**บทที่ 1**

**บทนำ**

**1.1 ที่มาและความสำคัญของโครงงาน**

ในปัจจุบันเทคโนโลยีถูกพัฒนาไปอย่างรวดเร็วทำให้อุปกรณ์ต่าง ๆ มีพัฒนาเพื่อให้สอดคล้องกับการใช้งานมากขึ้นไม่ว่าจะเป็นโทรศัพท์มือถือธรรมดาที่ในปัจจุบันกลายเป็นสมาร์ทโฟนที่ทุกคนต่างมีพกติดตัวตลอดเวลาและเข้ามามีบทบาทในชีวิตประจำวันของเราเป็นอย่างมาก เช่น ในทางธุรกิจมีอุปกรณ์สวิตช์ควบคุมไร้สายสั่งงานผ่าน อินเตอร์เน็ต ของ Sonoff [1] ที่ใช้ในการเปิด-ปิดอุปรณ์ต่าง ๆ โดยสั่งการผ่านแอปพลิชัน ซึ่งจะเห็นได้ว่าแอปพลิเคชันที่ทำงานอยู่บนสมาร์ทโฟนถูกพัฒนาออกมาอย่างต่อเนื่องเพื่อตอบสนองความต้องการของผู้ใช้งาน ทั้งนี้มีงานวิจัยเทคโนโลยี Internet of Things (IoT) มากมายที่พัฒนาโดยการนำแอปพลิเคชั่นมาใช้ในการควบคุมอุปกรณ์ ซึ่งในแต่ละงานวิจัยก็มีทั้งข้อดีและข้อจำกัดแตกต่างกันออกไป ดังนี้

**ตารางที่ 1.1** รายการแพลตฟอร์มด้านอุปกรณ์ IoT ที่เกี่ยวข้อง

|  |  |  |
| --- | --- | --- |
| **แพลตฟอร์มและรายละเอียด** | **ข้อดี** | **ข้อจำกัด** |
| 1. NETPIE แพลตฟอร์มที่ถูกพัฒนาขึ้นเพื่อตอบสนองผู้ใช้งานเชิงพาณิชย์จะช่วยธุรกิจให้มีประสิทธิภาพยิ่งขึ้น ด้วยเทคโนโลยีการเชื่อมต่อทุกสรรพสิ่ง หรือ Internet of Things (IoT) [2] | ช่วยให้อุปกรณ์ต่างๆ สามารถสื่อสารกันได้ เกิดการรับ-ส่งข้อมูลระหว่างอุปกรณ์แบบ เรียลไทม์ ทำให้ผู้ใช้งานทราบถึงข้อมูลของอุปกรณ์ ณ เวลานั้นๆ | เป็นเพียงเว็บไซต์ ทำให้ต้องใช้เบราว์เซอร์เพื่อใช้งานระบบ อาจไม่สะดวกต่อใช้งานสำหรับผู้ใช้งานสมาร์ทโฟน ~~และสามารถเชื่อมต่อได้เพียงแค่อุปกรณ์ 1 ตัวเท่านั้น~~ |
| 2. ThingSpeak คือ ผู้ให้บริการ Cloud ชนิดหนึ่ง ที่อนุญาตให้เราสามารถส่งค่าต่างๆขึ้นเพื่อไปเก็บบนพื้นที่ ที่เปิดให้เราใช้บริการ และยังเปิดให้เราสามารถเข้าถึงข้อมูลเหล่านี้ได้จากเวปบราวเซอร์ทั่ว ๆ ไปจากที่ไหนก็ได้ที่มี Internet จึงนับเป็นการเข้าถึงข้อมูลในรูปแบบ IOT (Internet of Things) [3] | สามารถจัดการข้อมูลการใช้งานอุปกรณ์ IoT ได้ง่ายขึ้นและสามารถออกแบบกราฟเอาไว้สรุปข้อมูลให้สามารถอ่านการใช้งานได้ง่ายมากขึ้น | ยังไม่มีการให้บริการรูปแบบ Mobile Application (\*\*\*สำรวจมาแล้ว?\*\*\*) ทำให้ต้องใช้งานระบบผ่านเบราว์เซอร์ อาจทำให้ไม่สะดวกในการใช้งานในกรณีที่ผู้ใช้งานไม่สะดวกใช้เว็บเบราว์เซอร์ |

จากแอปพลิเคชันที่ได้ยกตัวอย่างในข้างต้นปัญหาของการนำแอปพลิเคชั่นมาควบคุมเทคโนโลยี Internet of Things (IoT) ที่กล่าวมาในข้างต้น ที่กล่าวมาในข้างต้น ทำให้ได้เห็นถึงปัญหาในการใช้งานและการ Config code ที่ยุ่งยาก และมีความจำเป็นที่ต้องใช้งานผ่านเว็บเบราว์เซอร์เท่านั้น ประกอบกับเทคโนโลยี Sonoff [1] ที่มีท้องตลาดที่มี~~ปัญหา~~ข้อจำกัดในเรื่องมีฟังก์ชันที่ไม่เพียงพอต่อการใช้งานของผู้ใช้งาน เพราะทำได้เพียงสั่งการเปิด-ปิดเท่านั้น

ดังนั้น ทางผู้พัฒนาจึงได้เห็นปัญหาและแนวทางการนำแอปพลิเคชั่น (หรือ แอปพลิเคชัน) เข้ามาปรับใช้ให้ผู้ใช้งานสามารถนำไปใช้งานได้ ทางผู้พัฒนาออกแบบพัฒนาแอปพลิเคชันที่ใช้ควบคุมอุปกรณ์ IOT (หรือ IoT) ทีใช้ควบคุมในด้านต่าง ๆ ซึ่งระบบที่ผู้พัฒนานำเสนอจะช่วยในเรื่องของการติดตั้งและการดูแลรักษาให้ผู้ใช้มีความสะดวกมากยิ่งขึ้น เช่น การเชื่อมต่อแอปพลิเคชั่นกับอุปกรณ์สามารถทำได้ง่ายขึ้น โดยไม่จำเป็นที่จะต้องใช้ผู้เชี่ยวชาญในการเชื่อมต่อ ในแอปพลิเคชันที่ผู้พัฒนาได้ทำการพัฒนามาเน้นในกลุ่มของผู้ใช้งานที่ไม่เข้าใจการเขียน Code และกลุ่มใช้งานที่ต้องการนำ code ใช้ในการทดลองกับการใช้งานในด้านอื่นๆ

**1.2 วัตถุประสงค์**

1.2.1 เพื่อออกแบบและพัฒนาแอปพลิเคชันในการบริหารจัดการอุปกรณ์ไอโอทีให้สามารถใช้งานได้ และมีฟังก์ชันที่ครอบคลุมความต้องการของผู้ใช้งาน

1.2.2 เพื่อทดสอบและประเมินผลการเชื่อมต่อและควบคุมอุปกรณ์IOTผ่านแอปพลิเคชันco’act ได้ด้วยอุปกรณ์ต้นแบบอย่างน้อย 2 อุปกรณ์

**1.3 ขอบเขตของโครงการ**

1.3.1 แอปพลิเคชัน มีขอบเขตการดำเนินการ ดังนี้

1.3.1.1 สมัครสมาชิกและเข้าสู่ระบบได้

1.3.1.2 สามารถกู้และแก้ไขรหัสผ่านได้

1.3.1.3 สามารถแสดงรายการอุปกรณ์ที่เชื่อมต่อ

1.3.1.4 สามารถแสดง เพิ่มและลบรายการที่เชื่อมต่ออุปกรณ์ได้

1.3.1.5 สามารถจัดการอุปกรณ์ที่เชื่อมต่อได้ ดังนี้

1) เปลี่ยนชื่ออุปกรณ์ได้ (\*\*\*ไม่แน่ใจมีรายละเอียดอื่นด้วยหรือไม่?\*\*\*)

2) ผู้ใช้งานสามารถกำหนดเงื่อนไขในการควบคุมอุปกรณ์ได้ เช่น ตั้งเวลาในการทำงานของอุปกรณ์ หรือสั่งการทำงานด้วยตนเอง

1.3.1.6 สามารถดูประวัติการใช้งานอุปกรณ์ได้ในรูปแบบตาราง

1.3.1.7 สามารถดูประวัติการใช้งานอุปกรณ์ได้ในรูปแบบแดชบอร์ด

1.3.1.8 สามารถจัดการโปรไฟล์ของผู้ใช้งาน

1.3.1.9 สามารถเปลี่ยนภาษาของแอปพลิเคชันได้

1.3.1.10 สามารถเปลี่ยนโหมดสว่างหรือโหมดกลางคืนได้

1.3.1.11 สามารถเชื่อมต่ออุปกรณ์ด้วยบลูทูธผ่านแอปพลิเคชันได้

1.3.1.12 สามารถกำหนดการเชื่อมต่อไวไฟที่ต้องการให้อุปกรณ์เชื่อมต่อผ่านแอปพลิเคชันได้ ด้วยเทคโนโลยี local storage

**1.4 วิธีการดำเนินการ**

1.4.1 ศึกษาที่มาและความสำคัญของโครงงาน

1.4.2 กำหนดวัตถุประสงค์ของโครงงาน

1.4.3 กำหนดขอบเขตของโครงงาน

1.4.4 ศึกษาแนวคิดและทฤษฎีของโครงงาน

1.4.5 ศึกษาเครื่องมือที่ใช้ในการพัฒนาระบบ

1.4.6 พัฒนาระบบ

1.4.7 ทดสอบการทำงานของระบบ

1.4.8 ปรับปรุงแก้ไขการทำงานของระบบ

1.4.9 จัดทำเอกสารสรุปรายละเอียดของโครงงาน

**1.5 แผนการดำเนินงาน**

**ตารางที่ 1.2** ขั้นตอนการดำเนินงาน

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ขั้นตอนการดำเนินงาน** | **พ.ศ. 2566** | | | | | **พ.ศ. 2567** | | |
| ส.ค. | ก.ย. | ต.ค. | พ.ย. | ธ.ค. | ม.ค. | ก.พ | มี.ค |
| 1. ศึกษาที่มาและความสำคัญของโครงงาน |  |  |  |  |  |  |  |  |
| 2. กำหนดวัตถุประสงค์ของโครงงาน |  |  |  |  |  |  |  |  |
| 3. กำหนดขอบเขตของโครงงาน |  |  |  |  |  |  |  |  |
| 4. ศึกษาแนวคิดและทฤษฎีของโครงงาน |  |  |  |  |  |  |  |  |
| 5. ศึกษาเครื่องมือที่ใช้ในการพัฒนาระบบ |  |  |  |  |  |  |  |  |
| 6. พัฒนาระบบ |  |  |  |  |  |  |  |  |
| 7. ทดสอบการทำงานของระบบ |  |  |  |  |  |  |  |  |
| 8. ปรับปรุงแก้ไขการทำงานของระบบ |  |  |  |  |  |  |  |  |
| 9. จัดทำเอกสารสรุปรายละเอียดของโครงงาน |  |  |  |  |  |  |  |  |

**1.6 ประโยชน์ที่คาดว่าจะได้รับ**

1.6.1 สามารถช่วยการบริหารจัดการและควบคุมอุปกรณ์ได้สะดวกผ่านแอปพลิเคชัน

1.6.2 สามารถสร้างฟังก์ชั่น (หรือ ฟังก์ชัน) ในแอปพลิเคชั่นได้หลากหลายให้ผู้ใช้งานสามารถเลือกใช้งานได้ เบื้องต้นรองรับการใช้งานกับอุปกรณ์ในรูปแบบควบคุมการเปิด-ปิดไฟ และ ควบคุมการจ่ายน้ำ

1.6.3 สามารถนำแนวทางการออกแบบและพัฒนาผลิตภัณฑ์ไปต่อยอดในเชิงธุรกิจได้ในอนาคต เช่น การเพิ่มฟังก์ชันในแอปพลิเคชันให้สามารถควบคุมอุปกรณ์ที่ไปใช้ในด้านอื่น ๆ หรือ เพิ่มฟังก์ชันที่สามารถปรับแต่งแอปพลิเคชั่นได้เองตามที่ผู้ใช้งาน

**1.7 ทรัพยากรที่ใช้งาน**

**1.7.1 ทางด้านซอฟต์แวร์**

1.7.1.1 Visual Studio Code

1.7.1.2 HTML

1.7.1.3 Reactjs (es6/jsx)

1.7.1.4 C/C++

1.7.1.5 CSS

1.7.1.6 Nuxt3 js

1.7.1.7 React Native

1.7.1.8 Tailwind CSS

1.7.1.9 Firebase

1.7.1.10 Arduino IDE

1.7.1.11 JavaScript

1.7.1.12 Node js

**1.7.2 ทางด้านฮาร์ดแวร์**

1.7.2.1 MacBook Pro (13-inch, 2016, Two Thunderbolt 3 ports) - Technical Specifications

1) CPU Intel Core i5 Dual-Core 2.9 GHz

2) 8GB 2133MHz onboard LPDDR3 memory

3) PCIe onboard SSD, 512GB

4) Intel Iris Graphics 550

5) macOS Ventura 13.5.1

* + - 1. Lenovo Tab

**บทที่ 2**

**กรอบแนวคิด ทฤษฎี และการทบทวน**

การจัดทำโครงงานนี้กล่าวถึงองค์ความรู้พื้นฐานที่ใช้ในการพัฒนาเว็บแอปพลิเคชัน ผู้จัดทำได้ค้นคว่า ศึกษาตำรา บทความ และแอปพลิเคชันที่เกี่ยวข้อง โดยแบ่งข้อมูลต่าง ๆ ดังนี้

2.1 ภาษาจาวาสคริป (JavaScript)

2.2 JSX (JavaScript XML)

2.3 รีแอ็คส์เนทิฟ (React Native)

2.4 ภาษาเฮชทีเอ็มแอล (HTML)

2.5 ภาษาซีเอสเอส (CSS)

2.6 ไฟร์เบส (Firebase)

2.7 เอพีไอ (API)

2.8 ภาษาซีหรือภาษาซีพลัสพลัส (C/C++)

2.9 อาดูโน่ไอดีอี (Arduino IDE)

2.10 โนดด็อทเจเอส (Node.js)

**2.1 ภาษาจาวาสคริป (JavaScript)**

2.2.1 ความหมายของภาษาจาวาสคริปต์ [4]

ภาษาจาวาสคริปต์ หมายถึง ภาษาสคริปต์เชิงวัตถุ สำหรับการเขียนโปรแกรมบน

ระบบอินเทอร์เน็ต ในรูปแบบที่เรียกว่า Client-side Script ซึ่งจะประมวลผลบนเครื่องคอมพิวเตอร์

ของผู้เยี่ยมชมเว็บไซต์ โดยใช้โปรแกรมเว็บเบราว์เซอร์ ซึ่งใช้ในการสร้างและพัฒนาเว็บไซต์ ร่วมกับ

ภาษา HTML และภาษา Java เพื่อให้เว็บไซต์สามารถโต้ตอบและตอบสนองกับผู้ใช้งานได้มากขึ้น มัก

ใช้เขียนเป็นฟังก์ชันสำหรับการใช้งานต่าง ๆ เช่น ตรวจสอบความถูกต้องของข้อมูลในแบบฟอร์ม

2.2.2 ความสามารถของภาษาจาวาสคริปต์ [4]

2.2.2.1 สามารถใช้เขียนโปรแกรมแบบง่ายได้

2.2.2.2 สามารถเขียนหรือเปลี่ยนแปลง HTML Element ได้

2.2.2.3 สามารถใช้ตรวจสอบข้อมูลได้

2.2.2.4 สามารถใช้ในการตรวจสอบผู้ใช้ได้

2.2.2.5 สร้าง Cookies เก็บข้อมูลของผู้ใช้ในคอมพิวเตอร์ของผู้ใช้ได้

2.2.2.6 มีคำสั่งที่ตอบสนองกับผู้ใช้งาน

(\*\*\*ถ้ามีตัวอย่างคำสั่ง หรือ สคริปต์อย่างง่ายประกอบด้วยจะดีมากในแต่ละหัวข้อ\*\*\*)

**2.2 เจเอสเอ็กซ์ (JSX)**

JSX (JavaScript XML) [5] คือการเขียน JavaScript ที่มี HTML เราสามารถใช้งานความสามารถของทั้งสองร่วมกันได้

2.2.1 รูปแบบการเขียน JSX [5]

