

การพัฒนาระบบจัดเก็บข้อมูลทรัพย์สิน
(กรณีศึกษา ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสยาม)

The Development of an Asset Management System

Case Study: Department of Computer Science, Siam University

นายชรินทร์	รัตนวราหะ	6104800002
นายพัชรพงศ์	มั่งมีธนกกาญจน์	6104800040

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยสยาม

ปีการศึกษา 2566

หัวข้อปริญญาโท

การพัฒนาระบบจัดเก็บข้อมูลทรัพย์สิน (กรณีศึกษา
ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสยาม)
The Development of an Asset Management System
Case Study: Department of Computer Science, Siam University

หน่วยกิตของปริญญาโท

3 หน่วยกิต

รายชื่อคณะผู้จัดทำ

นาย ชรินทร์ รัตนวราหะ 6104800002
นาย พชรพงศ์ มั่งมีธนาการบุญ 6104800040

อาจารย์ที่ปรึกษา

อาจารย์ ธนาภรณ์ รอดชีวิต

ระดับการศึกษา

วิทยาศาสตร์บัณฑิต

ภาควิชา


วิทยาการคอมพิวเตอร์


ปีการศึกษา

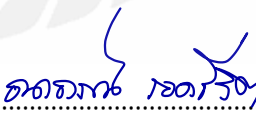
2566

อนุมัติให้ปริญญาโทนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์

คณะกรรมการสอบปริญญาโท


.....ประธานกรรมการ
(พล.อ.ท.ยศ.ดร. พาทรรณ สงวนโกศลย์)


.....กรรมการ
(อาจารย์ จรรยา แหยมเจริญ)


.....อาจารย์ที่ปรึกษา
(อาจารย์ ธนาภรณ์ รอดชีวิต)

หัวข้อปริญญานิพนธ์	การพัฒนาระบบจัดเก็บข้อมูลทรัพย์สิน (กรณีศึกษา ภาควิชา วิทยาการคอมพิวเตอร์ มหาวิทยาลัยสยาม)		
หน่วยกิตของปริญญานิพนธ์	3 หน่วยกิต		
รายชื่อคณะผู้จัดทำ	นาย ชรินทร์	รัตนวราหะ	6104800002
	นาย พชรพงศ์	มังมิธนาการญจน์	6104800040
อาจารย์ที่ปรึกษา	อาจารย์ ธนาภรณ์	รอดชีวิต	
ระดับการศึกษา	วิทยาศาสตรบัณฑิต		
ภาควิชา	วิทยาการคอมพิวเตอร์		
ปีการศึกษา	2566		

บทคัดย่อ

วัตถุประสงค์ในการจัดทำปริญญานิพนธ์นี้เพื่อพัฒนาระบบจัดเก็บข้อมูลทรัพย์สิน กรณีศึกษา ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสยาม เพื่อใช้สำหรับติดตามและเก็บข้อมูล รายการทรัพย์สิน ซึ่งเดิมทางภาควิชาไม่มีระบบในการจัดเก็บข้อมูลทรัพย์สินต่างๆ ทำให้ไม่สามารถตรวจสอบข้อมูลอุปกรณ์หรือเครื่องมือต่างๆ ได้ บางครั้งไม่ทราบว่าทรัพย์สินนั้นอยู่ที่ ตำแหน่งใดในภาควิชา หมาดการรับประกัน ไปหรือยัง หรือ ได้มีการส่งไปซ่อมที่ไหนหรือไม่ ก่อให้เกิดการสูญหายของทรัพย์สินตามมา ดังนั้นคณะผู้จัดทำจึงได้พัฒนาระบบจัดเก็บข้อมูล ทรัพย์สิน โดยพัฒนารูปแบบเว็บแอปพลิเคชัน และจัดเก็บข้อมูลลงฐานข้อมูล แบ่งผู้ใช้ออกเป็น 2 กลุ่ม ประกอบด้วย ผู้ใช้งาน สามารถเพิ่ม ลบ แก้ไข ค้นหา ข้อมูลทรัพย์สิน ตรวจสอบ แก้ไขสถานะ ของทรัพย์สิน ในส่วนของผู้ดูแลระบบ จะเพิ่มฟังก์ชันการเพิ่มผู้ใช้งาน และการออกรายงานรายการ ทรัพย์สิน ในการออกแบบส่วนติดต่อกับผู้ใช้ทางคณะผู้จัดทำได้ออกแบบตามหลักการของ UX และ UI เครื่องมือที่ใช้ในการพัฒนา ได้แก่ Visual Studio Code เขียนชุดคำสั่งด้วยภาษา JavaScript และ Typescript บริหารจัดการฐานข้อมูลด้วยโปรแกรม HeidiSQL โดยใช้ฐานข้อมูล PostgreSQL ด้วยภาษา SQL โดยระบบสามารถช่วยให้การดำเนินงานสะดวกสบาย แม่นยำ รวดเร็ว ส่งผลให้การ ทำงานมีประสิทธิภาพเพิ่มมากขึ้นในการจัดการทรัพย์สินภายในภาควิชามากยิ่งขึ้นอีกด้วย



คำสำคัญ: เว็บแอปพลิเคชัน, ทรัพย์สิน, ภาควิชาวิทยาการคอมพิวเตอร์

Project title	The Development of an Asset Management System		
	Case Study: Department of Computer Science, Siam University		
Project credits	3 Units		
By	Mr. Charinthon	Rattanawaraha	6104800002
	Mr. Patcharapong	Mungmeethanakan	6104800040
Advisor	Miss Thanaporn	Rodcheewit	
Degree	Bachelor of Science		
Major	Computer Science		
Faculty	Science		
Academic year	2023		

Abstract

The objective of this thesis is to develop an asset management system for the Computer Science Department at Siam University. The system is designed to track and store information about the department's assets, addressing the lack of an existing system for managing such data. Previously, the department faced challenges in locating equipment or tools, determining warranty status, and tracking repairs, which often led to asset loss. To address these issues, the project team developed a web application for asset management, with data stored in a database. Users are categorized into two groups: general users, who can add, delete, edit, and search for asset information, and administrators, who have additional functionalities for adding users and generating asset reports. The user interface was designed according to UX and UI principles. The development tools used include Visual Studio Code, with JavaScript and TypeScript for coding, and HeidiSQL for database management using PostgreSQL and SQL. The system is intended to improve convenience, accuracy, and speed in asset management, ultimately enhancing the department's operational efficiency.

Keywords: web applications, asset, department of computer science

 (Project-Advisor)	Approved by 
--	--

กิตติกรรมประกาศ (Acknowledgment)

การจัดทำวิทยานิพนธ์ฉบับนี้สำเร็จได้นั้น คณะผู้จัดทำได้รับความกรุณาจาก อาจารย์ผู้สอนทุกท่านที่ให้ข้อมูลต่าง ๆ ส่งผลให้คณะผู้จัดทำได้รับความรู้และประสบการณ์ต่าง ๆ ที่มีค่ามากมายสำหรับวิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดีจากความร่วมมือและสนับสนุนจากหลายฝ่ายดังนี้

1. อาจารย์ ชนาภรณ์ รอดชีวิต อาจารย์ที่ปรึกษา

คณะผู้จัดทำใคร่ขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ ที่ได้ให้คำแนะนำสำคัญเพื่อให้การสอบวิทยานิพนธ์ฉบับนี้ และผู้มีส่วนร่วมทุกท่าน รวมทั้งผู้ที่ไม่ได้กล่าวนาม ที่มีส่วนร่วมในการให้ข้อมูลให้ความช่วยเหลือ และเป็นทีปรึกษาให้คำแนะนำต่าง ๆ จนทำให้งานทุกอย่างประสบความสำเร็จไปด้วยดี และทำรายงานฉบับนี้จนเสร็จสมบูรณ์ ซึ่งคณะผู้จัดทำขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ด้วย

คณะผู้จัดทำ

นายชรินทร์ รัตนวราหะ

นายพัชรพงศ์ มั่งมีชนการญจน์

สารบัญ

	หน้า
บทคัดย่อ.....	ก
Abstract.....	ข
กิตติกรรมประกาศ.....	ค
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของปริญญาานิพนธ์.....	1
1.3 ขอบเขตของปริญญาานิพนธ์.....	2
1.4 ประโยชน์ที่ได้รับ.....	2
1.5 ขั้นตอนและวิธีดำเนินงานปริญญาานิพนธ์.....	3
1.6 ขั้นตอนและระยะเวลาในการดำเนินปริญญาานิพนธ์.....	5
1.7 อุปกรณ์และเครื่องมือที่ใช้ในการพัฒนาระบบ.....	5
1.8 อุปกรณ์และเครื่องมือที่รองรับ.....	6
บทที่ 2 การทบทวนเอกสารและวรรณกรรมที่เกี่ยวข้อง	
2.1 ระบบฐานข้อมูลเชิงสัมพันธ์.....	7
2.2 สถาปัตยกรรมไคลแอนท์/ เซิร์ฟเวอร์.....	9
2.3 เว็บแอปพลิเคชัน.....	9
2.4 HeidiSQL.....	11
2.5 ระบบฐานข้อมูลรูปแบบ PostgreSQL.....	12
2.6 Visual Studio Code.....	13
2.7 Docker Desktop	14
2.8 Figma	15
2.9 Bootstrap Vue.....	16
2.10 Nuxt js.....	17
บทที่ 3 การวิเคราะห์และออกแบบระบบ	
3.1 วิเคราะห์ระบบงานปัจจุบัน.....	18
3.2 วิเคราะห์ระบบงานใหม่.....	19
3.3 แสดงฟังก์ชันการทำงานของระบบด้วย Use Case Diagram.....	20
3.4 Sequence Diagram.....	29
3.5 Class Diagram.....	37

สารบัญ (ต่อ)

	หน้า
3.6 โครงสร้างฐานข้อมูลและความสัมพันธ์ระหว่างเอนทิตี.....	38
บทที่ 4 การออกแบบทางกายภาพ	
4.1 การออกแบบฐานข้อมูล	40
4.2 แผนผังโครงสร้างของเว็บไซต์.....	45
4.3 รายละเอียดของโครงสร้างเว็บไซต์.....	46
4.4 การออกแบบส่วนติดต่อผู้ใช้งาน (User Interface Design)	47
บทที่ 5 สรุปผลและข้อเสนอแนะ	
5.1 สรุปผลปริญญานิพนธ์.....	59
5.2 ข้อดีของระบบ.....	59
5.3 ข้อเสนอแนะ.....	60
บรรณานุกรม.....	61

สารบัญตาราง

	หน้า
ตารางที่ 1.1 ขั้นตอนและระยะเวลาในการดำเนินงาน.....	5
ตารางที่ 3.1 แสดงรายละเอียดของ Use case: Login.....	21
ตารางที่ 3.2 แสดงรายละเอียดของ Use case: View user.....	21
ตารางที่ 3.3 แสดงรายละเอียดของ Use case: Add User.....	22
ตารางที่ 3.4 แสดงรายละเอียดของ Use case: Edit User.....	22
ตารางที่ 3.5 แสดงรายละเอียดของ Use case: View Product.....	23
ตารางที่ 3.6 แสดงรายละเอียดของ Use case: Add Room.....	23
ตารางที่ 3.7 แสดงรายละเอียดของ Use case: Add Category.....	24
ตารางที่ 3.8 แสดงรายละเอียดของ Use case: View Maintenance.....	24
ตารางที่ 3.9 แสดงรายละเอียดของ Use case: Edit Maintenance.....	25
ตารางที่ 3.10 แสดงรายละเอียดของ Use case: View Expire.....	25
ตารางที่ 3.11 แสดงรายละเอียดของ Use case: Edit Expire.....	26
ตารางที่ 3.12 แสดงรายละเอียดของ Use case: View Damaged.....	26
ตารางที่ 3.13 แสดงรายละเอียดของ Use case: Edit Damaged.....	27
ตารางที่ 3.14 แสดงรายละเอียดของ Use case: View Report.....	27
ตารางที่ 3.15 แสดงรายละเอียดของ Use case: Export Excel.....	28
ตารางที่ 4.1 รายละเอียดของตารางข้อมูล Register.....	39
ตารางที่ 4.2 รายละเอียดของตารางข้อมูล Room.....	40
ตารางที่ 4.3 รายละเอียดของตารางข้อมูล Status.....	40
ตารางที่ 4.4 รายละเอียดของตารางข้อมูล Category.....	40
ตารางที่ 4.5 รายละเอียดของตารางข้อมูล Product.....	41
ตารางที่ 4.6 รายละเอียดของตารางข้อมูล Repair.....	42
ตารางที่ 4.7 รายละเอียดของตารางข้อมูล Expired.....	43
ตารางที่ 4.8 รายละเอียดของตารางข้อมูล Damaged.....	44
ตารางที่ 4.9 รายละเอียดของโครงสร้างของเว็บไซต์.....	46

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 Relational Database.....	7
รูปที่ 2.2 รูปแบบสถาปัตยกรรมไมโครเอ็นท์/เซิร์ฟเวอร์.....	9
รูปที่ 2.3 ลักษณะของเว็บแอปพลิเคชัน.....	10
รูปที่ 2.4 สัญลักษณ์ HeidiSQL	12
รูปที่ 2.5 ตัวอย่างหน้าจอ HeidiSQL.....	12
รูปที่ 2.6 สัญลักษณ์โปรแกรม Visual Studio Code.....	13
รูปที่ 2.7 ตัวอย่างหน้าจอ โปรแกรม Visual Studio Code.....	14
รูปที่ 2.8 สัญลักษณ์ Docker Desktop.....	14
รูปที่ 2.9 ตัวอย่างโปรแกรม Docker Desktop.....	15
รูปที่ 2.10 สัญลักษณ์ Figma.....	15
รูปที่ 2.11 ตัวอย่างหน้าจอ Figma.....	16
รูปที่ 2.12 สัญลักษณ์ Bootstrap Vue.....	16
รูปที่ 2.13 สัญลักษณ์ Nuxt.js.....	17
รูปที่ 3.1 Work Flow Diagram ของระบบจัดเก็บข้อมูลทรัพย์สิน.....	19
รูปที่ 3.2 Use Case Diagram ของระบบจัดเก็บข้อมูลทรัพย์สิน.....	29
รูปที่ 3.3 Sequence diagram : Login.....	29
รูปที่ 3.4 Sequence Diagram: Add user.....	29
รูปที่ 3.5 Sequence Diagram: Edit user.....	30
รูปที่ 3.6 Sequence Diagram: View user.....	30
รูปที่ 3.7 Sequence Diagram: View Product.....	31
รูปที่ 3.8 Sequence Diagram: Add Room.....	31
รูปที่ 3.9 Sequence Diagram: Add Category.....	32
รูปที่ 3.10 Sequence Diagram: View Maintenance.....	32
รูปที่ 3.11 Sequence Diagram: Edit Maintenance.....	33
รูปที่ 3.12 Sequence Diagram: View Expire.....	33
รูปที่ 3.13 Sequence Diagram: Edit Expire.....	34
รูปที่ 3.14 Sequence Diagram: View Damaged.....	34
รูปที่ 3.15 Sequence Diagram: Edit Damaged.....	35

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 3.16 Sequence Diagram: View Report.....	35
รูปที่ 3.17 Sequence Diagram: Export Excel.....	36
รูปที่ 3.18 Class diagram ระบบจัดเก็บข้อมูลทรัพย์สิน.....	37
รูปที่ 3.19 Entity Relationship Diagram ของระบบจัดเก็บข้อมูลทรัพย์สิน.....	38
รูปที่ 4.1 แผนผังโครงสร้างของเว็บไซต์.....	45
รูปที่ 4.2 หน้าเข้าสู่ระบบ (Login).....	47
รูปที่ 4.3 หน้าแรก (Dashboard).....	48
รูปที่ 4.4 หน้ารายการทรัพย์สิน (Products).....	48
รูปที่ 4.5 หน้าฟอร์มเพิ่มรายการทรัพย์สิน (Add Products).....	49
รูปที่ 4.6 หน้าฟอร์มเพิ่มประเภททรัพย์สิน (Add Category).....	49
รูปที่ 4.7 หน้าฟอร์มเพิ่มรหัสห้อง (Add Room).....	50
รูปที่ 4.8 หน้าฟอร์มแก้ไขข้อมูลทรัพย์สิน (Edit Product).....	50
รูปที่ 4.9 หน้าแสดงข้อมูลทรัพย์สินที่มีสถานะหมดอายุ/หมดประกัน (Expired).....	51
รูปที่ 4.10 หน้าฟอร์มแก้ไขข้อมูลทรัพย์สิน (Edit Product).....	51
รูปที่ 4.11 หน้าแสดงข้อมูลทรัพย์สินที่มีสถานะรอซ่อม/ ส่งซ่อม (Maintain).....	52
รูปที่ 4.12 หน้าฟอร์มแก้ไขข้อมูลทรัพย์สิน (Edit Product).....	52
รูปที่ 4.13 หน้าแสดงข้อมูลทรัพย์สินที่มีสถานะชำรุดเสียหาย (Damaged).....	53
รูปที่ 4.14 หน้าฟอร์มแก้ไขข้อมูลทรัพย์สิน (Edit Product).....	53
รูปที่ 4.15 หน้าแสดงข้อมูลผู้ใช้ (Manege Accounts).....	54
รูปที่ 4.16 หน้าฟอร์มเพิ่มข้อมูลผู้ใช้ (Add Account)	54
รูปที่ 4.17 หน้าฟอร์มแก้ไขข้อมูลผู้ใช้ (Edit Account).....	55
รูปที่ 4.18 หน้าออกรายงานทรัพย์สิน (Reports)	55
รูปที่ 4.19 หน้าฟอร์มปริ๊นและออกรายงานไฟล์ Excel (Export Excel).....	56
รูปที่ 4.20 หน้าเว็บเมื่อเข้าสู่ระบบด้วยบัญชีที่เป็น User	56
รูปที่ 4.21 หน้าเว็บเมื่อเข้าสู่ระบบด้วยบัญชีที่เป็น Administrator.....	57
รูปที่ 4.22 หน้าต่างป๊อปอัพเพื่อยืนยันค่าขอลบข้อมูลในหน้า Products.....	57
รูปที่ 4.23 หน้าต่างป๊อปอัพเพื่อยืนยันค่าขอลบข้อมูลในหน้า Manage Account.....	58

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ภาควิชาวิทยาการคอมพิวเตอร์เป็นสาขาวิชาที่อยู่ภายใต้คณะวิทยาศาสตร์ มหาวิทยาลัยสยาม เป็นสาขาวิชาที่ให้ความรู้ทางด้านวิทยาการข้อมูลและคอมพิวเตอร์ ภายในภาควิชาประกอบด้วยอาจารย์ เจ้าหน้าที่ และนักศึกษา ซึ่งในการดำเนินงานมีการใช้อุปกรณ์หรือเครื่องมือต่างๆ ที่เป็นทรัพย์สินภายในภาควิชาเพื่อสนับสนุนในการทำงานและการเรียนการสอน โดยในปัจจุบันทางภาควิชาไม่ได้มีการบันทึกข้อมูลทรัพย์สินต่างๆ ที่มีการสั่งซื้อเข้ามาใหม่ หรือรวมถึงอุปกรณ์ที่หมดการรับประกัน ชำรุด ทำให้ไม่สามารถตรวจสอบข้อมูลอุปกรณ์หรือเครื่องมือต่างๆ ได้ว่าในภาควิชานั้นมีทรัพย์สินอะไรอยู่บ้าง และบางครั้งไม่ทราบว่าทรัพย์สินนั้นอยู่ที่ตำแหน่งใดในภาควิชา หมดการรับประกันไปหรือไม่ หรือได้มีการส่งไปซ่อมที่ไหนหรือไม่ เกิดความเสี่ยงการสูญหายของทรัพย์สิน และไม่มีการแจ้งสถานะของทรัพย์สินทำให้ไม่ทราบว่าในภาควิชามีทรัพย์สินอะไรอยู่บ้าง หรือยังสามารถใช้งานได้ตามปกติหรือไม่

จากปัญหาดังกล่าวคณะผู้จัดทำจึงได้ศึกษาและคิดพัฒนาระบบจัดเก็บข้อมูลทรัพย์สินกรณีศึกษา ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสยาม ในรูปแบบเว็บแอปพลิเคชัน (Web Application) มีการเก็บข้อมูลด้วยเทคโนโลยีฐานข้อมูล (Database) และผู้ใช้งานสามารถบันทึกแก้ไข ข้อมูลทรัพย์สินได้ผ่านทางเว็บแอปพลิเคชันนี้ผ่านระบบเครือข่ายอินเทอร์เน็ต (Internet) และโปรแกรมเว็บเบราว์เซอร์ (Web Browser) ทำให้ผู้ใช้งานสามารถดำเนินการได้อย่างสะดวกสบาย แม่นยำ รวดเร็ว ช่วยให้การดำเนินงานมีประสิทธิภาพในการจัดการทรัพย์สินภายในภาควิชาวิทยาการคอมพิวเตอร์มากยิ่งขึ้นอีกด้วย

โดยระบบจัดเก็บข้อมูลทรัพย์สิน กรณีศึกษา ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสยาม แบ่งผู้ใช้ออกเป็น 2 กลุ่ม ประกอบด้วย ผู้ใช้งาน สามารถเพิ่ม ลบ แก้ไข ค้นหา ข้อมูลทรัพย์สิน ตรวจสอบ แก้ไขสถานะของทรัพย์สิน ในส่วนของผู้ดูแลระบบ จะเพิ่มฟังก์ชันการเพิ่มผู้ใช้งาน และการออกรายงานรายการทรัพย์สิน ในการออกแบบส่วนติดต่อกับผู้ใช้ทางคณะผู้จัดทำได้ออกแบบตามหลักการของ UX และ UI เครื่องมือที่ใช้ในการพัฒนา ได้แก่ Visual Studio Code และเขียนชุดคำสั่งด้วยภาษา JavaScript และ Typescript จำลองระบบฐานข้อมูลด้วย Docker Desktop จัดการฐานข้อมูลด้วย PostgreSQL ด้วยโปรแกรม HeidiSQL ตกแต่งหน้าเว็บไซต์ด้วย Bootstrap Vue, Nuxt.js

1.2 วัตถุประสงค์ของปฏิญานิพนธ์

เพื่อพัฒนาระบบจัดเก็บข้อมูลทรัพย์สิน กรณีศึกษา ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสยาม

1.3 ขอบเขตปฏิญานิพนธ์

- 1.3.1 สถาปัตยกรรมที่ใช้ในการพัฒนาปฏิญานิพนธ์เป็นแบบไคลเอนท์/ เซิร์ฟเวอร์
- 1.3.2 การพัฒนาซอฟต์แวร์บนแพลตฟอร์มเว็บแอปพลิเคชัน
- 1.3.3 จัดการฐานข้อมูลด้วย PostgreSQL
- 1.3.4 กลุ่มผู้ใช้งานประกอบด้วย 2 กลุ่ม ดังนี้
 - 1.3.4.1 ผู้ดูแลระบบ (Administrator)
 - 1.3.4.1.1 สามารถล็อกอินผ่าน Email
 - 1.3.4.1.2 สามารถเพิ่ม แก้ไข ลบ ข้อมูลผู้ใช้งาน ได้
 - 1.3.4.1.3 สามารถเพิ่ม แก้ไข ลบ ข้อมูลทรัพย์สินได้
 - 1.3.4.1.4 สามารถตรวจสอบ แก้ไข เพิ่ม ลบ ข้อมูลทั้งหมดในฐานข้อมูล
 - 1.3.4.1.5 สามารถออกรายงานทรัพย์สินตามประเภทต่างๆ รูปแบบไฟล์ Excel
 - 1.3.4.2 ผู้ใช้งาน (อาจารย์ และเจ้าหน้าที่)
 - 1.3.4.2.1 สามารถล็อกอินผ่าน Email
 - 1.3.4.2.2 สามารถดูรายละเอียดทรัพย์สินทั้งหมดได้
 - 1.3.4.2.3 สามารถอัปเดตสถานะของทรัพย์สินได้
 - 1.3.4.2.4 สามารถเพิ่ม ลบ แก้ไขข้อมูล ของทรัพย์สินได้
 - 1.3.4.2.6 สามารถค้นหาทรัพย์สินได้
 - 1.3.4.2.7 สามารถดูข้อมูลทรัพย์สินของห้องเรียนแต่ละห้องได้
 - 1.3.4.2.9 สามารถเพิ่มรหัสห้องเรียน ประเภททรัพย์สินได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 ช่วยให้มี ความถูกต้องและความน่าเชื่อถือของข้อมูลทรัพย์สิน
- 1.4.2 ช่วยลดความเสี่ยงในการสูญหายหรือเสียหายของทรัพย์สิน
- 1.4.3 ช่วยให้สามารถบำรุงรักษาและตรวจสอบทรัพย์สินอย่างสม่ำเสมอ
- 1.4.4 ช่วยให้ทราบสถานะและสถานที่ของทรัพย์สินในแต่ละช่วงเวลา

1.5 ขั้นตอนและวิธีดำเนินงานปริญญานิพนธ์

1.5.1 รวบรวมความต้องการและศึกษาข้อมูล (Detailed Study)

คณะผู้จัดทำได้สำรวจและรวบรวมข้อมูลเกี่ยวกับปัญหาที่เกิดขึ้นในการจัดการทรัพย์สินภายในภาควิชาวิทยาการคอมพิวเตอร์ เช่น การสูญหายของทรัพย์สิน การเสื่อมสภาพและหมดอายุรับประกันของอุปกรณ์ต่างๆ มีการประชุมและสัมภาษณ์กับอาจารย์ เพื่อรวบรวมความต้องการและความคิดเห็นในการพัฒนาระบบ นอกจากนี้ยังศึกษาระบบที่มีอยู่แล้วในตลาดหรือระบบที่คล้ายคลึงกันเพื่อนำมาประยุกต์ใช้ในการพัฒนาเว็บจัดการทรัพย์สินของภาควิชา

1.5.2 วิเคราะห์ระบบงาน (System Analysis)

โดยเริ่มจากการวิเคราะห์ข้อมูลที่ได้รับจากการศึกษาข้อมูลและรวบรวมความต้องการเพื่อกำหนดขอบเขตและคุณลักษณะของเว็บแอปพลิเคชันจัดการทรัพย์สิน จากนั้นออกแบบระบบงานเบื้องต้นโดยใช้เครื่องมือและเทคนิคต่างๆ เช่น การสร้าง Use Case Diagram, Class Diagram และ Sequence Diagram เพื่อแสดงภาพรวมของระบบงานและยังประเมินและเลือกเทคโนโลยีที่เหมาะสมในการพัฒนาระบบจัดเก็บข้อมูลทรัพย์สิน เช่น การเลือกภาษาโปรแกรมเครื่องมือที่ใช้ในการพัฒนาชุดคำสั่ง และฐานข้อมูล

