

โครงการวิชา 01418211 Software Construction

ภาคต้น ปีการศึกษา 2565

อ่านรายละเอียดอย่างระมัดระวัง

ให้นักศึกษาสร้าง Desktop Application สำหรับการแจ้งเรื่องร้องเรียนของนิสิตมหาวิทยาลัยเกษตรศาสตร์ ด้วย JavaFX (JavaSE 17 เท่านั้น) โดยต้องออกแบบและเขียนโปรแกรมที่ใช้หลักการโปรแกรมเชิงวัตถุ

รายละเอียดโครงการ

1. โครงการเป็นงานกลุ่ม กลุ่มละไม่เกิน 4 คน
2. โปรแกรมนี้เป็นโปรแกรมสำหรับการแจ้งเรื่องร้องเรียนของนิสิตมหาวิทยาลัยเกษตรศาสตร์
3. โปรแกรมจะต้องมีหน้าจอหรือส่วนสำหรับเจ้าหน้าที่จัดการเรื่องร้องเรียนของนิสิต
4. โปรแกรมจะต้องมีหน้าจอหรือส่วนสำหรับนิสิตแจ้งเรื่องร้องเรียนและติดตามความคืบหน้าของเรื่องร้องเรียน
5. ต้องใช้หลักการโปรแกรมเชิงวัตถุในการเขียนโปรแกรม
6. มี User Interface ที่ง่ายต่อการใช้งาน มีการแสดงข้อมูลที่ชัดเจน ใช้ขนาดข้อความและขนาดภาพที่เหมาะสม สีสันง่าย อ่านได้ง่าย เข้าใจง่าย

เกณฑ์การตรวจโครงการ

นิสิตต้องทำโครงการทั้งหมดด้วยตนเอง หากพบหรือสงสัย/ส่อว่า

(1) นิสิตคัดลอกงานมาจากอินเทอร์เน็ต ไม่ว่าส่วนใดส่วนหนึ่งหรือทั้งหมด นิสิตจะได้คะแนนโครงการเป็น 0 และได้รับระดับคะแนน F (ตก)

(2) นิสิตคัดลอกงานจากเพื่อน ไม่ว่าส่วนใดส่วนหนึ่งหรือทั้งหมด ทั้งผู้ลอกและผู้ให้ลอกจะได้คะแนนโครงการเป็น 0 และได้รับระดับคะแนน F (ตก)

(3) หากนิสิตนำโครงชุดคำสั่งของเพื่อนกลุ่มอื่นมาปรับใช้ในงานของตน คะแนนจะถูกหักตามจำนวนนิสิต (ไม่ใช่จำนวนกลุ่มของนิสิต) ที่ใช้โครงเดียวกัน หรืออาจถูกพิจารณาว่าคัดลอกงานจากเพื่อน

* นิสิตทุกคนในกลุ่มจะต้องเขียนโปรแกรมอย่างมีนัยสำคัญ โดยมีประวัติการ commit ในระบบ git

กำหนดการตรวจโครงงาน

นิสิตแต่ละกลุ่ม จะต้องส่งคลิปวิดีโอ เพื่อแสดงความก้าวหน้าของการพัฒนาระบบแต่ละครั้งและเพื่อแสดงโครงงานที่สมบูรณ์ ในช่วงวันที่ให้กำหนดส่ง

การส่งโครงงานที่สมบูรณ์ ผู้ตรวจจะตรวจการทำงานของโปรแกรมผ่าน executable file เป็นหลัก โดยรันโปรแกรมผ่าน JRE (version JavaSE 17) และตรวจสอบการออกแบบโปรแกรมจากชุดคำสั่ง โดยทั้ง executable file, source code และข้อมูลสำหรับทดสอบระบบ จะต้องอยู่ใน github repository ที่กำหนด (Github Classroom)

กำหนดส่ง

ครั้งที่	วันที่	สิ่งที่ส่ง
1	25 ก.ค. - 12 ส.ค. 17:00 น.	ความก้าวหน้าของระบบ
2	22 ส.ค. - 9 ก.ย. 17:00 น.	ความก้าวหน้าของระบบ
3	12 ก.ย. - 30 ก.ย. 17:00 น.	ความก้าวหน้าของระบบ
4	3 ต.ค. - 21 ต.ค. 17:00 น.	โครงงานที่สมบูรณ์

* โครงงานที่สมบูรณ์ของนิสิตกลุ่มใดที่ไม่ผ่านเกณฑ์ที่กำหนด หรือไม่ส่งความก้าวหน้าของระบบครั้งใดก็ตาม (หรือส่งเลยกำหนดทุกกรณี) จะได้เกรด F (ตก)

เกณฑ์การตรวจโครงงานและให้คะแนน

รายละเอียดด้านล่าง คือ เกณฑ์การตรวจโครงงาน **โดยนิสิตจะต้องทำข้อที่มีสีแดงให้ได้** หากนิสิตไม่สามารถทำข้อที่มีสีแดงได้ นิสิตจะได้ระดับคะแนน I (ไม่สมบูรณ์) จนกว่านิสิตจะทำได้ **แต่ไม่เกินวันที่กำหนดส่งโครงงานที่สมบูรณ์** ส่วนข้ออื่น นิสิตจะได้คะแนนตามผลงานที่นิสิตทำมา หากทำได้นิสิตจะได้คะแนนในส่วนนั้น หากไม่ได้ทำหรือทำแต่ไม่สมบูรณ์ นิสิตจะไม่ได้คะแนนส่วนนั้นหรือได้คะแนนแค่บางส่วน

