

ระบบประเมินผลสำหรับการเขียนโปรแกรมไพธอน

Automated Assessment of Python Programming



ศุภา จันทรพัฒน์

สหทัศน์ ลีวัฒนา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2565

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบประเมินผลสำหรับการเขียน โปรแกรมไพธอน

Automated Assessment of Python Programming

ผู้จัดทำ

1. นายภูฟ้า จันทรพัฒน์ รหัสนักศึกษา 62010713

2. นายสหทัศน์ ลีวัฒนา รหัสนักศึกษา 62010922



อาจารย์ที่ปรึกษา

(ผศ.ธนา หงษ์สุวรรณ)



อาจารย์ที่ปรึกษาร่วม

(ผศ.ดร.ชมพูนุท เต็งเจริญ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบประเมินผลสำหรับการเขียนโปรแกรมไพธอน

นายภูฟ้า	จันทรพัฒน์	62010713
นายสหทัศน์	ลีวัฒนา	62010922
ผศ.ธนา	หงษ์สุวรรณ	อาจารย์ที่ปรึกษา
ผศ.ดร.ชมพูนุท	เต็งเจริญ	อาจารย์ที่ปรึกษาร่วม

ปีการศึกษา 2565

บทคัดย่อ

ระบบประเมินผลสำหรับการเขียนโปรแกรมไพธอนจัดทำขึ้น โดยมีวัตถุประสงค์เพื่อสนับสนุนวิชาการเขียนโปรแกรมที่ใช้ภาษาไพธอน เพื่อสนับสนุนอาจารย์ผู้สอนที่ต้องการนำไปใช้ให้มีความสะดวกสบายและมีประสิทธิภาพมากขึ้นในการสอน รวมถึงผู้ใช้งานคนอื่นๆที่ต้องการนำไปใช้ต่อ

จากปัญหาที่พบในเว็บเกรดเดอร์ในปัจจุบัน ที่ไม่สามารถตรวจวิธีการเขียนโปรแกรมได้ ผู้จัดทำได้ทำเว็บไซต์เกรดเดอร์ขึ้นมาใหม่ โดยตัวเว็บไซต์มีความสามารถหลักๆคือตรวจสอบวิธีการเขียนโปรแกรม กำหนดฟังก์ชันที่ห้ามใช้ได้ ส่วนอื่นๆนอกจากการเขียนโปรแกรม จะเป็นการจัดการกลุ่มเรียนแต่ละกลุ่มโดยสามารถจัดการจำนวนแบบฝึกหัด และจำนวนสมาชิกได้

ผลการศึกษาและพัฒนาระบบ ทำให้ได้เว็บไซต์เกรดเดอร์สำหรับวิชาการเขียนโปรแกรมไพธอนที่ตรงตามความต้องการของผู้ใช้ สามารถนำระบบไปใช้งานได้จริงและมีประสิทธิภาพ

Automated Assessment of Python Programming

Mr. Pufa	Chantarapatana	62010713
Mr. Sahatas	Leevathana	62010922
Asst.Prof. Thana	Hongsuwan	Advisor
Asst.Prof.Dr. Chompoonuch	Tengcharoen	Co-Advisor

Academic Year 2022

ABSTRACT

Automated Assessment of Python Programming was created with the objective to support programming subjects that use the Python language, support teachers who want to apply to be more comfortable and effective in teaching including other users who want to continue using.

From the problems that occur in today's web graders You may check different methods, who need to renovate the grader website. The website itself is essentially checking the grading method for the session. that prohibit the use of other parts Here is an example of how to manage each learning group by managing the number of exercises and the number of members.

Results of study and system development results in a grader website for Python programming that meets the user's needs. Able to put the system into practice and efficiency.

สารบัญ

	หน้า
บทคำย่อภาษาไทย	I
บทคำย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	2
1.3 เป้าหมายของโครงการ	2
1.4 ขอบเขตของโครงการ	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	4
2.1 React	4
2.2 Tailwind CSS	5
2.3 Redux	6
2.4 Code Mirror	7
2.5 Django Rest Framework	8
2.6 SQLite	9
2.7 Python-grader-utils	9
2.8 งานวิจัยที่เกี่ยวข้อง	12

สารบัญ (ต่อ)

	หน้า
บทที่ 3 การออกแบบ	14
3.1 โครงสร้างของระบบ	14
3.2 Flow diagram	16
3.3 Use case diagram	17
3.4 Sequence diagram	22
3.5 Class diagram	28
3.6 โครงสร้างฐานข้อมูล	32
3.7 User Interface	34
บทที่ 4 ผลการทดลอง	35
4.1 การเข้าสู่ระบบ	35
4.2 ฟังก์ชันการจัดการกลุ่มเรียน	36
4.3 ฟังก์ชันต่าง ๆ ในกลุ่มเรียน	37
4.4 ฟังก์ชันแบบฝึกหัด	42
4.5 ฟังก์ชันการทำแบบฝึกหัด	43
4.6 ฟังก์ชันการแก้ไขข้อมูลบัญชี	43
4.7 ฟังก์ชันระบบการจัดการหลังบ้าน	44
บทที่ 5 สรุป	51
5.1 บทสรุป	51
5.2 ปัญหาและอุปสรรคที่พบ	52
5.3 แนวทางการพัฒนาต่อ	52
บรรณานุกรม	53

สารบัญตาราง

	หน้า
ตาราง 1.1 แผนการดำเนินงานตาราง.....	4
ตาราง 3.1 Use case การสมัครสมาชิก (Register).....	18
ตาราง 3.2 Use case การเข้าสู่ระบบ (Login).....	18
ตาราง 3.3 Use case การเข้ากลุ่มเรียน (Join team).....	18
ตาราง 3.4 Use case การทำแบบฝึกหัด (Do Exercise).....	19
ตาราง 3.5 Use case การจัดการกลุ่มเรียนที่ผู้ใช้งานเข้าร่วม/เป็นผู้ดูแล (Manage Team).....	19
ตาราง 3.6 Use case การจัดการแบบฝึกหัดในกลุ่มเรียนที่ผู้ใช้งานเป็นผู้ดูแล (Manage Exercise).....	19
ตาราง 3.7 Use case การจัดการสมาชิกในกลุ่มเรียนที่ผู้ใช้งานเป็นผู้ดูแล (Manage Member).....	20
ตาราง 3.8 Use case การตรวจสอบคะแนนของสมาชิกในกลุ่มเรียนที่ผู้ใช้งานเป็นผู้ดูแล (Check Member Score).....	20
ตาราง 3.8 Use case การตั้งค่าบัญชี (Setting Account).....	20

สารบัญรูป

	หน้า
รูป 2.1 ตัวอย่างของการใช้ Code Mirror กับภาษา HTML และ CSS.....	8
รูป 2.2 การติดต่อระหว่าง Server และผู้ใช้งาน.....	10
รูป 2.3 โครงสร้างของ JSON.....	11
รูป 3.1 System design.....	14
รูป 3.2 Flow diagram ของระบบ.....	16
รูป 3.3 Use case diagram ของระบบ.....	17
รูป 3.4 Sequence diagram การเข้าสู่ระบบ.....	22
รูป 3.5 Sequence diagram การสมัครสมาชิก.....	23
รูป 3.6 Sequence diagram การสร้างกลุ่มเรียน.....	23
รูป 3.7 Sequence diagram การสร้างแบบฝึกหัด.....	24
รูป 3.8 Sequence diagram การทำแบบฝึกหัด.....	24
รูป 3.9 Sequence diagram การแก้ไขกลุ่มเรียน.....	25
รูป 3.10 Sequence diagram การแก้ไขแบบฝึกหัด.....	25
รูป 3.11 Sequence diagram การตั้งค่าบัญชี.....	26
รูป 3.12 Sequence diagram การเชิญสมาชิกเข้ากลุ่มเรียน.....	26
รูป 3.13 Sequence diagram การลบกลุ่มเรียน.....	27
รูป 3.14 ภาพรวมของ Class diagram.....	28
รูป 3.15 ER Diagram.....	32
รูป 3.16 Color palette.....	34
รูป 3.17 Color palette.....	34
รูป 4.1 หน้าการเข้าสู่ระบบ.....	35
รูป 4.2 หน้าการสมัครสมาชิก.....	36
รูป 4.3 หน้ากลุ่มเรียนทั้งหมด.....	36
รูป 4.4 เมนูการ edit และ delete.....	37
รูป 4.5 เมนูการสร้างกลุ่มเรียน.....	37

สารบัญรูป (ต่อ)

	หน้า
รูป 4.6 หน้ากลุ่มเรียน.....	38
รูป 4.7 หน้าสร้างแบบฝึกหัด.....	38
รูป 4.8 การแก้ไขกลุ่มเรียน.....	39
รูป 4.9 การแก้ไข Unit test.....	39
รูป 4.10 การแก้ไข Config code.....	39
รูป 4.11 การแก้ไข Example และ Guideline Code.....	40
รูป 4.12 หน้า View Submission.....	40
รูป 4.13 หน้าสมาชิกทั้งหมด.....	41
รูป 4.14 หน้าสมาชิกรายบุคคล.....	41
รูป 4.15 หน้าแบบฝึกหัด.....	42
รูป 4.16 หน้า View Code.....	42
รูป 4.17 หน้าเขียนโค้ด.....	43
รูป 4.18 หน้าแก้ไขรายละเอียดบัญชี.....	43
รูป 4.19 Model ORM ที่ใช้เชื่อมต่อกับระบบฐานข้อมูล.....	44
รูป 4.20 Serializer.....	45
รูป 4.21 View.....	46
รูป 4.22 ตัวอย่าง api.....	47
รูป 4.23 ตัวอย่าง api กลุ่มเรียน.....	47
รูป 4.24 ตัวอย่าง api การเรียกใช้แบบฝึกหัด.....	48
รูป 4.25 ตัวอย่าง api ต่างๆ.....	48
รูป 4.26 ตัวอย่าง route authentication.....	49
รูป 4.27 ตัวอย่าง register.....	49
รูป 4.28 ตัวอย่าง Assessment.....	50

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ในการเรียนวิชาการเขียนโปรแกรมในภาควิชาวิศวกรรมคอมพิวเตอร์หรือในภาควิชาอื่น ๆ มักจะใช้เว็บไซต์ทำแบบฝึกหัดต่าง ๆ ให้นักศึกษาทำ (หรือที่เรียกกันว่า“เกรดเดอร์”) ซึ่งในปัจจุบันมีเว็บไซต์เกรดเดอร์มากมายที่สามารถให้บุคคลทั่วไปเลือกใช้ได้

โดยปัญหาของทุกๆเว็บไซต์ล้วนมีปัญหาเดียวกัน คือสามารถตรวจสอบคำตอบสุดท้ายของตัวโปรแกรมได้ แต่ไม่สามารถตรวจสอบวิธีการเขียนโปรแกรม ว่าใช้ขั้นตอน, วิธีการแบบไหนในการคิดคำนวณคำตอบเหล่านั้นออกมา เช่น โจทย์ต้องการให้เขียนโปรแกรมเช็คว่า 1+1 มีคำตอบเป็นเท่าไร แต่ตัวคนเขียนโปรแกรมใช้วิธีให้โปรแกรมเขียนเลข 2 ออกมาในทันที ซึ่งตัวเว็บไซต์ที่ตรวจสอบเพียงคำตอบสุดท้ายจึงให้ผลลัพธ์ว่าทำโจทย์นี้ได้ถูกต้อง ซึ่งหากไม่ใช้มนุษย์ในการตรวจสอบ ก็ไม่สามารถทราบได้เลยว่าได้คำตอบเหล่านั้นมาได้อย่างไร

ปัญหาดังกล่าวได้ส่งผลกับตัววิชาการเขียนโปรแกรมที่อาจารย์ต้องการใช้ตัวเว็บไซต์เพื่อความสะดวกสบายในการเรียนการสอน แต่นักศึกษาในรายวิชาใช้ประโยชน์จากปัญหาดังกล่าว ทำให้นักศึกษาได้คะแนนจากโจทย์เหล่านั้นโดยที่ไม่ได้ประโยชน์และความรู้เท่าที่ควร ซึ่งอาจารย์มักจะให้ผู้ช่วยอาจารย์ในการช่วยตรวจสอบว่าโปรแกรมที่นักศึกษาได้ส่งมาถูกต้องหรือไม่ อีกกรณีอาจารย์ก็จะตรวจสอบเอง ทำให้เสียเวลาในการที่ต้องตรวจสอบนักศึกษาทุกคนว่าได้ทำถูกต้องตามที่อาจารย์กำหนดมาหรือไม่

โดยในปัจจุบันมีเครื่องมือต่าง ๆ ให้ใช้ในการตรวจสอบความถูกต้องของตัวโปรแกรม (Syntax) มากมาย ที่สามารถตรวจสอบขั้นตอนการทำงานของโปรแกรม รวมถึงวิธีการในการเขียนได้ ผู้จัดทำจึงเล็งเห็นและต้องการนำสิ่งเหล่านั้นมาใช้ให้เกิดประโยชน์กับการเรียนวิชาการเขียนโปรแกรม รวมถึงความสะดวกสบายของอาจารย์ผู้สอนในการตรวจสอบงานของนักศึกษาและให้คะแนน

ดังนั้นทางผู้จัดทำจึงอยากนำเครื่องมือในการตรวจสอบความถูกต้องของตัวโปรแกรมเหล่านั้น มาผนวกเข้ากับเว็บไซต์เกรดเดอร์ ให้เกิดประโยชน์ในวิชาการเขียนโปรแกรมในอนาคต เพื่อให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาจารย์ผู้สอนได้รับความสะดวกสบายในการตรวจสอบงานของนักศึกษาและนักศึกษาได้รับประโยชน์และความรู้จากรายวิชาที่เรียน

1.2 วัตถุประสงค์

- 1.2.1 เพื่อสร้างเว็บไซต์เกรดเดอร์ ที่สามารถตรวจสอบความถูกต้องของตัวโปรแกรมได้ภายในตัว โดยมุ่งเน้นไปที่ภาษา Python เท่านั้น
- 1.2.2 นำตัวเว็บไซต์ไปใช้งานในวิชาการเขียน โปรแกรมภาษา Python
- 1.2.3 เพื่อนำความรู้ที่มีในการทำเว็บไซต์มาประยุกต์ใช้จริงให้เกิดประโยชน์

1.3 เป้าหมายของโครงการ

- 1.3.1 ตัว Grader สมบูรณ์และใช้งานได้จริง
- 1.3.2 ได้นำไปใช้งานในวิชาการเขียน โปรแกรมที่ใช้ภาษา Python ในอนาคต
- 1.3.3 อาจารย์ที่นำไปใช้งาน ได้รับความสะดวกสบายจากตัว Grader จริง

1.4 ขอบเขตของโครงการ

เว็บแอปพลิเคชันสำหรับการเขียน โปรแกรมภาษา Python (เว็บไซต์เกรดเดอร์) เป็นเว็บไซต์สำหรับใช้ในการเรียนการสอนวิชาการเขียน โปรแกรม หน้าหลักจะเป็นการที่อาจารย์สามารถมอบหมายแบบฝึกหัดให้นักศึกษาทำผ่านตัวเว็บไซต์ และนักศึกษาสามารถส่งคืนให้อาจารย์ตรวจได้ โดยตัวเว็บไซต์สามารถตรวจสอบความถูกต้องของตัวโปรแกรมที่นักศึกษาส่งมาได้ การพัฒนาตัวเว็บไซต์จะแบ่งผู้ใช้งานออกเป็น 3 ส่วนได้แก่ ผู้ดูแลเว็บไซต์ อาจารย์ นักศึกษา รายละเอียดการพัฒนาแต่ละส่วนมีดังนี้

1.4.1 พัฒนาระบบการจัดการของผู้ดูแล

- 1.4.1.1 สามารถสร้างกลุ่มเรียน (Team) และจัดการระบบทุกอย่างของ Team ได้
- 1.4.1.2 สามารถแก้ไขกลุ่มเรียนที่อาจารย์สร้างขึ้นมาได้ ในกรณีจำเป็น
- 1.4.1.3 สามารถจัดการรายละเอียดต่างๆของแต่ละบัญชีผู้ใช้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.2 พัฒนาระบบการจัดการของอาจารย์

- 1.4.2.1 สามารถเข้าใช้งานตัวเว็บไซต์ผ่านทางอีเมลได้
- 1.4.2.2 สามารถแก้ไขรายละเอียดบัญชีผู้ใช้ของตนเองได้
- 1.4.2.3 สามารถสร้างกลุ่มเรียนและแก้ไข กลุ่มเรียนที่ตนเองสร้างได้
- 1.4.2.4 สามารถจัดการสมาชิกในกลุ่มเรียนได้ ทั้งการเชิญเข้าและการนำออก
- 1.4.2.5 สามารถสร้างแบบฝึกหัดในกลุ่มเรียน, กำหนดวันส่ง, และมอบหมายให้นักศึกษาในกลุ่มเรียนทำได้
- 1.4.2.6 สามารถตรวจสอบตัวโปรแกรมที่นักศึกษาแต่ละคนส่งมาได้
- 1.4.2.7 สามารถตรวจสอบคะแนนที่นักศึกษาทำได้ในแต่ละแบบฝึกหัด
- 1.4.2.8 สามารถตรวจสอบคะแนนของนักศึกษารายบุคคลได้

1.4.3 พัฒนาระบบของนักศึกษา

- 1.4.3.1 สามารถเข้าใช้งานตัวเว็บไซต์ได้
- 1.4.3.2 สามารถแก้ไขรายละเอียดบัญชีผู้ใช้ของตนเองได้
- 1.4.3.3 สามารถเข้าร่วมกลุ่มเรียนที่อาจารย์เชิญได้
- 1.4.3.4 สามารถทำแบบฝึกหัดในกลุ่มเรียนได้
- 1.4.3.5 สามารถเช็ครายละเอียดของแต่ละแบบฝึกหัดได้

1.4.4 พัฒนาระบบฐานข้อมูล

- 1.4.4.1 จัดเก็บข้อมูลของบัญชีผู้ใช้
- 1.4.4.2 จัดเก็บข้อมูลของกลุ่มเรียนและแบบฝึกหัดในแต่ละกลุ่มเรียน
- 1.4.4.3 จัดเก็บโค้ดต่างๆที่นักศึกษาได้ทำไว้ในแต่ละแบบฝึกหัด
- 1.4.4.4 จัดเก็บคะแนนของนักศึกษาแต่ละคนในแต่ละแบบฝึกหัด

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 React

React เป็น JavaScript Library (ส่วนขยาย) พัฒนาขึ้นโดย Facebook เป็นเครื่องมือที่ใช้ในการสร้างส่วนติดต่อผู้ใช้ (User Interface) โดยสามารถสร้าง Component (ส่วนประกอบของส่วนติดต่อผู้ใช้งาน) แยกกัน ซึ่งแต่ละ Component นั้นจะมีความทำงานแยกจากกันอย่างอิสระ และสามารถนำแต่ละ Component มาใช้ซ้ำได้โดย React นั้นสามารถสร้างได้ทั้ง Single Page Application หรือ Mobile Application ด้วย React Native

การเขียน React นั้นจะเขียนในรูปแบบของ JSX ซึ่งเป็นการพัฒนาเพิ่มเติมจากภาษา JavaScript โดยตัว JSX เป็นการผสมผสานระหว่าง JavaScript และ HTML ซึ่งจะช่วยให้นักพัฒนาสามารถเขียนโปรแกรมได้สะดวกสบายมากขึ้นเนื่องจากการสร้างส่วนติดต่อผู้ใช้ ต้องใช้ภาษา HTML ส่วนการออกแบบ Logic การทำงานต้องใช้ภาษา JavaScript ทำให้สามารถเขียนไปพร้อมๆกันและถูกนำไปสร้างเป็น React Element (ส่วนประกอบ) จากนั้นจะถูกนำไปแสดงผลผ่าน DOM (Document Object Model หรือ Object ที่ใช้จัดเก็บโครงสร้าง HTML) และจะมีการอัปเดตตามการเปลี่ยนแปลงของ React Element นั้น ๆ

