รายงานวิจัยฉบับสมบูรณ์
โครงการวิจัยการพัฒนามอดูลการวัดความซับซ้อนของซอร์สโค้ด
สำหรับแพลตฟอร์มการเรียนรู้การเขียนโปรแกรม
(The development of a module for measuring code complexity
for learning programming platform)

นายพีระศักดิ์ เพียรประสิทธิ์ **หัวหน้าโครงการ**

นางสาวณัฐพร ภักดี **ผู้ร่วมวิจัย**

โครงการวิจัยประเภทเงินรายได้

คณะวิทยาการสารสนเทศ

มหาวิทยาลัยบูรพา

ประจำปีงบประมาณ พ.ศ. 2567

เลขที่สัญญา 07/2567

รายงานวิจัยฉบับสมบูรณ์

โครงการวิจัยการพัฒนามอดูลการวัดความซับซ้อนของซอร์สโค้ด สำหรับแพลตฟอร์มการเรียนรู้การเขียนโปรแกรม

(The development of a module for measuring code complexity for learning programming platform)

นายพีระศักดิ์ เพียรประสิทธิ์ หัวหน้าโครงการ

นางสาวณัฐพร ภักดี ผู้ร่วมวิจัย

คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

มกราคม 2567

กิติกรรมประกาศ

งานวิจัยนี้ได้รับทุนสนับสนุนการวิจัยจากงบประมาณโครงการวิจัยประเภทเงินรายได้ เพื่อส่งเสริมและพัฒนาศักยภาพผู้ช่วยวิจัย คณะวิทยาการสารสนเทศ ประจำปีงบประมาณ พ.ศ. 2567 เลขที่สัญญา 7/2567

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนามอดูลการวัดผลความซับซ้อนของซอร์สโค้ดสำหรับแพลตฟอร์มการเรียนรู้การเขียนโปรแกรม ซึ่งเป็นแพลตฟอร์มสำหรับการเรียนการสอนที่เกี่ยวข้องกับการเขียนโปรแกรม ปัจจุบันแพลตฟอร์มนี้มีฟังก์ชันต่างๆ เช่น การกำหนดปัญหา การมอบหมายงานให้นักเรียน และการตรวจประเมินความถูกต้องของซอร์สโค้ดและรายงานผลคะแนนของผู้เรียน วัตถุประสงค์ของแพลตฟอร์มนี้คือเพื่อทำให้การสอนและการเรียนรู้เข้าถึงได้ง่ายขึ้นสำหรับผู้สอนและนักเรียน มอดูลการวัดความซับซ้อนของซอร์สโค้ดที่พัฒนาขึ้นนี้จะใช้วัดค่าความซับซ้อนของซอร์สโค้ดของผู้เรียนในแต่ละโจทย์ปัญหา โดยใช้การวัดด้วย Cyclomatic metrics และ Object Oriented Design Metrics (C-K metrics) ผลลัพธ์จากรายงานความซับซ้อนของซอร์สโค้ดของผู้เรียนแต่ละโจทย์ปัญหาจะสะท้อนถึงระดับความยากง่ายของโจทย์ปัญหา และใช้เป็นข้อมูลสนับสนุนผู้สอนในการตัดสินใจเลือกโจทย์ปัญหาสำหรับการเขียนโปรแกรม โดยคำนึงถึงระดับความยากของคำถามที่เหมาะสมกับนักเรียน และสร้างผลลัพธ์การเรียนรู้ให้แก่ผู้เรียนให้เหมาะสม ข้อมูลนี้อาจถูกนำไปใช้ในการกำหนดงาน ประเมินระดับทักษะความรู้ความสามารถของนักเรียน และปรับกิจกรรมการเรียนรู้ให้สอดคล้องกับทักษะของผู้เรียน

Abstract

This research aims to present a measurement module that is an ongoing development for the Learning Programming Platform, which is a platform for teaching and learning programming-related subjects. Currently, the platform includes functions such as defining problems, assigning students assignments, and assessing source code and reports. The objective of this platform was to make teaching and learning more accessible for instructors and students. The measurement module will measure the complexity of the student source code for each problem. We use cyclomatic metrics and object oriented design metrics (C-K metrics) to measure student source codes. The question complexity reports support the instructor in deciding the coding questions, considering the difficulty level of coding questions well suited to students, and building the learning outcome of that course. This information may be utilized to establish assignments, adjust activities to be relevant skills for the class's learners, and assess the level of skills of students.

สารบัญ

หน้า
กิติกรรมประกาศ......ก
บทคัดย่อ......ข
Abstract......ค
สารบัญ......ง
สารบัญภาพ......ฉ
สารบัญตาราง......ฌ
บทที่ 1 บทนำ......1
1.1 ที่มาและความสำคัญของงานวิจัย......1
1.2 วัตถุประสงค์ของงานวิจัย......2
1.3 ขอบเขตของงานวิจัย......2
1.4 ประโยชน์ที่คาดว่าจะได้รับ......2
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง......4
2.1 มาตรวัดซอฟต์แวร์......6
2.2 การเรียนรู้แบบปรับได้ (Adaptive Learning)......8
2.3 การเรียนรู้ของเครื่อง (Machine Learning)......9
2.4 อัลกอริทึม Random Forest......10
2.5 งานวิจัยที่เกี่ยวข้อง......11
2.6 เทคโนโลยีและเฟรมเวิร์กที่เกี่ยวข้อง......14
2.7 เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์......17
บทที่ 3 รายละเอียดการดำเนินงานวิจัย......20
3.1 การออกแบบสถาปัตยกรรมซอฟต์แวร์......20
3.2 แผนภาพยูสเคส (Use Case Diagram)......23
3.3 คำอธิบายยูสเคส (Use Case Description)......24

สารบัญ (ต่อ)

หน้า

3.4 แผนภาพกิจกรรม (Activity Diagram)....44

3.5 แผนภาพความสัมพันธ์ของข้อมูล (Entity Relationship Diagram)....59

3.6 การออกแบบโมเดลสำหรับแนะนำโจทย์ปัญหาที่เหมาะสมกับผู้เรียน....61

บทที่ 4 ผลการดำเนินงาน....65

4.1 มอดูลประมวลผลความซับซ้อนของซอร์สโค้ด....65

4.2 มอดูลการเรียนรู้แบบปรับได้....79

4.3 การประเมินผลประสิทธิภาพของโมเดล Random Forest....87

บทที่ 5 สรุปและอภิปรายผล....90

5.1 สรุปผลการดำเนินงาน....90

5.2 ประโยชน์ที่ได้รับ....91

5.3 ข้อจำกัด....91

5.4 ข้อเสนอแนะในอนาคต....91

ภาคผนวก....92

ภาคผนวก ก บทความวิจัยที่ตีพิมพ์เผยแพร่....93

ภาคผนวก ข พจนานุกรมข้อมูล (Data Dictionary)....98

บรรณานุกรม....121

รายงานสรุปการเงิน....126

ประวัติคณะผู้วิจัย....128

สารบัญภาพ

หน้า

ภาพที่ 2-1 สัญลักษณ์ของ Lizard 14
ภาพที่ 2-2 สัญลักษณ์ของ Node.js 18
ภาพที่ 2-3 สัญลักษณ์ของ Vue.js 18
ภาพที่ 2-4 สัญลักษณ์ของ Nuxt.js 18
ภาพที่ 2-5 สัญลักษณ์ของ Express.js 19
ภาพที่ 2-6 สัญลักษณ์ของ Sequelize 19
ภาพที่ 2-7 สัญลักษณ์ของ MariaDB 19
ภาพที่ 3-1 สถาปัตยกรรมของซอฟต์แวร์ 21
ภาพที่ 3-2 แผนภาพยูสเคส 23
ภาพที่ 3-3 แผนภาพกิจกรรมเพิ่มคิว 44
ภาพที่ 3-4 แผนภาพกิจกรรมลบคิว 46
ภาพที่ 3-5 แผนภาพกิจกรรมประมวลผลความซับซ้อนของซอร์สโค้ด 47
ภาพที่ 3-6 แผนภาพกิจกรรมแสดงประวัติการประมวลผล 48
ภาพที่ 3-7 แผนภาพกิจกรรมเพิ่มโจทย์ปัญหา 49
ภาพที่ 3-8 แผนภาพกิจกรรมแสดงผลรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด 50
ภาพที่ 3-9 แผนภาพกิจกรรมรายงานค่าความซับซ้อนของโจทย์ปัญหา 51
ภาพที่ 3-10 แผนภาพกิจกรรมแสดงผลรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน 52
ภาพที่ 3-11 แผนภาพกิจกรรมรายงานค่าความซับซ้อนของโจทย์ปัญหาด้วยเมตริกเชิงวัตถุ 53
ภาพที่ 3-12 แผนภาพกิจกรรมการแนะนำการเรียนรู้รายบุคคล 55
ภาพที่ 3-13 แผนภาพกิจกรรมการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้ 56
ภาพที่ 3-14 แผนภาพกิจกรรมการเปลี่ยนโจทย์ปัญหาการเรียนรู้แบบปรับได้ 57
ภาพที่ 3-15 แผนภาพกิจกรรมการส่งคำตอบโจทย์ปัญหาการเรียนรู้แบบปรับได้ 58
ภาพที่ 3-16 แผนภาพกิจกรรมการแสดงหน้าจอประวัติการเรียนรู้แบบปรับได้ 59
ภาพที่ 3-17 แผนภาพความสัมพันธ์ข้อมูลส่วนของการวัดความซับซ้อนของซอร์สโค้ด 60
ภาพที่ 3-18 แผนภาพความสัมพันธ์ข้อมูลส่วนของการเรียนรู้แบบปรับได้ 60

สารบัญภาพ (ต่อ)

หน้า

ภาพที่ 3-19 ขั้นตอนการสร้างโมเดลสำหรับการแนะนำการเรียนรู้แบบปรับได้................................62
ภาพที่ 4-1 หน้าจอจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด.............................65
ภาพที่ 4-2 หน้าจอกรอกข้อมูลจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด...........66
ภาพที่ 4-3 หน้าจอแสดงข้อมูลจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด 66
ภาพที่ 4-4 หน้าจอเพิ่มคิว..67
ภาพที่ 4-5 หน้าจอการกรอกข้อมูลเพิ่มคิว..67
ภาพที่ 4-6 หน้าจอจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ดหลังการเพิ่มข้อมูล 68
ภาพที่ 4-7 หน้าจอการกรอกข้อมูลเพิ่มคิวไม่ครบถ้วน...68
ภาพที่ 4-8 หน้าจอลบคิว 69
ภาพที่ 4-9 หน้าจอยืนยันการลบคิว 69
ภาพที่ 4-10 หน้าจอแสดงตารางประวัติการประมวลผลความซับซ้อนของซอร์สโค้ด 70
ภาพที่ 4-11 หน้าจอกรอกข้อมูลรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด...........................72
ภาพที่ 4-12 หน้าจอแสดงกราฟรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด 72
ภาพที่ 4-13 หน้าจอแสดงตารางรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด 73
ภาพที่ 4-14 หน้าจอกรอกข้อมูลรายงานค่าความซับซ้อนของโจทย์ปัญหา......................................73
ภาพที่ 4-15 หน้าจอแสดงกราฟรายงานค่าความซับซ้อนของโจทย์ปัญหา 74
ภาพที่ 4-16 หน้าจอแสดงตารางรายงานค่าความซับซ้อนของโจทย์ปัญหา 74
ภาพที่ 4-17 หน้าจอกรอกข้อมูลรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน 75
ภาพที่ 4-18 หน้าจอแสดงกราฟรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน....................75
ภาพที่ 4-19 หน้าจอแสดงตารางรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน..................76
ภาพที่ 4-20 หน้าจอกรอกข้อมูลรายงานค่าความซับซ้อนด้วยเมตริกเชิงวัตถุ..................................76
ภาพที่ 4-21 หน้าจอแสดงรายงานค่าความซับซ้อนด้วยเมตริกเชิงวัตถุ...77
ภาพที่ 4-22 หน้าจอแสดงรายงานค่าความซับซ้อนของโจทย์ปัญหารายบุคคล 77
ภาพที่ 4-23 หน้าจอกรอกข้อมูลรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละข้อ..........................78
ภาพที่ 4-24 หน้าจอแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละข้อ........................79

สารบัญภาพ (ต่อ)

หน้า

ภาพที่ 4-25 หน้าจอการเลือกเมนูแนะนำการเรียนรู้รายบุคคล 80
ภาพที่ 4-26 หน้าจอแนะนำการเรียนรู้รายบุคคลส่วนของการแสดงโจทย์ปัญหา 80
ภาพที่ 4-27 หน้าจอแสดงแนะนำโจทย์ปัญหาถัดไปที่เหมาะสม..81
ภาพที่ 4-28 หน้าจอแสดงโจทย์ปัญหาที่ยังไม่เสร็จ...81
ภาพที่ 4-29 หน้าจอการทำโจทย์ประเมินตามความรู้ของนักศึกษา..82
ภาพที่ 4-30 หน้าจอการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้ส่วนของการเลือกภาษา....................82
ภาพที่ 4-31 หน้าจอการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้...83
ภาพที่ 4-32 หน้าจอการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้สำเร็จ 83
ภาพที่ 4-33 หน้าจอการคลิกปุ่มไปยังโจทย์ถัดไป...84
ภาพที่ 4-34 หน้าจอคลิกปุ่ม “เปลี่ยนโจทย์ปัญหา” 84
ภาพที่ 4-35 หน้าจอแสดงกล่องโต้ตอบยืนยันการเปลี่ยนโจทย์ปัญหา..84
ภาพที่ 4-36 หน้าจอแสดงโจทย์ปัญหาใหม่...85
ภาพที่ 4-37 หน้าจอการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้สำเร็จ 85
ภาพที่ 4-38 หน้าจอแสดงข้อผิดพลาดเมื่อคอมไพล์ไม่ผ่าน...86
ภาพที่ 4-39 หน้าจอแสดงผลการตรวจสอบตามกรณีทดสอบ 86
ภาพที่ 4-40 หน้าจอประวัติการเรียนรู้รายบุคคล 87

สารบัญตาราง

หน้า

ตารางที่ 2-1 คำศัพท์เฉพาะ 4
ตารางที่ 2-2 รายละเอียด Options Lizard 15
ตารางที่ 2-3 รายละเอียด Options Chidamber and Kemerer Java Metrics 17
ตารางที่ 3-1 คำอธิบายยูสเคสเพิ่มคิว 24
ตารางที่ 3-2 คำอธิบายยูสเคสลบคิว 25
ตารางที่ 3-3 คำอธิบายยูสเคสประมวลผลความซับซ้อนของซอร์สโค้ด 27
ตารางที่ 3-4 คำอธิบายยูสเคสแสดงประวัติการประมวลผล 28
ตารางที่ 3-5 คำอธิบายยูสเคสเพิ่มโจทย์ปัญหา 29
ตารางที่ 3-6 คำอธิบายยูสเคสรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด 31
ตารางที่ 3-7 คำอธิบายยูสเคสรายงานค่าความซับซ้อนของโจทย์ปัญหา 33
ตารางที่ 3-8 คำอธิบายยูสเคสรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน 35
ตารางที่ 3-9 คำอธิบายยูสเคสรายงานค่าความซับซ้อนด้วยเมตริกเชิงวัตถุ 37
ตารางที่ 3-10 คำอธิบายยูสเคสแนะนำโจทย์ปัญหาแบบปรับได้ 39
ตารางที่ 3-11 คำอธิบายยูสเคสการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้ 40
ตารางที่ 3-12 คำอธิบายยูสเคสการเปลี่ยนโจทย์ปัญหาการเรียนรู้แบบปรับได้ 41
ตารางที่ 3-13 คำอธิบายยูสเคสการส่งคำตอบโจทย์ปัญหาการเรียนรู้แบบปรับได้ 42
ตารางที่ 3-14 คำอธิบายยูสเคสการส่งคำตอบโจทย์ปัญหาการเรียนรู้แบบปรับได้ 43
ตารางที่ 4-1 รายงานผลการจำแนก (Classification Report) 88
ตารางที่ 4-2 Confusion Matrix ของโมเดล 88
ตารางที่ ข-1 รายละเอียดตาราง headerReport 98
ตารางที่ ข-2 รายละเอียดตาราง detailReportLizard 101
ตารางที่ ข-3 รายละเอียดตาราง detailReportOOP 102
ตารางที่ ข-4 รายละเอียดตาราง queue 104
ตารางที่ ข-5 รายละเอียดตาราง problems 106
ตารางที่ ข-6 รายละเอียดตาราง assignments 107

สารบัญตาราง (ต่อ)

หน้า

ตารางที่ ข-7 รายละเอียดตาราง sections.....109
ตารางที่ ข-8 รายละเอียดตาราง users.....110
ตารางที่ ข-9 รายละเอียดตาราง courses.....111
ตารางที่ ข-10 รายละเอียดตาราง problems.....112
ตารางที่ ข-11 รายละเอียดตาราง problemCodeFiles.....114
ตารางที่ ข-12 รายละเอียดตาราง solutions.....115
ตารางที่ ข-13 รายละเอียดตาราง adaptive.....116
ตารางที่ ข-14 รายละเอียดตาราง adaptiveUsersLogs.....117
ตารางที่ ข-15 รายละเอียดตาราง adaptiveCompileLogs.....119

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของงานวิจัย

คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา ได้พัฒนาเครื่องมือสนับสนุนการเรียนวิชาการเขียนโปรแกรม (Phakdee & Pianprasit, 2023) ซึ่งใช้เป็นเครื่องมือสำหรับให้อาจารย์ผู้สอนในรายวิชาการเขียนโปรแกรมให้สามารถจัดการรายวิชา กลุ่มเรียน เพิ่มนักศึกษาเข้ากลุ่มเรียนหรือรายวิชา กำหนดโจทย์และมอบหมายงานให้กับนักศึกษาของรายวิชาการเขียนโปรแกรม โดยนักศึกษาสามารถส่งไฟล์ซอร์สโค้ด (Source code) หรือเขียนโปรแกรมผ่านระบบเพื่อให้ระบบตรวจสอบการเขียนโปรแกรมตามชุดข้อมูลทดสอบ (Test set) ที่อาจารย์ผู้สอนกำหนดไว้ และแสดงผลการตรวจสอบผ่านหน้าจอของระบบ อย่างไรก็ตาม ปัจจุบันเครื่องมือสนับสนุนการเรียนวิชาการเขียนโปรแกรมที่ใช้งานนั้นยังไม่มีความสามารถในการวัดความซับซ้อนของโจทย์ปัญหา ทำให้อาจารย์ผู้สอนไม่มีข้อมูลที่ช่วยในการพิจารณาระดับความยากของโจทย์ปัญหา ส่งผลให้การปรับปรุงโจทย์ปัญหาให้มีระดับความยากที่เหมาะสมกับนักศึกษานั้นเป็นไปอย่างยากลำบาก เนื่องจากการประเมินความยากของโจทย์ปัญหานั้นเป็นการประเมินจากอาจารย์ผู้สอนเท่านั้น แต่ในขณะที่ผู้เรียนทำเขียนซอร์สโค้ดสำหรับแก้โจทย์ปัญหานั้นอาจใช้วิธีที่แตกต่างกันและอาจแตกต่างจากวิธีการที่อาจารย์ผู้สอนประเมินไว้นั้น ทำให้ผู้สอนและผู้เรียนพิจารณาว่าโจทย์ปัญหานั้นยากง่ายแตกต่างกัน ดังนั้น ผู้วิจัยจึงได้มีแนวคิดที่จะพัฒนาต่อยอดระบบเดิมให้เพิ่มขีดความสามารถในการวัดความซับซ้อนของซอร์สโค้ดของโจทย์ปัญหาเพิ่มเติม เพื่อใช้เป็นเครื่องมือวัดที่มีมาตรฐานและเป็นเครื่องมือช่วยให้อาจารย์สามารถปรับปรุงโจทย์ปัญหาหรือมอบหมายงานที่มีระดับความยากที่เหมาะสมกับนักศึกษาในชั้นเรียน และสามารถตรวจสอบคุณภาพของซอร์สโค้ดของนักศึกษา รวมถึงสะท้อนความสามารถของผู้เรียนในการเขียนโปรแกรมในการแก้ไขโจทย์ปัญหา โดยมาตรวัดที่ใช้วัดคุณภาพของซอร์สโค้ดจะใช้มาตรวัด Cyclomatic (“Cyclomatic Complexity,” 2018; “Cyclomatic Complexity,” 2024; T. J. McCabe, 1976) และมาตรวัดซอฟต์แวร์เชิงวัตถุ (Chidamber & Kemerer, 1994) ซึ่งผลลัพธ์จากมาตรวัดเหล่านี้จะถูกจัดเก็บในฐานข้อมูลและผู้ใช้งานสามารถเรียกดูรายงานที่แสดงผลการวัดค่าความซับซ้อนของโจทย์ปัญหาได้ ได้แก่ รายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด รายงานค่าความซับซ้อนของโจทย์ปัญหา รายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน และรายงานค่าความซับซ้อนด้วยเมตริก Object-Oriented

1.2 วัตถุประสงค์ของงานวิจัย

1) เพื่อพัฒนาต่อยอดระบบสารสนเทศเดิมให้มีความสามารถในการวัดประเมินความซับซ้อนของซอร์สโค้ด

2) เพื่อจัดเก็บข้อมูลผลความซับซ้อนของซอร์สโค้ดของผู้เรียนในการทำโจทย์ปัญหาแต่ละโจทย์ปัญหา

3) เพื่อใช้เป็นแนวทางการปรับปรุงการเรียนการสอนวิชาการเขียนโปรแกรม ในส่วนของการมอบหมายโจทย์ปัญหาให้มีระดับง่ายไปยังระดับยากมีความชัดเจนมากขึ้น

4) เพื่อพัฒนาต้นแบบการสร้างเครื่องมือประเมินความรู้ผู้เรียนแบบปรับได้ (Adaptive)

1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้เป็นการพัฒนาต่อยอดระบบสารสนเทศเดิม ซึ่งขอบเขตงานหลักจะมีการพัฒนาส่วนที่ในการวัดความซับซ้อนของซอร์สโค้ดโดยใช้มาตรวัด Cyclomatic และมาตรวัดซอฟต์แวร์เชิงวัตถุในการวัดค่าความซับซ้อนของซอร์สโค้ด โดยมีฟังก์ชันต่างๆ ดังนี้

1) ฟังก์ชันวัดค่าความซับซ้อนของซอร์สโค้ด คือการนำไฟล์ซอร์สโค้ดที่ต้องการวัดค่าความซับซ้อน โดยใช้มาตรวัด Cyclomatic Complexity และมาตรวัดซอฟต์แวร์เชิงวัตถุ (Object-Oriented) ในการวัดค่าความซับซ้อนของซอร์สโค้ดและบันทึกข้อมูลค่าความซับซ้อนของซอร์สโค้ดลงฐานข้อมูล

2) ฟังก์ชันแสดงผลรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด เป็นฟังก์ชันสำหรับการแสดงผลรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมดในแต่ละรายวิชาที่ได้จากการวัดค่าความซับซ้อนของซอร์สโค้ดโดยใช้มาตรวัด Cyclomatic

3) ฟังก์ชันแสดงผลรายงานค่าความซับซ้อนของโจทย์ปัญหา เป็นฟังก์ชันสำหรับการแสดงผลรายงานค่าความซับซ้อนของโจทย์ปัญหาของนักศึกษาทั้งหมดที่ได้จากการวัดค่าความซับซ้อนของซอร์สโค้ดโดยใช้มาตรวัด Cyclomatic

4) ฟังก์ชันแสดงผลรายงานค่าความซับซ้อนด้วยเมตริกเชิงวัตถุ (Object Oriented) เป็นฟังก์ชันสำหรับการแสดงผลรายงานค่าความซับซ้อนด้วยเมตริกเชิงวัตถุที่ได้จากการวัดค่าความซับซ้อนของซอร์สโค้ดที่นักศึกษาแต่ละคน โดยใช้มาตรวัดซอฟต์แวร์เชิงวัตถุ (Object Oriented)

5) ฟังก์ชันสร้างชุดโจทย์ปัญหาแบบ Adaptive สำหรับประเมินความรู้ของผู้เรียน

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) แพลตฟอร์มการเรียนรู้การเขียนโปรแกรมได้รับการพัฒนาให้มีขีดความสามารถเพิ่มขึ้นโดยมีฟีเจอร์การวัดความซับซ้อนของซอร์สโค้ดของผู้เรียน

2) มอดูลวัดความซับซ้อนของซอร์สโค้ดรองรับรูปแบบการเขียนโปรแกรม (Programming paradigms) ทั้งการเขียนโปรแกรมเชิงโครงสร้าง (Structured programming) และการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming: OOP)

3) รายงานค่าความซับซ้อนของซอร์สโค้ดต่างๆ ช่วยให้อาจารย์ผู้สอนนำไปใช้ในการปรับกิจกรรมการสอน/การมอบหมายงานต่างๆ ให้แก่ผู้เรียนให้มีความเหมาะสมยิ่งขึ้น

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

การพัฒนามอดูลการวัดความซับซ้อนของซอร์สโค้ดสำหรับแพลตฟอร์มการเรียนรู้การเขียนโปรแกรม ผู้วิจัยได้ศึกษางานวิจัย เทคโนโลยี ภาษา เฟรมเวิร์ก และซอฟต์แวร์ต่างๆ สำหรับการพัฒนาแพลตฟอร์มการเรียนรู้การเขียนโปรแกรม โดยบทนี้ ผู้วิจัยจะอธิบายคำศัพท์ที่เกี่ยวข้องเพื่อให้ผู้อ่านมีความเข้าใจตรงกัน หลังจากนั้นงานวิจัยที่เกี่ยวข้องและเทคโนโลยี ภาษา เฟรมเวิร์กที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ โดยมีรายละเอียดดังต่อไปนี้

ตารางที่ 2-1 คำศัพท์เฉพาะ

ลำดับ	คำศัพท์	ความหมาย
1	ซอร์สโค้ด (Source code)	แฟ้มข้อมูลที่เป็นตัวต้นฉบับของโปรแกรม ซึ่งภายในประกอบไปด้วยภาษาโปรแกรม ที่เขียนขึ้นสำหรับสั่งการคอมพิวเตอร์ให้ทำงานตามที่ผู้เขียนต้องการ
2	Non-Commenting Source Statements (NCSS)	ซอร์สโค้ด (Source code) ที่ไม่รวมคำอธิบายซอร์สโค้ด (Comment)
3	Cyclomatic Complexity	มาตรวัด Cyclomatic เสนอโดย Thomas McCabe ในปี 1976 ทำหน้าที่เป็นตัวชี้วัดซอฟต์แวร์เพื่อประเมินความซับซ้อนของโปรแกรม โดยวัดจำนวนเส้นทางอิสระเชิงเส้นภายในโค้ดต้นฉบับ
4	Cyclomatic Complexity Number (CCN)	ค่าความซับซ้อนของการใช้คำสั่งควบคุมในโปรแกรมที่ใช้คำสั่ง If, While, Repeat และ For
5	การเกาะกันเป็นก้อน (Cohesion)	การบอกถึงความสอดคล้องกันของการทำงานในหน่วยใดๆ เช่น ถ้าหากมองในมุมของ Coding ในระดับ Class ความสอดคล้องกันของ Class นั้น หมายถึง Method การทำงานต่างๆ ที่ควรจะเป็นไปในทางเดียวกัน
6	การเข้าคู่ (Coupling)	ระดับความเกี่ยวข้อง หรือ พึ่งพากันของ Module ต่างๆ ในระบบ ซึ่งระบบที่ดีควรออกแบบให้ Module ต่างๆ เป็นอิสระจากกันมากที่สุด ถ้าหากมองในมุม Coding คือการทำให้ Object ต่างๆ มีการอ้างอิงกันน้อยที่สุด เท่าที่จะเป็นไปได้

ลำดับ	คำศัพท์	ความหมาย
7	Object Oriented Design Metrics	มาตรวัดเชิงวัตถุ (Object Oriented) หรือเรียกว่า C-K metrics ซึ่งเป็นมาตรวัดซอร์สโค้ดเชิงวัตถุ (Object Oriented) นำเสนอโดย Chidamber และ Kemerer
8	มาตรวัดจำนวนเมธอดต่อคลาส (Weighted Methods per Class: WMC)	จำนวนเมธอดต่อคลาสที่ใช้ในการพัฒนาซอฟต์แวร์ รวมถึงแก้ไข และบำรุงรักษาซอฟต์แวร์
9	มาตรวัดความรับผิดชอบของคลาส (Response For a Class: RFC)	จำนวนเมธอดภายในคลาสที่สามารถตอบสนอง การเรียกใช้งานจากคลาสอื่นได้ หรือจากเมธอดบางตัวภายในคลาสนั้น
10	มาตรวัดระดับความลึกของการสืบทอดคุณสมบัติของคลาส (Depth of Inheritance Hierarchy: DIT)	ค่าระดับความลึกของการสืบทอดคุณสมบัติ (Inheritance) ของแต่ละคลาส
11	มาตรวัดจำนวนคลาสลูก (Number of Children: NOC)	จำนวนคลาสลูกทั้งหมดที่สืบทอดลงมาจากคลาสแม่
12	มาตรวัดการเข้าคู่กันระหว่างวัตถุ (Coupling Between Objects: CBO)	ค่าระดับความเกี่ยวข้องหรือพึ่งพากันของ Module ต่างๆ สำหรับซอฟต์แวร์เชิงวัตถุ
13	มาตรวัดระดับของการขาดความสัมพันธ์ภายในคลาส (Lack of Cohesion of Methods: LCOM)	ค่าขาดการเกาะกันเป็นก้อน (Cohesion) ของคลาส
14	Adaptive Assessment Tools	เป็นเครื่องมือที่ใช้เทคโนโลยีเพื่อปรับเปลี่ยนคำถามหรือเนื้อหาการประเมินตามระดับความสามารถหรือความรู้ของผู้เรียนแบบเรียลไทม์
15	Machine Learning	เกี่ยวข้องกับการสร้างแบบจำลอง (model) เพื่อให้คอมพิวเตอร์สามารถเรียนรู้และตัดสินใจหรือทำนายผลลัพธ์โดยอัตโนมัติ โดยไม่ต้องถูกตั้งโปรแกรมเฉพาะเจาะจงสำหรับทุกสถานการณ์

ลำดับ	คำศัพท์	ความหมาย
16	Random Forest	เป็นอัลกอริทึมการเรียนรู้ของเครื่องที่มีพื้นฐานจาก Decision Trees โดยรวมกลุ่มต้นไม้หลายต้นในลักษณะการเรียนรู้แบบ Ensemble เพื่อเพิ่มความแม่นยำลดความเอนเอียงและลดการกระจายตัว

2.1 มาตรวัดซอฟต์แวร์

การวัดซอฟต์แวร์ (Software Measurement) คือการแสดงขนาด จำนวน ปริมาณ หรือมิติของคุณลักษณะบางประการของกระบวนการหรือผลิตภัณฑ์ใดผลิตภัณฑ์หนึ่ง เมตริกของซอฟต์แวร์ (Software metric) คือการวัดระดับที่สิ่งใดสิ่งหนึ่งเป็นส่วนหนึ่งของผลิตภัณฑ์หรือกระบวนการของระบบ เมตริกซอฟต์แวร์เป็นการประเมินที่สามารถวัดหรือนับได้ของคุณลักษณะต่างๆ ของผลิตภัณฑ์ซอฟต์แวร์ (“Software Measurement and Metrics,” 2019) ในช่วงระยะเวลาของวงจรชีวิตซอฟต์แวร์ เมตริกซอฟต์แวร์จะให้ข้อมูลที่สนับสนุนการตัดสินใจในด้านการจัดการ ข้อมูลที่สนับสนุนการประเมินความเสี่ยงและการตัดสินใจบรรเทาความเสี่ยงในการพัฒนาซอฟต์แวร์เป็นสิ่งจำเป็นเพื่อให้สามารถสนับสนุนได้อย่างเพียงพอ (Fenton & Neil, 2000)

มาตรวัดซอฟต์แวร์ถูกกำหนดขึ้นมาเพื่อช่วยให้การวัดมีความง่ายขึ้น ในปัจจุบันได้มีนักวิจัยหลายท่านที่คิดค้นและนำเสนอมาตรวัดใหม่ๆ ขึ้นมา มาตรวัดที่เป็นที่นิยมใช้กันอย่างแพร่หลายและเป็นที่ยอมรับต่างๆ เหล่านี้สามารถแบ่งได้เป็น 2 ประเภทคือ มาตรวัดแบบดั้งเดิม (Traditional Metrics) และมาตรวัดเชิงวัตถุ (Object-Oriented Metrics) รายละเอียดของมาตรวัดแต่ละประเภทมีดังนี้

1) มาตรวัดดั้งเดิม

มาตรวัดที่นิยมใช้ คือ มาตรวัดจำนวนบรรทัด และมาตรวัดความซับซ้อนซึ่งเป็นมาตรวัดที่นำเสนอโดย Thomas McCabe ในปี ค.ศ. 1976 เมตริกซอฟต์แวร์ที่ใช้ประเมินความซับซ้อนของโปรแกรมที่พัฒนาขึ้น

1.1) มาตรวัดจำนวนบรรทัด (Lines of Code: LOC)

การหาค่าจำนวนบรรทัดนั้น จะไม่นับรวมบรรทัดว่าง บรรทัดที่เป็นเครื่องหมายบล็อก ({}) และบรรทัดที่เป็นข้อความอธิบาย (Comment) มากขึ้นด้วย และการประกาศตัวแปรหลายตัวอยู่บนบรรทัดเดียวกันจะนับจำนวนบรรทัดเท่ากับจำนวนตัวแปรที่ประกาศ

1.2) มาตรวัดวัดไซโคลเมตริกของแมคเคบ (Cyclomatic Complexity)

เป็นเมตริกซอฟต์แวร์ที่ใช้ประเมินความซับซ้อนของโปรแกรม โดยอิงจากจำนวนเส้นทางอิสระเชิงเส้นที่สามารถพบได้ในโค้ดต้นฉบับของโปรแกรม (“Cyclomatic Complexity,” 2018; “Cyclomatic Complexity,” 2024; T. J. McCabe, 1976) เมตริกนี้ใช้การแทนโปรแกรมด้วย กราฟการควบคุมการไหล (control-flow graph) ซึ่งแต่ละโหนดแทนชุดคำสั่งที่ไม่สามารถ

แบ่งย่อยได้ และแต่ละขอบ (Edge) แสดงถึงความเป็นไปได้ในการเปลี่ยนลำดับการทำงาน ซึ่งเป็นการวัดค่าความซับซ้อนของการใช้คำสั่งควบคุมในโปรแกรม ได้แก่ คำสั่ง If, While, Repeat และ For ถ้าโปรแกรมมีการใช้คำสั่งควบคุมเป็นจำนวนมาก จะทำให้ยากต่อคุณภาพในด้านความสามารถการทดสอบ (Testability) ความสามารถในการทำความเข้าใจ (Understandability) และความสามารถในการบำรุงรักษาซอฟต์แวร์

2) มาตรวัดเชิงวัตถุ (Object Oriented Design Metrics)

ชุดเมตริกที่เสนอโดย Chidamber และ Kemerer (1994) ซึ่งมักเรียกกันว่า “C-K metrics” ซึ่งเป็นเมตริกสำหรับการประเมินคุณภาพของซอฟต์แวร์เชิงวัตถุ (Object-Oriented Software เมตริกชุดนี้ได้รับการยอมรับอย่างกว้างขวางทั้งในแวดวงวิชาการและอุตสาหกรรมซอฟต์แวร์ เนื่องจากสามารถใช้วัดโครงสร้างและพฤติกรรมของคลาสในเชิงคุณภาพได้อย่างมีประสิทธิภาพ ชุดเมตริกนี้ประกอบด้วยเมตริกระดับคลาส 6 มาตรวัด ได้แก่

2.1) มาตรวัดจำนวนเมธอดต่อคลาส (Weighted Methods per Class: WMC)

เมตริกนี้วัดจำนวนเมธอดทั้งหมดในคลาสหนึ่งๆ โดยอาจนับรวมระดับความซับซ้อนของแต่ละเมธอด หากคลาสมีเมธอดมาก ย่อมสะท้อนถึงความซับซ้อนที่สูงขึ้น ซึ่งอาจทำให้คลาสนั้นยากต่อการเข้าใจและดูแลรักษา

2.2) มาตรวัดระดับความลึกของการสืบทอดคุณสมบัติของคลาส (Depth of Inheritance Hierarchy: DIT)

เป็นการหาค่าระดับความลึกของการสืบทอดคุณสมบัติ (Inheritance) ของแต่ละคลาส การสืบทอดคุณสมบัติที่มากขึ้นทำให้ความซับซ้อนของตัวโปรแกรมเพิ่มมากขึ้น และทำให้ยากต่อการทำนายพฤติกรรมของคลาส และการทำความเข้าใจระบบ ค่าระดับความลึกของการสืบทอดคุณสมบัติ ได้จากการนับจำนวนของระดับชั้น (Level) การสืบทอดคุณสมบัติของแต่ละคลาสที่กำลังพิจารณา ซึ่งคุณสมบัติของการสืบทอดคุณสมบัติเป็นลักษณะสำคัญต่อการวัดคุณภาพในด้านความสามารถนำกลับมาใช้ใหม่ (Reusability)

2.3) มาตรวัดจำนวนคลาสลูก (Number of Children: NOC)

เป็นการนับจำนวนคลาสลูกทั้งหมดที่สืบทอดลงมาจากคลาสแม่ที่กำลังพิจารณาในขณะนั้น มาตรวัดจำนวนคลาสลูกเป็นคุณลักษณะที่สำคัญต่อการวัดคุณภาพในด้านความสามารถในการนำกลับมาใช้ใหม่เช่นเดียวกับมาตรวัดระดับความลึกของการสืบทอดคุณสมบัติของคลาส

2.4) มาตรวัดการเข้าคู่กันระหว่างวัตถุ (Coupling Between Objects: CBO)

การเข้าคู่กัน (Coupling) สำหรับซอฟต์แวร์เชิงวัตถุ นิยมให้ค่าของการเข้าคู่กันมีค่าน้อยๆ เพื่อให้คลาสแยกเป็นอิสระจากกันมากที่สุด เพื่อความสะดวกในการแก้ไขข้อมูลจะได้ไม่ส่งผลกระทบต่อคลาสอื่น การเข้าคู่กันระหว่างคลาสเป็นการแสดงความสัมพันธ์ระหว่างวัตถุเมื่อมีการเรียกใช้ตัวแปรหรือเมธอดระหว่างกัน ดังนั้น ถ้าค่าของการเข้าคู่กันมีค่าสูงจะลดความเป็นมอดูลของคลาส ทำให้ความสามารถนำกลับมาใช้ใหม่ และบำรุงรักษาทำได้ยากขึ้น

2.5) มาตรวัดความรับผิดชอบของคลาส (Response for a Class: RFC)

ความรับผิดชอบของคลาส คือ จำนวนเมธอดภายในคลาสที่สามารถตอบสนองการเรียกใช้งานจากคลาสอื่นได้ หรือจากเมธอดบางตัวภายในคลาสนั้น มาตรวัดนี้แสดงค่าผลรวมของความซับซ้อนผ่านจำนวนเมธอด การเรียกใช้งานจากคลาสอื่น ถ้ามีเมธอดที่ถูกเรียกใช้งานได้เป็นจำนวนมากจะส่งผลให้คลาสมีความซับซ้อน ทั้งในด้านการทดสอบ (Testing) และด้านการแก้ไขข้อผิดพลาด (Debugging) ของซอฟต์แวร์มากขึ้นด้วย

2.6) มาตรวัดระดับของการขาดความสัมพันธ์ภายในคลาส (Lack of Cohesion of Methods: LCOM)

การเกาะกันเป็นก้อน (Cohesion) ของคลาส คือ การที่จะบอกว่าเมธอดของคลาสนั้นมีความสัมพันธ์กับเมธอดอื่นๆ อย่างไร การที่มีค่าของการเกาะกันเป็นก้อนสูง แสดงว่าคลาสมีการออกแบบที่ดี การเกาะกันเป็นก้อนของคลาสจะเป็นการบอกแนวโน้มของการห่อหุ้มของคลาส (Encapsulation) ถ้าค่าของการขาดการเกาะกันเป็นก้อนมีค่าสูงจะทำให้แนวโน้มของการห่อหุ้มของคลาส (Encapsulation) มีค่าต่ำ ซึ่งจะทำให้มีความซับซ้อนของคลาสสูง

2.2 การเรียนรู้แบบปรับได้ (Adaptive Learning)

การเรียนรู้แบบปรับได้ (Adaptive Learning) หมายถึง แนวทางการจัดการเรียนรู้ที่มุ่งเน้นการปรับเนื้อหา กิจกรรม หรือวิธีการเรียนการสอนให้เหมาะสมกับลักษณะเฉพาะของผู้เรียนแต่ละราย เพื่อตอบสนองความแตกต่างระหว่างผู้เรียนแต่ละคนอย่างเหมาะสมและเป็นระบบ (วิชัย วงษ์ใหญ่ และมารุต พัฒผล, 2564) การจัดการเรียนรู้ในลักษณะนี้จึงต้องออกแบบให้มีความยืดหยุ่น โดยสามารถปรับเปลี่ยนเนื้อหา กิจกรรม สื่อการสอน และเกณฑ์การประเมิน ให้สอดคล้องกับศักยภาพและระดับความสามารถของผู้เรียนแต่ละราย ซึ่งองค์ประกอบและขั้นตอนการออกแบบการเรียนรู้แบบปรับได้ (Adaptive Learning) มีดังต่อไปนี้

1) องค์ประกอบสำคัญของ Adaptive Learning

1.1) การวิเคราะห์ความสามารถของผู้เรียน (Learner Diagnosis)

ผู้สอนจะต้องเริ่มจากการสำรวจและวิเคราะห์ระดับความสามารถของผู้เรียนในด้านต่างๆ เช่น ความรู้พื้นฐาน ทักษะเฉพาะทาง และความสามารถในการคิดวิเคราะห์ ทั้งนี้เพื่อกำหนดแนวทางการเรียนรู้ที่สอดคล้องกับจุดเริ่มต้นของแต่ละคน

1.2) การกำหนดจุดประสงค์การเรียนรู้ที่แตกต่างตามศักยภาพ (Differentiated Learning Objectives)

แม้ว่าผู้เรียนทุกคนจะมุ่งสู่ผลลัพธ์การเรียนรู้เดียวกันตามหลักสูตร แต่การตั้งจุดประสงค์ในระดับย่อยควรมีความยืดหยุ่น เพื่อให้เหมาะสมกับความสามารถของแต่ละคน และสร้างความรู้สึกว่า “ทุกคนสามารถประสบความสำเร็จได้”

1.3) การออกแบบกิจกรรมการเรียนรู้ที่หลากหลาย (Flexible Learning Activities)

กิจกรรมการเรียนรู้ควรมีระดับความซับซ้อนที่หลากหลาย โดยผู้เรียนที่มีศักยภาพสูงอาจได้รับโจทย์ที่ท้าทายมากขึ้น ในขณะที่ผู้เรียนที่ยังต้องการเวลาในการพัฒนาจะได้รับกิจกรรมที่มีความซับซ้อนน้อยกว่า แต่ยังคงสอดคล้องกับผลลัพธ์การเรียนรู้เดียวกัน ตัวอย่างเช่น การแก้โจทย์คณิตศาสตร์ในระดับความยากต่างกันแต่มีจุดมุ่งหมายเดียวกันคือ “การเข้าใจแนวคิดเบื้องหลังของการแก้ปัญหา”

1.4) การเลือกใช้สื่อการเรียนรู้ที่เหมาะสม (Personalized Learning Materials)

สื่อการเรียนรู้ที่มีประสิทธิภาพคือสื่อที่สอดคล้องกับประสบการณ์เดิมของผู้เรียนและสามารถเชื่อมโยงความรู้ใหม่ได้อย่างราบรื่น การเลือกสื่อให้เหมาะสมกับผู้เรียนแต่ละกลุ่มจึงเป็นอีกองค์ประกอบสำคัญของ Adaptive Learning

1.5) การประเมินผลเพื่อการพัฒนา (Development-Oriented Assessment)

การประเมินผลใน Adaptive Learning มุ่งเน้นไปที่การติดตามความก้าวหน้าและการให้ข้อมูลย้อนกลับแก่ผู้เรียนมากกว่าการตัดสินว่าผ่านหรือไม่ผ่าน โดยอาจใช้การประเมินแบบ Authentic Assessment หรือ Portfolio ประกอบการสังเกตพฤติกรรมการเรียนรู้ และเน้นให้ผู้เรียนตระหนักถึงพัฒนาการของตนเอง

2) การออกแบบการเรียนรู้แบบ Adaptive Learning

ผู้สอนมีบทบาทสำคัญในการกระตุ้นความเชื่อมั่น ส่งเสริมการเรียนรู้ร่วมกัน ติดตามความก้าวหน้า และให้ข้อมูลย้อนกลับอย่างสร้างสรรค์ โดยไม่มุ่งเน้นการแข่งขัน แต่เน้นการพัฒนาศักยภาพของผู้เรียนให้เต็มความสามารถ Adaptive Learning จึงเป็นแนวทางที่ส่งเสริมให้ผู้เรียนบรรลุเป้าหมายทางการเรียนรู้ตามที่หลักสูตรกำหนด พร้อมทั้งส่งเสริมความเชื่อมั่นในตนเอง ซึ่งเป็นรากฐานของการเรียนรู้อย่างยั่งยืน ซึ่งขั้นตอนการออกแบบการเรียนรู้แบบปรับได้ประกอบด้วย 5 ขั้นตอนสำคัญ ได้แก่

2.1) การวิเคราะห์ระดับความสามารถของผู้เรียน

2.2) การกำหนดจุดประสงค์การเรียนรู้ให้เหมาะสมกับแต่ละระดับ

2.3) การออกแบบกิจกรรมการเรียนรู้ที่หลากหลายแต่สอดคล้องกับผลลัพธ์เดียวกัน

2.4) การเลือกสื่อการเรียนรู้ที่สัมพันธ์กับประสบการณ์และพื้นฐานของผู้เรียน

2.5) การประเมินผลที่มุ่งเน้นพัฒนาและสะท้อนกระบวนการเรียนรู้รายบุคคล

2.3 การเรียนรู้ของเครื่อง (Machine Learning)

Machine Learning (ML) หรือการเรียนรู้ของเครื่อง เป็นสาขาหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligence: AI) ที่มุ่งเน้นให้คอมพิวเตอร์สามารถเรียนรู้จากข้อมูลและพัฒนาแบบจำลองเพื่อช่วยในการตัดสินใจหรือคาดการณ์ผลลัพธ์ได้โดยอัตโนมัติ โดยไม่ต้องมีการตั้งโปรแกรมคำสั่งอย่างชัดเจน ซึ่งแตกต่างจากการเขียนโปรแกรมแบบดั้งเดิมที่ต้องกำหนดกฎเกณฑ์ล่วงหน้าอย่างแน่นอน ใน

อดีต คอมพิวเตอร์สามารถทำงานได้ตามที่โปรแกรมเมอร์ระบุเท่านั้น แต่ในโลกที่ข้อมูลมีปริมาณมากขึ้น การเขียนโค้ดที่ครอบคลุมทุกเงื่อนไขของข้อมูลที่เปลี่ยนแปลงตลอดเวลาเป็นไปได้ยาก Machine Learning จึงเข้ามามีบทบาทสำคัญโดยช่วยให้คอมพิวเตอร์สามารถเรียนรู้และปรับปรุงตนเองจากประสบการณ์ ซึ่งช่วยให้ระบบสามารถประมวลผลข้อมูลจำนวนมากและค้นหารูปแบบที่ซับซ้อนได้โดยอัตโนมัติ ในการทำงานของ Machine Learning สิ่งสำคัญที่ต้องคำนึงถึงคือข้อมูล ข้อมูลเป็นปัจจัยหลักที่ทำให้โมเดลสามารถเรียนรู้และทำงานได้อย่างมีประสิทธิภาพ ยิ่งข้อมูลมีคุณภาพสูงและมีปริมาณมาก โมเดลก็จะสามารถเรียนรู้ได้ดีขึ้น เมื่อได้รับข้อมูลมาแล้ว ข้อมูลมักจะต้องผ่านการประมวลผลและทำความสะอาดก่อน เนื่องจากข้อมูลที่ได้มักจะมีข้อผิดพลาด เช่น ข้อมูลซ้ำซ้อน ข้อมูลที่ขาดหาย หรือข้อมูลที่มีค่าผิดปกติ การทำความสะอาดข้อมูล (Data Cleaning) และการเลือกคุณลักษณะที่สำคัญ (Feature Engineering) เป็นกระบวนการที่จำเป็นก่อนที่โมเดลจะเริ่มเรียนรู้ หลังจากที่ข้อมูลพร้อมแล้ว ขั้นตอนต่อไปคือการเลือกอัลกอริทึมที่เหมาะสม มีอัลกอริทึม Machine Learning หลายประเภท เช่น Decision Tree, Random Forest, Support Vector Machine (SVM) และ Neural Networks แต่ละอัลกอริทึมมีข้อดีและข้อเสียแตกต่างกัน และสามารถแบ่งออกเป็น 3 ประเภทหลัก ได้แก่ Supervised Learning, Unsupervised Learning และ Reinforcement Learning ใน Supervised Learning โมเดลจะเรียนรู้จากข้อมูลที่มีป้ายกำกับ ซึ่งหมายความว่าในชุดข้อมูลตัวอย่าง มีคำตอบที่ถูกต้องให้โมเดลได้เรียนรู้ เช่น การจำแนกอีเมลเป็น Spam หรือ Not Spam หรือการทำนายราคาบ้านจากปัจจัยต่างๆ เช่น ขนาดบ้านและทำเลที่ตั้ง ในทางตรงกันข้าม Unsupervised Learning เป็นการเรียนรู้จากข้อมูลที่ไม่มีป้ายกำกับ โมเดลจะพยายามค้นหารูปแบบหรือโครงสร้างของข้อมูลเอง เช่น การแบ่งกลุ่มลูกค้าตามพฤติกรรมการซื้อสินค้าโดยไม่ทราบล่วงหน้าว่าลูกค้าแต่ละกลุ่มมีลักษณะอย่างไร ส่วน Reinforcement Learning เป็นการเรียนรู้แบบใช้ระบบรางวัลและบทลงโทษเพื่อให้โมเดลสามารถตัดสินใจที่ดีที่สุด เช่น การพัฒนาระบบ AI สำหรับเล่นเกมที่สามารถปรับตัวและพัฒนากลยุทธ์ได้เอง ในปัจจุบัน Machine Learning ถูกนำไปใช้งานอย่างแพร่หลายในหลากหลายอุตสาหกรรม เช่น ในด้านการแพทย์ ใช้ในการวิเคราะห์ภาพถ่ายทางการแพทย์เพื่อช่วยในการวินิจฉัยโรค การวิเคราะห์ข้อมูลจีโนมเพื่อพัฒนาแนวทางการรักษาเฉพาะบุคคล ในธุรกิจและการตลาด

2.4 อัลกอริทึม Random Forest

Random Forest เป็นอัลกอริทึมการเรียนรู้ของ Machine Learning ที่มีพื้นฐานจาก Decision Trees โดยรวมกลุ่มต้นไม้หลายต้นในลักษณะการเรียนรู้แบบ Ensemble เพื่อเพิ่มความแม่นยำ ลดความเอนเอียงและลดการกระจายตัวของการทำนาย แนวคิดหลักของ Random Forest คือการใช้การสุ่มตัวอย่างข้อมูลแบบ Bagging (Bootstrap Aggregating) และการสุ่มเลือกฟีเจอร์ในระหว่างการสร้าง Trees แต่ละ Trees เพื่อความหลากหลายและลดการพึ่งพาข้อมูลหรือฟีเจอร์เฉพาะที่อาจส่งผลต่อผลลัพธ์ ซึ่งการรวมผลจากต้นไม้หลายต้นจะช่วยให้การตัดสินใจของระบบมี

