

ระบบแสดงผลการตรวจสอบความคล้ายคลึงของซอร์สโค้ด SEAPlag

สิริวิญญู จิตินสุนทรลักษณ์ พรณภัส เขียวอิม พิระศักดิ์ เพียรประสิทธิ์
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา อำเภอเมือง จังหวัดชลบุรี
62160162@go.buu.ac.th, 62160158@go.buu.ac.th, peerasak@buu.ac.th

บทคัดย่อ

หลายสถาบันการศึกษาต่างก็มีรายวิชาที่เปิดสอนเกี่ยวกับพื้นฐานการเขียนโปรแกรมคอมพิวเตอร์ ซึ่งการจัดการเรียนการสอนมีทั้งการเรียนรู้ทฤษฎีและการฝึกปฏิบัติการด้วยการฝึกเขียนโปรแกรมตามโจทย์ปัญหา ซึ่งในขั้นตอนการทำโจทย์ปัญหานี้เองที่พบว่าอาจมีนักศึกษาบางกลุ่มทำการคัดลอกซอร์สโค้ดกัน ซึ่งเป็นสาเหตุของปัญหาที่ทำให้ให้นักศึกษาเขียนโปรแกรมไม่ได้และส่งผลกระทบต่อตัวนักศึกษาเอง ดังนั้น อาจารย์ผู้สอนส่วนใหญ่จึงมักมีการกำกับติดตาม และอาจมีการนำใช้เครื่องมือสำหรับตรวจสอบการคัดลอกซอร์สโค้ด ซึ่งเครื่องมือเหล่านี้ยังไม่สะดวกสบายต่อการใช้งานมากนัก เนื่องจากข้อมูลมักแสดงออกมาในรูปแบบตารางข้อมูล ดังนั้น ระบบแสดงผลการตรวจสอบความคล้ายคลึงของซอร์สโค้ด (SEAPlag: SEAP) จึงถูกพัฒนาขึ้นเพื่ออำนวยความสะดวกต่ออาจารย์ผู้สอนในสถาบันการศึกษาที่ต้องการตรวจสอบการคัดลอกซอร์สโค้ดของนักศึกษา โดยมีคุณลักษณะหลักของระบบประกอบไปด้วย โมดูลนำเข้าข้อมูลซอร์สโค้ด โมดูลแสดงผลการตรวจสอบความคล้ายคลึงของซอร์สโค้ด โมดูลส่งออกผลลัพธ์การตรวจสอบความคล้ายคลึงของซอร์สโค้ด โมดูลกำหนดขอบเขตการแสดงผล และโมดูลแสดงการเปรียบเทียบซอร์สโค้ดในส่วนที่เหมือนกัน ผลลัพธ์ของระบบจะอยู่ในรูปแบบของกราฟความสัมพันธ์ และตารางร้อยละความคล้ายคลึงของซอร์สโค้ด ผู้ใช้งานสามารถดูข้อมูลการเปรียบเทียบซอร์สโค้ดในส่วนที่คล้ายกัน

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. บทนำ

ความก้าวหน้าทางด้านเทคโนโลยีและสื่อโซเชียลมีเดีย นำมาซึ่งความสะดวกในการดำเนินชีวิตในด้านต่างๆ ไม่ว่าจะเป็นในด้านของเศรษฐกิจ การคมนาคม โทรคมนาคม หรือกระทั่งในด้านของการศึกษา ซึ่งเป็นสิ่งสำคัญที่ช่วยผลักดันให้นักศึกษาผู้แสวงหาความรู้ สามารถเข้าถึงข้อมูลข่าวสารที่มีปริมาณมหาศาลและมีความหลากหลายของโครงสร้างเนื้อหาได้อย่างไม่มีที่สิ้นสุด แต่ในขณะเดียวกัน ความก้าวหน้าก็มาพร้อมกับช่องทางที่ช่วยให้นักศึกษาบางกลุ่ม สามารถกระทำการอะไรบางอย่างอันเป็นการเอาเปรียบหรือละเมิดผลงานของผู้อื่น ซึ่งในที่นี่กล่าวถึงการคัดลอกซอร์สโค้ดภายในสถาบันการศึกษา ซึ่งเป็นปัญหาที่สำคัญ เนื่องจากถ้านักศึกษาทำการคัดลอกซอร์สโค้ดกัน ซึ่งเป็นต้นเหตุของการสร้างลักษณะนิสัยในการคัดลอกผลงานของผู้อื่น ทั้งโดยได้รับอนุญาตหรือไม่ได้รับอนุญาตจากผู้เป็นเจ้าของ นอกจากนี้ ปัญหาสำคัญอีกประการหนึ่งคือหากนักศึกษาทำการคัดลอกซอร์สโค้ดกันแล้ว นักศึกษาเหล่านั้นก็ไม่เกิดการเรียนรู้ ซึ่งในปัจจุบันมีเครื่องมือสำหรับตรวจสอบการความคล้ายคลึงของซอร์สโค้ด แต่เครื่องมือที่มีอยู่ในปัจจุบันนั้น อาจมีค่าใช้จ่ายในการใช้งานหรือซอฟต์แวร์ที่เป็นโอเพนซอร์สที่สามารถใช้งานได้ฟรีก็ยังมีการใช้งานที่ค่อนข้างยุ่งยากและผลลัพธ์ของการตรวจสอบความคล้ายคลึงของซอร์สโค้ดนั้น มีรายละเอียดของผลลัพธ์ที่ผู้ใช้งานต้องใช้เวลาในการทำความเข้าใจ

