**ชื่อผลงานภาษาไทย ระบบจัดการการแจ้งเตือนผ่านไลน์**

**ชื่อผลงานภาษาอังกฤษ Notification management system via Line**

ชวัลวิทย์ เลาหาง1, ณัฐชยา เตปันวงศ์1, พุฒิพงศ์ พนมไพร1, สมนึก สินธุปวน1\*, ก่องกาญจน์ ดุลยไชย1  และอรรถวิท ชังคมานนท์1

Chawanwit Laohang1, Natchaya Tepanwong1, Puttipong Panomprai1, Somnuek Sinthupuan1\*, Kongkarn Dullayachai1 and Attawit Changkamanon1

*1สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้เชียงใหม่ 50290 ประเทศไทย*

\*ผู้นิพนธ์ประสานงาน: สมนึก สินธุปวน อีเมล: somnuk@mju.ac.th

**บทคัดย่อ:**

การให้บริการในการแจ้งปัญหาของบริษัทสำหรับลูกค้า บริษัทได้ให้บริการกับลูกค้า โดยการให้ลูกค้าสามารถทำการแจ้งปัญหาการให้บริการเกี่ยวกับโปรแกรมที่ใช้งานผ่านบริการ line ของบริษัทโดยตรงโดยไม่มีระบบของบริษัทรองรับ ซึ่งการให้บริการดังกล่าวเกิดปัญหาคือ ลูกค้ามีการส่งข้อความปัญหาจำนวนมาก ไม่มีผู้รับเรื่องโดยตรงเพื่อไปดำเนินการ และข้อมูลการแจ้งปัญหาของลูกค้าเกิดการสูญหาย ดังนั้นงานวิจัยจึงมีวัตถุประสงค์เพื่อ พัฒนาระบบจัดการการแจ้งเตือนผ่านไลน์ โดยใช้เฟรมเวิร์ค React TypeScript ในการพัฒนาระบบและจัดเก็บข้อมูลในระบบฐานข้อมูล Firebase และการทดสอบระบบแบบอัตโนมัติโดยใช้โปรแกรม Cypress โดยระบบประกอบด้วยส่วนที่สำคัญคือ การจัดการค่าคงที่ แจ้งรายงานปัญหา และการแจ้งเตือนผ่านไลน์ ซึ่งมีผู้ดูแลระบบเป็นผู้ดูแลการรับเรื่องและการแจ้งเตือนสำหรับผู้ที่เกี่ยวข้องในการรับเรื่องผ่านเว็ปไซด์ของบริษัท การทดสอบระบบแบบอัตโนมัติได้กำหนดหัวข้อการทดสอบ 5 หัวข้อ ได้แก่ การเข้าสู่ระบบ การจัดการค่าคงที่ (เพิ่มบริษัท เพิ่มโครงการ) แจ้งรายงานปัญหา และการแจ้งเตือนผ่านไลน์ รวมถึงการทดสอบการทำงานในภาพรวม โดยแบ่งเป็นการทดสอบระบบจำนวน 5 รอบ พบว่า การทำงานของระบบสามารถแก้ไขทุกฟังก์ชันและภาพรวมได้อย่างครบถ้วนและถูกต้องตรงตามเวลาที่บริษัทกำหนด การวิจัยได้ดำเนินการกับกลุ่มตัวอย่างเป็นบริษัทจำนวน 2 บริษัท ในระยะเวลา 6 เดือน ทั้งสองบริษัทได้แจ้งรายงานปัญหาผ่านระบบจำนวนทั้งหมด 120 ปัญหา บริษัททเอได้ทำการแจ้งปัญหา 60 ปัญหาและบริษัทบี 60 ปัญหา โดยปัญหาที่ถูกแก้ไขได้ของบริษัทเอ คิดเป็นร้อยละ 58.3 และบริษัทบี คิดเป็นร้อยละ 76.9 ซึ่งพบว่าสูงกว่าการจัดการปัญหาแบบเดิมซึ่งบริษัทสามารถจัดการปัญหาได้เพียงร้อยละ 50 เท่านั้น

คำสำคัญ: การแจ้งเตือน, การรายงานปัญหา, ไลน์, การทดสอบระบบ, การทดสอบอัตโนมัติ

**Abstract**:

The company's maintenance services, and problem reporting are provided to customers. By allowing customers to report service problems regarding programs used directly through the company's line service without the company's system supporting it. The problem with providing such service is that customers are sending many problem messages. There is no direct recipient of the matter to act and customer problem reporting information is lost. Therefore, the research objective is to develop the notification management system via LINE. The system is developed using the React TypeScript framework, stores data in the Firebase database, and automates system testing using the Cypress program. The system consists of important parts: constant management, reporting a problem, and receiving notifications via LINE. There is an administrator who takes care of receiving matters and notifications for those involved in receiving matters through the company's website. Automated system testing has five test topics: Login. Constant management (add company, add project) Report a problem and receive notifications via Line, including overall functional testing. Testing was divided into 5 rounds of system testing, and it was found that the operation of the system was able to fix every function and overall picture completely and correctly within the time specified by the company. The research was conducted with a sample of two companies over a period of six months. Both companies reported a total of 120 problems with the system. Company A reported 60 problems, and Company B reported 60 problems. The problems were resolved, with Company A accounting for 58.3 percent and Company B accounting for 76.9 percent. This was found to be higher than traditional problem management, in which the company was able to manage only 50 percent of problems.

Keyword: Notification, Problem report, Line, Testing, Automate-test

**ที่มาและความสำคัญ:**

บริษัทคราฟติ้งแล็บ จำกัด ดำเนินธุรกิจเกี่ยวกับการให้คำปรึกษาด้านการตลาดและพัฒนาซอฟต์แวร์ตามความต้องการของลูกค้า ทางบริษัทได้พัฒนาระบบให้กับลูกค้าหลายๆ โปรเจกต์และการทำบริการหลังการขายและการแจ้งปัญหาของลูกค้าส่วนใหญ่จะส่งผ่านแอปพลิเคชัน Line หรือ ทาง Email ของบริษัท ซึ่งอาจทำให้เกิดปัญหาต่าง ๆ ตามมา เช่น การสื่อสารที่ไม่เป็นระบบ อาจทำให้ข้อมูลหรือการแจ้งปัญหาของลูกค้ากระจายอยู่ในหลายแชทหรือกลุ่ม ซึ่งอาจทำให้ข้อมูลสับสนหรือหายไปได้ และทำให้กระบวนการติดตามและแก้ไขปัญหาของลูกค้ายากขึ้น ความล่าช้าในการตอบกลับ การสื่อสารทางแชทอาจไม่ครอบคลุมข้อมูลหรือรายละเอียดที่เพียงพอในการอธิบายปัญหาหรือข้อสงสัย ทำให้การวิเคราะห์และแก้ไขปัญหายากขึ้นหรือต้องใช้เวลามากขึ้น และการสูญเสียประสิทธิภาพในการแก้ไขปัญหา อาจทำให้ข้อมูลหรือรายละเอียดที่ลูกค้าแจ้งไปหาผู้พัฒนาหรือทีมสนับสนุนไม่เพียงพอในการวิเคราะห์และแก้ไขปัญหา ทำให้การแก้ไขปัญหาล่าช้าหรือไม่ได้ผลอย่างที่ควร

จากปัญหาที่กล่าวมานั้น เพื่อการดำเนินการที่ราบรื่นและมีประสิทธิภาพ จึงเป็นที่มาของการพัฒนาระบบบริการดูแลและบำรุงรักษาระบบโปรแกรม(Craft MA) ที่เป็นระบบที่ใช้ในการแจ้งปัญหาของลูกค้าผู้ใช้งาน และมีในส่วนผู้ให้บริการที่จะมีการ เพิ่ม ลบ แก้ไข ข้อมูลบริษัท Project ต่าง ๆ ของลูกค้าที่ทางบริษัทได้ให้บริการอยู่ มีการสร้างสัญญา ให้รู้ว่าลูกค้าบริษัทไหนที่ยังมีการใช้บริการอยู่ และการรายงานปัญหาจากการใช้งานของลูกค้า และเก็บรวบรวมข้อมูลรายงานปัญหาและนำไปแก้ไข เพื่อการบริการหลังการขายที่มีประสิทธิภาพมากขึ้น และในขั้นตอนของการทดสอบระบบจะใช้ Cypress เพื่อทำการทดสอบระบบแบบอัตโนมัติแทนการทดสอบระบบแบบ Manual Test เนื่องจากการทำการทดสอบระบบแบบอัตโนมัติจะช่วยลดระยะเวลาของการทดสอบระบบ และให้ผลการทดสอบระบบที่แม่นยำมากขึ้น

**วัตถุประสงค์:**

1. เพื่อพัฒนาระบบบริการดูแลและบำรุงรักษาระบบโปรแกรมหลังการขาย

2. เพื่อทำการทดสอบระบบแบบอัตโนมัติแบบ End to End

**งานวิจัยที่เกี่ยวข้อง:**

กฤษณะ พิริยะกิจไพบูลย์ (2547) ได้ทำการวิจัยเรื่องการพัฒนาเครื่องมือสร้างข้อมูลทดสอบเพื่อสนับสนุนการทดสอบซอฟต์แวร์จากสคีมาของฐานข้อมูล งานวิจัยนี้ได้นำเสนอเครื่องมือที่ช่วยผู้ทดสอบสร้างข้อมูลทดสอบได้ง่ายขึ้น โดยเครื่องมือจะสุ่มสร้างข้อมูลทดสอบตามเงื่อนไขที่ได้มาจากผู้ทดสอบและฐานข้อมูลต้นทาง ซึ่งจากการทดสอบการใช้งานเครื่องมือกับระบบงานจริง พบว่าเครื่องมือสามารถสร้างข้อมูลทดสอบได้ตรงกับคุณสมบัติของข้อมูลที่มีอยู่จริง โดยข้อมูลทดสอบที่สร้างได้จะมีความถูกต้องตามการขึ้นต่อกันเชิงตรรกะและความคงสภาพในการอ้างอิงของสคีมาของฐานข้อมูลด้วย

พนารัตน์ เชี่ยววิทย์ (2545) ได้ทำการวิจัยเรื่องการพัฒนาเครื่องมือการทดสอบซอฟต์แวร์ในระดับหน่วยสำหรับกระบวนการพัฒนาเชิงวัตถุ งานวิจัยนี้ได้นำเสนอเครื่องมือ ทดสอบซอฟต์แวร์ในระดับหน่วยสำหรับกระบวนการพัฒนาเชิงวัตถุ โดยใช้ภาษาจาวาในการพัฒนา ซึ่งพิจารณาเงื่อนไขพื้นฐานต่างๆ ของภาษาจาวา เพื่อนำมาใช้ในการวิเคราะห์กรณีทดสอบ ต่างๆ โดยใช้ Basis Path Model สำหรับวิเคราะห์ Cyclomatic Complexity ซึ่งจากการทดลองพบว่า ภาษาจาวามีข้อจำกัดในการวิเคราะห์กรณีทดสอบ ดังนั้น เครื่องมือที่พัฒนานี้ จึงจำเป็นต้องมีการเพิ่มเติม Keyword บางตัว เพื่อ ทำให้โปรแกรมนั้นมีความถูกต้องสมบูรณ์ที่สุดก่อนที่จะนำมาวิเคราะห์กรณีทดสอบต่อไป โดยใช้ 20 Source Codes ที่ครอบคลุมเงื่อนไขทั่วๆ ไปของภาษาจาวา เพื่อทำการทดสอบ ความถูกต้องของภาษา การสร้างกรณีทดสอบและกระบวนการแสดงผลกรณีทดสอบ การพัฒนาต่อไปนั้น เครื่องมือการทดสอบซอฟต์แวร์นี้ ควรมีการสร้างชุดข้อมูล สำหรับทดสอบโปรแกรม และมีการนำเข้าไปทดสอบโดยอัตโนมัติ นอกจากนี้ อัลกอริทึมอื่นๆ อาจถูกนำมาใช้ในการทำ Parser Generator และวิเคราะห์กรณีทดสอบเพื่อให้เครื่องมือนี้ มีความสมบูรณ์มากยิ่งขึ้น

