

ชื่อกลุ่ม: Deadline-Driven-Development

ชื่อโครงการ: Scrum-Framework

สมาชิกกลุ่ม:

1. ชานน จรัสสุทธิกุล รหัสนักศึกษา 55010280 sec 1
2. ณัฐพงศ์ อมรบัญชาเวช รหัสนักศึกษา 55010371 sec 1

Git Repository:

<https://github.com/CE-KMITL-OOAD-2014/Scrum-Framework>

ชื่อรายวิชา: Object-Oriented Analysis and Design

ภาคเรียนที่ 1 ปีการศึกษา 2557

บทคัดย่อ

Scrum Framework เป็นการนำหลักการของ Scrum ซึ่งเป็น Agile Methodology มาประยุกต์สร้างเป็นระบบจัดการงานและคน (Project Management System) ซึ่งโดยทั่วไป Scrum จะมีการประชุมและบริหารงานเฉพาะภายในสำนักงาน และใช้อุปกรณ์ในสำนักงาน ซึ่งทำให้การประชุมทีมหรือการติดตามงานสามารถเกิดขึ้นได้เฉพาะภายในสำนักงาน ไม่สามารถบริหารงานและติดตามงานนอกสถานที่ได้ Scrum Framework จึงถูกพัฒนาขึ้นมาเพื่อเป็นเครื่องมือที่ช่วยในการจัดการบริหารงาน โดยใช้อุปกรณ์บนหน้าเว็บไซต์แทนอุปกรณ์สำนักงาน สามารถติดตามการทำงานได้โดยไม่จำเป็นต้องอยู่ในสำนักงาน Scrum Framework จึงถูกสร้างมาเพื่อเพิ่มความคล่องตัวในการทำงาน ลดปัญหาการรวมศูนย์กลาง และสนับสนุนการทำงานอย่างไม่ถูกจำกัดสถานที่ ซึ่งเป็นสิ่งสำคัญสำหรับการทำงานด้วยวิธีการ Agile

บทนำ

Scrum เป็นกรอบการบริหารงานในทีม เพื่อผลิตและพัฒนาผลิตภัณฑ์ ในขั้นแรกของการพัฒนาผลิตภัณฑ์นั้นจะกำหนดรายการคุณสมบัติของผลิตภัณฑ์ที่ลูกค้าต้องการ เรียกว่า Product Backlog โดย Scrum จะแบ่งออกเป็นรอบการทำงานซึ่งเรียกว่า Sprint ซึ่งในแต่ละ Sprint จะทำงานตาม Product Backlog ที่มีลำดับความสำคัญสูงก่อน และมีการประชุมทีม เพื่อกำหนดงานย่อยในแต่ละ Sprint เรียกว่า Sprint Backlog โดยทุกอย่างจะถูกกำหนดลงในกระดานที่โดยทั่วไปจะแบ่งออกเป็นสามส่วนคือ To do, In Progress และ Done ซึ่ง Sprint Backlog จะถูกย้ายไปยัง In Progress และ Done ตามความคืบหน้าของแต่ละ Sprint Backlog เมื่อครบเวลาของแต่ละ Sprint ทีมก็จะประชุมทีมอีกครั้ง สำหรับ Sprint รอบถัดไป

การจัดการบริหารงานโดยใช้ Scrum โดยทั่วไปจะทำอยู่ที่ใดที่หนึ่งเช่น สำนักงาน เป็นต้น และจะใช้อุปกรณ์ที่เป็นอุปกรณ์ทางกายภาพเช่น กระดาน และ Post-it เกิดปัญหาการรวมศูนย์กลาง เช่น ไม่สามารถติดตามงานเมื่ออยู่นอกสถานที่ทำงานได้, ตัวกระดานที่แสดงถึงงานสามารถชำรุดและหมึกเลือนหายไป อีกทั้ง Post-it ที่แปะลงบนกระดานอาจหลุดหรือเปียกได้ ด้วยเหตุนี้กระดานออนไลน์ของ Scrum Framework จึงมีวัตถุประสงค์ให้บริหารจัดการงานได้โดยไม่มีข้อจำกัดด้านสถานที่ สามารถเข้าถึงได้ตลอดเวลาที่ต้องการ และไม่ต้องกังวลว่าอุปกรณ์จะชำรุด รวมทั้งเป็นการส่งเสริมให้เกิดความคล่องตัวในการทำงานมากขึ้น

งานที่เกี่ยวข้อง

- Team Foundation Server มีจุดประสงค์ทำมาเพื่อ Scrum เหมือนกันกับ Scrum Framework มีข้อแตกต่างในด้านของ Team Foundation Server มีจำนวนคนในทีม ได้ 5 คน (หากต้องการจำนวนคนใน Organization มากกว่านี้จะมีค่าใช้จ่ายเพิ่มเติม), ต้องใช้ MSDN Subscription และระบบการใช้งานที่ซับซ้อนกว่า
- Jira มีจุดประสงค์เพื่อเป็นเครื่องมือจัดการโครงการและติดตามปัญหา ซึ่งมีในส่วนของจัดการทีมรวมอยู่ด้วย โดยสามารถทำได้หลายหน้าที่ ในขณะที่ Scrum Framework จะเน้นไปที่การเป็น Taskboard สำหรับทีมเพียงจุดประสงค์เดียวเท่านั้น

ผลการวิเคราะห์ความต้องการของผู้ใช้ระบบ

Functional Requirements:

