

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 Defect

Defect Management [2] คือ การจัดการข้อผิดพลาด (Defect) ที่พบในช่วงของ Phase Testing โดย Tool ที่ผู้ใช้คือ Mantis การนำ Tool เข้ามาใช้ใน Project นั้นๆ เพื่อใช้ Report Defect ที่พบเพื่อแจ้งทีมที่เกี่ยวข้อง และเพื่อใช้ข้อมูลสรุปปัญหาของโปรเจกต์นั้น ๆ ว่าจะสามารถ Launch ได้ตาม Plan หรือไม่

Defect คือ ปัญหาที่พบในการทดสอบระบบ ซึ่งปัญหาเหล่านี้อาจกระทบต่อ Function การทำงานของระบบ เช่น ระบบแสดง Error ต่างๆ หรือ Defect ที่พบอาจจะไม่กระทบกับ Function การทำงาน เช่น การแสดงผลที่อาจเกิดจาก Design หรือ การแสดงผลของข้อความ ซึ่งปัญหาเหล่านี้จะไม่ส่งผลกระทบต่อระบบ

Defect Life Cycle คือ วงจรการทำงานของ Defect ที่เริ่มตั้งแต่เมื่อพบ Defect แล้ว Assign ให้ทางทีม Developer แก้ไข จนกระทั่ง Developer แก้ไขเสร็จ ซึ่งมีรายละเอียด ดังนี้

2.1.1.1 Defect Status

Defect Status เป็นการบอกถึงสถานะของ Defect ที่พบ Status ที่ใช้ปัจจุบันจะมีอยู่ 5 Status

1. Assign : เมื่อพบ defect จะ Assign ไปให้ทาง Team Develop ที่รับผิดชอบ
2. Resolve : เมื่อ Developer แก้ไข Defect ที่พบ และ upload Code ขึ้น Test Environment
3. Feedback : ไม่แก้ไข Defect ที่พบเนื่องจากสาเหตุใดๆ
4. Acknowledged : การแก้ไขโปรเจกต์ ต้องรอข้อสรุป หรือ solution จากผู้ที่เกี่ยวข้องกับโปรเจกต์
5. Closed : Defect ได้รับการแก้ไขถูกต้อง

2.1.1.2 Defect Tracking

Defect Tracking สำหรับ Tool ที่ใช้เป็นตัวช่วยในการจัดการกับ Defect ที่พบ คือ Mantis ใช้ในการ log Defect ที่พบ เพื่อแจ้งทีมที่เกี่ยวข้องให้เข้าไปดูรายละเอียดของ Defect และแก้ไขในส่วนที่รับผิดชอบ ข้อมูล Defect เหล่านี้จะใช้ในการสรุปปัญหาของระบบว่าจะสามารถ Launch ได้ตาม Plan หรือไม่

2.1.1.3 Defect Severity

Defect Severity ตามคำจำกัดความของ ISTQB [3] คือ ระดับของผลกระทบที่ข้อบกพร่องมีต่อการพัฒนาหรือการทำงานของส่วนประกอบหรือระบบ

ความรุนแรงของข้อบกพร่อง หรือที่เรียกว่า Defect Severity เป็นการจำแนกประเภทของข้อบกพร่องของซอฟต์แวร์ (ข้อบกพร่อง) เพื่อระบุระดับของผลกระทบด้านลบต่อคุณภาพของซอฟต์แวร์ คำศัพท์ และความหมายอาจแตกต่างกันไปตามบุคคล โครงการ องค์กร หรือเครื่องมือติดตามข้อบกพร่อง แต่ในระดับทั่วไป มีระดับความรุนแรงดังต่อไปนี้ [4]

1. Critical / Show Stopper (S1): ข้อบกพร่องที่ขัดขวางหรือขัดขวางการทดสอบผลิตภัณฑ์/คุณลักษณะโดยสิ้นเชิงถือเป็นข้อบกพร่องที่สำคัญ ตัวอย่างจะเป็นกรณีของการทดสอบ UI ซึ่งหลังจากผ่านตัวช่วยสร้างแล้ว UI จะค้างอยู่ที่หน้าจอต่างเดียวหรือไม่ดำเนินการต่อไปเพื่อทริกเกอร์ฟังก์ชัน หรือในบางกรณีเมื่อพีเจอร์ที่พัฒนาขึ้นเองนั้นหายไปจากสร้าง

2. Major or Severe (สำคัญหรือรุนแรง) (S2): ข้อบกพร่องที่สำคัญเกิดขึ้นเมื่อฟังก์ชันทำงานผิดพลาดจากความคาดหวัง หรือไม่ทำในสิ่งที่ควรทำ ตัวอย่างเช่น ต้องการปรับใช้ VLAN บนสวิตช์ และกำลังใช้เทมเพลต UI ที่ทริกเกอร์ฟังก์ชันนี้ เมื่อเทมเพลตนี้กำหนดค่า VLAN ล้มเหลวบนสวิตช์ จะถูกจัดประเภทเป็นข้อเสียเปรียบของการทำงานที่รุนแรง

3. Moderate/ Normal (ปานกลาง/ปกติ) (S3): ข้อบกพร่องปานกลางเกิดขึ้นเมื่อผลิตภัณฑ์หรือแอปพลิเคชันไม่ตรงตามเกณฑ์ที่กำหนดหรือยังคงแสดงพฤติกรรมที่ผิดธรรมชาติ อย่างไรก็ตาม ฟังก์ชันการทำงานโดยรวมจะไม่ได้รับผลกระทบ ตัวอย่างเช่น ในการปรับใช้เทมเพลต VLAN ด้านบน ข้อบกพร่องระดับปานกลางหรือปกติจะเกิดขึ้นเมื่อเทมเพลตถูกปรับใช้สำเร็จบนสวิตช์ แต่ไม่มีข้อบ่งชี้ที่ส่งไปยังผู้ใช้

4. Low or Minor (ต่ำหรือเล็กน้อย) (S4): ข้อบกพร่องเล็กน้อยเกิดขึ้นเมื่อแทบไม่มีผลกระทบต่อการทำงาน แต่ยังคงเป็นข้อบกพร่องที่ถูกต้องที่ควรแก้ไข ตัวอย่างนี้อาจรวมถึงการสะกดผิดในข้อความแสดงข้อผิดพลาดที่พิมพ์ไปยังผู้ใช้หรือข้อบกพร่องเพื่อปรับปรุงรูปลักษณ์ของคุณลักษณะ

2.1.2 Bug Report

ตามนิยาม ISTQB® (International Software Testing Qualifications Board) [5] ให้คำนิยามของ Bug Report ไว้ว่า คือ เอกสารที่ใช้ในการสื่อสาร หรือบันทึกเรื่องราว อาการ สถานะ รายละเอียดของ Defect ที่เกิดขึ้น

| Section No. | Heading | Details |
|-------------|---------------------------------|--|
| 1 | Test incident Report identifier | The unique identifier assigned to this incident report. |
| 2 | Summary | A summary of the incident, detailing where expected an actual result differ, identifying at high level the items that are affected, and the step leading up to the recognition of the incident, e.g. if in test execution, which test was executed and the actual keystrokes and data loaded. |
| 3 | Incident Description | A detailed description of the incident, which should include <ul style="list-style-type: none"> • Inputs • Expected Result • Actual Result • Anomalies • Date and time • Procedure Step • Environment • Attempts to repeat • Tester's Comment • Observer's Comment |
| 4 | Impact | Should include any information regarding possible causes and solutions If known, document what impact the incident has on progress. |

ภาพประกอบที่ 2.1 Format ISTQB®

Bug Report ในแบบฉบับของ ISTQB® (International Software Testing Qualifications Board) จะประกอบไปด้วย 4 ส่วนคือ

1. Test incident report identifier — คือ ID ของ Test Incident Report
2. Summary — เป็นการสรุปว่า ใคร ทำอะไร และ เกิดอะไรขึ้น ผลลัพธ์นั้นต่างจากสิ่งที่คาดหวังอย่างไร
3. Incident Description — เป็นการให้รายละเอียดที่ควรมี เช่น Input , Expected Result, Actual, Result, Procedure Step (Test Step)
4. Impact — ส่วนนี้เป็นการประเมินว่า Incident / Bug ที่เกิดขึ้นนั้นส่งผลกระทบต่อระบบอย่างไร

| | |
|--------------------|---|
| Title | |
| Description | <p>A detailed description of the incident, which should include</p> <p>Report Date</p> <p>Environment Details</p> <ul style="list-style-type: none"> • Device: iPhone , Samsung (if it is Mobile Device) • Desktop: Windows or Mac (If it Desktop) • OS Version: Windows10 , Mac Mojave 10.14.6 • Mobile OS: iOS version 12.4.1 , Android version 8 • Browser Name and Browser Version: Chrome version 76 • URL: (if it is website testing) • App version (if it is mobile app testing) <p>Step to Reproduce</p> <p>Observed Result/Actual Result</p> <p>Expected Result</p> |
| Attachment | |

ภาพประกอบที่ 2.2 ตัวอย่าง Bug Report

Title : ส่วนที่จะบอกได้ว่า เกิดอะไรขึ้น

Description : รายละเอียดที่จำเป็นต่อผู้ที่แก้ไขปัญหา

Report Date : แจ้งวันที่เท่าไร

Environment Details : รายละเอียดของ Environment ที่เจอปัญหา เพราะเครื่องของผู้ใช้งานระบบกับเครื่องของผู้ที่แก้ไขปัญหามันจะไม่เหมือนกัน

Procedure Step ใน ISTQB จะเรียกว่า Step to Reproduce ตัวอย่างเช่น

- Open Browser
- Go to “Test” Homepage
- Login to Test
- Select Restaurant
- Select favorite icon

Actual Result : สิ่งที่เกิดขึ้นเมื่อทำตามขั้นตอนข้างต้น ว่าเกิดอะไรขึ้น

Expected Result : สิ่งที่คุณคาดหวังตาม Requirement

Attachment : สิ่งแนบมา หรือเอกสารที่แนบมา

2.1.3 Software Testing

Software Testing หรือ การทดสอบซอฟต์แวร์ [6] เป็นการค้นหาข้อผิดพลาดที่เกิดขึ้น หรือ อาจจะเกิดขึ้น เพราะในการพัฒนาซอฟต์แวร์ หรือการพัฒนาโปรแกรมนั้นย่อมมีความผิดพลาดเกิดขึ้น แล้วจะต้องทำการแก้ไข ป้องกันข้อผิดพลาดดังกล่าว รวมถึงประเมินผลที่ได้จากการทำงานของซอฟต์แวร์ ว่าครบถ้วนและถูกต้องตามข้อกำหนดความต้องการหรือไม่

