

ส่วนการบริการตนเองของระบบการจัด
การอาร์ติแฟคแบบไบนารี
SELF-SERVICE MANAGEMENT OF
BINARY ARTIFACT MANAGEMENT SYSTEM (BAMS)



สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SELF-SERVICE MANAGEMENT OF
BINARY ARTIFACT MANAGEMENT SYSTEM (BAMS)





TANATIP LUEANG-ON
AKKRAWUT KHANTHAWAN

A COOPERATIVE EDUCATION SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT FOR
THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE FACULTY OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2017

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สหกิจศึกษา	ส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟกต์แบบไบนารี Self-Service Management of Binary Artifact Management System (BAMS)	
ชื่อนักศึกษา	นาย ธนาธิป เหลืองอ่อน	รหัสนักศึกษา 57050246
	นาย อัครวุฒิ ชันธวรรณ	รหัสนักศึกษา 57050363
ปริญญา	วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)	
ภาควิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2560	
อาจารย์ที่ปรึกษา	ผศ.ดร.อนันตพร ทรราชคุณาฒย	

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้
สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา วิทยาศาสตร์บัณฑิต (วิทยาการ
คอมพิวเตอร์) ประจำปีการศึกษา 2560

คณะกรรมการสอบ	ลายมือชื่อ
อ.วิสันต์ ตั้งวงษ์เจริญ ประธานกรรมการ	
ผศ.ดร.อนันตพร ทรราชคุณาฒย กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	ส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟคแบบไบนารี	
ชื่อนักศึกษา	นาย ธนาธิป เหลืองอ่อน	รหัสนักศึกษา 57050246
	นาย อัครวุฒิ ชันธวรรณ	รหัสนักศึกษา 57050363
ปริญญา	วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)	
ภาควิชา	วิทยาการคอมพิวเตอร์	
คณะ	วิทยาศาสตร์	
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)	
ปีการศึกษา	2560	
อาจารย์ที่ปรึกษา	ผศ.ดร.อนันตพร หาราชคุณาฉัย	

บทคัดย่อ

เล่มสหกิจศึกษาฉบับนี้นำเสนอการพัฒนาระบบเว็บแอปพลิเคชันและส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชัน (API) ส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟคแบบไบนารี ซึ่งเป็นระบบส่วนกลางภายในองค์กร ที่มีหน้าที่ในการจัดการ ที่จัดเก็บข้อมูล (Repository) ที่อยู่ในระบบการจัดการอาร์ติแฟคแบบไบนารี (Binary Artifact Management System: BAMS) เนื่องจากในปัจจุบันระบบนั้นมี ส่วนต่อประสานกับผู้ใช้ (User Interface) ที่ค่อนข้างใช้งานได้ยาก ไม่เหมาะสมสำหรับผู้ใช้ทั่วไปภายในองค์กร รวมถึงส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชันยังขาดฟังก์ชันในส่วนที่จำเป็น ในการพัฒนานั้นผู้จัดทำได้วิเคราะห์ความต้องการของผู้ใช้งาน พร้อมทั้งออกแบบระบบเว็บแอปพลิเคชัน และส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชัน ที่สามารถใช้งานร่วมกับระบบการจัดการอาร์ติแฟคแบบไบนารีได้อย่างมีประสิทธิภาพ โดยใช้รูปแบบการทำงาน แบบ Agile ในรูปแบบของ Scrum ซึ่งระบบนั้น ในส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชันจะถูกพัฒนาด้วยภาษา TypeScript ส่วนเว็บแอปพลิเคชันจะถูกพัฒนาโดยใช้เฟรมเวิร์ก React

คำสำคัญ : เฟรมเวิร์ก React ภาษา TypeScript เว็บแอปพลิเคชัน ส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชัน (API)

Title	Self-Service Management of Binary Artifact Management System (BAMS)	
Students	Mr. Tanatip Lueang-on	Student ID 57050246
	Mr. Akkrawut Khanthawan	Student ID 57050363
Degree	Bachelor of Science (Computer Science)	
Department	Computer Science	
Faculty	Science	
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)	
Academic Year	2017	
Advisor	Asst.Prof.Dr. Anantaporn Hanskunatai	

Abstract

This cooperative workbook presents the development of a web application system and application programming interface (API) for Self-Service Management of Binary Artifact Management System. The system is a centralized system within the organization that manages the repositories in the Binary Artifact Management System (BAMS). At present, the user interface of the system is unsuitable for general users in the organization. The service interface of the application also lacks the necessary functionalities. In the development, the user requirements were analyzed. Then the web application and service interfaces of the application were designed compatible with the BAMS by using Agile methodology with Scrum. In a part of service interface of the application, it was developed by the TypeScript language. For the web application, it was developed by using the React Framework.

Keywords: React Framework, TypeScript, Web Application, Application programming interface (API)

กิตติกรรมประกาศ

การทำสหกิจศึกษาหัวข้อ ส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟกต์แบบไบนารี ฉบับนี้ สามารถดำเนินงานสำเร็จได้ด้วยดีเนื่องจากได้รับความช่วยเหลือ การสนับสนุน และความกรุณาของบุคคลหลายท่าน คณะผู้จัดทำขอกล่าวคำขอบพระคุณบุคคลดังต่อไปนี้

ขอขอบพระคุณบิดา มารดา ผู้ให้การดูแล อบรมสั่งสอน ให้การสนับสนุนด้านทุนการศึกษา ให้คำปรึกษาและคอยเป็นกำลังใจพร้อมทั้งแนะแนวทางที่สำคัญกับการใช้ชีวิต

ขอขอบพระคุณ ผศ.ดร.อนันตพร หารรรษคุณาพัฒน์ อาจารย์ที่ปรึกษาปัญหาพิเศษ ที่ได้ให้คำปรึกษา พร้อมทั้งแนะแนวทางแก้ปัญหา รวมทั้งตรวจแก้ปัญหาพิเศษฉบับนี้ให้มีความสมบูรณ์เพิ่มขึ้น

ขอขอบพระคุณพี่ในทีม Platform Application Management (PAM) ที่ให้โอกาสพร้อมทั้งคอยแนะนำให้ความรู้คำแนะนำ และคำปรึกษาในการทำงาน ตลอดระยะเวลาในการฝึกงาน

ขอขอบพระคุณ บริษัท รอยเตอร์ซอฟต์แวร์ (ประเทศไทย) จำกัด ที่ให้โอกาสในการเข้ารับการฝึกงานและได้รับประสบการณ์ในการทำงานในสภาพแวดล้อมจริง

ขอขอบพระคุณอาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้ความรู้และประสบการณ์ตลอดระยะเวลาที่ได้ทำการศึกษา

ธนธิป เหลืองอ่อน
อัศวุฒิ ชันชรวรรณ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ช
สารบัญรูป.....	ซ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	1
1.3 ขอบเขตของงานวิจัย.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 เครื่องมือที่ใช้.....	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 ระเบียบวิธีที่ใช้ในการพัฒนาระบบ.....	4
2.1.1 องค์ประกอบของ Scrum Methodology.....	5
2.1.2 กระบวนการทำงานของ Scrum Methodology.....	5
2.2 ระบบการจัดการอาร์ติแฟคแบบไบนารี.....	7
2.2.1 การระบุตัวตนในการเข้าใช้งานระบบ.....	8
2.2.2 การกำหนดชื่อของที่จัดเก็บอาร์ติแฟคแบบไบนารี.....	8
2.2.3 โครงสร้างภายในของระบบ.....	9
2.3 ส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟคแบบไบนารี.....	11
2.4 การพัฒนาส่วนติดต่อเซอร์วิสของแอปพลิเคชัน.....	11
2.4.1 ส่วนติดต่อเซอร์วิสของแอปพลิเคชัน.....	12
2.4.2 Restful API.....	12
2.4.3 การออกแบบ Restful API.....	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.5 ภาษาและเครื่องมือที่นำมาใช้ในการพัฒนาระบบ	14
2.5.1 JFrog Artifactory	14
2.5.2 Node.js	15
2.5.3 JavaScript (ES6)	16
2.5.4 Postman	17
2.5.5 Swagger	18
2.5.6 React.js	18
2.5.7 JSON Web Token	19
2.5.8 เครื่องมือที่ใช้ในการทำ Unit Test	20
2.5.9 Jenkins	20
บทที่ 3 ขั้นตอนในการพัฒนาระบบ	21
3.1 ขั้นตอนการศึกษาและเก็บรวบรวมข้อมูลของระบบ	21
3.2 ขั้นตอนการวิเคราะห์ระบบ	22
3.2.1 สถาปัตยกรรมของระบบ	22
3.2.2 Use Case Diagram	23
3.3 ขั้นตอนการออกแบบระบบ	24
3.3.1 Sequence Diagram	24
3.4 ขั้นตอนการออกแบบการทดสอบระบบ	33
3.5 ขั้นตอนการพัฒนาระบบ	39
บทที่ 4 ผลการพัฒนาระบบ	40
4.1 หน้าจอแสดงผลส่วนติดต่อผู้ใช้งานเว็บแอปพลิเคชัน	40
4.2 ผลการทดสอบเฉพาะส่วนของ BAMS API	47
บทที่ 5 สรุปผลการพัฒนาระบบและข้อเสนอแนะ	52
5.1 สรุปผลการวิจัย	52
5.2 ข้อจำกัดของระบบ	52
5.3 ข้อเสนอแนะ	52

สารบัญ(ต่อ)

	หน้า
เอกสารอ้างอิง	53
ภาคผนวก.....	54
ภาคผนวก ก การเตรียมเครื่องมือเพื่อใช้ในการพัฒนาระบบ.....	55



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 การทดสอบการใช้งาน LDAP ในระบบ BAMS API	34
3.2 การทดสอบการเข้าสู่ระบบการใช้งานของ BAMS API	34
3.3 การทดสอบระบบการสร้างพื้นที่จัดเก็บข้อมูล	35
3.4 การทดสอบระบบการอัปเดตพื้นที่จัดเก็บข้อมูล	36
3.5 การทดสอบระบบการลบพื้นที่จัดเก็บข้อมูล	36
3.6 การทดสอบระบบการแสดงรายละเอียดสิทธิ์การจัดการของผู้ใช้งานใน พื้นที่จัดเก็บข้อมูล	37
3.7 การทดสอบระบบการเพิ่มสิทธิ์การจัดการของผู้ใช้งาน	37
3.8 การทดสอบระบบการลบสิทธิ์การจัดการของผู้ใช้งาน	38
3.9 การทดสอบระบบการเปิดเผยหรือปกปิดพื้นที่จัดเก็บข้อมูลของผู้ใช้งาน	39
4.1 ผลการทดสอบการใช้งาน LDAP ในระบบ BAMS API	47
4.2 ผลการทดสอบการเข้าสู่ระบบการใช้งานของ BAMS API.....	47
4.3 ผลการทดสอบระบบการสร้างพื้นที่จัดเก็บข้อมูล	48
4.4 ผลการทดสอบระบบการอัปเดตพื้นที่จัดเก็บข้อมูล	48
4.5 ผลการทดสอบระบบการลบพื้นที่จัดเก็บข้อมูล	49
4.6 ผลการทดสอบระบบการแสดงรายละเอียดสิทธิ์การจัดการของผู้ใช้งานใน พื้นที่จัดเก็บข้อมูล	49
4.7 ผลการทดสอบระบบการเพิ่มสิทธิ์การจัดการของผู้ใช้งาน	50
4.8 ผลการทดสอบระบบการลบสิทธิ์การจัดการของผู้ใช้งาน.....	51
4.9 ผลการทดสอบระบบการเปิดเผยหรือปกปิดพื้นที่จัดเก็บข้อมูลของผู้ใช้งาน.....	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 องค์ประกอบของ Scrum Methodology.....	4
2.2 ส่วนหน้าจอในการจัดการ Product Backlog (Jira Dashboard).....	6
2.3 ส่วนหน้าจอผู้ใช้งานของ Binary Artifact Management System.....	7
2.4 โครงสร้างของการวางเครื่องเซิร์ฟเวอร์ที่กระจายไปในแต่ละภูมิภาค	9
2.5 โครงสร้างของการวางเครื่องเซิร์ฟเวอร์ที่ถูกปรับปรุงจากของเดิม	10
2.6 ส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟกแบบไบนารี	11
2.7 Flow การร้องขอและตอบสนองของ Restful API.....	12
2.8 ความสัมพันธ์ของเครื่องมือ อาร์ติแฟก หรือ แพ็คเกจต่าง ๆ ที่ JFrog Artifactory รองรับในการใช้งาน	15
2.9 Node.js.....	15
2.10 JavaScript.....	16
2.11 ความสัมพันธ์ของ TypeScript และ JavaScript.....	16
2.12 Postman.....	17
2.13 Swagger	18
2.14 React.js	18
2.15 Mocha.js.....	19
2.16 Jenkins.....	20
2.17 ส่วนหน้าจอผู้ใช้งานของ Jenkins	20
3.1 โครงสร้างสถาปัตยกรรมของระบบ	22
3.2 แผนภาพ Use Case Diagram ของระบบ	23
3.3 แผนภาพ Sequence Diagram แสดงการสร้างพื้นที่จัดเก็บข้อมูล	25
3.4 แผนภาพ Sequence Diagram แสดงการลบพื้นที่จัดเก็บข้อมูล	27
3.5 แผนภาพ Sequence Diagram แสดงการแสดงรายละเอียดสิทธิ์การจัดการของผู้ใช้งานใน พื้นที่จัดเก็บข้อมูล	28
3.6 แผนภาพ Sequence Diagram แสดงการเพิ่มสิทธิ์การจัดการของผู้ใช้งาน	30
3.7 แผนภาพ Sequence Diagram แสดงการลบสิทธิ์การจัดการของผู้ใช้งาน.....	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.1 หน้าเว็บแอปพลิเคชัน ในส่วนของหน้ายืนยันการเข้าสู่ระบบ.....	40
4.2 หน้าจอหลักเว็บแอปพลิเคชัน.....	41
4.3 ส่วนของหน้าจอ สร้างที่จัดเก็บข้อมูล (Create Repository).....	42
4.4 ส่วนการเลือกข้อมูล ในขั้นตอนการสร้างที่จัดเก็บข้อมูล.....	43
4.5 ส่วนการเลือกข้อมูล ในขั้นตอนการลบที่จัดเก็บข้อมูล.....	43
4.6 ส่วนการเลือกข้อมูล ในขั้นตอนการตั้ง Public/Private ให้ที่จัดเก็บข้อมูล.....	44
4.7 ส่วนของหน้าจอ แสดงสิทธิของผู้ใช้ (Show).....	44
4.8 ส่วนของหน้าจอ เพิ่มสิทธิของผู้ใช้ (Add User).....	45
4.9 ส่วนของหน้าจอเมื่อมีการใช้งานฟังก์ชัน เมื่อมีการประมวลผลสำเร็จ.....	46
4.10 ส่วนของหน้าจอเมื่อมีการใช้งานฟังก์ชัน เมื่อมีการประมวลผลไม่สำเร็จ.....	46
ก.1 หน้า Setup ของ Node.js.....	55
ก.2 หน้ายืนยันข้อตกลง ของ Node.js.....	55
ก.3 หน้าเลือกตำแหน่งติดตั้ง ของ Node.js.....	56
ก.4 หน้าแสดงรายการโปรแกรม ของ Node.js.....	56
ก.5 หน้าติดตั้ง ของ Node.js.....	57
ก.6 หน้าสิ้นสุดการติดตั้ง ของ Node.js.....	57
ก.7 ทดสอบการติดตั้ง ของ Node.js.....	58

