

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการจัดทำปริญญานิพนธ์การพัฒนาระบบแจ้งซ่อมออนไลน์สำหรับแผนกสารสนเทศทาง
การแพทย์ โรงพยาบาลหัวหิน เพื่อให้บรรลุตามวัตถุประสงค์จำเป็นต้องอาศัยหลักการ แนวคิด ทฤษฎี
และงานวิจัย เอกสาร ตำรา ที่เกี่ยวข้อง ดังต่อไปนี้

- 2.1**หลักการออกแบบและพัฒนาระบบ
- 2.2**Bootstrap Front-End Framework
- 2.3**การออกแบบเว็บแอปพลิเคชัน
- 2.4**ฐานข้อมูลและการออกแบบฐานข้อมูล
- 2.5**การวัดประสิทธิภาพและการวัดความพึงพอใจ
- 2.6**งานวิจัยที่เกี่ยวข้องกับการพัฒนาระบบแจ้งซ่อมออนไลน์

2.1**หลักการออกแบบและพัฒนาระบบ (SDLC)

ในการเขียนโปรแกรมเพื่อแก้ไขปัญหาต่างๆ จำเป็นต้องมีการวางแผนและออกแบบโปรแกรมไว้
ล่วงหน้า โดยกำหนดขั้นตอนวิธีการทำงานให้ชัดเจน ซึ่งกระบวนการวิเคราะห์และออกแบบโปรแกรม
เรียกว่า วัฏจักรการพัฒนาระบบงาน (System Development Life Cycle : SDLC) ซึ่งมีกระบวนการ
ทำงานเริ่มต้นจากการวิเคราะห์ปัญหาไปจนถึงการนำโปรแกรมไปใช้งาน และปรับปรุงพัฒนาระบบให้ดีขึ้น
โดยมีขั้นตอนดังนี้

2.1.1**การวิเคราะห์ปัญหา (Problem Analysis) ขั้นตอนนี้เป็นขั้นตอนแรกและมีความสำคัญ
อย่างยิ่งในการพัฒนาระบบ เนื่องจากการทำความเข้าใจถึงปัญหาและความต้องการของผู้ใช้งานอย่าง
แท้จริงกำหนดขอบเขตของปัญหา ทำความเข้าใจขอบเขตและลักษณะของปัญหา เพื่อกำหนดแนว
ทางการแก้ไขที่เหมาะสมกำหนดจุดประสงค์การทำงาน ระบุวัตถุประสงค์ของโปรแกรมที่ต้องการพัฒนา
เพื่อให้ทราบว่าโปรแกรมจะแก้ไขปัญหาอะไรและมีเป้าหมายอย่างไรเช่น โปรแกรมจะช่วยในการจัดการ
งานซ่อมบำรุงอย่างไร จะช่วยลดเวลาในการแจ้งซ่อมหรือไม่ จะช่วยให้การติดตามสถานะการซ่อมสะดวก
ขึ้นหรือไม่กำหนดเป้าหมายให้ชัดเจน กำหนดผลลัพธ์ที่ต้องการจากโปรแกรมอย่างชัดเจน เพื่อเป็น
แนวทางในการออกแบบและพัฒนาเช่น โปรแกรมต้องสามารถรองรับผู้ใช้งานได้กี่คน ต้องสามารถจัดการ
ข้อมูลการซ่อมได้มากน้อยแค่ไหน ต้องมีระบบแจ้งเตือนหรือไม่ ศึกษาวิธีการแก้ปัญหา ค้นคว้าและเลือก
วิธีการแก้ปัญหาที่เหมาะสมที่สุด โดยพิจารณาจากความเป็นไปได้และทรัพยากรที่มีอยู่

เช่น จะใช้เทคโนโลยีใดในการพัฒนาโปรแกรม จะใช้ฐานข้อมูลแบบใด จะมีการเชื่อมต่อกับระบบอื่นๆ หรือไม่กำหนดข้อจำกัดและศึกษาความเป็นไปได้ ระบุข้อจำกัดต่างๆที่อาจมีผลต่อการพัฒนาโปรแกรม เช่น ข้อจำกัดด้านเทคนิคงบประมาณ และเวลากำหนดข้อมูลนำเข้ากำหนดลักษณะการรับข้อมูล กำหนดวิธีการรับข้อมูลเข้าสู่โปรแกรม เช่น รับจากผู้ใช้ผ่านทางแป้นพิมพ์หรือรับจากไฟล์ข้อมูลเช่น ผู้ใช้งานจะต้องกรอกข้อมูลอะไรบ้างในแบบฟอร์มแจ้งซ่อม จะมีการอัปโหลดรูปภาพหรือไฟล์ประกอบหรือไม่ รูปแบบข้อมูลที่รับเข้ามา กำหนดรูปแบบของข้อมูลที่จะนำเข้า เช่น ตัวเลข ข้อความ หรือวันที่เช่น เลขที่ใบแจ้งซ่อมจะเป็นตัวเลขหรือไม่ ชื่อผู้แจ้งซ่อมจะเป็นตัวอักษรหรือไม่ ขอบเขตของข้อมูล กำหนดช่วงของค่าที่ข้อมูลสามารถมีได้ เช่น อายุต้องอยู่ระหว่าง 18 ถึง 60 ปีข้อจำกัดในการรับข้อมูล กำหนดเงื่อนไขหรือข้อจำกัดในการรับข้อมูล เช่น ต้องเป็นตัวเลขเท่านั้น หรือต้องไม่เป็นค่าว่างวิธีการประมวลผลกำหนดวิธีการหรือเทคนิคที่ใช้แก้ปัญหา เลือกวิธีการหรือเทคนิคที่เหมาะสมในการแก้ไขปัญหาที่ได้กำหนดไว้เช่น จะใช้อัลกอริทึมใดในการจัดลำดับความสำคัญของงานซ่อม จะใช้วิธีการใดในการแจ้งเตือนผู้ใช้งานกำหนดขั้นตอนทำงานให้ชัดเจน ออกแบบขั้นตอนการทำงานของโปรแกรมอย่างละเอียด เพื่อให้โปรแกรมสามารถแก้ไขปัญหาได้ตามลำดับที่ถูกต้องเช่น เมื่อผู้ใช้งานแจ้งซ่อมแล้ว ข้อมูลจะถูกส่งไปยังใคร เจ้าหน้าที่จะดำเนินการอย่างไรต่อ

2.1.2**การออกแบบโปรแกรม (Program Design)

ขั้นตอนการออกแบบโปรแกรมเป็นขั้นตอนสำคัญในการพัฒนาระบบแจ้งซ่อมออนไลน์ โดยมีวัตถุประสงค์เพื่อกำหนดรายละเอียดและโครงสร้างของระบบก่อนที่จะลงมือพัฒนาโปรแกรมจริง ซึ่งจะช่วยให้การพัฒนาเป็นไปอย่างมีประสิทธิภาพและตรงตามความต้องการของผู้ใช้งาน ออกแบบขั้นตอนวิธี (Algorithm) ขั้นตอนวิธี (Algorithm) คือ กระบวนการแก้ปัญหาที่เป็นลำดับขั้นตอน ในการออกแบบขั้นตอนวิธีสำหรับระบบแจ้งซ่อม จะต้องพิจารณาถึงขั้นตอนการทำงานหลักๆ ของระบบ เช่น ขั้นตอนการแจ้งซ่อม: ผู้ใช้งานกรอกข้อมูลรายละเอียดปัญหา เลือกอุปกรณ์ที่ต้องการซ่อม และส่งคำขอ ขั้นตอนการรับเรื่อง: ระบบตรวจสอบความถูกต้องของข้อมูล และส่งต่อคำขอไปยังเจ้าหน้าที่ที่รับผิดชอบ ขั้นตอนการดำเนินการซ่อม เจ้าหน้าที่ตรวจสอบปัญหา ประเมินระยะเวลา และดำเนินการซ่อม ขั้นตอนการปิดงาน: เจ้าหน้าที่บันทึกผลการซ่อมและแจ้งผู้ใช้งานเครื่องมือที่ใช้ในการออกแบบขั้นตอนวิธีได้แก่ ผังงาน (Flowchart) และรหัสจำลอง (Pseudocode) ซึ่งจะช่วยให้เห็นภาพรวมของขั้นตอนการทำงานและตรรกะของโปรแกรมได้ชัดเจนยิ่งขึ้น ออกแบบส่วนต่อประสานผู้ใช้ (User Interface Design) ส่วนต่อประสานผู้ใช้ (User Interface) คือ ส่วนที่ผู้ใช้งานจะ interacting กับระบบโดยตรง การออกแบบส่วนต่อประสานผู้ใช้ที่ดีจะช่วยให้ผู้ใช้งานสามารถใช้งานระบบได้ง่าย สะดวก และมีประสิทธิภาพในการออกแบบส่วนต่อประสานผู้ใช้สำหรับระบบแจ้งซ่อม ควรคำนึงถึงสิ่งต่อไปนี้ ความง่ายในการใช้งาน (Usability): ออกแบบให้เมนูและฟังก์ชันต่างๆ เข้าใจง่าย ไม่ซับซ้อน และใช้งานสะดวกความสวยงาม (Aesthetics): ออกแบบให้มีรูปแบบที่สวยงาม น่าใช้งาน และสอดคล้องกับภาพลักษณ์ขององค์กร การตอบสนองต่ออุปกรณ์ที่หลากหลาย (Responsiveness) ออกแบบให้สามารถใช้งานได้บนอุปกรณ์หลากหลายประเภท เช่น คอมพิวเตอร์ แท็บเล็ต และสมาร์ตโฟนหน้าจอและส่วนประกอบต่างๆ ที่ควรมีในระบบแจ้งซ่อม