Software Testing เป็นขั้นตอนหลักของการพัฒนาซอฟต์แวร์ [7] เป็นกระบวนการ ค้นหาข้อผิดพลาดที่มีอยู่ในระบบ ช่วยให้ซอฟต์แวร์ที่ พัฒนาขึ้นมีความถูกต้อง สมบูรณ์ ปลอดภัย และมีประสิทธิภาพและคุณภาพที่ดี ผู้ที่ทำการทดสอบเรียกว่า ทีมทดสอบ (test team) โดยทั่วไปประกอบไปด้วย นักทดสอบมืออาชีพ นักวิเคราะห์ นักออกแบบระบบ ผู้เชี่ยวชาญการจัดการโครงสร้าง และผู้ใช้งาน การทดสอบ ซอฟต์แวร์ประกอบด้วยหลายขั้นตอน และแบ่งออกเป็นหลายระยะ ดังรายละเอียดต่อไปนี้

2.1.3.1 ขั้นตอนการทดสอบความผิดพลาดซอฟต์แวร์

ขั้นตอนการทดสอบความผิดพลาดของซอฟต์แวร์แบ่งเป็น 2 ขั้นตอนคือ การทดสอบระดับโปรแกรมและการทดสอบระดับระบบ ดังมีรายละเอียดดังนี้

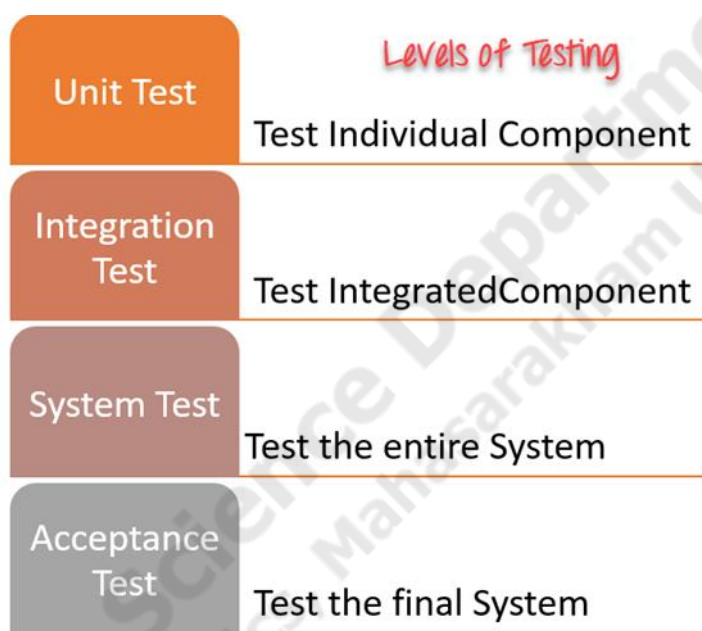
1. การทดสอบระดับโปรแกรม มีวัตถุประสงค์เพื่อ ตรวจสอบโปรแกรมที่ถูกสร้างขึ้นโดยโปรแกรมเมอร์เมื่อรวมกันเป็นระบบการทำงานแล้วสามารถทำงานได้ตามที่ออกแบบไว้ ความผิดพลาดที่อาจเกิดขึ้นในระดับนี้ ได้แก่ ความผิดพลาดของขั้นตอนวิธี (algorithm error) ความผิดพลาดด้านไวยากรณ์ (syntax error) ความผิดพลาดทางการคำนวณ (computation and precision error) ความผิดพลาดของเอกสาร (documentation error) ความผิดพลาดที่เกิดจากการปฏิบัติงานเกินสิ่งที่กำหนดไว้ (stress and overload error) ความผิดพลาดที่ระบบทำงานเกินประสิทธิภาพของ ระบบ (capacity or boundary error) ความผิดพลาด ของเวลา (timing or coordination error) ความผิดพลาด ในเรื่องประสิทธิภาพ (throughput หรือ performance error) ความผิดพลาดในการกู้คืน (recovery error) ความผิดพลาดในด้านฮาร์ดแวร์และซอฟต์แวร์ (hardware and system software error) ความผิดพลาดของมาตรฐาน และกระบวนการ (standard and procedure error) เป็นต้น

2. การทดสอบระดับระบบ มีวัตถุประสงค์เพื่อ ตรวจสอบว่าระบบที่พัฒนาขึ้นสามารถแก้ปัญหาได้ตรง ตามความต้องการที่ระบุในเอกสาร กำหนดความต้องการ ความผิดพลาดที่อาจเกิดขึ้นในระดับนี้ได้แก่ การวิเคราะห์ ความต้องการไม่ชัดเจน การแปลความหมายไม่ถูกต้อง การสื่อสารระหว่างทีมงานพัฒนาผิดพลาด การออกแบบระบบ และการออกแบบโปรแกรมผิดพลาด การพัฒนาโปรแกรม และเอกสารผิดพลาด ความผิดพลาดที่อาจเกิดขึ้นนี้ สามารถเกิดขึ้นจากลูกค้า นักออกแบบระบบ นักออกแบบ โปรแกรม โปรแกรมเมอร์ ทีมงานทดสอบ รวมทั้งทีมงาน ในการบำรุงรักษาระบบ

2.1.4 Level Software Testing

การทดสอบซอฟต์แวร์ส่วนใหญ่แบ่งออกได้เป็น 4 ระดับ [8]

1. Unit Testing: ตรวจสอบว่าส่วนประกอบซอฟต์แวร์มีคุณสมบัติตรงตามฟังก์ชันหรือไม่
2. Integration Testing: ตรวจสอบการไหลของข้อมูลจากโมดูลหนึ่งไปยังโมดูลอื่น
3. System Testing: ประเมินทั้งความต้องการใช้งาน และไม่ใช้หน้าที่สำหรับการทดสอบ
4. Acceptance Testing: ตรวจสอบข้อกำหนด หรือสัญญาตามการส่งมอบ



ภาพประกอบที่ 2.3 Levels of Testing

ที่มา : <https://www.guru99.com/levels-of-testing.html>

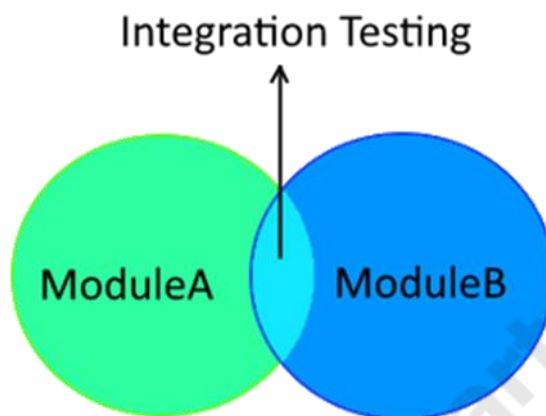
2.1.4.1 Module Testing หรือ Unit Testing [9]

เป็นการทดสอบหลังจากที่ผู้พัฒนาโปรแกรมได้เขียนโปรแกรมตามที่ได้รับมอบหมายเสร็จสิ้น โดยจะตรวจสอบหาข้อบกพร่องและโครงสร้างของโปรแกรมว่าทำงานได้ถูกต้องหรือไม่ การเขียน Unit Test เป็นวิธีการหนึ่งที่จะทำให้มั่นใจได้ว่า ส่วนของโปรแกรมที่เล็กๆ นั้นมันทำงานได้ถูกต้อง และเมื่อเกิดปัญหาจะได้มีการปรับปรุงแก้ไข Code ให้มีคุณภาพยิ่งขึ้น มีโปรแกรมเมอร์ และผู้พัฒนาระบบ เป็นผู้ทดสอบ

2.1.4.2 Integration Testing

เป็นการทดสอบการทำงานของ Module โปรแกรมทั้งหมด โดยทำการทดสอบในแต่ละ Function/Module ว่าสามารถทำงานได้หรือไม่ มีผลกระทบอะไรหรือไม่ ในส่วนนี้จะใช้วิธีการ

ทดสอบที่หลายหลายวิธีในการทดสอบ การ Test ระดับนี้จะแสดงให้เห็นบุคคลอื่นที่ไม่ใช่โปรแกรมเมอร์ได้ เห็นว่าระบบทั้งหมดสามารถทำงานร่วมกันได้ มี Tester เป็นผู้ทดสอบ

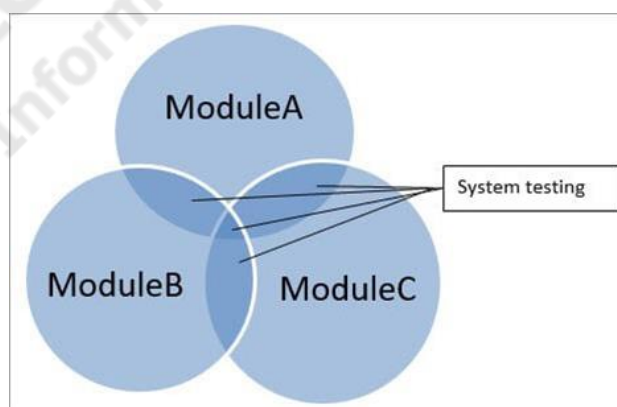


ภาพประกอบที่ 2.4 Integration Testing

ที่มา : <http://wow.in.th/rYsy>

2.1.4.3 System testing or System integration testing (SIT)

System Testing เป็นการทดสอบระบบหรือโปรแกรมโดยดูภาพรวมของการทำงาน ว่ามีการตอบสนองความต้องการทั้งในส่วนของฟังก์ชันการทำงานและประสิทธิภาพการทำงาน ว่าสอดคล้องกับความต้องการของลูกค้า (requirement specification) หรือไม่ มี Tester เป็นผู้ทดสอบ และในระดับนี้เองทางทีมผู้พัฒนาได้นำมาใช้ในการพัฒนาระบบ



ภาพประกอบที่ 2.5 System testing or System integration testing (SIT)

ที่มา : <http://wow.in.th/youOC>

2.1.4.4 User Acceptance Testing

เป็นกระบวนการทดสอบระบบก่อนใช้งานจริง เปรียบเทียบประสิทธิภาพของระบบกับความต้องการของลูกค้า โดยอ้างอิงจาก Requirement ที่ได้ ว่าตรงตาม Business Flow จริงๆ ของลูกค้า ขั้นตอนนี้ ผู้ใช้งานจริงๆ จะเข้ามามีส่วนร่วมในการทดสอบระบบก่อนที่จะใช้งานจริง ว่ามีส่วนไหนไม่เหมาะสมกับความต้องการหรือไม่ และทดสอบ Environment ที่ใกล้เคียงกับ Production มากที่สุด มี User , Tester (บางองค์กร Tester Test พร้อมลูกค้าไปด้วยเลย) เป็นผู้ทดสอบระบบ