เหตุผลหลักๆที่เลือกใช้เพราะ React รองรับ Library อื่นๆได้เยอะ เป็นที่นิยมอย่างกว้างขวาง สามารถหาข้อมูลการแก้ไขปัญหาต่างๆได้ง่ายบนอินเทอร์เน็ตเนื่องจากมีทั้ง Document บอกรายละเอียดต่างๆของ React และมีสังคมบนโลกอินเทอร์เน็ตที่ช่วยอธิบายการแก้ไขปัญหาที่บุคคลส่วนใหญ่พบเจอ

2.2 Tailwind CSS (CSS Framework)

CSS Framework เป็นเครื่องมือที่ช่วยให้นักพัฒนาสามารถออกแบบเว็บไซต์ได้รวดเร็วขึ้น เพราะตัว Framework จะเตรียมโครงสร้างพื้นฐานของตัวโปรแกรม ทำให้ไม่จำเป็นต้องเสียเวลาไปกับการเขียนโปรแกรมซ้ำ ๆ โดยทั่วไปตัว CSS Framework จะมาพร้อมกับตัว HTML และ JavaScript ซึ่งการใช้งานหลักๆสามารถนำคำสั่งโปรแกรมที่ตัว Framework (ส่วนขยาย) มีให้ไปใส่ในจุดที่ต้องการ ก็ทำให้โปรแกรมทำงานได้ตามที่ต้องการ โดยประโยชน์หลัก ๆ ของ CSS Framework มีดังนี้

1) ออกแบบเว็บไซต์ได้รวดเร็วขึ้น

เพราะตัว Framework มี Class ต่างๆที่ถูกสร้างไว้ให้พร้อมใช้งาน ซึ่งสามารถใช้งานได้ทันทีในกรณีที่ต้องการเปลี่ยน User Interface เช่น สี ขนาด การจัดวาง หรือปุ่มต่างๆ ทำให้นักพัฒนาไม่จำเป็นต้องแก้ไขรายละเอียดที่ยิบย่อยจนเกินไป

2) ลดปัญหาการเกิด Software bug (จุดบกพร่องในการทำงานของโปรแกรม)

ตัว CSS Framework ได้ออกแบบมาให้รองรับเบราว์เซอร์หลายๆแบบเช่นการใช้งานระหว่างคอมพิวเตอร์และโทรศัพท์มือถือ ทำให้ลดโอกาสในการเกิด Software bug

3) มีความยืดหยุ่นในการแก้ไขตัวโปรแกรม

เนื่องจากมี Class ต่างๆเตรียมไว้พร้อมแล้วใน Framework ทำให้นักพัฒนาสะดวกสบายในการแก้ไข User Interface ให้เป็นไปตามการออกแบบได้ใกล้เคียงที่สุดโดยใช้เวลาไม่นาน

ส่วน Tailwind CSS เป็นหนึ่งใน CSS Framework ที่ช่วยให้นักพัฒนาสร้าง User Interface ที่สำคัญได้ด้วยตนเองอย่างรวดเร็ว และยังสามารถปรับแต่งในรายละเอียดได้ง่าย เนื่องจากมี Class ต่างๆที่ถูกสร้างไว้ให้พร้อมใช้งาน ซึ่งสามารถใช้งานได้ทันทีในกรณีที่ต้องการเปลี่ยน User Interface เช่น สี ขนาด การจัดวาง หรือปุ่มต่างๆ ทำให้นักพัฒนาไม่จำเป็นต้องเข้าไปแก้ไขที่ไฟล์ CSS หลัก แต่สามารถทำผ่านส่วนของไฟล์ HTML ได้ในทันที

โดยเหตุผลหลักๆที่เลือก Tailwind CSS มีดังนี้

- 1) ปรับแต่งได้สะดวกและหลากหลาย (ต่างจาก Framework อื่นๆที่มาพร้อมกับ HTML และ JavaScript)
- 2) ทำ Responsive Design ได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Redux

Redux คือ Library(ส่วนขยาย) ทำหน้าที่เป็นตัวจัดการ State (State Management) สำหรับ JavaScript Application ซึ่งไม่จำเป็นต้องใช้แค่กับ React ตัวของ Redux สามารถประยุกต์ใช้ได้แทบทั้งหมด เช่นใช้กับ Angular หรือ Vue

โดยปกติการเขียน React ในการจัดการ State และ Props จะเกิดปัญหาใน Application ขนาดใหญ่ เมื่อต้องการส่งค่า State ไปมาระหว่างหลายๆ Components ซึ่ง Redux มีไว้เพื่อจัดการปัญหานี้

Redux มีส่วนประกอบ 3 ส่วน

- 1) Store เป็น Object ไว้จัดเก็บข้อมูล State ต่างๆ
- 2) Action เป็นข้อมูลการทำงานที่ส่งจาก Application มาที่ Store เพื่อเปลี่ยนแปลงหรือจัดเก็บ State
- 3) Reducer เป็นตัวกำหนด State ของตัว Application ผ่านการประมวลผลที่มาจาก Action ก่อนที่จะนำมาจัดเก็บใน Store

โดยหลักการในการทำงานของ Redux มีอยู่ 3 อย่าง ได้แก่

1) Single Source of Truth

โดย State ของ Application ต้องเก็บไว้ใน Store เดียว โดยกำหนดรูปแบบเป็น JavaScript Object (ตัวแปรที่เก็บค่าได้หลายค่า)ทุกอย่างไป

2) State is read-only

การเพิ่มข้อมูลของ State สามารถทำได้ผ่าน Action ทางเดียวเท่านั้น โดยในโปรแกรม 2.6 dispatch เป็น Function ที่กำหนดให้ตัว Action ทำการอัปเดต State ตาม type ที่กำหนดในตัว reducers

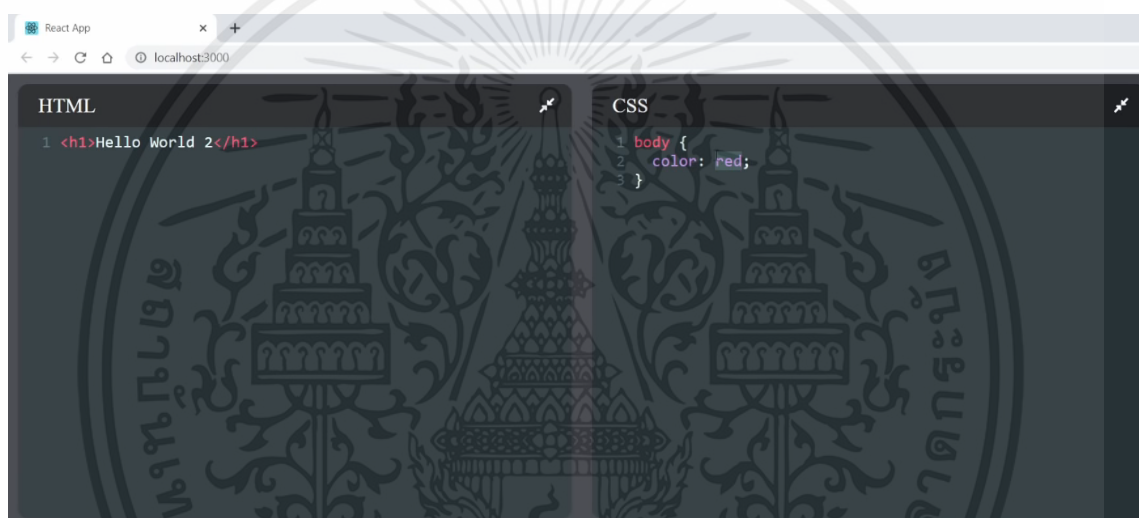
3) Changes are made with pure functions

การเปลี่ยนแปลง State จำเป็นต้องใช้ Pure function เท่านั้น หรือก็คือ ในการเปลี่ยน State จำเป็นต้องส่งค่าใหม่ของ State ที่ต้องการเปลี่ยนผ่านตัว Reducers (ไม่สามารถแก้ไขค่าเดิม แต่เป็นการส่งค่าใหม่เข้ามาแทน)

2.4 Code Mirror

Code Mirror เป็นหนึ่งใน JavaScript Component (ส่วนขยาย) ที่มีไว้สำหรับการสร้าง Code editor (โปรแกรมที่ถูกออกแบบมาเพื่อการเขียนโค้ด เรียกอีกอย่างว่า Text Editor) บนเว็บไซต์ โดยมีรูปแบบการใช้งานหลากหลายเพื่อให้สะดวกต่อการใช้งานหลายๆประเภท รองรับการใช้งานบนเบราว์เซอร์เกือบทุกแบบ

ตัว Code editor ของ Code Mirror รองรับการใช้งานโปรแกรมได้หลายภาษา เช่น HTML, CSS, JavaScript, Python รวมถึงมี Theme สีให้เลือกหลากหลาย, เลือก Style Font ได้มาก, รวมถึงเหตุผลอื่นๆอีกมากมายทำให้นักพัฒนาส่วนใหญ่ที่ต้องการใช้ Code editor บนเว็บไซต์เลือกใช้



รูป 2.1 ตัวอย่างของการใช้ Code Mirror กับภาษา HTML และ CSS

เหตุผลที่เลือกใช้ Code Mirror เพราะมีความสามารถครอบคลุมและตรงจุดในการใช้งาน โดยคุณสมบัติหลักของ Code Mirror มีดังนี้

- 1) มีการแยกสีตัวอักษรของคำสั่งในโปรแกรมอย่างชัดเจน
- 2) แสดงเลขบรรทัดให้นักพัฒนาเห็น
- 3) ในบางภาษาจะมีระบบ Autocompletion ให้ (เติมข้อความอัตโนมัติ)
- 4) ระบบย่อขยายส่วนของโปรแกรมที่อยู่ในวงเล็บ (บางส่วนเท่านั้น)
- 5) ระบบจับคู่วงเล็บปีกกา ใช้ในการย่อขยายโปรแกรม
- 6) สามารถค้นหาส่วนของโปรแกรมที่เขียนไปแล้วต้องการแก้ไขได้
- 7) รองรับการใช้งานบนโทรศัพท์มือถือบางระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 Django Rest Framework

Django คือ free open source, Python-based web framework ที่มีรูปแบบสถาปัตยกรรมแบบ Model-View-Template ที่ช่วยลดความยุ่งยากในการพัฒนาเว็บแอปพลิเคชันทำให้สามารถเขียนและสร้างแอปพลิเคชันได้อย่างรวดเร็ว โดย Model คือ ระบบจัดการฐานข้อมูล Template คือ หน้าตาของเว็บที่ใช้แสดงผลว่าแต่ละหน้าจะออกมาเป็นอย่างไร View คือ ส่วนสำหรับจัดการข้อมูลเพื่อนำไปแสดงผลใน template

REST (Representational State Transfer) คือ สถาปัตยกรรมอย่างหนึ่งของซอฟต์แวร์ที่อยู่บนพื้นฐานของ HTTP Protocol ที่เกี่ยวข้องกับการส่งข้อมูลระหว่าง Client และ Server

Django REST framework คือ Library ของภาษา Python ที่ใช้ในการสร้าง RESTful APIs (Representational State Transfer) ได้อย่างรวดเร็วและสมบูรณ์แบบ ประโยชน์สำคัญของ Django REST framework คือการทำให้การ serialization ได้ง่ายขึ้นอย่างมาก และ เหตุผลอื่นๆ เช่น

- 1) Django REST framework มี Web browsable API ที่เป็นตัวช่วยให้การพัฒนาเป็นไปอย่างมีประสิทธิภาพ
- 2) Django Rest framework สามารถสร้าง RESTful APIs ได้อย่างรวดเร็ว
- 3) มี Authentication policies สำหรับ OAuth (เป็นมาตรฐานแบบเปิดเกี่ยวกับการยืนยันสิทธิ์ และการจัดการสิทธิ์การใช้ระบบ) ที่ช่วยในการพัฒนา Web APIs
- 4) Serialization นั้นรองรับแหล่งข้อมูลทั้ง ORM (Object / Relational Mapping หมายถึง การ map ข้อมูลในตารางข้อมูลของฐานข้อมูลให้อยู่ในรูปของ object-oriented language ซึ่งจะเป็นการสร้าง Database แบบเสมือนขึ้นมาให้อยู่ในรูปภาษาโปรแกรม) และ non-ORM
- 5) มี Class Based Views ที่มีฟังก์ชันสำเร็จรูปต่าง ๆ พร้อมใช้งาน
- 6) Django REST framework สามารถปรับแต่งได้อิสระตามที่นักพัฒนาต้องการ
- 7) ง่ายต่อการเรียนรู้เพราะภาษา Python มีไวยากรณ์ที่อ่านเข้าใจง่ายเหมือนภาษาอังกฤษ
- 8) ได้รับความนิยมสูงและมีกลุ่มผู้ใช้งานขนาดใหญ่ มีแหล่งเรียนรู้หลากหลาย ไม่ว่าจะเป็นบทความ คลิปฝึกสอนและแหล่งเรียนรู้อื่น ๆ เพิ่มเติม
- 9) มีคู่มือเอกสารที่สมบูรณ์ เขียนไว้ได้อย่างครอบคลุมและละเอียด ทำให้สามารถใช้เป็นแหล่งอ้างอิงได้เป็นอย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

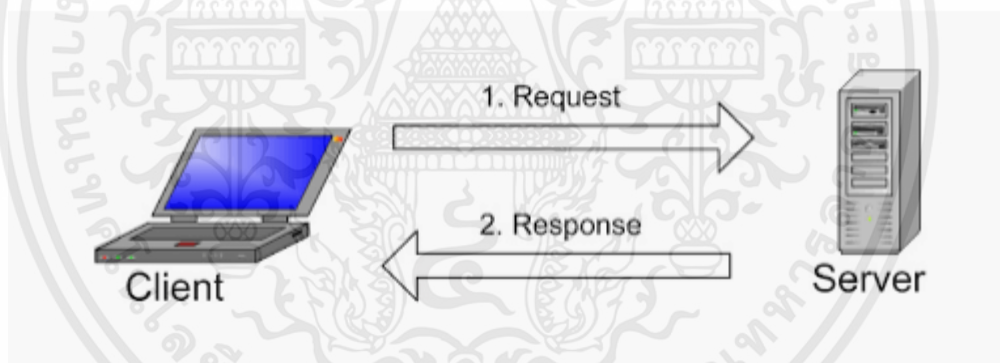
2.6 SQLite

SQLite เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) ที่ใช้ภาษา SQL เช่นกัน แต่แตกต่างจาก MySQL ตรงที่ SQLite เป็นฐานข้อมูลแบบ Embedded คือ ไม่ต้องการตัว Server ด้านนอกในการทำงาน ซึ่งทำให้ SQLite มีขนาดเล็กและประหยัดทรัพยากร ใช้งานง่าย และมีความเป็นอิสระมากขึ้น

SQLite เหมาะกับแอปพลิเคชันขนาดเล็กและปานกลางที่ต้องการความกระชับ ความสะดวกสบาย ไม่ต้องการความซับซ้อนของการตั้งค่าและจัดการ Server ฐานข้อมูล ซึ่งทำให้เหมาะสำหรับการพัฒนาแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ เช่น สมาร์ทโฟน แท็บเล็ต หรืออุปกรณ์ IoT

การทำงานของ SQLite โดยคร่าวๆ คือ ฐานข้อมูล SQLite จะถูกสร้างและจัดเก็บเป็นไฟล์เดี่ยวบนระบบไฟล์ ผู้ใช้งานสามารถเข้าถึงข้อมูลในฐานข้อมูล SQLite ได้โดยตรงผ่าน API หรือ Library ของ SQLite ที่มีให้ใช้งานในหลายภาษาการพัฒนา เช่น C, C++, Python, Java และอื่นๆ ไม่ต้องผ่านตัวกลางอื่นใด เนื่องจากไม่ต้องการตัว Server รองรับ

รวมถึง SQLite ไม่มีค่าใช้จ่ายในการใช้งาน เนื่องจากเป็นซอฟต์แวร์ประเภท Public Domain สามารถนำไปใช้งานและแจกจ่ายได้โดยไม่มีข้อจำกัดใดๆ



รูป 2.2 การติดต่อระหว่าง Server และผู้ใช้งาน

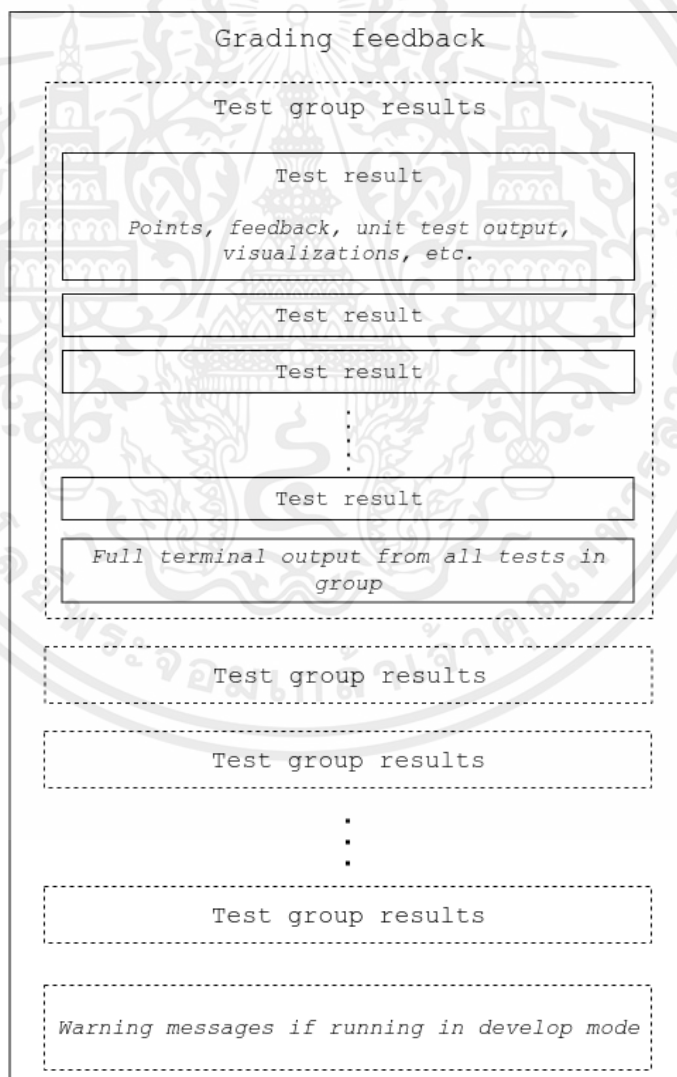
2.7 Python-grader-utils

Python-grader-utils เป็นไลบรารีของภาษา Python ที่ให้ความสามารถในการจัดการกับการตรวจคะแนน การตรวจสอบไฟล์ และการจัดรูปแบบข้อมูลผลตอบรับจากการทดสอบ ได้รับการพัฒนาขึ้นเพื่อรองรับการตรวจคะแนนของแบบฝึกหัดทางโปรแกรม โดยมีคุณสมบัติคือ

- 1) ช่วยให้สามารถเรียกใช้กรณีทดสอบที่ถูกกำหนดไว้ในคลาสย่อยของ unit test โดยจะประมวลผลทดสอบและสร้างผลลัพธ์เป็น JSON ที่สามารถแปลงเป็นรูปแบบ HTML เพื่อการนำเสนอที่ดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) สามารถตรวจสอบไฟล์ต่างๆก่อนการประมวลผลทดสอบ เพื่อให้มั่นใจว่าไฟล์ที่ส่งเข้ามาตรงตามความต้องการหรือข้อจำกัดที่กำหนด เช่น สามารถตรวจสอบไฟล์ตาม blacklist หรือ whitelist ของไวยากรณ์ Python ที่อนุญาตโดยใช้ชื่อโหนดในต้นไม้สัญลักษณ์ของ Abstract Syntax Tree (AST)
- 3) ช่วยให้สามารถจำกัดการใช้ไวยากรณ์ Python บางส่วนได้โดยกำหนด blacklist หรือ whitelist ของชื่อโหนดในต้นไม้สัญลักษณ์ (AST node names) ซึ่งช่วยในการบังคับมาตรฐานการเขียนโค้ดหรือป้องกันการใช้โครงสร้างที่ถูกจำกัดในไฟล์ที่ส่งเข้ามา
- 4) สามารถจัดรูปแบบ Traceback และข้อความข้อผิดพลาดเพื่อให้รวมข้อมูลที่สำคัญเท่านั้น ทำให้ง่ายต่อการเข้าใจและวินิจฉัยข้อผิดพลาด ไลบรารียังมีความสามารถในการเข้าถึง Traceback เต็มรูปแบบ (unformatted traceback) หากจำเป็น



