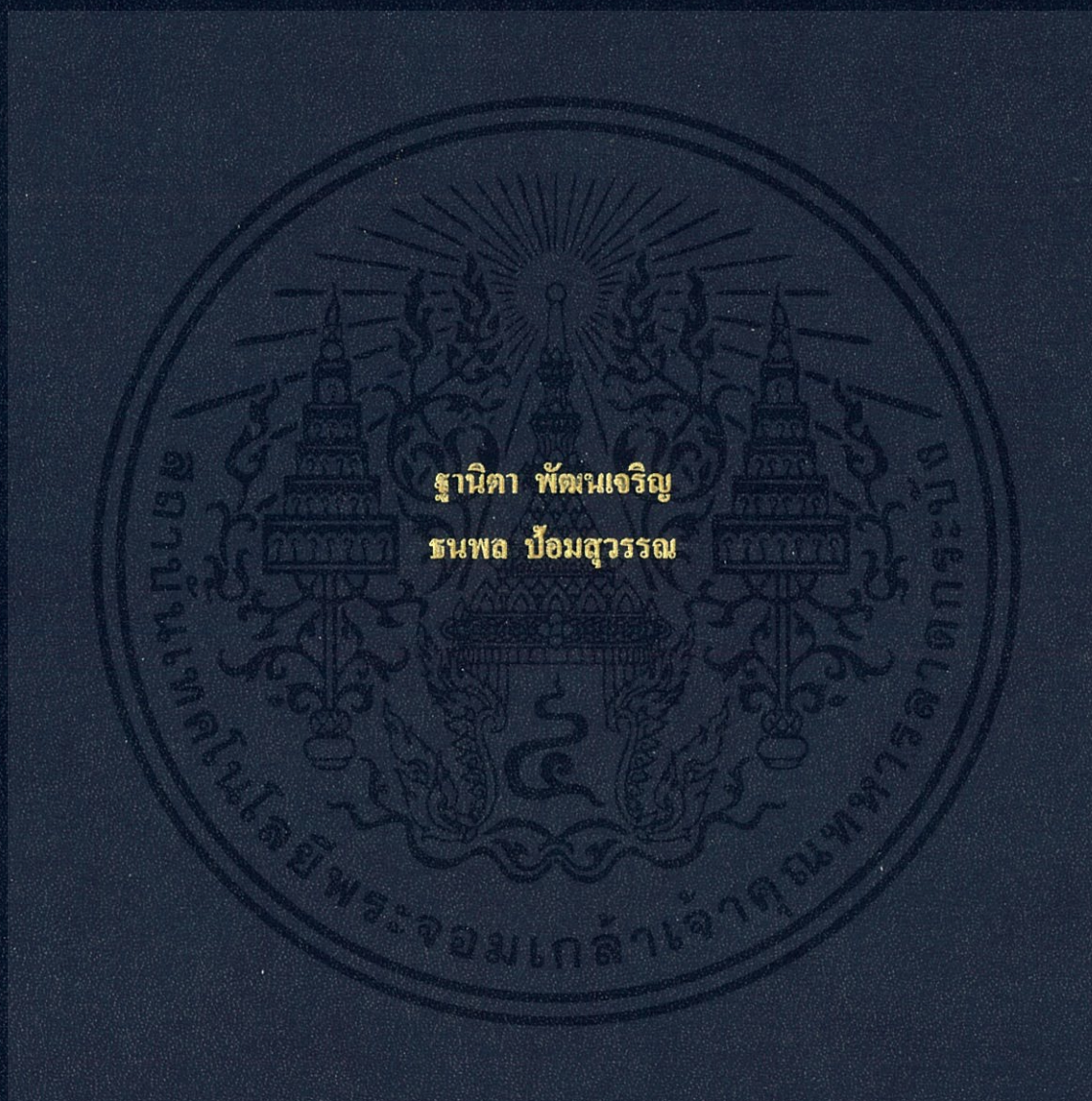


ระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซี
PROGRAMMING ANALYSIS AND PRACTICING SYSTEM FOR
C PROGRAMMING LANGUAGE



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

ระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซี

PROGRAMMING ANALYSIS AND PRACTICING SYSTEM FOR
C PROGRAMMING LANGUAGE



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2559

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบวิเคราะห์และฝึกฝนการเขียน โปรแกรมภาษาซี

PROGRAMMING ANALYSIS AND PRACTICING SYSTEM FOR C PROGRAMMING
LANGUAGE

ผู้จัดทำ

1. นางสาวฐานิตา พัฒนเจริญ รหัสนักศึกษา 56010332
2. นายชนพล ป้อมสุวรรณ รหัสนักศึกษา 56010525


อาจารย์ที่ปรึกษา
(ผศ. ดร.ชุติเมษณ์ ศรีนิลตา)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซี

นางสาวฐานิตา พัฒนเจริญ 56010332

นายชนพล ป้อมสุวรรณ 56010525

ผศ.ดร.ชุตินเมษฐ์ ศรีนิลทา อาจารย์ที่ปรึกษา

ปีการศึกษา 2559

บทคัดย่อ

ระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซี เป็นเว็บแอปพลิเคชันที่จัดทำขึ้นเพื่อช่วยเหลือผู้เรียนให้สามารถฝึกฝนการเขียนโปรแกรมได้ด้วยตัวเองโดยระบบจะช่วยตรวจสอบเบื้องต้นให้ว่าโปรแกรมที่เขียนขึ้นมานั้นทำงานได้ถูกต้องตามจุดประสงค์ที่ตั้งไว้หรือไม่และยังช่วยเหลืออาจารย์ผู้สอนในการประเมินผลการเรียนของนักศึกษาแต่ละคนให้ง่ายขึ้น โดยอาจารย์สามารถตั้งโจทย์และกำหนดเกณฑ์การผ่านของโจทย์แต่ละข้อได้ด้วยตัวเอง ซึ่งจะมีฟีเจอร์พิเศษที่สามารถตรวจได้ว่าฟังก์ชันนั้นถูกเขียนในลักษณะ recursion หรือไม่และสามารถนับจำนวน nested loop (for, while, do while) ที่ใช้ในฟังก์ชันได้ หากซอร์สโค้ดของนักศึกษาคนใดไม่เป็นไปตามเกณฑ์ที่กำหนดจะมีการแจ้งเตือนไปที่อาจารย์เพื่อที่อาจารย์จะเข้าไปดูซอร์สโค้ดของนักศึกษาคนนั้นแล้วนำมาประเมินผลความเข้าใจของนักศึกษาต่อไป

ระบบจะตรวจการเขียนโปรแกรมโดยคำนึงถึงความถูกต้องของผลลัพธ์ เวลาที่ใช้ในการประมวลผล ทรัพยากรที่ใช้ และเป็นไปตามเกณฑ์ที่อาจารย์กำหนด นอกจากนี้ระบบจะมี dashboard ที่จะแสดงสถิติต่าง ๆ ของนักศึกษาแต่ละคนเช่นสถิติการส่งงาน คะแนนรวม จำนวนครั้งที่ใช้ในการส่งซอร์สโค้ดแต่ละข้อ เป็นต้น เพื่อที่อาจารย์จะสามารถนำไปวิเคราะห์ความรู้และความเข้าใจของนักศึกษาแต่ละคนได้โดยคณะผู้จัดทำมีความหวังว่าจะสามารถนำ “ระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซี” ไปใช้ในวิชาปฏิบัติการของวิชาที่มีการเขียนโปรแกรมของภาควิชาวิศวกรรมคอมพิวเตอร์ต่อไปได้

PROGRAMMING ANALYSIS AND PRACTICING SYSTEM FOR C PROGRAMMING LANGUAGE

Ms. Thanita Pattanacharoen 56010332

Mr. Thanaphol Pomsuwan 56010525

Asst.Prof.Dr.Chutimet Srinilta Advisor

Academic Year 2016

ABSTRACT

Programming analysis and practicing system for C programming language is web application which helps students to practice coding the program by themselves. The system will check their source code, against objectives prior set by teacher. The system helps teacher evaluate each student easier. Teacher can set the questions and criterion for passing. Special features allows us to check whether a function is written in recursive style. The system can count number of nested loops (for, while, do while) in a function. If source code does not comply with the passing criterion, the system will alert teacher to check that source code again to re-evaluating the understanding of student.

System will check the code by the correctness of the answers, processing time, resources utilization and matched the criterion. Furthermore, the system also has a dashboard showing statistics of each student is performance such as the statistics of submitting, total scores, the number of submitting each question and so on, for helping teacher to evaluate about knowledge and understanding of each student. However, we hope that “Programming analysis and practicing system for C programming language” can be applied to practical term in every subject which is important to learn about coding in faculty of Computer Engineering.

กิตติกรรมประกาศ

ขอขอบคุณ ผศ.ดร.ชุตินเมษย์ ศรีนิลทาอาจารย์ที่ปรึกษาที่คอยช่วยเหลือ ดูแลเอาใจใส่ ให้คำแนะนำชี้แนะทางการปรับปรุงแก้ไขในโครงงานนี้มาโดยตลอดจนปริญญาณิพนธ์นี้สำเร็จลงได้ด้วยดี

ขอขอบคุณ ดร.ธนัญชัย ศรีภาค และ ผศ.ชนา หงษ์สุวรรณที่ให้แนวคิดในการพัฒนาโครงงานนี้ และขอขอบคุณ รศ. ดร.บุญธีร์ เครือตราชู และ รศ.กฤตวัน ศิริบุญรณ์ ผู้ควบคุมการปฏิบัติการห้องปฏิบัติการวิชา Data structure & algorithm รวมถึงอาจารย์คณัฐ ตั้งสินานนท์และอาจารย์ประสาร ตั้งติสานนท์ อาจารย์ผู้ควบคุมการปฏิบัติการวิชา Computer Programming 1 ที่อนุญาตให้คณะผู้จัดทำได้นำระบบเข้าไปทำการทดสอบ และอนุญาตให้ใช้ห้องปฏิบัติการสำหรับจัดการแข่งขันการเขียนโปรแกรมภาษาซี

ขอขอบพระคุณบิดา และมารดาของบรรดาข้าพเจ้าที่ให้การสนับสนุนในทุกด้านมาโดยตลอดและคอยให้กำลังใจในยามเหน็ดเหนื่อยอ่อนล้า และท้อแท้จากการเรียนและการทำโครงการจนกระทั่งทุกอย่างสำเร็จลุล่วงไปได้

สุดท้ายนี้ขอขอบคุณรุ่นพี่และเพื่อนๆทุกคนที่คอยให้คำปรึกษาแลกเปลี่ยนความคิดเห็นเป็นกำลังใจ และช่วยเหลือกันในเรื่องต่างๆตลอดจนทุกท่านที่ให้ความสนใจต่อการดำเนินงานของโครงงานนี้ด้วย

ฐานิตา พัฒนเจริญ
ธนพล ป้อมสุวรรณ

สารบัญ

	หน้า
บทคัดย่อ.....	I
ABSTRACT.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	2
1.1 ความเป็นมา.....	2
1.2 วัตถุประสงค์ของโครงการ.....	2
1.4 ขอบเขตของโครงการ.....	3
1.5 ขั้นตอนการดำเนินงาน.....	4
1.6 ข้อยกเว้นโปรแกรม.....	5
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	5
2.1 ความหมายของเว็บแอปพลิเคชัน.....	5
2.2 Static code analysis.....	10
2.3 ตัวอย่างระบบตรวจโค้ดโปรแกรมมิ่งในปัจจุบัน.....	10
2.4 ไคลเอนต์ - เซิร์ฟเวอร์ โมเดล.....	12
2.5 sandbox.....	13
2.6 Bootstrap.....	14
บทที่ 3 การออกแบบและพัฒนา.....	15
3.1 ความต้องการของระบบ และลักษณะสำคัญ.....	15
3.2 ภาพรวมของระบบ.....	17
3.3 การออกแบบอัลกอริทึม.....	21
3.4 รูปแบบและการใช้งานโปรแกรม.....	24

สารบัญ (ต่อ)

	หน้า
บทที่ 4 การทดลองและผลการทดลอง.....	32
4.1 การทดลองระบบครั้งที่ 1.....	32
4.2 การทดลองระบบตรวจคุณสมบัติฟังก์ชัน.....	38
4.3 การทดลองระบบครั้งที่ 2.....	41
4.4 การแข่งขันการเขียนโปรแกรมภาษาซี.....	46
บทที่ 5 สรุป.....	55
5.1 สรุป.....	55
5.2 ปัญหาและแนวทางการแก้ไข.....	55
5.3 แนวทางการพัฒนาต่อ.....	59

สารบัญตาราง

ตาราง	หน้า
4.1 อธิบายสัญลักษณ์และความหมาย.....	39
4.2 ผลการทดลองและความผิดพลาดที่พบ.....	40



สารบัญรูป

รูป	หน้า
รูปที่ 2.1 ส่วนประกอบของเว็บแอปพลิเคชัน	6
รูปที่ 2.2 การทำงานรวมกันของ Apache PHP และ MySQL.....	8
รูปที่ 2.4 ตัวอย่างหน้าใช้งานของอาจารย์	11
รูปที่ 2.5 ตัวอย่างการทำโจทย์ของนักศึกษา.....	11
รูปที่ 3.1 Activity Diagram : การส่งโค้ดของนักศึกษา	17
รูปที่ 3.2 Activity diagram : การเพิ่มโจทย์ของอาจารย์	18
รูปที่ 3.3 Data flow diagram level 0	19
รูปที่ 3.4 Data flow diagram level 1	20
รูปที่ 3.5 การคอมไพล์และประมวลผล	21
รูปที่ 3.6 การวิเคราะห์โค้ดโดยใช้ Static code analysis	22
รูปที่ 3.7 หน้าเข้าสู่ระบบ (Login).....	24
รูปที่ 3.8 ตัวอย่างหน้าเข้าสู่ระบบ (Login) เมื่อใส่ Username และ Password	24
รูปที่ 3.9 หน้า Dashboard	25
รูปที่ 3.10 ตัวอย่างการแสดงผลซอร์สโค้ด	25
รูปที่ 3.11 สัญลักษณ์ต่างๆ.....	26
รูปที่ 3.12 หน้าแสดงรายละเอียดการส่งครั้งที่ 220	26
รูปที่ 3.13 หน้า Problem Management	27
รูปที่ 3.14 แสดงโจทย์ของแถบ A.....	27
รูปที่ 3.15 Menu bar.....	28
รูปที่ 3.16 เพิ่ม โจทย์ปัญหา ชั้นที่ 1	28
รูปที่ 3.17 เพิ่ม test case ชั้นที่ 2	29
รูปที่ 3.18 ตัวอย่างการแก้ไข Problem	29
รูปที่ 3.19 หน้าหลัก.....	30
รูปที่ 3.20 แสดงรายละเอียดโจทย์เป็น pdf	30
รูปที่ 3.21 ตัวอย่างการส่งซอร์สโค้ด.....	31
รูปที่ 3.22 หน้า Submission	32

สารบัญญรูป (ต่อ)

รูป	หน้า
รูปที่ 3.23 หน้า Information.....	32
รูปที่ 4.1 หน้า log in ของระบบ.....	32
รูปที่ 4.2 หน้าส่งชอร์สโค้ดของนักศึกษา.....	33
รูปที่ 4.3 หน้าประวัติการส่งงานของนักศึกษา.....	33
รูปที่ 4.4 หน้าดูชอร์สโค้ดและรายละเอียดการส่งงาน.....	34
รูปที่ 4.5 แบบสอบถามความพึงพอใจภาคเรียนที่ 2/2558 (ส่วนที่ 1).....	34
รูปที่ 4.6 แบบสอบถามความพึงพอใจภาคเรียนที่ 2/2558 (ส่วนที่ 2).....	35
รูปที่ 4.7แบบสอบถามความพึงพอใจภาคเรียนที่ 2/2558(ส่วนที่ 3).....	35
รูปที่ 4.8 ผลที่ได้จากการตอบแบบสอบถามข้อ 1.....	36
รูปที่ 4.9 ผลที่ได้จากการตอบแบบสอบถามข้อ 2.....	36
รูปที่ 4.10 ผลที่ได้จากการตอบแบบสอบถามข้อ 3.....	37
รูปที่ 4.11 ผลที่ได้จากการตอบแบบสอบถามข้อ 4.....	37
รูปที่ 4.12 ตัวอย่างชอร์สโค้ดที่ 1.....	39
รูปที่ 4.13 ตัวอย่างผลการรันจากชอร์สโค้ดที่ 1.....	40
รูปที่ 4.14 ตัวอย่างชอร์สโค้ดที่ 2.....	40
รูปที่ 4.15 ตัวอย่างผลการรันชอร์สโค้ดที่ 2.....	40
รูปที่ 4.16 ตัวอย่างโค้ดในการแก้ปัญหา.....	42
รูปที่ 4.17 ภาพบรรยากาศการทดลองระบบครั้งที่ 2.....	43
รูปที่ 4.18 ภาพบรรยากาศการทดลองระบบครั้งที่ 2.....	43
รูปที่ 4.19 ภาพบรรยากาศการทดลองระบบครั้งที่ 2.....	44
รูปที่ 4.20 ภาพบรรยากาศการทดลองระบบครั้งที่ 2.....	44
รูปที่ 4.21 ภาพบรรยากาศการทดลองระบบครั้งที่ 2.....	45
รูปที่ 4.22 ภาพบรรยากาศการทดลองระบบครั้งที่ 2.....	45
รูปที่ 4.23 ภาพบรรยากาศการแข่งขัน.....	47
รูปที่ 4.23 ภาพบรรยากาศการแข่งขัน.....	48
รูปที่ 4.24 ภาพบรรยากาศการแข่งขัน.....	48

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 4.25 ภาพบรรยากาศการแข่งขัน.....	49
รูปที่ 4.26 ภาพบรรยากาศการแข่งขัน.....	49
รูปที่ 4.27 ภาพบรรยากาศการแข่งขัน.....	50
รูปที่ 4.28 ทีมผู้ชนะการแข่งขัน(ระดับยาก).....	50
รูปที่ 4.28 ทีมผู้ชนะการแข่งขัน(ระดับปานกลาง).....	51
รูปที่ 4.22 แบบสอบถามความพึงพอใจ (ส่วนที่ 1).....	52
รูปที่ 4.23 แบบสอบถามความพึงพอใจ (ส่วนที่ 2).....	52
รูปที่ 4.24 แบบสอบถามความพึงพอใจ (ส่วนที่ 3).....	52
รูปที่ 4.25 ผลการตอบแบบสอบถามความพึงพอใจข้อที่ 1.....	53
รูปที่ 4.26 ผลการตอบแบบสอบถามความพึงพอใจข้อที่ 2.....	53
รูปที่ 4.27 ผลการตอบแบบสอบถามความพึงพอใจข้อที่ 3.....	54
รูปที่ 5.1 Interface แบบที่ 1.....	56
รูปที่ 5.2 Interface แบบที่ 2.....	57
รูปที่ 5.3 Interface แบบที่ 3.....	58
รูปที่ 5.4 Interface ที่เลือกใช้จริง.....	59

บทที่ 1

บทนำ

1.1 ความเป็นมา

“ระบบวิเคราะห์และฝึกฝนการเขียน โปรแกรมภาษาซี” เกิดจากแนวคิดที่อยากจะสร้างเว็บ แอปพลิเคชันที่จะช่วยจัดการห้องปฏิบัติการ โปรแกรมมิ่ง เนื่องจากปัจจุบันนี้การเรียนการสอนใน หลายวิชามีส่วนเกี่ยวข้องกับการเขียน โปรแกรมเป็นจำนวนมากซึ่งมักจะดำเนินการสอนควบคู่กัน ไประหว่างภาคทฤษฎีและภาคปฏิบัติซึ่งการสอนภาคปฏิบัตินั้นจะให้นักศึกษาทดลองทำ แบบฝึกหัดด้วยตนเอง โดยอาจารย์ผู้สอนคอยให้คำแนะนำ ให้คำปรึกษาและตรวจแบบฝึกหัดทั้งนี้ ปัญหาเกิดขึ้นเมื่อจำนวนนักศึกษาเพิ่มมากขึ้นจากการสอบถามนักศึกษาและจากประสบการณ์ที่ ผ่านมาทางคณะผู้จัดทำได้พบปัญหาในส่วนของนักศึกษาดังนี้ดังต่อไปนี้

- 1) ไม่แน่ใจว่าโค้ดที่เขียนนั้นสามารถครอบคลุมได้ทุกกรณีที่อาจารย์ต้องการหรือไม่ เนื่องจาก ตัวนักศึกษาเองอาจคิด test case ได้ไม่ครอบคลุมมากพอ
- 2) ไม่สามารถทราบได้ว่าโค้ดที่เราเขียนนั้นดีหรือยังในแง่ performance เนื่องจาก ไม่มีตัวชี้วัด ให้เห็น
- 3) ใช้เวลารอสั่งโค้ดค่อนข้างนาน
- 4) ไม่มีการจัดการลำดับ(Queue)การส่งอย่างเป็นระบบ

นอกจากปัญหาที่นักศึกษาพบนั้น คณะผู้จัดทำได้ไปสอบถาม ดร.ธนัญชัย ตีรภาค และผศ.ธนา หงษ์สุวรรณ ผู้สอนวิชา Programing 1 ภาคการศึกษา 1/2016 พบว่าอาจารย์ผู้สอนต้องการระบบที่ สามารถให้ข้อมูลเพื่อช่วยในการประเมินความสามารถของนักศึกษาแต่ละคนได้เพื่อที่จะได้ให้ ความสนใจกับนักศึกษาที่อ่อนหรือยังไม่เข้าใจในบทเรียนดีนักได้มากขึ้น ซึ่งอาจารย์สามารถตั้ง โจทย์ พร้อมทั้งกำหนดเกณฑ์การผ่านของข้อนั้น ๆ และระบบสามารถช่วยตรวจโค้ดเบื้องต้นก่อน ได้ช่วยให้อาจารย์ตรวจได้ง่ายขึ้น

ระบบที่ช่วยเหลือเรื่องการเขียน โปรแกรมลักษณะนี้มีบริการให้ใช้อยู่หลายเว็บแต่ต้องเสีย ค่าบริการซึ่งโครงการนี้จึงเกิดขึ้นเพื่อที่จะได้มีระบบของตัวเองไว้ใช้โดยไม่เสียค่าใช้จ่าย

