

Document Report

The topic is grading platform for programming system.

Group 1

| | | |
|------------|-----------------|-------------|
| Piya | chatchaicharoen | 59340500038 |
| Ratirat | Sawangsi | 59340500068 |
| Thanphisai | sangharach | 59340500073 |
| Wutthichai | Buasanthea | 59340500070 |
| Natworpong | Loyswai | 59340500018 |
| Komkrit | Thongbunchu | 59340500006 |
| Phurin | sawangphul | 59340500046 |

1. Introduction (This section should include your problem statement, the objectives of the project, and the final product of the project. Briefly introduce the readers to your system, and what you are working on to point out the necessity of your project.)

The topic is grading platform for programming system.

การตรวจสอบโค้ดโปรแกรมสามารถทำได้ง่ายๆ เพียงแค่เราโหลดโปรแกรมอย่าง Pycharm Dev-c มาติดตั้งใน PC notebook หรืออุปกรณ์ทางอิเล็กทรอนิกส์ที่สามารถทำได้ เราก็สามารถตรวจสอบโค้ดโปรแกรมที่ต้องการโดยการเขียนโค้ดโปรแกรม และกด Run เพื่อ Compile ก็สามารถตรวจสอบความถูกต้องได้ แต่เป็นเพียงการตรวจสอบความถูกต้องที่คนเขียนสามารถดูได้เท่านั้น

ปัจจุบันมี Web application ที่สร้างมาเพื่อการฝึกเขียนโค้ดโปรแกรมและตรวจสอบโค้ดโปรแกรมไปพร้อมๆ กัน กล่าวง่ายๆ ให้เข้าใจคือ เมื่อเราเขียนโปรแกรมลงบน Web application ดังกล่าวก็สามารถกด Run เพื่อ Compile ภายใน Web application ได้เลยและสามารถตรวจสอบ Syntax หรือ Error ของโค้ดโปรแกรมได้ด้วย เช่น Codecademy Codewars แต่ในบางกรณีหากต้องการนำไปใช้ภายในองค์กร ต้องมีค่าใช้จ่ายเสมอ รวมไปถึงการใช้งาน Feature ต่างๆ ความยากของการใช้งาน Web application เราจึงคิดพัฒนา Web application เพื่อเป็นไปตามความต้องการของ customer ในด้านการใช้งาน ที่สะดวกสบายมากยิ่งขึ้น และเข้าใจง่าย มี Feature ที่ไม่ซับซ้อน มากจนเกินไป

Web application ที่เรากำลังพัฒนาขึ้นมานี้เป็น Web application เพื่อใช้ในการส่ง Code และตรวจสอบความถูกต้องของ Code โดยจะมี Customer อยู่ 2 กลุ่มคือ Admin (อาจารย์และ TA) และ User (นักศึกษา) เมื่อ Admin และ User ต้องการเข้าสู่ระบบ สามารถใช้ Teacher ID หรือ Student ID และ Password ที่ตั้งไว้ในระบบ Login เข้าสู่ระบบ

- Admin ซึ่งจะเป็นผู้สร้าง Classroom เพื่อให้ User เข้ามาส่งงานและ Activity อื่นๆ เมื่อ Admin สร้างห้องจะได้รับ Key code ของห้องนั้นๆและนำ Key code ไปให้แก่ User เพื่อให้ User join เข้ามาใน Classroom ทาง admin จะเป็นผู้สร้างโจทย์ที่สามารถกำหนดคุณลักษณะต่างๆได้ เช่น เวลาส่งงานของ User ภาษาที่ใช้ในการเขียน และกำหนด Test case ต่างๆเป็นต้น
- User จะได้รับ Key code จาก admin เพื่อ Join เข้าสู่ Classroom เมื่อ admin สร้างโจทย์ User จะเห็นโจทย์เวลาในการส่งงาน User จะเป็นผู้เขียนโค้ดได้ทั้งบนเว็บและส่งหรือเขียนบนโปรแกรมในคอมพิวเตอร์และอัปโหลดส่งผ่าน Web แล้วจะสามารถเห็นคะแนนได้เลย

2. Customer Requirements Adjustment (This section must include requirements from you get from your friends, and explain if you can use the requirements, or do you need to conduct additional user interviews? Explain if you change any of the requirements and/or specifications. How and why do you change it?)

Functional

Account Section

- มีข้อมูลUsername และ Password ทั้งหมดอยู่ใน Database ตั้งแต่แรก
(กำหนด Password ให้ก่อนแล้วเปลี่ยนเองทีหลัง)
- เมื่อ User ทำการ Log in ต้องไปทำการเทียบ Permission เพื่อที่จะได้รับทราบสถานะของ User
- Userต้องเข้าไป Verify Email เป็นอันดับแรก ก่อนทำการอย่างอื่น (สำหรับการเข้าสู่ระบบครั้งแรกเท่านั้น)
- User ต้องสามารถเข้าถึง System ด้วย Account ของตนเองได้
- User สามารถเข้าไปเปลี่ยน Password ได้ในภายหลัง Verify email (สำหรับการเข้าสู่ระบบครั้งแรกเท่านั้น)
- หาก User ลืม Password User ต้องสามารถใส่ Email ที่ใช้ตอนแรกเพื่อ ส่งลิงค์ Reset Password ใหม่ ไปยัง Email

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
 - สามารถกำหนดระยะเวลาที่สามารถส่งงานได้ ถ้าส่งช้าจะตัดคะแนน การตัดคะแนนเนื่องจากการส่งงานล่าช้า
- เช่น
- 1.ส่งช้าไปเกิน 2 ชั่วโมง ตัดคะแนน 10% จากคะแนนที่ได้
 - 2.ส่งช้าหลังจาก 2 ชั่วโมงแต่ไม่เกิน 24 ชั่วโมง ตัดคะแนน 50% จากคะแนนที่ได้
 - 3.ส่งช้าหลังจาก 24 ชั่วโมง ตัดคะแนน 100%