2.2.1.1 มี Tag เปิดต้องมี Tag ปิด ทุกครั้งเช่น

2.2.1.2 ต้องมี root note เสมอ ถ้าเราต้องการจะสร้าง HTML Tag หลายๆตัวเราจำเป็นจะต้องครอบเอาไว้ด้วย Tag ชั้นนอกสุดให้รวมเป็นตัวเดียวกัน หากไม่ต้องการให้มี Tag ที่ไม่จำเป็นใน root note สามารถใช้ <>...</> ครอบแทนได้

2.2.1.3 การใส่ Class Style ใน JSX จะใช้ className='...'แทน class='...' เนื่องจากคำว่า class เป็น keyword

2.2.1.4 Property HTML ใดๆก็ตามที่จากเดิมเขียนอยู่ในรูปแบบพิมพ์เล็กทั้งหมด จะต้องเขียนให้อยู่ในรูป camel case แทน

2.2.1.5 ในการใช้งานร่วมกับค่าของ JavaScript ให้เขียนภายในเครื่องหมาย {...} โดย JavaScript ที่จะสามารถใส่ได้ต้องเป็น expression เท่านั้น หากต้องการเขียนชุดเงื่อนไขสามารถเขียนในรูปแบบ ternary operator แทนได้

**2.3 รีแอ็คส์เนทิฟ (React Native)**

React Native [6] คือ Javascript Framework หรือ ชุดคำสั่งที่ถูกสร้างขึ้นมาสำหรับพัฒนา Mobile Application แบบ Cross-platform เราสามารถเขียนคำสั่งชุดเดียว หรือมีความแตกต่างกันแต่ไม่มาก แต่สามารถทำงานได้บนระบบปฏิบัติการที่แตกต่างกัน โดยใช้ Javascript เป็นภาษาหลักในการเขียนโปรแกรม

React Native สามารถพัฒนา Mobile Apps บนระบบปฏิบัติการ iOS ( Upload to Apple Apps Store ) และ ระบบปฏิบัติการ Android ( Upload to Google Play Store ) โดยมีความความเสถียรที่เทียบเท่าหรือใกล้เคียงกับ Mobile Apps ที่พัฒนาแบบ Native 100% ซึ่งส่วนนี้ ขึ้นอยู่หลายปัจจัย ทั้งตัวผู้พัฒนาเอง การบริหารจัดการ resource อื่น ๆ

**2.4 ภาษาเฮชทีเอ็มแอล (HTML)**

HTML [7] เป็นภาษาคอมพิวเตอร์ในรูปแบบหนึ่งที่ใช้สำหรับสร้างหน้าเว็บ (Web Page) เพื่อ

เก็บข่าวสารข้อมูลที่ต้องการในรูปของเอกสารไฮเปอร์เท็กซ์ (Hypertext) ที่มีคุณสมบัติสามารถ

เชื่อมโยงหน้าเว็บหนึ่งไปยังหน้าเว็บอื่นๆ ได้โดยโครงสร้างของ HTML จะมีตัวกำกับ หรือแท็ก (Tag)

สำหรับใช้ในการควบคุมการแสดงผลของข้อความ รูปภาพ ตาราง และวัตถุอื่นๆ ผ่านเว็บเบราว์เซอร์

(Web Browser)

**2.5 ภาษาซีเอสเอส (CSS)**

CSS [8] คือ ภาษาที่ใช้ส าหรับตกแต่งเอกสาร HTML/XHTML ให้มีหน้าตา สีสัน ระยะห่าง

พื้นหลัง เส้นขอบและอื่นๆ ตามที่ต้องการ CSS ย่อมาจาก Cascading Style Sheets มีลักษณะเป็น

ภาษาที่มีรูปแบบในการเขียน Syntax แบบเฉพาะและได้ถูกก าหนดมาตรฐานโดย W3C เป็นภาษา

หนึ่งในการตกแต่งเว็บไซต์ ได้รับความนิยมอย่างแพร่หลาย

**2.6 ไฟร์เบส (Firebase)**

Firebaseไฟร์เบส [9] คือ แพลตฟอร์มสำหรับนักพัฒนาโมบายแอปพลิเคชัน และเว็บแอป

พลิเคชัน มีเครื่องมือและโครงสร้างพื้นฐานสนับสนุนนักพัฒนาในการสร้างแอปพลิเคชัน บริษัทก่อตั้ง

ขึ้นในปี 2011 โดย James Tamplin และ Andrew Lee ให้เป็นฐานข้อมูลเรียลไทม์บริการสำหรับ

Backend ต่อมากูลเกิ้ลได้ซื้อกิจการในปี 2014 และพัฒนาให้สามารถบริการครบวงจรรองรับทั้งแบบ

Frontend และ Backend รองรับทั้ง iOS, Android, Web application ครอบคลุมทุกอย่างที่

นักพัฒนาต้องใช้

2.6.1 Firebase Cloud Functions

Firebase Cloud Functions [10] คือบริการที่ทำงานในฝั่ง server เพื่อตอบรับการ trigger จากบริการต่างๆใน Firebase โดยฟังก์ชันที่เขียนขึ้นมาทั้งหมดจะถูกเก็บไว้ที่ Google Cloud ซึ่งจะดูแลเรื่องความปลอดภัย ความเสถียร และการ scalable ให้แล้ว

การทำงานหลังจาก deploy โค้ดขึ้นไปที่ Cloud Functions for Firebase แล้วตัว Cloud Functions จะทำตัวเป็นนักดักฟัง (Listener) และรอรับ trigger จากบริการของFirebase ทันที และจะทำงานตามฟังก์ชันที่เขียนไว้ เช่น ส่งอีเมล, ส่ง Push Notification, สร้าง Thumbnail, กรองคำหยาบ, ต่อ Google APIs, ต่อ Third-party APIs (แพ๊คเกจที่ไม่ใช่ Spark), นับจำนวน Child ที่มี และอื่นๆ

2.6.2 Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) [11] เดิมเรียกว่าGoogle Cloud Messaging(GCM) เป็นโซลูชันระบบคลาวด์ข้ามแพลตฟอร์มสำหรับข้อความและการแจ้งเตือนสำหรับAndroid ,iOSและแอปพลิเคชันบนเว็บซึ่งในปี 2021 สามารถใช้ได้โดยไม่มีค่าใช้จ่าย Firebase CloudMessaging ช่วยให้นักพัฒนาแอพลิเคชันของบุคคลที่สามส่งการแจ้งเตือนหรือข้อความจากเซิร์ฟเวอร์เจ้าภาพโดย FCM กับผู้ใช้งานแพลตฟอร์มหรือผู้ใช้ปลายทาง

2.6.3 Firebase Realtime Database

Firebase Realtime Database [12] เป็น NoSQL cloud database ที่เก็บข้อมูลในรูปแบบของ JSON และมีการ sync ข้อมูลแบบ realtime กับทุก devices ที่เชื่อมต่อแบบอัตโนมัติในเสี้ยววินาที รองรับการทำงานเมื่อ offline (ข้อมูลจะถูกเก็บไว้ใน local จนกระทั่งกลับมา onlineก็จะทำการ sync ข้อมูลให้อัตโนมัติ) รวมถึงมี Security Rules ให้เราสามารถออกแบบเงื่อนไขการเข้าถึงข้อมูลทั้งการ read และ write ได้ดังใจ ทั้ง Android, iOS และ Web

**2.7 เอพีไอ (API)**

API [13] ย่อมาจาก Application Programming Interface คือ ช่องทางการเชื่อมต่อ ช่องทางหนึ่งที่จะเชื่อมต่อกับเว็บไซต์ผู้ให้บริการ API จากที่อื่น เป็นตัวกลางที่ทำให้โปรแกรมประยุกต์เชื่อมต่อกับโปรแกรมประยุกต์อื่น หรือเชื่ิอมการทำงานเข้ากับระบบปฏิบัติการ

ตัวอย่าง เช่น Twitter มีหลายเว็บ ที่มีการเชื่อมโยงข้อมูลกับ twitter ทั้งเป็นการอ่านข้อมูลจาก twitter หรือ ส่งข้อมูลเข้า twitter เองก็ตาม ซึ่งล้วนอาศัยการเชื่อมต่อแลกเปลี่ยนข้อมูลกัน ด้วย API นั้นเอง และอีกหนึ่งตัวอย่าง เช่น Google Maps API คือบริการของGoogle อีกรูปแบบหนึ่งที่เราสามารถนำข้อมูลของ Google Maps ที่ทาง Google ให้บริการโดยส่วนมากจะนำมาใช้กับเว็บไซต์ ของบริษัทฯหรือเว็บไซต์ห้างร้านต่างๆ เพื่อเป็นอีกช่องทางที่ให้ลูกค้ารู้ว่าบริษัทฯ หรือห้างร้านนั้น

2.7.1 ประโยชน์ของ API

2.7.1.1 สามารถรับส่งข้อมูลข้าม Server ได้

2.7.1.2 ไม่จำเป็นต้องเข้าหน้าเว็บหลัก ก็มีข้อมูลของเว็บหลัก จากเว็บที่ดึง APIเอพีไอ แบ่งเป็นเอพีไอที่ขึ้นกับภาษา (language-dependent API) คือ เอพีไอ ที่สามารถการเรียกใช้จากโปรแกรมที่เขียนขึ้นด้วยภาษาเพียงภาษาใดภาษาหนึ่ง และ เอพีไอไม่ขึ้นกับภาษา (language-independent API) คือ เอพีไอ ที่สามารถเรียกได้จากโปรแกรมหลายๆภาษา

**2.8 ภาษาซีหรือภาษาซีพลัสพลัส (C/C++)**

ภาษาซี [14] เป็นภาษาที่พัฒนาจากห้องปฏิบัติการเบลล์ของบริษัทเอทีแอนด์ทีในปี พ.ศ. 2515 หลังจากที่พัฒนาขึ้นได้ไม่นาน ภาษาซีก็กลายเป็นภาษาที่นิยมในหมู่นักเขียนโปรแกรมมาก และมีใช้งานในเครื่องทุกระดับ ทั้งนี้เนื่องจากภาษาซีได้รวมเอาข้อมูลของภาษาระดับสูงและภาษาระดับต่ำเข้าไว้ด้วยกัน กล่าวคือเป็นภาษาที่มีไวยากรณ์ที่เข้าใจง่าย ทำให้เขียนโปรแกรมได้ง่ายเช่นเดียวกับภาษาระดับสูงทั่วไป แต่ประสิทธิภาพและความเร็วในการทำงานดีกว่ามาก เนื่องจากมีการทำงานเหมือนภาษาระดับต่ำ สามารถทำงานได้ในระดับที่เป็นการควบคุมฮาร์ดแวร์ได้มากกว่าภาษาระดับสูงอื่น ๆ ดังจะเห็นว่าภาษาซีเป็นภาษาที่สามารถพัฒนาระบบปฏิบัติการได้ เช่น ระบบปฏิบัติการยูนิกซ์

นอกจากนี้เมื่อแนวคิดของการเขียนโปรแกรมแบบเชิงวัตถุ (Object Oriented Programming : OOP) ได้เข้ามามีบทบาทในวงการคอมพิวเตอร์มากขึ้น ภาษาซีก็ยังได้รับการพัฒนาโดยประยุกต์ใช้กับการเขียนโปรแกรมดังกล่าว เกิดเป็นภาษาใหม่ชื่อว่า “ภาษาซีพลัสพลัส” (C++)

**2.9 อาดูโน่ไอดีอี (Arduino IDE)**

Arduino IDE [15] เป็นโปรแกรมที่ “แจกฟรี” ในการใช้งานลักษณะ Open source ซึ่ง Arduino IDE จะทำหน้าที่ ติดต่อระหว่างคอมพิวเตอร์ ไม่ว่าจะเป็นระบบWindows, Mac OS X หรือ Linux กับ บอร์ด Arduino ซึ่งโปรแกรมนี้ออกแบบให้ง่ายต่อการเขียนโค้ดและอัปโหลดโปรแกรมที่เราเขียนเข้าสู่บอร์ด Arduino

Arduino IDE ส่วน IDE ย่อมาจก (Integrated Development Environment) คือ ส่วนเสริมของระบบการพัฒนา หรือตัวช่วยต่าง ๆ ที่จะคอยช่วยเหลือ Developer หรือช่วยเหลือคนที่พัฒนา Application เพื่อเสริมให้เกิดความรวดเร็ว ถูกต้อง แม่นยำ ตรวจสอบระบบที่จัดทำได้ ทำให้การพัฒนางานต่าง ๆ เร็วมากขึ้น

ส่วนในการเขียนโปรแกรมและคอมไพล์ลงบอร์ด โดยขนาดของโปรแกรม Arduino โดยปกติแล้วจะใหญ่กว่าโค้ด AVR ปกติเนื่องจากโค้ด AVR เป็นการเข้าถึงจากรีจิสเตอร์โดยตรง แต่โค้ด Arduino เข้าถึงผ่านฟังก์ชั่น เพื่อให้สามารถเขียนโค้ดได้ง่ายมากกว่าการเขียนโค้ดแบบ AVR หรือเวอร์ชั่นอื่นๆ ของ Arduino

**2.10 โนดด็อทเจเอส (Node.js)**

NodeJS [16] คือ Cross Platform Runtime Environment สำหรับฝั่ง Server เป็น Open Source และ Library ที่ใช้สำหรับพัฒนาเว็บแอปพลิเคชันต่าง ๆ ด้วยภาษา JavaScript เหมาะสำหรับการสร้างแอปพลิเคชันที่ต้องการใช้ข้อมูลจำนวนมาก และนิยมใช้ในการพัฒนาแอปพลิเคชันที่ใช้ข้อมูลแบบ Realtime สามารถทำงานได้ทุกระบบปฏิบัติการ โดยถูกนำมาเป็น Web Server, IoT, Webkit, TVOS, OS และอื่น ๆ เป็นต้น

**บทที่ 3**

**การออกแบบและพัฒนาระบบ**

วิธีการออกแบบและพัฒนาระบบ ผู้ศึกษาได้วิเคราะห์และออกแบบระบบด้วยวิธีการพัฒนา

เชิงวัตถุ (Object-oriented Design Methodology) ซึ่งทำให้ง่ายต่อความเข้าใจ โดยนำเสนอในรูปแบบของ UML (Unified Modeling Language) ดังต่อไปนี้

3.1 แผนภาพยูสเคส (Use Case Diagram)

3.2 คำอธิบายยูสเคส (Use Case Description)

3.3 แผนภาพกิจกรรม (Activity Diagram)

3.4 ออกแบบหน้าแอปพลิเคชั่น (Layout Application)

3.5 พจนานุกรมข้อมูล (Data Dictionary)

**3.1 แผนภาพยูสเคส (Use Case Diagram)**

แสดงภาพรวมของระบบว่าผู้ใช้ต้องการอะไรจากระบบ ซึ่งจะช่วยให้ผู้ออกแบบสามารถกำหนดรายละเอียดการทำงานของโปรแกรมให้ตอบสนองความต้องการของผู้ใช้ให้ได้มากที่สุดรายละเอียดแสดง ดังตารางที่ 3.1

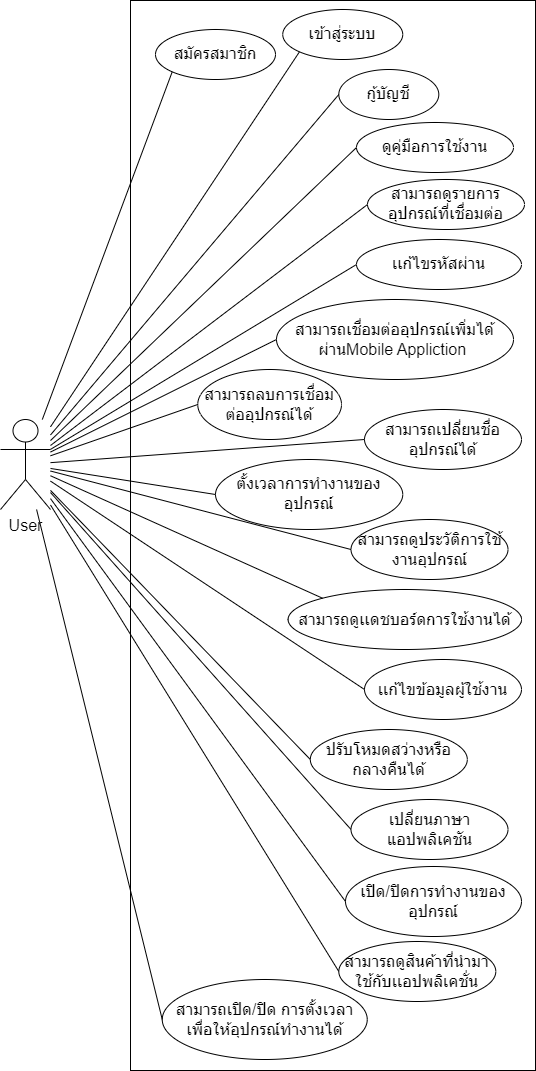
**ตารางที่ 3.1** ตัวอย่างสัญลักษณ์ Use Case Diagram (\*\*\*ถ้ามีอ้างอิงด้วยจะดีมาก\*\*\*)

|  |  |
| --- | --- |
| **สัญลักษณ์ที่ใช้** | **ความหมาย** |
|  | ผู้ที่เกี่ยวข้องกับระบบ อาจจะเป็นคนหรือระบบอื่นก็ได้ |
|  | ฟังก์ชันต่าง ๆ ที่เกิดขึ้นในระบบ |
|  | ขอบเขตระบบ |
|  | เส้นเชื่อมระหว่าง Actor กับ Use Case |
|  | ความสัมพันธ์ในกรณีที่ Use Case หนึ่งไปเรียกใช้อีก Use Case หนึ่ง |

