



รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบยืมคืนอุปกรณ์ผ่านตู้ล็อกเกอร์อัจฉริยะ
Smart Locker

นางสาวสุชาวลี โก

สาขาวิชาวิศวกรรมสารสนเทศ
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561



รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบยืมคืนอุปกรณ์ผ่านตู้ล็อกเกอร์อัจฉริยะ

Smart Locker

นางสาวสุชาวลี โก

สาขาวิชาวิศวกรรมสารสนเทศ

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจศึกษา ระบบยืมคืนอุปกรณ์ผ่านตู้ล็อกเกอร์อัจฉริยะ

ชื่อ-สกุล นักศึกษา นางสาวสุชาวลี โก

คณะ วิศวกรรมศาสตร์ ภาควิชา วิศวกรรมคอมพิวเตอร์ สาขา วิศวกรรมสารสนเทศ

ชื่อ-สกุล อาจารย์นิเทศ รศ.ดร.ชวลิต เบญจางคประเสริฐ

ชื่อ-สกุล ผู้นิเทศงาน คุณอานภาพ พิศลยบุตร

สถานประกอบการ บริษัท เอ็กซอนโมบิล จำกัด

บทคัดย่อ

โปรเจกต์ Smart Locker นั้นจัดทำขึ้นมาเพื่อลดความยุ่งยากซับซ้อน เพิ่มช่องทางและความสะดวกสบายในการยืมคืนอุปกรณ์สำนักงานที่จำเป็นต่อการทำงาน ผู้ใช้งานสามารถทำการยืมและคืนอุปกรณ์ได้ด้วยตนเอง โดยไม่ต้องเสียเวลาติดต่อกับเจ้าหน้าที่ อีกทั้งผู้ใช้งานสามารถตรวจสอบสถานะของอุปกรณ์ที่สามารถยืม และสถานะการยืมของตนเองผ่านทางเว็บไซต์ได้ ต่างจากการยืมคืนอุปกรณ์แบบเก่าที่ผู้ใช้งานจะต้องติดต่อกับเจ้าหน้าที่ เพื่อทำการสอบถามถึงอุปกรณ์ชิ้นนั้นก่อน ถึงจะทำการยืมอุปกรณ์ได้ หรือหากเกิดกรณีที่ไม่สามารถยืมได้ก็จะมีของให้ยืม ก็จะส่งผลให้เกิดการเสียเวลาโดยไม่จำเป็นขึ้น และในทางกลับกัน ด้านฝั่งผู้ดูแลก็ต้องมีเจ้าหน้าที่หลายคนคอยรับผิดชอบและดูแลเรื่องการยืมคืนอุปกรณ์ เพราะพนักงานทุกคนย่อมมีงานประจำในส่วนของตนเองอยู่แล้ว จึงไม่มีเจ้าหน้าที่คนไหนสามารถดูแลระบบได้ตลอดเวลา ทำให้เสียทรัพยากรมนุษย์โดยใช่เหตุ จึงเป็นที่มาของโปรเจกต์ Smart Locker ที่พัฒนาขึ้นมาเพื่อที่จะอำนวยความสะดวก ลดระยะเวลาที่ต้องเสียไปจากการยืมคืนอุปกรณ์แบบเก่า ลดทรัพยากรมนุษย์ที่ต้องเสียไปโดยไม่จำเป็น และเพิ่มประสิทธิภาพของระบบในการทำงานให้องค์กรอย่างสูงสุด

Cooperative Title: Smart Locker

Student Intern Name: Suchawalee Goh

Faculty: Engineering **Department:** Computer Engineering **Major:** Information Engineering

Advisor name: Assoc.Prof.Dr. Chawalit Benjangkaprasert

Mentor Name: Anuphab Pisalyaput

Company: ExxonMobil Limited

ABSTRACT

Smart Locker is made on purpose to reduce complexity and enhance comfortability about office supplies loaning system which the user can loan the item without contacting the staff. Also, having a channel to monitor equipment availability and your loaning status via the website which differs from the old loaning system that the user must ask the staff about the equipment to loan first but it may have a chance that item is not available in that time which is an unnecessary waste of time. On the other hand, it needs a lot of human resources to manage and responsible for equipment loaning system because all employees always have their routine work and make no one can handle the system all the time. So, developing the Smart Locker system can accommodate users, reduce the time use from the old system, prevent losing on human resources and increase the system efficiency for the organization.

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ข้าพเจ้าได้มีโอกาสร่วมทำงานที่บริษัท เอ็กซอนโมบิล จำกัด ในแผนก Web Content & Services ภายใต้โครงการฝึกงานภาคฤดูร้อน และโครงการสหกิจศึกษา ประจำปีการศึกษา 2561 ตั้งแต่วันที่ 1 มิถุนายน 2561 จนถึงวันที่ 21 ธันวาคม 2561 รวมแล้วเป็นระยะเวลา 140 วัน ซึ่งการได้ร่วมทำงานกับองค์กรขนาดใหญ่ ทำให้ข้าพเจ้าได้รับประสบการณ์ต่าง ๆ ในการทำงานมากมาย ทั้งด้านการต่อยอดจากที่เคยได้ศึกษามา และนอกเหนือจากที่ได้ศึกษามามากหลายเรื่อง อีกทั้งยังได้เข้าร่วมกิจกรรมต่าง ๆ ที่บริษัทได้จัดขึ้น ทำให้ได้รับประโยชน์ทั้งในด้านวิชาการ และในด้านทักษะและความคิดสร้างสรรค์มากมาย ซึ่งจะเป็นประโยชน์ต่อตัวของข้าพเจ้าเองในอนาคต

ข้าพเจ้าขอขอบพระคุณ บริษัท เอ็กซอนโมบิล จำกัด ที่ให้โอกาสข้าพเจ้าได้ร่วมทำงานและร่วมกิจกรรมต่าง ๆ มากมายที่สามารถนำไปปรับใช้ในอนาคต

ข้าพเจ้าขอขอบพระคุณ คุณอนุภาพ พิศลยบุตร คุณกรณห์หทัย กล้าคำ คุณพิรกร เอี่ยมเจริญยิ่ง คุณอนันท์ กานต์วิศิษฐ์ คุณณัฐนนท์ หาญวรเกียรติ คุณสนธิ กิตติกาญจนภา คุณพิพัฒน์ มุ่งจิตติวิศวกร และคุณนันทฉัตร มัลลิกะมาศ และอีกหลาย ๆ ท่านในทีม Web Content & Services ที่คอยดูแลข้าพเจ้าตลอดการทำงาน รวมถึงให้คำปรึกษา และช่วยเหลือในการทำงานของข้าพเจ้าจนสำเร็จลุล่วงไปได้ด้วยดี

ข้าพเจ้าขอขอบพระคุณ คุณกิตติศักดิ์ ธนอมธีระนันท์ คุณกายสิทธิ์ รัชชเทพา คุณชนิกานต์ ถาวรวงษ์ นางสาวพรสวรรค์ เริ่มศรี นายธนิชฐ์ โฆษิตศรีคุณากร และนายวสุพล จงสถิตเกียรติ ในฐานะสมาชิกทีม และเป็นส่วนหนึ่งของโปรเจกต์ Smart Locker ร่วมกันกับข้าพเจ้า โดยได้ร่วมกันปรึกษาหารือ ออกแนวความคิด ช่วยเหลือซึ่งกันและกัน แก้ปัญหา และฝ่าฟันอุปสรรคต่าง ๆ ที่เกิดขึ้นภายในโปรเจกต์นี้

ข้าพเจ้าขอขอบพระคุณ คุณพ่อ คุณแม่ ที่คอยดูแล ให้เวลา และกำลังใจแก่ข้าพเจ้าตลอดเวลา

ข้าพเจ้าขอขอบพระคุณ รศ.ดร.ชวลิต เบญจางคประเสริฐ ผู้ซึ่งเป็นอาจารย์ที่ปรึกษา ที่ให้คำแนะนำ คำปรึกษาทั้งในด้านการทำงาน และการทำรายงานวิจัยฉบับนี้ให้สำเร็จลุล่วงไปด้วยดี

และสุดท้าย ข้าพเจ้าขอขอบพระคุณเหล่าคณาจารย์และฝ่ายกิจการนักศึกษาคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้ความรู้ ทำให้ข้าพเจ้าได้มีโอกาสร่วมทำงานกับทางบริษัท และให้ความช่วยเหลือด้านคำปรึกษา คำแนะนำในการเข้าร่วมโครงการสหกิจศึกษาในครั้งนี้

สุชาวลี โก

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	I
ABSTRACT	II
กิตติกรรมประกาศ	III
สารบัญ.....	IV
สารบัญรูป	VI
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ขอบเขตของการวิจัย	2
1.4 วิธีการดำเนินงานวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	3
2.1 User Interface (UI).....	3
2.2 User Experience (UX)	3
2.3 Wireframe	3
2.4 Prototype	5
2.5 Adobe XD (Adobe Experience Design)	5
2.6 Visual Studio Code	7
2.7 Git.....	9
2.8 Vue.js.....	11
2.9 npm (Node.js package manager).....	13

2.10 Sass (Syntactically awesome style sheets)	13
2.11 jQuery	14
2.12 Web APIs (Application Programming Interface)	15
2.13 JSON (JavaScript Object Notation).....	15
บทที่ 3 วิธีการดำเนินงานวิจัย	17
3.1 รวบรวมความต้องการ (Requirements)	17
3.2 การออกแบบ และทดลองใช้งาน	19
3.3 การพัฒนาเว็บไซต์จริง.....	25
3.4 ทดสอบการใช้งานเว็บไซต์	42
บทที่ 4 ผลการวิจัย.....	44
4.1 ข้อมูลอุปกรณ์เปลี่ยนแปลงตามสถานที่ของผู้ใช้งาน	44
4.2 การใช้งานฟังก์ชัน Notify Me/Unnotify	46
4.3 การกรองข้อมูลอุปกรณ์.....	51
4.4 การตรวจสอบจำนวนอุปกรณ์ที่สามารถยืมได้ผ่านทางเว็บไซต์.....	52
บทที่ 5 สรุปผลการวิจัย และข้อเสนอแนะ	54
5.1 สรุปผลการวิจัย.....	54
5.2 ข้อเสนอแนะ	54
เอกสารอ้างอิง.....	55

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 การร่าง Wireframe ด้วยดินสอบนกระดาษ.....	4
2.2 การร่าง Wireframe ด้วยโปรแกรมคอมพิวเตอร์.....	4
2.3 โปรแกรม Adobe XD.....	5
2.4 ส่วน Design ใน Adobe XD.....	6
2.5 ส่วนร่าง Prototype ในโปรแกรม Adobe XD.....	6
2.6 Prototype Preview บนโปรแกรม Adobe XD.....	7
2.7 โปรแกรม Visual Studio Code.....	8
2.8 Interface ของโปรแกรม Visual Studio Code.....	8
2.9 โปรแกรม Git.....	9
2.10 ผู้ให้บริการ Remote Repository จากซ้ายไปขวา GitHub, GitLab, Bitbucket ตามลำดับ.....	9
2.11 การสร้าง Branch และ Merge Branch.....	10
2.12 Graph ของการ Commit บน Remote Repository.....	11
2.13 Vue.js.....	12
2.14 ตัวอย่างการเขียน Vue.js.....	12
2.15 npm.....	13
2.16 Sass.....	13
2.17 ตัวอย่างการเขียน Sass.....	14
2.18 jQuery.....	14
2.19 ตัวอย่างการเขียน jQuery.....	14
2.20 ตัวอย่าง API ที่ใช้งานในโปรเจกต์ Smart Locker.....	15
2.21 ตัวอย่างของข้อมูลแบบ JSON.....	16
2.22 ตัวอย่างการส่งข้อมูลแบบ JSON ผ่าน API.....	16
3.1 หน้า Dashboard ที่ได้ออกแบบไว้.....	19
3.2 หน้า Browse Item ที่ได้ออกแบบไว้.....	20
3.3 หน้า History ที่ได้ออกแบบไว้.....	21
3.4 ส่วนของ Owner ที่ได้ออกแบบไว้.....	22

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อมิรศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 หน้า Admin ที่ได้ออกแบบไว้.....	24
3.6 รูปและชื่อของ User ที่แสดงบน Navigation Bar.....	26
3.7 Select Location Modal	26
3.8 คำสั่งตรวจสอบว่ามีผู้ใช้งานในระบบหรือไม่.....	27
3.9 คำสั่งสร้างผู้ใช้งานนี้ เมื่อไม่พบผู้ใช้งานในระบบ	27
3.10 คำสั่งอัปเดตผู้ใช้งานนี้ เมื่อพบผู้ใช้งานในระบบ	28
3.11 การเปิด Select Location Modal อีกรูปแบบหนึ่ง.....	29
3.12 หน้า Dashboard บนเว็บไซต์จริง	29
3.13 การเขียน Template เพื่อแสดงข้อมูล	30
3.14 ส่วนของ My Items	30
3.15 คำสั่งเรียกดูข้อมูลการยืมของผู้ใช้งาน.....	31
3.16 หน้าข้อมูลของอุปกรณ์ที่กำลังยืมอยู่.....	32
3.17 Report Issue Modal.....	33
3.18 คำสั่งเรียกข้อมูลของ Watch List.....	33
3.19 หน้าข้อมูลของอุปกรณ์ที่ติดตามอยู่.....	34
3.20 ปุ่ม Notify Me.....	35
3.21 ปุ่ม Unnotify.....	35
3.22 คำสั่งเรียกข้อมูลของอุปกรณ์พร้อมกับสถานะการติดตามอุปกรณ์.....	35
3.23 เงื่อนไขการแสดงรูปปุ่ม Notify Me และ Unnotify	36
3.24 หน้า Browse Item	36
3.25 คำสั่งเรียกข้อมูล Category, Sub-category และข้อมูลอุปกรณ์ ผ่าน API.....	37
3.26 คำสั่ง createItems().....	38
3.27 การส่งค่า Parameters ผ่านทาง Route.....	39
3.28 คำสั่ง Filter อุปกรณ์ในหน้า Browse Item.....	39
3.29 คำสั่ง Search Item	39
3.30 หน้าข้อมูลเพิ่มเติมของอุปกรณ์ในกรณีที่ยังมีของให้ยืม	40
3.31 หน้าข้อมูลเพิ่มเติมของอุปกรณ์ในกรณีที่ของหมด.....	40
3.32 คำสั่งติดตามอุปกรณ์ และคำสั่งยกเลิกการติดตามอุปกรณ์.....	41
3.33 คำสั่ง postNotifyMeUser().....	41
3.34 Watch List หลังจากกดติดตามอุปกรณ์	42

3.35 คำสั่ง putUnnotifyMeUser().....	42
4.1 ข้อมูลที่แสดงของตึกहरिन्ธर ประเทศไทย.....	44
4.2 ทดลองเปลี่ยน Site ของผู้ใช้งาน.....	45
4.3 ข้อมูลที่แสดงของตึกเอสโซ่ ประเทศไทย.....	45
4.4 หน้า Dashboard ก่อนกดติดตามอุปกรณ์.....	46
4.5 หน้า Browse Item ที่แสดงอุปกรณ์ชื่อ Snowball.....	47
4.6 ข้อมูลเพิ่มเติมของอุปกรณ์ Snowball และปุ่ม Notify Me.....	47
4.7 หลังจากทำการกดปุ่ม Notify Me สีฟ้า.....	48
4.8 หน้า Dashboard หลังจากกดปุ่ม Notify Me.....	48
4.9 ทำการกดปุ่ม Unnotify เพื่อยกเลิกการติดตาม.....	49
4.10 ปุ่มได้เปลี่ยนกลับเป็นปุ่ม Notify สีฟ้า หลังจากกดยกเลิกการติดตาม.....	50
4.11 อุปกรณ์ Snowball ถูกลบออกจาก Watch List.....	50
4.12 กรองด้วยชื่อ Category.....	51
4.13 กรองด้วยชื่อ Sub-category.....	51
4.14 การพิมพ์ค้นหาด้วยตัวอักษร 'h' เพียงตัวเดียว.....	52
4.15 การพิมพ์ค้นหาด้วยตัวอักษร 'he'.....	52
4.16 กด View Item ของอุปกรณ์ที่สนใจจะตรวจสอบจำนวน.....	53
4.17 ข้อมูลเพิ่มเติมของอุปกรณ์ และจำนวนของที่เหลือ.....	53
4.18 เงื่อนไขของสีที่แสดงต่อจำนวนของที่เหลือ.....	53

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

บริษัท เอ็กซอนโมบิล จำกัด เป็นบริษัทที่ดำเนินธุรกิจด้านปิโตรเลียมแบบครบวงจร ทั้งด้านการค้นหาและทำการขุดเจาะน้ำมันดิบ (Upstream) การกลั่นน้ำมัน (Downstream) และในด้านเคมีภัณฑ์ (Chemicals) โดยภายในประเทศไทยจะมีโรงงานกลั่นน้ำมัน โรงงานอะโรเมติกส์ และเครือข่ายสถานีบริการน้ำมันทั่วประเทศ รวมไปถึงธุรกิจน้ำมันหล่อลื่น ภายใต้ชื่อการค้า “เอสโซ่” และ “โมบิล”

การดำเนินงานภายในประเทศไทย ทางบริษัทได้ก่อตั้งศูนย์บริการธุรกิจกรุงเทพ (Bangkok Business Support Center) ซึ่งเป็นศูนย์บริการธุรกิจที่ใหญ่ที่สุดของบริษัท เพื่อที่จะให้บริการแก่บริษัทในเครือเอ็กซอนโมบิล รวมไปถึงลูกค้าอีก 60 กว่าประเทศทั่วโลก ซึ่งหนึ่งในการให้บริการของศูนย์บริการธุรกิจ คือ ด้านเทคโนโลยีสารสนเทศ (ExxonMobil Information Technology หรือ EMIT) ซึ่งเป็นส่วนที่ผู้วิจัยได้มีโอกาสร่วมทำงานภายใต้โครงการฝึกงานภาคฤดูร้อนและโครงการสหกิจศึกษา ปีการศึกษา 2561

ผู้วิจัยได้เข้าไปร่วมทำงานภายใต้แผนก Web Content & Services (WCS) ซึ่ง WCS จะดูแลการให้บริการทางด้านเว็บไซต์ต่าง ๆ มากมายภายในองค์กร ไม่ว่าจะเป็นการเก็บข้อมูลทางด้านสถิติของการใช้เว็บไซต์นั้น ๆ แล้วนำข้อมูลมาวิเคราะห์ผล จึงทำให้ผู้วิจัยมีโอกาสได้รับงานเกี่ยวกับเว็บไซต์หลายงาน ไม่ว่าจะเป็นการสร้างเว็บไซต์ใหม่ให้กับแผนกอื่น หรือการปรับปรุงลักษณะภาพรวมของเว็บไซต์ และรวมถึงการสร้างเว็บไซต์ของโปรเจกต์ Smart Locker อีกด้วย

โปรเจกต์ Smart Locker เป็นโปรเจกต์ที่เกิดขึ้น จากการจัดกิจกรรมของทาง EMIT ที่ชื่อว่า EMIT Bootcamp 2018 ซึ่งเป็นกิจกรรมที่เปิดโอกาสให้พนักงานสามารถส่งหัวข้อความคิดสร้างสรรค์ที่ตนเองต้องการนำมาพัฒนาต่อยอด เพื่อที่จะสร้างคุณประโยชน์ให้กับทางบริษัท หรือช่วยทำให้การทำงานภายในองค์กรมีความสะดวกสบายมากขึ้น โดย Smart Locker ก็เป็นหนึ่งในหัวข้อที่ถูกนำเสนอขึ้นมาในกิจกรรมครั้งนี้ เนื่องจากโปรเจกต์ Smart Locker เคยได้รับการพัฒนามาก่อนแล้ว แต่ระบบไม่สามารถตอบสนองการใช้งานที่ดีให้กับผู้ใช้ได้ จึงทำให้มีการหยุดการใช้งานไป จนกระทั่งมีความต้องการที่จะพัฒนาระบบให้ตอบสนองแก่ผู้ใช้ให้ดียิ่งขึ้น จึงมีการนำเสนอมานเป็นหัวข้อ Smart Locker V.2 ซึ่งผู้วิจัยได้ทำงานร่วมกับสมาชิกทีมอื่น ๆ ประกอบไปด้วยนักศึกษาฝึกงานอีก 3 คน และพนักงานภายในบริษัทอีก 5 คน

เมื่อได้มีการพูดคุยกัน จึงได้มีการแบ่งงานในส่วนที่แต่ละคนต้องรับผิดชอบ โดยผู้วิจัยได้รับมอบหมายในส่วนของการออกแบบการทำงานเว็บไซต์โดยรวม และการสร้างเว็บไซต์ส่วนของผู้ใช้งาน (Users) ซึ่งจะเอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบไปด้วยหน้าเพจสรุปข้อมูลของบุคคลนั้น (Dashboard) และ หน้าเพจแสดงจำนวนอุปกรณ์ที่สามารถให้ยืมได้ โดยข้อมูลอุปกรณ์จะขึ้นอยู่กับการเลือกประเทศ และอาคารสำนักงานของบุคคลนั้น ๆ

1.2 วัตถุประสงค์ของการวิจัย

- 1.2.1 เพื่อเพิ่มความสะดวกสบาย และลดระยะเวลาในการยืมคืนอุปกรณ์ภายในองค์กร
- 1.2.2 แก้ไขปัญหาที่มีในรุ่นก่อนและพัฒนาเพื่อให้สามารถตอบสนองต่อการใช้งานให้ดียิ่งขึ้น
- 1.2.3 เพื่อที่จะให้ผู้ดูแลระบบสามารถดูแลระบบได้ง่ายขึ้น
- 1.2.4 พัฒนาระบบให้สามารถขยายและรองรับการใช้งานหลายสถานที่

1.3 ขอบเขตของการวิจัย

- 1.3.1 มีเว็บไซต์เพื่อให้ผู้ใช้งานสามารถเข้าถึงข้อมูลอุปกรณ์ที่สามารถยืมได้ง่ายยิ่งขึ้น
- 1.3.2 เว็บไซต์สามารถเชื่อมต่อไปยังฐานข้อมูลได้
- 1.3.3 เว็บไซต์สามารถตรวจสอบสถานะการยืมอุปกรณ์นั้น ๆ ได้

1.4 วิธีการดำเนินงานวิจัย

- 1.4.1 รวบรวมความต้องการต่าง ๆ และสิ่งที่ต้องการให้มีภายในเว็บไซต์
- 1.4.2 จัดลำดับความสำคัญงาน และวางแผนในการพัฒนาระบบ
- 1.4.3 ออกแบบเว็บไซต์ ลองให้ทดลองใช้งาน และเก็บรายละเอียด เพื่อนำไปแก้ไข
- 1.4.4 เริ่มพัฒนาเว็บไซต์จริง
- 1.4.5 ทดสอบใช้งานระบบร่วมกับฝั่งฐานข้อมูล

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 ลดความยุ่งยากในการยืมคืนอุปกรณ์ โดยการทำให้เป็นระบบ Self-Services
- 1.5.2 ลดทรัพยากรบุคคลที่ต้องคอยดูแลเรื่องการยืมคืนอุปกรณ์
- 1.5.3 ลดระยะเวลาที่ใช้เมื่อต้องการยืมหรือคืนอุปกรณ์
- 1.5.4 ผู้ใช้งานสามารถตรวจสอบสถานะอุปกรณ์ได้ผ่านทางเว็บไซต์โดยตรง
- 1.5.5 ผู้ใช้งานสามารถแจ้งเข้าระบบผ่านทางเว็บไซต์ได้ทันที หากอุปกรณ์ขัดข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อ 2 ระดับเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การพัฒนาเว็บไซต์นั้น จะแบ่งออกเป็นสองส่วนคือส่วนของการออกแบบเว็บไซต์โดยรวม หลังจากนั้นนำไปพัฒนาเป็นเว็บไซต์จริง จึงทำให้ผู้วิจัยต้องค้นคว้าและศึกษาทฤษฎีต่าง ๆ เพื่อที่จะนำมาใช้ในการทำงานให้สำเร็จลุล่วง โดยจะประกอบไปด้วยแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง ดังนี้

