

การจัดการห้องเรียนสำหรับการเขียนโปรแกรมอย่างมีประสิทธิภาพ

EFFECTIVE CLASSROOM MANAGEMENT FOR
PROGRAMMING COURSES



ปริญญาโทพนธ์นี้เป็นส่วนหนึ่งของกรรมกษาตามหลักสูตรวิชาศาสตรบรจรม

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การจัดการห้องเรียนสำหรับการเขียนโปรแกรมอย่างมีประสิทธิภาพ
EFFECTIVE CLASSROOM MANAGEMENT FOR PROGRAMMING
COURSES



T146243



เลขทะเบียน 146243
รับเดือนปี 25 10ย 2560

b. 12841663
l.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคเรียนที่ 2 ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**EFFECTIVE CLASSROOM MANAGEMENT FOR PROGRAMMING
COURSES**



**A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2/2015

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2016

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองปริญญาโท ประจำปีการศึกษา 2558
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การจัดการห้องเรียนสำหรับการเขียนโปรแกรมอย่างมีประสิทธิภาพ
EFFECTIVE CLASSROOM MANAGEMENT FOR
PROGRAMMING COURSES

ผู้จัดทำ

1. นางสาวเชาวนี จันทร์ทอง รหัสประจำตัว 55070030
2. นายณภัทร รุ่งรุจีเมฆ รหัสประจำตัว 55070035



..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร. ธนิศา นุ่มนนท์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงการ	ระบบการจัดการการเรียนรู้การเขียนโปรแกรมในห้องเรียนอย่างมีประสิทธิภาพ	
นักศึกษา	เขาวนีย์ จันทร์ทอง	รหัสนักศึกษา 55070030
	ณภัทร รุ่งรุจีเมฆ	รหัสนักศึกษา 55070035
ปริญญา	วิทยาศาสตร์บัณฑิต	
สาขาวิชา	เทคโนโลยีสารสนเทศ	
ปีการศึกษา	2558	
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. ธนิตา นุ่มนนท์	

บทคัดย่อ

ในปัจจุบัน การเรียนรู้การเขียนโปรแกรมถือเป็นเรื่องสำคัญมาก เนื่องจากเทคโนโลยีส่วนใหญ่ในปัจจุบัน จำเป็นต้องใช้การเขียนโปรแกรมในการพัฒนา แต่การเรียนรู้ด้านการเขียนโปรแกรมคอมพิวเตอร์ในห้องเรียนนั้นกลับยังไม่มีประสิทธิภาพเท่าที่ควร

ดังนั้นโครงการนี้จึงนำเสนอระบบการจัดการการเรียนรู้การเขียนโปรแกรมในห้องเรียนอย่างมีประสิทธิภาพ ซึ่งเป็นเว็บไซต์ที่สอนการเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์ต่างๆ โดยจะเป็นระบบห้องเรียนอิเล็กทรอนิกส์ ซึ่งเปรียบเสมือนห้องเรียนห้องหนึ่ง ที่จะมีรหัสกำหนดการเข้าถึงที่ถูกสร้างขึ้นมา และผู้เรียนต้องใช้รหัสนี้เพื่อเข้าใช้งานห้องเรียน ผู้เรียนและผู้สอนสามารถเขียนโปรแกรมและประมวลผลเพื่อดูผลลัพธ์ผ่านทางเว็บไซต์ โดยผู้เรียนสามารถเห็นโค้ดของผู้สอน และในขณะเดียวกัน ผู้สอนก็จะสามารถดูและแก้ไขโค้ดของผู้เรียนทุกคนในห้องเรียนได้เช่นกัน นอกจากนี้ยังมีระบบแบบทดสอบ ซึ่งเป็นระบบสำหรับช่วยอำนวยความสะดวกให้ผู้เรียนได้ฝึกฝนการทำโจทย์และให้ระบบทำการตรวจได้ทันที เพื่อให้ผู้สอนสามารถประเมินความรู้ความเข้าใจของผู้เรียนได้อีกด้วย

Project Title	Effective Classroom Management for Programming Courses	
Student	Chaowanee Janthong	Student ID 55070030
	Napat Rungruchimek	Student ID 55070035
Degree	Bachelor of Science	
Program	Information Technology	
Academic Year	2015	
Advisor	Assistant Professor Dr. Thanisa Numnonda	

ABSTRACT

Nowadays, learning computer programming still lack real-time online interaction and collaboration between teachers and students. This makes learning computer programming not enough efficiently.

Therefore, this project introduces Effective Classroom Management for Programming Courses. This system can be called Coder-Space which mainly provides real-time interactive classroom of various computer programming. A teacher can create a E-classroom and distribute the key that generated by the Website to all their desired students. Coder-Space allows teachers and students to write codes and see the results through the Website. All students in the classroom can view the teacher's code. In addition, teachers can monitor all their student's codes. Coder-Space also provides quiz system for most effective learning.

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่องระบบการจัดการการเรียนรู้การเขียนโปรแกรมในห้องเรียนอย่างมีประสิทธิภาพฉบับนี้นั้น สามารถสำเร็จลุล่วงได้ด้วยความรู้และการสนับสนุนจาก ผู้ช่วยศาสตราจารย์ ดร. ธนินา นุ่มนนท์ อาจารย์ที่ปรึกษาของโครงการและคณะอาจารย์จากคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่านที่ได้ให้แรงสนับสนุนและความร่วมมือเป็นอย่างดีในทุกๆ ด้าน ไม่ว่าจะเป็น ทางด้านความรู้และข้อเสนอแนะต่างๆ การให้ข้อมูล การเสนอแนวคิด และแนวทางในการแก้ปัญหาหรืออุปสรรคที่พบเจอขณะที่ได้ศึกษาและพัฒนาโครงการดังกล่าว จนสามารถเกิดเป็นปริญญานิพนธ์เล่มนี้ที่เสร็จสมบูรณ์ ทางผู้จัดทำจึงขอแสดงความขอบพระคุณเป็นอย่างยิ่งไว้ ณ ที่นี้

ผู้จัดทำขอแสดงความขอบพระคุณต่อเจ้าหน้าที่งานสนับสนุนเทคโนโลยีสารสนเทศ และเจ้าหน้าที่งานทะเบียน คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้ความร่วมมือและประสานงานด้วยดีตลอดการจัดทำจนสามารถเกิดเป็นปริญญานิพนธ์ฉบับนี้ได้

เชาวนี จันทร์ทอง
ณภัทร รุ่งรุจิเมฆ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการพัฒนา.....	1
1.3 ขอบเขตโครงการ.....	2
1.4 ขั้นตอนการพัฒนาระบบ.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีและหลักการ.....	3
2.1 Node.js.....	3
2.2 HTML5 CSS3 JavaScript.....	3
2.2.1 HTML5.....	3
2.2.2 CSS3.....	4
2.2.3 JavaScript.....	4
2.3 Mongo DB.....	5
2.4 Docker.....	5
2.5 Websocket.....	5
บทที่ 3 การวิเคราะห์และออกแบบระบบ.....	8
3.1 ศึกษางานระบบเดิม.....	8
3.1.1 ระบบเกี่ยวกับการเรียนการสอนโปรแกรม.....	8
3.1.2 ระบบที่มีหลักการทำงานใกล้เคียงกัน.....	10
3.2 ปัญหาที่พบในระบบปัจจุบัน.....	13
3.3 แนวคิดในการพัฒนา.....	13
3.3.1 การนำ Docker มาใช้.....	13
3.3.2 การนำ MongoDB มาใช้.....	14

สารบัญ (ต่อ)

	หน้า
3.4 การวิเคราะห์ความต้องการ.....	16
3.4.1 ความต้องการที่เป็นหน้าที่หลักของระบบ.....	16
3.4.2 ความต้องการที่ไม่ใช่หน้าที่หลักของระบบ.....	16
3.5 วิเคราะห์และวิจารณ์ระบบที่ต้องการออกแบบ.....	17
3.5.1 แผนภาพยูสเคส.....	17
3.5.1.1 ผู้เกี่ยวข้องในระบบ.....	17
3.5.1.2 ฟังก์ชันการทำงานหลัก.....	17
3.5.1.3 แผนภาพยูสเคส.....	18
3.5.1.4 รายละเอียดการทำงานของแต่ละยูสเคส.....	19
3.5.2 แผนภาพกิจกรรม.....	30
บทที่ 4 ระบบต้นแบบ.....	38
4.1 ผลการทดลองและการศึกษา.....	38
4.2 เว็บแอปพลิเคชัน (Web Application).....	38
4.2.1 ส่วนของผู้สอน.....	39
4.2.2 ส่วนของผู้เรียน.....	44
บทที่ 5 สรุปผลและข้อเสนอแนะ.....	49
5.1 สรุปผลการดำเนินงาน.....	49
5.2 ปัญหาและอุปสรรคที่พบในการพัฒนาระบบ.....	50
5.3 ข้อเสนอแนะและแนวทางการพัฒนาระบบในอนาคต.....	50
บรรณานุกรม.....	51
ภาคผนวก.....	52
ภาคผนวก ก. คู่มือการติดตั้งระบบและคู่มือการใช้งานระบบ.....	53
ประวัติผู้เขียน.....	72

สารบัญตาราง

ตารางที่	หน้า
3.1 คำอธิบายยูสเคสสร้างห้องเรียน Coder-Space.....	19
3.2 คำอธิบายยูสเคสเข้าห้องเรียน Coder-Space.....	20
3.3 คำอธิบายยูสเคสยกเลิกห้องเรียน Coder-Space.....	21
3.4 คำอธิบายยูสเคสออกจากห้องเรียน Coder-Space.....	22
3.5 คำอธิบายยูสเคสเรียกดู แก้ไขและประมวลผลโค้ดในหน้าต่างของผู้เรียนผ่านหน้าจอของผู้สอน.....	23
3.6 คำอธิบายยูสเคสเขียนโค้ดและประมวลผลโค้ดผ่านระบบ Coder-Space.....	24
3.7 คำอธิบายยูสเคสเรียกดูโค้ดในหน้าต่างของผู้สอนผ่านหน้าจอของผู้เรียน.....	25
3.8 คำอธิบายยูสเคสส่งข้อความถึงกัน.....	26
3.9 คำอธิบายยูสเคสสร้างแบบทดสอบประเภทตอบคำถามและประเภท Grader.....	27
3.10 คำอธิบายยูสเคส ทำแบบทดสอบประเภทตอบคำถามและประเภท Grader.....	28
3.11 คำอธิบายยูสเคสผลคะแนนการทำแบบทดสอบ.....	29

สารบัญรูป

รูปที่	หน้า
2.1 รูปภาพแสดงการติดต่อของ Docker	5
2.2 รูปภาพแสดงการทำงานของ Docker.....	6
2.3 รูปภาพแสดงการทำงานแบบ HTTP Server.....	7
2.4 รูปภาพแสดงการทำงานแบบ Websocket Server.....	7
3.1 ตัวอย่างหน้าเว็บไซต์ Codeschool.com.....	8
3.2 ตัวอย่างหน้าเว็บไซต์ Codecademy.com.....	9
3.3 ตัวอย่างระบบ Netsupport.....	10
3.4 ตัวอย่างระบบ Teamviewer.....	11
3.5 ภาพการทำงานของ Docker.....	13
3.6 ภาพ chat_log_schema.....	14
3.7 ภาพ classroom_schema.....	14
3.8 ภาพ container_schema.....	15
3.9 ภาพ quiz_schema.....	15
3.10 ภาพ quiz_log_schema.....	15
3.11 ภาพ user_schema.....	15
3.12 แผนภาพยูสเคส.....	18
3.13 แผนภาพกิจกรรมของการสร้างห้องเรียนในระบบ Coder-Space.....	30
3.14 แผนภาพกิจกรรมของการเข้าห้องเรียนในระบบ Coder-Space.....	31
3.15 แผนภาพกิจกรรมของการเขียนโค้ดและประมวลผลโค้ดของระบบ Coder-Space.....	32
3.16 แผนภาพกิจกรรมของการดู แก้ไข และประมวลผลโค้ดหน้า Editor ของผู้เรียน.....	33
3.17 แผนภาพกิจกรรมของการสร้างแบบทดสอบประเภทตอบคำถาม.....	34
3.18 แผนภาพกิจกรรมของการสร้างแบบทดสอบสำหรับระบบ Grader.....	35
3.19 แผนภาพกิจกรรมของการทำแบบทดสอบประเภทตอบคำถาม.....	36
3.20 แผนภาพกิจกรรมของการทำแบบทดสอบสำหรับระบบ Grader.....	37
4.1 หน้าแรกของเว็บไซต์ Coder Planet.....	38
4.2 หน้าสร้างห้องเรียน Coder-Space.....	39
4.3 หน้า Editor ของผู้สอน.....	40
4.4 คู่มือ Editor ของผู้เรียน.....	40

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.5 สร้างแบบทดสอบสำหรับตอบคำถาม.....	41
4.6 หน้าสร้างแบบทดสอบสำหรับระบบ Grader.....	42
4.7 หน้าดูผลคะแนนและผลประเมินการทดสอบทั้งหมด.....	43
4.8 หน้าส่งข้อความติดต่อกัน (Chat)	43
4.9 หน้ารวมห้องเรียนทั้งหมดใน Coder-Space.....	44
4.10 หน้า Editor ของผู้เรียน.....	45
4.11 ดูหน้า Editor ของผู้สอน.....	46
4.12 หน้าทำแบบทดสอบประเภทตอบคำถาม.....	47
4.13 หน้าทำแบบทดสอบประเภท Grader.....	48



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบัน การเรียนรู้การเขียนโปรแกรมถือเป็นเรื่องสำคัญมาก เนื่องจากเทคโนโลยีส่วนใหญ่ในปัจจุบัน จำเป็นต้องใช้การเขียนโปรแกรมในการพัฒนา แต่การเรียนรู้ด้านการเขียนโปรแกรมคอมพิวเตอร์ในห้องเรียนนั้นกลับยังไม่มีประสิทธิภาพเท่าที่ควร เช่น การเสียเวลาไปกับการลงเตรียมทรัพยากรที่จำเป็นต้องใช้ในการเรียนเขียนโปรแกรม หรือในขณะที่เรียนเขียนโปรแกรม หากผู้เรียนมีข้อสงสัยหรือปัญหา ผู้สอนจำเป็นต้องเดินไปอธิบาย แก้ไขปัญหาที่หน้าจอของผู้เรียน ซึ่งเป็นการเสียเวลา และยากต่อการที่จะดูแลผู้เรียนทุกคนได้อย่างทั่วถึง

จากที่กล่าวมาข้างต้น จึงได้มีการพัฒนาโครงการจัดการห้องเรียนสำหรับการเขียนโปรแกรมอย่างมีประสิทธิภาพ ซึ่งเป็นเว็บไซต์ที่สอนการเขียนโปรแกรมด้วยภาษาต่างๆ เป็นช่องทางให้ผู้เชี่ยวชาญสามารถถ่ายทอดความรู้ให้แก่ผู้ที่สนใจได้ผ่านระบบห้องเรียนอิเล็กทรอนิกส์ หรือ Coder-Space ซึ่งเปรียบเสมือนห้องเรียนห้องหนึ่ง ซึ่งจะมีรหัสกำหนดการเข้าถึง โดยถ้าผู้เรียนทราบรหัสของห้องเรียน ก็จะสามารถเข้าเรียนได้ ซึ่งภายในห้องเรียน ผู้เรียนและผู้สอนสามารถเขียนโปรแกรมและประมวลผลเพื่อผลลัพธ์ผ่านเว็บไซต์ และในขณะเดียวกัน ผู้สอนสามารถดูแลและแก้ไขโค้ดของผู้เรียนทุกคนในห้องเรียนได้เช่นกัน อีกทั้งยังมีระบบแบบทดสอบ โดยแบ่งออกเป็นแบบตอบคำถาม และแบบระบบ Grader ซึ่งเป็นระบบสำหรับช่วยอำนวยความสะดวกให้ผู้เรียนได้ฝึกฝนการทำโจทย์แล้วอัปขึ้นระบบเพื่อให้ระบบทำการตรวจและประเมินผลโปรแกรมโดยอัตโนมัติได้ทันที เพื่อให้ผู้สอนสามารถประเมินความรู้ความเข้าใจของผู้เรียนทั้งคลาสได้

1.2 ความมุ่งหมายและจุดประสงค์การพัฒนา

1. เพื่อให้ผู้สอนสามารถติดตามการกระทำต่างๆของผู้เรียนได้อย่างง่าย และมีประสิทธิภาพ
2. เพื่อให้ผู้สอนสามารถวัดผลการเรียนของผู้เรียนผ่านระบบได้
3. เพื่อให้ผู้เรียนและผู้สอนประหยัดเวลาในการเตรียมโปรแกรมที่จำเป็นสำหรับการเรียนและการสอน

1.3 ขอบเขตของโครงการ

1. เป็นเว็บไซต์ที่ให้ผู้สอนถ่ายทอดความรู้เฉพาะภาษาที่ใช้ในการเขียนโปรแกรมเท่านั้น
2. ระบบ Coder-Space ผู้สอนสามารถสอนได้เฉพาะภาษาที่สามารถเรียกใช้ผ่าน Command line ได้เท่านั้น
3. ระบบ Coder-Space ผู้สอนสามารถเลือกหน้าจอของผู้เรียน เพื่อที่จะเข้าไปดู แก้ไข และประมวลผลโค้ดได้เพียงทีละคนผ่านหน้าจอของผู้สอน
4. จำเป็นต้องเชื่อมต่ออินเทอร์เน็ต

1.4 ขั้นตอนการพัฒนา

1. ศึกษาความรู้ที่ต้องใช้ในการทำเว็บในส่วนของผู้เรียกใช้บริการ (Client-side) โดยมีการใช้ HTML5, CSS3 และ JavaScript
2. ศึกษาความรู้ที่ต้องใช้ในการทำเว็บในส่วนของผู้ให้บริการ (Server-side) โดยใช้ Node.js
3. ศึกษาวิธีการเก็บข้อมูล(Database) แบบ NoSQL ด้วย MongoDB
4. กำหนดขอบเขตของโครงการ
5. ออกแบบส่วนติดต่อกับผู้ใช้ตามหลัก UI และ UX ที่ดี
6. พัฒนาระบบในส่วนที่ถือเป็นส่วนหลักของระบบ
7. พัฒนาระบบทั้งหมด
8. ทำการสร้างเนื้อหา (Content) ของเว็บไซต์
9. ทดลองใช้ระบบ และปรับปรุงแก้ไขข้อผิดพลาดที่พบ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ความรู้ในการเขียนเว็บไซต์ทั้งในส่วนของผู้เรียกใช้บริการและผู้ให้บริการ
2. ได้ความรู้ในการเขียนเว็บไซต์ที่ตอบสนองผู้ใช้ได้ทันที (Real time web)
3. ได้เว็บไซต์ที่เป็นประโยชน์ในการนำไปใช้ประกอบการเรียนการสอน
4. ได้เว็บไซต์ที่เปิดโอกาสให้ผู้เชี่ยวชาญได้ถ่ายทอดความรู้ด้านนั้นๆ สู่ผู้ที่ต้องการเรียนรู้อื่น
5. สนับสนุนการเรียนการสอนเขียนโปรแกรมในห้องเรียนให้มีประสิทธิภาพมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 Node.js

Node.js คือ ภาษา JavaScript ที่ทำงานในฝั่งผู้ให้บริการ (Server) ซึ่งในปกติแล้วเว็บทั่วๆ ไปแทบทุกเว็บจำเป็นต้องใช้ภาษา JavaScript มาช่วยในการทำงานฝั่งผู้ใช้บริการ (Client) อยู่แล้ว จึงถือเป็นข้อดีข้อหนึ่ง ที่ทำให้ผู้พัฒนาคุ้นเคยและทำความเข้าใจ Node.js ได้ง่าย เนื่องจากมีโครงสร้างภาษา (Syntax) เหมือนกับภาษา JavaScript ที่ใช้อยู่แล้ว

นอกจากนั้น Node.js ยังทำงานภายใต้แนวคิดการทำงานแบบไม่พร้อมกัน (Asynchronous) ซึ่งจะช่วยให้สามารถเริ่มประมวลผลงานใหม่ โดยที่ไม่ต้องรอการประมวลผลงานเก่าให้เสร็จก่อน โดยแนวคิดนี้ถือเป็นแนวคิดสำคัญที่ช่วยให้สร้างเว็บไซต์ที่จะตอบสนองผู้ใช้ได้แบบทันที

ในแง่ของภาษาเขียน โปรแกรม Node.js ถือเป็นภาษาใหม่ ที่มีแนวโน้มการเจริญเติบโตสูงมาก อ้างอิงจากชุมชนต่างๆ ที่รวมนักพัฒนาโปรแกรมเข้าด้วยกัน และด้วยอัตราการเจริญเติบโตนี้ ทำให้ในชุมชนที่รวมนักพัฒนาโปรแกรม พร้อมทั้งจะให้คำแนะนำกับนักพัฒนาโปรแกรมใหม่ๆ ที่ต้องการความช่วยเหลือได้ทันที

2.2 HTML5 CSS3 และ JavaScript

2.2.1 HTML5

HTML5 เป็นภาษาที่ใช้ในการเขียนเว็บไซต์ พัฒนามาจาก HTML โดยโครงสร้างการเขียนจะอยู่ในรูปแบบของ tag ซึ่งทำหน้าที่ควบคุมการแสดงผลของข้อความ รูปภาพ เสียง หรือวิดีโอ โดยสามารถดูการแสดงผลจากที่เราเขียนได้ผ่านเว็บเบราว์เซอร์ (Web Browser) เช่น Google Chrome, Mozilla Firefox, Internet Explorer, Safari เป็นต้น

HTML5 เป็นเวอร์ชันล่าสุดที่ถูกพัฒนาขึ้นมา ซึ่งในเวอร์ชันนี้ได้มี tag เพิ่มขึ้นใหม่ และมีบาง tag ที่เลิกใช้ไป อีกทั้งยังมี feature ที่น่าสนใจเพิ่มขึ้นมา เช่น สามารถแสดงภาพ เสียง และสื่อกราฟิกแอนิเมชันต่าง ๆ โดยที่ไม่ต้องมีปลั๊กอิน และใช้ Canvas ในการวาดรูปได้ แต่จำเป็นต้องใช้ JQuery หรือ Javascript ช่วยควบคุมการทำงานของมันด้วย เป็นต้น ทำให้ HTML5 เป็นที่นิยมในปัจจุบันและสามารถแสดงผลได้ทุกเว็บเบราว์เซอร์

2.2.2 CSS3

CSS3 หรือ Cascading Style Sheet เป็นภาษาที่ใช้ร่วมกับ HTML ทำหน้าที่ในการกำหนดคุณสมบัติให้กับ Element ของ HTML เช่น <body> <p> เป็นต้น ซึ่งสามารถกำหนด สี ฟอนต์ ตำแหน่ง ขนาด พื้นหลัง เส้นขอบ และอื่นๆ เพื่อให้มีการแสดงผลที่สวยงาม มีสีสัน น่าดึงดูดมากยิ่งขึ้น

การใช้ CSS ในการจัดรูปแบบการแสดงผล จะช่วยลดการใช้ภาษา HTML ในการตกแต่งเอกสารเว็บเพจ ทำให้โค้ดภายในเอกสาร HTML เหลือเพียงเนื้อหา ทำให้เข้าใจง่ายขึ้น และสามารถกำหนดรูปแบบการแสดงผลจากคำสั่ง Style Sheet ชุดเดียวกัน ให้มีผลกับเอกสาร HTML ทั้งหน้าหรือในหลายๆหน้าได้ ทำให้สามารถทำการแก้ไขได้ง่าย ไม่ต้องไล่ตามแก้ไข HTML Tag ต่างๆที่ทั้งเอกสาร

โดย CSS จะเขียนไว้ในไฟล์เดียวกับ HTML หรือแยกเป็นไฟล์ Style Sheet ต่างหากแล้วเขียนคำสั่งเรียกใช้ภายหลังได้ ซึ่งคนส่วนใหญ่นิยมเขียนแยกไฟล์มากกว่า

2.2.3 JavaScript

JavaScript เป็นภาษา Script เริงวัตถุ ใช้เพื่อสร้างเว็บไซต์ โดยใช้ร่วมกับ HTML เพื่อให้การแสดงผลบางส่วนมีการเคลื่อนไหว หรือทำให้เว็บไซต์มีความน่าดึงดูดและน่าสนใจ อีกทั้ง JavaScript สามารถเขียนเปลี่ยนแปลง HTML Element ได้ นั่นคือสามารถเปลี่ยนรูปแบบการแสดงผลของเว็บไซต์ได้ และทำให้ตอบสนองผู้ใช้งานได้มากขึ้น เช่น เขียน JavaScript ร่วมกับ HTML เพื่อตรวจสอบความผิดพลาด เมื่อกรอกข้อมูลในช่องของอีเมลผิด ก็จะมีการแจ้งเตือนให้ผู้ใช้ทราบได้ทันทีว่าข้อมูลที่กรอกไม่ถูกต้อง เป็นต้น

