

แอปพลิเคชันสำหรับสนับสนุนกระบวนการพัฒนาแบบอไจล์

APPLICATION FOR SUPPORTING AGILE PROCESS

ณรงค์เดช ศักดิ์ศรีจันทร์
ทัชกฤต บุญชนะชัย

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แอปพลิเคชันสำหรับสนับสนุนกระบวนการพัฒนาแบบอไจล์

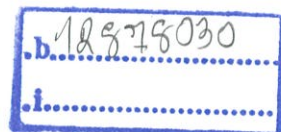
APPLICATION FOR SUPPORTING AGILE PROCESS



T148958

ณรงค์เดช ศักดิ์ศรีจันทร์
ทัชกฤต บุญชนะชัย

เลขหมู่.....
เลขทะเบียน 148958
วัน,เดือน,ปี 18 S.O. 2560



ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

หัวข้อโครงการพิเศษ แอปพลิเคชันสำหรับสนับสนุนกระบวนการพัฒนาแบบอใจล์

รายชื่อนักศึกษา นายณรงค์เดช ศักดิ์ศรีจันทร์ 55050282
 นายทักษกฤต บุญชนะชัย 55050310




ปริญญา วิทยาศาสตร์บัณฑิต

สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2558

อาจารย์ที่ปรึกษา ผศ.ดร.ศรัณย์ อินทโกสุม

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังอนุมัติให้ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2558

คณะกรรมการสอบ	ลายมือชื่อ
ดร.สันติภักดิ์ นรบิน ประธานกรรมการ	
อ.สันธนะ อู่อุดมยิ่ง กรรมการ	
ผศ.ดร.ศรัณย์ อินทโกสุม อาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อโครงการพิเศษ	แอปพลิเคชันสำหรับสนับสนุนกระบวนการพัฒนาแบบอไจล์		
รายชื่อนักศึกษา	นายณรงค์เดช	ศักดิ์ศรีจันทร์	55050282
	นายทัชกฤต	บุญชนะชัย	55050310
ปริญญา	วิทยาศาสตร์บัณฑิต		
สาขาวิชา	วิทยาการคอมพิวเตอร์		
ปีการศึกษา	2558		
อาจารย์ที่ปรึกษา	ผศ.ดร.ศรัณย์ อินทโกสุม		

บทคัดย่อ

กระบวนการพัฒนาแบบอไจล์เป็นหนึ่งในกระบวนการพัฒนาซอฟต์แวร์ที่ได้รับความนิยมในปัจจุบัน ปัจจัยหนึ่งที่ทำให้กระบวนการพัฒนาซอฟต์แวร์แบบอไจล์ประสบความสำเร็จคือการทำงานเป็นทีม ซึ่งภายในโปรเจกต์มีการแบ่งงานออกเป็นส่วนๆ เพื่อมอบหมายไปยังสมาชิกในทีมพัฒนา โดยสมาชิกแต่ละคนสามารถติดตามงานและแสดงความคิดเห็นเกี่ยวกับงานของสมาชิกคนอื่นๆ ได้ ซึ่งปัญหาพิเศษนี้มีจุดประสงค์เพื่อพัฒนาเว็บแอปพลิเคชันสำหรับสนับสนุนกระบวนการพัฒนาแบบอไจล์ โดยสามารถรองรับการแบ่งงานและการกระจายงานให้กับสมาชิกภายในทีม ซึ่งงานสามารถถูกแบ่งออกเป็นงานย่อย สมาชิกในทีมสามารถบันทึกความคืบหน้าของงานหรืองานย่อยที่ตนเองรับผิดชอบสู่ระบบ โดยสมาชิกในทีมคนอื่นๆ สามารถเข้าถึงข้อมูลความคืบหน้าของงานหรืองานย่อยและสามารถแสดงความคิดเห็นในงานหรืองานย่อยนั้นได้ ซึ่งสมาชิกภายในทีมสามารถติดตามงานได้จากสกรีมบอร์ดซึ่งจะแสดงความคืบหน้าของงานในแต่ละรอบการทำงาน อีกหนึ่งส่วนที่สำคัญของกระบวนการพัฒนาแบบอไจล์คือการรองรับการเปลี่ยนแปลงความต้องการ ซึ่งอาจต้องมีการเปลี่ยนแปลงงานหรืองานย่อยที่มีอยู่เดิม โดยแอปพลิเคชันมีการจัดการในส่วนของเวอร์ชัน ซึ่งช่วยให้สามารถติดตามและเปรียบเทียบความแตกต่างในแต่ละเวอร์ชันของงานและงานย่อย โดยแอปพลิเคชันได้ใช้กิตฮับ (GitHub) เป็นแหล่งเก็บซอร์สโค้ดเพื่อให้การบริหารจัดการซอร์สโค้ดเป็นไปโดยได้สะดวกภายใต้การพัฒนาโปรเจกต์ นอกจากนี้แอปพลิเคชันสามารถแสดงกราฟความคืบหน้าของรอบการทำงาน (Burndown Chart) ทำให้สามารถรับรู้ความคืบหน้าของการพัฒนาในรอบการทำงานนั้นเป็นผลให้สามารถง่ายต่อการวางแผนการดำเนินงานต่อไป

คำสำคัญ : กระบวนการพัฒนาแบบอไจล์, แอปพลิเคชันสำหรับสนับสนุนกระบวนการพัฒนาแบบอไจล์, กิตฮับ

Title	Application for Agile Supporting Process	
Students	Mr. Narongdet Saksrichan	55050282
	Mr. Touchakrit Boonchanachai	55050310
Degree	Bachelor of Science	
Major Program	Computer Science	
Academic Year	2015	
Advisor	Asst. Prof. Dr. Sarun Intakosum	

ABSTRACT

Agile is a software development technique that is considered popular today. One key factor to the success of an Agile based project is teamwork. A project will be divided into tasks that will be distributed to team members. Each member can keep track of each other works and give comments. The purpose of this special project is to develop a web application that can support agile process. The proposed application supports the division and distribution of tasks to each team member. A task can be divided into subtasks. Each member can easily record the progress of the tasks or subtasks under his/her responsibility to the system where others can see the progress and may leave some comments. In order to make the tracking process easily to understand, the electronics scrum board is provided so that the team members can keep track of others sprint by sprint. Another important issue for agile development process is to support requirement change which may require the modification of tasks or subtasks. The application provides the versioning system that allows the comparison and tracking of each version of tasks and subtasks. The application uses GitHub as a repository to organize source codes of the project under development. In addition, the burndown chart is provided so that the team member can easily plan the next step of the project.

Key Words: Agile development process, Agile supporting application, GitHub

กิตติกรรมประกาศ

การจัดทำปัญหาพิเศษแอปพลิเคชันสำหรับสนับสนุนกระบวนการพัฒนาแบบอโจล์ นี้สำเร็จ ลุล่วงได้ด้วยดีโดยได้รับการสนับสนุนจาก อาจารย์ศรัณย์ อินทโกสุม อาจารย์ที่ปรึกษาในปัญหาพิเศษ นี้ที่ให้คำปรึกษาการแก้ไขปัญหาและการออกแบบระบบ ซึ่งรวมถึงการตรวจสอบความถูกต้องของ เอกสารประกอบปัญหาพิเศษนี้

ขอขอบพระคุณอาจารย์สาขาวิชาวิทยาการคอมพิวเตอร์ทุกท่านที่ได้ให้วิชาความรู้ และ คำปรึกษาจนกระทั่งปัญหาพิเศษนี้สำเร็จด้วยดี

ขอขอบพระคุณบิดา มารดา และสมาชิกในครอบครัวผู้ซึ่งมีพระคุณอย่างมากที่ได้ให้กำเนิด เลี้ยงดูอบรมส่งเสริมให้ได้รับและกระทำในสิ่งที่ดีมอบสิ่งที่ดีให้กับชีวิตและอนาคตมาโดยตลอดรวมทั้ง เป็นกำลังใจให้เสมอมา

ท้ายนี้คณะผู้จัดทำขอขอบคุณเพื่อนในภาควิชาวิทยาการคอมพิวเตอร์ที่ให้คำปรึกษาและ กำลังใจเสมอมา จนกระทั่งปัญหาพิเศษฉบับนี้เสร็จสิ้นด้วยดี ทางคณะผู้จัดทำขอขอบคุณไว้ ณ ที่นี้

นายณรงค์เดช ศักดิ์ศรีจันทร์

นายทัชกฤต บุญชนะชัย

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของหัวข้อปัญหาพิเศษ.....	2
1.3 ขอบเขตของหัวข้อปัญหาพิเศษ.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 ขั้นตอนการดำเนินงาน.....	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 กระบวนการพัฒนาซอฟต์แวร์แบบอไจล์.....	4
2.2 การใช้ Github เป็นแหล่งเก็บข้อมูลเพื่อใช้ในการพัฒนาซอฟต์แวร์.....	8
2.3 ตัวอย่างระบบที่รองรับการพัฒนาซอฟต์แวร์แบบอไจล์ในรูปแบบของสกรีม.....	12
บทที่ 3 การวิเคราะห์และออกแบบระบบ.....	25
3.1 ขอบเขตความสามารถของระบบ.....	25
3.2 การวิเคราะห์ระบบ.....	26
3.3 การออกแบบระบบ.....	41
บทที่ 4 ผลการดำเนินการ.....	59
4.1 ภาพรวมของระบบ.....	59

สารบัญ (ต่อ)

เรื่อง	หน้า
4.2 ความสามารถของระบบ	59
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ	73
5.1 สรุปผลการดำเนินงาน	73
5.2 ข้อเสนอแนะของปัญหาพิเศษ	73
เอกสารอ้างอิง.....	75

สารบัญตาราง

ตารางที่	หน้า
2.1 ตารางสรุปลักษณะการทำงานของระบบที่เกี่ยวข้อง	23
3.1 เพิ่มสมาชิกในทีมเข้าสู่ระบบ	27
3.2 ลบสมาชิกออกจากระบบ	27
3.3 แก้ไขข้อมูลสมาชิก	28
3.4 สร้างโปรเจกต์	29
3.5 ลบโปรเจกต์	29
3.6 แก้ไขชื่อโปรเจกต์	30
3.7 เพิ่มสมาชิกเข้าสู่โปรเจกต์	30
3.8 ลบสมาชิกออกจากโปรเจกต์	31
3.9 เรียกดูข้อมูลของโปรเจกต์	31
3.10 สร้าง Issue	32
3.11 ลบ Issue	33
3.12 แก้ไขข้อมูลของ Issue	33
3.13 บันทึกรายละเอียดการพัฒนา (Sub Task)	34
3.14 เรียกดูรายละเอียดการพัฒนา (Sub Task)	34
3.15 เพิ่มความคิดเห็นให้กับ Issue	35
3.16 เพิ่ม Repository Path ให้กับ Issue	36
3.17 เรียกดูไฟล์ใน Repository Path ของ Issue จาก Github	37
3.18 สร้างรอบการทำงาน (Sprint)	38
3.19 เพิ่ม/ลบ Issue ในรอบการทำงาน	38
3.20 จัดการสถานะ Issue ในรอบการทำงาน	39
3.21 แก้ไขข้อมูลรอบการทำงาน	39
3.22 สร้าง Issue เวอร์ชันใหม่	40
3.23 เรียกดูรายการเวอร์ชันของ Issue	41
3.24 โครงสร้างของตาราง Member	42
3.25 โครงสร้างของตาราง Role	42
3.26 โครงสร้างของตาราง Project	43
3.27 โครงสร้างของตาราง Project_Member	43
3.28 โครงสร้างของตาราง Issue	44
3.29 โครงสร้างของตาราง assign_detail	45
3.30 โครงสร้างของตาราง Status	45

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
3.31 โครงสร้างของตาราง attached_file	46
3.32 โครงสร้างของตาราง issue_dev_history	46
3.33 โครงสร้างของตาราง Comment	47
3.34 โครงสร้างของตาราง Comment_read_history	48
3.35 โครงสร้างของตาราง Repository	49
3.36 โครงสร้างของตาราง Sprint	50
3.37 โครงสร้างของตาราง tracking	51
3.38 โครงสร้างของตาราง version_history	51

สารบัญรูป

รูปที่	หน้า
2.1 ตัวอย่างของ Scrum Board รูปแบบหนึ่ง แสดงให้เห็นถึงการแบ่งสถานะของงาน	6
2.2 การลงทะเบียนแอปพลิเคชันกับ Github	8
2.2-1 แสดง Client ID และ Client Secret ที่ได้รับการลงทะเบียน	9
2.2-2 การใช้ Client ID และ Client Key ในการเข้าสู่ระบบ Github จากแอปพลิเคชันที่สร้างขึ้น	9
2.3 รายการของ API ที่เกี่ยวกับที่เก็บข้อมูล (Repository)	10
2.4 ตัวอย่างการอธิบาย API ของ Github	10
2.5 ตัวอย่างโปรแกรมเพื่อขอรายการไฟล์ในที่เก็บไฟล์ จาก Github	11
2.6 ผลลัพธ์โปรแกรมเพื่อรับรายการไฟล์ในที่เก็บข้อมูลจาก Github	12
2.7 หน้าจอสำหรับการบอร์ดและสร้างโปรเจกต์	13
2.8 หน้าจอสำหรับสร้างบัญชีผู้ใช้และแสดงรายชื่อผู้ใช้	13
2.9 หน้าจอสำหรับการสร้างงานและสร้างรอบการทำงาน	14
2.10 หน้าจอแสดงรายการงานใน Backlog และรายละเอียดของงาน	14
2.11 หน้าจอการสร้างรอบการทำงานและเพิ่มงานเข้าสู่รอบการทำงาน	15
2.12 หน้าจอแสดงสถานะของแต่ละงานในรอบการทำงาน	15
2.13 แสดงวิธีการสร้าง Project	17
2.14 แสดงบทบาทของสมาชิกในทีมและการสร้างทีม	18
2.15 แสดงรายการใน Product Backlog และการสร้าง Backlog Item	19
2.16 แสดงรายละเอียดของ Backlog Item และการสร้าง Task งานใน Backlog Item	20
2.17 แสดงการสร้าง Sprint และแสดงรายการ Backlog Item ในแต่ละรอบ Sprint	21
2.18 แสดงสถานะ Task งานของแต่ละ Backlog Item บน Task board	21
3.1 แสดง Use Case Diagram สมาชิก	26
3.2 แสดง Use Case Diagram โปรเจกต์	28
3.3 Use Case Diagram สำหรับการจัดการ Issue	32
3.4 Use Case Diagram สำหรับการแสดงความคิดเห็นของ Issue	35
3.5 Use Case Diagram ของ Repository	36
3.6 Use Case Diagram ของรอบการทำงาน (Sprint)	37
3.7 Use Case Diagram ของการสร้างเวอร์ชันของ Issue	40
3.8 แสดง ER-Diagram สมาชิก	41
3.9 แสดง ER-Diagram ที่มีความเชื่อมโยงกับโปรเจกต์	43
3.10 แสดง ER-Diagram ของ Issue	44

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.11 แสดง ER-Diagram ของการแสดงความคิดเห็น	47
3.12 แสดง ER-Diagram ของ Repository	48
3.13 แสดง ER-Diagram ของรอบการทำงาน (Sprint)	49
3.14 แสดง ER-Diagram ของ Issue สำหรับการติดตามเวอร์ชัน	50
3.15 แสดง Work Flow การติดตามงานย่อย	52
3.16 แสดง การสร้างงานเวอร์ชันใหม่	53
3.17 แสดง Structure Chart ของระบบ	54
3.18 ส่วนติดต่อกับผู้ใช้สำหรับการจัดการโปรเจค	55
3.19 ส่วนติดต่อกับผู้ใช้สำหรับการจัดการสมาชิก	56
3.20 ส่วนติดต่อกับผู้ใช้สำหรับการจัดการ Issue ภายใน Product Backlog	56
3.21 ส่วนติดต่อกับผู้ใช้สำหรับการจัดการรอบการทำงาน (Sprint)	58
4.1 หน้าจอ Login	60
4.2 หน้าจอการจัดการสมาชิกของระบบ	60
4.3 หน้าจอการจัดการสมาชิกของระบบ	61
4.4 หน้าจอสำหรับจัดการสมาชิกในโปรเจกต์	61
4.5 หน้าจอแสดงรายละเอียดของแต่ละโปรเจกต์	62
4.6 หน้าจอแสดง Burndown Chart	62
4.7 หน้าจอสำหรับการจัดการ Product Backlog	63
4.8 หน้าจอสำหรับการจัดการ Product Backlog แสดง งานย่อย (Sub Task)	64
4.9 หน้าจอแสดงการเพิ่ม Repository จาก Github	64
4.10 หน้าจอแสดงข้อมูลใน Repository จาก Github	65
4.11 หน้าจอสำหรับการเพิ่ม Attach File ให้กับงาน	65
4.12 หน้าจอสำหรับแสดงการแจ้งเตือนข้อความที่ยังไม่ได้อ่าน	66
4.13 หน้าจอสำหรับแสดงความคิดเห็นของงาน	66
4.14 หน้าจอสำหรับการมอบหมายงานให้กับสมาชิกในทีม	67
4.15 หน้าจอแสดงงานในรอบการทำงาน	67
4.16 หน้าจอสำหรับจัดการ Sprint	68
4.17 หน้าจอแสดงข้อมูลของงานและงานย่อยในหน้า Sprint	69
4.18 หน้าจอแสดงรายละเอียดของงานย่อยในหน้า Sprint	69
4.19 หน้าจอแสดงประวัติความคืบหน้าของงานย่อยในรอบการทำงาน	70
4.20 หน้าจอแสดงหน้าต่างให้ผู้ใช้กรอกเหตุผลที่เลื่อนงานย้อนหลัง	71

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.21 หน้าจอแสดงข้อมูลสำหรับสร้างงานย่อยเวอร์ชันใหม่	71
4.22 หน้าจอแสดงเวอร์ชันทั้งหมดของงานย่อย	71

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

เนื่องจากปัจจุบันเทคโนโลยีได้เข้ามามีบทบาทในชีวิตประจำวันมากขึ้นจนกลายเป็นส่วนหนึ่งในการดำเนินชีวิต รวมถึงในองค์กรต่างๆ ซึ่งต้องพึ่งพาเทคโนโลยีเพื่ออำนวยความสะดวกในการดำเนินงานโดยการใช้เทคโนโลยี ที่เห็นเด่นชัดคือการใช้งานด้านซอฟต์แวร์หรือแอปพลิเคชันต่างๆ ผ่านทางระบบคอมพิวเตอร์ สมาร์ทโฟน แท็บเล็ต และอุปกรณ์อื่นๆ ทำให้การพัฒนาซอฟต์แวร์เพื่อรองรับความต้องการของผู้ใช้เป็นส่วนที่สำคัญ ในปัจจุบันมีกระบวนการพัฒนาซอฟต์แวร์ให้เลือกใช้หลากหลายโมเดล ซึ่งขึ้นอยู่กับความเหมาะสมกับซอฟต์แวร์ที่ต้องการพัฒนา หนึ่งในโมเดลที่ถูกนำมาใช้ในปัจจุบันคือการพัฒนาซอฟต์แวร์ในรูปแบบอจิล (Agile Process Model) เป็นกระบวนการพัฒนาที่เน้นการทำงานที่รวดเร็ว รองรับการเปลี่ยนแปลงความต้องการของลูกค้า (Requirement) มีจำนวนทีมงานผู้พัฒนาไม่มาก โดยมีการแบ่งงานออกเป็นงานย่อย ผู้พัฒนาจะร่วมกันดำเนินงานในส่วนต่างๆ โดยจะมีการติดตามความคืบหน้าของงานในแต่ละงานที่ได้ถูกมอบหมายให้กับสมาชิกภายในทีม ซึ่งการบริหารจัดการงานเพื่อให้เสร็จตามกำหนดเป็นสิ่งสำคัญในการพัฒนาซอฟต์แวร์ การพัฒนาในปัจจุบันมีการวางแผนงานโดยใช้กระดานบอร์ดเป็นเครื่องมือช่วยในการวางแผนงาน ซึ่งทำให้มีข้อจำกัดในการจัดการงานหากงานมีจำนวนมากจะทำให้ไม่สะดวกในการจัดการ รวมถึงการติดตามงานและการเรียกดูการทำงานย้อนหลัง

ผู้จัดทำจึงมีแนวคิดที่จะพัฒนาเว็บแอปพลิเคชัน เพื่อเป็นเครื่องมือที่ช่วยในการบริหารจัดการงานในการพัฒนาซอฟต์แวร์ ในรูปแบบของอจิล ซึ่งจะทำให้มีความสะดวกในการบริหารจัดการงาน

1.2 วัตถุประสงค์ของหัวข้อปัญหาพิเศษ

พัฒนาโปรแกรมเพื่อรองรับการบริหารจัดการงานในการพัฒนาซอฟต์แวร์ในรูปแบบอโใจล์ ให้สามารถบริหารจัดการงานได้สะดวกยิ่งขึ้น

1.3 ขอบเขตของหัวข้อปัญหาพิเศษ

เว็บแอปพลิเคชัน จัดการสมาชิกภายในทีมและบริหารจัดการงานตามรูปแบบอโใจล์ในส่วนของการสร้างงาน (Issues) การสร้างรอบการทำงาน (Sprint) การบริการจัดการงานในรอบการทำงานนั้นๆ รวมทั้งการควบคุมเวอร์ชันของงาน การแสดงความคิดเห็นเกี่ยวกับงาน และการเรียกดูไฟล์หรือซอร์สโค้ดที่เกี่ยวข้องกับงาน

ข้อจำกัด

- การเรียกดูซอร์สโค้ดเป็นการเรียกดูจากซอร์สโค้ดที่จัดเก็บไว้บนเว็บไซต์ Github เท่านั้น
- สามารถเข้าถึงที่จัดเก็บข้อมูลเว็บไซต์ Github ที่เป็นที่จัดเก็บข้อมูลแบบสาธารณะเท่านั้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ได้แอปพลิเคชันที่สามารถบริหารจัดการงานตามรูปแบบของอโใจล์ในเบื้องต้น

- ในส่วนการจัดการงาน
- ในส่วนแสดงความคิดเห็นเกี่ยวกับงาน
- ในส่วนการจัดการรอบการทำงาน
- ในส่วนการติดตามเวอร์ชันของงาน

1.5 ขั้นตอนการดำเนินงาน

1. กำหนดและเลือกหัวข้อโครงการ
2. เก็บรวบรวมความต้องการเพื่อใช้ในการพัฒนาแอปพลิเคชัน
3. ทำการวิเคราะห์และกำหนดขอบเขตของแอปพลิเคชัน
4. ทำการออกแบบกระบวนการทำงานและการเก็บข้อมูลในฐานข้อมูลของแอปพลิเคชัน
5. ทำการออกแบบส่วนติดต่อกับผู้ใช้ให้มีการใช้งานง่าย สามารถเรียนรู้และจดจำการใช้งานได้อย่างรวดเร็ว
6. ทำการพัฒนาแอปพลิเคชันตามที่ได้ออกแบบไว้

7. ทำการทดสอบกระบวนการทำงานทั้งหมดของแอปพลิเคชัน ให้ทำงานได้ถูกต้องตามที่ ออกแบบไว้และปรับปรุงให้สมบูรณ์
8. จัดทำคู่มือการใช้งานและการดูแลแอปพลิเคชัน

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้องกับการพัฒนาแอปพลิเคชันสำหรับรองรับการพัฒนาซอฟต์แวร์แบบโอโลจีซึ่งมีการใช้ที่เก็บข้อมูล (Repository) จากภายนอก และยกตัวอย่างแอปพลิเคชันสำหรับการรองรับการพัฒนาซอฟต์แวร์แบบโอโลจีโดยแสดงในหัวข้อต่อไปนี้

2.1 กระบวนการพัฒนาซอฟต์แวร์แบบโอโลจี

การพัฒนาซอฟต์แวร์ในรูปแบบดั้งเดิม ซอฟต์แวร์จะถูกพัฒนาในกระบวนการที่ชัดเจน เรียบง่าย เป็นขั้นตอน และไม่ค่อยมีการเปลี่ยนแปลงงานที่ได้พัฒนาไว้แล้ว เนื่องด้วยซอฟต์แวร์ส่วนใหญ่มีขั้นตอนการทำงานที่ไม่ซับซ้อนมาก แต่ในปัจจุบันซอฟต์แวร์มีขนาดที่ใหญ่มากขึ้น ระบบมีความสลับซับซ้อนมากขึ้น ลูกค้ามีความต้องการที่ไม่ชัดเจนและความต้องการนั้นยังมีการเปลี่ยนแปลงอยู่ตลอดเวลา ซึ่งความต้องการที่เปลี่ยนแปลงนั้น ทีมพัฒนาไม่สามารถควบคุมและคาดเดาได้ ดังนั้นหากทีมพัฒนายังยึดกระบวนการพัฒนาซอฟต์แวร์รูปแบบเดิมอยู่ อาจทำให้ไม่สามารถพัฒนาซอฟต์แวร์ให้สำเร็จลุล่วงได้ตามที่วางแผนไว้ ปัจจุบัน จึงมีผู้คิดค้นวิธีการพัฒนาซอฟต์แวร์รูปแบบใหม่ขึ้นมาหลากหลายวิธี เพื่อเป็นแนวทางให้ผู้พัฒนาสามารถเลือกใช้กระบวนการพัฒนาซอฟต์แวร์ได้ตามความเหมาะสมต่อระบบงานที่สุด

โอโลจี เป็นกระบวนการพัฒนาซอฟต์แวร์รูปแบบหนึ่ง มีหลักการสำคัญคือ พร้อมรับการเปลี่ยนแปลงความต้องการของลูกค้าตลอดเวลา สามารถส่งมอบงานที่มีคุณภาพให้ลูกค้าได้อย่างรวดเร็ว ต่อเนื่อง ผู้ใช้งานและผู้พัฒนาซอฟต์แวร์มีการติดต่อสื่อสารกันอย่างใกล้ชิด เพื่อให้งานที่ได้ตรงตามความต้องการของผู้ใช้ที่สุด การแลกเปลี่ยนข้อมูลกันจะใช้การพูดคุย ติดต่อสื่อสาร ซึ่งมีรูปแบบการทำงานหลากหลายรูปแบบที่นำเอากระบวนการพัฒนาซอฟต์แวร์แบบโอโลจีมาใช้ เช่น Crystal extreme programming Feature driven development Dynamic Systems Development Method และรูปแบบของ Scrum เป็นต้น [1] ซึ่งทางผู้จัดทำได้ให้ความสำคัญไปที่การทำงานในรูปแบบ Scrum เนื่องจากเป็นแบบแผนการทำงานที่สมาชิกภายในทีมสามารถรับรู้ความคืบหน้าของงานได้อย่างชัดเจน

สกรัม (Scrum) เป็นแบบแผนการทำงานที่นำมาใช้พัฒนาและสนับสนุนผลิตภัณฑ์ที่มีความซับซ้อน ใช้ทีมพัฒนาไม่ใหญ่มาก โดยที่สมาชิกภายในทีมต่างผลักดันซึ่งกันและกัน เน้นให้ทีมงานร่วมกันและมีการประชุมกันอย่างสม่ำเสมอ โดยที่เจ้าของผลิตภัณฑ์หรือผู้ใช้ระบบจะต้องทำงานร่วมกับผู้พัฒนาอย่างใกล้ชิด สกรัมใช้หลักการพัฒนาแบบทำซ้ำไปเรื่อยๆ (Iteration) โดยทั่วไปใน 1

รอบการทำงานทีมพัฒนาจะใช้เวลาประมาณ 2 สัปดาห์ ในการทำงานให้เกิดผลผลิตที่สามารถใช้งาน ได้จริงและเกิดประโยชน์สูงสุด เพื่อส่งมอบให้ลูกค้าได้ทดลองใช้งานอย่างต่อเนื่องสม่ำเสมอ [2]

สกรัม ประกอบด้วยสาระสำคัญดังต่อไปนี้

ส่วนของหน้าที่และบทบาท แบ่งได้ 3 บทบาท

1. Product Owner หรือเจ้าของผลิตภัณฑ์ มีหน้าที่เขียนความต้องการของระบบให้มีความ ครบถ้วนสมบูรณ์ รับผิดชอบจัดการและเรียงลำดับความสำคัญภายใน Product Backlog และมี หน้าที่อธิบายรายละเอียดต่างๆใน Product Backlog ให้ทีมสกรัมเข้าใจ

2. Scrum Development Team หรือทีมงานผู้พัฒนา คือกลุ่มบุคคลที่ปฏิบัติงานจริงมี จำนวนสมาชิกประมาณ 5 ถึง 9 คน โดยแต่ละคนในทีมไม่มีตำแหน่งหน้าที่เฉพาะ สามารถทำงาน ทดแทนกัน สมาชิกในทีมประกอบด้วยตำแหน่งงานทางด้านการพัฒนาซอฟต์แวร์ เช่น นักวิเคราะห์ และออกแบบระบบ นักออกแบบเว็บ นักทดสอบระบบ เป็นต้น สมาชิกในทีมจะช่วยกันทำงานให้ ลุล่วงตามเป้าหมายในเวลาที่กำหนด

3. Scrum Master มีหน้าที่อำนวยความสะดวกให้แก่สมาชิกภายในทีม คอยประสานการ ทำงานระหว่างทีมพัฒนาและเจ้าของผลิตภัณฑ์ให้สามารถทำงานร่วมกันได้อย่างเข้าใจและตรงตาม เป้าหมายที่วางไว้ แก้ไขอุปสรรคต่างๆที่อาจเกิดขึ้นระหว่างการทำงาน ควบคุมการทำงานให้ สอดคล้องตามกรอบการทำงานแบบสกรัม

ส่วนของกระบวนการทำงาน

เริ่มแรกเจ้าของผลิตภัณฑ์จะนำรายการความต้องการของระบบทั้งหมดมาจัดลำดับ ความสำคัญใส่ลงใน Product Backlog โดยแต่ละรายการใน Product Backlog จะเรียกว่า Product Backlog Items (PBI) ซึ่งลำดับงานที่ต้องการให้ทำก่อนจะถูกจัดไว้ด้านบนของ Product Backlog ไล่เรียงลงมาเรื่อยๆจากบนลงล่าง โดยทีมงานผู้พัฒนาจะเลือกลำดับรายการที่ถูกจัดลำดับ ความสำคัญไว้ส่วนหนึ่ง ทำการแตกงานออกมาเป็นส่วนๆ (Task) และจัดเตรียมงานเหล่านั้นเข้าสู่ รอบการทำงาน (Sprint) เมื่อเข้าสู่รอบการทำงานทีมพัฒนาจะใช้เวลาในการทำงานประมาณ 2 อาทิตย์ช่วยกันสร้างงานให้เสร็จตามแผนที่วางไว้ ซึ่งจุดเด่นอย่างหนึ่งของการทำงานแบบสกรัมคือ ใน ระหว่างทำงาน ทีมพัฒนาสามารถรับรู้ถึงรายละเอียดต่างๆของการทำงานในรอบนั้นเช่น รายการงาน ที่จะต้องทำในรอบการทำงานนั้นทั้งหมด รายละเอียดต่างๆของแต่ละงาน ชนิดของงาน ใครเป็นผู้ได้

รับผิดชอบงาน ระยะเวลาที่คาดว่าจะสามารถทำงานนั้นจนเสร็จสมบูรณ์ สถานะของงาน ซึ่งทีมพัฒนาสามารถรับรู้ข้อมูลต่างๆที่กล่าวมาทั้งหมดนั้นได้จากสิ่งที่เรียกว่า Scrum Board