2.1 User Interface (UI)

User Interface (UI) คือส่วนที่มีไว้สื่อสาร หรือตอบสนองระหว่างผู้ใช้กับระบบหรือสิ่งของ ไม่ว่าจะเป็นคอมพิวเตอร์ โทรศัพท์มือถือ อุปกรณ์ต่าง ๆ หรือเว็บไซต์ ซึ่งจุดประสงค์ของ UI คือ เพื่อให้ผู้ใช้งานระบบสามารถใช้งาน หรือควบคุมระบบให้เป็นไปตามที่ต้องการได้อย่างถูกต้อง จึงทำให้ผู้ออกแบบจะต้องออกแบบให้ระบบรับอินพุต (Input) ที่ถูกต้อง นำไปประมวลผล (Process) แล้วแสดงออกมาเป็นเอาต์พุต (Output) ที่ถูกต้อง แต่นอกเหนือจากนั้นจะต้องออกแบบโดยคำนึงถึงความเหมาะสมการใช้งาน และมีประสิทธิภาพ

2.2 User Experience (UX)

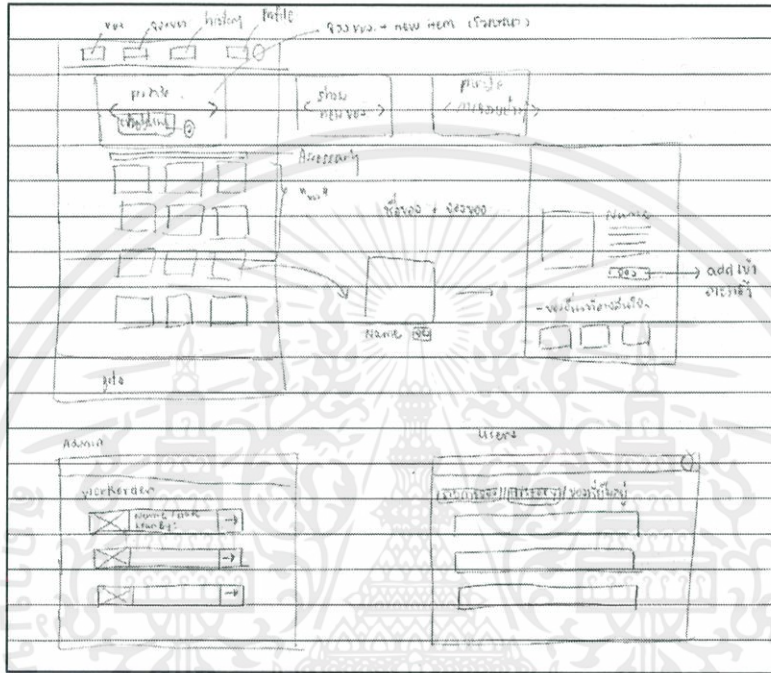
User Experience (UX) คือ ประสบการณ์ของผู้ใช้งานรู้สึกเมื่อทำการใช้งานระบบนั้น ว่าระบบนั้นมีความยากง่ายในการใช้งานต่างกันอย่างไร โดย User Experience เป็นผลพวงจากการออกแบบ User Interface ว่าจะทำให้มี User Experience แบบใด ซึ่งการมี User Experience ที่ดี หมายถึงระบบนั้นสามารถทำให้ผู้ใช้งานได้อย่างมีประสิทธิภาพ สามารถเข้าถึงข้อมูลได้ง่าย แต่ในทางกลับกัน เมื่อระบบนั้นมี User Experience ที่ไม่ดี ก็จะทำให้ผู้ใช้งานมีความรู้สึกว่ระบบมีความยากต่อการใช้งาน มีความซับซ้อน และส่งผลให้ผู้ใช้งาน ใช้งานระบบนี้ลดลง หรือเลิกใช้งานระบบเหล่านี้ไป

2.3 Wireframe

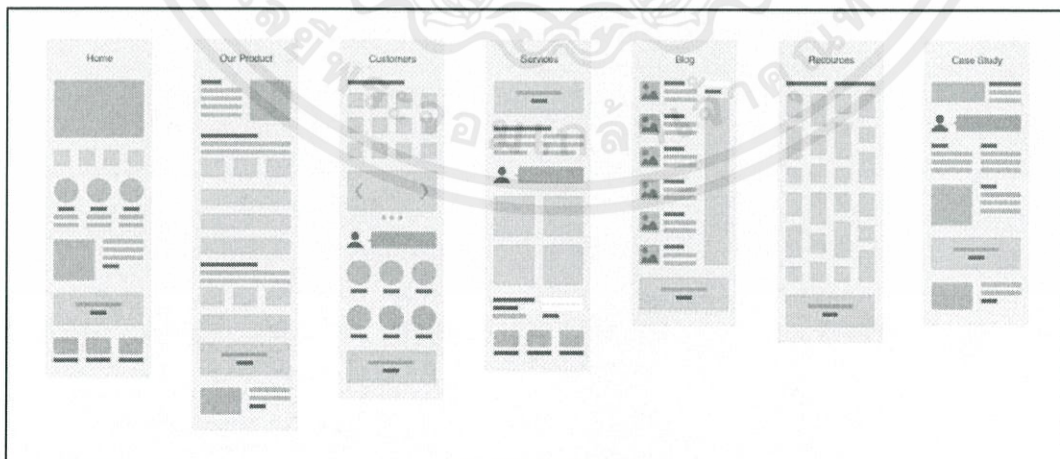
Wireframe คือ การร่างโครงหน้าของเว็บไซต์ หรือสามารถเรียกได้ว่าเป็น พิมพ์เขียวของเว็บไซต์ เพื่อให้ทราบว่าภายในหน้านั้น ๆ ประกอบไปด้วยข้อมูล หรือองค์ประกอบใดบ้าง อีกทั้งจะยังทำให้ทราบถึงว่าการจัดเรียงลักษณะข้อมูลภายในหน้านั้นเป็นอย่างไร มีลักษณะของ User Interface ที่ดีหรือไม่ แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลในแต่ละส่วนจะสามารถสื่อถึงภาพรวมเนื้อหาทั้งหมดภายในหน้านั้นได้หรือไม่ การทำ Wireframe นั้นจะไม่นิยมใส่ตัวหนังสือเข้าไป แต่จะแทนที่ด้วยกล่องสี่เหลี่ยมที่ระบุว่าเป็นส่วนนั้น คือองค์ประกอบอะไร โดยการทำ Wireframe จะนิยมอยู่ 2 แบบ ได้แก่ การร่างภาพด้วยดินสอบนกระดาษ หรือ การร่างภาพด้วยโปรแกรมคอมพิวเตอร์ ดังรูปที่ 2.1 และ 2.2 ตามลำดับ



รูปที่ 2.1 การร่าง Wireframe ด้วยดินสอบนกระดาษ



รูปที่ 2.2 การร่าง Wireframe ด้วยโปรแกรมคอมพิวเตอร์

(ที่มา: <https://www.gravitatedesign.com/blog/types-of-wireframes/>)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 Prototype

Prototype จะคล้ายกับการทำ Wireframe ซึ่งเป็นการร่างโครงร่างเว็บไซต์เหมือนกัน แต่จะแตกต่างกันตรงที่ ผู้ใช้งานสามารถทดลองใช้งานภายในตัว Prototype ได้ เช่น สามารถกดที่รูปปุ่ม เพื่อที่จะเปลี่ยนไปยังหน้าถัดไปได้ ทำให้สามารถทดลองใช้งานเพื่อดูว่าการจัดวางองค์ประกอบดังนี้จะทำให้ มี User Experience ที่ดีหรือไม่ โดยโปรแกรมที่นิยมใช้สร้าง Prototype ได้แก่ Adobe XD

2.5 Adobe XD (Adobe Experience Design)

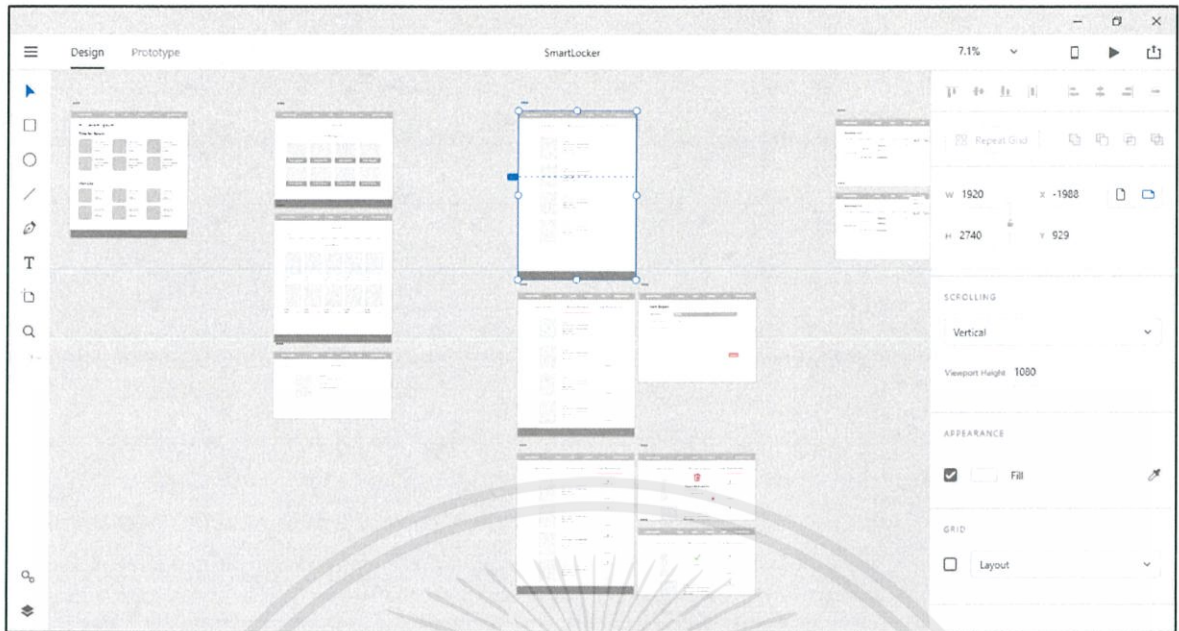
โปรแกรม Adobe XD โปรแกรมในเครือ Adobe ที่เปิดให้ใช้งานได้ฟรี เป็นเครื่องมือที่ใช้ในการ ออกแบบ User Experience สำหรับเว็บไซต์และแอปพลิเคชันบนโทรศัพท์มือถือ



รูปที่ 2.3 โปรแกรม Adobe XD

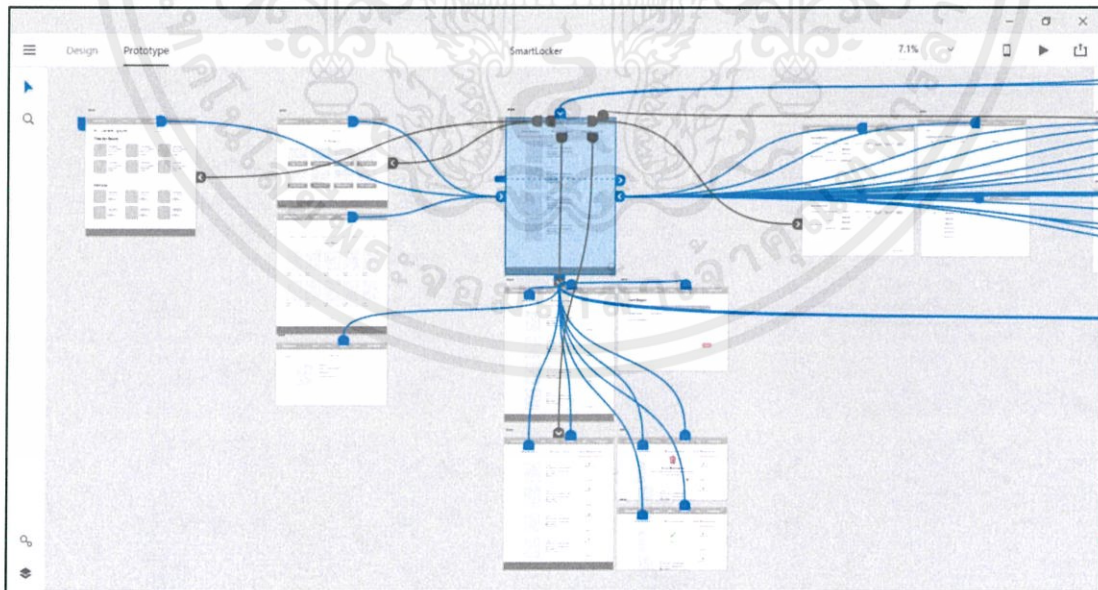
(ที่มา: https://commons.wikimedia.org/wiki/File:Adobe_XD_CC_icon.svg)

โปรแกรมจะแบ่งออกเป็น 2 ส่วน ได้แก่ ส่วน Design ซึ่งจะมีไว้ออกแบบ UI อย่างอิสระ ดังรูปที่ 2.4 ซึ่งผู้ออกแบบสามารถเลือกขนาดของ Artboard ได้ตามความต้องการ ไม่ว่าจะเป็นในขนาดของหน้าเว็บไซต์ หรือขนาดของหน้าจอโทรศัพท์มือถือ



รูปที่ 2.4 ส่วน Design ใน Adobe XD

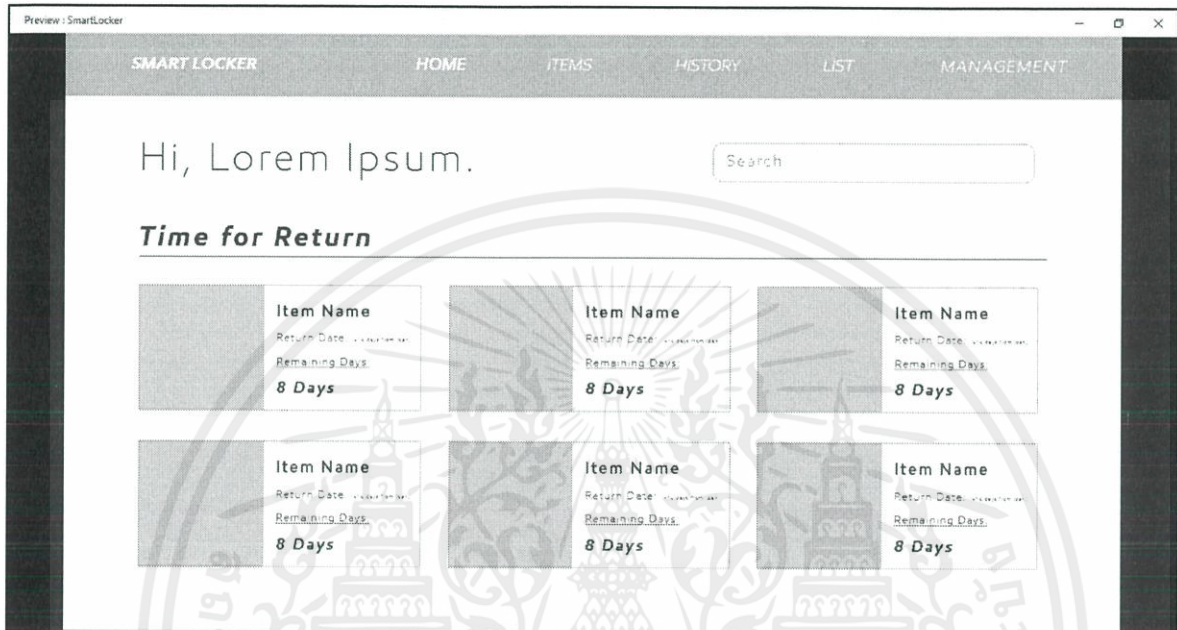
ส่วน Prototype ซึ่งจะเป็นการออกแบบในด้าน UX ว่าในแต่ละหน้าจะมีการเชื่อมต่อไปยังหน้าอื่นได้อย่างไร ดังรูปที่ 2.5



รูปที่ 2.5 ส่วนร่าง Prototype ในโปรแกรม Adobe XD

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการร่าง Prototype เสร็จแล้ว จะสามารถ Preview ตัว Prototype ได้ โดยที่ในหน้า Preview จะสามารถทดลองใช้งานหน้าที่ได้ออกแบบไว้ในส่วนพื้นฐาน เช่น การคลิกไปยังหน้าถัดไปได้ เหมือนกับการทดลองใช้บนเว็บไซต์จริง ดังรูปที่ 2.6



รูปที่ 2.6 Prototype Preview บนโปรแกรม Adobe XD

2.6 Visual Studio Code

Visual Studio Code เป็น Code Editor สำหรับนักพัฒนาโปรแกรมจากเครือ Microsoft ที่นิยมใช้กันอย่างแพร่หลาย เป็นโปรแกรมที่เปิดเป็น Open-source และให้ผู้ใช้สามารถใช้งานได้ฟรี โดยโปรแกรม Visual Studio Code นั้นรองรับได้หลายภาษาโปรแกรม อีกทั้งยังรองรับการทำ Debugging หรือการทำ Git Version Control และยังมี Extension อีกมากมายเพื่อที่จะเพิ่มลูกเล่นในการใช้งานให้กับโปรแกรมได้อีกด้วย

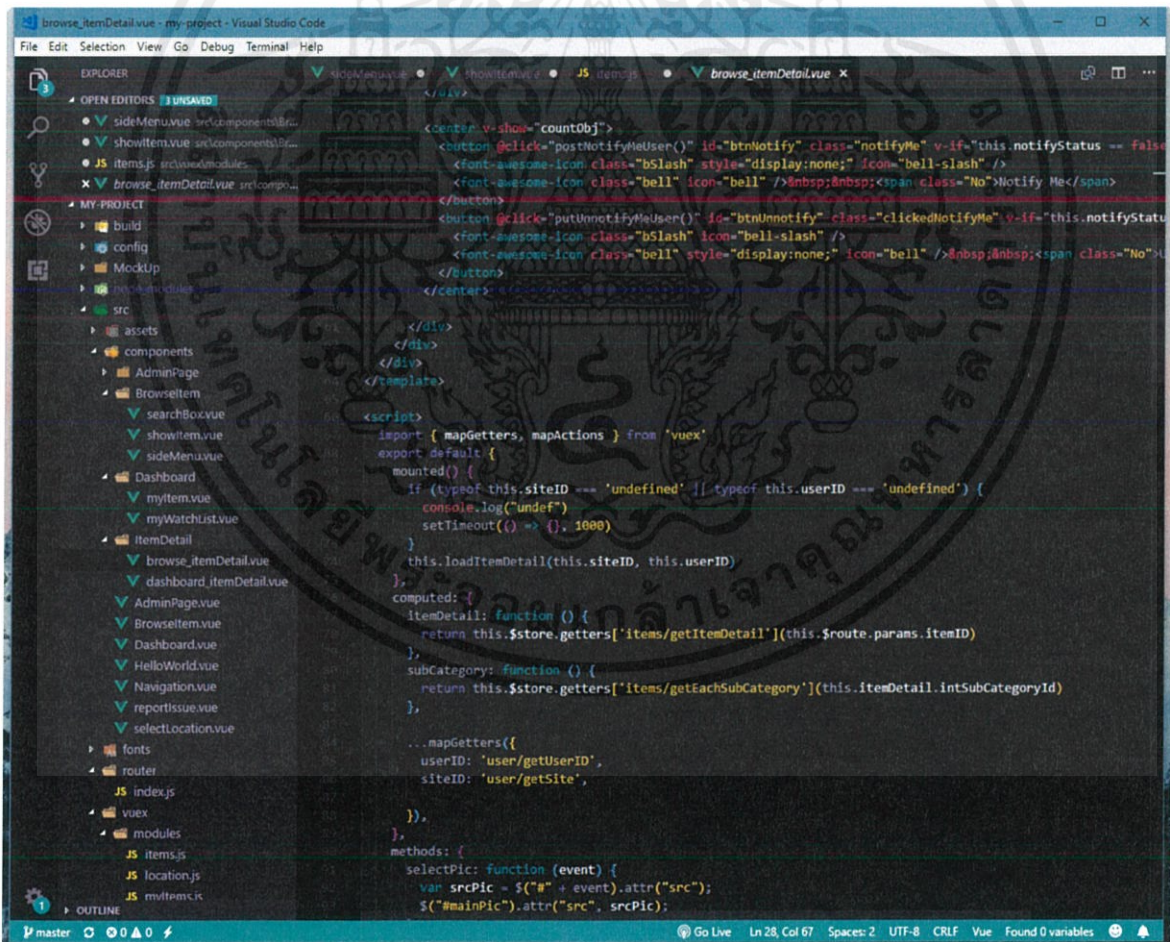
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Visual Studio Code

รูปที่ 2.7 โปรแกรม Visual Studio Code

(ที่มา: <https://launchdarkly.com/blog/launched-version-2-of-the-launchdarkly-visual-studio-code-extension/>)



รูปที่ 2.8 Interface ของโปรแกรม Visual Studio Code

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 Git

Git คือซอฟต์แวร์ที่ใช้ในการควบคุมรุ่นของซอฟต์แวร์หรือไฟล์ชนิดใดก็ได้ (Version Control) โดยตัวซอฟต์แวร์จะทำการติดตามความเปลี่ยนแปลงของไฟล์ อีกทั้งยังควบคุมการเปลี่ยนแปลงภายในไฟล์ในกรณีที่มีผู้ใช้งานไฟล์นั้น ๆ มากกว่า 1 คนขึ้นไป



รูปที่ 2.9 โปรแกรม Git

(ที่มา: <https://git-scm.com/images/logos/downloads/Git-Logo-2Color.png>)

Repository คือแหล่งข้อมูลที่เก็บไฟล์ หรือโปรแกรมที่กำลังอยู่ในขั้นตอนพัฒนาเอาไว้ ซึ่งจะมีอยู่สองแบบ คือ Local Repository ซึ่งเป็นโพลเดอร์ที่ใช้เก็บไฟล์เหล่านั้น และ Remote Repository ซึ่งเป็นการนำไฟล์ไปเก็บไว้บน Server ที่ให้บริการ Git เช่น GitHub GitLab หรือ Bitbucket เป็นต้น ซึ่งจะสามารถเก็บไฟล์ในรูปแบบ Public หรือ Private ก็ได้ และการเก็บไฟล์ไว้บน Server จะทำให้ผู้พัฒนาคนอื่นสามารถนำไฟล์ไปทำการพัฒนาต่อ หรือว่าแก้ไขได้ จึงเปรียบได้ว่า Remote Repository เป็น Central Repository และทำให้การทำงานเป็นแบบระบบ Centralized อีกด้วย