ภาพประกอบที่ 2.6 User Acceptance Testing

ที่มา : <http://wow.in.th/rc3b>

2.1.5 Software Development

Software Development หรือการพัฒนาซอฟต์แวร์ [10] เป็นกระบวนการก่อสร้าง กำหนด ออกแบบ เขียนโปรแกรม จัดทำคู่มือ ทดสอบ และแก้ไขบั๊กที่เกี่ยวข้องกับการสร้าง และการปรับปรุง แอปพลิเคชัน เฟรมเวิร์ค หรือส่วนประกอบซอฟต์แวร์อื่น ๆ อีกทั้งการพัฒนาซอฟต์แวร์ยังเป็นกระบวนการเขียนและปรับปรุงซอร์ซโค้ด แต่ในแง่มุมที่กว้างขึ้นนั้นการพัฒนาซอฟต์แวร์รวมทุกอย่างที่เกี่ยวข้องระหว่างแนวคิดของซอฟต์แวร์ที่ต้องการตลอดไปจนถึงการเสร็จสมบูรณ์ของซอฟต์แวร์ โดยในบางครั้งอาจอยู่ในกระบวนการวางแผนและวางโครงสร้าง ดังนั้นการพัฒนาซอฟต์แวร์จึงเกี่ยวข้องกับการค้นคว้า การพัฒนาใหม่ การสร้างต้นแบบ การแก้ไข การใช้ซ้ำ การออกแบบและสร้างอีกครั้ง การปรับปรุง หรือการกระทำใด ๆ ก็ตามที่เกี่ยวข้องให้เกิดผลิตภัณฑ์ซอฟต์แวร์

2.1.6 Web Services

Web Services [11] คือระบบซอฟต์แวร์ที่ออกแบบมา เพื่อสนับสนุนการแลกเปลี่ยนข้อมูลระหว่างเครื่องคอมพิวเตอร์ผ่านระบบเครือข่าย โดยที่ภาษาที่ใช้ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ คือ XML เว็บเซอร์วิสมีอินเทอร์เน็ตเฟส ที่ใช้อธิบายรูปแบบข้อมูลที่เครื่องคอมพิวเตอร์ประมวลผลได้ ลักษณะการให้บริการของ Web Services นั้น จะถูกเรียกใช้งานจาก application อื่นๆ ในรูปแบบ RPC (Remote Procedure Call) ซึ่งการให้บริการจะมีเอกสารที่อธิบายคุณสมบัติของบริการกำกับไว้ โดยภาษาที่ถูกใช้เพื่อสื่อในการแลกเปลี่ยนคือ XML ทำให้เราสามารถเรียกใช้ Component ใด ๆ ก็ได้ ใน ระบบ หรือ Platform ใด ๆ ก็ได้ บน Protocol HTTP ซึ่งเป็น Protocol สำหรับ World Wide Web หรืออินเทอร์เน็ต อันเป็นช่องทางที่ได้รับการยอมรับทั่วโลกในการติดต่อสื่อสารกันระหว่าง Application กับ Application ในปัจจุบัน

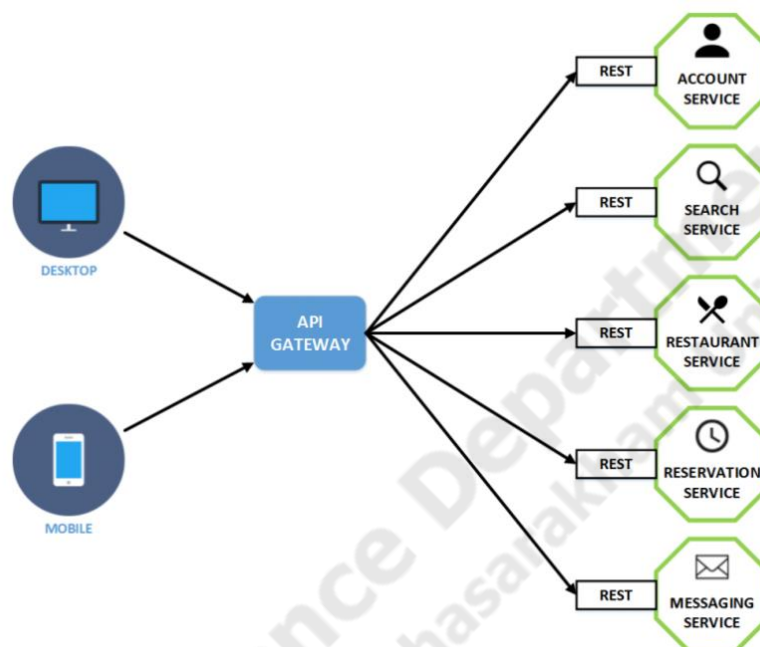
ประโยชน์ของ Web Services

1. Web Services ช่วยให้การเข้าถึงข้อมูลสารสนเทศจากแอปพลิเคชันที่ต่างกันเป็นไปได้โดยง่าย โดยแอปพลิเคชันนั้น ๆ สามารถเขียนด้วย Java และรันอยู่บน Sun Solaris Application Server หรืออาจจะเขียนด้วย C++ และรันอยู่บน Windows NT หรืออาจจะเขียนด้วย Perl และรันอยู่บนเครื่อง Linux ซึ่งมาตรฐานของ Web Service ทำให้อินเทอร์เน็ตเฟสของแอปพลิเคชันเหล่านี้ ถูกอธิบายโดย WSDL และทำให้อยู่ในมาตรฐานของ UDDI หลังจากนั้น จึงสามารถติดต่อสื่อสารถึงกันโดย XML ผ่าน SOAP อินเทอร์เน็ตเฟส
2. Web Services สามารถถูกเรียกใช้ภายในองค์กรเองหรือจากภายนอกองค์กร โดยผ่านไฟร์วอลล์ ดังนั้นจึงมีองค์กรใหญ่ๆ มากมาย กำลังพัฒนาระบบที่มีอยู่ให้เข้ากับ Web Services เนื่องจาก Web Services สามารถเพิ่มศักยภาพในการทำงานขององค์กร อีกทั้งลดค่าใช้จ่ายในการจัดการทรัพยากรขององค์กรได้อีกทางหนึ่ง
3. นอกจากนี้ Web Services ยังสามารถใช้ร่วมกับ Web Application โดยส่งผ่านข้อมูลทางอินเทอร์เน็ตได้อีกด้วยซึ่งนับเป็นวิธีที่มีประสิทธิภาพในการติดต่อสื่อสารกับลูกค้าหรือหุ้นส่วน ถึงแม้จะต้องคำนึงถึงระบบรักษาความปลอดภัย และการจัดการรายการของข้อมูลอยู่ก็ตาม แต่ Web Services ได้ใช้มาตรฐานทั่วไปของ internet เรื่องดังกล่าวจึงนับเป็นเรื่องธรรมดาของการสื่อสารผ่านระบบอิเล็กทรอนิกส์

2.1.6.1 Rest – Representational state transfer

Rest – Representational state transfer [12] เป็นรูปแบบสถาปัตยกรรมซอฟต์แวร์ (architecture) ที่ใช้ประโยชน์จากเทคโนโลยี Web Protocol เพื่อใช้ในการสร้าง Web Service โดยอาศัยรูปแบบของ HTTP Method เช่น GET POST PUT DELETE ในการทำงาน และจะส่งค่ากลับมา

เป็น json หรือ XML ส่งผลให้สามารถรับส่งข้อมูลไปมาข้าม platform ได้อย่างสะดวก เพราะเป็นการเรียกผ่าน HTTP Protocol ที่ใช้ในการเรียกใช้เว็บไซต์ การทำงานของ rest นั้นจะอาศัย URI/URL ของ request เพื่อเป็นตัวค้นหาและประมวลผลแล้วตอบกลับโดย response



ภาพประกอบที่ 2.7 Representational state transfer

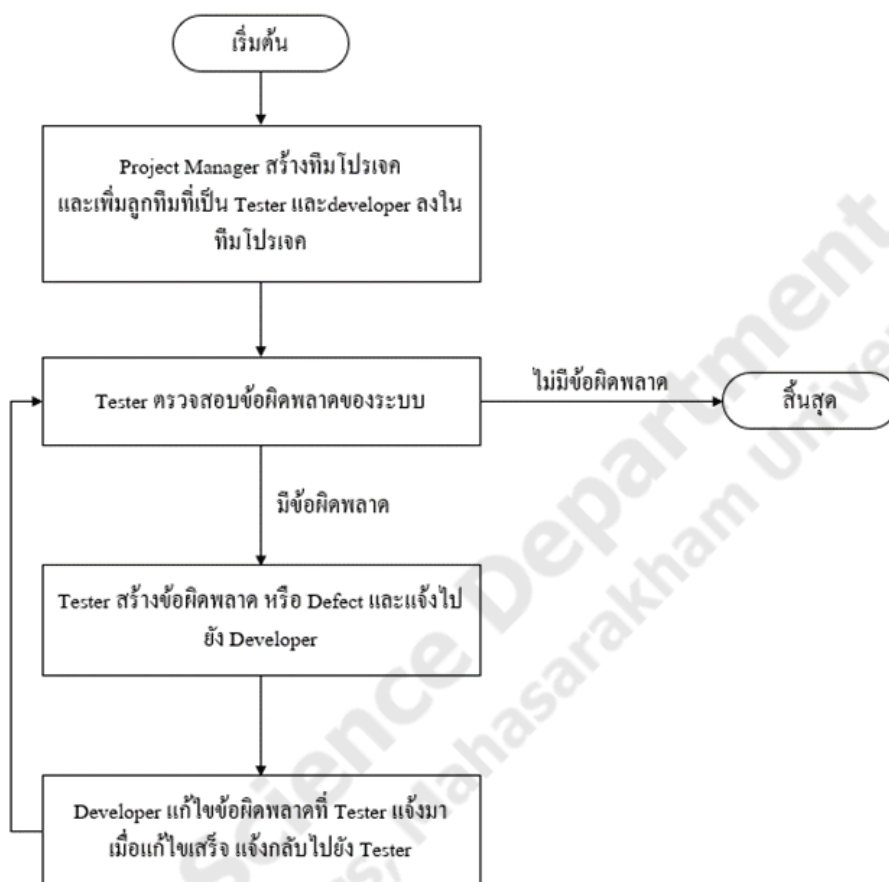
ที่มา : <https://www.4x-treme.com/restful-api-คืออะไร/>