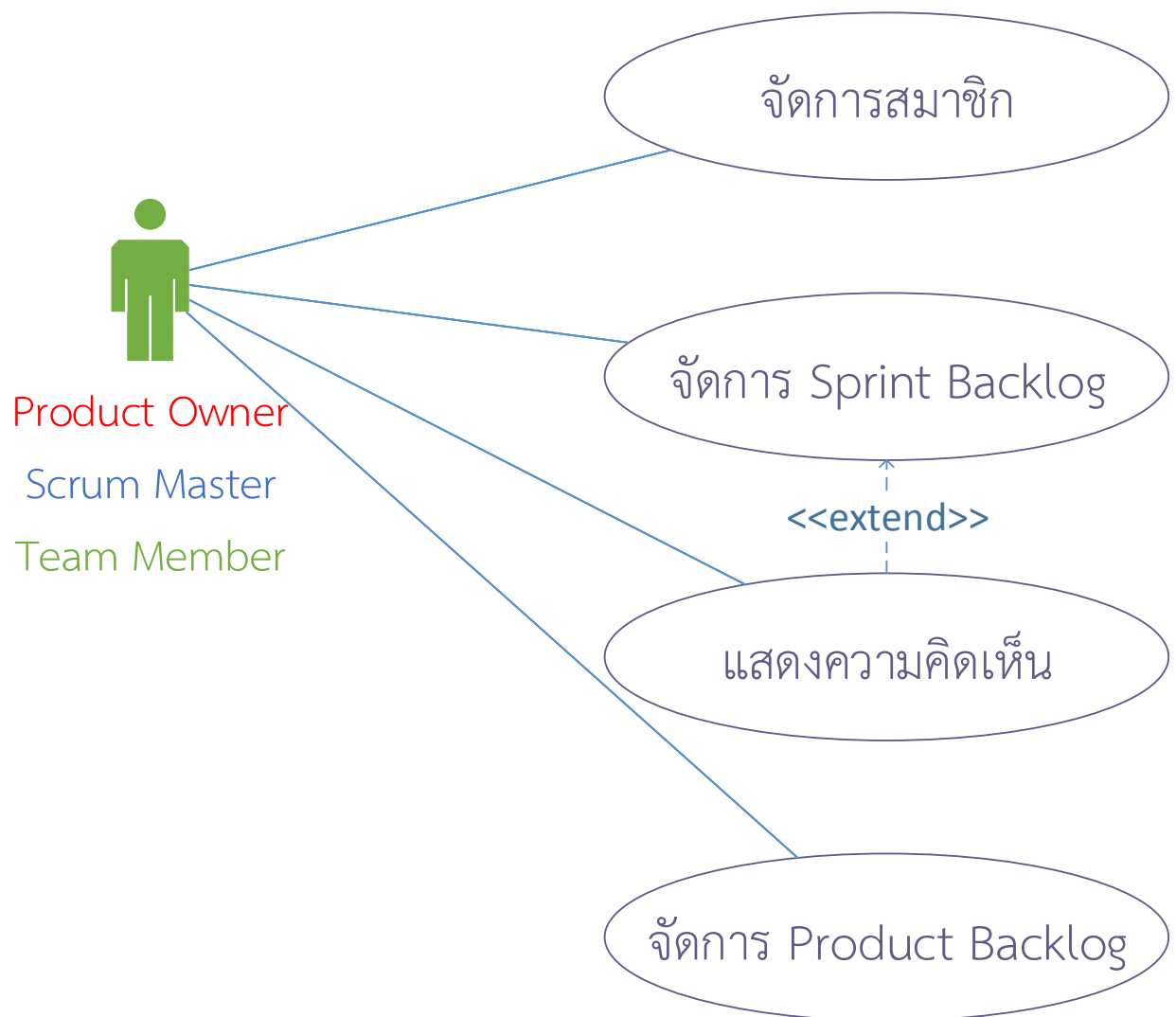
1. สามารถทำการเข้าสู่ระบบได้
2. สามารถสมัครสมาชิกได้
3. สามารถจัดการทีมได้
4. สามารถดูและจัดการ Product Backlog ได้
5. สามารถดูและจัดการ Sprint Backlog ได้
6. สามารถเรียกดู Task Board ได้
7. สามารถแสดงความคิดเห็นใน ส่วนต่าง ๆ ของระบบได้
8. มีส่วนที่สามารถถูกเรียกใช้เพื่อเชื่อมต่อแอปพลิเคชันโดยสามารถดึงข้อมูล Sprint Backlog จากแอปพลิเคชันได้

Non-Functional Requirements

1. ระบบมีความทนทานต่อความผิดพลาดภายในระบบ
2. ระบบสามารถเข้าถึงได้จากทุกที่ตลอดเวลา
3. ระบบมีความปลอดภัยโดยมีการป้องกันการโจมตีผ่านทาง SQL (SQL injection), ป้องกันการโจมตีด้วยการฝังรหัส (Cross-Site Scripting), เข้ารหัสช่องทางการสื่อสาร (https)
4. ระบบสามารถเข้าถึงได้จากอุปกรณ์เคลื่อนที่ และคอมพิวเตอร์ตั้งโต๊ะ
5. รองรับจำนวนผู้ใช้งานได้มากกว่า 100 คน พร้อม ๆ กัน
6. ระบบสามารถใช้งานได้ง่าย
7. ระบบสามารถรองรับการเติบโตของฐานผู้ใช้งาน

แนวทางการใช้งาน

Use case diagram:



Use case specifications

1. Use case name: จัดการ Sprint Backlog (Manage Sprint Backlog)

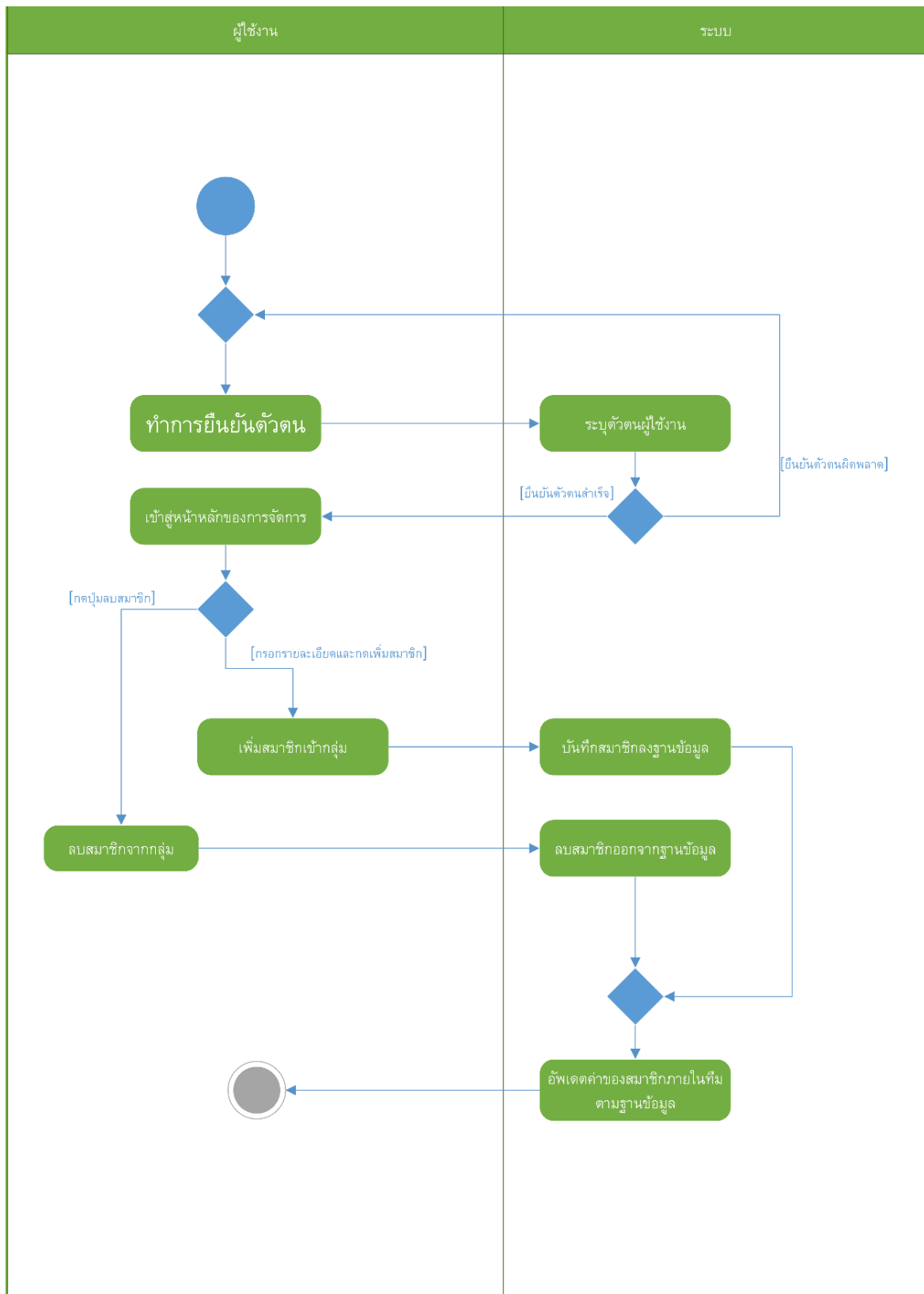
- **Use case purpose:** ให้หัวหน้าทีมและสมาชิกในทีมสามารถที่จะเพิ่ม – ลบ Product Backlog ของ Taskboard ปัจจุบันได้
- **Actors:** Scrum Master, Product Owner, Team members
- **Pre-conditions:**
 - i. ผู้ใช้งานผ่านการยืนยันตัวตน (Authentication) เรียบร้อยแล้ว
 - ii. ผู้ใช้งานเป็นหนึ่งในสมาชิกของ Taskboard นั้น ๆ
- **Post-conditions:**
 - i. มีความเปลี่ยนแปลงของ Sprint backlog ขึ้นมาใน Taskboard
- **Limitations:** ไม่สามารถเพิ่ม Sprint backlog ได้นอกเหนือไปจาก To Do, In Progress, Done
- **Assumptions:** ผู้ใช้ทราบว่าต้องการจะเปลี่ยนแปลงแก้ไขหรือเพิ่ม Sprint Backlog อย่างไรในทีม
- **Primary Scenario:**
 - A. ผู้ใช้งานทำการเลือก Taskboard ที่ต้องการจะทำการจัดการเปลี่ยนแปลงแก้ไข
 - B. Scrum Master และสมาชิกในทีมทำการเพิ่ม Sprint backlog
 - C. Scrum Master หรือสมาชิกในทีมกำหนดสถานะของแต่ละ Sprint Backlog ได้
- **Alternate Scenarios:**
 - **เงื่อนไขที่ 1 :** Scrum Master หรือสมาชิกในทีมต้องการแก้ไข Sprint Backlog
 - B1. Scrum Master หรือสมาชิกในทีมลบ Sprint backlog เดิม
 - B2. Scrum Master หรือสมาชิกในทีม เพิ่ม Sprint backlog ใหม่
 - B3. Scrum Master หรือสมาชิกในทีม บันทึกข้อมูล Sprint Backlog สู่ระบบ

2. Use case name: เพิ่มความคิดเห็น (Add Comment)

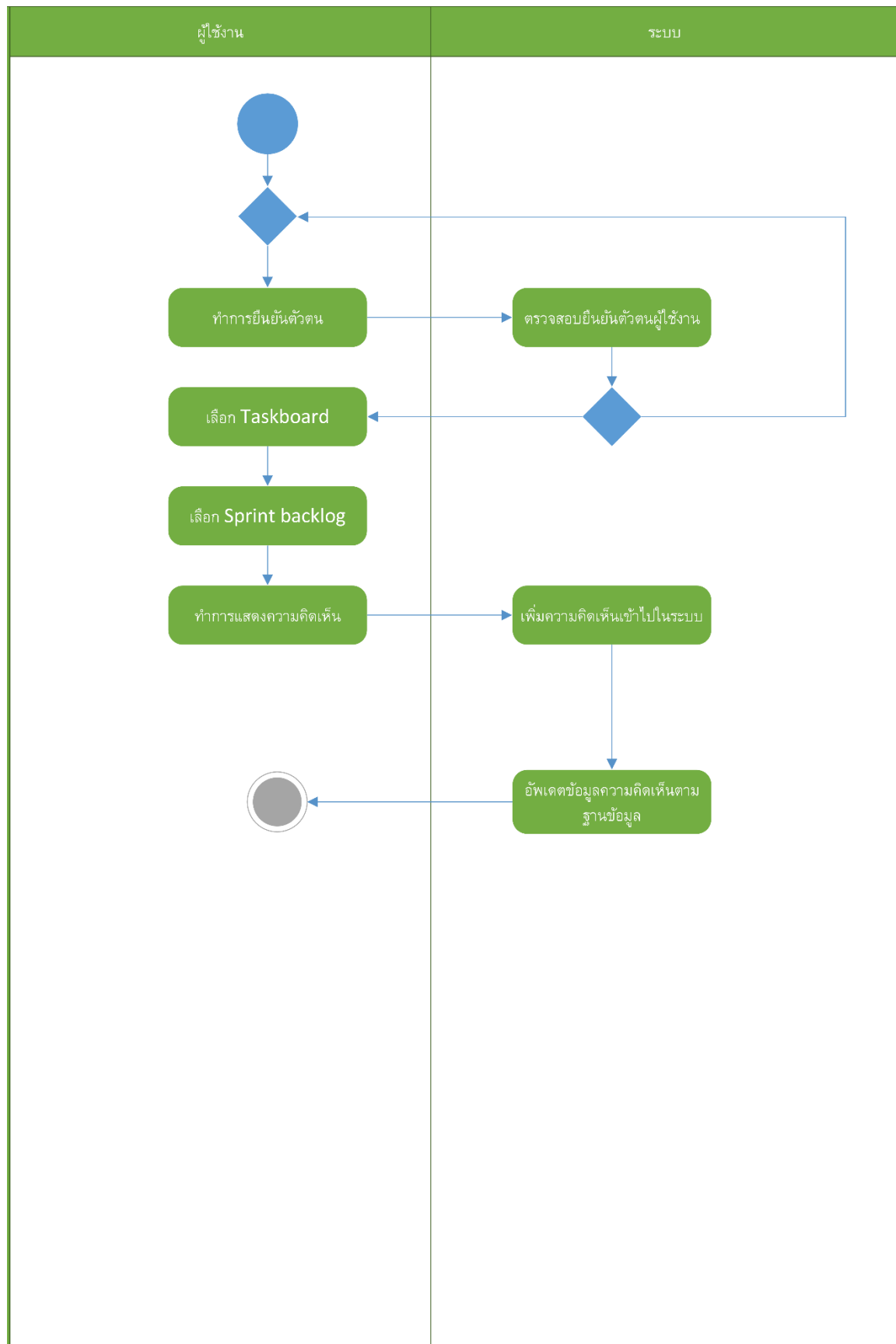
- **Use case purpose:** ให้สมาชิกที่มีส่วนเกี่ยวข้องกับทีม เพิ่มความคิดเห็นลงใน Sprint backlog และสามารถลบความคิดเห็นได้
- **Actors:** Scrum Master, Product Owner, Team Member
- **Pre-conditions:**
 - i. ผู้ใช้มีความต้องการที่จะเพิ่มความคิดเห็น
 - ii. ผู้ใช้ต้องผ่านการยืนยันตัวตน (Authentication) เรียบร้อยแล้ว
 - iii. ผู้ใช้สามารถเข้าถึง Taskboard ที่ต้องการแสดงความคิดเห็นได้
- **Post-conditions:**
 - i. ความคิดเห็นถูกบันทึกในระบบ
 - ii. ความคิดเห็นถูกแสดงผลบน Taskboard
- **Limitations:** ไม่สามารถทำการแก้ไขความคิดเห็นได้
- **Assumptions:** ผู้ใช้ทราบก่อนแล้วว่าต้องการจะแสดงความคิดเห็นลงไปใน Sprint backlog ใด
- **Primary Scenario:**
 - A. ผู้ใช้เข้าสู่ Taskboard ที่ต้องการ
 - B. ผู้ใช้ทำการเลือก Sprint backlog ที่จะแสดงความคิดเห็น
 - C. ผู้ใช้ทำการแสดงความคิดเห็นบน Sprint backlog ที่ได้เลือกไว้
- **Alternate Scenarios:**
 - **เงื่อนไขที่ 1 :** ผู้ใช้ต้องการลบความคิดเห็น
 - C1. ผู้ใช้กดปุ่มลบความคิดเห็น
 - C2. สิ้นสุดกระบวนการทำงาน
 - **เงื่อนไขที่ 2 :** ผู้ใช้ต้องการแก้ไขความคิดเห็น
 - B1. ผู้ใช้กดปุ่มลบความคิดเห็น
 - B2. กลับไปที่ทางเลือก C ของ Primary Scenario

