



CODING PLATFORM

MR. KRID HEPRAKRONE

MR. JATETANAN KANCHANAWAT

MS. PIMMADA LAISUAN

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)
FACULTY OF ENGINEERING
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI
2023

Coding Platform

Mr. Krid Heprakrone

Mr. Jatetanan Kanchanawat

Ms. Pimmada Laisuan

A Project Submitted in Partial Fulfillment
of the Requirements for
the Degree of Bachelor of Engineering (Computer Engineering)
Faculty of Engineering
King Mongkut's University of Technology Thonburi
2023

Project Committee

Project Advisor

(Assoc. Prof. Natasha Dejdumrong, D. Tech. Sci.)

Committee Member

(Asst. Prof. Priyakorn Pusawiro, Ph.D.)

Committee Member

(Asst. Prof. Rajchawit Sarochawikasit, M.D.)

Committee Member

(Jaturon Harnsomburana, Ph.D.)

| | |
|-----------------|---|
| Project Title | Coding Platform |
| Credits | 3 |
| Member(s) | Mr. Krid Heprakrone Mr. Jatetanan Kanchanawat Ms. Pimmada Laisuan |
| Project Advisor | Assoc. Prof. Natasha Dejdumrong, D. Tech. Sci. |
| Program | Bachelor of Engineering |
| Field of Study | Computer Engineering |
| Department | Computer Engineering |
| Faculty | Engineering |
| Academic Year | 2023 |

Abstract

Proficiency in computer programming is a vital skill in computer science, leading to a surge in student enrollment for programming courses. However, the effectiveness of teaching programming diminishes as class sizes increase, and instructors find it challenging to manually assess each student's assignments promptly. To address this issue, we propose the development of a software solution, Codern, a Coding Platform designed to automate and streamline the grading process. Codern evaluates student submissions against predefined test cases, measures resource usage efficiency, and instantly provides feedback and scores. This not only reduces the burden on instructors but also fosters a competitive and engaging learning environment. The platform also maintains a comprehensive record of student performance, aiding both instructors in refining their teaching methods and students in showcasing their achievements.

Keywords: Source Code / Program / Grading / Evaluation / Application

| | |
|--------------------|--|
| หัวข้อปริญญาบัณฑิต | โปรแกรม/แอปพลิเคชัน/เว็บแอปพลิเคชัน สำหรับตรวจสอบบ้านและข้อสอบในรายวิชา การเรียนการสอนเชื่ินโปรแกรมภาษาคอมพิวเตอร์ |
| หน่วยกิต | 3 |
| ผู้เขียน | นายกฤษฎี เอ่ปะโคน นายเจตనันท์ กัญจนวัฒน์ นางสาวพิมพ์มาดา ไอลรุวน |
| อาจารย์ที่ปรึกษา | รศ.ดร.น้ำชา เดชดำรง |
| หลักสูตร | วิศวกรรมศาสตรบัณฑิต |
| สาขาวิชา | วิศวกรรมคอมพิวเตอร์ |
| ภาควิชา | วิศวกรรมคอมพิวเตอร์ |
| คณะ | วิศวกรรมศาสตร์ |
| ปีการศึกษา | 2566 |

บทคัดย่อ

ความเชี่ยวชาญในการเขียนโปรแกรมคอมพิวเตอร์เป็นทักษะสำคัญในวิทยาการคอมพิวเตอร์ ส่งผลให้ในช่วงมาหลัง นักศึกษาหันมาลงทายเป็นในหลักสูตรการเรียนโปรแกรมเพิ่มมากขึ้น ส่งผลให้ประสิทธิภาพของการสอนการเขียนโปรแกรมจะลดลงเมื่อจำนวนผู้เรียนเพิ่มมากขึ้น อาจารย์ผู้สอนพบว่าการประเมินงานมอบหมายของนักเรียนแต่ละคนด้วยตนเองเป็นเรื่องท้าทาย เพื่อแก้ไขปัญหานี้ คณะผู้จัดทำเสนอแนวทางแก้ไขปัญหาในรูปแบบของซอฟต์แวร์ชื่อ Coding Platform (หรือ Codern) ซึ่งเป็นแพลตฟอร์มสำหรับเขียนและตรวจโปรแกรมแล้วให้ผลในทันทีทันใด ซอฟต์แวร์ดังกล่าวตรวจสอบและประเมินงานที่ถูกส่งมาเทียบกับกรณีทดสอบที่กำหนดไว้ล่วงหน้า วัดประสิทธิภาพการใช้ทรัพยากร แสดงผลคะแนนได้ทันที สิ่งนี้ไม่เพียงแต่ช่วยลดภาระของผู้สอนเท่านั้น แพลตฟอร์มดังกล่าวยังรักษาบันทึกผลและข้อมูลการใช้งานของผู้ใช้อย่างครอบคลุมเพื่อนำมาเป็นข้อมูลให้อาจารย์ผู้สอนปรับปรุงวิธีการสอนของตน และจัดแสดงความสำเร็จของนักศึกษา เพื่อส่งเสริมสภาพแวดล้อมการเรียนรู้ที่มีการแข่งขันและมีส่วนร่วมอีกด้วย

คำสำคัญ: รหัสต้นฉบับ / โปรแกรม / การประเมิน / การวัดผลสัมฤทธิ์ / แอปพลิเคชัน

กิตติกรรมประกาศ

โครงการพัฒนาโปรแกรม/แอปพลิเคชัน/เว็บแอปพลิเคชัน สำหรับตรวจภาระงานและข้อสอบในรายวิชาการเรียนการสอนเขียนโปรแกรมภาษาคอมพิวเตอร์ หรือ Coding Platform สามารถบรรลุเป้าหมายได้ด้วยดี ทางคณบัญชีจัดทำ ขอขอบคุณผู้สนับสนุนทุกท่านที่ให้การช่วยเหลือในด้านต่าง ๆ ขอขอบคุณ ศศ.ดร.ณัฐชา เดชคำรัง ที่มาเป็นอาจารย์ที่ปรึกษาโครง และสละเวลาในการให้ความรู้และคำแนะนำตลอดการพัฒนาทั้งโครงการ ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ที่ให้การส่งเสริมและสนับสนุนทางคณบัญชีจัดทำ ตลอดจนเอื้อเฟื้อสถานที่และอุปกรณ์สำหรับการพัฒนาโครงการ ขอขอบคุณนักศึกษาที่เรียนวิชา CPE222 Algorithm Design, CPE100 Computer Programming for Engineers, และนักเรียนมารยมศึกษาตอนปลายที่เข้าร่วมโครงการกิจกรรมการแข่งขันเขียนโปรแกรมคอมพิวเตอร์ Bangmod Hackathon 2024 ในปีการศึกษาพุทธศักราช 2566 ที่สละเวลา มาทดลองใช้ซอฟต์แวร์ต้นแบบ

สารบัญ

หน้า

| | |
|--|--------|
| ABSTRACT | ii |
| บทคัดย่อ | iii |
| กิตติกรรมประกาศ | iv |
| สารบัญ | v |
| สารบัญตาราง | vii |
| สารบัญรูปภาพ | viii |
| บทที่ 1 บทนำ | 1 |
| 1.1 ที่มาและความสำคัญ | 1 |
| 1.2 วัตถุประสงค์ | 1 |
| 1.3 ขอบเขตของโครงงาน | 1 |
| 1.4 ประโยชน์ที่คาดว่าจะได้รับ | 3 |
| 1.5 ตารางการดำเนินงาน | 4 |
| บทที่ 2 ทฤษฎีความรู้และงานที่เกี่ยวข้อง | 5 |
| 2.1 ทฤษฎีที่เกี่ยวข้อง | 5 |
| 2.1.1 Separation of Concern | 5 |
| 2.1.2 Version Control | 5 |
| 2.1.3 Continuous Integration และ Continuous Delivery | 6 |
| 2.2 โครงงานหรืองานวิจัยที่เกี่ยวข้อง | 6 |
| 2.2.1 IPST Program Grader | 6 |
| 2.2.2 Codeforces | 6 |
| 2.2.3 Programming Grader System | 7 |
| 2.3 ภาษาคอมพิวเตอร์และเทคโนโลยี | 8 |
| 2.3.1 ภาษาคอมพิวเตอร์ | 8 |
| 2.3.1.1 Go | 8 |
| 2.3.1.2 JavaScript | 8 |
| 2.3.1.3 TypeScript | 9 |
| 2.3.2 เครื่องมือ | 9 |
| 2.3.2.1 React | 9 |
| 2.3.2.2 GoFiber | 10 |
| 2.3.2.3 Grafana | 10 |
| 2.3.2.4 GitHub | 10 |
| 2.3.2.5 Docker | 11 |
| 2.3.2.6 Terraform | 11 |
| 2.3.2.7 Kubernetes | 11 |
| 2.3.3 โครงสร้างพื้นฐาน | 11 |
| 2.3.3.1 MySQL | 11 |
| 2.3.3.2 InfluxDB | 12 |
| 2.3.3.3 Prometheus | 12 |
| 2.3.3.4 RabbitMQ | 12 |
| บทที่ 3 วิธีการดำเนินงาน | 13 |

| | | |
|---------|--|----|
| 3.1 | รายละเอียดของโครงงาน | 13 |
| 3.1.1 | ข้อกำหนดและความต้องการของระบบ | 13 |
| 3.1.2 | กรณีการใช้งาน (Use Cases) | 15 |
| 3.1.3 | คำบรรยายกรณีใช้งาน (Use Cases Narratives) | 16 |
| 3.1.3.1 | Guest User (บุคคลทั่วไป) | 16 |
| 3.1.3.2 | Registered User (ผู้ใช้ทั่วไป) | 16 |
| 3.1.3.3 | Attendee (ผู้เข้าร่วมกิจกรรม) | 16 |
| 3.1.3.4 | Student (นักเรียนหรือนักศึกษา) | 17 |
| 3.1.3.5 | Contestant (ผู้เข้าร่วมการแข่งขัน) | 17 |
| 3.1.3.6 | Moderator (ผู้ช่วยเจ้าของห้องหรือผู้ช่วยสอน) | 17 |
| 3.1.3.7 | Owner (เจ้าของห้องหรืออาจารย์ผู้สอน) | 17 |
| 3.1.3.8 | Admin (เจ้าของระบบหรือยอดมิน) | 17 |
| 3.2 | สถาปัตยกรรมระบบ | 18 |
| 3.3 | ส่วนประสานต่อผู้ใช้ | 19 |
| 3.4 | โครงสร้างฐานข้อมูล | 37 |
| 3.4.1 | พจนานุกรมข้อมูล | 41 |
| 3.5 | แผนผัง UML | 43 |
| 3.5.1 | แผนผังแสดงโครงสร้างและส่วนประกอบของซอฟต์แวร์ | 43 |
| 3.5.2 | แผนภาพลำดับการทำงาน | 45 |
| 3.5.3 | แผนผังแสดงโครงสร้างและการนำทางระหว่างหน้าของเว็บไซต์ | 46 |

สารบัญตาราง

| ตารางที่ | หน้า |
|---|------|
| 1.1 ตารางการดำเนินการในภาคการศึกษาที่ 1 | 4 |
| 1.2 ตารางการดำเนินการในภาคการศึกษาที่ 2 | 4 |
| 3.1 ข้อกำหนดของซอฟต์แวร์ (1) | 13 |
| 3.2 ข้อกำหนดของซอฟต์แวร์ (2) | 14 |
| 3.3 พจนานุกรมข้อมูลของตาราง User | 41 |
| 3.4 พจนานุกรมข้อมูลของตาราง Session | 41 |
| 3.5 พจนานุกรมข้อมูลของตาราง Workspace | 41 |
| 3.6 พจนานุกรมข้อมูลของตาราง Workspace Participant | 41 |
| 3.7 พจนานุกรมข้อมูลของตาราง Assignment | 42 |
| 3.8 พจนานุกรมข้อมูลของตาราง TestCase | 42 |
| 3.9 พจนานุกรมข้อมูลของตาราง Submission | 42 |
| 3.10 พจนานุกรมข้อมูลของตาราง Submission Result | 42 |

สารบัญ

| รุปที่ | หน้า |
|--|------|
| 2.1 วิจัยการการทำงานของ React | 9 |
| 2.2 ตัวอย่างหน้าจอฟ์ต์แวร์ Grafana | 10 |
| 3.1 ภาพແນພັດກຣອື່ນເຂົ້າງານ | 15 |
| 3.2 ภาพແນພັດສາບັດຍກຣມຮະບບຂອງອົບົດົວົວ | 18 |
| 3.3 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າທີ່ສູ່ຮະບບ | 19 |
| 3.4 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າແຜງຄວບຄຸມ (1) | 19 |
| 3.5 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າແຜງຄວບຄຸມ (2) | 20 |
| 3.6 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າແຜງຄວບຄຸມ (3) | 21 |
| 3.7 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າແຜງຄວບຄຸມ (4) | 21 |
| 3.8 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າແຜງຄວບຄຸມ (5) | 22 |
| 3.9 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າແຜງຄວບຄຸມ (6) | 22 |
| 3.10 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າຕັ້ງຄ່າຂອງຸ່ນໄຟ້ | 23 |
| 3.11 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າແຜງຄວບຄຸມຂອງຸ່ດູແລຮະບບ | 24 |
| 3.12 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າຮາຍກາໂຈໂທຍັບໝາຫາ | 24 |
| 3.13 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າຮາຍກາໂຈໂທຍັບໝາຫາຂອງຸ່ດູແລຮະບບ | 25 |
| 3.14 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າຝອຣມສຽາໂຈໂທຍັບໝາຫາ | 26 |
| 3.15 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າຝອຣມແກ້ໄຂໂຈໂທຍັບໝາຫາ | 27 |
| 3.16 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າຮາຍຂໍ້ອຸ່ນໄຟ້ໃນ Workspace ຂອງຸ່ດູແລຮະບບ | 28 |
| 3.17 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າຮາຍກາໂປຣແກຣມທີ່ສ່າງເຂົ້າມາໃນ Workspace ຂອງຸ່ດູແລຮະບບ | 29 |
| 3.18 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າຮາຍຂໍ້ໆສາມືກທ່ອງເຮັດວຽກທີ່ໄດ້ສ່າງຄຳຕອບຂອງໂປຣແກຣມເຂົ້າມາໃນ Workspace ຂອງຸ່ດູແລຮະບບ | 30 |
| 3.19 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າແສດງຄຳຕອບຂອງໂປຣແກຣມທີ່ໄຟ້ໄດ້ສ່າງເຂົ້າມາໃນ Workspace ຂອງຸ່ດູແລຮະບບ | 31 |
| 3.20 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າຕັ້ງຄ່າ Workspace ຂອງຸ່ດູແລຮະບບ (1) | 32 |
| 3.21 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າຕັ້ງຄ່າ Workspace ຂອງຸ່ດູແລຮະບບ (2) | 33 |
| 3.22 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າຕັ້ງຄ່າ Workspace ຂອງຸ່ດູແລຮະບບ (3) | 34 |
| 3.23 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າແຜງເຂື້ອງໂປຣແກຣມ (1) | 35 |
| 3.24 ສ່ວນປະສານຕ່ອຸ່ນໄຟ້ ຫ້າແຜງເຂື້ອງໂປຣແກຣມ (2) | 35 |
| 3.25 ສ່ວນປະສານຈານຕ່ອຸ່ນໄຟ້ ຫ້າແສດງຂໍ້ອຸ່ນໄຟ້ | 36 |
| 3.26 ແພນກາພູານຂໍ້ອຸ່ນໄຟ້ | 37 |
| 3.27 ແພນພັດແສດງຄວາມສົມພັນຮະຫວ່າງຕາງໆ User ແລະ Session | 38 |
| 3.28 ແພນພັດແສດງຄວາມສົມພັນຮະຫວ່າງຕາງໆ User ແລະ Workspace | 38 |
| 3.29 ແພນພັດແສດງຄວາມສົມພັນຮະຫວ່າງຕາງໆ User ແລະ WorkspaceParticipant | 38 |
| 3.30 ແພນພັດແສດງຄວາມສົມພັນຮະຫວ່າງຕາງໆ User ແລະ Submission | 38 |
| 3.31 ແພນພັດແສດງຄວາມສົມພັນຮະຫວ່າງຕາງໆ Workspace ແລະ WorkspaceParticipant | 39 |
| 3.32 ແພນພັດແສດງຄວາມສົມພັນຮະຫວ່າງຕາງໆ Workspace ແລະ Assignment | 39 |
| 3.33 ແພນພັດແສດງຄວາມສົມພັນຮະຫວ່າງຕາງໆ Assignment ແລະ Submission | 39 |
| 3.34 ແພນພັດແສດງຄວາມສົມພັນຮະຫວ່າງຕາງໆ Assignment ແລະ TestCase | 39 |
| 3.35 ແພນພັດແສດງຄວາມສົມພັນຮະຫວ່າງຕາງໆ Assignment ແລະ TestCase | 40 |
| 3.36 ກາພແນພັດແສດງໂຄຮງສ້າງແລະສ່ວນປະກອບຂອງອົບົດົວົວ | 43 |
| 3.37 ກາພແນພັດກາລົດັບການ | 45 |
| 3.38 ກາພແນພັດແສດງໂຄຮງສ້າງແລະການນຳທາງຮະຫວ່າງຫ້າຂອງເຈັບໄຊ | 46 |

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญ

ทักษะการเขียนโปรแกรมคอมพิวเตอร์เป็นทักษะที่สำคัญสำหรับวิชาชีพในสาขาวิชาการคอมพิวเตอร์ เป็นเหตุให้นักศึกษาจำนวนมากหันมาสมัครหรือลงทะเบียนโปรแกรมคอมพิวเตอร์เพิ่มมากขึ้น ทำให้การเรียนการสอนวิชาเขียนโปรแกรมคอมพิวเตอร์มีประสิทธิภาพที่ดีลง เพราะจำนวนผู้เรียนมากขึ้น อาจารย์ผู้สอนไม่สามารถจะดูแลนักศึกษาได้ครบถ้วน การตรวจงานและให้คะแนนใช้ระยะเวลา lange เพราะอาจารย์ผู้สอนจะต้องนับไฟล์งานของนักศึกษามาตรฐานที่ลีฟเลฟส์ อีกทั้งยังมีโอกาสเกิดข้อผิดพลาดในการตรวจอีกด้วย

จากปัญหาดังกล่าว กลุ่มผู้จัดทำได้มีแนวทางแก้ปัญหาด้วยการสร้างเป็นโครงงานพัฒนาซอฟต์แวร์ Coding Platform (หรือ Codern) เพื่อให้อาจารย์ผู้สอนได้นำไปใช้เป็นเครื่องมือช่วยตรวจงานของนักศึกษาโดยยังคงต้นมัตติ ด้วยแนวทางนี้อาจารย์ผู้สอนจะได้เมื่อต้องทุ่มเททรัพยากรและเวลาไปกับการตรวจการบ้าน แล้วสามารถที่จะทุ่มเทไปยังงานในด้านการสอน และพัฒนาคุณภาพการเรียนการสอนในรายวิชา

โดยซอฟต์แวร์จะนำไฟล์ชุดคำสั่งที่นักศึกษาส่งเข้ามาในระบบ มาตรวจสอบความถูกต้องจากชุดข้อมูลทดสอบ (Test Case) ที่อาจารย์ผู้สอนได้กำหนดไว้ในโจทย์แต่ละข้อ จากนั้นนำไปวัดและประเมินประสิทธิภาพของการใช้ทรัพยากรของโปรแกรม เช่น ระยะเวลาหรือปริมาณหน่วยความจำที่ถูกโปรแกรมใช้ในประมวลผลหากำตอบ ก็จะจากตรวจสอบ เรื่องระบบจะบันทึกผลลัพธ์และคะแนนทันทีให้นักศึกษาได้เห็นทันที ไปแสดงผลบนระบบด้านคะแนนที่ทุกคนสามารถเข้าไปดูได้ เพื่อเสริมสร้างความสนุกสนาน การแข่งขันและส่งเสริมนักศึกษาให้พัฒนาตนเอง อีกทั้งประวัติการใช้งาน งานทั้งหมดที่นักศึกษาได้ส่ง จะถูกนำไปเก็บไว้ในฐานข้อมูลผู้ใช้อีกด้วย เพื่อให้อาจารย์ผู้สอนนำไปใช้การปรับปรุงหรือพัฒนาการสอนต่อๆ ไป หรือเพื่อให้นักศึกษานำมาอ้างอิงที่ได้ทำในระบบ ไปยังประกอบกับพอร์ตโฟลิโอยืนยันหลักฐานและผลงานความสำเร็จ

1.2 วัตถุประสงค์

ทางกลุ่มคณะผู้จัดทำ ได้ดำเนินโครงการพัฒนาซอฟต์แวร์ดังกล่าว ด้วยวัตถุประสงค์ดังต่อไปนี้

- เพื่อนำความรู้การบริหารจัดการงานด้านซอฟต์แวร์จากวิชาวิศวกรรมซอฟต์แวร์ มาประยุกต์ใช้กับการทำงาน
- เพื่อสร้างโอกาสทดลอง ออกรอบและสร้างช่องทางใหม่ๆ ในรูปแบบมืออาชีพ
- เพื่อนำหลักการออกแบบในทางวิศวกรรม ที่เรียนมาจากรายวิชาของภาควิชาวิศวกรรมคอมพิวเตอร์ มาออกแบบสร้างระบบซอฟต์แวร์ที่มีนวัตกรรมและยั่งยืน
- เพื่อช่วยอาจารย์ผู้สอนตรวจงาน วัดและประเมินแบบฝึกหัด การบ้าน หรือข้อสอบนักศึกษาในรายวิชาการเขียนโปรแกรมคอมพิวเตอร์
- เพื่อยกระดับกระบวนการเรียนรู้ของนักศึกษา ด้วยการนำเสนอข้อมูลที่น่าสนใจ ตื่นเต้น น่าเรียนรู้ ตรวจสอบในโครงการแข่งขันเขียนโปรแกรมคอมพิวเตอร์ที่จัดขึ้นในภาควิชา
- เพื่อกี๊ดข้อมูลการเรียนการสอน ทั้งการงานที่นักศึกษาได้ทำ และผลสอบของนักศึกษาเข้าฐานข้อมูลผู้ใช้ เพื่อให้อาจารย์ผู้สอนนำไปอ้างอิงและพัฒนาคุณภาพการสอนต่อๆ ไป

1.3 ขอบเขตของโครงงาน

- ซอฟต์แวร์รองรับการแสดงผล 2 ภาษาได้แก่ ภาษาไทยและภาษาอังกฤษ
- ซอฟต์แวร์เน้นการใช้งานบนหน้าจอ Desktop
- มีระบบลงทะเบียนหรือยืนยันตัวตนผู้ใช้ (User Registration and Authentication System)
- มีหน้าแสดงโจทย์ปัญหา ให้นักศึกษา/ผู้เข้าร่วมการแข่งขัน/ผู้ใช้ทั่วไปดู
 - รองรับการแสดงโจทย์แบบ Markdown, ไฟล์ PDF, และรูปภาพ

5. มีແຜ່ສຳຫັບເຂົ້ານໂປຣແກຣມທີ່ໄດ້ໃຫ້ໂຄດ (Built-in Code Editor) ຄູ້ປົກໂຈທຍ໌ ໃນໜ້າແສດງໂຈທຍ໌
6. ມີຮະບຄອມໄພລເລອ່ຽນສຳຫັບຮັນໂປຣແກຣມທີ່ສິ່ງເຂົ້າມາ (Built-in Code Compiler)
 - (a) ຮອງຮັບພາຫາ C ແລະ C++ ເປົ້າຫຼັກ
 - (b) ຮອງຮັບການຮັນໂປຣແກຣມທີ່ປະກອບດ້ວຍ ໄຟລົງຈານເຂົ້ານຊຸດຄຳສົ່ງນາກກວ່າໜຶ່ງໄຟລ໌
 - (c) ຄອມໄພລເລອ່ຽນໂປຣແກຣມທີ່ສິ່ງເຂົ້າມາ ເພື່ອຕຽບສອບຄວາມຄຸກຕ້ອງໃນໜັງຮຽກຈານ (grading)
 - (d) ຄອມໄພລເລອ່ຽນໂປຣແກຣມເພື່ອແສດງຜລັກຮີເຫັນສຶກສາດູ ໃນໜ້າແຜ່ເຂົ້ານໂປຣແກຣມເພື່ອຕຽບໂປຣແກຣມກ່ອນສິ່ງ (debugging)
7. ມີຮະບຕຽບແລະປະເມີນໂຄດທີ່ໄດ້ສິ່ງເຂົ້າມາ (Automatic Grader and Evaluation) ໂດຍປະເມີນຈາກ
 - (a) ຄວາມຄຸກຕ້ອງ (Correction) ໂດຍເຫັນຊຸດໜີ້ມູລສິ່ງອອກ (Output) ຂອງໂປຣແກຣມກັບຊຸດໜີ້ມູລທົດສອບ (Test Case) ທີ່ກຳທັນດໄວ້ໂຈທຍ໌ຂັ້ນນີ້
 - (b) ປະສິທິກາພ (Performance) ປະເມີນໂດຍເຫັນກັບຕົວແປຣ/ຕົວໜີ້ (Metrics) ຕ່ອໄປນີ້
 - ຮະຢະເວລາທີ່ໃໝ່ໃນການທາຄາດອບ (Runtime)
 - ປະມານຫຼີ້ວ່າຈຳນວນໜ່ວຍຄວາມຈຳທີ່ໂປຣແກຣມໃໝ່ (Memory Usage)
8. ມີຮະບທ້ອງເຮືອນ/ກລຸ່ມເຮືອນ/ກລຸ່ມງານ (Class/Group/Workspace) ຕາມຮາຍວິຊາສຳຫັບເກົ້າໂຈທຍ໌
9. ມີແຜ່ຄວບຄຸມສຳຫັບອາຈາරຍ໌/ຜູ້ສອນ/ແອດມິນ (Admin/Instructor/Lecturer Dashboard) ໂດຍໃນແຜ່ຄວບຄຸມດັ່ງກ່າວອາຈາරຍ໌/ຜູ້ສອນ/ແອດມິນສາມາດຮັບ
 - (a) ສ້າງ/ແກ້ໄຂ/ລບ ແອດເຄີນນັກສຶກສາ/ຜູ້ເຂົ້າຮ່ວມການແຂ່ງຂັນ/ຜູ້ໃໝ່ທີ່ໄປໄດ້
 - (b) ສ້າງ/ແກ້ໄຂ/ລບ ອ້ອງເຮືອນທີ່ກລຸ່ມງານທີ່ໄດ້
 - (c) ເພີ່ມ/ລບ ນັກສຶກສາ/ຜູ້ເຂົ້າຮ່ວມການແຂ່ງຂັນ/ຜູ້ໃໝ່ທີ່ໄປເຂົ້າທ້ອງເຮືອນທີ່ກລຸ່ມງານທີ່ໄດ້
 - (d) ເພີ່ມ/ແກ້ໄຂ/ລບ ໂຈຍີບັນຫາໄດ້ ແລະສາມາດກຳທັນເຈືອນໄຂອອງໂຈທຍ໌ແຕ່ລະຂ້ອໄດ້ ເຖິງໜ່ວຍຄວາມຈຳທີ່ອຳນຸຍາດໃຫ້ເຂົ້າມາກທີ່ສຸດ (memory limit), ຂ່າວເວລາທີ່ໂປຣແກຣມກວ່າໄດ້ກຳທັນ (runtime limit) ອີ່ຈະໃຫ້ໃໝ່ ລາຍ
10. ຮອງຮັບການໃສ່ຂໍ້ມູນທົດສອບ (Test Case) ຂອງໂຈທຍ໌ແຕ່ລະຂ້ອໄດ້ ທີ່ປະກອບດ້ວຍ
 - (a) ຊຸດໜີ້ມູນນຳເຂົ້າ (Input) ຜ່ານ standard input
 - (b) ຊຸດໜີ້ມູນສິ່ງອອກ (Output) ຜ່ານ standard output
11. ມີແຜ່ຄວບຄຸມສຳຫັບນັກສຶກສາ/ຜູ້ເຂົ້າຮ່ວມການແຂ່ງຂັນ/ຜູ້ໃໝ່ທີ່ໄປ (Student/User Dashboard) ໂດຍໃນແຜ່ຄວບຄຸມດັ່ງກ່າວນັກສຶກສາ/ຜູ້ເຂົ້າຮ່ວມການແຂ່ງຂັນ/ຜູ້ໃໝ່ທີ່ໄປສາມາດຮັບ
 - (a) ເຂົ້າໄປເຂົ້ານໂປຣແກຣມແກ້ໄຈທີ່ບັນຫາໃນ້ທ້ອງເຮືອນ/ກລຸ່ມເຮືອນ/ກລຸ່ມງານທີ່ຕົນອູ້ໄດ້
12. ມີແລ່ງເກົ່າວຽກຮັບຮັບຮົມເອກສາຮັບຮົມເອກສາແລະຄຸ້ມື້ອກການໃໝ່ຈານ (Documentation, User Manual, and Learning Resource Repository)
13. ມີຮູ້ານຂໍ້ມູນທີ່ຮັບຮົມເກົ່າວຽກຮັບຮົມຂໍ້ມູນຜູ້ໃໝ່ (User Database) ທີ່ຈະເກັບປະວັດການໃໝ່ຈານ ໂຈຍີບັນຫາທີ່ໄດ້ທຳ ໄວເຂົ້າວັງອີ່ງຕ່ອໄປ
14. ມີຮະບຕາຮາງຄະແນນ (Scoreboard) ປະຈຳທ້ອງເຮືອນ/ກລຸ່ມເຮືອນ/ກລຸ່ມງານ ທີ່ແສດງຄຳດັບຂອງນັກສຶກສາ/ຜູ້ເຂົ້າຮ່ວມການແຂ່ງຂັນ/ຜູ້ໃໝ່ທີ່ໄປໃນກລຸ່ມນັ້ນນີ້