สันต์ทศน์ สุริยันต์ (2543) ได้ทำการวิจัยเรื่องการพัฒนาเครื่องมือสำหรับจัดการกระบวนการวางแผนการทดสอบซอฟต์แวร์ วิทยานิพนธ์นี้ได้ออกแบบและพัฒนาเครื่องมือสำหรับจัดการกระบวนการวางแผนการทดสอบซอฟต์แวร์ใน 3 ระดับ ได้แก่ การทดสอบระดับหน่วย การทดสอบการรวม และการทดสอบระบบ เครื่องมือดังกล่าวสามารถเก็บรวบรวมข้อมูลที่เกิดจากการทดสอบโดยอิงกับมาตรฐานของเอกสารที่ใช้ในการทดสอบ นอกจากนี้ยังช่วยติดตามสถานะของการทดสอบได้อีกด้วย เครื่องมือดังกล่าวได้รับการทดสอบโดยใช้ข้อมูลจริงจากบริษัทพัฒนาซอฟต์แวร์แห่งหนึ่ง โดยสามารถเก็บรวบรวมข้อมูลได้ทั้งหมดและออกรายงานได้ตามที่ผู้ใช้ต้องการ

ธนพล สิชฌนุกฤษฎ์ (2544) ได้ทำการวิจัยเรื่องการพัฒนาระบบจัดการกรณีทดสอบซอฟต์แวร์ งานวิจัยนี้ได้พัฒนาเครื่องมือระบบจัดการกรณีทดสอบซอฟต์แวร์ เพื่อนำมาช่วยผู้ทดสอบในการสร้างกรณีทดสอบ และนอกจากนั้นยังช่วยในการเก็บข้อมูลกรณีทดสอบ การตรวจสอบ การติดตาม การแก้ไข เพิ่ม-ลด กรณีทดสอบผู้ที่ทดสอบใช้ในการทดสอบทั้งหมด เพื่อให้การทดสอบทำได้สะดวกยิ่งขึ้น และเป็นการลดภาระของผู้ทดสอบอีกด้วย โดยการเก็บข้อมูลกรณีทดสอบ และผลการทดสอบภายในระบบ

Chai Phonbopit ได้เขียนยทความเกี่ยวกับวิธีการทำระบบสมัครสมาชิก และเข้าสู่ระบบ โดยใช้ React.js ร่วมด้วย React Router และ Firebase Authentication ซึ่งเป็นบริการของทาง Firebase ในส่วนของการทำ Routing คือ เมื่อคลิกที่ลิงก์แล้วเปลี่ยนไปยังหน้าเว็บเพจปลายทางที่เราได้ทำการกำหนดไว้ได้นั่นเอง

[Prudchaya Palee](https://prudchayapalee.medium.com/?source=post_page-----606ac55a7287--------------------------------) เขียนบทความเกี่ยวกับการสร้าง REST API โดยใช้ Firebase โดยใช้บริการของ FIrebase Functions ซึ่งบริการตัวนี้เป็น Serverless ที่รันบน NodeJS ซึ่งใช้ภาษา Javascript ในการในการเขียนได้ แล้วยังใช้ร่วมกับเฟรมเวิร์คอื่น ๆ ได้ดวยเช่น Express Loopback และใช้บริการ Cloud firestore , Storage เป็นที่เก็บข้อมูล

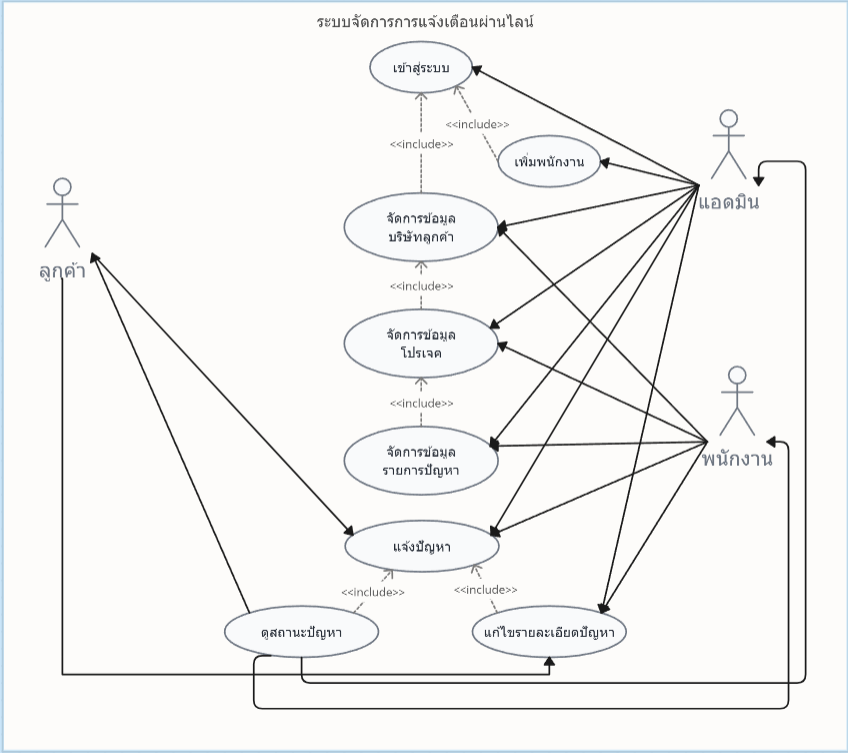
Chai Phonbopit นำเสนอบทความเกี่ยวกับการ Deploy งานของเราขึ้น Hosting แน่นอนว่าการจะหา Hosting เพื่อฝากไฟล์เว็บของเราไว้ บางทีก็มีค่าใช้จ่ายในการเช่าใช้งานแต่ก็ยังมีหลายๆที่ที่ไม่เสียค่าใช้จ่าย Firebase Hosting เป็น Hosting ที่สามารถใช้งานได้ฟรีนอกจากนี้ตัว Firebase Console ยังมีหน้า Hosting ให้เราดู State ด้วยว่าเว็บมีการดาวน์โหลดเท่าไหร่ดู History การ Deploy รวมถึงสามารถ Rollback กลับไปเวอร์ชั่นเก่ากว่าได้และยังสามารถ Connect Domain เพื่อ Custom domain เองได้

**วิธีการศึกษาและพัฒนา:**

1). การพัฒนาระบบบริการดูแลและบำรุงรักษาระบบโปรแกรม

1.1 วิเคราะห์และออกแบบระบบ

1.1.1 ยูสเคส(Use Case)



**รูปที่ ค.1** ยูสเคสไดอะแกรมของผู้ใช้งาน(User) ผู้ดูแลระบบ(Admin) และพนักงาน(Employee)

1.2 การพัฒนาระบบ

ขั้นตอนแรกคือการเพิ่มข้อมูลบริษัทเข้าสู่ระบบ โดยผู้ดูแลกดปุ่มเพิ่มข้อมูลบริษัทเพื่อเข้าสู่หน้าฟอร์มเพิ่มข้อมูล บริษัท แล้วกรอกข้อมูลของบริษัทลูกค้า

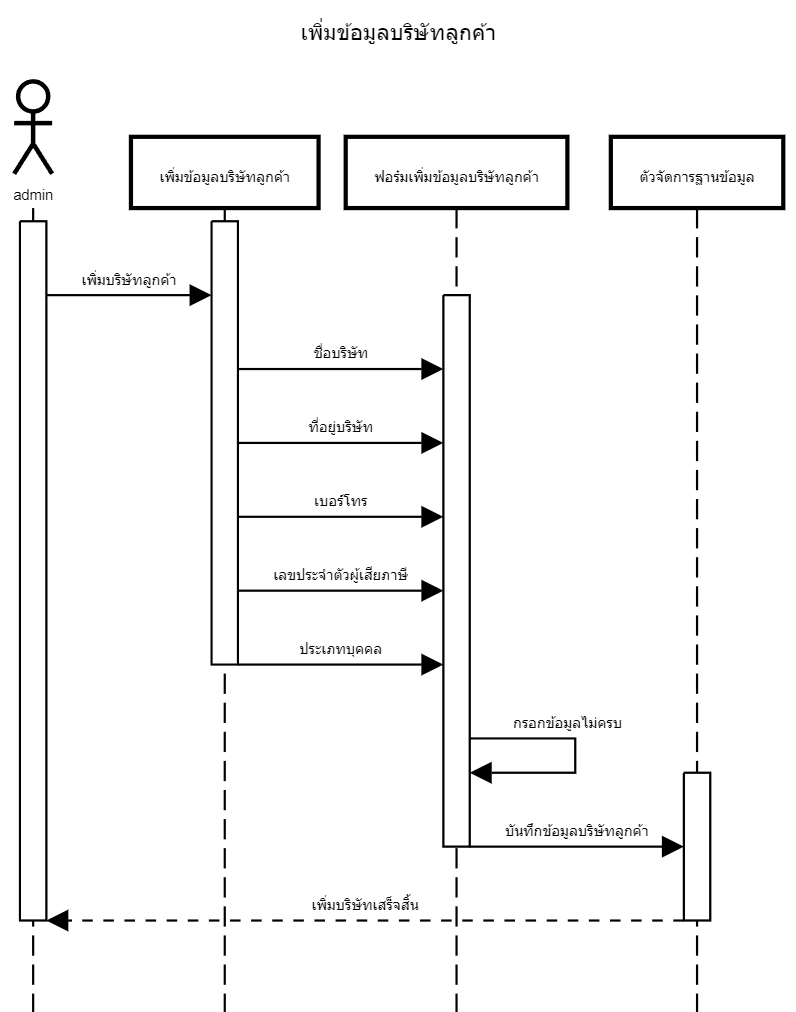
A blue and white text box

Description automatically generated

A screenshot of a phone

Description automatically generated

A blue and white text box

Description automatically generated

**รูปที่ ค.2** ภาพซีเควนซ์ได้อะแกรมหน้าจอการเพิ่มบริษัท

ขั้นตอนการเพิ่มโปรเจกต์เพื่อให้แต่ละโปรเจกต์สามารถแจ้งปัญหาเข้ามาได้หลังจากเพิ่มบริษัทแล้ว โดยเลือกบริษัทที่ ต้องการเพิ่มโปรเจกต์จากนั้นกดปุ่มเพิ่มโปรเจกต์เพื่อเข้าสู่หน้าฟอร์มแล้วกรอกข้อมูลโปรเจกต์

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

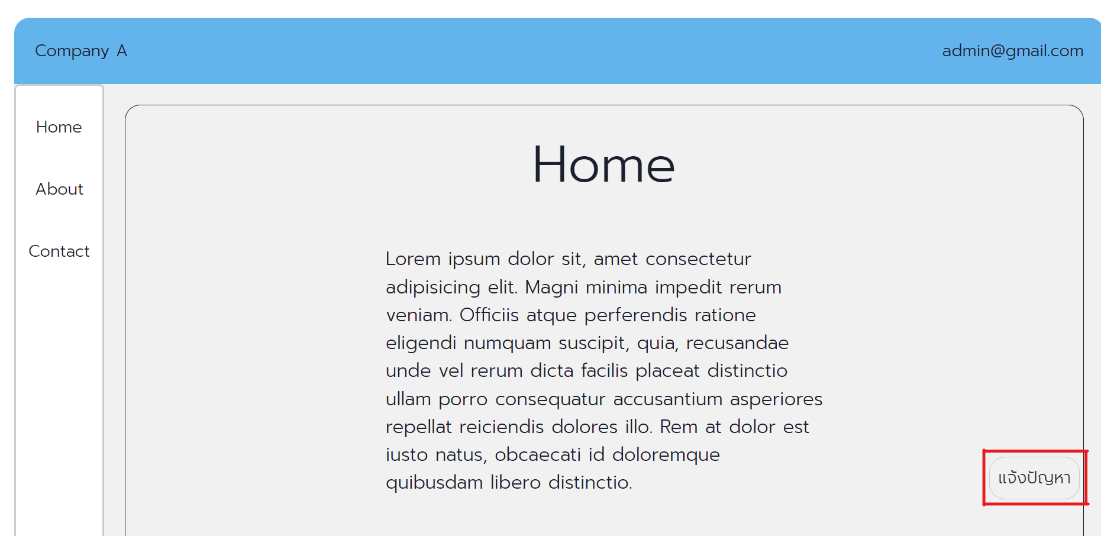
A screenshot of a computer

Description automatically generated



**รูปที่ ค.3** ภาพซีเควนซ์ได้อะแกรมหน้าเพิ่มโปรเจกต์

ขั้นตอนการแจ้งปัญหา กดที่ปุ่มแจ้งปัญหาจากหน้าเว็บไซต์ ของลูกค้า จากนั้นกดปุ่มแจ้งปัญหาการใช้งาน จากนั้นกรอกปัญหากับรายละเอียด เมื่อกดบันทึก จะมีการแจ้งเตือนผ่าน Line notify และ Email ไปยังผู้ดูแลระบบ



