

บทที่ 3

ขั้นตอนการดำเนินงาน

สำหรับในบทนี้จะกล่าวถึงขั้นตอนในการดำเนินงานของโครงการปริญญาโท ซึ่งจะทำให้ทราบถึงการวิเคราะห์ และการออกแบบเว็บไซต์ โดยขั้นตอนในการดำเนินมีรายละเอียดดังต่อไปนี้

3.1 แผนภาพการไหลของข้อมูล (Data Flow Diagram)

3.1.1 Context Diagram

3.1.2 Data Flow Diagram Level 1

3.1.3 Data Flow Diagram Level 2

3.2 พจนานุกรม (Data Dictionary)

3.2.1 External Entity Description

3.2.2 Data Flow Description and Data Structure

3.2.3 Data Store Description and Data Structure

3.2.4 Process Description

3.3 ความสัมพันธ์ (ER-Diagram)

3.4 การออกแบบฐานข้อมูล (Database Design)

3.5 Code ในส่วนของ Back-end

3.6 การออกแบบหน้าเว็บไซต์

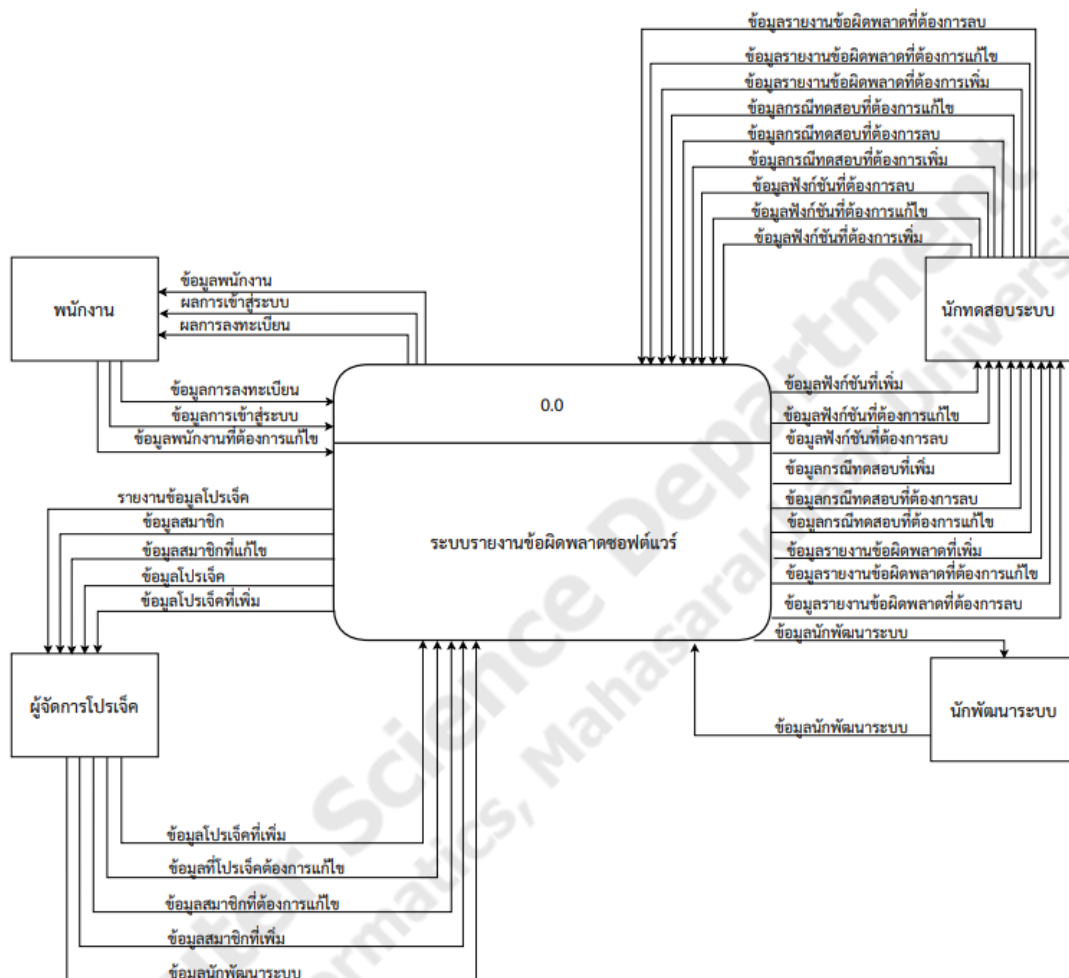
3.6.1 การออกแบบหน้าเว็บไซต์การเข้าสู่ระบบ และสมัครสมาชิก

3.6.2 การออกแบบหน้าเว็บไซต์ในการทำงานของผู้จัดการโปรเจ็ค (Project Manager)

3.6.3 การออกแบบหน้าเว็บไซต์ในการทำงานของนักทดสอบระบบ (Tester) และนักพัฒนา ระบบ (Developer)

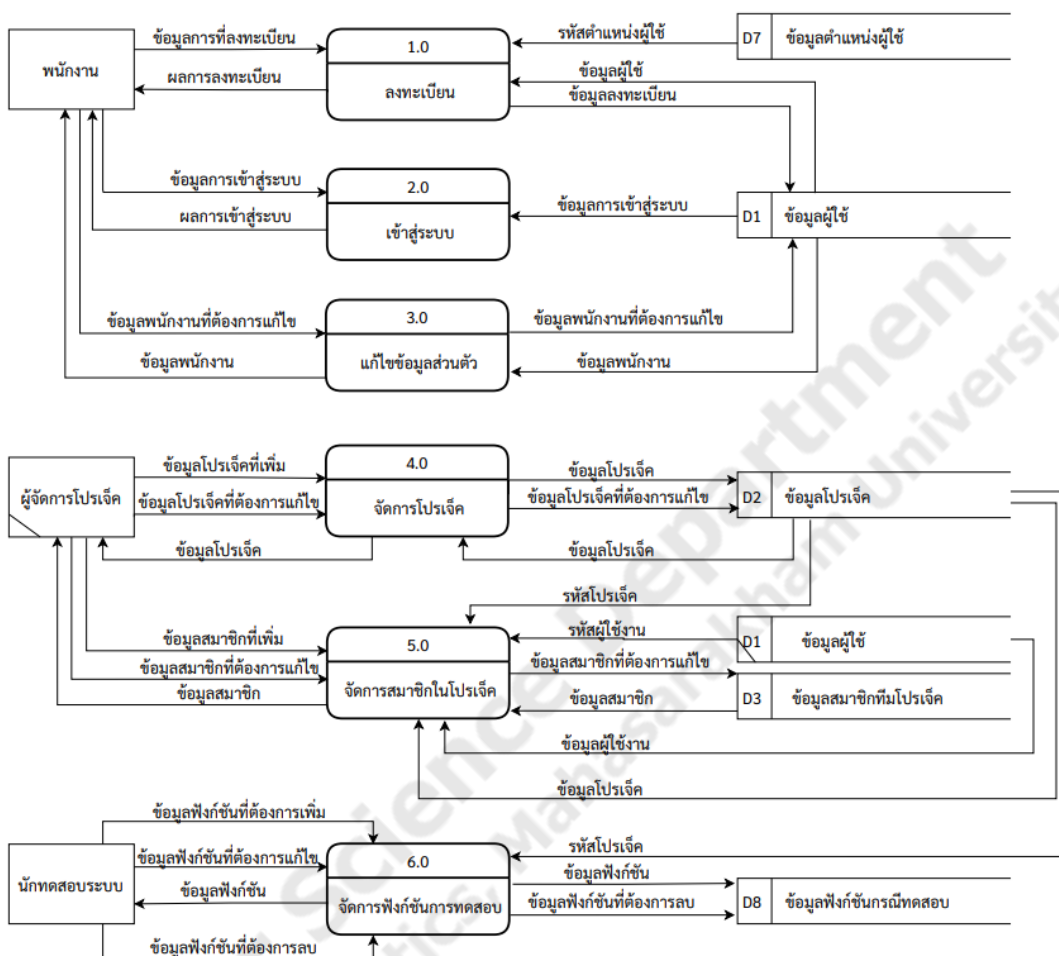
3.1 แผนภาพการไหลของข้อมูล (Data Flow Diagram)

3.1.1 Context Diagram



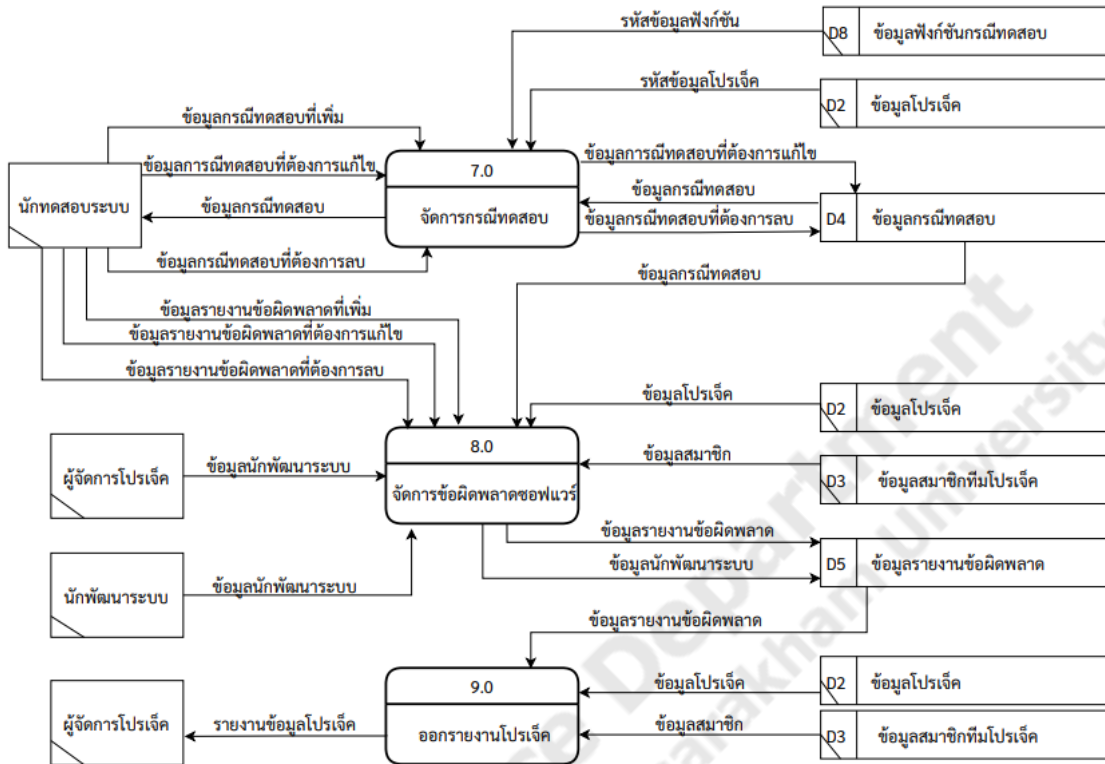
ภาพประกอบที่ 3.1 Context Diagram ของระบบรายงานข้อมูลบุคลากรซอฟต์แวร์

3.1.2 Data Flow Diagram Level 1



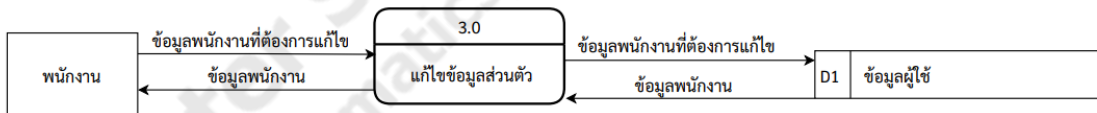
ภาพประกอบที่ 3.2 Data Flow Diagram Level 1

Computer Science Department
Faculty of Informatics Mahachulalongkornrajavidyalaya University

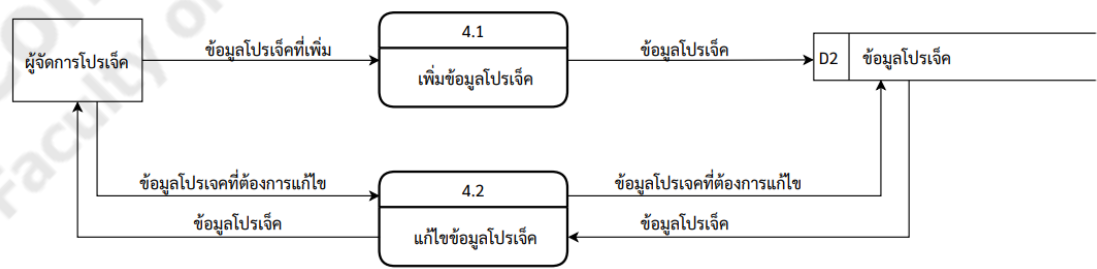


ภาพประกอบที่ 3.2 Data Flow Diagram Level 1 (ต่อ)

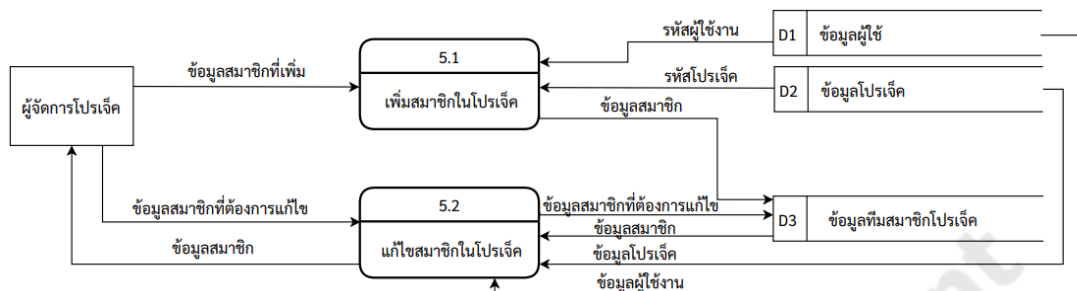
3.1.3 Data Flow Diagram Level 2



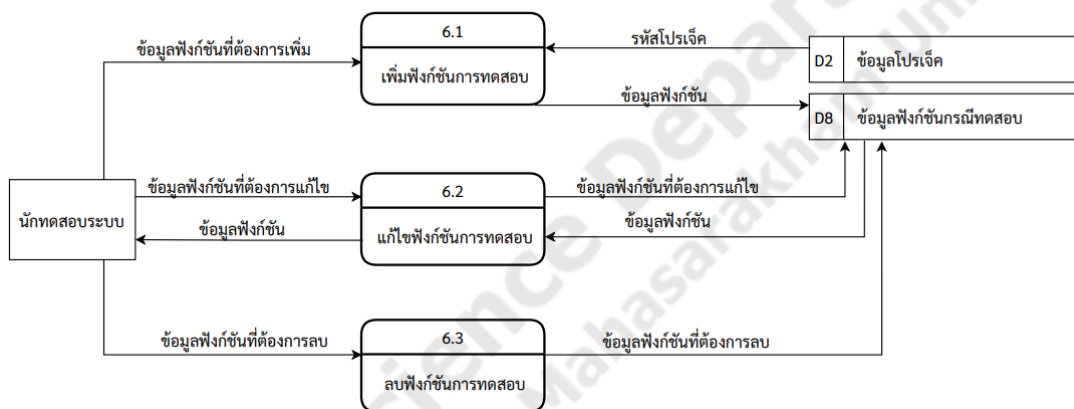
ภาพประกอบที่ 3.3 Data Flow Diagram Level 2 Process ที่ 3