Activity Diagrams

จัดการสมาชิก



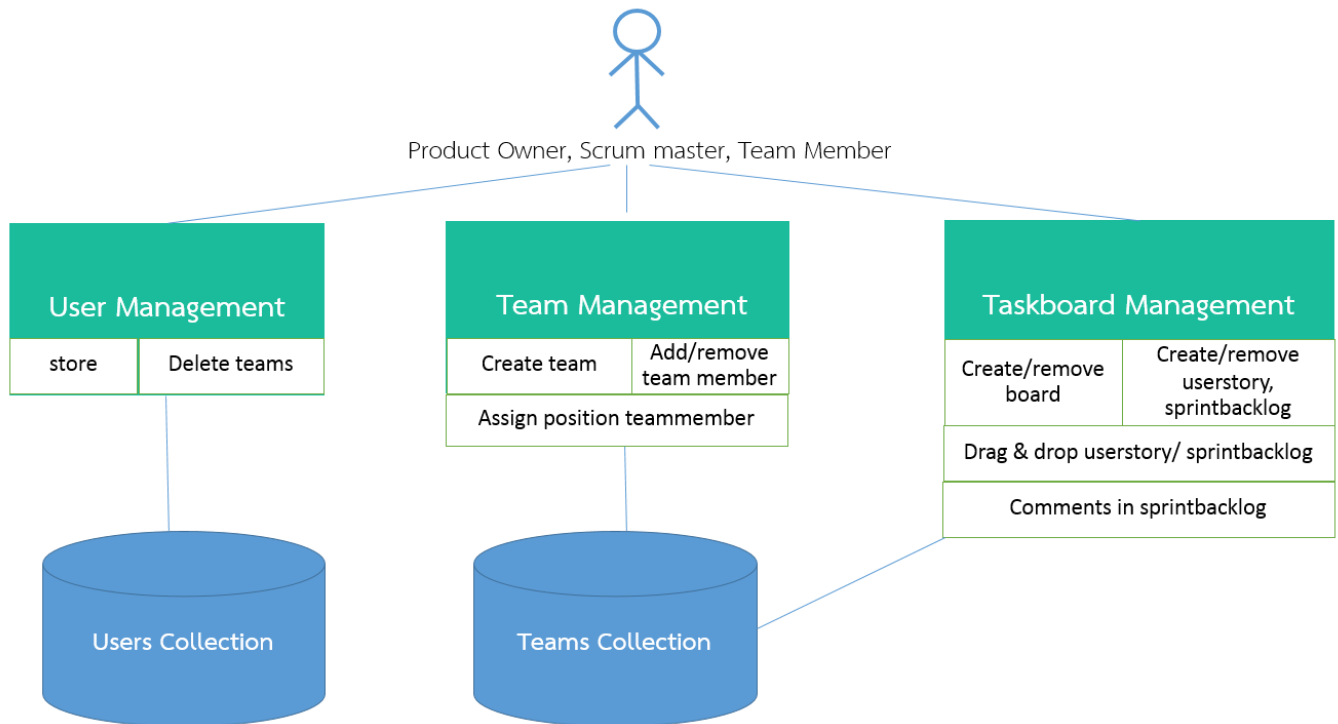
เพิ่มความคิดเห็น



Problem Analysis

- Components
 - ส่วนจัดการสิทธิ์ของสมาชิกในทีม
 - ส่วนสำหรับเพิ่ม ลบ สมาชิกในทีม
 - ส่วนสำหรับเพิ่ม User Story
 - ส่วนสำหรับถอน User Story
 - ส่วนจัดลำดับความสำคัญของ User Story
 - ส่วนสำหรับเลือก User Story เข้าไปดำเนินการใน Sprint
 - ส่วนสำหรับเพิ่ม หรือถอนงานใน Task Board
 - ส่วนสำหรับมอบหมายงานให้สมาชิกในทีม
 - ส่วนสำหรับกำหนดสถานะให้กับงานใน Task Board
 - ส่วนสำหรับเพิ่มความคิดเห็น
 - ส่วนสำหรับแก้ไขความคิดเห็น
 - ส่วนสำหรับลบความคิดเห็น - ส่วนสำหรับแจ้งเตือนผู้ใช้
- Abstraction
 - กระดาน Taskboard ที่แบ่งออกเป็น 3 ส่วน คือ To do, In progress และ Done
 - รายการความต้องการของลูกค้า (Product Backlog) แสดงรายการความต้องการของลูกค้าตามลำดับความสำคัญที่กำหนดไว้
 - รายการงานย่อย (Sprint Backlog) แสดงรายละเอียดงานย่อยๆ ที่แบ่งออกมาจาก Product Backlog
 - บันทึกแสดงความคิดเห็น (Comments) แสดงความคิดเห็น และ/หรือ อธิบายรายละเอียดงานเพิ่มเติมจาก Sprint backlog