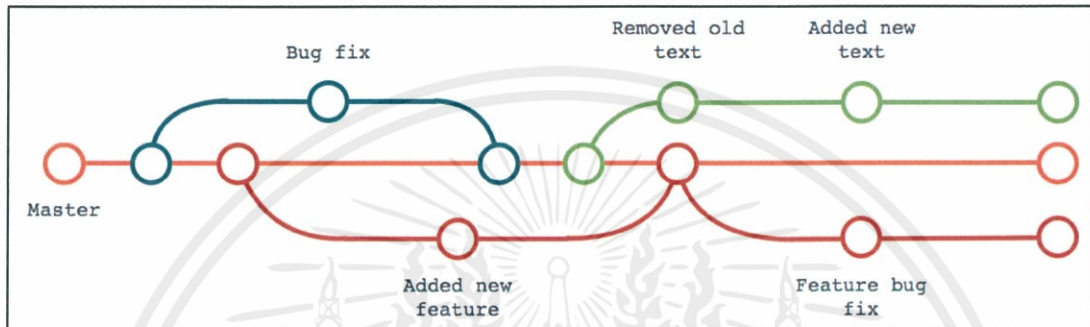


รูปที่ 2.10 ผู้ให้บริการ Remote Repository จากซ้ายไปขวา GitHub, GitLab, Bitbucket ตามลำดับ

(ที่มา: https://seesparkbox.com/foundry/github_vs_gitlab_vs_bitbucket)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

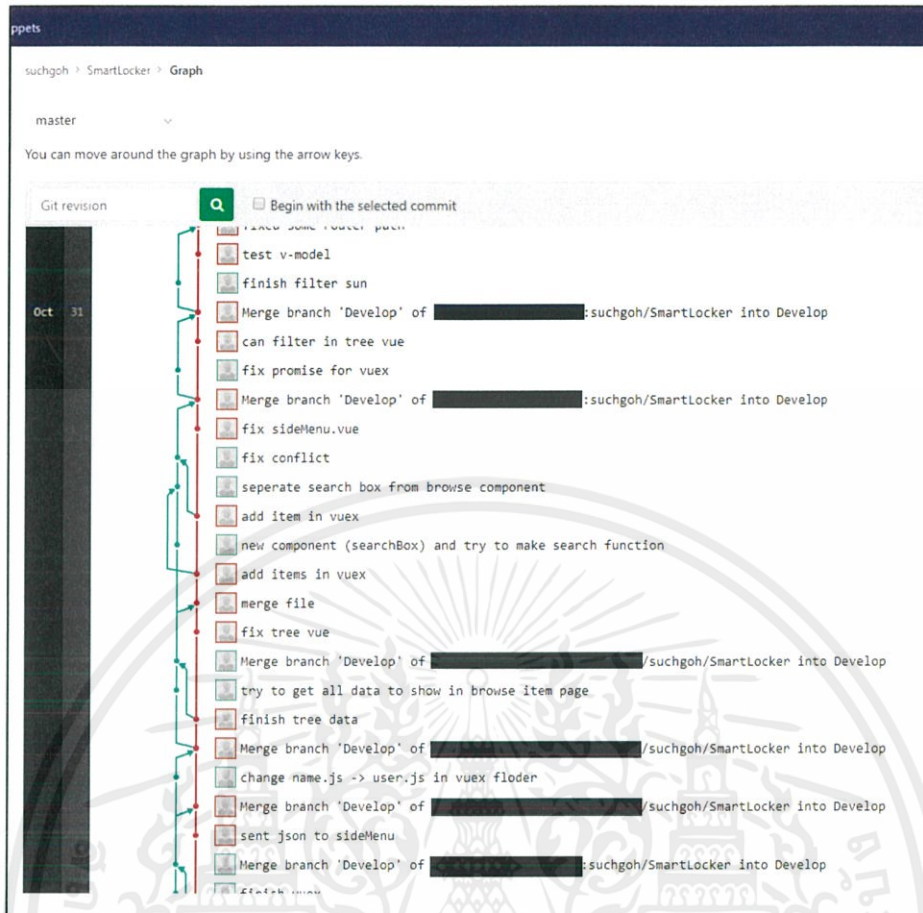
โดยโปรแกรม Git นั้นจะรองรับการสร้าง Branch โดยที่ตัวโปรแกรมหลักจะอยู่ใน Branch ที่ชื่อว่า Master เสมอ แต่เมื่อต้องการนำเป็นนำไฟล์ทั้งหมดไปแก้ไข Bug ที่เกิดขึ้น หรือเพิ่มฟีเจอร์ใหม่ ๆ จะต้องทำการสร้าง Branch ใหม่อีกอันเพื่อที่จะแยกออกมาจาก Master เพื่อที่จะให้การแก้ไขเหล่านั้น ไม่กระทบกับตัวโปรแกรมหลัก และเมื่อทำการแก้ไขเสร็จสิ้น ก็จะทำ Merge Branch ที่สร้างเข้ามาใหม่ กลับไปยัง Master เพื่อให้ตัวโปรแกรมหลักอัปเดตมาเป็นรุ่นที่ได้รับการแก้ไข หรือเพิ่มฟีเจอร์ใหม่แล้ว



รูปที่ 2.11 การสร้าง Branch และ Merge Branch

(ที่มา: <https://blog.cpanel.com/git-version-control-series-what-is-git/>)

หลังจากทำการแก้ไข หรือเพิ่มฟีเจอร์ใหม่ จะต้องทำการ Commit และ Push เพื่อให้ไฟล์ที่ได้รับการแก้ไขแล้วถูกอัปเดตขึ้นไปบน Remote Repository ด้วย โดยที่ Git จะทำการเก็บบันทึกทุกครั้งที่มีการเปลี่ยนแปลงภายใน Repository นั้น ๆ ว่าบุคคลใดเป็นผู้แก้ไข



รูปที่ 2.12 Graph ของการ Commit บน Remote Repository

2.8 Vue.js

Vue.js เป็น JavaScript Framework ที่ช่วยในการในการสร้างหน้า User Interface ให้ง่ายยิ่งขึ้น โดยตัว Vue.js เป็น Framework ที่นิยมเนื่องจากการเขียนแบบ HTML-Based ซึ่งจะมีความใกล้เคียงกับการเขียน HTML ปกติ ทำให้มีระยะเวลาในการศึกษาเรียนรู้สั้นกว่า Framework ตัวอื่น อีกทั้ง Vue.js ยังรองรับการทำ Single-Page Application (SPA) ซึ่งหมายถึงจะเป็นการแสดงผลหน้าเว็บเพียงหน้าเดียว แต่จะมีการเปลี่ยนข้อมูลไปตามที่ผู้ใช้งานกระทำต่อระบบ โดยตัวหน้าเว็บจะทำการเปลี่ยนแปลงลักษณะ หรือข้อมูลภายในหน้านั้น ๆ ด้วยการทำให้ Dynamic Rendering แทนการโหลดทั้งหน้าใหม่จาก Server จึงทำให้เว็บที่เป็น SPA จะมีความลื่นไหลมาก และสามารถใช้งานได้อย่างไม่ติดขัด



รูปที่ 2.13 Vue.js

(ที่มา: <https://github.com/vuejs/art/blob/master/logo.png>)

```
<template>
  <div class="treeNavigation">
    <li>
      <router-link :to="{ name: 'BrowseItem', params: {categoryID: 0} }">All</router-link>
    </li>
    <li v-for="(category, index) in model" :key="index" :category="model">
      <span v-if="isFolder(category)" @click="toggle(category)">{{ category.open ? '&#x25BD;' : '&#x25B6;' }}</span>
      <router-link :to="{ name: 'BrowseItem', params: {categoryID: category.id} }">&nbsp;{{category.name}}</router-link>
      <ul v-show="category.open">
        <li v-for="(subCat, indexChild) in category.children" :key="indexChild" :subCat="category" class="subCatMenu">
          <router-link :to="{ name: 'BrowseItem', params: {subCategoryID: subCat.id , categoryID: category.id} }">
            &nbsp;{{subCat.name}}
          </router-link>
        </li>
      </ul>
    </li>
  </div>
</template>
<script>
  import { mapGetters, mapActions } from 'vuex'

  export default {
    data() {
      return {
        model: []
      }
    },
    mounted() {
      // console.log("going to create item")
      this.isExist()
      this.$router.push({
        name: 'BrowseItem',
        params: {
          categoryID: 0
        }
      })
    }
  },
}
```

รูปที่ 2.14 ตัวอย่างการเขียน Vue.js

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 npm (Node.js package manager)

npm เป็นสิ่งที่ถูกติดตั้งมากับ Node.js โดยจะทำหน้าที่ในการจัดการ Package เสริม ที่สามารถดาวน์โหลด Modules หรือ Library ต่าง ๆ ลงมาใช้งานร่วมกับโปรแกรมที่กำลังพัฒนาอยู่ ซึ่งจะทำให้การพัฒนาโปรแกรมเป็นไปได้อย่างง่ายมากขึ้น



รูปที่ 2.15 npm

(ที่มา: <https://github.com/npm/logos/blob/master/npm%20logo/classic/npm-2009.png>)

2.10 Sass (Syntactically awesome style sheets)

SASS เป็นภาษาสคริปต์ชนิด Preprocessor ซึ่งต่างจากการเขียน CSS แบบเดิม ซึ่ง Sass จะมีฟังก์ชันมาให้ใช้งานได้ง่ายขึ้น เช่น สามารถสร้างตัวแปรเก็บค่าแล้วนำไปใช้งานหลาย ๆ ที่ได้ แต่ใน CSS เดิมไม่สามารถทำได้ การใช้งาน Sass สามารถทำได้โดยที่โปรแกรมจำเป็นจะต้องนำ Sass ไป Compile ให้เป็น CSS ก่อน จึงจะสามารถใช้งานได้ตามปกติ



รูปที่ 2.16 Sass

(ที่มา: <https://sass-lang.com/styleguide/brand>)

```
.menu {
  display: flex;
  align-items: center;
  justify-content: flex-end;
  padding-right: 0;

  p {
    cursor: pointer;
    &:hover,
    &:active,
    &:focus,
    {
      font-weight: 600;
    }
  }
}
```

รูปที่ 2.17 ตัวอย่างการเขียน Sass

2.11 jQuery

jQuery คือ JavaScript Library ที่สร้างเพื่อช่วยให้ผู้พัฒนาโปรแกรมเขียน JavaScript ได้สะดวกและง่ายขึ้น โดยมีความสามารถหลากหลายด้าน เช่น การทำ Event Handling, การทำ Animation หรือการทำแอปพลิเคชันแบบ Ajax (Asynchronous JavaScript and XML)



รูปที่ 2.18 jQuery

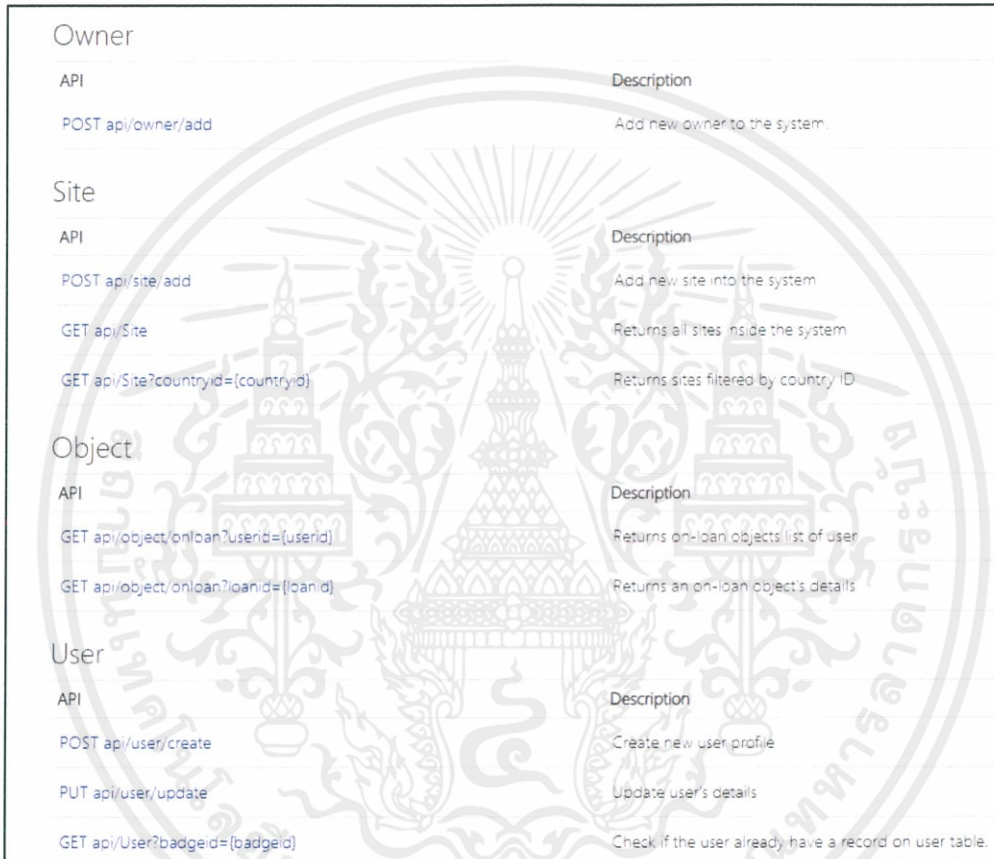
(ที่มา: <https://brand.jquery.org/logos/#the-mark>)

```
selectPic: function (event) {
  var srcPic = $("#" + event).attr("src");
  $("#mainPic").attr("src", srcPic);
},
```

รูปที่ 2.19 ตัวอย่างการเขียน jQuery

2.12 Web APIs (Application Programming Interface)

Web APIs คือส่วนติดต่อกันระหว่างเว็บไซต์และ Server หรือฐานข้อมูล เพื่อที่จะทำการแลกเปลี่ยนข้อมูลกัน หรือทำการเปลี่ยนแปลงข้อมูลในระบบ โดยในจะมีการรับส่งข้อมูลในรูปแบบ JSON ซึ่งจะทำงานโดยใช้ HTTP Request ในการติดต่อกับตัว API เช่น หากต้องการดูข้อมูลของอุปกรณ์ชิ้นนี้ ก็จะต้อง GET ข้อมูลจาก API ก่อนแล้วจึงนำมาแสดงผลบนเว็บไซต์



Owner		
API		Description
POST api/owner/add		Add new owner to the system.

Site		
API		Description
POST api/site/add		Add new site into the system
GET api/site		Returns all sites inside the system
GET api/site?countryid={countryid}		Returns sites filtered by country ID

Object		
API		Description
GET api/object/onloan?userid={userid}		Returns on-loan objects list of user
GET api/object/onloan?loanid={loanid}		Returns an on-loan object's details

User		
API		Description
POST api/user/create		Create new user profile
PUT api/user/update		Update user's details
GET api/user?badgeid={badgeid}		Check if the user already have a record on user table.

รูปที่ 2.20 ตัวอย่าง API ที่ใช้งานในโปรเจกต์ Smart Locker

2.13 JSON (JavaScript Object Notation)

JSON คือ Format ของการรับส่งข้อมูลแบบ Object ที่เขียนอยู่ในรูปแบบ Text-based ทำให้สามารถอ่านและเข้าใจได้ง่าย ซึ่งจะนิยมใช้กับการติดต่อระหว่าง Browser กับ Server แบบ Asynchronous โดยโครงสร้างของ JSON จะประกอบไปด้วย Object ที่เป็นคู่ของ Name และ Value หรือ จะเป็น Array ที่จะเก็บ Value หลายค่าไว้เป็น Collection

```

items: [
  { name: "All" },
  {
    name: "Input Devices",
    children: [
      { name: "Keyboard" },
      { name: "Mouse" },
      { name: "Bundled Devices" },
      { name: "Badge Scanner" }
    ]
  },
  {
    name: "Laptop Accessories",
    children: [
      { name: "Laptop Adapter" },
      { name: "Cable Lock" }
    ]
  }
]

```

รูปที่ 2.21 ตัวอย่างของข้อมูลแบบ JSON

```

this.$axios.put('http://localhost:3000/api/user/update', {
  UserId: this.userID,
  UserBadgeId: this.userData.userBadgeID,
  UserRole: this.userData.userRole,
  Description: this.userData.userLanID,
  SiteId: userSite,
  UpdaterId: this.userID,
  UpdaterBadgeId: this.userData.userBadgeID

```

รูปที่ 2.22 ตัวอย่างการส่งข้อมูลแบบ JSON ผ่าน API

บทที่ 3

วิธีการดำเนินงานวิจัย

เนื่องจากโปรเจกต์นี้เป็นโปรเจกต์กลุ่มที่เข้าร่วมในกิจกรรม EMIT Bootcamp 2018 ทำให้ผู้วิจัยต้องทำงานร่วมกับเพื่อนนักศึกษาฝึกงานคนอื่น และพี่ ๆ ร่วมแผนก โดยงานที่ได้รับมอบหมายจะถูกแบ่งให้กับนักศึกษาฝึกงานเป็นส่วนใหญ่ ส่วนพี่ ๆ ร่วมแผนกจะเป็นฝ่ายให้คำปรึกษา และคอยช่วยเหลือเมื่อพบปัญหา ตลอดการทำงานจะมีการประชุมงาน 1 วัน ต่ออาทิตย์ เพื่อเป็นการสรุปความคืบหน้าที่ได้ทำมาในแต่ละอาทิตย์ และวางแผนที่จะทำในส่วนใดต่อ หรือบอกกล่าวถึงปัญหาที่เกิดขึ้นในอาทิตย์นั้น และหาทางแก้ไขปัญหาร่วมกัน โดยในการวิจัยจะเริ่มต้นจากการเก็บความต้องการ (Requirements) ของแต่ละฝ่าย จากนั้นนำมารวบรวมและจัดลำดับความสำคัญ วางแผนการทำงานและเริ่มการออกแบบ เมื่อออกแบบเสร็จสิ้นจะทดสอบการใช้งานเพื่อให้เว็บไซต์มี User Experience ที่ดี เมื่อทำการแก้ไขทั้งหมดจะเข้าสู่การพัฒนาเว็บไซต์จริงและทดสอบการใช้งานเพื่อค้นหาจุดบกพร่องเพื่อทำการแก้ไข ซึ่งแผนการทำงานจะเป็นไปตาม ตารางที่ 3.1

ระยะเวลา	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน				ธันวาคม			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
หัวข้อ																				
รวบรวมความต้องการ																				
ออกแบบเว็บไซต์																				
พัฒนาเว็บไซต์																				
ทดสอบการใช้งาน																				

ตารางที่ 3.1 แผนการทำงานวิจัย

3.1 รวบรวมความต้องการ (Requirements)

เนื่องจากเดิมโปรเจกต์ Smart Locker นั้นเคยถูกพัฒนามาแล้วครั้งหนึ่ง แต่ระบบไม่ตอบสนองอย่างมีประสิทธิภาพ จึงทำให้เลิกใช้งานไป ผู้วิจัยจึงต้องค้นคว้าหาสาเหตุ และเก็บรวบรวมความต้องการจากผู้ใช้งานระบบ ซึ่งจะแบ่งออกเป็น 3 กลุ่มของผู้ใช้งาน ได้แก่ ผู้ใช้งานทั่วไป (Users) ผู้ใช้งานที่เป็นเจ้าของอุปกรณ์ที่ให้อืม (Owner) และผู้ใช้งานที่เป็นผู้ดูแลระบบ (Admin) ซึ่งในแต่ละกลุ่ม ก็จะมีความต้องการแตกต่างกันออกไป ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 ผู้ใช้งานทั่วไป (Users)

1. เว็บไซต์สามารถบอกจำนวนของอุปกรณ์แต่ละชิ้นที่เหลืออยู่
2. สามารถบอกวันที่ต้องคืนอุปกรณ์ให้ทราบ
3. มีการแจ้งเตือนเมื่อใกล้ถึงวันที่ต้องคืนอุปกรณ์นั้น ๆ

3.1.2 ผู้ใช้งานที่เป็นเจ้าของอุปกรณ์ที่ให้ยืม (Owner)

อุปกรณ์ต่าง ๆ ที่ให้ยืม จะต้องมีส่วนที่เป็นเจ้าของคอยดูแลอุปกรณ์เหล่านั้น หรือทำการซื้อเข้ามาเพิ่มในกรณีที่มีความต้องการใช้งานสูง จึงทำให้ฝ่าย Owner ต้องการให้ภายในระบบมีดังนี้

1. สามารถทำรายงานแจ้งจำนวนการยืมอุปกรณ์ประจำเดือน
2. จัดลำดับอุปกรณ์ที่มีแนวโน้มการยืมสูง เพื่อนำไปพิจารณาในการซื้อเพิ่ม
3. ทำรายงานแจ้งเตือนเมื่อมีการแจ้งเข้ามาในระบบว่ามีของชำรุด

3.1.3 ผู้ใช้งานที่เป็นผู้ดูแลระบบ (Admin)

ตามปกติแล้ว Admin จะเป็นคนดูแลเรื่องการยืมคืนของอยู่ภายในฝ่าย ITSC (IT Solution Center) ของแต่ละอาคาร โดยการยืมทุกครั้งจะมีการสร้าง Work Order ขึ้นมาเพื่อที่จะระบุว่าผู้ใช้รายนี้เป็นบุคคลที่ทำการยืมของไป ซึ่ง Work Order เป็นเหมือนคำสั่งงานที่ใช้ภายในบริษัทว่าผู้ใช้รายนี้ มีความต้องการที่จะทำอะไรต่อระบบ หรือมีการร้องขอให้ทำงานอะไร โดยจะมีการเก็บข้อมูลของผู้ร้องขอ วันที่ร้องขอ และการกำหนดผู้ดูแลในงานนั้น ๆ เมื่อมีการนำของมาคืน ผู้ดูแลก็จะทำการปิด Work Order หมายเลขนั้น เพื่อเป็นการระบุว่ามีการคืนอุปกรณ์เรียบร้อยแล้ว แต่ถ้าหากเมื่อระบบยืมคืนของถูกเปลี่ยนเป็นระบบอัตโนมัติ จะทำให้ Admin ไม่สามารถติดตามของที่ถูกยืมไปได้ จึงทำให้ทางฝ่าย Admin มีความต้องการดังนี้

1. สามารถดูของที่มีอยู่ในระบบได้ทั้งหมด
2. สามารถเพิ่ม ลด แก้ไข อุปกรณ์ที่ให้ยืมได้ผ่านทางเว็บไซต์
3. สามารถติดตามของที่ถูกยืมได้

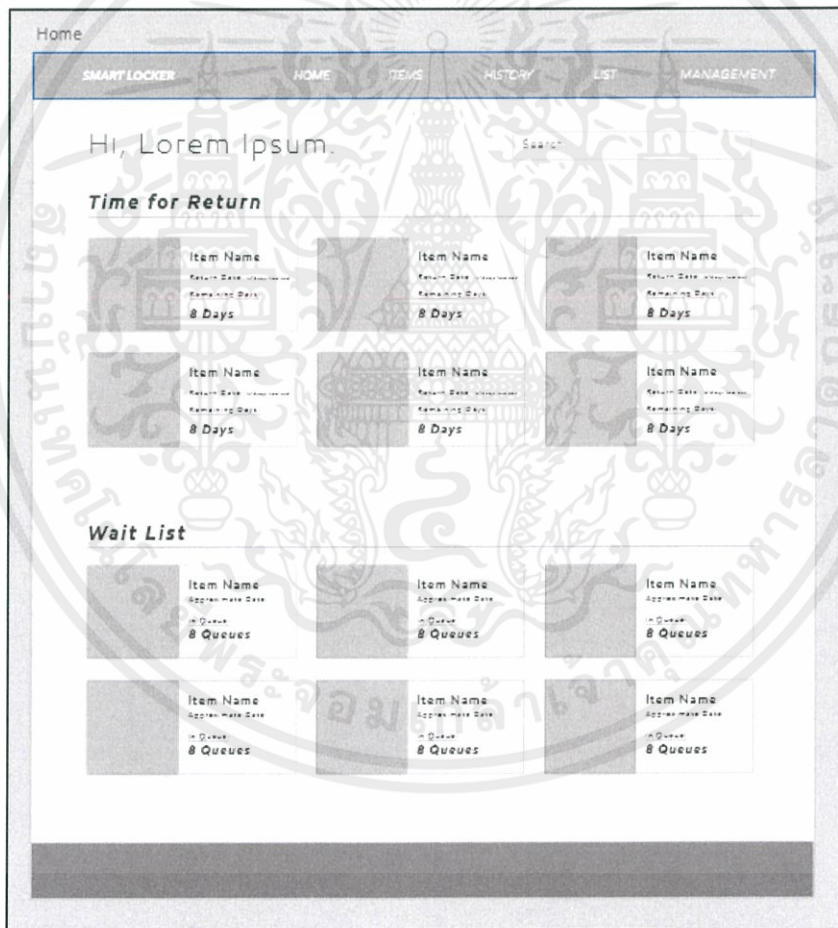
เมื่อได้ทำการรวบรวมข้อมูลความต้องการทั้งหมดที่ผู้ใช้ต้องการให้มีในระบบ จากนั้นจะนำทั้งหมดมาจัดเรียงความสำคัญ จากนั้นจะเข้าสู่ขั้นตอนของการออกแบบ