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในการพัฒนาซอฟต์แวร์ในแผนกต่างๆ ของบริษัท รอยเตอร์ซอฟต์แวร์ (ประเทศไทย) จำกัด จะมีการจัดเก็บไฟล์ของซอฟต์แวร์ที่บริษัทพัฒนาขึ้นมาเก็บไว้ในรูปแบบต่างๆ ไม่ว่าจะเป็นในรูปแบบของ Source code หรือ อาร์ติแฟกต์แบบไบนารี (Binary Artifact) ดังนั้นเมื่อมีการพัฒนาซอฟต์แวร์จนเสร็จสิ้นพร้อมใช้งานแล้วก็ต้องมีการนำซอฟต์แวร์หรือแพ็คเกจไปรวบรวมเพื่อนำไปติดตั้งเพื่อนำไปใช้งานจริงในเซิร์ฟเวอร์ที่ระบบนั้น ๆ ต้องการ ดังนั้นทำให้ต้องมีการสร้าง ระบบการจัดการอาร์ติแฟกต์แบบไบนารี ขึ้นมาเพื่อจัดเก็บซอฟต์แวร์ หรือ แพ็คเกจต่าง ๆ เพื่อให้ผู้ใช้ที่ต้องการใช้สามารถเรียกใช้งาน เข้าถึง ค้นหาเพื่อนำไปใช้งาน และติดตั้งลงบนเครื่องเซิร์ฟเวอร์ที่ต้องการใช้งานได้ง่ายสะดวก ระบบการจัดการอาร์ติแฟกต์แบบไบนารี (Binary Artifact Management System: BAMS) จึงเป็นเว็บแอปพลิเคชันที่ทำหน้าที่ในการแสดงข้อมูลของที่จัดเก็บข้อมูล (Repository) ซึ่ง BAMS นั้นมีส่วนที่เรียกว่า ส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟกต์แบบไบนารี (BAMS-self-service) เป็นส่วนเว็บแอปพลิเคชัน อีกส่วนที่สามารถให้ผู้ใช้จัดการกับ BAMS โดยผู้ใช้สามารถจัดการกับ Repository อีกทั้งผู้ใช้สามารถเพิ่มผู้ที่มีสิทธิ์เข้าถึงและกำหนดสิทธิ์ในการใช้งานให้กับผู้ใช้แต่ละคนกับ Repository ได้

เนื่องจากในปัจจุบันระบบ BAMS Self-Service นั้นมี User Interface ที่ค่อนข้างใช้งานได้ยาก ไม่เหมาะสมสำหรับผู้ใช้งานทั่วไปภายในองค์กร อีกทั้งส่วนติดต่อเซอร์วิสของแอปพลิเคชัน (API) ยังขาดฟังก์ชันในส่วนต่าง ๆ เช่น การกำหนดให้ผู้ใช้สามารถสร้าง Repository ได้เอง เป็นต้น ทำให้ต้องมีการปรับปรุงในส่วนของเว็บแอปพลิเคชันของ BAMS Self-Service และ ส่วนของ API เพื่อประสิทธิภาพในการใช้งานร่วมกับ BAMS

1.2 วัตถุประสงค์ของงานวิจัย

- 1) ศึกษาและทำความเข้าใจกระบวนการพัฒนาซอฟต์แวร์โดยอาศัยหลักการ Agile Methodology แบบ Scrum development
- 2) ทำการออกแบบและพัฒนาระบบ BAMS Self-Service Management

1.3 ขอบเขตของงานวิจัย

- 1) ระบบสามารถให้บริการจัดการแบบ Self-service แก่ BAMS (Binary Artifact Management System)
- 2) ระบบสามารถใช้งานและแสดงผลผ่านทาง Web Browsers โดยผ่านการออกแบบ User interface ที่เป็นมิตรกับผู้ใช้มากยิ่งขึ้น
- 3) ระบบนั้นสามารถติดตั้งลงบนเซิร์ฟเวอร์และสามารถเรียกใช้งานระบบเว็บแอปพลิเคชันได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ในส่วนของประโยชน์ที่คาดว่าจะได้รับ สามารถจำแนกได้ออกเป็นสองส่วนด้วยกัน ได้แก่

- 1) ประโยชน์ต่อผู้พัฒนา
 - 1.1) ได้รับความรู้ความเข้าใจเกี่ยวกับการพัฒนาโปรแกรมโดยอาศัย Agile Methodology
 - 1.2) ได้รับความรู้เกี่ยวกับกระบวนการพัฒนาซอฟต์แวร์โดยอาศัยหลักการของ Scrum Methodology
 - 1.3) ได้รับความรู้ความเข้าใจเกี่ยวกับภาษาที่ใช้ในการพัฒนาเว็บแอปพลิเคชัน เช่น REST API, TypeScript, React.JS เป็นต้น
 - 1.4) ได้รับความรู้ความเข้าใจเกี่ยวกับการพัฒนาเว็บแอปพลิเคชัน ทั้งด้านการแสดงผลและการประมวลผล
 - 1.5) มีประสบการณ์จริงในการทำงานภายในองค์กร
- 2) ประโยชน์ต่อองค์กร
 - 2.1) ส่วนติดต่อเซิร์ฟเวอร์ของเว็บแอปพลิเคชัน มีความสอดคล้องกับการเปลี่ยนแปลงของโครงสร้างภายในของระบบในช่วงเวลานั้นขององค์กร
 - 2.2) ส่วนหน้าจของผู้ใช้ ของเว็บแอปพลิเคชันได้รับการปรับปรุงให้ผู้ใช้สามารถใช้งานได้สะดวกขึ้นกว่าของเดิมที่องค์กรใช้อยู่

1.5 เครื่องมือที่ใช้

1) ฮาร์ดแวร์

1.1) เครื่องคอมพิวเตอร์

- หน่วยประมวลผลกลาง (CPU) Intel core i5-3210m @2.5 GHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หน่วยความจำหลัก (RAM) 8GB 1600MHz
- หน่วยความจำสำรอง 320GB 1600MHz
- การ์ดจอ (Graphic processor) Intel HD Graphics 4000
- Windows 7 Enterprise 64-bit Operating System ใช้เป็นระบบปฏิบัติการ

2) ซอฟต์แวร์

- 2.1) JFrog Artifactory (API) ใช้สำหรับเก็บแพ็คเกจและไบนารี
- 2.2) Visual Studio Code ใช้ในการเขียนเว็บแอปพลิเคชันและส่วนติดต่อเซิร์ฟเวอร์ของเว็บแอปพลิเคชัน
- 2.3) Google Chrome ใช้เป็นเว็บเบราว์เซอร์
- 2.4) Postman และ Swagger ใช้ในการทดสอบส่วนติดต่อเซิร์ฟเวอร์ของเว็บแอปพลิเคชัน
- 2.5) Mocha, Chai และ Sinon ใช้ในการทำ Unit Test ของส่วนติดต่อเซิร์ฟเวอร์ของเว็บแอปพลิเคชัน
- 2.6) React.js ใช้เป็น Framework สำหรับการทำหน้าเว็บแอปพลิเคชัน
- 2.7) mRemote NG ใช้ในการเข้าถึงเครื่องเซิร์ฟเวอร์
- 2.8) Jenkins ใช้ในการช่วยจัดการ การติดตั้ง (Deploy) ระบบลงเซิร์ฟเวอร์
- 2.9) Justinmind ใช้ในการออกแบบเว็บแอปพลิเคชัน
- 2.10) Jira Dashboard ใช้ในการช่วยวางแผนในการทำงาน

3) ภาษา

- 3.1) JavaScript (TypeScript)
- 3.2) HTML
- 3.3) CSS
- 3.4) Shell Script Linux

บทที่ 2

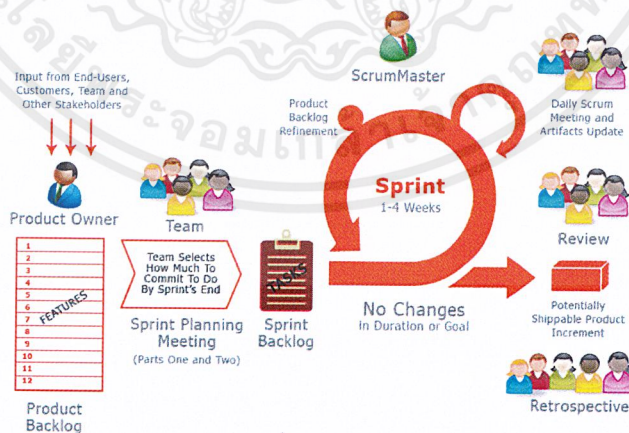
ทฤษฎีและเทคโนโลยีที่ใช้ในการพัฒนาระบบ

ในบทนี้จะอธิบายถึงรายละเอียดของส่วนประกอบของเทคโนโลยีต่าง ๆ และข้อมูลพื้นฐานที่ควรทราบ พร้อมทั้งนำความรู้ที่ได้ไป ใช้ควบคู่กับการพัฒนาระบบเว็บแอปพลิเคชัน ส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟคแบบไบนารี โดยแบ่งเป็นหัวข้อต่าง ๆ ดังต่อไปนี้

- 2.1 ระเบียบวิธีที่ใช้ในการพัฒนาระบบ (Methodology)
- 2.2 ระบบการจัดการอาร์ติแฟคแบบไบนารี (Binary Artifact Management System)
- 2.3 ส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟคแบบไบนารี (BAMS-Self service)
- 2.4 ส่วนติดต่อเซอร์วิสของแอปพลิเคชัน (Application Programming Interface)
- 2.5 ภาษาและเครื่องมือที่นำมาใช้ในการพัฒนาระบบ

2.1 ระเบียบวิธีที่ใช้ในการพัฒนาระบบ (Methodology)

Scrum Methodology เป็นหนึ่งในระเบียบวิธีที่ใช้ในการพัฒนาซอฟต์แวร์ซึ่ง Scrum เป็นหนึ่งในวิธีการพัฒนาซอฟต์แวร์ของ Agile โดยที่การพัฒนาซอฟต์แวร์แบบ Agile นั้นจะเน้นเป็นรูปแบบการทำงานที่สมาชิกภายในกลุ่มผู้พัฒนาซอฟต์แวร์ต่างผลักดันซึ่งกันและกัน และเน้นให้ความสำคัญกับการ ทำงานร่วมกันโดยจะมีการประชุมอย่างสม่ำเสมอ การพัฒนาด้วยระเบียบวิธีดังกล่าวผู้พัฒนาระบบจะต้อง ทำงานอย่างใกล้ชิดกับผู้ใช้งาน และในการทำงานจะมีการจัดอันดับความสำคัญในแต่ละชิ้นงานอยู่ใน รูปแบบของ Product Backlog ซึ่งจะมี Product Owner เป็นผู้จัดอันดับความสำคัญของงาน



รูปที่ 2.1 องค์ประกอบของ Scrum Methodology

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 องค์ประกอบของ Scrum Methodology

Scrum Methodology ประกอบไปด้วยส่วนสำคัญต่าง ๆ ดังต่อไปนี้

1. User Stories เป็นรายละเอียดของแต่ละงานซึ่งจะบรรยายหัวข้อ Story เป็นประโยค สั้นๆ ความยาว 1-9 ประโยค อธิบายความความต้องการรวมถึงรายละเอียดของระบบซอฟต์แวร์ที่จะถูกพัฒนา

2. Sprint เป็นช่วงเวลาของการพัฒนาระบบโดยระหว่างการพัฒนาจะมีการกำหนด Check point ซึ่งเป็นการกำหนดให้การทำงานเป็นไปตามแต่ละ Check point โดยทั่วไปแล้วช่วงเวลาของ Sprint จะเป็นระยะเวลาตั้งแต่ 2 ถึง 4 สัปดาห์

