

# สิ่งแวดล้อมเพื่อการพัฒนาซอฟต์แวร์สำหรับการเขียนโปรแกรมแบบจับคู่แบบทางไกล

ธาราทิพย์ มาลัยมัลย์ ปวริศา ทองเงิน และ ชีรพงศ์ ลีลานุภาพ

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ

Emails: tharathip.malaimarn@gmail.com, pawarisa.thng@gmail.com, teerapong@it.kmitl.ac.th

## บทคัดย่อ

การเขียนโปรแกรมแบบจับคู่เป็นหนึ่งในวิธีการปฏิบัติของเอ็กซ์ทรีมโปรแกรมมิ่งที่มีประสิทธิภาพ ซึ่งโปรแกรมเมอร์ 2 คน มีส่วนร่วมในการพัฒนาซอฟต์แวร์ที่เวิร์คสเตชันเพียงเครื่องเดียว ถึงแม้ว่าการเขียนโปรแกรมแบบจับคู่จะได้รับการยอมรับอย่างกว้างขวางในด้านการเพิ่มคุณภาพและความสามารถในการผลิตซอฟต์แวร์ที่มีประโยชน์ในการนำไปประยุกต์ใช้ในสิ่งแวดล้อมของการพัฒนาซอฟต์แวร์จริง หรือเพื่อส่งเสริมการเรียนรู้ในรายวิชาการเขียนโปรแกรม แต่ถูกจำกัดในเรื่องสถานที่และพฤติกรรมการเขียนโปรแกรม บทความนี้นำเสนอสิ่งแวดล้อมเพื่อการพัฒนาซอฟต์แวร์สำหรับการเขียนโปรแกรมแบบจับคู่แบบทางไกล ที่ถูกพัฒนาขึ้นเป็นลักษณะของเว็บแอปพลิเคชัน ซึ่งทำให้ผู้เขียนโปรแกรมทั้งคู่สามารถทำงานร่วมกันได้จากทางไกลเสมือนทำงานอยู่บนเวิร์คสเตชันเดียวกัน ระบบมีความสามารถที่จำเป็นมากมายที่ถูกพัฒนามาตามหลักการการเขียนโปรแกรมแบบจับคู่ เช่น ส่วนการแสดงผลหน้าสำหรับเขียนโปรแกรมของผู้เขียนโปรแกรมไปยังหน้าจอของผู้ตรวจทานโค้ด ส่วนแสดงผลผลลัพธ์ของโปรแกรม ช่องทางการสื่อสารเสมือนอยู่ต่อหน้ากัน เช่น วิดีโอการสนทนาระหว่างบุคคล ระบบส่งข้อความแบบทันที การสลับบทบาทด้วยตนเองและแบบอัตโนมัติ การวิเคราะห์คุณภาพของโค้ดเพื่อแนะนำผู้เขียนโปรแกรมที่มีทักษะใกล้เคียงกัน การตรวจทานโค้ดทั้งแบบโดยนัยและแบบชัดเจนสำหรับติดตามการมีส่วนร่วมของผู้ตรวจทาน

**คำสำคัญ**-- การเขียนโปรแกรมแบบจับคู่; เอ็กซ์ทรีมโปรแกรมมิ่ง; การตรวจทานโค้ด; การวิเคราะห์คุณภาพโค้ด

## Abstract

Pair Programming (PP) is one of the effective Extreme Programming practices, by which two programmers participate in one software development effort at one workstation. Although PP has been widely accepted for an increment of software quality and productivity, its benefits in applying this method to real-world software development environments or to programming courses for encouraging students' learning are often limited due to location constraints and coding habits. In this paper, we present the development of a remote software development environment for pair programming. The environment is developed as a web application, allowing two programmers to remotely work together just as they are working at a single workstation. The system has

many necessary features complying with the principles of PP, such as mirroring the display of code editor of a driver to the screen of a navigator, output terminal, face-to-face like communication channels (i.e., video calling and instant text messaging), automatic and manual role switching, code quality analysis for suggesting a pair of programmers with similar skills, implicit and explicit code review for tracking a navigator's engagement, etc.

**Keywords**-- Pair Programming; Extreme Programming; Code Review; Code Quality Analysis

## 1. บทนำ

การเขียนโปรแกรมร่วมกันเป็นแนวปฏิบัติที่ดีในการการพัฒนาซอฟต์แวร์เพื่อให้เกิดประสิทธิภาพสูงสุดและทำให้นักพัฒนาได้พัฒนาตนเองควบคู่ไปกับการทำงาน ซึ่งเทคนิคที่มักนำมาใช้นั้น คือ การเขียนโปรแกรมแบบจับคู่ (Pair Programming) โดยเทคนิคนี้เป็นส่วนหนึ่งของแนวคิดในการพัฒนาซอฟต์แวร์แบบเอ็กซ์ทรีมโปรแกรมมิ่ง ซึ่งเป็นส่วนหนึ่งของการพัฒนาซอฟต์แวร์แบบอไจล์ (Agile Development) แม้การเขียนโปรแกรมแบบจับคู่จะใช้เวลาในการเขียนโปรแกรมช่วงแรกในการเริ่มเขียนโปรแกรมมากกว่าการเขียนแบบคนเดียว แต่กลับช่วยลดความผิดพลาดที่เกิดจากการเขียนโปรแกรมและใช้เวลาในการแก้ไขภายหลังน้อยลง นำไปสู่โปรแกรมที่มีคุณภาพและทำให้เกิดการแลกเปลี่ยนเรียนรู้ซึ่งกันและกันภายในทีม [1], [2]

