



ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอฟต์แวร์สโค์ด:

Back-End

สิริวิชญ์ ฐิติสุนทรลักษณ์

โครงการนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

ปีการศึกษา 2565

ลิขสิทธิ์ของคณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

**SEAPLAG: BACK-END**

**SIRAWIT THITISOONTORNLAK**

**A PROJECT IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE  
BACHELOR'S DEGREE OF SCIENCE IN SOFTWARE ENGINEERING**

**FACULTY OF INFORMATICS, BURAPHA UNIVERSITY**

**2022.**



คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

ใบรับรองโครงการ

หัวข้อโครงการ	ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด: Back-End
ผู้อนุมัติ	นายสิริวิชญ์ ฐิติสุนทรลักษณ์
รหัสประจำตัว	62160162
อาจารย์ที่ปรึกษา	นายพิริศักดิ์ เพียรประสีห์
วันที่สอบ	24 กุมภาพันธ์ 2566

โครงการฉบับนี้ได้ผ่านการเห็นชอบจากคณะกรรมการสอบ  
ให้เป็นโครงการหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาบริหารธุรกิจ แฟร์แวร์

จิราภรณ์ อานัน्द  
(นายจิราภรณ์ อานัน्द)

ประธานกรรมการ

พิริศักดิ์ เพียรประสีห์  
(นายพิริศักดิ์ เพียรประสีห์)

กรรมการ

วันที่ 8 เดือน มีนาคม พ.ศ. 2566

# SEAPlag: SEAP

ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด

นายสิริวิชญ์ ฐิติสุนทรลักษณ์ นางสาวพรนภัส เขียวอิม และนายพีระศักดิ์ เพียรประสิทธิ์

สาขาวิชาวิเคราะห์ซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา



## Abstract

ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดพัฒนาขึ้น เพื่อวิเคราะห์ความคล้ายคลึงของชอร์สโค้ดของนักศึกษาที่ต้องการตรวจสอบการติดลอกชอร์สโค้ดของนักศึกษา โดยมอดูลของระบบประกอบด้วย

- มอดูลเน้าเข้าข้อมูลชอร์สโค้ด
- มอดูลแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด
- มอดูลส่งออกผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด
- มอดูลกำหนดขอบเขตการแสดงผลลัพธ์
- มอดูลแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน

ผู้ใช้งานสามารถลงทะเบียนภาษาโปรแกรม เช่น Java, C, C++, C#, Python เพื่อตรวจสอบความคล้ายคลึงของชอร์สโค้ด หลังจากประเมินผลลัพธ์จะแสดงผลลัพธ์ที่มีรูปแบบกราฟและตารางข้อมูล ที่ช่วยให้ผู้ใช้งานมองเห็นความเชื่อมโยงความคล้ายคลึงระหว่างชอร์สโค้ดที่คล้ายคลึงกันได้โดยง่าย และสามารถเลือกดูระดับความคล้ายคลึงที่สนใจได้ และสามารถดูผลการเปรียบเทียบชอร์สโค้ดในส่วนที่คล้ายกันได้ง่ายขึ้น

## Tools



เฟรมเวิร์ก (Frameworks)



Node.js



JPlag



Monaco



กราฟแบบแรงดึงดูด (Force-directed graph)

ไลบรารี (Libraries)

## Results



จำนวนไฟล์เดิม	จำนวนไฟล์ในแต่ละไฟล์เดิม	จำนวนไฟล์รวม	เวลาประมวลผล (วินาที)
10	1	10	2.75
50	1	50	4.23
100	1	100	6.00
10	3	30	3.06
50	3	150	5.96
100	3	300	7.96

ผลลัพธ์จากการวัดประสิทธิภาพการตรวจสอบความคล้ายคลึงของชอร์สโค้ดโดยระบบ พบร่วมกับชั้วโมงในการประมวลผลในแต่ละไฟล์เดิม ในการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

## Methodology



## Conclusion

ระบบสามารถตรวจสอบความคล้ายคลึงของชอร์สโค้ด โดยแสดงผลลัพธ์ในรูปแบบของกราฟและตาราง ผู้ใช้งานสามารถมองเห็นความเชื่อมโยงของผลลัพธ์ตรวจสอบความคล้ายคลึงและสามารถดูชื่อชอร์สโค้ดและสามารถเปรียบเทียบชอร์สโค้ดในส่วนที่คล้ายคลึงกันได้ หลังจากพัฒนาระบบเสร็จสิ้นแล้ว ระบบจะถูกนำไปให้กับทางมหาวิทยาลัยบูรพา ซึ่งระบบจะได้รับการพัฒนาเพิ่มเติมความสามารถต่อไป

## Recommendations

เพิ่มเติมความสามารถของ JPlag เพื่อให้ระบบสามารถตรวจสอบชอร์สโค้ดที่ถูกเขียนขึ้นโดยภาษาต่างๆ ได้อย่างครอบคลุมมากยิ่งขึ้น รวมถึงเพิ่มในส่วนของการจัดเก็บข้อมูลของผู้ใช้งาน ในกรณีที่ผู้ใช้งานต้องการจัดเก็บข้อมูลชอร์สโค้ดที่ผ่านการประมวลผลโดยระบบแล้ว

## ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด

### SEAPlag

สิริวิชญ์ ฐิติสุนทรลักษณ์ พرنภัส เอียวอิน พีระศักดิ์ เพียรประสีทธิ์

คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา อำเภอเมือง จังหวัดชลบุรี

62160162@go.buu.ac.th, 62160158@go.buu.ac.th, peersak@buu.ac.th

#### บทคัดย่อ

หลายสถาบันการศึกษาต่างก็มีรายวิชาที่เปิดสอนเกี่ยวกับพื้นฐานการเขียนโปรแกรมคอมพิวเตอร์ ซึ่งการจัดการเรียนการสอนมีทั้งการเรียนรู้ทฤษฎีและการฝึกปฏิบัติการด้วยการฝึกเขียนโปรแกรมตามโจทย์ปัญหา ซึ่งในขั้นตอนการทำโจทย์ปัญหานี้เองที่พบว่าอาจมีนักศึกษาบางกลุ่มทำการคัดลอกชอร์สโค้ดกัน ซึ่งเป็นสาเหตุของปัญหาที่ทำให้นักศึกษาเขียนโปรแกรมไม่ได้และส่งผลกระทบต่อตัวนักศึกษาเอง ดังนั้น อาจารย์ผู้สอนส่วนใหญ่มักมีการกำกับติดตาม และอาจมีการนำเครื่องมือเหล่านี้ยังไม่สะดวกสบายต่อการใช้งานมากนัก เนื่องจากข้อมูลมักแสดงออกมากในรูปแบบตารางข้อมูล ดังนั้น ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด (SEAPlag: SEAP) จึงถูกพัฒนาขึ้นเพื่ออำนวยความสะดวกต่ออาจารย์ผู้สอนในสถาบันการศึกษาที่ต้องการตรวจสอบการคัดลอกชอร์สโค้ดของนักศึกษา โดยมอดูลหลักของระบบประกอบไปด้วย มอดูลนำเข้าข้อมูลชอร์สโค้ด มอดูลแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด มอดูลส่งออกผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด มอดูลกำหนดขอบเขตการแสดงผลลัพธ์ และมอดูลแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน ผลลัพธ์ของระบบจะอยู่ในรูปแบบของกราฟความสัมพันธ์ และตารางร้อยละความคล้ายคลึงของชอร์สโค้ด ผู้ใช้งานสามารถดูข้อมูลการเปรียบเทียบชอร์สโค้ดในส่วนที่คล้ายกัน

ได้ง่ายขึ้น ซึ่งการพัฒนาระบบนี้ได้มีการใช้เครื่องมือต่างๆ ในการพัฒนาระบบ คือ Angular, Node.js และ Visual Studio Code

**คำสำคัญ:** Source Code, Plagiarism, Comparison

#### Abstract

Many educational institutions offer courses on the fundamentals of computer programming. The teaching and learning include both theoretical and practical training by writing programs to solve the problem. In the process of doing this problem, it was discovered that some groups of students might copy the source code. This causes problems that make students unable to program and affects students themselves. Therefore, most teachers tend to supervise, monitor and may use tools to verify source code copying. These tools are not very comfortable to use because often data displayed in tabular form. Therefore, the SEAPlag was developed for the convenience of teachers in academic who wants to check the similarity of the source code. The main system's module consists of import source code module, output graph module, export output module,

scope output module, and compare source code module. The results of the system are in the form of correlation graphs and source code similarity percentage tables. Users can choose to view source code comparison data on similar sections. The tools developed in this system consist of Angular, Node.js, and Visual Studio Code.

**Keywords:** Source Code, Plagiarism, Comparison

## 1. ບໜ້າ

ความก้าวหน้าทางด้านเทคโนโลยีและสื่อโซเชียลมีเดีย  
นำมายังความสะดวกในการดำเนินชีวิตในด้านต่างๆ ไม่ว่า  
จะเป็นในด้านของเศรษฐกิจ การคมนาคม โทรคมนาคม  
หรือกระทั้งในด้านของการศึกษา ซึ่งเป็นสิ่งสำคัญ  
ที่ช่วยผลักดันให้นักศึกษาผู้แสวงหาความรู้ สามารถเข้าถึง  
ข้อมูลข่าวสารที่มีปริมาณมหาศาลและมีความหลากหลาย  
ของโครงสร้างเนื้อหาได้อย่างไม่มีที่สิ้นสุด  
แต่ในขณะเดียวกัน ความก้าวหน้าก็มาพร้อมกับช่องทาง  
ที่ช่วยให้นักศึกษางอกกลุ่ม สามารถกระทำการอะไร  
บางอย่างอันเป็นการเอาเปรียบทหรือละเมิดผลงานของผู้อื่น  
ซึ่งในที่นี้ก่อตัวถึงการคัดลอกชอร์สโค้ดภายใน  
สถาบันการศึกษา ซึ่งเป็นปัญหาที่สำคัญ เนื่องจาก  
ถ้านักศึกษาทำการคัดลอกชอร์สโค้ดกัน ซึ่งเป็นต้นเหตุ  
ของการสร้างลักษณะนิสัยในการคัดลอกผลงานของผู้อื่น  
ทั้งโดยได้รับอนุญาตหรือไม่ได้รับอนุญาตจากผู้เป็นเจ้า  
ของ นอกจาคนั้น ปัญหาสำคัญอีกประการหนึ่ง  
คือหากนักศึกษาทำการคัดลอกชอร์สโค้ดกันแล้ว นักศึกษา  
เหล่านั้นก็ไม่เกิดการเรียนรู้ ซึ่งในปัจจุบันมีเครื่องมือสำหรับ  
ตรวจสอบการความคล้ายคลึงของชอร์สโค้ด แต่เครื่องมือ  
ที่มีอยู่ในปัจจุบันนั้น อาจมีค่าใช้จ่ายในการใช้งาน  
หรือซอฟต์แวร์ที่เป็นโอเพนซอร์สที่สามารถใช้งานได้ฟรี  
ก็ยังมีการใช้งานที่ค่อนข้างยุ่งยากและผลลัพธ์ของการ  
ตรวจสอบความคล้ายคลึงของชอร์สโค้ดนั้น มีรายละเอียด  
ของผลลัพธ์ที่ผู้ใช้งานต้องใช้เวลาในการทำความเข้าใจ

## 2. วัตถุประสงค์ในการจัดทำโครงงาน

- 2.1) เพื่อพัฒนาระบบสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ดให้สามารถทำงานได้อย่างมีประสิทธิภาพ
  - 2.2) เพื่ออำนวยความสะดวกให้กับผู้ใช้งานที่ต้องการใช้เครื่องมือที่เหมาะสมสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ด
  - 2.3) เพื่อแสดงให้เห็นถึงความเชื่อมโยงและความสัมพันธ์ของผลลัพธ์จากการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

### 3. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ผู้ดำเนินโครงการได้ทำการวิเคราะห์ปัญหาของการคัดลอกซอร์สโค้ดภายใต้สถานการณ์การศึกษา แล้วจึงทำการการศึกษาค้นคว้าหาข้อมูลและเครื่องมือที่เหมาะสมสำหรับใช้ในการพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอฟต์แวร์ โดยเริ่มศึกษาจากพฤติกรรมการคัดลอกซอร์สโค้ดของนักศึกษาและรูปแบบของการคัดลอก ซอฟต์แวร์ จากนั้นจึงทำการศึกษาการทำงานของเครื่องมือที่จะใช้ในการตรวจสอบความคล้ายคลึงของซอฟต์แวร์โดยมีรายละเอียด ดังนี้

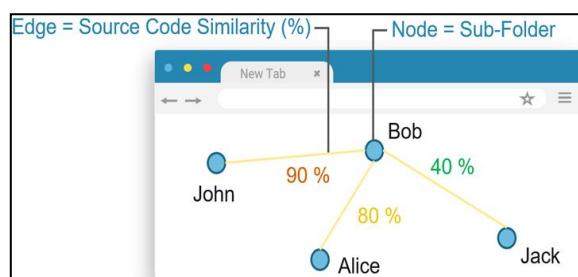
### 3.1 ความคล้ายคลึงของชอร์สโค้ด

การตรวจสอบส่วนที่เหมือนกันหรือคล้ายคลึงกันระหว่างชุดข้อมูลของอร์สโค้ด 2 ชุด โดยการตรวจสอบความคล้ายคลึงของชอร์สโค้ดนี้ มีวัตถุประสงค์เพื่อค้นหาชอร์สโค้ดในส่วนที่เหมือนกันหรือคลดลอกกันมา อันเป็นวิธีการที่จะช่วยให้เห็นถึงความสัมพันธ์ระหว่างชุดข้อมูลชอร์สโค้ดชุดนั้นๆ ซึ่งผลลัพธ์ของการแสดงความคล้ายคลึงของชอร์สโค้ดจะอยู่ในรูปของตัวเลข ซึ่งเป็นตัวเลขที่แสดงถึงร้อยละความคล้ายคลึงของชอร์สโค้ดระหว่างชุดข้อมูลชอร์สโค้ด 2 ชุด อันเป็นข้อมูลเพื่อผู้ใช้งานสามารถทำความเข้าใจว่าข้อมูลชอร์สโค้ดทั้ง 2 ชุดนั้นมีความสัมพันธ์กันอย่างไร ซึ่งระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดนี้ ได้ถูกออกแบบมาเพื่อให้สามารถตรวจสอบชอร์สโค้ดที่ถูกเขียนขึ้นโดยภาษาโปรแกรมภาษาเดียวกันเท่านั้น ซึ่งในการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

ระบบจะนำไฟล์ข้อมูลชอร์สโค้ดจำนวน 2 ไฟล์ มาเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน

### 3.2 Data Visualization

การแปลงข้อมูลในลักษณะของข้อความให้มาอยู่ในรูปแบบของแผนภาพ กราฟ หรือวิดีโอ มีวัตถุประสงค์เพื่อใช้ในการนำเสนอความสัมพันธ์ของข้อมูลชุดนั้นๆ ตามความต้องการของผู้จัดทำ ทำให้ผู้รับฟังสามารถทำความเข้าใจข้อมูลได้อย่างรวดเร็วและชัดเจน โดยเฉพาะอย่างยิ่งกับชุดข้อมูลที่มีข้อมูลจำนวนมากและมีการเปลี่ยนแปลงของข้อมูลอย่างต่อเนื่อง ซึ่งในปัจจุบันนี้ มีเครื่องมือสำหรับจัดทำ Data Visualization ที่อำนวยความสะดวกให้ผู้จัดทำอยู่มากมาย และหลากหลาย ทั้งแบบที่ไม่มีค่าใช้จ่ายเพิ่มเติมและมีค่าใช้จ่ายเพิ่มเติม โดยการแสดงผลลัพธ์จากการตรวจสอบความคล้ายคลึงของระบบ ผู้ดำเนินโครงการได้เลือกการแสดงผลในรูปแบบของกราฟความสัมพันธ์ ซึ่งเป็นรูปแบบที่เหมาะสมสำหรับให้ผู้ใช้งานทำความเข้าใจผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด รวมถึงการแสดงผลลัพธ์ในรูปแบบของตารางผลลัพธ์ความคล้ายคลึงของชอร์สโค้ด สำหรับให้ผู้ใช้งานดูรายละเอียดของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด โดยผู้ดำเนินโครงการได้กำหนดรูปแบบการแสดงผลลัพธ์ของระบบ ดังนี้



ภาพที่ 1 การแสดงผลลัพธ์ในรูปแบบกราฟ

จากภาพที่ 1 Node หมายถึง ไฟล์เดอร์หรือไดเรกทอรี ย่อย ซึ่งใช้แทนผู้เป็นเจ้าของชอร์สโค้ด ชุดนั้นๆ โดยแต่ละ Node จะมีการแสดงชื่อของผู้เป็นเจ้าของชุดชอร์สโค้ดกำกับ เอาไว้ และ Edge หมายถึง ความสัมพันธ์ระหว่าง Node ซึ่งในที่นี้คือร้อยละความคล้ายคลึงกันระหว่างชุดชอร์สโค้ด (Source Code Similarity)

Matched	Source Code Similarity (%)
Bob - Alice	100.00
Bob - Jack	80.00
Alice - Jack	50.00

ภาพที่ 2 การแสดงผลลัพธ์ในรูปตาราง

จากภาพที่ 2 คอลัมน์ Matched คือ ความสัมพันธ์ระหว่างผู้เป็นเจ้าของชุดข้อมูลชอร์สโค้ดที่ผ่านการตรวจสอบความคล้ายคลึงของชอร์สโค้ด และคอลัมน์ Source Code Similarity (%) คือ ผลลัพธ์ร้อยละความคล้ายคลึงกันของชุดข้อมูลชอร์สโค้ดที่ผ่านการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

### 3.3 JPlag

ไลบรารีที่ถูกสร้างขึ้นเพื่อใช้สำหรับประมวลการคัดลอกชอร์สโค้ด โดยไลบรารีจะทำการตรวจสอบชอร์สโค้ดที่ผู้ใช้งานต้องการ ผ่านการนำเข้าข้อมูลชอร์สโค้ด (Import Data) และส่งออกผลลัพธ์ของการตรวจสอบในรูปแบบข้อมูลที่อยู่ในโครงสร้างข้อมูลแบบ JavaScript Object Notation (JSON) ซึ่งไลบรารีรองรับการตรวจสอบในรูปแบบของชอร์สโค้ดที่ถูกเขียนขึ้นจากหลากหลายภาษา ยกตัวอย่างเช่น Java, C++, C#, Go, Python, Scala และ Swift เป็นต้น ซึ่ง JPlag เป็นเครื่องมือสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ดที่ผู้ดำเนินโครงการเลือกใช้ เนื่องจาก JPlag รองรับภาษาโปรแกรมจำนวนมากและเป็นเครื่องมือประเภท Open Source ซึ่งสามารถใช้งานได้โดยไม่มีค่าใช้จ่ายเพิ่มเติม ผลลัพธ์จากการประมวลผลโดย JPlag มีรายละเอียดดัง ภาพที่ 3



ภาพที่ 3 การแสดงผลลัพธ์ของ JPlag

## 4. การวิเคราะห์และออกแบบระบบ

การดำเนินโครงการครั้งนี้ผู้ดำเนินโครงการได้รับมอบหมายให้พัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด (SEAPlag) โดยในบทนี้จะกล่าวถึงการวิเคราะห์ และพัฒนาเว็บแอปพลิเคชัน ซึ่งเป็นการกำหนดองค์ประกอบโดยรวมของระบบ ประเภทของผู้ใช้งาน และปฏิสัมพันธ์ระหว่างผู้ใช้งานกับระบบ นอกจากนั้นยังได้นำความรู้ในการเรียนจากวิชาต่างๆ ตลอดระยะเวลาที่ผ่านมาของผู้ดำเนินโครงการเข้ามาประยุกต์ใช้เพื่อพัฒนาระบบ โดยเนื้อหาสาระในบทนี้ ครอบคลุม ในเรื่องของการออกแบบระบบ ลำดับการทำงานของระบบ การออกแบบหน้าจอแสดงผล รวมถึงการทำงานของระบบในแต่ละมอดูล

### 4.1 การออกแบบการทำงานของระบบ

- 4.1.1) วางแผนการดำเนินโครงการ
- 4.1.2) ออกแบบหน้าจอแสดงผลและการทำงานของระบบ
- 4.1.3) พัฒนาระบบ
- 4.1.4) ทดสอบการทำงานของระบบ
- 4.1.5) ส่งมอบและติดตั้งระบบ

### 4.2 การออกแบบการทำงานของระบบ

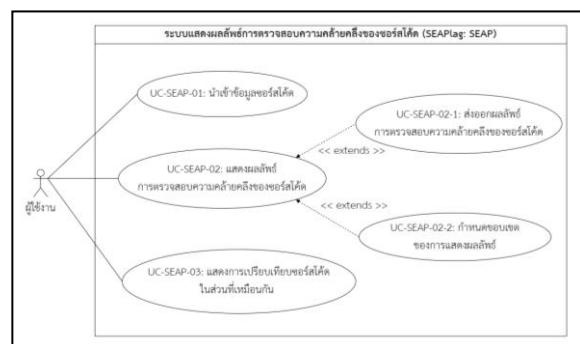
ลำดับการทำงานของระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด คือ รายละเอียดการทำงานของระบบในส่วนของการนำเข้าไฟล์ข้อมูลชอร์สโค้ด จนถึงการแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด โดยมีรายละเอียด ดังภาพที่ 4



ภาพที่ 4 ลำดับการทำงานของระบบ

### 4.3 แผนภาพยูสเคส (Use Case Diagram)

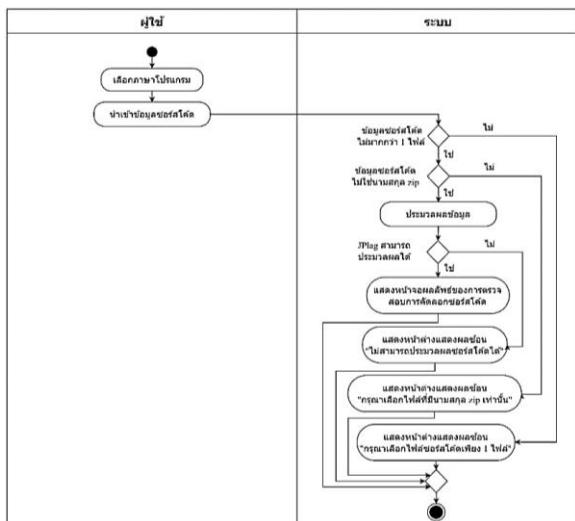
ในการวิเคราะห์และออกแบบโครงสร้างการทำงานของระบบ ผู้ดำเนินโครงการได้จำแนกมอดูลการทำงานของระบบออกเป็น 5 มอดูล และผู้มีส่วนเกี่ยวข้องกับระบบในที่นี้ คือ ผู้ใช้งาน ได้แก่ มอดูลนำเข้าข้อมูลชอร์สโค้ด มอดูลแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด มอดูลกำหนดขอบเขตของการแสดงผลลัพธ์ และ มอดูลแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน ซึ่งผู้ใช้งานระบบในที่นี้ คือ ผู้ที่มีความต้องการในการใช้งานระบบสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ด โดยมีรายละเอียด ดังภาพที่ 5



ภาพที่ 5 แผนภาพยูสเคส

### 4.4 แผนภาพกิจกรรม (Activity Diagram)

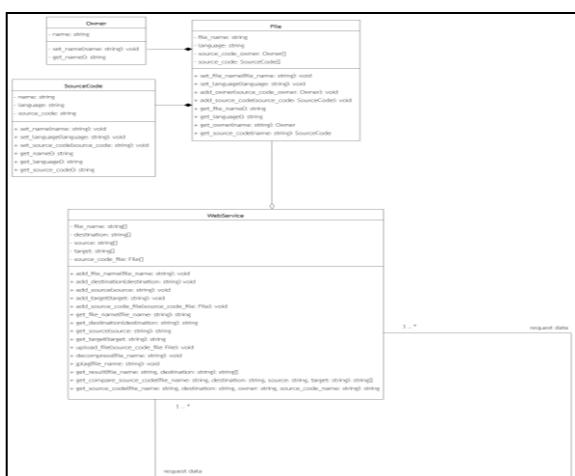
ในส่วนของการออกแบบแผนภาพกิจกรรม เป็นการอธิบายถึงลำดับขั้นตอนการทำงานของแต่ละมอดูล โดยแสดงให้เห็นถึงการปฏิสัมพันธ์กันระหว่างผู้ใช้งานกับระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดอย่างเป็นลำดับขั้นตอน โดยมอดูลนำเข้าข้อมูลชอร์สโค้ด เมื่อผู้ใช้งานต้องการนำเข้าข้อมูลชอร์สโค้ดเพื่อตรวจสอบความคล้ายคลึงของชอร์สโค้ด ซึ่งผู้ใช้งานต้องทำการเลือกภาษาโปรแกรมที่ต้องการตรวจสอบ และทำการนำเข้าข้อมูลชอร์สโค้ดที่ต้องการ เพื่อนำไปใช้ในการตรวจสอบความคล้ายคลึงของชอร์สโค้ดภายในโปรแกรม ต่อไป โดยมีรายละเอียดแผนภาพกิจกรรม ดังภาพที่ 6



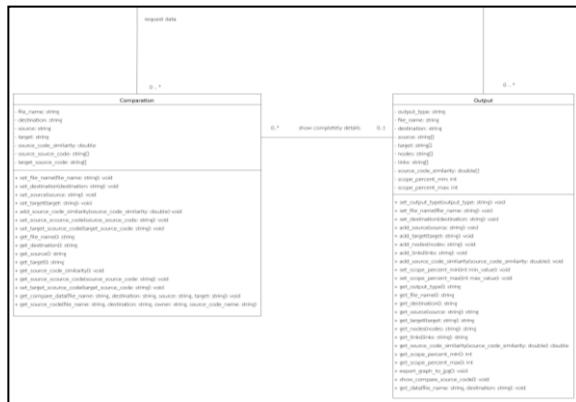
ภาพที่ 6 แผนภาพกิจกรรมมодูลน้ำเข้าเข้า  
ข้อมูลชอร์สโค้ด

#### 4.5 แผนภาพคลาส (Class Diagram)

ผู้ดำเนินโครงการได้ทำการจำแนกองค์ประกอบของหลักของระบบ ออกเป็นทั้งหมด 6 ส่วน ได้แก่ ส่วนของผู้เป็นเจ้าของชอร์สโค้ด ส่วนของชอร์สโค้ด ส่วนของไฟล์ที่รวมข้อมูลชอร์สโค้ด ส่วนบริการของระบบ ส่วนของการแสดงผลลัพธ์ความคล้ายคลึงของชอร์สโค้ด และส่วนของการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน จากนั้นจึงนำองค์ประกอบดังกล่าวบันทึกลงบนแผนภาพคลาสเพื่อใช้อธิบายถึงคุณลักษณะ (Attribute) และความสามารถ (Method) ของแต่ละองค์ประกอบ โดยผู้ดำเนินโครงการได้ระบุความสัมพันธ์ระหว่างองค์ประกอบ (Relationship) ในแผนภาพ โดยใช้เส้นความสัมพันธ์ ซึ่งมีรายละเอียดดังภาพที่ 7



ภาพที่ 7 แผนภาพคลาส



ภาพที่ 7 แผนภาพคลาส (ต่อ)

#### 5. ผลการดำเนินงาน

ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดนั้นมีความสามารถในการนำไลบรารี JPlag มาช่วยตรวจสอบความคล้ายคลึงของชอร์สโค้ดได้ และสามารถนำผลลัพธ์จากการตรวจสอบมาแสดงผลในรูปแบบตารางข้อมูล และรูปแบบกราฟแสดงความสัมพันธ์ นอกจากนี้ผู้ใช้งานยังสามารถกำหนดขอบเขตการตรวจสอบความคล้ายคลึงกันของข้อมูลเพื่อแสดงผลลัพธ์ระดับความคล้ายคลึงที่ผู้ใช้งานได้ทำการกำหนด และส่งออกผลลัพธ์ในรูปแบบไฟล์รูปภาพให้แก่ผู้ใช้งานระบบได้ ซึ่งมีผลลัพธ์ในการพัฒนาระบบ ดังนี้

##### 5.1 มодูลน้ำเข้าข้อมูลชอร์สโค้ด

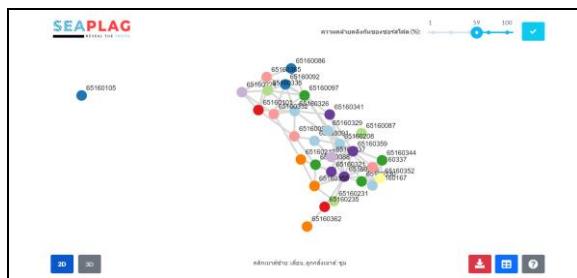
มอดูลสำหรับให้ผู้ใช้งาน นำเข้าชุดข้อมูลชอร์สโค้ดที่ต้องการตรวจสอบความคล้ายคลึงของชอร์สโค้ด โดยข้อมูลที่ผู้ใช้งานต้องเตรียมไฟล์เดอร์หรือไดเรกทอรีสำหรับเก็บชุดข้อมูลชอร์สโค้ด ผู้ใช้งานต้องทำการเข้า zip ไฟล์ข้อมูลชอร์สโค้ด จากนั้นเลือกภาษาโปรแกรม และนำไฟล์เข้าระบบ ซึ่งมีรายละเอียด ดังภาพที่ 8



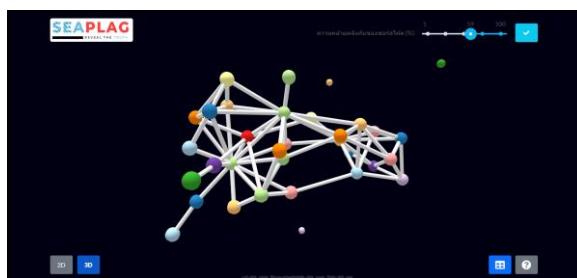
ภาพที่ 8 หน้าจอของมอดูลน้ำเข้าข้อมูลชอร์สโค้ด

## 5.2 ผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด

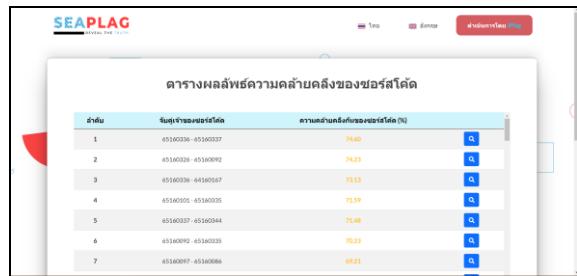
การแสดงผลลัพธ์ในรูปแบบของกราฟ โดยได้มีการกำหนดองค์ประกอบของกราฟซึ่ง ณ ที่นี้ กำหนดให้ Node แทนผู้เป็นเจ้าของชุดข้อมูลชอร์สโค้ด และให้ Edge แทนเส้นความสัมพันธ์โดยใช้ร้อยละความคล้ายคลึงกันของชุดข้อมูลชอร์สโค้ดเป็นข้อมูลสำหรับบอกความสัมพันธ์ระหว่างผู้เป็นเจ้าของชอร์สโค้ด ซึ่งมีการแสดงผลในรูปแบบ 2 มิติ และ 3 มิติ รวมถึงการแสดงผลลัพธ์ในรูปแบบตารางดังภาพที่ 9 ภาพที่ 10 และภาพที่ 11



ภาพที่ 9 หน้าจอมอดูลแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบกราฟ 2 มิติ



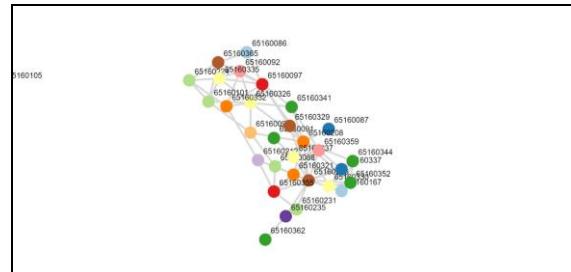
ภาพที่ 10 หน้าจอมอดูลแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบกราฟ 3 มิติ



ภาพที่ 7 หน้าจอแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดรูปแบบตารางข้อมูล

## 5.3 ผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด

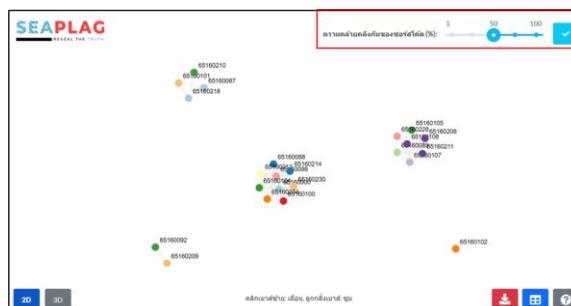
ผู้ใช้ต้องทำการเลือกปุ่ม (Button) “Export” โดยระบบจะทำการประมวลผลเพื่อส่งออกผลลัพธ์ ซึ่งผลลัพธ์จะอยู่ในรูปแบบของไฟล์รูปภาพที่มีนามสกุลไฟล์ .jpg โดยมีการกำหนดชื่อเริ่มต้น คือ “Graph\_SEAPlag” ทั้งนี้ทางผู้ใช้งานจะสามารถส่งออกผลลัพธ์ได้ในรูปแบบของไฟล์ 2 มิติ เพียงอย่างเดียว ซึ่งทางผู้ใช้งานจะได้รับภาพตัวอย่างผลลัพธ์การส่งออกข้อมูลของกราฟ 2 มิติ ดังภาพที่ 12



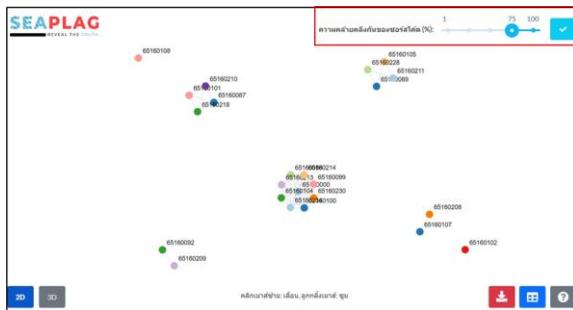
ภาพที่ 8 ตัวอย่างผลลัพธ์การส่งออกข้อมูลผลลัพธ์ของกราฟ 2 มิติ

## 5.4 ผลลัพธ์การกำหนดขอบเขตของการแสดงผลลัพธ์

การกำหนดขอบเขตของการแสดงผลลัพธ์จากการตรวจสอบความคล้ายคลึงของชอร์สโค้ด ในรูปแบบของกราฟ ผู้ใช้งานสามารถเลือกขอบเขตว้อยละความคล้ายคลึงกันของชุดข้อมูลชอร์สโค้ดที่ผ่านการตรวจสอบความคล้ายคลึงโดย JPlag ดังภาพที่ 13 และภาพที่ 14



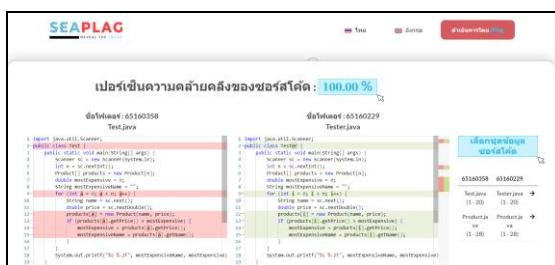
ภาพที่ 13 หน้าจอมอดูลกำหนดขอบเขตการแสดงผลลัพธ์ ภายในขอบเขตว้อยละ 50 ถึง 100