1.5.3 ออกแบบระบบงาน (System Design)

ในขั้นตอนนี้จะทำการออกแบบแอปพลิเคชันเพื่อให้สามารถนำไปใช้งานได้จริงและเป็นแนวทางในการพัฒนาระบบต่อไป

1.5.3.1 ออกแบบฐานข้อมูล (Database) โดยใช้รูปแบบ HeidiSQL ผ่าน Prisma เพื่อจัดเก็บข้อมูลของผู้ใช้งาน ข้อมูลทรัพย์สิน ข้อมูลสถานะ ข้อมูลรายงาน

1.5.3.2 ออกแบบหน้าจอส่วนติดต่อผู้ใช้ (User Interface Design)

ออกแบบส่วนติดต่อกับผู้ใช้หรือหน้าจอให้สามารถกรอกข้อมูลส่งข้อมูลไปยังระบบฐานข้อมูลได้ถูกต้อง ครบถ้วน มีการทำงานที่ชัดเจน ไม่ซับซ้อน เพื่อง่ายต่อความเข้าใจของผู้ใช้งาน

1.5.3.3 กำหนดเครื่องมือในการพัฒนาระบบ

1.5.3.3.1 โปรแกรม Visual Studio Code นำมาใช้ในการรัน Code และทดสอบระบบการจัดการข้อมูล

1.5.3.3.2 โปรแกรม Docker Desktop นำมาใช้ทำงานในลักษณะจำลองสภาพแวดล้อมขึ้นมาบนเครื่อง Server เพื่อใช้ในการ run service

1.5.3.3.3 โปรแกรม Figma นำมาใช้ออกแบบ UX/ UI หน้าตาของแอปพลิเคชัน

1.5.3.3.4 โปรแกรม Prisma นำมาใช้ในการบริหารจัดการเก็บข้อมูลต่างๆ

1.5.3.3.5 โปรแกรม HeidiSQL มีคุณสมบัติในการดึง แก้ไขข้อมูล สร้างคำสั่ง SQL และการจัดการ โครงสร้างฐานข้อมูลต่างๆ

1.5.4 พัฒนาระบบ (System Development)

ขั้นตอนการพัฒนาระบบ (System Development) เริ่มต้นด้วยการใช้ Docker Desktop และจัดการฐานข้อมูล PostgreSQL ด้วย HeidiSQL และ Prisma ในการเชื่อมต่อและจัดการข้อมูล การพัฒนาเว็บไซต์ใช้ Nuxt.js และ Bootstrap Vue สำหรับสร้างเว็บแอปพลิเคชัน ใช้ VS Code เป็นเครื่องมือหลักในการเขียนชุดคำสั่ง และออกแบบ UI/UX ด้วย Figma การทดสอบ API ใช้ Postman เพื่อทดสอบการทำงานของ API และการสื่อสารระหว่างฟรอนต์เอนด์และแบ็กเอนด์ การพัฒนาระบบดำเนินไปตามขั้นตอนที่กำหนด โดยมีการทดสอบและปรับปรุงอย่างต่อเนื่อง เพื่อให้มั่นใจในประสิทธิภาพและความถูกต้องของระบบ

1.5.5 ทดสอบระบบ (System Testing)

คณะผู้จัดทำได้ทำการทดสอบระบบนี้ โดยการทดสอบฟังก์ชันแต่ละส่วนของเว็บไซต์ เช่น การเข้าสู่ระบบ การจัดการทรัพย์สิน การออกรายงาน และระบบการแจ้งเตือน เพื่อให้แน่ใจว่าทุกฟังก์ชันทำงานถูกต้องตามที่กำหนดการทดสอบนี้จะใช้เครื่องมือ Postman เพื่อสร้างและทดสอบ API ทุกส่วน และทำการทดสอบระบบทั้งหมดภายใต้สถานการณ์ที่เป็นไปได้ต่างๆ เพื่อตรวจสอบความเสถียรและประสิทธิภาพของระบบ หลังจากที่ผ่านมาการทดสอบ และได้รับการยืนยันว่าระบบทำงานได้อย่างถูกต้อง และประสิทธิภาพสูง ขั้นตอนต่อไปคือการปรับปรุงแก้ไขตามผลการทดสอบ (Bug Fixing) และเตรียมตัวสำหรับการใช้งานจริง

1.5.6 จัดทำเอกสารประกอบปริญญานิพนธ์ (Documentation)

เป็นการจัดทำเอกสารแนวทางประกอบปริญญานิพนธ์ วิธีการและขั้นตอนการดำเนินปริญญานิพนธ์ เพื่อนำเสนอรายงานต่ออาจารย์ที่ปรึกษา และคณะกรรมการสอบปริญญานิพนธ์ เพื่อขอรับคำแนะนำ และนำไปใช้เป็นเอกสารอ้างอิงในอนาคต

1.6 ขั้นตอนและระยะเวลาในการดำเนินงานปริญญานิพนธ์

ตารางที่ 1.1 ขั้นตอนและระยะเวลาในการดำเนินงาน

ขั้นตอนการดำเนินงาน	2567						
	ม.ค.	ก.พ.	มี.ค.	เม.ย	พ.ค.	มิ.ย.	ก.ค.
1. รวบรวมความต้องการ	↔						
2. วิเคราะห์ระบบ		↔	↔				
3. ออกแบบระบบ			↔				
4. พัฒนาระบบ				↔	↔	↔	
5. ทดสอบระบบ							↔
6. จัดทำเอกสาร							↔

1.7 อุปกรณ์และเครื่องมือที่ใช้ในการพัฒนา

1.7.1 ฮาร์ดแวร์ (Hardware)

1.7.1.1 เครื่องคอมพิวเตอร์

- Intel(R) Core (TM) i5-10400F
- NVIDIA GeForce RTX 3060 Ti
- RAM 32 GB
- 500 GB SSD PCIe M.2

1.7.2 ซอฟต์แวร์ (Software)

1.7.2.1 โปรแกรม Visual Studio Code

1.7.2.2 โปรแกรม Prisma

1.7.2.3 โปรแกรม Figma

1.7.2.4 โปรแกรม HeidiSQL

1.7.2.5 เว็บเบราว์เซอร์ Google Chrome

1.7.2.6 โปรแกรม Docker Desktop

1.8 อุปกรณ์และเครื่องมือที่รองรับระบบ

1.8.1 ฮาร์ดแวร์ (Hardware)

1.8.1.1 Computer

- CPU Core I3
- Ram 8 GB
- VGA Onboard
- พื้นที่เก็บข้อมูล 240 Gb

1.8.2 ซอฟต์แวร์ (Software)

- 1.8.2.1 ระบบปฏิบัติการ OS Windows 7
- 1.8.2.2 เว็บเบราว์เซอร์ Google Chrome
- 1.8.2.3 เว็บเบราว์เซอร์ Microsoft Edge

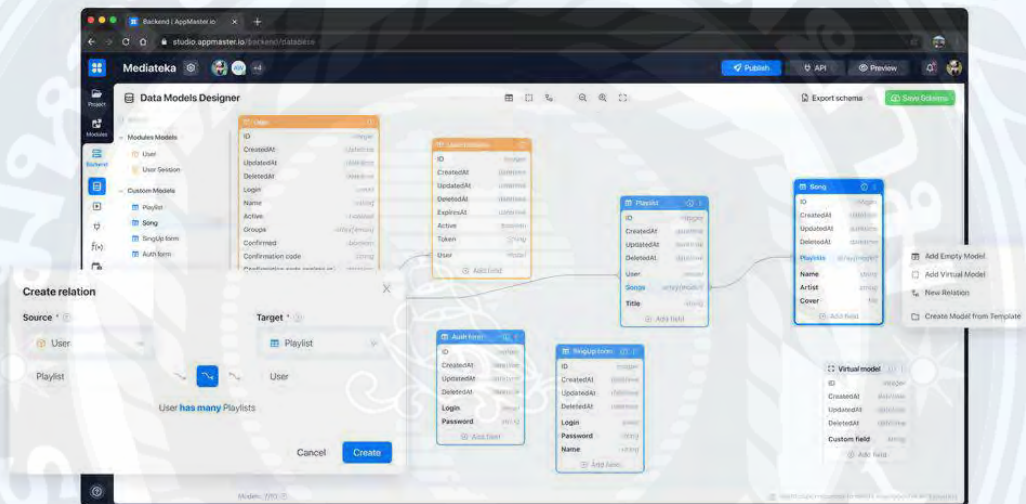


บทที่ 2

การทบทวนวรรณกรรมที่เกี่ยวข้อง

การพัฒนาเว็บแอปพลิเคชันที่เก็บข้อมูลทรัพย์สินทางปัญญา มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสยาม คณะผู้จัดทำได้ทำการการศึกษาข้อมูล แนวคิดและทฤษฎีที่เกี่ยวข้อง เพื่อเป็นกรอบแนวทางในการพัฒนาระบบ ซึ่งสามารถแบ่งเป็นหัวข้อ โดยมีรายละเอียด ดังต่อไปนี้

2.1 ระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database)¹



รูปที่ 2.1 Relational Database

ฐานข้อมูลเชิงสัมพันธ์มีโครงสร้างโดยใช้ตาราง ซึ่งเรียกอีกอย่างว่าความสัมพันธ์ แต่ละตารางประกอบด้วยแถวและคอลัมน์ โดยแต่ละแถวแสดงถึงระเบียบหรืออินสแตนซ์ของข้อมูล และแต่ละคอลัมน์จะแสดงถึงแอตทริบิวต์หรือฟิลด์เฉพาะของข้อมูล ชุดของแอตทริบิวต์หรือชนิดข้อมูล เช่น ชื่อ ความยาว วันที่ หรือบูลีน กำหนดคอลัมน์ในตาราง แต่ละคอลัมน์ยังมีชื่อเฉพาะซึ่งช่วยระบุประเภทข้อมูลที่จัดเก็บไว้ในคอลัมน์นั้น

แถวในตารางแสดงถึงแต่ละระเบียบหรืออินสแตนซ์ของข้อมูล แต่ละแถวมีตัวระบุที่ไม่ซ้ำกัน ซึ่งเรียกว่าคีย์หลัก คีย์หลักใช้เพื่อเชื่อมโยงระเบียบระหว่างตารางต่างๆ ในฐานข้อมูล ความสัมพันธ์ระหว่างตารางในฐานข้อมูลเชิงสัมพันธ์ถูกกำหนดโดยใช้คีย์นอก Foreign Key คือ

¹ <https://appmaster.io/th/blog/thaankh-muulechingsamphanthkhuue-aair>

คอลัมน์ในตารางหนึ่งอ้างอิงถึงคีย์หลักของตารางอื่น ซึ่งช่วยให้สามารถเชื่อมโยงและเข้าถึงข้อมูลที่เกี่ยวข้องจากตารางต่างๆ ในฐานข้อมูลได้

นอกจากตารางแล้ว ฐานข้อมูลเชิงสัมพันธ์ยังใช้ข้อจำกัดเพื่อให้อุ่นใจถึงความสมบูรณ์และความสอดคล้องของข้อมูลอีกด้วย ข้อจำกัดคือกฎหรือเงื่อนไขที่ต้องปฏิบัติตามก่อนที่จะสามารถแทรก ปรับปรุง หรือลบข้อมูลออกจากฐานข้อมูลได้ ตัวอย่างของข้อจำกัด ได้แก่ คีย์หลัก คีย์นอก ข้อจำกัดเฉพาะ และข้อจำกัดการตรวจสอบ

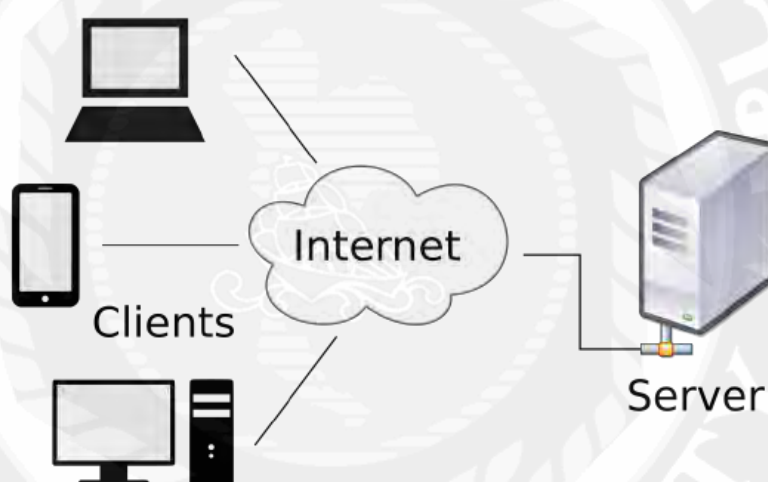
ประโยชน์ของระบบจัดการฐานข้อมูลเชิงสัมพันธ์

ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS) ให้ประโยชน์มากมาย ซึ่งบางส่วนรวมถึงต่อไปนี้:

- **ความสมบูรณ์ของข้อมูล** : RDBMS ใช้ข้อจำกัดต่างๆ เช่น คีย์หลัก คีย์นอก และข้อจำกัดในการตรวจสอบเพื่อให้แน่ใจว่าข้อมูลมีความสมบูรณ์ ซึ่งจะช่วยรักษาความถูกต้องและความสอดคล้องของข้อมูล
- **ความสามารถในการปรับขนาด** : RDBMS สามารถจัดการข้อมูลจำนวนมากและสามารถเพิ่มหรือลดขนาดได้อย่างง่ายดายตามต้องการ นอกจากนี้ยังสามารถรองรับผู้ใช้และแอปพลิเคชันได้หลายคนพร้อมกัน
- **ความยืดหยุ่น** : RDBMS มอบวิธีการที่ยืดหยุ่นในการจัดระเบียบและจัดเก็บข้อมูล เนื่องจากสามารถเพิ่ม ลบ หรือแก้ไขตารางเพื่อรองรับความต้องการข้อมูลที่เปลี่ยนแปลงได้
- **ใช้งานง่าย** : ภาษา SQL ที่ใช้ใน RDBMS นั้นง่ายต่อการเรียนรู้และใช้งาน และมีวิธีมาตรฐานและสอดคล้องกันในการโต้ตอบกับฐานข้อมูล
- **ความปลอดภัยของข้อมูล** : RDBMS มีคุณลักษณะด้านความปลอดภัยในตัว เช่น การควบคุมการเข้าถึงและการรับรองความถูกต้องของผู้ใช้ เพื่อให้แน่ใจว่าเฉพาะผู้ใช้ที่ได้รับอนุญาตเท่านั้นที่สามารถเข้าถึงและแก้ไขข้อมูลได้
- **ความสอดคล้องของข้อมูล** : RDBMS ใช้ธุรกรรมเพื่อให้แน่ใจว่าข้อมูลมีความสอดคล้องและเชื่อถือได้ แม้ในระหว่างที่ระบบล้มเหลวหรือหยุดชะงัก
- **การแบ่งปันข้อมูล** : RDBMS สามารถแบ่งปันข้อมูลระหว่างแอปพลิเคชันและแพลตฟอร์มต่างๆ ปรับปรุงการทำงานร่วมกันและประสิทธิภาพการทำงานในองค์กร

2.2 สถาปัตยกรรมไคลเอนต์/ เซิร์ฟเวอร์ (Client/ Server Architecture)²

สถาปัตยกรรมไคลเอนต์/ เซิร์ฟเวอร์ (Client/ Server Architecture) คือรูปแบบการออกแบบระบบที่แบ่งการทำงานออกเป็นสองส่วนหลักๆ คือ ไคลเอนต์ (Client) ซึ่งเป็นผู้ใช้หรือแอปพลิเคชันที่ร้องขอบริการหรือข้อมูล และเซิร์ฟเวอร์ (Server) ซึ่งเป็นผู้ให้บริการหรือจัดการข้อมูล กระบวนการทำงานเริ่มจากไคลเอนต์ส่งคำร้องขอไปยังเซิร์ฟเวอร์ จากนั้นเซิร์ฟเวอร์จะประมวลผลคำร้องขอและส่งคำตอบกลับไปยังไคลเอนต์ สถาปัตยกรรมนี้มีประโยชน์ในการแบ่งหน้าที่ชัดเจน ง่ายต่อการพัฒนาและดูแล สามารถขยายตัว (Scalability) ได้โดยเพิ่มจำนวนไคลเอนต์หรือเซิร์ฟเวอร์ และช่วยให้เซิร์ฟเวอร์จัดการทรัพยากรได้อย่างมีประสิทธิภาพ ตัวอย่างการใช้งานที่พบได้บ่อย ได้แก่ เว็บแอปพลิเคชันที่มีเบราว์เซอร์เป็นไคลเอนต์ และเว็บเซิร์ฟเวอร์ หรือระบบฐานข้อมูลที่มีแอปพลิเคชันเป็นไคลเอนต์และฐานข้อมูลเซิร์ฟเวอร์ สถาปัตยกรรมนี้เป็นพื้นฐานสำคัญของการพัฒนาแอปพลิเคชันในเครือข่ายและการบริการข้อมูลทางอินเทอร์เน็ต



รูปที่ 2.2 รูปแบบสถาปัตยกรรมไคลเอนต์/ เซิร์ฟเวอร์

2.3 เว็บแอปพลิเคชัน (Web Application)³

เว็บแอปพลิเคชัน คือซอฟต์แวร์ที่ทำงานบนเว็บเบราว์เซอร์ ธุรกิจจะต้องแลกเปลี่ยนข้อมูลและให้บริการจากระยะไกล พวกเขาเลือกใช้เว็บแอปพลิเคชันเพื่อเชื่อมต่อกับลูกค้าได้อย่างสะดวกและปลอดภัย ฟีเจอร์ทั่วไปของเว็บไซต์ เช่น ตะกร้าสินค้า การค้นหาและการกรองสินค้า การส่งข้อความโต้ตอบแบบทันที และฟีดข่าวโซเชียลมีเดียล้วนเป็นเว็บแอปพลิเคชันที่มีการออกแบบ ช่วยให้ผู้ใช้เข้าถึงการทำงานที่ซับซ้อนได้โดยไม่ต้องติดตั้งหรือกำหนดค่าซอฟต์แวร์

² <https://www.mindphp.com/คู่มือ/73-คืออะไร/2052-client-server-คืออะไร.html>

³ <https://aws.amazon.com/th/what-is/web-application/>



รูปที่ 2.3 ลักษณะของเว็บแอปพลิเคชัน

ประโยชน์ของเว็บแอปพลิเคชัน

1. การช่วยการเข้าถึง

สามารถเข้าถึงเว็บแอปได้จากเว็บเบราว์เซอร์ทั้งหมดและผ่านอุปกรณ์ส่วนบุคคลและอุปกรณ์ทางธุรกิจต่างๆ ได้ ทีมงานต่างๆ ที่อยู่คนละสถานที่กันจะสามารถเข้าถึงเอกสาร ระบบจัดการเนื้อหา และบริการทางธุรกิจอื่นๆ ที่ใช้ร่วมกันได้ผ่านเว็บแอปพลิเคชันที่มีการสมัครใช้บริการ

2. การพัฒนาอย่างมีประสิทธิภาพ

จากรายละเอียดข้างต้น กระบวนการพัฒนาสำหรับเว็บแอปนั้นค่อนข้างง่ายและคุ้มค่าสำหรับธุรกิจ ทีมงานขนาดเล็กสามารถบรรลุวงจรพัฒนาระยะสั้นได้ ทำให้เว็บแอปพลิเคชันเป็นวิธีการสร้างโปรแกรมคอมพิวเตอร์ที่ถือว่ามีประสิทธิภาพและราคาข่อมเยา นอกจากนี้ เนื่องจากสามารถใช้งานเวอร์ชันเดียวกันกับเบราว์เซอร์และอุปกรณ์สมัยใหม่ได้ทั้งหมด คุณจึงไม่มีความจำเป็นที่จะต้องสร้างการทำซ้ำหลายๆ ครั้งสำหรับแพลตฟอร์มต่างๆ

3. ความเรียบง่ายของผู้ใช้

เนื่องจากผู้ใช้ไม่จำเป็นต้องดาวน์โหลดเว็บแอป จึงทำให้เว็บแอปสามารถเข้าถึงได้ง่าย โดยไม่จำเป็นต้องมีการบำรุงรักษาจากผู้ใช้ปลายทางรวมถึงความต้องการด้านความจุของฮาร์ดไดรฟ์ เว็บแอปพลิเคชันจะได้รับการอัปเดตซอฟต์แวร์และความปลอดภัยโดยอัตโนมัติ หมายความว่าเว็บแอปพลิเคชันเหล่านี้จะได้รับการอัปเดตอยู่เสมอและมีความเสี่ยงน้อยต่อการละเมิดความปลอดภัย

4. ความสามารถในการปรับขนาด

ธุรกิจที่ใช้เว็บแอปสามารถเพิ่มผู้ใช้ได้ตามต้องการ โดยไม่ต้องมีโครงสร้างพื้นฐานเพิ่มเติมหรือฮาร์ดแวร์ที่มีราคาแพง นอกจากนี้ ระบบจะจัดเก็บข้อมูลเว็บแอปพลิเคชันส่วนใหญ่เอาไว้ใน

ระบบคลาวด์ ซึ่งหมายความว่าธุรกิจของคุณจะไม่มี ความจำเป็นที่จะต้องลงทุนในด้านความจุของพื้นที่จัดเก็บเพิ่มเติมเพื่อเรียกใช้เว็บแอป

เว็บแอปพลิเคชันทำงานอย่างไร

เว็บแอปพลิเคชันมีสถาปัตยกรรมแบบไคลเอนต์ฝั่งเซิร์ฟเวอร์ โค้ดของแอปเหล่านั้นจะแบ่งออกเป็นสองส่วน ได้แก่ สคริปต์ฝั่งไคลเอนต์และสคริปต์ฝั่งเซิร์ฟเวอร์

1. สถาปัตยกรรมฝั่งไคลเอนต์

สคริปต์ฝั่งไคลเอนต์จะเกี่ยวข้องกับฟังก์ชันการใช้งานของอินเทอร์เน็ตเบราว์เซอร์ เช่น ปุ่มและกล่องแบบดรอปดาวน์ต่างๆ เมื่อผู้ใช้คลิกลิงก์ของเว็บแอป เว็บเบราว์เซอร์จะโหลดสคริปต์ฝั่งไคลเอนต์และแสดงองค์ประกอบกราฟิกและข้อความสำหรับอินเทอร์เน็ตเบราว์เซอร์ของผู้ใช้ ตัวอย่างเช่น ผู้ใช้สามารถอ่านเนื้อหา คู่มือไอ หรือกรอกรายละเอียดในแบบฟอร์มการติดต่อได้ การดำเนินการต่างๆ อย่างเช่น การคลิกปุ่มส่งไปที่เซิร์ฟเวอร์ตามคำขอของลูกค้า

2. สถาปัตยกรรมฝั่งเซิร์ฟเวอร์

สคริปต์ฝั่งเซิร์ฟเวอร์จะเกี่ยวข้องกับการประมวลผลข้อมูล เซิร์ฟเวอร์ของเว็บแอปพลิเคชันจะประมวลผลคำขอของไคลเอนต์และส่งกลับการตอบสนอง โดยปกติแล้ว คำขอมักจะเป็นการขอข้อมูลเพิ่มเติมหรือเป็นคำขอสำหรับการแก้ไขหรือการบันทึกข้อมูลใหม่ ตัวอย่างเช่น หากผู้ใช้คลิกที่ปุ่มอ่านเพิ่มเติม เว็บแอปพลิเคชันเซิร์ฟเวอร์จะส่งเนื้อหาหลักไปยังผู้ใช้นั้น หากผู้ใช้คลิกปุ่มส่งแอปพลิเคชันเซิร์ฟเวอร์จะบันทึกข้อมูลผู้ใช้ไว้ในฐานข้อมูล ในบางกรณี เซิร์ฟเวอร์จะดำเนินการตามคำขอข้อมูลและส่งหน้า HTML ที่สมบูรณ์กลับไปยังไคลเอนต์ วิธีนี้เรียกว่าการแสดงผลฝั่งเซิร์ฟเวอร์

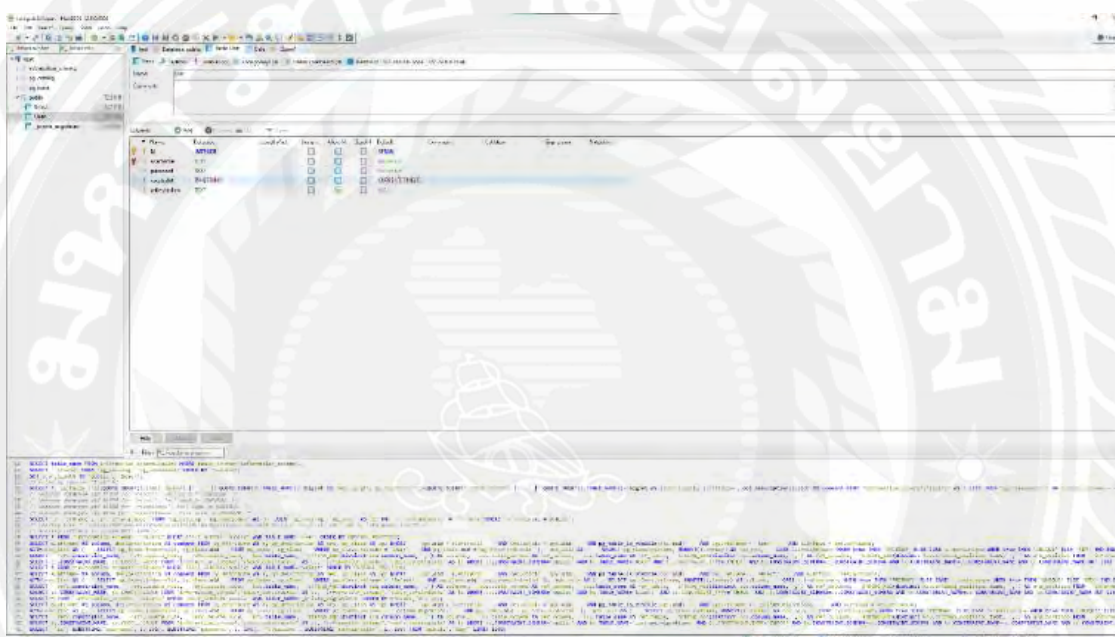
2.4 โปรแกรม HeidiSQL⁴

HeidiSQL เป็นโปรแกรมจัดการฐานข้อมูลที่ใช้งานได้ง่ายและฟรี เป็นเครื่องมือที่ช่วยในการเชื่อมต่อและจัดการกับฐานข้อมูล MySQL, MariaDB, PostgreSQL, SQLite, Microsoft SQL Server และ Oracle ได้อย่างมีประสิทธิภาพ มีคุณสมบัติในการดึงข้อมูล แก้ไขข้อมูล สร้างคำสั่ง SQL และการจัดการโครงสร้างฐานข้อมูลต่างๆ HeidiSQL เป็นเครื่องมือที่นิยมใช้ในการพัฒนาแอปพลิเคชันเว็บ และโปรแกรมคอมพิวเตอร์ที่ต้องการเชื่อมต่อกับฐานข้อมูลเพื่อจัดการข้อมูลและการประมวลผลข้อมูลในระบบต่างๆ โดยคณะผู้จัดทำได้นำโปรแกรม HeidiSQL มาใช้ในการเชื่อมต่อกับ PostgreSQL