1.4 ประโยชน์ที่คาดว่าจะได้รับ

หลังจากสิ้นสุดโครงการ ทางคณะผู้จัดทำคาดหวังว่า โครงการดังกล่าวจะให้ผลประโยชน์ต่อทั้งตนและผู้อื่น ดังดังไปนี้

1. ทางคณะผู้จัดทำจะได้รับข้อคิด บทเรียนและประสบการณ์อันเป็นประโยชน์ต่อชีวิตการทำงานในอนาคต จากการบริหารจัดการวางแผน การออกแบบและสร้างซอฟต์แวร์ตัวนี้
2. อาจารย์ผู้สอนในรายวิชาการเขียนโปรแกรมคอมพิวเตอร์ จะได้รับความ satisfaction สามารถตรวจข้อสอบแข่งขันได้อย่างรวดเร็ว สามารถประเมินผลนักศึกษาได้อย่างรวดเร็วและมีประสิทธิภาพ
3. คณะกรรมการและคณะผู้จัดการแข่งขันเขียนโปรแกรมในภาควิชา จะสามารถตรวจข้อสอบแข่งขันได้อย่างรวดเร็ว สามารถประเมินผลแข่งขันได้ในทันทีและเกิดข้อผิดพลาดน้อยที่สุด
4. ทางภาควิชาศึกษากรรมคอมพิวเตอร์ จะสามารถนำข้อมูลการใช้งาน ข้อมูลคะแนนและการวัดผล ข้อมูลการเรียนการสอนจากระบบซอฟต์แวร์ดังกล่าวไปพัฒนาหลักสูตรและคุณภาพการสอนได้

1.5 ตารางการดำเนินงาน

ตารางที่ 1.1 ตารางการดำเนินการในภาคการศึกษาที่ 1

| สิงหาคม | กันยายน | ตุลาคม | พฤษจิกายน | ธันวาคม |
|--|---------|--------|-----------|---------|
| 1. ศึกษาค้นคว้า วิเคราะห์หน้าปัญหา และที่มา | | | | |
| 2. เสนอหัวข้อโครงการ | | | | |
| 3. ศึกษาค้นคว้า หาข้อมูลที่เกี่ยวข้อง | | | | |
| 4. นำเสนอโครงการให้กับอาจารย์ที่ปรึกษา | | | | |
| 5. จัดทำข้อเสนอโครงการ | | | | |
| 6. นำเสนอข้อเสนอโครงการ | | | | |
| 7. จัดทำรายงานของภาคการศึกษาที่ 1 | | | | |
| 8. วิเคราะห์และออกแบบ UX/UI | | | | |
| 9. พัฒนาซอฟต์แวร์และระบบ | | | | |
| 10. ทดสอบการทำงานของระบบ จากการใช้งานจริงครั้งที่ 1 | | | | |
| 11. นำเสนอและรายงานโครงการของภาคการศึกษาที่ 1 | | | | |

ตารางที่ 1.2 ตารางการดำเนินการในภาคการศึกษาที่ 2

| มกราคม | กุมภาพันธ์ | มีนาคม | เมษายน | |
|--|------------|--------|--------|--|
| 12. พัฒนาซอฟต์แวร์และระบบ (ต่อ จากภาคการศึกษาที่ 1) | | | | |
| 13. ทดสอบการทำงานของระบบ จากการใช้งานจริงครั้งที่ 2 | | | | |
| 14. ปรับปรุงและแก้ไขซอฟต์แวร์และ ระบบเพิ่มเติม | | | | |
| 15. จัดทำและส่งรายงานโครงการ ฉบับสมบูรณ์ | | | | |
| 16. นำเสนอโครงการ | | | | |

บทที่ 2 ทฤษฎีความรู้และงานที่เกี่ยวข้อง

ในบทที่ 2 จะกล่าวถึง ทฤษฎีที่เกี่ยวข้องที่ใช้ในการทำงานและการศึกษาเพื่อบรรลุตั้งแต่ประสงค์ การอธิบายคำศัพท์หรือภาษาทางคอมพิวเตอร์ ที่ได้กล่าวถึงและใช้งานในการแก้ไขปัญหา รวมไปถึงงานวิจัยหรือโครงการที่ได้ใช้อ้างอิงในการทำงาน

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 Separation of Concern

หลักการ Separation of Concern (SoC Principle) เป็นแนวคิดพื้นฐานในด้านวิศวกรรมซอฟต์แวร์และการออกแบบ โดยเน้นถึงความจำเป็นในการแบ่งระบบที่ซับซ้อนออกเป็นส่วนที่แตกต่างกันและจัดการได้ ซึ่งแต่ละส่วนจะกล่าวถึงลักษณะเฉพาะของฟังก์ชันการทำงาน (function) หรือภาระหน้าที่ (concern หรือ responsibility) [?]

หลักการนี้มีบทบาทสำคัญในการปรับปรุงคุณภาพซอฟต์แวร์ การบำรุงรักษา และความสามารถในการขยาย [?] ในขอบเขตของสถาปัตยกรรมซอฟต์แวร์ การแยกข้อกังวลมักจะเกี่ยวข้องกับการแบ่งระบบออกเป็นโมดูล (modules) หรือส่วนประกอบ (components) โดยแต่ละส่วนจะเกี่ยวข้องกับลักษณะเฉพาะของแอปพลิเคชัน เช่น ส่วนติดต่อผู้ใช้ การจัดเก็บข้อมูล หรือตระหง่านทางธุรกิจ

แนวทางนี้จะลดการพึ่งพาซึ่งกันและกันระหว่างส่วนต่างๆ ของซอฟต์แวร์ ทำให้ง่ายต่อการพัฒนา แก้ไข และดีบากซอฟต์แวร์ เนื่องจาก การเปลี่ยนแปลงในด้านหนึ่งมีอิทธิพลน้อยที่จะส่งผลกระทบต่อส่วนอื่น ๆ มันส่งเสริมฐานรากที่มีการจัดระเบียบและเข้าใจได้มากขึ้น ซึ่งอำนวยความสะดวกในการทำงานร่วมกันระหว่างนักพัฒนา

ตามความเห็นในของ Meyer [?] การสร้างซอฟต์แวร์ที่เป็นคุณภาพและมั่นคงคือการรักษาความเข้าใจและความถูกต้องของประสิทธิภาพ ของแต่ละส่วนในระบบ หนึ่งการพัฒนาที่มีแนวคิดในเรื่องนี้คือการใช้หลักการการแยกปัญหาที่ซัดเจนในองค์กร การเป็นที่รู้ในประเด็นนี้ และปรับใช้หลักการการแยกปัญหาอย่างมีประสิทธิภาพสามารถนำไปสู่การพัฒนาระบบซอฟต์แวร์ที่มีประสิทธิภาพและเครื่องมือที่มีคุณภาพสูง

2.1.2 Version Control

Version Control คือเครื่องมือที่ใช้สำหรับในการจัดการรหัสต้นฉบับ (หรือ source code) ซึ่งเป็นกระบวนการที่สำคัญในการพัฒนาซอฟต์แวร์ จำเป็นต้องมีการติดตามและการจัดการการเปลี่ยนแปลงใน code base ของโครงการ [?]

โดยเครื่องมือดังกล่าวช่วยให้นักพัฒนาซอฟต์แวร์หลายคน สามารถทำงานร่วมกันในโครงการซอฟต์แวร์โดยจัดให้มีวิธีที่เป็นระบบในการติดตามการเปลี่ยนแปลงที่เกิดขึ้นกับรหัสต้นฉบับ แนวทางปฏิบัตินี้ช่วยให้มองค์กร ความโปร่งใส และความเสถียรของโครงการซอฟต์แวร์ ช่วยให้นักพัฒนาสามารถดูและเข้าใจประวัติการเปลี่ยนแปลง แยกและแก้ไขปัญหา และจัดการงานที่เกิดขึ้นพร้อมกันได้อย่างมีประสิทธิภาพ Git ซึ่งเป็นเครื่องมือ Version Control ยอดนิยมระบบตัวหนึ่ง [?]

โดย Git เสนอแนวทางแบบกระจายและอนุกประสงค์ ซึ่งสนับสนุนการแยกสาขา (branch) และการผสานรวม (merge) ช่วยให้นักพัฒนาสามารถทำงานกับคุณสมบัติต่างๆ ได้พร้อมกัน และรวมงานของพัฒนาให้ดียิ่งขึ้นในภายหลัง หลักการนี้จำเป็นสำหรับการรับประทานความสมบูรณ์ของโครงการซอฟต์แวร์โดยจัดให้มีกลไกในการเปลี่ยนกลับเป็น Version ก่อนหน้า คุ้นเคยกับการติดตั้ง (rollbacks) และรักษาประวัติการเปลี่ยนแปลงโดยที่เชื่อถือได้ ซึ่งทั้งหมดนี้มีส่วนช่วยในการจัดการโครงการ และทำให้แน่ใจว่าการทำงานจะดำเนินไปอย่างราบรื่น

ในโครงการปริญนานิพนธ์ของ นายณัฐวัฒน์และคณะ [?] ได้มีการนำหลักการ Version Control มาใช้ร่วมและควบคู่กับการบริหารซอฟต์แวร์ในโครงการ เนื่องจากพัฒนาซอฟต์แวร์ต้องทำงานกันเป็นกลุ่ม การใช้ Version Control เข้ามาช่วยในการจัดการบริหารการเปลี่ยนแปลงของรหัสต้นฉบับ ช่วยให้การทำงานสะดวกยิ่งขึ้น ไม่เสียเวลา กับการจัดการไฟล์รหัสต้นฉบับ แล้วสามารถเอาเวลาไปมุ่งเน้นกับการพัฒนาโปรแกรม

2.1.3 Continuous Integration และ Continuous Delivery

Continuous Integration (CI) และ Continuous Delivery (CD) นิยมเรียกรวมกันว่าหลักการ CI/CD เป็นหลักการพื้นฐานในการพัฒนาซอฟต์แวร์สมัยใหม่ที่เน้นไปที่การพัฒนาซอฟต์แวร์ การเปลี่ยนแปลงรั้งสัตตนับควบคู่กับการส่งมอบซอฟต์แวร์ไปสู่การผลิต (production) โดยอัตโนมัติ เพื่อการทำงานที่มีประสิทธิภาพมากที่สุด

โดยเกี่ยวข้องกับการรวมการเปลี่ยนแปลงโค้ด (code merging) เข้ากับหนึ่งที่เก็บข้อมูลที่ใช้ร่วมกัน (shared repository) สำหรับงาน จากนั้นจะถูกสร้างขึ้น (production build), ทดสอบ (test), และเปิดใช้ (deploy) กับสภาพแวดล้อมต่างๆ (environment) โดยอัตโนมัติ [?]

แนวทางปฏิบัติหรือกระบวนการดังกล่าว จะช่วยเร่งการพัฒนา ลดบัญหาในขั้นตอนการบูรณาการ (integration) และทำให้มันใจได้ว่า ซอฟต์แวร์จะอยู่ในสภาพที่พร้อมใช้งานอยู่เสมอ

2.2 โครงงานหรืองานวิจัยที่เกี่ยวข้อง

2.2.1 IPST Program Grader

IPST Program Grader เป็นเว็บไซต์ แอปพลิเคชันที่สร้างขึ้นโดยสถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี (สวท.) [?] ถูกนำไปปรับปรุงใหม่โดยกลุ่มนักเรียนค่ายโอลิมปิกวิชาการคอมพิวเตอร์ในช่วงปีหลัง ๆ สร้างขึ้นมาเพื่อให้ผู้ใช้สามารถเข้ามาฝึกฝนทักษะการเขียนโปรแกรม เรียนรู้การเขียนโปรแกรม เรียนรู้เกี่ยวกับโครงสร้างข้อมูล และฝึกเขียนอัลгорิズึมที่มีประสิทธิภาพ

เว็บไซต์มีฐานข้อมูลโจทย์ปัญหาที่แต่งโดยทางสวท. ที่ผู้ใช้สามารถกดเลือกเข้าไปทำข้อไหนก็ได้ มีระบบสมัครสมาชิก และมี learning resource สำหรับให้ผู้ใช้ได้อ่าน ได้ฝึกฝน ทำความเข้าใจและเรียนรู้การเขียนโปรแกรม

- ข้อดี

1. เว็บไซต์มีระบบการตรวจและประเมินผลโปรแกรมที่รวดเร็ว ผู้ใช้สามารถรับรู้ผลได้ทันที
2. ส่วนประสานผู้ใช้ (User Interface) ถูกออกแบบมาอย่างดี เพื่อความสะดวกสบายของผู้ใช้
3. เว็บไซต์มีโจทย์ปัญหาที่หลากหลาย แต่แต่ระดับง่ายสุด ไปยังระดับการแข่งขันระดับนานาชาติ
4. เว็บไซต์มีระบบจัดหมวดหมู่โจทย์ปัญหา ทำให้ผู้ใช้หาโจทย์ปัญหาที่ต้องการทำได้ง่าย

- ข้อเสีย

1. เว็บไซต์ไม่สามารถจะใช้งานเครื่องข่ายเฉพาะได้ เพราะเว็บไซต์ตั้งกล่าวอยู่ในเครือข่ายสาธารณะ ทำให้เว็บไซต์นี้ไม่สามารถนำมายังเครื่องที่ไม่สามารถใช้ในการแข่งขันภายในได้
2. ไม่มีระบบสื่อสาร ไม่มีระบบกระทุกสนทนา ไม่มีช่องทางการสื่อสารให้ผู้ใช้ได้คุยกับกันเรื่องโจทย์
3. ผู้ใช้ไม่สามารถเพิ่มโจทย์ปัญหาเองได้ โจทย์ปัญหาถูกควบคุมและเพิ่มโดยผู้ดูแลเว็บเท่านั้น

2.2.2 Codeforces

ซอฟต์แวร์ **Codeforces** เป็นเว็บไซต์แอปพลิเคชัน ที่สร้างขึ้นโปรแกรมเมอร์ที่ชื่อ Mike Mirzayanov [?] ในปี 2553 สร้างขึ้นมาด้วยวัดถูประศักดิ์เดียวกับ programming.in.th เพื่อให้ผู้ใช้สามารถเข้ามาฝึกฝนทักษะการเขียนโปรแกรม เรียนรู้การเขียนโปรแกรม

เว็บไซต์มีฐานข้อมูลโจทย์ปัญหาที่แต่งโดยทางผู้ดูแลเว็บไซต์ เทเมื่อนับเว็บไซต์ programming.in.th ที่ผู้ใช้สามารถเข้าไปเลือกทำได้ แต่ข้อแตกต่างที่ทำให้เว็บไซต์นี้แตกต่างคือระบบการแข่งขัน ระบบตารางคะแนน

เว็บไซต์มีการจัดการแข่งขันเป็นช่วง ๆ มีการจัดลำดับผู้เข้าแข่งขันด้วยระบบการตรวจและประเมินโปรแกรม คะแนนจะมากน้อยขึ้นกับสองปัจจัยหลัก ประกอบด้วยความถูกต้อง (correction) และประสิทธิภาพ (performance or efficiency), โปรแกรมที่ให้คำตอบที่ถูกต้องและมีต่อรอง ใช้ทรัพยากรน้อย และใช้เวลาคำตอบน้อยจะได้คะแนนสูง ผู้เข้าแข่งขันคนไหนที่สามารถแก้โจทย์ปัญหาได้ถูกต้องและมีประสิทธิภาพ มากที่สุดในเวลาที่จำกัดจะถือเป็นผู้ชนะไป

• ข้อดี

1. เว็บไซต์มีระบบการตรวจสอบและประเมินผลโปรแกรมที่รวดเร็ว ผู้ใช้สามารถรับรู้ผลได้ทันที
2. มีระบบตารางคะแนน มีการจัดอันดับคะแนนผู้ใช้ ส่งเสริมให้ผู้ใช้พัฒนาตนเอง ส่งเสริมให้เกิดการแข่งขัน
3. เว็บไซต์มีโจทย์ปัญหาที่หลากหลาย จัดเป็นหมวดหมู่เรียบร้อย จัดระดับความยากง่าย ผู้ใช้สามารถหาโจทย์ปัญหาที่ต้องการทำได้จ่าย
4. มีระบบสื่อสาร มีระบบกระทุกหน้า มีช่องทางการสื่อสารให้ผู้ใช้ได้คุยกับกันเรื่องโจทย์ มีช่องทางประกาศประชาสัมพันธ์ ข่าวสารและผลการแข่งขัน

• ข้อเสีย

1. เว็บไซต์ไม่สามารถจะใช้งานเครือข่ายเฉพาะได้ เว็บไซต์อยู่ในเครือข่ายสาธารณะ เว็บไซต์นี้ไม่สามารถนำมาใช้ในการแข่งขันที่มีการจำกัดการเข้าถึงข้อมูลออนไลน์
2. ส่วนประสานผู้ใช้ (User Interface) ออกแบบมาไม่ดี ไม่สะดวกต่อการใช้งาน

2.2.3 Programming Grader System

ซอฟต์แวร์ PGS หรือชื่อเดิมโครงการพัฒนาระบบตรวจสอบผลลัพธ์ของโจทย์ปัญหาจากชั้ดรหัสต้นฉบับเพื่อการเรียนการสอน (ภาษาอังกฤษคือ Programming Grader System) เป็นโครงงานพัฒนาซอฟต์แวร์ของนักศึกษาชั้นปี 4 ในรายวิชา CPE401 ปีการศึกษา 2563 [?]

โดยซอฟต์แวร์ดังกล่าวถูกตั้งขึ้นมาเพื่อแก้ปัญหาการตรวจสอบงานที่ยุ่งยาก ในรายวิชาการเรียนการสอนเขียนโปรแกรมภาษาคอมพิวเตอร์ ซึ่งเป็นปัญหาเดียวที่นักศึกษาที่คณผู้จัดทำมุ่งมั่นได้ตั้งไว้ และมุ่งมั่นจะแก้ไขในโครงการนี้

เว็บไซต์นี้ถูกสร้างและออกแบบ โดยนำเอาข้อดีและข้อเสียของเว็บไซต์ programming.in.th และ codeforces.com มาพิจารณาร่วมด้วย เป็นเหตุให้มีระบบการทำงานและคุณลักษณะ (feature) มากกว่าซอฟต์แวร์ในทั่วโลก่อน

• ข้อดี

1. เว็บไซต์มีระบบการตรวจสอบและประเมินผลโปรแกรมที่รวดเร็ว ผู้ใช้สามารถรับรู้ผลได้หลังจากส่งชุดคำสั่งขึ้นตรวจสอบ
2. มีระบบตารางคะแนน มีการจัดอันดับคะแนนผู้ใช้ ส่งเสริมให้ผู้ใช้พัฒนาตนเอง ส่งเสริมให้เกิดการแข่งขัน
3. เว็บไซต์มีโจทย์ปัญหาที่หลากหลาย จัดเป็นหมวดหมู่ตามห้องเรียนเรียบร้อย จัดระดับความยากง่าย ผู้ใช้สามารถหาโจทย์ปัญหาที่ต้องการทำได้จ่าย
4. มีระบบการแสดงความเห็นในแต่ละชุดคำสั่งที่ผู้ใช้ได้ส่งขึ้นมา ที่อาจารย์ผู้สอนหรือผู้ดูแลสามารถมาให้ความเห็นหรือความแน่นหนาได้
5. ผู้ใช้ที่สามารถเพิ่มโจทย์ปัญหาเองได้ หากได้รับอนุญาตจากผู้ดูแลเว็บไซต์

• ข้อเสีย

1. เว็บไซต์ไม่สามารถรองรับการทำงานของผู้ใช้หลาย ๆ คน พร้อมกันได้
2. ส่วนประสานงานผู้ใช้ (หรือ User Interface) ไม่เป็นมิตร ไม่สะดวกต่อผู้ใช้
3. การเพิ่มหรือแก้ไขโจทย์นั้นยากลำบากและไม่สะดวก ไม่สามารถจะนำชุดโจทย์ ชุดเทสเคสมาเพิ่มพร้อมกันได้ ต้องทำการเพิ่มทีละรายการ
4. ในการตรวจสอบชุดคำสั่ง ต้องคัดลอก (copy) จากบนเว็บไซต์มาวาง (paste) บน IDE บนเครื่อง ไม่สามารถที่จะนำชุดคำสั่งออกจากเว็บไซต์เป็นไฟล์ (export file) ออกมานำไปได้
5. ค่าคะแนนที่ได้บันทึกไว้บนตารางคะแนน ไม่สามารถจะนำออกมายกเป็นไฟล์ excel สำหรับตรวจและประเมินได้

2.3 ภาษาคอมพิวเตอร์และเทคโนโลยี

2.3.1 ภาษาคอมพิวเตอร์

2.3.1.1 Go

Go หรือที่รู้จักกันอย่างแพร่หลายว่า Golang เป็นภาษาโปรแกรมเปิดต้นทางที่พัฒนาโดยทีมที่ Google โดยรวมถึง Robert Griesemer, Rob Pike, และ Ken Thompson ภาษาเนื้อกอกออกแบบขึ้น เพื่อแก้ไขข้อจำกัดของภาษาโปรแกรมที่มีอยู่แล้วทั้งในด้านประสิทธิภาพ (Efficiency) ความเรียบง่าย (Simplicity) และการสนับสนุนการทำงานพร้อมกัน (Concurrency มีไวยากรณ์ (Syntax) ที่กระชับ การกำหนดประเภท (Type) ที่แข็งแรง พร้อมทั้งระบบ GC (Garbage Collection) ที่มีประสิทธิภาพ [?, ?]

จากความเห็นของ Pike R. [?] หนึ่งในลักษณะที่เด่นของ Go คือการให้ความสำคัญกับความเรียบง่ายและความอ่านง่าย ทำให้เป็นทางเลือกที่ยอดเยี่ยมสำหรับทั้งผู้เริ่มต้นและนักพัฒนาที่มีประสบการณ์มาก มันนำเสนอแนวทางที่มีความเรียบง่าย ป้องกันความซับซ้อนที่ไม่จำเป็น และมีไวยากรณ์ที่ถูกต้องและโดยเด่น ภาษาเนี้ยรวมถึงคุณสมบัติที่สนับสนุนการทำงานพร้อมกันที่ซึ่งช่วยให้หักพัฒนาสามารถเขียนแอปพลิเคชันที่มีประสิทธิภาพและมีขนาดใหญ่ได้

การทำงานพร้อมกันหรือ Concurrency ก็เป็นแข็งแกร่งของ Go อย่างหนึ่ง การทำ Concurrency ในภาษา ทำผ่าน Channel และ Goroutines ซึ่งเป็น thread ที่ light-weighted ทำให้สามารถสร้างโปรแกรมที่ทำงานพร้อมกันได้โดยไม่มีความซับซ้อน [?]

Go ได้รับความนิยมในหลาย ๆ ด้าน เช่น การสร้างหรือพัฒนาเว็บไซต์ เริ่มแอปพลิเคชัน การสร้างระบบบริการ Cloud การสร้างซอฟต์แวร์ระบบที่มีประสิทธิภาพ เป็นต้น ภาษา Go เป็นภาษาที่ใช้งานง่าย และสามารถรันโปรแกรมที่อาศัยการทำงานหรือประมวลแบบ Parallel และ Concurrent [?]

2.3.1.2 JavaScript

JavaScript เป็นภาษาโปรแกรมที่หลากหลายและใช้กันอย่างแพร่หลาย โดยหลักแล้วเป็นที่รู้จักจากบทบาทในการพัฒนาเว็บ พัฒนาโดย Netscape และได้กลายเป็นองค์ประกอบพื้นฐานสำคัญของการสร้างเว็บไซต์ซึ่งต้องตอบและดำเนิน JavaScript ถูกดำเนินการบนฝั่งclient ทำให้เบราว์เซอร์สามารถจัดการ Document Object Model (DOM) และตอบสนองต่อการต้องขอของผู้ใช้ ไวยากรณ์ของมันได้รับอิทธิพลจาก Java ทำให้นักพัฒนาที่คุ้นเคยกับภาษา C สามารถเข้าถึงได้ [?]