A screenshot of a computer

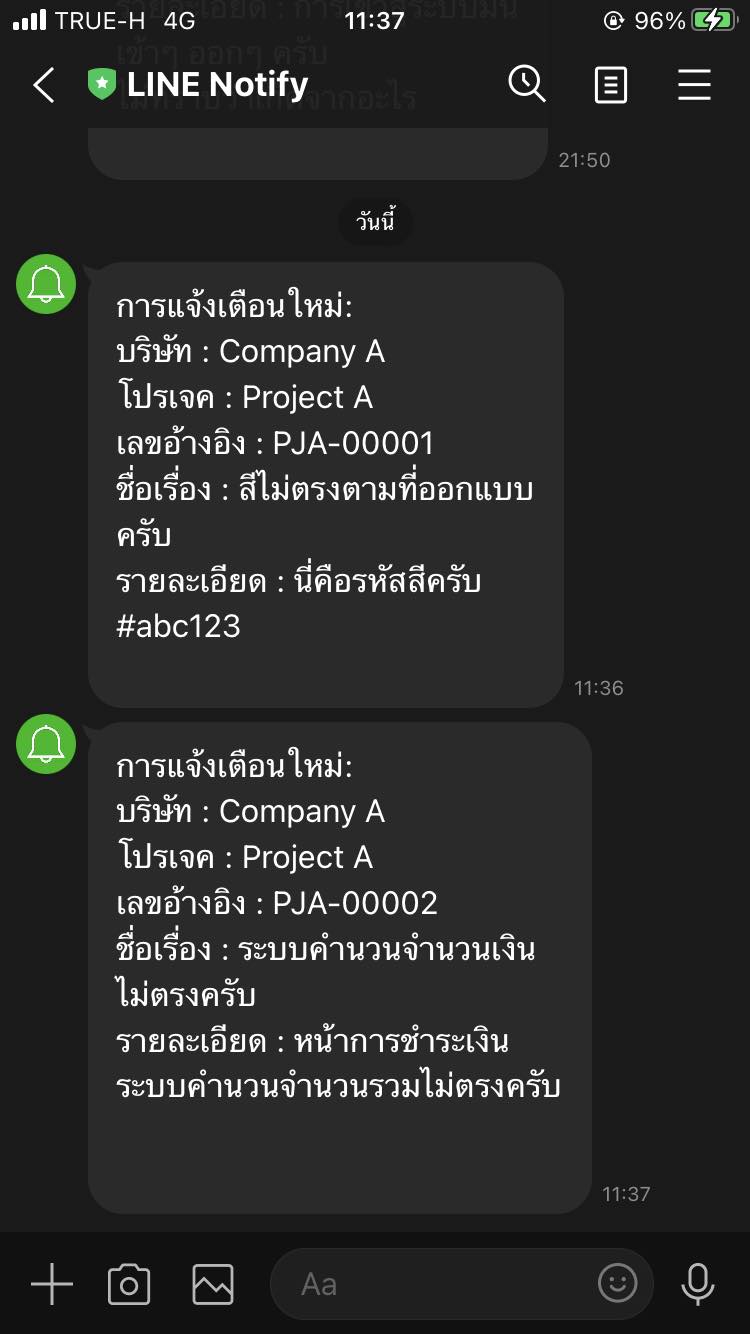
Description automatically generated

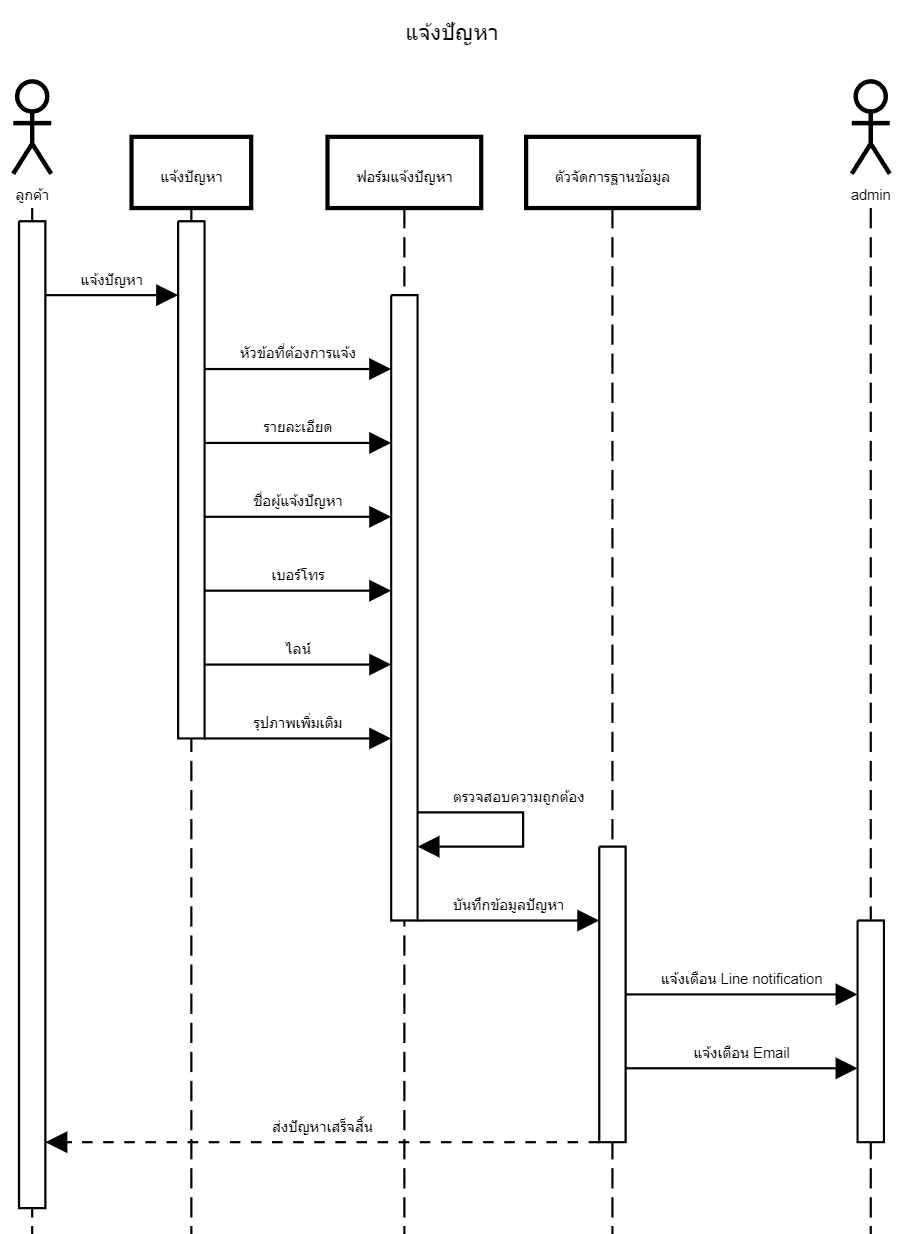
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated





**รูปที่ ค.4** ภาพแสดงซีเควนซ์ไดอะแกรมการแจ้งปัญหา

ขั้นตอนการในการตอบปัญหาเลือกบริษัทและโปรเจกต์ที่มีการแจ้งปัญหาจาก Line Notify เลือกปัญหาที่ต้องการเปลี่ยนสถานะ กดที่แถวปัญหานั้นเพื่อเข้าสู่หน้าตอบปัญหาและเปลี่ยนสถานะปัญหา

A screenshot of a computer

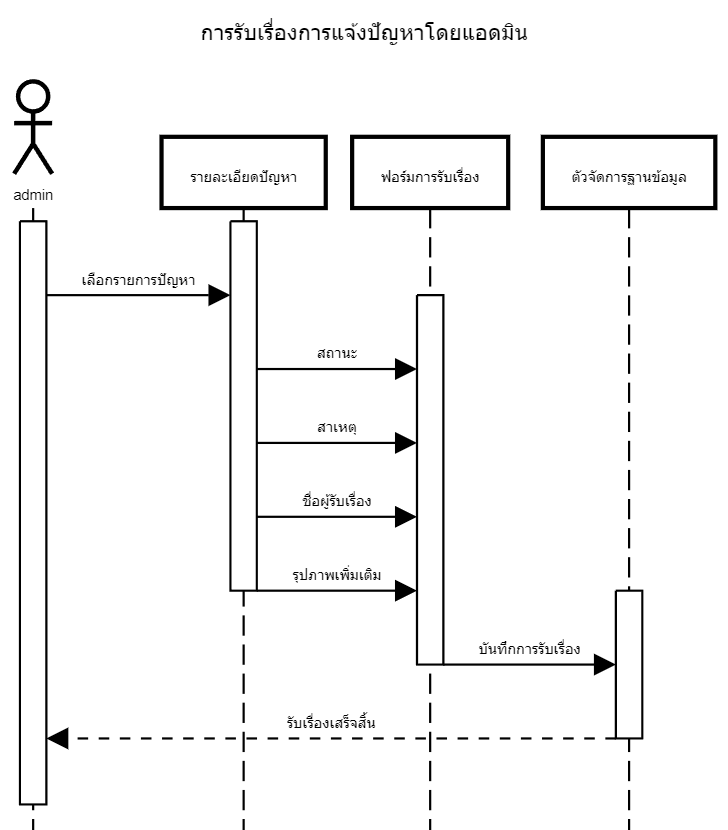
Description automatically generated

A screenshot of a computer

Description automatically generated

A blue and white rectangular box with black text

Description automatically generated



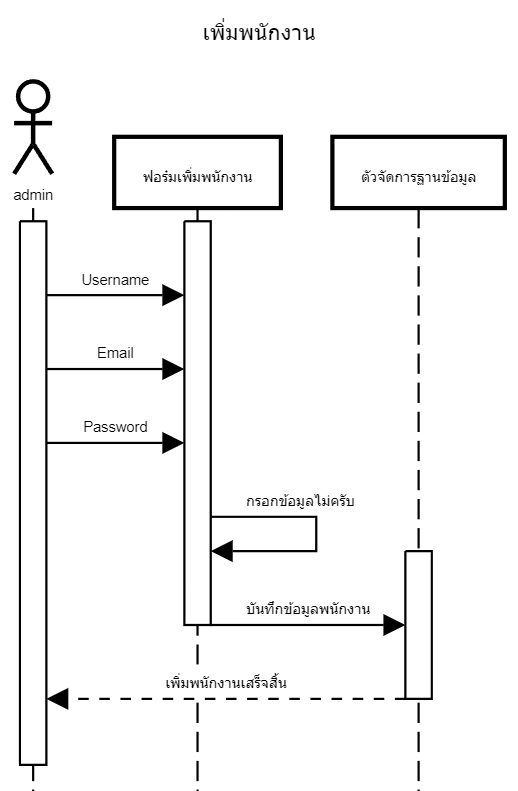
**รูปที่ ค.5** ภาพซีเควนซ์ไดอะแกรมการตอบและการเปลี่ยนสถานะของปัญหาของ Admin

ขั้นตอนการเพิ่มผู้เข้ามาตอบปัญหา โดยกดที่รูปโปรไฟล์ที่อยู่มุมขวาบนจากนั้นกดที่แถบเพิ่มพนักงานเพื่อเข้าสู่ หน้าเพิ่มพนักงานแล้วทำการกรอกข้อมูลพนักงาน