⁴ <https://www.glurgeek.com/education/การจัดการฐานข้อมูลโดย/>



รูปที่ 2.4 สัญลักษณ์ HeidiSQL



รูปที่ 2.5 ตัวอย่างหน้าจอ HeidiSQL ที่ใช้ในการพัฒนาแอปพลิเคชัน

2.5 ระบบฐานข้อมูล PostgreSQL⁵

PostgreSQL เป็นระบบฐานข้อมูลที่เป็นที่นิยมอย่างมากในโลก Open Source และฐานข้อมูลที่มีความสามารถและประสิทธิภาพสูง เป็นฐานข้อมูลแบบ relational database management system (RDBMS) ที่สามารถใช้งานได้ทั้งบนแพลตฟอร์ม Windows, macOS, Linux, และ Unix-like systems ซึ่งได้รับการพัฒนาและรองรับอย่างต่อเนื่องจากชุมชน Open Source และบริษัทใหญ่ๆ ที่เล็งเห็นความสำคัญของการพัฒนาซอฟต์แวร์โอเพนซอร์สและฐานข้อมูลที่เสถียรและมีประสิทธิภาพสูง คุณสมบัติของ PostgreSQL รวมถึงความสามารถในการจัดการข้อมูลใน

⁵ <https://appmaster.io/th/blog/postgresql-khuue-air>

รูปแบบที่มีความยืดหยุ่นสูง โดยคณะผู้จัดทำได้นำ PostgreSQL มาใช้ในการจัดเก็บข้อมูลของระบบจัดการทรัพย์สินภายในภาควิชาวิทยาการคอมพิวเตอร์

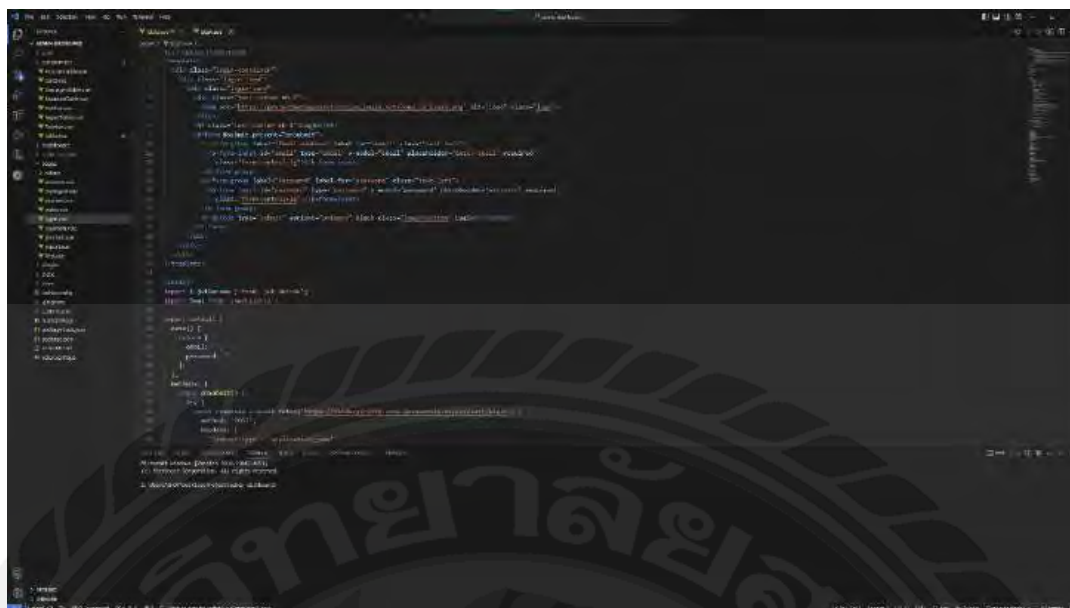
2.6 Visual Studio Code⁶

Visual Studio Code หรือ Vs Code เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ด จากค่ายไมโครซอฟท์ สามารถนำมาใช้งานได้แบบฟรี ซึ่ง Visual Studio Code เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์ม รองรับการใช้งานทั้งบน Windows, macOS และ Linux สนับสนุนทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่างๆ ให้เลือกใช้อย่างมากมายไม่ว่าจะเป็น 1.การเปิดใช้งานภาษาอื่น ๆ ทั้ง ภาษา C++, C#, Java, Python, PHP หรือ Go 2.Themes 3.Debugger 4.Commands เป็นต้น โดยคณะผู้จัดทำได้นำ Visual Studio Code มาใช้ในการเขียนชุดคำสั่ง (Code) และทดสอบระบบ



รูปที่ 2.6 สัญลักษณ์โปรแกรม Visual Studio Code

⁶ <https://www.mindphp.com/บทความ/microsoft/4829-visual-studio-code.html>



รูปที่ 2.7 ตัวอย่างหน้าจอ โปรแกรม Visual Studio Code ที่ใช้ในการพัฒนา

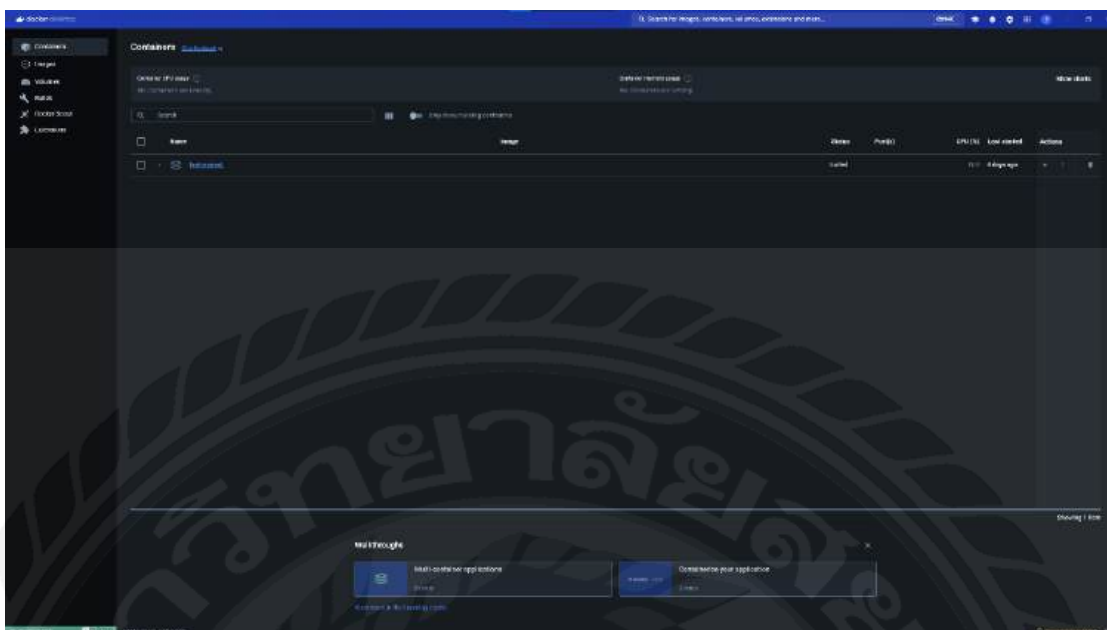
2.7 Docker Desktop⁷

Docker คือ engine ตัวหนึ่งที่มีการทำงานในลักษณะจำลองสภาพแวดล้อมขึ้นมาบนเครื่อง server เพื่อใช้ในการ run service ที่ต้องการ มีการทำงานคล้ายคลึงกับ Virtual Machine เช่น VMWare, VirtualBox, XEN, KVM แต่ข้อแตกต่างที่ชัดเจนคือ Virtual Machine ที่รู้จักกันก่อนหน้านี้ นั้นเป็นการจำลองทั้ง OS เพื่อใช้งานและหากต้องการใช้งาน service ใดๆ จึงทำการติดตั้งเพิ่มเติมบน OS นั้นๆ แต่สำหรับ docker แล้วจะใช้ container ในการจำลองสภาพแวดล้อมขึ้นมา เพื่อใช้งานสำหรับ 1 service ที่ต้องการใช้งานเท่านั้น โดยไม่ต้องมีส่วนของ OS เข้าไปเกี่ยวข้องเหมือน Virtual Machines อื่นๆ โดยคณะผู้จัดทำได้นำ Docker Desktop มาใช้ในการช่วยให้ย้ายระบบจากเครื่องพัฒนาสู่เครื่องเซิร์ฟเวอร์ หรือจากคลาวด์หนึ่งไปยังอีกคลาวด์หนึ่งได้



รูปที่ 2.8 สัญลักษณ์ Docker Desktop

⁷ <https://medium.com/@rachatatongpagdee/docker-คืออะไร-ใช้งานอย่างไร-7e77145967b6>



รูปที่ 2.9 ตัวอย่างโปรแกรม Docker Desktop

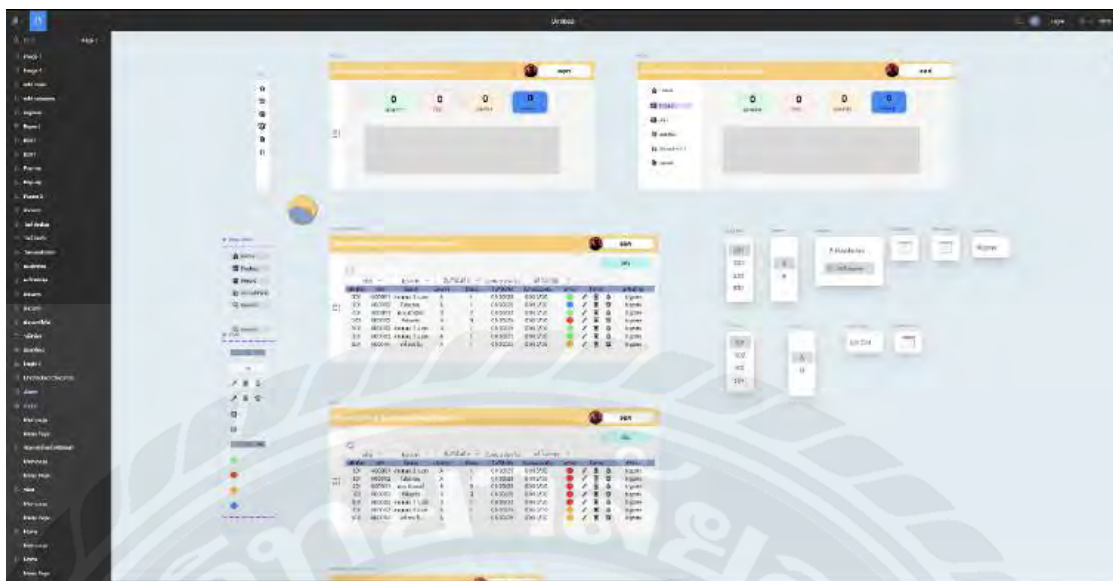
2.8 Figma⁸

Figma คือ เครื่องมือออกแบบเว็บไซต์ หรือแอปพลิเคชันต่างๆ ที่เกิดมาเพื่อช่วยนักออกแบบ UX/ UI โดยสามารถใช้งานได้ผ่านทาง web browser ไม่จำเป็นต้อง install (แต่ก็มีแอปฯ ให้สามารถ install ลงเครื่อง) ทำให้สะดวกในการใช้งานมาก โดยตัวเครื่องมือออกแบบมาให้เหมาะกับคนที่จำเป็นต้องทำโปรเจกต์ร่วมกันกับทีม หรือต้องการหาคำปรึกษาเวลาคุยโปรเจกต์กับลูกค้า และลูกค้าเปิดไฟล์งานไม่ได้ อีกทั้งตัวเครื่องมือยังมีฟีเจอร์ที่น่าสนใจ ช่วยให้สามารถทำงานได้ง่ายขึ้นอีกด้วย โดยคณะผู้จัดทำได้นำฟิกมาร์มาออกแบบ UX/ UI หน้าตาของแอปพลิเคชัน



รูปที่ 2.10 สัญลักษณ์ Figma

⁸ <https://dev.classmethod.jp/articles/whats-about-figma/>



รูปที่ 2.11 ตัวอย่างหน้าจอ UX/ UI ที่ออกแบบด้วย Figma

2.9 Bootstrap Vue⁹

Bootstrap Vue คือ เฟรมเวิร์กสำหรับการพัฒนาเว็บที่ผสมการใช้งาน Bootstrap และ Vue.js เข้าด้วยกัน ทำให้สามารถสร้างอินเทอร์เฟซผู้ใช้ (UI) ที่ตอบสนองและทันสมัยได้ง่ายขึ้น โดยใช้ประโยชน์จากคอมโพเนนต์ และเครื่องมือต่างๆ ของ Bootstrap ผ่านการเขียนโค้ดในรูปแบบของ Vue.js ซึ่งช่วยให้การพัฒนาเว็บแอปพลิเคชันมีความรวดเร็วและมีประสิทธิภาพมากขึ้นโดยคณะผู้จัดทำได้นำมาใช้เพื่อช่วยให้การเขียนชุดคำสั่ง ที่เกี่ยวกับ UX/ UI สะดวกและง่ายขึ้น



รูปที่ 2.12 สัญลักษณ์ Bootstrap Vue

⁹ <https://bootstrap-vue.org/docs>

2.10 Nuxt.js¹⁰



รูปที่ 2.13 สัญลักษณ์ Nuxt.js

Nuxt.js คือ เฟรมเวิร์กสำหรับการพัฒนาแอปพลิเคชันเว็บที่สร้างขึ้นบน Vue.js โดยเน้นการพัฒนาแอปพลิเคชันที่มีการเรนเดอร์ฝั่งเซิร์ฟเวอร์ (Server-Side Rendering - SSR) และการสร้างเว็บไซต์แบบสถิต (Static Site Generation - SSG) ช่วยให้การพัฒนาเว็บแอปพลิเคชันมีประสิทธิภาพสูงขึ้น ด้วยคุณสมบัติ เช่น การจัดการเส้นทาง (Routing) การแยกโค้ด (Code Splitting) และการปรับแต่ง SEO โดยคณะผู้จัดทำได้นำมาใช้เพื่อช่วยให้เขียนโค้ดที่เกี่ยวข้องกับเว็บแอปพลิเคชันได้สะดวกขึ้น

¹⁰ <https://nuxt.com/docs/getting-started/introduction>

บทที่ 3 การวิเคราะห์ระบบ

3.1 วิเคราะห์ระบบงานปัจจุบัน (As-Is System Analysis)

3.1.1 รายละเอียดของระบบงานปัจจุบัน

ในการทำงานปัจจุบันของภาควิชาวิทยาการคอมพิวเตอร์ยังไม่มีระบบสำหรับเก็บข้อมูลของอุปกรณ์ชนิดต่างๆ ภายในภาควิชา ทางคณะผู้จัดทำจึงได้ทำการพัฒนาระบบเข้ามาช่วยแทนการจดบันทึกข้อมูลของอุปกรณ์ชนิดต่างๆ

3.1.2 ปัญหาของระบบงานปัจจุบัน

ปัญหาของระบบงานปัจจุบัน โดยการเก็บข้อมูลอุปกรณ์ชนิดๆ ต่างของภาควิชาวิทยาการคอมพิวเตอร์พบการทำงานที่เกิดข้อผิดพลาดบ่อย เช่น การสูญหายของใบเสร็จหรือ มีการใช้เวลานานในการหาข้อมูลใบเสร็จของอุปกรณ์ต่างๆ เนื่องจากระบบการทำงานปัจจุบันไม่มีการจดบันทึกข้อมูลของอุปกรณ์ต่างๆ นอกเหนือจากการเก็บใบเสร็จของอุปกรณ์ต่างๆ เป็นต้น

3.1.3 แนวทางการแก้ไขปัญหา

แนวทางการแก้ปัญหาทางคณะผู้จัดทำจึงได้ทำการพัฒนาโปรแกรมขึ้นมาเพื่อช่วยในการทำงานให้มีประสิทธิภาพยิ่งขึ้น โดยระบบสามารถช่วยในด้านของการบันทึกข้อมูลลงในระบบแทนการเก็บใบเสร็จ โดยพัฒนาระบบให้สามารถบันทึก ข้อมูลชื่ออุปกรณ์ ประเภทของอุปกรณ์ วันที่ชื่ออุปกรณ์ ชื่อของคนที่เพิ่ม/ แก้ไขอุปกรณ์ รวมถึงการเก็บข้อมูลอื่นๆ ซึ่งระบบช่วยเพิ่มความสะดวกรวดเร็วเพิ่มมากขึ้น อุปกรณ์ ชื่อของคนที่เพิ่ม/ แก้ไขอุปกรณ์ รวมถึงการเก็บข้อมูลอื่นๆ ซึ่งระบบช่วยเพิ่มความสะดวกรวดเร็วเพิ่มมากขึ้น

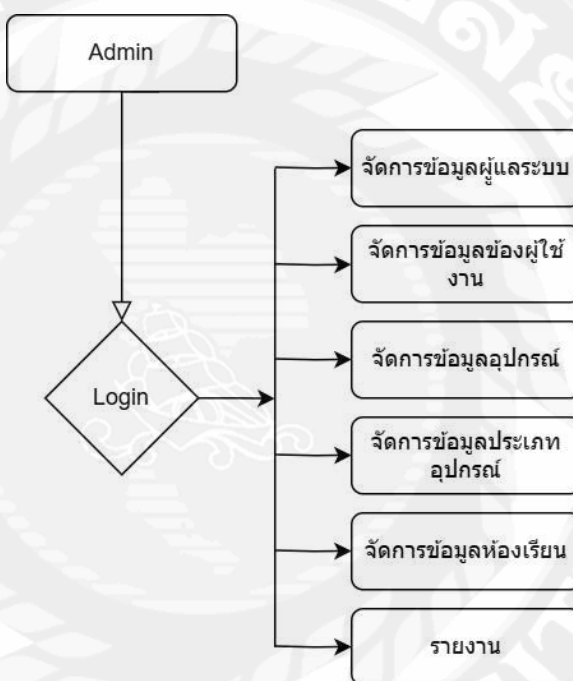
3.2 วิเคราะห์ระบบงานใหม่(New System Analysis)

3.2.1 รายละเอียดของระบบงานใหม่ (New System Analysis)

ระบบจัดเก็บข้อมูลทรัพย์สิน กรณีศึกษา ภาควิชาวิทยาการคอมพิวเตอร์มหาวิทยาลัยสยาม เป็นเว็บไซต์ที่ถูกพัฒนามาเพื่อเก็บข้อมูลอุปกรณ์ต่างๆ ภายในภาควิชาวิทยาการคอมพิวเตอร์ระบบจะทำงานโดยเว็บแอปพลิเคชันจะแสดงข้อมูลของอุปกรณ์ภายในภาควิชาวิทยาการคอมพิวเตอร์ ข้อมูลผู้ใช้ ข้อมูลรายงาน เป็นต้น ในการพัฒนาระบบประกอบด้วยผู้ใช้งาน และผู้ดูแลระบบ โดยส่วนของเว็บแอปพลิเคชันผู้ดูแลระบบจะต้องลงทะเบียนให้กับผู้ใช้งาน (Add user) ผ่านระบบเพื่อเข้าใช้งานระบบ เมื่อผู้ดูแลระบบเข้าใช้งานระบบจะแสดงข้อมูลของอุปกรณ์ภายในภาควิชาวิทยาการคอมพิวเตอร์ ข้อมูลผู้ใช้งาน ข้อมูลรายงาน ผู้ดูแลระบบสามารถแก้ไขหรือเพิ่มข้อมูลในส่วนต่างๆ ได้ตามความต้องการ ในส่วนของผู้ใช้งานระบบจะแสดงข้อมูลของอุปกรณ์ภายใน

ภาควิชาวิทยาการคอมพิวเตอร์ ข้อมูลผู้ใช้งาน ข้อมูลรายงาน และผู้ใช้งานสามารถที่จะค้นหาอุปกรณ์จากการใส่ชื่อของอุปกรณ์เพื่อทำการค้นหาอุปกรณ์ที่ผู้ใช้งานต้องการ หลังจากที่ได้ข้อมูลของอุปกรณ์แล้ว ผู้ใช้งานสามารถเพิ่มอุปกรณ์ได้โดยการใช้ข้อมูลต่างๆของอุปกรณ์ เพื่อบันทึกลงฐานข้อมูลเพื่อแสดงข้อมูลให้ผู้ใช้งานสามารถดูรายละเอียดได้ เครื่องมือที่ใช้ในการพัฒนาประกอบด้วย Visual Studio Code, Docker Desktop , Postmanเขียนชุดคำสั่งด้วยภาษา JavaScript และใช้ Nuxt.js, Bootstrap Vue ในการสร้าง UI ข้อมูลของผู้ใช้งาน และข้อมูลต่างๆ ระบบจัดการฐานข้อมูลรูปแบบ SQL ด้วย PostgreSQL

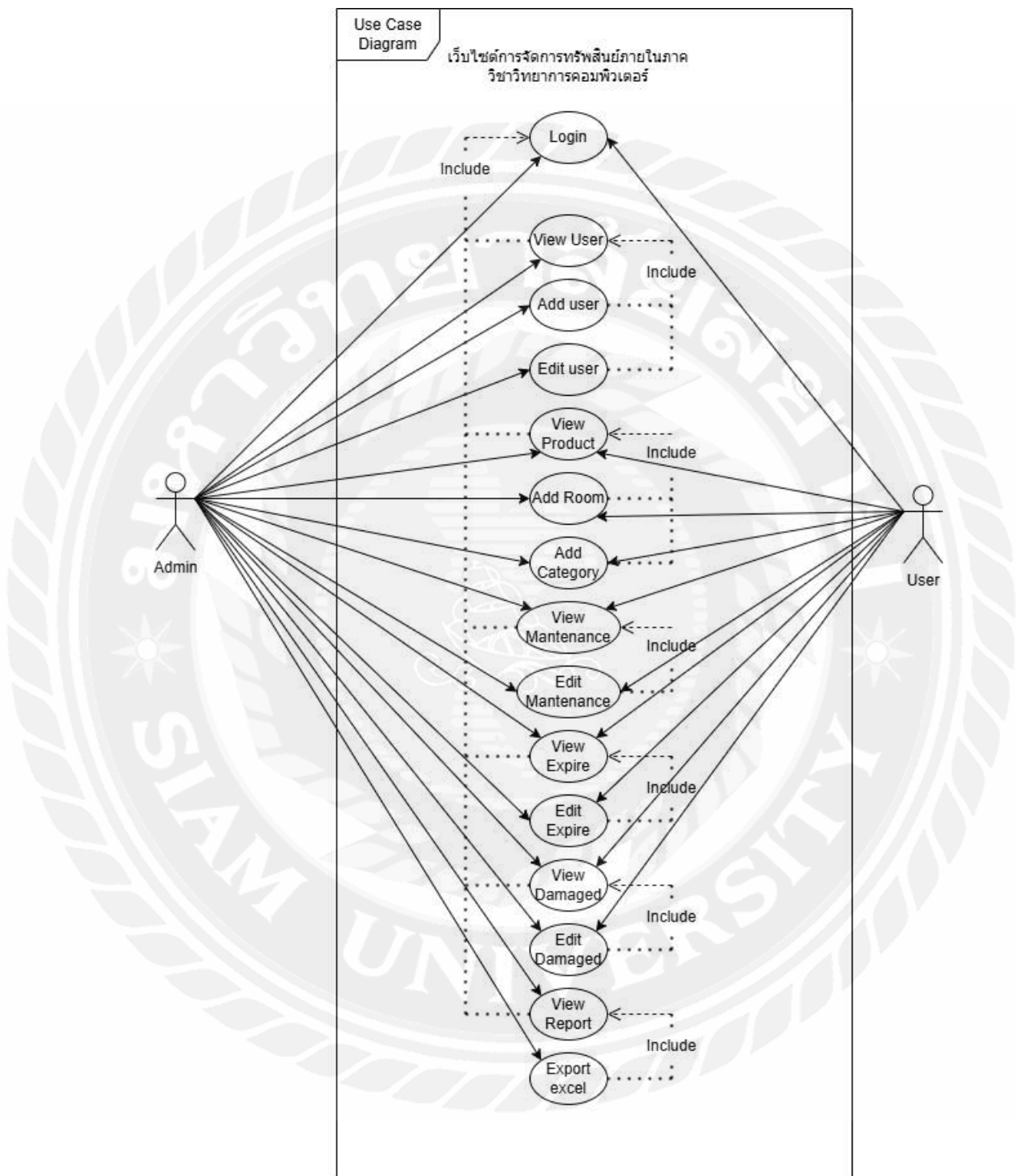
3.2.2 ขั้นตอนการทำงานของระบบงานใหม่ (New System Work Flow Diagram)



รูปที่ 3.1 Work Flow Diagram ของระบบจัดเก็บข้อมูลทรัพย์สิน กรณีศึกษา ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสยาม

3.3 แสดงภาพรวมฟังก์ชันการทำงานของระบบด้วย Use Case Diagram

3.3.1 Use Case Diagram



รูปที่ 3.2 Use Case Diagram ของระบบจัดเก็บข้อมูลทรัพย์สิน กรณีศึกษา ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสยาม

3.3.2 แสดงรายละเอียดของยูสเคส (Use Case Description)

ตารางที่ 3.1 แสดงรายละเอียดของ Use case: Login

Use Case Name	Login
Use Case ID	UC1
Brief Description	สำหรับการยืนยันตัวตนของผู้ใช้โดยใช้อีเมล และรหัสผ่าน
Primary Actors	Administrator, User
Secondary Actors	-
Preconditions	ผู้ใช้งานต้องเป็นสมาชิกของระบบก่อนถึงจะสามารถใช้งานได้
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้งานเข้าสู่เว็บไซต์ 2. ระบบจะแสดงช่องสำหรับการป้อน Email และ Password 3. ผู้ใช้คลิกปุ่ม Login 4. ระบบจะไปที่หน้า Dashboard
Post Condition	สามารถเข้าสู่ระบบและใช้งานระบบได้
Alternative Flows	-