ข้อดีอย่างหนึ่งที่โดดเด่นของ JavaScript คือการแพร่หลายไปทั่วเว็บเบราว์เซอร์ เนื่องจากได้รับการสนับสนุนโดยเบราว์เซอร์หลักๆ ทั่วหมด เช่น Chrome, Firefox และ Safari การนำไปใช้อย่างแพร่หลายนี้ทำให้มันใจได้ว่าแอปพลิเคชันที่ขับเคลื่อนด้วย JavaScript สามารถเข้าถึงผู้ชุมชนจำนวนมากได้โดยไม่มีปัญหาเรื่องความเข้ากันได้ [?] นอกจากนี้ JavaScript ยังเปิดใช้งานการเขียนโปรแกรมแบบօชิง โครงสร้างคุณสมบัติเช่นการโทรกลับและคำสัญญา ซึ่งอำนวยความสะดวกในการพัฒนาแอปพลิเคชันเว็บที่ตอบสนองและมีประสิทธิภาพ ลักษณะที่ไม่ซับซ้อนของภาษาช่วยให้โหลดหน้าเร็วได้เร็วขึ้น ปรับปรุงประสบการณ์ผู้ใช้โดยรวม [?]

อย่างไรก็ตาม JavaScript ก็มีข้อเสียอยู่เช่นกัน ข้อเสียเบี่ยงเบี้ยที่สำคัญประการหนึ่งคือโอกาสเกิดข้อห่วงด้านความปลอดภัย โดยเฉพาะอย่างยิ่งเมื่อดำเนินการบนฝั่งclient เนื่องจากช่องโหว่ที่สามารถโจมตีได้โดยผู้ใช้ จึงสามารถจัดการเพื่อยุดประสงค์ที่เป็นอันตรายได้ซึ่งนำไปสู่ความเสี่ยงด้านความปลอดภัย เช่น การโจมตี Cross-Site Scripting (XSS) [?] นอกจากนี้ การขาดการพิมพ์ที่ชัดเจนอาจทำให้การตรวจจับข้อผิดพลาดบางประเภทในระหว่างการพัฒนาเป็นเรื่องยาก ซึ่งอาจนำไปสู่ปัญหารันไทม์ที่ยากต่อการวินิจฉัยและแก้ไข [?]

แม้จะมีข้อจำกัด JavaScript ยังคงเป็นส่วนสำคัญของการพัฒนาเว็บไซต์ใหม่ เนื่องจากความสามารถรอบด้านและระบบนำเว็บที่กว้างขวาง ของไลบรารีและเฟรมเวิร์ก เช่น React และ Angular ที่สร้างขึ้นจากมัน [?] การพัฒนาอย่างต่อเนื่องของภาษาและความพยายามในการแก้ไขข้อบกพร่องมีส่วนช่วยให้ภาษามีความเกี่ยวข้องอย่างต่อเนื่องในภูมิทัศน์แบบดำเนินมีกิจกรรมของการพัฒนาเว็บ [?]

2.3.1.3 TypeScript

TypeScript เป็นชุดที่เหนือกว่าของ JavaScript ที่แนะนำการพิมพ์แบบคงที่ให้กับภาษา ช่วยให้นักพัฒนาซอฟต์แวร์มีเครื่องมือที่ได้รับการปรับปรุง การบำรุงรักษาได้ดีที่ได้รับการปรับปรุง และการตรวจสอบข้อผิดพลาดตั้งแต่เนิ่นๆ ในระหว่างการพัฒนา TypeScript เป็นโอเพนซอร์สและดูแลโดย Microsoft และคอมไพล์เป็น JavaScript ธรรมชาติ ทำให้เข้ากันได้กับโค้ดเบส JavaScript ที่มีอยู่ [?, ?]

ข้อดีหลักประการหนึ่งของ TypeScript คือระบบการพิมพ์แบบคงที่ ด้วยการรวมคำอธิบายประกอบประเภท นักพัฒนาสามารถกำหนดประเภทของตัวแปร พารามิเตอร์ฟังก์ชัน และค่าที่ส่งคืนที่คาดเดาได้ (Deterministic Behavior) ซึ่งจะช่วยตรวจสอบข้อผิดพลาดที่เกี่ยวข้องกับประเภทในระหว่างการพัฒนา ลดจุดบกพร่องหรือข้อผิดพลาด อีกทั้งปรับปรุงคุณภาพของโปรแกรม [?, ?]

นอกจากนี้ TypeScript ยังรองรับฟีเจอร์ ECMAScript สมัยใหม่ และสามารถใช้เพื่อสร้างโค้ดที่มีทั้ง Solubility และ Maintainability สำหรับแอปพลิเคชันขนาดใหญ่ ภาษา TypeScript ให้การสนับสนุนหลักการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) รวมถึงคลาสและอินเทอร์เฟซ (Class and Interface) อำนวยความสะดวกในการพัฒนาซอฟต์แวร์ที่แข็งแกร่งและมีโครงสร้าง [?]

อย่างไรก็ตาม TypeScript ก็มีข้อเสียเช่นกัน ข้อเสียเบื้องต้นคือเรียนรู้ใช้งานยาก โดยเฉพาะใช้การเขียนโปรแกรมแบบ Static Typing โดยเฉพาะอย่างยิ่งสำหรับนักพัฒนาที่คุ้นเคยกับรูปแบบเดิมๆ ของ JavaScript ข้อเสียเห็นได้ชัดเจนสำหรับโครงการขนาดเล็กที่ประโยชน์ของ Static Typing อาจไม่เด่นชัดเท่าที่ควร [?]

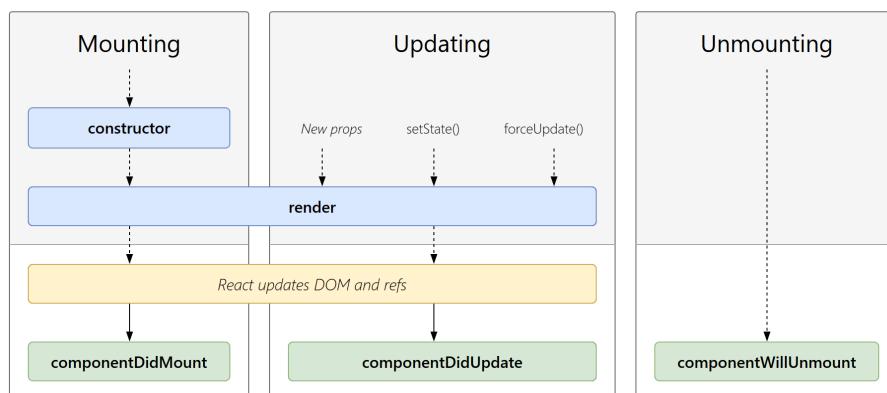
ข้อควรพิจารณาประการหนึ่งที่ Microsoft [?] ได้กล่าวในบทความก่อตัวของ TypeScript ว่าเป็นต้องมีถูกการแปลงภาษาเป็น JavaScript เสมอในทุกครั้งที่ build โปรแกรม เป็นเหตุให้ในขั้นการ build มีความซับซ้อนเพิ่มขึ้น และกินทรัพยากรไม่菲常มาก

แม้จะมีความท้าทายเหล่านี้ TypeScript ยังคงเป็นตัวเลือกยอดนิยมสำหรับนักพัฒนาซอฟต์แวร์ที่ต้องการใช้ประโยชน์จากการเขียนโปรแกรมแบบ Static Type และ Deterministic Behavior ของ TypeScript

2.3.2 เครื่องมือ

2.3.2.1 React

React เป็นไลบรารี JavaScript ที่ถูกพัฒนาขึ้นโดย Facebook สำหรับสร้าง User Interface (UI) ที่ได้รับความนิยมอย่างแพร่หลายในการพัฒนาเว็บแอปพลิเคชัน (web applications) [?]



รูปที่ 2.1: วัฏจักรการ Render, Mount-Unmount และ Update ของ React (React Lifecycle) จาก projects.wojtekmaj.pl

React ให้เครื่องมือและโครงสร้างที่สำหรับการสร้าง UI components ที่หลากหลายและมีประสิทธิภาพ [?] หนึ่งในข้อได้เปรียบของ React อยู่ที่การใช้ Virtual DOM ที่ช่วยเพิ่มประสิทธิภาพของการ render องค์ประกอบของ UI และการสร้างแอปพลิเคชันที่มีประสิทธิภาพและสามารถบริหารจัดการ state ของแอปพลิเคชันได้ง่าย [?]

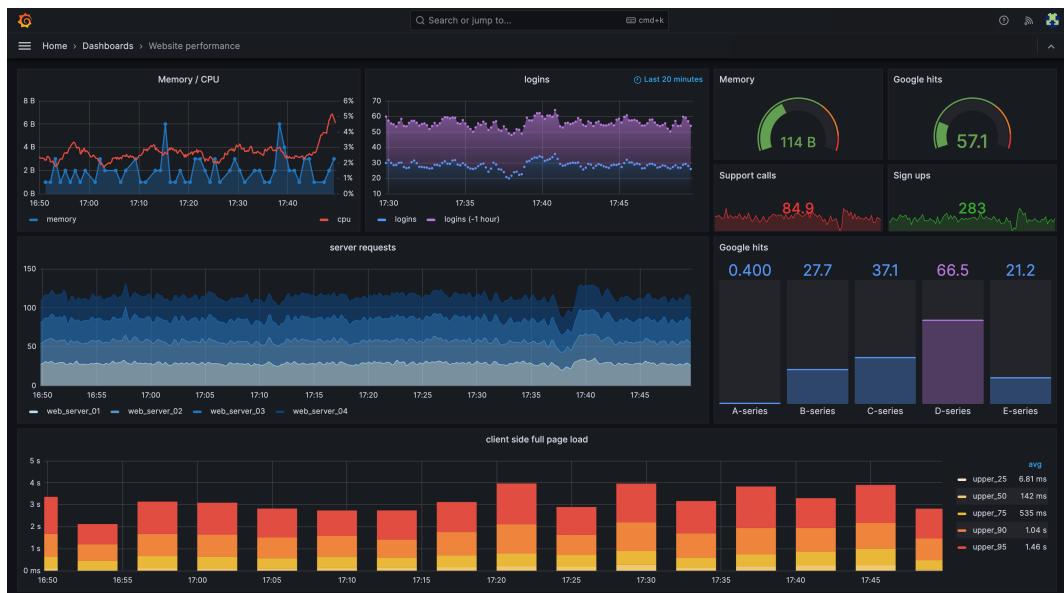
2.3.2.2 GoFiber

GoFiber เป็นเฟรมเวิร์คเบ็ดเสร็จที่เขียนขึ้นมาในภาษา Go (Golang) ที่ใช้สำหรับการพัฒนาเว็บแอปพลิเคชันแบบ high-performance พัฒนาขึ้นเพื่อเป็นตัวเลือกที่มีประสิทธิภาพสูงกว่าสำหรับเว็บแอปพลิเคชันที่ใช้งาน Go ทั่วไป โดย GoFiber ถูกออกแบบให้ง่ายต่อการใช้งานและเข้าใจ ด้วยสมรรถนะที่สูงในการดำเนินการต่าง ๆ รวมถึงการทำ Routing และการมี Middleware ต่าง ๆ ทำให้ประสิทธิภาพการทำงานของตัว GoFiber นั้นสูง [?]

จากตัวอย่างสำหรับ demo ตัว GoFiber ที่ทำโดย Ali Y. [?] แสดงให้เห็นถึงความสามารถที่โดดเด่นของ GoFiber ได้แก่ความสามารถในการทำงานที่สูง มีการจัดการ Routing ที่มีประสิทธิภาพ นอกจากนี้ GoFiber ยังมีความยืดหยุ่นในการใช้งาน Middleware และการจัดการ HTTP Context ที่เข้าใจง่าย จึงเป็นทางเลือกที่สำหรับการพัฒนาเว็บแอปพลิเคชันที่ต้องการประสิทธิภาพสูงในภาษา Go

จากความเห็นของ Chouhan A. [?] ในบทความของเขาว่า การใช้ GoFiber อาจมีข้อจำกัดในเบื้องต้นที่ยังไม่ครอบคลุมความสามารถทั้งหมดของภาษา Go และต้องพึ่งกับชุดคำสั่งที่จำกัดอย่างเล็กน้อย

2.3.2.3 Grafana



รูปที่ 2.2: ตัวอย่างหน้าจอฟ์ต์แวร์ Grafana จาก หน้า Home Page ของ Grafana

Grafana เป็นเครื่องมือสำหรับการแสดงผลข้อมูลแบบ Real-time และ Monitoring ที่มีความนิยมอย่างแพร่หลายในวงการ IT และพัฒนาซอฟต์แวร์ มันมีความสามารถในการเชื่อมต่อกับแหล่งข้อมูลต่าง ๆ เช่น ฐานข้อมูล, ระบบ Monitor, และ Web Services เพื่อนำข้อมูลมาแสดงผลในรูปแบบกราฟ, ตาราง, และแผนที่เพื่อให้ผู้ใช้งานสามารถวิเคราะห์และทำความเข้าใจข้อมูลได้ง่ายขึ้น [?]

Grafana นั้นมีความยืดหยุ่นสูงในการตั้งค่าและการปรับแต่งการแสดงผล ทำให้มีความสามารถที่ใหญ่ขึ้นในการปรับแต่ง Dashboard ตามความต้องการของผู้ใช้ [?]

จากรายงานของ Sunil K. & Savaranan C. [?] ถึงแม้ว่าความสามารถในการสร้าง Alert และ Notification จะมีจำกัดและใช้ยาก แต่ Grafana นั้นเต็มไปด้วยฟีเจอร์อื่น ๆ มากมาย ทั้งใช้งานง่ายและมีความยืดหยุ่นสูง มี option ที่สามารถกำหนดเองได้ ทำให้แห่งการแสดงภาพพิ้นกลางหลาย ซึ่งทำให้การ Monitor หรือ Visualize ข้อมูล มีประสิทธิภาพมากขึ้น

2.3.2.4 GitHub

GitHub เป็นเว็บแพลตฟอร์มที่ให้บริการ Hosting ระบบ Version Control แบบ Distributed ในรูปแบบของ Git นอกจากนี้ GitHub ยังเป็นสถานที่ที่นักพัฒนาสามารถทำงานร่วมกันและแบ่งปันโค้ดต่าง ๆ ภายใต้รูปแบบของ Repository ทำให้เป็นที่นิยมสำหรับการพัฒนาซอฟต์แวร์ Open Source และโครงการที่มีทีมพัฒนาขนาดใหญ่ [?]

GitHub มีความสามารถในการทำงานร่วมกันในโปรเจคผ่าน Git ซึ่งมีข้อได้เปรียบในการจัดการเวอร์ชันของโค้ดและการตรวจสอบการเปลี่ยนแปลง การให้บริการ Issue Tracking และ Pull Request ทำให้งานพัฒนาไปได้อย่างมีระบบและมีความเป็นระบบ [?, ?]

แต่ GitHub มีค่าบริการสำหรับการให้บริการ Version Control แต่องค์กรหรือทีมนักพัฒนาที่สูงขึ้นเมื่อเวลาผ่านไป [?]

2.3.2.5 Docker

Docker เป็นแพลตฟอร์ม Containerization ที่ช่วยให้นักพัฒนาสามารถพัฒนาและใช้งานโปรแกรมได้่ายั่งขึ้น โดยไม่ต้องสนใจว่าจะทำงานบนระบบปฏิบัติการ (OS) อะไรทั้งสิ้น [?] หลักการของการ Containerization คือการแนบ Dependencies และ Software ทั้งหมดที่ต้องใช้เข้าไปในภาษานะหรือกล่อง ๆ เดียวเป็น bundle หรือ container ทำให้สามารถเรียกใช้งานได้ทันที โดยที่ไม่ต้องติดตั้งหรือตั้งค่าเครื่อง ลงโปรแกรมหรือระบบปฏิบัติการเอง [?, ?] อีกทั้ง Docker มีความสามารถในการทำงานในรูปแบบ Isolation, มีการทำ Resource Optimization, และมีความ Portability ทำให้เป็นเครื่องมือที่ได้รับความนิยมในการพัฒนาและการดำเนินการระบบ [?]

Docker ช่วยลดปัญหาที่เกิดขึ้นจากความแตกต่างของระบบปฏิบัติการและสภาพแวดล้อมการทำงานระหว่าง Development และ Production โดยลดปัญหา "It works on my machine" ซึ่งเกิดปัญหาที่โปรแกรมจะทำงานในเครื่องคอมพิวเตอร์เครื่องหนึ่งแต่ไม่ทำงานในอีกเครื่องหนึ่ง [?] จากบทความของ Chesterwood R. [?] Docker ยังช่วยในการปรับใช้โปรแกรมในลักษณะ Microservices Architecture ทำให้การพัฒนาและการดูแลรักษาระบบเป็นไปอย่างมีประสิทธิภาพ

เช่นเดียวกับเครื่องมือก่อน ๆ Docker อาจมีความซับซ้อนสูงขึ้นและเรียนรู้การใช้งานยาก ในการกำหนดค่าและจัดการ Container หลาย ๆ ตัวพร้อมกันในระบบก็เป็นเรื่องที่ยากและใช้ความรู้มากคาดการณ์ [?] และจากความเห็นของ Yıldız H. [?] ถึงแม้ Docker จะเป็นซอฟต์แวร์ที่ light-weighted ในบางกรณี Docker อาจส่งผลกระทบต่อการทำงานโดยภาพรวมของระบบ System Performance หากบริหารไม่ดี

2.3.2.6 Terraform

Terraform เป็นเครื่องมือสำหรับ Infrastructure as Code (IaC) ที่ออกแบบมาเพื่อจัดการและกำหนดค่า Infrastructure ทั้งใน Cloud และ On-Premises อย่างมีประสิทธิภาพ [?] ผู้ใช้สามารถใช้ภาษาที่อ่านง่ายเพื่อกำหนดค่าและสร้าง Infrastructure ที่ต้องการ เช่น Virtual Machines, Networks, Databases และ Resources อื่น ๆ ในรูปแบบของ Code [?] จากบทความเรื่องข้อดีของ Terraform โดย Tripathi P. [?] ได้พูดถึงข้อดีสำคัญของ Terraform หลายประการตั้งแต่ความยืดหยุ่น, ทนทาน, ไปจนถึงความสามารถในการร่วมกับหลาย Cloud Provider อย่างมีประสิทธิภาพ

การใช้ Terraform ช่วยลดการพิมพ์ Configuration Code ที่จะใช้กำหนดค่า Infrastructure ลงมาใน Code ของโปรเจคและช่วยในการตรวจสอบสถานะและการดำเนินการ Infrastructure อย่างทันที แต่ผู้ใช้ต้องเรียนรู้และทำความเข้าใจการใช้งาน Terraform อย่างลึกซึ้ง เพื่อให้การใช้งานมีประสิทธิภาพ [?]

2.3.2.7 Kubernetes

Kubernetes เป็นระบบ Orchestrator ที่ใช้ในการจัดการและควบคุม Containers ในรูปแบบ Cluster มีความสามารถในการปรับขยายและจัดการสถานะของ Containers ต่าง ๆ ในทันที ทำให้เป็นเครื่องมือที่สำหรับการปรับใช้และการจัดการ Microservices และ Applications ที่ทำงานบน Containers อีกทั้ง Kubernetes ช่วยลดภาระงานในการสร้าง, ดูแล, และปรับขยายระบบที่มีมากถึงหลาย hundreds of Containers [?]

ความสามารถในการตรวจสอบและแก้ไขข้อผิดพลาดในการทำงานของ Containers รวมถึงการปรับเปลี่ยนอัตโนมัติในกรณีที่มี Containers ล้ม ทำให้ระบบยืดหยุ่นและทนทานมากขึ้น [?] แต่เช่นเดียวกับตัว Docker เอง การจัดการ Kubernetes อาจมีความซับซ้อนและต้องการเรียนรู้การใช้งานในระดับที่สูง

2.3.3 โครงสร้างพื้นฐาน

2.3.3.1 MySQL

MySQL เป็นระบบจัดการฐานข้อมูลแบบ Relational Database Management System (RDBMS) ที่เป็น Open Source และได้รับความนิยมอย่างแพร่หลาย มีความสามารถในการจัดเก็บและจัดการข้อมูลในรูปแบบตารางที่สอดคล้องกับโครงสร้างของฐานข้อมูล [?] ทำให้เป็นทางเลือกที่ดีสำหรับโปรเจคที่มีโครงสร้างข้อมูลที่แน่นอน เนื่องด้วยความเป็นที่นิยมและการที่ยังมีคนใช้ประโยชน์จากตัวซอฟต์แวร์ตันอยู่ MySQL มี Community Edition ที่ให้บริการฟรีและมีการพัฒนาที่ต่อเนื่องอย่าง [?]

ข้อดีของ MySQL รวมถึงประสิทธิภาพในการค้นหาและดึงข้อมูล, รองรับการใช้งานที่หนักหน่วง, และมีความปลอดภัยในการจัดการข้อมูล แต่ก็เป็น MySQL จะเหมาะสมกับทุกโปรเจค บางครั้ง MySQL อาจจะไม่เหมาะสมกับโปรเจคที่มีความซับซ้อนและต้องการการจัดการข้อมูลในมิติที่หลากหลาย

2.3.3.2 InfluxDB

InfluxDB เป็นฐานข้อมูลแบบ Time Series Database ที่ออกแบบมาเพื่อจัดเก็บและจัดการข้อมูลที่มีการเปลี่ยนแปลงตลอดเวลา นิยมถูกใช้ในงานที่ต้องการจัดเก็บข้อมูลการวัดหรือเวลาจริง เช่น ข้อมูลเซ็นเซอร์ หรือข้อมูลการทดสอบและข้อมูลการวิเคราะห์จากต้นทางที่ส่งมาพร้อมเวลา timestamp [?] มีความยืดหยุ่นและสามารถทำงานร่วมกับหลายแพลตฟอร์ม [?]

ความเร็วในการทำงานและการดึงข้อมูลใน InfluxDB เป็นจุดเด่นที่สำคัญ โดยเฉพาะเมื่อมีความต้องการในการจัดเก็บและดึงข้อมูลจำนวนมากในช่วงเวลาที่สั้น แต่ก็อาจมีข้อจำกัดในการจัดเก็บข้อมูลที่ไม่ใช่ Time Series และการรองรับขนาดข้อมูลที่ใหญ่

2.3.3.3 Prometheus

Prometheus เป็นระบบการตรวจสอบและติดตาม (monitoring and alerting) ที่ถูกออกแบบมาสำหรับระบบที่มีโครงสร้างที่ยืดหยุ่น และต้องการการดูแลที่ส่วนตัว [?] มีความสามารถในการเก็บข้อมูลและวิเคราะห์ข้อมูลเพื่อตรวจสอบปัญหาและเป็นเครื่องมือที่มีประสิทธิภาพในการจัดการระบบที่ใหญ่

Prometheus มีการส่งเสริมการใช้งานร่วมกับ Kubernetes และมีการส่งเสริมที่ดีในการใช้งานที่แพร่หลายในระบบคลาวด์

ความยืดหยุ่นและการติดตามเหตุการณ์ในเวลาที่เป็นเรียลไทม์เป็นจุดเด่นสำคัญของ Prometheus, ทำให้เป็นเครื่องมือที่เหมาะสมสำหรับการค้นหาและแก้ไขปัญหาที่เกิดขึ้นทันที แต่ในระบบที่มีขนาดใหญ่และต้องการการจัดเก็บข้อมูลที่ยาวนาน แต่พื้นที่เก็บข้อมูลของ Prometheus อาจมีความจำกัดและข้อมูลอาจล้างเกินได้

2.3.3.4 RabbitMQ

RabbitMQ เป็นระบบ Message Broker ที่มีคุณสมบัติและบริการในการจัดการและส่งข้อมูลระหว่างแอปพลิเคชันต่าง ๆ [?] ในรูปแบบที่มีประสิทธิภาพ ด้วยความที่ซอฟต์แวร์เป็น Open Source ทำให้ได้รับความนิยมเป็นอย่างสูง ได้ถูกนำมาใช้การทำงานในสภาพแวดล้อมที่ต้องการการกระจายข้อมูลและการคอมมูนิเคชัน อีกทั้งการใช้ RabbitMQ ช่วยลดความซับซ้อนในการสื่อสารระหว่างบริการและแอปพลิเคชัน, ทำให้ระบบสามารถทำงานได้อย่างยืดหยุ่นและมีประสิทธิภาพ [?]

ความยืดหยุ่นของ RabbitMQ ทำให้เป็นเครื่องมือที่มีประสิทธิภาพสำหรับการทำ Message Queue และการจัดการกับการแลกเปลี่ยนข้อมูลที่ต้องการการทำงานแบบอนุกรม [?] ในขณะเดียวกันการจัดการ RabbitMQ ในสภาพแวดล้อมที่มีใช้งานการสื่อสารมาก ๆ อาจมีความซับซ้อน [?]