Story	To Do	In Process	To Verify	Done	
As a user, I... 8 points	Code the... 9 Code the... 2 Test the... 8	Test the... 8 Code the... 8 Test the... 8 Test the... 4	Code the... DC 4 Test the... SC 8	Test the... SC 6	Code the... SC 8 Test the... SC 8 Test the... SC 8 Test the... SC 6
As a user, I... 5 points	Code the... 8 Code the... 4	Test the... 8 Code the... 6	Code the... DC 8		Test the... SC 8 Test the... SC 8 Test the... SC 6

รูปที่ 2.1 ตัวอย่างของ Scrum Board รูปแบบหนึ่ง แสดงให้เห็นถึงการแบ่งสถานะของงาน [3]

Scrum Board หรือ Task Board หรือ Sprint Board เป็นเครื่องมือที่ช่วยให้ทีมพัฒนาซอฟต์แวร์สามารถรับรู้รายละเอียดต่างๆเกี่ยวกับงานทั้งหมดในรอบการทำงานนั้น สามารถรับรู้ถึงความคืบหน้าของงานทั้งหมด เพื่อให้ทีมพัฒนาสามารถประเมิน วางแผน แจกจ่ายงานได้สะดวกขึ้น Scrum Board ส่วนใหญ่จะอยู่ในรูปแบบของ Board เพื่อให้ทีมพัฒนาทุกคนสามารถรับรู้ข้อมูลสถานะของงานได้ง่าย Scrum Board จะแบ่งสถานะของงานออกเป็นช่วงๆ เช่น ช่วงของงานที่จะทำงานที่กำลังทำ งานที่ทำเสร็จสิ้นแล้ว แต่สถานะของงานนั้นมีรูปแบบไม่ตายตัว สามารถเปลี่ยนแปลงเพิ่มเติมได้ตามความเหมาะสมของแต่ละองค์กร งานแต่ละงานที่อยู่ในรอบการทำงานนั้นจะถูกใส่รายละเอียดต่างๆเช่น ชื่องาน ชนิดงาน ผู้รับผิดชอบ ระยะเวลาที่คาดว่าจะสามารถทำงานนั้นจนเสร็จ และอาจมีรายละเอียดเพิ่มเติมได้แล้วแต่องค์กร ทีมผู้พัฒนาจะนำงานทุกงานงานมาติดไว้ที่ Board เพื่อกำหนดสถานะของงานแรกเริ่ม และสามารถขยับงานแต่ละงานไปสถานะอื่นตามความเหมาะสม โดยทีมผู้พัฒนาทุกคนจะกำหนด กฎ กติกาต่างๆ สำหรับ การเปลี่ยนแปลงสถานะของงาน การเปลี่ยนผู้รับผิดชอบงาน หรือการเปลี่ยนแปลงข้อมูลในงานบน Board กันเองตามความเหมาะสมของแต่ละองค์กร Scrum Board จะช่วยให้การบริหาร จัดการงานของทีมพัฒนาทำได้สะดวกยิ่งขึ้น

ในระหว่างรอบการทำงาน เพื่อให้การทำงานในแต่ละรอบนั้นสามารถสร้างงานได้อย่างเหมาะสมที่สุด ทีมพัฒนาสามารถวางแผน พิจารณาปรับแต่งรายการใน Product Backlog นำงานเข้ามาเพิ่มหรือนำงานออกจากรอบการทำงานได้ โดยงานทุกงานที่จะกระทำนั้นต้องได้รับการอนุมัติจากเจ้าของผลิตภัณฑ์ก่อน เมื่อสิ้นสุดรอบการทำงานเจ้าของผลิตภัณฑ์จะประเมินงานทุกงาน ว่างานไหนเสร็จสมบูรณ์งานไหนที่ยังต้องแก้ไข ปรับปรุง เพื่อใช้สำหรับวางแผนงานในรอบการทำงานถัดไป หลังจากนั้นเจ้าของผลิตภัณฑ์และทีมพัฒนาจะมาประชุมวางแผนถึงรายการ Product Backlog ต่อไปที่ถูกปรับปรุงจัดลำดับความสำคัญใหม่เพื่อจะนำเข้าสู่รอบการทำงานต่อไป ทำซ้ำไปเรื่อยๆจนเสร็จสิ้นโครงการ

ซึ่งการพัฒนาซอฟต์แวร์ในแบบแผนการทำงานแบบสกรัม สามารถช่วยในเรื่องของการบริหารโครงการให้มีการทำงานเป็นขั้นตอน การจัดลำดับความสำคัญของรายการงานที่จะทำ ทำให้ทีมพัฒนาสามารถสร้างงานเป็นลำดับ เนื่องจากการแบ่งงานที่จะทำออกเป็นส่วนๆ ทำให้การทำงานในแต่ละรอบการทำงานที่ทีมพัฒนาสามารถมองเห็นเป้าหมายและผลลัพธ์ที่จะเกิดขึ้นได้ชัดเจน ทีมพัฒนาสามารถสร้างงานในแต่ละรอบการทำงานได้อย่างเต็มที่ การมี Scrum Board ทำให้ทีมพัฒนาและลูกค้าสามารถรับรู้และติดตามความคืบหน้าของงานทุกงานในรอบการทำงานนั้นได้อย่างสม่ำเสมอ สามารถวางแผนกระจายงานให้คนภายในทีมพัฒนารับผิดชอบได้โดยทุกคนภายในทีมสามารถรับรู้โดยทั่วกัน

การพัฒนาซอฟต์แวร์ไม่ว่าจะเลือกใช้กระบวนการใดในการพัฒนา ผู้พัฒนาจำเป็นต้องมีพื้นที่สำหรับเก็บข้อมูลต่างๆ ที่ได้มีการพัฒนาขึ้นมาตัวอย่างเช่น รูปภาพ ซอร์สโค้ด ไฟล์ต่างๆ เป็นต้น เพื่อให้สามารถเข้าถึงได้จากบุคคลอื่นๆ ภายในทีมที่ร่วมพัฒนา ดังนั้นการเลือกใช้พื้นที่เก็บข้อมูลจึงเป็นสิ่งที่สำคัญ ในปัจจุบันมีเว็บไซต์บริการสำหรับเก็บข้อมูลเพื่อให้ง่ายต่อการเข้าถึงของผู้ใช้ โดยเว็บไซต์ Github เป็นเว็บไซต์ที่ให้บริการพื้นที่เก็บข้อมูล ซึ่งมีทั้งแบบฟรีและมีค่าใช้จ่าย ทำให้ช่วยเพิ่มความสะดวกให้ทีมผู้พัฒนาซอฟต์แวร์

2.2 การใช้ Github เป็นแหล่งเก็บข้อมูลเพื่อใช้ในการพัฒนาซอฟต์แวร์

Github เป็นเว็บไซต์ให้บริการที่เก็บข้อมูล (Repository) สำหรับการพัฒนาซอฟต์แวร์ โดยสามารถจัดการการจัดการซอร์สโค้ด (Source Code Management) และไฟล์ต่างๆ รวมไปถึงการควบคุมเวอร์ชันจากการแก้ไขของสมาชิกในทีมของการพัฒนาซอฟต์แวร์ เป็นบริการที่ให้ความสะดวกกับผู้พัฒนาซอฟต์แวร์

วิธีการติดต่อเพื่อร้องขอข้อมูลจาก Github

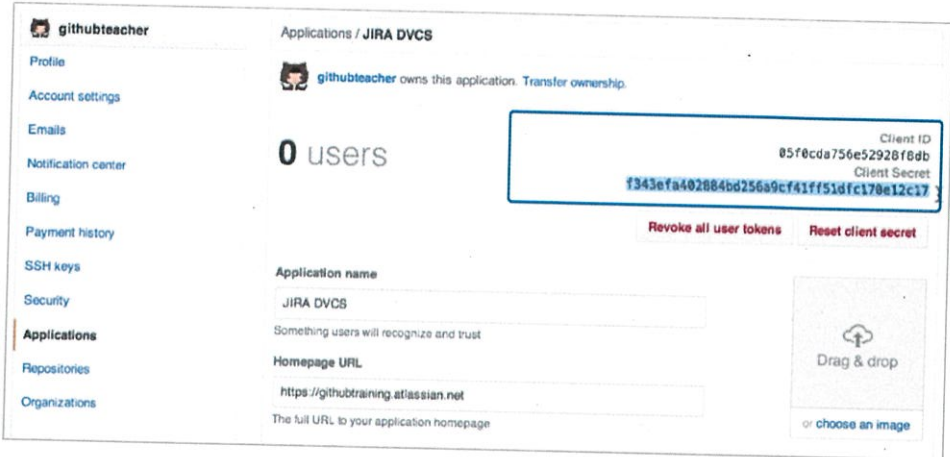
การติดต่อกับ Github เพื่อร้องขอข้อมูลของการพัฒนาซอฟต์แวร์ ตัวอย่างเช่น การเรียกดูไฟล์ในที่เก็บข้อมูล (Repository) การเรียกดูซอร์สโค้ด เป็นต้น โดย Github ได้เตรียม API สำหรับให้ผู้พัฒนาติดต่อเพื่อขอรับข้อมูลต่างๆ โดยเป็นลักษณะ URI ซึ่งดำเนินโดยใช้ Http โพรโตคอล โดยมีโดยวิธีการดังนี้

1. ผู้ใช้งานสมัครบัญชีผู้ใช้ Github และลงทะเบียนแอปพลิเคชันของตนกับ Github จะทำให้ผู้ใช้สามารถเข้าสู่ระบบ (Log in) ของ Github จากแอปพลิเคชันของตนได้ จากรูปที่ 2.2 ผู้ใช้ทำการกรอกข้อมูลชื่อแอปพลิเคชัน URL ของเว็บแอปพลิเคชัน คำอธิบายแอปพลิเคชัน และ URL ที่ทำการตอบกลับหลังจาก Github ทำการตรวจสอบการเข้าสู่ระบบ

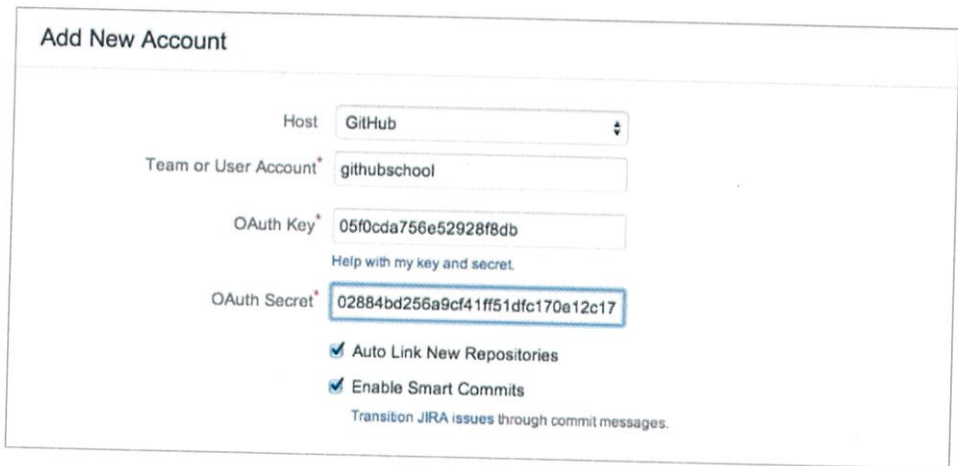
The screenshot shows the GitHub interface for registering a new OAuth application. On the left is a sidebar with the user's profile 'githubteacher' and various settings. The main area is titled 'Applications / Register a new OAuth application'. It contains several form fields: 'Application name' with the value 'JIRA DVCS', 'Homepage URL' with 'https://githubtraining.atlassian.net', 'Application description' with 'The JIRA Instance for the GitHub Training Team', and 'Authorization callback URL' with 'https://githubtraining.atlassian.net'. There is a 'Register application' button at the bottom of the form.

รูปที่ 2.2 การลงทะเบียนแอปพลิเคชันกับ Github

โดยจากรูปที่ 2.2-1 เมื่อทำการลงทะเบียนแล้วผู้ใช้จะได้รับ Client ID และ Client Key สำหรับใช้ในการเข้าสู่ระบบ Github จากแอปพลิเคชันของตนเอง โดยรูปที่ 2.2-2 ตัวอย่างการนำ Client ID และ Client Key ไปใช้ในการเข้าสู่ระบบ Github จากแอปพลิเคชันที่สร้างขึ้น ซึ่งการเข้าสู่ระบบ ทำให้สามารถเข้าถึงที่เก็บข้อมูลที่เป็นส่วนตัว (Private) ที่ต้องการการยืนยันตัวตนได้



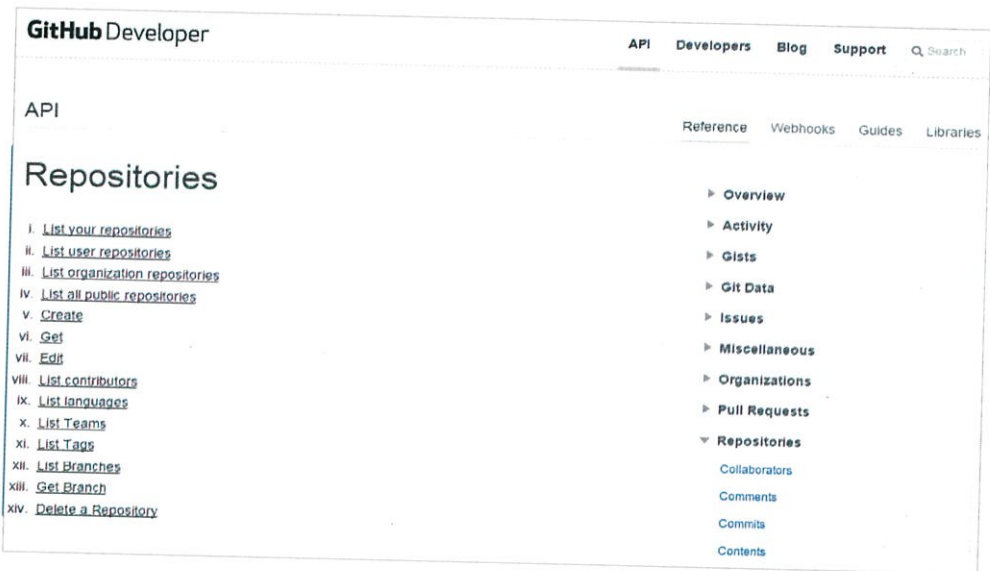
รูปที่ 2.2-1 แสดง Client ID และ Client Secret ที่ได้รับจากการลงทะเบียน



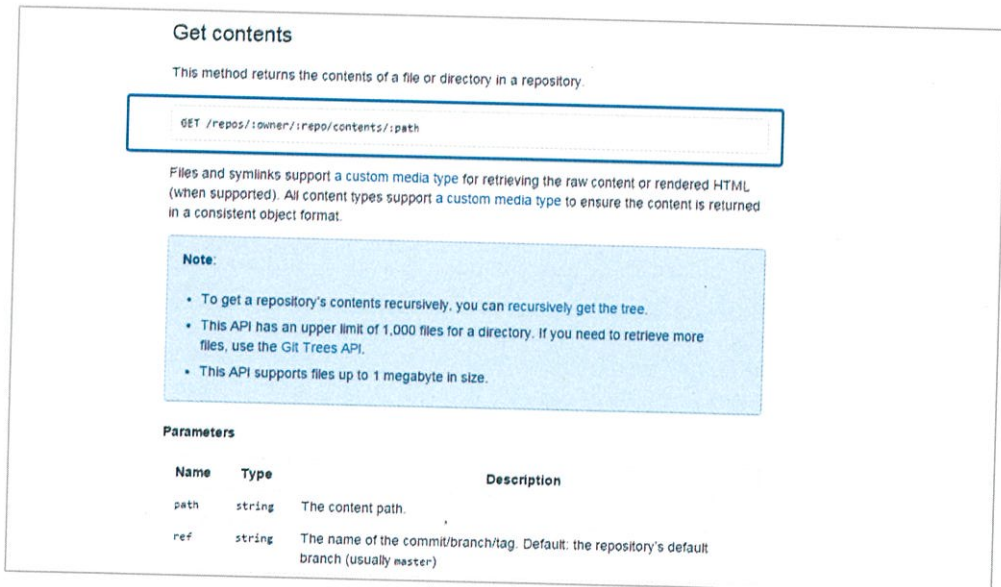
รูปที่ 2.2-2 การใช้ Client ID และ Client Key ในการเข้าสู่ระบบ Github จากแอปพลิเคชันที่สร้างขึ้น

ในกรณีที่ผู้ใช้ไม่ต้องการเข้าสู่ระบบ Github (ไม่มีการระบุตัวตน) สามารถข้ามขั้นตอนนี้ได้ ซึ่งจะทำให้ผู้ใช้สามารถเข้าถึงที่เก็บข้อมูลแค่เพียงแบบสาธารณะ (Public) เท่านั้น และไม่สามารถดำเนินการใดๆ ที่ต้องการการระบุตัวตนหรือการเข้าถึงที่เก็บข้อมูลที่เป็นส่วนตัว (Private) ได้

2. ทำการตรวจสอบ API ที่ Github ได้เตรียมไว้สำหรับข้อมูลที่ต้องการร้องขอโดย ทำการตรวจสอบได้จากเว็บไซต์ของ Github ซึ่งระบุรายละเอียดและวิธีการใช้งานของ API นั้นๆ จากรูปที่ 2.3 จะเห็นได้ว่าในกรอบที่ 1 ได้มีการจัดหมวดของ API ซึ่งเป็นไปตามหัวข้อการดำเนินการต่างๆ ที่ผู้ใช้สามารถทำได้ ซึ่งในกรอบที่ 2 เป็น API ในหมวดหมู่ของที่เก็บข้อมูล (Repository) โดยมีการแสดงรายการ API เรียงตามการดำเนินการที่สามารถทำได้ ตัวอย่างเช่น การรับข้อมูลรายการของที่เก็บข้อมูลที่เป็นสาธารณะ การรับรายการไฟล์ในที่เก็บข้อมูล เป็นต้น



รูปที่ 2.3 รายการของ API ที่เกี่ยวกับที่เก็บข้อมูล (Repository)



รูปที่ 2.4 ตัวอย่างการอธิบาย API ของ Github

จากรูปที่ 2.4 เป็นการอธิบายการใช้ API สำหรับการเข้าถึงข้อมูลในที่เก็บข้อมูล โดยเป็นการรับรายการไฟล์ที่อยู่ในที่เก็บข้อมูลนั้น (Get Contents) โดยมีการใช้ API ที่แสดงในกรอบคือ GET /repos/:owner/:repo/contents/:path ซึ่งมี GET เป็นชนิดของเมธอดการรับข้อมูล repos เป็นการระบุเพื่อให้ทราบว่าเป็นการเข้าถึงที่เก็บข้อมูล :owner เป็นการระบุชื่อผู้ใช้ที่เป็นเจ้าของที่เก็บข้อมูล :repo เป็นระบุชื่อของที่เก็บข้อมูล contents เป็นการระบุเพื่อให้ทราบว่าเป็นการเข้าถึงรายการไฟล์ในที่เก็บข้อมูล :path เป็นการระบุที่อยู่ภายในที่เก็บข้อมูลที่ต้องการรายชื่อไฟล์ ตัวอย่าง เช่น GET /repos/federicson/specialproblem/contents เป็นต้น

3. ดำเนินการติดต่อกับ Github โดย API สำหรับการดำเนินการที่ต้องการ จากนั้นทำการเขียนโปรแกรมเพื่อติดต่อขอข้อมูลและแสดงผลข้อมูลที่ได้จาก Github โดยใช้จาวาสคริปต์ (JavaScript) ตัวอย่างโปรแกรมแสดงดังรูปที่ 2.5

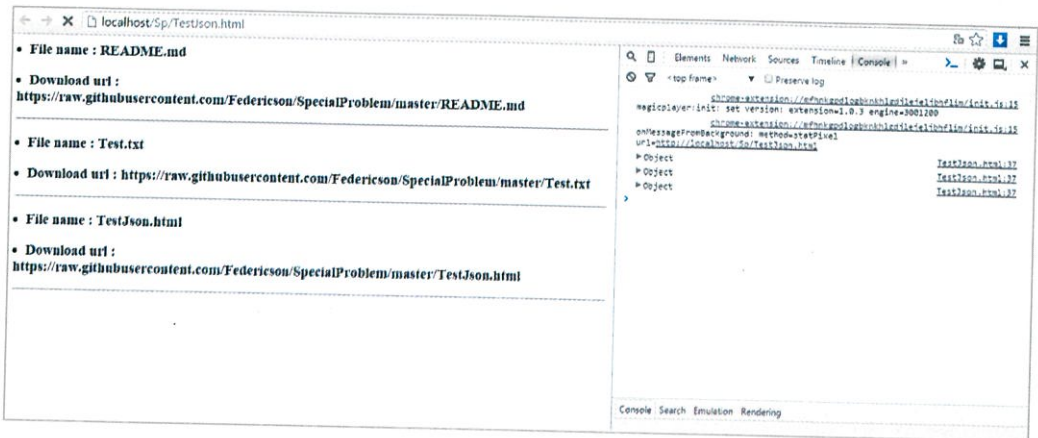
```

1 <html>
2 <head><title> Test Json Ajax</title>
3 </head>
4 <body>
5     <h1>Federicson 's repository</h1>
6
7     <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>
8     <script>
9         $.ajax({
10
11             url : 'https://api.github.com/repos/federicson/specialproblem/contents',
12             dataType : 'jsonp',
13             type : 'get',
14             cache: false,
15             success: function (data){
16
17                 $.each(data.data , function(key,value){
18                     document.write("<h3><li>"+"File name : "+value.name+"</li></h3>");
19                     document.write("<h3><li>"+"Download url : "+value.download_url+"</li></h3>");
20                     document.write("<hr />");
21                     console.log(value);
22                 });
23             }
24         });
25     });
26 </script>
27 </body>
28 </html>

```

รูปที่ 2.5 ตัวอย่างโปรแกรมเพื่อขอรายการไฟล์ในที่เก็บไฟล์ จาก Github

จากรูปที่ 2.5 การใช้ API ในบรรทัดที่ 11 ซึ่งเป็นการขอข้อมูลรายการไฟล์ที่เก็บไฟล์ <https://api.github.com/repos/federicson/specialproblem/contents> ซึ่งเป็นของผู้ใช้ที่ชื่อ federicson โดยในบรรทัดที่ 12 เป็นการกำหนดชนิดของข้อมูลซึ่งเป็นการกำหนดเป็น JSON เมื่อได้ข้อมูลในรูปแบบ JSON จะถูกนำมาใช้งานเพื่อแสดงผลในฟังก์ชันในบรรทัดที่ 15 โดยผลลัพธ์ของโปรแกรมแสดงดังรูปที่ 2.6



รูปที่ 2.6 ผลลัพธ์โปรแกรมเพื่อรับรายการไฟล์ในที่เก็บข้อมูลจาก Github

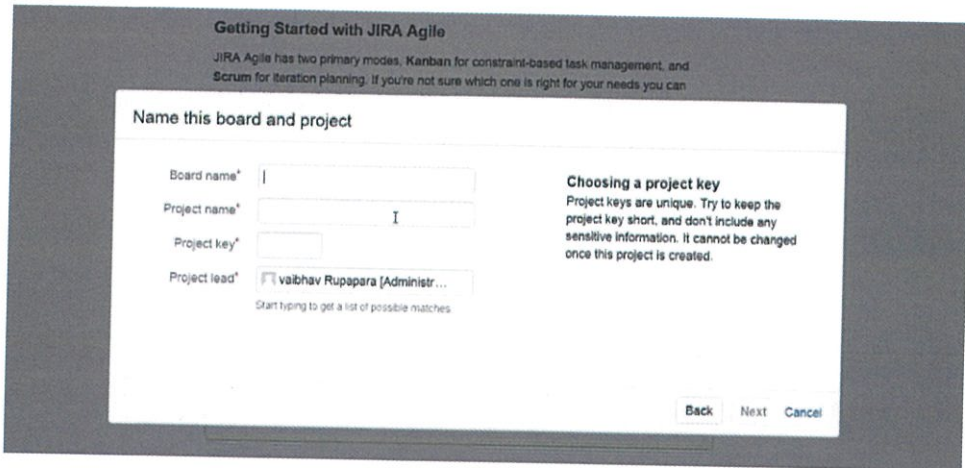
จากรูปที่ 2.6 ภายในกรอบที่ 1 เป็นการแสดงถึงข้อมูล JSON ที่ได้รับกลับมาจากเว็บไซต์ Github ซึ่งมีลักษณะเป็นข้อมูลเป็น JSON ออบเจ็ค โดยมีทั้งหมด 3 ออบเจ็คซึ่งหมายถึงจำนวนไฟล์ในที่เก็บข้อมูลนั้นมีไฟล์ทั้งหมด 3 ไฟล์และภายในออบเจ็คประกอบด้วยข้อมูลที่เกี่ยวข้องกับไฟล์นั้นๆ ซึ่งสามารถนำมาแสดงได้ผ่านทางหน้าบราวเซอร์ ตัวอย่างในกรอบที่ 2 มีการนำชื่อไฟล์และ URL ในการดาวน์โหลดไฟล์นั้นขึ้นมาแสดง เป็นต้น

ในปัจจุบันมีซอฟต์แวร์ที่เป็นเครื่องมือสำหรับอำนวยความสะดวกในการพัฒนาซอฟต์แวร์ในรูปแบบอไจล์ โดยมีทั้งแบบที่มีที่เก็บข้อมูลและไม่มีที่เก็บข้อมูล ซึ่งสามารถอำนวยความสะดวกให้กับผู้พัฒนาซอฟต์แวร์ในรูปแบบอไจล์

2.3 ตัวอย่างระบบที่รองรับการพัฒนาซอฟต์แวร์แบบอไจล์ในรูปแบบของสกรีม

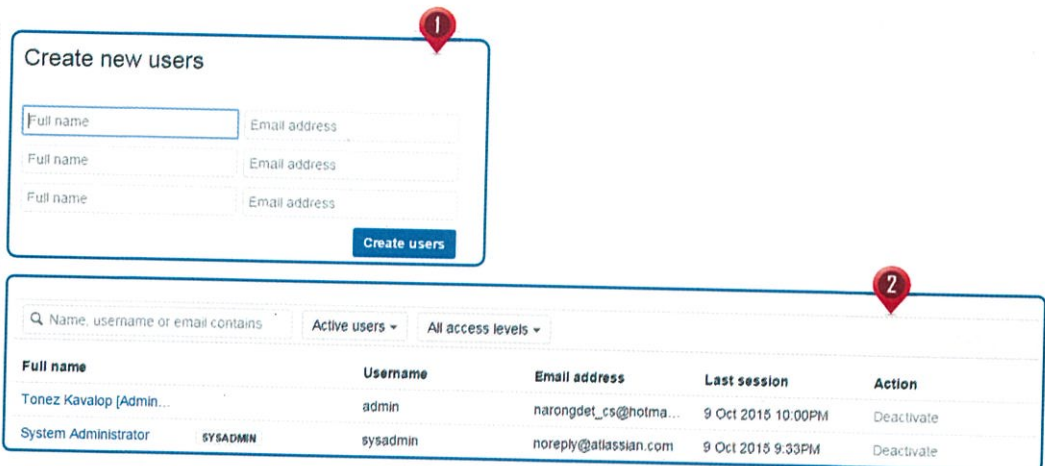
2.3.1 Jira Agile Tool

เป็นเครื่องมือสำหรับการพัฒนาซอฟต์แวร์ในรูปแบบอไจล์ซึ่งมีลักษณะเป็นเว็บแอปพลิเคชัน ถูกพัฒนาโดยบริษัท Atlassian โดยสามารถโปรเจกต์และสร้างงานของการพัฒนาซอฟต์แวร์และมอบหมายงานให้กับสมาชิกในทีม โดยสามารถเพิ่มงานที่ได้สร้างขึ้นเข้าสู่รอบการทำงาน (Sprint) ตามหลักของอไจล์และติดตามสถานะของงานนั้นๆ ได้



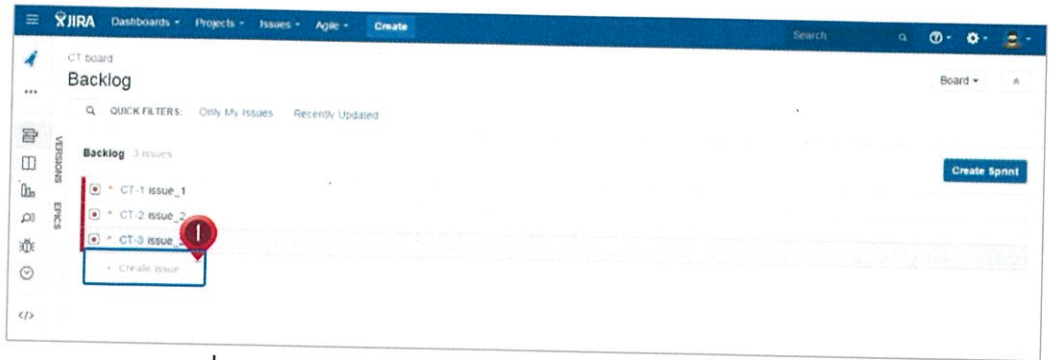
รูปที่ 2.7 หน้าจอสำหรับการบอร์ดและสร้างโปรเจกต์

จากรูปที่ 2.7 ผู้ใช้ทำการกรอกข้อมูลเพื่อสร้างบอร์ด โดยกรอกชื่อบอร์ดในช่อง Board name สร้างโปรเจกต์โดยการกรอกชื่อโปรเจกต์ในช่อง Project name สร้างรหัสของโปรเจกต์ในช่อง Project key และระบุผู้นำในการดำเนินโปรเจกต์ในช่อง Project lead



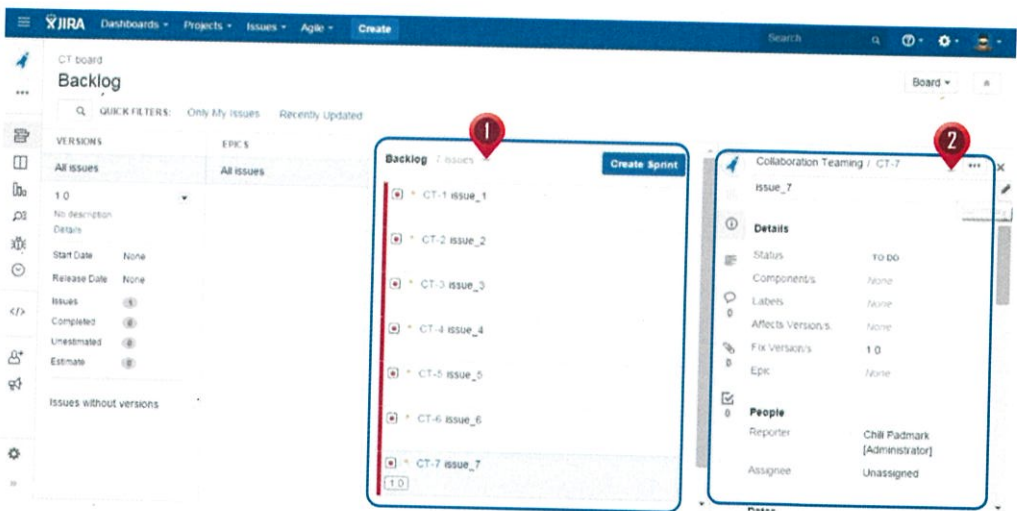
รูปที่ 2.8 หน้าจอสำหรับสร้างบัญชีผู้ใช้และแสดงรายชื่อผู้ใช้

จากรูปที่ 2.8 ในกรอบที่ 1 เป็นส่วนของการสร้างบัญชีผู้ใช้โดยทำการกรอกชื่อผู้ใช้ในช่อง Full name และกรอกอีเมลในช่อง Email address จากนั้นกดปุ่ม Create users ในกรอบที่ 2 เป็นการแสดงรายชื่อบัญชีผู้ใช้ที่มีอยู่ในระบบ