2.1.6.2 RESTful – RESTful Web Service(RWS)

RESTful – RESTful Web Service(RWS) คือ Web Service ที่ใช้สถาปัตยกรรม Rest ซึ่งเจ้าตัว RWS อนุญาตให้ระบบ Request และเข้าถึง Resource บนเว็บโดยใช้ชุดคำสั่งที่กำหนดเอาไว้ล่วงหน้า โดยที่การโต้ตอบของระบบที่ใช้ REST จะอยู่บนพื้นฐานของ Hypertext Transfer Protocol (HTTP). Request จะส่งคำขอไปยัง URI ที่กำหนด และเอา response กลับมาเป็น Payload ในแบบ HTML, XML, JSON หรือ format อื่น ๆ โดย RESTful จะประกอบไปด้วย

- Client – ผู้ที่เข้ามาเป็น Request Resource
- Server – ผู้ที่ให้บริการ Resource

2.2 หลักการทำงาน



ภาพประกอบที่ 2.8 หลักการทำงานของระบบ

จากภาพประกอบที่ 2.8 อธิบายหลักการทำงานของระบบได้ดังนี้

Project Manager จะทำการสร้างทีมโปรเจกต์ และทำการเพิ่มลูกทีมเข้ามาในทีมโปรเจกต์ โดยลูกทีมจะแบ่งออกเป็น 2 ฝ่าย คือ Tester และ Developer

Tester จะทำการตรวจสอบหาข้อผิดพลาดของระบบ เมื่อเจอข้อผิดพลาด Tester จะทำการสร้างข้อผิดพลาด และแจ้งไปยัง Developer ให้แก้ไขข้อผิดพลาดนั้น

Developer จะทำการแก้ไขข้อผิดพลาดที่ได้รับจาก Tester เมื่อแก้ไขเสร็จก็จะแจ้งกลับไปยัง Tester ให้ทดสอบระบบอีกรอบ

หากไม่เจอข้อผิดพลาดแล้ว จะถือว่าเป็นการสิ้นสุดการทำงาน

2.3 สถาปัตยกรรมในการพัฒนา

2.3.1 Angular

Angular [13] เป็น Front-end JavaScript Framework ที่ทำงานบนฝั่ง Client ที่นำไปสร้าง Reactive Single Page Applications (SPA) ซึ่งก็คือทุกๆหน้าจะถูกโหลดมารวมอยู่ในหน้าเดียว การคลิกเปลี่ยนหน้าหรือการคลิกปุ่มต่างๆ จะทำให้รู้สึกเหมือนเป็น Desktop Application ที่ไม่มีการโหลดเปลี่ยนหน้า Angular เป็น Model-View-Controller (MVC) และยังเป็น Model-View-View Model (MVVM) อีกด้วย มีการเชื่อมการทำงานระหว่าง JavaScript เข้ากับ DOM Element ของ HTML ใช้การทำงาน client-side template สามารถสร้าง template ไปใส่ไว้ในที่กำหนดไว้ได้ และเป็น 2-way data binding เพื่อ sync Model กับ View

Angular คือ เฟรมเวิร์ค (framework) [14] สำหรับสร้างแอปพลิเคชันในฝั่งไคลเอนต์ในรูปแบบของ HTML, CSS และ JavaScript/TypeScript ซึ่ง TypeScript จะถูก compile ไปเป็น JavaScript

ประโยชน์เมื่อใช้ Angular

1. ทำให้ Application ที่มีโครงสร้างแบบ Clean Structure หรือ มีความง่ายต่อการ Maintain
2. มี re-usable code ต่างๆ มากมาย เช่น navigation หรือ browser history เป็นต้น
3. ทำให้ application มี test ที่หลากหลาย เช่น automatic test ซึ่งเป็นส่วนหนึ่งใน application

Angular Framework [15] นั้นมีโครงสร้างการทำงานในรูปแบบของคอมโพเนนต์ (Component) โดยแต่ละคอมโพเนนต์นั้นมีความเกี่ยวข้องซึ่งกันและกัน ในรูปแบบของ Parent และ Child ส่วนประกอบหลาย ๆ คอมโพเนนต์ถูกรวมกันจนเป็นหน้าเว็บไซต์ ซึ่งภายใต้การทำงานของแต่ละคอมโพเนนต์ถูกแบ่งออกเป็น 3 องค์ประกอบ คือ

1. ส่วนควบคุมการทำงาน – โดยใช้คำสั่ง TypeScript ในการควบคุม
2. ส่วนแสดงผล – เพื่อติดต่อกับผู้ใช้ โดยใช้ภาษา HTML
3. ส่วนจัดการรูปแบบ – เพื่อควบคุมการแสดงผลในรูปแบบของ CSS ไฟล์

โดยทั้ง 3 ส่วนจะทำงานสัมพันธ์กัน โดยสามารถแทรกคำสั่งหรือเรียกใช้ตัวแปรของภาษา TypeScript ภายใต้ Tag HTML ได้ ซึ่งเป็นสิ่งที่สะดวกมาก สำหรับการทำงานในกรณีที่ต้องมีการวนลูปเพื่อแสดงผลซ้ำ ๆ ในรูปแบบของการสร้าง Tag select หรือ การแสดงผลในรูปแบบตาราง นอกจากการทำงานร่วมกันภายในคอมโพเนนต์แล้ว ระหว่างคอมโพเนนต์ยังมีการทำงานระหว่างกันได้อีก เพื่อใช้สำหรับการส่งค่าหรือเรียกใช้ฟังก์ชันงานระหว่างกัน

2.4 ภาษาที่ใช้ในการพัฒนาระบบ

2.4.1 HTML

HTML [16] ย่อมาจาก Hypertext Markup Language เป็นภาษาคอมพิวเตอร์ที่ใช้สร้างหน้าเว็บ (Web Page) ในรูปแบบของไฟล์ HTML (คือไฟล์ที่มีนามสกุลเป็น .htm หรือ .html) ซึ่งมีเว็บเบราว์เซอร์ (Web Browser) เป็นโปรแกรมที่ใช้แปลงไฟล์ HTML เพื่อแสดงผลในรูปแบบของหน้าเว็บ

ไฟล์ HTML เป็นไฟล์รหัสแอสกี (ASCII) ถูกบันทึกในรูปแบบของไฟล์เอกสาร (Text File) ที่สามารถถูกสร้างจากโปรแกรมสร้างไฟล์ข้อความ (Text Editor) เช่น Notepad หรือ Word Processing ทั่ว ๆ ไป ซึ่งลักษณะของไฟล์ HTML ประกอบไปด้วยแท็ก (Tag) ต่างๆ ที่เป็นคำสั่งของ HTML ซึ่งแท็กจะอยู่ภายในเครื่องหมาย < และ >

แท็กใน HTML แบ่งเป็น 2 ประเภทคือคอนเทนเนอร์แท็ก (Container Tag) และแท็กเปล่า (Empty Tag) โดยที่คอนเทนเนอร์แท็ก ประกอบไปด้วยแท็กเปิด และแท็กปิด โดยที่แท็กปิดจะมีเครื่องหมาย/ นำหน้าแท็ก เช่น <H1> . . . </H1> ส่วนแท็กเปล่า จะมีแท็กเปิดอย่างเดียว เช่น <HR> ซึ่งแท็กจะถูกเขียนด้วยตัวอักษรพิมพ์ใหญ่หรือพิมพ์เล็กก็ได้จะไม่มีผลต่อการแสดงผลของเว็บเบราว์เซอร์ เช่น
,
,
 หรือ
 เว็บเบราว์เซอร์จะแปลความหมายเหมือนกัน

โครงสร้างไฟล์ HTML แบ่งออกเป็น 2 ส่วน คือ ส่วนหัวเรื่อง (Head Section) และส่วนเนื้อหา (Body Section) โดยจะมีแท็ก <HTML> และ </HTML> เป็นตัวกำหนดขอบเขตไฟล์ซึ่งส่วนหัวเรื่อง มีไว้กำหนดข้อมูลเฉพาะของหน้าเว็บ เช่น ชื่อเรื่องของเว็บภายในแท็ก <HEAD> และ </HEAD> และสำหรับส่วนเนื้อหามีไว้กำหนดรายละเอียดต่างๆ ที่ต้องการแสดงบนหน้าเว็บ เช่น ข้อความ และรูปภาพภายในแท็ก <BODY> และ </BODY>

2.4.2 CSS

CSS ย่อมาจาก Cascading Style Sheet [17] มักเรียกโดยย่อว่า "สไตลชีต" คือภาษาที่ใช้เป็นส่วนของการจัดรูปแบบการแสดงผลเอกสาร HTML โดยที่ CSS กำหนดกฎเกณฑ์ในการระบุรูปแบบ (หรือ "Style") ของเนื้อหาในเอกสาร อันได้แก่ สีของข้อความ สีพื้นหลัง ประเภทตัวอักษร และการจัดวางข้อความ ซึ่งการกำหนดรูปแบบ หรือ Style นี้ใช้หลักการของการแยกเนื้อหาเอกสาร HTML ออกจากคำสั่งที่ใช้ในการจัดรูปแบบการแสดงผล กำหนดให้รูปแบบของการแสดงผลเอกสาร ไม่ขึ้นอยู่กับเนื้อหาของเอกสาร เพื่อให้ง่ายต่อการจัดรูปแบบการแสดงผลลัพท์ของเอกสาร HTML โดยเฉพาะในกรณีที่มีการเปลี่ยนแปลงเนื้อหาเอกสารบ่อยครั้ง หรือต้องการควบคุมให้รูปแบบการแสดงผลเอกสาร HTML มีลักษณะของความสม่ำเสมอทั่วกันทุกหน้าเอกสารภายในเว็บไซต์เดียวกัน โดยกฎเกณฑ์ในการกำหนดรูปแบบ (Style) เอกสาร HTML ถูกเพิ่มเข้ามาครั้งแรกใน HTML 4.0 เมื่อปีพ.ศ. 2539 ในรูปแบบของ CSS level 1 Recommendations ที่กำหนดโดย องค์กร World Wide Web Consortium หรือ W3C

CSS มีประโยชน์อย่างไร

ภาษา CSS (Cascading Style Sheets) มีประโยชน์หลายอย่าง ซึ่งทำให้การพัฒนาเว็บเพจด้วยภาษา HTML เป็นเรื่องที่ย่างมากขึ้น