บทที่ 3 วิธีการดำเนินงาน

ในบทที่ 3 จะกล่าวถึงภาพรวมของซอฟต์แวร์ของโครงงาน โครงสร้างระบบที่ทางคณะผู้จัดทำโครงการได้วางเอาไว้ ลักษณะฐานข้อมูลที่ใช้ในการจัดเก็บข้อมูล การใช้งานระหว่างระบบออกแบบมาตอบโจทย์ผู้ใช้งาน รวมไปถึงการจัดการภัยในระบบที่ได้ออกแบบไว้อย่างดี

สถาปัตยกรรมระบบที่ได้วางไว้มีความเป็นระเบียบและเหมาะสม ระบบฐานข้อมูลออกแบบมาให้เรียบง่าย สมมพسانกับการใช้เทคโนโลยีที่เหมาะสมและทันสมัย ทางคณะผู้จัดทำโครงการ เชื่อมั่นว่าซอฟต์แวร์ที่ออกแบบมา จะสามารถทำงานอย่างมีประสิทธิภาพ ประมวลผลเร็ว

จากสถาปัตยกรรมระบบที่ได้ออกแบบไว้ใน [หัวข้อสถาปัตยกรรมระบบ](#) การใช้ตัวประสานงาน (หรือ “nginx ingress” ในรูปที่ 3.2) ระหว่างผู้ใช้ (Client) กับระบบ จะสามารถแบ่งเบาและแจกจ่ายคำขอจากผู้ใช้อย่างมีประสิทธิภาพ และอีกทั้งการใช้ตัวสื่อสาร (Message Broker หรือ “RabbitMQ” ในรูปที่ 3.2) ระหว่างตัวซอฟต์แวร์หลักกับตัวโมดูลตรวจงานเขียนโปรแกรม (Grading Core) จะทำให้การจัดการคิวคำขอตรวจงานเขียนโปรแกรมย่างขึ้น ทำให้การตรวจให้คะแนน ประกาศผลคะแนนและบันทึกลงฐานข้อมูล ดำเนินการตามลำดับอย่างต่อเนื่อง

จากเหตุผลข้างต้น ในเชิงอุดมคุณ ซอฟต์แวร์ที่ทางกลุ่มผู้จัดทำโครงการออกแบบมาจะสามารถรับมือกับคำขอจากผู้ใช้ได้จำนวนมหาศาล รองรับจำนวนผู้ใช้เข้ามาใช้งานในระบบพร้อมกันได้จำนวนมาก

3.1 รายละเอียดของโครงงาน

3.1.1 ข้อกำหนดและความต้องการของระบบ

ในส่วนนี้ จะกล่าวถึงข้อกำหนดของซอฟต์แวร์หรือ User Requirements ซึ่งประกอบด้วยการทำงานในส่วนต่าง ๆ แยกประเภทเป็นตามหมวดหมู่ผู้ใช้ดังนี้ เจ้าของระบบหรือผู้ดูแลระบบ (Admin), เจ้าของห้องเรียน เจ้าของกลุ่มเรียนหรืออาจารย์ผู้สอน (Owner), ผู้ช่วยเจ้าของห้องหรือผู้ช่วยสอน (Moderator), นักศึกษา (Student), ผู้เข้าร่วมกิจกรรมหรือการแข่งขัน (Attendee), ผู้ใช้ทั่วไป (Registered User) และบุคคลทั่วไปหรือบุคคลภายนอก (Guest User)

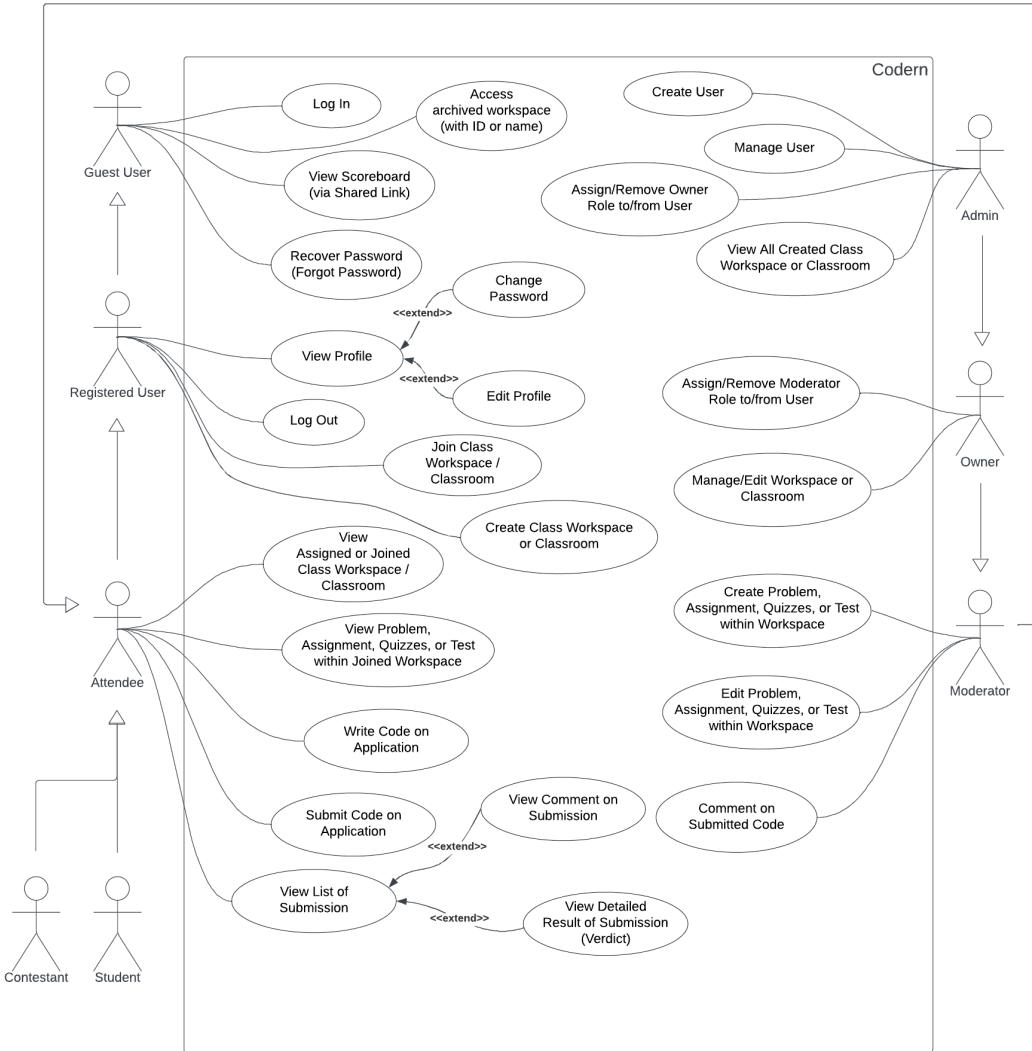
ตารางที่ 3.1 ข้อกำหนดของซอฟต์แวร์ (1)

| Feature | Guest User | Registered User | Student | Attendee | Moderator | Owner | Admin |
|----------------------------|------------|-----------------|---------|----------|-----------|-------|-------|
| เข้าสู่ระบบ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ออกจากระบบ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| กู้รหัสผ่าน | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ดูข้อมูลส่วนตัว | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| แก้ไขข้อมูลส่วนตัว | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| เข้าร่วมห้องหรือกลุ่มเรียน | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

ตารางที่ 3.2 ข้อกำหนดของซอฟต์แวร์ (2)

| Feature | Guest User | Registered User | Student | Attendee | Moderator | Owner | Admin |
|--|------------|-----------------|---------|----------|-----------|-------|-------|
| ดูรายการห้องที่ได้เข้าร่วม ทั้งหมด | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| ดูโจทย์และภาระงานใน ห้องที่ได้เข้าร่วม | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| เขียนโปรแกรมบน แอปพลิเคชัน | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| เขียนโปรแกรมบน แอปพลิเคชัน | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| ส่งไฟล์โปรแกรมเข้าตรวจสอบ บนแอปพลิเคชัน | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| ดูรายการส่งไฟล์ โปรแกรมทั้งหมด | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| ดูข้อความเห็น/คำแนะนำ/ คำติชมของรายการส่งไฟล์ นั้น ๆ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| ดูผลการตรวจและราย ละเอียดของรายการส่งไฟล์ โปรแกรมนั้น ๆ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| ดูตารางคะแนนผ่านลิงก์ แชร์ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ดูตารางคะแนนในห้องที่ได้ เข้าร่วม | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| สามารถที่จะดูข้อมูลห้อง เก่าที่เคยได้เข้าร่วม | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| สามารถที่จะดูโจทย์และ ภาระงาน ประวัติการใช้ งาน ข้อมูลของห้องเก่าที่ เคยได้เข้าร่วม | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| สร้างโจทย์และภาระงาน ในห้อง | | | | | ✓ | ✓ | ✓ |
| แก้ไขหรือจัดการโจทย์และ ภาระงานในห้อง | | | | | ✓ | ✓ | ✓ |
| ส่งข้อความเห็น/คำ แนะนำ/คำติชมของ รายการส่งไฟล์นั้น ๆ | | | | | ✓ | ✓ | ✓ |

3.1.2 กรณีการใช้งาน (Use Cases)



รูปที่ 3.1: ภาพแผนผังกรณีใช้งาน วัดด้วย LucidChart อ่านคำบรรยายได้เพิ่มเติมได้ในหัวข้อถัดไป

3.1.3 คำบรรยายกรณีใช้งาน (Use Cases Narratives)

จากรูปที่ 3.1 ในหัวข้อที่แล้ว เป็นรูปที่แสดงความสัมพันธ์ระหว่างผู้ใช้งานและซอฟต์แวร์ ในโครงงานซอฟต์แวร์นี้ได้แบ่งประเภทผู้ใช้งานเป็น 6 ประเภท ได้แก่บุคคลทั่วไป (Guest User), ผู้ใช้ทั่วไป (Registered User), นักศึกษา (Student), ผู้เข้าร่วมกิจกรรมหรือการแข่งขัน (Attendee), เจ้าของห้องหรืออาจารย์ผู้สอน (Owner), ผู้ช่วยเจ้าของห้องหรือผู้ช่วยสอน (Moderator) และเจ้าของระบบหรือผู้ดูแลระบบ (Admin) โดยแต่ละประเภทผู้ใช้งานมีเป้าหมายหลัก และกรณีการใช้งานดังต่อไปนี้

3.1.3.1 Guest User (บุคคลทั่วไป)

บุคคลทั่วไป สามารถเข้าใช้งาน feature พื้นฐานของซอฟต์แวร์ได้ดังนี้

1. “Log In” คือ การล็อกอินเข้าสู่ระบบ
2. “View Scoreboard (via Shared Link)” คือ การเข้าถึงตารางคะแนนด้วยลิงก์ที่ถูกแชร์มา
3. “Recover Password” คือ การรีเซ็ตรหัสผ่าน กรณีที่หลืมหรือลืมรหัส
4. “Access Archived Workspace (with ID or Name)” คือ การเข้าถึงห้องเรียน กลุ่มเรียนเก่าที่ผู้ใช้เคยได้เข้าร่วม ที่ได้ถูกเก็บรักษาไว้ (archived) ผ่านการค้นด้วยรหัสหรือชื่อ

3.1.3.2 Registered User (ผู้ใช้ทั่วไป)

ผู้ใช้ทั่วไปที่มีบัญชีในระบบ (Admin อาจได้สร้างให้ หรือไม่ก็สมัครเอง) โดยผู้ใช้ทั่วไปสามารถเข้าถึง feature พื้นฐานของซอฟต์แวร์ได้เหมือนกับบุคคลทั่วไป (Guest User) และยังสามารถเข้าใช้ feature เพิ่มเติมได้ดังต่อไปนี้

1. “View Profile” คือ การเข้าดูข้อมูลส่วนตัวของตัวผู้ใช้เอง
2. “Change Password” คือ การที่สามารถจะเปลี่ยนรหัสผ่านบัญชีตนเองได้
3. “Edit Profile” คือ การที่สามารถจะแก้ไขข้อมูลส่วนตัวของบัญชีตนเองได้
4. “Join Workspace” คือ การที่สามารถจะเข้าร่วมกลุ่มเรียน ห้องเรียนเองได้
5. “Log Out” คือ การล็อกอутออกจากระบบ
6. “Create Class Workspace or Classroom” คือ การที่สามารถจะสร้างห้องเรียนหรือกลุ่มเรียนได้

3.1.3.3 Attendee (ผู้เข้าร่วมกิจกรรม)

ผู้เข้าร่วมกิจกรรมที่มีบัญชีในระบบ (Registered User) และยังสามารถเข้าใช้ feature เพิ่มเติมได้ดังต่อไปนี้

1. “View Assigned or Joined Class Workspace / Classroom” คือ การเข้าถึงรายการห้องเรียน กลุ่มเรียนทั้งหมด ที่ได้เข้าร่วม หรือถูกดึงเข้า
2. “View Problems, Assignments, Quizzes, or Tests with Joined Workspace” คือการเข้าถึงโจทย์ปัญหา ภาระงาน และข้อสอบทั้งหมดในกลุ่มเรียนหรือห้องเรียนที่ได้เข้าร่วมหรือถูกดึงเข้า
3. “Write Code on Application” คือ การที่สามารถจะเขียนโปรแกรมบนตัวซอฟต์แวร์ได้
4. “Submit Code on Application” คือ การที่สามารถจะส่งโปรแกรมให้ซอฟต์แวร์ไปตรวจ
5. “View List of Submission” คือ การเข้าถึงรายการที่แสดงการส่งโปรแกรมให้ซอฟต์แวร์ไปตรวจ ที่ตัวนักศึกษาหรือผู้เข้าร่วม กิจกรรม/การแข่งขัน เป็นคนส่งทั้งหมด
6. “View Comment on Submission” คือ การเข้าถึงข้อความคิดเห็น คำแนะนำ คำติชมเกี่ยวกับงานเขียนโปรแกรมที่ได้ส่งไป ที่อาจารย์ผู้สอน ผู้ช่วยสอน ได้พิมพ์เอาไว้ ในการส่งโปรแกรมที่ส่งตรวจนัดๆ ๆ
7. “View Detailed Result of Submission (Verdict)” คือ การเข้าถึง รายละเอียดและผลตรวจแบบละเอียดของโปรแกรมที่ส่ง ตรวจในแต่ละครั้ง

3.1.3.4 Student (นักเรียนหรือนักศึกษา)

นักเรียนหรือนักศึกษา ก็คือผู้เข้าร่วมกิจกรรมที่ได้เข้าร่วม (joined) หรือถูกตัดเข้า (assigned) ห้องเรียน กลุ่มเรียน (workspace) ดังนั้น นักเรียนหรือนักศึกษาก็จะเข้าถึง feature พื้นฐานของซอฟต์แวร์ได้เหมือนกับผู้เข้าร่วมกิจกรรม (Attendee) เมื่อ่อนกันหมดทุกประการ

3.1.3.5 Contestant (ผู้เข้าร่วมการแข่งขัน)

ผู้เข้าร่วมการแข่งขัน ก็คือผู้เข้าร่วมกิจกรรมที่ได้เข้าร่วม (joined) หรือถูกตัดเข้า (assigned) ห้องเรียน กลุ่มเรียน (workspace) เช่นเดียวกัน ดังนั้นผู้เข้าร่วมการแข่งขันเข้าถึง feature พื้นฐานของซอฟต์แวร์ได้เหมือนกับผู้เข้าร่วมกิจกรรม (Attendee) เมื่อ่อนกันหมดทุกประการ หรือได้ feature เทียบเท่ากับนักเรียนหรือนักศึกษา

3.1.3.6 Moderator (ผู้ช่วยเจ้าของห้องหรือผู้ช่วยสอน)

ผู้ช่วยเจ้าของห้องหรือผู้ช่วยสอน เป็นผู้ใช้ที่ถูกใจเจ้าของห้องหรืออาจารย์ผู้สอนหรือเจ้าระบบแต่ตั้งให้มาเป็นผู้ช่วยบริหารห้องเรียนหรือกลุ่มเรียน ผู้ใช้หมวดหมุนี้สามารถเข้าถึง feature ของซอฟต์แวร์ได้เหมือนกับนักศึกษาหรือผู้เข้าแข่งขันฯ (Student or Attendee) แต่ก็เข้าถึง feature เพิ่มเติมได้ดังนี้

1. “Create Problems, Assignments, Quizzes, or Test within Workspace” คือ การที่สามารถสร้างโจทย์ปัญหา ภาระงานและออกข้อสอบภายในห้องเรียน หรือกลุ่มเรียนที่ตนได้คุมอยู่ได้
2. “Edit Problems, Assignments, Quizzes, or Test within Workspace” คือ การที่สามารถแก้ไขโจทย์ปัญหา ภาระงานและออกข้อสอบภายในห้องเรียน หรือกลุ่มเรียนที่ตนได้คุมอยู่ได้
3. “Comment on Submitted Code” คือ การที่สามารถจะให้ความเห็น ให้คำแนะนำหรือให้คำติชม แก่งานเขียนโปรแกรมที่นักศึกษา หรือผู้เข้าร่วมการแข่งขันฯ ได้ส่งเข้ามาในระบบได้

3.1.3.7 Owner (เจ้าของห้องหรืออาจารย์ผู้สอน)

เจ้าของห้องหรืออาจารย์ผู้สอน คือผู้ใช้ที่สามารถสร้างและบริหารจัดการห้องเรียนหรือกลุ่มเรียนได้ ผู้ใช้หมวดหมุนี้สามารถเข้าถึง feature ของซอฟต์แวร์ได้เหมือนกับผู้ช่วยเจ้าของห้อง (Moderator) และเข้าถึง feature เพิ่มเติมได้ดังนี้

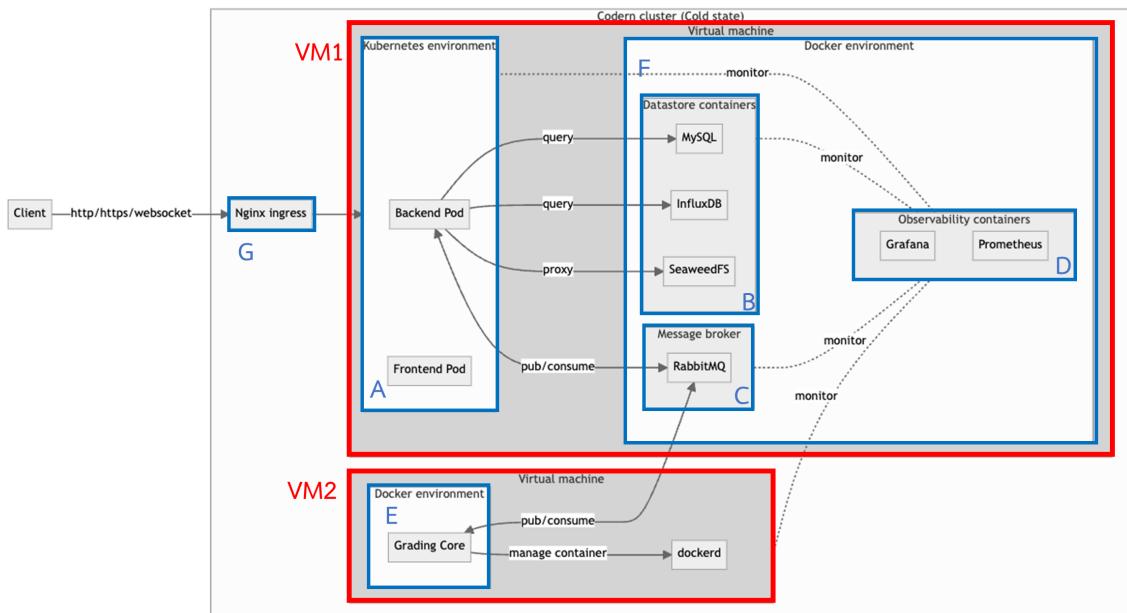
1. “Assign/Remove Moderator Role to/from User” คือ ความสามารถที่จะแต่งตั้งหรือลดบทบาทผู้ใช้ทั่วไปให้มาเป็นผู้ช่วยสอน หรือผู้ช่วยเจ้าของห้อง (moderator)
2. “Manage or Edit Workspace or Classroom” คือ ความสามารถที่จะแก้ไข บริหารจัดการ (ลบ) ห้องเรียนหรือกลุ่มเรียนทั้งหมด ที่ตนเป็นเจ้าของ

3.1.3.8 Admin (เจ้าของระบบหรือแอดมิน)

เจ้าของระบบหรือแอดมิน สามารถเข้าถึง feature ของซอฟต์แวร์ได้เหมือนกับเจ้าของห้องหรืออาจารย์ผู้สอน (Owner) แต่เข้าถึง feature พิเศษเพิ่มเติมได้ ดังนี้

1. “Create User” คือ ความสามารถในการสร้างบัญชีผู้ใช้ได้
2. “Manage User” คือ การที่สามารถที่จะบริหารจัดการ (ลบ) ข้อมูลบัญชีผู้ใช้ทั้งหมดได้
3. “Assign/Remove Owner Role to/from User” คือ ความสามารถที่จะแต่งตั้งหรือลดบทบาทผู้ใช้ ให้มาเป็นเจ้าของห้อง (owner)
4. “View All Created Class Workspace or Classroom” คือ การเข้าถึงรายการแสดงผลห้องเรียนหรือกลุ่มเรียนทั้งหมดที่ได้สร้างขึ้นในระบบ

3.2 สถาปัตยกรรมระบบ



รูปที่ 3.2: ภาพแผนผังสถาปัตยกรรมระบบของซอฟต์แวร์ วัดด้วย Mermaid JS

แผนภาพในรูปที่ 3.2 แสดงสถาปัตยกรรมระบบแบบกระจาย เครื่องเสมือนสองเครื่อง (เรียกว่า VM1 และ VM2) ภายใน cluster ที่เรียกว่า “Codern Cluster”

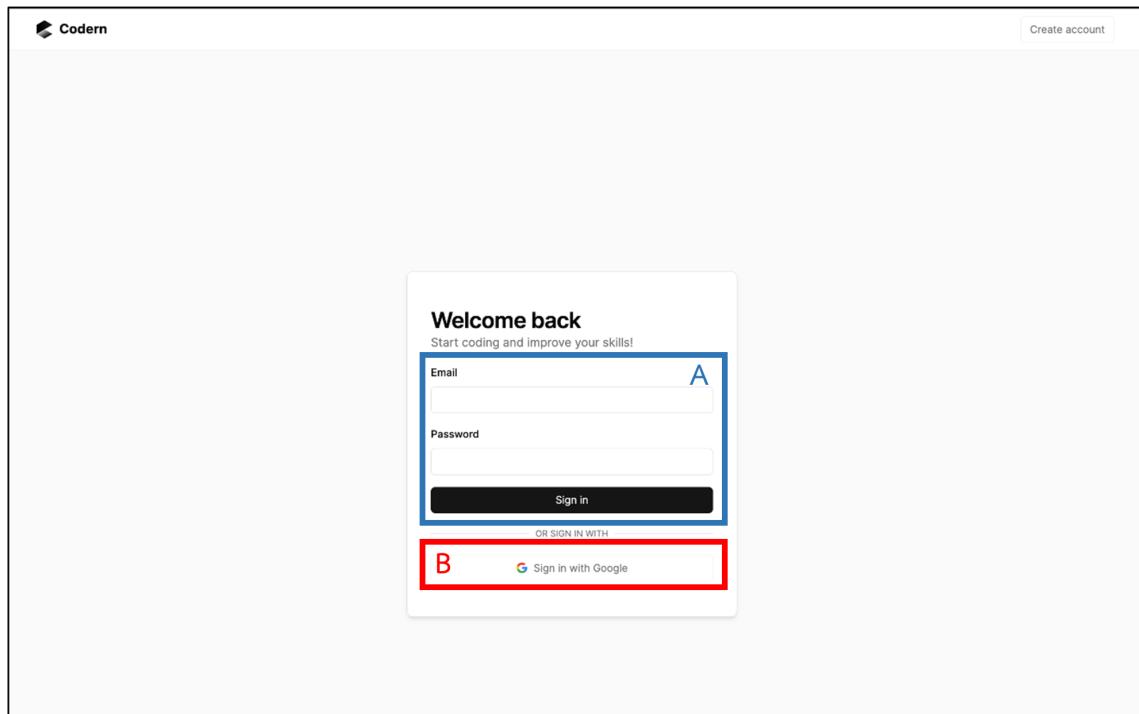
ใน VM1 มีการวางรูปแบบซอฟต์แวร์เป็นรูปแบบ Kubernetes (กรอบ A ในรูป) ที่มี pod ชื่อ backend และ frontend pod โดย pod เหล่านี้จะติดต่อสื่อสารกับส่วนประมวลผลต่างอื่นๆ ใน container ที่ host โดย Docker ในเครื่องเดียวกัน Docker (กรอบ F ในรูป) ในเครื่อง VM1 จะ host คอนเทนเนอร์ 3 อัน ได้แก่แหล่งเก็บข้อมูลหรือฐานข้อมูลต่างๆ (กรอบ B) อาทิเช่น MySQL, InfluxDB และ SeaweedFS, ตัว message broker (กรอบ C) หรือ RabbitMQ ซึ่งคือโมดูลที่เปิดบริการสำหรับการส่งข้อมูลและสื่อสารไปยัง VM2 เพื่อส่งไฟล์โปรแกรมไปตรวจ และสุดท้ายก็คือ monitoring tool อย่าง Grafana และ Prometheus (กรอบ D) ซึ่งเป็นเครื่องมือสำหรับการตรวจสอบ สอดส่องดูแล วิเคราะห์หาข้อบกพร่อง และวัดประสิทธิภาพการทำงานของแต่ละส่วนของซอฟต์แวร์

ใน VM2 ที่มี Docker ที่ host ตัว container อยู่ เช่นกัน โดย container นี้จะเปิด Grading Core ไว้ ซึ่งทำหน้าที่เป็นตัวโนดโปรแกรมสำหรับตรวจไฟล์โปรแกรมที่ผู้ใช้ได้ส่งเข้ามาจาก VM1 ผ่าน message broker หลังจากตรวจสอบเสร็จก็จะส่งผลตรวจลับไปยัง message broker

คำขอของผู้ใช้จะถูกประมวลผลโดยตัวควบคุม Nginx Ingress (กรอบ G ในรูป) เพื่อส่งคำขอของ client ไป Kubernetes cluster ที่ได้ host ไว้ เพื่อประมวลผล

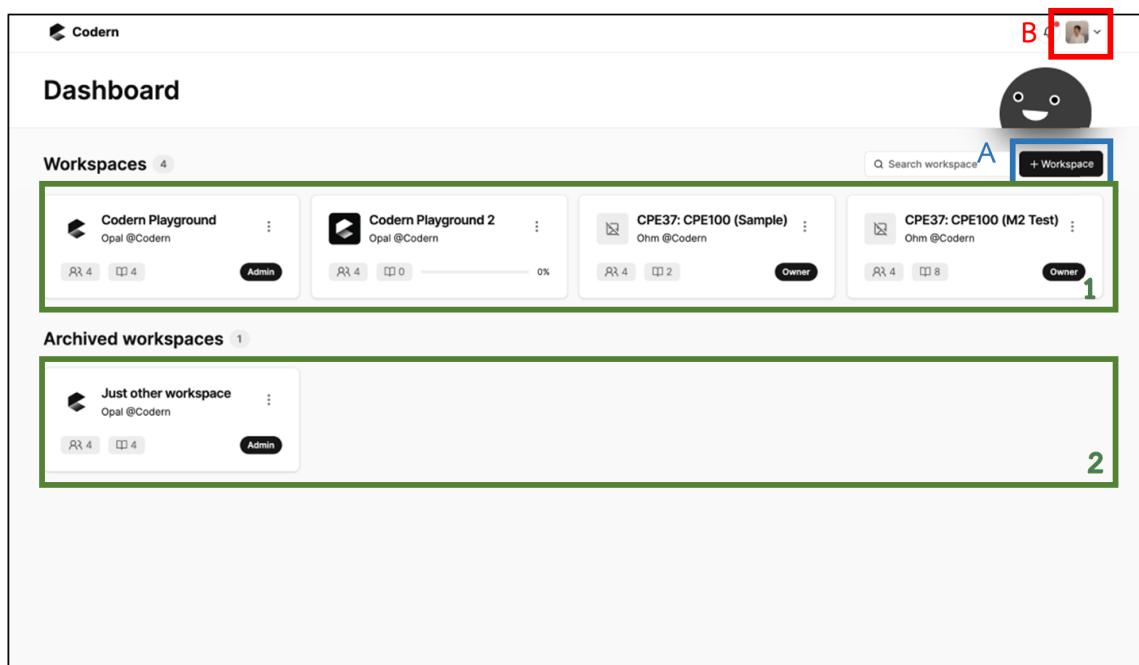
3.3 ส่วนประสานต่อผู้ใช้

สำหรับในส่วน UI ทางคณะผู้จัดทำจะนำเสนอบนหน้าส่วนประสานต่อผู้ใช้ที่ได้ออกแบบไว้ทั้งหมด ดังนี้