ความแม่นยำมากยิ่งขึ้น โดยกระบวนการทำงานเริ่มต้นจากการสุ่มตัวอย่างข้อมูลจากชุดข้อมูลต้นฉบับเพื่อสร้างชุดข้อมูลย่อยสำหรับฝึก Tree แต่ละตัวใน Decision Trees จากนั้น Tree จะถูกสร้างขึ้นโดยสุ่มเลือกฟีเจอร์บางส่วนในแต่ละ node โดยเมื่อสร้าง Tree ครบแล้วจะรวมผลลัพธ์จาก Tree ทั้งหมดเพื่อตัดสินใจ โดยสำหรับ Classification จะใช้การลงคะแนนเสียงของผลลัพธ์จาก Tree และสำหรับ Regression จะใช้ค่าเฉลี่ยของผลลัพธ์ซึ่งมีข้อดีคือสามารถจัดการกับปัญหาการ Overfitting ที่จะมีความแม่นยำสูงมากเมื่อประเมินด้วยชุดข้อมูลเดิมแต่เมื่อใช้งานกับข้อมูลใหม่หรือชุดข้อมูลที่ไม่เคยเห็นมาก่อนผลลัพธ์มักจะไม่แม่นยำ ซึ่งการมี Tree ที่มากและหลากหลายจะช่วยลดความลำเอียงและเพิ่มความแม่นยำได้ นอกจากนี้ Random Forest ยังสามารถรองรับข้อมูลที่มีฟีเจอร์จำนวนมาก เช่น Numerical, Categorical, Text, Date/Time และการมีข้อมูลที่ไม่เกี่ยวข้องหรือผิดปกติในชุดข้อมูล (noise) ได้ดี อีกทั้งยังช่วยวัดความสำคัญของฟีเจอร์ในชุดข้อมูลได้อย่างมีประสิทธิภาพ อย่างไรก็ตาม Random Forest มีข้อเสียในด้านความซับซ้อนของการประมวลผลและการใช้ทรัพยากรสูงเมื่อเทียบกับโมเดลง่ายๆ เช่น Logistic Regression รวมถึงการตีความผลลัพธ์ที่ยากกว่าเมื่อเปรียบเทียบกับ Decision Trees เดี่ยวและในบางกรณีที่ข้อมูลมี noise มาก Random Forest อาจทำให้ผลลัพธ์ด้อยลงได้

2.5 งานวิจัยที่เกี่ยวข้อง

การประเมินความยากของโจทย์เขียนโปรแกรมเป็นหัวข้อสำคัญในวงการศึกษาวิทยาการคอมพิวเตอร์ โดยเฉพาะในบริบทของการออกแบบข้อสอบและการประเมินผลที่ยึดตามความสามารถของผู้เรียน งานวิจัยที่ผ่านมาเสนอวิธีการที่หลากหลายในการประเมินความยากของโจทย์ ซึ่ง Software Metrics ได้รับความสนใจมากขึ้นในฐานะเครื่องมือเชิงปริมาณที่ช่วยประเมินความซับซ้อนของโค้ดอย่างเป็นระบบ

Yoshee Jain และ Kathryn Cunningham (2023) ได้นำเสนอกรอบการประเมินการเรียนรู้ของผู้เขียนโปรแกรมมือใหม่ โดยใช้เมตริกประกอบด้วย 3 องค์ประกอบ ได้แก่ ความถี่ในการใช้งานในวิชาชีพ (Frequency in Professional Use) ความเข้าใจได้สำหรับผู้เริ่มต้น (Comprehensibility to Novices) และความสอดคล้องกับบริบทของโดเมน (Alignment with Domain Context) โดยระบุว่าแผนการเรียนรู้ควรตั้งอยู่บนพื้นฐานของโครงสร้างโค้ดที่มีการใช้จริง เข้าใจได้ และเชื่อมโยงกับเป้าหมายของงานเขียนโปรแกรม

ด้านการประเมินความยากของโจทย์ Said Elnaffar (2016) ได้พัฒนาเครื่องมือชื่อว่า Predicted Difficulty Index (PDI) ซึ่งใช้ Software Metrics จากแนวคำตอบ (Sample solutions) เพื่อจัดอันดับความยากของโจทย์เขียนโปรแกรม เครื่องมือนี้มีบทบาทในการสนับสนุนการออกแบบข้อสอบให้มีความเหมาะสมกับระดับของผู้เรียน รวมถึงการกำหนดคะแนนให้สอดคล้องกับความยากที่แท้จริงของโจทย์

Kasto และ Whalley (2013) ศึกษาการใช้ Software Metrics ในการประเมินความยากของภารกิจความเข้าใจโค้ด (Code comprehension tasks) ในการสอบของผู้เรียนชั้นปีแรกของการเรียนการเขียนโปรแกรม โดยพบว่าเมตริก เช่น Cyclomatic Complexity สามารถสะท้อนความยากของโค้ดได้อย่างชัดเจน ผลการศึกษานี้แสดงให้เห็นว่าเมตริกสามารถช่วยให้อาจารย์สามารถออกแบบโจทย์ให้เหมาะสมกับระดับของผู้เรียน และลดความคลาดเคลื่อนในการวัดผลการเรียนรู้

Whalley และ Kasto (2014) ขยายผลการศึกษาเพิ่มเติม โดยยืนยันว่า Cyclomatic Complexity เป็นตัวชี้วัดที่มีประสิทธิภาพในการประเมินความซับซ้อนของการควบคุมลำดับการทำงานในโปรแกรม และมีความสัมพันธ์กับความยากของโจทย์การเขียนโปรแกรม กล่าวคือ ค่ายิ่งสูงยิ่งสะท้อนถึงโค้ดที่ยากต่อความเข้าใจและการเขียน ซึ่งสอดคล้องกับข้อค้นพบที่ว่า เมตริกสามารถใช้ในการออกแบบและจัดลำดับความยากของคำถามได้

นอกจากนี้ Sheard et al. (2013) และ Chung และ Hsiao (2022) ยังเน้นย้ำถึงความสำคัญของการวิเคราะห์ความซับซ้อนของโจทย์ว่าเป็นปัจจัยสำคัญที่ส่งผลต่อผลสัมฤทธิ์ทางการเรียนของผู้เรียน การออกแบบข้อสอบที่มีระดับความยากที่เหมาะสมจึงเป็นองค์ประกอบสำคัญในการประเมินทักษะและความรู้ที่คาดหวังเมื่อสิ้นสุดการเรียนการสอน

จากงานวิจัยที่กล่าวมาข้างต้น สามารถสรุปได้ว่า Software Metrics โดยเฉพาะเมตริกด้านความซับซ้อน เช่น Cyclomatic Complexity, LOC, และ Coupling มีศักยภาพสูงในการนำมาใช้ประเมินระดับความยากของโจทย์การเขียนโปรแกรมอย่างมีประสิทธิภาพ และยังสามารถนำมาใช้ร่วมกับเครื่องมือเชิงวิเคราะห์เพื่อสนับสนุนการออกแบบข้อสอบที่มีความยุติธรรมและแม่นยำมากขึ้น

ระบบการเรียนรู้แบบปรับตัว (Adaptive Learning Systems) ได้รับการศึกษาอย่างกว้างขวางในหลากหลายสาขา ไม่ว่าจะเป็นคณิตศาสตร์ การเรียนรู้ภาษาต่างประเทศ และการศึกษาด้านวิทยาการคอมพิวเตอร์ ระบบรุ่นแรกมักอิงกับกฎที่ผู้เชี่ยวชาญกำหนดไว้ล่วงหน้า (rule-based) โดยมีจุดมุ่งหมายเพื่อควบคุมเส้นทางการเรียนรู้ของผู้เรียน ตัวอย่างสำคัญของระบบลักษณะนี้ เช่น ASSISTments (Heffernan & Heffernan, 2014) และระบบติวเตอร์อัจฉริยะรุ่นแรก (Vanlehn, 2006) ซึ่งให้ข้อเสนอแนะแบบคงที่และมีความสามารถในการปรับให้เหมาะสมกับผู้เรียนในระดับจำกัด อย่างไรก็ตาม ระบบแบบปรับตัวรุ่นใหม่มีแนวโน้มที่จะสร้างแบบจำลองผู้เรียนแบบไดนามิก (Dynamic learner model) โดยอิงจากพฤติกรรมการโต้ตอบและข้อมูลเชิงวิเคราะห์แบบเรียลไทม์ (Learning analytics) ซึ่งช่วยให้ระบบสามารถปรับการเรียนรู้ได้อย่างยืดหยุ่นและตรงกับความต้องการของแต่ละบุคคลมากยิ่งขึ้น จากการทบทวนวรรณกรรมโดย Barbosa et al. (2024) พบว่าการประยุกต์ใช้ระบบ adaptive learning มีความแพร่หลายในสาขาวิทยาการคอมพิวเตอร์ โดยเฉพาะในหัวข้อการเขียนโปรแกรมและฐานข้อมูล โดยเน้นไปที่การประเมินความรู้พื้นฐานของผู้เรียน สไตล์การเรียนรู้ และพฤติกรรมการเรียน

นอกจากนี้ งานวิจัยในช่วงหลังยังให้ความสนใจกับการผสาน แนวคิดเกมมิฟิเคชัน (Gamification) เข้ากับระบบการเรียนรู้แบบปรับตัว เพื่อเสริมสร้างแรงจูงใจและความมีส่วนร่วมของ

ผู้เรียน ตัวอย่างหนึ่งคือ Gridlock ซึ่งเป็นเกมเล่าเรื่องแบบปรับตัวที่ใช้ตัวจำแนกแบบ Random Forest ในการปรับคำใบ้และการแทรกแซงตามพฤติกรรมของผู้เล่นแบบเรียลไทม์ (Ying Tang & Hare, 2020) นอกจากนี้ Bennani et al. (2022) ยังพบว่า กลยุทธ์ adaptive gamification ที่ออกแบบองค์ประกอบของเกมตามโปรไฟล์ผู้เรียน มีประสิทธิภาพในการส่งเสริมผลลัพธ์การเรียนรู้มากกว่าการใช้กลยุทธ์เกมมิฟิเคชันแบบคงที่

ในแง่ของการใช้ ข้อมูลการเรียนรู้ (Learning Analytics) และ การพยากรณ์ (Predictive modeling) ระบบ Adaptive learning ในยุคปัจจุบันมักผสานอัลกอริธึมการเรียนรู้ของเครื่อง เช่น K-Nearest Neighbors และ Random Forests เพื่อคาดการณ์ความสำเร็จทางการเรียน และเพื่อวางแผนการแทรกแซงในระดับกลุ่มเป้าหมาย (Rincón-Flores et al., 2020) อย่างไรก็ตาม ความสามารถในการตีความผลลัพธ์ของโมเดล และการปรับใช้เชิงเวลาแบบเรียลไทม์ยังคงเป็นความท้าทาย Luan และ Tsai (2021) ได้ทบทวนการใช้ Machine Learning ในบริบทของ Precision Education และพบว่า วิธีแบบ supervised learning ยังคงเป็นแนวทางหลักในการคาดการณ์ผลลัพธ์และการปรับการเรียนเฉพาะบุคคล แม้ยังมีข้อจำกัดในการเชื่อมโยงผลการพยากรณ์กับการออกแบบการเรียนการสอนจริง

ในบริบทของ การเรียนรู้ด้านการเขียนโปรแกรม ระบบในยุคแรก เช่น Web-CAT และ CodeLab ถูกใช้เพื่อให้ข้อเสนอแนะแบบอัตโนมัติจากการส่งโค้ดของผู้เรียน โดยเฉพาะในรายวิชา CS1 งานของ Barr และ Trytten (2016) แสดงให้เห็นว่า CodeLab ช่วยให้ผู้เรียนได้รับข้อเสนอแนะแบบทันที (immediate feedback) และสามารถฝึกฝนอย่างต่อเนื่อง ส่งผลให้ผู้เรียนมีความเข้าใจในแนวคิดพื้นฐานของการเขียนโปรแกรมเพิ่มขึ้น อย่างไรก็ตาม ระบบประเภท rule-based ดังกล่าวแม้จะให้ข้อเสนอแนะแบบโปร่งใสและเหมาะกับการประเมินรูปแบบฟอร์แมทีฟ แต่กลับขาดความสามารถในการปรับตัวและไม่สามารถตอบสนองต่อความแตกต่างของผู้เรียนได้อย่างเพียงพอ (Edwards & Perez-Quinones, 2008)

งานวิจัยล่าสุดพยายามแก้ไขข้อจำกัดเหล่านี้ผ่านระบบการเรียนรู้แบบปรับได้ (Adaptive learning) ที่ออกแบบเฉพาะสำหรับบริบทการเรียนเขียนโปรแกรม เช่น งานวิจัยของ Lohr et al. (2024) ที่พัฒนาระบบซึ่งให้ข้อเสนอแนะแบบฟอร์แมทีฟแบบปรับตัว โดยอิงจากผลการเรียนแบบเรียลไทม์ เพื่อส่งเสริมการมีส่วนร่วมและการเรียนรู้เชิงลึกยิ่งขึ้น นอกจากนี้ ยังมีการประยุกต์ใช้เทคนิคการเรียนรู้ของเครื่อง (Machine Learning) เพื่อปรับแต่งข้อเสนอแนะแบบเจาะจงตามพฤติกรรมของผู้เรียน โดยศึกษาลักษณะการเขียนโค้ดและใช้โมเดลวิเคราะห์เพื่อกำหนดการแทรกแซงที่เหมาะสม (Alshammari, 2025)

จากงานวิจัยที่กล่าวมา ข้อค้นพบหลักชี้ให้เห็นว่าการออกแบบระบบการเรียนรู้แบบปรับตัวโดยอิงจากข้อมูลผู้เรียนและเมตริกที่ตีความได้ มีแนวโน้มที่จะส่งเสริมผลสัมฤทธิ์ทางการเรียนและประสบการณ์การเรียนรู้ได้ดียิ่งขึ้น โดยเฉพาะในรายวิชาการเขียนโปรแกรมที่ผู้เรียนมีความแตกต่างกันอย่างหลากหลาย

จากการทบทวนงานวิจัยที่เกี่ยวข้องพบว่า Machine Learning มีบทบาทสำคัญในการยกระดับการเรียนรู้ด้านการเขียนโปรแกรม โดยสามารถเชื่อมโยงทั้งการประเมินความยากของโจทย์ การวิเคราะห์พฤติกรรมผู้เรียน และการออกแบบระบบการเรียนรู้แบบปรับตัวอย่างมีประสิทธิภาพ งานวิจัยจำนวนมากชี้ให้เห็นว่าเมตริกซอฟต์แวร์ เช่น Cyclomatic Complexity และ Lines of Code (LOC) สามารถนำมาใช้เป็นฟีเจอร์ในการพยากรณ์ระดับความยากของโจทย์ด้วยโมเดล Machine Learning เพื่อช่วยออกแบบข้อสอบให้เหมาะสมกับระดับของผู้เรียน ขณะเดียวกันการวิเคราะห์โปรไฟล์ผู้เรียนโดยอิงจากความเข้าใจได้ ความถี่ในการใช้งาน และความเชื่อมโยงกับบริบทของโดเมน ช่วยให้โมเดลสามารถจำแนกระดับผู้เรียนและแนะนำเนื้อหาที่เหมาะสมเป็นรายบุคคล ระบบ Adaptive Learning รุ่นใหม่จึงมักผสานอัลกอริธึม Machine Learning เช่น Random Forest และ K-Nearest Neighbors เพื่อสร้างแบบจำลองผู้เรียนแบบไดนามิกจากข้อมูลการเรียนรู้ของผู้เรียนแบบเรียลไทม์ ทั้งด้านการแนะนำข้อเสนอแนะแบบเจาะจงและการพยากรณ์แนวโน้มความสำเร็จทางการเรียน งานวิจัยล่าสุดยังชี้ให้เห็นว่าแม้ Machine Learning จะมีศักยภาพสูงใน Precision Education แต่ประเด็นด้านความสามารถในการตีความผลลัพธ์และการนำไปใช้งานจริงยังคงเป็นความท้าทายที่ควรได้รับการพัฒนาอย่างต่อเนื่อง ทั้งนี้ การบูรณาการ Machine Learning เข้ากับการออกแบบกิจกรรมการเรียนรู้และข้อสอบอย่างเป็นระบบ จะเป็นรากฐานสำคัญของการพัฒนาระบบการเรียนการสอนที่ยืดหยุ่น ยุติธรรม และสอดคล้องกับศักยภาพของผู้เรียนแต่ละคน

2.6 เทคโนโลยีและเฟรมเวิร์กที่เกี่ยวข้อง

2.6.1 Lizard

ผู้พัฒนาเครื่องมือ Lizard ขึ้นมาคือ Mr. Terry Yin (Yin, 2012/2025) โดยเครื่องมือ Lizard เป็นเครื่องมือสำหรับวัดค่าความซับซ้อนของ Cyclomatic ที่ขยายได้สำหรับภาษาโปรแกรมต่างๆ รวมถึงภาษา C/C++ (ไม่ต้องใช้ Header File หรือ Java Imports) นอกจากนี้ยังตรวจจับการการวิเคราะห์ซอร์สโค้ดแบบคงที่ (Static code analysis) โดยมีสัญลักษณ์ดังภาพที่ 2-1

ภาพที่ 2-1 สัญลักษณ์ของ Lizard

วิธีการใช้งานคือการรันคำสั่งผ่านทาง Terminal หรือ Command Prompt โดยจำเป็นจะต้องเขียนให้อยู่ในรูปแบบ “lizard [options] [PATH or FILE] [PATH] ...” โดย “Options” ที่สามารถใช้งานได้มีทั้งหมด 22 Options ดังตารางที่ 2-2

ตารางที่ 2-2 รายละเอียด Options Lizard

ลำดับ	Options	รายละเอียด
1	-h, --help	แสดงข้อความช่วยเหลือ และ Options ต่างๆ
2	--version	แสดงเวอร์ชันของ Lizard
3	-l, --languages	การวัดค่าความซับซ้อนตามภาษาที่ต้องการ เช่น “lizard -l cpp -l java”
4	-V, --verbose	แสดงผลลัพธ์ในโหมดของ Verbose (ชื่อฟังก์ชันแบบยาว)
5	-C, --CCN	เป็นการกำหนดจำนวนความซับซ้อนแบบไซโคลมาติก ซึ่งค่าเริ่มต้นคือ 15 โดยฟังก์ชันที่มี CCN มากกว่าจะสร้างการแจ้งเตือน
6	-f, --input_file	เป็นการรับรายชื่อไฟล์จากไฟล์ที่กำหนด
7	-o, --output_file	เป็นการนำผลลัพธ์ที่ได้ลงไฟล์ที่กำหนด เช่น “lizard -o no_v.txt” ซึ่งสามารถใช้ร่วมกับ –xml, --csv หรือ –html
8	-L, --length	เป็นการกำหนดความยาวของฟังก์ชันสูงสุด ซึ่งค่าเริ่มต้นคือ 1,000 โดยฟังก์ชันที่มีความยาวมากกว่าจะมีการแจ้งเตือน
9	-a, --argument	เป็นการกำหนดจำนวนของพารามิเตอร์ ซึ่งจำนวนเริ่มต้นคือ 100 โดยฟังก์ชันที่มีจำนวนพารามิเตอร์มากกว่า 100 จะมีการแจ้งเตือน
10	-w, --warning_only	แสดงการแจ้งเตือนเท่านั้น โดยใช้รูปแบบคำเตือนของ clang/gcc
11	--warning-msvs	แสดงการแจ้งเตือนเท่านั้น โดยใช้รูปแบบคำเตือนของ Visual Studio
12	-x, --exclude	เป็นการยกเว้นไฟล์ที่ตรงกับรูปแบบที่กำหนด โดย “*” หมายถึง exclude file ทั้งหมดที่ exclude ได้เช่น ‘lizard -x “./folde/*”’

ลำดับ	Options	รายละเอียด
13	-t, --working_threads	เป็นการกำหนดเธรดการทำงาน ค่าเริ่มต้นคือ 1 ยิ่งเพิ่มจำนวนเธรดมากก็ทำให้สามารถใช้ CPU ได้อย่างเต็มที่และเร็วขึ้น
14	-i, --ignore_warnings	ถ้าหากจำนวนการแจ้งเตือน (Warnings) น้อยกว่าหรือเท่ากับจำนวนที่กำหนด เครื่องมือจะแสดงผลตามปกติ ถ้าหากจำนวนที่กำหนดมีค่าเป็นจำนวนลบ เครื่องมือจะแสดงผลตามปกติ โดยไม่คำนึงถึงจำนวนการแจ้งเตือน ซึ่งมีประโยชน์ในการ makefile legacy code เช่น “lizard -i 20” หรือ “lizard -i -1”
15	-X, --xml	เป็นการส่งไฟล์ออกในรูปแบบ XML ซึ่งมักใช้คู่กับคำสั่ง -o เช่น “lizard -o no_v.xml --xml”
16	--csv	เป็นการส่งไฟล์ออกในรูปแบบ CSV ซึ่งมักใช้คู่กับคำสั่ง -o เช่น “lizard -o no_v.csv --csv”
17	-H, --html	เป็นการส่งไฟล์ออกในรูปแบบ HTML ซึ่งมักใช้คู่กับคำสั่ง -o เช่น “lizard -o no_v.html --html”
18	-m, --modified	คำนวณจำนวนความซับซ้อน Cyclomatic Complexity Number ซึ่งจะนับ Switch/Case ที่มีหลายกรณีให้เป็น 1 ตัว
19	-E, --extenstion	เป็นการใช้งานส่วนขยายต่างๆ ของ Lizard เช่น “lizard -Eduplicate <path to your code>” ที่เป็นการตรวจสอบการคัดลอกซอร์สโค้ด ซึ่งผลลัพธ์จะออกมาเป็นเปอร์เซ็นของการคัดลอก
20	-s, --sort	เป็นการเรียงคำเตือนของฟิลด์ต่าง โดยฟิลด์สามารถเป็น nloc, cyclomatic_complexity, token_count, Parameter_count ฯลฯ หรือฟิลด์ที่กำหนดเอง
21	-T, --Threshold	กำหนดค่าจำกัดของฟิลด์ โดยฟิลด์สามารถเป็น nloc, cyclomatic_complexity, token_count, parameter_count ฯลฯ lizard จะรายงานคำเตือนหากฟังก์ชันเกินขีดจำกัด เช่น “lizard -T nloc=100”
22	-W, --whitelist	เป็นการกำหนดพาธและชื่อไฟล์ไปยังไฟล์ไวท์ลิสต์ คือ './whitelizard.txt'

ลำดับ	Options	รายละเอียด
		โดยค่าเริ่มต้น “.” ไฟล์ whitelist คือไฟล์สำหรับกำหนดไม่สนใจคำเตือนของ Lizard

2.6.2 Chidamber and Kemerer Java Metrics (CKJM)

ผู้พัฒนาเครื่องมือ CKJM ขึ้นมาคือ Mr. Diomidis Spinellis โดย CKJM เป็นเครื่องมือสำหรับคำนวณเมตริกเชิงวัตถุของ Chidamber และ Kemerer โดยการประมวลผล Bytecode ของไฟล์ Java ที่คอมไพล์แล้ว โปรแกรมจะคำนวณสำหรับแต่ละคลาสตาม 8 เมตริกที่เสนอโดย Chidamber และ Kemerer โดยวิธีการใช้งานเครื่องมือ Chidamber and Kemerer Java Metrics (CKJM) คือ การรันคำสั่งผ่านทาง Terminal หรือ Command Prompt จำเป็นจะต้องเขียนให้อยู่ในรูปแบบที่ใช้งานได้มีอยู่ทั้งหมด 4 รูปแบบ ดังตารางที่ 2-3

ตารางที่ 2-3 รายละเอียด Options Chidamber and Kemerer Java Metrics

ลำดับ	รูปแบบ	รายละเอียด
1	find build -name '*.class' -mtime -7 -print | java -jar [Path File ckjm.jar]	แสดงผลลัพธ์เมตริกจากไฟล์คลาสที่แก้ไขในช่วงสัปดาห์ที่แล้ว
2	find build -name ' *.class ' | fgrep -v ' $ ' | java -jar [Path File ckjm.jar]	ไม่แสดงผลลัพธ์ของเมตริกที่มี “$” ในไฟล์คลาส
3	find build lib -name ' *.class ' -print | java -jar [Path File ckjm.jar]	เป็นคำสั่งสำหรับประมวลผลไฟล์คลาสที่อยู่หลายโฟลเดอร์
4	java -jar [Path File ckjm.jar] [Path File .class] | [ตำแหน่งไฟล์ ckjm2xml] >[ตำแหน่งไฟล์ที่ต้องการเก็บไฟล์ XML]”	เป็นคำสั่งสำหรับประมวลผลไฟล์คลาสและนำผลลัพธ์ลงไฟล์ XML

2.7 เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์

1) Node.js

สภาพแวดล้อมการทำงานของภาษา JavaScript ที่ทำงานด้วย V8 engine ซึ่งให้ผู้พัฒนาโปรแกรมสามารถพัฒนาแอปพลิเคชันแบบ Command line แอปพลิเคชัน Desktop หรือเว็บเซิร์ฟเวอร์ได้ (The OpenJS Foundation, n.d.) โดยที่ Node.js จะมี APIs ที่เราสามารถใช้สำหรับทำงานกับระบบปฏิบัติการ เช่น การรับค่าและการแสดงผล การอ่านเขียนไฟล์ และการทำงานกับเน็ตเวิร์ก ซึ่ง Node.js มีสัญลักษณ์ดังภาพที่ 2-2

ภาพที่ 2-2 สัญลักษณ์ของ Node.js

2) Vue.js

เป็น Progressive Framework สำหรับสร้าง User Interface (Evan You, n.d.) ซึ่ง Progressive ก็คือการที่ Vue.js นั้นใช้เป็นเหมือนส่วนเพิ่มความสามารถที่เอาไปใช้งานกับ HTML ซึ่งชื่อ Vue นั้นก็มีที่มาจากการที่ไลบรารีนี้จะโฟกัสที่การจัดการในส่วนของ View เท่านั้น เพื่อให้สามารถนำ Vue ไปใช้งานร่วมกับไลบรารีอื่นๆ ได้สะดวกและยังเหมาะกับการทำ Single-Page Applications (SPA) และมีสัญลักษณ์ของ Vue.js ดังภาพที่ 2-3

ภาพที่ 2-3 สัญลักษณ์ของ Vue.js

3) Nuxt.js

เป็น JavaScript Framework ที่ใช้งานร่วมกับ Vue.js Library (Vue, Vue-router และ Vuex) และเครื่องมือพัฒนาที่มีประสิทธิภาพ (Webpack, Babel และ PostCSS) เป้าหมายของ Nuxt คือการทำให้การพัฒนาเว็บไซต์มีประสิทธิภาพโดยคำนึงถึงประสบการณ์ของนักพัฒนาที่ยอดเยี่ยม (Nuxt, n.d.) มีสัญลักษณ์ของ Nuxt.js ดังภาพที่ 2-4

ภาพที่ 2-4 สัญลักษณ์ของ Nuxt.js

4) Express.js

เป็นเว็บเฟรมเวิร์กที่ใช้สำหรับพัฒนาเว็บแอปพลิเคชันหรือเว็บไซต์บน Node.js ที่ทำงานที่ฝั่งของ Backend ตัวของเฟรมเวิร์กนั้นถูกพัฒนามาจากโมดูล HTTP ซึ่งเป็นโมดูลของ Node.js เอง (OpenJS Foundation, n.d.) ซึ่งเฟรมเวิร์กนี้ช่วยให้ผู้พัฒนาซอฟต์แวร์พัฒนาเว็บแอปพลิเคชันด้วย

Node.js ได้ง่ายขึ้น เช่น การจัดการ route การจัดการข้อผิดพลาด การจัดการ middleware การค้นหาข้อผิดพลาด ซึ่งสัญลักษณ์ Express.js ดังภาพที่ 2-5

ภาพที่ 2-5 สัญลักษณ์ของ Express.js

5) Sequelize

คือ Object Relational Mapping (ORM) เป็นเครื่องมือที่ใช้สำหรับจัดการฐานข้อมูล Relational database ได้แก่ Postgres, MySQL, MariaDB, SQLite และ Microsoft SQL Server ซึ่งมีความสามารถในการ Query ข้อมูล เพิ่ม ลบ และแก้ไขฐานข้อมูลได้มีฟังก์ชันการ Query ข้อมูลที่มี Relation กัน และการทำ Eager load (Sequelize Contributors, n.d.) โดยที่ Sequelize มีสัญลักษณ์ดังภาพที่ 2-6

ภาพที่ 2-6 สัญลักษณ์ของ Sequelize

6) MariaDB

ระบบฐานข้อมูลที่พัฒนาต่อยอดจากระบบฐานข้อมูล MySQL ที่ถูก Fork ออกมาจากซอร์สโค้ดต้นสาย MySQL โดย Michael Widenius ซึ่งเป็นผู้ก่อตั้ง MySQL เดิม (MariaDB, n.d.) โครงสร้างต่างๆ ของ MariaDB แทบจะเรียกได้ว่าเหมือนกับ MySQL เช่น โครงสร้างไฟล์ โครงสร้างของ Database และ Table สามารถใช้งานของ MySQL ได้แทบจะทั้งหมด เพราะฉะนั้นการใช้งาน MariaDB แทนที่ MySQL แทบจะไม่ต้องทำอะไรเพิ่มเติมเลยแม้แต่ Code ของ PHP ก็สามารถที่จะใช้งาน function ของ MySQL ได้เช่นเดิม โดยที่สัญลักษณ์ของ MariaDB แสดงดังภาพที่ 2-7

ภาพที่ 2-7 สัญลักษณ์ของ MariaDB

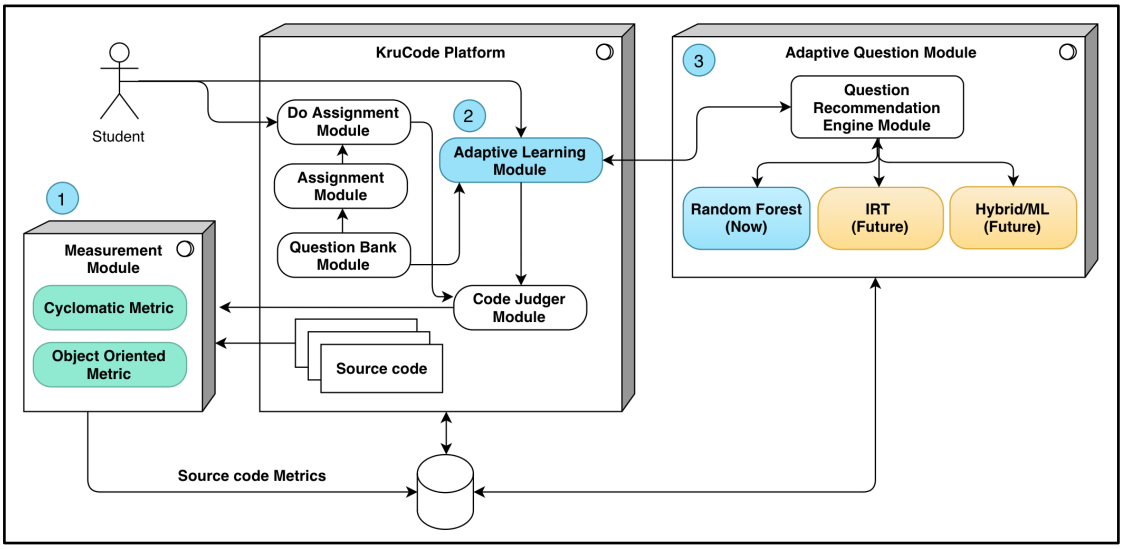
บทที่ 3
รายละเอียดการดำเนินงานวิจัย

การพัฒนามอดูลการวัดความซับซ้อนของซอร์สโค้ดสำหรับแพลตฟอร์มการเรียนรู้การเขียนโปรแกรม ผู้วิจัยจะต้องทำการวิเคราะห์และออกแบบซอฟต์แวร์ตามระเบียบวิธีการออกแบบและพัฒนาซอฟต์แวร์ ตั้งแต่การออกแบบสถาปัตยกรรมซอฟต์แวร์ระดับสูงและรายละเอียดการทำงานซอฟต์แวร์แต่ละส่วนด้วยแผนภาพต่างๆ เช่น แผนภาพยูสเคส แผนภาพกิจกรรม และแผนภาพความสัมพันธ์ของข้อมูล โดยมีรายละเอียดดังต่อไปนี้

3.1 การออกแบบสถาปัตยกรรมซอฟต์แวร์

ภาพรวมของแพลตฟอร์มการเรียนรู้การเขียนโปรแกรม ซึ่งมีผู้ใช้งานหลักคือ อาจารย์และนักศึกษา โดยที่อาจารย์ผู้สอนสามารถจัดการรายวิชา กลุ่มเรียน เพิ่มนักศึกษาเข้ากลุ่มเรียนหรือรายวิชา กำหนดโจทย์และมอบหมายงานให้กับนักศึกษาของรายวิชาการเขียนโปรแกรม ส่วนนักศึกษาสามารถส่งไฟล์ซอร์สโค้ด (Source code) หรือเขียนโปรแกรมผ่านระบบเพื่อให้ระบบตรวจสอบการเขียนโปรแกรมตามชุดข้อมูลทดสอบ (Test set) ที่อาจารย์ผู้สอนกำหนดไว้ และแสดงผลการตรวจสอบผ่านหน้าจอของระบบ ซึ่งก็จะประกอบไปด้วยมอดูลหลักคือ Question Bank Module สำหรับการจัดการโจทย์ปัญหา Assignment Module สำหรับมอบหมายงาน Do Assignment Module สำหรับให้นักศึกษาทำโจทย์ปัญหา ส่วน Code Judger Module เป็นส่วนที่ใช้ในการตรวจสอบความถูกต้องของซอร์สโค้ดตามกรณีทดสอบ (Test case) ที่ได้กำหนดไว้

สำหรับในงานวิจัยนี้ได้มีการออกแบบ Measurement Module, Adaptive Learning Module และ Question Recommendation Module เพิ่มเติม เพื่อใช้สำหรับในการวัดความซับซ้อนของซอร์สโค้ด การเรียนรู้แบบปรับได้ (Adaptive Learning) และการแนะนำโจทย์ปัญหาที่เหมาะสมกับระดับความสามารถของผู้เรียน รวมถึงปรับปรุงแพลตฟอร์มการเรียนรู้การเขียนโปรแกรมเดิมให้เข้ากับมอดูลที่พัฒนาขึ้นใหม่ด้วย โดยสถาปัตยกรรมซอฟต์แวร์ที่คณะผู้วิจัยได้เลือกใช้คือ Service based Architecture โดยที่มอดูลต่างๆ มีหน้าที่การทำงานเฉพาะ แต่ยังคงจัดเก็บอยู่บนฐานข้อมูลเดียวกัน โดยสถาปัตยกรรมซอฟต์แวร์ที่คณะผู้วิจัยได้ทำการออกแบบแสดงดังภาพที่ 3-1



ภาพที่ 3-1 สถาปัตยกรรมของซอฟต์แวร์

3.1.1 มอดูลการวัดความซับซ้อนของซอร์สโค้ด

จากภาพที่ 3-1 หมายเลข 1 ซึ่งเป็น Measurement Module โดยหน้าที่หลักของมอดูลนี้ คือ วัดความซับซ้อนของซอร์สโค้ด ซึ่งการวัดค่านี้จะใช้มาตรวัดคือ มาตรวัด Cyclomatic และมาตรวัดเชิงวัตถุ (Object Oriented Metrics) การทำงานของมอดูลนี้มีขั้นตอนดังต่อไปนี้

1) ขั้นตอนแรกจะเป็นการดึงข้อมูลซอร์สโค้ดแต่ละโจทย์ปัญหาของผู้เรียนจากแพลตฟอร์มการเรียนรู้การเขียนโปรแกรม เพื่อให้เป็นข้อมูลนำเข้า (Input) ในการประมวลผลถัดไป โดยข้อมูลในส่วนนี้จะเป็นหมายเลขโจทย์ปัญหาและไฟล์ซอร์สโค้ด

2) ขั้นตอนประมวลผลความซับซ้อนของซอร์สโค้ด ขั้นตอนนี้จะเป็นการวัดค่าความซับซ้อนของซอร์สโค้ด ซึ่งผู้วิจัยได้เลือกใช้เมตริกมาตรวัดดั้งเดิมและมาตรวัดเชิงวัตถุ เนื่องจากลักษณะการเขียนโปรแกรม (Programming paradigm) ที่ใช้ในการเรียนการสอนโปรแกรมจะเป็นแบบเชิงโครงสร้าง (Structured) และแบบเชิงวัตถุ (Object oriented) โดยรายละเอียดมาตรวัดมีดังนี้

2.1) มาตรวัดดั้งเดิม ประกอบด้วย มาตรวัดจำนวนบรรทัดและมาตรวัดความซับซ้อนซึ่งเป็นมาตรวัดที่นำเสนอโดยแมคเคบ (McCabe) ซึ่งผลลัพธ์ของการวัดค่าความซับซ้อนจะประกอบด้วยจำนวนบรรทัด (Lines of Code: LOC) และค่าความซับซ้อน (Cyclomatic Complexity)

2.2) มาตรวัดเชิงวัตถุ ซึ่งงานวิจัยนี้เลือกใช้มาตรวัดของ Chidamber and Kemerer (Chidamber & Kemerer, 1994) ซึ่งมาตรวัดเป็นที่ยอมรับและนิยมใช้งานกัน โดยค่าผลลัพธ์ประกอบด้วย 6 ค่า ดังนี้

2.2.1) มาตรวัดจำนวนเมธอดต่อคลาส (Weighted Methods per Class: WMC)

2.2.2) มาตรวัดความรับผิดชอบของคลาส (Response For a Class: RFC)

2.2.3) มาตรวัดระดับความลึกของการสืบทอดคุณสมบัติของคลาส (Depth of Inheritance Hierarchy: DIT)

2.2.4) มาตรวัดจำนวนคลาสลูก (Number of Children: NOC)

2.2.5) มาตรวัดการเข้าคู่กันระหว่างวัตถุ (Coupling Between Objects: CBO)

2.2.6) มาตรวัดระดับของการขาดความสัมพันธ์ภายในคลาส (Lack of Cohesion of Methods: LCOM)

3) เมื่อได้ผลลัพธ์จากการวัดค่าความซับซ้อนของซอร์สโค้ดแล้วจะนำข้อมูลส่วนนี้ไปประมวลผลด้วยวิธีการเชิงสถิติแล้วนำไปปรับปรุงค่าความซับซ้อนของโจทย์ปัญหาในแพลตฟอร์มฯ และสร้างรายงานต่างๆ เพื่อให้อาจารย์สามารถใช้ประกอบการตัดสินในการปรับปรุงโจทย์ปัญหา/มอบหมายโจทย์ปัญหาการเขียนโปรแกรมแก่ผู้เรียนถัดไป

3.1.2 มอดูลการเรียนรู้แบบปรับได้

จากภาพที่ 3-1 หมายเลข 2 Adaptive Learning Module เป็นมอดูลที่สนับสนุนให้ผู้เรียนสามารถเรียนรู้การเขียนโปรแกรม โดยที่ระบบจะแนะนำโจทย์ปัญหาที่เหมาะสมกับผู้เรียนรายบุคคล โดยมอดูล Adaptive Learning นี้จะทำหน้าที่เก็บข้อมูลความซับซ้อนของซอร์สโค้ดที่นักศึกษาทำโจทย์ปัญหาล่าสุด เวลาที่ใช้ในการทำโจทย์ปัญหา คะแนนความถูกต้องตามชุดข้อมูลทดสอบ และทำการส่งค่าเหล่านี้ไปยัง Question Recommendation Module (ภาพที่ 3-1 หมายเลข 3) เพื่อให้ทำการแนะนำโจทย์ปัญหาที่เหมาะสมกับระดับความสามารถของผู้เรียน ณ ขณะนั้น ซึ่งมอดูลนี้ได้มีการออกแบบสำหรับขยายผลต่อในอนาคต สำหรับการพัฒนาปรับปรุงต่อยอดโมเดลที่แนะนำโจทย์ปัญหาที่เหมาะสมกับผู้เรียนที่มีระดับความแม่นยำที่สูงขึ้น งานวิจัยนี้ได้มีการพัฒนาโมเดลสำหรับแนะนำโจทย์ปัญหาโดยวิธีการ Random Forest เพื่อทำนายระดับความยากของคำถามถัดไปที่เหมาะสม ผลลัพธ์จะถูกส่งกลับไปยังส่วนหน้าเพื่อแสดงคำแนะนำให้กับผู้เรียน โดยที่ข้อมูลที่ถูกเลือกนำมาใช้ในการพัฒนาโมเดล Random Forest ประกอบด้วยข้อมูลดังต่อไปนี้

1) Lines of Code (LOC) จำนวนบรรทัดทั้งหมดของโค้ดที่ผู้เรียนส่งเข้ามา

2) Cyclomatic Complexity (CCN) ระดับความซับซ้อนเชิงตรรกะของโค้ด โดยคำนวณจากจำนวนเส้นทางที่เป็นอิสระในโปรแกรม

3) Complexity ป้ายกำกับความยากของโจทย์ที่กำหนดโดยผู้สอน

4) Time Spent ระยะเวลาที่ผู้เรียนใช้ในการทำโจทย์ ตั้งแต่เริ่มต้นจนถึงการส่งคำตอบสุดท้าย

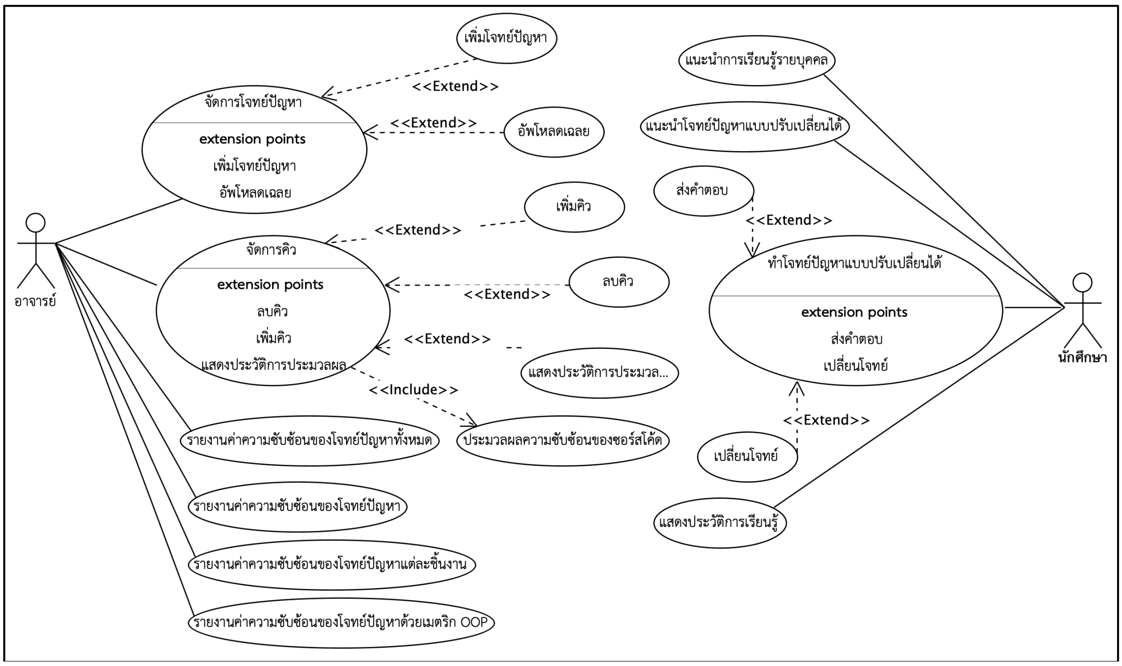
5) Correct อัตราส่วนของจำนวนกรณีทดสอบที่ถูกต้องต่อจำนวนกรณีทดสอบทั้งหมด

6) Attempt จำนวนครั้งที่ผู้เรียนพยายามส่งคำตอบ ซึ่งสะท้อนถึงความพยายามหรือระดับความยากที่ผู้เรียนรับรู้

7) Current Level ระดับปัจจุบันของผู้เรียนในระบบ ซึ่งแม้จะไม่ใช่ตัวชี้วัดโดยตรงของความสามารถ แต่ใช้เป็นตัวแทนของระดับความก้าวหน้าในปัจจุบัน และช่วยให้แบบจำลองสามารถปรับระดับคำถามให้เหมาะสมได้

3.2 แผนภาพยูสเคส (Use Case Diagram)

งานวิจัยนี้พัฒนาต่อยอดแพลตฟอร์มเรียนรู้การเขียนโปรแกรม ส่วนของมอดูลวัดค่าความซับซ้อนของซอร์สโค้ดและมอดูลการเรียนรู้แบบปรับได้ ซึ่งผู้วิจัยจะทำการออกแบบรายละเอียดของแผนภาพยูสเคส เพื่อให้ผู้มีส่วนเกี่ยวข้องสามารถเข้าใจได้โดยง่าย ดังภาพที่ 3-2



ภาพที่ 3-2 แผนภาพยูสเคส

ผู้ใช้งานระบบ คือ อาจารย์และนักศึกษา โดยอาจารย์จะทำการกำหนดคิวสำหรับการประมวลผลซอร์สโค้ด เพื่อกำหนดเวลาสำหรับให้ระบบดึงซอร์สโค้ดไปประมวลผลหาค่าความซับซ้อนของซอร์สโค้ดและจัดเก็บข้อมูลเหล่านั้นในฐานข้อมูล และเมื่อระบบประมวลผลเสร็จสิ้นแล้ว อาจารย์สามารถดูรายงานความซับซ้อนของซอร์สโค้ดที่นักศึกษาเขียนได้ โดยรายงานแสดงทั้งมาตรวัดแบบดั้งเดิมและมาตรวัดเชิงวัตถุ ส่วนนักศึกษาสามารถเห็นโจทย์ปัญหาที่แนะนำแบบปรับเปลี่ยนได้ ซึ่งโจทย์ปัญหาเหล่านี้ได้มาจากการเลือกโจทย์ปัญหาที่มีระดับความยากและนำมาแนะนำให้เหมาะสมกับระดับความสามารถของผู้เรียน และผู้ใช้งานสามารถเลือกโจทย์ปัญหาและสามารถทำโจทย์ปัญหาและส่งซอร์สโค้ดเพื่อตรวจสอบความถูกต้องได้ โดยที่ระยะเวลาการทำโจทย์ปัญหา คะแนนที่ได้

จำนวนครั้งของการส่ง จำนวนบรรทัดของซอร์สโค้ด ค่าความซับซ้อนของซอร์สโค้ดเหล่านี้จะถูกนำไปจัดเก็บในฐานข้อมูลและบ่งบอกถึงระดับความสามารถปัจจุบันและนำไปใช้ในการแนะนำโจทย์ปัญหาแบบปรับเปลี่ยนได้ในครั้งถัดไป

3.3 คำอธิบายยูสเคส (Use Case Description)

1) มอดูลจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด

มอดูลนี้ผู้ที่มีสิทธิ์การใช้งานคือ อาจารย์ โดยอาจารย์สามารถทำการเพิ่มคิว ลบคิว และดูประวัติการประมวลผล เพื่อนำข้อมูลที่อยู่ในคิวไปใช้สำหรับการประมวลผลความซับซ้อน ซึ่งมีรายละเอียดการทำงาน ดังนี้

1.1) ยูสเคสเพิ่มคิว

อาจารย์ต้องทำการเลือกการบ้านที่ต้องการ และกดปุ่มสัญลักษณ์ “เครื่องหมายบวก” เพื่อเพิ่มการบ้านที่ต้องการเข้าไปในคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด ซึ่งมีคำอธิบายรายละเอียดการทำงาน ดังตารางที่ 3-1

ตารางที่ 3-1 คำอธิบายยูสเคสเพิ่มคิว

ชื่อยูสเคส : เพิ่มคิว	รหัส : UC-1	ระดับความสำคัญ : ปานกลาง
ผู้กระทำหลัก : อาจารย์		ระดับความซับซ้อน : ปานกลาง
ผู้มีส่วนเกี่ยวข้อง : อาจารย์		
คำอธิบาย : การเพิ่มคิว ผู้ใช้งานต้องทำการเลือกการบ้านที่ต้องการ จากนั้นกดปุ่มสัญลักษณ์ “เครื่องหมายบวก” เพื่อทำการเพิ่มการบ้านที่ต้องการเข้าไปในคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด		
สิ่งกระตุ้น : ผู้ใช้งานต้องการเพิ่มการบ้านเข้าไปในคิว		
ประเภทของสิ่งกระตุ้น : ภายนอก		
เงื่อนไขก่อนการทำงาน : ผู้ใช้ต้องทำการเข้าสู่ระบบก่อน		
เงื่อนไขหลังการทำงาน : ระบบบันทึกข้อมูลการบ้านที่นำเข้าคิว		
ขั้นตอนการทำงานปกติ :	ผู้ใช้งาน	ระบบ
	1. กดเมนู “การประมวลผลความซับซ้อน”	
		2. แสดงหน้าจอจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด

	3. กรอกข้อมูลปีการศึกษา รายวิชา และกลุ่มเรียน 4. กดปุ่ม “ค้นหาข้อมูล”	 5. แสดงตารางจัดการคิว
	6. เลือกการบ้านที่ต้องการแล้ว กดปุ่มสัญลักษณ์ “เครื่องหมาย บวก”	 7. แสดงหน้าต่างแสดงผลซ้อน สำหรับการเพิ่มคิว
	8. กรอกข้อมูลเวลาที่ต้องการ 9. กดปุ่ม “บันทึก”	 10. ปิดหน้าต่างแสดงผลซ้อน 11. บันทึกข้อมูล 12. แสดงแจ้งเตือนการเพิ่มข้อมูลสำเร็จ
เงื่อนไขการทำงานพิเศษ :	3. ต้องกรอกข้อมูลให้ครบถ้วนตามลำดับ 4. หากกรอกข้อมูไม่ครบถ้วน ระบบจะแสดงข้อความแจ้งเตือนว่า “กรุณากรอกข้อมูลที่จำเป็นให้ครบถ้วน”	

1.2) ยูสเคสลบคิว

อาจารย์ต้องทำการเลือกการบ้านที่ต้องการ และกดปุ่มสัญลักษณ์ “ถังขยะ” เพื่อลบการบ้านที่ต้องการออกจากคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด ซึ่งมีคำอธิบายรายละเอียดการทำงาน ดังตารางที่ 3-2

ตารางที่ 3-2 คำอธิบายยูสเคสลบคิว

ชื่อยูสเคส : ลบคิว	รหัส : UC-2	ระดับความสำคัญ : ปานกลาง
ผู้กระทำหลัก : อาจารย์		ระดับความซับซ้อน : ปานกลาง
ผู้มีส่วนเกี่ยวข้อง : อาจารย์		
คำอธิบาย : การลบคิว ผู้ใช้งานต้องทำการเลือกการบ้านที่ต้องการ จากนั้นกดปุ่มสัญลักษณ์ “ถังขยะ” เพื่อทำการลบการบ้านที่ต้องการออกจากคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด		
สิ่งกระตุ้น : ผู้ใช้งานต้องการลบการบ้านออกจากคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด		

<table>
<tr><td colspan="3">ประเภทของสิ่งกระตุ้น : ภายนอก</td></tr>
<tr><td colspan="3">เงื่อนไขก่อนการทำงาน : ผู้ใช้ต้องทำการเข้าสู่ระบบก่อน</td></tr>
<tr><td colspan="3">เงื่อนไขหลังการทำงาน : ระบบบันทึกข้อมูลการบ้านที่นำออกจากคิว</td></tr>
<tr><td rowspan="2">ขั้นตอนการทำงานปกติ :</td><td>ผู้ใช้งาน</td><td>ระบบ</td></tr>
<tr><td>1. กดเมนู “การประมวลผลความซับซ้อน”

3. กรอกข้อมูลปีการศึกษา รายวิชา และกลุ่มเรียน
4. กดปุ่ม “ค้นหาข้อมูล”

6. เลือกการบ้านที่ต้องการแล้วกดปุ่มสัญลักษณ์ “ถังขยะ”

8. กดปุ่ม “ยืนยัน”</td><td>2. แสดงหน้าจอจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด

5. แสดงตารางจัดการคิว

7. แสดงหน้าต่างแสดงผลซ้อนสำหรับยืนยันการลบข้อมูล

9. ปิดหน้าต่างแสดงผลซ้อน
10. บันทึกข้อมูล
11. แสดงแจ้งเตือนการลบข้อมูลสำเร็จ</td></tr>
<tr><td>เงื่อนไขการทำงานพิเศษ :</td><td colspan="2">3. ต้องกรอกข้อมูลให้ครบถ้วนตามลำดับ
4. หากกรอกข้อมูไม่ครบถ้วน ระบบจะแสดงข้อความแจ้งเตือนว่า “กรุณากรอกข้อมูลที่จำเป็นให้ครบถ้วน”</td></tr>
</table>