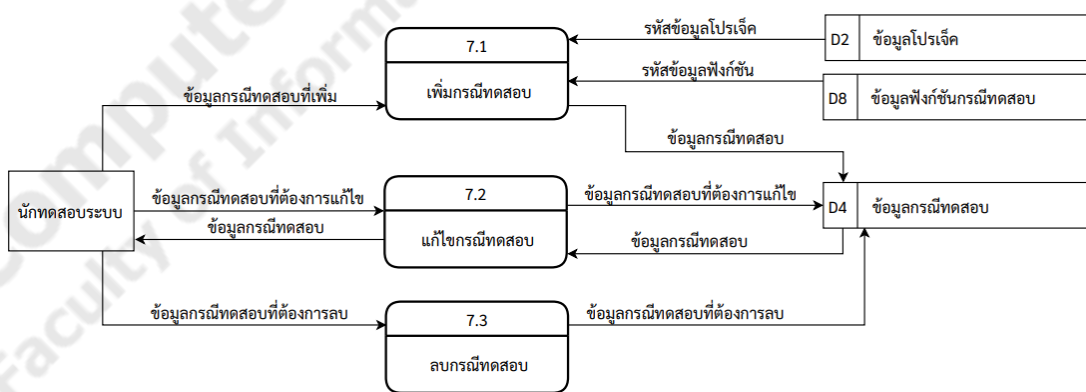
ภาพประกอบที่ 3.4 Data Flow Diagram Level 2 Process ที่ 4



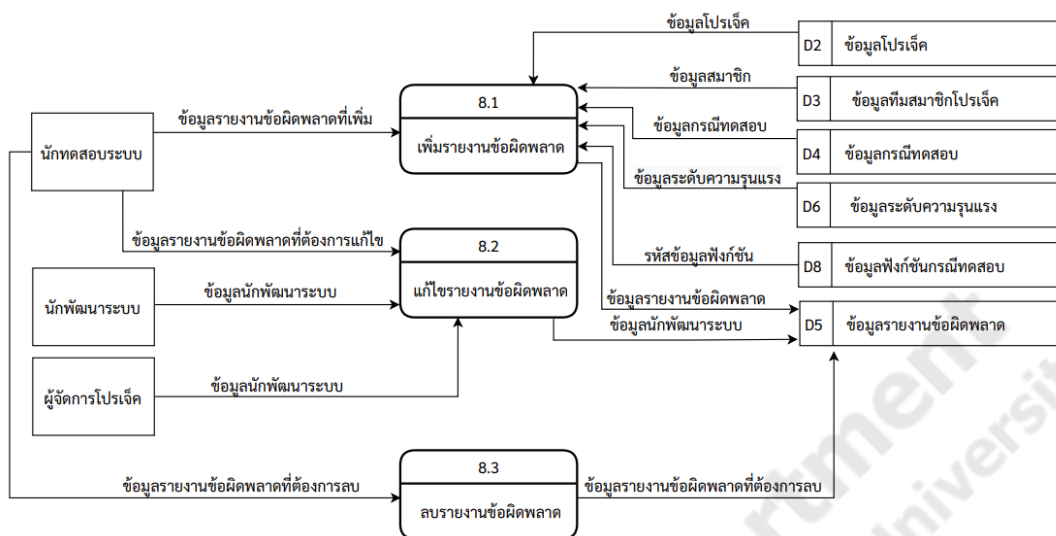
ภาพประกอบที่ 3.5 Data Flow Diagram Level 2 Process ที่ 5



ภาพประกอบที่ 3.6 Data Flow Diagram Level 2 Process ที่ 6



ภาพประกอบที่ 3.7 Data Flow Diagram Level 2 Process ที่ 7



ภาพประกอบที่ 3.8 Data Flow Diagram Level 2 Process ที่ 8

3.2 พจนานุกรม (Data Dictionary)

3.2.1 External Entity Description

ตารางที่ 3.1 External Entity Description

Name	Description	Input Data Flow	Output Data Flow
พนักงาน	- ลงทะเบียน	- ข้อมูลการที่ลงทะเบียน - ข้อมูลการเข้าสู่ระบบ - ข้อมูลพนักงานที่ต้องการแก้ไข	- ผลการลงทะเบียน - ผลการเข้าสู่ระบบ - ข้อมูลพนักงาน
ผู้จัดการโปรเจ็ค	- จัดการโปรเจ็ค - จัดการสมาชิกในโปรเจ็ค - จัดการข้อผิดพลาดซอฟต์แวร์ - รายงานโปรเจ็ค	- ข้อมูลโปรเจ็คที่เพิ่มและแก้ไข - ข้อมูลสมาชิกโปรเจ็คที่เพิ่ม และแก้ไข - ข้อมูลนักพัฒนาระบบ	- ข้อมูลโปรเจ็คที่เพิ่มและแก้ไข - ข้อมูลสมาชิกโปรเจ็คที่เพิ่ม และแก้ไข - ข้อมูลโปรเจ็ค - ข้อมูลสมาชิก - รายงานข้อมูลโปรเจ็ค

ตารางที่ 3.1 External Entity Description (ต่อ)

Name	Description	Input Data Flow	Output Data Flow
นักพัฒนาระบบ	- จัดการข้อผิดพลาดซอฟต์แวร์	- ข้อมูลนักพัฒนาระบบ	-
นักทดสอบระบบ	- จัดการกรณีทดสอบ - จัดการข้อมูลฟังก์ชันการทดสอบ - จัดการข้อผิดพลาดซอฟต์แวร์	- ข้อมูลฟังก์ชันที่ต้องการเพิ่ม - ข้อมูลฟังก์ชันที่ต้องการแก้ไข - ข้อมูลฟังก์ชันที่ต้องการลบ - ข้อมูลทดสอบที่ต้องการเพิ่ม - ข้อมูลกรณีทดสอบที่ต้องการแก้ไข - ข้อมูลทดสอบที่ต้องการลบ	- ข้อมูลฟังก์ชัน - ข้อมูลกรณีทดสอบ - ข้อมูลรายงานข้อผิดพลาด

3.2.2 Data Flow Description and Data Structure

ตารางที่ 3.2 Data Flow Description and Data Structure

Name	Description	Source	Destination	Data Structure
ข้อมูลพนักงานที่ต้องการแก้ไข	รายละเอียดข้อมูลพนักงานที่ต้องการแก้ไข	พนักงาน	Process 3.1 แก้ไขข้อมูลส่วนตัว	รหัสพนักงาน+ชื่อพนักงาน +นามสกุล+ชื่อผู้ใช้+
	รายละเอียดข้อมูลพนักงานที่ต้องการแก้ไข	Process 3.1 แก้ไขข้อมูลส่วนตัว	D1 ข้อมูลผู้ใช้	รหัสผ่าน +เบอร์โทรศัพท์+รูปโปรไฟล์ +รหัสตำแหน่ง+ชื่อตำแหน่ง

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลพนักงาน	รายละเอียด ข้อมูลพนักงาน	D1 ข้อมูลผู้ใช้	Process 3.1 แก้ไขข้อมูล ส่วนตัว	รหัสพนักงาน+ชื่อ พนักงาน +นามสกุล+ชื่อผู้ใช้+
	รายละเอียด ข้อมูลพนักงาน	Process 3.1 แก้ไขข้อมูล ส่วนตัว	พนักงาน	รหัสผ่าน +เบอร์โทรศัพท์+รูป โปรไฟล์ +รหัสตำแหน่ง+ชื่อ ตำแหน่ง
ข้อมูลโปรเจ็ค ที่เพิ่ม	รายละเอียด ข้อมูลโปรเจ็คที่ ต้องการเพิ่ม	ผู้จัดการโปรเจ็ค	Process 4.1 เพิ่ม ข้อมูลโปรเจ็ค	
ข้อมูลโปรเจ็ค	รายละเอียด ข้อมูลโปรเจ็ค	Process 4.1 เพิ่ม ข้อมูลโปรเจ็ค	D2 ข้อมูลโปรเจ็ค	
ข้อมูลโปรเจ็ค ที่ต้องการ แก้ไข	รายละเอียด ข้อมูลโปรเจ็คที่ ต้องการแก้ไข	ผู้จัดการโปรเจ็ค	Process 4.2 แก้ไขข้อมูลโปร เจ็ค	รหัสโปรเจ็ค+วันที่ สร้างโปรเจ็ค+ชื่อ โปรเจ็ค+
	รายละเอียด ข้อมูลโปรเจ็คที่ ต้องการแก้ไข	Process 4.2 แก้ไขข้อมูลโปร เจ็ค	D2 ข้อมูลโปรเจ็ค	รายละเอียดของโปร เจ็ค+รูปภาพ หน้าปก+สถานะของ โปรเจ็ค(เปิด, ปิด)+ วันที่ปิดโปรเจ็ค
ข้อมูลโปรเจ็ค	รายละเอียด ข้อมูลโปรเจ็ค	D2 ข้อมูลโปรเจ็ค	Process 4.2 แก้ไขข้อมูลโปร เจ็ค	
	รายละเอียด ข้อมูลโปรเจ็ค	Process 4.2 แก้ไขข้อมูลโปร เจ็ค	ผู้จัดการโปรเจ็ค	
ข้อมูลสมาชิกที่ เพิ่ม	รายละเอียด สมาชิกที่เพิ่มใน โปรเจ็ค	ผู้จัดการโปรเจ็ค	Process 5.1 เพิ่ม สมาชิกในโปรเจ็ค	รหัสโปรเจ็ค+รหัส พนักงาน+สถานะ (อยู่ในโปรเจ็ค, ไม่อยู่ ในโปรเจ็ค)

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลสมาชิก	รายละเอียด ข้อมูลสมาชิก	Process 5.1 เพิ่ม สมาชิกในโปรเจ็ค	D3 ข้อมูลทีม สมาชิกในโปรเจ็ค	รหัสโปรเจ็ค+รหัส พนักงาน+สถานะ (อยู่ในโปรเจ็ค, ไม่อยู่ ในโปรเจ็ค)
รหัสผู้ใช้งาน	รหัสของผู้ใช้งาน	D1 ข้อมูลผู้ใช้	Process 5.1 เพิ่ม สมาชิกในโปรเจ็ค	รหัสพนักงาน
รหัสโปรเจ็ค	รหัสของโปรเจ็ค	D2 ข้อมูลโปรเจ็ค	Process 5.1 เพิ่ม สมาชิกในโปรเจ็ค	รหัสโปรเจ็ค
ข้อมูลสมาชิกที่ ต้องการแก้ไข	รายละเอียด ข้อมูลสมาชิกที่ ต้องการแก้ไข	ผู้จัดการโปรเจ็ค	Process 5.2 แก้ไขสมาชิกใน โปรเจ็ค	รหัสโปรเจ็ค+รหัส พนักงาน+สถานะ (อยู่ในโปรเจ็ค, ไม่อยู่ ในโปรเจ็ค)
	รายละเอียด ข้อมูลสมาชิกที่ ต้องการแก้ไข	Process 5.2 แก้ไขสมาชิกใน โปรเจ็ค	D3 ข้อมูลทีม สมาชิกในโปรเจ็ค	
ข้อมูลผู้ใช้งาน	รายละเอียด ข้อมูลผู้ใช้งาน	D1 ข้อมูลผู้ใช้	Process 5.2 แก้ไขสมาชิกใน โปรเจ็ค	รหัสพนักงาน+ชื่อ พนักงาน +นามสกุล+ชื่อผู้ใช้+ รหัสผ่าน +เบอร์โทรศัพท์+รูป โปรไฟล์ +รหัสตำแหน่ง+ชื่อ ตำแหน่ง
ข้อมูลโปรเจ็ค	รายละเอียด ข้อมูลโปรเจ็ค	D2 ข้อมูลโปรเจ็ค	Process 5.2 แก้ไขสมาชิกใน โปรเจ็ค	รหัสโปรเจ็ค+วันที่ สร้างโปรเจ็ค+ชื่อ โปรเจ็ค+ รายละเอียดของโปร เจ็ค+รูปภาพ หน้าปก+สถานะของ โปรเจ็ค(เปิด, ปิด)+

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
				วันที่ปิดโปรเจ็ค
ข้อมูลสมาชิก	รายละเอียดข้อมูลสมาชิก	D3 ข้อมูลทีมสมาชิกในโปรเจ็ค	Process 5.2 แก้ไขสมาชิกในโปรเจ็ค	รหัสโปรเจ็ค+รหัสพนักงาน+สถานะ (อยู่ในโปรเจ็ค, ไม่อยู่ในโปรเจ็ค)
	รายละเอียดข้อมูลสมาชิก	Process 5.2 แก้ไขสมาชิกในโปรเจ็ค	ผู้จัดการโปรเจ็ค	
ข้อมูลฟังก์ชันที่ต้องการเพิ่ม	รายละเอียดข้อมูลฟังก์ชันที่ต้องการเพิ่ม	นักทดสอบระบบ	Process 6.1 เพิ่มฟังก์ชันการทดสอบ	รหัสฟังก์ชัน+ชื่อฟังก์ชัน+รหัสโปรเจ็ค
ข้อมูลฟังก์ชัน	รายละเอียดข้อมูลฟังก์ชัน	Process 6.1 เพิ่มฟังก์ชันการทดสอบ	D8 ข้อมูลฟังก์ชันกรณีทดสอบ	
รหัสโปรเจ็ค	รหัสโปรเจ็ค	D8 ข้อมูลฟังก์ชันกรณีทดสอบ	Process 6.1 เพิ่มฟังก์ชันการทดสอบ	รหัสโปรเจ็ค
ข้อมูลฟังก์ชันที่ต้องการแก้ไข	รายละเอียดข้อมูลฟังก์ชันที่ต้องการแก้ไข	นักทดสอบระบบ	Process 6.2 แก้ไขฟังก์ชันการทดสอบ	รหัสฟังก์ชัน+ชื่อฟังก์ชัน+รหัสโปรเจ็ค
	รายละเอียดข้อมูลฟังก์ชันที่ต้องการแก้ไข	Process 6.2 แก้ไขฟังก์ชันการทดสอบ	D8 ข้อมูลฟังก์ชันกรณีทดสอบ	
ข้อมูลฟังก์ชัน	รายละเอียดข้อมูลฟังก์ชัน	D8 ข้อมูลฟังก์ชันกรณีทดสอบ	Process 6.2 แก้ไขฟังก์ชันการทดสอบ	รหัสฟังก์ชัน+ชื่อฟังก์ชัน+รหัสโปรเจ็ค
	รายละเอียดข้อมูลฟังก์ชัน	Process 6.2 แก้ไขฟังก์ชันการทดสอบ	นักทดสอบระบบ	