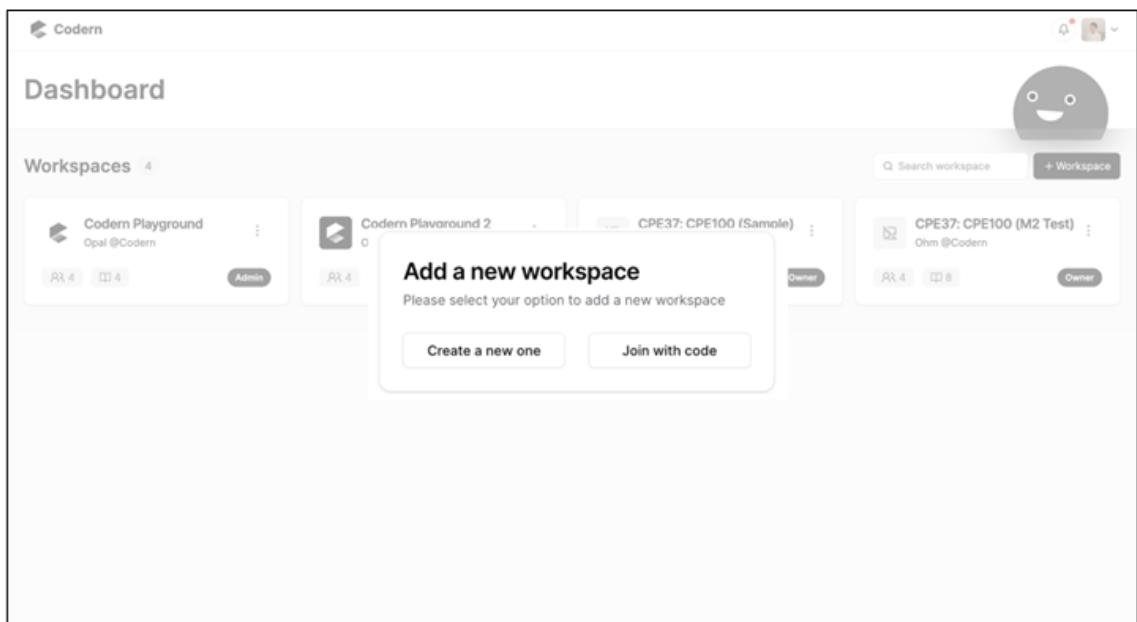
รูปที่ 3.3: ส่วนประสานต่อผู้ใช้ หน้าเข้าสู่ระบบ

ในรูปที่ 3.3 เป็นหน้าสำหรับล็อกอินเข้าระบบไปใช้งานทั้งหมดของซอฟต์แวร์ โดยสามารถล็อกอินด้วยชื่อผู้ใช้และรหัสผ่าน (กรอบสีเงิน A) และล็อกอินผ่านกุเก็ล (กรอบสีแดง B)



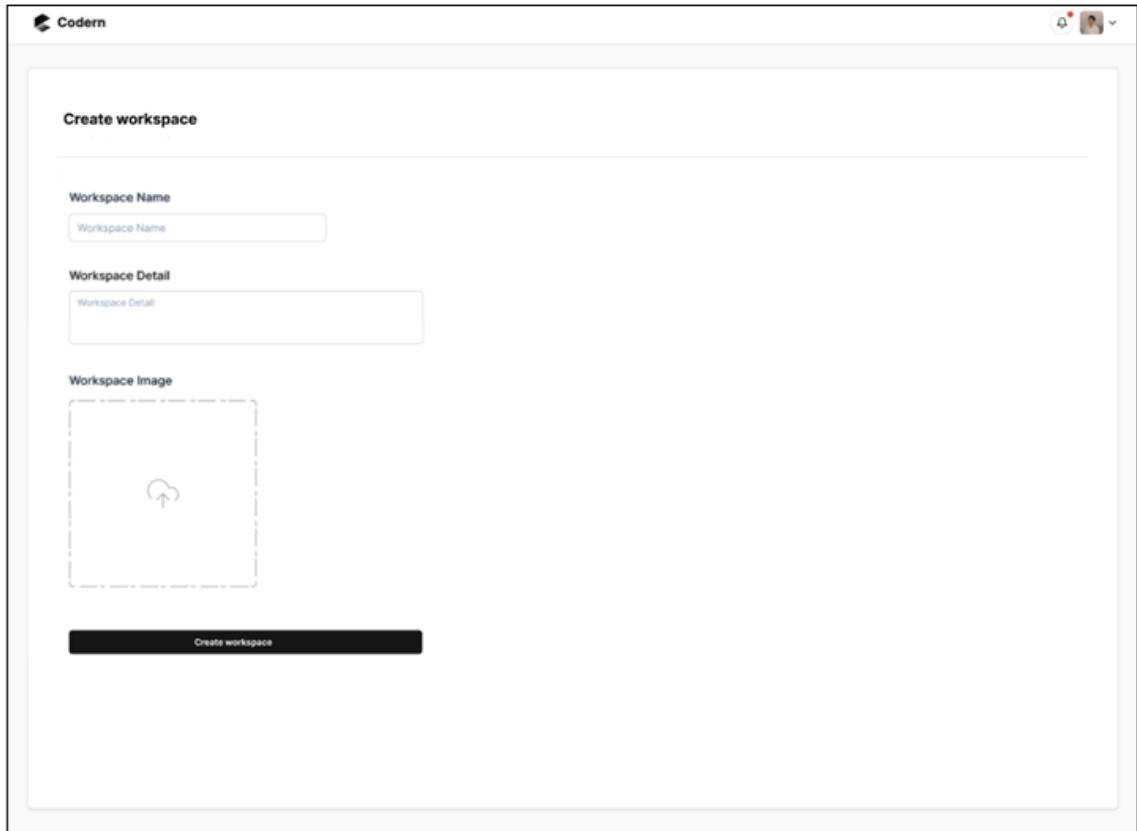
รูปที่ 3.4: ส่วนประสานต่อผู้ใช้ หน้าแสดงความคุม

เมื่อเข้าสู่ระบบเรียบร้อยแล้ว จะกันนั่งจะไปสู่หน้าแสดงรายการห้องทั้งหมดที่ผู้ใช้ได้เข้าร่วม ดังรูปที่ 3.4 ส่วนต่อประสานกับผู้ใช้หน้าแสดงรายการห้องที่ผู้ใช้ได้เข้าร่วมอยู่ มีทั้งการแสดงห้องเรียนหรือกลุ่มเรียนทั้งหมดที่เคยเข้าร่วมในส่วน “All Workspaces” (โฉนดที่ 1) และแสดงห้องเรียนหรือกลุ่มเรียนที่ได้สืบสุดคอร์สเรียนหรือสืบสุกการใช้งานไปแล้วในส่วน “Archived workspaces” (โฉนดที่ 2) ผู้ใช้สามารถที่จะกดเข้าร่วมห้องใหม่ได้ที่ปุ่ม “Add Workspace” ตรงมุมขวาบนในกรอบสีน้ำเงิน A และสามารถกดเพื่อแก้ไขข้อมูลเกี่ยวกับบัญชีของตัวเองได้ที่ปุ่ม “ไอคอนavatarรูปของตนบริเวณมุมขวาบนสุดในกรอบสีแดง B

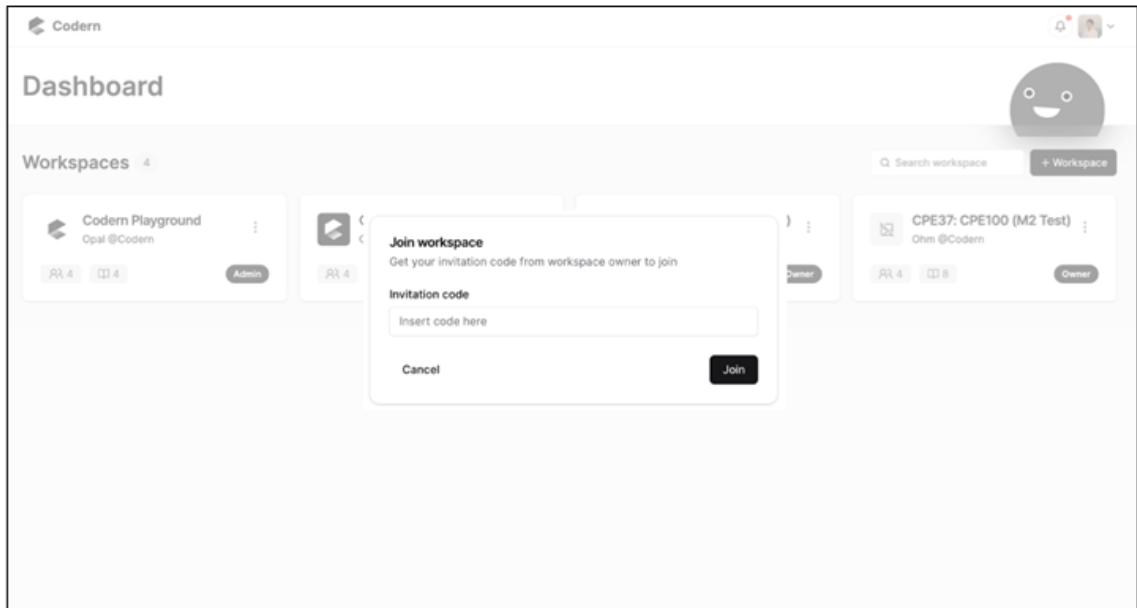


รูปที่ 3.5: ส่วนประสานต่อผู้ใช้ หน้าແນວຄວບຄຸມ หลังกดปຸ່ມເພີ່ມ Workspace

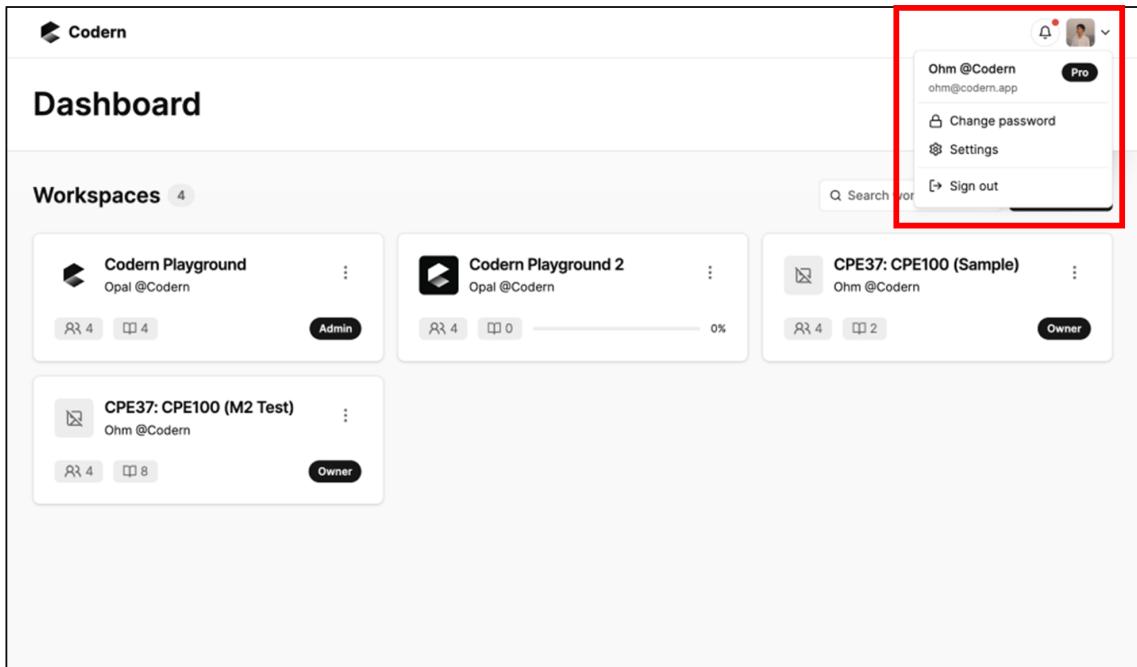
โดยเมื่อผู้ใช้กดเข้าร่วมห้องใหม่ที่ปุ่ม “Add Workspace” แล้ว จะปรากฏกล่องข้อความแจ้งเตือนเพื่อแสดงสองทางเลือก ดังภาพที่ 3.5 ได้แก่ “Create a new one” ซึ่งผู้ใช้จะสามารถกดเพื่อเข้าสู่หน้าการสร้างห้องเรียนใหม่ โดยกรอกข้อมูลชื่อของห้องเรียน รายละเอียด และรูปภาพ เมื่อกดปุ่ม “Create a new one” แล้ว ห้องเรียนก็จะถูกสร้าง ดังภาพที่ 3.6 และ “Join with code” ซึ่งผู้ใช้สามารถกดเพื่อกรอกรหัสเข้าร่วมห้องเรียนที่ได้มีการสร้างไว้ก่อนอยู่แล้วได้ ดังภาพที่ 3.7



รูปที่ 3.6: ส่วนประสานต่อผู้ใช้ หน้าແຜງຄວບຄຸມ ພັນກົດປຸ່ມ "Create a new one" ໃນຮູບທີ 3.5

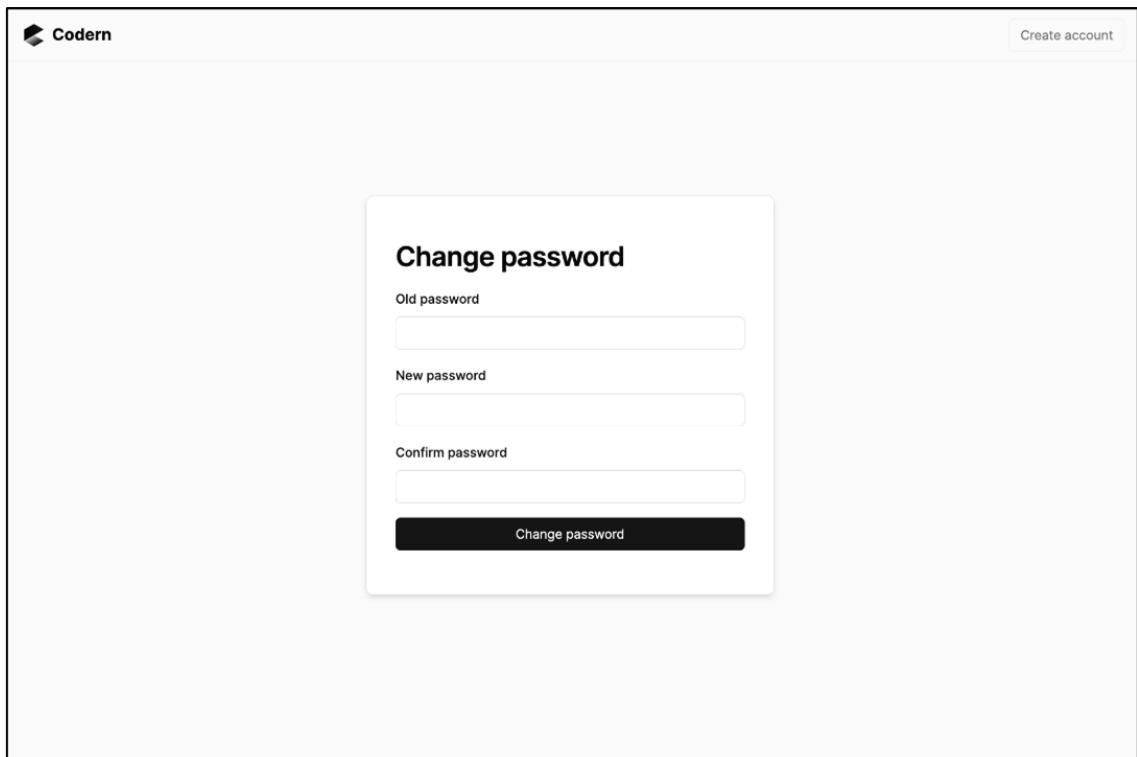


ຮູບທີ 3.7: ສ່ວນປະສານຕ່ອຸ່ນໃຈ້ ບໍ່ມີເອົາໄຫວ້ ພັນກົດປຸ່ມ "Join with code" ໃນຮູບທີ 3.5



รูปที่ 3.8: ส่วนประisanต่อผู้ใช้ หน้าແຜງຄວາມ หากกดรูป Avatar ตรงมุมขวาบน ในรูปที่ 3.5

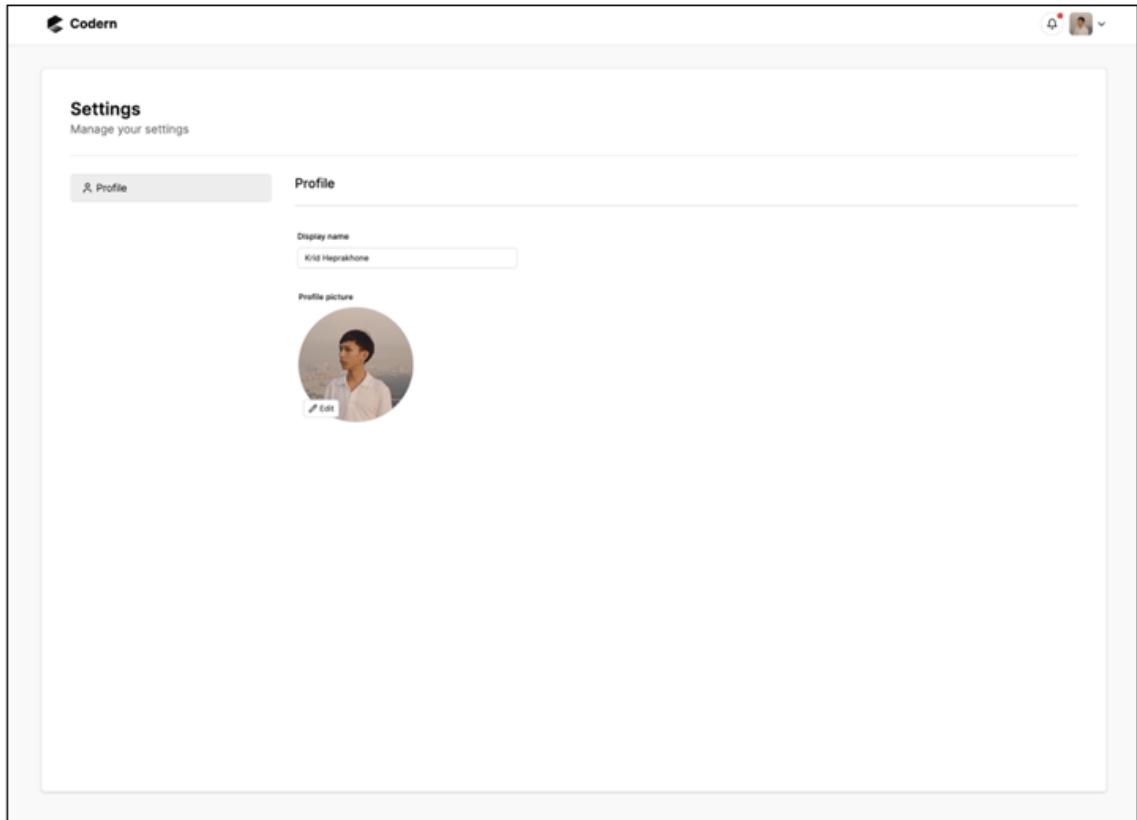
หากกดปุ่ม "ໂອຄອນອວາຕາຣຽບ" ของดັນບຣີເລັນມຸນຂວາບສຸດໃນກຣອບສີແແງ B ในກາພທີ 3.5 ກົຈະປຣກູ້ Dropdown ລົງມາດັ່ງກາພທີ 3.8 (ກຣອບສີແແງ) ປະກອບດ້ວຍ 3 ເມື່ອງ "ເທົ່າແກ່" "Change password", "Settings" ແລະ "Sign out"



รูปที่ 3.9: ส่วนประisanต่อผู้ใช้ หน้าແຜງຄວາມ หากกดປຸ່ມ "Change Password" ในรูปที่ 3.8

เมื่อกดปุ่ม "Change password" ในรูปที่ 3.8 จะนำไปสู่หน้าเปลี่ยนรหัสผ่านเข้าบัญชี ดังภาพที่ 3.9 โดยจะเปลี่ยนรหัสผ่านบัญชีได้โดยการกรอกรหัสผ่านเก่าให้ถูกต้อง จากนั้นก็กรอกรหัสผ่านใหม่ที่ต้องการ และทำการยืนยันรหัสผ่านใหม่อีกครั้ง เมื่อกรอกรหัสทั้งหมดถูกต้อง เผยกดปุ่ม "Change password" ก็จะสามารถเปลี่ยนรหัสผ่านบัญชีใหม่ได้

ถ้ากดปุ่ม Settings ในรูปที่ 3.8 จะนำไปสู่หน้าแก้ไขข้อมูลส่วนตัวของเจ้าของบัญชี ได้แก่ ชื่อบัญชี และรูปภาพavatar ดังภาพที่ 3.10



รูปที่ 3.10: ส่วนประisanต่อผู้ใช้ หน้าตั้งค่าของผู้ใช้ หากกดปุ่ม "Settings" ในรูปที่ 3.8

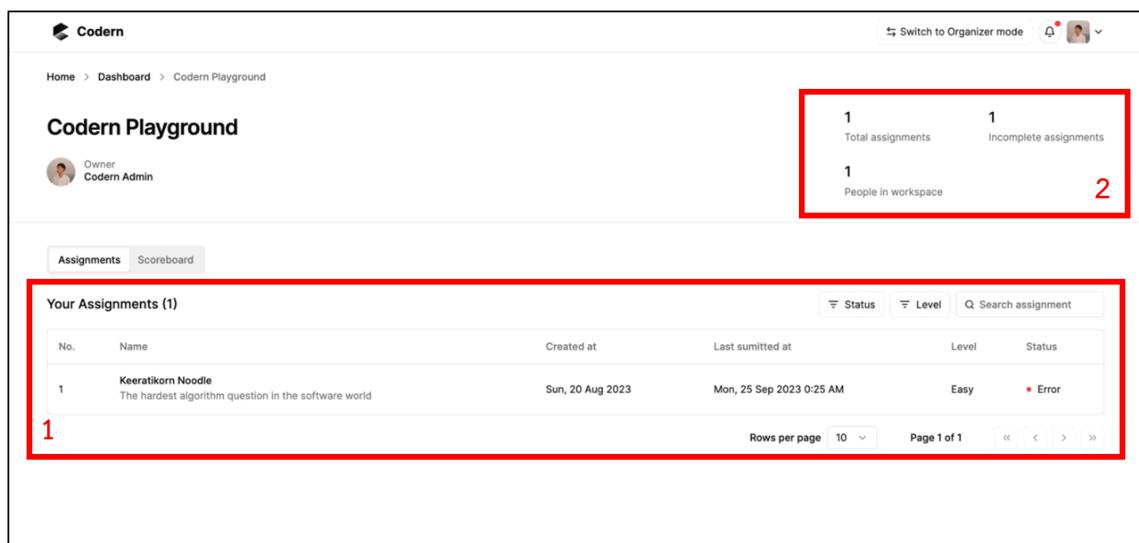
และถ้ากดปุ่ม Sign out ในรูปที่ 3.8 จะกลับไปสู่หน้าล็อกอินและสุดของระบบ ดังภาพที่ 3.3

หากเป็นผู้ดูแล ในการแสดงรายการห้องทั้งหมด จะเปลี่ยนเป็นมุมมองของผู้ดูแลระบบดังรูปที่ 3.11 แสดงควบคุณดังกล่าว มีบ탕ภาพและ การจัดวางที่คล้ายคลึงกันกับฟังของผู้ใช้ทั่วไปในรูปที่ 3.4 ต่างกันที่การจัดหมวดหมู่ห้องเรียน ทั้งนี้ก็จะมีปุ่มกดที่มุมขวาบนเป็นปุ่มที่สำคัญ ที่สุดคือ “Create Workspace” ที่เป็นปุ่มในการสร้างห้องเรียนหรือกลุ่มเรียนใหม่ (กรอบสีแดง A ในรูป 3.11) ซึ่งหากกดต่อไป ก็จะไปสู่หน้าสร้างห้องเรียนหรือกลุ่มเรียนใหม่เหมือนกับในรูปที่ 3.6



รูปที่ 3.11: ส่วนประisanต่อผู้ใช้ หน้าແນ່ງគົບຄຸມໃນມູນມອງຂອງຜູ້ດູແລະຮບບ

ถ้าหากกดเข้าห้องเรียนหรือกลุ่มเรียนใด ๆ ในหน้าก่อน (ในรูปที่ 3.4) ก็จะนำมาสู่หน้าจอ ดังรูปที่ 3.12 (หรือรูปที่ 3.13 ถ้าหากเป็นแอคມินหรือผู้ดูแล) โดยในหน้านี้ ที่ใช้นี้ ที่ใช้นี่ 1 จะแสดงโจทย์ปัญหา การบ้านหรือข้อสอบทั้งหมดที่เจ้าของห้อง/อาจารย์ผู้สอนได้สร้างไว้ แต่ละรายการก็จะแสดงชื่อมูลตั้งแต่ชื่องาน, รายละเอียดงาน, ระดับความยาก, จนไปถึงสถานะ นอกจากแสดงโจทย์แล้ว ในส่วนบน ตรงโซนที่ 2 ยังมีการแสดงข้อมูลของห้องเรียนหรือกลุ่มเรียนด้วย ตั้งแต่ชื่อเจ้าของห้อง จำนวนงานในกลุ่มหรือห้อง จำนวนงานที่ผู้ใช้ยังไม่เสร็จ จำนวนสมาชิกที่ได้เข้าร่วมห้องหรือกลุ่มนี้



รูปที่ 3.12: ส่วนประisanต่อผู้ใช้ หน้ารายการโจทย์ປັບປຸງ

The screenshot shows the Codern Playground dashboard. At the top, there are statistics: 4 total assignments, 4 incomplete assignments, and 4 people in workspace. Below this is a navigation bar with tabs: Assignments (highlighted with a red box), Participant, Submission, and Settings. A blue box labeled 'A' highlights the '+ Create' button. A red box labeled 'B' highlights the three-dot menu icon on the right side of the assignment table. A red box labeled 'C' highlights the tab bar. The main area displays a table of assignments:

| No. | Name | Level | Due date | Submission | Status |
|-----|--|-------|--------------------------|------------|--------|
| 1 | Keeratikorn Noodle The hardest algorithm question in the software world | Easy | Fri, 24 Nov 2023 7:05 AM | In time | Failed |
| 2 | Story of Keeratikorn The hardest algorithm question in the software world | Hard | Fri, 24 Nov 2023 7:19 AM | Late | Failed |
| 3 | Q test just test the super long | Hard | No due date | - | Todo |
| 4 | Keeratikorn Saga I โดยที่ประชาชั้นเนื่องสถาปัตยกรรมไทยที่มี... | Easy | No due date | - | Todo |

At the bottom, there are pagination controls: 'Rows per page' (10), 'Page 1 of 1', and navigation arrows.

รูปที่ 3.13: ส่วนประสานต่อผู้ใช้ หน้ารายการโจทย์ปัญหา ในมุมมองของผู้ดูแลระบบ

ในอีกฝั่ง ในส่วนของผู้ดูแลในรูปที่ 3.13 ก็จะแสดงผลคล้ายคลึงกับฝั่งของผู้ใช้ในรูปที่ 3.12 มีความต่างที่มีปุ่ม “Create Assignment” (กรอบสีน้ำเงิน A) ซึ่งเมื่อกดแล้วจะนำไปสู่รูปที่ 3.14 ซึ่งเป็นฟอร์มสำหรับสร้างการบ้าน โจทย์ปัญหาหรือข้อสอบ สำหรับให้อาจารย์ผู้สอน หรือเจ้าของห้องได้ใช้, บุ่มรูป "Kebab Menu" ท้ายสุดของแต่ละແควาเพื่อเข้ามา (กรอบสีแดง B) ซึ่งเมื่อกดแล้วจะนำพาไปหน้าแก้ไขโจทย์ปัญหาข้อหนึ่งๆ ในหน้านั้น (รูปที่ 3.15) ผู้ดูแลระบบหรือเจ้าของห้องสามารถแก้ไขข้อมูลของ Assignment นั้น ๆ ได้

สามารถใช้แบบปุ่ม Tab (ในกรอบ C) สามารถที่จะย้ายหน้าไปยังแบบอื่นได้ แบบตั้งกล่าวประกอบด้วยหน้า Participants ดังรูปที่ 3.16, Submission ดังรูปที่ 3.17, Settings ดังรูปที่ 3.20 3.21 3.22

The screenshot shows the 'Create assignment' interface on the Codern Playground. At the top, there's a navigation bar with 'Dashboard > Codern Playground > Create assignment'. Below it, the title 'Codern Playground' is displayed, along with a creation date 'Created on Tuesday, 14 November 2023'. A user profile for 'Owner Opal @Codern' is shown on the right.

