

# บทที่ 1

## บทนำ

### 1.1 หลักการและเหตุผล

งานด้านการพัฒนาซอฟต์แวร์ (software development) มีวิธีการจัดการโครงการ(Project Management Methodology) ยกตัวอย่างการใช้รูปแบบการจัดการแบบน้ำตก (Waterfall Modal) [1] แบ่งเป็นลำดับการทำงานดังนี้ 1. การวางแผน (Planning) เป็นกระบวนการวิเคราะห์ความต้องการของลูกค้าในการพัฒนาซอฟต์แวร์ 2. การออกแบบ (Design Software) เป็นกระบวนการ ออกแบบซอฟต์แวร์ให้ตรงตามความต้องการ 3.การพัฒนา (Software development) เป็นกระบวนการพัฒนาซอฟต์แวร์ตามที่ออกแบบ 4.การทดสอบ (Software Testing) เป็นกระบวนการทดสอบซอฟต์แวร์ 5. การนำไปใช้ (Integration) เป็นกระบวนการนำซอฟต์แวร์ไปใช้

ในกระบวนการทำงานข้างต้น การสื่อสารระหว่าง นักทดสอบ (Software Tester) กับ นักพัฒนา (Software Developer) มีการจัดการรายงานข้อบกพร่อง (Defect Management Report) ที่นักทดสอบตรวจพบ เพื่อส่งให้นักพัฒนาได้แก้ไขจุดนั้น เนื่องจากการทำงานแบบ Work From Home ทำให้การสื่อสารลำบากจากแบบปกติทำให้เกิดความล่าช้าและข้อผิดพลาดในการสื่อสาร

ที่ผู้พัฒนาได้คิดพัฒนาระบบรายงานข้อผิดพลาดซอฟต์แวร์จากนักทดสอบกับนักพัฒนาให้มีการสื่อสารที่สะดวกและรวดเร็วขึ้น และด้วยการที่ต้องทำงานที่บ้านทำให้การใช้ระบบรายงานนี้เหมาะสมกับสถานการณ์ในปัจจุบันวัตถุประสงค์ของโครงการ

### 1.2 ขอบเขตของโครงการ

#### 1.2.1 ผู้เกี่ยวข้องในระบบ

ผู้เกี่ยวข้องในระบบประกอบด้วย ผู้จัดการโครงการ (Project Manager), นักทดสอบ (Software Tester ) และนักพัฒนา (Software Developer)

#### 1. ฟังก์ชันการจัดการและการทำงานของ Project Manager

- สามารถเข้าสู่ระบบได้
- สามารถจัดการข้อมูลส่วนตัวได้
- สามารถจัดการสมาชิกทีม ซึ่งประกอบไปด้วย Software Tester และ Software Developer ได้
- สามารถจัดการ Project ได้
- สามารถดูภาพรวมของ Project ได้

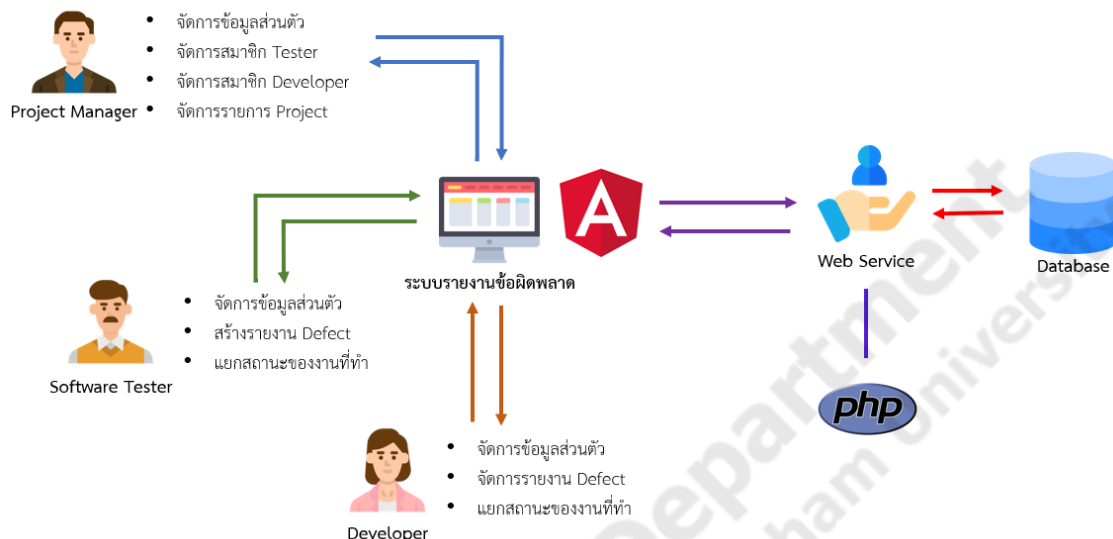
- กำหนดการแก้ไขรายงาน Defect ให้กับ Software Developer ได้