3. Scrum Roles หน้าที่ของบุคคลในทีมพัฒนาซอฟต์แวร์

3.1 Scrum Team เป็นบุคคลที่ปฏิบัติงานจริงมีจำนวน 5-2 คนงานที่แต่ละบุคคลได้รับจะไม่มีการกำหนดอย่างตายตัว สมาชิกแต่ละคนจะสามารถทำงานทดแทน กันได้เสมอ สมาชิกภายในกลุ่มจะประกอบไปด้วยหลายตำแหน่งงาน เช่น นักพัฒนาระบบ และนักทดสอบระบบ สมาชิกทุกคนมีบทบาทและความสำคัญเท่ากัน สามารถ แสดงความคิดเห็นได้ทุกเรื่อง สมาชิกทุกคนมีหน้าที่ประเมินเวลาและงานของตนเพื่อให้ งานสำเร็จลุล่วงตามวัตถุประสงค์และเวลาที่ได้ถูกกำหนดเอาไว้

3.2 Product Owner เป็นบุคคลที่สนับสนุนให้งานสามารถสำเร็จลุล่วงได้โดย มีหน้าที่กำหนดขอบเขตของการทำงานที่จะทำการพัฒนา จัดการ Product backlog เพื่อให้ Scrum Team สามารถเห็นแนวทางในการปฏิบัติงาน Product Owner มี หน้าที่ผลักดันให้กลุ่มผู้พัฒนาซอฟต์แวร์สามารถสร้างผลงานออกมาตามลำดับ ความสำคัญของงาน

3.3 Scrum Master เป็นบุคคลที่คอยอำนวยความสะดวกให้กับสมาชิกภายใน กลุ่มผู้พัฒนาซอฟต์แวร์ เป็นผู้ที่รับผิดชอบคุณภาพของผลงานและเวลา รวมถึงการประสานงานระหว่างสมาชิกภายในทีม นอกจากนี้ Scrum master ยังมีหน้าที่ คอยแก้ไขปัญหาและอุปสรรคต่าง ๆ ที่อาจเกิดขึ้นระหว่างการทำงาน

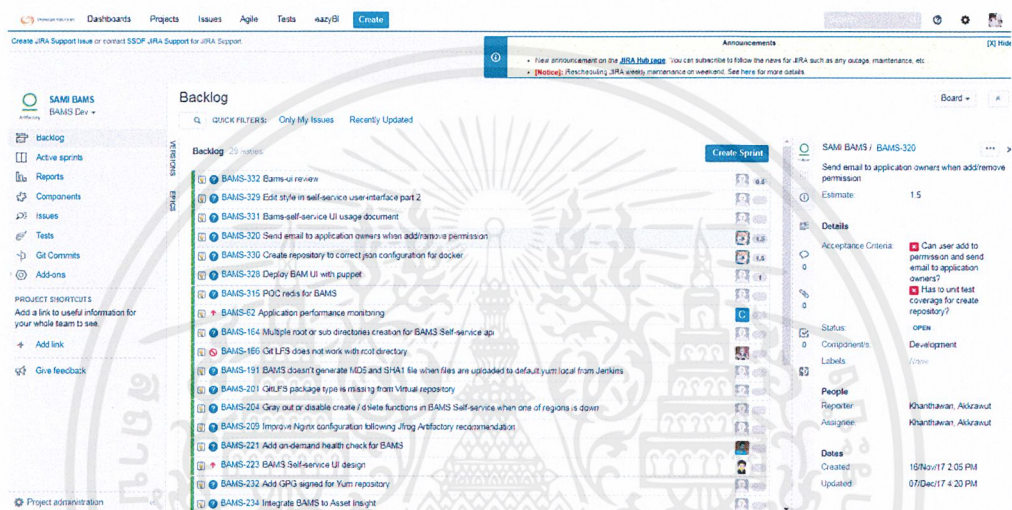
2.1.2 กระบวนการทำงานของ Scrum Methodology

กระบวนการทำงานของ Scrum Methodology มีลักษณะที่สำคัญดังนี้

1. พัฒนา Product Backlog เป็นรายการของงานที่จะต้องทำซึ่งได้มาจากการรวบรวม ความต้องการของผู้ใช้งาน ซึ่ง Product Owner จะเป็นคนจัดลำดับของงานตามความสำคัญ โดย เลือกมาจาก User Stories เมื่องานได้ถูกจัดลำดับแล้วจะถูกนำเข้าไป Sprint เพื่อจัดการกับงาน และรายละเอียดของงานนั้น ๆ ต่อไป

2. Sprint Meeting เป็นการประชุมระหว่างสมาชิกภายในทีมเพื่อให้การทำงานในแต่ละ Sprint นั้นมีประสิทธิภาพมากยิ่งขึ้นโดยในการประชุมจะถูกแบ่งออกเป็น 4 ประเภทคือ

2.1 Sprint Planning Meeting เป็นการประชุมก่อนเริ่มทำงานใน Sprint เพื่อพูดคุยเกี่ยวกับรายละเอียดของงานใน Product Backlog จากนั้นจะมีการวางแผน บุคคลที่จะพัฒนางาน และจะมีการกำหนดระยะเวลาในการพัฒนาของงานแต่ละชิ้น เป็น Sprint Phase หากชิ้นงานมีขนาดใหญ่จะมีการแยกงานออกเป็น Task ย่อย ๆ และ กำหนดว่าจะต้องใช้เวลาเท่าไรในการทำงานชิ้นนั้น รวมถึงมีการกำหนดว่าในแต่ละ Sprint จะต้องมีส่วนงานย่อยส่วนไหนเสร็จบ้าง



รูปที่ 2.2 ส่วนหน้าจอในการจัดการ Product Backlog (Jira Dashboard)

2.2 Daily Scrum เป็นการประชุมอย่างไม่เป็นทางการ โดยจะต้องมีการพูดคุย กันทุกวัน สมาชิกทุกคนจะต้องมีส่วนร่วมใน Daily scrum เพื่อพูดคุยถึงความคืบหน้า ของชิ้นงานที่ตนเองกำลังรับผิดชอบอยู่ และปัญหาที่อาจจะเกิดขึ้นระหว่างทำงาน การประชุม ดังกล่าวจะช่วยให้งานสามารถเสร็จลุส่งไปได้ Daily Scrum จะประกอบไปด้วย 3 คำถามหลัก ๆ คือ

- เมื่อวานได้ทำอะไรไปแล้วบ้าง
- วันนี้จะต้องทำงานอะไร
- มีปัญหาอะไรระหว่างการทำงานหรือเปล่า งานจะสามารถสำเร็จได้ตามที่วางแผนไว้หรือไม่

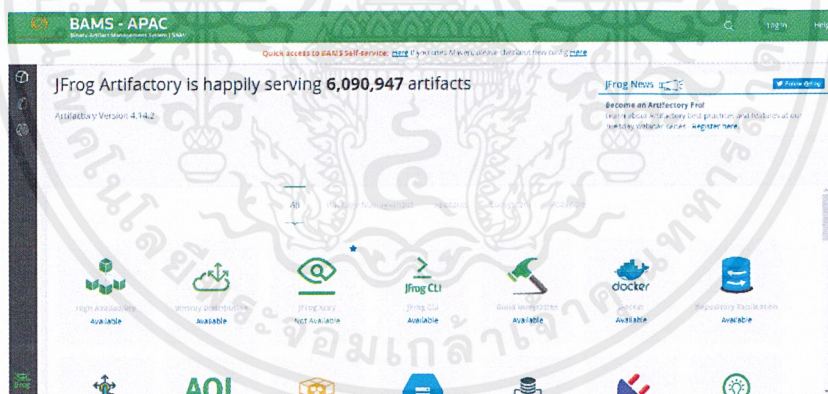
2.3 Sprint Review Meeting เป็นการประชุมเมื่อต้องส่งมอบงานให้กับลูกค้า โดย ผู้พัฒนาจะต้องแสดงชิ้นงานให้กับลูกค้าดูว่างานนั้น ๆ ตรงกับความตรงการของลูกค้า และเป็นไปตามที่ได้พูดคุยกันไว้หรือไม่

2.4 Retrospective Meeting เมื่อจบSprint จะมีการพูดคุยกันระหว่างในทีมอีกครั้ง หลังจากที่เราส่งมอบงานให้กับลูกค้า โดยจะมีการพูดคุยเกี่ยวกับระหว่างการทำงาน มีการทำงานส่วนไหนที่ควรแก้ไข ซึ่งสามารถเป็นเรื่องทั่วไปที่เกี่ยวข้องกับการทำงาน สมาชิกก็จะช่วยกันออกความคิดเห็นเพื่อหาทางออกให้กับปัญหานั้น ๆ และร่วมกัน แก้ปัญหา เพื่อนำมาเพิ่มประสิทธิภาพการทำงานใน Sprint ถัดไป

2.2 ระบบการจัดการอาร์ติแฟกต์แบบไบนารี

ระบบการจัดการอาร์ติแฟกต์แบบไบนารี (Binary Artifact Management System: BAMS) เป็นซอฟต์แวร์ที่ทำหน้าที่เป็นศูนย์รวมจัดเก็บไฟล์ประเภทอาร์ติแฟกต์และแพ็คเกจ ที่มีนามสกุลและมีวิธีการนำไปใช้งานที่แตกต่างกัน ซึ่งระบบนี้เป็นขั้นพื้นฐานในการกระบวนการพัฒนาซอฟต์แวร์ สำหรับในองค์กรที่มีกระบวนการทำงานที่แต่ละทีมมีการทำงานที่แตกต่างกัน หรือระหว่างการทำงาน ต้องมีแลกเปลี่ยนไฟล์ประเภทต่าง ๆ ระหว่างทีมภายในองค์กร

ซึ่งภายในระบบจะมี ที่จัดเก็บข้อมูล (Repository) ซึ่งทำหน้าที่จัดเก็บไฟล์ประเภทอาร์ติแฟกต์และแพ็คเกจที่มีความหลากหลายและซับซ้อนให้มีการแยกสัดส่วนสำหรับการเรียกใช้งานอย่างเป็นระบบ โดยมีคุณสมบัติที่กำหนดให้ผู้ใช้สามารถจัดการได้ ได้แก่ การตั้งชื่อ การเลือกประเภทของอาร์ติแฟกต์และแพ็คเกจเฉพาะสำหรับที่จัดเก็บนั้น ๆ การกำหนดสิทธิ์ในการเข้าถึงที่จัดเก็บนั้น ๆ



รูปที่ 2.3 ส่วนหน้าจอผู้ใช้งานของ Binary Artifact Management System

ซึ่งก่อนที่ผู้ใช้งานจะเข้าไปใช้งานระบบนั้น ผู้ใช้งานควรที่จะทราบเกี่ยวกับข้อมูลพื้นฐานในการใช้ก่อน โดยมีการแบ่งเป็นหัวข้อดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 การระบุตัวตนในการเข้าใช้งานระบบ

เนื่องจากระบบเป็นส่วนกลางของภายในองค์กร จึงมีผู้ใช้งานภายในองค์กรเป็นจำนวนมาก ระบบจึงต้องมีส่วนสำหรับภาระระบุตัวตนของผู้ใช้ เพื่อความปลอดภัยของข้อมูล และความเป็นระเบียบเรียบร้อยในการจัดการไฟล์ต่าง ๆ ภายในระบบ ซึ่งระบบ BAMS นั้นใช้ การเข้าใช้งานแบบ Single Sign-On เป็นรูปแบบของระบบการยืนยันตัวตนบุคคล (Authentication) ที่รองรับการให้ผู้ใช้งานลงชื่อเข้าใช้งานระบบ (Login) ครั้งเดียว แล้วสามารถเข้าใช้งานระบบหลายระบบได้ โดยไม่ต้องลงชื่อเข้าใช้งานซ้ำอีก

การเข้าใช้งานระบบต้องมีการเข้าใช้งานโดยใช้ ระบบยืนยันตัวตนบุคคลของผู้ใช้ ที่จัดทำขึ้นมาและใช้ภายในองค์กร โดยใช้ LDAP หรือว่า Lightweight Directory Access Protocol ซึ่งเป็น โพรโตคอล (Protocol) ที่ใช้สำหรับช่วยในการเข้าถึงและค้นหาข้อมูล ซึ่ง LDAP เป็น “Lightweight” นั้นหมายถึง มีขนาดเล็กลง ส่วนใหญ่นำเอามาใช้กับข้อมูลรายละเอียดพนักงาน เช่น ชื่อ นามสกุล ตำแหน่ง ที่อยู่

เนื่องจาก LDAP นั้นมีคุณสมบัติเป็น Service Account ซึ่งหมายความว่าบัญชีนั้นจะสามารถใช้บริการต่าง ๆ ภายในองค์กรที่ระบบอนุญาต ให้สามารถใช้บริการได้โดยอัตโนมัติ

ผู้ใช้งานจะเข้าสู่ระบบของ BAMS โดยการใส่ชื่อกับนามสกุลและรหัสผ่าน ในระบบฐานข้อมูลของพนักงานภายในองค์กรเพื่อทำการเข้าใช้งานระบบ

2.2.2 การกำหนดชื่อของที่จัดเก็บอาร์ติแฟคแบบไบนารี

ระบบการจัดการอาร์ติแฟคแบบไบนารีนั้นมี ที่จัดเก็บข้อมูล (Repository) ที่มีความแตกต่างกันหลายรูปแบบ อีกทั้งระบบมีผู้ใช้เข้ามาใช้งานจำนวนมากจึงต้องมีการกำหนดมาตรฐานในการตั้งชื่อ เพื่อความเข้าใจและความสะดวกในการค้นหาที่จัดเก็บข้อมูล เพื่อเรียกใช้อาร์ติแฟคและแพ็คเกจที่ผู้ใช้ต้องการได้

