

## Software Requirements Report

The topic is grading platform for programming system.

Group 1

1. Introduction (This section should include your problem statement, the objectives of the project, and the final product of the project. Briefly introduce the readers to your system, and what you are working on to point out the necessity of your project.)

The topic is grading platform for programming system.

การตรวจสอบโค้ดโปรแกรมสามารถทำได้ง่ายๆ เพียงแค่เราโหลดโปรแกรมอย่าง Pycharm Dev-c มาติดตั้งใน PC notebook หรืออุปกรณ์ทางอิเล็กทรอนิกส์ที่สามารถทำได้ เราก็สามารถตรวจสอบโค้ดโปรแกรมที่ต้องการโดยการเขียนโค้ดโปรแกรม และกด Run เพื่อ Compile ก็สามารถตรวจสอบความถูกต้องได้ แต่เป็นเพียงการตรวจสอบความถูกต้องที่คนเขียนสามารถดูได้เท่านั้น

ปัจจุบันมี Web application ที่สร้างมาเพื่อการฝึกเขียนโค้ดโปรแกรมและตรวจสอบโค้ดโปรแกรมไปพร้อมๆ กัน กล่าวง่ายๆ ให้เข้าใจคือ เมื่อเราเขียนโปรแกรมลงบน Web application ดังกล่าวก็สามารถกด Run เพื่อ Compile ภายใน Web application ได้เลยและสามารถตรวจสอบ Syntax หรือ Error ของโค้ดโปรแกรมได้ด้วย เช่น Codecademy Codewars แต่ในบางกรณีหากต้องการนำไปใช้ภายในองค์กร ต้องมีค่าใช้จ่ายเสมอ รวมไปถึงการใช้งาน Feature ต่างๆ ความยากของการใช้งาน Web application เราจึงคิดพัฒนา Web application เพื่อเป็นไปตามความต้องการของ customer ในด้านการใช้งาน ที่สะดวกสบายมากยิ่งขึ้น และเข้าใจง่าย มี Feature ที่ไม่ซับซ้อน มากจนเกินไป

Web application ที่เรากำลังพัฒนาขึ้นมานี้เป็น Web application เพื่อใช้ในการส่ง Code และตรวจสอบความถูกต้องของ Code โดยจะมี Customer อยู่ 2 กลุ่มคือ Admin (อาจารย์และ TA) และ User (นักศึกษา) เมื่อ Admin และ User ต้องการเข้าสู่ระบบ สามารถใช้ Teacher ID หรือ Student ID และ Password ที่ตั้งไว้ในระบบ Login เข้าสู่ระบบ

- Admin ซึ่งจะเป็นผู้สร้าง Class room เพื่อให้ User เข้ามาส่งงานและ Activity อื่นๆ เมื่อ Admin สร้างห้องจะได้รับ Key code ของห้องนั้นๆและนำ Key code ไปให้แก่ User เพื่อให้ User join เข้ามาใน Class room ทางadmin จะเป็นผู้สร้างโจทย์ที่สามารถกำหนดคุณลักษณะต่างๆได้ เช่น เวลาส่งงานของ User ภาษาที่ใช้ในการเขียน และกำหนด Test case ต่างๆเป็นต้น
- User จะได้รับ Key code จาก admin เพื่อ Join เข้าสู่ Class room เมื่อ admin สร้างโจทย์ User จะเห็นโจทย์ เวลาในการส่งงาน User จะเป็นผู้เขียนโค้ดได้ทั้งบนเว็บและส่งหรือเขียนบนโปรแกรมในคอมพิวเตอร์และอัปโหลดส่งผ่าน Web แล้วจะสามารถเห็นคะแนนได้เลย

2. User Stories (This section should include user stories for your software. Putting requirement in customers' perspective, you should describe what you want to do and how you expect to do that. You should explain user stories for every requirement.)