1.3) ยูสเคสประมวลผลความซับซ้อนของซอร์สโค้ด

ยูสเคสนี้มีวัตถุประสงค์เพื่อให้ระบบสามารถวัดค่าความซับซ้อนของซอร์สโค้ดที่ต้องการ โดยระบบทำการรันสคริปต์สำหรับการวัดความซับซ้อน ซึ่งมีรายละเอียดดังตารางที่ 3-3

ตารางที่ 3-3 คำอธิบายยูสเคสประมวลผลความซับซ้อนของซอร์สโค้ด

<table>
<tr><td colspan="2">ชื่อยูสเคส : ประมวลผลความซับซ้อนของซอร์สโค้ด</td><td>รหัส : UC-3</td><td>ระดับความสำคัญ : มาก</td></tr>
<tr><td colspan="3">ผู้กระทำหลัก : ระบบ</td><td>ระดับความซับซ้อน : มาก</td></tr>
<tr><td colspan="4">ผู้มีส่วนเกี่ยวข้อง : อาจารย์</td></tr>
<tr><td colspan="4">คำอธิบาย : เมื่อมีการเรียกใช้งานฟังก์ชันวัดค่าความซับซ้อนของซอร์สโค้ด ระบบจะทำการรันสคริปต์สำหรับการวัดความซับซ้อนด้วยมาตรวัดของ CKJM โดยระบบจะทำการประมวลผลความซับซ้อน หากสามารถวัดค่าความซับซ้อนได้ระบบจะนำข้อมูลค่าความซับซ้อนออกมาในรูปแบบไฟล์ XML จากนั้นระบบจะนำข้อมูลที่ได้จากไฟล์ XML เก็บลงตัวแปร และรันสคริปต์สำหรับการวัดความซับซ้อนด้วยมาตรวัดของ Lizard โดยระบบจะทำการประมวลผลความซับซ้อน หากสามารถวัดค่าความซับซ้อนได้ระบบจะนำข้อมูลค่าความซับซ้อนออกมาในรูปแบบไฟล์ XML จากนั้นระบบจะนำข้อมูลที่ได้จากไฟล์ XML เก็บลงตัวแปร</td></tr>
<tr><td colspan="4">สิ่งกระตุ้น : ระบบมีการเรียกใช้งานฟังก์ชันวัดค่าความซับซ้อนของซอร์สโค้ด</td></tr>
<tr><td colspan="4">ประเภทของสิ่งกระตุ้น : ภายใน</td></tr>
<tr><td colspan="4">เงื่อนไขก่อนการทำงาน : มีข้อมูลข้อมูลตำแหน่งของซอร์สโค้ด</td></tr>
<tr><td colspan="4">เงื่อนไขหลังการทำงาน : ระบบได้ข้อมูลค่าความซับซ้อน</td></tr>
<tr><td>ขั้นตอนการทำงานปกติ :</td><td colspan="3">ระบบ</td></tr>
<tr><td></td><td colspan="3">1. ตรวจสอบคิวประมวลผลว่ามีคิวหรือไม่
2. ตั้งเวลาในการวัดค่าความซับซ้อนของซอร์สโค้ด
3. ดึงข้อมูลตำแหน่งที่อยู่ของซอร์สโค้ดที่อยู่ในคิว
4. บันทึกข้อมูลลงตัวแปรสำหรับเก็บข้อมูลข้อมูลตำแหน่งที่อยู่ของซอร์สโค้ด
5. เปลี่ยนสถานะของคิวที่ทำการดึงข้อมูลให้เป็น “Processing”
6. ส่งข้อมูลที่อยู่ของซอร์สโค้ดที่อยู่ในคิวประมวลผลไปยังฟังก์ชันวัดค่าความซับซ้อนของซอร์สโค้ด
7. รันสคริปต์สำหรับการวัดความซับซ้อนด้วยมาตรวัดของ CKJM
8. ทำการประมวลผลความซับซ้อน
9. นำข้อมูลที่ได้จากไฟล์ XML เก็บลงตัวแปร
10. รันสคริปต์สำหรับการวัดความซับซ้อนด้วยมาตรวัดของ Lizard
11. ทำการประมวลผลความซับซ้อน</td></tr>
</table>

	12. บันทึกข้อมูลลงฐานข้อมูล
เงื่อนไขการทำงานพิเศษ :	1. หากไม่มีคิวประมวลผลระบบจะไม่ดึงข้อมูลตำแหน่งที่อยู่ของซอร์สโค้ด 7. หากไม่สามารถวัดค่าความซับซ้อนได้ ระบบจะเปลี่ยนสถานะของคิวที่ทำการดึงข้อมูลให้เป็น “Error”

1.4) ยูสเคสแสดงประวัติการประมวลผล

อาจารย์สามารถทราบประวัติการประมวลผลของชุดการบ้านที่ถูกประมวลผลไปแล้วได้ เพื่อช่วยให้อาจารย์สามารถทราบได้ว่าชุดการบ้านใดที่ถูกประมวลผลไปแล้วบ้าง ซึ่งมีคำอธิบายรายละเอียดการทำงาน ดังตารางที่ 3-4

ตารางที่ 3-4 คำอธิบายยูสเคสแสดงประวัติการประมวลผล

<table>
<tr><td>ชื่อยูสเคส : แสดงประวัติการประมวลผล</td><td>รหัส : UC0-4</td><td colspan="2">ระดับความสำคัญ : ปานกลาง</td></tr>
<tr><td colspan="2">ผู้กระทำหลัก : อาจารย์</td><td colspan="2">ระดับความซับซ้อน : น้อย</td></tr>
<tr><td colspan="4">ผู้มีส่วนเกี่ยวข้อง : อาจารย์</td></tr>
<tr><td colspan="4">คำอธิบาย : ผู้ใช้งานสามารถทราบประวัติการประมวลผลของชุดการบ้านที่ถูกประมวลผลไปแล้วได้ เพื่อช่วยให้ผู้ใช้งานสามารถทราบได้ว่าชุดการบ้านใดที่ถูกประมวลผลไปแล้วบ้าง</td></tr>
<tr><td colspan="4">สิ่งกระตุ้น : ผู้ใช้งานต้องการทราบว่าชุดการบ้านไหนที่ถูกประมวลผลไปแล้วบ้าง</td></tr>
<tr><td colspan="4">ประเภทของสิ่งกระตุ้น : ภายนอก</td></tr>
<tr><td colspan="4">เงื่อนไขก่อนการทำงาน : ผู้ใช้ต้องทำการเข้าสู่ระบบก่อน</td></tr>
<tr><td colspan="4">เงื่อนไขหลังการทำงาน : ผู้ใช้ทราบได้ว่าชุดการบ้านใดถูกประมวลผลไปแล้วบ้าง</td></tr>
<tr><td rowspan="2">ขั้นตอนการทำงานปกติ :</td><td colspan="2">ผู้ใช้งาน</td><td>ระบบ</td></tr>
<tr><td colspan="2">1. กดเมนู “การประมวลผลความซับซ้อน”

3. กรอกข้อมูลปีการศึกษา รายวิชา และกลุ่มเรียน
4. กดปุ่ม “ค้นหาข้อมูล”</td><td>2. แสดงหน้าจอจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด</td></tr>
</table>

		5. แสดงตารางประวัติการประมวลผลความซับซ้อนของซอร์สโค้ด
เงื่อนไขการทำงานพิเศษ :	3. ต้องกรอกข้อมูลให้ครบถ้วนตามลำดับ 4. หากกรอกข้อมูไม่ครบถ้วน ระบบจะแสดงข้อความแจ้งเตือนว่า “กรุณากรอกข้อมูลที่จำเป็นให้ครบถ้วน”	

1.5) ยูสเคสเพิ่มโจทย์ปัญหา

อาจารย์จะสามารถเพิ่มข้อมูลของโจทย์ปัญหาได้โดยการกรอกข้อมูล ชื่อโจทย์ รายละเอียดโจทย์ แท็ก สถานะการใช้งาน และคำตอบของโจทย์ปัญหา ระดับความยากของโจทย์ปัญหา และไฟล์ซอร์สโค้ดเฉลยของโจทย์ปัญหาซึ่งจะมีรายละเอียดคำอธิบายยูสเคสดังตารางที่ 3-5

ตารางที่ 3-5 คำอธิบายยูสเคสเพิ่มโจทย์ปัญหา

ชื่อยูสเคส : เพิ่มโจทย์ปัญหา	รหัส : UC0-5	ระดับความสำคัญ : ปานกลาง
ผู้กระทำหลัก : อาจารย์		ระดับความซับซ้อน : ปานกลาง
ผู้มีส่วนเกี่ยวข้อง : อาจารย์		
คำอธิบาย : ในการเพิ่มโจทย์ปัญหา อาจารย์จำเป็นต้องกรอกข้อมูลชื่อโจทย์ รายละเอียดโจทย์ และสถานะการใช้งานให้ครบถ้วน โดยที่รูปภาพประกอบ แท็ก และคำตอบของโจทย์ปัญหา สามารถเพิ่มหรือไม่ก็ได้ ซึ่งสถานะการใช้งานได้ถูกกำหนดค่าเริ่มต้นเป็น “ใช้งาน”		
สิ่งกระตุ้น : อาจารย์ต้องการเพิ่มโจทย์ปัญหา		
ประเภทสิ่งกระตุ้น : ภายนอก		
เงื่อนไขก่อนการทำงาน : ต้องมีอาจารย์อย่างน้อย 1 คน		
เงื่อนไขหลังการทำงาน : ข้อมูลโจทย์ปัญหาถูกบันทึก		
ขั้นตอนการทำงานปกติ :	ผู้ดูแลระบบ	ระบบ
	1. เลือกเมนู “คลังโจทย์ปัญหา” 3. คลิกเมนู “จัดการโจทย์ปัญหา” 5. คลิกปุ่ม “เพิ่มโจทย์ปัญหา”	2. แสดงเมนูย่อย 4. แสดงหน้าจอจัดการโจทย์ปัญหา 6. แสดงหน้าจอเพิ่มโจทย์ปัญหา

	7. กรอกข้อมูลโจทย์ปัญหา 8. เลือกแท็ก 9. เลือกระดับความซับซ้อนของโจทย์ปัญหา 10. คลิกแท็บ “สร้างชุดข้อมูลทดสอบ”	
		11. แสดงหน้าจอเพิ่มชุดข้อมูลทดสอบ
	12. กรอกข้อมูลทดสอบ 13. คลิกปุ่ม “เพิ่ม”	
		14. แสดงชุดข้อมูลทดสอบในตาราง
	15. คลิกปุ่ม “บันทึก”	
		16. บันทึกข้อมูล 17. แสดงแจ้งเตือน “บันทึกข้อมูลเสร็จสิ้น”
	18. คลิกแท็บ “สร้างเฉลย” 19. คลิกเลือกภาษา	
		20. แสดงกล่องโต้ตอบเพิ่มเฉลย
	21. คลิกปุ่ม “อัปโหลดไฟล์” 22. เลือกไฟล์อัปโหลด 23. คลิกปุ่ม “บันทึก”	
		24. แสดงแจ้งเตือน “อัปโหลดไฟล์สำเร็จ” 25. บันทึกข้อมูลไฟล์
เงื่อนไขการทำงานพิเศษ :	9. เลือกระดับความซับซ้อนของโจทย์ปัญหา หากไม่เลือกจะถูกตั้งค่าเป็นระดับง่าย	

1.6) ยูสเคสรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด

ฟังก์ชันนี้เป็นส่วนที่ผู้ใช้งานสามารถดูรายงานค่าความซับซ้อนของโจทย์ปัญหาในแต่ละรายวิชาได้ โดยผู้ใช้งานจะต้องเลือกเมนูรายงาน ระบบจะแสดงเมนูย่อย และเลือกเมนูรายงานค่าความซับซ้อนด้วยเมตริก Cyclomatic ระบบจะแสดงเมนูย่อย จากนั้นผู้ใช้งานสามารถเลือกเมนูรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด ซึ่งผู้ใช้งานสามารถเลือกปีการศึกษา รายวิชา และกลุ่มเรียนที่ต้องการค้นหาได้ จากนั้นกดปุ่ม “ค้นหารายงาน” เพื่อค้นหารายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด ระบบจะแสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหาต่อค่าเฉลี่ยจำนวน Line of Code และตารางรายงานค่าความซับซ้อนของโจทย์ปัญหาที่ค้นหา ซึ่งมีลำดับการทำงาน ดังตารางที่ 3-6

ตารางที่ 3-6 คำอธิบายยูสเคสรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด

<table>
<tr><td>ชื่อยูสเคส : รายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด</td><td>รหัส : UC-6</td><td>ระดับความสำคัญ : ปานกลาง</td></tr>
<tr><td colspan="2">ผู้กระทำหลัก : อาจารย์</td><td>ระดับความซับซ้อน : ปานกลาง</td></tr>
<tr><td colspan="3">ผู้มีส่วนเกี่ยวข้อง : อาจารย์</td></tr>
<tr><td colspan="3">คำอธิบาย : ผู้ใช้งานสามารถทำได้โดยการกดเมนู “รายงาน” ระบบจะแสดงเมนูย่อย และเลือกเมนูรายงานค่าความซับซ้อนด้วยเมตริก Cyclomatic ระบบจะแสดงเมนูย่อย จากนั้นกดเมนู “รายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด” ระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด ซึ่งผู้ใช้งานสามารถเลือกปีการศึกษา รายวิชา และกลุ่มเรียนได้ จากนั้นกดปุ่ม “ค้นหารายงาน” เพื่อค้นหารายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด ระบบจะแสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหาต่อค่าเฉลี่ยจำนวน Line of Code และตารางรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมดที่ค้นหา</td></tr>
<tr><td colspan="3">สิ่งกระตุ้น : อาจารย์ต้องการดูรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด</td></tr>
<tr><td colspan="3">ประเภทของสิ่งกระตุ้น : ภายนอก</td></tr>
<tr><td colspan="3">เงื่อนไขก่อนการทำงาน : มีข้อมูลค่าความซับซ้อนของโจทย์ปัญหาที่วัดโดยใช้มาตรวัด Cyclomatic</td></tr>
<tr><td colspan="3">เงื่อนไขหลังการทำงาน : ระบบแสดงรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด</td></tr>
<tr><td>ขั้นตอนการทำงานปกติ :</td><td>ผู้ใช้งาน</td><td>ระบบ</td></tr>
<tr><td></td><td>1. กดเมนู “รายงาน”</td><td>2. แสดงเมนูย่อย</td></tr>
</table>

	3. กดเมนู “รายงานค่าความซับซ้อนด้วยเมตริก Cyclomatic” 5. กดเมนู “รายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด” 7. เลือกปีการศึกษา 9. เลือกรายวิชา 11. เลือกกลุ่มเรียน 12. กดปุ่ม “ค้นหารายงาน”	4. แสดงเมนูย่อย 6. แสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด 8. แสดงรายวิชา 10. แสดงกลุ่มเรียน 13. แสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหาต่อค่าเฉลี่ยจำนวน Line of Code 14. แสดงตารางรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด
เงื่อนไขการทำงานพิเศษ :	12. หากกรอกข้อมูลไม่ครบ ระบบจะแสดงแจ้งเตือนว่า “กรุณากรอกข้อมูลที่จำเป็นให้ครบถ้วน” 12. หากระบบไม่พบข้อมูลที่ค้นหา ระบบจะแสดงแจ้งเตือนว่า “ไม่พบข้อมูลที่ค้นหา”	

1.7) ยูสเคสรายงานค่าความซับซ้อนของโจทย์ปัญหา

ยูสเคสนี้เป็นส่วนที่ผู้ใช้งานสามารถดูรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมดของแต่ละการบ้านได้ โดยผู้ใช้งานจะต้องเลือกเมนูรายงาน ระบบจะแสดงเมนูย่อย และเลือกเมนูรายงานค่าความซับซ้อนด้วยเมตริก Cyclomatic ระบบจะแสดงเมนูย่อย จากนั้นผู้ใช้งานสามารถเลือกเมนูรายงานค่าความซับซ้อนของโจทย์ปัญหา ซึ่งผู้ใช้งานสามารถเลือกปีการศึกษา รายวิชา กลุ่มเรียน และการบ้านที่ต้องการค้นหาได้ จากนั้นกดปุ่ม “ค้นหารายงาน” เพื่อค้นหารายงานค่าความ

ซับซ้อนของโจทย์ปัญหา ระบบจะแสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหาต่อค่าเฉลี่ยจำนวน Line of Code และตารางรายงานค่าความซับซ้อนของโจทย์ปัญหาที่ค้นหา ซึ่งมีลำดับการทำงานแสดงดังตารางที่ 3-7

ตารางที่ 3-7 คำอธิบายยูสเคสรายงานค่าความซับซ้อนของโจทย์ปัญหา

ชื่อยูสเคส : รายงานค่าความซับซ้อนของโจทย์ปัญหา	รหัส : UC-7	ระดับความสำคัญ : ปานกลาง
ผู้กระทำหลัก : อาจารย์		ระดับความซับซ้อน : ปานกลาง
ผู้มีส่วนเกี่ยวข้อง : อาจารย์		
คำอธิบาย : ผู้ใช้งานสามารถทำได้โดยการกดเมนู “รายงาน” ระบบจะแสดงเมนูย่อย และเลือกเมนูรายงานค่าความซับซ้อนด้วยเมตริก Cyclomatic ระบบจะแสดงเมนูย่อย จากนั้นกดเมนู “รายงานค่าความซับซ้อนของโจทย์ปัญหา” ระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหา ซึ่งผู้ใช้งานสามารถเลือกปีการศึกษา รายวิชา กลุ่มเรียน และการบ้านได้ จากนั้นกดปุ่ม “ค้นหารายงาน” เพื่อค้นหารายงานค่าความซับซ้อนของโจทย์ปัญหา ระบบจะแสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหาต่อค่าเฉลี่ยจำนวน Line of Code และตารางรายงานค่าความซับซ้อนของโจทย์ปัญหาที่ค้นหา		
สิ่งกระตุ้น : อาจารย์ต้องการดูรายงานค่าความซับซ้อนของโจทย์ปัญหา		
ประเภทของสิ่งกระตุ้น : ภายนอก		
เงื่อนไขก่อนการทำงาน : มีข้อมูลค่าความซับซ้อนของโจทย์ปัญหาที่วัดโดยใช้มาตรวัด Cyclomatic		
เงื่อนไขหลังการทำงาน : ระบบแสดงรายงานค่าความซับซ้อนของโจทย์ปัญหา		

ขั้นตอนการทำงานปกติ :	ผู้ใช้งาน	ระบบ
	1. กดเมนู “รายงาน”	
		2. แสดงเมนูย่อย
	3. กดเมนู “รายงานค่าความซับซ้อนด้วยเมตริก Cyclomatic”	
		4. แสดงเมนูย่อย
	5. กดเมนู “รายงานค่าความซับซ้อนของโจทย์ปัญหา”	
		6. แสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหา
	7. เลือกปีการศึกษา	

	9. เลือกรายวิชา 11. เลือกกลุ่มเรียน 13. เลือกการบ้าน 14. กดปุ่ม "ค้นหารายงาน"	8. แสดงรายวิชา 10. แสดงกลุ่มเรียน 12. แสดงการบ้าน 15. แสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหาต่อค่าเฉลี่ยจำนวน Line of Code 16. แสดงตารางรายงานค่าความซับซ้อนของโจทย์ปัญหา
เงื่อนไขการทำงานพิเศษ :	14. หากกรอกข้อมูลไม่ครบ ระบบจะแสดงแจ้งเตือนว่า "กรุณากรอกข้อมูลที่จำเป็นให้ครบถ้วน" 14. หากระบบไม่พบข้อมูลที่ค้นหา ระบบจะแสดงแจ้งเตือนว่า "ไม่พบข้อมูลที่ค้นหา"	

1.8) ยูสเคสรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน

ยูสเคสนี้เป็นส่วนที่ผู้ใช้งานสามารถดูรายงานค่าความซับซ้อนของโจทย์ปัญหาได้ โดยผู้ใช้งานจะต้องเลือกเมนูรายงาน ระบบจะแสดงเมนูย่อย และเลือกเมนูรายงานค่าความซับซ้อนด้วยเมตริก Cyclomatic ระบบจะแสดงเมนูย่อย จากนั้นผู้ใช้งานสามารถเมนูรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงานได้ ระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน ซึ่งผู้ใช้งานสามารถเลือกปีการศึกษา รายวิชา กลุ่มเรียน การบ้าน และโจทย์ปัญหาที่ต้องการค้นหาได้ จากนั้นกดปุ่ม "ค้นหารายงาน" เพื่อค้นหารายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน ระบบจะแสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหาต่อค่าเฉลี่ยจำนวน Line of Code ของนักศึกษาทั้งหมดและตารางรายงานค่าความซับซ้อนของโจทย์ปัญหาที่ค้นหา ซึ่งมีลำดับการทำงาน ดังตารางที่ 3-8

ตารางที่ 3-8 คำอธิบายยูสเคสรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน

<table>
<tr><td colspan="2">ชื่อยูสเคส : รายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน</td><td>รหัส : UC-8</td><td>ระดับความสำคัญ : ปานกลาง</td></tr>
<tr><td colspan="3">ผู้กระทำหลัก : อาจารย์</td><td>ระดับความซับซ้อน : ปานกลาง</td></tr>
<tr><td colspan="4">ผู้มีส่วนเกี่ยวข้อง : อาจารย์</td></tr>
<tr><td colspan="4">คำอธิบาย : ผู้ใช้งานสามารถทำได้โดยการกดเมนู “รายงาน” ระบบจะแสดงเมนูย่อย และเลือกเมนูรายงานค่าความซับซ้อนด้วยเมตริก Cyclomatic ระบบจะแสดงเมนูย่อย จากนั้นกดเมนู “รายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน” ระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน ซึ่งผู้ใช้งานสามารถเลือกปีการศึกษา รายวิชา กลุ่มเรียน การบ้าน และโจทย์ปัญหาได้ จากนั้นกดปุ่ม “ค้นหารายงาน” เพื่อค้นหารายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน ระบบจะแสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหาต่อค่าเฉลี่ยจำนวน Line of Code ของนักศึกษาทั้งหมด และตารางรายงานค่าความซับซ้อนของโจทย์ปัญหาที่ค้นหา</td></tr>
<tr><td colspan="4">สิ่งกระตุ้น : อาจารย์ต้องการดูรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน</td></tr>
<tr><td colspan="4">ประเภทของสิ่งกระตุ้น : ภายนอก</td></tr>
<tr><td colspan="4">เงื่อนไขก่อนการทำงาน : มีข้อมูลค่าความซับซ้อนของโจทย์ปัญหาที่วัดโดยใช้มาตรวัด Cyclomatic</td></tr>
<tr><td colspan="4">เงื่อนไขหลังการทำงาน : ระบบแสดงรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน</td></tr>
<tr><td>ขั้นตอนการทำงานปกติ :</td><td colspan="2">ผู้ใช้งาน</td><td>ระบบ</td></tr>
<tr><td></td><td colspan="2">1. กดเมนู “รายงาน”

3. กดเมนู “รายงานค่าความซับซ้อนด้วยเมตริก Cyclomatic”

5. กดเมนู “รายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน”

7. เลือกปีการศึกษา</td><td>2. แสดงเมนูย่อย

4. แสดงเมนูย่อย

6. แสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน

8. แสดงรายวิชา</td></tr>
</table>

	9. เลือกรายวิชา 11. เลือกกลุ่มเรียน 13. เลือกการบ้าน 15. เลือกโจทย์ปัญหา 16. กดปุ่ม “ค้นหารายงาน”	10. แสดงกลุ่มเรียน 12. แสดงการบ้าน 14. แสดงโจทย์ปัญหา 17. แสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหาต่อค่าเฉลี่ยจำนวน Line of Code ของนักศึกษาทั้งหมด 18. แสดงตารางรายงานค่าความซับซ้อนของโจทย์ปัญหา
เงื่อนไขการทำงานพิเศษ :	16. หากกรอกข้อมูลไม่ครบ ระบบจะแสดงแจ้งเตือนว่า “กรุณากรอกข้อมูลที่จำเป็นให้ครบถ้วน” 16. หากระบบไม่พบข้อมูลที่ค้นหา ระบบจะแสดงแจ้งเตือนว่า “ไม่พบข้อมูลที่ค้นหา”	

1.9) ยูสเคสรายงานค่าความซับซ้อนด้วยเมตริกเชิงวัตถุ (OOP)

ยูสเคสนี้เป็นส่วนที่ผู้ใช้งานสามารถดูรายงานค่าความซับซ้อนด้วยเมตริกเชิงวัตถุ (OOP)ของนักศึกษาแต่ละคนได้ โดยผู้ใช้งานจะต้องเลือกเมนูรายงาน ระบบจะแสดงเมนูย่อย จากนั้นผู้ใช้งานจะสามารถเลือกเมนูรายงานค่าความซับซ้อนด้วยเมตริก OOP ระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนด้วยเมตริก OOP ซึ่งผู้ใช้งานสามารถเลือกปีการศึกษา รายวิชา กลุ่มเรียน การบ้าน และโจทย์ปัญหาที่ต้องการค้นหา จากนั้นกดปุ่ม “ค้นหารายงาน” เพื่อค้นหารายงานค่าความซับซ้อนด้วยเมตริก OOP ระบบจะแสดงตารางรายงานค่าความซับซ้อนด้วยเมตริก OOP ที่ค้นหา ผู้ใช้งานสามารถเลือกนักศึกษาที่ต้องการดูรายงานค่าความซับซ้อนของโจทย์ปัญหาได้ ระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาของนักศึกษาที่เลือก ซึ่งมีลำดับการทำงานดังตารางที่ 3-9

ตารางที่ 3-9 คำอธิบายยูสเคสรายงานค่าความซับซ้อนด้วยเมตริกเชิงวัตถุ

<table>
<tr><td colspan="2">ชื่อยูสเคส : แสดงผลรายงานค่าความซับซ้อนของโจทย์ปัญหาด้วยเมตริกเชิงวัตถุ</td><td>รหัส : UC-9</td><td colspan="2">ระดับความสำคัญ : ปานกลาง</td></tr>
<tr><td colspan="3">ผู้กระทำหลัก : อาจารย์</td><td colspan="2">ระดับความซับซ้อน : ปานกลาง</td></tr>
<tr><td colspan="5">ผู้มีส่วนเกี่ยวข้อง : อาจารย์</td></tr>
<tr><td colspan="5">คำอธิบาย : ผู้ใช้งานสามารถทำได้โดยการกดเมนู “รายงาน” ระบบจะแสดงเมนูย่อย จากนั้นกดเมนู “รายงานค่าความซับซ้อนด้วยเมตริก OOP” ระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนด้วยเมตริก OOP ซึ่งผู้ใช้งานสามารถเลือกปีการศึกษา รายวิชา กลุ่มเรียน การบ้าน และโจทย์ปัญหาได้ จากนั้นกดปุ่ม “ค้นหารายงาน” เพื่อค้นหารายงานค่าความซับซ้อนด้วยเมตริก OOP ระบบจะแสดงตารางรายงานค่าความซับซ้อนด้วยเมตริก OOP ที่ค้นหา ผู้ใช้งานสามารถเลือกนักศึกษาที่ต้องการดูรายงานค่าความซับซ้อนของโจทย์ปัญหาได้ ระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาของนักศึกษาที่เลือก</td></tr>
<tr><td colspan="5">สิ่งกระตุ้น : อาจารย์ต้องการดูรายงานค่าความซับซ้อนของโจทย์ปัญหาของนักศึกษาแต่ละคน</td></tr>
<tr><td colspan="5">ประเภทของสิ่งกระตุ้น : ภายนอก</td></tr>
<tr><td colspan="5">เงื่อนไขก่อนการทำงาน : มีข้อมูลค่าความซับซ้อนของโจทย์ปัญหาที่วัดโดยใช้มาตรวัด OOP</td></tr>
<tr><td colspan="5">เงื่อนไขหลังการทำงาน : ระบบแสดงรายงานค่าความซับซ้อนด้วยเมตริก OOP</td></tr>
<tr><td>ขั้นตอนการทำงานปกติ :</td><td colspan="3">ผู้ใช้งาน</td><td>ระบบ</td></tr>
<tr><td></td><td colspan="3">1. กดเมนู “รายงาน”</td><td></td></tr>
<tr><td></td><td colspan="3"></td><td>2. แสดงเมนูย่อย</td></tr>
<tr><td></td><td colspan="3">3. กดเมนู “รายงานค่าความซับซ้อนด้วยเมตริก OOP”</td><td></td></tr>
<tr><td></td><td colspan="3"></td><td>4. แสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหารายบุคคล</td></tr>
<tr><td></td><td colspan="3">5. เลือกปีการศึกษา</td><td></td></tr>
<tr><td></td><td colspan="3"></td><td>6. แสดงรายวิชา</td></tr>
<tr><td></td><td colspan="3">7. เลือกรายวิชา</td><td></td></tr>
<tr><td></td><td colspan="3"></td><td>8. แสดงกลุ่มเรียน</td></tr>
<tr><td></td><td colspan="3">9. เลือกกลุ่มเรียน</td><td></td></tr>
<tr><td></td><td colspan="3"></td><td>10. แสดงการบ้าน</td></tr>
</table>

	11. เลือกการบ้าน 13. เลือกโจทย์ปัญหา 14. กดปุ่ม “ค้นหารายงาน” 16. เลือกนักศึกษาที่ต้องการดูรายงานค่าความซับซ้อนของโจทย์ปัญหา	 12. แสดงโจทย์ปัญหา 15. แสดงตารางรายงานค่าความซับซ้อนด้วยเมตริก OOP 17. แสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาของนักศึกษาที่เลือก
เงื่อนไขการทำงานพิเศษ :	14. หากกรอกข้อมูลไม่ครบ ระบบจะแสดงแจ้งเตือนว่า “กรุณากรอกข้อมูลที่จำเป็นให้ครบถ้วน” 14. หากระบบไม่พบข้อมูลที่ค้นหา ระบบจะแสดงแจ้งเตือนว่า “ไม่พบข้อมูลที่ค้นหา”	

2) มอดูลการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้

นักศึกษาสามารถดำเนินการทำโจทย์การเรียนรู้แบบปรับได้โดยการพัฒนาและส่งไฟล์ซอร์สโค้ด (Source code) เพื่อให้ระบบทำการประมวลผลและตรวจสอบความถูกต้องของโค้ดที่เขียนขึ้น โดยกระบวนการนี้จะเกี่ยวข้องกับการคอมไพล์ซอร์สโค้ด หลังจากนั้นระบบจะทำการรันโค้ดที่ถูกคอมไพล์แล้ว พร้อมตรวจสอบผลลัพธ์โดยเปรียบเทียบกับชุดข้อมูลทดสอบ (Test set) เพื่อวิเคราะห์ความถูกต้องและประสิทธิภาพของโค้ดที่นักศึกษาส่งมาทั้งนี้ รายละเอียดเพิ่มเติมเกี่ยวกับกระบวนการดำเนินงาน การส่งไฟล์ การคอมไพล์โค้ด ตลอดจนวิธีการตรวจสอบผลลัพธ์ของชุดทดสอบ

2.1) ยูสเคสแนะนำโจทย์ปัญหาแบบปรับได้

นักศึกษาสามารถเลือกโจทย์ปัญหาจากแถบแนะนำการเรียนรู้ได้โดยจะแสดงรายการโจทย์ปัญหาที่มีความซับซ้อนระดับปานกลางให้นักศึกษาเลือกกรณีที่นักศึกษาเข้ามาในครั้งแรก แสดงรายละเอียดของข้อมูลการทำโจทย์ปัจจุบันกรณีที่นักศึกษาเคยเลือกทำโจทย์ปัญหาแล้ว และแสดงโจทย์ปัญหาถัดไปที่เหมาะสมให้นักศึกษาเลือกกรณีที่นักศึกษาทำโจทย์ปัญหาสำเร็จแสดงตารางที่ 3-10

ตารางที่ 3-10 คำอธิบายยูสเคสแนะนำโจทย์ปัญหาแบบปรับได้

ชื่อยูสเคส : แนะนำโจทย์ปัญหาแบบปรับได้	รหัส : UC-10	ระดับความสำคัญ : ปานกลาง
ผู้กระทำหลัก : นักศึกษา		ระดับความซับซ้อน : ปานกลาง
ผู้มีส่วนเกี่ยวข้อง : นักศึกษา		
คำอธิบาย : นักศึกษาสามารถเข้าชมสามารถเข้าชมหน้าจอแนะนำการเรียนรู้รายบุคคลระบบจะแสดงรายละเอียดของข้อมูลการทำโจทย์ปัจจุบัน โดยจะแสดงรายการโจทย์ปัญหาที่มีความซับซ้อนระดับปานกลางให้นักศึกษาเลือกกรณีที่นักศึกษาเข้ามาในครั้งแรก		
สิ่งกระตุ้น : นักศึกษาต้องการดูหน้าจอแนะนำการเรียนรู้รายบุคคล		
ประเภทสิ่งกระตุ้น : ภายนอก		
เงื่อนไขก่อนการทำงาน : -		
เงื่อนไขหลังการทำงาน : แสดงโจทย์ปัญหาที่เหมาะสมกับนักศึกษา		
ขั้นตอนการทำงานปกติ :	ผู้ดูแลระบบ	ระบบ
	1. เลือกเมนู “แนะนำการเรียนรู้รายบุคคล”	
		2. แสดงหน้าจอแนะนำการเรียนรู้รายบุคคลส่วนของแนะนำการเรียนเรียนรู้รายบุคคล
	3. เลือกโจทย์ปัญหา	
		4. แสดงหน้าจอทำโจทย์ปัญหาการเรียนรู้แบบปรับได้
เงื่อนไขการทำงานพิเศษ :	2. กรณีนักศึกษายังไม่เคยเลือกโจทย์ปัญหา จะแสดงโจทย์ปัญหาระดับปานกลางให้นักศึกษาเลือก กรณีที่นักศึกษาเคยเลือกทำโจทย์ปัญหาแล้วแต่ยังไม่เสร็จจะแสดงรายการโจทย์ปัญหาที่ยังทำไม่เสร็จ และกรณีที่นักศึกษาทำโจทย์ปัญหาสำเร็จแล้วจะแสดงรายการโจทย์ปัญหาถัดไปที่เหมาะสมให้นักศึกษาเลือกเพื่อทำโจทย์ปัญหาต่อไป	

2.2) ยูสเคสทำโจทย์ปัญหาการเรียนรู้แบบปรับได้

นักศึกษาสามารทำโจทย์ปัญหาการเรียนรู้แบบปรับได้ โดยเลือกจากรายการโจทย์ปัญหาที่ระบบแนะนำตามระดับความสามารถของนักศึกษา ซึ่งรายละเอียดการทำงานของยูสเคสแสดงดังตารางที่ 3-11

ตารางที่ 3-11 คำอธิบายยูสเคสการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้

ชื่อยูสเคส : ทำโจทย์ประเมินตามความรู้ของนักศึกษา	รหัส : UC-11	ระดับความสำคัญ : ปานกลาง
ผู้กระทำหลัก : นักศึกษา		ระดับความซับซ้อน : ปานกลาง
ผู้มีส่วนเกี่ยวข้อง : นักศึกษา		
คำอธิบาย : ในการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้สามารถทำได้โดยการส่งไฟล์ซอร์สโค้ด (Source code) และ Compile เพื่อดูผลลัพธ์ Test set ของการทำงานของซอร์สโค้ด หากผลลัพธ์ของ Test set ถูกต้องทั้งหมดจะสามารถไปยังโจทย์ปัญหาถัดไปได้		
สิ่งกระตุ้น : นักศึกษาต้องการทำโจทย์ปัญหาตามความรู้ของนักศึกษา		
ประเภทสิ่งกระตุ้น : ภายนอก		
เงื่อนไขก่อนการทำงาน : ต้องมีโจทย์ปัญหา		
เงื่อนไขหลังการทำงาน : ข้อมูลการทำโจทย์ปัญหาถูกบันทึก		
ขั้นตอนการทำงานปกติ :	ผู้ดูแลระบบ	ระบบ
	1. เลือกเมนู “แนะนำการเรียนรู้รายบุคคล”	
		2. แสดงหน้าจอแนะนำการเรียนรู้รายบุคคล
	3. คลิกเลือกโจทย์ปัญหา	
		4. แสดงหน้าจอการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้
	5. เลือกภาษาของซอร์สโค้ด (Source code)	
	6. อัปโหลดหรือเขียนซอร์สโค้ด (Source code)	
	7. กรอกข้อมูลนำเข้า	
	8. คลิกปุ่ม “Run Code”	9. แสดงผลลัพท์การ Compile และแสดงรายการผล Test Case
		10. แสดงกล่องแจ้งเตือนแก้โจทย์ปัญหานี้สำเร็จแล้ว
		12. บันทึกข้อมูลการทำโจทย์ปัญหา
	11. คลิกปุ่ม “ไปโจทย์ปัญหาถัดไป”	

		13. แสดงหน้าจอแนะนำการเรียนรู้รายบุคคล”
เงื่อนไขการทำงานพิเศษ :	2. กรณีไม่เคยทำโจทย์ปัญหาตามความสามารถมาก่อนจะสุ่มแสดงโจทย์ระดับปานกลางมาให้นักศึกษาเลือกทั้งหมด 5 ข้อ 2. กรณีนักศึกษาเคยทำโจทย์ปัญหาสำเร็จแล้วจะแสดงรายการโจทย์ปัญหาถัดไปที่เหมาะสม 5 ข้อให้นักศึกษาเลือก 6. ภาษาจะถูกเลือกค่าเริ่มต้นไว้เป็นภาษา C 10. กรณี Compile แล้ว Test Case ไม่ผ่านทุก Case จะไม่แสดงกล่องแจ้งเตือนแก้โจทย์ปัญหานี้สำเร็จแล้ว 10. กรณี Compile แล้ว Test Case ไม่ผ่านทุก Case จะไม่แสดงปุ่มไปโจทย์ปัญหาถัดไป แต่จะแสดงปุ่มเปลี่ยนโจทย์ปัญหาแทน	

2.3) ยูสเคสการเปลี่ยนโจทย์ปัญหาการเรียนรู้แบบปรับได้

ในการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้หากนักศึกษาพบว่าไม่สามารถทำโจทย์ที่ได้รับมอบหมายได้หรือโจทย์ดังกล่าวไม่เหมาะสมกับระดับความรู้และความสามารถของตน นักศึกษาสามารถเปลี่ยนโจทย์ประเมินได้ โดยจะต้องเป็นไปตามเงื่อนไขหรือข้อกำหนดที่กำหนดไว้ดังรายละเอียดเกี่ยวกับกรณีการเปลี่ยนโจทย์แสดงดังตารางที่ 3-12

ตารางที่ 3-12 คำอธิบายยูสเคสการเปลี่ยนโจทย์ปัญหาการเรียนรู้แบบปรับได้

ชื่อยูสเคส : เปลี่ยนโจทย์ปัญหาการเรียนรู้แบบปรับได้	รหัส : UC-12	ระดับความสำคัญ : ปานกลาง
ผู้กระทำหลัก : นักศึกษา		ระดับความซับซ้อน : ปานกลาง
ผู้มีส่วนเกี่ยวข้อง : นักศึกษา		
คำอธิบาย : ในการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้ หากนักศึกษาพบว่าไม่สามารถทำโจทย์ที่ได้รับมอบหมายได้หรือโจทย์ดังกล่าวไม่เหมาะสมกับระดับความรู้และความสามารถของตน นักศึกษาสามารถเปลี่ยนโจทย์ได้ในหน้าจอการทำโจทย์การเรียนรู้แบบปรับได้		
สิ่งกระตุ้น : นักศึกษาต้องการเปลี่ยนโจทย์ปัญหา		
ประเภทสิ่งกระตุ้น : ภายนอก		
เงื่อนไขก่อนการทำงาน : นักศึกษาต้องเคยเลือกทำโจทย์ปัญหานั้นก่อน		
เงื่อนไขหลังการทำงาน : ข้อมูลการเปลี่ยนโจทย์ปัญหาถูกบันทึก		
ขั้นตอนการทำงานปกติ :	ผู้ดูแลระบบ	ระบบ
	1. เลือกเมนู “แนะนำการเรียนรู้รายบุคคล”	

	3. คลิกเลือกโจทย์ปัญหา 5. คลิกปุ่ม “เปลี่ยนโจทย์ปัญหา”	2. แสดงหน้าจอแนะนำการเรียนรู้รายบุคคล” 4. แสดงหน้าจอการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้ 6. บันทึกการเปลี่ยนโจทย์ปัญหา 7. แสดงโจทย์ปัญหาใหม่
เงื่อนไขการทำงานพิเศษ :	-	

2.4) ยูสเคสการส่งคำตอบโจทย์ปัญหาการเรียนรู้แบบปรับได้

นักศึกษาสามารถดำเนินการส่งคำตอบโจทย์ปัญหาการเรียนรู้แบบปรับได้โดยการส่งไฟล์ซอร์สโค้ด (Source code) เพื่อให้ระบบทำการตรวจสอบความถูกต้องของซอร์สโค้ด ซึ่งกระบวนการนี้จะเกี่ยวข้องกับการคอมไพล์ (Compile) โดยขั้นตอนแสดงดังตารางที่ 3-13

ตารางที่ 3-13 คำอธิบายยูสเคสการส่งคำตอบโจทย์ปัญหาการเรียนรู้แบบปรับได้

ชื่อยูสเคส : การส่งคำตอบโจทย์ปัญหาการเรียนรู้แบบปรับได้	รหัส : UC-13	ระดับความสำคัญ : ปานกลาง
ผู้กระทำหลัก : นักศึกษา		ระดับความซับซ้อน : ปานกลาง
ผู้มีส่วนเกี่ยวข้อง : นักศึกษา		
คำอธิบาย : นักศึกษาสามารถดำเนินการส่งคำตอบโจทย์ปัญหาการเรียนรู้แบบปรับได้ โดยการส่งไฟล์ซอร์สโค้ด (Source code) เพื่อให้ระบบทำการประมวลผลและตรวจสอบความถูกต้องของโค้ดที่เขียนขึ้นโดยกระบวนการนี้จะเกี่ยวข้องกับการคอมไพล์ (Compile) ซอร์สโค้ด (Source code)		
สิ่งกระตุ้น : นักศึกษาต้องการเปลี่ยนโจทย์ปัญหา		
ประเภทสิ่งกระตุ้น : ภายนอก		
เงื่อนไขก่อนการทำงาน : นักศึกษาต้องทำโจทย์ปัญหานั้นก่อน		
เงื่อนไขหลังการทำงาน : ข้อมูลโจทย์ปัญหาถูกบันทึก		
ขั้นตอนการทำงานปกติ :	ผู้ดูแลระบบ	ระบบ
	1. คลิกเลือกโจทย์ปัญหา 3. เลือกภาษาของ Source Code 4. อัปโหลดหรือสร้างไฟล์	2. แสดงหน้าจอแนะนำการเรียนรู้รายบุคคล

	5. คลิกปุ่ม "Submit Code"	6. แสดงหน้าจอการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้ 7. แสดงผลลัพธ์การ Compile 8. แสดงกล่องแจ้งเตือนแก้โจทย์ปัญหานี้สำเร็จแล้ว
เงื่อนไขการทำงานพิเศษ :	8. กรณี Compile แล้ว Test Case ไม่ผ่านทุก Case จะไม่แสดงกล่องแจ้งเตือนแก้โจทย์ปัญหานี้สำเร็จแล้ว	

2.5) การแสดงประวัติการเรียนรู้แบบปรับได้

นักศึกษาสามารถเข้าชมประวัติการเรียนรู้แบบปรับได้เพื่อวิเคราะห์และติดตามพัฒนาการทางการเรียนรู้ของตนเองได้ โดยในหน้าจอนี้จะมีการแสดงข้อมูลสำคัญต่างๆ ที่ช่วยให้สามารถประเมินความก้าวหน้าในการทำโจทย์ปัญหาได้ โดยข้อมูลที่แสดงประกอบด้วยกราฟความถี่ในการส่งงาน ซึ่งช่วยให้เห็นแนวโน้มและพฤติกรรมการส่งงานของนักศึกษาในแต่ละช่วงเวลา นอกจากนี้ยังมีกราฟเส้นที่แสดงถึงระดับความรู้ของนักศึกษาที่เปลี่ยนแปลงไปตามการทำโจทย์ปัญหา ในส่วนของโจทย์ปัญหานั้น หน้าจอจะแสดงข้อมูลเกี่ยวกับโจทย์ที่นักศึกษาเลือกทำ รวมถึงประวัติการส่งงานเพื่อให้สามารถย้อนดูพัฒนาการของตนเองได้ อีกทั้งยังมีข้อมูลร้อยละของการคอมไพล์สำเร็จ ซึ่งช่วยสะท้อนถึงความถูกต้องของโค้ดที่นักศึกษาเขียน และข้อมูลอื่นๆ ได้แก่ จำนวนโจทย์ที่ทำสำเร็จ ระดับความยากของโจทย์ที่เลือกทำ และคะแนนรวมที่ได้รับ โดยรายละเอียดเกี่ยวกับยูสเคสการแสดงหน้าจอประวัติการเรียนรู้แบบปรับได้แสดงตารางที่ 3-14

ตารางที่ 3-14 คำอธิบายยูสเคสการส่งคำตอบโจทย์ปัญหาการเรียนรู้แบบปรับได้

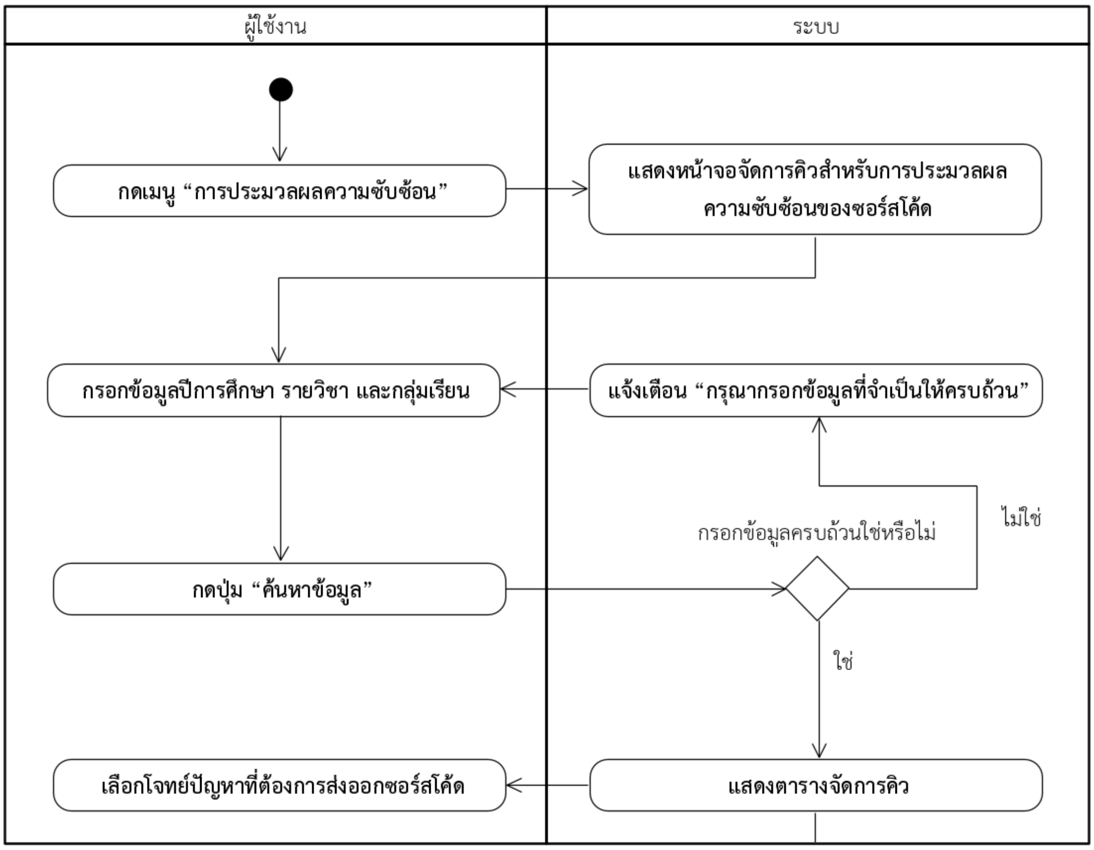
ชื่อยูสเคส : แสดงประวัติการเรียนรู้แบบปรับได้	รหัส : UC-14	ระดับความสำคัญ : ปานกลาง
ผู้กระทำหลัก : นักศึกษา		ระดับความซับซ้อน : ปานกลาง
ผู้มีส่วนเกี่ยวข้อง : นักศึกษา		
คำอธิบาย : นักศึกษาสามารถเข้าชมแสดงประวัติการเรียนรู้แบบปรับได้เพื่อวิเคราะห์และติดตามพัฒนาการทางการเรียนรู้ของตนเองได้ โดยในหน้าจอนี้จะมีการแสดงข้อมูลสำคัญต่างๆ ที่ช่วยให้สามารถประเมินความก้าวหน้าในการทำโจทย์ปัญหาได้		
สิ่งกระตุ้น : นักศึกษาต้องการดูหน้าจอแนะนำการเรียนรู้รายบุคคล		
ประเภทสิ่งกระตุ้น : ภายนอก		
เงื่อนไขก่อนการทำงาน : -		
เงื่อนไขหลังการทำงาน : -		

ขั้นตอนการทำงานปกติ :	ผู้ดูแลระบบ	ระบบ
	1. เลือกเมนู “แนะนำการเรียนรู้รายบุคคล”	2. แสดงหน้าจอแนะนำการเรียนรู้รายบุคคลส่วนของประวัติการเรียนรู้แบบปรับได้
เงื่อนไขการทำงานพิเศษ :	-	

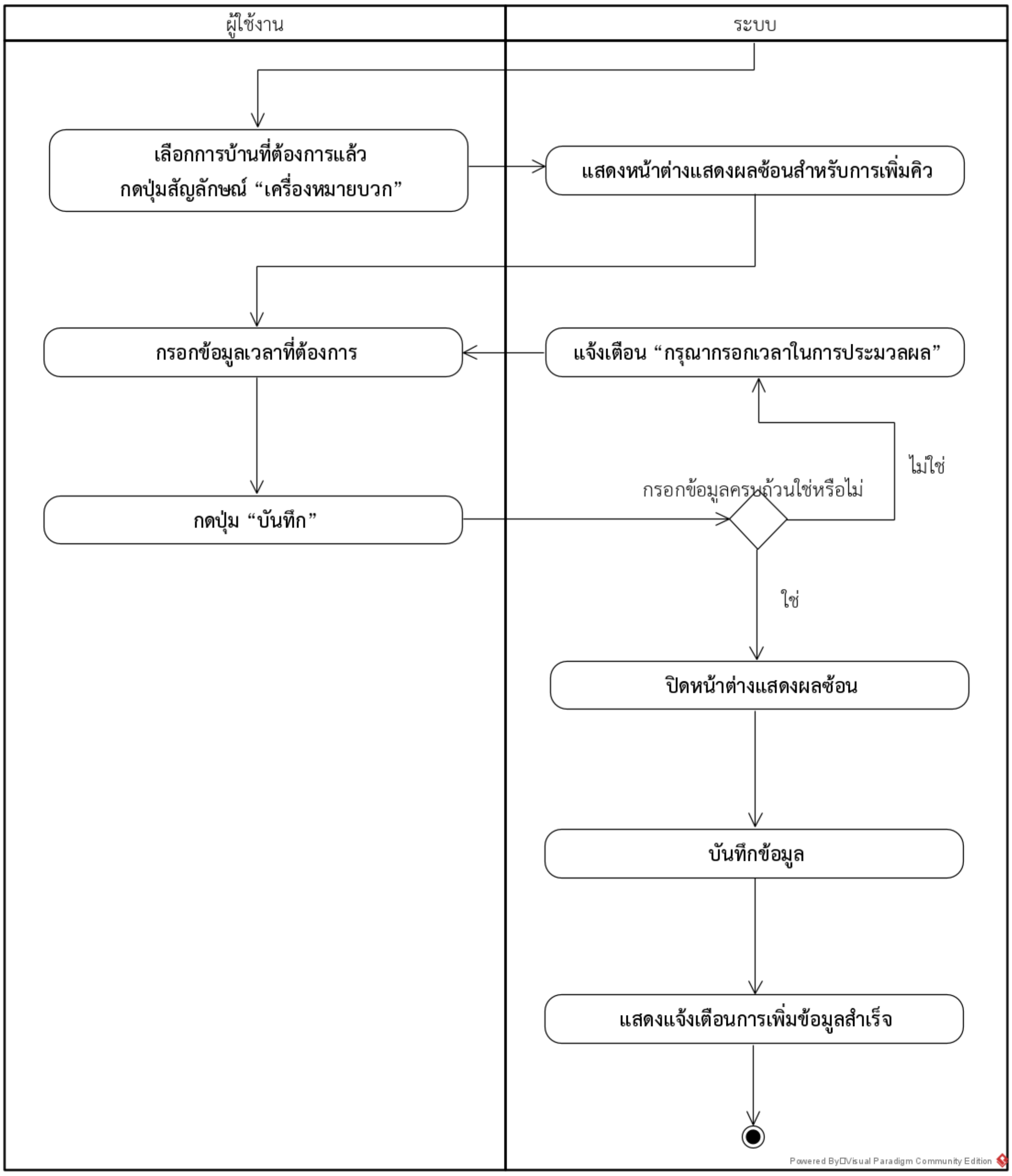
3.4 แผนภาพกิจกรรม (Activity Diagram)

