

FRA241 SRS Report

Grading Platform for Programming System

<u>CPO</u>	นายธรรมวัช	ทอสุวรรณ	59340500023
<u>CTO</u>	นายตรีเทพ	พรหมวิเศษ	59340500020
	นายธีรวัชร	อภิชนาพงศ์	59340500027
	นายปรเมศร์	เฟื่องพานิชเจริญ	59340500034
	นายวฤกษ์	อนันตประยูร	59340500051
	นายอุกฤษฏ์	เลิศวรรณการ	59340500062
	นายชวลิตวิทย์	ณรงค์พันธ์	59340500063

Introduction

ระบบ Grading Platform for Programming

คือระบบที่จะช่วยตรวจให้คะแนนการเขียนโปรแกรม เพื่อให้การสอบ เป็นไปอย่างสะดวก

แต่เดิม ระบบนี้เป็นเพียงระบบตรวจสอบ code ด้วย test case และบันทึกคะแนน โดยใช้วิธีการอัปโหลดไฟล์ไปยังเซิร์ฟเวอร์

จากนั้นคอมไพล์โปรแกรมบนเซิร์ฟเวอร์และเปรียบเทียบผลลัพธ์ของฟังก์ชัน

แต่ด้วยความไม่สะดวกในการเปิดหน้าเว็บ และ IDE ไปพร้อมกัน

ทำให้เกิดความซ้ำซ้อนในการทำงาน นอกจากนี้ การนำทางในหน้าเว็บ การสรุปคะแนน

และการป้อนโจทย์เข้าไปในระบบ ยังทำได้ยาก เราจึงพัฒนาปรับปรุงระบบขึ้นมาใหม่

โดยมีรากฐานจากจุดอ่อนของระบบเดิมและปัญหาที่พบของโปรแกรมรุ่นก่อน

รวมถึงความต้องการที่มากขึ้นของผู้ใช้งาน เพื่อสร้างระบบที่มีประสิทธิภาพ

เนื่องจากระบบของเราเป็นระบบที่มีขนาดใหญ่ เราจึงจัดหมวดหมู่ของ SRS เป็น Section เพื่อให้ง่ายต่อการรายงาน ดังนี้

1. Account Section
2. Quiz Setup Section
3. Student Section
4. Teacher / TA Sub Section
5. IDE Section
6. User Interface Section

User story

ผู้ใช้งานจะได้รหัสโดยอัตโนมัติแต่สามารถที่จะเปลี่ยนรหัสผ่านได้ในภายหลัง
สถานะตั้งต้นของผู้ใช้ทุกคนคือนักศึกษา และอาจารย์ที่มี ID Admin สามารถ Approve
สถานะของผู้ใช้จากนักศึกษา สู่อาจารย์ผู้สอน หรือผู้ช่วยอาจารย์ได้

อาจารย์และ TA สามารถนำข้อสอบเข้าสู่ระบบ(โจทย์ , test case)
โดยกำหนดวันที่เปิดรับการส่งงาน และกำหนดการปิดการรับงานได้(ระบบ Time limit)
นอกจากนี้อาจารย์สามารถกำหนดเกณฑ์การตรวจข้อสอบได้ (แบบ checking(pass/fail)
หรือแบบให้คะแนน(Scoring) นอกจากนี้จะมีการตัดคะแนนเนื่องจากการส่งงานล่าช้าได้ (2 hrs. :
-10%, 24 hrs : -50% , else : -100%)

เมื่อถึงวันเวลาที่กำหนด นักเรียนจะเข้าสู่ระบบ และส่งงาน โดยส่งโดยการอัปโหลดไฟล์
หรือพิมพ์บน Text field และรันบนเซิร์ฟเวอร์ โดยที่เมื่อมีการเขียน Code Text field จะมีการ
highlight code เพื่อให้ง่ายในการอ่านและแก้ไข และ จะมีระบบแจ้งเตือนตอนใกล้ถึงเวลาส่ง
หลังจากส่งแล้วเซิร์ฟเวอร์จะตรวจสอบผลการรัน Test case หลังจากนั้นจะมีการโชว์คะแนนที่ได้

อาจารย์และผู้ช่วยสามารถตรวจดู code และประเมินคะแนนในระบบ scoring
และสามารถเพิ่มลดเทสเคสในระบบ checking pass fail ได้