ตารางที่ 3.2 แสดงรายละเอียดของ Use case: View user

Use Case Name	View user
Use Case ID	UC2
Brief Description	หน้าแสดงข้อมูลของผู้ใช้งาน
Primary Actors	Administrator
Secondary Actors	-
Preconditions	ผู้ดูแลระบบจะต้องเข้าสู่ระบบก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ดูแลระบบเข้าสู่เว็บไซต์และทำการเข้าสู่ระบบ 2. ให้เลือกไปที่หัวข้อ Manage Accounts 3. แสดงข้อมูลบุคลากร
Post Condition	ระบบแสดงข้อมูลบุคลากรที่มีอยู่ภายในระบบ
Alternative Flows	-

ตารางที่ 3.3 แสดงรายละเอียดของ Use case: Add User

Use Case Name	Add User
Use Case ID	UC3
Brief Description	สำหรับเพิ่มข้อมูลของผู้ใช้งาน
Primary Actors	Administrator
Secondary Actors	-
Preconditions	ผู้ดูแลระบบจะต้องเข้าสู่ระบบก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ดูแลระบบเข้าสู่เว็บไซต์และทำการเข้าสู่ระบบ 2. ให้เลือกไปที่หัวข้อ Manage Accounts 3. ให้คลิกปุ่ม + Account 4. กรอกข้อมูลที่ต้องการลงในField 5. คลิกปุ่ม save
Post Condition	ระบบทำการบันทึกข้อมูลผู้ใช้งาน
Alternative Flows	-

ตารางที่ 3.4 แสดงรายละเอียดของ Use case: Edit user

Use Case Name	Edit user
Use Case ID	UC4
Brief Description	สำหรับแก้ไขข้อมูลของผู้ใช้งาน
Primary Actors	Administrator
Secondary Actors	-
Preconditions	ผู้ดูแลระบบจะต้องเข้าสู่ระบบก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ดูแลระบบเข้าสู่เว็บไซต์และทำการเข้าสู่ระบบ 2. ให้เลือกไปที่หัวข้อ Manage Accounts 3. แสดงข้อมูลของผู้ใช้งานที่มีการบันทึกในระบบ 4. ให้คลิกที่ปุ่ม Edit ของผู้ใช้งานที่ต้องการแก้ไข 5. กรอกข้อมูลที่ต้องการแก้ไขลงในField 6. คลิกปุ่ม Save
Post Condition	ระบบทำการแสดงข้อมูลของผู้ใช้งานที่ถูกแก้ไขเรียบร้อยแล้ว
Alternative Flows	-

ตารางที่ 3.5 แสดงรายละเอียดของ Use case: View Product

Use Case Name	View Product
Use Case ID	UC5
Brief Description	หน้าแสดงข้อมูลของอุปกรณ์
Primary Actors	Administrator, User
Secondary Actors	-
Preconditions	ผู้ใช้งานจะต้องเข้าสู่ระบบก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ดูแลระบบเข้าสู่เว็บไซต์และทำการเข้าสู่ระบบ 2. ให้เลือกไปที่หัวข้อ Products 3. แสดงข้อมูลอุปกรณ์
Post Condition	ระบบจะแสดงข้อมูลอุปกรณ์ที่มีอยู่ในระบบ
Alternative Flows	-

ตารางที่ 3.6 แสดงรายละเอียดของ Use case: Add Room

Use Case Name	Add Room
Use Case ID	UC6
Brief Description	สำหรับเพิ่มข้อมูลห้องเรียน
Primary Actors	Administrator, User
Secondary Actors	-
Preconditions	ผู้ใช้งานระบบจะต้องเข้าสู่ระบบก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ดูแลระบบเข้าสู่เว็บไซต์และทำการเข้าสู่ระบบ 2. ให้เลือกไปที่หัวข้อ Product 3. ให้คลิกปุ่ม + Room 4. กรอกข้อมูลที่ต้องการลงใน Field 5. คลิกปุ่ม Save
Post Condition	ระบบทำการบันทึกข้อมูลห้องเรียน
Alternative Flows	-

ตารางที่ 3.7 แสดงรายละเอียดของ Use case: Add Category

Use Case Name	Add Category
Use Case ID	UC7
Brief Description	สำหรับเพิ่มข้อมูลสถานะ
Primary Actors	Administrator, User
Secondary Actors	-
Preconditions	ผู้ใช้งานระบบจะต้องเข้าสู่ระบบก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ดูแลระบบเข้าสู่เว็บไซต์และทำการเข้าสู่ระบบ 2. ให้เลือกไปที่หัวข้อ Product 3. ให้คลิกปุ่ม + Category 4. กรอกข้อมูลที่ต้องการลงในField 5. คลิกปุ่ม Save
Post Condition	ระบบทำการบันทึกข้อมูลสถานะ
Alternative Flows	-

ตารางที่ 3.8 แสดงรายละเอียดของ Use case: View Maintenance

Use Case Name	View Maintenance
Use Case ID	UC8
Brief Description	หน้าแสดงข้อมูลของอุปกรณ์ที่ซ่อมบำรุง
Primary Actors	Administrator, User
Secondary Actors	-
Preconditions	ผู้ใช้งานจะต้องเข้าสู่ระบบก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ดูแลระบบเข้าสู่เว็บไซต์และทำการเข้าสู่ระบบ 2. ให้เลือกไปที่หัวข้อ Maintenance 3. แสดงข้อมูลของอุปกรณ์ที่อยู่ในสถานะซ่อมบำรุง
Post Condition	ระบบจะแสดงอุปกรณ์ที่อยู่ในสถานะซ่อมบำรุง
Alternative Flows	-

ตารางที่ 3.9 แสดงรายละเอียดของ Use case: Edit Maintenance

Use Case Name	Edit Maintenance
Use Case ID	UC9
Brief Description	หน้าแก้ไขข้อมูลของอุปกรณ์ที่ซ่อมบำรุง
Primary Actors	Administrator, User
Secondary Actors	-
Preconditions	ผู้ใช้งานจะต้องเข้าสู่ระบบก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ดูแลระบบเข้าสู่เว็บไซต์และทำการเข้าสู่ระบบ 2. ให้เลือกไปที่หัวข้อ Maintenance 3. แสดงข้อมูลของอุปกรณ์ที่อยู่ในสถานะซ่อมบำรุง 4. คลิกปุ่ม Edit ของอุปกรณ์ที่ต้องการแก้ไข 5. กรอกข้อมูลที่ต้องการแก้ไขลงใน Field 6. คลิกปุ่ม Save
Post Condition	ระบบจะบันทึกข้อมูลที่แก้ไข
Alternative Flows	-

ตารางที่ 3.10 แสดงรายละเอียดของ Use case: View Expire

Use Case Name	View Expire
Use Case ID	UC10
Brief Description	หน้าแสดงข้อมูลของอุปกรณ์ที่หมดอายุ/ หมดประกัน
Primary Actors	Administrator, User
Secondary Actors	-
Preconditions	ผู้ใช้งานจะต้องเข้าสู่ระบบก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ดูแลระบบเข้าสู่เว็บไซต์และทำการเข้าสู่ระบบ 2. ให้เลือกไปที่หัวข้อ Expire 3. แสดงข้อมูลของอุปกรณ์ที่อยู่ในสถานะหมดอายุ/ หมดประกัน
Post Condition	ระบบจะแสดงอุปกรณ์ที่อยู่ในสถานะหมดอายุ/ หมดประกัน
Alternative Flows	-

ตารางที่ 3.11 แสดงรายละเอียดของ Use case: Edit Expire

Use Case Name	Edit Expire
Use Case ID	UC11
Brief Description	หน้าแก้ไขข้อมูลของอุปกรณ์ที่หมดอายุ/ หมดประกัน
Primary Actors	Administrator, User
Secondary Actors	-
Preconditions	ผู้ใช้งานจะต้องเข้าสู่ระบบก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ดูแลระบบเข้าสู่เว็บไซต์และทำการเข้าสู่ระบบ 2. ให้เลือกไปที่หัวข้อ Expire 3. แสดงข้อมูลของอุปกรณ์ที่อยู่ในสถานะหมดอายุ/หมดประกัน 4. คลิกปุ่ม Edit ของอุปกรณ์ที่ต้องการแก้ไข 5. กรอกข้อมูลที่ต้องการแก้ไขลงใน Field 6. คลิกปุ่ม Save
Post Condition	ระบบจะบันทึกข้อมูลที่แก้ไข
Alternative Flows	-

ตารางที่ 3.12 แสดงรายละเอียดของ Use case: View Damaged

Use Case Name	View Damaged
Use Case ID	UC12
Brief Description	หน้าแสดงข้อมูลของอุปกรณ์ที่เสียหาย
Primary Actors	Administrator, User
Secondary Actors	-
Preconditions	ผู้ใช้งานจะต้องเข้าสู่ระบบก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ดูแลระบบเข้าสู่เว็บไซต์และทำการเข้าสู่ระบบ 2. ให้เลือกไปที่หัวข้อ Damaged 3. แสดงข้อมูลของอุปกรณ์ที่อยู่ในสถานะเสียหาย
Post Condition	ระบบจะแสดงอุปกรณ์ที่อยู่ในสถานะเสียหาย
Alternative Flows	-

ตารางที่ 3.13 แสดงรายละเอียดของ Use case: Edit Damaged

Use Case Name	Edit Damaged
Use Case ID	UC13
Brief Description	หน้าแก้ไขข้อมูลของอุปกรณ์ที่เสียหาย
Primary Actors	Administrator, User
Secondary Actors	-
Preconditions	ผู้ใช้งานจะต้องเข้าสู่ระบบก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ดูแลระบบเข้าสู่เว็บไซต์และทำการเข้าสู่ระบบ 2. ให้เลือกไปที่หัวข้อ Damaged 3. แสดงข้อมูลของอุปกรณ์ที่อยู่ในสถานะเสียหาย 4. คลิกปุ่ม Edit ของอุปกรณ์ที่ต้องการแก้ไข 5. กรอกข้อมูลที่ต้องการแก้ไขลงใน Field 6. คลิกปุ่ม Save
Post Condition	ระบบจะบันทึกข้อมูลที่แก้ไข
Alternative Flows	-

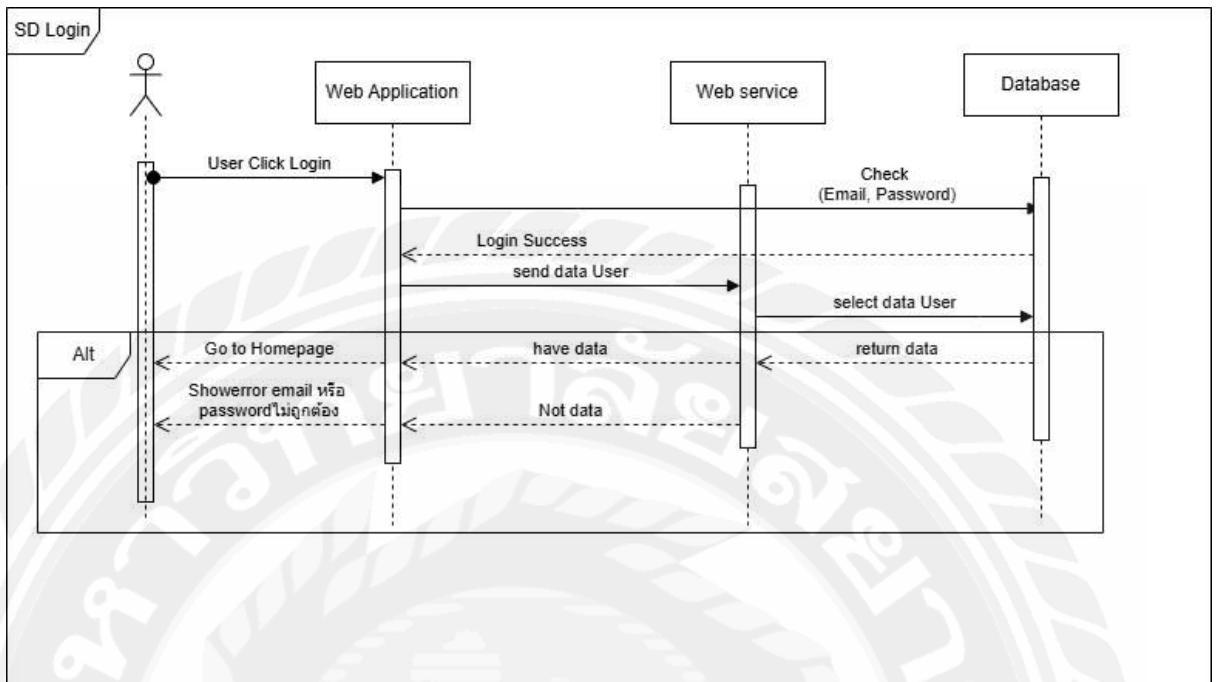
ตารางที่ 3.14 แสดงรายละเอียดของ Use case: View Report

Use Case Name	View Report
Use Case ID	UC14
Brief Description	หน้าแสดงข้อมูลอุปกรณ์ที่ต้องการออกรายงาน
Primary Actors	Administrator
Secondary Actors	-
Preconditions	ผู้ดูแลระบบจะต้องเข้าสู่ระบบก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ดูแลระบบเข้าสู่เว็บไซต์และทำการเข้าสู่ระบบ 2. ให้เลือกไปที่หัวข้อ Report 3. แสดงข้อมูลของอุปกรณ์
Post Condition	ระบบจะแสดงอุปกรณ์สำหรับออกรายงาน
Alternative Flows	-

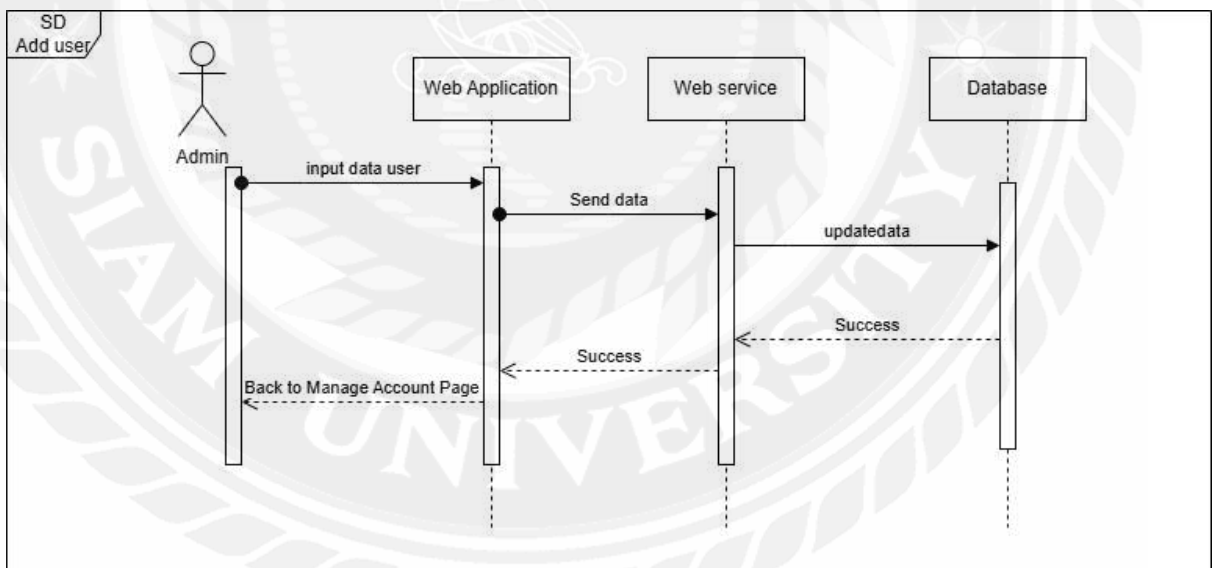
ตารางที่ 3.15 แสดงรายละเอียดของ Use case: Export Excel

Use Case Name	Export Excel
Use Case ID	UC15
Brief Description	หน้าแสดงข้อมูลอุปกรณ์ที่ต้องการออกรายงาน
Primary Actors	Administrator
Secondary Actors	-
Preconditions	ผู้ดูแลระบบจะต้องเข้าสู่ระบบก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ดูแลระบบเข้าสู่เว็บไซต์และทำการเข้าสู่ระบบ 2. ให้เลือกไปที่หัวข้อ Report 3. ให้คลิกที่ปุ่ม Export excel 4. ผู้ใช้งานสามารถเลือกข้อมูลวันที่บันทึก วันที่ซื้อ วันที่หมดอายุ/หมดประกัน หรือสถานะเพื่อกรองข้อมูลได้ 5. จะแสดงข้อมูลที่กรองแล้ว 6. คลิกปุ่ม Export
Post Condition	ระบบจะดาวน์โหลดไฟล์รายงานเป็นไฟล์ Excel
Alternative Flows	-

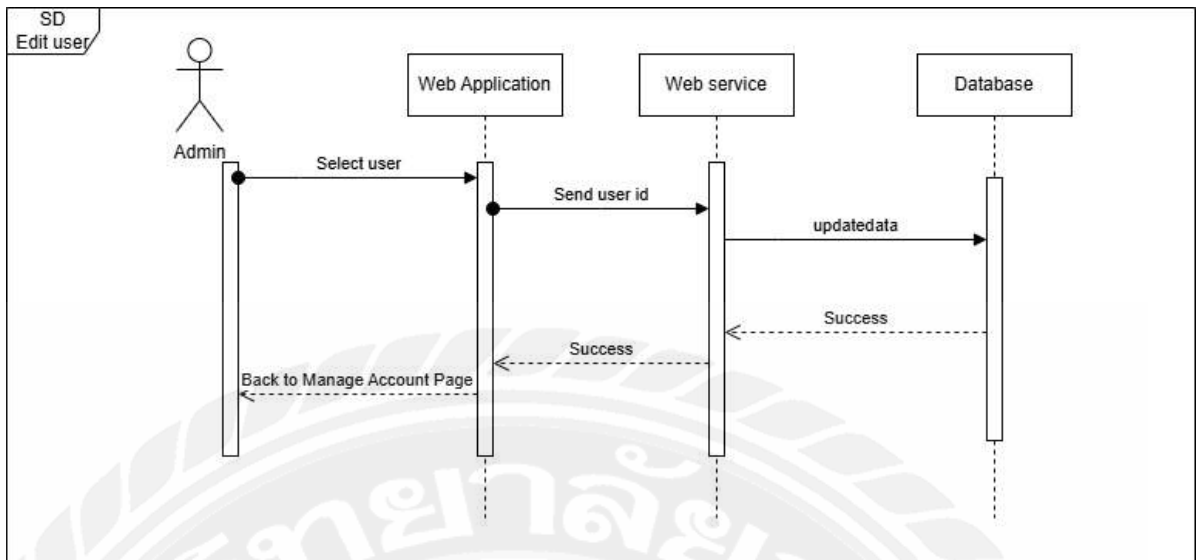
3.4 แสดงขั้นตอนการทำงานของฟังก์ชัน (Sequence Diagram)