จากการศึกษาทดลองเขียนโปรแกรมแบบจับคู่แล้ว มีรายงานว่ายังคงมีปัญหที่เกิดขึ้นในเรื่องของความไม่ยืดหยุ่นของสถานที่ที่จำเป็นต้องเขียนโปรแกรมที่โต๊ะเดียวกันอยู่ ซึ่งหากอยู่ไกลกันก็ไม่สามารถที่จะเขียนโปรแกรมแบบจับคู่ได้ อีกทั้งการเขียนโปรแกรมแบบจับคู่จะมีประสิทธิภาพหรือไม่ขึ้นอยู่กับผู้ของนักพัฒนาที่ได้จับคู่กันด้วย เช่น ทักษะความรู้ที่มีของแต่ละคนและรูปแบบการเขียนโค้ด ซึ่งหากปัจจัยเหล่านี้ต่างกันมาก ๆ ก็จะทำให้ผลของการเขียนโปรแกรมแบบจับคู่ออกมาไม่มีประสิทธิภาพ ส่งผลให้ต้นทุนความสับสนและความเชื่อมั่นในการเขียนโปรแกรม เกิดความเบื่อหน่ายและไม่เอื้อต่อการเรียนรู้ของนักพัฒนาทั้งคู่ที่อาจได้จากการแลกเปลี่ยนระหว่างเขียนโปรแกรมร่วมกัน รวมทั้งการแลกเปลี่ยนความคิดเห็น เทคนิค และวิธีใหม่ ๆ ซึ่งส่งผลต่อทีมพัฒนา

เมื่อพบปัญหาข้างต้น ผู้พัฒนาจึงมีแนวคิดที่จะพัฒนาสิ่งแวดล้อมเพื่อการพัฒนาซอฟต์แวร์สำหรับการเขียนโปรแกรมแบบจับคู่แบบทางไกล เพื่อให้สามารถเขียนโปรแกรมแบบจับคู่ได้ แม้อยู่ไกล

กันไม่ว่าที่ใดก็ตาม และแก้ปัญหาในเรื่องของทักษะของนักพัฒนาที่ต่างกันมากเกินไป ด้วยการวิเคราะห์คุณภาพโค้ดเพื่อนำมาใช้ในการให้คะแนนและแนะนำการจับคู่ของนักพัฒนาที่ควรเขียนโปรแกรมร่วมกัน และเพิ่มส่วนที่ช่วยในการตรวจทานโค้ดให้มีประสิทธิภาพมากขึ้น

## 2. ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องจะเกี่ยวกับแนวทางการปฏิบัติ กระบวนการการทำงานที่เกี่ยวข้องกับการเขียนโปรแกรมแบบจับคู่ ซึ่งมีดังต่อไปนี้

### 2.1. เอ็กซ์ทรีมโปรแกรมมิ่ง (Extreme Programming)

เป็นแบบจำลองของวิธีการการพัฒนาซอฟต์แวร์แบบบอจล์ ที่ช่วยเพิ่มคุณภาพของซอฟต์แวร์ โดยมุ่งเน้นไปที่ซอฟต์แวร์ที่ทำงานได้จริง การสร้างความพึงพอใจให้กับลูกค้า การทำงานร่วมกัน และการปฏิสัมพันธ์กันระหว่างทีมพัฒนานักพัฒนาได้ส่วนเสียกับระบบ

เอ็กซ์ทรีมโปรแกรมมิ่งเป็นวิธีที่มีความยืดหยุ่นมาก สามารถปรับเปลี่ยนแผนงานให้เป็นไปตามความต้องการของผู้ใช้ได้ตลอดเวลา ซึ่งกระบวนการของเอ็กซ์ทรีมโปรแกรมมิ่งมีดังนี้ คือ วางแผน ออกแบบ เขียนโค้ด และทดสอบ ในขั้นตอนการเขียนโค้ดนั้น จะมักจะใช้แนวทางการปฏิบัติที่นิยมกันมากก็คือ การเขียนโปรแกรมแบบจับคู่ (Pair Programming) เพื่อช่วยเพิ่มประสิทธิภาพของการเขียนโปรแกรม [3]

### 2.2. การเขียนโปรแกรมแบบจับคู่ (Pair Programming)

การเขียนโปรแกรมแบบจับคู่หรือ Pair programming เป็นการเขียนโปรแกรมร่วมกันระหว่างผู้เขียนโปรแกรม 2 คน โดยบทบาทมีดังนี้ [4]

1. Driver หรืออาจเรียกว่า Coder จะเป็นผู้คิดและเขียนโค้ดลงบนคอมพิวเตอร์
2. Navigator หรืออาจเรียกว่า Reviewer จะเป็นคนที่นั่งข้าง ๆ Driver และจดจ่ออยู่กับการดูโค้ดที่ Driver เขียน เพื่อคอยตรวจสอบว่าโค้ดมีข้อผิดพลาดหรือไม่ ต้องแก้ไขอะไรหรือไม่

โดยนักพัฒนาจับคู่กันและร่วมกันเขียนโปรแกรมซึ่งจะมีหลักการดังต่อไปนี้

1. Driver และ Navigator ต้องสลับหน้าที่กันทุก ๆ 15 นาที
2. คู่เขียนโปรแกรมต้องอยู่หน้าคอนโซลเดียวกัน ใช้หน้าจอเมาส์ และคีย์บอร์ดเดียวกัน
3. คู่เขียนโปรแกรมควรจะมีทักษะการเขียนโปรแกรมที่ใกล้เคียงกัน

การจับคู่กันในการเขียนโปรแกรมแบบจับคู่กันนั้น มีการแบ่งประเภทการจับคู่จากประสบการณ์ของผู้เขียนโปรแกรมแต่ละคนที่มาจับคู่กันออกเป็น 3 ประเภท [5] ดังนี้

1. Expert – Expert (ผู้เชี่ยวชาญกับผู้เชี่ยวชาญ) การจับคู่แบบนี้เหมาะกับการเขียนโปรแกรมที่ต้องการแก้ปัญหาที่ซับซ้อนหรือเป็นงานในส่วนที่วิกฤต มีความรวดเร็วในการพัฒนามากที่สุดใน 3 แบบ เนื่องจากไม่ต้องการอธิบายงานในส่วนที่สำเร็จเพราะทั้งคู่มีความชำนาญอยู่แล้ว
2. Expert – Novice (ผู้เชี่ยวชาญกับมือใหม่) เป็นการจับคู่ที่แนะนำสำหรับผู้เขียนโปรแกรมมือใหม่ การจับคู่แบบนี้จะช่วย