มีระบบช่วยในการตรวจจับการลอกกัน
มีระบบสุ่มข้อสอบในคลังเพื่อไม่ให้ซ้ำกันในการทดสอบแต่ละครั้ง
ในระหว่างการเรียนรู้ อาจารย์สามารถตรวจสอบความคืบหน้าของชั้นเรียนได้ โดยระบบจะแสดง
ความคืบหน้าในการส่งงาน และนักศึกษาที่มีต้องเฝ้าระวัง โดยแสดงข้อมูลการติดตามตัวนักศึกษา
เช่นชื่อเล่น ข้อมูลการติดต่อ รูปภาพนักศึกษา (User Profile)

เมื่อสิ้นสุดชั้นเรียน อาจารย์สามารถดูผลสรุปของ class และคะแนนของนักเรียนได้
และอาจารย์สามารถ ดาวน์โหลดจากเว็บมาเป็น Excel ได้

มีระบบ schedule และแสดงวันที่จะแสดงโจทย์เพื่อให้นักศึกษาเข้ามาดูได้อย่างสะดวก

User Requirement

Account Section

- ระบบจะมีบัญชีผู้ใช้ไว้แล้วตั้งแต่แรก
- ผู้ใช้สามารถเข้าใช้งานระบบได้
- ระบบมีการส่งลิงค์ไปยัง email ของผู้ใช้งานในตอนแรก เพื่อยืนยันว่า email ที่ผู้ใช้ใส่ มีอยู่จริง และเพื่อตรวจสอบว่าผู้ใช้งานใส่ข้อมูลผิดหรือไม่
- ผู้ใช้สามารถเปลี่ยนรหัสผ่านได้ในภายหลัง

Quiz Setup Section

- การนำโจทย์เข้าสู่ระบบสามารถเป็นได้ทั้ง text field และ การอัปโหลดไฟล์จากภายนอกได้
- สามารถใส่คำตอบของแต่ละโจทย์ได้ ไว้ตรวจสอบถูกผิด
- ภาษาที่ใช้อัปโหลดโจทย์ต้องเป็น python
- ผู้ที่สามารถอัปโหลดโจทย์ได้จะมีแค่อาจารย์กับผู้ช่วยอาจารย์เท่านั้น
- สามารถตรวจ code ที่นักเรียนส่งมาว่าถูกต้องตามเงื่อนไขหรือไม่
- สามารถกำหนดเกณฑ์การให้คะแนนของโจทย์ได้เป็นแบบ ผ่าน/ไม่ผ่าน กับ แบบ พิจารณาคะแนน
- สามารถกำหนดระยะเวลาที่สามารถส่งงานได้
- สามารถตัดคะแนนตามเกณฑ์ ถ้าเกิดมีการส่งงานล่าช้า

Student Section

- เมื่อทำการ submit code ตัวของ code จะคงอยู่ในระบบ สามารถเรียกดู code ที่ submit ล่าสุดได้
- สามารถแสดง profile ของนักศึกษาแต่ละคนได้ (ชื่อ นามสกุล รหัสนักศึกษา และรูปภาพนักศึกษา)
- สามารถแสดง progress student (ความคืบหน้าของงาน)ได้
- เมื่อเวลาในการทำส่งแบบฝึกหัดและข้อสอบใกล้เคียงจะมีการแจ้งเตือน
- สามารถกดแสดงคะแนนของนักศึกษาได้

Teacher / TA Sub Section

- สามารถเข้าไปดู Progress และ Profile ของนักศึกษาทุกคนที่อยู่ใน Class นั้นๆได้
- สามารถ Export Score และ Profile บางส่วนของนักศึกษาใน Class นั้นๆให้เป็น Excel ได้

IDE Section

- ระบบการ Submit code ที่ใช้งานง่าย ไม่ซับซ้อน

- สามารถเขียน code หรือ upload file python ได้
- แสดงผลการรัน และ syntax error ได้
- สามารถตรวจสอบโปรแกรมว่าทำงานถูกต้องตามโจทย์กำหนดหรือไม่
- แสดงผลการทดสอบ test case และบันทึกคะแนนได้
- สามารถย้อนดูโปรแกรมที่ส่งไปแล้วได้
- สามารถจำกัดเวลาในการรัน เพื่อป้องกันการทำซ้ำเรื่อยๆได้
- สามารถหมุนเวียนโจทย์ในรอบการสอบเดียวกัน
และสุ่มโจทย์ให้นักเรียนได้รับโจทย์ต่างกันในรอบการสอบ
และไม่ซ้ำกันในการสอบรอบถัดไป
- ตรวจสอบความคล้ายคลึงของ code ได้
- แสดง highlight บน code
- รองรับการนำโจทย์เข้าสู่ระบบด้วยภาษา Markdown ได้
- อนุญาตให้มีการ submit เฉพาะเวลาที่กำหนดเท่านั้น