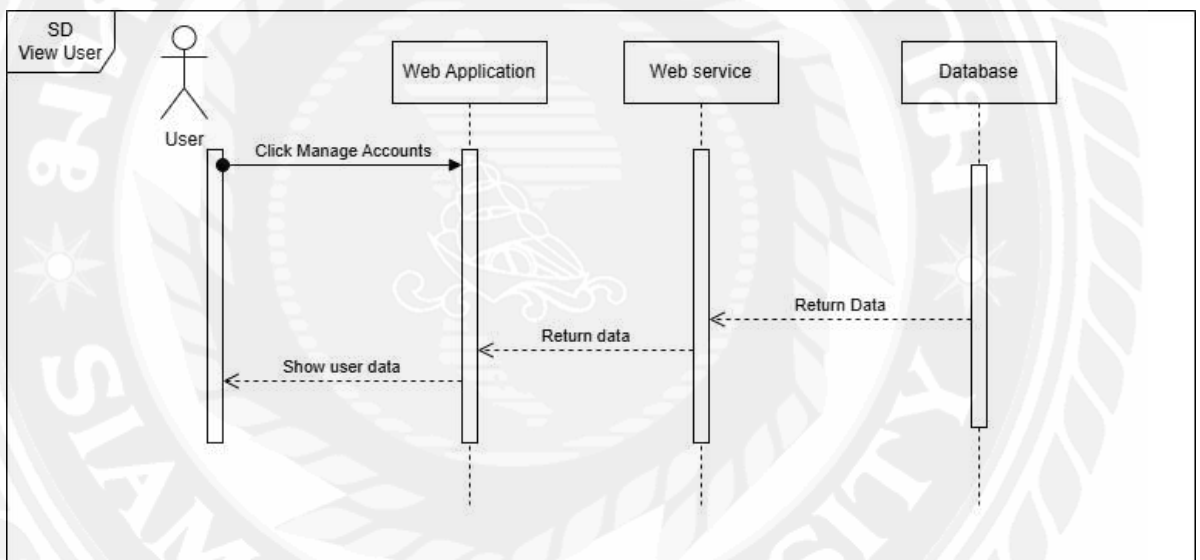
รูปที่ 3.3 Sequence Diagram: Login



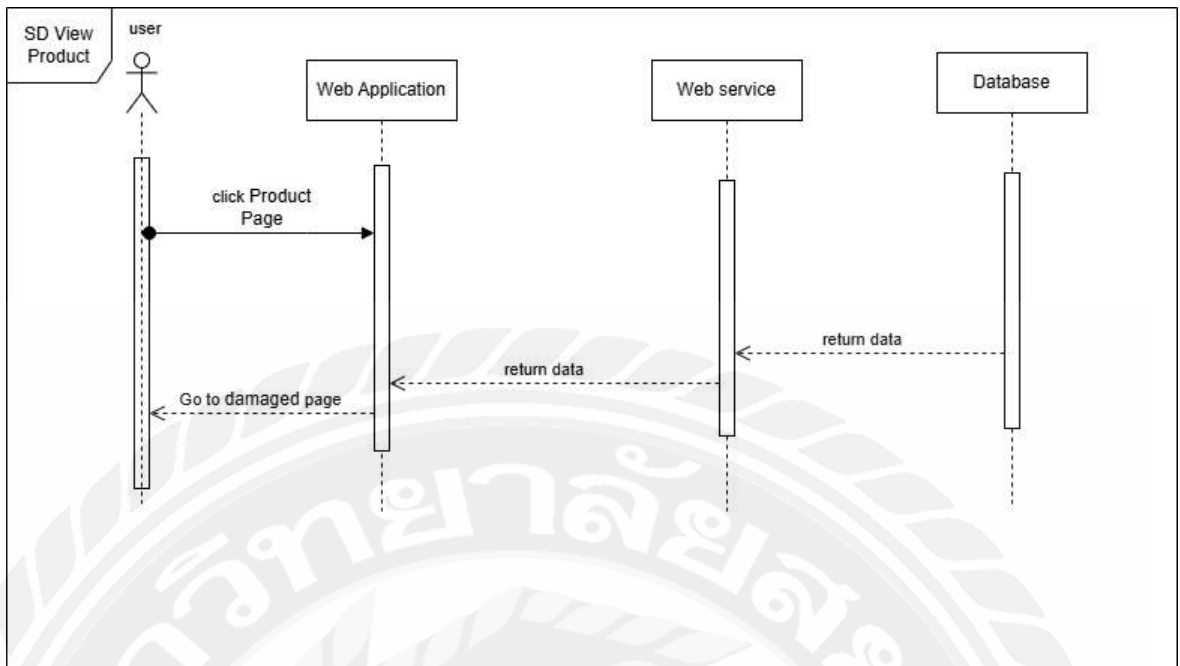
รูปที่ 3.4 Sequence Diagram: Add user



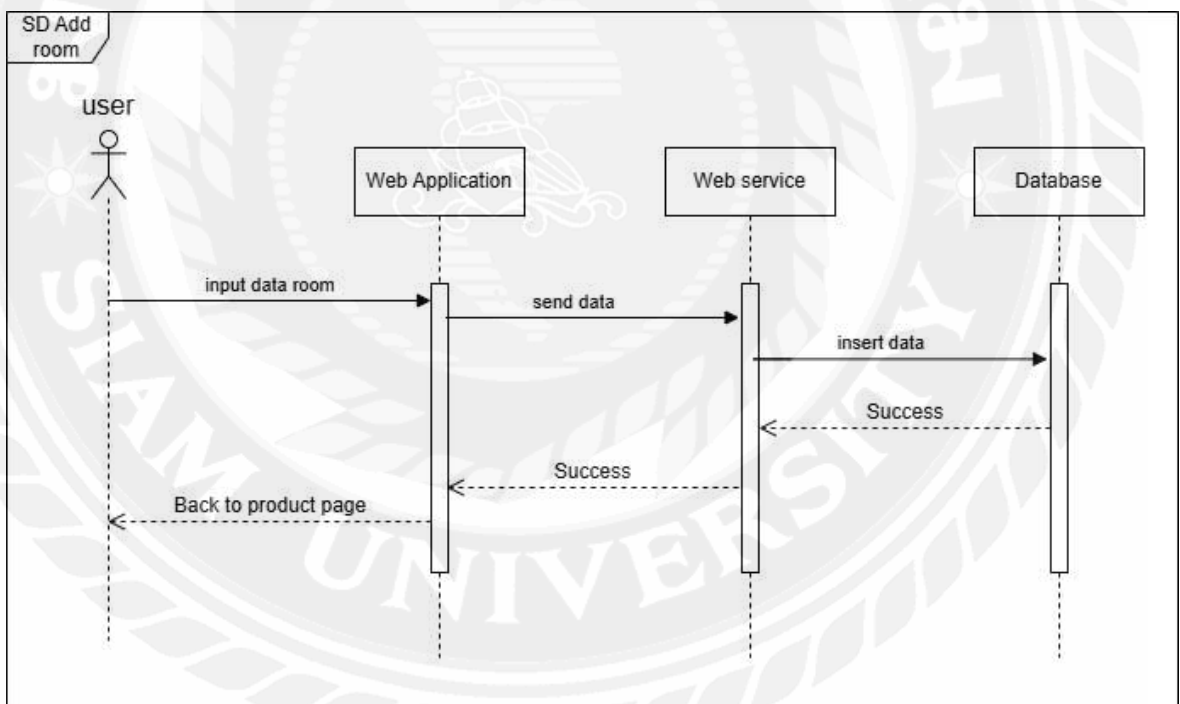
รูปที่ 3.5 Sequence Diagram: Edit user



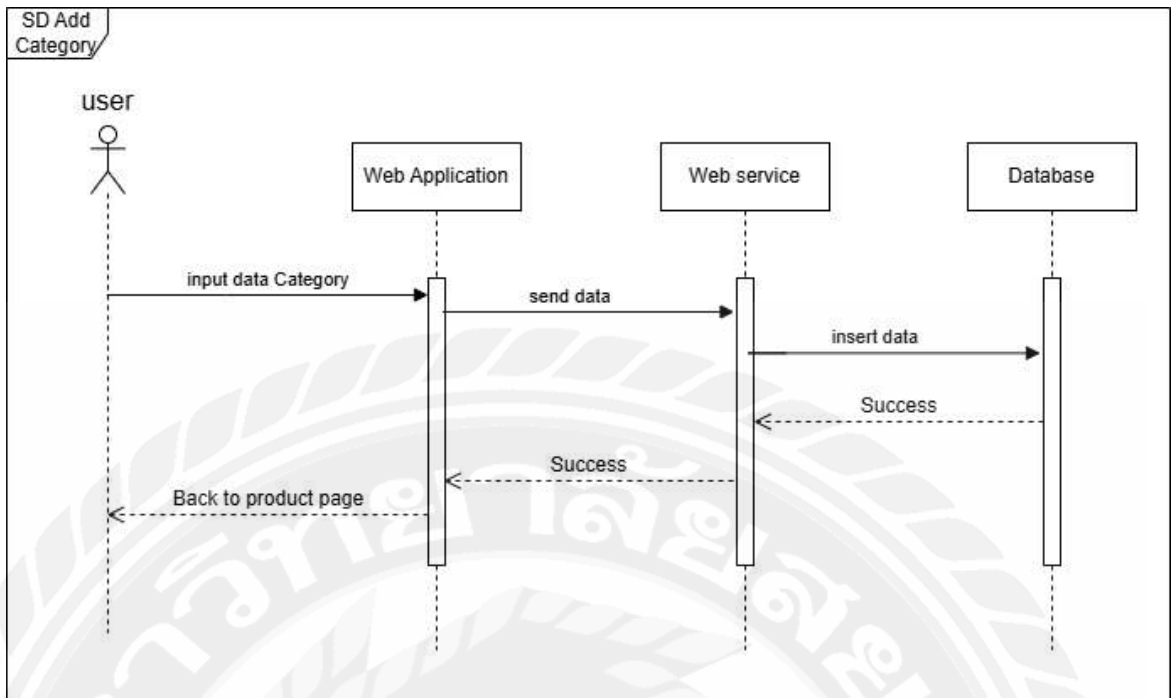
รูปที่ 3.6 Sequence Diagram: View user



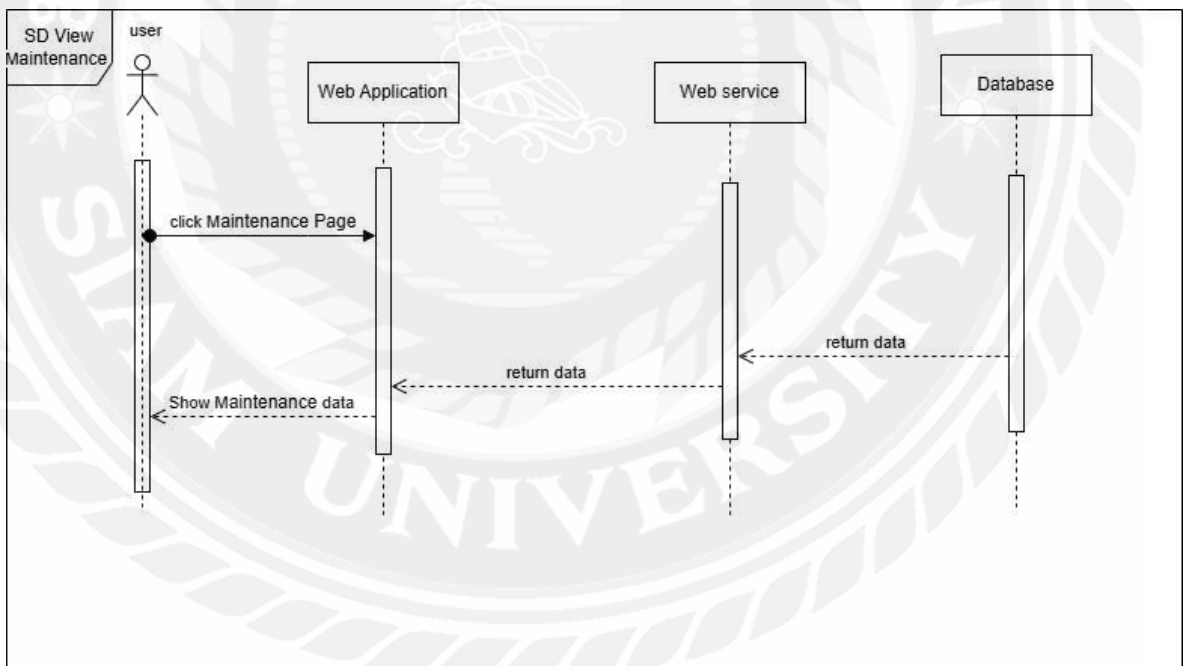
รูปที่ 3.7 Sequence Diagram: View Product



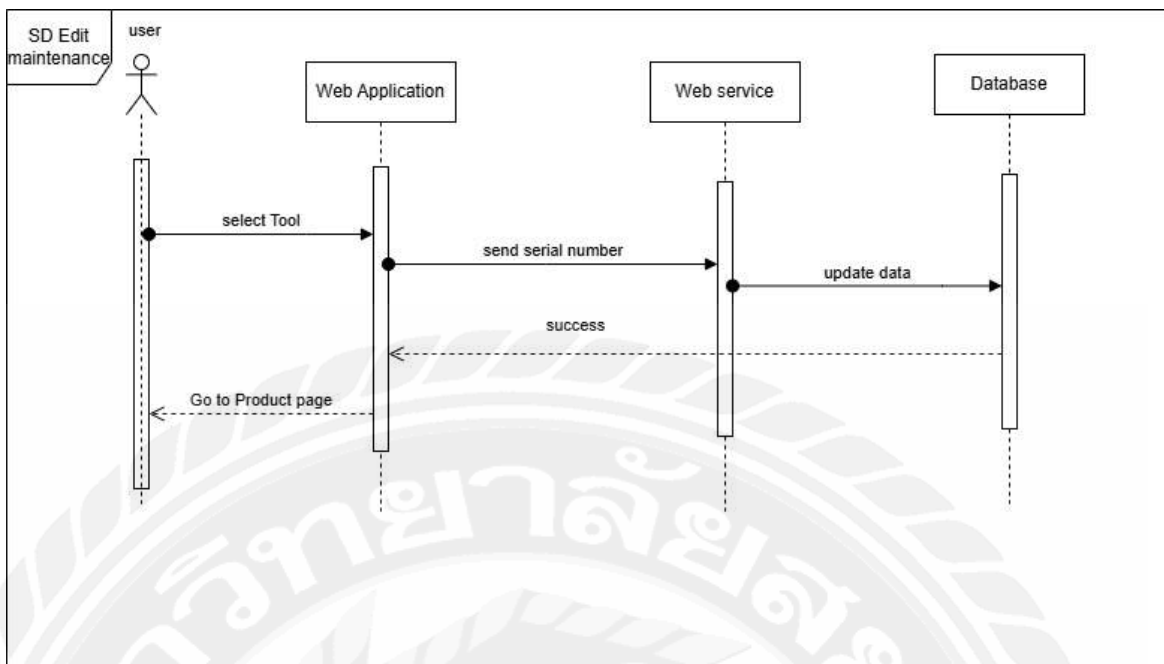
รูปที่ 3.8 Sequence Diagram: Add Room



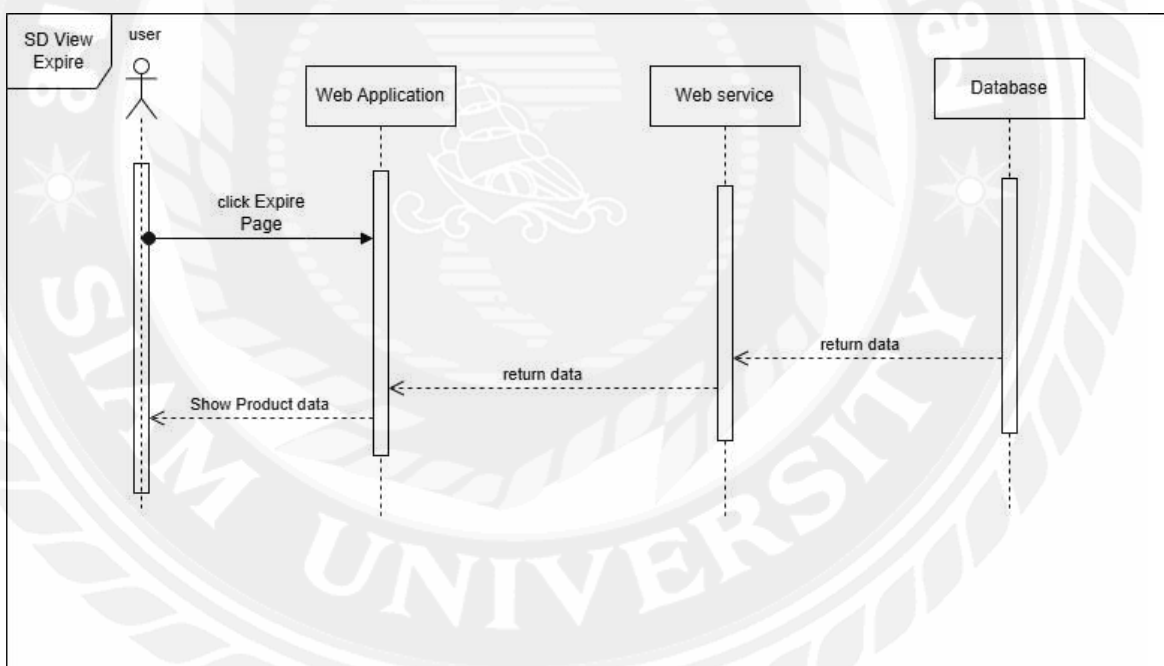
รูปที่ 3.9 Sequence Diagram: Add Category



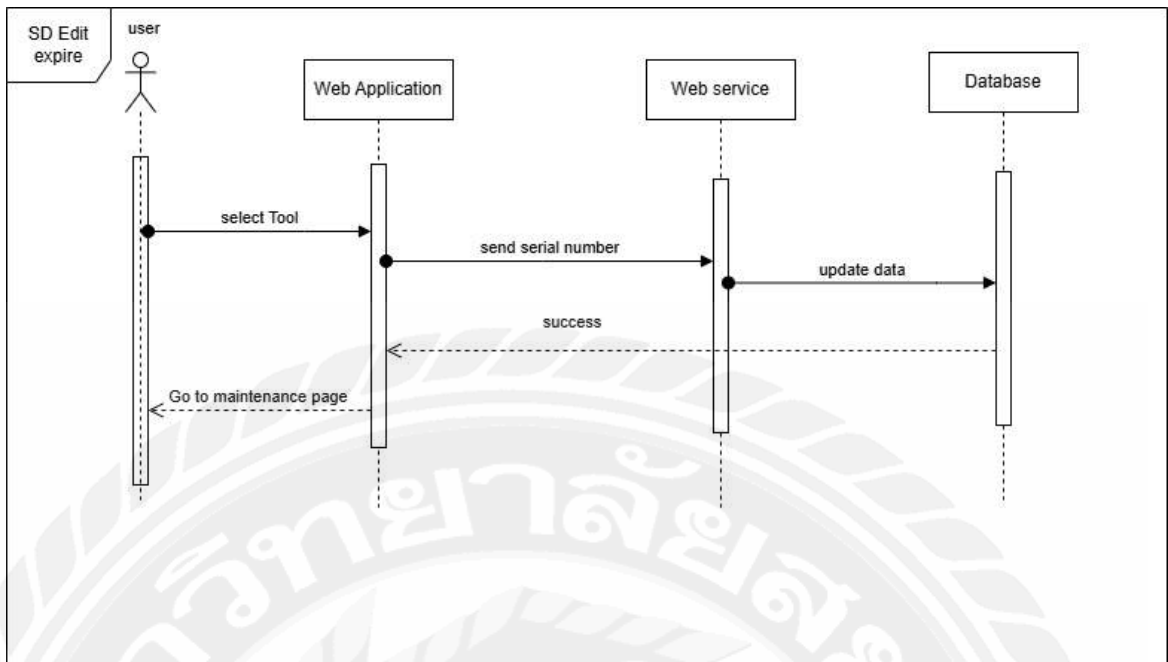
รูปที่ 3.10 Sequence Diagram: View Maintenance



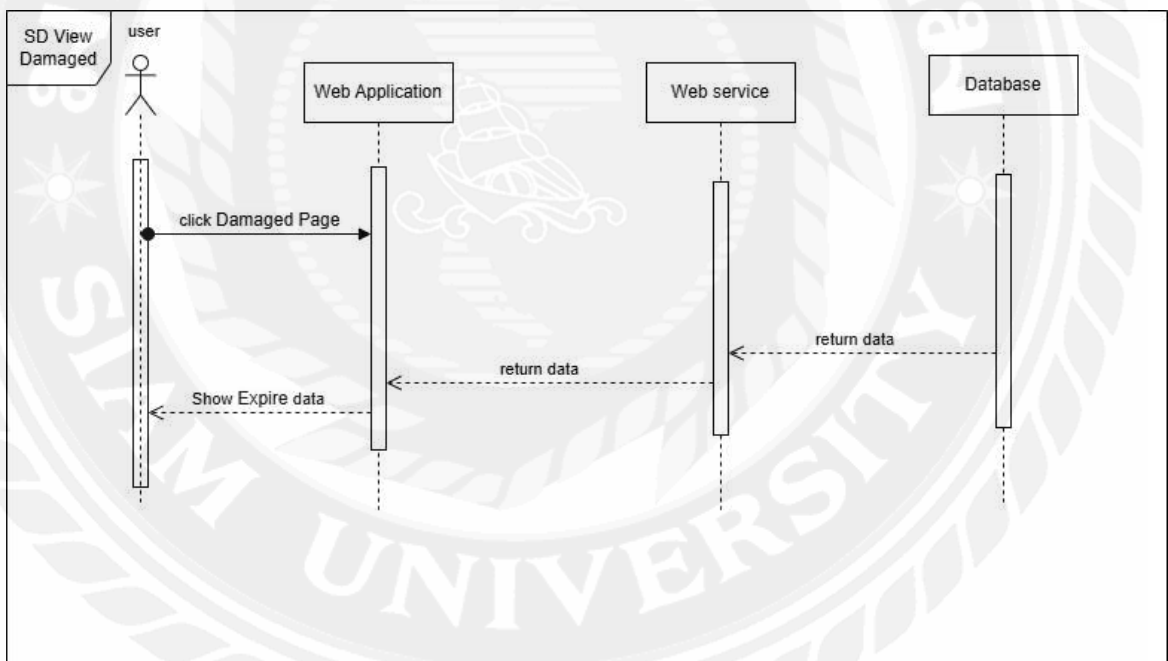
รูปที่ 3.11 Sequence Diagram: Edit Maintenance



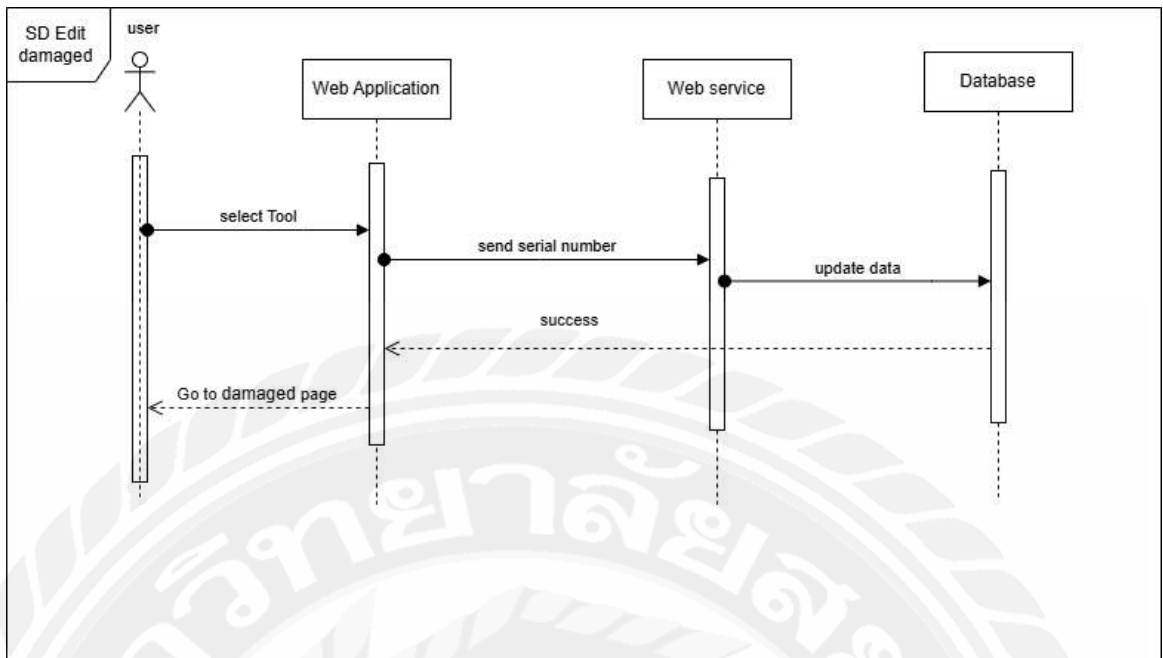
รูปที่ 3.12 Sequence Diagram: View Expire



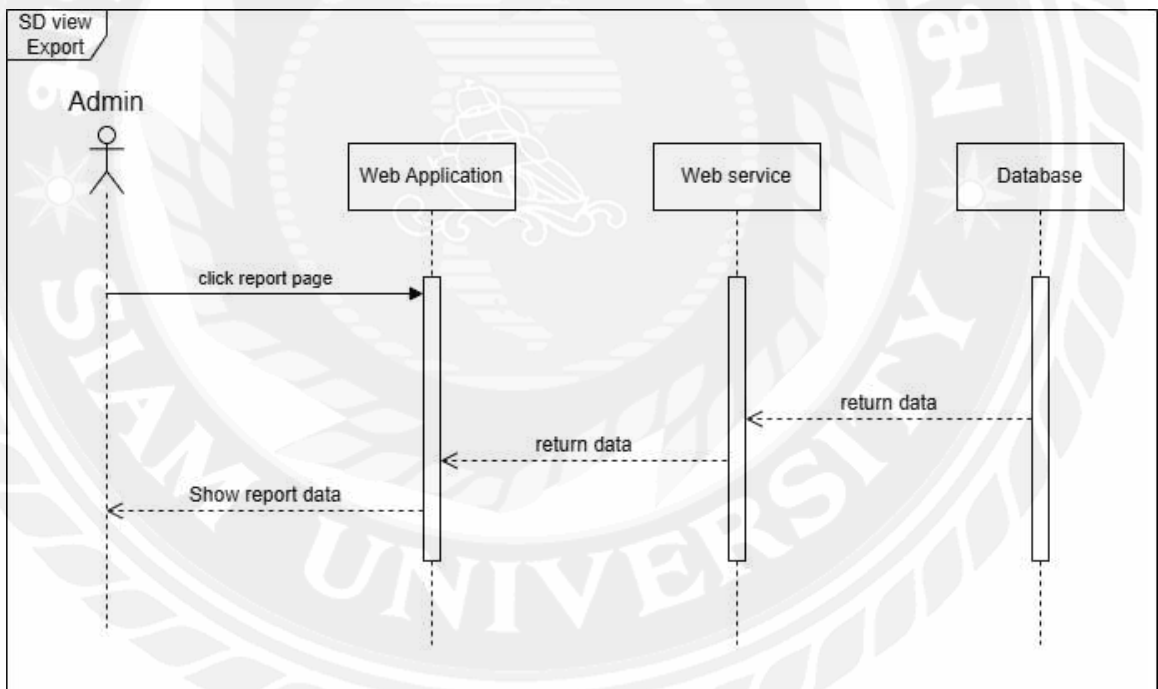
รูปที่ 3.13 Sequence Diagram: Edit Expire



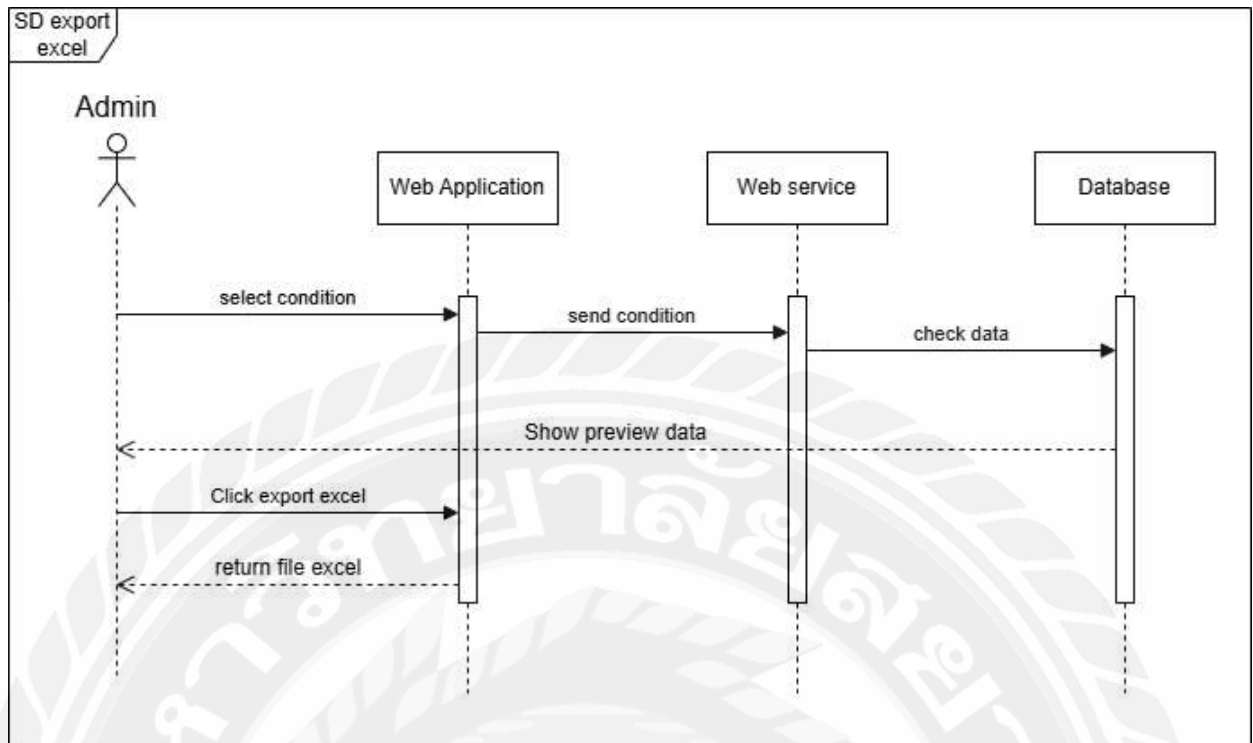
รูปที่ 3.14 Sequence Diagram: View Damaged



รูปที่ 3.15 Sequence Diagram: Edit Damaged

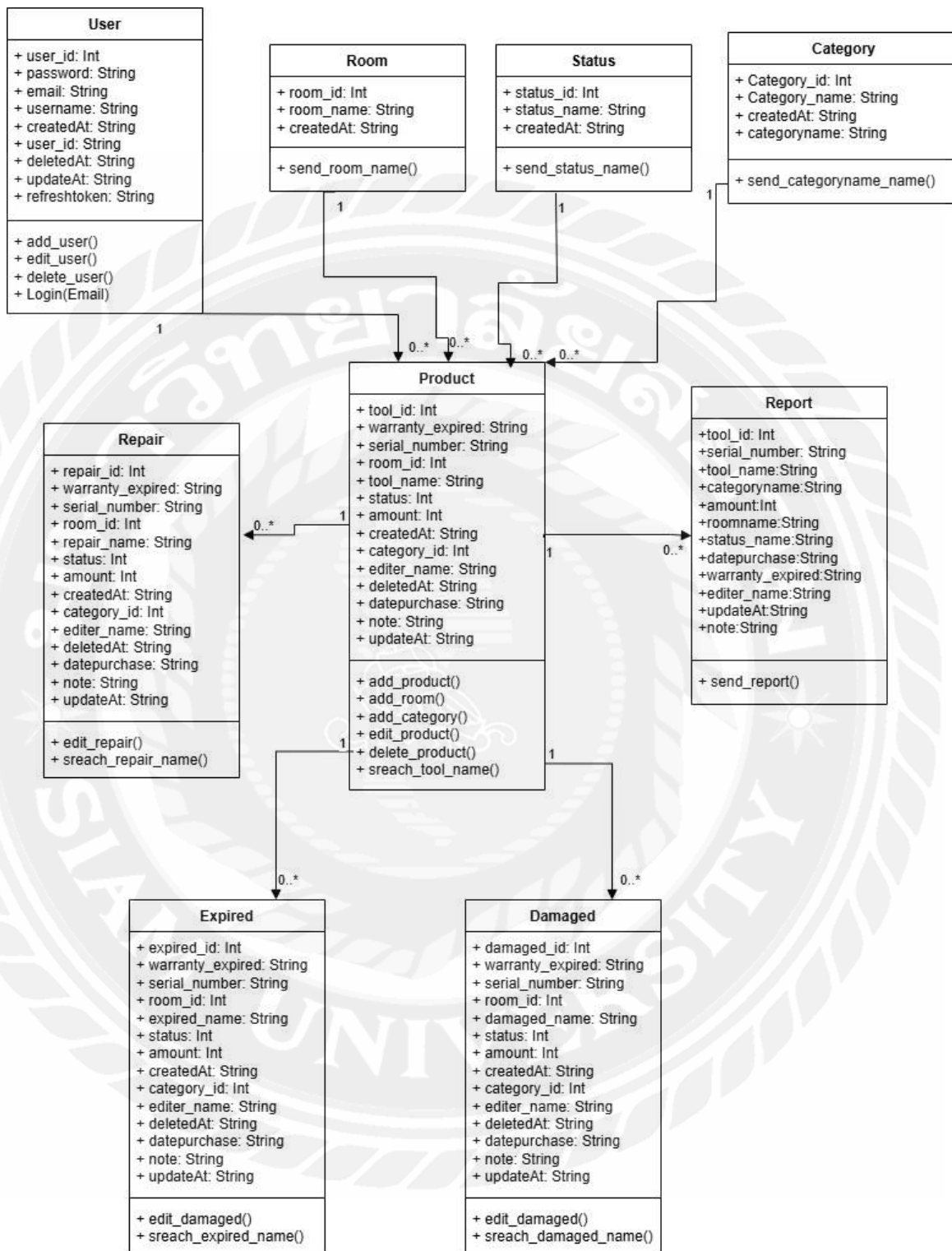


รูปที่ 3.16 Sequence Diagram: View Report



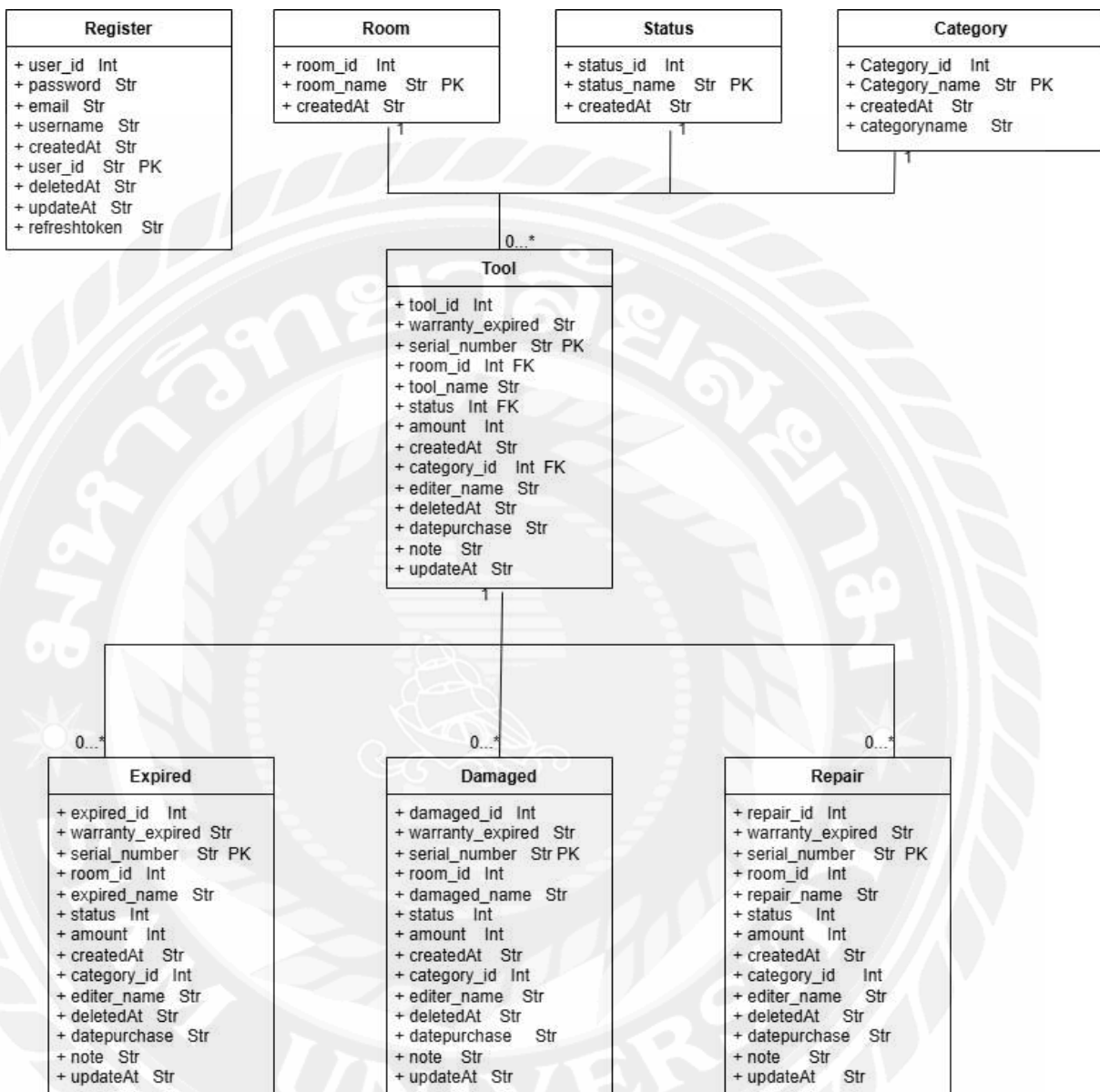
รูปที่ 3.17 Sequence Diagram: Export Excel