Application Architecture



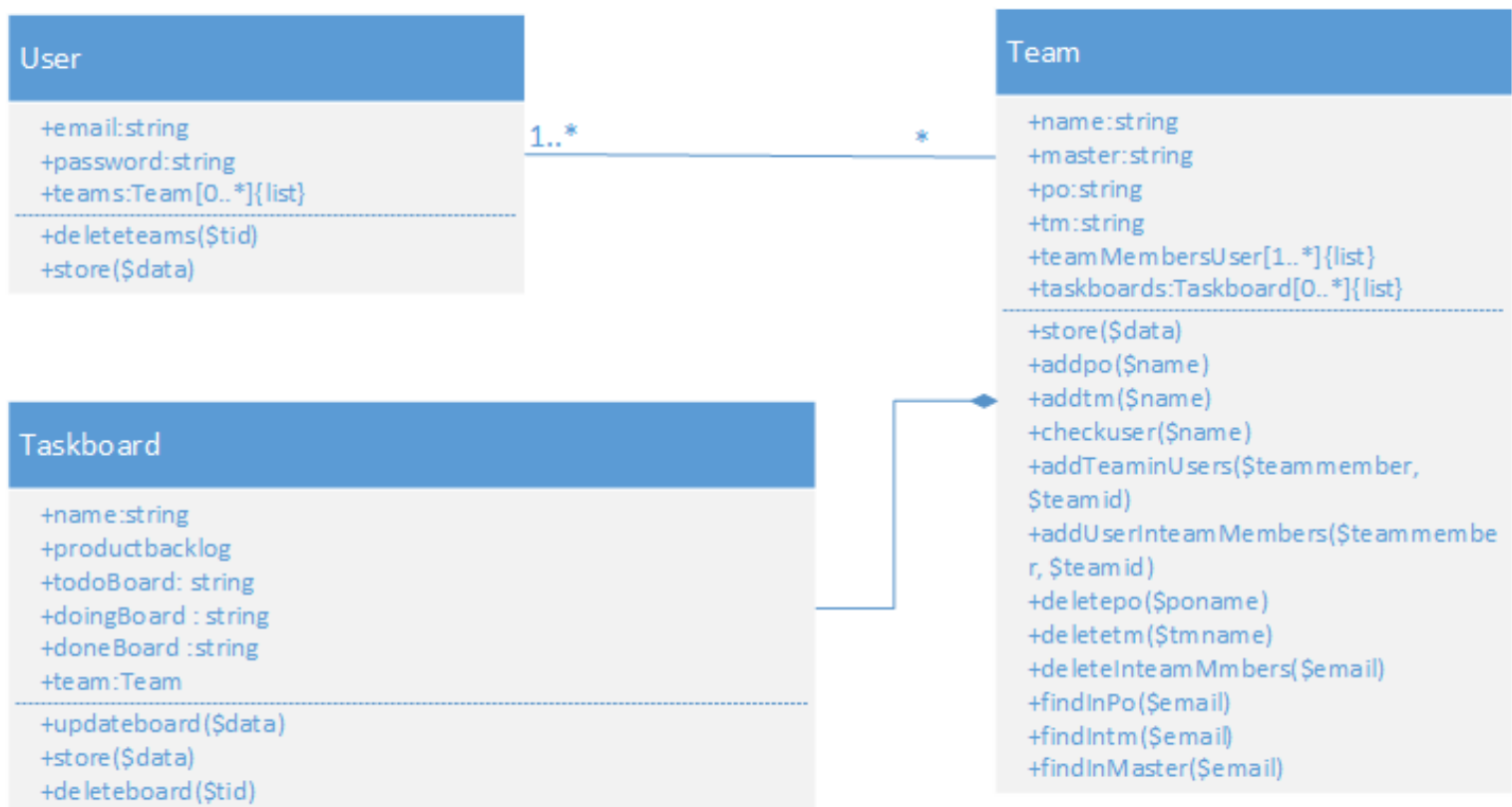
- **ระบบบริหารจัดการผู้ใช้:** เป็นระบบที่ใช้สำหรับการจัดการเกี่ยวกับผู้ใช้ โดยสามารถบันทึกข้อมูลผู้ใช้งานในระบบ และสามารถลบทีมที่ผู้ใช้เป็นสมาชิกได้กรณีที่ผู้ใช้ถูกเพิ่มเข้ามาในทีมโดย Scrum Master
- **ระบบบริหารจัดการทีม:** เป็นระบบที่ใช้จัดการเกี่ยวกับทีม โดยผู้ใช้สามารถสร้างทีมได้ เพิ่ม, ลบสมาชิกในทีมได้ สามารถให้ตำแหน่งของสมาชิกในทีมได้ว่าจะให้เป็น Product Owner หรือ Team Member
- **ระบบบริหารจัดการกระดาน:** เป็นระบบที่สามารถให้ผู้ใช้สร้างหรือลบกระดาน (Board) ได้โดยภายในกระดานมีการจัดการ Taskboard ที่แบ่งออกเป็นสี่ส่วนคือ Product Backlog, Todo, Doing และ Done ผู้ใช้สามารถสร้างและลบ userstory และ/หรือ sprintbacklog ได้และสามารถเคลื่อนย้าย userstory และ sprintbacklog ไปยัง Todo, Doing และ Done ได้ นอกจากนี้ผู้ใช้ยังสามารถแสดงความคิดเห็นในแต่ละ sprintbacklog ได้