ภาพที่ 14 หน้าจอมดูลกำหนดขอบเขตการแสดงผลลัพธ์ ภายในขอบเขตอยู่ที่ 75 ถึง 100

## 5.5 มดูลแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน

ผู้ใช้งานสามารถเลือกดูข้อมูลชอร์สโค้ดที่คล้ายคลึงกันทั้งในรูปแบบการแสดงผลแบบกราฟและรูปแบบการแสดงผลลัพธ์แบบตาราง ระบบจะแสดงหน้าจอเปรียบเทียบข้อมูลชอร์สโค้ดที่คล้ายคลึงกัน โดยผู้ใช้งานสามารถเลือกชุดข้อมูลชอร์สโค้ดที่ต้องการให้แสดงผลได้ ดังภาพที่ 15



ภาพที่ 15 หน้าจอแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน

## 5.6 ประสิทธิภาพการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

ผู้ดำเนินโครงการได้ทำการวัดประสิทธิภาพการตรวจสอบความคล้ายคลึงของชอร์สโค้ดของระบบ โดยใช้วิธีการจับเวลาเริ่มต้นและผู้ใช้งานอัปโหลดไฟล์ชอร์สโค้ดไปยังระบบ จนถึงขั้นตอนที่ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบของกราฟความสัมพันธ์ ผู้ดำเนินโครงการทำการทดลองบนเครื่องโน้ตบุ๊ค Acer Nitro AN515-54 ระบบปฏิบัติการ Windows 11

CPU INTEL Core i5-9300H 2.40 GHz RAM 8 GB ซึ่งมีรายละเอียด ดังตารางที่ 1  
ตารางที่ 1 ประสิทธิภาพการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

จำนวนไฟล์เดอร์	จำนวนไฟล์ในแต่ละไฟล์เดอร์	จำนวนไฟล์รวม	เวลาประมวลผล (วินาที)
10	1	10	2.75
50	1	50	4.23
100	1	100	6.00
10	3	30	3.06
50	3	150	5.96
100	3	300	7.96

จากการที่ 1 พบว่าประสิทธิภาพการตรวจสอบความคล้ายคลึงของชอร์สโค้ด โดยเริ่มนับตั้งแต่ผู้ใช้งานอัปโหลดไฟล์ชอร์สโค้ด จนกระทั่งระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบตารางความสัมพันธ์ ใช้เวลาไม่นาน ซึ่งหากสังเกตจากตารางที่ 1 ใน colum เวลาประมวลผล (วินาที) พบว่าเวลาที่ใช้ในการประมวลผลของระบบอยู่ค่อนอยู่ในหน่วยของวินาที แม้จะเพิ่มจำนวนไฟล์เดอร์หรือจำนวนไฟล์ชอร์สโค้ดภายในไฟล์เดอร์นั้นๆ ซึ่งเฉลี่ยแล้วระบบใช้เวลาในการประมวลผลต่างกันอยู่ที่ประมาณ 2 วินาที ซึ่งผู้ดำเนินโครงการสามารถสรุปผลได้ว่าระบบสามารถประมวลผลโดยใช้เวลาไม่นานในการแสดงกราฟความสัมพันธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

## 6. สรุปผลการวิจัย

แนวคิดการดำเนินโครงการฉบับนี้ เริ่มต้นที่ผู้ดำเนินโครงการได้มองเห็นถึงปัญหาของการคัดลอกชอร์สโค้ดภายในสถาบันการศึกษาและพบว่าสถาบันการศึกษามีการนำใช้เครื่องมือสำหรับตรวจสอบความคล้ายคลึงของนักศึกษาที่ยังไม่สะดวกต่อการใช้งานหลังจากได้มองเห็นถึงปัญหาที่เกิดขึ้น ผู้ดำเนินโครงการจึงได้

เริ่มคิดค้นระบบที่จะช่วยให้ผู้ใช้งานที่ต้องตรวจสอบความคล้ายคลึงของซอฟต์แวร์สามารถใช้ระบบในการตรวจสอบความคล้ายคลึงของซอฟต์แวร์ได้อย่างสะดวกและสามารถทำความเข้าใจผลลัพธ์ได้โดยใช้เวลาไม่นาน

เมื่อผู้ดำเนินโครงการได้ทำการพัฒนาเสร็จเป็นที่เรียบร้อยแล้วนั้น ผู้ดำเนินโครงการจึงทำการนำระบบมาทวนความต้องการ รวมไปถึงนำมาเทียบกับข้อกำหนดคุณลักษณะซอฟต์แวร์ที่ได้กำหนดไว้ หลังจากนั้นจึงนำมาทดสอบ และปรับแก้ในส่วนที่มีการค้นพบว่ามีข้อผิดพลาดขึ้น โดยระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอฟต์แวร์ (SEAPlag) สามารถตรวจสอบความคล้ายคลึงของซอฟต์แวร์ได้ และมีการนำผลลัพธ์ที่ตรวจสอบได้มาแสดงผลบนระบบให้ผู้ใช้งานสามารถมองเห็นภาพได้ชัดเจน และเข้าใจข้อมูลได้ง่าย ซึ่งจะมีการแสดงผลใน 2 รูปแบบ คือ รูปแบบของกราฟความสัมพันธ์ระหว่างซอฟต์แวร์ที่มีความคล้ายคลึงกันโดยผู้ใช้งานสามารถกำหนดขอบเขต ระดับความคล้ายคลึงที่ต้องการได้ และรูปแบบของตารางข้อมูลที่แสดงรายละเอียดของข้อมูลที่ชัดเจน ทั้งนี้ ผู้ใช้งานยังสามารถบันทึกภาพกราฟที่แสดงผลบนหน้าจอ เข้าสู่เครื่องคอมพิวเตอร์ได้ นอกจากนี้ผู้ใช้งานยังสามารถตรวจสอบความคล้ายคลึงกันของซอฟต์แวร์ของชุดข้อมูล 2 ชุด ที่มีความคล้ายคลึงมาเปรียบเทียบได้ เมื่อผู้ดำเนินโครงการนั้นได้ทำการพัฒนา และปรับแก้จนแล้วเสร็จ ตลอดจนถึงได้ทำการทดสอบระบบเรียบร้อยแล้วนั้น ผู้ดำเนินโครงการจึงได้ทำการส่งมอบระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอฟต์แวร์ (SEAPlag) ให้ทางคณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา เพื่อนำไปใช้งานต่อไป

## 7. เอกสารอ้างอิง

H. Cheers, Y. Lin and S. P. Smith, "Academic Source Code Plagiarism Detection by Measuring Program Behavioral Similarity," in IEEE Access, vol. 9, pp. 50391- 50412, 2021, doi: 10.1109/ACCESS.2021.3069367.

JPlag. [อ ออนไลน์]. เข้าถึงได้จาก <https://github.com/jplag/JPlag>.

Force directed graph. [ออนไลน์]. เข้าถึงได้จาก <https://www.npmjs.com/package/force-graph>.

3D force-directed graph. [ออนไลน์]. เข้าถึงได้จาก <https://github.com/vasturiano/3d-force-graph>.

Tables Bootstrap. [ออนไลน์]. เข้าถึงได้จาก <https://getbootstrap.com/docs/4.1/content/tables/>.

Visual Studio Code. [ออนไลน์]. เข้าถึงได้จาก <https://code.visualstudio.com/docs/setup/getting-started-overview>.

Angular. [ออนไลน์]. เข้าถึงได้จาก <https://angular.io/guide/what-is-angular>.

Node.js. [ออนไลน์]. เข้าถึงได้จาก <https://nodejs.org/en/docs/>.

Express.js. [ออนไลน์]. เข้าถึงได้จาก <https://expressjs.com/en/starter/basic-routing>

Connect an Angular App to a Node.js API. [ออนไลน์]. เข้าถึงได้จาก <https://jasonwatmore.com/post/2022/11/20/angular-nodejs-connect-an-angular-app-to-a-nodejs-api>.

Express JS Tutorial: What is Express in Node JS. [ออนไลน์]. เข้าถึงได้จาก <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js>.

Child process. [ออนไลน์]. เข้าถึงได้จาก <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js>.

Spawnsync. [ออนไลน์]. เข้าถึงได้จาก [https://www.tabnine.com/code/javascript/functions/child\\_process/spawnsync](https://www.tabnine.com/code/javascript/functions/child_process/spawnsync)

## กิจกรรมประจำ

โครงการฉบับนี้ จะไม่สำเร็จลุล่วงหากปราศจากความอนุเคราะห์ และการสนับสนุนของบุคคลเหล่านี้ ซึ่งผู้ดำเนินโครงการได้ขอรับขอบคุณอย่างสูงไว้ ณ โอกาสนี้

- 1) นายพีระศักดิ์ เพียรประสิทธิ์ อาจารย์ที่ปรึกษาโครงการ ที่เคยช่วยเหลือและให้คำแนะนำในทุกๆ เรื่อง ทำให้ผลลัพธ์ของโครงการออกมาในรูปแบบที่ครบถ้วนสมบูรณ์และถูกต้องที่สุด
- 2) นายจิรายุส อับกิง อาจารย์ที่เคยให้แนวทางและคำปรึกษาในการแก้ไขปัญหาที่เกิดขึ้นระหว่างดำเนินโครงการ ทำให้สามารถดำเนินโครงการสำเร็จลุล่วงไปได้ด้วยดี
- 3) นางสาวพรนภัส เอียวอิม เพื่อนที่เคยช่วยเหลือและเปลี่ยนความคิดเห็นและให้การช่วยเหลือในเรื่องต่างๆ รวมถึงรับฟังปัญหาที่เกิดขึ้นในการดำเนินโครงการ

สิริวิชญ์ ฐิติสุนทรลักษณ์

กุมภาพันธ์ 2566

หัวข้อรายงาน	ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด: Back-End
นิสิต	นายสิริวิชญ์ ฐิติสุนทรลักษณ์
รหัสประจำตัว	62160162
อาจารย์ที่ปรึกษา	นายพีระศักดิ์ เพียรประสิทธิ์
ระดับการศึกษา	วิทยาศาสตรบัณฑิต สาขาวิชาศิวกรรมซอฟต์แวร์
คณะ	คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
ปีการศึกษา	2565

## บทคัดย่อ

hely สถาบันการศึกษาต่างก็มีรายวิชาที่เปิดสอนเกี่ยวกับพื้นฐานการเขียนโปรแกรมคอมพิวเตอร์ ซึ่งการจัดการเรียนการสอนมีทั้งการเรียนรู้ทฤษฎีและการฝึกปฏิบัติการด้วยการฝึกเขียนโปรแกรมตามโจทย์ปัญหา ซึ่งในขั้นตอนการทำโจทย์ปัญหานี้เองที่พบว่าอาจมีนักศึกษาบางกลุ่มทำการคัดลอกชอร์สโค้ดกัน ซึ่งเป็นสาเหตุของปัญหาที่ทำให้นักศึกษาเขียนโปรแกรมไม่ได้ และส่งผลกระทบต่อตัวนักศึกษาเอง ดังนั้น อาจารย์ผู้สอนส่วนใหญ่มักมีการกำกับ ติดตาม และอาจมีการนำใช้เครื่องมือสำหรับตรวจสอบการคัดลอกชอร์สโค้ด ซึ่งเครื่องมือเหล่านี้ยังไม่สะดวกสบายต่อการใช้งานมากนัก เนื่องจากข้อมูลมักแสดงออกมาในรูปแบบตารางข้อมูล ดังนั้น ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด (SEAPlag: SEAP) จึงถูกพัฒนาขึ้นเพื่ออำนวยความสะดวก ความสะดวกต่ออาจารย์ผู้สอนในสถาบันการศึกษาที่ต้องการตรวจสอบการคัดลอกชอร์สโค้ด ของนักศึกษา โดยมอดูลหลักของระบบประกอบไปด้วย มอดูลนำเข้าข้อมูลชอร์สโค้ด มอดูลแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด มอดูลกำหนดขอบเขตการแสดงผลลัพธ์ และมอดูลแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน ผลลัพธ์ของระบบจะอยู่ในรูปแบบของกราฟความสัมพันธ์ และตารางร้อยละความคล้ายคลึงของชอร์สโค้ด ผู้ใช้งานสามารถดูข้อมูลการเปรียบเทียบชอร์สโค้ดในส่วนที่คล้ายกันได้ง่ายขึ้น ซึ่งการพัฒนาระบบนี้ได้มีการใช้เครื่องมือต่างๆ ในการพัฒนาระบบ คือ Angular, Node.js และ Visual Studio Code

<b>Project Title</b>	SEAPlag: Back-End
<b>Student Name</b>	Mr. Sirawit Thitisontornlak
<b>Student ID</b>	62160162
<b>Advisor</b>	Mr. Peerasak Pianprasit
<b>Level of Study</b>	Bachelor's degree of Science in Software Engineering
<b>Faculty</b>	Faculty of Informatics, Burapha University
<b>Academic Year</b>	2022

## ABSTRACT

Many educational institutions offer courses on the fundamentals of computer programming. The teaching and learning include both theoretical and practical training by writing programs to solve the problem. In the process of doing this problem, it was discovered that some groups of students might copy the source code. This causes problems that make students unable to program and affects students themselves. Therefore, most teachers tend to supervise, monitor and may use tools to verify source code copying. These tools are not very comfortable to use because often data displayed in tabular form. Therefore, the SEAPlag was developed for the convenience of teachers in academic who wants to check the similarity of the source code. The main system's module consists of import source code module, output graph module, export output module, scope output module, and compare source code module. The results of the system are in the form of correlation graphs and source code similarity percentage tables. Users can choose to view source code comparison data on similar sections. The tools developed in this system consist of Angular, Node.js, and Visual Studio Code.

## สารบัญ

หน้า

กิตติกรรมประกาศ.....	ก
บทคัดย่อ .....	ข
สารบัญ.....	ง
สารบัญรูปภาพ .....	ช
สารบัญตาราง.....	ญ
บทที่	
1 บทนำ	
1.1 ที่มาและความสำคัญของโครงการ .....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบเขตและรายละเอียดของโครงการ .....	3
1.4 เครื่องมือที่ใช้ในการพัฒนา .....	5
1.5 ขั้นตอนการดำเนินโครงการ .....	7
1.6 แผนการดำเนินโครงการ .....	8
1.7 ดัชนีวัดความสำเร็จของโครงการ .....	10
2 หลักการและทฤษฎีที่เกี่ยวข้อง	
2.1 นิยามคำศัพท์เฉพาะ .....	12
2.2 งานวิจัยหรือบทความที่เกี่ยวข้อง .....	13
2.3 เครื่องมือและเทคโนโลยีที่ใช้ในการดำเนินโครงการ .....	18
3 รายละเอียดของการดำเนินโครงการ	
3.1 การออกแบบการทำงานของระบบ .....	26
3.2 แผนภาพยูสเคส (Use Case Diagram).....	27

## สารบัญ

บทที่	หน้า
<b>3 รายละเอียดของการดำเนินโครงการ (ต่อ)</b>	
3.3 คำอธิบายสเคส (Use Case Description).....	28
3.4 แผนภาพกิจกรรม (Activity Diagram).....	34
3.5 แผนภาพคลาส (Class Diagram).....	37
3.6 แผนภาพลำดับการทำงาน (Sequence Diagram).....	41
3.7 การออกแบบหน้าจอแสดงผล .....	46
3.8 การทำงานของ JPlag .....	51
3.9 การออกแบบเว็บเซอร์วิส (Web Service).....	55
<b>4 ผลการดำเนินการ</b>	
4.1 มอดูลนำเข้าข้อมูลซอฟต์แวร์ (Import Source Code Module).....	57
4.2 มอดูลแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของซอฟต์แวร์ (Output Plagiarism Module).....	59
4.3 การพัฒนามอดูลกำหนดขอบเขตการแสดงผลลัพธ์ (Scope Output Module).....	61
4.4 การพัฒนามอดูลแสดงการเปรียบเทียบซอฟต์แวร์ในส่วนที่เหมือนกัน (Compare Source Code Module).....	62
4.5 ประสิทธิภาพการตรวจสอบความคล้ายคลึงของซอฟต์แวร์ .....	63
<b>5 สรุปผลการดำเนินงาน</b>	
5.1 สรุปผลการดำเนินโครงการ.....	64
5.2 ประโยชน์ที่ได้รับ .....	65
5.3 ข้อจำกัดของโครงการ .....	65
5.4 ปัญหาและอุปสรรค .....	66

## สารบัญ

บทที่	หน้า
<b>5 สรุปผลการดำเนินงาน (ต่อ)</b>	
5.5 ข้อเสนอแนะ.....	66
5.6 แนวโน้มหรือทิศทางการพัฒนาในอนาคต .....	66
บรรณานุกรม.....	67
ประวัติย่อของผู้ดำเนินโครงการ.....	69

## สารบัญรูปภาพ

ภาพที่	หน้า
2-1 การประยุกต์ใช้ Bootstrap Table .....	17
2-2 การประยุกต์ใช้ Force-directed Graph .....	17
2-3 สัญลักษณ์ของ TypeScript .....	19
2-4 สัญลักษณ์ของ HTML .....	19
2-5 สัญลักษณ์ของ CSS .....	20
2-6 สัญลักษณ์ของ JavaScript .....	21
2-7 สัญลักษณ์ของ Visual Studio Code .....	22
2-8 สัญลักษณ์ของ Google Chrome .....	23
2-9 สัญลักษณ์ของ GitLab .....	23
2-10 สัญลักษณ์ของ Angular .....	24
2-11 สัญลักษณ์ของ Node.js .....	25
3-1 ลำดับการทำงานของระบบ .....	26
3-2 รายละเอียดแผนภาพภูมิศาสตร์ .....	27
3-3 แผนภาพกิจกรรมนำเข้าข้อมูลชอร์สโค้ด .....	35
3-4 แผนภาพกิจกรรมแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด .....	36
3-5 แผนภาพกิจกรรมกำหนดขอบเขตการแสดงผลลัพธ์ .....	36
3-6 แผนภาพกิจกรรมแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน .....	37
3-7 แผนภาพคลาส .....	38
3-8 แผนภาพลำดับการทำงานนำเข้าข้อมูลชอร์สโค้ด .....	42
3-9 แผนภาพลำดับการทำงานแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด .....	43

## สารบัญภาพ

ภาพที่	หน้า
3-10 แผนภาพลำดับการทำน้ำหนาดของเบตการแสดงผลลัพธ์.....	44
3-11 แผนภาพลำดับการทำงานแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน .....	45
3-12 หน้าจอเข้าข้อมูลชอร์สโค้ด.....	46
3-13 หน้าจอผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบกราฟ .....	47
3-14 การแสดงร้อยละความคล้ายคลึงของชอร์สโค้ดในรูปแบบกราฟ .....	47
3-15 หน้าจอผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบตาราง .....	48
3-16 กล่องสำหรับกำหนดของเบตการแสดงผลลัพธ์ .....	48
3-17 หน้าจอการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน .....	49
3-18 ส่วนของการแสดงร้อยละความคล้ายคลึงของชอร์สโค้ด .....	49
3-19 ส่วนของการแสดงชุดข้อมูลชอร์สโค้ดที่เหมือนกันของผู้เป็นเจ้าของชอร์สโค้ด .....	50
3-20 ส่วนของการแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน .....	51
3-21 เมนู Releases ใน GitHub ของ JPlag .....	52
3-22 รายการเรอร์ชันต่างๆ ของ JPlag .....	52
3-23 ตัวเลือกภาษาโปรแกรมที่ JPlag รองรับ .....	53
4-1 หน้าจอมอถุลนำเข้าข้อมูลชอร์สโค้ด.....	57
4-2 หน้าต่างแจ้งเตือน กรณีที่ผู้ใช้งานนำเข้าไฟล์มากกว่า 1 ไฟล์.....	58
4-3 หน้าต่างแจ้งเตือน กรณีที่ผู้ใช้งานนำเข้าไฟล์ที่ไม่ใช่นามสกุล .zip .....	58
4-4 หน้าจอมอถุลนำเข้าข้อมูลชอร์สโค้ด เมื่อผู้ใช้งานนำเข้าไฟล์ข้อมูลชอร์สโค้ด .....	59
4-5 หน้าจอมอถุลแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบกราฟ 2 มิติ .....	60

## สารบัญภาพ

ภาพที่	หน้า
4-6 หน้าจอมอตูลแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบกราฟ 3 มิติ .....	60
4-7 หน้าจอมอตูลแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบตาราง .....	61
4-8 หน้าจอมอตูลกำหนดขอบเขตการแสดงผลลัพธ์ ภายในขอบเขตอยู่ละ 50 ถึง 100 .....	61
4-9 หน้าจอมอตูลกำหนดขอบเขตการแสดงผลลัพธ์ ภายในขอบเขตอยู่ละ 75 ถึง 100 .....	62
4-10 หน้าจอมอตูลแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน .....	62

## สารบัญตาราง

ตารางที่	หน้า
1-1 แผนการดำเนินโครงการ .....	8
2-1 นิยามคำศัพท์เฉพาะ .....	12
2-2 การเปรียบเทียบเครื่องมือที่ใช้ในการตรวจสอบความคล้ายคลึงของชอร์สโค้ด .....	15
3-1 นำเข้าข้อมูลชอร์สโค้ด .....	28
3-2 แสดงผลลัพธ์ของการตรวจสอบการคัดลอกชอร์สโค้ด .....	30
3-3 กำหนดขอบเขตของการแสดงผลลัพธ์ .....	31
3-4 แสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน .....	33
4-1 ประสิทธิภาพการตรวจสอบความคล้ายคลึงของชอร์สโค้ด .....	63

## บทที่ 1

### บทนำ

ในปัจจุบันเทคโนโลยีสารสนเทศได้เข้ามามีบทบาทสำคัญในการดำเนินชีวิตประจำวัน โดยเฉพาะในด้านของการศึกษา ซึ่งเป็นปัจจัยสำคัญที่ช่วยส่งเสริมให้นักศึกษาผู้เรียนความต้องการที่จะศึกษาค้นคว้าหาความรู้เพื่อใช้สำหรับการประกอบวิชาชีพในอนาคต สามารถเข้าถึงข้อมูลข่าวสารที่มีปริมาณมากและมีความหลากหลายของเนื้อหาสาระ ทำให้นักศึกษาสามารถเลือกสรรและวิเคราะห์ข้อมูลเหล่านั้นได้อย่างมีประสิทธิภาพมากยิ่งขึ้น ในขณะเดียวกันนี้ เทคโนโลยีได้มีการพัฒนาเพิ่มเติมความสามารถอย่างรวดเร็วและต่อเนื่อง ทำให้การศึกษาเกี่ยวกับเทคโนโลยีในปัจจุบัน เป็นสิ่งสำคัญที่จะช่วยให้นักศึกษาสามารถเรียนรู้เท่าทันการเปลี่ยนแปลงของเทคโนโลยี และเพื่อที่จะได้เลือกใช้เทคโนโลยีที่เหมาะสมในการสร้างสรรค์เทคโนโลยีใหม่ๆ ที่ตอบโจทย์ความต้องการของผู้คนในสังคมอันไร้ที่สิ้นสุด ทั้งนี้ รายวิชาโครงงานวิศวกรรมซอฟต์แวร์ (Software Engineering Project) จึงถูกจัดตั้งขึ้นเพื่อให้นักศึกษาในรายวิชา ได้นำเอาความรู้ทั้งทางภาคทฤษฎีและทางภาคปฏิบัติจากการศึกษาประจำรายวิชาต่างๆ รวมถึงหัวข้อที่นักศึกษาสนใจมาจัดทำโครงงาน เพื่อให้นักศึกษาได้บูรณาการความรู้ที่ได้เรียนมาตลอดระยะเวลาการศึกษา มาใช้ในการพัฒนาซอฟต์แวร์ที่ตอบโจทย์ความต้องการของกลุ่มเป้าหมาย โดยเนื้อหาสาระในบทนี้ ครอบคลุมในเรื่องของข้อมูลที่เกี่ยวข้องกับการทำเนินโครงงานของผู้ดำเนินโครงงาน ประกอบไปด้วย ที่มาและความสำคัญของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตและรายละเอียดของโครงงาน เครื่องมือที่ใช้ในการพัฒนา ขั้นตอนการทำเนินโครงงาน แผนการทำเนินโครงงาน และตัวชี้วัดความสำเร็จของโครงงาน

#### 1.1 ที่มาและความสำคัญของโครงงาน

ความก้าวหน้าทางด้านเทคโนโลยีและสื่อโซเชียลมีเดีย นำมาซึ่งความสะดวกในการดำเนินชีวิตในด้านต่างๆ ไม่ว่าจะเป็นในด้านของเศรษฐกิจ การคุณภาพ โทรคมนาคม หรือกระทิ่งในด้านของการศึกษา ซึ่งเป็นสิ่งสำคัญที่ช่วยผลักดันให้นักศึกษาผู้เรียนหาความรู้ สามารถเข้าถึงข้อมูลข่าวสารที่มีปริมาณมหาศาลและมีความหลากหลายของโครงสร้างเนื้อหาได้อย่างไม่มีที่สิ้นสุด แต่ในขณะเดียวกัน ความก้าวหน้าก็มาพร้อมกับช่องทางที่ช่วยให้นักศึกษาบางกลุ่ม สามารถกระทำ

การอะไรบางอย่างอันเป็นการเอาเปรียบหรือละเมิดผลงานของผู้อื่น ซึ่งในที่นี้กล่าวถึงการคัดลอกชอร์สโค้ดภายในสถาบันการศึกษา ซึ่งเป็นปัญหาที่สำคัญ เนื่องจากถ้านักศึกษาทำการคัดลอกชอร์สโค้ดกัน ซึ่งเป็นต้นเหตุของการสร้างลักษณะนิสัยในการคัดลอกผลงานของผู้อื่นทั้งโดยได้รับอนุญาตหรือไม่ได้รับอนุญาตจากผู้เป็นเจ้าของ นอกจานั้น ปัญหาสำคัญอีกประการหนึ่งคือ หากนักศึกษาทำการคัดลอกชอร์สโค้ดกันแล้ว นักศึกษาเหล่านั้นก็ไม่เกิดการเรียนรู้ ซึ่งในปัจจุบัน มีเครื่องมือสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ด แต่เครื่องมือที่มีอยู่ในปัจจุบันนั้น อาจมีค่าใช้จ่ายในการใช้งาน หรือซอฟต์แวร์ที่เป็นโอลิเวนชอร์สที่สามารถใช้งานได้ฟรี ก็ยังมีการใช้งานที่ค่อนข้างยุ่งยากและผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ดนั้น มีรายละเอียดของผลลัพธ์ที่ผู้ใช้งานต้องใช้เวลาในการทำความเข้าใจ

ผู้ดำเนินโครงการได้มองเห็นถึงปัญหาของการคัดลอกชอร์สโค้ดภายในสถาบันการศึกษา ตามที่ได้กล่าวไปในข้างต้น ผู้ดำเนินโครงการจึงได้ทำการศึกษาข้อมูลเกี่ยวกับเครื่องมือที่ใช้ในการตรวจสอบการคัดลอกชอร์สโค้ดที่มีความแม่นยำและเที่ยงตรง รวมถึงค้นหาวิธีการหรือรูปแบบการแสดงผลลัพธ์ของการตรวจสอบการคัดลอกชอร์สโค้ดที่เหมาะสม เพื่อใช้ในการพัฒนาระบบตรวจสอบและแสดงผลลัพธ์ของการคัดลอกชอร์สโค้ด โดยมีความคาดหวังว่าระบบที่ผู้ดำเนินโครงการได้ทำการพัฒนาขึ้นนี้ จะช่วยให้ผู้ใช้งานสามารถตรวจสอบการคัดลอกชอร์สโค้ดได้อย่างสะดวก และมีประสิทธิภาพมากยิ่งขึ้น

## 1.2 วัตถุประสงค์ของโครงการ

ผู้ดำเนินโครงการได้ทำการศึกษาถึงปัญหาของการคัดลอกชอร์สโค้ดภายในสถาบันการศึกษา และค้นคว้าหาวิธีการที่เหมาะสมที่ใช้ในการตรวจสอบการคัดลอกชอร์สโค้ดให้เป็นไปอย่างสะดวก และมีประสิทธิภาพ โดยผู้ดำเนินโครงการได้กำหนดวัตถุประสงค์ของการจัดทำโครงการ ดังนี้

- 1) เพื่อพัฒนาระบบสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ดให้สามารถทำงานได้อย่างมีประสิทธิภาพ
- 2) เพื่ออำนวยความสะดวกให้กับผู้ใช้งานที่ต้องการใช้เครื่องมือที่เหมาะสมสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ด
- 3) เพื่อแสดงให้เห็นถึงความเชื่อมโยงและความสัมพันธ์ของผลลัพธ์จากการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

### 1.3 ขอบเขตและรายละเอียดของโครงการ

การดำเนินโครงการ ได้มีการกำหนดขอบเขตสำหรับการพัฒนาระบบแสดงผลลัพธ์ การตรวจสอบความคล้ายคลึงของชอร์สโค้ด เพื่อให้เห็นถึงประเด็นหรือองค์ประกอบที่สำคัญ ในการพัฒนาระบบ โดยมีรายละเอียด ดังนี้

#### 1.3.1 การพัฒนามODULEนำเข้าข้อมูลชอร์สโค้ด (Import Source Code Module)

มодูลสำหรับนำเข้าข้อมูลชอร์สโค้ด โดยผู้ใช้งานทำการเตรียมไฟล์เดอร์หรือไดร์เควทอร์ ซึ่งภายในประกอบไปด้วยไฟล์เดอร์หรือไดร์เควทอร์ที่เป็นข้อมูลของชอร์สโค้ดที่ผู้ใช้งานต้องการ ตรวจสอบการคัดลอกชอร์สโค้ด ผู้ใช้งานต้องเลือกภาษาและนำไฟล์เดอร์ที่จัดเตรียมนี้ป้อนเข้าสู่ระบบ ระบบจะทำการตรวจสอบการคัดลอกชอร์สโค้ดที่อยู่ภายในไฟล์ และทำการบันทึกผลลัพธ์ ของการตรวจสอบการคัดลอกชอร์สโค้ดเอาไว้ โดยรายละเอียดการทำงานในส่วนบริการของระบบ โดยมีรายละเอียด ดังนี้

- 1) บันทึกไฟล์ข้อมูลชอร์สโค้ดนามสกุล .zip ของผู้ใช้งาน
- 2) แตกไฟล์ข้อมูลชอร์สโค้ดของผู้ใช้งาน จาก .zip เป็นไฟล์เดอร์หรือไดร์เควทอร์
- 3) บันทึกไฟล์เดอร์หรือไดร์เควทอร์ข้อมูลชอร์สโค้ดของผู้ใช้งาน
- 4) สั่งการให้ไลบรารี JPlag ประมวลผลข้อมูลชอร์สโค้ดภายในไฟล์เดอร์หรือไดร์เควทอร์
- 5) บันทึกไฟล์ข้อมูลผลลัพธ์จากการประมวลผลของไลบรารี JPlag นามสกุล .zip
- 6) แตกไฟล์ข้อมูลผลลัพธ์จากการประมวลผลของไลบรารี JPlag จากนามสกุล .zip เป็นไฟล์เดอร์หรือไดร์เควทอร์ ซึ่งภายในเป็นไฟล์ข้อมูลผลลัพธ์นามสกุล .json
- 7) บันทึกไฟล์เดอร์หรือไดร์เควทอร์ข้อมูลผลลัพธ์จากการประมวลผลของไลบรารี JPlag ซึ่งภายในเป็นไฟล์ข้อมูลผลลัพธ์นามสกุล .json

#### 1.3.2 การพัฒนามODULEแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด (Output Plagiarism Module)

มอดูลแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดที่ระบบได้ทำการประมวลผล และบันทึกผลลัพธ์เอาไว้ โดยการแสดงผลลัพธ์จะอยู่ในรูปแบบของกราฟ (Force-directed Graph) โดยใช้ Node แทนชื่อไฟล์เดอร์หรือไดร์เควทอร์ที่อยู่และแสดงความเชื่อมโยงระหว่าง Node โดยใช้เส้นความสัมพันธ์ซึ่งมีข้อความแสดงร้อยละความเหมือนกันของชอร์สโค้ด (Source Code Similarity) และการแสดงผลในรูปแบบตาราง (Table) ซึ่งเป็นรายละเอียดผลลัพธ์จากการ

ตรวจสอบการคัดลอกชอร์สโค้ด ซึ่งแสดงผลในรูปแบบของตารางความสัมพันธ์ระหว่างผู้เป็นเจ้าของชอร์สโค้ด โดยรายละเอียดการทำงานในส่วนบริการของระบบ ซึ่งมีรายละเอียด ดังนี้

1) ส่งออกข้อมูลผลลัพธ์จากการประมวลผลของไลบรารี JPlag ชื่อไฟล์ overview.json

2) ระบบอ่านไฟล์ overview.json

3) ระบบจัดเตรียมข้อมูลจากไฟล์ overview.json เพื่อให้สามารถใช้งานร่วมกับไลบรารีแสดงผลข้อมูลในรูปแบบกราฟ

4) ระบบแสดงข้อมูลที่อ่านได้จากไฟล์ overview.json ในรูปแบบกราฟหรือตาราง

### 1.3.3 การพัฒนามอดูลกำหนดขอบเขตการแสดงผลลัพธ์ (Scope Output Module)

มอดูลจัดการและกำหนดขอบเขตในการแสดงผลลัพธ์ของการตรวจสอบการคัดลอกชอร์สโค้ดในรูปแบบของกราฟ โดยผู้ใช้งานสามารถเลือกขอบเขตตัวอย่างความเหมือนกันของชอร์สโค้ดเพื่อกำหนดขอบเขตการแสดงผลลัพธ์ของกราฟ โดยมีรายละเอียด ดังนี้

1) ผู้ใช้งานสามารถกรอกขอบเขตของร้อยละความคล้ายคลึงของชอร์สโค้ด (Range input)

2) ผู้ใช้งานสามารถกดปุ่ม (Button) “Confirm” เพื่อทำการยืนยันขอบเขตการแสดงผลลัพธ์

3) ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบกราฟ (Graph output) ภายใต้ขอบเขตที่ผู้ใช้งานกรอกขอบเขตที่ต้องการและทำการยืนยันขอบเขตนั้น

4) ผู้ใช้งานสามารถกดปุ่ม (Button) “Reset” เพื่อทำการล้างขอบเขตการแสดงผลลัพธ์

5) ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบกราฟ (Graph output) โดยแสดงข้อมูลผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดทั้งหมด

### 1.3.4 การพัฒนามอดูลแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน (Compare Source Code Module)

มอดูลสำหรับแสดงผลข้อมูลเชิงลึกของการคัดลอกชอร์สโค้ด โดยผู้ใช้งานทำการเลือกที่ข้อความแสดงร้อยละความเหมือนกันของชอร์สโค้ดที่ต้องการ ระบบจะแสดงหน้าจอข้อมูลของเส้นความสัมพันธ์นั้น ซึ่งประกอบไปด้วย ร้อยละความเหมือนกันของชอร์สโค้ด (Source Code Similarity) และข้อมูลชอร์สโค้ดในส่วนที่ตรวจพบว่าเหมือนกัน โดยรายละเอียดการทำงานในส่วนบริการของระบบ ซึ่งมีรายละเอียด ดังนี้