หน้าจอหลัก (Dashboard) แสดงภาพรวมของสถานการณ์แจ้งซ่อมต่างๆ หน้าจอแจ้งซ่อม (Submit Request) ให้ผู้ใช้งานกรอกรายละเอียดปัญหาและส่งคำขอซ่อม หน้าจอติดตามสถานะ (Track Status) ให้ผู้ใช้งานตรวจสอบสถานการณ์ซ่อมของคำขอที่ตนเองแจ้ง หน้าจอประวัติการซ่อม (History) แสดงประวัติการแจ้งซ่อมและการแก้ไขปัญหาที่ผ่านมา หน้าจอรายงาน (Reports) แสดงรายงานสรุปข้อมูลการซ่อมต่างๆ ออกแบบฐานข้อมูล (Database Design) ฐานข้อมูล (Database) คือ ส่วนที่ใช้จัดเก็บข้อมูลของระบบ การออกแบบฐานข้อมูลที่ดีจะช่วยให้ระบบสามารถจัดเก็บและเรียกใช้ข้อมูลได้อย่างมีประสิทธิภาพ และถูกต้อง ในการออกแบบฐานข้อมูลสำหรับระบบแจ้งซ่อม ควรคำนึงถึงสิ่งต่อไปนี้ข้อมูลที่ต้องจัดเก็บ: กำหนดข้อมูลที่จะจำเป็นต้องจัดเก็บในระบบ เช่น ข้อมูลผู้ใช้งาน ข้อมูลอุปกรณ์ ข้อมูลการแจ้งซ่อม และข้อมูลการดำเนินการซ่อม ความสัมพันธ์ระหว่างข้อมูล กำหนดความสัมพันธ์ระหว่างข้อมูลต่างๆ เพื่อให้สามารถเชื่อมโยงข้อมูลได้อย่างถูกต้อง ประสิทธิภาพ ออกแบบโครงสร้างฐานข้อมูลให้สามารถจัดเก็บและเรียกใช้ข้อมูลได้อย่างรวดเร็ว ความปลอดภัย ออกแบบระบบรักษาความปลอดภัยของฐานข้อมูล เพื่อป้องกันการสูญหายหรือการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต ตารางที่ควรมีในฐานข้อมูลระบบแจ้งซ่อม ได้แก่ ตารางผู้ใช้งาน (Users) เก็บข้อมูลผู้ใช้งาน เช่น ชื่อผู้ใช้ รหัสผ่าน อีเมล และเบอร์โทรศัพท์ ตารางอุปกรณ์ (Devices) เก็บข้อมูลอุปกรณ์ เช่น ชื่ออุปกรณ์ ประเภท รุ่น และสถานที่ติดตั้ง ตารางการแจ้งซ่อม (Requests) เก็บข้อมูลการแจ้งซ่อม เช่น ผู้แจ้ง วันที่แจ้ง รายละเอียดปัญหา และสถานการณ์ซ่อม ตารางการดำเนินการซ่อม (Actions) เก็บข้อมูลการดำเนินการซ่อม เช่น เจ้าหน้าที่ที่รับผิดชอบ วันที่ดำเนินการ รายละเอียดการแก้ไข และผลการซ่อม การออกแบบโปรแกรมที่ดีจะช่วยให้การพัฒนาระบบแจ้งซ่อมออนไลน์เป็นไปอย่างมีประสิทธิภาพและได้ระบบที่มีคุณภาพ ตรงตามความต้องการของผู้ใช้งานและใช้งานได้จริง

2.1.3**การเขียนโปรแกรม (Program Coding) ในขั้นตอนนี้ จะเป็นการลงมือเขียนโปรแกรมตามที่ได้ออกแบบไว้ โดยใช้ภาษาโปรแกรมที่เหมาะสมกับระบบแจ้งซ่อมออนไลน์ ซึ่งในที่นี้จะใช้ PHP, HTML, CSS, JavaScript, และ SQL ร่วมกัน PHP (Hypertext Preprocessor) ภาษาสคริปต์ฝั่งเซิร์ฟเวอร์ (Server-side scripting language) ที่ใช้ในการจัดการตรรกะของระบบ เช่น การรับข้อมูลจากผู้ใช้ การประมวลผลข้อมูล การติดต่อกับฐานข้อมูล และการสร้างเนื้อหาเว็บเพจแบบไดนามิก HTML (Hypertext Markup Language) ภาษาหลักที่ใช้ในการสร้างโครงสร้างของเว็บเพจ กำหนดองค์ประกอบต่างๆ ของหน้าเว็บ เช่น หัวข้อ ย่อหน้า รูปภาพ ตาราง และแบบฟอร์ม CSS (Cascading Style Sheets) ภาษาที่ใช้ในการตกแต่งรูปแบบของเว็บเพจ เช่น สี ขนาดตัวอักษร รูปแบบตัวอักษร และการจัดวางองค์ประกอบต่างๆ JavaScript ภาษาสคริปต์ฝั่งไคลเอนต์ (Client-side scripting language) ที่ใช้ในการเพิ่มความสามารถในการโต้ตอบกับผู้ใช้ เช่น การตรวจสอบความถูกต้องของข้อมูล การแสดงผลแบบไดนามิก และการสร้างเอฟเฟกต์ต่างๆ SQL (Structured Query Language): ภาษาที่ใช้ในการจัดการฐานข้อมูล เช่น การสร้างตาราง การเพิ่ม ลบ แก้ไขข้อมูล และการค้นหาข้อมูล

2.1.3.1**ขั้นตอนการเขียนโปรแกรม

1)**สร้างโครงสร้างไฟล์: สร้างไฟล์ต่างๆ ที่จำเป็นสำหรับระบบ เช่นไฟล์ PHP สำหรับจัดการตรรกะของระบบ (เช่น index.php, login.php, submit_request.php)ไฟล์ HTML สำหรับโครงสร้างของหน้าเว็บ (เช่น index.html, login.html, submit_request.html)ไฟล์ CSS สำหรับตกแต่งรูปแบบเว็บเพจ (เช่น style.css)ไฟล์ JavaScript สำหรับเพิ่มความสามารถในการโต้ตอบ (เช่น script.js)