## 2. ฟังก์ชันการจัดการและการทำงานของ Software Tester

- สามารถเข้าสู่ระบบได้
- สามารถจัดการข้อมูลส่วนตัวได้
- สามารถเข้าดูได้เฉพาะ Project ที่เป็นสมาชิก
- สามารถสร้าง Test Case ประกอบด้วยข้อมูลดังนี้
  - ชื่อ Project
  - ชื่อผู้สร้าง
  - รหัสฟังก์ชัน(Test Module ID)
  - ชื่อฟังก์ชัน(Test Module Name)
  - รหัสกรณีทดสอบ(Test Case ID)
  - กรณีที่ทดสอบ(Test Case)
  - ข้อมูลที่ใช้ทดสอบ(Test Data)
  - ขั้นตอนการทดสอบ(Test Steps)
  - ผลลัพธ์ที่คาดหวัง(Expected Result)
  - สถานะ(Status) ได้แก่ Pass และ Fail
- สามารถสร้างรายงานข้อผิดพลาด (Defect) ได้ ประกอบด้วยข้อมูลดังนี้
  - รหัสรายงาน
  - ชื่อ Project
  - ชื่อผู้รายงาน
  - ระดับความรุนแรงได้แก่ Critical, Major, Normal และ Minor
  - วันที่ครบกำหนด
  - สถานะ ประกอบด้วย Backlog, Doing, Ready for test และ Done
  - วันที่สร้างรายงาน
  - รายละเอียด Defect ประกอบด้วย
    - สิ่งที่เกิดขึ้น (Title)

- รายละเอียดของสภาพแวดล้อมระบบที่เจอปัญหา (Environment Details)
  - ขั้นตอนการทำให้เกิดปัญหา (Step To Reproduce)
  - ผลลัพธ์ที่เกิดขึ้น (Actual Result)
  - ผลลัพธ์ที่คาดหวังตาม Requirement (Expected Result)
  - เอกสารแนบ (Attachment) ได้แก่ รูปภาพ วิดีโอ และไฟล์เอกสาร
- สามารถแจ้งรายงาน Defect ไปยัง Project Manager (Team Leader) และ Software Developer
  - สามารถแสดงลำดับความสำคัญของ Defect ที่ทดสอบได้
  - สามารถแยกสถานะของงานที่ทดสอบแล้วไม่มีปัญหาได้
  - สร้างรายงาน Defect โดยการนำข้อมูลจาก Defect ที่มีอยู่แล้วได้
3. ฟังก์ชันการจัดการและการทำงานของ Software Developer
- สามารถเข้าสู่ระบบได้
  - สามารถจัดการข้อมูลส่วนตัวได้
  - สามารถเข้าดูได้เฉพาะ Project ที่เป็นสมาชิก
  - สามารถเลือกแก้ไขรายงาน Defect ได้
  - สามารถจัดการรายงาน Defect ได้ ประกอบด้วยข้อมูลดังนี้
    - ยืนยันการแก้ไข ข้อมูลจะประกอบด้วย
      - วันที่รับรายงาน
      - ชื่อผู้แก้ไข Defect
    - ปฏิเสธการแก้ไข ประกอบด้วยสาเหตุที่ปฏิเสธ
  - แจ้งรายงาน Defect ไปยัง Software Tester
  - สามารถแสดงลำดับความสำคัญของ Defect ที่ต้องแก้ไขได้
  - สามารถแยกสถานะของงาน ประกอบด้วย งานที่ต้องทำ งานที่กำลังทำ และงานที่ทำเสร็จแล้วได้
4. ฟังก์ชันการแจ้งเตือน
- แจ้งเตือนผู้ทดสอบ (Software Tester) กับผู้พัฒนา (Software Developer) เมื่อมีการรายงาน