1) ส่งออกข้อมูลผลลัพธ์จากการประมวลผลของไลบรารี JPlag นามสกุล .json ในส่วนของข้อมูลการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน

2) ระบบอ่านไฟล์ข้อมูลนามสกุล .json ซึ่งเป็นข้อมูลการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน

3) ระบบอ่านไฟล์ชอร์สโค้ดจำนวน 2 ไฟล์ ซึ่งทั้ง 2 ไฟล์ชอร์สโค้ดนั้นๆ มีส่วนของชอร์สโค้ดที่ตรวจสอบแล้วพบว่าเหมือนกัน

## 1.4 เครื่องมือที่ใช้ในการพัฒนา

เครื่องมือที่ใช้ในการพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดประกอบไปด้วยภาษาที่ใช้ในการแสดงผลผ่านทางหน้าจอ และการประมวลผลเพื่อดำเนินการในส่วนต่างๆ ของระบบ โดยมีรายละเอียดของภาษาที่ใช้ในการพัฒนาระบบ ดังนี้

### 1.4.1 ภาษาที่ใช้ในการพัฒนา

ส่วนนี้เป็นส่วนที่กล่าวถึงภาษาที่ใช้ในการพัฒนาระบบ ซึ่งประกอบไปด้วยภาษาที่ใช้ในการควบคุมการทำงานขององค์ประกอบต่างๆ ของหน้าจอแสดงผล และภาษาที่ใช้ในการควบคุมลำดับการทำงานในแต่ละส่วนต่างๆ ของระบบ โดยมีรายละเอียด ดังนี้

#### 1) TypeScript

ภาษาที่ถูกพัฒนามาจากภาษา JavaScript โดยเพิ่มความสามารถในการตรวจสอบข้อผิดพลาด (Error) ที่เกิดขึ้นจากการทำงานของโปรแกรมที่ถูกพัฒนาขึ้นจากภาษา JavaScript โดยในการประมวลผลโปรแกรมจะแจ้งข้อผิดพลาดที่เกิดขึ้นและตรวจพบให้ผู้พัฒนาได้ทราบ และทำการแก้ไขก่อนจะนำระบบที่พัฒนาไปใช้งานจริง ช่วยให้ผู้พัฒนาสามารถควบคุมและทบทวนการทำงานของโปรแกรมได้อย่างมีประสิทธิภาพ

#### 2) HTML (HyperText Markup Language)

ภาษาที่ถูกสร้างขึ้นเพื่อใช้ในการจัดวางโครงสร้างต่างๆ บนหน้าจอเว็บเบราว์เซอร์ โดยมีคุณสมบัติพิเศษที่สามารถเชื่อมโยงกับหน้าเว็บอื่นๆ ที่มีโครงสร้างของ HTML โดยใช้ Tag ในการควบคุมการแสดงผล และจัดวางโครงสร้างของกล่องข้อความ ปุ่มกด รูปภาพ วิดีโอ หรือองค์ประกอบส่วนต่างๆ บนหน้าจอแสดงผล

### 3) CSS (Cascading Style Sheets)

ภาษาที่ใช้ในการจัดรูปแบบการแสดงผลอย่างการตกแต่ง การลงสี การกำหนดลักษณะของตัวหนังสือที่ใช้ในการแสดงผล มักใช้งานร่วมกับภาษา HTML เพื่อให้ง่ายต่อการจัดรูปแบบการแสดงผลและใช้ในการควบคุมความสม่ำเสมอขององค์ประกอบต่างๆ บนหน้าจอแสดงผล

### 4) JavaScript

ภาษาที่ใช้สำหรับการเขียนโปรแกรมหรือพัฒนาแอปพลิเคชันต่างๆ ที่ได้รับความนิยมมาอย่างยาวนาน โดยมักใช้ในการเพิ่มความสามารถในการแสดงผล การเคลื่อนไหวขององค์ประกอบ การปฏิสัมพันธ์ และการกำหนดการทำงานขององค์ประกอบบนเว็บเบราว์เซอร์เมื่อเกิดเหตุการณ์ต่างๆ ขึ้นบนหน้าจอแสดงผล

#### 1.4.2 ซอฟต์แวร์ที่ใช้ในการพัฒนา

ซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ เป็นเครื่องมือที่อำนวยความสะดวกต่อผู้ดำเนินโครงการในการพัฒนาระบบ รวมถึงการควบคุมเวอร์ชันของซอฟต์แวร์สโตร์ และการแสดงผลบนเว็บเบราว์เซอร์ โดยมีรายละเอียด ดังนี้

##### 1) Visual Studio Code

เครื่องมือที่ถูกพัฒนาขึ้นสำหรับใช้ในการเขียนโปรแกรมภาษาต่างๆ ซึ่งมาพร้อมกับเครื่องมือเสริมที่ช่วยให้การปรับแก้ซอฟต์แวร์สโตร์มีประสิทธิภาพมากยิ่งขึ้น ไม่ว่าจะเป็นการตกแต่งไวยากรณ์ของภาษาให้ง่ายต่อการเขียนโปรแกรม และยังครอบคลุมถึงการตรวจสอบข้อผิดพลาดที่เกิดขึ้นระหว่างการเขียนโปรแกรม นอกจากนี้ยังสามารถเชื่อมต่อกับซอฟต์แวร์ที่ใช้ควบคุมเวอร์ชันของซอฟต์แวร์สโตร์ได้อีกด้วย

##### 2) Google Chrome

เบราว์เซอร์ที่ได้รับความนิยมสำหรับใช้ในการแสดงผลแอปพลิเคชันผ่านเครือข่ายอินเทอร์เน็ตที่มีความปลอดภัยในการเข้าใช้งานแอปพลิเคชันต่างๆ และมาพร้อมส่วนเสริมต่างๆ ที่จะช่วยอำนวยความสะดวกสบายให้กับผู้ใช้งาน ทั้งในระดับของผู้ใช้งานทั่วไปและผู้ใช้งานที่เป็นผู้พัฒนาแอปพลิเคชัน

##### 3) GitLab

ซอฟต์แวร์ที่ใช้สำหรับควบคุมเวอร์ชันของซอฟต์แวร์สโตร์ที่มีประสิทธิภาพและได้รับการยอมรับจากผู้ใช้งานทั่วโลก สำหรับใช้ในการควบคุมและติดตามเวอร์ชันของซอฟต์แวร์สโตร์

โดยมีความสามารถในการจัดเก็บข้อมูลสโคด และการดึงเอาข้อมูลสโคดเวอร์ชันก่อนหน้าเวอร์ชันล่าสุด กลับมาใช้งานได้ ทำให้ผู้พัฒนาสามารถจัดการกับเวอร์ชันของซอฟต์แวร์สโคดได้อย่างมีประสิทธิภาพ

#### 1.4.3 เครื่องแม่ข่ายที่ใช้ในการพัฒนา

ระบบที่ผู้ดำเนินโครงการได้พัฒนาขึ้นนี้ได้รับความร่วมมือจากคณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา ให้สามารถใช้งานเครื่องแม่ข่าย (Server) ของคณะ อันเป็นเครื่องมือสำคัญ ที่ช่วยส่งเสริมให้การพัฒนาระบบเป็นไปอย่างสะดวกและมีประสิทธิภาพ

#### 1.4.4 เฟรมเวิร์กที่ใช้ในการพัฒนา (Framework)

ส่วนนี้เป็นส่วนที่กล่าวถึงเฟรมเวิร์กที่ใช้ในการพัฒนาระบบ ซึ่งเป็นเครื่องมือที่ช่วยในการจัดการระบบที่กำลังพัฒนาอย่างมีโครงสร้าง ช่วยให้ผู้ดำเนินโครงการสามารถจัดวางโครงสร้างขององค์ประกอบต่างๆ บนหน้าจอแสดงผลและจัดการกับส่วนของการดำเนินการกับข้อมูลในระบบ โดยในการพัฒนาระบบ โดยมีรายละเอียด ดังนี้

##### 1) Angular

เฟรมเวิร์กทางฝั่งที่ปฏิสัมพันธ์กับผู้ใช้งานโดยตรง (Front-End Framework) ถูกพัฒนาขึ้น โดยบริษัท Google สำหรับใช้ในการพัฒนาเว็บแอปพลิเคชันโดยใช้ภาษา TypeScript ในการพัฒนา และสามารถเชื่อมต่อกับฐานข้อมูลเพื่อใช้ในการติดต่อร้องขอข้อมูลระหว่างแอปพลิเคชันกับฐานข้อมูล

##### 2) Node.js

เฟรมเวิร์กสำหรับพัฒนาแอปพลิเคชันหรือเว็บเซิร์ฟเวอร์ได้ ด้วยภาษา JavaScript เหมาะสมสำหรับการสร้างแอปพลิเคชันที่ต้องการใช้ข้อมูลจำนวนมาก และนิยมใช้ในการพัฒนาแอปพลิเคชันที่ใช้ข้อมูลแบบ Realtime สามารถทำงานได้ทุกระบบปฏิบัติการ

### 1.5 ขั้นตอนการดำเนินโครงการ

ผู้ดำเนินโครงการได้ทำการกำหนดขั้นตอนในการดำเนินโครงการ สำหรับใช้เป็นแนวทางในการวางแผนการพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอฟต์แวร์สโคด โดยมีรายละเอียด ดังนี้

##### 1) วางแผนการดำเนินโครงการ

##### 2) ออกแบบหน้าจอแสดงผลและการทำงานของระบบ

##### 3) พัฒนาระบบ

4) ทดสอบการทำงานของระบบ

5) ส่งมอบและติดตั้งระบบ

### 1.6 แผนการดำเนินโครงการ

ในการดำเนินโครงการ ผู้ดำเนินโครงการได้ทำการวางแผนการสำหรับพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด เพื่อใช้เป็นแนวทางในการจัดสรรเวลาและทรัพยากรที่เหมาะสมสำหรับดำเนินโครงการ โดยมีรายละเอียด ดังนี้

ตารางที่ 1-1 แผนการดำเนินโครงการ

หัวข้องาน	วันที่คาดว่าจะเริ่ม	วันที่คาดว่าจะเสร็จ	วันที่เริ่ม	วันที่เสร็จ
<b>ตั้งแต่วันที่ 19 พฤศจิกายน พ.ศ. 2565 ถึงวันที่ 30 พฤศจิกายน พ.ศ. 2565</b>				
<b>สัปดาห์ที่ 1</b>				
1. ศึกษาไลบรารี JPlag	19 พ.ย. 65	22 พ.ย. 65	19 พ.ย. 65	22 พ.ย. 65
2. ทดลองใช้งาน JPlag	23 พ.ย. 65	26 พ.ย. 65	23 พ.ย. 65	26 พ.ย. 65
<b>สัปดาห์ที่ 2</b>				
1. ศึกษาการทำงานของ JPlag	27 พ.ย. 65	30 พ.ย. 65	27 พ.ย. 65	30 พ.ย. 65
2. ทดลองใช้งาน JPlag	27 พ.ย. 65	30 พ.ย. 65	27 พ.ย. 65	30 พ.ย. 65
<b>ตั้งแต่วันที่ 1 ธันวาคม พ.ศ. 2565 ถึงวันที่ 31 ธันวาคม พ.ศ. 2565</b>				
<b>สัปดาห์ที่ 1</b>				
1. ออกรายบุคคลความสามารถ และการทำงานของระบบ	1 ธ.ค. 65	3 ธ.ค. 65	1 ธ.ค. 65	3 ธ.ค. 65
2. ศึกษาไลบราราราฟ	1 ธ.ค. 65	3 ธ.ค. 65	1 ธ.ค. 65	3 ธ.ค. 65
<b>สัปดาห์ที่ 2</b>				
1. ทดลองใช้งานไลบราราราฟ	4 ธ.ค. 65	5 ธ.ค. 65	4 ธ.ค. 65	5 ธ.ค. 65
2. กำหนดขอบเขตของระบบ	6 ธ.ค. 65	10 ธ.ค. 65	6 ธ.ค. 65	10 ธ.ค. 65

ตารางที่ 1-1 แผนการดำเนินโครงการ (2)

หัวข้องาน	วันที่คาด ว่าจะเริ่ม	วันที่คาด ว่าจะเสร็จ	วันที่เริ่ม	วันที่เสร็จ
<b>ตั้งแต่วันที่ 1 ธันวาคม พ.ศ. 2565 ถึงวันที่ 31 ธันวาคม พ.ศ. 2565 (ต่อ)</b>				
<b>สัปดาห์ที่ 3</b>				
1. ออกรูปแบบแพนกวานิช Use Case	11 ธ.ค. 65	13 ธ.ค. 65	11 ธ.ค. 65	13 ธ.ค. 65
2. ออกรูปแบบหน้าจอของระบบ	14 ธ.ค. 65	17 ธ.ค. 65	14 ธ.ค. 65	17 ธ.ค. 65
<b>สัปดาห์ที่ 4</b>				
1. จัดทำคำอธิบายสเปคสเปค	18 ธ.ค. 65	20 ธ.ค. 65	18 ธ.ค. 65	20 ธ.ค. 65
2. ออกรูปแบบแพนกวานิกิจกรรม	18 ธ.ค. 65	20 ธ.ค. 65	18 ธ.ค. 65	20 ธ.ค. 65
3. ออกรูปแบบแพนกิจกรรมคลาส	21 ธ.ค. 65	24 ธ.ค. 65	21 ธ.ค. 65	24 ธ.ค. 65
<b>สัปดาห์ที่ 5</b>				
1. ออกรูปแบบแพนกิจกรรมสำหรับ การทำงาน	25 ธ.ค. 65	27 ธ.ค. 65	25 ธ.ค. 65	27 ธ.ค. 65
2. ติดตั้ง Angular และไลบรารี	28 ธ.ค. 65	31 ธ.ค. 65	28 ธ.ค. 65	31 ธ.ค. 65
<b>ตั้งแต่วันที่ 1 มกราคม พ.ศ. 2566 ถึงวันที่ 31 มกราคม พ.ศ. 2566</b>				
<b>สัปดาห์ที่ 1</b>				
1. พัฒนามอดูล Import Source Code	1 ม.ค. 66	7 ม.ค. 66	1 ม.ค. 66	7 ม.ค. 66
2. ทดสอบการทำงานของมอดูล Import Source Code	1 ม.ค. 66	7 ม.ค. 66	1 ม.ค. 66	7 ม.ค. 66
<b>สัปดาห์ที่ 2</b>				
1. พัฒนามอดูล Output Graph	8 ม.ค. 65	14 ม.ค. 65	8 ม.ค. 65	14 ม.ค. 65
2. ทดสอบการทำงานของมอดูล Output Graph	8 ม.ค. 65	14 ม.ค. 65	8 ม.ค. 65	14 ม.ค. 65

### ตารางที่ 1-1 แผนการดำเนินโครงการ (3)

หัวข้องาน	วันที่คาด ว่าจะเริ่ม	วันที่คาด ว่าจะเสร็จ	วันที่เริ่ม	วันที่เสร็จ
<b>ตั้งแต่วันที่ 1 มกราคม พ.ศ. 2566 ถึงวันที่ 31 มกราคม พ.ศ. 2566 (ต่อ)</b>				
<b>สัปดาห์ที่ 3</b>				
1. พัฒนา module Scope Output	15 ม.ค. 65	21 ม.ค. 65	15 ม.ค. 65	21 ม.ค. 65
2. ทดสอบการทำงานของ module Scope Output	15 ม.ค. 65	21 ม.ค. 65	15 ม.ค. 65	21 ม.ค. 65
<b>สัปดาห์ที่ 4</b>				
1. พัฒนามอดูล Compare Source Code	22 ม.ค. 65	28 ม.ค. 65	22 ม.ค. 65	28 ม.ค. 65
2. ทดสอบมอดูลการทำงานของมอดูล Compare Source Code	22 ม.ค. 65	28 ม.ค. 65	22 ม.ค. 65	28 ม.ค. 65
<b>สัปดาห์ที่ 5</b>				
1. ทบทวนการทำงานของระบบ	29 ม.ค. 65	31 ม.ค. 65	29 ม.ค. 65	31 ม.ค. 65
2. ส่งมอบระบบ	29 ม.ค. 65	31 ม.ค. 65	29 ม.ค. 65	31 ม.ค. 65

## 1.7 ดัชนีวัดความสำเร็จของโครงการ

ผู้ดำเนินโครงการได้กำหนดตัวชี้วัดสำหรับวัดความสำเร็จของโครงการ เพื่อใช้เป็นแบบประเมินสำหรับประเมินความสามารถในด้านต่างๆ ใน การดำเนินโครงการ และวัดปริมาณความคืบหน้าของการดำเนินโครงการ โดยมีรายละเอียด ดังนี้

### 1.7.1 ความสามารถในการออกแบบการทำงานของระบบ

หลังจากได้ศึกษาเครื่องมือที่จะใช้ในการพัฒนาระบบแล้ว ผู้ดำเนินโครงการได้ทำการออกแบบโครงสร้างและลำดับการทำงานของระบบ รวมถึงศึกษาการทำงานของระบบใกล้เคียง เพื่อใช้เป็นแนวในการออกแบบระบบ โดยจะประเมินผู้ดำเนินโครงการได้ ดังนี้

1) ประเมินจากผู้ดำเนินโครงการ สามารถออกแบบโครงสร้างและลำดับการทำงานของระบบได้ตรงกับแผนดำเนินโครงการที่ได้วางเอาไว้

2) ประเมินจากผู้ดำเนินโครงการ สามารถออกแบบโครงสร้างและลำดับการทำงานของระบบได้ตรงกับขอบเขตของระบบที่ได้ระบุเอาไว้

3) ประเมินจากผู้ดำเนินโครงการ สามารถออกแบบโครงสร้างและลำดับการทำงานของระบบโดยมีรายละเอียดที่ครบถ้วนและเพียงพอสำหรับใช้ในการพัฒนาระบบ

### 1.7.2 ความสามารถในการพัฒนาซอฟต์แวร์

การพัฒนาซอฟต์แวร์ของดำเนินโครงการนั้น เริ่มตั้งแต่ศึกษาเครื่องมือที่จะใช้ในการพัฒนาระบบ ออกแบบลำดับการทำงานของระบบ การออกแบบหน้าจอแสดงผลของระบบ การเขียนโปรแกรมเพื่อพัฒนาระบบ การทดสอบระบบ และการส่งมอบระบบ ซึ่งคำนึงถึงการพัฒนาซอฟต์แวร์ที่อำนวยความสะดวกต่อผู้ใช้งานเป็นสิ่งสำคัญ โดยจะประเมินผู้ดำเนินโครงการได้ดังนี้

1) ประเมินจากผู้ดำเนินโครงการ สามารถพัฒนาระบบได้ตรงกับที่ออกแบบเอาไว้

2) ประเมินจากผู้ดำเนินโครงการ สามารถพัฒนาระบบได้ตรงกับขอบเขตของระบบ และครอบคลุมรายละเอียดการทำงานของระบบอย่างครบถ้วนสมบูรณ์

### 1.7.3 ความมีวินัยในการรายงานความคืบหน้าของการดำเนินโครงการ

ในการดำเนินโครงการ ผู้ดำเนินโครงการได้มีการรายงานความคืบหน้าและปัญหาที่พบระหว่างดำเนินโครงการกับอาจารย์ที่ปรึกษา ซึ่งมีการนัดหมายวันที่ เวลา และสถานที่สำหรับพูดคุยกับอาจารย์ที่ปรึกษา โดยจะประเมินผู้ดำเนินโครงการได้ดังนี้

1) ประเมินจากผู้ดำเนินโครงการ สามารถรายงานความคืบหน้าของการดำเนินโครงการซึ่งมีรายละเอียดที่ครบถ้วนและสมบูรณ์

2) ประเมินจากผู้ดำเนินโครงการ สามารถอธิบายรายละเอียดของปัญหาที่พบระหว่างดำเนินโครงการได้อย่างชัดเจนและมีรายละเอียดที่สมบูรณ์

3) ประเมินจากผู้ดำเนินโครงการ สามารถเตรียมการรายงานความคืบหน้าของโครงการได้ตรงกับเวลาและสถานที่นัดหมายสำหรับรายงานความคืบหน้า

## บทที่ 2

### หลักการและทฤษฎีที่เกี่ยวข้อง

ผู้ดำเนินโครงการได้ทำการวิเคราะห์ปัญหาของการคัดลอกซอฟต์แวร์ในสถาบันการศึกษาแล้วจึงทำการการศึกษาค้นคว้าหาข้อมูลและเครื่องมือที่เหมาะสมสำหรับใช้ในการพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอฟต์แวร์โดยเริ่มศึกษาจากพฤติกรรมการคัดลอกซอฟต์แวร์ของนักศึกษาและรูปแบบของการคัดลอกซอฟต์แวร์ จนนั้นจึงทำการศึกษาการทำงานของเครื่องมือที่จะใช้ในการตรวจสอบความคล้ายคลึงของซอฟต์แวร์ โดยเนื้อหาสาระในบทนี้ครอบคลุมในเรื่องขององค์ความรู้และเครื่องมือที่ใช้ในการพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอฟต์แวร์ ประกอบไปด้วย นิยามคำศัพท์เฉพาะ งานวิจัยหรือบทความที่เกี่ยวข้องรวมถึงเครื่องมือและเทคโนโลยีที่ใช้ในการดำเนินโครงการ

#### 2.1 นิยามคำศัพท์เฉพาะ

ในการศึกษาค้นคว้าหาความรู้เพื่อใช้ในการพัฒนาระบบแสดงผลลัพธ์การตรวจสอบคล้ายคลึงของซอฟต์แวร์ ผู้ดำเนินโครงการได้ทำการระบุคำศัพท์ที่เกี่ยวข้องเพื่อใช้อธิบายความหมายของคำศัพท์ให้มีความเข้าใจตรงกัน ณ ที่นี่ โดยมีรายละเอียด ดังต่อไปนี้

ตารางที่ 2-1 นิยามคำศัพท์เฉพาะ

ลำดับ	คำศัพท์	คำอธิบาย
1.	ผู้ใช้งาน (User)	บุคคลที่มีปฏิกรรมโนําตัวตอบกับระบบ ดำเนินการกับระบบเพื่อให้บรรลุความต้องการของตน
2.	การคัดลอก (Plagiarism)	การกระทำที่เป็นการแอบอ้างผลงานดังเดิมของผู้อื่น ทั้งหมดหรือนำบางส่วน มาใส่หรือมาใช้ในผลงานของตนเอง
3.	ซอฟต์แวร์ (Source Code)	แฟ้มข้อมูลที่เป็นตัวต้นฉบับของโปรแกรม ซึ่งภายในประกอบไปด้วยภาษาโปรแกรม ที่เขียนขึ้นสำหรับสั่งการคอมพิวเตอร์ให้ทำงานตามที่ผู้เขียนต้องการ

ตารางที่ 2-1 นิยามคำศัพท์เฉพาะ (ต่อ)

ลำดับ	คำศัพท์	คำอธิบาย
4.	กราฟ (Graph)	แผนภูมิที่ประกอบไปด้วยเส้น จุด หรือภาพ เพื่อแสดงความสัมพันธ์ของสิ่งที่ผู้สร้างให้ความสนใจ
5.	การเปรียบเทียบ (Comparison)	การนำสิ่งที่สนใจตั้งแต่ 2 ชิ้นขึ้นไป มาเปรียบเทียบ เพื่อสังเกตองค์ประกอบที่เหมือนกันหรือแตกต่างกัน
6.	ชอร์สโค้ด (Source Code)	แฟ้มข้อมูลที่เป็นตัวต้นฉบับของโปรแกรม ซึ่งภายในประกอบไปด้วยภาษาโปรแกรม ที่เขียนขึ้นสำหรับ สั่งการคอมพิวเตอร์ให้ทำงานตามที่ผู้เขียนต้องการ

## 2.2 งานวิจัยหรือบทความที่เกี่ยวข้อง

ในการดำเนินโครงการ ผู้ดำเนินโครงการได้มีการศึกษาหาความรู้เกี่ยวกับเครื่องมือ และเทคนิคต่างๆ ที่ต้องใช้ในการพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึง ของชอร์สโค้ด รวมถึงการศึกษาในหัวข้อของเครื่องมือที่ใช้ในการตรวจสอบการคัดลอกชอร์สโค้ดที่มี ผลลัพธ์แม่นยำและเที่ยงตรง รวมถึงวิธีการที่จะใช้ในการแสดงผลลัพธ์จากการตรวจสอบการคัดลอก ชอร์สโค้ดที่ช่วยให้ผู้ใช้งานระบบสามารถมองเห็นถึงความเชื่อมโยงของผลลัพธ์จากการตรวจสอบการ คัดลอกชอร์สโค้ดได้อย่างสะดวกและมีประสิทธิภาพ เพื่อให้สามารถพัฒนาระบบดังกล่าวได้อย่าง ครบถ้วนสมบูรณ์ ผู้ดำเนินโครงการจึงได้ทำการค้นคว้าหาข้อมูลจากสื่อหลากหลายประเภท ไม่ว่าจะเป็น หนังสือ คลิปวิดีโอ บทความออนไลน์ และการสอบถามกับอาจารย์ที่ปรึกษา โดยมีรายละเอียด ดังต่อไปนี้

### 2.2.1 ความคล้ายคลึงของชอร์สโค้ด (Source Code Similarity)

การตรวจสอบส่วนที่เหมือนกันหรือคล้ายคลึงกันระหว่างชุดข้อมูลชอร์สโค้ด 2 ชุด โดยการ ตรวจสอบความคล้ายคลึงของชอร์สโค้ดนี้ มีวัตถุประสงค์เพื่อค้นหาชอร์สโค้ดในส่วนที่เหมือนกัน หรือคัดลอกกันมา อันเป็นวิธีการที่จะช่วยให้เห็นถึงความสัมพันธ์ระหว่างชุดข้อมูลชอร์สโค้ดชุดนั้นๆ ซึ่งผลลัพธ์ของการแสดงความคล้ายคลึงของชอร์สโค้ดจะอยู่ในรูปของตัวเลข ซึ่งเป็นตัวเลขที่แสดงถึง ร้อยละความคล้ายคลึงของชอร์สโค้ดระหว่างชุดข้อมูลชอร์สโค้ด 2 ชุด อันเป็นข้อมูลเพื่อผู้ใช้งาน สามารถทำความเข้าใจว่าข้อมูลชอร์สโค้ดทั้ง 2 ชุดนั้นมีความสัมพันธ์กันอย่างไร ซึ่งระบบแสดง ผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดนี้ ได้ถูกออกแบบมาเพื่อให้สามารถตรวจสอบ

ชอร์สโค้ดที่ถูกเขียนขึ้นโดยภาษาโปรแกรมภาษาเดียวกันเท่านั้น ซึ่งในการตรวจสอบความคล้ายคลึงของชอร์สโค้ด ระบบจะนำไฟล์ข้อมูลชอร์สโค้ดจำนวน 2 ไฟล์ มาเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน โดยเริ่มต้นจากการนำชอร์สโค้ดในส่วนที่เป็นการเรียกใช้คำสั่งหรือไลบรารีของภาษานั้นๆ จากนั้น ระบบจะทำการเปรียบเทียบชอร์สโค้ดในส่วนที่เหลืออยู่ โดยจะเป็นการเทียบชอร์สโค้ดทั้งในรูปแบบของภาพรวมและรูปแบบบรรทัดต่อบรรทัด เมื่อตรวจสอบเสร็จแล้ว ระบบก็จะสร้างไฟล์ผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ดขึ้นมา

## 2.2.2 เครื่องมือที่ใช้ตรวจสอบความคล้ายคลึงของชอร์สโค้ด

เครื่องมือที่ถูกพัฒนาขึ้นสำหรับใช้ในการตรวจสอบความคล้ายคลึงของชอร์สโค้ดโดยเฉพาะโดยในปัจจุบัน ได้มีการพัฒนาเครื่องมือสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ดอยู่จำนวนมาก ซึ่งแต่ละเครื่องมือก็มีจุดเด่นและข้อจำกัดที่แตกต่างกันออกไป โดยเครื่องมือที่ผู้ดำเนินโครงการได้ทำการศึกษาโครงสร้างและวิธีการทำงาน โดยมีรายละเอียด ดังนี้

### 1) JPlag

ไลบรารีที่ถูกสร้างขึ้นเพื่อใช้สำหรับประมวลการคัดลอกชอร์สโค้ด โดยไลบรารีจะทำการตรวจสอบชอร์สโค้ดที่ผู้ใช้งานต้องการ ผ่านการนำเข้าข้อมูลชอร์สโค้ด (Import data) และส่งออกผลลัพธ์ของการตรวจสอบในรูปแบบข้อมูลที่อยู่ในโครงสร้างข้อมูลแบบ JavaScript Object Notation (JSON) ซึ่งไลบรารีรองรับการตรวจสอบชอร์สโค้ดที่ถูกเขียนขึ้นจากหลากหลายภาษา ยกตัวอย่างเช่น Java, C++, C#, Go, Python, Scala และ Swift เป็นต้น

### 2) Moss

Moss เป็นซอฟต์แวร์สำหรับตรวจสอบความคล้ายคลึงกันของซอฟต์แวร์ โดยมีวัตถุประสงค์เพื่อตรวจจับการคัดลอกผลงานหรือชอร์สโค้ดภายในชั้นเรียนการเขียนโปรแกรมซึ่งรองรับภาษาสำหรับเขียนโปรแกรม ยกตัวอย่างเช่น C, C++, Java, C#, Python, Visual Basic, Scheme, Pascal และ JavaScript เป็นต้น

### 3) Plaggle

เครื่องมือตรวจจับการลอกเลียนแบบชอร์สโค้ดสำหรับใช้ในรายวิชาการเขียนโปรแกรมภาษา Java โดยมีการทำงานและหน้าจอแสดงผลในรูปแบบแบบกราฟิก ช่วยให้ผู้ใช้งานสามารถใช้งานได้อย่างง่ายดาย ซึ่งผู้ใช้งานจำเป็นต้องติดตั้ง Plaggle ลงบนระบบปฏิบัติการเพื่อใช้งาน

#### 4) Marble

โปรแกรมสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ด โดยมุ่งเน้นไปที่การตรวจสอบโปรแกรมที่ถูกเขียนขึ้นด้วยภาษา Java เป็นหลัก มีการประมวลผลที่รวดเร็ว ทำให้ผู้ใช้งานสามารถใช้งานและเข้าถึงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ดได้อย่างสะดวกรวดเร็ว เครื่องมือสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ดที่ผู้ดำเนินโครงการได้ศึกษาและทำความเข้าใจการทำงานนั้น มักมีคุณสมบัติและผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ดที่แตกต่างกันไปในแต่เครื่องมือ ดังนั้น ผู้ดำเนินโครงการจึงได้สรุปรายละเอียดของแต่ละเครื่องมือ ดังตารางที่ 2-2

ตารางที่ 2-2 การเปรียบเทียบเครื่องมือที่ใช้ในการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

หัวข้อ	JPlag	Moss	Plaggie	Marble
1. แหล่งอ้างอิง	<a href="https://github.com/jplag/JPlag">https://github.com/jplag/JPlag</a>	<a href="https://theory.stanford.edu/~aiken/moss/">https://theory.stanford.edu/~aiken/moss/</a>	<a href="https://www.cs.hut.fi/Software/Plaggie/">https://www.cs.hut.fi/Software/Plaggie/</a>	<a href="https://webspace.science.uu.nl/~hage0101/downloads/marble-slides.pdf">https://webspace.science.uu.nl/~hage0101/downloads/marble-slides.pdf</a>
2. ประเภทซอฟต์แวร์	Open Source	Freeware	Freeware	Freeware
3. จำนวนภาษาที่รองรับ	12	26	1	1
4. แพลตฟอร์มที่รองรับ	Web	Web	Local	Local

จากตารางที่ 2-2 พบร่วมกับ Plaggie และ Marble ในหัวข้อของจำนวนภาษาที่รองรับ มีเพียง 1 ภาษาเท่านั้น จึงไม่เหมาะสมที่จะเป็นเครื่องมือสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ดภายในสถาบันการศึกษา เนื่องจากรายวิชาเขียนโปรแกรมภายใต้สถาบันการศึกษาอาจมีการเขียนโปรแกรมที่ใช้ภาษามากกว่า 1 ภาษา ดังนั้น เครื่องมือที่ผู้ดำเนินโครงการให้ความสนใจที่หลงเหลืออยู่คือ JPlag และ Moss ซึ่งมีจำนวนภาษาที่รองรับจำนวนมาก เช่นเดียวกัน ผู้ดำเนินโครงการจึงให้ความสนใจในหัวข้อประเภทของซอฟต์แวร์ โดยในที่นี้พบว่า Moss เป็นเครื่องมือประเภท Freeware ซึ่งมีเงื่อนไขหรือลิขสิทธิ์ในการปรับแก้การทำงานของเครื่องมือและสามารถนำเครื่องมือที่ปรับแก้แล้วมาใช้ในขณะเดียวกัน JPlag ซึ่งเป็นเครื่องมือประเภท Open Source ที่สามารถนำเครื่องมือมาปรับ

และใช้งานได้อย่างอิสระโดยไม่มีเสียค่าใช้จ่าย จึงเป็นเหตุผลที่ผู้ดำเนินโครงการเลือกใช้ JPlag เป็นเครื่องมือสำหรับใช้ในการพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด

### 2.2.3 Data Visualization

การแปลงข้อมูลในลักษณะของข้อความให้มาอยู่ในรูปแบบของแผนภาพ กราฟ หรือวิดีโอ มีวัตถุประสงค์เพื่อใช้ในการนำเสนอความสัมพันธ์ของข้อมูลชุดนั้นๆ ตามความต้องการของผู้จัดทำ ทำให้ผู้รับฟังสามารถทำความเข้าใจข้อมูลได้อย่างรวดเร็วและซัดเจน โดยเฉพาะอย่างยิ่งกับชุดข้อมูล ที่มีข้อมูลจำนวนมากและมีการเปลี่ยนแปลงของข้อมูลอย่างต่อเนื่อง ซึ่งในปัจจุบันนี้ มีเครื่องมือ สำหรับจัดทำ Data Visualization ที่อำนวยความสะดวกให้ผู้จัดทำอยู่มากมายและหลากหลาย ทั้งแบบที่ไม่มีค่าใช้จ่ายเพิ่มเติมและมีค่าใช้จ่ายเพิ่มเติม โดยการแสดงผลลัพธ์จากการตรวจสอบ ความคล้ายคลึงของระบบ ผู้ดำเนินโครงการได้เลือกการแสดงผลในรูปแบบของกราฟความสัมพันธ์ ซึ่งเป็นรูปแบบที่เหมาะสมสำหรับให้ผู้ใช้งานทำความเข้าใจผลลัพธ์ของการตรวจสอบความคล้ายคลึง ของชอร์สโค้ด รวมถึงการแสดงผลลัพธ์ในรูปแบบของตารางผลลัพธ์ความคล้ายคลึงของชอร์สโค้ด สำหรับให้ผู้ใช้งานดูรายละเอียดของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

### 2.2.4 การแสดงผลลัพธ์ในรูปแบบของตารางและกราฟ

รูปแบบการแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด ผู้ดำเนินโครงการ ได้เลือกรูปแบบของตาราง (Table) และกราฟ (Graph) ซึ่งเป็นวิธีการที่เหมาะสมในการแสดงผลลัพธ์ จากการประมวลผลของ JPlag โดยมีรายละเอียด ดังนี้