การทำงานของ JavaScript เกิดขึ้นบนบราวเซอร์ (เรียกว่าเป็น Client-Side Script) ดังนั้นไม่จำเป็นต้องใช้เซิร์ฟเวอร์อะไร หรือที่ไหน ก็ยังคงสามารถใช้ JavaScript ในเว็บเพจได้ ต่างกับภาษาสคริปต์อื่น เช่น PHP ซึ่งต้องทำงานที่ตัวเครื่องเซิร์ฟเวอร์ (Server-Side Script) ดังนั้นจึงต้องใช้บนเซิร์ฟเวอร์ ที่สนับสนุนภาษาเหล่านี้เท่านั้น จากลักษณะดังกล่าวทำให้ JavaScript มีข้อจำกัด คือไม่สามารถรับและส่งข้อมูลต่างๆ กับเซิร์ฟเวอร์โดยตรง เช่น การอ่านไฟล์จากเซิร์ฟเวอร์ เพื่อนำมาแสดงบนเว็บเพจ หรือรับข้อมูลจากผู้ใช้ เพื่อนำไปเก็บบนเซิร์ฟเวอร์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 MongoDB

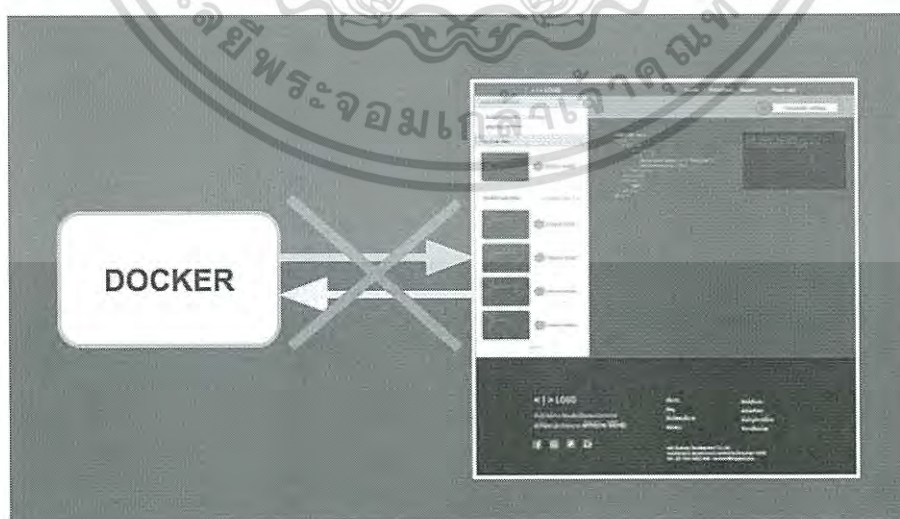
MongoDB คือ ฐานข้อมูล แบบ NoSQL(Not Only SQL) ชนิดหนึ่ง ซึ่งรูปแบบของฐานข้อมูลที่ใช้กันส่วนใหญ่ในปัจจุบันจะเป็นฐานข้อมูล แบบ Relational Database โดยที่ฐานข้อมูลแบบ NoSQL จะมีข้อดีกว่าฐานข้อมูลแบบ Relational Database ในด้านความเร็วในการเขียนข้อมูลลงฐานข้อมูล ซึ่งถือเป็นส่วนสำคัญสำหรับโปรแกรมที่ต้องการจะเก็บข้อมูลบ่อยๆ เพื่อนำข้อมูลเหล่านั้นไปวิเคราะห์

ในส่วนของ MongoDB จะทำการเก็บข้อมูลเป็นแบบเชิงเอกสาร (Document-Oriented) ซึ่งมีข้อดีคือ รูปแบบของข้อมูลแต่ละชุด ไม่จำเป็นที่จะต้องเหมือนกัน สามารถเพิ่มความซับซ้อนให้ข้อมูลแต่ละตัวได้ตามความเหมาะสม

ข้อดีที่กล่าวมาข้างต้น ถือเป็นข้อได้เปรียบสำคัญที่ทำให้ต้องเลือกใช้ฐานข้อมูลนี้ แต่นอกจากข้อดีที่กล่าวมาข้างต้น ยังมีข้อดีอีกหลายอย่าง เช่น การเพิ่มขนาดของฐานข้อมูลที่สามารถทำได้อย่างรวดเร็ว การสำรองข้อมูล (Backup) ที่ทำได้โดยง่าย การแก้ไขข้อมูลได้อย่างรวดเร็วซึ่งถ้าเทียบกับการแก้ไขข้อมูลของ Relational Database จะถือว่ามีความเร็ว และความต่อเนื่องกว่ามาก เป็นต้น

2.4 Docker

Docker คือ โปรแกรมที่จำลองระบบปฏิบัติการ (Operating System) โดยที่มีจุดเด่นคือ เราสามารถสร้างแม่พิมพ์ของระบบปฏิบัติการ (Image) เพื่อที่จะใช้แม่พิมพ์นั้นมาสร้างเป็นระบบปฏิบัติการย่อยๆ (Container) ที่เหมือนกับแม่พิมพ์ทุกประการ โดยที่มีขนาดเล็กและกินทรัพยากรเครื่องน้อย และ Docker ไม่ได้ทำการติดต่อกับเว็บไซต์โดยตรง ดังรูปที่ 2.1



รูปที่ 2.1 รูปภาพแสดงการติดต่อของ Docker

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Docker จะถูกนำไปประยุกต์ใช้เพื่อให้นักพัฒนาโปรแกรมต่างๆทำได้ง่ายขึ้น เช่น เมื่อโปรแกรมที่เราจะพัฒนา ต้องการโปรแกรมอื่นๆ เพื่อที่จะทำงานได้ เช่น ฐานข้อมูล เซิร์ฟเวอร์ จำลอง เป็นต้น เมื่อเราต้องการที่จะพัฒนาโปรแกรมของเรานี้ที่เครื่องอื่น ก็จะสามารถทำได้ง่าย โดยลงโปรแกรมที่จำเป็นในระบบปฏิบัติการที่ทำงานอยู่บน Docker แล้วนำระบบปฏิบัติการนั้น มาสร้างเป็นแม่พิมพ์ของระบบปฏิบัติการ ทำให้ทุกครั้งที่ต้องการจะพัฒนาโปรแกรมนี้นในเครื่องอื่นๆ จะสามารถทำได้ง่าย เพียงแค่สร้างระบบปฏิบัติการจากแม่พิมพ์ของระบบปฏิบัติการที่เราสร้างไว้ สามารถดูการทำงานของ Docker ได้ดังรูปที่ 2.2



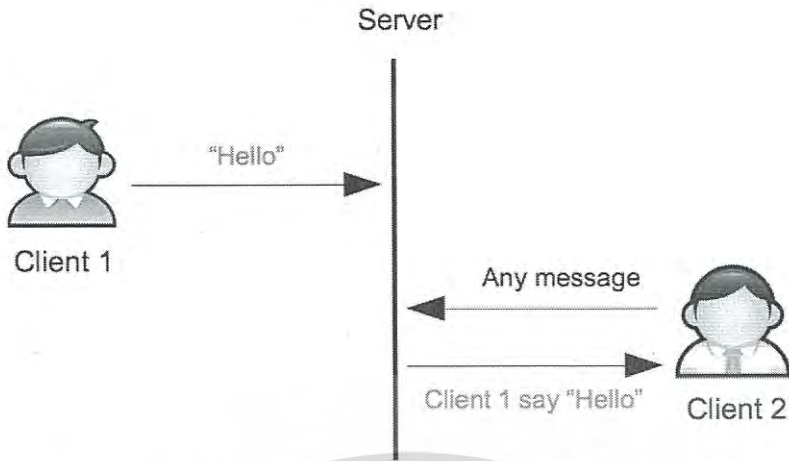
รูปที่ 2.2 รูปภาพแสดงการทำงานของ Docker

2.5 Websocket

Websocket คือ ช่องทางการสื่อสารระหว่างเซิร์ฟเวอร์กับผู้ใช้รูปแบบหนึ่ง โดยที่การทำงานของ Websocket จะทำงาน โดยการสร้างท่อระหว่างผู้ใช้กับเซิร์ฟเวอร์ หากผู้ใช้ต้องการส่งข้อมูลไปให้เซิร์ฟเวอร์ ก็สามารถส่งได้ทันที ในขณะที่เดียวกัน หากเซิร์ฟเวอร์ต้องการส่งข้อมูลให้ผู้ใช้ ก็สามารถทำได้ทันที โดยผู้ใช้ไม่ต้องร้องขอ (Request) ด้วยหลักการทำงานนี้ ทำให้ Websocket เหมาะสำหรับนำมาใช้สำหรับเว็บไซต์ที่ทำงานแบบ Real time

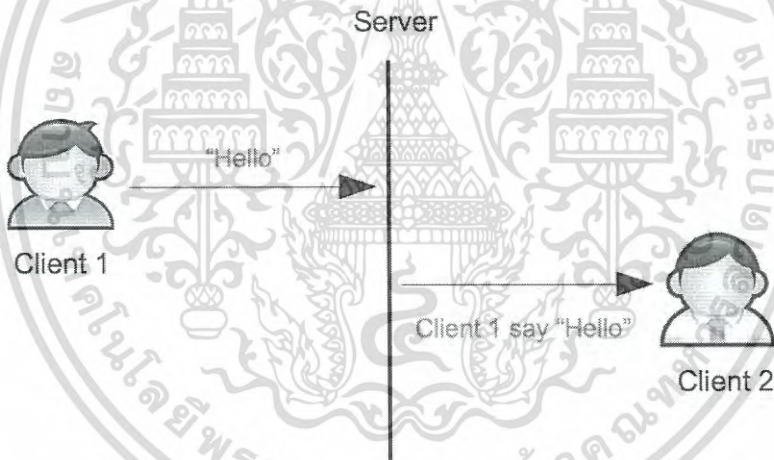
การทำงานแบบ HTTP Server ดังรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 รูปภาพแสดงการทำงานแบบ HTTP Server

การทำงานแบบ WebSocket Server ดังรูปที่ 2.4



รูปที่ 2.4 รูปภาพแสดงการทำงานแบบ WebSocket Server

จากรูปที่ 2.4 จะเห็นว่า Client 2 ไม่จำเป็นต้องร้องขอไปที่ Server เพื่อความีข้อความใหม่มาหรือไม่ แต่เมื่อใดที่ Server ได้รับข้อความจาก Client 1 ก็จะส่งข้อความนั้นไปให้ Client 2 ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การวิเคราะห์และออกแบบระบบ

3.1 ศึกษางานระบบปัจจุบัน

3.1.1 ระบบเกี่ยวกับการเรียนการเขียนโปรแกรม

ในปัจจุบันจากการค้นคว้า ทำให้เห็นว่าเว็บไซต์ที่ต้องการเผยแพร่ความรู้เกี่ยวกับการเขียนโปรแกรม จะมีอยู่จำนวนหนึ่ง ซึ่งขอยกตัวอย่างเว็บไซต์ที่มีลักษณะใกล้เคียงกับระบบที่ต้องการจะนำมา 2 เว็บไซต์ ดังนี้

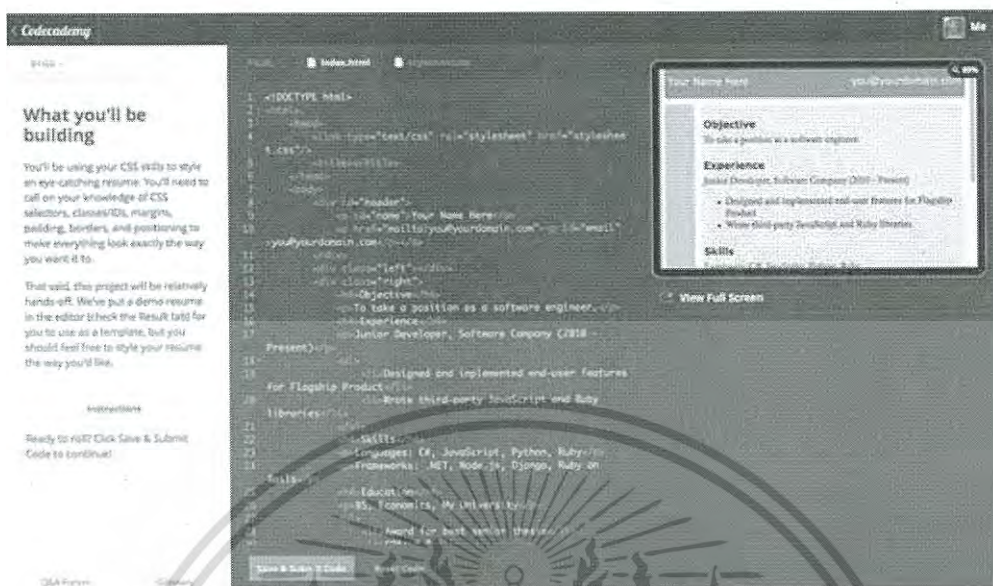
3.1.1.1 Codeschool.com ตัวอย่างหน้าเว็บไซต์ดังรูปที่ 3.1



รูปที่ 3.1 ตัวอย่างหน้าเว็บไซต์ Codeschool.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.2 Codecademy.com ตัวอย่างหน้าเว็บไซต์ดังรูปที่ 3.2



รูปที่ 3.2 ตัวอย่างหน้าเว็บไซต์ Codecademy.com

รูปแบบของ Codeschool และ Codecademy จะเป็นเว็บไซต์ที่มีลักษณะเด่น ดังนี้

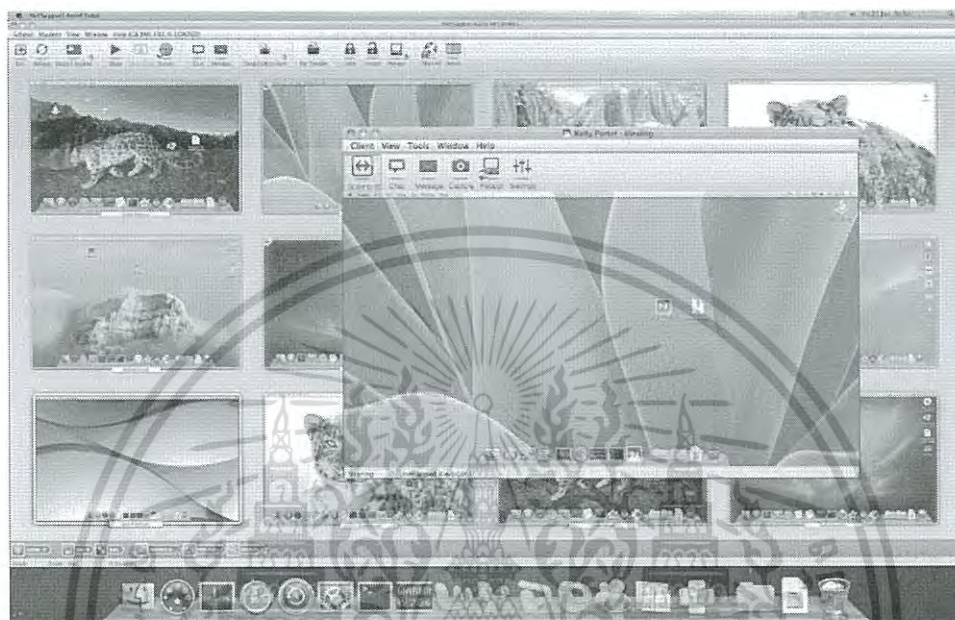
1. แต่ละคอร์สเรียน จะแบ่งออกเป็นบทย่อยๆ และแต่ละบทย่อยๆ จะแบ่งออกเป็นขั้นตอนให้ ผู้เรียนทำตาม เช่น ขั้นตอนแรกของบท 1 ให้ผู้เรียนศึกษาจากคลิป ขั้นตอนที่สองของบท 1 ให้ผู้เรียนทำการตอบคำถาม ขั้นตอนที่สามของบท ให้ผู้เรียนทดลองเขียน โปรแกรมตาม โจทย์กำหนด โดยแต่ละบท จะมีขั้นตอนไม่เหมือนกัน ขึ้นอยู่กับความเหมาะสมของบท นั้นๆ
2. มีระบบเส้นทาง (Path) ที่จะสามารถแนะนำผู้เรียนได้ว่า ถ้าจะไปเส้นทางนี้ ควรจะเรียน คอร์สใดบ้าง
3. มีเว็บบอร์ดสำหรับให้ผู้ใช้ได้ถามคำถาม หรือเสนอข้อเสนอแนะต่างๆ ได้
4. มีระบบ Screencasts ที่ทางผู้ดูแลระบบจะทำการอัดวิดีโอเพื่อที่มีเนื้อหาเกี่ยวกับการเขียน โปรแกรม หรือเครื่องมือการเขียน โปรแกรม ต่างๆ มากมาย ให้ผู้เรียนสามารถเลือกดูได้ อย่างอิสระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 ระบบที่มีหลักการทำงานใกล้เคียงกัน

จากการค้นคว้า พบระบบที่มีหลักการทำงานใกล้เคียงกันกับระบบที่ต้องการจะทำ ซึ่งขอ ยกตัวอย่างมา 2 ระบบ ดังนี้

3.1.1.1 Netsupport ดูตัวอย่างของระบบดังรูปที่ 3.3



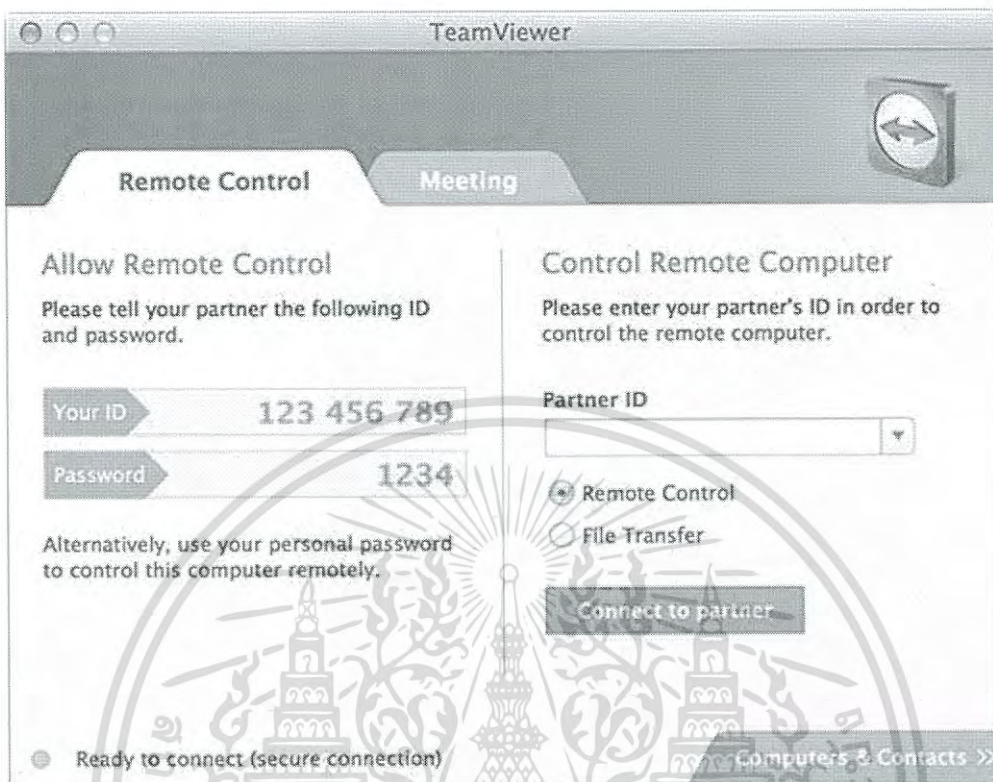
รูปที่ 3.3 ตัวอย่างระบบ Netsupport

รูปแบบของ Netsupport จะมีลักษณะดังนี้

1. ลง Netsupport ในเครื่องที่ต้องการใช้งาน
2. หลังจากติดตั้งจะมีหนึ่งเครื่องที่ทำงานเป็นเครื่องแม่ โดยจะสามารถสั่งให้เครื่องลูกทำอะไรก็ได้ตามที่ต้องการ ไม่ว่าจะเป็น สั่งให้รีสตาร์ทเครื่อง สั่งปิดเครื่อง สั่งล็อกหน้าจอ เป็นต้น
3. เครื่องแม่สามารถควบคุมการเคลื่อนไหวของเครื่องลูกได้อย่างอิสระอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.2 Teamviewer ดูตัวอย่างของระบบดังรูปที่ 3.4



รูปที่ 3.4 ตัวอย่างระบบ Teamviewer

รูปแบบของ Teamviewer จะมีลักษณะดังนี้

1. ลง Teamviewer ในเครื่องที่ต้องการใช้งาน
2. จะมีเครื่องผู้ควบคุม และเครื่องเป้าหมาย หากเครื่องเป้าหมายยอมให้ควบคุม จะทำให้เครื่องผู้ควบคุมสามารถเข้ามาควบคุมได้ โดยจะสั่งการได้ทุกอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ปัญหาที่พบในระบบปัจจุบัน

3.2.1 ระบบเกี่ยวกับการเรียนการเขียนโปรแกรม

จากการลองใช้งาน Codeschool และ Codecademy ทำให้เห็นปัญหาของเว็บไซต์นี้ได้ดังนี้

1. ระบบรองรับเฉพาะภาษาอังกฤษ รวมไปถึงเนื้อหาของคอร์สทั้งหมดจะมีเนื้อหาเป็นภาษาอังกฤษ ซึ่งอาจจะเป็นปัญหาสำหรับคนที่อยากจะเรียนรู้การเขียนโปรแกรม แต่ไม่มีทักษะด้านภาษาอังกฤษ
2. เนื้อหาของเว็บไซต์จะมาจากทางผู้ดูแลระบบเท่านั้น บุคคลอื่นที่ต้องการจะถ่ายทอดความรู้ จะไม่สามารถใช้ประโยชน์จากระบบนี้ได้ อีกทั้งเนื้อหายังมีไม่หลากหลาย
3. ไม่มีระบบสำหรับการอำนวยความสะดวกให้ผู้สอน ถ้าหากผู้สอนต้องการสอนในลักษณะเป็นกลุ่ม เช่น ลักษณะการสอนในห้องเรียน ระบบนี้ไม่สามารถรองรับได้

3.2.2 ระบบที่มีการทำงานใกล้เคียงกัน

3.2.2.1 จากการลองใช้งาน Teamviewer ทำให้เห็นปัญหาและข้อจำกัดดังนี้

1. สามารถควบคุมได้ที่ละเครื่อง แต่ถ้าต้องการจะเปลี่ยนเครื่องควบคุมจะใช้เวลามาก เนื่องจาก Teamviewer ไม่ได้ออกแบบมาใช้สำหรับควบคุมหลายๆเครื่อง
2. ใช้ทรัพยากรเครื่องเยอะ เนื่องจากออกแบบมาให้ควบคุมทุกอย่างได้ จึงเหมาะสำหรับแก้ปัญหาสั้นๆมากกว่า ช่วยกันทำงาน

3.2.2.2 จากการศึกษาการทำงานของ Netsupport ทำให้เห็นปัญหาและข้อจำกัดดังนี้

1. จำเป็นต้องลง Software ต่างๆทีละเครื่อง ไม่ว่าจะเป็นตัว Netsupport เอง หรือตัว Software ที่ต้องใช้ในการเรียนการสอน เช่น Compiler และ Database เป็นต้น
2. มีค่า License ที่ค่อนข้างแพง เหมาะสำหรับองค์กรใหญ่ๆเท่านั้น

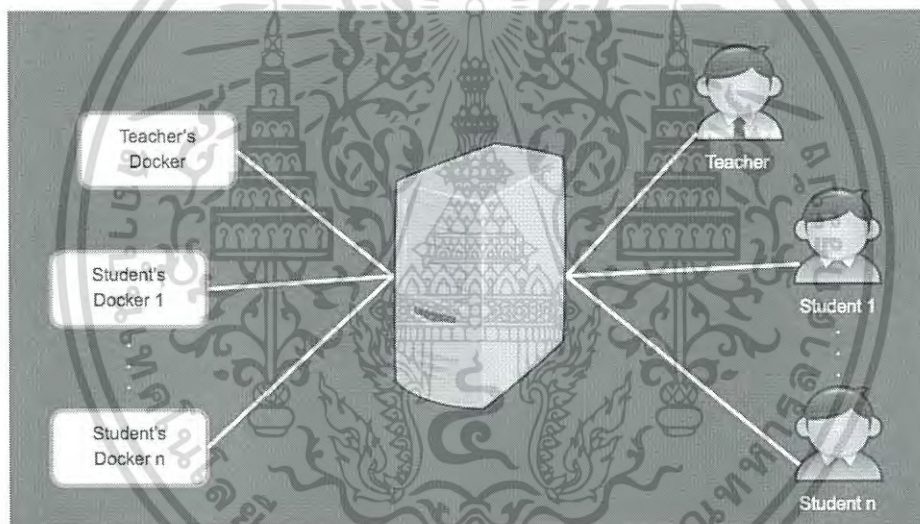
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 แนวคิดในการพัฒนา

3.3.1 การนำ Docker มาใช้

การใช้งาน Docker เพื่อเอื้ออำนวยความสะดวกต่างๆ ที่จำเป็นต้องใช้ในการเขียนโปรแกรมให้กับผู้เรียนและผู้สอนนั้น ทุกคนในห้องเรียนจะมี Docker เป็นของตัวเอง โดยที่ Docker นั้นจะทำงานอยู่ที่ฝั่ง Server และสามารถสั่งงาน Docker ของตนเองให้ทำงานได้ตามที่ต้องการผ่าน Terminal ที่อยู่หน้าจอของตนเอง

โดยที่ Terminal ที่เห็นนั้น เป็นการจำลองผลลัพธ์ขึ้นมาที่หน้าจอ ไม่ใช่ Terminal จริงๆ โดยที่หลักการคือ เมื่อเจ้าของ Docker ต้องการจะสั่งให้ Docker ของตนทำอะไร เจ้าของก็จะส่งคำสั่งนั้นไปยังที่ Server และให้ Server ส่งต่อไปที่ Docker เพื่อให้ Docker ประมวลผลคำสั่งนั้น และ Docker จะส่งผลลัพธ์ของคำสั่งนั้นกลับไปยัง Server เมื่อ Server ได้รับความผลลัพธ์ ก็จะส่งผลลัพธ์นั้นกลับไปที่เจ้าของ ดังรูปที่ 3.5