ผู้พัฒนาระบบมีการกำหนด มาตรฐานของการกำหนดชื่อของที่จัดเก็บข้อมูล กันว่าจะกำหนดให้ผู้ใช้ใช้ชื่อตามรูปแบบดังนี้

“{Application Short Name}::{Repository Type}::{Repository Class}”

ตัวอย่างการตั้งชื่อเช่น “bams.generic.local” โดยที่ชื่อแต่ละส่วนประกอบจะมีการค้นระหว่างกันโดย” ซึ่งข้อมูลที่น่ามาใช้ประกอบไปด้วย 3 ส่วนดังนี้

1. Application Short name นั้นเป็นชื่อโดยย่อของซอฟต์แวร์แต่ละตัวที่ถูกพัฒนาขึ้นภายในองค์กร มีเพื่อระบุว่าอาร์ติแฟคและแพ็คเกจที่อยู่ในที่จัดเก็บข้อมูล นำไปใช้กับการพัฒนาหรือมีความเกี่ยวข้องกับซอฟต์แวร์นั้น ๆ ซึ่งข้อมูลของ Application Short name จะถูกเรียกนำมาใช้โดยจากการใช้บริการที่ชื่อว่า CUDL (Compass Unified Data Layer) ซึ่งเป็นบริการส่วนกลางขององค์กร ในการเข้าถึงข้อมูลที่เกี่ยวข้องกับองค์กร

2. Repository Type นั้นคือประเภทของที่จัดเก็บข้อมูล โดยจะเอามาจากชื่อประเภทของอาร์ติแฟกต์และแพ็คเกจ ที่สามารถจัดการได้และมีผู้ใช้ ใช้เป็นส่วนใหญ่ใน BAMS มาใช้ในการกำหนดชื่อของที่จัดเก็บข้อมูล เป็นหลัก

3. Repository Class นั้นเป็นการแบ่งประเภทของที่จัดเก็บตามลักษณะการเข้าถึงของผู้ใช้งาน ซึ่งแบ่งเป็น 3 ประเภทย่อยดังนี้

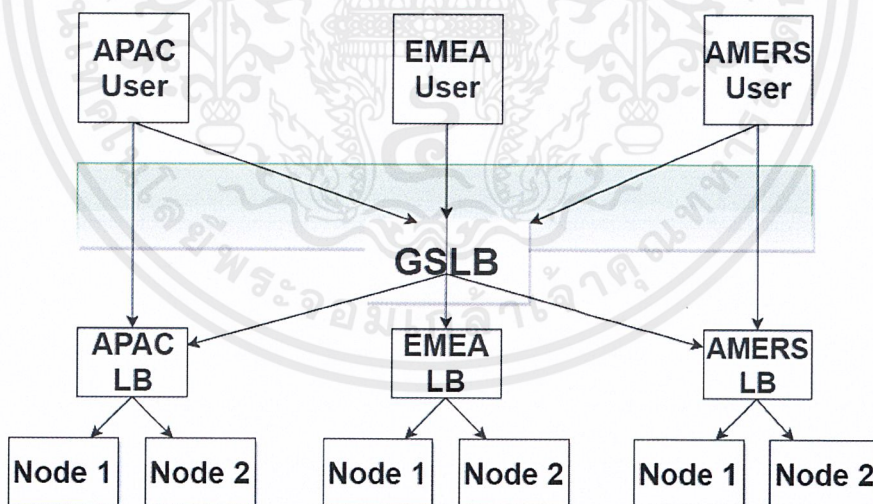
3.1 Local repositories เป็นประเภทของที่จัดเก็บข้อมูล ที่จะสามารถให้ผู้ใช้สามารถจัดการเอาแพ็คเกจไฟล์ไปจัดเก็บและเรียกใช้ได้ด้วยตัวผู้ใช้งานเอง

3.2 Virtual repositories เป็นประเภทของที่จัดเก็บข้อมูล ที่รวบรวมหลายๆ ของที่จัดเก็บข้อมูล ไว้ใน URL เดียวกัน ซึ่ง Virtual repositories สามารถเรียกใช้งานได้เพียงอย่างเดียว ซึ่งผู้ใช้ไม่สามารถจัดการเอาแพ็คเกจไฟล์ไปจัดเก็บได้

3.3 Remote repositories เป็นประเภทของที่จัดเก็บข้อมูล ที่มีการติดต่อออกไป URL ข้างนอกของระบบการจัดการอาร์ติแฟกต์แบบไบนารี

2.2.3 โครงสร้างภายในของระบบ

BAMS นั้นเป็นระบบกลางขององค์กร ที่มีการใช้งานทั้งองค์กร ซึ่งโดยโครงสร้างขององค์กรนั้นมีการเปิดสาขากระจายอยู่ในหลายประเทศ ทำให้ระบบนั้นมีการเข้าถึงมาจากหลายภูมิภาค ระบบจึงต้องมีการกระจายโครงสร้างของเซิร์ฟเวอร์เพื่อการเข้าถึงที่สะดวกและรวดเร็วสำหรับการทำงาน

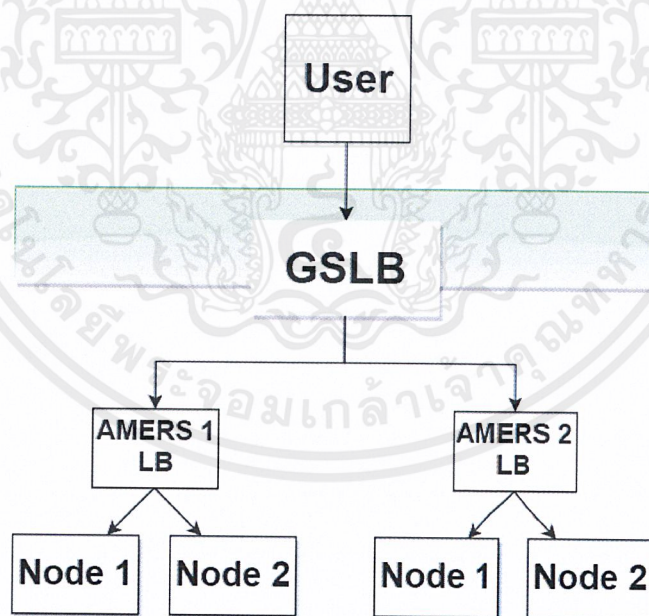


รูปที่ 2.4 โครงสร้างของการวางเครื่องเซิร์ฟเวอร์ที่กระจายไปในแต่ละภูมิภาค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 2.4 นั้นอธิบายถึงโครงสร้างของการเชื่อมต่อกันของเครื่องเซิร์ฟเวอร์ที่ใช้ในการจัดเก็บอาร์ติแฟกต์และแพ็คเกจ (JFrog Artifactory) ซึ่งมีการกระจายไปในแต่ละภูมิภาคของผู้ใช้งานภายในองค์กร จากโครงสร้างนั้นมีการแบ่งผู้ใช้เป็นสามภูมิภาคด้วยกันก็คือ เอเชียแปซิฟิก (APAC), ยุโรป (EMEA), อเมริกา (AMERS) ซึ่ง URL สำหรับให้ผู้ใช้ทำการใช้ในการเลือกในการเข้าถึงบริการของ BAMS นั้นก็จะแตกต่างกันไปตามภูมิภาค เช่น bams-emea-sami.int.thomsonreuters.com เป็น URL ของเซิร์ฟเวอร์ที่อยู่ประเทศโซนยุโรปโดยทั้งสามภูมิภาคนั้นมี GSLB (Global load balancing) เป็นส่วนที่จะเข้ามาช่วยจัดการการกระจายของข้อมูล (Load balancing) ไปเก็บยังเซิร์ฟเวอร์ของภูมิภาคต่าง ๆ ซึ่งในส่วนของแต่ละภูมิภาคก็จะมีการทำ Load balancing อีกครั้งไปยังเครื่องเซิร์ฟเวอร์สอง เครื่อง (Node) เพื่อความเสถียรในขั้นตอนของการบำรุงรักษาและการปรับปรุงระบบ BAMS

แต่เนื่องจากทางองค์กรต้องการให้มีการปรับปรุงโครงสร้างของเครื่องเซิร์ฟเวอร์ที่ใช้ในการจัดเก็บอาร์ติแฟกต์และแพ็คเกจให้อยู่ที่ AMER เท่านั้น รวมถึงการปรับปรุงในส่วนของ URL สำหรับให้ผู้ใช้ในเข้าถึงส่วนของการจัดการการจัดเก็บอาร์ติแฟกต์และแพ็คเกจ ให้เหลือแค่เพียง URL เดียวเพื่อความเสถียรในการใช้งานของผู้ใช้ ทำให้จึงต้องมีการปรับปรุงโครงสร้างของเซิร์ฟเวอร์ให้เหมาะสมกับลักษณะการใช้งาน



รูปที่ 2.5 โครงสร้างของการวางเครื่องเซิร์ฟเวอร์ที่ถูกปรับปรุงจากของเดิม

จากรูปที่ 2.5 อธิบายถึงโครงสร้างของการเชื่อมต่อกันของเครื่องเซิร์ฟเวอร์ที่ใช้ในการจัดเก็บอาร์ติแฟกต์และแพ็คเกจที่ถูกปรับปรุงจากของเดิม ซึ่งมีหลักการคล้ายกับของเดิมแต่มีการลดทอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนเครื่องเซิร์ฟเวอร์ที่ทำการกระจายการเก็บข้อมูลให้น้อยลง จากเดิมที่มีอยู่สามภูมิภาค ให้เหลือแค่เพียงเซิร์ฟเวอร์ที่อยู่ประเทศโซนอเมริกาและการเข้าใช้ BAMS ผ่าน URL ของผู้ใช้มีแค่เพียง URL เดียว

2.3 ส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟกแบบไบนารี (BAMS-Self service)

ส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟกแบบไบนารี (BAMS-self-service) คือระบบเว็บแอปพลิเคชันกลางที่ใช้ภายในองค์กรที่เข้ามาทำหน้าที่จัดการกับระบบจัดการอาร์ติแฟกแบบไบนารี (Binary Artifact Management System: BAMS) โดยทำการติดต่อโดยใช้ ส่วนติดต่อเซอร์วิสของแอปพลิเคชัน ทำการติดต่อไปยังระบบเพื่อเข้าไปจัดการภายในระบบ BAMS โดยผู้ใช้งานสามารถใช้งานผ่านเว็บแอปพลิเคชันที่มีส่วนต่อประสานกับผู้ใช้ (User interface) ซึ่งมีส่วนหลักที่คอยทำหน้าที่ในการจัดการดังนี้ ที่จัดเก็บข้อมูล (Repository) เพิ่มข้อมูล (Directory) การกำหนดสิทธิในการเข้าไปจัดการ (Permission)

BAMS Self-Service API
BAMS directory and permission management

1. Root directory creation

PUT /rootDirectories/{pathToRootDirectory}

Implementation Notes
Create your root working directory under target repository.
I.e. you want to create root directory name "yourteam" under repository "default.generic.local"
enter following value
Path to directory that you want to create with {targetRepository}/{rootDirectory} e.g. default.generic.local/AAA For maven repository type: Path to directory is {targetRepository}/com/trireuters/thomsonreuters/txall/trgr/westgroup/{rootDirectory} e.g. release.maven.local.com/tri/AAA

Parameter	Value	Description
pathToRootDirectory	(required)	Path to root directory that you want to create e.g. {targetRepository}/{rootDirectory}/ For maven repository type: {targetRepository}/com/trireuters/thomsonreuters/txall/trgr/westgroup/{rootD

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	successful operation		

[Try it out!](#)

รูปที่ 2.6 ส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟกแบบไบนารี

2.4 การพัฒนา ส่วนติดต่อเซอร์วิสของแอปพลิเคชัน (Application Programming Interface)

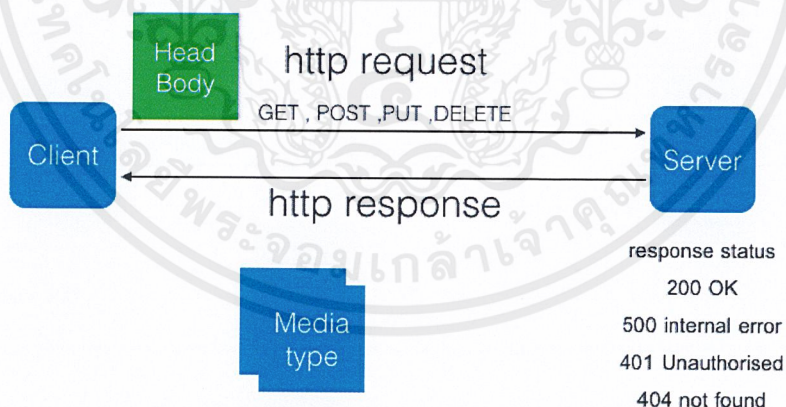
ก่อนจะเริ่มต้นในการพัฒนาส่วนติดต่อเซอร์วิสของแอปพลิเคชัน (Application Programming Interface: API) นั้นผู้พัฒนาต้องมีความรู้ความเข้าใจในการพัฒนา ก่อน ซึ่งจะมีการแบ่งหัวข้อที่จะต้องไปทำการศึกษาดังนี้

2.4.1 ส่วนติดต่อเซิร์ฟวิสของแอปพลิเคชัน (Application Programming Interface)