2. วัตถุประสงค์ในการจัดทำโครงการ

- 2.1) เพื่อพัฒนาระบบสำหรับตรวจสอบความคล้ายคลึงของซอร์สโค้ดให้สามารถทำงานได้อย่างมีประสิทธิภาพ
- 2.2) เพื่ออำนวยความสะดวกให้กับผู้ใช้งานที่ต้องการใช้เครื่องมือที่เหมาะสมสำหรับตรวจสอบความคล้ายคลึงของซอร์สโค้ด
- 2.3) เพื่อแสดงให้เห็นถึงความเชื่อมโยงและความสัมพันธ์ของผลลัพธ์จากการตรวจสอบความคล้ายคลึงของซอร์สโค้ด

3. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ผู้ดำเนินโครงการได้ทำการวิเคราะห์ปัญหาของการคัดลอกซอร์สโค้ดภายในสถาบันการศึกษา แล้วจึงทำการศึกษาค้นคว้าหาข้อมูลและเครื่องมือที่เหมาะสมสำหรับใช้ในการพัฒนาระบบแสดงผลการตรวจสอบความคล้ายคลึงของซอร์สโค้ด โดยเริ่มศึกษาจากพฤติกรรมของการคัดลอกซอร์สโค้ดของนักศึกษาและรูปแบบของการคัดลอกซอร์สโค้ด จากนั้นจึงทำการศึกษาการทำงานของเครื่องมือที่จะใช้ในการตรวจสอบความคล้ายคลึงของซอร์สโค้ดโดยมีรายละเอียด ดังนี้

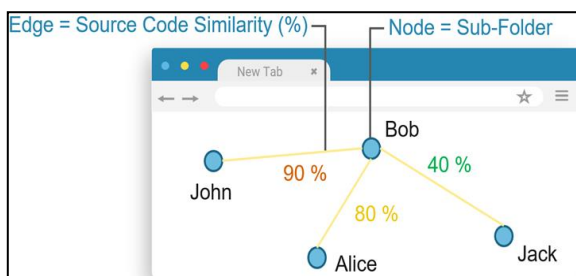
3.1 ความคล้ายคลึงของซอร์สโค้ด

การตรวจสอบส่วนที่เหมือนกันหรือคล้ายคลึงกันระหว่างชุดข้อมูลซอร์สโค้ด 2 ชุด โดยการตรวจสอบความคล้ายคลึงของซอร์สโค้ดนี้มีวัตถุประสงค์เพื่อค้นหาซอร์สโค้ดในส่วนที่เหมือนกันหรือคัดลอกกันมา อันเป็นวิธีการที่จะช่วยให้เห็นถึงความสัมพันธ์ระหว่างชุดข้อมูลซอร์สโค้ดชุดนั้นๆ ซึ่งผลลัพธ์ของการแสดงความคล้ายคลึงของซอร์สโค้ดจะอยู่ในรูปของตัวเลข ซึ่งเป็นตัวเลขที่แสดงถึงร้อยละความคล้ายคลึงของซอร์สโค้ดระหว่างชุดข้อมูลซอร์สโค้ด 2 ชุด อันเป็นข้อมูลเพื่อผู้ใช้งานสามารถทำความเข้าใจว่าข้อมูลซอร์สโค้ดทั้ง 2 ชุดนั้นมีความสัมพันธ์กันอย่างไร ซึ่งระบบแสดงผลการตรวจสอบความคล้ายคลึงของซอร์สโค้ดนี้ได้ถูกออกแบบมาเพื่อให้สามารถตรวจสอบซอร์สโค้ดที่ถูกเขียนขึ้นโดยภาษาโปรแกรมภาษาเดียวกันเท่านั้น ซึ่งในการตรวจสอบความคล้ายคลึงของซอร์สโค้ด