3.2 การออกแบบ และทดลองใช้งาน

จากการเก็บข้อมูลความต้องการทั้งหมด สรุปได้ว่าจะต้องมีหน้าทั้งหมด 3 ส่วน ได้แก่ ส่วนที่
ผู้ใช้งานทั่วไปใช้งาน ส่วนเสริมของเจ้าของอุปกรณ์ และส่วนเสริมของผู้ดูแลระบบ โดยผู้ใช้งานทั่วไป จะไม่ได้รับ
สิทธิ์ให้เข้าไปในหน้าส่วนเสริมของเจ้าของอุปกรณ์หรือว่าของผู้ดูแลระบบ จึงทำให้หน้าของการออกแบบมี
ทั้งหมดดังนี้

3.2.1 หน้าสำหรับผู้ใช้งานทั่วไป

ในการออกแบบช่วงแรก มีการกำหนดว่าจะมีหน้าสำหรับผู้ใช้งาน ทั้งหมด 3 หน้า ได้แก่ หน้า
Dashboard, หน้า Browse Item และ หน้า History ดังรูปที่ 3.1 รูปที่ 3.2 และ รูปที่ 3.3 ตามลำดับ



รูปที่ 3.1 หน้า Dashboard ที่ได้ออกแบบไว้

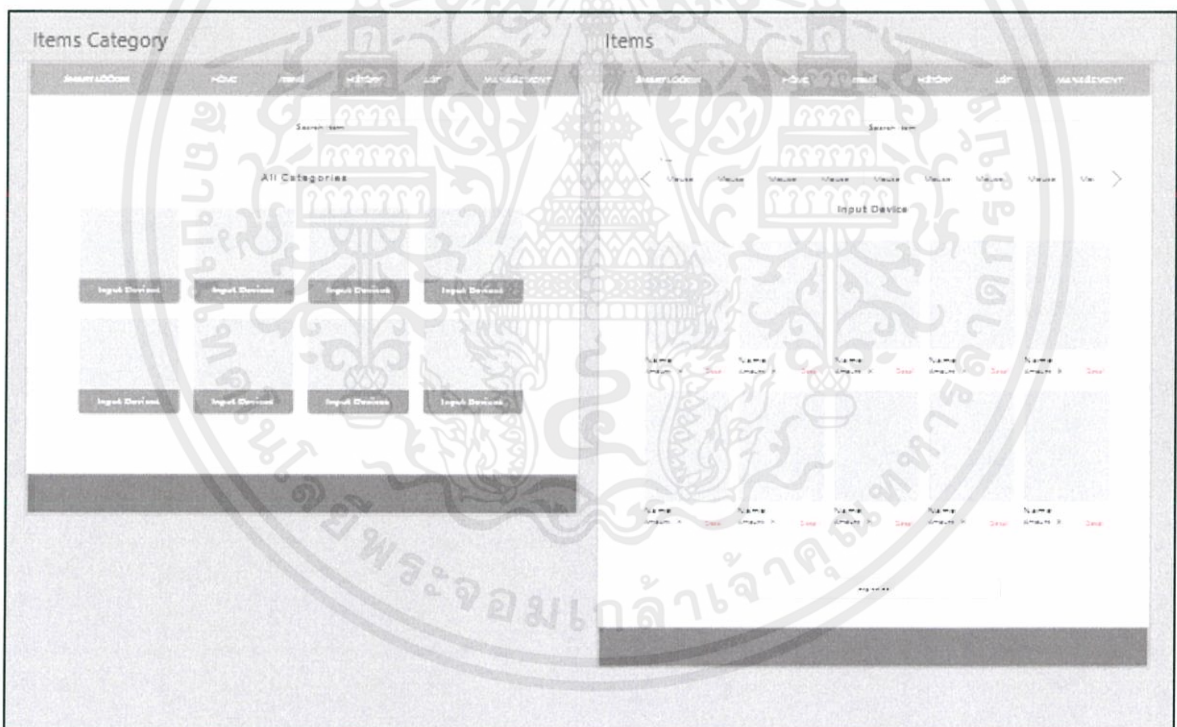
ส่วนหน้า Dashboard จะมีการแสดงข้อมูลในส่วนของผู้ใช้งานโดยสังเขป และมีการแบ่งข้อมูลภายในออกเป็นสองส่วน ได้แก่

1. My Loaned Item

เป็นส่วนที่จะแสดงข้อมูลของอุปกรณ์ที่ผู้ใช้งานกำลังยืมอยู่ในขณะนี้ พร้อมจำนวนวันที่เหลือก่อนจะต้องคืนอุปกรณ์ ซึ่งจะสามารถกดเข้าไปดูข้อมูลเพิ่มเติมของอุปกรณ์ชิ้นนี้ได้อีกด้วย

2. My Watch List

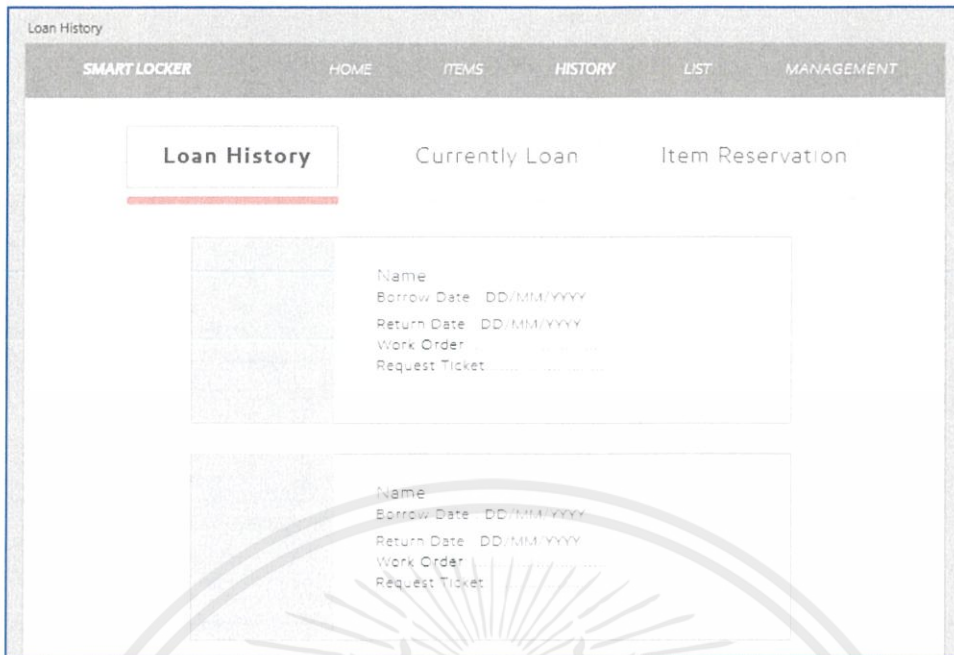
เป็นส่วนที่จะแสดงข้อมูลอุปกรณ์ที่ผู้ใช้งานได้มีการกดติดตามอุปกรณ์ชิ้นนั้นเอาไว้เพราะมีความต้องการที่จะยืม แต่ในปัจจุบันอุปกรณ์ชิ้นนี้มีจำนวนให้ไม่เพียงพอต่อการยืม อีกทั้งยังสามารถบอกได้ว่าเหลือจำนวนคิวที่ต้องรอเป็นจำนวนเท่าไร



รูปที่ 3.2 หน้า Browse Item ที่ได้ออกแบบไว้

ในส่วนของหน้า Browse Item นั้น เนื่องจากต้องการให้มีความสะดวก และง่ายต่อการค้นหา มากขึ้น จึงมีการจัดหมวดหมู่ของอุปกรณ์ ให้ผู้ใช้ทำการเลือกหมวดหมู่ของอุปกรณ์ ก่อนที่จะเข้าไปยังส่วน ของข้อมูลที่อยู่ในหมวดนั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อ 20 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 หน้า History ที่ได้ออกแบบไว้

ภายในหน้า History จะมีลักษณะคล้ายกับ Dashboard แต่การแสดงผลประวัติของอุปกรณ์จะแสดงข้อมูลครบทั้งหมด ต่างจาก Dashboard ที่จะแสดงข้อมูลเพียงสังเขปเท่านั้น ภายในหน้านี้ จะสามารถแบ่งออกเป็นสามส่วนใหญ่ ๆ ได้แก่

1. Loan History

เป็นส่วนที่จะแสดงอุปกรณ์ที่ผู้ใช้ระบบเคยยืมไปก่อนหน้านี้ โดยจะมีการระบุวันที่ที่ยืม และวันที่ที่ยืม อีกทั้งยังมีการระบุเลข Work Order และ Request Number อีกด้วย

2. Currently Loan

ในส่วนนี้จะคล้ายกับส่วน My Loaned Item ในหน้า Dashboard แต่ภายในหน้านี้จะมีการบอกรายละเอียดเฉพาะอย่างชัดเจน เช่น ยืมอุปกรณ์มาและต้องคืนอุปกรณ์ในวันที่เท่าไร และจะมีปุ่ม Report Issue ในกรณีที่อุปกรณ์ที่ยืมมาชำรุด หรือเกิดความเสียหายขึ้น

3. Reservation List

ในส่วนนี้จะคล้ายกับส่วน My Watch List ในหน้า Dashboard ที่จะบอกว่าผู้ใช้งานได้กดติดตามอุปกรณ์ชิ้นไหนไว้บ้าง แต่ภายในหน้านี้จะมีปุ่มยกเลิกการติดตาม ในกรณีที่ไม่ต้องการที่จะยืมอุปกรณ์ชิ้นนี้แล้ว

3.2.2 หน้าสำหรับผู้ใช้งานที่เป็นเจ้าของอุปกรณ์ที่ใหืม

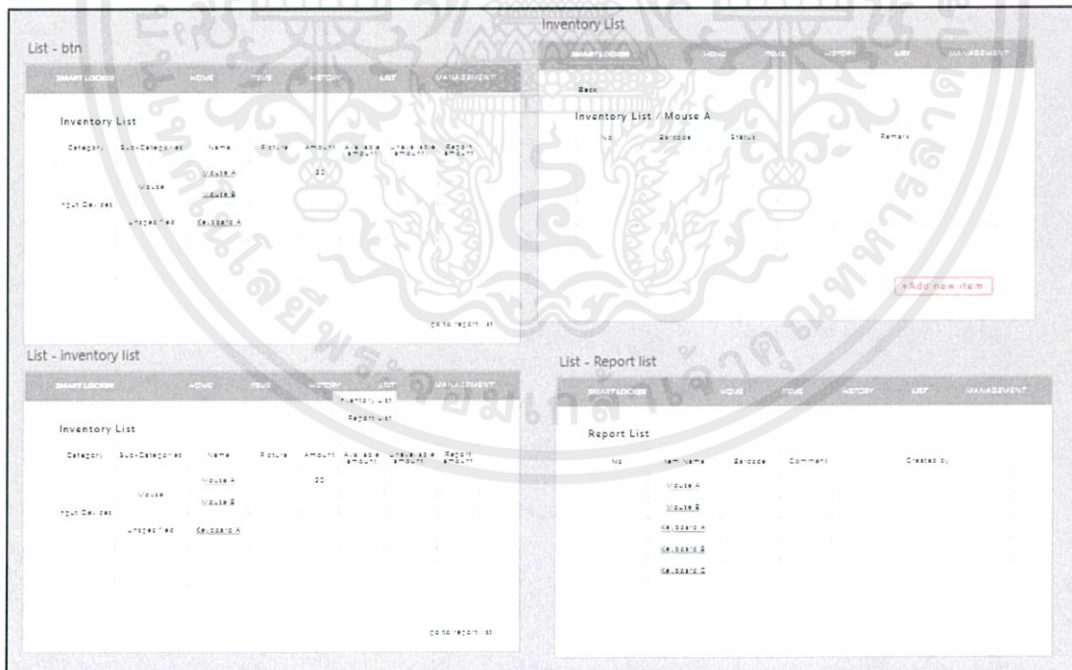
ตามความต้องการของเจ้าของอุปกรณ์ที่ได้รับรองมา ภายในหน้านี้จะมีการแสดง Report ของอุปกรณ์ที่ทำการรับผิดชอบอยู่ ดังรูปที่ 3.4 โดยจะสามารถแบ่งออกเป็นสองส่วน ได้แก่

1. Inventory List

จะเป็นหน้า Report ที่บอกอุปกรณ์ทั้งหมดที่อยู่ภายใต้การดูแลของบุคคลนั้น โดยจะมีการบอก Category Sub-category ชื่ออุปกรณ์ จำนวนของอุปกรณ์ จำนวนที่ถูกยืม จำนวนคงเหลือ และจำนวนที่มีการรายงานมาว่าเกิดความเสียหาย หรือชำรุด โดยสามารถกดเข้าไปดูภายในอุปกรณ์ชิ้นนั้น ๆ เพื่อดูรายงานของอุปกรณ์ชิ้นนั้นทุกชิ้น

2. Report List

หน้านี้เป็นหน้าเฉพาะสำหรับของที่ถูกแจ้งเรื่องความเสียหาย หรือชำรุด จากส่วนของ Report Issue ทางฝั่งผู้ใช้งาน โดยจะระบุว่าเป็นอุปกรณ์ชื่อใด หมายเลขอุปกรณ์อะไร ชำรุดหรือเสียหาย ด้านไหน และถูกรายงานโดยใคร



รูปที่ 3.4 ส่วนของ Owner ที่ได้ออกแบบไว้

3.2.3 หน้าสำหรับผู้ใช้งานที่เป็นผู้ดูแลระบบ

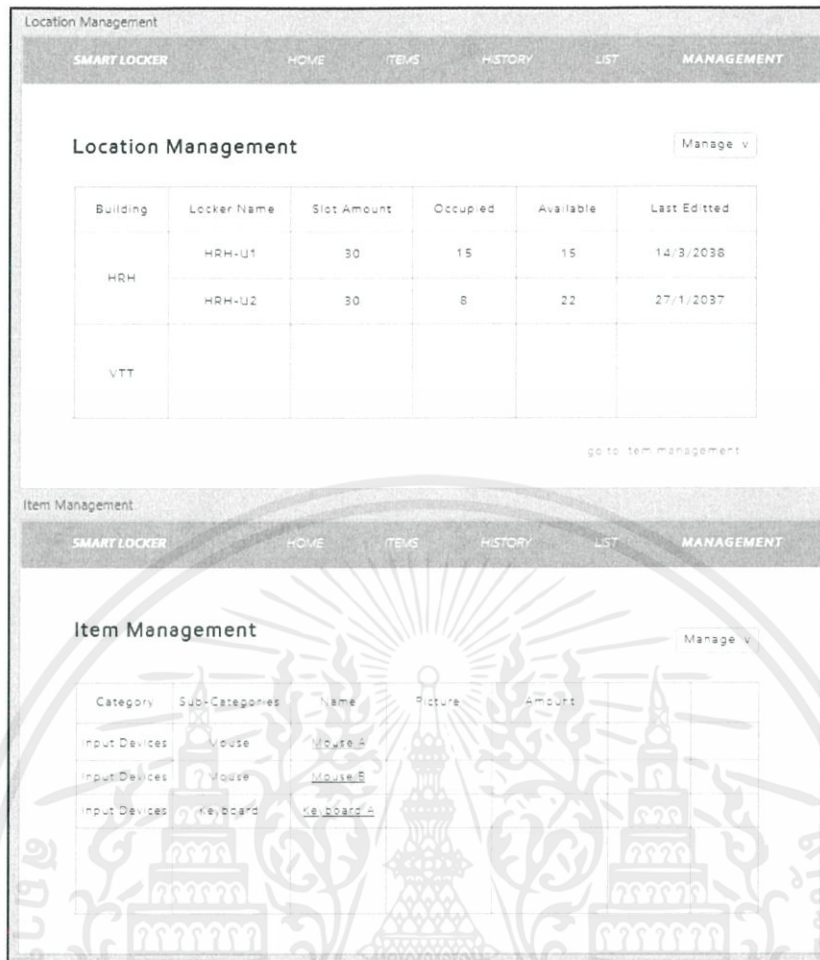
เนื่องจากผู้ดูแลระบบจะมีสิทธิ์เข้าถึงข้อมูลทั้งหมดของอุปกรณ์ที่สามารถยืมได้ รวมถึงสามารถทำการ เพิ่ม ลด และแก้ไขอุปกรณ์ชิ้นนั้น ๆ ได้ หรือจะทำการเพิ่ม ลด และแก้ไข จำนวนของสถานที่ให้บริการ หรือจำนวนของตู้ล็อกเกอร์ ก็จะสามารถทำได้เช่นกัน ดังรูปที่ 3.5 จึงทำให้ในส่วนของผู้ดูแลระบบจะมีการจัดการแบ่งออกเป็นสองส่วน ดังนี้

1. Location Management

ภายในหน้านี้ จะเป็นการบอกชื่อของอาคารที่รองรับระบบนี้ และจำนวนของตู้ล็อกเกอร์ในแต่ละอาคาร รวมถึงรายละเอียดของตู้ล็อกเกอร์นั้น ว่ามีจำนวนช่องกี่ช่อง วางทั้งหมดกี่ช่อง และมีการแก้ไขข้อมูลส่วนนี้ครั้งสุดท้ายเมื่อไหร่ โดยภายในหน้านี้ผู้ดูแลระบบ ก็จะสามารถเพิ่ม ลด แก้ไขทั้งตัวของอาคารที่รองรับระบบและตัวตู้ล็อกเกอร์ได้

2. Item Management

หน้านี้จะเป็นหน้าที่จะบอกรายละเอียดของอุปกรณ์ที่มีให้ยืมทั้งหมดในปัจจุบัน โดยจะแสดง Category Sub-category ชื่ออุปกรณ์ และจำนวนที่มีอยู่ ซึ่งก็ยังสามารถเข้าไปดูในแต่ละอุปกรณ์ ก็จะเป็นจำนวนของอุปกรณ์ชิ้นนั้นทั้งหมดที่มีอยู่ในระบบอีกด้วย ซึ่งก็เหมือนกันกับด้าน Location Management ที่ ผู้ดูแลระบบจะสามารถเพิ่ม ลด แก้ไขอุปกรณ์ชิ้นนั้นได้เช่นเดียวกัน



รูปที่ 3.5 หน้า Admin ที่ได้ออกแบบไว้

หลังจากการออกแบบหน้าทั้งหมดเสร็จ ทางผู้วิจัยก็นำตัว Prototype ไปเสนอกับผู้ใช้งานทั้งสามฝ่าย และทำการเก็บข้อมูลการใช้งานอีกครั้งเพื่อนำไปแก้ไข ก่อนจะนำไปพัฒนาเป็นตัวเว็บไซต์จริง

ซึ่งเมื่อนำการออกแบบรุ่นแรกมาทดลองใช้ จึงได้รับความคิดเห็นมาว่า ในส่วนของผู้ใช้งานระบบทั่วไป ไม่มีความจำเป็นที่จะต้อง มีหน้า History เพื่อแสดงประวัติของการยืมก่อนหน้านี้ เพราะไม่มีความจำเป็นที่จะต้องทราบมาก่อนหน้านี้เคยยืมอุปกรณ์อะไรไปบ้าง และในส่วนของ การติดตามอุปกรณ์ที่ต้องการยืม จะตัดส่วนของจำนวนคิวที่ต้องรอก่อนที่จะยืมอุปกรณ์ได้ออก เนื่องจากอุปกรณ์ที่มีให้ยืมนั้น มีอุปกรณ์ที่เป็นชนิดเดียวกันให้ยืมเป็นจำนวนหลายชิ้น ดังนั้นจึงจะไม่สามารถกำหนดได้ว่า แต่ละบุคคลที่ทำการยืมไป จะนำอุปกรณ์มาคืนภายในวันไหน และอาจจะทำให้เกิดความสับสนในอนาคต จึงได้มีการนำออก จากหน้าผู้ใช้ทั่วไปก่อน ส่วนของ Owner ต้องการรายละเอียดของการทำรายงานเพิ่มเติม ว่าอาจจะทำให้สามารถ Export ออกมาเป็นไฟล์ หรือส่งรายงานของแต่ละเดือนผ่านอีเมลอัตโนมัติ

หลังจากได้รับความคิดเห็นมาแล้ว เมื่อทำการประชุมกัน จึงมีผลมติให้หน้าของผู้ใช้งานระบบทั่วไป เหลือแค่ 2 หน้าหลัก ได้แก่ Dashboard และ Browse Item ซึ่งจากการสรุปผลการประชุมแล้ว จะทำให้ เริ่มเข้าสู่การพัฒนาเว็บไซต์ของส่วนผู้ใช้หลักเป็นส่วนสำคัญอันดับแรก

เนื่องจากหน้าต่าง ๆ ที่ได้ออกแบบไว้เป็นเพียงแค่ตัว Prototype และ Wireframe ทำให้พนักงานร่วมทีม รับหน้าที่ในส่วนของการดีไซน์ตัวหน้าเว็บไซต์ให้เข้ากับบริษัท เนื่องจากทางบริษัทจะมีการ กำหนดโทนสีของแต่ละแผนก ลักษณะการวางข้อมูลต่าง ๆ ภายในหน้าเว็บไซต์ให้ออกมาในทางเดียวกัน ซึ่ง โทนสีของทางฝ่าย EMIT จะเป็นสีม่วงและสีฟ้า เมื่อได้รับดีไซน์หน้าเว็บไซต์ที่เรียบร้อยแล้ว ก็จะเข้าสู่ ขั้นตอนของการพัฒนาเว็บไซต์จริงในหัวข้อถัดไป

3.3 การพัฒนาเว็บไซต์จริง

ผู้วิจัยได้เริ่มพัฒนาเว็บไซต์ในส่วนของผู้ใช้งานทั่วไปเป็นจำนวน 2 หน้าใหญ่ ซึ่งจะประกอบไปด้วย หน้า Dashboard ดังรูปที่ 3.12 และ หน้า Browse Item ดังรูปที่ 3.24 ตามลำดับ โดยที่ภายในหน้าหลักนี้ จะมีการทำงานย่อยเบื้องหลังภายในเว็บไซต์ แบ่งได้เป็น 8 ฟังก์ชันหลัก ซึ่งจะจำแนกแต่ละฟังก์ชันออกเป็น 3 ส่วน ได้แก่ ฟังก์ชันที่ทำงานทุกหน้า ฟังก์ชันในหน้า Dashboard และฟังก์ชันในหน้า Browse Item