ส่วนติดต่อเซิร์ฟวิสของแอปพลิเคชัน (Application Programming Interface : API) หมายถึงการพัฒนาโปรแกรมสำหรับเป็นส่วนติดต่อเซิร์ฟวิสของแอปพลิเคชัน หรือโมดูลต่าง ๆ เพื่อให้คนภายนอกมาเรียกใช้งานหรือกล่าวอย่างง่าย ๆ เป็นการเขียนโปรแกรมเพื่อให้บริการสำหรับให้คนอื่นมาเรียกใช้งาน ซึ่งแนวคิดเรื่องการสร้าง API เพื่อการใช้งานก็มีมาอย่างยาวนานแล้ว ยกตัวอย่างเช่นตั้งแต่ที่มีการสร้างระบบปฏิบัติการ (OS) ก็จะมีการติดต่อ API ของ driver อุปกรณ์ฮาร์ดแวร์ต่าง ๆ ของเครื่องคอมพิวเตอร์เพื่อเรียกใช้งานอุปกรณ์นั้น ๆ ตามคำสั่งของนักพัฒนา

2.4.2 Restful API

Representational state transfer หรือ REST คือ มาตรฐานหนึ่งในการเขียนเว็บเซิร์ฟวิส หรือ เว็บ Api (Application Programming Interface) เพื่อให้เทคโนโลยีและแพลตฟอร์มต่าง ๆ สามารถแลกเปลี่ยนข้อมูลกันได้ ไม่ว่าจะเป็น OS, Web, Mobile Application ในการทำงานจะใช้หลักการแบบ stateless คือไม่มี web session การทำงานของ Restful เว็บเซิร์ฟวิส จะอาศัย URIs (Uniform Resource Identifiers) และ URL (Uniform Resource Locator) เพื่อร้องขอ (Request) เพื่อค้นหาและประมวลผลแล้วตอบสนอง (Response) กลับไปในรูปแบบของไฟล์ประเภท XML, HTML, JSON โดยสิ่งที่ตอบกลับมาจะเป็นการยืนยันผลของคำสั่งที่ส่งเข้าไป ซึ่ง Restful API สามารถพัฒนาด้วยภาษาที่หลากหลาย



รูปที่ 2.7 Flow การร้องขอและตอบสนองของ Restful API

ซึ่งในการติดต่อเพื่อร้องขอในการทำงานของ Restful API จะใช้ HTTP methods เป็นคำสั่งพื้นฐานในโครงสร้างของระบบ ซึ่งมีคำสั่งดังนี้

- GET: เป็นคำสั่งสำหรับใช้ในการทำการดึงข้อมูลจาก URL ที่กำหนด
- PUT: เป็นคำสั่งสำหรับใช้ในการเพิ่มข้อมูลใหม่

- POST: เป็นคำสั่งสำหรับใช้ในการแก้ไขข้อมูลหรือเพิ่มข้อมูลใหม่
- DELETE: เป็นคำสั่งสำหรับใช้ในการลบข้อมูล

2.4.3 การออกแบบ Restful API

API เป็นช่องทาง เพื่อให้ส่วนอื่นของระบบเรียกใช้งาน จึงเปรียบเสมือนกับประตู ดังนั้นการออกแบบ Restful API ที่ดีนั้นต้องคำนึงถึงในหลาย ๆ ด้าน ไม่ว่าจะเป็นในด้านของมุมมองของผู้ใช้ เมื่อผู้ใช้เข้ามาเรียกใช้งานนั้นผู้ใช้จะต้องสามารถคาดเดาได้ ว่าเมื่อเรียกใช้งานแล้วจะเข้าไปทำงานอะไร แล้วได้ผลลัพธ์ออกมาอย่างไร อีกทั้งต้องง่ายต่อการปรับปรุงหรือแก้ไข ระบบของผู้พัฒนาระบบอีกด้วย

สิ่งสำคัญที่ควรคำนึงถึงในการออกแบบ Restful API ก็คือ ผู้พัฒนาต้องคำนึงถึง ชื่อของเส้นทางของ URL ที่จะทำการเข้าถึงสำหรับใช้งานระบบในส่วนนั้น ตัวอย่างเช่น URL นั้นมีเส้นทางชื่อ localhost:3000/GetAllEmployee คือเส้นทางเพื่อที่จะเข้าไปดึงข้อมูลรายละเอียดรายชื่อของพนักงานออกมา แต่ถ้าหากมีฟังก์ชันอื่น ๆ เพิ่มเติมอีก เช่น การลบ การเพิ่มชื่อพนักงานเพิ่มเติมทำให้ผู้พัฒนาระบบต้องมีการ ตั้งชื่อเส้นทางเพิ่มเติมอีกเพื่อกำหนดการกระทำลงไปเป็นชื่อของเส้นทางด้วย ซึ่งผู้พัฒนาสามารถตั้งชื่ออย่างนี้ได้ไม่ผิด แต่จะผู้ใช้จะต้องจำชื่อของเส้นทางเยอะ เนื่องจาก Restful API มี HTTP methods (GET, POST, DELETE, PUT) สำหรับกำหนดการกระทำอยู่แล้วทำให้จึงไม่ต้องตั้งชื่อให้ยาว ซึ่งทำให้ผู้พัฒนานั้นสามารถย่อ URL จากของเดิมให้เหลือแค่ localhost:3000/employee ได้

การส่ง HTTP response status codes ควรส่งกลับมาให้ผู้ใช้ทราบอย่างชัดเจน เข้าใจและทราบว่า การทำงานนั้นที่ส่งไปนั้นสำเร็จ ถ้าเกิดข้อผิดพลาดก็ต้องแจ้งให้ผู้ใช้เข้าใจว่าเกิดข้อผิดพลาดนั้นเกิดเพราะอะไร ไม่ว่าจะเป็นที่ตัวผู้ใช้เองกรอกข้อมูลไม่ตรงกับที่ระบบต้องการ หรือเกิดจากข้อผิดพลาดจากเซิร์ฟเวอร์ก็ตาม ระบบก็ต้องสามารถแจ้งให้ผู้ใช้ทราบ ซึ่งรูปแบบของข้อความตอบกลับจากเซิร์ฟเวอร์นั้นควรที่จะมี Status Code ที่ตอบกลับมายู่ในรูปแบบดังต่อไปนี้

1. Status Code 2xx (ประเภทเข้าถึงได้สำเร็จ)

สถานะที่จะส่งกลับมาจากเซิร์ฟเวอร์นั้น จะแสดงถึงการร้องขอจากผู้ใช้ที่ได้รับผลการประมวลผลที่เซิร์ฟเวอร์นั้นได้สำเร็จ

- 200 Ok: เป็นมาตรฐานของการตอบกลับแบบ HTTP รูปแบบทั่วไปแต่ใช้ในการตอบกลับเมื่อการประมวลผลที่เซิร์ฟเวอร์ในเสร็จสมบูรณ์

- 201 Created: สถานะนี้นั้นจะมีการคืนค่าเมื่อมีการเพิ่มข้อมูลเข้าไปโดยการใช้ POST method และข้อมูลนั้นมีการเพิ่มเข้าไปได้สำเร็จ

- 204 No Content: สถานะนี้นั้นจะคืนค่าเมื่อมีการเพิ่มข้อมูลเข้าไปโดยการใช้ DELETE method เพื่อทำการลบข้อมูล แล้วส่งกลับมายืนยันกับผู้ใช้ว่าข้อมูลนั้นได้หายไปแล้ว

2. Status Code 3xx (ประเภทมีการเปลี่ยนแปลงเส้นทาง)
 - 304 Not Modified: เป็นสถานะที่บ่งบอกว่าผู้ใช้มีการตอบสนองจากข้อมูลนั้นอยู่แล้วในแคช ระบบจึงไม่ต้องทำการตอบสนองข้อมูลนั้นอีก
3. Status Code 4xx (ประเภทเกิดข้อผิดพลาดจากฝั่งงาน)

เป็นสถานะที่แสดงให้ผู้ใช้ได้ระบบรู้ว่าระบบเกิดข้อผิดพลาดมาจากผู้ใช้ เนื่องจากผู้ใช้ได้ส่งคำขอที่ไม่ถูกต้อง

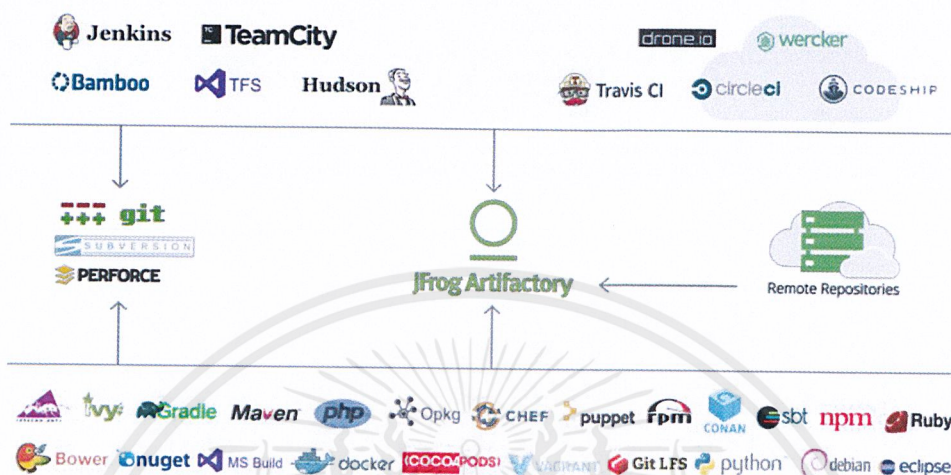
 - 400 Bad Request: เป็นสถานะที่บ่งบอกว่าคำขอของผู้ใช้นั้นไม่ได้รับการประมวลผลที่เซิร์ฟเวอร์ เนื่องจากเซิร์ฟเวอร์ไม่อาจเข้าใจในสิ่งที่ผู้ใช้ต้องการได้
 - 401 Unauthorized: เป็นสถานะที่บ่งบอกว่าผู้ใช้นั้นไม่ถูกอนุญาตให้เข้าถึงข้อมูลได้ และควรส่งคำขอพร้อมใส่ข้อมูลที่จำเป็นสำหรับการใช้ระบุตัวตนในการเข้าใช้งานระบบ
 - 403 Forbidden: เป็นสถานะที่บ่งบอกว่าผู้ใช้นั้นได้ระบุคำขอเข้าใช้งานได้ถูกต้องแล้ว และได้มีการระบุตัวตนในการยืนยันเพื่อเข้าใช้งานแล้ว แต่ผู้ใช้ไม่ได้รับอนุญาตให้เข้าถึงข้อมูลในหน้าของเว็บเพจนั้นด้วยเหตุผลบางประการ เช่น บางครั้งผู้ใช้นั้นไม่ได้รับสิทธิ์ให้เข้าถึงข้อมูลในไดเรกทอรีนั้นบนเซิร์ฟเวอร์
 - 404 Not Found: เป็นสถานะที่บ่งบอกว่าข้อมูลที่ทำการร้องขอนั้นไม่สามารถใช้งานได้ในขณะนี้
 - 410 Gone: เป็นสถานะที่บ่งบอกว่าข้อมูลที่ทำการร้องขอนั้นไม่สามารถเข้าถึงได้อีกต่อไปเพราะถูกย้ายจากผู้ดูแลระบบโดยเจตนา
4. Status Code 5xx (ประเภทเกิดข้อผิดพลาดจากเซิร์ฟเวอร์)
 - 500 Internal Server Error: เป็นสถานะที่บ่งบอกว่าผู้ใช้นั้นได้ระบุคำขอเข้าใช้งานได้ถูกต้องแต่เนื่องจากเซิร์ฟเวอร์มีข้อผิดพลาด เลยทำการร้องขอเงื่อนไขบางอย่างแก่ผู้ใช้
 - 503 Service Unavailable: เป็นสถานะที่บ่งบอกว่าระบบเซิร์ฟเวอร์นั้นไม่สามารถเข้าใช้งานได้หรือไม่สามารถรับคำขอเข้าใช้งานได้ ในกรณีส่วนใหญ่จะเป็นเพราะว่าเซิร์ฟเวอร์นั้นอยู่ในระหว่างการบำรุงรักษาอยู่

2.5 ภาษาและเครื่องมือที่นำมาใช้ในการพัฒนาระบบ

2.5.1 JFrog Artifactory

JFrog Artifactory เป็น ระบบการจัดการอาร์ติแฟกแบบไบนารีที่มียี่ห้อหนึ่งที่มีชื่อเสียง ซึ่งการที่อาร์ติแฟกหรือแพ็คเกจ ต่างก็มีเวอร์ชันต่าง ๆ ที่เพิ่มขึ้นเรื่อย ๆ ทำให้ต้องมีการจัดการที่เป็นระบบมากขึ้น ดังนั้นเพื่อสร้างมาตรฐานในองค์กรที่มีการพัฒนาซอฟต์แวร์ JFrog Artifactory จึงเป็น

เครื่องมือที่เข้ามาช่วยในการจัดการไฟล์แพ็คเกจ ช่วยในการเพิ่มความปลอดภัยในการจัดเก็บ และมีระบบการควบคุมผู้ใช้งานในการเข้าถึงไฟล์อาร์ติแฟคหรือแพ็คเกจอีกด้วย



รูปที่ 2.8 ความสัมพันธ์ของเครื่องมือ อาร์ติแฟค หรือ แพ็คเกจต่าง ๆ ที่ JFrog Artifactory รองรับในการใช้งาน