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลฟังก์ชันที่ต้องการลบ	รายละเอียดข้อมูลฟังก์ชันที่ต้องการลบ	นักทดสอบระบบ	Process 6.3 แก้ไขฟังก์ชันการลบ	รหัสฟังก์ชัน+ชื่อฟังก์ชัน+รหัสโปรเจ็ค
	รายละเอียดข้อมูลฟังก์ชันที่ต้องการลบ	Process 6.3 แก้ไขฟังก์ชันการลบ	D8 ข้อมูลฟังก์ชันกรณีทดสอบ	
ข้อมูลกรณีทดสอบที่เพิ่ม	รายละเอียดข้อมูลกรณีทดสอบที่ต้องการเพิ่ม	นักทดสอบระบบ	Process 7.1 เพิ่มกรณีทดสอบ	รหัสของกรณีทดสอบ+ชื่อกรณีทดสอบ+ข้อมูลที่ทดสอบ
ข้อมูลกรณีทดสอบ	รายละเอียดข้อมูลกรณีทดสอบ	Process 7.1 เพิ่มกรณีทดสอบ	D4 ข้อมูลกรณีทดสอบ	+ผลลัพธ์ที่คาดหวัง+สถานะ +หมายเหตุ+รหัสโปรเจ็ค +รหัสของโปรเจ็ค+รหัสฟังก์ชัน
รหัสข้อมูลโปรเจ็ค	รหัสข้อมูลโปรเจ็ค	D2 ข้อมูลโปรเจ็ค	Process 7.1 เพิ่มกรณีทดสอบ	รหัสข้อมูลโปรเจ็ค
รหัสข้อมูลฟังก์ชัน	รหัสข้อมูลฟังก์ชัน	D8 ข้อมูลฟังก์ชันกรณีทดสอบ	Process 7.1 เพิ่มกรณีทดสอบ	รหัสข้อมูลฟังก์ชัน
ข้อมูลกรณีทดสอบที่ต้องการแก้ไข	รายละเอียดข้อมูลกรณีทดสอบที่ต้องการแก้ไข	นักทดสอบระบบ	Process 7.2 แก้ไขกรณีทดสอบ	รหัสของกรณีทดสอบ+ชื่อกรณีทดสอบ+ข้อมูลที่ทดสอบ+ผลลัพธ์ที่คาดหวัง+สถานะ+หมายเหตุ+รหัสโปรเจ็ค+รหัสของโปรเจ็ค+รหัสฟังก์ชัน
	รายละเอียดข้อมูลกรณีทดสอบที่ต้องการแก้ไข	Process 7.2 แก้ไขกรณีทดสอบ	D4 ข้อมูลกรณีทดสอบ	คาดหวัง+สถานะ+หมายเหตุ+รหัสโปรเจ็ค+รหัสของโปรเจ็ค+รหัสฟังก์ชัน

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
ข้อมูลกรณีทดสอบ	รายละเอียดข้อมูลกรณีทดสอบ	D4 ข้อมูลกรณีทดสอบ	Process 7.2 แกะไขกรณีทดสอบ	รหัสของกรณีทดสอบ+ชื่อกรณีทดสอบ+ข้อมูลที่ทดสอบ
	รายละเอียดข้อมูลกรณีทดสอบ	Process 7.2 แกะไขกรณีทดสอบ	นักทดสอบระบบ	
ข้อมูลกรณีทดสอบที่ต้องการลบ	รายละเอียดข้อมูลกรณีทดสอบที่ต้องการลบ	นักทดสอบระบบ	Process 7.3 ลบกรณีทดสอบ	+ผลลัพธ์ที่คาดหวัง+สถานะ+หมายเหตุ+รหัสโปรเจ็ค
	รายละเอียดข้อมูลกรณีทดสอบที่ต้องการลบ	Process 7.3 ลบกรณีทดสอบ	D4 ข้อมูลกรณีทดสอบ	+รหัสของโปรเจ็ค+รหัสฟังก์ชัน
ข้อมูลรายงานข้อผิดพลาดที่เพิ่ม	รายละเอียดข้อมูลรายงานข้อผิดพลาดที่ต้องการเพิ่ม	นักทดสอบระบบ	Process 8.1 เพิ่มรายงานข้อผิดพลาด	รหัสรายงานข้อผิดพลาด+ชื่อหัวข้อของรายงาน+สภาพแวดล้อมของระบบ+ ผลลัพธ์ที่เกิดขึ้น+วันที่ครบกำหนด+วันที่สร้าง+สถานะ+ความคิดเห็น+สื่อที่แนบมา+รอบรายงานข้อผิดพลาด+รหัสความรุนแรง+รหัสสมาชิก+รหัสโปรเจ็ค+รหัสของกรณีทดสอบ
ข้อมูลรายงานข้อผิดพลาด	รายละเอียดข้อมูลรายงานข้อผิดพลาด	Process 8.1 เพิ่มรายงานข้อผิดพลาด	D5 ข้อมูลรายงานข้อผิดพลาด	

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
				+รหัสสำนักทดสอบ ระบบ+รหัส นักพัฒนาระบบ+ รหัสผู้จัดการโปรเจ็ค
ข้อมูลโปรเจ็ค	รายละเอียด ข้อมูลโปรเจ็ค	D2 ข้อมูลโปรเจ็ค	Process 8.1 เพิ่ม รายงาน ข้อผิดพลาด	รหัสโปรเจ็ค+วันที่ สร้างโปรเจ็ค+ชื่อ โปรเจ็ค+ รายละเอียดของโปร เจ็ค+รูปภาพ หน้าปก+สถานะของ โปรเจ็ค(เปิด, ปิด)+ วันที่ปิดโปรเจ็ค
ข้อมูลสมาชิก	รายละเอียด ข้อมูลสมาชิก	D3 ข้อมูลที่ม สมาชิกในโปรเจ็ค	Process 8.1 เพิ่ม รายงาน ข้อผิดพลาด	รหัสโปรเจ็ค+รหัส พนักงาน+สถานะ (อยู่ในโปรเจ็ค, ไม่อยู่ ในโปรเจ็ค)
ข้อมูลกรณี ทดสอบ	รายละเอียด ข้อมูลกรณี ทดสอบ	D4 ข้อมูลกรณี ทดสอบ	Process 8.1 เพิ่ม รายงาน ข้อผิดพลาด	รหัสของกรณี ทดสอบ+ชื่อกรณี ทดสอบ+ข้อมูลที่ ทดสอบ +ผลลัพธ์ที่คาดหวัง+ สถานะ +หมายเหตุ+รหัส โปรเจ็ค +รหัสของโปรเจ็ค+ รหัสฟังก์ชัน
ข้อมูลระดับ ความรุนแรง	รายละเอียด ข้อมูลระดับความ รุนแรง	D6 ข้อมูลระดับ ความรุนแรง	Process 8.1 เพิ่ม รายงาน ข้อผิดพลาด	รหัสความรุนแรง+ ชื่อความรุนแรง

ตารางที่ 3.2 Data Flow Description and Data Structure (ต่อ)

Name	Description	Source	Destination	Data Structure
รหัสข้อมูล ฟังก์ชัน	รหัสข้อมูลฟังก์ชัน	D8 ข้อมูลฟังก์ชัน กรณีทดสอบ	Process 8.1 เพิ่ม รายงาน ข้อผิดพลาด	รหัสฟังก์ชัน+ชื่อ ฟังก์ชัน+รหัสโปร เจ็ค
ข้อมูลรายงาน ข้อผิดพลาดที่ ต้องการแก้ไข	รายละเอียด ข้อมูลรายงาน ข้อผิดพลาดที่ ต้องการแก้ไข	นักทดสอบระบบ	Process 8.2 แก้ไขรายงาน ข้อผิดพลาด	รหัสรายงาน ข้อผิดพลาด+ชื่อ หัวข้อของรายงาน+ สภาพแวดล้อมของ
ข้อมูลรายงาน ข้อผิดพลาด	รายละเอียด ข้อมูลรายงาน ข้อผิดพลาด	นักพัฒนาระบบ	Process 8.2 แก้ไขรายงาน ข้อผิดพลาด	ระบบ+ ผลลัพธ์ที่ เกิดขึ้น+วันที่ครบ กำหนด+วันที่สร้าง+
		ผู้จัดการโปรเจ็ค	Process 8.2 แก้ไขรายงาน ข้อผิดพลาด	สถานะ+ความ คิดเห็น+สื่อที่แนบ มา+รอบรายงาน
		Process 8.2 แก้ไขรายงาน ข้อผิดพลาด	D5 ข้อมูลรายงาน ข้อผิดพลาด	ข้อผิดพลาด+รหัส ความรุนแรง+รหัส สมาชิก+รหัสโปร เจ็ค
ข้อมูลรายงาน ข้อผิดพลาดที่ ต้องการลบ	รายละเอียด ข้อมูลรายงาน ข้อผิดพลาดที่ ต้องการลบ	นักพัฒนาระบบ	Process 8.3 ลบ รายงาน ข้อผิดพลาด	เจ็ค+รหัสของกรณี ทดสอบ+รหัสนัก ทดสอบระบบ+รหัส นักพัฒนาระบบ+
	รายละเอียด ข้อมูลรายงาน ข้อผิดพลาดที่ ต้องการลบ	Process 8.3 ลบ รายงาน ข้อผิดพลาด	D5 ข้อมูลรายงาน ข้อผิดพลาด	รหัสผู้จัดการโปรเจ็ค

3.2.3 Data Store Description and Data Structure

ตารางที่ 3.3 Data Store Description and Data Structure

ID	Data Store Name	Description	Data Structure
D1	ข้อมูลผู้ใช้	เป็นแฟ้มข้อมูลที่เก็บข้อมูล	รหัสพนักงาน+ชื่อพนักงาน

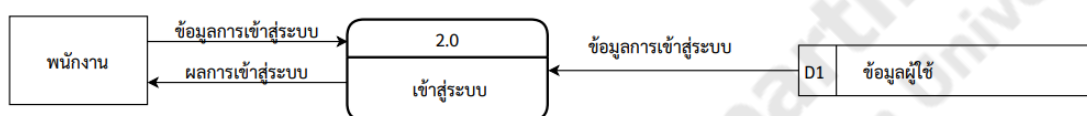
ตารางที่ 3.3 Data Store Description and Data Structure (ต่อ)

ID	Data Store Name	Description	Data Structure
		ของผู้ใช้	+นามสกุล+ชื่อผู้ใช้+รหัสผ่าน +เบอร์โทรศัพท์+รูปโปรไฟล์ +รหัสตำแหน่ง+ชื่อตำแหน่ง
D2	ข้อมูลโปรเจ็ค	เป็นแฟ้มข้อมูลที่เก็บข้อมูล ของโปรเจ็ค	รหัสโปรเจ็ค+วันที่สร้างโปรเจ็ค+ ชื่อโปรเจ็ค+รายละเอียดของโปร เจ็ค+รูปภาพหน้าปก+สถานะของ โปรเจ็ค(เปิด, ปิด)+ วันที่ปิดโปร เจ็ค
D3	ข้อมูลสมาชิกทีมโปรเจ็ค	เป็นแฟ้มข้อมูลที่เก็บข้อมูล ของสมาชิกทีมโปรเจ็ค	รหัสโปรเจ็ค+รหัสพนักงาน+ สถานะ(อยู่ในโปรเจ็ค, ไม่อยู่ในโปร เจ็ค)
D4	ข้อมูลกรณีทดสอบ	เป็นแฟ้มข้อมูลที่เก็บข้อมูล ของกรณีทดสอบ	รหัสของกรณีทดสอบ+ชื่อกรณี ทดสอบ+ข้อมูลที่ทดสอบ +ผลลัพธ์ที่คาดหวัง+สถานะ +หมายเหตุ+รหัสโปรเจ็ค +รหัสของโปรเจ็ค+รหัสฟังก์ชัน
D5	ข้อมูลรายงาน ข้อผิดพลาด	เป็นแฟ้มข้อมูลที่เก็บข้อมูล ของรายงานข้อผิดพลาด	รหัสรายงานข้อผิดพลาด+ชื่อหัวข้อ ของรายงาน+สภาพแวดล้อมของ ระบบ+ ผลลัพธ์ที่เกิดขึ้น+วันที่ ครบกำหนด+วันที่สร้าง+สถานะ+ ความคิดเห็น+สื่อที่แนบมา+รอบ รายงานข้อผิดพลาด+รหัสความ รุนแรง+รหัสสมาชิก+รหัสโปร เจ็ค+รหัสของกรณีทดสอบ+รหัส นักทดสอบระบบ+รหัสสำนักพัฒนา ระบบ+รหัสผู้จัดการโปรเจ็ค
D6	ข้อมูลระดับความรุนแรง	เป็นระดับความรุนแรงของ รายงานข้อผิดพลาด	รหัสความรุนแรง+ชื่อความรุนแรง

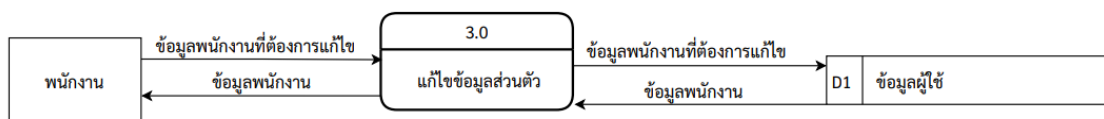
ตารางที่ 3.3 Data Store Description and Data Structure (ต่อ)

ID	Data Store Name	Description	Data Structure
D7	ข้อมูลตำแหน่งผู้ใช้	เป็นข้อมูลของตำแหน่ง พนักงาน	รหัสตำแหน่ง+ชื่อตำแหน่ง
D8	ข้อมูลฟังก์ชันกรณี ทดสอบ	เป็นข้อมูลของฟังก์ชันกรณี ทดสอบ	รหัสฟังก์ชัน+ชื่อฟังก์ชัน+รหัสโปร เจ็ค

3.2.4 Process Description



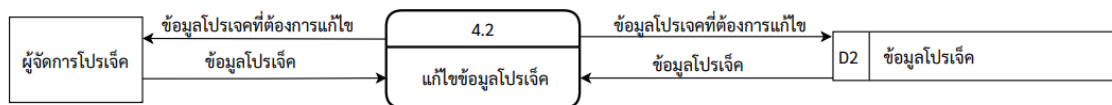
ID	2.0
Name	การเข้าสู่ระบบ
Description	การเข้าสู่ระบบรายงานข้อผิดพลาดซอฟต์แวร์
Input Data Flows	- ข้อมูลการเข้าสู่ระบบ
Output Data Flows	- ผลการเข้าสู่ระบบ
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> รับข้อมูลที่เป็น Username และ Password ในการเข้าสู่ระบบ ตรวจสอบว่าข้อมูล Username และ Password มีสิทธิ์ในการเข้าสู่ระบบหรือไม่ <ul style="list-style-type: none"> ถ้า (มีข้อมูล) <ul style="list-style-type: none"> - ให้สิทธิ์ในการเข้าสู่ระบบ ถ้า (ไม่มีข้อมูล) <ul style="list-style-type: none"> - แสดงข้อความไม่สามารถเข้าสู่ระบบได้ <p>จบการทำงาน</p>



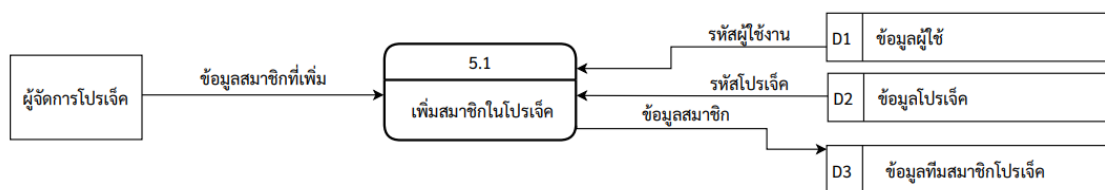
ID	3.0
Name	แก้ไขข้อมูลส่วนตัว
Description	การแก้ไขข้อมูลส่วนตัว
Input Data Flows	- ข้อมูลพนักงานที่ต้องการแก้ไข
Output Data Flows	- ข้อมูลพนักงาน
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลส่วนตัวที่ต้องการแก้ไข 2. ตรวจสอบว่ามีข้อมูลครบถ้วนหรือไม่ ถ้า (มีข้อมูล) <ul style="list-style-type: none"> - ดึงข้อมูลผู้ใช้งานระบบ - ตรวจสอบข้อมูลครบถ้วนถูกต้อง - บันทึกลงฐานข้อมูล ถ้า (ไม่มีข้อมูล) <ul style="list-style-type: none"> - แสดงข้อความว่าไม่ข้อมูล <p>จบการทำงาน</p>