รูป 2.3 โครงสร้างของ JSON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน Python-grader-utils จะต้องมีการเขียน unit test และ config เพื่อใช้ส่งให้ระบบสามารถทำการตรวจสอบได้โดยอัตโนมัติ โดยรายละเอียดของไฟล์ต่างๆที่จำเป็นต้องใช้ในการตรวจสอบ มีดังนี้

- ไฟล์จากตัว grader-utils โดยเป็นไฟล์ที่ใช้ในกระบวนการทดสอบแบบอัตโนมัติโดยใช้เครื่องมือ grader-utils ที่มีความสามารถเพิ่มเติมเหนือจาก unit test ทั่วไป
- ไฟล์ unit test ที่ผู้ใช้เขียน เป็นไฟล์ที่ใช้ในการเขียนและรวบรวมเคสทดสอบ (test cases) สำหรับการทดสอบโมดูลหรือโค้ดที่ต้องการทดสอบ โดยในไฟล์นี้สามารถเพิ่มเคสทดสอบเพิ่มเติมหรือแก้ไขเคสทดสอบที่มีอยู่ได้
- ไฟล์ตัวอย่างโค้ดที่ผู้ใช้เขียน เพื่อใช้เป็น โมเดลหรือฟังก์ชันที่เกี่ยวข้องกับการทดสอบ
- ไฟล์โค้ดจากฝั่งคนส่งคำตอบจากโจทย์ที่กำหนด โดยทดสอบว่าโค้ดที่เขียนตรงตามคำอธิบายและข้อกำหนดที่ได้รับจากทั้งตัวไฟล์ unit test และไฟล์ตัวอย่างโค้ดหรือไม่ โค้ดในไฟล์นี้จะถูกตรวจสอบโดยเคสทดสอบที่ระบุในไฟล์ unit test
- ไฟล์ config เป็นไฟล์ที่ใช้ในการกำหนดการตรวจสอบและการตรวจสอบว่าโมดูลและโค้ดที่ผู้ใช้ส่งมีการใช้ไวยากรณ์ที่ถูกต้องหรือไม่ ไฟล์นี้เป็นส่วนหนึ่งของกระบวนการตรวจสอบแบบอัตโนมัติ (automated grading) โดยใช้โมดูลของตัว grader-utils ในการประมวลผลและประเมินคะแนนของการส่งงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 งานวิจัยที่เกี่ยวข้อง (Literature Review)

2.8.1. Automated Assessment of Python Code

งานวิจัยชิ้นนี้เป็นงานวิจัยเกี่ยวกับการประเมินความสมบูรณ์ของโปรแกรมโดยส่วนที่สำคัญที่สุดของการประเมินโปรแกรมคือการตรวจสอบความถูกต้องของผลลัพธ์และการทำงานของโปรแกรมซึ่งโปรแกรมควรที่จะสามารถคาดเดาได้และสามารถนำไปใช้ซ้ำได้ในสภาพแวดล้อมที่มีองค์ประกอบแบบเหมือนกัน สามารถทดสอบได้โดยการใช้ unit test

Unit test โดยปกติคือส่วนฟังก์ชันเล็กๆที่มีไว้เพื่อประเมินโปรแกรมในส่วนที่เจาะจง ใน python มี standard library สำหรับ Unit test คือ Unit test module ที่คอยช่วยในการเขียนและรัน Unit test และยังมีทางเลือกอื่นๆอีกที่ให้ความยืดหยุ่นและรวดเร็วกว่าเช่น Pytest

งานวิจัยยังกล่าวถึงการประเมินโปรแกรมที่ดี โดยโปรแกรมที่ดีหมายถึงโปรแกรมที่ต้องจัดการกับข้อผิดพลาดที่เกิดขึ้นได้อย่างมีประสิทธิภาพและสามารถแก้ไขความผิดพลาดได้ลักษณะของโปรแกรมที่ดีคือโปรแกรมต้องถูกเขียนขึ้นตามลักษณะแนวทางของการเขียนโปรแกรมเช่น แนวทางการเขียนโปรแกรมแบบ PEP8 สามารถตรวจสอบลักษณะการเขียนโปรแกรมที่ดีได้โดยการใช้ linter เช่น PyLint และ Flake

อีกส่วนสำคัญของระบบประเมินผล คือใช้คำแนะนำเพื่อให้สามารถนักศึกษาสามารถปรับปรุงพัฒนาทักษะการเขียนโปรแกรมให้ดียิ่งขึ้น และการได้รับคำแนะนำจากนักศึกษาเพื่อใช้ในการปรับปรุงและพัฒนาโปรแกรม

2.8.2. Automatic Assessment of Programming Assignments

งานวิจัยชิ้นงานวิจัยชิ้นนี้เป็นงานวิจัยเกี่ยวกับการรวบรวมปัจจัยที่เกี่ยวข้องกับการประเมินโปรแกรมผ่านการวิเคราะห์จากประสบการณ์ของนักวิจัยมากมายและอธิบายการออกแบบของตัวโปรแกรมการประเมินความสมบูรณ์ของโปรแกรมรวมถึงบอกข้อควรระวังในการออกแบบ

โดยสรุป ปัจจัยความสำเร็จในการเขียนโปรแกรมการประเมินความสมบูรณ์ของโปรแกรมมีอยู่ 6 ข้อคือ

- 1) คุณภาพของโจทย์
- 2) มีโจทย์ที่ชัดเจน
- 3) เลือกข้อมูลที่จะนำมาทดสอบให้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) มีคำแนะนำที่ดี
- 5) สามารถส่งคำตอบได้ไม่จำกัด
- 6) ความเข้าใจของนักเรียน

ปัญหาที่จะเจอคือการหา test case ที่ครบถ้วนในโจทย์ที่ยากนั้นต้องใช้ความพยายามอย่างมาก ในกรณีการออกแบบ โจทย์ประเภทเติมคำตอบนั้นมีประโยชน์ แต่จะเป็นการตีกรอบความคิดที่ในการแก้ปัญหาเพราะในความเป็นจริงการแก้ปัญหามีหลายวิธีและอาจค้นพบวิธีการแก้ปัญหที่ดีกว่าเดิมได้โดยการใช้ความคิดมากขึ้น

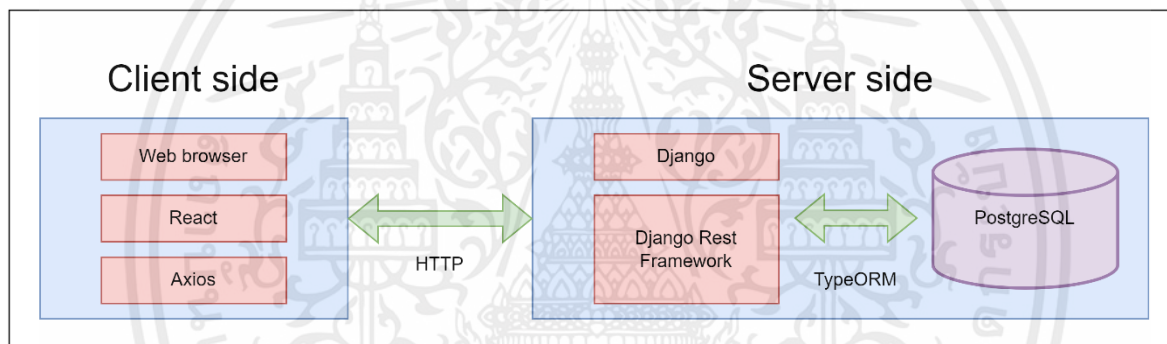
อีกหนึ่งปัญหาคือต้องคำนึงถึงความปลอดภัยในการรับ โปรแกรมที่เขียนเข้ามา เพราะบางกรณีอาจเป็นอันตรายต่อตัว โปรแกรมการประเมินความสมบูรณ์ของ โปรแกรม (เช่นการเขียนโปรแกรมที่วนการทำงานแบบไม่จำกัดเข้ามา อาจทำให้ระบบล่ม)

บทที่ 3

การออกแบบ

3.1 โครงสร้างของระบบ

ในหัวข้อโครงสร้างของระบบจะอธิบายถึงการเชื่อมต่อกันในแต่ละส่วนของเว็บไซต์ระบบ ประเมินผลสำหรับการเขียนโปรแกรมไพธอน ตั้งแต่ผู้ใช้งานเข้าสู่ระบบของเว็บแอปพลิเคชัน, เว็บแอปพลิเคชันติดต่อไปยังระบบฟังก์ชันเบื้องหลัง และการติดต่อกับระบบฐานข้อมูล



รูป 3.1 System design

จากรูป 3.1 เป็นแผนภาพแสดงการเชื่อมต่อระหว่างฝั่งผู้ใช้งานบนเว็บแอปพลิเคชันกับระบบฟังก์ชันเบื้องหลังไปจนถึงระบบฐานข้อมูล โดยมีชื่อเครื่องมือหลัก ๆ ที่ใช้แสดงอยู่ด้วย โดยจะแบ่งการอธิบายเป็น 2 ส่วนหลักๆ คือส่วนของเว็บแอปพลิเคชัน และส่วนของระบบเบื้องหลัง

3.1.1 เว็บแอปพลิเคชัน (Frontend)

หลักการทำงานเริ่มจากการที่ผู้ใช้งานเลือกใช้เบราว์เซอร์ต่าง ๆ ในการเข้าถึงเว็บแอปพลิเคชัน ซึ่งเว็บเบราว์เซอร์จะได้รับไฟล์ Javascript มาหลังจากนั้นจะเป็นหน้าที่ของ React ในการที่จะทำการแสดงผล user interface ตามแต่ละ URL ที่ผู้ใช้งานเข้าถึง เมื่อผู้ใช้งานใช้การทำงานที่ต้องเรียกระบบฐานข้อมูล ตัว React จะทำการจัดการสร้าง HTTP request ขึ้นมาและส่ง request นั้นๆ ไปที่ระบบเบื้องหลังเพื่อจัดการการทำงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

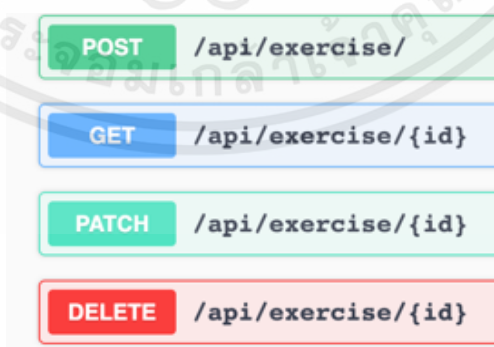
3.1.2 ระบบเบื้องหลัง (Backend)

หลักการทำงานในส่วนนี้จะเริ่มทำงานจากเมื่อเซิร์ฟเวอร์ได้รับ HTTP request เข้ามาทาง path หรือ routing ที่กำหนดไว้ไว้หลักจากนั้นตัว class view หรือ controller ของ Django จะทำหน้าที่ในการจัดการกับคำขอต่างๆ และในการแปลงข้อมูลจาก request ที่ส่งมาในรูปแบบ JSON เป็น datatype ของ python ทาง Django REST framework ก็ได้มี class serializer มาให้เพื่อเปลี่ยนรูปแบบของข้อมูลให้เหมาะสมในการคำนวณหรือประมวลผลต่อไป หากต้องการเข้าถึงข้อมูลในฐานข้อมูลก็จะเรียกใช้ไปที่ class model ก็คือระบบ ORM ของ Django จะทำการ query ข้อมูลจากฐานข้อมูลมาให้หรือหากไม่มีข้อมูลระบบก็จะทำการประมวลผลแล้วส่งข้อมูลผลลัพธ์กลับไป

3.1.3 ความสัมพันธ์ระหว่าง Frontend และ Backend

การเชื่อมต่อจะใช้ Axios สำหรับเชื่อมต่อกับ API Service เพื่อการรับส่งข้อมูลแบบ RESTful API โดยที่ Axios ทำหน้าที่เป็นตัวกลางในการจัดการ method, data, headers, security และอื่น ๆ

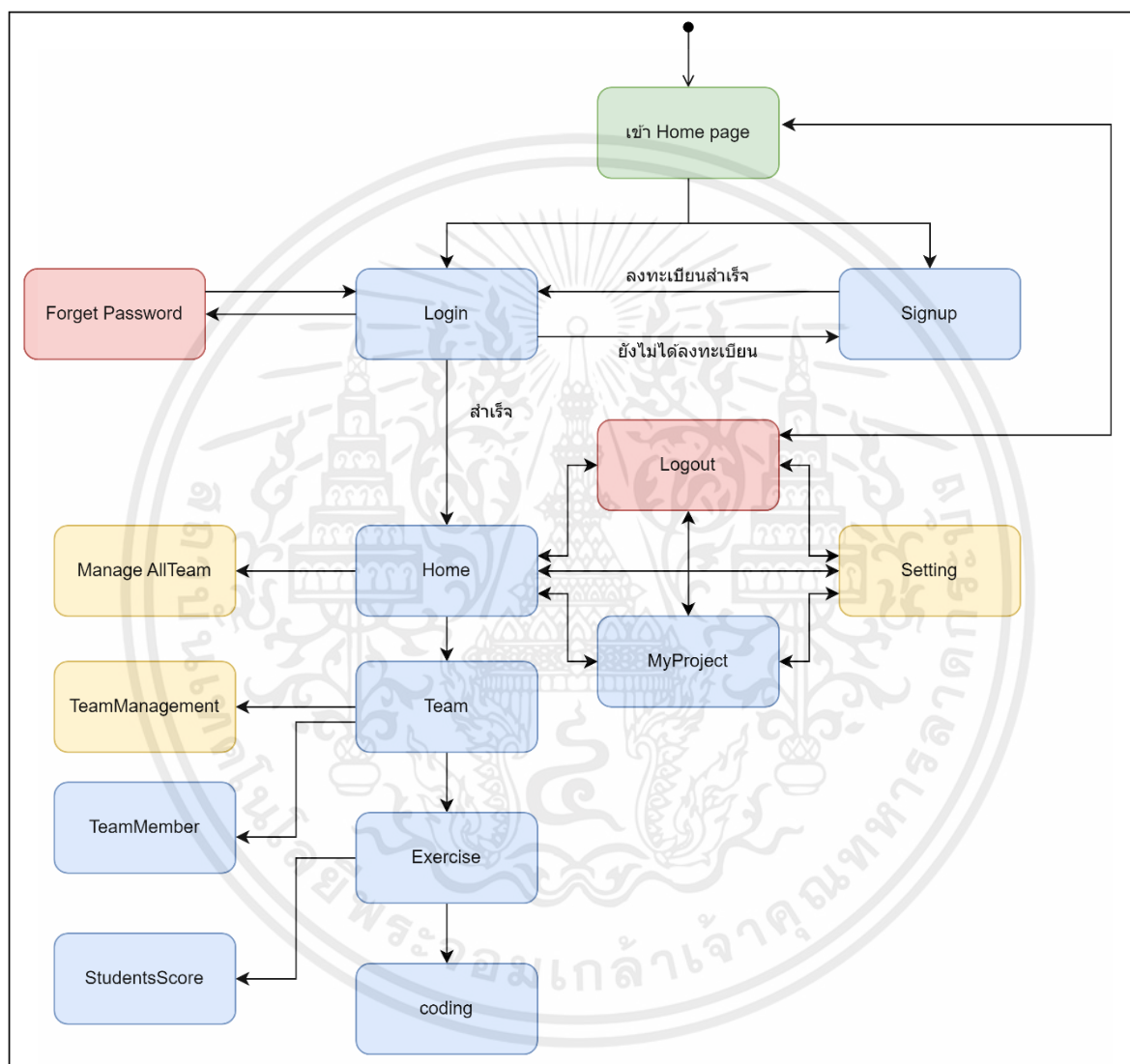
ในการเรียกข้อมูลจาก Backend ตัว Frontend จะทำการสร้าง HTTP request โดยใช้ ตัว Axios เข้ามาช่วย และส่ง Request นั้นๆผ่าน URL ที่หลังบ้านสร้างไว้รองรับ และให้ระบบ Backend ส่งข้อมูลกลับมาตามที่ต้องการ ในรูปจะเป็นตัวอย่างของ Api ในการสร้าง เรียกใช้ แก้ไข และลบแบบฝึกหัด



รูป 3.2 ตัวอย่าง api การเรียกใช้แบบฝึกหัด

3.2 Flow Diagram

Flow diagram ของเว็บไซต์ระบบประเมินผลสำหรับการเขียน โปรแกรมไพธอน ที่ออกแบบไว้ เป็น conceptual ให้เห็นภาพการทำงาน โดยรวม มีรายละเอียดดังนี้

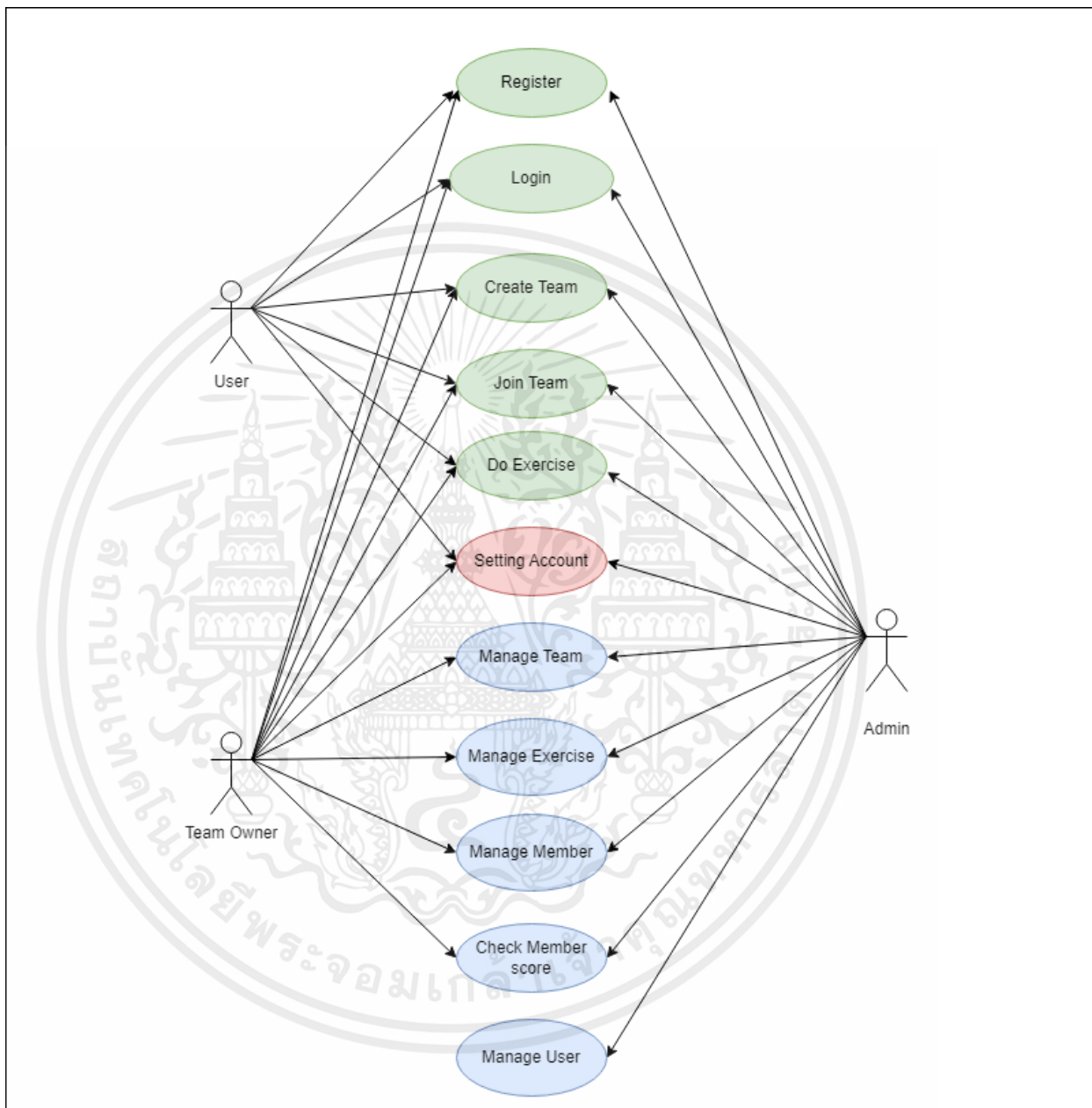


รูป 3.3 Flow diagram ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 Use case Diagram