1. ภาษา CSS จะช่วยในการจัดรูปแบบแสดงผลให้กับภาษา HTML ซึ่งจะช่วยลดการใช้ภาษา HTML ให้น้อยลง โดยเหลือเพียงแต่ส่วนที่เป็นเอกสารที่เป็นภาษา HTML เท่านั้นทำให้มีการแก้ไขและทำความเข้าใจได้ง่ายขึ้น
2. ทำให้ขนาดไฟล์ HTML น้อยลงเนื่องจาก ภาษา CSS จะช่วยการใช้ภาษา HTML ทำให้ขนาดไฟล์นั้นก็เล็กลงไปด้วยเช่นกัน
3. ภาษา CSS เป็นภาษา Style Sheets โดย Style Sheets ชุดเดียวสามารถใช้กำหนดรูปแบบการแสดงผลให้เอกสาร HTML ทั้งหน้า หรือทุกหน้ามีผลเหมือนกันได้ จึงทำให้เวลาที่มีการแก้ไขก็จะแก้ไขได้ง่ายขึ้นเพียงแก้ไข Style Sheets ที่ใช้งานเพียงชุดเดียวเท่านั้น
4. ทำให้เว็บไซต์มีมาตรฐาน เพราะการใช้งาน CSS นั้นจะทำให้การแสดงผลในสื่อต่าง ๆ ถูกปรับเปลี่ยนไปได้เหมาะสม เช่น การแสดงผลบนหน้าจอ และการแสดงผลในมือถือ
5. CSS สามารถที่จะใช้งานได้หลากหลาย เบ็บบราวเซอร์ ทำให้การใช้งานนั้นสะดวกมากยิ่งขึ้น
6. CSS สามารถกำหนดแยกไว้ต่างหากจากไฟล์เอกสาร HTML และสามารถนำมาใช้ร่วมกับเอกสารหลายไฟล์ได้ การแก้ไขก็แก้เพียง จุดเดียวก็มีผลกับเอกสารทั้งหมด

2.4.3 TypeScript

TypeScript [18] เป็นภาษาโปรแกรมที่รวมความสามารถที่ ES2015 เองมีอยู่ สิ่งที่เพิ่มขึ้นมาคือสนับสนุน Type System รวมถึงคุณสมบัติอื่นๆที่เพิ่มมากขึ้น เช่น Enum และความสามารถที่เพิ่มขึ้นมาของการโปรแกรมเชิงวัตถุ TypeScript นั้นเป็น transpiler เหมือน Babel นั้นหมายความว่า ตัวแปลภาษาของ TypeScript จะแปลโค้ดที่เราเขียนให้เป็น JavaScript อีกทีหนึ่ง ผลลัพธ์สุดท้ายจะสามารถใช้งานได้บนเว็บเบราว์เซอร์ทั่วไป

ข้อดีของการใช้ TypeScript

1. TypeScript ทำให้ใช้ JavaScript สมัยใหม่ได้ในปัจจุบัน ความสามารถของ ES2015 และอื่นๆ ได้รวมไว้ใน TypeScript
2. ตัวแปรที่ประกาศแล้วใน TypeScript จะเปลี่ยนชนิดข้อมูลไม่ได้อีกต่อไป ข้อผิดพลาดในโปรแกรมจะน้อยลงเพราะไม่มีโอกาสพลาดในการใส่ข้อมูลผิดชนิด
3. TypeScript มีการตรวจสอบโค้ดในช่วง compile time ทำให้ดักจับข้อผิดพลาดได้ตั้งแต่นั้น ไม่ปล่อยให้ข้อผิดพลาดไปอยู่ในตอนทำงานจริง (runtime)
4. IDE และ Text Editor ที่ดีสนับสนุนให้ใช้งาน TypeScript ได้อย่างสมบูรณ์

2.4.4 PHP

PHP (พีเอชพี) [19] ปัจจุบันย่อมาจากคำว่า PHP Hypertext Preprocessor ภาษา PHP เป็นโปรแกรมภาษาคอมพิวเตอร์ระดับสูง สามารถใช้งานกับระบบปฏิบัติการ (Operating Systems) ที่หลากหลาย เช่น Linux (HP-UX, Solaris, และ OpenBSD), Microsoft, macOS และสามารถใช้งานได้กับเว็บเซิร์ฟเวอร์ เช่น Apache, Microsoft Internet Information Services (IIS) ได้ นอกจากนั้นแล้ว PHP ยังสนับสนุนฐานข้อมูลรูปแบบต่างๆ ตั้งแต่ MySQL, PDO หรือ Open Database Connection ด้วยเหตุผลดังกล่าวจึงทำให้เลือกและออกแบบระบบที่ต้องการใช้งานได้ง่ายขึ้น

PHP เป็นภาษาสคริปต์ (Scripting Language) คำสั่งต่างๆ จะเก็บในรูปแบบของข้อความ (Text) อาจเขียนแทรกอยู่ในภาษา HTML หรือใช้งานอิสระก็ได้ แต่ในการใช้งานจริงมักใช้งานร่วมกับภาษา HTML ดังนั้นการเขียนโปรแกรมนี้ต้องมีความรู้ด้านภาษา HTML เป็นอย่างดี อย่างไรก็ตามเราสามารถใช้อุปกรณ์ช่วยอำนวยความสะดวกในการสร้างงานได้ เช่น Macromedia Dreamweaver หรือโปรแกรมประเภท Editor เช่น Edit Plus ฯลฯ โปรแกรมเหล่านี้จะช่วยจำแนกคำ เช่น คำสั่ง คำทั่วไป ตัวแปร ฯลฯ ให้มีสีต่างกันเพื่อสะดวกในการสังเกตและมีตัวเลขบอกบรรทัดทำให้สะดวกในการแก้ไข

PHP คือ ภาษาคอมพิวเตอร์ Server-Side Script ซึ่งใช้ในการจัดทำเว็บไซต์และสามารถประมวลผลออกมาในรูปแบบ HTML โดยมีรากฐานโครงสร้างคำสั่งมาจากภาษา ภาษาซี ภาษาจาวา และ ภาษาเพิร์ล เป้าหมายหลักของภาษา PHP คือให้นักพัฒนาเว็บไซต์สามารถเขียนเว็บเพจ ที่มีความตอบโต้ได้อย่างรวดเร็ว

PHP สามารถทำงานในสิ่งที่โปรแกรม CGI [20] สามารถทำได้และทำได้มากกว่าด้วยยกตัวอย่างเช่น การเก็บข้อมูล การสร้างหน้าเนื้อหาที่เป็นลักษณะไดนามิก การส่งและรับคุกกี้ เป็นต้น

สคริปต์ของ PHP จะถูกใช้งานใน 3 รูปแบบใหญ่ ๆ ได้แก่

1. สคริปต์ PHP ที่เซิร์ฟเวอร์ (Server-side scripting) เป็นลักษณะการเขียนโค้ดรูปแบบการใช้งานดั้งเดิมและนิยมใช้กัน โดยจะต้องมีส่วนประกอบ 3 ส่วนได้แก่ PHP parser, เว็บเซิร์ฟเวอร์ และเว็บ เบรราวเซอร์ โค้ดจะทำการประมวลผลที่เซิร์ฟเวอร์ และแสดงที่เครื่องคอมพิวเตอร์ของตนเอง
2. สคริปต์แบบคอมมานด์ไลน์ (Command line scripting) เป็นลักษณะการเขียนโค้ดที่ไม่จำเป็นต้องมีเซิร์ฟเวอร์หรือเบรราวเซอร์ในการใช้งาน ใช้เพียง PHP parser
3. การเขียนแอปพลิเคชันสำหรับคอมพิวเตอร์ (Writing desktop applications) เหมาะกับโปรแกรมเมอร์ที่มีความเชี่ยวชาญในการใช้งาน PHP และต้องการใช้งานพีเอชพี PHP ขั้นสูง

2.4.5 phpMyAdmin

phpMyAdmin [21] คือโปรแกรมที่ถูกพัฒนาโดยใช้ภาษา PHP เพื่อใช้ในการบริหารจัดการฐานข้อมูล MySQL แทนการคีย์คำสั่ง เนื่องจากถ้าจะใช้ฐานข้อมูลที่เป็น MySQL บางครั้งจะมีความลำบากและยุ่งยากในการใช้งาน ดังนั้นจึงมีเครื่องมือในการจัดการฐานข้อมูล MySQL ขึ้นมาเพื่อให้สามารถจัดการ ตัวDBMS ที่เป็น MySQL ได้ง่ายและสะดวกยิ่งขึ้น โดย phpMyAdmin ก็ถือเป็นเครื่องมือชนิดหนึ่งในการจัดการ

phpMyAdmin เป็นส่วนต่อประสานที่สร้างโดยภาษาพีเอชพี ซึ่งใช้จัดการฐานข้อมูล MySQL ผ่านเว็บเบราว์เซอร์ โดยสามารถที่จะทำการสร้างฐานข้อมูลใหม่ หรือทำการสร้าง TABLE ใหม่ๆ และยังมี function ที่ใช้สำหรับการทดสอบการ query ข้อมูลด้วยภาษา SQL พร้อมกันนั้น ยังสามารถทำการ insert delete update หรือแม้กระทั่งใช้ คำสั่งต่างๆ เหมือนกับกับการใช้ภาษา SQL ในการสร้างตารางข้อมูล

phpMyAdmin เป็นโปรแกรมประเภท MySQL Client ตัวหนึ่งที่ใช้ในการจัดการข้อมูล MySQL ผ่านweb browser ได้โดยตรง phpMyAdmin ตัวนี้จะทำงานบน Web server เป็น PHP Application ที่ใช้ควบคุมจัดการ MySQL Server ความสามารถของ phpMyAdmin คือ

1. สร้างและลบ Database
2. สร้างและจัดการ Table เช่น แทรก record, ลบ record, แก้ไข record, ลบ Table, แก้ไข field
3. โหลดเท็กซ์ไฟล์เข้าไปเก็บเป็นข้อมูลในตารางได้
4. หาผลสรุป (Query) ด้วยคำสั่ง SQL