2)**เขียนโค้ด PHPเชื่อมต่อกับฐานข้อมูล MySQLเขียนฟังก์ชันสำหรับรับข้อมูลจากผู้ใช้ (เช่น จากแบบฟอร์มแจ้งซ่อม)เขียนฟังก์ชันสำหรับตรวจสอบความถูกต้องของข้อมูลเขียนฟังก์ชันสำหรับบันทึกข้อมูลลงในฐานข้อมูลเขียนฟังก์ชันสำหรับดึงข้อมูลจากฐานข้อมูลมาแสดงผล (เช่น สถานะการซ่อม, ประวัติการซ่อม) เขียนฟังก์ชันสำหรับสร้างรายงาน เขียนฟังก์ชันสำหรับการจัดการผู้ใช้ (เช่น การลงทะเบียน, การเข้าสู่ระบบ, การแก้ไขข้อมูลส่วนตัว)

3)**เขียนโค้ด HTMLสร้างโครงสร้างของหน้าเว็บต่างๆ สร้างแบบฟอร์มสำหรับรับข้อมูลจากผู้ใช้ สร้างตารางหรือรูปแบบอื่นๆ สำหรับแสดงผลข้อมูล

4)**เขียนโค้ด CSS ตกแต่งรูปแบบของหน้าเว็บให้สวยงามและนำใช้งาน ปรับแต่งรูปแบบให้เหมาะสมกับอุปกรณ์ที่หลากหลาย (Responsive Design)

5)**เขียนโค้ด JavaScript:เพิ่มความสามารถในการโต้ตอบให้กับหน้าเว็บ ตรวจสอบความถูกต้องของข้อมูลในแบบฟอร์ม แสดงผลแบบไดนามิก (เช่น อัปเดตสถานะการซ่อมโดยไม่ต้องรีเฟรชหน้าเว็บ)

6)**เขียนคำสั่ง SQLสร้างตารางในฐานข้อมูลเขียนคำสั่งสำหรับเพิ่ม ลบ แก้ไข และค้นหาข้อมูลในฐานข้อมูล

7)**การทดสอบโปรแกรมเบื้องต้นเมื่อเขียนโปรแกรมเสร็จแต่ละส่วน ควรทดสอบการทำงานเบื้องต้นเพื่อตรวจสอบข้อผิดพลาดและแก้ไขให้ถูกต้องก่อนที่จะไปยังขั้นตอนถัดไป การทดสอบสามารถทำได้โดย:ทดสอบการทำงานของแต่ละฟังก์ชัน ตรวจสอบว่าฟังก์ชันต่างๆ ทำงานได้ถูกต้องตามที่ออกแบบไว้หรือไม่ ทดสอบการรับส่งข้อมูล ตรวจสอบว่าข้อมูลที่รับจากผู้ใช้ถูกบันทึกและแสดงผลอย่างถูกต้องหรือไม่ ทดสอบการใช้งานบนอุปกรณ์ต่างๆ ตรวจสอบว่าเว็บแอปพลิเคชันสามารถใช้งานได้บนอุปกรณ์ต่างๆ เช่น คอมพิวเตอร์ แท็บเล็ต และสมาร์ตโฟนการทดสอบโปรแกรมเบื้องต้นจะช่วยให้สามารถแก้ไขข้อผิดพลาดได้ตั้งแต่เนิ่นๆ ซึ่งจะช่วยลดเวลาและค่าใช้จ่ายในการแก้ไขปัญหาในภายหลัง

2.1.4**การทดสอบและแก้ไขโปรแกรม (Program Testing & Verification) การทดสอบและแก้ไขโปรแกรมเป็นขั้นตอนที่สำคัญอย่างยิ่งในการพัฒนาระบบแจ้งซ่อมออนไลน์ เพื่อให้มั่นใจว่าระบบสามารถทำงานได้อย่างถูกต้องตามที่ออกแบบไว้ และมีประสิทธิภาพในการใช้งานจริง โดยมีกระบวนการดังนี้

2.1.4.1)**การทดสอบโปรแกรมอย่างละเอียด (Comprehensive Testing)

การทดสอบหน่วย (Unit Testing) ทดสอบการทำงานของแต่ละส่วนย่อยของโปรแกรม (เช่น ฟังก์ชัน, โมดูล) เพื่อให้แน่ใจว่าแต่ละส่วนทำงานได้อย่างถูกต้อง การทดสอบการรวม (Integration Testing) ทดสอบการทำงานร่วมกันของส่วนต่างๆ ของโปรแกรม เพื่อให้แน่ใจว่าส่วนต่างๆ สามารถทำงานร่วมกันได้อย่างราบรื่น การทดสอบระบบ (System Testing) ทดสอบการทำงานของระบบทั้งหมด เพื่อให้แน่ใจว่าระบบสามารถทำงานได้ตามที่ออกแบบไว้และตรงตามความต้องการของผู้ใช้งาน การทดสอบการยอมรับของผู้ใช้ (User Acceptance Testing) ให้ผู้ใช้งานจริงทดสอบระบบ เพื่อให้มั่นใจว่าระบบตรงตามความต้องการและใช้งานได้จริง การทดสอบความเข้ากันได้ (Compatibility Testing) ทดสอบการทำงานของระบบบนเบราว์เซอร์และอุปกรณ์ต่างๆ เพื่อให้แน่ใจว่าระบบสามารถใช้งานได้บนแพลตฟอร์มที่หลากหลาย การทดสอบประสิทธิภาพ (Performance Testing) ทดสอบประสิทธิภาพของระบบภายใต้สภาวะโหลดต่างๆ เพื่อให้แน่ใจว่าระบบสามารถทำงานได้อย่างมีประสิทธิภาพเมื่อมีผู้ใช้งานจำนวนมาก การทดสอบความปลอดภัย (Security Testing) ทดสอบความปลอดภัยของระบบ เพื่อป้องกันการโจมตีจากผู้ไม่หวังดี และปกป้องข้อมูลของผู้ใช้งาน

2.1.4.2)**การแก้ไขข้อผิดพลาด (Bug Fixing) การระบุข้อผิดพลาด บันทึกลงและจัดลำดับความสำคัญของข้อผิดพลาดที่พบในระหว่างการทดสอบ การแก้ไขข้อผิดพลาด แก้ไขข้อผิดพลาดตามลำดับความสำคัญ โดยเริ่มจากข้อผิดพลาดที่ส่งผลกระทบต่อการทำงานของระบบมากที่สุด การทดสอบซ้ำ ทดสอบระบบซ้ำหลังจากแก้ไขข้อผิดพลาด เพื่อให้แน่ใจว่าการแก้ไขไม่ทำให้เกิดปัญหาใหม่

2.1.4.3)**การปรับปรุงและเพิ่มประสิทธิภาพ (Refinement and Optimization) การปรับปรุงประสิทธิภาพ ปรับปรุงประสิทธิภาพของระบบให้ดียิ่งขึ้น เช่น เพิ่มความเร็วในการประมวลผล ลดการใช้ทรัพยากร หรือปรับปรุงการแสดงผล

การปรับปรุงการใช้งาน ปรับปรุงส่วนต่อประสานผู้ใช้ (UI) ให้ใช้งานง่ายและสะดวกยิ่งขึ้น การเพิ่มฟังก์ชันใหม่ เพิ่มเติมฟังก์ชันการทำงานใหม่ๆ ตามความต้องการของผู้ใช้งาน หรือเพื่อเพิ่มประสิทธิภาพของระบบ

2.1.4.4)**การทดสอบอย่างต่อเนื่อง (Continuous Testing) การทดสอบการถดถอย (Regression Testing) ทดสอบระบบซ้ำหลังจากมีการเปลี่ยนแปลงโค้ด เพื่อให้แน่ใจว่าการเปลี่ยนแปลงไม่ทำให้เกิดปัญหาใหม่ การตรวจสอบความปลอดภัยอย่างต่อเนื่อง ตรวจสอบและปรับปรุงระบบรักษาความปลอดภัยอย่างสม่ำเสมอ เพื่อป้องกันภัยคุกคามใหม่ๆ ตัวอย่างการทดสอบระบบแจ้งซ่อมออนไลน์ การแจ้งซ่อม ทดสอบการกรอกแบบฟอร์มแจ้งซ่อม การอัปโหลดไฟล์แนบ และการส่งคำขอ การอัปเดตสถานะ ทดสอบการเปลี่ยนแปลงสถานะการซ่อมโดยเจ้าหน้าที่ และการแจ้งเตือนผู้ใช้งาน การออก