### ในด้านของ USER

แฮร์รี่ เอ้ย!! แฮร์รี่ เอ้ย!! แฮร์รี่ เอ้ย!!! ถูกแล้ว แฮร์รี่ได้มัธยมปลายผู้เรียนอยู่ในสายคอมพิวเตอร์ วันหนึ่งอาจารย์ได้สั่งการบ้านให้แฮร์รี่นั้นไปเขียน Code เพื่อคำนวณเกรดของเพื่อนนักเรียน โดยอาจารย์นั้นได้ให้แฮร์รี่เขียนแล้วนำมาส่งในเว็บไซด์ที่อาจารย์ให้มา โดยมีเวลาให้ทำ 10 วัน นับตั้งแต่วันที่ส่ง และอาจารย์บอกว่า “ถ้าใครส่งช้ากว่ากำหนดจะถูกหักคะแนนวันละ 10% แต่ถ้าเกิน 3 วันจะส่งไม่ได้อีก” หลังจากที่แฮร์รี่เขียน Code เสร็จ เขาก็เข้าไปที่เว็บไซด์ที่อาจารย์ให้แฮร์รี่เข้าใช้งานเป็นครั้งแรกเขาเลยกดสมัคร โดยใช้ Email ของโรงเรียนที่จะมีทุกคน หลังจากแฮร์รี่สมัครเสร็จแล้วเข้ามาสู่หน้าการใช้งาน และแฮร์รี่ได้นำ Key code ที่อาจารย์ให้มากรอกลงไปเพื่อเข้าสู่ห้องส่งงานที่อาจารย์ได้สร้างไว้ เขาก็ร้องขึ้นมาว่า “อ้าว!!! ทำไมเราไม่มาเขียน Code ในเว็บนะ” จากนั้นแฮร์รี่ก็กดไปที่ปุ่ม Upload เพื่อส่งงาน แต่พอแฮร์รี่กด Submit งานที่ส่งไป ปรากฏว่ามัน Compile ไม่ผ่าน แฮร์รี่ก็เลยแก้ไข Code แล้ว Upload เข้าไปใหม่แล้วก็ยัง Compile ไม่ผ่าน แฮร์รี่ก็เลยขอคำปรึกษาจากเพื่อนร่วมห้องชื่อ “สมิง” และสมิงได้ให้คำแนะนำแก่แฮร์รี่ หลังจากแฮร์รี่ Compile ผ่าน แฮร์รี่รู้สึกดีใจมาก เพราะว่าเขาได้คะแนน 9 คะแนน จาก 10 คะแนนเต็ม หลังจากที่แฮร์รี่ส่งงานเสร็จเขาก็ลองใช้งานเว็บไซด์ในส่วนต่างๆ อยู่ๆก็มีหน้าต่างแจ้งเตือนขึ้นมา เป็นข้อความจากอาจารย์ของแฮร์รี่ อาจารย์ได้เข้ามาชมเชยการเขียน Code ของแฮร์รี่ ผ่านมาสองสัปดาห์แฮร์รี่จะเข้าไปทำงานชิ้นใหม่ แต่ปรากฏว่าแฮร์รี่ลืม Password สำหรับ Login แฮร์รี่จึงทำการเข้าฉานแล้วนึกขึ้นได้ว่ามีปุ่ม Reset password นี้นะ แฮร์รี่จึงกดปุ่ม Reset password และได้ Password ใหม่ แล้วจึงใช้งานได้ ....  
Coming soon