ระบบจะนำไฟล์ข้อมูลซอร์สโค้ดจำนวน 2 ไฟล์ มาเปรียบเทียบซอร์สโค้ดในส่วนที่เหมือนกัน

3.2 Data Visualization

การแปลงข้อมูลในลักษณะของข้อความให้มาอยู่ในรูปแบบของแผนภาพ กราฟ หรือวิดีโอ มีวัตถุประสงค์เพื่อใช้ในการนำเสนอความสัมพันธ์ของข้อมูลชุดนั้นๆ ตามความต้องการของผู้จัดทำ ทำให้ผู้รับฟังสามารถทำความเข้าใจข้อมูลได้อย่างรวดเร็วและชัดเจน โดยเฉพาะอย่างยิ่งกับชุดข้อมูลที่มีข้อมูลจำนวนมากและมีการเปลี่ยนแปลงของข้อมูลอย่างต่อเนื่อง ซึ่งในปัจจุบันนี้ มีเครื่องมือสำหรับจัดทำ Data Visualization ที่อำนวยความสะดวกให้ผู้จัดทำอยู่มากมาย และหลากหลาย ทั้งแบบที่ไม่มีค่าใช้จ่ายเพิ่มเติมและมีค่าใช้จ่ายเพิ่มเติม โดยการแสดงผลจากการตรวจสอบความคล้ายคลึงของระบบ ผู้ดำเนินโครงการได้เลือกการแสดงผลในรูปแบบของกราฟความสัมพันธ์ ซึ่งเป็นรูปแบบที่เหมาะสมสำหรับให้ผู้ใช้งานทำความเข้าใจผลลัพธ์ของการตรวจสอบความคล้ายคลึงของซอร์สโค้ด รวมถึงการแสดงผลในรูปแบบของตารางผลลัพธ์ความคล้ายคลึงของซอร์สโค้ด สำหรับให้ผู้ใช้งานดูรายละเอียดของการตรวจสอบความคล้ายคลึงของซอร์สโค้ด โดยผู้ดำเนินโครงการได้กำหนดรูปแบบการแสดงผลของระบบ ดังนี้



ภาพที่ 1 การแสดงผลในรูปแบบกราฟ

จากภาพที่ 1 Node หมายถึง โฟลเดอร์หรือไดเรกทอรีย่อย ซึ่งใช้แทนผู้เป็นเจ้าของซอร์สโค้ด ชุดนั้นๆ โดยแต่ละ Node จะมีการแสดงชื่อของผู้เป็นเจ้าของชุดซอร์สโค้ดกำกับเอาไว้ และ Edge หมายถึง ความสัมพันธ์ระหว่าง Node ซึ่งในที่นี้คือร้อยละความคล้ายคลึงกันระหว่างชุดซอร์สโค้ด (Source Code Similarity)

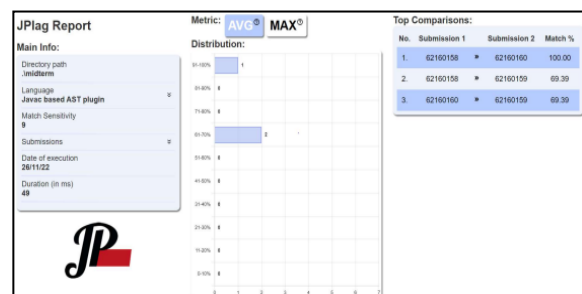
Matched	Source Code Similarity (%)
Bob - Alice	100.00
Bob - Jack	80.00
Alice - Jack	50.00

ภาพที่ 2 การแสดงผลในรูปแบบตาราง

จากภาพที่ 2 คอลัมน์ Matched คือ ความสัมพันธ์ระหว่างผู้เป็นเจ้าของชุดซอร์สโค้ดที่ผ่านการตรวจสอบความคล้ายคลึงของซอร์สโค้ด และคอลัมน์ Source Code Similarity (%) คือ ผลลัพธ์ร้อยละความคล้ายคลึงกันของชุดซอร์สโค้ดที่ผ่านการตรวจสอบความคล้ายคลึงของซอร์สโค้ด

3.3 JPlag

ไลบรารีที่ถูกสร้างขึ้นเพื่อใช้สำหรับประมวลผลการคัดลอกซอร์สโค้ด โดยไลบรารีจะทำการตรวจสอบซอร์สโค้ดที่ผู้ใช้งานต้องการ ผ่านการนำเข้าข้อมูลซอร์สโค้ด (Import Data) และส่งออกผลลัพธ์ของการตรวจสอบในรูปแบบข้อมูลที่อยู่ในโครงสร้างข้อมูลแบบ JavaScript Object Notation (JSON) ซึ่งไลบรารีรองรับการตรวจสอบซอร์สโค้ดที่ถูกเขียนขึ้นจากหลากหลายภาษา ยกตัวอย่างเช่น Java, C++, C#, Go, Python, Scala และ Swift เป็นต้น ซึ่ง JPlag เป็นเครื่องมือสำหรับตรวจสอบความคล้ายคลึงของซอร์สโค้ดที่ผู้ดำเนินโครงการเลือกใช้ เนื่องจาก JPlag รองรับภาษาโปรแกรมจำนวนมากและเป็นเครื่องมือประเภท Open Source ซึ่งสามารถใช้งานได้โดยไม่มีค่าใช้จ่ายเพิ่มเติม ผลลัพธ์จากการประมวลผลโดย JPlag มีรายละเอียดดัง ภาพที่ 3