ลดระยะเวลาในการฝึกฝนของมือใหม่ได้ วิธีนี้อาจจะดูเป็นการเสียเวลาสำหรับผู้เขียนโปรแกรมที่เชี่ยวชาญ แต่ก็ยังมีข้อดี เพราะบางครั้งมือใหม่ก็อาจจะช่วยผู้เชี่ยวชาญหาข้อผิดพลาดโดยการถามเกี่ยวกับเมธอดของโปรแกรมนั้น ๆ อย่างไรก็ตาม สิ่งที่สำคัญก็คือผู้เชี่ยวชาญจะต้องมีความอดทน มีความยินดีที่จะสอนมือใหม่ มิฉะนั้นการจับคู่แบบนี้ก็จะได้ประโยชน์อะไรเลย ปัญหาอื่น ๆ ที่อาจเกิดขึ้นกับการจับคู่แบบนี้คือ ความมีทริคของผู้เขียนโปรแกรมที่เชี่ยวชาญเขาจะรู้สึกว่าเขาดีกว่าคู่เขียนโปรแกรมของเขา ทำให้เกิดบรรยากาศในการทำงานที่ไม่ดี ส่วนผู้เขียนโปรแกรมมือใหม่อาจเกิดความไม่มั่นใจ และให้ความร่วมมือน้อยลง ทำให้การเขียนโปรแกรมแบบจับคู่กลายเป็นการเขียนโปรแกรมแบบเดี่ยวได้

3. Novice – Novice (มือใหม่กับมือใหม่) การจับคู่กันระหว่างมือใหม่กับมือใหม่เป็นเทคนิคที่ดีที่ทำให้เกิดประสบการณ์การเรียนรู้ในการเขียนโปรแกรมไปสู่ทั้งคู่ หากคนหนึ่งมีความรู้ที่แตกต่างจะทำให้เกิดการเติมเต็มความรู้ที่อีกคนยังไม่มี การจับคู่แบบนี้ทำเพื่อลดเวลาของผู้ดูแล แต่ยังคงต้องการผู้ดูแล

## 3. การวิเคราะห์และออกแบบระบบ

### 3.1. การวิเคราะห์ระบบงานเดิม

จากการศึกษาเพื่อวิเคราะห์ระบบงานเดิม โดยใช้กรณีศึกษาของคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งมีการสอนทักษะการเขียนโปรแกรมแบบจับคู่ในวิชา Problem Solving in Information Technology โดยใช้ภาษา Python ผู้พัฒนาพบว่า เมื่อถึงชั่วโมงเรียนปฏิบัติที่จะต้องเขียนโปรแกรมแบบจับคู่กัน พบปัญหาต่าง ๆ ดังนี้

#### 1. ปัญหาการจับคู่เขียนโปรแกรมของนักศึกษาที่ไม่เหมาะสม

เกิดจากการสุ่มของระบบ ไม่มีการจับคู่ตามทักษะของแต่ละบุคคล บางครั้งส่งผลให้ทักษะของคู่เขียนโปรแกรมนั้นต่างกันมากเกินไป ทำให้คู่เขียนโปรแกรมอาจจะไม่เข้าใจโค้ดที่อีกคนเขียน ผู้เขียนโปรแกรมอาจต้องใช้เวลาในการอธิบายมาก จนเกิดความไม่พอใจและรู้สึกที่ไม่อยากเขียนโปรแกรมแบบจับคู่อีก ในขณะที่เดียวกันผู้ที่ถูกอธิบายก็อาจเกิดความรู้สึกท้อแท้ เกิดความไม่สบายใจ ถึงแม้บางครั้งอาจได้ความรู้ใหม่แต่กลับไม่เข้าใจ จนรู้สึกหงุดหงิดและหมดกำลังใจได้

#### 2. ปัญหาเรื่องข้อจำกัดการเขียนโปรแกรมแบบจับคู่ที่ต้องอยู่

สถานที่เดียวกัน ในชั่วโมงเรียนมีเวลาทำโจทย์กับคู่เขียนโปรแกรมแค่ระยะเวลาสั้น ๆ และหลังจากหมดเวลาเรียน การนัดพบกับคู่เขียนโปรแกรมเพื่อทำโจทย์ต่อก็ทำได้ลำบาก เนื่องจากไม่ได้อยู่ ณ สถานที่เดียวกัน ส่งผลให้ผู้เรียนกลับไปฝึกทำโจทย์โดยใช้วิธีการการเขียนโปรแกรมแบบจับคู่ไม่ได้

#### 3. ปัญหา Navigator ไม่ตรวจทานโค้ด เมื่อถึงเวลาที่อีกคนทำหน้าที่เป็น Driver

โดยขณะเขียนโปรแกรม ผู้เป็น Navigator จะต้อง

คอยตรวจทานโค้ดของ Driver ไปด้วย แต่ Navigator มักไม่สนใจที่จะตรวจทานโค้ด ไม่ให้ความคิดเห็นต่าง ๆ ทำให้โค้ดที่ได้ยังคงมีข้อผิดพลาด

**4. ปัญหาการลิมจับเวลาในแต่ละช่วงเพื่อเตรียมสลับหน้าที่ระหว่าง Driver กับ Navigator** เนื่องจากในขณะที่เขียนโปรแกรมแบบจับคู่กัน Driver ต้องเขียนโค้ด ในขณะที่ Navigator ก็ต้องตรวจทานตามไปด้วย ทำให้ทั้งคู่จ้องอยู่กับที่การเขียนโปรแกรม จนลิมจับเวลาและลิมสลับบทบาทกันเมื่อครบเวลา

**5. ปัญหาการเก็บประวัติการแก้ไขโค้ดและการสนทนา** เมื่อเขียนโปรแกรมแบบจับคู่แล้วสลับบทบาทไปอย่างต่อเนื่อง ทำให้ไม่สามารถกลับมาดูประวัติการเขียนโค้ดได้ว่า ใครเป็นคนแก้ส่วนใด หากต้องการย้อนกลับไปที่โค้ดส่วนเดิมที่เคยเขียนไว้ รวมทั้งบทสนทนาต่าง ๆ ที่เกิดในระหว่างเขียนโค้ดก็ไม่สามารถทำได้