ไลบรารีสำหรับแสดงผลข้อมูลในรูปแบบของกราฟบนหน้าเว็บไซต์ เป็นการจัดรูปแบบ ข้อมูลให้เป็นระเบียบ สามารถดูรายละเอียดและข้อเปรียบเทียบที่ต้องการได้ง่าย ซึ่งตาราง ที่ ผู้ดำเนินโครงการได้นำมาใช้ โดยมีรายละเอียด ดังนี้

1.1) คอลัมน์ Matched คือ ความสัมพันธ์ระหว่างผู้เป็นเจ้าของชุดข้อมูลชอร์สโค้ด ที่ผ่านการตรวจสอบความคล้ายคลึงของชอร์สโค้ดโดย JPlag

1.2) คอลัมน์ Source Code Similarity (%) คือ ผลลัพธ์ร้อยละความคล้ายคลึงกัน ของชุดข้อมูลชอร์สโค้ดที่ผ่านการตรวจสอบความคล้ายคลึงของชอร์สโค้ดโดย JPlag ตัวอย่าง ตารางข้อมูล แสดงดังภาพที่ 2-1

Matched	Source Code Similarity (%)
Bob - Alice	100.00
Bob - Jack	80.00
Alice - Jack	50.00

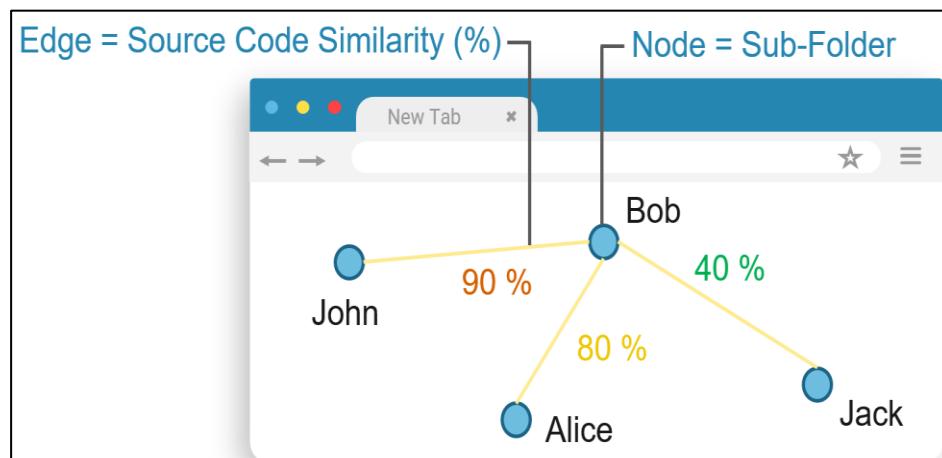
ภาพที่ 2-1 การประยุกต์ใช้ Bootstrap Table

ไลบรารีสำหรับแสดงผลข้อมูลในรูปแบบของกราฟบนหน้าเว็บไซต์ โดยเป็นไลบรารีรุ่นพัฒนาต่อยอดมาจากการไลบรารีชื่อ Chart.js ซึ่งเป็นไลบรารีสำหรับจัดการการแสดงผลข้อมูลในรูปแบบของแผนภูมิหรือกราฟประเภทต่างๆ อันเป็นเครื่องมือสำคัญที่ช่วยให้การแสดงผลข้อมูลในรูปแบบที่ช่วยให้ผู้ใช้งาน สามารถทำความเข้าใจข้อมูลหรือวิเคราะห์ข้อมูลเชิงลึกได้อย่างสะดวกและมีประสิทธิภาพมากยิ่งขึ้น โดยผู้ดำเนินโครงการได้มีการนำเอาไลบรารีตัวนี้ มาประยุกต์ใช้กับการแสดงผลลัพธ์จากการตรวจสอบการคัดลอกชอร์สโค้ด

ผู้ดำเนินโครงการได้ทำการศึกษาโครงสร้างของไลบรารี Force-directed graph และนำเอาสิ่งที่วิเคราะห์ได้ มาประยุกต์ใช้กับการแสดงผลลัพธ์ที่ได้จากการตรวจสอบการคัดลอกชอร์สโค้ดโดยไลบรารี JPlag ซึ่งมีรายละเอียด ดังนี้

2.1) Node หมายถึง โฟลเดอร์หรือไดเรกทอรีอยู่ ซึ่งใช้แทนผู้เป็นเจ้าของชอร์สโค้ด ชุดหนึ่งๆ โดยแต่ละ Node จะมีการแสดงชื่อของผู้เป็นเจ้าของชุดชอร์สโค้ดกำกับเอาไว้

2.2) Edge หมายถึง ความสัมพันธ์ระหว่าง Node ซึ่งในที่นี้คือร้อยละความคล้ายคลึงกันระหว่างชุดชอร์สโค้ด (Source Code Similarity) ตัวอย่างกราฟความสัมพันธ์ แสดงดังภาพที่ 2-2



ภาพที่ 2-2 การประยุกต์ใช้ Force-directed Graph

## 2.3 เครื่องมือและเทคโนโลยีที่ใช้ในการดำเนินโครงการ

การพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด มีการใช้เครื่องมือและเทคโนโลยีต่างๆ เช่นมาช่วยในการพัฒนาระบบ เพื่ออำนวยความสะดวกในการพัฒนาระบบ และเพื่อพัฒนาระบบที่มีการทำงานที่ครอบคลุมของเขตที่ได้กำหนดเอาไว้อย่างครบถ้วนสมบูรณ์ ซึ่งสามารถแบ่งประเภทของเครื่องมือและเทคโนโลยีที่ใช้ในการดำเนินโครงการได้ 3 ส่วน คือ ภาษาที่ใช้ในการพัฒนา ซอฟต์แวร์ที่ใช้ในการพัฒนา และเฟรมเวิร์กที่ใช้ในการพัฒนา โดยมีรายละเอียด ดังต่อไปนี้

### 2.3.1 ภาษาที่ใช้ในการพัฒนา

ภาษาสำหรับการเขียนโปรแกรมเพื่อพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึง ของชอร์สโค้ด ประกอบไปด้วยทั้งหมด 4 ภาษา ได้แก่ TypeScript, HTML, CSS, และ JavaScript โดยข้อมูลของแต่ละภาษา โดยมีรายละเอียด ดังนี้

#### 1) TypeScript

เป็นภาษาคอมพิวเตอร์ที่พัฒนาขึ้นโดยบริษัทไมโครซอฟท์ โดยได้รับการร่วมมือ จาก 3 บริษัททางด้านไอทีและเทคโนโลยีระดับโลก ซึ่งประกอบไปด้วย Google, Microsoft และ Facebook ซึ่งมีการพิจารณาตรงกันว่าภาษา JavaScript นั้นเป็นภาษาที่ทำให้ผู้พัฒนาทำการตัดจับหรือตรวจสอบข้อผิดพลาดของระบบได้ยาก เนื่องจากภาษา JavaScript เป็นภาษาที่เป็น Interpreted Language ที่ไม่จำเป็นต้องทำการคอมไพล์โปรแกรมก่อนที่จะทำการรัน โปรแกรมนั้นขึ้นมา ทำให้ผู้พัฒนาอาจตรวจสอบแล้วไม่พบข้อผิดพลาดที่ซ่อนอยู่ภายในโปรแกรม ซึ่งอาจส่งผลกระทบแรงให้โปรแกรมทำงานผิดพลาด ไม่ตรงกับที่ผู้พัฒนาได้กำหนดการทำงานเอาไว้ นอกจากนี้ในด้านของการประกาศตัวแปร ภาษา JavaScript มีการประกาศตัวแปร โดยไม่ต้องระบุประเภทของตัวแปร (Dynamic type) นำมาซึ่งการกำหนดค่าใส่ตัวแปรอาจมีความผิดพลาดขึ้น เนื่องจากหากระบุค่าของตัวแปรไม่ตรงกับการทำงานของโปรแกรมที่กำหนดไว้อาจส่งผลให้โปรแกรมทำงานผิดพลาดไป ซึ่งให้ผู้พัฒนาสามารถมองเห็นข้อผิดพลาดของระบบได้อย่างชัดเจนมากยิ่งขึ้น โดยภาพลักษณ์ของภาษา TypeScript แสดงดังภาพที่ 2-3



ภาพที่ 2-3 สัญลักษณ์ของ TypeScript

จากโครงสร้างการทำงานของภาษา JavaScript ที่อาจส่งผลกระทบให้ระบบทำงานผิดพลาด ดังนั้น ภาษา TypeScript จึงถูกพัฒนาขึ้นมาเพื่อกำจัดซ่อง霍่ที่เกิดขึ้น กับภาษา JavaScript ซึ่งเพิ่มความสามารถในด้านของการตักจับข้อผิดพลาด โดยภาษา TypeScript จะมีการตักจับข้อผิดพลาดที่เกิดขึ้นระหว่างการคอมไพล์โปรแกรม และแจ้งข้อผิดพลาดเหล่านั้นให้ผู้พัฒนาได้รับทราบและทำการแก้ไขก่อนที่โปรแกรมจะถูกแปลงเป็นภาษา JavaScript เพื่อนำไปใช้งานจริง ซึ่งการทำงานนี้จะช่วยลดโอกาสที่จะเกิดข้อผิดพลาดที่ซ่อนอยู่ โดยที่ผู้พัฒนาไม่ทราบได้

## 2) HTML (HyperText Markup Language)

เป็นภาษา HTML ถูกพัฒนาขึ้นภายในศูนย์ปฏิบัติการวิจัยทางอนุภาคพิสิกส์ของยุโรป (CERN) โดยทิม เบอร์เนอร์ส-ลี (Tim Berners Lee) ซึ่งมีแนวความคิดที่จะให้นักวิทยาศาสตร์ภายในสถาบันสามารถติดต่อสื่อสารกันได้อย่างสะดวก ซึ่งได้มีการพัฒนาวิธีการที่จะช่วยให้การติดต่อสื่อสาร หรือการส่งต่อข้อมูลซึ่งอยู่ในรูปแบบเอกสารที่แต่ละหน้าเชื่อมโยงกันอย่างมีโครงสร้าง จึงได้มีการการพัฒนาเครื่องมือที่เรียกว่าเบราว์เซอร์ (Web Browser) เพื่อใช้ในการอ่านข้อมูล และส่งต่อเอกสารโดยใช้ HTTP (Hypertext Transport Protocol) ซึ่งเป็นโปรโตคอล (Protocol) หรือมาตรฐานในการรับส่งข้อมูลเพื่อใช้ในการติดต่อสื่อสารระหว่างนักวิทยาศาสตร์ภายในศูนย์วิจัย โดยภาพสัญลักษณ์ของภาษา HTML แสดงดังภาพที่ 2-4



ภาพที่ 2-4 สัญลักษณ์ของ HTML

ต่อมา HTML ได้รับการพัฒนาอย่างต่อเนื่องจนกลายเป็นภาษาคอมพิวเตอร์ที่มีบทบาทอย่างมากในการพัฒนาเว็บไซต์หรือเว็บแอปพลิเคชันในปัจจุบัน ซึ่งโครงสร้างของภาษา HTML จะอยู่ในรูปแบบของ Tag ต่างๆ โดยมีการทำงานและถูกประมวลผลโดยเว็บเบราว์เซอร์ จะแปลความหมายของ Tag ที่ถูกเขียนขึ้นโดยภาษา HTML เป็นหน้าตาของเว็บไซต์ นอกจากนี้ด้วยความสามารถของภาษา HTML ที่มีลักษณะเป็นเอกสารแบบไฮเปอร์เทกซ์ซึ่งมีคุณสมบัติที่สามารถเชื่อมโยงไปยังหน้าเอกสารอื่นหรือหน้าเว็บไซต์หน้าอื่นๆ ทำให้การเชื่อมโยงไปยังข้อมูลส่วนต่างๆ เป็นไปอย่างสะดวกและรวดเร็วต่อผู้ใช้งาน ซึ่งในปัจจุบัน ภาษา HTML ได้ถูกพัฒนาอย่างต่อเนื่อง ซึ่งช่วยให้การปรับแต่งและการจัดวางโครงสร้างขององค์ประกอบต่างๆ บนเว็บไซต์นั้นเป็นไปอย่างมีโครงสร้างและมีประสิทธิภาพ

### 3) CSS (Cascading Style Sheets)

ภาษา CSS ถูกพัฒนาขึ้นมาเพื่อใช้งานร่วมกับภาษา HTML ในด้านของการตกแต่งหน้าจอเว็บไซต์หรือหน้าเอกสารโดยการใส่สีสัน การกำหนดระยะห่างระหว่างองค์ประกอบ การเพิ่มกรอบข้อความ การกำหนดสีพื้นหลัง การกำหนดฟอนต์ของตัวหนังสือ การกำหนดขนาดพื้นที่แสดงผล การกำหนดขนาดของตัวอักษร และอื่นๆ เพื่อเพิ่มความสวยงามและความดึงดูดน่าใช้งานให้กับผู้ใช้งาน โดย CSS มักถูกใช้งานร่วมกับไลบรารีต่างๆ ที่ช่วยให้การตกแต่งหน้าจอแสดงผลที่ง่ายต่อการใช้งานสำหรับผู้พัฒนา ซึ่งช่วยประหยัดเวลาในการพัฒนาและทำให้หน้าจอแสดงผลมีความสวยงามมากยิ่งขึ้น โดยภาพสัญลักษณ์ของภาษา CSS แสดงดังภาพที่ 2-5



ภาพที่ 2-5 สัญลักษณ์ของ CSS

โดยภาษา CSS เริ่มพัฒนาขึ้นในปี ค.ศ. 1994 ซึ่งมีวัตถุประสงค์ในการลดการใช้ภาษา HTML ใน การตกแต่งเอกสารหรือหน้าเว็บไซต์ ทำให้รหัสโค้ด (Source Code) ภายในเอกสาร HTML เหลือเพียงส่วนเนื้อหาของเนื้อหาและองค์ประกอบต่างๆ ซึ่งช่วยให้ผู้พัฒนาสามารถทำการคัดแยกระหว่างส่วนของการแสดงผลและส่วนของการตกแต่งได้อย่างชัดเจน โดยที่ภาษา CSS ได้มีการกำหนดคำสั่งในการระบุรูปแบบ ของเนื้อหาในเอกสาร อันได้แก่ สีของข้อความ สีพื้นหลัง

ประเภทตัวอักษร และการจัดวางข้อความซึ่งการกำหนดรูปแบบ หรือที่เรียกว่า สไตล์ (Style) ใช้หลักการของการแยกเนื้อหาเอกสาร HTML ออกจากคำสั่งที่ใช้ในการจัดรูปแบบการแสดงผลเพื่อความสะดวกและง่ายต่อการจัดรูปแบบ การแสดงผลลัพธ์ของเอกสาร HTML โดยเฉพาะในกรณีที่มีการเปลี่ยนแปลงเนื้อหาเอกสารบ่อยครั้ง หรือต้องการควบคุมให้รูปแบบการแสดงผลเอกสาร HTML มีลักษณะของความสม่ำเสมอทั่วทุกหน้าของเอกสารที่แสดงผลภายใต้เว็บไซต์เดียวกัน เพื่อความสม่ำเสมอในการแสดงผลของแต่ละหน้าของเว็บไซต์ ซึ่งเป็นส่วนสำคัญที่ช่วยให้ผู้ใช้งานสามารถใช้งานเว็บไซต์ได้อย่างสะดวก และยังช่วยให้ผู้ใช้งานสามารถทำความคุ้นเคยและแยกแยะองค์ประกอบต่างๆ ที่แสดงผลบนหน้าเว็บไซต์ได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

#### 4) JavaScript

JavaScript เป็นภาษาโปรแกรมที่ถูกสร้างขึ้นเพื่อในการสำหรับการสร้างหน้าเว็บไซต์ในด้านของการควบคุมเหตุการณ์ต่างๆ ที่เกิดขึ้นบนหน้าเว็บไซต์ โดยความสามารถของ JavaScript จะช่วยในการปรับปรุงประสบการณ์ที่ผู้ใช้จะได้รับจากการใช้งานเว็บไซต์ นอกจากนี้ JavaScript สามารถช่วยในการจัดการงานต่างๆ โดยไม่มีความจำเป็นที่จะต้องทำการโหลดหน้าเว็บใหม่ๆ รวมถึงใช้ในการกำหนดการแสดงผลของข้อความบางข้อความบนหน้าเว็บไซต์หรือซ่อนข้อความนั้น และการควบคุมการเปลี่ยนแปลงสีหรือขนาดตัวหนังสือของตัวอักษร โดยภาพสัญลักษณ์ของภาษา JavaScript แสดงดังภาพที่ 2-6



ภาพที่ 2-6 สัญลักษณ์ของ JavaScript

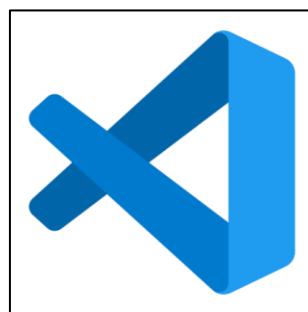
ในช่วงแรกของการพัฒนา ภาษา JavaScript ถูกพัฒนาขึ้นมาสำหรับเป็นภาษาเพื่อทำงานบนหน้าเว็บไซต์ที่ทำงานบนเบราว์เซอร์เท่านั้น แต่ในปัจจุบัน ได้มีการพัฒนาอย่างต่อเนื่อง นั้นทำให้ในปัจจุบันการเขียนโปรแกรมภาษา JavaScript ไม่ใช่เพื่อสำหรับจัดการหน้าจอแสดงผลบนเว็บไซต์เพียงอย่างเดียว ดังนั้น ภาษา JavaScript จึงได้รับความนิยมเป็นอย่างสูง รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA โดยการทำงานของ JavaScript จะต้องมีการแปลความคำสั่ง ซึ่งขึ้นตอนนี้จะถูกจัดการโดยเบราว์เซอร์

### 2.3.2 ซอฟต์แวร์ที่ใช้ในการพัฒนา

ในการพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดนอกเหนือจากภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนาแล้ว ผู้ดำเนินโครงการยังมีการใช้เครื่องมือหรือซอฟต์แวร์ที่ช่วยอำนวยความสะดวกในการพัฒนาระบบ โดยมีรายละเอียด ดังต่อไปนี้

#### 1) Visual Studio Code

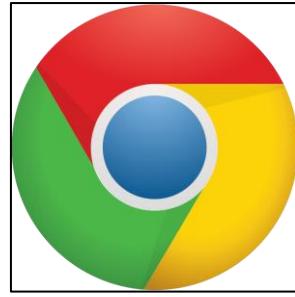
Visual Studio Code คือเครื่องมือสำหรับพัฒนาซอฟต์แวร์ ถูกพัฒนาขึ้นโดยบริษัท Microsoft เป็น Opensource Software ผู้พัฒนาสามารถใช้งานได้โดยไม่มีค่าใช้จ่ายเพิ่มเติมรองรับการใช้งานทั้งบนระบบปฏิบัติการที่แตกต่างกัน และรองรับภาษาโปรแกรมมากมายโดยมาพร้อมกับเครื่องมือและไลบรารีเสริมต่างๆ ให้ผู้พัฒนาเลือกใช้งาน เช่น เครื่องมือสำหรับการทำ Debug เพื่อตรวจสอบข้อผิดพลาดของการเขียนโปรแกรม มีโครงสร้างการทำงานที่ไม่ซับซ้อน สามารถเรียนรู้และใช้งานได้ง่าย ซึ่งเป็นเครื่องมือสำหรับการเขียนโปรแกรมที่มีผู้ใช้งานเป็นจำนวนมาก และมีชุมชนสำหรับให้ผู้พัฒนาได้พูดคุยแลกเปลี่ยนความรู้กัน โดยภาพสัญลักษณ์ของโปรแกรม Visual Studio Code แสดงดังภาพที่ 2-7



ภาพที่ 2-7 สัญลักษณ์ของ Visual Studio Code

#### 2) Google Chrome

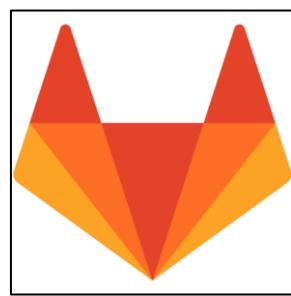
เว็บเบราว์เซอร์ที่ใช้สำหรับค้นหาเว็บไซต์ พัฒนาขึ้นโดยบริษัท Google ได้รับความนิยมจากผู้ใช้งานที่ต้องการสืบค้นข้อมูลต่างๆ บนเครือข่ายอินเทอร์เน็ตทั่วโลกและมีความปลอดภัยในการเข้าใช้งานสูง ซึ่งมาพร้อมกับโปรแกรมส่วนเสริมอีกมากมาย ถือว่ามีจุดเด่นในการรักษาประสิทธิภาพด้านความปลอดภัยและปกป้องข้อมูลการค้นหาเว็บไซต์ในด้านของความเป็นส่วนตัวของข้อมูลของผู้ใช้งาน และในมุมมองของผู้พัฒนาเว็บแอปพลิเคชัน Google Chrome ก็มีเครื่องมือที่ช่วยให้ผู้พัฒนาสามารถตรวจสอบและควบคุมการแสดงผลของเว็บแอปพลิเคชันได้อย่างสะดวกและคล่องแคล่ว โดยภาพสัญลักษณ์ของ Google Chrome แสดงดังภาพที่ 2-8



ภาพที่ 2-8 สัญลักษณ์ของ Google Chrome

### 3) GitLab

GitLab คือเครื่องมือที่ใช้สำหรับจัดเก็บซอฟต์แวร์สโคడ์และควบคุมเวอร์ชันของซอฟต์แวร์โดยใช้ Git ซึ่งเป็นเครื่องมือที่ผู้สร้างขึ้นมาเพื่อควบคุมเวอร์ชันของซอฟต์แวร์ ซึ่ง GitLab ทำงานอยู่บนเครือข่ายอินเทอร์เน็ต ซึ่งผู้ใช้งานสามารถเข้าถึงข้อมูลและจัดการไปผ่านเว็บเบราว์เซอร์โดยไม่ต้องเสียค่าใช้จ่ายเพิ่มเติม ทำให้ผู้ใช้งานสามารถเข้าถึงข้อมูลซอฟต์แวร์สโคడ์ของตัวเองได้โดยเข้าใช้งานผ่านหน้าเว็บไซต์ได้ทุกที่ ทุกเวลา และในการใช้งาน ผู้ใช้สามารถควบคุมและบริหารจัดการโครงการและซอฟต์แวร์สโคడ์ ได้อย่างสะดวกและมีประสิทธิภาพ และมีชุมชนสำหรับให้ผู้พัฒนาสามารถพูดคุย ปรึกษาปัญหา หรือแลกเปลี่ยนความคิดเห็นกันได้อย่างกว้างขวาง โดยภาพสัญลักษณ์ของ GitLab แสดงดังภาพที่ 2-9



ภาพที่ 2-9 สัญลักษณ์ของ GitLab

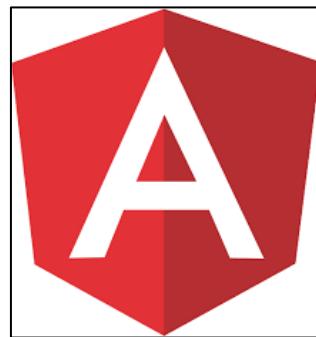
#### 2.3.3 เฟรมเวิร์กที่ใช้ในการพัฒนา

เฟรมเวิร์กคือเครื่องมือสำคัญในการพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอฟต์แวร์สโคడ์ ซึ่งช่วยให้ผู้พัฒนาสามารถพัฒนาระบบในส่วนของการจัดวางองค์ประกอบบนหน้าจอแสดงผล และส่วนของการติดต่อกับเชิร์ฟเวอร์ นอกจากนี้ เฟรมเวิร์กสำหรับพัฒนาเว็บแอปพลิเคชันยังช่วยให้ติดต่อกับฐานข้อมูลหรือติดต่อกับเชิร์ฟเวอร์ได้อย่างง่ายดาย โดยมีขั้นตอน

และกระบวนการทำงานที่น่าเบื่อที่ใช้ในการบันทึกและเก็บรักษาข้อมูลของผู้ใช้ เฟรมเวิร์กเหล่านี้ ยังช่วยในการจัดระเบียบองค์ประกอบต่างๆที่แสดงผลบนหน้าเว็บแอปพลิเคชันได้อย่างสวยงาม และอำนวยความสะดวกในการพัฒนาให้กับผู้พัฒนา โดยมีรายละเอียด ดังต่อไปนี้

### 1) Angular

Angular เป็นเครื่องมือที่ใช้สำหรับพัฒนาเว็บแอปพลิเคชันในด้านของ Front-end Web Development โดยเฉพาะ ซึ่งทำงานในฝั่งของคลาสเซ้นต์ ในรูปแบบของภาษา HTML, ภาษา CSS และภาษา JavaScript หรือภาษา TypeScript ซึ่ง TypeScript จะถูกประมวลผล และตรวจสอบข้อผิดพลาดของโครงสร้างและไวยากรณ์ ก่อนจะได้รับการแปลงไปเป็นภาษา JavaScript สำหรับใช้งานบนระบบจริง โดย Angular มีหน้าที่คือเป็นเครื่องมือซึ่งช่วยให้ผู้พัฒนา สามารถพัฒนาเว็บแอปพลิเคชัน ได้อย่างสะดวกและรวดเร็วอย่างมีโครงสร้าง โดยมีความสามารถ ในการจัดวางโครงสร้างขององค์ประกอบบนหน้าจอแสดงผล และการควบคุมการไหลของข้อมูล ไปยังส่วนต่างๆ ของเว็บแอปพลิเคชัน ซึ่งผู้พัฒนาสามารถติดตั้งไลบรารีหรือส่วนเสริมซึ่งเป็นจุดเด่น ของภาษา TypeScript ที่ได้รับการพัฒนาต่ออยอดมาจากการ JavaScript ดังนั้น Angular จึงเป็นเครื่องมือสำหรับพัฒนาเว็บแอปพลิเคชันที่ได้รับความนิยมอย่างรวดเร็วจากนักพัฒนา เว็บแอปพลิเคชันทั่วโลก โดยภาพสัญลักษณ์ของ Angular แสดงดังภาพที่ 2-10



ภาพที่ 2-10 สัญลักษณ์ของ Angular

Angular ถูกพัฒนาขึ้นโดยบริษัท Google เพื่อลดปัญหาด้านของการใช้งานของข้อมูล ในเครื่องเซิร์ฟเวอร์ โดยมีวัตถุประสงค์เพื่อใช้ในการพัฒนาเว็บแอปพลิเคชันแบบ Single-page Application ซึ่งเปรียบเสมือนการจัดวางองค์ประกอบหลายๆ ส่วนของหน้าจอแสดงผลมาจัดวาง บนหน้าเว็บเพจเพียงหน้าเดียว ซึ่งช่วยให้ประสิทธิภาพในการแสดงผลของหน้าเว็บเพจเสถียร มากยิ่งขึ้นและใช้ทรัพยากรลดลง นอกเหนือนี้ ยังมาพร้อมกับเครื่องมือและไลบรารีเสริมต่างๆ ที่สามารถรองรับการทำงานร่วมกันกับภาษาโปรแกรมภาษาอื่นๆ ได้หลากหลาย

## 2) Node.js

ซอฟต์แวร์หรือเครื่องมือประเภท Open Source ซึ่งเปิดให้ผู้พัฒนาสามารถเข้าถึงและนำไปใช้งานได้โดยไม่มีค่าใช้จ่ายเพิ่มเติม มีการทำงานแบบ Asynchronous ซึ่งหมายความว่าการพัฒนาระบบหรือเซิร์ฟเวอร์ที่มีผู้ใช้งานจำนวนมากและมีการเปลี่ยนแปลงหรือติดต่อกับข้อมูลอยู่ตลอดเวลา โดยใช้ในการแปลงคำสั่งของ JavaScript เป็น JavaScript ที่สามารถทำงานบนระบบปฏิบัติการที่แตกต่างกันได้ Node.js ถูกสร้างขึ้นมาเพื่อทำงานผ่าน Server เป็นหลัก สามารถใช้ Node.js ในการพัฒนาแอปพลิเคชันหรือแม้แต่เว็บไซต์ฟรีได้ เพื่อใช้เป็นเว็บไซต์ฟรีที่สามารถรองรับจำนวนการใช้งานได้มากขึ้นกว่าเว็บเซิร์ฟเวอร์ในรูปแบบเดิมอย่าง Apache HTTP Server โดยใช้หลักของการเขียนโปรแกรมแบบ Asynchronous หรือ Non-blocking I/O ที่ใช้ฟังก์ชัน Callback และ Event Loop ในการจัดการกับ Request แทนการสร้าง Thread และ Process ที่ต้องใช้ทรัพยากรของเว็บเซิร์ฟเวอร์เป็นจำนวนมาก เพื่อทำให้หน้าเว็บสามารถตอบสนองได้แบบ Real-time สิ่งหนึ่งที่ทำให้ Node.js เลือกภาษา JavaScript เป็นภาษาหลักในการเขียนโปรแกรม รวมถึงทำให้การพัฒนาเว็บไซต์ทำได้ง่ายและสะดวกขึ้นสำหรับนักพัฒนา นอกจากนี้ Node.js ยังเป็นต้นแบบให้กับเฟรมเวิร์กต่างๆ ที่ได้รับการพัฒนาและต่อยอดความสามารถมากจาก Node.js อันเป็นการหลักฐานที่ยืนยันว่า Node.js มีคุณสมบัติและเป็นเครื่องมือที่เหมาะสมสำหรับใช้ในการพัฒนาแอปพลิเคชันโดยภาพสัญลักษณ์ของ Node.js แสดงดังภาพที่ 2-11



ภาพที่ 2-11 สัญลักษณ์ของ Node.js

## บทที่ 3

### รายละเอียดของการดำเนินโครงการ

การพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด ผู้ดำเนินโครงการได้ศึกษาเครื่องมือสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ด และระบบใกล้เคียงที่ถูกพัฒนาขึ้น สำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ดเฉพาะ จากนั้น ผู้ดำเนินโครงการได้ทำการวิเคราะห์ และออกแบบโครงสร้างการทำงานของระบบ ซึ่งเป็นการกำหนดองค์ประกอบโดยรวมของระบบ ประเภทของผู้ใช้งาน และปฏิสัมพันธ์ระหว่างผู้ใช้งานกับระบบ โดยเนื้อหาสาระในบทนี้ ครอบคลุม ในเรื่องของการออกแบบลำดับการทำงานของระบบ การออกแบบหน้าจอแสดงผล รวมถึงการทำงานของระบบในแต่ละมอดูล โดยใช้แผนภาพสำหรับอธิบายการทำงานของระบบ ประกอบไปด้วย แผนภาพยูสเคส (Use Case Diagram) แผนภาพคำอธิบายยูสเคส (Use Case Description) แผนภาพกิจกรรม (Activity Diagram) แผนภาพคลาส (Class Diagram) และแผนภาพลำดับการทำงานของระบบ (Sequence Diagram)

#### 3.1 การออกแบบการทำงานของระบบ

ลำดับการทำงานของระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด คือรายละเอียดการทำงานของระบบในส่วนของการนำเข้าไฟล์ข้อมูลชอร์สโค้ด จนถึงการแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด โดยมีรายละเอียด ดังภาพที่ 3-1

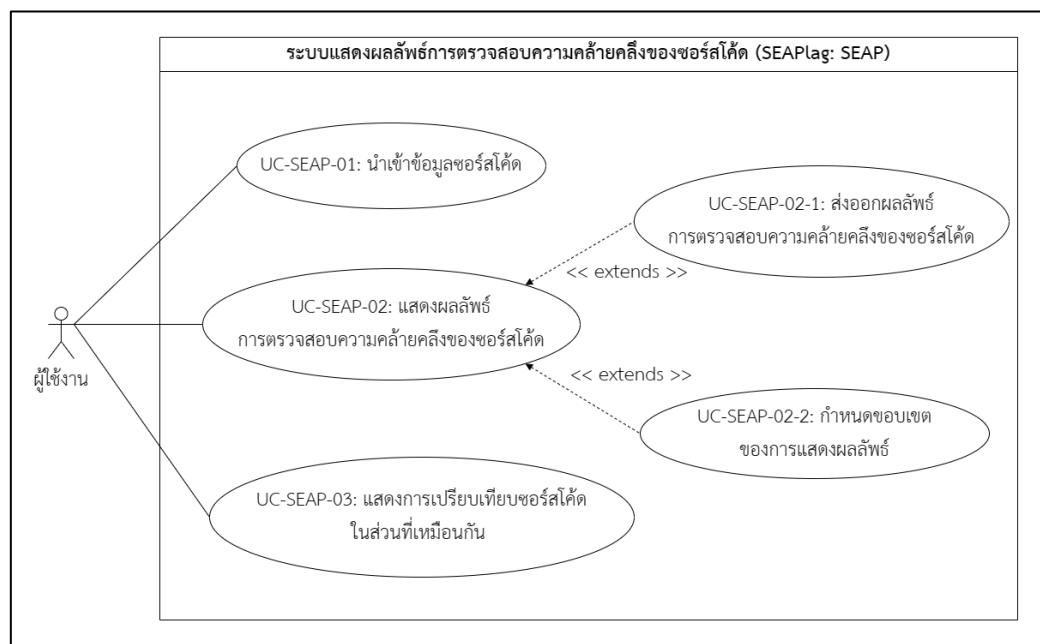


ภาพที่ 3-1 ลำดับการทำงานของระบบ

จากภาพที่ 3-1 มีการทำงานเริ่มต้นจากการที่ผู้ใช้งานจัดเตรียมไฟล์ข้อมูลของสโค์ดสำหรับนำเข้าเข้าระบบ เมื่อผู้ใช้งานนำไฟล์ของสโค์ดเข้าสู่ระบบแล้ว ระบบจะส่งไฟล์ของสโค์ดนั้นไปยังเครื่องเซิร์ฟเวอร์ เซิร์ฟเวอร์จะสั่งการให้ JPlag ประมวลผลสโค์ดที่อยู่ภายในไฟล์เพื่อหาความคล้ายคลึงของชอร์สโค้ด เมื่อ JPlag ประมวลผลเสร็จแล้ว เซิร์ฟเวอร์จะทำการส่งข้อมูลผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ดกลับไปยังระบบ ระบบจะนำข้อมูลการตรวจสอบความคล้ายคลึงของชอร์สโค้ดมาแสดงผลทางหน้าจอในรูปแบบของกราฟ

### 3.2 แผนภาพยุสเคส (Use Case Diagram)

ในการวิเคราะห์และออกแบบโครงสร้างการทำงานของระบบ ผู้ดำเนินโครงการได้จำแนก模ดูลการทำงานของระบบออกเป็น 5 มอดูล และผู้มีส่วนเกี่ยวข้องกับระบบในที่นี้คือผู้ใช้งาน โดยมีรายละเอียด ดังภาพที่ 3-2