รายงาน ทดสอบการสร้างรายงานสรุปข้อมูลการซ่อมต่างๆ การใช้งานบนอุปกรณ์ต่างๆ ทดสอบการใช้งานระบบบนคอมพิวเตอร์ แท็บเล็ต และสมาร์ทโฟนเครื่องมือที่ใช้ในการทดสอบเครื่องมือทดสอบอัตโนมัติ (Automated Testing Tools) ช่วยในการทดสอบระบบซ้ำๆ โดยอัตโนมัติ เครื่องมือตรวจสอบประสิทธิภาพ (Performance Testing Tools): ช่วยในการทดสอบประสิทธิภาพของระบบภายใต้สภาวะโหลดต่างๆ เครื่องมือตรวจสอบความปลอดภัย (Security Testing Tools): ช่วยในการตรวจสอบช่องโหว่และความเสี่ยงด้านความปลอดภัยของระบบการทดสอบและแก้ไขโปรแกรมอย่างละเอียดและต่อเนื่อง จะช่วยให้ระบบแจ้งซ่อมออนไลน์มีความสมบูรณ์และมีประสิทธิภาพในการใช้งานจริง ซึ่งจะช่วยให้เพิ่มประสิทธิภาพในการทำงานของแผนกสารสนเทศทางการแพทย์โรงพยาบาลหัวหิน และสร้างความพึงพอใจให้กับผู้ใช้งาน

2.1.5**การจัดทำเอกสารและคู่มือการใช้งาน (Program Documentation)

การจัดทำเอกสารและคู่มือการใช้งานเป็นขั้นตอนสำคัญในวัฏจักรการพัฒนา (SDLC) ที่มักถูกมองข้าม แต่มีความสำคัญอย่างยิ่งต่อความสำเร็จของโครงการพัฒนาระบบ โดยเฉพาะอย่างยิ่งในระบบแจ้งซ่อมออนไลน์สำหรับแผนกสารสนเทศทางการแพทย์ โรงพยาบาลหัวหิน เอกสารและคู่มือจะช่วยให้ผู้ใช้งานทุกระดับสามารถเข้าใจและใช้งานระบบได้อย่างมีประสิทธิภาพ รวมถึงเป็นแหล่งอ้างอิงสำหรับการแก้ไขปัญหาและบำรุงรักษาระบบในอนาคต

2.1.5.1)**การจัดทำเอกสารและคู่มือการใช้งานเป็นขั้นตอนสำคัญในวัฏจักรการพัฒนา (SDLC) ที่มักถูกมองข้าม แต่มีความสำคัญอย่างยิ่งต่อความสำเร็จของโครงการพัฒนาระบบ โดยเฉพาะอย่างยิ่งในระบบแจ้งซ่อมออนไลน์สำหรับแผนกสารสนเทศทางการแพทย์ โรงพยาบาลหัวหิน เอกสารและคู่มือจะช่วยให้ผู้ใช้งานทุกระดับสามารถเข้าใจและใช้งานระบบได้อย่างมีประสิทธิภาพ รวมถึงเป็นแหล่งอ้างอิงสำหรับการแก้ไขปัญหาและบำรุงรักษาระบบในอนาคต

1)**จัดทำเอกสารประกอบโปรแกรม เอกสารประกอบโปรแกรมมีจุดประสงค์เพื่อให้รายละเอียดเชิงลึกเกี่ยวกับการออกแบบและการพัฒนาโปรแกรม ซึ่งจะเป็นประโยชน์ต่อนักพัฒนาและผู้ดูแลระบบในการทำความเข้าใจระบบ และสามารถแก้ไขปัญหาหรือปรับปรุงระบบได้ในอนาคต เอกสารประกอบโปรแกรมควรประกอบด้วย ข้อมูลการออกแบบ (Design Specification): อธิบายแนวคิดและเหตุผลในการออกแบบระบบ ฟังก์ชันการทำงานต่างๆ โครงสร้างของระบบ และส่วนประกอบต่างๆ ของระบบ โครงสร้างฐานข้อมูล (Database Schema): อธิบายรายละเอียดของตารางในฐานข้อมูล ความสัมพันธ์ระหว่างตาราง และข้อจำกัดต่างๆ ของฐานข้อมูล ขั้นตอนการทำงานของระบบ (Workflow Diagram): อธิบายขั้นตอนการทำงานของระบบตั้งแต่เริ่มต้นจนจบกระบวนการ รวมถึงการไหลของข้อมูล และการโต้ตอบระหว่างผู้ใช้งานกับระบบ รายละเอียดโค้ดโปรแกรม (Code Documentation) อธิบายโค้ดโปรแกรมแต่ละส่วนอย่างละเอียด เพื่อให้ นักพัฒนาสามารถทำความเข้าใจและแก้ไขโค้ดได้ การติดตั้งและตั้งค่าระบบ (Installation and Configuration Guide): อธิบายขั้นตอนการติดตั้งและตั้งค่าระบบสำหรับผู้ดูแลระบบ การแก้ไขปัญหาเบื้องต้น (Troubleshooting Guide): อธิบายปัญหาที่อาจเกิดขึ้นและวิธีการแก้ไขปัญหาเบื้องต้น

2)**จัดทำคู่มือการใช้งานคู่มือการใช้งานมีจุดประสงค์เพื่อให้ผู้ใช้งานสามารถเรียนรู้และใช้งานระบบได้อย่างถูกต้องและมีประสิทธิภาพ คู่มือการใช้งานควรประกอบด้วยบทนำ (Introduction): แนะนำระบบ วัตถุประสงค์ และประโยชน์ของระบบการเริ่มต้นใช้งาน (Getting Started): อธิบายขั้นตอนการเข้าสู่ระบบ การลงทะเบียน (ถ้ามี) และการตั้งค่าเบื้องต้น การใช้งานฟังก์ชันต่างๆ อธิบายวิธีการใช้งานฟังก์ชันต่างๆ ของระบบอย่างละเอียด เช่น การแจ้งซ่อม: วิธีการกรอกแบบฟอร์มแจ้งซ่อม การแนบไฟล์ และการส่งคำขอ การติดตามสถานะ: วิธีการตรวจสอบสถานะการซ่อม และการดูรายละเอียดการดำเนินการ การดูประวัติการซ่อม: วิธีการค้นหาและดูประวัติการซ่อมที่ผ่านมา คำถามที่พบบ่อย (FAQ): รวบรวมคำถามที่ผู้ใช้งานมักถามบ่อยและคำตอบ ช่องทางการติดต่อ (Contact Information): ให้ข้อมูลการติดต่อสำหรับขอความช่วยเหลือหรือรายงานปัญหา

2.1.6**การใช้งานจริง (Program Implement) ขั้นตอนนี้เป็นการนำระบบที่พัฒนาแล้วไปใช้งานจริง โดยมีรายละเอียดดังนี้

2.1.6.1**การติดตั้งและปรับแต่งระบบ (System Installation and Configuration)

1)**การติดตั้งบนเซิร์ฟเวอร์: ระบบแจ้งซ่อมออนไลน์จะถูกติดตั้งบนเว็บเซิร์ฟเวอร์ (Web Server) ที่มีการเชื่อมต่ออินเทอร์เน็ต เพื่อให้ผู้ใช้งานสามารถเข้าถึงระบบได้จากทุกที่ผ่านเว็บ

2)**การกำหนดค่า (Configuration): ผู้ดูแลระบบจะทำการตั้งค่าระบบเบื้องต้น เช่น การกำหนดชื่อผู้ใช้และรหัสผ่านสำหรับผู้ดูแลระบบและเจ้าหน้าที่ซ่อมบำรุง การตั้งค่าการเชื่อมต่อกับฐานข้อมูล และการกำหนดค่าการแจ้งเตือนต่างๆ

3)**การทดสอบระบบ (System Testing): ก่อนที่จะเปิดให้ใช้งานจริง จะมีการทดสอบระบบอย่างละเอียดอีกครั้ง เพื่อให้แน่ใจว่าระบบสามารถทำงานได้อย่างถูกต้องและมีประสิทธิภาพ