ภาพประกอบที่ 2.9 ตัวอย่างการใช้งานโปรแกรม phpMyAdmin

ที่มา : <https://www.mindphp.com/คู่มือ/73-คืออะไร/2285-phpmyadmin-คืออะไร.html>

2.5 เครื่องมือที่ใช้ในการพัฒนาระบบ

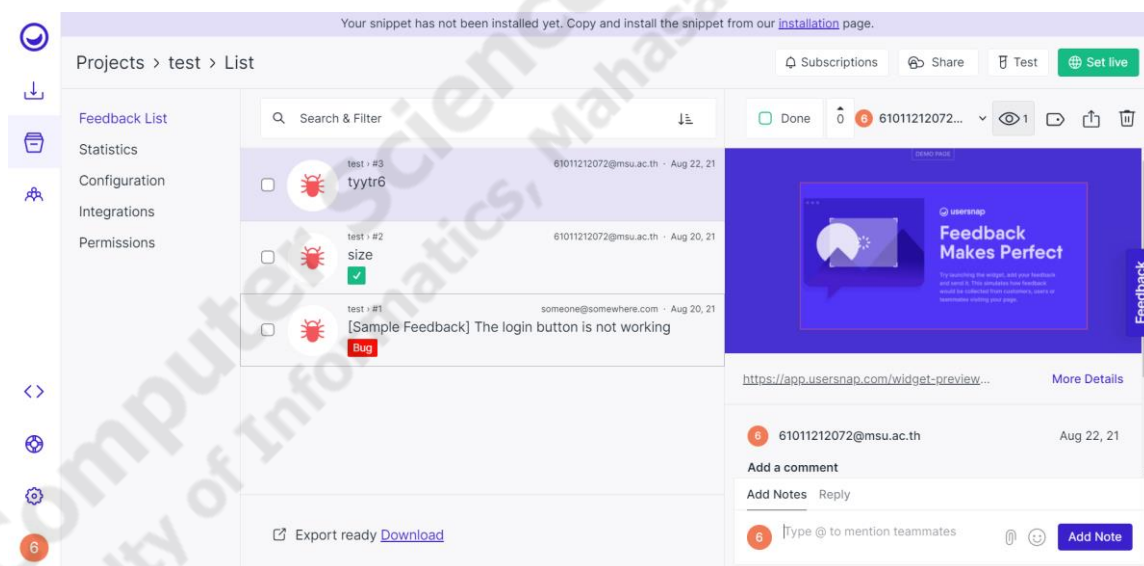
2.5.1 Visual Studio Code

Visual Studio Code หรือ VSCode [22] เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ด จากค่ายไมโครซอฟท์ มีการพัฒนาออกมาในรูปแบบของ Open Source จึงสามารถนำมาใช้งานได้แบบฟรี ๆ ที่ต้องการความเป็นมืออาชีพ

Visual Studio Code นั้นเหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์มรองรับการใช้งานทั้งบน Windows, macOS และ Linux สนับสนุนทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่าง ๆ ให้เลือกใช้อย่างมากมาย ไม่ว่าจะเป็น การเปิดใช้งานภาษาอื่น ๆ ทั้ง ภาษา C++, C#, Java, Python, PHP หรือ Go เป็นต้น

2.6 ระบบงานที่เกี่ยวข้อง

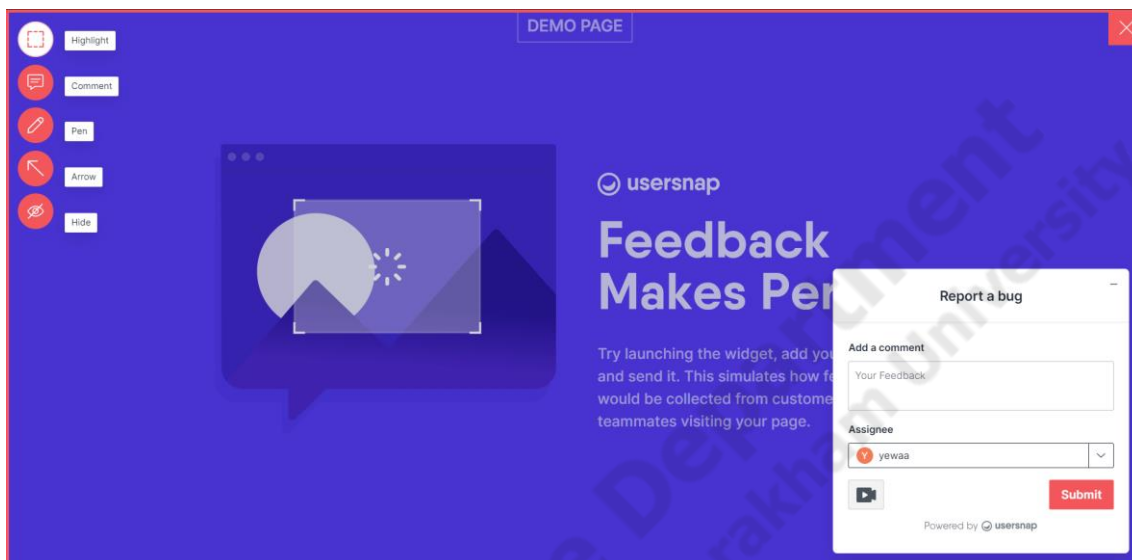
2.6.1 User snap



ภาพประกอบที่ 2.10 หน้าแรกหลังจากเข้าสู่ระบบของเว็บไซต์ usersnap

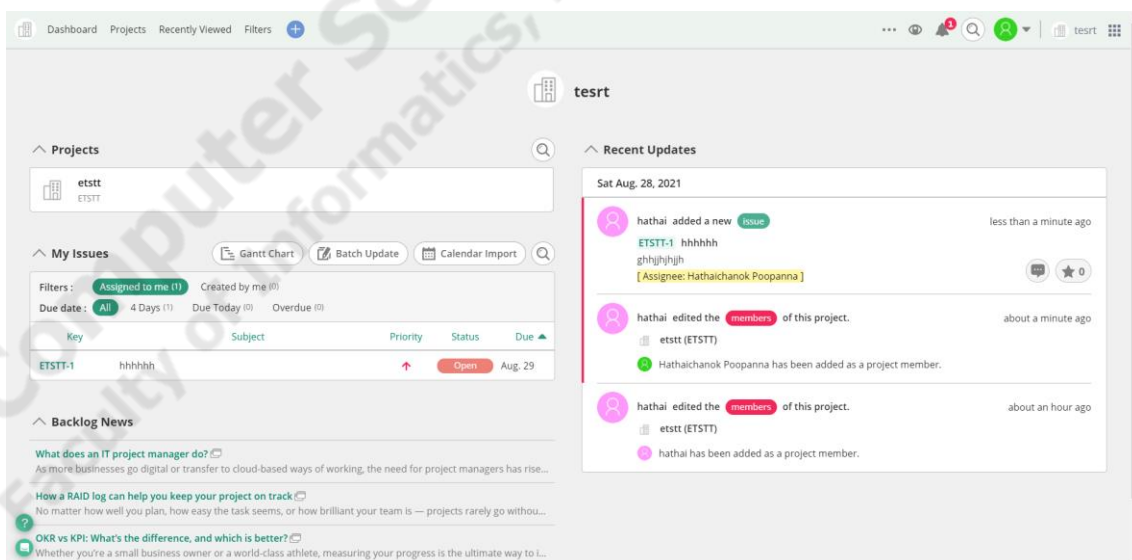
Usersnap [23] คือเว็บไซต์การรายงานจุดบกพร่อง สำหรับทีมักพัฒนา และนักออกแบบ หนึ่งในปัญหาการทำงานร่วมกันที่ใหญ่ที่สุดคือการรายงานจุดบกพร่อง เมื่อ Tester ต้องการรายงานปัญหา จะต้องเปิดแอปแยกต่างหาก และแจ้งให้นักพัฒนาทราบว่าจะเกิดอะไรขึ้น นี่เป็นสิ่งที่สะดวก และมีโอกาสที่ข้อความบางส่วนจะหายไป กระบวนการนี้ ไม่ต้องพูดถึงความจริงที่ว่านักพัฒนาซอฟต์แวร์ต้องหาจุดบกพร่องโดยใช้คำอธิบายจาก Tester

Usersnap ช่วยติดตามจุดบกพร่องได้โดยตรงจากเบราว์เซอร์ สามารถใส่คำอธิบายประกอบ ภาพหน้าจอและร้องขอการเปลี่ยนแปลงได้ทันที แทนการต้องกรอกแบบฟอร์ม และรายงาน ทำให้การติดตาม และการรายงานจุดบกพร่องรวดเร็ว และมีประสิทธิภาพมากกว่าที่เคย