Student Section

- HTML และ CSS จะนำมาใช้ส่วนของ Front-end ซึ่งจะใช้การทำหน้าเว็บไซต์
- Python จะนำมาใช้ในส่วนของ Back-End ใช้ทำส่วนของ Server
- นักศึกษาที่ทำการ Submit Code ตัวของ Code ที่จะหายไปโดยการใช้ Java script มาแก้ไข
- แสดง Profile ส่วนตัวของนักศึกษาโดยการใช้ Python , CSS และ HTML
- ต้องการแสดงให้เห็นถึงความคืบหน้าในการทำแบบฝึกหัดและข้อสอบของนักศึกษาได้
- ตอนทำข้อสอบก็จะมีแจ้งเตือนเมื่อเวลาใกล้จะหมด(แจ้งแค่หน้าเว็บ)
- ต้องสามารถแสดงคะแนนของนักศึกษาออกมาให้ได้
- สามารถเข้าไปทำแบบฝึกหัดและข้อสอบในเวลาที่กำหนด
- แจ้งเตือนอัตโนมัติ (อาจจะแจ้งไปทาง Email)
- มีระบบ Navigation ใช้ในการนำทาง

Teacher / TA Sub Section

- สามารถเข้าถึง Progress และ Profile ของนักเรียนทุกคนใน Class นั้น เพื่อให้อาจารย์นั้น สามารถทราบได้ถึง ความคืบหน้า ของงานที่ได้รับมอบหมายให้นักศึกษา
- ระบบที่สามารถ Export Profile บางส่วนของนักศึกษาและ คะแนนจากส่วนของ Quiz ของนักศึกษาทั้งหมดที่อยู่ใน Class นั้นให้มาอยู่ใน Excel ได้

IDE Section

- สามารถ Input บน text field ของหน้าเว็บ หรือ import.py โดยเลือกจากไดเรกทอรีได้ (browse directory)
- เมื่อ Submit จะเก็บ code ล่าสุดสำหรับโจทย์ข้อนั้นไว้ใน 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 ไปแล้วจะแสดง Code ล่าสุดที่เคย submit โดยสามารถแก้ไขต่อได้
- มีระบบ Watch dog time ซึ่งจะ Terminate Program เมื่อใช้เวลาในการรันนานเกินไป
- การสอบหนึ่งครั้ง สามารถใส่โจทย์ได้หลายข้อ และสามารถเลือกรูปแบบการสุ่มได้ตาม Section , Seat หรือรอบการสอบได้

- การแสดงผล code ใน text field จะมีการ highlight คำที่เป็น keyword ของภาษา python ด้วย*
- การ Submit จะทำได้เมื่ออยู่ในช่วงเวลาที่กำหนดเท่านั้น
- Admin สามารถ Force ผลการตรวจได้ ในกรณีข้อผิดพลาด หรือการทุจริต
- สามารถตรวจสอบความคล้ายคลึงของโปรแกรม (MOSS) ได้*

User Interface Section

- สามารถ link ไปที่ส่วนอื่น ๆ ของเว็บที่เกี่ยวข้อง จากหน้าใดๆภายในระบบ
- Navigation bar ต้องมีในทุกหน้า และลักษณะเหมือนกันในทุกหน้าของ UI (User Interface)
- รูปแบบของแต่ละหน้า UI จะมีลักษณะคล้ายคลึงกัน

Non-Function

Account Section

- หน้า Login สวยงาม ใช้งานง่าย

Quiz Set-up Section

- รองรับการเข้าใช้งานพร้อมกันได้มากถึง 80 user
- สามารถตรวจสอบได้ว่าส่งงานล่าช้าจากกำหนดนานขนาดไหน

Student Section

- ใช้งานง่าย และสวยงาม

Teacher and TA Sub Section

- สามารถใช้งานได้ง่ายและสวยงาม

IDE Section

- การทำงานของ Web App ต้องไม่เป็นอันตรายต่อ Web Server
- รองรับการเข้าใช้งานพร้อมกันได้มากถึง 80 user

User Interface Section

- สามารถทำให้ผู้ใช้เข้าใจวิธีการทำงานได้ โดยไม่ต้องมีคนมาอธิบายเสริม

อธิบายเหตุผลที่ถูกตัดออก *

IDE Section

-การแสดงผล code ใน text field จะมีการ highlight คำที่เป็น keyword ของภาษา python ด้วย

หลังจากที่กลุ่มของพวกเราได้รับ Requirement ของอีกกลุ่มหนึ่งและได้นำ Requirement ไปสอบถามความคิดเห็นของ User ว่าอยากให้ปรับแก้ ตัดออก หรือต้องเพิ่มเติมอะไรอีก คำตอบที่ได้ คือ ตัดในการแสดงผล Code ใน Text field จะมีการ Highlight คำที่เป็น Keyword ของภาษา python เพราะว่าค่อยไม่จำเป็น