ข้อที่มีสีน้ำเงิน เป็นข้อพึงระวัง นิสิตอาจไม่ได้รับการตรวจหรือถูกหักคะแนนหรือมีผลให้ได้เกรด F

คะแนนที่มีคำว่า extra นำหน้า เป็นคะแนนพิเศษ

คะแนนเต็ม 200 คะแนน (คิดเป็น 25% ของคะแนนรายวิชา)

คะแนนพิเศษ +50 คะแนน (คิดเป็น +5% ของคะแนนรายวิชา)

* นิสิตแต่ละคนในกลุ่ม อาจได้คะแนนไม่เท่ากัน ขึ้นกับสัดส่วนของความร่วมมือในการพัฒนาโปรแกรม

1. (5 คะแนน) นิสิตจะต้อง commit โครงการ ไปไว้ที่ Github Classroom ที่กำหนด โดยแต่ละครั้งที่ส่งความก้าวหน้าของระบบต้องมีอย่างน้อย 1 commit จากนิสิตแต่ละคนในกลุ่ม
2. นิสิตต้องส่งความก้าวหน้าของระบบหรือโครงการที่สมบูรณ์ภายในวันที่กำหนดเท่านั้น หากไม่ส่งหรือส่งเลยกำหนด นิสิตทุกคนในกลุ่มจะได้เกรด F (ตก)
3. การส่งความก้าวหน้าของระบบหรือโครงการที่สมบูรณ์ครั้งใดก็ตาม นิสิตจะต้องมี commit ของการเขียนโปรแกรมอย่างมีนัยสำคัญ นิสิตคนใดที่ไม่มี commit หรือ commit นั้นไม่มีการเขียนโปรแกรมอย่างมีนัยสำคัญ เฉพาะนิสิตคนนั้นจะได้เกรด F (ตก) และไม่มีการจัดกลุ่มให้ใหม่

การส่งงานทุกครั้ง จะต้องมีความคืบหน้าที่ต่อเนื่องกัน

กลุ่มใดที่ทำ commit ใน repository หาย จะได้เกรด F

และความผิดพลาดที่เกิดขึ้นจากการใช้คำสั่ง git ไม่ถูกต้อง ใช้เป็นข้ออ้างในการไม่ส่งงานไม่ได้

4. ใน repository ครั้งที่ส่งโครงการที่สมบูรณ์ จะต้อง มีไฟล์ README.md ที่เขียนเรียบร้อย แสดงถึงความตั้งใจเขียน และมีเนื้อหาที่อธิบาย วิธีการติดตั้งหรือรันโปรแกรม การวางโครงสร้างไฟล์ (บอกว่า folder ใน repository แต่ละ folder เก็บอะไร) และสรุปสิ่งที่พัฒนาในแต่ละครั้งที่ส่งความก้าวหน้าของระบบ หรือครั้งที่ส่งโครงการที่สมบูรณ์
5. นิสิตต้องตั้งชื่อไฟล์ และ directory ให้ตรงกับบริบทของโครงการ หากไม่ตั้งชื่อตามบริบทของงาน จะถูกหักคะแนนอย่างน้อยจุดละ 10 คะแนน
6. (5 คะแนน) ต้องมีโปรแกรม executable file (.jar หรือ .sh) ที่ดับเบิลคลิกแล้วรันได้เลย หรือหากต้องรันด้วย command ต้องระบุ command ในไฟล์ README.md ในส่วน วิธีการติดตั้งหรือรันโปรแกรม หากผู้ตรวจไม่สามารถหา executable file ได้จาก repository หรือ README.md ไม่อธิบายวิธีการติดตั้ง โปรแกรมจนผู้ตรวจไม่สามารถเปิดโปรแกรมได้ โครงการของนิสิตจะไม่ได้รับการตรวจ และนิสิตทุกคนในกลุ่ม จะได้เกรด F (ตก)
7. *** ต้องใช้หลักการโปรแกรมเชิงวัตถุพื้นฐาน
 - 7.1. (5 คะแนน) ออกแบบคลาสให้สามารถสร้าง object ที่มีข้อมูลต่างกันผ่านการส่งค่าให้ parameter ใน constructor ได้ ดังนั้น นิสิตควรสร้าง object ที่เก็บข้อมูลต่างกัน โดยใช้คลาสเดียวกันได้
 - 7.2. (5 คะแนน) สร้าง attributes ได้ถูกต้อง ไม่ใช่ local variable มาเป็น attribute
 - 7.3. (5 คะแนน) ระบุ access modifier ที่เหมาะสม คลาสอยู่ใน package ที่เหมาะสม และไม่ใช่ static
 - 7.4. (5 คะแนน) ตั้งชื่อ class, method, ตัวแปร ให้สื่อถึงบริบทหรือการใช้งาน ถูกต้องตามหลักการตั้งชื่อ
8. ต้องเขียนโปรแกรมในรูปแบบของ Model View Controller (MVC) Architecture