รูปที่ 3.5 ภาพการทำงานของ Docker

จากรูปที่ 3.5 การทำงานของ Docker แบบนี้ ทำให้เหมือนผู้ใช้สามารถสั่งงาน Docker ได้ผ่านหน้าจอตนเองจริงๆ โดยการส่งข้อความทั้งหมดนี้ จะส่งผ่าน Websocket เนื่องจากจำเป็นต้องให้การตอบสนองเป็นแบบ Real time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การนำ Mongo DB มาใช้

ในส่วนของฐานข้อมูลที่ใช้ในระบบนี้นั้น คือ MongoDB ซึ่งเป็นฐานข้อมูลแบบ NoSQL โดยที่ฐานข้อมูลแบบ NoSQL จะมีข้อดีกว่าฐานข้อมูลแบบ Relational Database ในด้านความเร็วในการเขียนข้อมูลลงฐานข้อมูล ซึ่งถือเป็นส่วนสำคัญสำหรับโปรแกรมที่ต้องการจะเก็บข้อมูลบ่อยๆ เพื่อนำข้อมูลเหล่านั้นไปวิเคราะห์

ในส่วนของ MongoDB จะทำการเก็บข้อมูลเป็นแบบเชิงเอกสาร (Document-Oriented) ซึ่งมีข้อดีคือ รูปแบบของข้อมูลแต่ละชุด ไม่จำเป็นที่จะต้องเหมือนกัน สามารถเพิ่มความซับซ้อนให้ข้อมูลแต่ละตัวได้ตามความเหมาะสม

ข้อดีที่กล่าวมาข้างต้น ถือเป็นข้อได้เปรียบสำคัญที่ทำให้ต้องเลือกใช้ฐานข้อมูลนี้ แต่นอกจากข้อดีที่กล่าวมาข้างต้น ยังมีข้อดีอีกหลายอย่าง เช่น การเพิ่มขนาดของฐานข้อมูลที่สามารถทำได้อย่างรวดเร็ว การสำรองข้อมูล (Backup) ที่ทำได้โดยง่าย การแก้ไขข้อมูลได้อย่างรวดเร็วซึ่งถ้าเทียบกับการแก้ไขข้อมูลของ Relational Database จะถือว่ามีความเร็ว และความต่อเนื่องกว่ามาก เป็นต้น

โดยในการพัฒนาโครงการจัดการห้องเรียนสำหรับการเขียนโปรแกรมอย่างมีประสิทธิภาพได้นำ MongoDB มาเก็บข้อมูลตาม Schema ดังรูปที่ 3.6 ถึง 3.11

```

1 |
2 chat_log_schema = mongoose.Schema
3   classroom name: type: String
4   message: type: String
5   sender: type: String, default: null
6   timestamp: type: String, default: moment()
7   is_teacher: type: Boolean
8

```

รูปที่ 3.6 ภาพ chat_log_schema

```

7
8 classroom_schema = mongoose.Schema
9   raw_name: type: String
10  key: type: String
11  name: type: String
12  max_student: type: Number
13  student_count: type: Number, default: 0
14  students: type: Array, default: []
15  teacher: type: mongoose.Schema.Types.Mixed
16  teacher_name: String
17  status: type: String, default: 'allowed'
18

```

รูปที่ 3.7 ภาพ classroom_schema

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

5
6 container_schema = mongoose.Schema
7   container_id: type: String
8   classroom_id: type: String
9   text: type: String, default: null
10  owner: type: String, default: null
11  status: type: String, default: 'running'
12  term: type: mongoose.Schema.Types.Mixed
13

```

รูปที่ 3.8 ภาพ container_schema

```

2
3 quiz_schema = mongoose.Schema
4   classroom_name: type: String
5   quiz_name: type: String
6   item_count: type: String
7   students: type: mongoose.Schema.Types.Mixed, default: {}
8   items: type: Object, default: {}
9   timestamp: type: String, default: moment()
10  corrected_choice: type: Array
11  time: type: Number
12

```

รูปที่ 3.9 ภาพ quiz_schema

```

1
2 quiz_log_schema = mongoose.Schema
3   classroom_name: type: String
4   quiz_name: type: String
5   student: type: String
6   selected_choice: type: String
7   item: type: String
8   timestamp: type: String, default: moment()
9   is_correct: type: Boolean, default: false
10

```

รูปที่ 3.10 ภาพ quiz_log_schema