ภาพที่ 3-2 รายละเอียดแผนภาพยุสเคส

จากภาพที่ 3-2 แผนภาพยุสเคสนี้ประกอบไปด้วย模ดูลของระบบจำนวน 5 มอดูล ได้แก่ มอดูลนำเข้าข้อมูลของสโค์ด มอดูลแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด มอดูลส่งออกผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด มอดูลกำหนดขอบเขตของการแสดงผลลัพธ์ และมอดูลแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน ซึ่งระบบมีผู้ใช้งาน ในที่นี้คือผู้ที่มีความต้องการในการใช้งานระบบสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ด โดยผู้ดำเนินโครงการได้รับมอบหมายให้พัฒนามอดูลของระบบจำนวน 4 มอดูล ซึ่งมอดูลที่ผู้ดำเนินโครงการได้รับผิดชอบ ประกอบไปด้วย มอดูลนำเข้าข้อมูลของสโค์ด มอดูล

แสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด มอດูลกำหนดขอบเขตของการแสดงผลลัพธ์ และมอດูลแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน

### 3.3 คำอธิบายยูสเคส (Use Case Description)

ในส่วนของคำอธิบายยูสเคส เป็นการอธิบายการทำงานของแต่ละมอડูล โดยอธิบายถึงข้อมูล ผู้มีส่วนเกี่ยวข้องกับการทำงานของมอડูล คำอธิบายการทำงานของมอડูล เงื่อนไขการทำงาน ของมอડูล เหตุการณ์ที่อาจทำให้ระบบทำงานผิดพลาด และลำดับขั้นตอนการทำงานของมอડูล ผู้ดำเนินโครงการได้รับมอบหมายให้พัฒนามอડูลจำนวนทั้งสิ้น 4 มอડูล มีรายละเอียดดังนี้

#### 3.3.1 มอଡูลนำเข้าข้อมูลชอร์สโค้ด (Import Source Code Module)

เริ่มต้นจากผู้ใช้งานจัดเตรียมไฟล์ข้อมูลชอร์สโค้ดสำหรับตรวจสอบความคล้ายคลึง ของชอร์สโค้ด โดยผู้ใช้งานต้องเลือกภาษาและนำเข้าไฟล์ข้อมูลชอร์สโค้ดที่มีนามสกุล zip เท่านั้น จึงจะสามารถนำไฟล์ชอร์สโค้ดเข้าสู่ระบบได้ โดยมีรายละเอียด ดังตารางที่ 3-1

ตารางที่ 3-1 นำเข้าข้อมูลชอร์สโค้ด

ชื่อยูสเคส: นำเข้าข้อมูลชอร์สโค้ด	รหัส: UC-SEAP-01	ระดับความสำคัญ: สูง
ผู้กระทำหลัก: ผู้ใช้งาน	ระดับความซับซ้อน: ปานกลาง	
ผู้มีส่วนเกี่ยวข้องและการใช้ประโยชน์: ผู้ใช้งาน		
คำอธิบาย: ผู้ใช้งานทำการเลือกภาษาโปรแกรมที่ต้องการตรวจสอบ และทำการนำเข้าข้อมูลชอร์สโค้ดที่ต้องการตรวจสอบความคล้ายคลึงของชอร์สโค้ด เพื่อนำไปใช้ในการตรวจสอบความคล้ายคลึงของชอร์สโค้ดภายในโปรแกรม		
สิ่งรับตุน: ผู้ใช้งานต้องการนำเข้าข้อมูลชอร์สโค้ดเพื่อตรวจสอบความคล้ายคลึงของชอร์สโค้ด		
ประเภทสิ่งรับตุน: ภายนอก		
ความสัมพันธ์:		
ความเกี่ยวเนื่อง: -		
การรวม: -		
การขยาย: -		
การรับทอดคุณสมบัติ: -		

ตารางที่ 3-1 นำเข้าข้อมูลชอร์สโค้ด (ต่อ)

ชื่อยูสเซอร์: นำเข้าข้อมูลชอร์สโค้ด	รหัส: UC-SEAP-01	ระดับความสำคัญ: สูง
เงื่อนไขก่อนการทำงาน	ผู้ใช้งานต้องทำการรวมไฟล์ และบันทึกในรูปแบบของไฟล์ ประเภท zip ไฟล์	
เงื่อนไขหลังการทำงาน	ผู้ใช้งานสามารถตรวจสอบความคล้ายคลึงของไฟล์ชอร์สโค้ด ที่ต้องการได้	
ขั้นตอนการทำงานปกติ	ผู้ใช้งาน	ระบบ
	1. เลือกภาษาโปรแกรม 2. นำเข้าข้อมูลชอร์สโค้ด	3. ประมวลผลข้อมูล 4. แสดงหน้าจอผลลัพธ์ ของการตรวจสอบการคัดลอกชอร์สโค้ด
เงื่อนไขการทำงานพิเศษ	1. ผู้ใช้นำเข้าข้อมูลชอร์สโค้ดมากกว่า 1 ไฟล์ 2. ผู้ใช้นำเข้าข้อมูลชอร์สโค้ดที่ไม่ใช่ไฟล์ประเภท zip ไฟล์ 3. ระบบประมวลผลไม่ผ่าน	

### 3.3.2 มอดูลแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด (Output Plagiarism Module)

หลังจากผู้ใช้งานนำเข้าไฟล์ข้อมูลชอร์สโค้ดที่ต้องการตรวจสอบความคล้ายคลึงของชอร์สโค้ด ไปยังระบบแล้ว ระบบจะทำการประมวลและตรวจสอบความคล้ายคลึงของชอร์สโค้ด ที่อยู่ภายในไฟล์ของผู้ใช้งาน และแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด โดยมีรายละเอียด ดังตารางที่ 3-2

ตารางที่ 3-2 แสดงผลลัพธ์ของการตรวจสอบการคัดลอกซอฟต์แวร์สโค้ด

ชื่อยูสเซอร์: แสดงผลลัพธ์ของการตรวจสอบการคัดลอกซอฟต์แวร์สโค้ด	รหัส: UC-SEAP-02	ระดับความสำคัญ: สูง
ผู้กระทำหลัก: ผู้ใช้งาน	ระดับความซับซ้อน: สูง	
<b>ผู้มีส่วนเกี่ยวข้องและการใช้ประโยชน์:</b> ผู้ใช้งาน		
<b>คำอธิบาย:</b> ระบบทำการแสดงหน้าจอผลลัพธ์ โดยผู้ใช้งานสามารถทำการตรวจสอบผลลัพธ์ของการตรวจสอบการคัดลอกซอฟต์แวร์สโค้ดในรูปแบบต่างๆ ทั้งในรูปแบบของกราฟและรูปแบบของตาราง เพื่อให้เข้าใจผลลัพธ์ของข้อมูลได้โดยง่ายมากยิ่งขึ้น		
<b>สิ่งกระตุ้น:</b> ผู้ใช้งานต้องการตรวจสอบความคล้ายคลึงซอฟต์แวร์สโค้ด เพื่อให้เข้าใจข้อมูลได้ง่าย		
<b>ประเภทสิ่งกระตุ้น:</b> ภายนอก		
<b>ความสัมพันธ์:</b>  ความเกี่ยวเนื่อง: -  การรวม: -  การขยาย: -  การรับทอดคุณสมบัติ: -		
เงื่อนไขก่อนการทำงาน	ผู้ใช้งานนำเข้าข้อมูลซอฟต์แวร์สโค้ดเข้าสู่ระบบเพื่อตรวจสอบข้อมูล	
เงื่อนไขหลังการทำงาน	ผู้ใช้งานสามารถเห็นผลลัพธ์ของการตรวจสอบการคัดลอกซอฟต์แวร์สโค้ด และเข้าใจผลลัพธ์ได้ง่าย	

ตารางที่ 3-2 แสดงผลลัพธ์ของการตรวจสอบการคัดลอกซอฟต์แวร์ (ต่อ)

ชื่อยูสเซอร์: แสดงผลลัพธ์ของการตรวจสอบการคัดลอกซอฟต์แวร์	รหัส: UC-SEAP-02	ระดับความสำคัญ: สูง
ขั้นตอนการทำงานปกติ	ผู้ใช้งาน	ระบบ
		1. แสดงหน้าจอผลลัพธ์การตรวจสอบการคัดลอกซอฟต์แวร์ 2. เลือกรูปแบบการแสดงผลลัพธ์ที่ต้องการ 3. แสดงรูปแบบผลลัพธ์ที่ผู้ใช้เลือก
เงื่อนไขการทำงานพิเศษ	-	

### 3.3.3 มอดูลกำหนดขอบเขตการแสดงผลลัพธ์ (Scope Output Module)

การแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอฟต์แวร์ในรูปแบบของกราฟผู้ใช้งานสามารถระบุขอบเขตของร้อยละความคล้ายคลึงของซอฟต์แวร์ภายในขอบเขต 1 ถึง 100 ระบบจะแสดงผลลัพธ์ร้อยละความคล้ายคลึงของซอฟต์แวร์โดยเฉพาะขอบเขตที่ผู้ใช้งานระบุโดยมีรายละเอียด ดังตารางที่ 3-3

ตารางที่ 3-3 กำหนดขอบเขตของการแสดงผลลัพธ์

ชื่อยูสเซอร์: กำหนดขอบเขตของการแสดงผลลัพธ์	รหัส: UC-SEAP-02-2	ระดับความสำคัญ: ปานกลาง
ผู้กระทำหลัก: ผู้ใช้งาน	ระดับความซับซ้อน: ต่ำ	
ผู้ร่วมเกี่ยวข้องและการใช้ประโยชน์: ผู้ใช้งาน		
คำอธิบาย: ผู้ใช้งานทำการกำหนดขอบเขตของผู้ใช้ได้จากร้อยละความคล้ายคลึงซอฟต์แวร์ที่ต้องการ โดยจะกำหนดข้อมูลได้ตั้งแต่ร้อยละ 1 ถึง 100 ระบบจะแสดงข้อมูลผลลัพธ์ของการตรวจสอบความคล้ายคลึงของซอฟต์แวร์ที่อยู่ภายใต้ขอบเขตที่ผู้ใช้งานกำหนด		

ตารางที่ 3-3 กำหนดขอบเขตการแสดงผลลัพธ์ (ต่อ)

ชื่อข้อมูลสเคเดส: กำหนดขอบเขตของ การแสดงผลลัพธ์	รหัส: UC-SEAP-02-2	ระดับความสำคัญ: ปานกลาง
<b>สิ่งกระตุ้น:</b> ผู้ใช้งานต้องการตรวจสอบความคล้ายคลึงชอร์สโค้ดตามขอบเขตทั้งหมด ความคล้ายคลึงชอร์สโค้ด เพื่อตรวจสอบในชุดข้อมูลของขอบเขตที่ระบุไว้		
<b>ประเภทสิ่งกระตุ้น:</b> ภายนอก		
ความสัมพันธ์:  ความเกี่ยวเนื่อง: -  การรวม: -  การขยาย: -  การรับחוัดคุณสมบัติ: -		
เงื่อนไขก่อนการทำงาน	ผู้ใช้งานเข้าถึงหน้าจอแสดงผลลัพธ์ของการตรวจสอบการคัดลอกชอร์สโค้ด และต้องการกำหนดขอบเขตทั้งหมด ความคล้ายคลึงชอร์สโค้ด เพื่อตรวจสอบผลลัพธ์ตามขอบเขต	
เงื่อนไขหลังการทำงาน	ผู้ใช้งานสามารถเห็นผลลัพธ์ของการตรวจสอบการคัดลอกชอร์สโค้ดในขอบเขตของร้อยละที่ระบุไว้	
ขั้นตอนการทำงานปกติ	ผู้ใช้งาน	ระบบ
	1. กำหนดขอบเขตทั้งหมดที่ ต้องการ	2. แสดงผลลัพธ์ของการ ตรวจสอบการคัดลอกชอร์ส โค้ดในขอบเขตของร้อยละ ที่ระบุ
เงื่อนไขการทำงานพิเศษ	-	

### 3.3.4 module แสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน (Compare Source Code Module)

ผู้ใช้งานสามารถดูรายละเอียดของชอร์สโค้ดที่เหมือนกันได้ โดยเลือกที่คุ่ของผู้เป็นเจ้าของชอร์สโค้ดข้อมูล ระบบจะแสดงหน้าจอระหว่างผู้เป็นเจ้าของชอร์สโค้ด 2 คน และไฟล์ชอร์สโค้ดที่มีส่วนของชอร์สโค้ดที่เหมือนกันให้ผู้ใช้งานเลือกตู โดยมีรายละเอียด ดังตารางที่ 3-4

ตารางที่ 3-4 แสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน

ชื่อยูสเคส: แสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน	รหัส: UC-SEAP-03	ระดับความสำคัญ: สูง
ผู้กระทำหลัก: ผู้ใช้งาน	ระดับความซับซ้อน: สูง	
<b>ผู้มีส่วนเกี่ยวข้องและการใช้ประโยชน์:</b> ผู้ใช้งาน		
คำอธิบาย: ผู้ใช้ทำการเลือก เอจ (edge) หรือเส้นความเชื่อมโยงของข้อมูล 2 ชุด เพื่อไปสู่หน้าจอแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน ระบบจะทำการแสดงไฟล์ข้อมูลที่มีความคล้ายคลึงของชุดข้อมูลมาแสดง โดยผู้ใช้สามารถเลือกตรวจสอบไฟล์ต่างๆ ได้		
<b>สิ่งกระตุ้น:</b> ผู้ใช้ต้องการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน		
<b>ประเภทสิ่งกระตุ้น:</b> ภายนอก		
<b>ความสัมพันธ์:</b>  ความเกี่ยวเนื่อง: -  การรวม: -  การขยาย: -  การรับทอดคุณสมบัติ: -		
เงื่อนไขก่อนการทำงาน	ผู้ใช้งานต้องการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน เพื่อตรวจสอบความคล้ายคลึงชอร์สโค้ดของชุดข้อมูล 2 ชุด	
เงื่อนไขหลังการทำงาน	ผู้ใช้งานสามารถเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกันได้ เพื่อตรวจสอบตรวจสอบความคล้ายคลึงชอร์สโค้ดของชุดข้อมูล 2 ชุด และนำไปใช้งานต่อในอนาคต	

ตารางที่ 3-4 แสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน (ต่อ)

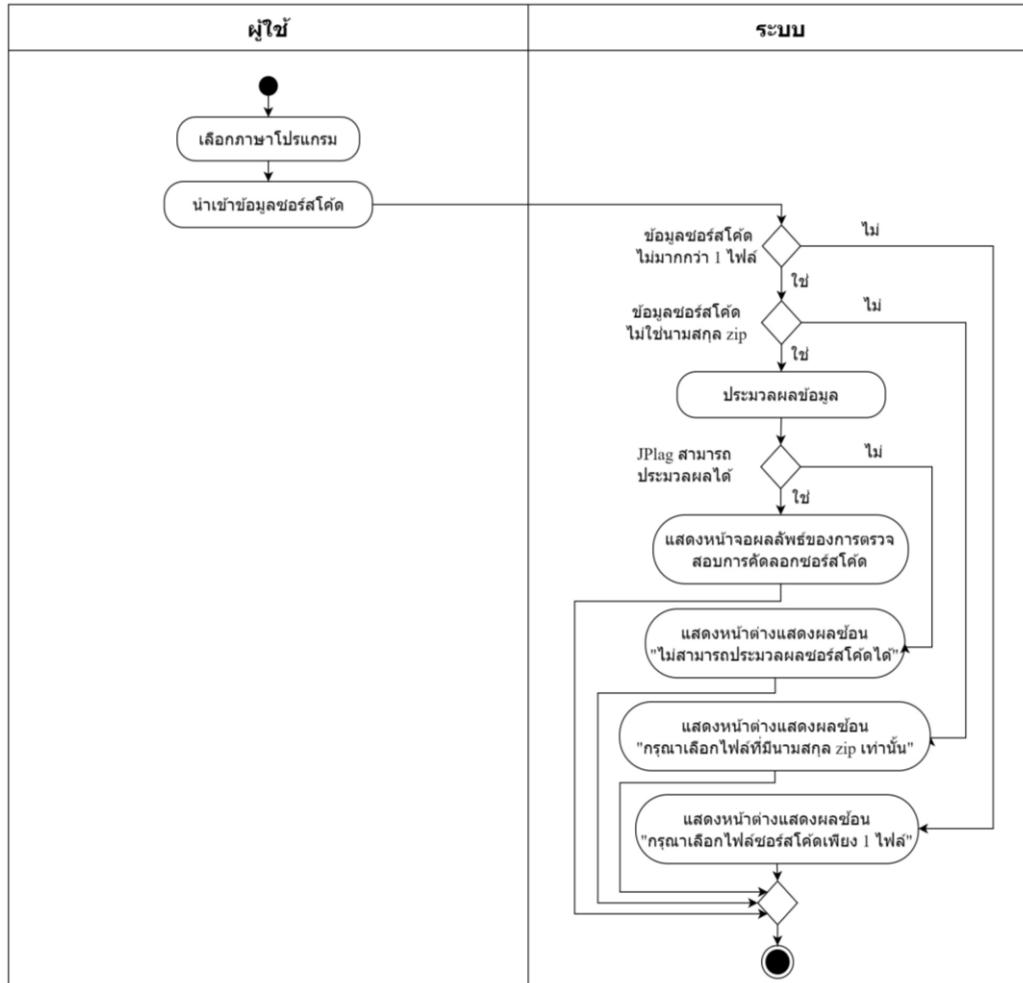
ชื่อยุสเคส: แสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน	รหัส: UC-SEAP-03	ระดับความสำคัญ: สูง
ขั้นตอนการทำงานปกติ	ผู้ใช้งาน	ระบบ
	1. เลือกเส้นความเชื่อมโยง ของข้อมูล 2 ชุด  3. ผู้ใช้เลือกไฟล์ที่ต้องการ ตรวจสอบ  4. แสดงไฟล์ชอร์สโค้ด ของข้อมูลที่ต้องการ	2. แสดงหน้าจอแสดงการ เปรียบเทียบชอร์สโค้ดในส่วน ที่เหมือนกัน
เงื่อนไขการทำงานพิเศษ	-	

### 3.4 แผนภาพกิจกรรม (Activity Diagram)

ในส่วนของการออกแบบแผนภาพกิจกรรมเป็นการอธิบายถึงลำดับขั้นตอนการทำงานของแต่ละมодูล โดยแสดงให้เห็นถึงการปฏิสัมพันธ์กันระหว่างผู้ใช้งานกับระบบแสดงผลลัพธ์ การตรวจสอบความคล้ายคลึงของชอร์สโค้ดอย่างเป็นลำดับขั้นตอน ซึ่งมีรายละเอียด ดังต่อไปนี้

#### 3.4.1 มอดูลนำเข้าข้อมูลชอร์สโค้ด (Import Source Code Module)

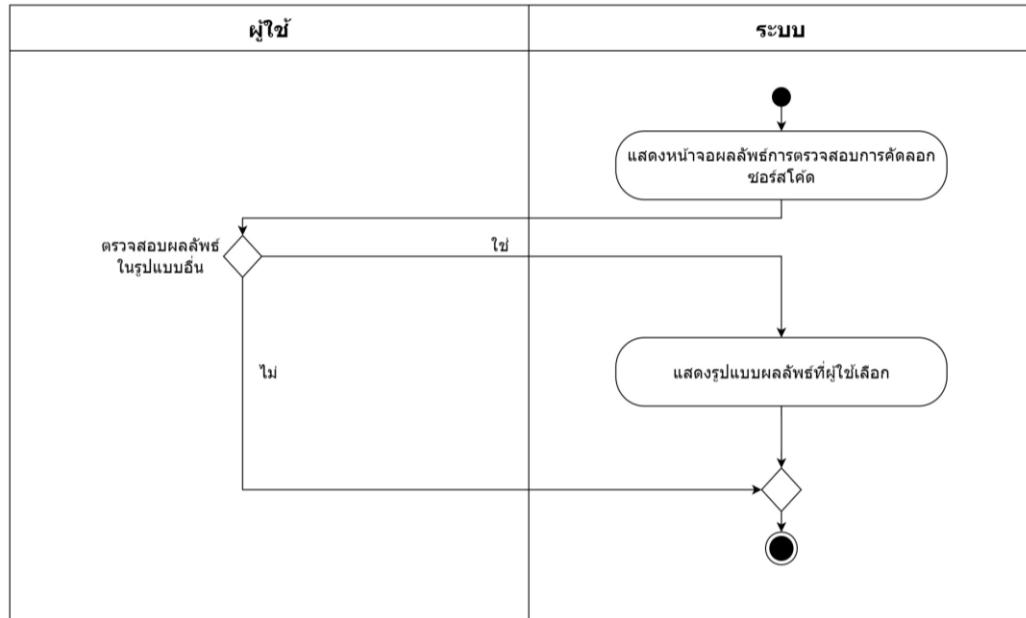
ผู้ใช้งานจัดเตรียมไฟล์ชอร์สโค้ดที่ต้องการตรวจสอบความคล้ายคลึงของชอร์สโค้ด จากนั้นเลือกภาษาและนำเข้าไฟล์ชอร์สโค้ดเข้าสู่ระบบ ซึ่งระบบได้กำหนดให้ผู้ใช้งานสามารถนำเข้าไฟล์ได้เพียง 1 ไฟล์และไฟล์นั้นต้องมีนามสกุล zip เท่านั้น โดยมีรายละเอียด ดังภาพที่ 3-3



ภาพที่ 3-3 แผนภาพกิจกรรมนำเข้าข้อมูลชอร์สโค้ด

### 3.4.2 มอดูลแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด (Output Plagiarism Module)

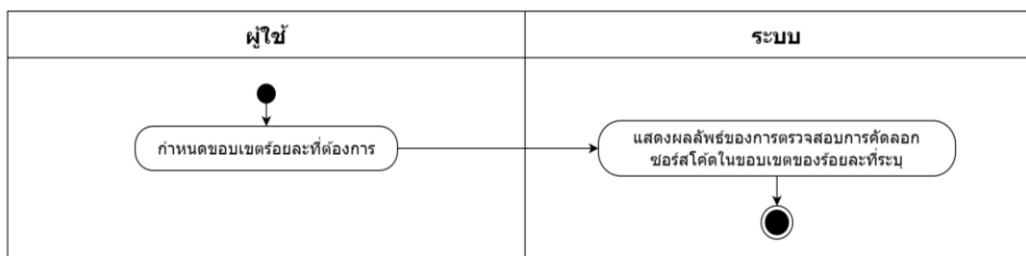
หลังจากที่ระบบได้รับไฟล์ข้อมูลชอร์สโค้ดแล้ว ระบบจะทำการตรวจสอบความคล้ายคลึงของชอร์สโค้ดภายในไฟล์โดยใช้ JPlag ในการตรวจสอบ จากนั้น ระบบจะแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ดภายในไฟล์ โดยมีรายละเอียด ดังภาพที่ 3-4



ภาพที่ 3-4 แผนภาพกิจกรรมแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

### 3.4.3 มอดูลกำหนดขอบเขตการแสดงผลลัพธ์ (Scope Output Module)

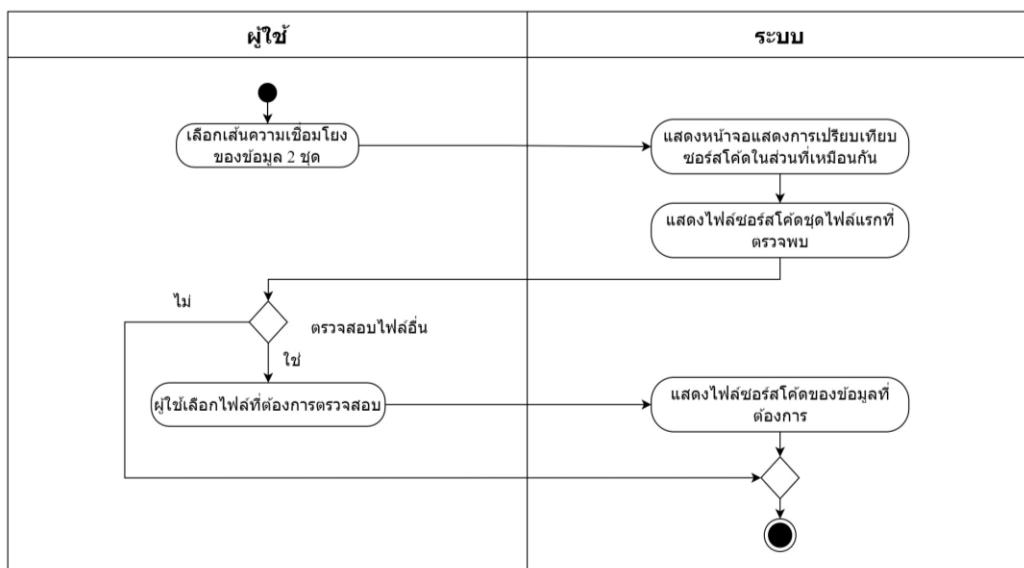
ในการแสดงผลลัพธ์ในรูปแบบของกราฟ ผู้ใช้งานสามารถกำหนดขอบเขตของร้อยละความคล้ายคลึงของชอร์สโค้ดที่อยู่ระหว่าง 1 ถึง 100 เมื่อผู้ใช้กำหนดขอบเขตแล้ว ระบบจะแสดงข้อมูลความคล้ายคลึงของชอร์สโค้ดตามภายในขอบเขตที่กำหนดไว้ โดยมีรายละเอียด ดังภาพที่ 3-5



ภาพที่ 3-5 แผนภาพกิจกรรมกำหนดขอบเขตการแสดงผลลัพธ์

### 3.4.4 模ดูลแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน (Compare Source Code Module)

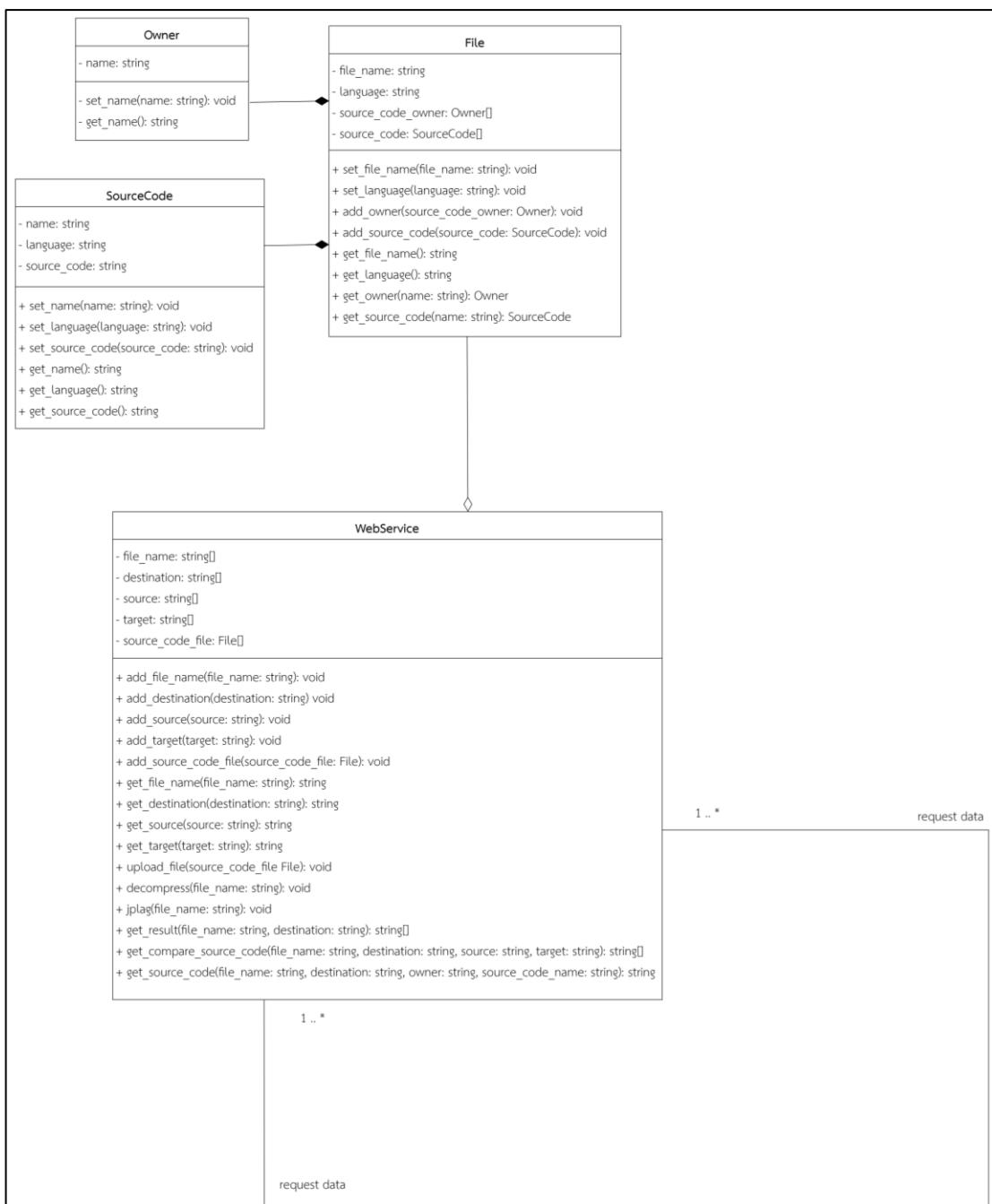
ระบบสามารถแสดงชอร์สโค้ดในส่วนที่เหมือนกันระหว่างผู้เป็นเจ้าของชอร์สโค้ดที่มีส่วนของข้อมูลชอร์สโค้ดที่เหมือนกัน โดยให้ผู้ใช้งานทำการเลือกคู่ชุดข้อมูลชอร์สโค้ดของผู้เป็นเจ้าของชอร์สโค้ด ระบบจะแสดงหน้าจอชอร์สโค้ดในส่วนที่เหมือนกัน โดยมีรายละเอียด ดังภาพที่ 3-6



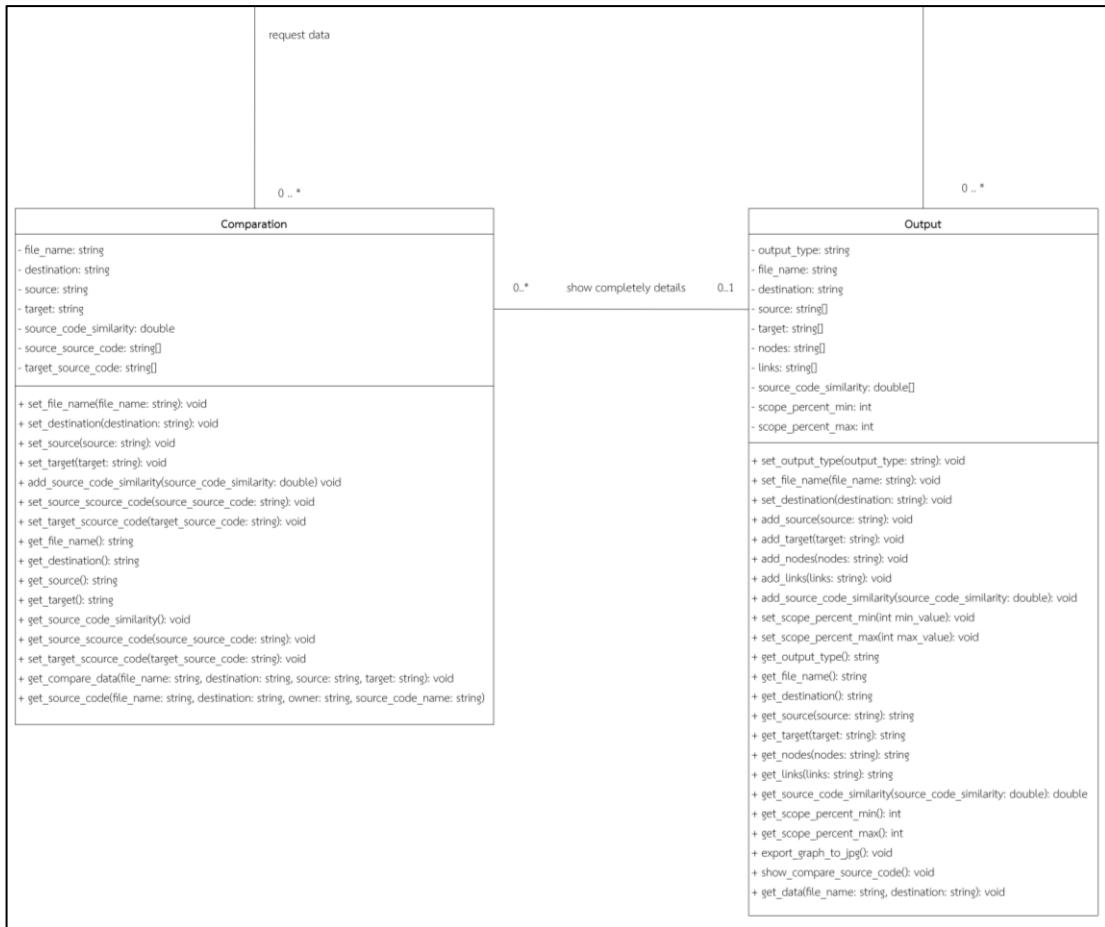
ภาพที่ 3-6 แผนภาพกิจกรรมแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน

### 3.5 แผนภาพคลาส (Class Diagram)

ผู้ดำเนินโครงการได้ทำการจำแนกองค์ประกอบหลักของระบบ ออกเป็นทั้งหมด 6 ส่วน ได้แก่ ส่วนของผู้เป็นเจ้าของชอร์สโค้ด ส่วนของชอร์สโค้ด ส่วนของไฟล์ที่รวมข้อมูลชอร์สโค้ด ส่วนบริการของระบบ ส่วนของการแสดงผลลัพธ์ความคล้ายคลึงของชอร์สโค้ด และส่วนของการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน จากนั้นจึงนำองค์ประกอบดังกล่าวบันทึกลงบนแผนภาพคลาสเพื่อใช้อธิบายถึงคุณลักษณะ (Attribute) และความสามารถ (Method) ของแต่ละองค์ประกอบโดยผู้ดำเนินโครงการได้ระบุความสัมพันธ์ระหว่างองค์ประกอบ (Relationship) ในแผนภาพโดยใช้เส้นความสัมพันธ์ ซึ่งมีรายละเอียด ดังภาพที่ 3-7



ภาพที่ 3-7 แผนภาพคลาส



ภาพที่ 3-7 แผนภูมิคลาส (ต่อ)

จากภาพที่ 3-7 แผนภูมิคลาส (Class Diagram) ของระบบ ประกอบไปด้วยคลาสทั้งหมด 6 คลาส ได้แก่ คลาสเจ้าของซอฟต์แวร์สโค็ด (Owner Class) คลาสซอฟต์แวร์สโค็ด (Source Code Class) คลาสไฟล์ข้อมูล (File Class) คลาสเว็บเซอร์วิส (Web Service Class) คลาสผลลัพธ์การตรวจสอบความคล้ายคลึง (Output Class) และคลาสการเปรียบเทียบซอฟต์แวร์สโค็ดในส่วนที่เหมือนกัน (Comparison Class) โดยแต่ละคลาสมีรายละเอียดของคุณสมบัติ (Attributes) และความสามารถ (Methods) ดังนี้

### 3.5.1 คลาสเจ้าของซอฟต์แวร์สโค็ด (Owner Class)

ชื่อโฟลเดอร์ซึ่งเป็นชื่อของเจ้าของชุดข้อมูลซอฟต์แวร์สโค็ดชุดนั้นๆ เพื่อสื่อสารกับระบบว่า ซอฟต์แวร์สโค็ดชุดที่กำลังประมวลผลอยู่นี้ มีใครเป็นเจ้าของ ซึ่ง Attribute หลักของคลาสนี้คือชื่อผู้เป็นเจ้าของซอฟต์แวร์สโค็ด (Name) และ Method สำหรับกำหนดชื่อผู้เป็นเจ้าของซอฟต์แวร์สโค็ด (Set Name)