### 1.3 ภาพรวมของระบบ



ภาพประกอบที่ 1.1 ภาพรวมระบบ

จากภาพประกอบที่ 1.1 เป็นภาพรวมของระบบสามารถอธิบายได้ดังนี้

Project Manager สามารถแก้ไขข้อมูลส่วนตัว สามารถเพิ่มสมาชิกของทีม Tester และเพิ่มสมาชิกทีม Developer ได้ และสามารถเพิ่มโปรเจกต์ใหม่เข้าระบบได้

Software Developer สามารถแก้ไขข้อมูลส่วนตัวได้ สามารถเลือกได้ว่าจะแก้ไข Defect ไหน และสามารถแยกสถานะของงานที่กำลังทำอยู่ได้ว่างานไหนที่กำลังทำ หรือทำเสร็จแล้ว

Software Tester สามารถแก้ไขข้อมูลส่วนตัวได้ เมื่อทดสอบระบบนั้น ๆ แล้วเกิด Defect สามารถสร้างรายงาน Defect ได้ และสามารถแยกสถานะของงานได้ว่าทำเสร็จแล้ว

### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

ได้เว็บไซต์ที่ใช้ในการสื่อสารกันระหว่าง Software Tester และ Software Developer ให้ทำงานร่วมกันได้อย่างสะดวกสบาย

### 1.5 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินงาน

#### 1.5.1 ฮาร์ดแวร์

เครื่องคอมพิวเตอร์ 2 เครื่องมีคุณสมบัติดังต่อไปนี้

##### 1. Asus

- CPU AMD RYZEN 5 4600H

- RAM 16 GB DDR4 3200MHz
- 64-bit operating system, x64-based processor

## 2. Lenovo

- Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz
- RAM 8 GB DDR4
- 64-bit operating system, x64-based processor

### 1.5.2 ซอฟต์แวร์

1. ภาษาที่ใช้ในการเขียน HTML, CSS, TypeScript, php, Angular
2. โปรแกรมที่ใช้ Visual Studio Code
3. ฐานข้อมูล MySQL
4. โปรแกรมจัดการฐานข้อมูล phpMyAdmin

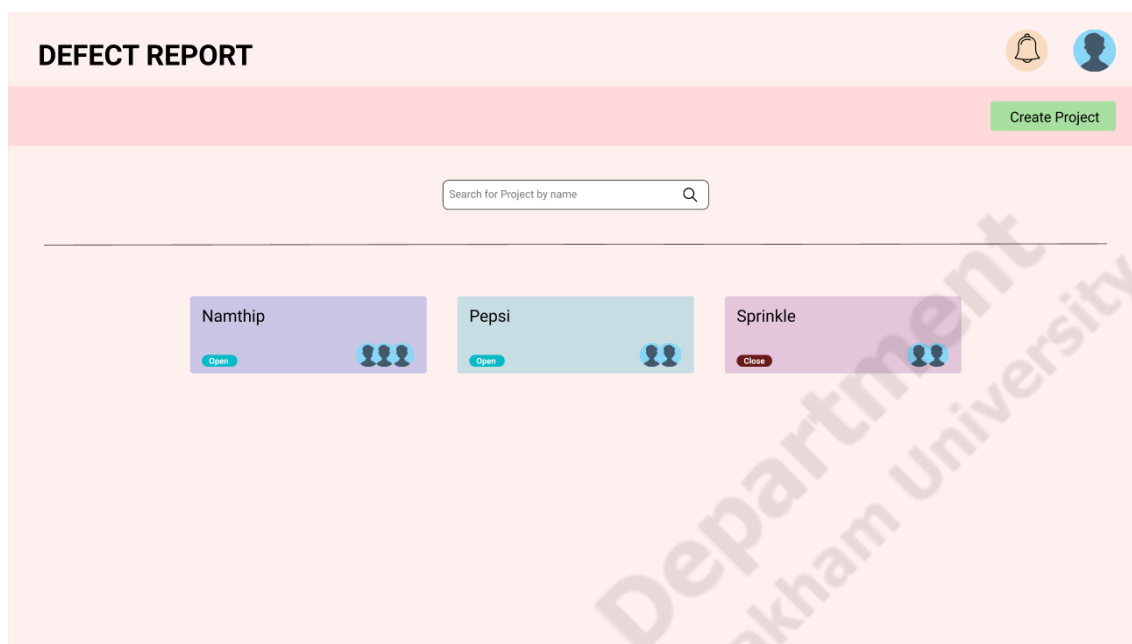
### 1.6 แผนการดำเนินงาน

โครงการปริญญาโทฉบับนี้ ดำเนินงาน ณ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคามระหว่างเดือน สิงหาคม 2564 ถึง ตุลาคม 2564