3.5 แผนภาพแสดงองค์ประกอบคลาส (Class Diagram)



รูปที่ 3.18 Class diagram ระบบจัดเก็บข้อมูลทรัพย์สิน กรณีศึกษา ภาควิชาวิทยาการคอมพิวเตอร์

3.6 โครงสร้างฐานข้อมูลและความสัมพันธ์ระหว่างเอนทิตี (Entity Relationship Diagram)



รูปที่ 3.19 Entity Relationship Diagram ของระบบจัดเก็บข้อมูลทรัพย์สิน กรณีศึกษา ภาควิชา
 วิทยาการคอมพิวเตอร์ มหาวิทยาลัยสยาม

บทที่ 4

การออกแบบทางกายภาพ

4.1 การออกแบบฐานข้อมูล (Database Design)

การออกแบบฐานข้อมูลเชิงสัมพันธ์ (Relation Database System) เป็นการจัดการข้อมูลต่างๆ ให้อยู่ในรูปของตาราง 2 มิติ ประกอบด้วย แถว (Row) และคอลัมน์ (Column) และมีการกำหนดความสัมพันธ์ระหว่างข้อมูล (Relationship) โดยสามารถอธิบายได้ดังนี้

1. ชื่อของข้อมูล (Field Name)
2. ชนิดของข้อมูล (Type)
3. ขนาดของข้อมูล (Length)
4. คำอธิบายของข้อมูล (Description)
5. อ้างอิงข้อมูล (Reference)

ตารางที่ 4.1 รายละเอียดของตารางข้อมูล Register

Table Name: Register			
Field Name	Type	Length	Description
User_id	Integer	-	Id ผู้ใช้งาน
password	String	20	รหัสผ่านผู้ใช้งาน
createdAt	Timestamp	-	วันที่สมัคร
refreshToken	String	500	Token ผู้ใช้งาน
Email	String	20	อีเมล ผู้ใช้งาน
username	String	20	ชื่อผู้ใช้งาน
userid	String	20	รหัสประจำตัวผู้ใช้งาน
deletedAt	Timestamp	-	วันที่ลบผู้ใช้งาน
Primary Key : userid			
Foreign Key : -			

ตารางที่ 4.2 รายละเอียดของตารางข้อมูล Room

Table Name: Room			
Field Name	Type	Length	Description
Room_id	Integer	-	Id ห้องเรียน
Roomname	String	20	ชื่อห้องเรียน
createdAt	Timestamp	-	วันที่เพิ่มห้องเรียนเข้าในระบบ
Primary Key : Roomname			
Foreign Key : -			

ตารางที่ 4.3 รายละเอียดของตารางข้อมูล Status

Table Name: Status			
Field Name	Type	Length	Description
status_id	Integer	-	Id สถานะ
Status_name	String	20	ชื่อสถานะ
createdAt	Timestamp	-	วันที่เพิ่มข้อมูลสถานะ
Primary Key : Status_name			
Foreign Key : -			

ตารางที่ 4.4 รายละเอียดของตารางข้อมูล Category

Table Name: Category			
Field Name	Type	Length	Description
category_id	Integer	-	Id สถานะ
category_name	String	20	ชื่อสถานะ (ตัวย่อภาษาอังกฤษ)
categoryname	String	20	ชื่อสถานะ(ภาษาไทย)
createdAt	Timestamp	-	วันที่เพิ่มข้อมูลสถานะ
Primary Key : category_name			
Foreign Key : -			

ตารางที่ 4.5 รายละเอียดของตารางข้อมูล Product

Table Name: Product			
Field Name	Type	Length	Description
tool_id	Integer	-	Id อุปกรณ์
warranty_expired	String	20	วันหมดอายุ/ ประกัน
serial_number	String	20	เลขเฉพาะของอุปกรณ์
createdAt	Timestamp	-	วันที่เพิ่มข้อมูลอุปกรณ์
room_id	Integer	-	Id ห้อง
tool_name	String	20	ชื่ออุปกรณ์
status	Integer	-	สถานะอุปกรณ์
amount	Integer	-	จำนวนอุปกรณ์
category_id	Integer	-	Id ประเภท
editer_name	String	50	ชื่อผู้แก้ไข
deletedAt	Timestamp	-	วันที่ลบข้อมูล
datepurchase	String	50	วันที่ซื้ออุปกรณ์
note	String	500	Note สำหรับเขียนรายละเอียดเพิ่มเติม
updateAt	Timestamp	-	วันที่อัปเดตข้อมูล
Primary Key : serial_number			
Foreign Key : room_id, status, category_id			

ตารางที่ 4.6 รายละเอียดของตารางข้อมูล Repair

Table Name: Product			
Field Name	Type	Length	Description
repair_id	Integer	-	Id อุปกรณ์ที่ซ่อมบำรุง
warranty_expired	String	20	วันหมดอายุ/ ประกัน
serial_number	String	20	เลขเฉพาะของอุปกรณ์
createdAt	Timestamp	-	วันที่เพิ่มข้อมูลอุปกรณ์
room_id	Integer	-	Id ห้อง
repair_name	String	20	ชื่ออุปกรณ์ที่ซ่อมบำรุง
status	Integer	-	สถานะอุปกรณ์
amount	Integer	-	จำนวนอุปกรณ์
category_id	Integer	-	Id ประเภท
editer_name	String	50	ชื่อผู้แก้ไข
deletedAt	Timestamp	-	วันที่ลบข้อมูล
datepurchase	String	50	วันที่ซื้ออุปกรณ์
note	String	500	Note สำหรับเขียนรายละเอียดเพิ่มเติม
updateAt	Timestamp	-	วันที่อัปเดตข้อมูล
Primary Key : serial_number			
Foreign Key : room_id, status, category_id			

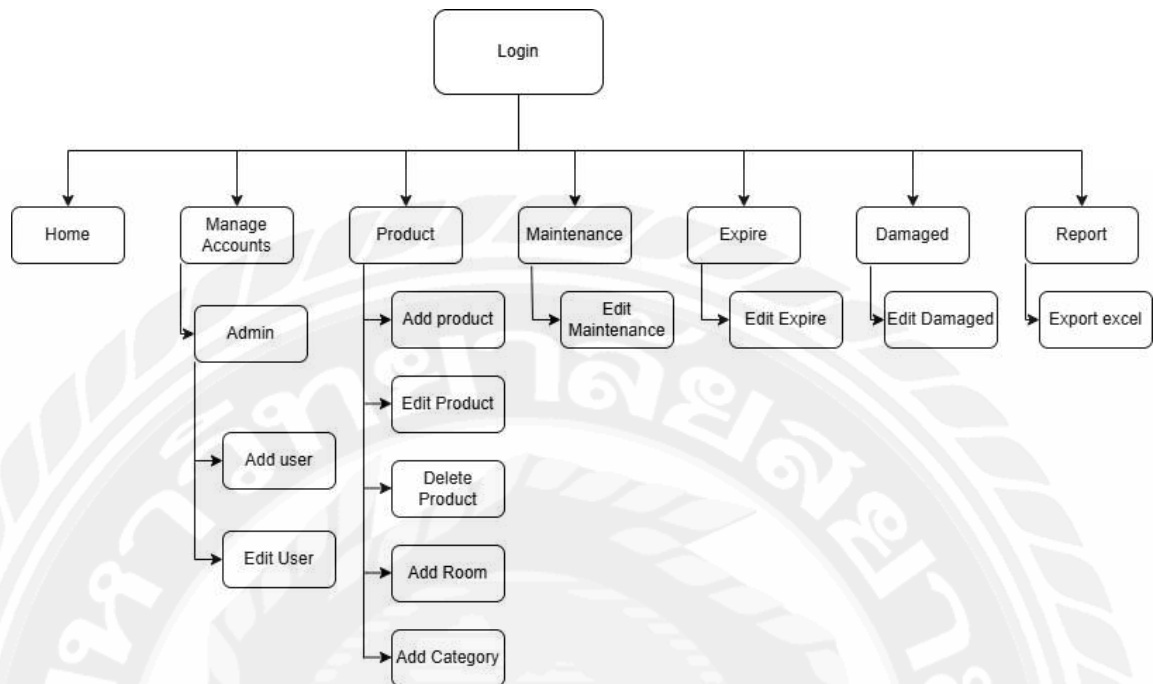
ตารางที่ 4.7 รายละเอียดของตารางข้อมูล Expired

Table Name: Product			
Field Name	Type	Length	Description
expired_id	Integer	-	Id อุปกรณ์ที่หมดอายุ/ หมดประกัน
warranty_expired	String	20	วันหมดอายุ/ ประกัน
serial_number	String	20	เลขเฉพาะของอุปกรณ์
createdAt	Timestamp	-	วันที่เพิ่มข้อมูลอุปกรณ์
room_id	Integer	-	Id ห้อง
expired_name	String	20	ชื่ออุปกรณ์ที่หมดอายุ/ หมดประกัน
status	Integer	-	สถานะอุปกรณ์
amount	Integer	-	จำนวนอุปกรณ์
category_id	Integer	-	Id ประเภท
editer_name	String	50	ชื่อผู้แก้ไข
deletedAt	Timestamp	-	วันที่ลบข้อมูล
datepurchase	String	50	วันที่ซื้ออุปกรณ์
note	String	500	Note สำหรับเขียนรายละเอียดเพิ่มเติม
updateAt	Timestamp	-	วันที่อัปเดตข้อมูล
Primary Key : serial_number			
Foreign Key : room_id, status, category_id			

ตารางที่ 4.8 รายละเอียดของตารางข้อมูล Damaged

Table Name: Product			
Field Name	Type	Length	Description
damaged_id	Integer	-	Id อุปกรณ์ที่พัง
warranty_expired	String	20	วันหมดอายุ/ ประกัน
serial_number	String	20	เลขเฉพาะของอุปกรณ์
createdAt	Timestamp	-	วันที่เพิ่มข้อมูลอุปกรณ์
room_id	Integer	-	Id ห้อง
damaged_name	String	20	ชื่ออุปกรณ์ที่พัง
status	Integer	-	สถานะอุปกรณ์
amount	Integer	-	จำนวนอุปกรณ์
category_id	Integer	-	Id ประเภท
editer_name	String	50	ชื่อผู้แก้ไข
deletedAt	Timestamp	-	วันที่ลบข้อมูล
datepurchase	String	50	วันที่ซื้ออุปกรณ์
note	String	500	Note สำหรับเขียนรายละเอียดเพิ่มเติม
updateAt	Timestamp	-	วันที่อัปเดตข้อมูล
Primary Key : serial_number			
Foreign Key : room_id, status, category_id			

4.2 แผนผังโครงสร้างของเว็บไซต์ (Site Map)



รูปที่ 4.1 แผนผังโครงสร้างของเว็บไซต์

4.3 รายละเอียดของโครงสร้างเว็บไซต์ (Site Map Detail)

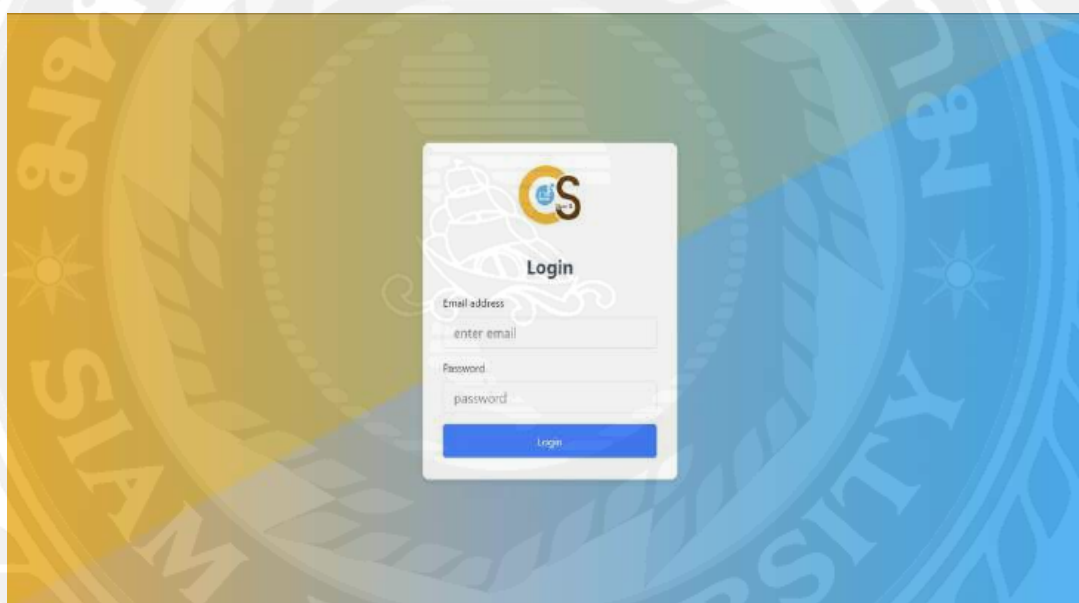
ตารางที่ 4.9 รายละเอียดของโครงสร้างของเว็บไซต์

ชื่อรายการ	คำอธิบาย
pages\login.vue	หน้าเข้าสู่ระบบ
pages\admin\dashboard.vue	หน้าหลัก
pages\product.vue	หน้ารายละเอียดรายการทรัพย์สิน
pages\expired.vue	หน้ารายละเอียดรายการทรัพย์สินที่หมดประกัน/ อายุ
pages\maintain.vue	หน้ารายละเอียดรายการทรัพย์สินแจ้งซ่อม
pages\damaged.vue	หน้ารายละเอียดรายการทรัพย์สินที่เสียหาย
pages\account.vue	หน้ารายละเอียดรายชื่อผู้ใช้
pages\report.vue	หน้ารายละเอียดรายการทรัพย์สินที่ต้องการ Export
components\Sidebar.vue	แสดงรายชื่อเมนู
components\Navbar.vue	แสดงชื่อผู้ใช้ที่กำลังใช้งาน
components\Card.vue	แสดงจำนวนของแต่ละสถานะ
components\Table.vue	ตารางแสดงรายการทรัพย์สิน
components\Table.vue\Add Room	ปุ่มเพิ่มรหัสห้อง
components\Table.vue\Add Category	ปุ่มเพิ่มเพิ่มประเภท
components\Table.vue\Add Product	ปุ่มเพิ่มทรัพย์สิน
components\Table.vue\Delete Product	ปุ่มลบทรัพย์สิน
components\Table.vue\Edit	ปุ่มแก้ไขข้อมูล
components\ExpiredTable.vue	ตารางแสดงรายการทรัพย์สิน
components\ExpiredTable.vue\Edit	ปุ่มแก้ไขข้อมูล
components\MaintainTable.vue	ตารางแสดงรายการทรัพย์สิน
components\MaintainTable.vue\Edit	ปุ่มแก้ไขข้อมูล
components\DamagedTable.vue	ตารางแสดงรายการทรัพย์สิน
components\DamagedTable.vue\Edit	ปุ่มแก้ไขข้อมูล
components\AccountTable.vue	ตารางแสดงรายชื่อผู้ใช้

components\AccountTable.vue\Add Account	ปุ่มเพิ่มผู้ใช้
components\AccountTable.vue\Edit user	ปุ่มแก้ไขข้อมูลผู้ใช้
components\ReportTable.vue	ตารางแสดงรายการทรัพย์สินที่ต้องการ Export
components\ReportTable.vue\Export Xcel	ปุ่มออกรายงานทรัพย์สิน

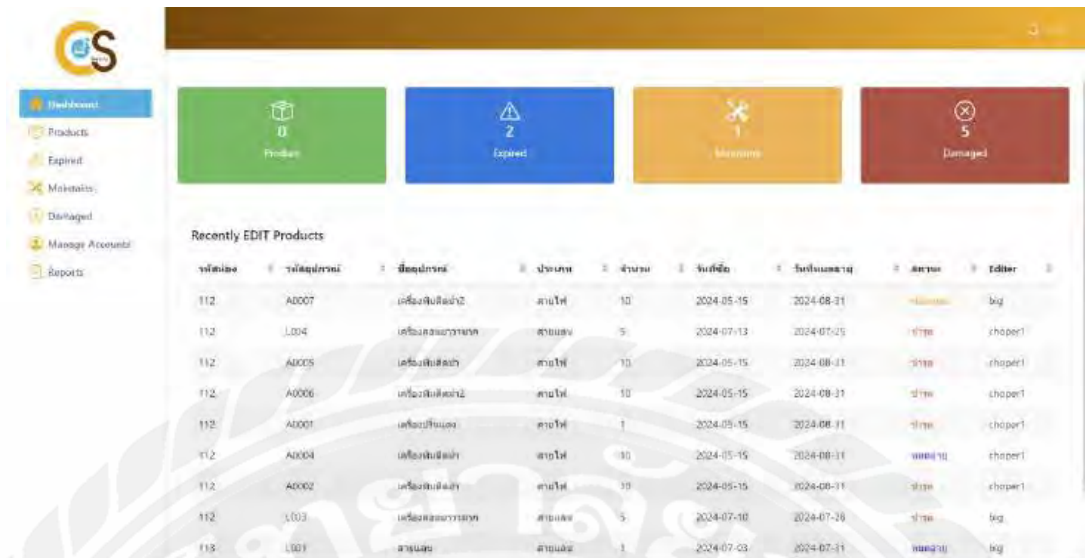
4.4 การออกแบบส่วนติดต่อกับผู้ใช้ (User Interface Design)

คือ ส่วนต่อประสานกับผู้ใช้งาน หรือ ส่วนที่ใช้ในการเชื่อมต่อกับผู้ใช้งาน กล่าวคือ ส่วนที่ให้ผู้ใช้งานสามารถโต้ตอบกับการใช้งานผลิตภัณฑ์ได้ ซึ่งจะมุ่งเน้นไปที่เรื่องของหน้าตา การออกแบบ และการดีไซน์ ยกตัวอย่างเช่น หน้าจอ แพลตฟอร์ม เมนู ฟอรัมต่าง ๆ การวางภาพ ขนาด ตัวอักษร ปุ่ม แป้นพิมพ์ เสียง หรือแม้แต่แสงไฟ เป็นต้น



รูปที่ 4.2 หน้าจอเข้าสู่ระบบ (Login)

จากรูปที่ 4.2 แสดงหน้าจอเข้าสู่ระบบ (Login) ซึ่งเป็นส่วนของหน้าแรกสำหรับผู้ดูแลระบบ และผู้ใช้เข้า โดยผู้ใช้งานสามารถล็อกอินด้วย อีเมล (Email) และรหัสผ่าน (Password) ที่ได้รับจากผู้ดูแลระบบ

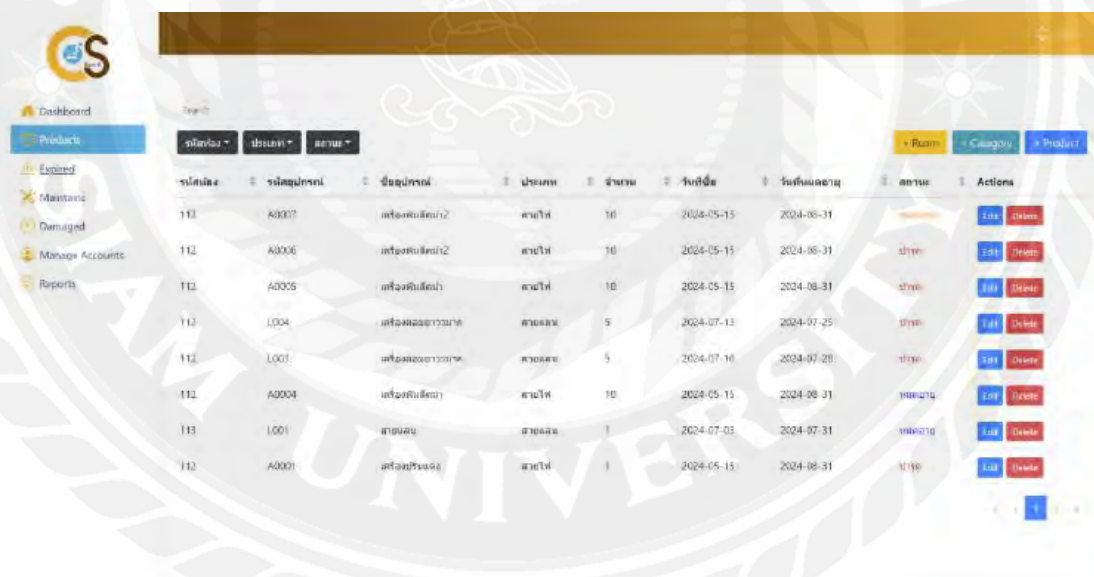


The dashboard features a sidebar with navigation options: Dashboard, Products, Expired, Maintains, Damaged, Manage Accounts, and Reports. The main area displays four status cards: Products (0), Expired (2), Maintains (1), and Damaged (5). Below these is a table titled 'Recently EDIT Products'.

รหัสสินค้า	รหัสอุปกรณ์	ชื่ออุปกรณ์	ประเภท	จำนวน	วันที่ซื้อ	วันหมดอายุ	สถานะ	Editor
112	A0007	เครื่องปรับอากาศ	สาขาไฟ	10	2024-05-15	2024-08-31	完好	big
112	L004	เครื่องลดความชื้น	สาขาแสง	5	2024-07-13	2024-07-25	ชำรุด	shoper1
112	A0005	เครื่องปรับอากาศ	สาขาไฟ	10	2024-05-15	2024-08-31	ชำรุด	shoper1
112	A0006	เครื่องปรับอากาศ	สาขาไฟ	10	2024-05-15	2024-08-31	ชำรุด	shoper1
112	A0001	เครื่องปรับอากาศ	สาขาไฟ	1	2024-05-15	2024-08-31	ชำรุด	shoper1
112	A0004	เครื่องปรับอากาศ	สาขาไฟ	10	2024-05-15	2024-08-31	完好	shoper1
112	A0002	เครื่องปรับอากาศ	สาขาไฟ	10	2024-05-15	2024-08-31	ชำรุด	shoper1
112	L003	เครื่องลดความชื้น	สาขาแสง	5	2024-07-10	2024-07-26	ชำรุด	big
113	L001	สาขาแสง	สาขาแสง	1	2024-07-03	2024-07-31	完好	big

รูปที่ 4.3 หน้าแรกของเว็บไซต์ (Dashboard)

จากรูปที่ 4.3 หน้าแรก (Dashboard) จะแสดงจำนวนของข้อมูลสถานะและรายการทรัพย์สินที่มีการเปลี่ยนแปลงหรือเพิ่มเข้ามาล่าสุดที่มีอยู่ในระบบฐานข้อมูลโดยจะแบ่งเป็น 2 ส่วนประกอบด้วย 1) ข้อมูลจำนวนสถานะ (Status) 2) ข้อมูลทรัพย์สิน (Product)

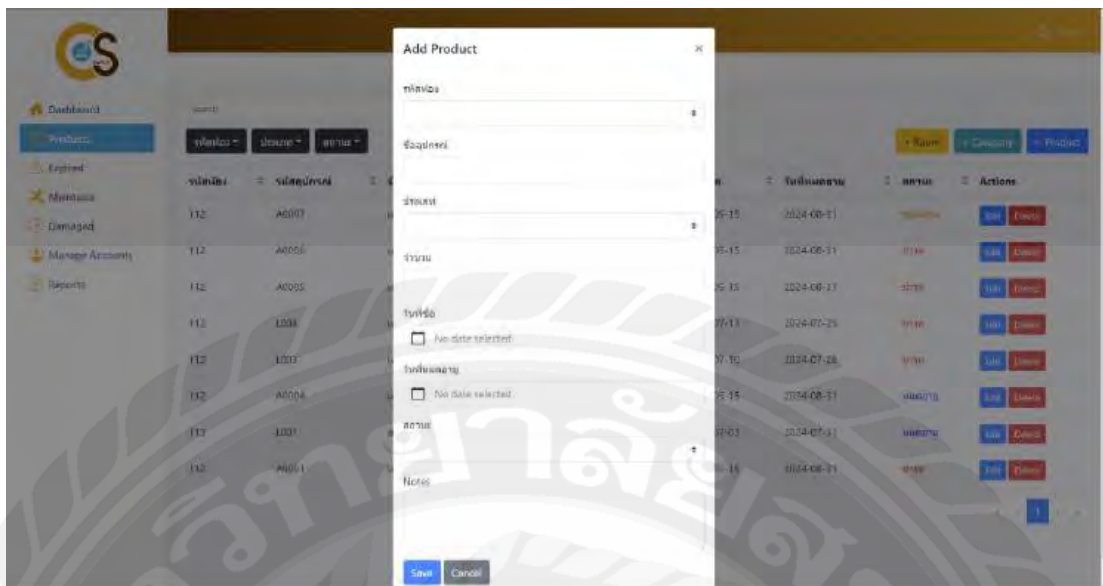


The products page includes a sidebar with navigation options: Dashboard, Products, Expired, Maintains, Damaged, Manage Accounts, and Reports. The main area features filter buttons for 'รหัสสินค้า', 'ประเภท', and 'สถานะ'. Below the filters is a table of products with 'Actions' (Edit, Delete) for each row.