### 3.2. การออกแบบระบบสิ่งแวดล้อมเพื่อเขียนโปรแกรมแบบจับคู่

#### 3.2.1. เป้าหมายการพัฒนาความสามารถของระบบ

ผู้พัฒนาจึงได้กำหนดเป้าหมายในการพัฒนาระบบสิ่งแวดล้อมเพื่อการพัฒนาซอฟต์แวร์สำหรับการเขียนโปรแกรมแบบจับคู่แบบทางไกลขึ้นมา ในรูปแบบของเว็บแอปพลิเคชันที่สามารถเข้าใช้งานได้จากทุกที่ทุกเวลา โดยผู้เขียนโปรแกรมสามารถเข้าสู่ระบบเพื่อสร้างโปรเจกต์และเพิ่มคู่เขียนโปรแกรมได้จากการส่งคำเชิญผู้ใช้ของคู่เขียนโปรแกรมด้วยตนเองหรือเลือกคู่เขียนโปรแกรมจากที่ระบบแนะนำให้ ซึ่งระบบจะทำการวิเคราะห์คะแนนของคู่เขียนโปรแกรมที่เหมาะสมจากโค้ดที่คนนั้น ๆ ได้เคยเขียนไว้ก่อนหน้านี้ โดยวิเคราะห์คุณภาพของโค้ดจาก Coding Standard ตามข้อแนะนำของภาษา Python ที่เรียกว่า PEP8 เช่น White space, Naming, Documentation String เป็นต้น นอกจากนั้นยังวิเคราะห์จาก Error Detection และ Duplicated Code และจะแนะนำเฉพาะรายชื่อผู้ใช้งานที่เหมาะสม

หลังจากสร้างโปรเจกต์ คำขอร่วมเข้าร่วมโปรเจกต์จะถูกส่งไปยังผู้เขียนโปรแกรมอีกฝ่าย เมื่ออีกฝ่ายทำการตอบรับ ผู้เขียนโปรแกรมทั้งคู่จะสามารถเข้าสู่หน้าพื้นที่การพัฒนาโปรแกรมที่อยู่บนเว็บไซต์ได้ทันที ซึ่งผู้ที่เข้าสู่หน้าพื้นที่การพัฒนาโปรแกรมก่อนจะได้สิทธิ์ในการเลือกบทบาทของตนเอง ส่วนผู้ใช้งานที่เข้าภายหลังจะได้บทบาทตรงข้าม ผู้ที่เป็น Driver จะสามารถเขียนโค้ดบน Editor ได้ ดูความคิดเห็นที่ Navigator ส่งมาได้ พร้อมกับสามารถรันโค้ดเพื่อดูผลลัพธ์ของโปรแกรมที่เขียนได้ ซึ่งทั้งคู่จะเห็นโค้ดที่กำลังพิมพ์และผลลัพธ์ที่ถูกรันไปพร้อม ๆ กัน บนหน้าจอของตนเองที่แสดงส่วนเขียนโปรแกรมเดียวกัน ในส่วนของ Navigator นั้นจะไม่สามารถพิมพ์โค้ดใน Editor ได้ แต่สามารถแสดงความคิดเห็นต่อโค้ดแต่ละบรรทัด ซึ่งถือเป็นการตรวจทานแบบซัดแจ้ง และระบบจะตรวจสอบว่าโค้ดที่ Driver ได้เขียนนั้นถูกตรวจทานแล้วจริงแบบโดยนัย ถ้า Navigator ได้นั่งอยู่ที่หน้าจอในขณะนั้นเป็นเวลาอย่างน้อย 2 นาที หากไม่เพิ่มความเห็นจะถือว่าโค้ดในส่วนนั้นได้ถูกตรวจทานเรียบร้อยแล้ว ระบบจะทำการสลับบทบาทอัตโนมัติเมื่อ

ครบตามเวลาที่เลือกไว้ เช่น สลับบทบาททุก ๆ 15 นาทีหรืออาจสลับบทบาทด้วยตนเองก็ได้ และเว็บแอปพลิเคชันนี้ยังสามารถติดต่อสื่อสารกันผ่านการส่งข้อความแบบทันทีและมีวิธีการสนทนาระหว่างบุคคลเพื่อจำลองให้เสมือนว่าผู้เขียนโปรแกรมทั้งคู่อยู่ต่อหน้ากัน ทำให้เกิดการแลกเปลี่ยนเรียนรู้ร่วมกัน

จากข้างต้น จึงได้เป้าหมายการพัฒนาความสามารถหลักของระบบ ดังตารางที่ 1. ดังนี้

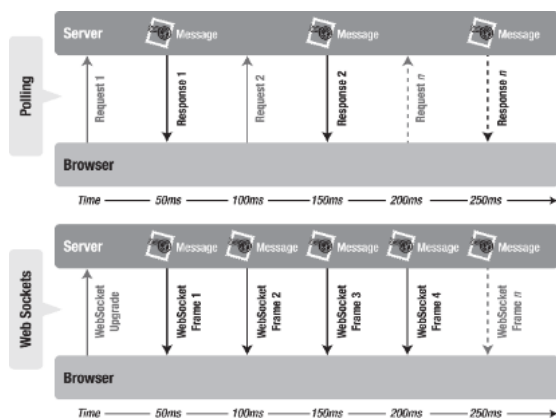
ตาราง 1. แสดงความสามารถหลักของระบบ

| รายการที่ | ความสามารถของระบบ                                                                                                                                   |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 1         | วิเคราะห์คุณภาพโค้ดที่ผู้เขียนโปรแกรมเคยเขียนไว้จาก PEP8 ซึ่งเป็น Coding Standard ของภาษา Python                                                    |
| 2         | แนะนำคู่เขียนโปรแกรมจากคะแนนการวิเคราะห์คุณภาพโค้ด                                                                                                  |
| 3         | ตรวจสอบการทำหน้าที่ของ Navigator ในส่วนการตรวจทานโค้ดแบบโดยนัย ซึ่งจะดูว่า Navigator ได้นั่งดูโค้ดที่หน้าจอในขณะนั้นเป็นเวลาอย่างน้อย 2 นาทีหรือไม่ |
| 4         | Navigator สามารถทำการตรวจทานโค้ดแบบซัดแจ้งได้ โดยเป็นความเพิ่มความเห็นลงไปในโค้ดแต่ละบรรทัด                                                         |
| 5         | Driver สามารถเขียนและรันโค้ดได้                                                                                                                     |
| 6         | Driver และ Navigator สามารถส่งข้อความแบบทันทีและสนทนาผ่านวิธีการสนทนาระหว่างบุคคลได้                                                                |
| 7         | สามารถสลับบทบาทระหว่าง Driver และ Navigator ได้ทั้งแบบอัตโนมัติเมื่อครบตามช่วงเวลาหรือสลับด้วยตนเอง                                                 |
| 8         | สามารถจัดเก็บและแสดงประวัติการแก้ไขโค้ดได้                                                                                                          |