3.3.1 ฟังก์ชันที่ทำงานในทุกหน้า

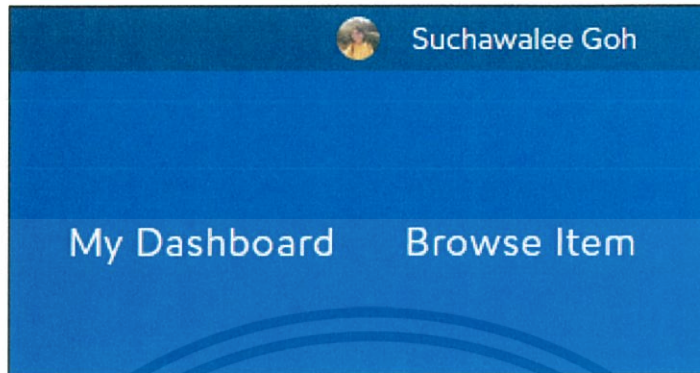
เนื่องจากเว็บไซต์จำเป็นต้องทราบรายละเอียดส่วนบุคคล เพื่อที่จะดึงข้อมูลการยืนยันของ บุคคลนั้น จึงทำให้เว็บไซต์นี้จะต้องทราบโดยอัตโนมัติว่าผู้ใช้งานในขณะนี้คือใคร โดยส่วนที่จะต้องแสดง ข้อมูลของผู้ใช้มีอยู่ 2 ส่วน ได้แก่ รูปและชื่อ บน Navigation Bar ดังรูปที่ 3.6 และส่วนการโหลดข้อมูลของ หน้า Browse Item เนื่องจากการจัดเก็บข้อมูลในฐานะข้อมูลเป็นการเก็บอุปกรณ์โดยขึ้นอยู่กับสถานที่ที่ผู้ ใช้ได้เลือกไว้

ซึ่งข้อมูลของผู้ใช้งานจะถูกเรียกมาจาก API ทั้งหมด 3 ตัว ได้แก่

- Badge ID API ที่จะได้เลขบนบัตรประจำตัวของพนักงาน
- Ticket API ที่จะได้ข้อมูลทั้งหมดของพนักงานที่ใช้คอมพิวเตอร์ของบริษัทเครื่องนั้น
- User API เป็น API ของทาง Smart Locker เพื่อที่จะเชื่อมข้อมูลระหว่างเว็บไซต์และตู้ล็อกเกอร์อัจฉริยะ เนื่องจากระบบการยืนยันผ่านตู้ล็อกเกอร์ จะต้องใช้บัตรพนักงานในการยืนยันตัวตนจึงจำเป็นต้องเก็บทั้ง เลขบัตรประจำตัวพนักงาน ตำแหน่งที่ตั้ง และ Lan ID ที่เป็น ID ระบุตัวตนของพนักงานคน นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อ 25 รัชกาลเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ข้อมูลบุคคล ส่วนหนึ่งก็จะทำหน้าที่นำมาแสดงผลบน Navigation Bar ดังรูปที่ 3.6



รูปที่ 3.6 รูปและชื่อของ User ที่แสดงบน Navigation Bar

และในอีกส่วนหนึ่งก็จะนำเอามาตรวจสอบว่าบุคคลนี้ได้มีการลงทะเบียนกับระบบ Smart Locker หรือไม่ ซึ่งการตรวจสอบข้อมูลในส่วนนี้ จะถูกตรวจสอบเมื่อทำการเรียก User API มา ถ้าหากพบว่าผู้ใช้ในระบบแล้ว ก็จะไม่เกิดอะไรขึ้น แต่หากไม่พบผู้ใช้ในระบบ จะมีการสั่งเปิด Select Location Modal ดังรูปที่ 3.7 เพื่อที่จะเก็บข้อมูลสถานที่ของผู้ใช้

รูปที่ 3.7 Select Location Modal

หลังจากเปิด Select Location Modal แล้ว จะมีการตรวจสอบว่ามีผู้ใช้งานนี้ในระบบหรือไม่อีกครั้ง ดังรูปที่ 3.8 ถ้าไม่มี ผู้ใช้งานจำเป็นจะต้องเลือกประเทศก่อน จากนั้นระบบจะทำการกรองข้อมูลของอาคารสำนักงานเพื่อให้แสดงเฉพาะอาคารที่อยู่ในประเทศที่เลือกก่อน จากนั้นผู้ใช้งานจึงจะสามารถเลือกอาคารสำนักงานได้ โดยที่ไม่สามารถปิด Modal ได้จนกว่าจะ Submit ข้อมูล ตามคำสั่งดังรูปที่ 3.9 แต่ถ้าหากพบผู้ใช้งานนี้ในระบบ แล้วจะกลายเป็นคำสั่งเพื่ออัปเดตสถานที่ของผู้ใช้งาน ตามคำสั่งรูปที่ 3.10

```
checkUserExist() {
  var userSite = this.all.find(element => {
    return element.strLocationName === this.userSite
  }).intLocationId

  if (this.userData.userSite == -1) {
    this.postUserDetail()
    this.hide()
  } else {
    this.putUserSite()
    this.hide()
  }
},
```

รูปที่ 3.8 คำสั่งตรวจสอบว่ามีผู้ใช้งานในระบบหรือไม่

```
postUserDetail() {
  var userSite = this.all.find(element => {
    return element.strLocationName === this.userSite
  }).intLocationId

  this.$axios.post('http://localhost:3000/api/user/create', {
    BadgeId: this.userData.userBadgeID,
    Description: this.userData.userLanID,
    SiteId: userSite
  }).then(result => {
    if (result.data.StatusDescription == "SUCCESS") {
      this.$store.dispatch('user/setSite', userSite)
    } else {
      console.log("POST FAILED")
    }
  })
},
```

รูปที่ 3.9 คำสั่งสร้างผู้ใช้งานนี้ เมื่อไม่พบผู้ใช้งานในระบบ

```

putUserSite() {
  console.log("put user detail")
  var userSite = this.all.find(element => {
    return element.strLocationName === this.userSite
  }).intLocationId

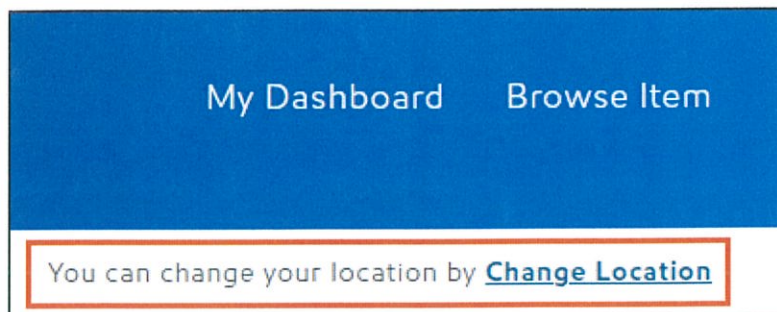
  var userSiteName = this.all.find(element => {
    return element.strLocationName === this.userSite
  }).strLocationName

  this.$axios.put('██████████/api/user/update', {
    UserId: this.userID,
    UserBadgeId: this.userData.userBadgeID,
    UserRole: this.userData.userRole,
    Description: this.userData.userLanID,
    SiteId: userSite,
    UpdaterId: this.userID,
    UpdaterBadgeId: this.userData.userBadgeID
  }).then(result => {
    // console.log("Put Lan ID to DB", result)
    if (result.data.StatusDescription == "SUCCESS") {
      this.$store.dispatch('user/setSite', userSite)
      // console.log("userdata", this.userData)
    } else {
      console.log("PUT FAILED")
      this.loadNewUserData()
    }
  }).catch(error => {
    console.log("error", error)
    this.loadNewUserData()
  })
},

```

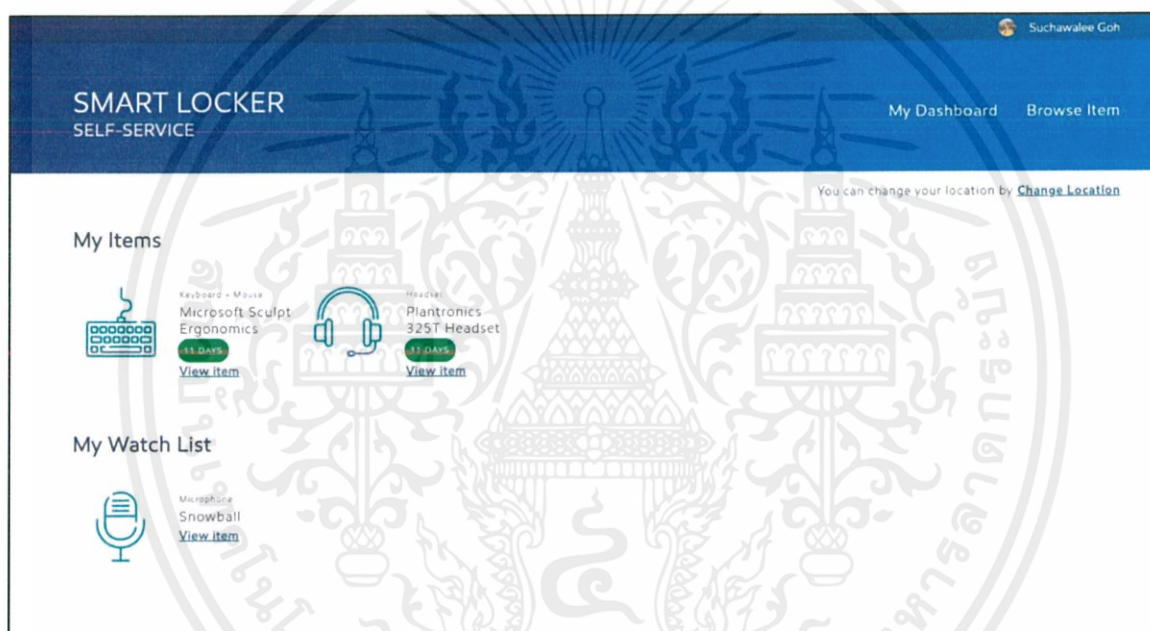
รูปที่ 3.10 คำสั่งอัปเดตผู้ใช้งานนี้ เมื่อพบผู้ใช้งานในระบบ

สาเหตุที่จะต้องตรวจสอบว่ามีผู้ใช้งานนี้ในระบบอีกรอบหรือไม่ เนื่องจาก Modal นี้ มีไว้สำหรับในกรณีที่ต้องการเปลี่ยนสถานที่ของตนเอง ซึ่งมีหลายกรณีที่พนักงานจะต้องย้ายไปทำงานที่อาคารอื่น จึงจำเป็นต้องให้มีการย้ายสถานที่ไปมาได้ โดยผู้ใช้งานสามารถเปิด Modal เมื่อต้องการย้ายสถานที่ได้จากตำแหน่งใต้ Navigation Bar ดังรูปที่ 3.11



รูปที่ 3.11 การเปิด Select Location Modal อีกรูปแบบหนึ่ง

3.3.2 ฟังก์ชันในหน้า Dashboard



รูปที่ 3.12 หน้า Dashboard บนเว็บไซต์จริง

ภายในหน้า Dashboard จะมีฟังก์ชันการทำงานอยู่ 2 ส่วนใหญ่ ได้แก่ ในส่วน My Items และ My Watch List ซึ่งภายในสองส่วนนี้ ก็จะมีฟังก์ชันการทำงานในแต่ละส่วนแตกต่างกันไป จึงจะจำแนกฟังก์ชันการทำงานในแต่ละส่วน ดังนี้

1. My Items

ฟังก์ชันการทำงานแรกสุด คือการเรียกข้อมูลจาก API เพื่อจะแสดงข้อมูลของอุปกรณ์ที่กำลังยืมอยู่ในขณะนี้ และเนื่องจากใน Vue.js มีฟังก์ชันการทำงานที่สามารถแสดงกลุ่มของข้อมูลโดยเขียน

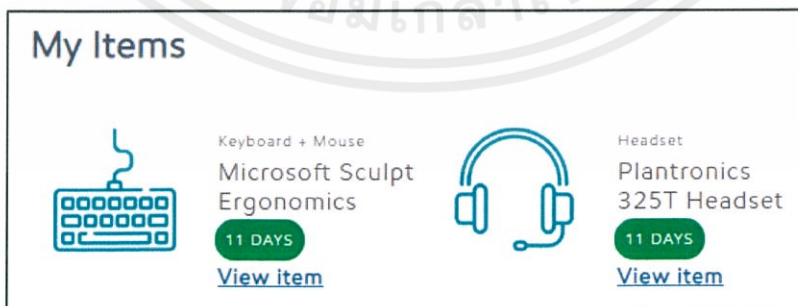
โค้ดเพียงชุดเดียว ดังรูปที่ 3.13 จึงทำให้ลดระยะเวลาการทำงานที่ซ้ำซ้อนลงได้ และจะได้ผลลัพธ์ออกมาดังรูปที่ 3.14 โดยจะมีการบอก Sub-category ชื่อของอุปกรณ์ และจำนวนวันที่เหลืออยู่ ซึ่งภายในรูปที่ 3.13 จะมีการเขียนเงื่อนไขในการแสดงจำนวนวันที่เหลืออยู่ ถ้าเหลือมากกว่าหรือเท่ากับ 10 วัน จะให้แสดงพื้นหลังสีเขียว ถ้าเหลือน้อยกว่า 10 วัน แต่มากกว่าหรือเท่ากับ 5 วัน จะแสดงพื้นหลังสีเหลือง แต่ถ้าเหลือน้อยกว่า 5 วัน แต่มากกว่าหรือเท่ากับ 2 วัน พื้นหลังจะเปลี่ยนเป็นสีแดงโดยอัตโนมัติ

```

<template>
  <div class="box-component">
    <div class="item-list" v-for="item in loanedItem" :key="item.name">
      <div class="item-pic">
        
      </div>
      <div class="item-detail">
        <div class="item-subcat">
          <p>{{item.strSubCategoryName}}</p>
        </div>
        <div class="item-name">
          <p>{{item.strItemName}}</p>
        </div>
        <div id class="item-remainingDays remain-green" v-if="item.remainingDays >= 10 ">
          <p>{{item.remainingDays}} DAYS</p>
        </div>
        <div id class="item-remainingDays remain-yellow" v-else-if="item.remainingDays >= 5 ">
          <p>{{item.remainingDays}} DAYS</p>
        </div>
        <div id class="item-remainingDays remain-red" v-else-if="item.remainingDays >= 2 ">
          <p>{{item.remainingDays}} DAYS</p>
        </div>
        <div id class="item-remainingDays remain-red" v-else-if="item.remainingDays <= 1 ">
          <p>{{item.remainingDays}} DAY</p>
        </div>
        <div class="item-viewMore">
          <router-link :to="{name: 'itemDetail-from-myLoanItem', params: {loanID: item.intLoanId }}">View item</router-link>
        </div>
      </div>
    </div>
  </div>
</template>

```

รูปที่ 3.13 การเขียน Template เพื่อแสดงข้อมูล



รูปที่ 3.14 ส่วนของ My Items

ในส่วนของฟังก์ชันการเรียกข้อมูลจาก API นั้น จำเป็นจะต้องใช้ข้อมูลของ Badge ID หรือเลขบัตรประจำตัวของพนักงาน ที่เคยได้กล่าวไว้ในหัวข้อ 3.3.1 เพื่อที่จะเรียกข้อมูลการยืมของผู้ใช้งาน รายนี้ ดังรูปที่ 3.15 ซึ่งเมื่อได้รับข้อมูลมาแล้ว จะมีการนำมาคำนวณเพื่อหาจำนวนวันที่เหลือในการยืมเพื่อนำไปแสดงในรูปที่ 3.14 อีกด้วย

```
this.$axios.get('.../api/object/onloan?userid=' + this.userID)
  .then((result) => {
    var loanedItem = result.data.Message

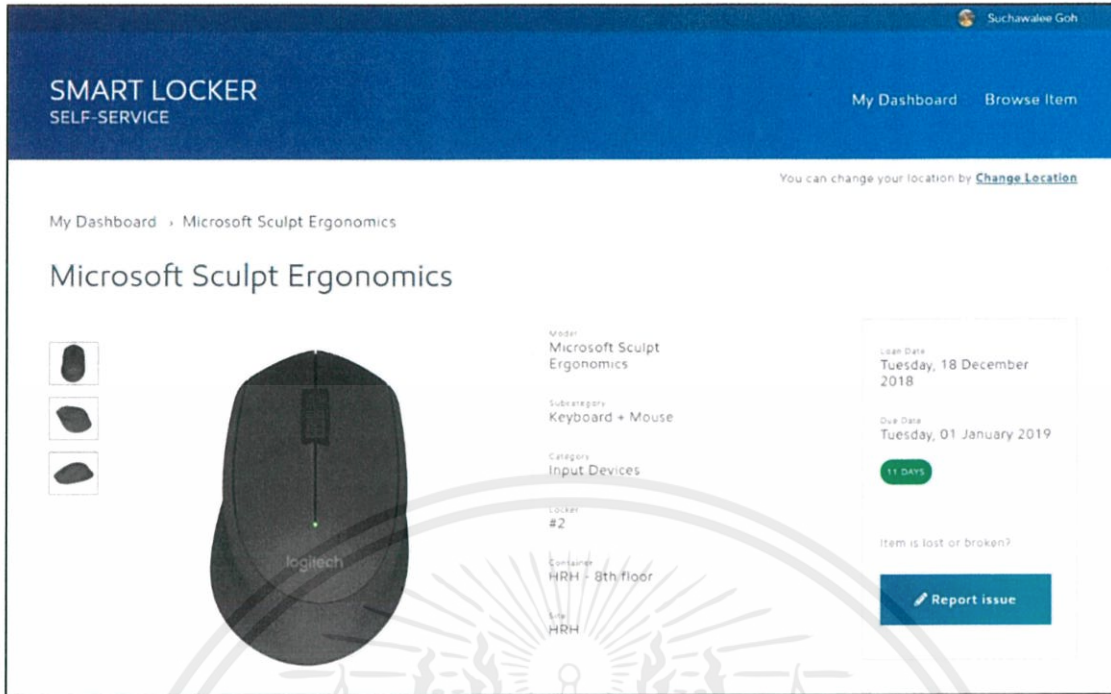
    loanedItem.forEach(element => {
      var loanID = element.intLoanId
      this.$store.dispatch('myItems/setLoanID', loanID)

      var loanDate = this.$moment(element.datLoanDate)
      var dueDate = this.$moment(element.datDueDate)
      var today = this.$moment()
      var remainingDays = dueDate.diff(today, 'days')

      element.remainingDays = remainingDays
      this.loanedItem.push(element)
      this.$store.dispatch('myItems/setMyLoanedItem', {
        itemName: element.strItemName,
        remainingDays: remainingDays
      })
    })
  })
```

รูปที่ 3.15 คำสั่งเรียกดูข้อมูลการยืมของผู้ใช้งาน

เมื่อผู้ใช้งานต้องการดูข้อมูลของอุปกรณ์ชิ้นนี้เพิ่มเติม ผู้ใช้งานสามารถกดที่คำว่า 'View Item' ดังรูปที่ 3.14 จากนั้นเว็บไซต์จะทำการเปลี่ยนหน้าไปเป็นหน้าข้อมูลของอุปกรณ์ ดังรูปที่ 3.16



รูปที่ 3.16 หน้าข้อมูลของอุปกรณ์ที่กำลังยืมอยู่

โดยภายในหน้านี้จะมีการบอกข้อมูลของอุปกรณ์อย่างละเอียด จนรวมถึงว่านำอุปกรณ์มาจากล็อกเกอร์ใด และช่องอะไร และทางด้านขวาจะมีบอกวันที่ที่ไปยืมของ และวันที่ที่จะต้องนำของไปคืน นอกจากนี้ยังมีปุ่ม Report Issue ในกรณีที่อุปกรณ์ชิ้นนี้มีความเสียหาย หรือชำรุด ซึ่งเมื่อกดเข้าไปจะปรากฏหน้าต่าง Modal ขึ้นมา ดังรูปที่ 3.17 โดยภายใน Modal นี้จะมีการระบุชื่อผู้รายงานโดยอัตโนมัติ มีตัวเลือกกว่าอุปกรณ์ชำรุด หรือสูญหายพร้อมกับสามารถกรอกข้อมูลเพิ่มเติมได้

รูปที่ 3.17 Report Issue Modal

2. Watch List

ฟังก์ชันในส่วนของ Watch List จะมีความใกล้เคียงกับส่วน My Items เนื่องจากต้องมีการดึงข้อมูลมาจาก API โดยจำเป็นต้องใช้เลขประจำตัวบัตรพนักงานเช่นเดียวกัน ดังรูปที่ 3.18 แต่ข้อมูลจะมีข้อมูลใน Watch List เกิดจากการกดติดตามอุปกรณ์ (Notify Me) ในกรณีที่อุปกรณ์ชนิดนั้นมีไม่เพียงพอต่อการยืม ซึ่งคำสั่งการกดติดตามอุปกรณ์ จะอยู่ในหัวข้อที่ 3.3.3

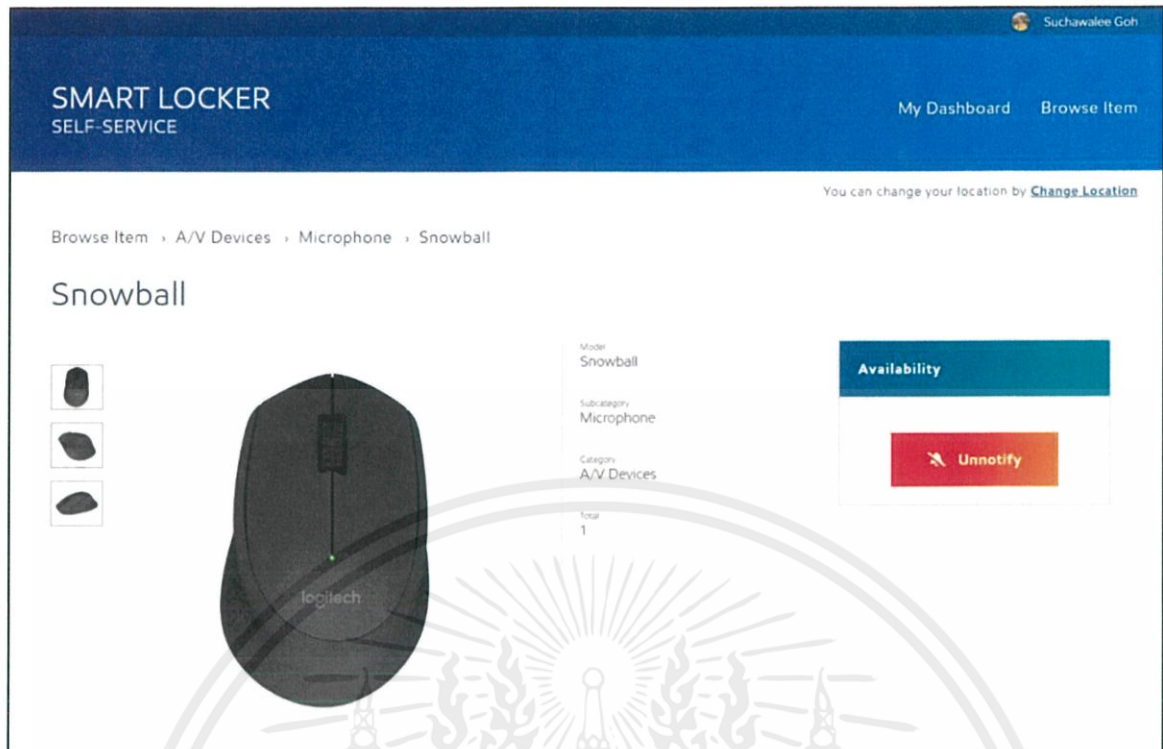
```

this.$axios.get('.../api/item/watchlist?userid=' + this.userID)
  .then((result) => {
    var watchList = result.data.Message
  })