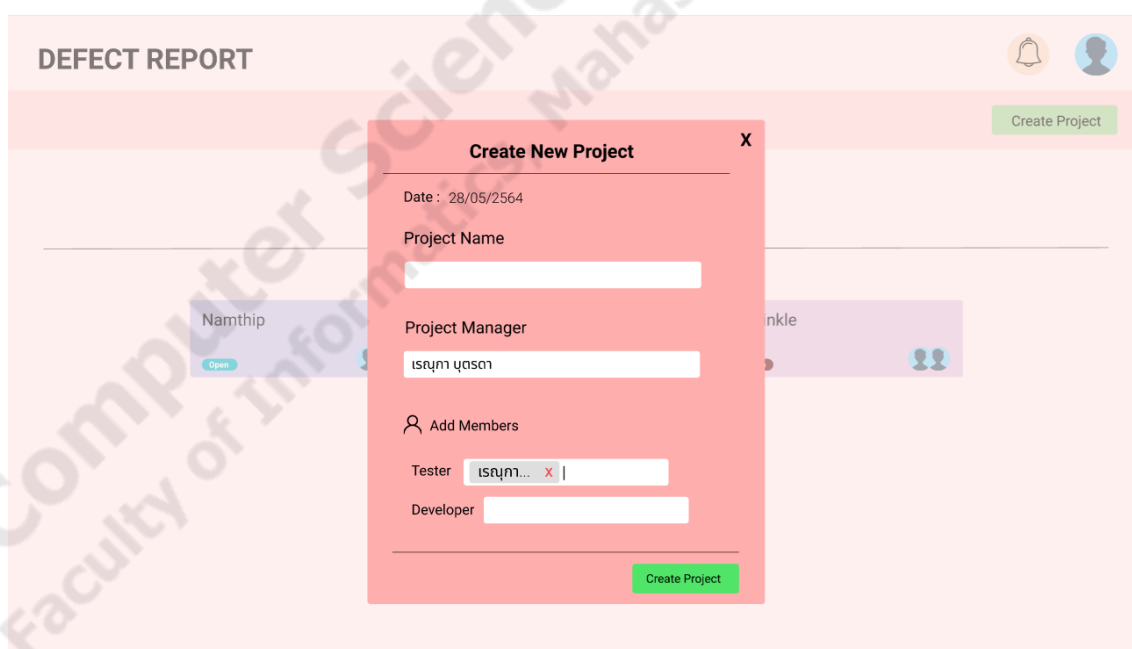
#### ตารางที่ 1.1 แผนการดำเนินงาน

กิจกรรม	เดือน		
	ส.ค	ก.ย	ต.ค
1.ศึกษาและรวบรวมข้อมูล			
2.วิเคราะห์และกำหนดขอบเขต			
3. ออกแบบระบบ			
4.พัฒนาโปรแกรม			
5. ทดสอบระบบ			
6. ทำรายงานสรุป			
7.นำเสนอโครงการ			