Subsystems / Components

1. ระบบบริหารจัดการทีม (Team Management System) เป็นส่วนที่บริหารจัดการทีม การเพิ่มสมาชิกเข้ามาในทีม, การลบสมาชิกออกจากทีม และ การให้ตำแหน่งสมาชิกในทีม มีส่วนประกอบย่อย ๆ ดังนี้
 - i. ส่วนจัดการสิทธิ์ของสมาชิกในทีม สิทธิ์ของทีมแบ่งตามตำแหน่งต่างๆ คือ หัวหน้า Scrum (Scrum Master), เจ้าของผลิตภัณฑ์ซึ่งเป็นลูกค้าหรือผู้ติดต่อลูกค้า (Product Owner), สมาชิกในทีม (Team Member)
 - ii. ส่วนสำหรับเพิ่มสมาชิกในทีม
 - iii. ส่วนสำหรับลบสมาชิกในทีม ในส่วนนี้สมาชิกที่มีตำแหน่ง Scrum Master เป็นผู้ใช้งาน โดยจะติดต่อกับฐานข้อมูลของทีมและผู้ใช้งาน
2. ระบบจัดการ Product Backlog (Product Backlog Management System) เป็นส่วนที่บริหารจัดการ Product Backlog โดย สามารถเพิ่ม user story, การลบ user story, การกำหนดลำดับความสำคัญของ user story, และการนำ user story ไปใส่ในแต่ละ sprint มีส่วนประกอบย่อยๆ ดังนี้
 - i. ส่วนสำหรับเพิ่ม User Story
 - ii. ส่วนสำหรับถอน User Story
 - iii. ส่วนจัดลำดับความสำคัญของ User Story
 - iv. ส่วนสำหรับเลือก User Story เข้าไปดำเนินการใน Sprint ในส่วนนี้ สมาชิกในทีมที่มีตำแหน่งเป็น Product Owner เป็นผู้ใช้งาน โดยจะติดต่อกับฐานข้อมูลจัดการ Backlog
3. ระบบจัดการ Sprint Backlog (Sprint Backlog Management System) เป็นส่วนที่บริหารจัดการ Sprint Backlog โดยสามารถเพิ่มงาน, ลบงาน, มอบงานให้กับสมาชิกในทีม, การย้ายสถานะของงาน (มี 3 สถานะ คือ To-do In progress และ Done) และ Comment มีส่วนประกอบย่อยๆดังนี้
 - i. ส่วนสำหรับเพิ่ม หรือถอนงานใน Task Board
 - ii. ส่วนสำหรับมอบหมายงานให้สมาชิกในทีม
 - iii. ส่วนสำหรับกำหนดสถานะให้กับงานใน Task Board ในส่วนนี้มีสมาชิกของทีม และ Scrum Master เป็นผู้ใช้งาน โดยจะเป็นส่วนหลักที่ใช้ในการทำงาน และมีการติดต่อกับฐานข้อมูลจัดการ Backlog เช่นเดียวกันกับระบบจัดการ Product Backlog

4. ระบบแสดงความคิดเห็น (Comment System) เป็นส่วนที่สามารถใส่รายละเอียดหรือให้สมาชิกในทีมแสดงความคิดเห็นต่องานใน Sprint Backlog โดยสามารถ เพิ่มความคิดเห็น, แก้ไขความคิดเห็น และ ลบความคิดเห็น มีส่วนประกอบย่อย ๆ ดังนี้
 - i. ส่วนสำหรับเพิ่มความคิดเห็น
 - ii. ส่วนสำหรับแก้ไขความคิดเห็น
 - iii. ส่วนสำหรับลบความคิดเห็น ในส่วนนี้จะมีสมาชิกในทีม และ Scrum Master เป็นผู้ใช้งาน โดยมีการติดต่อกับฐานข้อมูลความคิดเห็น
5. ฐานข้อมูลสมาชิกและทีม ใช้เพื่อเก็บรายละเอียดของผู้ใช้งานแต่ละคน รายละเอียดของทีม หน้าที่ของสมาชิก และรายชื่อสมาชิกของทีม
6. ฐานข้อมูล Backlog ใช้เพื่อเก็บข้อมูลของ Project ที่ได้มีการสร้างขึ้น โดยประกอบไปด้วย Product Backlog และ Sprint Backlog ของแต่ละ Project
7. ฐานข้อมูลความคิดเห็น ใช้เพื่อเก็บความคิดเห็นทั้งหมดจาก Sprint Backlog ที่มีการเพิ่มเข้ามาในระบบ

Class Diagram



รายละเอียด Class diagram:

จากภาพ Class Diagram ข้างต้นสามารถพิจารณาออกเป็น 3 Class ได้ดังนี้ Class User สำหรับเก็บข้อมูลลง database และ ลบทีมที่ user ได้เป็นสมาชิกในทีมนั้น

Class Team เป็น Class ที่สามารถเก็บข้อมูลของทีมได้ สามารถเพิ่มหรือลบสมาชิกในทีมและสามารถให้ตำแหน่งของสมาชิกในทีมได้

ความสัมพันธ์ของ Class User กับ Class Team จะเป็นความสัมพันธ์แบบ Many to Many โดย Class User ก็จะมี teams เป็น property และ Class Team ก็จะมี teamMembers เป็น property

Class Taskboard เป็น class ที่ทำหน้าที่จัดการเกี่ยวกับ taskboard โดยสามารถสร้าง productbacklog / sprintbacklog, เปลี่ยนค่าของ product backlog/sprint backlog เมื่อผู้ใช้ทำการเคลื่อนย้ายไปยังส่วนๆต่างของกระดาน และสามารถลบ productbacklog/ sprintbacklog ได้

ความสัมพันธ์ของ Class Team กับ Class Taskboard จะเป็นความสัมพันธ์แบบ EmbeddedMany กล่าวคือ Taskboard จะอยู่ภายใน Team (embedded ใน Class Team)

Test

1. รายละเอียด: ทดสอบการดึงข้อมูล Board File name: BoardTest.php

Directory: <https://github.com/CE-KMITL-OOAD-2014/Scrum-Framework/blob/master/scrumframework/app/tests/BoardTest.php>

```
1  <?php
2
3  class BoardTest extends TestCase {
4
5      /**
6       * A basic functional test example.
7       *
8       * @return void
9       */
10     public function testBoard()
11     {
12         $taskboardController = new TaskboardController;
13         $id = "544f7ec36b26f73c048b457d";
14         $bid = "544fc1786b26f72b1b8b4571";
15         $taskboardController = $taskboardController->processGetTaskboard($id, $bid);
16         $this->assertEquals('ExampleBoard', $taskboardController);
17     }
18
19 }
```

2. รายละเอียด: ตรวจสอบการเพิ่ม product owner ในทีม : addProductOwnerTest.php

Directory: [https://github.com/CE-KMITL-OOAD-2014/Scrum-](https://github.com/CE-KMITL-OOAD-2014/Scrum-Framework/blob/master/scrumframework/app/tests/addProductOwnerTest.php)

[Framework/blob/master/scrumframework/app/tests/addProductOwnerTest.php](https://github.com/CE-KMITL-OOAD-2014/Scrum-Framework/blob/master/scrumframework/app/tests/addProductOwnerTest.php)

```
1 <?php
2
3 class addProductOwnerTest extends TestCase {
4
5     public function testpo()
6     {
7         $email = "example@email.com";
8         $team = new Team;
9
10        $testteam = $team->addpo($email);
11        $this->assertTrue($testteam);
12    }
13
14 }
```

ผลการรัน

```
dscanon@dscanon:/var/www/Scrum-Framework/scrumframework$ phpunit
PHPUnit 3.7.38 by Sebastian Bergmann.

Configuration read from /var/www/Scrum-Framework/scrumframework/phpunit.xml

.....

Time: 145 ms, Memory: 13.50Mb

OK (6 tests, 4 assertions)
dscanon@dscanon:/var/www/Scrum-Framework/scrumframework$
```