```

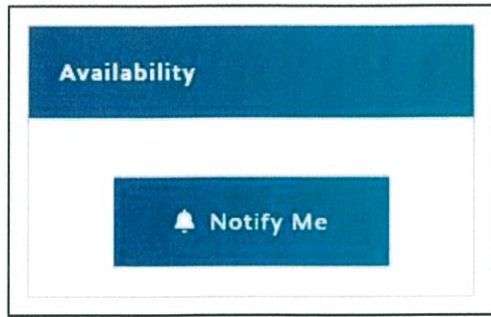
รูปที่ 3.18 คำสั่งเรียกข้อมูลของ Watch List

โดยเมื่อกด 'View Item' ของอุปกรณ์ที่อยู่ใน Watch List เว็บไซต์จะทำการเปลี่ยนหน้าไปเป็นหน้าที่แสดงข้อมูลของอุปกรณ์นั้น ดังรูปที่ 3.19

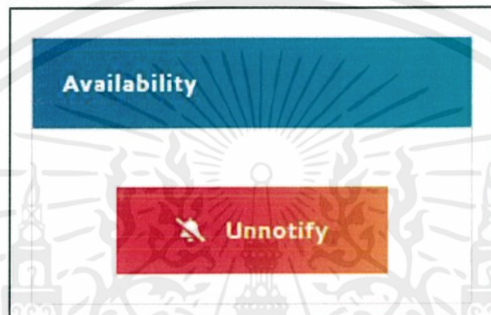


รูปที่ 3.19 หน้าข้อมูลของอุปกรณ์ที่ติดตามอยู่

เมื่อเข้ามายังหน้าข้อมูลของอุปกรณ์ที่ติดตามอยู่ จะพบว่าหน้านี้มีลักษณะแบบเดียวกับหน้าข้อมูลของอุปกรณ์ที่สามารถยืมได้ จากหน้า Browse Item แต่จะแตกต่างตรงที่ส่วนทางด้านขวาของหน้า จะเป็นรูปปุ่มแทน เป็นการแสดงว่าอุปกรณ์ชิ้นนี้ มีจำนวนไม่เพียงพอต่อการยืม และหากว่าปุ่มเป็นสีฟ้า พร้อมแสดงคำว่า Notify Me ดังรูปที่ 3.20 หมายถึงยังไม่ได้ทำการกดติดตามอุปกรณ์ชิ้นนี้ แต่ถ้าปุ่มเป็นสีส้ม พร้อมแสดงคำว่า Unnotify ดังรูปที่ 3.21 แปลว่าผู้ใช้งานได้กดติดตามอุปกรณ์ชิ้นนี้เรียบร้อยแล้ว และสถานะของปุ่มนี้ภายในหน้า Watch List กับหน้าข้อมูลของภายใน Browse Item จะเชื่อมกัน เมื่อมีการเปลี่ยนแปลงในหน้าใดหน้าหนึ่ง อีกหน้าก็จะเปลี่ยนไปด้วยเช่นกัน



รูปที่ 3.20 ปุ่ม Notify Me



รูปที่ 3.21 ปุ่ม Unnotify

โดยการที่ปุ่มจะแสดงเป็นสีใดนั้น จะขึ้นอยู่กับตอนเรียกข้อมูลมาจาก API เนื่องจากใน API จะมีการส่งสถานะของอุปกรณ์ว่าอุปกรณ์นี้ได้ถูกติดตามอยู่หรือไม่ ดังรูปที่ 3.22

```
loadItemDetail: function (siteID, userID) {
  this.$axios.get('https://em-smartlocker.azurewebsites.net/api/item/details?siteid=' + this.siteID +
    '&userid=' + this.userID + '&itemid=' + this.$route.params.itemID)
    .then(response => {
      this.message = response.data.Message
      this.createDescription()
      this.messageStatus = true

      var notifyStatus = response.data.Message.IsNotifyThisSite
      this.notifyStatus = notifyStatus
    })
},
```

รูปที่ 3.22 คำสั่งเรียกข้อมูลของอุปกรณ์พร้อมกับสถานะการติดตามอุปกรณ์

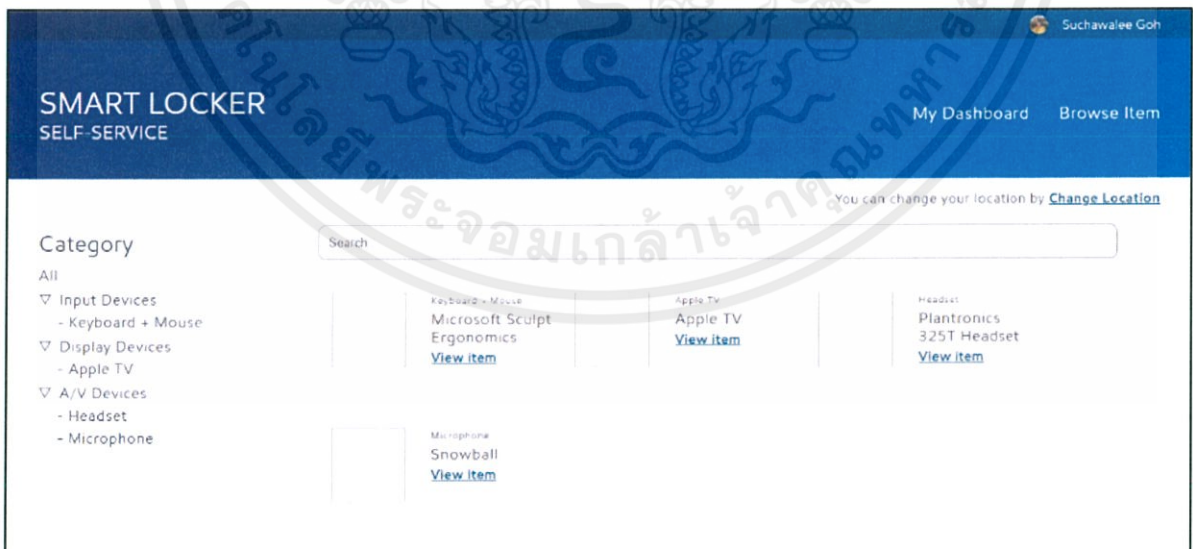
ค่าตัวแปร notifyStatus จะมีค่าเป็น true เมื่อกดติดตาม หรือ false เมื่อไม่ได้กดติดตาม และจะถูกใช้เป็นเงื่อนไขในการแสดงปุ่มบนหน้าเว็บ ว่าเมื่อตัวแปรนี้มีค่าเป็น false จะให้แสดงปุ่ม Notify Me แต่ถ้าตัวแปรนี้มีค่าเป็น true จะทำการแสดงปุ่ม Unnotify ดังรูปที่ 3.23

```
<center v-show="countObj">
  <button @click="postNotifyMeUser()" id="btnNotify" class="notifyMe" v-if="this.notifyStatus == false">
    <font-awesome-icon class="bSlash" style="display:none;" icon="bell-slash" />
    <font-awesome-icon class="bell" icon="bell" />&nbsp;&nbsp;&nbsp;<span class="No">Notify Me</span>
  </button>
  <button @click="putUnnotifyMeUser()" id="btnUnnotify" class="clickedNotifyMe" v-if="this.notifyStatus == true">
    <font-awesome-icon class="bSlash" icon="bell-slash" />
    <font-awesome-icon class="bell" style="display:none;" icon="bell" />&nbsp;&nbsp;&nbsp;<span class="No">Unnotify</span>
  </button>
</center>
```

รูปที่ 3.23 เงื่อนไขการแสดงปุ่ม Notify Me และ Unnotify

3.3.2 ฟังก์ชันในหน้า Browse Item

ในหน้า Browse Item จะแบ่งออกเป็น 3 Components หลัก ได้แก่ แถบเมนูด้านข้าง (Side Menu) กล่อง Search และข้อมูลอุปกรณ์ที่สามารถยืมได้ ดังรูปที่ 3.24 โดยในแต่ละส่วนก็จะมีการทำงาน เชื่อมกัน จึงจะขออธิบายในภาพรวม



รูปที่ 3.24 หน้า Browse Item

การแสดงผล Category และ Sub-category ของอุปกรณ์นั้น จะขึ้นอยู่กับสถานที่ที่ผู้ใช้งานได้เลือกไว้ภายใน Select Location Modal โดยจะเริ่มจากการนำค่าตัวแปรของสถานที่ มาใช้ในการเรียก API ของข้อมูล ดังรูปที่ 3.25

```
multipleAxios() {
  this.$axios.all([
    this.$axios.get(`${this.siteID}/api/Category?siteid=` + this.siteID),
    this.$axios.get(`${this.siteID}/api/Subcategory?siteid=` + this.siteID),
    this.$axios.get(`${this.siteID}/api/Item?siteid=` + this.siteID)
  ]).then(this.$axios.spread((firstResponse, secondResponse, thirdResponse) => {

    if (firstResponse.data.StatusDescription == "SUCCESS") {
      this.$store.dispatch('items/setCategory', firstResponse.data.Message);
    }
    if (secondResponse.data.StatusDescription == "SUCCESS") {
      this.$store.dispatch('items/setSubCategory', secondResponse.data.Message);
    }
    if (thirdResponse.data.StatusDescription == "SUCCESS") {
      this.$store.dispatch('items/setAllItem', thirdResponse.data.Message);
    }
    this.createItems()
  }));
},
```

รูปที่ 3.25 คำสั่งเรียกข้อมูล Category, Sub-category และข้อมูลอุปกรณ์ ผ่าน API

ข้อมูลของ Category จะเก็บอยู่ในตัวแปร firstResponse โดยภายในตัวแปรจะเป็นชื่อของ Category ทั้งหมดที่มีอยู่ในอาคารแห่งนี้ เช่นเดียวกับ Sub-category จะถูกเก็บอยู่ในตัวแปร secondResponse และภายในตัวแปรจะเป็นชื่อ Sub-category ทั้งหมดในอาคารนี้ โดยจะมีตัวเชื่อมบอก ว่า Sub-category นี้ อยู่ภายใต้ Category ไต จึงทำให้จะต้องนำมาเรียบเรียงใหม่เพื่อให้มีการแสดงชื่อของ Category เป็นหัวข้อหลัก แล้วภายในหัวข้อหลักก็จะประกอบไปด้วย Sub-category เป็นหัวข้อย่อยภายใน โดยจะเข้าสู่ฟังก์ชัน createItems() ดังรูปที่ 3.26 ซึ่งจะเป็นการจัดเรียงหมวดหมู่หลัก และหมวดหมู่ย่อยให้ตรงกัน เช่น หมวดหมู่ Input Devices จะมีหมวดหมู่ย่อยเป็น Keyboard และ Mouse

ส่วนข้อมูลของอุปกรณ์ทั้งหมดที่สามารถยืมได้ ภายในอาคารสำนักงานนี้ จะถูกเก็บและไว้ในตัวแปร thirdResponse และนำมาแสดงภายในหน้า Browse Item

```

createItems() {
  var items = [];
  var category = this.category;
  var subCategory = this.subCategory;

  category.forEach(function (cat) {
    var sub = subCategory.filter(val => {
      return cat.strCategoryId == val.strCategoryId;
    });
    var children = []
    sub.forEach(function (subCat) {
      children.push({
        name: subCat.strSubCategoryId,
        id: subCat.intSubCategoryId,
      })
    });
    items.push({
      name: cat.strCategoryId,
      id: cat.intCategoryId,
      open: true,
      children: children
    });
  });
  this.model = items;
  return this.model
}

```

รูปที่ 3.26 คำสั่ง createItems()

นอกจาก Side Menu จะสามารถบอกว่ามี Category และ Sub-category ใดบ้างสำหรับอาคารสำนักงานนี้ Side Menu ยังสามารถกดเพื่อที่จะทำการกรองเพื่อดูอุปกรณ์ทั้งหมดที่อยู่ใน Sub-category ที่ทำการเลือกได้อีกด้วย โดยเมื่อทำการกดจะมีการส่งค่าของ ID ของ Category และ Sub-category ภายใน Parameters ผ่านทาง Route ดังรูปที่ 3.27 และจะทำการสั่งให้ส่วนที่แสดงข้อมูลของอุปกรณ์ทั้งหมด ทำการกรองผลลัพธ์ออกมา ตามคำสั่งในรูปที่ 3.28

```

<template>
  <div class="treeNavigation">
    <li>
      <router-link :to="{ name: 'BrowseItem', params: {categoryID: 0} }">All</router-link>
    </li>
    <li v-for="(category,index) in model" :key="index" :category="model">
      <span v-if="isFolder(category)" @click="toggle(category)">{{ category.open ? '&#x25BD;' : '&#x25B6;' }}</span>
      <router-link :to="{ name: 'BrowseItem', params: {categoryID: category.id} }">&nbsp;{{category.name}}</router-link>
      <ul v-show="category.open">
        <li v-for="(subCat,indexChild) in category.children" :key="indexChild" :subCat="category" class="subCatMenu">
          <router-link :to="{ name: 'BrowseItem', params: {subCategoryId: subCat.id , categoryID: category.id} }">
            &nbsp;{{subCat.name}}
          </router-link>
        </li>
      </ul>
    </li>
  </div>
</template>

```

รูปที่ 3.27 การส่งค่า Parameters ผ่านทาง Route

```

filterItem: function(){
  if(typeof this.$route.params.categoryID === 'undefined'){
    return this.$store.getters['items/searchItem']
  }
  else if (typeof this.$route.params.subCategoryId === 'undefined') {
    return this.$store.getters['items/getFilterItemCatId'](this.$route.params.categoryID)}
  else
    return this.$store.getters['items/getFilterItemSubId'](this.$route.params.subCategoryId)
}

```

รูปที่ 3.28 คำสั่ง Filter อุปกรณ์ในหน้า Browse Item

ในการพิมพ์ข้อมูลค้นหาบนกล่อง Search ก็จะใช้คำสั่งดังรูปที่ 3.28 เช่นเดียวกัน โดยจะเข้าไปในเงื่อนไขแรก มีการสั่งให้ทำฟังก์ชันที่ชื่อว่า searchItem ดังรูปที่ 3.29 โดยคำสั่งนี้จะนำตัวอักษรที่พิมพ์เข้าไป ไปทำการเปรียบเทียบกับชื่อของอุปกรณ์ ชื่อ Category หรือ ชื่อของ Sub-category ที่ถูกเก็บเอาไว้ในตัวแปรแล้วจะนำมาแสดงผลในหน้า Browse Item

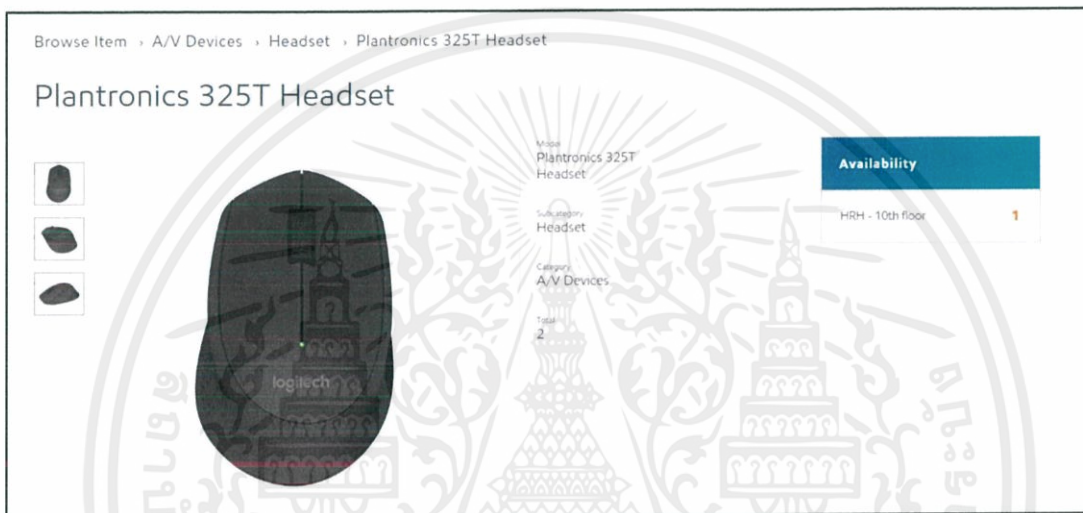
```

searchItem: (state) => {
  if (state.searchText != "") {
    return state.allItem.filter(item => {
      var text = item.strItemName.toLowerCase()
      var text2 = state.subCategory.find(items => items.intSubCategoryId === item.intSubCategoryId).strSubCategoryName.toLowerCase()
      return text.includes(state.searchText.toLowerCase()) || text2.includes(state.searchText.toLowerCase())
    })
  }
  else
    return state.allItem
}

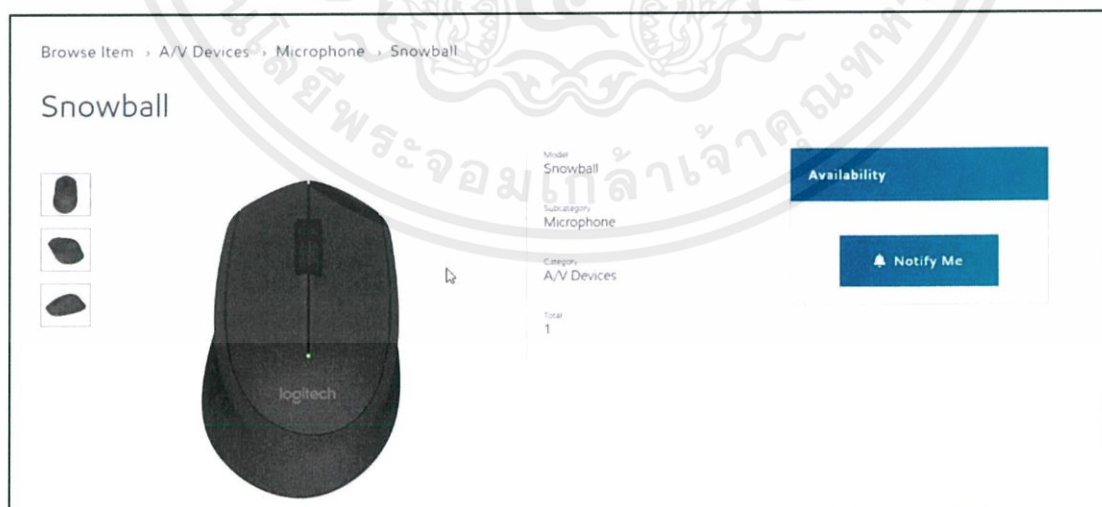
```

รูปที่ 3.29 คำสั่ง Search Item

ข้อมูลของอุปกรณ์ที่แสดงภายในหน้า Browse Item จะขึ้นอยู่กับสถานที่ที่ผู้ใช้งานได้เลือกเอาไว้ ซึ่งถูกเก็บเอาไว้ในตัวแปร thirdResponse ตามที่กล่าวไว้ข้างต้น ดังนั้นถ้าเปลี่ยนสถานที่ข้อมูลทั้งหมดก็จะเปลี่ยนไป แต่ถ้าหากอยากจะดูข้อมูลเพิ่มเติมของอุปกรณ์ชิ้นนั้น ก็สามารถกด 'View Item' เพื่อดูข้อมูลเพิ่มเติมได้ โดยหน้าที่แสดงจะมีลักษณะเดียวกันกับหน้าข้อมูลเพิ่มเติมเมื่อออกจาก Watch List จะต่างที่ว่าหน้าจากฝั่ง Browse Item จะแสดงจำนวนของที่ยังเหลือ และของที่สามารถกดติดตามเพื่อให้เก็บเข้าไปใน Watch List ได้ ดังรูปที่ 3.30 และ 3.31 ตามลำดับ



รูปที่ 3.30 หน้าข้อมูลเพิ่มเติมของอุปกรณ์ในกรณีที่ยังมีของให้ยืม



รูปที่ 3.31 หน้าข้อมูลเพิ่มเติมของอุปกรณ์ในกรณีที่ของหมด

โดยในกรณีนี้ของหมด และผู้ใช้งานต้องการที่จะกดติดตามอุปกรณ์ชิ้นนี้ ให้ปรากฏใน Watch List หรือจะกดยกเลิกการติดตาม ก็จะขึ้นอยู่กับปุ่มที่ปรากฏอยู่บนหน้าเว็บ ดังที่ได้กล่าวไปข้างต้น เมื่อกดติดตามไปแล้ว ปุ่มจะเปลี่ยนเป็นปุ่ม Unnotify สีส้ม แต่ถ้ายังไม่ได้กดติดตาม ปุ่มจะเป็นปุ่ม Notify Me สีฟ้า โดยคำสั่งที่จะทำให้ปุ่มทำงาน และส่งข้อมูลการติดตามกลับไปยังฐานข้อมูล จะเป็นดังรูปที่ 3.32

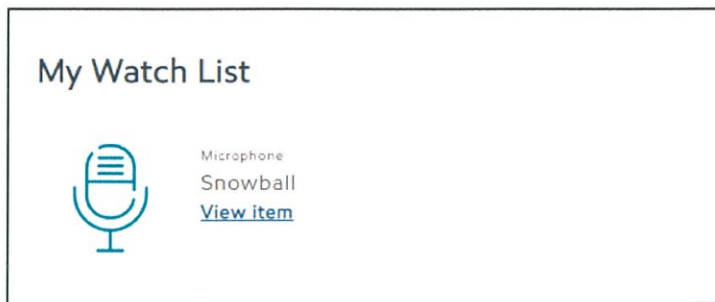
```
<center v-show="countObj">
  <button @click="postNotifyMeUser()" id="btnNotify" class="notifyMe" v-if="this.notifyStatus == false">
    <font-awesome-icon class="bSlash" style="display:none;" icon="bell-slash" />
    <font-awesome-icon class="bell" icon="bell" />&nbsp;&nbsp;&nbsp;<span class="No">Notify Me</span>
  </button>
  <button @click="putUnnotifyMeUser()" id="btnUnnotify" class="clickedNotifyMe" v-if="this.notifyStatus == true">
    <font-awesome-icon class="bSlash" icon="bell-slash" />
    <font-awesome-icon class="bell" style="display:none;" icon="bell" />&nbsp;&nbsp;&nbsp;<span class="No">Unnotify</span>
  </button>
</center>
```

รูปที่ 3.32 คำสั่งติดตามอุปกรณ์ และคำสั่งยกเลิกการติดตามอุปกรณ์

เมื่อผู้ใช้งานต้องการติดตามอุปกรณ์ เมื่อกดที่ปุ่ม จะมีการเรียกใช้ฟังก์ชันที่มีชื่อว่า postNotifyMeUser() ซึ่งจะทำการส่งข้อมูลผ่าน API ไปยังฐานข้อมูลว่าผู้ใช้งานได้ทำการติดตามอุปกรณ์ หมายเลขนี้ ที่ Location นี้ และจะทำการโหลดข้อมูลใหม่ ดังรูปที่ 3.33 ซึ่งจะทำให้สีของปุ่มเปลี่ยนไปเป็น Unnotify สีส้ม และเมื่อเปิดไปยังหน้า Dashboard ส่วน Watch List ก็จะปรากฏอุปกรณ์ที่ได้ทำการกดติดตามไปทันที ดังรูปที่ 3.34

```
postNotifyMeUser() {
  $('#btnNotify').attr("disabled", true)
  this.$axios.post('http://localhost:3000/api/item/watchlist/notify', {
    SiteId: this.siteID,
    UserId: this.userID,
    ItemId: parseInt(this.$route.params.itemID)
  })
  .then((result) => {
    $('#btnNotify').attr("disabled", false)
    this.loadItemDetail()
  }).catch((err) => {
    console.log('error', err)
    this.loadItemDetail()
  });
},
```