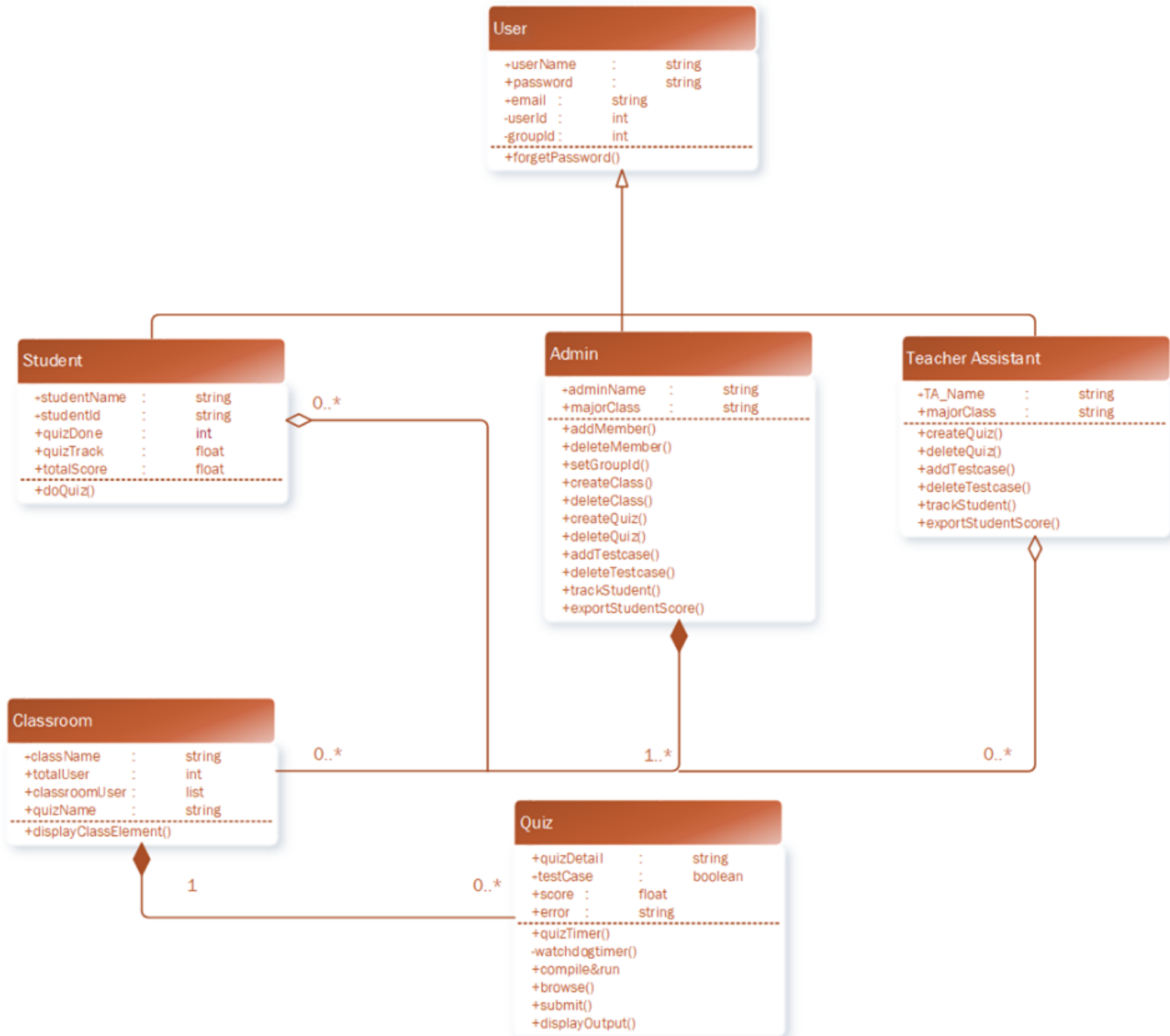
-สามารถตรวจสอบความคล้ายคลึงของโปรแกรม (MOSS) ได้

ในการตรวจสอบความคล้ายคลึงของโปรแกรม(MOSS)เป็น Requirement ที่ค่อนข้างใช้เวลาในการพัฒนามากพอสมควร ดังนั้น MOSS จึงถูกตัดออกจาก Functional ก่อน แต่ก็ยังเป็นสิ่งที่ควรมีในระบบ

3. Software Design

Back-end

UML Diagram for the full system



Class Diagram

Explain the design of the software you will implement.

ออกแบบให้มี 6 คลาส โดยคลาสแรกคือ User ที่มี object ตามที่กำหนด แล้ว inherit Admin Class, Teacher Assistant Class, Student Class โดยกำหนด permission(groupId) ให้ตามที่ได้ออกแบบไว้, Classroom เป็น Composition ของ Class Admin และเป็น Aggregation ของ Class Student และ Class Teacher Assistant โดยที่ 1 Classroom จะต้องมีย่อยอย่างน้อย 1 Admin และ 0 หรือมากกว่า สำหรับ Class Teacher Assistant, Class Student แล้วให้ Class Quiz เป็น Composition ของ Class Classroom โดยที่ 1 Class Classroom จะมี Quiz เป็น 0 หรือมากกว่า และ Class Quiz จะมี Classroom เป็น 1 หรือมากกว่า

How many classes are there?

มีทั้งหมด 6 คลาส

What is the class diagram for your system?

1. User
2. Admin
3. Teacher Assistant
4. Student
5. Classroom
6. Quiz

Why do you need those classes?

เพราะเป็นคลาสหลักที่สำคัญ และเราจำเป็นต้องนำคลาสทั้งหมดนี้มาใช้สำหรับ Full System