The main form is divided into three sections:

- 01 General**: A brief information of an assignment. It includes fields for 'Name' (Assignment name), 'Description' (Description of an assignment), and 'Level' (Select an assignment level).
- 02 Instruction**: An instruction, how to complete an assignment. It includes 'Time limit (ms)' (Time limit in millisecond) and 'Memory limit (kB)' (Memory limit in kilobyte). There's also a 'Detail' text area and a 'Support markdown' button.
- 03 Test case**: All possible testcases to test assignment submissions.

At the bottom right of the form is a large black 'Create' button.

รูปที่ 3.14: ส่วนประسانต่อผู้ใช้ หน้าพอร์มสร้างโจทย์ปัญหาของผู้ดูและบบ

The screenshot shows the Codern Playground 'Edit assignment' interface. At the top, there's a navigation bar with 'Dashboard > Codern Playground > Edit assignment'. The main title is 'Codern Playground' with a creation date of 'Created on Tuesday, 14 November 2023'. On the right, there's an 'Owner' section with a profile picture of 'Opal @Codern'. The interface is divided into three main sections:

- 01 General**: Fields for Name (Keeratikorn Noodle), Description (The hardest algorithm question in the software world), and Level (Hard).
- 02 Instruction**: Fields for Time limit (ms) (100) and Memory limit (kB) (500). There's also a 'Detail' text area and a 'Support markdown' button.
- 03 Test case**: A section for test cases with a 'Create' button.

A 'Preview' button is located at the bottom of the instruction section.

รูปที่ 3.15: ส่วนประسانต่อผู้ใช้ หน้าพร้อมแก้ไขโจทย์ปัญหาของผู้ดูแลระบบ

ชิ้นภายในรูปที่ 3.14 นั้น อาจารย์หรือผู้ดูแลสามารถเพิ่มโจทย์ปัญหาใหม่เข้าสู่ระบบได้ โดยจะมีข้อมูลของโจทย์ปัญหาให้กรอกตามกำหนดประกอบไปด้วย ชื่องาน, รายละเอียดงาน, ระดับความยาก, จำนวนหน่วยความจำที่ใช้ได้ (Memory Limit), ระยะเวลาที่ใช้ในการทำคำตอบ (Runtime Limit), รายละเอียดของโจทย์, ตัวอย่างผลลัพธ์ (Test case) และ nokจากนี้ยังสามารถกด Preview เพื่อดูตัวอย่างของโจทย์ที่จะออกแบบได้อีกด้วย เมื่อกดปุ่ม "Create" ด้านล่างสุด โจทย์ปัญหาก็จะถูกสร้างขึ้น

ส่วนภายในหน้า Edit Assignment ดังภาพที่ 3.15 นั้น จะเป็นหน้าที่อาจารย์หรือผู้ดูแลสามารถแก้ไขข้อมูลของโจทย์ปัญหานั้นๆ ที่ถูกสร้างไว้แล้วได้ โดยจะสามารถแก้ไขข้อมูลได้ทั้งหมด ดังแต่ ชื่องาน, รายละเอียดงาน, ระดับความยาก, จำนวนหน่วยความจำที่ใช้ได้ (Memory Limit), ระยะเวลาที่ใช้ในการทำคำตอบ (Runtime Limit), รายละเอียดของโจทย์ และตัวอย่างผลลัพธ์ (Test case)

The screenshot shows the Codern Playground dashboard. At the top right, there are user icons and a dropdown menu. Below the header, a summary box displays statistics: 4 Total assignments, 4 Incompleted assignments, and 4 People in workspace. A cartoon character icon is also present. The main area features a navigation bar with tabs: Assignments (selected), Participant (highlighted with a blue box), Submission, and Settings. Below the tabs, a section titled "Participants (4)" lists four users with their names, join dates, and a "Delete" column (highlighted with a red box). At the bottom right of the table, there is a "+ Add" button (highlighted with a blue box). The footer includes pagination controls: "Rows per page" (set to 10), "Page 1 of 1", and navigation arrows.

| No. | Name | Join date | Delete |
|-----|-------------------------|--------------------------|--------|
| 1 | Pimmaida Laisusun | Fri, 24 Nov 2023 7:05 AM | |
| 2 | Wichayaporn Pinjinan | Fri, 24 Nov 2023 7:05 AM | |
| 3 | Thanimika Tanajaroenkij | Fri, 24 Nov 2023 7:05 AM | |
| 4 | Wannapa Pongpaisan | Fri, 24 Nov 2023 7:05 AM | |

รูปที่ 3.16: ส่วนประisanต่อผู้ใช้ หน้ารายชื่อผู้ใช้ใน Workspace ทั้งหมด ในมุมมองของผู้ดูแลระบบ

หากกดตรง Tab ตรง "Participant" (ตรงกรอบ C รูปที่ 3.13) หน้าในรูปที่ 3.16 จะแสดงขึ้นมา โดยหน้าดังกล่าวเป็นหน้าแสดงรายละเอียดสมาชิกของห้องเรียนนั้น ๆ โดยจะแสดงจำนวนสมาชิก รายชื่อชื่อ นามสกุล วันที่สมาชิกคนนั้น ๆ เข้าร่วมห้องเรียน และผู้ดูแลระบบ จะสามารถลบสมาชิกคนนั้น ๆ ออกจากห้องเรียนได้ด้วยปุ่ม "Delete" ที่ท้ายรายชื่อคนนั้น ๆ (กรอบสีเขียว A) รวมถึงสามารถเพิ่มสมาชิกใหม่เข้ามาได้ด้วยปุ่ม "Add" (กรอบสีแดง B) แต่ถ้าหากเป็นผู้ใช้กรรมดาทัวไป จะมองไม่เห็นปุ่ม "Add" หรือปุ่ม "Delete"

The screenshot shows the Codern Playground dashboard. At the top right, there are user icons and a search bar. Below the header, it displays 'Total assignments' (4), 'Incomplete assignments' (4), and 'People in workspace' (4). A small emoji icon is also present. The main area features a table titled 'Submission (4)' with columns: No., Name, Publish date, Due date, Submitted, Not submitted, and Score. The table lists four submissions:

| No. | Name | Publish date | Due date | Submitted | Not submitted | Score |
|-----|--|--------------------------|--------------------------|-----------|---------------|-------|
| 1 | Kearatkorn Noodle The hardest algorithm question in the software world | Fri, 24 Nov 2023 7:05 AM | Fri, 24 Nov 2023 7:05 AM | 34 | 0 | 99 |
| 2 | Story of Kearatkorn The hardest algorithm question in the software world | Fri, 24 Nov 2023 7:05 AM | Fri, 24 Nov 2023 7:05 AM | 32 | 2 | 95 |
| 3 | Q test just test the super long | Fri, 24 Nov 2023 7:05 AM | Fri, 24 Nov 2023 7:05 AM | 34 | 0 | 98 |
| 4 | Kearatkorn Saga I ใบอนุญาตใช้งานซอฟต์แวร์ที่มีลิขสิทธิ์... | Fri, 24 Nov 2023 7:05 AM | Fri, 24 Nov 2023 7:05 AM | 31 | 3 | 91 |

At the bottom, there are buttons for 'Rows per page' (10), 'Page 1 of 1', and navigation arrows.

รูปที่ 3.17: ส่วนประสานต่อผู้ใช้ หน้ารายการโปรแกรมที่ผู้ใช้ส่งเข้ามาใน Workspace ในมุมมองของผู้ดูแลระบบ

ต่อมา เมื่อกดปุ่ม Tab ที่ "Submission" แล้ว จะปรากฏหน้าดังรูปที่ 3.17 โดยหน้าดังกล่าวเป็นหน้าแสดงรายละเอียดรายการโปรแกรมทั้งหมดที่มีของห้องเรียนนั้น ๆ โดยจะแสดงรายชื่อและรายละเอียดของโปรแกรมนั้น ๆ วันที่เผยแพร่โปรแกรมนั้น ๆ วันสุดท้ายที่กำหนดส่งโปรแกรมนั้น ๆ จำนวนคนที่ส่งคำตอบเข้ามาแล้วทั้งหมด จำนวนคนที่ยังไม่ส่งคำตอบเข้ามา และผลคะแนนโดยเฉลี่ยของโปรแกรมชื่อนั้น ๆ ผู้ดูแลระบบจะสามารถกดเข้าไปในชื่อโปรแกรมนั้น ๆ เพื่อดูรายละเอียดของห้องเรียนที่ได้ทำการส่งคำตอบเข้ามาแล้วได้ ซึ่งจะปรากฏดังรูปที่ 3.18

The screenshot shows the Codern Playground dashboard. At the top, there are navigation links: Dashboard > Codern Playground. On the right, there are statistics: 4 Total assignments, 4 Incompleted assignments, and 4 People in workspace. Below this is a smiley face icon.

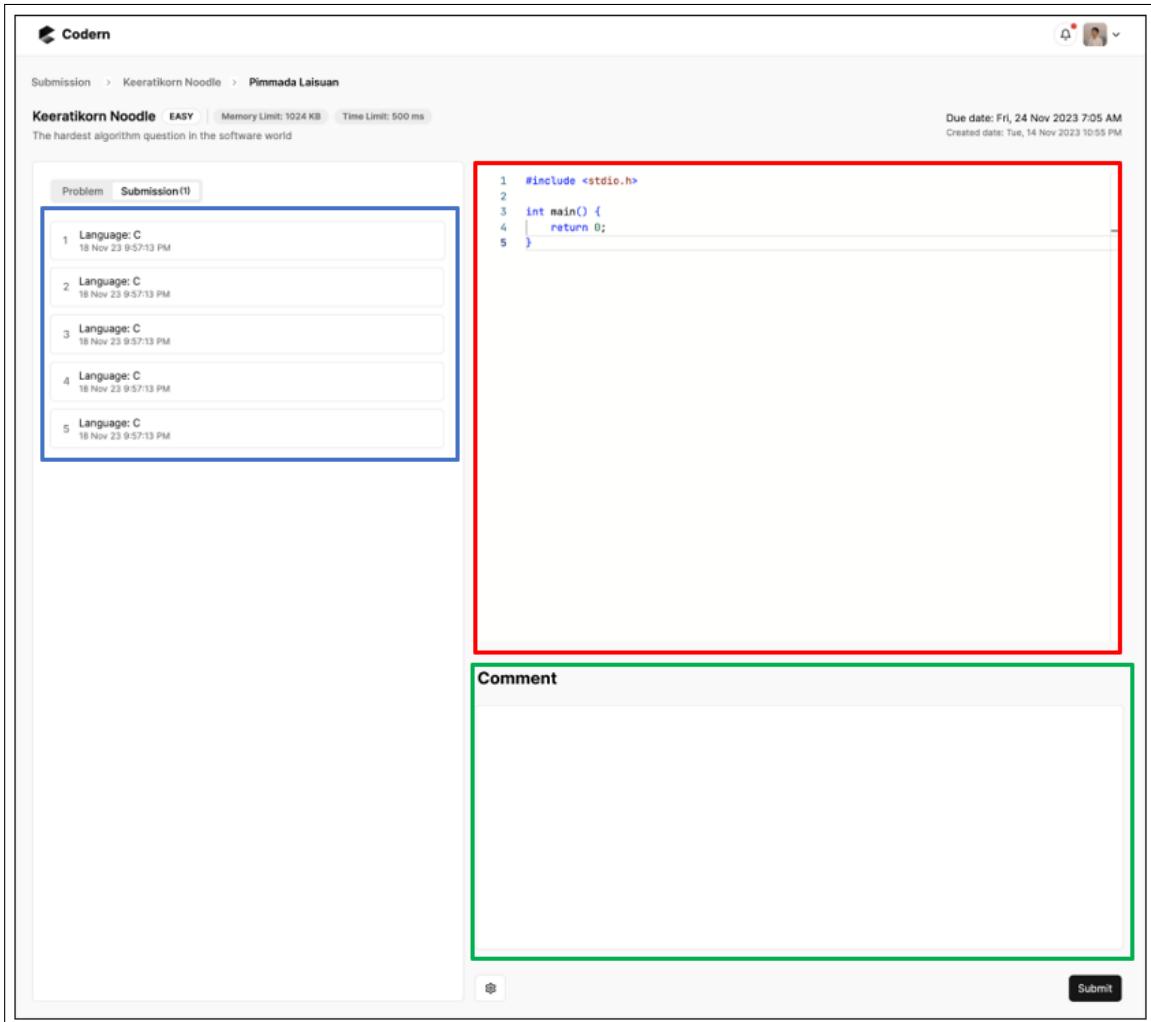
Below the stats is a table titled "Keeratikorn Noodle (34)" showing assignment submissions:

| No. | Name | Submit date | Test case | Score |
|-----|-------------------------|--------------------------|-----------|-------|
| 1 | Pimmaida Laisuan | Fri, 24 Nov 2023 7:05 AM | 5 | 99 |
| 2 | Wichayaporn Pinijnan | Fri, 24 Nov 2023 7:05 AM | 4 | 95 |
| 3 | Thananika Tanajaroenkij | Fri, 24 Nov 2023 7:05 AM | 5 | 98 |
| 4 | Wannapa Pongpaisan | Fri, 24 Nov 2023 7:05 AM | 2 | 91 |

At the bottom of the table are buttons for "Rows per page" (10), "Page 1 of 1", and navigation arrows.

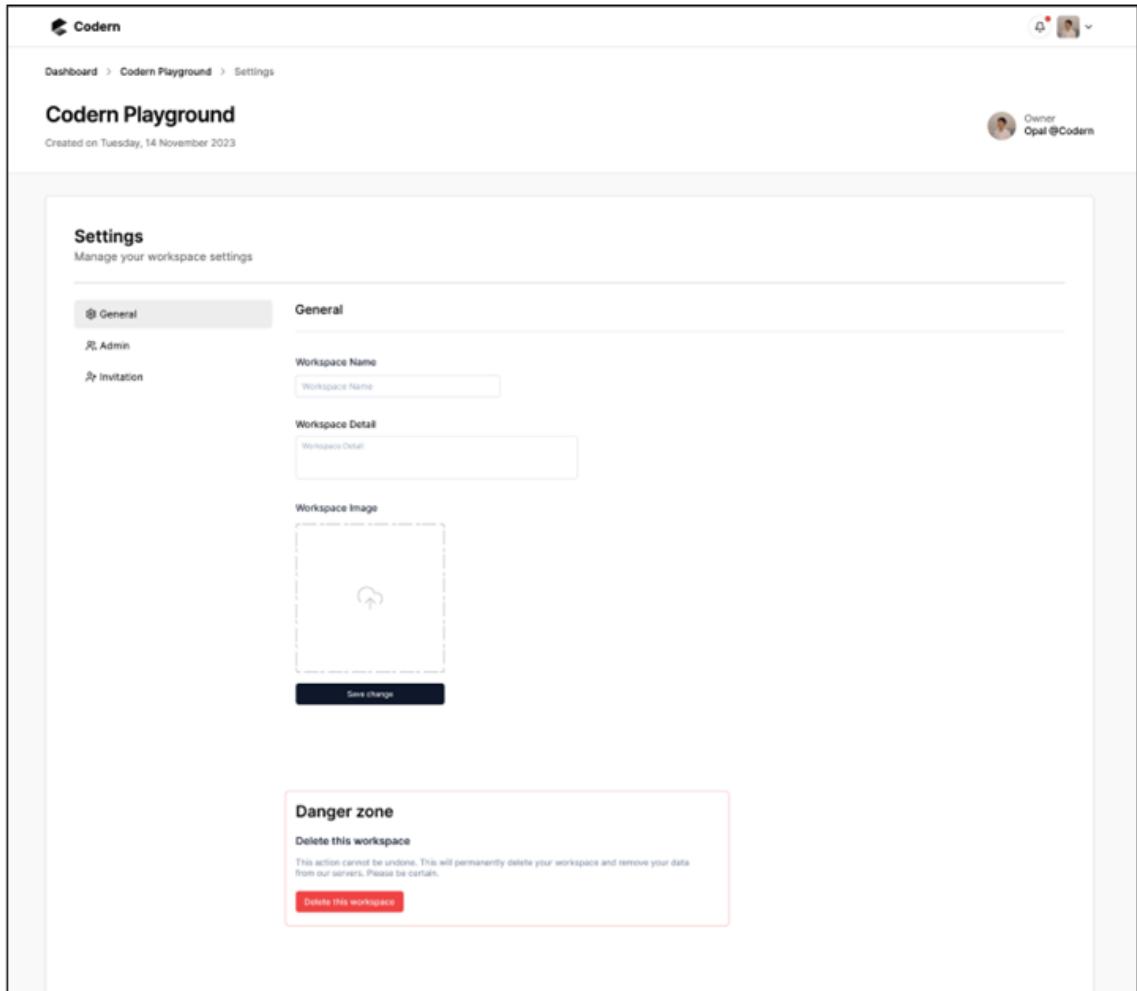
รูปที่ 3.18: ส่วนประisanต่อผู้ใช้ หน้ารายชื่อสมาชิกห้องเรียนที่ได้ส่งคำตอบของโปรแกรมนั้น ๆ เข้ามาใน Workspace ของผู้ดูแลระบบภายใน

ภายในหน้าดังกล่าว จะแสดงรายชื่อสมาชิกทั้งหมดที่ได้ทำการส่งคำตอบของโปรแกรมนั้น ๆ เข้ามาใน Workspace และดังรูปที่ 3.18 และจะแสดงรายละเอียดวันเวลาที่ส่งคำตอบเข้ามา จำนวนทดสอบที่ถูกต้อง และผลคะแนนที่ได้ของโปรแกรมนั้น ๆ ผู้ดูแลระบบจะสามารถกดเข้าไปในชื่อผู้ใช้คนนั้น ๆ เพื่อดูคำตอบของโปรแกรมที่ผู้ใช้คนนั้น ๆ เข้ามาได้ ทั้งผู้ดูแลยังสามารถแสดงความคิดเห็นของตนส่งให้ผู้ใช้คนนั้น ๆ ได้อีกด้วย ซึ่งรายละเอียดหน้าดังกล่าวจะปรากฏตั้งแต่รูปที่ 3.19



รูปที่ 3.19: ส่วนประisanต่อผู้ใช้ หน้าแสดงคำตอบของโปรแกรมที่ผู้ใช้ได้ส่งเข้ามาใน Workspace ของผู้ดูแลระบบ

หลังจากเข้ามาแล้ว ภายในหน้านี้จะแสดงคำตอบของโปรแกรมที่ผู้ใช้ค้นนั้น ๆ ได้ส่งเข้ามาใน Workspace ดังรูปที่ 3.19 โดยรายละเอียดจะแบ่งได้เป็นสามส่วนหลักคือ รายการคำตอบโปรแกรมของแต่ละทดสอบที่ผู้ใช้ได้ส่งเข้ามา (ตรงกรอบสีน้ำเงินทางซ้ายมือรูปที่ 3.19) เมื่อกดที่รายการคำตอบแล้ว จะแสดงคำตอบโดยโปรแกรมของรายการนั้น ๆ ในกล่อง (ตรงกรอบสีแดงทางขวาบนรูปที่ 3.19) และทางด้านขวาล่างจะมีพื้นที่สำหรับให้ผู้ดูแลแสดงความคิดเห็นของตน (ตรงกรอบสีเขียวทางขวาล่างรูปที่ 3.19) ซึ่งเมื่อกดปุ่ม "Submit" ทางขวาล่างสุดแล้ว ความคิดเห็นก็จะถูกส่งไปยังผู้ใช้คันนั้น ๆ



รูปที่ 3.20: ส่วนประisanต่อผู้ใช้ หน้าตั้งค่า Workspace ใน Section "General" ในมุมมองของผู้ดูแลระบบ

หลังจากกดตรง Tab ตรง "Settings" (ตรงกรอบ C รูปที่ 3.13 ซึ่งมีแต่ผู้ดูแลระบบเท่านั้นที่จะเห็น) หน้าในรูปที่ 3.20 จะปรากฏขึ้นมา ซึ่งหน้าข้างต้นเป็นหน้าสำหรับตั้งค่าข้อมูลต่าง ๆ ภายในห้องเรียนหรือกลุ่มเรียนนั้น ๆ ซึ่งมีเฉพาะผู้ดูแลเท่านั้นที่สามารถเข้าไปดูและแก้ไขได้เท่านั้น โดยจะแบ่งเป็น 3 ส่วนย่อย ประกอบไปด้วย General settings ซึ่งจะเป็นเมนูแรกสุดดังภาพที่ 3.20 ตามมาด้วย Admin settings และ Invitation settings ดังภาพที่ 3.21 และภาพที่ 3.22 ตามลำดับ

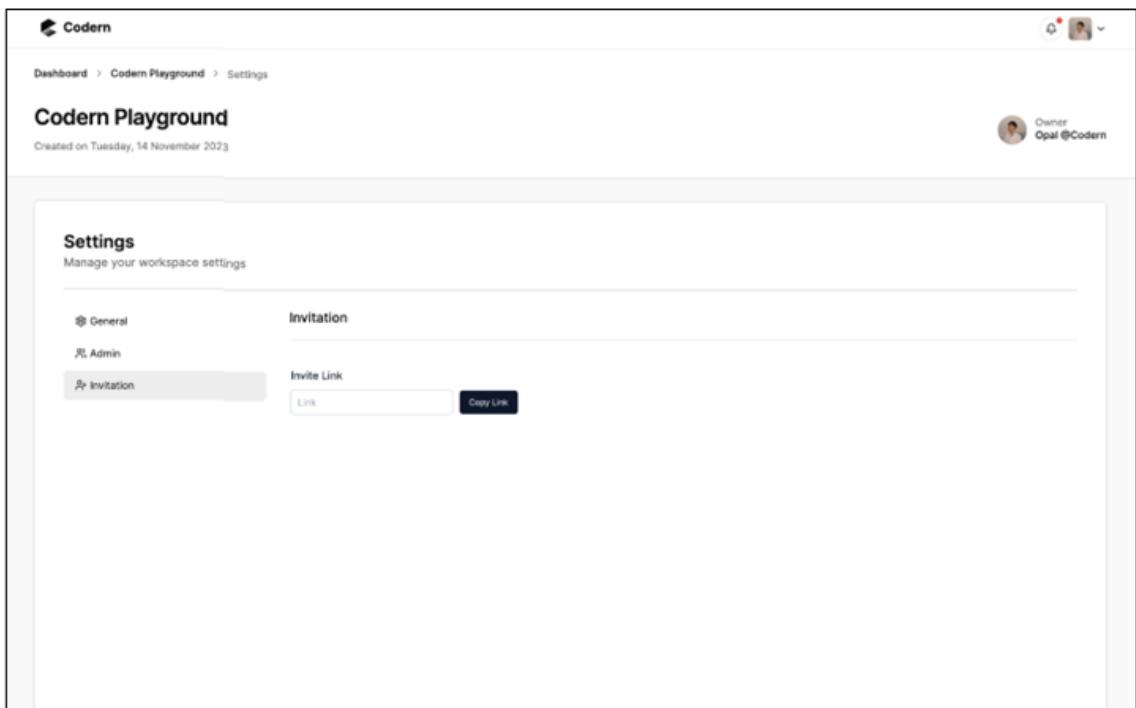
โดย General settings ดังภาพที่ 3.20 จะเป็นหน้าสำหรับการแก้ไขข้อมูลทั่วไปของห้องเรียนนั้น ๆ เช่น ชื่อของห้องเรียน รายละเอียดของห้องเรียน หรือ รูปภาพที่แสดง และที่สำคัญยังสามารถลบห้องเรียนนั้น ๆ ได้เมื่อกดที่ปุ่ม Delete this workspace ภายใน Danger zone ซึ่งการลบห้องเรียนนั้นจะเป็นการลบอย่างถาวร ข้อมูลทั้งหมดจะถูกลบโดยไม่สามารถคืนได้อีก

The screenshot shows the Codern Playground settings interface. At the top, there's a navigation bar with 'Dashboard' > 'Codern Playground' > 'Settings'. Below it, the workspace name 'Codern Playground' is displayed, along with a creation date 'Created on Tuesday, 14 November 2023'. On the right, there's a user profile for 'Owner Opal @Codern'.

The main area is titled 'Settings' with the subtitle 'Manage your workspace settings'. It features a tab navigation with 'General' (selected), 'Admin' (highlighted in blue), and 'Invitation'. The 'Admin' section contains a sub-section titled 'Invite admin' with a search bar labeled 'Search by username'. Three user profiles are listed under this search bar, each with a placeholder 'Username Admin'. A green 'Add Admin' button is located at the bottom of this section.