- 8.1. (5 คะแนน) Model Class เก็บข้อมูลและ method ที่ทำงานในการแก้ไขข้อมูลของคลาส และ/หรือ คำนวณ/ประมวลผลกับข้อมูลของคลาส
- 8.2. (5 คะแนน) มี Collection Class เพื่อจัดการ collection ของ model object
- 8.3. ต้องไม่มีส่วนของ Controller หรือ View ใน Model Class
- 8.4. (5 คะแนน) Controller Class เป็นส่วนจัดการการรับข้อมูลจากผู้ใช้งาน การส่งค่าไปคำนวณ/ประมวลผล ที่ Model Class และการส่งค่าไปแสดงผลที่ View ดังนั้นจะต้องไม่มีการคำนวณค่าของข้อมูลใน model object เกิดขึ้นโดยตรงที่ Controller Class
- 8.5. View (JavaFX) เป็นส่วนการแสดงผลข้อมูลของโปรแกรม นิสิตจะเขียนเขียนไฟล์ .fxml หรือจะสร้าง component สำหรับการแสดงผลจากใน Controller Class ก็ได้
9. ใช้หลักการโปรแกรมเชิงวัตถุขั้นสูง
 - 9.1. (5 คะแนน) มี superclass/subclass ที่เหมาะสม หมายความว่า ควรตรงกับคุณสมบัติ IS-A เช่น Car เป็น subclass ของ Vehicle นั้นเหมาะสม (Car IS-A Vehicle) แต่ Book เป็น subclass ของ Library นั้นไม่เหมาะสม (Book IS-A Library?)
 - 9.2. อาจมีหรือไม่มี interface ก็ได้ ให้นิสิตพิจารณาตามความเหมาะสม และเขียนอย่างถูกต้อง
 - 9.3. (5 คะแนน) superclass และ subclass ต้องมีการทำงานบางอย่างที่เหมือนกัน และบางอย่างที่แตกต่างกันอย่างเหมาะสม โดย superclass ควรระบุ method ทั้งหมดที่ subclass ใช้ร่วมกัน เพื่อให้เกิด polymorphism ได้ และ subclass ควรออกแบบให้ดีเพื่อให้ตรงกับคุณสมบัติดังกล่าว หาก subclass ต่างจาก superclass แค่ constructor แสดงว่ายังออกแบบไม่ดี หรือหาก superclass ไม่มี method ใดเลย แต่ subclass มี method ที่แตกต่างกันไปหมด แสดงว่ายังออกแบบไม่ดี
 - 9.4. (5 คะแนน) เขียนโค้ดให้เห็นชัดเจนว่ามีการใช้งาน polymorphism ที่เหมาะสม โดยควรออกแบบคลาสที่ใช้ object ของ super type/sub type ให้สามารถใช้ object ของ sub type/super type เหล่านี้แทนได้
10. (5 คะแนน) มี UML Class Diagram ที่แสดงถึงคลาสในโปรแกรม โดยเก็บเป็นไฟล์รูป เช่น .png, .jpg
 - 10.1. UML Class Diagram 1 รูป สำหรับ Model Class ทั้งหมด
 - 10.2. UML Class Diagram 1 รูป สำหรับ Controller Class ทั้งหมด
 - 10.3. UML Class Diagram 1 รูป สำหรับ Utility Class ทั้งหมด
 - 10.4. ไม่ต้องใส่ UML Class Diagram ของคลาสที่ไม่ได้เขียนเอง
11. (5 คะแนน) ต้องมีไฟล์ pdf เพื่ออธิบายรายละเอียดและการใช้งานโปรแกรม โดยมีหัวข้อดังต่อไปนี้
 - 11.1. “วิธีการใช้งาน<ชื่อ application ของนิสิต>” เพื่ออธิบายวิธีการใช้งานโปรแกรมของนิสิต ซึ่งต้องมีการอธิบายอย่างเป็นขั้นตอน ใช้ภาษาที่เข้าใจง่าย และมีภาพประกอบ