คลาส User มี object

1. userName
2. passWord
3. email
4. userId
5. groupId

คลาส Admin มี object

1.adminName

2.majorClass

คลาส Teacher Assistant มี object

1.TA_Name

2.majorClass

คลาส Student มี object

1.studentName

2.studentId

3.quizDone

4.quizTrack

5.totalScore

คลาส Classroom มี object

1.className

2.totalUser

3.classroomUser

4.quizName

คลาส Quiz มี object

1.quizDetail

2.testCase

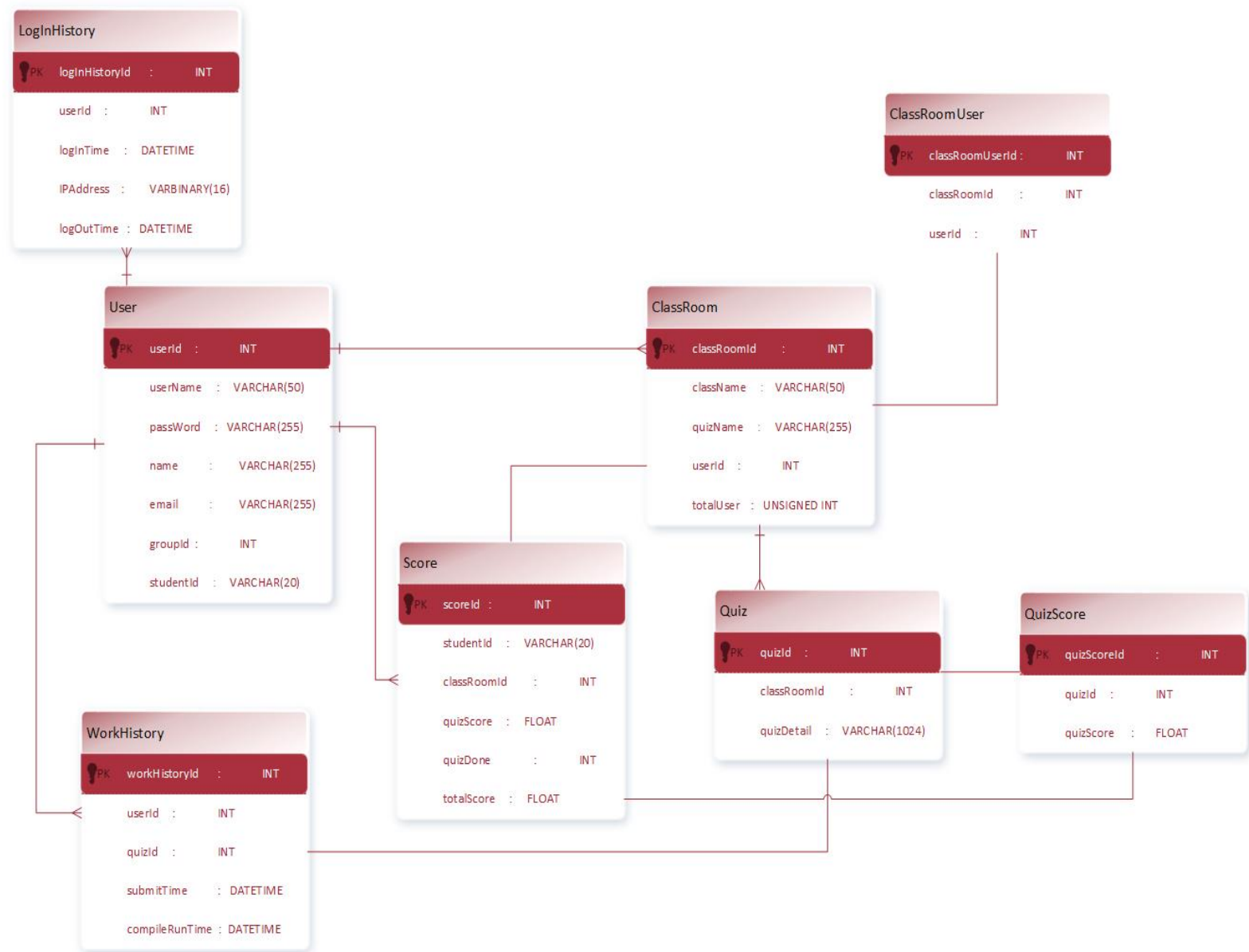
3.score

4.error

How do they interact between classes?

- Class Admin, Teacher Assistant และ Student Inherit จาก Class User
- Class Classroom เป็น Composition ของ Class Admin และเป็น Aggregation ของ Class Student และ Class Teacher Assistant
- Class Quiz เป็น Composition ของ Class Classroom

Database Diagram for the full system



Database Diagram

Explain the interaction in your database.

เมื่อมีการ Login ของ User จะมีการเก็บ History ไว้ในตาราง History รวมถึงการตรวจสอบว่ามี groupId ใดซึ่งสัมพันธ์แบบ one to one กับตาราง User แล้วจะตรวจสอบว่า User นั้นเป็นสมาชิกของ Classroom ใด ซึ่งจะสัมพันธ์แบบ one to many กับตาราง Classroom เนื่องจากว่า 1 User สามารถเข้าได้หลาย Classroom จากนั้นดูว่าใน Classroom มี Quiz หรือไม่ ซึ่งใน 1 Classroom สามารถมีได้หลาย Quiz จะเห็นว่า Classroom สัมพันธ์กับ Quiz Class แบบ one to many ซึ่งจะสำเร็จ Quiz ได้ด้วยการ compile&run code หรือ submit code และจะมีการเก็บไว้เป็น History ที่สัมพันธ์กับตาราง WorkHistory Quiz Table สัมพันธ์กับ QuizScore Table แบบ one to one แล้วยังสัมพันธ์กับ Score Table แบบ one to one ด้วย

How many tables are there?

มีทั้งหมด 8 ตาราง

What information does each table hold?

- ตาราง LoginHistory มี Primary Key คอลัมน์ชื่อ loginHistoryId AUTO INCREMENT และเก็บค่าประเภท INT ,มี Foreign Key คอลัมน์ชื่อ userId และ เก็บค่าประเภท INT,
- คอลัมน์ชื่อ loginTime ประเภท Datetime, คอลัมน์ IPAddress เก็บค่าประเภท varbinary(16) และ คอลัมน์ logOutTime เก็บค่าประเภท DATETIME
- ตาราง User มี Primary Key เป็นตัวแปรชื่อ userId ประเภท INT และ AUTO INCREMENT, มี Foreign Key คอลัมน์ชื่อ คอลัมน์ชื่อ userName เก็บค่าประเภท VARCHAR(50), คอลัมน์ชื่อ passWord เก็บค่าประเภท VARCHAR(255), คอลัมน์ชื่อ name ประเภท VARCHAR(255), คอลัมน์ email เก็บค่าประเภท VARCHAR(255),คอลัมน์ groupId เก็บค่าประเภท INT, คอลัมน์ studentId เก็บค่าประเภท VARCHAR(20)
- ตาราง Classroom มี Primary Key เป็นคอลัมน์ชื่อ classroomId ประเภท INT และ AUTO INCREMENT และมี Foreign Key คือ userId เก็บค่าประเภท INT, คอลัมน์ชื่อ className เก็บค่าประเภท VARCHAR(50),คอลัมน์ชื่อ quizName เก็บค่าประเภท VARCHAR(255) และ คอลัมน์ชื่อ totalUser เก็บค่าประเภท UNSIGNED INT

- ตาราง ClassroomUser มี Primary Key เป็นคอลัมน์ชื่อ classroomUserId ประเภท INT และ AUTO INCREMENT, มี Foreign Key คอลัมน์ชื่อ roomId ประเภท INT และ userId ประเภท INT
- ตาราง Quiz มี Primary Key เป็นคอลัมน์ชื่อ quizId ประเภท INT และ AUTO INCREMENT, มี Foreign Key คอลัมน์ชื่อ roomId ประเภท INT , คอลัมน์ชื่อ quizDetail เก็บค่าประเภท VARCHAR(1024)
- ตาราง QuizScore มี Primary Key เป็นคอลัมน์ชื่อ quizScoreId ประเภท INT และ AUTO INCREMENT, มี Foreign Key คอลัมน์ชื่อ quizId ประเภท INT และ คอลัมน์ชื่อ quizScore เก็บค่าประเภท FLOAT
- ตาราง Score มี Primary Key เป็นคอลัมน์ชื่อ scoreId ประเภท INT และ AUTO INCREMENT, มี Foreign Key คอลัมน์ชื่อ studentId ประเภท VARCHAR(20) และ roomId ประเภท INT, มีคอลัมน์ชื่อ quizScore เก็บค่าประเภท FLOAT, คอลัมน์ชื่อ quizDone เก็บค่าประเภท INT และคอลัมน์ชื่อ totalScore เก็บค่าประเภท FLOAT
- ตาราง WorkHistory มี Primary Key คอลัมน์ชื่อ workHistoryId AUTO INCREMENT และ เก็บค่าประเภท INT ,มี Foreign Key คอลัมน์ชื่อ userId และ quizId เก็บค่าประเภท INT,
- คอลัมน์ชื่อ submitTime เก็บค่าประเภท DATETIME, คอลัมน์ชื่อ compileRunTime เก็บค่าประเภท DATETIME

What are the connections between tables?