รูปที่ 2.9 หน้าจอสำหรับการสร้างงานและสร้างรอบการทำงาน

จากรูปที่ 2.9 ในกรอบที่ 1 ผู้ใช้สามารถสร้างงานโดยคลิกที่ Create issue โดยงานที่สร้างขึ้นจะถูกนำมาเรียงและจัดลำดับความสำคัญภายใน Backlog ซึ่งผู้ใช้สามารถให้รายละเอียดเกี่ยวกับงานนั้นๆ ได้ซึ่งแสดงในรูปที่ 2.10



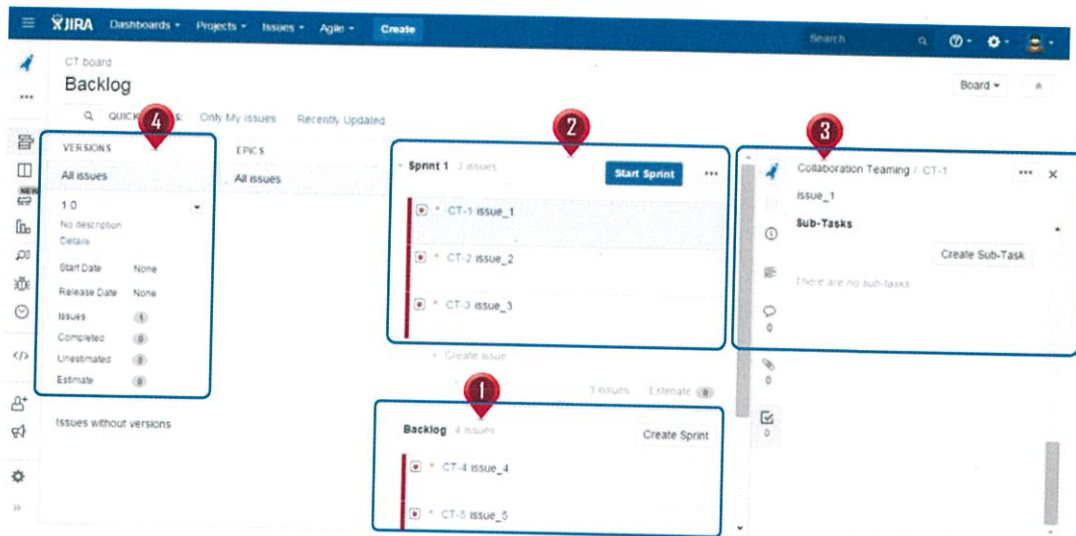
รูปที่ 2.10 หน้าจอแสดงรายการงานใน Backlog และรายละเอียดของงาน

จากรูปที่ 2.10 ในกรอบที่ 1 เป็นการแสดงรายการงานทั้งหมดใน Backlog ซึ่งผู้ใช้สามารถจัดลำดับความสำคัญในแต่ละงานโดยลำดับความสำคัญจากบนลงล่าง โดยในกรอบที่ 2 เป็นการแสดงรายละเอียดของงานหนึ่งงานซึ่งประกอบไปด้วยหัวข้อดังต่อไปนี้

- ชื่องาน (Name)
- รายละเอียด (Detail)
- บุคคลที่เกี่ยวข้อง (People) (ผู้สร้างงานและผู้ที่ได้รับมอบหมายงาน)
- วันที่สร้าง วันที่อัปเดต (Date)
- คำอธิบาย (Description)
- ความคิดเห็น (Comment)

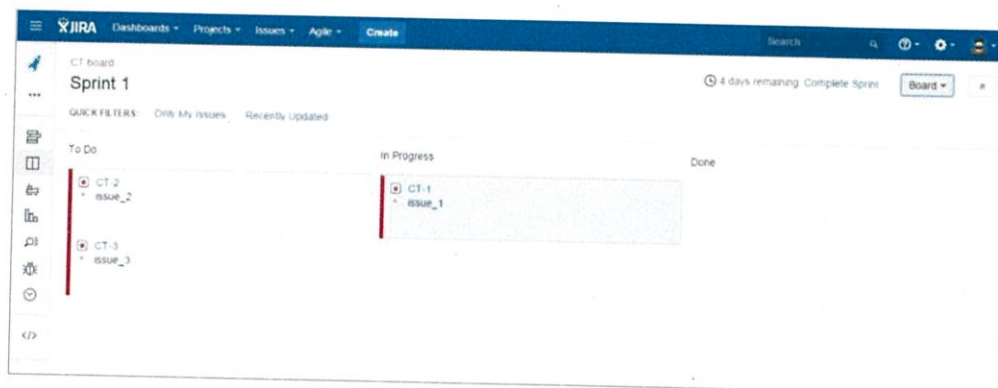
- ไฟล์แนบ (Attachment)

หลังจากที่ผู้ใช้งานสร้างงานขึ้นภายใน Backlog ผู้ใช้สามารถสร้างรอบการทำงาน (Sprint) ตามหลักการของไจล์และเพิ่มงานที่ได้สร้างขึ้นเข้าสู่รอบการทำงาน ซึ่งแสดงดังรูปที่ 2.11



รูปที่ 2.11 หน้าจอการสร้างรอบการทำงานและเพิ่มงานเข้าสู่รอบการทำงาน

จากรูปที่ 2.11 ในกรอบที่ 1 เป็นรายการงานที่ผู้ใช้งานสร้างขึ้นและมีปุ่มสำหรับสร้างรอบการทำงาน (Create Sprint) หลังจากที่ใช้กดปุ่มเพื่อสร้างรอบการทำงาน จะแสดงดังกรอบที่ 2 โดยผู้ใช้งานสามารถเพิ่มงานใน Backlog เข้าสู่รอบการทำงานได้โดยการลาก (Drag) จากนั้นผู้ใช้งานสามารถเริ่มการทำงานของรอบการทำงานนั้นโดยกดปุ่ม Start Sprint โดยจะหน้าจอสำหรับกำหนดวันเริ่มต้นทำงานและวันสิ้นสุดการทำงานของรอบการทำงานนั้นโดยแสดงดังกรอบที่ 3 โดยในกรอบที่ 4 เป็นการสร้างเวอร์ชันสำหรับกำหนดเวอร์ชันให้แก่ผลงาน โดยสามารถเพิ่มงานเข้าสู่เวอร์ชันที่มีการสร้างไว้



รูปที่ 2.12 หน้าจอแสดงสถานะของแต่ละงานในรอบการทำงาน

จากรูปที่ 2.12 เป็นการแสดงสถานะของงานแต่ละงานในรอบการทำงานนั้น เมื่อสามารถเปลี่ยนสถานะการทำงานเมื่อพัฒนางานนั้นได้ตามสถานะที่ถูกกำหนดไว้ โดยในภาพมีทั้งหมด 3 สถานะดังนี้

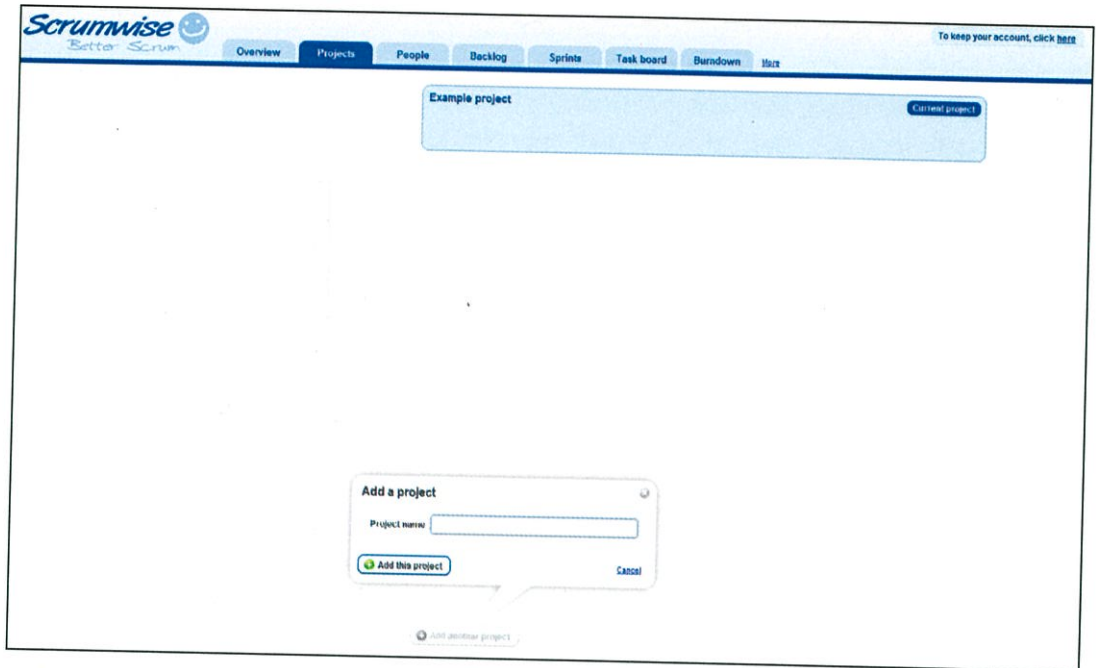
- To do หมายถึง งานที่จะพัฒนาซึ่งยังไม่มีกรเริ่มพัฒนา
- In Progress หมายถึง งานที่กำลังดำเนินการพัฒนา
- Done หมายถึง งานที่ได้ดำเนินการพัฒนาเสร็จเรียบร้อยแล้ว

โดยสรุปแล้วความสามารถของ jira agile tools ประกอบด้วยความสามารถในการจัดการระบบสมาชิกภายในทีม ความสามารถในการสร้างงานและให้รายละเอียดงาน ความสามารถในการจัดการและติดตามงานในรอบการทำงาน ซึ่งโดยรวม jira agile tools มีความสามารถที่ค่อนข้างครบถ้วนแต่ยังคงมีข้อจำกัดบางประการดังนี้

- ไม่สามารถแสดงความคืบหน้าของรอบการทำงานในรูปแบบของเปอร์เซ็นต์
- ไม่รองรับการติดตามงาน ระหว่างงานเวอร์ชันใหม่กับงานเวอร์ชันเก่า
- ไม่สามารถเก็บข้อมูลประวัติการแก้ไขงานระหว่างงานเวอร์ชันเก่าและงานเวอร์ชันใหม่
- เสียค่าบริการในการใช้งาน (ฟรีเฉพาะการทดลองใช้) โดยราคาขึ้นอยู่กับจำนวนผู้ใช้ภายในทีมพัฒนาซอฟต์แวร์

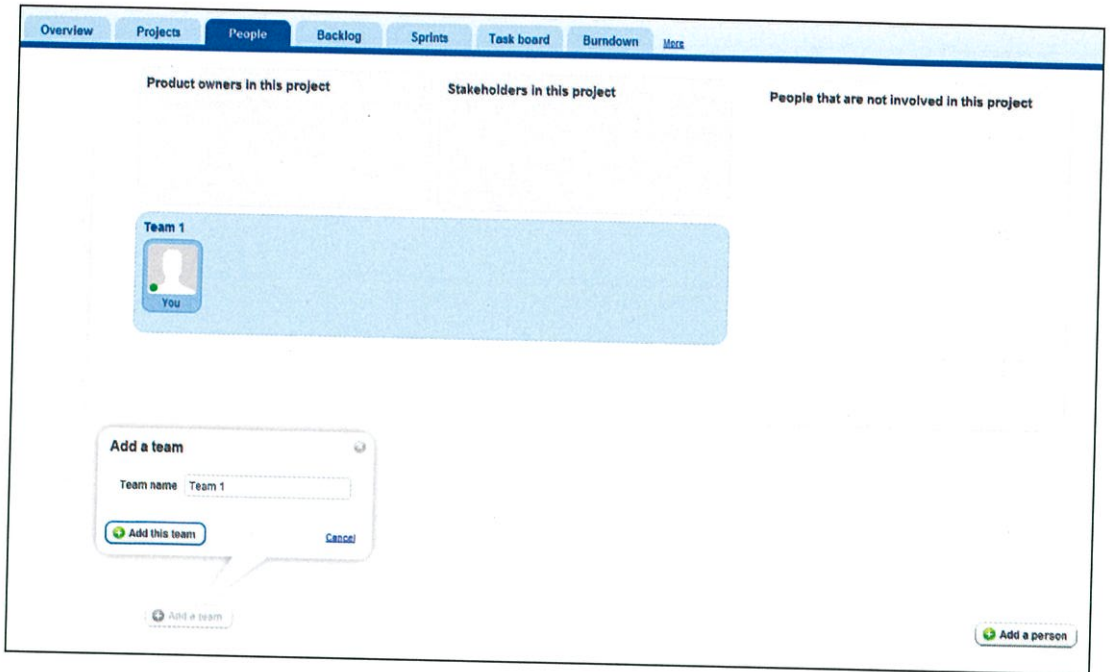
2.3.2 โปรแกรม Scrumwise

โปรแกรม Scrumwise ถูกสร้างขึ้นในปี ค.ศ. 2009 เป็นเครื่องมือที่ช่วยสนับสนุนการพัฒนาซอฟต์แวร์ในรูปแบบของ Scrum โดย Scrumwise มีลักษณะเด่น เน้นให้ผู้ใช้สามารถใช้งานได้ง่าย เป็นมิตรกับผู้ใช้ แต่ยังคงครอบคลุมทุกฟังก์ชันการทำงานที่จำเป็นต่อการทำงานในรูปแบบของ Scrum



รูปที่ 2.13 แสดงวิธีการสร้าง Project

จากรูปที่ 2.13 เป็นหน้าแรกของแอปพลิเคชัน สามารถสร้างหัวข้อโปรเจกต์ได้โดยการกด Add another project ที่ด้านล่างหน้าจอ และกรอกหัวข้อชื่อโปรเจกต์ที่ต้องการสร้าง

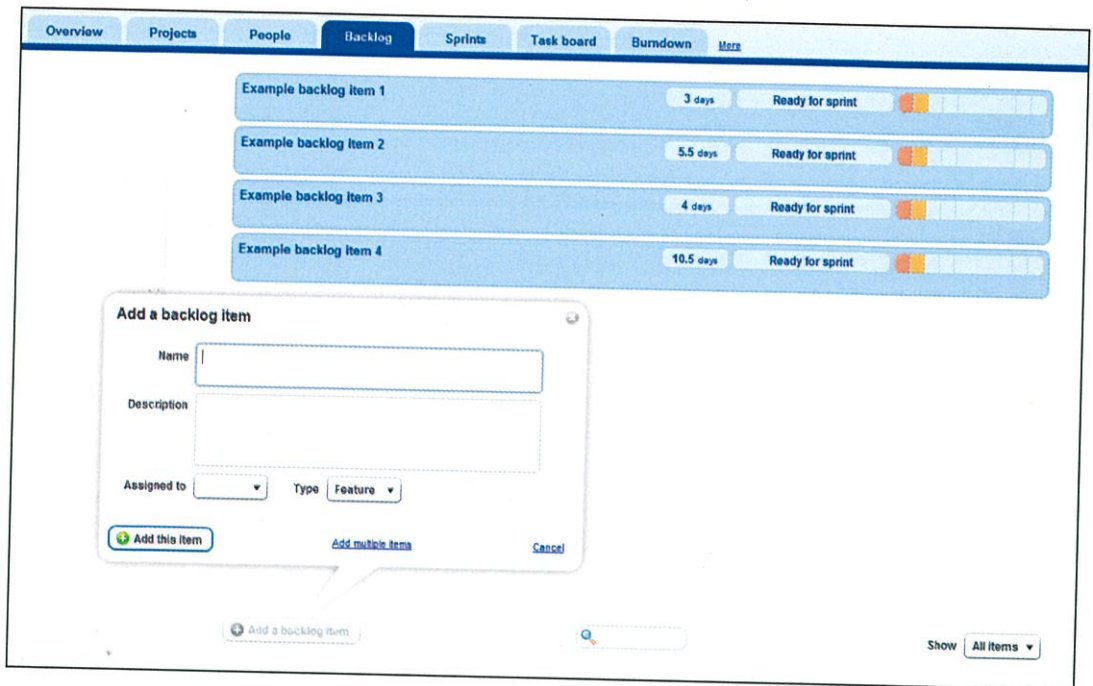


รูปที่ 2.14 แสดงบทบาทของสมาชิกในทีมและการสร้างทีม

ในหน้าของ People คือหน้าจัดการสมาชิก โดยมีสถานะเริ่มต้นดังนี้

- People that are not involved in this project ผู้ไม่เกี่ยวข้องกับโปรเจกต์นี้
- Product owners in this project เจ้าของผลิตภัณฑ์ของโปรเจกต์นี้
- Stakeholders in this project ผู้มีส่วนได้ส่วนเสียกับโปรเจกต์นี้

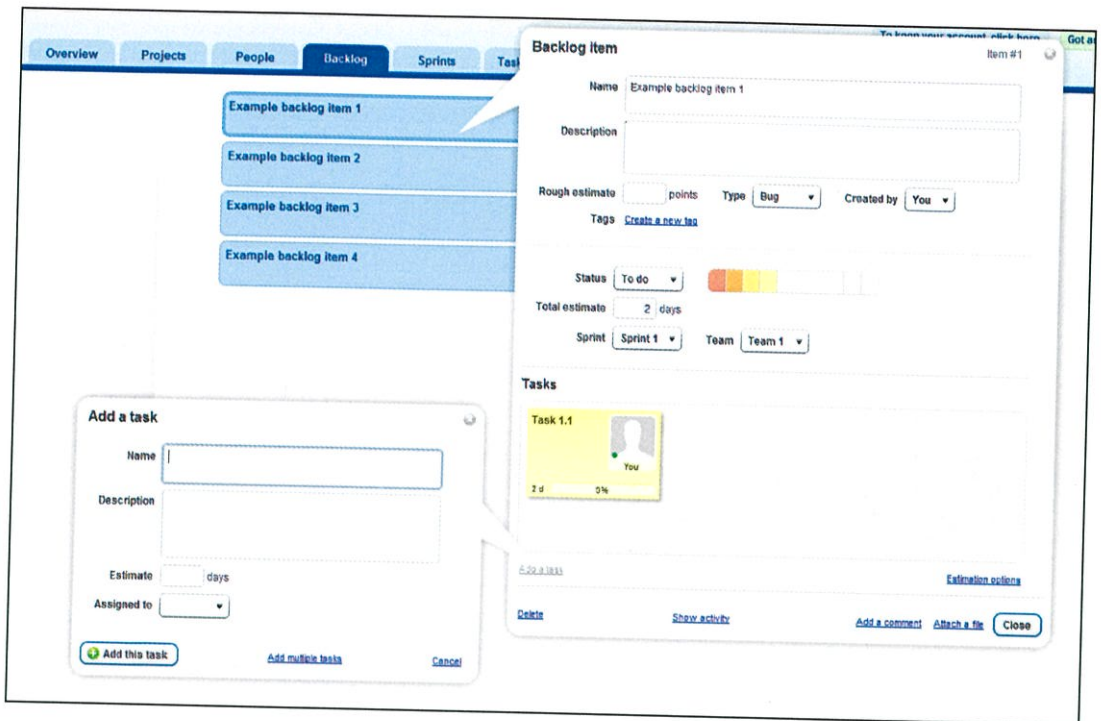
จากรูปที่ 2.14 ผู้ใช้สามารถสร้างทีมเพิ่มได้ โดยกด Add a team ที่ด้านล่างหน้าจอ แล้วกรอกชื่อทีมที่ต้องการสร้าง เช่น ผู้ใช้สร้างทีมโดยใช้ชื่อทีมว่า Team 1



รูปที่ 2.15 แสดงรายการใน Product Backlog และการสร้าง Backlog Item

จากรูปที่ 2.15 ในหน้าของ backlog แสดงรายการความต้องการ Backlog Item ที่อยู่ใน Product Backlog ทั้งหมด สามารถสร้าง Backlog Item เพิ่มเติมโดยกด Add a backlog item ที่ด้านล่างหน้าจอ โดยมีรายละเอียดให้กรอกดังนี้

- Name ชื่อรายการ Backlog Item
- Description ผู้ใช้สามารถอธิบายรายละเอียดของ Backlog Item เพิ่มเติมได้
- Assigned to กำหนดผู้รับผิดชอบ Backlog Item นี้
- Type กำหนดประเภทของ Backlog Item ได้ เช่น เป็นรายการฟีเจอร์ใหม่หรือเป็นการแก้บัค เป็นต้น

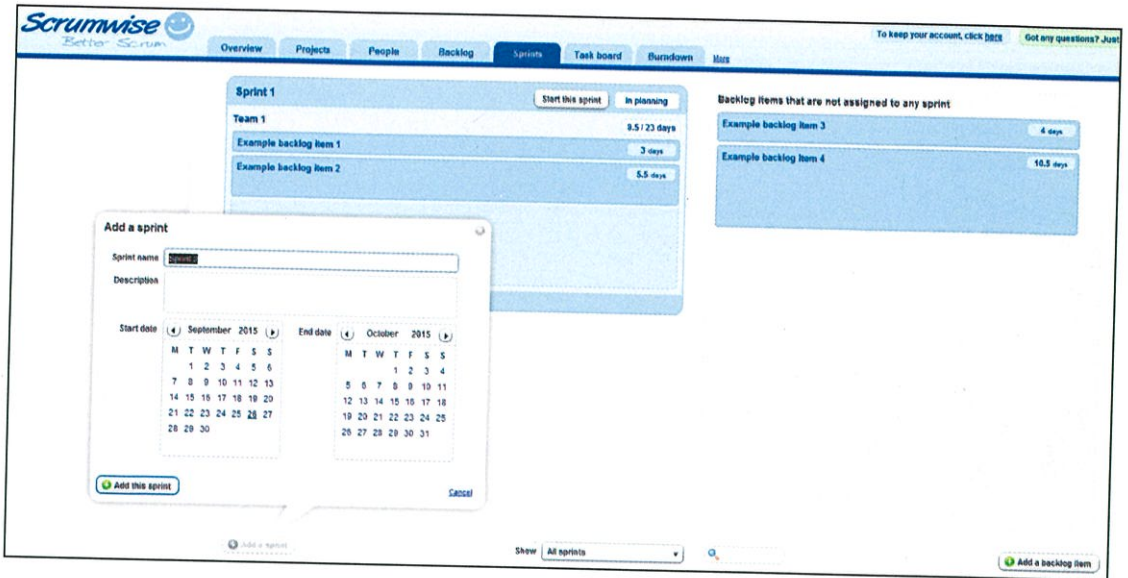


รูปที่ 2.16 แสดงรายละเอียดของ Backlog Item และการสร้าง Task งานใน Backlog Item

จากรูปที่ 2.16 แสดงรายละเอียดของ Backlog Item ดังนี้ ชื่อ Backlog Item คำอธิบายเพิ่มเติม ชนิดของรายการ Backlog Item ผู้สร้างรายการ Backlog Item นี้ โดยมีรายละเอียดเพิ่มเติมดังนี้

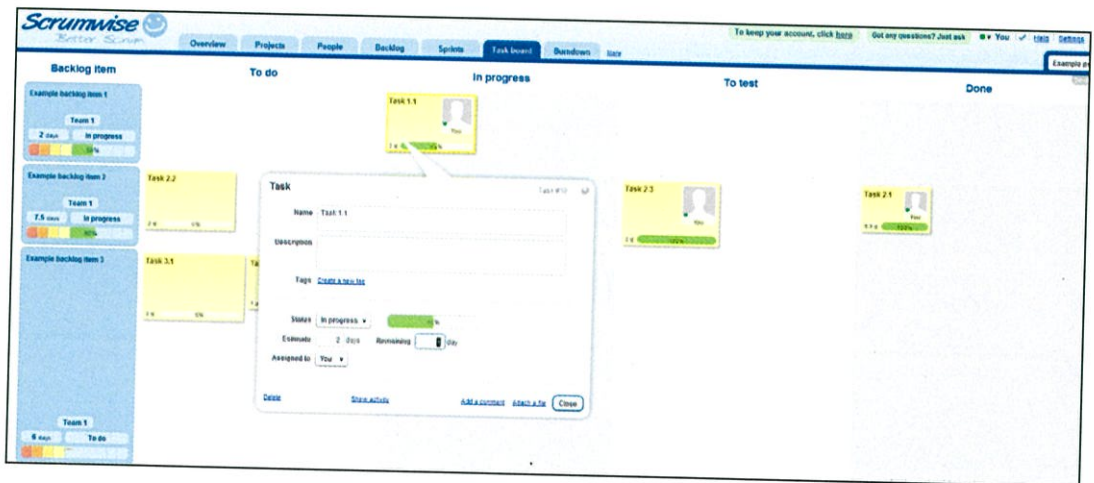
- Status บอกสถานะของรายการ Backlog Item เช่น เป็นรายการที่ยังไม่ทำ รายการที่จะทำ รายการที่กำลังทำ หรือรายการที่ทำเสร็จสิ้นแล้ว เป็นต้น
- Total estimate บอกระยะเวลารวมที่จะกระทำงานใน Backlog Item นั้นจนเสร็จ
- Sprint บอกถึง Backlog Item นี้ อยู่ในรอบ Sprint ไต
- Team บอกถึง Backlog Item นี้ อยู่ในความรับผิดชอบของทีมใด

สามารถสร้าง Task งานได้โดยกด Add a task แล้วกรอกชื่อ Task งานนั้น อธิบายงานนั้นเพิ่มเติม คาดคะเนระยะเวลาที่จะทำงานนั้นจนเสร็จ และกำหนดผู้รับผิดชอบงานนั้น



รูปที่ 2.17 แสดงการสร้าง Sprint และแสดงรายการ Backlog Item ในแต่ละรอบ Sprint

จากรูปที่ 2.17 ในหน้าของ Sprint สามารถสร้างรอบ Sprint ได้โดยกดที่ Add this sprint ที่ด้านล่างหน้าจอ สร้างชื่อ Sprint และกำหนดระยะเวลาในรอบ Sprint สามารถนำรายการ Backlog Item ที่อยู่ใน Product Backlog มาจัดวางในรอบ Sprint ได้



รูปที่ 2.18 แสดงสถานะ Task งานของแต่ละ Backlog Item บน Task board

จากรูปที่ 2.18 ในหน้าของ Task board แสดง Task งานของแต่ละ Backlog Item บน Task board ในรอบ Sprint นั้น โดยจะมีการแบ่งสถานะ ออกเป็น

- To do งานที่จะทำ
- In progress งานที่กำลังดำเนินการ
- To test งานที่กำลังทดสอบ
- Done งานที่ทำเสร็จ

โดย Task งานแต่ละ Task จะถูกกำหนดสถานะเริ่มต้นที่ To do และเปลี่ยนสถานะไปตามการดำเนินงาน จน Task งานนั้นเสร็จสิ้นเมื่ออยู่ในสถานะ Done โดยแต่ละ Task มีรายละเอียดเพิ่มขึ้น คือ

- Remaining คือ ระยะเวลาที่เหลืออยู่ของ Task งานนั้น
- Add a comment สามารถแสดงความคิดเห็นได้
- Attach a file สามารถแนบไฟล์งานได้

โดยสรุปแล้วความสามารถของ สามารถสร้างหัวข้อโปรเจกต์ พร้อมสร้างระบบสมาชิก ระบบบทบาทของสมาชิกที่เกี่ยวข้องกับโปรเจกต์ สร้างรายการโปรดักแบ็กล็อก และสร้างงานย่อยในแต่ละรายการโปรดักแบ็กล็อก สร้างรอบการทำงานและระบุงานที่จะทำในแต่ละรอบการทำงานได้ สามารถแสดงสถานะของงานในแต่ละรอบการทำงานได้ในรูปแบบ สกรัมบอร์ด สามารถสร้างสถานะงานเพิ่มเติมบนสกรัมบอร์ด ซึ่งโดยรวม scrumwise มีความสามารถที่ค่อนข้างครบถ้วนแต่ยังคงมีข้อจำกัดบางประการดังนี้

- ไม่สามารถควบคุมเวอร์ชันของงานได้
- ไม่สามารถเชื่อมต่อกับแหล่งเก็บข้อมูล (Repository) ภายนอกได้
- ไม่สามารถแสดงความคืบหน้าของรอบการทำงานในรูปแบบเปอร์เซ็นต์
- เสียค่าบริการในการใช้งาน (พิเศษเฉพาะการทดลองใช้) โดยราคาอยู่ที่ \$9 ต่อผู้ใช้งาน 1 คน

ระบบที่เกี่ยวข้อง	Jira Agile Tool	Scrumwise	Application for Supporting Agile Process (ระบบที่พัฒนาขึ้น)
ความสามารถ			
เว็บแอปพลิเคชัน	/	/	/
ระบบสมาชิก	/	/	/
สร้างงานและให้รายละเอียดงาน	/	/	/
มอบหมายงานให้กับสมาชิก	/	/	/
สร้างรอบการทำงาน	/	/	/
การคำนวณจำนวนวันของรอบการทำงาน	X	X	/
เพิ่มงานเข้าสู่รอบการทำงาน	/	/	/
เปลี่ยนสถานะงานในรอบการทำงาน	/	/	/
เพิ่มความคิดเห็น (Comment) ในแต่ละงาน	/	/	/
การแนบไฟล์งาน	/	/	/
การติดต่อกับที่เก็บข้อมูลภายนอก	/	X	/
การควบคุมเวอร์ชันของงาน	/	X	/
ระบุความคืบหน้าของรอบการทำงานในรูปแบบของเปอร์เซ็นต์	X	X	/

การติดตามงานระหว่างเวอร์ชันเก่า กับงานเวอร์ชันใหม่	X	X	/
เก็บข้อมูลประวัติการแก้ไขระหว่าง งานเวอร์ชันเก่ากับเวอร์ชันใหม่	X	X	/
เก็บข้อมูลประวัติความคืบหน้าของ งานบนรอบการทำงาน	X	X	/

หมายเหตุ / หมายถึง สามารถทำได้ X หมายถึง ไม่สามารถทำได้

ตารางที่ 2.1 ตารางสรุปลักษณะการทำงานของระบบที่เกี่ยวข้อง

บทที่ 3

การวิเคราะห์และออกแบบระบบ

แอปพลิเคชันสำหรับสนับสนุนการพัฒนาแบบอเจิล พัฒนาขึ้นมาเพื่อให้ทีมพัฒนาซอฟต์แวร์ที่ใช้กระบวนการแบบอเจิลมีความสะดวกมากยิ่งขึ้นในการติดตามงานต่างๆ ในการพัฒนาซอฟต์แวร์

3.1 ขอบเขตความสามารถของระบบ

3.1.1 ความสามารถโดยรวมของระบบ

ระบบสามารถจัดการสมาชิกภายในทีมและบริหารจัดการงานตามรูปแบบอเจิลในส่วนของ การสร้างงาน (Issues) การสร้างรอบการทำงาน (Sprint) การบริการจัดการงานในรอบการทำงาน นั้นๆ รวมทั้งการควบคุมเวอร์ชันของงาน การแสดงความคิดเห็นเกี่ยวกับงาน และการเรียกดูไฟล์ หรือซอร์สโค้ดที่เกี่ยวข้องกับงานจากแหล่งที่เก็บข้อมูล