- 11.2. “ตัวอย่างข้อมูลผู้ใช้งาน” เพื่อระบุตัวอย่างของ username และ password ในการเข้าใช้ระบบของผู้ใช้แต่ละ role
 - 11.3. “รายละเอียดของไฟล์ CSV” เพื่ออธิบายข้อมูลในไฟล์ .csv ว่า ไฟล์จะถูกสร้างไว้ที่ใด ข้อมูลที่เก็บในแต่ละ column ของไฟล์หมายถึงอะไร พร้อมยกตัวอย่างข้อมูล 1 แถวมาอธิบาย
 - 11.4. “Extra Features” หากนิสิตทำข้อ extra มาด้วย ให้ระบุด้วยว่าทำข้อใดบ้าง อยู่ส่วนใดของโปรแกรม พร้อมอธิบายด้วยว่าเพราะเหตุใดกลุ่มของนิสิตจึงควรจะได้คะแนนจากข้อ Extra นี้
 - 11.5. หากผู้ตรวจอ่านไฟล์ pdf แล้วไม่เข้าใจได้โดยง่าย จะหักคะแนนอย่างน้อย 30 คะแนน
 - 11.6. ตรวจสอบความถูกต้องของข้อความที่ปรากฏในไฟล์ pdf หากพบว่ามีคำผิด หักคำที่ผิดจุดละ 5 คะแนน (จุดที่แตกต่างกัน หากพบคำผิดเดียวกัน จะถือว่าเป็นคนละจุดผิด)
12. ความสวยงามและประสบการณ์ของผู้ใช้
- 12.1. (5 คะแนน) ใช้โทนสีและองค์ประกอบต่าง ๆ ของ GUI ที่ดูได้ชัดเจน นำใช้ไม่สับสน ไม่ลวงตา ทุกหน้าของโปรแกรมใช้โทนสีเดียวกัน ขนาดของตัวอักษรที่เหมาะสม ไม่เล็ก ไม่ใหญ่เกินไป โดยลองนึกว่าหากเราทำโปรแกรมนี้นี้ให้คนทั่วไปใช้จริง ควรแสดงองค์ประกอบเหล่านี้อย่างไร
 - 12.2. ตรวจสอบความถูกต้องของข้อความที่ปรากฏในส่วนต่าง ๆ ของโปรแกรม หากพบว่ามีคำผิด หักคำที่ผิดจุดละ 5 คะแนน (จุดที่แตกต่างกัน หากพบคำผิดเดียวกัน จะถือว่าเป็นคนละจุดผิด)
 - 12.3. (10 คะแนน) Graphic User Interface (GUI) ควรมีรูปแบบที่เข้าใจง่าย มีการแสดงข้อมูลที่ชัดเจน มี Label กำกับข้อมูลที่ชัดเจน เพื่อให้เข้าใจว่าค่าที่แสดงนั้นคือค่าอะไร หรือค่าที่ต้องการจากผู้ใช้คือค่าอะไร มีการแสดงองค์ประกอบของโปรแกรมในขนาดที่เหมาะสม ไม่เล็กหรือไม่ใหญ่จนเกินไป ขนาดหน้าจอของโปรแกรมต้องมีความสูงไม่เกิน 1024 pixel มีความกว้างของหน้าจอโปรแกรมเหมาะสมสำหรับผู้ใช้นิตบุ๊กทั่วไป แสดงข้อความแนะนำผู้ใช้ที่ชัดเจนหากผู้ใช้ใช้งานโปรแกรมไม่ถูกต้อง
 - 12.4. (Extra 5 คะแนน) GUI มี effect เคลื่อนไหวที่น่าสนใจ เมื่อมี action ต่าง ๆ อย่างเหมาะสม
 - 12.5. (Extra 10 คะแนน) ผู้ใช้โปรแกรมสามารถปรับเปลี่ยน Theme ของ Application ได้ เช่น เปลี่ยนโทนสีของโปรแกรม เปลี่ยนขนาดอักษร หรือเปลี่ยนฟอนต์ของตัวอักษรได้ โดยจะต้องเปลี่ยนให้สอดคล้องกันทุกหน้า

Features ของ Application มีดังนี้

13. (5 คะแนน) ต้องมีส่วนของ GUI ที่แสดงถึงข้อมูลต่อไปนี้ โดยเข้าถึงข้อมูลนี้ผ่านเมนู หรือมีปุ่มกดที่หน้าแรก

13.1. ข้อมูลนิสิตผู้จัดทำโปรแกรม ได้แก่ รูปที่แสดงหน้าตา (หน้ายิ้ม) ที่ชัดเจน ชื่อ นามสกุล ชื่อเล่น และรหัสนิสิต

13.2. ข้อมูลคำสั่งหรือคำแนะนำในการใช้โปรแกรมที่นิสิตสร้างขึ้นมา

14. ระบบของผู้ดูแลระบบ ระบบส่วนนี้ถูกจำกัดสิทธิ์สำหรับผู้ดูแลระบบเท่านั้น ผู้ใช้ที่ไม่ใช่ผู้ดูแลระบบต้องเข้าใช้งานส่วนนี้ไม่ได้

* โปรแกรมไม่ต้องมีระบบการสร้างบัญชีสำหรับผู้ดูแลระบบ ให้นิสิตระบุ username และ password ของผู้ดูแลระบบในไฟล์ pdf (ข้อ 11.2)

14.1. (5 คะแนน) ผู้ดูแลระบบสามารถเปลี่ยนรหัสผ่านของตนเองได้ โดยต้องระบุ username และ password เดิมที่ถูกต้องด้วย และรหัสผ่านใหม่ต้องใช้ได้

14.2. (5 คะแนน) มีหน้าแสดงรายชื่อของผู้ใช้ระบบ โดยต้องแสดงภาพผู้ใช้ username ชื่อของผู้ใช้ระบบ ชื่อหน่วยงาน (กรณีผู้ใช้ระบบเป็นเจ้าหน้าที่) และวันเวลาที่เข้าใช้ล่าสุดของผู้ใช้ระบบคนนั้น เรียงลำดับตามวันเวลาที่ผู้ใช้ระบบเข้าใช้งานล่าสุดก่อน