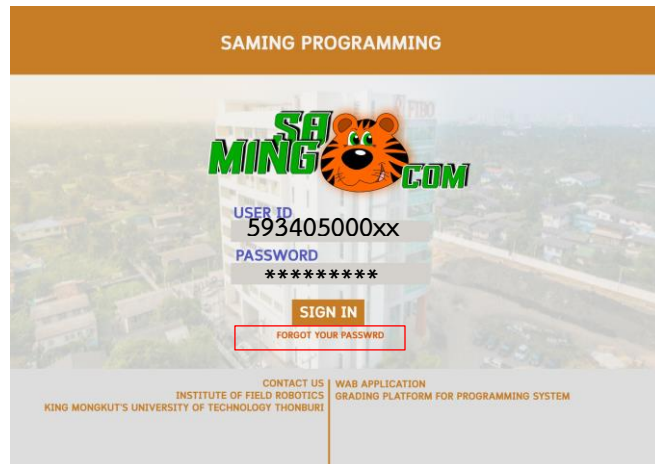
ตาราง User เชื่อมกับตาราง LoginHistory, WorkHistory แบบ one to many และตาราง Score แบบ one to many, ตาราง User เชื่อมกับตาราง Classroom แบบ one to many และเชื่อมกับ, ตาราง Classroom เชื่อมกับตาราง ClassroomUser แบบ one to many และ ตาราง Classroom เชื่อมกับตาราง Quiz แบบ one to many, ตาราง Quiz เชื่อมกับตาราง WorkHistory แบบ one to one และตาราง Quiz เชื่อมกับตาราง QuizScore แบบ one to one และตาราง QuizScore เชื่อมกับ ตาราง Score แบบ one to one

How can your system access the database?

ระบบจะเชื่อมต่อกับ Database ได้ด้วย MySQL และการใช้ภาษา SQL

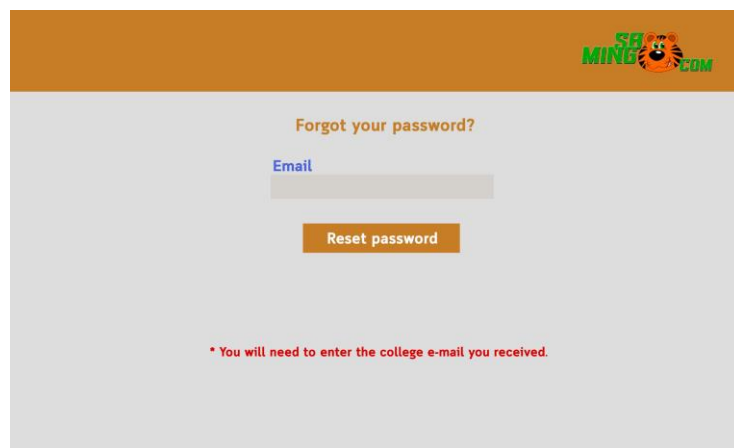
Front-end

Interface Design



1

Sign in page คือ หน้าแรกของเว็บสำหรับการเข้าสู่ระบบ โดยการป้อน User ID และ Password



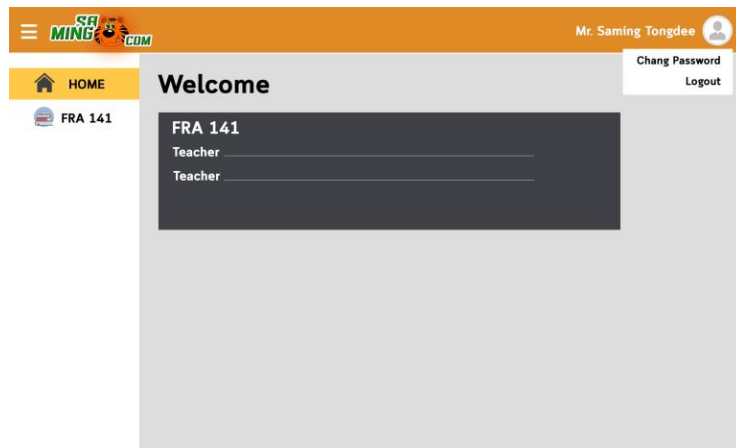
2

จากรูปที่ 1 เมื่อ User ลืม Password แล้วกด Forgot your password จะเข้าสู่หน้านี้ (2)

คลิกที่นี่'." data-bbox="261 600 725 801"/>

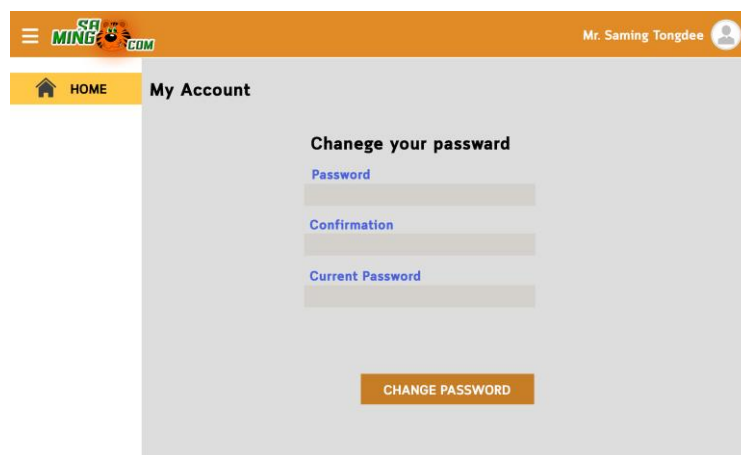
3

จากรูปที่ 2 เมื่อป้อน Email แล้วกด Reset password ระบบจะขึ้นหน้า 3 ดังรูปด้านบน (3) หลังจากนั้นระบบจะสุ่ม Password ชั่วคราว เพื่อทำการ Sign in เข้าสู่ระบบ โดยการกดปุ่ม Confirm และจะเข้าสู่ระบบทันที