3.1.2 Functional Requirements

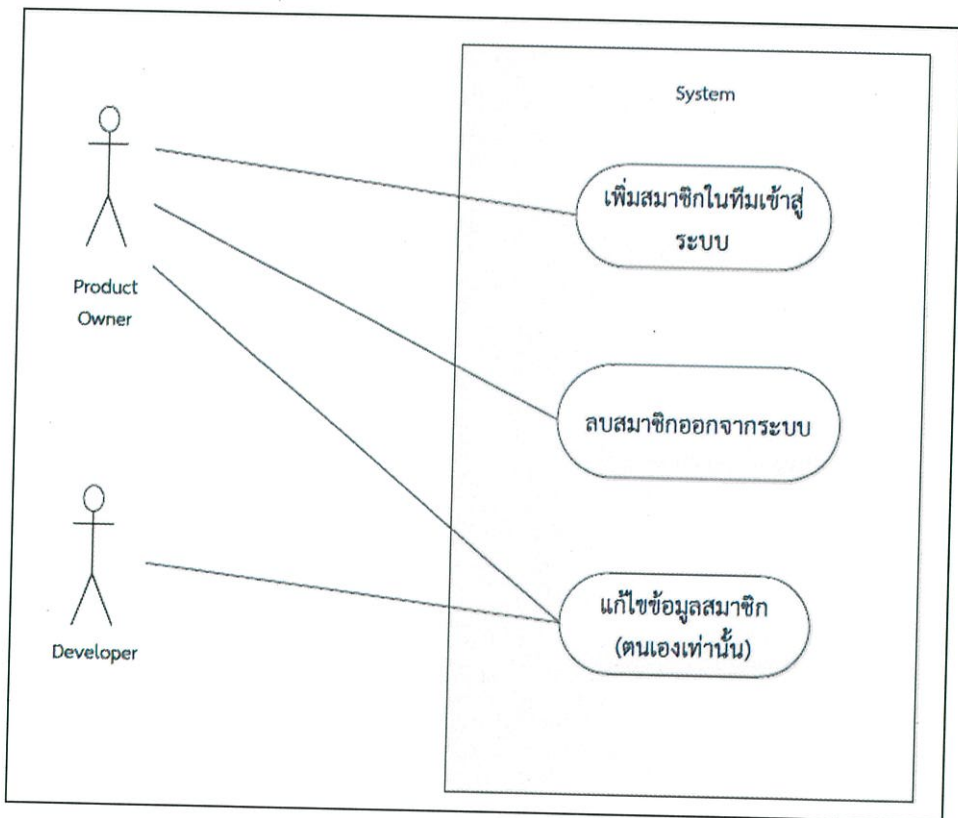
1. ผู้ใช้งานสามารถจัดการสมาชิกภายในทีม โดยสามารถเพิ่มสมาชิกเข้าสู่ระบบ ลบสมาชิก ออกจากระบบ แก้ไขข้อมูลสมาชิกได้
2. ผู้ใช้สามารถจัดการโปรเจกต์ โดยสามารถสร้างโปรเจกต์ สามารถเพิ่มสมาชิกเข้าสู่โปรเจกต์ ลบสมาชิกออกจากโปรเจกต์ได้
3. ผู้ใช้สามารถจัดการงาน โดยสามารถสร้าง Issue ซึ่งแบ่งเป็น 2 ประเภทคืองาน (Task) และงานย่อย (Sub-Task) โดยสามารถให้รายละเอียดต่างๆ ประกอบไปด้วย ชื่องาน ผู้รับผิดชอบ คำอธิบาย จำนวนวันที่คาดว่าจะเสร็จและไฟล์แนบได้
4. ผู้ใช้สามารถสร้างรอบการทำงาน โดยสามารถเพิ่มงานเข้าสู่รอบการทำงาน สามารถจัดการสถานะงานในรอบการทำงาน ซึ่งผู้ใช้สามารถเพิ่มหรือลบงานออกจากรอบการทำงาน โดยระบบสามารถคำนวณจำนวนวันของรอบการทำงานได้
5. ระบบสามารถติดตามเวอร์ชันงานได้ โดยผู้ใช้สามารถเรียกดูข้อมูลประวัติการเปลี่ยนแปลงของแต่ละเวอร์ชันได้
6. ผู้ใช้สามารถแสดงความคิดเห็นเกี่ยวกับงานและงานย่อยได้
7. ผู้ใช้สามารถเรียกดูไฟล์ที่เกี่ยวข้องกับงานซึ่งเป็นแหล่งที่เก็บข้อมูลของ Github โดยสามารถเรียกดูรายการไฟล์ ซอร์สโค้ดได้

8. ระบบสามารถบันทึกการเปลี่ยนสถานะของงานย่อยและสามารถบันทึกข้อมูลเกี่ยวกับการดำเนินการที่เกิดขึ้นกับงานย่อยนั้นๆ ในแต่ละสถานะได้
9. ระบบสามารถข้อมูลของแต่ละโปรเจกต์โดยสามารถแสดงข้อมูล จำนวนสมาชิก จำนวนงาน จำนวนงานย่อย จำนวนรอบการทำงานและแสดงข้อมูลกราฟของแต่ละรอบการทำงาน (Burndown Chart) ได้

3.2 การวิเคราะห์ระบบ

การวิเคราะห์ระบบประกอบไปด้วย Use Case Diagram โดยแสดงการวิเคราะห์ในแต่ละฟังก์ชันของระบบดังนี้

3.2.1 สมาชิก



รูปที่ 3.1 แสดง Use Case Diagram สมาชิก

จากรูปที่ 3.1 เป็นการแสดงยูสเคสของระบบสมาชิก โดยรายละเอียดของยูสเคสแสดงในตารางตามลำดับต่อไปนี้

ตารางที่ 3.1 เพิ่มสมาชิกในทีมเข้าสู่ระบบ

หัวข้อ	คำอธิบาย
Use Case Name	เพิ่มสมาชิกในทีมเข้าสู่ระบบ
Brief Description	ทำการเพิ่มสมาชิกในทีมเพื่อให้สามารถใช้งานระบบ
Actor	Product Owner
Trigger	Product Owner เลือกเพิ่มสมาชิกเข้าสู่ระบบ
Preconditions	-
Postconditions	สมาชิกใหม่ในระบบ
Flow of Events	<ol style="list-style-type: none"> 1. Product Owner เลือกเพิ่มสมาชิกเข้าสู่ระบบ 2. Product Owner ทำการกรอกข้อมูลของสมาชิกใหม่ โดยประกอบด้วย ชื่อ อีเมลล์ และเลือกประเภทของสมาชิกจากนั้นคลิกบันทึก 3. ระบบทำการเพิ่มสมาชิกใหม่และแสดงผลการดำเนินการ

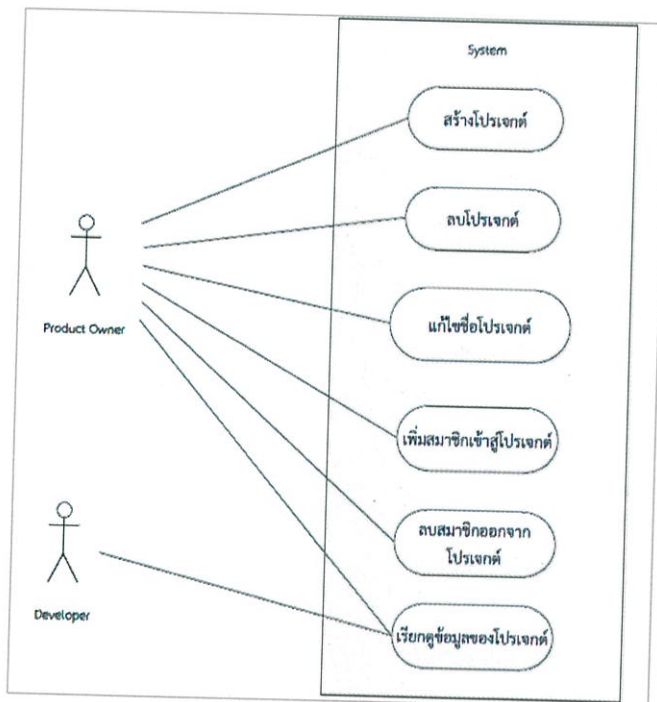
ตารางที่ 3.2 ลบสมาชิกออกจากระบบ

หัวข้อ	คำอธิบาย
Use Case Name	ลบสมาชิกออกจากระบบ
Brief Description	ทำการลบสมาชิกที่มีอยู่ในระบบออกจากระบบ
Actor	Product Owner
Trigger	Product Owner เลือกลบสมาชิกออกจากระบบ
Preconditions	-
Postconditions	ลบสมาชิก
Flow of Events	<ol style="list-style-type: none"> 1. Product Owner เลือกลบสมาชิกออกจากระบบ 2. Product Owner ทำการเลือกสมาชิกที่ต้องการลบ จากนั้นคลิกปุ่มลบ 3. ระบบทำการลบสมาชิกที่ถูกเลือกออกจากระบบ และแสดงผลการดำเนินการ

ตารางที่ 3.3 แก้ไขข้อมูลสมาชิก

หัวข้อ	คำอธิบาย
Use Case Name	แก้ไขข้อมูลสมาชิก
Brief Description	ทำการแก้ไขข้อมูลสมาชิก
Actor	ผู้ใช้
Trigger	ผู้ใช้เลือกแก้ไขข้อมูลสมาชิก
Preconditions	-
Postconditions	ข้อมูลสมาชิกถูกแก้ไข
Flow of Events	<ol style="list-style-type: none"> 1. ผู้ใช้เลือกแก้ไขข้อมูลสมาชิก 2. ผู้ใช้ทำการเลือกสมาชิกที่ต้องการแก้ไข จากนั้นคลิกปุ่มแก้ไข 3. ผู้ใช้แก้ไขข้อมูลสมาชิก ซึ่งสามารถแก้ไข ชื่อ อีเมลล์ และประเภทของสมาชิก จากนั้นคลิกปุ่มบันทึก 4. ระบบทำการบันทึกการแก้ไขและแจ้งผลการดำเนินการ

3.2.2 โปรเจกต์



รูปที่ 3.2 แสดง Use Case Diagram โปรเจกต์

จากรูปที่ 3.2 เป็นการแสดงยูสเคสของโปรเจกต์ โดยรายละเอียดของยูสเคสแสดง ใน ตารางตามลำดับต่อไปนี้

ตารางที่ 3.4 สร้างโปรเจกต์

หัวข้อ	คำอธิบาย
Use Case Name	สร้างโปรเจกต์
Brief Description	ทำการสร้างโปรเจกต์ใหม่
Actor	Product Owner
Trigger	Product Owner เลือกสร้างโปรเจกต์
Preconditions	-
Postconditions	โปรเจกต์ใหม่
Flow of Events	<ol style="list-style-type: none"> 1. Product Owner เลือกสร้างโปรเจกต์ 2. Product Owner ทำการกรอกชื่อของโปรเจกต์ จากนั้นคลิกบันทึก 3. ระบบทำการสร้างโปรเจกต์และแสดงผลการดำเนินการ

ตารางที่ 3.5 ลบโปรเจกต์

หัวข้อ	คำอธิบาย
Use Case Name	ลบโปรเจกต์
Brief Description	ทำการลบโปรเจกต์ที่อยู่ในระบบ
Actor	Product Owner
Trigger	Product Owner เลือกลบโปรเจกต์
Preconditions	-
Postconditions	ลบโปรเจกต์จากระบบ
Flow of Events	<ol style="list-style-type: none"> 1. Product Owner เลือกลบโปรเจกต์ 2. Product Owner ทำการเลือกโปรเจกต์ที่ต้องการลบ จากนั้นคลิกลบ 3. ระบบทำการลบโปรเจกต์และแสดงผลการดำเนินการ

ตารางที่ 3.6 แก้ไขชื่อโปรเจกต์

หัวข้อ	คำอธิบาย
Use Case Name	แก้ไขชื่อโปรเจกต์
Brief Description	ทำการแก้ไขชื่อโปรเจกต์
Actor	Product Owner
Trigger	Product Owner เลือกแก้ไขชื่อโปรเจกต์
Preconditions	-
Postconditions	โปรเจกต์ถูกแก้ไขชื่อ
Flow of Events	<ol style="list-style-type: none"> 1. Product Owner ทำการเลือกโปรเจกต์ที่ต้องแก้ไขชื่อ และทำการแก้ไขชื่อนั้นคลิกบันทึก 2. ระบบทำการบันทึกการแก้ไขและแสดงผล

ตารางที่ 3.7 เพิ่มสมาชิกเข้าสู่โปรเจกต์

หัวข้อ	คำอธิบาย
Use Case Name	เพิ่มสมาชิกเข้าสู่โปรเจกต์
Brief Description	ทำการเพิ่มสมาชิกเข้าสู่โปรเจกต์เพื่อพัฒนางานในโปรเจกต์
Actor	Product Owner
Trigger	Product Owner เลือกเพิ่มสมาชิกเข้าสู่โปรเจกต์
Preconditions	-
Postconditions	สมาชิกอยู่ในโปรเจกต์
Flow of Events	<ol style="list-style-type: none"> 1. Product Owner เลือกเพิ่มสมาชิกเข้าสู่โปรเจกต์ 2. Product Owner ทำการเลือกโปรเจกต์ที่ต้องการเพิ่มสมาชิก 3. ทำการเลือกสมาชิกที่ต้องการเพิ่มเข้าสู่โปรเจกต์ โดยระบบแสดงรายการสมาชิกทั้งหมด จากนั้นคลิกปุ่มเพิ่ม 4. ระบบทำการเพิ่มสมาชิกเข้าสู่โปรเจกต์ และแสดงผลการดำเนินการ

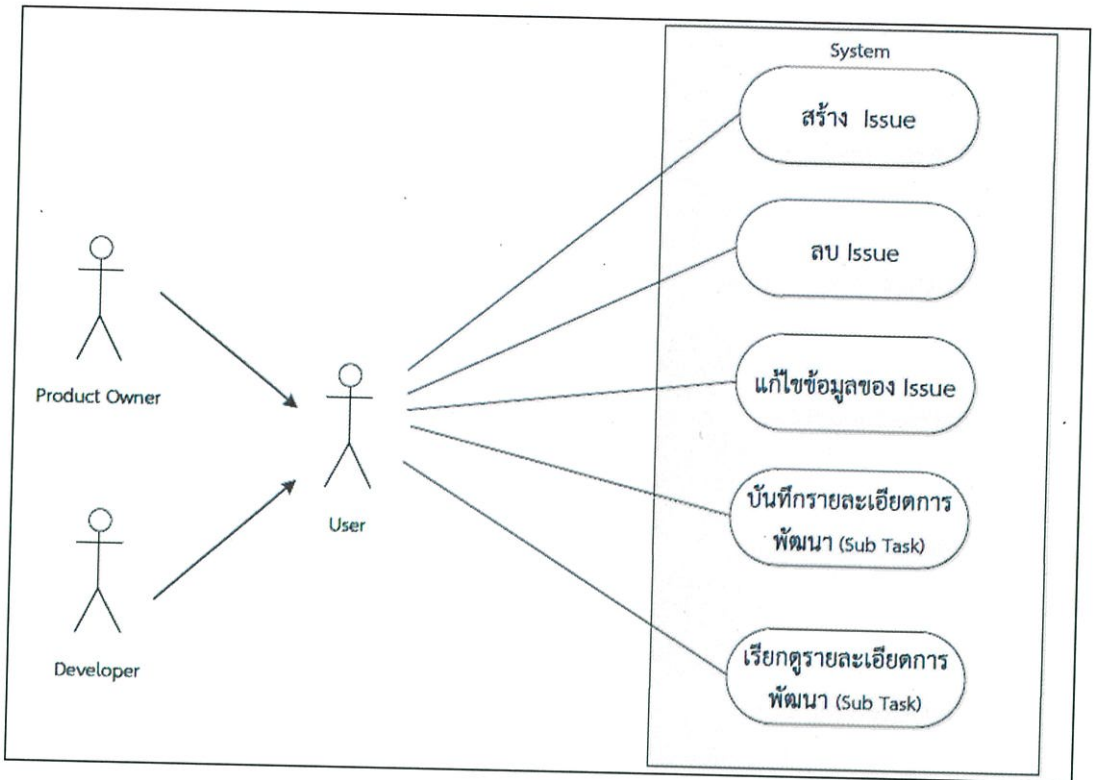
ตารางที่ 3.8 ลบสมาชิกออกจากโปรเจกต์

หัวข้อ	คำอธิบาย
Use Case Name	ลบสมาชิกออกจากโปรเจกต์
Brief Description	ทำการลบสมาชิกออกจากโปรเจกต์
Actor	Product Owner
Trigger	Product Owner เลือกลบสมาชิกออกจากโปรเจกต์
Preconditions	-
Postconditions	สมาชิกถูกลบจากโปรเจกต์
Flow of Events	<ol style="list-style-type: none"> 1. Product Owner เลือกดูรายการชื่อสมาชิกในโปรเจกต์ 2. Product Owner ทำการเลือกสมาชิกที่ต้องการลบจากนั้นคลิกปุ่มลบ 3. ระบบทำการลบสมาชิกออกจากโปรเจกต์และแสดงผล

ตารางที่ 3.9 เรียกดูข้อมูลของโปรเจกต์

หัวข้อ	คำอธิบาย
Use Case Name	เรียกดูข้อมูลของโปรเจกต์
Brief Description	เรียกดูข้อมูลของโปรเจกต์ โดยเป็นการสรุปข้อมูลของโปรเจกต์นั้นประกอบด้วย จำนวนสมาชิก จำนวนงาน จำนวนงานย่อย จำนวนรอบการทำงาน และแสดง Burndown Chart ของแต่ละรอบการทำงาน
Actor	Product Owner , Developer
Trigger	Product Owner , Developer เลือกเรียกดูข้อมูลของโปรเจกต์
Preconditions	-
Postconditions	ระบบแสดงข้อมูลของโปรเจกต์
Flow of Events	<ol style="list-style-type: none"> 1. Product Owner , Developer เลือกโปรเจกต์ที่ต้องการดูข้อมูล 2. ระบบแสดงข้อมูลของโปรเจกต์

3.2.3 การจัดการ Issue



รูปที่ 3.3 Use Case Diagram สำหรับการจัดการ Issue

จากรูปที่ 3.3 เป็นการแสดงยูสเคสของการจัดการ Issue โดย Issue มี 2 ประเภทคือ งานและงานย่อย โดยรายละเอียดของยูสเคสแสดงในตารางตามลำดับต่อไปนี้

ตารางที่ 3.10 สร้าง Issue

หัวข้อ	คำอธิบาย
Use Case Name	สร้าง Issue
Brief Description	ทำการสร้าง Issue โดยมีประเภทคืองานและงานย่อย
Actor	ผู้ใช้
Trigger	ผู้ใช้เลือกสร้างงานหรืองานย่อย
Preconditions	-
Postconditions	ได้ Issue ใหม่
Flow of Events	<ol style="list-style-type: none"> 1. ผู้ใช้เลือกสร้าง Issue โดยทำการเลือกคลิกที่สร้างงานหรืองานย่อย 2. ผู้ใช้ทำการกรอกข้อมูล ชื่อ คำอธิบาย จำนวนวันที่คาดว่าจะเสร็จ จากนั้นคลิกปุ่มสร้าง (ในกรณีการสร้างงานย่อยระบบจะทำการเชื่อมโยงงานย่อยกับงานหลักเข้าด้วยกันด้วย) 3. ระบบทำการสร้าง Issue และแสดงผลการดำเนินการ

ตารางที่ 3.11 ลบ Issue

หัวข้อ	คำอธิบาย
Use Case Name	ลบ Issue
Brief Description	ทำการลบ Issue ตามที่ผู้ใช้ต้องการ
Actor	ผู้ใช้
Trigger	ผู้ใช้เลือกลบ Issue
Preconditions	-
Postconditions	Issue ที่เลือกถูกลบ
Flow of Events	<ol style="list-style-type: none"> 1. ผู้ใช้เลือกลบ Issue 2. ผู้ใช้ทำการเลือก Issue ที่ต้องการลบ จากนั้นคลิกปุ่มลบ 3. ระบบทำการลบ Issue และแสดงผลการดำเนินการ

ตารางที่ 3.12 แก้ไขข้อมูลของ Issue

หัวข้อ	คำอธิบาย
Use Case Name	แก้ไขข้อมูลของ Issue
Brief Description	ทำการแก้ไขข้อมูลของ Issue เมื่อ Issue นั้นมีการเปลี่ยนแปลง
Actor	ผู้ใช้
Trigger	ผู้ใช้เลือกแก้ไขข้อมูลของ Issue
Preconditions	-
Postconditions	ข้อมูลของ Issue ที่แก้ไขถูกบันทึก
Flow of Events	<ol style="list-style-type: none"> 1. ผู้ใช้เลือกแก้ไข Issue 2. ผู้ใช้ทำการเลือก Issue ที่ต้องการแก้ไข จากนั้นทำการแก้ไข โดยสามารถแก้ไข ชื่อ คำอธิบาย จำนวนวันที่คาดว่าจะเสร็จ จากนั้นคลิกปุ่มบันทึก 3. ระบบทำการบันทึกข้อมูล Issue ที่ถูกแก้ไขและแสดงผลการดำเนินการ

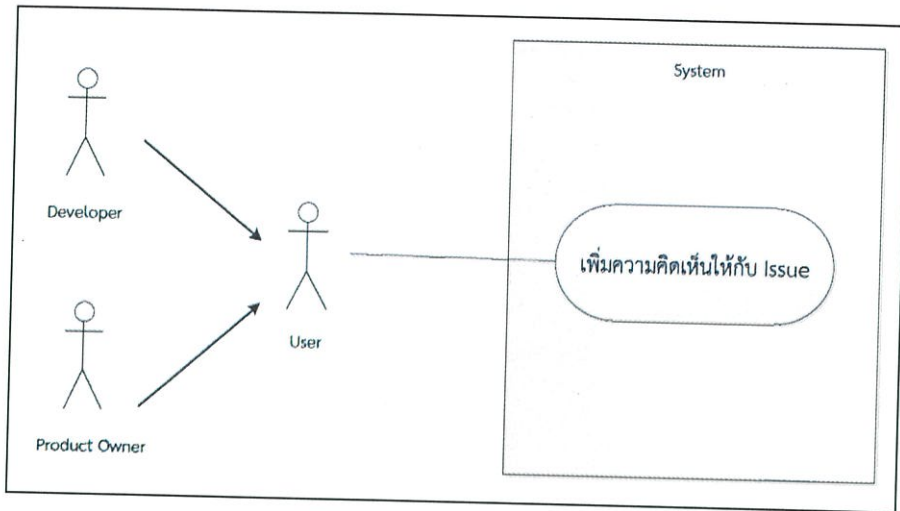
ตารางที่ 3.13 บันทึกรายละเอียดการพัฒนา (Sub Task)

หัวข้อ	คำอธิบาย
Use Case Name	บันทึกรายละเอียดการพัฒนา (Sub Task)
Brief Description	ทำการบันทึกรายละเอียดการพัฒนาที่ได้ดำเนินการกับงานย่อยนั้น
Actor	ผู้ใช้
Trigger	ผู้ใช้กรอกรายละเอียดการพัฒนาและบันทึก
Preconditions	-
Postconditions	บันทึกรายละเอียดการพัฒนา
Flow of Events	<ol style="list-style-type: none"> 1. ผู้ใช้เลือกงานย่อยที่ต้องการบันทึกรายละเอียดการพัฒนา 2. ผู้ใช้ทำการกรอกรายละเอียดการพัฒนาและทำการคลิกปุ่มบันทึก 3. ระบบทำการบันทึกรายละเอียดการพัฒนาของงานย่อยนั้น

ตารางที่ 3.14 เรียกดูรายละเอียดการพัฒนา (Sub Task)

หัวข้อ	คำอธิบาย
Use Case Name	เรียกดูรายละเอียดการพัฒนา (Sub Task)
Brief Description	ทำการเรียกดูรายละเอียดการพัฒนาของงานย่อยนั้นที่ได้มีการบันทึกไว้รวมถึงข้อมูลการเปลี่ยนสถานะของงานย่อยนั้น
Actor	ผู้ใช้
Trigger	ผู้ใช้เรียกดูรายละเอียดการพัฒนา
Preconditions	-
Postconditions	ระบบแสดงรายละเอียดการพัฒนาของงานย่อย
Flow of Events	<ol style="list-style-type: none"> 1. ผู้ใช้เลือกงานย่อยที่ต้องการเรียกดูรายละเอียดการพัฒนา 2. ระบบทำการแสดงรายละเอียดการพัฒนาและข้อมูลการเปลี่ยนสถานะของงานย่อย

3.2.4 เพิ่มความคิดเห็นให้กับ Issue



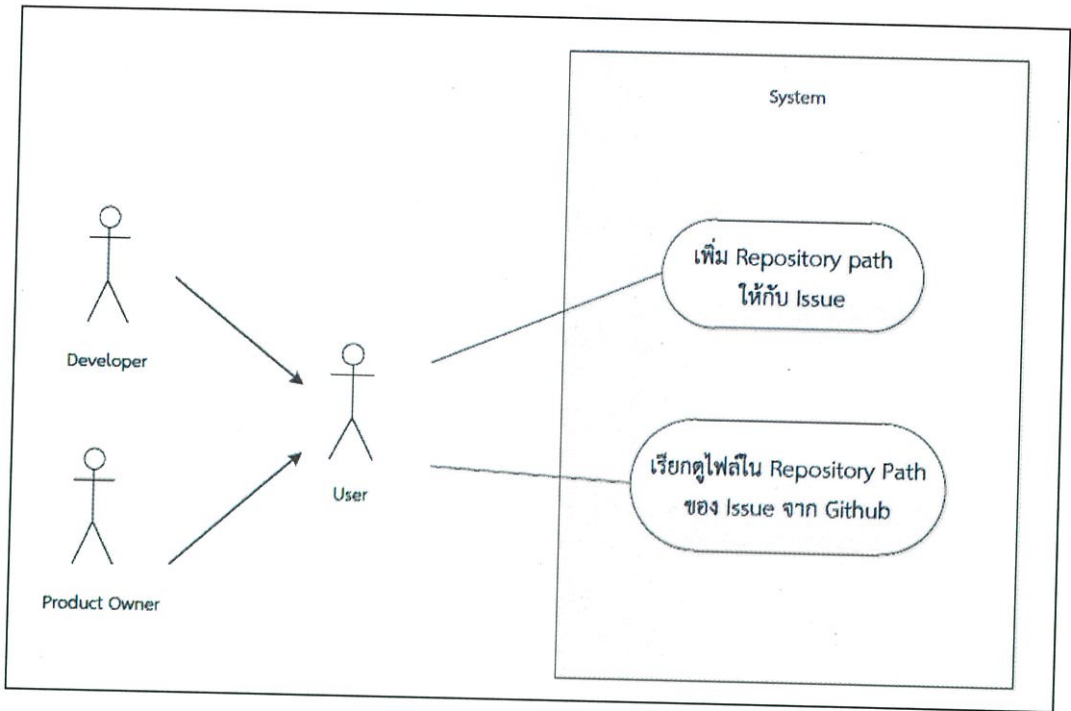
รูปที่ 3.4 Use Case Diagram สำหรับการแสดงความคิดเห็นของ Issue

จากรูปที่ 3.4 เป็นการแสดงยูสเคสการแสดงความคิดเห็นของ Issue โดยรายละเอียดของยูสเคสแสดงในตารางตามลำดับต่อไปนี้

ตารางที่ 3.15 เพิ่มความคิดเห็นให้กับ Issue

หัวข้อ	คำอธิบาย
Use Case Name	เพิ่มความคิดเห็นให้กับ Issue
Brief Description	ทำการเพิ่มความคิดเห็นของผู้ใช้ให้กับ Issue ซึ่งเป็นข้อมูลที่มีผลต่อ Issue
Actor	ผู้ใช้
Trigger	ผู้ใช้เลือกเพิ่มความคิดเห็นให้กับ Issue
Preconditions	-
Postconditions	เพิ่มความคิดเห็นที่ Issue
Flow of Events	<ol style="list-style-type: none"> 1. ผู้ใช้เลือก Issue ที่ต้องการแสดงความคิดเห็น 2. ผู้ใช้พิมพ์ความคิดเห็นลงในช่องแสดงความคิดเห็นของ Issue จากนั้นกดปุ่ม Enter 3. ระบบทำการบันทึกความคิดเห็นที่ Issue นั้น

3.3.5 Github Repository



รูปที่ 3.5 Use Case Diagram ของ Repository

จากรูปที่ 3.5 เป็นการแสดงยูสเคสของ Repository โดยรายละเอียดของยูสเคสแสดงในตารางตามลำดับต่อไปนี้

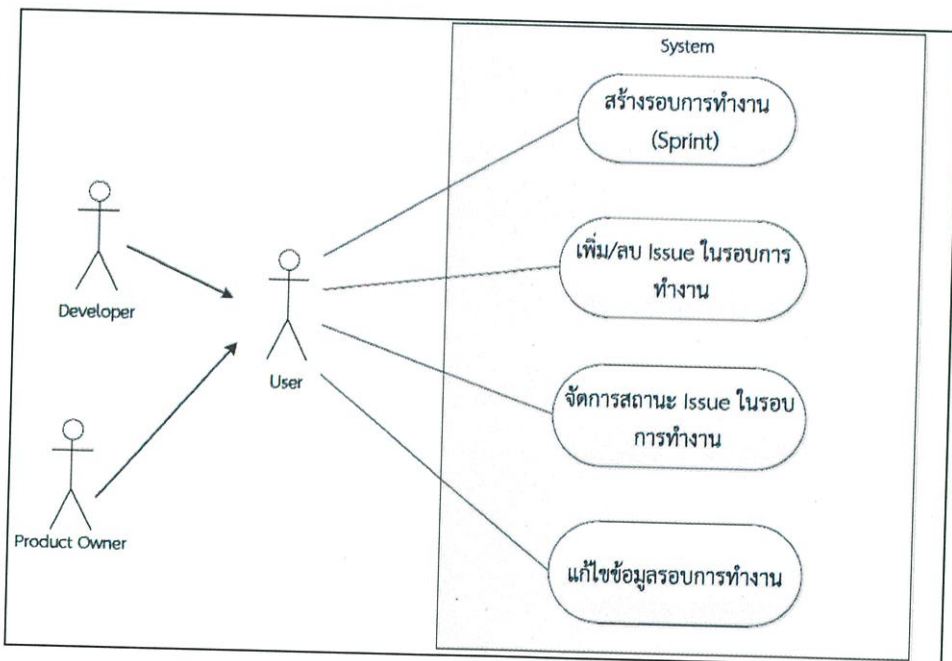
ตารางที่ 3.16 เพิ่ม Repository Path ให้กับ Issue

หัวข้อ	คำอธิบาย
Use Case Name	เพิ่ม Repository Path ให้กับ Issue
Brief Description	ทำการเพิ่ม Repository path ให้กับ Issue โดยเป็น path ที่อยู่ของไฟล์งานที่เกี่ยวข้องกับ Issue นั้น ซึ่งถูกเก็บอยู่บน Github (1 issue มีได้มากกว่า 1 Repository path)
Actor	ผู้ใช้
Trigger	ผู้ใช้เลือกเพิ่ม Repository Path ให้กับ Issue
Preconditions	-
Postconditions	Repository Path ที่ Issue
Flow of Events	<ol style="list-style-type: none"> 1. ผู้ใช้เลือก Issue ที่ต้องการเพิ่ม Repository Path 2. ผู้ใช้กรอก Repository Path ลงในช่องกรอกข้อมูล จากนั้นกดปุ่ม Enter 3. ระบบทำการเพิ่ม Repository Path ให้กับ Issue

