

ใบตรวจโปรแกรมวิชา SE (มีเครื่องหมาย X คือไม่ครบ = Incomplete = ติด I) ####
หมายเหตุ การตรวจจะหยุดทันทีเมื่อเจอจุดผิด หรือ ไม่ครบถ้วน

รหัส _____ ชื่อ _____ กลุ่มเรียน _____
โครงการกลุ่มที่ _____ ชื่อระบบ _____

- ☐ คนที่ commit แล้วทำให้งานของสมาชิกในกลุ่มพังจะติด I ทันที
- ☐ รันโปรแกรมได้ด้วยตนเอง
- ☐ รัน Integration Test ได้ด้วยตนเอง
- ☐ รัน Selenium ได้ด้วยตนเอง
- ☐ สามารถเปิดโค้ดที่ถูกลบเพื่อตอบคำถามได้
- ☐ สามารถตอบคำถามของระบบตัวเองได้

Git

- ☐ มี User story ที่ทำด้วยตัวเอง
- ☐ มีการแบ่ง Task และ commit ด้วยตัวเอง
- ☐ Task เลื่อนไป Done แบบ auto โดยการ commit ด้วยตัวเอง
- ☐ มี Code ของโปรแกรมใน Git และหมายเลข commit ตรงกับในเครื่อง
- ☐ มี Integration ใน Git
- ☐ มี Selenium ใน Git

Sprint#1

Program

- ☐ มี Entity Class อย่างน้อย 4 class
- ☐ มีการใช้ Data type ที่เหมาะสม
- ☐ มีการโยงความสัมพันธ์ อย่างน้อย 3 เส้น
- ☐ Entity Class ตรงตาม Class diagram
- ☐ Data type ตรงตาม Class diagram
- ☐ ความสัมพันธ์ ตรงตาม Class diagram
- ☐ การทำงานของโปรแกรมตรงตาม Activity diagram
- ☐ Controller มีการดึง object, โยง object และบันทึก object ตรงตาม Communication diagram และไม่ hard-code
- ☐ การเปลี่ยนแปลงของข้อมูลเกิดขึ้นจริงใน Database ตามการออกแบบ

Source Code Program (ตรวจที่หลังใน Git)

- ☐ ไม่มีการ hard-code

Integration Test

- ☐ เขียนใน Spring Boot มีการตรวจสอบครบทุกฟิลด์ และไม่ต่ำกว่า 5 ฟิลด์
- ☐ Entity Class มีการใช้ Annotation สำหรับการตรวจสอบข้อมูล อย่างน้อย 4 แบบ

Selenium Test

- ☐ มี Selenium Test Case ทุกหน้า UI
- ☐ มี Selenium Test Case กรณีใส่ข้อมูลถูกครบทุกหน้า
- ☐ มี Selenium Test Case กรณีใส่ข้อมูลไม่ถูกต้องครบทุกหน้า
- ☐ Selenium Test Case มีการ Assert ทุก Test Case
- ☐ Selenium Test Case มีการ Assert ข้อความหรือตำแหน่งบนหน้าจออย่างมีเหตุผล

Sprint#2

Program

- ☐ มี Entity Class เพิ่มอย่างน้อย 2 class
- ☐ มีการใช้ Data type ที่เหมาะสม
- ☐ มีการโยงความสัมพันธ์ เพิ่มอย่างน้อย 2 เส้น
- ☐ Entity Class ตรงตาม Class diagram
- ☐ Data type ตรงตาม Class diagram
- ☐ ความสัมพันธ์ ตรงตาม Class diagram
- ☐ การทำงานของโปรแกรมตรงตาม Activity diagram
- ☐ Controller มีการดึง object, โยง object และบันทึก object ตรงตาม Communication diagram และไม่ hard-code
- ☐ การเปลี่ยนแปลงของข้อมูลเกิดขึ้นจริงใน Database ตามการออกแบบ

Source Code Program (ตรวจที่หลังใน Git)

- ☐ ไม่มีการ hard-code

Integration Test

- ☐ เขียนใน Spring Boot มีการตรวจสอบครบทุกฟิลด์ และไม่ต่ำกว่า 5 ฟิลด์
- ☐ Entity Class มีการใช้ Annotation สำหรับการตรวจสอบข้อมูล อย่างน้อย 4 แบบ

Selenium Test

- ☐ มี Selenium Test Case ทุกหน้า UI
- ☐ มี Selenium Test Case กรณีใส่ข้อมูลถูกครบทุกหน้า
- ☐ มี Selenium Test Case กรณีใส่ข้อมูลไม่ถูกต้องครบทุกหน้า
- ☐ Selenium Test Case มีการ Assert ทุก Test Case
- ☐ Selenium Test Case มีการ Assert ข้อความหรือตำแหน่งบนหน้าจออย่างมีเหตุผล