4

จากที่ 4 เมื่อเข้าสู่ระบบแล้วต้องการ Change password กด icon ด้านบนขวามือ(วงกลมสีแดง) จะมี option ขึ้นมาให้ Change password และ Log out ออกจากระบบ



5

เมื่อเลือก option Change password จากรูปที่ 4 จะขึ้นหน้า ดังรูปที่ 5 เพื่อทำการ Change password



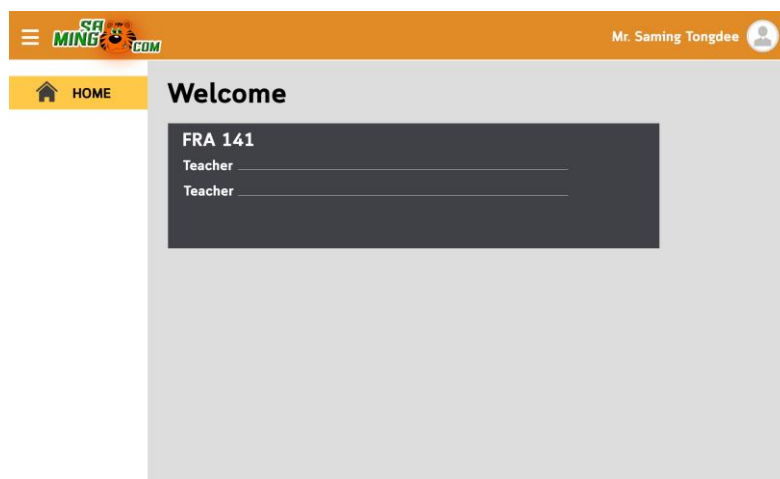
6

จากรูปที่ 5 เมื่อทำการกด confirm แล้ว จะออกมาสู่หน้าแรกหลังจากทำการ Log in เข้าสู่ระบบ ดังรูปที่ 6



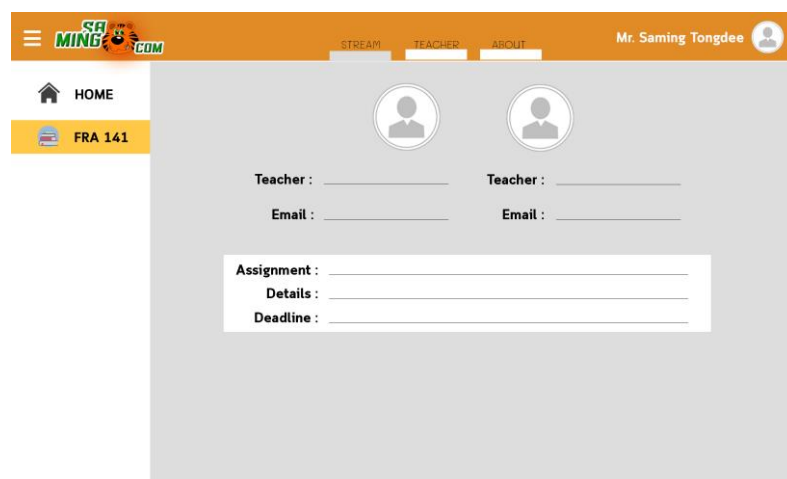
7

หากต้องการที่ทำการเข้าสู่หน้างาน User จะต้องทำการเลือก ชั้นปีการศึกษา คือ นักศึกษา ปี 1 ปี เป็นต้น



8

เมื่อ User ได้เลือกชั้นปีการศึกษาแล้ว ระบบจะขึ้นหน้า ดังรูปที่ 8 โดยจะแสดงรายละเอียดของชื่อวิชาและชื่ออาจารย์ผู้สอน



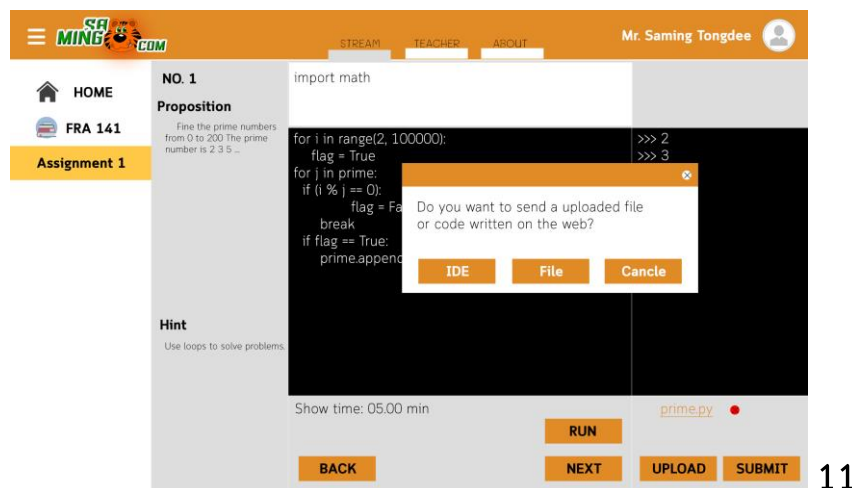
9

เมื่อ User ทำการกดเพื่อเข้าสู่วิชานั้น ๆ แล้วจะแสดงรายละเอียดดังรูปที่ 9 ว่ามีงานที่ Admin สั่งทั้งหมดกี่งาน

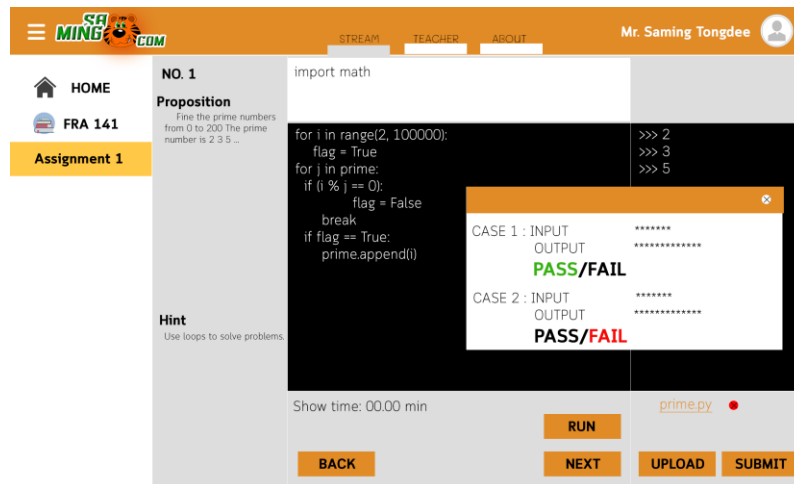


และเมื่อ User กดเลือกงานที่จะทำ ระบบจะทำการเข้าสู่หน้านี้ (10) โดยรายละเอียดจะเป็นดังนี้