Use case ของเว็บไซต์ระบบประเมินผลสำหรับการเขียนโปรแกรมไพธอน มี User 2 แบบคือ ผู้ใช้งาน(นักศึกษากับอาจารย์), และผู้ดูแลระบบ โดยมีรายละเอียดดังต่อไปนี้



รูป 3.4 Use case diagram ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.1 Use case การสมัครสมาชิก (Register)

Use case: สมัครสมาชิก
Actor: ผู้ใช้งาน
Pre-condition: ยังไม่มีบัญชีผู้ใช้งาน
Flow of Events: 1. กรอก Email, Passwords, ชื่อ-นามสกุล, รหัสนักศึกษา(กรณีที่เป็นนักศึกษา) 2. กดปุ่มสมัครสมาชิก
Post-condition: เพิ่มข้อมูลบัญชีผู้ใช้งานและสามารถเข้าใช้งานระบบได้

ตาราง 3.2 Use case การเข้าสู่ระบบ (Login)

Use case: เข้าสู่ระบบ
Actor: ผู้ใช้งาน
Pre-condition: มีบัญชีผู้ใช้งาน
Flow of Events: 1. กรอก Email, Passwords 2. กดปุ่มเข้าสู่ระบบ
Post-condition: เข้าใช้งานระบบ

ตาราง 3.3 Use case การเข้ากลุ่มเรียน (Join team)

Use case: เข้ากลุ่มเรียน
Actor: ผู้ใช้งาน
Pre-condition: ลงชื่อเข้าใช้แล้ว
Flow of Events: 1. กดปุ่มสร้าง/เข้าร่วมกลุ่มเรียน 2. กรอก Id ของกลุ่มเรียนนั้น 3. กดตกลง
Post-condition: เข้าร่วมกลุ่มเรียนสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 3.4 Use case การทำแบบฝึกหัด (Do Exercise)

Use case: ทำแบบฝึกหัด
Actor: ผู้ใช้งาน
Pre-condition: ลงชื่อเข้าใช้และเข้าร่วมกลุ่มเรียนแล้ว
Flow of Events: 1. เข้าไปที่กลุ่มเรียน 2. เลือกแบบฝึกหัดที่มีอยู่ในกลุ่มเรียน 3. เริ่มทำแบบฝึกหัด 4. กดบันทึกในกรณีที่ทำเสร็จแล้วหรือต้องการทำต่อในภายหลัง
Post-condition: ทำแบบฝึกหัดสำเร็จ

ตาราง 3.5 Use case การจัดการกลุ่มเรียนที่ผู้ใช้งานเข้าร่วม/เป็นผู้ดูแล (Manage Team)

Use case: การจัดการกลุ่มเรียน
Actor: เจ้าของกลุ่มเรียน
Pre-condition: ลงชื่อเข้าใช้งานแล้ว
Flow of Events: 1. เข้าไปที่หน้าจัดการกลุ่มเรียน 2. เพิ่มกลุ่มเรียนใหม่/แก้ไขกลุ่มเรียนที่ผู้ใช้งานเป็นผู้ดูแล/ออกจากกลุ่มเรียนที่ผู้ใช้งานอยู่ 3. บันทึกข้อมูล
Post-condition: ข้อมูลกลุ่มเรียนของผู้ใช้งานถูกเพิ่ม/แก้ไข

ตาราง 3.6 Use case การจัดการแบบฝึกหัดในกลุ่มเรียนที่ผู้ใช้งานเป็นผู้ดูแล (Manage Exercise)

Use case: การจัดการแบบฝึกหัด
Actor: เจ้าของกลุ่มเรียน
Pre-condition: ลงชื่อเข้าใช้งานแล้วและเป็นผู้ดูแลกลุ่มเรียน
Flow of Events: 1. เข้าไปที่กลุ่มเรียน 2. เข้าไปที่หน้าการจัดการแบบฝึกหัด 3. เพิ่ม/แก้ไขแบบฝึกหัดในกลุ่มเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. กดปุ่มบันทึกข้อมูล
Post-condition: ข้อมูลแบบฝึกหัดในกลุ่มเรียนของผู้ใช้งานถูกเพิ่ม/แก้ไข

ตาราง 3.7 Use case การจัดการสมาชิกในกลุ่มเรียนที่ผู้ใช้งานเป็นผู้ดูแล (Manage Member)

Use case: การจัดการสมาชิก
Actor: เจ้าของกลุ่มเรียน
Pre-condition: ลงชื่อเข้าใช้งานแล้วและเป็นผู้ดูแลกลุ่มเรียน
Flow of Events: <ol style="list-style-type: none"> 1. เข้าไปที่กลุ่มเรียน 2. เข้าไปที่หน้าการจัดการสมาชิกในกลุ่ม 3. เพิ่ม/แก้ไขสมาชิกในกลุ่มเรียน 4. กดปุ่มบันทึกข้อมูล
Post-condition: ข้อมูลสมาชิกในกลุ่มเรียนของผู้ใช้งานถูกแก้ไข

ตาราง 3.8 Use case การตรวจสอบคะแนนของสมาชิกในกลุ่มเรียนที่ผู้ใช้งานเป็นผู้ดูแล (Check Member Score)

Use case: การตรวจสอบคะแนนของสมาชิก
Actor: เจ้าของกลุ่มเรียน
Pre-condition: ลงชื่อเข้าใช้งานแล้วและเป็นผู้ดูแลกลุ่มเรียน
Flow of Events: <ol style="list-style-type: none"> 1. เข้าไปที่กลุ่มเรียน 2. เข้าไปที่หน้าแบบฝึกหัดที่ต้องการเช็คคะแนน 3. สามารถตรวจสอบคะแนนของสมาชิกแต่ละคนว่าทำแบบฝึกหัดได้คะแนนเท่าไร
Post-condition: ได้รู้คะแนนของสมาชิกแต่ละคนในกลุ่มเรียน

ตาราง 3.9 Use case การตั้งค่าบัญชี (Setting Account)

Use case: การตั้งค่าบัญชี
Actor: ผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pre-condition: ลงชื่อเข้าใช้งานแล้ว
Flow of Events: 1. เลือกเมนูการตั้งค่า 2. แก้ไขข้อมูลการตั้งค่าต่างๆของระบบ 3. กดปุ่มบันทึกข้อมูล
Post-condition: ข้อมูลการตั้งค่าถูกบันทึก

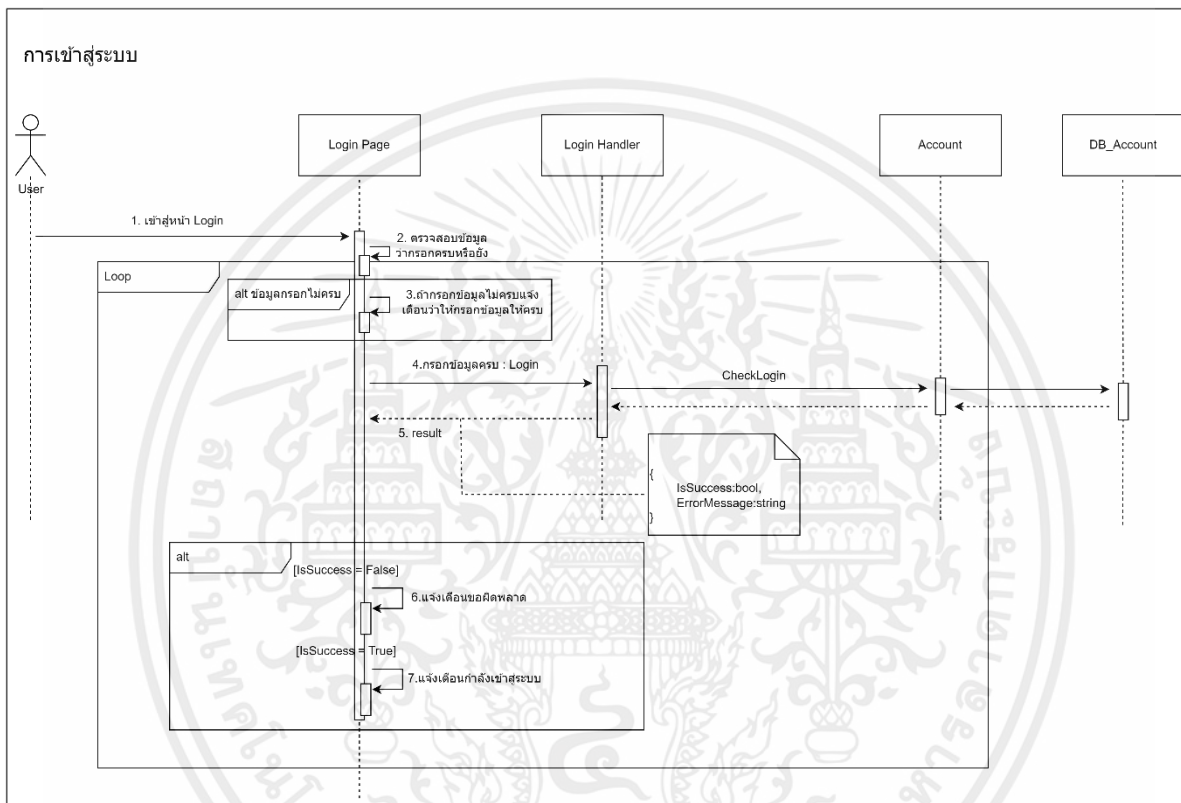
ตาราง 3.10 Use case การจัดการผู้ใช้งาน

Use case: การจัดการผู้ใช้งาน
Actor: ผู้ดูแลระบบ
Pre-condition: ลงชื่อเข้าใช้งานแล้วในฐานะผู้ดูแลระบบ
Flow of Events: 1. เข้าไปที่การดูแลระบบ 2. เข้าไปที่หน้าการจัดการใช้งาน 3. เพิ่ม/ลบ/แก้ไขผู้ใช้งาน 4. กดปุ่มบันทึกข้อมูล
Post-condition: ได้รับรู้คะแนนของสมาชิกแต่ละคนในกลุ่มเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

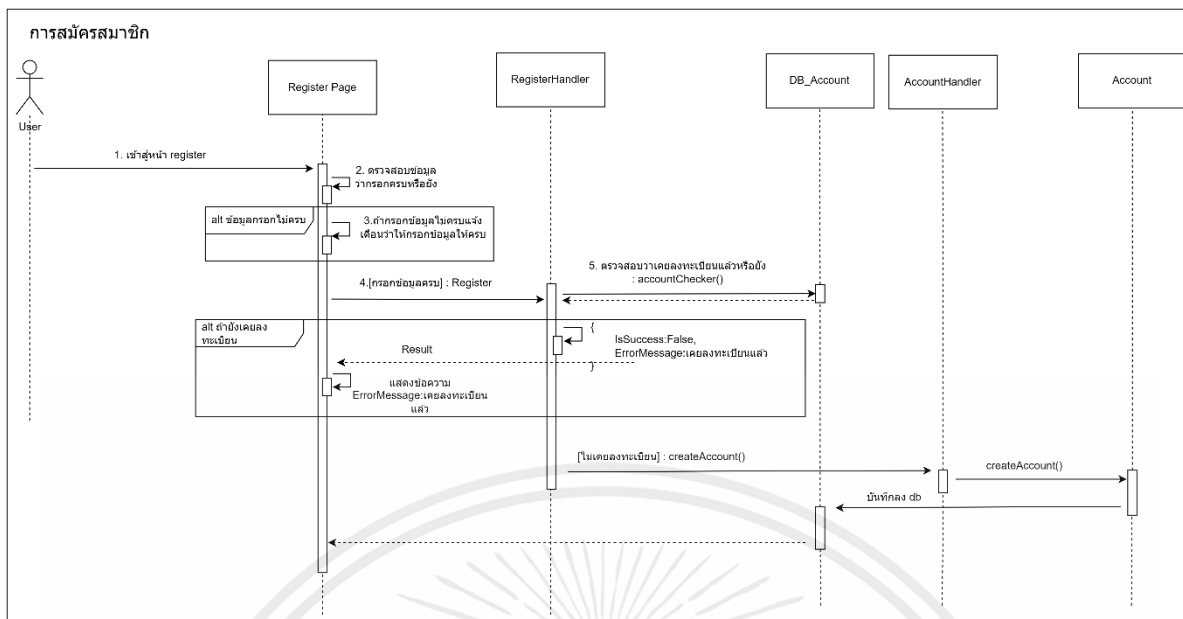
3.4 Sequence diagram

Sequence diagram แสดงการรับส่งข้อมูลของแต่ละส่วนการทำงานแบบ conceptual สำหรับรองรับการทำงานต่างๆของเว็บไซต์ระบบประเมินผลสำหรับการเขียน โปรแกรมไพธอน โดยที่ ออกแบบมีรายละเอียดต่างๆดังนี้

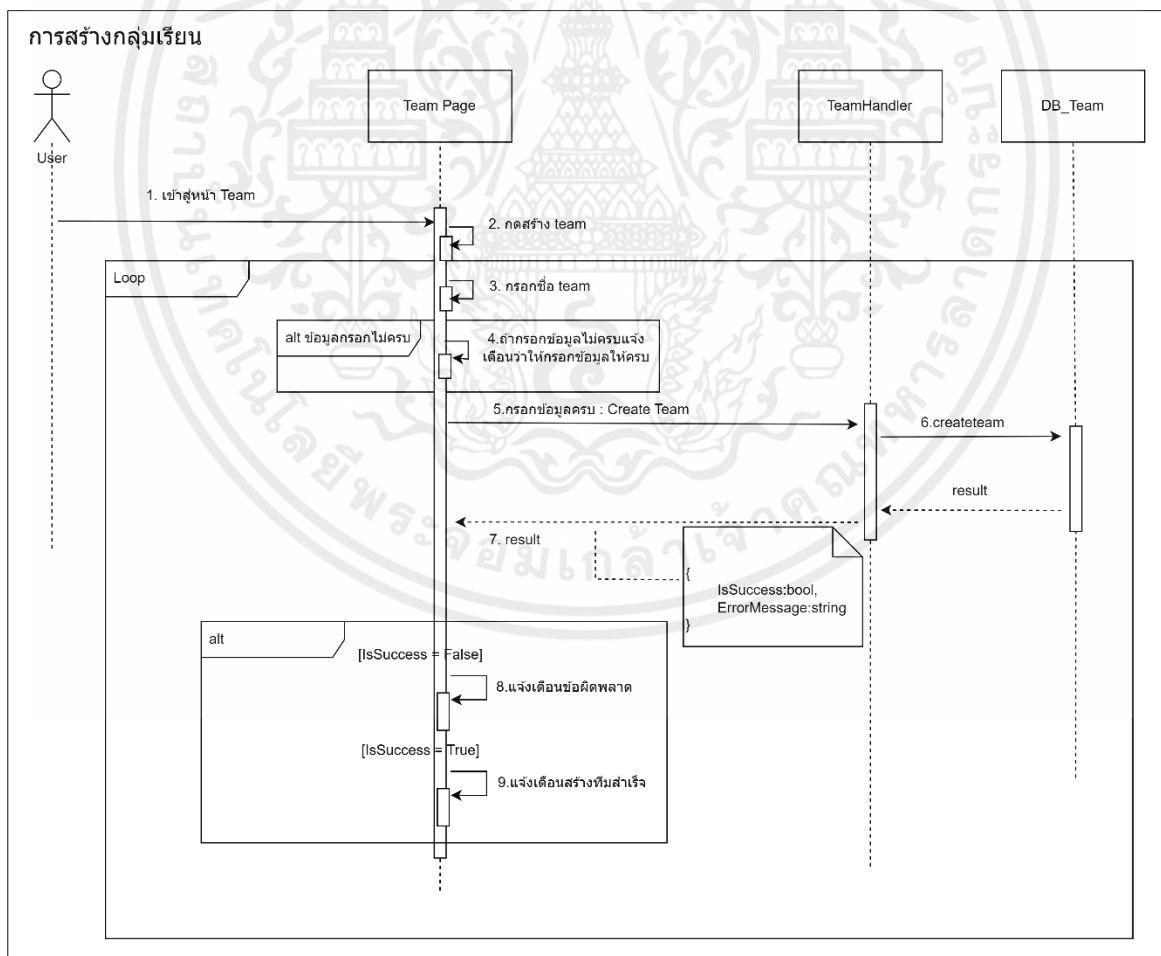


รูป 3.5 Sequence diagram การเข้าสู่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

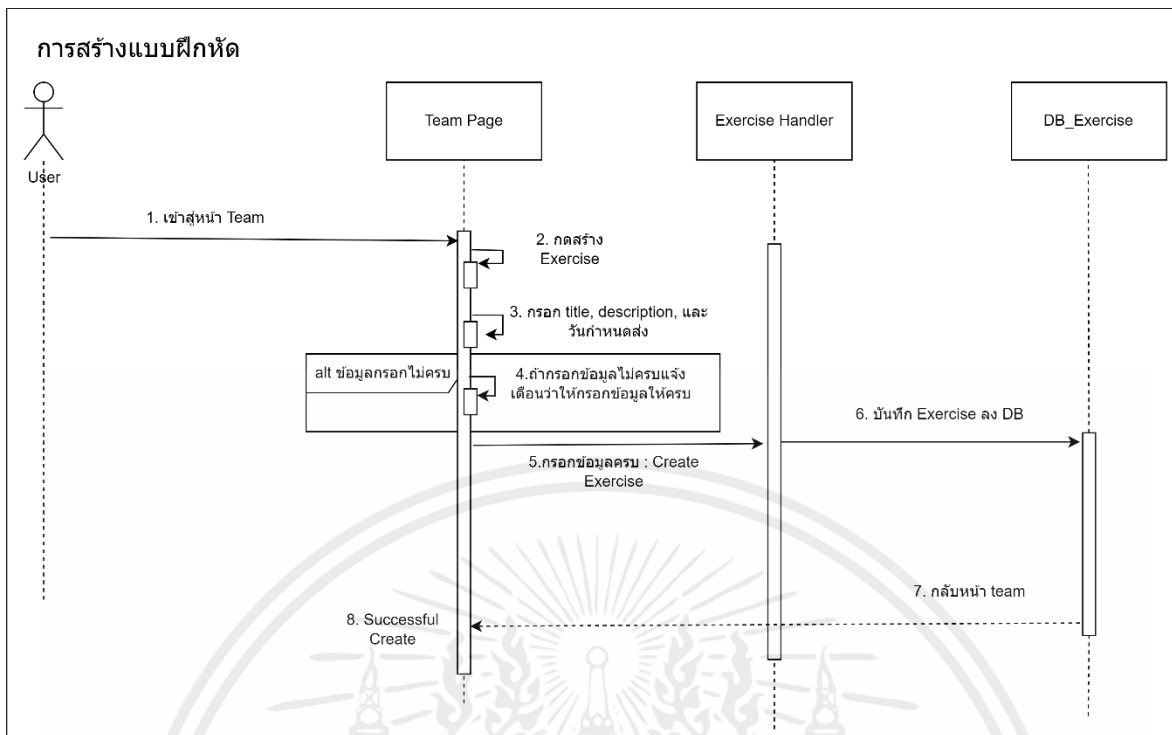


รูป 3.6 Sequence diagram การสมัครสมาชิก

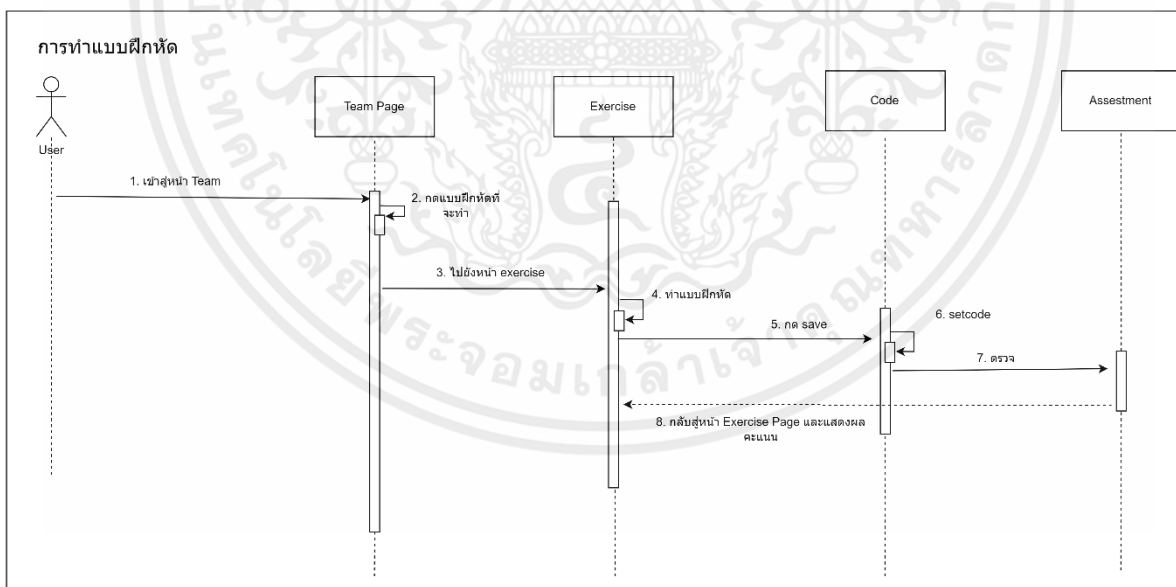


รูป 3.7 Sequence diagram การสร้างกลุ่มเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