#### 3.2.2. เทคโนโลยีที่ใช้

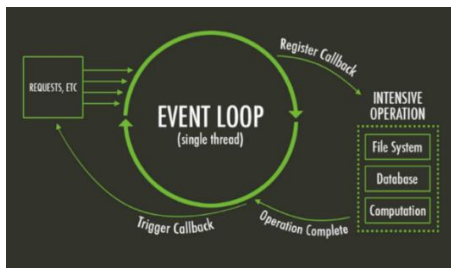
จากเป้าหมายการพัฒนาความสามารถของระบบ ผู้พัฒนาจึงเลือกเครื่องมือเหล่านี้มาพัฒนา ซึ่งมีดังต่อไปนี้

**1. WebSocket** - เป็น Protocol ที่ถูกระบุใน RFC6455 ใช้สำหรับติดต่อสื่อสารระหว่าง Server กับ Client แบบสองทิศทาง (bidirectional communication) ที่เป็น full-duplex communications channel คือมีทิศทางการส่งข้อมูลที่ส่งพร้อม ๆ กันได้สองทิศทางในเวลาเดียวกัน Server จะคอยตรวจสอบสถานะของ Client และทำการส่งข้อมูลไปยัง Client ได้ทันทีเมื่อเกิดการเปลี่ยนแปลง โดยที่ Client ไม่จำเป็นต้องส่ง Request ไปหา Server เหมาะกับระบบที่ต้องการการตอบสนองอย่างทันที โดยเทคโนโลยีนี้จะช่วยเพิ่มความยืดหยุ่นกว่าแบบ Client-Server หรือแบบ HTTP ดังรูปที่ 1. ที่ Client จะต้องทำการส่ง Request ไปยัง Server ก่อน และ Server ถึงจะส่ง Response กลับมาให้ Client [6]



รูปที่ 1. แสดงความแตกต่างของการทำงานระหว่าง HTTP กับ WebSocket (ที่มา <http://apress.jensimmons.com/v5/pro-html5-programming/ch7.html>)

**2. Node.js** - เป็น Cross Platform Javascript Environment สำหรับประมวลผลคำสั่งที่เขียนด้วยภาษา Javascript ทางฝั่ง Server โดยใช้สถาปัตยกรรมแบบขับเคลื่อนโดยเหตุการณ์ (Event-driven development) ที่จะมีลักษณะการทำงานแบบตอบสนองต่อเหตุการณ์ (event) ต่าง ๆ ที่เกิดขึ้นตามที่ถูกโปรแกรมไว้ โดยใช้ event loop ที่ทำงานบน Thread เดียวคอยวนเพื่อรอรับเหตุการณ์อยู่ตลอดเวลา ดังรูปที่ 2. อีกทั้ง Node.js ยังเป็น Non Blocking I/O ที่ใช้ Asynchronous I/O ทำให้ไม่ต้องหยุดรอให้ทำ I/O (Input and Output เช่น การอ่านและการเขียนไฟล์) เสร็จก่อน ก็สามารถทำงานต่อไปได้ไม่เกิดการขัดขวางการทำงาน (blocking) ทำให้สามารถข้ามไปทำส่วนอื่นต่อก่อนได้โดยไม่ต้องรอผลลัพธ์ โดยค่าที่ได้จะส่งกลับมาพร้อมกับ callback function ในภายหลัง [7]



รูปที่ 2. ลักษณะการทำงานแบบ Event-driven ของ Node.js (ที่มา <http://webmpires.net/index.php/blog/all-you-need-to-know-about-nodejs-as-a-beginner>)

**3. Socket.io** - เป็น Javascript Library สำหรับเรียลไทม์แอปพลิเคชันที่มีการสื่อสารแบบสองทาง (bidirectional communication) ระหว่าง Client กับ Server โดยมีการแบ่งออกเป็น 2 ส่วน คือ library ทางฝั่ง Client ที่จะทำงานบน browser และ library ทางฝั่ง Server สำหรับใช้งานกับ Node.js โดยใช้ WebSocket Protocol ในการเชื่อมต่อค้างเอาไว้เพื่อเป็นช่องทางในการรับส่งข้อมูล

**4. Express.js** - เป็น framework ของ Node.js ที่ช่วยในการทำเว็บแอปพลิเคชัน ในส่วนการทำงานพื้นฐาน เช่น การติดต่อฐานข้อมูลต่าง ๆ เช่น MongoDB หรือ MySQL เก็บ cookie เก็บสถานะการทำงาน (session) ช่วยในการทำ RESTful (Representational State Transfer) [8]

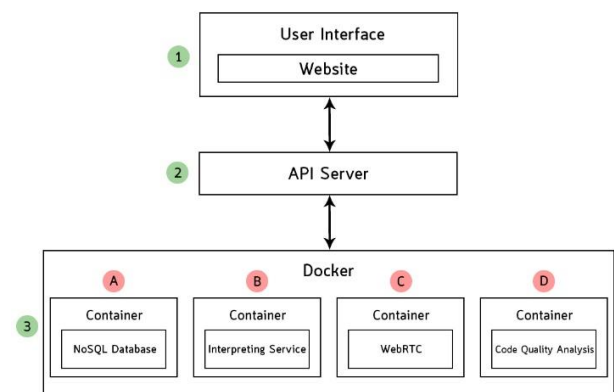
**5. MongoDB** - เทคโนโลยีฐานข้อมูลที่เป็นแบบ NoSQL ซึ่งใช้มากในเรียลไทม์เว็บแอปพลิเคชัน จะทำการบันทึกข้อมูลแบบ Document ในรูปแบบไฟล์นามสกุล JSON (JavaScript Object Notation) รองรับการทำงานที่มีการบันทึกข้อมูลในปริมาณมาก ๆ หรือเรียกดูข้อมูลบ่อย ๆ

