลผลข้อมูลจากฐานข้อมูล, โดยมักจะเป็นตัวกลางระหว่างแอปพลิเคชันและแหล่งข้อมูลเหล่านั้น เพื่อให้การเข้าถึงข้อมูลเป็นไปอย่างมีระเบียบและมีประสิทธิภาพ

7. DTOs ย่อมาจาก Data Transfer Objects คือ แบบจำลองข้อมูลที่ใช้ในการโอนย้ายข้อมูลแอปพลิเคชันกับฐานข้อมูล โดยใช้เครื่องมือ Firebase SDK (Software Development Kit)

8. Data Source คือ ฐานข้อมูล และนำข้อมูลเหล่านั้นมาใช้ในการแสดงผลหรือประมวลผลในแอปพลิเคชัน โดยใช้เครื่องมือ Firebase

4. ผลการดำเนินงาน

ในบทนี้จะแสดงตัวอย่างหน้าจอ ทดสอบการทำงาน และผลการสำรวจความพึงพอใจของแอปพลิเคชัน

4.1 ตัวอย่างหน้าจอและการทำงานฟังก์ชันการเปิดบิลซ่อม



ภาพที่ 4.1 ภาพหน้าจอบันทึกข้อมูลการซ่อม

4.2 ผลประเมินการทดสอบฟังก์ชันการทำงาน

การทดสอบนี้จะทำการทดสอบฟังก์ชันของโปรแกรม ช่วยให้นักพัฒนาสามารถตรวจสอบความถูกต้องและประสิทธิภาพของโค้ดได้ง่ายขึ้น ซึ่งทำให้การจัดการข้อผิดพลาดและการแก้ไขโค้ดเป็นไปได้อย่างรวดเร็ว มีตัวอย่างผลการทดสอบดังนี้

4.2.1 การทดสอบฟังก์ชันการทำงานหน้าจอเปิดบิลซ่อม

เป็นหน้าจอบันทึกการซ่อม โดยเริ่มการทำงานโดยเริ่มจากเลือกรถแล้วระบบจะนำทางไปที่หน้าจอเลือกอะไหล่ เมื่อเลือกอะไหล่ในรายการ ข้อมูลอะไหล่จะถูกส่งไปที่หน้าจอรายการบิลซ่อม เมื่อกดปุ่มชำระเงิน ระบบจะบันทึกข้อมูลการซ่อมลงในฐานข้อมูล

ตารางที่ 4.1 ทดสอบฟังก์ชัน TC_Openbill_01

Test Case ID	TC_Openbill_01
Testcase	ตรวจสอบ Openbill กรณีมีอะไหล่ในรายการ
Pre-Requisite	1. เข้าสู่ระบบด้วยที่มีสถานะเป็นเจ้าของร้าน/ผู้จัดการ/พนักงาน

Test Step Description	1. เลือกมอเตอร์ไซค์เพื่อนำทางไปที่หน้าจอเลือกอะไหล่ (Click 125) 2. เลือกอะไหล่ระบบจะแสดงอะไหล่ในรายการอะไหล่ (ผ้าเบรคหน้า) 3. กดปุ่ม “ชำระเงิน” ในหน้าจอรายการอะไหล่เพื่อบันทึกข้อมูลการซ่อม
Expected Results	ระบบจะบันทึกข้อมูลซ่อมและนำทางกลับไปหน้าจอหลัก
Actual Result	ระบบจะบันทึกข้อมูลซ่อมและนำทางกลับไปหน้าจอหลัก
Status	ผ่าน

ตารางที่ 4.2 ทดสอบฟังก์ชัน TC_Openbill _02

Test Case ID	TC_Openbill _02
Testcase	ตรวจสอบ Openbill กรณีไม่มีอะไหล่ในรายการ
Pre-Requisite	1. เข้าสู่ระบบด้วยที่มีสถานะเป็นเจ้าของร้าน/ผู้จัดการ/พนักงาน
Test Step Description	1. เลือกมอเตอร์ไซค์เพื่อนำทางไปที่หน้าจอเลือกอะไหล่ (Click 125) 2. กดปุ่ม “ชำระเงิน” ในหน้าจอรายการอะไหล่เพื่อบันทึกข้อมูลการซ่อม
Expected Results	ระบบจะแจ้งเตือนว่าต้องมีอะไหล่ในรายการ
Actual Result	ระบบจะแจ้งเตือนว่าต้องมีอะไหล่ในรายการ
Status	ไม่ผ่าน

4.3 ผลการประเมินความพึงพอใจ

การประเมินผลผู้จัดทำจะพิจารณาความพอใจจากบุคคลที่ทำงานในธุรกิจซ่อมบำรุงรักษารถจักรยานยนต์อัตโนมัติที่ทำการทดลองใช้งานแอปพลิเคชัน เพื่อนำข้อมูลมาวิเคราะห์โดยใช้แบบสอบถามจาก Google Form เพื่อประเมินความพึงพอใจของผู้ใช้งาน คือ เจ้าของร้าน ผู้จัดการ และพนักงาน