1.2 วัตถุประสงค์ของโครงการ

วัตถุประสงค์ของโครงการนี้จัดทำขึ้นเพื่อที่จะช่วยเหลือการจัดการภายในห้องปฏิบัติการ ให้ เป็นระบบมากยิ่งขึ้น สามารถแก้ปัญหาที่กล่าวมาข้างต้นและสนับสนุนการเรียนภาคปฏิบัติใน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องปฏิบัติการให้ดีขึ้น ช่วยเหลือนักศึกษาให้สามารถฝึกฝนการเขียนโค้ดด้วยตัวเอง และช่วยประหยัดเวลาให้อาจารย์ในการตรวจโค้ดของเด็กแต่ละคน เนื่องจากโปรแกรมจะตรวจเบื้องต้นให้ก่อนว่าโค้ดสามารถทำงานได้ผลลัพธ์ที่ถูกต้องตรงตามจุดประสงค์และยังสามารถตรวจได้ว่าโค้ดเป็น recursion หรือ ไม่ หรือใช้ loop ในการแก้ปัญหาหรือไม่ สำหรับข้อที่อาจารย์ต้องการให้แก้ไขด้วย recursion หรือ loop

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้รับความรู้ความเข้าใจในเรื่องของเว็บเทคโนโลยี, การสื่อสารระหว่างไคลเอนต์และเซิร์ฟเวอร์
- 2) ได้รับความรู้ความเข้าใจถึงกระบวนการพัฒนาระบบตั้งแต่การวางแผน, การวิเคราะห์, การออกแบบจนถึงการสร้างและการนำระบบไปใช้งาน
- 3) ได้ความรู้เพิ่มเติมในเรื่องการเขียนเว็บแอปพลิเคชันโดยใช้ภาษาต่าง ๆ เช่น Ruby, HTML, C++ เป็นต้น
- 4) สามารถนำระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซีไปใช้กับภาคปฏิบัติในวิชาต่าง ๆ ของภาควิชาวิศวกรรมคอมพิวเตอร์ได้

1.4 ขอบเขตของโครงการ

ระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซีแบ่งผู้ใช้งานออกเป็น 2 แบบได้แก่อาจารย์และนักศึกษาโดยผู้ใช้ที่เป็นอาจารย์สามารถตั้งโจทย์ ตั้ง test case และกำหนดเกณฑ์การผ่านของโจทย์ข้อนั้น ๆ ได้ซึ่งจะมีความพิเศษของระบบที่สามารถตรวจได้ว่าฟังก์ชันนั้นเขียนแบบ recursion หรือ ไม่ และสามารถนับจำนวนลูปที่ใช้ในฟังก์ชันได้ หากโค้ดของนักศึกษาค้นใดไม่เป็นไปตามเกณฑ์ที่กำหนดจะมีการแจ้งเตือนไปที่อาจารย์ เพื่อที่อาจารย์จะสามารถเข้าไปตรวจสอบโค้ดของนักศึกษาเพิ่มเติมได้ นอกจากนั้นยังมีส่วนเสริมอื่น ๆ เช่นมี dashboard ที่จะแสดงสถิติต่าง ๆ ของนักศึกษาแต่ละคนเช่นสถิติการส่งงาน คะแนนรวม จำนวนครั้งที่ใช้ในการส่งซอร์สโค้ดแต่ละข้อ เป็นต้น เพื่อที่อาจารย์จะสามารถนำไปวิเคราะห์ความรู้และความเข้าใจของนักศึกษาแต่ละคนได้

ส่วนผู้ใช้ที่เป็นนักศึกษาจะสามารถฝึกฝนการเขียนโปรแกรมได้ด้วยตัวเอง โดยนักศึกษาสามารถอัปโหลดซอร์สโค้ดของโปรแกรมเข้าสู่ระบบเพื่อให้ระบบนำไปคอมไพล์และตรวจ

ผลลัพธ์พร้อมทั้งนำไปเทียบกับเกณฑ์ที่อาจารย์กำหนดไว้และมีระบบการเข้าคิวส่งงานเพื่อความ เป็นระเบียบมากยิ่งขึ้นนอกจากนั้นยังมี dashboard แสดงประวัติการส่งงาน และแจ้งเตือนงานต่าง ๆ

1.5 ขั้นตอนการดำเนินงาน

เป้าหมาย	Due Date
คิดหัวข้อโครงการงาน	เทอม 2/2558
ศึกษาสถาปัตยกรรมของเว็บแอปพลิเคชัน	เทอม 2/2558
ออกแบบวางโครงสร้างระบบ	เทอม 2/2558
ติดตั้งโปรแกรมที่จำเป็นทั้งหมด	เทอม 2/2558
ออกแบบหน้าเว็บ	เทอม 2/2558
Implement : ส่วน Authentication, Check Result และ Database	เทอม 2/2558
นำระบบไปทดลองใช้กับวิชา Data Structure and Algorithm	เทอม 2/2558
เก็บ Feedback	เทอม 2/2558
วิเคราะห์ปัญหาที่เกิดขึ้น	เทอม 2/2558
ส่งรายงานความคืบหน้าครั้งที่ 1	8/8/2559
เริ่มใช้เครื่องมือช่วยในการจัดการงาน(Project management tools): Asana	9/8/2559
อ่านรายงานโปรเจกต์เก่า (ระบบการจัดการการเขียนโปรแกรม)เพื่อ ศึกษาปัญหาที่เกิดขึ้นในโปรเจกต์นั้น ๆ	18/8/2559
ออกแบบโครงสร้างระบบที่จะอธิบายถึงส่วนของการตรวจโค้ดนักศึกษาเพิ่มเติม และส่วนของการประมวลผลข้อมูล(Admin dashboard)	18/8/2559
ส่งรายงานความคืบหน้าครั้งที่ 2	22/8/2559
เก็บ Requirement เพิ่มเติมจากอาจารย์ธัญชัย ศรีภาคและอาจารย์ธนาหงษ์ สุวรรณ	30/8/2559
ส่งรายงานความคืบหน้าครั้งที่ 3	5/9/2559
สรุปขอบเขตของโครงการงาน	3/10/2559
ส่งรายงานความคืบหน้าครั้งที่ 4	3/10/2559
ศึกษาการใช้ Microsoft Word เพิ่มเติม	15/10/2559
เขียนรายงาน	7/11/2559
ศึกษาข้อมูลเกี่ยวกับ Static code analysis	13/10/2559
ส่งรายงานความคืบหน้าครั้งที่ 5	17/10/2559
สร้าง Feature สำหรับตรวจคุณสมบัติของโค้ด C, C++ โดยอาศัยหลักการ Static	31/10/2559

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

code analyzer เพื่อตรวจพฤติกรรมการทำงานฟังก์ชัน	
Implement : ส่วน Check Condition	31/10/2559
Implement : ส่วน Dashboard	31/10/2559
ส่งรายงานความคืบหน้าครั้งที่ 6	31/10/2559
รวมระบบทั้งหมด	30/11/2559
เตรียม Presentation	5/12/2559
ทดสอบระบบ	ก่อนเทอม 2/2559
เพิ่มพื้นที่จัดเก็บของ virtual machine เพื่อให้ลง Package กับเพิ่มขนาดของ Database ได้มากขึ้น (สำรองในกรณีนักศึกษาส่งโค้ดมาเป็นจำนวนมาก)	ก่อนเทอม 2/2559
เพิ่มคู่มือการใช้งานในหน้าเว็บสำหรับผู้เริ่มต้น	ก่อนเทอม 2/2559
ส่งรายงานความคืบหน้าครั้งที่ 1	23/1/2560
ทดสอบระบบส่วน Identify	26/1/2560
ส่งรายงานความคืบหน้าครั้งที่ 2	1/2/2560
วางแผนและเตรียมจัดการแข่งขันการเขียนโปรแกรมภาษาซี	13/2/2560
ส่งรายงานความคืบหน้าครั้งที่ 3	15/2/2560
เตรียมการแข่งขันการเขียนโปรแกรมภาษาซี	3/3/2560
ส่งรายงานความคืบหน้าครั้งที่ 4 (เทอม 2)	6/3/2560
นำระบบไปทดลองใช้ในวิชา Computer Programming 1	9/3/2560
จัดการแข่งขันการเขียนโปรแกรมภาษาซี	9/3/2560
ปรับปรุง UI / Front end	20/3/2560
ปรับปรุงรูปเล่มรายงานให้เสร็จสมบูรณ์	24/3/2560
ส่งเล่มรายงาน	28/3/2560
เตรียม Presentation	30/3/2560

1.6 ข้อจำกัดโปรแกรม

- 1) เป็นโปรแกรมออฟไลน์ ใช้สำหรับเครือข่ายภายใน (Local Network)
- 2) ระบบตรวจเช็คผลการรัน ได้เฉพาะโปรแกรมที่เขียนด้วยภาษาซีและซีพลัสพลัสเท่านั้น
- 3) สามารถตรวจนับ Loop ได้เฉพาะชนิด for, while และ do while เท่านั้น
- 4) สามารถตรวจ Algorithm ได้เฉพาะว่าเป็นแบบ Recursion หรือไม่เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5) ระบบจะแจ้งเตือนเมื่อโค้ดของนักศึกษาไม่เป็นไปตามเกณฑ์ที่อาจารย์กำหนด
- 6) นักศึกษาจะต้องรับอินพุตและแสดงเอาต์พุตตามรูปแบบที่กำหนดเท่านั้น



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 ความหมายของเว็บแอปพลิเคชัน

เว็บแอปพลิเคชันหรือโปรแกรมประยุกต์บนเว็บคือเว็บที่ถูกพัฒนาขึ้นเพื่อตอบสนองต่องานบางอย่าง สามารถเข้าใช้งานได้ผ่านทาง Web Browser โดยไม่ต้องติดตั้งโปรแกรมอื่น ๆ เพิ่มเติม

2.1.1 ทำไมถึงใช้เว็บแอปพลิเคชัน

คนส่วนใหญ่มักจะคุ้นเคยกับการใช้งาน โปรแกรมพวก Microsoft Office ที่ประกอบด้วย Word ที่สำหรับพิมพ์เอกสาร Excel สำหรับสร้างตารางคำนวณ โปรแกรมพวกนี้เราจะเรียกมันว่า Desktop Application ซึ่งโปรแกรมจะต้องถูกติดตั้งบนเครื่องคอมพิวเตอร์จึงจะสามารถใช้งานได้ หรือโปรแกรมสำหรับงานบัญชี ที่บางหน่วยงานติดตั้งที่เครื่องคอมพิวเตอร์เป็นลักษณะ Client-Server Application โดยเก็บฐานข้อมูลไว้ที่เซิร์ฟเวอร์ (server) และติดตั้งตัวโปรแกรมบัญชีที่เครื่องใช้งาน (client) ซึ่งตอบสนองความต้องการเพิ่มขึ้นในด้าน Multi-User หรือใช้งานพร้อม ๆ กันได้หลายคน โดยใช้ฐานข้อมูลเดียวกัน เก็บฐานข้อมูลไว้ที่ส่วนกลาง

เทคโนโลยี Desktop Application ไม่สามารถตอบสนองความต้องการในเรื่องการบริหารจัดการได้ โดยเฉพาะการทำธุรกิจที่ต้องปรับเปลี่ยนไปตลอดเวลา ข้อมูลมีการเคลื่อนไหวตลอดเวลา เพื่อตอบสนองภาวะตลาดที่แปรเปลี่ยน ระบบ Client-Server Application ตัวโปรแกรมมีความซับซ้อน ต่อการแก้ไข และการ Upgrade ทำได้ยุ่งยาก อย่างกรณี หากต้องการ Upgrade หรือเพิ่มคุณสมบัติเพิ่มเติมให้กับแอปพลิเคชันที่ตัวเซิร์ฟเวอร์ต้องทำการหยุดระบบทั้งหมด และเมื่อ Upgrade ที่เซิร์ฟเวอร์แล้ว ก็จำเป็นต้อง Upgrade ที่ Client ด้วย หากระบบมีผู้ใช้งานจำนวนมาก จะยิ่งเพิ่มความยุ่งยากมากขึ้น

นอกจากนั้นเครื่อง Client แต่ละเครื่องมีความหลากหลายและแตกต่างกัน เช่น OS (Operating System) ที่ต่างกัน สเปคเครื่องที่แตกต่างกัน ซึ่งหากการ Upgrade แล้วมีความจำเป็นต้องใช้สเปคเครื่องที่สูงขึ้นที่ฝั่ง Client จำเป็นต้อง Upgrade ตัวเครื่องคอมพิวเตอร์ตามไปด้วย

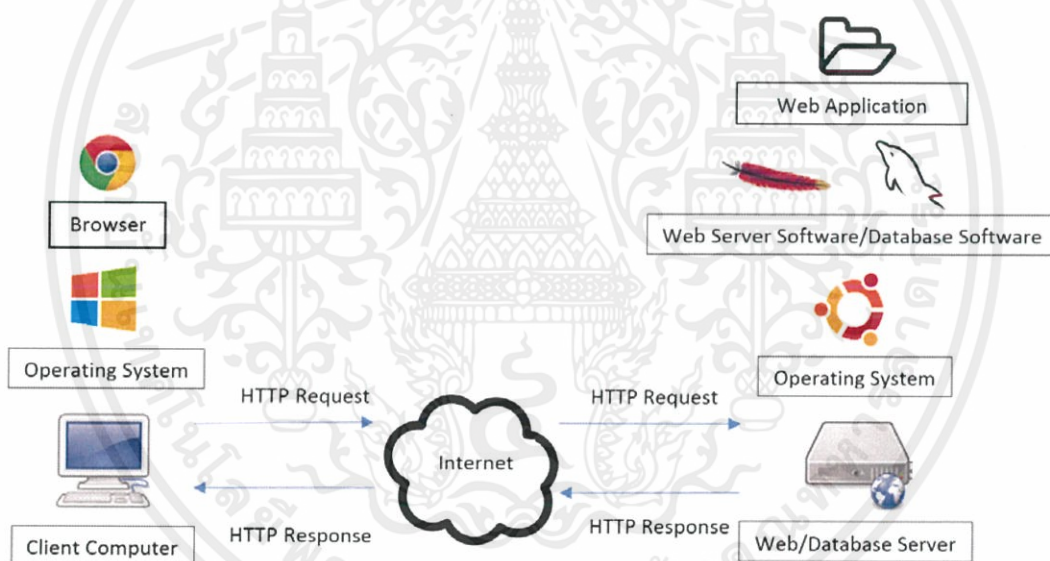
จากตัวอย่างปัญหาเหล่านี้คณะผู้จัดทำจึงเลือกใช้เทคโนโลยีเว็บแอปพลิเคชันในการเขียนโปรแกรม “ระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซี” เพราะเว็บแอปพลิเคชันสามารถตอบสนองปัญหาข้างต้นได้เป็นอย่างดี และสามารถแทนที่ Desktop Application ที่เป็น Client-Server Application ได้เป็นอย่างดี ตัวโปรแกรมของเว็บแอปพลิเคชันจะถูกติดตั้งไว้ที่ Server คอยให้บริการกับ Client โดยที่ Client ก็ไม่ต้องติดตั้งโปรแกรมเพิ่มเติม สามารถเข้าใช้งานระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่าน WebBrowser ที่ติดมากับ OS ได้ทันที อย่าง Internet Explorer หรือโปรแกรมฟรี ได้แก่ FireFox และ Google Chrome ด้วยความสามารถของ Browser ที่หลากหลาย ทำให้ไม่จำกัดว่าเครื่องที่ใช้เป็น OS อะไร หรืออุปกรณ์อะไร อย่างอุปกรณ์ TouchPad หรือ SmartPhone ก็สามารถเรียกใช้งานได้ และยังคงลข้อจำกัดเรื่องสถานที่ใช้งานอีกด้วย

2.1.2 ส่วนประกอบของเว็บแอปพลิเคชัน

เว็บแอปพลิเคชันในปัจจุบันประกอบไปด้วยการทำงานของเทคโนโลยีต่าง ๆ มากมาย ไม่ว่าจะเป็นตัวโปรแกรมเว็บแอปพลิเคชัน (web application) เว็บเซิร์ฟเวอร์ (web server) เว็บเซิร์ฟเวอร์ซอฟต์แวร์ (web server software) ฐานข้อมูล (database) เว็บเบราว์เซอร์ (web browser) และอื่น ๆ ซึ่งแต่ละส่วนก็จะมีหน้าที่และการทำงานที่แตกต่างกันออกไป เราสามารถแยกส่วนประกอบของการทำงานของเว็บแอปพลิเคชันออกเป็นสองส่วนหลัก ๆ คือ เทคโนโลยีฝั่งผู้ใช้งาน (client-side technology) และเทคโนโลยีฝั่งเซิร์ฟเวอร์ (server-side technology)



รูปที่ 2.1 ส่วนประกอบของเว็บแอปพลิเคชัน

2.1.2.1 เทคโนโลยีฝั่งผู้ใช้งาน (client-side technology)

จากรูปที่ 2.1 ฝั่งซ้ายของรูปคือส่วนฝั่งผู้ใช้งานประกอบด้วย 3 ส่วนหลักดังต่อไปนี้

- 1) เครื่องคอมพิวเตอร์ของผู้ใช้ (Client Computer)
- 2) ระบบปฏิบัติการ (Operating System) ระบบปฏิบัติการทำหน้าที่ในการจัดการกับทรัพยากรของเครื่องคอมพิวเตอร์ ทำหน้าที่ในการรับ HTTP request จากเบราว์เซอร์และส่งต่อไปให้กับอินเทอร์เน็ต DNS ในระบบปฏิบัติการทำหน้าที่ในการแปลง URL ให้เป็น IP Address

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อค้นหาเครื่องเว็บเซิร์ฟเวอร์ สร้างการเชื่อมต่อ (TCP connection) ระหว่างเครื่องผู้ใช้งานและเครื่องเซิร์ฟเวอร์ ดังนั้นการทำงานของระบบปฏิบัติการจะเป็นสิ่งที่ผู้ใช้งานมองไม่เห็นแต่ก็มีความสำคัญมาก

3) เว็บเบราว์เซอร์ (Web Browser) เว็บเบราว์เซอร์ เป็นซอฟต์แวร์ที่ผู้ใช้งานใช้ในการเข้าถึงเว็บแอปพลิเคชัน โดยที่เมื่อเริ่มต้น ผู้ใช้งานทำการใส่ URL หรือว่าชื่อของเว็บไซต์ที่ต้องการเข้าใช้งาน เช่น <https://www.google.com> เมื่อเบราว์เซอร์ได้รับชื่อของเว็บไซต์ก็จะทำการแปลงจากชื่อของเว็บไซต์เป็น IP address ผ่านทาง DNS หลังจากนั้นเว็บเบราว์เซอร์จะทำการสร้าง HTTP request เพื่อส่งคำร้องไปยังเว็บเซิร์ฟเวอร์ผ่านทางเครือข่ายอินเทอร์เน็ต เมื่อได้รับ HTTP response จากเว็บเซิร์ฟเวอร์ เว็บเบราว์เซอร์จะทำหน้าที่ในการอ่าน และแปลง HTTP response ให้เป็นข้อมูลที่ใช้ในการแสดงผลให้กับผู้ใช้งาน ดังนั้นหน้าที่ของเว็บเบราว์เซอร์จะประกอบไปด้วย

- 1) รับข้อมูลและคำสั่งจากผู้ใช้งาน
- 2) แปลงคำสั่งของผู้ใช้งานให้เป็น HTTP request เพื่อส่งไปให้กับเว็บเซิร์ฟเวอร์
- 3) ประมวลผล HTTP response
- 4) แปลงภาษา HTML, CSS, JavaScript ให้ข้อมูลสำหรับแสดงผลให้กับผู้ใช้งาน
- 5) จัดจำข้อมูลผู้ใช้งานเช่น ประวัติการใช้งาน ข้อมูล session และ cookie

2.1.2.2 เทคโนโลยีฝั่งเซิร์ฟเวอร์ (server-side technology)

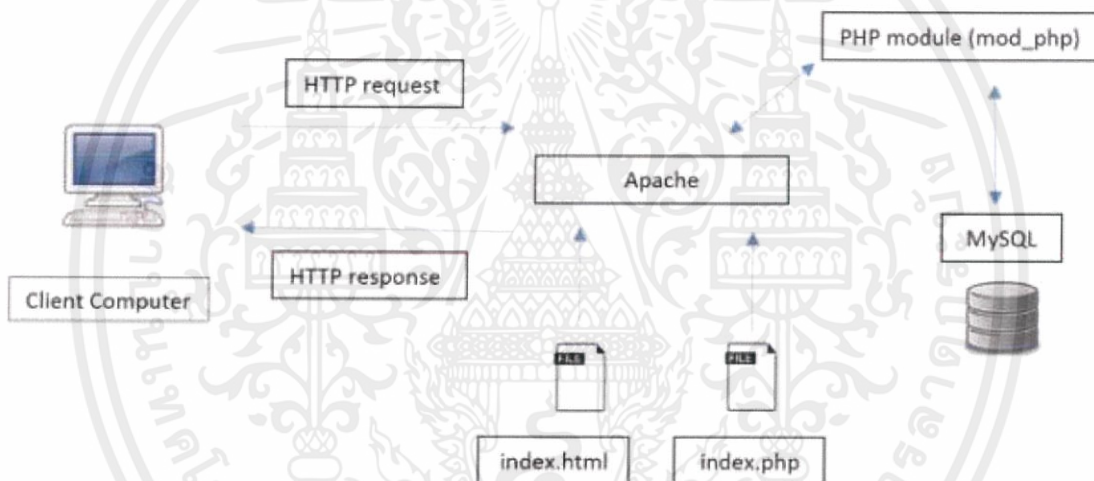
จากรูปที่ 2.1 ฝั่งขวาของรูปคือส่วนฝั่งเซิร์ฟเวอร์ประกอบด้วย 4 ส่วนหลักดังต่อไปนี้

- 1) เครื่องเซิร์ฟเวอร์ (Web/Database Server)
- 2) ระบบปฏิบัติการ (Operating System) ระบบปฏิบัติการบนฝั่งของเซิร์ฟเวอร์มีหน้าที่ในการจัดการกับทรัพยากรของเครื่องเซิร์ฟเวอร์ เช่น CPU memory และ bandwidth เป็นต้น เนื่องจากเว็บแอปพลิเคชันเป็นบริการที่เปิดให้ผู้ใช้งานเข้าถึงได้ตลอดเวลา ดังนั้น ระบบปฏิบัติการบนเซิร์ฟเวอร์จึงต้องมีความเสถียรและสามารถจัดการกับทรัพยากรของเครื่องได้อย่างมีประสิทธิภาพ