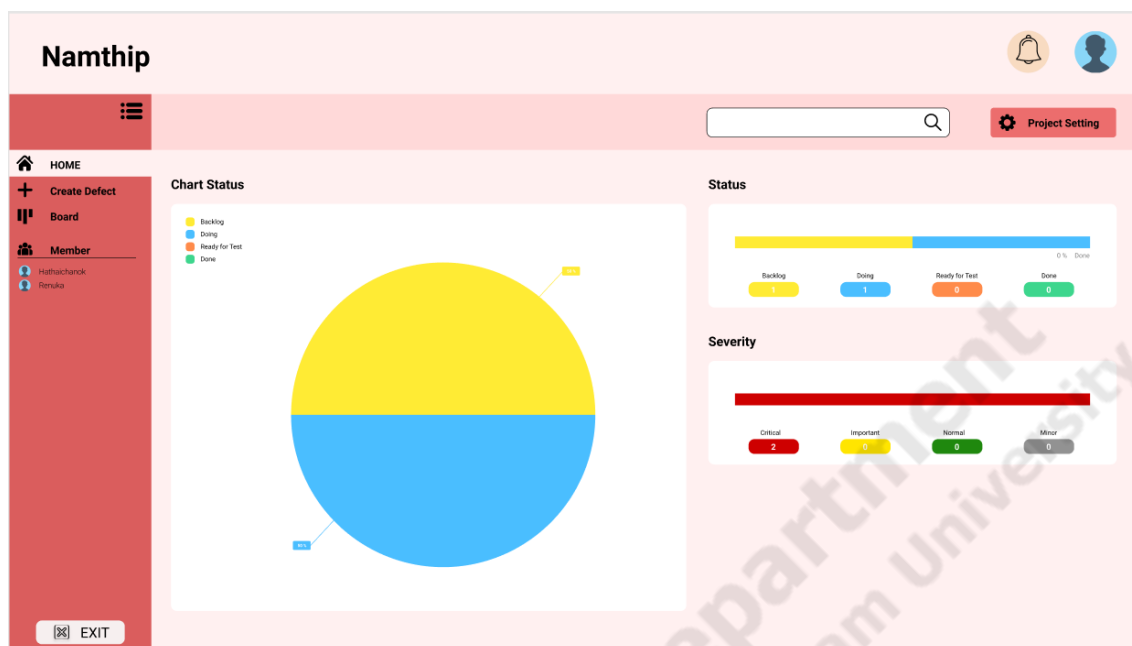
## 1.7 ตัวอย่างโปรแกรม



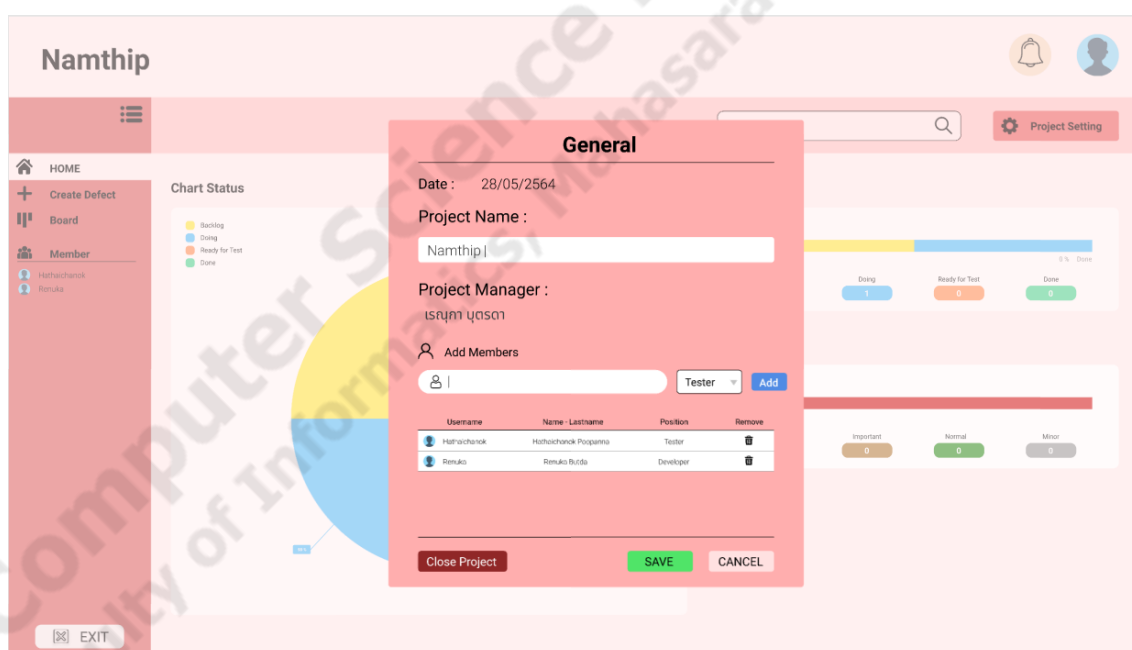
ภาพประกอบที่ 1.2 หน้าแรกหลังจากเข้าสู่ระบบ