ภาพที่ 3 การแสดงผลของ JPlag

4. การวิเคราะห์และออกแบบระบบ

การดำเนินโครงการครั้งนี้ผู้ดำเนินโครงการได้รับมอบหมายให้พัฒนาระบบแสดงผลการตรวจสอบความคล้ายคลึงของซอร์สโค้ด (SEAPlag) โดยในบทนี้จะกล่าวถึงการวิเคราะห์ และพัฒนาเว็บแอปพลิเคชัน ซึ่งเป็นการกำหนดองค์ประกอบโดยรวมของระบบ ประเภทของผู้ใช้งาน และปฏิสัมพันธ์ระหว่างผู้ใช้งานกับระบบ นอกจากนี้ยังได้นำความรู้ในการเรียนจากวิชาต่างๆ ตลอดระยะเวลาที่ผ่านมาของผู้ดำเนินโครงการเข้ามาประยุกต์ใช้เพื่อพัฒนาระบบ โดยเนื้อหาสาระในบทนี้ ครอบคลุมในเรื่องของการออกแบบลำดับการทำงานของระบบ การออกแบบหน้าจอแสดงผล รวมถึงการทำงานของระบบในแต่ละโมดูล

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

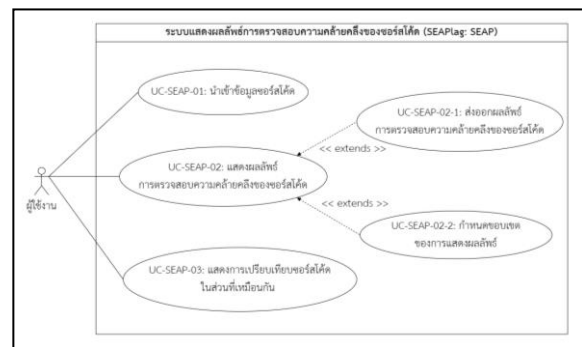
ลำดับการทำงานของระบบแสดงผลการตรวจสอบความคล้ายคลึงของซอร์สโค้ด คือ รายละเอียดการทำงานของระบบในส่วนของการนำเข้าไฟล์ข้อมูลซอร์สโค้ด จนถึง การแสดงผลการตรวจสอบความคล้ายคลึงของซอร์สโค้ด โดยมีรายละเอียด ดังภาพที่ 4



ภาพที่ 4 ลำดับการทำงานของระบบ

4.3 แผนภาพยูสเคส (Use Case Diagram)

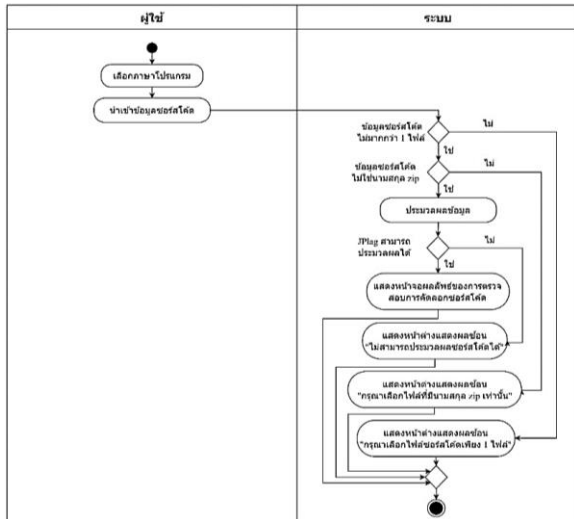
ในการวิเคราะห์และออกแบบโครงสร้างการทำงานของระบบ ผู้ดำเนินโครงการได้จำแนกมอดูลการทำงานของระบบออกเป็น 5 มอดูล และผู้มีส่วนเกี่ยวข้องกับระบบในที่นี้ คือ ผู้ใช้งาน ได้แก่ มอดูลนำเข้าข้อมูลซอร์สโค้ด มอดูลแสดงผลการตรวจสอบความคล้ายคลึงของซอร์สโค้ด มอดูลส่งออกผลลัพธ์การตรวจสอบความคล้ายคลึงของซอร์สโค้ด มอดูลกำหนดขอบเขตของการแสดงผล และ มอดูลแสดงการเปรียบเทียบซอร์สโค้ดในส่วนที่เหมือนกัน ซึ่งผู้ใช้งานระบบในที่นี้ คือ ผู้ที่มีความต้องการในการใช้งานระบบสำหรับตรวจสอบความคล้ายคลึงของซอร์สโค้ด โดยมีรายละเอียด ดังภาพที่ 5



ภาพที่ 5 แผนภาพยูสเคส

4.4 แผนภาพกิจกรรม (Activity Diagram)