- 3) เว็บเซิร์ฟเวอร์ซอฟต์แวร์ (Web Server Software) Web server software เป็นโปรแกรมที่ทำงานอยู่บน web server ซึ่งหน้าที่หลักของ web server software คือการประมวลผล HTTP request ที่ได้รับมาและตอบกลับด้วย HTTP response ให้กับผู้ใช้งาน ปัจจุบันมี web server software หลายตัวที่ถูกใช้งานอย่างแพร่หลายเช่น Apache HTTP server, Internet Information Service (IIS) และ Nginx ยังมี web server software ตัวอื่นอีกมากในท้องตลาดที่ไม่ได้กล่าวถึงในที่

นี่อย่างไรก็ตาม web server software ที่ได้รับความนิยมอย่างแพร่หลายมากที่สุดในปัจจุบันคือ Apache HTTP server และผู้ใช้งานมักจะใช้คู่กับ PHP (ตัวแปลภาษา PHP) และ MySQL (ฐานข้อมูล)

Apache HTTP server เป็น web server software ที่ได้รับความนิยมสูงสุด (ข้อมูล ณ วันที่ 21 ตุลาคม 2558) เนื่องจากความสามารถที่หลากหลาย และเป็น freeware ที่อนุญาตให้นำไปใช้งานได้ฟรีทางการค้า Apache HTTP server ปัจจุบันออกเวอร์ชัน 2.4 (วันที่ข้อมูล) ซึ่งมีโครงสร้างการทำงานเป็นแบบ module นั่นคือ ผู้ใช้งานสามารถเพิ่มความมารถของ web server software ได้โดยการติดตั้ง module เพิ่มเติม ตัวอย่างเช่น หากต้องการให้ Apache HTTP server รองรับภาษา PHP ก็สามารถติดตั้ง module ที่สามารถช่วยให้ Apache ประมวลผลเว็บแอปพลิเคชันที่เขียนด้วยภาษา PHP ได้



รูปที่ 2.2 การทำงานร่วมกันของ Apache PHP และ MySQL

การทำงานของ Apache, PHP, และ MySQL

เมื่อได้รับ HTTP request มาจากผู้ใช้งาน Apache จะทำการประมวลผล HTTP request เพื่อตรวจสอบประเภทของไฟล์ที่ร้องขอ หากไฟล์ที่ร้องขอเป็นไฟล์ข้อมูล เช่น .jpeg .html หรือ .pdf Apache สามารถอ่านไฟล์เหล่านี้และส่งเป็น HTTP response กลับไปให้กับผู้ใช้งานได้ทันที แต่หากไฟล์ที่ HTTP request ร้องขอมาเป็นไฟล์โปรแกรมที่ต้องการประมวลผล เช่น .php Apache จะทำการเรียกใช้ PHP module ในการประมวลผลไฟล์ก่อน ซึ่งในการประมวลผลไฟล์ อาจจะมีการติดต่อกับฐานข้อมูลเช่น MySQL เพื่อทำการเรียกดู หรือแก้ไขข้อมูลก็สามารถทำได้ เมื่อ PHP module ทำการประมวลผลไฟล์ .php เสร็จแล้วก็จะทำการส่งค่าให้ Apache นำไปสร้างเป็น HTTP response เพื่อส่งกลับให้กับผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) เว็บแอปพลิเคชัน (Web Application) ถือเป็นหัวใจหลักของเว็บไซต์เนื่องจากทำหน้าที่ติดต่อกับผู้ใช้งาน รับและแสดงข้อมูล ประมวลผลข้อมูล จัดการข้อมูลในฐานข้อมูล และอื่นๆ เรียกได้ว่าเว็บแอปพลิเคชันเป็นซอฟต์แวร์ที่ให้บริการผู้ใช้งานทั่วโลกผ่านอินเทอร์เน็ต หากนักพัฒนาได้เขียนเว็บแอปพลิเคชันตาม Model-View-Controller (MVC) แล้วก็จะสามารถแบ่งเว็บแอปพลิเคชันออกได้เป็นสามส่วนหลัก ๆ คือ

ส่วนที่ติดต่อกับผู้ใช้งานเพื่อรับข้อมูลและแสดงผล (View)

ส่วนที่ประมวลผลการทำงาน (Controller)

ส่วนที่ใช้ในการติดต่อและจัดการกับข้อมูลและฐานข้อมูล (Model)

นักพัฒนาสามารถพัฒนาเว็บแอปพลิเคชันได้ด้วยภาษาคอมพิวเตอร์ที่หลากหลาย เราสามารถแบ่งภาษาที่ใช้ในการพัฒนาเว็บแอปพลิเคชันออกเป็นสองส่วนคือ Front-End Technology ใช้สำหรับพัฒนา View (ส่วนติดต่อกับผู้ใช้งาน) และ Back-End Technology ใช้สำหรับพัฒนา Model และ Controller (ส่วนประมวลผลและจัดการข้อมูล)

Front-End Web Technology จะหมายถึงส่วนของเทคโนโลยีที่ใช้ในการสร้างส่วนติดต่อกับผู้ใช้งาน ในการสร้างเว็บแอปพลิเคชัน Front-End Technology ที่เป็นที่แพร่หลายได้แก่ HTML, CSS, และ JavaScript ซึ่งภาษาคอมพิวเตอร์เหล่านี้ถูกใช้อย่างแพร่หลายในการสร้างส่วนติดต่อกับผู้ใช้งานของเว็บแอปพลิเคชัน ความหลากหลายของ Front-End Web Technology ถูกจำกัดด้วยมาตรฐานกลางที่ออกโดยองค์กรที่ไม่แสดงหาผลกำไรอย่าง World Wide Web Consortium (W3C) ซึ่งเป็นผู้กำหนดมาตรฐาน HTML, CSS, และ JavaScript เพื่อให้ผู้พัฒนาเบราว์เซอร์ให้แสดงผลข้อมูลในรูปแบบเดียวกัน เพื่อความสะดวกแก่ผู้ใช้งานและนักพัฒนา ซึ่งเบราว์เซอร์ในปัจจุบันต่างรองรับการประมวลผลของ HTML, CSS และ JavaScript โดยสมบูรณ์ แม้ว่าจะมีความแตกต่างในการแสดงผลไปบ้าง(เล็กน้อย)ในบางเบราว์เซอร์

Back-End Web Technology จะหมายถึงส่วนของเทคโนโลยีที่เป็นส่วนประมวลผลตรรกะและการทำงานของเว็บแอปพลิเคชัน ไม่ว่าจะเป็นการตรวจสอบสิทธิ์การเข้าใช้ การเรียกดูและจัดเก็บข้อมูล การทำงานของเว็บแอปพลิเคชันในส่วนของ Back-End จะเริ่มหลังจากเว็บแอปพลิเคชันได้รับ HTTP request มาจากผู้ใช้งาน ทำการประมวลผล และส่งข้อมูลกลับไปให้กับผู้ใช้งาน เทคโนโลยีที่ใช้ในการพัฒนา Back-End ของเว็บแอปพลิเคชันจะมีความหลากหลายกว่า Front-End เนื่องจากไม่มีข้อจำกัดด้านมาตรฐานกลางดัง Front-End technology ที่ต้องรองรับมาตรฐานที่กำหนดโดย W3C เพื่อให้ทำงานกับเว็บเบราว์เซอร์ได้อย่างไม่มีปัญหา

2.2 Static code analysis

การเขียนโปรแกรมบางเทคนิคจะไม่จำเป็นต้องใช้บางคำสั่งเช่นการเขียนโปรแกรมแบบ Recursion ซึ่งจะต้องไม่ใช่คำสั่งการวนลูป(Loop) อย่าง for, while และ do...while เข้าช่วยในการเขียนโปรแกรม

ในยุคแรกๆของระบบตรวจโค้ดโปรแกรมการกำหนดข้อจำกัดเพื่อให้ผู้เรียนแก้ปัญหาด้วยเทคนิค Recursion เท่านั้นจะทำโดยการตั้งเงื่อนไขว่าห้ามใช้คำสั่ง for โดยจะตั้งเงื่อนไขในลักษณะของการกำหนด define ไว้ในส่วนหัวโปรแกรมเช่น #define for 123 จากคำสั่งนี้จะหมายถึงการประกาศให้คำสั่ง for นั้นถูกแทนที่ด้วย 123 ในขั้นตอนของการทำ preprocessor ของภาษา C และภาษา C++ ทำให้ภายในโค้ดทั้งหมดไม่สามารถใช้คำสั่ง for ได้หมายถึงหากต้องการเขียนโปรแกรมเพื่อแก้ปัญหานี้จะไม่สามารถใช้ for ในการแก้ปัญหาได้แต่ก็จะส่งผลกับการใช้ for ในการเขียนส่วนแสดงผลไปด้วยซึ่งควรจำกัดการใช้คำสั่งแต่ละเฉพาะในส่วนของการแก้ปัญหาเท่านั้นไม่ควรรวมถึงการเขียนคำสั่งเพื่อแสดงผลด้วย

จากการศึกษาการวิเคราะห์โค้ด (Code analysis) นั้นแบ่งออกเป็น 2 วิธี

1) Static code analysis เป็นการวิเคราะห์คุณภาพและคุณสมบัติจากซอร์สโค้ดโดยตรงโดยไม่ต้องทำการรันโปรแกรม

2) Dynamic code analysis เป็นการวิเคราะห์คุณภาพและคุณสมบัติจากการรันโปรแกรมซึ่งโดยปกติการทดสอบจะได้ผลลัพธ์ที่ถูกต้องจะต้องมีค่าอินพุต(Input tests) ที่มากพอ

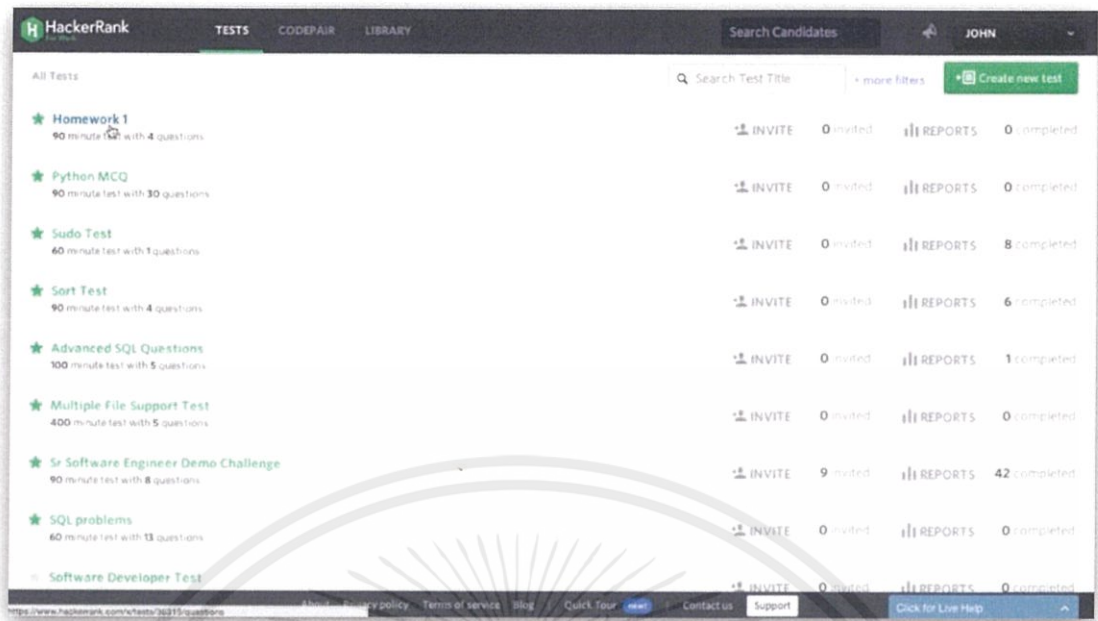
ซึ่งจากปัญหาข้างต้นหากเราต้องการตรวจสอบฟังก์ชันใดๆว่าเป็นไปตามข้อกำหนดหรือไม่โดยข้อกำหนดมีดังนี้

- 1) เป็นฟังก์ชัน Recursion หรือไม่
- 2) มีลูปซ้อนลูป(Nested loop) สูงสุดกี่ชั้น

เราสามารถเลือกใช้รูปแบบของ Static code analysis เพื่อมาตรวจสอบได้มีข้อดีในส่วนของความเร็วในการทำงานเนื่องจากไม่ต้องนำซอร์สโค้ดไปประมวลผล

2.3 ตัวอย่างระบบตรวจโค้ดโปรแกรมมิ่งในปัจจุบัน

ปัจจุบันมีระบบตรวจโค้ดให้บริการในเชิงธุรกิจอยู่เป็นจำนวนมากหนึ่งเว็บไซต์ที่เป็นที่นิยมให้บริการคือ www.hackerrank.com เป็นระบบให้บริการตรวจโค้ดนักศึกษาออนไลน์, จัดการโจทย์ปัญหาและให้คะแนนอัตโนมัติ



รูปที่ 2.3 ตัวอย่างหน้าใช้งานของอาจารย์



รูปที่ 2.4 ตัวอย่างการทำโจทย์ของนักศึกษา

การทำงานของระบบ Hackerrank คืออาจารย์จะเป็นผู้ตั้งโจทย์ปัญหาและอัปโหลดขึ้นสู่เซิร์ฟเวอร์จากนั้นนักศึกษาจะเข้าสู่ระบบเพื่อเขียนโปรแกรมแก้ปัญหาดังกล่าวระบบจะจัดเก็บคะแนนนักศึกษาอัตโนมัติ

ข้อจำกัดของบริการดังกล่าวนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1).ไม่สามารถใช้งานออฟไลน์ได้
- 2).บริการดังกล่าวไม่สามารถแก้ไขการตั้งค่าได้เนื่องจากเป็นบริการ Software as a service (SaaS)

จากการศึกษาระบบตรวจโค้ดนี้จึงนำมาสู่การพัฒนา “ระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซี” ซึ่งทำให้สามารถกำหนดขอบเขตของเครือข่ายการใช้งานและสามารถพัฒนาเพิ่มเติมในส่วนของฟีเจอร์อื่นๆต่อไปได้

2.4 โคลเอนต์ - เซิร์ฟเวอร์โมเดล

โคลเอนต์ - เซิร์ฟเวอร์เป็นการอธิบายความสัมพันธ์ระหว่างโปรแกรมคอมพิวเตอร์สองโปรแกรมโดยโปรแกรมหนึ่งเป็นโคลเอนต์จะสร้างคำร้องบริการจากอีกโปรแกรมหรือเซิร์ฟเวอร์แนวคิดนี้เป็นแนวคิดสำคัญในระบบเครือข่ายแบบจำลอง โคลเอนต์ - เซิร์ฟเวอร์ให้แบบแผนการติดต่อภายในโปรแกรมที่ให้ประสิทธิภาพการกระจายข้ามตำแหน่งที่ต่างกัน

แบบจำลอง โคลเอนต์ - เซิร์ฟเวอร์จะมีเซิร์ฟเวอร์หนึ่งเครื่องและคอปียคำร้องขอของโคลเอนต์โดยปกติโปรแกรมโคลเอนต์หลายโปรแกรมใช้บริการเซิร์ฟเวอร์ร่วมกันซึ่งมักจะเป็นส่วนหนึ่งของโปรแกรมประยุกต์ขนาดใหญ่เมื่อเปรียบเทียบกับอินเทอร์เน็ตเว็บเบราว์เซอร์ของผู้ใช้โปรแกรมเป็นโคลเอนต์ที่ขอบริการจากเว็บเซิร์ฟเวอร์ในทางเทคนิคเรียกว่า Hypertext Transfer Protocol Server ซึ่งเป็นคอมพิวเตอร์อีกเครื่องบนอินเทอร์เน็ต

เครือข่ายแบบ โคลเอนต์ - เซิร์ฟเวอร์เหมาะกับระบบเครือข่ายที่ต้องการเชื่อมต่อกับเครื่องโคลเอนต์จำนวนมากโดยการรองรับจำนวนเครื่องโคลเอนต์อาจมากถึงหลักพันเพราะฉะนั้นเครื่องที่จะนำมาทำหน้าที่ให้บริการจะต้องเป็นเครื่องที่มีประสิทธิภาพสูงเนื่องจากต้องสามารถทนทานต่อความผิดพลาด (Fault Tolerance) และต้องคอยให้บริการทรัพยากรให้กับเครื่องลูกข่ายตลอดเวลาสำหรับองค์กรขนาดใหญ่อาจมีการติดตั้งหลายๆเซิร์ฟเวอร์เพื่อแบ่งภาระการทำงาน

2.4.1 โคลเอนต์

เป็นโปรแกรมที่ถูกรันอยู่บนเครื่องของผู้ใช้เพื่อเรียกใช้บริการจากเซิร์ฟเวอร์โคลเอนต์จะเปิดช่องทางสื่อสารกับเซิร์ฟเวอร์โดยใช้ไอพีแอดเดรสและหมายเลขพอร์ตของเซิร์ฟเวอร์ซึ่งเรียกวิธีแบบนี้ว่า active open เมื่อช่องทางสื่อสารเปิดออกแล้วโคลเอนต์สามารถส่งคำร้องและรับบริการจากเซิร์ฟเวอร์ได้

2.4.2 เซิร์ฟเวอร์

เป็นโปรแกรมที่ถูกรันอยู่บนเครื่องที่จะคอยให้บริการกับเครื่องอื่นๆเมื่อมีการรันโปรแกรมและมีการเปิดช่องทางการสื่อสารเอาไว้เพื่อให้ไคลเอนต์สามารถติดต่อเข้ามาได้แต่จะยังไม่ให้บริการใดๆจนกระทั่งมีคำร้องขอมาจากไคลเอนต์เราเรียกวิธีการแบบนี้ว่า passive open โดยตัวอย่างประเภทของเซิร์ฟเวอร์มีดังต่อไปนี้

2.4.2.1 เว็บเซิร์ฟเวอร์ (Web Server)

คือเครื่องคอมพิวเตอร์ที่ติดตั้งโปรแกรมคอมพิวเตอร์ที่ทำหน้าที่ให้บริการข้อมูลในรูปแบบสื่อผสมผ่านระบบเครือข่ายโดยสามารถแสดงผลผ่านโปรแกรมอินเทอร์เน็ตเบราว์เซอร์ทางด้านของผู้ขอใช้บริการ

2.4.2.2 ดาต้าเบสเซิร์ฟเวอร์ (Database Server)

คือเครื่องคอมพิวเตอร์ที่ทำหน้าที่ให้บริการจัดเก็บข้อมูลต่างๆด้วยระบบฐานข้อมูล

2.1.2.3 โดเมนเนมเซิร์ฟเวอร์ (Domain Name Server)

คือเครื่องคอมพิวเตอร์ที่ติดตั้งโปรแกรมสำหรับทำหน้าที่จัดการชื่อต่างๆของ Server ทำให้สามารถเรียกใช้งานได้สะดวกรวดเร็วขึ้น

2.5 sandbox

Sandbox เป็นตัวทดลองการทำงานของแอปพลิเคชันหรือ execute file โดยเป็นสภาพแวดล้อมเดียวกันกับการใช้งานจริง (production environment) เนื่องจากการทำงานในผ่านเว็บแอปพลิเคชันจะมีการทำงานบางส่วนที่จะต้องไปเรียกใช้ execute command เพื่อที่จะสั่งทำงานภายในเซิร์ฟเวอร์ โดยที่การทำงานภายใน Sandbox สามารถ execute command และ execute code ได้เหมือนปกติ จุดประสงค์ของการทำงานผ่าน Sandbox นั้นเพื่อป้องกันปัญหา command ทำงานนอกเหนือจากรายการเช่นการเขียน/ลบไฟล์ระบบ(system file), การเปิดช่องติดต่อโดยตรง (directly open a socket) การทำงานบางคำสั่งและการทำงานผ่าน root user (require root access) เป็นต้น โดยที่สามารถกำหนดสิทธิ์การทำงานที่อนุญาตผ่าน โพรไฟล์ที่ใส่ไปได้

การใช้งานหลักของ Sandbox เป็นใน ส่วนของการคอมไพล์และรันไฟล์ภาษาซี (compile/execute C file) ในสภาพแวดล้อมของ Sandbox โค้ดจะทำงานภายในเงื่อนไขทำให้ระบบเซิร์ฟเวอร์ไม่ถูกรบกวนเช่นคำสั่ง sleep, shutdown, reboot จากผู้ส่งโค้ด

2.6 Bootstrap

Bootstrap คือชุดคำสั่งที่ประกอบด้วยภาษา CSS, HTML และ Javascript เป็นชุดคำสั่งที่ถูกพัฒนาขึ้นมาเพื่อกำหนดกรอบหรือรูปแบบการพัฒนาเว็บไซต์ในส่วนของการปฏิสัมพันธ์กับผู้ใช้งานเว็บไซต์ (user interface) ถือได้ว่า Bootstrap เป็น front-end framework สำหรับการพัฒนาเว็บไซต์ในส่วนการแสดงผล

Bootstrap ถูกพัฒนาขึ้นโดย Mark Otto และ Jacob Thornton ทีมพัฒนาของ Twitter Inc. ก่อนหน้านี้ใช้ชื่อว่า Twitter Blueprint และเปิดให้นักพัฒนาสามารถนำไปใช้งานพัฒนาเว็บไซต์ได้แบบฟรี (open source) ในชื่อว่า Bootstrap framework

เนื่องจาก Bootstrap เป็น framework การใช้งานจึงจำเป็นต้องศึกษาวิธีการจัดวางและการแสดงผลให้ถูกต้องตามที่ Bootstrap ได้จัดเตรียมไว้ทำให้เว็บไซต์รองรับหน้าจออุปกรณ์ทุกรูปแบบซึ่งเป็นการจัดหน้าจอแบบ responsive web และสามารถเปลี่ยนสไตล์เพิ่มเติมได้ในเว็บไซต์ของ Bootswatch โดยทั้งคู่มีใบอนุญาต MIT(MIT License) ที่อนุญาตให้นำไปใช้และแก้ไขได้