### 3.5.2 คลาสชอร์สโค้ด (Source Code Class)

ไฟล์ชอร์สโค้ดซึ่งเป็นไฟล์ข้อมูลชอร์สโค้ดที่ใช้สำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ดซึ่ง Attributes ได้แก่ ชื่อไฟล์ (Name) ภาษาที่ใช้เขียน (Language) และชอร์สโค้ด (Source Code) ในส่วนของ Methods ประกอบไปด้วย กำหนดชื่อไฟล์ (Set Name) กำหนดภาษาที่ใช้เขียน (Set Language) และกำหนดชอร์สโค้ด (Set Source Code)

### 3.5.3 คลาสไฟล์ข้อมูล (File Class)

ไฟล์ข้อมูลที่ประกอบไปด้วยไฟล์ข้อมูลชอร์สโค้ดและผู้เป็นเจ้าของไฟล์ข้อมูลชอร์สโค้ดชุดนั้นๆ ซึ่งเป็นไฟล์สำหรับนำเข้าสู่ระบบ โดยมี Attributes ประกอบไปด้วย ชื่อไฟล์ (File Name) ภาษาของชอร์สโค้ด (Language) ผู้เจ้าของชอร์สโค้ด (Source Code Owner) และไฟล์ชอร์สโค้ด (Source Code) ในส่วนของ Methods ประกอบไปด้วย กำหนดชื่อไฟล์ (Set File Name) กำหนดภาษาของชอร์สโค้ด (Set Language) เพิ่มผู้เป็นเจ้าของชอร์สโค้ด (Add Source Code Owner) และเพิ่มไฟล์ชอร์สโค้ด (Add Source Code)

### 3.5.4 คลาสวีเบิร์วิส (Web Service Class)

เว็บเซอร์วิสที่ทำหน้าที่ในการรับไฟล์ข้อมูล (File) ซึ่งไฟล์ข้อมูลต้องผ่านกระบวนการบีบอัดไฟล์แล้วได้ไฟล์ที่มีนามสกุล zip เท่านั้น โดยภายในไฟล์ประกอบไปด้วยไฟล์ชอร์สโค้ด (Source Code) และผู้เป็นเจ้าของชอร์สโค้ดชุดนั้นๆ (Source Code Owner) เมื่อได้รับไฟล์ข้อมูลแล้ว เว็บเซอร์วิสจะทำการบันทึกไฟล์ข้อมูล แล้วทำการแตกไฟล์ข้อมูลที่มีนามสกุล zip นั้น จากนั้นจึงส่งการให้ JPlag ซึ่งเป็นไลบรารีสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ดโดยเฉพาะ ประมวลไฟล์ชอร์สโค้ดที่อยู่ภายในไฟล์ข้อมูลทั้งหมด เมื่อประมวลผลเสร็จแล้ว JPlag จะทำการสร้างไฟล์ข้อมูลผลลัพธ์ขึ้นที่เว็บเซอร์วิส จากนั้นเว็บเซอร์วิสจึงทำการส่งไฟล์ข้อมูลผลลัพธ์กลับไปยังไคลเอนต์ต้นทาง ซึ่งมี Attributes ประกอบไปด้วย ชื่อไฟล์ข้อมูลที่เกิดจากการสุม (File Name) ชื่อไฟล์ข้อมูลดั้งเดิม (Destination) ชื่อผู้เป็นเจ้าของชอร์สโค้ดที่เป็นต้นตอ (Source) ชื่อผู้เป็นเจ้าของชอร์สโค้ดที่เป็นตัวเปรียบเทียบ (Target) และไฟล์ข้อมูล (Source Code File) ในส่วนของ Methods ประกอบไปด้วย เพิ่มไฟล์ข้อมูลที่เกิดจากการสุม (Add File Name) เพิ่มไฟล์ข้อมูลดั้งเดิม (Add Destination) เพิ่มผู้เป็นเจ้าของชอร์สโค้ดที่เป็นต้นตอ (Add Source) เพิ่มผู้เป็นเจ้าของชอร์สโค้ดที่เป็นตัวเปรียบเทียบ (Add Target) เพิ่มไฟล์ข้อมูล (Add Source Code File) อัปโหลดไฟล์ข้อมูล (Upload File) แตกไฟล์ข้อมูล (Decompress) และประมวลผลไฟล์ข้อมูล (JPlag)

### 3.5.5 คลาสผลลัพธ์การตรวจสอบความคล้ายคลึง (Output Class)

การแสดงผลลัพธ์ในรูปแบบของกราฟความสัมพันธ์หรือตารางข้อมูล โดยใช้ข้อมูลผลลัพธ์ การตรวจสอบความคล้ายคลึงของชอร์สโค้ดของ JPlag ซึ่งเมื่อ JPlag ประมวลผลเสร็จแล้ว เว็บเซอร์วิสจะทำการส่งไฟล์ข้อมูลผลลัพธ์กลับไปยังโคลเลนต์ต้นทาง โดยคลาสๆ นี้เป็นคลาสสำหรับ แปลงข้อมูลผลลัพธ์ให้อยู่ในรูปแบบของกราฟความสัมพันธ์หรือตารางข้อมูล แล้วจึงแสดงข้อมูล ผลลัพธ์ที่ทำการแปลงแล้วนี้ แสดงให้กับผู้ใช้งานได้เห็น ซึ่งมี Attributes ประกอบไปด้วย ประเภท การแสดงผลลัพธ์ (Output Type) โหนด (Nodes) ความสัมพันธ์ (Links) ร้อยละความคล้ายคลึง ของชอร์สโค้ด (Source Code Similarity) ในส่วนของ Methods ประกอบไปด้วย กำหนดประเภท การแสดงผลลัพธ์ (Set Scope Output) เพิ่มโหนด (Add Nodes) เพิ่มความสัมพันธ์ (Add Links) เพิ่มร้อยละความคล้ายคลึงของชอร์สโค้ด (Add Source Code Similarity) และส่งออกภาพผลลัพธ์ การตรวจสอบความคล้ายคลึงของชอร์สโค้ด (Export Graph To JPG)

### 3.5.6 คลาสการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน (Comparison Class)

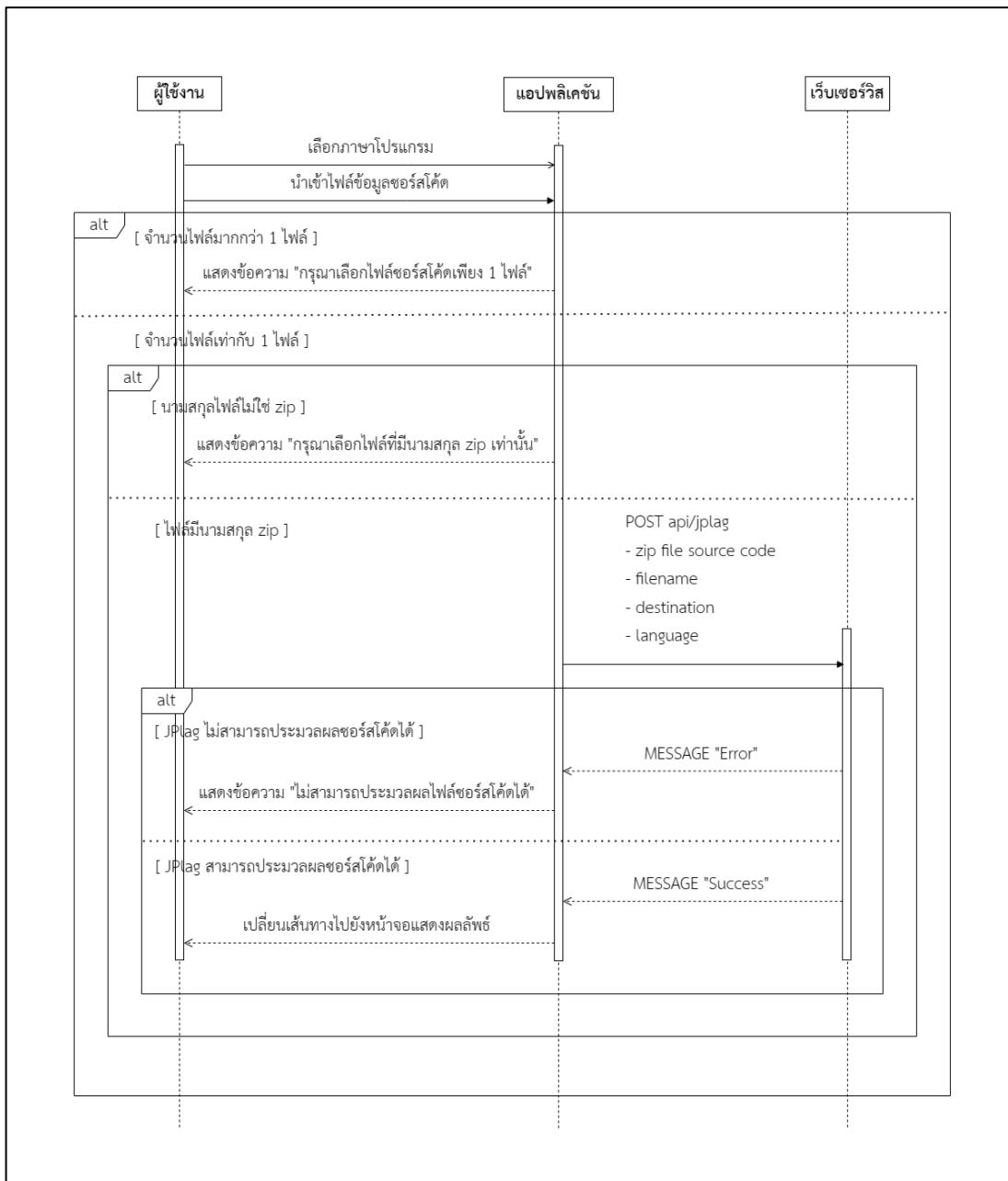
การแสดงข้อมูลการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน ซึ่งเป็นการแสดงชอร์สโค้ด ที่เหมือนกันของผู้เป็นเจ้าของชอร์สโค้ด 2 คน ซึ่งมี Attributes ประกอบไปด้วยชื่อผู้เป็นเจ้าของ ชอร์สโค้ดที่เป็นต้นตอ (Source) ชื่อผู้เป็นเจ้าของชอร์สโค้ดที่เป็นตัวเปรียบเทียบ (Target) ร้อยละ ความคล้ายคลึงของชอร์สโค้ด (Source Code Similarity) ไฟล์ชอร์สโค้ดของผู้เป็นเจ้าของ ชอร์สโค้ดที่ เป็นต้นตอ (Source Source Code) และไฟล์ชอร์สโค้ดผู้เป็นเจ้าของชอร์สโค้ด ที่เป็นตัวเปรียบเทียบ (Target Source Code)

## 3.6 แผนภาพลำดับการทำงาน (Sequence Diagram)

แผนภาพลำดับการทำงานใช้สำหรับอธิบายลำดับและขั้นตอนการทำงานของแต่ละโมดูล โดยมีรายละเอียดข้อมูลที่เป็นข้อมูลขาเข้า (Input data) และข้อมูลขาออก (Output data) เพื่อแสดงให้เห็นถึงลำดับการทำงานและลำดับการไหลของข้อมูล (Data flow) ในมอดูลนั้นๆ และมีการระบุเงื่อนไขในการทำงานภายในมอดูลต่างๆ โดยมีรายละเอียด ดังนี้

### 3.6.1 มอดูลนำเข้าข้อมูลชอร์สโค้ด (Import Source Code Module)

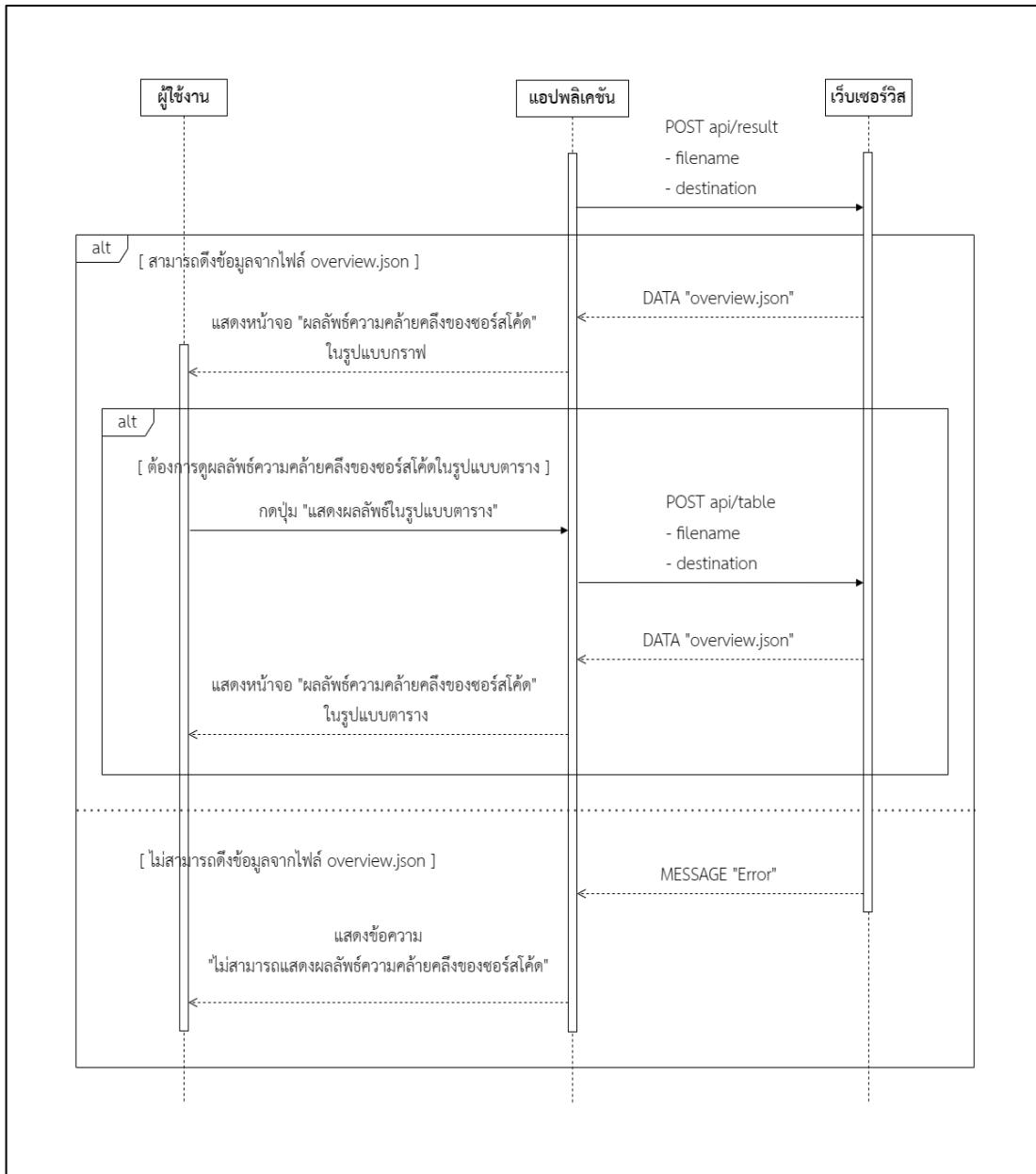
ผู้ใช้งานนำเข้าไฟล์ชอร์สโค้ดที่ต้องการตรวจสอบความคล้ายคลึง เมื่อระบบได้รับไฟล์ ชอร์สโค้ดแล้ว ระบบจะส่งไฟล์ชอร์สโค้ดไปยังเซิร์ฟเวอร์ที่เป็นเว็บเซอร์วิส เซิร์ฟเวอร์จะสั่งการให้ JPlag ประมวลผลชอร์สโค้ดในไฟล์ หาก JPlag สามารถประมวลได้ เซิร์ฟเวอร์จะตอบกลับระบบทว่า JPlag สามารถประมวลผลความคล้ายคลึงของชอร์สโค้ดในไฟล์ได้ ซึ่งมีรายละเอียด ดังภาพที่ 3-8



ภาพที่ 3-8 แผนภาพลำดับการทำงานนำเข้าข้อมูลของรีสโค้ด

### 3.6.2 มอดูลแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด (Output Plagiarism Module)

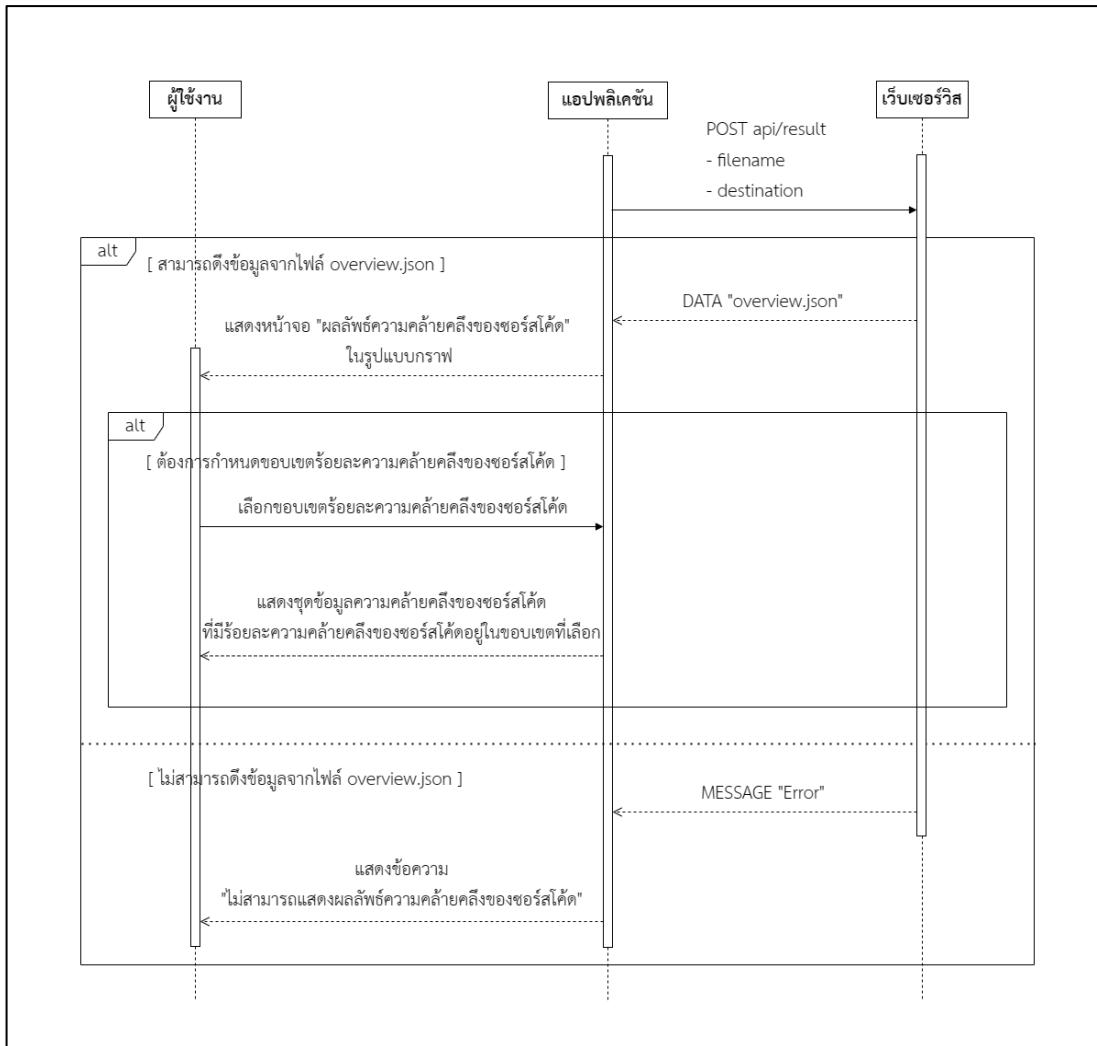
หลังจากที่ระบบได้รับการตอบกลับจากเซิร์ฟเวอร์ ระบบจะแสดงหน้าจอผลลัพธ์ การตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบกราฟ โดยติดต่อไปยังเซิร์ฟเวอร์เพื่อร้องขอ ข้อมูลผลลัพธ์จากการตรวจสอบความคล้ายคลึงของชอร์สโค้ด ซึ่งมีรายละเอียด ดังภาพที่ 3-9



ภาพที่ 3-9 แผนภาพลำดับการทำงานแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

### 3.6.3 มอduลกำหนดขอบเขตการแสดงผลลัพธ์ (Scope Output Module)

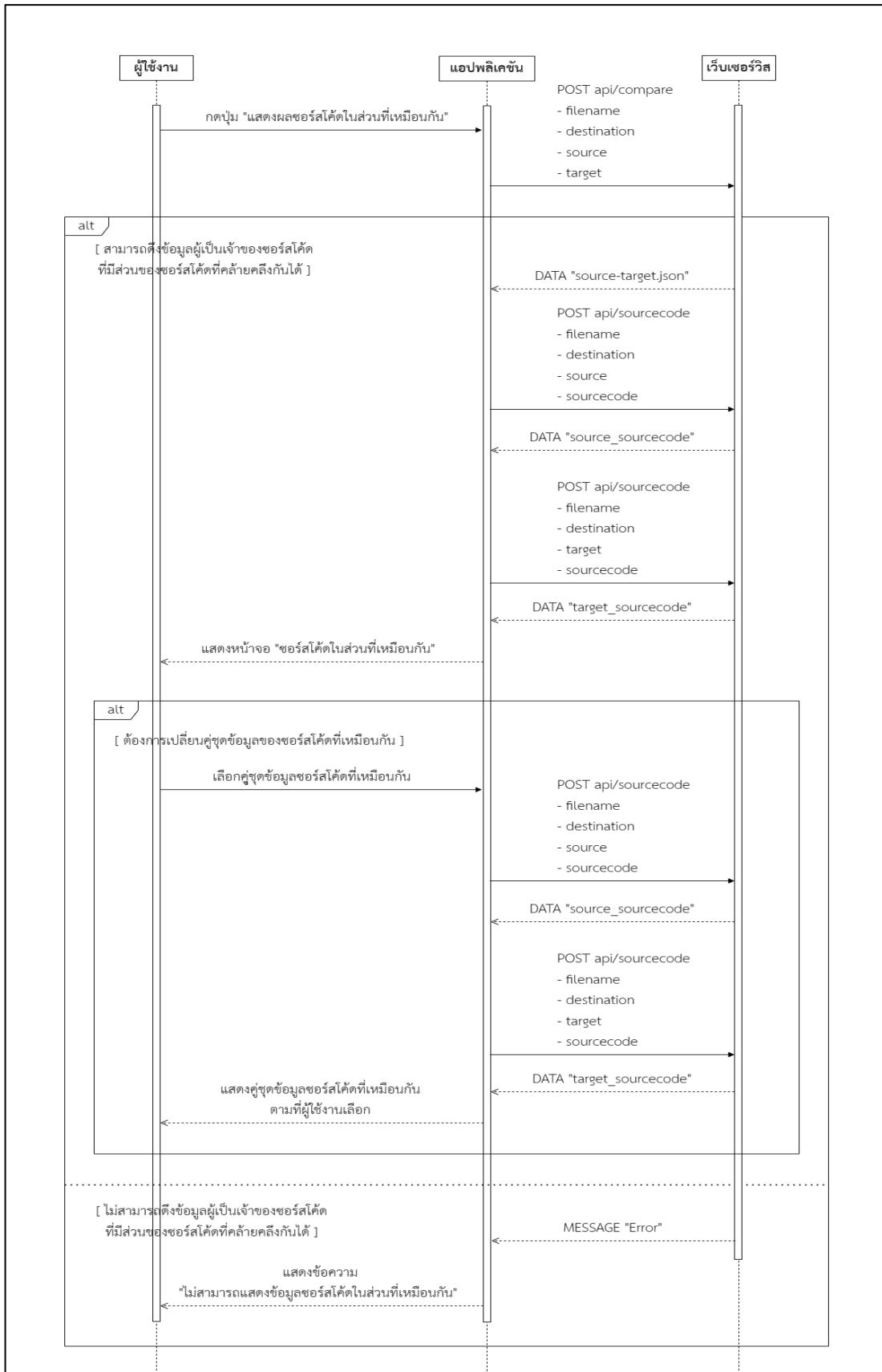
ผู้ใช้งานสามารถกำหนดขอบเขตของร้อยละความคล้ายคลึงของชอร์สโค้ดในรูปแบบ การแสดงผลแบบกราฟ ที่อยู่ในขอบเขต 1 ถึง 100 ระบบจะแสดงผลลัพธ์ความคล้ายคลึงของชอร์สโค้ด ที่ร้อยละความคล้ายคลึงของชอร์สโค้ดอยู่ในขอบเขตที่กำหนด ซึ่งมีรายละเอียด ดังภาพที่ 3-10



ภาพที่ 3-10 แผนภาพลำดับการทำงานกำหนดขอบเขตการแสดงผลลัพธ์

### 3.6.4 มอดูลแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน (Compare Source Code Module)

มอดูลสำหรับแสดงผลชอร์สโค้ดในส่วนที่เหมือนกัน โดยผู้ใช้งานทำการเลือกดูส่วนของชอร์สโค้ดที่เหมือนกันระหว่างผู้เป็นเจ้าของชอร์สโค้ดที่ระบบตรวจสอบพบว่ามีส่วนของชอร์สโค้ดที่เหมือนกัน ระบบจะแสดงหน้าจอชอร์สโค้ดที่เหมือนกันให้ผู้ใช้งานเลือกดูซึ่งมีรายละเอียด ดังภาพที่ 3-11



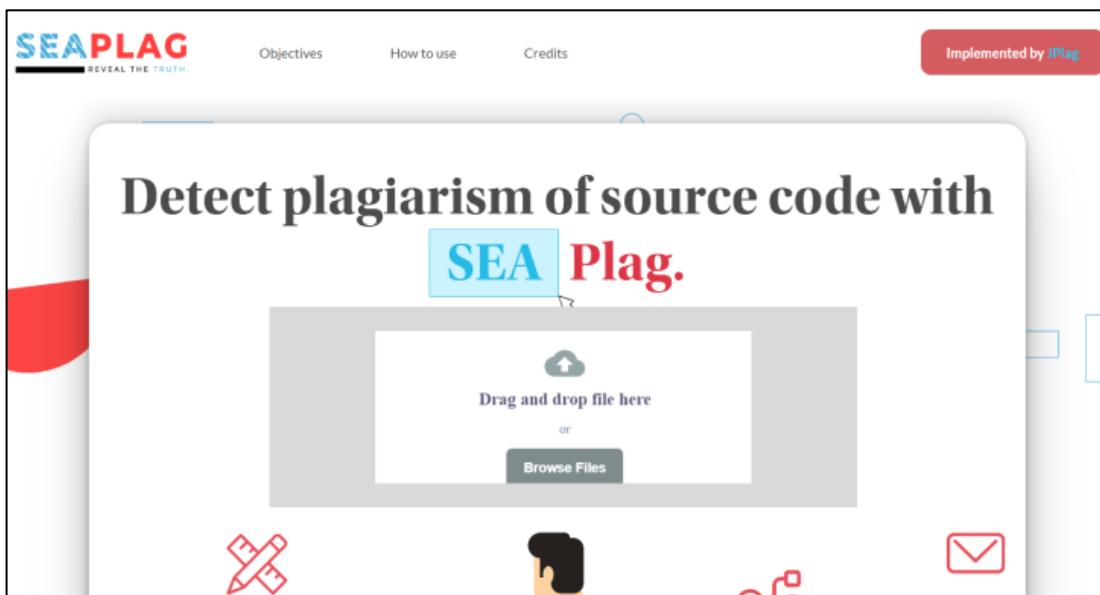
ภาพที่ 3-11 แผนภาพลำดับการทำงานแสดงการเปรียบเทียบชุดข้อมูลรีสโค็ตในส่วนที่เหมือนกัน

### 3.7 การอุกแบบหน้าจอแสดงผล

หลังจากวิเคราะห์และอุกแบบโครงสร้างการทำงานของระบบเสร็จสิ้นแล้ว ผู้ดำเนินโครงการจึงได้ทำการอุกแบบหน้าจอแสดงผลของระบบ โดยได้อุกแบบหน้าจอแยกตามแต่ละมอดูลจำนวนทั้งสิ้น 4 มอดูล โดยมีรายละเอียด ดังนี้

#### 3.7.1 มอดูลนำเข้าข้อมูลชอร์สโค้ด (Import Source Code Module)

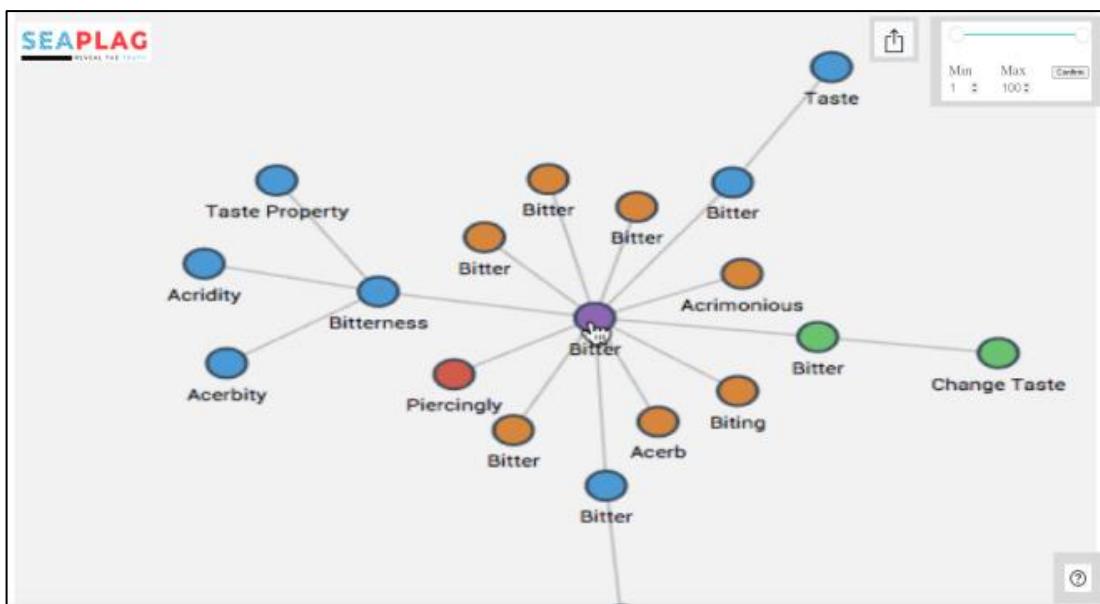
หน้าจอแสดงผลสำหรับให้ผู้ใช้งานนำเข้าข้อมูลไฟล์ชอร์สโค้ด โดยผู้ใช้งานสามารถเลือกไฟล์ชอร์สโค้ดหรือลากไฟล์ชอร์สโค้ดมาวางที่กล่องสำหรับนำเข้าไฟล์ชอร์สโค้ด โดยผู้ใช้งานสามารถเลือกไฟล์ชอร์สโค้ดได้เพียง 1 ไฟล์ และไฟล์ต้องมีนามสกุล zip เท่านั้น ซึ่งมีรายละเอียด ดังภาพที่ 3-12



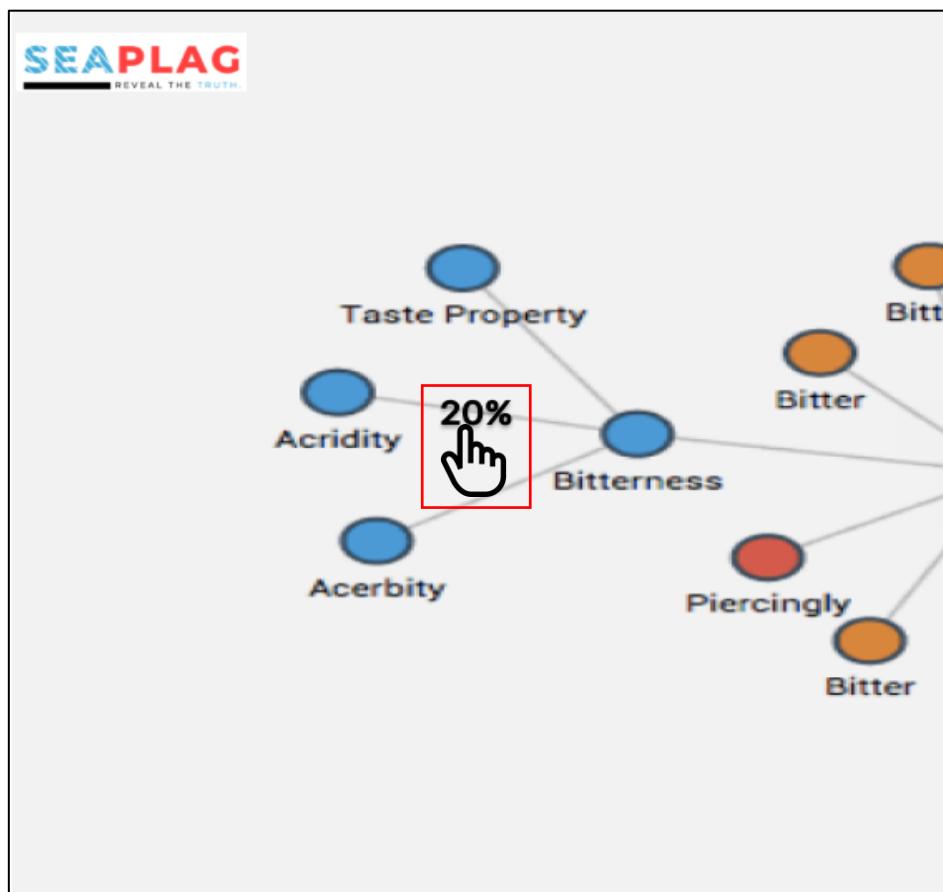
ภาพที่ 3-12 หน้าจอนำเข้าข้อมูลชอร์สโค้ด

#### 3.7.2 มอดูลแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด (Output Plagiarism Module)

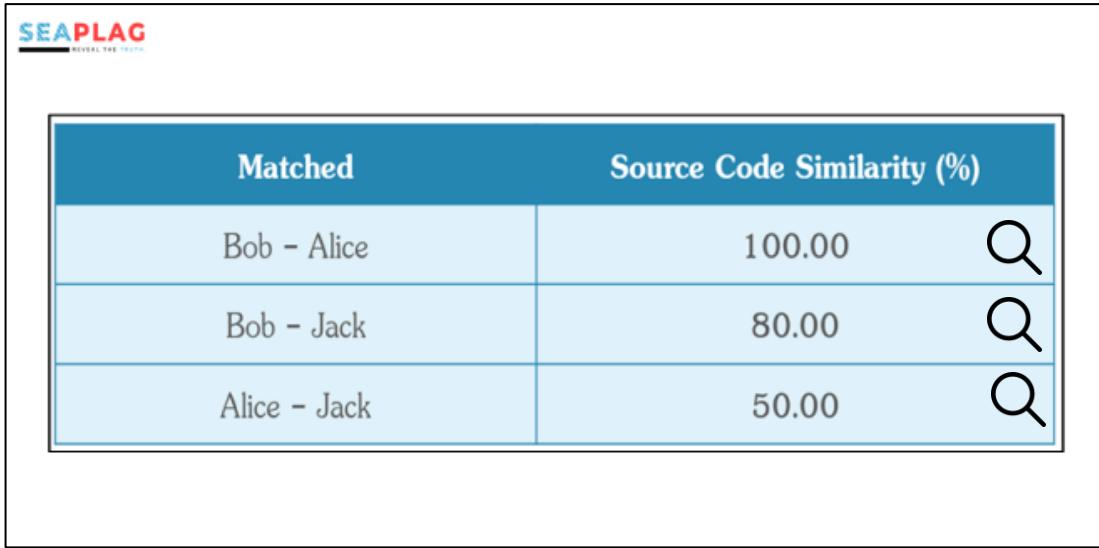
ส่วนของการอุกแบบหน้าจอสำหรับแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบกราฟหรือตาราง โดยระบบจะแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบของกราฟเป็นค่าเริ่มต้น ซึ่งมีรายละเอียด ดังภาพที่ 3-13 ภาพที่ 3-14 และภาพที่ 3-15



ภาพที่ 3-13 หน้าจอผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบกราฟ



ภาพที่ 3-14 การแสดงร้อยละความคล้ายคลึงของชอร์สโค้ดในรูปแบบกราฟ



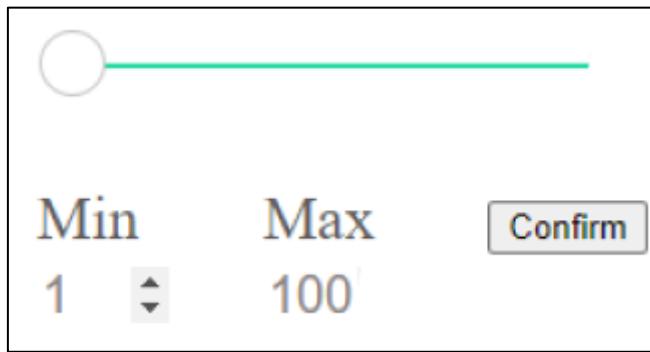
The screenshot shows a software interface with the logo "SEAPLAG" at the top left. Below it is a table titled "Matched" and "Source Code Similarity (%)".