User Interface Section

- ทุกส่วนในระบบสามารถ link ไปยังส่วนอื่นๆของระบบ
- สามารถดูแล้ว ใช้งานได้ทันที ไม่ต้องมีคนมาแนะนำ
- มี Navigation bar ในทุกหน้าเว็บ
- ลักษณะของแต่ละหน้า มีลักษณะรูปแบบเหมือนกัน

System Requirement

Functional Requirement

Account Section

- มีข้อมูล Username และ Password ทั้งหมดอยู่ใน database ตั้งแต่แรก
- เมื่อ User ทำการ Log in ต้องไปทำการเทียบ permission ใน database เพื่อที่จะได้ทราบสถานะของ User
- User ต้องเข้าไป verify email เป็นอันดับแรก ก่อนทำอย่างอื่น (สำหรับการเข้าสู่ระบบครั้งแรกเท่านั้น)
- User จะต้องสามารถเข้าถึง System ด้วย Account ของตนเองได้
- User ต้องสามารถเปลี่ยน password ได้ในภายหลังการ verify email (สำหรับการเข้าสู่ระบบครั้งแรกเท่านั้น)
- หาก User ลืม password User ต้องสามารถใส่ email ที่ใช้ตอนแรก เพื่อ Reset password ใหม่ได้

Quiz Setup Section

- ข้อมูลโจทย์ที่ทำการอัปโหลดทั้งหมดต้องนำไปเก็บไว้ใน database
- สามารถอัปโหลดได้ทั้งแบบไฟล์ text field หรือ เขียนโค้ดลงบนระบบได้
- ต้องสามารถใส่คำตอบเพื่อทำเป็น test case ได้ไม่จำกัดจำนวน
- ผู้ที่สามารถอัปโหลดโจทย์ได้ต้องมีสถานะ User เป็นอาจารย์กับผู้ช่วยอาจารย์เท่านั้น
- จะเลือกได้ว่าจะใช้เกณฑ์คะแนนแบบไหน Checking หรือ Scoring
- Checking มี 2 แบบ pass(ผ่าน) กับ fail(ไม่ผ่าน)
- Scoring ระบบให้คะแนน พิจารณาโดยอาจารย์ผู้สอน
- การตัดคะแนนเนื่องจากการส่งงานล่าช้า
 1. ส่งช้าไม่เกิน 2 ชั่วโมง ตัดคะแนน 10% จากคะแนนที่ได้
 2. ส่งช้าหลังจาก 2 ชั่วโมง แต่ไม่เกิน 24 ชม. ตัดคะแนน 50% จากคะแนนที่ได้
 3. ส่งช้าหลังจาก 24 ชั่วโมง ตัดคะแนน 100%
- สามารถกำหนดระยะเวลาที่สามารถส่งงานได้ ถ้าส่งช้าจะตัดคะแนนตาม การตัดคะแนนเนื่องจากการส่งงานล่าช้า

Student Section

- HTMLและ CSS จะนำมาใช้ในส่วนของ front-end ซึ่งจะใช้การทำหน้าเว็บไซต์

- python จะนำมาใช้ในส่วนของ back-end ใช้ทำส่วนของ Server
- นักศึกษาที่ทำการ submit code ตัวของ code จะไม่หายไปโดยการใช้ Javascript มาแก้ไข
- แสดง Profile ส่วนตัว of นักศึกษาโดยใช้ python, CSS และ HTML
- ต้องสามารถแสดงให้เห็นถึงความคืบหน้าในการทำแบบฝึกหัดและข้อสอบของนักศึกษาได้
- เมื่อเวลาใกล้หมดจะมีการเตือนไปที่อีเมลและเว็บไซต์ผ่าน ID นั้นหรือตอนทำข้อสอบก็จะมีแจ้งเตือนเมื่อเวลาใกล้หมด
- ต้องสามารถแสดงคะแนนของนักศึกษาออกมาให้ดูได้
- สามารถเข้าไปทำแบบฝึกหัดและข้อสอบในเวลาที่กำหนด
- สามารถเข้าไปดู Schedule ได้
- มีระบบ navigation ใช้ในการนำทาง