2.5.2 Node.js



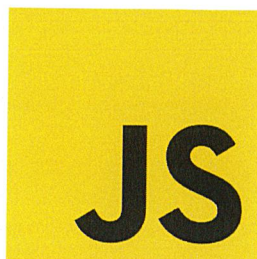
รูปที่ 2.9 Node.js

Node.js เป็น Platform ที่ทำงานอยู่บนฝั่ง Server ถูกพัฒนาโดย Ryan Dahl ในปี 2009 เป็น Open Source Node.js นั้นเป็น JavaScript runtime สร้างบน Chrome's V8 JavaScript engine โดย Node.js นั้นจะใช้ event-driven, non-blocking I/O model ทำให้ไม่กินทรัพยากรเครื่องและทำให้มีประสิทธิภาพสูงในการทำงาน

Node.js นั้นมีส่วนเสริม (module/library/plugin/package) หลากหลาย ในคำสั่ง npm (node.js Package Manager หรือ JavaScript Package Manager) ซึ่งประโยชน์ของมันก็คือเราสามารถติดตั้งแพ็คเกจต่าง ๆ ที่พิเศษ ได้ในคำสั่งเดียว เช่น ถ้าผู้ใช้ต้องการเขียน node.js ให้ทำงานกับแพ็คเกจตัวใดตัวหนึ่งก็ไม่ต้องเขียนโค้ดยาวจึงจะสามารถทำงานได้ ผู้ใช้สามารถสั่งติดตั้งแพ็คเกจนั้นโดยใช้เพียงคำสั่ง npm install ตามด้วยชื่อแพ็คเกจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 JavaScript (ES6)

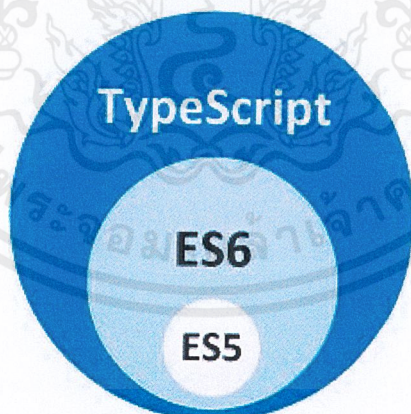


รูปที่ 2.10 JavaScript

จาวาสคริปต์ (JavaScript) เป็นภาษาสคริปต์ที่มีลักษณะการเขียนแบบ Prototyped-based Programming ส่วนมากใช้ในหน้าเว็บเพื่อประมวลผลข้อมูลที่ฝั่งของผู้ใช้งาน พัฒนาโดย Brendan Eich ภาษา JavaScript ไม่มีความสัมพันธ์กับภาษา Java แต่มีโครงสร้างภาษาที่คล้ายกันมากเนื่องจากได้รับ การพัฒนาต่อมาจากภาษาซีทั้งคู่

เพื่อกำหนดและดูแลมาตรฐานของภาษา เนื่องจาก JavaScript เริ่มเป็นที่นิยมและรองรับในเว็บเบราว์เซอร์ต่าง ๆ ทำให้มีการกำหนดมาตรฐานของภาษาคือ ECMAScript (ES) ปัจจุบันก็มีการพัฒนาภาษานี้ไปอย่างต่อเนื่อง และมีชื่อเรียกตาม มาตรฐาน ECMAScript เช่น มาตรฐาน ECMAScript ที่เป็นที่ประกาศใช้ ในเดือนมิถุนายน 2015 โดยเรียกย่อได้ว่า ES2015 หรือ ES6 (ECMAScript Version 6)

โดยในการทำงาน จะใช้ความสามารถของ TypeScript เข้ามาช่วยในการพัฒนาทั้งในส่วนของการติดต่อเซิร์ฟเวอร์ของแอปพลิเคชัน



รูปที่ 2.11 ความสัมพันธ์ของ TypeScript และ JavaScript

TypeScript เป็นภาษาที่มารอบ JavaScript ซึ่งถูกพัฒนาโดย Microsoft หมายความว่า TypeScript สามารถทำงานร่วมกับ JavaScript ได้ โดยที่จะรวมความสามารถที่ ES6 นั้นมีอยู่ สิ่งที่เพิ่มขึ้นมาคือสนับสนุน Type System รวมถึงคุณสมบัติอื่น ๆ ที่เพิ่มมากขึ้น ความสามารถที่เพิ่มขึ้นของการเขียนโปรแกรมเชิงวัตถุ ซึ่งต้องใช้ตัวแปลภาษาของ TypeScript

(Transpiler) โดยจะแปล source code ที่ผู้พัฒนาเขียนให้เป็นภาษา JavaScript อีกทีหนึ่ง จึงจะทำให้มั่นใจได้ว่าผลลัพธ์สุดท้ายนั้นจะสามารถใช้งานได้บนเว็บเบราว์เซอร์ทั่วไป ซึ่งข้อดีของการใช้ TypeScript นั้นมีดังต่อไปนี้

- TypeScript นั้นได้รวมไว้ความสามารถของ ES2015 และ ความสามารถอื่น ๆ ของJavaScript สมัยใหม่ที่ใช้ได้ในปัจจุบันไว้แล้วใน TypeScript
- ตัวแปรที่ทำการประกาศไว้แล้วใน TypeScript จะเปลี่ยนชนิดของข้อมูลไม่ได้อีกต่อไป ซึ่งทำให้ข้อผิดพลาดในการเขียนโปรแกรมนั้นน้อยลง
- TypeScript มีการตรวจสอบ Source code ในช่วง compile time ทำให้ผู้พัฒนานั้นสามารถดักจับข้อผิดพลาดได้แต่ต้น โดยไม่ปล่อยให้ข้อผิดพลาดนั้นไปปรากฏให้เห็นในตอนทำงานจริง (Runtime)
- IDE และ Text Editor ส่วนใหญ่สนับสนุนให้ผู้พัฒนาสามารถใช้งาน TypeScript ได้อย่างสมบูรณ์

2.5.4 Postman



รูปที่ 2.12 Postman

Postman คือเครื่องมือสำหรับช่วยในการพัฒนา API ทดสอบการทำงานของ Service รวมถึงการ Mock Service อีกด้วย ซึ่งช่วยเราทำเรื่องยาก ๆ ให้กลายเป็นเรื่องง่าย โดยความนิยมของ Postman หลัก ๆ มาจาก UI ที่สวยงามใช้งานง่ายกว่า Tools อื่น ๆ โดยผู้ใช้งานไม่จำเป็นต้องมีความรู้เรื่องภาษาโปรแกรมมิ่งก็สามารถใช้งานได้สบาย

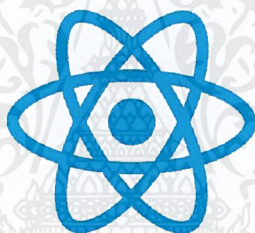
2.5.5 Swagger



รูปที่ 2.13 Swagger

Swagger เป็นเครื่องมือที่เข้ามาช่วยในเรื่องของการกำหนดมาตรฐาน ของรูปแบบของการออกแบบและมีส่วนของภาษาในการสร้าง User interface ของ Restful API ซึ่งจะเข้ามาเป็นตัวช่วย ให้มนุษย์และคอมพิวเตอร์เข้าใจส่วนที่ให้บริการของระบบมากขึ้น โดยไม่ต้องอ่าน Source code หรือว่าเอกสารคู่มือการใช้งานมาก่อน ก็สามารถใช้งานได้อย่างเข้าใจ อีกทั้งยังเป็นส่วนที่ใช้ในการทดสอบการเข้าถึง Restful API ได้ง่ายอีกด้วย

2.5.6 React.js



รูปที่ 2.14 React.js

React.js นั้นเป็น Java Script Framework ซึ่งรองรับความสามารถของ Java Script ECMA 6 ซึ่งถูกพัฒนาโดย Facebook และเปิดให้นักพัฒนาทั่วไปได้นำมาใช้ฟรี ซึ่งเว็บไซต์ในปัจจุบันของ Facebook.com ก็ใช้ React.js อยู่เช่นกัน

React ทำหน้าที่เฉพาะส่วน View (จาก M-V-C: Model-View-Controller) เหมาะกับงาน Web Front-End ที่สามารถแบ่งออกเป็น Web Component ย่อย ๆ โดยหลักการวิเคราะห์ควรแยกให้ย่อยที่สุดเท่าที่จะทำได้ ซึ่งสามารถแบ่ง component ออกเป็น 2 รูปแบบใหญ่ ๆ คือ

- Container สำหรับบรรจุ component ย่อยอื่น ๆ ไม่ควรมีการเก็บค่าใด ๆ (สามารถทำหน้าที่เป็นตัวกลางในการส่งผ่านค่าได้) เน้นไปที่การจัด Layout และบรรจุตัวลูก

- Web Component คือตัวที่ต้อง interact กับผู้ใช้จริง ๆ เช่น ช่องกรอกข้อมูล ลิสต์แสดงข้อมูล ลาเบล ปุ่ม เป็นต้น ซึ่งอาจมีการเก็บค่าบางค่าเอาไว้ (State) เพื่อนำมาแสดงผล

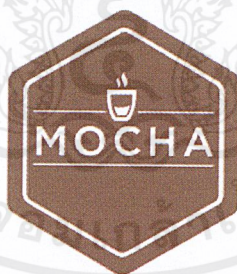
โดย React.js มีส่วนที่เรียกว่า Redux เป็นไลบรารีตัวหนึ่งที่ช่วยให้คุณให้การไหลเวียนข้อมูลภายในเว็บแอปพลิเคชันให้ดีขึ้น กล่าวคือผู้ใช้สามารถคาดการณ์ได้ว่าเมื่อเกิดเหตุการณ์หนึ่ง ๆ ในคอมโพเนนต์แล้วจะมีผลกระทบต่อเว็บแอปพลิเคชันของเราอย่างไร โดยที่ Redux จะเน้นในเรื่องของการจัดการ สถานะต่าง ๆ (State) ของเว็บแอปพลิเคชัน เนื่องจากในระบบงานนั้น จะมี State จำนวนมาก ยิ่งระบบงานมีขนาดใหญ่ขึ้น ยิ่งมากขึ้นไปอีก รวมทั้ง State ส่วนใหญ่มักจะถูกเปลี่ยนแปลงได้ตลอดเวลา Redux จึงเป็นส่วนสำคัญที่เข้ามาช่วยจัดการในส่วนนี้

2.5.7 JSON Web Token

JSON Web Token (JWT) เป็นรหัสชุดหนึ่งที่เอาไว้สำหรับทดแทน session ซึ่งเอาไว้เพื่อเก็บข้อมูลของผู้ใช้งานที่กำลังใช้งานเว็บอยู่ขณะนั้น โดยจะสร้างจากจากข้อมูลที่อยู่ในรูปแบบ JSON แล้วทำการเข้ารหัสด้วย Base64Url Encoded โดย JWT สามารถใช้ได้กับทุกภาษาที่รองรับข้อมูลแบบ JSON และสามารถส่งผ่าน HTTP Methods, URL ได้ง่าย รวดเร็ว และมีความปลอดภัย

2.5.8 เครื่องมือที่ใช้ในการทำ Unit Test

ในการทำ Unit Test ของส่วนติดต่อเซอร์วิสของแอปพลิเคชัน นั้นมีเครื่องมือที่นำมาใช้ช่วยในการทำดังนี้



รูปที่ 2.15 Mocha.js

- Mocha.js เป็น Test Framework ของภาษา Java Script ที่ทำงานอยู่บน Node.js ช่วยให้การทดสอบทำงานอย่างเป็นลำดับและถูกต้อง
- Chai ซึ่งเป็นส่วนเสริม (library) ที่ทำให้ Mocha.js ทำงานได้ดียิ่งขึ้น เช่น การจัดรูปแบบของการเขียน Test ให้ผู้พัฒนาสามารถอ่านได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Sinon เป็นส่วนช่วยในการกำหนดค่า (stubs, mocks) ในการทดสอบ Unit Test บน Test Framework ของภาษา Java Script

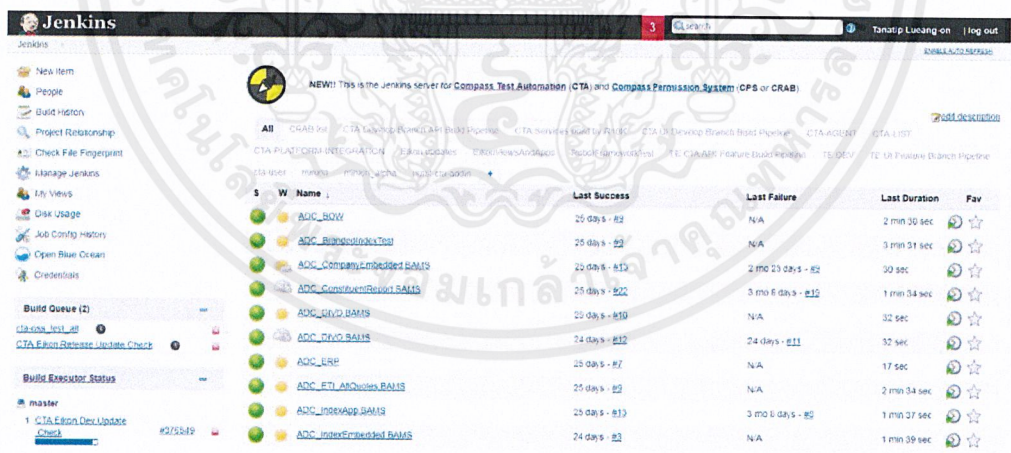
2.5.9 Jenkins



Jenkins

รูปที่ 2.16 Jenkins

Jenkins เป็นเครื่องมือที่ใช้ในการทำงานประเภท ระบบการทำงานอัตโนมัติ (Automation) โดยเป็นซอฟต์แวร์ ประเภท Open Source ถูกพัฒนาขึ้นโดยภาษา Java เพื่อตอบสนองการทำงานด้วยแนวคิด แบบ Continuous Integration (CI) คือเมื่อมีการเปลี่ยนแปลงของ Source Code ระบบจะทำการ Integrate ทุกครั้งและจะทำการ Build และ Automate Test โดยเร็วที่สุด เพื่อให้ได้ผลตอบรับกลับมา การทำงานของ Jenkins จะมาทำหน้าที่ในกระบวนการทำงาน Continuous Integration ตั้งแต่การ Pulling Repository Build Software ทำการทดสอบ และส่งผลแจ้งเตือน Email