### ในด้านของ Admin

ชื่อครึ่งหลังของพระเอกหนังที่มีสายฟ้าอยู่บนหน้าผาก? แฮร์รี่พ็อตเตอร์ ผิด!! แฮร์รี่ ผิด !! พ็อตเตอร์ ผิด!! เอ้ย!! ถูกแล้ว พ็อตเตอร์สำเร็จการศึกษาจากโรงเรียนมัธยมสมิงคอนเวน แล้วยื่นแอดเข้าคณะวิศวกรรมคอมพิวเตอร์ ที่มหาวิทยาลัยเทคโนโลยีเสือสมิง พ็อตเตอร์เคยมีประสบการณ์ด้านการเขียนโปรแกรมมาก่อน และชอบเขียนโปรแกรมเป็นงานอดิเรก ได้ทำ Project มากมายเกี่ยวกับโปรแกรม เช่น โปรแกรมคำนวณเกรด โปรแกรมคำนวณค่าขนส่ง เป็นต้น หลังจากเปิดเทอมในรั้วมหาวิทยาลัยกระยะหนึ่ง อาจารย์ได้มีการนัดสอบเขียนโปรแกรม โดยให้เขียนเป็นภาษาPython และให้เซฟชื่อไฟล์เป็นรหัสนักศึกษา การส่งไฟล์ คือ อาจารย์จะมี URL ไว้ให้สำหรับส่งงาน เมื่อเราเข้าถึง URL จะมีช่องให้ Browse ไฟล์ แล้วกดอัปโหลด แล้วจะขึ้นเป็น Test case ทั้งหมด 3 ครั้งและจะบอกว่าเราผ่าน Case ไหนบ้าง หลังจากการสอบ พ็อตเตอร์ได้กลับไปที่พัก และมีไอเดียอยากจะทำเว็บสำหรับการส่งโค้ดเข้าไปตรวจ เหมือนที่อาจารย์ให้ส่งเข้าไป Test case แต่จะเพิ่ม Optionทั้งด้าน Front end และ ด้าน Back end ให้มากกว่าของอาจารย์ วันรุ่งขึ้นพ็อตเตอร์ได้ไปหาอาจารย์ที่สอนโปรแกรมนิ่ง และปรึกษาเกี่ยวกับหลักการทำงานของเว็บที่อาจารย์ให้ส่งไฟล์ไป Test case และได้ถามอาจารย์ต่อเกี่ยวกับปัญหาต่างๆ และควรมีฟังก์ชันอะไรเพิ่มเติมอีก ได้คำตอบว่า อยากได้ IP Addressที่login เก็บ History interface ที่ตรงตามความต้องการของ User การแอดและลบนักศึกษาจากคลาส การเพิ่ม User เข้าคลาสแบบอัตโนมัติ การ Login ให้เป็น Universal แต่ User กับ Admin จะเห็น Function ที่ต่างกันและไม่ควรจะทำเป็น Pop-up login Admin สามารถ Keep track หรือติดตาม User ได้ (หมายถึง Data log หรือเก็บ Statistic จาก User ได้) Admin และ User สามารถติดต่อกันผ่านหน้าเว็บได้เลย แสดง Test case แคบงกรณี เช่น มี 10 Test case แต่แสดงแค่ 5 Test case มี Storage ไว้เก็บ Code และเวลา Submit เข้ามาจะต้องมี IP, Score, When, Code แจ้งให้ Admin ทราบ กำหนดขนาดไฟล์ให้ไม่เกินที่กำหนด ระบบ Auto เปิดปิดเวลาส่งงาน ( มี Penalty หรือไม่ขึ้นอยู่กับ Admin ) สามารถเอา History หรือ Statistic ของ User ออกมา Plot กราฟได้ ( ในกรณี Admin ต้องการสำรวจ User ของ Admin ว่ามีความกระตือรือร้น หรือ ต้องการทำอะไรแล้วแต่ ) พ็อตเตอร์จึงอาสาพัฒนาเว็บนี้ต่อ

3. Customer Requirements (This section must include requirements from your customer. You must conduct customer interviews, and summarize the requirements from the user. You must include both functional and non-functional requirements, and explain the details. Make sure to focus on “WHAT” you will implement to satisfy each requirement. Additional features may be included.

+ Functional.

- Non-Functional.

\* How we archive this.