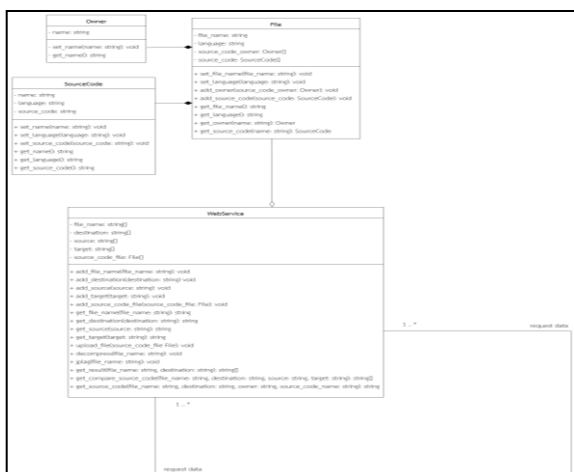
ในส่วนของการออกแบบแผนภาพกิจกรรมเป็นการอธิบายถึงลำดับขั้นตอนการทำงานของแต่ละมอดูล โดยแสดงให้เห็นถึงการปฏิสัมพันธ์กันระหว่างผู้ใช้งานกับระบบแสดงผลการตรวจสอบความคล้ายคลึงของซอร์สโค้ดอย่างเป็นลำดับขั้นตอน โดยมอดูลนำเข้าข้อมูลซอร์สโค้ด เมื่อผู้ใช้งานต้องการนำเข้าข้อมูลซอร์สโค้ดเพื่อตรวจสอบความคล้ายคลึงของซอร์สโค้ด ซึ่งผู้ใช้งานต้องทำการเลือกภาษาโปรแกรมที่ต้องการตรวจสอบ และทำการนำเข้าข้อมูลซอร์สโค้ดที่ต้องการ เพื่อนำไปใช้ในการตรวจสอบความคล้ายคลึงของซอร์สโค้ดภายในโปรแกรมต่อไป โดยมีรายละเอียดแผนภาพกิจกรรม ดังภาพที่ 6



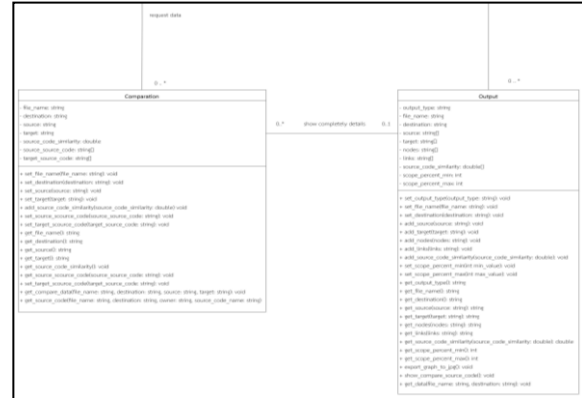
ภาพที่ 6 แผนภาพกิจกรรมมอดูลนำเข้าข้อมูลซอร์สโค้ด

4.5 แผนภาพคลาส (Class Diagram)

ผู้ดำเนินโครงการได้ทำการจำแนกองค์ประกอบหลักของระบบ ออกเป็นทั้งหมด 6 ส่วน ได้แก่ ส่วนของผู้เป็นเจ้าของซอร์สโค้ด ส่วนของซอร์สโค้ด ส่วนของไฟล์ที่รวมข้อมูลซอร์สโค้ด ส่วนบริการของระบบ ส่วนของการแสดงผลความคล้ายคลึงของซอร์สโค้ด และส่วนของการเปรียบเทียบซอร์สโค้ดในส่วนที่เหมือนกัน จากนั้นจึงนำองค์ประกอบดังกล่าวบันทึกลงบนแผนภาพคลาสเพื่อใช้อธิบายถึงคุณลักษณะ (Attribute) และความสามารถ (Method) ของแต่ละองค์ประกอบ โดยผู้ดำเนินโครงการได้ระบุความสัมพันธ์ระหว่างองค์ประกอบ (Relationship) ในแผนภาพ โดยใช้เส้นความสัมพันธ์ ซึ่งมีรายละเอียดดังภาพที่ 7



ภาพที่ 7 แผนภาพคลาส



ภาพที่ 7 แผนภาพคลาส (ต่อ)

5. ผลการดำเนินงาน

ระบบแสดงผลการตรวจสอบความคล้ายคลึงของซอร์สโค้ดนั้นมีความสามารถในการนำไลบรารี JPLag มาช่วยตรวจสอบความคล้ายคลึงของซอร์สโค้ดได้ และสามารถนำผลลัพธ์จากการตรวจสอบมาแสดงผลในรูปแบบตารางข้อมูล และรูปแบบกราฟแสดงความสัมพันธ์ นอกจากนี้ผู้ใช้งานยังสามารถกำหนดขอบเขตการตรวจสอบความคล้ายคลึงกันของข้อมูลเพื่อแสดงผลระดับความคล้ายคลึงที่ผู้ใช้งานได้ทำการกำหนด และส่งออกผลลัพธ์ในรูปแบบไฟล์รูปภาพให้แก่ผู้ใช้งานระบบได้ ซึ่งมีผลลัพธ์ในการพัฒนาระบบ ดังนี้