ตารางที่ 3.17 เรียกดูไฟล์ใน Repository Path ของ Issue จาก Github

หัวข้อ	คำอธิบาย
Use Case Name	เรียกดูไฟล์ใน Repository Path ของ Issue จาก Github
Brief Description	เรียกดูไฟล์ใน Repository Path ของ Issue จาก Github โดยจะแสดงรายการไฟล์ใน path และแสดงรายละเอียดของไฟล์ เช่น ประเภทไฟล์ ลิ้งเพื่อเรียกดูซอร์สโค้ด เป็นต้น
Actor	ผู้ใช้
Trigger	ผู้ใช้เลือกเรียกดูไฟล์ใน Repository Path ของ Issue
Preconditions	เชื่อมต่ออินเทอร์เน็ต
Postconditions	รายการไฟล์ใน Repository Path ของ Issue
Flow of Events	<ol style="list-style-type: none"> 1. ผู้ใช้เลือก Issue ที่ต้องการดูไฟล์ใน Repository Path 2. ผู้ใช้เลือกคลิกที่ Repository Path ที่ต้องการ 3. ระบบทำการเชื่อมต่อไปยัง Github เพื่อขอรายการไฟล์ใน Repository Path นั้น 4. ระบบทำการแสดงผลรายการไฟล์ใน Repository Path นั้น (ผู้ใช้สามารถคลิกเพื่อดูซอร์สโค้ดหรือคลิกเพื่อไปที่เว็บไซต์ของ Github ได้)

3.3.6 รอบการทำงาน (Sprint)



รูปที่ 3.6 Use Case Diagram ของรอบการทำงาน (Sprint)

จากรูปที่ 3.6 เป็นการแสดงยูสเคสของรอบการทำงาน โดยรายละเอียดของยูสเคสแสดงในตารางตามลำดับต่อไปนี้

ตารางที่ 3.18 สร้างรอบการทำงาน (Sprint)

หัวข้อ	คำอธิบาย
Use Case Name	สร้างรอบการทำงาน (Sprint)
Brief Description	ทำการสร้างรอบการทำงานเพื่อนำ Issue เข้าสู่รอบการทำงาน
Actor	ผู้ใช้
Trigger	ผู้ใช้เลือกสร้างรอบการทำงาน
Preconditions	-
Postconditions	ได้รอบการทำงาน (Sprint)
Flow of Events	<ol style="list-style-type: none"> 1. ผู้ใช้เลือกสร้างรอบการทำงาน 2. ผู้ใช้กรอกข้อมูล ชื่อรอบการทำงาน วันเริ่มต้น วันสิ้นสุด จากนั้นคลิกปุ่มสร้าง 3. ระบบทำการสร้างรอบการทำงาน

ตารางที่ 3.19 เพิ่ม/ลบ Issue ในรอบการทำงาน

หัวข้อ	คำอธิบาย
Use Case Name	เพิ่ม/ลบ Issue ในรอบการทำงาน
Brief Description	สามารถทำการเพิ่มหรือลบ Issue ระหว่างรอบทำงานได้
Actor	ผู้ใช้
Trigger	ผู้ใช้เลือกเพิ่มหรือลบ Issue ในรอบการทำงาน
Preconditions	-
Postconditions	Issue ถูกเพิ่มหรือลบในรอบการทำงาน
Flow of Events	<ol style="list-style-type: none"> 1. ผู้ใช้เลือกรอบการทำงานที่ต้องการเพิ่มหรือลบ Issue 2. <ul style="list-style-type: none"> • ผู้ใช้เลือกเพิ่ม Issue ผู้ใช้คลิกปุ่มเพิ่ม Issue จากนั้นเลือก Issue ที่ต้องการเพิ่ม • ผู้ใช้เลือกลบ Issue ผู้ใช้เลือก Issue ที่ต้องการลบ จากนั้นคลิกปุ่มลบ 3. ระบบทำการเพิ่มหรือลบและทำการเก็บลงในประวัติ

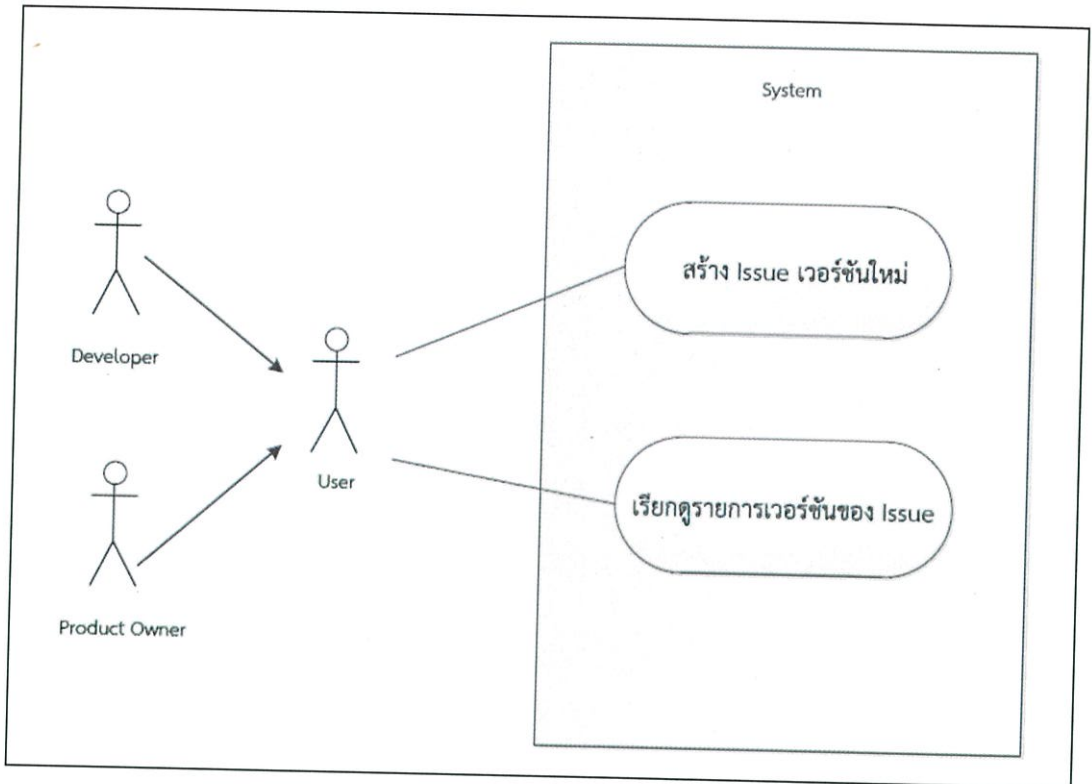
ตารางที่ 3.20 จัดการสถานะ Issue ในรอบการทำงาน

หัวข้อ	คำอธิบาย
Use Case Name	จัดการสถานะ Issue ในรอบการทำงาน
Brief Description	ทำการเปลี่ยนสถานะของ Issue เมื่อมีการเปลี่ยนแปลงความคืบหน้าของงานหรือการแก้ไขงาน
Actor	ผู้ใช้
Trigger	ผู้ใช้ เลือกจัดการสถานะ Issue ในรอบการทำงาน
Preconditions	-
Postconditions	Issue ถูกเปลี่ยนสถานะ
Flow of Events	<ol style="list-style-type: none"> 1. ผู้ใช้ เลือกรอบการทำงานที่ต้องการจัดการสถานะ Issue 2. ผู้ใช้ เลือก Issue ที่ต้องการเปลี่ยนสถานะ ทำการเปลี่ยนสถานะ โดยพื้นฐานประกอบด้วย To Do In Progress Testing Done และ Valid 3. ระบบบันทึกการเปลี่ยนแปลงสถานะ

ตารางที่ 3.21 แก้ไขข้อมูลรอบการทำงาน

หัวข้อ	คำอธิบาย
Use Case Name	แก้ไขข้อมูลรอบการทำงาน
Brief Description	ทำการแก้ไขข้อมูลของรอบการทำงาน โดยสามารถแก้ไขวันที่เริ่มและวันที่สิ้นสุดของรอบการทำงาน
Actor	ผู้ใช้
Trigger	เลือกแก้ไขรอบการทำงาน
Preconditions	-
Postconditions	ข้อมูลรอบการทำงานถูกแก้ไข
Flow of Events	<ol style="list-style-type: none"> 1. ผู้ใช้เลือกแก้ไขรอบการทำงานที่ต้องการ 2. ผู้ใช้ทำการแก้ไขวันที่ของรอบการทำงาน 3. ระบบทำการบันทึกการแก้ไขข้อมูลรอบการทำงาน

3.3.7 เวอร์ชันของ Issue



รูปที่ 3.7 Use Case Diagram ของการสร้างเวอร์ชันของ Issue

จากรูปที่ 3.7 เป็นการแสดงยูสเคสของการสร้างเวอร์ชันของ Issue โดยรายละเอียดของยูสเคสแสดงในตารางตามลำดับต่อไปนี้

ตารางที่ 3.22 สร้าง Issue เวอร์ชันใหม่ .

หัวข้อ	คำอธิบาย
Use Case Name	สร้าง Issue เวอร์ชันใหม่
Brief Description	ทำการสร้าง Issue เวอร์ชันใหม่จาก Issue เดิม - Issue ใน 1 รอบการทำงานสามารถมีได้มากกว่า 1 เวอร์ชัน
Actor	ผู้ใช้
Trigger	ผู้ใช้เลือกสร้าง Issue เวอร์ชันใหม่
Preconditions	มี Issue เดิมที่ต้องการแก้ไข
Postconditions	Issue เวอร์ชันใหม่
Flow of Events	<ol style="list-style-type: none"> 1. ผู้ใช้เรียกดูรายการ Issue 2. ผู้ใช้เลือก Issue ที่ต้องการสร้างเวอร์ชันใหม่ จากนั้นคลิกปุ่มสร้างเวอร์ชันใหม่ 3. ระบบทำการสร้าง Issue เวอร์ชันใหม่

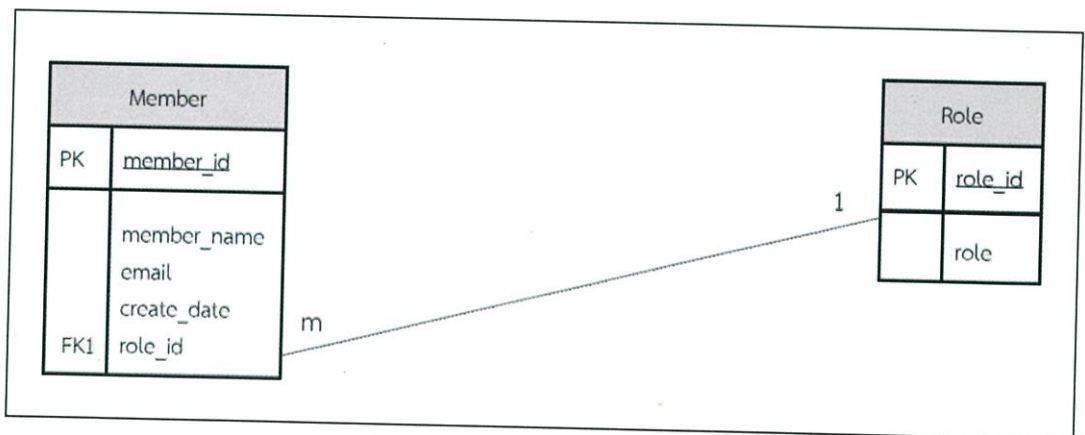
ตารางที่ 3.23 เรียกดูรายการเวอร์ชันของ Issue

หัวข้อ	คำอธิบาย
Use Case Name	เรียกดูรายการเวอร์ชันของ Issue
Brief Description	ทำการเรียกดูรายการเวอร์ชันของ Issue ทั้งหมดที่เป็นเวอร์ชันของ Issue นั้นทั้งเวอร์ชันเก่าและเวอร์ชันใหม่
Actor	ผู้ใช้
Trigger	ผู้ใช้เรียกดูรายการเวอร์ชันของ Issue
Preconditions	-
Postconditions	รายการเวอร์ชันของ Issue
Flow of Events	<ol style="list-style-type: none"> 1. ผู้ใช้เรียกดูรายการ Issue 2. ผู้ใช้เลือก Issue ที่ต้องการเรียกดูรายการเวอร์ชัน จากนั้นคลิกปุ่ม เรียกดู 3. ระบบทำการค้นหาและแสดงรายการเวอร์ชันทั้งหมด

3.3 การออกแบบระบบ

การออกแบบระบบประกอบไปด้วย ER-Diagram โดยอธิบายตามฟังก์ชันต่อไปนี้

3.3.1 สมาชิก



รูปที่ 3.8 แสดง ER-Diagram สมาชิก

จากรูปที่ 3.8 เป็นการแสดงอ็อบเจกต์เพื่อรองรับระบบสมาชิก โดยรายละเอียดของอ็อบเจกต์แสดงในตารางตามลำดับต่อไปนี้

ตารางที่ 3.24 โครงสร้างของตาราง Member

คำอธิบายตาราง : ตารางสำหรับเก็บข้อมูลสมาชิก

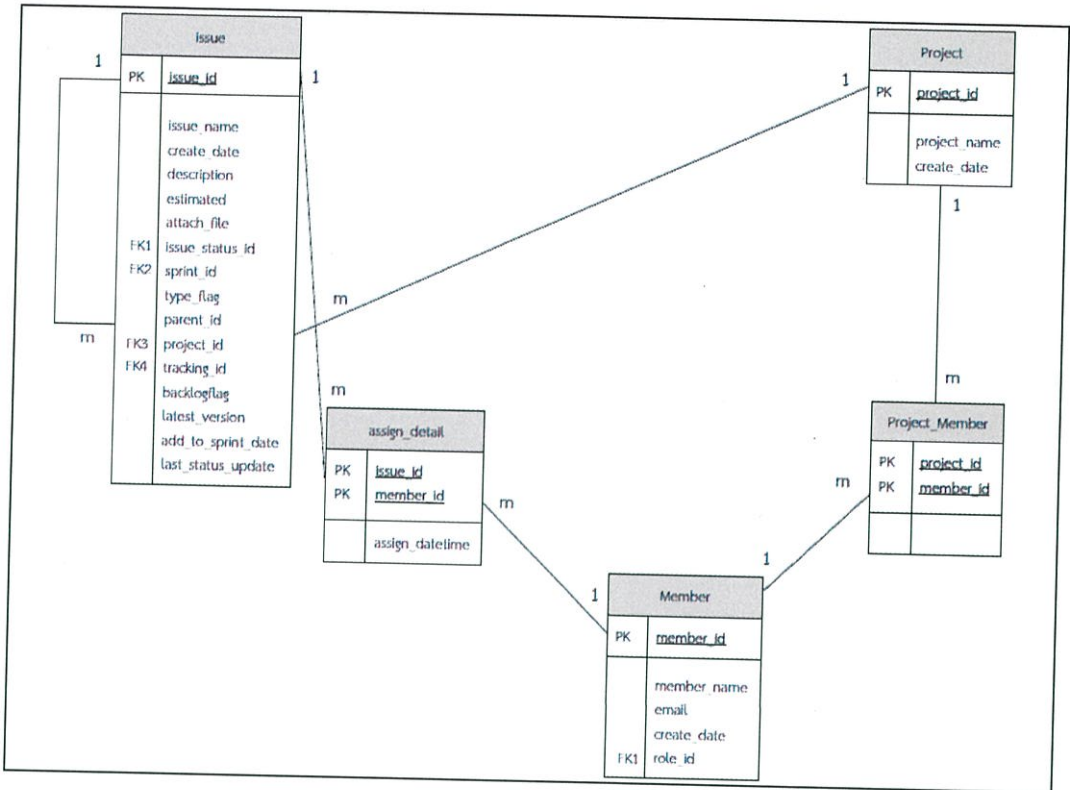
Key	Column Name	Description	Data Type	Field Length
PK	member_id	รหัสของสมาชิก	INTEGER	4
	member_name	ชื่อของสมาชิก	VARCHAR	50
	password	รหัสผ่าน	VARCHAR	8
	email	อีเมลของสมาชิก	VARCHAR	50
	create_date	วันที่สร้างสมาชิก	DATE	
FK	role_id	รหัสของบทบาทสมาชิก	INTEGER	4

ตารางที่ 3.25 โครงสร้างของตาราง Role

คำอธิบายตาราง : ตารางสำหรับเก็บข้อมูลบทบาทของสมาชิก

Key	Column Name	Description	Data Type	Field Length
PK	role_id	รหัสของบทบาทสมาชิก	INTEGER	4
	role_name	ชื่อของบทบาท	VARCHAR	50

3.3.2 โครงการ



รูปที่ 3.9 แสดง ER-Diagram ที่มีความเชื่อมโยงกับโปรเจก

จากรูปที่ 3.9 เป็นการแสดงอีอาร์เพื่อรองรับโปรเจก โดยรายละเอียดของอีอาร์แสดง ใน ตารางตามลำดับต่อไปนี้

ตารางที่ 3.26 โครงสร้างของตาราง Project

คำอธิบายตาราง : ตารางสำหรับเก็บข้อมูลโปรเจก

Key	Column Name	Description	Data Type	Field Length
PK	project_id	รหัสของโปรเจก	INTEGER	4
	project_name	ชื่อของโปรเจก	VARCHAR	50
	create_date	วันที่สร้างโปรเจก	DATE	

ตารางที่ 3.27 โครงสร้างของตาราง Project_Member

คำอธิบายตาราง : ตารางสำหรับเก็บข้อมูลสมาชิกของโปรเจก

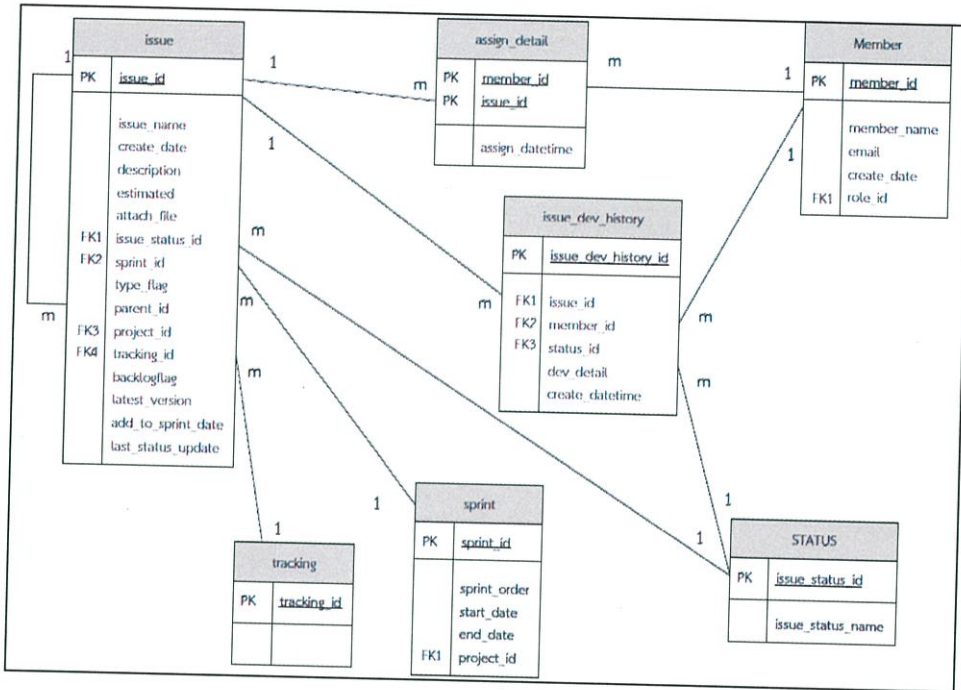
Key	Column Name	Description	Data Type	Field Length
PK	project_id	รหัสของโปรเจก	INTEGER	4
PK	Member_id	ชื่อของโปรเจก	INTEGER	4

โครงสร้างของตาราง Member สามารถดูรายละเอียดได้ในตารางที่ 3.24

โครงสร้างของตาราง Issue สามารถดูรายละเอียดได้ในตารางที่ 3.28

โครงสร้างของตาราง assign_detail สามารถดูรายละเอียดได้ในตารางที่ 3.29

3.3.3 การจัดการ Issue (งานและงานย่อย)



รูปที่ 3.10 แสดง ER-Diagram ของ Issue

จากรูปที่ 3.10 เป็นการแสดงอ็อบเจกต์เพื่อรองรับ Issue โดยรายละเอียดของอ็อบเจกต์แสดง ใน ตารางตามลำดับต่อไปนี้

ตารางที่ 3.28 โครงสร้างของตาราง Issue

คำอธิบายตาราง : ตารางสำหรับเก็บข้อมูล Issue

Key	Column Name	Description	Data Type	Field Length
PK	issue_id	รหัสของ Issue	INTEGER	4
	issue_name	ชื่อของ Issue	VARCHAR	50
	create_date	วันและเวลาที่สร้าง Issue	DATETIME	
	description	คำอธิบายของ Issue	VARCHAR	250
	Estimated	จำนวนวันที่คาดว่าจะเสร็จ	INTEGER	11

FK	issue_status_id	สถานะของ Issue	INTEGER	4
FK	sprint_id	รหัสของรอบการทำงานที่งานหรืองานย่อยนั้นอยู่	INTEGER	4
	type_flag	แสดงชนิดของ Issue เป็นงานหรืองานย่อย	BINARY	1
	parent_id	รหัสสำหรับเชื่อมโยง Issue ที่เป็นงานและงานย่อย	INTEGER	4
FK	project_id	รหัสของโปรเจกงานที่งานหรืองานย่อยนั้นอยู่	INTEGER	4
FK	tracking_id	รหัสสำหรับติดตามเวอร์ชัน	INTEGER	4
	backlog_flag	แสดงสถานะการเป็น Product Backlog หรือไม่ (แสดงการไม่ถูกนำเข้าไปในรอบการทำงาน)	BINARY	1
	latest_version	แสดงสถานะการเป็นเวอร์ชันล่าสุด	BINARY	1
	add_to_sprint_date	วันที่ทำการเพิ่มเข้าสู่รอบการทำงาน	DATE	
	last_status_update	วันที่เปลี่ยนสถานะล่าสุด	DATE	

ตารางที่ 3.29 โครงสร้างของตาราง assign_detail

คำอธิบายตาราง : ตารางสำหรับเก็บข้อมูล assign_detail โดยการมอบหมายสำหรับหนึ่ง Issue สามารถมอบหมายให้สมาชิกได้มากกว่า 1 คน

Key	Column Name	Description	Data Type	Field Length
PK	member_id	รหัสของสมาชิกที่ได้รับมอบหมาย	INTEGER	4
PK	issue_id	รหัสของ Issue ที่มอบหมายให้สมาชิก	INTEGER	4
	assign_datetime	วันและเวลาได้รับมอบหมาย	DATETIME	

ตารางที่ 3.30 โครงสร้างของตาราง Status

คำอธิบายตาราง : ตารางสำหรับเก็บข้อมูลสถานะของ Issue

Key	Column Name	Description	Data Type	Field Length
PK	issue_status_id	รหัสของสถานะ	INTEGER	4
	status_name	ชื่อสถานะ	VARCHAR	50

ตารางที่ 3.31 โครงสร้างของตาราง attached_file

คำอธิบายตาราง : ตารางสำหรับเก็บข้อมูลการแนบไฟล์ ของ Issue

Key	Column Name	Description	Data Type	Field Length
PK	file_id	รหัสของไฟล์	INTEGER	4
	file_path	ชื่อ file_path	VARCHAR	50
FK	issue_id	รหัสของ Issue ที่มีการแนบไฟล์	INTEGER	4

ตารางที่ 3.32 โครงสร้างของตาราง issue_dev_history

คำอธิบายตาราง : ตารางสำหรับเก็บข้อมูลประวัติความคืบหน้าของ Issue ในรอบการทำงาน

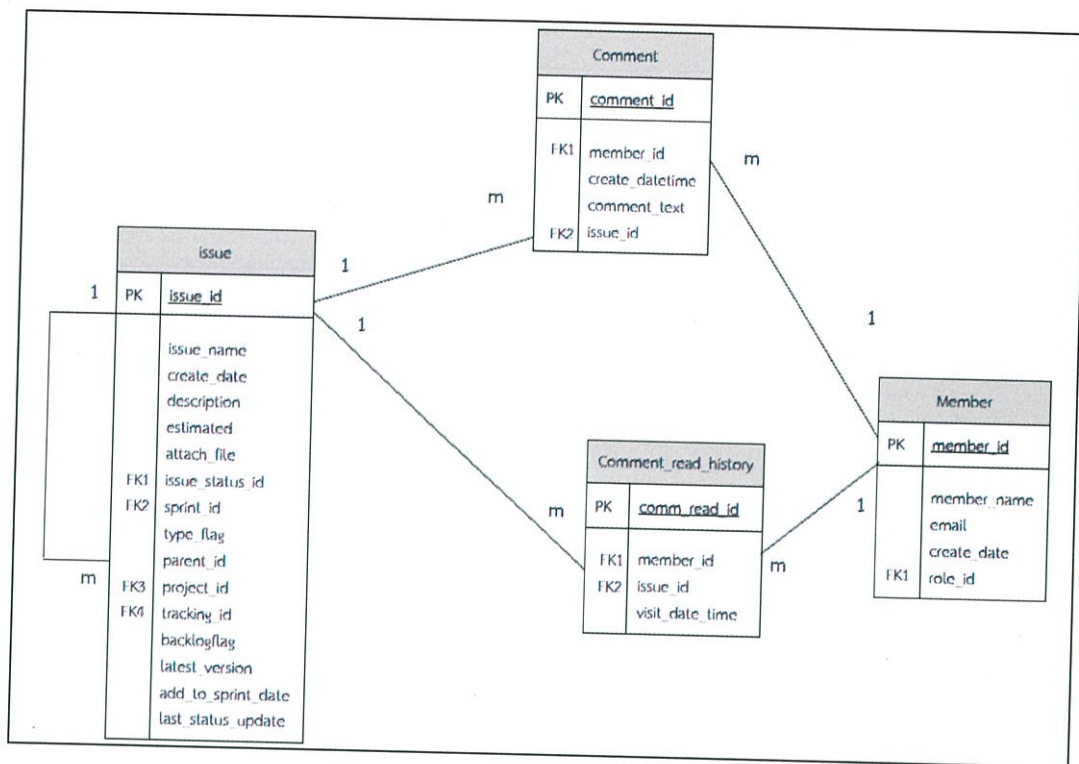
Key	Column Name	Description	Data Type	Field Length
PK	issue_dev_history_id	รหัสข้อมูลการพัฒนาของ Issue	INTEGER	4
FK	issue_id	รหัสของ Issue	INTEGER	4
FK	member_id	รหัสของสมาชิกที่ดำเนินการ	INTEGER	4
FK	status_id	รหัสของสถานะ Issue	INTEGER	4
	dev_detail	รายละเอียดข้อมูลความคืบหน้าของ Issue	VARCHAR	255
	create_datetime	วันและเวลาที่มีการดำเนินการ	DATETIME	

โครงสร้างของตาราง Member สามารถดูรายละเอียดได้ในตารางที่ 3.24

โครงสร้างของตาราง Sprint สามารถดูรายละเอียดได้ในตารางที่ 3.36

โครงสร้างของตาราง Tracking สามารถดูรายละเอียดได้ในตารางที่ 3.37

3.3.4 แสดงความคิดเห็นเกี่ยวกับ Issue



รูปที่ 3.11 แสดง ER-Diagram ของการแสดงความคิดเห็น

จากรูปที่ 3.11 เป็นการแสดงอ็อบเจกต์เพื่อรองรับการแสดงความคิดเห็น โดยรายละเอียดของอ็อบเจกต์แสดง ในตารางตามลำดับต่อไปนี้

ตารางที่ 3.33 โครงสร้างของตาราง Comment

คำอธิบายตาราง : ตารางสำหรับเก็บข้อมูลความคิดเห็น

Key	Column Name	Description	Data Type	Field Length
PK	comment_id	รหัสของ Comment	INTEGER	4
FK	member_id	รหัสของผู้ที่แสดงความคิดเห็น	INTEGER	4
FK	issue_id	รหัสของ Issue ที่ถูกแสดงความคิดเห็น	INTEGER	4
	comment_text	ความคิดเห็น	VARCHAR	250
	create_datetime	วันที่เวลาที่แสดงความคิดเห็น	DATETIME	

ตารางที่ 3.34 โครงสร้างของตาราง Comment_read_history

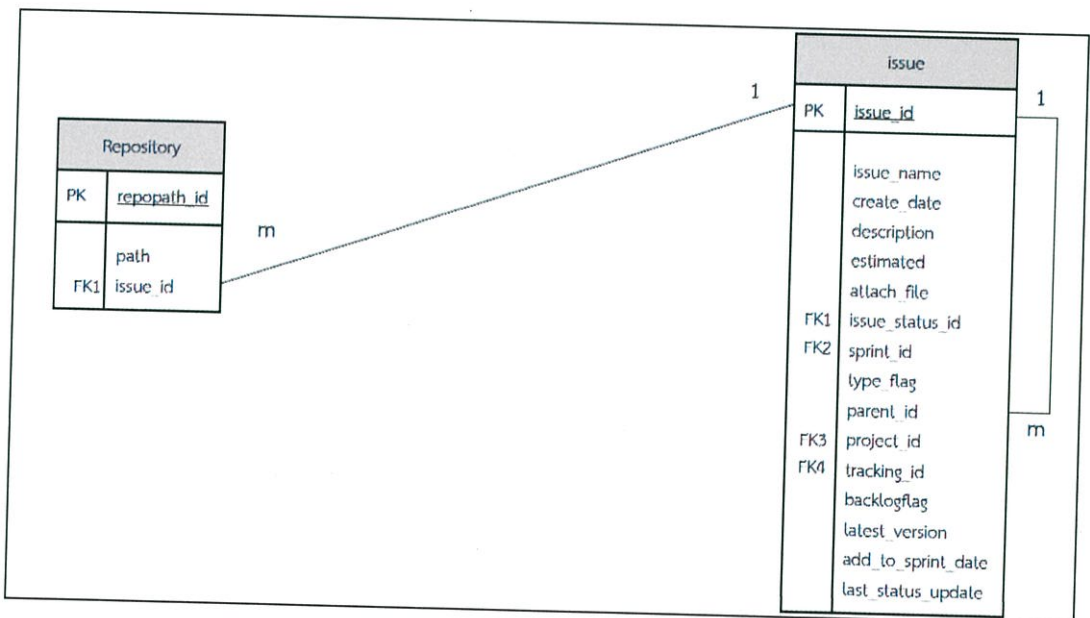
คำอธิบายตาราง : ตารางสำหรับเก็บข้อมูลวันและเวลาที่สมาชิกเข้ามาอ่านความคิดเห็นของแต่ละ Issue ครั้งล่าสุด

Key	Column Name	Description	Data Type	Field Length
PK	comm_read_id	รหัสของการเข้ามาอ่านความคิดเห็น	INTEGER	4
FK	member_id	รหัสของผู้ที่เข้ามาอ่านความคิดเห็น	INTEGER	4
FK	issue_id	รหัสของ Issue ที่ถูกอ่านความคิดเห็น	INTEGER	4
	visit_date_time	วันและเวลาล่าสุดที่ถูกอ่านความคิดเห็น	DATETIME	

โครงสร้างของตาราง Member สามารถดูรายละเอียดได้ในตารางที่ 3.24

โครงสร้างของตาราง Issue สามารถดูรายละเอียดได้ในตารางที่ 3.28

3.3.5 Github Repository ของ Issue



รูปที่ 3.12 แสดง ER-Diagram ของ Repository

จากรูปที่ 3.12 เป็นการแสดงอีอาร์เพื่อรองรับ Repository ของ Issue โดยรายละเอียดของอีอาร์แสดง ในตารางตามลำดับต่อไปนี้