```

1
2 user_schema = mongoose.Schema
3   username: type: String
4   password: type: String
5   status: type: String, default: 'login'
6   role: type: String, default: 'student'
7   in_classroom: type: String, default: null
8

```

รูปที่ 3.11 ภาพ user_schema

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การวิเคราะห์ความต้องการ (System requirement analysis)

3.4.1 ความต้องการที่เป็นหน้าที่หลักของระบบ (Functional Requirement)

1. มีระบบสมาชิกสำหรับบุคคลที่ต้องการจะเรียน และบุคคลที่ต้องการจะสอน
2. ระบบ Coder-Space สามารถให้ผู้สอนทำการสอนผ่านระบบนี้ได้ เปรียบเสมือนห้องเรียนออนไลน์
3. ระบบ Coder-Space สามารถให้ผู้สอนทำการสอนผ่านระบบนี้ได้เปรียบเสมือนห้องเรียนออนไลน์
4. ระบบ Coder-Space ผู้สอนสามารถเขียนโค้ดของตัวเองผ่านระบบนี้ได้
5. ระบบ Coder-Space ผู้สอนสามารถสามารถดู และแก้ไขโค้ดของผู้เรียนได้
6. ระบบ Coder-Space ผู้เรียนสามารถเขียนโค้ดของตัวเองผ่านระบบนี้ได้
7. ระบบ Coder-Space ผู้เรียนสามารถดู โค้ดของผู้สอน ได้ก็ต่อเมื่อผู้สอนเปิดการเข้าถึงให้ดูได้เท่านั้น
8. ระบบ Coder-Space ผู้สอนและผู้เรียนสามารถประมวลผลโค้ดผ่านระบบนี้ได้
9. สามารถส่งข้อความติดต่อถึงกันได้
10. ผู้สอนสามารถสร้างแบบทดสอบประเภทตอบคำถามและประเภท Grader ได้
11. ผู้เรียนสามารถทำแบบทดสอบประเภทตอบคำถามและประเภท Grader ได้
12. ผู้สอนสามารถดูผลคะแนนการทำแบบทดสอบได้

3.4.2 ความต้องการที่ไม่ใช่หน้าที่หลักของระบบ (Non-Functional Requirement)

1. รองรับส่วนเพิ่มเติมขยายของระบบได้
2. ระบบจะต้องมีส่วนติดต่อของผู้ใช้ที่เข้าใจได้ง่าย

3.5 วิเคราะห์และวิจารณ์ระบบที่ต้องการออกแบบ

3.5.1 แผนภาพยูสเคส (Use case Diagram)

เป็นแผนภาพที่แสดงความสัมพันธ์ระหว่าง Use case และ Actor เพื่อบอกว่าระบบใหม่มีกิจกรรมอะไรบ้าง และใครเป็นผู้เกี่ยวข้อง และเกี่ยวข้องอย่างไรกับกิจกรรมในระบบนี้

3.5.1.1 ผู้เกี่ยวข้องในระบบ (Actor) ประกอบด้วย

1. ผู้สอน
2. ผู้เรียน

3.5.1.2 ฟังก์ชันการทำงานหลัก

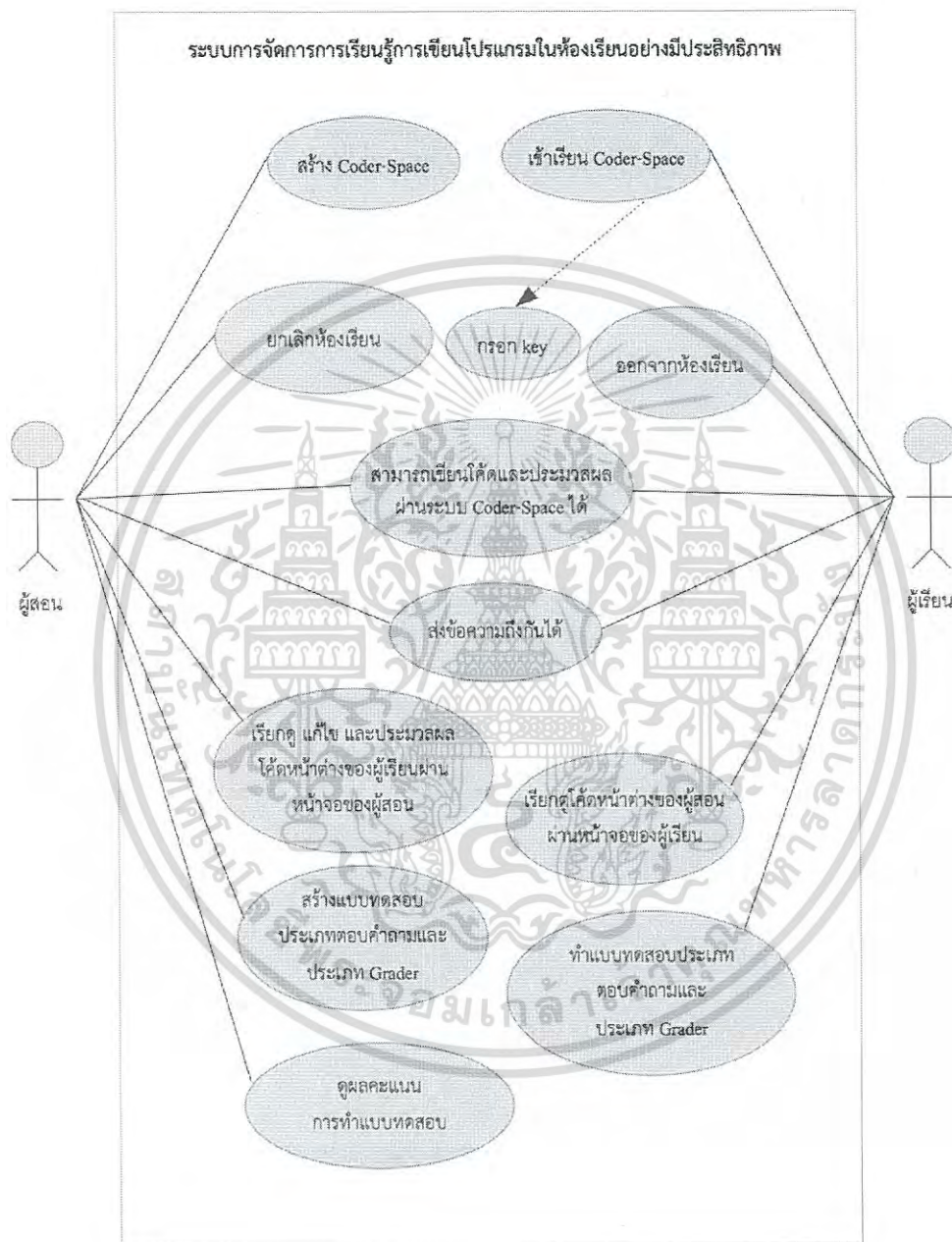
1. สร้างห้องเรียน Coder-Space
2. เข้าห้องเรียน Coder-Space
3. ยกเลิกห้องเรียน Coder-Space
4. ออกจากห้องเรียน Coder-Space
5. เขียนโค้ดและประมวลผลโค้ดผ่านระบบ Coder-Space
6. ผู้สอนดู แก้ไข และประมวลผลโค้ดหน้าต่างของผู้เรียนผ่านหน้าจอผู้สอนได้
7. ผู้เรียนดูโค้ดในหน้าต่างของผู้สอนผ่านหน้าจอผู้เรียนได้
8. ส่งข้อความถึงกัน
9. สร้างแบบทดสอบประเภทตอบคำถามและประเภท Grader
10. ทำแบบทดสอบประเภทตอบคำถามและประเภท Grader
11. ดูผลคะแนนการทำแบบทดสอบ

146243

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1.3 แผนภาพยูสเคส (Use Case Diagram)

แผนภาพ Use Case Diagram ของระบบการจัดการการเรียนรู้การเขียนโปรแกรมในห้องเรียนอย่างมีประสิทธิภาพ ในห้องเรียนอย่างมีประสิทธิภาพ ดังรูปที่ 3.12



รูปที่ 3.12 แผนภาพยูสเคส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1.4 รายละเอียดการทำงานของแต่ละยูสเคส (Use case description)

ตารางที่ 3.1 คำอธิบายยูสเคสสร้างห้องเรียน Coder-Space

Use Case Name	สร้างห้องเรียน Coder-Space
Actor	ผู้สอน
Description	สร้างห้องเรียนจำลองที่จะช่วยให้การเรียนการสอนเป็นไปได้อย่างขึ้น เช่น ผู้สอนสามารถดูได้ว่าผู้เรียนกำลังพิมพ์อะไรอยู่ในหน้าต่าง โค้ด และสามารถแก้ไขได้หากต้องการ
Pre-conditions	1. ล็อกอินเข้าใช้ระบบ
Post conditions	ใส่รายละเอียดของ Coder-Space ให้ครบ
Flow of Events	<ol style="list-style-type: none"> 1. ล็อกอินเข้าสู่ระบบ 2. เลือกสร้างห้องเรียน Coder-Space 3. เลือกภาษาเขียนโปรแกรมที่ต้องการสอน 4. กำหนดจำนวนผู้เรียนที่เข้าเรียนในคลาสนี้ 5. กดปุ่มยืนยันสร้างห้องเรียน Coder-Space 6. นำรหัส(Key) ไปแจกจ่ายให้กับคนที่ต้องการจะเรียน
Alternative Flow	<p>A1 Actor ต้องการจะลงโปรแกรมนเอง</p> <p>A2 ทำการอัปโหลด Image Docker</p>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 คำอธิบายยูสเคสเข้าห้องเรียน Coder-Space

Use Case Name	เข้าห้องเรียน Coder-Space
Actor	ผู้เรียน
Description	ทำการเรียนผ่าน Coder-Space โดยเลือกห้องเรียนที่ต้องการและกรอกรหัส(Key) เพื่อเข้าห้องห้อง
Pre-conditions	1. ล็อกอินเข้าใช้ระบบ 2. มีรหัสการเข้าห้อง ซึ่งจะได้จากผู้สอน
Post conditions	-
Flow of Events	1. ล็อกอินเข้าสู่ระบบ 2. ทำการเลือกห้องเรียนCoder-Space ที่ต้องการเรียน 3. ใส่อรหัส(Key) เพื่อเข้าห้องเรียนซึ่งได้จากผู้สอน
Alternative Flow	A1 Actor ใ้ Key ผิด A2 ไม่สามารถเข้าเรียนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 คำอธิบายยูสเคสยกเลิกห้องเรียน Coder-Space

Use Case Name	ยกเลิกห้องเรียน Coder-Space
Actor	ผู้สอน
Description	ยกเลิกห้องเรียน Coder-Space ได้ เมื่อยกเลิกแล้วผู้เรียนทุกคนในห้องเรียนนั้น จะออกจากระบบห้องนั้นทันทีเช่นกัน
Pre-conditions	1. ล็อกอินเข้าใช้ระบบ 2. สร้างห้องเรียน Coder-Space
Post conditions	-
Flow of Events	1. ล็อกอินเข้าสู่ระบบ 2. สร้างห้องเรียน Coder-Space 3. ยกเลิกห้องเรียน Coder-Space
Alternative Flow	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 คำอธิบายยูสเคสออกจากห้องเรียน Coder-Space

Use Case Name	ออกจากห้องเรียนCoder-Space
Actor	ผู้เรียน
Description	ออกจากห้องเรียน Coder-Space จะไม่สามารถเข้าห้องเรียนนั้นได้อีกจนกว่าห้องเรียนห้องนั้นจะถูกยกเลิกไปโดยผู้สอน
Pre conditions	1. ล็อกอินเข้าใช้ระบบ 2. มีรหัสการเข้าห้อง ซึ่งจะได้จากผู้สอน
Post conditions	-
Flow of Events	1. ล็อกอินเข้าสู่ระบบ 2. ทำการเลือกห้องเรียน Coder-Space ที่ต้องการเรียน 3. ใส่รหัส(Key) เพื่อเข้าห้องเรียนซึ่งได้จากผู้สอน 4. ออกจากห้องเรียน Coder-Space
Alternative Flow	A1 ผู้สอนทำการยกเลิกห้องเรียน Coder-Space

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 คำอธิบายยูสเคสเรียกดู แก้ไขและประมวลผลโค้ดในหน้าต่างของผู้เรียนผ่านหน้าจอของผู้สอน

Use Case Name	เรียกดูแก้ไขและประมวลผลโค้ดในหน้าต่างของผู้เรียนผ่านหน้าจอของผู้สอน
Actor	ผู้สอน
Description	สามารถดูได้ว่าผู้เรียนคนไหนกำลังพิมพ์อะไรอยู่ในหน้าต่างโค้ดและสามารถทำการแก้ไข โค้ดและประมวลผลโค้ดในหน้าต่างของผู้เรียนคนนั้นได้ผ่านหน้าจอของผู้สอน
Pre-conditions	<ol style="list-style-type: none"> 1. ล็อกอินเข้าใช้ระบบ 2. สร้างห้องเรียน Coder-Space 3. มีผู้เรียนเข้ามาเรียนในห้องเรียน Coder-Space ของตัวเอง
Post conditions	-
Flow of Events	<ol style="list-style-type: none"> 1. ล็อกอินเข้าสู่ระบบ 2. สร้างห้องเรียน Coder-Space 3. ดูว่าในห้องเรียน Coder-Space นี้มีผู้เรียนคนไหนบ้าง 4. เข้าไปดูในหน้าต่างของผู้เรียนที่สนใจ 5. แก้ไขโค้ดของผู้เรียนและสามารถประมวลผลโค้ดได้ถ้าต้องการ
Alternative Flow	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 คำอธิบายยูสเคสเขียนโค้ดและประมวลผลโค้ดผ่านระบบ Coder-Space

Use Case Name	เขียนโค้ดและประมวลผลโค้ดผ่านระบบ Coder-Space
Actor	ผู้เรียนและผู้สอน
Description	เขียนโค้ดและทำการส่งโค้ดที่พิมพ์ในหน้าต่างโค้ดไปที่เซิร์ฟเวอร์เพื่อนำผลลัพธ์ออกมาแสดง
Pre-conditions	1. ล็อกอินเข้าใช้ระบบ 2. ได้เรียนหรือสอนในห้องเรียนCoder-Space ใดๆ
Post conditions	-
Flow of Events	1. ล็อกอินเข้าสู่ระบบ 2. ทำการเข้าเรียน Coder-Space สำหรับผู้เรียนหรือสร้างห้องเรียน Coder-Space สำหรับผู้สอน 3. ทำการพิมพ์ในโค้ดในส่วนของหน้าต่างของตนเอง สำหรับผู้เรียนหรือ เข้าไปที่หน้าต่างโค้ดของผู้เรียนสำหรับผู้สอน 4. กดเลือกประมวลผลเพื่อส่งโค้ดไปประมวลผลที่เซิร์ฟเวอร์
Alternative Flow	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.7 คำอธิบายยูสเคสเรียกดูโค้ดในหน้าต่างของผู้สอนผ่านหน้าจอของผู้เรียน

Use Case Name	เรียกดูโค้ดในหน้าต่างของผู้สอน
Actor	ผู้เรียน
Description	สามารถดูได้ว่าผู้สอนพิมพ์โค้ดอะไรในหน้าต่างนั้นซึ่งสามารถใช้ในการนำมาเปรียบเทียบกับโค้ดของตัวเองหรือใช้ในการดูประกอบการอธิบายของผู้สอน
Pre-conditions	<ol style="list-style-type: none"> 1. ล็อกอินเข้าใช้ระบบ 2. เข้าเรียน Coder-Space 3. ผู้สอนเปิดให้ผู้เรียนเข้าถึงหน้าต่างของผู้สอน
Post conditions	-
Flow of Events	<ol style="list-style-type: none"> 1. ล็อกอินเข้าสู่ระบบ 2. เข้าห้องเรียน Coder-Space 3. เข้าไปดูโค้ดในหน้าต่างของผู้สอน
Alternative Flow	A1 ผู้สอนปิดการเข้าถึงไม่ให้เข้ามาดูโค้ดในหน้าต่างได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 คำอธิบายยูสเคสส่งข้อความถึงกัน

Use Case Name	ส่งข้อความถึงกัน
Actor	ผู้เรียนและผู้สอน
Description	ส่งข้อความถึงกัน หากมีข้อสงสัยหรือต้องการข้อมูลใดๆ
Pre-conditions	1. ล็อกอินเข้าใช้ระบบ
Post conditions	-
Flow of Events	<ol style="list-style-type: none"> 1. ล็อกอินเข้าสู่ระบบ 2. เลือกบุคคลที่ต้องการจะส่งข้อความถึง 3. ทำการส่งข้อความ
Alternative Flow	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 คำอธิบายยูสเคสสร้างแบบทดสอบประเภทตอบคำถามและประเภท Grader

Use Case Name	สร้างแบบทดสอบประเภทตอบคำถามและประเภท Grader
Actor	ผู้สอน
Description	<p>แบบทดสอบประเภทตอบคำถาม ผู้สอนสามารถสร้างโจทย์และตัวเลือกทั้งหมด 4 ตัวเลือก โดยต้องทำเครื่องหมายว่าตัวเลือกใดที่เป็นคำตอบที่ถูกต้องด้วย</p> <p>แบบทดสอบประเภท grader ซึ่งเป็นระบบ ที่ให้ผู้เรียนเขียนโค้ดและอัปโหลดขึ้นเพื่อให้ระบบของเราทำการตรวจสอบผลลัพธ์อัตโนมัติ โดยจะมี Testcase ที่แตกต่างกันไปตามที่ผู้สอนกำหนด ในการสร้างจำเป็นต้องกรอก Input และ Output สำหรับ Testcase ด้วย</p>
Pre-conditions	1. ล็อกอินเข้าใช้ระบบ
Post conditions	-
Flow of Events	<ol style="list-style-type: none"> 1. ล็อกอินเข้าสู่ระบบ 2. สร้างห้องเรียน Coder-Space 3. มีผู้เรียนเข้ามาเรียนในห้องเรียน Coder-Space 4. กดหน้าสร้างแบบทดสอบ 5. กดปุ่ม Create New Quiz หรือ Create New Problem
Alternative Flow	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.10 คำอธิบายยูสเคส ทำแบบทดสอบประเภทตอบคำถาม

Use Case Name	ทำแบบทดสอบประเภทตอบคำถามและประเภท Grader
Actor	ผู้เรียน
Description	ทำแบบทดสอบประเภทตอบคำถาม ผู้เรียนต้องเลือกคำตอบที่ถูกเพียงคำตอบเดียวจากตัวเลือกทั้งหมด 4 ตัวเลือก ทำแบบทดสอบประเภท Grader โดยเขียนโปรแกรมลงใน Editor แล้ว กดปุ่ม Submit หรือจะอัปโหลดไฟล์จากภายนอก ได้เช่นกัน
Pre-conditions	1. ล็อกอินเข้าใช้ระบบ
Post conditions	-
Flow of Events	1. ล็อกอินเข้าสู่ระบบ 2. เข้าห้องเรียน Coder-Space 3. ทำแบบทดสอบ
Alternative Flow	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.11 คำอธิบายยูสเคส คูผลคะแนนการทำแบบทดสอบ

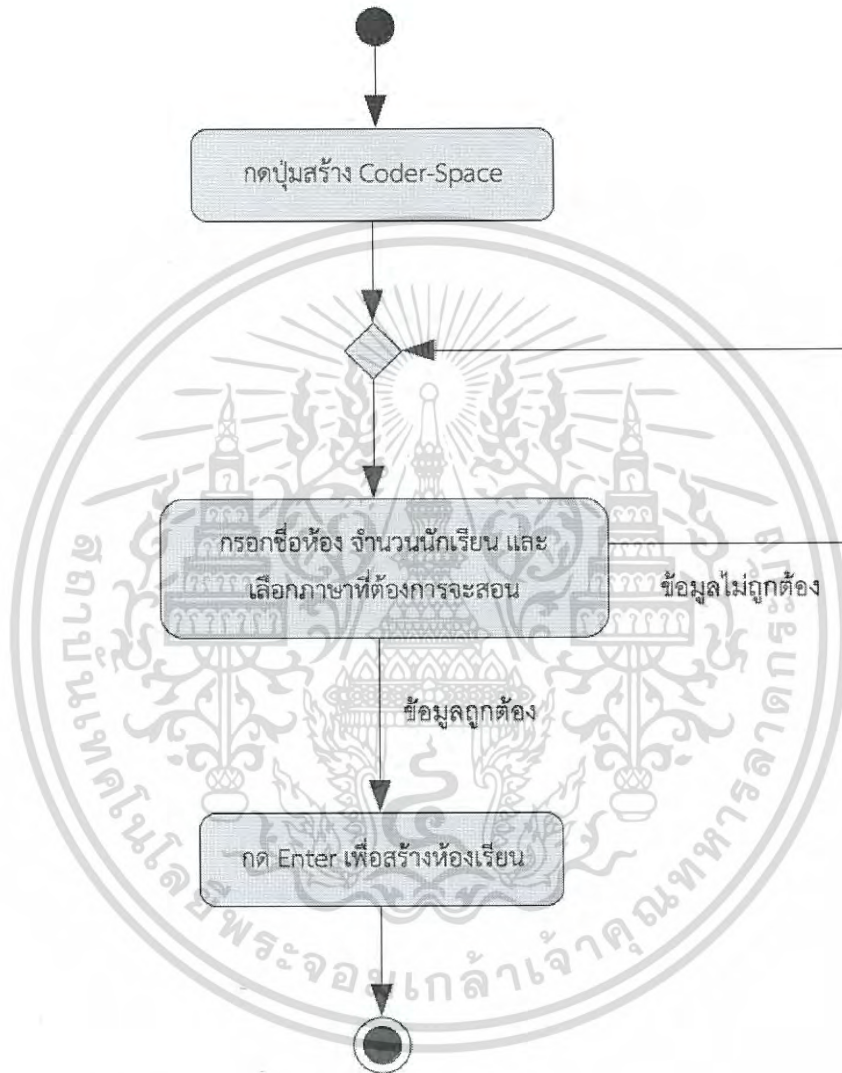
Use Case Name	คูผลคะแนนการทำแบบทดสอบ
Actor	ผู้สอน
Description	ทำการคูผลคะแนนการทำแบบทั้งสอบของนักเรียนทั้งคลาส มีการแสดงผลภาพรวมในรูปแบบของ Graph และผลคะแนนอย่างละเอียดเป็นรายบุคคล
Pre-conditions	1. ล็อกอินเข้าใช้ระบบ
Post conditions	-
Flow of Events	<ol style="list-style-type: none"> 1. ล็อกอินเข้าสู่ระบบ 2. สร้างห้องเรียน Coder-Space 3. สร้างแบบทดสอบ 4. คูผลแบบทดสอบ
Alternative Flow	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2 แผนภาพกิจกรรม (Activity Diagram)

3.5.2.1 แผนภาพกิจกรรมของการสร้าง Coder-Space

แผนภาพ Activity Diagram ของกิจกรรมของการสร้าง Coder-Space แสดงดังรูปที่ 3.13

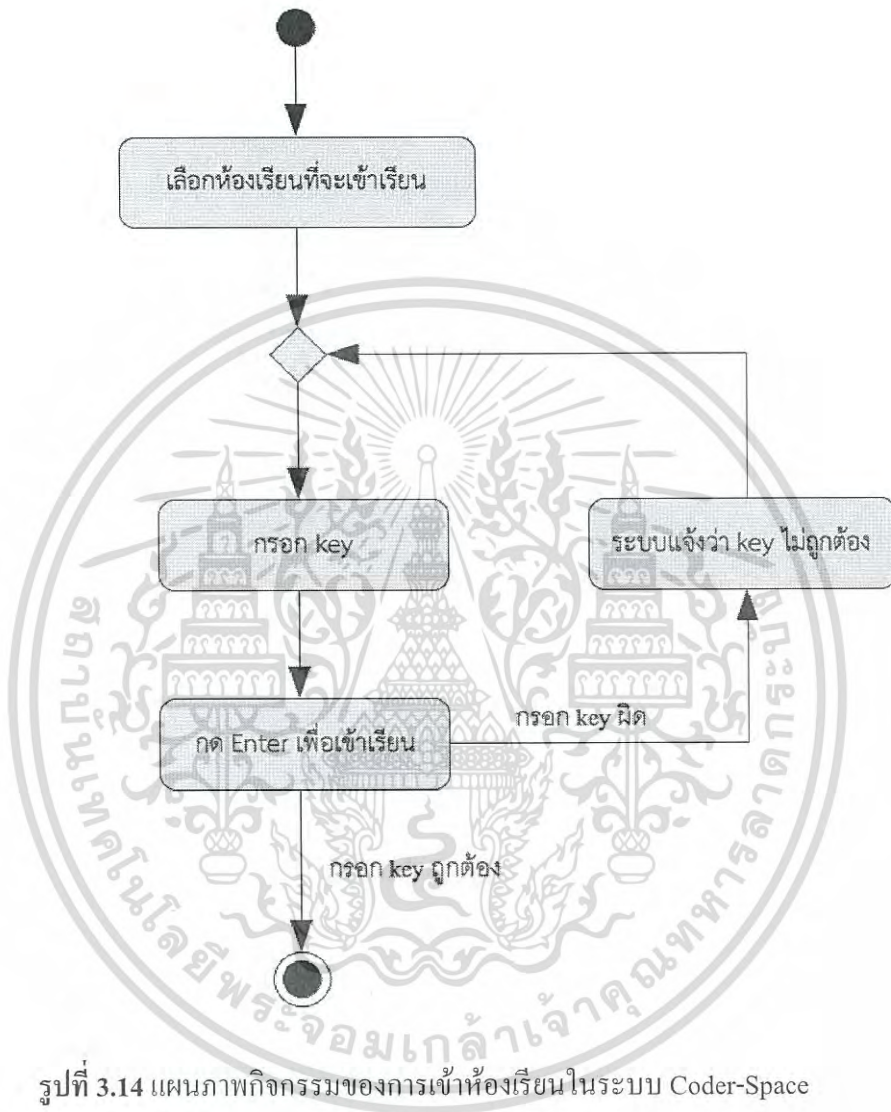


รูปที่ 3.13 แผนภาพกิจกรรมของการสร้างห้องเรียนในระบบ Coder-Space

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2.2 แผนภาพกิจกรรมของการเข้าเรียน Coder-Space

แผนภาพ Activity Diagram ของกิจกรรมของการเข้าเรียน Coder-Space แสดงดังรูปที่ 3.14

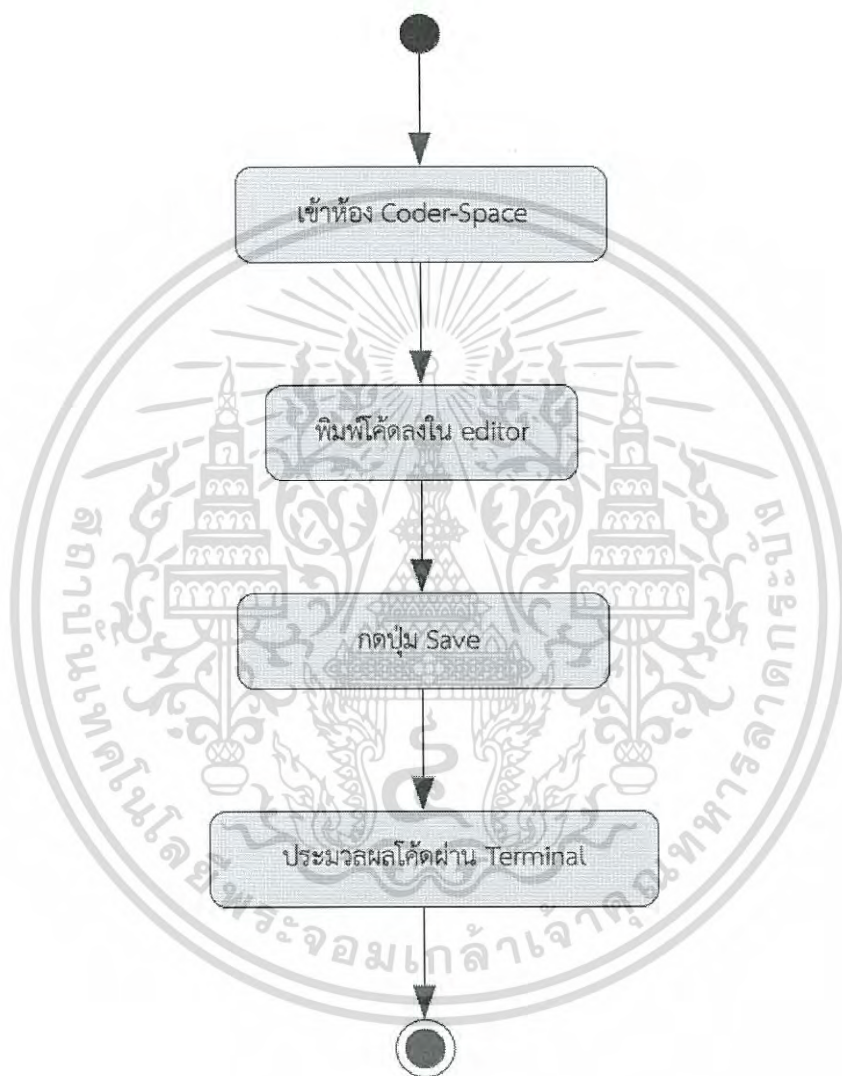


รูปที่ 3.14 แผนภาพกิจกรรมของการเข้าห้องเรียนในระบบ Coder-Space

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2.3 แผนภาพกิจกรรมของการเขียนโค้ดและประมวลผลโค้ดของระบบ Coder-Space

แผนภาพ Activity Diagram ของกิจกรรมของการเขียนโค้ดและประมวลผลโค้ดของระบบ Coder-Space แสดงดังรูปที่ 3.15

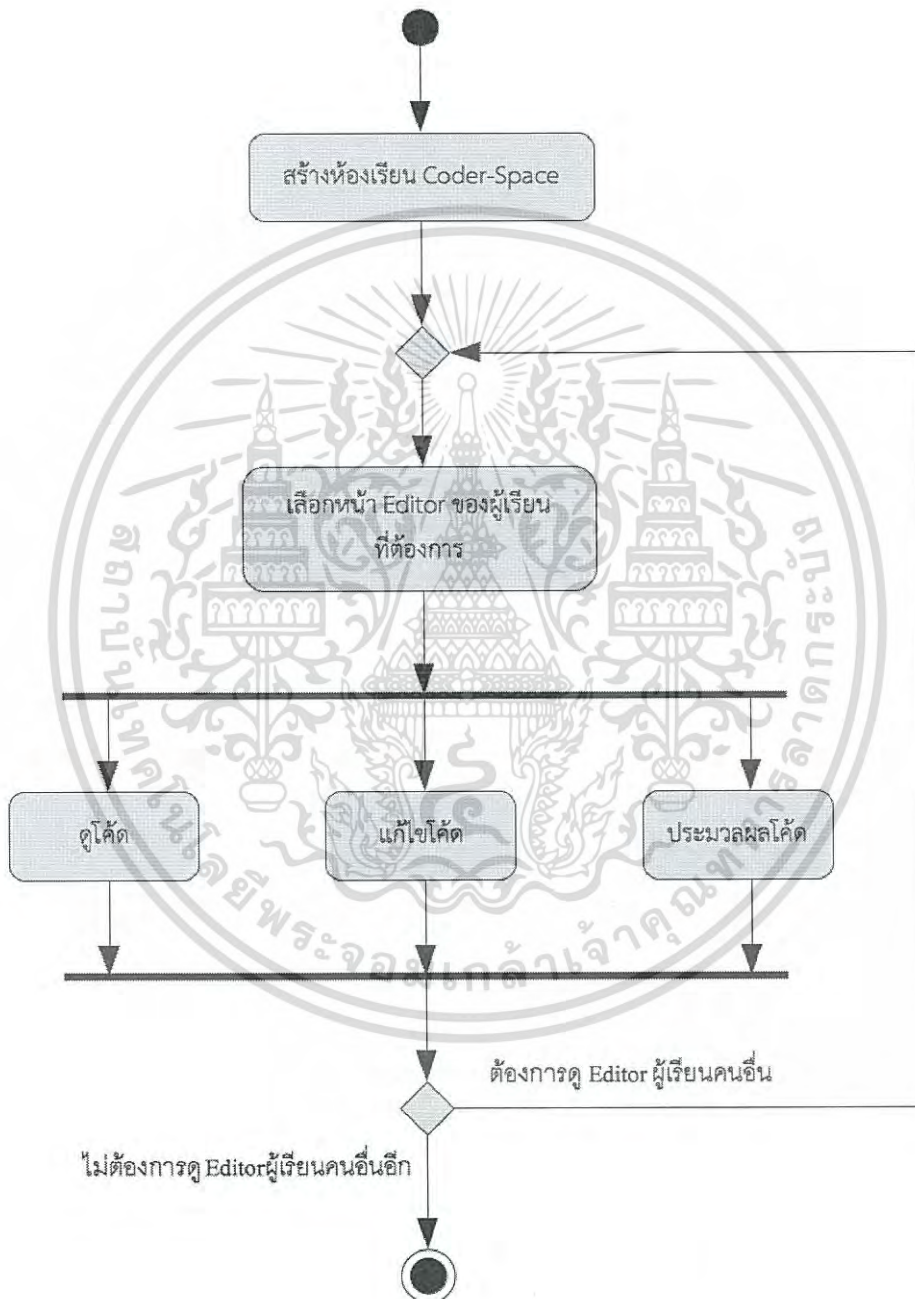


รูปที่ 3.15 แผนภาพกิจกรรมของการเขียนโค้ดและประมวลผลโค้ดของระบบ Coder-Space

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2.4 แผนภาพกิจกรรมของการดู แก้ไข และประมวลผลโค้ดหน้า Editor ของผู้เรียน

แผนภาพ Activity Diagram ของกิจกรรมของการดู แก้ไข และประมวลผลโค้ดหน้า Editor ของผู้เรียนผ่านหน้าจอของผู้สอน แสดงดังรูปที่ 3.16

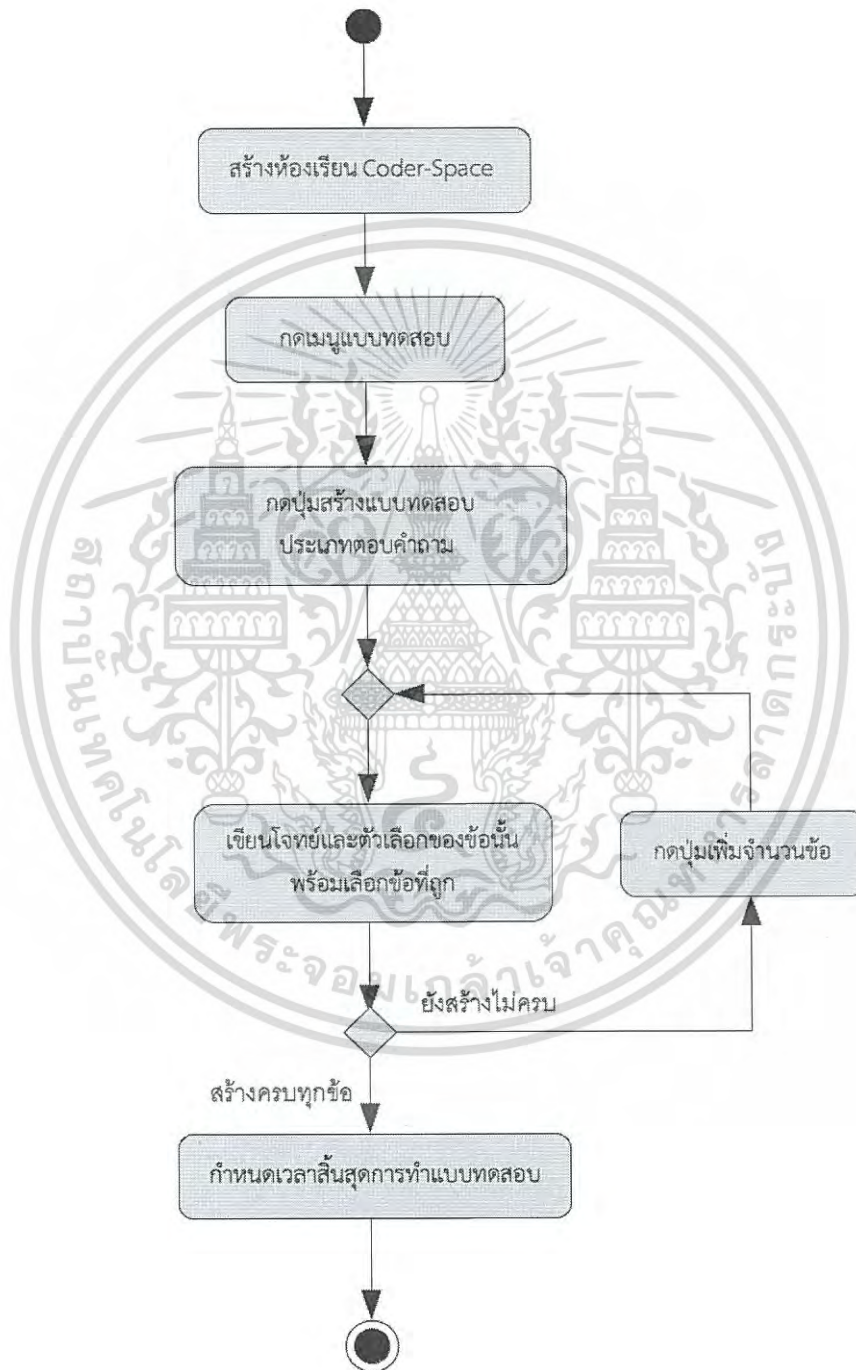


รูปที่ 3.16 แผนภาพกิจกรรมของการดู แก้ไข และประมวลผลโค้ดหน้า Editor ของผู้เรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2.5 แผนภาพกิจกรรมของการสร้างแบบทดสอบประเภทตอบคำถาม

แผนภาพ Activity Diagram ของกิจกรรมของการสร้างแบบทดสอบประเภทตอบคำถาม แสดงดังรูปที่ 3.17

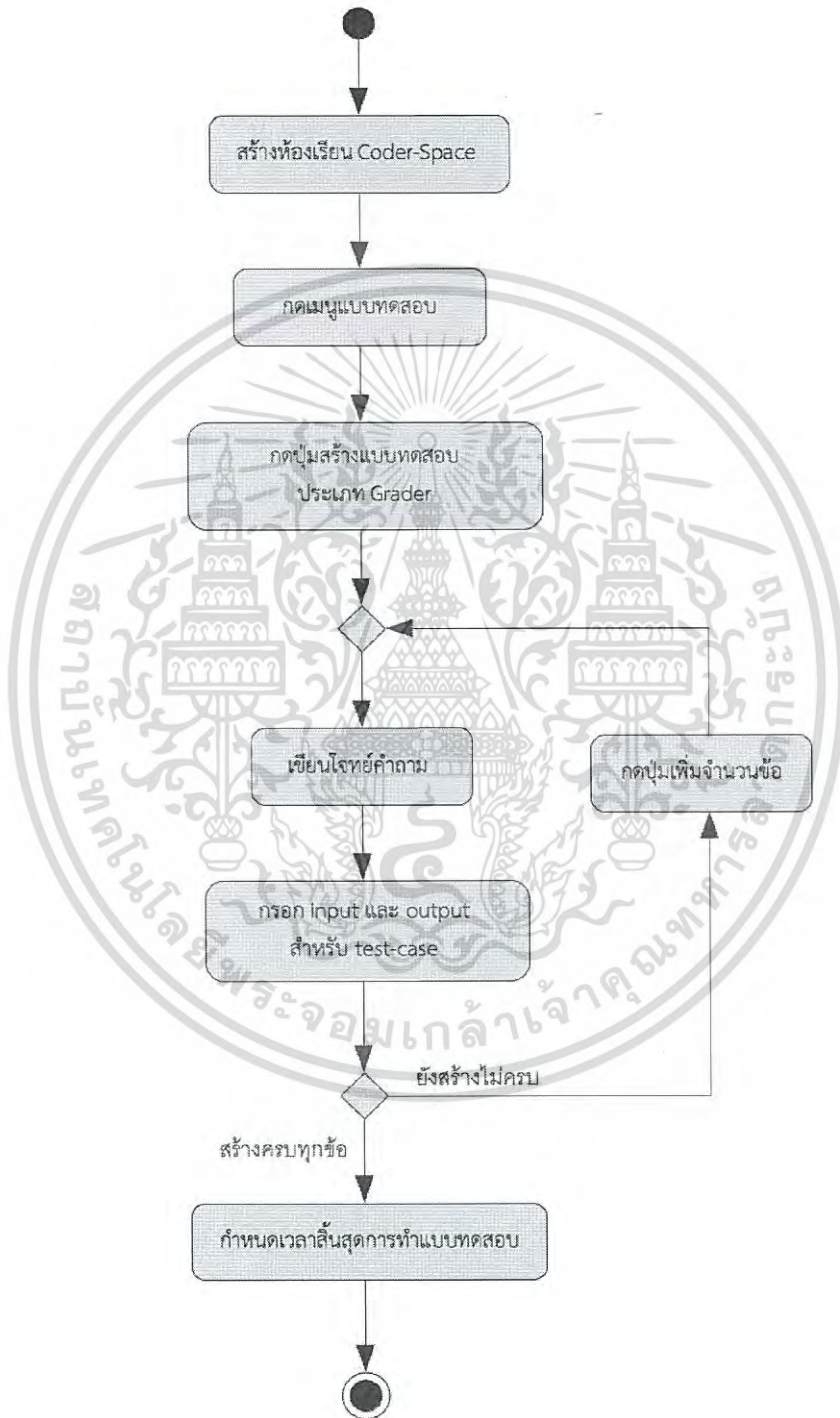


รูปที่ 3.17 แผนภาพกิจกรรมของการสร้างแบบทดสอบประเภทตอบคำถาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2.6 แผนภาพกิจกรรมของการสร้างแบบทดสอบสำหรับระบบ Grader

แผนภาพ Activity Diagram ของกิจกรรมของการสร้างแบบทดสอบประเภท Grader แสดง
 ดังรูปที่ 3.18

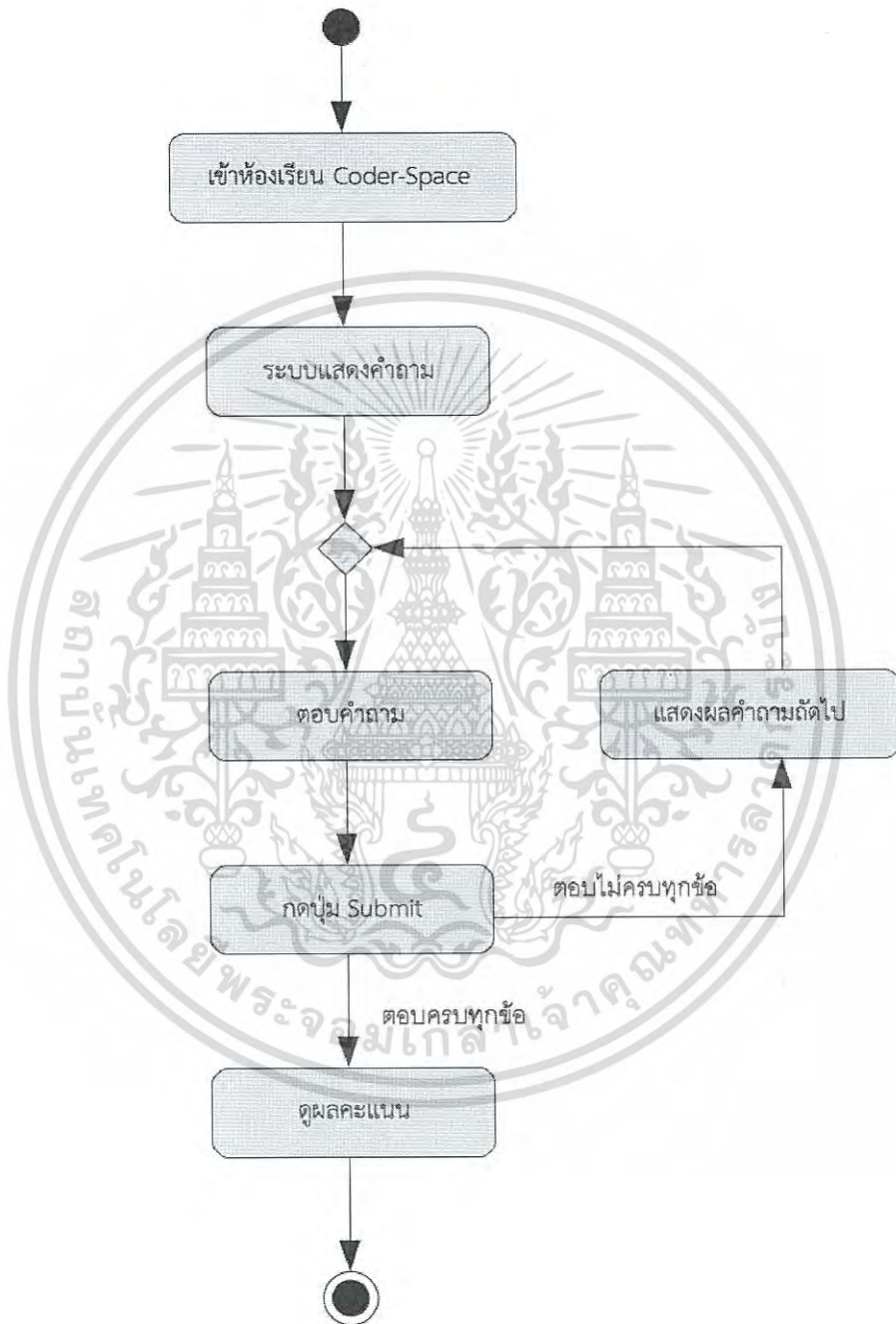


รูปที่ 3.18 แผนภาพกิจกรรมของการสร้างแบบทดสอบสำหรับระบบ Grader

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2.7 แผนภาพกิจกรรมของการทำแบบทดสอบประเภทตอบคำถาม

แผนภาพ Activity Diagram ของกิจกรรมของการทำแบบทดสอบประเภทตอบคำถาม แสดงดังรูปที่ 3.19

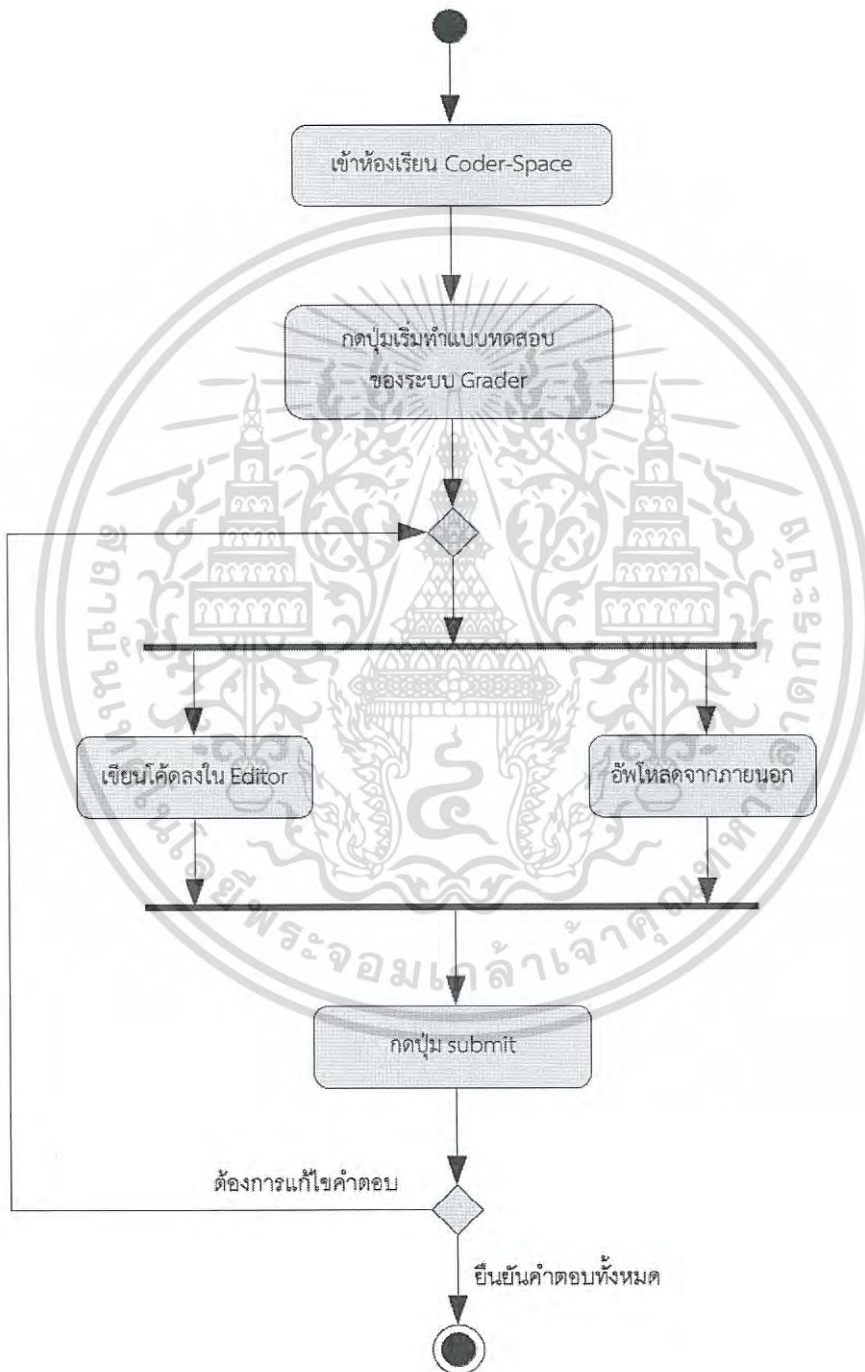


รูปที่ 3.19 แผนภาพกิจกรรมของการทำแบบทดสอบประเภทตอบคำถาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2.8 แผนภาพกิจกรรมของการทำแบบทดสอบสำหรับระบบ Grader

แผนภาพ Activity Diagram ของกิจกรรมของการทำแบบทดสอบประเภท Grader แสดงดังรูปที่ 3.20



รูปที่ 3.20 แผนภาพกิจกรรมของการทำแบบทดสอบสำหรับระบบ Grader

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ระบบต้นแบบ

4.1 ผลการทดลองและการศึกษา

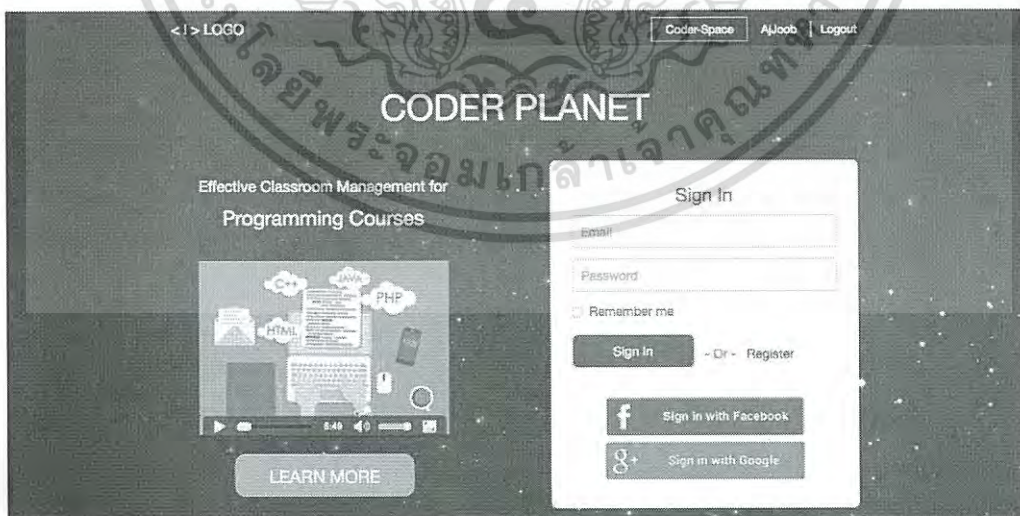
4.1.1 ผลการใช้งาน Docker

จากการนำ Docker มาประยุกต์ใช้กับระบบ เพื่อเอื้ออำนวยทรัพยากรต่างๆที่จำเป็นต้องใช้ในการเขียนโปรแกรมให้กับผู้ใช้ เนื่องจาก Docker เป็นเทคโนโลยีค่อนข้างใหม่ จึงมีปัญหาลึกลับๆ ในหลายๆด้าน มีความไม่เสถียรสูง ทำให้ยังไม่เหมาะกับการนำมาใช้กับระบบจริง แต่ Docker ถือเป็นเทคโนโลยีที่ตอบโจทย์กับนักพัฒนาโปรแกรมบางกลุ่มเป็นอย่างมาก จึงทำให้มีคนช่วยพัฒนา รวมทั้งคนที่คอยแจ้งปัญหาจากการใช้งาน Docker อย่างมากมาย จากสาเหตุข้างต้น ทำให้การพัฒนา Docker เป็นไปอย่างถูกต้องและรวดเร็ว ในไม่ช้า Docker จะถูกนำมาใช้อย่างแพร่หลาย และทำประโยชน์ได้อย่างมากมาย

4.2 เว็บแอปพลิเคชัน (Web Application)

4.2.1 ส่วนของหน้าแรก

หน้านี้ คือหน้าแรกที่ใช้จะให้เห็นรายละเอียดภาพรวม เกี่ยวกับการทำงานของเว็บไซต์ ดังรูปที่ 4.1



รูปที่ 4.1 หน้าแรกของเว็บไซต์ Coder Planet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ส่วนของผู้สอน

4.2.2.1 หน้าสร้างห้องเรียน Coder-Space

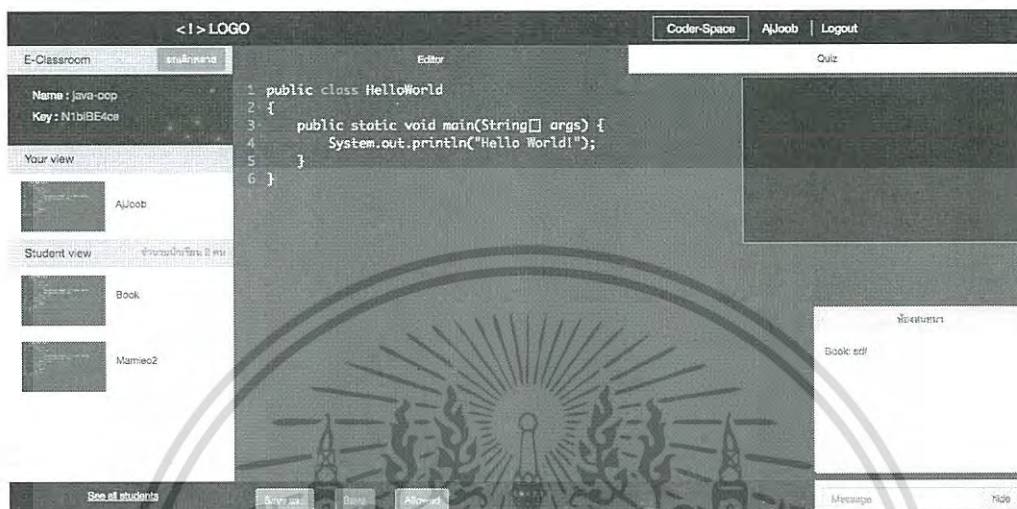
หน้านี้ เป็นหน้าสร้างห้องเรียน Coder-Space ผู้สอนต้องกรอกรายละเอียดดังกล่าวด้วยบนเมื่อคลิกปุ่มสร้างห้องแล้ว ระบบจะทำการสร้างรหัสมาให้ ผู้สอนต้องนำรหัสนี้ไปแจกจ่ายผู้เรียน ผู้เรียนจำเป็นต้องกรอกรหัสนี้ จึงสามารถเข้าเรียนห้อง Coder-Space ที่ผู้สอนสร้างขึ้นมาได้ ดังรูปที่ 4.2

รูปที่ 4.2 หน้าสร้างห้องเรียน Coder-Space

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.2 หน้า Editor ของผู้สอน

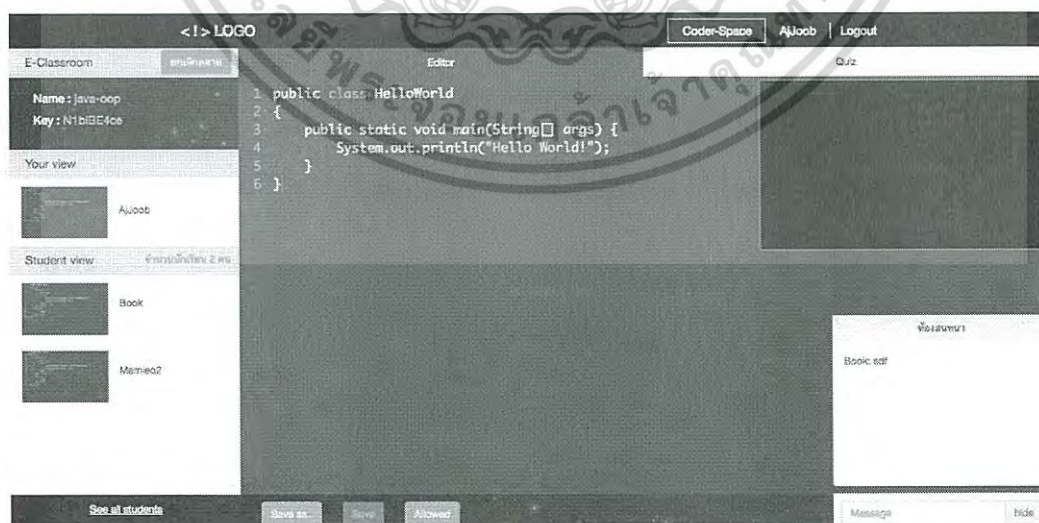
หน้านี้ ผู้สอนสามารถเขียนโค้ดลงไป ใน Editor และสามารถประมวลผลโค้ดผ่านระบบนี้ได้ แถบด้านข้างจะแสดง Editor ของผู้เรียนที่อยู่ในห้องเรียนนี้ทั้งหมด ผู้สอนสามารถกดเข้าไปดูแก้ไข และประมวลผลโค้ดใน Editor ของผู้เรียนคนใดที่สนใจได้ ดังรูปที่ 4.3



รูปที่ 4.3 หน้า Editor ของผู้สอน

4.2.2.3 คูหน้า Editor ของผู้เรียน

หน้านี้ เมื่อผู้สอนเลือก Editor ของผู้เรียนที่ต้องการ ส่วนของ Editor ของผู้สอนจะเปลี่ยนเป็น Editor ของผู้เรียนคนนั้นทันที ผู้สอนสามารถดูแก้ไข และประมวลผลโค้ดใน Editor ของผู้เรียนคนนั้นได้ ดังรูปที่ 4.4

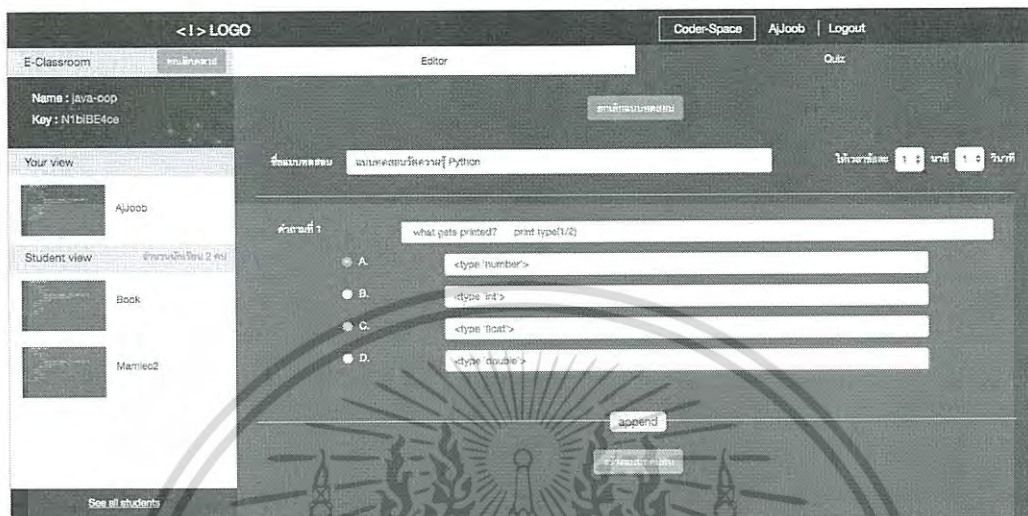


รูปที่ 4.4 คูหน้า Editor ของผู้เรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.4 หน้าสร้างแบบทดสอบสำหรับตอบคำถาม

ผู้สอนต้องสร้างโจทย์สำหรับทำแบบทดสอบและเขียนคำตอบลงตัวเลือกทั้ง 4 ข้อ โดยจำเป็นต้องทำเครื่องหมายลงในตัวเลือกที่เป็นคำตอบที่ถูกต้องด้วย ดังรูปที่ 4.5



รูปที่ 4.5 สร้างแบบทดสอบสำหรับตอบคำถาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.5 หน้าสร้างแบบทดสอบสำหรับระบบ Grader

ผู้สอนต้องสร้างโจทย์สำหรับทำแบบทดสอบ ซึ่งจำเป็นต้องกรอก Input และ Output สำหรับ Test case ข้อนั้นๆ เพื่อให้ผู้เรียนสามารถเขียนโค้ดและระบบจะทำการใส่ Input ที่ผู้สอนกรอกลงในโค้ดของผู้เรียนนั้น และทำการประมวลผลแล้วตรวจสอบว่าได้ผลลัพธ์เป็นไปตาม Output หรือไม่ ดังรูปที่ 4.6

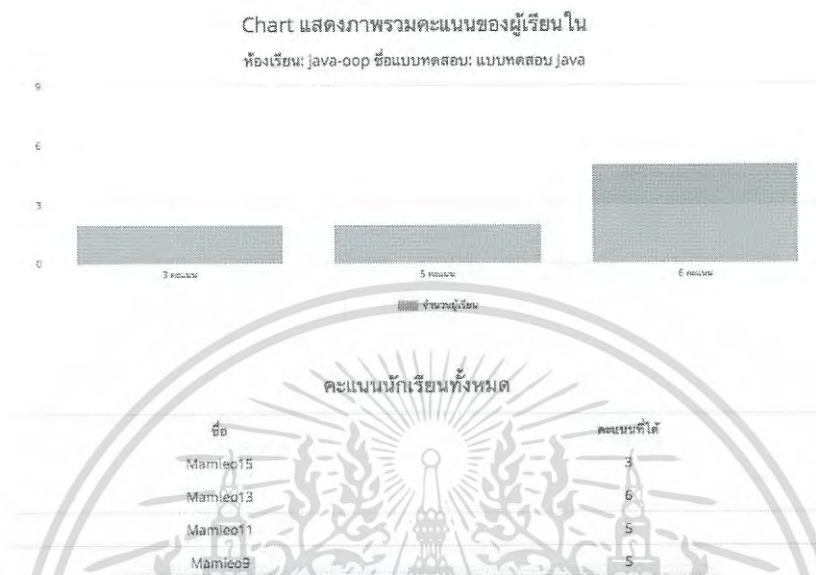
The screenshot shows a web interface for a Grader system. At the top, there's a header with '<1> LOGO', 'Coder-Space', 'Ajooob', and 'Logout'. Below the header, there's a navigation bar with 'E-Classroom', 'Ajooob', 'Editor', and 'Quiz'. The main content area is divided into a sidebar on the left and a main panel on the right. The sidebar has 'Your view' with 'Ajooob' and 'Student view' with 'จำนวนนักเรียน 2 คน', 'Book', and 'Mameo2'. The main panel has a 'Topic Name' field with the value 'เลขคณิตเบื้องต้น'. Below that is 'Problem 1' with a Thai text description: 'หอยทากไต่เสาสูงในหลุม หากหอยทากจะไต่ขึ้นได้ 5 หูด ใช้เวลา 1 นาที และในขณะไต่หอยทากจะไต่ลง 4 หูด ใช้เวลา 20 วินาที มันจะขึ้นถึงยอดเสาสูงหาว่าใช้เวลานานเท่าไรจึงได้ขึ้นถึงปากหลุมได้เวลาทั้งหมดเท่าไร'. Below the problem description are 'Input (Description)' with 'สามคู่แรก', 'Output (Description)' with 'เจ็ดคือจะหาได้ยังไง', and 'Test-case' with 'Input' and 'Output' fields.

รูปที่ 4.6 หน้าสร้างแบบทดสอบสำหรับระบบ Grader

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.6 หน้าดูผลคะแนนและผลประเมินการทดสอบทั้งหมด

หน้านี้ ผู้สอนสามารถดูคะแนนได้ทั้งหมด โดยมีการแสดงผลในรูปแบบกราฟเพื่อให้มองเห็นภาพรวม และแสดงผลคะแนนรายบุคคล ดังรูปที่ 4.7

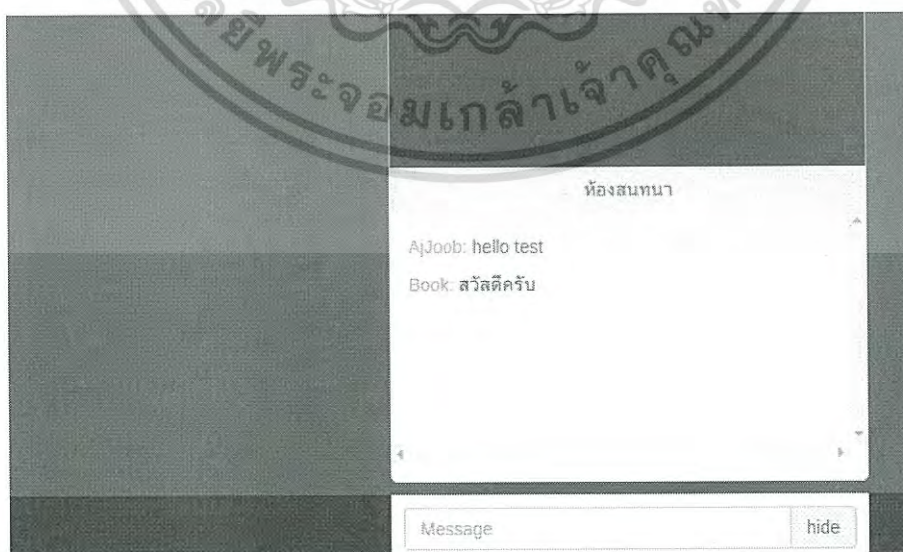


รูปที่ 4.7 หน้าดูผลคะแนนและผลประเมินการทดสอบทั้งหมด

4.2.2.7 หน้าส่งข้อความติดต่อ (Chat)

ผู้สอนและผู้เรียนสามารถส่งข้อความติดต่อกันได้ เป็นแชทภายในห้องเรียนห้องนั้น ดัง

รูปที่ 4.8

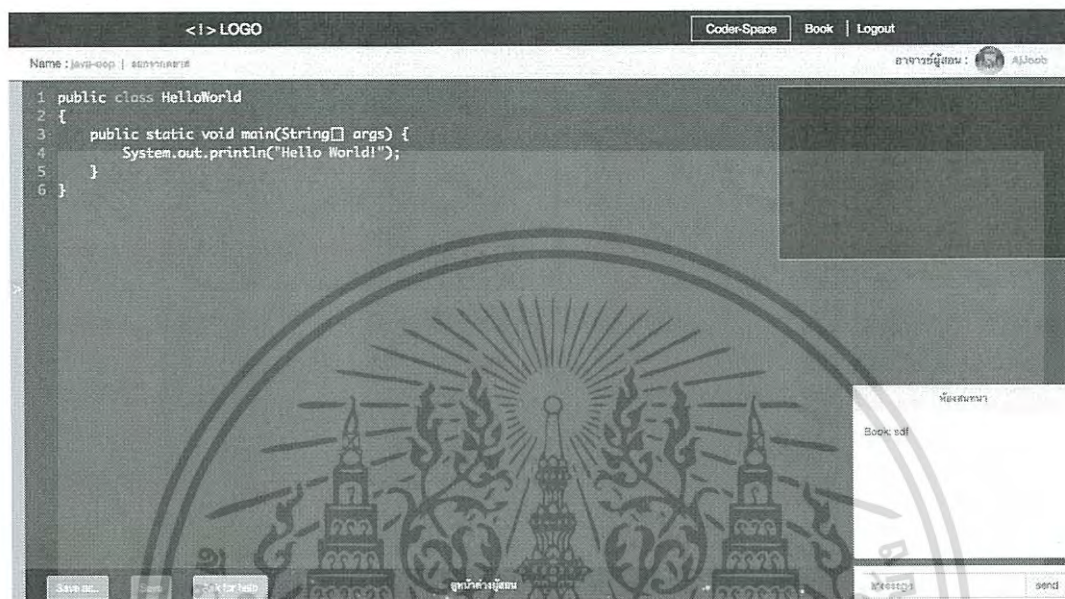


รูปที่ 4.8 หน้าส่งข้อความติดต่อกัน (Chat)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3.2 หน้า Editor ของผู้เรียน

หน้านี้ ผู้เรียนสามารถเขียนโค้ดลงไป Editor และสามารถประมวลผลโค้ดผ่านระบบนี้ได้ ที่ตรงส่วนของ Terminal ที่อยู่มุมขวามือ นอกจากนี้ยังสามารถดูโค้ดของผู้สอนได้ ซึ่งสามารถดูได้เมื่อผู้สอนเปิดการเข้าถึงให้ดูได้เท่านั้น ดังรูปที่ 4.10



รูปที่ 4.10 หน้า Editor ของผู้เรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3.3 หน้า Editor ของผู้สอน

หน้านี้ ผู้เรียนสามารถดูโค้ดของผู้สอนได้แต่ไม่สามารถแก้ไข และประมวลผลโค้ดของผู้สอนได้ ซึ่งผู้เรียนสามารถดูหน้า Editor ของผู้สอนได้ก็ต่อเมื่อผู้สอนเปิดการเข้าถึงให้ดูได้เท่านั้น ดังรูปที่ 4.11

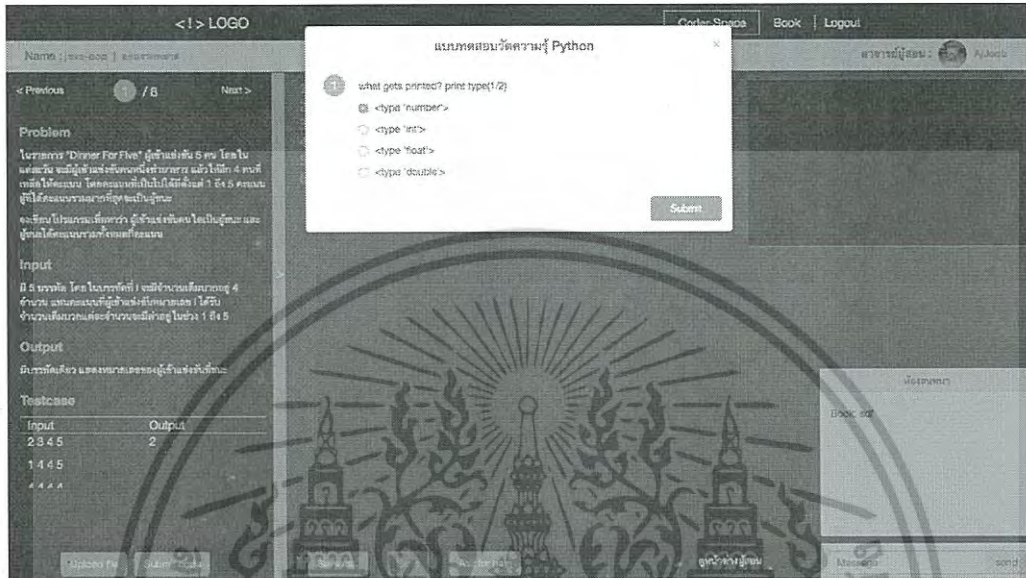
```
localhost:3000/e-classroom/java-oop/teacher/student
1 public class HelloWorld
2 {
3     public static void main(String[] args) {
4         System.out.println("Hello World!");
5     }
6 }
```

รูปที่ 4.11 หน้า Editor ของผู้สอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3.4 หน้าทำแบบทดสอบประเภทตอบคำถาม

หน้านี้ เมื่อผู้สอนสร้างแบบทดสอบ แบบทดสอบนั้นจะแสดงบนหน้าจอของผู้เรียน และผู้เรียนต้องทำให้เสร็จก่อนเวลาที่กำหนด หากเลยเวลาไปแล้ว ผู้เรียนจะไม่สามารถทำแบบทดสอบนั้นได้ ดังรูปที่ 4.12



รูปที่ 4.12 หน้า Editor ของผู้สอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3.5 หน้าทำแบบทดสอบของระบบ Grader

หน้านี้ เมื่อผู้สร้างสร้างโจทย์สำหรับระบบ Grader เรียบร้อยแล้ว จะมี Alert แจ้งว่าสามารถเริ่มทำโจทย์ได้ โดยผู้เรียนต้องเขียนโค้ดลงใน Editor หรือ อัปโหลดไฟล์จากภายนอก จากนั้นกด Submit เพื่อส่งโค้ด ไปให้ระบบทำการตรวจสอบจาก Test Case ที่ผู้สอนสร้างขึ้นมา ถ้าหากผิดในบางเทสเคส สามารถแก้ไขข้อเดิมหรือข้ามไปยังข้อถัดไปได้ จนกว่าจะหมดเวลาการทำแบบทดสอบ ดังรูปที่ 4.13

The screenshot displays the Grader system interface. At the top, there is a navigation bar with a logo, 'Codar-Space', 'Book', and 'Logout' links. Below this, the user's name 'Name: java-oop | สอนทศพร' and the instructor's name 'อาจารย์ผู้สอน: AlJobb' are visible. The main content area is divided into several sections:

- Problem:** A text description in Thai about a 'Dinner For Five' problem involving 5 people and 4 tables.
- Input:** A text input field containing the number '5'.
- Output:** A text input field containing the number '2'.
- Testcase:** A table showing the expected input and output for the test case.

Input	Output
2 3 4 5	2
1 4 4 5	
4 4 4 4	
- Code Editor:** A text area containing the following Java code:


```
1 public class HelloWorld
2 {
3     public static void main(String[] args) {
4         System.out.println("Hello World!");
5     }
6 }
```
- Buttons:** At the bottom, there are buttons for 'Upload file', 'Submit code', 'Run', and 'View help'.

รูปที่ 4.13 หน้าทำแบบทดสอบของระบบ Grader

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

โครงการนี้มีรูปแบบเป็นเว็บแอปพลิเคชัน ที่ช่วยทำให้การเรียนและการสอนเขียนโปรแกรมมีประสิทธิภาพมากขึ้น ซึ่งในส่วนของฝั่งเซิร์ฟเวอร์พัฒนาขึ้นโดยภาษา Node.js และใช้เทคโนโลยี Docker เข้ามาช่วยในการทำให้เว็บมีประสิทธิภาพมากยิ่งขึ้น

การทำงานของระบบนี้ จะแบ่งหน้าที่ออกเป็นสองหน้าที่ คือ ผู้เรียน และผู้สอน โดยที่ผู้สอนจะสามารถสร้างห้องเรียนที่จะให้ผู้เรียนเข้ามาเรียนได้ เมื่อผู้เรียนเข้ามาเรียนจะพบกับสองหน้าจอ คือ หน้าจอของตนเอง และหน้าจอของผู้สอน ซึ่งในขณะที่สอนอยู่ผู้เรียนจะดูในส่วนของหน้าจอของผู้สอน และเมื่อผู้สอนต้องการให้ผู้เรียนลองฝึกเขียนโปรแกรมด้วยตนเอง ผู้เรียนจะสามารถเขียนโปรแกรมในส่วนของหน้าจอตนเองได้ ในขณะที่ผู้เรียนเขียนโปรแกรมอยู่นั้น หากผู้สอนต้องการดูโค้ดของผู้เรียน ผู้สอนจะสามารถเข้าไปดู และสามารถแก้ไขโค้ดของผู้เรียนได้ โดยการแสดงผลทั้งหมดที่กล่าวมานั้น จะเป็นการแสดงผลแบบ Real time ผ่านเทคโนโลยีที่ชื่อว่า Websocket

นอกจากนี้ยังมีระบบแบบทดสอบ ผู้สอนจะสามารถสร้างแบบทดสอบโดยกำหนดคำถามและตัวเลือกคำตอบเองได้ อีกทั้งยังสามารถตั้งช่วงเวลาในการทำแบบทดสอบได้อีกด้วย หากผู้เรียนไม่ได้เริ่มทำแบบทดสอบภายในระยะเวลาที่กำหนด ผู้เรียนก็ไม่มีสิทธิ์ทำแบบทดสอบได้อีก เมื่อผู้เรียนทำแบบทดสอบเสร็จเรียบร้อยแล้ว จะสามารถดูผลคะแนนของตนเองและทราบคำตอบที่ถูกต้องได้ และผู้สอนสามารถดูคะแนนทั้งหมดของผู้เรียนทั้งหมดได้

อีกทั้งยังมีระบบ Grader เป็นระบบที่ให้ผู้เรียนเขียนโค้ดและอัปโหลดขึ้นเพื่อให้ระบบของเราทำการตรวจสอบผลคลังข้ออัตโนมัติ โดยจะมีเทสเคสที่แตกต่างกันไปตามที่ผู้สอนกำหนด และผู้สอนสามารถดูผลคะแนนของเด็กได้ สามารถดูได้ว่าผู้เรียนผ่านเทสเคสใดบ้าง เพื่อประเมินทั้งภาพรวมและรายบุคคลว่าผู้เรียนเข้าใจมากน้อยเพียงใด

5.2 ปัญหาและอุปสรรคที่พบในการพัฒนาระบบ

1. เทคโนโลยีที่นำมาใช้ ไม่ว่าจะเป็น Websocket หรือ Docker ล้วนเป็นเทคโนโลยีใหม่ที่ยังไม่นิ่ง และมีการพัฒนาอยู่ตลอดเวลา ทำให้ในบางอย่างซึ่งเป็น Feature ที่ควรรู้ได้ กลับใช้ไม่ได้ เนื่องจากนักพัฒนายังพัฒนาเทคโนโลยีเหล่านี้ไม่สมบูรณ์ และรวมไปถึงเทคโนโลยีใหม่ๆเหล่านี้ มีข้อมูลให้ศึกษาค่อนข้างน้อย
2. เนื่องจากโครงการนี้เป็นรูปแบบที่สามารถนำโค้ดขึ้นมาประมวลผลบนเซิร์ฟเวอร์ได้ จึงจำเป็นต้องป้องกันเป็นพิเศษ เพราะอาจมีผู้ไม่หวังดี มาโจมตีได้
3. เพื่อให้ผู้ใช้สามารถใช้งานเว็บแอปพลิเคชันเราได้ง่ายที่สุด จำเป็นต้องให้การตอบสนองต่างๆ เป็นแบบ Real time ทำให้ยากต่อการพัฒนาเป็นอย่างมาก

5.3 ข้อเสนอแนะและแนวทางการพัฒนาระบบในอนาคต

1. ปรับปรุงรูปแบบเว็บให้สวยงาม และเข้าใจง่ายขึ้น
2. มีระบบเก็บประวัติการเรียนของผู้เรียนทั้งหมด
3. อาจมีการเพิ่มระบบเสียงในภายหลัง เพื่อสนับสนุนการสอนในเชิงออนไลน์มากขึ้น ทำให้การสอนมีประสิทธิภาพและสื่อสารให้เข้าใจกันได้ง่ายมากขึ้น
4. อาจมีระบบคอร์สเรียน โดยให้ผู้สอนสามารถกำหนดรูปแบบของคอร์สได้เอง ว่าจะเป็นคลิปวิดีโอเผยแพร่ความรู้หรือว่าเป็น โจทย์แบบทดสอบให้ทำระบบนี้แยกจากระบบ Coder-Space ผู้เรียนสามารถเข้าไปเรียนรู้จากระบบคอร์สเมื่อใดก็ได้

บรรณานุกรม

- [1] Node.js Foundation “**Node.js v5.1.0 Documentation.**” [cited 2016 Apr 4]
 [Online].Available from: <https://nodejs.org/api/>
- [2] Socket.io Development Team “**Socket.io Documentation.**” [cited 2016 Apr 4]
 [Online].Available from: <http://socket.io/docs/>
- [3] Express.js Development Team “**Express 4.x API.**” [cited 2016 Apr 4]
 [Online].Available from: <http://expressjs.com/4x/api.html>
- [4] Docker Engineering Team “**Docker Documentation.**” [cited 2016 Apr 4]
 [Online].Available from: <https://docs.docker.com/>





ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก
คู่มือการติดตั้งระบบและคู่มือการใช้งานระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการติดตั้ง

สำหรับระบบปฏิบัติการที่จะทำการรัน Project นี้ได้ จำเป็นต้องเป็น Ubuntu หรือ OS X เท่านั้น โดยในเอกสารนี้จะแสดงถึงการติดตั้ง ในรูปแบบของ Ubuntu

1. เปิด Terminal

2. ทำการติดตั้ง Curl