14.3. (10 คะแนน) มีหน้าแสดงรายการของรายงานความไม่เหมาะสมของผู้ใช้ระบบ (เช่น มีเนื้อหาไม่เหมาะสม ข่าวดปลอม เนื้อหาล่อแหลม เนื้อหาที่มีความรุนแรง ฯลฯ)

14.3.1. ผู้ดูแลระบบสามารถระงับสิทธิ์การเข้าใช้งานของผู้ใช้ระบบได้

14.3.2. ผู้ใช้ระบบที่ถูกระงับสิทธิ์จะไม่สามารถเข้าใช้งานโปรแกรมได้ โดยมีข้อความแจ้งเตือนการถูกระงับสิทธิ์เมื่อเข้าสู่ระบบ

14.3.3. ผู้ดูแลระบบสามารถทราบถึงจำนวนครั้งในการพยายามเข้าใช้โปรแกรมในขณะที่ผู้ใช้ระบบถูกระงับสิทธิ์

14.3.4. ผู้ใช้ที่ถูกระงับสิทธิ์ สามารถขอคืนสิทธิ์ โดยแจ้ง username พร้อมเหตุผล การขอคืนสิทธิ์ให้ผู้ดูแลระบบทราบ ผ่านหน้าจอโปรแกรม

14.3.5. ผู้ดูแลระบบสามารถคืนสิทธิ์การเข้าใช้งานของผู้ใช้ระบบได้

14.4. (5 คะแนน) มีหน้าสำหรับสร้างบัญชีผู้ใช้ของเจ้าหน้าที่ โดยระบุ ชื่อสำหรับเข้าสู่ระบบ (username) รหัสผ่าน ยืนยันรหัสผ่าน ชื่อของเจ้าหน้าที่ ภาพของเจ้าหน้าที่ และหน่วยงานที่รับผิดชอบ โดยมีการตรวจสอบ username จะต้องไม่ซ้ำกับผู้ใช้ระบบที่มีอยู่แล้ว

14.5. (extra 5 คะแนน) มีเมนูจัดการหน่วยงานของเจ้าหน้าที่

14.5.1. ผู้ดูแลระบบสามารถสร้างหน่วยงานใหม่ หรือแก้ไขชื่อหน่วยงานเดิมได้

14.5.2. ผู้ดูแลระบบสามารถกำหนดได้ว่าหน่วยงานอยู่ในความรับผิดชอบของเจ้าหน้าที่คนใด

14.5.3. หากการจัดการหน่วยงานเป็นแบบ hard code

หรือไม่สามารถเพิ่มหมวดหมู่ใหม่ได้จากในโปรแกรม จะไม่ได้คะแนน extra ส่วนนี้

14.6. (extra 5 คะแนน) มีเมนูจัดการหมวดหมู่เรื่องร้องเรียน

14.6.1. ผู้ดูแลระบบสามารถสร้างหมวดหมู่เรื่องร้องเรียนได้

14.6.2. หมวดหมู่เรื่องร้องเรียนที่แตกต่างกัน มีคุณลักษณะของเรื่องร้องเรียนที่แตกต่างกัน

เช่น หมวดหมู่การจัดการกิจกรรมไม่เหมาะสม จะให้เพิ่มคุณลักษณะ

ชมรม/หน่วยงานที่จัดกิจกรรม ชื่อ/กลุ่มผู้ดูแลกิจกรรม แต่หมวดหมู่อาคารสถานที่ชำรุด

จะให้เพิ่มคุณลักษณะ สถานที่ที่เกิดความเสียหาย ภาพประกอบของความเสียหาย เป็นต้น

14.6.3. หากการจัดการหมวดหมู่เรื่องร้องเรียนเป็นแบบ hardcode

หรือไม่สามารถเพิ่มหมวดหมู่เรื่องร้องเรียนใหม่ได้จากในโปรแกรม จะไม่ได้คะแนน extra ข้อ

14.6 นี้

14.7. คุณลักษณะของเรื่องร้องเรียนภายในหมวดหมู่

14.7.1. (extra 5 คะแนน) คุณลักษณะที่แตกต่างกันที่จัดการในข้อ 14.6.2

สามารถเพิ่มตัวเลือกของข้อมูลได้ โดยแสดงตัวเลือกของข้อมูลสำหรับคุณลักษณะนี้ (ข้อ

16.1) แบบ dropdown

14.7.2. (extra 15 คะแนน) คุณลักษณะที่แตกต่างกันที่จัดการในข้อ 14.6.2

สามารถเพิ่มคุณลักษณะประเภทที่ต้องการให้ upload ภาพ โดยการแสดงผล (ข้อ 16.1)

จะต้องให้ผู้ใช้งานระบบ upload รูปภาพได้ และต้องแสดงภาพในรายละเอียดของเรื่องร้องเรียน
ได้ถูกต้อง