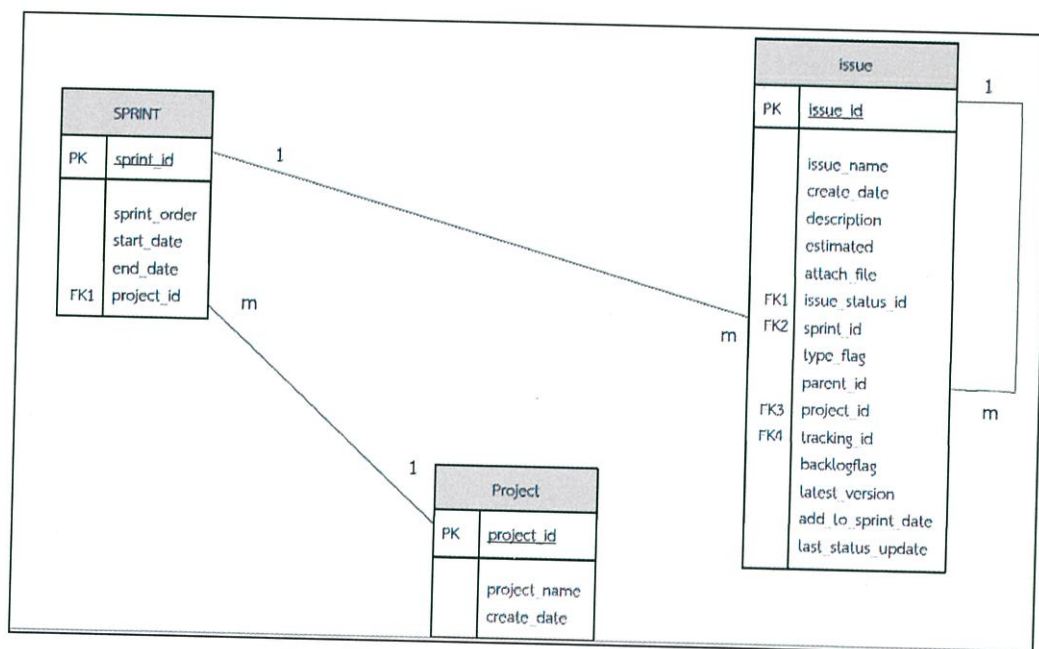
ตารางที่ 3.35 โครงสร้างของตาราง Repository

คำอธิบายตาราง : ตารางสำหรับเก็บ Repository ของ Issue

Key	Column Name	Description	Data Type	Field Length
PK	repopath_id	รหัสของ Repository path	INTEGER	4
FK	issue_id	รหัสของ issue	INTEGER	4
	repository_path	Repository path ของ Github	VARCHAR	100

โครงสร้างของตาราง Issue สามารถดูรายละเอียดได้ในตารางที่ 3.28

3.3.6 รอบการทำงาน (Sprint)



รูปที่ 3.13 แสดง ER-Diagram ของรอบการทำงาน (Sprint)

จากรูปที่ 3.13 เป็นการแสดงอีอาร์เพื่อรองรับของรอบการทำงาน โดยรายละเอียดของอีอาร์แสดง ในตารางตามลำดับต่อไปนี้

ตารางที่ 3.36 โครงสร้างของตาราง Sprint

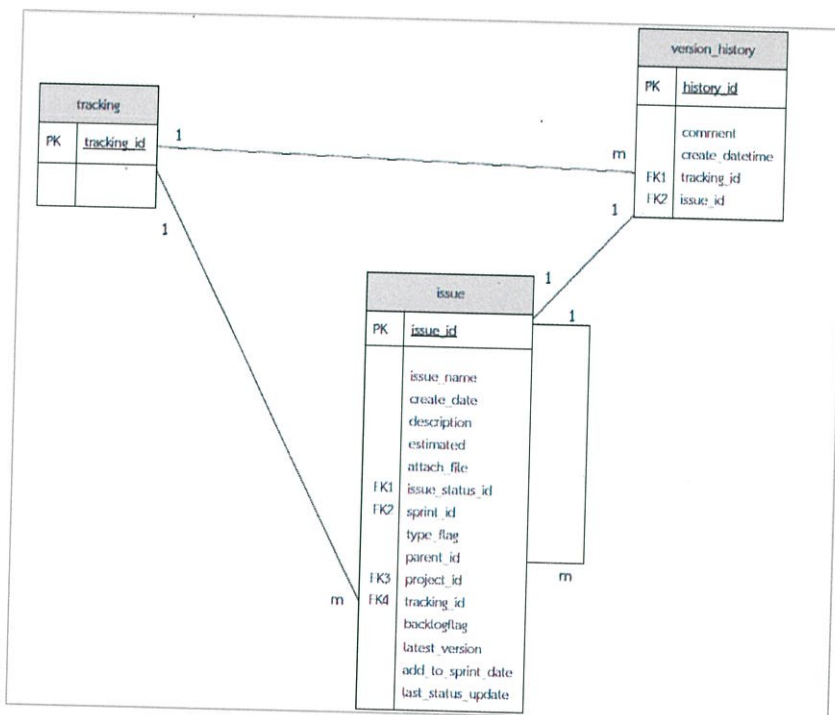
คำอธิบายตาราง : ตารางสำหรับเก็บรอบการทำงาน

Key	Column Name	Description	Data Type	Field Length
PK	sprint_id	รหัสรอบการทำงาน	INTEGER	4
	sprint_order	ลำดับของรอบการทำงาน	INTEGER	2
	start_date	วันเริ่มต้นของรอบการทำงาน	DATE	
	end_date	วันสิ้นสุดรอบการทำงาน	DATE	
	remain_day	วันที่เหลือในรอบการทำงาน	INTEGER	2
	comp_percent	เปอร์เซ็นต์ที่เสร็จสิ้นของรอบการทำงาน	INTEGER	3
FK	project_id	รหัสของโปรเจกต์ของรอบการทำงาน	INTEGER	4
	sprint_status	สถานะของรอบการทำงาน	BINARY	1

โครงสร้างของตาราง Issue สามารถดูรายละเอียดได้ในตารางที่ 3.28

โครงสร้างของตาราง Project สามารถดูรายละเอียดได้ในตารางที่ 3.26

3.3.7 การติดตามเวอร์ชัน Issue



รูปที่ 3.14 แสดง ER-Diagram ของ Issue สำหรับการติดตามเวอร์ชัน

จากรูปที่ 3.14 เป็นการแสดงอีอาร์เพื่อรองรับการติดตามเวอร์ชันของ Issue โดยรายละเอียดของตารางแสดงดังต่อไปนี้

ตารางที่ 3.37 โครงสร้างของตาราง tracking

คำอธิบายตาราง : ตารางสำหรับเก็บ tracking_id สำหรับติดตาม issue ที่มีมากกว่า 1 เวอร์ชัน โดยทุกเวอร์ชันจะมีค่า tracking_id เดียวกัน

Key	Column Name	Description	Data Type	Field Length
PK	tracking_id	รหัสสำหรับติดตามเวอร์ชัน	INTEGER	4

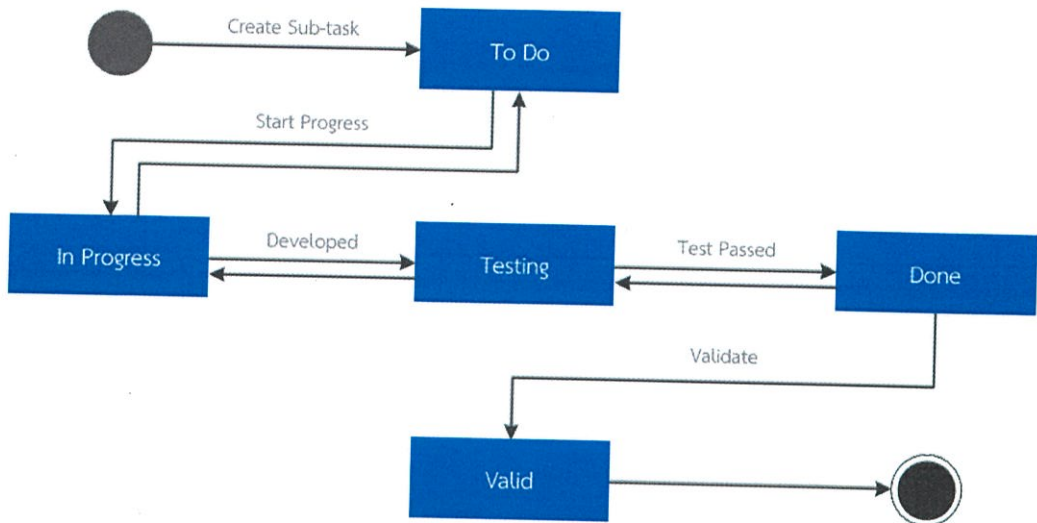
ตารางที่ 3.38 โครงสร้างของตาราง version_history

คำอธิบายตาราง : ตารางสำหรับเก็บข้อมูลประวัติการเปลี่ยนแปลง issue ในเวอร์ชันต่างๆ

Key	Column Name	Description	Data Type	Field Length
PK	history_id	รหัสของประวัติการแก้ไข	INTEGER	4
	Comment	รายละเอียดการแก้ไข issue	VARCHAR	250
FK	member_id	รหัสของสมาชิกที่สร้าง issue เวอร์ชันใหม่	INTEGER	4
	create_datetime	วันและเวลาที่ทำการแก้ไข	DATETIME	
FK	tracking_id	รหัสสำหรับติดตามเวอร์ชัน	INTEGER	4
FK	issue_id	รหัส issue ที่ทำการแก้ไข	INTEGER	4

โครงสร้างของตาราง Issue สามารถดูรายละเอียดได้ในตารางที่ 3.28

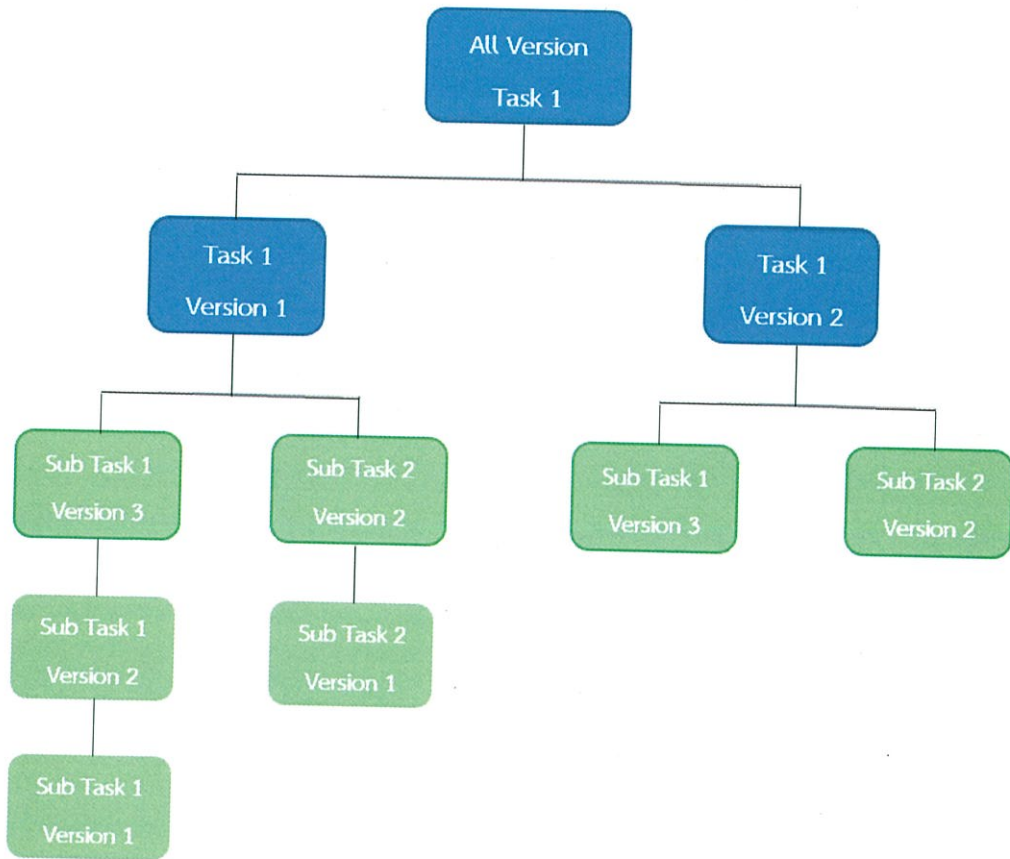
Work Flow การติดตามงานย่อย (Sub-task)



รูปที่ 3.15 แสดง Work Flow การติดตามงานย่อย

จากรูปที่ 3.15 เมื่อมีการสร้างงานย่อยสถานะงานจะอยู่ในสถานะ To Do เพื่อรอการดำเนินงาน เมื่อผู้ใช้เริ่มทำการพัฒนางานย่อยนั้นสถานะงานย่อยจะถูกเปลี่ยนเป็น In Progress เมื่องานย่อยถูกพัฒนาจนเสร็จสิ้น สถานะของงานย่อยจะเปลี่ยนเป็น Testing ซึ่งเป็นกระบวนการทดสอบงานย่อยตามที่ได้มีการพัฒนาในสถานะ In Progress นั้น หลังจากที่ผ่านมากระบวนการทดสอบสถานะของงานย่อยนั้นจะเปลี่ยน Done เป็นการระบุว่าพัฒนางานย่อยนั้นได้เสร็จสิ้นแล้ว หลังจากนั้นจะเข้าสู่กระบวนการตรวจสอบความถูกต้องของงานย่อยนั้นเมื่อผ่านการตรวจสอบแล้วสถานะของงานย่อยจะเปลี่ยนเป็น Valid เป็นอันสิ้นสุดกระบวนการพัฒนางานย่อยนั้น

การสร้างงานเวอร์ชันใหม่

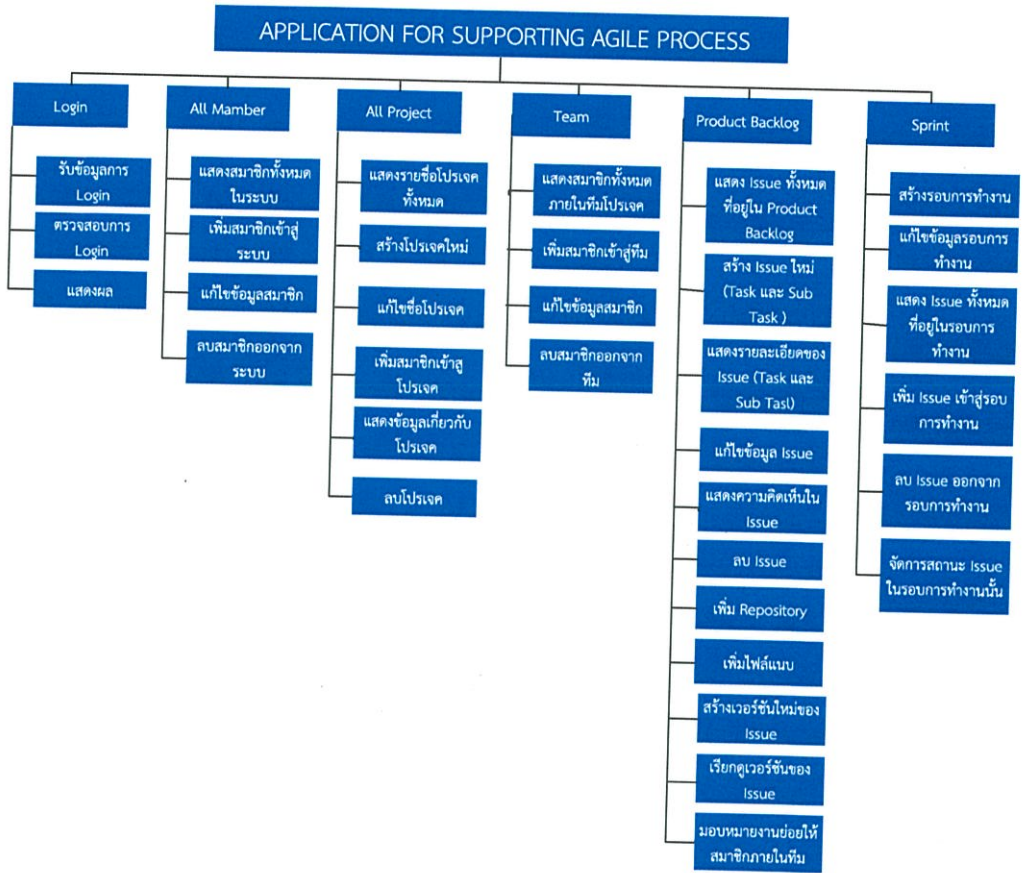


รูปที่ 3.16 แสดง การสร้างงานเวอร์ชันใหม่

จากรูปที่ 3.16 เป็นการแสดงตัวอย่างการสร้างงานเวอร์ชันใหม่ (Task 1 Version 2) โดยที่งานเวอร์ชันก่อนหน้านั้น (Task 1 Version 1) มีงานย่อยเวอร์ชันล่าสุด 2 งานย่อย คือ Sub Task 1 Version 3 และ Sub Task 2 Version 2 โดยที่งานย่อยแต่ละงานย่อยมีเวอร์ชันเก่าอยู่ เมื่อมีการสร้างงานเวอร์ชันใหม่ระบบจะนำข้อมูลของงานย่อยเวอร์ชันล่าสุดทั้งหมดของงานเวอร์ชันก่อนหน้านั้น คือ Sub Task 1 Version 3 และ Sub Task 2 Version 2 มาสร้างเป็นงานย่อยใหม่ให้กับงานเวอร์ชันใหม่โดยอัตโนมัติ

Structure Chart

จากการออกแบบระบบข้างต้น ผู้พัฒนาสามารถนำมาเขียนแผนภาพโครงสร้าง (Structure Chart) ได้ดังนี้



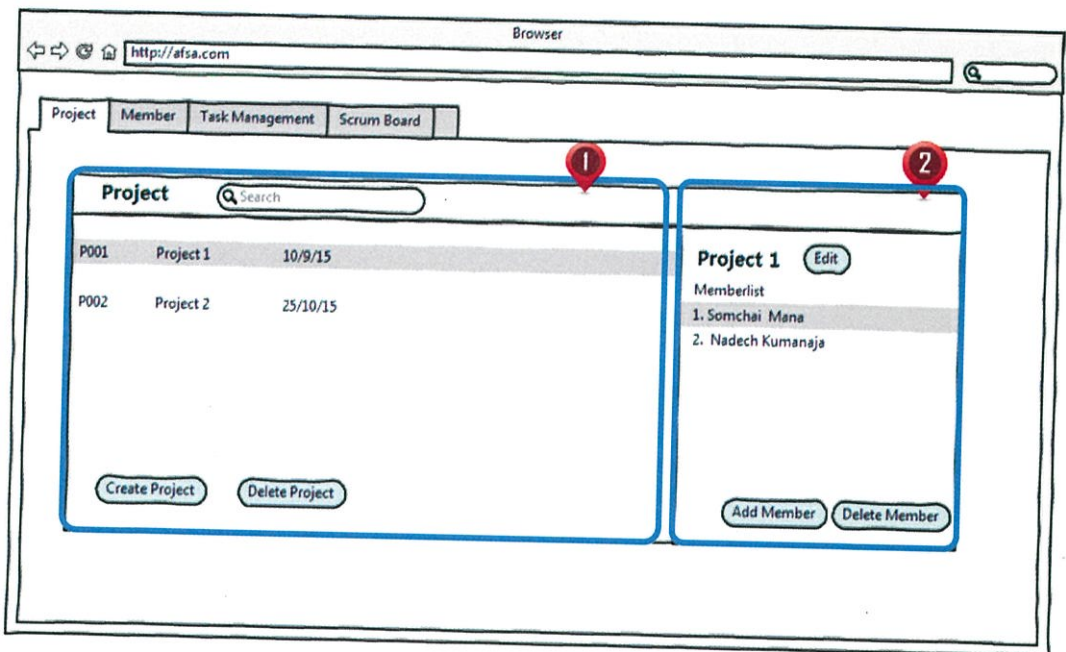
รูปที่ 3.17 แสดง Structure Chart ของระบบ

การออกแบบส่วนการติดต่อกับผู้ใช้

ส่วนการติดต่อกับผู้ใช้มีการออกแบบเป็น 4 ส่วนประกอบด้วย

1. การจัดการโปรเจค
2. การจัดการสมาชิก
3. การจัดการ Issue ภายใน Product Backlog
4. การจัดการรอบการทำงาน (Sprint)

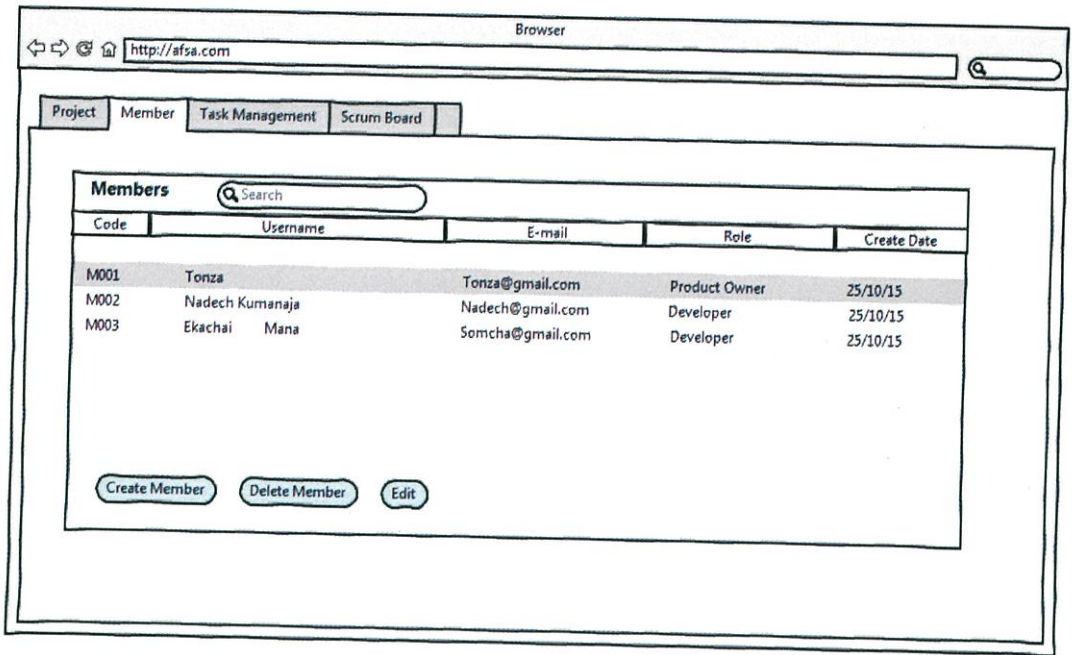
โดยแต่ละส่วนมีรายละเอียดดังนี้



รูปที่ 3.18 ส่วนติดต่อกับผู้ใช้สำหรับการจัดการโปรเจค

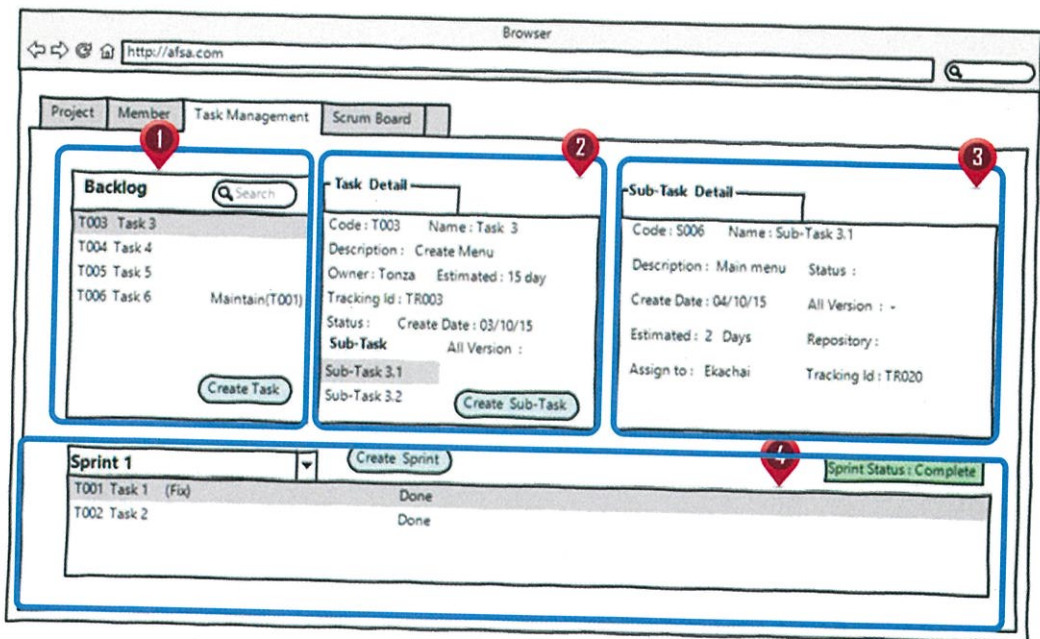
จากรูปที่ 3.18 เป็นส่วนสำหรับการจัดการโปรเจค โดยในกรอบที่ 1 เป็นส่วนสำหรับแสดงรายการโปรเจค โดยสามารถสร้างโปรเจคได้ โดยกดที่ปุ่ม Create Project และลบโปรเจคได้โดยการกดปุ่ม Delete Project

ในกรอบที่ 2 เป็นส่วนสำหรับแสดงรายชื่อของสมาชิกที่อยู่ในแต่ละโปรเจค โดยสามารถเพิ่มสมาชิกเข้าสู่โปรเจคและลบสมาชิกออกจากโปรเจคได้โดยกดที่ปุ่ม Add Member และปุ่ม Delete Member ตามลำดับ



รูปที่ 3.19 ส่วนติดต่อกับผู้ใช้สำหรับการจัดการสมาชิก

จากรูปที่ 3.19 เป็นส่วนสำหรับการจัดการสมาชิก โดยเป็นส่วนสำหรับแสดงข้อมูลของสมาชิกที่มีอยู่ในระบบซึ่งประกอบไปด้วย รหัสสมาชิก ชื่อสมาชิก อีเมลล์ บทบาทของสมาชิก และวันที่สร้างบัญชีของสมาชิก โดยสามารถสร้างสมาชิกใหม่ได้โดยคลิกที่ปุ่ม Create Member และลบสมาชิกได้โดยคลิกปุ่ม Delete Member และแก้ไขข้อมูลของสมาชิกได้โดยคลิกที่ปุ่ม Edit



รูปที่ 3.20 ส่วนติดต่อกับผู้ใช้สำหรับการจัดการ Issue ภายใน Product Backlog

จากรูปที่ 3.20 เป็นส่วนสำหรับการจัดการ Issue ภายใน Product Backlog โดย ในกรอบที่ 1 เป็นส่วนที่ใช้แสดงรายการงาน ที่อยู่ใน Product Backlog โดยสามารถสร้างงานใหม่ได้โดยคลิกที่ปุ่ม Create Task

ในกรอบที่ 2 เป็นส่วนที่ใช้แสดงรายละเอียดข้อมูลของแต่ละงาน และแสดงรายการของงานย่อยที่อยู่ในงานนั้น ซึ่งประกอบไปด้วย รหัสงาน (code) ชื่องาน (name) คำอธิบาย (Description) ผู้รับผิดชอบงาน (assign to) จำนวนวันที่คาดว่าจะเสร็จ (estimated) รหัสสำหรับติดตามเวอร์ชัน (tracking id) สถานะงาน (status) วันที่สร้างงาน (create date) เวอร์ชันของงานย่อยนั้น (all version) และรายการงานย่อย (sub task) ของงานนั้น โดยสามารถสร้างงานย่อยเพิ่มได้โดยกดที่ปุ่ม Create Sub-Task

ในกรอบที่ 3 เป็นส่วนที่ใช้แสดงรายละเอียดข้อมูลของงานย่อย โดยผู้ใช้คลิกเลือกที่รายการงานย่อยจากกรอบที่ 2 ระบบจะแสดงรายละเอียดงานย่อยนั้น ซึ่งประกอบไปด้วย รหัสงานย่อย (code) ชื่องานย่อย (name) คำอธิบาย (Description) วันที่สร้างงานย่อย (create date) จำนวนวันที่คาดว่าจะเสร็จ (estimated) ผู้รับผิดชอบงาน (assign to) สถานะงานย่อย (status) เวอร์ชันของงานย่อยนั้น (all version) พาทที่เก็บไฟล์ข้อมูล (repository path) รหัสสำหรับติดตามเวอร์ชัน (tracking id)

ในกรอบที่ 4 เป็นส่วนที่ใช้นำงานเข้าสู่รอบการทำงานและแสดงสถานะของงานในรอบการทำงานนั้น โดยสามารถสร้างรอบการทำงานใหม่ได้โดยกดที่ปุ่ม Create Sprint และสามารถรับรู้สถานะของรอบการทำงานได้

Browser: http://afsa.com

Sprint 2 | Start Date: 5/2/59 | End Date: 20/2/59 | Sprint Progress 50%

Task	To Do	In Progress	Testing	Done	Valid
T03 Task 3	T10 SubTask 3.1	T11 SubTask 3.2			T12 SubTask 3.3
T04 Task 4	T13 SubTask 4.1 T14 SubTask 4.2			T15 SubTask 4.3	T16 SubTask 4.4

รูปที่ 3.21 ส่วนติดต่อกับผู้ใช้สำหรับการจัดการรอบการทำงาน (Sprint)

จากรูปที่ 3.18 เป็นส่วนสำหรับการรอบการทำงาน (Sprint) โดยในกรอบที่ 1 เป็นส่วนที่ใช้แสดงรายการงานที่อยู่ในรอบการทำงานนั้น

ในกรอบที่ 2 เป็นส่วนที่แสดงรายการงานย่อยของแต่ละงาน โดยงานย่อยจัดแบ่งตามสถานะของงาน โดยพื้นฐานประกอบด้วย To Do In Progress Testing Done และ Valid

ในกรอบที่ 3 เป็นส่วนที่ใช้แสดงจำนวนวันที่เหลือในรอบการทำงาน แสดงความคืบหน้าของงานในรอบการทำงานนั้นเป็นรูปแบบเปอร์เซ็นต์

บทที่ 4

ผลการดำเนินการ

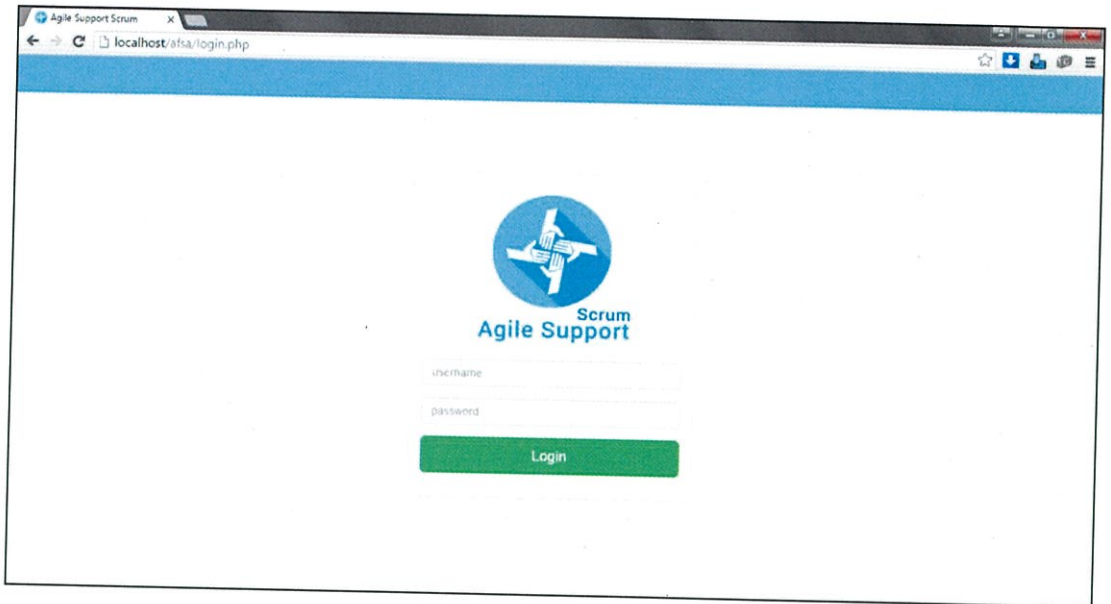
4.1 ภาพรวมของระบบ

แอปพลิเคชันสำหรับสนับสนุนกระบวนการพัฒนาแบบอไจล์ เป็นแอปพลิเคชันที่พัฒนาขึ้นมาเพื่อช่วยให้ผู้ใช้สามารถบริหารจัดการและติดตามงานในการพัฒนาซอฟต์แวร์แบบอไจล์ได้สะดวกขึ้น โดยระบบมีความสามารถในการสร้างสมาชิก สร้างโปรเจกต์จัดการสมาชิกเข้าสู่โปรเจกต์ ผู้ใช้สามารถดูรายละเอียดของแต่ละโปรเจกต์ได้ และผู้ใช้สามารถรับรู้ความคืบหน้าของรอบการทำงานได้ในรูปแบบ Burndown Chart ผู้ใช้สามารถสร้างงาน ใส่ข้อมูลรายละเอียดงาน และสามารถเพิ่ม Repository จาก Github ได้ สามารถเพิ่ม Attach File ให้กับงาน โดยผู้ใช้สามารถแสดงความคิดเห็น (Comment) เกี่ยวกับงาน ผู้ใช้สามารถสร้างงานย่อยของแต่ละงานและมอบหมายงานย่อยให้กับสมาชิกภายในโปรเจกต์ได้ ผู้ใช้สามารถสร้างรอบการทำงาน และนำงานเข้าสู่รอบการทำงาน และสามารถบริหารจัดการงานและงานย่อยในรอบการทำงานในรูปแบบสกรัมบอร์ด ผู้ใช้สามารถจัดการงาน ติดตามและรับรู้สถานะของงานในรอบการทำงานได้ สามารถระบุและรับรู้ความคืบหน้าของงานย่อยในรอบการทำงาน ผู้ใช้สามารถสร้างเวอร์ชันของงานและงานย่อยขึ้นใหม่ และสามารถเรียกดูข้อมูลของงานและงานย่อยทุกเวอร์ชันได้ และผู้ใช้สามารถกรอกรายละเอียดความแตกต่างของงานและงานย่อยแต่ละเวอร์ชันได้