2.1.6.2**การฝึกอบรมผู้ใช้ (User Training)

1)**การจัดทำคู่มือการใช้งาน: จัดทำคู่มือการใช้งานที่เข้าใจง่าย ครอบคลุมทุกฟังก์ชันของระบบ และมีภาพประกอบที่ชัดเจน

2)**การจัดอบรม: จัดอบรมให้กับผู้ใช้งานทุกกลุ่ม (ผู้แจ้งซ่อม เจ้าหน้าที่ซ่อมบำรุง และผู้ดูแลระบบ) เพื่อให้สามารถใช้งานระบบได้อย่างถูกต้องและมีประสิทธิภาพ

3)**การสนับสนุนหลังการใช้งาน: ให้การสนับสนุนและช่วยเหลือผู้ใช้งานหลังจากเริ่มใช้งานระบบ เช่น การตอบคำถาม การแก้ไขปัญหา และการให้คำแนะนำ

2.1.6.3**การติดตามและประเมินผล (Monitoring and Evaluation)

1)**การติดตามการใช้งาน: ติดตามการใช้งานระบบอย่างสม่ำเสมอ เพื่อตรวจสอบประสิทธิภาพและความพึงพอใจของผู้ใช้

2)**การเก็บข้อมูล: เก็บข้อมูลการใช้งานระบบ เช่น จำนวนการแจ้งซ่อม เวลาในการแก้ไขปัญหา และความคิดเห็นของผู้ใช้

3)**การวิเคราะห์ข้อมูล: วิเคราะห์ข้อมูลที่เก็บได้ เพื่อระบุปัญหาและจุดที่ต้องปรับปรุง

4)**การปรับปรุงระบบ: ปรับปรุงระบบตามผลการประเมิน เพื่อให้ระบบมีประสิทธิภาพและตอบสนองความต้องการของผู้ใช้งานได้ดียิ่งขึ้น

2.1.7**การปรับปรุงและพัฒนาโปรแกรม (Program Maintenance) วิเคราะห์เสร็จแล้วการปรับปรุงและพัฒนาโปรแกรม (Program Maintenance)การบำรุงรักษาโปรแกรมเป็นขั้นตอนต่อเนื่องหลังจากที่ระบบได้เปิดใช้งานจริงแล้ว โดยมีวัตถุประสงค์เพื่อให้ระบบสามารถทำงานได้อย่างมีประสิทธิภาพและตอบสนองความต้องการของผู้ใช้งานที่เปลี่ยนแปลงไปอยู่เสมอ ซึ่งประกอบด้วย 3 กิจกรรมหลัก ได้แก่

2.1.7.1**การแก้ไขข้อผิดพลาด (Corrective Maintenance): เป็นการแก้ไขข้อผิดพลาดต่างๆ ที่เกิดขึ้นระหว่างการใช้งานจริง ซึ่งอาจเกิดจากความผิดพลาดในการเขียนโปรแกรม ข้อผิดพลาดในการออกแบบ หรือปัญหาที่เกิดจากสภาพแวดล้อมการใช้งานจริง

1)**การตรวจสอบข้อผิดพลาด (Bug Tracking): ผู้ใช้งานหรือผู้ดูแลระบบจะรายงานข้อผิดพลาดที่พบให้กับทีมพัฒนา

2)**การวิเคราะห์สาเหตุ (Root Cause Analysis): ทีมพัฒนาจะทำการวิเคราะห์สาเหตุของข้อผิดพลาด เพื่อหาแนวทางแก้ไขที่เหมาะสม

3)**การแก้ไขข้อผิดพลาด (Bug Fixing): ทีมพัฒนาจะทำการแก้ไขโค้ดโปรแกรมหรือส่วนประกอบอื่นๆ ของระบบเพื่อแก้ไขข้อผิดพลาด

4)**การทดสอบ (Testing): หลังจากแก้ไขข้อผิดพลาดแล้ว จะต้องทำการทดสอบระบบอีกครั้ง เพื่อให้แน่ใจว่าการแก้ไขไม่ส่งผลกระทบต่อส่วนอื่นๆ ของระบบ

2.1.7.2**การปรับปรุงประสิทธิภาพ (Perfective Maintenance) เป็นการปรับปรุงประสิทธิภาพการทำงานของโปรแกรมให้ดียิ่งขึ้น เช่น เพิ่มความเร็วในการประมวลผล ลดการใช้ทรัพยากรหรือปรับปรุงการแสดงผล

1)**การวิเคราะห์ ประสิทธิภาพ (Performance Analysis): ตรวจสอบประสิทธิภาพของระบบในด้านต่างๆ เช่น เวลาในการตอบสนอง การใช้หน่วยความจำ และการใช้ CPU

2)**การปรับปรุงโค้ด (Code Optimization): ปรับปรุงโค้ดโปรแกรมเพื่อเพิ่มประสิทธิภาพในการทำงาน

3)**การปรับปรุงฐานข้อมูล (Database Optimization): ปรับแต่งฐานข้อมูลเพื่อให้การเข้าถึงและประมวลผลข้อมูลรวดเร็วขึ้น

4)**การปรับปรุงโครงสร้างระบบ (System Architecture Optimization): ปรับปรุงโครงสร้างของระบบ เพื่อให้ระบบสามารถรองรับการใช้งานที่เพิ่มขึ้นได้

2.1.7.3**การเพิ่มเติมฟังก์ชันใหม่ (Adaptive Maintenance): เป็นการเพิ่มฟังก์ชันการทำงานใหม่ๆ หรือปรับปรุงฟังก์ชันเดิมให้ตอบสนองความต้องการของผู้ใช้งานที่เปลี่ยนแปลงไป

1)**การรวบรวมความต้องการใหม่ (Requirement Gathering): รับฟังและรวบรวมความต้องการใหม่ๆ จากผู้ใช้งาน

2)**การวิเคราะห์ความต้องการ (Requirement Analysis): วิเคราะห์ความต้องการใหม่เพื่อประเมินความเป็นไปได้และผลกระทบต่อระบบ

3)**การวิเคราะห์ความต้องการ (Requirement Analysis): วิเคราะห์ความต้องการใหม่เพื่อประเมินความเป็นไปได้และผลกระทบต่อระบบ

4)**การทดสอบ (Testing): ทดสอบฟังก์ชันใหม่หรือฟังก์ชันที่ปรับปรุงแล้วเพื่อให้แน่ใจว่าทำงานได้ถูกต้อง

2.2**Bootstrap Front-End Framework

Bootstrap เป็นเฟรมเวิร์ก (Framework) แบบ Open-source สำหรับการพัฒนาเว็บแอปพลิเคชันและเว็บไซต์ โดยเน้นการออกแบบที่ตอบสนองต่อทุกขนาดหน้าจอ (Responsive Design) และใช้งานง่าย Bootstrap ได้รับความนิยมอย่างแพร่หลายในหมู่นักพัฒนาเว็บ เนื่องจากช่วยลดเวลาและความยุ่งยากในการสร้างเว็บไซต์ที่มีความสวยงามและทันสมัย Bootstrap คือชุดเครื่องมือ (Toolkit) ที่ประกอบด้วยโค้ด HTML, CSS, และ JavaScript ที่เขียนไว้ล่วงหน้า ซึ่งมีองค์ประกอบต่างๆ ที่จำเป็นสำหรับการสร้างเว็บไซต์ เช่น ระบบ Grid สำหรับจัดวางเลย์เอาต์, ส่วนประกอบ UI (User Interface) ต่างๆ เช่น ปุ่ม, เมนู, แบบฟอร์ม, และอื่นๆ, รวมถึงปลั๊กอิน JavaScript ที่ช่วยเพิ่มความสามารถในการโต้ตอบและสร้างลูกเล่นต่างๆ ให้กับเว็บไซต์ Bootstrap สามารถนำไปใช้ในการพัฒนาเว็บไซต์และเว็บแอปพลิเคชันได้หลากหลายประเภท เช่น

2.2.1**เว็บไซต์ทั่วไป: เว็บไซต์บริษัท, เว็บไซต์ส่วนตัว, เว็บไซต์ข่าวสาร, เว็บไซต์พอร์ทัลโอ