**6. Redis** - เป็นระบบเก็บข้อมูลแบบกึ่งโครงสร้างที่ทำงานบนหน่วยความจำหลัก (in-memory data structure store) จึงทำงานได้เร็ว มักนิยมถูกนำมาใช้เป็น caching และฐานข้อมูลชั่วคราว ซึ่งสามารถรองรับโครงสร้างข้อมูลได้หลายรูปแบบ เช่น string, hashes, lists, set และ sorted set โดยมีความสามารถในการทำ publish/subscribe ทำให้เหมาะกับการนำมาใช้ในเรียลไทม์แอปพลิเคชัน

**7. Pylint** - เป็นเครื่องมือที่ใช้ในการตรวจสอบคุณภาพและให้คะแนนโค้ด ที่เขียนด้วยภาษา Python โดยดูจาก Coding Standard, Error detection และ Duplicated code ซึ่งจะอ้างอิงตามข้อเสนอแนะการเขียนโปรแกรม PEP8 [9]

**8. WebRTC.js** - เป็นเทคโนโลยี ที่สร้าง Protocol การสื่อสารแบบเรียลไทม์ ช่วยในการใช้งานทางด้านต่าง ๆ เช่น การส่งต่อข้อความแบบทันที วิดีโอสนทนา การโอนย้ายไฟล์ หรือการแชร์หน้าจอระหว่างกัน

### 3.2.3. โครงสร้างการทำงานของระบบ



รูปที่ 3. โครงสร้างการทำงานทั้งหมดของระบบ

สถาปัตยกรรมและโครงสร้างของระบบจะประกอบไปด้วย 3 ส่วน ดังรูปที่ 3. ซึ่งมีรายละเอียดดังต่อไปนี้

#### 1. User Interface (ส่วนต่อประสานกับผู้ใช้)

ทำหน้าที่เชื่อมต่อเพื่อให้เกิดการสื่อสารระหว่างผู้ใช้งานกับระบบ สามารถรับข้อมูลหรือคำสั่งจากผู้ใช้งานและแสดงผลลัพธ์ให้ผู้ใช้งานได้ทราบ โดยออกแบบให้อยู่ในรูปแบบของเว็บแอปพลิเคชัน ที่ทำงานเชื่อมต่อกับ API Server

#### 2. API Server (เครื่องแม่ข่ายส่วนต่อประสานโปรแกรมประยุกต์)

ถูกพัฒนาให้รองรับการเรียกใช้บริการจากผู้ใช้งาน เป็นส่วนรับ Request ที่มาจาก Client ผ่านทาง User Interface ที่ออกแบบในลักษณะ RESTful (Representational State Transfer) ซึ่งจะมีระบบรักษาความปลอดภัยในการเข้าถึงจากภายนอกที่ไม่ได้ถูกเรียกบริการโดยระบบเอง

#### 3. Docker

ทำหน้าที่สร้างสภาพแวดล้อมของ Application Service ต่าง ๆ ที่พัฒนาขึ้น เพื่อให้บริการ service โดยจำลอง container ขึ้นมาทั้งหมด

4 container ซึ่งแต่ละ container จะให้บริการเพียงแค่ 1 service เท่านั้น โดยมี service ดังต่อไปนี้

- NoSQL Database จะใช้งานฐานข้อมูล MongoDB ที่เก็บข้อมูลในไฟล์ JSON เพื่ออำนวยความสะดวกในการใช้งานในระบบเรียลไทม์รองรับการอัปเดตและอ่านข้อมูลที่เกิดขึ้นตลอดเวลา
- Interpreting Service ส่วนบริการสั่งทำงานโค้ดที่ผู้ใช้เขียนขึ้นมาในระบบ และแสดงผลการทำงานของโค้ด
- WebRTC เป็นส่วนที่ใช้ในการติดต่อสื่อสารระหว่าง Driver และ Navigator ทั้งวิธีการสนทนาและการส่งข้อความแบบทันที
- Code Quality Analysis สำหรับการประมวลผลและวิเคราะห์คุณภาพโค้ด ด้วย Pylint ที่เป็นเครื่องมือวิเคราะห์คุณภาพโค้ดภาษา Python ตามข้อแนะนำการเขียนโปรแกรม PEP8

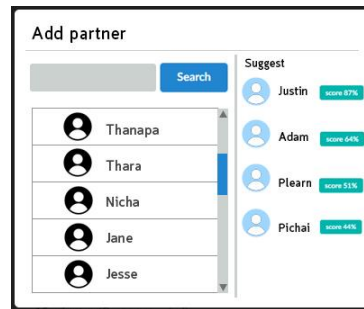
## 5. ผลการพัฒนา

จากการพัฒนาสิ่งแวดล้อมเพื่อการพัฒนาซอฟต์แวร์สำหรับการเขียนโปรแกรมแบบจับคู่แบบทางไกลเป็นรูปแบบเว็บแอปพลิเคชันแล้ว ได้ผลการพัฒนา ดังนี้

หลังจากผู้ใช้งานเข้าสู่ระบบ จะมายังหน้า Dashboard เป็นส่วนที่แสดงโปรเจกต์ที่มีทั้งหมด พร้อมทั้งข้อมูลการแก้ไขโค้ดล่าสุด โดยผู้ใช้งานสามารถไปตั้งค่าต่าง ๆ ในแต่ละโปรเจกต์ได้ในส่วนของด้านบน สามารถรับการแจ้งเตือนต่าง ๆ ได้ เช่น การแจ้งเตือนเมื่อมีผู้ใช้งานคนอื่นสร้างโปรเจกต์และเพิ่มตนเองเป็นผู้ร่วมโปรเจกต์ด้วย หรือการแจ้งเตือนสำหรับคำขอเพื่อรอการอนุมัติสำหรับเชิญเข้าร่วมโปรเจกต์