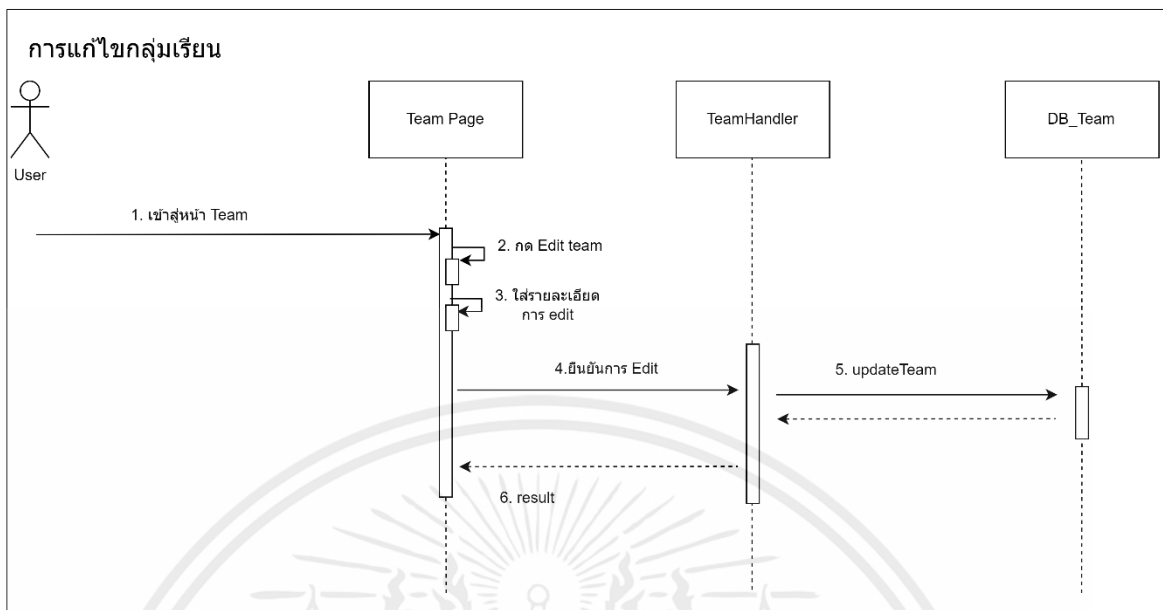


รูป 3.8 Sequence diagram การสร้างแบบฝึกหัด

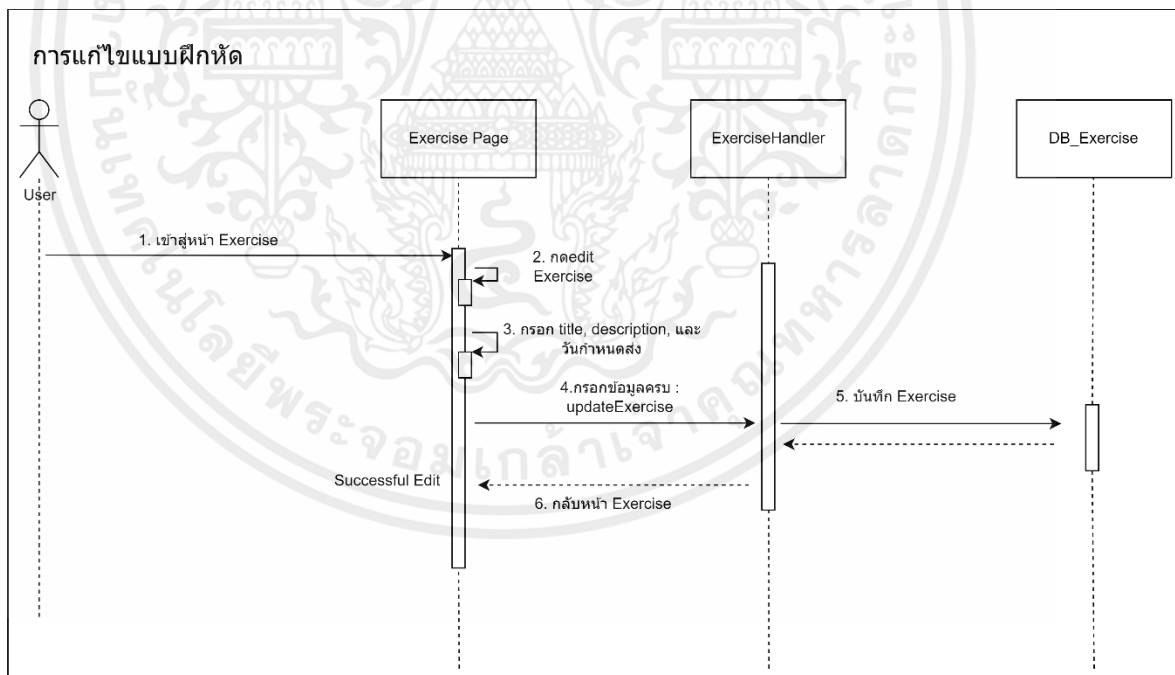


รูป 3.9 Sequence diagram การทำแบบฝึกหัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

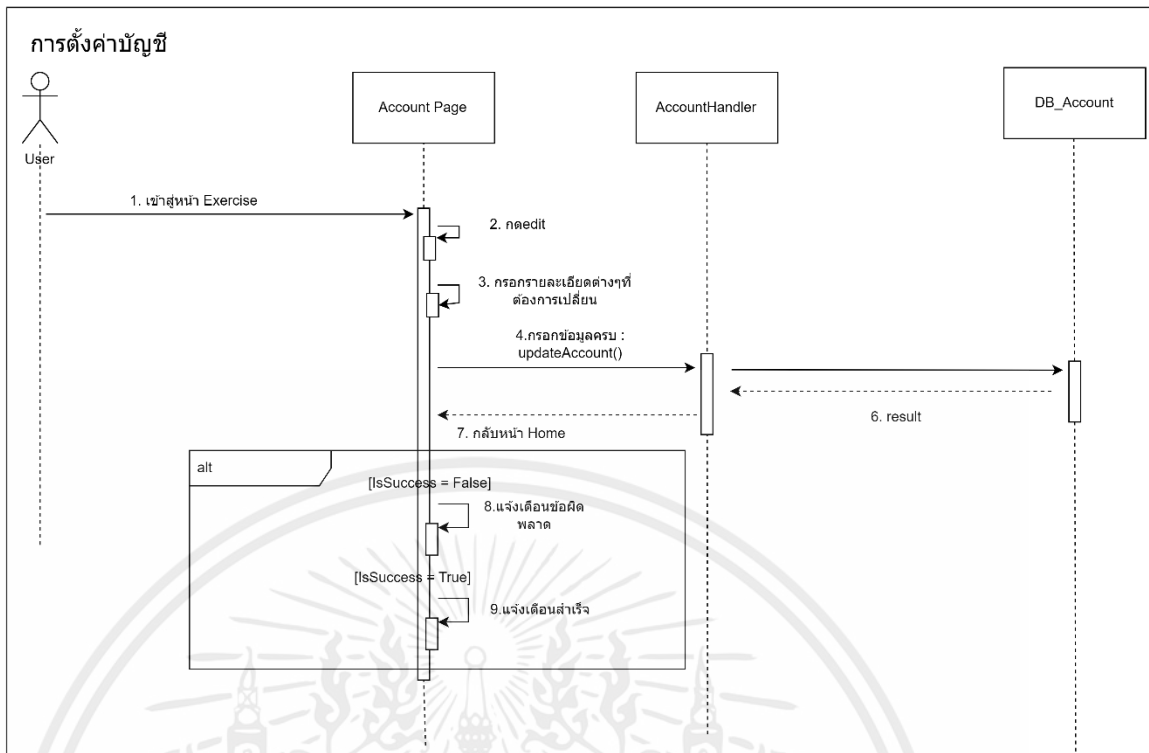


รูป 3.10 Sequence diagram การแก้ไขกลุ่มเรียน

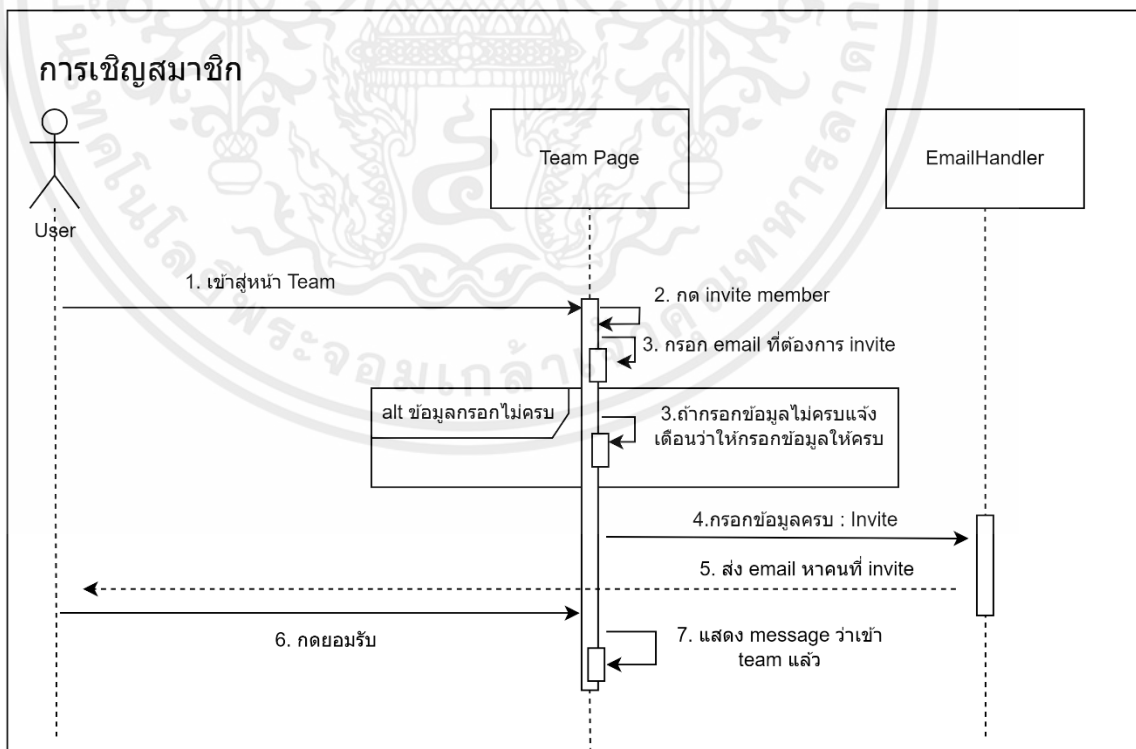


รูป 3.11 Sequence diagram การแก้ไขแบบฝึกหัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

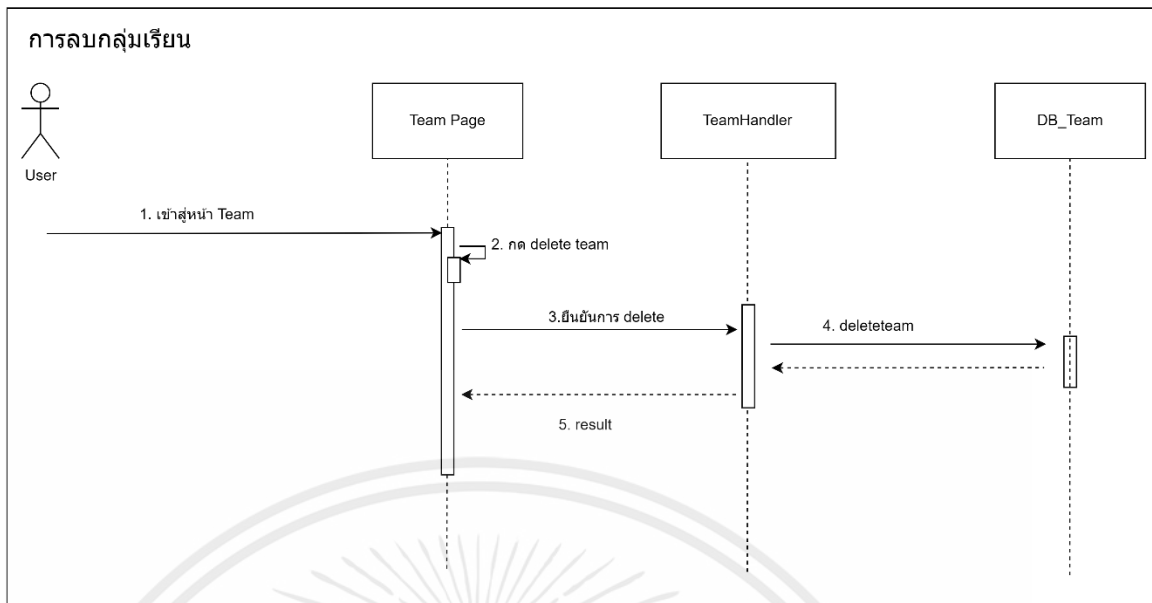


รูป 3.12 Sequence diagram การตั้งค่าบัญชี



รูป 3.13 Sequence diagram การเชิญสมาชิกเข้ากลุ่มเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

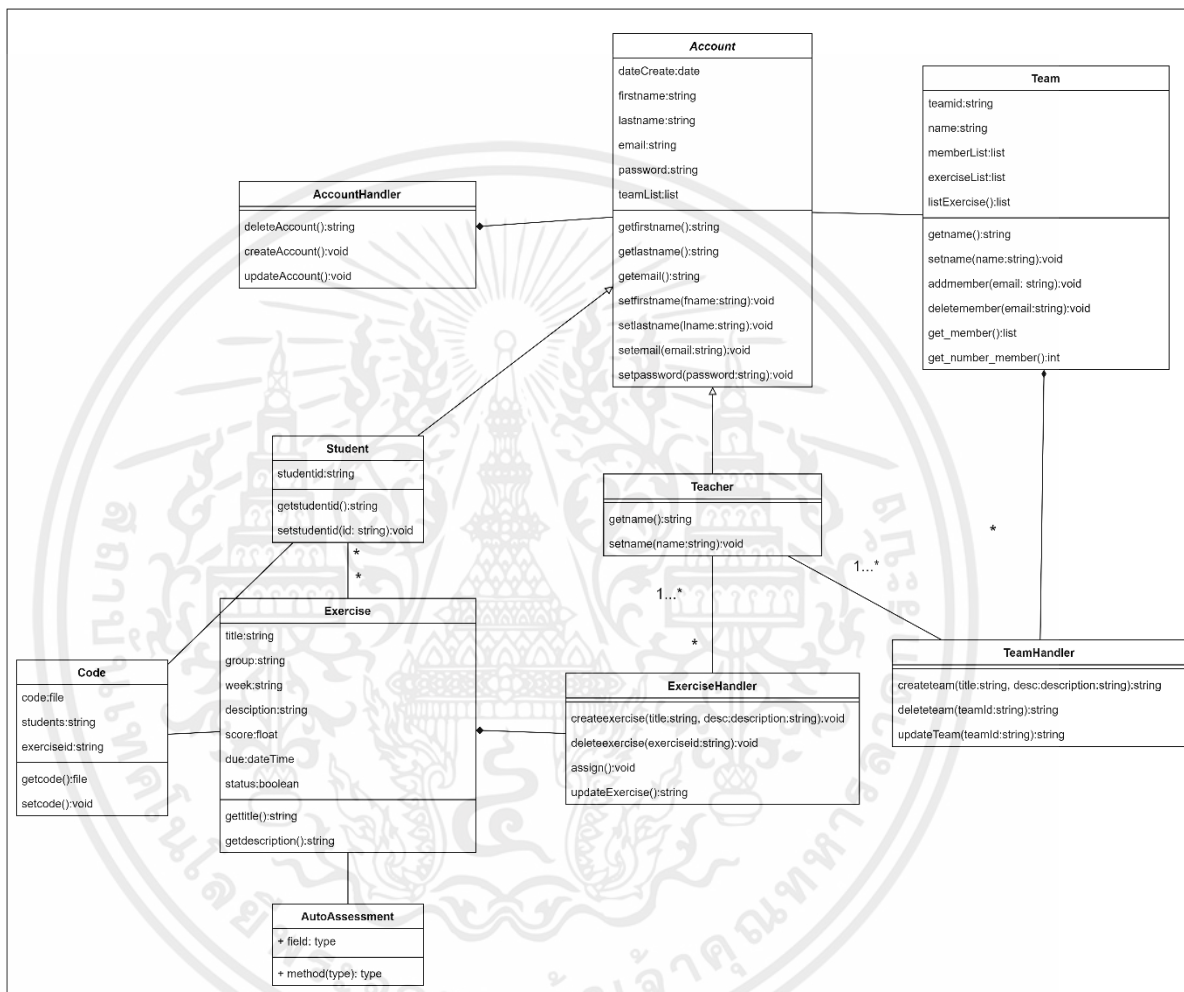


รูป 3.14 Sequence diagram การลบกลุ่มเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 Class diagram

ออกแบบเพื่อใช้ในการทำงานเบื้องหลังของเว็บไซต์ระบบประเมินผลสำหรับการเขียนโปรแกรมไพธอน มีรายละเอียดดังนี้



รูป 3.15 ภาพรวมของ Class diagram

3.5.1 Account

มี Attributes ดังนี้

- 1) dateCreate (date) มีไว้เก็บวันที่สร้างบัญชี
- 2) firstName (string) มีไว้เก็บชื่อ
- 3) lastName (string) มีไว้เก็บนามสกุล
- 4) email (string) มีไว้เก็บอีเมลผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5) password (string) มีไว้เก็บรหัสผ่าน
- 6) teamlist (list) มีไว้เก็บข้อมูลของกลุ่มเรียนที่ผู้ใช้เข้าร่วม

มี Methods ดังนี้

- 1) getfirstname มีไว้ใช้ในการเรียกใช้ชื่อผู้ใช้งาน
- 2) getlastname มีไว้ใช้ในการเรียกใช้นามสกุลผู้ใช้งาน
- 3) getemail มีไว้ใช้ในการเรียกใช้อีเมลผู้ใช้งาน
- 4) setfirstname มีไว้ใช้ในการตั้งชื่อผู้ใช้งาน
- 5) setlastname มีไว้ใช้ในการตั้งชื่อนามสกุลผู้ใช้งาน
- 6) setemail มีไว้ใช้ในการตั้งอีเมลผู้ใช้งาน
- 7) setpassword มีไว้ใช้ในการตั้งรหัสผ่านผู้ใช้งาน

3.5.2 AccountHandler

มี Methods ดังนี้

- 1) createAccount มีไว้รองรับการสร้างบัญชีผู้ใช้
- 2) deleteAccount มีไว้รองรับการลบบัญชีผู้ใช้
- 3) updateAccount มีไว้รองรับการแก้ไขบัญชีผู้ใช้