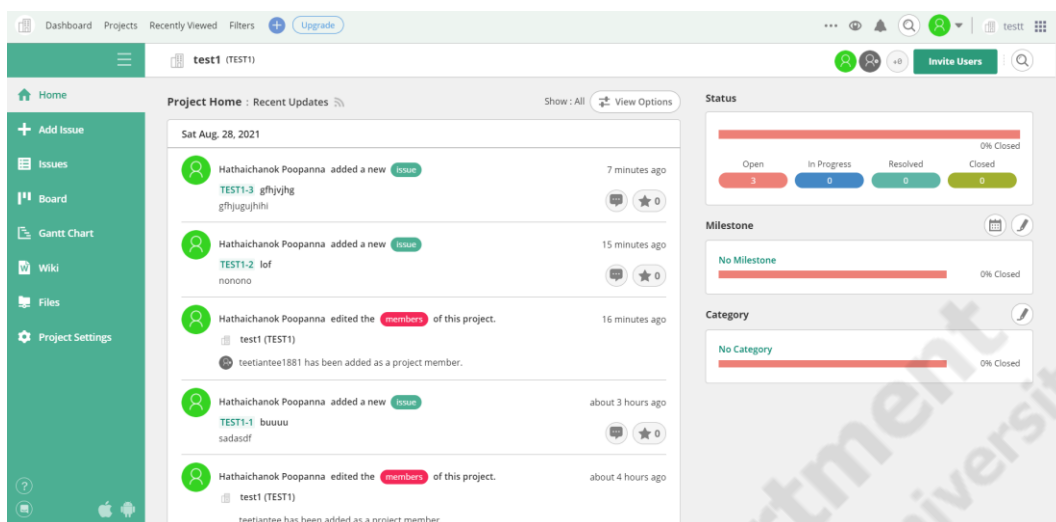


ภาพประกอบที่ 2.11 หน้าการเพิ่ม bug ของเว็บ usersnap

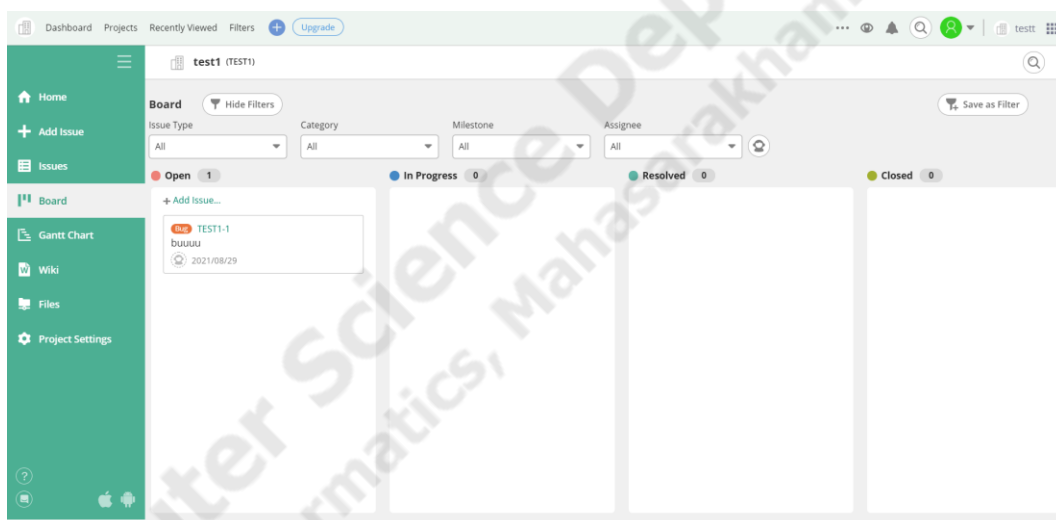
2.6.2 Backlog



ภาพประกอบที่ 2.12 หน้าแรกหลังจากเข้าสู่ระบบของเว็บไซต์ backlog



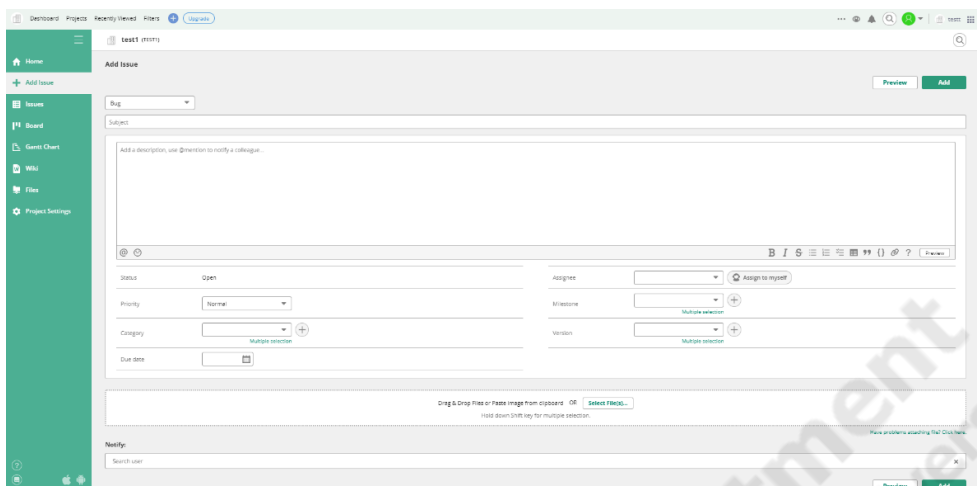
ภาพประกอบที่ 2.13 หน้าแรกหลังจากเข้าโปรเจกต์ของเว็บไซต์ backlog



ภาพประกอบที่ 2.14 หน้าบอร์ดแสดงการทำงานของ backlog

Backlog คือ ซอฟต์แวร์ติดตามจุดบกพร่อง และการจัดการโครงการออนไลน์ที่สร้างขึ้นสำหรับทีมพัฒนาเมื่อใช้ซอฟต์แวร์ติดตามจุดบกพร่อง ทำให้ทั้งทีมสามารถทำงานร่วมกันเพื่อแก้ไขจุดบกพร่อง โดยการบันทึกเป็นปัญหา ปัญหาคือศูนย์กลางสำหรับการพูดคุย ดำเนินการ และแก้ไขจุดบกพร่องแต่ละข้อ โดยสมาชิกในทีมจะให้รายละเอียดข้อมูลที่สำคัญเกี่ยวกับการเปลี่ยนแปลงต่างๆ ทั้งหมดที่ทำได้ และติดตามว่าใครกำลังทำงานในส่วนนี้ และทำงานเมื่อใด

นอกจากนี้ เครื่องมือติดตามจุดบกพร่อง และปัญหาช่วยให้ทั้งทีมและผู้จัดการได้รับมุมมองในระดับสูงว่าทุกจุดบกพร่องในเวิร์กโฟลว์อยู่ที่ใด นอกจากนี้ สมาชิกในทีมทุกคนในปัจจุบันและในอนาคตสามารถเข้าใจประวัติซอฟต์แวร์ของงาน รวมทั้งจุดบกพร่อง และการแก้ไขทั้งหมดด้วยคลังซอฟต์แวร์

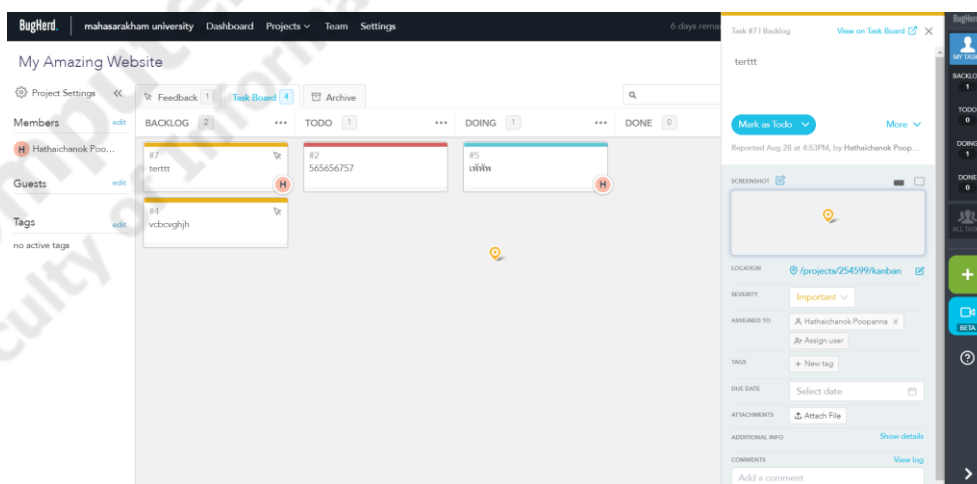


ภาพประกอบที่ 2.15 หน้าการเพิ่มปัญหา หรือข้อผิดพลาด ของเว็บ backlog

การบันทึกข้อมูล [24] ต้องบันทึกว่าใคร อะไร ที่ไหน เมื่อไร และเพราะเหตุใดของจุดบกพร่อง เป็นสิ่งสำคัญสำหรับการแก้ไขโดยไม่มีข้อผิดพลาดเพิ่มเติม แต่ละปัญหาใน Backlog มาพร้อมกับพื้นที่สำหรับชื่อเรื่อง คำอธิบาย ไฟล์แนบ และฟิลด์ที่กำหนดเอง เพื่อช่วยกำหนดจุดบกพร่อง

จัดลำดับความสำคัญ ข้อบกพร่องทั้งหมดต้องได้รับการจัดลำดับความสำคัญ และกำหนดตามความเร่งด่วนของจุดบกพร่องและปริมาณงานปัจจุบันของทีม ด้วย Backlog สามารถกำหนดลำดับความสำคัญสำหรับแต่ละจุดบกพร่อง มอบหมายให้กับทีม และเปลี่ยนวันที่ครบกำหนดตามความจำเป็นเพื่อรองรับโครงการที่เข้ามาและจุดบกพร่องใหม่

2.6.3 BugHerd

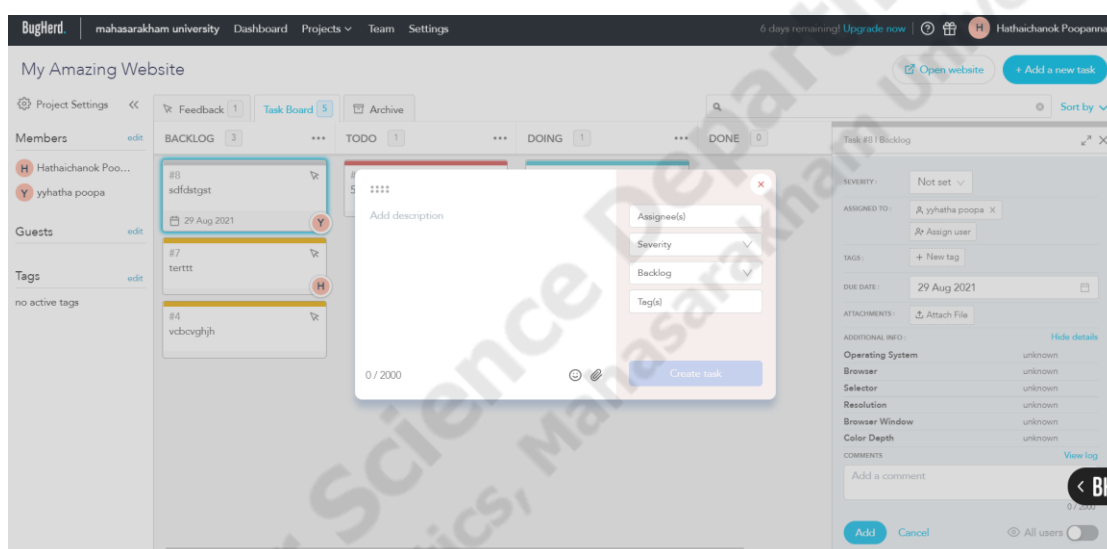


ภาพประกอบที่ 2.16 หน้าการทำงานหลักของเว็บไซต์ bugherd

BugHerd เป็นวิธีที่ง่ายที่สุดในการติดตามจุดบกพร่อง รวบรวมและจัดการข้อเสนอแนะสำหรับหน้าเว็บ ทีมงานและลูกค้า มีการปิกหมุดความคิดเห็นไว้ที่องค์ประกอบต่างๆ บนเว็บเพจ เพื่อระบุตำแหน่งปัญหาได้อย่างแม่นยำ