1) ยูสเคสเพิ่มคิว

ส่วนที่อธิบายลำดับขั้นตอนของการเพิ่มการบ้านที่ต้องการเข้าไปในคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด โดยทำการเลือกการบ้านที่ต้องการ และกดปุ่มสัญลักษณ์ “เครื่องหมายบวก” ซึ่งรายละเอียดการทำงาน ดังภาพที่ 3-3



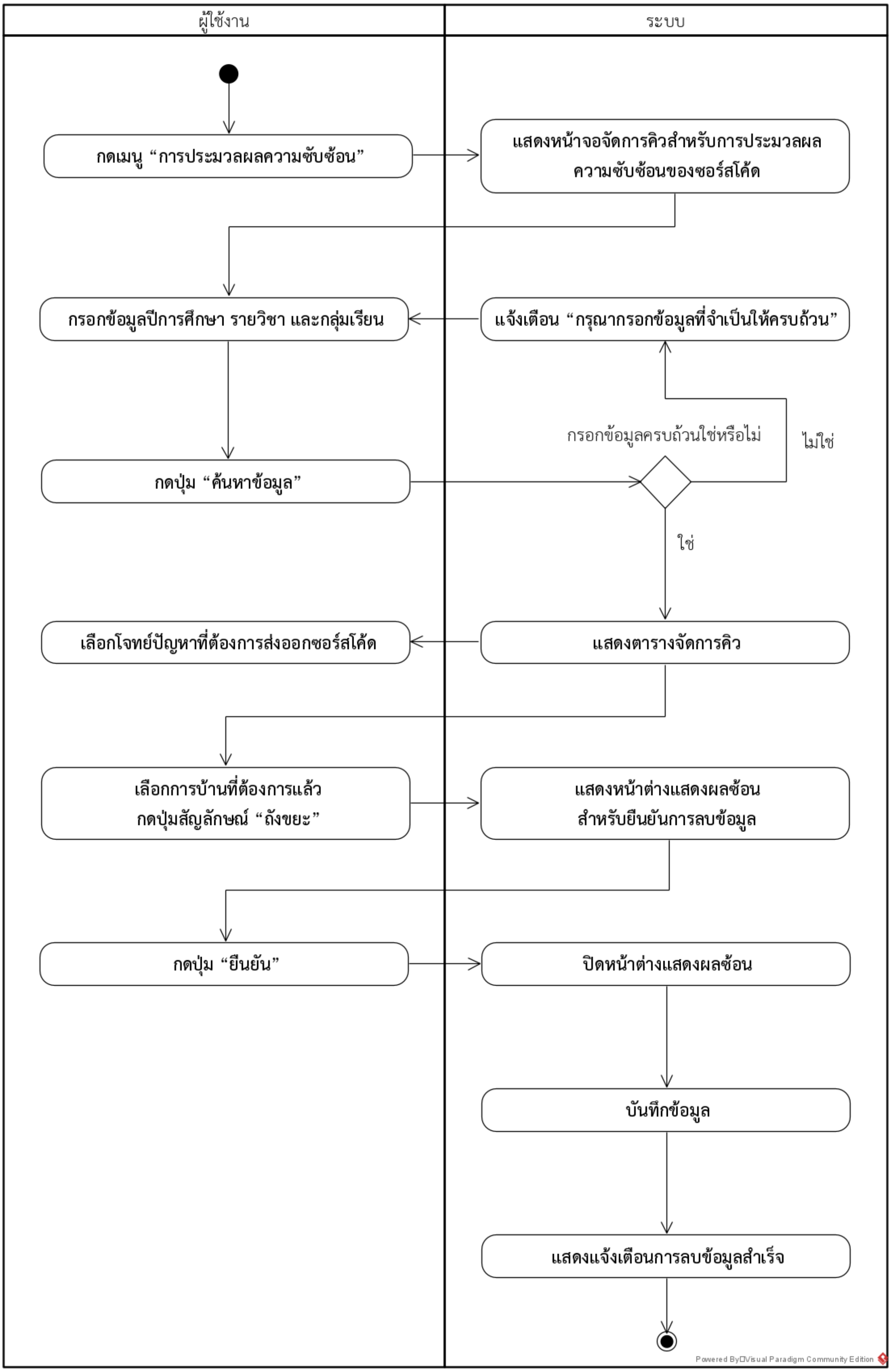
ภาพที่ 3-3 แผนภาพกิจกรรมเพิ่มคิว



ภาพที่ 3-3 แผนภาพกิจกรรมเพิ่มคิว (ต่อ)

2) ยูสเคสลบคิว

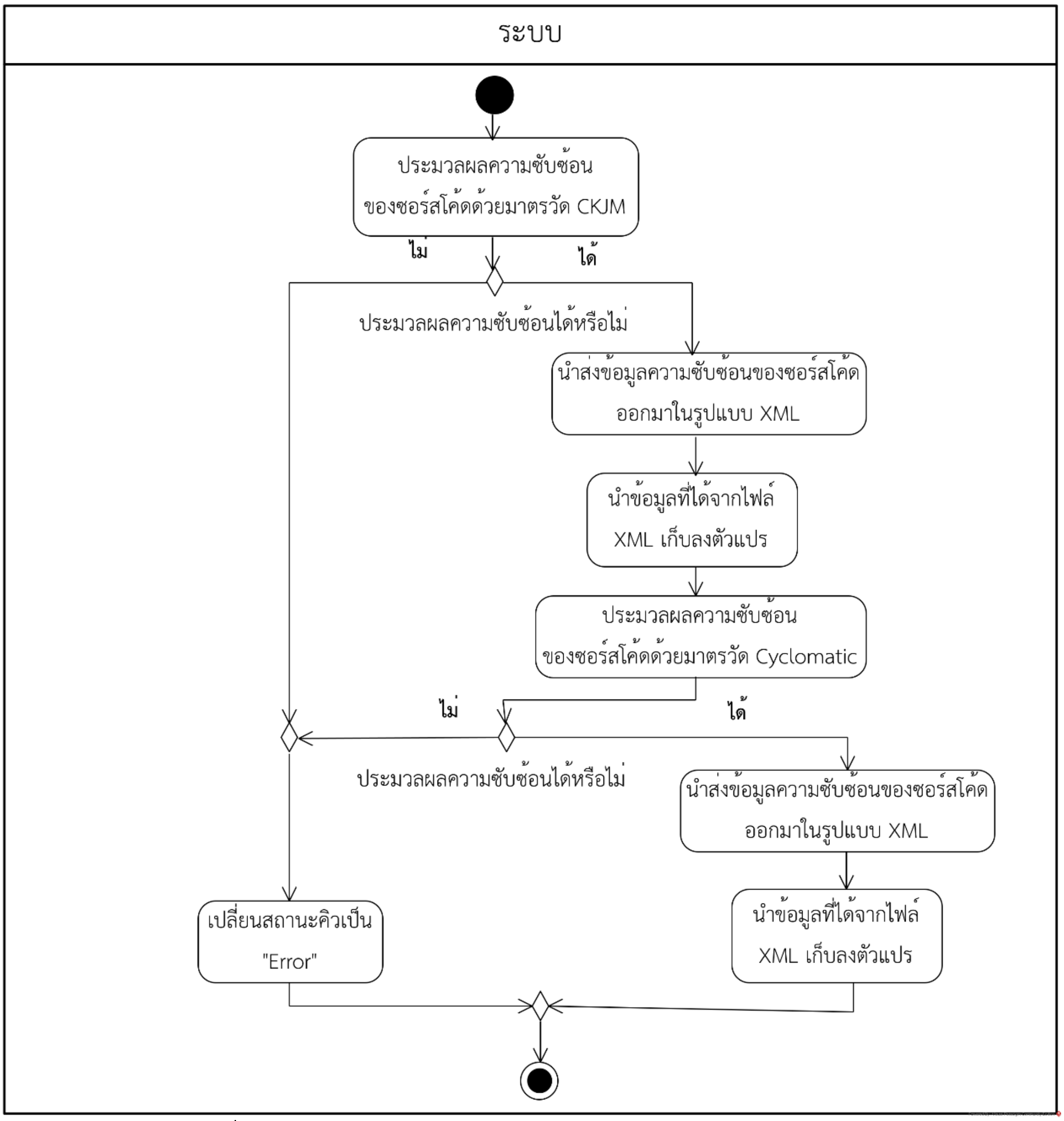
เป็นส่วนที่อธิบายลำดับขั้นตอนของการลบการบ้านที่ต้องการออกจากคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด โดยทำการเลือกการบ้านที่ต้องการ และกดปุ่มสัญลักษณ์ “ถังขยะ” ซึ่งมีคำอธิบายรายละเอียดการทำงาน ดังภาพที่ 3-4



ภาพที่ 3-4 แผนภาพกิจกรรมลบคิว

3) ยูสเคสประมวลผลความซับซ้อนของซอร์สโค้ด

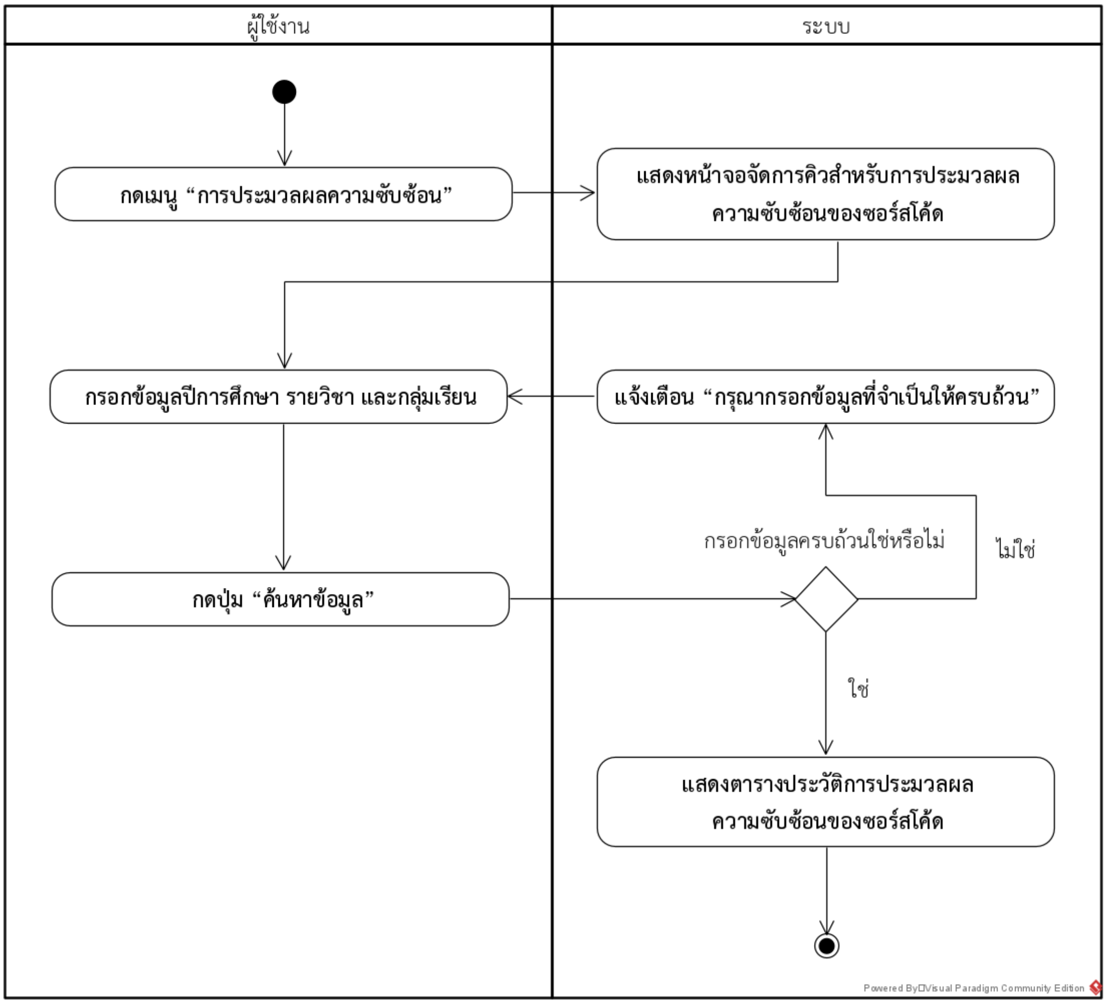
เป็นส่วนที่ระบบจะทำการรันสคริปต์สำหรับการวัดความซับซ้อนด้วยมาตรวัดของ CKJM โดยระบบจะทำการประมวลผลความซับซ้อน หากสามารถวัดค่าความซับซ้อนได้ระบบจะนำข้อมูลค่าความซับซ้อนออกมาในรูปแบบไฟล์ XML จากนั้นระบบจะนำข้อมูลที่ได้จากไฟล์ XML เก็บลงตัวแปร แต่หากไม่สามารถวัดค่าความซับซ้อนได้ระบบจะเปลี่ยนสถานะของคิวเป็น “Error” จากนั้นจะรันสคริปต์สำหรับการวัดความซับซ้อนด้วยมาตรวัดของ Cyclomatic โดยระบบจะทำการประมวลผลความซับซ้อน หากสามารถวัดค่าความซับซ้อนได้ระบบจะนำข้อมูลค่าความซับซ้อนออกมาในรูปแบบไฟล์ XML แต่หากไม่สามารถวัดค่าความซับซ้อนได้ระบบจะเปลี่ยนสถานะของคิวเป็น “Error” จากนั้นระบบจะนำข้อมูลที่ได้จากไฟล์ XML เก็บลงตัวแปร ดังภาพที่ 3-5



ภาพที่ 3-5 แผนภาพกิจกรรมประมวลผลความซับซ้อนของซอร์สโค้ด

4) ยูสเคสแสดงประวัติการประมวลผล

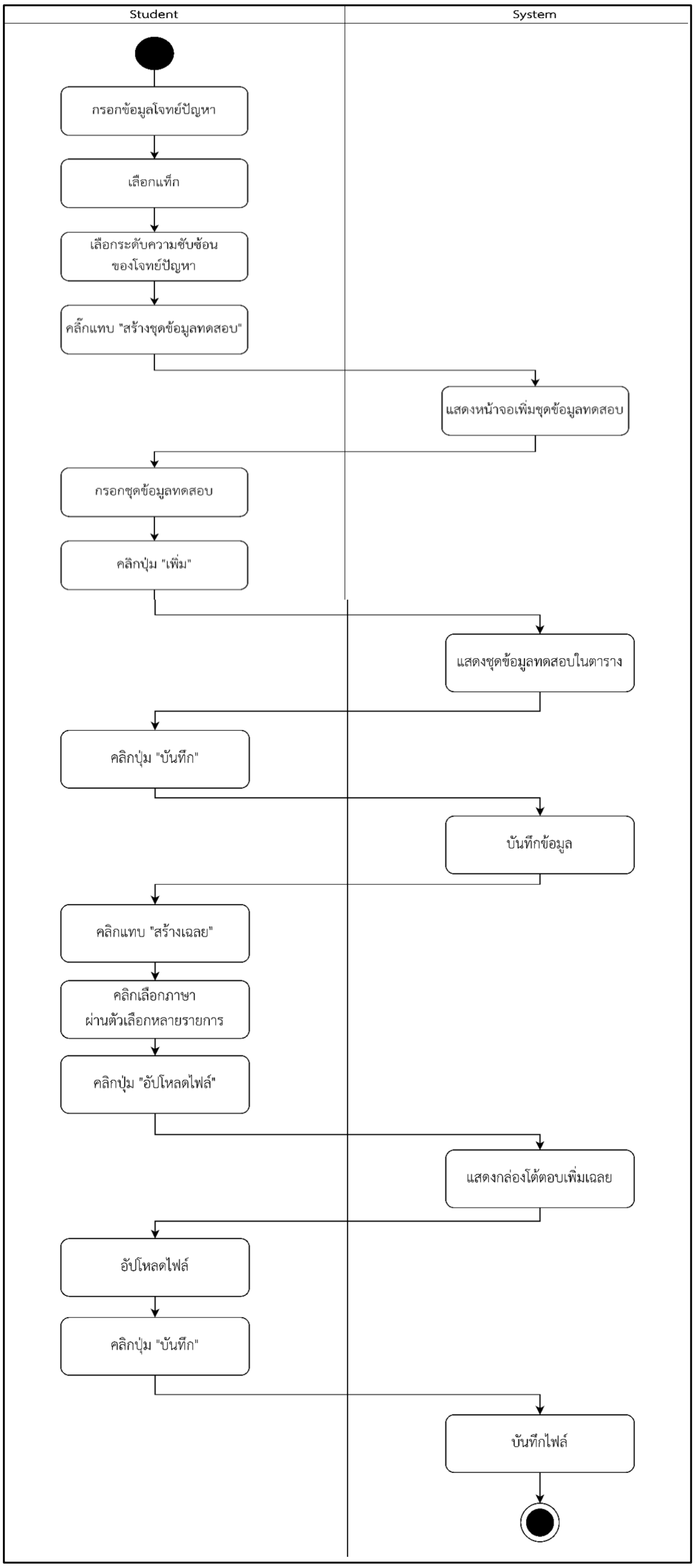
ส่วนที่อธิบายลำดับขั้นตอนของการดูประวัติการประมวลผลความซับซ้อนของซอร์สโค้ด เพื่อช่วยให้อาจารย์สามารถดูได้ว่าชุดการบ้านใดที่ถูกประมวลผลไปแล้วบ้าง ซึ่งมีคำอธิบายรายละเอียดการทำงาน ดังภาพที่ 3-6



ภาพที่ 3-6 แผนภาพกิจกรรมแสดงประวัติการประมวลผล

5) ยูสเคสเพิ่มโจทย์ปัญหา

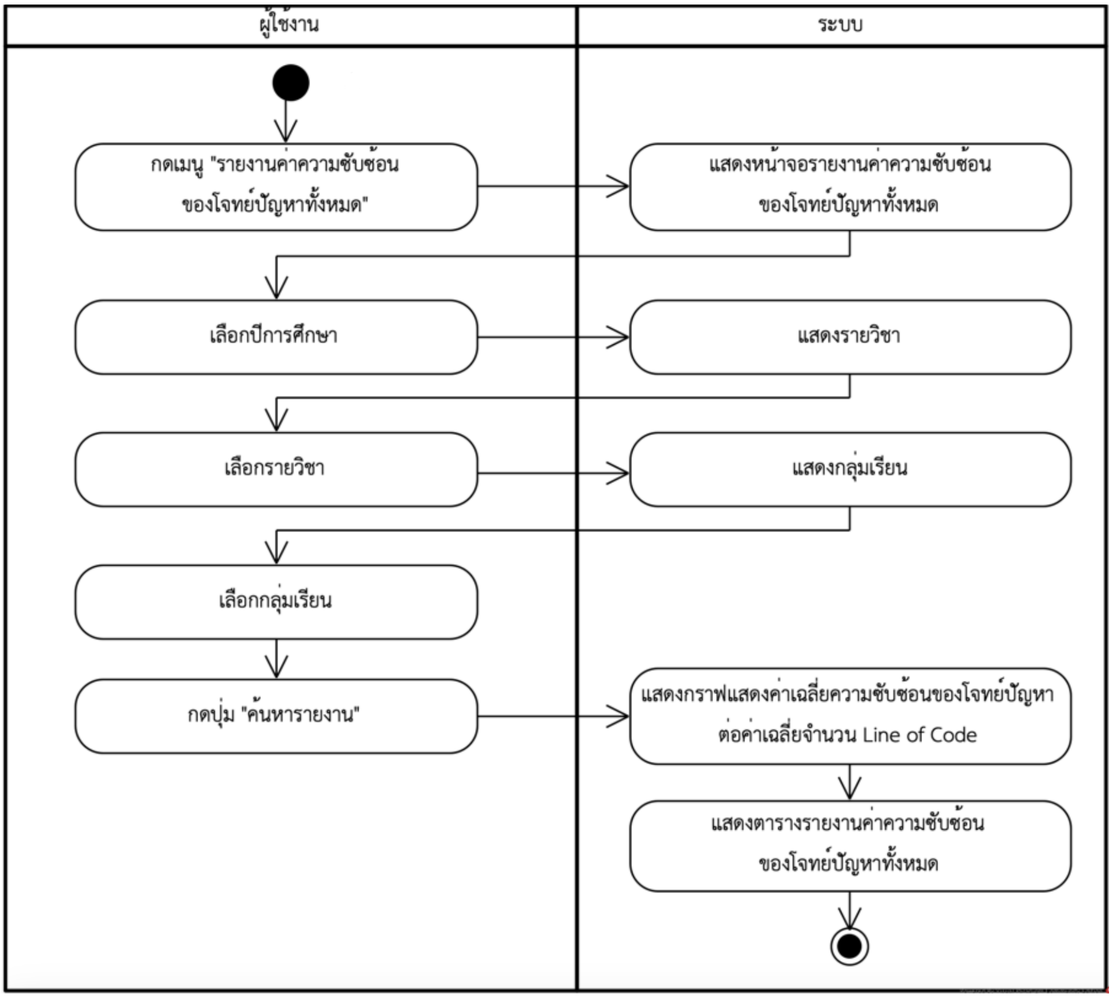
อาจารย์จะสามารถเพิ่มข้อมูลของโจทย์ปัญหาได้โดยการกรอกข้อมูล ชื่อโจทย์ รายละเอียดโจทย์ แท็ก สถานะการใช้งาน และคำตอบของโจทย์ปัญหา ระดับความยากของโจทย์ปัญหา และไฟล์ซอร์สโค้ดเฉลยของโจทย์ปัญหาซึ่งจะมีรายละเอียดคำอธิบายยูสเคสดังภาพที่ 3-7



ภาพที่ 3-7 แผนภาพกิจกรรมเพิ่มโจทย์ปัญหา

6) ยูสเคสรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด

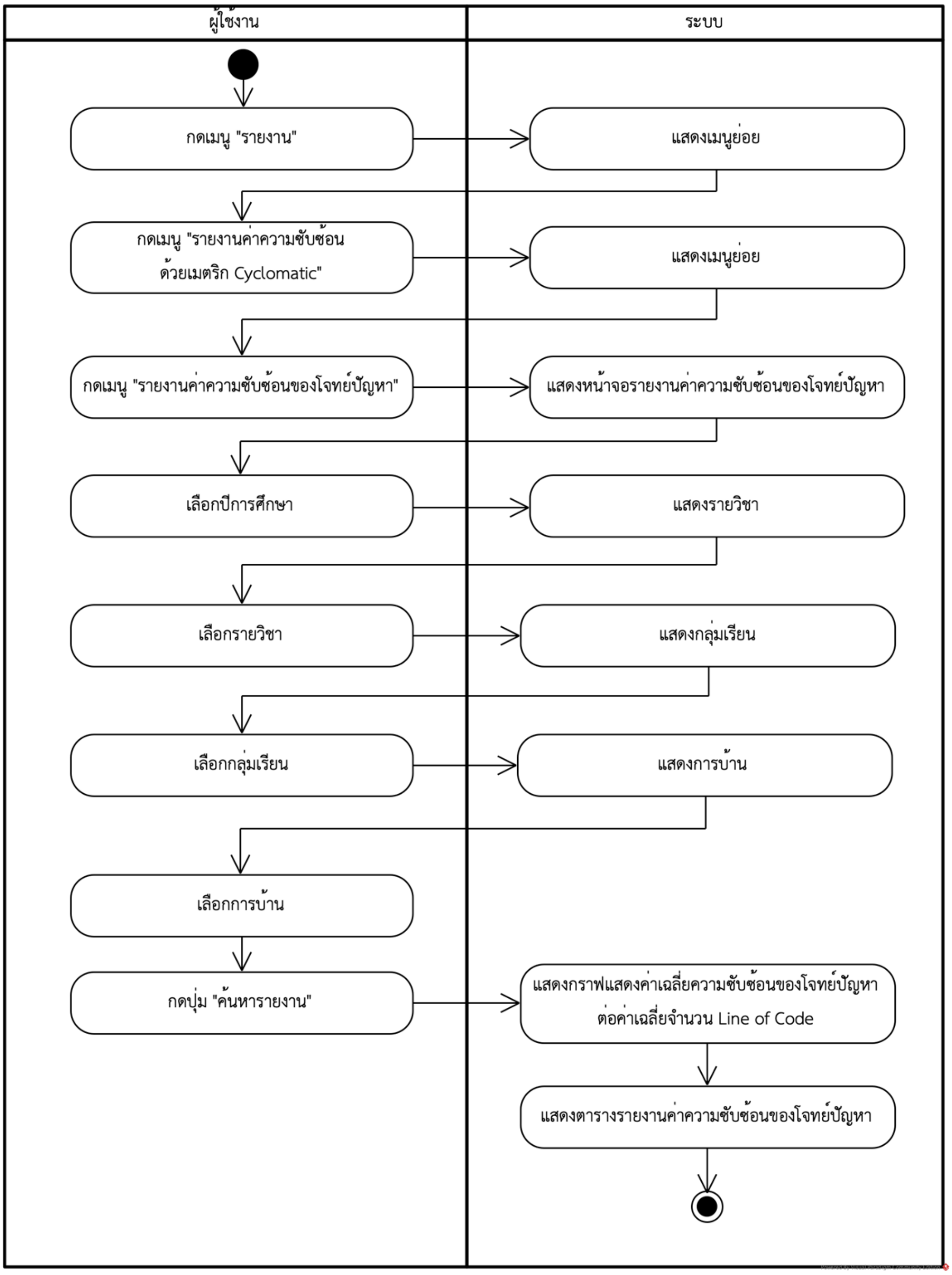
เป็นส่วนที่ผู้ใช้งานสามารถดูรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมดของแต่ละรายวิชาได้ โดยผู้ใช้งานจะต้องเลือกเมนูรายงาน ระบบจะแสดงเมนูย่อย และเลือกเมนูรายงานค่าความซับซ้อนด้วยเมตริก Cyclomatic ระบบจะแสดงเมนูย่อย จากนั้นจึงเลือกเมนูค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด ระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด ซึ่งผู้ใช้งานจะต้องเลือกปีการศึกษา รายวิชา และกลุ่มเรียนก่อน จากนั้นกดปุ่ม “ค้นหารายงาน” เพื่อค้นหารายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด ระบบจะแสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหาต่อค่าเฉลี่ยจำนวน Line of Code และตารางรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด ดังภาพที่ 3-8



ภาพที่ 3-8 แผนภาพกิจกรรมแสดงผลรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด

7) ยูสเคสรายงานค่าความซับซ้อนของโจทย์ปัญหา

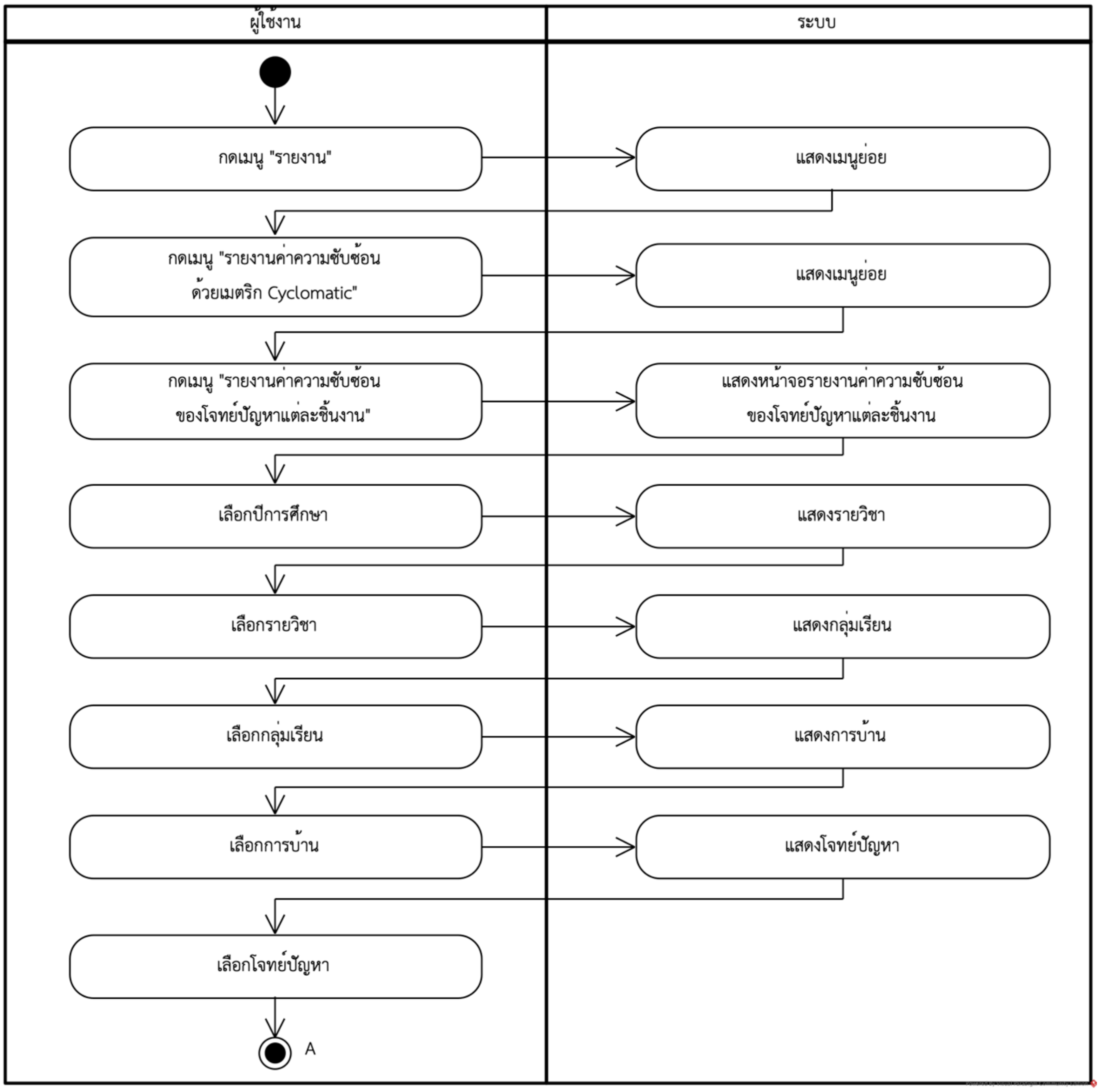
เป็นส่วนที่ผู้ใช้งานสามารถดูรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมดของแต่ละการบ้านได้ โดยผู้ใช้งานจะต้องเลือกเลือกเมนูรายงานค่าความซับซ้อนของโจทย์ปัญหา ระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหา ระบบจะแสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหาต่อค่าเฉลี่ยจำนวน Line of Code และตารางรายงานค่าความซับซ้อนของโจทย์ปัญหา ดังภาพที่ 3-9



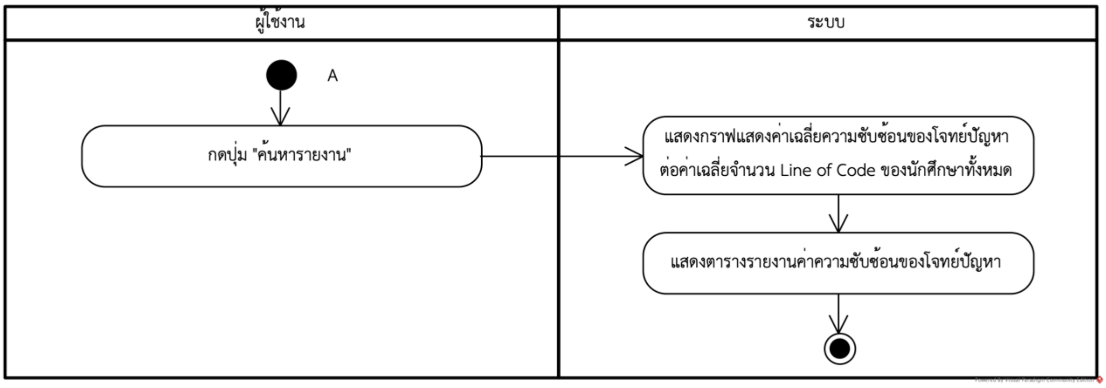
ภาพที่ 3-9 แผนภาพกิจกรรมรายงานค่าความซับซ้อนของโจทย์ปัญหา

8) ยูสเคสรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน

เป็นส่วนที่ผู้ใช้งานสามารถดูรายงานค่าความซับซ้อนของโจทย์ปัญหาได้ โดยผู้ใช้งานจะต้องเลือกเมนูรายงาน ระบบจะแสดงเมนูย่อย และเลือกเมนูรายงานค่าความซับซ้อนด้วยเมตริก Cyclomatic ระบบจะแสดงเมนูย่อย จากนั้นผู้ใช้งานจะต้องเลือกเมนูรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน ระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน ซึ่งผู้ใช้งานจะต้องเลือกปีการศึกษา รายวิชา กลุ่มเรียน การบ้านและโจทย์ปัญหาก่อน จากนั้นกดปุ่ม “ค้นหารายงาน” เพื่อค้นหารายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน ระบบจะแสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหาต่อค่าเฉลี่ยจำนวน Line of Code และตารางรายงานค่าความซับซ้อนของโจทย์ปัญหา ดังภาพที่ 3-10



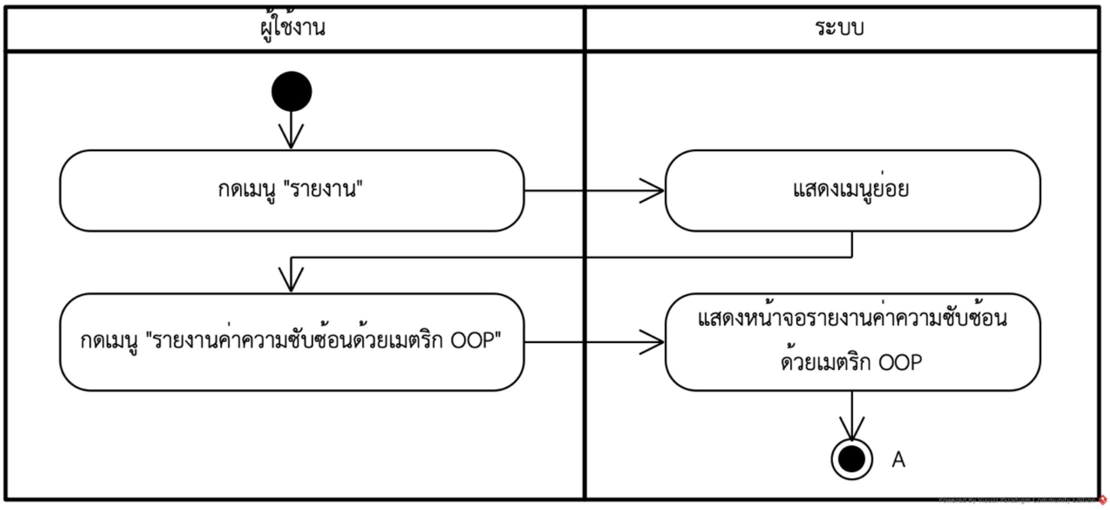
ภาพที่ 3-10 แผนภาพกิจกรรมแสดงผลรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน



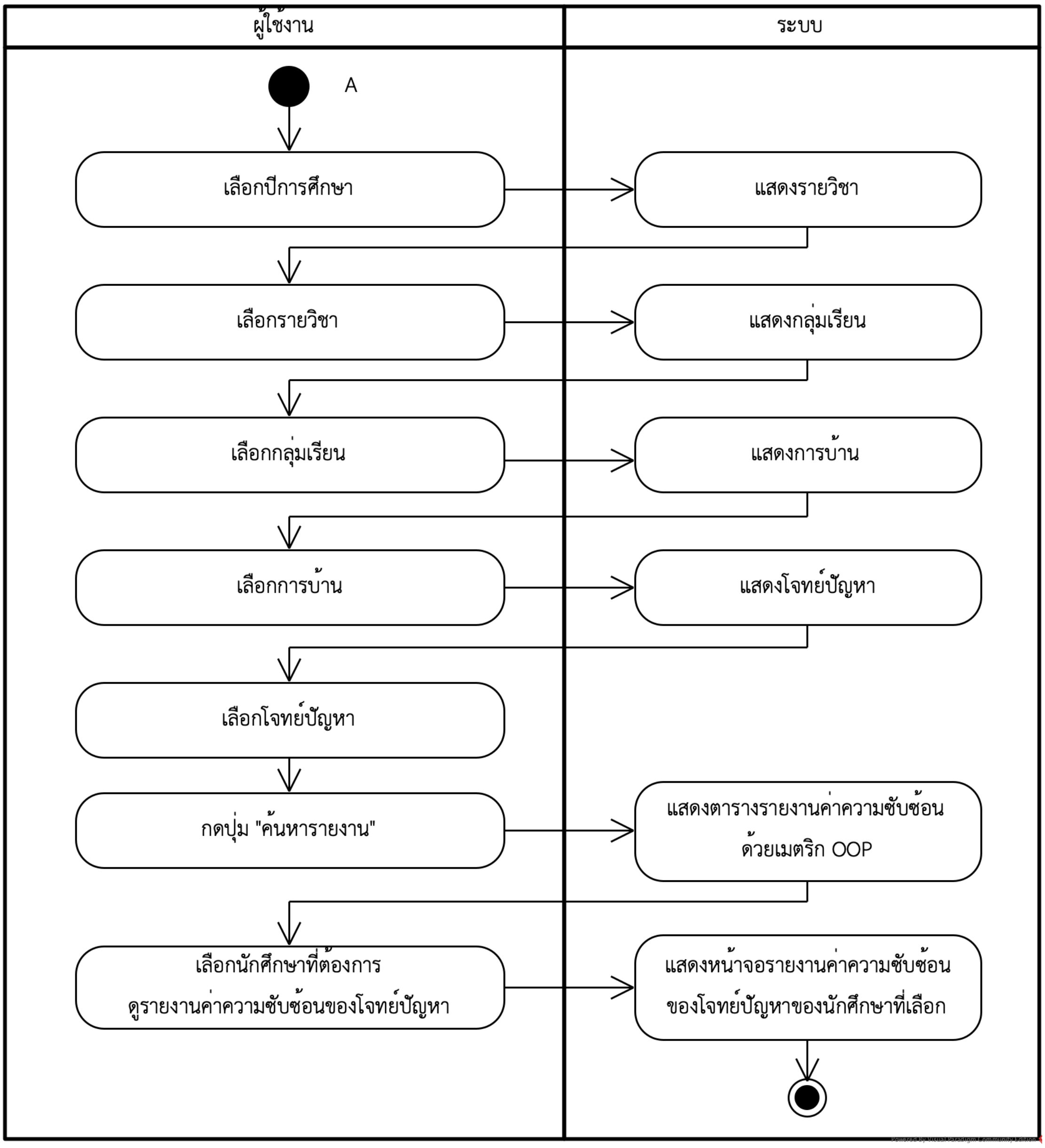
ภาพที่ 3-10 แผนภาพกิจกรรมแสดงผลรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน (ต่อ)

9) ยูสเคสรายงานค่าความซับซ้อนของโจทย์ปัญหาด้วยเมตริกเชิงวัตถุ (OOP)

เป็นส่วนที่ผู้ใช้งานสามารถดูรายงานค่าความซับซ้อนของโจทย์ปัญหาของนักศึกษาแต่ละคนได้ โดยผู้ใช้งานจะต้องเลือกเมนูรายงาน ระบบจะแสดงเมนูย่อย จากนั้นผู้ใช้งานจะต้องเลือกเมนูรายงานค่าความซับซ้อนของโจทย์ปัญหาด้วยเมตริก OOP ระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหารายบุคคล ซึ่งผู้ใช้งานจะต้องเลือกปีการศึกษา รายวิชา กลุ่มเรียน การบ้าน และโจทย์ปัญหาก่อน จากนั้นกดปุ่ม “ค้นหารายงาน” เพื่อค้นหารายงานค่าความซับซ้อนของโจทย์ปัญหาด้วยเมตริก OOP ระบบจะแสดงตารางรายงานค่าความซับซ้อนของโจทย์ปัญหาด้วยเมตริก OOP ผู้ใช้งานจะสามารถเลือกนักศึกษาที่ต้องการดูรายงานค่าความซับซ้อนของโจทย์ปัญหา ระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาของนักศึกษาที่เลือก ดังภาพที่ 3-11



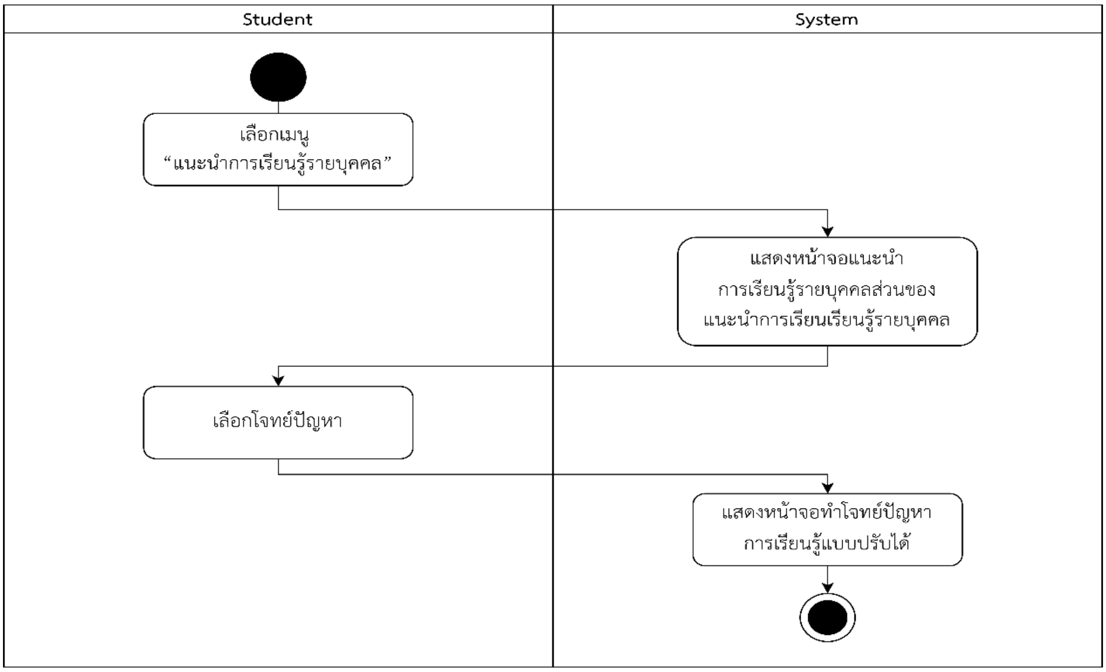
ภาพที่ 3-11 แผนภาพกิจกรรมรายงานค่าความซับซ้อนของโจทย์ปัญหาด้วยเมตริกเชิงวัตถุ



ภาพที่ 3-11 แผนภาพกิจกรรมรายงานค่าความซับซ้อนของโจทย์ปัญหาด้วยเมตริกเชิงวัตถุ (ต่อ)

10) ยูสเคสแนะนำการเรียนรู้รายบุคคล

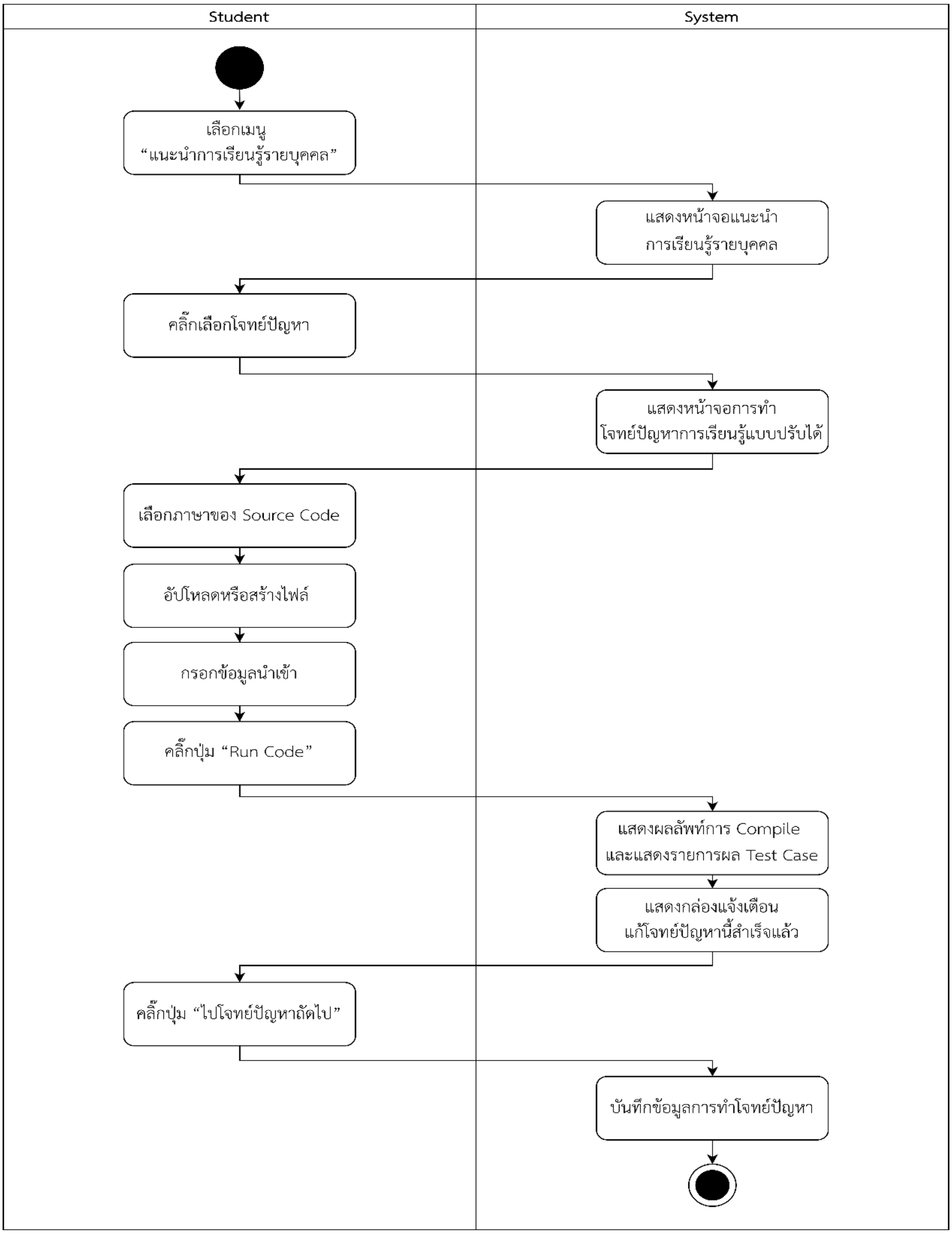
นักศึกษาสามารถเข้าชมหน้าจอแนะนำการเรียนรู้รายบุคคลเพื่อวิเคราะห์และติดตามพัฒนาการทางการเรียนรู้ของตนเองได้อย่างละเอียด โดยในหน้าจอนี้จะมีการแสดงข้อมูลสำคัญต่างๆ ที่ช่วยให้สามารถประเมินความก้าวหน้าในการทำโจทย์ปัญหาได้โดยแสดงภาพที่ 3-12



ภาพที่ 3-12 แผนภาพกิจกรรมการแนะนำการเรียนรู้รายบุคคล

11) ยูสเคสการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้

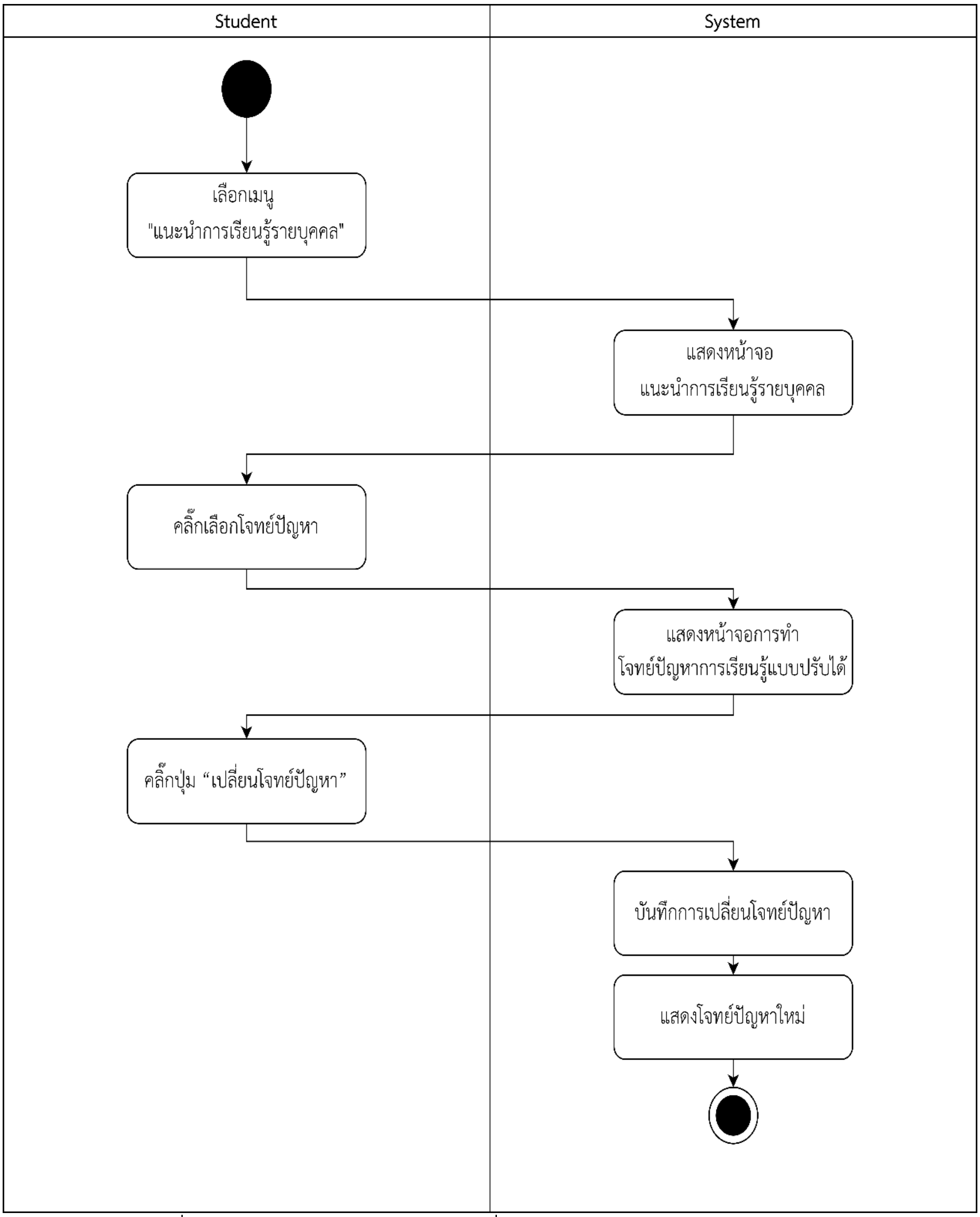
นักศึกษาสามารถดำเนินการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้โดยการพัฒนาและส่งไฟล์ซอร์สโค้ด (Source code) เพื่อให้ระบบทำการประมวลผลและตรวจสอบความถูกต้องของโค้ดเพิ่มเติมเกี่ยวกับกระบวนการดำเนินงาน การส่งไฟล์ การคอมไพล์โค้ด แสดงดังภาพที่ 3-13



ภาพที่ 3-13 แผนภาพกิจกรรมการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้