2.2.2**เว็บแอปพลิเคชัน: ระบบจัดการข้อมูล, ระบบอีคอมเมิร์ซ, ระบบลงทะเบียน, ระบบแจ้งซ่อม

2.2.3** Dashboard: หน้าจอแสดงผลข้อมูลสรุปสำหรับผู้ดูแลระบบ

2.2.4** Prototype: สร้างต้นแบบเว็บไซต์หรือแอปพลิเคชันเพื่อทดสอบแนวคิดก่อนการพัฒนาจริงเป็นต้น

2.2.5**ประโยชน์ของ Bootstrap

2.2.5.1**ประหยัดเวลาในการพัฒนา: Bootstrap มีองค์ประกอบต่างๆ ที่จำเป็นสำหรับการสร้างเว็บไซต์เตรียมไว้ให้แล้ว ทำให้นักพัฒนาไม่ต้องเสียเวลาเขียนโค้ดตั้งแต่เริ่มต้น

2.2.5.2**รองรับการแสดงผลบนอุปกรณ์ที่หลากหลาย (Responsive Design): Bootstrap ถูกออกแบบมาให้อรองรับการแสดงผลบนอุปกรณ์ที่หลากหลาย เช่น คอมพิวเตอร์ตั้งโต๊ะ แท็บเล็ต และสมาร์ทโฟน โดยเว็บไซต์จะปรับขนาดและรูปแบบการแสดงผลให้เหมาะสมกับขนาดหน้าจอของอุปกรณ์นั้นๆ โดยอัตโนมัติ

2.2.5.3**ใช้งานง่าย: Bootstrap มีเอกสารคู่มือที่ละเอียดและชัดเจน พร้อมตัวอย่างโค้ดที่สามารถนำไปใช้งานได้ทันที ทำให้นักพัฒนาสามารถเรียนรู้และใช้งานได้ง่ายแม้ไม่มีประสบการณ์มาก่อน

2.2.5.4**มีชุมชนผู้ใช้งานขนาดใหญ่: Bootstrap มีชุมชนผู้ใช้งานและนักพัฒนาที่แข็งแกร่ง ทำให้สามารถหาข้อมูล คำแนะนำ และการแก้ไขปัญหาได้ง่าย

2.2.5.5**ปรับแต่งได้ง่าย: Bootstrap สามารถปรับแต่งรูปแบบและลักษณะการทำงานได้หลากหลายตามความต้องการของผู้พัฒนา โดยสามารถปรับแต่งได้ทั้ง CSS และ JavaScript ฟรีและเป็น Open Source: Bootstrap สามารถดาวน์โหลดและใช้งานได้ฟรี และเปิดให้ผู้ใช้งานสามารถนำไปพัฒนาต่อยอดได้

2.2.5.6**มีคอมโพเนนต์ที่หลากหลาย: Bootstrap มีคอมโพเนนต์สำเร็จรูปมากมาย เช่น ปุ่ม, เมนู, แบบฟอร์ม, รูปภาพ, ไอคอน, ตาราง, และอื่นๆ ทำให้สามารถสร้างเว็บไซต์ได้อย่างรวดเร็ว

2.2.5.7**มีระบบ Grid ที่ยืดหยุ่น: ระบบ Grid ของ Bootstrap ช่วยให้สามารถจัดวางเลย์เอาต์ของเว็บไซต์ได้ง่ายและยืดหยุ่น รองรับการแสดงผลบนอุปกรณ์ที่หลากหลาย

2.2.5.8**มีปลั๊กอิน JavaScript ที่มีประโยชน์: Bootstrap มาพร้อมกับปลั๊กอิน JavaScript ที่ช่วยเพิ่มความสามารถในการโต้ตอบให้กับเว็บไซต์ เช่น การสร้าง Carousel, Modal, Tooltip, Popover, และอื่นๆ

2.2.5.9**มีธีม (Theme) ให้เลือกใช้งาน: Bootstrap มีธีมให้เลือกใช้งานมากมาย ซึ่งช่วยให้สามารถปรับแต่งรูปแบบของเว็บไซต์ได้ง่ายและรวดเร็ว

2.2.5.10**เข้ากันได้กับเบราว์เซอร์ส่วนใหญ่: Bootstrap ได้รับการทดสอบและปรับแต่งให้เข้ากันได้กับเบราว์เซอร์ส่วนใหญ่ ทำให้มั่นใจได้ว่าเว็บไซต์จะแสดงผลได้อย่างถูกต้องบนเบราว์เซอร์ต่างๆ

2.3**การออกแบบเว็บแอปพลิเคชัน

การออกแบบเว็บแอปพลิเคชัน (Web Application Design) คือ กระบวนการสร้างและวางแผนส่วนประกอบต่างๆ ของเว็บแอปพลิเคชัน ซึ่งรวมถึงโครงสร้างเนื้อหา (Information Architecture), รูปแบบการใช้งาน (User Interface), ประสบการณ์ผู้ใช้ (User Experience), และการออกแบบภาพ (Visual Design) โดยมีจุดมุ่งหมายเพื่อสร้างเว็บแอปพลิเคชันที่ใช้งานง่าย มีประสิทธิภาพ และตอบสนองความต้องการของผู้ใช้งานได้อย่างครบถ้วน หลักการออกแบบเว็บแอปพลิเคชัน มีดังนี้

2.3.1**ความชัดเจนและเข้าใจง่าย (Clarity and Simplicity): เว็บแอปพลิเคชันควรมีโครงสร้างที่ชัดเจน ใช้งานง่าย และไม่ซับซ้อน ผู้ใช้งานควรสามารถเข้าใจและใช้งานฟังก์ชันต่างๆ ได้โดยไม่ต้องใช้ความพยายามมากนัก การนำทาง (Navigation) ควรชัดเจนและใช้งานง่าย ผู้ใช้ควรสามารถค้นหาสิ่งที่ต้องการได้อย่างรวดเร็วการออกแบบควรเรียบง่ายและไม่รกเกินไป ใช้พื้นที่ว่าง (White space) อย่างเหมาะสมเพื่อให้หน้าเว็บดูสะอาดตาใช้ภาษาที่เข้าใจง่ายและตรงไปตรงมา หลีกเลี่ยงการใช้คำศัพท์เฉพาะทางที่ผู้ใช้อาจไม่เข้าใจ

2.3.2**ความสอดคล้อง (Consistency): การออกแบบควรมีความสอดคล้องกันทั้งในเรื่องของรูปแบบ สี สัน ตัวอักษร และการจัดวางองค์ประกอบต่างๆ เพื่อให้ผู้ใช้งานคุ้นเคยและใช้งานได้ง่ายใช้รูปแบบการออกแบบ (Design pattern) ที่เป็นมาตรฐานและคุ้นเคยกับผู้ใช้ ใช้สีสันและตัวอักษรที่สอดคล้องกันตลอดทั้งเว็บแอปพลิเคชัน จัดวางองค์ประกอบต่างๆ ในตำแหน่งที่สอดคล้องกันในทุกหน้า

2.3.3**ความสวยงาม (Aesthetics): เว็บแอปพลิเคชันควรมีการออกแบบที่สวยงาม น่าสนใจ และทันสมัย เพื่อดึงดูดความสนใจของผู้ใช้งาน ใช้ภาพและกราฟิกที่สวยงามและมีความละเอียดสูงใช้สีสันทันที่เหมาะสมและสบายตา ใช้ตัวอักษรที่อ่านง่ายและสวยงามออกแบบโดยคำนึงถึงหลักการออกแบบที่ดี เช่น ความสมดุล ความเปรียบเทียบ และจุดเด่น

2.3.4**เว็บแอปพลิเคชันควรตอบสนองต่อการกระทำของผู้ใช้ได้อย่างรวดเร็วและแม่นยำ เพื่อให้ผู้ใช้งานรู้สึกว่าการระบบมีประสิทธิภาพ ลดเวลาในการโหลดหน้าเว็บให้น้อยที่สุดให้ feedback แก่ผู้ใช้เมื่อทำการกระทำใดๆ เช่น การคลิกปุ่มหลักเกี่ยวกับการใช้ animation หรือ transition ที่มากเกินไป ซึ่งอาจทำให้ผู้ใช้รู้สึกเวียนหัวหรือเสียสมาธิ