รหัสสินค้า	รหัสอุปกรณ์	ชื่ออุปกรณ์	ประเภท	จำนวน	วันที่ซื้อ	วันหมดอายุ	สถานะ	Actions
113	A0007	เครื่องปรับอากาศ	สาขาไฟ	10	2024-05-15	2024-08-31	完好	Edit Delete
112	A0006	เครื่องปรับอากาศ	สาขาไฟ	10	2024-05-15	2024-08-31	ชำรุด	Edit Delete
112	A0005	เครื่องปรับอากาศ	สาขาไฟ	10	2024-05-15	2024-08-31	ชำรุด	Edit Delete
112	L004	เครื่องลดความชื้น	สาขาแสง	5	2024-07-13	2024-07-25	ชำรุด	Edit Delete
112	L003	เครื่องลดความชื้น	สาขาแสง	5	2024-07-10	2024-07-26	ชำรุด	Edit Delete
112	A0004	เครื่องปรับอากาศ	สาขาไฟ	10	2024-05-15	2024-08-31	完好	Edit Delete
113	L001	สาขาแสง	สาขาแสง	1	2024-07-03	2024-07-31	完好	Edit Delete
112	A0001	เครื่องปรับอากาศ	สาขาไฟ	1	2024-05-15	2024-08-31	ชำรุด	Edit Delete

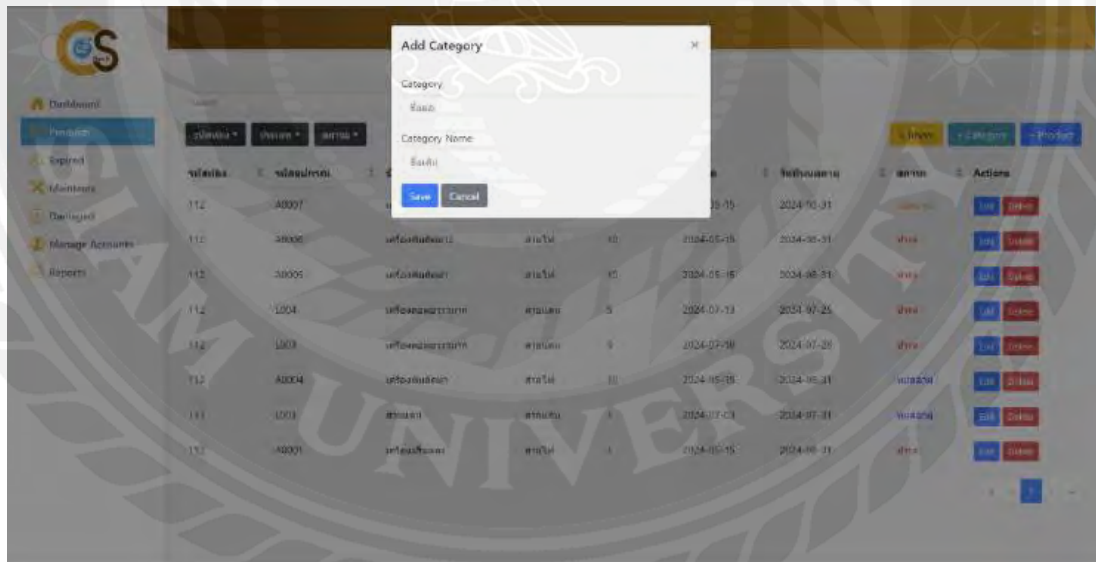
รูปที่ 4.4 หน้ารายการทรัพย์สิน (Products)

จากรูปที่ 4.4 หน้ารายการทรัพย์สิน (Products) จะแสดงข้อมูลรายการทรัพย์สินที่มีอยู่ในระบบฐานข้อมูลโดยจะแบ่งเป็น 2 ส่วนประกอบด้วย 1) ฟังก์ชันต่างๆ (Filter room_id, Filter category_id, Filter status_id) 2) แสดงข้อมูลต่างๆ ได้แก่ รหัสห้อง (roomname) รหัสอุปกรณ์ (product_id) ชื่ออุปกรณ์ (product_name) ประเภท (category_id) จำนวน (amount) วันที่ซื้อ (datepur) วันหมดอายุ (warranty_expired) สถานะ(status)



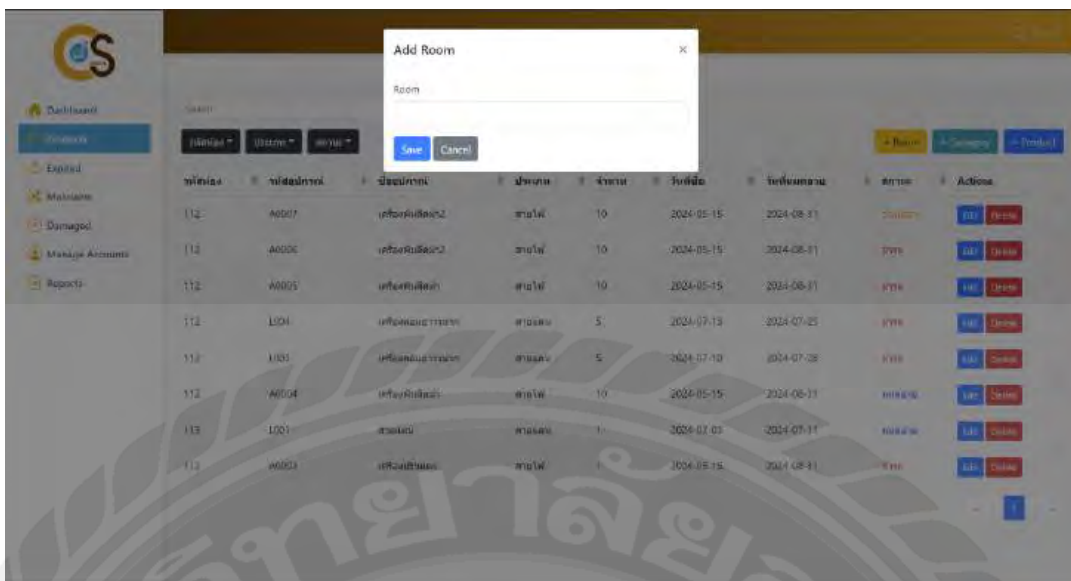
รูปที่ 4.5 หน้าเพิ่มรายการทรัพย์สิน (Add Products)

จากรูปที่ 4.5 หน้าเพิ่มรายการทรัพย์สิน (Add Products) จะแสดงหน้าสำหรับเพิ่มข้อมูลทรัพย์สิน โดยผู้ใช้งานต้องกรอกข้อมูลของทรัพย์สินเพื่อเพิ่มเข้าในระบบ



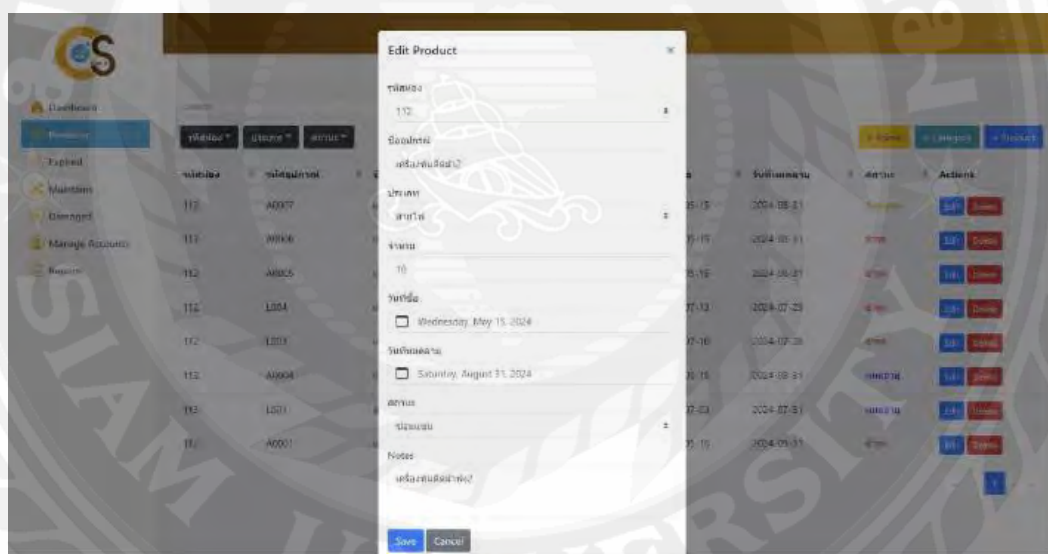
รูปที่ 4.6 หน้าเพิ่มประเภททรัพย์สิน (Add Category)

จากรูปที่ 4.6 หน้าเพิ่มประเภททรัพย์สิน (Add Category) แสดงหน้าสำหรับเพิ่มข้อมูลประเภทของทรัพย์สิน โดยผู้ใช้งานต้องกรอกข้อมูลเพื่อเพิ่มเข้าในระบบ



รูปที่ 4.7 หน้าเพิ่มรหัสห้อง (Add Room)

จากรูปที่ 4.7 หน้าเพิ่มรหัสห้อง (Add Room) แสดงหน้าสำหรับเพิ่มข้อมูลรหัสห้องโดยผู้ใช้งานกรอกข้อมูลเพื่อเพิ่มข้อมูลห้องเรียนเข้าสู่ระบบ



รูปที่ 4.8 หน้าแก้ไขข้อมูลทรัพย์สิน (Edit Product)

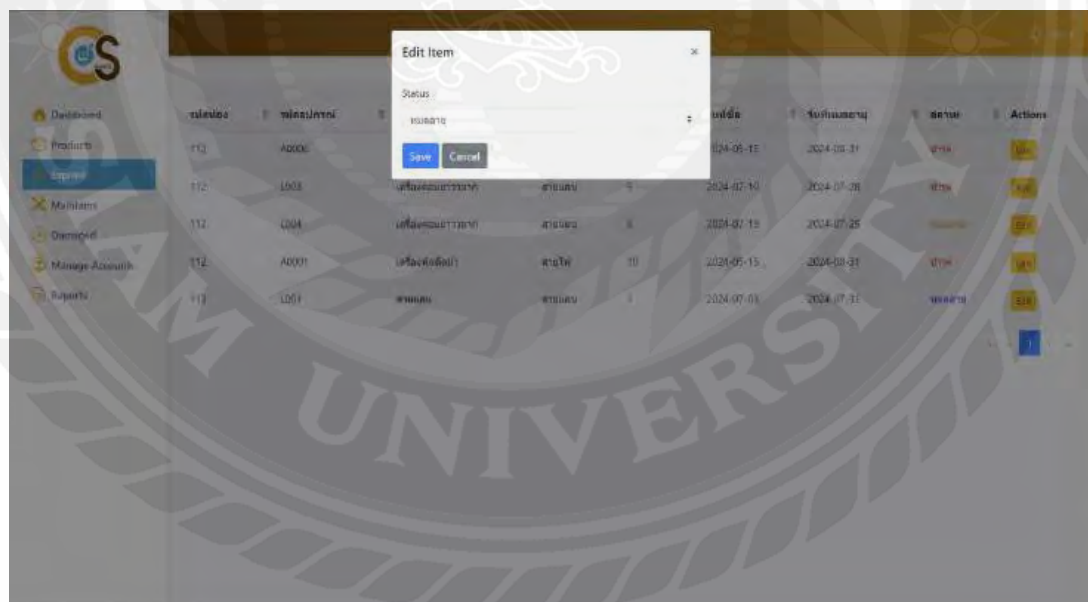
จากรูปที่ 4.8 หน้าแก้ไขข้อมูลทรัพย์สิน (Edit Product) แสดงหน้าสำหรับแก้ไขข้อมูลทรัพย์สินโดยผู้ใช้งานกรอกข้อมูลเพื่อแก้ไขและบันทึกในระบบ



รหัสของ	รหัสผลิตภัณฑ์	ชื่อผลิตภัณฑ์	ประเภท	จำนวน	วันที่ซื้อ	วันที่หมดอายุ	สถานะ	Actions
112	A0006	เครื่องตัดหญ้า	ช่างไฟ	10	2024-05-15	2024-08-31	ชำรุด	EDIT
112	L003	เครื่องผสมยาปราบศัตรูพืช	ช่างผสม	5	2024-07-10	2024-07-28	ชำรุด	EDIT
112	L004	เครื่องผสมยาปราบศัตรูพืช	ช่างผสม	5	2024-07-13	2024-07-25	ชำรุด	EDIT
112	A0001	เครื่องตัดหญ้า	ช่างไฟ	10	2024-05-15	2024-08-31	ชำรุด	EDIT
112	L001	ช่างผสม	ช่างผสม	1	2024-07-03	2024-07-31	หมดอายุ	EDIT

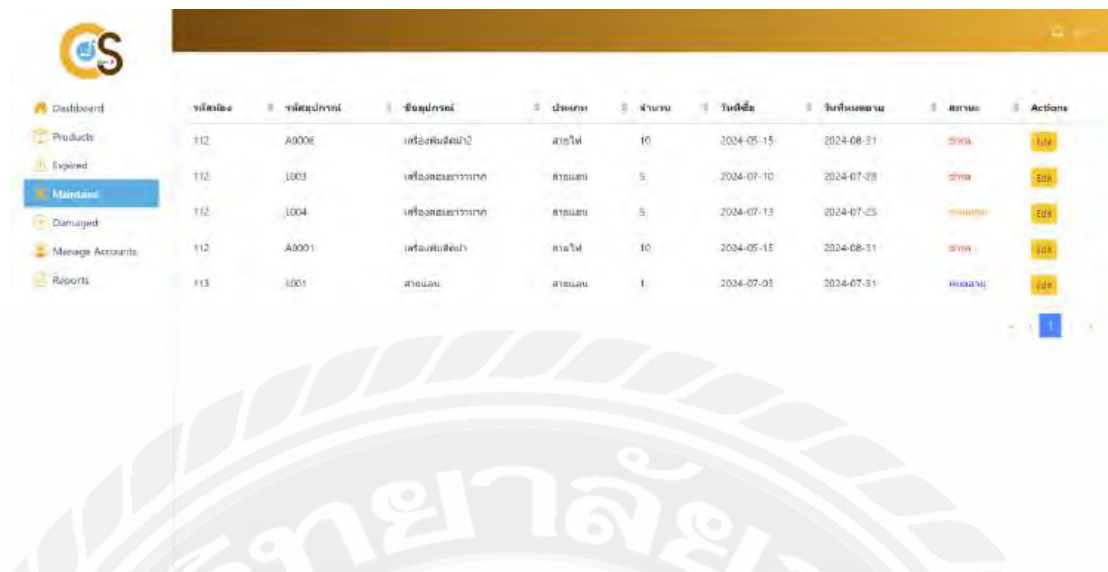
รูปที่ 4.9 หน้าแสดงข้อมูลทรัพย์สินที่มีสถานะหมดอายุ/ หมดประกัน (Expired)

จากรูปที่ 4.9 หน้าแสดงข้อมูลทรัพย์สินที่มีสถานะหมดอายุ/ หมดประกัน (Expired) แสดงข้อมูลต่างๆ ได้แก่ รหัสห้อง (roomname) รหัสอุปกรณ์ (product_id) ชื่ออุปกรณ์ (product_name) ประเภท (category_id) จำนวน (amount) วันที่ซื้อ (datepur) วันที่หมดประกัน (warranty_expired) สถานะ (status)



รูปที่ 4.10 หน้าแก้ไขข้อมูลทรัพย์สิน (Edit Product)

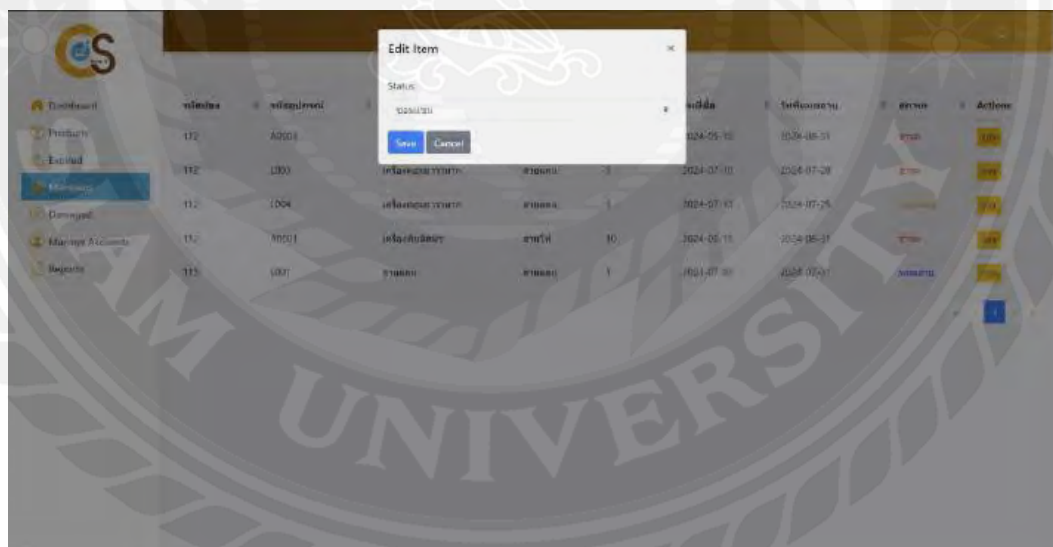
จากรูปที่ 4.10 หน้าแก้ไขข้อมูลทรัพย์สิน (Edit Product) แสดงหน้าสำหรับแก้ไขข้อมูล ทรัพย์สิน โดยผู้ใช้ต้องกรอกข้อมูลเพื่อแก้ไขและบันทึกในระบบ



รหัสห้อง	รหัสอุปกรณ์	ชื่ออุปกรณ์	ประเภท	จำนวน	วันที่ซื้อ	วันที่หมดอายุ	สถานะ	Actions
112	A0006	เครื่องปรับอากาศ	สาขาไฟ	10	2024-05-15	2024-08-31	完好	View
112	1003	เครื่องปรับอากาศ	สาขาไฟ	5	2024-07-10	2024-07-28	完好	Edit
112	1004	เครื่องปรับอากาศ	สาขาไฟ	5	2024-07-13	2024-07-25	完好	Edit
112	A0001	เครื่องปรับอากาศ	สาขาไฟ	10	2024-05-15	2024-08-31	完好	View
113	1001	สาขาไฟ	สาขาไฟ	1	2024-07-01	2024-07-31	完好	Edit

รูปที่ 4.11 หน้าแสดงข้อมูลทรัพย์สินที่มีสถานะรอซ่อม/ ส่งซ่อม (Maintain)

จากรูปที่ 4.11 หน้าแสดงข้อมูลทรัพย์สินที่มีสถานะรอซ่อม/ ส่งซ่อม (Maintain) แสดงข้อมูลต่างๆ ได้แก่ รหัสห้อง (roomname) รหัสอุปกรณ์ (product_id) ชื่ออุปกรณ์ (product_name) ประเภททรัพย์สิน (category_id) จำนวน (amount) วันที่ซื้อ (datepur) วันหมดการรับประกัน (warranty_expired) สถานะ (status)



รูปที่ 4.12 หน้าแก้ไขข้อมูลทรัพย์สิน (Edit Product)

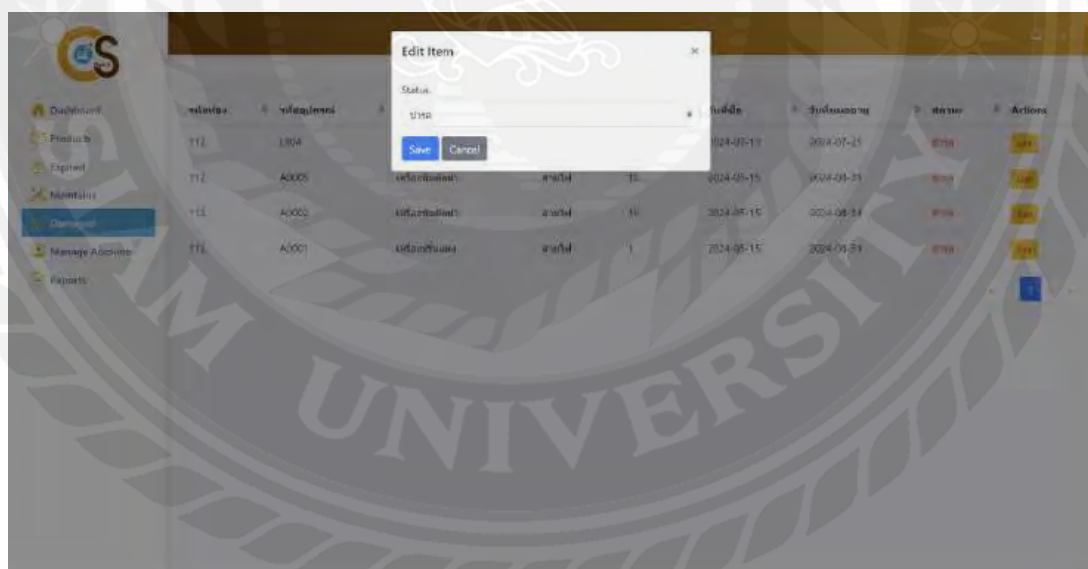
จากรูปที่ 4.12 หน้าแก้ไขข้อมูลทรัพย์สิน (Edit Product) แสดงหน้าสำหรับแก้ไขข้อมูลทรัพย์สิน โดยผู้ใช้งานต้องกรอกข้อมูลเพื่อแก้ไขและบันทึกในระบบ



หมายเลข	รหัสอุปกรณ์	ชื่ออุปกรณ์	ประเภท	จำนวน	วันที่ซื้อ	วันที่หมดอายุ	สถานะ	Actions
112	L004	เครื่องคอมพิวเตอร์	สายเคเบิ้ล	5	2024-07-13	2024-07-25	ชำรุด	Edit
112	A005	เครื่องปรับอากาศ	สายไฟ	10	2024-05-15	2024-08-31	ชำรุด	Edit
112	A002	เครื่องปรับอากาศ	สายไฟ	10	2024-05-15	2024-08-31	ชำรุด	Edit
112	A001	เครื่องปรับอากาศ	สายไฟ	1	2024-05-15	2024-08-31	ชำรุด	Edit

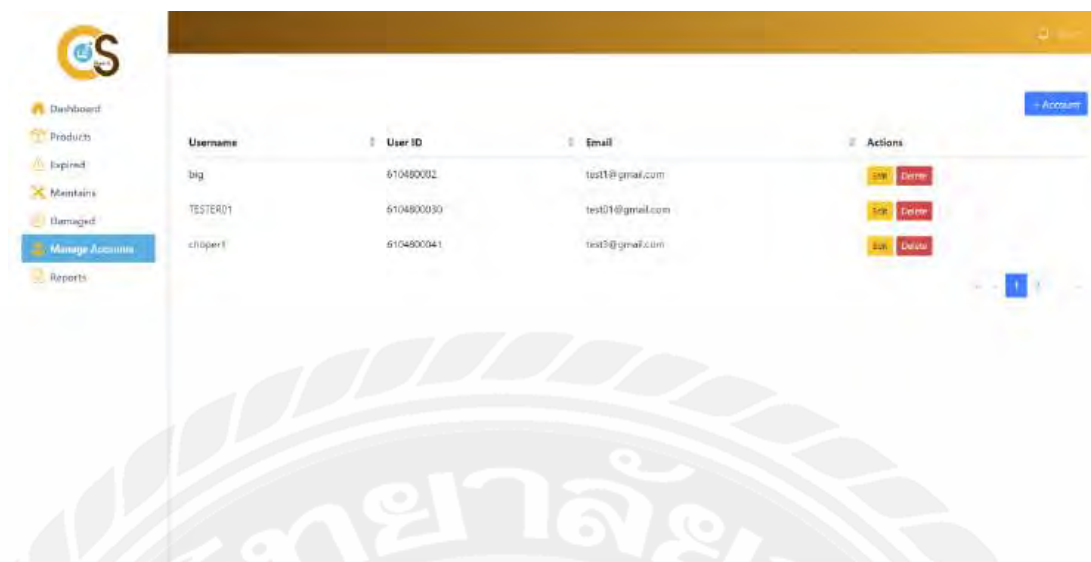
รูปที่ 4.13 หน้าแสดงข้อมูลทรัพย์สินที่มีสถานะชำรุดเสียหาย (Damaged)

จากรูปที่ 4.13 หน้าแสดงข้อมูลทรัพย์สินที่มีสถานะรอซ่อม/ส่งซ่อม (Maintain) แสดงข้อมูลต่างๆ ได้แก่ รหัสห้อง (roomname) รหัสอุปกรณ์ (product_id) ชื่ออุปกรณ์ (product_name) ประเภท (catagory_id) จำนวน (amount) วันที่ซื้อ (datepur) วันหมดประกัน (warranty_expired) สถานะ(status)