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

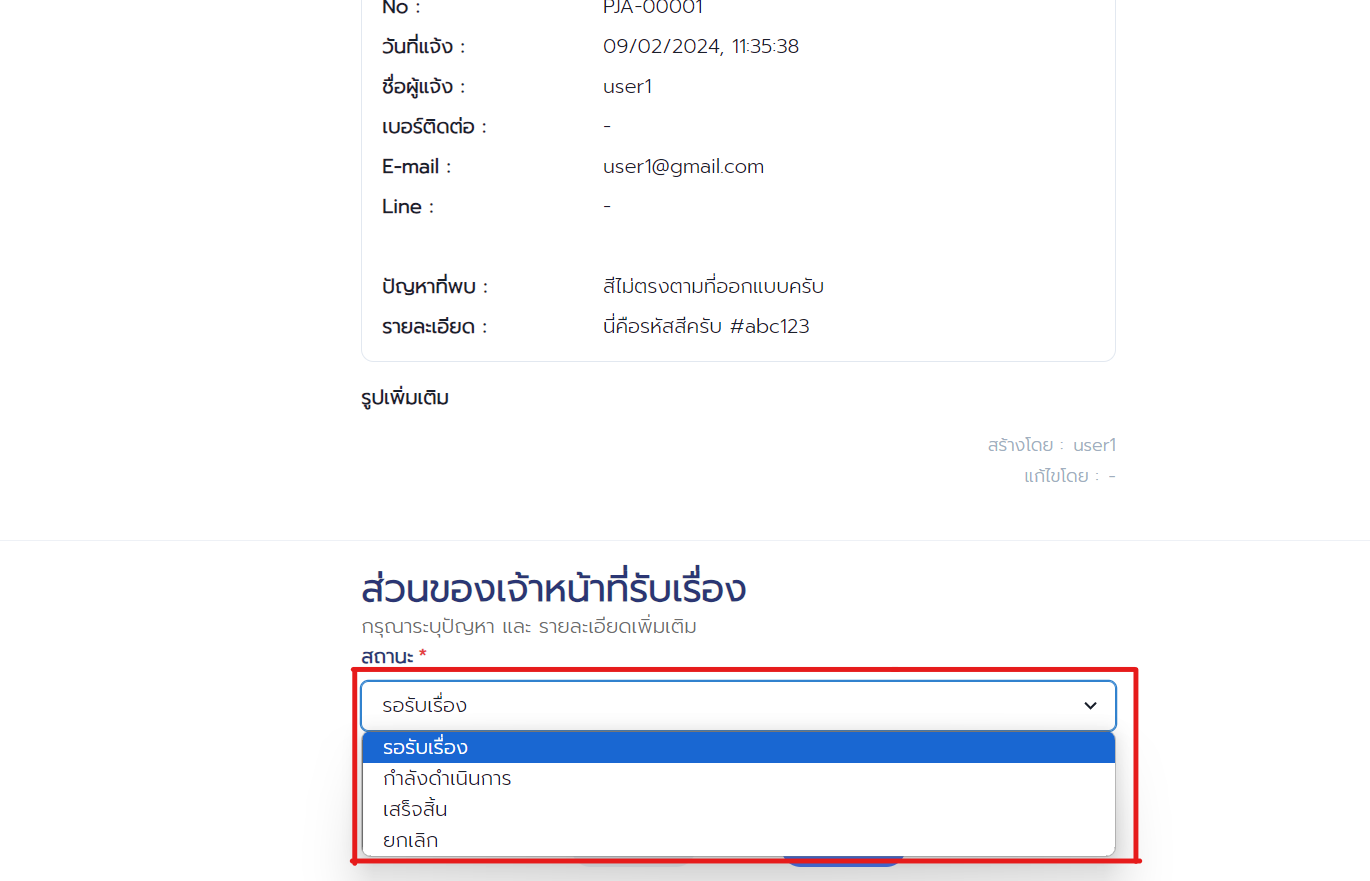


**รูปที่ ค.6** ภาพซีเควนซ์ไดอะแกรมการเพิ่มข้อมูลพนักงานบริษัท

ขั้นตอนผู้ตอบปัญหาเข้ามาตอบปัญหาเลือกบริษัทและโปรเจกต์ที่มีการแจ้งปัญหาเข้ามา แล้วกดแถวปัญหาที่ ต้องการตอบและเปลี่ยนสถานะของปัญหานั้น แล้วทำการตอบกลับและเปลี่ยนสถานะ

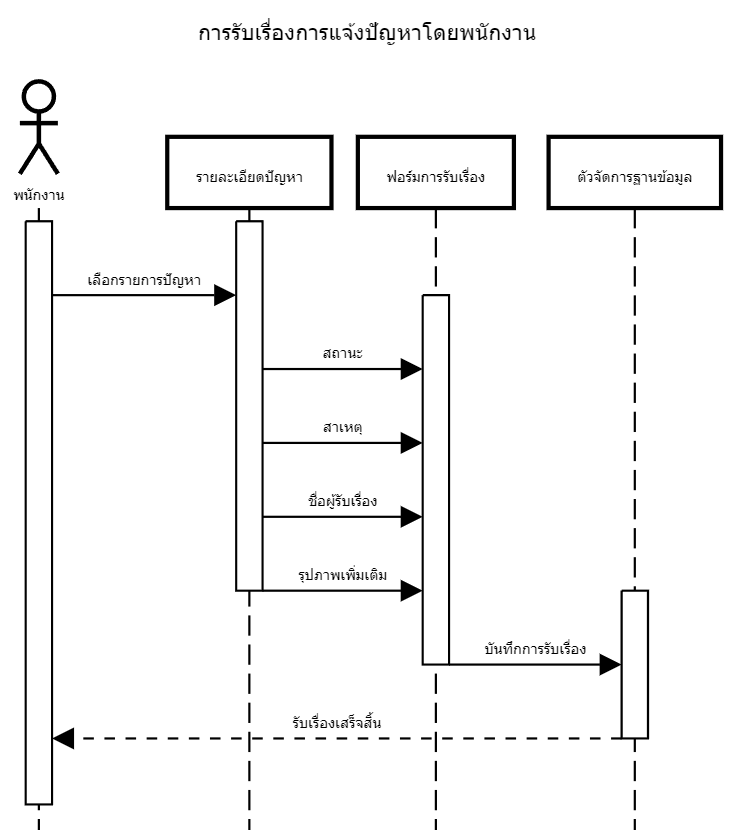
A blue and white rectangular box with text

Description automatically generated



A screenshot of a computer

Description automatically generated



**รูปที่ ค.7** ภาพซีเควนซ์ไดอะแกรมการตอบและการเปลี่ยนสถานะของปัญหาของพนักงาน

2. การทดสอบระบบแบบอัตโนมัติ

2.1 วิเคราะห์และออกแบบการทดสอบระบบ

ในขั้นตอนการทดสอบระบบแบบอัตโนมัติได้มีการออกแบบ Flow การทำงานของระบบ มีขั้นตอนดังรูปที่ ค.7ภาพ Flow การทำงานของระบบ โดยเริ่มจาก Admin เพิ่มบริษัทลูกค้าและเพิ่ม Project ของบริษัทลูกค้า จากนั้นเมื่อลูกค้า เมื่อพบปัญหาจากการใช้งานจะแจ้งปัญหาเข้ามาในระบบ Admin จะเข้ามารับเรื่องรายงานปัญหาและตอบปัญหาของลูกค้า หากเป็นปัญหาที่ Admin ไม่สามารถแก้ไขได้ จะส่งต่อไปยังพนักงานที่ดูแลในส่วนที่เกี่ยวข้องและทำการแก้ไข เมื่อ พนักงานแก้ไขเสร็จสิ้น พนักงานจะเข้ามาเปลี่ยนสถานะรายงานปัญหาและแจ้งสาเหตุและรายละเอียดการแก้ไขให้กับลูกค้า

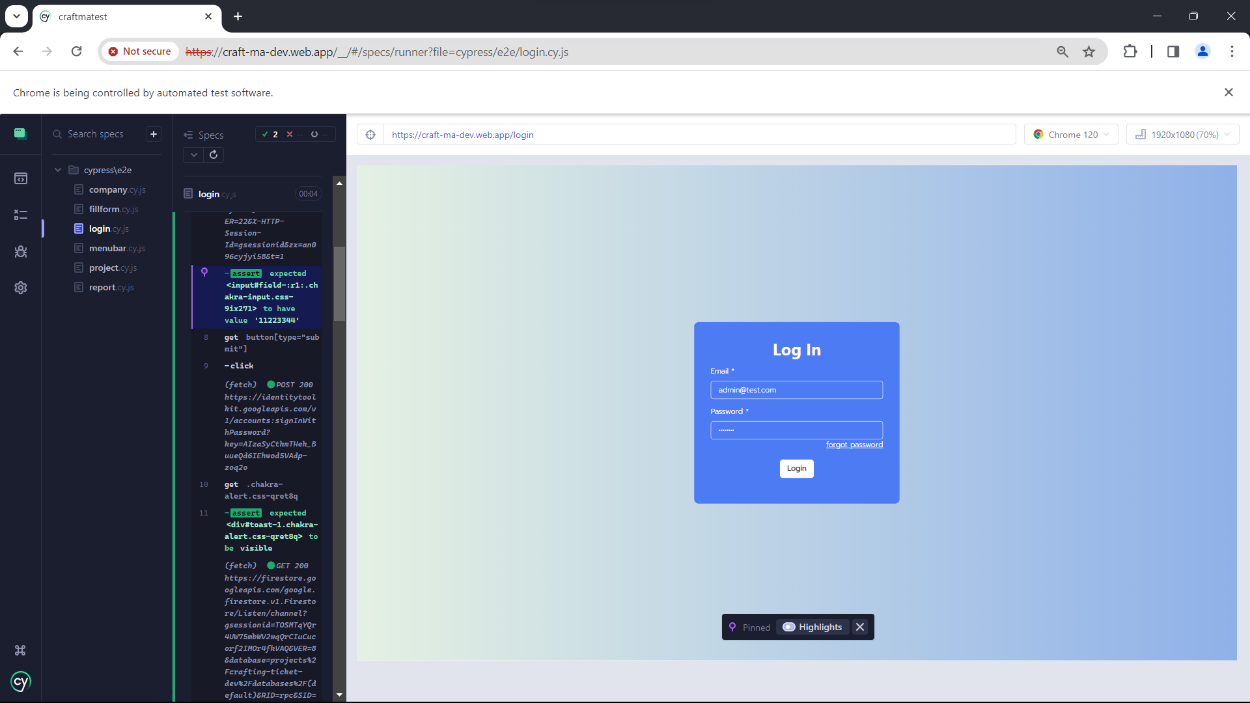
**A diagram of a diagram

Description automatically generated**

**รูปที่ ค.8** ภาพ Flow การทำงานของระบบ

2.2 การทดสอบระบบ

2.2.1 เข้าสู่ระบบ ในการทดสอบการเข้าสู่ระบบเริ่มต้นจากเข้าสู่หน้าเข้าสู่ระบบ จากนั้นกรอก Email และ Password มีการตรวจสอบการกรอกข้อมูลว่าถูกกรอกลงในช่องหรือไม่ จากนั้นกดปุ่ม Login ถ้า Email หรือ Password ถูกต้องจะตรวจสอบว่ามีข้อความ Success Message แสดงขึ้นมา และ ถ้า Email หรือ Password ไม่ถูกต้องจะตรวจสอบ ว่ามีข้อความ Error Message แสดงขึ้นมา