5.1 มอดูลนำเข้าข้อมูลซอร์สโค้ด

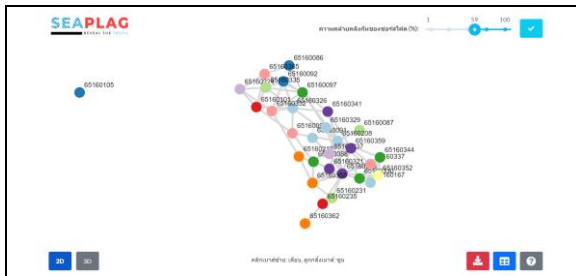
มอดูลสำหรับให้ผู้ใช้งาน นำเข้าชุดข้อมูลซอร์สโค้ดที่ต้องการตรวจสอบความคล้ายคลึงของซอร์สโค้ด โดยข้อมูลที่ผู้ใช้งานต้องเตรียมไฟล์เตอร์หรือไดเรกทอรีสำหรับเก็บชุดข้อมูลซอร์สโค้ด ผู้ใช้งานต้องทำการเข้า zip ไฟล์ข้อมูลซอร์สโค้ด จากนั้นเลือกภาษาโปรแกรม และนำไฟล์เข้าระบบ ซึ่งมีรายละเอียด ดังภาพที่ 8



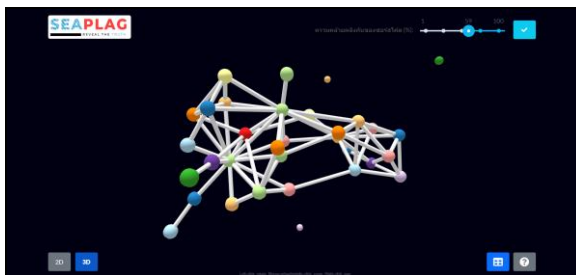
ภาพที่ 8 หน้าจอมอดูลนำเข้าข้อมูลซอร์สโค้ด

5.2 มอดูลแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอร์สโค้ด

การแสดงผลลัพธ์ในรูปแบบของกราฟ โดยได้มีการกำหนดองค์ประกอบของกราฟซึ่ง ณ ที่นี้ กำหนดให้ Node แทนผู้เป็นเจ้าของชุดข้อมูลซอร์สโค้ด และให้ Edge แทนเส้นความสัมพันธ์โดยใช้ร้อยละความคล้ายคลึงกันของชุดข้อมูลซอร์สโค้ดเป็นข้อมูลสำหรับบ่งบอกความสัมพันธ์ระหว่างผู้เป็นเจ้าของซอร์สโค้ด ซึ่งมีการแสดงผลในรูปแบบ 2 มิติและ 3 มิติ รวมถึงการแสดงผลลัพธ์ในรูปแบบตาราง ดังภาพที่ 9 ภาพที่ 10 และภาพที่ 11



ภาพที่ 9 หน้าจอแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอร์สโค้ดในรูปแบบกราฟ 2 มิติ



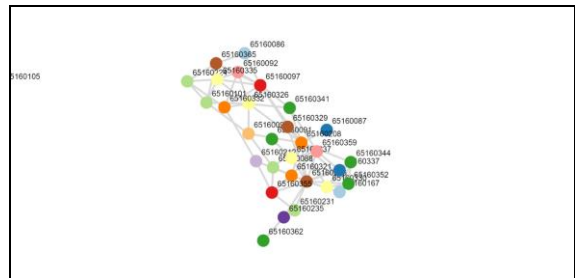
ภาพที่ 10 หน้าจอแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอร์สโค้ดในรูปแบบกราฟ 3 มิติ

ลำดับ	ชื่อผู้เจ้าของซอร์สโค้ด	ความคล้ายคลึงของซอร์สโค้ด (%)
1	65160036-65160037	74.46
2	65160038-65160092	74.23
3	65160036-65160167	73.13
4	65160001-65160035	71.59
5	65160037-65160044	71.48
6	65160092-65160035	70.23
7	65160097-65160086	69.01

ภาพที่ 11 หน้าจอแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอร์สโค้ดรูปแบบตารางข้อมูล