4.2 ความสามารถของระบบ

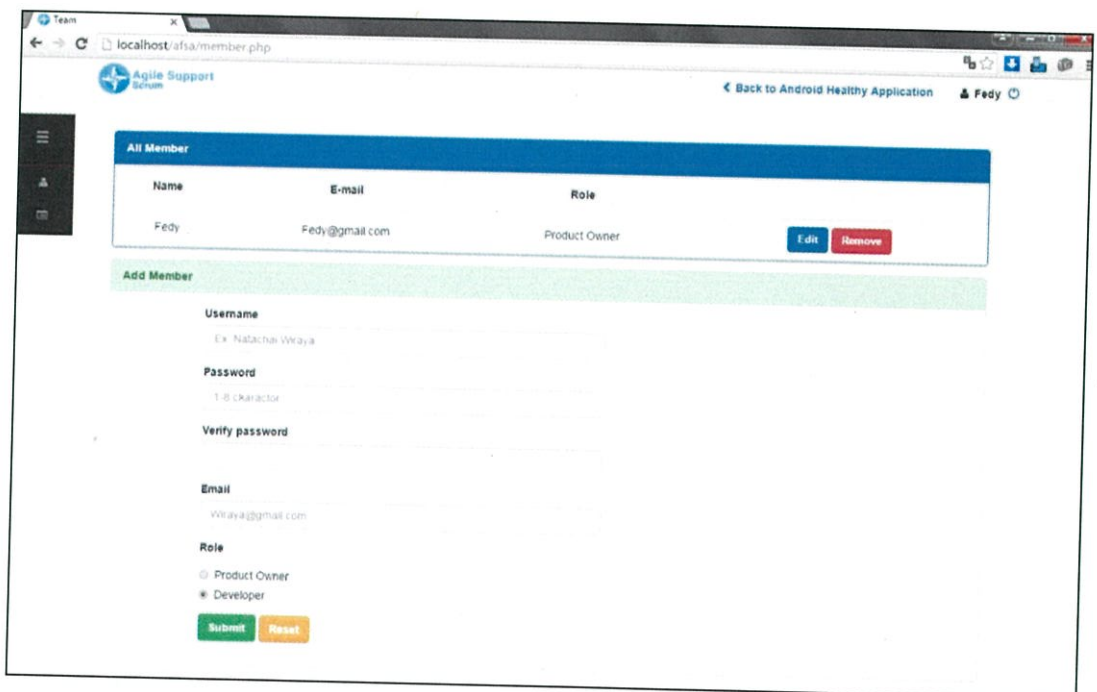
4.2.1 ส่วนของการจัดการสมาชิก

- สามารถสร้างสมาชิก
- สามารถลบสมาชิก
- สามารถแก้ไขข้อมูลส่วนตัวสมาชิก



รูปที่ 4.1 หน้าจอ Login

จากรูปที่ 4.1 เมื่อเข้าสู่แอปพลิเคชันผู้ใช้งานจะต้องล็อกอินเข้าสู่ระบบ โดยการป้อนชื่อผู้ใช้และรหัสผ่าน

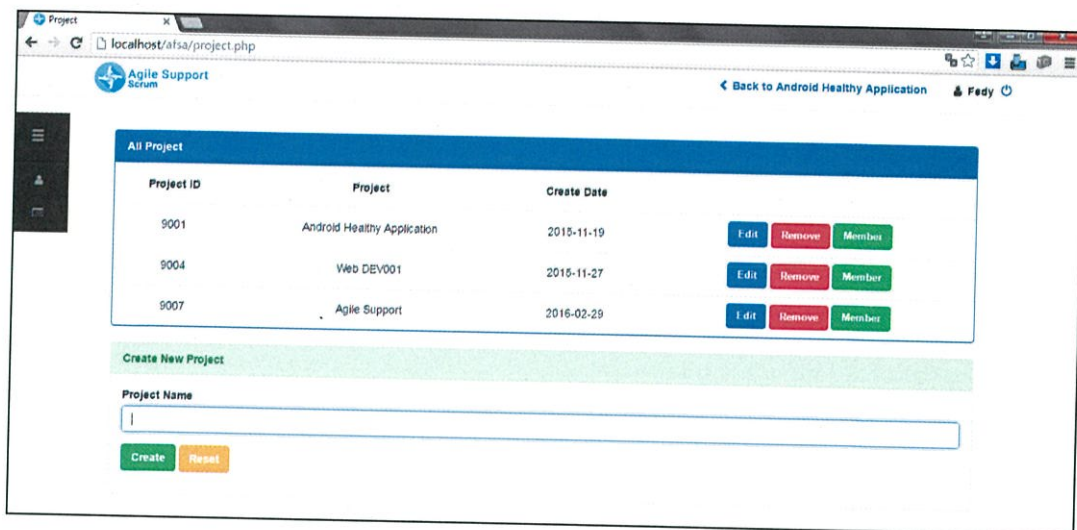


รูปที่ 4.2 หน้าจอการจัดการสมาชิกของระบบ

จากรูปที่ 4.2 จะเป็นการแสดงรายชื่อสมาชิกทั้งหมดของระบบ โดยผู้ใช้ที่เป็นสมาชิกโดยมีบทบาทเป็น Product Owner สามารถเพิ่มสมาชิกเข้าสู่ระบบ โดยการกรอกฟอร์มสำหรับการเพิ่มสมาชิก และสมาชิกสามารถแก้ไขข้อมูลของตนเองได้ และ Product Owner สามารถลบสมาชิกจากระบบได้

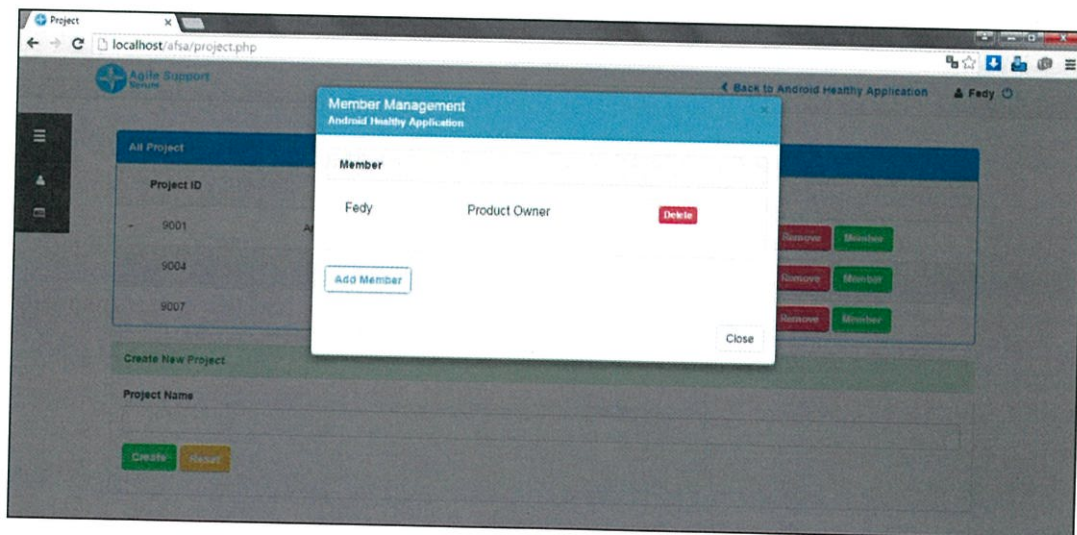
4.2.2 ส่วนของการจัดการโปรเจกต์

ระบบสามารถสร้างโปรเจกต์ ลบโปรเจกต์ แก้ไขชื่อโปรเจกต์ จัดการสมาชิกภายในโปรเจกต์ แสดงข้อมูลรายละเอียดโดยรวมของโปรเจกต์ พร้อมแสดงความคืบหน้าของรอบการทำงานในรูปแบบ Burndown Chart



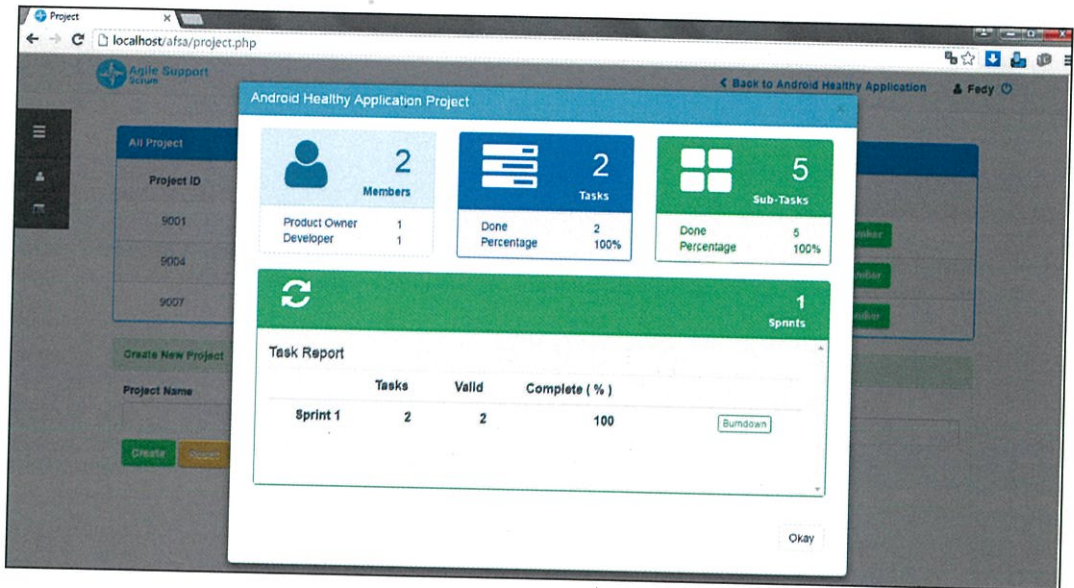
รูปที่ 4.3 หน้าจอการจัดการสมาชิกของระบบ

จากรูปที่ 4.3 จะเป็นการแสดงรายชื่อโปรเจกต์ทั้งหมดของระบบ โดยผู้ใช้ที่เป็นสมาชิก โดยมีบทบาทเป็น Product Owner สามารถสร้างโปรเจกต์ใหม่ได้ โดยสามารถแก้ไขชื่อโปรเจกต์ ลบโปรเจกต์ และจัดการสมาชิกภายในโปรเจกต์ โดยการจัดการสมาชิกโปรเจกต์ระบบจะแสดงหน้าจอดังรูปที่ 4.4



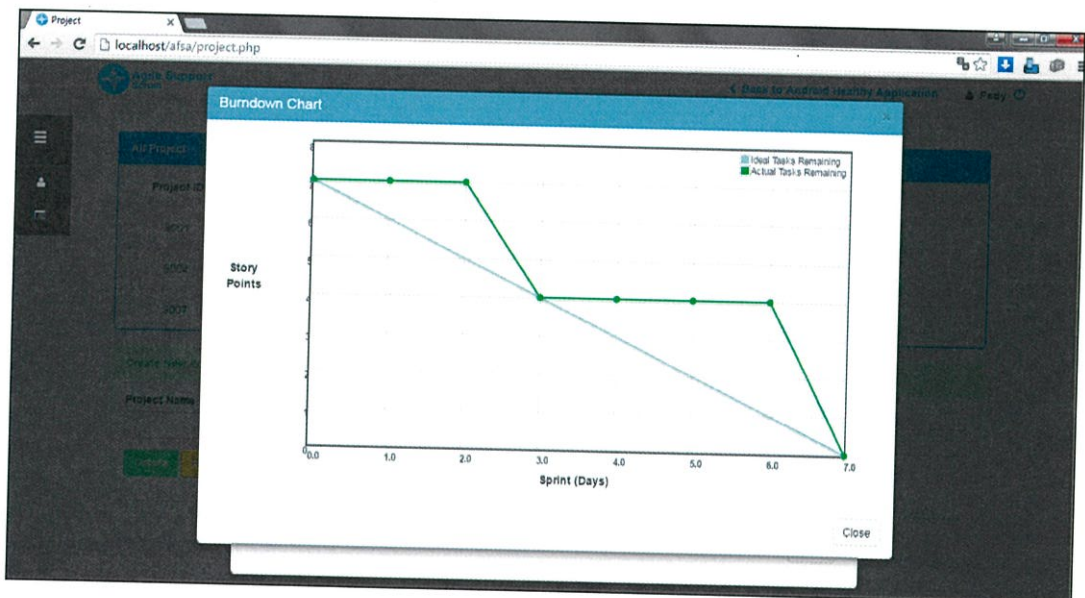
รูปที่ 4.4 หน้าจอสำหรับจัดการสมาชิกในโปรเจกต์

จากรูปที่ 4.4 ระบบจะแสดงรายชื่อของสมาชิกที่อยู่ในโปรเจกต์นั้น โดยผู้ใช้ที่มีบทบาทเป็น Product Owner สามารถลบสมาชิกออกจากโปรเจกต์และเพิ่มสมาชิกเข้าสู่โปรเจกต์ได้ ผู้ใช้สามารถคลิกที่โปรเจกต์เพื่อดูรายละเอียดของแต่ละโปรเจกต์ได้โดยระบบจะแสดงหน้าจอ ดังรูปที่ 4.5



รูปที่ 4.5 หน้าจอแสดงรายละเอียดของแต่ละโปรเจกต์

จากรูปที่ 4.5 ระบบจะแสดงข้อมูลของแต่ละโปรเจกต์ โดยแสดงข้อมูลจำนวนสมาชิก จำนวนงาน จำนวนงานย่อย แสดงเปอร์เซ็นต์ความคืบหน้าของงาน งานย่อย และข้อมูลของรอบการทำงานในโปรเจกต์นั้น โดยผู้ใช้สามารถรับรู้ความคืบหน้าของงานในรอบการทำงานได้โดยการเรียกดู Burndown Chart ของแต่ละรอบการทำงานได้ โดยระบบจะแสดงหน้าจอ ดังรูปที่ 4.6

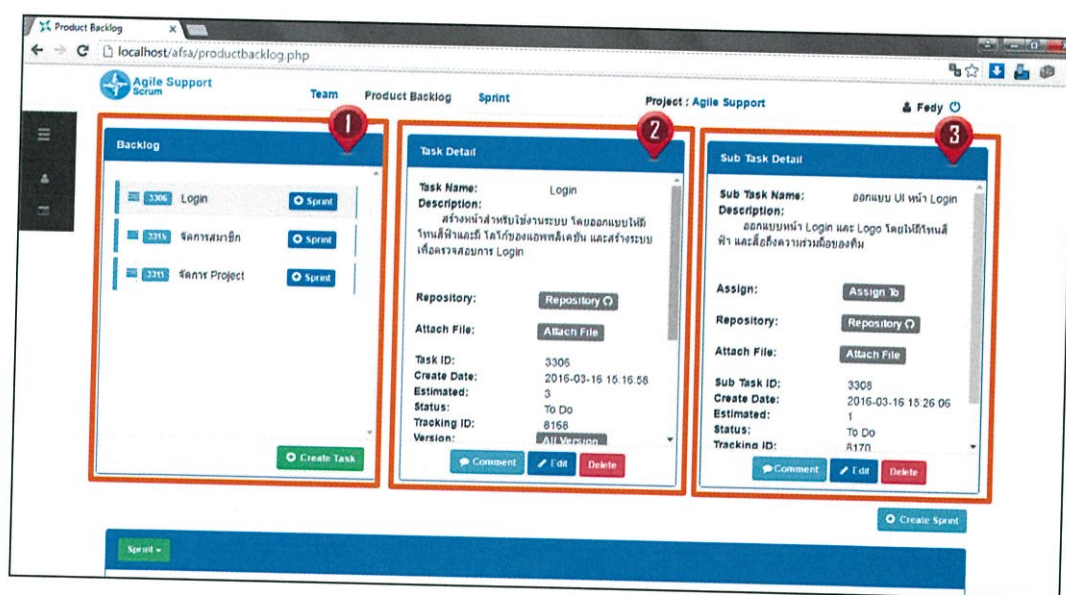


รูปที่ 4.6 หน้าจอแสดง Burndown Chart

จากรูปที่ 4.6 ระบบจะแสดง Burndown Chart โดยแสดง Story points ในแนวแกน Y และจำนวนวันของรอบการทำงานในแนวแกน X

4.2.3 ส่วนของการจัดการ Product Backlog

Product Backlog เป็นหน้าสำหรับจัดการงาน (Task) และงานย่อย (Sub Task) โดยสามารถสร้างงานและงานย่อยได้ ใส่รายละเอียดของงานและงานย่อย สามารถเรียกดู Repository จาก Github เพิ่ม Attach File ให้กับงาน แสดงความคิดเห็นเกี่ยวกับงาน เรียกดูงานทุกเวอร์ชัน และสามารถสร้างรอบการทำงาน เพิ่มงานเข้าสู่รอบการทำงานได้ โดยระบบจะแสดงดังรูปที่ 4.7

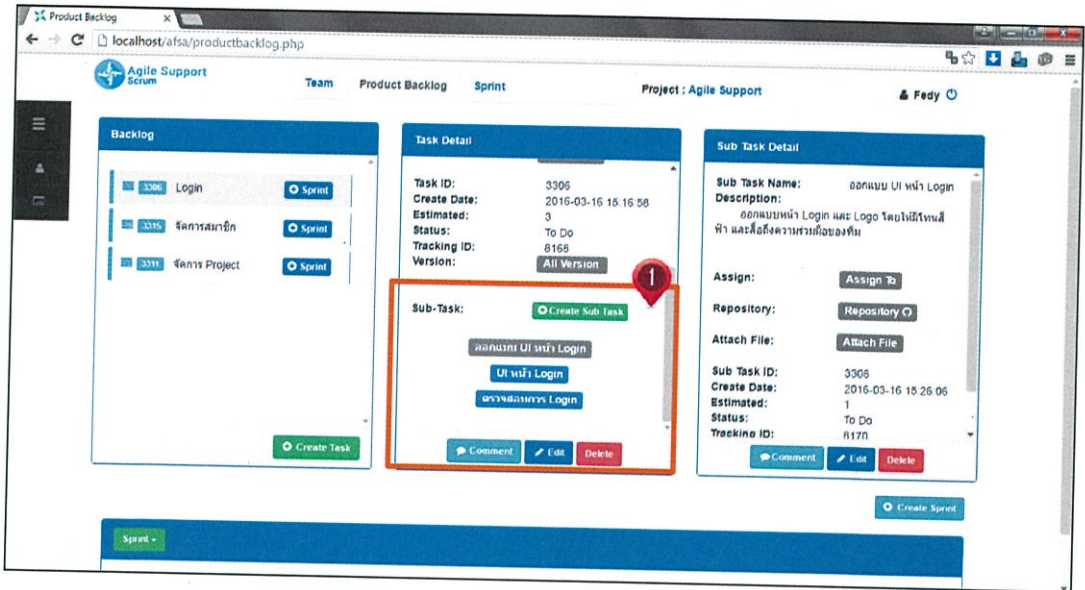


รูปที่ 4.7 หน้าจอสำหรับการจัดการ Product Backlog

จากรูปที่ 4.7 ระบบแสดงหน้าจอสำหรับการจัดการ Product Backlog โดยแบ่งเป็น 3 ส่วนคือ ในกรอบที่ 1 เป็นรายการของงาน (Task) ที่ผู้ใช้สร้างขึ้น โดยสามารถเรียงลำดับความสำคัญจากบนลงล่างได้ และเลือกงานนั้นเข้าสู่รอบการทำงาน (Sprint) ได้ เมื่อผู้ใช้คลิกที่งานระบบจะแสดงรายละเอียดงานในกรอบที่ 2

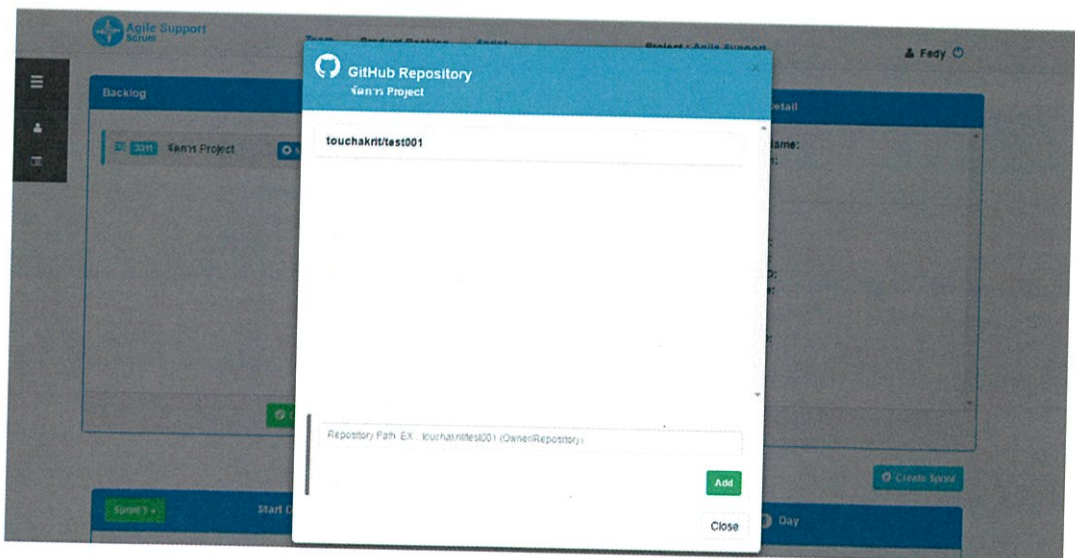
กรอบที่ 2 แสดงรายละเอียดงาน และสามารถเพิ่ม Repository จาก Github ตามรูปที่ 4.9 และสามารถดูข้อมูลภายใน Repository ได้ ดังรูปที่ 4.10 สามารถเพิ่ม Attach File ได้ดังรูปที่ 4.11 ผู้ใช้สามารถเรียกดูเวอร์ชันทั้งหมดของงานนั้นได้ ดังรูปที่ 4.21 นอกจากนั้นผู้ใช้สามารถแก้ไขรายละเอียดของงาน มีระบบแจ้งเตือนความคิดเห็นที่ผู้ใช้ยังไม่ได้อ่านดังรูปที่ 4.12 สามารถแสดงความคิดเห็น (Comment) เกี่ยวกับงานได้ดังรูปที่ 4.13 และลบงานนั้นได้ ซึ่งผู้ใช้สามารถสร้างงานย่อย (Sub Task) ของงานนั้นได้โดยรายการงานย่อยจะแสดงดังรูปที่ 4.8 ในกรอบที่ 1 เมื่อผู้ใช้คลิกที่งานย่อยระบบจะแสดงรายละเอียดงานย่อยในรูปที่ 4.7 กรอบที่ 3

กรอบที่ 3 แสดงรายละเอียดงานย่อย และสามารถเพิ่ม Repository จาก Github เพิ่ม Attach File และมอบหมายงานย่อย (Assign) ให้กับสมาชิกภายในทีมได้ ดังรูปที่ 4.14 ผู้ใช้สามารถเรียกดูเวอร์ชันทั้งหมดของงานนั้นได้ นอกจากนี้ผู้ใช้สามารถแก้ไขรายละเอียดของงานย่อยสามารถแสดงความคิดเห็นและลบงานนั้นได้



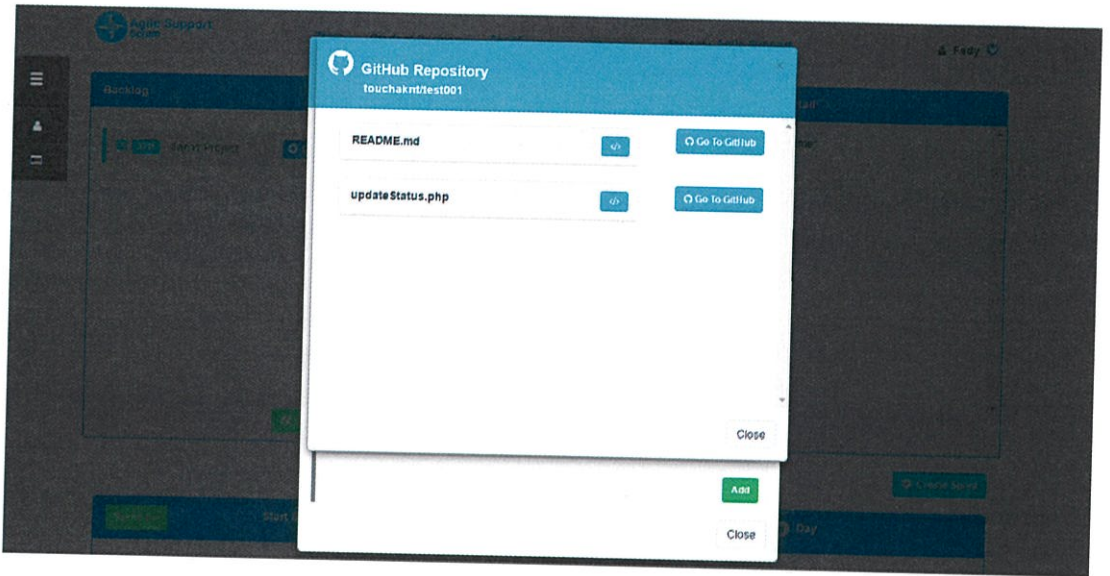
รูปที่ 4.8 หน้าจอสำหรับการจัดการ Product Backlog แสดง งานย่อย (Sub Task)

ซึ่งผู้ใช้สามารถดูรายละเอียดของงานได้ โดยสามารถดู Repository จาก Github ดังรูปที่ 4.9



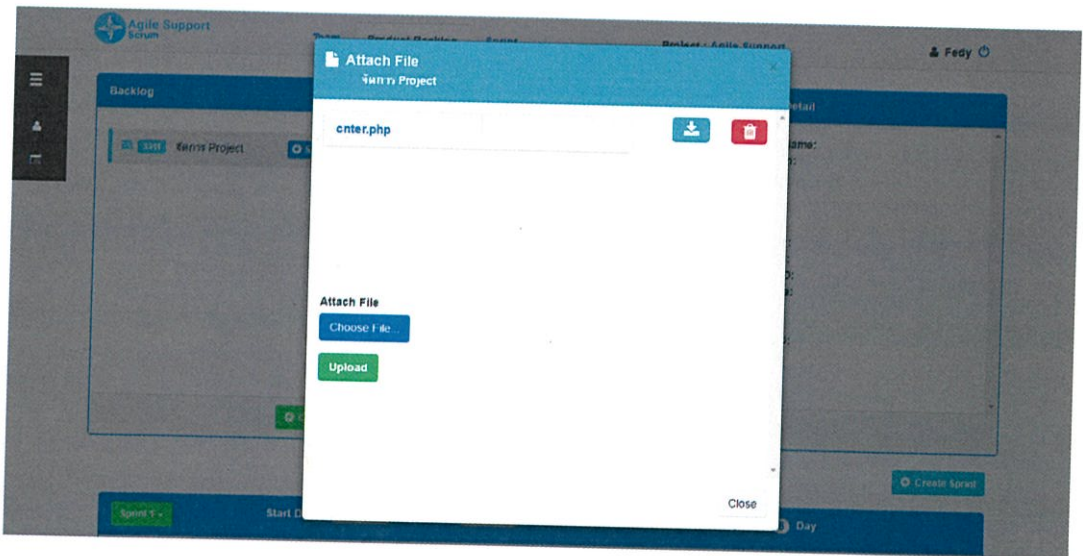
รูปที่ 4.9 หน้าจอแสดงการเพิ่ม Repository จาก Github

จากรูปที่ 4.9 ผู้ใช้สามารถนำข้อมูลจาก Github มาใส่ได้โดยการใส่ชื่อ Repository Path จากนั้นผู้ใช้สามารถเข้าไปดูข้อมูลของ Repository ได้ ดังรูปที่ 4.10



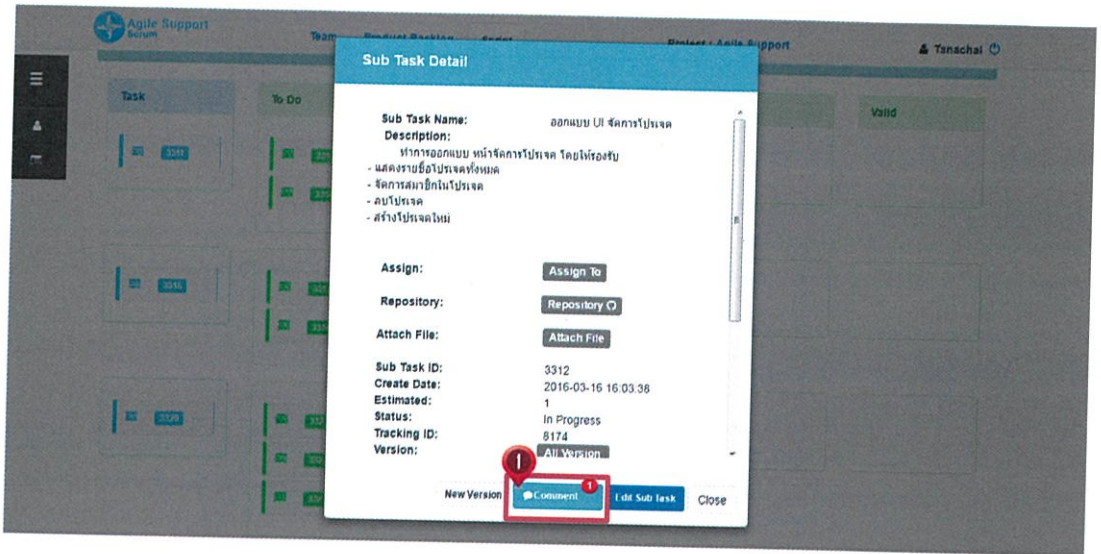
รูปที่ 4.10 หน้าจอแสดงข้อมูลใน Repository จาก Github