**รูปที่ ค.9** ภาพ Cypress ตอนกรอก Email และ Password

A screenshot of a computer

Description automatically generated

**รูปที่ ค.10** ภาพ Cypress ตอนตรวจสอบข้อความ Success Message แสดงขึ้นมา

A screenshot of a computer

Description automatically generated

**รูปที่ ค.11** ภาพ Cypress ตอนตรวจสอบข้อความ Error Message แสดงขึ้นมา

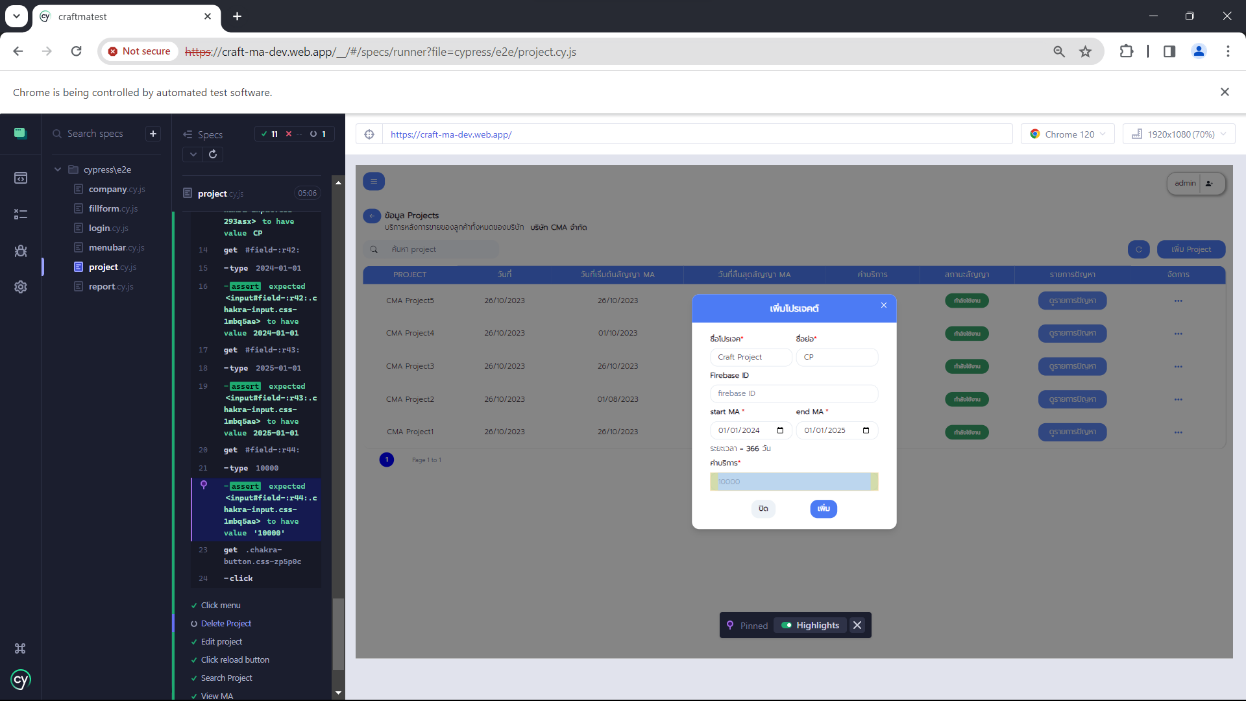
2.2.2 เพิ่มข้อมูลบริษัท การเพิ่มข้อมูลบริษัทจะเริ่มจากเข้าสู่หน้าข้อมูลลูกค้า และกดปุ่มเพิ่มข้อมูลบริษัท จะมี ฟอร์มเพิ่มข้อมูลบริษัทแสดงขึ้นมา จากนั้นกรอกข้อมูลบริษัทลุกค้าลงในฟอร์มเพิ่มข้อมูลบริษัท จะมีการตรวจสอบว่าข้อมูล ถูกกรอกลงในแต่ละช่องหรือไม่ และทำการกดบันทึก จะมีการตรวจสอบว่าข้อมูลที่บันทึกลงไปมีในตารางข้อมูลลูกค้าหรือ ไม่

A screenshot of a computer

Description automatically generated

**รูปที่ ค.12** ภาพ Cypress ตอนเพิ่มข้อมูลบริษัท

2.2.3 เพิ่มโปรเจกต์ เริ่มจากเข้าสู่หน้าข้อมูล Projects และกดปุ่ม เพิ่ม project ระบบจะแสดงฟอร์ม เพิ่มโปรเจกต์ขึ้นมา จากนั้นกรอกข้อมูลโปรเจกต์ลงในฟอร์มและตรวจสอบว่าข้อมูลสามารถกรอกลงในฟอร์มได้ หรือไม่ จากนั้นกดปุ่ม เพิ่ม และตรวจสอบว่าข้อมูลที่เพิ่ม มีในตารางข้องมูล Projects หรือไม่



**รูปที่ ค.13** ภาพ Cypress ตอนเพิ่มโปรเจกต์

2.2.4 แจ้งรายงานปัญหา เริ่มจากเข้าสู่หน้าแจ้งปัญหาการใช้งานระบบ และทำการกรอกข้อมูลรายงานปัญหาและ ตรวจสอบการกรอกข้อมูล จากนั้นกดบันทึกและตรวจสอบว่าข้อมูลรายงานปัญหาถูกบันทึกลงในตารางรายงานปัญหา หรือไม่

A screenshot of a computer

Description automatically generated

**รูปที่ ค.14** ภาพ Cypress ตอนแจ้งรายงานปัญหา

2.2.5 ตอบรายงานปัญหา เริ่มจากเข้าสู่หน้าแจ้งปัญหาการใช้งานระบบ และเลือกรายงานปัญหาที่ต้องการตอบ จากนั้นทำการแก้ไขสถานะและกรอกข้อมูลสาเหตุ รายละเอียดปัญหา และตรวจสอบการกรอกข้อมูล จากนั้นกดบันทึกและ ตรวจสอบว่าสถานะรายงานปัญหาถูกแก้ไขในตารางรายงานปัญหาหรือไม่

A screenshot of a computer

Description automatically generated

**รูปที่ ค.15** ภาพ Cypress ตอนตอบรายงานปัญหา

2.2.7 การทดสอบตั้งแต่เริ่มต้น Flow จนจบการทำงาน จะทดสอบตั้งแต่การเข้าสู่ระบบ Admin เพิ่มบริษัทลูกค้า และเพิ่ม Project ของบริษัทลูกค้า จากนั้นแจ้งปัญหาเข้ามาในระบบ และตอบรายงานปัญหาทั้ง 3 สถานะ

A screenshot of a computer

Description automatically generated

**รูปที่ ค.16** ภาพ Cypress ตอนทดสอบตั้งแต่เริ่มต้น Flow จนจบการทำงาน

**ผลการวิจัย:**

1. การพัฒนาระบบ

ได้ทำการทดสอบระบบในระยะเวลา 6 เดือน โดยกำหนดให้มีแอดมินสำหรับรับเรื่องรายงานปัญหา 1 คน พนักงาน 4 คน และบริษัทลูกค้า 2  บริษัท แต่ละบริษัทแจ้งปัญหาเข้ามาในละบบบริษัทละ 50 ปัญหา

**ตัวอย่างปัญหา:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ชื่อบริษัท** | **ลำดับ** | **หัวข้อปัญหา** | **รายละเอียดของปัญหา** | **พนักงาน** | **สาเหตุ** | **เสร็จสิ้น** | **กำลังดำเนินการ** | **รอรับเรื่อง** | **ยกเลิก** |
| Company A | 1 | อัพโหลดภาพไม่ได้ | หน้าฟอร์มรายงานปัญหาไม่สามารถอัพโหลดภาพได้ | Admin | ตัวอย่าง:  -ไฟล์ใหญ่เกินไป  -รับไฟล์ JPNG,PNG เท่านั้น | 7 | 2 | 2 | 2 |
| 2 | เข้าหน้ารายงานปัญหาไม่ได้ | เข้าหน้ารายงานปัญหาแล้วหน้าจอขาว | Dev | ตัวอย่าง:  -URL ผิด  -หน้าปัญหาไม่รีเฟรช  -อาจยังไม่มีการรายงานปัญหา | 8 | 2 | 2 | 0 |
| 3 | วันที่แจ้งรายงานปัญหาไม่ถูกต้อง | หลังจากแจ้งรายงานปัญหาแล้ว วันที่ที่แจ้งรายงานปัญหาแสดงไม่ถูกต้อง | Tester | ตัวอย่าง:  -ไวยกรณ์วันที่ไม่ตรงกัน | 10 | 2 | 2 | 0 |
| 4 | บันทึกรายงานปัญหาไม่ได้ | กรอกข้อมูลรายงานปัญหาแล้วกดปุ่มบันทึกแล้วไม่มีอะไรเกิดขึ้น รายงานปัญหาไม่แสดงในตารางรายงานปัญหา | HR | ตัวอย่าง:  -API มีการอัพเดต  -กรอกข้อมูลไม่ครบถ้วน  -เซิร์ฟเวอร์ไม่ตอบสนอง | 5 | 2 | 3 | 1 |
| 5 | รูปภาพไม่แสดง | ตอนเพิ่มรายงานปัญหา กรอกรายละเอียดและเพิ่มรูปภาพเพิ่มเติม แต่ที่หน้ารายละเอียดรายงานปัญหาไม่แสดงรูปภาพที่อัพโหลด | Dev | ตัวอย่าง:  -อินเทอร์เน็ตล่าช้า  -ไฟล์รูปภาพเสียหาย | 5 | 2 | 1 | 2 |
| Company B | 6 | เข้าสู่ระบบไม่ได้ | กรอกEmail Passwordแล้วกด login แต่ไม่สามารถเข้าสู่ระบบได้ | Dev | ตัวอย่าง:  -ยังไม่มีบัญชีในระบบ  -รหัสผ่านหรืออีเมลไม่่ถูกต้อง | 5 | 3 | 0 | 0 |
| 7 | แก้ไขรายละเอียดรายงานปัญหาไม่ได้ | หลังจากเพิ่มรายงานปัญหาแล้วต้องการจะแก้ไขในหน้ารายละเอียดรายงานปัญหา กรอกข้อมูลแล้วกดบันทึกแต่รายละเอียดที่แก้ไขไม่อัพเดท | Admin | ตัวอย่าง:  -ยังไม่ได้กดรีเฟรช  -API มีการอัพเดต | 7 | 1 | 0 | 0 |
| 8 | กดดูรายงานปัญหาไม่ได้ | กดที่รายงานปัญหาเพื่อดูรายละเอียดแล้วหน้ารายละเอียดรายงานปัญหาขึ้นจอขาว | Dev | ตัวอย่าง:  -ยังไม่มีรายการปัญหา  -URL ผิด | 5 | 2 | 3 | 1 |
| 9 | แก้ไขรูปภาพหน้ารายงานปัญหาไม่ได้ | ต้องการแก้ไขรูปภาพที่หน้ารายงานปัญหาแต่รูปไม่เปลี่ยนเป็นรูปที่แก้ไข | HR | ตัวอย่าง:  -สามารถอัพโหลดรูปได้ไม่เกิน 4 รุป  -รูปภาพที่มีขนาดมากเกินไป | 15 | 2 | 2 | 2 |
| 10 | ชื่อ Project ผิด | ที่หน้ารายงานปัญหาแสดงชื่อ Project ผิด | UX/UI | ตัวอย่าง:  -ใส่ตัวแปรผิด | 10 | 2 | 0 | 0 |