```
$ sudo apt-get update
```

```
$ sudo apt-get install curl
```

3. ใช้ Curl โหลด และติดตั้ง Docker

```
$ curl -fsSL https://get.docker.com/ | sh
```

```
$ curl -fsSL https://get.docker.com/gpg | sudo apt-key add -
```

4. ติดตั้ง Node.js และ Npm

```
$ sudo apt-get install nodejs
```

```
$ sudo apt-get install npm
```

5. ติดตั้ง Redis Server

```
$ wget http://download.redis.io/redis-stable.tar.gz
```

```
$ tar xvzf redis-stable.tar.gz
```

```
$ cd redis-stable
```

```
$ make
```

6. ติดตั้ง Mongo DB

```
$ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv EA312927
```

```
$ echo "deb http://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3.2  
multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.2.list
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install -y mongodb-org
```

```
$ sudo service mongod start
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ดาวน์โหลด Source code ของโปรเจก จาก

https://github.com/flipre/Nsc_project/archive/master.zip

8. เปลี่ยน directory ไปที่ folder ของโปรเจก

9. ทำการติดตั้ง package ที่จำเป็นในการรัน โปรเจก

\$ npm install

10. รัน โปรเจก

\$ node index.js

11. ทำการสร้าง User โดยเข้าไปที่

localhost:3000/create/teacher/AjJoob

localhost:3000/create/student/Book

12. เข้าหน้าเว็บไซต์

localhost:3000



คู่มือการใช้งาน

1. เมื่อกดลิ้งค์ localhost:3000 จะเจอกับหน้าแรก ผู้ใช้จะได้เห็นรายละเอียดภาพรวมเกี่ยวกับการทำงานของเว็บไซต์ สามารถสมัครสมาชิกและล็อกอินเข้าสู่ระบบได้ ดังรูปที่ ก.1



รูปที่ ก.1 หน้าแรกของเว็บไซต์ Coder Planet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีเป็นผู้สอน

1. ต้องการสร้าง Coder-Space

1.1 กดปุ่ม Coder-Space จะขึ้นให้กรอกรายละเอียด ดังรูปที่ ก.2

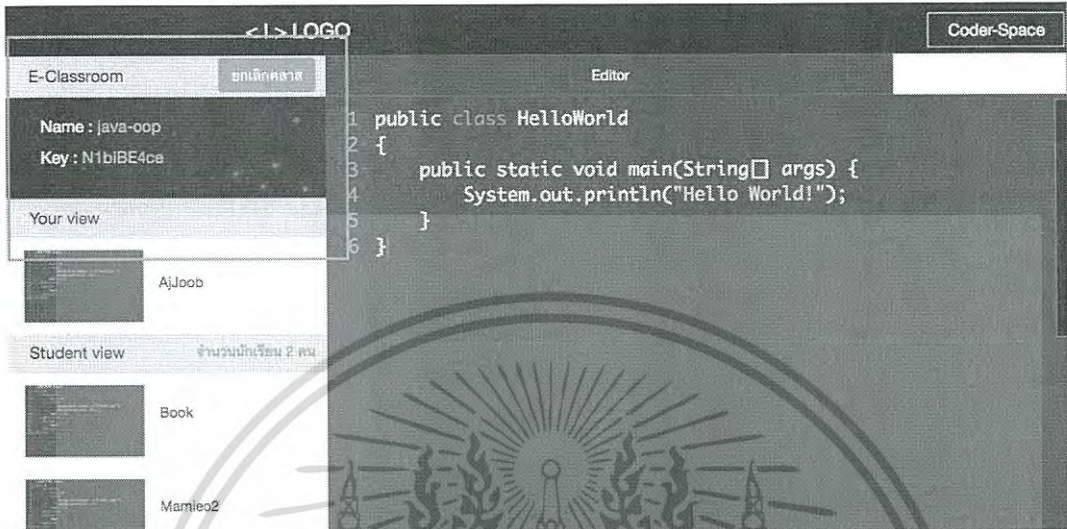
รูปที่ ก.2 หน้าสร้างห้องเรียน Coder-Space

1.2 เมื่อกดสร้างห้องแล้ว จะเข้าสู่ระบบ Coder-Space ดังรูปที่ ก.3

รูปที่ ก.3 หน้าห้องเรียน Coder-Space

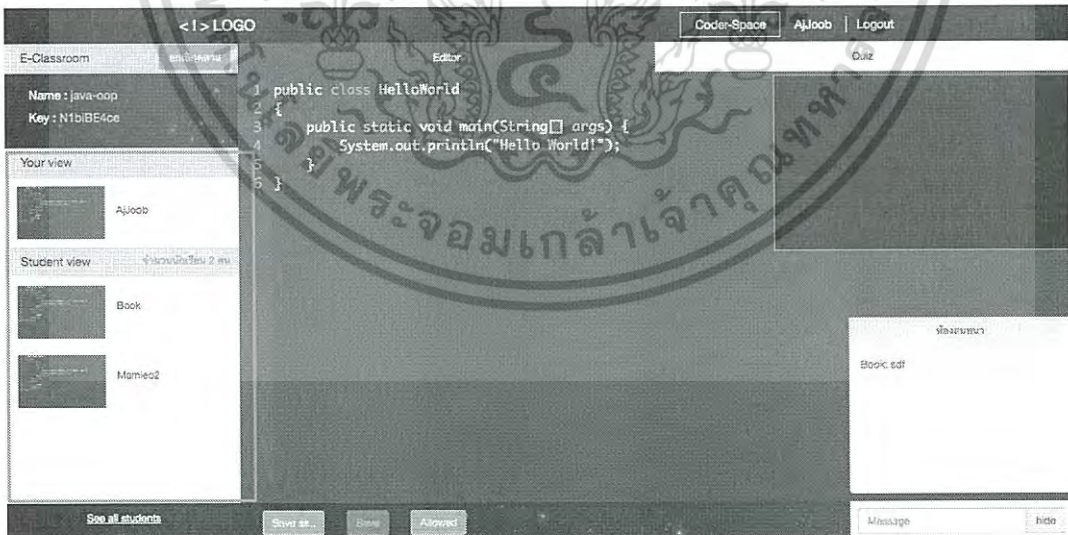
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 จะเห็นว่ามุมมองนักเรียน จะมีรหัสเข้าห้อง ดังภาพด้านล่าง โดยรหัสนี้ระบบจะสร้างขึ้นทุก การสร้างห้องเรียน Coder-Space ผู้สอนต้องนำรหัสเข้าห้องนี้ไปแจกจ่ายให้กับผู้เรียนที่ผู้สอน ต้องการให้เข้าเรียนในคลาสนี้ ดังรูปที่ ก.4



รูปที่ ก.4 แสดงรหัสเข้าห้อง

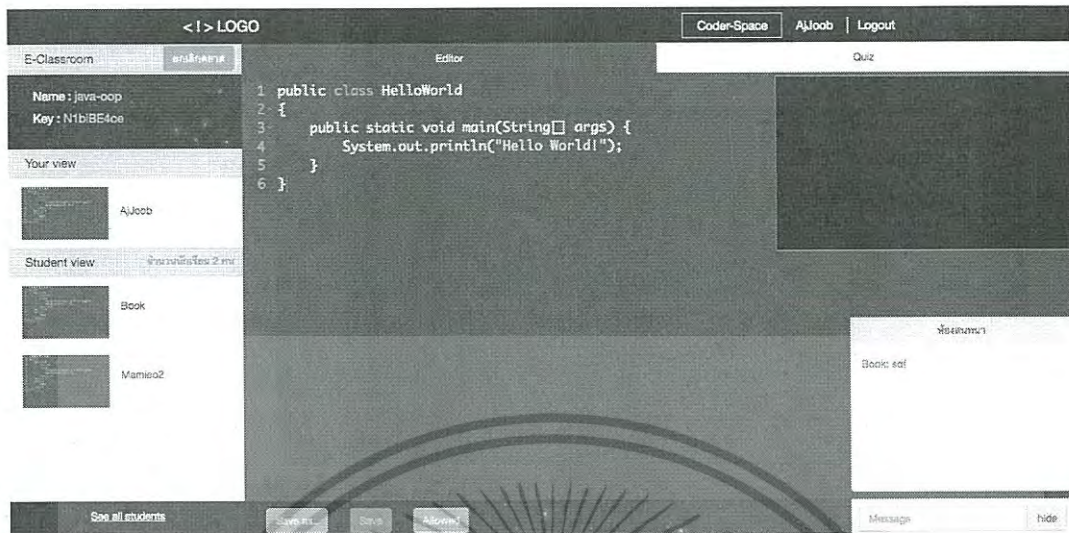
1.4 เมื่อผู้เรียนเข้าห้องเรียนในระบบ Coder-Space แล้ว จะขึ้นแถบด้านข้าง ผู้สอนสามารถ กดเข้าไปดู แก้ไข และประมวลผล โค้ดของผู้เรียนที่ต้องการได้ ดังรูปที่ ก.5



รูปที่ ก.5 แถบ Sidebar

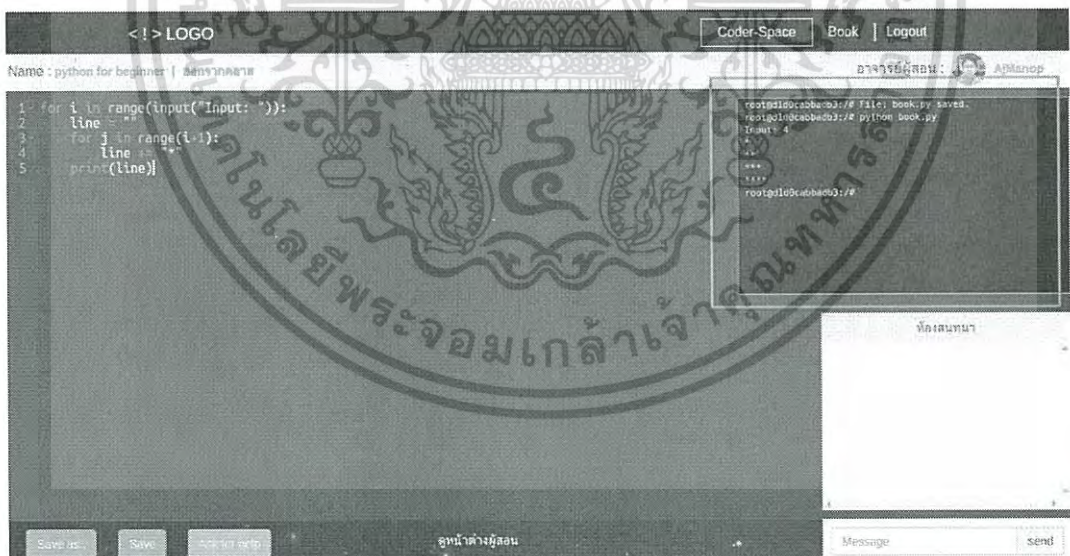
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้สอนสามารถเขียนโค้ดลงใน Editor ของระบบ Coder-Space ได้ ดังรูปที่ ก.6



รูปที่ ก.6 Editor ของระบบ Coder-Space

หากต้องการจะประมวลผลโค้ด จำเป็นต้องกดปุ่ม save เพื่อจัดเก็บไฟล์ไว้บน Docker ของเครื่องนั้น จากนั้นทำการประมวลผลผ่านทาง Terminal ที่อยู่มุมขวาบนของหน้าจอ ดังรูปที่ ก.7



รูปที่ ก.7 การประมวลผลผ่าน Terminal

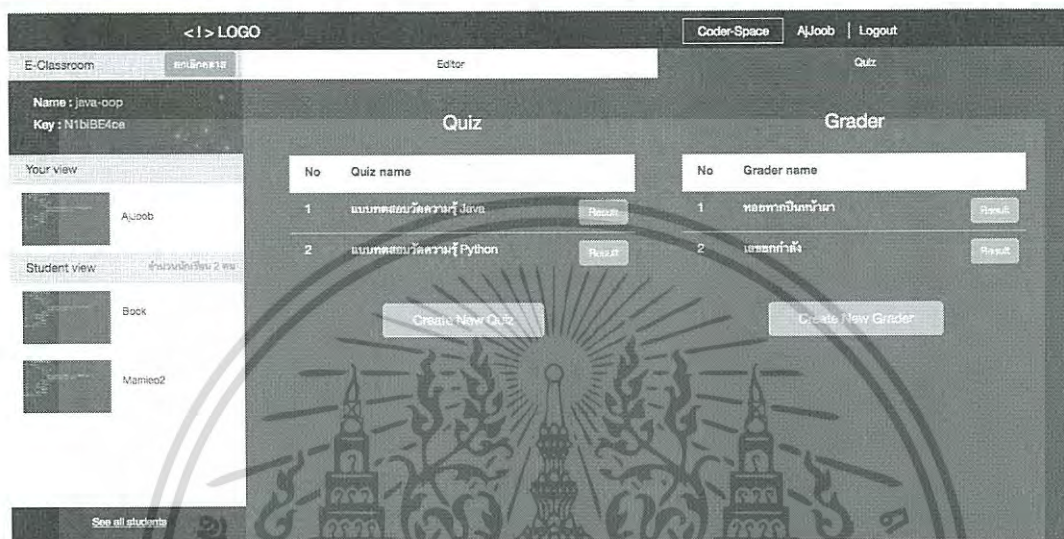
ในส่วนของ Editor จะรองรับแค่ภาษาเขียนโปรแกรมที่สามารถประมวลผลผ่านทาง Command line ได้เท่านั้น ระบบ Coder-Space นี้ รองรับภาษา Node.js Python Java และ JavaScript

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ต้องการสร้างแบบทดสอบ

2.1 สร้างแบบทดสอบประเภทตอบคำถาม

2.1.1 จะเห็นว่ามี tab อยู่ 2 tab คือ Editor และ Quiz ให้กด tab ชื่อ Quiz จะแสดงผลดังภาพด้านล่าง ซึ่งจะแบ่งออกเป็น 2 ตาราง คือตารางแสดงประวัติการสร้างแบบทดสอบประเภทตอบคำถามและแบบระบบ Grader ในคลาสนี้ทั้งหมด ดังรูปที่ ก.8



รูปที่ ก.8 หน้าแสดงแบบทดสอบทั้งหมด

2.1.2 กดปุ่ม Create New Quiz ดังรูปที่ ก.9



รูปที่ ก.9 หน้าสร้างแบบทดสอบประเภทตอบคำถาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรอกรายละเอียดตามภาพด้านบนให้ครบถ้วน จำเป็นต้องเลือกตัวเลือกที่ถูกต้องเพียงข้อเดียว เพื่อเอาไว้ตรวจสอบเมื่อผู้เรียนทำแบบทดสอบ แล้วประมวลผลคะแนนออกมา

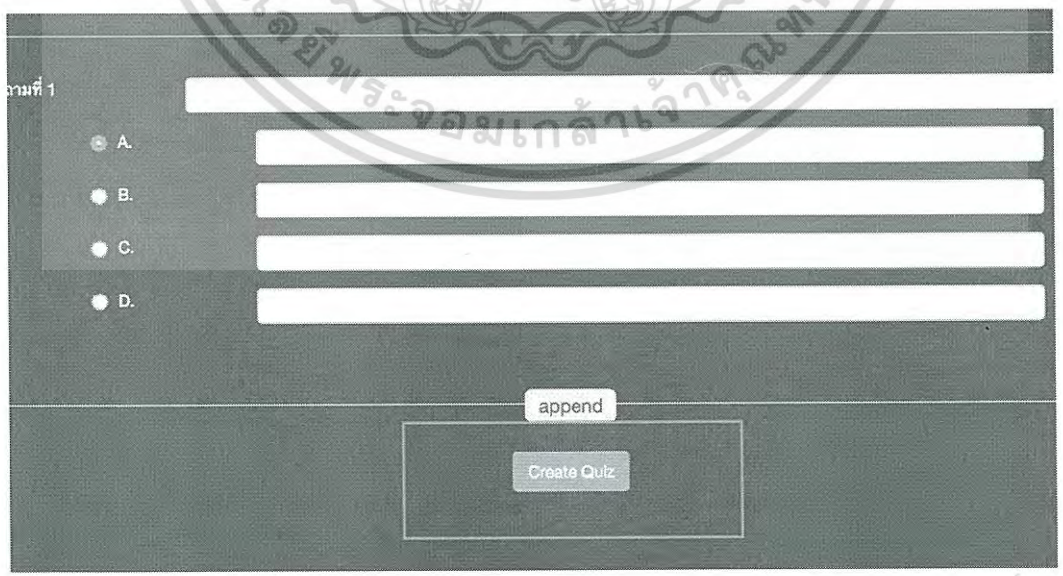
2.2.3 ตั้งเวลาทำแบบทดสอบ

สามารถตั้งช่วงระยะเวลาในการทำแบบทดสอบได้ ถ้าหากผู้เรียนเข้ามาในระบบหลังจากเลยระยะเวลาที่กำหนดไว้แล้ว จะไม่สามารถทำแบบทดสอบได้อีก ดังรูปที่ ก.10