Matched	Source Code Similarity (%)	
Bob - Alice	100.00	
Bob - Jack	80.00	
Alice - Jack	50.00	

ภาพที่ 3-15 หน้าจอผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบตาราง

### 3.7.3 มอดูลกำหนดขอบเขตการแสดงผลลัพธ์ (Scope Output Module)

องค์ประกอบของการกำหนดขอบเขตหรือร้อยละความคล้ายคลึงของชอร์สโค้ด โดยผู้ใช้งานสามารถเลือกได้เฉพาะค่าต่ำสุดของร้อยละความคล้ายคลึงของชอร์สโค้ดเท่านั้น โดยค่าสูงสุดของร้อยละความคล้ายคลึงของชอร์สโค้ดจะถูกกำหนดคงที่ไว้ที่ 100 ซึ่งมีรายละเอียด ดังภาพที่ 3-16



The screenshot shows a search interface with a magnifying glass icon at the top left. Below it is a horizontal slider with a green track and a white circular handle. Below the slider are two input fields labeled "Min" and "Max". The "Min" field contains the value "1" with up and down arrows. The "Max" field contains the value "100". To the right of the input fields is a "Confirm" button.

ภาพที่ 3-16 กล่องสำหรับกำหนดขอบเขตการแสดงผลลัพธ์

### 3.7.4 មօទូលសេដករាប្រើបាយបើយចន្ទកុំគិតិកុំណូនកុំណូន (Compare Source Code Module)

SEAPLAG  
BETTER YOUR CODE

Implemented by iFlag

# Similarity: 70.00 %

## Human.java

```
ople.data.users();
e = client.api.statuses.user_timeline.get(screen_
Get, lantime,response,data, 'tweets from', lscreen
response,data) != 0;
ate = response.data[0]['created_at'];
ate2 = datetime.strptime(ate,"%a %b %d %H:%M%
ay = datetime.now();
long = (today -ldate2).days
howlong < daywindow;
print lscreen_name, 'has tweeted in the past' ,
totaltweets == len(response.data)
ople.data.users();
e = client.api.statuses.user_timeline.get(screen_
Get, lantime,response,data, 'tweets from', lscreen
response,data) != 0;
ate = response.data[0]['created_at'];
ate2 = datetime.strptime(ate,"%a %b %d %H:%M%
ay = datetime.now();
long = (today -ldate2).days
howlong < daywindow;
print lscreen_name, 'has tweeted in the past' ,
totaltweets == len(response.data)
for j in response.data:
    if j.entities.urls:
        for k in j.entities.urls:
            newurl = k['expanded_url']
            urlset.add(newurl, j.user.screen_na
```

## Student.java

```
ople.data.users();
e = client.api.statuses.user_timeline.get(screen_
Get, lantime,response,data, 'tweets from', lscreen
response,data) != 0;
ate = response.data[0]['created_at'];
ate2 = datetime.strptime(ate,"%a %b %d %H:%M%
ay = datetime.now();
long = (today -ldate2).days
howlong < daywindow;
print lscreen_name, 'has tweeted in the past' ,
totaltweets == len(response.data)
ople.data.users();
e = client.api.statuses.user_timeline.get(screen_
Get, lantime,response,data, 'tweets from', lscreen
response,data) != 0;
ate = response.data[0]['created_at'];
ate2 = datetime.strptime(ate,"%a %b %d %H:%M%
ay = datetime.now();
long = (today -ldate2).days
howlong < daywindow;
print lscreen_name, 'has tweeted in the past' ,
totaltweets == len(response.data)
for j in response.data:
    if j.entities.urls:
        for k in j.entities.urls:
            newurl = k['expanded_url']
            urlset.add(newurl, j.user.screen_na
```

listness

Taste Property

Human.java

Student.java

Main.java

Main.java

ภาพที่ 3-17 หน้าจอการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน

จากภาพที่ 3-17 เมื่อผู้ใช้งานกดเลือกตัวส่วนของชอร์สโค้ดที่เหมือนกันระหว่างผู้เป็นเจ้าของข้อมูลชอร์สโค้ด ระบบจะแสดงหน้าจอเปรียบเทียบชอร์สโค้ด ให้ผู้ใช้งานสามารถดูรายละเอียดของชอร์สโค้ดที่เหมือนกัน ซึ่งองค์ประกอบของหน้าจอมีรายละเอียด ดังนี้

### 1) ร้อยละความคล้ายคลึงของชอร์สโค้ด (Source Code Similarity)

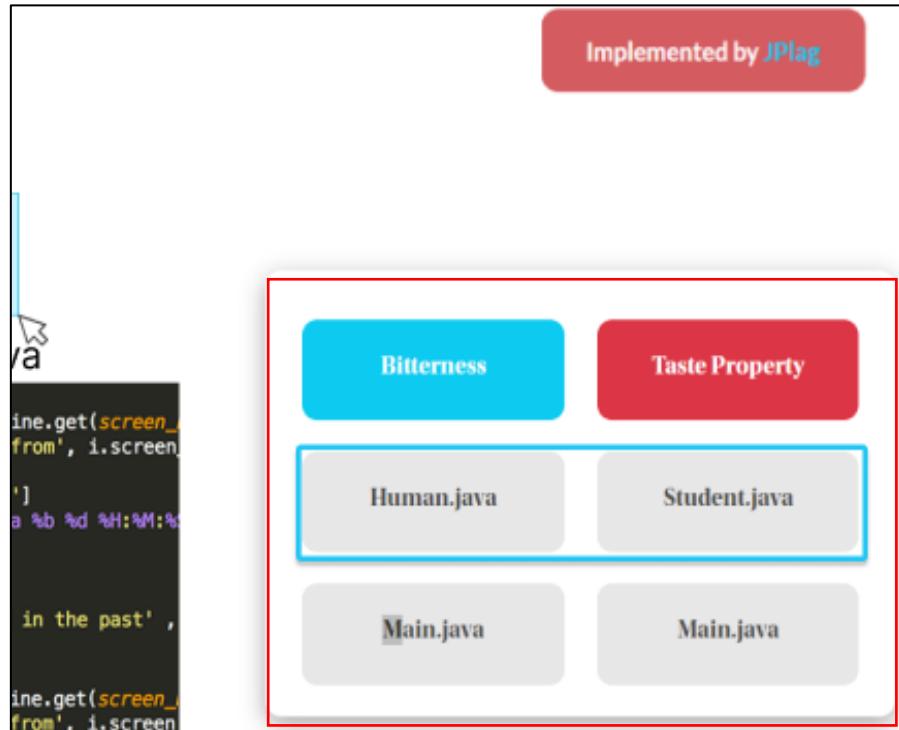
การแสดงร้อยละความคล้ายคลึงของชอร์สโค้ด เพื่อบ่งบอกผู้ใช้งานว่าข้อมูลชอร์สโค้ดที่ผู้ใช้งานเลือกนั้น มีความเหมือนกันอยู่มากน้อยเพียงใด โดยการแสดงผลร้อยละความคล้ายคลึงกันของชอร์สโค้ดนี้เป็นข้อมูลระหว่างผู้เป็นเจ้าของชอร์สโค้ดเท่านั้น ซึ่งมีรายละเอียด ดังภาพที่ 3-18

The image shows a screenshot of a plagiarism detection software interface. At the top left is the logo 'SEAPLAG' with the tagline 'REVEAL THE TRUTH.' Below the logo is a large red rectangular box containing the text 'Similarity: 70.00 %'. To the right of this box is a light blue rectangular area containing the file names 'Human.java' and 'Student.java'. A mouse cursor is visible over the 'Student.java' file name. The background of the interface is white.

ภาพที่ 3-18 ส่วนของการแสดงร้อยละความคล้ายคลึงของอรรถโคดี

## 2) ชุดข้อมูลชอร์สโค้ดที่เหมือนกันของผู้เป็นเจ้าของชอร์สโค้ด

ส่วนของการเลือกตุรายละเอียดชุดข้อมูลชอร์สโค้ดที่เหมือนกันและข้อมูลของผู้เป็นเจ้าของชอร์สโค้ดชุดนั้นๆ ผู้ใช้งานสามารถเลือกตุชอร์สโค้ดที่ต้องการได้ โดยกดเลือกที่ชื่อไฟล์ชอร์สโค้ดที่ต้องการดู ซึ่งมีรายละเอียด ดังภาพที่ 3-19



ภาพที่ 3-19 ส่วนของการแสดงชุดข้อมูลชอร์สโค้ดที่เหมือนกันของผู้เป็นเจ้าของชอร์สโค้ด

## 3) การแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน

กล่องสำหรับแสดงผลชอร์สโค้ดในส่วนที่เหมือนกัน โดยระบบจะอ่านไฟล์ข้อมูลชอร์สโค้ดที่เหมือนกันจำนวน 2 ไฟล์ เมื่ออ่านไฟล์ชอร์สโค้ดเสร็จแล้ว ระบบจะแสดงชอร์สโค้ดที่อ่านได้ในรูปแบบของกล่องข้อความ กล่องข้อความสามารถยืดขยายขนาดได้ตามความเหมาะสมของผู้ใช้งาน ดังภาพที่ 3-20

The screenshot shows the JPlag interface with two Java files side-by-side:

```

Human.java
ople.data.users:
e = client.api.statuses.user_timeline.get(screen_
'Got', len(response.data), 'tweets from', i.screen_
response.data >= 0;
ate = response.data[0]['created_at'];
ate2 = datetime.strptime(ltdate, '%a %b %d %H:%M:%S'
ay = datetime.now());
long = (today-ltdate).days;
howlong < daywindow;
print i.screen_name, 'has tweeted in the past',
totaltweets += len(response.data);
ople.data.users:
e = client.api.statuses.user_timeline.get(screen_
'Got', len(response.data), 'tweets from', i.screen_
response.data >= 0;
ate = response.data[0]['created_at'];
ate2 = datetime.strptime(ltdate, '%a %b %d %H:%M:%S'
ay = datetime.now());
long = (today-ltdate).days;
howlong < daywindow;
print i.screen_name, 'has tweeted in the past',
totaltweets += len(response.data)
for j in response.data:
    if j.entities.urls:
        for k in j.entities.urls:
            newurl = k['expanded_url']
            uriset.add(newurl, j.user.screen_name)

Student.java
ople.data.users:
e = client.api.statuses.user_timeline.get(screen_
'Got', len(response.data), 'tweets from', i.screen_
response.data >= 0;
ate = response.data[0]['created_at'];
ate2 = datetime.strptime(ltdate, '%a %b %d %H:%M:%S'
ay = datetime.now());
long = (today-ltdate).days;
howlong < daywindow;
print i.screen_name, 'has tweeted in the past',
totaltweets += len(response.data);
ople.data.users:
e = client.api.statuses.user_timeline.get(screen_
'Got', len(response.data), 'tweets from', i.screen_
response.data >= 0;
ate = response.data[0]['created_at'];
ate2 = datetime.strptime(ltdate, '%a %b %d %H:%M:%S'
ay = datetime.now());
long = (today-ltdate).days;
howlong < daywindow;
print i.screen_name, 'has tweeted in the past',
totaltweets += len(response.data)
for j in response.data:
    if j.entities.urls:
        for k in j.entities.urls:
            newurl = k['expanded_url']
            uriset.add(newurl, j.user.screen_name)

```

The interface includes a "Similarity: 70.00%" indicator at the top, and a sidebar with tabs for "Biterness" and "Taste Property". Below the tabs are four buttons: "Human.java", "Student.java", "Main.java", and "Main.java". A red button labeled "Implemented by JPlag" is located in the top right corner.

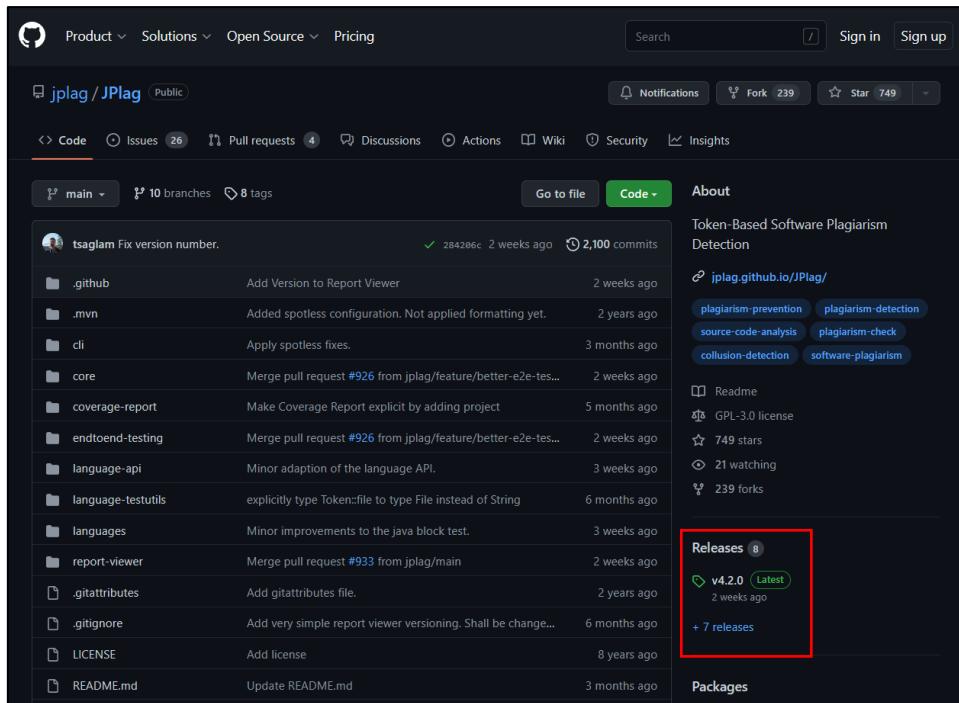
ภาพที่ 3-20 ส่วนของการแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน

### 3.8 การทำงานของ JPlag

JPlag เป็นเครื่องมือสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ดที่ผู้ดำเนินโครงการเลือกใช้ เนื่องจาก JPlag รองรับภาษาโปรแกรมจำนวนมากและเป็นเครื่องมือประเภท Open Source ซึ่งสามารถใช้งานได้โดยไม่มีค่าใช้จ่ายเพิ่มเติม โดยสามารถเข้าถึงชอร์สโค้ดของ JPlag ผ่านทาง GitHub ของ JPlag ได้โดยตรง

#### 3.8.1 วิธีการติดตั้งและใช้งาน JPlag

เมื่อเข้าถึง GitHub ของ JPlag และ จะปรากฏข้อมูลของ JPlag ประกอบไปด้วยชอร์สโค้ดของ JPlag วิธีการติดตั้ง (Installation) วิธีการใช้งาน (Building) ภาษาโปรแกรมที่รองรับ (Supported Languages) โดยเลือกที่เมนู Releases จะแสดงหน้าจอของ JPlag เวอร์ชันต่างๆที่เปิดให้ใช้งาน ซึ่งเวอร์ชันของ JPlag ที่ระบบใช้งานอยู่ในปัจจุบันนี้คือ เวอร์ชัน 4.1.0 ซึ่งมีรายละเอียดดังภาพที่ 3-21 และภาพที่ 3-22



ภาพที่ 3-21 เมนู Releases ใน GitHub ของ JPlag

ภาพที่ 3-22 รายการเวอร์ชันต่างๆ ของ JPlag

จากภาพที่ 3-22 ทำการดาวน์โหลดไฟล์ข้อมูลชื่อ jplag-4.1.0-jar-with-dependencies.jar ซึ่งเป็นไฟล์สำหรับประมวลผลความคล้ายคลึงของชอร์สโค้ด เมื่อดาวน์โหลดเสร็จแล้ว ในส่วนของวิธีการใช้งานของ JPlag ผ่าน Command Line Interface (CMD) มีรายละเอียดดังนี้

1) `java -jar ./jplag-4.1.0-jar-with-dependencies.jar`

คำสั่งสำหรับสั่งการให้ JPlag ประมวลผลไฟล์ชอร์สโค้ด ซึ่ง JPlag เป็นไลบรารีที่ถูกเขียนขึ้นโดยภาษา Java สำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ดโดยเฉพาะ ดังนั้น คำสั่งนี้จึงเป็นคำสั่งเริ่มต้น ซึ่งกำลังจะสั่งการให้ JPlag ประมวลผลไฟล์ชอร์สโค้ด

2) `-l language`

คำสั่งสำหรับเลือกภาษาโปรแกรมของชอร์สโค้ด หากไม่ทำการระบุ JPlag จะกำหนดค่าเริ่มต้นไว้เป็นภาษา Java ซึ่งสามารถกำหนดภาษาได้โดยใช้คำสั่ง `-l` ตามด้วยภาษาของโปรแกรม โดยสามารถเลือกภาษาโปรแกรมที่ JPlag รองรับได้ ดังภาพที่ 3-23

Language	Version	CLI Argument Name	state	parser
Java	17	java	mature	JavaC
C++	11	cpp	legacy	JavaCC
C#	8	csharp	beta	ANTLR 4
Go	1.17	golang	beta	ANTLR 4
Kotlin	1.3	kotlin	beta	ANTLR 4
Python	3.6	python3	legacy	ANTLR 4
R	3.5.0	rlang	beta	ANTLR 4
Rust	1.60.0	rust	beta	ANTLR 4
Scala	2.13.8	scala	beta	Scalameta
Scheme	?	scheme	unknown	JavaCC
Swift	5.4	swift	beta	ANTLR 4
EMF Metamodel	2.25.0	emf	alpha	EMF
Text (naive)	-	text	legacy	CoreNLP

ภาพที่ 3-23 ตัวเลือกภาษาโปรแกรมที่ JPlag รองรับ

### 3) -r path

คำสั่งกำหนดตำแหน่งวางไฟล์ผลลัพธ์จากการประมวลผลของ JPlag ซึ่งไฟล์ผลลัพธ์ของ JPlag จะอยู่ในรูปแบบของ zip ไฟล์ ซึ่งคำสั่งนี้เป็นการระบุชื่อไฟล์ผลลัพธ์และตำแหน่งวางไฟล์ผลลัพธ์ของ JPlag เมื่อ JPlag ประมวลผลเสร็จแล้ว JPlag วางไฟล์ผลลัพธ์ ณ ตำแหน่งที่ระบุไว้

### 4) -new path

คำสั่งเลือกตำแหน่งวางไฟล์ซึ่งภายในประกอบไปด้วยชอร์สโค้ดที่ต้องการตรวจสอบความคล้ายคลึง โดยภายในไฟล์จะประกอบไปด้วยไฟล์เดอร์หรือไดเรกทอรีอย่างผู้เป็นเจ้าของชอร์สโค้ด แต่ละคน ซึ่งภายในไฟล์เดอร์หรือไดเรกทอรีอยู่นั้น จะประกอบไปด้วยไฟล์ชอร์สโค้ด

## 3.8.2 ผลลัพธ์ของ JPlag

หลังจาก JPlag ประมวลผลไฟล์ชอร์สโค้ดเสร็จ JPlag จะทำการสร้าง zip ไฟล์ผลลัพธ์ขึ้น โดยเมื่อแตกไฟล์ผลลัพธ์ออกแล้ว ภายในไฟล์จะประกอบไปด้วยข้อมูล 3 รูปแบบ มีรายละเอียดดังนี้

### 1) ไฟล์ overview.json

ข้อมูลรวมของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด ซึ่งภายในมีข้อมูลสำหรับสร้างกราฟความสัมพันธ์ โดยค้นหาคีย์ที่ซื้อ metrics และจัดเรียงคีย์ ซื้อ topComparisons ซึ่งภายในจะประกอบไปด้วยคีย์จำนวน 3 คีย์ ได้แก่ first\_submission และ second\_submission โดยกำหนดให้ทั้ง 2 ข้อมูลนี้เป็น Nodes ของกราฟความสัมพันธ์ ในส่วนคีย์ที่ซื้อ similarity หมายถึง ร้อยละความคลึงของชอร์สโค้ด กำหนดให้แทนความสัมพันธ์ระหว่าง Nodes บนกราฟความสัมพันธ์

### 2) ไฟล์ผู้เป็นเจ้าของชอร์สโค้ดลำดับที่ n - ผู้เป็นเจ้าของชอร์สโค้ดลำดับที่ m.json

ข้อมูลที่เป็นรายละเอียดของความคล้ายคลึงกันระหว่างผู้เป็นเจ้าของข้อมูลชอร์สโค้ด 2 คน ซึ่งข้อมูลจะประกอบไปด้วย ผู้เป็นเจ้าของข้อมูลชอร์สโค้ดทั้ง 2 คนที่เปรียบเทียบกัน ร้อยละความคลึงของชอร์สโค้ด (similarity) รวมถึงซื้อของไฟล์ชอร์สโค้ดและบรรทัดของไฟล์ที่คล้ายคลึงกัน

### 3) ไฟล์เดอร์ submissions

ข้อมูลชอร์สโค้ดของผู้เป็นเจ้าของชอร์สโค้ดแต่ละคน ใช้สำหรับแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกันระหว่างผู้เป็นเจ้าของชอร์สโค้ด 2 คน ซึ่งระบบจะทำการแสดงการเปรียบเทียบชอร์สโค้ดที่เหมือนกันแบบไฟล์ต่อไฟล์ อ้างอิงตามผลลัพธ์ของ JPlag

### 3.9 การออกแบบเว็บเซอร์วิส (Web Service)

การออกแบบในส่วนของ Application Programming Interface (API) สำหรับใช้ในการติดต่อสื่อสารกันระหว่างระบบฝั่ง Front-End ที่ให้ผู้ใช้งานทำการอัปโหลดไฟล์ข้อมูลชอร์สโค้ดที่ต้องการตรวจสอบความคล้ายคลึง และระบบฝั่ง Back-End ที่รับผิดชอบในการสั่งการ JPlag ให้ประมวลผลไฟล์ชอร์สโค้ดที่ได้รับมาจากรอบฝั่ง Front-End ซึ่งมีรายละเอียด ดังนี้

#### 3.9.1 API สำหรับอัปโหลดไฟล์ชอร์สโค้ดและประมวลผลไฟล์ชอร์สโค้ด

เมื่อผู้ใช้งานทำการอัปโหลดไฟล์ชอร์สโค้ดแล้ว ระบบจะทำการส่งต่อไฟล์มายังระบบฝั่ง Back-End จากนั้นทำการแตกไฟล์ เมื่อแตกไฟล์เสร็จแล้ว ระบบจะสั่งการให้ JPlag ประมวลผลไฟล์ชอร์สโค้ด จากนั้น JPlag จะสร้างไฟล์ผลลัพธ์ขึ้นมา ซึ่งมีรายละเอียด ดังนี้

Path: /api/jplag

Parameters: 1.1) file คือ ไฟล์ชอร์สโค้ดที่มีนามสกุล zip

1.2) filename คือ ชื่อไฟล์ชอร์สโค้ดที่สุ่มขึ้นมา

1.3) destination คือ ชื่อไฟล์ชอร์สโค้ดดั้งเดิม

1.4) language คือ ภาษาที่ใช้ในการเขียนโปรแกรม

#### 3.9.2 API สำหรับคืนผลลัพธ์ของการประมวลผลไฟล์ชอร์สโค้ด

สำหรับคืนค่าไฟล์ผลลัพธ์จากการประมวลผลไฟล์ชอร์สโค้ดของ JPlag ซึ่งไฟล์ overview.json ซึ่งภายในไฟล์เป็นข้อมูลสำหรับสร้างกราฟความสัมพันธ์และตารางข้อมูลซึ่งมีรายละเอียด ดังนี้

Path: /api/result

Parameters: 1.1) filename คือ ชื่อไฟล์ชอร์สโค้ดที่สุ่มขึ้นมา

1.2) destination คือ ชื่อไฟล์ชอร์สโค้ดดั้งเดิม

#### 3.9.3 API สำหรับคืนผลลัพธ์สำหรับแสดงการเปรียบเทียบชอร์สโค้ดส่วนที่เหมือนกัน

ข้อมูลสำหรับแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกันระหว่างผู้เป็นเจ้าของชอร์สโค้ด เมื่อผู้ใช้งานทำการเลือกชุดข้อมูลชอร์สโค้ดที่คล้ายคลึงกัน ระบบจะทำการคืนข้อมูลชอร์สโค้ดที่เหมือนกันของผู้เป็นเจ้าของชอร์สโค้ด 2 คน ซึ่งมีรายละเอียด ดังนี้

Path: /api/compare

Parameters: 1.1) filename คือ ชื่อไฟล์ซอร์สโค้ดที่สู่มีขึ้นมา

1.2) destination คือ ชื่อไฟล์ซอร์สโค้ดตั้งเดิม

1.3) source คือ ชื่อผู้เป็นเจ้าของซอร์สโค้ดลำดับที่ n

1.4) target คือ ชื่อผู้เป็นเจ้าของซอร์สโค้ดลำดับที่ m

#### 3.9.4 API สำหรับคืนข้อมูลซอร์สโค้ด

ข้อมูลซอร์สโค้ดสำหรับใช้ในการการแสดงการเปรียบเทียบซอร์สโค้ดในส่วนที่เหมือนกันระบบฝั่ง Front-End จะมาติดต่อกับระบบฝั่ง Back-End เพื่อร้องขอข้อมูลซอร์สโค้ดสำหรับใช้ในการแสดงการเปรียบเทียบซอร์สโค้ดระหว่างชุดข้อมูลซอร์สโค้ด ซึ่งมีรายละเอียด ดังนี้

Path: /api/sourcecode

Parameters: 1.1) filename คือ ชื่อไฟล์ซอร์สโค้ดที่สู่มีขึ้นมา

1.2) destination คือ ชื่อไฟล์ซอร์สโค้ดตั้งเดิม

1.3) source คือ ชื่อผู้เป็นเจ้าของซอร์สโค้ด

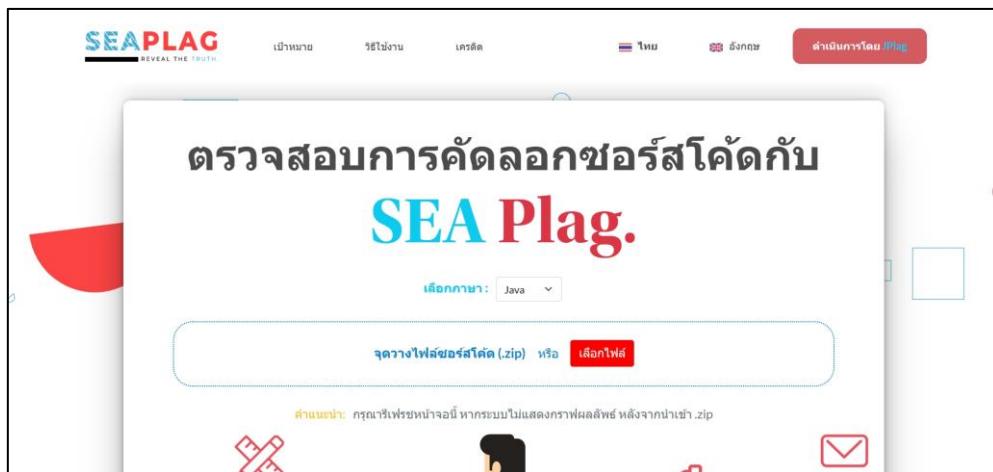
1.4) sourcecode คือ ชื่อไฟล์ซอร์สโค้ด

## บทที่ 4

### ผลการดำเนินการ

ในการดำเนินโครงการ ผู้ดำเนินโครงการได้ศึกษาค้นคว้าหาข้อมูลเกี่ยวกับเครื่องมือที่ใช้ในการตรวจสอบความคล้ายคลึงของชอร์สโค้ด รวมถึงออกแบบระบบการทำงานของระบบ และพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด ซึ่งในระหว่างการพัฒนาระบบ ผู้ดำเนินโครงการได้ทำการทดสอบและปรับแก้รายละเอียดการทำงานของแต่องค์ประกอบในระบบ เพื่อให้ระบบสามารถตอบโจทย์ผู้ใช้งานในด้านของความสะดวกในการใช้งาน และรูปแบบการแสดงผลลัพธ์จากการตรวจสอบความคล้ายคลึงของชอร์สโค้ดที่ช่วยให้ผู้ใช้งานสามารถทำความเข้าใจได้โดยง่าย โดยเนื้อหาสาระในบทนี้ คือรายละเอียดผลลัพธ์จากการพัฒนาระบบแสดงผลลัพธ์ การตรวจสอบความคล้ายคลึงของชอร์สโค้ด โดยผลลัพธ์การพัฒนาจะอยู่ในรูปแบบของหน้าจอแสดงผลแยกตามมодูลที่ผู้ดำเนินโครงการได้ออกแบบไว้จำนวนทั้งสิ้น 4 มอดูล ซึ่งผลลัพธ์การพัฒนาในแต่ละมอดูลของระบบ โดยมีรายละเอียด ดังนี้

#### 4.1 มอดูลนำเข้าข้อมูลชอร์สโค้ด (Import Source Code Module)

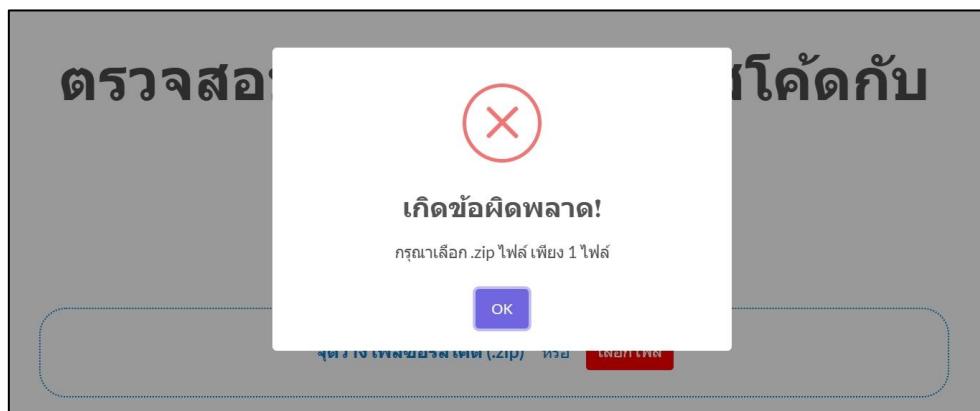


ภาพที่ 4-1 หน้าจอมอดูลนำเข้าข้อมูลชอร์สโค้ด

มอตุลสำหรับให้ผู้ใช้งาน นำเข้าชุดข้อมูลชอร์สโค้ดที่ต้องการตรวจสอบความคล้ายคลึงของชอร์สโค้ด โดยข้อมูลที่ผู้ใช้งานต้องเตรียมไฟล์เดอร์หรือไดร์ก็อกหรือสำหรับเก็บชุดข้อมูลชอร์สโค้ด ผู้ใช้งานต้องทำการเข้า zip ไฟล์ข้อมูลชอร์สโค้ด จากนั้นเลือกภาษาโปรแกรม และนำไฟล์เข้าระบบซึ่งมีรายละเอียด ดังนี้

#### 4.1.1 ผู้ใช้งานนำเข้าไฟล์ข้อมูลที่มีจำนวนมากกว่า 1 ไฟล์

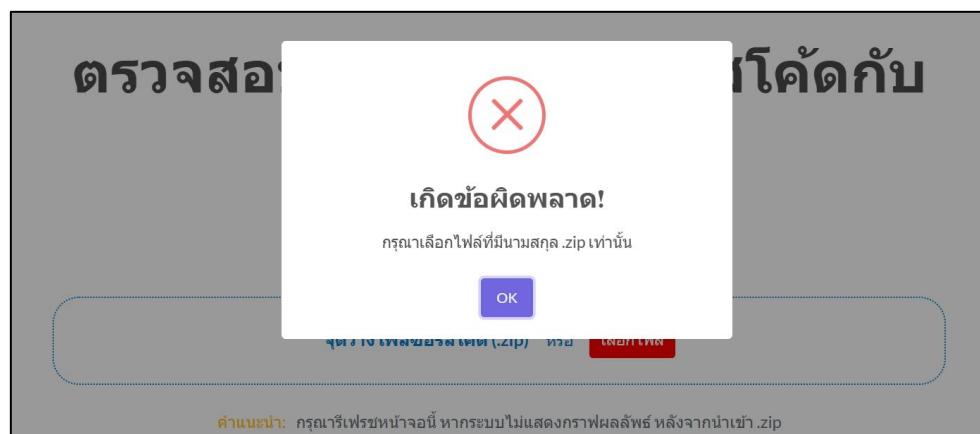
ในการนำเข้าข้อมูลสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ด ระบบได้กำหนดจำนวนไฟล์ข้อมูลที่ผู้ใช้งานสามารถนำเข้าได้เพียง 1 ไฟล์ต่อการตรวจสอบ 1 ครั้ง หากผู้ใช้งานนำเข้าไฟล์ข้อมูลซึ่งมีจำนวนมากกว่า 1 ไฟล์ ระบบจะแสดงหน้าต่างแจ้งเตือน ดังภาพที่ 4-2