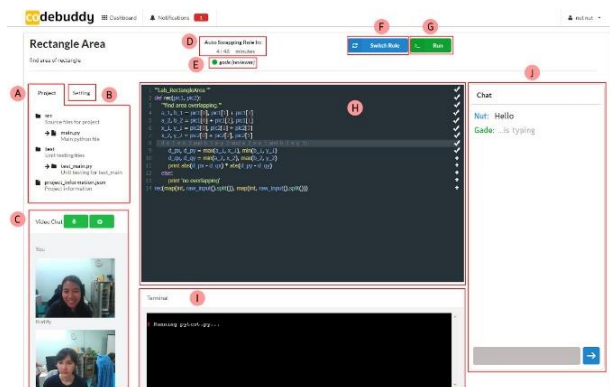
ผู้ใช้งานสามารถสร้างโปรเจกต์เพื่อใช้ในการเขียนโปรแกรมได้ดังรูปที่ 4. โดยต้องกรอกข้อมูลที่เป็นเกี่ยวกับโปรเจกต์ ได้แก่ ชื่อโปรเจกต์ คำอธิบายโปรเจกต์ ภาษาที่ต้องการเขียน ช่วงเวลาที่ต้องการใช้ในการสลับบทบาทของการเขียนโปรแกรมแบบจับคู่ และผู้เขียนโปรแกรมที่จับคู่ด้วย โดยสามารถเพิ่มได้ 2 แบบ คือ การเพิ่มจากการใส่ชื่อผู้ใช้ของผู้ที่ต้องการให้เข้าร่วมเพื่อส่งคำเชิญหรือการเพิ่มโดยเลือกจากผู้เขียนโปรแกรมที่เหมาะสมที่ระบบแนะนำจากการวิเคราะห์โค้ด โดยนำโค้ดที่ผู้เขียนโปรแกรมเคยเขียนไว้เข้าสู่ Pylint เพื่อให้คะแนนในด้านต่าง ๆ แต่เนื่องจากคะแนนในบางด้าน อาจไม่เกี่ยวข้องกับคุณภาพของการเขียนโค้ดที่แท้จริง ทำให้ระบบต้องนำคะแนนด้านต่าง ๆ มาให้น้ำหนักอีกครั้งหนึ่งจนได้เป็นคะแนนสุดท้าย จากนั้นจึงนำมาจัดอันดับ (Ranking) เพื่อแนะนำผู้เขียนโปรแกรมที่เหมาะสม ดังรูปที่ 5.

รูปที่ 4. หน้ากรอกรายละเอียดในการสร้างโปรเจกต์ใหม่



รูปที่ 5. หน้าสำหรับเลือกผู้เขียนโปรแกรมที่ระบบแนะนำโดยการวิเคราะห์โค้ด

เมื่อเข้าสู่หน้าหลักสำหรับการเขียนโปรแกรม ผู้ที่เข้าสู่พื้นที่การพัฒนาโปรแกรมก่อนจะได้สิทธิ์ในการเลือกบทบาทสำหรับการเขียนโปรแกรมแบบจับคู่ในช่วงแรกระหว่าง Driver หรือ Navigator ส่วนผู้ที่เข้าตามมาจะได้บทบาทที่ไม่ถูกเลือก



รูปที่ 6. หน้าแสดงพื้นที่พัฒนาโปรแกรมร่วมกัน<sup>1</sup>

จากรูปที่ 6. เป็นส่วนของหน้าจอหลักที่ใช้สำหรับการเขียนโปรแกรมแบบจับคู่โดยแบ่งออกเป็น ส่วน ๆ ดังนี้

- ส่วนของการแสดงโครงสร้างของโปรเจกต์ เป็นส่วนแสดงไฟล์ต่าง ๆ ในโปรเจกต์
- ส่วนเมนู Setting จะอนุญาตให้ผู้เลือกใช้สลับสีพื้นหลังระหว่างสีโทนมืดและโทนสว่าง
- ส่วนของวิดีโอสนทนาระหว่างบุคคล โดยจะแสดงภาพสถานะการทำงานของผู้เขียนโปรแกรมทั้งคู่ในขณะนั้น พร้อมเสียงสนทนา
- ส่วนแสดงเวลานับถอยหลังเพื่อเปลี่ยนหน้าที่ของผู้เขียนโปรแกรม (Driver หรือ Navigator)
- ส่วนแสดงสถานะการออนไลน์ของผู้เขียนโปรแกรมอีกคนที่กำลังจับคู่กับตนเอง โดยแสดงชื่อ หน้าโปรไฟล์นั้น (Driver หรือ Navigator) และวงกลมแสดงสถานะ มี 2 สถานะ ได้แก่
  - สีเขียว แสดงถึง การออนไลน์
  - สีเทา แสดงถึง การออฟไลน์
- ปุ่มสำหรับสลับหน้าที่ระหว่างผู้เขียนโปรแกรมด้วยตนเอง
- ปุ่มสำหรับส่งรหัสโค้ด

<sup>1</sup> สำหรับผู้ที่สนใจสามารถดูรูปภาพขนาดใหญ่ได้ที่

[http://www.it.kmitl.ac.th/~teerapong/ncit2017/NCIT17\\_PairProg.jpg](http://www.it.kmitl.ac.th/~teerapong/ncit2017/NCIT17_PairProg.jpg)

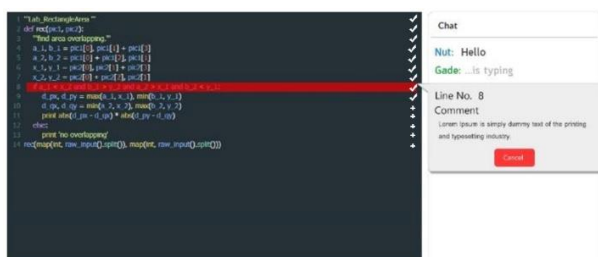
- H. ส่วนสำหรับการแก้ไขโค้ด เป็นพื้นที่สำหรับการเขียนโปรแกรม มีการไฮไลต์ตาม Syntax ของภาษา เพื่อให้สะดวกต่อการการอ่านและเขียนโค้ด เมื่อ Driver แก้ไขโค้ดแล้ว Navigator จะสามารถเห็นโค้ดและตรวจทานคำสั่งได้ทันที โดยสามารถเพิ่มความคิดเห็นในแต่ละบรรทัดของโค้ดได้
- I. ส่วนการแสดงผลลัพธ์จากการรันโปรแกรม เป็นส่วนที่จะแสดงผลหลังจาก Driver กดปุ่มรัน โดยจะนำโค้ดไปประมวลผลในเซิร์ฟเวอร์ แล้วส่งกลับมาแสดงผล เมื่อ Driver หรือ Navigator กดปุ่มรัน ทั้งคู่ก็จะสามารถเห็นผลลัพธ์ของโค้ดได้พร้อม ๆ กัน
- J. ส่วนของระบบส่งข้อความแบบทันที เป็นส่วนที่แสดงข้อความที่ถูกส่งจากผู้เขียนโปรแกรมทั้งคู่ โดยจะมีช่องด้านล่างสำหรับกรอกข้อความที่ต้องการส่งและมีปุ่มสำหรับส่งข้อความ