**ตารางที่ 3.1** ตัวอย่างสัญลักษณ์ Use Case Diagram (ต่อ)

|  |  |
| --- | --- |
| **สัญลักษณ์ที่ใช้** | **ความหมาย** |
|  | ความสัมพันธ์ในกรณีที่ Use Case หนึ่งไปเรียกใช้อีก Use Case หนึ่ง หรือไม่ก็ได้ |
|  | ความสัมพันธ์ในกรณีที่ Use Case หนึ่งไปเรียกใช้อีก Use Case หนึ่ง |

3.1.1 Use Case Diagram ของ ผู้ใช้งานแอปพลิเคชัน



**ภาพที่ 3.1** Use Case ของผู้ใช้งานแอปพลิเคชัน

จากภาพที่ 3.1 แสดงถึงแผนภาพยูสเคสของผู้ใช้งานแอปพลิเคชันที่ประกอบด้วย 18 ยูสเคส ดังนี้ xxx xxx …. xxx

(\*\*\*ใน use case ควรระบุชื่อระบบด้วย และ บาง use case สามารถแยกกิจกรรมกันได้ เช่น เปิด/ปิด ตั้งเวลา / อยากให้เพิ่มเติมการอธิบายในแผนภาพพอเข้าใจ (ถ้า) สามารถปรับตัวอักษรในแผนภาพเป็นรูปแบบเดียวกันในเอกสารจะดีมาก ๆ\*\*\*)

**3.2 คำอธิบายยูสเคส (Use Case Description)**

**ตารางที่ 3.2** คําอธิบายยูสเคสเมื่อผูใชงาน สมัครสมาชิก

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | สมัครสมาชิก | |
| **Triggering Event :** | ผู้ใช้งานทำการสมัครสมาชิก | |
| **Brief Description :** | ให้กรอกข้อมูลที่ใช้ในการสมัครเข้าใช้งาน | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | ให้กรอกข้อมูลที่ใช้ในการสมัครเข้าใช้งาน | |
| **Post-Conditions :** | บันทึกข้อมูลที่ผู้ใช้งานที่ได้ทำการสมัคร | |
| **Flow of Events :** | **Actor** | **System** |
| 1.ผู้ใช้งาน กรอกข้อมูลในหน้า แอปพลิเคชัน  2.เลือกปุ่ม Signup เพื่อทำการยืนยันการสมัคร | 1.1 ระบบตรวจสอบข้อมูลและนำข้อมูลเข้าสู่ระบบ  2.1 นำผู้ใช้งานไปที่หน้า Signin |
| **Exception Conditions :** | ถาผูใชงานกรอกขอมูลไมครบจะไมสามารถสมัครสมาชิกได | |

**ตารางที่ 3.3** คําอธิบายยูสเคสเมื่อผูใชงาน เข้าสู่ระบบ

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | เข้าสู่ระบบ | |
| **Triggering Event :** | ผูใชงานทําการเขาสูระบบ แอปพลิเคชัน | |
| **Brief Description :** | ให้ผูใชงานกรอกชื่อผู้ใช้งานและรหัสผาน | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | ใหผูใชกรอกอีเมลและรหัสผาน | |
| **Post-Conditions :** | แสดงขอมูลการล็อคอิน | |
| **Flow of Events :** | **Actor** | **System** |
| 1 .ผู้ใช้งาน กรอกชื่อผู้ใช้และรหัสผ่าน  2. ผู้ใช้งานกดปุ่มเพื่อเข้าสู่ระบบ | 1.1 ระบบทำการตรวจสอบชื่อ  บัญชีและรหัสผาน ของผูใชงาน  2.1นำผู้ใช้งานเข้าสู่ระบบ |
| **Exception Conditions :** | ถาผูใชกรอกขอมูลไมถูกตองจะไมสามารถเขาสูระบบได้ | |

**ตารางที่ 3.4** คําอธิบายยูสเคสเมื่อผูใชงาน กู้บัญชี

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | กู้บัญชี | |
| **Triggering Event :** | ผู้ใช้งานจำรหัสผ่านไม่ได้ | |
| **Brief Description :** | ให้ผู้ใช้กดปุ่มลืมรหัสผ่านและระบบจะทำการส่งรหัสผ่านไปตามอีเมลที่ผู้ใช้กรอก | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | ให้ผู้ใช้งานกรอกรหัสผ่านใหม่ที่ต้องการ | |
| **Post-Conditions :** | ส่งรหัสไปที่อีเมลของผู้ใช้งาน | |
| **Flow of Events :** | **Actor** | **System** |
| 1 .กดปุ่มลืมรหัสผ่าน  2. กรอกอีเมลของผู้ใช้งานและกดปุ่มยืนยันเพื่อส่งข้อความ | 1.1 แสดงหน้าให้ผู้ใช้กรอกอีเมล  2.1 รับอีเมลที่ผู้ใช้งานกรอก และส่งรหัสผ่านไปยังอีเมลที่รับ |
| **Exception Conditions :** | ถ้าผู้ใช้งานกรอกอีเมลไม่ถูกต้องจะไม่ได้รับข้อความ | |

**ตารางที่ 3.5** คําอธิบายยูสเคสเมื่อผูใชงาน แก้ไขรหัสผ่าน

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | แก้ไขรหัสผ่าน | |
| **Triggering Event :** | ต้องการเปลี่ยนรหัสผ่าน | |
| **Brief Description :** | ให้ผู้ใช้งานกดไปที่เปลี่ยนรหัสผ่านและกรอกรหัสผ่านใหม่ในหน้าเว็บไซต์ | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | ให้ผู้ใช้กรอกรหัสผ่านใหม่โดยทำผ่านหน้าเบราว์เซอร์ | |
| **Post-Conditions :** | บันทึกรหัสผ่านใหม่ลงฐานข้อมูล | |
| **Flow of Events :** | **Actor** | **System** |
| 1. กดเข้าไปที่หน้าตั้งค่า  2. กดที่เปลี่ยนรหัสผ่าน  3. กรอกรหัสผ่านใหม่ที่ต้องการเปลี่ยน | 1.1 แสดงหน้าตั้งค่า  2.1 แสดงหน้าเบราว์เซอร์  3.1 บันทึกรหัสผ่านที่รับมาในฐานข้อมูล |
| **Exception Conditions :** | - | |

**ตารางที่ 3.6** คําอธิบายยูสเคสเมื่อผูใชงาน ดูรายการเชื่อมต่ออุปกรณ์

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | ดูรายการเชื่อมต่ออุปกรณ์ | |
| **Triggering Event :** | ดูอุปกรณ์ที่ได้เชื่อมต่อทั้งหมด | |
| **Brief Description :** | ให้ผู้ใช้งานกดไปที่หน้าแรกเพื่อดูอุปกรณ์ทั้งหมด | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | เลือกหน้าแรกเพื่อดูรายการเชื่อมต่ออุปกรณ์ | |
| **Post-Conditions :** | แสดงรายการทั้งหมดที่ผู้ใช้งานได้เชื่อมต่ออุปกรณ์ | |
| **Flow of Events :** | **Actor** | **System** |
| 1.ผู้ใช้งานเลือกเมนูหน้าแรก | 1.1 ดึงอุปกรณ์ที่ผู้ใช้งานได้เชื่อมต่อทั้งหมดมาแสดงในหน้าแรก |
| **Exception Conditions :** | - | |

**ตารางที่ 3.7** คําอธิบายยูสเคสเมื่อผูใชงาน เชื่อมต่ออุปกรณ์เพิ่มผ่าน Mobile Application

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | เชื่อมต่ออุปกรณ์เพิ่ม | |
| **Triggering Event :** | ต้องการเชื่อมต่ออุปกรณ์เพิ่ม | |
| **Brief Description :** | ผู้ใช้งานต้องทำการเพิ่มโดยการค้นหาจาก Wifi ของ Device | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | เลือก Wifi ของDeviceที่ต้องเชื่อมต่อ | |
| **Post-Conditions :** | บันทึกข้อมูล Wifi ลงในฐานข้อมูล | |
| **Flow of Events :** | **Actor** | **System** |
| 1. เชื่อมต่อ Wifi ที่มาจากDevice ผ่านการตั้งค่าในโทรศัพท์มือถือ และกดยืนยันว่าได้เชื่อมต่อแล้ว  2.เลือก Wifi ที่ต้องการเชื่อมต่อ อินเตอร์เน็ต  3.กรอกรหัสผ่าน Wifi และ ยืนยันการเชื่อมต่อ | 1.1 ตรวจสอบ Wifi ว่าได้เชื่อมต่อกับWifi Deviceหรือไม่  1.2 แสดงรายการ Wifi ที่มี อินเตอร์เน็ต  2.1 แสดงหน้าให้กรอกรหัสผ่าน  3.1 ตรวจสอบรหัสผ่านของ Wifi และทำการเชื่อมต่อกับ Wifi |
| **Exception Conditions :** | Wifi ที่เชื่อมต่อไม่มีอินเตอร์เน็ตทำให้ไม่สามารถใช้งานได้ , กรอกชื่อหรือรหัสผ่านไม่ถูกต้อง | |

**ตารางที่ 3.8** คําอธิบายยูสเคสเมื่อผูใชงาน ลบการเชื่อมต่ออุปกรณ์

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | ลบการเชื่อมต่ออุปกรณ์ | |
| **Triggering Event :** | ต้องการลบอุปกรณ์ที่ไม่ต้องการใช้งาน | |
| **Brief Description :** | ผู้ใช้งานต้องมีการเลือกอุปกรณ์เพื่อให้แสดงรายละเอียดทั้งหมดของอุปกรณ์นั้น และกดที่ปุ่มลบอุปกรณ์ | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | เลือกอุปกรณ์ที่ต้องการลบการเชื่อมต่อ | |
| **Post-Conditions :** | ลบอุปกรณ์นั้นออกจากฐานข้อมูลของผู้ใช้งาน | |
| **Flow of Events :** | **Actor** | **System** |
| 1. ผู้ใช้งานเข้าที่หน้า Home ของ Application  2. เลือกรายการที่ต้องการลบ  3. กดปุ่มลบอุปกรณ์ | 1.1 แสดงรายการอุปกรณ์ทั้งหมดในหน้า home  2.1 รับค่าจากผู้ใช้งานและแสดงรายละเอียดอุปกรณ์ที่ผู้ใช้งานเรียกดู  3.1 รับค่าและทำการลบอุปกรณ์นั้นออกจากฐานข้อมูลของผู้ใช้งาน |
| **Exception Conditions :** | - | |

**ตารางที่ 3.9** คําอธิบายยูสเคสเมื่อผูใชงาน เปลี่ยนชื่ออุปกรณ์

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | เปลี่ยนชื่ออุปกรณ์ | |
| **Triggering Event :** | ผู้ใช้งานต้องการเปลี่ยนชื่ออุปกรณ์ของตัวเอง | |
| **Brief Description :** | เข้าไปที่อุปกรณ์ที่ต้องการเปลี่ยนชื่อ และแก้ไขชื่อที่แถบเปลี่ยนชื่อ | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | เลือกรายการที่ต้องการเปลี่ยนชื่อ | |
| **Post-Conditions :** | แก้ไขชื่ออุปกรณ์ที่ผู้ใช้แก้ไข | |
| **Flow of Events :** | **Actor** | **System** |
| 1. ผู้ใช้งานเข้าที่หน้า Home ของ Application  2. เลือกรายการที่ต้องการแก้ไข  3. แก้ไขชื่อที่แถบชื่อ และ กดยืนยัน | 1.1 แสดงรายการอุปกรณ์ทั้งหมดในหน้า home  2.1 รับค่าจากผู้ใช้งานและแสดงรายละเอียดอุปกรณ์ที่ผู้ใช้งานเรียกดู  3.1 รับค่าและทำการแก้ไขชื่ออุปกรณ์นั้นในฐานข้อมูล |
| **Exception Conditions :** | - | |

**ตารางที่ 3.10** คําอธิบายยูสเคสเมื่อผูใชงาน ตั้งเวลาการทำงานของอุปกรณ์

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | ตั้งเวลาการทำงานของอุปกรณ์ | |
| **Triggering Event :** | ผู้ใช้งานต้องการตั้งเวลาเพื่อให้อุปกรณ์ทำงาน | |
| **Brief Description :** | เลือกแถบอุปกรณ์ที่ต้องการตั้งเวลาการทำงานอุปกรณ์ | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | เลือกรายการที่ต้องการตั้งเวลา และตั้งเวลาที่ต้องการให้ทำงาน | |
| **Post-Conditions :** | บันทึกเวลาที่ผู้ใช้งานได้ตั้งค่าไว้ในอุปกรณ์นั้น | |
| **Flow of Events :** | **Actor** | **System** |
| 1. ผู้ใช้งานเลือกอุปกรณ์ที่ต้องการตั้งเวลา  2. กดปุ่มเพิ่มเวลา  3.กรอกข้อมูลทั้งหมดในหน้ากรอกข้อมูลเวลา และกดบันทึกข้อมูลเวลาที่ต้องการ | 1.1 แสดงเวลาที่ถูกกำหนดไว้ทั้งหมดของอุปกรณ์นั้น  2.1 แสดงหน้ากรอกข้อมูลเวลา  3.1 รับข้อมูลจากหน้ากรอกข้อมูลเวลา และบันทึกลงฐานข้อมูลของอุปกรณ์นั้น |
| **Exception Conditions :** | - | |

**ตารางที่ 3.11** คําอธิบายยูสเคสเมื่อผูใชงาน ดูประวัติการใช้งานอุปกรณ์

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | ดูประวัติการใช้งานอุปกรณ์ | |
| **Triggering Event :** | ผู้ใช้งานต้องการดูประวัติการใช้งานอุปกรณ์ | |
| **Brief Description :** | เลือกเมนูประวัติการใช้งานเพื่อดูแล้วเลือกประเภทที่ต้องการแสดง ซึ่งมี 2 รูปแบบ คือรูปแบบกราฟ และ รูปแบบตาราง รวมถึงสามารถกำหนดชื่ออุปกรณ์ได้ | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | เลือกเมนูประวัติการใช้งาน และ รูปแบบของการแสดง | |
| **Post-Conditions :** | แสดงประวัติการใช้งาน ตามรูปแบบ และอุปกรณ์ตามที่ผู้ใช้งานกำหนด | |
| **Flow of Events :** | **Actor** | **System** |
| 1. กดเลือกเมนูประวัติการใช้งาน  2. กำหนดการค้นหาที่แถบค้นหา | 1.1 แสดงหน้าประวัติการใช้งาน และ แถบค้นหา  2.1 รับค่าที่ต้องการค้นหา และนำไปค้นหาเพื่อนำมาแสดงหน้าแอปพลิเคชั่น  2.2 แสดงค่าที่ได้จากการค้นหาให้ผู้ใช้งานสามารถดูข้อมูลที่ต้องการได้ |
| **Exception Conditions :** | - | |