5.3 มอดูลส่งออกผลลัพธ์การตรวจสอบความคล้ายคลึงของซอร์สโค้ด

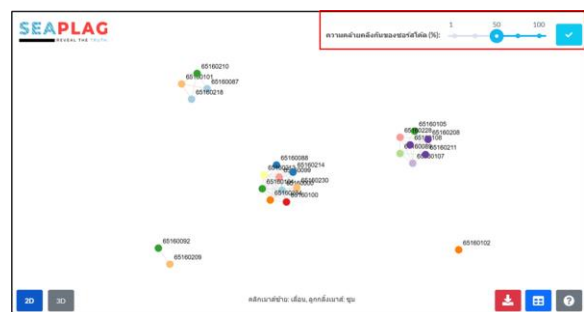
ผู้ใช้งานต้องทำการเลือกปุ่ม (Button) “Export” โดยระบบจะทำการประมวลผลเพื่อส่งออกผลลัพธ์ ซึ่งผลลัพธ์จะอยู่ในรูปแบบของไฟล์รูปภาพที่มีนามสกุลไฟล์ .jpg โดยมีการกำหนดชื่อเริ่มต้น คือ “Graph_SEAPlag” ทั้งนี้ทางผู้ใช้งานจะสามารถส่งออกผลลัพธ์ได้ในรูปแบบของไฟล์ 2 มิติเพียงอย่างเดียว ซึ่งทางผู้ใช้งานจะได้รับภาพตัวอย่างผลลัพธ์การส่งออกข้อมูลของกราฟ 2 มิติ ดังภาพที่ 12



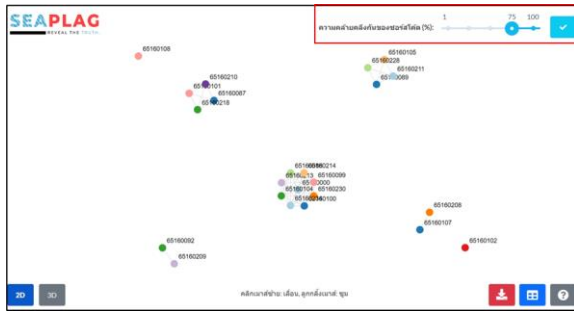
ภาพที่ 12 ตัวอย่างผลลัพธ์การส่งออกข้อมูลผลลัพธ์ของกราฟ 2 มิติ

5.4 มอดูลแสดงกำหนดขอบเขตของการแสดงผลลัพธ์

การกำหนดขอบเขตของการแสดงผลลัพธ์จากการตรวจสอบความคล้ายคลึงของซอร์สโค้ด ในรูปแบบของกราฟ ผู้ใช้งานสามารถเลือกขอบเขตร้อยละความคล้ายคลึงกันของชุดข้อมูลซอร์สโค้ดที่ผ่านการตรวจสอบความคล้ายคลึงโดย JPlag ดังภาพที่ 13 และภาพที่ 14



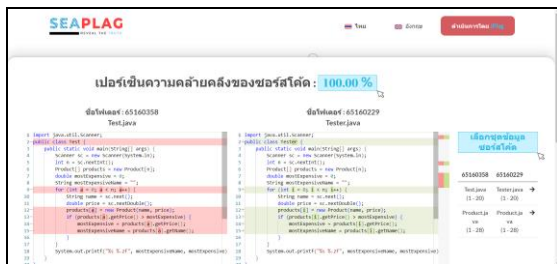
ภาพที่ 13 หน้าจอแสดงกำหนดขอบเขตการแสดงผลลัพธ์ ภายในขอบเขตร้อยละ 50 ถึง 100



ภาพที่ 14 หน้าจอแสดงการกำหนดขอบเขตการแสดงผล
ผลลัพธ์ ภายในขอบเขตร้อยละ 75 ถึง 100

5.5 มอดูลแสดงการเปรียบเทียบซอร์สโค้ดในส่วนที่เหมือนกัน

ผู้ใช้งานสามารถเลือกดูข้อมูลซอร์สโค้ดที่คล้ายคลึงกันทั้งในรูปแบบการแสดงผลแบบกราฟและรูปแบบการแสดงผลผลลัพธ์แบบตาราง ระบบจะแสดงหน้าจอเปรียบเทียบข้อมูลซอร์สโค้ดที่คล้ายคลึงกัน โดยผู้ใช้งานสามารถเลือกชุดข้อมูลซอร์สโค้ดที่ต้องการให้แสดงผลได้ ดังภาพที่ 15



ภาพที่ 15 หน้าจอแสดงการเปรียบเทียบซอร์สโค้ด
ในส่วนที่เหมือนกัน

5.6 ประสิทธิภาพการตรวจสอบความคล้ายคลึงของซอร์สโค้ด

ผู้ดำเนินโครงการได้ทำการวัดประสิทธิภาพการตรวจสอบความคล้ายคลึงของซอร์สโค้ดของระบบ โดยใช้วิธีการจับเวลาเริ่มตั้งแต่ผู้ใช้งานอัปโหลดไฟล์ซอร์สโค้ดไปยังระบบ จนถึงขั้นตอนที่ระบบแสดงผลการตรวจสอบความคล้ายคลึงของซอร์สโค้ดในรูปแบบของกราฟความสัมพันธ์ ผู้ดำเนินโครงการทำการทดลองบนเครื่องโน้ตบุ๊ค Acer Nitro AN515-54 ระบบปฏิบัติการ Windows 11