รูปที่ 3.21: ส่วนประสาณต่อผู้ใช้ หน้าตั้งค่า Workspace ใน Section "Admin" ในมุมมองของผู้ดูแลระบบ

ในของ "Admin" Settings ดังภาพที่ 3.21 จะเป็นหน้าสำหรับการเพิ่มแอดมินหรือผู้ดูแลเข้าสู่ห้องเรียนนั้น ๆ โดยจะประกอบด้วยกล่องค้นหา หากค้นหาพบคนที่ต้องการแล้ว เมื่อกดปุ่ม "Add admin" จะเป็นการเพิ่มผู้ดูแลคนนั้น ๆ เข้าสู่ห้องเรียน



รูปที่ 3.22: ส่วนประسانต่อผู้ใช้ หน้าตั้งค่า Workspace ใน Section "Invitation" ในมุมมองของผู้ดูแลระบบ

สุดท้ายในส่วนของ "Invitation" Settings ดังภาพที่ 3.22 เป็นหน้าสำหรับการเชิญผู้ใช้ที่ว่าไปเข้าสู่ห้องเรียนนั้น ๆ โดยตัวผู้ดูแล ซึ่งทำได้โดยจะมีการสร้างรหัสเชิญขึ้น หากกดปุ่ม "Copy link" จะเป็นการคัดลอกรหัสเชิญไปยังคลิปบอร์ด และเมื่อผู้ใช้กรอกรหัสตั้งกล่าวลงในกล่อง "Join with code" ในภาพที่ 3.7 และกดปุ่ม "Join" จะเป็นการเข้าร่วมห้องเรียนนั้น ๆ

ถ้าหากกดรายการการบ้าน โจทย์ปัญหา มาสักรายการ (ในรูปที่ 3.12) ก็จะนำพามาสู่หน้าโจทย์ปัญหาในรูปที่ 3.23 โดยในหน้านี้ในโซนที่ 1 จะเป็นช่องແингเขียนโปรแกรม สามารถที่จะเขียนโปรแกรมภาษาคอมพิวเตอร์ในโซนนี้ได้ ในโซนที่สอง จะเป็นแสดงรายละเอียดของโจทย์ตั้งแต่เนื้อหาโจทย์ ไปจนถึงจำนวนหน่วยความจำที่ใช้ได้ (Memory Limit) และระยะเวลาที่ใช้ในการทำคำตอบ (Runtime Limit) ถ้าหากต้องการส่งที่สามารถที่จะกดปุ่ม "Submit" ที่กรอบน้ำเงิน A เพื่อส่งงานเขียนโปรแกรมไปให้ตรวจสอบ และอีกทั้งก่อนส่ง ก็ยังสามารถที่จะตั้งค่าและเลือก compiler ที่ต้องการจะใช้ในการทำงานเขียนโปรแกรมดังกล่าวได้ที่ Dropdown ที่กรอบน้ำเงิน B หลังจากผู้ใช้กดส่งไปตรวจแล้ว สามารถติดตามผลได้ในรายการ submission ซึ่งสามารถกดดูได้ตรงกรอบสีเขียว C ในรูปที่ 3.23

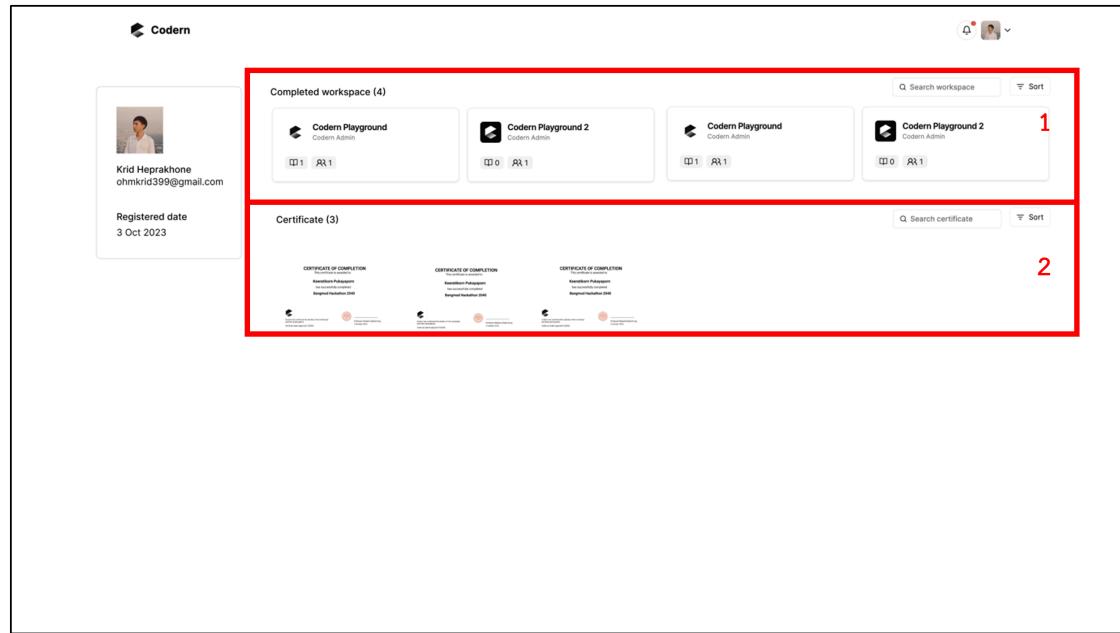
หลังจากกดเลือก Tab ชื่อ "Submission" ແงแสดงผลเนื้อหาโจทย์ จะถูกเปลี่ยนมาเป็นรายการผลส่งการตรวจทั้งหมด ในโซนกรอบแดงที่ 1 ในรูป 3.24 โดยในรายการก็จะแสดงรายละเอียดการส่งแต่ละครั้งที่ผู้ใช้ส่ง มีรายละเอียดต่าง ๆ อาทิเช่น ภาษาที่ส่งไปตรวจ และผลการตรวจของแต่ละ Test Case (ถ้า PASS แปลว่า output ของโปรแกรมที่เขียนให้ผลตรงตาม Test Case ที่โจทย์ข้อนี้ตั้งไว้, แต่ถ้า FAIL ก็จะแสดงว่า output ของโปรแกรมที่เขียนไม่ได้ให้ผลตรงตาม Test Case ที่โจทย์ข้อนี้ตั้งไว้)

The screenshot shows a programming challenge titled "Keeratikorn Noodle" (EASY) with a memory limit of 1024 KB and a time limit of 500 ms. The challenge description is in Thai and asks for a program to find the maximum value in a 16x16 grid. The submission area has two tabs: "A" (selected) and "B". A red box labeled "1" highlights the submission area, and another red box labeled "2" highlights the problem description.

รูปที่ 3.23: ส่วนประสานต่อผู้ใช้ หน้าโจทย์ปัญหาและແຜງເຂີຍໂປຣແກຣມ

The screenshot shows a submitted solution for challenge "kridheprakone-home" with a memory limit of 1 KB and a time limit of 1 ms. The submission status is "1 / 1" and it was submitted on 04/10/23 at 07:10:27 AM. The code is a simple C program that prints "0" and returns 0. A red box labeled "1" highlights the submission status, and another red box labeled "2" highlights the code editor area.

รูปที่ 3.24: ส่วนประสานงานຕ่อผู้ใช้ หน้าโจทย์ปัญหาและແຜງເຂີຍໂປຣແກຣມ ພ້ອມຮາຍການສ່ວນແລະຜົດການຕ້ອງກວດທີ່ຈະມີ



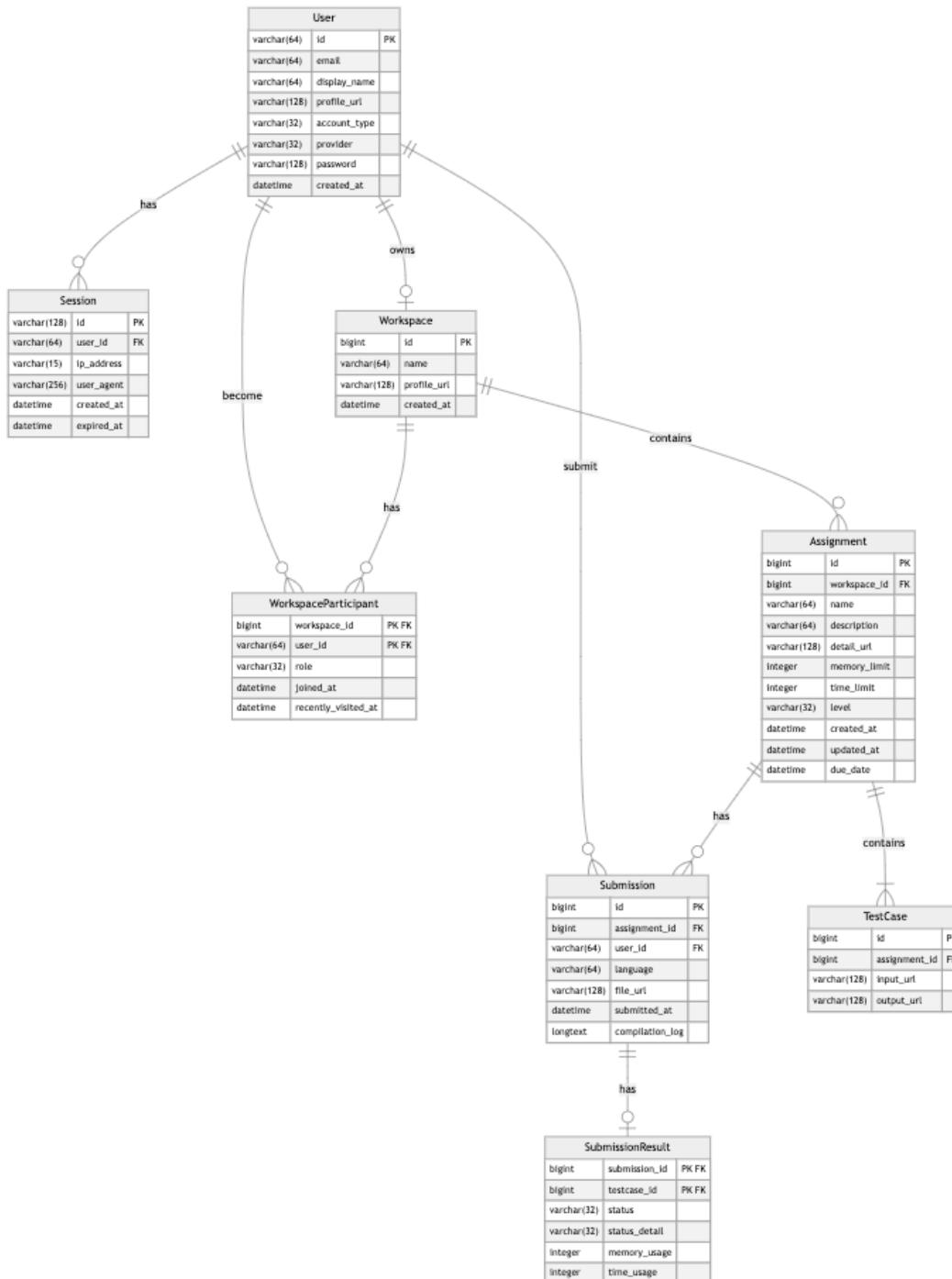
รูปที่ 3.25: ส่วนประisanงานต่อผู้ใช้ หน้าแสดงข้อมูล ประวัติการใช้งาน และเกียรติบัตรทั้งหมด ของผู้ใช้เก่า

ในอีกโหมดแยกจากระบบทหลัก (ดูแผนผังการนำทางระหว่างหน้าของเว็บไซต์ ในรูปที่ 3.38) หน้าแสดงข้อมูลและประวัติการใช้งานของผู้ใช้เก่า (ในรูปที่ 3.25) เช่น นักเรียนหรือนักศึกษาที่ได้จบการศึกษาไปแล้ว เป็นต้น

ผู้ใช้ทั่วไป (ไม่จำเป็นต้องมีบัญชีบนซอฟต์แวร์ ณ เวลานั้น) สามารถที่จะเข้ามาค้นดูประวัติ ข้อมูลการใช้งาน ภาระงานที่ได้ทำ ห้องหรือกลุ่มเรียนที่เคยได้เข้าร่วม (โซนกรอบแดงที่ 1) พร้อมทั้งค้นເօາเกียรติบัตร (โซนกรอบแดงที่ 2) ไปพิมพ์ใช้งานต่อได้ (เกียรติบัตรจะถูกสร้างขึ้นแก่ผู้ใช้ทุกคนที่ต่อเมื่อเจ้าของห้อง ปิดห้องหรือกลุ่มเรียน) สำหรับชื่อนี้หรือให้การศึกษาต่อ ใช้สมัครงานต่อไปก็ได้

3.4 โครงสร้างฐานข้อมูล

จากรูปที่ 3.26 ด้านล่าง เป็นแผนภาพแสดงฐานข้อมูลของซอฟต์แวร์ที่ทางคณะผู้จัดทำได้ออกแบบไว้เบื้องต้น ฐานข้อมูลดังกล่าวเป็น Relation Database Management System ที่จะเก็บข้อมูลบนซอฟต์แวร์ตั้งแต่ข้อมูลผู้ใช้ จนไปถึงข้อมูลทดสอบห้อง/ห้องเรียน ข้อมูลโจทย์ ทั้งเนื้อหาของโจทย์พร้อมทั้ง Test Case ของโจทย์แต่ละข้อ



รูปที่ 3.26: แผนภาพฐานข้อมูล วาดด้วย Mermaid JS

ในระบบฐานข้อมูลนี้ ประกอบด้วยตารางดังต่อไปนี้

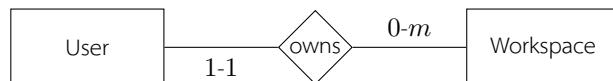
1. **User** เป็นตารางที่เก็บข้อมูลของผู้ใช้แต่ละคน เช่น อีเมล์, ชื่อ-นามสกุลของผู้ใช้ เป็นต้น จากความสัมพันธ์ของตารางดังกล่าว กับตารางอื่น ๆ สามารถสรุปความสัมพันธ์ได้ดังนี้

- จากรูปที่ 3.27 ผู้ใช้สามารถที่จะเข้าสู่ระบบได้หลายเครื่องพร้อม ๆ กัน คือผู้ใช้เข้าสู่ระบบได้ (has) หลาย Session พร้อมกัน



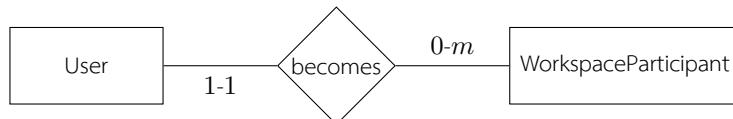
รูปที่ 3.27: แผนผังแสดงความสัมพันธ์ระหว่างตาราง User และ Session

- จากรูปที่ 3.28 ผู้ใช้สามารถที่จะครอบครอง (owns) Workspace ได้มากกว่า 1 Workspace



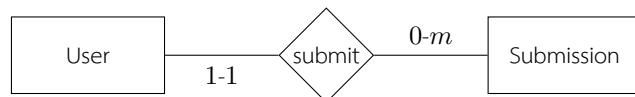
รูปที่ 3.28: แผนผังแสดงความสัมพันธ์ระหว่างตาราง User และ Workspace

- ความสัมพันธ์ในรูป 3.29 ผู้ใช้สามารถที่จะเป็นคนเข้าร่วม (join workspace) ในกลุ่มเรียนหรือห้องเรียน (หรือ Workspace Participant) ได้มากกว่า 1 ห้องเรียนหรือกลุ่มเรียน หรือ Workspace



รูปที่ 3.29: แผนผังแสดงความสัมพันธ์ระหว่างตาราง User และ WorkspaceParticipant

- จากรูปที่ 3.30 ผู้ใช้สามารถที่จะส่ง (submit) ได้มากกว่า 1 ห้องเรียนหรือกลุ่มเรียน หรือ Workspace



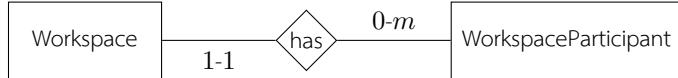
รูปที่ 3.30: แผนผังแสดงความสัมพันธ์ระหว่างตาราง User และ Submission

2. **Session** เป็นตารางที่จะเก็บข้อมูล Session หรือข้อมูลเครื่อง ข้อมูลช่องทางที่ผู้ใช้เข้าสู่ระบบ สามารถสรุปความสัมพันธ์ได้ดังนี้

- ถึงแม้ผู้ใช้หนึ่งท่านสามารถที่จะมีเด็หลาย Session คือผู้ใช้สามารถเข้าสู่ระบบได้หลายเครื่องพร้อมกัน แต่ Session เป็นของผู้ใช้แค่คนเดียวเท่านั้นตามรูปที่ 3.27

3. **Workspace** เป็นตารางที่สร้างขึ้นมาเก็บข้อมูลห้องเรียนหรือกลุ่มเรียน (Workspace) มีความสัมพันธ์กับตารางอื่นดังต่อไปนี้

- จากรูปที่ 3.27 ห้องเรียนหรือกลุ่มเรียนมีผู้ใช้ที่เป็นเจ้าของห้องได้เพียงแค่หนึ่งท่านเท่านั้น ตามในรูปเดิม รูปที่ 3.28
- ในหนึ่งกลุ่มเรียนหรือห้องเรียน สามารถมี (has) ผู้เข้าร่วมหรือ Workspace Participant ได้หลายคน ตามในรูปที่ 3.31



รูปที่ 3.31: แผนผังแสดงความสัมพันธ์ระหว่างตาราง Workspace และ WorkspaceParticipant

- จากรูปที่ 3.32 ในหนึ่งกลุ่มเรียนหรือห้องเรียน สามารถที่จะมี (contains) โจทย์ปัญหา การบ้านหรือข้อสอบ (Assignment) ได้หลายรายการ



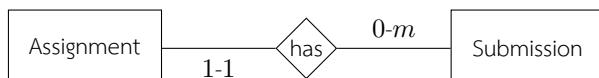
รูปที่ 3.32: แผนผังแสดงความสัมพันธ์ระหว่างตาราง Workspace และ Assignment

- WorkspaceParticipant เป็นตารางที่สร้างขึ้นมาเก็บข้อมูลของผู้เข้าร่วมในแต่ละห้องเรียนหรือกลุ่มเรียน (Workspace) มีความสัมพันธ์กับตารางอื่นดังต่อไปนี้

- อ้างอิงจากรูปที่ 3.29 สถานะของการเป็นผู้เข้าร่วมห้องไดห้องหนึ่งๆ คือ เป็นของผู้ใช้เพียงหนึ่งท่านเท่านั้น
- สถานะของการเป็นผู้เข้าร่วมห้องไดห้องหนึ่งๆ คือ เป็นสถานะของห้องเรียนหรือกลุ่มเรียนเดียวเท่านั้น ตามความสัมพันธ์ในรูปที่ 3.31

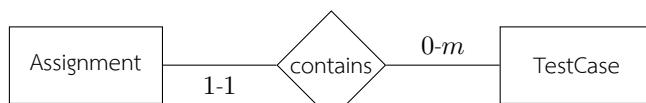
- Assignment เป็นตารางที่สร้างขึ้นมาเก็บข้อมูลและรายละเอียดของการบ้าน ข้อสอบและโจทย์ปัญหาทั้งหมด (Assignment) สามารถสรุปความสัมพันธ์กับตารางอื่นได้ดังต่อไปนี้

- ดูจากรูปที่ 3.32 การบ้าน ข้อสอบและโจทย์ปัญหาเป็นของ Workspace เดียวเท่านั้น
- โจทย์ปัญหา การบ้านหรือข้อสอบแต่ละข้อมี (has) ได้ข้อมูลส่งงาน (Submission) ได้หลายรายการ ตามรูปที่ 3.33



รูปที่ 3.33: แผนผังแสดงความสัมพันธ์ระหว่างตาราง Assignment และ Submission

- โจทย์ปัญหา การบ้านหรือข้อสอบแต่ละข้อมี (contains) ได้หลายกรณีทดสอบหรือ teskcase (Test Case) ตามในรูปที่ 3.34

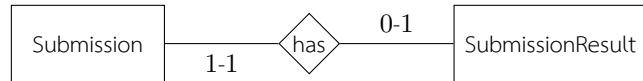


รูปที่ 3.34: แผนผังแสดงความสัมพันธ์ระหว่างตาราง Assignment และ TestCase

- Submission เป็นตารางที่เก็บบันทึกรายการส่งงาน รายละเอียดและข้อมูลไฟล์งานเขียนโปรแกรมที่ผู้ใช้งานมาให้ตรวจสอบ มีความสัมพันธ์ดังนี้

- ดูจากรูปที่ 3.30 ข้อมูลส่งงานหนึ่งรายการ เป็นของผู้ใช้หนึ่งท่านเท่านั้น
- ข้อมูลส่งงานหนึ่งรายการ จะเชื่อมกับโจทย์ปัญหา การบ้านหรือข้อสอบตัวเดียวเท่านั้น อ้างอิงจากความสัมพันธ์ในรูปที่ 3.33

- ข้อมูลส่งงานหนึ่งรายการ (Submission มีได้ (has) แค่เพียงหนึ่งผลลัพธ์การตรวจเท่านั้น (Submission Result) ตามความสัมพันธ์ในรูปที่ 3.35



รูปที่ 3.35: แผนผังแสดงความสัมพันธ์ระหว่างตาราง Assignment และ TestCase

- TestCase เป็นตารางที่เก็บกรณีทดสอบหรือตรวจสอบโปรแกรม หรือเทสเคส (Test Case) ประกอบด้วยความสัมพันธ์ต่อไปนี้
 - ดูจากรูปที่ 3.34 หนึ่งกรณีทดสอบ (Test Case) เชื่อมกับหนึ่งโจทย์ปัญหา ข้อสอบหรือการบ้าน (Assignment) หนึ่งชิ้นเท่านั้น
- SubmissionResult เป็นตารางที่เก็บบันทึกผลลัพธ์การตรวจทั้งหมด ประกอบด้วยความสัมพันธ์ดังต่อไปนี้
 - จากความสัมพันธ์ในรูปที่ 3.35 ผลลัพธ์การตรวจหนึ่งรายการ (Submission Result) เชื่อมกับแค่หนึ่งข้อมูลส่งงาน (Submission) เท่านั้น

3.4.1 พจนานุกรมข้อมูล

ในส่วนนี้ จะเป็นพจนานุกรมสำหรับอธิบายตัวแปรและตารางต่าง ๆ ที่อยู่ในแผนภาพฐานข้อมูลในรูปที่ 3.4 ในหัวข้อก่อน

ตารางที่ 3.3 พจนานุกรมข้อมูลของตาราง User

| Attribute Name | Description | Data Type | Constraints | References |
|-----------------|--------------------------|--------------|-----------------------|------------|
| id | รหัส ID การส่งงาน | bigint | Primary Key | - |
| assignment_id | ID ของการงานหรือโจทย์ | bigint | Foreign Key, Not Null | Assignment |
| user_id | ID ของผู้ใช้ | varchar(64) | Foreign Key, Not Null | User |
| language | ภาษาที่ใช้ในการเขียนโค้ด | varchar(64) | Not Null | - |
| file_url | ลิงก์ไฟล์งาน | varchar(128) | Not Null | - |
| submitted_at | วันเวลาส่งงาน | datetime | Current Timestamp | - |
| compilation_log | ผลการคอมไพล์ | longtext | - | - |

ตารางที่ 3.4 พจนานุกรมข้อมูลของตาราง Session

| Attribute Name | Data Description | Data Type | Constraints | References |
|----------------|----------------------------|--------------|-----------------------|------------|
| id | ID ของเซสชัน | varchar(128) | Primary Key | - |
| user_id | ID ผู้ใช้งาน | varchar(64) | Foreign Key, Not Null | User |
| ip_address | ที่อยู่ IP ของผู้ใช้งาน | varchar(15) | Not Null | - |
| user_agent | Browser ที่ผู้ใช้เข้ามายัง | varchar(256) | Not Null | - |
| created_at | วันเวลาเริ่มเซสชัน | datetime | Not Null | - |
| expired_at | วันเวลาจะปิดเซสชัน | datetime | Not Null | - |

ตารางที่ 3.5 พจนานุกรมข้อมูลของตาราง Workspace

| Attribute Name | Data Description | Data Type | Constraints | References |
|----------------|--------------------------------|--------------|-------------------|------------|
| id | ID ของห้องหรือกลุ่มเรียน | bigint | Primary Key | - |
| name | ชื่อของห้องหรือกลุ่มเรียน | varchar(64) | Not Null | - |
| profile_url | ลิงก์ของห้องหรือกลุ่มเรียน | varchar(128) | Not Null | - |
| created_at | วันเวลาสร้างห้องหรือกลุ่มเรียน | datetime | Current Timestamp | - |

ตารางที่ 3.6 พจนานุกรมข้อมูลของตาราง Workspace Participant

| Attribute Name | Data Description | Data Type | Constraints | References |
|---------------------|--------------------------|-------------|------------------------------------|------------|
| workspace_id | ID ของห้องหรือกลุ่มเรียน | bigint | Primary Key, Foreign Key | Workspace |
| user_id | ID ของผู้ใช้ | varchar(64) | Primary Key, Foreign Key, Not Null | User |
| role | บทบาทในห้องเรียน | varchar(32) | Not Null | - |
| joined_at | วันเวลาเข้าร่วม | datetime | Current Timestamp | - |
| recently_visited_at | วันเวลาเข้าชมล่าสุด | datetime | Current Timestamp | - |

ตารางที่ 3.7 พจนานุกรมข้อมูลของตาราง Assignment

| Attribute Name | Data Description | Data Type | Constraints | References |
|----------------|-------------------------------|--------------|-----------------------|------------|
| id | ID ของภาระงานหรือโจทย์ | bigint | Primary Key | - |
| workspace_id | ID ของห้องหรือกลุ่มเรียน | bigint | Foreign Key, Not Null | Workspace |
| name | ชื่อภาระงานหรือโจทย์ | varchar(64) | Not Null | - |
| description | คำอธิบายงาน (สั้น) | varchar(64) | Not Null | - |
| detail_url | ลิงก์ไฟล์รายละเอียดงาน | varchar(128) | Not Null | - |
| memory_limit | ขีดจำกัดหน่วยความจำ | integer | Not Null | - |
| time_limit | ขีดจำกัดเวลา | integer | Not Null | - |
| level | ระดับความยาก | varchar(32) | Not Null | - |
| created_at | วันเวลาที่ภาระงานถูกสร้างขึ้น | datetime | Current Timestamp | - |
| updated_at | วันเวลาอัปเดตงานล่าสุด | datetime | Current Timestamp | - |
| due_date | วันเวลาที่กำหนดส่งงาน | datetime | - | - |

ตารางที่ 3.8 พจนานุกรมข้อมูลของตาราง TestCase

| Attribute Name | Data Description | Data Type | Constraints | References |
|----------------|------------------------|--------------|-----------------------|------------|
| id | รหัส ID ของ Test Case | bigint | Primary Key | - |
| assignment_id | ID ของภาระงานหรือโจทย์ | bigint | Foreign Key, Not Null | Assignment |
| input_url | ลิงก์ไฟล์ข้อมูลนำเข้า | varchar(128) | Not Null | - |
| output_url | ลิงก์ไฟล์ข้อมูลผลลัพธ์ | varchar(128) | Not Null | - |

ตารางที่ 3.9 พจนานุกรมข้อมูลของตาราง Submission

| Attribute Name | Data Description | Data Type | Constraints | References |
|-----------------|--------------------------|--------------|-----------------------|------------|
| id | รหัส ID การส่งงาน | bigint | Primary Key | - |
| assignment_id | ID ของภาระงานหรือโจทย์ | bigint | Foreign Key, Not Null | Assignment |
| user_id | ID ของผู้ใช้ | varchar(64) | Foreign Key, Not Null | User |
| language | ภาษาที่ใช้ในการเขียนโค้ด | varchar(64) | Not Null | - |
| file_url | ลิงก์ไฟล์งาน | varchar(128) | Not Null | - |
| submitted_at | วันเวลาส่งงาน | datetime | Current Timestamp | - |
| compilation_log | ผลการคอมไพล์ | longtext | - | - |