- กรอบที่ 1 จะเป็นการแสดงรายละเอียดของโจทย์ และมี Hint (Admin เป็นคนกำหนดว่าจะมีหรือไม่)
- กรอบที่ 2 ส่วนสีขาวจะเป็นส่วนโปรแกรมที่ห้ามแก้ไขตาม Admin กำหนด ส่วนสีดำด้านล่าง จะเป็น IDE สำหรับ ใช้ในการเขียนโปรแกรม
- กรอบที่ 3 จะเป็นส่วนของการแสดงผลโปรแกรม

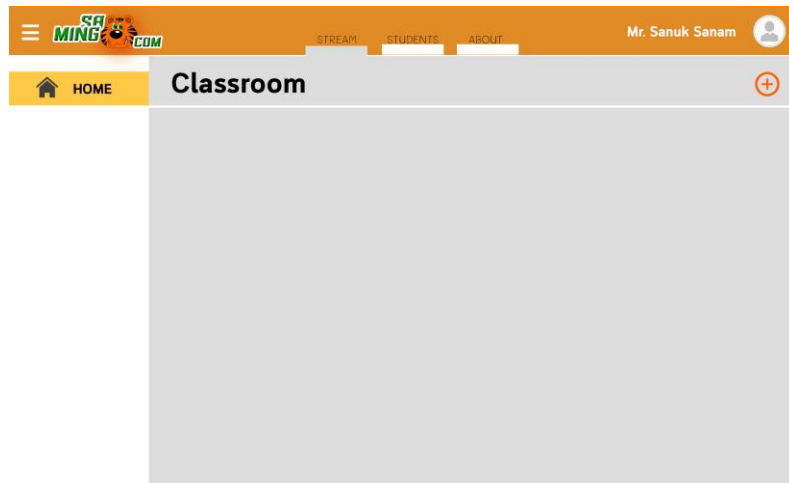


จากรูปที่ 10 หาก User ทำการ Upload file พร้อมกับการเขียน Code บนเว็บ แล้วทำการ กด Submit จะมีหน้าต่างปรากฏขึ้นมาให้เลือก ว่าเราจะส่ง File ที่ Upload หรือที่เขียนบนเว็บ



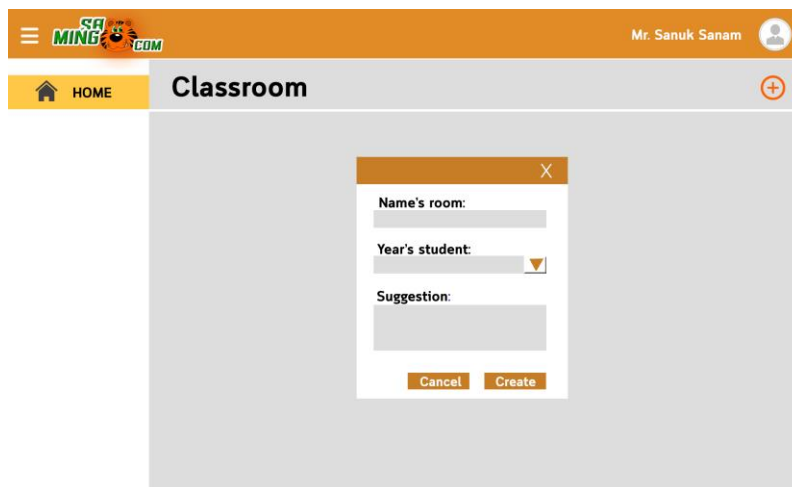
12

จากรูปที่ 11 เมื่อเราทำการเลือก File ที่ต้องการ Submit แล้ว จะปรากฏหน้าต่างแสดง Test case ขึ้นมา
ว่า User ทำผ่าน Test case ไດบ้าง และอาจมีคะแนนที่ทำได้แสดงด้วย