รูปที่ 2.17 ส่วนหน้าจอผู้ใช้งานของ Jenkins

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ขั้นตอนในการพัฒนาระบบ

ในการจัดทำส่วนติดต่อเซิร์ฟเวอร์ของเว็บแอปพลิเคชันและเว็บแอปพลิเคชัน ผู้จัดทำต้องการมุ่งเน้นการทำงานที่ได้ผลลัพธ์ที่มีความแม่นยำ มีความรวดเร็ว และมีประสิทธิภาพ โดยผู้จัดทำได้แบ่งขั้นตอนในการดำเนินงานออกเป็น 4 ขั้นตอน ดังนี้

- 3.1 ขั้นตอนการศึกษาและเก็บรวบรวมข้อมูลของระบบ
- 3.2 ขั้นตอนการวิเคราะห์ระบบ
- 3.3 ขั้นตอนการออกแบบระบบ
- 3.4 ขั้นตอนการออกแบบการทดสอบระบบ
- 3.5 ขั้นตอนการพัฒนาระบบ

3.1 ขั้นตอนการศึกษาและเก็บรวบรวมข้อมูลของระบบ

1. ศึกษาโครงสร้างของระบบเดิมที่มีอยู่ เพื่อที่จะได้เข้าใจว่าระบบมีหลักการ การทำงานอย่างไร โดยแบ่งลักษณะการศึกษาได้ดังต่อไปนี้

- ศึกษาโครงสร้างภายในของระบบ (Infrastructure) ซึ่งจะทำให้มีความรู้ ไปใช้ในการพัฒนาระบบให้สอดคล้องกับโครงสร้างภายในของระบบ รวมไปถึงในส่วนขั้นตอนของการติดตั้งระบบลงในเซิร์ฟเวอร์อีกด้วย

- ศึกษาวิธีการใช้งานระบบเดิมของผู้ใช้ ซึ่งผู้พัฒนาระบบจะนำความเข้าใจในส่วนนี้ไปปรับปรุงพัฒนาระบบ

2. ศึกษาหลักการพัฒนา ส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชัน (Application Programming Interface: API) เพื่อที่นำความรู้ในส่วนนี้ ไปใช้ในการเข้าใจหลักการทำงาน, มาตรฐานของการใช้งาน, ข้อกำหนดของการทำงาน ระหว่างส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชันกับระบบที่ทำการพัฒนา โดยแบ่งลักษณะการศึกษาได้ดังต่อไปนี้

- ศึกษาหลักการของการออกแบบ API

- ศึกษาวิธีการใช้งานของ JFrog API

- ศึกษาข้อกำหนดในการใช้งาน API ที่อาจทำให้เกิดผลกระทบแก่ระบบ

3. ศึกษาภาษาและเครื่องมือที่ใช้ในการพัฒนาระบบ

- ศึกษาหลักการทางภาษาของภาษา TypeScript ที่เกี่ยวข้องกับการใช้ส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชัน เพื่อให้ระบบที่พัฒนาสามารถติดต่อเข้าไปใช้งานได้มีประสิทธิภาพ

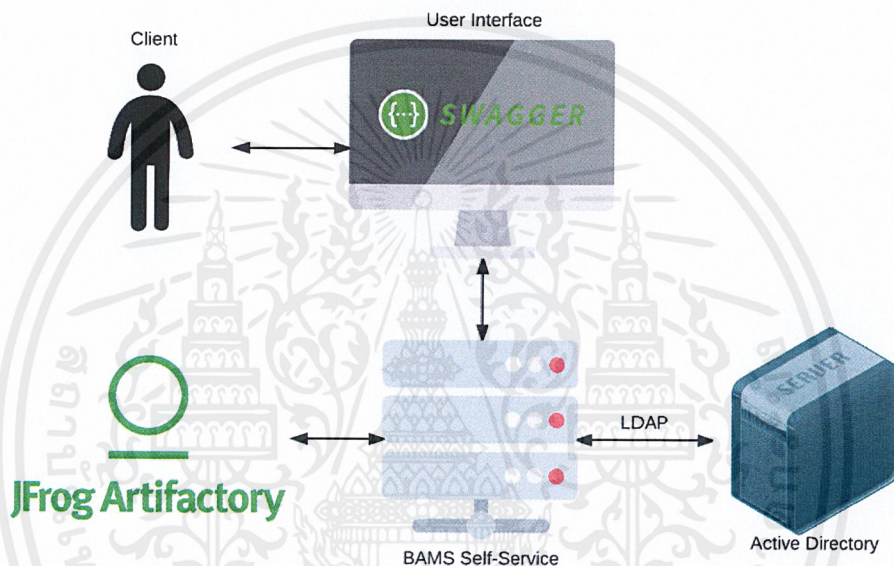
- ศึกษาหลักการทางภาษาของภาษา TypeScript และเครื่องมือที่ใช้ในการเขียน Unit Test ที่เกี่ยวข้องกับการใช้ส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชัน เพื่อความสะดวกในการพัฒนาและทดสอบระบบให้สมบูรณ์

- ศึกษาหลักการใช้งานและเครื่องมือที่เกี่ยวข้องกับการใช้งานร่วมกับ React.js, Redux ที่เกี่ยวข้องกับการพัฒนาหน้าจอส่วนติดต่อผู้ใช้ (User Interface) ของเว็บแอปพลิเคชัน
- ศึกษาวิธีการและเครื่องมือที่เกี่ยวข้องกับการติดตั้งระบบลงในเซิร์ฟเวอร์

3.2 ขั้นตอนการวิเคราะห์ระบบ

3.2.1 สถาปัตยกรรมของระบบ

สถาปัตยกรรมของในส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟกต์แบบไบนารี แสดงได้ดังรูปที่ 3.1

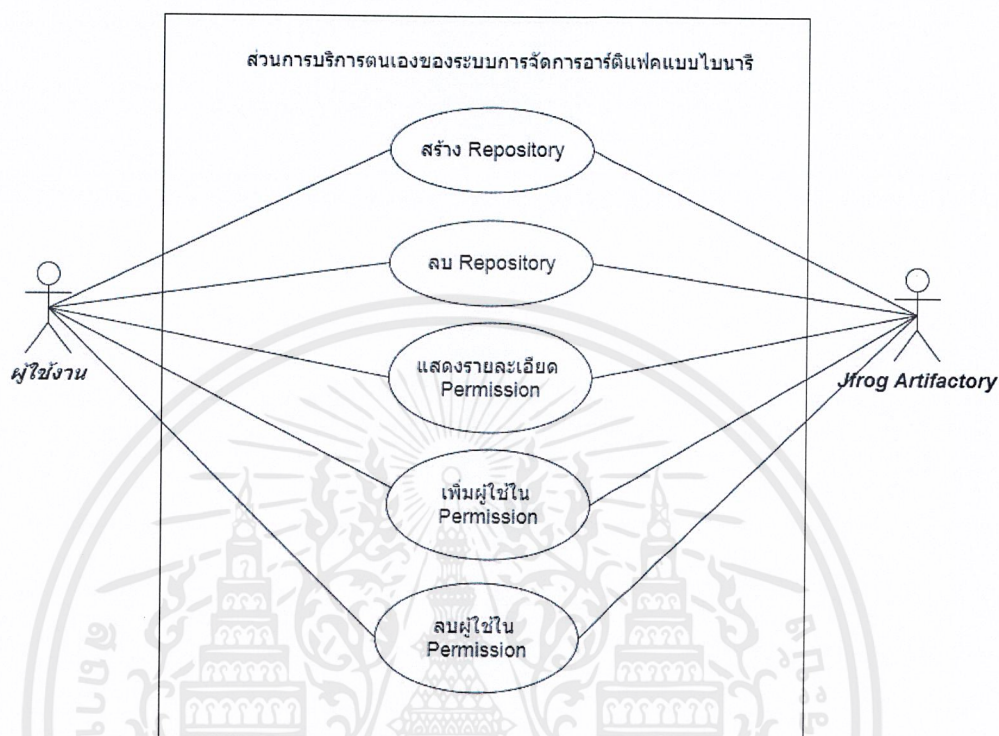


รูปที่ 3.1 โครงสร้างสถาปัตยกรรมของระบบ

จากรูปที่ 3.1 เป็นสถาปัตยกรรมของส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟกต์แบบไบนารี ซึ่งเมื่อผู้ใช้ทำการเชื่อมต่อมายังเว็บแอปพลิเคชัน ระบบจะนำผู้ใช้ไปยังส่วนที่เป็น User interface สำหรับผู้ใช้งาน ซึ่งจะทำการเชื่อมต่อ กับระบบส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟกต์แบบไบนารีโดยในส่วนนี้จะมีการเชื่อมต่อไปหา Active Directory โดยใช้โปรโตคอลที่ชื่อว่า LDAP เพื่อเข้าไปเรียกข้อมูลของการยืนยันตัวตนบุคคลในการเข้าใช้งานระบบ และ เชื่อมต่อกับอีกส่วนก็คือ JFrog Artifactory เพื่อให้ระบบส่วนบริการตนเอง ส่งข้อมูลไปจัดการกับ ระบบการจัดการอาร์ติแฟกต์แบบไบนารี

3.2.2 Use Case Diagram

ในส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟคแบบไบนารี สามารถแสดงผล ได้ดังนี้



รูปที่ 3.2 แผนภาพ Use Case Diagram ของระบบ

จากแผนภาพของ Use Case Diagram เป็นการแสดงองค์ประกอบของส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟคแบบไบนารี โดยสามารถอธิบายได้ดังนี้

1. เลือกรการสร้าง Repository คือ ผู้ใช้งานสามารถสร้างพื้นที่จัดเก็บข้อมูลเป็นของตัวเอง และสามารถนำของเข้ามาไว้ในพื้นที่จัดเก็บข้อมูลนี้ได้
2. เลือกรลบ Repository คือ ผู้ใช้งานสามารถลบพื้นที่จัดเก็บข้อมูลเป็นของตัวเอง และสามารถนำของเข้ามาไว้ในพื้นที่จัดเก็บข้อมูลนี้ได้
3. เลือกรแสดงรายละเอียด Permission คือ แสดงรายละเอียดสิทธิ์การจัดการของผู้ใช้งานในพื้นที่จัดเก็บข้อมูล เพื่อให้รู้ว่าพื้นที่จัดเก็บข้อมูลนี้มีใครมีสิทธิ์ที่จะใช้งานได้อยู่บ้าง
4. เลือกรเพิ่มผู้ใช้งานใน Permission คือ เพิ่มสิทธิ์การจัดการของผู้ใช้งาน เพื่อให้ผู้ใช้งานสามารถเพิ่มผู้ใช้งานคนอื่นมาจัดการพื้นที่จัดเก็บข้อมูลเดียวกันได้
5. เลือกรลบผู้ใช้งานใน Permission คือ ลบสิทธิ์การจัดการของผู้ใช้งาน เพื่อให้ผู้ใช้งานสามารถลบผู้ใช้งานคนอื่นที่หมดสิทธิ์การออกจากพื้นที่จัดเก็บข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ขั้นตอนการออกแบบระบบ

3.3.1 Sequence Diagram

ขั้นตอนกระบวนการทำงานของส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟกต์แบบไบนารี สามารถอธิบายได้อยู่ในรูปแบบของ Sequence Diagram