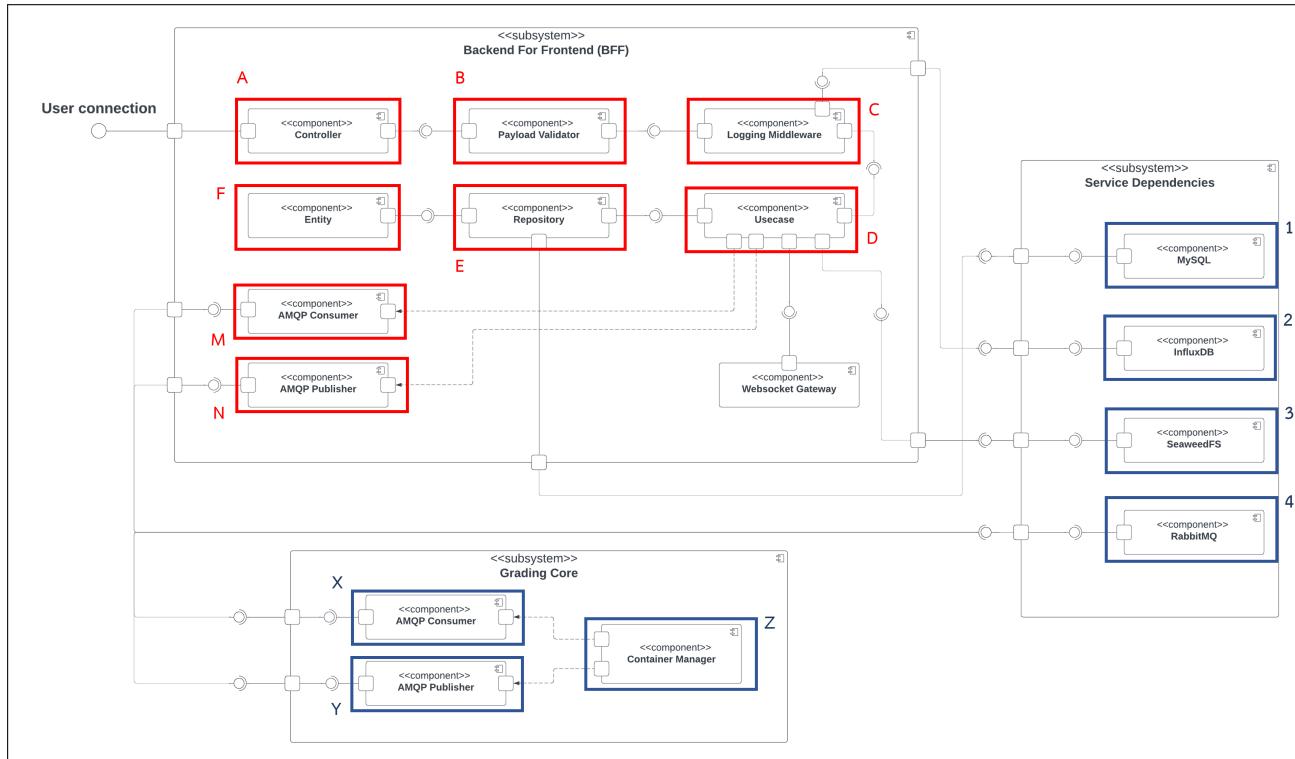
ตารางที่ 3.10 พจนานุกรมข้อมูลของตาราง Submission Result

| Attribute Name | Data Description | Data Type | Constraints | References |
|----------------|-------------------------------------|-------------|------------------------------------|------------|
| submission_id | รหัส ID การส่งงาน | bigint | Primary Key, Foreign Key, Not Null | Submission |
| testcase_id | รหัส ID ของ Test Case | bigint | Primary Key, Foreign Key, Not Null | TestCase |
| status | สถานะผลการทดสอบงาน | varchar(32) | - | - |
| status_detail | รายละเอียดสถานะการทดสอบงาน | varchar(32) | - | - |
| memory_usage | หน่วยความจำที่ถูกใช้ไปในการประมวลผล | integer | - | - |
| time_usage | เวลาทั้งหมดที่ใช้ในการประมวลผล | integer | - | - |

3.5 แผนผัง UML

สำหรับในส่วนแผนผัง UML ทางคุณผู้จัดทำจะนำเสนอแผนผังมาตรฐานแบบ UML ที่อธิบายโครงสร้างและการทำงานของซอฟต์แวร์ ที่ได้ออกแบบไว้ทั้งหมด ซึ่งประกอบด้วยแผนผังทั้งหมด 3 ประเภทดังต่อไปนี้

3.5.1 แผนผังแสดงโครงสร้างและส่วนประกอบของซอฟต์แวร์



รูปที่ 3.36: แผนภาพแสดงโครงสร้างและส่วนประกอบของซอฟต์แวร์ วาดด้วย LucidChart

จากแผนภาพในรูปที่ 3.36 องค์ประกอบหลักของระบบซอฟต์แวร์ประกอบไปด้วย 3 ส่วนหรือระบบย่อยดังนี้

1. Backend For Frontend (BFF) คือระบบย่อยหลักที่ทำหน้าที่การประมวลผลการแสดงออกหน้าเว็บไซต์ โดยมีองค์ประกอบดังนี้

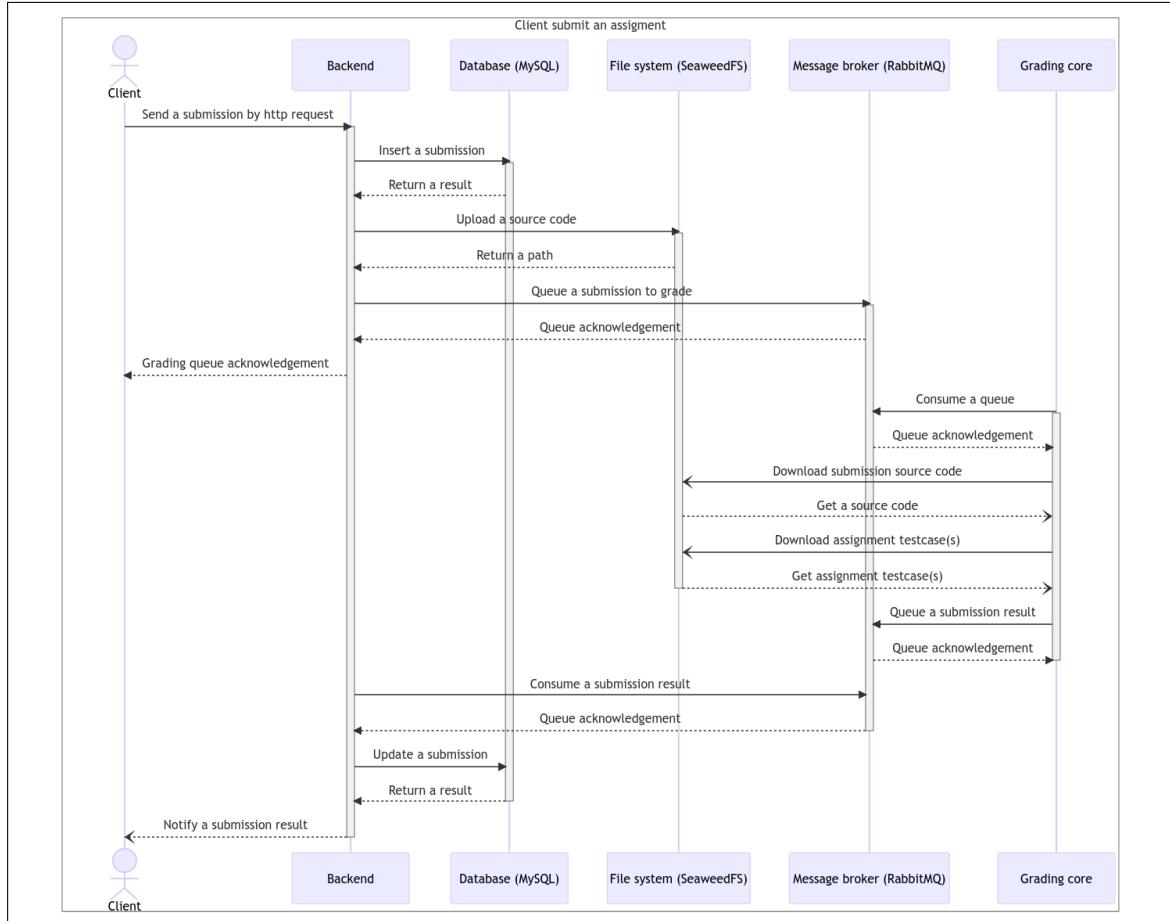
- ส่วนแรกที่ผู้ใช้เข้ามายังระบบ คือ Controller (กรอบแดง A) เป็นส่วนที่คอยรับการเชื่อมต่อของผู้ใช้เข้ามาภายใต้ระบบ
- Controller จะส่งต่อการทำงานให้อีกองค์ประกอบหนึ่งที่ชื่อ Payload Validator (กรอบแดง B) ซึ่งจะทำการตรวจสอบความถูกต้องของการเชื่อมต่อ (Connection)
- การเชื่อมต่อทั้งหมดจะถูกเก็บเป็นสถิติ ผ่านองค์ประกอบที่ชื่อว่า Logging Middleware (กรอบแดง C)
- ส่วนประกอบที่ชื่อ Usecase (กรอบแดง D) จะทำหน้าที่ประมวลผลการเรียกข้อมูล (Request) จากผู้ใช้
- ถ้าหากต้องการเข้าถึงฐานข้อมูล ส่วนประกอบที่ชื่อ Repository (กรอบแดง E) จะทำเข้าถึงฐานข้อมูล และหยิบดึง (Query) ข้อมูลส่งกลับให้ผู้ใช้
- ข้อมูลที่ถูกดึงมาจากฐานข้อมูล มีรูปแบบหรือลักษณะ (Format) ที่ถูกนิยามโดยส่วนประกอบที่ชื่อ Entity (กรอบแดง F)

2. Service Dependencies คือระบบย่อยสำหรับเก็บข้อมูลโดยเฉพาะ ประกอบด้วยองค์ประกอบย่อยดังนี้

- MySQL (กรอบน้ำเงิน 1) เป็นฐานข้อมูลตารางหรือ Relational Database ที่ทำหน้าที่เก็บข้อมูลที่ໄปที่จำเป็นต่อ feature ต่าง ๆ ของซอฟต์แวร์ ตั้งแต่ข้อมูลผู้ใช้ไปจนถึงข้อมูลโลจิสติก ภาระน้ำหนา การบ้านหรือข้อสอบทั้งหมด พร้อมทั้งกรณีทดสอบหรือเทส เสส ระบบ BFF เข้าถึงฐานข้อมูลนี้ผ่านองค์ประกอบย่อยชื่อ Repository (กรอบแดง E)

- ตามที่ได้กล่าวไปข้างต้น ข้อมูลการเขียนต่อทั้งหมดจะถูกเก็บเป็นสถิติในองค์ประกอบย่อย Logging Middleware โดย ข้อมูลการเขียนต่อทั้งหมดนั้นจะถูกนำไปเก็บใน InfluxDB (กรอบน้ำเงิน 2) ซึ่งเป็นฐานข้อมูลประเภทหัวเวลาหรือ Time-series Database ที่จะเก็บข้อมูลการเขียนต่อทั้งหมดนั้นในเป็นช่วงๆ เวลาไป
 - ข้อมูลที่เป็นไฟล์ทั้งหมด ไม่ว่าจะเป็นไฟล์ pdf ของโจทย์ (Assignment), ไฟล์ expected input และ output (Test Case) หรือแม้กระทั่งรูปไฟล์ของผู้ใช้ เป็นต้น จะถูกเก็บในระบบบริการไฟล์หรือ File Service หรือ Filer ซึ่ง SeaweedFS (กรอบน้ำเงิน 3)
 - เนื่องจากระบบซอฟต์แวร์นี้ใช้การ Publish และ Consume ในการแลกเปลี่ยน request และ response ระหว่างระบบ ย่อย BFF และ Grading Core จึงจำเป็นต้องมี Message Broker ที่จะเก็บพัก message ต่างๆ เหล่านี้ ซึ่ง RabbitMQ (กรอบน้ำเงิน 4)
3. **Grading Core** คือระบบย่อยสำหรับการประมวลผลและ Compile ไฟล์งานหรือโปรแกรมของผู้ใช้ ที่ส่งเข้ามาในระบบให้ตรวจสอบ
- การรับการเรียกขอ (request) ตรวจงานจากระบบย่อย Backend For Frontend จะทำผ่านองค์ประกอบ AMQP Consumer (กรอบน้ำเงิน X) โดย Consumer นี้จะไปรับคำขอที่ทาง BFF ส่งเข้า RabbitMQ (กรอบน้ำเงิน 4) ผ่าน Publisher ของตัว BFF (กรอบแดง N)
 - หลังจากตรวจสอบ ก็จะส่งผลลัพธ์ผ่านองค์ประกอบ AMQP Publisher ของ Grading Core (กรอบน้ำเงิน Y) เข้า RabbitMQ (กรอบน้ำเงิน 4) เพื่อให้องค์ประกอบ AMQP Consumer ที่อยู่ในระบบย่อย BFF (กรอบน้ำเงิน M) Consume ไปแสดงผล ที่หน้าเว็บ

3.5.2 แผนภาพลำดับการทำงาน



รูปที่ 3.37: แผนภาพลำดับการทำงานของการส่งโปรแกรมขึ้นระบบตรวจ วาดด้วย Mermaid JS

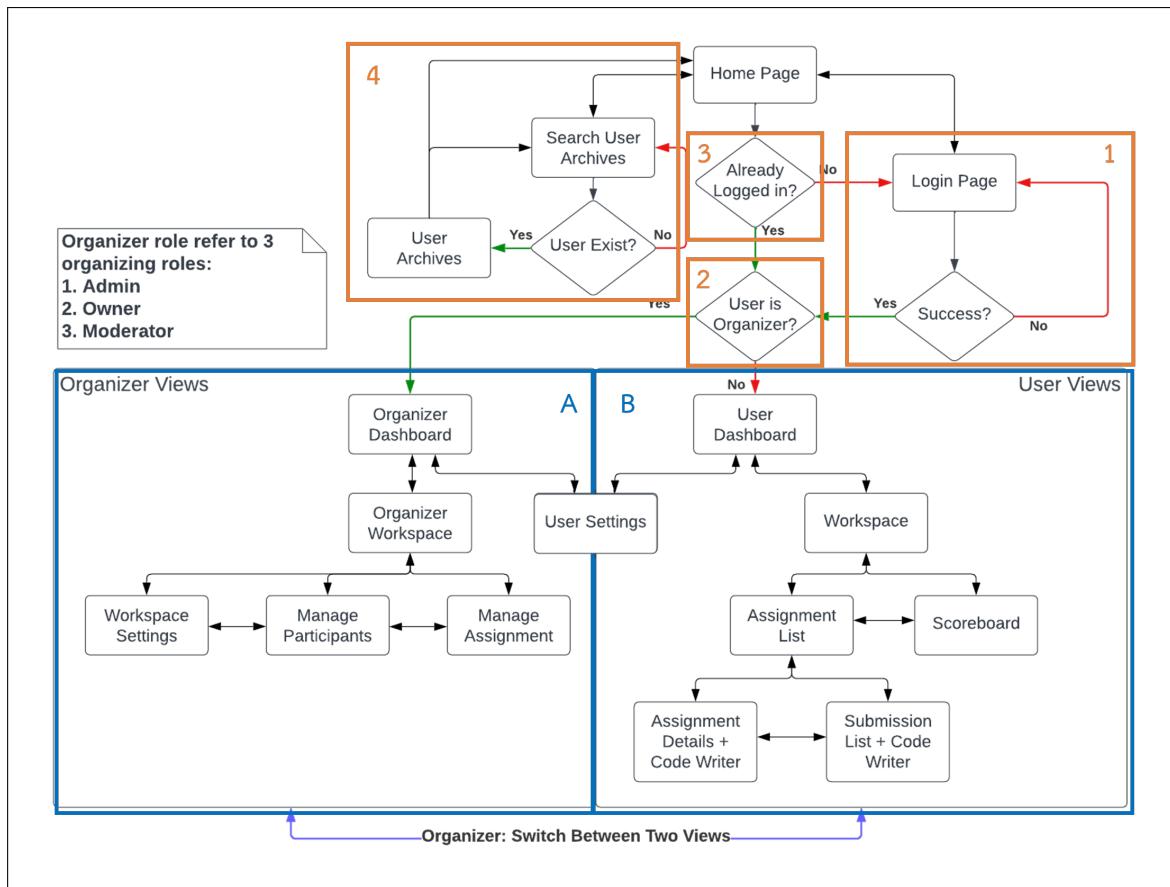
แผนภาพลำดับการทำงานในรูปที่ 3.25 อธิบายกระบวนการทำงานของระบบการตรวจไฟล์งานเขียนโปรแกรมที่ผู้ใช้งานเข้ามาให้ตรวจ โดยมีการ actor ดังนี้ (เรียงลำดับขั้นตอน จากบนสุดสู่ล่างสุดของแผนผัง)

- Client (ผู้ใช้) เริ่มต้นกระบวนการโดยส่งคำขอทาง HTTP ไปยัง Backend เพื่อส่งไฟล์งานเขียนโปรแกรม
- Backend (ส่วนหลักของระบบ) รับคำขอและทำการบันทึกไฟล์งานเขียนโปรแกรมลงในฐานข้อมูล MySQL หลังจากนั้นส่งผลการทำงานกลับไปยัง Client พร้อมกับข้อมูลผลการบันทึก
- File system (SeaweedFS) ใช้เพื่อกีบโค้ดของงานและข้อมูลเกี่ยวกับงานต่างๆ เมื่อ Backend อัพโหลดไฟล์งานเขียนให้ File system ก็จะคืน path ที่เก็บไฟล์กลับไปให้ Backend
- Message broker (RabbitMQ) มีหน้าที่จัดการการส่งคิวของงานตรวจสอบความแน่นโค้ด หลังจากนั้นส่งการยืนยันการจัดคิวกลับไปยัง Backend และจากนั้น Backend ก็จะส่งการยืนยันต่อไปให้ Client ด้วย
- Grading core เป็นส่วนของระบบที่รับผิดชอบในการดำเนินการตรวจความแน่นโค้ด โดยมีกระบวนการตั้งต่อไปนี้
 - Grading core ทำการเช็ค (consume) queue ใน Message broker เพื่อดูว่ามีไฟล์งานเขียนโปรแกรมถูกส่งมาให้ตรวจหรือไม่
 - Grading core เริ่มต้นกระบวนการโดยการทำการตั้งคิวของงานและข้อมูลของงานจาก File system
 - หลังจากที่ทำการตรวจความแน่นเสร็จ Grading core จะทำการส่งผลการตรวจความแน่นกลับไปยัง Message broker เพื่อจัดคิวการแจ้งผลตรวจความแน่นของงาน

- Backend ต่อมาทำการดึงผลการตรวจสอบคะแนนที่ Message broker แจ้งกลับมา และทำการอัพเดทข้อมูลของงานในฐานข้อมูล MySQL พร้อมส่งผลการตรวจสอบคะแนนกลับมา มาแจ้งให้ Client

กระบวนการนี้ช่วยในการจัดการและควบคุมกระบวนการตรวจสอบคะแนนงานเขียนโค้ดของระบบ ให้เป็นระเบียบและมีประสิทธิภาพ มีการใช้ Message broker เพื่อรวมและควบคุมการทำงานของ Grading Core และ Backend ให้ทำงานร่วมกันได้อย่างมีประสิทธิภาพ

3.5.3 แผนผังแสดงโครงสร้างและการนำทางระหว่างหน้าของเว็บไซต์



รูปที่ 3.38: แผนผังแสดงโครงสร้างและการนำทางระหว่างหน้าของเว็บไซต์ วาดด้วย LucidChart

จากรูปที่ 3.26 ข้างต้น เป็นแผนผังโครงสร้างของหน้าเว็บของซอฟต์แวร์ของเรามา และการนำทาง ไปมาระหว่างแต่ละหน้าเว็บของซอฟต์แวร์ เริ่มต้นจากหน้าแรก (Home page) ของซอฟต์แวร์เรา จากหน้าแรกก็จะสามารถเข้าถึงได้สองหน้า โดยไม่ต้องเข้าสู่ระบบคือหน้าค้น หาเบียนประวัติผู้ใช้เก่า (Search User Archives) (กรอบที่ 4 3.26) ซึ่งเป็นจะเป็นหน้าสำหรับกรอกรหัสหรือชื่อของผู้ใช้ที่ต้องการจะค้น หาเบียนประวัติ ถ้าหากค้นแล้วไม่พบผู้ใช้ขึ้นดังกล่าว ก็จะถูกตีกลับไปที่หน้าเดิม แต่ถ้าค้นเจอก็จะถูกนำพามาหน้าแสดงข้อมูลของผู้ใช้ที่ได้ ค้นไปเมื่อครู่ (ตรงกับหน้าjoinในรูปที่ 3.22)

จากหน้าแรก ถ้ากดปุ่ม “Login” ก็จะนำพาผู้ใช้เข้าสู่ระบบยืนยันตัวตน ถ้าหากผู้ใช้ยังไม่ได้เข้าสู่ระบบหรือไม่ได้มี session อยู่ในระบบ ณ ตอนนั้น ก็จะถูกนำไปป้อนหน้าเข้าสู่ระบบ (Login Page) (กรอบที่ 1 ในรูปที่ 3.26) ถ้าหากกรอกข้อมูลผิดอาทิเช่น ชื่อผู้ใช้/อีเมล รหัสผ่าน เป็นต้น ก็จะถูกนำทางกลับไปเข้าสู่หน้าเข้าสู่ระบบหน้าเดิม จนกว่าจะกรอกข้อมูลถูกต้อง

ในทางกลับกันถ้าหากผู้ใช้เข้าสู่ระบบแล้ว หรือมี session อยู่ในระบบ ณ ตอนนั้น ก็จะถูกนำพาเข้าสู่แผงควบคุมทันที (กรอบที่ 3 ในรูปที่ 3.26) ทั้งนี้ก็ขึ้นอยู่กับบทบาท (role) ของผู้ใช้ที่เป็น Organizer (หรือบทบาทผู้ดูแลทั้งสามตำแหน่ง: Admin, Owner, และ Moderator ตาม requirements) หรือเป็นผู้ใช้ทั่วไป (หรือ Registered User, Student, Attendee ตาม requirements) บทบาทผู้ใช้แต่ละบทบาท จะถูกนำไปสู่รูปแบบແນວคุณลักษณะแบบ (กรอบที่ 2)

ถ้าหากเป็น Organizer ก็จะนำพาไปແຜງຄວບຄຸມທີ່ໃນມູນມອງຂອງ Organizer (Organizer view) (กรอบ A ໃນຮູບທີ່ [3.26](#)) ຜຶ້ງສອດຄລ້ອງກັບສ່ວນປະສານຜູ້ໃຊ້ຮູບທີ່ [3.11](#) ໂດຍຈາກແຜງຄວບຄຸມຂອງ Organizer ກົດສາມາດຈະເຂົ້າສົ່ງ ໜ້າຕັ້ງຄ່າຂອງບັນຫຼິດນອງ (User Settings) ແລະທ້ອງເຮືອນຫວູ້ອົກລຸ່ມເຮືອນ (Organizer workspace) ທັ້ງໝາດທີ່ Organizer ດັນນັ້ນ ຖໍ່ເປັນເຈົ້າຂອງຫວູ້ເຂົ້າສົ່ງໄດ້ ພັດງຈາກກົດເລືອກທ້ອງເຮືອນຫວູ້ອົກລຸ່ມເຮືອນເຂົ້າມາແລ້ວ ກົດສູກນຳພາເຂົ້າມາໜ້າທັກຂອງທ້ອງເຮືອນຫວູ້ອົກລຸ່ມເຮືອນນັ້ນ ຈຸ່າ (ຕຽບກັບໜ້າຈອໃນຮູບທີ່ [3.13](#)) ຜຶ້ງສາມາດເຂົ້າສົ່ງ ໜ້າຕັ້ງຄ່າທ້ອງເຮືອນຫວູ້ອົກລຸ່ມເຮືອນ (Workspace Settings), ໜ້າບວິທາຮັດການສາມາຊືກທ້ອງເຮືອນຫວູ້ອົກລຸ່ມເຮືອນ (Manage Participants), ແລະໜ້າບວິທາຮັດການກວະກາງ (Manage Assignment)

ຄ້າຜູ້ໃຊ້ໄດ້ມືບທາຫາເປັນ Organizer ແຕ່ເປັນຜູ້ໃຊ້ທົ່ວໄປ (User) ກົດສູກນຳພາແຜງຄວບຄຸມໃນມູນມອງຂອງ User (User View) (กรอบ B ໃນຮູບທີ່ [3.26](#)) ຜຶ້ງຕຽບກັບຮູບໜ້າຈອໃນຮູບທີ່ [3.4](#) ຈາກໜ້າແຜງຄວບຄຸມທັກລ່າງ User ສາມາດຈະເຂົ້າສົ່ງ ໜ້າຕັ້ງຄ່າຂອງບັນຫຼິດນອງ (User Settings) ແລະທ້ອງເຮືອນຫວູ້ອົກລຸ່ມເຮືອນ (Workspace) ທັ້ງໝາດທີ່ User ດັນນັ້ນ ຈຸ່າ ໄດ້ເຂົ້າວ່າມ່ວຍຫວູ້ອົກຍັດເຂົ້າ ພັດງຈາກກົດເລືອກທ້ອງເຮືອນຫວູ້ອົກລຸ່ມເຮືອນເຂົ້າມາແລ້ວ ກົດສູກນຳທາງໜ້າທັກຂອງທ້ອງເຮືອນຫວູ້ອົກລຸ່ມເຮືອນນັ້ນ ຈຸ່າ ຜຶ້ງກົດໜ້າຈອແສດງຮາຍກາງກວະກາງ (Assignment List) ທັ້ງໝາດໃນກລຸ່ມເຮືອນຫວູ້ອົກລຸ່ມເຮືອນນັ້ນ ຈຸ່າ (ເປັນໜ້າຈອໃນຮູບທີ່ [3.12](#)) ນອກຈາກໜ້າຮາຍກາງກວະກາງແລ້ວ User ກົດສູກນຳກົດໄປຢູ່ຕາງໆຄະແນນ (Scoreboard) ຂອງທ້ອງເຮືອນຫວູ້ອົກລຸ່ມເຮືອນນັ້ນໄດ້ຕ້ວຍ

ຄ້າຜູ້ໃຊ້ທົ່ວໄປຫວູ້ອົກລຸ່ມເຮືອນ ກົດເລືອກກວະກາງໃນໜ້າຮາຍກາງແສດງກວະກາງ ກົດສູກນຳພາໄປໜ້າເຂົ້າເປີຍໂປຣແກຣມທີ່ປະກອບແຜງເທິ່ງໂປຣແກຣມ (Code Writer) ກັບສ່ວນແສດງເນື້ອຫາໂຈທຍ (Assignment Details) ໜ້າຈອດັກລ່າວທຽບກັບໜ້າຈອໃນຮູບທີ່ [3.20](#) ຜູ້ໃຊ້ສາມາດທີ່ຈະເປີ່ຍັນແຜງແສດງເນື້ອຫາໂຈທຍເປັນແຜງແສດງຮາຍກາງກວະກາງສ່າງໂປຣແກຣມຕຽບ (Submission List ໄດ້) ຜຶ້ງຕຽບກັບໜ້າຈອໃນຮູບທີ່ [3.21](#)

ຂອ້ອກທີ່ຄວຽ້ງເພີ່ມເຕີມຄື້ອງ ຄ້າຫາກຜູ້ໃຊ້ມືບທາຫາເປັນ Organizer ຜູ້ໃຊ້ຈະສາມາດສັບເປີ່ຍັນມູນມອງ (Switch View) ຈາກ Organizer ເປັນ User ຫວູ້ອົກລຸ່ມເຮືອນ ແຕ່ ເປັນ Organizer ໄດ້ ຕ້ວຍປຸ່ມ “Switch to Normal Mode” ຫວູ້ອົກລຸ່ມເຮືອນ ແຕ່ “Switch to Organizer Mode” ຕາມຄຳຕັບ (ມີເຖິງເຫັນໃນຮູບທີ່ [3.11](#) ແລະ [3.12](#) ດຽວມູນຂວາບນ)