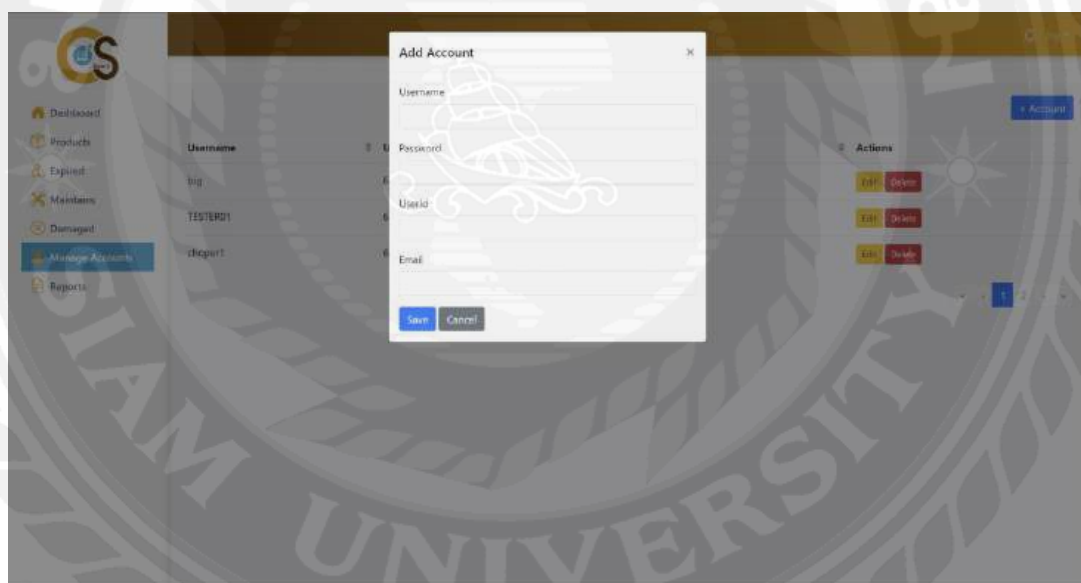
รูปที่ 4.14 หน้าแก้ไขข้อมูลทรัพย์สิน (Edit Product)

จากรูปที่ 4.14 หน้าแก้ไขข้อมูลทรัพย์สิน (Edit Product) แสดงหน้าสำหรับแก้ไขข้อมูลทรัพย์สิน โดยผู้ใช้ต้องกรอกข้อมูลเพื่อแก้ไขและบันทึกในระบบ



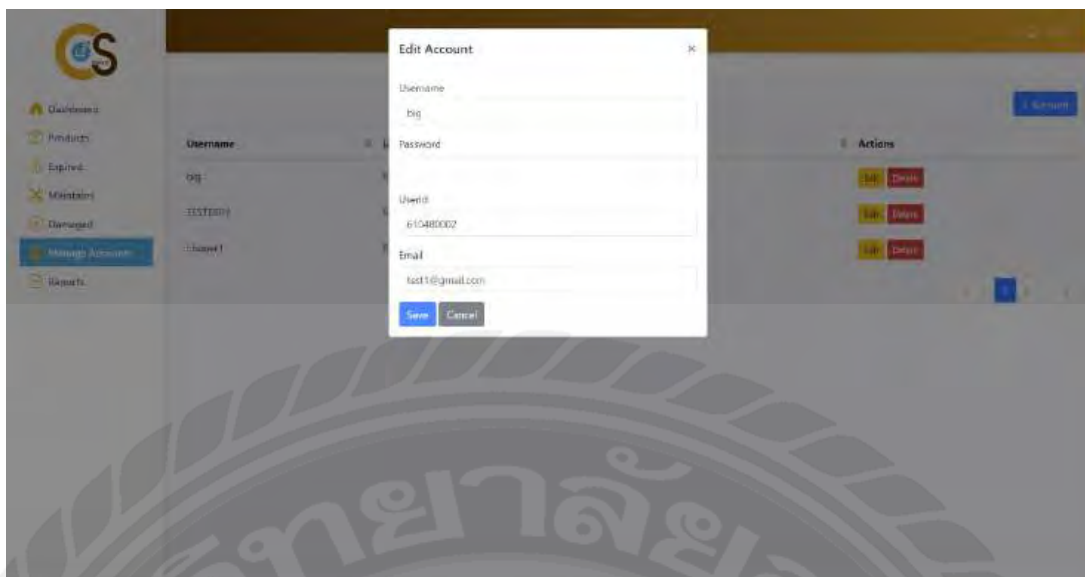
รูปที่ 4.15 หน้าแสดงข้อมูลผู้ใช้ (Manage Accounts)

จากรูปที่ 4.15 หน้าแสดงข้อมูลผู้ใช้ (Manage Accounts) แสดงข้อมูลต่างๆ ได้แก่ Username (username), UserID (userid), Email (email)



รูปที่ 4.16 หน้าเพิ่มข้อมูลผู้ใช้ (Add Account)

จากรูปที่ 4.16 หน้าเพิ่มข้อมูลผู้ใช้ (Add Account) แสดงหน้าสำหรับเพิ่มข้อมูลผู้ใช้ โดยผู้ดูแลระบบต้องกรอกข้อมูลและบันทึกในระบบ



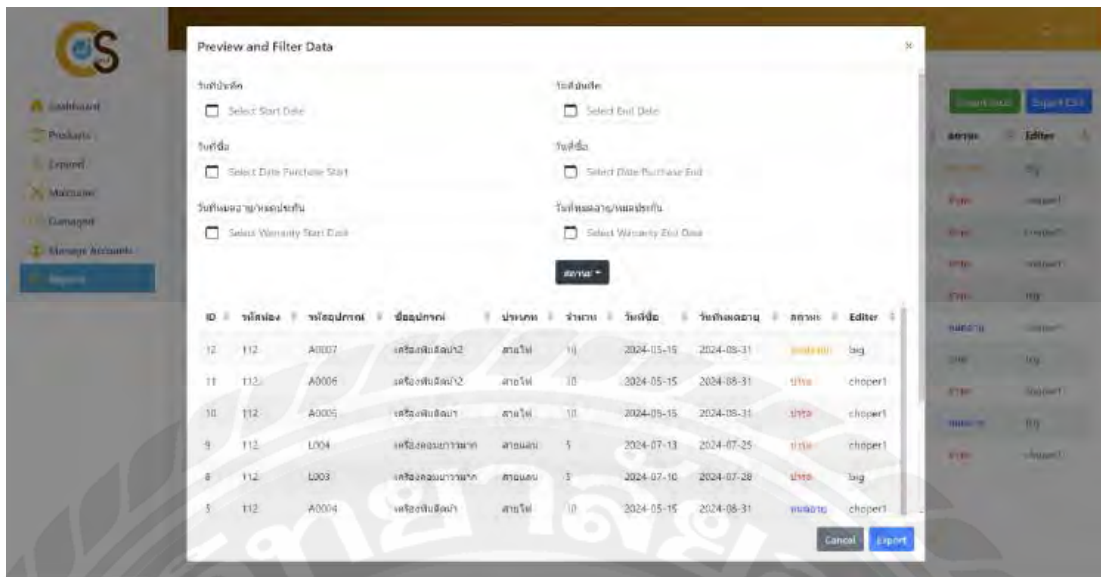
รูปที่ 4.17 หน้าแก้ไขข้อมูลผู้ใช้ (Edit Account)

จากรูปที่ 4.17 หน้าแก้ไขข้อมูลผู้ใช้ (Edit Account) แสดงสำหรับแก้ไขข้อมูลผู้ใช้ โดยผู้ดูแลระบบต้องกรอกข้อมูลและบันทึกในระบบ

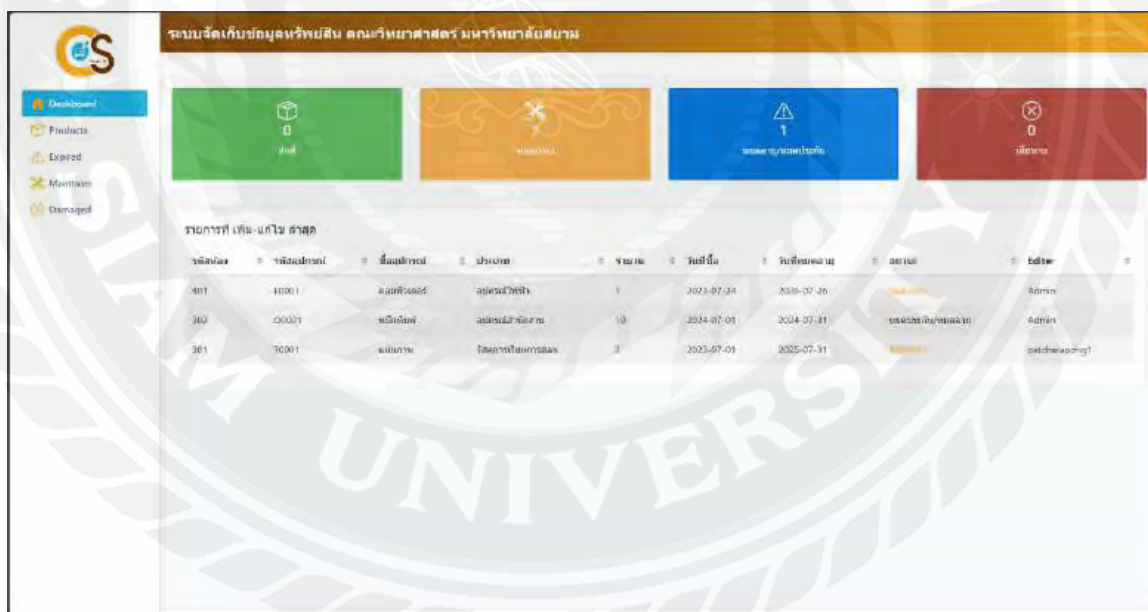
ID	รหัสห้อง	รหัสอุปกรณ์	ชื่ออุปกรณ์	ประเภท	จำนวน	วันที่ซื้อ	วันหมดอายุ	สถานะ	Editor
12	112	A0007	เครื่องพิมพ์ดีด	สายไฟ	12	2024-05-15	2024-08-31	พร้อมใช้งาน	big
11	112	A0006	เครื่องพิมพ์ดีด	สายไฟ	10	2024-05-15	2024-08-31	ชำรุด	choper1
10	112	A0005	เครื่องพิมพ์ดีด	สายไฟ	10	2024-05-15	2024-08-31	ชำรุด	choper1
9	112	1004	เครื่องคอมพิวเตอร์	สายเคเบิล	5	2024-07-13	2024-07-25	ชำรุด	choper1
8	112	1003	เครื่องคอมพิวเตอร์	สายเคเบิล	5	2024-07-10	2024-07-28	ชำรุด	big
6	112	A0004	เครื่องพิมพ์ดีด	สายไฟ	10	2024-05-15	2024-08-31	พร้อมใช้งาน	choper1
4	112	A0003	สายไฟ	สายไฟ	3	2024-07-04	2024-07-31	ปกติ	big
3	112	A0002	เครื่องพิมพ์ดีด	สายไฟ	10	2024-05-15	2024-08-31	ชำรุด	choper1
2	112	1001	สายเคเบิล	สายเคเบิล	1	2024-07-03	2024-07-31	พร้อมใช้งาน	big
1	112	A0001	เครื่องพิมพ์ดีด	สายไฟ	1	2024-05-15	2024-08-31	ชำรุด	choper1

รูปที่ 4.18 หน้าออกรายงานทรัพย์สิน (Reports)

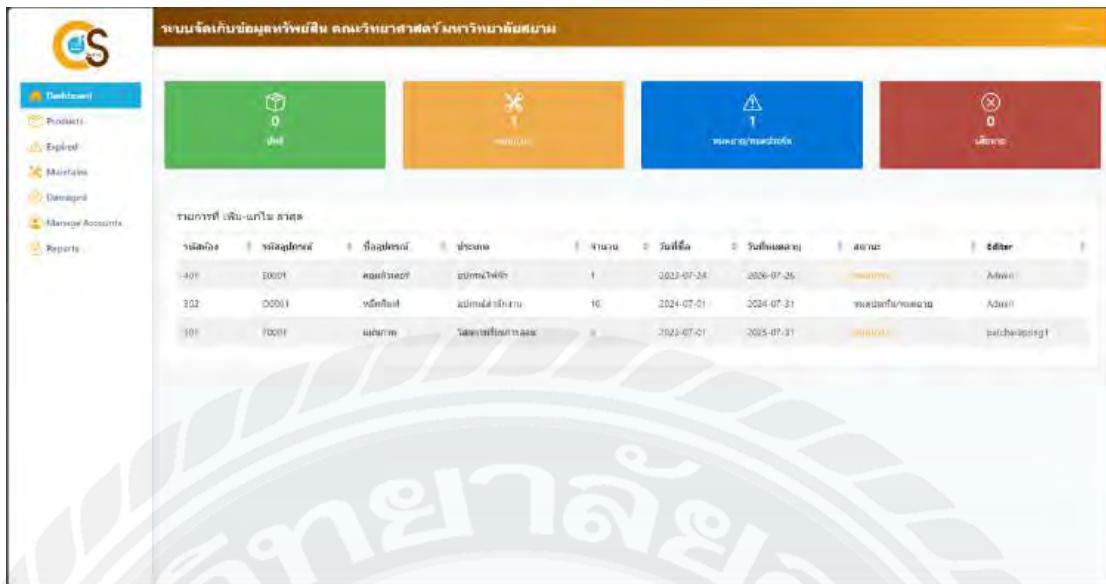
จากรูปที่ 4.18 หน้าออกรายงานทรัพย์สิน (Reports) จะแสดงข้อมูลรายการทรัพย์สินที่มีอยู่ในระบบฐานข้อมูล แสดงข้อมูลต่างๆ ได้แก่ รหัสห้อง(roomname) รหัสอุปกรณ์ (product_id) ชื่ออุปกรณ์ (product_name) ประเภท (category_id) จำนวน (amount) วันที่ซื้อ (datepur) วันหมดอายุ (warranty_expired) สถานะ (status)



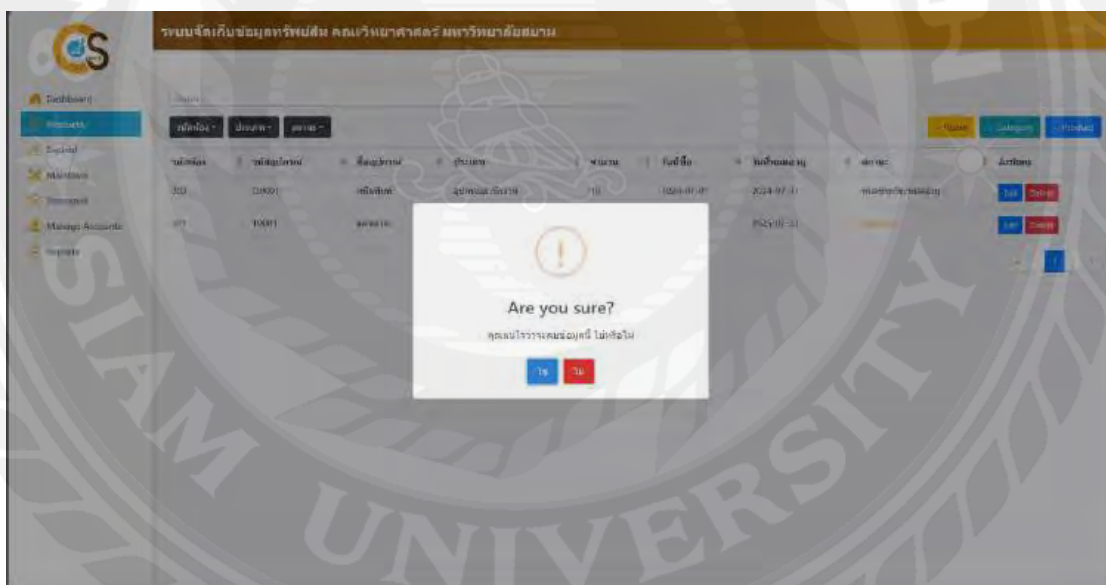
รูปที่ 4.19 หน้าแสดงตัวอย่างก่อนออกรายงานรูปแบบไฟล์ Excel (Export Excel)
 จากรูปที่ 4.19 หน้าแสดงตัวอย่างก่อนออกรายงานรูปแบบไฟล์ Excel จะแสดงข้อมูล
 สำหรับเลือกกรองข้อมูลทรัพย์สินที่ต้องการ โดยผู้ใช้ระบบต้องเลือกกรองข้อมูลและแปลงเป็น
 เอกสารที่ต้องการจากระบบ



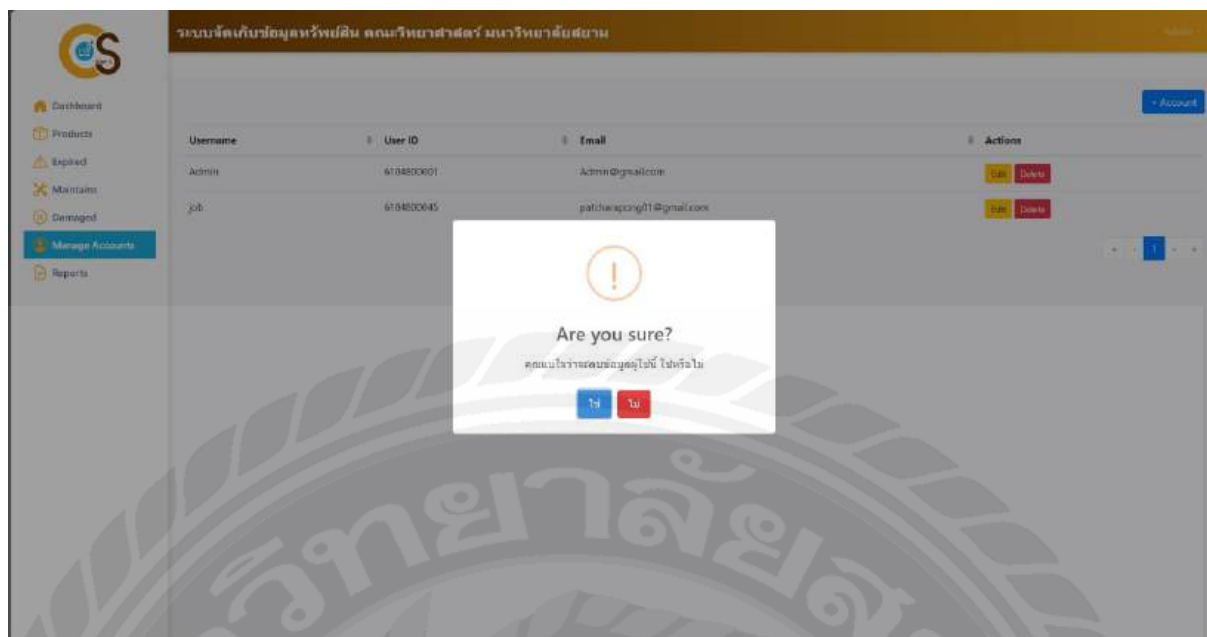
รูปที่ 4.20 หน้าเว็บเมื่อเข้าสู่ระบบด้วยบัญชีที่เป็น User
 จากรูปที่ 4.20 รายชื่อเมนู Manage Accounts และ Reports จะถูกซ่อนไว้จากการมองเห็นของ User



รูปที่ 4.21 หน้าเว็บเมื่อเข้าสู่ระบบด้วยบัญชีที่เป็น Administrator
จากรูปที่ 4.21 รายชื่อเมนู Manage Accounts และ Reports จะแสดงออกมามองเห็นเฉพาะบัญชีที่เป็น Administrator



รูปที่ 4.22 หน้าต่างป๊อปอัพเพื่อยืนยันค่าขอลบข้อมูลในหน้า Products
จากรูปที่ 4.22 หน้าต่างป๊อปอัพเพื่อยืนยันค่าขอลบข้อมูลรายการทรัพย์สิน จะแสดงหน้าต่างยืนยันเมื่อผู้ใช้กดปุ่มลบข้อมูลรายการทรัพย์สิน โดยผู้ใช้ต้องเลือก “ใช่” เพื่อบันทึกข้อมูลรายการทรัพย์สิน หรือ “ไม่” เพื่อยกเลิกการลบข้อมูลรายการทรัพย์สิน



รูปที่ 4.23 หน้าต่างป๊อปอัพเพื่อยืนยันค่าลบข้อมูลในหน้า Manage Account
จากรูปที่ 4.23 หน้าต่างป๊อปอัพเพื่อยืนยันค่าลบข้อมูลรายการชื่อผู้ใช้ จะแสดงหน้าต่างยืนยันเมื่อผู้ดูแลคลิกปุ่มลบข้อมูลรายการชื่อผู้ใช้ โดยผู้ดูแลต้องเลือก “ใช่” เพื่อลบข้อมูลรายการผู้ใช้ หรือ “ไม่” เพื่อยกเลิกการลบข้อมูลรายการผู้ใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลปริญญานิพนธ์

พัฒนาระบบจัดเก็บข้อมูลทรัพย์สิน กรณีศึกษา ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสยาม ได้พัฒนาเสร็จสิ้นตามวัตถุประสงค์ที่ตั้งไว้ โดยสามารถบันทึกข้อมูลผ่านหน้าเว็บแอปพลิเคชัน และจัดเก็บข้อมูลลงฐานข้อมูล โดยผู้ใช้งาน สามารถเพิ่ม ลบ แก้ไข ค้นหา ข้อมูลทรัพย์สิน ตรวจสอบ แก้ไขสถานะของทรัพย์สิน ในส่วนของผู้ดูแลระบบ จะสามารถเพิ่มผู้ใช้งาน แก้ไข ลบ ข้อมูลผู้ใช้ได้ และสามารถออกรายงานรายการทรัพย์สินในรูปแบบ Excel จากการพัฒนาระบบทำให้ผู้ใช้งานสามารถดำเนินการได้อย่างสะดวกสบาย แม่นยำ รวดเร็ว ช่วยให้การดำเนินงานมีประสิทธิภาพในการจัดการทรัพย์สินภายในภาควิชาวิทยาการคอมพิวเตอร์มากยิ่งขึ้น และสามารถนำไฟล์รายงานรูปแบบ Excel ส่งต่อให้กับฝ่ายจัดซื้อของมหาวิทยาลัยในการตรวจทรัพย์สินประจำปีได้อีกด้วย

5.2 ข้อดีของระบบ

- 5.2.1 ช่วยในการค้นหาทรัพย์สินได้รวดเร็วยิ่งขึ้น
- 5.2.2 ช่วยลดขั้นตอนและความผิดพลาดในการจัดการข้อมูลทรัพย์สิน
- 5.2.3 มีระบบฐานข้อมูลเพื่อเก็บข้อมูลต่างๆ ให้มีความเป็นระเบียบ
- 5.2.4 ช่วยในเรื่องการจัดการข้อมูลทรัพย์สินต่างๆ เช่น การแก้ไข หรือ การลบข้อมูล
- 5.2.5 สามารถกำหนดช่วงเวลาในการเรียกดูรายงานที่ต้องการได้
- 5.2.6 ช่วยติดตามสถานะของทรัพย์สินภายในภาควิชาได้
- 5.2.7 ช่วยเพิ่มความสะดวกสบายและลดเวลาในการบริหารจัดการทรัพย์สิน
- 5.2.8 ช่วยให้ทราบระยะเวลาในการใช้งานของทรัพย์สินแต่ละชิ้น
- 5.2.9 นำสารสนเทศที่ได้ไปใช้ประโยชน์ด้านการวางแผนการบริหารจัดการอุปกรณ์และสิ่งสนับสนุนการเรียนได้ต่อไป
- 5.2.10 สามารถนำไฟล์รายงานรูปแบบ Excel ส่งต่อให้กับฝ่ายจัดซื้อของมหาวิทยาลัยในการตรวจทรัพย์สินประจำปีได้

5.3 ข้อเสนอแนะ

เพื่อให้ระบบจัดเก็บข้อมูลทรัพย์สิน กรณีศึกษา ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยสยามมีประสิทธิภาพมากยิ่งขึ้น ควรพัฒนาฟังก์ชันเพิ่มเติม ดังนี้

- 5.3.1 ปรับปรุงระบบเพิ่มเติมเพื่อให้มีความยืดหยุ่นมากยิ่งขึ้นในการใช้งาน
- 5.3.2 ปรับปรุงระบบการแจ้งเตือน และการติดตามสถานะการซ่อม
- 5.3.3 เพิ่มฟังก์ชันเก็บรูปภาพอุปกรณ์ทรัพย์สิน ใบเสร็จในการซื้ออุปกรณ์ หรือใบรับประกัน
- 5.3.4 ปรับปรุงให้สามารถพิมพ์สติ๊กเกอร์ห้สอุปกรณ์มาติดที่ทรัพย์สินแต่ละชิ้น



บรรณานุกรม

คลาสเมธอด. (2567, 10 มกราคม). *Figma*. Developers IO.

<https://dev.classmethod.jp/articles/whats-about-figma/>

ไคลเอ็นท์ เซิร์ฟเวอร์. (2560, 16 พฤษภาคม). *สถาปัตยกรรมไคลเอ็นท์/เซิร์ฟเวอร์ (Client/ Server Architecture)*. Mindphp. <https://www.mindphp.com/คู่มือ/73-คืออะไร/2052-client-server-คืออะไร.html>

จักกะปรีด ธรรมบุรวง. (2560,10 เมษายน). *โปรแกรม HeidiSQL*. GlurGeek.

<https://www.glurgeek.com/education/การจัดการฐานข้อมูลโดย/>

นัก. (2567, 2 กุมภาพันธ์). *Nuxt.js*. Nuxt. <https://nuxt.com/docs/getting-started/introduction>

ปวี. (2566, 20 ธันวาคม). *Bootstrap Vue*. Bootstrap-Vue. <https://bootstrap-vue.org/docs>

ไมโครซอฟท์. (2560, 27 ตุลาคม). *Visual Studio Code*. Mindphp. <https://www.mindphp.com/บทความ/microsoft/4829-visual-studio-code.html>

รชฎา ทองภักดี. (2561, 1 มีนาคม). *Docker Desktop*. Medium.

<https://medium.com/@rachatatongpagdee/docker-คืออะไร-ใช้งานอย่างไร-7e77145967b6>

อเมซอน. (2567, 15 มกราคม). *เว็บแอปพลิเคชัน (Web Application)*. Amazon.

<https://aws.amazon.com/th/what-is/web-application/>

แอปมาสเตอร์. (2566, 9 มีนาคม). *ระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database)*.

[เว็บไซต์]. <https://appmaster.io/th/blog/taankh-muulechingsamphanthkhuue-aair>

แอปมาสเตอร์. (2565, 29 ตุลาคม). *ระบบฐานข้อมูล PostgreSQL*. [เว็บไซต์].

<https://appmaster.io/th/blog/postgresql-khuue-aair>