Teacher / TA Sub Section

- สามารถเข้าถึง Progress และ Profile ของนักเรียนทุกคนใน Class นั้น เพื่อให้อาจารย์นั้น สามารถทราบได้ถึง ความคืบหน้าของงานที่ได้มอบหมายให้นักศึกษา
- เป็นระบบที่สามารถ Export Profile บางส่วนของนักศึกษา และคะแนนจากส่วนของ Quiz ของนักศึกษาทั้งหมดที่อยู่ใน Class นั้นให้มาอยู่ใน Excel ได้

IDE Section

- สามารถ input บน text field ของหน้าเว็บ หรือ import .py โดยเลือกจากไดเรกทอรีได้ (browse directory)
- เมื่อ submit จะเก็บ code ล่าสุดสำหรับโจทย์ข้อนั้นๆไว้ใน database จนกว่าจะมีการลบโจทย์ออกจากระบบ
- เมื่อ submit , webserver จะรัน code ที่ส่งเข้ามา และทดสอบฟังก์ชันด้วย test case

นำผลลัพธ์ไปตรวจเทียบกับผลลัพธ์ที่ถูกต้อง แสดงผลการตรวจ
และบันทึกผลการตรวจ/คะแนนไว้ยัง database ตาม ID ของ user

- สามารถแสดง debug และ syntax error ได้
- เมื่อ user เปิดโจทย์ที่เคย submit ไปแล้ว จะแสดง code
ล่าสุดที่เคย submit โดยสามารถแก้ไขต่อ หรือ clear ได้
- มีระบบ watch dog timer ซึ่งจะ terminate program
เมื่อใช้เวลาในการรันนานเกินไป
- การสอบหนึ่งรอบ สามารถใส่โจทย์ได้หลายข้อ
และสามารถเลือกรูปแบบการสุ่มได้ตาม section , seat ,
หรือรอบการสอบได้
- การแสดงผล code ใน text field จะมีการ highlight คำที่เป็น
keyword ของภาษา python ด้วย
- การ submit จะทำได้เมื่ออยู่ในช่วงเวลาที่กำหนดเท่านั้น
- admin สามารถ force ผลการตรวจได้ ในกรณีเกิดข้อผิดพลาด
หรือการทุจริต
- สามารถตรวจสอบความคล้ายคลึงของโปรแกรม (MOSS) ได้

User Interface Section

- สามารถ link ไปที่ส่วนอื่นๆของเว็บที่เกี่ยวข้อง
จากหน้าใดๆภายในระบบ
- Navigation bar ต้องมีในทุกหน้า และ
ลักษณะเหมือนกันในทุกหน้าของ UI(User Interface)
- รูปแบบ ของแต่ละหน้า UI จะมีลักษณะคล้ายคลึงกัน

\

Non-Functional Requirement

Account Section

- หน้า log in สวยงามและใช้งานง่าย

Quiz Setup Section

- รองรับการใช้งานพร้อมกันได้มากกว่า 80 User
- ต้องสามารถตรวจสอบได้ว่าการส่งงานล่าช้าจากกำหนดขนาดไหน

Student Section

- ใช้งานได้ง่ายและสวยงาม

Teacher / TA Sub Section

- สามารถใช้งานได้ง่าย และ สวยงาม

IDE section

- การทำงานของ web app ต้องไม่ป็นอันตรายต่อ web server
- สามารถรองรับการใช้งานของ user ได้พร้อมกันมากกว่า 80 user

User Interface Section

- สามารถทำให้ผู้ใช้เข้าใจวิธีการทำงานได้
โดยไม่ต้องมีคนมาอธิบายเสริม

สำหรับอธิบาย คำศัพท์เฉพาะทาง หรือตัวย่อสากลต่างๆ เพื่อให้เพื่อนเข้าใจ

Appendix

UI(User Interface)

IDE (Integrated development environment)

คือเครื่องมือพัฒนาซอฟต์แวร์ประกอบด้วย **source code editor** สำหรับเขียน และแก้ไขโปรแกรม และ **build automation** สำหรับคอมไพล์ รัน และแสดงผลการรันของโปรแกรม

Moss

(for a Measure Of Software Similarity) is an automatic system for determining the similarity of programs. To date, the main application of Moss has been in detecting plagiarism in programming classes. Since its development in 1994, Moss has been very effective in this role. The algorithm behind moss is a significant improvement over other cheating detection algorithms (at least, over those known to us).