3.5.3 Student

มี Attributes ดังนี้

- 1) studentid (string) มีไว้เก็บรหัสนักศึกษา

มี Methods ดังนี้

- 1) getstudentid มีไว้รองรับการเรียกใช้รหัสนักศึกษา
- 2) setstudentid มีไว้รองรับการแก้ไขรหัสนักศึกษา

3.5.4 Teacher

มี Methods ดังนี้

- 1) getname มีไว้รองรับการเรียกใช้ชื่อ
- 2) setname มีไว้รองรับการแก้ไขชื่อ

3.5.5 Team

มี Attributes ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1) teamid (string) มีไว้เก็บชื่อเฉพาะ (ID) ของกลุ่มเรียน
- 2) name (string) มีไว้เก็บชื่อของกลุ่มเรียน
- 3) memberList (list) มีไว้เก็บรายชื่อสมาชิกกลุ่มเรียน
- 4) exerciseList (list) มีไว้เก็บแบบฝึกหัดทั้งหมดของกลุ่มเรียน

มี Methods ดังนี้

- 1) getname มีไว้รองรับการเรียกใช้ชื่อกลุ่มเรียน
- 2) setname มีไว้รองรับการแก้ไขชื่อกลุ่มเรียน
- 3) addmember มีไว้รองรับการเพิ่มสมาชิกกลุ่มเรียน
- 4) deletemember มีไว้รองรับการลบสมาชิกกลุ่มเรียน
- 5) get_member มีไว้รองรับการเรียกใช้รายชื่อสมาชิกทั้งหมดของกลุ่มเรียน
- 6) get_number_member มีไว้รองรับการเรียกใช้จำนวนสมาชิกทั้งหมดของกลุ่มเรียน

3.5.6 TeamHandler

มี Methods ดังนี้

- 1) createteam มีไว้รองรับการสร้างกลุ่มเรียน
- 2) deleteteam มีไว้รองรับการลบกลุ่มเรียน
- 3) updateteam มีไว้รองรับการแก้ไขกลุ่มเรียน

3.5.7 Exercise

มี Attributes ดังนี้

- 1) title (string) มีไว้เก็บชื่อแบบฝึกหัด
- 2) exerciseid (string) มีไว้เก็บชื่อเฉพาะ (ID) ของแบบฝึกหัด
- 3) group (string) มีไว้เก็บชื่อกลุ่มเรียน
- 4) week (string) มีไว้เก็บสัปดาห์ที่มอบหมายแบบฝึกหัด
- 5) description (string) มีไว้เก็บรายละเอียดของแบบฝึกหัด
- 6) score (float) มีไว้เก็บคะแนนของแบบฝึกหัด
- 7) due (date) มีไว้เก็บวันที่กำหนดส่งแบบฝึกหัด
- 8) status (boolean) มีไว้เก็บสถานการณ์ส่งแบบฝึกหัดของผู้ใช้งาน

มี Methods ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1) gettitle มีไว้รองรับการเรียกใช้ชื่อแบบฝึกหัด
- 2) getdescription มีไว้รองรับการเรียกใช้รายละเอียดแบบฝึกหัด

3.5.8 ExerciseHandler

มี Methods ดังนี้

- 1) createexercise มีไว้รองรับการสร้างแบบฝึกหัด
- 2) deleteexercise มีไว้รองรับการลบแบบฝึกหัด
- 3) assign มีไว้รองรับการมอบหมายแบบฝึกหัดให้สมาชิกทำ
- 4) updateexercise มีไว้รองรับการแก้ไขแบบฝึกหัด

3.5.9 Code

มี Attributes ดังนี้

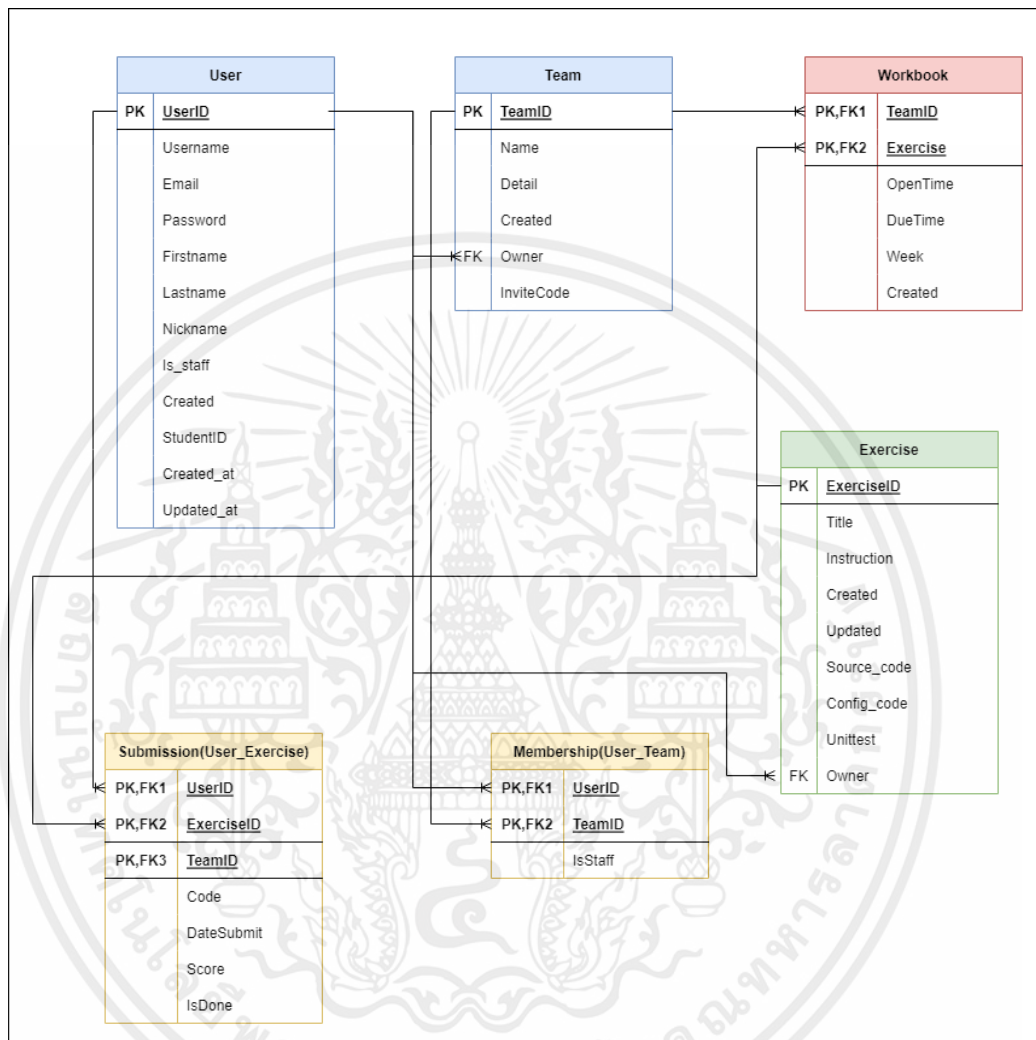
- 1) code (file) มีไว้เก็บโปรแกรมที่ผู้ใช้งานส่งเข้ามาผ่านแบบฝึกหัด
- 2) student (string) มีไว้เก็บชื่อของผู้ใช้งานที่ส่งโปรแกรมเข้ามา
- 3) exerciseid (string) มีไว้เก็บชื่อเฉพาะ (ID) ของแบบฝึกหัดที่ผู้ใช้งานทำแล้วส่งโปรแกรมเข้ามา

มี Methods ดังนี้

- 1) getcode มีไว้รองรับการเรียกดูโปรแกรมที่ผู้ใช้งานส่งเข้ามา
- 2) setcode มีไว้รองรับการแก้ไขโปรแกรมที่ผู้ใช้งานส่งเข้ามา

3.6 โครงสร้างฐานข้อมูล

ฐานข้อมูลที่ถูกจัดทำเลือกใช้คือ ProgreSQL ซึ่งเป็นประเภทตารางเชิงสัมพันธ์ โดยฐานข้อมูลของเว็บไซต์ระบบประเมินผลสำหรับการเขียนโปรแกรมไพธอน



รูป 3.16 ER Diagram

3.6.1 Table User

- 1) UserID มีไว้เก็บชื่อเฉพาะของผู้ใช้งาน
- 2) Email มีไว้เก็บอีเมลของผู้ใช้งาน
- 3) Password มีไว้เก็บรหัสผ่านของผู้ใช้งาน
- 4) Firstname มีไว้เก็บชื่อของผู้ใช้งาน
- 5) Lastname มีไว้เก็บนามสกุลของผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 6) Detail เก็บรายละเอียดของผู้ใช้งาน
- 7) Created มีไว้เก็บวันที่สร้างบัญชีผู้ใช้งาน
- 8) StudentID มีไว้เก็บรหัสนักศึกษา (Optional)

3.6.2 Table Team

- 1) TeamID มีไว้เก็บชื่อเฉพาะของกลุ่มเรียน
- 2) Name มีไว้เก็บชื่อกลุ่มเรียน
- 3) Detail มีไว้เก็บรายละเอียดของกลุ่มเรียน
- 4) Created มีไว้เก็บวันที่สร้างกลุ่มเรียน
- 5) Owner มีไว้เก็บรายละเอียดผู้ใช้งานที่เป็นเจ้าของกลุ่มเรียน

3.6.3 Table Exercise

- 1) ExerciseID มีไว้เก็บชื่อเฉพาะของแบบฝึกหัด
- 2) Title มีไว้เก็บชื่อของแบบฝึกหัด
- 3) Instruction มีไว้เก็บคำอธิบายของแบบฝึกหัด
- 4) Created มีไว้เก็บวันที่สร้างแบบฝึกหัด
- 5) Sourcecode มีไว้เก็บตัวอย่างการเขียน โปรแกรมของแบบฝึกหัด

3.6.4 Table Team_Exercise

- 1) TeamID มีไว้เชื่อมความสัมพันธ์ไปที่ Table Team
- 2) Exercise มีไว้เชื่อมความสัมพันธ์ไปที่ Table Exercise
- 3) OpenTime มีไว้เก็บวันที่เริ่มให้ทำแบบฝึกหัด
- 4) DueTime มีไว้เก็บวันกำหนดส่งแบบฝึกหัด

3.6.5 Table Achievement (User_Exercise)

- 1) UserID มีไว้เชื่อมความสัมพันธ์ไปที่ Table User
- 2) ExerciseID มีไว้เชื่อมความสัมพันธ์ไปที่ Table Exercise
- 3) Code มีไว้เก็บ โปรแกรมที่สมาชิกในกลุ่มเรียนส่งในแบบฝึกหัด
- 4) DateSubmit มีไว้เก็บวันที่ส่งแบบฝึกหัดไป

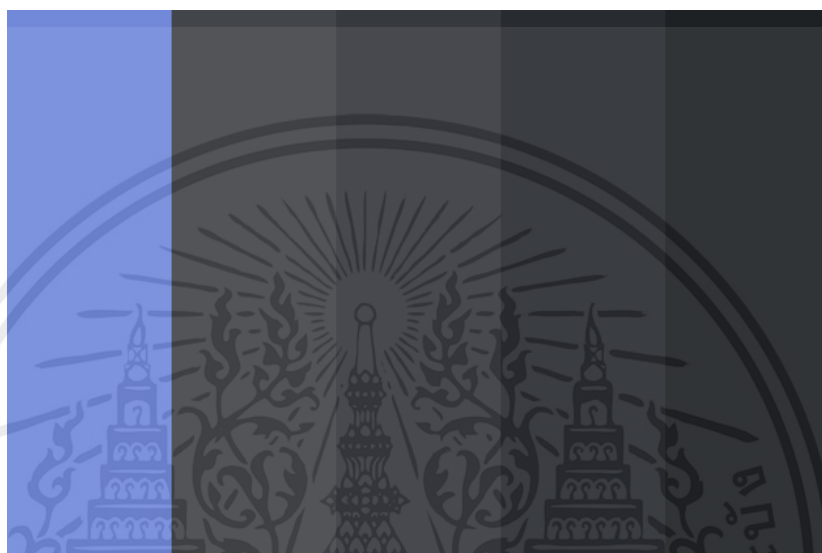
3.6.6 Table Member (User_Team)

- 1) UserID มีไว้เชื่อมความสัมพันธ์ไปที่ Table User
- 2) TeamID มีไว้เชื่อมความสัมพันธ์ไปที่ Table Team
- 3) IsStaff มีไว้เก็บสถานะของสมาชิกว่าเป็นผู้ช่วยของผู้ดูแลกลุ่มเรียนหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 User Interface

การออกแบบ User interface ของระบบจะเน้นการใช้สีที่ดูไม่เยอะ เหมาะสำหรับผู้ใช้งานทั่วไป ให้ใช้งานได้สะดวก ออกแบบผ่านโปรแกรม Figma ซึ่งเมื่อเป็นการใช้สีที่มีสีดูไม่เยอะมาก จึงเลือกใช้โทนขาว, ดำ, เทา แล้วเติมแต่งจุดที่ต้องการให้เด่นชัดด้วยสีน้ำเงินและสีแดง



รูป 3.17 Color palette



รูป 3.18 Color palette

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

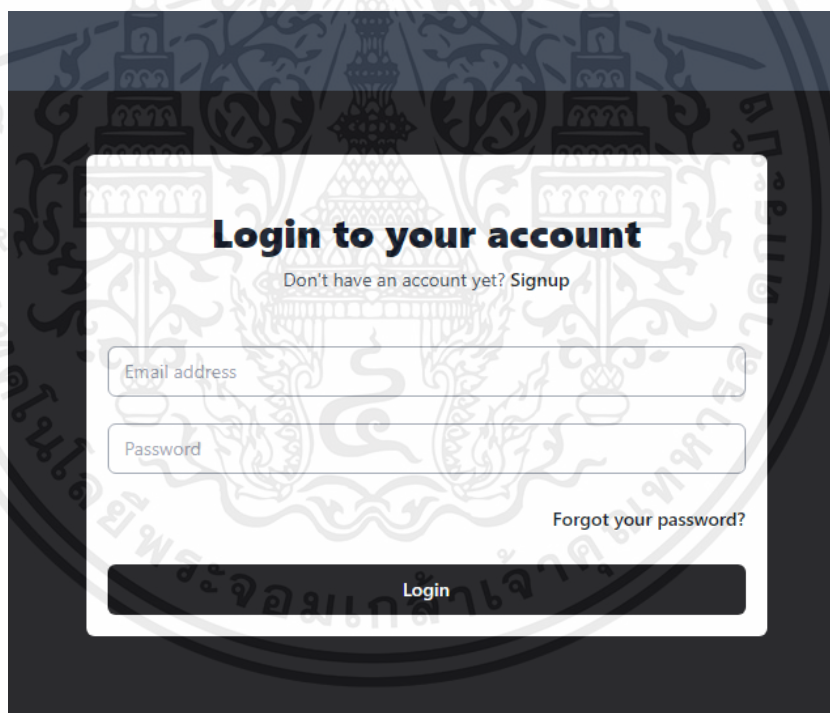
บทที่ 4

ผลการทดลอง

ในบทนี้จะอธิบายถึงงานที่ได้ทำมาตลอดภาคการศึกษา เพื่อแสดงให้เห็นถึงความก้าวหน้าในโครงการ โดยจะอธิบายขั้นตอนการพัฒนาตั้งแต่การเข้าสู่ระบบเว็บไซต์

4.1 การเข้าสู่ระบบ

เป็นการเข้าสู่ระบบทั่วไปผ่านอีเมลและรหัสผ่าน หากไม่ได้เป็นสมาชิกต้องทำการสมัครสมาชิก โดยกรอกอีเมล, ชื่อ, นามสกุล, รหัสผ่าน, และยืนยันรหัสผ่าน

The image shows a login form with a white background and a dark blue header. The header contains the text "Login to your account" in bold, followed by "Don't have an account yet? Signup" in a smaller font. Below the header are two input fields: "Email address" and "Password". To the right of the "Password" field is a link that says "Forgot your password?". At the bottom of the form is a dark blue button with the text "Login" in white. The form is overlaid on a dark blue background with a faint watermark of a university seal.

รูป 4.1 หน้าการเข้าสู่ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signup to create an account
 Already have an account? [Login](#)

Email address

First name

Last name

Password

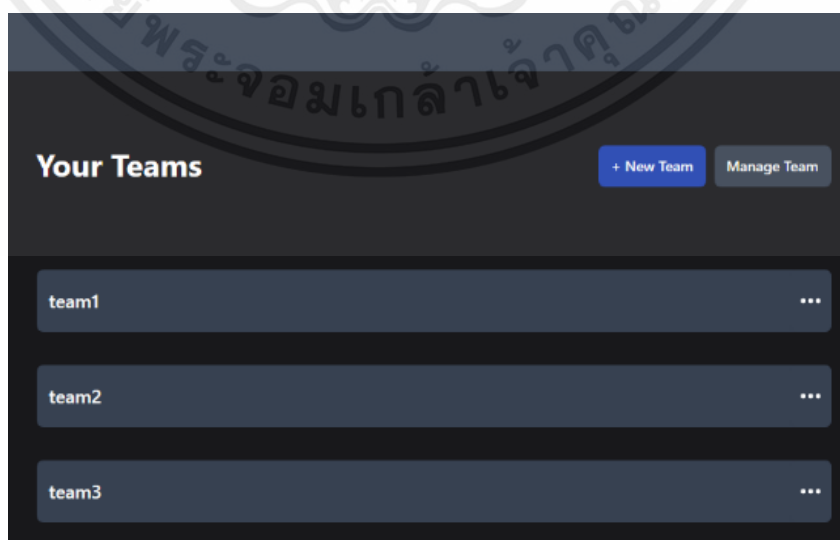
Confirm password

Signup

รูป 4.2 หน้าการสมัครสมาชิก

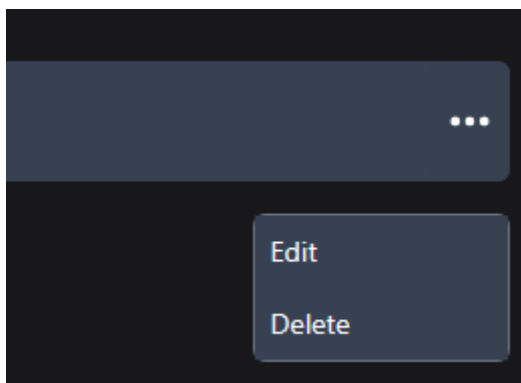
4.2 ฟังก์ชันการจัดการกลุ่มเรียน

มีหน้าที่ในการจัดการกลุ่มเรียนของทั้งที่เป็นเจ้าของและไม่เป็นเจ้าของ โดยเมื่อกดที่ปุ่มเมนูขวามือของแต่ละกลุ่มเรียนสามารถกด delete เพื่อลบออก หรือสามารถแก้ไขกลุ่มเรียนได้เช่นเดียวกัน



รูป 4.3 หน้ากลุ่มเรียนทั้งหมด

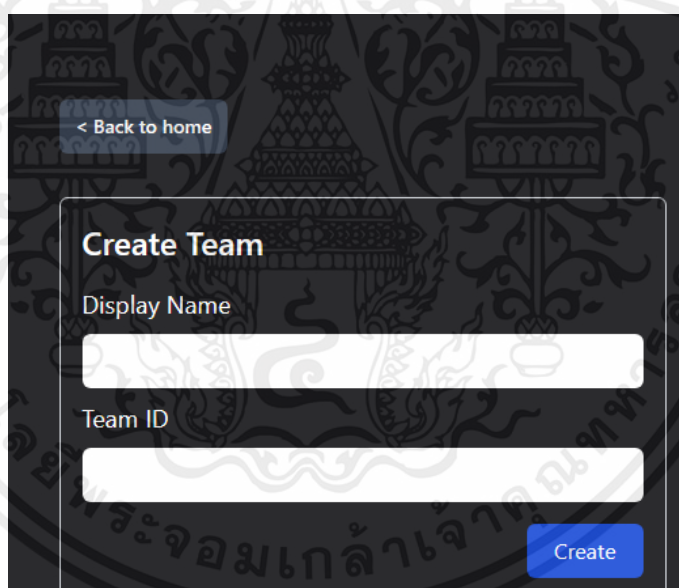
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.4 เมนูการ edit และ delete