2. การทดสอบระบบแบบอัตโนมัติ

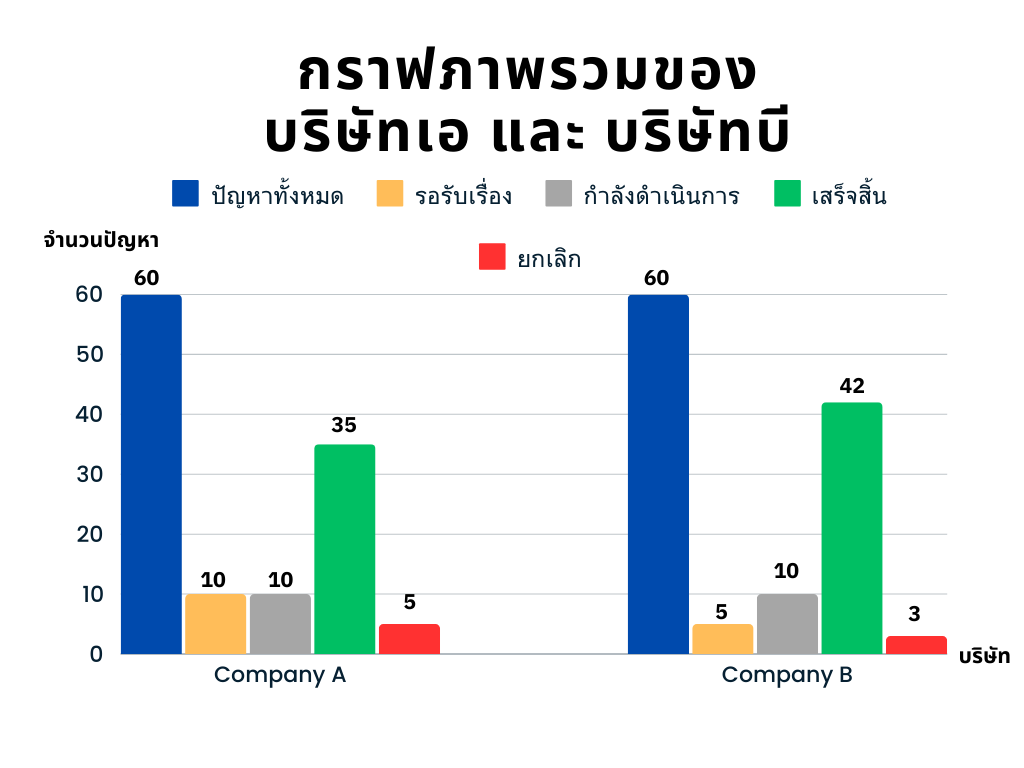
ตารางแสดงการทดสอบระบบแบบอัตโนมัติ โดยมีการทำการทดสอบ 5 ครั้ง กำหนดหัวข้อการทดสอบ 5 หัวข้อได้แก่ เข้าสู่ระบบ เพิ่มบริษัท เพิ่มโปรเจกต์ แจ้งรายงานปัญหา และการทดสอบการทำงานตั้งแต่เริ่มต้น flow จนจบการทำงาน แต่ละหัวข้อจะมีกรณีทดสอบและการใช้คำสั่ง Should ในการทดสอบ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ครั้งที่** | **หัวข้อการทดสอบ** | **กรณีทดสอบระบบ** | **จำนวนที่ใช้คำสั่ง Should ในการทดสอบ** | **จำนวนครั้งที่ผ่าน** | **จำนวนครั้งที่พบข้อผิดพลาด** |
| 1  วันที่  14  ส.ค.  2566 | เข้าสู่ระบบ | 2 กรณี | 6 ครั้ง | 6 | 0 |
| เพิ่มบริษัท | 9 กรณี | 25 ครั้ง | 17 | 8 |
| เพิ่มโปรเจกต์ | 4 กรณี | 7 ครั้ง | 5 | 2 |
| แจ้งรายงานปัญหา | 6 กรณี | 23 ครั้ง | 17 | 6 |
| การทดสอบการทำงานตั้งแต่เริ่มต้นจนจบการทำงาน  1. เข้าสู่ระบบ  2. เพิ่มบริษัท  3. เพิ่มโปรเจกต์  4. แจ้งรายงานปัญหา  5. เปลี่ยนสถานะรายงานปัญหา “กำลังดำเนินการ”  6. เปลี่ยนสถานะรายงานปัญหา “เสร็จสิ้น”  7. เปลี่ยนสถานะรายงานปัญหา “ยกเลิก” | 9 กรณี | 35 ครั้ง | 12 | 23 |
| 2  วันที่  20  ก.ย.  2566 | เข้าสู่ระบบ | 2 กรณี | 6 ครั้ง | 6 | 0 |
| เพิ่มบริษัท | 9 กรณี | 25 ครั้ง | 25 | 0 |
| เพิ่มโปรเจกต์ | 4 กรณี | 7 ครั้ง | 4 | 2 |
| แจ้งรายงานปัญหา | 6 กรณี | 23 ครั้ง | 15 | 8 |
| การทดสอบการทำงานตั้งแต่เริ่มต้นจนจบการทำงาน  1. เข้าสู่ระบบ  2. เพิ่มบริษัท  3. เพิ่มโปรเจกต์  4. แจ้งรายงานปัญหา  5. เปลี่ยนสถานะรายงานปัญหา “กำลังดำเนินการ”  6. เปลี่ยนสถานะรายงานปัญหา “เสร็จสิ้น”  7. เปลี่ยนสถานะรายงานปัญหา “ยกเลิก” | 9 กรณี | 35 ครั้ง | 24 | 11 |
| 3  วันที่  11  ต.ค.  2566 | เข้าสู่ระบบ | 2 กรณี | 6 ครั้ง | 6 | 0 |
| เพิ่มบริษัท | 9 กรณี | 25 ครั้ง | 25 | 0 |
| เพิ่มโปรเจกต์ | 4 กรณี | 7 ครั้ง | 7 | 0 |
| แจ้งรายงานปัญหา | 6 กรณี | 23 ครั้ง | 21 | 3 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ครั้งที่ | **หัวข้อการทดสอบ** | **กรณีทดสอบระบบ** | **จำนวนที่ใช้คำสั่ง Should ในการทดสอบ** | **จำนวนครั้งที่ผ่าน** | **จำนวนครั้งที่พบข้อผิดพลาด** |
|  | การทดสอบการทำงานตั้งแต่เริ่มต้นจนจบการทำงาน  1. เข้าสู่ระบบ  2. เพิ่มบริษัท  3. เพิ่มโปรเจกต์  4. แจ้งรายงานปัญหา  5. เปลี่ยนสถานะรายงานปัญหา “กำลังดำเนินการ”  6. เปลี่ยนสถานะรายงานปัญหา “เสร็จสิ้น”  7. เปลี่ยนสถานะรายงานปัญหา “ยกเลิก” | 9 กรณี | 35 ครั้ง | 27 | 8 |
| 4  วันที่  19  ต.ค.  2566 | เข้าสู่ระบบ | 2 กรณี | 6 ครั้ง | 6 | 0 |
| เพิ่มบริษัท | 9 กรณี | 25 ครั้ง | 25 | 0 |
| เพิ่มโปรเจกต์ | 4 กรณี | 7 ครั้ง | 6 | 1 |
| แจ้งรายงานปัญหา | 6 กรณี | 23 ครั้ง | 23 | 0 |
| การทดสอบการทำงานตั้งแต่เริ่มต้นจนจบการทำงาน  1. เข้าสู่ระบบ  2. เพิ่มบริษัท  3. เพิ่มโปรเจกต์  4. แจ้งรายงานปัญหา  5. เปลี่ยนสถานะรายงานปัญหา “กำลังดำเนินการ”  6. เปลี่ยนสถานะรายงานปัญหา “เสร็จสิ้น”  7. เปลี่ยนสถานะรายงานปัญหา “ยกเลิก” | 9 กรณี | 35 ครั้ง | 32 | 3 |
| 5  วันที่  22  ต.ค.  2566 | เข้าสู่ระบบ | 2 กรณี | 6 ครั้ง | 6 | 0 |
| เพิ่มบริษัท | 9 กรณี | 25 ครั้ง | 25 | 0 |
| เพิ่มโปรเจกต์ | 4 กรณี | 7 ครั้ง | 5 | 0 |
| แจ้งรายงานปัญหา | 6 กรณี | 23 ครั้ง | 23 | 0 |
| การทดสอบการทำงานตั้งแต่เริ่มต้นจนจบการทำงาน  1. เข้าสู่ระบบ  2. เพิ่มบริษัท  3. เพิ่มโปรเจกต์  4. แจ้งรายงานปัญหา  5. เปลี่ยนสถานะรายงานปัญหา “กำลังดำเนินการ”  6. เปลี่ยนสถานะรายงานปัญหา “เสร็จสิ้น”  7. เปลี่ยนสถานะรายงานปัญหา “ยกเลิก” | 9 กรณี | 35 ครั้ง | 35 | 0 |

**วิจารณ์และสรุปผล:**

1. การพัฒนาระบบ



**กราฟ** : ภาพรวมการรายงานปัญหา

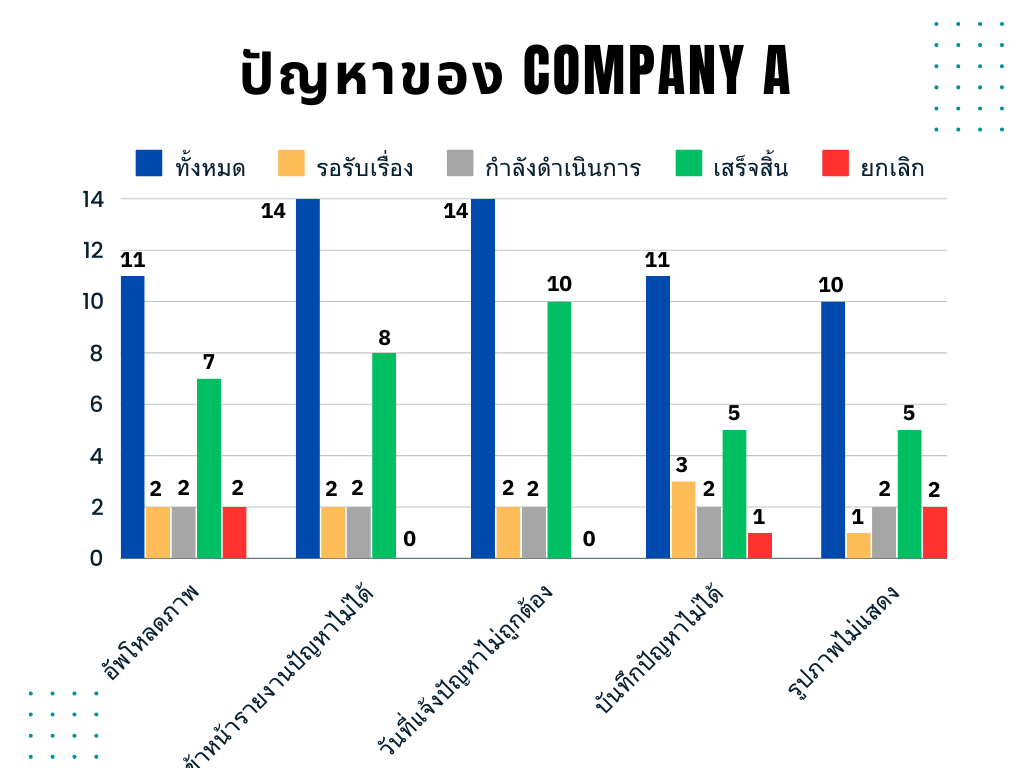
จากกราฟ Company A มีปัญหาที่อยู่ในสถานะเสร็จสิ้น 35 ปัญหา รอดำเนินการและกำลังดำเนินการอย่างละ 10 ปัญหา สถานะปัญหาที่ถูกยกเลิกแล้ว 5 ปัญหา และ Company B ปัญหาที่อยู่ในสถานะเสร็จสิ้น 42 ปัญหา สถานะรอดำเนินการ 10 ปัญหา สถานะรอรับเรื่อง 5 และ สถานะยกเลิก 3 ปัญหา สรุปแล้วแต่ละบริษัทมีการแจ้งปัญหาทั้งหมด 60 ปัญหา

A close-up of a chart

Description automatically generated

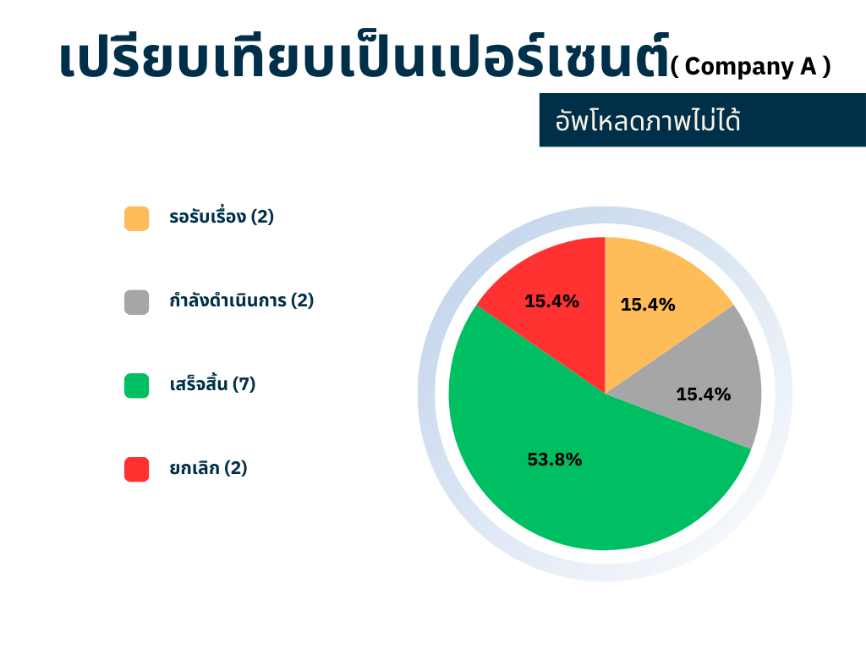
**กราฟ** : เปรียบเทียบเปอร์เซนต์ของ 2 บริษัท

จากกราฟเทียบเปอร์เซนต์ของ 2 บริษัท บริษัทบีมีการแก้ไขปัญหาเสร็จสิ้นมากกว่าบริษัทเอและปัญหาที่อยู่ในสถานะรอรับเรื่องและสถานะรอดำเนินการก็น้อยกว่าบริษัทเอ