2.3.5**การเข้าถึงได้ง่าย (Accessibility): เว็บแอปพลิเคชันควรออกแบบให้ผู้ใช้งานทุกกลุ่มสามารถเข้าถึงและใช้งานได้อย่างเท่าเทียมกัน รวมถึงผู้พิการหรือผู้ที่มีข้อจำกัดในการใช้งาน

2.3.6**ความปลอดภัย (Security): เว็บแอปพลิเคชันควรมีระบบรักษาความปลอดภัยที่แข็งแกร่ง เพื่อป้องกันการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต และป้องกันการโจมตีจากผู้ไม่หวังดี ขั้นตอนการออกแบบเว็บแอปพลิเคชัน

2.3.7**การวิเคราะห์ความต้องการ (Requirement Analysis): ศึกษาและทำความเข้าใจความต้องการของผู้ใช้งานและวัตถุประสงค์ของเว็บแอปพลิเคชัน

2.3.8**การออกแบบโครงสร้าง (Information Architecture Design): ออกแบบโครงสร้างของเนื้อหาและการจัดหมวดหมู่ข้อมูล เพื่อให้ผู้ใช้งานสามารถค้นหาข้อมูลที่ต้องการได้ง่าย

2.3.9**การออกแบบรูปแบบการใช้งาน (User Interface Design): ออกแบบหน้าจอและส่วนประกอบต่างๆ ของเว็บแอปพลิเคชัน เช่น ปุ่ม เมนู แบบฟอร์ม และการแสดงผลข้อมูล

2.3.10**การออกแบบประสบการณ์ผู้ใช้ (User Experience Design): ออกแบบให้เว็บแอปพลิเคชันใช้งานง่าย สะดวก และเป็นไปตามความคาดหวังของผู้ใช้

2.1.11**การออกแบบภาพ (Visual Design): ออกแบบภาพลักษณ์โดยรวมของเว็บแอปพลิเคชัน เช่น สี สัน ตัวอักษร และรูปแบบกราฟิก

2.1.12**การพัฒนา (Development): นำการออกแบบไปพัฒนาเป็นเว็บแอปพลิเคชันที่ใช้งานได้จริง

2.2.13**การทดสอบ (Testing): ทดสอบการทำงานของเว็บแอปพลิเคชัน เพื่อให้แน่ใจว่าระบบทำงานได้อย่างถูกต้องและมีประสิทธิภาพ

ข้อจำกัดของ Bootstrap ขนาดไฟล์ใหญ่ ไฟล์ CSS และ JavaScript ของ Bootstrap มีขนาดค่อนข้างใหญ่ ซึ่งอาจส่งผลต่อความเร็วในการโหลดหน้าเว็บ อย่างไรก็ตาม สามารถลดขนาดไฟล์ได้โดยการเลือกใช้เฉพาะส่วนประกอบที่จำเป็น ความซับซ้อนของโค้ด เนื่องจาก Bootstrap มีคอมโพเนนต์สำเร็จรูปมากมาย ทำให้บางครั้งอาจเกิดความซ้ำซ้อนของโค้ดได้

อาจไม่เหมาะกับเว็บไซต์ที่มีความซับซ้อน Bootstrap เหมาะสำหรับการสร้างเว็บไซต์ที่มีรูปแบบมาตรฐาน แต่หากต้องการสร้างเว็บไซต์ที่มีรูปแบบที่ซับซ้อนหรือแตกต่างจากเดิมมาก อาจต้องใช้ความพยายามในการปรับแต่งเพิ่มเติม

2.4**ฐานข้อมูลและการออกแบบฐานข้อมูล

ฐานข้อมูล (Database) คือ แหล่งรวบรวมข้อมูลที่สำคัญของระบบแจ้งซ่อม โดยทำหน้าที่จัดเก็บข้อมูลต่างๆ ที่เกี่ยวข้องกับการแจ้งซ่อม การดำเนินการซ่อมบำรุง และข้อมูลอื่นๆ ที่เกี่ยวข้อง การออกแบบฐานข้อมูลที่ดียังจะช่วยให้ระบบสามารถจัดเก็บและเรียกใช้ข้อมูลได้อย่างมีประสิทธิภาพ ถูกต้อง และปลอดภัยการจัดเก็บข้อมูล: ฐานข้อมูลทำหน้าที่จัดเก็บข้อมูลที่จำเป็นสำหรับการทำงานของระบบแจ้งซ่อม เช่น ข้อมูลผู้ใช้งาน ข้อมูลอุปกรณ์ ข้อมูลการแจ้งซ่อม และข้อมูลการดำเนินการซ่อมบำรุง ซึ่งข้อมูลเหล่านี้มีความสำคัญต่อการทำงานของระบบและการตัดสินใจของผู้ใช้งานและผู้ดูแลระบบการจัดการข้อมูล: ฐานข้อมูลมีระบบการจัดการข้อมูลที่มีประสิทธิภาพ ช่วยให้สามารถจัดเก็บข้อมูลจำนวนมากได้อย่างเป็นระบบและมีโครงสร้าง ทำให้สามารถค้นหา เรียกใช้ และวิเคราะห์ข้อมูลได้ง่ายและรวดเร็วความต้องการของข้อมูล: ฐานข้อมูลมีกลไกในการตรวจสอบและรักษาความถูกต้องของข้อมูล เช่น การตรวจสอบความซ้ำซ้อน การตรวจสอบรูปแบบข้อมูล และการตรวจสอบความสอดคล้องของข้อมูล ซึ่งช่วยลดความผิดพลาดที่อาจเกิดขึ้นจากการบันทึกข้อมูลด้วยมือความปลอดภัยของข้อมูล: ฐานข้อมูลมีระบบรักษาความปลอดภัยเพื่อป้องกันการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต และป้องกันการสูญหายของข้อมูลจากภัยคุกคามต่างๆ เช่น การโจมตีจากภายนอก อุบัติเหตุ หรือความผิดพลาดของระบบ เช่น การกำหนดสิทธิ์การเข้าถึง การเข้ารหัสข้อมูล และการสำรองข้อมูลการสนับสนุนการทำงานของระบบ: ฐานข้อมูลเป็นแหล่งข้อมูลที่สำคัญสำหรับการทำงานของระบบแจ้งซ่อม เช่น การแสดงสถานะการซ่อม การสร้างรายงาน และการวิเคราะห์ข้อมูล ซึ่งช่วยให้ผู้ใช้งานและเจ้าหน้าที่สามารถเข้าถึงข้อมูลที่จำเป็นได้อย่างสะดวกและรวดเร็ว นอกจากนี้ ฐานข้อมูลยังสามารถใช้ในการสร้างรายงานสรุปข้อมูลต่างๆ ที่เกี่ยวข้องกับการซ่อมบำรุง เพื่อนำไปวิเคราะห์และปรับปรุงประสิทธิภาพในการทำงานได้โดยสรุปแล้ว ฐานข้อมูลมีบทบาทสำคัญในการจัดเก็บ จัดการ และรักษาความปลอดภัยของข้อมูล ซึ่งเป็นองค์ประกอบสำคัญที่ช่วยให้ระบบแจ้งซ่อมออนไลน์ทำงานได้อย่างมีประสิทธิภาพและเป็นประโยชน์ต่อการบริหารจัดการงานซ่อมบำรุงภายในองค์กร

2.4.1**ความสำคัญของฐานข้อมูลในการพัฒนาระบบแจ้งซ่อม

ฐานข้อมูล (Database) เป็นองค์ประกอบสำคัญของระบบแจ้งซ่อม ทำหน้าที่จัดเก็บข้อมูลที่เกี่ยวข้องกับการซ่อมบำรุงทั้งหมด อาทิ ข้อมูลผู้ใช้งาน ข้อมูลอุปกรณ์ ข้อมูลการแจ้งซ่อม และข้อมูลการดำเนินการ