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

ตารางที่ 1 ประสิทธิภาพการตรวจสอบความคล้ายคลึงของซอร์สโค้ด

จำนวน ไฟล์เดอร์	จำนวนไฟล์ ในแต่ละ ไฟล์เดอร์	จำนวน ไฟล์รวม	เวลา ประมวลผล (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 1 พบว่าประสิทธิภาพการตรวจสอบความคล้ายคลึงของซอร์สโค้ด โดยเริ่มนับตั้งแต่ผู้ใช้งานอัปโหลดไฟล์ซอร์สโค้ด จนกระทั่งระบบแสดงผลการตรวจสอบความคล้ายคลึงของซอร์สโค้ดในรูปแบบตารางความสัมพันธ์ใช้เวลาไม่นาน ซึ่งหากสังเกตจากตารางที่ 1 ในคอลัมน์เวลาประมวลผล (วินาที) พบว่าเวลาที่ใช้ในการประมวลผลของระบบยังคงอยู่ในหน่วยของวินาที แม้จะเพิ่มจำนวนไฟล์เดอร์หรือจำนวนไฟล์ซอร์สโค้ดภายในไฟล์เดอร์นั้นๆ ซึ่งเฉลี่ยแล้วระบบใช้เวลาในการประมวลผลต่างกันอยู่ที่ประมาณ 2 วินาที ซึ่งผู้ดำเนินโครงการสามารถสรุปผลได้ว่าระบบสามารถประมวลผลโดยใช้เวลาไม่มากในการแสดงกราฟความสัมพันธ์ของการตรวจสอบความคล้ายคลึงของซอร์สโค้ด

6. สรุปผลการวิจัย

แนวความคิดการดำเนินโครงการฉบับนี้ เริ่มต้นที่ผู้ดำเนินโครงการได้มองเห็นถึงปัญหาของการคัดลอกซอร์สโค้ด ภายในสถาบันการศึกษาและพบว่าสถาบันการศึกษามีการนำใช้เครื่องมือสำหรับตรวจสอบความคล้ายคลึงของนักศึกษาที่ยังไม่สะดวกต่อการใช้งานหลังจากได้มองเห็นถึงปัญหาที่เกิดขึ้น ผู้ดำเนินโครงการจึงได้

เริ่มคิดค้นระบบที่จะช่วยให้ผู้ใช้งานที่ต้องตรวจสอบความคล้ายคลึงของซอร์สโค้ด สามารถใช้ระบบในการตรวจสอบความคล้ายคลึงของซอร์สโค้ด ได้อย่างสะดวกและสามารถทำความเข้าใจผลลัพธ์ได้โดยใช้เวลาไม่นาน

เมื่อผู้ดำเนินโครงการได้ทำการพัฒนาเสร็จเป็นที่เรียบร้อยแล้วนั้น ผู้ดำเนินโครงการจึงทำการนำระบบมาทวนความต้องการ รวมไปถึงนำมาเทียบกับข้อกำหนดคุณลักษณะซอฟต์แวร์ที่ได้กำหนดไว้ หลังจากนั้นจึงนำมาทดสอบ และปรับแก้ในส่วนที่มีการค้นพบว่ามีความผิดพลาดขึ้น โดยระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอร์สโค้ด (SEAPlag) สามารถตรวจสอบความคล้ายคลึงของซอร์สโค้ดได้ และมีการนำผลลัพธ์ที่ตรวจสอบได้มาแสดงผลบนระบบให้ผู้ใช้งานสามารถมองเห็นภาพได้ชัดเจนและเข้าใจข้อมูลได้ง่าย ซึ่งจะมีการแสดงผลใน 2 รูปแบบคือ รูปแบบของกราฟความสัมพันธ์ระหว่างซอร์สโค้ดที่มีความคล้ายคลึงกันโดยผู้ใช้งานสามารถกำหนดขอบเขตระดับความคล้ายคลึงที่ต้องการได้ และรูปแบบของตารางข้อมูลที่แสดงรายละเอียดของข้อมูลที่ชัดเจน ทั้งนี้ผู้ใช้งานยังสามารถบันทึกภาพกราฟที่แสดงผลบนหน้าจอเข้าสู่เครื่องคอมพิวเตอร์ได้ นอกจากนี้ผู้ใช้งานยังสามารถตรวจสอบความคล้ายคลึงกันของซอร์สโค้ดของชุดข้อมูล 2 ชุดที่มีความคล้ายคลึงมาเปรียบเทียบได้ เมื่อผู้ดำเนินโครงการนั้นได้ทำการพัฒนา และปรับแก้จนแล้วเสร็จ ตลอดจนได้ทำการทดสอบระบบเรียบร้อยแล้วนั้น ผู้ดำเนินโครงการจึงได้ทำการส่งมอบระบบแสดงผลลัพธ์การตรวจสอบความคล้ายคลึงของซอร์สโค้ด (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. [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/setup-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/basicrouting>

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