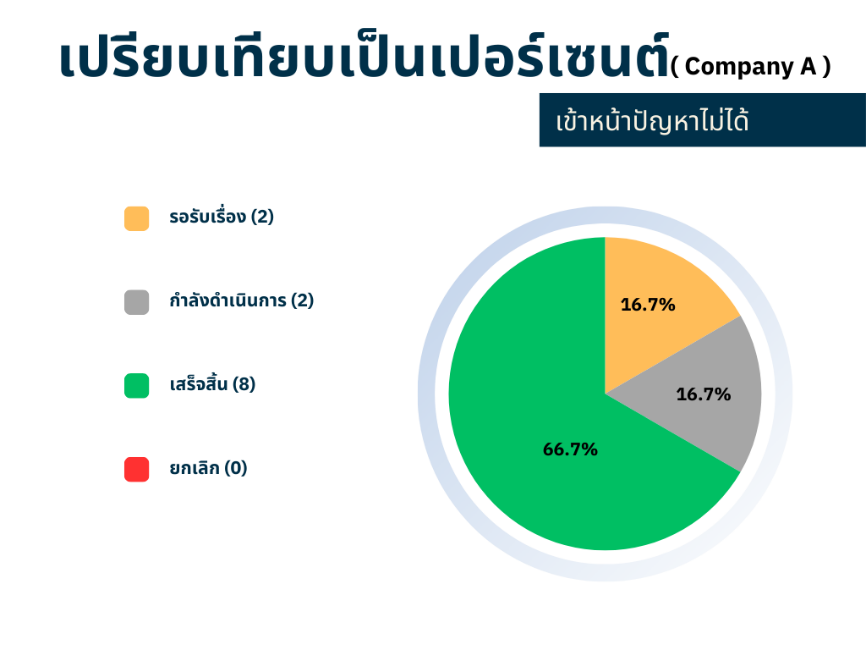
**กราฟ** : แสดงสถานะปัญหาทั้งหมดของแต่ละปัญหาของ Company A

จากกราฟปัญหา Company A ปัญหาเข้าหน้ารายงานปัญหาไม่ได้ กับ ปัญหาวันที่แจ้งปัญหาไม่ถูกต้อง มีการแจ้งเข้าสู่ระบบมากที่สุดคือ 14 ปัญหา ส่วน ปัญหารูปภาพไม่แสดง มีการแจ้งน้อยที่สุด คือ 10 ปัญหา ส่วนปัญหาที่มีการแก้ไขมากที่สุดคือ ปัญหาวันที่แจ้งปัญหาไม่ถูกต้อง และ ปัญหาบันทักปัญหาไม่ได้ กับ ปัญหารูปภาพไม่แสดง มีการแก้ไขน้อยที่สุดคือ 5 ปัญหา



**กราฟ** : แสดงสถานะปัญหาอัพโหลดภาพไม่ได้

จากกราฟเมื่อนำแต่ละสถานะของปัญหาอัพโหลดภาพไม่ได้ ปัญหาที่มีสถานะรอรับเรื่อง 2 ปัญหาคิดเป็น 18.2 % ปัญหาที่มีสถานะกำลังดำเนินการมี 2 ปัญหา คิดเป็น 18.2 % ปัญหาที่มีสถานะเสร็จสิ้นมี 7 ปัญหา คิดเป็น 63.6 %



**กราฟ** : แสดงสถานะปัญหาเข้าหน้าปัญหาไม่ได้

จากกราฟ แสดงสถานะปัญหาเข้าหน้าปัญหาไม่ได้ ปัญหาที่มีสถานะรอรับเรื่อง 2 ปัญหาคิดเป็น 14.3 % ปัญหาที่มีสถานะกำลังดำเนินการ 2 ปัญหา คิดเป็น 14.3 % ปัญหาที่มีสถานะเสร็จสิ้น 7 ปัญหา คิดเป็น 63.6 % และ ปัญหาที่มีสถานะยกเลิก 2 ปัญหา คิดเป็น 14.3 %

A pie chart with different colored circles

Description automatically generated

**กราฟ** : แสดงสถานะปัญหาวันที่แจ้งปัญหาไม่ถูกต้อง

จากกราฟแสดงสถานะปัญหาวันที่แจ้งปัญหาไม่ถูกต้อง ปัญหาที่มีสถานะรอรับเรื่อง 2 ปัญหาคิดเป็น 14.3 % ปัญหาที่มีสถานะกำลังดำเนินการ 2 ปัญหา คิดเป็น 14.3 % และ ปัญหาที่มีสถานะเสร็จสิ้น 10 ปัญหา คิดเป็น 71.4 %

A pie chart with numbers and a few words

Description automatically generated with medium confidence

**กราฟ** : แสดงสถานะปัญหาบันทึกปัญหาไม่ได้

จากกราฟ แสดงสถานะปัญหาเข้าหน้าปัญหาไม่ได้ ปัญหาที่มีสถานะรอรับเรื่อง 3 ปัญหาคิดเป็น21.3 % ปัญหาที่มีสถานะกำลังดำเนินการ 2 ปัญหา คิดเป็น 18.2 % ปัญหาที่มีสถานะเสร็จสิ้น 5 ปัญหา คิดเป็น 45.5 % และ ปัญหาที่มีสถานะยกเลิก 1 ปัญหา คิดเป็น 9.1 %

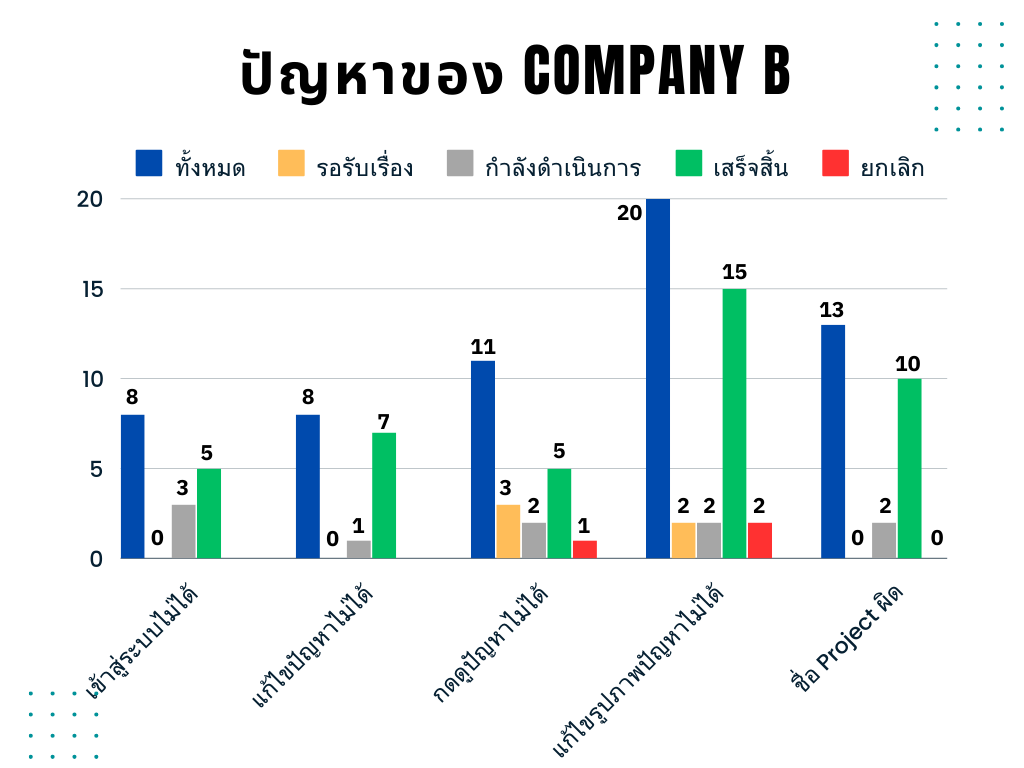
A pie chart with different colored circles

Description automatically generated

**กราฟ :** แสดงสถานะปัญหารูปภาพไม่แสดง

จากกราฟแสดงสถานะปัญหารูปภาพไม่แสดง ปัญหาที่มีสถานะรอรับเรื่อง 1 ปัญหาคิดเป็น 10 % ปัญหาที่มีสถานะกำลังดำเนินการ 2 ปัญหา คิดเป็น 20 % ปัญหาที่มีสถานะเสร็จสิ้น 5 ปัญหา คิดเป็น 50 % และ ปัญหาที่มีสถานะยกเลิก 2 ปัญหา

คิดเป็น 20 %



**กราฟ** : แสดงสถานะปัญหาทั้งหมดของแต่ละปัญหาของ Company B

จากกราฟปัญหา Conpany B ปัญหาแก้ไไจรูกภาพปัญหาไม่ได้ มีการแจ้งเข้าสู่ระบบมากที่สุดคือ 20 ปัญหา ส่วน ปัญหาเข้าสู่ระบบไม่ได้ กับ แก้ไขปัญหาไม่ได้ มีการแจ้งน้อยที่สุด คือ 8 ปัญหา ส่วนปัญหาที่มีการแก้ไขมากที่สุดคือ ปัญหาแก้ไขรูปภาพปัญหาไม่ได้คือ 15 ปัญหา และปัญหาที่มีการแก้ไขน้อยที่สุดคือปัญหาเข้าสู่ระบบไม่ได้ คือ 5 ปัญหา

A screenshot of a graph

Description automatically generated

**กราฟ :** แสดงสถานะปัญหาเข้าสู่ระบบไม่ได้

จากกราฟแสดงสถานะปัญหาเข้าสู่ระบบไม่ได้ ปัญหาที่มีสถานะกำลังดำเนินการ 3 ปัญหา คิดเป็น 37.5 % ปัญหาที่มีสถานะเสร็จสิ้น 5 ปัญหา คิดเป็น 62.5 %

A green pie chart with a green circle

Description automatically generated

**กราฟ :** แสดงสถานะปัญหาแก้ไขรายละเอียดปัญหาไม่ได้

จากกราฟแสดงสถานะปัญหาแก้ไขรายละเอียดปัญหาไม่ได้ ปัญหาที่มีสถานะกำลังดำเนินการ 1 ปัญหา  คิดเป็น 12.5 % ปัญหาที่มีสถานะเสร็จสิ้น 7 ปัญหา คิดเป็น 87.5 %

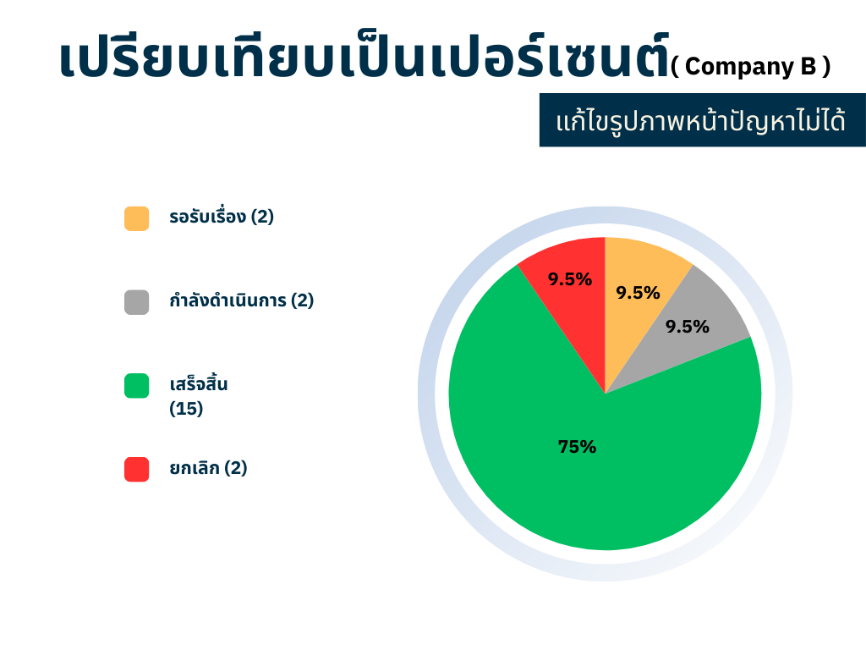
A pie chart with numbers and a few words

Description automatically generated with medium confidence

**กราฟ :** แสดงสถานะปัญหากดดูปัญหาไม่ได้

จากกราฟแสดงสถานะปัญหากดดูปัญหาไม่ได้ ปัญหาที่มีสถานะรอรับเรื่อง 3 ปัญหาคิดเป็น 27.3 % ปัญหาที่มีสถานะกำลังดำเนินการ 2 ปัญหา คิดเป็น 18.2 % ปัญหาที่มีสถานะเสร็จสิ้น 5 ปัญหา คิดเป็น 45.5 % และ ปัญหาที่มีสถานะยกเลิก 1 ปัญหา

คิดเป็น 9.1 %



**กราฟ :** แสดงสถานะปัญหาแก้ไขรูปภาพปัญหาไม่ได้

จากกราฟแสดงสถานะปัญหาแก้ไขรูปภาพปัญหาไม่ได้ ปัญหาที่มีสถานะรอรับเรื่อง 2 ปัญหาคิดเป็น 10 % ปัญหาที่มีสถานะกำลังดำเนินการ 2 ปัญหา คิดเป็น 10 % ปัญหาที่มีสถานะเสร็จสิ้น 15 ปัญหา คิดเป็น 75 % และ ปัญหาที่มีสถานะยกเลิก 1 ปัญหา