ภาพที่ 4-2 หน้าต่างแจ้งเตือน กรณีที่ผู้ใช้งานนำเข้าไฟล์มากกว่า 1 ไฟล์

#### 4.1.2 ผู้ใช้งานนำเข้าไฟล์ข้อมูลซึ่งนามสกุลของไฟล์ข้อมูลนั้นไม่ใช่ .zip

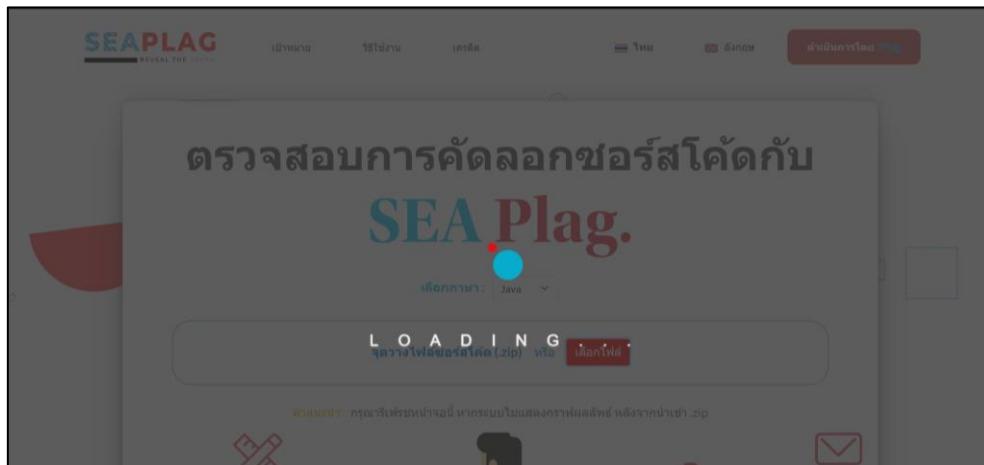
หากผู้ใช้งานนำเข้าไฟล์ข้อมูลซึ่งไม่ตรงกับประเภทไฟล์ข้อมูลที่ระบบได้กำหนดไว้ ซึ่งในที่นี้ระบบได้กำหนดประเภทไฟล์ข้อมูลชอร์สโค้ดที่มีนามสกุล .zip เท่านั้น เมื่อผู้ใช้งานนำเข้าไฟล์ข้อมูลที่ไม่ใช่ไฟล์ข้อมูลที่มีนามสกุล .zip แล้ว ระบบจะแสดงหน้าต่างแจ้งเตือน ดังภาพที่ 4-3



ภาพที่ 4-3 หน้าต่างแจ้งเตือน กรณีที่ผู้ใช้งานนำเข้าไฟล์ที่ไม่ใช่นามสกุล .zip

### 4.1.3 ผู้ใช้งานนำเข้าไฟล์ข้อมูลชื่�名สกุลของไฟล์ข้อมูลนั้นเป็น zip

หลังจากที่ผู้ใช้งานเตรียมไฟล์ข้อมูลชื่อร์สโค้ดที่ต้องการตรวจสอบความคล้ายคลึงของชอร์สโค้ดและนำไฟล์นั้นไปผ่านกระบวนการบีบอัดไฟล์ แล้วได้ไฟล์ที่มีนามสกุล .zip แล้ว เมื่อผู้ใช้งานนำเข้าไฟล์ข้อมูลนั้นที่ระบบแล้ว ระบบจะแสดงหน้าจอกำลังประมวลผลข้อมูล ก่อนที่จะเปลี่ยนเส้นทางของผู้ใช้งานไปยังหน้าจอแสดงผลลัพธ์ ดังภาพที่ 4-4



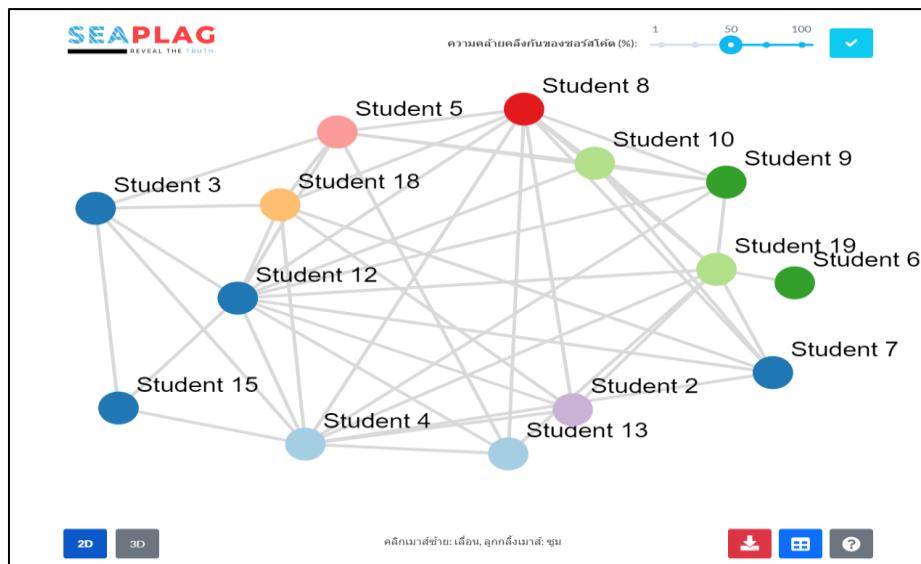
ภาพที่ 4-4 หน้าจอมดูลนำเข้าข้อมูลชอร์สโค้ด เมื่อผู้ใช้งานนำเข้าไฟล์ข้อมูลชอร์สโค้ด

## 4.2 มอดูลแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด (Output Plagiarism Module)

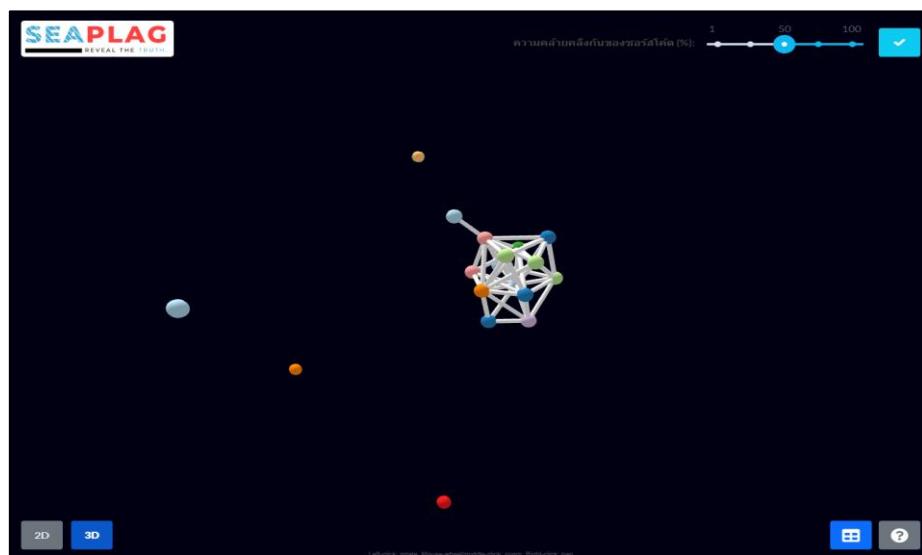
ระบบแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด หลังจากที่ผู้ใช้งานนำเข้าไฟล์ข้อมูลชอร์สโค้ดที่มีนามสกุล .zip แล้ว ระบบจะทำการตรวจสอบข้อมูลชอร์สโค้ดของเป็นเจ้าของชอร์สโค้ดแต่ละคน ซึ่งรูปแบบการแสดงผลลัพธ์ ซึ่งมีรายละเอียด ดังนี้

### 4.2.1 กราฟผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

การแสดงผลลัพธ์ในรูปแบบของกราฟ โดยได้มีการกำหนดองค์ประกอบของกราฟซึ่ง ณ ที่นี่กำหนดให้ Node แทนผู้เป็นเจ้าของชุดข้อมูลชอร์สโค้ด และให้ Edge แทนเส้นความสัมพันธ์โดยใช้ร้อยละความคล้ายคลึงกันของชุดข้อมูลชอร์สโค้ด เป็นข้อมูลสำหรับบ่งบอกความสัมพันธ์ระหว่างผู้เป็นเจ้าของชอร์สโค้ด ซึ่งมีการแสดงผลในรูปแบบ 2 มิติและ 3 มิติ ดังภาพที่ 4-5 และภาพที่ 4-6



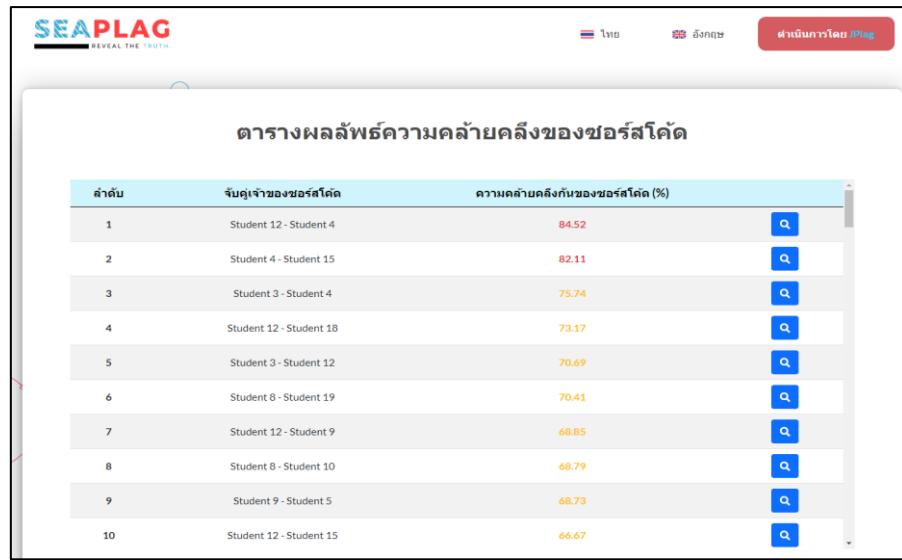
ภาพที่ 4-5 หน้าจอมอุปกรณ์แสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด  
ในรูปแบบกราฟ 2 มิติ



ภาพที่ 4-6 หน้าจอมอุปกรณ์แสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด  
ในรูปแบบกราฟ 3 มิติ

#### 4.2.2 ตารางผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

รูปแบบการแสดงผลลัพธ์จากการตรวจสอบความคล้ายคลึงของชอร์สโค้ดที่อยู่ในรูปแบบของตาราง โดยมีหัวตารางประกอบไปด้วย จับคู่เจ้าของข้อมูลชอร์สโค้ด ร้อยละความคล้ายคลึงกันของชอร์สโค้ด และการปุ่มสำหรับแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน ดังภาพที่ 4-7



The screenshot shows a table titled "ตารางผลลัพธ์ความคล้ายคลึงของชอร์สโค้ด" (Table of plagiarism results for source code). The table has three columns: "ลำดับ" (Rank), "ชื่อเจ้าของชอร์สโค้ด" (Owner of the source code), and "ความคล้ายคลึงกันของชอร์สโค้ด (%)" (Similarity percentage). The data is as follows:

ลำดับ	ชื่อเจ้าของชอร์สโค้ด	ความคล้ายคลึงกันของชอร์สโค้ด (%)
1	Student 12 - Student 4	84.52
2	Student 4 - Student 15	82.11
3	Student 3 - Student 4	75.74
4	Student 12 - Student 18	73.17
5	Student 3 - Student 12	70.69
6	Student 8 - Student 19	70.41
7	Student 12 - Student 9	68.85
8	Student 8 - Student 10	68.79
9	Student 9 - Student 5	68.73
10	Student 12 - Student 15	66.67

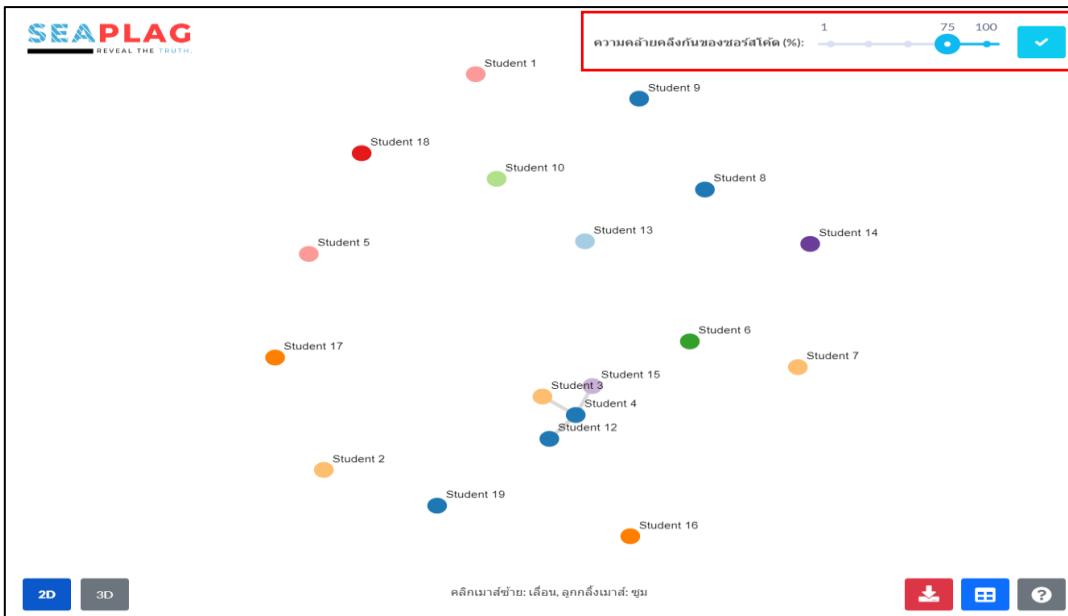
ภาพที่ 4-7 หน้าจอแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด  
ในรูปแบบตาราง

#### 4.3 การพัฒนามодูลกำหนดขอบเขตการแสดงผลลัพธ์ (Scope Output Module)

การกำหนดขอบเขตของการแสดงผลลัพธ์จากการตรวจสอบความคล้ายคลึงของชอร์สโค้ด  
ในรูปแบบของกราฟ ผู้ใช้งานสามารถเลือกขอบเขตว้อยละความคล้ายคลึงกันของชุดข้อมูลชอร์สโค้ด  
ที่ผ่านการตรวจสอบความคล้ายคลึงโดย JPlag ดังภาพที่ 4-8 และภาพที่ 4-9

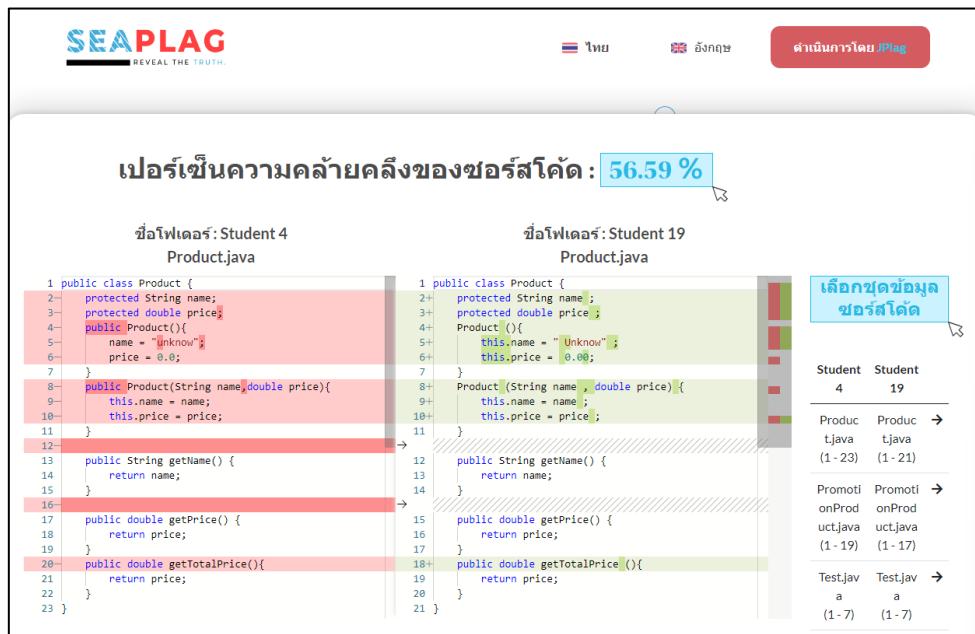


ภาพที่ 4-8 หน้าจอแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด ภายใต้ขอบเขตว้อยละ 50 ถึง 100



#### 4.4 การพัฒนามอคุลแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน (Compare Source Code Module)

ผู้ใช้งานสามารถเลือกคู่ข้อมูลชอร์สโค้ดที่คล้ายคลึงกัน ทั้งในรูปแบบการแสดงผลแบบกราฟ และรูปแบบการแสดงผลลัพธ์แบบตาราง ระบบจะแสดงหน้าจอเปรียบเทียบข้อมูลชอร์สโค้ดที่คล้ายคลึงกัน โดยผู้ใช้งานสามารถเลือกชุดข้อมูลชอร์สโค้ดที่ต้องการให้แสดงผลได้ ดังภาพที่ 4-10



ภาพที่ 4-10 หน้าจอมอคุลแสดงการเปรียบเทียบชอร์สโค้ดในส่วนที่เหมือนกัน

#### 4.5 ประสิทธิภาพการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

ผู้ดำเนินโครงการได้ทำการวัดประสิทธิภาพการตรวจสอบความคล้ายคลึงของชอร์สโค้ดของระบบ โดยใช้วิธีการจับเวลาเริ่มตั้งแต่ผู้ใช้งานอปปะโหลดไฟล์ชอร์สโค้ดไปยังระบบ จนถึงขั้นตอนที่ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบของกราฟความสัมพันธ์ ผู้ดำเนินโครงการทำการทดลองบนเครื่องโน๊ตบุ๊ค Acer Nitro AN515-54 ระบบปฏิบัติการ Windows 11 CPU INTEL Core i5-9300H 2.40 GHz RAM 8 GB ซึ่งมีรายละเอียด ดังตารางที่ 4-1

ตารางที่ 4-1 ประสิทธิภาพการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

จำนวนไฟล์เดอร์	จำนวนไฟล์ในแต่ละไฟล์เดอร์	จำนวนไฟล์รวม	เวลาประมาณ (วินาที)
10	1	10	2.75
50	1	50	4.23
100	1	100	6.00
10	3	30	3.06
50	3	150	5.96
100	3	300	7.96

จากตารางที่ 4-1 พบร่วมกันว่าประสิทธิภาพการตรวจสอบความคล้ายคลึงของชอร์สโค้ด โดยเริ่มนับตั้งแต่ผู้ใช้งานอปปะโหลดไฟล์ชอร์สโค้ด จนกระทั่งระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบตารางความสัมพันธ์ ใช้เวลาไม่นาน ซึ่งหากสังเกตจากตารางที่ 4-1 ในคอลัมน์ เวลาประมาณ (วินาที) พบร่วมกันว่าเวลาที่ใช้ในการประมวลผลของระบบอยู่ในหน่วยของวินาที แม้จะเพิ่มจำนวนไฟล์เดอร์หรือจำนวนไฟล์ชอร์สโค้ดภายในไฟล์เดอร์นั้นๆ ซึ่งเฉลี่ยแล้วระบบใช้เวลาในการประมวลผลต่างกันอยู่ที่ประมาณ 2 วินาที ซึ่งผู้ดำเนินโครงการสามารถสรุปผลได้ว่าระบบสามารถประมวลผลโดยใช้เวลาไม่นานในการแสดงกราฟความสัมพันธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด

## บทที่ 5

### สรุปผลการดำเนินงาน

ตลอดระยะเวลาการดำเนินโครงการ ผู้ดำเนินโครงการได้ทำการวิเคราะห์ปัญหาของการคัดลอกซอฟต์แวร์สโคดภายในสถาบันการศึกษาและเครื่องมือที่สถาบันศึกษาใช้ในการตรวจสอบความคล้ายคลึงของซอฟต์แวร์สโคดที่ขาดความสะอาดในการใช้งาน รวมถึงผู้ใช้งานอาจต้องใช้เวลาในการทำความเข้าใจผลลัพธ์ของการตรวจสอบ จากนั้น ผู้ดำเนินโครงการจึงได้ทำการศึกษาค้นคว้า เครื่องมือที่เหมาะสมสำหรับตรวจสอบความคล้ายคลึงของซอฟต์แวร์สโคด และได้ทำการออกแบบระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอฟต์แวร์สโคด (SEAPlag: SEAP) โดยคำนึงถึงความสะอาดสบายของผู้ใช้งานในการตรวจสอบความคล้ายคลึงของซอฟต์แวร์สโคด โดยใช้เวลาไม่นานในการทำความเข้าใจผลลัพธ์ของการตรวจสอบความคล้ายคลึงของซอฟต์แวร์สโคด

#### 5.1 สรุปผลการดำเนินโครงการ

แนวคิดการดำเนินโครงการฉบับนี้ เริ่มต้นที่ผู้ดำเนินโครงการได้มองเห็นถึงปัญหาของการคัดลอกซอฟต์แวร์สโคดภายในสถาบันการศึกษาและพบว่าสถาบันการศึกษามีการนำใช้เครื่องมือสำหรับตรวจสอบความคล้ายคลึงของนักศึกษาที่ยังไม่สะอาดต่อการใช้งาน หลังจากได้มองเห็นถึงปัญหาที่เกิดขึ้น ผู้ดำเนินโครงการจึงได้เริ่มคิดค้นระบบที่จะช่วยให้ผู้ใช้งานที่ต้องตรวจสอบความคล้ายคลึงของซอฟต์แวร์สโคด สามารถใช้ระบบในการตรวจสอบความคล้ายคลึงของซอฟต์แวร์สโคด ได้อย่างสะอาดและสามารถทำความเข้าใจผลลัพธ์ได้โดยใช้เวลาไม่นาน

ผู้ดำเนินโครงการได้เริ่มศึกษาเครื่องมือที่จะใช้สำหรับพัฒนาระบบตรวจสอบความคล้ายคลึงของซอฟต์แวร์สโคด หลังจากที่ผู้ดำเนินโครงการได้ศึกษาและทดลองใช้เครื่องมือหลากหลายประเภทแล้ว จึงพบว่า JPlag เป็นเครื่องมือที่เหมาะสมสำหรับใช้ในการพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอฟต์แวร์สโคด เนื่องจากมีภาษาที่รองรับจำนวนมากและไม่มีซ้ำกันในการใช้งาน

หลังจากเลือกเครื่องมือสำหรับใช้ในการพัฒนาระบบแล้ว ผู้ดำเนินโครงการจึงเริ่มวางแผนดำเนินโครงการ จากนั้น จึงได้ทำการออกแบบโครงสร้างและลำดับการทำงานของระบบ แล้วจึงเริ่มพัฒนาระบบโดยใช้ Angular เป็นเฟรมเวิร์กทางฝั่ง Front-End และใช้ Node.js เป็นเฟรมเวิร์กทางฝั่ง Back-End โดยผู้ดำเนินโครงการได้รับผิดชอบในการพัฒนามодูลของระบบจำนวน 4 มอดูล

ประกอบไปด้วย มодูลนำเข้าข้อมูลชอร์สโค้ด มодูลแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด มодูลกำหนดขอบเขตการแสดงผลลัพธ์ และมодูลแสดงการเปรียบเทียบชอร์สโค้ด ในส่วนที่เหมือนกัน

เมื่อพัฒนาระบบจนเสร็จสิ้นแล้ว ผู้ดำเนินโครงการได้ทำการทดสอบที่ได้ออกแบบไว้ กับระบบที่ได้ทำการพัฒนาขึ้นมา และได้ทำการทดสอบระบบในส่วนของการทำงานของแต่ละมодูล รวมถึงการปรับแก้การแสดงผลบางส่วนของหน้าจอแสดงผลให้มีความเหมาะสมมากยิ่งขึ้น เมื่อทดสอบและปรับแก้ระบบจนสิ้นสุดแล้ว ผู้ดำเนินโครงการจึงได้ส่งมอบระบบแสดงผลลัพธ์ การตรวจสอบความคล้ายคลึงของชอร์สโค้ดนี้ให้กับคณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

## 5.2 ประโยชน์ที่ได้รับ

การพัฒนาระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด ผู้ดำเนินโครงการ ได้ทำการกำหนดประโยชน์ของการจัดทำโครงการฉบับนี้ โดยมุ่งเน้นไปที่ประโยชน์ต่อผู้ใช้งาน ในการใช้งานระบบอย่างสะดวกและใช้เวลาไม่นานในการทำความเข้าใจผลลัพธ์ของการตรวจสอบ ความคล้ายคลึงของชอร์สโค้ด โดยมีรายละเอียด ดังนี้

- 1) ผู้ใช้งานสามารถใช้งานระบบสำหรับตรวจสอบความคล้ายคลึงของชอร์สโค้ดได้สะดวก และใช้เวลาไม่นานในการทำความเข้าใจผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด
- 2) สถาบันการศึกษาได้มีเครื่องมือที่เหมาะสมและสะดวกในการใช้งาน เพื่อช่วยให้ การตรวจสอบความคล้ายคลึงของชอร์สโค้ดเป็นไปอย่างมีประสิทธิภาพ
- 3) ผู้ที่สนใจสามารถนำความรู้หรือแนวคิดที่ได้จากโครงการฉบับนี้ นำไปประยุกต์ใช้ หรือต่อยอดกับงานวิจัยของตนเองต่อไป
- 4) ผู้ดำเนินโครงการได้เรียนรู้เครื่องมือและเทคนิคต่างๆ สำหรับใช้ในการพัฒนาระบบ

## 5.3 ข้อจำกัดของโครงการ

เนื่องจากผู้ดำเนินโครงการ ได้ทำการพัฒนาระบบแสดงผลลัพธ์การตรวจสอบ ความคล้ายคลึงของชอร์สโค้ดโดยใช้ JPlag ซึ่งเป็นเครื่องมือสำหรับตรวจสอบความคล้ายคลึง ของชอร์สโค้ดโดยเฉพาะ ดังนั้น ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด จึงสามารถตรวจสอบภาษาได้ตาม JPlag เท่านั้น ซึ่ง JPlag ยังเป็นเครื่องมือที่มีข้อจำกัดในเรื่องของ ภาษาโปรแกรมที่ JPlag สามารถประมวลผลและตรวจสอบความคล้ายคลึงของชอร์สโค้ดได้

## 5.4 ปัญหาและอุปสรรค

หลังจากผู้ใช้งานนำเข้าไฟล์ข้อมูลชอร์สโค้ดที่ต้องการตรวจสอบความคล้ายคลึงของชอร์สโค้ดไฟล์ข้อมูลชอร์สโค้ดนั้นจะถูกบันทึกที่เครื่องเซิร์ฟเวอร์ จากนั้นเซิร์ฟเวอร์จะสั่งการให้ JPlag ประมวลผลข้อมูลชอร์สโค้ดในไฟล์เพื่อตรวจหาชุดข้อมูลชอร์สโค้ดที่คล้ายคลึงกัน โดยปกติแล้ว หลังจากประมวลผลเสร็จ JPlag จะสร้างไฟล์ผลลัพธ์ที่ชื่อ overview.json ขึ้น ซึ่งเป็นไฟล์ผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด โดยผู้ดำเนินโครงการได้pub เจอบัญชาของการตรวจสอบความคล้ายคลึงของชอร์สโค้ดโดยใช้ JPlag พบร่วมหลายครั้งที่ JPlag ไม่สร้างไฟล์ผลลัพธ์ที่ชื่อ overview.json ให้ ทำให้ระบบไม่สามารถแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ดได้ ทั้งในรูปแบบของกราฟและตาราง

## 5.5 ข้อเสนอแนะ

ข้อจำกัดทางด้านภาษาโปรแกรมที่ JPlag รองรับนั้น ทำให้ระบบไม่สามารถประมวลผลความคล้ายคลึงของชอร์สโค้ดที่ถูกเขียนขึ้นด้วยภาษาหนึ่งได้ ดังนั้น ผู้ดำเนินโครงการจึงข้อเสนอแนะให้เพิ่มเติมความสามารถในการรองรับภาษาโปรแกรมของ JPlag ซึ่งจะช่วยให้ระบบสามารถประมวลผลชอร์สโค้ดที่ถูกเขียนขึ้นด้วยภาษาโปรแกรมอื่นๆ ได้อย่างครอบคลุมและตอบสนองต่อความต้องการของผู้ใช้งานได้มากยิ่งขึ้น

## 5.6 แนวโน้มหรือทิศทางการพัฒนาในอนาคต

ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ด มีวัตถุประสงค์การทำงานของระบบซึ่งให้ความสำคัญกับการแสดงผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ดในรูปแบบของกราฟ ซึ่งผู้ดำเนินโครงการได้มองว่าเป็นรูปแบบการแสดงผลลัพธ์ที่เหมาะสมและช่วยให้ผู้ใช้งานสามารถทำความเข้าใจผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ดโดยใช้เวลาไม่นาน แต่ ณ ปัจจุบันนี้ ระบบยังไม่มีการเก็บข้อมูลการประมวลผลชอร์สโค้ดของผู้ใช้งาน โดยหากผู้ใช้งานปิดระบบหลังจากใช้งานระบบเสร็จสิ้นแล้ว ข้อมูลที่ผู้ใช้งานเคยใช้ในการตรวจสอบความคล้ายคลึงของชอร์สโค้ดจะถูกเก็บบันทึกไว้บนเครื่องเซิร์ฟเวอร์ และไม่มีความสามารถเข้าถึงข้อมูลนั้นได้ ดังนั้น ผู้ดำเนินโครงการจึงอยากให้ระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของชอร์สโค้ดนี้ มีฟังก์ชันการทำงานในส่วนของการบริหารจัดการเกี่ยวกับข้อมูลชอร์สโค้ดที่นำเข้า และผลลัพธ์ของการตรวจสอบความคล้ายคลึงของชอร์สโค้ด โดยแยกเป็นของผู้ใช้งานแต่ละคน ซึ่งจะช่วยให้ผู้ใช้งานสามารถใช้งานระบบได้อย่างสะดวกและมีประสิทธิภาพมากยิ่งขึ้น

## บรรณานุกรม

- [1] H. Cheers, Y. Lin and S. P. Smith, "Academic Source Code Plagiarism Detection by Measuring Program Behavioral Similarity," in IEEE Access, vol. 9, pp. 50391-50412, 2021, doi: 10.1109/ACCESS.2021.3069367.
- [2] JPlag. [ออนไลน์]. เข้าถึงได้จาก: <https://github.com/jplag/JPlag>. (วันที่สืบค้น: 19 พฤษภาคม 2565)
- [3] Moss. [ออนไลน์]. เข้าถึงได้จาก: <https://theory.stanford.edu/~aiken/moss/>. (วันที่สืบค้น: 19 พฤษภาคม 2565)
- [4] Plaggie. [ออนไลน์]. เข้าถึงได้จาก: <https://www.cs.hut.fi/Software/Plaggie>. (วันที่สืบค้น: 19 พฤษภาคม 2565)
- [5] Marble. [ออนไลน์]. เข้าถึงได้จาก: <https://webspace.science.uu.nl/~hage0101/downloads/marble-slides.pdf>. (วันที่สืบค้น: 19 พฤษภาคม 2565)
- [6] TypeScript. [ออนไลน์]. เข้าถึงได้จาก: <https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>. (วันที่สืบค้น: 25 พฤษภาคม 2565)
- [7] HTML. [ออนไลน์]. เข้าถึงได้จาก: <https://www.w3schools.com/html/>. (วันที่สืบค้น: 2 ธันวาคม 2565)
- [8] CSS. [ออนไลน์]. เข้าถึงได้จาก: <https://www.w3schools.com/w3css/default.asp>. (วันที่สืบค้น: 2 ธันวาคม 2565)
- [9] JavaScript. [ออนไลน์]. เข้าถึงได้จาก: <https://www.w3schools.com/js/>. (วันที่สืบค้น: 4 ธันวาคม 2565)
- [10] Angular. [ออนไลน์]. เข้าถึงได้จาก: <https://angular.io/guide/what-is-angular>. (วันที่สืบค้น: 10 ธันวาคม 2565)
- [11] Node.js. [ออนไลน์]. เข้าถึงได้จาก: <https://nodejs.org/en/docs/>. (วันที่สืบค้น: 13 ธันวาคม 2565)

## บรรณานุกรม

- [12] Express.js. [ออนไลน์]. เข้าถึงได้จาก: <https://expressjs.com/en/starter/basic-routing.html>. (วันที่สืบค้น: 15 ธันวาคม 2565)
- [13] Force directed graph. [ออนไลน์]. เข้าถึงได้จาก: <https://www.npmjs.com/package/force-graph>. (วันที่สืบค้น: 20 ธันวาคม 2565)
- [14] 3D force-directed graph. [ออนไลน์]. เข้าถึงได้จาก: <https://github.com/vasturiano/3d-force-graph>. (วันที่สืบค้น: 22 ธันวาคม 2565)
- [15] Tables Bootstrap. [ออนไลน์]. เข้าถึงได้จาก: <https://getbootstrap.com/docs/4.1/content/tables/>. (วันที่สืบค้น: 23 ธันวาคม 2565)
- [16] Visual Studio Code. [ออนไลน์]. เข้าถึงได้จาก: <https://code.visualstudio.com/docs/setup/setup-overview>. (วันที่สืบค้น: 25 ธันวาคม 2565)
- [17] Google Chrome [ออนไลน์]. เข้าถึงได้จาก: <https://www.google.com/chrome/dev/>. (วันที่สืบค้น: 25 ธันวาคม 2565)
- [18] GitLab. [ออนไลน์]. เข้าถึงได้จาก: <https://about.gitlab.com/>. (วันที่สืบค้น: 25 ธันวาคม 2565)
- [19] Connect an Angular App to a Node.js API. [ออนไลน์]. เข้าถึงได้จาก: <https://jasonwatmore.com/post/2022/11/20/angular-nodejs-connect-an-angular-app-to-a-nodejs-api>. (วันที่สืบค้น: 5 มกราคม 2566)
- [20] Express JS Tutorial: What is Express in Node JS. [ออนไลน์]. เข้าถึงได้จาก: <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js/>. (วันที่สืบค้น: 7 มกราคม 2566)
- [21] Child process. [ออนไลน์]. เข้าถึงได้จาก: <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js/>. (วันที่สืบค้น: 12 มกราคม 2566)
- [22] Spawnsync. [ออนไลน์]. เข้าถึงได้จาก: [https://www.tabnine.com/code/javascript/functions/child\\_process/spawnsync](https://www.tabnine.com/code/javascript/functions/child_process/spawnsync) (วันที่สืบค้น: 12 มกราคม 2566)

## ประวัติย่อของผู้ดำเนินโครงการ

**ชื่อ-สกุล** นายสิริวิชญ์ นิติสุนทรลักษณ์  
**วัน เดือน ปีเกิด** 15 กุมภาพันธ์ พ.ศ. 2544  
**สถานที่เกิด** จังหวัดชลบุรี  
**ที่อยู่** 199/369 ซอย 15 หมู่บ้านรุ่งกิจวิลล่า 7 ถนนร่มเกล้า  
**หมายเลขคลอง** แขวงคลองประเวศ เขตลาดกระบัง กรุงเทพมหานคร 10520  
**โทรศัพท์** 094 – 770 – 5119

### ประวัติการศึกษา

**พ.ศ. 2562 – ปัจจุบัน** กำลังศึกษาอยู่ที่คณะวิทยาการสารสนเทศ สาขาวิชาบริหารธุรกิจและเทคโนโลยีสารสนเทศ มหาวิทยาลัยบูรพา  
**พ.ศ. 2561** จบชั้นมัธยมศึกษาตอนปลาย โรงเรียนเทพศิรินทร์ร่วมเกล้า