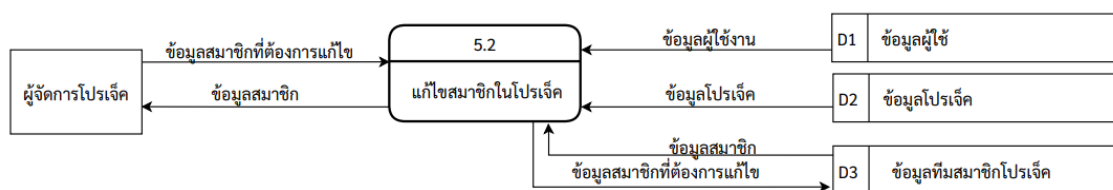
ID	4.1
Name	เพิ่มข้อมูลโปรเจ็ค
Description	การเพิ่มข้อมูลโปรเจ็ค
Input Data Flows	- ข้อมูลโปรเจ็ค
Output Data Flows	-
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลของโปรเจ็ค 2. ตรวจสอบว่ามีข้อมูลถูกต้องหรือไม่ <ul style="list-style-type: none"> ถ้า (ข้อมูลโปรเจ็คครบถ้วนและถูกต้อง) <ul style="list-style-type: none"> 2.1 ตรวจสอบข้อมูลจากฐานข้อมูลโปรเจ็ค ถ้า (มีข้อมูลโปรเจ็ค) <ul style="list-style-type: none"> - แสดงข้อความว่าไม่สามารถเพิ่มโปรเจ็คได้ ถ้า (ไม่มีข้อมูลโปรเจ็ค) <ul style="list-style-type: none"> - เพิ่มข้อมูลโปรเจ็ค - บันทึกลงฐานข้อมูล ถ้า (ข้อมูลไม่ครบถ้วนและไม่ถูกต้อง) <ul style="list-style-type: none"> - แสดงข้อความว่าข้อมูลไม่ถูกต้อง <p>จบการทำงาน</p>



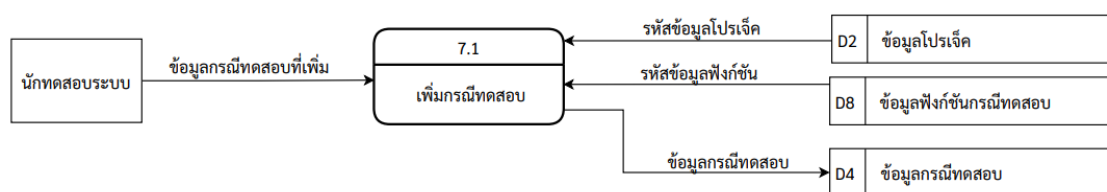
ID	4.2
Name	แก้ไขข้อมูลโปรเจ็ค
Description	การแก้ไขข้อมูลสถานะโปรเจ็ค
Input Data Flows	- ข้อมูลโปรเจ็คที่ต้องการแก้ไข
Output Data Flows	- ข้อมูลโปรเจ็ค
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลของโปรเจ็คที่ต้องการแก้ไข 2. ตรวจสอบว่ามีข้อมูลหรือไม่ <p>ถ้า (มีข้อมูล)</p> <ul style="list-style-type: none"> - ดึงฐานข้อมูลจากข้อมูลโปรเจ็ค - รับข้อมูลที่ต้องการแก้ไข - บันทึกลงฐานข้อมูล <p>ถ้า (ไม่มีข้อมูล)</p> <ul style="list-style-type: none"> - แสดงข้อความว่าไม่พบข้อมูล <p>จบการทำงาน</p>



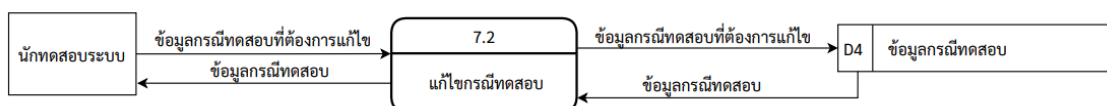
ID	5.1
Name	เพิ่มสมาชิกในโปรเจ็ค
Description	การเพิ่มสมาชิกในโปรเจ็ค
Input Data Flows	- ข้อมูลสมาชิกที่เพิ่ม
Output Data Flows	-
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. เพิ่มข้อมูลสมาชิกในโปรเจ็ค 2. ตรวจสอบว่ามีข้อมูลผู้ใช้หรือไม่ <ol style="list-style-type: none"> 3.1 ตรวจสอบข้อมูลจากฐานข้อมูลผู้ใช้และข้อมูลโปรเจ็ค <p>ถ้า (มีข้อมูล)</p> <ul style="list-style-type: none"> - เพิ่มสมาชิกในโปรเจ็ค - บันทึกลงฐานข้อมูล <p>ถ้า (ไม่มีมีข้อมูล)</p> <ul style="list-style-type: none"> - แสดงข้อความว่าไม่พบข้อมูล <p>จบการทำงาน</p>



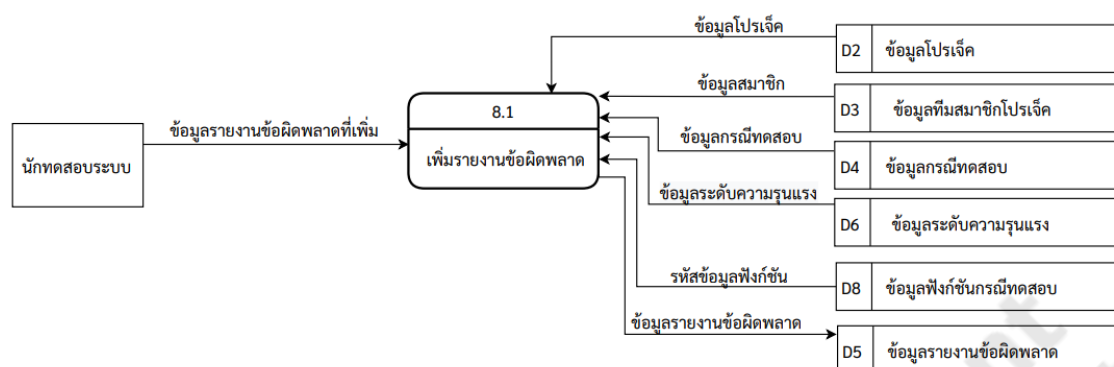
ID	5.2
Name	แก้ไขสมาชิกในโปรเจ็ค
Description	การแก้ไขสมาชิกในโปรเจ็ค
Input Data Flows	- ข้อมูลสมาชิกที่ต้องการแก้ไข
Output Data Flows	- ข้อมูลสมาชิก
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลสมาชิกที่ต้องการแก้ไข 2. ตรวจสอบว่ามีข้อมูลสมาชิกหรือไม่ ถ้า (มีข้อมูล) <ul style="list-style-type: none"> - ดึงฐานข้อมูลจากข้อมูลสมาชิกทีมโปรเจ็ค - รับข้อมูลที่ต้องการแก้ไข - บันทึกลงฐานข้อมูล ถ้า (ไม่มีข้อมูล) <ul style="list-style-type: none"> - แสดงข้อความว่าไม่พบข้อมูล <p>จบการทำงาน</p>



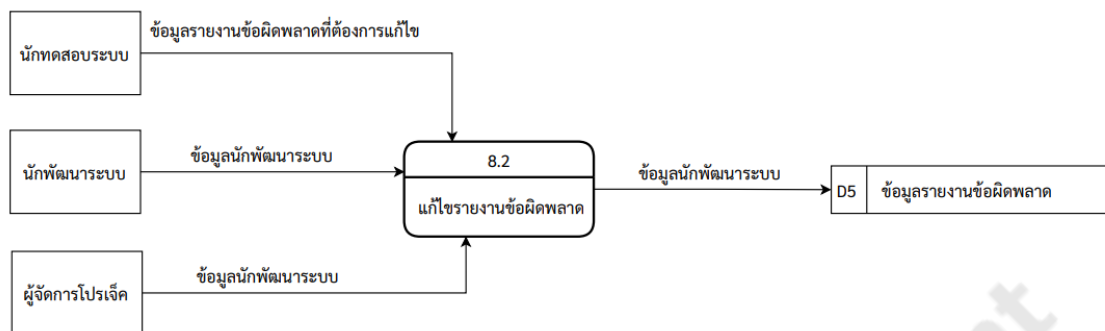
ID	7.1
Name	เพิ่มกรณีทดสอบ
Description	การเพิ่มข้อมูลกรณีทดสอบ
Input Data Flows	- ข้อมูลกรณีทดสอบที่เพิ่ม
Output Data Flows	-
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลกรณีทดสอบที่ต้องการเพิ่ม 2. ตรวจสอบว่ามีข้อมูลถูกต้องหรือไม่ ถ้า (ข้อมูลครบถ้วนและถูกต้อง) <ol style="list-style-type: none"> 2.1 ตรวจสอบข้อมูลจากฐานข้อมูลโปรเจ็ค ถ้า (มีข้อมูล) <ul style="list-style-type: none"> - แสดงข้อความว่าไม่สามารถเพิ่มข้อมูลกรณีการทดสอบได้ ถ้า (ไม่มีข้อมูล) <ul style="list-style-type: none"> - เพิ่มข้อมูลกรณีทดสอบ - บันทึกลงฐานข้อมูล ถ้า (ข้อมูลไม่ครบถ้วนและไม่ถูกต้อง) <ul style="list-style-type: none"> - แสดงข้อความว่าข้อมูลไม่ถูกต้อง <p>จบการทำงาน</p>



ID	7.2
Name	แก้ไขกรณีทดสอบ
Description	การแก้ไขกรณีทดสอบ
Input Data Flows	- ข้อมูลกรณีทดสอบที่ต้องการแก้ไข
Output Data Flows	- ข้อมูลกรณีทดสอบ
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลกรณีทดสอบที่ต้องการแก้ไข 2. ตรวจสอบว่ามีข้อมูลหรือไม่ <p>ถ้า (มีข้อมูล)</p> <ul style="list-style-type: none"> - แก้ไขข้อมูลที่ต้องการแก้ไข - บันทึกลงฐานข้อมูล <p>ถ้า (ไม่มีข้อมูล)</p> <ul style="list-style-type: none"> - แสดงข้อความว่าไม่พบข้อมูล <p>จบการทำงาน</p>

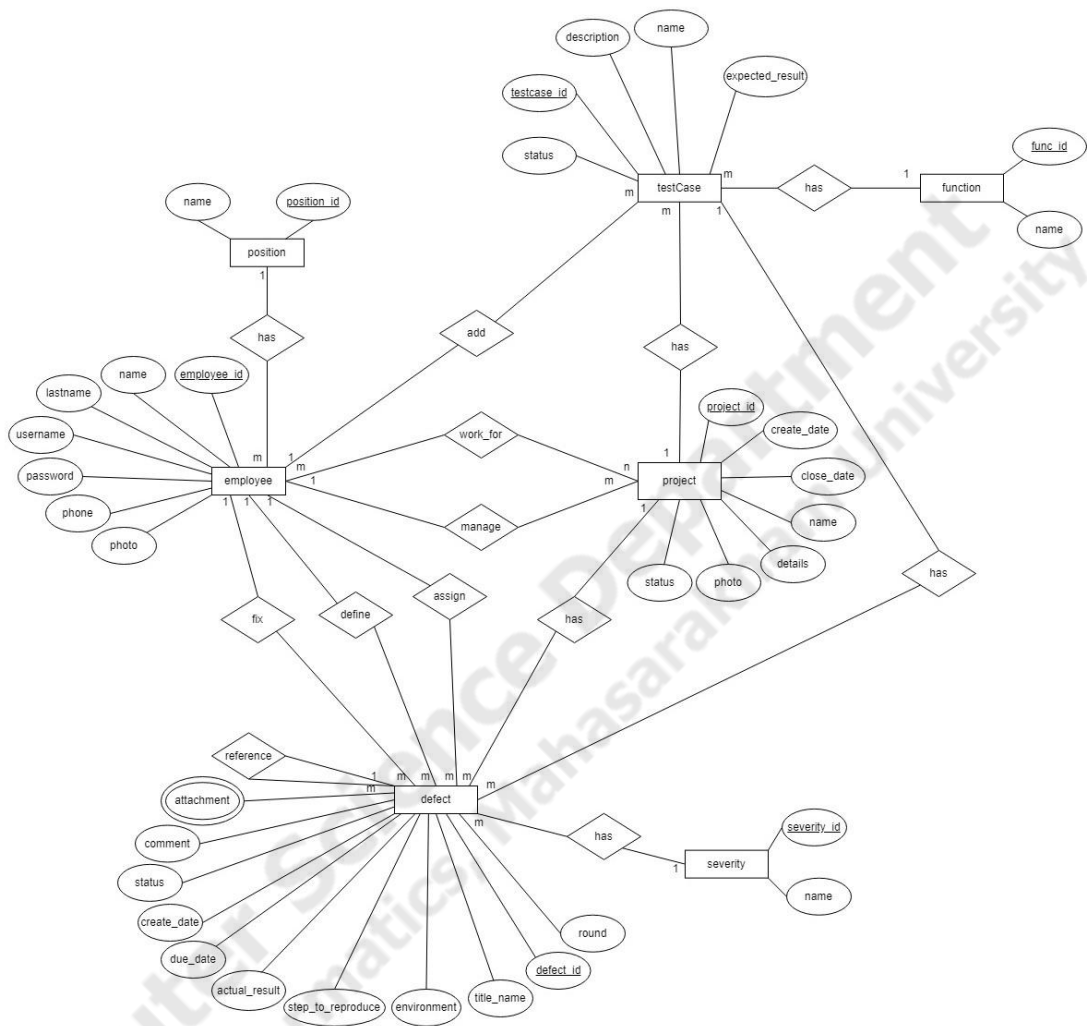


ID	8.1
Name	เพิ่มรายงานข้อผิดพลาด
Description	การเพิ่มรายงานข้อผิดพลาด
Input Data Flows	- ข้อมูลรายงานข้อผิดพลาดที่เพิ่ม
Output Data Flows	-
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1. รับข้อมูลรายงานข้อผิดพลาด 2. ตรวจสอบว่ามีข้อมูลถูกต้องหรือไม่ <ul style="list-style-type: none"> ถ้า (ข้อมูลครบถ้วนและถูกต้อง) <ul style="list-style-type: none"> 2.1 ตรวจสอบข้อมูลจากฐานข้อมูลรายงานข้อผิดพลาด ถ้า (มีข้อมูล) <ul style="list-style-type: none"> - แสดงข้อความว่าไม่สามารถเพิ่มรายงานข้อผิดพลาดได้ ถ้า (ไม่มีข้อมูล) <ul style="list-style-type: none"> - เพิ่มข้อมูลรายงานข้อผิดพลาด ถ้า (ข้อมูลไม่ครบถ้วนและไม่ถูกต้อง) <ul style="list-style-type: none"> - แสดงข้อความว่าข้อมูลไม่ถูกต้อง <p>จบการทำงาน</p>



ID	8.2
Name	แก้ไขรายงานข้อผิดพลาด
Description	การแก้ไขรายงานข้อผิดพลาด
Input Data Flows	- ข้อมูลรายงานข้อผิดพลาดที่ต้องการแก้ไข - ข้อมูลนักพัฒนาระบบ
Output Data Flows	-
Process Description	<p>เริ่มต้น</p> <ol style="list-style-type: none"> 1 รับข้อมูลรายงานข้อผิดพลาดที่ต้องการแก้ไข 2 ตรวจสอบว่ามีข้อมูลหรือไม่ <p>ถ้า (มีข้อมูล)</p> <ul style="list-style-type: none"> - แก้ไขข้อมูลที่ต้องการแก้ไข - บันทึกลงฐานข้อมูล <p>ถ้า (ไม่มีข้อมูล)</p> <ul style="list-style-type: none"> - แสดงข้อความว่าไม่พบข้อมูล <p>จบการทำงาน</p>