ซ่อมบำรุง ฐานข้อมูลที่มีประสิทธิภาพจะช่วยเพิ่มประสิทธิภาพการทำงานของระบบแจ้งซ่อม ลดความผิดพลาด และเพิ่มความพึงพอใจของผู้ใช้งานความสำคัญของฐานข้อมูล มีดังนี้

2.4.1.1**การจัดเก็บข้อมูลอย่างเป็นระบบ: ฐานข้อมูลช่วยจัดเก็บข้อมูลอย่างเป็นระบบและมีโครงสร้าง ทำให้สามารถค้นหาและเรียกใช้ข้อมูลได้ง่าย ลดความซับซ้อนในการจัดเก็บข้อมูลจำนวนมาก และช่วยให้สามารถเข้าถึงข้อมูลที่ต้องการได้อย่างรวดเร็ว

2.4.1.2**การจัดเก็บข้อมูลอย่างเป็นระบบ: ฐานข้อมูลช่วยจัดเก็บข้อมูลอย่างเป็นระบบและมีโครงสร้าง ทำให้สามารถค้นหาและเรียกใช้ข้อมูลได้ง่าย ลดความซับซ้อนในการจัดเก็บข้อมูลจำนวนมาก และช่วยให้สามารถเข้าถึงข้อมูลที่ต้องการได้อย่างรวดเร็ว

2.4.1.3**การรักษาความปลอดภัยของข้อมูล: ฐานข้อมูลมีระบบรักษาความปลอดภัยเพื่อป้องกันการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต และป้องกันการสูญหายของข้อมูลจากภัยคุกคามต่างๆ เช่น การโจมตีจากภายนอก อุบัติเหตุ หรือความผิดพลาดของระบบ

2.4.1.4**การสนับสนุนการทำงานของระบบ: ฐานข้อมูลเป็นแหล่งข้อมูลที่สำคัญสำหรับการทำงานของระบบแจ้งซ่อม เช่น การแสดงสถานะการซ่อม การสร้างรายงาน และการวิเคราะห์ข้อมูล ซึ่งช่วยให้ผู้ใช้งานและเจ้าหน้าที่สามารถเข้าถึงข้อมูลที่จำเป็นได้อย่างสะดวกและรวดเร็ว

2.4.1.5**การสนับสนุนการทำงานของระบบ: ฐานข้อมูลเป็นแหล่งข้อมูลที่สำคัญสำหรับการทำงานของระบบแจ้งซ่อม เช่น การแสดงสถานะการซ่อม การสร้างรายงาน และการวิเคราะห์ข้อมูล ซึ่งช่วยให้ผู้ใช้งานและเจ้าหน้าที่สามารถเข้าถึงข้อมูลที่จำเป็นได้อย่างสะดวกและรวดเร็ว

2.4.2**การออกแบบฐานข้อมูล (Database Design) การออกแบบฐานข้อมูลเป็นขั้นตอนสำคัญในการพัฒนาระบบแจ้งซ่อม โดยต้องคำนึงถึงปัจจัยต่างๆ ดังนี้

2.4.2.1**การวิเคราะห์ความต้องการ (Requirement Analysis): ในขั้นตอนนี้ จะทำการศึกษาและทำความเข้าใจข้อมูลที่ต้องจัดเก็บในระบบ เช่น ข้อมูลผู้ใช้งาน, ข้อมูลอุปกรณ์, ข้อมูลการแจ้งซ่อม, ข้อมูลสถานะการซ่อม, และข้อมูลอื่นๆ ที่เกี่ยวข้อง นอกจากนี้ยังต้องพิจารณาถึงความสัมพันธ์ระหว่างข้อมูลต่างๆ เพื่อให้สามารถเชื่อมโยงข้อมูลได้อย่างถูกต้อง

2.4.2.2**การออกแบบโครงสร้าง (Conceptual Design): ขั้นตอนนี้จะแปลงความต้องการที่ได้จากการวิเคราะห์ มาเป็นแผนภาพแสดงความสัมพันธ์ของเอนทิตี (Entity Relationship Diagram ERD) ซึ่งจะแสดงให้เห็นถึงเอนทิตี (สิ่งที่ต้องการจัดเก็บข้อมูล) แอตทริบิวต์ (คุณสมบัติของเอนทิตี) และความสัมพันธ์ระหว่างเอนทิตี โดยมี 3 รูปแบบ One to One: เอนทิตีหนึ่งสามารถเชื่อมโยงกับอีกเอนทิตีหนึ่งได้เพียงหนึ่งต่อหนึ่งเท่านั้น One to Many: เอนทิตีหนึ่งสามารถเชื่อมโยงกับอีกเอนทิตีได้หลายเอนทิตี แต่เอนทิตีที่สองจะเชื่อมโยงกับเอนทิตีแรกได้เพียงหนึ่งเอนทิตี Many to Many: เอนทิตีหนึ่งสามารถเชื่อมโยงกับอีกเอนทิตีได้หลายเอนทิตี และในทางกลับกัน

2.4.2.3**การเลือกชนิดข้อมูล (Data Type Selection): เลือกชนิดข้อมูลที่เหมาะสมสำหรับแต่ละฟิลด์ เช่น ข้อความ (varchar), ตัวเลข (int), วันที่ (date), หรือชนิดข้อมูลอื่นๆ ที่เหมาะสม

กับลักษณะของข้อมูล เพื่อให้สามารถจัดเก็บข้อมูลได้อย่างถูกต้องและมีประสิทธิภาพ

2.4.2.4**การออกแบบดัชนี (Index Design) สร้างดัชนี (Index) ให้กับฟิลด์ที่ใช้ในการค้นหามือๆ เพื่อเพิ่มประสิทธิภาพในการค้นหาข้อมูล ทำให้การค้นหาข้อมูลทำได้รวดเร็วขึ้น

2.4.2.5**การปรับแต่ง (Normalization) ปรับปรุงโครงสร้างฐานข้อมูลให้เป็นไปตามหลักการ Normalization เพื่อลดความซ้ำซ้อนของข้อมูลและเพิ่มความยืดหยุ่นในการจัดการข้อมูล ซึ่งจะช่วยลดปัญหาความผิดพลาดของข้อมูลและเพิ่มประสิทธิภาพในการจัดเก็บและเรียกใช้ข้อมูล

ตาราง 2.4 การออกแบบฐานข้อมูลสำหรับระบบแจ้งซ่อม

ตาราง	ฟิลด์	คำอธิบาย
users	user_id, username,password,name, email, role	เก็บข้อมูลผู้ใช้งานระบบ เช่น ชื่อผู้ใช้ รหัสผ่าน ชื่อ อีเมล และบทบาท (ผู้ใช้งาน, เจ้าหน้าที่, ผู้ดูแลระบบ)
devices	device_id, device_name, type, model, location	เก็บข้อมูลอุปกรณ์ต่างๆ ในแผนก เช่น ชื่ออุปกรณ์ ประเภท รุ่น และสถานที่ตั้ง
requests	request_id,user_id,device_id,problem_description, status, created_at, updated_at	เก็บข้อมูลการแจ้งซ่อม เช่น เลขที่ใบแจ้งซ่อม ผู้แจ้ง อุปกรณ์ที่แจ้งซ่อม รายละเอียดปัญหา สถานะการซ่อม และวันที่แจ้งซ่อม
statuses	status_id, status_name	เก็บสถานะการซ่อมต่างๆ เช่น รอตรวจสอบ กำลังดำเนินการ ซ่อมเสร็จแล้ว
problem_types	problem_type_id, problem_type_name	เก็บประเภทของปัญหาที่อาจเกิดขึ้นกับอุปกรณ์ เช่น ปัญหาฮาร์ดแวร์ ปัญหาซอฟต์แวร์ ปัญหาเครือข่าย
repair_histories	repair_history_id, request_id, technician_id, action_taken	เก็บประวัติการซ่อมอุปกรณ์แต่ละครั้ง โดยเชื่อมโยงกับตาราง requests และ technicians เพื่อบันทึกว่าการซ่อมนั้นเกี่ยวข้องกับใบแจ้งซ่อมใด