14.7.3. จะต้องได้คะแนน extra จากข้อ 14.6 ก่อน และต้องไม่ใช่ hard code จึงจะพิจารณา

คะแนน extra ข้อ 14.7 นี้

15. การสร้างบัญชีของผู้ใช้ระบบ

15.1. (5 คะแนน) มีระบบลงทะเบียน (registration) สำหรับผู้ใช้ระบบ โดยผู้ใช้ระบบที่ลงทะเบียนนี้

จะเป็นนิสิตเท่านั้น ข้อมูลที่ใช้ในการลงทะเบียน ได้แก่ ชื่อสำหรับเข้าสู่ระบบ (username) รหัสผ่าน
ยืนยันรหัสผ่าน ชื่อของผู้ใช้ระบบ โดยมีการตรวจสอบ username ต้องไม่ซ้ำกับผู้ใช้ระบบที่มีอยู่แล้ว

15.2. (5 คะแนน) ผู้ใช้ระบบสามารถเปลี่ยนรหัสผ่านของตนเองได้ โดยต้องระบุ username และ password

เดิมที่ถูกต้องด้วย และรหัสผ่านใหม่ต้องใช้ได้

15.3. (5 คะแนน) ผู้ใช้ระบบ upload รูปภาพ เพื่อใช้เป็นภาพของผู้ใช้ระบบ และสามารถเปลี่ยนรูปภาพได้

หากผู้ใช้ระบบยังไม่กำหนดภาพ ระบบจะให้ใช้ภาพ default แทนไปก่อน

16. ระบบสำหรับนิสิต

* ผู้ใช้ระบบที่ลงทะเบียนตามข้อ 15.1 จะเป็นนิสิตโดยปริยาย

16.1. (5 คะแนน) มีหน้าแจ้งเรื่องร้องเรียนภายในมหาวิทยาลัย ข้อมูลเรื่องร้องเรียนประกอบด้วย หัวเรื่อง รายละเอียดเรื่องร้องเรียน หมวดหมู่เรื่องร้องเรียน และข้อมูลที่ต้องการในหมวดหมู่เรื่องร้องเรียนนั้น

16.2. มีหน้ารวมเรื่องร้องเรียนทั้งหมดจากทุกหมวดหมู่ โดยแยกเรื่องร้องเรียนที่ถูกจัดการแล้ว อยู่ระหว่างดำเนินการ และยังไม่ถูกจัดการ

16.2.1. (5 คะแนน) มีส่วนให้ผู้ใช้ระบบเลือกเรียงลำดับเรื่องร้องเรียนจากเวลาที่แจ้งล่าสุด หรือเก่าสุดขึ้นแสดงก่อน

16.2.2. (5 คะแนน) มีส่วนให้ผู้ใช้ระบบเลือกเรียงลำดับเรื่องร้องเรียนจากคะแนนโหวต มากที่สุด หรือน้อยที่สุดขึ้นแสดงก่อน

16.2.3. ผู้จัดทำโปรแกรมจะต้องเป็นผู้เขียน logic การเรียงลำดับเอง จึงจะได้คะแนน

16.3. ในหน้ารวมเรื่องร้องเรียนทั้งหมด (ข้อ 16.2) มีส่วนให้ผู้ใช้ระบบเลือกการแสดงผลที่สนใจ

16.3.1. (5 คะแนน) เลือกการแสดงผลเฉพาะหมวดหมู่เรื่องร้องเรียนที่สนใจ โดยยังคงเรียงลำดับการแสดงผลจากข้อ 16.2.1 หรือข้อ 16.2.2 ได้

16.3.2. (5 คะแนน) เลือกการแสดงผลเฉพาะเรื่องร้องเรียนที่มีช่วงคะแนนโหวตที่ผู้ใช้ระบบ กำหนดเองมากกว่า 1 รูปแบบ เช่น
รูปแบบที่ 1 ผู้ใช้เลือกแสดงเรื่องร้องเรียนที่มีมากกว่า x คะแนนโหวต หรือ
รูปแบบที่ 2 ผู้ใช้เลือกแสดงเรื่องร้องเรียนที่มีตั้งแต่ m – n คะแนนโหวต
โดยยังคงเรียงลำดับการแสดงผลจากข้อ 16.2.1 หรือข้อ 16.2.2 ได้

16.4. ผู้ใช้ระบบเข้าดูรายละเอียดของเรื่องร้องเรียนจากหน้ารวมเรื่องร้องเรียนทั้งหมดได้

16.4.1. (5 คะแนน) รายละเอียดของเรื่องร้องเรียนจะต้องแสดงข้อมูลครบถ้วน ตามที่
หมวดหมู่ของเรื่องร้องเรียนนั้นกำหนด โดยไม่แสดงคุณสมบัติอื่นที่ไม่เกี่ยวข้องกับ
หมวดหมู่ของเรื่องร้องเรียนนั้น แสดงคะแนนโหวตของเรื่องร้องเรียนนั้น