รูปที่ 3.33 คำสั่ง postNotifyMeUser()



รูปที่ 3.34 Watch List หลังจากกดติดตามอุปกรณ์

แต่ถ้าหากต้องการยกเลิกการติดตามอุปกรณ์ ก็สามารถกดที่ปุ่ม Unnotify สีส้ม ก็จะเป็นการเรียกใช้ฟังก์ชันที่มีชื่อว่า putUnnotifyMeUser() ดังรูปที่ 3.32 ก็จะทำให้เกิดการส่งข้อมูลอัปเดตผ่าน API ไปยังฐานข้อมูล ว่าผู้ใช้งานหมายเลขนี้ ได้กดยกเลิกติดตามอุปกรณ์หมายเลขนี้เรียบร้อยแล้ว ดังรูปที่ 3.35 จากนั้นก็จะทำการโหลดข้อมูลใหม่ ทำให้ปุ่มเปลี่ยนกลับไปเป็นปุ่ม Notify Me สีฟ้า และในส่วนของ Watch List ก็จะไม่ปรากฏอุปกรณ์ชิ้นนั้นแล้ว

```
putUnnotifyMeUser() {
  $('#btnUnnotify').attr("disabled", true)
  this.$axios.put('http://localhost:3000/api/item/watchlist/unnotify', {
    SiteId: this.siteID,
    UserId: this.userID,
    ItemId: parseInt(this.$route.params.itemID)
  })
  .then((result) => {
    $('#btnUnnotify').attr("disabled", false)
    this.loadItemDetail()
  }).catch((err) => {
    console.log('error', err)
    this.loadItemDetail()
  });
},
```

รูปที่ 3.35 คำสั่ง putUnnotifyMeUser()

3.4 ทดสอบการใช้งานเว็บไซต์

ในขณะที่ผู้วิจัยได้ทำการพัฒนาโปรแกรม ผู้วิจัยก็ได้ทำการทดสอบไปด้วยในเวลาเดียวกัน ซึ่งเมื่อพบปัญหาใด ผู้วิจัยก็จะทำการแก้ไขในทันที แต่ก่อนที่ผู้วิจัยจะสำเร็จในโครงการสหกิจศึกษา ผู้วิจัยได้ทำการทดสอบการใช้งานเว็บไซต์ครั้งใหญ่ โดยทดสอบในทุกกรณีที่สามารถเกิดขึ้นได้ และทดสอบร่วมกับทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฝั่ง Back-end ทำให้พบปัญหาหลาย ๆ ปัญหาเกิดขึ้น ทั้งในฝั่งของตัวเว็บไซต์ (Front-end) และฝั่งของ Back-end ผู้วิจัยจึงได้ใช้เวลาที่เหลืออยู่ในการแก้ไขปัญหาร่วมกับสมาชิกทีม จนกระทั่งทุกปัญหาที่ผู้วิจัยพบได้รับการแก้ไขเรียบร้อย และสำเร็จลุล่วงไปได้ด้วยดี ผู้วิจัยจึงได้เขียนเอกสารเพื่อส่งมอบงานให้กับพนักงานร่วมทีมไว้ใช้ในการพัฒนาต่อไป



บทที่ 4

ผลการวิจัย

เนื่องจากเว็บไซต์นี้ เป็นเว็บไซต์ที่สร้างขึ้นมาเพื่อแก้ไขจุดบกพร่องของโปรเจกต์ Smart Locker ในรุ่นก่อน การสร้างเว็บไซต์จึงจะต้องสร้างขึ้นตามความต้องการที่ถูกระบุมา โดยในส่วนของผู้วิจัยที่ทำการพัฒนาเว็บไซต์ในส่วนของผู้ใช้งานทั่วไป จำแนกส่วนการใช้งานที่ผู้วิจัยได้พัฒนาว่าตรงตามความต้องการที่ถูกระบุไว้ในตอนต้นหรือไม่ โดยจะแบ่งออกเป็น 4 การใช้งาน ดังนี้

4.1 ข้อมูลอุปกรณ์เปลี่ยนแปลงตามสถานที่ของผู้ใช้งาน

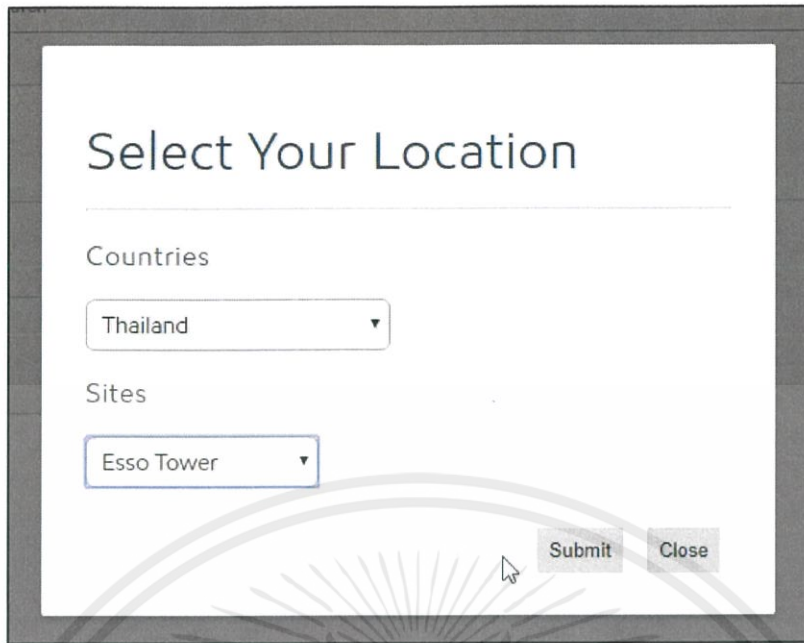
ผู้วิจัยได้ทำการเลือกสถานที่ครั้งแรก โดยเลือก Country เป็น Thailand และเลือก Site เป็น Harindhorn (ตึกहरिनธร) ซึ่งข้อมูลของอุปกรณ์ ไม่ว่าจะเป็น Side Menu ทางด้านซ้าย ที่แสดง Category และ Sub-category ภายใต้ Site นี้ และรายชื่อของอุปกรณ์ที่สามารถยืมได้ ที่ถูกแสดงภายในหน้า Browse Item เป็นดังรูปที่ 4.1



รูปที่ 4.1 ข้อมูลที่แสดงของตึกहरिनธร ประเทศไทย

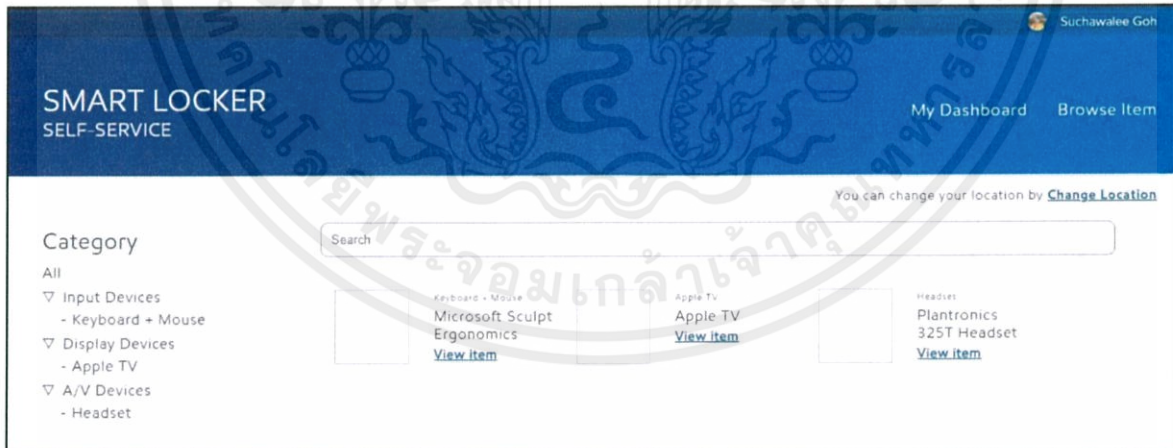
จากนั้นผู้วิจัยทำการทดลองเปลี่ยนสถานที่ใหม่ โดยที่ยังคงอยู่ในประเทศไทย แต่เปลี่ยน Site เป็น Esso Tower ดังรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 ทดลองเปลี่ยน Site ของผู้ใช้งาน

เมื่อทำการโหลดข้อมูลใหม่ พบว่าข้อมูลต่าง ๆ ภายในหน้า Browse Item ได้เปลี่ยนแปลงไปทั้งหมดแล้ว ไม่ว่าจะเป็น Side Menu ด้านซ้าย และรายชื่ออุปกรณ์ทั้งหมดที่สามารถยืมได้ ดังรูปที่ 4.3



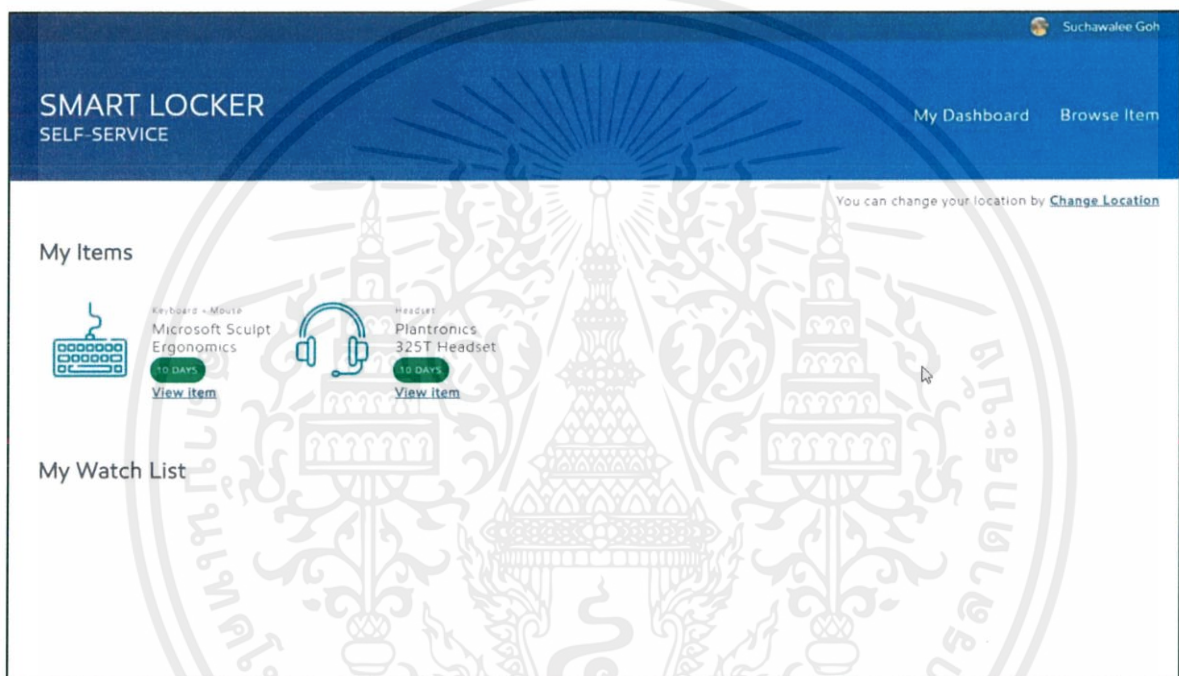
รูปที่ 4.3 ข้อมูลที่แสดงของตึกเอสไอ ประเทศไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อ 45 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การใช้งานฟังก์ชัน Notify Me/Unnotify

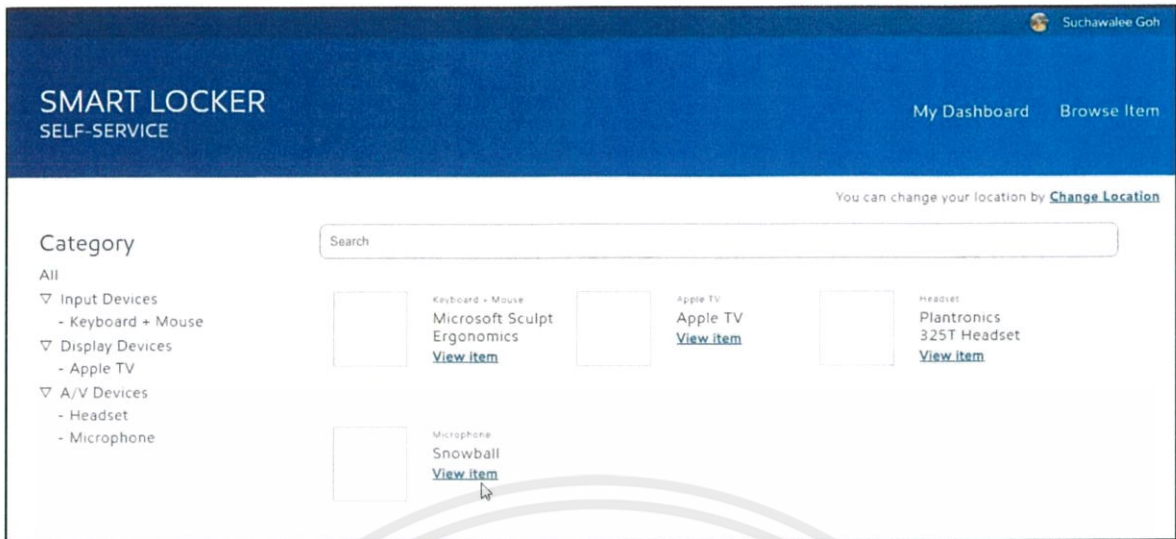
ภายในฐานข้อมูล ได้มีการกำหนดให้อุปกรณ์ที่ชื่อว่า Snowball มีจำนวนเหลือ 0 เพื่อที่จะนำมาทดลองเรื่องการติดตามอุปกรณ์ และการนำผลลัพธ์มาแสดงในส่วนของ Watch List ภายในหน้า Dashboard

อันดับแรกจะเริ่มจากการตรวจสอบหน้า Dashboard ก่อนที่จะทำการกดติดตามอุปกรณ์ชิ้นนี้ ซึ่งให้ผลลัพธ์ดังรูปที่ 4.4

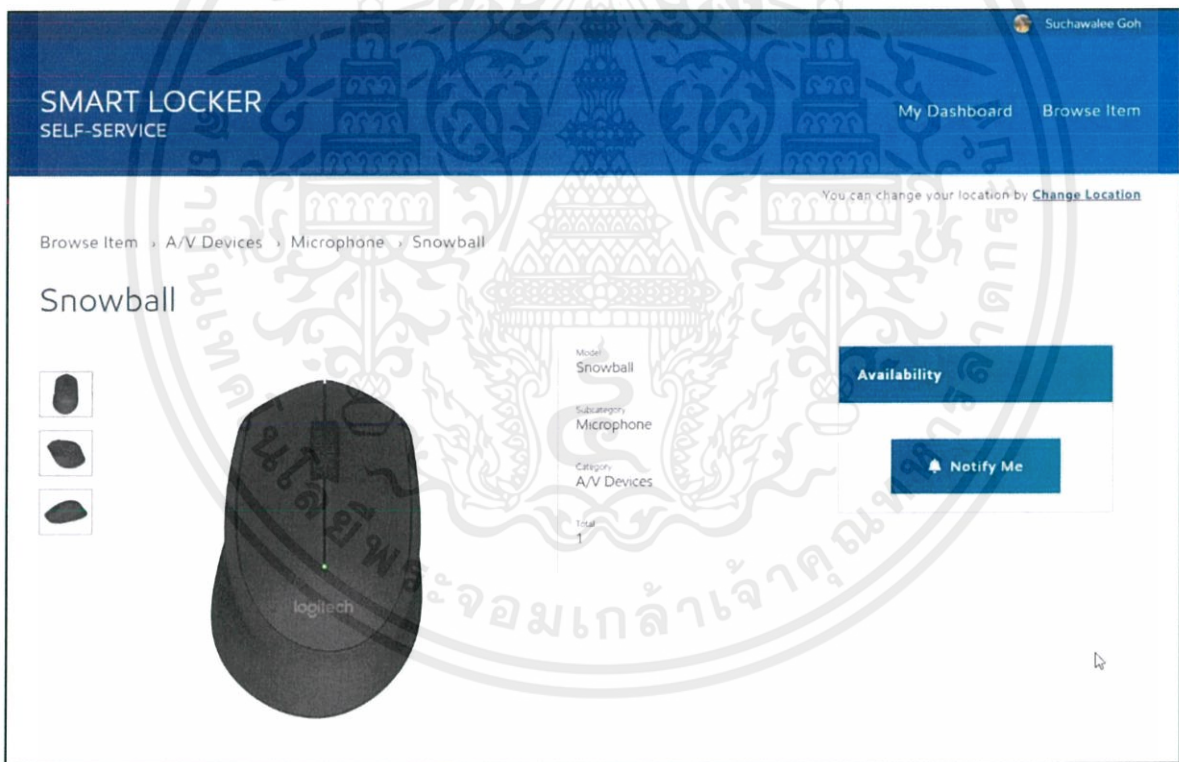


รูปที่ 4.4 หน้า Dashboard ก่อนกดติดตามอุปกรณ์

จากนั้นไปที่หน้า Browse Item และไปยังอุปกรณ์ที่ชื่อว่า Snowball เมื่อกดเข้าไปจะพบว่ามีการแสดงปุ่ม Notify Me สีฟ้า หมายความว่าอุปกรณ์ชิ้นนี้ยังไม่ถูกติดตาม จึงทำการกดไปที่ปุ่ม Notify Me สีฟ้า และปุ่มได้ทำการเปลี่ยนเป็นปุ่ม Unnotify สีส้ม เรียบร้อยแล้ว ดังรูปที่ 4.5 รูปที่ 4.6 และรูปที่ 4.7 ตามลำดับ

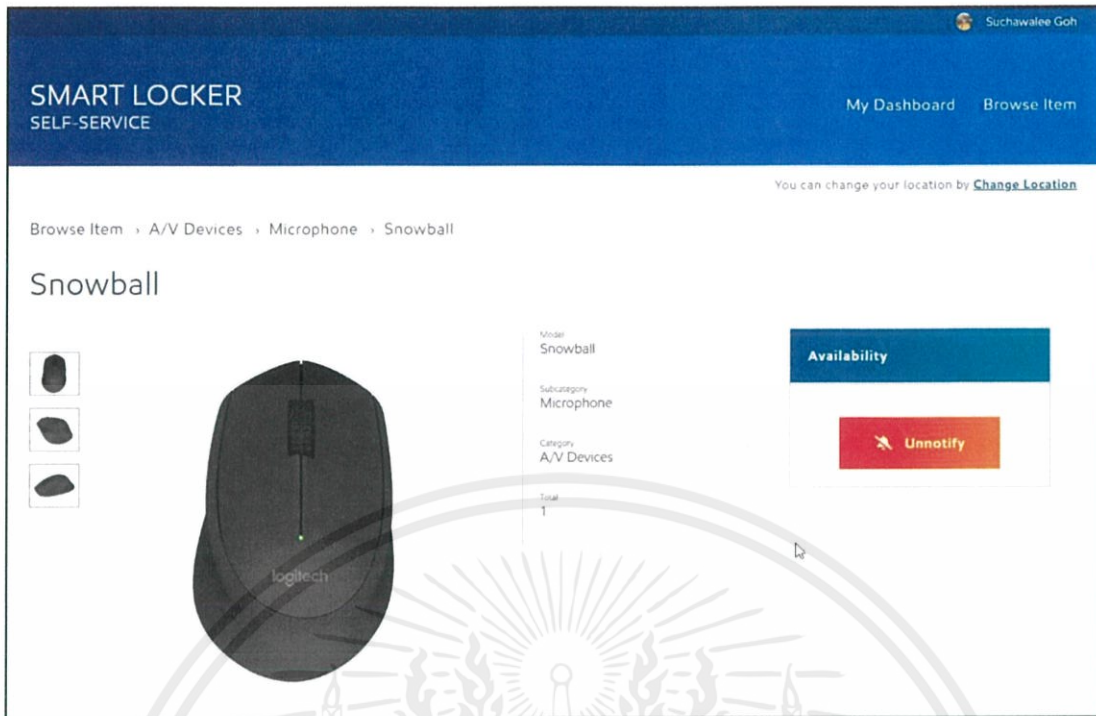


รูปที่ 4.5 หน้า Browse Item ที่แสดงอุปกรณ์ชื่อ Snowball



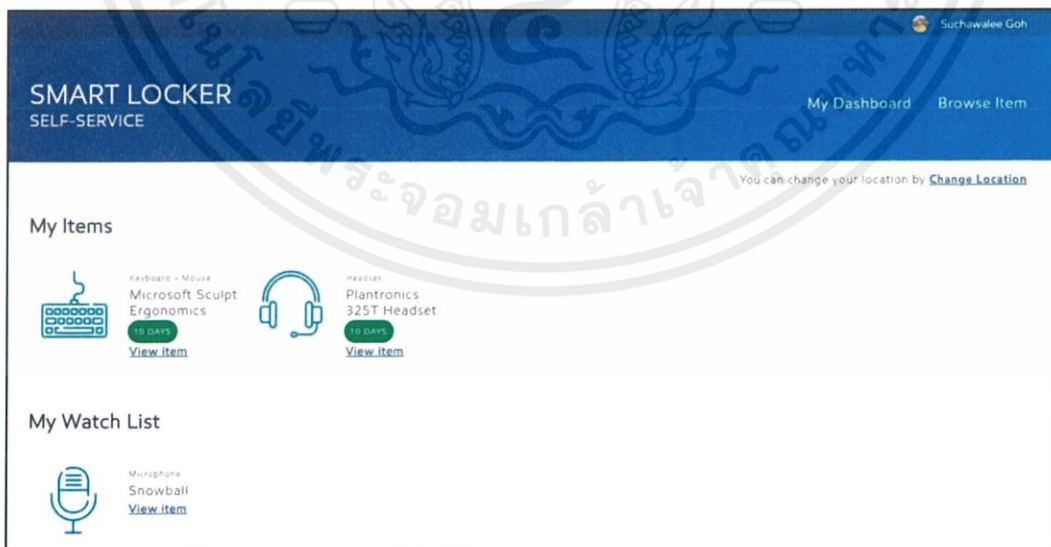
รูปที่ 4.6 ข้อมูลเพิ่มเติมของอุปกรณ์ Snowball และปุ่ม Notify Me

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อ 47 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 หลังจากทำการกดปุ่ม Notify Me สีฟ้า