12) ยูสเคสการเปลี่ยนโจทย์ปัญหาการเรียนรู้แบบปรับได้

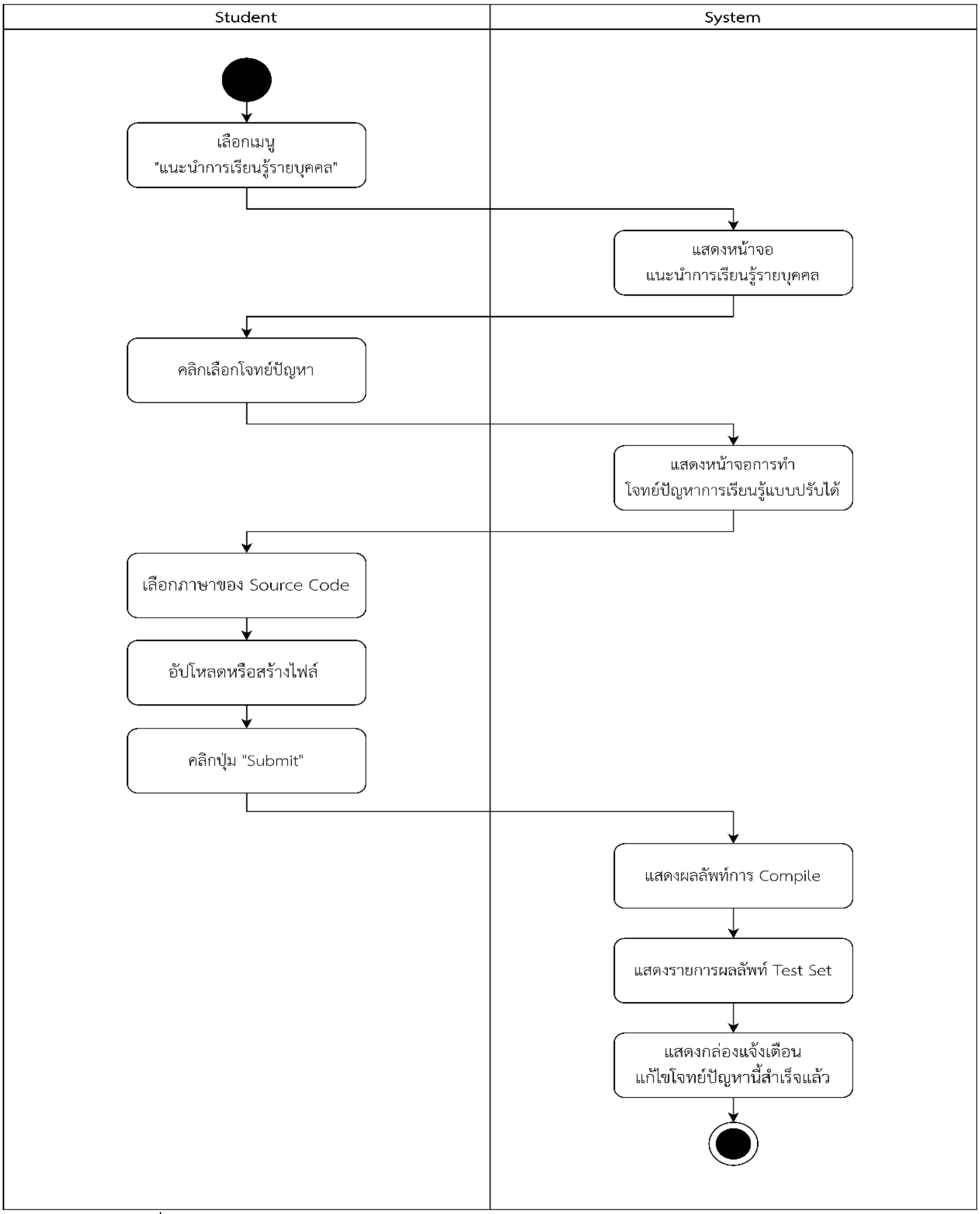
ในการประเมินความรู้ของนักศึกษา หากนักศึกษาพบว่าไม่สามารถทำโจทย์ที่ได้รับมอบหมายได้หรือโจทย์ดังกล่าวไม่เหมาะสมกับระดับความรู้และความสามารถของตน นักศึกษาสามารถเปลี่ยนโจทย์ประเมินได้ แสดงดังภาพที่ 3-14



ภาพที่ 3-14 แผนภาพกิจกรรมการเปลี่ยนโจทย์ปัญหาการเรียนรู้แบบปรับได้

13) ยูสเคสการส่งคำตอบโจทย์ปัญหาการเรียนรู้แบบปรับได้

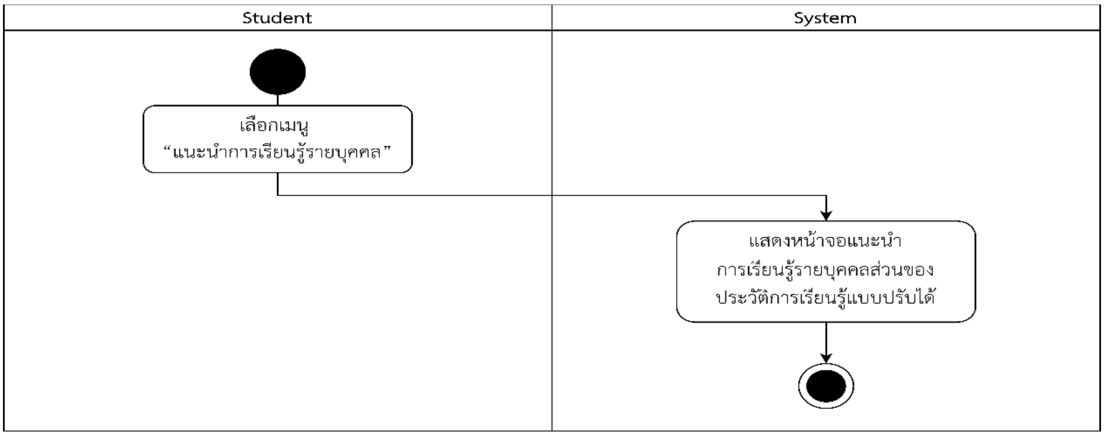
นักศึกษาสามารถดำเนินการส่งคำตอบโจทย์ได้โดยการส่งไฟล์ซอร์สโค้ด (Source code) เพื่อให้ระบบทำการประมวลผลและตรวจสอบความถูกต้องของโค้ดที่เขียนขึ้นโดยกระบวนการนี้จะเกี่ยวข้องกับการคอมไพล์ (Compile) ซอร์สโค้ด และแสดงรายการผลลัพธ์การตรวจสอบตาม Test Case ของโจทย์ปัญหานั้นแสดงดังภาพที่ 3-15



ภาพที่ 3-15 แผนภาพกิจกรรมการส่งคำตอบโจทย์ปัญหาการเรียนรู้แบบปรับได้

14) ยูสเคสแสดงหน้าจอประวัติการเรียนรู้แบบปรับได้

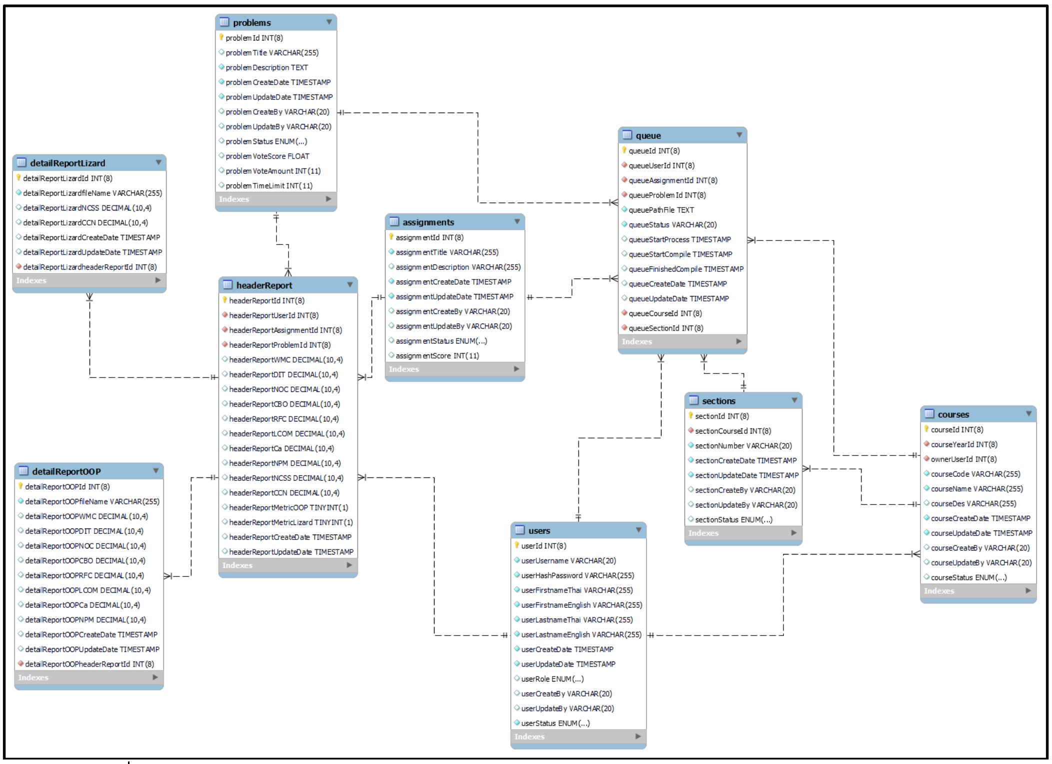
นักศึกษาสามารถเข้าชมหน้าจอแนะนำการเรียนรู้รายบุคคลเพื่อวิเคราะห์และติดตามพัฒนาการทางการเรียนรู้ของตนเองได้อย่างละเอียด โดยในหน้าจอนี้จะมีการแสดงข้อมูลสำคัญต่างๆ ที่ช่วยให้สามารถประเมินความก้าวหน้าในการทำโจทย์ปัญหาได้โดยแสดงภาพที่ 3-16



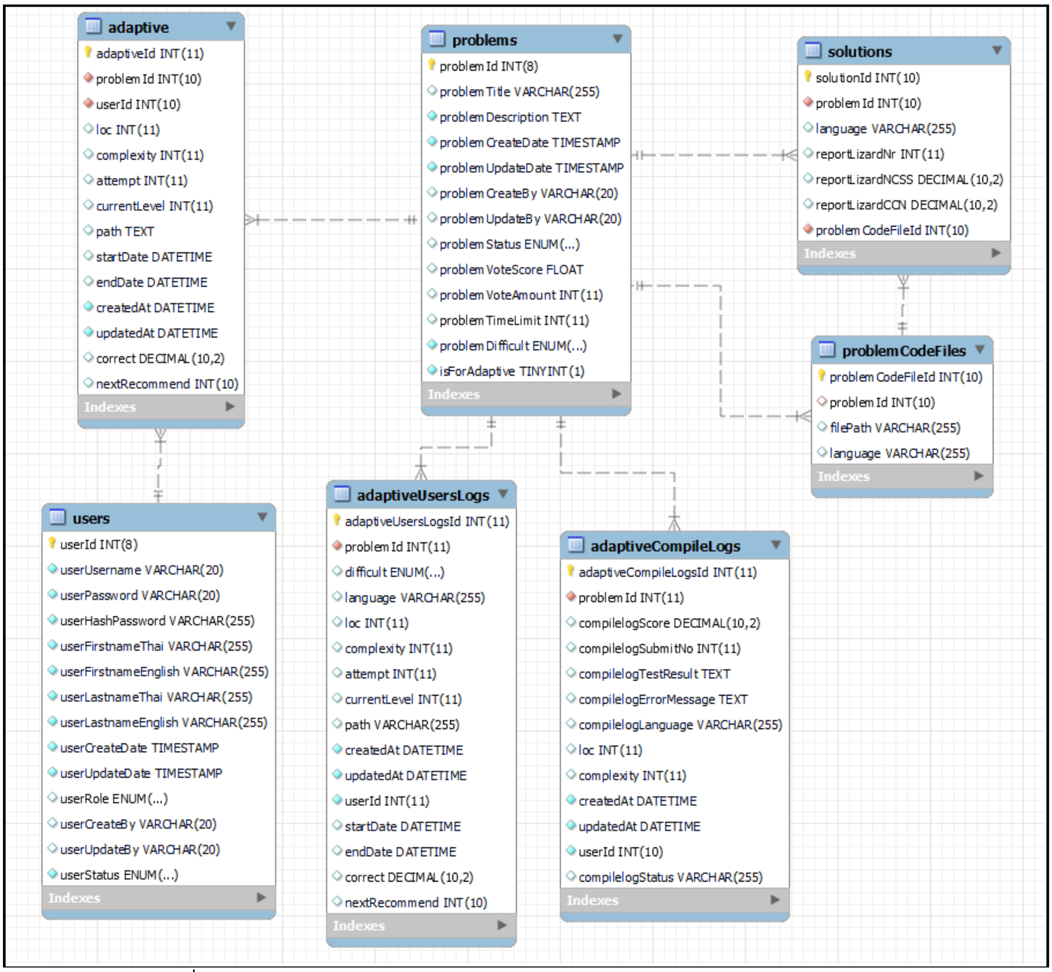
ภาพที่ 3-16 แผนภาพกิจกรรมการแสดงหน้าจอประวัติการเรียนรู้แบบปรับได้

3.5 แผนภาพความสัมพันธ์ของข้อมูล (Entity Relationship Diagram)

การออกแบบแผนภาพความสัมพันธ์ของข้อมูลของแพลตฟอร์มเรียนรู้การเขียนโปรแกรมจะแสดงให้เห็นถึงความสัมพันธ์ระหว่างตารางและองค์ประกอบต่างๆ ของตารางทั้งหมดภายในระบบ เพื่อให้ง่ายต่อการพัฒนาแพลตฟอร์มเรียนรู้การเขียนโปรแกรมมอดูลวัดความซับซ้อนของซอร์สโค้ด โดยรายละเอียดดังภาพที่ 3-17 ที่แสดงถึงความสัมพันธ์ของข้อมูลที่จัดเก็บเกี่ยวกับการวัดความซับซ้อนของซอร์สโค้ดที่นักศึกษาส่งเข้ามาในระบบ ส่วนภาพที่ 3-18 แสดงถึง ความสัมพันธ์ของข้อมูลที่จัดเก็บที่เกี่ยวข้องกับการจัดเก็บข้อมูลสำหรับใช้ในการแนะนำการเรียนรู้แบบปรับได้ โดยรายละเอียดของตารางข้อมูลอยู่ที่ภาคผนวก ข



ภาพที่ 3-17 แผนภาพความสัมพันธ์ข้อมูลส่วนของการวัดความซับซ้อนของซอร์สโค้ด



ภาพที่ 3-18 แผนภาพความสัมพันธ์ข้อมูลส่วนของการเรียนรู้แบบปรับได้

3.6 การออกแบบโมเดลสำหรับแนะนำโจทย์ปัญหาที่เหมาะสมกับผู้เรียน

3.6.1 ขั้นตอนการออกแบบโมเดล

การออกแบบโมเดลในงานวิจัยนี้ได้อิงตามวงจรชีวิตของการเรียนรู้ของเครื่อง (Machine Learning Lifecycle) ซึ่งประกอบด้วยขั้นตอนดังต่อไปนี้

1) การกำหนดปัญหา (Problem definition) เป็นขั้นตอนที่ใช้ในการระบุวัตถุประสงค์ของระบบ ซึ่งในงานวิจัยนี้มีวัตถุประสงค์คือ การแนะนำคำถามที่เหมาะสมกับระดับความสามารถของผู้เรียน

2) การรวบรวมข้อมูล (Data collection) โดยขั้นตอนนี้จะเป็นการดึงข้อมูลจากแหล่งต่างๆ เช่น ฐานข้อมูล ไฟล์ CSV ซึ่งผู้วิจัยได้ใช้ข้อมูลจากฐานข้อมูลแพลตฟอร์มการเรียนรู้การเขียนโปรแกรม

3) การเตรียมและทำความสะอาดข้อมูล (Data preprocessing) เป็นขั้นตอนที่ใช้ในการกำจัดค่าผิดปกติ (Outliers) ค่าที่หายไป (Missing values) และปรับรูปแบบข้อมูลให้อยู่ในรูปแบบที่เหมาะสม ซึ่งผู้วิจัยได้ทำการคัดเลือกข้อมูลเฉพาะซอร์สโค้ดที่สามารถคอมไพล์ได้ และแปลงค่าข้อมูลต่างๆ เช่น ระยะเวลาในการทำโจทย์ปัญหา คะแนนที่ได้ ให้อยู่ในรูปแบบที่เหมาะสม

4) การสร้างและคัดเลือกคุณลักษณะ (Feature engineering) ขั้นตอนนี้เป็นการคัดเลือกฟีเจอร์ที่มีความสำคัญสูง (Feature selection) หรือสร้างฟีเจอร์ใหม่จากข้อมูลเดิม (Feature extraction)

5) การเลือกโมเดล (Model selection) เป็นขั้นตอนการเลือกอัลกอริธึมที่เหมาะสม เช่น Decision Tree, Random Forest, SVM, Neural Network โดยพิจารณาจากประเภทของปัญหาและลักษณะของข้อมูล

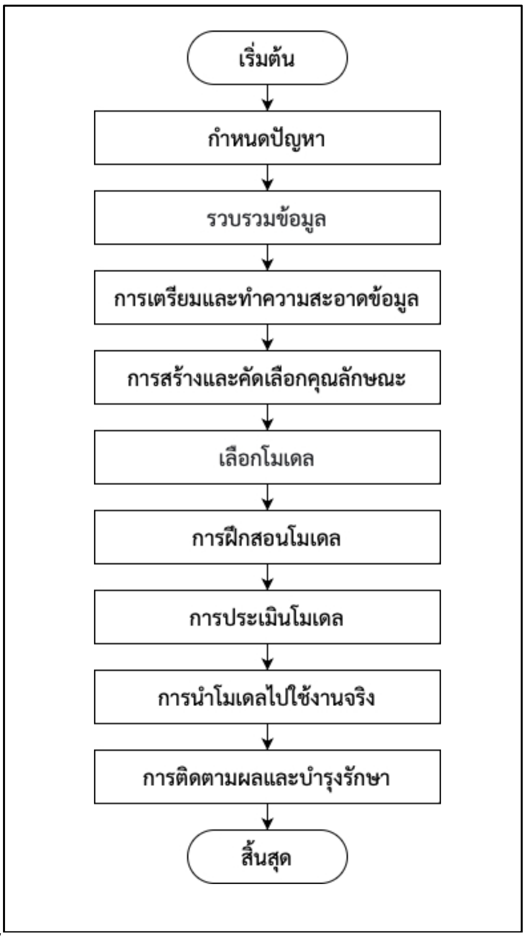
6) การฝึกสอนโมเดล (Model Training) เป็นการแบ่งข้อมูลเป็นชุดฝึกสอนและทดสอบ (Train/Test Split) รวมไปถึงการปรับค่าพารามิเตอร์ของโมเดล (Hyperparameter Tuning)

7) การประเมินโมเดล (Model Evaluation) โดยใช้ตัวชี้วัดทางสถิติเพื่อวัดประสิทธิภาพ เช่น Accuracy, Precision, Recall, F1-score

8) การนำโมเดลไปใช้งานจริง (Model Deployment) เป็นขั้นตอนการทำให้โมเดลพร้อมให้บริการ เช่น ผ่าน API หรือแอปพลิเคชัน

9) การติดตามผลและบำรุงรักษา (Monitoring and Maintenance) เป็นขั้นตอนตรวจสอบประสิทธิภาพของโมเดลในสภาพแวดล้อมจริง (Concept Drift) ตลอดจนอัปเดตหรือฝึกสอนใหม่

โดยขั้นตอนการออกแบบโมเดลนี้นแสดงด้วยผังงานดังภาพที่ 3-19



ภาพที่ 3-19 ขั้นตอนการสร้างโมเดลสำหรับการแนะนำการเรียนรู้แบบปรับได้

3.6.2 คุณลักษณะข้อมูลนำเข้า (Input Features)

หลังจากนักศึกษาใช้งานระบบแต่ละครั้ง ระบบจะทำการรวบรวมชุดข้อมูลคุณลักษณะ (Features) เพื่อสะท้อนถึงผลการเรียนรู้ของผู้เรียน ณ ปัจจุบัน และลักษณะเฉพาะของซอร์สโค้ดของโจทย์ที่ทำเสร็จแล้ว โดยคุณลักษณะที่รวบรวมมีดังนี้

1) จำนวนบรรทัดของโค้ด (Lines of Code: LOC) หมายถึงจำนวนบรรทัดซอร์สโค้ดของผู้เรียนส่งเป็นคำตอบในแต่ละโจทย์ปัญหา

2) ความซับซ้อนของโค้ด (Cyclomatic Complexity: CCN) เป็นการวัดความซับซ้อนเชิงตรรกะของโค้ด โดยระบุจำนวนเส้นทางอิสระที่มีอยู่ในโปรแกรม

3) ระดับความซับซ้อนของโจทย์ (Complexity) เป็นป้ายกำกับ (label) ที่กำหนดโดยผู้สอน เพื่อระบุระดับความยากของโจทย์แต่ละข้อ

4) ระยะเวลาในการทำโจทย์ (Time spent) คำนวณจากเวลาที่ผู้เรียนเริ่มต้นทำโจทย์จนถึงเวลาที่มีการส่งคำตอบครั้งสุดท้าย โดยใช้หน่วยนับเป็นนาที

5) อัตราความถูกต้อง (Correct) เป็นอัตราส่วนของจำนวนกรณีทดสอบ (Test case) ที่ผ่านการตรวจสอบว่าถูกต้องเทียบกับจำนวนกรณีทดสอบทั้งหมด โดยมีค่าระหว่าง 0 ถึง 1

6) จำนวนครั้งที่ส่งคำตอบ (Attempt) คือ จำนวนครั้งที่ผู้เรียนส่งคำตอบในแต่ละโจทย์ปัญหา ซึ่งสามารถสะท้อนถึงความพยายามหรือระดับความยากที่ผู้เรียนรับรู้

7) ระดับปัจจุบันของผู้เรียน (Current Level) คือ ระดับความสามารถของผู้เรียน ณ ขณะนั้น ที่ค่าเริ่มต้นคือ ระดับปานกลาง หลังจากผู้เรียนเริ่มต้นใช้งานระบบ ระบบจะทำการประเมินระดับความสามารถของผู้เรียนและเก็บข้อมูลนี้ไว้เพื่อใช้ในการแนะนำโจทย์ปัญหาถัดไป ถึงแม้ว่าค่านี้จะไม่ใช่การวัดความสามารถของผู้เรียนโดยตรง แต่ใช้เป็นตัวแทนเพื่อบ่งชี้ระดับความก้าวหน้าของผู้เรียน และช่วยให้ระบบสามารถปรับระดับของโจทย์ถัดไปได้อย่างเหมาะสม

3.6.3 การเตรียมข้อมูลก่อนฝึกสอนโมเดล

ขั้นตอนการเตรียมข้อมูลก่อนกระบวนการฝึกสอน (training) เพื่อให้ข้อมูลมีความถูกต้องครบถ้วน และอยู่ในรูปแบบที่เหมาะสมกับการประมวลผลของโมเดล Machine Learning โดยขั้นตอนการทำงานมีดังนี้

1) การกรองและคัดเลือกข้อมูล (Filtering and Selection) ใช้วิธีการเลือกจากกิจกรรมที่มีการจำกัดเวลา ได้แก่ การทดสอบย่อย การสอบกลางภาค และการสอบปลายภาค เพื่อให้ค่าคุณลักษณะระยะเวลาในการทำโจทย์ (Time Spent) มีความสม่ำเสมอ เนื่องจากกิจกรรมเหล่านี้จะถูกจำกัดเวลาไม่เกิน 3 ชั่วโมง ซึ่งส่งผลให้การวัดผลตามเวลาแม่นยำและสม่ำเสมอกว่างานที่มอบหมายอื่นๆ เช่น การบ้านที่นักศึกษาอาจใช้เวลานานถึง 6 วันในการทำโจทย์ปัญหา

2) การจัดการค่าผิดปกติ (Outlier Handling) เป็นการตรวจสอบค่าที่มีค่าผิดปกติและทำการตัดข้อมูลนั้นออกหรือแปลงให้อยู่ในช่วงที่เหมาะสม เพื่อป้องกันผลกระทบต่อการฝึกโมเดล

3) การแปลงข้อมูล (Feature Transformation) เป็นการเปลี่ยนค่าคุณลักษณะเชิงปริมาณบางรายการ เช่น Time Spent แปลงให้อยู่ในรูปแบบนาที เพื่อให้อยู่ในช่วงที่โมเดลสามารถประมวลผลได้อย่างเหมาะสมและมีเสถียรภาพ

4) การจัดการค่าที่หายไป (Missing Value Handling กรณีที่คุณลักษณะบางรายการขาดหาย เช่น Complexity จะเติมค่าด้วยค่ามัธยฐาน

5) การจัดโครงสร้างข้อมูล (Data Structuring and Encoding) ข้อมูลถูกรวบรวมในรูปแบบตาราง โดยแต่ละแถวแทนถึงข้อมูลของนักศึกษาที่ทำการเขียนโปรแกรมต่อหนึ่งโจทย์ปัญหา แต่ละคอลัมน์ประกอบด้วย ค่าความซับซ้อนของโจทย์ปัญหา ระยะเวลาทำการโจทย์ จำนวนครั้งที่ส่งคำตอบ อัตราความถูกต้อง คุณลักษณะของซอร์สโค้ด คือ จำนวนบรรทัดของซอร์สโค้ด (LOC) ค่าความซับซ้อนของซอร์สโค้ด (CCN) ส่วนตัวแปรเป้าหมาย (Target) คือระดับความยากที่ผู้เรียนควรได้รับในคำถามถัดไป

6) การแบ่งชุดข้อมูล (Train-Test Split) โดยข้อมูลถูกแบ่งออกเป็น 2 ชุด ได้แก่ ชุดฝึกอบรม (80%) และชุดทดสอบ (20%)

3.6.4 กระบวนการฝึกสอนโมเดล (Model Training Process)

โมเดล Random Forest ถูกฝึกด้วยข้อมูลในอดีตของผู้เรียนที่เก็บรวบรวมจากรายวิชาการเขียนโปรแกรมเบื้องต้น โดยชุดข้อมูลดังกล่าวประกอบด้วยคุณลักษณะด้านพฤติกรรมการเขียนโปรแกรม และป้ายกำกับระดับความยาก ที่กำหนดโดยอาจารย์ผู้สอน

ก่อนการฝึก ระบบได้ดำเนินการเตรียมข้อมูล (Data preprocessing) เพื่อให้ข้อมูลมีความสม่ำเสมอและจัดการกับค่าผิดปกติที่อาจเกิดขึ้น โดยเลือกใช้ข้อมูลจากแบบทดสอบย่อย (quiz) การสอบกลางภาค และการสอบปลายภาค เป็นชุดข้อมูลฝึกโมเดล เนื่องจากการประเมินเหล่านี้มีการจำกัดเวลา (โดยทั่วไปไม่เกิน 3 ชั่วโมง) ซึ่งส่งผลให้การวัดผลตามเวลาแม่นยำและสม่ำเสมอกว่างานที่มอบหมายอื่นๆ เช่น การบ้านที่นักศึกษาอาจใช้เวลานานถึง 6 วันในการทำโจทย์ปัญหา

ชุดข้อมูลถูกแบ่งเป็น 2 ส่วนในสัดส่วน 80/20 สำหรับการฝึกและการทดสอบ โมเดล Random Forest ตั้งค่าด้วยจำนวนต้นไม้ (estimators) เท่ากับ 100 และใช้ค่าพารามิเตอร์เริ่มต้นของไลบรารี Scikit-learn โดยพิจารณาโจทย์นี้เป็นปัญหาการจัดจำแนกประเภท (classification) โมเดลเรียนรู้รูปแบบจากการผสมกันของคุณลักษณะเพื่อนำไปใช้ในการพยากรณ์ระดับความยากที่เหมาะสมที่สุด

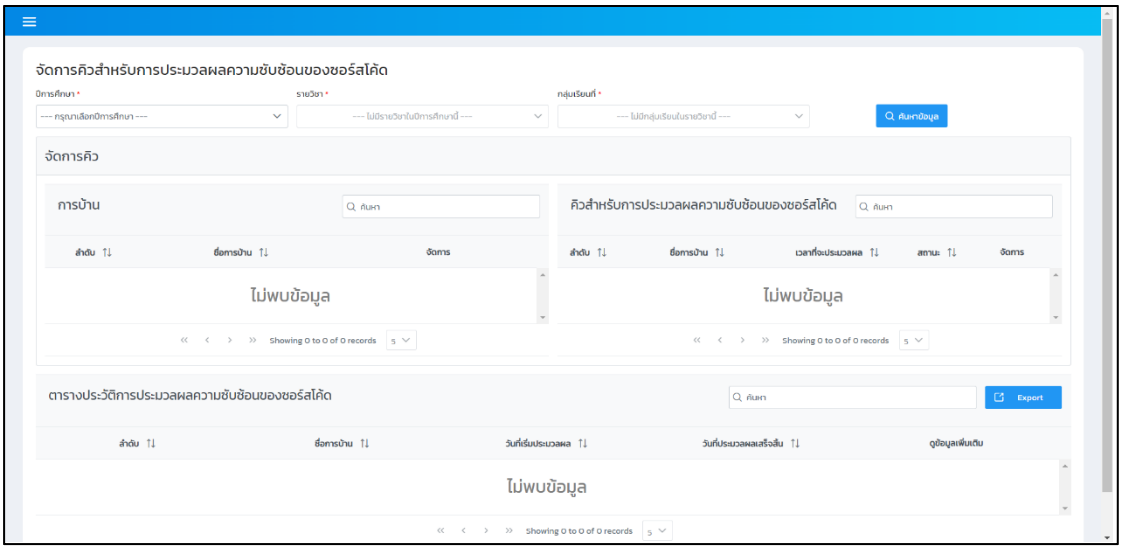
บทที่ 4
ผลการดำเนินงาน

ผลการดำเนินโครงงานการวิจัยมอดูลการวัดความซับซ้อนของซอร์สโค้ด สำหรับแพลตฟอร์มการเรียนรู้การเขียนโปรแกรม คณะผู้วิจัยได้วิเคราะห์ ออกแบบ และพัฒนาซอฟต์แวร์ โดยผลการดำเนินงานพัฒนาซอฟต์แวร์มีรายละเอียดดังต่อไปนี้

4.1 มอดูลประมวลผลความซับซ้อนของซอร์สโค้ด

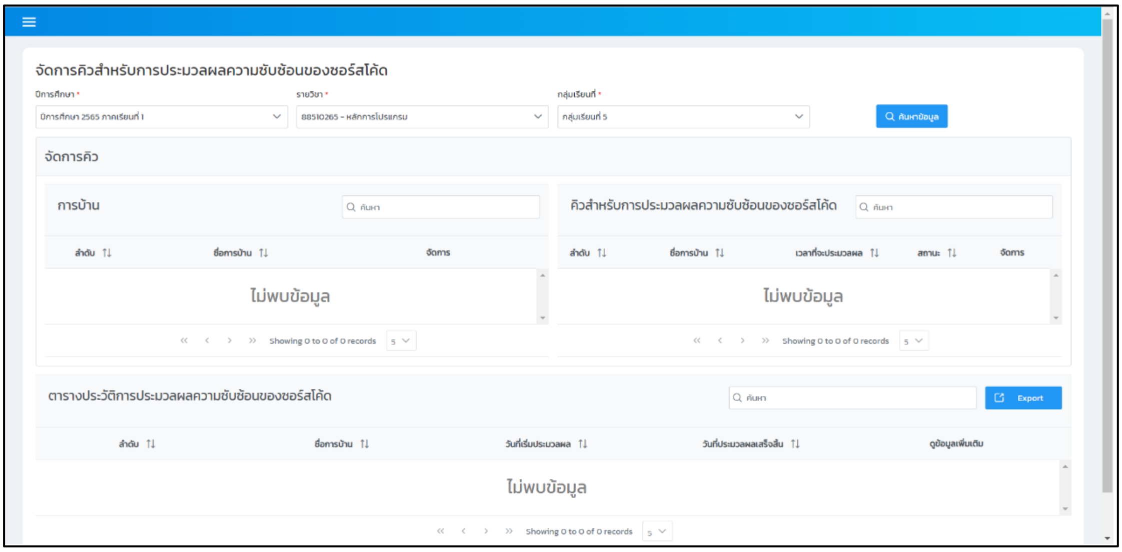
4.1.1 การจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด

การจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด เป็นการสร้างลำดับคิวในการประมวลผลหาความซับซ้อนของโจทย์ปัญหาที่ต้องการ โดยอาจารย์สามารถเข้าไปที่เมนู “การประมวลผลความซับซ้อน” ระบบจะทำการแสดงหน้าจอจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด ดังภาพที่ 4-1



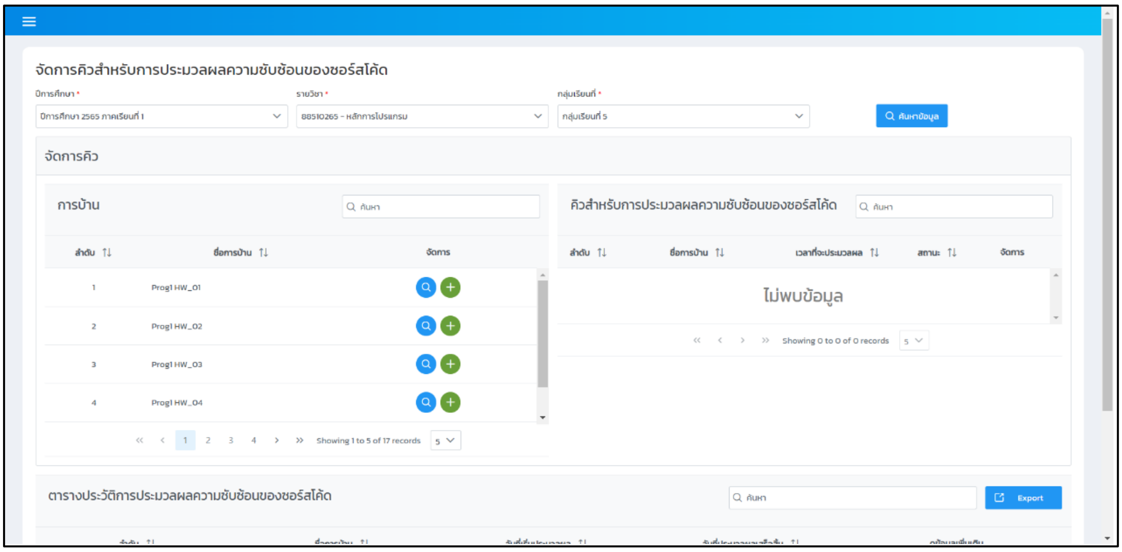
ภาพที่ 4-1 หน้าจอจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด

จากนั้นให้อาจารย์ทำการกรอกข้อมูลจากซ้ายไปขวาตามลำดับให้ครบถ้วน และจำเป็นต้องทำการเลือกปีการศึกษาแล้ว ถึงจะสามารถกรอกข้อมูลรายวิชา และกลุ่มเรียนได้ โดยมีตัวอย่างสำหรับกรอกข้อมูล ดังภาพที่ 4-2



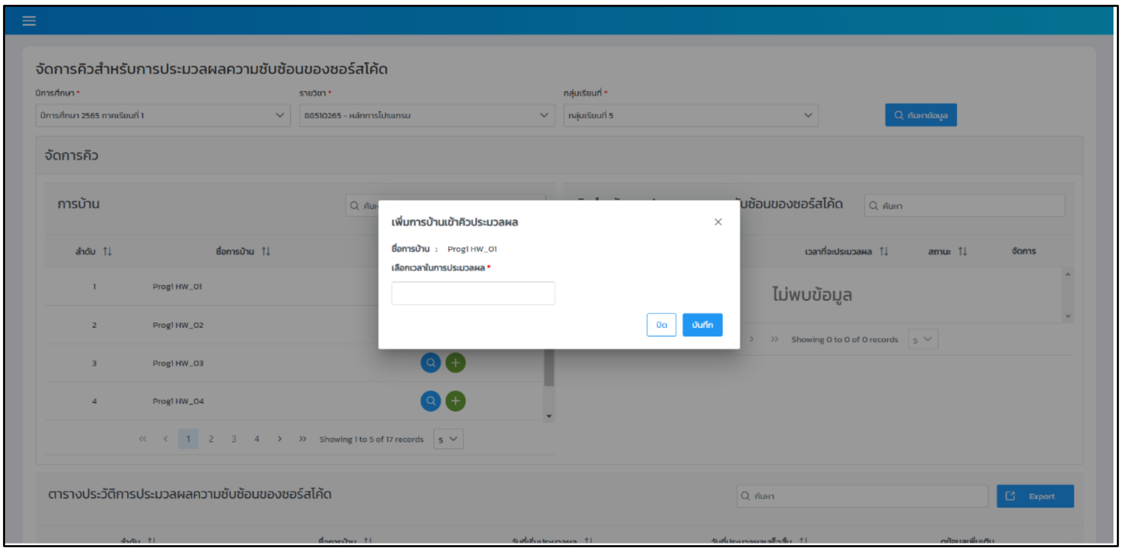
ภาพที่ 4-2 หน้าจอกรอกข้อมูลจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด

จากภาพที่ 4-2 ในกรณีที่อาจารย์กรอกข้อมูลครบถ้วน ให้อาจารย์ทำการกดปุ่ม "ค้นหาข้อมูล" ระบบจะแสดงข้อมูลการจัดการคิวตามที่อาจารย์กรอกข้อมูลก่อนทำการค้นหาข้อมูล ดังภาพที่ 4-3

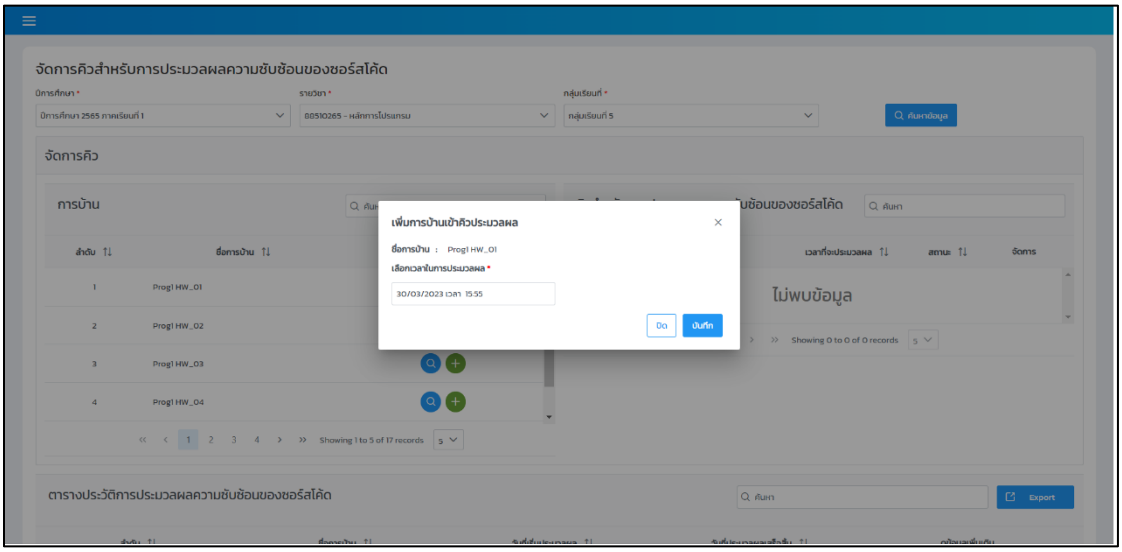


ภาพที่ 4-3 หน้าจอแสดงข้อมูลจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด

จากนั้นอาจารย์สามารถเพิ่มข้อมูลเข้าไปในคิวได้ โดยการกดปุ่มสัญลักษณ์ “เครื่องหมายบวก” ระบบจะแสดงหน้าต่างแสดงผลซ้อนสำหรับเพิ่มคิว ดังภาพที่ 4-4 และมีตัวอย่างสำหรับกรอกข้อมูลดังภาพที่ 4-5 ตามลำดับ

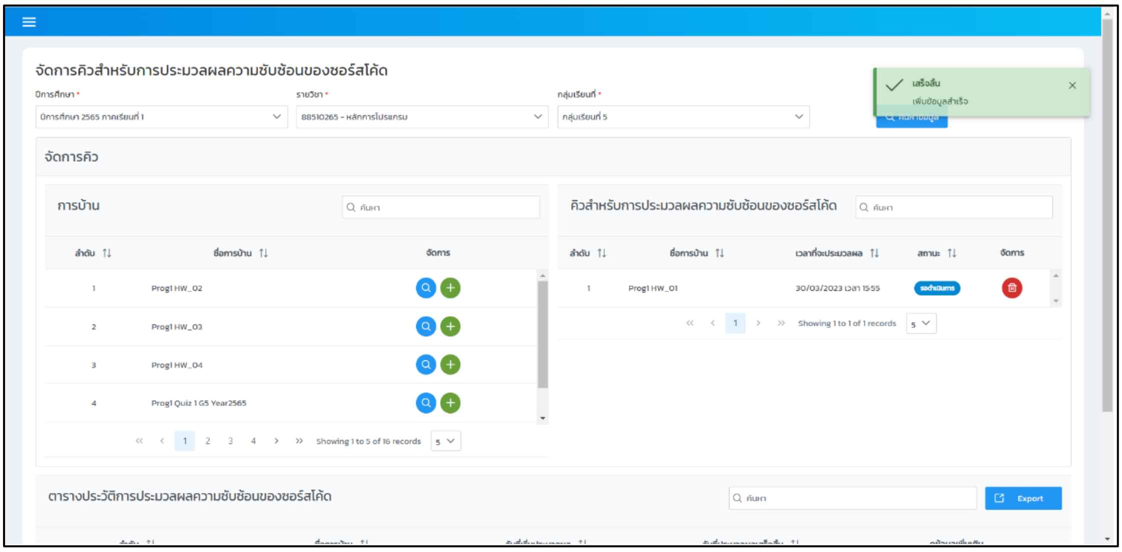


ภาพที่ 4-4 หน้าจอเพิ่มคิว



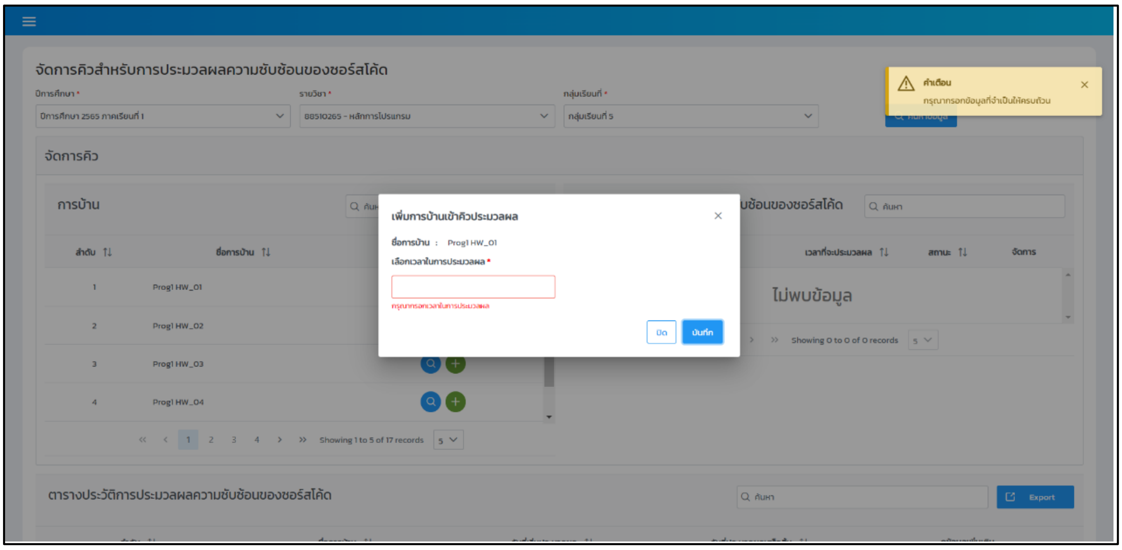
ภาพที่ 4-5 หน้าจอการกรอกข้อมูลเพิ่มคิว

เมื่ออาจารย์กรอกข้อมูลครบถ้วน แล้วอาจารย์ทำการกดปุ่ม “บันทึก” ระบบจะทำการบันทึกข้อมูลของการบ้านที่เลือกลงในตารางคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ดเพื่อรอทำการประมวลผลตามเวลาที่ตั้งไว้ ดังภาพที่ 4-6



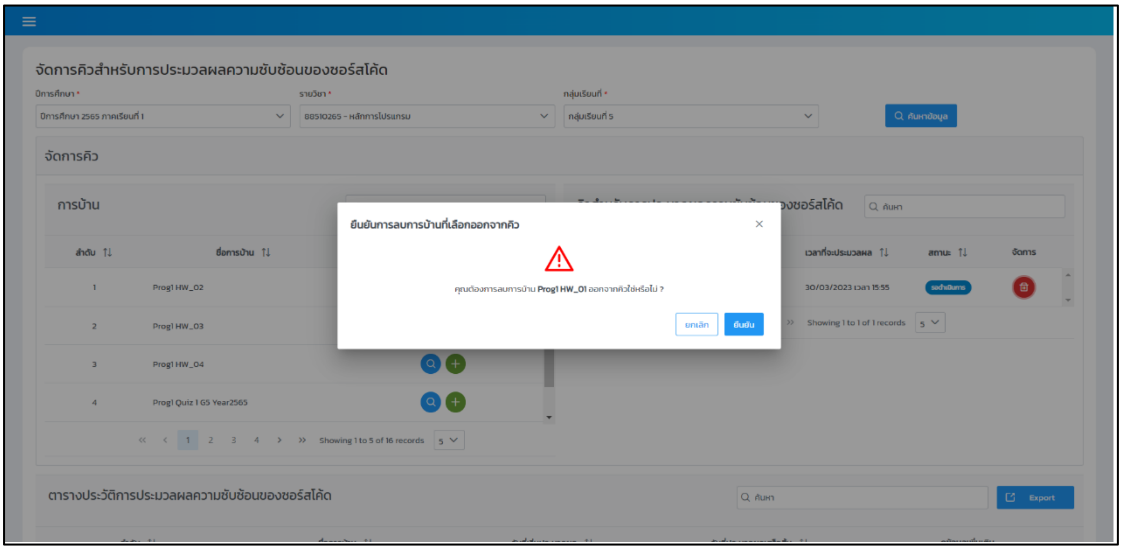
ภาพที่ 4-6 หน้าจอจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ดหลังการเพิ่มข้อมูล

จากภาพที่ 4-6 ในกรณีที่อาจารย์กรอกข้อมูลไม่ครบถ้วนแล้วอาจารย์ทำการกดปุ่ม “บันทึก” ระบบจะแสดงการแจ้งเตือนว่า “กรุณากรอกข้อมูลที่จำเป็นให้ครบถ้วน” และมีการแสดงข้อความสีแดงกำกับในส่วนที่ยังไม่ได้ทำการกรอกข้อมูล ดังภาพที่ 4-7



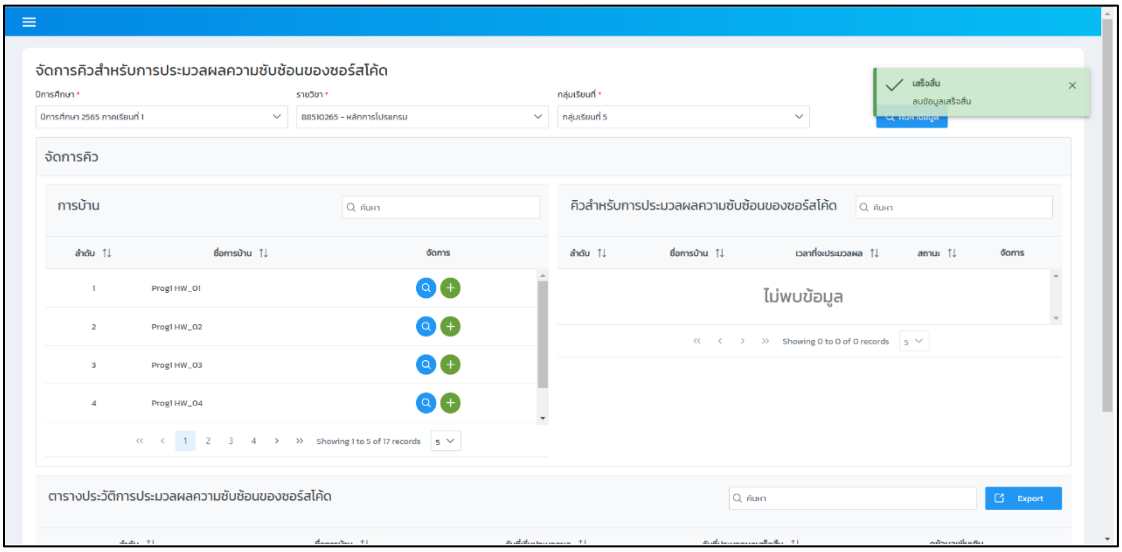
ภาพที่ 4-7 หน้าจอการกรอกข้อมูลเพิ่มคิวไม่ครบถ้วน

เมื่ออาจารย์ต้องการลบข้อมูลในคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด สามารถลบข้อมูลออกจากคิวได้ โดยการกดปุ่มสัญลักษณ์ “ถังขยะ” ระบบจะแสดงหน้าต่างแสดงผลซ้อนสำหรับยืนยัน ดังภาพที่ 4-8



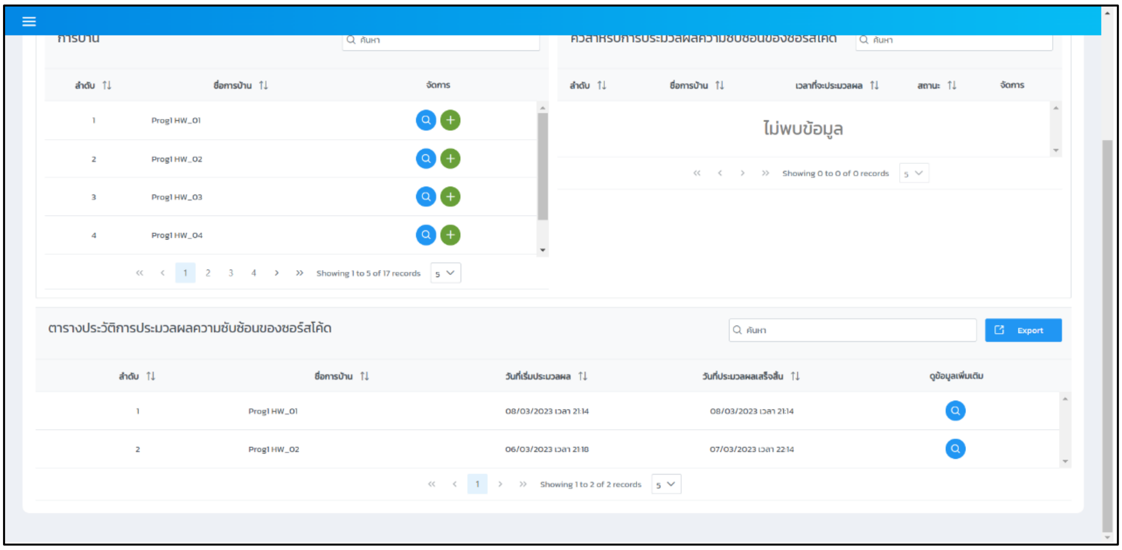
ภาพที่ 4-8 หน้าจอลบคิว

หากอาจารย์ยืนยันที่จะลบข้อมูล ให้อาจารย์ทำการกดปุ่ม “ยืนยัน” ระบบจะทำการลบข้อมูลของการบ้านที่เลือกออกจากตารางคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด ดังภาพที่ 4-9



ภาพที่ 4-9 หน้าจอยืนยันการลบคิว

เมื่ออาจารย์ต้องการตรวจสอบประวัติการประมวลผลความซับซ้อนของซอร์สโค้ด สามารถตรวจสอบได้จากตารางประวัติการประมวลผลความซับซ้อนของซอร์สโค้ดทางด้านล่างของหน้าจอจัดการคิวสำหรับการประมวลผลความซับซ้อนของซอร์สโค้ด ดังภาพที่ 4-10