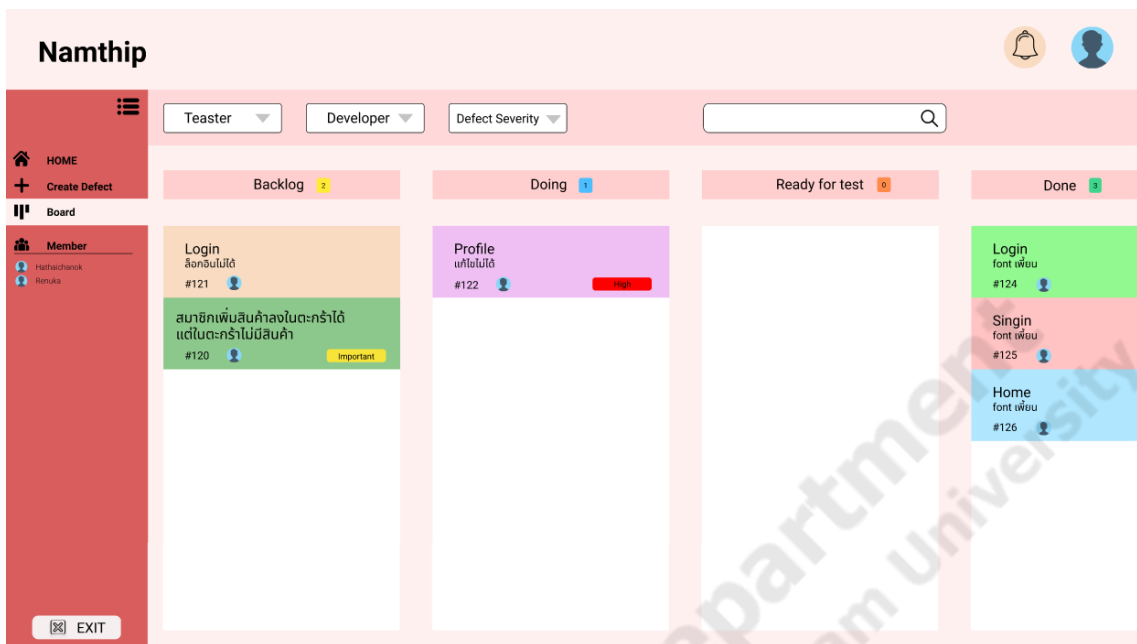
ภาพประกอบที่ 1.3 หน้าการสร้างโปรเจ็ค



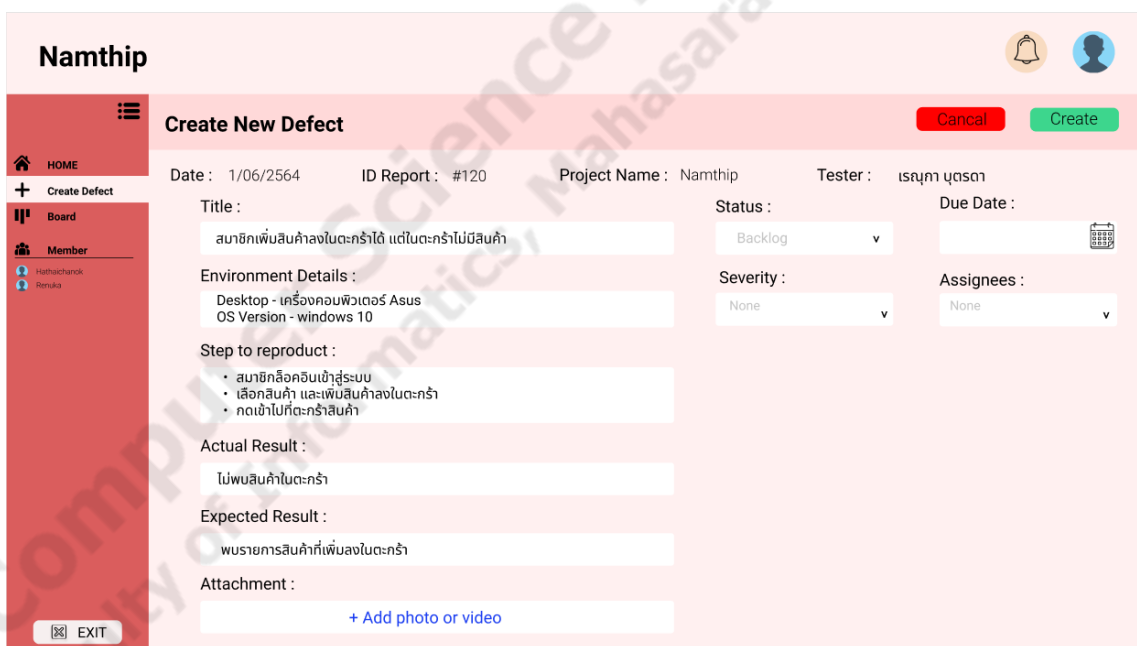
ภาพประกอบที่ 1.4 หน้า Home แสดงผลของโปรเจ็ค