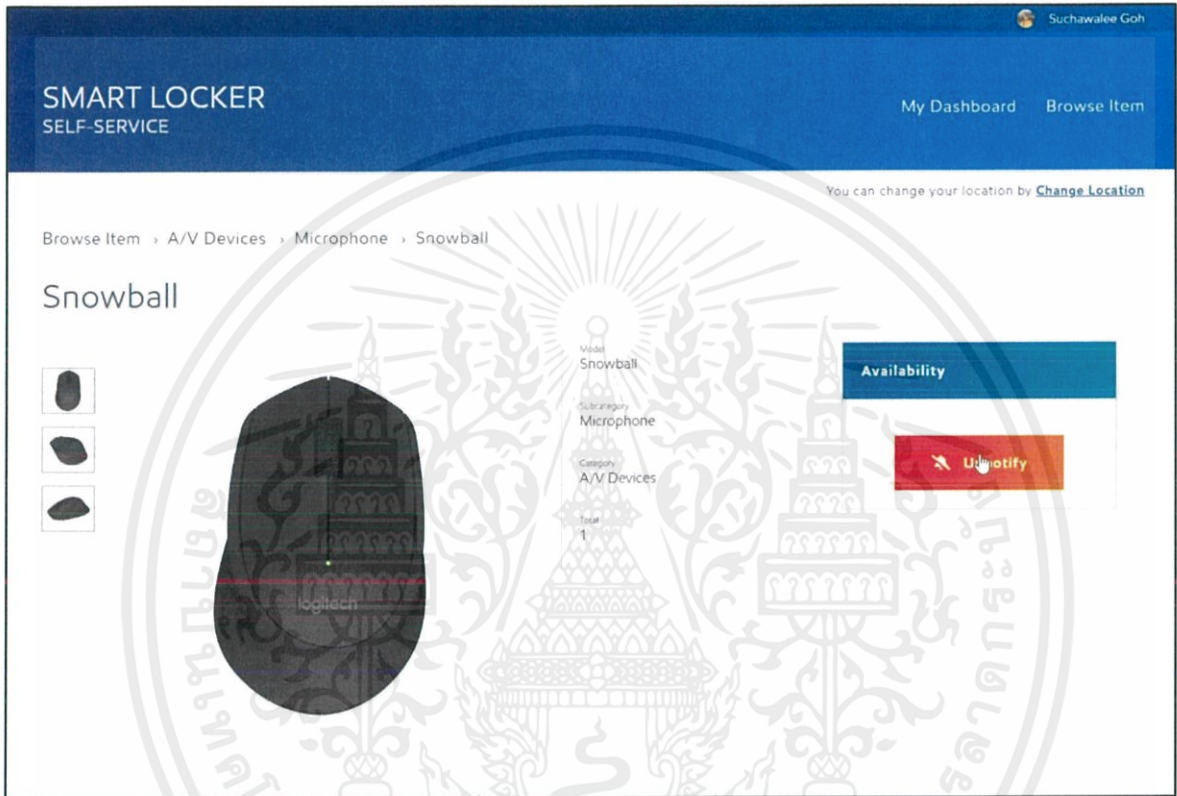
จากนั้นจึงกลับไปหน้าจอ Dashboard อีกครั้งว่าอุปกรณ์ที่ได้ติดตามปรากฏขึ้นมาตรง Watch List หรือไม่ ดังรูปที่ 4.8



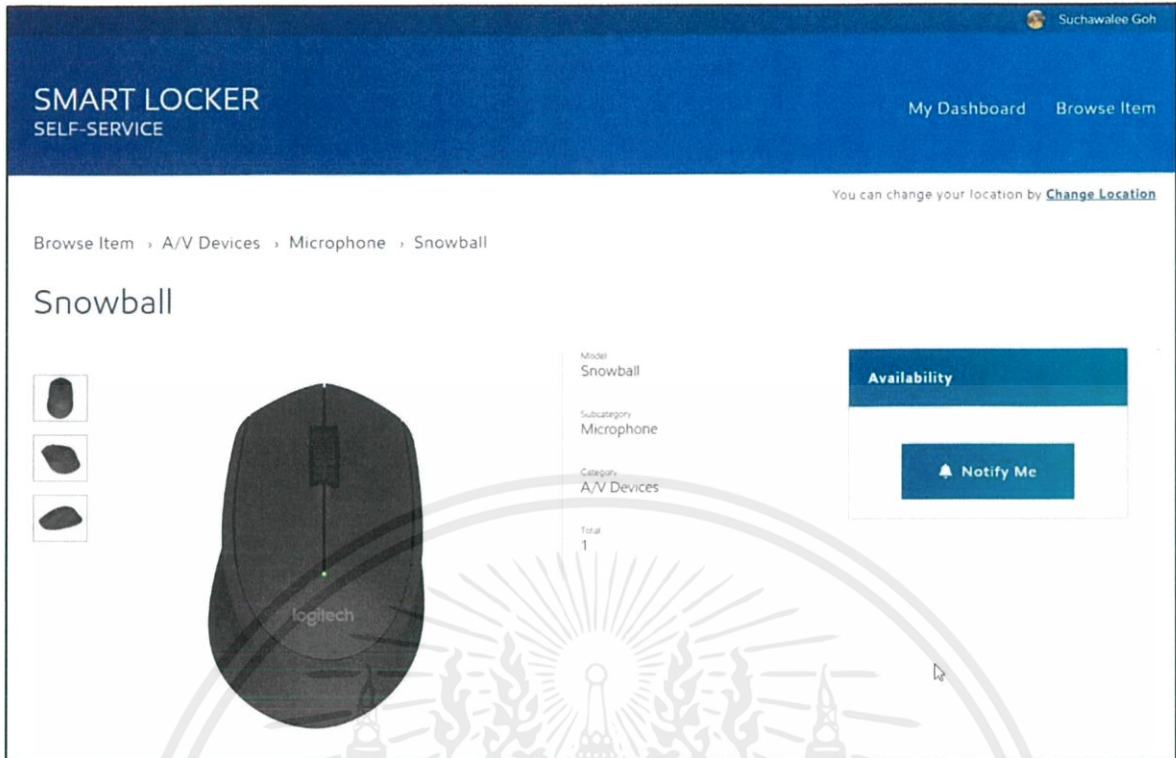
รูปที่ 4.8 หน้า Dashboard หลังจากกดปุ่ม Notify Me

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อ 48 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

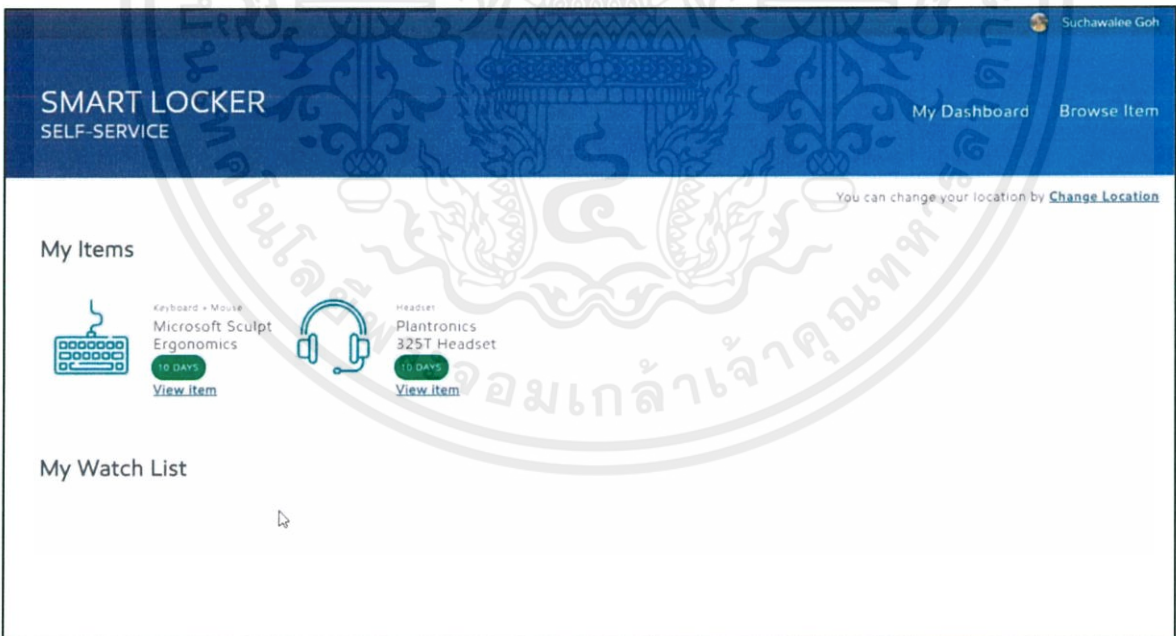
และเพื่อการทดสอบในทางกลับกันจึงจะต้องทำการกดยกเลิกการติดตามด้วย ผู้วิจัยจึงเข้าไปกดยกเลิกการติดตามจากหน้า Dashboard จะพบว่าปุ่มได้เปลี่ยนกลับไปเป็นปุ่ม Notify Me สีฟ้า แล้วเมื่อกลับไปดูหน้า Dashboard อีกครั้ง ก็พบว่าอุปกรณ์ Snowball ได้หายไปจาก Watch List เรียบร้อยแล้ว ดังรูปที่ 4.9 รูปที่ 4.10 และรูปที่ 4.11 ตามลำดับ



รูปที่ 4.9 ทำการกดปุ่ม Unnotify เพื่อยกเลิกการติดตาม



รูปที่ 4.10 ปุ่มได้เปลี่ยนกลับเป็นปุ่ม Notify สีฟ้า หลังจากกดยกเลิกการติดตาม



รูปที่ 4.11 อุปกรณ์ Snowball ถูกลบออกจาก Watch List

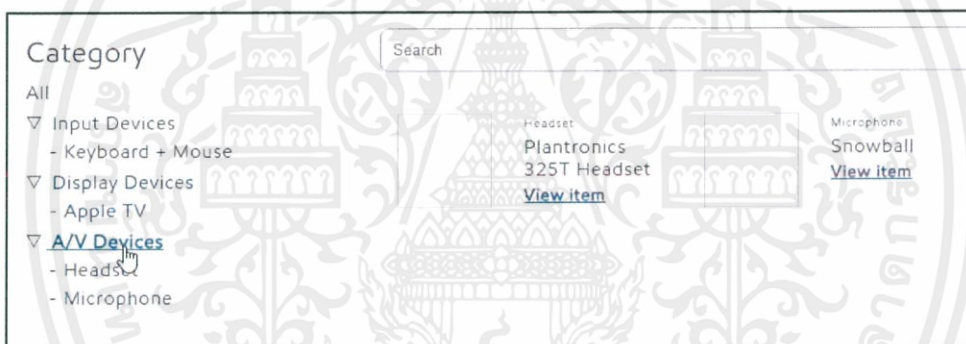
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อ 50 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การกรองข้อมูลอุปกรณ์

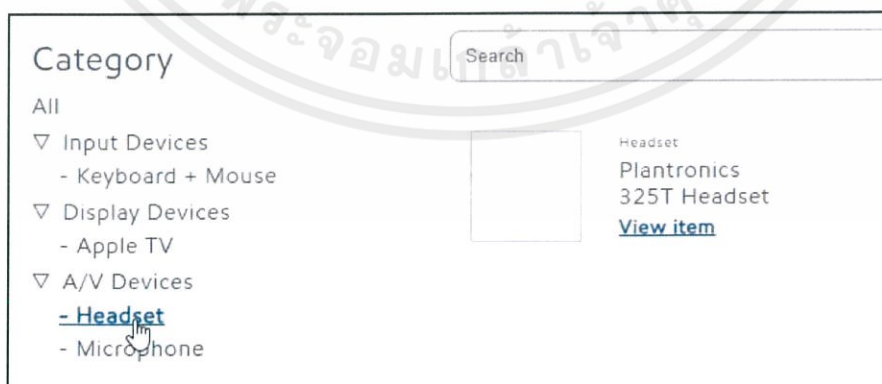
ภายในหน้า Browse Item จะมีอยู่สองจุดที่สามารถทำการกรองข้อมูลที่แสดงได้ คือการคลิกที่ชื่อ Category หรือ Sub-category ก็จะทำให้การกรองผลลัพธ์ที่อยู่ภายใต้ Category หรือ Sub-category นั้น ๆ และอีกวิธีคือการค้นหา Search Box การทดลองในครั้งนี้จะทำการทดลองทั้งสองกรณี เพื่อที่จะตรวจสอบว่าผลลัพธ์ที่ได้ออกมา นั้นถูกต้องตามต้องการหรือไม่

1. การกรองข้อมูลบน Side Menu

การกรองสามารถทำได้จากการคลิกชื่อของ Category หรือ ชื่อของ Sub-category ได้ทั้งคู่ โดยที่ถ้าคลิกที่ชื่อ Category นั้น ก็จะแสดงของทุกชิ้นที่อยู่ภายใต้ Category นั้น ซึ่งของอาจจะไม่ได้อยู่ใน Sub-category เดียวกันก็ตาม ดังรูปที่ 4.12 หรือถ้าคลิกที่ชื่อ Sub-category ก็จะแสดงของทุกชิ้นที่อยู่ภายใต้ Sub-category เดียวกัน ดังรูปที่ 4.13



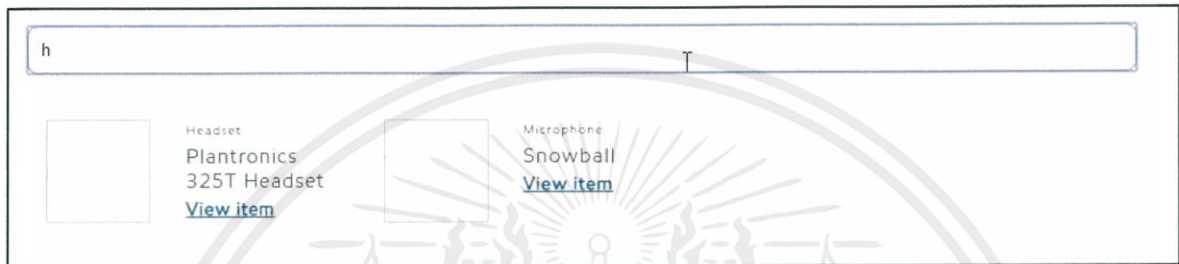
รูปที่ 4.12 กรองด้วยชื่อ Category



รูปที่ 4.13 กรองด้วยชื่อ Sub-category

2. การกรองข้อมูลด้วย Search Box

การพิมพ์ชื่อเข้าไปใน Search Box ไม่ว่าจะป็นชื่อของอุปกรณ์ ชื่อ Category หรือชื่อ Sub-category ก็สามารถกรองผลลัพธ์ออกมาได้ทุกรูปแบบ โดยการกรองจะเป็นการค้นหาในแต่ละตัวอักษร ดังนั้นเพียงพิมพ์ข้อมูลเพิ่มอีก 1 ตัวอักษร ก็จะสามารถทำให้ผลลัพธ์แตกต่างกันได้ ดังรูปที่ 4.14 และรูปที่ 4.15 ตามลำดับ



รูปที่ 4.14 การพิมพ์ค้นหาด้วยตัวอักษร 'h' เพียงตัวเดียว



รูปที่ 4.15 การพิมพ์ค้นหาด้วยตัวอักษร 'he'

4.4 การตรวจสอบจำนวนอุปกรณ์ที่สามารถยืมได้ผ่านทางเว็บไซต์

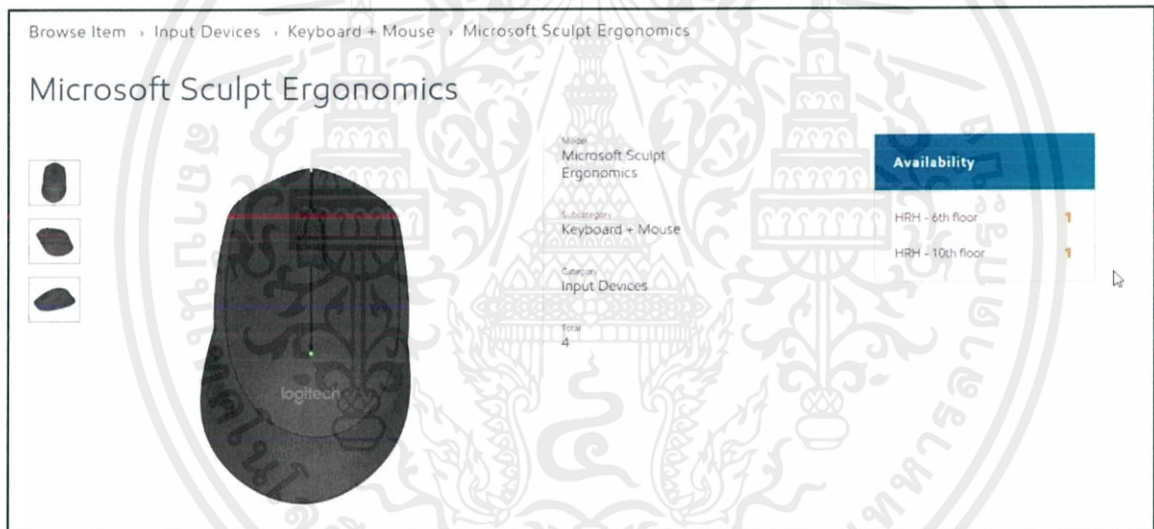
เมื่อเปิดไปยังหน้า Browse Item ผู้ใช้งานสามารถตรวจสอบจำนวนของได้จากการกด 'View Item' ของอุปกรณ์ที่สนใจ จะมีตาราง Availability บอกจำนวนของที่เหลืออยู่ภายในล็อกเกอร์ใดบ้าง ดังรูปที่ 4.16 และรูปที่ 4.17 ตามลำดับ

โดยมีเงื่อนไขภายในโค้ด ดังรูปที่ 4.18 ในกรณีที่มีจำนวนมากกว่าหรือเท่ากับ 10 ชิ้น ตัวเลขแสดงจะเป็นสีเขียว ถ้ามีจำนวนน้อยกว่า 10 ชิ้น แต่มากกว่าหรือเท่ากับ 5 ชิ้น จะแสดงตัวเลขสีเหลือง แต่ถ้าเป็นคำว่า N/A (Not Available) จะเป็นตัวอักษรสีแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อ 52 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 กด View Item ของอุปกรณ์ที่สนใจจะตรวจสอบจำนวน



รูปที่ 4.17 ข้อมูลเพิ่มเติมของอุปกรณ์ และจำนวนของที่เหลือ

```

<div class="available-list">
  <table width="100%">
    <tr v-for="(index) in availability" :key="index.topic">
      <td class="item-topic">{{index.topic}}</td>
      <td class="item-info outOfStock" align="right" v-if="index.info === 'N/A'">{{index.info}}</td>
      <td class="item-info lowInStock" align="right" v-else-if="index.info <= 5">{{index.info}}</td>
      <td class="item-info plentyInStock" align="right" v-else-if="index.info >= 10">{{index.info}}</td>
    </tr>
  </table>
</div>

```

รูปที่ 4.18 เงื่อนไขของสีที่แสดงต่อจำนวนของที่เหลือ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อ 53 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัย และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

การทำวิจัยนี้มีจุดประสงค์ที่จะแก้ไขข้อบกพร่องของระบบในรุ่นก่อน เพื่อนำมาปรับปรุงใหม่ให้สามารถใช้งานได้อย่างมีประสิทธิภาพ มีความสะดวกสบาย และง่ายต่อการใช้งานมากขึ้น โดยผู้วิจัยได้รับหน้าที่ในส่วนของการพัฒนาเว็บไซต์สำหรับผู้ใช้งานทั่วไป

ซึ่งหลังจากการพัฒนาเสร็จสิ้น ผู้วิจัยได้ทำการทดลองและตรวจสอบระบบ พบว่าเว็บไซต์สามารถทำงานได้อย่างมีประสิทธิภาพ และให้ผลลัพธ์ที่ถูกต้อง อีกทั้งยังช่วยให้เข้าถึงข้อมูลของอุปกรณ์ได้ง่ายและสะดวกมากขึ้น รวมไปถึงมีฟังก์ชันการทำงานครบตามความต้องการที่ได้ระบุไว้ ซึ่งเป็นผลดีต่อระบบโดยรวม เนื่องจากมีแรงจูงใจให้มีผู้เข้ามาใช้งานระบบได้มากขึ้น และนำไปสู่การพัฒนาให้ดียิ่งขึ้นในอนาคต

5.2 ข้อเสนอแนะ

เนื่องจากการทำงานภายในองค์กรขนาดใหญ่ การที่จะนำเทคโนโลยีใหม่ ๆ เข้ามาใช้งานจึงไม่ค่อยนิยมกันมากนัก ทำให้เมื่อต้องการนำเทคโนโลยีใหม่เข้ามาใช้งาน จึงมักจะขาดบุคลากรที่มีประสบการณ์ใช้งานในเทคโนโลยีนั้น ๆ ส่งผลให้จะต้องใช้เวลาในการเรียนรู้เทคโนโลยีใหม่เป็นเวลานาน อีกทั้งเมื่อเวลาที่จะนำเทคโนโลยีใหม่ ๆ มาใช้งานภายในบริษัท จะทำได้ค่อนข้างยาก เนื่องจากจะต้องมีการเขียนคำร้องขอและรอในการอนุมัติอีกด้วย จึงทำให้เกิดการเสียเวลาขึ้น รวมไปถึงการลงโปรแกรมเพื่อใช้งานภายในคอมพิวเตอร์ของบริษัทก็จะยากขึ้น เนื่องจากเหตุผลทางด้านความปลอดภัย จึงต้องทำให้เสียเวลาในทุกขั้นตอนการทำงาน ไม่ว่าจะเป็นขั้นตอนการเขียนคำร้องขอลงโปรแกรม ขั้นตอนการลงโปรแกรม และในขั้นตอนการศึกษาเทคโนโลยีใหม่ ๆ เพื่อให้การทำงานมีประสิทธิภาพอย่างสูงสุด

เอกสารอ้างอิง

1. Designil. 2015. Wireframe คืออะไร ทำไมดีไซน์เนอร์ต้องรู้ พร้อมวิธีทำ + Tool ฟรี !!. สืบค้นเมื่อ 12 มกราคม 2562, จาก <https://www.designil.com/wireframe-website-free-download.html>
2. Vvkungx. 2016. Adobe เปิดตัว Experience Design เครื่องมือออกแบบ UX สำหรับนักพัฒนา. สืบค้นเมื่อ 12 มกราคม 2562, จาก <https://www.blognone.com/node/79081>
3. Ake Exorcist. 2017. มาเรียนรู้ Git แบบง่าย ๆ กันเถอะ. สืบค้นเมื่อ 12 มกราคม 2562, จาก <https://blog.nextzy.me/มาเรียนรู้-git-แบบง่าย ๆ กันเถอะ-427398e62f82>
4. Sarah K. 2018. Git Version Control series: What is Git?. สืบค้นเมื่อ 12 มกราคม 2562, จาก <https://blog.cpanel.com/git-version-control-series-what-is-git/>
5. Nuttavut Thongjor. 2016. รู้จัก Vue.js และคุณสมบัติใหม่ใน Vue 2. สืบค้นเมื่อ 12 มกราคม 2562, จาก <https://www.babelcoder.com/blog/posts/vue2-introduction-to-vue2>
6. Warat Wongmaneekeit. 2018. ทำไมผมถึงเลือก Vue.js ทั้งที่คนอื่น ๆ ใช้ React กัน. สืบค้นเมื่อ 12 มกราคม 2562, จาก <https://medium.com/skooldio/ทำไมผมถึงเลือก-vue-js-ทั้งที่คนอื่น ๆ ใช้-react-กัน-4020ffe8a186>
7. Scott Morris. 2018. What Is a JavaScript Framework? Here's Everything You Need to Know. สืบค้นเมื่อ 12 มกราคม 2562, จาก <https://skillcrush.com/2018/07/23/what-is-a-javascript-framework/>
8. Designil. 2015. SASS คืออะไร? เทคนิคช่วยให้การเขียน CSS เป็นเรื่องง่าย ๆ ใช้กันทั่วโลก !!. สืบค้นเมื่อ 13 มกราคม 2562, จาก <https://www.designil.com/sass-css-scss-tutorial.html>
9. The jQuery Team. jQuery API Documentation. สืบค้นเมื่อ 13 มกราคม 2562, จาก <http://api.jquery.com/>
10. Krist Wongsuphasawat. 2017. API คืออะไร? อธิบายแบบคนไม่เขียนโปรแกรมรู้เรื่องได้มั้ย?. สืบค้นเมื่อ 13 มกราคม 2562, จาก <https://medium.com/skooldio/api-คืออะไร-264ee4186f2c>