ผู้ที่ทำหน้าที่เป็น Navigator เท่านั้น จะสามารถดับเบิลคลิกที่บรรทัดที่ต้องการแสดงความคิดเห็นได้ จากนั้นจะมีหน้าต่างขึ้นมาด้านข้างเพื่อให้กรอกความคิดเห็นของโค้ดบรรทัดนั้น ๆ และเมื่อ Navigator กดที่ปุ่ม Comment ระบบจะไฮไลต์บรรทัดที่ถูกแสดงความคิดเห็นด้วยสีแดงให้ Driver สังเกตได้ง่าย ซึ่งการแสดงความคิดเห็นดังกล่าวถือเป็นการตรวจทานโค้ดแบบซัดเจ้งและระบบจะตรวจสอบสถานะการทำงานของ Navigator ด้วยเพื่อการตรวจทานโค้ดแบบโดยนัย ถ้า Navigator นั่งอยู่ที่หน้าจอในขณะนั้นเป็นเวลาอย่างน้อย 2 นาที หากไม่เพิ่มความคิดเห็นจะถือว่าโค้ดในส่วนนั้นได้ถูกตรวจทานเรียบร้อยแล้ว โดยจะขึ้นเป็นเครื่องหมายถูกที่หลังบรรทัดนั้น ดังรูปที่ 7.



รูปที่ 7. หน้าแสดงความคิดเห็น

Driver จะสามารถอ่านความคิดเห็นของ Navigator ได้โดยการดับเบิลคลิกไปยังบรรทัดที่ถูกไฮไลต์ด้วยสีแดง และมีหน้าต่างที่แสดงความคิดเห็นจาก Navigator ให้ Driver ได้อ่าน ดังรูปที่ 8.



รูปที่ 8. หน้าอ่านความคิดเห็น

## 6. บทสรุป

สิ่งแวดล้อมเพื่อการพัฒนาซอฟต์แวร์สำหรับการเขียนโปรแกรมแบบจับคู่แบบทางไกลได้จัดทำในรูปแบบของเว็บแอปพลิเคชันที่ช่วยให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมแบบจับคู่ได้สะดวกมากขึ้น แม้มีข้อจำกัดในเรื่องของสถานที่และความกังวลในเรื่องทักษะของผู้ที่ต้องการจับคู่ที่อาจต่างกันมากเกินไป โดยสามารถเลือกผู้เขียนโปรแกรมที่ต้องการจับคู่ได้เองหรือเลือกจากการแนะนำผู้เขียนโปรแกรมที่มีพฤติกรรมในการเขียนโค้ดที่คล้ายคลึงกันโดยได้มาจากการวิเคราะห์โค้ดการนำหลักการของการเขียนโปรแกรมแบบจับคู่มาใช้ (เช่น การกำหนดผู้เขียนโปรแกรมให้เป็น Driver และ Navigator ที่ทำไดเฉพาะสิ่งที่แต่ละหน้าที่ทำได้ การสลับหน้าที่ตามช่วงเวลาที่ได้ตั้งใจไว้หรือการสลับด้วยตนเอง) รวมถึงแสดงส่วนที่ใช้ในการแก้ไขโค้ด ส่วนแสดงผลลัพธ์ในหน้าจอของทั้งคู่ พร้อมทั้งการส่งข้อความแบบทันทีและวิดีโอการสนทนาระหว่างบุคคลเสมือนเขียนโปรแกรมอยู่ในสถานที่เดียวกัน เพื่อให้เกิดประสิทธิภาพมากขึ้นและช่วยให้เกิดการถ่ายทอดประสบการณ์การเรียนรู้ไปยังผู้เขียนโปรแกรมทั้งคู่ได้ดีขึ้น ทำให้สอดคล้องกับเป้าหมายของการเขียนโปรแกรมแบบจับคู่ที่แท้จริง

สำหรับระบบที่ได้พัฒนานั้น สามารถนำไปใช้งานจริงได้ทั้งผู้เขียนโปรแกรมทั่วไปที่ต้องการแลกเปลี่ยนเรียนรู้กันในส่วนขอเทคนิคการแก้ปัญหาในการเขียนโปรแกรม สถานศึกษาหรือสถาบันการศึกษาต่าง ๆ ที่ต้องการให้นักศึกษาได้เรียนรู้ซึ่งกันและกัน ส่งเสริมให้เกิดความพึงพอใจและปฏิบัติตามแนวปฏิบัติที่ดีในการเขียนโปรแกรมมากขึ้น

## เอกสารอ้างอิง

- [1] L. Williams, "Pair Programming," Encyclopedia of Software Engineering 2010, pp. 651-659
- [2] F. Spaven, "Pair Programming - So how good is it, really?," 2015. [Online]. Available: <https://raygun.com/blog/how-good-is-pair-programming-really/>. [Accessed: 02-Sep-2017].
- [3] K. Beck and C. Andres, *Extreme Programming Explained: embrace change*, 2nd ed. Upper Saddle River: Pearson Education, Inc., 2012.
- [4] J. Shore and S. Warden, *The Art of Agile Development*, 1st ed. Sebastopol: O'Reilly Media, Inc., 2008.
- [5] L. Williams and R. Kessler, *Pair Programming Illuminated*, 1st ed. Addison-Wesley, 2002.
- [6] A. Lombardi, *WebSocket*, 1st ed. Sebastopol: O'Reilly Media, Inc., 2015
- [7] M. Cantelon, M. Harter, T.J. Holowaychuk and N. Rajlich, *Node.js in Action*, Shelter Island: Manning Publications Co., 2014
- [8] E. Brown, *Web Development with Node & Express*, 1st ed. Sebastopol: O'Reilly Media, Inc., 2014
- [9] "What is Pylint?," 2015. [Online]. Available: <https://pylint.readthedocs.io/en/latest/intro.html>. [Accessed: 12-Sep-2017].