คิดเป็น 5 %



**กราฟ :** แสดงสถานะปัญหาชื่อ Project ผิด

จากกราฟแสดงสถานะปัญหาชื่อ Project ผิด ปัญหาที่มีสถานะกำลังดำเนินการ 2 ปัญหา คิดเป็น 15.4 % ปัญหาที่มีสถานะเสร็จสิ้น 10 ปัญหา คิดเป็น 76.9 % และ ปัญหาที่มีสถานะยกเลิก 1 ปัญหา คิดเป็น 7.7 %

2. การทดสอบระบบแบบอัตโนมัติ

จากผลการทดสอบระบบแบบอัตโนมัติ โดยกำหนดหัวข้อการทดสอบ 5 หัวข้อ ได้แก่ เข้าสู่ระบบ เพิ่มบริษัท เพิ่มโปรเจกต์ แจ้งรายงานปัญหา และการทดสอบการทำงานตั้งแต่เริ่มต้น flow จนจบการทำงาน

ครั้งที่ 1 ประกอบด้วยหัวข้อการทดสอบระบบได้แก่ เข้าสู่ระบบ มีการใช้ Should ในการทดสอบทั้งหมด 6 ครั้ง ผ่าน 6 ครั้ง เพิ่มบริษัท มีการใช้ Should ในการทดสอบทั้งหมด 25 ครั้ง ผ่าน 17 ครั้ง ไม่ผ่าน 8 ครั้ง เพิ่มโปรเจกต์ มีการใช้ Should ในการทดสอบทั้งหมด 7 ครั้ง ผ่าน 5 ครั้ง ไม่ผ่าน 2 ครั้ง แจ้งรายงานปัญหา มีการใช้ Should ในการทดสอบทั้งหมด 23 ครั้ง ผ่าน 17 ครั้ง ไม่ผ่าน 6 ครั้ง และการทดสอบการทำงานตั้งแต่เริ่มต้น flow จนจบการทำงาน มีการใช้ Should ในการทดสอบทั้งหมด 35 ครั้ง ผ่าน 12 ครั้ง ไม่ผ่าน 23 ครั้ง

ครั้งที่ 2 ประกอบด้วยหัวข้อการทดสอบระบบได้แก่ เข้าสู่ระบบ มีการใช้ Should ในการทดสอบทั้งหมด 6 ครั้ง ผ่าน 6 ครั้ง เพิ่มบริษัท มีการใช้ Should ในการทดสอบทั้งหมด 25 ครั้ง ผ่าน 25 ครั้ง เพิ่มโปรเจกต์ มีการใช้ Should ในการทดสอบทั้งหมด 7 ครั้ง ผ่าน 4 ครั้ง ไม่ผ่าน 3 ครั้ง แจ้งรายงานปัญหา มีการใช้ Should ในการทดสอบทั้งหมด 23 ครั้ง ผ่าน 15 ครั้ง ไม่ผ่าน 8 ครั้ง และการทดสอบการทำงานตั้งแต่เริ่มต้น flow จนจบการทำงาน มีการใช้ Should ในการทดสอบทั้งหมด 35 ครั้ง ผ่าน 24 ครั้ง ไม่ผ่าน 11 ครั้ง

ครั้งที่ 3 ประกอบด้วยหัวข้อการทดสอบระบบได้แก่ เข้าสู่ระบบ มีการใช้ Should ในการทดสอบทั้งหมด 6 ครั้ง ผ่าน 6 ครั้ง เพิ่มบริษัท มีการใช้ Should ในการทดสอบทั้งหมด 25 ครั้ง ผ่าน 25 ครั้ง เพิ่มโปรเจกต์ มีการใช้ Should ในการทดสอบทั้งหมด 7 ครั้ง ผ่าน 7 ครั้ง แจ้งรายงานปัญหา มีการใช้ Should ในการทดสอบทั้งหมด 23 ครั้ง ผ่าน 21 ครั้ง ไม่ผ่าน 2 ครั้ง และการทดสอบการทำงานตั้งแต่เริ่มต้น flow จนจบการทำงาน มีการใช้ Should ในการทดสอบทั้งหมด 35 ครั้ง ผ่าน 27 ครั้ง ไม่ผ่าน 8 ครั้ง

ครั้งที่ 4 ประกอบด้วยหัวข้อการทดสอบระบบได้แก่ เข้าสู่ระบบ มีการใช้ Should ในการทดสอบทั้งหมด 6 ครั้ง ผ่าน 6 ครั้ง เพิ่มบริษัท มีการใช้ Should ในการทดสอบทั้งหมด 25 ครั้ง ผ่าน 25 ครั้ง เพิ่มโปรเจกต์ มีการใช้ Should ในการทดสอบทั้งหมด 7 ครั้ง ผ่าน 6 ครั้ง ไม่ผ่าน 1 ครั้ง แจ้งรายงานปัญหา มีการใช้ Should ในการทดสอบทั้งหมด 23 ครั้ง ผ่าน 23 ครั้ง และการทดสอบการทำงานตั้งแต่เริ่มต้น flow จนจบการทำงาน มีการใช้ Should ในการทดสอบทั้งหมด 35 ครั้ง ผ่าน 32 ครั้ง ไม่ผ่าน 3 ครั้ง

ครั้งที่ 5 ประกอบด้วยหัวข้อการทดสอบระบบได้แก่ เข้าสู่ระบบ มีการใช้ Should ในการทดสอบทั้งหมด 6 ครั้ง ผ่าน 6 ครั้ง เพิ่มบริษัท มีการใช้ Should ในการทดสอบทั้งหมด 25 ครั้ง ผ่าน 25 ครั้ง เพิ่มโปรเจกต์ มีการใช้ Should ในการทดสอบทั้งหมด 7 ครั้ง ผ่าน 7 ครั้ง แจ้งรายงานปัญหา มีการใช้ Should ในการทดสอบทั้งหมด 23 ครั้ง ผ่าน 23 ครั้ง และการทดสอบการทำงานตั้งแต่เริ่มต้น flow จนจบการทำงาน มีการใช้ Should ในการทดสอบทั้งหมด 35 ครั้ง ผ่าน 35 ครั้ง



จากกราฟการทดสอบระบบแบบอัตโนมัติ จะเห็นได้ว่าการทดสอบระบบแบบอัตโนมัติทั้งหมด 5 ครั้ง ในครั้งที่ 1 จนถึง ครั้งที่ 4 พบว่ามีการทดสอบแล้วพบปัญหาของระบบน้อยลงตามลำดับ และในครั้งที่ 5 สามารถทดสอบแล้วไม่พบปัญหาของระบบ

ดังนั้นจากการทำการทดสอบระบบแบบอัตโนมัติ สามารถทดสอบระบบเพื่อมาทดสอบระบบบริการดูแลและบำรุงรักษาระบบโปรแกรมได้ตามหัวข้อและวัตถุประสงค์ที่กำหนด

**กิตติกรรมประกาศ:**

โครงงานคอมพิวเตอร์ “ระบบบริการดูแลและบำรุงรักษาระบบโปรแกรม” สำเร็จลุล่วงได้ด้วยดีเนื่องจากได้รับความกรุณาอย่างสูงจาก นายอภิรัฐ เลิศพฤกษ์ ประธานริษัทคราฟติ้งแล็บ จำกัด ที่ให้คำแนะนำเกี่ยวกับการพัฒนา และให้ความรู้เกี่ยวกับการพัฒนาระบบ ทำให้การทำงานวิจัยครั้งนี้มีความสมบูรณ์ อีกทั้งสำเร็จลุล่วงได้ด้วยดี และขอขอบคุณ อาจารย์ ดร.สมนึก สินธุปวน ประธานกรรมการ ผู้ช่วยศาตราจารย์ก่องกาญจน์ ดุลยไชย และ อาจารย์อรรถวิท ชังคมานนท์ กรรมการ ที่ให้ความรู้คำปรึกษาตลอดจนชี้แนะแนวทางอันเป็นประโยชน์ต่องานวิจัย การปรับปรุงแก้ไขข้อบกพร่องต่างๆ ในครั้งนี้จนทำให้งานวิจัยสมบูรณ์ ด้วยความเอาใจใส่อย่างดียิ่ง ผู้จัดทำขอขอบพระคุณอย่างสูงไว้ ณ ที่นี้ด้วย

**เอกสารอ้างอิง:**

กฤษณะ พิริยะกิจไพบูลย์, การพัฒนาเครื่องมือสร้างข้อมูลทดสอบเพื่อสนับสนุนการทดสอบซอฟต์แวร์จากสคีมาของฐานข้อมูล (วิทยานิพนธ์ปริญญามหาบัณฑิต, กรุงเทพฯ: จุฬาลงกรณ์มหาวิทยาลัย, 2547).

พนารัตน์ เชี่ยววิทย์, การพัฒนาเครื่องมือการทดสอบซอฟต์แวร์ในระดับหน่วยสำหรับกระบวนการพัฒนาเชิงวัตถุ (วิทยานิพนธ์ปริญญา มหาบัณฑิต, กรุงเทพฯ: มหาวิทยาลัยมหิดล, 2545).

สันต์ทศน์ สุริยันต์, การพัฒนาเครื่องมือสำหรับจัดการกระบวนการวางแผนการทดสอบซอฟต์แวร์ (วิทยานิพนธ์ปริญญามหาบัณฑิต, กรุงเทพฯ: จุฬาลงกรณ์มหาวิทยาลัย, 2543).

ธนพล สิชฌนุกฤษฎ์, การพัฒนาระบบจัดการกรณีทดสอบซอฟต์แวร์ (วิทยานิพนธ์ปริญญามหาบัณฑิต, กรุงเทพฯ: จุฬาลงกรณ์ มหาวิทยาลัย, 2544).

รู้จัก Cypress: Web Test Framework ที่จะทำให้คุณลืม Selenium ไปได้เลย [ระบบออนไลน์].

แหล่งที่มา https://medium.com/cypress-io-thailand/ (วันที่สืบค้น 31 สิงหาคม 2566).

เริ่มต้นเขียน Automated Test ด้วย Cypress.io แบบลงมือทำ[ระบบออนไลน์].

แหล่งที่มา https://medium.com/nellika/ (วันที่สืบค้น 31 สิงหาคม 2566).

Chakra UI เป็นตัวจัดการ UI ที่ใช้ร่วมกับ React

แหล่งที่มา https://chakra-ui.com/ (วันที่สืบค้น 31 สิงหาคม 2566).

Firebase เป็นตัวฐานข้อมูล

แหล่งที่มา https://firebase.google.com/ (วันที่สืบค้น 31 สิงหาคม 2566).

[QA] Cypress คืออะไร | ลองใช้ Cypress ฉบับ Beginner [ระบบออนไลน์].

แหล่งที่มา https://medium.com/@mintminttttt/qa-cypress-for-beginner-4c6a5a623292 (วันที่สืบค้น 31 สิงหาคม 2566).

React เป็น JavaScript library ที่ใช้สร้าง user interface

แหล่งที่มา https://react.dev/ (วันที่สืบค้น 31 สิงหาคม 2566).

React Icon เป็นแหล่งรวม Icon ต่างๆของ React

แหล่งที่มา https://react-icons.github.io/react-icons/ (วันที่สืบค้น 31 สิงหาคม 2566).

Visual Studio Code ใช้ในการพัฒนาระบบ

แหล่งที่มา https://code.visualstudio.com/ (วันที่สืบค้น 31 สิงหาคม 2566)

Chai Phonbopit.(2565). ทำระบบ Login/Register ด้วย React Router และ Firebase Auth (JavaScript). สืบค้น 20 กรกฎาคม 2566, จาก https://devahoy.com/workshop-react-firebase-authentication-javascript

Prudchaya Palee. (2563). สร้าง REST API ด้วย Firebase cloud functions และ Express.สืบค้น 5 สิงหาคม 2566, จาก https://prudchayapalee.medium.com/cloud-firestore-firebase-functions-express-js-606ac55a7287

Chai Phonbopit. (2562). Deploy เว็บไซต์ React ฟรีๆ ด้วย Firebase Hosting. สืบค้น 10 ตุลาคม 2566, จาก https://blog.devahoy.com/blog/2019/01/how-to-deploy-react-with-firebase-hosting