**ตารางที่ 3.12** คําอธิบายยูสเคสเมื่อผูใชงาน ดูแดชบอร์ดการใช้งาน

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | ดูแดชบอร์ดการใช้งาน | |
| **Triggering Event :** | ผู้ใช้งานต้องการดูแดชบอร์ดการใช้งาน | |
| **Brief Description :** | เป็นการสรุปข้อมูลที่เป็นรายเดือนและรายปี | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | เลือกเมนูแดชบอร์ด และชื่ออุปกรณ์ที่ผู้ใช้งานต้องการค้นหา | |
| **Post-Conditions :** | แสดงแดชบอร์ด ตามที่ผู้ใช้ต้องการใช้งาน | |
| **Flow of Events :** | **Actor** | **System** |
| 1. เลือกเมนูแดชบอร์ด  2. กรอกชื่ออุปกรณ์ที่ต้องการค้นหา | 1.1 แสดงหน้าแดชบอร์ด  2.1 รับค่าที่ต้องการค้นหา และนำไปค้นหาเพื่อนำมาแสดงหน้าแอปพลิเคชั่น  2.2 แสดงค่าที่ได้จากการค้นหาให้ผู้ใช้งานสามารถดูข้อมูลที่ต้องการได้ |
| **Exception Conditions :** | - | |

**ตารางที่ 3.13** คําอธิบายยูสเคสเมื่อผูใชงาน แก้ไขข้อมูลผู้ใช้งาน

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | แก้ไขข้อมูลผู้ใช้งาน | |
| **Triggering Event :** | ผู้ใช้งานต้องการที่แก้ไขข้อมูลส่วนตัว | |
| **Brief Description :** | ผู้ใช้งานต้อวเข้าไปที่หน้าตั้งค่าและในแถบที่แสดงหน้ารูปโปรไฟล์เมื่อกดเข้าไปจะแสดงหน้าเพื่อให้ผู้ใช้งานสามารถแก้ไขได้ | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | ผู้ใช้งานต้องแก้ไขข้อมูลและกดบันทึกข้อมูลที่แก้ไข | |
| **Post-Conditions :** | บันทึกข้อมูลที่ผู้ใช้งานได้ทำการแก้ไข | |
| **Flow of Events :** | **Actor** | **System** |
| 1. ผู้ใช้งานเข้าหน้าตั้งค่า  2. เข้าในแถบโปรไฟล์  3. แก้ไขข้อมูลและกดบันทึกข้อมูล | 1.1 แสดงหน้าตั้งค่า  2.1 แสดงหน้าแก้ไขโปรไฟล์  3.1 รับค่าและบันทึกลงฐานข้อมูล |
| **Exception Conditions :** | ผู้ใช้งานกรอกข้อมูลผิดทำให้ไม่สามารถบันทึกข้อมูลได้ | |

**ตารางที่ 3.14** คําอธิบายยูสเคสเมื่อผูใชงาน ปรับโหมดสว่างหรือกลางคืน

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | ปรับโหมดสว่างหรือกลางคืน | |
| **Triggering Event :** | ผู้ใช้งานต้องการปรับโหมดในแอปพลิเคชัน | |
| **Brief Description :** | ในการตั้งค่าจะมีแถบที่ใช้สำหรับปรับโหมดของแอปพลิเคชัน | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | ผู้ใช้งานปรับโหมดกลางคืนหรือโหมดสว่าง | |
| **Post-Conditions :** | รับค่าและปรับตามค่าที่รับจากแอปพลิเคชัน | |
| **Flow of Events :** | **Actor** | **System** |
| 1. ผู้ใช้งานเข้าเมนูตั้งค่า  2. ผู้ใช้งานปรับโหมดเป็นโหมดกลางคืนหรือโหมดสว่าง | 1.1 แสดงหน้าการตั้งค่า  1.2 รับค่าและเปลี่ยนตามการตั้งค่าที่ผู้ใช้งานปรับ |
| **Exception Conditions :** | - | |

**ตารางที่ 3.15** คําอธิบายยูสเคสเมื่อผูใชงาน เปลี่ยนภาษาแอปพลิเคชัน

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | เปลี่ยนภาษาแอปพลิเคชัน | |
| **Triggering Event :** | ผู้ใช้งานต้องการปรับภาษาในแอปพลิเคชัน | |
| **Brief Description :** | ในการตั้งค่าจะมีแถบที่ใช้สำหรับปรับภาษาโดยสามารถเลือกได้เป็นภาษาอังกฤษหรือภาษาไทย | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | ผู้ใช้งานปรับเป็นภาษาไทยหรือภาษาอังกฤษ | |
| **Post-Conditions :** | รับค่าและปรับตามค่าที่รับจากแอปพลิเคชัน | |
| **Flow of Events :** | **Actor** | **System** |
| 1. ผู้ใช้งานเข้าเมนูตั้งค่า  2. ผู้ใช้งานปรับภาษาตามที่ต้องการ | 1.1 แสดงหน้าการตั้งค่า  1.2 รับค่าและเปลี่ยนตามการตั้งค่าที่ผู้ใช้งานปรับ |
| **Exception Conditions :** | - | |

**ตารางที่ 3.16** คําอธิบายยูสเคสเมื่อผูใชงาน สามารถควบคุมการทำงานอุปกรณ์แบบ Manual

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | สามารถควบคุมการทำงานอุปกรณ์แบบ Manual | |
| **Triggering Event :** | เมื่อผู้ใช้งานต้องการเปิดปิดอุปกรณ์เอง | |
| **Brief Description :** | ในรายการของอุปกรณ์แต่ละอุปกรณ์จะมีปุ่มสำหรับการเปิดปิดการทำงานของอุปกรณ์ | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | ผู้ใช้งานกดปุ่มเพื่อทำการปิดหรือเปิดอุปกรณ์ | |
| **Post-Conditions :** | รับค่าและสั่งที่อุปกรณ์ว่าให้เปิดการทำงานหรือปิดการทำงาน | |
| **Flow of Events :** | **Actor** | **System** |
| 1. เข้าหน้าหลักของแอปพลิเคชัน  2. กดปุ่มเปิดหรือปิดตามชื่ออุปกรณ์ที่ต้องการ | 1.1แสดงหน้าแรกแอปพลิเคชัน  1.2 รับค่าและสั่งที่อุปกรณ์ว่าให้เปิดการทำงานหรือปิดการทำงาน |
| **Exception Conditions :** | - | |

**ตารางที่ 3.17** คําอธิบายยูสเคสเมื่อผูใชงาน สามารถดูสินค้าที่นำมาใช้กับเเอปพลิเคชั่น

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | สามารถดูสินค้าที่นำมาใช้กับเเอปพลิเคชั่น | |
| **Triggering Event :** | ต้องการหาซื้อสินค้าเพื่อมาใช้กับแอปพลิเคชัน | |
| **Brief Description :** | แสดงสินค้าโดยที่แต่ละสินค้าจะถูกเชื่อมไปที่ร้านค้าออนไลน์ที่ขายสินค้านั้น | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | กดเข้าเมนูดูสินค้า | |
| **Post-Conditions :** | หน้าร้านค้าออนไลน์โดยเชื่อมผ่านลิงค์ | |
| **Flow of Events :** | **Actor** | **System** |
|  | 1.กดเมนูสินค้า  2.กดเลือกสินค้าที่ต้องการ | 1.1 แสดงรายการสินค้า  1.2 ไปตามลิงค์ที่เชื่อมไปสินค้าไว้ |
| **Exception Conditions :** | - | |

**ตารางที่ 3.18** คําอธิบายยูสเคสเมื่อผูใชงาน สามารถเปิด/ปิด การตั้งเวลาเพื่อให้อุปกรณ์ทำงานได้

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | สามารถเปิด/ปิด การตั้งเวลาเพื่อให้อุปกรณ์ทำงานได้ | |
| **Triggering Event :** | ผู้ใช้งานต้องการเปิดหรือปิดเวลาที่ได้ตั้งไว้อยู่แล้ว | |
| **Brief Description :** | ในแต่ละรายการช่วงเวลาจะมีปุ่มที่สามารถเปิดหรือปิดช่วงเวลาของการทำงานของอุปกรณ์ | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | กดปุ่มเพื่อปิดหรือเปิด ช่วงเวลาที่ตั้งไว้ | |
| **Post-Conditions :** | เปลี่ยนสถานะของช่วงเวลาที่ผู้ใช้งานได้กดปุ่ม | |
| **Flow of Events :** | **Actor** | **System** |
| 1. ผู้ใช้งานเข้าไปกดเข้าไปดูรายละเอียดของอุปกรณ์ที่ต้องการจัดการช่วงเวลา  2. กดปุ่มเปิดหรือปิด ช่วงเวลาที่ต้องการจัดการ | 1.1 แสดงรายละเอียดช่วงเวลาของอุปกรณ์ที่ผู้ใช้งานเลือกในหน้ารายการอุปกรณ์  2.1 บันทึกสถานะใหม่ที่ผู้ใช้งานเปลี่ยนสถานะใหม่ |
| **Exception Conditions :** | **-** | |

**ตารางที่ 3.19** คําอธิบายยูสเคสเมื่อผูใชงาน ดูคู่มือการใช้งาน

|  |  |  |
| --- | --- | --- |
| **Use Case Name :** | ดูคู่มือการใช้งาน | |
| **Triggering Event :** | ผู้ใช้งานต้องการที่จะเรียนรู้การช้งานแอปพลิเคชัน | |
| **Brief Description :** | เป็นหน้าสำหรับสอนวิธีการใช้งานแอปพลิเคชัน โดยมี 2 คู่มือ คือ คู่มือการเชื่อมต่ออุปกรณ์ และ คู่มือการใช้แอปพลิเคชัน | |
| **Actors :** | ผู้ใช้งาน | |
| **Related Use Case :** | - | |
| **Pre-Conditions :** | ผู้ใช้งานเลือกเมนูคู่มือ | |
| **Post-Conditions :** | แสดงหน้าคู่มือการใช้งานแอปพลิเคชัน | |
| **Flow of Events :** | **Actor** | **System** |
| 1. ผู้ใช้งานเลือกเมนูคู่มือ  2.ผู้ใช้งานเลือกคู่มือที่ต้องการ | 1.1 แสดงหน้าให้ผู้ใช้เลือกคู่มือที่ต้องการ  2.1 แสดงคู่มือที่ผู้ใช้งานเลือก |
| **Exception Conditions :** | **-** | |

(\*\*\*ในแผนภาพ use case และคำอธิบายยูสเคส ควรนำเสนอให้สอดคล้องกัน\*\*\*)

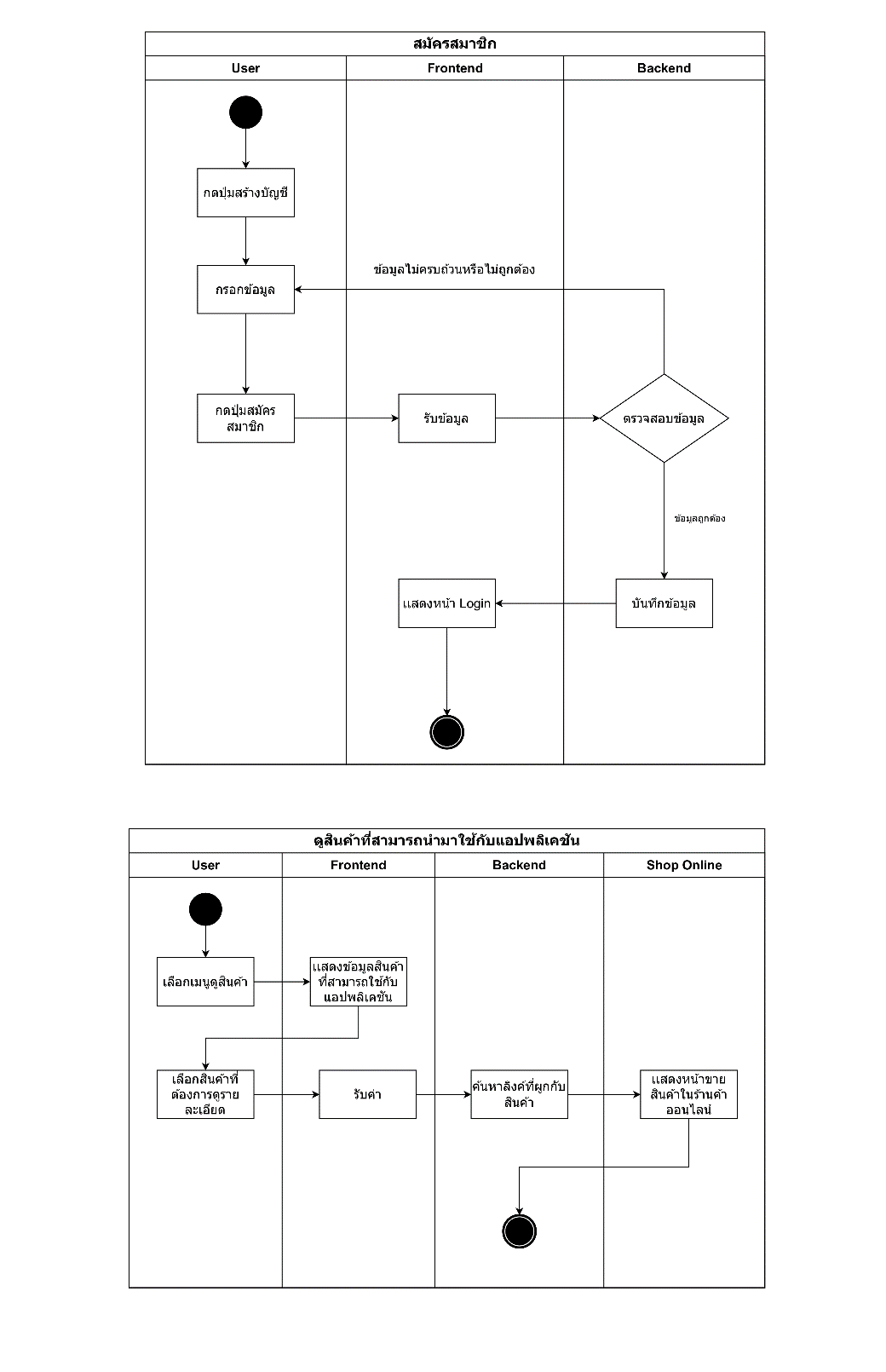
**3.3 แผนภาพกิจกรรม (Activity Diagram)**

(\*\*\*ตรวจสอบเส้นทางการไหลของหัวลูกศร และ คำที่แสดงถึงการทำงานของเงื่อนไขการตัดสินใจในแต่ละทางเลือก และ อยากให้เพิ่มเติมการอธิบายในแต่ละแผนภาพพอเข้าใจ (ถ้า) สามารถปรับตัวอักษรในแผนภาพเป็นรูปแบบเดียวกันในเอกสารจะดีมาก ๆ\*\*\*)