ภาพประกอบที่ 1.5 หน้าการแก้ไขข้อมูลโปรเจ็ค ของProject Manager

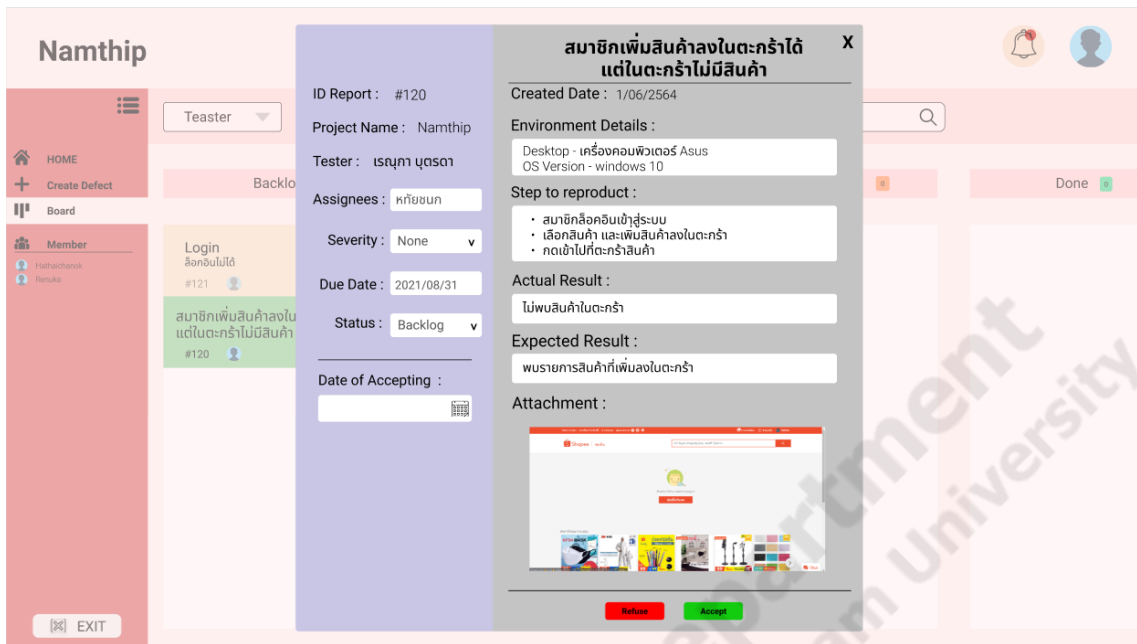


ภาพประกอบที่ 1.6 หน้าหลักการทำงาน

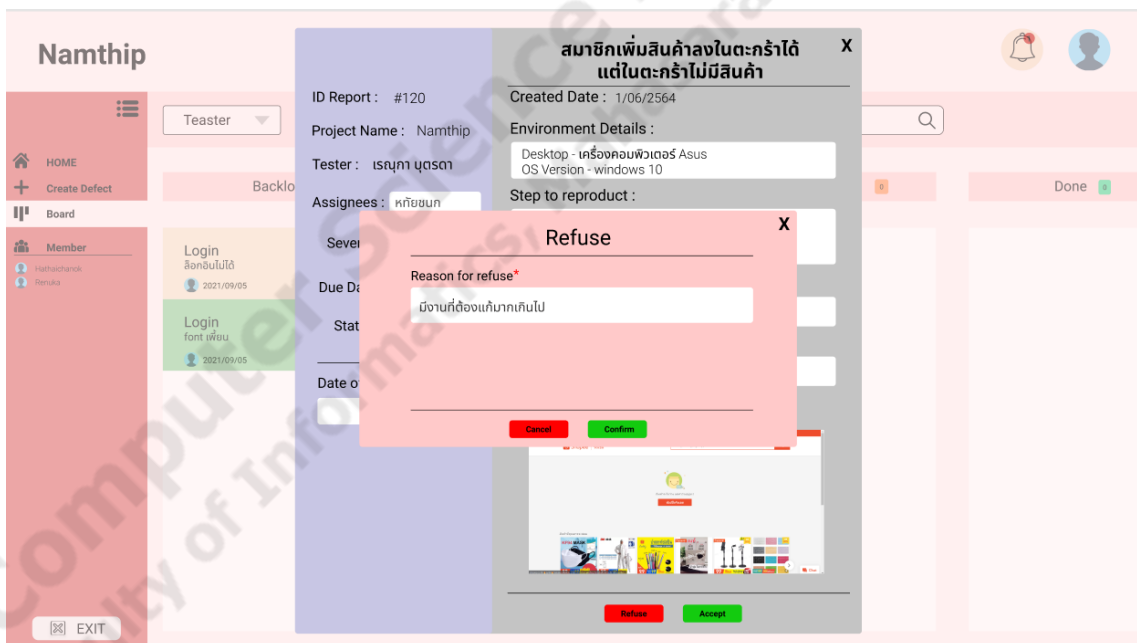


ภาพประกอบที่ 1.7 หน้าการสร้าง Defect