บทที่ 3

การออกแบบและพัฒนา

3.1 ความต้องการของระบบ และลักษณะสำคัญ

3.1.1 ลักษณะของโจทย์

โจทย์จะต้องประกอบไปด้วยการรับค่าข้อมูลนำเข้าค่าข้อมูลส่งออกและข้อกำหนดต่างๆซึ่งค่าข้อมูลนำเข้าและค่าข้อมูลส่งออกนั้นสามารถมีจำนวนได้ตั้ง 1 ขึ้นไปพร้อมกับคำอธิบายว่าข้อมูลนั้นๆเป็นประเภทใดเช่น string, char, int เป็นต้นและบอกช่วงของค่าข้อมูลนำเข้าและข้อมูลส่งออกที่เป็นไปได้ทั้งหมดหรือบอกว่าเป็นกรณีใดได้บ้างเช่นค่าข้อมูลนำเข้าทั้ง 10 กรณีอยู่ในช่วงระหว่าง 0 - 1,000 และค่าข้อมูลส่งออกคือ Yes หรือ No เท่านั้น เป็นต้น

ในส่วน of ข้อกำหนดต่างๆนั้นจะประกอบไปด้วย

3.1.1.1 การจำกัดเวลา(Time limit)

อาจารย์สามารถจำกัดเวลาเพื่อให้การรันโค้ดทำงานภายในเวลาที่กำหนดและป้องกันการรันโค้ดที่มีลูปไม่รู้จบได้นอกจากนั้นการจำกัดเวลานี้ทำให้นักศึกษาจำเป็นต้องออกแบบอัลกอริทึมให้มีประสิทธิภาพเพื่อที่โค้ดจะสามารถทำงานได้ภายในเวลาที่กำหนดโดยทั่วไปการจำกัดเวลาถูกตั้งไว้ที่ 1 วินาทีต่อการทดสอบ 1 ชุด

3.1.1.2 การจำกัดหน่วยความจำ(Memory limit)

อาจารย์สามารถจำกัดหน่วยความจำเพื่อให้การรันโค้ดใช้ทรัพยากรอย่างเหมาะสมและป้องกันปัญหาการจองหน่วยความจำมากเกินไป (Stack overflow) ซึ่งจำกัดเฉพาะส่วน static memory allocation โดยทั่วไปการจำกัดหน่วยความจำถูกตั้งไว้ที่ 32 MB ต่อการทดสอบ 1 ชุด

3.1.1.3 การจำกัดเงื่อนไขของฟังก์ชัน(Condition of function)

ผู้ออกโจทย์สามารถกำหนดให้ฟังก์ชันที่เขียนในภาษาซีมีเงื่อนไขดังต่อไปนี้ได้ (สามารถกำหนดได้เพียง 1 ฟังก์ชันต่อ 1 ปัญหา) โดยการจำกัดเงื่อนไขของฟังก์ชันจะตรวจสอบด้วยหลักการของ static code analysis

1. Recursion property: ฟังก์ชันที่กำหนดจะต้องมีคุณสมบัติเป็นลักษณะ recursion คือเป็นฟังก์ชันที่มีการเรียกตัวเองซ้ำ

2. Loop layers: ฟังก์ชันที่กำหนดดังกล่าวจะต้องมี nested loop ภายในที่ผู้ออกโจทย์กำหนดซึ่งสามารถระบุได้ตั้งแต่ 0 จนถึง 5 nested loop โดย 0 หมายถึงจะต้องไม่มี loop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปรากฏขึ้นและ 5 หมายถึงมีลูปซ้อนกันได้สูงสุด 5 ชั้นซึ่ง nested loop ที่นับจะรวมฟังก์ชันอื่น (callee function) ที่ฟังก์ชันที่กำหนด(caller function) ไปเรียกใช้ด้วย

ข้อมูลเหล่านี้มีความสำคัญมากต่อการเขียน โปรแกรมซึ่งจะต้องอธิบายไว้ใน เอกสารอธิบายโจทย์โดยที่เอกสารอธิบายโจทย์ที่ใช้ในการอัปโหลดขึ้นเซิร์ฟเวอร์จะต้องเป็นไฟล์ ประเภท Portable Document Format (pdf) เท่านั้นเพื่อความสะดวกต่อการอ่านผ่านเว็บและรูปแบบ ของเอกสารที่ไม่คลาดเคลื่อน

3.1.2 ลักษณะของข้อมูลทดสอบ (Test case)

โจทย์ปัญหาในแต่ละข้อจะประกอบไปด้วยชุดข้อมูลที่ใช้ในการทดสอบการทำงานของ โค้ดซึ่งชุดข้อมูลทดสอบนี้ใช้ในการยืนยันว่าโค้ดนั้นสามารถทำงานในกรณีต่างๆไปหรือกรณี ที่กำหนดได้โดยปกติจำนวนของชุดทดสอบที่ใช้จะมี 10 ชุดถือได้ว่าเป็นการทดสอบการทำงานใน หลายกรณี

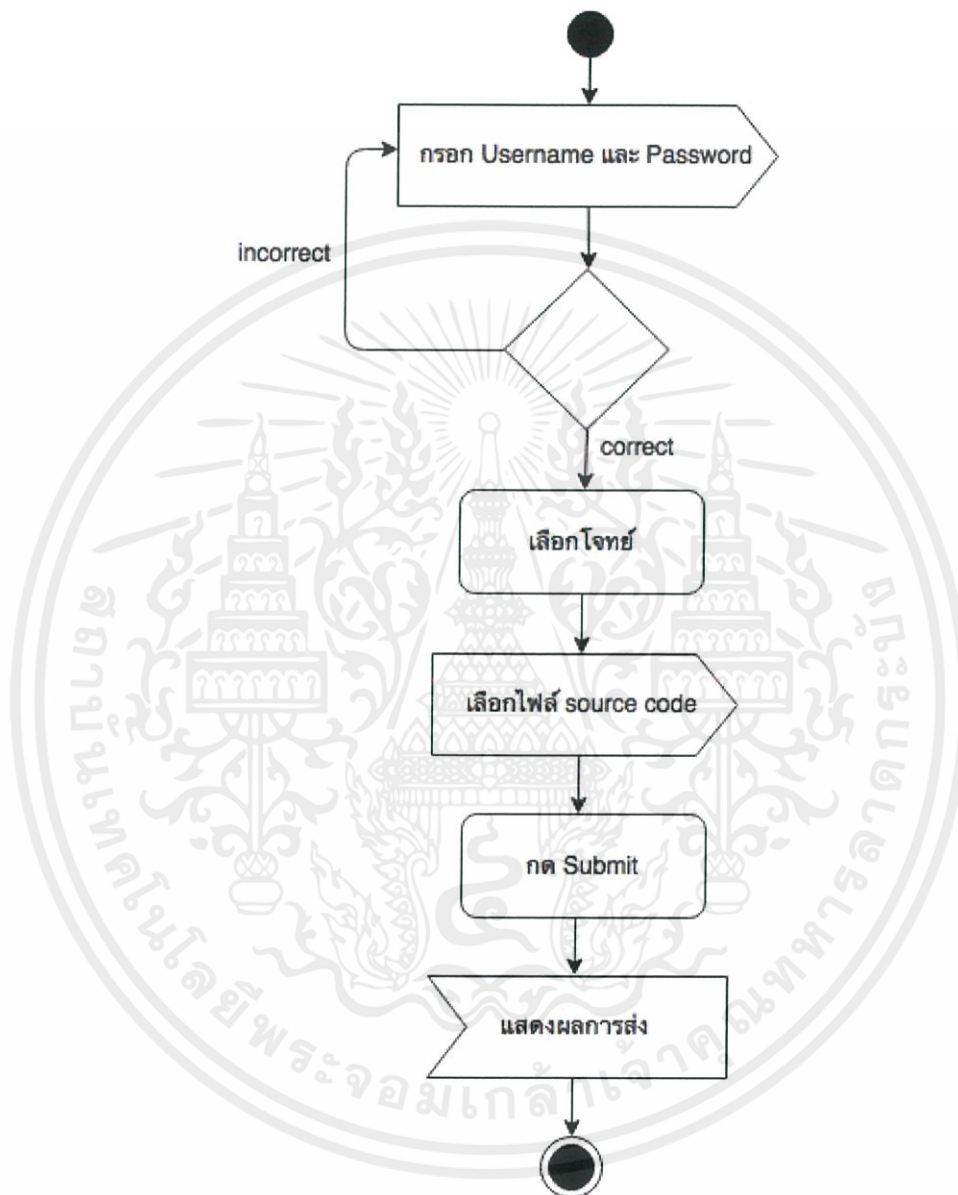
ชุดทดสอบ 1 กรณีจะประกอบไปด้วยไฟล์ 2 ไฟล์คือไฟล์ข้อมูลนำเข้า(input file) และ ไฟล์ข้อมูลส่งออก(output file) โดยมีการทำงานคร่าวๆดังนี้ นำซอร์สโค้ดของนักศึกษาไป ประมวลผลและใช้ไฟล์ข้อมูลนำเข้าเป็นพารามิเตอร์(parameter) จากนั้นนำผลลัพธ์ที่ได้จากการ ทดลองมาเปรียบเทียบกับไฟล์ข้อมูลส่งออกว่าเหมือนกันทั้งหมดหรือไม่ถ้าเหมือนกันทั้งหมดก็ จะได้คะแนนในข้อนั้นๆในกรณีอื่นๆ เช่น ผลการรันที่ได้ไม่ถูกต้องหรือเกินเวลาที่กำหนดจะไม่ได้ คะแนน

จะกล่าวได้ว่าหากต้องการทดสอบทั้งหมด 10 กรณีจะต้องเตรียมไฟล์ทั้งหมด 20 ไฟล์ โดยที่ไฟล์ข้อมูลนำเข้ากำหนดให้มีนามสกุลไฟล์เป็น .in และไฟล์ข้อมูลส่งออกมีนามสกุลไฟล์ เป็น .sol ชื่อไฟล์หมายถึงหมายเลขกรณีนั้นๆเช่น ในกรณีทดสอบที่ 1 (ชุดทดสอบ 1) จะประกอบไป ด้วยไฟล์ 1.in และ 1.sol

3.2 ภาพรวมของระบบ

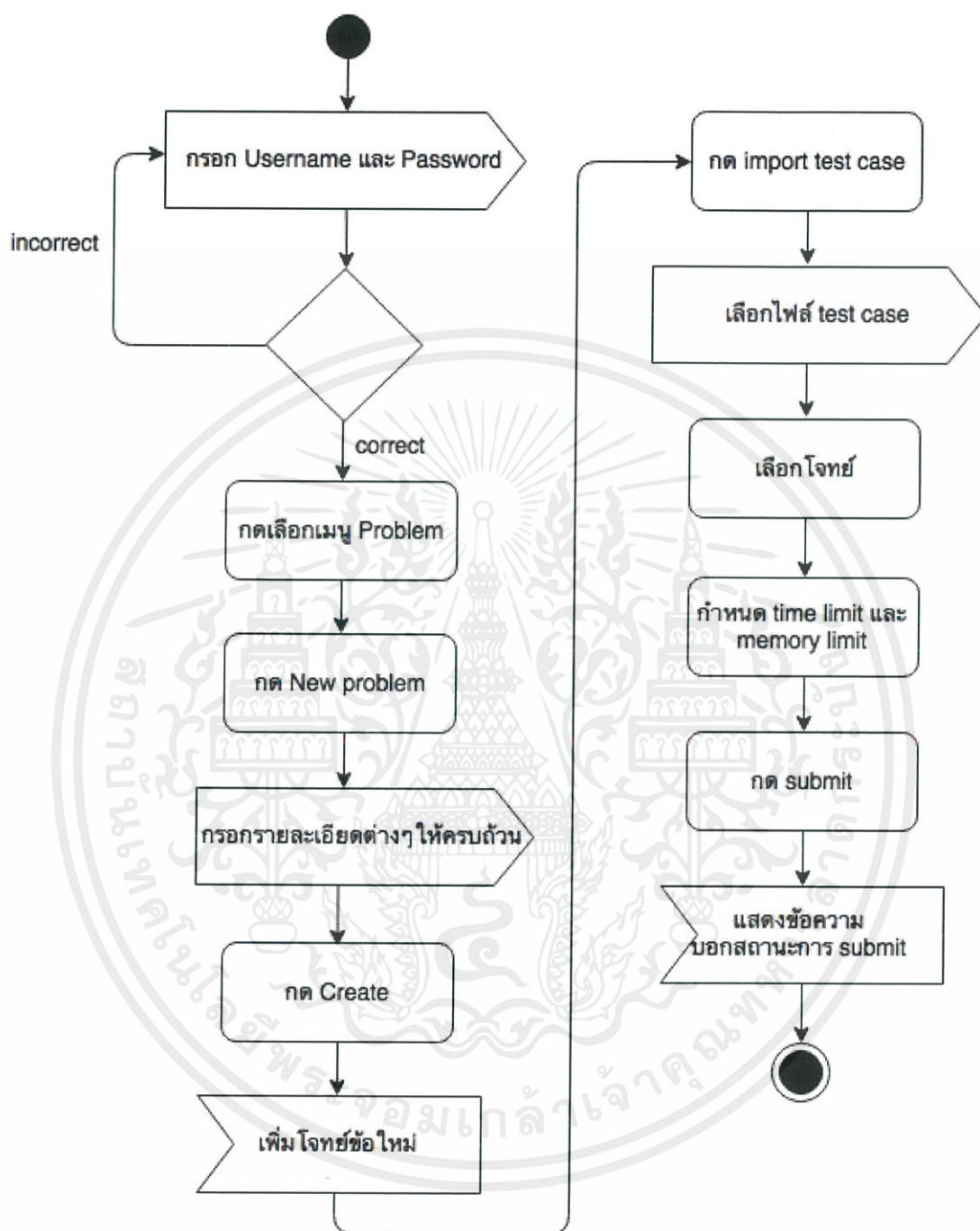
3.2.1 Activity Diagram

3.2.1.1 การส่งโค้ดของนักศึกษา



รูปที่ 3.1 Activity Diagram : การส่งโค้ดของนักศึกษา

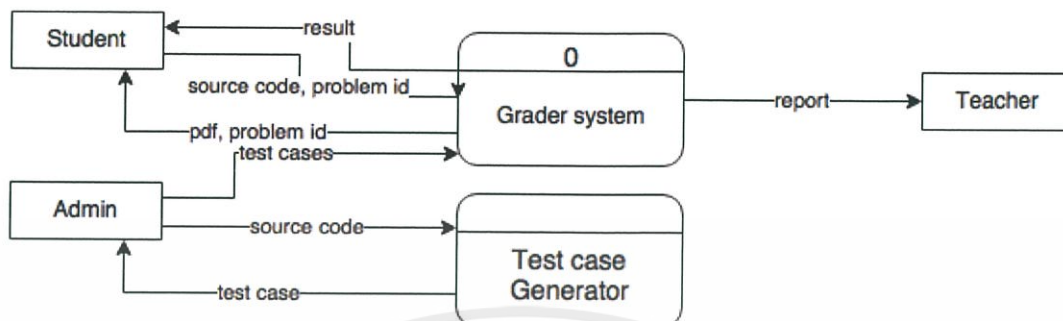
3.2.1.2 การเพิ่มโจทย์ของอาจารย์



รูปที่ 3.2 Activity diagram : การเพิ่มโจทย์ของอาจารย์

3.2.2 Data flow diagram

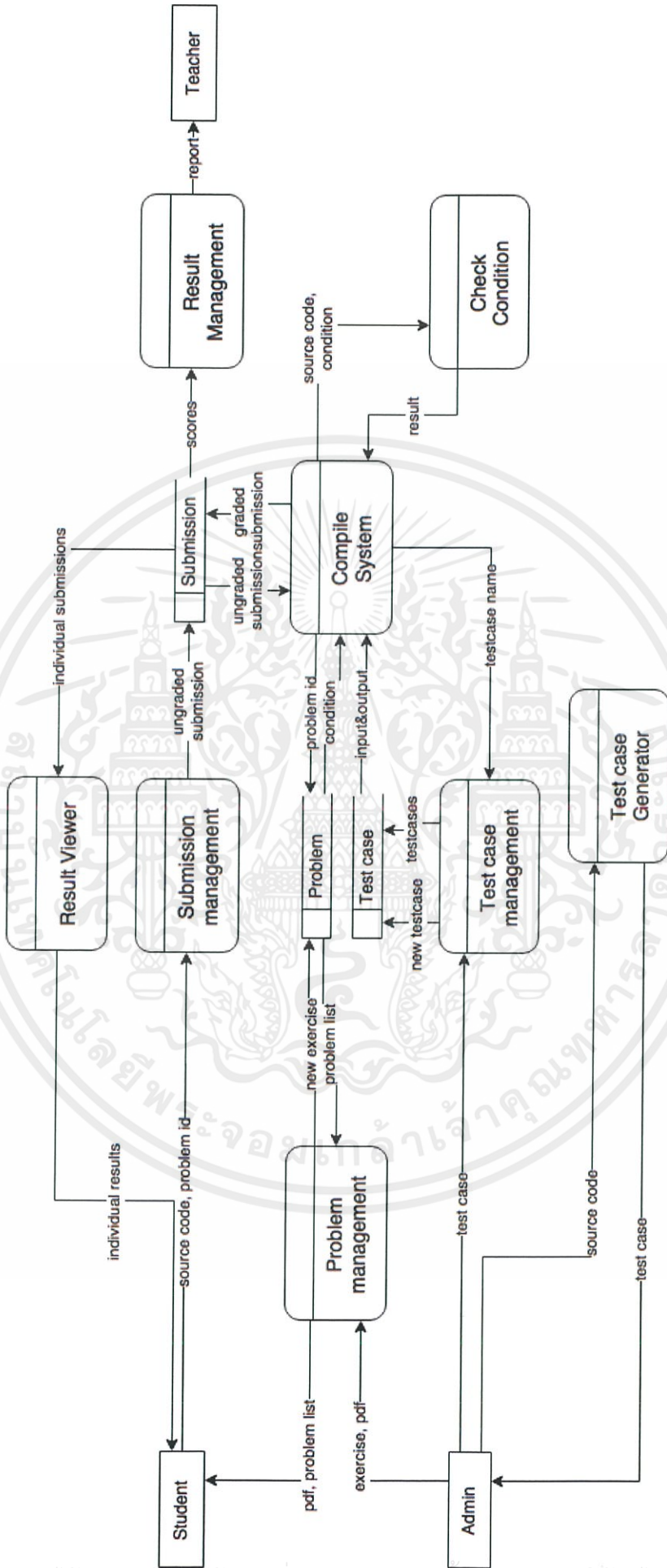
3.2.2.1 Data flow diagram level 0



รูปที่ 3.3 Data flow diagram level 0



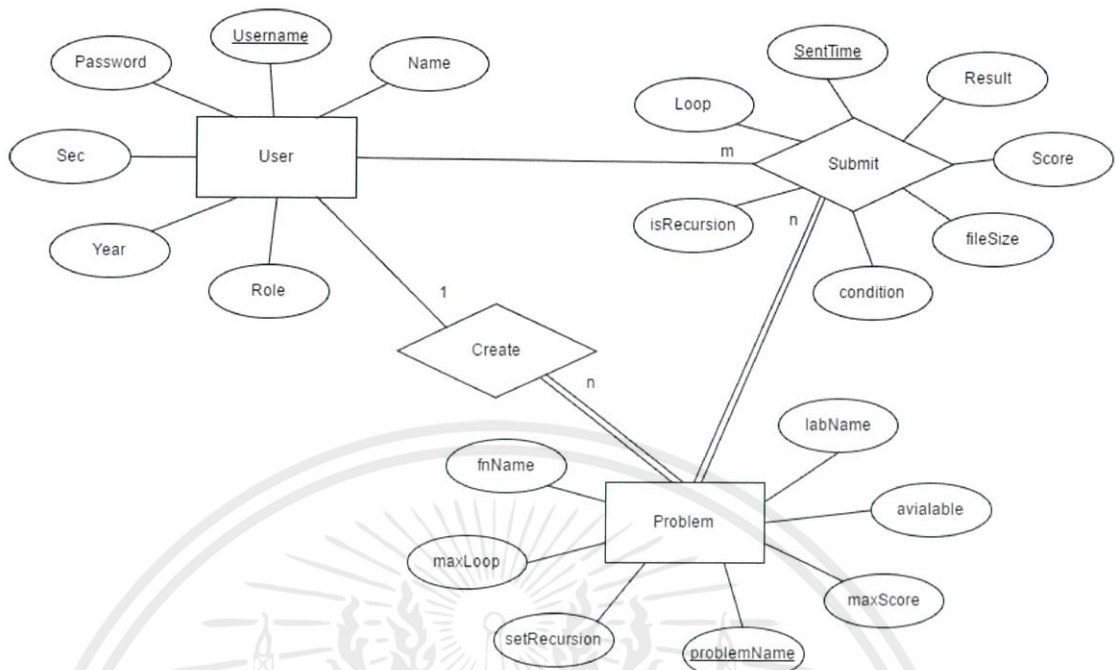
3.2.2.2 Data flow diagram level 1



รูปที่ 3.4 Data flow diagram level 1

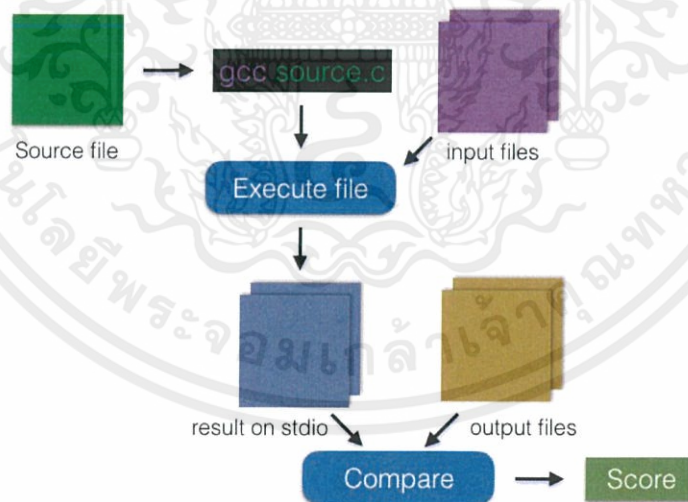
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ER Diagram