16.4.2. (5 คะแนน) ในหน้ารายละเอียดของเรื่องร้องเรียน มีส่วนที่ให้ผู้ระบบแจ้งรายงาน
ความไม่เหมาะสมของเนื้อหา หรือของผู้ใช้ระบบ ซึ่งจะไปแสดงผลในข้อ 14.3
โดยต้องระบุสิ่งที่รายงาน (รายงานเนื้อหาหรือรายงานผู้ใช้ระบบ)
ประเภทของความไม่เหมาะสม และข้อมูลเพิ่มเติม

16.4.2.1. ผู้ดูแลระบบสามารถพิจารณาลบเนื้อหาได้ หากเป็นการรายงานเนื้อหา

16.4.2.2. ผู้ดูแลระบบสามารถพิจารณาระงับสิทธิ์การใช้งานของผู้ใช้ได้
หากเป็นการรายงานผู้ใช้

- 16.4.2.3. หากไม่มีการรายงาน ผู้ดูแลระบบจะลบเนื้อหา หรือระงับสิทธิการใช้งานของผู้ใช้ไม่ได้
- 16.4.3. (5 คะแนน) ในหน้ารายละเอียดของเรื่องร้องเรียน มีส่วนให้ผู้ใช้ระบบสามารถให้คะแนนโหวตได้ 1 คะแนนโหวตต่อครั้ง
 - 16.4.3.1. (extra 5 คะแนน) ผู้ใช้ระบบให้คะแนนโหวตในเรื่องร้องเรียนเดิมซ้ำไม่ได้
- 16.5. (5 คะแนน) มีหน้ารวมเรื่องร้องเรียนเฉพาะเรื่องร้องเรียนที่ผู้ใช้ระบบที่กำลัง login อยู่เป็นผู้แจ้งเรื่องร้องเรียน ซึ่งแสดงสถานะการจัดการเรื่องร้องเรียน และสามารถเข้าดูรายละเอียดของเรื่องร้องเรียน
- 17. ระบบสำหรับเจ้าหน้าที่
 - 17.1. (5 คะแนน) เจ้าหน้าที่สามารถเปลี่ยนรหัสผ่านของตนเองได้ โดยต้องระบุ username และ password เดิมที่ถูกตัดด้วย และรหัสผ่านใหม่ต้องใช้ได้
 - 17.2. มีหน้าจัดการเรื่องร้องเรียน เฉพาะหมวดหมู่ที่ตนเองรับผิดชอบ และไม่สามารถจัดการเรื่องร้องเรียนของหมวดหมู่อื่นที่ไม่อยู่ในความรับผิดชอบ
 - 17.2.1. (5 คะแนน) หน้าจัดการเรื่องร้องเรียนนี้ แสดงรายการเรื่องร้องเรียนทั้งหมดในหมวดหมู่ที่ตนเองรับผิดชอบ และแสดงสถานะว่าจัดการไปแล้วหรือยัง
 - 17.2.2. (5 คะแนน) สามารถเลือกแสดงรายละเอียดของเรื่องร้องเรียนในหมวดหมู่ที่ตนเองรับผิดชอบ ในหน้าแสดงรายละเอียดมีส่วนให้จัดการเรื่องร้องเรียน โดยต้องระบุรายละเอียดของวิธีการจัดการ และเลือกสถานะได้ว่าอยู่ระหว่างดำเนินการหรือเสร็จสิ้นแล้ว
 - * สถานะของเรื่องร้องเรียน และรายละเอียดของวิธีการจัดการ จะต้องแสดงในหน้ารายละเอียดของเรื่องร้องเรียนให้ผู้ใช้ทั่วไปเห็นได้
 - 17.2.3. รายละเอียดของเรื่องร้องเรียนต้องไม่แสดงชื่อของผู้ร้องเรียน แต่แสดงหน่วยงานของเจ้าหน้าที่ที่จัดการเรื่องร้องเรียน โดยไม่แสดงชื่อเจ้าหน้าที่ให้ผู้ใช้ทั่วไปเห็น แต่ (5 คะแนน) แสดงชื่อเจ้าหน้าที่ที่จัดการเรื่องร้องเรียนให้เจ้าหน้าที่ในหน่วยงานเดียวกันเห็น
- 18. โปรแกรมต้องบันทึกค่าของข้อมูลต่าง ๆ ในรูปแบบของไฟล์ csv (comma-separated-values) และสามารถโหลดไฟล์ที่บันทึกไว้นั้นมาแสดงผลในโปรแกรมได้อย่างถูกต้อง
 - 18.1. ต้องมีการบันทึกค่าของข้อมูลที่จำเป็นในรูปแบบไฟล์ csv
 - 18.2. เมื่อเปิดโปรแกรมใหม่ หรือเมื่อโปรแกรมโหลดไฟล์ csv จะต้องแสดงข้อมูลที่ได้นับบันทึกไว้อย่างถูกต้อง
- 19. ต้องมีข้อมูลเริ่มต้นในการทดสอบโปรแกรม เสมือนว่าโปรแกรมถูกใช้งานมาประมาณ 3 เดือนเป็นอย่างน้อย เช่น