ภาพที่ 4-10 หน้าจอแสดงตารางประวัติการประมวลผลความซับซ้อนของซอร์สโค้ด

4.1.2 การประมวลผลค่าความซับซ้อนของซอร์สโค้ด

เป็นฟังก์ชันสำหรับดึงข้อมูลตำแหน่งที่เก็บของซอร์สโค้ด โดยระบบจะส่ง API ไปยัง Back-end เพื่อดึงข้อมูลตำแหน่งที่เก็บของซอร์สโค้ด รหัสผู้ใช้งาน (Userid) รหัสโจทย์ปัญหา (Problemid) ลำดับคิว (Queueid) และรหัสงานที่มอบหมาย (Assigmentid) และเป็นฟังก์ชันสำหรับประมวลผลความซับซ้อนของซอร์สโค้ด และบันทึกค่าความซับซ้อนลงในฐานข้อมูล โดยระบบจะทำการวัดค่าความซับซ้อนของซอร์สโค้ดด้วย 2 มาตรวัด ได้แก่

1) Cyclomatic Complexity

การวัดค่าความซับซ้อนของการใช้คำสั่งควบคุมในโปรแกรม ได้แก่ คำสั่ง If, While, Repeat และ For ถ้าโปรแกรมมีการใช้คำสั่งควบคุมเป็นจำนวนมาก จะทำให้ยากต่อคุณภาพในด้านความสามารถการทดสอบ (Testabiliy) ความสามารถในการทำความเข้าใจ (Understandability) และความสามารถในการบำรุงรักษาซอฟต์แวร์ ซึ่งจะได้ผลลัพธ์ Cyclomatic Complexity Number (CCN) หรือมาตรวัดไซโคลเมตริกของแมคเคบ โดยเป็นการนับจำนวนความซับซ้อนไซโคลเมตริกของแมคเคบ

2) Chidamber and Kemerer Java Metrics (CKJM)

การคำนวณเมตริกเชิงวัตถุของ Chidamber และ Kemerer โดยการประมวลผล bytecode ของไฟล์ Java ที่คอมไพล์แล้ว โปรแกรมจะคำนวณสำหรับแต่ละคลาส ซึ่งจะได้ผลลัพธ์จำนวน 6 เมตริก ดังนี้

1) มาตรวัดจำนวนเมธอดต่อคลาส (Weighted Methods per Class: WMC) คือ จำนวนเมธอดต่อคลาสที่ใช้ในการพัฒนาซอฟต์แวร์ รวมถึงแก้ไข และบำรุงรักษาซอฟต์แวร์

2) มาตรวัดความรับผิดชอบของคลาส (Response For a Class: RFC) คือ จำนวนเมธอดภายในคลาสที่สามารถตอบสนอง การเรียกใช้งานจากคลาสอื่นได้ หรือจากเมธอดบางตัวภายในคลาสนั้น

3) มาตรวัดระดับความลึกของการสืบทอดคุณสมบัติของคลาส (Depth of Inheritance Hierarchy: DIT) คือ ค่าระดับความลึกของการสืบทอดคุณสมบัติ (Inheritance) ของแต่ละคลาส

4) มาตรวัดจำนวนคลาสลูก (Number of Children: NOC) คือ จำนวนคลาสลูกทั้งหมดที่สืบทอดลงมาจากคลาสแม่

5) มาตรวัดการเข้าคู่กันระหว่างวัตถุ (Coupling Between Objects: CBO) คือ ค่าระดับความเกี่ยวข้องหรือพึ่งพากันของ Module ต่างๆ สำหรับซอฟต์แวร์เชิงวัตถุ

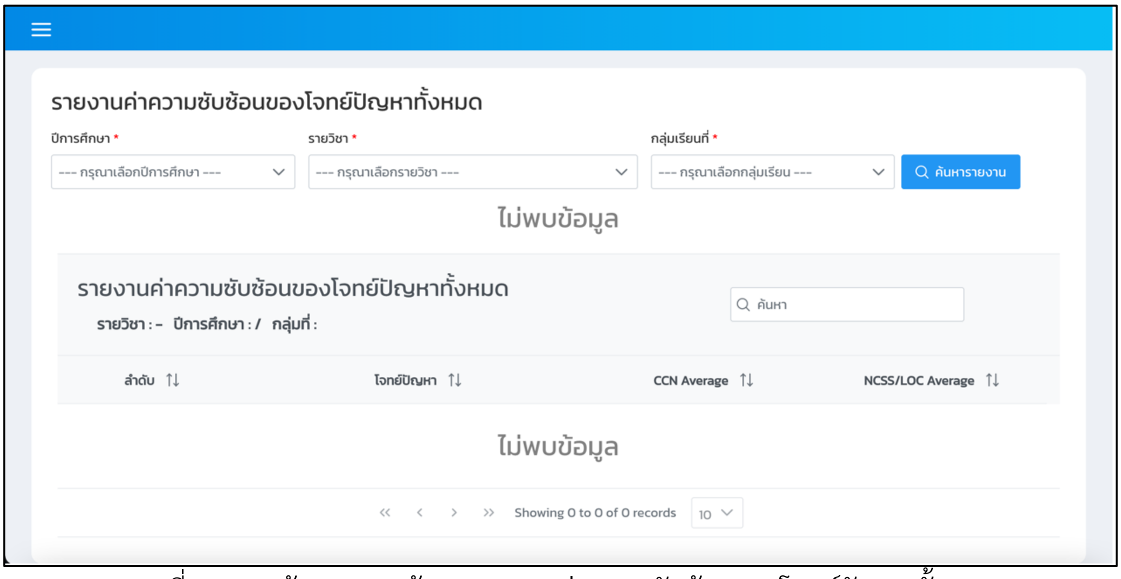
6) มาตรวัดระดับของการขาดความสัมพันธ์ภายในคลาส (Lack of Cohesion of Methods: LCOM) คือ ค่าขาดการเกาะกันเป็นก้อน (Cohesion) ของคลาส ว่าเมธอดของคลาสนั้นมีความสัมพันธ์กับเมธอดอื่นๆ อย่างไร

4.1.3 รายงานค่าความซับซ้อนของซอร์สโค้ด

การทำงานของมอดูลรายงานค่าความซับซ้อนของโจทย์ปัญหา มีวัตถุประสงค์เพื่อให้อาจารย์สามารถดูรายงานค่าความซับซ้อนของโจทย์ปัญหาที่ได้จากการวัดความซับซ้อนของซอร์สโค้ดที่นักศึกษาได้ทำการส่งเข้ามาในระบบโดยใช้มาตรวัด Cyclomatic และมาตรวัดซอฟต์แวร์เชิงวัตถุ ซึ่งมอดูลนี้ประกอบไปด้วย 5 รายงานดังนี้

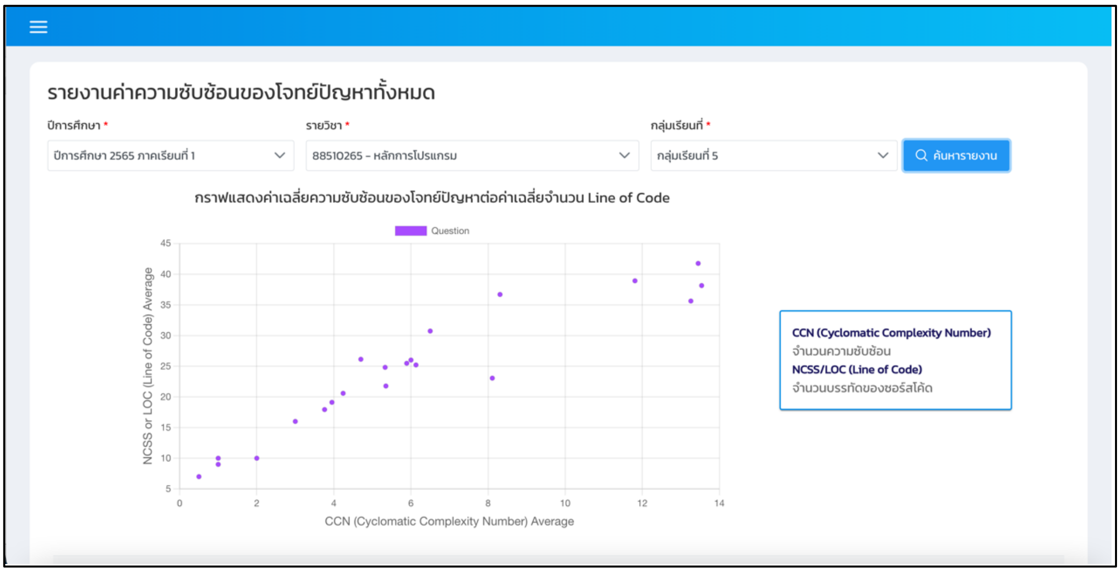
1) รายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด

เป็นฟังก์ชันแสดงผลรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมดในแต่ละรายวิชา ซึ่งอาจารย์สามารถดูรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมดได้ โดยเลือกเมนู “รายงาน” และเลือกเมนู “รายงานค่าความซับซ้อนด้วยเมตริก Cyclomatic” จากนั้นจึงเลือกเมนู “รายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด” ซึ่งระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด ดังภาพที่ 4-11

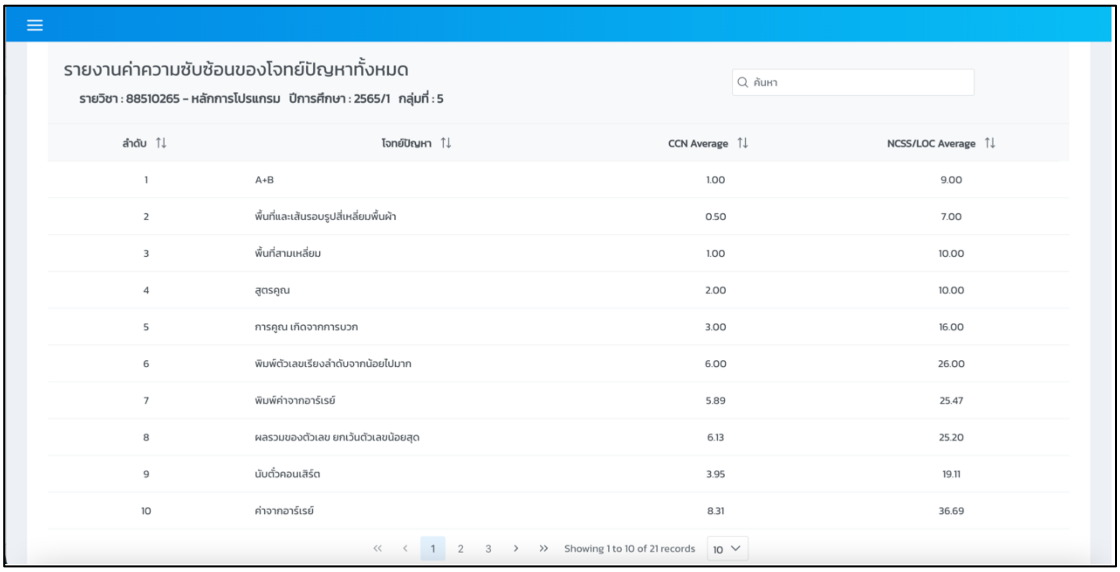


ภาพที่ 4-11 หน้าจอกรอกข้อมูลรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด

จากภาพที่ 4-11 อาจารย์จะต้องทำการกรอกข้อมูลที่จำเป็นให้ครบถ้วนตามลำดับจากซ้ายไปขวา เมื่อเลือกปีการศึกษา รายวิชา และกลุ่มเรียนที่ต้องการแล้ว ให้ทำการกดที่ปุ่ม “ค้นหารายงาน” ระบบจะแสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหาต่อค่าเฉลี่ยจำนวน Line of Code และแสดงตารางรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด ดังภาพที่ 4-12 และภาพที่ 4-13



ภาพที่ 4-12 หน้าจอแสดงกราฟรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด

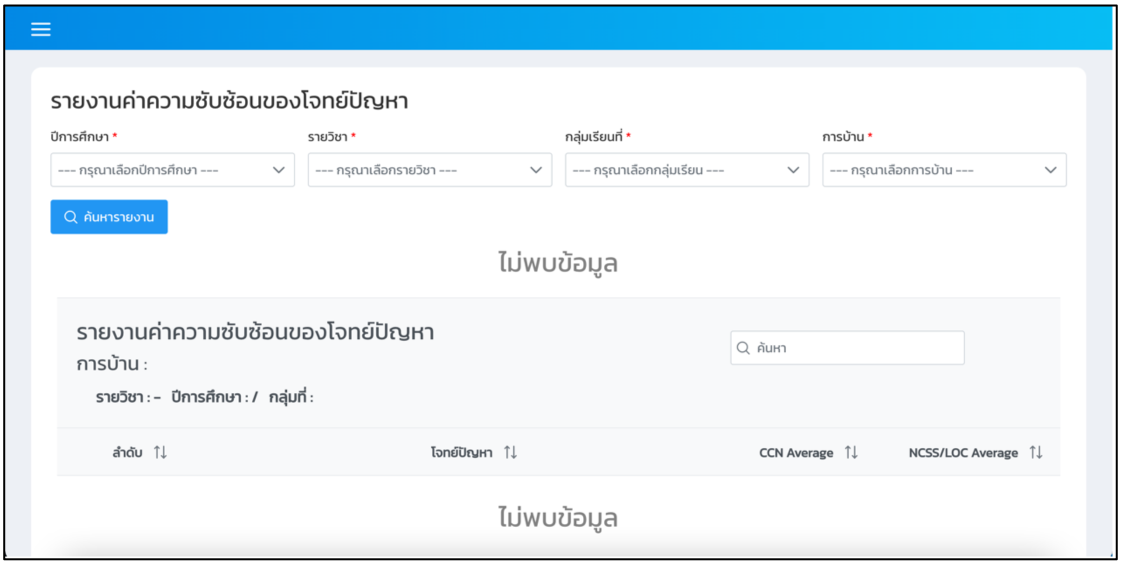


ลำดับ	โจทย์ปัญหา	CCN Average	NCSS/LOC Average
1	A+B	1.00	9.00
2	พื้นที่และเส้นรอบรูปสี่เหลี่ยมพื้นผ้า	0.50	7.00
3	พื้นที่สามเหลี่ยม	1.00	10.00
4	สูตรคูณ	2.00	10.00
5	การคูณ เกิดจากการบวก	3.00	16.00
6	พิมพ์ตัวเลขเรียงลำดับจากน้อยไปมาก	6.00	26.00
7	พิมพ์ค่าจากอาร์เรย์	5.89	25.47
8	ผลรวมของตัวเลข ยกเว้นตัวเลขน้อยสุด	6.13	25.20
9	นับตั๋วคอนเสิร์ต	3.95	19.11
10	ค่าจากอาร์เรย์	8.31	36.69

ภาพที่ 4-13 หน้าจอแสดงตารางรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด

2) รายงานค่าความซับซ้อนของโจทย์ปัญหา

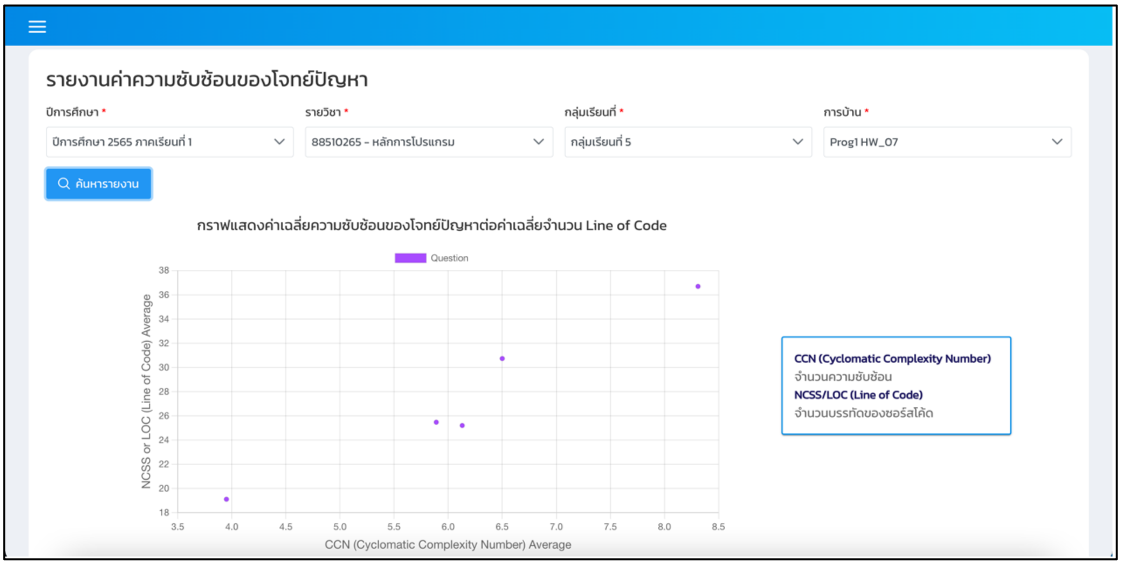
เป็นฟังก์ชันแสดงผลรายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมดในแต่ละการบ้าน ซึ่งอาจารย์สามารถดูรายงานค่าความซับซ้อนของโจทย์ปัญหาได้ โดยเลือกเมนู “รายงาน” และเลือกเมนู “รายงานค่าความซับซ้อนด้วยเมตริก Cyclomatic” จากนั้นจึงเลือกเมนู “รายงานค่าความซับซ้อนของโจทย์ปัญหา” ซึ่งระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหา ดังภาพที่ 4-14



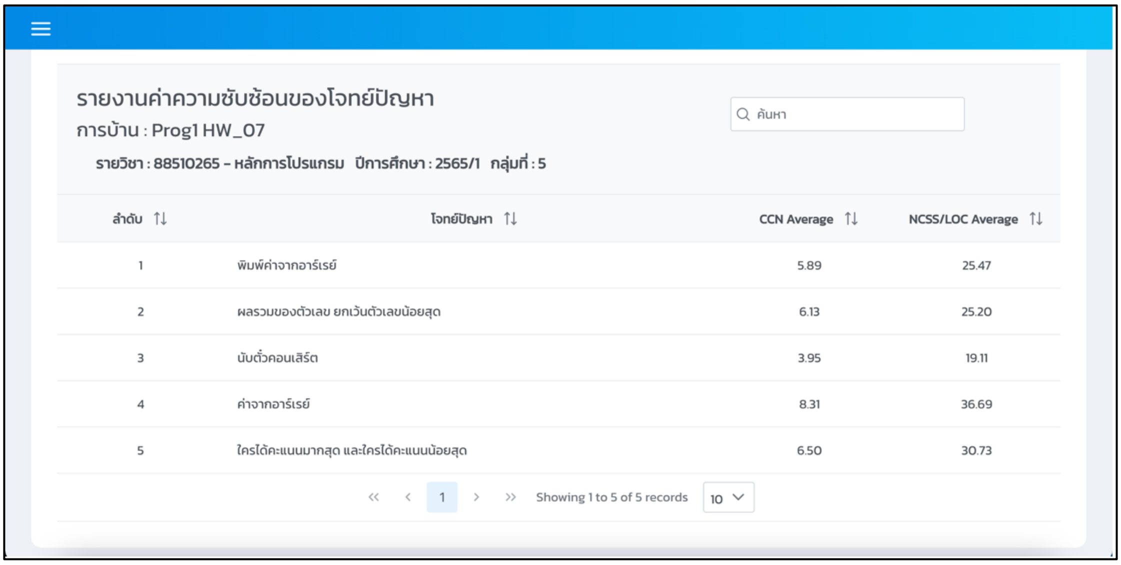
ภาพที่ 4-14 หน้าจอกรอกข้อมูลรายงานค่าความซับซ้อนของโจทย์ปัญหา

จากภาพที่ 4-14 อาจารย์จะต้องทำการกรอกข้อมูลที่จำเป็นให้ครบถ้วนตามลำดับ จากซ้ายไปขวา เมื่อเลือกปีการศึกษา รายวิชา กลุ่มเรียนและการบ้านที่ต้องการแล้ว

ให้ทำการกดที่ปุ่ม "ค้นหารายงาน" ระบบจะแสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหาต่อค่าเฉลี่ยจำนวน Line of Code และแสดงตารางรายงานค่าความซับซ้อนของโจทย์ปัญหา ดังภาพที่ 4-15 และภาพที่ 4-16



ภาพที่ 4-15 หน้าจอแสดงกราฟรายงานค่าความซับซ้อนของโจทย์ปัญหา


รายงานค่าความซับซ้อนของโจทย์ปัญหา

การบ้าน : Prog1 HW_07

รายวิชา : 88510265 – หลักการโปรแกรม ปีการศึกษา : 2565/1 กลุ่มที่ : 5

ค้นหา

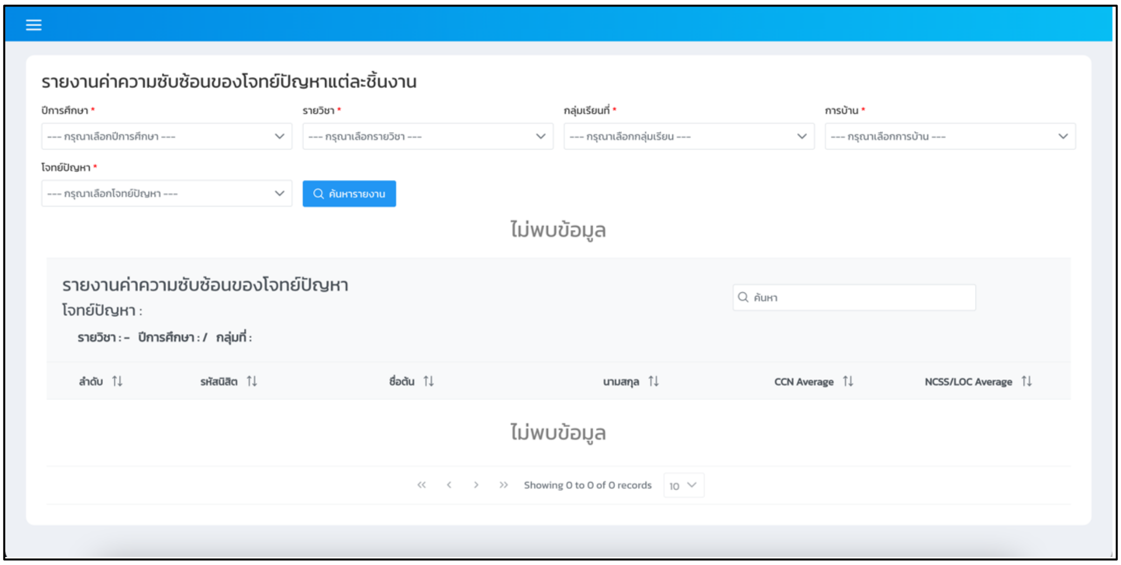
ลำดับ	โจทย์ปัญหา	CCN Average	NCSS/LOC Average
1	พิมพ์ค่าจากอาร์เรย์	5.89	25.47
2	ผลรวมของตัวเลข ยกเว้นตัวเลขน้อยสุด	6.13	25.20
3	นับตัวคอนเสิร์ต	3.95	19.11
4	ค่าจากอาร์เรย์	8.31	36.69
5	ใครได้คะแนนมากสุด และใครได้คะแนนน้อยสุด	6.50	30.73

Showing 1 to 5 of 5 records

ภาพที่ 4-16 หน้าจอแสดงตารางรายงานค่าความซับซ้อนของโจทย์ปัญหา

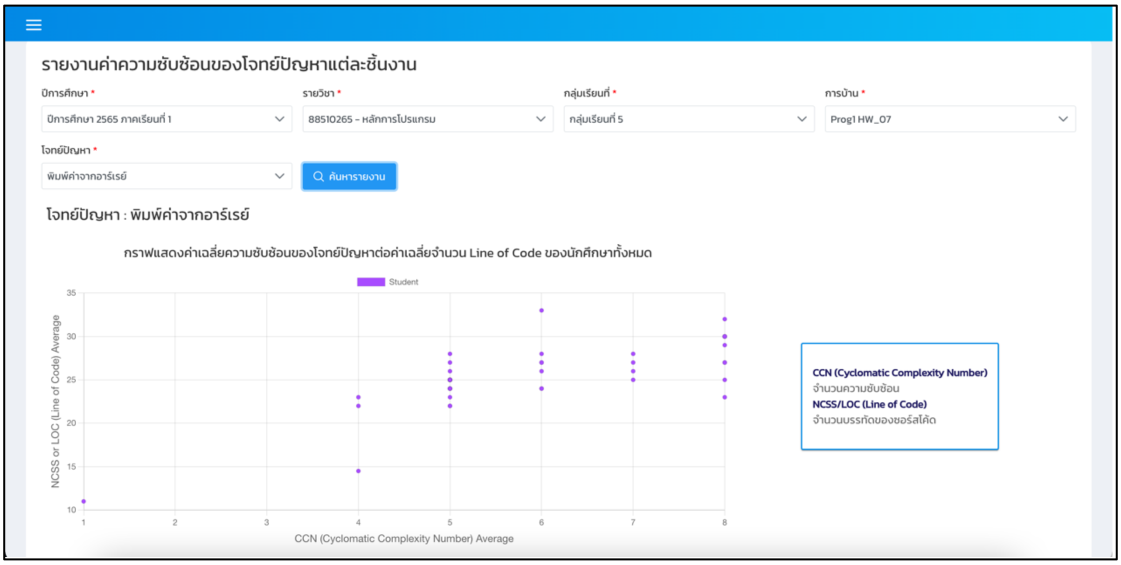
3) ฟังก์ชันแสดงผลรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน

เป็นฟังก์ชันแสดงผลรายงานค่าความซับซ้อนของโจทย์ปัญหา ซึ่งอาจารย์สามารถดูรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงานได้ โดยเลือกเมนู "รายงาน" และเลือกเมนู "รายงานค่าความซับซ้อนด้วยเมตริก Cyclomatic" จากนั้นจึงเลือกเมนู "รายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน" ซึ่งระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน ดังภาพที่ 4-17

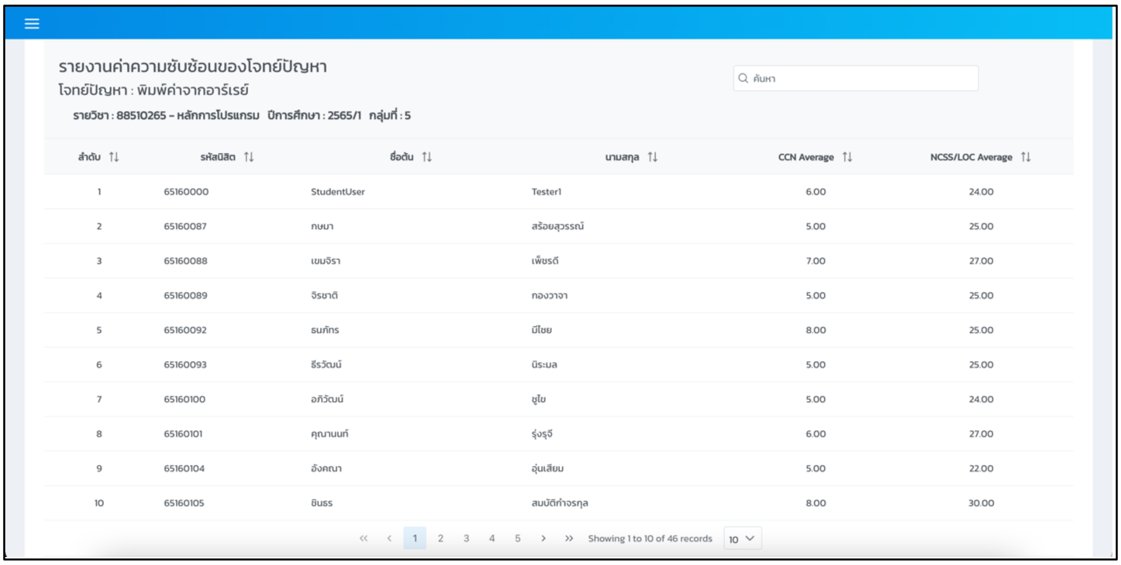


ภาพที่ 4-17 หน้าจอกรอกข้อมูลรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน

จากภาพที่ 4-17 อาจารย์จะต้องทำการกรอกข้อมูลที่จำเป็นให้ครบถ้วนตามลำดับจากซ้ายไปขวา เมื่อเลือกปีการศึกษา รายวิชา กลุ่มเรียน การบ้าน และโจทย์ปัญหาที่ต้องการแล้ว ให้ทำการกดที่ปุ่ม “ค้นหารายงาน” ระบบจะแสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหาต่อค่าเฉลี่ยจำนวน Line of Code และแสดงตารางรายงานค่าความซับซ้อนของโจทย์ปัญหา ดังภาพที่ 4-18 และภาพที่ 4-19



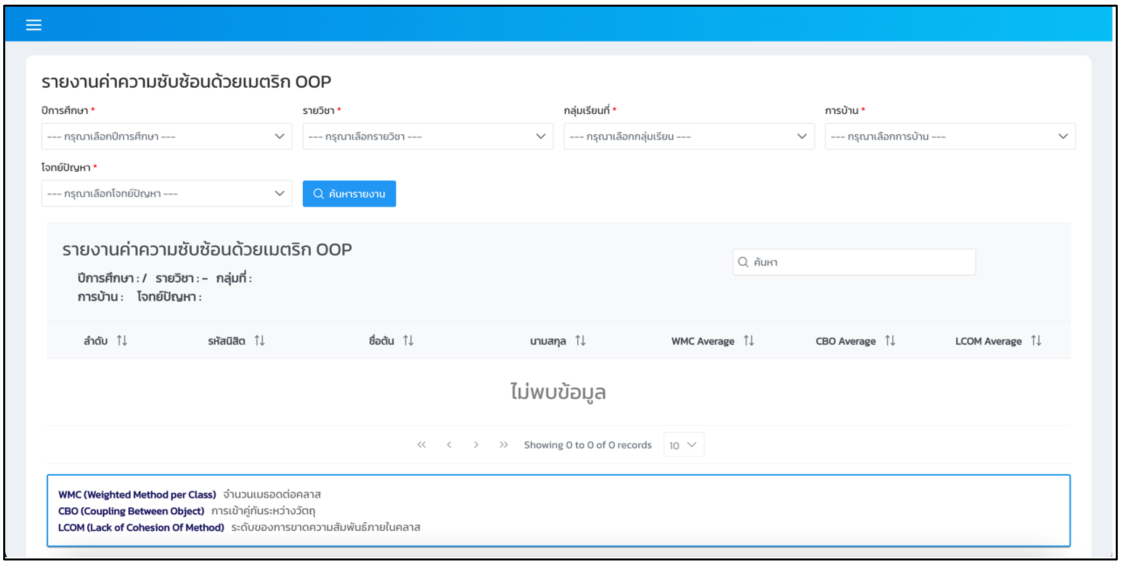
ภาพที่ 4-18 หน้าจอแสดงกราฟรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน



ภาพที่ 4-19 หน้าจอแสดงตารางรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละชิ้นงาน

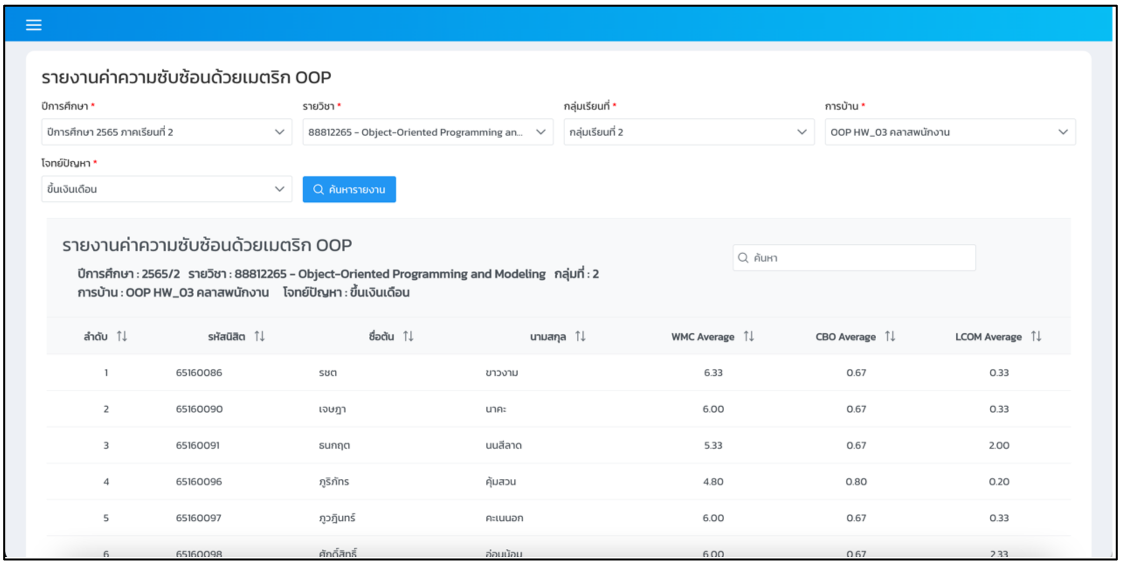
4) ฟังก์ชันแสดงผลรายงานค่าความซับซ้อนด้วยเมตริก OOP

เป็นฟังก์ชันแสดงผลรายงานค่าความซับซ้อนด้วยเมตริก OOP ซึ่งอาจารย์สามารถดูรายงานค่าความซับซ้อนของโจทย์ปัญหาด้วยเมตริก OOP ได้ โดยเลือกเมนู “รายงาน” จากนั้นเลือกเมนู “รายงานค่าความซับซ้อนด้วยเมตริก OOP” ซึ่งระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนด้วยเมตริก OOP ดังภาพที่ 4-20



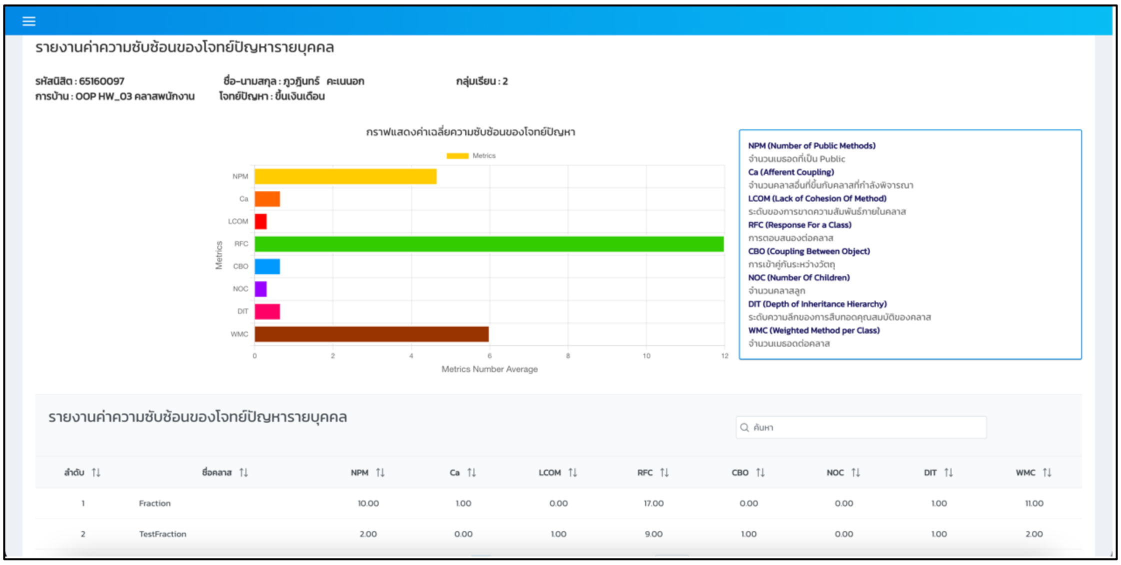
ภาพที่ 4-20 หน้าจอกรอกข้อมูลรายงานค่าความซับซ้อนด้วยเมตริกเชิงวัตถุ

จากภาพที่ 4-20 อาจารย์จะต้องทำการกรอกข้อมูลที่จำเป็นให้ครบถ้วนตามลำดับจากซ้ายไปขวา เมื่อเลือกปีการศึกษา รายวิชา กลุ่มเรียน การบ้าน และโจทย์ปัญหาที่ต้องการแล้ว ให้ทำการกดที่ปุ่ม “ค้นหารายงาน” ระบบจะแสดงตารางรายงานค่าความซับซ้อนด้วยเมตริก OOP ดังภาพที่ 4-21



ภาพที่ 4-21 หน้าจอแสดงรายงานค่าความซับซ้อนด้วยเมตริกเชิงวัตถุ

จากภาพที่ 4-21 อาจารย์สามารถเลือกรายการนักศึกษาที่ต้องการดูรายงานค่าความซับซ้อนของโจทย์ปัญหา ระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาของนักศึกษาที่เลือก ดังภาพที่ 4-22

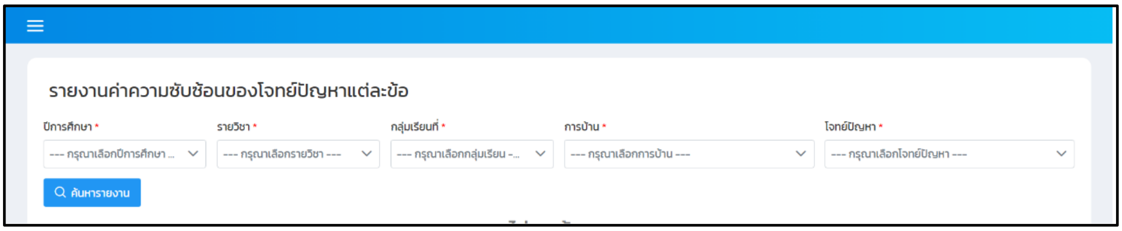


ภาพที่ 4-22 หน้าจอแสดงรายงานค่าความซับซ้อนของโจทย์ปัญหารายบุคคล

5) รายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละข้อ

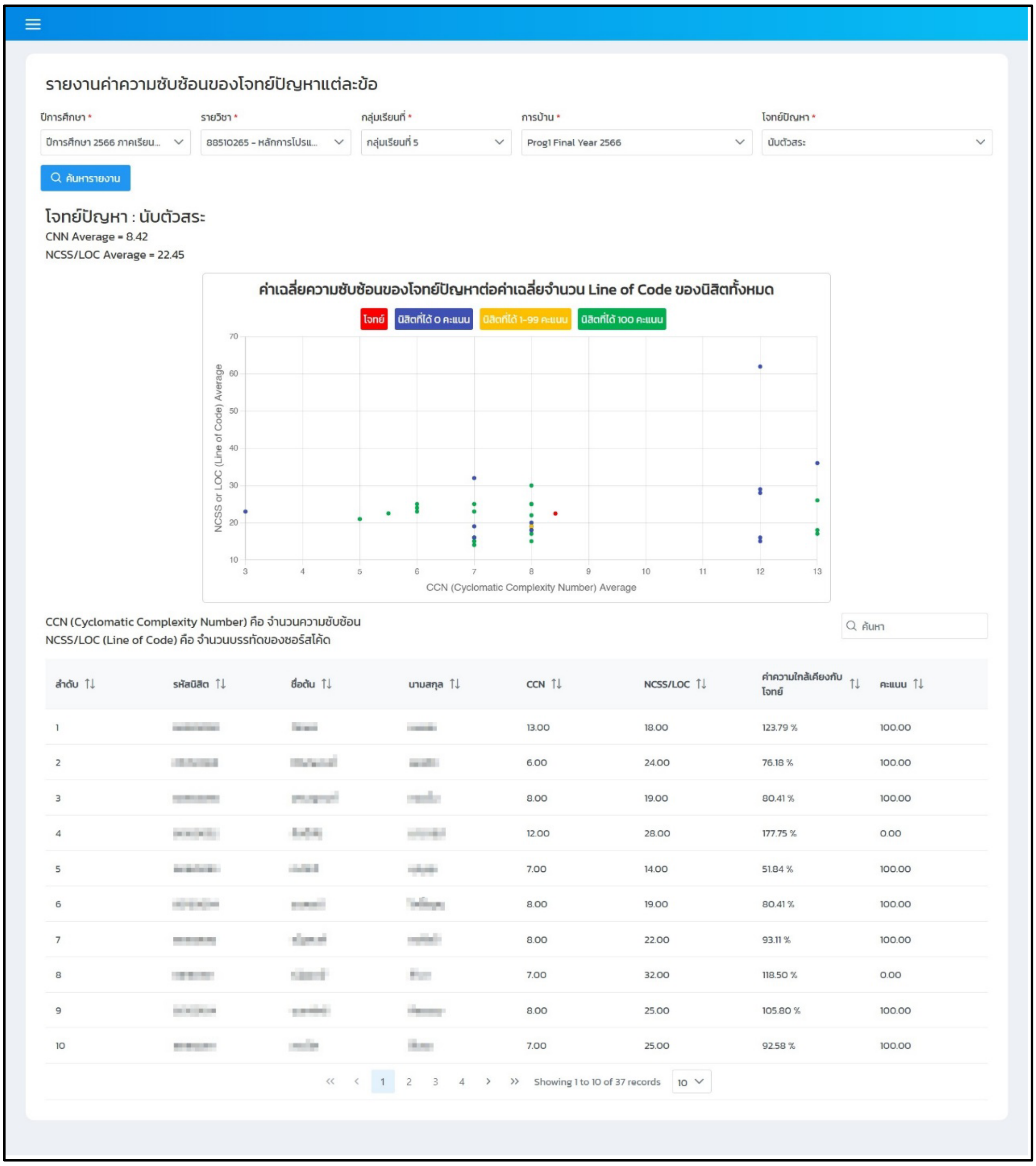
การทำงานของมอดูลดูรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละข้อ มีวัตถุประสงค์เพื่อให้อาจารย์สามารถดูรายงานค่าความซับซ้อนของโจทย์ปัญหาที่ได้จากการวัดค่าความซับซ้อนของซอร์สโค้ดที่นักศึกษาได้ทำการส่งเข้ามาในระบบเปรียบเทียบกับค่าความซับซ้อนของโจทย์ปัญหา โดยใช้มาตรวัด Cyclomatic Complexity เพื่อนำผลการตรวจความซับซ้อนของซอร์สโค้ดที่ผู้เรียนส่งงานมานั้นมาเปรียบเทียบว่ามีความใกล้เคียงกับข้อมูลพื้นฐาน (Baseline) หรือไม่ สำหรับใช้ในการ

ประเมินความสามารถของผู้เรียนหรือใช้ทบทวนรายละเอียดของโจทย์ปัญหาที่มอบหมายแก่ผู้เรียน โดยผู้ใช้งานสามารถดูรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละข้อ โดยเลือกเมนู “รายงาน” และเลือกเมนู “รายงานค่าความซับซ้อนด้วยเมตริก Cyclomatic” จากนั้นเลือกเมนู “รายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละข้อ” ซึ่งระบบจะแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละข้อ ดังภาพที่ 4-23



ภาพที่ 4-23 หน้าจอกรอกข้อมูลรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละข้อ

จากภาพที่ 4-23 ผู้ใช้งานจะต้องทำการกรอกข้อมูลที่จำเป็นให้ครบถ้วนก่อน โดยจะประกอบไปด้วย ปีการศึกษา รายวิชา กลุ่มเรียน การบ้าน และโจทย์ปัญหาที่ต้องการ จากนั้นให้ทำการกดที่ปุ่ม “ค้นหารายงาน” ระบบจะแสดงกราฟแสดงค่าเฉลี่ยความซับซ้อนของโจทย์ปัญหา ค่าเฉลี่ยจำนวน Line of Code และตารางรายงานค่าความซับซ้อนของโจทย์ปัญหา ดังภาพที่ 4-24



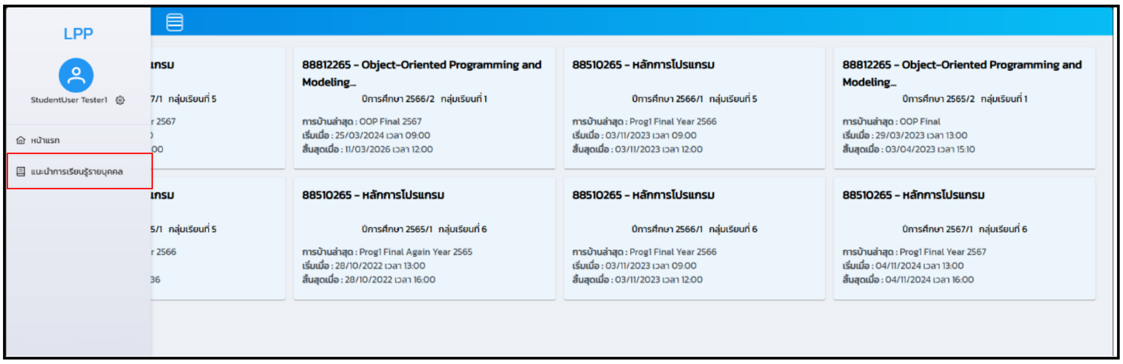
ภาพที่ 4-24 หน้าจอแสดงหน้าจอรายงานค่าความซับซ้อนของโจทย์ปัญหาแต่ละข้อ

4.2 มอดูลการเรียนรู้แบบปรับได้

ในส่วนของมอดูลการทำโจทย์ปัญหาตามความสามารถของนักศึกษานั้นเป็นกลไกสำคัญที่ใช้การวิเคราะห์และประเมินความสามารถปัจจุบันของนักศึกษาโดยการมอบโจทย์ที่เหมาะสม โดยช่วยส่งเสริมการเรียนรู้ ช่วยลดความเครียดและความเบื่อหน่ายของนักศึกษาโดยการจัดความยากของโจทย์ให้สอดคล้องกับศักยภาพของนักศึกษาเพื่อให้เหมาะสมกับเป้าหมายการเรียนรู้ในระยะยาวและช่วยให้นักศึกษาพัฒนาได้อย่างต่อเนื่องตามลำดับขั้นของความรู้ดังต่อไปนี้

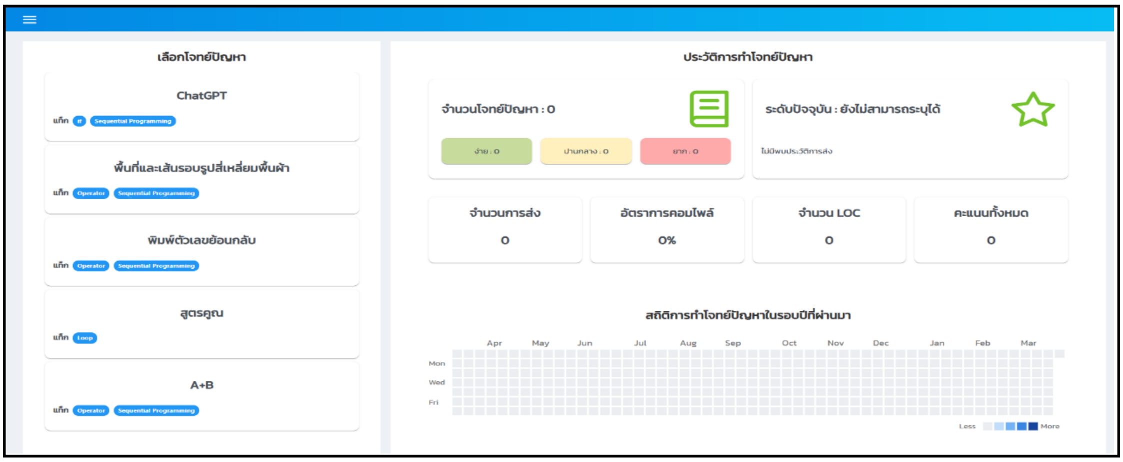
4.2.1 การแนะนำการเรียนรู้รายบุคคล

นักศึกษาสามารถดำเนินการทำโจทย์ประเมินความรู้ของตนเองได้โดยคลิกที่เมนูแนะนำการเรียนรู้รายบุคคลเพื่อแสดงหน้าจอแนะนำการเรียนรู้รายบุคคลที่แสดงถึงข้อมูลในการทำโจทย์ปัญหาตามความสามารถของนักศึกษาในรูปแบบข้อความและกราฟแสดงผลแสดงดังภาพที่ 4-25



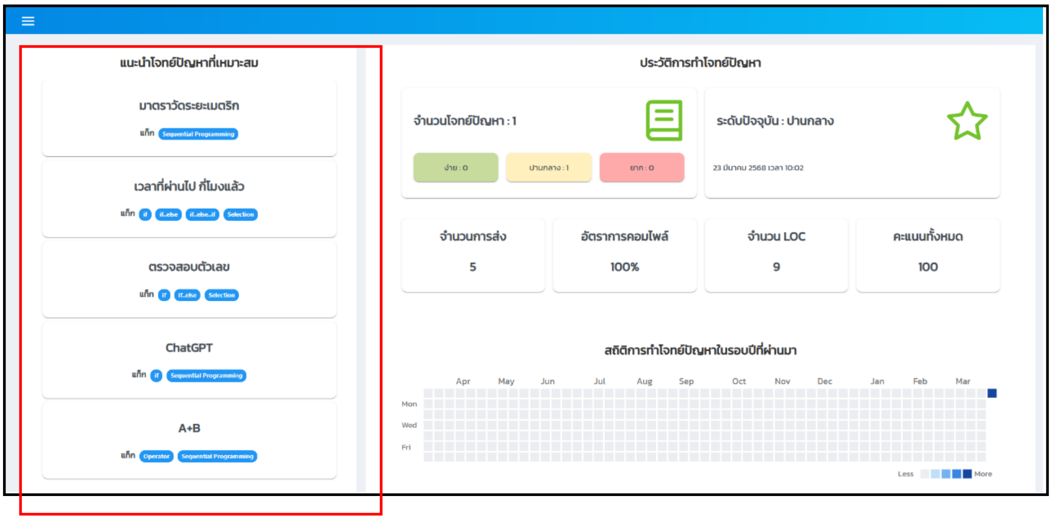
ภาพที่ 4-25 หน้าจอการเลือกเมนูแนะนำการเรียนรู้รายบุคคล

เมื่อนักศึกษาเข้าสู่หน้าจอแนะนำการเรียนรู้รายบุคคลระบบจะแสดงข้อมูลการทำโจทย์ปัจจุบัน โดยจะแสดงรายการโจทย์ปัญหาที่มีความซับซ้อนระดับปานกลางให้นักศึกษาเลือกกรณีที่นักศึกษาเข้าใช้งานระบบในครั้งแรก โดยจะแสดงชื่อโจทย์และแท็ก แสดงดังภาพที่ 4-26



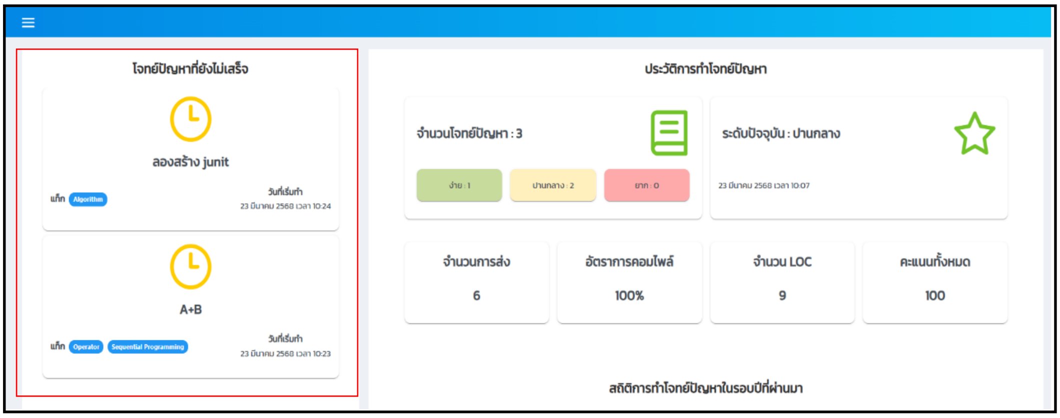
ภาพที่ 4-26 หน้าจอแนะนำการเรียนรู้รายบุคคลส่วนของการแสดงโจทย์ปัญหา

กรณีที่นักศึกษาทำโจทย์ปัญหาเสร็จแล้ว และนักศึกษาเข้าสู่หน้าจอแนะนำการเรียนรู้รายบุคคลระบบจะแสดงข้อมูลแนะนำโจทย์ปัญหาถัดไปที่เหมาะสม โดยจะแสดงชื่อโจทย์และแท็กแสดงดังภาพที่ 4-27