4.2.1 ฟังก์ชันการสร้างกลุ่มเรียนใหม่และการเข้าร่วมกลุ่มเรียน

เมื่อกดที่ปุ่ม New Team สีฟ้าด้านขวาบนจะแสดงหน้าที่สามารถใช้เพื่อสร้างกลุ่มเรียนขึ้นมาใหม่ได้

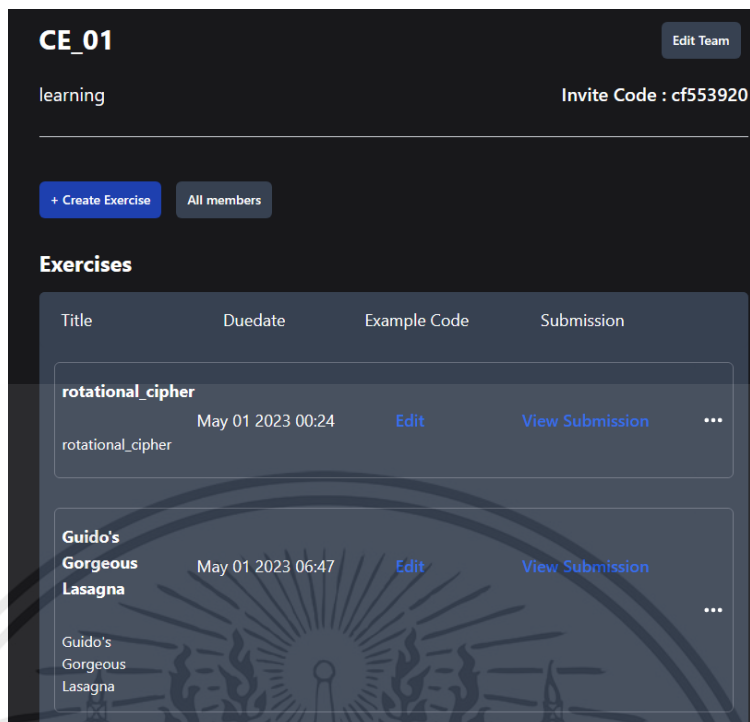


รูป 4.5 เมนูการสร้างกลุ่มเรียน

4.3 ฟังก์ชันในกลุ่มเรียน

เป็นหน้าหลักของกลุ่มเรียน โดยจะแสดงแบบฝึกหัดทั้งหมดที่มีอยู่ แสดงรายละเอียดแบบฝึกหัดและวันกำหนดส่ง โดยมีรายละเอียดย่อย ๆ ดังต่อไปนี้

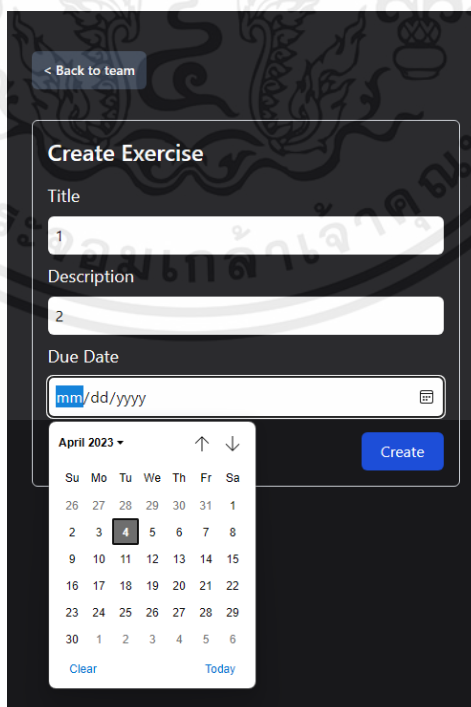
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.6 หน้ากลุ่มเรียน

4.3.1 ฟังก์ชันการสร้างแบบฝึกหัด

เมื่อกดที่ปุ่ม Create Exercise จะเข้าสู่หน้าสร้างแบบฝึกหัดใหม่ โดยจำเป็นต้องกรอกชื่อแบบฝึกหัด รายละเอียดแบบฝึกหัด และวันกำหนดส่ง ส่วนการใส่ตัวอย่างโค้ดกับตัว Unit Test จะใส่ในภายหลัง



รูป 4.7 หน้าสร้างแบบฝึกหัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 ฟังก์ชันการแก้ไขกลุ่มเรียน

เมื่อกดที่ปุ่ม Edit Team จะสามารถแก้ไขกลุ่มเรียนได้ ทั้งชื่อและชื่อเฉพาะของกลุ่มเรียน

รูป 4.8 การแก้ไขกลุ่มเรียน

4.3.3 ฟังก์ชันการได้ unit test, config code, example code, guideline code

โดยเมื่อกดปุ่ม edit ของแต่ละแบบฝึกหัดจะสามารถเข้าสู่หน้าตั้งค่า unit test, config code, example code(ใช้ให้ระบบคำนวณ) และ guideline code(โครงร่างโปรแกรม)

```

Unit Test
1 import unittest
2 from graderutils.graderunittest import points
3
4 from rotational_cipher import (
5     rotate,
6 )
7 # Tests adapted from 'problem-specifications/canonical-data.json'
8 class RotationalCipherTest(unittest.TestCase):
9
10     @points(5, msg_on_fail='test 01 fail')

```

รูป 4.9 การแก้ไข Unit test

```

Config Code
1 test_groups:
2   - module: grader_tests
3     display_name: test blacklist
4 validation:
5   display_name: Test config.yaml
6 tasks:
7   - type: python_whitelist
8     display_name: "Want Loops"
9     description: "Need Loops"
10    file: my_solution.py
11    node_names:

```

รูป 4.10 การแก้ไข Config code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The image shows a code editor interface with two sections. The top section, titled 'Example Code', contains a single line of code: `1 string`. The bottom section, titled 'Guideline Code', contains two lines of code: `1 def rotate(text, key):` and `2 pass`.

รูป 4.11 การแก้ไข Example และ Guideline Code

4.3.4 ฟังก์ชันการดูคะแนนของสมาชิกในกลุ่มเรียน

เมื่อกด View Submission จะสามารถดูคะแนนของสมาชิกในกลุ่มรวมถึงเวลาที่ได้ทำการส่งมาด้วย

The screenshot shows a web interface for viewing submissions. At the top, there is a button '< Back to team' and a title 'Prime number' with an 'Edit Example Code' button. Below the title, it says 'write function to test prime number' and 'Due Date: Apr 28 2023 21:00'. A 'Submission' section features a search bar 'Search for member...' and a table with columns: Students, Scores, Submitted Date, and Submission.

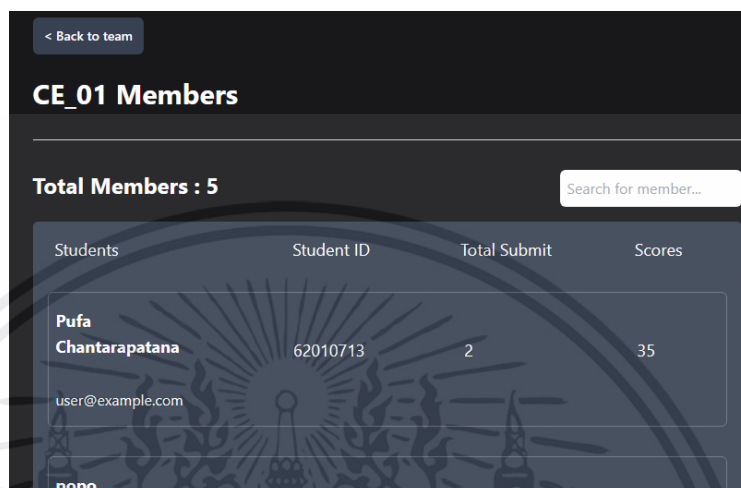
Students	Scores	Submitted Date	Submission
light 12345789	5	Apr 24 2023 20:24	View Code
Sahatas 62010922	15	Apr 26 2023 13:27	View Code

รูป 4.12 หน้า View Submission

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.5 ฟังก์ชันการจัดการสมาชิกในกลุ่มเรียน

เมื่อกด All member จะเข้าสู่หน้าสมาชิกทั้งหมดซึ่งจะแสดงชื่อ, รหัสนักศึกษา, จำนวนการส่งงาน และคะแนนที่ได้ สามารถค้นหาสมาชิกในกลุ่มด้วยรหัสนักศึกษาเพื่อดูคะแนนแยกแต่ละคนได้

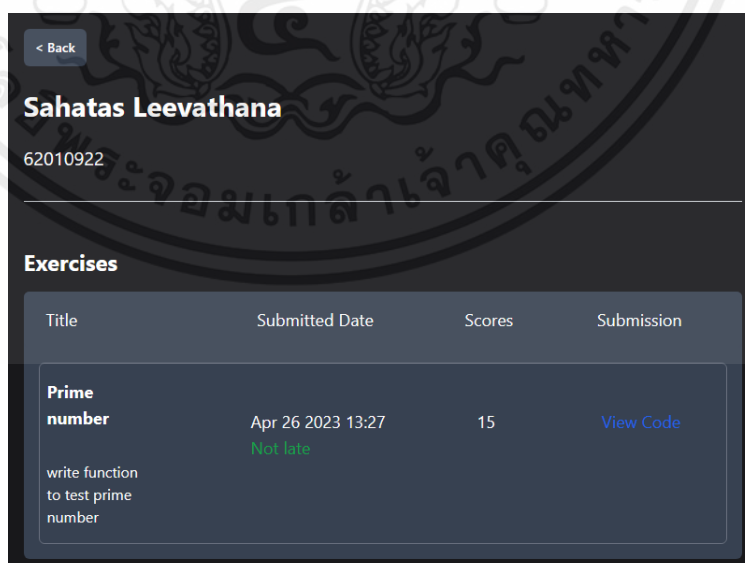


Students	Student ID	Total Submit	Scores
Pufa Chantarapatana user@example.com	62010713	2	35
popo			

รูป 4.13 หน้าสมาชิกทั้งหมด

4.3.6 ฟังก์ชันการเช็คคะแนนสมาชิกรายบุคคล

สามารถเช็คคะแนนรายบุคคลได้ทุกแบบฝึกหัดโดยจะแสดงวันที่ส่งเข้ามา แสดงคะแนน และกดดูโค้ดที่ส่งมาได้



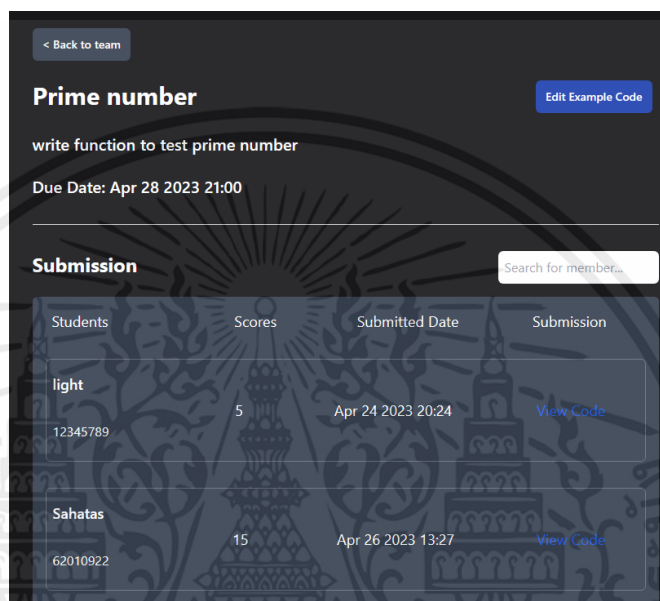
Title	Submitted Date	Scores	Submission
Prime number write function to test prime number	Apr 26 2023 13:27 Not late	15	View Code

รูป 4.14 หน้าสมาชิกรายบุคคล

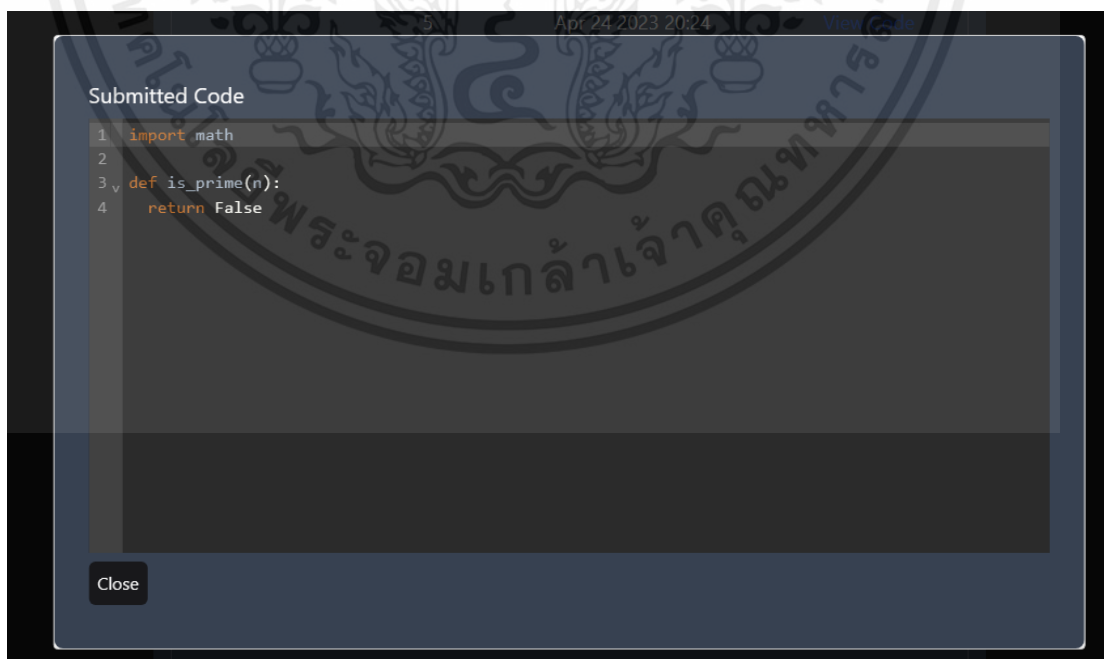
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ฟังก์ชันแบบฝึกหัด

ในหน้าแบบฝึกหัดจะแสดงรายละเอียดต่างๆของแบบฝึกหัด สามารถดูโค้ดของสมาชิกแต่ละคนที่ส่งเข้ามาได้ โดยกดที่ส่วน View Code เมื่อกด Edit Example Code จะสามารถแก้ไขตัวอย่างโค้ดรวมถึง Unit test และเมื่อกด Edit description จะแก้ไขชื่อ, รายละเอียด, และวันกำหนดส่งของแบบฝึกหัดได้



รูป 4.15 หน้าแบบฝึกหัด

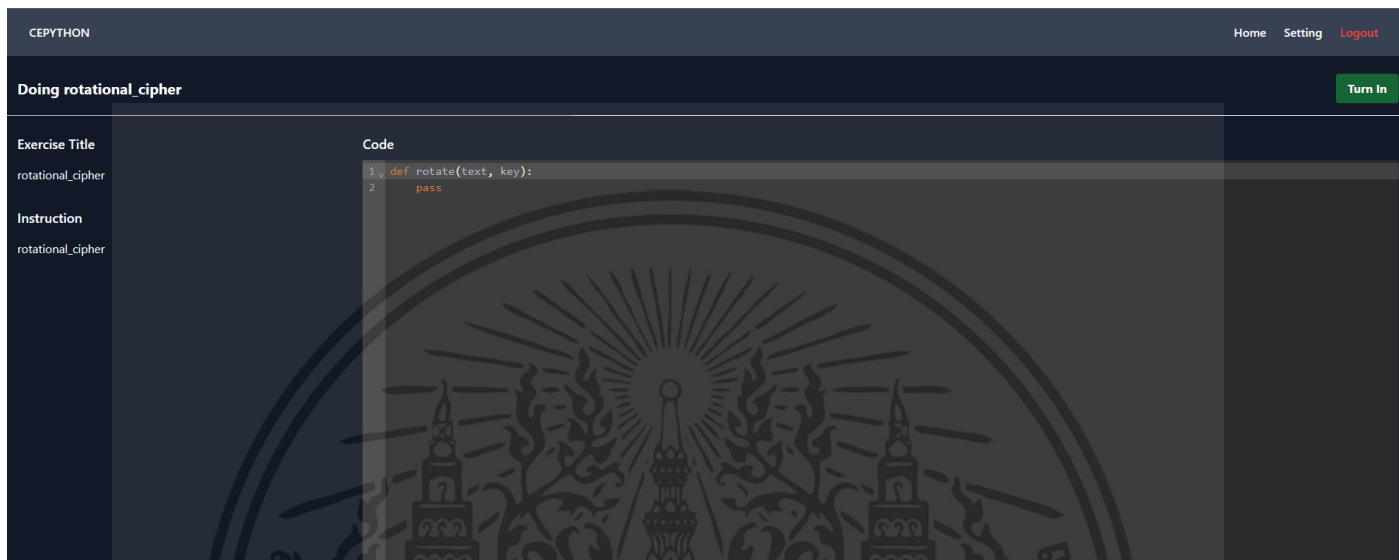


รูป 4.16 หน้า View Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ฟังก์ชันการทำแบบฝึกหัด

ในหน้านี้จะเป็นการทำแบบฝึกหัดของสมาชิก โดยตัว Editor เมื่อส่งงานแล้วสามารถตรวจสอบตัวโค้ดและให้คะแนนได้ผ่านตัวระบบหลังบ้าน (Auto Assessment) โดยจำเป็นต้องใช้ตัวอย่างโค้ดของอาจารย์และโค้ดของนักศึกษาที่ส่งมาเป็นอย่างต่ำ



รูป 4.17 หน้าเขียนโค้ด

4.6 ฟังก์ชันการตั้งค่าข้อมูลส่วนตัว

โดยตัวผู้ใช้งานสามารถแก้ไขอีเมล ชื่อ รหัสนักศึกษา รหัสผ่าน ได้ผ่านหน้า Setting นี้

 The image shows a form titled 'Edit Account'. It contains several input fields:

- Email: user@example.com
- First Name: Pufa
- Last Name: Chantarapatana
- Student ID: 62010713

 At the bottom of the form is a 'Save Changes' button.