**ตารางที่ 3.19** ตัวอย่างสัญลักษณ์ Activity Diagram

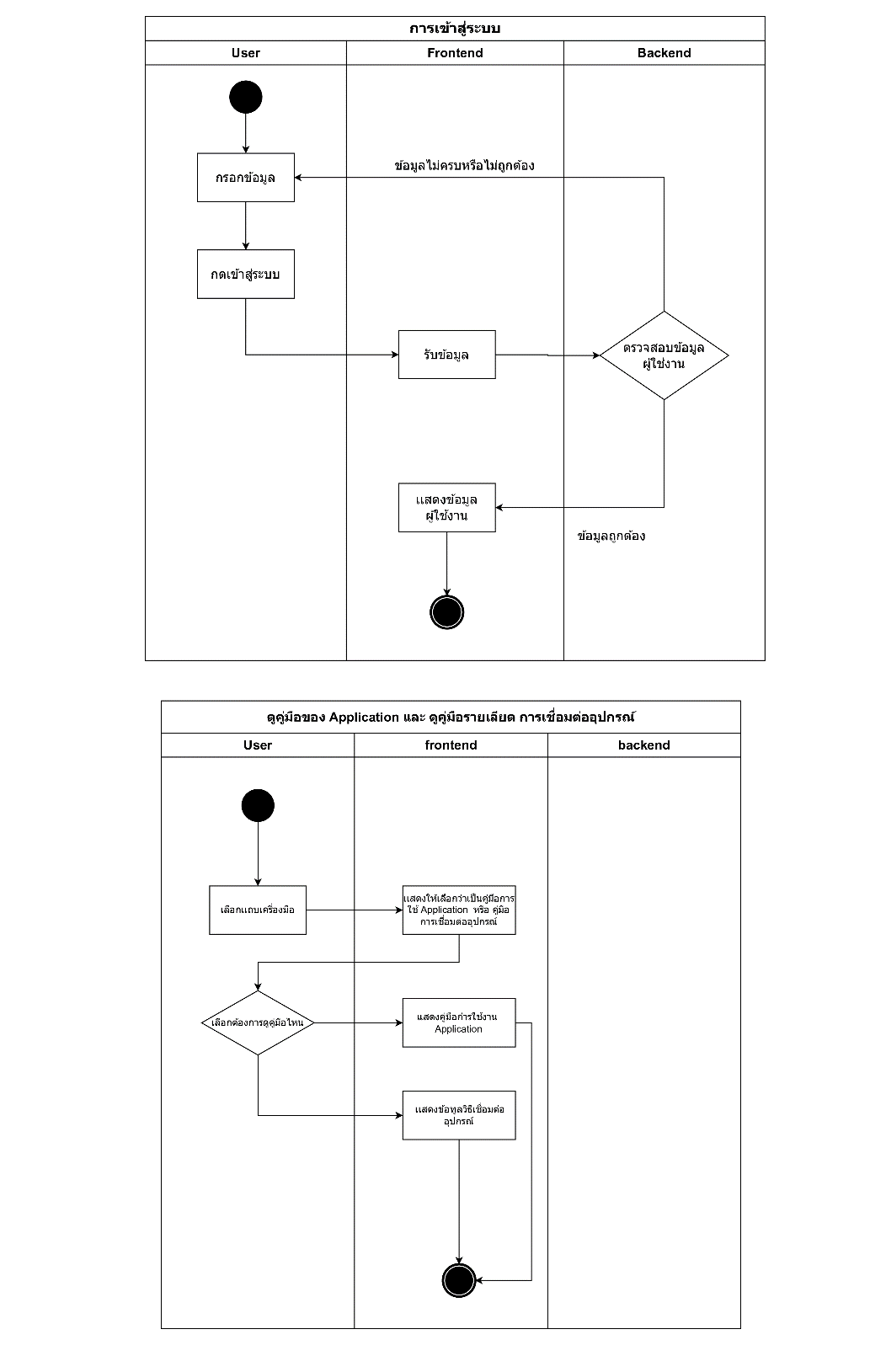
|  |  |
| --- | --- |
| **สัญลักษณ์ที่ใช้** | **ความหมาย** |
|  | กิจกรรม (Activity) |
|  | เส้นทางการไหลของกิจกรรม |
|  | กรณี Synchronization และ Join |
|  | จุดเริ่มต้น |
|  | จุดสิ้นสุด |
|  | สวิมแลนส (SWIMLANES) |
|  | แสดงการไหลของออบเจกต (Object Flow) |

3.3.1 Activity Diagram สมัครสมาชิก



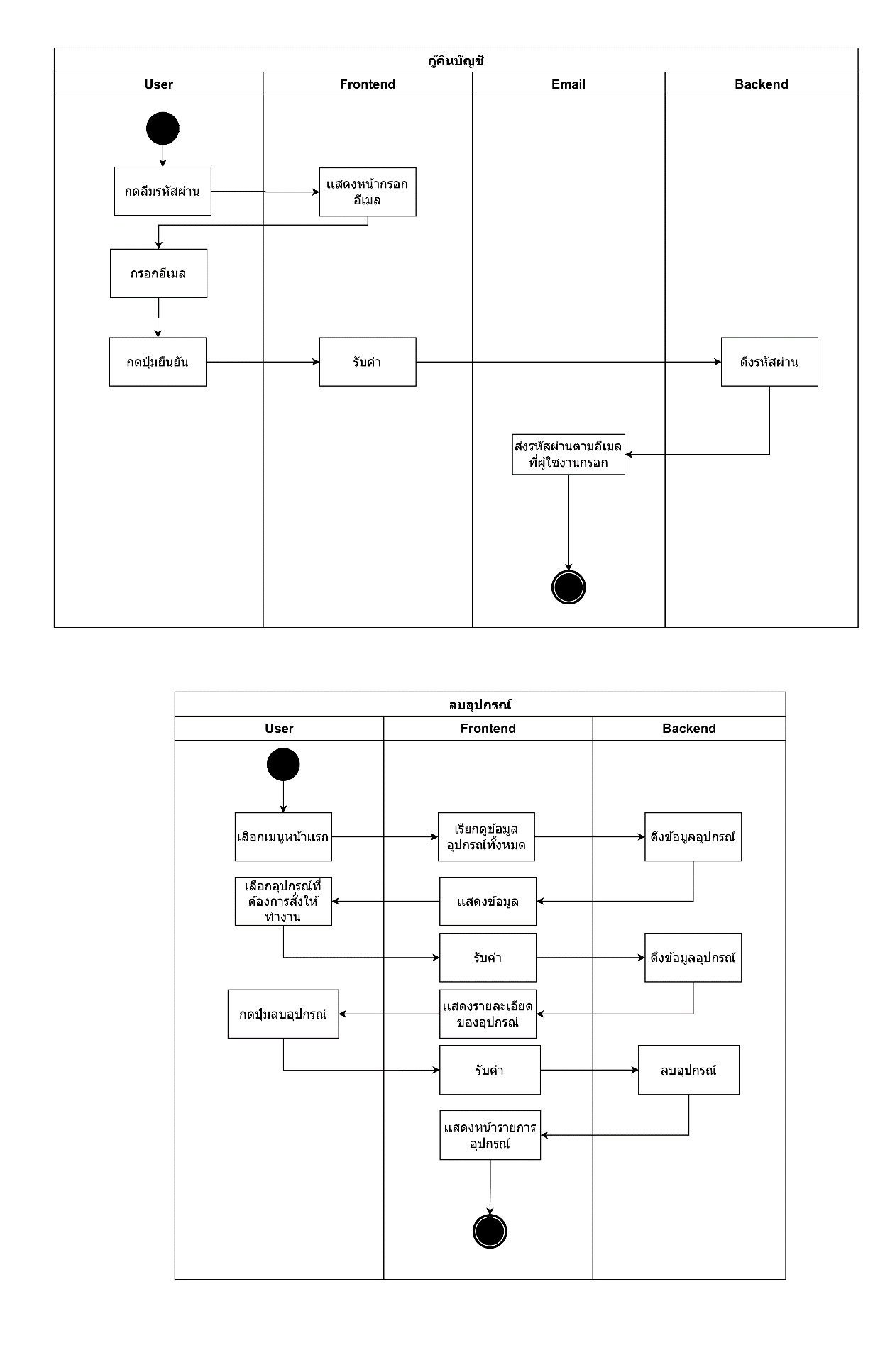
**ภาพที่ 3.2** Activity Diagram สมัครสมาชิก

3.3.2 Activity Diagram เข้าสู่ระบบ



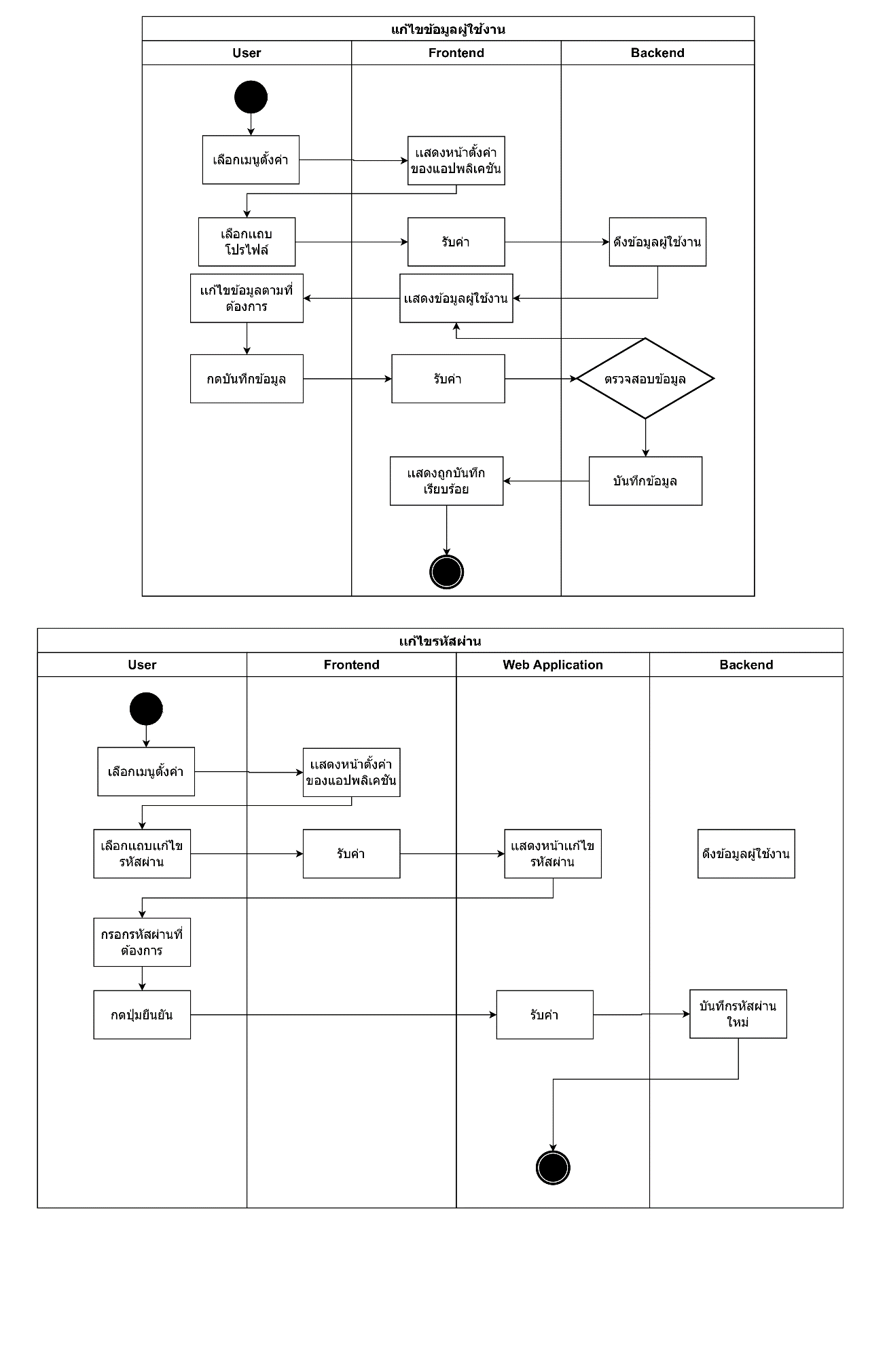
**ภาพที่ 3.3** Activity Diagram เข้าสู่ระบบ

3.3.3 Activity Diagram กู้คืนบัญชี



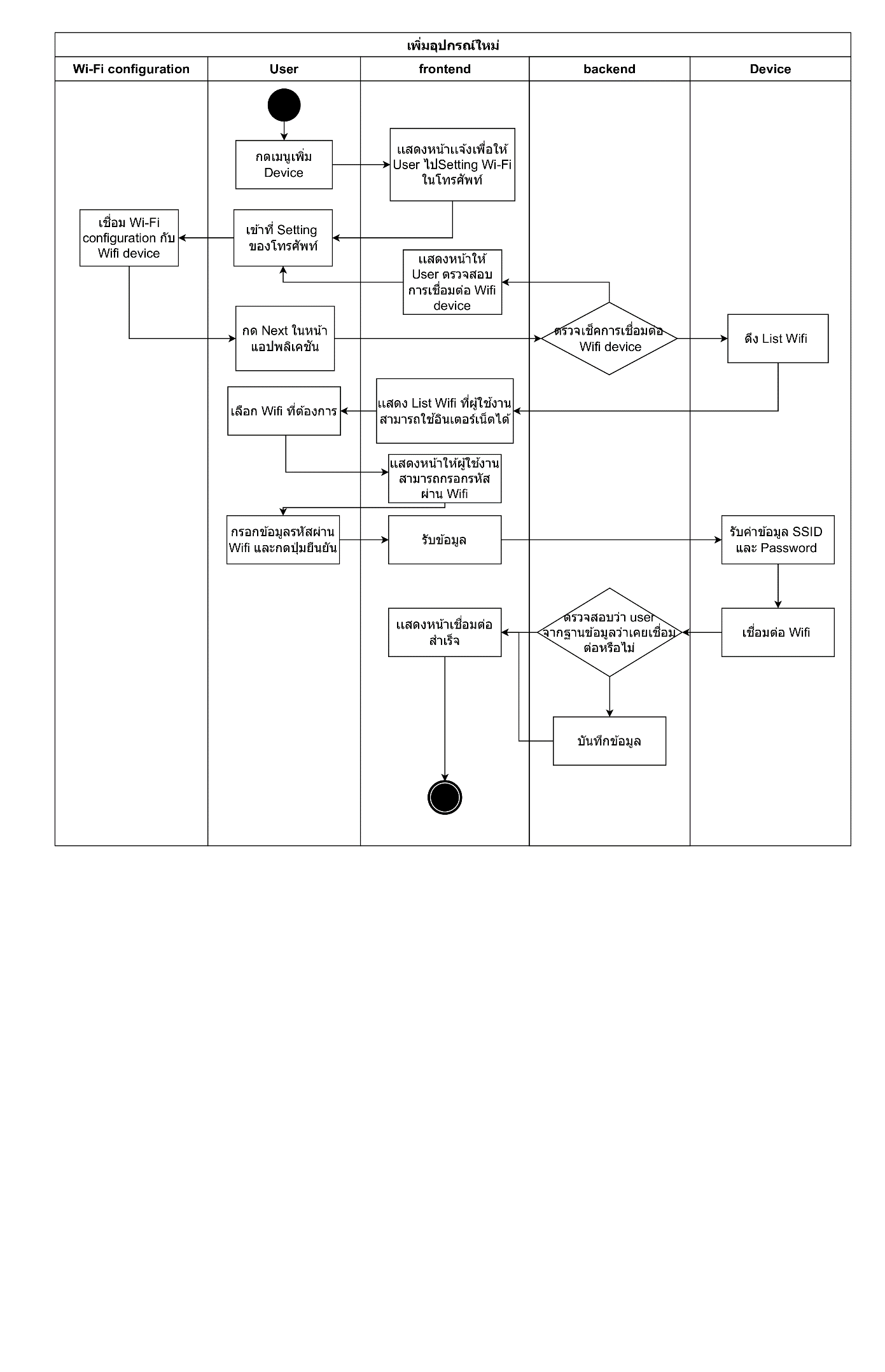
**ภาพที่ 3.4** Activity Diagram กู้คืนบัญชี

3.3.4 Activity Diagram แก้ไขรหัสผ่าน



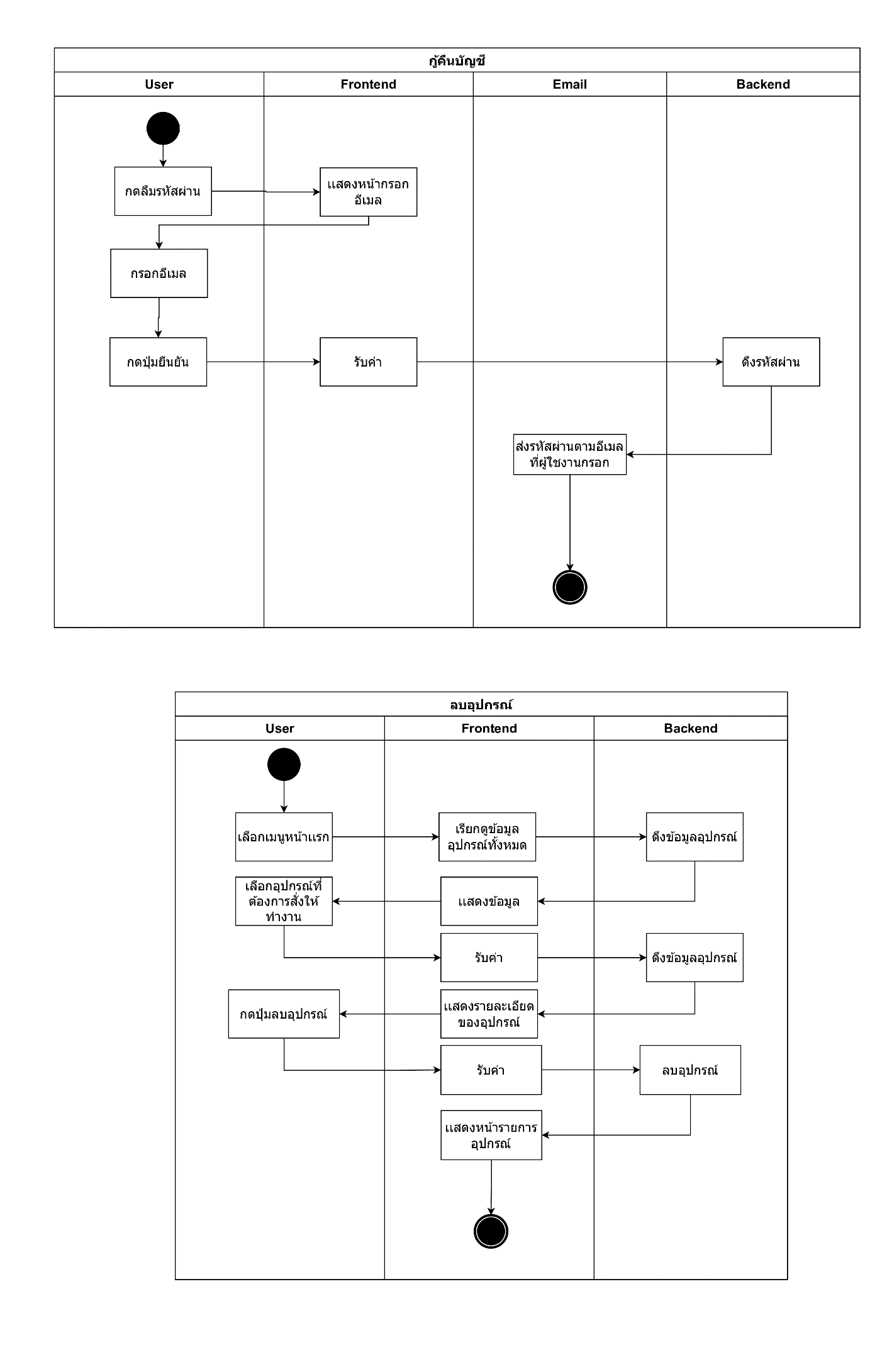
**ภาพที่ 3.5** Activity Diagram แก้ไขรหัสผ่าน