รูปที่ ก.10 การตั้งเวลาทำแบบทดสอบ

2.2.4 กดปุ่ม Create Quiz ดังรูปที่ ก.11



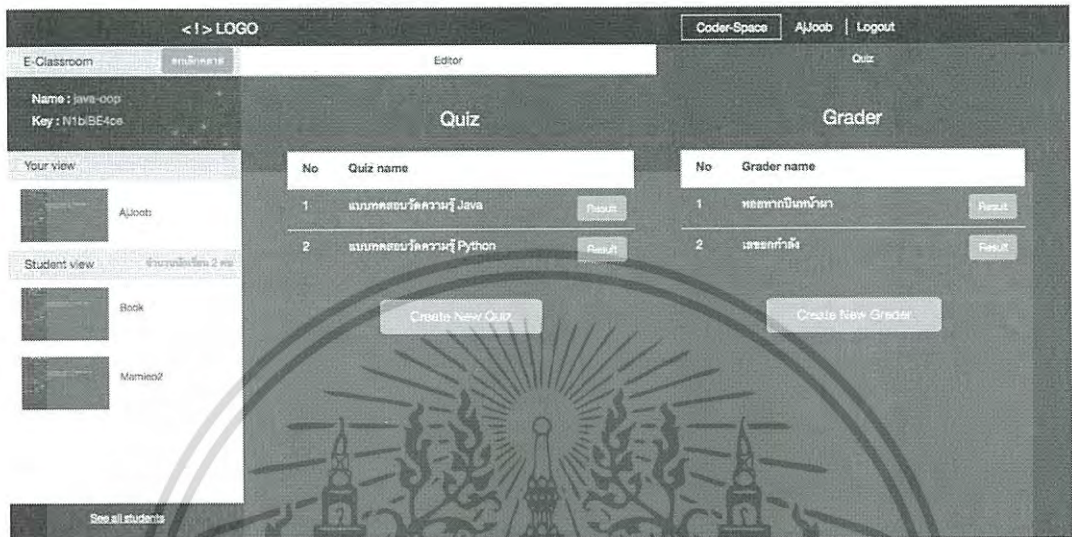
รูปที่ ก.11 ปุ่ม Create Quiz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ผลคะแนนของผู้เรียน

ผู้สอนสามารถดูผลคะแนนแบบทดสอบที่ผู้เรียนทำในระบบ Coder-Space คลาสนั้นได้ โดยมีขั้นตอนดังนี้

3.1 กดไปยัง tab ของ Quiz ดังรูปที่ ก.14



รูปที่ ก.14 หน้าแสดงแบบทดสอบทั้งหมด

3.2 กดปุ่ม Result ของแบบทดสอบที่ต้องการ จะแสดงผลคะแนน ดังรูปที่ ก.15



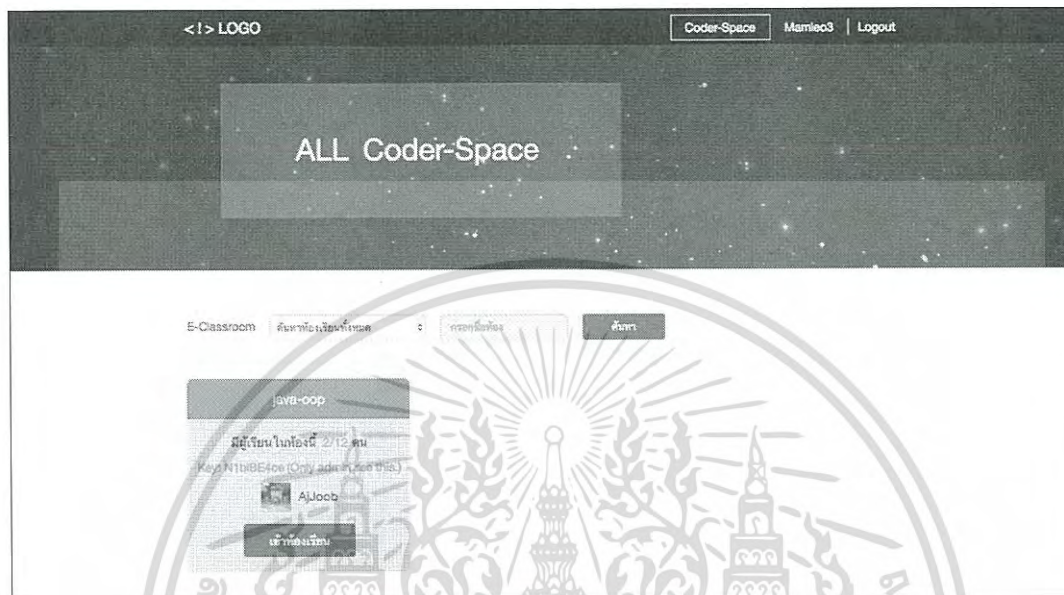
รูปที่ ก.15 หน้าแสดงแสดงผลคะแนนและผลประเมินแบบทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีเป็นผู้เรียน

1. เข้าห้องเรียน Coder-Space

1.1 เมื่อล็อกอินเข้ามาจะเจอกับหน้ารวมห้องเรียน Coder-Space ทั้งหมด ดังรูปที่ ก.16



รูปที่ ก.16 หน้าแสดงห้องเรียน Coder-Space ทั้งหมด

1.2 ผู้เรียนเลือกห้องเรียนที่ต้องการเรียน และกรอกรหัสเข้าห้องเรียนดังภาพด้านล่าง รหัสเข้าห้องจะได้จากผู้สอนใน Coder-Space ห้องนั้นๆ ดังรูปที่ ก.17



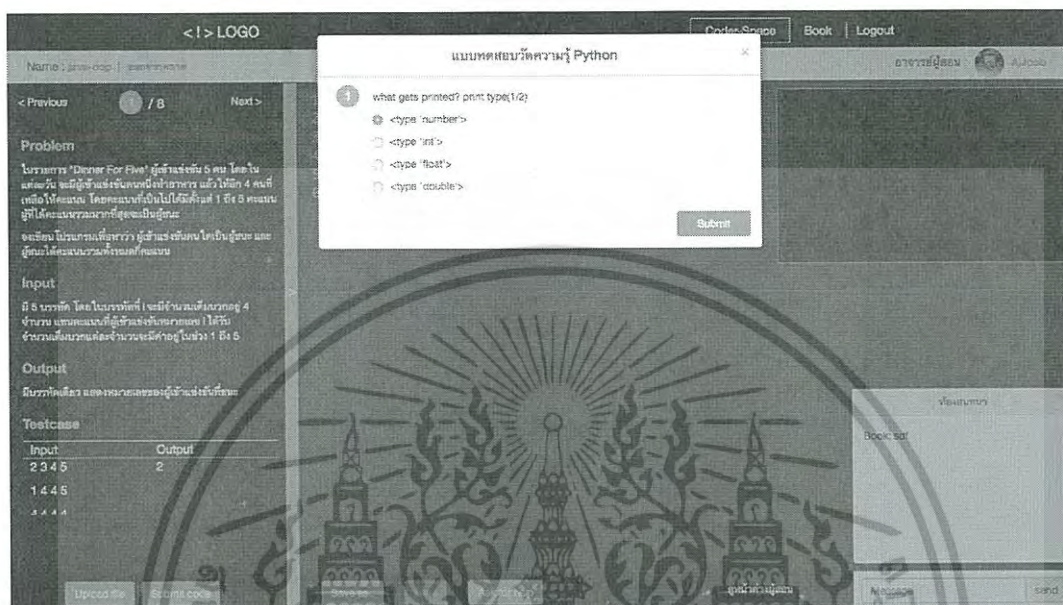
รูปที่ ก.17 แสดงการกรอกรหัสเข้าห้องเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทำแบบทดสอบ

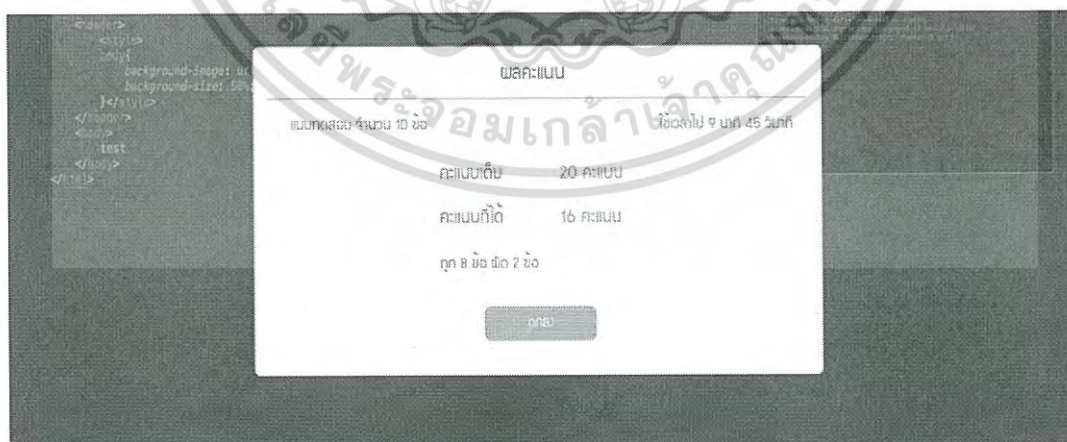
2.1 ทำแบบทดสอบประเภทตอบคำถาม

2.1.1 เมื่อผู้สอนทำการสร้างแบบทดสอบในระบบของผู้สอน เมื่อผู้เรียนเข้ามาในหน้า Coder-Space จะเป็นการเริ่มแบบทดสอบทันที ผู้เรียนจะเห็น ดังรูปที่ ก.20



รูปที่ ก.20 หน้าทำแบบทดสอบประเภทตอบคำถาม

2.1.2 ผู้เรียนต้องทำแบบทดสอบ ให้เสร็จทันภายในระยะเวลาที่กำหนด เมื่อทำเสร็จครบทุกข้อจะสามารถดูผลคะแนนได้ ดังภาพด้านล่าง ดังรูปที่ ก.21

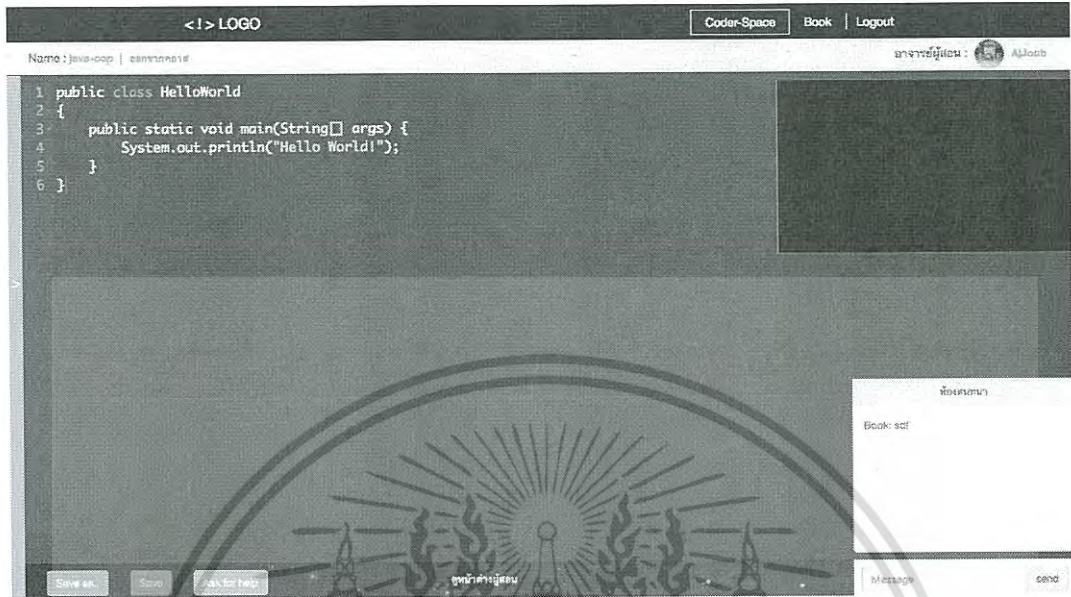


รูปที่ ก.21 แสดงผลคะแนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

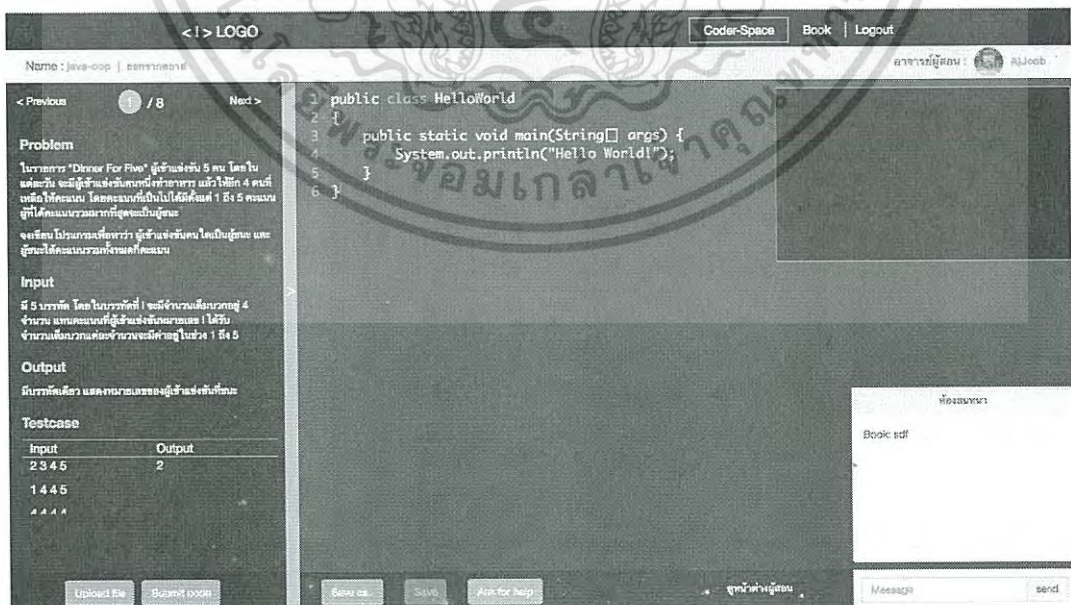
2.2 ทำแบบทดสอบระบบ Grader

2.2.1 ในขณะที่ผู้เรียนทำการเขียนโปรแกรมใน Coder-Space จากเดิมที่เห็น ดังรูปที่ ก.22



รูปที่ ก.22 หน้าห้องเรียน Coder-Space ของผู้เรียน

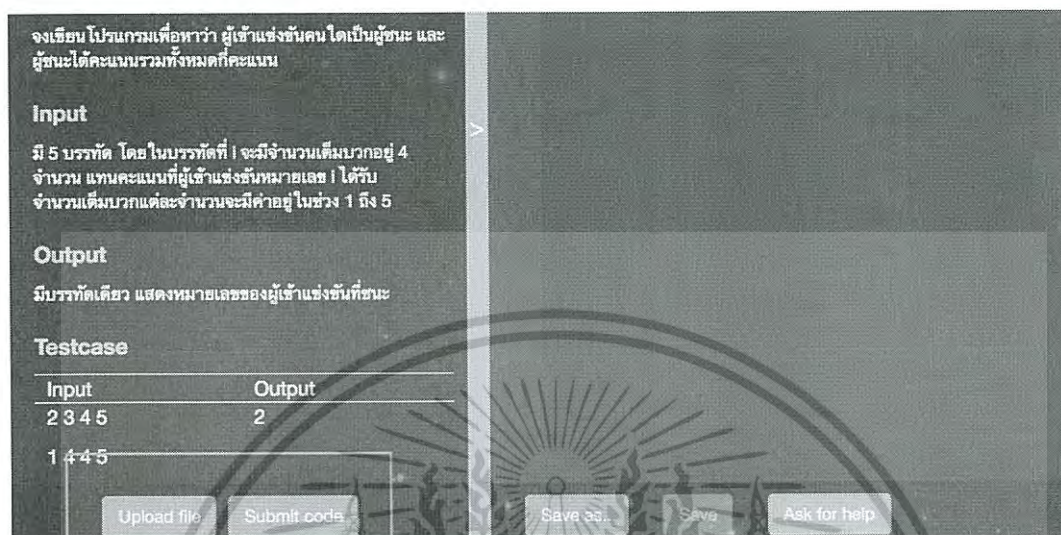
เมื่อผู้สอนทำการสร้างแบบทดสอบสำหรับระบบ Grader จะมีแถบด้านข้างซึ่งเป็นโจทย์สไลด์ออกมา เพื่อให้ผู้เรียนทราบว่าผู้สอนได้สร้างแบบทดสอบสำหรับระบบ Grader แล้ว และผู้เรียนสามารถเริ่มทำแบบทดสอบนี้ได้ ดังรูปที่ ก.23



รูปที่ ก.23 หน้าห้องเรียน Coder-Space ของผู้เรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ผู้เรียนต้องเขียนโปรแกรมเพื่อนแก้โจทย์ปัญหา โดยสามารถเขียนใน Editor ของระบบ Coder-Space ได้และกดปุ่ม Submit หรือหากต้องการอัปโหลดไฟล์จากภายนอกก็ทำได้เช่นกัน โดยกดปุ่ม Upload file ดังรูปที่ ก.24



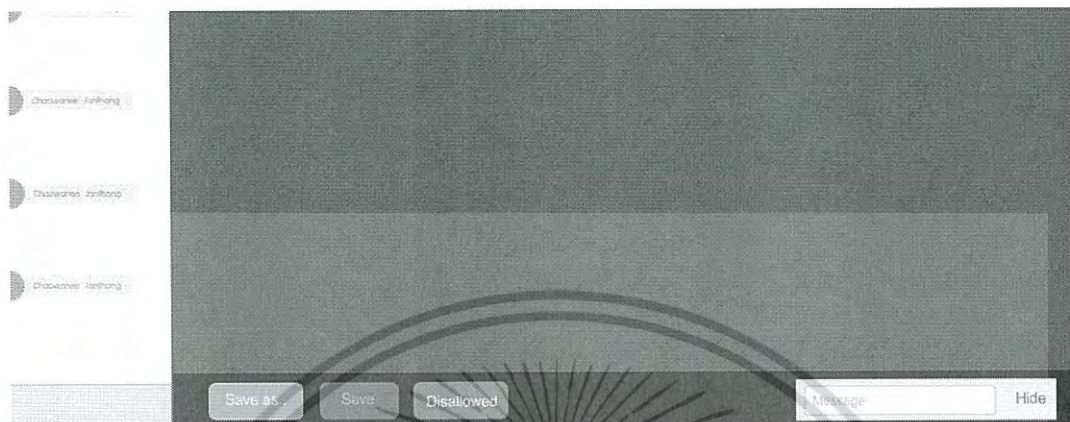
รูปที่ ก.24 การอัปโหลดไฟล์สำหรับระบบ Grader

ระบบจะทำการตรวจสอบอัตโนมัติ จาก Test-Case ทุก Test-Case ที่ผู้สอนสร้าง และจะทำการประมวลผลคะแนนออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เมื่อผู้เรียนต้องการดู Editor ของผู้สอน

3.1 ผู้สอนจำเป็นต้องกดปุ่ม Disallowed เพื่อเปิดการเข้าถึงให้ผู้เรียนสามารถดูโค้ดของผู้สอนได้ ดังรูปที่ ก.25



รูปที่ ก.25 ปุ่ม Disallowed

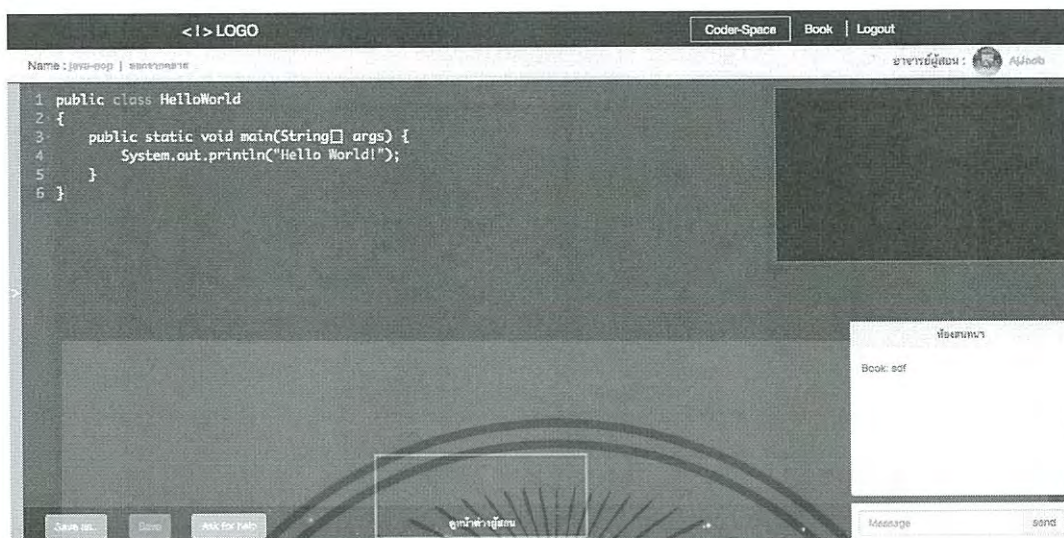
เมื่อกดแล้วจะขึ้นปุ่ม Allowed แสดงว่าได้มีการเปิดการเข้าถึงแล้ว ดังรูปที่ ก.26



รูปที่ ก.26 ปุ่ม Allowed

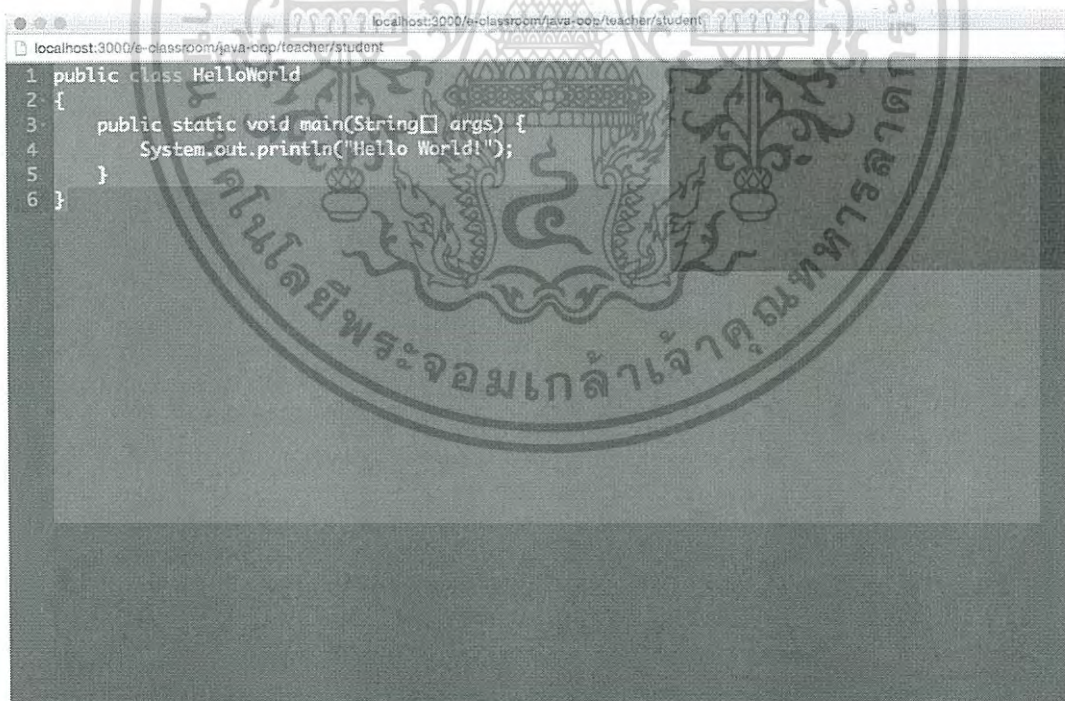
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ผู้เรียนกดข้อความที่เขียนว่า ดู Editor ของผู้สอน ที่อยู่แถบด้านล่าง ดังรูปที่ ก.27



รูปที่ ก.27 ปุ่มดูหน้าต่างของผู้สอน

3.3 เมื่อกดแล้วจะแสดงผลรูปภาพ ผู้เรียนสามารถดูโค้ดของผู้สอนได้เพียงอย่างเดียว ไม่สามารถแก้ไข หรือประมวลผลโค้ดของผู้สอนได้ ดังรูปที่ ก.28