ภาพที่ 4-27 หน้าจอแสดงแนะนำโจทย์ปัญหาถัดไปที่เหมาะสม

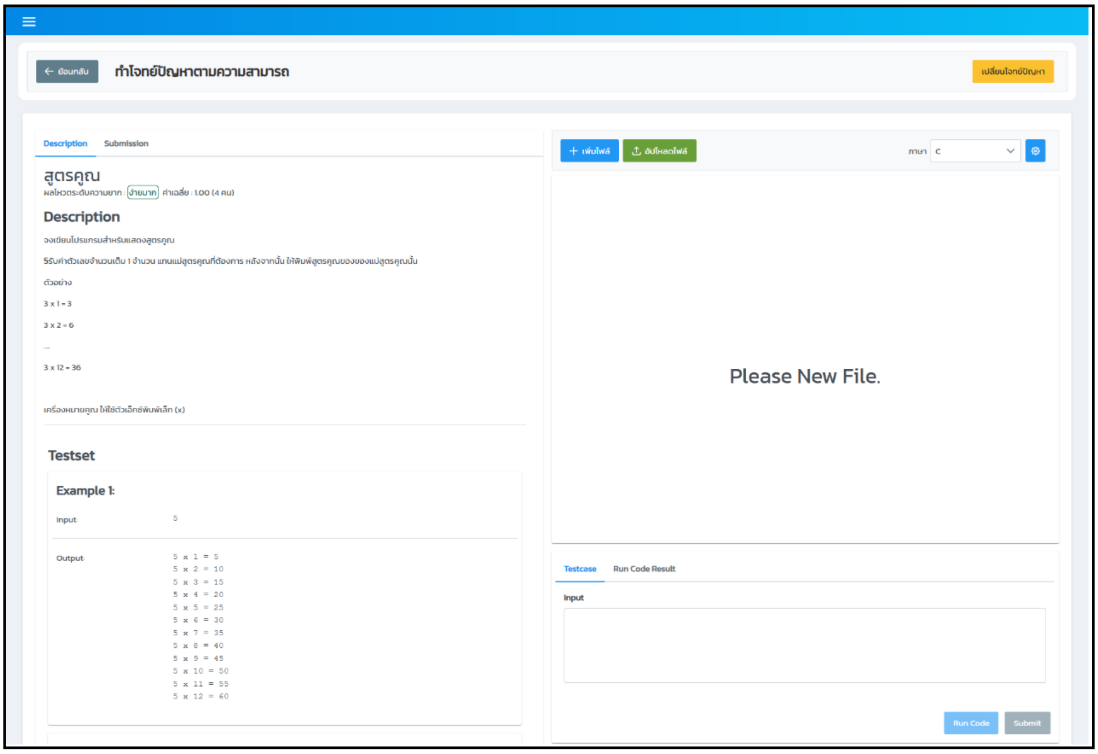
กรณีที่นักศึกษาเคยเลือกทำโจทย์ปัญหาแล้วและนักศึกษาเข้าสู่หน้าจอแนะนำการเรียนรู้รายบุคคลระบบจะแสดงข้อมูลการทำโจทย์ปัจจุบัน โดยจะแสดงรายการโจทย์ปัญหาที่นักศึกษายังทำไม่สำเร็จ ชื่อโจทย์ แท็ก และวันเวลาที่เริ่มทำโจทย์ปัญหา แสดงดังภาพที่ 4-28



ภาพที่ 4-28 หน้าจอแสดงโจทย์ปัญหาที่ยังไม่เสร็จ

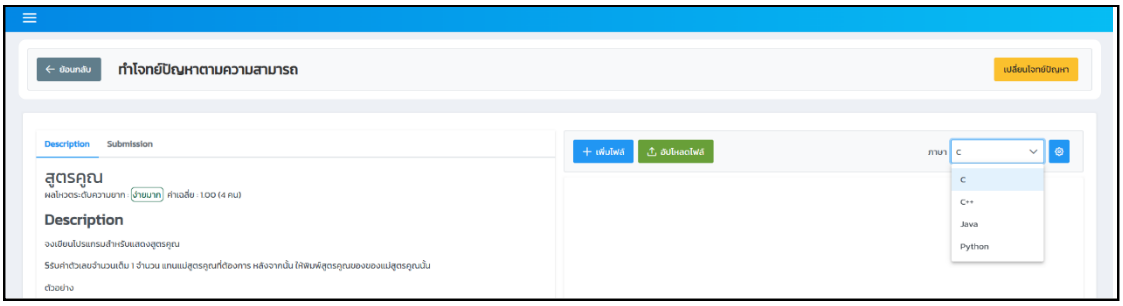
4.2.2 การทำโจทย์ปัญหาการเรียนรู้แบบปรับได้

หลังจากนักศึกษาเลือกโจทย์ปัญหา จะแสดงหน้าจอการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้โดยมีองค์ประกอบของหน้าจอได้แก่ชื่อโจทย์ แสดงชื่อของโจทย์ปัญหาที่นักศึกษาเลือก คำอธิบายรายละเอียดของโจทย์ปัญหา ชุดข้อมูลทดสอบที่ใช้ในการตรวจสอบความถูกต้องของโค้ดที่นักศึกษาเขียน แท็บแสดงซอร์สโค้ด แท็บแสดงผลการรันซอร์สโค้ด แสดงผลลัพธ์ที่ได้จากการรันโค้ด รวมถึงข้อผิดพลาด และตัวเลือกภาษาผ่านตัวเลือกหลายรายการแสดงดังภาพที่ 4-29

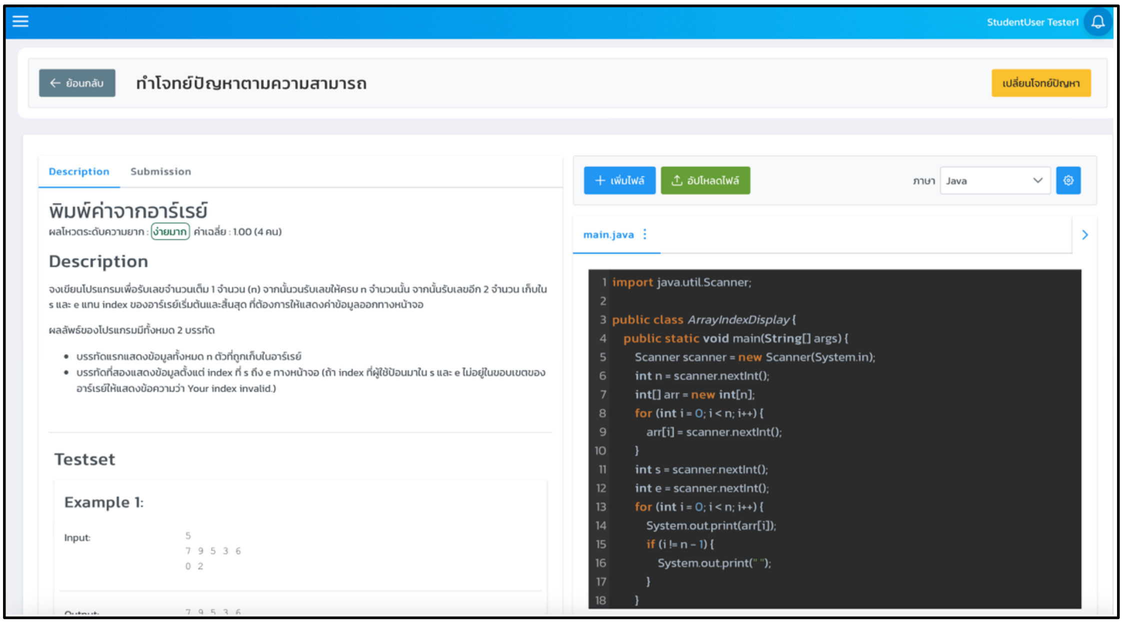


ภาพที่ 4-29 หน้าจอการทำโจทย์ประเมินตามความรู้ของนักศึกษา

นักศึกษาสามารถดำเนินการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้โดยการพัฒนาและส่งไฟล์ซอร์สโค้ด (Source code) ผ่านระบบ ซึ่งมีตัวเลือกให้สามารถอัปโหลดไฟล์จากเครื่องของตนเองหรือสร้างไฟล์ขึ้นใหม่ภายในระบบ โดยกระบวนการนี้ครอบคลุมหลายขั้นตอน ตั้งแต่การรับไฟล์ที่นักศึกษาส่งเข้ามาการคอมไพล์โค้ดเพื่อแปลงเป็นรูปแบบที่สามารถทำงานได้ รวมถึงการตรวจสอบข้อผิดพลาดที่อาจเกิดขึ้นในการคอมไพล์ หลังจากนั้น ระบบจะทำการทดสอบโค้ดที่ได้รับโดยเปรียบเทียบผลลัพธ์ที่ได้กับคำตอบที่คาดหวัง โดยแสดงดังภาพที่ 4-30 ถึงภาพที่ 4-31

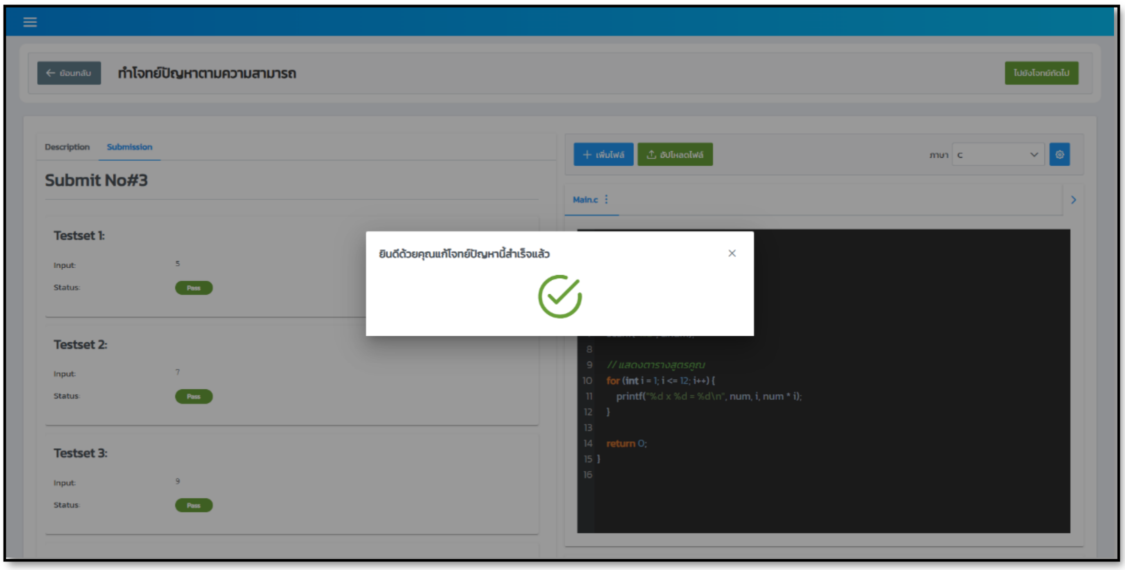


ภาพที่ 4-30 หน้าจอการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้ส่วนของการเลือกภาษา

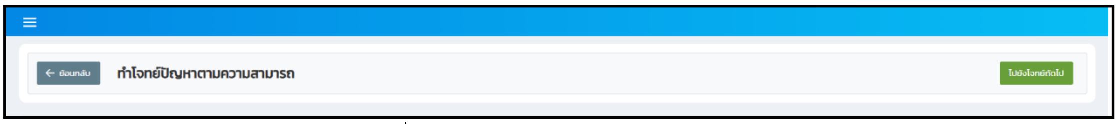


ภาพที่ 4-31 หน้าจอการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้

หลังจากที่นักศึกษาทำโจทย์ปัญหาสำเร็จและส่งคำตอบเรียบร้อย หากผลลัพธ์ถูกต้องทั้งหมดตาม Test set จะแสดงกล่องแจ้งเตือนยินดีด้วยคุณแก้โจทย์ปัญหานี้สำเร็จแล้วพร้อมแสดงข้อมูลผลการทดสอบตาม Test set และจะแสดงปุ่มไปยังโจทย์ถัดไปเพื่อไปยังหน้าจอแนะนำการเรียนรู้รายบุคคลเพื่อแสดงรายการโจทย์ถัดไปที่เหมาะสมกับระดับความรู้ของนักศึกษาแสดงดังภาพที่ 4-32 ถึงภาพที่ 4-33



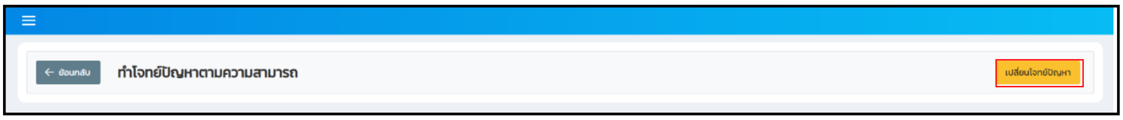
ภาพที่ 4-32 หน้าจอการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้สำเร็จ



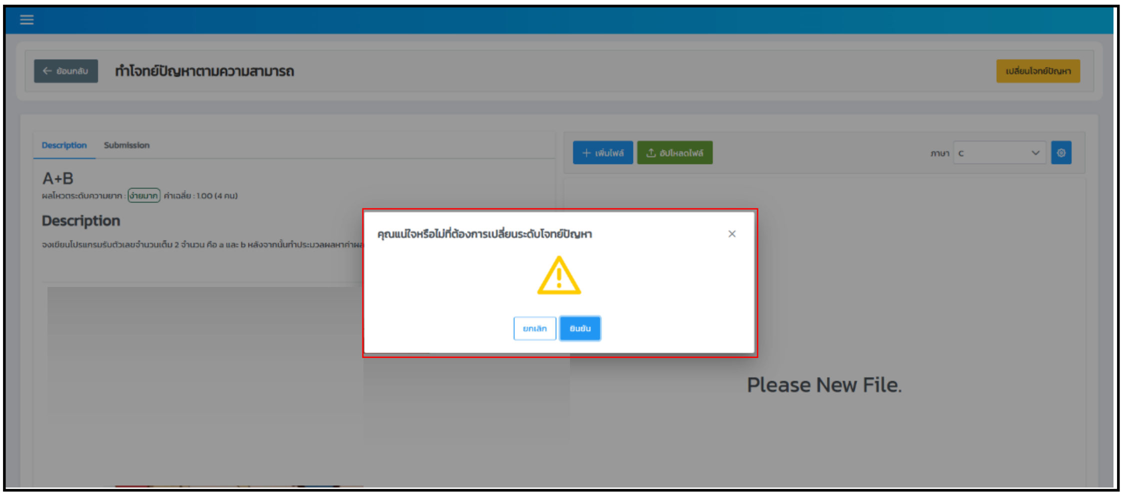
ภาพที่ 4-33 หน้าจอการคลิกปุ่มไปยังโจทย์ถัดไป

4.2.3 การเปลี่ยนโจทย์ปัญหาการเรียนรู้แบบปรับได้

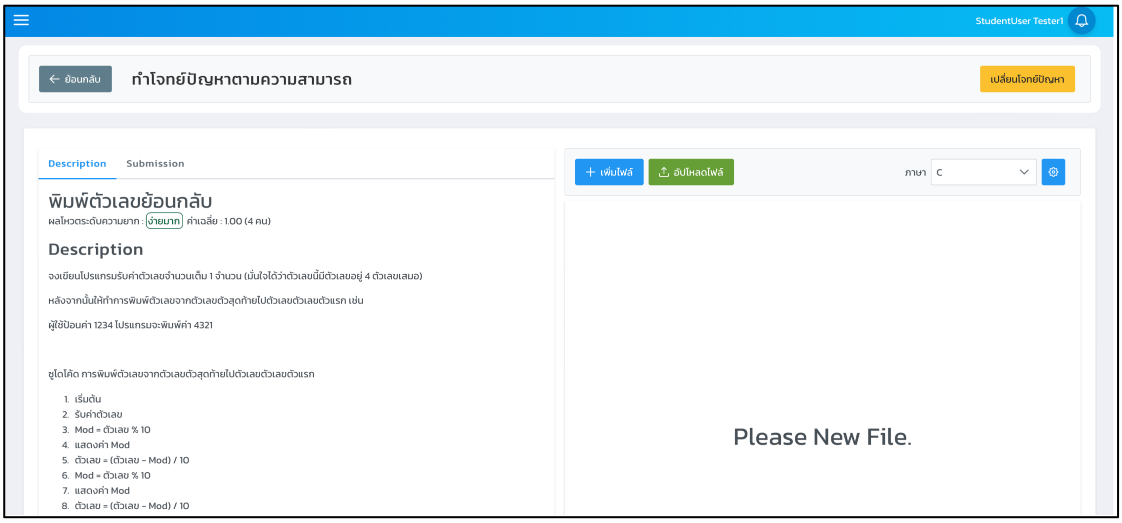
หากนักศึกษาพบว่าโจทย์ปัญหาที่ได้รับมอบหมายมีความซับซ้อนเกินกว่าระดับความสามารถของตนเอง หรือเนื้อหาในโจทย์ไม่ตรงกับระดับความรู้และทักษะที่มีอยู่ในปัจจุบัน นักศึกษาสามารถเลือกเปลี่ยนโจทย์ปัญหาได้ ระบบได้รับการออกแบบมาเพื่อให้ความยืดหยุ่นกับผู้ใช้งาน โดยเปิดโอกาสให้นักศึกษาสามารถฝึกฝนและพัฒนาทักษะการเขียนโปรแกรมได้อย่างต่อเนื่อง โดยนักศึกษาสามารถกดเปลี่ยนโจทย์ปัญหาได้ ดังภาพที่ 4-34 ถึงภาพที่ 4-36



ภาพที่ 4-34 หน้าจอคลิกปุ่ม “เปลี่ยนโจทย์ปัญหา”



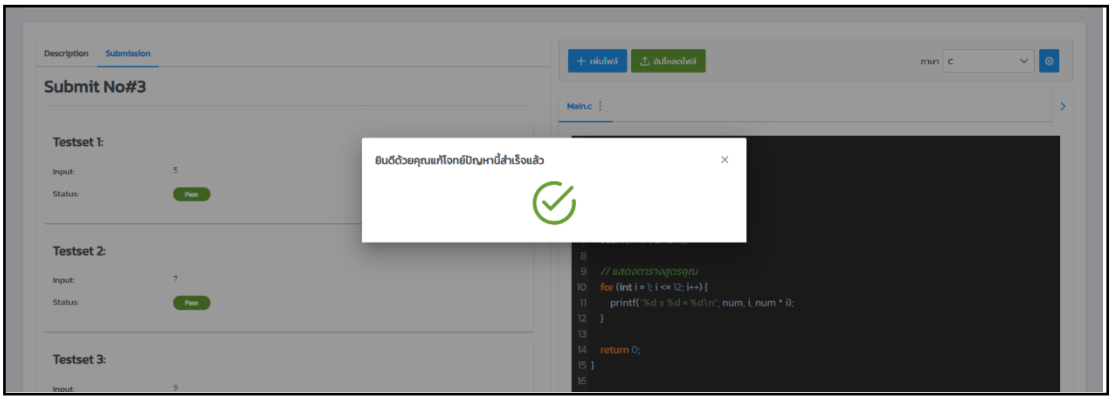
ภาพที่ 4-35 หน้าจอแสดงกล่องโต้ตอบยืนยันการเปลี่ยนโจทย์ปัญหา



ภาพที่ 4-36 หน้าจอแสดงโจทย์ปัญหาใหม่

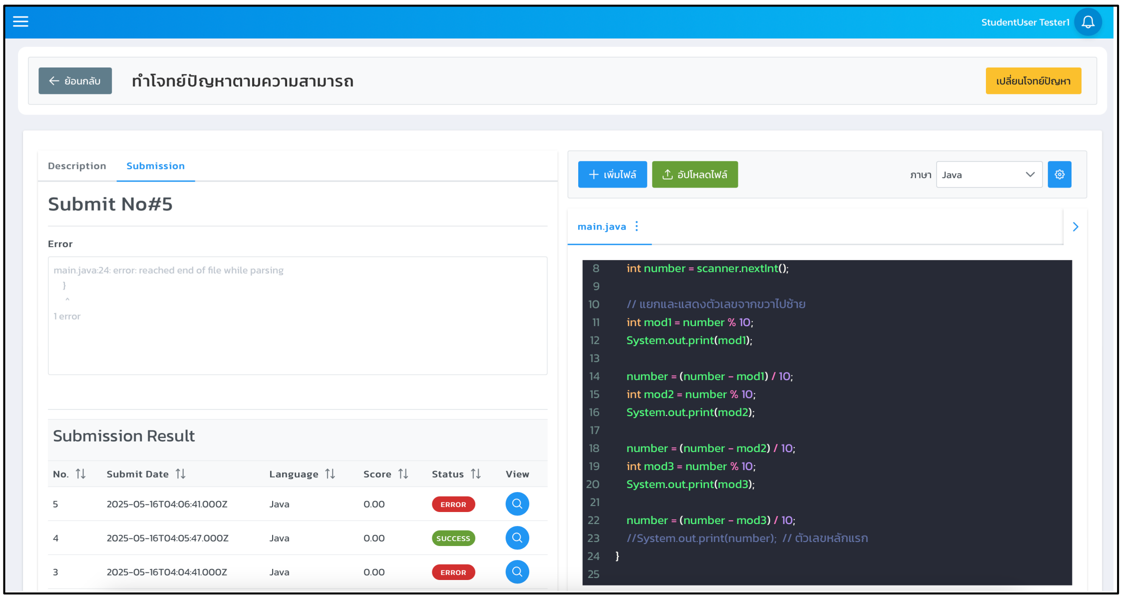
4.2.4 การส่งคำตอบโจทย์ปัญหาการเรียนรู้แบบปรับได้

หลังจากที่นักศึกษาทำโจทย์ปัญหาสำเร็จและส่งคำตอบเรียบร้อยหากผลการตรวจสอบกรณีทดสอบของโจทย์ปัญหานั้นถูกต้องทั้งหมดตาม Test set จะแสดงกล่องแจ้งเตือนยินดีด้วยคุณแก้โจทย์ปัญหานี้สำเร็จแล้วพร้อมแสดงข้อมูลผลเรียนรู้รายบุคคลเพื่อแสดงรายการโจทย์ถัดไปที่เหมาะสมกับระดับความรู้ของนักศึกษาแสดงดังภาพที่ 4-37



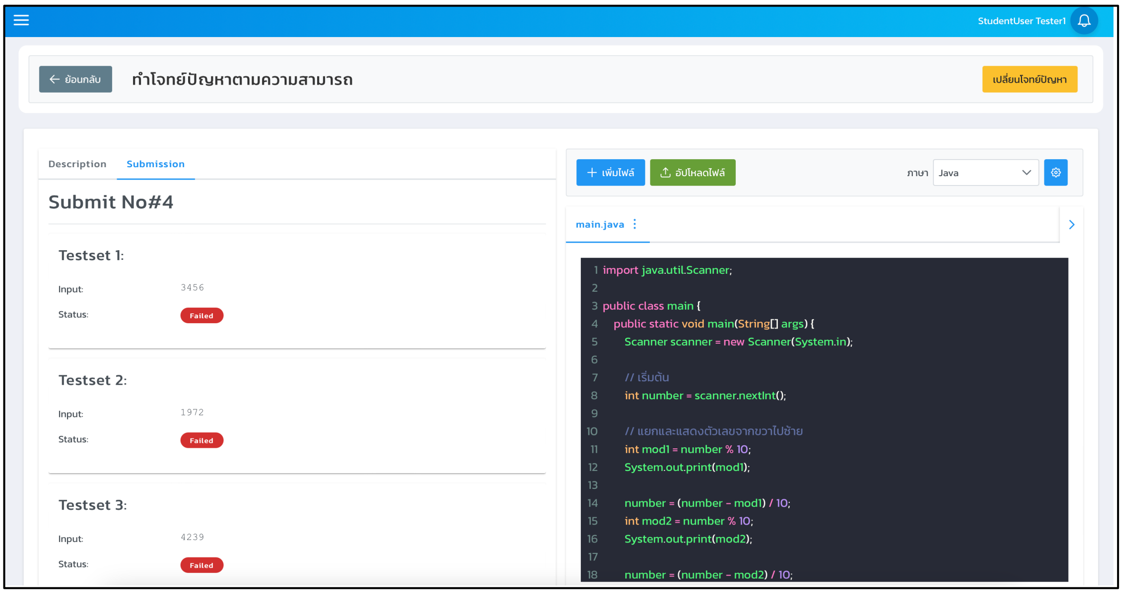
ภาพที่ 4-37 หน้าจอการทำโจทย์ปัญหาการเรียนรู้แบบปรับได้สำเร็จ

หากนักศึกษาส่งซอร์สโค้ดไม่ถูกต้อง กรณีที่นักศึกษาเขียนซอร์สโค้ดไม่เป็นไปตามหลักไวยากรณ์ เมื่อระบบทำการคอมไพล์ซอร์สโค้ดแล้วพบว่าซอร์สโค้ดไม่ถูกต้องจะทำการแจ้งข้อผิดพลาดที่เกิดขึ้นเพื่อให้นักศึกษาแก้ไขซอร์สโค้ด ดังภาพที่ 4-38



ภาพที่ 4-38 หน้าจอแสดงข้อผิดพลาดเมื่อคอมไพล์ไม่ผ่าน

หากนักศึกษาเขียนเขียนซอร์สโค้ดถูกต้องตามหลักไวยากรณ์แล้ว ระบบจะทำการคอมไพล์และตรวจสอบตามกรณีทดสอบตามโจทย์ปัญหาที่ได้กำหนดเอาไว้ ซึ่งผลการทดสอบแต่ละกรณีทดสอบจะแสดงดังภาพที่ 4-39

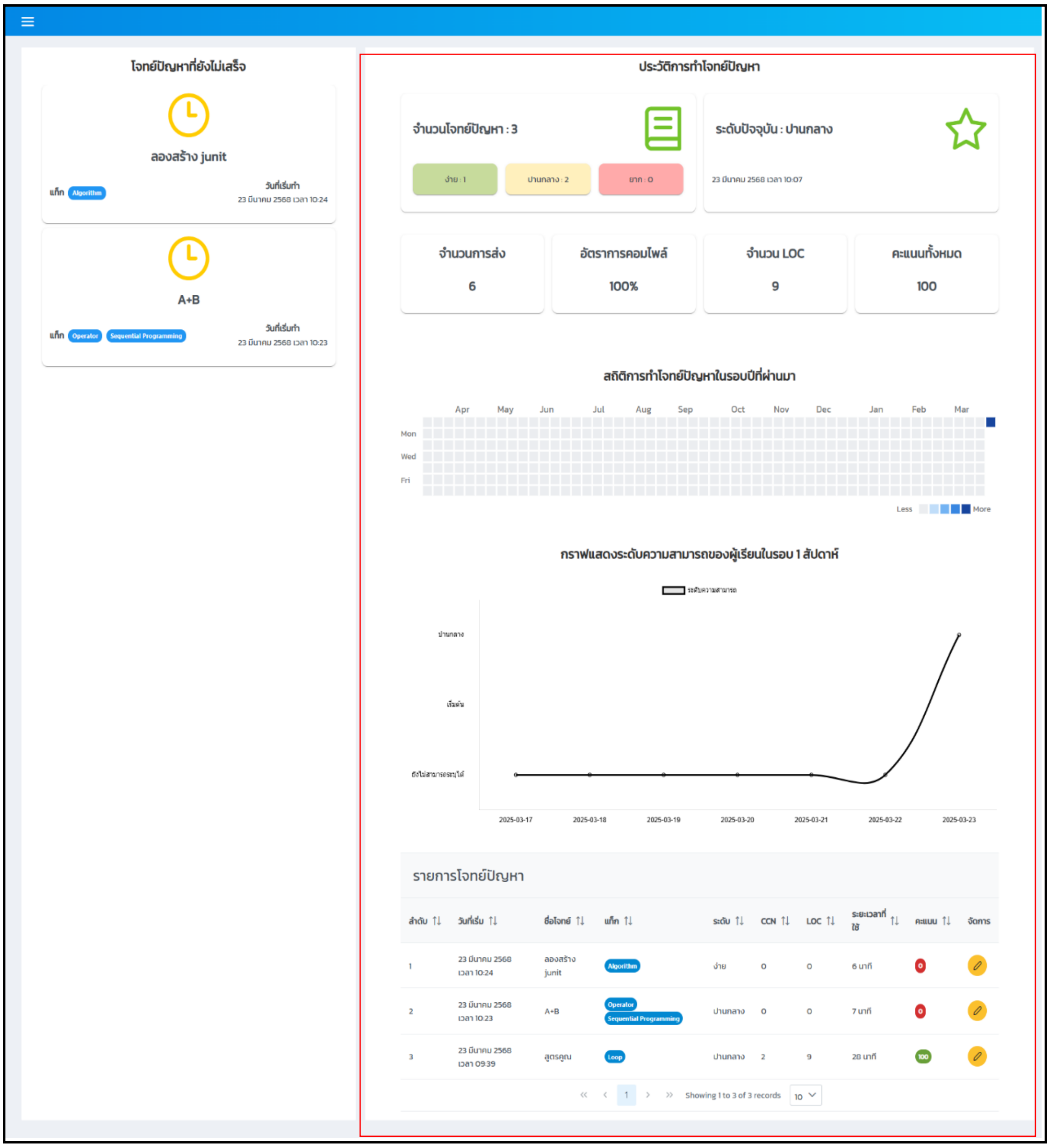


ภาพที่ 4-39 หน้าจอแสดงผลการตรวจสอบตามกรณีทดสอบ

4.2.5 การแสดงประวัติการเรียนรู้รายบุคคล

กรณีที่นักศึกษาเคยทำโจทย์ประเมินตามความรู้ของนักศึกษาแล้วการเข้าสู่หน้าจอแนะนำการเรียนรู้รายบุคคลระบบจะแสดงรายละเอียดของข้อมูลการทำโจทย์ปัจจุบันที่แสดงถึงจำนวนโจทย์ปัญหาที่นักศึกษาเคยทำ ระดับความสามารถปัจจุบัน จำนวนการส่ง อัตราการคอมไพล์ จำนวน LOC

ทั้งหมด คะแนนทั้งหมด กราฟ Heat Map ที่แสดงถึงสถิติความถี่ในการทำโจทย์ปัญหาในรอบปีที่ผ่านมา กราฟแสดงระดับความสามารถของผู้เรียนในรอบ 1 สัปดาห์ ตารางรายการโจทย์ปัญหาที่นักศึกษาเคยทำที่สามารถเข้าดูรายละเอียดรวมถึงการทำโจทย์เดิมซ้ำใหม่ได้อีกรอบ และแสดงรายการโจทย์ปัญหาที่นักศึกษายังทำไม่เสร็จให้นักศึกษาเลือกทำต่อ โดยแสดงข้อมูลชื่อโจทย์ แท็ก และวันที่เริ่มทำ ซึ่งหน้าจอประวัติการทำโจทย์ปัญหาแบบปรับได้แสดงดังภาพที่ 4-40



ภาพที่ 4-40 หน้าจอประวัติการเรียนรู้รายบุคคล

4.3 การประเมินผลประสิทธิภาพของโมเดล Random Forest

มอดูลการเรียนรู้แบบปรับได้นั้นได้ใช้โมเดล Random Forest ในการแนะนำระดับความยากของโจทย์ปัญหาที่เหมาะกับผู้เรียน ซึ่งงานวิจัยนี้ได้ประเมินประสิทธิภาพของโมเดลจากผลการฝึกสอน

โดยใช้โมเดลใช้เมตริกพื้นฐาน ได้แก่ ความแม่นยำ (Accuracy) ค่าเรียกคืน (Recall) ความเที่ยง (Precision) ค่าเฉลี่ยกลมกลืน (F1-Score) โดยเปรียบเทียบกับป้ายกำกับที่ผู้สอนกำหนดไว้ เพื่อวัดความสามารถของโมเดลในการทำนายระดับความยากที่เหมาะสมสำหรับผู้เรียน ซึ่งจากการรวบรวมข้อมูลการทำโจทย์ปัญหาการเขียนโปรแกรมของนักศึกษาจำนวน 1010 รายการ และแบ่งชุดฝึกอบรม (80%) และชุดทดสอบ (20%) ซึ่งผลการประเมินโมเดลโดยใช้ชุดข้อมูลทดสอบแสดงดังตารางที่ 4-1

ตารางที่ 4-1 รายงานผลการจำแนก (Classification Report)

ระดับความยาก (Class)	**ความเที่ยง (Precision)**	**ค่าเรียกคืน (Recall)**	**ค่าเฉลี่ยกลมกลืน (F1-Score)**	**จำนวนตัวอย่าง (Support)**
1 (ระดับง่าย)	0.98	1.00	0.99	129
2 (ระดับปานกลาง)	0.98	0.95	0.97	61
3 (ระดับยาก)	1.00	0.92	0.96	12
ความแม่นยำ (Accuracy)			0.98	202
Macro Average	0.99	0.96	0.97	202
Weighted Average	0.98	0.98	0.98	202

ผลการประเมินแสดงให้เห็นว่าโมเดลมีความแม่นยำโดยรวมสูงถึง 98% โดยเฉพาะในระดับความยาก "ง่าย" ที่โมเดลสามารถทำนายได้ถูกต้อง 100% (Recall = 1.00) ในขณะที่ระดับ "ยาก" ซึ่งมีข้อมูลน้อยที่สุด (12 ตัวอย่าง) ก็ยังได้ค่า F1-Score สูงถึง 0.96 สำหรับการวิเคราะห์รายละเอียดเพิ่มเติมของการทำนายในแต่ละระดับ สามารถดูได้จาก Confusion Matrix ดังแสดงในตารางที่ 4-2 ซึ่งแสดงจำนวนการทำนายถูกและผิดในแต่ละระดับความยาก

ตารางที่ 4-2 Confusion Matrix ของโมเดล

ค่าความจริง \ ค่าที่ทำนาย	**ระดับง่าย (Level 1)**	**ระดับปานกลาง (Level 2)**	**ระดับยาก (Level 3)**
ระดับง่าย (Level 1)	129	0	0
ระดับปานกลาง (Level 2)	3	58	0
ระดับยาก (Level 3)	0	1	11

เมื่อวิเคราะห์ข้อมูลจาก Confusion Matrix แล้วพบว่า

- ผู้เรียนในระดับง่ายจำนวน 129 คน ถูกจำแนกได้อย่างถูกต้องทั้งหมด
- ในกลุ่มระดับปานกลาง มีผู้เรียน 3 คนที่โมเดลทำนายผิดเป็นระดับง่าย
- สำหรับระดับยาก มีผู้เรียน 1 คนถูกทำนายคลาดเคลื่อนไปเป็นระดับปานกลาง

การที่โมเดลไม่มีการทำนายคลาดเคลื่อนจากระดับง่ายไปยาก หรือจากยากไปง่าย แสดงให้เห็นว่ามีความระมัดระวังในการทำนาย และสามารถแยกแยะลำดับของความยากได้อย่างมีลำดับชั้น ซึ่งเป็นลักษณะที่เหมาะสมกับระบบแนะนำคำถามแบบปรับได้

บทที่ 5
สรุปและอภิปรายผล

5.1 สรุปผลการดำเนินงาน

การดำเนินงานวิจัยการพัฒนามอดูลการวัดความซับซ้อนของซอร์สโค้ดสำหรับแพลตฟอร์มการเรียนรู้การเขียนโปรแกรม คณะผู้วิจัยได้ทำการศึกษางานวิจัยที่เกี่ยวข้อง ตลอดจนวิเคราะห์ออกแบบ และปรับปรุงแพลตฟอร์มการเรียนรู้การเขียนโปรแกรม เพื่อให้แพลตฟอร์มสามารถวัดค่าความซับซ้อนของซอร์สโค้ดของนักศึกษาที่ทำการส่งงานที่ได้รับมอบหมายซึ่งเป็นโจทย์ปัญหาสำหรับในการเขียนโปรแกรม ซึ่งมอดูลการวัดความซับซ้อนของซอร์สโค้ดและถูกออกแบบให้มีการวัดค่าความซับซ้อน 2 มาตรวัด คือ มาตรวัด Cyclomatic และมาตรวัดซอฟต์แวร์เชิงวัตถุ ซึ่งมาตรวัด 2 มาตรวัดนี้สามารถบ่งบอกถึงลักษณะการเขียนโปรแกรมทั้งการเขียนโปรแกรมเชิงโครงสร้าง (Structured Programming) และการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming: OOP) และสะท้อนถึงขีดความสามารถของนักศึกษาในการแก้โจทย์ปัญหาแต่ละข้อด้วย ซึ่งอาจารย์ผู้สอนสามารถดูรายงานค่าความซับซ้อนของซอร์สโค้ดของนักศึกษาทั้งรายโจทย์ปัญหาและภาพรวมของโจทย์ปัญหาทั้งหมดในรายวิชาได้ ซึ่งสะท้อนถึงระดับความยากของโจทย์ และช่วยให้อาจารย์สามารถปรับปรุงโจทย์ปัญหาหรือจัดลำดับความยากง่ายของโจทย์ เพื่อสร้างประสบการณ์การเรียนรู้อย่างเป็นระบบแก่นักศึกษาในชั้นเรียน นอกจากนี้ งานวิจัยนี้ยังได้พัฒนามอดูลการเรียนรู้แบบปรับได้เพื่อให้นักศึกษาสามารถเข้ามาฝึกฝนการเขียนโปรแกรมได้ด้วยตนเอง โดยที่แพลตฟอร์มนี้จะทำหน้าที่ในการแนะนำโจทย์ปัญหาการเขียนโปรแกรมที่เหมาะสมกับระดับความสามารถของนักศึกษาที่เข้ามาใช้งาน โดยการแนะนำโจทย์ปัญหานี้ได้ใช้วิธีการการเรียนรู้ของเครื่อง (Machine Learning) วิธีการ Random Forest ซึ่งพิจารณาจากคุณลักษณะของซอร์สโค้ดที่นักศึกษาทำการส่งล่าสุด เช่น จำนวนบรรทัดของซอร์สโค้ด ค่าความซับซ้อนของซอร์สโค้ด และคุณลักษณะเชิงพฤติกรรมของนักศึกษา เช่น เวลาที่นักศึกษาใช้ในการทำโจทย์ปัญหา จำนวนครั้งที่นักศึกษาทำการส่งซอร์สโค้ด คะแนนความถูกต้องของซอร์สโค้ด รวมถึงระดับความยากของโจทย์ปัญหา และระดับความสามารถของนักศึกษาในปัจจุบัน ซึ่งผลลัพธ์ของโมเดลจะแนะนำระดับความยากของโจทย์ปัญหาที่เหมาะสมกับนักศึกษา หลังจากนั้นแพลตฟอร์มจะทำการแนะนำโจทย์ปัญหาจำนวน 5 ข้อให้แก่นักศึกษาเพื่อทำการเลือกทำโจทย์ปัญหาเหล่านั้น ซึ่งกลไกนี้จะช่วยให้นักศึกษาไม่พบกับโจทย์ปัญหาที่ยากจนเกินไปหรือง่ายจนเกินไป และสามารถฝึกฝนการเขียนโปรแกรมได้อย่างต่อเนื่อง

5.2 ประโยชน์ที่ได้รับ

1) อาจารย์ผู้สอนสามารถดูรายงานค่าความซับซ้อนของซอร์สโค้ดที่นักศึกษาทำการเขียนขึ้นมาในแต่ละโจทย์ปัญหา และเปรียบเทียบกับค่าความซับซ้อนของเฉลยของโจทย์ปัญหานั้น เพื่อสะท้อนถึงระดับความสามารถการเขียนโปรแกรมของนิสิตในแต่ละโจทย์ปัญหา

2) รายงานค่าความซับซ้อนของซอร์สโค้ดต่างๆ ช่วยให้อาจารย์ผู้สอนนำไปใช้ในการปรับกิจกรรมการสอน/การมอบหมายงานต่างๆ ในการจัดประสบการณ์การเรียนรู้แก่นักศึกษาอย่างเป็นระบบ

3) นักศึกษาสามารถเลือกทำโจทย์ปัญหาการเขียนโปรแกรมที่ระบบแนะนำได้ตามระดับความสามารถของนักศึกษารายบุคคล

5.3 ข้อจำกัด

1) ฟีเจอร์ที่ใช้ในการฝึกโมเดล เช่น จำนวนบรรทัดของโค้ด (Lines of Code: LOC) และค่าความซับซ้อนแบบไซโคลมาติก (Cyclomatic Complexity: CCN) มีข้อจำกัดในตัวเอง โดย LOC อาจได้รับอิทธิพลจากรูปแบบสไตล์การเขียนโค้ดของนักศึกษาแต่ละคน และ CCN ไม่สามารถสะท้อนมิติทั้งหมดของความซับซ้อนทางอัลกอริธึมหรือคุณภาพในการแก้ปัญหาได้อย่างครอบคลุม

2) โมเดลปัจจุบันอาศัยข้อมูลสถานะของผู้เรียนในช่วงเวลาหนึ่ง โดยไม่ได้พิจารณาถึงแนวโน้มของพฤติกรรมผู้เรียนในระยะยาว จึงอาจจำกัดความสามารถของระบบในการตอบสนองต่อการเปลี่ยนแปลงอย่างต่อเนื่อง ไม่ว่าจะเป็นพัฒนาการหรือการถดถอยของผู้เรียน

5.4 ข้อเสนอแนะในอนาคต

1) พัฒนาแดชบอร์ดเพื่อให้นักเรียนและอาจารย์สามารถมองเห็นพัฒนาการของตนเอง หรือของนักศึกษาแต่ละคนในแต่ละชั้นเรียนได้อย่างชัดเจน

2) ประยุกต์ใช้เทคนิคการสร้างแบบจำลองที่สามารถติดตามความก้าวหน้าของผู้เรียนในระยะยาว (Dynamic Modeling) การเปรียบเทียบกับวิธีการอื่นๆ เช่น ทฤษฎีการตอบสนองต่อข้อสอบ (Item Response Theory: IRT) และการเรียนรู้แบบเสริมแรง (Reinforcement Learning) ตลอดจนการนำข้อมูลป้อนกลับจากผู้เรียนมาใช้ในการปรับปรุงระบบแบบเรียลไทม์ (Real-time Feedback)

ภาคผนวก

ภาคผนวก ก

บทความวิจัยที่ตีพิมพ์เผยแพร่

2024 16th International Conference on Information Technology and Electrical Engineering (ICITEE)

Measurement Module for Supporting Arrange Difficulty Coding Questions on KruCode Platform

Peerasak Pianprasit
Software Engineering
Faculty of Informatics
Burapha University
Chonburi, Thailand
peerasak@buu.ac.th

Nuttaporn Phakdee
Software Engineering
Faculty of Informatics
Burapha University
Chonburi, Thailand
nuttaporn@go.buu.ac.th

2024 16th International Conference on Information Technology and Electrical Engineering (ICITEE) | 979-8-3503-7581-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICITEE62483.2024.10808506

***Abstract*— This paper aims to present a measurement module that is an ongoing development for the KruCode platform, which is a platform for teaching and learning programming-related subjects. Currently, the platform includes functions such as defining problems, assigning students assignments, and assessing source code and reports. The objective of this platform was to make teaching and learning more accessible for instructors and students. The measurement module will measure the complexity of the student source code for each problem. We use cyclomatic metrics and object oriented design metrics (C-K metrics) to measure student source codes. The question complexity reports support the instructor in deciding the coding questions, considering the difficulty level of coding questions well suited to students, and building the learning outcome of that course. This information may be utilized to establish assignments, adjust activities to be relevant skills for the class's learners, and assess the level of skills of students.

Keywords*— *measurement, cyclomatic metrics, object oriented design metrics, difficulty coding question

I. Introduction

It is widely acknowledged that practising solving various programming challenges enhances programming skills. Therefore, programming instructors often design teaching methodologies that impart knowledge while assigning programming tasks, ensuring that the lessons are neither too difficult nor too easy. This approach aims to progressively foster expertise in programming skills, allowing each student to tackle increasingly complex problems. That aligns with Vygotsky's concept of the zone of proximal development, which continues to inspire educational design and development efforts [1].

Instructors evaluate the complexity of assignments regularly, especially in programming courses. Assessing the correctness, coverage, and complexity of students' program source codes is expected in programming teaching. However, determining how one coding question is more complex than another and attempting to sequence coding questions in assignments from easy to difficult is challenging. Therefore, most instructors rely on their experience to design appropriate teaching methodologies, including inspecting students' source code for complexity. Nevertheless, the methods for thinking about and writing programs may vary, and each student's perspective on the difficulty of solving the same problem. Nonetheless, previous research [2] implies that examination questions in programming courses often correlate easy-level questions with straightforward thinking processes, a small number of source code lines, and clear question formulations. Higher-level questions typically involve complex concepts and much more source code.

Therefore, this research presents the design of an application that utilizes software complexity measurement processes to create a metric for assessing the complexity level of coding questions. This tool aims to reflect whether the instructional designs implemented by instructors are appropriate. Furthermore, it serves as a tool for simplifying the assessment process and reducing time while providing accurate information about the complexity of source code.

II. Background of Research

A. Software Measurement and Metrics

Software measurement is the process of measuring different characteristics of a process or product, such as the size, amount, or dimension of a specific feature. Software metrics measure the level at which any impute belongs to a system product or process. For example, a basic measure like lines of code (LOC) counts the number of lines in source code. LOC is typically used to predict the effort required to develop a program. Similarly, evaluating effort expended in activities such as unit testing demands a clear understanding of the process boundaries to establish when it begins and ends. Software metrics are a quantifiable or countable assessment of the attributes of a software product [3]. Software metrics support risk assessment and mitigation decisions during software development [4].

B. Cyclomatic Complexity

Cyclomatic complexity (CC), proposed by Thomas McCabe in 1976, serves as a software metric to assess a program's complexity by quantifying the number of linearly independent paths within its source code [5], [6], [7]. Utilizing a control-flow graph representation, where nodes signify indivisible code segments and edges denote potential execution flow transfers, this metric aids in determining program stability and confidence levels, with lower complexity indicating higher understandability and lower modification risks [8]. By interpreting programs as strongly connected directed graphs, McCabe's metric enables the analysis and regulation of path proliferation, correlating a program's complexity with the topological intricacy of its graph structure. Previous research [9], [10] has proposed this metric for software maintenance productivity.

C. Object Oriented Design Metrics

The metrics suite [11] proposed by Chidamber and Kemerer, also known as "C-K metrics," has garnered significant attention from researchers and software engineers, leading to numerous empirical studies evaluating these metrics. This suite comprises six class-level metrics, consist Weighted Methods per Class (WMC), Response For a Class

Authorized licensed use limited to: Burapha University provided by UniNet. Downloaded on May 19,2025 at 03:32:56 UTC from IEEE Xplore. Restrictions apply.

(RFC), Depth of Inheritance Hierarchy (DIT), Number of Children (NOC), Coupling Between Objects (CBO), and Lack of Cohesion of Methods (LCOM). The C-K metrics have been so popular that researchers have examined and empirically validated them in various projects. These metrics have become widely accepted and utilized in object oriented design assessments, contributing to the demand for software measures aimed at process improvement.

D. *Case Tools For Software Metrics*

Various CASE (Computer Aided Software Engineering) tools are available for measuring software, catering to different needs and preferences [12], [13], [14], [15], [16]. For example, Analyst4j operates on the Eclipse platform and provides comprehensive analysis and reporting capabilities for Java programs; CCCC is an open-source command-line tool that supports both C++ and Java codebases with various metrics. Metrix++ is an extension tool for the collection and analysis of code metrics. Dependency Finder is a suite of tools for analyzing compiled Java code and extracting dependency graphs. Platform-specific tools like SonarQube, Jenkins, Code Climate, GitLab, and Lizard, each with unique features and supported languages, aim to streamline continuous inspection, code review, and performance analysis processes. Other tools such as CKJM, OOMeter, and Semmle cater to specific needs, such as measuring object-oriented design metrics in Java codebases or providing customizable code analysis through dedicated query languages. Each tool has strengths and limitations, requiring careful consideration based on the project requirements and preferences.

E. *Related Work*

Software metrics are used to measure software or processes in the software development process. It has also been used in the field of education, for example, in following research. Yoshee Jain and Kathryn Cunningham propose a matrix for curriculum planning, emphasizing criteria such as the frequency of professional usage, comprehensibility to novices, and alignment with domain context [17]. Said Elnaffar developed the Predicted Difficulty Index (PDI) tool to assess coding question difficulty, assisting in better exam design using software metrics from sample solutions [18]. The metrics estimate the difficulty of code writing tasks, assisting in designing and selecting exam questions, with higher cyclomatic complexity indicating greater question complexity [19]. Software metrics are pivotal in assessing code comprehension tasks, providing valuable insights into code tracing difficulties, particularly in first-year programming assessments [20]. Programming question complexity is a multifaceted aspect influencing student performance and learning standards [2], [21]. Assessing the difficulty of exam questions is crucial in gauging the expected skills and knowledge level at the end of a course.

III. Our Approach

A. *KruCode Platform*

KruCode Platform [22] used the Minimum Viable Product (MVP) principle to design a platform supporting programming education. This platform serves as a tool for instructors to manage courses, student groups, assignments, and code submissions. The platform focuses on providing immediate feedback. The platform aims to formulate a coding culture among university students, offering continuous feedback learning, problem-solving assignments, and support for various programming languages. However, it cannot measure the complexity of coding questions; instructors may use more effort to assess the difficulty level of assignments. Consequently, adjusting assignments (coding questions) to match students' skill levels.

B. *Complexity Measurement Module Design*

In selecting metrics for measuring code complexity, we considered undergraduate software engineering curricula, which represent the primary user group for current platforms. We observed various programming-related subjects, such as programming principles, object-oriented programming, and data structures and algorithms. Cyclomatic Complexity (CC) metrics and object oriented metrics were deemed particularly crucial for helping instructors assess code complexity and gain clear insights into student performance. Cyclomatic complexity supports understanding the program's complexity; object oriented metrics shed light on class characteristics. However, these metrics may only partially reflect code quality or usability. Nevertheless, they could serve as indicators for evaluating the suitability of object oriented design and fostering more collaborative discussions between students and instructors.

The modules' design leverages a service-based architecture due to its flexibility and seamless integration capabilities with existing systems. The modules that measure code complexity also continue interacting with the existing database and platform, facilitating more accessible data presentation and analysis of difficult coding questions.

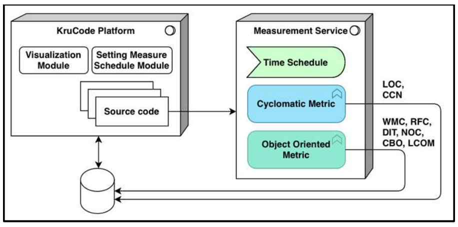


Fig. 1. The measurement module design

Fig. 1 shows the topology of the KruCode platform and measurement module. The process begins with retrieving source code data for each student's programming problem from the KruCode platform, including the problem ID and source code files. After that, the measurement service measures the complexity of the source code and stores the results in a database. Afterwards, statistical methods are applied to this data to refine the complexity of future programming problems on the KruCode platform. Miscellaneous reports are generated to aid instructors in making informed decisions about improving or assigning programming problems for learners.

There are many metrics for measuring source code complexity, for this research, we selected the following metrics: LOC, Cyclomatic Complexity, as well as object oriented metrics like Weighted Methods per Class (WMC) and Coupling Between Objects (CBO) for their relevance to both structural and object oriented complexity. These metrics are particularly suited to the KruCode platform's objectives

Authorized licensed use limited to: Burapha University provided by UniNet. Downloaded on May 19,2025 at 03:32:56 UTC from IEEE Xplore. Restrictions apply.

because they are simple, widely understood, and provide actionable insights that help instructors make informed decisions. Adding more metrics would risk redundancy, as many overlap without significantly improving the assessment. Given the wide range of problems tackled by KruCode, these metrics offer the most practical, relevant, and interpretable results for instructors and students.

IV. System Implementation

First of all, the instructors sets the queue for processing the complexity of the source code, which is a process of creating a queue sequence to process the complexity of the desired problem. Users must first select the academic year before entering course and class group data and specifying the processing time for the source code complexity analysis, as shown in Fig. 2.

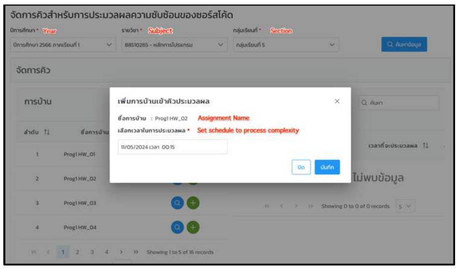


Fig. 2. Setting schedule for process complexity of the source code

After the processing is completed, instructors can access reports on the average complexity of the source code submitted by students for each problem statement. These reports reflect whether the coding questions are arranged from easy to difficult, as shown in Fig. 3.