3.3 การออกแบบอัลกอริทึม

3.3.1 การคอมไพล์และประมวลผล



รูปที่ 3.5 การคอมไพล์และประมวลผล

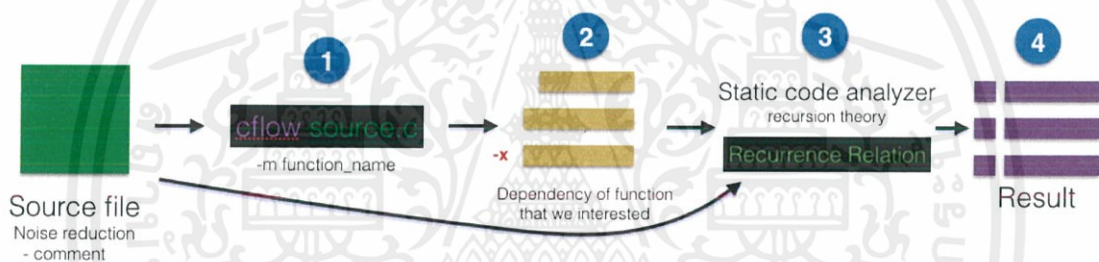
ซอร์สโค้ดของนักศึกษาจะถูกนำไปคอมไพล์ภายในเซิร์ฟเวอร์โดยการคอมไพล์นั้นจะดูจากภาษาที่ใช้หากเป็นภาษาซีจะใช้คอมไพเลอร์ gcc (version 4.8.4) ถ้าเป็นภาษา C++ จะใช้

คอมไพเลอร์ g++ (version 4.8.4) จากนั้นนำไปประมวลผลผ่าน sandbox และใช้ไฟล์ข้อมูลนำเข้า (.in) ของชุดทดสอบ (testcase) ที่เตรียมไว้สำหรับ โจทย์แต่ละข้อเป็นพารามิเตอร์

การให้คะแนนจะสนใจเฉพาะสิ่งที่แสดงออกทางหน้าจอเท่านั้น(stdout) โดยนำผลลัพธ์ที่ได้ไปเปรียบเทียบกับไฟล์ข้อมูลส่งออก (.sol) ของชุดทดสอบ ซึ่งต้องได้ผลลัพธ์ตรงกันทั้งหมด ใช้หน่วยความจำไม่เกินกว่าค่าที่กำหนดและสามารถรันโค้ดได้เสร็จสิ้นภายในระยะเวลาที่กำหนด หากไม่ถูกต้องทั้งหมดจะไม่ได้คะแนนในชุดทดสอบนั้น ๆ

3.3.2 การตรวจสอบคุณสมบัติของฟังก์ชัน (Check condition)

การตรวจสอบคุณสมบัติของฟังก์ชันนั้นอาศัยหลักการของ static code analysis เป็นการตรวจสอบจากโค้ดภาษาซีโดยตรง โดยเริ่มต้นตรวจการทำงานจากในฟังก์ชันที่กำหนด จนจบคำสั่งฟังก์ชันนั้น ๆ จะได้ทราบว่า ฟังก์ชันนั้นเป็นฟังก์ชัน recursive หรือไม่ และมี nested loop แต่ละชั้นอย่างละเท่าไร โดยมีขั้นตอนการตรวจสอบดังนี้



รูปที่ 3.6 การวิเคราะห์โค้ดโดยใช้ Static code analysis

1. รันคำสั่ง cflow ซึ่งเป็นคำสั่งใน GNU (อ่านว่ากนู) ด้วยการนำซอร์สไฟล์ที่ผ่านการลบคอมเม้น(ข้อความหลัง // และข้อความภายใน /*..*/) เป็นพารามิเตอร์พร้อมกับชื่อฟังก์ชันที่ต้องการตรวจสอบและตำแหน่งที่อยู่ไฟล์
2. ได้ผลลัพธ์จากการทำงานในคำสั่งข้อที่ 1 เป็นรายชื่อความสัมพันธ์ของฟังก์ชันที่ต้องการตรวจสอบ
3. เขียนโปรแกรมวิเคราะห์ฟังก์ชัน โดยที่ตรวจสอบว่าเป็นฟังก์ชัน recursion หรือไม่ และมีลูปซ้อนลูปอย่างละกี่จำนวน โดยตรวจสอบตั้งแต่บรรทัดแรกของฟังก์ชันที่ต้องการตรวจสอบ หากมีการเรียกฟังก์ชันอื่น (callee) ก็จะไปตรวจสอบที่ฟังก์ชันนั้นด้วยซึ่งจะไปเริ่มตั้งแต่บรรทัดแรกของฟังก์ชันนั้น ๆ การตรวจสอบเช่นเป็นการทำงานลักษณะเวียนบังเกิด (Recurrence relation) เพื่อเป็นจำลองแบบเดียวกันกับการทำงานของโปรแกรมจริง ๆ

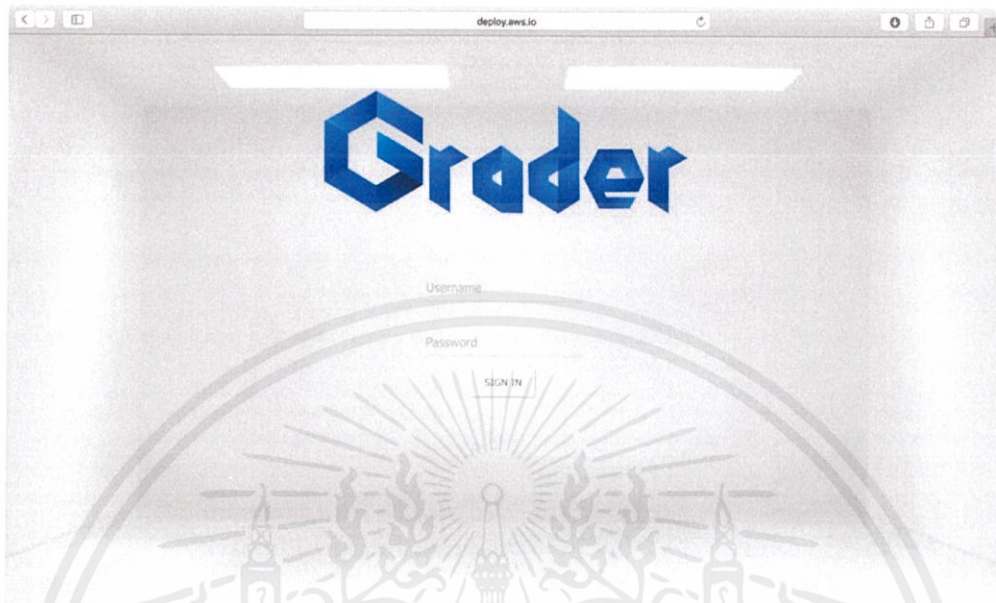
4. ได้ผลลัพธ์ว่าฟังก์ชันที่ต้องการตรวจสอบเป็นฟังก์ชันrecursionหรือไม่และมีลูปซ้อนลูปอย่างละกี่จำนวน โดยจำกัดลูปซ้อนลูปไว้สูงสุด 5 ชั้นเนื่องจากการเขียนโดยทั่วไปแล้วมักมีจำนวน nested loop ไม่เกิน 3 ชั้น(จากการเก็บตัวอย่างโค้ดของนักศึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

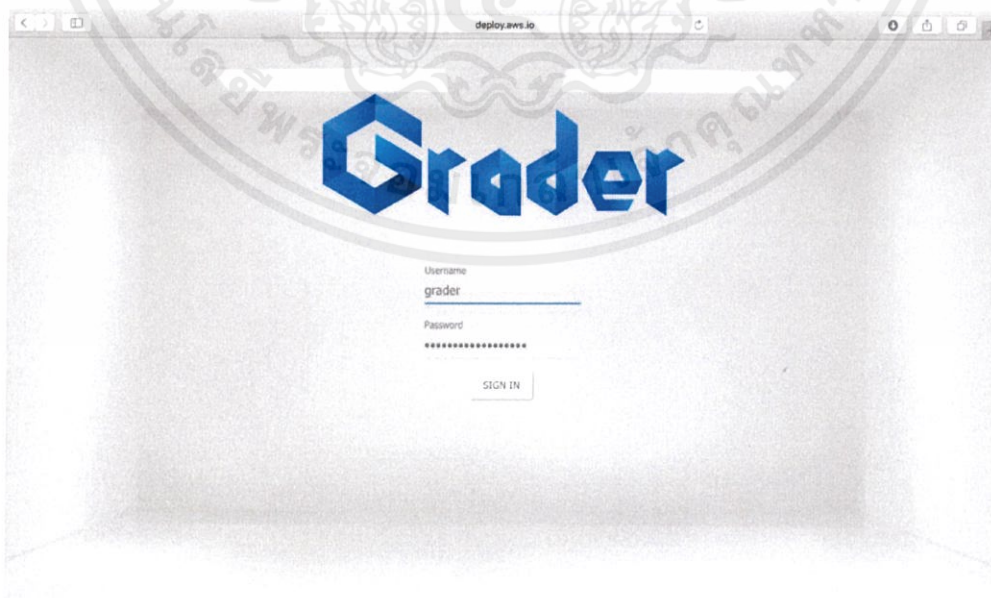
3.4 รูปแบบและการใช้งานโปรแกรม

3.4.1 เข้าสู่ระบบ(Login)



รูปที่ 3.7 หน้าเข้าสู่ระบบ (Login)

เริ่มต้นเข้าใช้งานระบบเราจะพบกับหน้าเข้าสู่ระบบ(Login) เป็นหน้าแรก ให้ผู้ใช้กรอก Username และ Password ที่ได้รับจากผู้ดูแลระบบ จากนั้นกด Sign In เพื่อเข้าใช้งานระบบดังภาพต่อไปนี



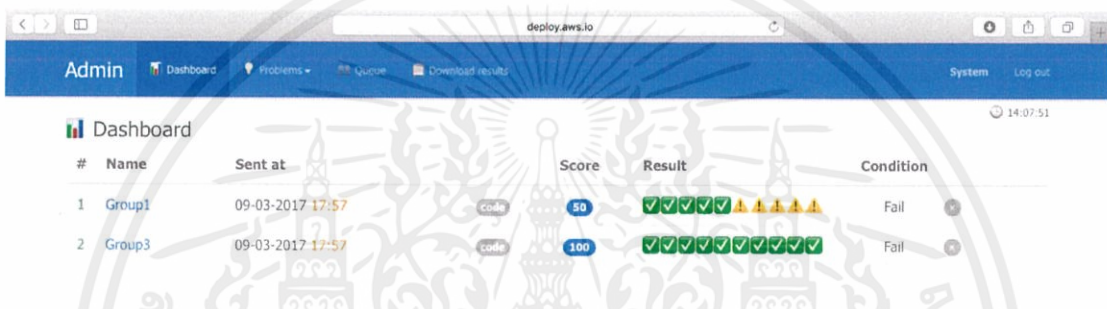
รูปที่ 3.8 ตัวอย่างหน้าเข้าสู่ระบบ (Login) เมื่อใส่ Username และ Password

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

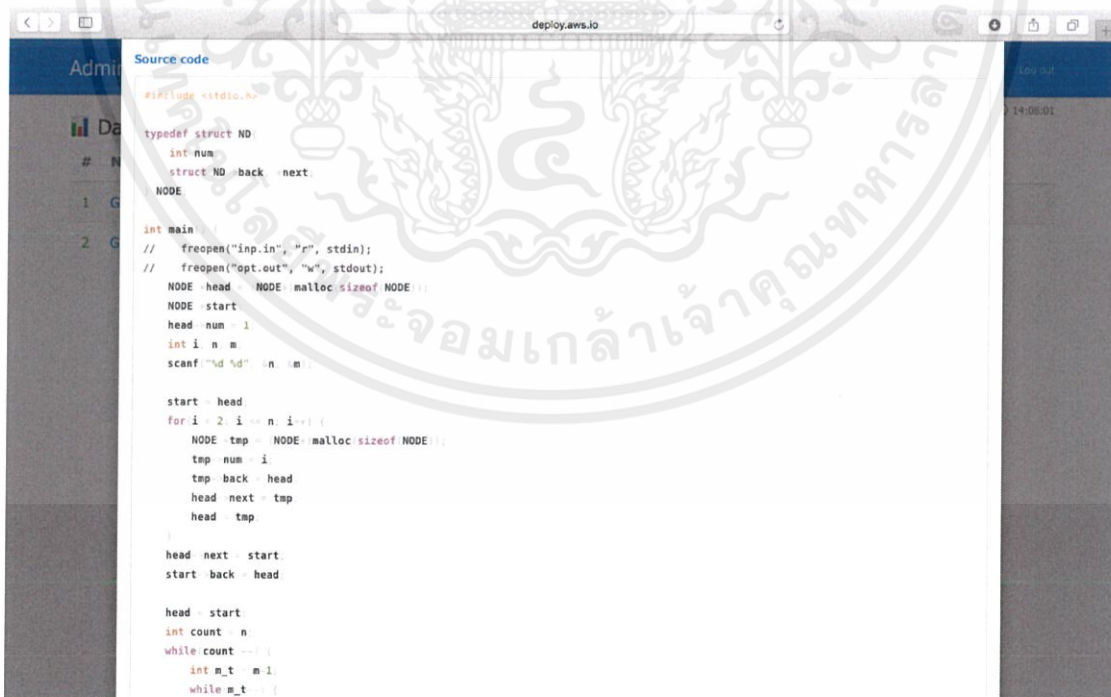
3.4.2 หน้าต่างผู้ใช้งานประเภทแอดมินหรืออาจารย์(Admin)

3.4.2.1 Dashboard

Dashboard คือหน้าต่างที่จะแสดงผลการส่งโค้ดของนักศึกษาแต่ละคน โดยจะแสดงวัน เวลา คะแนนที่ได้ จำนวน test case ที่ผ่าน และแสดงผลว่าผ่านเงื่อนไขที่ตั้งไว้หรือไม่ ซึ่งอาจารย์สามารถกดที่คำว่า code เพื่อเข้าไปตรวจสอบซอร์สโค้ดของนักศึกษาแต่ละคนได้ดังรูปที่ 3.10 ในส่วนของ Result จะแสดงผลลัพท์การรัน โค้ดว่าผ่าน test case หรือไม่ โดยจะมี test case ทั้งหมด 10 ชุด ซึ่งจะแสดงผลลัพท์เป็นสัญลักษณ์ต่างๆดังรูปที่ 3.11 อาจารย์สามารถนำมาส์ไปชี้ที่สัญลักษณ์เพื่ออ่านข้อผิดพลาดที่เกิดขึ้นได้



รูปที่ 3.9 หน้า Dashboard



รูปที่ 3.10 ตัวอย่างการแสดงผลซอร์สโค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

✓ Correct

หมายถึง โปรแกรมให้คำตอบถูกต้องในชุดทดสอบนั้นๆ และโค๊ดคะแนนเต็ม โดยปกติโค๊ดได้10คะแนน

✗ Wrong

หมายถึง โปรแกรมให้คำตอบผิดในชุดทดสอบนั้นๆ เช่น คำวนตอบออกมาผิด, รูปแบบคำตอบในตรงกันตัวอย่าง, อักษรตัวเล็ก-ตัวใหญ่

🕒 Time out

หมายถึง โปรแกรมใช้เวลารับเกินกว่าที่โจทย์ระบุไว้

⚠️ Error

หมายถึง โปรแกรมเกิดข้อผิดพลาดระหว่างการรัน หรือ ขนาดของโปรแกรม(File size)เกินกำหนด ซึ่งเกิดจาก Library ที่ include เข้ามา และ ตัวแปรที่ประกาศเป็นหลัก

🕒 waiting

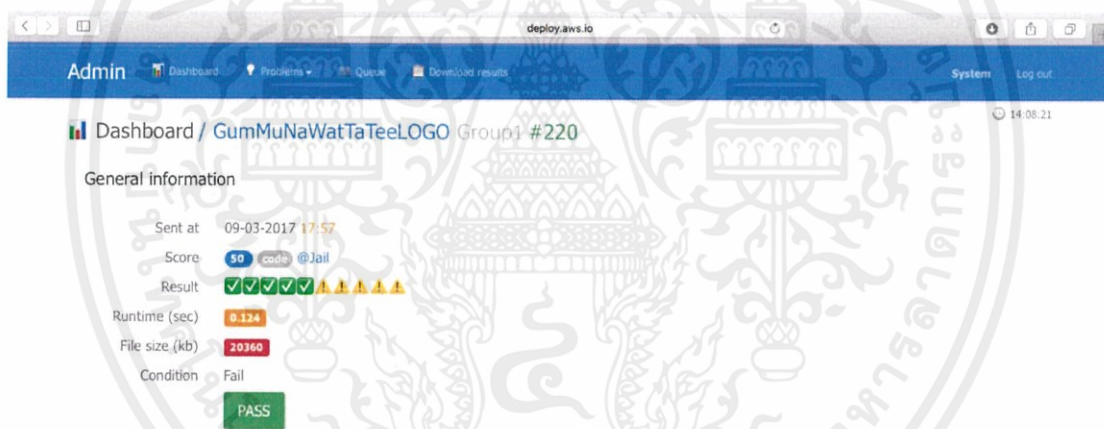
คือตรวจการประมวลผลจากระบบ(อยู่ในคิวระบบ)

🚫 error

คือคอมไพเลอร์ไม่ผ่าน คู่มือการคอมไพเลอร์ใน Submissions

รูปที่ 3.11 สัญลักษณ์ต่างๆ

นอกจากนั้นสามารถเลือกเข้าไปดูรายละเอียดการส่งแต่ละครั้งได้ ซึ่งหากอาจารย์เห็นว่าโค้ดของนักศึกษาถูกต้องสามารถกด Pass ให้การส่งครั้งนั้นๆได้



รูปที่ 3.12 หน้าแสดงรายละเอียดการส่งครั้งที่ 220

3.4.2.2 Problem Management

หน้า Problem Management จะแสดง โจทย์ทั้งหมดที่มีในระบบและใช้สำหรับจัดการเพิ่มหรือแก้ไข โจทย์ปัญหาต่างๆภายในระบบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Lab category	Name	Level	Score	Detect
ALL	SumEven	medium	100	-
11	Factorialz	easy	100	-
A	Grade	easy	100	main()
B	Starfall	easy	100	-
Test	Tiling	easy	100	main()
	Untitled	easy	100	-
	NewTill	easy	100	-
	Brute force	hard	100	-
	Car	hard	100	main()
	Jail	hard	100	main()
	Segment	hard	100	-
	abProblem	medium	100	main()

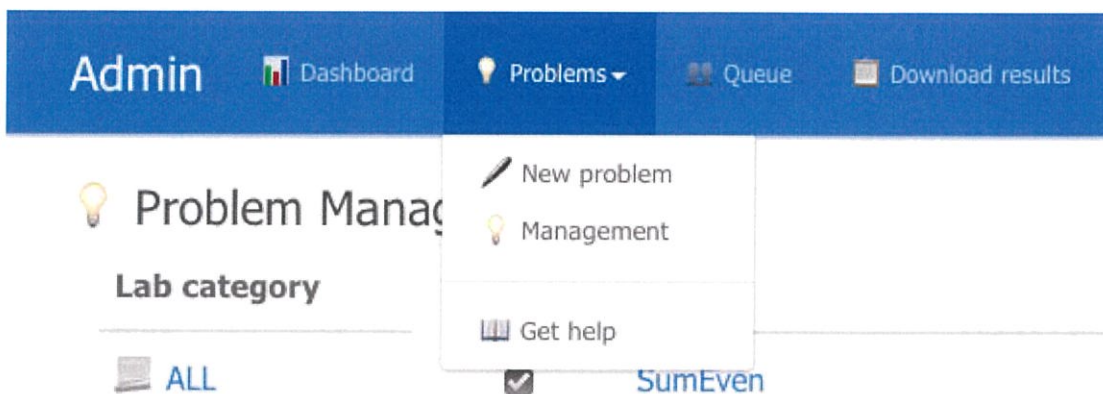
รูปที่ 3.13 หน้า Problem Management

นอกจากนี้ยังสามารถคัดกรองโจทย์จากชื่อแลบได้โดยคัดเลือกจาก Lab category ที่เคยสร้างไว้ ก็จะแสดงเฉพาะ โจทย์ที่เกี่ยวข้องกับแลบนั้นๆ ดังรูปที่ 3.14

Lab category	Name	Level	Score	Detect
ALL	Factorialz	easy	100	-
11	Grade	easy	100	main()
A	Starfall	easy	100	-
B	Tiling	easy	100	main()
Test	Untitled	easy	100	-
	NewTill	easy	100	-

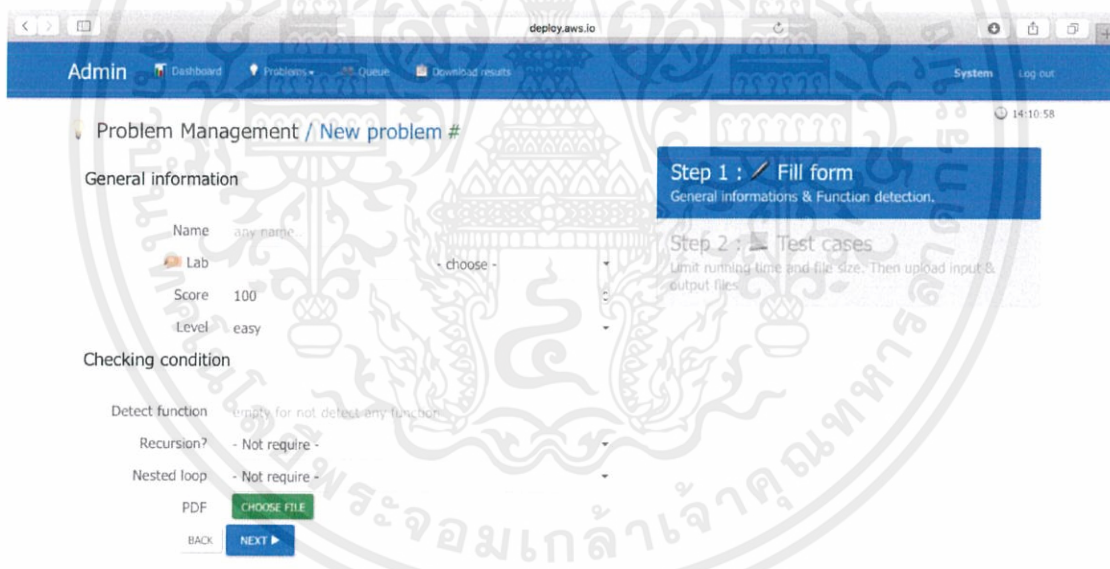
รูปที่ 3.14 แสดงโจทย์ของแลบ A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