3.3.5 Activity Diagram เพิ่มอุปกรณ์ใหม่



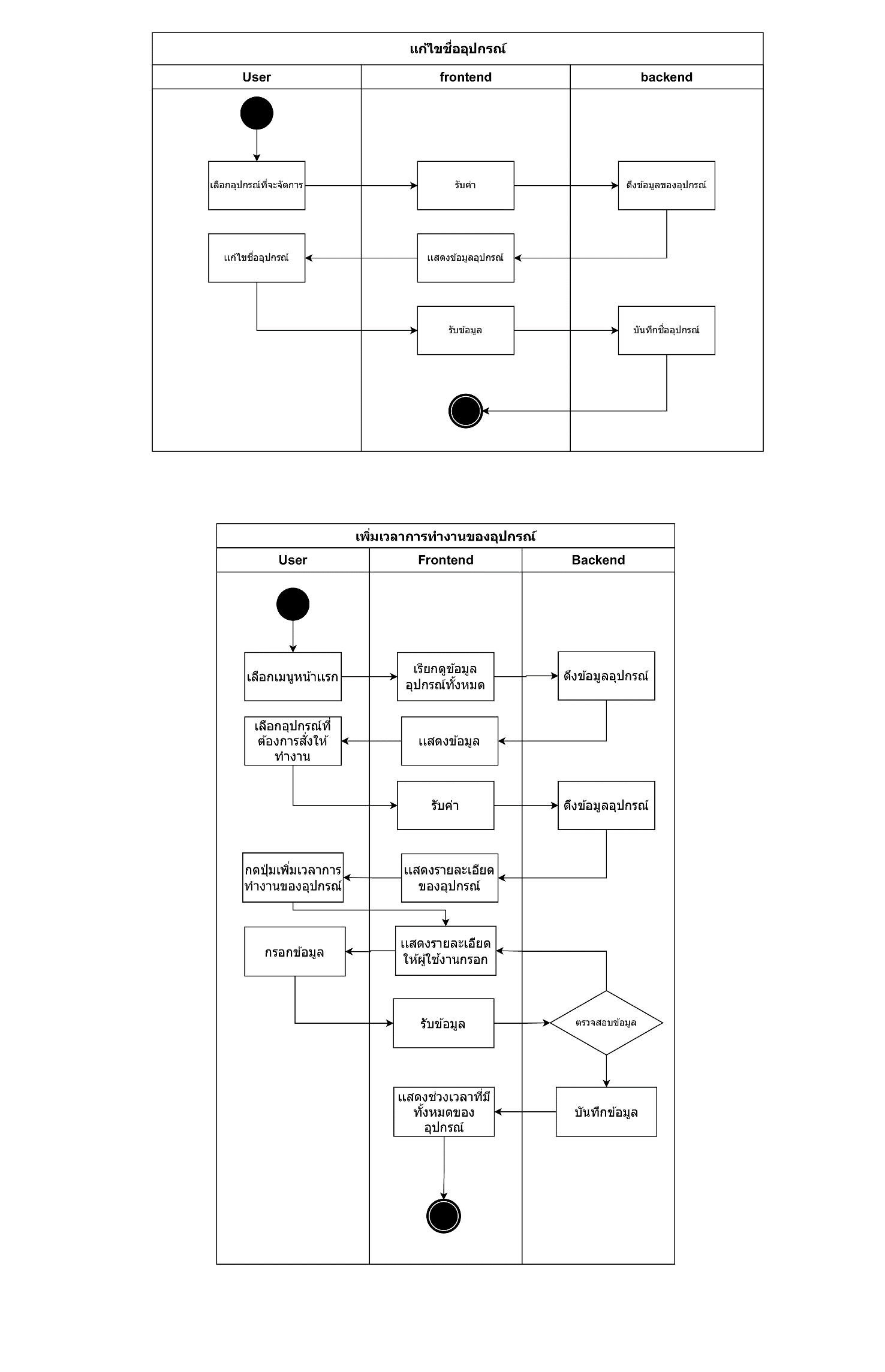
**ภาพที่ 3.6** Activity Diagram เพิ่มอุปกรณ์ใหม่

3.3.6 Activity Diagram ลบอุปกรณ์



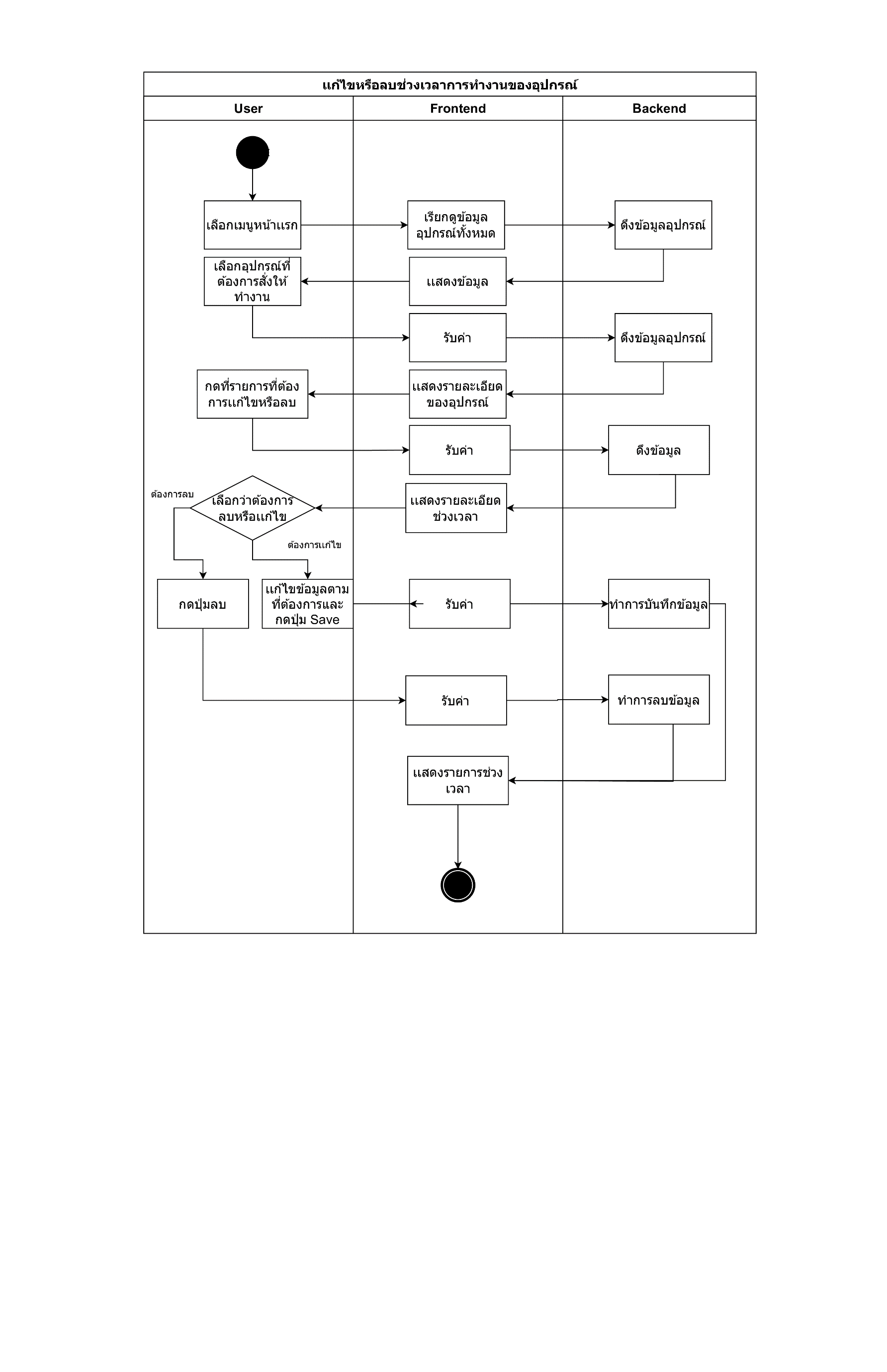
**ภาพที่ 3.7** Activity Diagram เพิ่มอุปกรณ์ใหม่

3.3.7 Activity Diagram แก้ไขชื่ออุปกรณ์



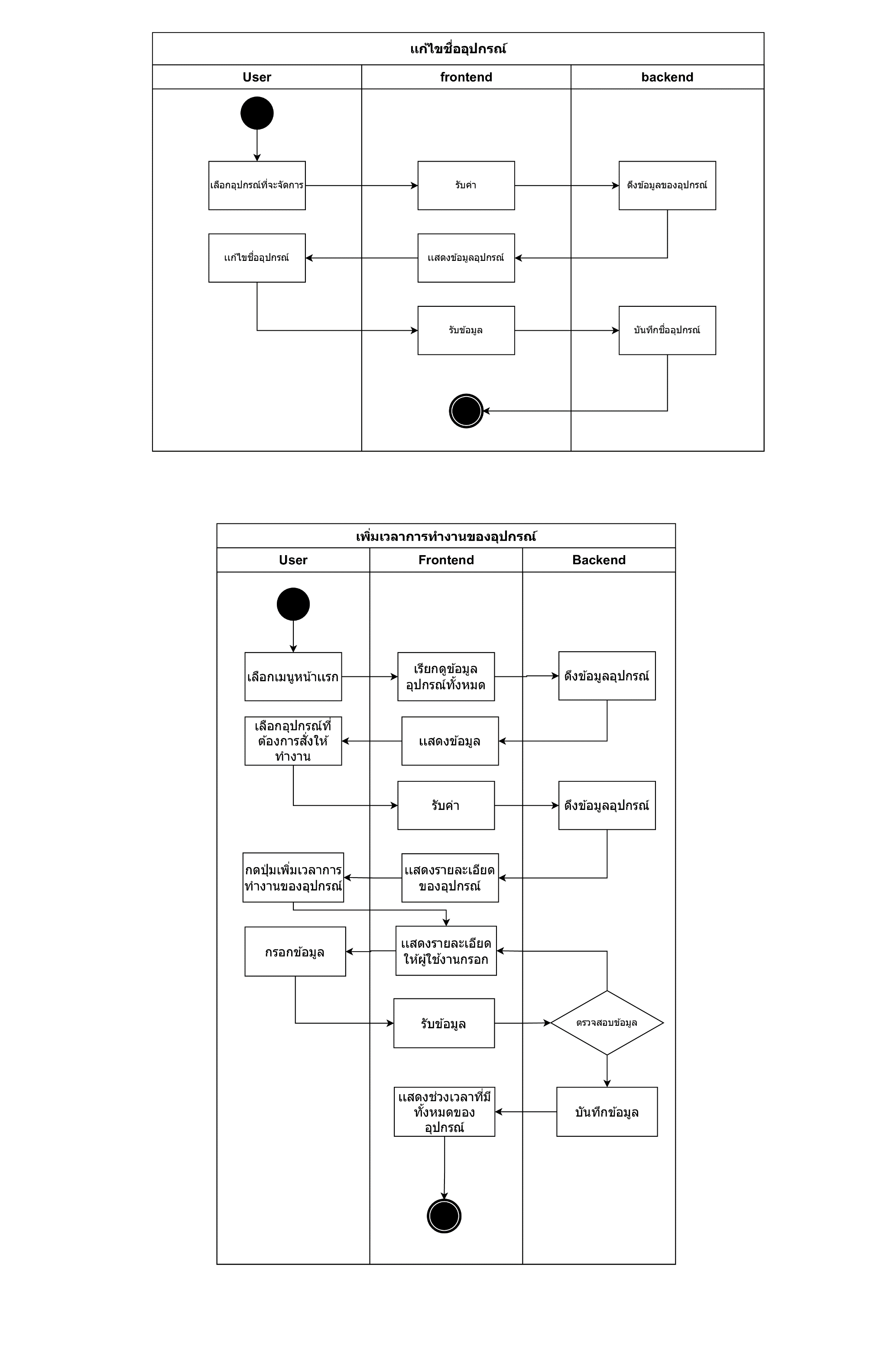
**ภาพที่ 3.8** Activity Diagram แก้ไขชื่ออุปกรณ์

3.3.8 Activity Diagram แก้ไขหรือลบช่วงเวลาการทำงานของอุปกรณ์



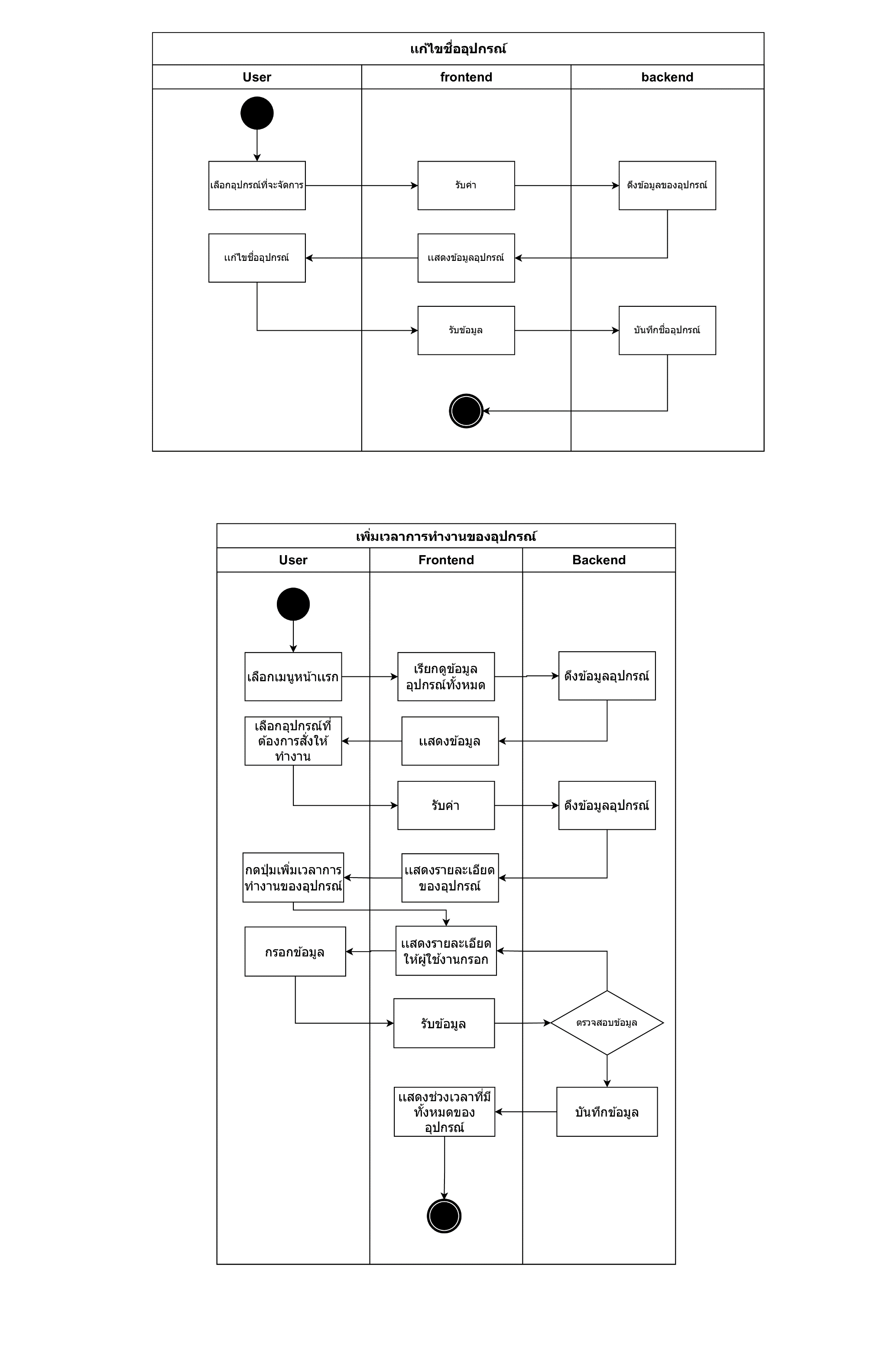
**ภาพที่ 3.9** Activity Diagram แก้ไขหรือลบช่วงเวลาการทำงานของอุปกรณ์