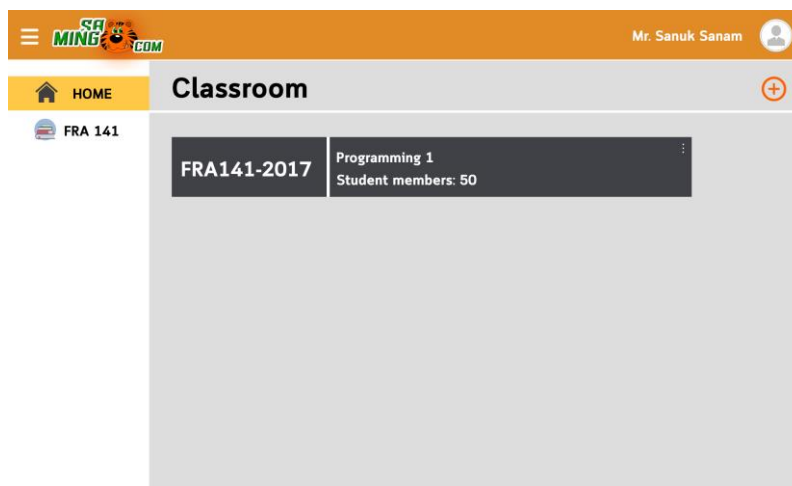
13

เมื่อ Teacher login เข้าสู่ระบบ จะพบกับหน้าแรก ซึ่งเป็นหน้าของชั้นเรียน



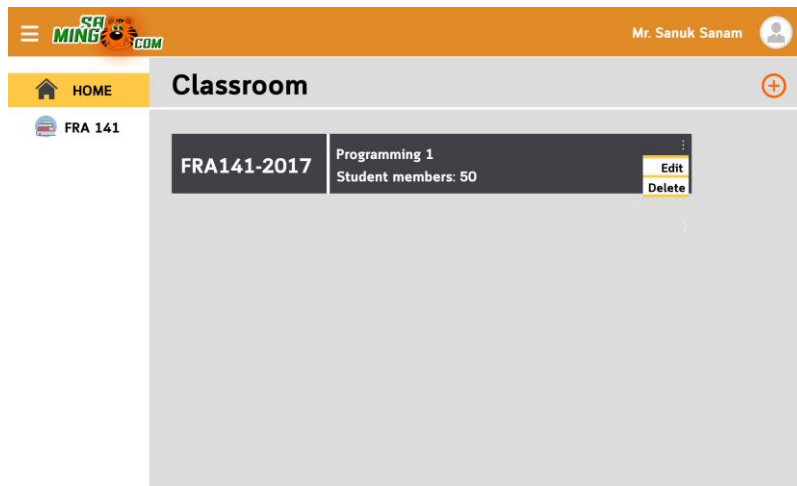
14

จากรูปที่ 13 เมื่อ Teacher กดปุ่ม + จะปรากฏหน้าต่าง เพื่อสร้างชั้นเรียน และกรอกรายละเอียดของชั้นเรียน เลือกชั้นปี



15

จากรูปที่ 14 เมื่อกด Create จะขึ้นหน้าชั้นเรียนที่สร้างเสร็จแล้ว



16

จากรูปที่ 15 เมื่อสร้างชั้นเรียนเสร็จแล้ว สามารถเข้าไปแก้ไขหรือลบชั้นเรียน ได้จากการกด Icon จากมุมมองด้านบนขวาของหน้าต่างชั้นเรียนนั้น ๆ



17

จากรูปที่ 16 เมื่อกดเข้าชั้นเรียน จะพบหน้าที่ใช้สำหรับสร้างงาน

18

จากรูปที่ 17 เมื่อเรากดปุ่ม + จะเข้าไปสู่หน้าดังรูปที่ 18 และมีช่องให้กรอกรายละเอียดงานต่าง ๆ

19

จากรูปที่ 18 หลังจากที่เรากรอกรายละเอียดงานเสร็จ กดปุ่ม Test Case จะขึ้นหน้าต่าง Test Case เพื่อระบุจำนวน Case คะแนนของแต่ละ Case และ Show case รวมทั้ง Show Score กด Save เพื่อบันทึกงาน

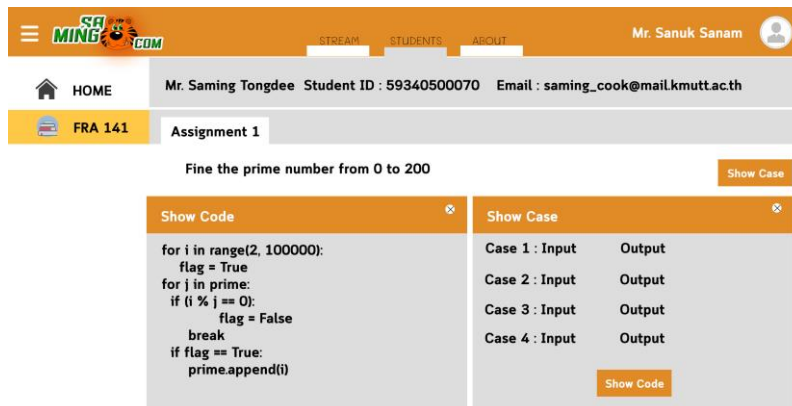
| Student ID | Name - Lastname | Assignment 1 | Assignment 2 |
|------------|--------------------|--------------|--------------|
| 123456780 | Mr. Saming Tongdee | 10 | 5 |
| 123456781 | Mr. Samai Sutee | 10 | 7 |
| 123456782 | Ms. Suunee Painan | 10 | 10 |
| 123456783 | Mr. Sanam Kadam | 5 | 10 |

20

เมื่อ Admin กดมาที่แถบ Students ที่อยู่ด้านบน จะแสดงตารางรายชื่อ Students ที่อยู่ในชั้นเรียนนั้น

21

จากรูปที่ 20 เมื่อกดไปที่ชื่อของ Student คนนั้น ก็จะปรากฏรายละเอียดของ Student คนนั้น



Mr. Saming Tongdee Student ID : 59340500070 Email : saming_cook@mail.kmutt.ac.th

FRA 141 Assignment 1

Fine the prime number from 0 to 200 [Show Case](#)

Show Code

```

for i in range(2, 100000):
    flag = True
    for j in prime:
        if (i % j == 0):
            flag = False
            break
    if flag == True:
        prime.append(i)

```

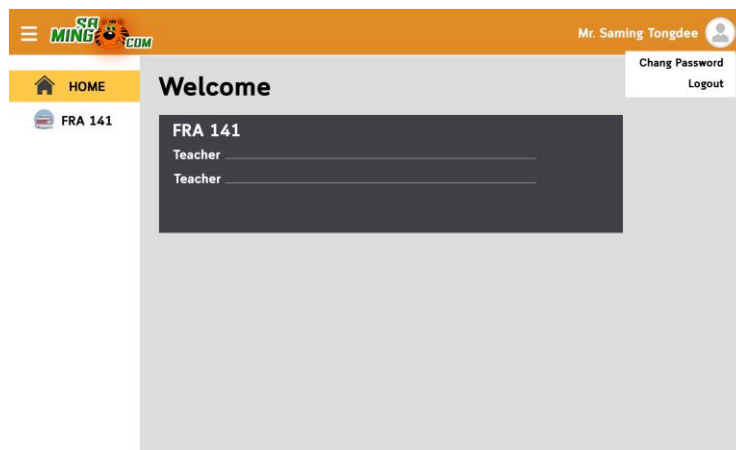
Show Case

| Case | Input | Output |
|--------|-------|--------|
| Case 1 | Input | Output |
| Case 2 | Input | Output |
| Case 3 | Input | Output |
| Case 4 | Input | Output |

[Show Code](#)

22

เมื่อ Admin กดที่ปุ่ม Show case จะมีหน้าต่างแสดง Test case ของข้อนั้นขึ้นมา และ Admin สามารถกดปุ่ม Show code เพื่อแสดง Code ที่ Student คนนั้นส่งมา ดังรูปที่ 22



Mr. Saming Tongdee [Chang Password](#) [Logout](#)

HOME FRA 141

Welcome

FRA 141

Teacher

Teacher

23

หากต้องการ Log out ก็สามารถ กด Icon ด้านบนขวามือ แล้วจะมีแถบขึ้นให้ กด Log out ออกจากระบบ