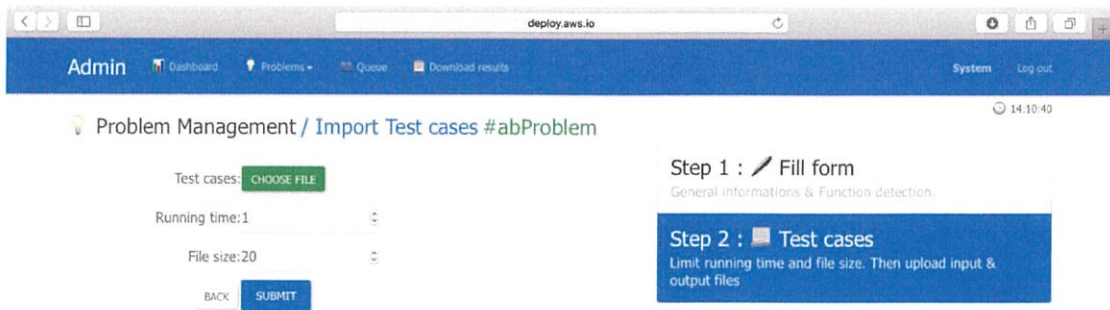
รูปที่ 3.15 Menu bar

วิธีการเพิ่มโจทย์ปัญหาให้เข้าไปที่เมนู Problem -> New problem จะมีด้วยกัน 2 ขั้นตอน ขั้นที่ 1 ใส่รายละเอียดเช่น ชื่อ ชื่อแถบ(หากเคยเพิ่มแถบไว้แล้วสามารถเลือกจาก Dropdown ได้เลยไม่ต้องกรอกใหม่) คะแนน ระดับความยากง่าย กำหนดเงื่อนไขการผ่านของโจทย์ ข้อนี้ๆ และเพิ่มไฟล์โจทย์เป็น pdf



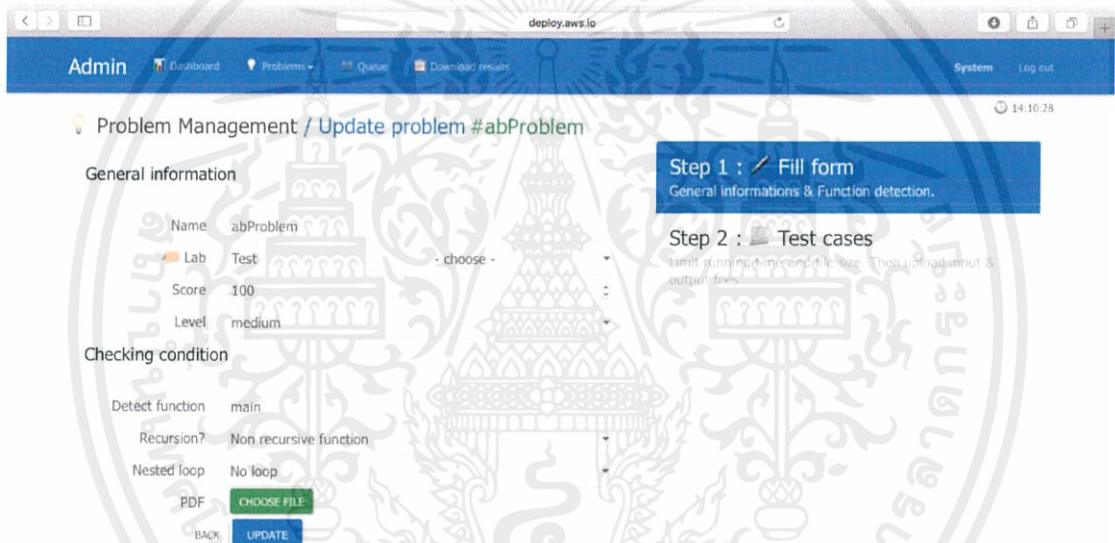
รูปที่ 3.16 เพิ่มโจทย์ปัญหา ขั้นที่ 1

ขั้นที่ 2 เพิ่ม test case สำหรับโจทย์ข้อนี้ๆ โดย test case จะประกอบด้วยไฟล์ อินพุตซึ่งใช้นามสกุลเป็น .in และ ไฟล์เอาต์พุตซึ่งใช้นามสกุลเป็น .sol อย่างละ 10 ไฟล์ จากนั้นทำการบีบอัดไฟล์เป็น .zip แล้วอัปโหลดเข้าสู่ระบบเป็นอันเสร็จสิ้นการเพิ่มโจทย์



รูปที่ 3.17 เพิ่ม test case ขั้นที่ 2

ในส่วนของการแก้ไขโจทย์จะคล้ายกับการเพิ่ม โจทย์โดยสามารถกดเข้าไปแก้ไข โจทย์ได้จากเมนู Problem -> Management แล้วเลือกที่โจทย์ที่ต้องการแก้ไขจะแสดงดังรูปที่ 3.18



รูปที่ 3.18 ตัวอย่างการแก้ไข Problem

3.4.3 หน้าต่างผู้ใช้งานประเภทนักศึกษา(Student)

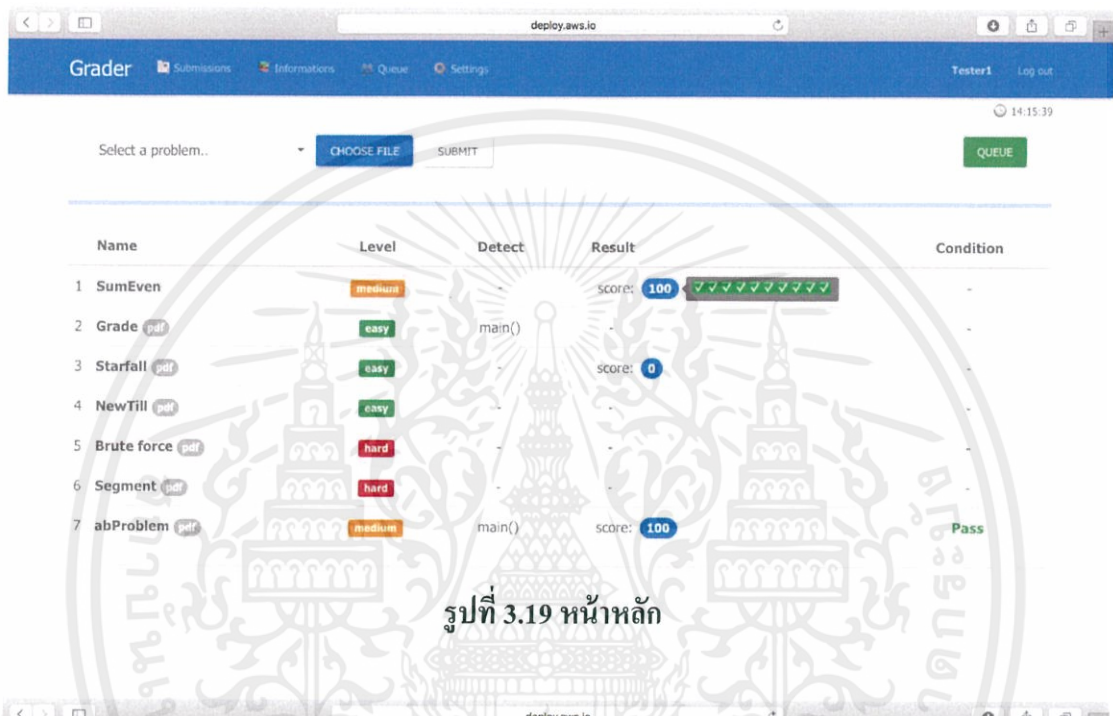
3.4.3.1 หน้าหลัก (Main Page)

หน้าหลักของนักศึกษาจะแสดงโจทย์ทั้งหมดที่อาจารย์เปิดไว้ให้ โดยจะแสดง รายละเอียดของโจทย์แต่ละข้อตามรูปที่ 3.19 ดังนี้

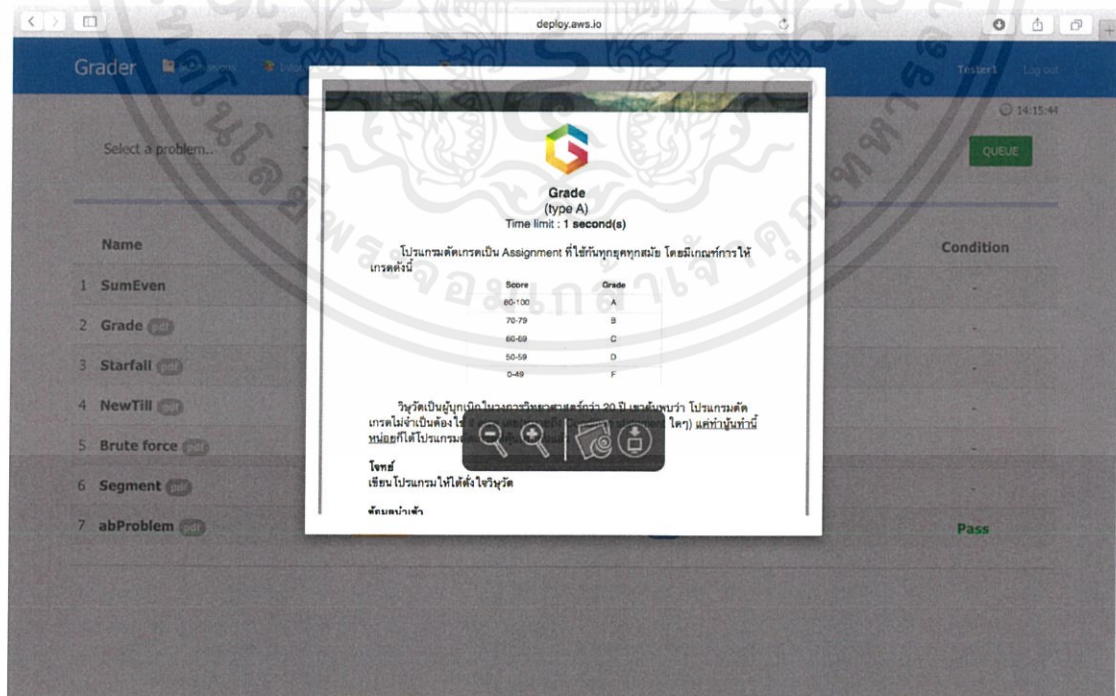
- Name : ชื่อ โจทย์
- Level : ระดับความยากง่ายของ โจทย์
- Detect : ระบุชื่อฟังก์ชันที่จะทำการตรวจสอบว่าทำงานตรงตามเงื่อนไขที่ อาจารย์ตั้งไว้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Result : แสดงคะแนนที่ได้ เมื่อส่งโค้ดโปรแกรมแล้ว และแสดงสถานะของทั้ง 10 test case ว่าผ่านครบหรือไม่ หากไม่ผ่านนักศึกษาสามารถนำเมาส์ไปชี้เพื่ออ่านข้อผิดพลาดที่เกิดขึ้นได้
- Condition : จะแสดงผลการรันว่าผ่านตามเงื่อนไขที่อาจารย์ตั้งไว้หรือไม่
- ปุ่ม pdf : แสดงรายละเอียดโจทย์เป็นไฟล์ pdf ซึ่งจะแสดงดังรูปที่ 3.20



รูปที่ 3.19 หน้าหลัก



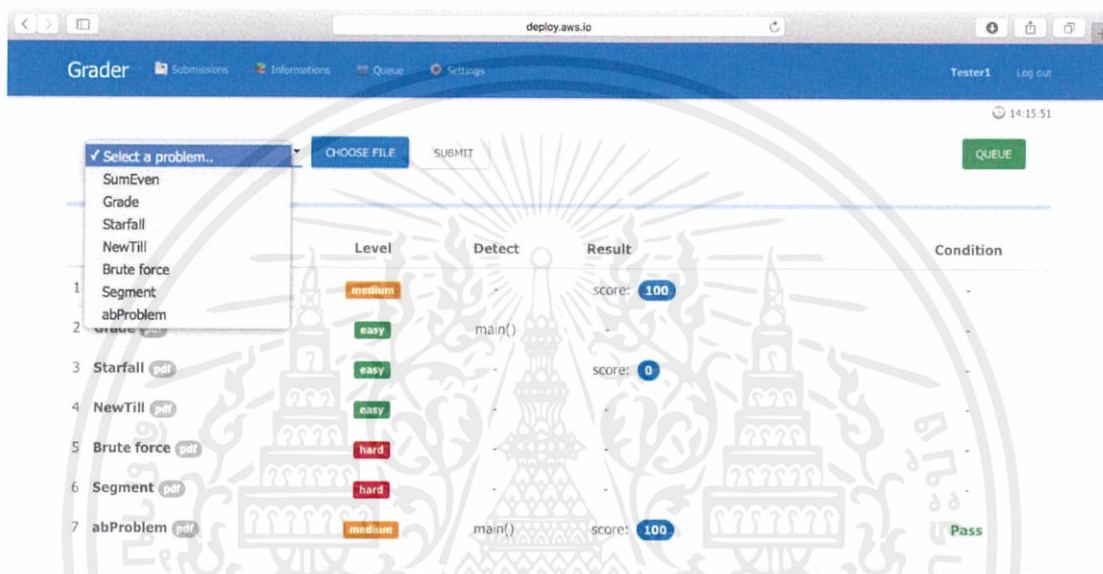
รูปที่ 3.20 แสดงรายละเอียดโจทย์เป็น pdf

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ยังสามารถส่งซอร์สโค้ดได้จากหน้านี้ โดยมีวิธีการดังนี้

วิธีการ Upload file

1. เลือกโจทย์ปัญหาที่ต้องการทำการส่ง
2. กด Choose File
3. กด Submit
4. รอระบบตรวจสักรุ่นแล้วกด refresh เพื่อดูคะแนนที่ได้



รูปที่ 3.21 ตัวอย่างการส่งซอร์สโค้ด

3.4.3.2 หน้า Submission

หน้า Submission จะแสดงประวัติการส่งงานของนักศึกษา โดยสามารถเข้าสู่หน้า Submission ได้จากเมนูด้านบน จากนั้นนักศึกษาสามารถเลือกดูประวัติการส่งของโจทย์แต่ละข้อได้ ซึ่งในตอนต้นจะแสดงการส่งครั้งสุดท้ายไว้ให้ก่อนดังรูปที่ 3.22

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองระบบครั้งที่ 1


คณะผู้จัดทำได้นำระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซีไปทดลองใช้ในวิชาปฏิบัติการของวิชา Data Structure & algorithm ในช่วงครึ่งหลังของภาคเรียนที่ 2 ปีการศึกษา 2558 โดยผู้ที่ทดลองใช้ระบบได้แก่ TA ประจำวิชาและนักศึกษา

4.1.1 สภาพแวดล้อมที่ใช้ในระบบ

1. เครื่องคอมพิวเตอร์ในห้องปฏิบัติการ 706
2. Software : VM Ware
3. Ubuntu version 14.04

4.1.2 ขั้นตอนการทดลอง

1. ติดตั้ง VM Ware ในเครื่องเซิร์ฟเวอร์ เพื่อติดตั้ง Ubuntu และระบบที่เตรียมไว้
2. ตั้งค่า IP ในเครื่องเซิร์ฟเวอร์เพื่อให้นักศึกษาสามารถเข้าใช้งานได้
3. เพิ่มรายชื่อนักศึกษาและ TA ลงในฐานข้อมูล
4. เพิ่มโจทย์ปัญหา
5. แจกจ่าย username และ password ให้กับนักศึกษาทุกคน เพื่อเข้าใช้งานระบบ



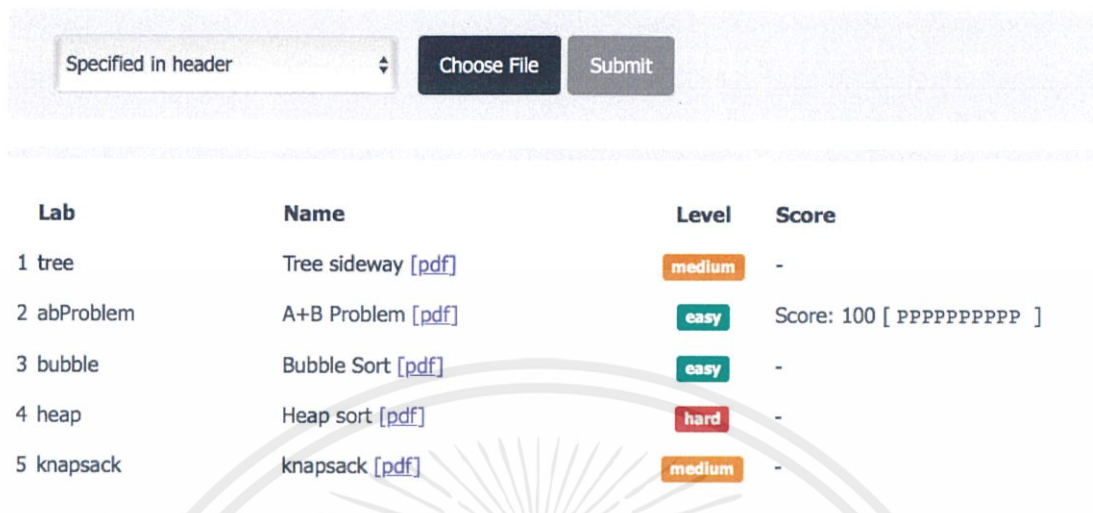
Login: user1

Password: ●●●●

Login

รูปที่ 4.1 หน้า log in ของระบบ

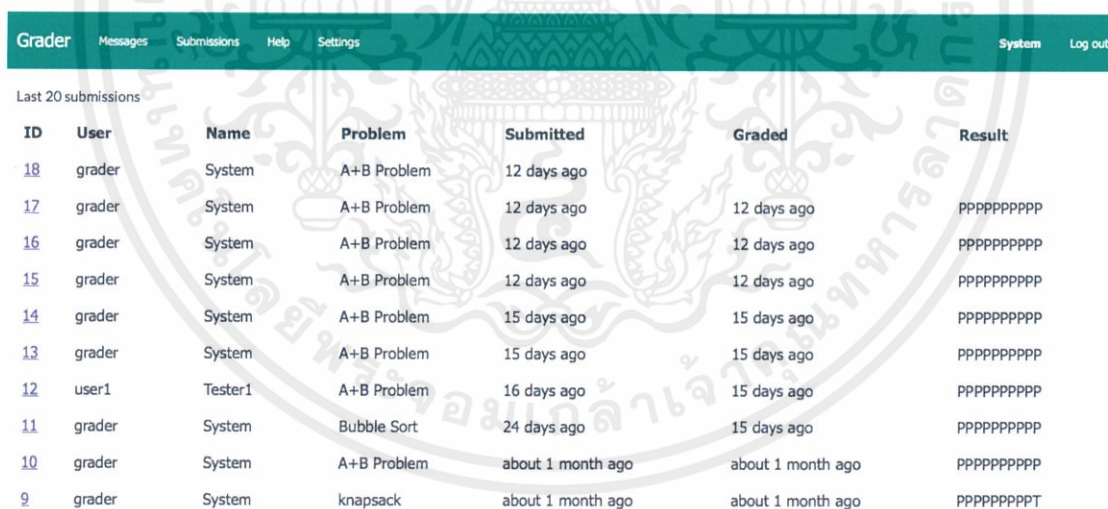
6. นักศึกษาส่งซอร์สโค้ดให้ระบบตรวจสอบ



Lab	Name	Level	Score
1 tree	Tree sideways [pdf]	medium	-
2 abProblem	A+B Problem [pdf]	easy	Score: 100 [P P P P P P P P P P]
3 bubble	Bubble Sort [pdf]	easy	-
4 heap	Heap sort [pdf]	hard	-
5 knapsack	knapsack [pdf]	medium	-

รูปที่ 4.2 หน้าส่งซอร์สโค้ดของนักศึกษา

7. TA สามารถเข้าดูซอร์สโค้ดของนักศึกษาแต่ละคน และสามารถรู้ถึงผลการรันโปรแกรมได้



ID	User	Name	Problem	Submitted	Graded	Result
18	grader	System	A+B Problem	12 days ago		
17	grader	System	A+B Problem	12 days ago	12 days ago	PPPPPPPPPP
16	grader	System	A+B Problem	12 days ago	12 days ago	PPPPPPPPPP
15	grader	System	A+B Problem	12 days ago	12 days ago	PPPPPPPPPP
14	grader	System	A+B Problem	15 days ago	15 days ago	PPPPPPPPPP
13	grader	System	A+B Problem	15 days ago	15 days ago	PPPPPPPPPP
12	user1	Tester1	A+B Problem	16 days ago	15 days ago	PPPPPPPPPP
11	grader	System	Bubble Sort	24 days ago	15 days ago	PPPPPPPPPP
10	grader	System	A+B Problem	about 1 month ago	about 1 month ago	PPPPPPPPPP
9	grader	System	knapsack	about 1 month ago	about 1 month ago	PPPPPPPPPT

รูปที่ 4.3 หน้าประวัติการส่งงานของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Submission: 17
Stat

Field	Value
User:	(grader) System
Problem:	(abProblem) A+B Problem
Tries:	10
Submitted:	12 days ago
Graded:	12 days ago
Points:	100/100
Result:	PPPPPPPPPP
Runtime (s):	0.0
Memory (kb):	1696

Source code

```

/*
LANG: C++
TASK: abProblem
*/
#include<iostream>

using namespace std;

int main ()
{
    int a b;
    scanf ("%d %d", &a, &b);
    printf ("%d" a b);
    return 0;
}

```

รูปที่ 4.4 หน้าดูซอร์สโค้ดและรายละเอียดการส่งงาน

4.1.3 ผลการทดลอง

สรุปผลการทดลองครั้งที่ 1 จากการทดลองครั้งที่ 1 ได้มีการให้นักศึกษาทุกคนทำแบบสอบถามเพื่อนำผลลัพธ์ที่ได้มาประเมินผลต่อไป

Grader Evaluation

Grader เป็นระบบที่ช่วยตรวจโค้ดโปรแกรมมี Feedback ที่จะช่วยพัฒนาระบบให้เหมาะสมกับการใช้งานมากขึ้น

รหัสนักศึกษา *

Short answer text

การเขียนโปรแกรม ทำใจทยอยแต่ละข้อ *

- พอเขียนได้
- ใช้เวลานานหน่อย
- คิดว่า ค่อนข้างยากสำหรับตัวเอง

รูปที่ 4.5 แบบสอบถามความพึงพอใจภาคเรียนที่ 2/2558 (ส่วนที่ 1)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่เห็นว่าเกิดขึ้น"ค่อนข้างบ่อย"

- อ่านโจทย์แล้วไม่เข้าใจ ใช้เวลาทำความเข้าใจนาน
- แสดงผลลัพธ์ผิด ทำให้ไม่ผ่านเคส แม้ว่าผิดเพียงเล็กน้อย
- เวลาผลลัพธ์คะแนนไม่เต็ม[P-P-P] มักไม่รู้ตัวเองไม่ผ่านในกรณีไหนเป็นเวลานาน
- ระบบมักตอบสนองช้า หรือลื่นอยู่บ่อยๆ
- มัก Compile error บน Grader
- Other..