รูป 4.18 หน้าแก้ไขรายละเอียดบัญชี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 การพัฒนาระบบการจัดการหลังบ้าน (Backend)

4.7.1 การเชื่อมต่อกับ Database โดยใช้ ORM

ในปัจจุบันได้พัฒนา model(ORM) ที่ใช้ในการเชื่อมต่อกับระบบฐานข้อมูลครบทุก table และได้สร้าง serializers ตาม model ที่สร้างไว้ สร้าง API ที่ใช้ติดต่อกับระบบฐานข้อมูล

```

1 from django.db import models
2 from django.contrib.auth.models import User
3
4
5 # Create your models here.
6 class Team(models.Model):
7     name = models.CharField(max_length=255, unique=True) # require
8     detail = models.TextField(default='')
9     created = models.DateTimeField(auto_now_add=True)
10    users = models.ManyToManyField(User, through='Membership')
11
12    class Meta:
13        ordering = ['created']
14
15    def __str__(self) -> str:
16        return self.name
17
18    class Exercise(models.Model):
19        title = models.CharField(max_length=255) # require
20        instruction = models.TextField(default='')
21        created = models.DateTimeField(auto_now_add=True)
22        source_code = models.TextField(default='')
23
24        class Meta:
25            ordering = ['created']
26
27        def __str__(self) -> str:
28            return self.title
29
30    # many to many model
31    class Membership(models.Model):
32        user = models.ForeignKey(User, on_delete=models.CASCADE, related_name='members', related_query_name='member')
33        team = models.ForeignKey(Team, on_delete=models.CASCADE, related_name='members', related_query_name='member')
34        isStaff = models.BooleanField(default=False)
35
36        def __str__(self) -> str:
37            return "{}, {}".format(self.user, self.team)
38
39    # userExercise
40    class Submission(models.Model):
41        user = models.ForeignKey(User, on_delete=models.CASCADE, related_name='submissions', related_query_name='submission')
42        exercise = models.ForeignKey(Exercise, on_delete=models.CASCADE, related_name='submissions', related_query_name='submission')
43        dateSubmit = models.DateTimeField(auto_now=True)
44        code = models.TextField(blank=True)
45
46        def __str__(self) -> str:
47            return "{}, {}".format(self.user, self.exercise)
48
49    # Workbook = Exercise + Team
50    class Workbook(models.Model):
51        openTime = models.DateTimeField()
52        dueTime = models.DateTimeField(blank=True)
53        isOpen = models.BooleanField(default=True)
54        exercise = models.ForeignKey(Exercise, on_delete=models.CASCADE, related_name='workbooks', related_query_name='workbook')
55        team = models.ForeignKey(Team, on_delete=models.CASCADE, related_name='workbooks', related_query_name='workbook')
56
57

```

รูป 4.19 Model ORM ที่ใช้เชื่อมต่อกับระบบฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1 # from rest_framework import serializers
2 from rest_flex_fields import FlexFieldsModelSerializer
3 from .models import Team, Exercise, Membership, Submission, Workbook
4 from django.contrib.auth.models import User
5
6 class UserSerializer(FlexFieldsModelSerializer):
7     class Meta:
8         model = User
9         fields = ['username', 'email', ]
10        expandable_fields = {
11            'members': ('api.MemberSerializer', {'many': True}),
12            'submissions': ('api.SubmissionSerializer', {'many': True}),
13        }
14
15 class TeamSerializer(FlexFieldsModelSerializer):
16     class Meta:
17         model = Team
18         fields = ['pk', 'name', 'detail', 'created']
19         expandable_fields = {
20            'users': ('api.UserSerializer', {'many': True}),
21            'members': ('api.MemberSerializer', {'many': True}),
22            'workbooks': ('api.WorkbookSerializer', {'many': True}),
23        }
24
25
26 class ExerciseSerializer(FlexFieldsModelSerializer):
27     class Meta:
28         model = Exercise
29         fields = ['id', 'title', 'instruction', 'created', 'source_code']
30         expandable_fields = {
31            'submissions': ('api.SubmissionSerializer', {'many': True}),
32            'workbooks': ('api.WorkbookSerializer', {'many': True}),
33        }
34
35 class MemberSerializer(FlexFieldsModelSerializer):
36     class Meta:
37         model = Membership
38         # fields = ['user', 'team', 'isStaff']
39         fields = ['isStaff']
40         expandable_fields = {
41            'user': ('api.UserSerializer'),
42            'team': ('api.TeamSerializer')
43        }
44
45 class SubmissionSerializer(FlexFieldsModelSerializer):
46     class Meta:
47         model = Submission
48         # fields = ['user', 'team', 'isStaff']
49         fields = ['code', 'dateSubmit']
50         expandable_fields = {
51            'user': ('api.UserSerializer'),
52            'exercise': ('api.ExerciseSerializer'),
53        }
54
55 class WorkbookSerializer(FlexFieldsModelSerializer):
56     class Meta:
57         model = Workbook
58         fields = ['openTime', 'dueTime', 'isOpen']
59         expandable_fields = {
60            'team': ('api.TeamSerializer'),
61            'exercise': ('api.ExerciseSerializer'),
62        }

```

รูป 4.20 Serializer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7.2 การแปลงข้อมูล JSON ให้เป็น Datatype Python

Serializer จะทำการแปลงข้อมูลจาก model ให้เป็น Datatype python ที่สามารถนำมาใช้งานต่อได้ และจะเปลี่ยนข้อมูลจาก JSON เป็น Datatype python ด้วย

```

1 from .models import Team, Exercise, Membership
2 from .serializers import TeamSerializer, ExerciseSerializer
3 from rest_framework import permissions
4 from rest_framework.views import FlexFieldsMixin
5 from rest_framework.viewsets import ReadOnlyModelViewSet, ModelViewSet
6 from rest_framework import is_expanded
7
8 class TeamViewSet(FlexFieldsMixin, ReadOnlyModelViewSet):
9     permission_classes = [permissions.IsAuthenticated]
10    serializer_class = TeamSerializer
11    # queryset = Team.objects.all()
12    permit_list_expands = ['users', 'members', 'workbooks', 'workbooks.exercise']
13
14    def get_queryset(self):
15        queryset = Team.objects.all()
16
17        if is_expanded(self.request, 'users'):
18            queryset = queryset.prefetch_related('users')
19
20        if is_expanded(self.request, 'members'):
21            queryset = queryset.prefetch_related('members')
22
23        if is_expanded(self.request, 'workbooks'):
24            queryset = queryset.prefetch_related('workbooks')
25
26        if is_expanded(self.request, 'workbooks__exercise'):
27            queryset = queryset.prefetch_related('workbooks__exercise')
28
29        return queryset
30
31 class ExerciseViewSet(FlexFieldsMixin, ReadOnlyModelViewSet):
32    permission_classes = [permissions.IsAuthenticated]
33    serializer_class = ExerciseSerializer
34    permit_list_expands = ['submissions', 'submissions.user', 'workbooks']
35
36    def get_queryset(self):
37        queryset = Exercise.objects.all()
38
39        if is_expanded(self.request, 'submissions'):
40            queryset = queryset.prefetch_related('submissions')
41
42        if is_expanded(self.request, 'submissions__user'):
43            queryset = queryset.prefetch_related('submissions__user')
44
45        if is_expanded(self.request, 'workbooks'):
46            queryset = queryset.prefetch_related('workbooks')
47
48        return queryset

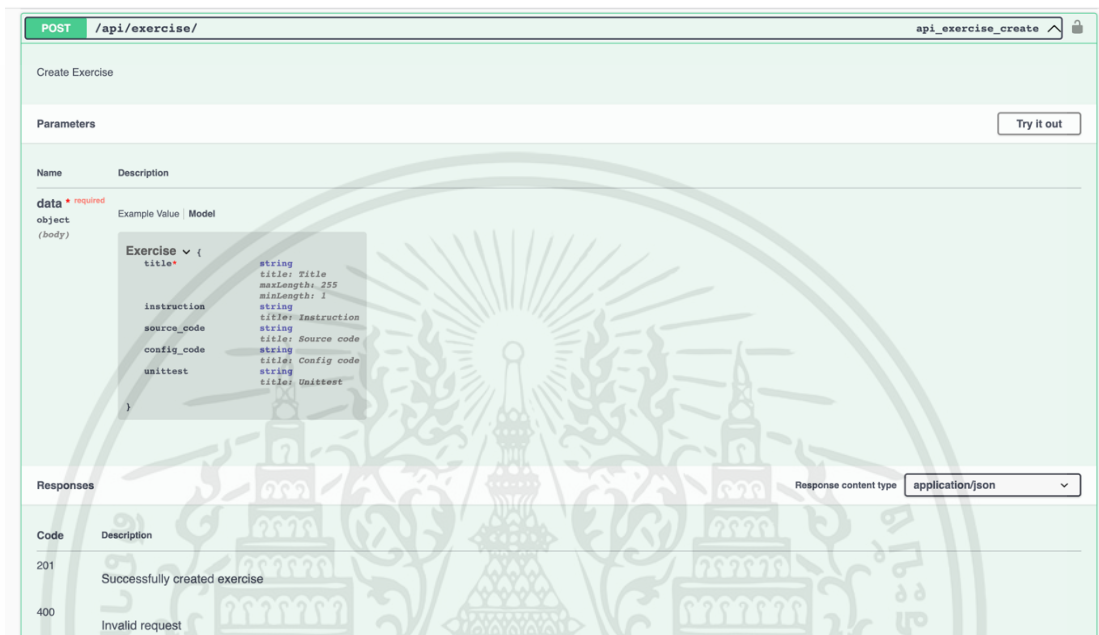
```

รูป 4.21 View

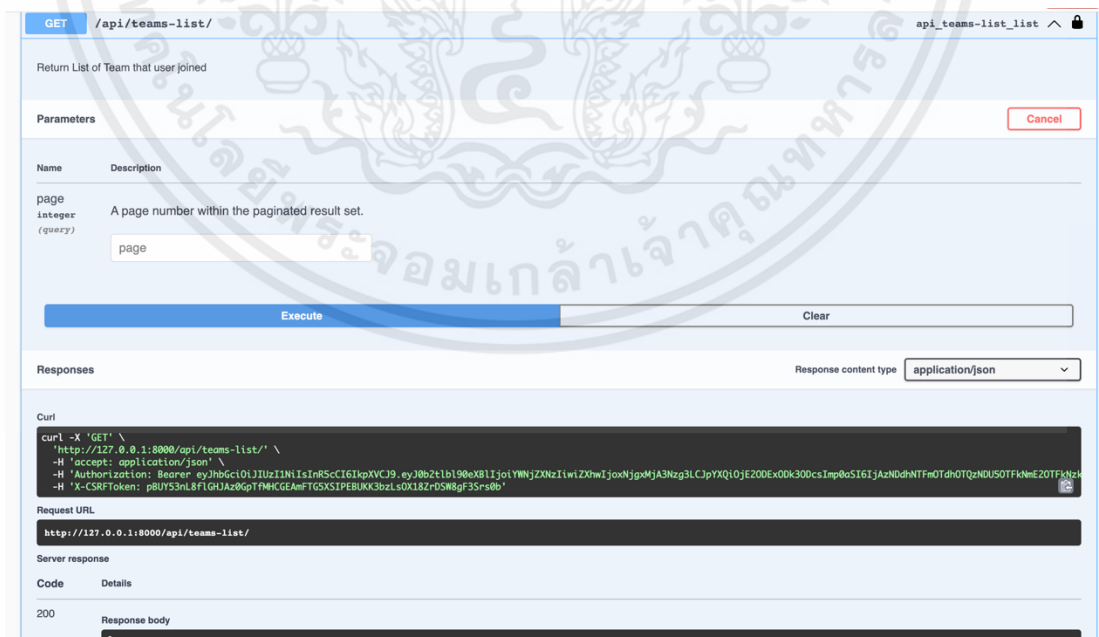
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7.3 API

View เป็น controller ของ Django framework ทำหน้าที่ในการควบคุมการรับส่งข้อมูลและสิทธิ์การเข้าถึงข้อมูลอีกด้วย



รูป 4.22 ตัวอย่าง api



รูป 4.23 ตัวอย่าง api กลุ่มเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Django REST framework admin ▾

Api Root / Exercise List

Exercise List OPTIONS GET ▾

GET /exercise/expand=submissions.user

```

HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "id": 1,
    "title": "Test",
    "instruction": "",
    "created": "2022-11-17T18:34:23.965890Z",
    "source_code": "",
    "submissions": [
      {
        "code": "TestCode01",
        "dateSubmit": "2022-11-26T15:51:01.635820Z",
        "user": {
          "username": "admin",
          "email": "admin@admin.com"
        }
      },
      {
        "code": "TestCode02",
        "dateSubmit": "2022-11-26T15:51:09.373263Z",
        "user": {
          "username": "admin",
          "email": "admin@admin.com"
        }
      }
    ]
  },
  {
    "id": 2,
    "title": "1",

```

รูป 4.24 ตัวอย่าง api การเรียกใช้แบบฝึกหัด

POST	/api/exercise/	api_exercise_create
GET	/api/exercise/{id}	api_exercise_read
PATCH	/api/exercise/{id}	api_exercise_partial_update
DELETE	/api/exercise/{id}	api_exercise_delete
POST	/api/exercise/{id}/submit	upload_file
POST	/api/multi-upload-file/	upload_multi_files
GET	/api/submission/{exerciseId}/team/{teamId}	api_submission_team_read
GET	/api/submissions-list/	api_submissions-list_list
POST	/api/team/	api_team_create
GET	/api/team/{id}	api_team_read
PATCH	/api/team/{id}	api_team_partial_update
DELETE	/api/team/{id}	api_team_delete
GET	/api/team/{id}/exercises/	api_team_exercises_list
GET	/api/team/{id}/members/	api_team_members_list
GET	/api/teams-list/	api_teams-list_list

รูป 4.25 ตัวอย่าง api ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7.5 Assessment

```

1 class FileSubmissionView(generics.GenericAPIView):
2     parser_classes = [MultiPartParser]
3     # serializer_class = FileUploadSerializer
4
5     @swagger_auto_schema(
6         operation_id="upload_file",
7         operation_description="Upload a file",
8         manual_parameters=[openapi.Parameter(
9             name="file",
10            in=openapi.IN_FORM,
11            type=openapi.TYPE_FILE,
12            # required=True,
13            description="Document"
14        )],
15    )
16    def post(self, request, pk, format=None):
17        serializer = FileUploadSerializer(data=request.data)
18
19        if serializer.is_valid():
20            # Save the uploaded file to the server
21            uploaded_file = serializer.validated_data['file']
22            file_name = uploaded_file.name
23            file_name = default_storage.save(uploaded_file.name, uploaded_file)
24            print("Uploaded file name:", file_name)
25
26            exercise = Exercise.objects.get(pk=pk)
27            filename = exercise.title
28            code = exercise.source_code
29            config = exercise.config_code
30            unittest = exercise.unittest
31            if config.strip() == '' or unittest.strip() == '':
32                return Response({'error': 'configcode or unittest error'}, status=status.HTTP_400_BAD_REQUEST)
33
34            with open(filename + '.py', 'w') as outfile:
35                outfile.write(code)
36            with open(filename + '.yaml', 'w') as outfile:
37                outfile.write(config)
38            with open(filename + '_tests.py', 'w') as outfile:
39                outfile.write(unittest)
40
41            results = 'results.json'
42
43            cmd = ['python3', '-m', 'graderutils.main', filename+'.yaml', '--develop-mode']
44            with open(results, 'w') as outfile:
45                subprocess.run(cmd, stdout=outfile)
46            with open(results, 'r') as infile:
47                data = json.load(infile)
48
49            os.remove(filename + '.py')
50            os.remove(filename + '.yaml')
51            os.remove(filename + '_tests.py')
52            os.remove(uploaded_file.name)
53            os.remove(results)
54
55            return Response(data, status=status.HTTP_200_OK)
56        else:
57            return Response(serializer.errors, status=400)

```

รูป 4.28 ตัวอย่าง Assessment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุป

5.1 บทสรุป

ภายในโครงการมีการทำงานหลักอยู่ 6 ส่วน คือ ระบบการจัดการกลุ่มเรียน, ฟังก์ชันต่าง ๆ ในกลุ่มเรียน, ระบบการทำแบบฝึกหัดในกลุ่มเรียน, ระบบการแก้ไขบัญชี, ระบบการคิดคะแนนของแต่ละแบบฝึกหัด

5.1.1 ระบบการจัดการกลุ่มเรียน

- 1) เว็บแอปพลิเคชันสามารถเรียกดูข้อมูลกลุ่มเรียนที่ผู้ใช้งานอยู่ได้
- 2) เว็บแอปพลิเคชันสามารถเพิ่มกลุ่มเรียนของผู้ใช้งานได้
- 3) เว็บแอปพลิเคชันสามารถแก้ไขกลุ่มเรียนของผู้ใช้งานได้
- 4) เว็บแอปพลิเคชันสามารถลบกลุ่มเรียนของผู้ใช้งานได้

5.1.2 ฟังก์ชันต่างๆในกลุ่มเรียน

- 1) เว็บแอปพลิเคชันสามารถใช้สร้างแบบฝึกหัดในกลุ่มเรียนได้
- 2) เว็บแอปพลิเคชันสามารถจัดการสมาชิกในกลุ่มเรียนได้
- 3) เว็บแอปพลิเคชันสามารถเรียกดูรายละเอียดการทำงานของสมาชิกได้

5.1.3 ระบบการทำแบบฝึกหัดในกลุ่มเรียน

- 1) เว็บแอปพลิเคชันสามารถทำแบบฝึกหัดและจัดเก็บโค้ดได้
- 2) เว็บแอปพลิเคชันสามารถเรียกดูโค้ดที่สมาชิกได้ทำการส่งมาแล้วได้
- 3) เว็บแอปพลิเคชันสามารถแก้ไขรายละเอียดต่างๆของแบบฝึกหัดได้
- 4) เว็บแอปพลิเคชันสามารถเพิ่มตัวอย่าง โค้ดและ Unit test ของแต่ละแบบฝึกหัด

5.1.4 ระบบการแก้ไขบัญชี

- 1) เว็บแอปพลิเคชันสามารถใช้แก้ไขรายละเอียดต่างๆของบัญชีผู้ใช้ได้

5.1.5 ระบบการคิดคะแนน

- 1) เว็บแอปพลิเคชันสามารถตรวจสอบจุดผิดพลาดของโค้ดได้
- 1) เว็บแอปพลิเคชันสามารถให้คะแนนตามความถูกต้องของโค้ดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ปัญหาและอุปสรรคที่พบ

5.2.1 ปัญหาในด้านการ Design

จากการที่ทำ Design ไม่ละเอียดถี่ถ้วน ทำให้การทำงานเป็นไปได้ล่าช้า โดยต้องทำการแก้ไข Design ควบคู่กับการเขียนโค้ดไปด้วย

5.2.2 ปัญหาในการสื่อสารกับเพื่อนร่วมงาน

เนื่องจากกำหนด Requirement ไม่ชัดเจนและการประชุมงานน้อย ทำให้งานทำได้ล่าช้า และเข้าใจไม่ตรงกันในบางส่วนงาน

5.2.3 ปัญหาการเขียนโปรแกรม Frontend

ปัญหาของผู้พัฒนาที่ไม่ทำการศึกษาให้เข้าใจถี่ถ้วน รวมกับความสะเพร่าทำให้งานในส่วน Frontend ดำเนินไปได้อย่างล่าช้ากว่าที่ควรจะเป็น งานจึงเสร็จล่าช้า

5.3 แนวทางการพัฒนาต่อ

- 1) ออกแบบระบบให้ใช้งานได้สะดวกกว่านี้
- 2) ในส่วนการตรวจคะแนน (Auto Assessment) สามารถทำให้ใช้งานได้ง่ายกว่านี้ผ่านตัว User Interface
- 3) รองรับภาษาการใช้งานอื่น

บรรณานุกรม

Sam Morley. 2020. **Automated Assessment of Python Code**. [Online].

Available : <https://journals.gre.ac.uk/index.php/msor/article/view/1089>

Vreda Pieterse. 2013. **Automated Assessment of Programming Assignments**. [Online].

Available :

https://www.researchgate.net/publication/262328132_Automated_Assessment_of_Programming_Assignments

React A JavaScript library for building user interfaces. [Online].

Available : <https://reactjs.org/>

Developer Team, BorntoDev Co., Ltd. 2563. **React 101**. [Online].

Available : <https://www.borntodev.com/2020/07/15/react-101/>

Redux A predictable state container for JavaScript apps. [Online].

Available : <https://redux.js.org/>

Rungpharit. 2563. **Redux คืออะไร?**. [Online].

Available : [https://medium.com/@rungpharit.sir/react-redux-](https://medium.com/@rungpharit.sir/react-redux-%E0%B8%87%E0%B9%88%E0%B8%B2%E0%B8%A2%E0%B9%86-4d36c2cf11cc)

[%E0%B8%87%E0%B9%88%E0%B8%B2%E0%B8%A2%E0%B9%86-4d36c2cf11cc](https://medium.com/@rungpharit.sir/react-redux-%E0%B8%87%E0%B9%88%E0%B8%B2%E0%B8%A2%E0%B9%86-4d36c2cf11cc)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tailwind CSS – Get started with tailwind CSS. [Online].

Available : <https://tailwindcss.com/docs/installation>

Charintorn Ruanglaikram. 2565. **Tailwind CSS เฟรมเวิร์กที่ช่วยให้ Dev ทำงานง่ายขึ้น.**

[Online]. Available : <https://morphos.is/th/blog/tailwind-css-a-framework-that-makes-dev-work-easier>

Axios – Getting Started. [Online].

Available : <https://axios-http.com/docs/intro>

MySQL. [Online].

Available : <https://www.mysql.com/>

LiftBytes. 2565. **คุณสมบัติ MySQL.** [Online].

Available : <https://vidabytes.com/th/characteristicas-del-mysql/>

Code Mirror. [Online].

Available : <https://codemirror.net/>

Ihalajl. **Python-grader-utils.** [Online].

Available : <https://github.com/apluslms/python-grader-utils>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้