+ Admin อยากให้ Admin และ User ติดต่อกันผ่านหน้า Web application ได้เลย

- Admin และ User จะต้องติดต่อกันได้ตลอดเวลาที่ยัง Online อยู่หรือทิ้งเป็นข้อความ Offline

\* สร้างได้ด้วยการให้ Front end & Back end โดยนำข้อมูลที่รับมา ยิงเข้า Database เช่น IP Address, Username, Time, Activities แล้วเก็บเป็น History ด้วย

+ Admin ต้องการระบบ Auto เปิด-ปิดเวลาส่งงาน

- เมื่อ User เริ่มทำงานจะเริ่มนับเวลาถอยหลัง (หรือด้วยวิธีอื่น) จนหมดเวลาส่งงานก็จะปิดระบบส่งงาน

\* อาจจะเป็นด้วยการนับเวลาถอยหลังหรือ Time ของระบบเป็นตัวบ่งชี้ (ไม่ใช่เวลาของ Client แต่เป็นของ Server)

+ Admin เชื่อมแต่ละคลาสด้วยรูปแบบคล้ายคลึงกับ Google classroom

- เมื่อ User Log-In เข้ามาแล้ว จะมีสิทธิ์เข้าคลาสไหนก็ได้ ที่รู้ Key

\* มีสัญลักษณ์หรือปุ่มให้กดเมื่อแสดงหน้าต่างใส่ Key ถ้า Key ถูกต้อง ระบบจะ Grant access คลาสนั้นๆ ให้กับ User นั้นๆ

+ Admin อัปเดต Test case เดียวจบ

- อัปเดต Test case เดียวแล้วจะต้องครอบคลุมทุก Test case ให้ผลลัพธ์ถูกต้องตามที่ควร

\* ต่อ Code เข้าใน Code ของ User แล้วตรวจสอบความถูกต้องด้วย (อาจจะใช้ Built-In Function) Test case ที่เขียนมาอย่างดีแล้ว

+ Admin สามารถกำหนดตำแหน่งและสิทธิ์

- ผู้มีสิทธิ์จะต้องมีความสามารถในการจัดการตามสิทธิ์ที่มีได้

\* สร้าง Scope auth ให้ในแต่ละตำแหน่ง

+ Admin จะกำหนด Template คร่าวๆ ให้ User

- User เขียน Code และจะต้องไม่มี Error ที่เกิดจากสิ่งที่ไม่ควรเกิดขึ้น เช่น ลืม Import module เป็นต้น

\* กำหนด Template ให้ User เลย จะ Edit ได้หรือไม่ขึ้นอยู่กับ Admin

## User

+ Function แก้ปัญหาลืม Password

- ใช้ Email Kmutt ช่วยแก้ปัญหาได้ User จะต้องเห็น Password หรือ Reset password ได้

\* ส่ง Email เป็นลิงค์ Reset password หรือแค่แสดงไปที่email Kmutt

+ มี Output ให้ User ดู

- User จะต้องรู้ว่า Code นั้นผ่านหรือไม่อ่านและได้กี่คะแนน Case ไหนผ่านไม่ผ่านบ้าง

\* ตรวจเช็คค่า Case ทั้งหมดผ่านหรือไม่ผ่าน แล้วนำ Case ผ่านหรือไม่ผ่านเป็นคะแนนที่เป็น Sum ของทุก Case ไปแสดงเป็น Output ยกเว้น Hidden case

## Admin & User

- + Log-In ที่เป็น Universal

- Admin และ User จะต้อง Log-In ทางเดียวกันแต่ Api ที่ Server จะ Provide จะต้องแบ่งกันอย่างถูกต้อง Admin ส่วน Admin User ส่วน User

- + User interface (UI)

- ใช้งานง่าย เข้าใจง่าย เข้าใจได้ทันทีว่าควรจะใช้งานอย่างไร

- \* ขึ้นอยู่กับทีม Front end และ การ Design เป็นอย่างมาก