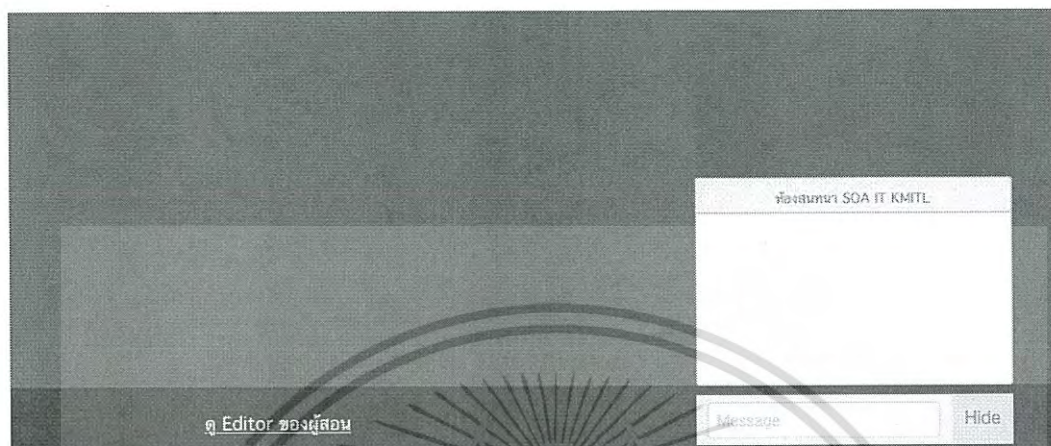


รูปที่ ก.28 ดูหน้าต่าง Editor ของผู้สอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ระบบแชท

นอกจากนี้ยังมีระบบแชท ที่สามารถส่งข้อความถึงกันใน Coder-Space คลาสนั้นๆ ได้ ดังรูปที่ ก.29



รูปที่ ก.29 ระบบแชท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ นางสาวเขาวนีย์ จันทร์ทอง
 วัน เดือน ปี เกิด 19 ตุลาคม 2536
 ที่อยู่ เลขที่ 29/35 หมู่ 5 ตำบล เกาะหลัก อำเภอ เมือง จังหวัด ประจวบคีรีขันธ์ 77000
 ประวัติการศึกษา วิทยาศาสตรบัณฑิต สาขาเทคโนโลยีสารสนเทศ
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อ นายณภัทร รุ่งรุจิเมฆ
 วัน เดือน ปี เกิด 30 มีนาคม 2537
 ที่อยู่ เลขที่ 12/4 หมู่ที่ 4 ตำบล ชุนศรี อำเภอ ไทรน้อย จังหวัด นนทบุรี 11140
 ประวัติการศึกษา วิทยาศาสตรบัณฑิต สาขาเทคโนโลยีสารสนเทศ
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดการห้องเรียนสำหรับการเขียนโปรแกรมอย่างมีประสิทธิภาพ

เขาวนิจันทรทออง¹ และ ณภัทร รุ่งรุจิเมฆ²

¹คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ

Emails: bookotll@gmail.com, mamieoiz123@gmail.com

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอโครงการจัดการห้องเรียนสำหรับการเขียนโปรแกรมอย่างมีประสิทธิภาพ ซึ่งเป็นเว็บไซต์ที่สอนการเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์ต่างๆ โดยจะเป็นระบบห้องเรียนอิเล็กทรอนิกส์ ซึ่งเปรียบเสมือนห้องเรียนห้องหนึ่ง ที่จะมีรหัสกำหนดการเข้าถึงที่ถูกสร้างขึ้นมา และผู้เรียนต้องใช้รหัสนี้เพื่อเข้าใช้งานห้องเรียน ผู้เรียนและผู้สอนสามารถเขียนโปรแกรมและประมวลผลเพื่อดูผลลัพธ์ผ่านทางเว็บไซต์ โดยผู้เรียนสามารถเห็นโค้ดของผู้สอน และในขณะเดียวกัน ผู้สอนก็จะสามารถดูและแก้ไขโค้ดของผู้เรียนทุกคนในห้องเรียนได้เช่นกัน นอกจากนี้ยังมีระบบแบบทดสอบ ซึ่งเป็นระบบสำหรับช่วยอำนวยความสะดวกให้ผู้เรียนได้ฝึกฝนการทำโจทย์และให้ระบบทำการตรวจได้ทันที เพื่อให้ผู้สอนสามารถประเมินความรู้ความเข้าใจของผู้เรียนได้อีกด้วย

คำสำคัญ – ห้องเรียน; เขียนโปรแกรม; ตรวจโค้ด

1. บทนำ

ในปัจจุบัน การเรียนรู้การเขียนโปรแกรมถือเป็นเรื่องสำคัญมาก เนื่องจากเทคโนโลยีส่วนใหญ่ในปัจจุบัน จำเป็นต้องใช้การเขียนโปรแกรมในการพัฒนา แต่การเรียนรู้ด้านการเขียนโปรแกรมคอมพิวเตอร์ในห้องเรียนนั้นกลับยังไม่มีประสิทธิภาพเท่าที่ควร เช่น การเสียเวลาไปกับการลงเตรียมทรัพยากรที่จำเป็นต้องใช้ในการเรียนเขียนโปรแกรม หรือในขณะที่เรียนเขียนโปรแกรม หากผู้เรียนมีข้อสงสัยหรือปัญหา

ผู้สอนจำเป็นต้องเดินไปอธิบาย แก้ไขปัญหาที่หน้าจอของผู้เรียน ซึ่งเป็นการเสียเวลา และยากต่อการที่จะดูแลผู้เรียนทุกคนได้อย่างทั่วถึง จากที่กล่าวมาข้างต้น จึงได้มีการพัฒนาโครงการจัดการห้องเรียนสำหรับการเขียนโปรแกรมอย่างมีประสิทธิภาพ ซึ่งเป็นเว็บไซต์ที่สอนการเขียนโปรแกรมด้วยภาษาต่างๆ เป็นช่องทางให้ผู้เชี่ยวชาญสามารถถ่ายทอดความรู้ให้แก่ผู้ที่สนใจได้ ผ่านระบบห้องเรียนอิเล็กทรอนิกส์ หรือ Coder-Space ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปรียบเสมือนห้องเรียนห้องหนึ่ง ซึ่งจะมีรหัสกำหนดการเข้าถึง โดยถ้าผู้เรียนทราบรหัสของห้องเรียน ก็จะสามารถเข้าเรียนได้ ซึ่งภายในห้องเรียน ผู้เรียนและผู้สอนสามารถเขียนโปรแกรมและประมวลผลเพื่อดูผลลัพธ์ผ่านเว็บไซต์ และในขณะเดียวกัน ผู้สอนสามารถดูและแก้ไขโค้ดของผู้เรียนทุกคนในห้องเรียนได้เช่นกัน อีกทั้งยังมีระบบแบบทดสอบโดยแบ่งออกเป็นแบบตอบคำถาม และแบบระบบ Grader ซึ่งเป็นระบบสำหรับช่วยอำนวยความสะดวกให้ผู้เรียนได้ฝึกฝนการทำโจทย์แล้วอัปขึ้นระบบเพื่อให้ระบบทำการตรวจและประเมินผลโปรแกรมโดยอัตโนมัติได้ทันที เพื่อให้ผู้สอนสามารถประเมินความรู้ความเข้าใจของผู้เรียนทั้งคลาสได้

2. ทฤษฎีและหลักการ

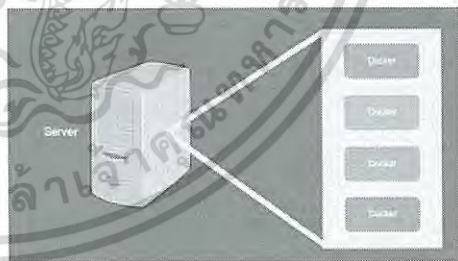
2.1 Node.js

เป็นภาษา JavaScript ที่ทำงานในฝั่งผู้ให้บริการ (Server) ซึ่งในปกติแล้วเว็บทั่วไปแทบทุกเว็บจำเป็นต้องใช้ภาษา JavaScript มาช่วยในการทำงานฝั่งผู้ใช้บริการ (Client) อยู่แล้ว จึงถือเป็นข้อดีข้อหนึ่ง ที่ทำให้ผู้พัฒนาคุ้นเคยและทำความเข้าใจ Node.js ได้ง่ายเนื่องจากมีโครงสร้างภาษา(Syntax) เหมือนกับภาษา JavaScript ที่ใช้อยู่แล้ว นอกจากนี้ Node.js ยังทำงานภายใต้แนวคิดการทำงานแบบไม่พร้อมกัน (Asynchronous) ซึ่งจะช่วยให้สามารถเริ่มประมวลผลงานใหม่โดยที่ไม่ต้องรอการประมวลผลงานเก่าให้เสร็จ

ก่อน โดยแนวคิดนี้ถือเป็นแนวคิดสำคัญที่ช่วยให้สร้างเว็บไซต์ที่จะตอบสนองผู้ใช้ได้แบบทันทีในแง่ของภาษาเขียนโปรแกรม Node.js ถือเป็นภาษาใหม่ ที่มีแนวโน้มการเจริญเติบโตสูงมาก อ้างอิงจากชุมชนต่างๆที่ร่วมนักพัฒนาโปรแกรมเข้าด้วยกัน และด้วยอัตราการเจริญเติบโตนี้ ทำให้ในชุมชนที่ร่วมนักพัฒนาโปรแกรม พร้อมทั้งจะให้คำแนะนำกับนักพัฒนาโปรแกรมใหม่ ๆ ที่ต้องการความช่วยเหลือได้ทันที

2.2 Docker

Docker คือ โปรแกรมที่จำลองระบบปฏิบัติการ (Operating System) โดยที่มีจุดเด่นคือ เราสามารถสร้างแม่พิมพ์ของระบบปฏิบัติการ (Image) เพื่อที่จะใช้แม่พิมพ์นั้นมาสร้างเป็นระบบปฏิบัติการย่อยๆ (Container) ที่เหมือนกับแม่พิมพ์ทุกประการ โดยที่มีขนาดเล็กและกินทรัพยากรเครื่องน้อย และ Docker ไม่ได้ทำการติดต่อกับเว็บไซต์โดยตรง



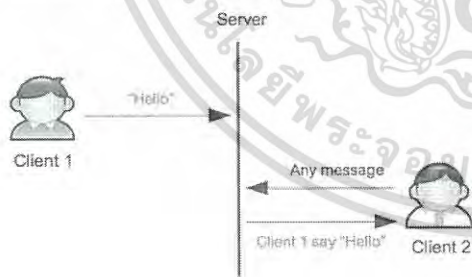
รูปที่ 1. รูปภาพแสดงการทำงานของ Docker

Docker จะถูกนำไปประยุกต์ใช้เพื่อให้นักพัฒนาโปรแกรมต่างๆทำได้ง่ายขึ้น เช่น เมื่อโปรแกรมที่เราจะพัฒนา ต้องการโปรแกรมอื่นๆ เพื่อที่จะทำงานได้ เช่น ฐานข้อมูล เซิร์ฟเวอร์ จำลอง เป็นต้น เมื่อเราต้องการที่จะพัฒนา

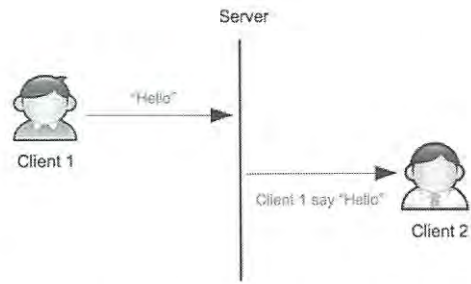
โปรแกรมของเราในที่เครื่องอื่น ก็จะสามารถทำได้ง่าย โดยลงโปรแกรมที่จำเป็นในระบบปฏิบัติการที่ทำงานอยู่บน Docker แล้วนำระบบปฏิบัติการนั้น มาสร้างเป็นแม่พิมพ์ของระบบปฏิบัติการ ทำให้ทุกครั้งที่ต้องการจะพัฒนาโปรแกรมนั้นในเครื่องอื่นๆ จะสามารถทำได้ง่าย เพียงแค่สร้างระบบปฏิบัติการจากแม่พิมพ์ของระบบปฏิบัติการที่เราสร้างไว้

2.3 Websocket

Websocket คือ ช่องทางการสื่อสารระหว่างเซิร์ฟเวอร์กับผู้ใช้รูปแบบหนึ่ง โดยที่การทำงานของ Websocket จะทำงานโดยการสร้างท่อระหว่างผู้ใช้กับเซิร์ฟเวอร์ หากผู้ใช้ต้องการส่งข้อมูลไปให้เซิร์ฟเวอร์ ก็สามารถส่งได้ทันที ในขณะที่เดียวกัน หากเซิร์ฟเวอร์ต้องการส่งข้อมูลให้ผู้ใช้ ก็สามารถทำได้ทันที โดยผู้ใช้ไม่ต้องร้องขอ (Request) ด้วยหลักการทำงานนี้ ทำให้ Websocket เหมาะสำหรับนำมาใช้สำหรับเว็บไซต์ที่ทำงานแบบ Real time



รูปที่ 2 การทำงานแบบ HTTP Server



รูปที่ 3 การทำงานแบบ Websocket Server

3. การวิเคราะห์และออกแบบระบบ

3.1 ศึกษาจากรูปแบบปัจจุบัน

ในปัจจุบันจากการค้นคว้า ทำให้เห็นว่าเว็บไซต์ที่ต้องการเผยแพร่ความรู้เกี่ยวกับการเขียนโปรแกรม จะมีอยู่จำนวนหนึ่ง ซึ่งขอยกตัวอย่างเว็บไซต์ที่มีลักษณะใกล้เคียงกับระบบที่ต้องการจะนำมา 2 เว็บไซต์ ดังนี้

3.1.1 เว็บไซต์ Codeschool.com และเว็บไซต์ Codecademy.com

รูปแบบของเว็บไซต์ Codeschool และเว็บไซต์ Codecademy จะเป็นเว็บไซต์ที่มีลักษณะเด่นดังนี้

1. แต่ละคอร์สเรียน จะแบ่งออกเป็นบทย่อยๆ และแต่ละบทย่อยๆ จะแบ่งออกเป็นขั้นตอนให้ผู้เรียนทำตาม เช่น ขั้นตอนแรกของบท 1 ให้ผู้เรียนศึกษาจากคลิป ขั้นตอนที่สองของบท 1 ให้ผู้เรียนทำการตอบคำถาม ขั้นตอนที่สามของบท 1 ให้ผู้เรียนทดลองเขียนโปรแกรมตามโจทย์กำหนด โดยแต่ละบท จะมีขั้นตอนไม่เหมือนกัน ขึ้นอยู่กับความเหมาะสมของบทนั้นๆ
2. มีระบบเส้นทาง (Path) ที่จะสามารถแนะนำผู้เรียนได้ว่า ถ้าจะไปเส้นทางนี้ ควรจะเรียนคอร์สใดบ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. มีเว็บไซต์สำหรับให้ผู้ใช้ได้ถามคำถาม หรือเสนอข้อเสนอนั้นๆได้

4. มีระบบ Screencasts ที่ทางผู้ดูแลระบบจะทำการอัปโหลดวิดีโอที่มีเนื้อหาเกี่ยวกับการเขียนโปรแกรม หรือเครื่องมือการเขียนโปรแกรม ต่างๆ มากมาย ให้ผู้เรียนสามารถเลือกดูได้อย่างอิสระ

นอกจากนี้จากการค้นคว้า พบระบบที่มีหลักการทำงานใกล้เคียงกันกับระบบที่ต้องการจะทำ ซึ่งขอยกตัวอย่างมา 2 ระบบ ดังนี้

3.1.2 Netsupport

รูปแบบของ Netsupport จะมีลักษณะดังนี้

1. ลง Netsupport ในเครื่องที่ต้องการใช้งาน
2. หลังจากติดตั้งจะมีหนึ่งเครื่องที่ทำงานเป็นเครื่องแม่ โดยจะสามารถสั่งให้เครื่องลูกทำอะไรก็ได้ตามที่ต้องการ ไม่ว่าจะสั่งให้รีสตาร์ทเครื่อง สั่งปิดเครื่อง สั่งล็อคหน้าจอ เป็นต้น
3. เครื่องแม่สามารถควบคุมการเคลื่อนไหวของเครื่องลูกได้อย่างอิสระอีกด้วย

3.1.3 Teamviewer

รูปแบบของ Teamviewer จะมีลักษณะดังนี้

1. ลง Teamviewer ในเครื่องที่ต้องการใช้งาน
2. จะมีเครื่องผู้ควบคุม และเครื่องเป้าหมาย หากเครื่องเป้าหมายยอมให้ควบคุม จะทำให้เครื่องผู้ควบคุมสามารถเข้ามาควบคุมได้ โดยจะสั่งการได้ทุกอย่าง

3.2 ปัญหาที่พบในระบบปัจจุบัน

3.2.1 เว็บไซต์ Codeschool.com และเว็บไซต์ Codecademy.com

จากการได้มีการลองใช้งาน Codeschool และ Codecademy ทำให้เห็นปัญหาของเว็บไซต์นี้ได้ดังนี้

1. ระบบรองรับเฉพาะภาษาอังกฤษ รวมไปถึงเนื้อหาของคอร์สทั้งหมดจะมีเนื้อหาเป็นภาษาอังกฤษ ซึ่งอาจจะเป็นปัญหาสำหรับคนที่อยากจะเรียนรู้การเขียนโปรแกรม แต่ไม่มีทักษะด้านภาษาอังกฤษ

2. เนื้อหาของเว็บไซต์จะมาจากทางผู้ดูแลระบบเท่านั้น บุคคลอื่นที่ต้องการจะถ่ายทอดความรู้ จะไม่สามารถใช้ประโยชน์จากระบบนี้ได้ อีกทั้งเนื้อหายังมีไม่หลากหลาย 3. ไม่มีระบบสำหรับการอำนวยความสะดวกให้ผู้สอน ถ้าหากผู้สอนต้องการสอนในลักษณะเป็นกลุ่ม เช่น ลักษณะการสอนในห้องเรียน ระบบนี้ไม่สามารถรองรับได้

3.2.2 Netsupport

จากการศึกษาการทำงานของ Netsupport ทำให้เห็นปัญหาและข้อจำกัดดังนี้

1. จำเป็นต้องลง Software ต่างๆที่ละเครื่อง ไม่ว่าจะเป็นตัว Netsupport เอง หรือตัว Software ที่ต้องใช้ในการเรียนการสอน เช่น Compiler และ Database เป็นต้น
2. มีค่า License ที่ค่อนข้างแพง เหมาะสำหรับองค์กรใหญ่ๆเท่านั้น

3.2.3 Teamviewer

จากการลองใช้งาน Teamviewer ทำให้เห็นปัญหาและข้อจำกัดดังนี้

1. สามารถควบคุมได้ที่ละเครื่อง และถ้าต้องการจะเปลี่ยนเครื่องควบคุมจะใช้เวลาามาก เนื่องจาก Teamviewer ไม่ได้ออกแบบมาใช้สำหรับควบคุมหลายๆเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นักพัฒนาโปรแกรมบางกลุ่มเป็นอย่างมาก จึงทำให้มีคนช่วยพัฒนา รวมทั้งคนที่คอยแจ้งปัญหาจากการใช้งาน Docker อย่างมากมาย จากสาเหตุข้างต้น ทำให้การพัฒนา Docker เป็นไปอย่างถูกต้องและรวดเร็ว ในไม่ช้า Docker จะถูกนำมาใช้อย่างแพร่หลาย และทำประโยชน์ได้อย่างมากมาย

4.2 เว็บแอปพลิเคชัน (Web Application)

4.2.1 ส่วนของหน้าแรก

หน้านี้ คือหน้าแรกที่ผู้ใช้จะได้เห็นรายละเอียดภาพรวม เกี่ยวกับการทำงานของเว็บไซต์



รูปที่ 6 หน้าแรกของเว็บไซต์ Coder Planet

4.2.1 ผู้สอน

1. หน้าสร้างห้องเรียน Coder-Space

หน้านี้ เป็นหน้าสร้างห้องเรียน Coder-Space ผู้สอนต้องกรอกรายละเอียดดังภาพด้วยบน เมื่อ กดปุ่มสร้างห้องแล้ว ระบบจะทำการสร้างรหัสมาให้ ผู้สอนต้องนำรหัสนี้ไปแจกจ่ายผู้เรียน ผู้เรียนจำเป็นต้องกรอกรหัสนี้ จึงสามารถเข้าเรียนห้อง Coder-Space ที่ผู้สอนสร้างขึ้นมาได้



รูปที่ 7 หน้าสร้างห้องเรียน Coder-Space

2. หน้า Editor ของผู้สอน

หน้านี้ ผู้สอนสามารถเขียนโค้ดลงไป ใน Editor และสามารถประมวลผลโค้ดผ่านระบบนี้ได้ แถบด้านข้างจะแสดง Editor ของผู้เรียนที่อยู่ในห้องเรียนนี้ทั้งหมด ผู้สอนสามารถกดเข้าไปดู แก้ไข และประมวลผลโค้ดใน Editor ของผู้เรียนคนใดที่สนใจได้



รูปที่ 8 สร้างแบบทดสอบสำหรับตอบคำถาม

3. หน้าสร้างแบบทดสอบสำหรับตอบคำถาม

ผู้สอนต้องสร้างโจทย์สำหรับทำแบบทดสอบ และเขียนคำตอบลงตัวเลือกทั้ง 4 ข้อ โดยจำเป็นต้องทำเครื่องหมายลงในตัวเลือกที่เป็นคำตอบที่ถูกต้องด้วย



รูปที่ 9 สร้างแบบทดสอบสำหรับตอบคำถาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเดิมหรือข้ามไปยังข้อถัดไปได้ จนกว่าจะหมดเวลาการทำแบบทดสอบ



รูปที่ 13 หน้าการทำแบบทดสอบของระบบ Grader

5. สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

โครงการนี้มีรูปแบบเป็นเว็บแอปพลิเคชัน ที่ช่วยทำให้การเรียนและการสอนเขียนโปรแกรมมีประสิทธิภาพมากขึ้น ซึ่งในส่วนของฝั่งเซิร์ฟเวอร์พัฒนาขึ้นโดยภาษา Node.js และใช้เทคโนโลยี Docker เข้ามาช่วยในการทำให้เว็บมีประสิทธิภาพมากยิ่งขึ้น

การทำงานของระบบนี้ จะแบ่งหน้าที่ออกเป็นสองหน้าที่ คือ ผู้เรียน และผู้สอน โดยที่ผู้สอนจะสามารถสร้างห้องเรียนที่จะให้ผู้เรียนเข้ามาเรียนได้ เมื่อผู้เรียนเข้ามาเรียนจะพบกับสองหน้าจอ คือ หน้าจอของตนเอง และหน้าจอของผู้สอน ซึ่งในขณะที่สอนอยู่ผู้เรียนจะดูในส่วนหน้าจอของผู้สอน และเมื่อผู้สอนต้องการให้ผู้เรียนลองฝึกเขียนโปรแกรมด้วยตนเอง ผู้เรียนจะสามารถเขียนโปรแกรมในส่วนหน้าจอของตนเองได้ ในขณะที่ผู้เรียนเขียนโปรแกรมอยู่นั้น หากผู้สอนต้องการดูโค้ดของผู้เรียน ผู้สอนจะสามารถเข้าไปดู และสามารถแก้ไขโค้ดของผู้เรียนได้ โดยการแสดงผลทั้งหมดที่กล่าวมานั้น

จะเป็นการแสดงผลแบบ Real time ผ่านเทคโนโลยีที่ชื่อว่า Websocket

นอกจากนี้ยังมีระบบแบบทดสอบ ผู้สอนจะสามารถสร้างแบบทดสอบโดยกำหนดคำถามและตัวเลือกคำตอบเองได้ อีกทั้งยังสามารถตั้งช่วงเวลาในการทำแบบทดสอบได้อีกด้วย หากผู้เรียนไม่ได้เริ่มทำแบบทดสอบภายในระยะเวลาที่กำหนด ผู้เรียนก็ไม่มีสิทธิ์ทำแบบทดสอบได้อีก เมื่อผู้เรียนทำแบบทดสอบเสร็จเรียบร้อยแล้ว จะสามารถดูผลคะแนนของตัวเองและทราบคำตอบที่ถูกต้องได้ และผู้สอนสามารถดูคะแนนทั้งหมดของผู้เรียนทั้งคลาสได้อีกทั้งยังมีระบบ Grader เป็นระบบที่ให้ผู้เรียนเขียนโค้ดและอัปโหลดขึ้นเพื่อให้ระบบของเราทำการตรวจสอบผลลัพธ์อัตโนมัติ โดยจะมีเทสเคสที่แตกต่างกันไปตามที่ผู้สอนกำหนด และผู้สอนสามารถดูผลคะแนนของเด็กได้ สามารถดูได้ว่าผู้เรียนผ่านเทสเคสใดบ้าง เพื่อประเมินทั้งภาพรวมและรายบุคคลว่าผู้เรียนเข้าใจมากน้อย

เอกสารอ้างอิง

[1] Node.js Foundation “Node.js v5.1.0 Documentation.”

[cited 2016 Apr 4]

[Online].Available from:

<https://nodejs.org/api/>

[2] Socket.io Development Team

“Socket.io Documentation.”

[cited 2016 Apr 4]

[Online].Available from:

<http://socket.io/docs/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[3] Express.js Development Team

“Express 4.x API.”

[cited 2016 Apr 4]

[Online].Available from:

<http://expressjs.com/4x/api.html>

[4] Docker Engineering Team

“Docker Documentation.”

[cited 2016 Apr 4]

[Online].Available from:

<https://docs.docker.com/>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้