Deployment

- ระบบนี้จะใช้ Software และ Technology ดังต่อไปนี้:
 - Infrastructure:
 - Ubuntu 14.04 LTS
 - Nginx Web Server
 - Microsoft Azure Load Balancer

- Microsoft Azure DNS Server
- **Back-end**
 - PHP 5.5: ภาษาที่ใช้ในการสร้าง Application นี้
 - MongoDB: สำหรับเก็บข้อมูลใน sprint backlog เป็นรูปแบบ document และใช้ในการ backup และ restore ข้อมูล
 - Laravel: ใช้เพื่อให้ง่ายต่อการพัฒนาเว็บแอปพลิเคชัน โดยใช้ eloquent ORM ในการติดต่อด้านข้อมูล
 - Composer: เป็นตัวจัดการกับ Dependency ของ PHP
- **Front-end**
 - AngularJS: ควบคุมการทำงานของส่วน user interface ให้มีการตอบสนองกับผู้ใช้ในเชิงตรรกะ เช่นการ ย้าย sprint backlog ไปตามกระดานต่าง ๆ
 - Twitter Bootstrap: ตกแต่ง user interface หน้าเว็บไซต์ รวมถึงการทำให้เว็บไซต์นั้นรองรับการแสดงผลทั้งบน Mobile และ Desktop (responsive design)
 - HTML5: เป็นโครงสร้างของ User Interface เบื้องต้นของระบบ
 - CSS3: ใช้ในการตกแต่ง User Interface ให้สวยงาม
- โดย Software ในส่วนของ Application หลักจะทำงานบน VM (<http://168.63.175.57/>) และมี database เริ่มต้นเป็นจำนวน 3 instance

Implementation plan

การดำเนินงาน/ระยะเวลาที่ดำเนินการ	13-19 ต.ค.	20 - 26 ต.ค.	27 ต.ค.- 2 พ.ย.	3 - 9 พ.ย.	10 - 16 พ.ย.
สร้างระบบ Web Server					
ออกแบบและสร้างระบบ database					
ตั้งค่าระบบ Load Balance ของ Virtual Machine ที่ใช้					
ตั้งค่าระบบ SSL และ https					
ตั้งค่าระบบ Replication ของ database					
ตั้งค่าระบบ <u>Sharding</u> ของ database					
ทดสอบการตั้งค่างานและแก้ไขข้อผิดพลาด					

ผู้รับผิดชอบหลัก : ณัฐพงศ์ อมรบุญชเวช

การดำเนินงาน/ระยะเวลาที่ดำเนินการ	13-19 ต.ค.	20 - 26 ต.ค.	27 ต.ค.- 2 พ.ย.	3 - 9 พ.ย.	10 - 16 พ.ย.
พัฒนาระบบ login & register					
ออกแบบ User Interface					
สร้างส่วนของ User Interface					
พัฒนาระบบ Team Management					
พัฒนาระบบจัดการ Product Backlog					
พัฒนาระบบจัดการ Sprint Backlog					
สร้างระบบ Comment					

ผู้รับผิดชอบหลัก : ชานน จรัสสุทธิกุล

Evaluation

การทดลองที่สามารถทำได้ตามเป้าหมาย

การทดลองที่ 1 การสร้าง Board

จุดประสงค์การทดลอง (Goal of experiment)

-เพื่อตรวจสอบว่า Board สามารถสร้างได้

สิ่งที่วัด (Measurement Metrics)

-มีส่วนของการสร้าง Board บน User Interface สามารถกรอกชื่อ Board กับชื่อทีม และกดสร้าง Board ได้

วิธีการทดลองและสิ่งที่ต้องใช้ในการทดลอง

(Setup and Methods of your experiment)

วิธีการทดลอง

- 1.) เข้าสู่แอปพลิเคชัน, กดปุ่ม signup, กรอกรายละเอียดต่างๆเพื่อเข้าเป็นสมาชิกของระบบ
- 2.) กดปุ่ม Log in เพื่อผ่านกระบวนการยืนยันตัวตนของระบบ
- 3.) ผู้ใช้ต้องมีทีมก่อนโดยสามารถสร้างทีมโดยใส่ชื่อทีมแล้วกดปุ่ม create team หรือใช้ทีมที่มีอยู่แล้วเพื่อใช้ในการสร้าง board ได้
- 4.) กดปุ่ม add board หลังจากนั้น กรอกชื่อ Board ที่ต้องการจะสร้าง และกรอกชื่อทีม จากนั้นกด Create
- 5.) ตรวจสอบ User Interface ว่ามี Board แสดงออกมาหรือไม่ โดยเมื่อคลิกที่ Board จะแสดงหน้า Board ซึ่งประกอบด้วย 3 ส่วนหลักๆ คือ To do, In progress(Doing) และ Done และจะแสดง Product Backlog ข้างซ้ายของ Board

สิ่งที่ใช้ในการทดลอง

- 1.) อุปกรณ์ที่สามารถเข้าใช้งาน Web Browser ได้ เช่น Laptop เป็นต้น
- 2.) URL สำหรับเข้าถึง Web Application นี้ (Scrum Framework) ในที่นี้คือ scrumframework01.cloudapp.net

3.) Virtual Machine ที่ deploy scrum framework application และ run บน Microsoft Azure

4.) E-Mail และ Password ที่ใช้ในการสมัคร

5.) ทีมที่ใช้ในการสร้าง Board

6.) ชื่อของ Board ที่จะกรอกตอนสร้าง Board

ผลที่ได้จากการทดลอง (Experimental results)

- 1) สามารถกดปุ่มสร้างทีมเพื่อสร้างทีมขึ้นมาก่อนการสร้าง Board ได้
- 2) สามารถสร้าง Board โดยมีช่องให้กรอกชื่อ Board และ ชื่อทีม
- 3) เมื่อกดสร้าง Board จะมีชื่อ Board ปรากฏบริเวณทีมที่เราเลือกตอนสร้าง Board

สรุปและสิ่งที่ได้จากการทดลองนี้ (Conclusion)

ระบบสามารถสร้าง Board ได้ตามวิธีการทดลองที่กล่าวข้างต้น

การทดลองที่ 2 การแสดงผล User Interface ในส่วนของการเคลื่อนย้าย Sprint Backlog

จุดประสงค์การทดลอง (Goal of experiment)

เพื่อตรวจสอบว่าสามารถเคลื่อนย้าย Sprint Backlog ได้ โดยสามารถลาก Sprint Backlog จาก To do ไปยัง Doing ได้

สิ่งที่วัด (Measurement Metrics)