3.3 ความสัมพันธ์ (ER-Diagram)



ภาพประกอบที่ 3.9 ER-Diagram

3.4 การออกแบบฐานข้อมูล (Database Design)

ตารางที่ 3.4 ข้อมูลโปรเจ็ค

Attribute	Type	Description	Ex. Data	Constrain
project_id	int(11)	รหัสโปรเจ็ค	1	primary key
create_date	date	วันที่สร้างโปรเจ็ค	2021-09-02	
close_date				
name	varchar(100)	ชื่อโปรเจ็ค	Namthip	
details	varchar(500)	รายละเอียดของโปรเจ็ค	โปรเจ็คน้ำดื่ม	
photo	varchar(100)	รูปภาพหน้าปก	/img/background/image1.jpg	
status	int(1)	สถานะของโปรเจ็ค	1(1 = กำลังทำอยู่, 2 = ไม่ได้ใช้งาน)	
pm_id	int(11)	รหัสฟังก์ชัน	1	foreign key

ตารางที่ 3.5 ข้อมูลพนักงาน

Attribute	Type	Description	Ex. Data	Constrain
employee_id	int(11)	รหัสพนักงาน	1	primary key
name	varchar(50)	ชื่อพนักงาน	เรณูกา	
lastname	varchar(100)	นามสกุล	บุตรดา	
username	varchar(50)	ชื่อผู้ใช้	renuka@gmail.com	
password	varchar(64)	รหัสผ่าน	butda2564	
phone	varchar(10)	เบอร์โทรศัพท์	0935461321	
photo	varchar(100)	รูปโปรไฟล์	/img/member/image1.jpg	
position_id	int(11)	รหัสตำแหน่ง	1(1 = Project Manager, 2 = Software Tester, 3 = software Developer)	foreign key

ตารางที่ 3.6 ข้อมูลฟังก์ชัน

Attribute	Type	Description	Ex. Data	Constrain
func_id	int(11)	รหัสโปรเจ็ค	1	primary key
name	varchar(100)	ชื่อฟังก์ชัน	การเข้าสู่ระบบ	
project_id	int(11)	รหัสโปรเจ็ค	1	foreign key

ตารางที่ 3.7 ข้อมูลกรณีทดสอบ

Attribute	Type	Description	Ex. Data	Constrain
testcase_id	int(11)	รหัสของกรณีทดสอบ	1	primary key
name	varchar(100)	ชื่อกรณีทดสอบ	ตรวจสอบการเข้าฟังก์ชันสร้างบัญชีผู้ใช้	
test_data	varchar(200)	ข้อมูลที่ทดสอบ	username : renuka.com password : 2564002	
test_steps	varchar(200)	ขั้นตอนการทดสอบ	1.กรอกข้อมูล 2.คลิกปุ่ม login	
expected_result	varchar(200)	ผลลัพธ์ที่คาดหวัง	เข้าสู่ระบบได้	
status	int(1)	สถานะ	2 (1 = pass, 2 = fail)	
remark	varchar(100)	หมายเหตุ	-	
func_id	int(11)	รหัสโปรเจ็ค	1	foreign key
project_id	int(11)	รหัสของโปรเจ็ค	2	foreign key

ตารางที่ 3.8 ข้อมูลรายงานข้อผิดพลาด

Attribute	Type	Description	Ex. Data	Constrain
defect_id	int(11)	รหัสรายงานข้อผิดพลาด	1	primary key
title_name	varchar(100)	ชื่อหัวข้อรายงาน	สมาชิกเพิ่มสินค้าในตะกร้าไม่ได้	

ตารางที่ 3.9 ข้อมูลรายงานข้อผิดพลาด (ต่อ)

Attribute	Type	Description	Ex. Data	Constrain
environment	varchar(500)	สภาพแวดล้อมของระบบ	Desktop : เครื่องคอมพิวเตอร์ asus OS Version : Windows 10	
step_to_reproduce	varchar(500)	ขั้นตอนการทดสอบ	1.สมาชิก login เข้าสู่ระบบ 2.เลือกสินค้า และเพิ่มสินค้าลงตะกร้า 3.กดไปที่ตะกร้า	
actual_result	varchar(200)	ผลลัพธ์ที่เกิดขึ้น	ไม่พบสินค้า	
due_date	date	วันที่กำหนด	25-08-2021	
create_date	date	วันที่สร้าง	20-08-2021	
status	varchar(13)	สถานะ	1 (1 = Backlog, 2 = Doing, 3 = Ready for Test, 4 = Done)	
comment	varchar(200)	ความคิดเห็น	ลองมาหลายรอบ	
attachment	int(2)	สื่อที่แนบมา	1	
round	int(1)	จำนวนรอบ	1	
testcase_id	int(11)	รหัสความรุนแรง	3	foreign key
severity_id	int(11)	รหัสสมาชิก	3	foreign key
func_id	int(11)	รหัสโปรเจ็ค	1	foreign key
project_id	int(11)	รหัสของกรณีทดสอบ	3	foreign key
tester_id	int(11)	รหัสนักทดสอบระบบ	5	foreign key

ตารางที่ 3.8 ข้อมูลรายงานข้อผิดพลาด (ต่อ)

Attribute	Type	Description	Ex. Data	Constrain
dev_id	int(11)	รหัสนักพัฒนา ระบบ	1	foreign key
pm_id	int(11)	รหัส Project Manager	2	foreign key

ตารางที่ 3.10 ข้อมูลตำแหน่ง

Attribute	Type	Description	Ex. Data	Constrain
position_id	int(11)	รหัสตำแหน่ง	3	primary key
name	varchar(50)	ชื่อตำแหน่ง	Software Developer	

ตารางที่ 3.11 ข้อมูลระดับความรุนแรง

Attribute	Type	Description	Ex. Data	Constrain
severity_id	int(11)	รหัสความรุนแรง	1	primary key
name	varchar(20)	ชื่อระดับความรุนแรง	Minor	

ตารางที่ 3.12 ข้อมูลไฟล์ที่แนบมา

Attribute	Type	Description	Ex. Data	Constrain
attachment_id	int(11)	รหัสไฟล์	2	primary key
name	varchar(50)	ชื่อไฟล์	login.Namthip	
defect_id	int(11)	รหัสรายงาน ข้อผิดพลาด	4	foreign key

ตารางที่ 3.13 ข้อมูลสมาชิกในโปรเจ็ค

Attribute	Type	Description	Ex. Data	Constrain
project_id	int(11)	รหัสโปรเจ็ค	2	foreign key
employee_id	int(11)	รหัสพนักงาน	1, 2, 3, 4	foreign key
status	varchar(20)	สถานะในโปรเจ็ค	อยู่	

3.5 Code ในส่วนของ Back-end

```

8  function hash_password($password){
9  |     $hash = password_hash($password, PASSWORD_DEFAULT);
10 |     return $hash;
11 | }

```

ภาพประกอบที่ 3.10 ฟังก์ชัน hash password

บรรทัดที่ 8 จะรับ password มาจากผู้ใช้ที่กรอกข้อมูลมา

บรรทัดที่ 9 เอา password ที่ผู้ใช้กรอกมาทำการ hash

บรรทัดที่ 10 return รหัสที่ hash

```

13 function verify_password($password, $passwordINDB){
14 |
15 |     $check = password_verify($password, $passwordINDB);
16 |     return $check;
17 | }

```

ภาพประกอบที่ 3.11 ฟังก์ชัน verify password

บรรทัดที่ 13 จะรับค่า password ที่ผู้ใช้กรอกและรับค่า passwordINDB คือรหัสผ่านของผู้ใช้ที่อยู่ในดาต้าเบส

บรรทัดที่ 15 เปรียบเทียบรหัสทั้งสองมาเทียบกันว่าตรงกันหรือไม่

บรรทัดที่ 16 return ตัวแปร check

```

19 function getPasswordFromDB($conn, $username){
20 |     $stmt = $conn->prepare("select password from employee where username=?");
21 |     $stmt->bind_param("s", $username);
22 |     $stmt->execute();
23 |     $result = $stmt->get_result();
24 |     if($result->num_rows == 1){
25 |         $row = $result->fetch_assoc();
26 |         return $row["password"];
27 |     }else{
28 |         return "";
29 |     }
30 | }

```

ภาพประกอบที่ 3.12 ฟังก์ชัน get password from DB

บรรทัดที่ 19 รับตัวแปรสองตัว

บรรทัดที่ 20 เป็นคำสั่งที่ select ข้อมูลจากดาต้าเบส

บรรทัดที่ 21 ส่งค่าตัวแปรไปใช้ในคำสั่ง sql

บรรทัดที่ 22 และทำการเรียกใช้คำสั่ง sql

บรรทัดที่ 23 เอาข้อมูลเก็บไว้ที่ตัวแปร result

บรรทัดที่ 24 ถ้าตัวแปร result มีข้อมูล 1 แถว

บรรทัดที่ 25 เก็บข้อมูลตัวแปร result ไว้เป็น array โดยใช้เมธอด fetch_assoc()

บรรทัดที่ 26 return คอลัมน์ password

```

50 //login
51 $app->post('/login', function (Request $request, Response $response, $args) {
52     $body = $request->getBody();
53     $json = json_decode($body,true);
54     $username = $json['username'];
55     $password = $json['password'];
56
57     $conn = $GLOBALS['dbconn'];
58     $passwordINDB = getPasswordFromDB($conn, $username);
59
60     $check = verify_password($password, $passwordINDB);
61     $mes = "";
62
63     if($check){
64         $mes = "success";
65     }
66     else{
67         $mes = "fail";
68     }
69
70     $response->getBody()->write("status, $mes $passwordINDB");
71     return $response;
72 });

```

ภาพประกอบที่ 3.13 Service การเข้าสู่ระบบ

บรรทัดที่ 51 เรียกเมธอด post โดยเรียก path /login

บรรทัดที่ 52 ขอข้อมูลโดยใช้เมธอด getBody()

บรรทัดที่ 53 รับข้อมูลในรูปแบบ json

บรรทัดที่ 54 รับค่าข้อมูล username เก็บไว้ที่ \$username

บรรทัดที่ 55 รับค่าข้อมูล password เก็บไว้ที่ \$password

บรรทัดที่ 58 เอาข้อมูล password จากดาต้าเบส เรียกใช้ฟังก์ชัน getPasswordFromDB และส่งค่า \$conn และ \$username

บรรทัดที่ 60 เช็ครหัส password จากผู้ใช้ที่กรอก กับรหัสในดาต้าเบส

บรรทัดที่ 61 ประกาศตัวแปร

บรรทัดที่ 63-68 ถ้าตัวแปร check มีค่าเท่ากับ true ให้แสดงข้อความว่า success ถ้าไม่ใช่ให้แสดงข้อความว่า fail

บรรทัดที่ 70-71 ส่งข้อมูลกลับไปโดยมีข้อมูล \$mes และ password ในดาต้าเบส


```

74 //register
75 $app->post('/register', function (Request $request, Response $response, $args){
76     $body = $request->getBody();
77     $json = json_decode($body,true);
78     $conn = $GLOBALS['dbconn'];
79     $status = insertEmployee($conn, $json);
80
81     $response->getBody()->write("status, $status");
82     return $response;
83 });

```

ภาพประกอบที่ 3.14 Service การสมัครสมาชิก

บรรทัดที่ 75 เรียกเมธอด post โดยเรียก path /register

บรรทัดที่ 76 ขอข้อมูลโดยใช้เมธอด getBody()

บรรทัดที่ 77 รับข้อมูลในรูปแบบ json

บรรทัดที่ 79 เรียกใช้ฟังก์ชัน insertEmployee และส่งค่า \$conn และ \$json

บรรทัดที่ 81-82 ส่งข้อมูลกลับไปโดยบอกสถานะ

```

8 function insertProject($conn, $json){
9     $now = new DateTime();
10    $date = $now->format('Y-m-d');
11    $name = $json['name'];
12    $details = $json['details'];
13    $photo = $json['photo'];
14    $status = 1;
15    $pm_id = $json['pm_id'];
16
17    $stmt = $conn->prepare("INSERT INTO `project`(`create_date`, `name`, `details`, `photo`, `status`, `pm_id`)
18    ." VALUES (?,?,,?,?,?)");
19    $stmt->bind_param("ssssi", $date, $name, $details, $photo, $status, $pm_id);
20    $stmt->execute();
21    $result = $stmt->affected_rows;
22
23    return $result;
24 }

```

ภาพประกอบที่ 3.15 ฟังก์ชันการเพิ่มโปรเจ็ค

บรรทัดที่ 8 ฟังก์ชัน insertProject โดยรับค่าตัวแปร \$conn และ \$json

บรรทัดที่ 9 สร้าง object ของคลาส DateTime()

บรรทัดที่ 10-15 รับค่าข้อมูล

บรรทัดที่ 17-19 เป็นคำสั่ง sql เพิ่มข้อมูลโปรเจ็คลงในดาต้าเบส

บรรทัดที่ 20 เรียกใช้คำสั่ง sql

บรรทัดที่ 21-23 แถวที่ได้รับการประมวลผล และ return กลับไป

```

26 function addMember($conn, $json){
27     $project_id = $json['project_id'];
28     $member = $json['member'];
29     $result = 1 ;
30     foreach ($member as $s){
31         $employee_id = $s["employee_id"];
32         $stmt = $conn->prepare("INSERT INTO `project_member`(`project_id`, `employee_id`)
33             . "VALUES (?,?)");
34         $stmt->bind_param("ii",$project_id, $employee_id);
35         $stmt->execute();
36         $result = $stmt->affected_rows;
37     }
38 }
39
40 return $result;
41 }

```

ภาพประกอบที่ 3.16 ฟังก์ชันการเพิ่มสมาชิกโปรเจ็ค

บรรทัดที่ 26 ฟังก์ชัน addMember โดยรับค่า \$conn และ \$json

บรรทัดที่ 27-28 รับค่าตัวแปร project_id, member

บรรทัดที่ 30-38 เป็นคำสั่งลูปที่เพิ่มสมาชิกลงในดาต้าเบส

บรรทัดที่ 40 return result

```

44 //select project
45 $app->get('/project', function (Request $request, Response $response, $args) {
46
47     $conn = $GLOBALS['dbconn'];
48
49     $stmt = $conn->query("select * from project ");
50     $data = array();
51     while($row = $stmt->fetch_assoc()){
52         array_push($data, $row);
53     }
54
55     $json = json_encode($data);
56     $response->getBody()->write($json);
57     return $response->withHeader('Content-Type', 'application/json');
58 });

```

ภาพประกอบที่ 3.17 Service select ข้อมูลโปรเจ็ค

บรรทัดที่ 45 เรียกเมธอดแบบ get โดยใช้ path /project

บรรทัดที่ 49 คำสั่ง sql select ข้อมูลของโปรเจ็ค

บรรทัดที่ 50 ประกาศตัวแปรอาร์เรย์

บรรทัดที่ 51-53 เป็นลูป while ที่เก็บข้อมูลไว้ในอาร์เรย์

บรรทัดที่ 55 เอาข้อมูลที่เก็บเป็นอาร์เรย์มาแปลงเป็นข้อมูลแบบ json

บรรทัดที่ 56-57 ส่งข้อมูลกลับไปแบบ json

```

61 //insert project
62 $app->post('/createproject', function (Request $request, Response $response, $args) {
63     $conn = $GLOBALS['dbconn'];
64     $body = $request->getBody();
65     $json = json_decode($body,true);
66     $status = insertProject($conn, $json);
67
68     $response->getBody()->write("status, $status");
69     return $response;
70 });