จากรูปที่ 4.10 ระบบทำการแสดงข้อมูลใน Repository Path ซึ่งผู้ใช้สามารถดูซอร์ซโค้ดหรือกดเข้าไปดูข้อมูลใน Github โดยตรงได้ และสามารถเพิ่ม Attach File ได้ดังรูปที่ 4.11



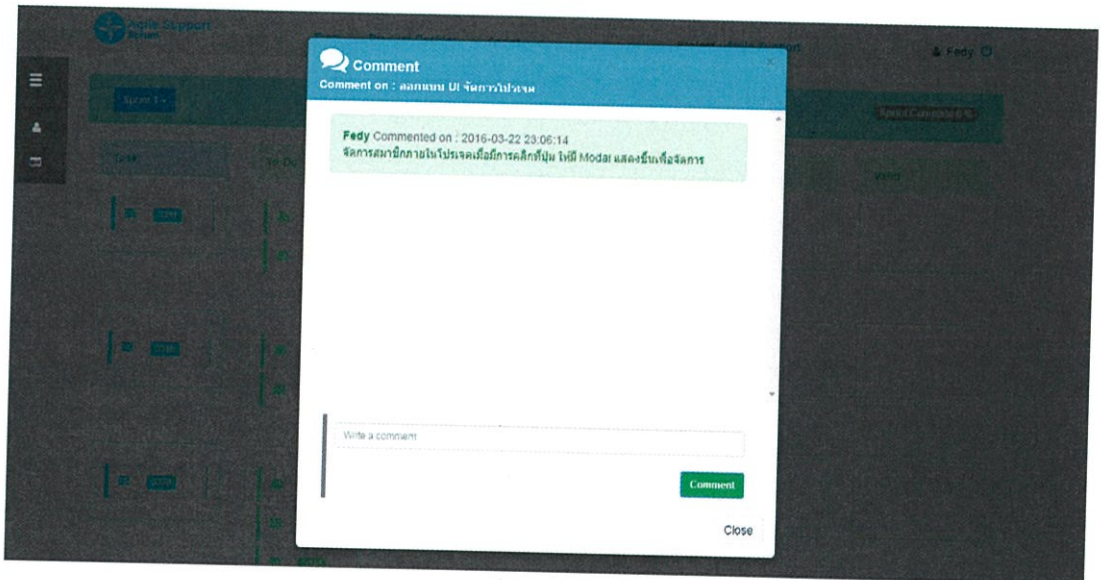
รูปที่ 4.11 หน้าจอสำหรับการเพิ่ม Attach File ให้กับงาน

จากรูปที่ 4.11 ผู้ใช้สามารถ เพิ่ม Attach File ให้กับงานและสามารถดาวน์โหลดไฟล์งานได้ เมื่อมีผู้ใช้อื่นเข้ามาแสดงความคิดเห็นในงานหรืองานย่อย ระบบจะมีการแสดงข้อความแจ้งเตือนความคิดเห็นที่ผู้ใช้อยู่ยังไม่อ่านขึ้นมารูปที่ 4.12



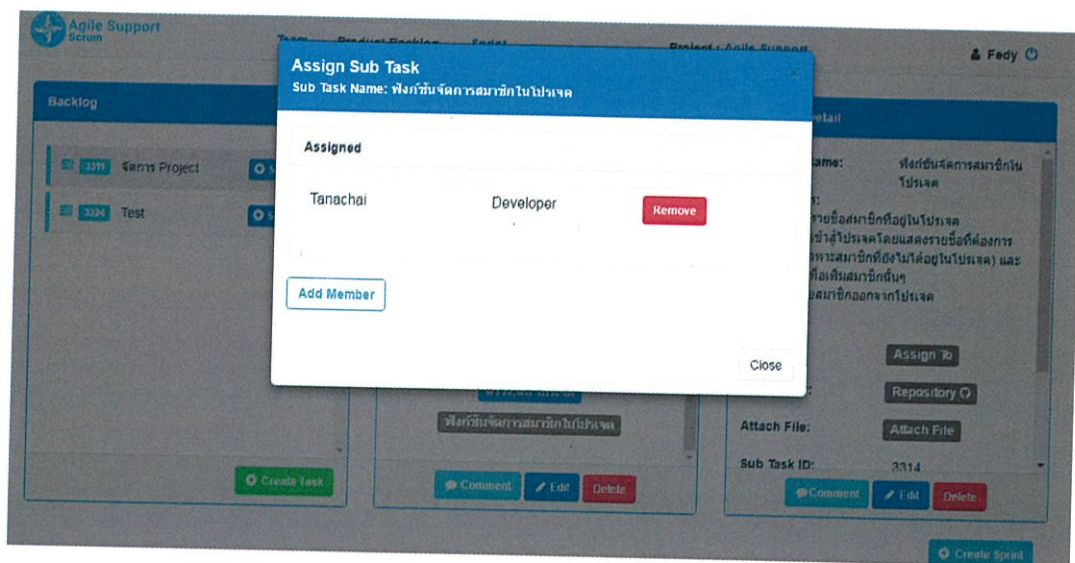
รูปที่ 4.12 หน้าจอสำหรับแสดงการแจ้งเตือนข้อความที่ยังไม่ได้อ่าน

จากรูปที่ 4.12 กรอบที่ 1 หน้าจอแสดงตัวเลขแจ้งเตือนข้อความที่ยังไม่ได้อ่าน ผู้ใช้สามารถเข้าไปอ่านและแสดงความคิดเห็นในงานได้ดังรูปที่ 4.13



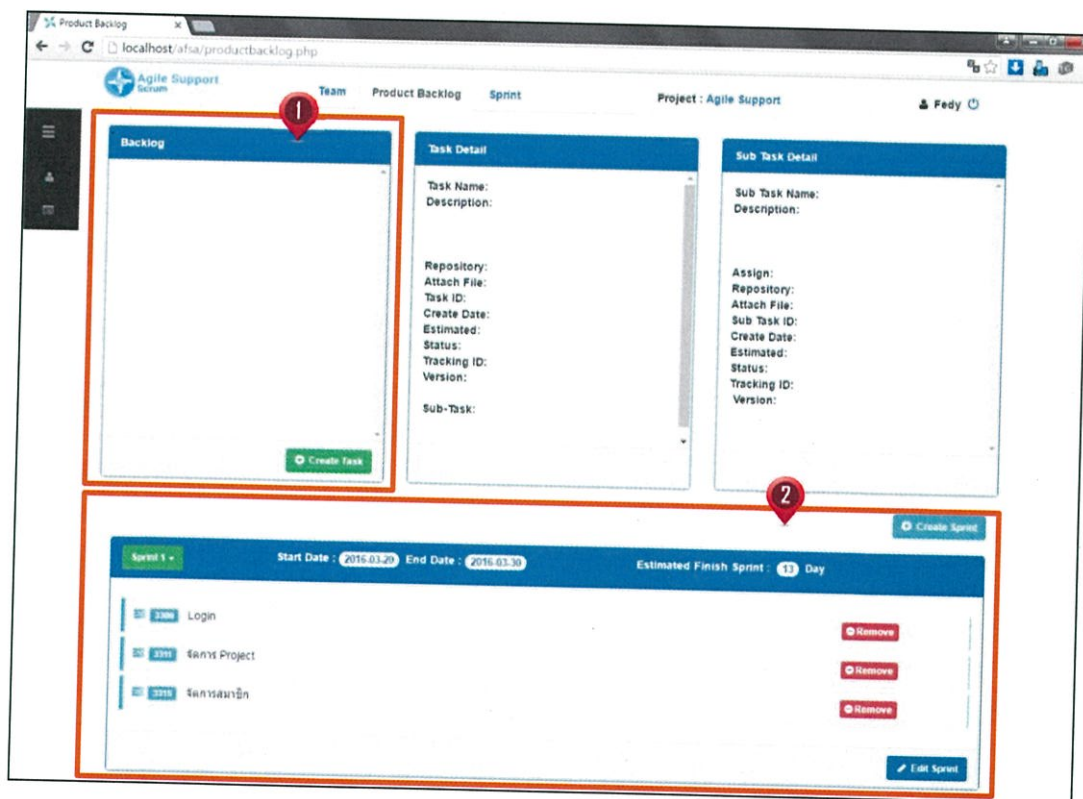
รูปที่ 4.13 หน้าจอสำหรับแสดงความคิดเห็นของงาน

ในส่วนของงานย่อย (Sub Task) ผู้ใช้สามารถมอบหมายงานให้กับสมาชิกภายในทีมได้ดังรูปที่ 4.14



รูปที่ 4.14 หน้าจอสำหรับการมอบหมายงานให้กับสมาชิกในทีม

การจัดการงานใน Product Backlog ผู้ใช้สามารถสร้างรอบการทำงาน (Sprint) และนำงานที่เข้าสู่รอบการทำงาน (Sprint) โดยแสดงดังรูปที่ 4.15

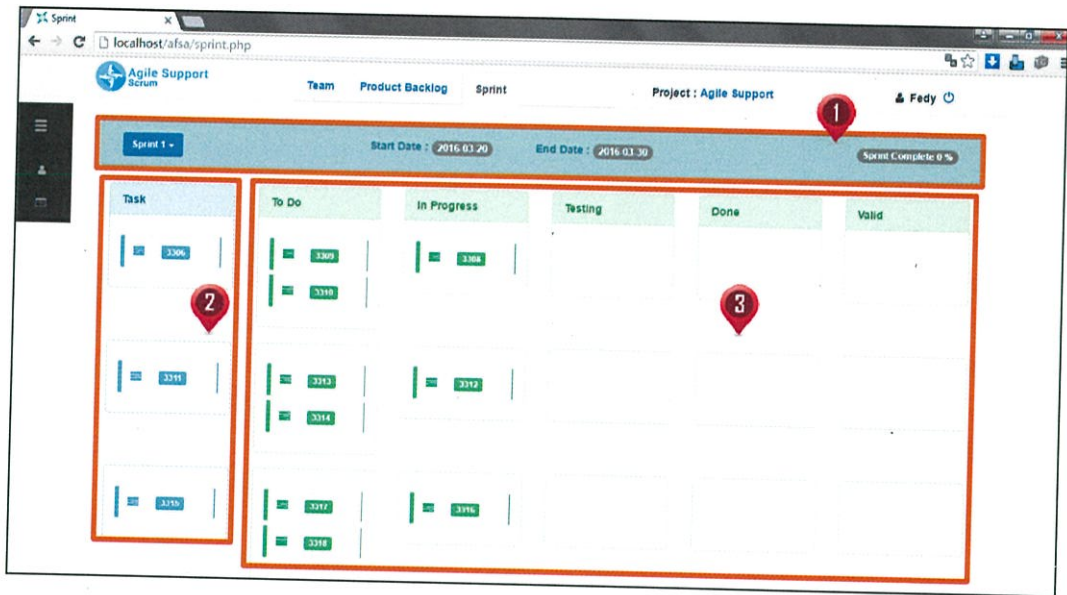


รูปที่ 4.15 หน้าจอแสดงงานในรอบการทำงาน (กรอบที่ 2)

จากรูปที่ 4.15 หลังจากที้นำงานจากกรอบที่ 1 เข้าสู่รอบการทำงานแล้วระบบจะแสดงข้อมูลงานภายในรอบการทำงานในกรอบที่ 2 โดยผู้ใช้สามารถลบงานออกจากรอบการทำงานได้ ซึ่งงานที่ถูกลบจะกลับคืนสู่กรอบที่ 1

4.2.4 การจัดการรอบการทำงาน (Sprint)

Sprint เป็นหน้าสำหรับจัดการงาน (Task) และงานย่อย (Sub Task) ที่อยู่ในรอบการทำงาน (Sprint) โดยผู้ใช้สามารถปรับเปลี่ยนสถานะของงานย่อยได้ ผู้ใช้สามารถทราบประวัติการเปลี่ยนแปลงสถานะของงานย่อยและสามารถใส่รายละเอียดข้อมูลการดำเนินการและความคืบหน้าของงานย่อยได้ สามารถสร้างเวอร์ชันของงานและงานย่อยขึ้นใหม่ พร้อมสามารถใส่ข้อมูลความแตกต่างของงานแต่ละเวอร์ชันได้ โดยระบบจะแสดงดังรูปที่ 4.16

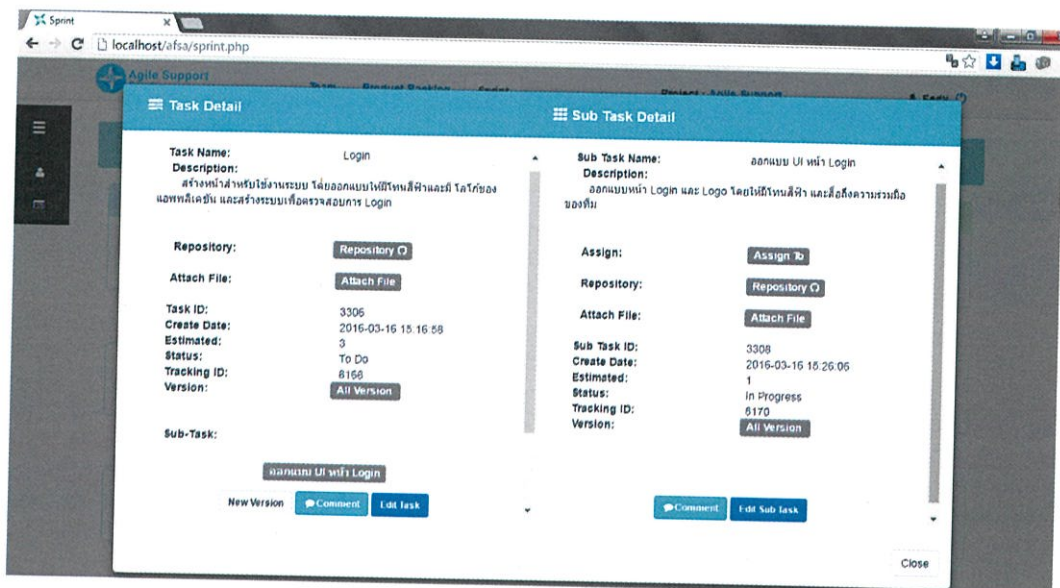


รูปที่ 4.16 หน้าจอสำหรับจัดการ Sprint

จากรูปที่ 4.16 ระบบแสดงหน้าจอสำหรับจัดการ Sprint ในรูปแบบของบอร์ดการจัดการแบบสกรีม โดยแบ่งเป็น 3 ส่วนคือ

ในรอบที่ 1 แสดงลำดับของรอบการทำงานนั้น แสดงวันที่เริ่มและวันที่สิ้นสุดของรอบการทำงาน และแสดงเปอร์เซ็นต์ความคืบหน้าของงานทั้งหมดในรอบการทำงาน ผู้ใช้สามารถเลือกลำดับรอบการทำงานได้จากกรอบนี้

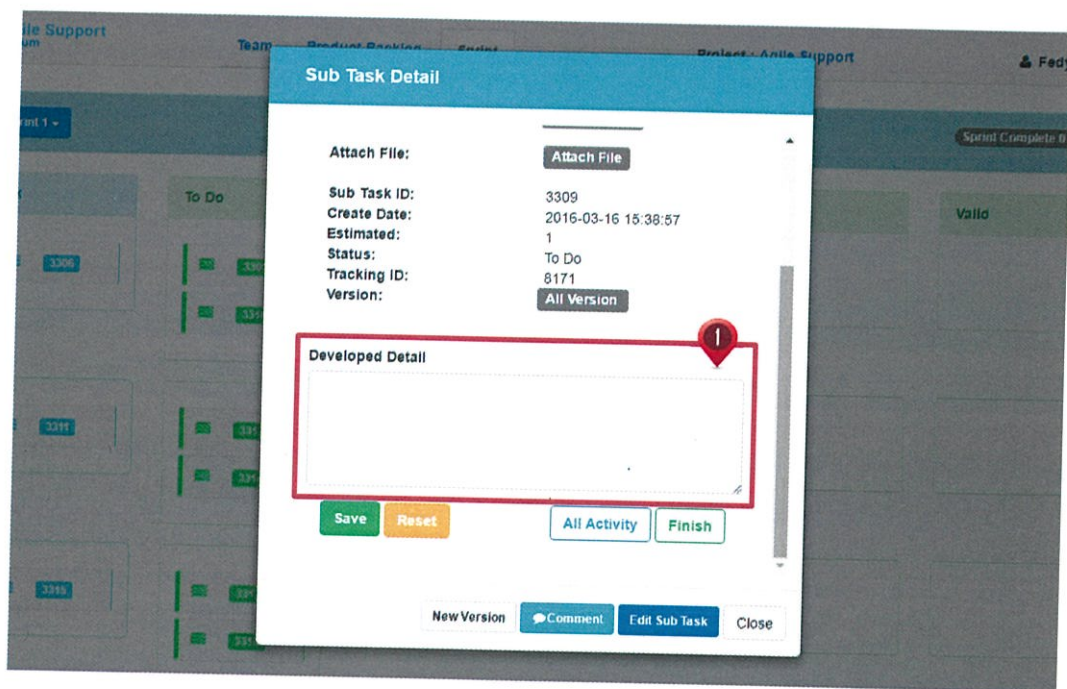
กรอบที่ 2 แสดงรายการของงาน (Task) ที่ผู้ใช้นำเข้าสู่รอบการทำงาน โดยผู้ใช้สามารถคลิกที่งานนั้นเพื่อดูรายละเอียดของงานได้ ตามรูปที่ 4.17



รูปที่ 4.17 หน้าจอแสดงข้อมูลของงานและงานย่อยในหน้า Sprint

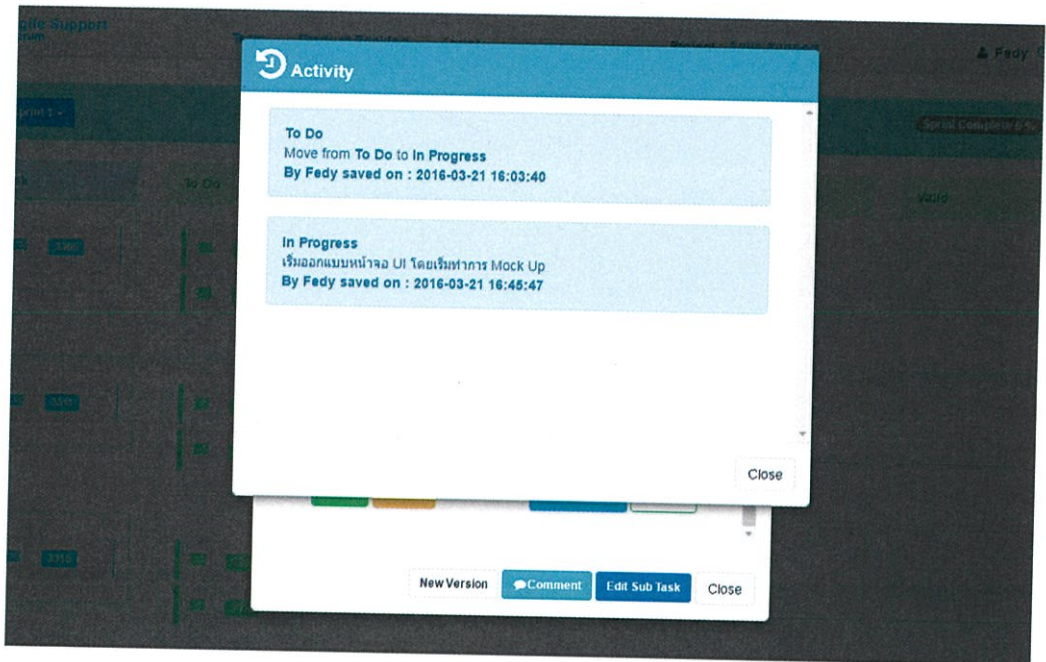
จากรูปที่ 4.17 ระบบแสดงรายละเอียดของงานและรายการงานย่อยของงานนั้น โดยผู้ใช้สามารถเรียกดูและแก้ไขรายละเอียดของงานและงานย่อยได้เหมือนในหน้าการจัดการ Product Backlog

กรอบที่ 3 แสดงรายการงานย่อย (Sub Task) ของแต่ละงาน ตามสถานะของแต่ละงานย่อย โดยผู้ใช้สามารถปรับเปลี่ยนสถานะของงานย่อยได้ ผู้ใช้สามารถดูรายละเอียดของงานย่อยได้ ตามรูปที่ 4.18



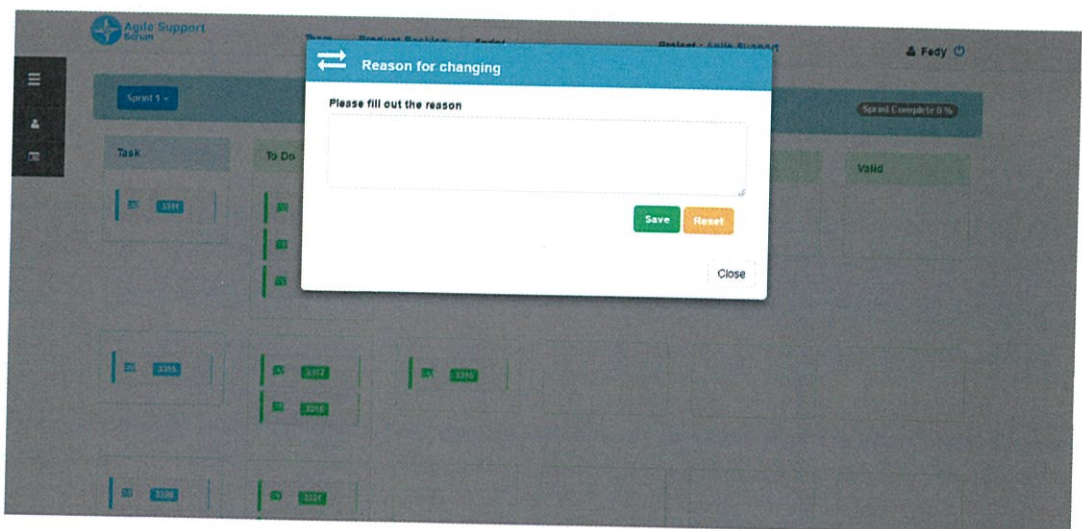
รูปที่ 4.18 หน้าจอแสดงรายละเอียดของงานย่อยในหน้า Sprint

จากรูปที่ 4.18 ระบบแสดงรายละเอียดของงานย่อย โดยผู้ใช้สามารถเรียกดูและแก้ไขรายละเอียดงานย่อยได้เหมือนในหน้าการจัดการ Product Backlog โดยผู้ใช้สามารถบันทึกการดำเนินการที่เกิดขึ้นในงานนั้นได้ดังรูปที่ 4.18 กรอบที่ 1 และระบบจะเก็บประวัติการเปลี่ยนสถานะของงาน ซึ่งจะแสดงผลตามรูปที่ 4.19



รูปที่ 4.19 หน้าจอแสดงประวัติความคืบหน้าของงานย่อยในรอบการทำงาน

จากรูปที่ 4.19 ผู้ใช้สามารถทราบประวัติการเปลี่ยนแปลงสถานะของงานย่อยและสามารถใส่รายละเอียดข้อมูลการดำเนินการและความคืบหน้าของงานย่อยได้ ซึ่งหากผู้ใช้ต้องการเลื่อนสถานะของงานย้อนกลับด้วยเหตุผลใดก็ตาม ผู้ใช้ต้องกรอกเหตุผลของการเลื่อนงานกลับด้วย ดังรูปที่ 4.20



รูปที่ 4.20 หน้าจอแสดงหน้าต่างให้ผู้ใช้กรอกเหตุผลที่เลื่อนงานย้อนหลัง

จากรูปที่ 4.20 เมื่อผู้ใช้กรอกเหตุผลที่เลื่อนงานย้อนหลัง ข้อมูลที่กรอกนั้นจะแสดงในหน้าประวัติความคืบหน้าของงาน

ในหน้าการจัดการ Sprint ผู้ใช้สามารถสร้างเวอร์ชันของงานและงานย่อยใหม่ได้ ดังรูปที่ 4.21

รูปที่ 4.21 หน้าจอแสดงข้อมูลสำหรับสร้างงานย่อยเวอร์ชันใหม่

จากรูปที่ 4.21 เมื่อผู้ใช้ต้องการสร้างงานหรืองานย่อยเวอร์ชันใหม่ ผู้ใช้สามารถใส่ข้อมูลรายละเอียดความแตกต่างระหว่างงานหรืองานย่อยเวอร์ชันเดิมกับเวอร์ชันใหม่ได้ โดยสามารถกรอกข้อมูลความแตกต่างได้ดังรูปที่ 4.21 ในกรอบที่ 1 ซึ่งผู้ใช้สามารถเรียกดูข้อมูลเวอร์ชันทั้งหมดของงานหรืองานย่อยได้ตามรูปที่ 4.22

ID	Version	Created Date and Time	Created By	Comment
3319	ออกแบบ UI หน้า Login V2	2016-03-21 16:54:57	Fedy	ปรับโทนสีเป็น เมสชั่น LOGO เป็นตามที่แบบใหม่ เพิ่ม Alert message ในกรณี login ผิดพลาด
3308	ออกแบบ UI หน้า Login	2016-03-16 16:26:06	Fedy	

รูปที่ 4.22 หน้าจอแสดงเวอร์ชันทั้งหมดของงานย่อย

จากรูปที่ 4.22 ระบบจะแสดงรายละเอียดของงานหรืองานย่อยทุกเวอร์ชัน ในกรอบที่ 1 แสดงข้อมูลของงานเวอร์ชันล่าสุด ในกรอบที่ 2 แสดงข้อมูลของงานเวอร์ชันอื่นๆ ซึ่งผู้ใช้สามารถคลิกดูข้อมูลรายละเอียดทั้งหมดของงานหรืองานย่อยทุกเวอร์ชันได้ และผู้ใช้สามารถเรียกคืนงานหรืองานย่อยเวอร์ชันเดิมขึ้นแทนงานหรืองานย่อยเวอร์ชันล่าสุดได้

ในกรณีผู้ใช้สร้างงานเวอร์ชันใหม่ หากงานเวอร์ชันเดิมมีงานย่อย ระบบจะนำข้อมูลงานย่อยเวอร์ชันล่าสุดทุกงานย่อยของงานเวอร์ชันเดิมเข้าสู่งานเวอร์ชันใหม่ให้ทันที

บทที่ 5

สรุปผลการดำเนินงาน และข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

กระบวนการพัฒนาระบบแบบบอโจล์ เป็นกระบวนการพัฒนาระบบที่ได้รับความนิยมในปัจจุบัน ซึ่งกระบวนการดังกล่าวก็เป็นเหมือนกับกระบวนการพัฒนาระบบด้วยวิธีอื่นๆ ซึ่งต้องมีการสร้างทีมทำงาน มีการแบ่งงาน มอบหมายงานและติดตามความคืบหน้าของงาน ในปัจจุบันมีแอปพลิเคชันที่ช่วยการทำงานแบบบอโจล์อยู่บ้าง แต่ยังไม่มีความสมบูรณ์ในแง่ของการติดตามการทำงาน แอปพลิเคชันสำหรับสนับสนุนกระบวนการพัฒนาแบบบอโจล์ที่พัฒนาขึ้นมา มีจุดประสงค์เพื่อช่วยในการติดตามงานในกระบวนการพัฒนาซอฟต์แวร์แบบบอโจล์ให้สมบูรณ์ขึ้น โดยระบบที่พัฒนาขึ้นเป็นระบบเว็บแอปพลิเคชัน โดยผู้ใช้ที่มีบทบาทเป็น Product Owner จะเป็นผู้สร้างโปรเจกต์และเพิ่มสมาชิกเข้าสู่ทีมพัฒนา จากนั้นสมาชิกสามารถสร้างงานและงานย่อย และมอบหมายให้กับสมาชิกในทีม โดยสมาชิกในทีมทุกคนสามารถดูรายละเอียดและติดตามความคืบหน้าของงานและงานย่อยได้ โดยสามารถติดตามได้จากสกรีนบอร์ดตามรอบการทำงาน สมาชิกสามารถแสดงความคิดเห็นเกี่ยวกับงานหรืองานย่อยนั้น ในส่วนของซอร์สโค้ดผู้พัฒนาได้ใช้ Github เป็นแหล่งที่เก็บซอร์สโค้ดเพื่อทำให้การบริหารจัดการซอร์สโค้ดเป็นไปโดยได้สะดวก นอกจากนี้สมาชิกสามารถแนบไฟล์งานที่เกี่ยวข้องกับงานหรืองานย่อย เช่นไฟล์รูปภาพการออกแบบจอภาพเพื่อให้สมาชิกภายในทีมได้ออกความคิดเห็น ระบบยังสนับสนุนการบริหารจัดการเวอร์ชันของงานหรืองานย่อย เพื่อให้ทีมงานสามารถที่ย้อนกลับมาใช้การออกแบบหรือพัฒนาในเวอร์ชันก่อนหน้าได้โดยง่าย

5.2 ข้อเสนอแนะของปัญหาพิเศษ

1. พัฒนาโปรแกรมให้อยู่ในรูปแบบของโมบายแอปพลิเคชัน

ในปัจจุบันแอปพลิเคชันสำหรับสนับสนุนกระบวนการพัฒนาแบบบอโจล์ รองรับการทำงานในรูปแบบเว็บแอปพลิเคชันเท่านั้น ซึ่งสามารถพัฒนาต่อยอดให้โปรแกรมสามารถทำงานในรูปแบบของโมบายแอปพลิเคชันได้

2. พัฒนาโปรแกรมให้มีระบบแจ้งเตือนการอัปเดตข้อมูลภายในโปรเจกต์อัตโนมัติ

เพื่อให้ผู้ใช้สามารถรับรู้ความเคลื่อนไหวภายในโปรเจกต์ ควรมีการพัฒนาให้แอปพลิเคชันสามารถแจ้งเตือนการอัปเดตข้อมูลต่างๆภายในโปรเจกต์ให้ผู้ใช้ทราบได้โดยอัตโนมัติ เช่น มีการแจ้งเตือนเมื่อมีผู้ทำอันสร้างงานใหม่ (Task) เมื่องานย่อย (Sub Task) มีการเปลี่ยนสถานะในรอบการทำงาน เป็นต้น

3. พัฒนาโปรแกรมให้สามารถสนับสนุนการมอบหมายงานให้เหมาะสมกับสมาชิกในทีม

ควรพัฒนาแอปพลิเคชันให้สามารถเก็บข้อมูลทักษะความสามารถของผู้ใช้ เพื่อให้สามารถมอบหมายงานได้ตรงกับทักษะของผู้ใช้ และเก็บข้อมูลปริมาณงานที่ผู้ใช้รับผิดชอบอยู่ เพื่อให้สามารถมอบหมายงานให้ผู้ใช้ในปริมาณที่เหมาะสม

เอกสารอ้างอิง

- [1] Agile แนวความคิดใหม่ในการพัฒนา Software. , URL:<http://sites.google.com/a/acith.com/acith/home/ag> , สืบค้นเมื่อ 5 กันยายน 2558
- [2] Scrum Reference Card. , URL: <http://scrumreferencecard.com/ScrumReferenceCard.pdf> , สืบค้นเมื่อ 5 กันยายน 2558
- [3] Scrum Task Board. , URL: <http://www.mountangoatsoftware.com/agile/scrum/task-boards> , สืบค้นเมื่อ 5 กันยายน 2558
- [4] Nui, Z. , Yang, C. and Zhang, Y. , “A Design of Cross - terminal Web System Based on JSON and REST”, In Proceedings of 5th Beijing International Conference on Software Engineering and Service Science (ICSESS) , June 2014 , China .
- [5] Github API. , URL: <http://developer.github.com/v3> , สืบค้นเมื่อ 5 กันยายน 2558