ตำแหน่งของ Sprint Backlog เมื่อลาก Sprint Backlog เคลื่อนย้ายจาก To do ไปยัง Doing

วิธีการทดลองและสิ่งที่ต้องใช้ในการทดลอง (Setup and Methods of your experiment)

วิธีการทดลอง

- 1.) เข้าสู่แอปพลิเคชัน, กดปุ่ม signup, กรอกรายละเอียดต่างๆเพื่อเข้าเป็นสมาชิกของระบบ
- 2.) กดปุ่ม Log in เพื่อผ่านกระบวนการยืนยันตัวตนของระบบ
- 3.) ผู้ใช้ต้องมีทีมก่อนโดยสามารถสร้างทีมโดยใส่ชื่อทีมแล้วกดปุ่ม create team หรือใช้ทีมที่มีอยู่แล้วเพื่อใช้ในการสร้าง board ได้

4.) กดปุ่ม add board หลังจากนั้น กรอกชื่อ Board ที่ต้องการจะสร้าง และกรอกชื่อทีม จากนั้นกด Create

5.) สร้าง Sprint Backlog ขึ้นมาโดยเมื่อสร้างแล้ว Sprint backlog จะอยู่ในส่วนของ To do บน Board

6.) ลาก (Drag) Sprint Backlog จาก To do ไปวาง (Drop) บน Doing ของ Board

7.) ตรวจสอบว่า Sprint Backlog ถูกย้ายมาส่วนของ Doing บน Board หรือไม่

สิ่งที่ใช้ในการทดลอง

- 1.) อุปกรณ์ที่สามารถเข้าใช้งาน Web Browser ได้ เช่น Laptop เป็นต้น
- 2.) URL สำหรับเข้าถึง Web Application นี้ (Scrum Framework) ในที่นี้คือ scrumframework01.cloudapp.net
- 3.) Virtual Machine ที่ deploy scrum framework application และ run บน Microsoft Azure
- 4.) E-Mail และ Password ที่ใช้ในการสมัคร
- 5.) ชื่อของ Board ที่จะกรอกตอนสร้าง Board
- 6.) รายละเอียดของ Sprint Backlog ในการสร้าง Sprint Backlog

ผลที่ได้จากการทดลอง (Experimental results)

- 1) สามารถสร้าง Sprintbacklog ได้ โดยปรากฏบน Board ในส่วนของ Todo
- 2) สามารถกดปุ่มบน sprintbacklog และสามารถลาก (Drag) จาก Board ในส่วนของ To Do ไปวาง (Drop) ในส่วนของ Doing ใน Board ได้

สรุปและสิ่งที่ได้จากการทดลองนี้ (Conclusion)

ระบบสามารถลาก SprintBacklog จาก Todo ไป Doing ได้จากวิธีการทดลองข้างต้น

บทสรุป

สิ่งที่ทางผู้พัฒนาคาดว่าผู้ใช้งานจะได้รับจากการใช้งานเว็บแอปพลิเคชันนี้คือ เป็นกระดานจดบันทึกที่ช่วยในการทำงานตามแบบของ Scrum ซึ่งเป็นการทำงานแบบ Agile ชนิดหนึ่งที่เน้นความสามารถตอบสนองต่อการเปลี่ยนแปลงได้โดยง่าย โดยเน้นไปที่ความสามารถของการทำงานโดยไม่ยึดติดกับสถานที่ใดสถานที่หนึ่ง เนื่องจากสิ่งนี้เป็นอุปสรรคอย่างหนึ่งในเรื่องของความคล่องตัวในการทำงานที่สามารถแก้ไขได้ในขณะที่แอปพลิเคชันต้องไม่ซับซ้อนหรือทำหน้าที่ที่นอกประสงค์เกินไป ทั้งนี้ก็เพื่อให้สามารถนำไปประยุกต์ใช้งานกับระบบอื่น ๆ ที่มีอยู่แล้วได้ หรือนำไปเป็นองค์ประกอบย่อยของระบบอื่นที่ต้องการ เนื่องจากระบบนี้เป็นระบบที่เรียบง่าย ไม่ซับซ้อนมาก เน้นการทำหน้าที่อย่างใดอย่างหนึ่งให้ดีในตัวของมันเอง จึงมีความคาดหวังว่าจะสามารถนำไปพัฒนาประกอบร่วมกับองค์ประกอบอื่น ๆ ได้โดยไม่ซับซ้อน

สิ่งที่ผู้พัฒนาได้รับจากการจัดทำโครงงานชิ้นนี้ขึ้นมาคือ ได้เห็นถึงข้อดีและข้อเสียของการออกแบบแอปพลิเคชันเชิงวัตถุ ซึ่งเป็นสิ่งที่ได้เคยเรียนมาแต่ไม่ทราบถึงจุดประสงค์และการประยุกต์ของการออกแบบเชิงวัตถุ แต่เป็นสิ่งที่สามารถทำให้ระบบซอฟต์แวร์ขนาดใหญ่สามารถเกิดขึ้นมาได้และยังดูแลรักษาหรือเพิ่มความสามารถต่อไปได้ รวมไปถึงกระบวนการสร้างซอฟต์แวร์ตั้งแต่ต้นกระบวนการไปจนงานเสร็จสิ้น โดยทางผู้พัฒนาก็ได้พบกับปัญหาและอุปสรรคของการสร้างซอฟต์แวร์ขึ้นมาจริง ๆ โดยที่ปัญหาของแต่ละคนถึงแม้ว่าจะใช้เครื่องมือพัฒนาที่คล้ายกัน แต่ปัญหาที่พบก็สามารถมีความแตกต่างกันอย่างสิ้นเชิงได้ ทั้งนี้ ก็ขึ้นอยู่กับว่าซอฟต์แวร์ที่สร้างขึ้นนั้นสร้างขึ้นเพื่อตอบโจทย์หรือแก้ปัญหาใด และได้รู้ถึงปัญหาที่มักจะพบเจอได้บ่อย ๆ ในกระบวนการสร้างซอฟต์แวร์อย่างเช่น การที่ใช้เวลาไปกับการศึกษา หรือเลือกเทคโนโลยีที่จะนำมาใช้มากเกินไปจากความคาดหวังว่าเทคโนโลยีนั้น ๆ จะช่วยให้การพัฒนาเป็นไปได้ด้วยความรวดเร็วมากขึ้น แต่กลายเป็นว่าเสียเวลามากเกินไปกับเรื่องนี้แทน และสุดท้ายก็ทำให้ซอฟต์แวร์ที่ออกมามีคุณสมบัติไม่ครบหรือสร้างไม่ทันตามที่วางแผนไว้แต่แรก ซึ่งปัญหานี้ก็คือการที่มุ่งเน้นให้ลำดับความสำคัญไม่ถูกลำดับเมื่อเทียบกับปัญหาอื่น ๆ เป็นต้น