4.3.1 หัวข้อในการประเมินความพึงพอใจการใช้งานแอปพลิเคชัน กำหนดเกณฑ์การให้คะแนนความพึงพอใจมีทั้งหมด 5 ระดับ มีหัวข้อในการประเมิน ดังนี้

- 1 การประเมินความพึงพอใจด้านการทำงานตามฟังก์ชันของระบบ
- 2 การประเมินความสะดวกในการใช้งานของระบบ

4.3.2 ผลการประเมินความพึงพอใจ

ผลการประเมินความพึงพอใจด้านการทำงานตามฟังก์ชันของระบบ มีค่าเฉลี่ยอยู่ที่ 4.4 โดยมีหัวข้อที่มีการประเมินที่มีคะแนนมากที่สุดคือ ความพอใจฟังก์ชันการชำระเงิน ซึ่งมีคะแนนเฉลี่ย 4.8 และมีหัวข้อที่มีการประเมินที่มีคะแนนน้อยที่สุดคือ ความพอใจฟังก์ชันจัดการข้อมูลพนักงาน (เฉพาะเจ้าของร้าน)

มีคะแนนเฉลี่ยอยู่ที่ 4 และผลการประเมินความสะดวกในการใช้งานของระบบมีค่าเฉลี่ยอยู่ที่ 4.36 โดยมีหัวข้อที่มีการประเมินที่มีคะแนนมากที่สุดคือ ความเหมาะสมในการใช้สี และขนาดของตัวอักษรซึ่งมีคะแนนเฉลี่ย 4.6 และมีหัวข้อที่มีการประเมินที่มีคะแนนน้อยที่สุดคือ ความเหมาะสมของปริมาณข้อมูลที่แสดงในแต่ละหน้าจอ มีคะแนนเฉลี่ยอยู่ที่ 4.2

5. สรุปผลการวิจัย อภิปรายผล

การดำเนินการพัฒนา ANSpart : แอปพลิเคชันจัดการสินค้าคงคลังสำหรับธุรกิจซ่อมรถจักรยานยนต์อัตโนมัติที่มีเทอร์โมเตอร์ โดยใช้ Flutter เป็นเครื่องมือในการพัฒนาแอปพลิเคชัน และ Firebase เป็นเครื่องมือในการบันทึกข้อมูลและการยืนยันตัวตนของผู้ใช้งาน ฟังก์ชันและการทำงานในแอปพลิเคชันได้จากการวิเคราะห์ข้อมูล ศึกษาปัญหา มารวมกันเพื่อให้ออกแบบแอปพลิเคชัน ให้ผู้งานและเรียนรู้ได้ง่ายมากที่สุด ดังนั้นแอปพลิเคชันจึงสามารถใช้กับ ธุรกิจซ่อมบำรุงและดูแลรักษารถจักรยานยนต์ได้

ในการประเมินความพึงพอใจของผู้ใช้งาน จากแบบสอบถามความพึงพอใจใน Google Form ผลการประเมินความพึงพอใจของผู้ใช้งานที่มีต่อแอปพลิเคชัน โดยมี 3 กลุ่มผู้ใช้งานคือ พนักงาน ผู้จัดการ และเจ้าของร้าน ผลการวิเคราะห์ความพึงพอใจที่มีต่อแอปพลิเคชัน ในการทดลองใช้งานของผู้ใช้งานทั้งหมดจำนวน 5 คน ผลการประเมินอยู่ระดับมาก โดยมีค่าเฉลี่ยเท่ากับ 4.43 ค่าเบี่ยงเบนมาตรฐานเท่ากับ 0.64 ซึ่งมีหัวข้อผลการประเมินความพึงพอใจด้านการทำงานตามฟังก์ชันของระบบ มีค่าเฉลี่ยมากที่สุดอยู่ที่ 4.4 และมีผลหัวข้อการประเมินความสะดวกในการใช้งานของระบบมีค่าเฉลี่ยน้อยที่สุดอยู่ที่ 4.36

บรรณานุกรม

- Blognone. (2566). รู้จักภาษา Kotlin ภาษาที่สองของโลก Android ใช้ทดแทน-ควบคู่กับ Java ได้ 100%. Retrieved from <https://www.blognone.com/node/92537/>.
- Expert-programming-tutor. (2566). รู้จักกับ Swift: ภาษาโปรแกรมมิ่งอัจฉริยะสำหรับ iOS คืออะไร. Retrieved from https://expert-programming-tutor.com/tutorial/article/KG000314_Meet_Swift_-_The_Smart_Programming_Language_for_iOS.php.
- Flutter คืออะไร” aws. Retrieved on 30 October 2023 from <https://aws.amazon.com/th/what-is/flutter/>.
- Talence Hiring Guide: React Native Developer #4” talence. Retrieved from <https://www.talence.tech/blog/react-native-talence-hiring-guide-developer-4/>.