```

ภาพประกอบที่ 3.18 Service การเพิ่มข้อมูลโปรเจ็ค

บรรทัดที่ 62 เรียกเมธอดแบบ post โดยใช้ path /createproject

บรรทัดที่ 64 ขอข้อมูลโดยใช้เมธอด getBody()

บรรทัดที่ 65 รับข้อมูลในรูปแบบ json

บรรทัดที่ 66 เรียกใช้ฟังก์ชัน insertProject

บรรทัดที่ 68-69 ส่งข้อมูลกลับไป

```

73 //edit status project
74 $app->post('/editProject', function (Request $request, Response $response, $args) {
75     $conn = $GLOBALS['dbconn'];
76
77     $body = $request->getBody();
78     $json = json_decode($body,true);
79
80     $project_id = $json['project_id'];
81     $status = $json['status'];
82     $sql = "UPDATE `project` SET `status` = ".$status." WHERE `project_id` = ".$project_id;
83
84     $stmt = $conn->query($sql);
85     if ($stmt === TRUE) {
86         $str = "Record updated successfully";
87     } else {
88         $str = "Error updating record: " . $conn->error;
89     }
90     $response->getBody()->write($str);
91     return $response;
92 });

```

ภาพประกอบที่ 3.19 Service การแก้ไขข้อมูลโปรเจ็ค

บรรทัดที่ 74 เรียกเมธอดแบบ post โดยใช้ path /editProject

บรรทัดที่ 77 ขอข้อมูลโดยใช้เมธอด getBody()

บรรทัดที่ 78 รับข้อมูลในรูปแบบของ json

บรรทัดที่ 80-81 รับข้อมูลตัวแปร

บรรทัดที่ 82 เป็นคำสั่ง sql ของการแก้ไขข้อมูลโปรเจ็ค โดยการรับค่า project_id และ status

บรรทัดที่ 84 เขียนลงในดาต้าเบส

บรรทัดที่ 85-89 เงื่อนไขถ้าแก้ไขข้อมูลได้จะแสดงข้อความ ถ้าไม่ได้ให้แสดงข้อความและแสดง error

บรรทัดที่ 90-91 return string กลับไป

```

95 //delete member
96 $app->post('/deleteMember', function (Request $request, Response $response, $args) {
97     $conn = $GLOBALS['dbconn'];
98
99     $body = $request->getBody();
100     $json = json_decode($body,true);
101
102     $project_id = $json['project_id'];
103     $member = $json['employee_id'];
104     $sql = "DELETE FROM `project_member` WHERE `project_id` = '".$project_id']."' AND `employee_id` = '".$member'";
105
106     $stmt = $conn->query($sql);
107     if ($stmt === TRUE) {
108         $str = "Delete successfully";
109     } else {
110         $str = "Error Delete: " . $conn->error;
111     }
112     $response->getBody()->write($str);
113     return $response;
114 });

```

ภาพประกอบที่ 3.20 Service การลบข้อมูลโปรเจ็ค

บรรทัดที่ 96 เรียกเมธอดแบบ post โดยใช้ path /deleteMember

บรรทัดที่ 99 ขอข้อมูลโดยใช้เมธอด getBody()

บรรทัดที่ 100 รับข้อมูลในรูปแบบของ json

บรรทัดที่ 102-103 รับข้อมูลตัวแปร

บรรทัดที่ 104 เป็นคำสั่ง sql ที่ใช้ลบข้อมูลสมาชิกโปรเจ็ค

บรรทัดที่ 107-108 ถ้าลบข้อมูลได้จะแสดงข้อความ Delete successfully

บรรทัดที่ 109-110 ถ้าลบข้อมูลไม่ได้จะแสดงข้อความว่า Error Delete และแสดง error

บรรทัด 112-113 return string กลับไป

```

116 //-----ADD MEMBER-----//
117 $app->post('/addMember', function (Request $request, Response $response, $args) {
118     $conn = $GLOBALS['dbconn'];
119
120     $body = $request->getBody();
121     $json = json_decode($body,true);
122
123     $status = addMember($conn, $json);
124
125     $response->getBody()->write("status, $status");
126     return $response;
127 });

```

ภาพประกอบที่ 3.21 Service การเพิ่มสมาชิกโปรเจ็ค

บรรทัดที่ 117 เรียกเมธอดแบบ post โดยใช้ path /addMember

บรรทัดที่ 120 ขอข้อมูลโดยใช้เมธอด getBody()

บรรทัดที่ 121 รับข้อมูลในรูปแบบของ json

บรรทัดที่ 123 เรียกฟังก์ชัน addMember

บรรทัดที่ 125-126 return ข้อมูลกลับไป

```

9 //select employee
10 $app->get('/emp', function (Request $request, Response $response, $args) {
11
12     $conn = $GLOBALS['dbconn'];
13
14     $stmt = $conn->query("select * from employee ");
15     $data = array();
16     while($row = $stmt->fetch_assoc()){
17         array_push($data, $row);
18     }
19
20     $json = json_encode($data);
21     $response->getBody()->write($json);
22
23     return $response->withHeader('Content-Type', 'application/json');
24 });

```

ภาพประกอบที่ 3.22 Service select ข้อมูลพนักงาน

บรรทัดที่ 10 เรียกเมธอดแบบ get โดยใช้ path /emp

บรรทัดที่ 14 เป็นคำสั่ง sql ที่จะ select ข้อมูลของ employee

บรรทัดที่ 15 ประกาศตัวแปรอาร์เรย์

บรรทัดที่ 16-18 เป็นลูปที่เก็บข้อมูลไว้เป็นอาร์เรย์

บรรทัดที่ 20 เอาค่าข้อมูลที่อยู่ในตัวแปร data ออกมาเป็นข้อมูลแบบ json

บรรทัดที่ 21-23 return ข้อมูลกลับไปเป็น json

```

27 //edit employee
28 $app->post('/editEmp', function (Request $request, Response $response, $args) {
29     $conn = $GLOBALS['dbconn'];
30
31     $body = $request->getBody();
32     $json = json_decode($body,true);
33
34     $str = "";
35     foreach ($json as $key => $value) {
36         if($key == 'employee_id'){
37             continue;
38         }
39         // update
40         $sql = "UPDATE `employee` SET `".$key."` = '".$value'"
41             ." WHERE `employee_id`='".$json['employee_id']";
42         if ($conn->query($sql) === TRUE) {
43             $str = "Record updated successfully";
44         } else {
45             $str = "Error updating record: " . $conn->error;
46         }
47     }
48
49     $response->getBody()->write($str);
50     return $response;
51 });

```

ภาพประกอบที่ 3.23 Service แก้ไขข้อมูลพนักงาน

บรรทัดที่ 28 เรียกเมธอดแบบ post โดยใช้ path /editEmp

บรรทัดที่ 31 ขอข้อมูลโดยใช้เมธอด getBody()

บรรทัดที่ 32 รับข้อมูลในรูปแบบของ json

บรรทัดที่ 35 เป็นลูปข้อมูล json ที่ให้ตัวแปร key มีค่าเท่ากับ ตัวแปร value

บรรทัดที่ 36-38 ถ้าตัวแปร key เท่ากับ employee_id ให้ทำต่อ

บรรทัดที่ 40-41 เป็นคำสั่ง sql จะ set key เท่ากับ value ที่ employee_id

บรรทัดที่ 42-43 ถ้าแก้ไขข้อมูลได้จะแสดงข้อความว่า Record updated successfully

บรรทัดที่ 44-45 ถ้าแก้ไขข้อมูลไม่ได้จะแสดงข้อความว่า Error updating record: พร้อมกับ error

บรรทัดที่ 49-50 return string กลับไป

```

8  function insertFunction($conn, $json) {
9      $name = $json['name'];
10     $project_id = $json['project_id'];
11
12     $stmt = $conn->prepare("INSERT INTO `function`(`name`, `project_id`)
13     | . "VALUES (?,?)");
14     $stmt->bind_param("si", $name, $project_id);
15     $stmt->execute();
16     $result = $stmt->affected_rows;
17
18     return $result;
19 }

```

ภาพประกอบที่ 3.24 ฟังก์ชันเพิ่มข้อมูลของฟังก์ชัน

บรรทัดที่ 8 ฟังก์ชัน insertFunction รับค่าข้อมูลเข้ามา

บรรทัดที่ 9-10 รับค่าข้อมูลตัวแปร

บรรทัดที่ 12-16 เป็นคำสั่ง sql ที่จะเพิ่มข้อมูลของฟังก์ชัน

บรรทัดที่ 18 return ข้อมูล

```

21 function inserttestcase($conn, $json) {
22     $name = $json['name'];
23     $test_data = $json['test_data'];
24     $test_step = $json['test_step'];
25     $expected_result = $json['expected_result'];
26     $status = $json['status'];
27     $remark = $json['remark'];
28     $func_id = $json['func_id'];
29     $project_id = $json['project_id'];
30
31     $stmt = $conn->prepare("INSERT INTO `testcase`(`name`, `test_data`, `test_step`, `expected_result`, `status`, `remark`, `func_id`, `project_id`)
32     | . "VALUES (?,?,?,?,?,?,?,?)");
33     $stmt->bind_param("ssssssii", $name, $test_data, $test_step, $expected_result, $status, $remark, $func_id, $project_id);
34     $stmt->execute();
35     $result = $stmt->affected_rows;
36
37     return $result;
38 }

```

ภาพประกอบที่ 3.25 ฟังก์ชันเพิ่มข้อมูลของกรณีทดสอบ

บรรทัดที่ 21 เป็นฟังก์ชัน inserttestcase รับค่าข้อมูลเข้ามา

บรรทัดที่ 22-29 รับค่าข้อมูลตามตัวแปร

บรรทัดที่ 31-35 เป็นคำสั่ง sql ที่จะเพิ่มข้อมูลของการฝึกที่จะทดสอบ

บรรทัดที่ 37 return ข้อมูลกลับไป

```

41 //select function
42 $app->get('/function', function (Request $request, Response $response, $args) {
43
44     $conn = $GLOBALS['dbconn'];
45
46     $stmt = $conn->query("select * from function ");
47     $data = array();
48     while($row = $stmt->fetch_assoc()){
49         array_push($data, $row);
50     }
51
52     $json = json_encode($data);
53     $response->getBody()->write($json);
54
55     return $response->withHeader('Content-Type', 'application/json');
56 });

```

ภาพประกอบที่ 3.26 Service select ข้อมูลของฟังก์ชันที่ทดสอบระบบ

บรรทัดที่ 42 เรียกเมธอดแบบ get โดยใช้ path /function

บรรทัดที่ 46 เป็นคำสั่ง select ข้อมูลของฟังก์ชัน

บรรทัดที่ 47 ประกาศตัวแปรอาร์เรย์

บรรทัดที่ 48-49 เป็นลูปที่เก็บข้อมูลอาร์เรย์

บรรทัดที่ 52 เอาข้อมูลที่เก็บไว้เป็นอาร์เรย์ออกมาเป็นข้อมูลแบบ json

บรรทัดที่ return ข้อมูลในรูปแบบ json

```

58 //select testcase
59 $app->get('/testcase', function (Request $request, Response $response, $args) {
60
61     $conn = $GLOBALS['dbconn'];
62
63     $stmt = $conn->query("select * from testcase ");
64     $data = array();
65     while($row = $stmt->fetch_assoc()){
66         array_push($data, $row);
67     }
68
69     $json = json_encode($data);
70     $response->getBody()->write($json);
71
72     return $response->withHeader('Content-Type', 'application/json');
73 });

```

ภาพประกอบที่ 3.27 Service select ข้อมูลของกรณีทดสอบ

- บรรทัดที่ 59 เรียกเมธอดแบบ get โดยใช้ path /testcase
- บรรทัดที่ 63 เป็นคำสั่ง sql ที่จะ select ข้อมูลของกรณีทดสอบ
- บรรทัดที่ 64 ประกาศตัวแปรอาร์เรย์
- บรรทัดที่ 65-66 เป็นลูปที่เก็บข้อมูลไว้เป็นอาร์เรย์
- บรรทัดที่ 69 เอาข้อมูลอาร์เรย์ออกมาเป็นข้อมูลแบบ json
- บรรทัดที่ 70-72 return ข้อมูลกลับไป

```

76 //create function
77 $app->post('/createfunction',function (Request $request, Response $response, $args) {
78     $conn = $GLOBALS['dbconn'];
79     $body = $request->getBody();
80     $json = json_decode($body,true);
81     $status = insertFunction($conn, $json);
82
83     $response->getBody()->write("status, $status");
84     return $response;
85 });

```

ภาพประกอบที่ 3.28 Service การเพิ่มฟังก์ชันที่ทดสอบ

- บรรทัดที่ 77 เรียกเมธอดแบบ post โดยใช้ path /createfunction
- บรรทัดที่ 79 ขอข้อมูลโดยใช้เมธอด getBody()
- บรรทัดที่ 80 รับข้อมูลในรูปแบบของ json
- บรรทัดที่ 81 เรียกใช้ฟังก์ชัน insertFunction โดยส่งค่าตัวแปรไป
- บรรทัดที่ 83-84 return ข้อมูล

```

87 //create testcase
88 $app->post('/createtestcase',function (Request $request, Response $response, $args) {
89     $conn = $GLOBALS['dbconn'];
90     $body = $request->getBody();
91     $json = json_decode($body,true);
92     $status = inserttestcase($conn, $json);
93
94     $response->getBody()->write("status, $status");
95     return $response;
96 });

```

ภาพประกอบที่ 3.29 Service การเพิ่มกรณีทดสอบ

- บรรทัดที่ 88 เรียกเมธอดแบบ post โดยใช้ path /createtestcase
- บรรทัดที่ 90 ขอข้อมูลโดยใช้เมธอด getBody()
- บรรทัดที่ 91 รับข้อมูลในรูปแบบของ json
- บรรทัดที่ 92 เรียกใช้ฟังก์ชัน inserttestcase โดยส่งค่าตัวแปรไป
- บรรทัดที่ 94-95 return ข้อมูล


```

99 // edit function
100 $app->post('/editFunction',function (Request $request, Response $response, $args) {
101     $conn = $GLOBALS['dbconn'];
102
103     $body = $request->getBody();
104     $json = json_decode($body,true);
105
106     foreach ($json as $key => $value) {
107         if($key == 'func_id'){
108             continue;
109         }
110         // update
111         $sql = "UPDATE `function` SET `".$key."` = '".$value'"
112             ." WHERE `func_id`='".$json['func_id']";
113         if ($conn->query($sql) === TRUE) {
114             $str = "Record updated successfully";
115         } else {
116             $str = "Error updating record: " . $conn->error;
117         }
118     }
119     $response->getBody()->write($str);
120     return $response;
121 });

```

ภาพประกอบที่ 3.30 Service การแก้ไขข้อมูลของฟังก์ชัน

บรรทัดที่ 100 เรียกเมธอดแบบ post โดยใช้ path /editFunction

บรรทัดที่ 103 ขอข้อมูลโดยใช้เมธอด getBody()

บรรทัดที่ 104 รับข้อมูลในรูปแบบของ json

บรรทัดที่ 111-112 เป็นคำสั่ง sql ที่แก้ไขข้อมูลของฟังก์ชันที่ set key เท่ากับ value ที่ func_id

บรรทัดที่ 113-114 ถ้าแก้ไขข้อมูลถูกต้องจะแสดงข้อความว่า Record updated successfully

บรรทัดที่ 115-116 ถ้าแก้ไขข้อมูลไม่ถูกต้องจะแสดงข้อความว่า Error updating record: และแสดง error

