ใบตรวจเล่มรายงานโครงงานวิชา SE 2/65 ** สำหรับผู้ตรวจ **(มีเครื่องหมาย X ในข้อ * = ติด F)**

รหัส..... ชื่อ..... กลุ่มเรียนที่.....

โครงงานกลุ่มที่..... ชื่อระบบหลัก..... ชื่อระบบย่อย.....

- ☐ * รันโปรแกรมได้ด้วยตนเอง
- ☐ * สามารถเปิดโค้ดที่ถูกถามเพื่อตอบคำถามได้ (3 นาที)
- ☐ มีระบบย่อย Sprint 1 เพิ่มข้อมูลเข้าระบบ
- ☐ มี Code ของโปรแกรมใน Git และหมายเลข commit ตรงกับโปรแกรมที่รันในเครื่อง
- ☐ * สามารถตอบคำถามของระบบตัวเองได้ (10 นาที)
- ☐ ตอบคำถามได้ถูกต้อง
- ☐ มีระบบย่อย Sprint 2 เพิ่มข้อมูลเข้าระบบ
- ☐ Git Status Clean

Source Code โปรแกรมที่ตรงกับการออกแบบในรายงาน

- ☐ ระบบต้องไม่ซ้ำและไม่คล้ายกับสมาชิกอื่นในกลุ่ม
- ☐ มี Source Code ของ UI (HTML)
- ☐ มี Source Code ที่เขียนด้วย React.js
- ☐ มี Source Code ที่เขียนด้วย Go / Gin / GORM
- ☐ UI เหมาะสมกับ Business (มี input ฟิลด์ ที่ไปในทิศทางเดียวกับระบบย่อย)
- ☐ โปรแกรมทำงานตรงกับ System Activity Diagram
- ☐ Go struct ตรงกับ Class Diagram
- ☐ Data Type ตรงกับ Class Diagram
- ☐ Data Type ของฟิลด์ (เลข, ข้อความ, วันที่, เวลา) ถูกต้องเหมาะกับชนิดของข้อมูล
- ☐ ไม่มีฟิลด์ที่ทำให้การใส่ข้อมูลเกิดความผิดพลาดซ้ำซ้อน (เช่น type, status)
- ☐ มี Entity อย่างน้อย 4 entity (1 หลัก + 3 สนับสนุน)
- ☐ มีการประกาศ field เพื่อโยงความสัมพันธ์ระหว่าง class ไม่ต่ำกว่า 3 เส้น (หรือเทียบเท่าในกรณีที่ระบบมี many-to-many โยงจาก Entity หลัก)
- ☐ ใน Entity หลักมีการประกาศ data type ของ field ข้อมูลที่แตกต่างกันอย่างน้อย 3 แบบ (เช่น string, int, date)
- ☐ ใน Entity หลัก มีการ validate ความถูกต้องของ field ข้อมูล อย่างน้อย 3 field ที่แบบของการ validate ไม่ซ้ำกัน
- ☐ ในหน้า UI หลัก มีการ validate ความถูกต้องของ field ข้อมูลอย่างน้อย 3 field ที่แบบของการ validate ไม่ซ้ำกัน
- ☐ ความสัมพันธ์แบบ 1 – Many ตอบโจทย์ของ Requirement หรือแบบ 1 – 1 ตอบโจทย์ของ Requirement
- ☐ มีโค้ด Go ของ Controller ที่สร้าง object ของ Entity หลัก
- ☐ มีโค้ด Go ของ Controller ที่ set ค่าให้ field ข้อมูลที่จำเป็น
- ☐ มีโค้ด Go ของ Controller ที่สั่งบันทึก object (save)
- ☐ โปรแกรมตรงกับ Communication Diagram (อนุโลมให้มีเฉพาะ Action หลักของระบบย่อย)
- ☐ ไม่มี hard-code ข้อมูลไว้ในหน้า UI / HTML
- ☐ ไม่มี hard-code ข้อมูลไว้ในหน้า React.js / TS / JS
- ☐ ไม่มี hard-code ข้อมูลไว้ในโปรแกรมภาษา Go ยกเว้นตัวโหลด database (Data Loader)
- ☐ ข้อมูลเกิดขึ้นจริงใน Database (ใน SQLite) ตามการออกแบบ

การทดสอบ

- ☐ มี Unit Test ของ Entity หลัก ทั้งแบบ positive (1 กรณี) และ negative (3 กรณี) ใน Git
- ☐ มี CI Pipeline สำหรับรัน Unit Test บน GitHub
- ☐ มี UAT (Selenium) ของ Entity หลักทั้งแบบ positive (1 กรณี) และ negative (3 กรณี) ใน Git

หมายเหตุ

ระดับความสมบูรณ์ของโปรแกรม (กรณีที่ผ่าน)

- ○ A) โปรแกรมสมบูรณ์ ตรงตาม user story
- ○ B) โปรแกรมตรง user story แต่ผิดพลาดบ้าง เล็กน้อย
- ○ C) โปรแกรม ผิดพลาดมาก หรือ ไม่ตรง user story
- ○ D) โปรแกรมพัง ทำงานไม่ได้

ผู้ตรวจ