3.3.9 Activity Diagram เพิ่มเวลาการทำงานของอุปกรณ์



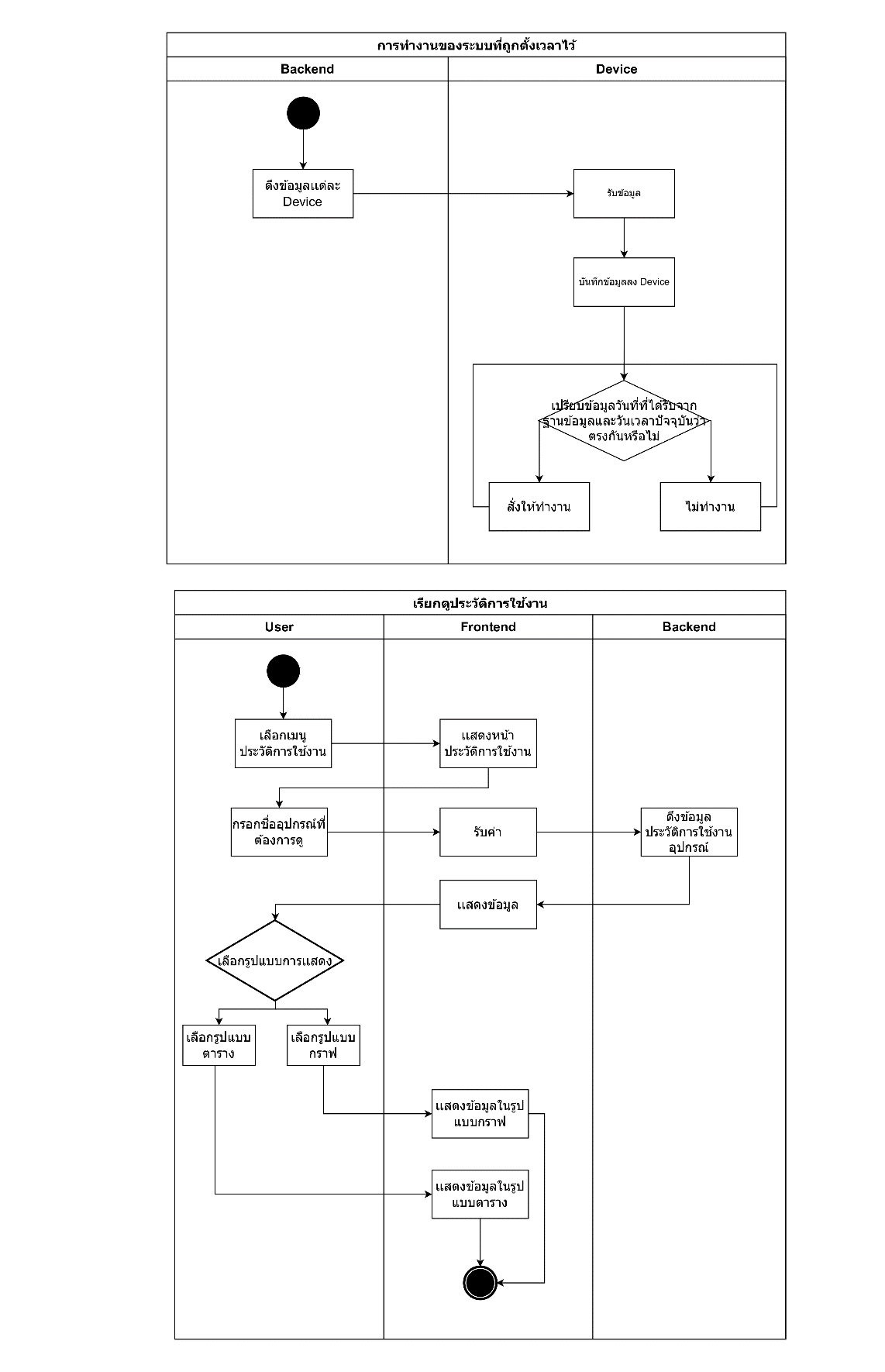
**ภาพที่ 3.10** Activity Diagram เพิ่มเวลาการทำงานของอุปกรณ์

3.3.10 Activity Diagram เพิ่มเวลาการทำงานของอุปกรณ์



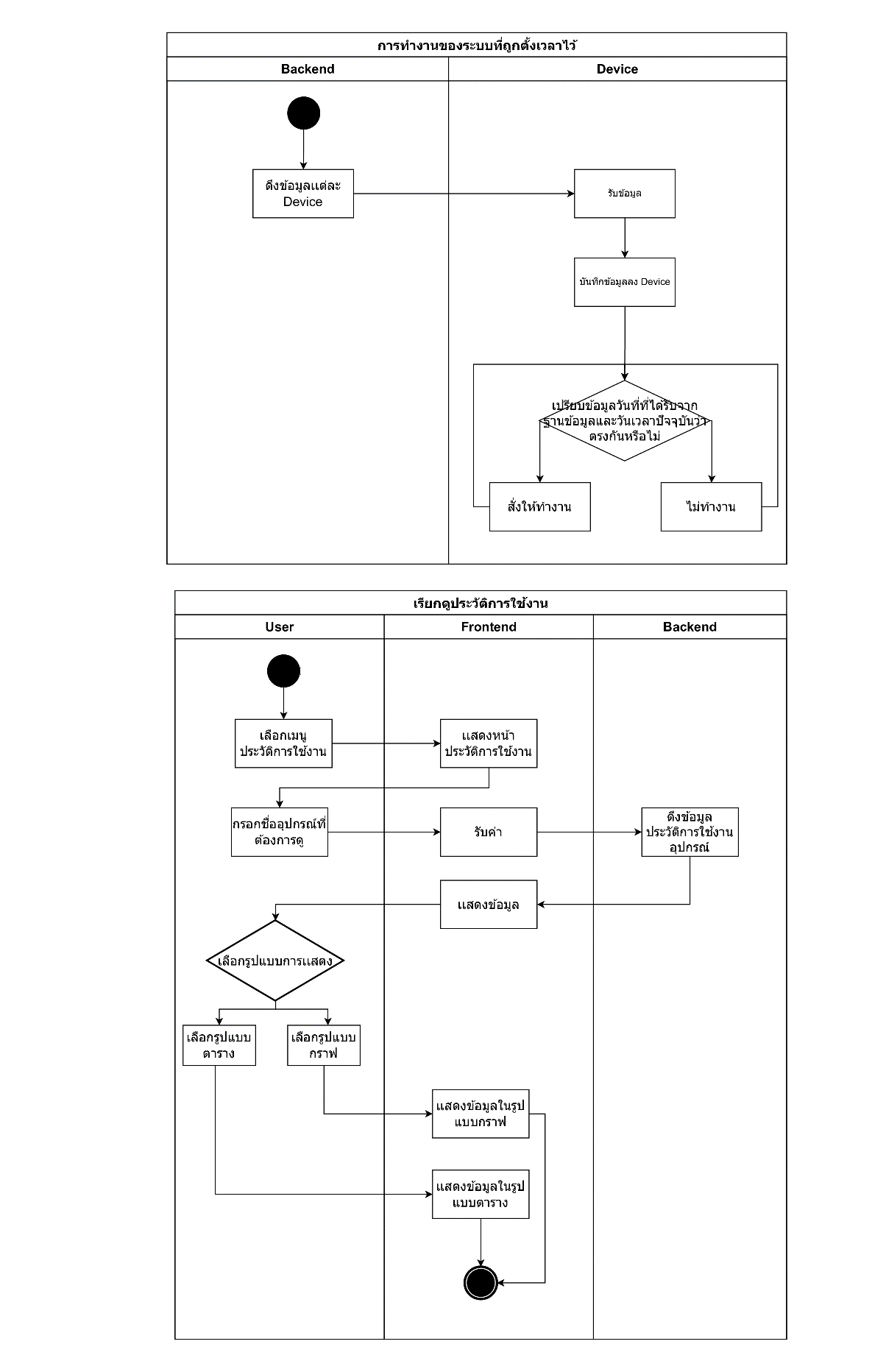
**ภาพที่ 3.11** Activity Diagram เพิ่มเวลาการทำงานของอุปกรณ์

3.3.11 Activity Diagram การทำงานของระบบที่ถูกตั้งเวลาไว้



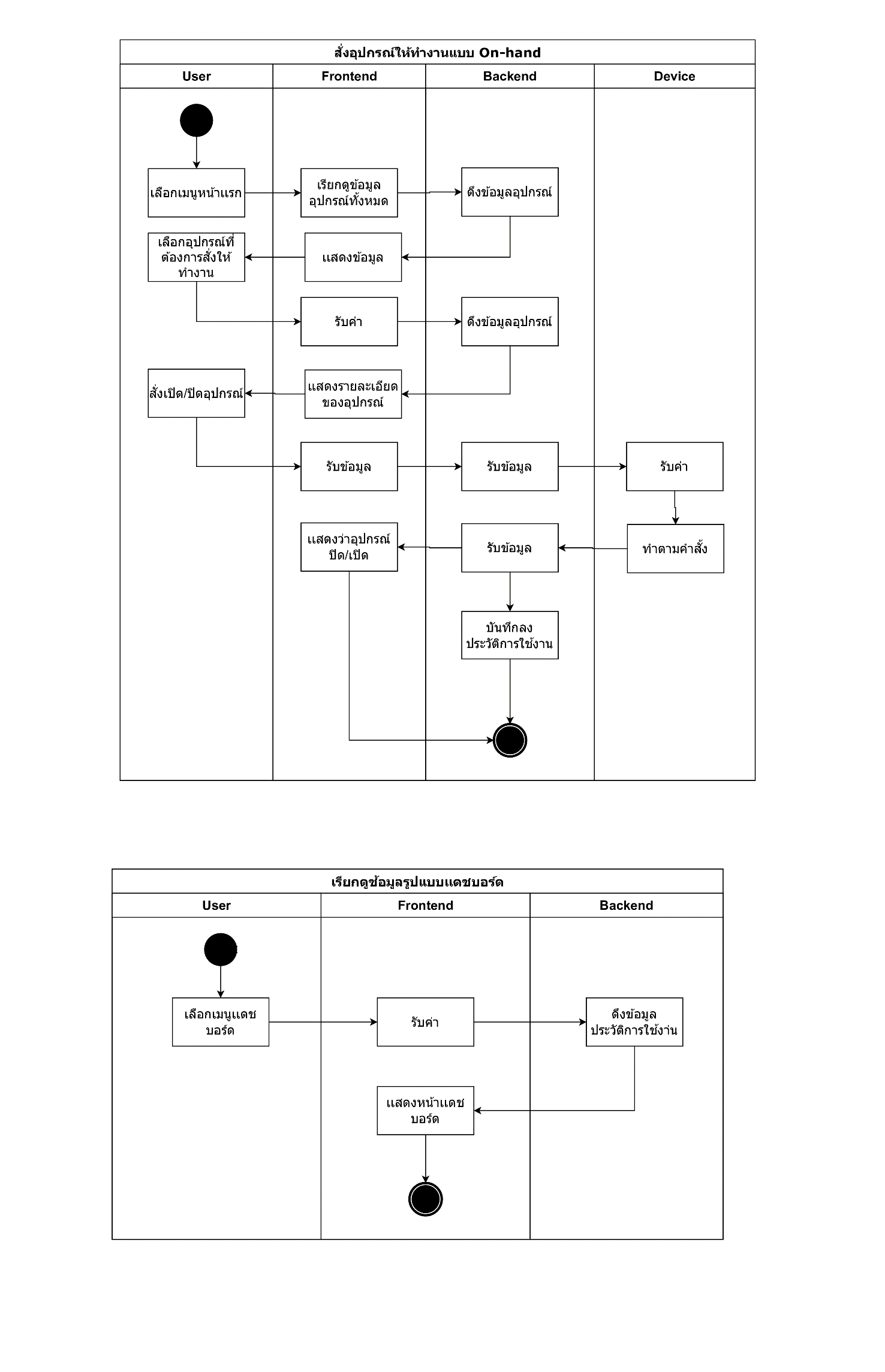
**ภาพที่ 3.12** Activity Diagram การทำงานของระบบที่ถูกตั้งเวลาไว้

3.3.12 Activity Diagram เรียกดูประวัติการใช้งาน



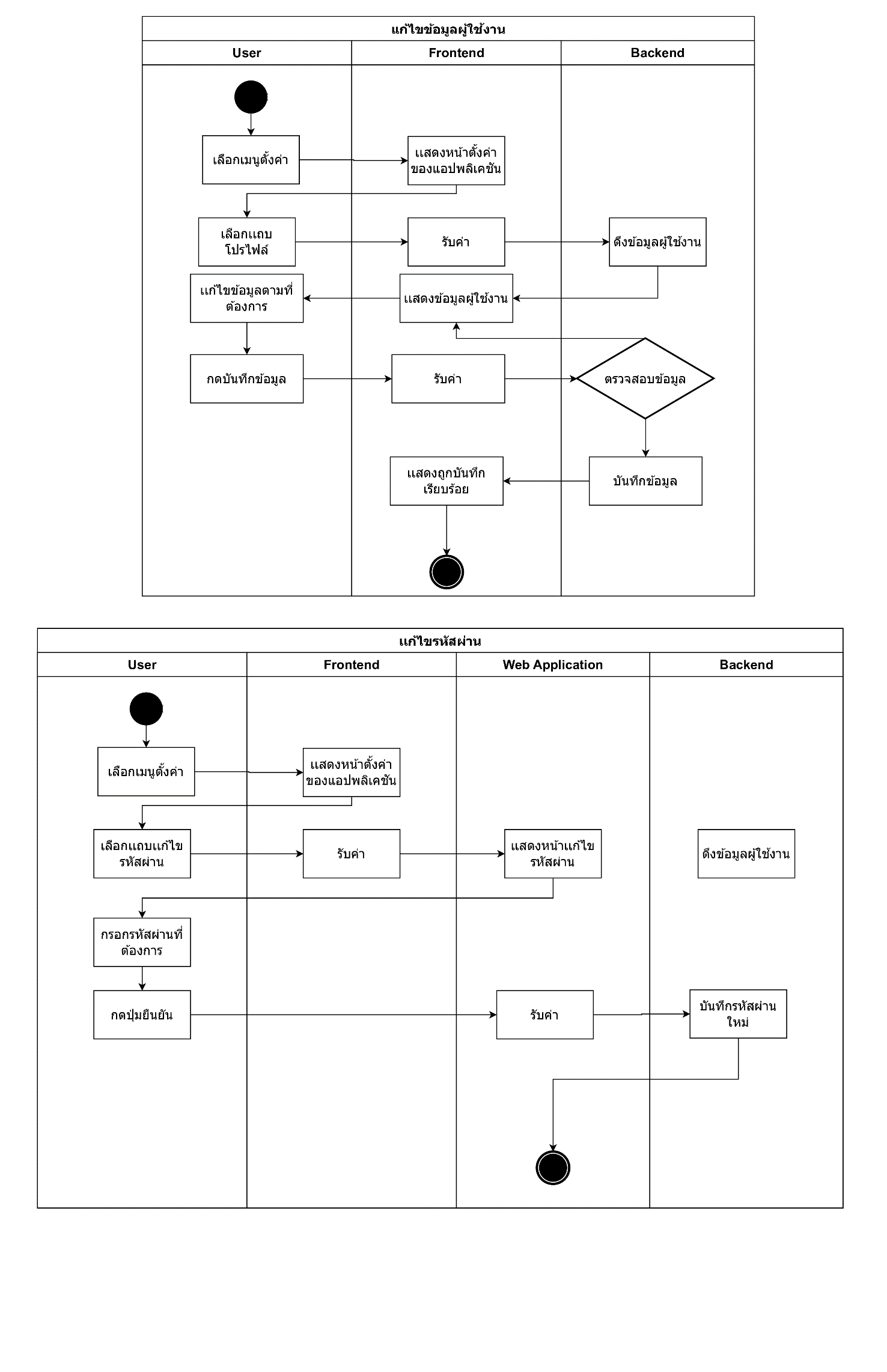
**ภาพที่ 3.13** Activity Diagram เรียกดูประวัติการใช้งาน