รูปที่ 4.6 แบบสอบถามความพึงพอใจภาคเรียนที่ 2/2558 (ส่วนที่ 2)

ถ้าหาก"ไม่มีปัญหา"ดังกล่าว เห็นด้วยกับสิ่งนี้หรือไม่?

- Grader ให้คำตอบรวดเร็วกว่าการตรวจแบบปกติ
- หากใช้ Grader ในช่วงการสอบ จะลดเวลาในการรออาจารย์มาตรวจคำตอบ
- Grader น่าจะเหมาะกับวิชา Computer Programming 1 Lab ด้วย
- ถ้าเซิร์ฟเวอร์ออนไลน์ตลอดเวลา จะสะดวกในการส่งLab จะได้ส่งเมื่อไรก็ได้

เรื่องทั่วไป

และข้อเสนอแนะ

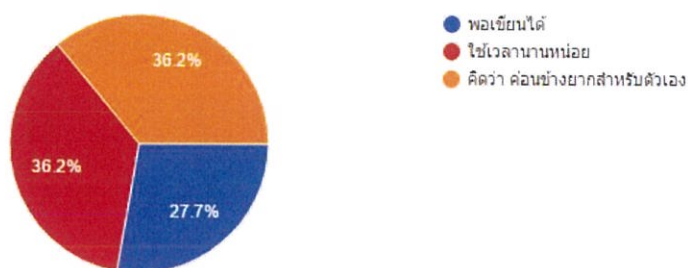
- อยากให้มีการจัดแข่งเขียนโปรแกรมแบบซ้ำๆ xD
- สนใจ Project Grader [Web Application] ว่าใช้เทคโนโลยีอะไรบ้างในการสร้าง
- อยากได้ Grader ไปประยุกต์ใช้ในงานอื่นๆ
- Other..

รูปที่ 4.7แบบสอบถามความพึงพอใจภาคเรียนที่ 2/2558(ส่วนที่ 3)

ผลที่ได้จากการตอบแบบสอบถามของนักศึกษาทั้ง 47 คนเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

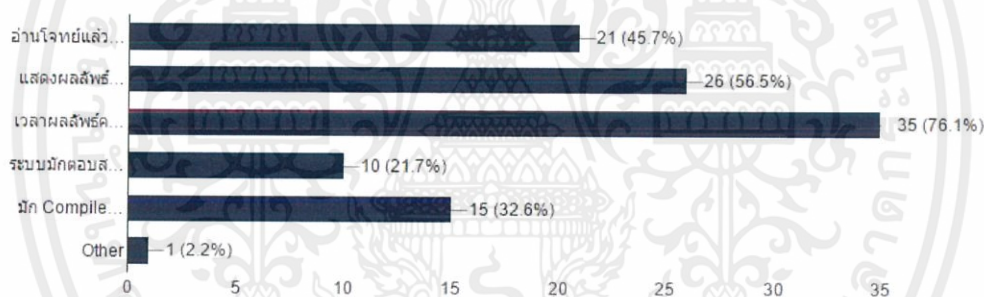
การเขียนโปรแกรม ทำโจทย์แลปแต่ละข้อ (47 responses)



รูปที่ 4.8 ผลที่ได้จากการตอบแบบสอบถามข้อ 1

จากผลการตอบแบบสอบถามพบว่านักศึกษาส่วนใหญ่คิดว่าโจทย์ค่อนข้างยากสำหรับตัวเอง และสามารถทำโจทย์ได้แต่ต้องใช้เวลาานสักหน่อยในสัดส่วนที่เท่าๆกัน น้อยที่สุดคือพอเขียนได้ แต่ก็ไม่ได้แตกต่างกันมากนัก

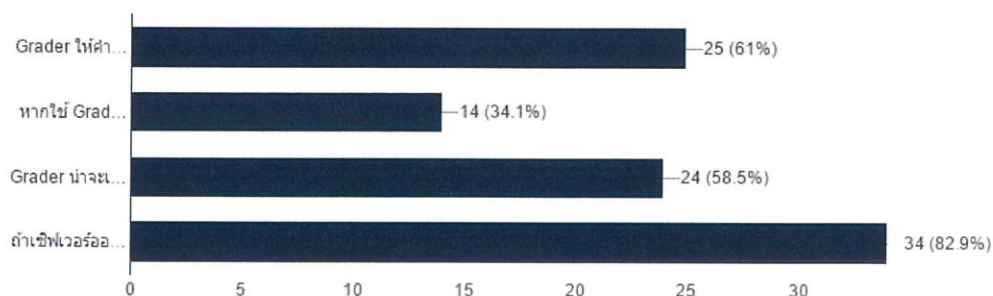
ปัญหาที่เห็นว่าเกิดขึ้น "ค่อนข้างบ่อย" (46 responses)



รูปที่ 4.9 ผลที่ได้จากการตอบแบบสอบถามข้อ 2

พบว่าปัญหาที่เกิดขึ้นค่อนข้างบ่อยคือเมื่อทำโจทย์แล้วไม่ได้คะแนนเต็ม(ไม่ผ่าน test case ทุกกรณี) มักจะไม่รู้ว่าตัวเองไม่ผ่านในกรณีไหนเป็นเวลานาน เนื่องจากระบบไม่ได้แสดง test case ที่ใช้ทั้งหมดให้กับนักศึกษาแต่ละจะบอกแค่ช่วงคำตอบที่เป็นไปได้เท่านั้น ที่เป็นเช่นนี้เพื่อป้องกันนักศึกษาเขียนโค้ดแบบจะงาคำตอบ ส่วนปัญหาที่พบน้อยที่สุดรองจาก other คือระบบตอบสนองช้าหรือลุ่มอยู่บ่อยๆ ซึ่งแสดงถึงประสิทธิภาพการทำงานที่ดีของระบบ

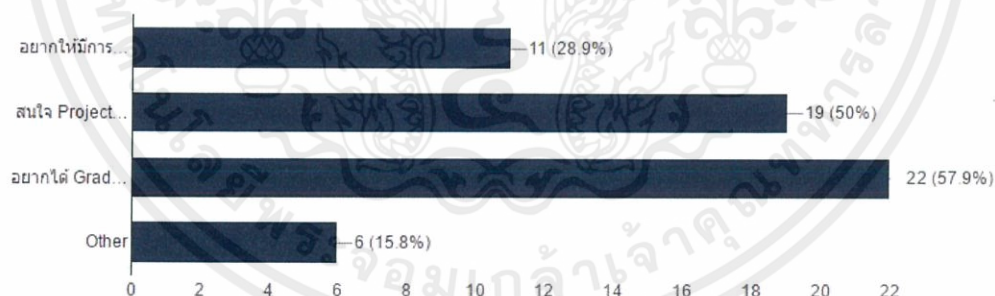
ถ้าหาก"ไม่มีปัญหา"ดังกล่าว เห็นด้วยกับสิ่งนี้หรือไม่? (41 responses)



รูปที่ 4.10 ผลที่ได้จากการตอบแบบสอบถามข้อ 3

จากปัญหาข้างต้นพบว่าหากไม่มีปัญหาดังกล่าวนักศึกษาเห็นด้วยกับข้อที่ว่า “ถ้าเซิร์ฟเวอร์ออนไลน์ตลอดเวลา จะสะดวกในการส่งแลบมากขึ้น จะส่งเมื่อไหร่ก็ได้” มากที่สุด ซึ่งทำให้นักศึกษาสามารถส่งแลบนอกห้องเรียนได้จากการพิจารณาแล้วทางคณะผู้จัดทำเห็นว่าการที่ระบบเป็นออฟไลน์นั้นเพียงพอต่อการให้ช่วยเหลือการจัดการในห้องปฏิบัติการซึ่งระบบออนไลน์เป็นฟีเจอร์ที่น่าสนใจในอนาคต ส่วนข้อรองลงมาคือ “Grader ให้คำตอบรวดเร็วกว่าการตรวจแบบปกติ” แสดงว่าระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซีสามารถช่วยเหลือการจัดการภายในห้องปฏิบัติการได้จริงตามจุดประสงค์

เรื่องทั่วไป (38 responses)



รูปที่ 4.11 ผลที่ได้จากการตอบแบบสอบถามข้อ 4

สำหรับเรื่องทั่วไปพบว่านักศึกษายกได้ระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซีไปประยุกต์ใช้ในงานอื่นๆมากที่สุด และรองลงมาคือสนใจว่าระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซีใช้เทคโนโลยีอะไรบ้างในการสร้าง

4.2 การทดลองระบบตรวจสอบสมบัติฟังก์ชัน

ผู้จัดทำได้นำซอร์สโค้ดของนักศึกษาในวิชาปฏิบัติการของวิชา Data structure & algorithm ภาคเรียนที่ 2 ปีการศึกษา 2558 ที่ถูกเก็บไว้ในเซิร์ฟเวอร์กว่า 900 ซอร์สโค้ด มาใช้ในการทดสอบนี้ เพื่อวัดประสิทธิภาพของระบบตรวจสอบสมบัติฟังก์ชันว่ามีความแม่นยำเพียงใด โดยมีสภาพแวดล้อมที่ใช้ในการทดลองและขั้นตอนการทดลองดังนี้

4.2.1 สภาพแวดล้อมที่ใช้ในระบบ

1. Operating System : Ubuntu server version 14.04
2. RAM : 1 GB
3. CPU : core i5 2.4 GHz
4. Disk space : 7 GB
5. Software : VMware Fusion version 8.0.1
6. ForkLift โปรแกรมจัดการ File Transfer Protocol (FTP)

4.2.2 ขั้นตอนการทดลอง

1. เขียนไฟล์ตรวจสอบสมบัติเพิ่มเติมเพื่อให้รองรับการตรวจไฟล์หลาย record บนดาต้าเบสและสามารถเขียนผลลัพธ์ลงไฟล์เดิมเพื่อบอกว่าแต่ละ record ได้ผลลัพธ์การตรวจเป็นอย่างไร โดยชื่อฟังก์ชันที่ตรวจเป็นฟังก์ชันชื่อ main() ซึ่งเป็นฟังก์ชันหลัก
2. ส่งไฟล์ตรวจสอบสมบัติฟังก์ชันผ่านโปรแกรม FTP ไปยังเซิร์ฟเวอร์บน VMware
3. คอมไพล์ไฟล์ดังกล่าวซึ่งไฟล์เป็นไฟล์ภาษาซีพลัสพลัส(C++ file) ด้วยคอมไพเลอร์ g++ เหตุผลที่ต้องนำไฟล์ไปคอมไพล์บนเซิร์ฟเวอร์เพื่อที่จะให้ตัวประมวลผลหรือ execute file เป็นในสภาพแวดล้อมเดียวกันกับระบบปฏิบัติการกันปัญหาคำสั่งที่ไม่รู้จักที่จะเกิดขึ้นขณะคอมไพล์
4. นำข้อมูลการส่งของนักศึกษาจาก database มาจัดเตรียมก่อนนำไปประมวลผลโดยเลือกเฉพาะคอลัมน์ source file
5. นำข้อมูลจากข้อ 4 เป็นพารามิเตอร์สำหรับ execute file ในข้อ 3 จะได้ผลลัพธ์การส่งทั้งหมดโดยที่จะทราบว่าเป็น Recursion หรือไม่และมี nested loop 1, 2, 3, 4, 5 ชั้นอย่างละกี่ตัว
6. หาประสิทธิภาพของการตรวจด้วยการหาผู้ที่มีความสามารถในการอ่านโค้ดขั้นพื้นฐานว่าโค้ดนั้นเป็นลักษณะ Recursion หรือไม่และนับจำนวนลูปที่เกิดขึ้นจากนั้น

เปรียบเทียบกับระบบประมวลผลว่าคำตอบถูกต้องทั้งหมดเป็นจำนวนเท่าไรและคิดออกมาเป็นเปอร์เซ็นต์

ตารางที่ 4.1 อธิบายสัญลักษณ์และความหมาย

ลำดับ	สัญลักษณ์	ความหมาย
1	R	ฟังก์ชันนั้นเป็น recursive function หรือมีการเรียก recursive function
	N	ฟังก์ชันนั้นไม่เป็น recursive function
	D	ไม่สามารถค้นหาฟังก์ชันตามชื่อที่กำหนดได้
	E	ระบบไม่สามารถทำงานได้อย่างถูกต้อง
2	0, 1, 2, ...	จำนวน Nested loop 1 ชั้นที่พบ
3	0, 1, 2, ...	จำนวน Nested loop 2 ชั้นที่พบ
4	0, 1, 2, ...	จำนวน Nested loop 3 ชั้นที่พบ
5	0, 1, 2, ...	จำนวน Nested loop 4 ชั้นที่พบ
6	0, 1, 2, ...	จำนวน Nested loop 5 ชั้นที่พบ

ตัวอย่างที่ 1

ซอร์สโค้ดเป็นดังนี้

```
#include<stdio.h>
int main()
{
    int i,j;
    scanf("%d %d",&i,&j);
    printf("%d\n",i+j);
    return 0;
}
```

รูปที่ 4.12 ตัวอย่างซอร์สโค้ดที่ 1

ผลลัพธ์การทำงานของระบบตรวจสอบสมบัติฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

N 0 0 0 0 0

รูปที่ 4.13 ตัวอย่างผลการรันจากซอร์สโค้ดที่ 1

ตัวอย่างที่ 2

```
#include<stdio.h>

void rec(int k){
    if(k==0) return;
    rec(k-1);
}

int main()
{
    int a,b;
    int count = 3;
    for (int i = 0; i < count; ++i)
        for (int j = 0; j < count; ++j)
            printf("hello word");
    rec(3);
    printf("%d\n",a+b);
    return 0;
}
```

รูปที่ 4.14 ตัวอย่างซอร์สโค้ดที่ 2

ผลลัพธ์การทำงานของระบบตรวจสอบสมบัติฟังก์ชัน

R 0 1 0 0 0

รูปที่ 4.15 ตัวอย่างผลการรันซอร์สโค้ดที่ 2

4.2.3 ผลการทดลอง

ตารางที่ 4.2 ผลการทดลองและความผิดพลาดที่พบ

Detect Result	R+N(Complete)	D(unDetect)	E(Error)	SUM
Amount(files)	880	20	4	904

จากโค้ดทั้งหมด 904 ไฟล์สามารถตรวจและหาผลลัพธ์ได้ 880 ไฟล์ไม่สามารถหาฟังก์ชัน main ได้ 20 ไฟล์และระบบเกิดความผิดพลาดจากการตรวจฟังก์ชัน 4 ไฟล์จากการตรวจสอบของไฟล์ที่ไม่สามารถตรวจหาฟังก์ชัน main ได้ พบว่ามีสาเหตุมาจากไม่ได้ส่งไฟล์ที่เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษาซีหรือภาษา C++ และไฟล์ที่ทำให้ระบบเกิดความผิดพลาดทั้ง 4 ไฟล์ พบว่ามีสาเหตุมาจากการเขียนโปรแกรมที่ไม่ถูกต้อง

จากข้อมูลข้างต้นสามารถสรุปได้ว่า หากไม่คำนวณไฟล์ที่ผิดปกติแบบ(ไฟล์ที่ไม่สามารถหา main ได้) จะได้ค่าความแม่นยำของการตรวจคุณสมบัติฟังก์ชันเป็น 99.54 %ซึ่งจากข้อผิดพลาดดังกล่าวจะนำไปแก้ไขในส่วนของ การตรวจสอบต่อไปเพื่อให้ได้ผลลัพธ์ที่น่าพอใจ

4.3 การทดลองระบบครั้งที่ 2

วันที่ 9 มีนาคม พ.ศ.2559 คณะผู้จัดทำได้นำระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซีไปทดลองใช้ในวิชาปฏิบัติการของวิชา Computer Programming 1 ภาคเรียนที่ 2 ปีการศึกษา 2559 โดยผู้ทดลองใช้ระบบในการทดลองครั้งนี้คือนักศึกษาจำนวน 38คน

4.3.1 สภาพแวดล้อมที่ใช้ในระบบ

1. เครื่องคอมพิวเตอร์ในห้องปฏิบัติการ 706
2. Software : VM Ware
3. Ubuntu version 14.04

4.3.2 ขั้นตอนการทดลอง

1. ติดตั้ง VM Ware ในเครื่องเซิร์ฟเวอร์ เพื่อติดตั้ง Ubuntu และระบบที่เตรียมไว้
2. ตั้งค่า IP ในเครื่องเซิร์ฟเวอร์เพื่อให้นักศึกษาสามารถเข้าใช้งานได้
3. เพิ่มรายชื่อนักศึกษาทุกคนลงในฐานข้อมูล
4. เพิ่มโจทย์ปัญหา

ตัวอย่าง: จงรับค่าจำนวนเต็มวนไปจนกว่าค่าที่ป้อนเข้ามาจะเป็น 0 จากนั้นให้แสดงผลรวมเลขคู่ทั้งหมดและแสดงจำนวนของเลขคู่ทั้งหมดออกมา