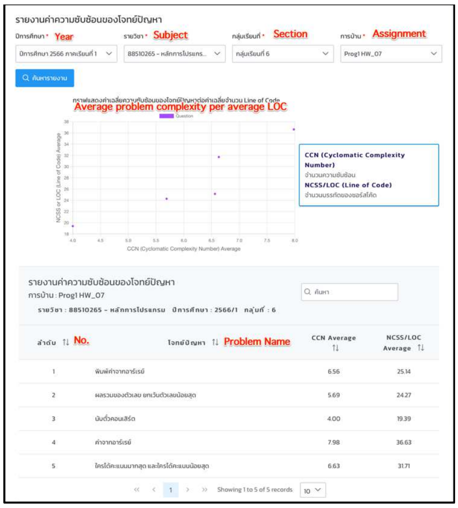


Fig. 3. Average complexity of the source code submitted by students for each problem statement

The instructor can view the cyclomatic complexity (CCN) and the number of lines of code (LOC) of each student's source code for each coding question, comparing it with the complexity of the instructor's solution. Red points indicate the complexity of the instructor's solution, green points represent students' full score, yellow points represent students' partial score, and blue points represent students' zero scores (indicating that the program did not meet the problem statement expectations). This report suggests that source code with full scores but high complexity or LOC may require conceptual refinement for program optimization. Meanwhile, when many students receive a score of zero, it suggests the need to teach test design and error reduction during program execution. Additionally, if many students receive zero scores, the instructors may need to review the teaching approach by explaining the concepts and program design, as shown in Fig. 4.

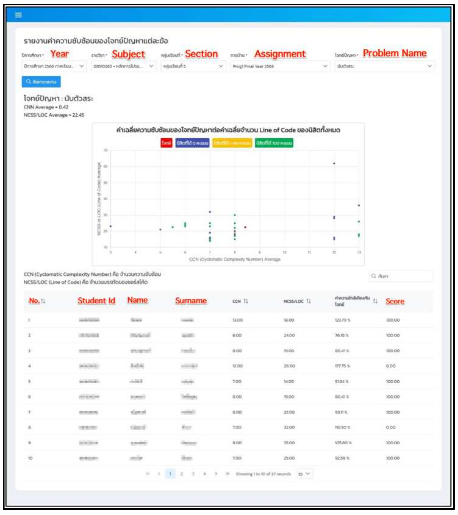


Fig. 4. Comparison of complexity, LOC between instructor solution and students source code

This information from the system can answer these questions. Can students write the source code to solve the problem or not? Are the coding questions arranged in order from easy to difficult? This information is important for instructors to rearrange the coding questions suitably to create an appropriate learning curve, reflect a picture of students' understanding, and adjust their teaching to suit students' needs during that semester, which is a short-term operation. For long-term operation, when the same subject is taught multiple times, the coding questions can be reused for the new group of students enrolling in the class. Since learners have similar background knowledge and the same problems, it will lead to a reflection of the complexity of the problem that is appropriate for the learner at that level and to see the overall picture of the whole course, whether the sequence of the problem is an appropriate sequence of increasing difficulty or review the coding question to adjust accordingly.

Authorized licensed use limited to: Burapha University provided by UniNet. Downloaded on May 19,2025 at 03:32:56 UTC from IEEE Xplore. Restrictions apply.

Instructors can access reports on coding question complexity for the entire course in a specific class, as shown in Fig. 5 and Fig. 6. It shows a scatter chart and a table of the problem complexity. The user can hover the mouse over any point. Then, the system displays the problem number, the average of lines of code, and the average complexity. This supports deciding if adjustments are needed for future iterations of the course, enabling informed decisions on curriculum enhancements based on observed complexity trends in programming assignments.

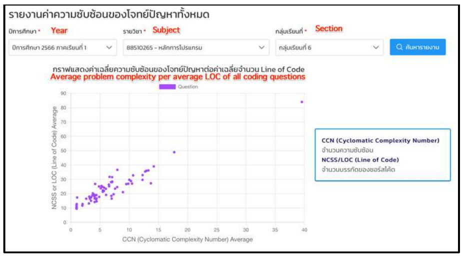


Fig. 5. Graph represents the complexity of all coding questions in course

รายงานค่าความซับซ้อนของโจทย์ปัญหาทั้งหมด Cyclomatic complexity of all coding questions

รายวิชา : 88510265 - หลักการโปรแกรม ปีการศึกษา : 2566/1 กลุ่มที่ : 6

ลำดับ No.	โจทย์ปัญหา Problem Name	CCN Average	NCSS/LOC Average
1	A+B	0.97	9.78
2	พื้นที่และเส้นรอบรูปสี่เหลี่ยมผืนผ้า	0.98	11.77
3	พื้นที่สามเหลี่ยม	1.00	10.60
4	องศาเซลเซียส เป็น องศาฟาเรนไฮต์	1.00	9.52
5	พิมพ์ตัวเลขย้อนกลับ	1.09	17.32
6	จำนวนเต็ม	3.47	15.07
7	ตรวจสอบตัวเลข	4.09	14.19
8	ค่าน้อยที่สุดจากเลข 3 จำนวน	4.87	17.75
9	ค่าจอดรถ	4.33	16.22
10	เวลาที่ผ่านไป กี่โมงแล้ว	7.95	24.62

Showing 1 to 10 of 67 records

Fig. 6. Cyclomatic complexity of all coding questions in course

The results of the object oriented metrics showcase various values according to the metrics for each coding problem in the students' source code, as shown in Fig. 7.

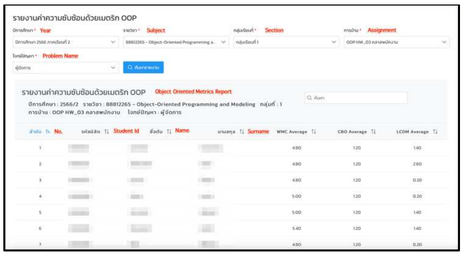


Fig. 7. Object oriented metrics report

Additionally, instructors can access individual details of each student to demonstrate how various metrics are reflected in each student's source code files, as shown in Fig. 8.

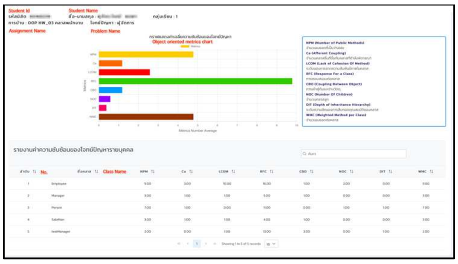


Fig. 8. Individual's source code object oriented metrics

V. Conclusion and Discussion

This research used software metrics, including cyclomatic complexity and C-K metrics, to analyze the source code submitted by students, aiming to measure various complexities according to the object oriented metrics. The average complexity values of the problem statements then displayed, reflecting the difficulty level of the problem statements and aiding instructors in ranking them for a structured learning experience. Additionally, object oriented metrics were presented to illustrate the quantity and magnitude of different complexities based on object oriented metrics.

For future research, a dashboard will be developed to allow students and instructors to visualize their own development progress or the development of students in each class clearly. Additionally, machine learning technology may be utilized to recommend problem statements suitable for each student.

Acknowledgment

This research is fully funded, and facilities are provided by the Faculty of Informatics at the Burapha University of Thailand. This authors would like to send gratitude to our developers' team in Software Engineering Department for the research direction and collaboration.

References

[1] W. Schnotz and C. Kürschner, "A Reconsideration of Cognitive Load Theory," *Educational Psychology Review*, vol. 19, no. 4, pp. 469–508, Dec. 2007, doi: 10.1007/s10648-007-9053-4.

[2] J. Sheard *et al.*, *How difficult are exams? A framework for assessing the complexity of introductory programming exams*. 2013.

[3] "Software Measurement and Metrics," GeeksforGeeks. Accessed: Feb. 05, 2024. [Online]. Available: https://www.geeksforgeeks.org/software-measurement-and-metrics/

[4] N. Fenton and M. Neil, "Software Metrics: Roadmap," *Proceedings of the Conference on the Future of Software Engineering*, Sep. 2000, doi: 10.1145/336512.336588.

[5] T. J. McCabe, "A Complexity Measure," *IEEE Transactions on Software Engineering*, vol. SE-2, no. 4, pp. 308–320, Dec. 1976, doi: 10.1109/TSE.1976.233837.

[6] "Cyclomatic complexity," *Wikipedia*. May 01, 2024. Accessed: May 15, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Cyclomatic_complexity&oldid=1221707517

Authorized licensed use limited to: Burapha University provided by UniNet. Downloaded on May 19,2025 at 03:32:56 UTC from IEEE Xplore. Restrictions apply.

[7] "Cyclomatic Complexity," GeeksforGeeks. Accessed: Feb. 05, 2024. [Online]. Available: https://www.geeksforgeeks.org/cyclomatic-complexity/

[8] "IBM Documentation." Accessed: Feb. 05, 2024. [Online]. Available: https://www.ibm.com/docs/en/raa/6.1?topic=metrics-cyclomatic-complexity

[9] G. K. Gill and C. F. Kemerer, "Cyclomatic complexity density and software maintenance productivity," *IEEE Transactions on Software Engineering*, vol. 17, no. 12, pp. 1284–1288, Dec. 1991, doi: 10.1109/32.106988.

[10] J. P. Mittal, P. Bhatia, and H. Mittal, "Software maintenance productivity assessment using fuzzy logic," *SIGSOFT Softw. Eng. Notes*, vol. 34, no. 5, pp. 1–4, Oct. 2009, doi: 10.1145/1598732.1598739.

[11] S. R. Chidamber and C. F. Kemerer, "A metrics suite for object oriented design," *IEEE Transactions on Software Engineering*, vol. 20, no. 6, pp. 476–493, 1994, doi: 10.1109/32.295895.

[12] R. Lincke, J. Lundberg, and W. Löwe, "Comparing software metrics tools," in *Proceedings of the 2008 International Symposium on Software Testing and Analysis*, in ISSTA '08. New York, NY, USA: Association for Computing Machinery, 2008, pp. 131–142. doi: 10.1145/1390630.1390648.

[13] "Software Engineering | Case Tools For Software Metrics - javatpoint." Accessed: May 15, 2024. [Online]. Available: https://www.javatpoint.com/software-engineering-case-tools-for-software-metrics

[14] R. Lincke, J. Lundberg, and W. Löwe, *Comparing software metrics tools*. 2008, p. 142. doi: 10.1145/1390630.1390648.

[15] Sandeep Kaur and Navjot Kaur, "Software Metrics and Metric Tools- A Review," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, no. 4, pp. 2076–2079.

[16] "How to Measure Code Complexity," TIOBE. Accessed: May 15, 2024. [Online]. Available: https://www.tiobe.com/knowledge/article/controlling-code-complexity/

[17] Y. Jain and K. Cunningham, "Towards Methods for Identifying High-Quality Domain-Specific Programming Plans," in *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 2*, in ICER '23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 18–19. doi: 10.1145/3568812.3603478.

[18] S. Elnaffar, "Using software metrics to predict the difficulty of code writing questions," in *2016 IEEE Global Engineering Education Conference (EDUCON)*, Apr. 2016, pp. 513–518. doi: 10.1109/EDUCON.2016.7474601.

[19] J. Whalley and N. Kasto, "How difficult are novice code writing tasks? a software metrics approach," in *Proceedings of the Sixteenth Australasian Computing Education Conference - Volume 148*, in ACE '14. AUS: Australian Computer Society, Inc., 2014, pp. 105–112.

[20] N. Kasto and J. Whalley, "Measuring the difficulty of code comprehension tasks using software metrics," in *Proceedings of the Fifteenth Australasian Computing Education Conference - Volume 136*, in ACE '13. AUS: Australian Computer Society, Inc., 2013, pp. 59–65.

[21] C.-Y. Chung and I.-H. Hsiao, "Semantic Modeling of Programming Practices with Local Knowledge Graphs: The Effects of Question Complexity on Student Performance," in *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners' and Doctoral Consortium: 23rd International Conference, AIED 2022, Durham, UK, July 27–31, 2022, Proceedings, Part II*, Berlin, Heidelberg: Springer-Verlag, 2022, pp. 245–249. doi: 10.1007/978-3-031-11647-6_44.

[22] N. Phakdee and P. Pianprasit, "KruCode: An Online Learning Programming Platform for Programming Course," in *2023 27th International Computer Science and Engineering Conference (ICSEC)*, 2023, pp. 79–84. doi: 10.1109/ICSEC59635.2023.10329732.

Authorized licensed use limited to: Burapha University provided by UniNet. Downloaded on May 19,2025 at 03:32:56 UTC from IEEE Xplore. Restrictions apply.

ภาคผนวก ข

พจนานุกรมข้อมูล (Data Dictionary)

พจนานุกรมข้อมูลจะทำให้สามารถอธิบายรายละเอียดเกี่ยวกับข้อมูลตามแผนภาพความสัมพันธ์ของข้อมูลที่ไม่ได้อธิบายไว้ ซึ่งเป็นเครื่องมือที่นิยมทำควบคู่กับแผนภาพความสัมพันธ์ของข้อมูล โดยพจนานุกรมข้อมูลถูกนำใช้อ้างอิง เพื่อขยายความหมายว่าข้อมูลที่ได้รับมานั้นมีคำอธิบายว่าอะไรบ้าง อธิบายนิยามคำจำกัดความข้อมูล รวมไปถึงชุดข้อมูลนั้นอยู่ในรูปแบบใด มีโครงสร้างข้อมูลและค่าของข้อมูลเป็นอย่างไร เป็นต้น หากมีการแก้ไขหรือเปลี่ยนแปลงเกี่ยวกับข้อมูลในระบบงานผู้ดำเนินโครงงานจะต้องทำการปรับปรุงพจนานุกรมข้อมูลทุกครั้ง เพื่อให้ง่ายต่อการพัฒนาและปรับปรุงต่อยอดระบบงานในอนาคตสามารถแสดงรายละเอียดดังต่อไปนี้

ตารางที่ ข-1 รายละเอียดตาราง headerReport

ชื่อตาราง		headerReport				
คำอธิบาย		หัวตารางรายงานค่าความซับซ้อนของซอร์สโค้ด				
คีย์หลัก (PK)		headerReportId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
1	headerReportId (PK)	INT	8	-	รหัสหัวตารางรายงานค่าความซับซ้อนของซอร์สโค้ด	2
2	headerReportUserId (FK)	INT	8	-	รหัสผู้ใช้งาน	3
3	headerReportAssignmentId (FK)	INT	8	-	รหัสงานที่ได้รับมอบหมาย	3
4	headerReport ProblemId (FK)	INT	8	-	รหัสโจทย์ปัญหา	3

ชื่อตาราง		headerReport				
คำอธิบาย		หัวตารางรายงานค่าความซับซ้อนของซอร์สโค้ด				
คีย์หลัก (PK)		headerReportId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
5	headerReportWMC	decimal	(10,4)	-	จำนวนเฉลี่ยของเมธอดต่อคลาส	12.5
6	headerReportDIT	decimal	(10,4)	-	ค่าเฉลี่ยระดับความลึกของการสืบทอด	12.5
7	headerReportNOC	decimal	(10,4)	-	จำนวนเฉลี่ยของคลาสลูกทั้งหมดที่สืบทอด	12.5
8	headerReportCBO	decimal	(10,4)	-	ค่าเฉลี่ยระดับความเกี่ยวข้อง	12.5
9	headerReportRFC	decimal	(10,4)	-	จำนวนเฉลี่ยเมธอดภายในคลาสที่สามารถตอบสนอง	12.5
10	headerReportLCOM	decimal	(10,4)	-	ค่าเฉลี่ยการเกาะกันเป็นก้อนของคลาส	12.5
11	headerReportCa	decimal	(10,4)	-	จำนวนเฉลี่ยคลาสอื่นที่ขึ้นกับคลาส	12.5

ชื่อตาราง		headerReport				
คำอธิบาย		หัวตารางรายงานค่าความซับซ้อนของซอร์สโค้ด				
คีย์หลัก (PK)		headerReportId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
12	headerReportNPM	decimal	(10,4)	-	จำนวนเฉลี่ยของเมธอด	12.5
13	headerReportNCSS	decimal	(10,4)	-	จำนวนเฉี่ยของ LOC	12.5
14	headerReportCCN	decimal	(10,4)	-	จำนวนเฉลี่ยของความซับซ้อนไซโคลเมตริกของแมคเคบ	12.5
15	headerReportMetricOOP	TINYINT	1	-	ตัวเช็คข้อมูลของมาตรวัด CKJM	1
16	headerReportMetricLizard	TINYINT	1	-	ตัวเช็คข้อมูลของมาตรวัดไซโคลเมตริกของแมคเคบ	0
17	headerReportCreateDate	TIMESTAMP	-	CURREN_TIMESTAMP	เป็นคอลัมน์เก็บวันที่สร้างรายงาน	2023-03-01 05:49:13.000
18	headerReport UpdateDate	TIMESTAMP	-	CURREN_TIMESTAMP	เป็นคอลัมน์เก็บวันที่แก้ไขรายงาน	2023-03-01 05:49:13.000

ตารางที่ ข-2 รายละเอียดตาราง detailReportLizard

ชื่อตาราง		detailReportLizard				
คำอธิบาย		ค่าความซับซ้อนของมาตรวัดไซโคลเมตริกของแมคเคบ				
คีย์หลัก (PK)		detailReportLizardId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
1	detailReportLizardId (PK)	INT	8	-	รหัสตารางค่าความซับซ้อนของมาตรวัดไซโคลเมตริกของแมคเคบ	2
2	detailReportLizard fileName	VAR CHAR	255	-	ชื่อไฟล์ที่วัดค่าความซับซ้อน	Main .java
3	detailReportLizard NCSS	decimal	(10,4)	-	จำนวนของ LOC	13
4	detailReportLizard CCN	decimal	(10,4)	-	จำนวนของความซับซ้อนไซโคลเมตริกของแมคเคบ	34
5	detailReportLizard CreateDate	TIMEST AMP	-	CURREN_ TIMESTA MP	เป็นคอลัมน์เก็บวันที่สร้างรายงาน	2023-03-01 05:49:1 3.000
6	detailReportLizard UpdateDate	TIMEST AMP	-	CURREN_ TIMESTA MP	เป็นคอลัมน์เก็บวันที่แก้ไขรายงาน	2023-03-01 05:49:1 3.000

ชื่อตาราง	detailReportLizard					
คำอธิบาย	ค่าความซับซ้อนของมาตรวัดไซโคลเมตริกของแมคเคบ					
คีย์หลัก (PK)	detailReportLizardId					
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
7	detailReportLizard headerReportId (FK)	INT	8	-	รหัสหัวตารางรายงานค่าความซับซ้อนของซอร์สโค้ด	3

ตารางที่ ข-3 รายละเอียดตาราง detailReportOOP

ชื่อตาราง	detailReportOOP					
คำอธิบาย	ตารางค่าความซับซ้อนของ CKJM					
คีย์หลัก (PK)	detailReportOOPId					
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
1	detailReportOOPId (PK)	INT	8	-	รหัสตารางค่าความซับซ้อนของ CKJM	2
2	detailReportOOPfile Name	VAR CHAR	255	-	ชื่อไฟล์ที่วัดค่าความซับซ้อน	Main .java
3	detailReportOOP WMC	decimal	(10,4)	-	จำนวนของเมธอดต่อคลาส	13
4	detailReportOOPDIT	decimal	(10,4)	-	ค่าระดับความลึกของการสืบทอด	34

ชื่อตาราง		detailReportOOP				
คำอธิบาย		ตารางค่าความซับซ้อนของ CKJM				
คีย์หลัก (PK)		detailReportOOPId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
5	detailReportOOPNOC	decimal	(10,4)	-	จำนวนของคลาสลูกทั้งหมดที่สืบทอด	3
6	detailReportOOPCBO	decimal	(10,4)	-	ค่าระดับความเกี่ยวข้อง	5
7	detailReportOOPRFC	decimal	(10,4)	-	จำนวนเมธอดภายในคลาสที่สามารถตอบสนอง	3
8	detailReportOOPLCOM	decimal	(10,4)	-	ค่าการเกาะกันเป็นก้อนของคลาส	0
9	detailReportOOPCa	decimal	(10,4)	-	จำนวนคลาสอื่นที่ขึ้นกับคลาส	12
10	detailReportOOPNPM	decimal	(10,4)	-	จำนวนของเมธอด	4
11	detailReportOOPCreateDate	TIMESTAMP	-	CURREN_TIMESTAMP	เป็นคอลัมน์เก็บวันที่สร้างรายงาน	2023-03-01 05:49:13.000
12	detailReportOOPUpdateDate	TIMESTAMP	-	CURREN_TIMESTAMP	เป็นคอลัมน์เก็บวันที่	2023-03-01

ชื่อตาราง		detailReportOOP				
คำอธิบาย		ตารางค่าความซับซ้อนของ CKJM				
คีย์หลัก (PK)		detailReportOOPId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
					แก้ไขรายงาน	05:49:13.000
13	detailReportOOP headerReportId (FK)	INT	8	-	รหัสหัวตารางรายงานค่าความซับซ้อนของซอร์สโค้ด	6

ตารางที่ ข-4 รายละเอียดตาราง queue

ชื่อตาราง		queue				
คำอธิบาย		ตารางจัดลำดับคิววัดค่าความซับซ้อน				
คีย์หลัก (PK)		queueId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
1	queueId (PK)	INT	8	-	รหัสตารางค่าความซับซ้อนของ CKJM	2
2	queueUserId (FK)	INT	8	-	รหัสผู้ใช้งาน	107
3	queueAssignmentId (FK)	INT	8	-	รหัสงานที่ได้รับมอบหมาย	13
4	queueProblemId (FK)	INT	8	-	รหัสโจทย์ปัญหาที่ได้รับมอบหมาย	34

ชื่อตาราง		queue				
คำอธิบาย		ตารางจัดลำดับคิววัดค่าความซับซ้อน				
คีย์หลัก (PK)		queueId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
5	queuePathFile	TEXT	-	-	ตำแหน่งไฟล์ที่ต้องการวัดความซับซ้อน	/storage/source/2565/1/88510265/5/65160367/24/59/current
6	queueStatus	VARCHAR	20	-	สถานะของคิว	Pending
7	queueStartProcess	TIMESTAMP	-	-	เป็นคอลัมน์เก็บวันที่ที่จะเริ่มวัดความซับซ้อน	2023-03-01 05:49:13.000
8	queueStartCompile	TIMESTAMP	-	-	เป็นคอลัมน์เก็บวันที่ที่เริ่มวัดความซับซ้อน	2023-03-01 05:49:13.000
9	queueFinishedCompile	TIMESTAMP	-	-	เป็นคอลัมน์เก็บวันที่ที่วัดความซับซ้อนเสร็จสิ้น	2023-03-01 05:49:13.000
10	queueCreateDate	TIMESTAMP	-	CURRENT_TIMESTAMP	เป็นคอลัมน์เก็บวันที่สร้างคิว	2023-03-01 05:49:13.000

ชื่อตาราง		queue				
คำอธิบาย		ตารางจัดลำดับคิววัดค่าความซับซ้อน				
คีย์หลัก (PK)		queueId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
11	queueUpdateDate	TIMESTAMP	-	CURREN_TIMESTAMP	เป็นคอลัมน์เก็บวันที่แก้ไขคิว	2023-03-01 05:49:13.000
12	queueCourseId (FK)	INT	8	-	รหัสหลักสูตร	2
13	queueSectionId (FK)	INT	8	-	รหัสกลุ่มเรียน	4

ตารางที่ ข-5 รายละเอียดตาราง problems

ชื่อตาราง		problems				
คำอธิบาย		ข้อมูลโจทย์ปัญหา				
คีย์หลัก (PK)		problemId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
1	problemId (PK)	INT	8	-	รหัสโจทย์ปัญหา	59
2	problemTitle	VARCHAR	255	NULL	ชื่อหัวข้อโจทย์ปัญหา	116
3	Problem Description	TEXT	-	-	รายละเอียดของโจทย์ปัญหา	88
4	Problem CreateDate	TIMESTAMP	-	CURRENT_TIMESTAMP	วันที่และเวลาสร้าง	2022-07-04 11:06:19.000
5	Problem UpdateDate	TIMESTAMP	-	CURRENT	วันที่และเวลาอัปเดต	2022-07-04

ชื่อตาราง		problems				
คำอธิบาย		ข้อมูลโจทย์ปัญหา				
คีย์หลัก (PK)		problemId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
				_TIMESTAMP		11:06:19.000
6	Problem CreateBy	VARCHAR	20	NULL	ชื่อผู้สร้างโจทย์ปัญหา	peerasak
7	Problem UpdateBy	VARCHAR	20	NULL	ชื่อผู้อัปเดตโจทย์ปัญหา	peerasak
8	problemStatus	ENUM	-	‘active’	สถานะของโจทย์ปัญหา	‘active’
9	Problem VoteScore	FLOAT	-	NULL	ค่าเฉลี่ยคะแนนโหวต	3.5
10	problemVote Amount	INT	11	NULL	จำนวนผู้โหวต	15
11	problemTime Limit	INT	11	NULL	ให้รันโค้ดได้นานแค่ไหน	100

ตารางที่ ข-6 รายละเอียดตาราง assignments

ชื่อตาราง		assignments				
คำอธิบาย		หัวข้อของงานที่มอบหมายให้นักศึกษา				
คีย์หลัก (PK)		assignmentId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
1	assignmentId (PK)	INT	8	-	รหัสหัวข้องาน	59
2	assignmentTitle	VARCHAR	255	-	ชื่อหัวข้องาน	Prog1HW_11

ชื่อตาราง		assignments				
คำอธิบาย		หัวข้อของงานที่มอบหมายให้นักศึกษา				
คีย์หลัก (PK)		assignmentId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
3	Assignment Description	VAR CHAR	255	NULL	รายละเอียดหัวข้องาน	Quiz 2 Array and Method
4	Assignment CreateDate	TIME STAMP	-	CURRENT _TIMESTA MP	วันที่และเวลาสร้าง	2022-07-04 11:06:1 9.000
5	Assignment UpdateDate	TIME STAMP	-	CURRENT _TIMESTA MP	วันที่และเวลาอัปเดต	2022-07-04 11:06:1 9.000
6	Assignment CreateBy	VAR CHAR	20	NULL	ชื่อผู้สร้างหัวข้องาน	peera sak
7	Assignment UpdateBy	VAR CHAR	20	NULL	ชื่อผู้อัปเดตหัวข้องาน	peera sak
8	Assignment Status	ENUM	-	‘active’	สถานะของหัวข้องาน	‘active’
9	Assignment Score	INT	11	NULL	คะแนนของหัวข้องาน	40
10	problemVote Amount	INT	11	NULL	จำนวนผู้โหวต	15
11	problemTime Limit	INT	11	NULL	เวลารันได้นานเท่าไหร่	100

ตารางที่ ข-7 รายละเอียดตาราง sections

ชื่อตาราง		sections				
คำอธิบาย		ตารางข้อมูลกลุ่มเรียน				
คีย์หลัก (PK)		sectionId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
1	sectionId (PK)	INT	8	-	รหัสกลุ่มเรียน	1
2	sectionCourseId (FK)	INT	8	-	รหัสหลักสูตร	2
3	sectionNumber	VARCHAR	20	-	เลขกลุ่มเรียน	4
4	sectionCreateDate	TIMESTAMP	-	CURRENT_TIMESTAMP	วันที่และเวลาสร้าง	2022-07-04 11:06:19.000
5	sectionUpdateDate	TIMESTAMP	-	CURRENT_TIMESTAMP	วันที่และเวลาอัปเดต	2022-07-04 11:06:19.000
6	sectionCreateBy	VARCHAR	20	NULL	ชื่อผู้สร้าง	peerasak
7	sectionUpdateBy	VARCHAR	20	NULL	ชื่อผู้อัปเดต	peerasak
8	sectionStatus	ENUM	-	'active'	สถานะของหัวข้องาน	‘active’

ตารางที่ ข-8 รายละเอียดตาราง users

ชื่อตาราง		users				
คำอธิบาย		ตารางข้อมูลผู้ใช้งาน				
คีย์หลัก (PK)		userId , userUsername				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
1	userId (PK)	INT	8	-	รหัสของผู้ใช้งาน	20
2	userUsername (PK)	VARCHAR	20	-	ชื่อสำหรับเข้าใช้งานระบบ	62160105
3	userHashPassword	VARCHAR	255	-	รหัสผ่าน	hxZetL3kgw3lv9hkda
4	userFirstnameThai	VARCHAR	255	-	ชื่อจริงของผู้ใช้งานเป็นภาษาไทย	คเชนทร์
5	userFirstnameEnglish	VARCHAR	255	-	ชื่อจริงของผู้ใช้งานเป็นภาษาอังกฤษ	Kachen
6	userLastnameThai	VARCHAR	255	-	ชื่อสกุลของผู้ใช้งานเป็นภาษาไทย	เข็มทอง
7	userLastnameEnglish	VARCHAR	255	-	ชื่อสกุลของผู้ใช้งานเป็นภาษาอังกฤษ	Khemthong
8	userCreateDate	TIMESTAMP	-	CURRENT_TIMESTAMP	วันที่และเวลาสร้าง	2022-10-17 15:01:24.000

ชื่อตาราง		users				
คำอธิบาย		ตารางข้อมูลผู้ใช้งาน				
คีย์หลัก (PK)		userId , userUsername				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
9	userUpdateDate	TIMESTAMP	-	CURRENT_TIMESTAMP	วันที่และเวลาอัปเดต	2022-10-17 15:13:39.000
10	userRole	ENUM	-	NULL	ประเภทของผู้ใช้งาน	student
11	userCreateBy	VARCHAR	20	NULL	ชื่อผู้สร้างงาน	Pomrapee
12	userUpdateBy	VARCHAR	20	NULL	ชื่อผู้อัปเดตงาน	Pomrapee
13	userStatus	ENUM	-	-	สถานะของผู้ใช้งาน	'active'

ตารางที่ ข-9 รายละเอียดตาราง courses

ชื่อตาราง		courses				
คำอธิบาย		ตารางข้อมูลหลักสูตรเรียน				
คีย์หลัก (PK)		courseId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
1	courseId	INT	8	-	รหัสหลักดสูตร	2
2	courseYearId	INT	8	-	รหัสปีการศึกษา	2
3	ownerUserId	INT	8	-	รหัสผู้ใช้งาน	2
4	courseCode	VARCHAR	255	-	รหัสรายวิชา	88510265

ชื่อตาราง		courses				
คำอธิบาย		ตารางข้อมูลหลักสูตรเรียน				
คีย์หลัก (PK)		courseId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
5	courseName	VAR CHAR	255	-	ชื่อรายวิชา	หลักการโปรแกรม
6	courseDes	VAR CHAR	255	-	คำอธิบายรายวิชา	หลักการโปรแกรม
7	courseCreateDate	TIMESTAMP	-	CURRENT_TIMESTAMP	วันที่และเวลาสร้าง	2022-07-04 11:06:19.000
8	courseUpdateDate	TIMESTAMP	-	CURRENT_TIMESTAMP	วันที่และเวลาอัปเดต	2022-07-04 11:06:19.000
9	courseCreateBy	VAR CHAR	20	-	ชื่อผู้สร้าง	peerasak
10	courseUpdateBy	VAR CHAR	20	-	ชื่อผู้อัปเดต	peerasak
11	courseStatus	ENUM	-	'active'	สถานะของหลักสูตร	peerasak

ตารางที่ ข-10 รายละเอียดตาราง problems

ชื่อตาราง		problems				
คำอธิบาย		ตารางเก็บข้อมูลโจทย์ปัญหา				
คีย์หลัก (PK)		problemId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
1	problemId	INT	10	-	ลำดับโจทย์ปัญหา	1

ชื่อตาราง		problems				
คำอธิบาย		ตารางเก็บข้อมูลโจทย์ปัญหา				
คีย์หลัก (PK)		problemId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
2	problemTitle	VARCHAR	255	NULL	ชื่อหัวข้อโจทย์ปัญหา	รับค่าตัวเลข
3	problem Description	TEXT	-	-	รายละเอียดของโจทย์ปัญหา	จงเขียนโปรแกรมรับค่าตัวเลข
4	problem CreateDate	TIME STAMP	-	CURRENT _TIME STAM	วันที่และเวลาสร้าง	2024-04-04 07:55:33
5	problem UpdateDate	TIME STAMP	-	CURRENT _TIME STAM	วันที่และเวลาอัปเดต	2024-04-04 07:55:33
6	problem CreateBy	VAR CHAR	20	NULL	ชื่อผู้สร้างโจทย์ปัญหา	Adam
7	problem UpdateBy	VAR CHAR	20	NULL	ชื่อผู้อัปเดตโจทย์ปัญหา	Adam
8	problemStatus	ENUM	-	‘active’	สถานะ	‘active’
9	problem VoteScore	FLOAT	-	NULL	ค่าเฉลี่ยคะแนนโหวต	2.5
10	problem VoteAmount	INT	11	NULL	จำนวนผู้โหวต	35
11	problem TimeLimit	INT	11	NULL	จำกัดเวลาในการรัน	60
12	problemDifficult	ENUM	-	‘easy’	ความซับซ้อนของโจทย์ปัญหา	‘easy’

ชื่อตาราง		problems				
คำอธิบาย		ตารางเก็บข้อมูลโจทย์ปัญหา				
คีย์หลัก (PK)		problemId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
13	IsForAdaptive	TINYINT	1	0	การอนุญาตให้นำไปใช้ในการทำโจทย์ปัญหาแบบปรับได้	0

ตารางที่ ข-11 รายละเอียดตาราง problemCodeFiles

ชื่อตาราง		problemCodeFiles				
คำอธิบาย		ตารางเก็บข้อมูลไฟล์ซอร์สโค้ดและภาษาของโจทย์ปัญหา				
คีย์หลัก (PK)		ProblemCodeFileId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
1	Problem CodeFileId	INT	10	-	ลำดับข้อมูลไฟล์ซอร์สโค้ด	1
2	problemId	INT	10	-	ลำดับโจทย์ปัญหา	1
3	filePath	VAR CHAR	255	NULL	ชื่อของไฟล์ซอร์สโค้ด	Main.cpp
4	language	VAR CHAR	255	NULL	ภาษาของไฟล์ซอร์สโค้ด	CPP

ตารางที่ ข-12 รายละเอียดตาราง solutions

ชื่อตารางภาษาอังกฤษ		solutions				
คำอธิบาย		ตารางเก็บข้อมูลความซับซ้อนของไฟล์ซอร์สโค้ด				
คีย์หลัก (PK)		solutionId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
1	solutionId	INT	10	-	ลำดับข้อมูลความซับซ้อนของไฟล์ซอร์สโค้ด	1
2	problemId	INT	10	-	ลำดับโจทย์ปัญหา	1
3	language	VARCHAR	255	NULL	ภาษาของไฟล์ซอร์สโค้ด	CPP
4	reportLizardNr	INT	11	NULL	จำนวนฟังก์ชัน	1
5	reportLizardNCSS	DECIMAL	10,2	NULL	จำนวนบรรทัด	10.00
6	reportLizardCCN	DECIMAL	10,2	NULL	ความซับซ้อนของโจทย์ปัญหา	1.00
7	problemCodeFileId	INT	10	-	ลำดับข้อมูลไฟล์ซอร์สโค้ด	1

ตารางที่ ข-13 รายละเอียดตาราง adaptive

ชื่อตารางภาษาอังกฤษ		adaptive				
คำอธิบาย		ตารางเก็บข้อมูลปัจจุบันในการทำโจทย์ปัญหาแบบปรับได้				
คีย์หลัก (PK)		adaptiveId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
1	adaptiveId	INT	11	-	ลำดับข้อมูลปัจจุบันในการทำโจทย์ปัญหา	1
2	problemId	INT	10	-	ลำดับโจทย์ปัญหา	1
3	userId	INT	10	-	ลำดับของผู้ใช้	1
4	loc	INT	11	NULL	จำนวนบรรทัด	1
5	complexity	INT	11	NULL	ความซับซ้อนของคำตอบ	10.00
6	attempt	INT	11	NULL	ครั้งที่ส่ง	1.00
7	currentLevel	INT	11	NULL	ระดับปัจจุบันของผู้ใช้	1
8	path	TEXT	-	NULL	ที่อยู่ของไฟล์การทำโจทย์ปัญหา	..\storage\adaptive\65160000\20
9	startDate	TIMESTAMP	-	CURRENT_TIMESTAM	วันที่และเวลาที่เริ่มทำโจทย์ปัญหา	2024-04-04 07:55:33
10	endDate	TIMESTAMP	-	CURRENT_TIMESTAM	วันที่และเวลาที่ทำโจทย์ปัญหาสำเร็จ	2024-04-04 07:55:33

ชื่อตารางภาษาอังกฤษ		adaptive				
คำอธิบาย		ตารางเก็บข้อมูลปัจจุบันในการทำโจทย์ปัญหาแบบปรับได้				
คีย์หลัก (PK)		adaptiveId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
11	createdAt	TIME STAMP	-	CURRENT _TIME STAM	วันที่และเวลาที่สร้าง	2024-04-04 07:55:33
12	updatedAt	TIME STAMP	-	CURRENT _TIME STAM	วันที่และเวลาที่อัปเดต	2024-04-04 07:55:33
13	correct	DECIMAL	10,2	NULL	ความถูกต้องของคำตอบ	1.00
14	nextRecommend	INT	10	NULL	ระดับโจทย์ถัดไปที่เหมาะสม	2

ตารางที่ ข-14 รายละเอียดตาราง adaptiveUsersLogs

ชื่อตารางภาษาอังกฤษ		adaptiveUsersLogs				
คำอธิบาย		ตารางเก็บข้อมูลประวัติในการทำโจทย์ปัญหาแบบปรับได้				
คีย์หลัก (PK)		adaptiveUsersLogsId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
1	adaptive UsersLogsId	INT	11	-	ลำดับข้อมูลประวัติในการทำโจทย์ปัญหา	1
2	problemId	INT	11	-	ลำดับโจทย์ปัญหา	1
3	difficult	ENUM	-	NULL	ความซับซ้อนของโจทย์ปัญหา	‘easy’

ชื่อตารางภาษาอังกฤษ		adaptiveUsersLogs				
คำอธิบาย		ตารางเก็บข้อมูลประวัติในการทำโจทย์ปัญหาแบบปรับได้				
คีย์หลัก (PK)		adaptiveUsersLogsId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
4	language	VAR CHAR	255	NULL	ภาษาของไฟล์ซอร์สโค้ด	CPP
5	loc	INT	11	NULL	จำนวนบรรทัด	1
6	complexity	INT	11	NULL	ความซับซ้อนของคำตอบ	10.00
7	attempt	INT	11	NULL	ครั้งที่ส่ง	1.00
8	currentLevel	INT	11	NULL	ระดับปัจจุบันของผู้ใช้	1
9	path	TEXT	-	NULL	ที่อยู่ของไฟล์การทำโจทย์ปัญหา	..\storage\adaptive\65160000\20
10	createdAt	TIME STAMP	-	CURRENT _TIME STAM	วันที่และเวลาที่สร้าง	2024-04-04 07:55:33
11	updatedAt	TIME STAMP	-	CURRENT _TIME STAM	วันที่และเวลาที่อัปเดต	2024-04-04 07:55:33
12	userId	INT	10	-	ลำดับของผู้ใช้	1
13	startDate	TIME STAMP	-	CURRENT _TIME STAM	วันที่และเวลาที่เริ่มทำโจทย์ปัญหา	2024-04-04 07:55:33
14	endDate	TIME STAMP	-	CURRENT _TIME STAM	วันที่และเวลาที่ทำ	2024-04-04 07:55:33

ชื่อตารางภาษาอังกฤษ		adaptiveUsersLogs				
คำอธิบาย		ตารางเก็บข้อมูลประวัติในการทำโจทย์ปัญหาแบบปรับได้				
คีย์หลัก (PK)		adaptiveUsersLogsId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
					โจทย์ปัญหาสำเร็จ	
15	correct	DECIMAL	10,2	NULL	ความถูกต้องของคำตอบ	1.00
16	nextRecommend	INT	10	NULL	ระดับโจทย์ถัดไปที่เหมาะสม	2

ตารางที่ ข-15 รายละเอียดตาราง adaptiveCompileLogs

ชื่อตารางภาษาอังกฤษ		adaptiveCompileLogs				
คำอธิบาย		ตารางเก็บข้อมูลประวัติในการส่งโจทย์ปัญหาแบบปรับได้				
คีย์หลัก (PK)		adaptiveId				
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
1	adaptive CompileLogsId	INT	11	-	ลำดับข้อมูลประวัติในการทำโจทย์ปัญหา	1
2	problemId	INT	11	-	ลำดับโจทย์ปัญหา	1
3	compilelogScore	DECIMAL	10,2	NULL	คะแนน	100.00
4	Compilelog SubmitNo	INT	11	NULL	ครั้งที่ส่ง	1
5	Compilelog TestResult	TEXT	-	NULL	ผลลัพท์การรันสำเร็จ	[{"output":"Input: 5 x 1 = 5}]

ชื่อตารางภาษาอังกฤษ	adaptiveCompileLogs					
คำอธิบาย	ตารางเก็บข้อมูลประวัติในการส่งโจทย์ปัญหาแบบปรับได้					
คีย์หลัก (PK)	adaptiveId					
ลำดับ	**ชื่อคอลัมน์**	**ประเภท**	**ขนาด**	**กำหนดค่า**	**รายละเอียด**	**ตัวอย่างข้อมูล**
6	Compilelog ErrorMessage	TEXT	-	NULL	ผลลัพท์การรันผิดพลาด	error: expected ‘;’ before ‘return’ return 0;
7	compilelogLanguage	VAR CHAR	255	NULL	ภาษาของไฟล์ซอร์สโค้ด	CPP
8	loc	INT	11	NULL	จำนวนบรรทัด	1
9	complexity	INT	11	NULL	ความซับซ้อนของคำตอบ	10.00
10	createdAt	TIME STAMP	-	CURRENT _TIME STAM	วันที่และเวลาที่สร้าง	2024-04-04 07:55:33
11	updatedAt	TIME STAMP	-	CURRENT _TIME STAM	วันที่และเวลาที่อัปเดต	2024-04-04 07:55:33
12	userId	INT	10	-	ลำดับของผู้ใช้	1
13	Compilelog Status	VAR CHAR	255	NULL	สถานะการรัน	SUCCESS

บรรณานุกรม

Barbosa, P. L. S., Carmo, R. A. F. do, Gomes, J. P. P., & Viana, W. (2024). Adaptive learning in computer science education: A scoping review. *Education and Information Technologies*, *29*(8), 9139–9188. https://doi.org/10.1007/s10639-023-12066-z

Barr, V., & Trytten, D. (2016). Using turing's craft codelab to support CS1 students as they learn to program. *ACM Inroads*, *7*(2), 67–75. https://doi.org/10.1145/2903724

Bennani, S., Maalel, A., & Ben Ghezala, H. (2022). Adaptive gamification in E-learning: A literature review and future challenges. *Computer Applications in Engineering Education*, *30*(2), 628–642. https://doi.org/10.1002/cae.22477

Chidamber, S. R., & Kemerer, C. F. (1994). A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, *20*(6), 476–493. https://doi.org/10.1109/32.295895

Chung, C.-Y., & Hsiao, I.-H. (2022). Semantic Modeling of Programming Practices with Local Knowledge Graphs: The Effects of Question Complexity on Student Performance. *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners' and Doctoral Consortium: 23rd International Conference, AIED 2022, Durham, UK, July 27–31, 2022, Proceedings, Part II*, 245–249. https://doi.org/10.1007/978-3-031-11647-6_44

Cyclomatic Complexity. (2018, June 20). *GeeksforGeeks*. https://www.geeksforgeeks.org/cyclomatic-complexity/

Cyclomatic complexity. (2024). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Cyclomatic_complexity&oldid=1221707517

Edwards, S. H., & Perez-Quinones, M. A. (2008). Web-CAT: automatically grading programming assignments. *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education*, 328. https://doi.org/10.1145/1384271.1384371

Evan You. (n.d.). *Vue.js—The Progressive JavaScript Framework | Vue.js*. Retrieved May 16, 2023, from https://vuejs.org/

Fenton, N., & Neil, M. (2000). Software Metrics: Roadmap. *Proceedings of the Conference on the Future of Software Engineering*. https://doi.org/10.1145/336512.336588

Gill, G. K., & Kemerer, C. F. (1991). Cyclomatic complexity density and software maintenance productivity. *IEEE Transactions on Software Engineering*, *17*(12), 1284–1288. IEEE Transactions on Software Engineering. https://doi.org/10.1109/32.106988

Heffernan, N. T., & Heffernan, C. L. (2014). The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning and Teaching. *International Journal of Artificial Intelligence in Education*, *24*(4), 470–497. https://doi.org/10.1007/s40593-014-0024-x

How to Measure Code Complexity. (n.d.). TIOBE. Retrieved May 15, 2024, from https://www.tiobe.com/knowledge/article/controlling-code-complexity/

IBM Documentation. (2021, March 8). https://www.ibm.com/docs/en/raa/6.1?topic=metrics-cyclomatic-complexity

Jain, Y., & Cunningham, K. (2023). Towards Methods for Identifying High-Quality Domain-Specific Programming Plans. *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 2*, 18–19. https://doi.org/10.1145/3568812.3603478

Kasto, N., & Whalley, J. (2013). Measuring the difficulty of code comprehension tasks using software metrics. *Proceedings of the Fifteenth Australasian Computing Education Conference - Volume 136*, 59–65.

Lincke, R., Lundberg, J., & Löwe, W. (2008a). Comparing software metrics tools. *Proceedings of the 2008 International Symposium on Software Testing and Analysis*, 131–142. https://doi.org/10.1145/1390630.1390648

Lincke, R., Lundberg, J., & Löwe, W. (2008b). Comparing software metrics tools. In *ISSTA'08: Proceedings of the 2008 International Symposium on Software Testing and Analysis 2008* (p. 142). https://doi.org/10.1145/1390630.1390648

Lohr, D., Berges, M., Chugh, A., & Striewe, M. (2024). *Adaptive Learning Systems in Programming Education: A Prototype for Enhanced Formative Feedback*. https://doi.org/10.18420/delfi2024_57

Luan, H., & Tsai, C.-C. (2021). A Review of Using Machine Learning Approaches for Precision Education. *Educational Technology & Society*, *24*(1), 250–266. JSTOR.

MariaDB. (n.d.). *Open Source Database (RDBMS) for the Enterprise*. MariaDB. Retrieved May 16, 2023, from https://mariadb.com/

Mittal, J. P., Bhatia, P., & Mittal, H. (2009). Software maintenance productivity assessment using fuzzy logic. *SIGSOFT Softw. Eng. Notes*, *34*(5), 1–4. https://doi.org/10.1145/1598732.1598739

Mohammad T. Alshammari. (2025). Machine Learning-Enabled Personalization of Programming Learning Feedback. *International Journal of Advanced Computer Science and Applications(Ijacsa)*, *16*(2), 1091–1097. https://doi.org/10.14569/IJACSA.2025.01602108

Nuxt. (n.d.). *Documentation · Nuxt*. Nuxt. Retrieved May 16, 2023, from https://nuxt.com/docs

OpenJS Foundation. (n.d.). *Express—Node.js web application framework*. Retrieved May 16, 2023, from https://expressjs.com/

Phakdee, N., & Pianprasit, P. (2023). KruCode: An Online Learning Programming Platform for Programming Course. *2023 27th International Computer Science and Engineering Conference (ICSEC)*, 79–84. https://doi.org/10.1109/ICSEC59635.2023.10329732

Phakdee, N., & Pianprasit, P. (2023). KruCode: An Online Learning Programming Platform for Programming Course. *2023 27th International Computer Science and Engineering Conference (ICSEC)*, 79–84. https://doi.org/10.1109/ICSEC59635.2023.10329732

Rincón-Flores, E. G., López-Camacho, E., Mena, J., & López, O. O. (2020). Predicting academic performance with Artificial Intelligence (AI), a new tool for teachers and

students. *2020 IEEE Global Engineering Education Conference (EDUCON)*, 1049–1054. https://doi.org/10.1109/EDUCON45650.2020.9125141

S. Elnaffar. (2016). Using software metrics to predict the difficulty of code writing questions. *2016 IEEE Global Engineering Education Conference (EDUCON)*, 513–518. https://doi.org/10.1109/EDUCON.2016.7474601

Sandeep Kaur & Navjot Kaur. (n.d.). Software Metrics and Metric Tools- A Review. *International Journal on Recent and Innovation Trends in Computing and Communication*, *3*(4), 2076–2079.

Schnotz, W., & Kürschner, C. (2007). A Reconsideration of Cognitive Load Theory. *Educational Psychology Review*, *19*(4), 469–508. https://doi.org/10.1007/s10648-007-9053-4

Sequelize Contributors. (n.d.). *Sequelize*. Retrieved May 16, 2023, from https://sequelize.org/

Sheard, J., Simon, Carbone, A., Chinn, D., Clear, T., Corney, M., D'Souza, D., Fenwick, J., Harland, J., Laakso, M.-J., & Teague, D. (2013). *How difficult are exams? A framework for assessing the complexity of introductory programming exams*.

Software Engineering | Case Tools For Software Metrics—Javatpoint. (n.d.). Retrieved May 15, 2024, from https://www.javatpoint.com/software-engineering-case-tools-for-software-metrics

Software Measurement and Metrics. (2019, March 18). *GeeksforGeeks*. https://www.geeksforgeeks.org/software-measurement-and-metrics/

T. J. McCabe. (1976). A Complexity Measure. *IEEE Transactions on Software Engineering*, *SE-2*(4), 308–320. https://doi.org/10.1109/TSE.1976.233837

The OpenJS Foundation. (n.d.). About. Node.Js. Retrieved May 16, 2023, from https://nodejs.org/en/about

Vanlehn, K. (2006). The Behavior of Tutoring Systems. *Int. J. Artif. Intell. Ed.*, *16*(3), 227–265.

Whalley, J., & Kasto, N. (2014). How difficult are novice code writing tasks? A software metrics approach. *Proceedings of the Sixteenth Australasian Computing Education Conference - Volume 148*, 105–112.

Yin, T. (2025). *Terryyin/lizard* [Python]. https://github.com/terryyin/lizard (Original work published 2012)

Ying Tang & Ryan Hare. (2020, June 22). *A Random Forest Model for Personalized Learning in a Narrative Game*. https://doi.org/10.18260/1-2--34041

วิชัย วงษ์ใหญ่, & มารุต พัฒผล. (2564). การเรียนรู้ปรับได้ (Adaptive Learning). บัณฑิตวิทยาลัย มหาวิทยาลัยศรีนครินทรวิโรฒ.

รายงานสรุปการเงิน

สัญญาเลขที่ 7/2567

โครงการวิจัยประเภทเงินรายได้คณะวิทยาการสารสนเทศ

ประจำปีงบประมาณ พ.ศ. 2567

ชื่อโครงการ โครงการวิจัยการพัฒนามอดูลการวัดความซับซ้อนของซอร์สโค้ด สำหรับแพลตฟอร์มการเรียนรู้การเขียนโปรแกรม

(The development of a module for measuring code complexity for learning programming platform)

ชื่อหัวหน้าโครงการวิจัยผู้รับทุน นายพีระศักดิ์ เพียรประสิทธิ์

รายงานในช่วงตั้งแต่วันที่ 23 มกราคม 2567 ถึง 22 มกราคม 2568

ขยายเวลาครั้งที่ 1 ถึงวันที่ 22 กรกฎาคม 2568

ระยะเวลาดำเนินการ 1 ปี

รายรับ

จำนวนเงินที่ได้รับ

งวดที่ 1 (50%) 50,000 บาท เมื่อ 22 มกราคม 2567

งวดที่ 2 (40%) 40,000 บาท เมื่อ 25 กรกฎาคม 2567

งวดที่ 3 (10%) 10,000 บาท เมื่อ มิถุนายน 2568

รวม หนึ่งแสนบาทถ้วน

รายจ่าย

รายการ	**งบประมาณที่ตั้งไว้**	**งบประมาณที่ใช้จริง**	**จำนวนเงินคงเหลือ/เกิน**
1. ค่าจ้างผู้ช่วยนักวิจัย วุฒิการศึกษา ม.6 จำนวน 2 คน จำนวน 5 เดือน อัตรา 8,500 บาท	85,000	85,000	0
2. ค่าตอบแทนคณะวิจัย	10,000	10,000	0
3. ค่าวัสดุสำนักงาน	1,000	943	57
4. ค่าวัสดุคอมพิวเตอร์	4,000	4,000	0
รวมทั้งสิ้น	100,000	100,000	57

....

นายพีระศักดิ์ เพียรประสิทธิ์