- 19.1. มีข้อมูลผู้ดูแลระบบ และข้อมูลเจ้าหน้าที่อย่างน้อย 5 หน่วยงาน หน่วยงานละอย่างน้อย 3 บัญชีผู้ใช้
 - 19.2. มีข้อมูลหมวดหมู่เรื่องร้องเรียนอย่างน้อย 3 หมวดหมู่ ที่มีคุณลักษณะที่แตกต่างกัน
 - 19.3. มีข้อมูลผู้ใช้ระบบอย่างน้อย 10 บัญชี แต่ละคนมีภาพที่แตกต่างกัน
 - 19.4. ผู้ใช้ระบบอย่างน้อย 3 บัญชี แจ้งเรื่องร้องเรียนอย่างน้อยคนละ 3 หมวดหมู่ หมวดหมู่น้อย
5 เรื่องร้องเรียน
 - 19.5. หากมีข้อมูลเริ่มต้นในการทดสอบโปรแกรมน้อยกว่าที่กำหนด จะไม่ตรวจให้
และถือว่าไม่ผ่านเกณฑ์โครงการ
20. การทำงานทุก features ของโปรแกรมจะต้องสมบูรณ์ในตัวโปรแกรม
- การแก้ไขโค้ดหรือการแก้ไขข้อมูลในไฟล์ csv เพื่อให้โปรแกรมทำงานถูกต้อง
ไม่ถือว่าเป็นการทำงานของโปรแกรมที่สมบูรณ์

ขั้นตอนการเริ่มต้นโครงการ

1. ตัวแทนสมาชิกในกลุ่ม 1 คน กรอกข้อมูลในฟอร์ม <https://forms.office.com/r/FvnWGY56T7> (ใช้ account @live.ku.th) เพื่อระบุรายชื่อสมาชิกในกลุ่ม และชื่อกลุ่ม พร้อมรับทราบเงื่อนไขการได้เกรด F ภายในวันอาทิตย์ที่ 24 กรกฎาคม 2565 เวลา 17.00 น. นิสิตที่ไม่กรอกข้อมูลภายในเวลาที่กำหนด ให้ถอนการลงทะเบียน หรือได้เกรด F ไปโดยปริยาย
2. ตัวแทนสมาชิกในกลุ่ม 1 คน สร้าง repository เริ่มต้นที่ <https://classroom.github.com/a/tXO-ATi4> โดยระบุชื่อกลุ่มให้ตรงกับที่กรอกข้อมูลในฟอร์ม
3. หลังจากนั้นสมาชิกที่เหลือในกลุ่ม เข้าไปที่เดียวกัน <https://classroom.github.com/a/tXO-ATi4> เลือกกลุ่มตามชื่อที่ตกลงกันได้
4. สมาชิกในกลุ่ม 1 คน join MS Teams ด้วยโค้ด `i9xcxcg` เพื่อส่งงาน และเป็นตัวแทนในการส่งคลิปความก้าวหน้าของระบบ และคลิปโครงการที่สมบูรณ์ สมาชิกที่เหลือในกลุ่ม ไม่ต้อง join (หาก join มา จะถือว่าเป็นอีกกลุ่ม และถ้าไม่มีคลิปส่ง จะถือว่าไม่ได้ส่งทั้งกลุ่ม และได้ F ทั้งกลุ่ม แม้ว่าสมาชิกอีกคนจะส่งก็ตาม) สมาชิกที่ส่งคลิปความก้าวหน้าของระบบและคลิปโครงการที่สมบูรณ์ จะต้องเป็นคนเดิม ไม่อนุญาตให้เปลี่ยนคน หากสมาชิกคนนี้อถอนการลงทะเบียน จะต้องถอนการลงทะเบียนทั้งกลุ่ม
5. Clone repository ลงมาในเครื่อง แล้วเริ่มเขียนโปรแกรมได้

การส่งงานเพื่อนำเสนอความก้าวหน้า

1. สมาชิกทุกคนในทีม ต้องมี commit งานในส่วนของตนเองอย่างมีนัยสำคัญ ซึ่งหมายถึงการพัฒนาบาง feature ที่ทำได้อย่างสมบูรณ์ด้วยตนเอง และเข้าใจในสิ่งที่ตนเองเขียน
2. แยกส่วนของ executable files + csv ออกจาก directory target และสร้าง directory ใหม่ เพื่อใช้ในการส่ง executable files + csv
3. push ทุกอย่างที่จะส่งผ่านระบบ git เข้าไปที่ GitHub Classroom Repository ของกลุ่มตนเองก่อนการส่งคลิป
4. อัปเดตวิดีโอ ด้วยการบันทึกหน้าจอความคืบหน้าของโปรแกรม แสดงให้เห็นว่าโปรแกรมทำงานได้ตามความคืบหน้า และแสดงโค้ดในส่วนที่ตนเองเขียน พร้อมพูดบรรยายจากสมาชิกในกลุ่ม และแสดงใบหน้าของผู้พูดในคลิปขณะพูดบรรยายประกอบ (ใครทำส่วนไหน ก็ควรบรรยายส่วนนั้น)
5. ส่งคลิปวิดีโอใน Assignment ใน Microsoft Teams ที่กำหนด ภายในเวลาที่กำหนด