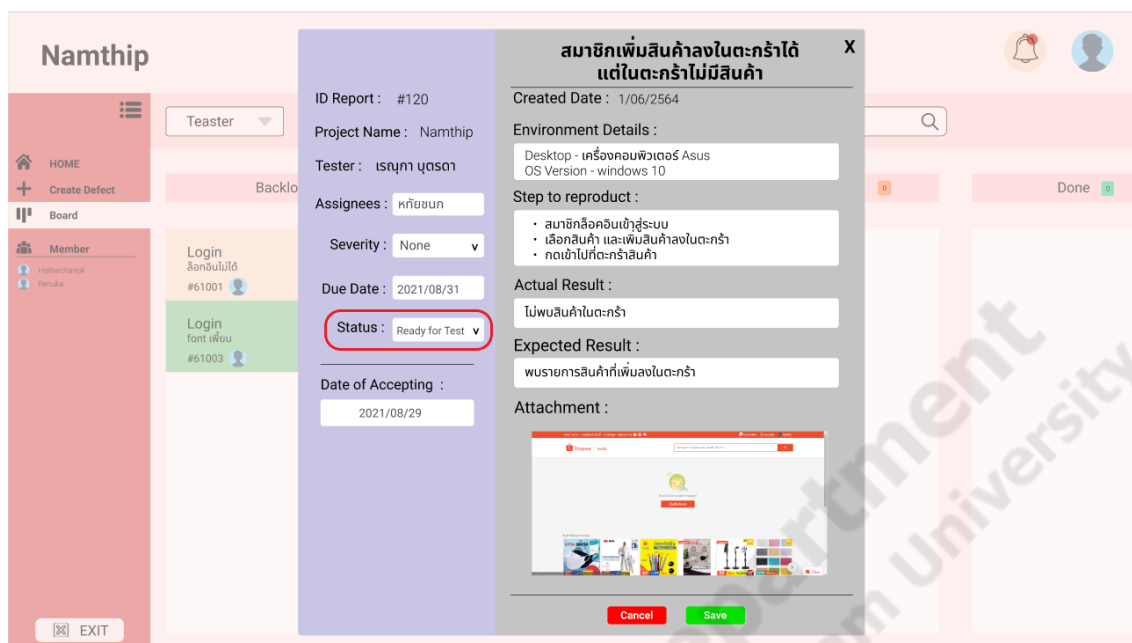




ภาพประกอบที่ 1.8 การแสดงการปฏิเสธ หรือยืนยันการรับงานของ Developer



ภาพประกอบที่ 1.9 การปฏิเสธการแก้ไข



ภาพประกอบที่ 1.10 การเปลี่ยน Status ของการทำงาน