3.3.13 Activity Diagram เรียกดูข้อมูลรูปแบบแดชบอร์ด



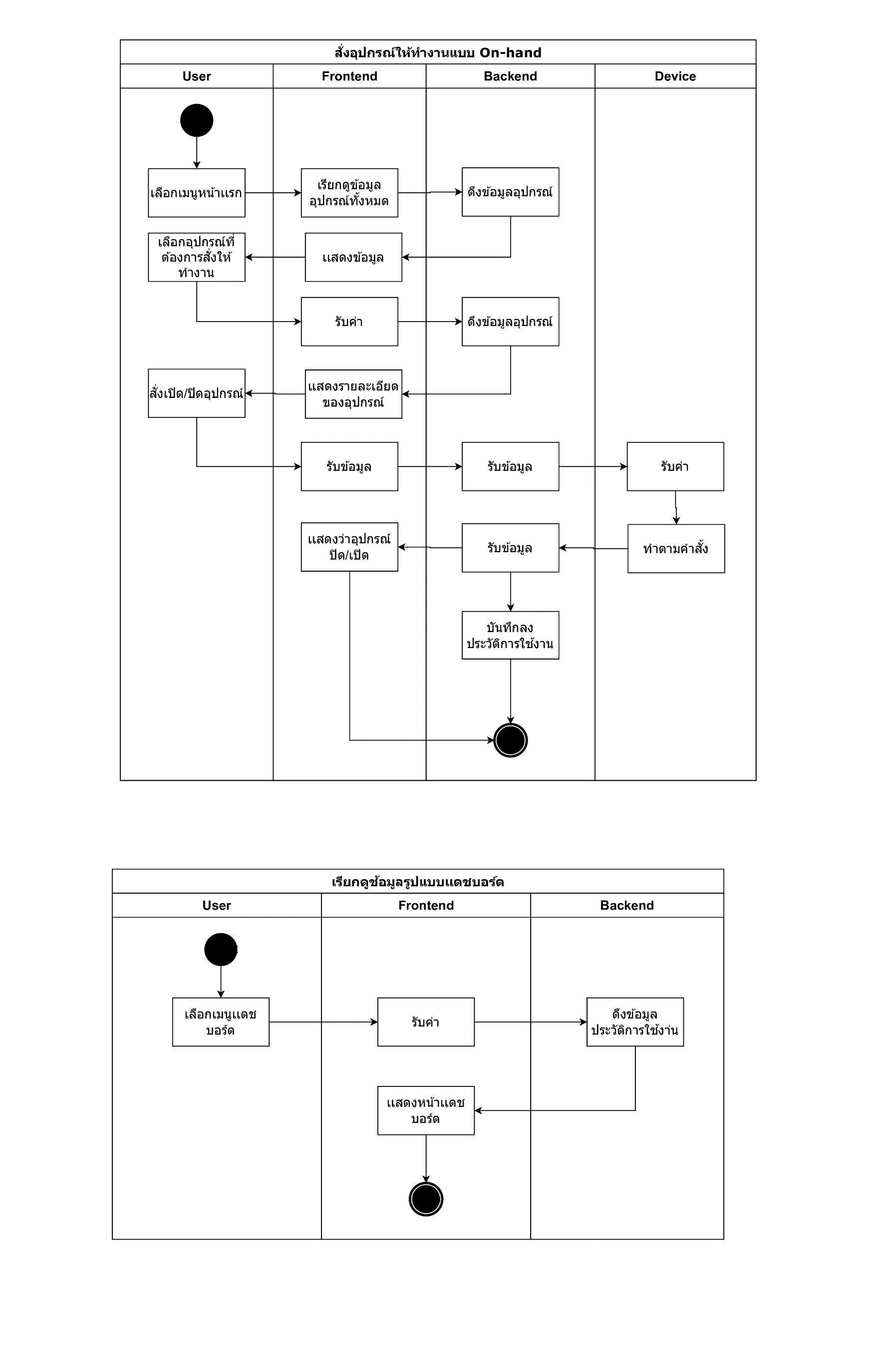
**ภาพที่ 3.14** Activity Diagram เรียกดูข้อมูลรูปแบบแดชบอร์ด

3.3.14 Activity Diagram แก้ไขข้อมูลผู้ใช้งาน



**ภาพที่ 3.15** Activity Diagram แก้ไขข้อมูลผู้ใช้งาน

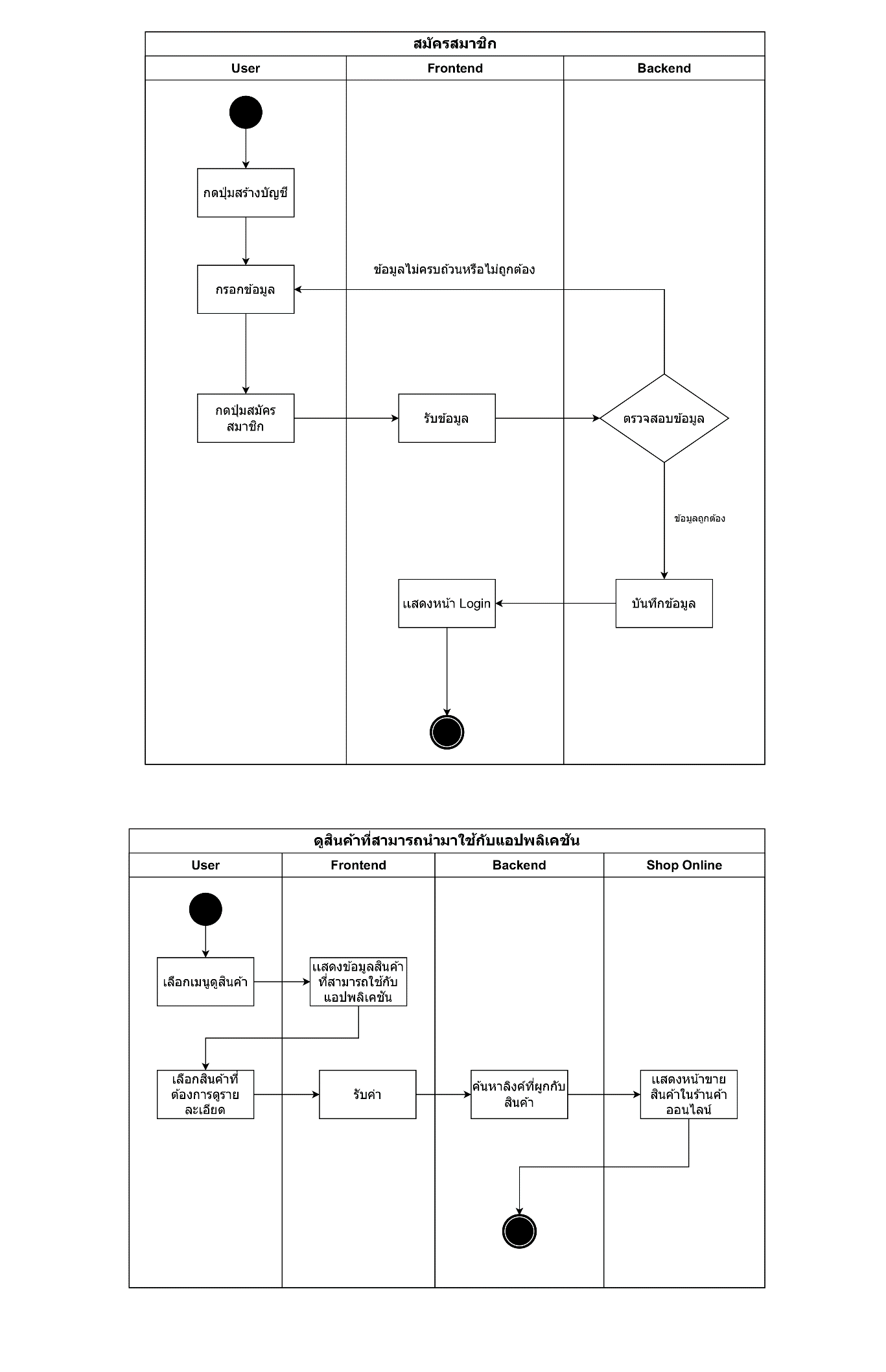
3.3.15 Activity Diagram สั่งอุปกรณ์ทำงานแบบ On-hand



**ภาพที่ 3.16** Activity Diagram สั่งอุปกรณ์ทำงานแบบ On-hand

‘

3.3.16 Activity Diagram สั่งอุปกรณ์ทำงานแบบ On-hand

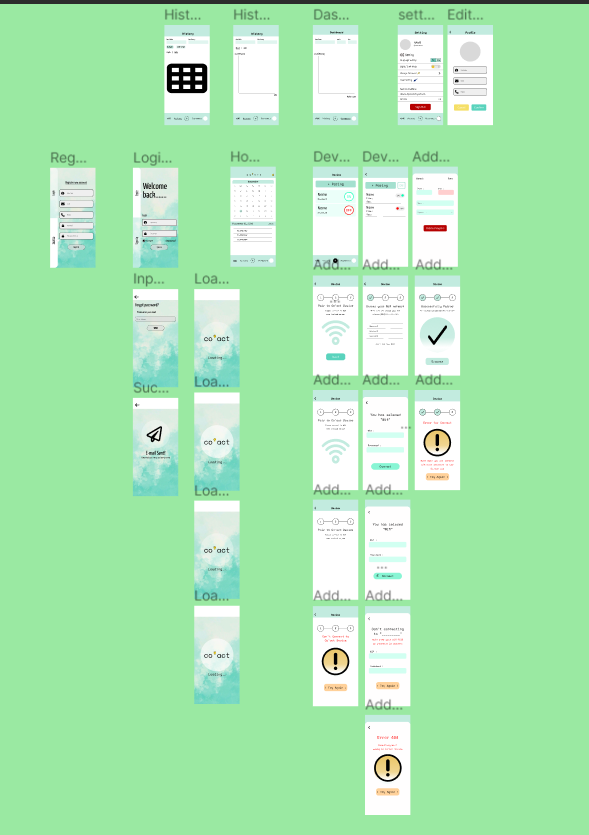


**ภาพที่ 3.17** Activity Diagram สั่งอุปกรณ์ทำงานแบบ On-hand

**3.4 การออกแบบโครงสร้างการจัดเก็บข้อมูล**

(\*\*\*สามารถนำเสนอด้วย JSON หรือ ER-Diagram ก็ได้\*\*\*)

**3.5 การออกแบบหน้าแอปพลิเคชั่น (Layout Application)**



**ภาพที่ 3.18** ออกแบบหน้า Mobile Application

(\*\*\*ควรแยกหน้าจอเพื่ออธิบายให้ชัดเจน\*\*\*)

**เอกสารที่เกี่ยวข้อง**

[1] Niran Kasri. (2562). Sonoff คืออะไร Sonoff อุปกรณ์สวิตช์ควบคุมไร้สายสั่งงานผ่านอิน

เตอร์เน็ต. [สืบค้นวันที่ 16 กันยายน 2566], จาก [https://www.blogsdit.com/](https://www.blogsdit.com/2019/06/so)2019

/06/so noff.html

[2] NETPIE 2020 Overview. (2563). NETPIE 2020 Overview. [สืบค้นวันที่ 8 ธันวาคม 2566],

จาก https://docs.netpie.io/overview-netpie.html

[3] automation360blog. (2560). ThingSpeak คืออะไร. [สืบค้นวันที่ 8 ธันวาคม 2566], จาก

https://automation360blog.wordpress.com/2017/11/25/what-is-thingspeak/

[4] JavaScript คืออะไร. (2560). JavaScript คืออะไร. [สืบค้นวันที่ 10 ธันวาคม 2566], จาก

https://www.mindphp.com/คู่มือ/73-คืออะไร/2187-java-javascript-คืออะไร.html

[5] Dektai Image. (2565). React.js #02 Component. [สืบค้นวันที่ 10 ธันวาคม 2566], จาก

<https://medium.com/@dektaiimage/react-js-02-component-925b1f615c1>

#:~:text=JSX%20(JavaScript%20XML)%20คือการ,Tag%20ปิด%20ทุกครั้งเช่น

[6] Jedsada Saengow. (2561). [React Native] คืออะไร ทำความรู้จัก และเริ่มต้นสร้าง Project.

[สืบค้นวันที่ 10 ธันวาคม 2566], จากhttps://medium.com/jed-ng/react-native-ทำ

ความรู้จัก-และเริ่มต้นสร้าง-project-91788ef6cac3

[7] HTML คืออะไร. (2564). HTML คืออะไร. [สืบค้นวันที่ 10 ธันวาคม 2566], จาก

https://www.bu.ac.th/knowledgecenter/executive\_journal/july\_sep\_11/pdf/aw

32.pdf

[8] CSS คืออะไร. (2561). CSS คืออะไร. [สืบค้นวันที่ 10 ธันวาคม 2566], จ าก

<https://blog.sogoodweb.com/Article/Detail/79237/CSS-คืออะไร-มีประโยชน์->

อย่างไร

[9] Firebase คืออะไร. (2563). Firebase คืออะไร. [สืบค้นวันที่ 10 ธันวาคม 2566], จาก

http://www.thaiall.com/firebase/index.html

[10] Firebase Cloud Functions คืออะไร. (2560). Firebase Cloud Functions คืออะไร. [สืบค้น

วันที่ 10 ธันวาคม 2566]. จาก https://medium.com/firebasethailand/

[11] Firebase Cloud Messaging คืออะไร. (2563). Firebase Cloud Messaging คืออะไร. [สืบค้น

วันที่ 10 ธันวาคม 2566]. จาก https://hmong.in.th/wiki/Firebase\_Cloud\_Messaging

[12] Firebase Realtime Database คืออะไร. (2559). Firebase Realtime Database คืออะไร.

[สืบค้นวันที่ 10 ธันวาคม 2566], จาก <https://medium.com/firebasethailand/-fire>

base-realtime-database

[13] API. (2563). API คืออะไร เอพีไอ คือ ช่องทางหนึ่งที่จะเชื่อมต่อระบบเรากับ กับเว็บไซต์ผู้ให้บริ

การ API จากที่อื่น. [สืบค้นวันที่ 10 ธันวาคม 2566], จาก https://www.mindphp.com/

คู่มือ/73-คืออะไร/2038-api-คืออะไร.html

[14] ปริยา เสนา. (2556). ภาษาซีและซีพลัสพลัส. [สืบค้นวันที่ 10 ธันวาคม 2566], จาก <https://de>

kinter2010.wordpress.com/บทเรียน-2/หน่วยที่-2/ภาษาคอมพิวเตอร์/ภาษาระดับสูง/

ภาษาซีและซีพลัสพลัส/

[15] บริษัท เอไอซีเอส จำกัด. (2564). RDUINO IDE คืออะไร มีวิธีการติดตั้งโปรแกรมอย่างไร และการ

ใช้โปรแกรมยังไงกันนะ. [สืบค้นวันที่ 10 ธันวาคม 2566], จาก <https://www.ai-corpor>ati

on.net/2021/11/18/what-is-arduino-ide/#:~:text=Arduino%20IDE%20เป็นโปรแก

รมที่,เขียนเข้าสู่บอร์ด%20Arduino

[16] Knowledge Base. (2565). NodeJSคืออะไร ?มาทำความรู้จักตัวช่วยพัฒนาเว็บไซต์ยอดนิยม !.

[สืบค้นวันที่ 10 ธันวาคม 2566], จาก https://blog.openlandscape.cloud/nodejs