บรรทัดที่ 119-120 return ข้อมูล

```

123 //edit testcase
124 $app->post('/editTestcase',function (Request $request, Response $response, $args) {
125     $conn = $GLOBALS['dbconn'];
126
127     $body = $request->getBody();
128     $json = json_decode($body,true);
129
130     $str = "";
131     foreach ($json as $key => $value) {
132         if($key == 'testcase_id'){
133             continue;
134         }
135         // update
136         $sql = "UPDATE `testcase` SET `".$key."` = '".$value'"
137             ." WHERE `testcase_id`='".$json['testcase_id']";
138         if ($conn->query($sql) === TRUE) {
139             $str = "Record updated successfully";
140         } else {
141             $str = "Error updating record: " . $conn->error;
142         }
143     }
144
145     $response->getBody()->write($str);
146     return $response;
147 });

```

ภาพประกอบที่ 3.31 Service การแก้ไขข้อมูลของกรณีทดสอบ

บรรทัดที่ 124 เรียกเมธอดแบบ post โดยใช้ path /editTestcase

บรรทัดที่ 127 ขอข้อมูลโดยใช้เมธอด getBody()

บรรทัดที่ 128 รับข้อมูลในรูปแบบของ json

บรรทัดที่ 136-137 เป็นคำสั่ง sql ที่แก้ไขข้อมูลของฟังก์ชันที่ set key เท่ากับ value ที่ testcase_id

บรรทัดที่ 138-139 ถ้าแก้ไขข้อมูลได้จะแสดงข้อความ Record updated successfully

บรรทัดที่ 141-142 ถ้าแก้ไขข้อมูลไม่ถูกต้องจะแสดงข้อความว่า Error updating record: และแสดง error

บรรทัดที่ 145-146 return ข้อมูล

```

150 //delete function
151 $app->post('/deleteFunction',function (Request $request, Response $response, $args) {
152     $conn = $GLOBALS['dbconn'];
153
154     $body = $request->getBody();
155     $json = json_decode($body,true);
156
157     $func_id = $json['func_id'];
158     $sql = "DELETE FROM `function` WHERE `func_id` = ".$func_id;
159     $stmt = $conn->query($sql);
160     if ($stmt === TRUE) {
161         $str = "Delete successfully";
162     } else {
163         $str = "Error delete record: " . $conn->error;
164     }
165     $response->getBody()->write($str);
166     return $response;
167 });

```

ภาพประกอบที่ 3.32 Service การลบข้อมูลฟังก์ชัน

บรรทัดที่ 151 เรียกเมธอดแบบ post โดยใช้ path /deleteFunction

บรรทัดที่ 154 ขอข้อมูลโดยใช้เมธอด getBody()

บรรทัดที่ 155 รับข้อมูลในรูปแบบของ json

บรรทัดที่ 158 เป็นคำสั่ง sql ที่ใช้ลบข้อมูลฟังก์ชันในดาต้าเบส โดยการส่งค่า func_id

บรรทัดที่ 160-161 ถ่าลบข้อมูลได้จะแสดงข้อความว่า Delete successfully

บรรทัดที่ 162-163 ถ่าลบข้อมูลไม่สำเร็จจะแสดงข้อความว่า Error delete record: และแสดง error

บรรทัดที่ 165-166 return ข้อมูล

```

169 //delete testcase
170 $app->post('/deleteTestcase',function (Request $request, Response $response, $args) {
171     $conn = $GLOBALS['dbconn'];
172
173     $body = $request->getBody();
174     $json = json_decode($body,true);
175
176     $testcase_id = $json['testcase_id'];
177     $sql = "DELETE FROM `testcase` WHERE `testcase_id` =".$testcase_id;
178     $stmt = $conn->query($sql);
179     if ($stmt === TRUE) {
180         $str = "Delete successfully";
181     } else {
182         $str = "Error delete record: " . $conn->error;
183     }
184     $response->getBody()->write($str);
185     return $response;
186 });

```

ภาพประกอบที่ 3.33 Service การลบข้อมูลของกรณีทดสอบ

บรรทัดที่ 170 เรียกเมธอดแบบ post โดยใช้ path /deleteTestcase

บรรทัดที่ 173 ขอข้อมูลโดยใช้เมธอด getBody()

บรรทัดที่ 174 รับข้อมูลในรูปแบบของ json

บรรทัดที่ 177 เป็นคำสั่ง sql ที่ใช้ลบข้อมูลกรณีทดสอบในดาต้าเบส โดยการส่งค่า testcase_id

บรรทัดที่ 179-180 ถ้ลบข้อมูลได้จะแสดงข้อความว่า Delete successfully

บรรทัดที่ 181-182 ถ้ลบข้อมูลไม่สำเร็จจะแสดงข้อความว่า Error delete record: และแสดง error

บรรทัดที่ 184-185 return ข้อมูล

```

8 function getPM($conn,$project_id) {
9     $conn = $GLOBALS['dbconn'];
10
11     $stmt = $conn->query("SELECT `pm_id` FROM `project` WHERE `project_id` =".$project_id);
12     $pm_id = 0 ;
13     while($row = $stmt->fetch_assoc()){
14         $pm_id = $row['pm_id'] ;
15     }
16     return $pm_id;
17 }

```

ภาพประกอบที่ 3.34 ฟังก์ชัน getPM

บรรทัดที่ 8 ฟังก์ชัน getPm จะรับค่า project_id

บรรทัดที่ 11 เป็นคำสั่ง sql ที่ select pm_id จาก table project

บรรทัดที่ 13-14 เป็นลูปการเก็บข้อมูล

บรรทัดที่ 16 return ข้อมูล

```

55 //select defect
56 $app->get('/defect', function (Request $request, Response $response, $args) {
57
58     $conn = $GLOBALS['dbconn'];
59
60     $stmt = $conn->query("select * from defect ");
61     $data = array();
62     while($row = $stmt->fetch_assoc()){
63         array_push($data, $row);
64     }
65
66     $json = json_encode($data);
67     $response->getBody()->write($json);
68     return $response->withHeader('Content-Type', 'application/json');
69 });

```

ภาพประกอบที่ 3.35 Service select ข้อมูลรายงานข้อผิดพลาดซอฟต์แวร์

บรรทัดที่ 56 เรียกเมธอดแบบ get โดยใช้ path /defect

บรรทัดที่ 60 เป็นคำสั่ง sql ที่ select ข้อมูลรายงานข้อผิดพลาดมาจากดาต้าเบส

บรรทัดที่ 61 ประกาศตัวแปรอาร์เรย์

บรรทัดที่ 62-63 เป็นลูปการเก็บข้อมูลไว้เป็นอาร์เรย์

บรรทัดที่ 66 เอาข้อมูลที่เก็บเป็นอาร์เรย์ออกมาเป็นข้อมูลแบบ json

บรรทัดที่ 67-68 return ข้อมูล

```

72 //create defect
73 $app->post('/createdefect',function (Request $request, Response $response, $args) {
74     $conn = $GLOBALS['dbconn'];
75
76     $body = $request->getBody();
77     $json = json_decode($body,true);
78     $status = insertDefect($conn, $json);
79
80     $response->getBody()->write("status, $status");
81     return $response;
82 });

```

ภาพประกอบที่ 3.36 Service การเพิ่มข้อมูลรายงานข้อผิดพลาด

บรรทัดที่ 73 เรียกเมธอดแบบ post โดยใช้ path /createdefect

บรรทัดที่ 76 ขอข้อมูลโดยใช้เมธอด getBody()

บรรทัดที่ 77 รับข้อมูลในรูปแบบของ json

บรรทัดที่ 78 เรียกใช้ฟังก์ชัน insertDefect โดยส่งค่า \$conn และ \$json

บรรทัดที่ 80-81 return ข้อมูล

```

104 //edit defect
105 $app->post('/editDefect',function (Request $request, Response $response, $args) {
106     $conn = $GLOBALS['dbconn'];
107
108     $body = $request->getBody();
109     $json = json_decode($body,true);
110
111     $str = "";
112     foreach ($json as $key => $value) {
113         if($key == 'defect_id'){
114             continue;
115         }
116         // update
117         $sql = "UPDATE `defect` SET `".$key."` = '".$value.'"
118             ." WHERE `defect_id`='".$json['defect_id']";
119         if ($conn->query($sql) === TRUE) {
120             $str = "Record updated successfully";
121         } else {
122             $str = "Error updating record: " . $conn->error;
123         }
124     }
125
126     $response->getBody()->write($str);
127     return $response;
128 });

```

ภาพประกอบที่ 3.37 Service แก้ไขรายงานข้อผิดพลาด

บรรทัดที่ 105 เรียกเมธอดแบบ post โดยใช้ path /editDefect

บรรทัดที่ 108 ขอข้อมูลโดยใช้เมธอด getBody()

บรรทัดที่ 109 รับข้อมูลในรูปแบบของ json

บรรทัดที่ 112 เป็นลูปที่ให้ข้อมูล json key = value

บรรทัดที่ 113-114 ถ้า key มีค่าเท่ากับ defect_id ให้ทำต่อ

บรรทัดที่ 117-118 เป็นคำสั่ง sql ที่แก้ไขข้อมูลของฟังก์ชันที่ set key เท่ากับ value ที่ defect_id

บรรทัดที่ 119-120 ถ้าแก้ไขข้อมูลสำเร็จจะแสดงข้อความว่า Record updated successfully

บรรทัดที่ 121-122 ถ้าแก้ไขข้อมูลไม่สำเร็จจะแสดงข้อความว่า Error updating record: และแสดง error

บรรทัดที่ 126-127 return ข้อมูล

```

131 //delete Defect
132 $app->post('/deleteDefect',function (Request $request, Response $response, $args) {
133     $conn = $GLOBALS['dbconn'];
134
135     $body = $request->getBody();
136     $json = json_decode($body,true);
137
138     $defect_id = $json['defect_id'];
139     $sql = "DELETE FROM `defect` WHERE `defect_id` =".$defect_id;
140     $stmt = $conn->query($sql);
141     if ($stmt === TRUE) {
142         $str = "Delete successfully";
143     } else {
144         $str = "Error delete record: " . $conn->error;
145     }
146     $response->getBody()->write($str);
147     return $response;
148 });

```

ภาพประกอบที่ 3.38 Service การลบรายงานข้อผิดพลาด

บรรทัดที่ 132 เรียกเมธอดแบบ post โดยใช้ path /deleteDefect

บรรทัดที่ 135 ขอข้อมูลโดยใช้เมธอด getBody()

บรรทัดที่ 136 รับข้อมูลในรูปแบบของ json

บรรทัดที่ 139 เป็นคำสั่ง sql ที่จะลบข้อมูลในดาต้าเบส โดยการส่ง defect_id

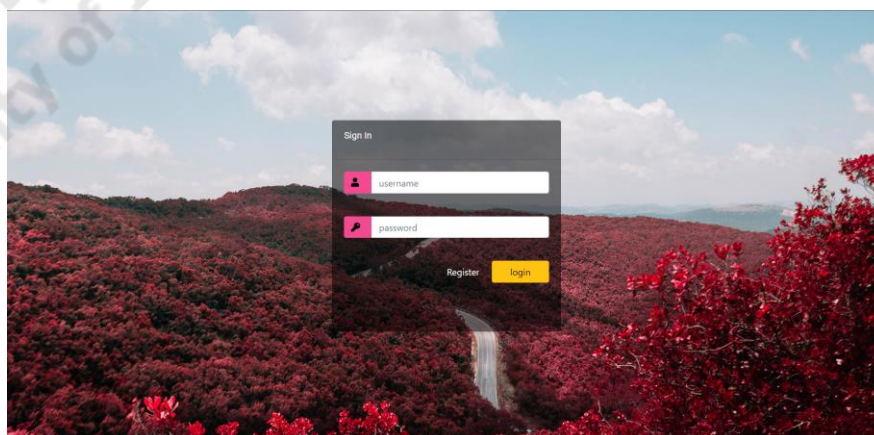
บรรทัดที่ 141-142 ถ้าลบข้อมูลได้จะแสดงข้อความว่า Delete successfully

บรรทัดที่ 143-144 ถ้าลบข้อมูลไม่ได้จะแสดงข้อความว่า Error delete record: และจะแสดง error