ตัวอย่างอินพุต

10 4 9 1 3 5 9 4 10 1 6 3 3 8 4 8 10 0

ตัวอย่างเอาต์พุต

64

9

จะเห็นได้ว่า 64 คือผลรวมเลขคู่ทั้งหมดที่เป็นอินพุต ได้แก่ $10+4+4+10+6+8+4+8+10$

และ 9 คือจำนวนเลขคู่ทั้งหมด

ตัวอย่างโค้ดในการแก้ปัญหา

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main(int argc, char const *argv[])
6  {
7      int count = 0;
8      int sum = 0;
9      int temp;
10     while(true){
11         scanf("%d", &temp );
12         if (temp == 0)
13             break;
14         if (temp % 2 == 0)
15         {
16             sum += temp;
17             count++;
18         }
19     }
20     printf("%d\n%d\n", sum, count);
21     return 0;
22 }

```

รูปที่ 4.16 ตัวอย่างโค้ดในการแก้ปัญหา

5. แจกจ่าย username และ password ให้กับนักศึกษาทุกคน เพื่อเข้าใช้งานระบบ
6. นักศึกษาอัปโหลดชอว์สโค้ดเข้าสู่ระบบและทราบถึงผลลัพธ์ที่ได้จากการตรวจสอบ
7. ผู้ดูแลระบบสามารถเข้าดูชอว์สโค้ดของนักศึกษาแต่ละคน และสามารถรู้ถึงผลการรันโปรแกรมได้



รูปที่ 4.17 ภาพบรรยากาศการทดลองระบบครั้งที่ 2



รูปที่ 4.18 ภาพบรรยากาศการทดลองระบบครั้งที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 ภาพบรรยากาศการทดลองระบบครั้งที่ 2



รูปที่ 4.20 ภาพบรรยากาศการทดลองระบบครั้งที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 ภาพบรรยากาศการทดลองระบบครั้งที่ 2



รูปที่ 4.22 ภาพบรรยากาศการทดลองระบบครั้งที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 ผลการทดลอง

จากการทดลองพบว่าระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซีสามารถทำงานได้ตามจุดประสงค์ที่มุ่งหวังไว้ นักศึกษาทุกคนสามารถเข้าใช้งานระบบได้จริง สามารถตรวจคำตอบและสามารถเช็คเงื่อนไขตามที่กำหนดได้อย่างถูกต้อง

ผู้ทดลองใช้ระบบในการทดลองครั้งนี้คือนักศึกษาจำนวน 38 คน โดยมีผู้ส่งซอร์สโค้ดผ่านจำนวน 33 คน และมีผู้ที่ไม่ผ่านจำนวน 5 คน โดยมีสาเหตุมาจากโค้ดของนักศึกษาทำงานไม่ถูกต้อง หรือแสดงผลลัพธ์ไม่ถูกต้องตามรูปแบบที่กำหนดเป็นต้น

จากการทดลองคณะผู้จัดทำได้พบปัญหาเล็กน้อยที่ส่งผลให้การตรวจสอบผิดพลาด ซึ่งเกิดจากการที่นักศึกษาเขียนอินพุตและเอาต์พุตไม่ตรงตามรูปแบบที่กำหนด ทำให้การตรวจสอบคลาดเคลื่อนไป ซึ่งทางคณะผู้จัดทำแก้ปัญหาโดยการเข้าไปตรวจสอบซอร์สโค้ดที่นักศึกษาส่งเข้ามาในระบบและแจ้งถึงสาเหตุกลับไปยังนักศึกษา เพื่อให้นักศึกษาได้ทำการแก้ไขและส่งซอร์สโค้ดกลับเข้ามาในระบบอีกครั้ง

4.4 การแข่งขันการเขียนโปรแกรมภาษาซี

การแข่งขันการเขียนโปรแกรมภาษาซีได้ถูกจัดขึ้นในวันที่ 9 มีนาคม พ.ศ.2559 โดยมีผู้เข้าร่วมการแข่งขันทั้งหมด 8 ทีม ซึ่งกติกาการแข่งขันในครั้งนี้จะทำการแข่งขันเป็นทีมทีมละ 2 คน โดยที่โจทย์จะแบ่งเป็น 2 ระดับ ได้แก่ระดับยากจำนวน 4 ข้อและระดับปานกลางจำนวน 5 ข้อ ผู้เข้าแข่งขันจะต้องทำการปรึกษากันภายในทีมว่าจะเลือกทำโจทย์ระดับใด ซึ่งแต่ละทีมสามารถเลือกทำได้เพียงระดับเดียวเท่านั้น ให้อเวลาในการแข่งขัน 90 นาที ผู้เข้าแข่งขันจะทราบโจทย์ทั้งหมดเมื่อเริ่มการแข่งขันซึ่งทำให้ผู้เข้าแข่งขันต้องอ่านโจทย์และตัดสินใจให้ไวที่สุดเพื่อให้เหลือเวลาในการเขียนโค้ดมากที่สุด โดยทีมที่ชนะเลิศการแข่งขันในโจทย์ระดับยากจะได้รับรางวัลมูลค่า 2000 บาท และทีมที่ชนะเลิศการแข่งขันในโจทย์ระดับปานกลางจะได้รับรางวัลมูลค่า 1000 บาท

4.4.1 สภาพแวดล้อมที่ใช้ในระบบ

1. เครื่องคอมพิวเตอร์ในห้องปฏิบัติการ 706
2. Software : VM Ware
3. Ubuntu version 14.04

4.4.2 ขั้นตอนการทดลอง

1. ติดตั้ง VM Ware ในเครื่องเซิร์ฟเวอร์ เพื่อติดตั้ง Ubuntu และระบบที่เตรียมไว้
2. ตั้งค่า IP ในเครื่องเซิร์ฟเวอร์เพื่อให้นักศึกษาสามารถเข้าใช้งานได้

3. สร้าง username และ password ให้แต่ละทีมเพื่อใช้ในการเข้าใช้งานระบบ
4. เพิ่มโจทย์ปัญหาเข้าสู่ระบบพร้อมทั้งตั้งเงื่อนไขในการผ่านของโจทย์แต่ละข้อ
5. เริ่มการแข่งขันแต่ละทีมอัป โหลดซอร์สโค้ดเข้าสู่ระบบและทราบถึงผลลัพธ์ที่ได้จากการตรวจสอบ
6. ผู้ดูแลระบบสามารถเข้าดูซอร์สโค้ดของนักศึกษาแต่ละคน และสามารถรู้ถึงผลการรันโปรแกรมได้
7. สรุปคะแนนของแต่ละทีม และประกาศผลการแข่งขัน



รูปที่ 4.23 ภาพบรรยากาศการแข่งขัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

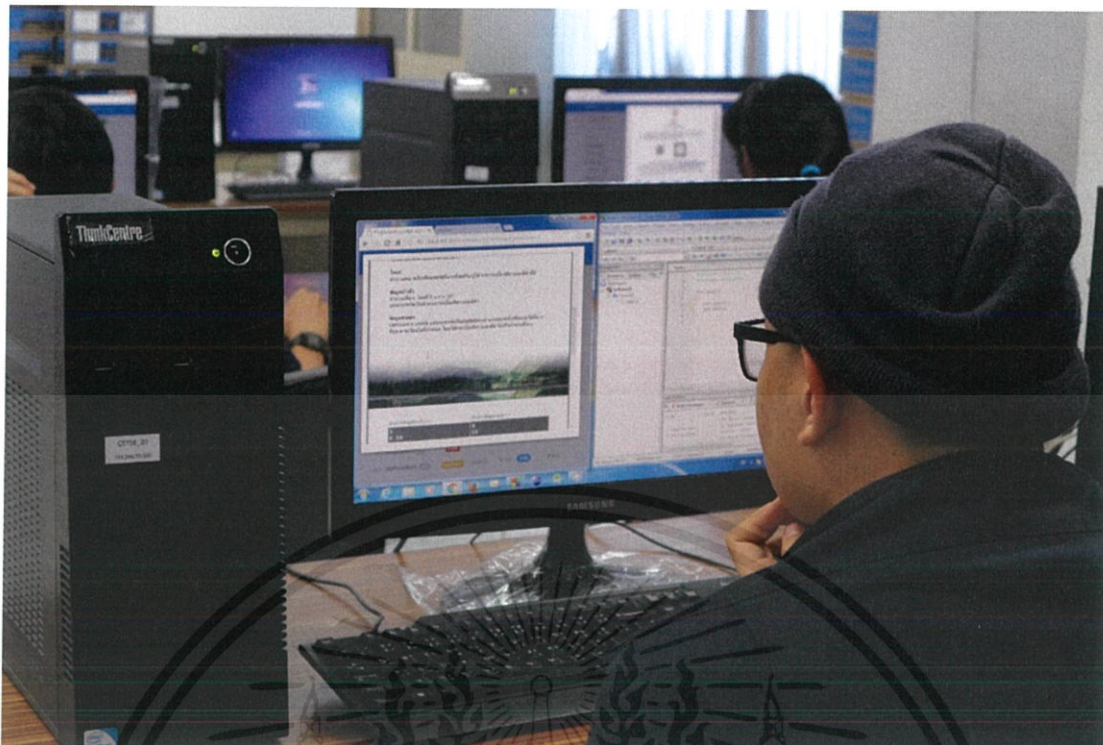


รูปที่ 4.24 ภาพบรรยากาศการแข่งขัน



รูปที่ 4.25 ภาพบรรยากาศการแข่งขัน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 ภาพบรรยากาศการแข่งขัน



รูปที่ 4.27 ภาพบรรยากาศการแข่งขัน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.28 ภาพบรรยากาศการแข่งขัน



รูปที่ 4.29 ทีมผู้ชนะการแข่งขัน(ระดับภาค)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.30 ทีมผู้ชนะการแข่งขัน(ระดับปานกลาง)

4.4.3 สรุปผลการทดลอง

จากผลการทดลองสรุปได้ว่าระบบวิเคราะห์และฝึกฝนการเขียน โปรแกรมภาษาซี สามารถใช้งานได้จริงในสภาพแวดล้อมการใช้งานจริงและทำงานได้ดีตามจุดประสงค์ที่ตั้งไว้ในตอนต้น นอกจากนี้หลังจบการแข่งขันคณะผู้จัดทำได้ให้ผู้เข้าแข่งขันทำแบบสอบถามความพึงพอใจ เพื่อนำผลลัพธ์ที่ได้มาประเมินผลต่อไป โดยแบบสอบถามเป็นดังนี้

Grader Evaluation

ขอบคุณที่ร่วมกิจกรรม G day แบบสอบถามสั้นๆเพื่อวัดความพึงพอใจ

ความง่ายในการใช้งาน

- ง่าย
- ปานกลาง
- ค่อนข้างเข้าใจยาก
- Other

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.31 แบบสอบถามความพึงพอใจ (ส่วนที่ 1)

ปัญหาที่เห็นว่าเกิดขึ้น "ค่อนข้างบ่อย" ในการใช้งาน Grader

- อ่านโจทย์แล้วไม่เข้าใจ ใช้เวลาทำความเข้าใจนาน
- แสดงผลลัพธ์ผิด ทำให้ไม่ผ่านเคส แม้ว่าผิดเพียงเล็กน้อย
- เวลาผลลัพธ์คะแนนไม่ได้เต็ม[P-P] มักไม่รู้ตัวเองไม่ผ่านในกรณีไหนเป็นเวลานาน
- ระบบมักตอบสนองช้า หรือล่มอยู่บ่อยๆ
- มัก Compile error บน Grader
- Other...

รูปที่ 4.32 แบบสอบถามความพึงพอใจ (ส่วนที่ 2)

ถ้าหาก "ไม่มีปัญหา" ดังกล่าว เห็นด้วยกับสิ่งนี้หรือไม่?

***กรณีนำเข้ามาใช้ในห้องแลป Programming

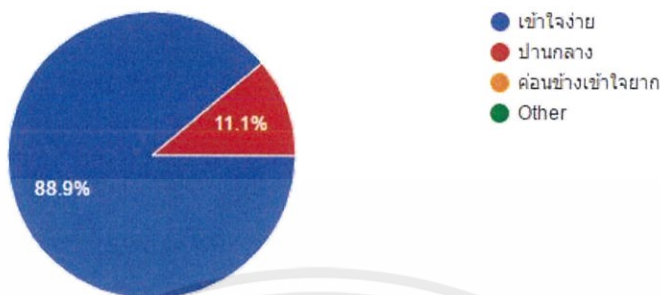
- Grader ให้คำตอบรวดเร็วกว่าการตรวจแบบปกติ
- หากใช้ Grader ในช่วงการสอบ จะลดเวลาในการ[รอ]อาจารย์มาตรวจคำตอบ
- Grader น่าจะเหมาะกับวิชา Computer Programming 1 Lab
- ถ้าเซิร์ฟเวอร์ออนไลน์ตลอดเวลา จะสะดวกในการส่งLab จะได้ส่งเมื่อไรก็ได้

รูปที่ 4.33 แบบสอบถามความพึงพอใจ (ส่วนที่ 3)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลที่ได้จากการตอบแบบสอบถามของผู้เข้าแข่งขันเป็นดังนี้

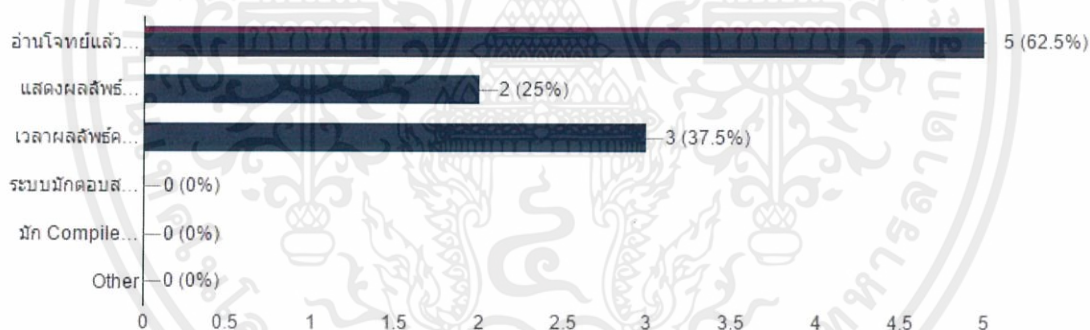
ความง่ายในการใช้งาน (9 responses)



รูปที่ 4.34 ผลการตอบแบบสอบถามความพึงพอใจข้อที่ 1

88.9% ของผู้เข้าแข่งขันเห็นว่าระบบวิเคราะห์และฝึกฝนการเขียน โปรแกรมภาษาซี สามารถเข้าใจและใช้งานได้ง่าย

ปัญหาที่เห็นว่าเกิดขึ้น "ค่อนข้างบ่อย" ในการใช้งาน Grader (8 responses)



รูปที่ 4.35 ผลการตอบแบบสอบถามความพึงพอใจข้อที่ 2

ปัญหาที่พบว่าเกิดขึ้นค่อนข้างบ่อยที่สุดคืออ่าน โจทย์แล้วไม่เข้าใจทำให้ต้องใช้เวลาในการทำความเข้าใจนาน ซึ่งปัญหานี้สามารถแก้ไขได้โดยการเขียนคำอธิบาย โจทย์ให้ชัดเจนและเข้าใจง่ายมากยิ่งขึ้นและปัญหาถัดมาที่พบว่าเกิดขึ้นค่อนข้างบ่อยคือเวลาผลลัพธ์คะแนน ไม่ได้เต็ม[P-P-P] มักไม่รู้ตัวเองไม่ผ่านในกรณีไหนเป็นเวลานาน ซึ่งปัญหานี้เกิดขึ้นเนื่องจากนักศึกษาเขียน โค้ด แก้ปัญหาได้ไม่ครอบคลุมมากพอนักศึกษาต้องหมั่นฝึกฝนพัฒนาลจิกและเขียน โค้ดให้รอบคอบมากยิ่งขึ้น นอกจากนี้จะพบว่าปัญหาอื่น ๆ เช่นระบบตอบสนองช้าหรือ compile error ได้ถูกแก้ไขแล้วเมื่อเปรียบเทียบกับผลการทดลองครั้งที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าหาก"ไม่มีปัญหา"ดังกล่าว เห็นด้วยกับสิ่งนี้หรือไม่? (9 responses)



รูปที่ 4.36 ผลการตอบแบบสอบถามความพึงพอใจข้อที่ 3

จากคำถามข้อที่ 3 พบว่านักศึกษาส่วนใหญ่เห็นว่าหากระบบเป็นแบบออนไลน์จะทำให้สะดวกในการส่งมากยิ่งขึ้น ซึ่งเป็นสิ่งที่ควรพัฒนาต่อไปในอนาคต และอันดับถัดมานักศึกษาเห็นด้วยว่าระบบวิเคราะห์และฝึกฝนการเขียน โปรแกรมภาษาซี(Grader)ให้คำตอบรวดเร็วกว่าการตรวจแบบปกติและน่าจะเหมาะที่จะนำไปประยุกต์ใช้กับภาคปฏิบัติของวิชา Computer Programming 1 ต่อไปในอนาคต

บทที่ 5

สรุป

5.1 สรุป

ระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซีเป็นเว็บแอปพลิเคชันที่แบ่งผู้ใช้งานออกเป็น 2 ประเภท ได้แก่อาจารย์และนักศึกษาซึ่งจะทำงานต่างกัน โดยเมื่อผู้ใช้เข้าสู่ระบบ ระบบจะทำการ authentication และ authorization ก่อนเสมอเพื่อให้สิทธิ์การเข้าใช้งานที่ถูกต้องและเหมาะสมกับผู้ใช้แต่ละคน สำหรับผู้ใช้ที่เป็นอาจารย์จะสามารถสร้างโจทย์ปัญหา แก้ไข และควบคุมการเปิดปิด โจทย์เหล่านั้นได้ รวมถึงเข้าดูประวัติการส่งงานของนักศึกษาแต่ละคนได้ สำหรับผู้ใช้ที่เป็นนักศึกษาก็จะสามารถส่งซอร์สโค้ดเข้าสู่ระบบเพื่อตรวจสอบความถูกต้องของโค้ด และสามารถเข้าดูประวัติการส่ง รวมถึงผลการรันในแต่ละครั้งได้

จากการทดลองในบทที่ 4 พบว่าระบบวิเคราะห์และฝึกฝนการเขียนโปรแกรมภาษาซีสามารถทำงานได้ตามจุดประสงค์ที่ตั้งไว้ สามารถเข้าใช้งานระบบได้จริง ระบบทำงานอย่างมีประสิทธิภาพ และผู้ใช้งานมีความพึงพอใจในระบบ

5.2 ปัญหาและแนวทางการแก้ไข

1. ต้องใช้ source code จำนวนมากในการนำมาทดสอบระบบเพื่อพิสูจน์ว่าระบบสามารถทำงานได้อย่างถูกต้องและน่าเชื่อถือ ผู้จัดทำจึงได้นำระบบเข้าไปทดลองใช้ในวิชา Data structure & algorithm ภาคเรียนที่ 2 ปีการศึกษา 2558 และวิชา Computer Programming 1 ภาคเรียนที่ 2 ปีการศึกษา 2559 รวมถึงจัดการแข่งขันการเขียนโปรแกรมภาษาซีเพื่อทดสอบระบบในสถานการณ์จริงและทำให้มี source code ที่ใช้ทดสอบหลากหลายมากขึ้น

2. การทดสอบระบบเพื่อพิสูจน์ว่าระบบสามารถรองรับการทำงานหนักๆได้ เช่น ไฟล์นำเข้ามีขนาด 20 Mb การวนลูปหลักพันล้านครั้ง จำเป็นต้องออกแบบโจทย์ที่ให้ผู้ทดลองใช้งานต้องใช้ทักษะในการเขียนโปรแกรมมากในระดับหนึ่ง และมีความเข้าใจในการเขียนโค้ดเป็นอย่างดี ช่วยในการตรวจเพื่อยืนยันความถูกต้องของระบบ ผู้จัดทำจึงได้ขอความร่วมมือจากพี่ๆและเพื่อนๆในการช่วยกันตรวจสอบ source code เหล่านั้น

3. การคอมไพล์ตัวโปรแกรมนั้น จะต้องใช้คำสั่งบนหน้าจอกอนโซลในการรันคอมไพล์เลอร์ โดยที่ภาษาซีจะใช้คำสั่ง gcc ในการประมวล ซึ่งจะต้องมีลำดับของพารามิเตอร์ที่แน่นอน

4. ในกรณีนักศึกษาส่งโค้ดมาเพื่อให้ระบบประมวลผล จะเกิดปัญหาขึ้นเมื่อชื่อไฟล์มีอักขรพิเศษ เช่น ' '(space) ประกอบอยู่เพราะชื่อของไฟล์จะไม่สมบูรณ์ เช่น ชื่อไฟล์ที่ใช้ในการส่งเป็น `ab Problem.cpp` เวลานำไปประมวลผลจะได้เป็น gcc ab เท่านั้น ส่งผลการคอมไพล์ผิดพลาด ดังนั้นผู้จัดทำจึงแก้ปัญหาโดยการนำไฟล์ที่ส่งเข้ามาในระบบไปแปลงเป็นชื่อที่เหมาะสม

5. ปัญหาจากความไม่เชี่ยวชาญในการออกแบบหน้าเว็บทำให้ต้องเปลี่ยนลักษณะของหน้าเว็บหลายครั้ง เพื่อให้ผู้ใช้งานเข้าใจง่าย เช่นการเลือกโจทย์ปัญหาและเลือกไฟล์ที่จะในการส่ง

แบบที่ 1

Name	Level	Result
1 knapsack2 <small>pdf</small>	medium	score: 100
2 Problem Sum 2	easy	compile: error

รูปที่ 5.1 Interface แบบที่ 1

แบบดั้งเดิมในการส่งโค้ดทั่วไป โดยเลือกโจทย์ที่ต้องการจะส่งจาก Dropdown component จากนั้นเลือกอัปโหลดไฟล์ซอร์สโค้ด (file selection)

ข้อดี การออกแบบเรียบง่าย ใช้คอมพิวเตอร์น้อย

ข้อเสีย ใช้การคลิกถึง 5 ครั้งต่อ 1 การส่ง

แบบที่ 2

Name	Level	Result
1 knapsack2 pdf CHOOSE FILE SEND medium score: 100	medium	score: 100
2 Problem Sum 2 CHOOSE FILE SEND easy compile: error	easy	compile: error
3 temp pdf CHOOSE FILE SEND easy score: 0	easy	score: 0
4 A+B Problem pdf CHOOSE FILE SEND easy score: 0	easy	score: 0
5 Heap sort pdf CHOOSE FILE SEND hard -	hard	-
6 knapsack pdf CHOOSE FILE SEND medium score: 0	medium	score: 0
7 Tree sideway pdf CHOOSE FILE SEND medium	medium	

รูปที่ 5.2 Interface แบบที่ 2

สามารถคลิก CHOOSE FILE ได้โดยตรง และเลือกอัปโหลดโค้ดตามลำดับ(file selection)

ข้อดี ใช้การคลิกน้อยกว่าเดิม

ข้อเสีย จำนวนคอมโพเนนต์ของเว็บมากขึ้นตามจำนวนโจทย์

แบบที่ 3

Name	Level	Result
1 knapsack2 <small>pdf</small> CHOOSE FILE & SEND medium score: 100		
2 Problem Sum 2 CHOOSE FILE & SEND easy compile: error		
3 temp <small>pdf</small> CHOOSE FILE & SEND easy score: 0		
4 A+B Problem <small>pdf</small> CHOOSE FILE & SEND easy score: 0		
5 Heap sort <small>pdf</small> CHOOSE FILE & SEND hard -		
6 knapsack <small>pdf</small> CHOOSE FILE & SEND medium score: 0		
7 Tree sideway <small>pdf</small> CHOOSE FILE & SEND medium -		

รูปที่ 5.3 Interface แบบที่ 3

สามารถเลือกไฟล์โค้ด (file selection) และส่งทันที

ข้อดี ใช้การคลิกเพียง 2 ครั้งต่อ 1 ครั้งการส่ง(submission)

ข้อเสีย กรณีเลือกไฟล์ผิดพลาด อาจส่งผลกระทบต่อระบบโดยตรง

แบบที่เลือกใช้ในงานจริงในปัจจุบัน

SumEven CHOOSE FILE... SUBMIT

Name	Level	Detect
1 SumEven	medium	-
2 Grade pdf	easy	main()

รูปที่ 5.4 Interface ที่เลือกใช้งานจริง

เป็นลักษณะแบบเดียวกับแบบที่ 1 แต่มีการจำ요ข้อที่เลือกส่งล่าสุด ซึ่งส่งผลคืนในกรณีส่งซ้ำหลายๆครั้ง จะใช้การคลิกเพียง 3 ครั้ง

6. Hard disk เกิดความเสียหายทำให้ต้องใช้เวลาในการกู้คืนข้อมูล ทำให้งานล่าช้ากว่าที่คาดการณ์

5.3 แนวทางการพัฒนาต่อ

1. พัฒนาระบบให้เป็นแบบออนไลน์
2. พัฒนา User Interface ให้สวยงามและใช้งานง่ายมากยิ่งขึ้น
3. พัฒนาให้รองรับภาษาอื่นๆ มากขึ้น
4. เชื่อมต่อกับระบบ Moss ของ Stanford CS สำหรับการตรวจโค้ดที่มีความใกล้เคียงกัน
ปัจจุบัน ได้ขอ username และ password สำหรับการใช้งาน application program interface(api)

บรรณานุกรม

ณิชมน วศินสกุล และดวงชนก สุวรรณเวช. 2557. ระบบจัดการห้องปฏิบัติการโปรแกรมมิ่ง. กรุงเทพฯ :

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

Chris Rebert. 2016. **Bootstrap 3.3.7**. [online].

Available : <https://blog.getbootstrap.com/2016/07/25/bootstrap-3-3-7-released/>

Johnson and Vlissides. 2012. **Model-View-Controller**. [Online].

Available : <http://msdn.microsoft.com/en-us/library/ff649643.aspx>

Justin Ellingwood. 2014. **How To Add Swap on Ubuntu 14.04**. [online].

Available : <https://www.digitalocean.com/community/tutorials/how-to-add-swap-on-ubuntu-14-04>

Martin Mareš. 2015. **Library** สำหรับการป้องกันเซิร์ฟเวอร์ต่อระหว่างโปรแกรมทำงาน. [online].

Available : <http://www.ucw.cz/moe/>

Narzt, Camden. 2016. **Passenger: Stability and speed improvements**. [online].

Available : <https://blog.phusion.nl/2016/07/25/passenger-5-0-30/>

Ohm. 2016. **ส่วนประกอบของ Web Application (Website)**. [Online].

Available : <https://blog.wisered.com/ส่วนประกอบของ-web-application-website/>

Paul. 2015. **Bootswatch-Sass**. [online].

Available : <https://github.com/log0ymxm/bootswatch-scss>

Phusion. 2016. **Passenger : Web application on server**. [online].

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

Richard Matthew Stallman. 2011. **GNU cflow analyzes a collection of C source files and prints a graph, charting control flow within the program.** [online]

Available : <https://www.gnu.org/software/cflow/>



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้