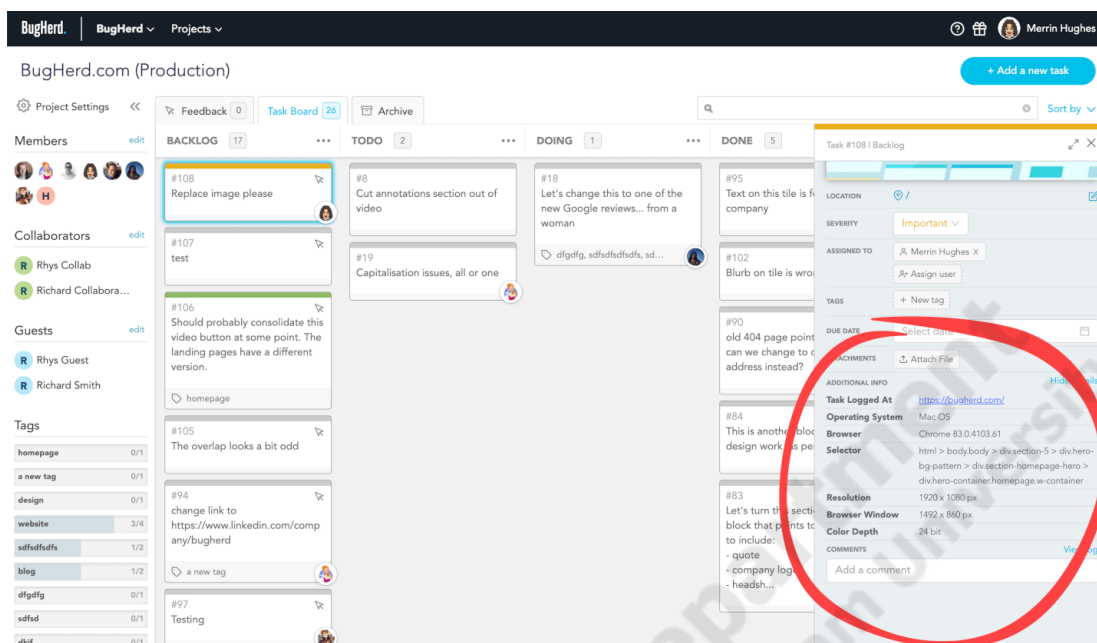
BugHerd ยังรวบรวมข้อมูลที่ต้องการเพื่อจำลองและแก้ไขข้อบกพร่องอย่างรวดเร็ว เช่น เบราวเซอร์ ข้อมูลตัวเลือก CSS ระบบปฏิบัติการ หรือภาพหน้าจอ

บักและข้อเสนอแนะ พร้อมกับข้อมูลทางเทคนิค จะถูกป้อนไปยังกระดานงานแบบ Kanban ซึ่งสามารถกำหนดและจัดการจุดบกพร่องได้จนจบ BugHerd ยังสามารถรวมเข้ากับเครื่องมือการจัดการโครงการที่มีอยู่ ช่วยให้ทีมมีความเข้าใจตรงกันพร้อมการแก้ไขจุดบกพร่อง



ภาพประกอบที่ 2.17 หน้าการเพิ่มงาน หรือข้อผิดพลาด ของ bugherd

BugHerd [25] สามารถแจ้งให้ทีมทราบถึงสิ่งที่ต้องทำได้อย่างง่ายดายด้วย Task Board ข้อเสนอแนะทั้งหมดจะถูกรวบรวม และจัดเรียงไว้ ดังนั้นจะไม่พลาดจุดบกพร่องหรือปัญหาใดๆ และสมาชิกในทีมรู้ว่าเมื่อไหร่แก้ไขแล้วและยังต้องทำอะไรอีก คุณสามารถจัดสรรงานให้กับสมาชิกในทีม แสดงความคิดเห็นเกี่ยวกับงานเพื่อชี้แจง และดูข้อมูลที่จำเป็นในการแก้ไขจุดบกพร่องที่อาจเกิดขึ้น เช่น ระบบปฏิบัติการหรือเบราว์เซอร์ที่โคลเอ็นต์ใช้

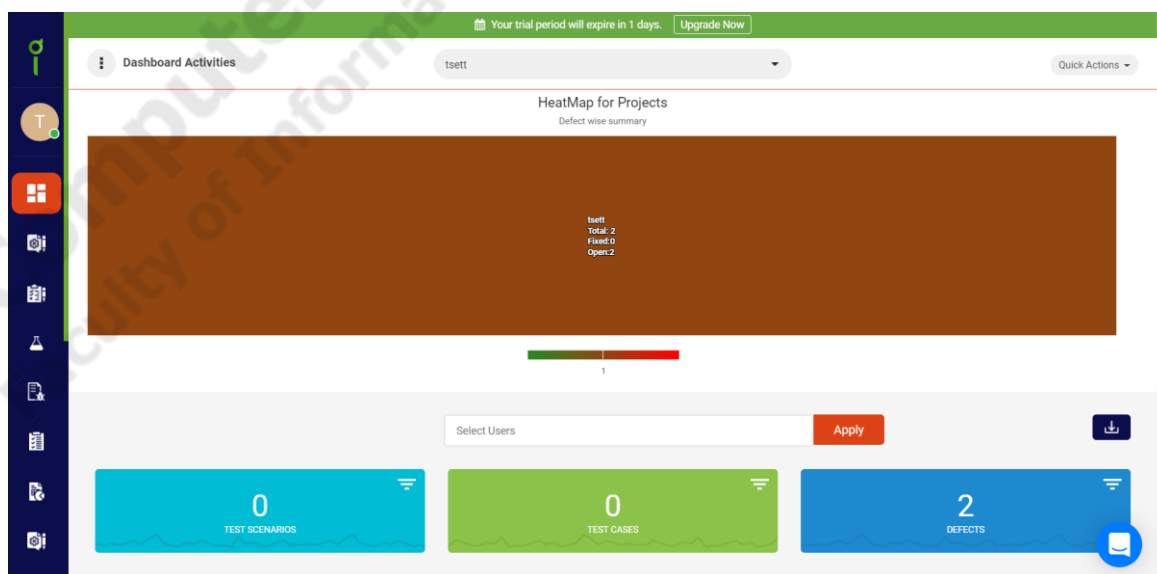


ภาพประกอบที่ 2.18 รายละเอียดของข้อผิดพลาด ของเว็บ bugherd

ที่มา : <https://www.affde.com/th/bugherd-overview.html>

การแก้ไขปัญหาและแก้ไขข้อผิดพลาด ด้วยข้อมูลที่ BugHerd รวบรวมมาให้ เมื่อสร้างงานคำติชม งานจะจับเบราว์เซอร์ ระบบปฏิบัติการ URL ของหน้าเว็บและภาพหน้าจอของหน้า และแนบไปกับงาน ไม่จำเป็นต้องถามลูกค้าเพื่อขอข้อมูลเพิ่มเติม

2.6.4 Kualitee



ภาพประกอบที่ 2.19 หน้าแรกหลังจากเข้าสู่ระบบของเว็บไซต์ kualitee

Kualitee [26] เป็นซอฟต์แวร์จัดการทดสอบที่ใช้โดย ผู้ทดสอบ นักพัฒนา และผู้ทำงานร่วมกันอื่น ๆ เพื่อจัดการความพยายามในการทดสอบตามข้อกำหนด มีแดชบอร์ดแบบไดนามิกที่ให้ข้อมูลเกี่ยวกับโครงการปัจจุบันและให้มุมมอง 360 องศาของโครงการ และสถิติ สมาชิกต้องเข้าสู่ระบบก่อนหลังจากนั้นจึงสามารถดูสถานะเข้าถึงสถานการณ์ทดสอบและกรณีทดสอบ และตรวจสอบรายงานข้อบกพร่องได้

ซอฟต์แวร์จัดทำโครงการที่ครอบคลุมโมดูลการจัดการที่ช่วยให้จัดการงานทั้งหมด นอกจากนี้ยังสามารถเชื่อมต่อโครงการกับข้อกำหนดที่กำหนด และให้สิทธิ์แก่สมาชิกในทีมในการอัปเดตข้อมูล และแก้ไขข้อบกพร่อง Kualitee นำเสนอองค์ประกอบการจัดการการทดสอบที่ไม่เหมือนใครซึ่งช่วยให้ผู้ทดสอบ / นักพัฒนาสามารถสร้างแผนงานการทดสอบจัดบันทึกสถานการณ์การทดสอบสร้างกรณีทดสอบเรียกใช้และดูผลลัพธ์ได้

กรณีการทดสอบทั้งหมดจะถูกบันทึกไว้ในซอฟต์แวร์ช่วยให้สามารถนำกลับมาใช้ใหม่ได้ในอนาคตโดยไม่ต้องเริ่มต้นใหม่ สามารถรวมเข้ากับแอปพลิเคชันที่โดดเด่นเช่น Asana, Selenium, GitLab, Redmine, Jira และอื่น ๆ ได้

ภาพประกอบที่ 2.20 หน้าการเพิ่มข้อผิดพลาด ของเว็บ kualitee

คุณสมบัติ Kualitee

- สร้าง กำหนด และติดตามข้อบกพร่อง
- การตรวจสอบย้อนกลับระหว่างข้อบกพร่อง ข้อกำหนด และการทดสอบ
- ข้อบกพร่อง กรณีทดสอบ และรอบการทดสอบที่ใช้ซ้ำได้ง่าย
- สิทธิ์ที่ปรับแต่งได้ ฟิลด์ และการรายงาน
- แดชบอร์ดแบบโต้ตอบ และชาญฉลาด

- การประสานการทำงานกับบุคคลที่สาม และ REST API's
- user-friendly interface และใช้งานง่าย

ตารางที่ 2.1 ตารางแสดงการเปรียบเทียบของระบบงานที่เกี่ยวข้อง

| ฟังก์ชันการทำงาน | Usersnap | Backlog | BugHerd | Kualitee | Software Defect Reporting System |
|---|----------|---------|---------|----------|----------------------------------|
| สามารถจัดการข้อมูลส่วนตัวได้ | / | / | / | / | / |
| สามารถจัดการสมาชิกทีมได้ | / | / | / | / | / |
| สามารถจัดการโปรเจ็คได้ | / | / | / | / | / |
| สามารถสร้างรายงานข้อผิดพลาดได้ | / | / | / | / | / |
| แจ้งเตือนเมื่อมีรายงานข้อผิดพลาดผ่านกล่องแจ้งเตือนบนเว็บไซต์ได้ | - | / | - | - | / |
| สามารถแสดงลำดับความสำคัญของงานได้ | / | / | / | / | / |
| สามารถแยกสถานะของงานได้ | / | / | / | / | / |
| สามารถแสดงรายละเอียดของข้อผิดพลาดได้ | / | / | / | / | / |
| สามารถคอมเมนต์ในข้อผิดพลาดที่สร้างได้ | / | / | / | / | - |
| สามารถแคปภาพ หรืออัปเดตหน้าจอบนเว็บไซต์เพื่อแสดงผลผิดพลาดได้ | / | - | / | - | - |
| สามารถลากวางรายงานข้อมูลผิดพลาดได้ | - | / | / | - | - |