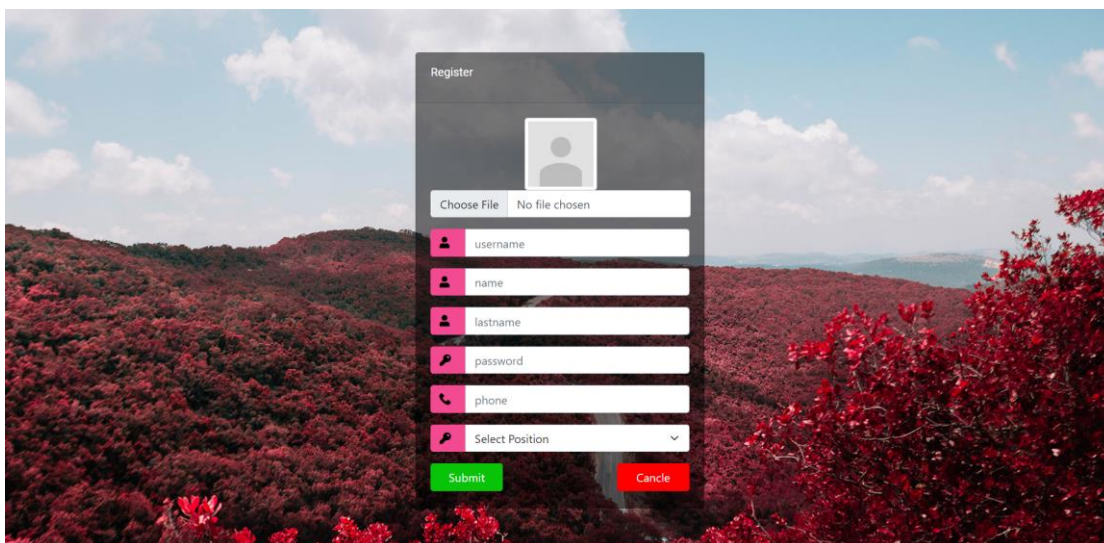
บรรทัดที่ 146-147 return ข้อมูล

3.6 การออกแบบหน้าเว็บไซต์

3.6.1 การออกแบบหน้าเว็บไซต์การเข้าสู่ระบบ และสมัครสมาชิก



ภาพประกอบที่ 3.39 หน้าการเข้าสู่ระบบ

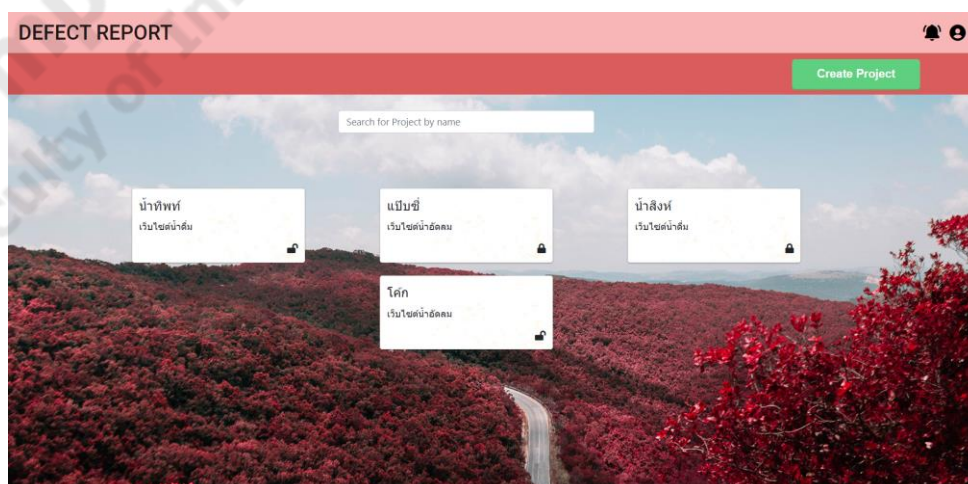


ภาพประกอบที่ 3.40 หน้าการสมัครสมาชิก

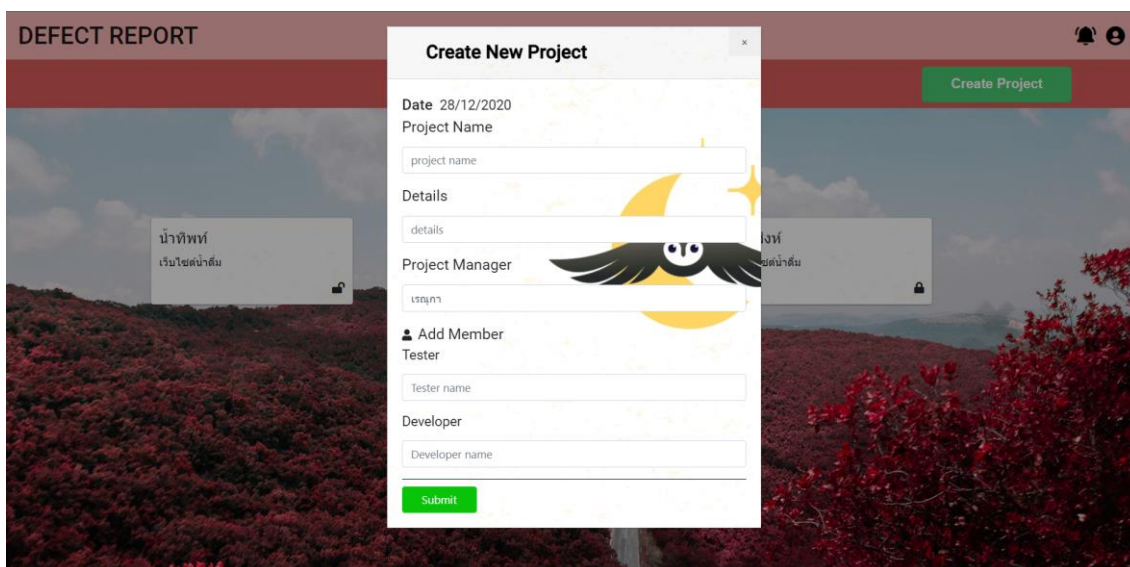
จากภาพประกอบที่ 3.40 เป็นหน้าการสมัครสมาชิกของพนักงาน ซึ่งจะประกอบไปด้วย

- choose file คือ ช่องที่ต้องเลือกรูปโปรไฟล์
- username คือ ช่องที่ต้องกรอก username
- name คือ ช่องที่ต้องกรอกชื่อจริง
- last name คือ ช่องที่ต้องกรอกนามสกุล
- password คือ ช่องที่ต้องกรอกรหัสผ่าน
- phone คือ ช่องที่ต้องกรอกเบอร์โทร
- select position คือ ช่องที่ต้องเลือกตำแหน่งงาน

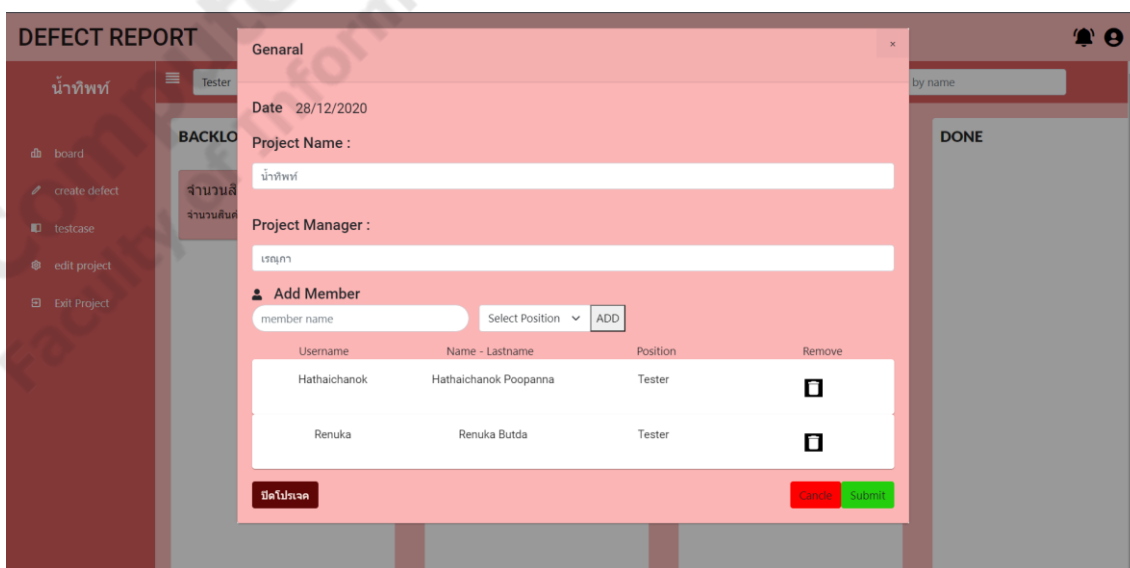
3.6.2 การออกแบบหน้าเว็บไซต์ในการทำงานของผู้จัดการโปรเจ็ค (Project Manager)



ภาพประกอบที่ 3.41 หน้าการทำงานของผู้จัดการโปรเจ็ค แสดงโปรเจ็คที่จัดการ



- ภาพประกอบที่ 3.42 หน้าการทำงานของผู้จัดการโปรเจ็ค การสร้างโปรเจ็ค
- จากภาพประกอบที่ 3.42 เป็นหน้าการสมัครสมาชิกของพนักงาน ซึ่งจะประกอบไปด้วย
- date วันที่ ณ.ปัจจุบันที่สร้างโปรเจ็ค
 - project name คือ ช่องที่ต้องกรอกชื่อโปรเจ็ค
 - details คือ ช่องที่ต้องกรอกรายละเอียดเพิ่มเติมของโปรเจ็ค
 - project manager คือ ชื่อผู้สร้างโปรเจ็ค
 - tester คือ ช่องที่ต้องกรอกชื่อนักทดสอบระบบ
 - developer คือ ช่องที่ต้องกรอกชื่อนักพัฒนาระบบ

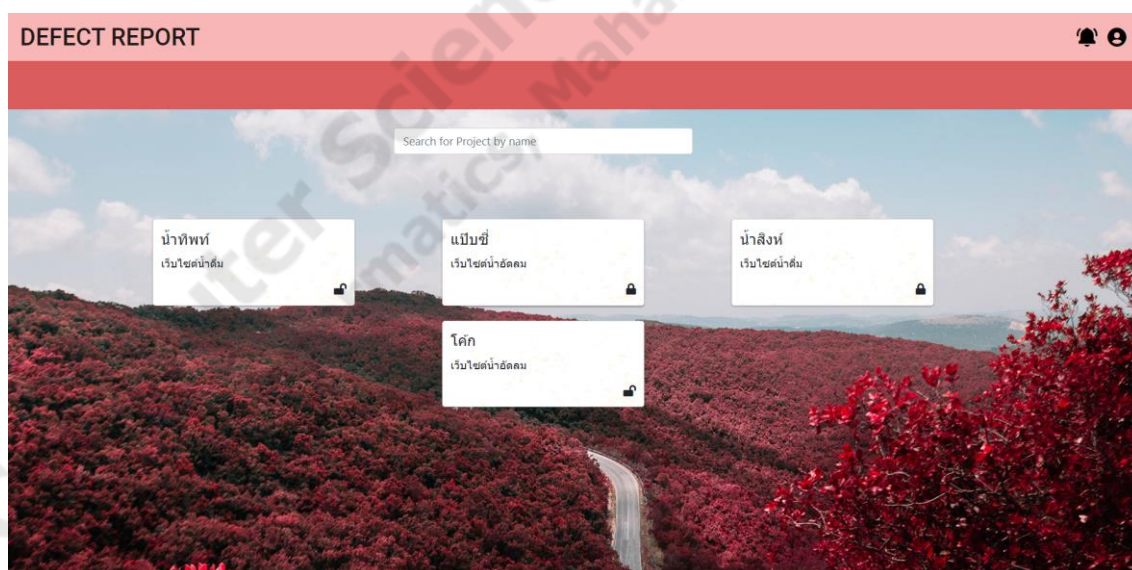


ภาพประกอบที่ 3.43 หน้าการทำงานของผู้จัดการโปรเจ็ค การแก้ไขข้อมูลโปรเจ็ค

จากภาพประกอบที่ 3.43 เป็นหน้าการแก้ไขข้อมูลโปรเจ็ค การเพิ่ม-ลบสมาชิก และการปิดโปรเจ็ค ซึ่งจะประกอบไปด้วย

- Date คือ วันที่สร้างโปรเจ็ค
- Project name คือ ชื่อโปรเจ็ค
- Project manager คือ ชื่อผู้จัดการโปรเจ็ค
- Add member คือ ช่องที่ต้องกรอกชื่อสมาชิกที่ต้องการเพิ่ม
- Select position คือ ช่องที่ต้องเลือกตำแหน่งให้สมาชิก
- Username คือ ช่องแสดง username
- Username คือ ช่องแสดงชื่อ และนามสกุล
- Position คือ ช่องแสดงตำแหน่งงาน
- Remove คือ ปุ่มที่ต้องการลบสมาชิกออกจากทีม
- ปิดโปรเจ็ค คือ ปุ่มที่ต้องการปิดโปรเจ็ค

3.6.3 การออกแบบหน้าเว็บไซต์ในการทำงานของนักทดสอบระบบ (Tester) และนักพัฒนาระบบ (Developer)



ภาพประกอบที่ 3.44 หน้าการทำงานของนักทดสอบระบบ และนักพัฒนาระบบ แสดงโปรเจ็คที่เป็นสมาชิกอยู่

The screenshot shows a web application interface for defect reporting. The main header is 'DEFECT REPORT'. On the left, there is a sidebar with navigation options: 'หน้าศัพท์', 'board', 'create defect', 'testcase', and 'Exit Project'. The main content area is titled 'Create testcase' and contains a table with columns 'No.TS' and 'TestModuleName'. A modal window titled 'Add testcase' is overlaid on the table, containing the following fields:

- Module Name :
- Test Data :
- Test Step :
- Expected Result :
- Remark :

At the bottom of the modal, there are 'Submit' and 'Cancel' buttons.

ภาพประกอบที่ 3.45 หน้าการทำงานของนักทดสอบระบบ การสร้าง Testcase จากภาพประกอบที่ 3.45 เป็นหน้าการสร้างข้อผิดพลาด ซึ่งจะประกอบไปด้วย

- Test Module Name คือ ช่องที่ต้องกรอกชื่อฟังก์ชัน
- Test Data คือ ช่องที่ต้องกรอกข้อมูลที่ใช้ทดสอบ
- Test Steps คือ ช่องที่ต้องกรอกขั้นตอนการทดสอบ
- Expected Result คือ ช่องที่ต้องกรอกผลลัพธ์ที่คาดหวัง
- Remark คือ ช่องที่ต้องกรอกหมายเหตุ

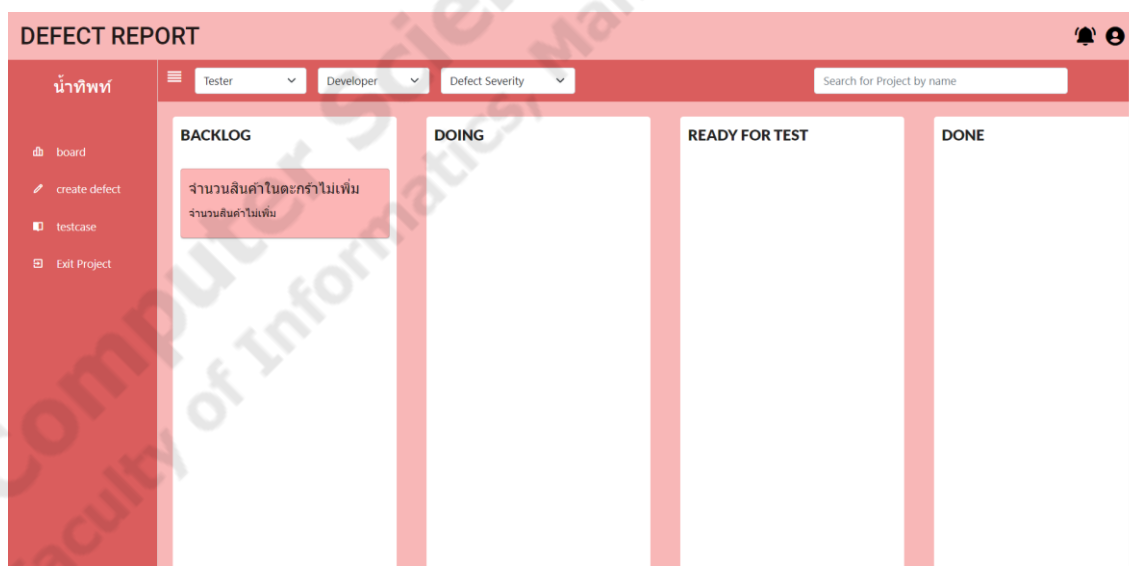
The screenshot shows the 'Create New Defect' form in the web application. The form is divided into several sections:

- Header:** Date : 28/12/2020, ID Report : #120, Project name : Numtip, Tester : เรณูภา มุศรดา
- Title:**
- Environment Details:**
- Step to reproduce:**
- Actual Result:**
- Expected Result:**
- Attachment:** No file chosen
- Metadata:**
 - Status :
 - Severity :
 - Due Date :
 - Assignees :

At the top right of the form, there is an 'Add Defect' button.

ภาพประกอบที่ 3.46 หน้าการทำงานของนักทดสอบระบบ การสร้างข้อผิดพลาด จากภาพประกอบที่ 3.46 เป็นหน้าการสร้างข้อผิดพลาด ซึ่งจะประกอบไปด้วย

- Date คือ วันที่สร้างโปรเจ็ค
- Id report คือ รหัสโปรเจ็ค
- Project name คือ ชื่อโปรเจ็ค
- Tester คือ ชื่อนักทดสอบระบบที่สร้างข้อผิดพลาด
- Title คือ ช่องที่ต้องกรอกหัวข้อของข้อผิดพลาด
- Environment Details คือ ช่องที่ต้องกรอกรายละเอียดของสภาพแวดล้อมระบบที่เจอปัญหา
- Step To Reproduce คือ ช่องที่ต้องกรอกขั้นตอนการทำให้เกิดปัญหา
- Actual Result คือ ช่องที่ต้องกรอกผลลัพธ์ที่เกิดขึ้น
- Expected Result คือ ช่องที่ต้องกรอกผลลัพธ์ที่คาดหวังตาม Requirement
- Attachment คือ ช่องที่ต้องเลือกเอกสารแนบ ได้แก่ รูปภาพ วิดีโอ และไฟล์เอกสาร
- Status คือ ช่องที่ต้องเลือกสถานะของการทำงาน
- Severity คือ ช่องที่ต้องเลือกระดับความรุนแรง
- Due Date คือ ช่องที่ต้องเลือกวันที่วันสุดท้ายของการทำงาน
- Assignees คือ ช่องที่ต้องเลือกกว่าให้นักพัฒนาคคนใดมาแก้ไข



ภาพประกอบที่ 3.47 หน้าการทำงานของนักทดสอบระบบ และนักพัฒนาระบบ แสดงข้อมูลของงานที่ต้องทำ