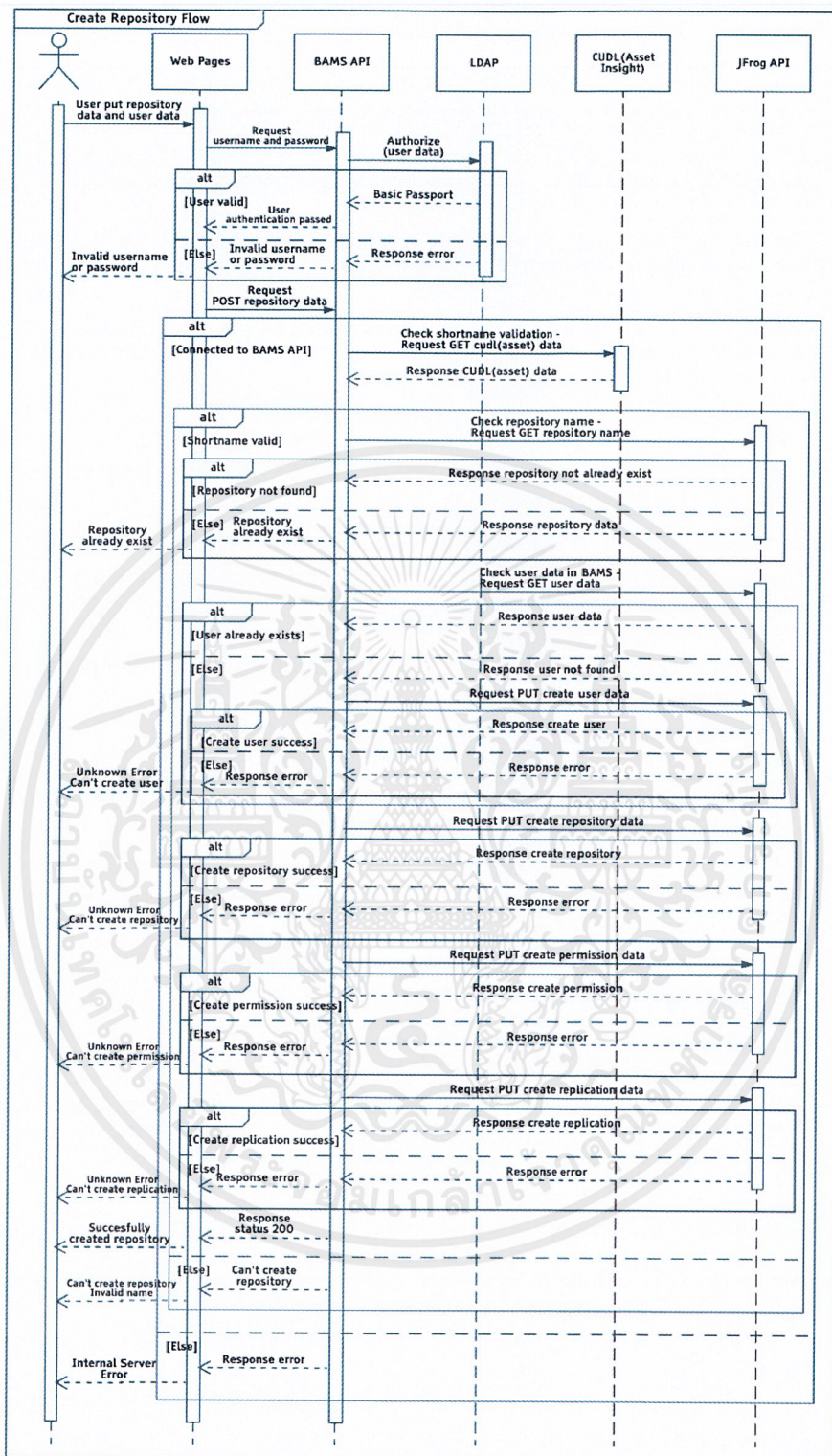
1) ระบบการสร้างพื้นที่จัดเก็บข้อมูล โดยการทำงานจะเริ่มจากผู้ใช้งานเข้ามายังหน้า User Interface และรับรายละเอียดจากผู้ใช้งานซึ่งประกอบไปด้วยข้อมูลของผู้ใช้และข้อมูลการสร้างพื้นที่จัดเก็บข้อมูล จากนั้นทำการส่งข้อมูลผู้ใช้เข้าไปที่เซิร์ฟเวอร์ของ BAMS API เพื่อทำการเรียก LDAP มาตรวจสอบข้อมูลผู้ใช้งานว่ามีข้อมูลผู้ใช้นั้นอยู่จริงหรือไม่ ถ้าไม่มีอยู่ก็จะแสดงผลกลับไปหน้า Web Pages แต่ถ้ามีอยู่ก็จะสามารถเริ่มทำการสร้างพื้นที่จัดเก็บข้อมูลได้ โดยทำการร้องขอในรูปแบบ POST จากหน้าเว็บ ไปยังเซิร์ฟเวอร์ของ BAMS API ซึ่งประกอบไปด้วย Application Short name, Repository Type และ Repository Class

จากนั้นทำการส่ง Short name ไปยัง CUDL จะไปเรียกข้อมูลของ Short name ที่เก็บไว้ใน Asset Insight เพื่อมาตรวจสอบว่า Short name ที่ส่งไปมีอยู่จริงหรือไม่ เมื่อทำการตรวจสอบข้อมูลการสร้างพื้นที่จัดเก็บข้อมูลเสร็จสิ้น ถ้าตรวจสอบชื่อผิดพลาดจะไม่สามารถสร้างพื้นที่จัดเก็บข้อมูลได้ แต่ถ้าตรวจสอบชื่อผ่านจะนำข้อมูลทั้ง 3 อย่างมารวมกันเป็น Repository name หรือชื่อของพื้นที่เก็บข้อมูล โดยนำชื่อทำการร้องขอในรูปแบบ GET ไปยัง JFrog API เพื่อตรวจสอบว่ามีชื่อของพื้นที่เก็บข้อมูลอยู่หรือไม่ ถ้ามีชื่อนี้ ผู้ใช้งานไม่สามารถสร้างพื้นที่จัดเก็บข้อมูลได้ แต่ถ้าไม่มีจะสามารถดำเนินการต่อไปคือระบบตรวจสอบผู้ใช้

โดยนำข้อมูลผู้ใช้ทำการร้องขอในรูปแบบ GET ไปยัง JFrog API เพื่อตรวจสอบว่าข้อมูลผู้ใช้นั้นเคยมีในระบบหรือไม่ ถ้าไม่เคยมีจะนำข้อมูลผู้ใช้ทำการร้องขอในรูปแบบ PUT ไปยัง JFrog API ทำการสร้างข้อมูลของผู้ใช้งานนั้น เพื่อครั้งต่อไปจะได้ไม่ต้องมาสร้างข้อมูลผู้ใช้อีก แต่ถ้าเคยมีอยู่แล้วก็ข้ามขั้นตอนนี้ไป จากนั้นจะเป็นการสร้างพื้นที่จัดเก็บข้อมูล

โดยนำข้อมูลทำการร้องขอในรูปแบบ PUT ไปยัง JFrog API ทำการสร้างพื้นที่จัดเก็บข้อมูล เมื่อทำการสร้างเสร็จ ต่อไประบบจะทำการสร้างสิทธิ์ความเป็นเจ้าของพื้นที่จัดเก็บข้อมูลนั้น โดยนำชื่อผู้ใช้และชื่อพื้นที่จัดเก็บข้อมูลการร้องขอในรูปแบบ PUT ไปยัง JFrog API เพื่อทำการสร้างสิทธิ์ในการจัดการพื้นที่จัดเก็บข้อมูล เมื่อทำการสร้างเสร็จ ต่อไประบบจะทำการสร้างการจำลองแบบไปอีกเซิร์ฟเวอร์หนึ่ง เพื่อข้อมูลในพื้นที่จัดเก็บข้อมูลในเซิร์ฟเวอร์หนึ่งมีการเปลี่ยนแปลง จะทำให้เซิร์ฟเวอร์ที่มีการสร้างจำลองแบบไว้มีข้อมูลนั้นเปลี่ยนแปลงตามไปด้วย

โดยทำการร้องขอในรูปแบบ PUT ไปยัง JFrog API เมื่อสร้างเสร็จระบบจะทำการส่งผลลัพธ์ในรูปแบบรหัส 200 และแสดงผลไปหาผู้ใช้งานว่าสร้างพื้นที่จัดเก็บข้อมูลเสร็จสิ้นทุกขั้นตอนในทุกขั้นตอนของการสร้างพื้นที่จัดเก็บข้อมูล ถ้าเกิดข้อผิดพลาดจะทำการแสดงผลข้อผิดพลาดไปหาผู้ใช้งานทันที



รูปที่ 3.3 แผนภาพ Sequence Diagram แสดงการสร้างพื้นที่จัดเก็บข้อมูล

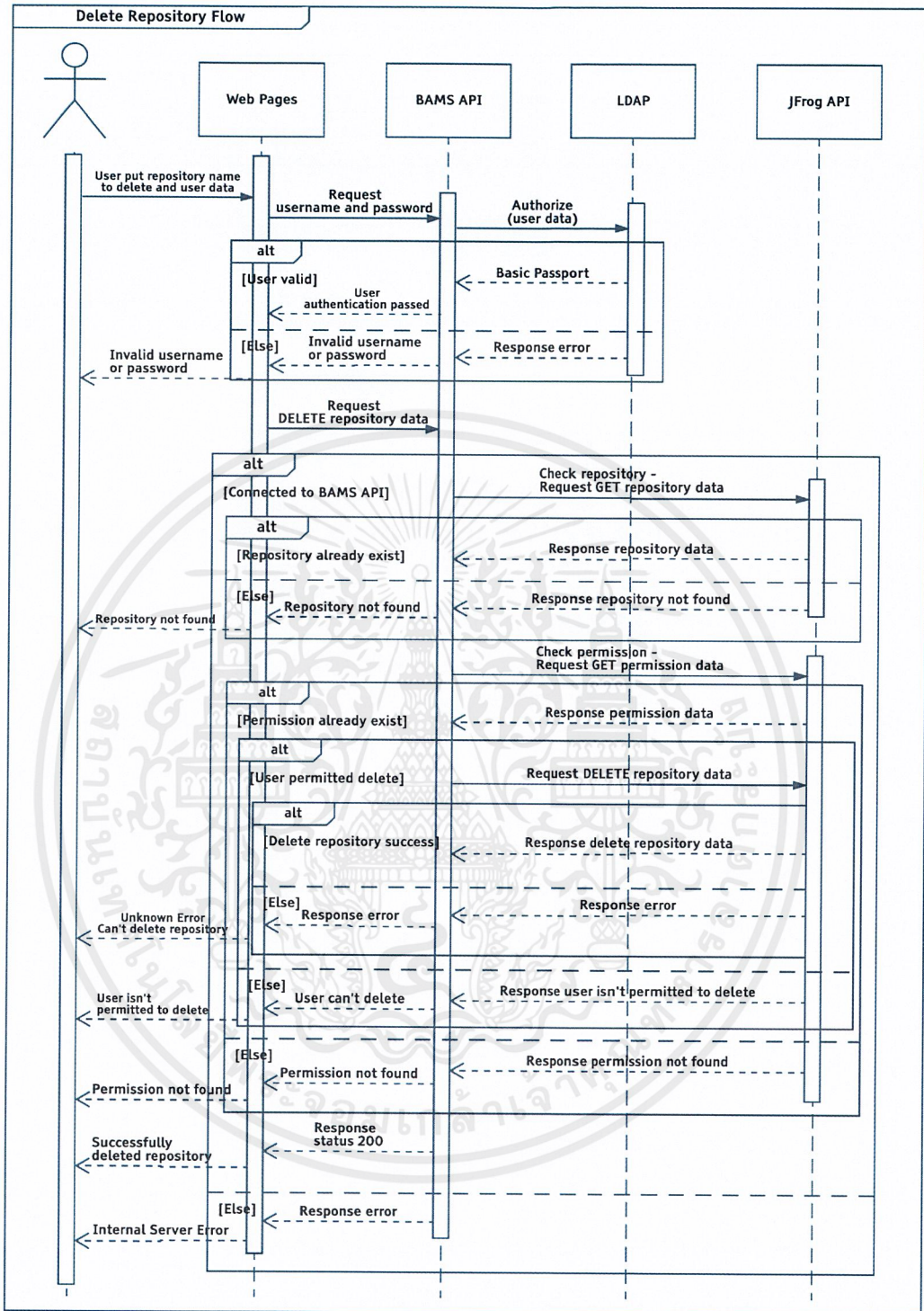
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ระบบการลบพื้นที่จัดเก็บข้อมูล โดยการทำงานจะเริ่มจากผู้ใช้งานเข้ามายังหน้า User Interface และรับรายละเอียดจากผู้ใช้งานซึ่งประกอบไปด้วยข้อมูลของผู้ใช้และชื่อพื้นที่จัดเก็บข้อมูลที่จะทำการลบ จากนั้นทำการส่งข้อมูลผู้ใช้เข้าไปที่เซิร์ฟเวอร์ของ BAMS API เพื่อทำการเรียก LDAP มาตรวจสอบข้อมูลผู้ใช้งานว่ามีข้อมูลผู้ใช้นั้นอยู่จริงหรือไม่ ถ้าไม่มีอยู่ก็จะแสดงผลกลับไปหน้า Web Pages แต่ถ้ามีอยู่ก็จะสามารถเริ่มทำการลบพื้นที่จัดเก็บข้อมูลได้

โดยทำการร้องขอในรูปแบบ DELETE จากหน้าเว็บ ไปยังเซิร์ฟเวอร์ BAMS API และทำการตรวจสอบว่าชื่อพื้นที่จัดเก็บข้อมูลนี้มีอยู่จริงหรือไม่ โดยนำชื่อทำการร้องขอในรูปแบบ GET ไปยัง JFrog API เพื่อตรวจสอบว่ามีชื่อของพื้นที่เก็บข้อมูลอยู่หรือไม่ ถ้าไม่มีชื่อนี้ ผู้ใช้งานไม่สามารถลบพื้นที่จัดเก็บข้อมูลได้และแสดงผลกลับไปหาผู้ใช้งานว่าไม่เจอพื้นที่จัดเก็บข้อมูล แต่ถ้ามีจะสามารถดำเนินการต่อไปคือระบบตรวจสอบสิทธิ์การลบของผู้ใช้

โดยนำชื่อพื้นที่จัดเก็บข้อมูลทำการร้องขอในรูปแบบ GET ไปยัง JFrog API ถ้าหาสิทธิ์การจัดการไม่เจอจะแสดงผลกลับไปหาผู้ใช้งานว่าไม่เจอสิทธิ์การจัดการนี้ ถ้าเจอก็จะได้ข้อมูลมาจาก JFrog API จะทำการตรวจสอบว่าผู้ใช้สามารถมีสิทธิ์ที่จะลบหรือไม่ ถ้าไม่มีจะแสดงผลกลับไปหาผู้ใช้งานว่าไม่สามารถลบพื้นที่จัดเก็บข้อมูลนี้ได้ แต่ถ้ามีก็จะสามารถทำการลบพื้นที่จัดเก็บข้อมูลได้

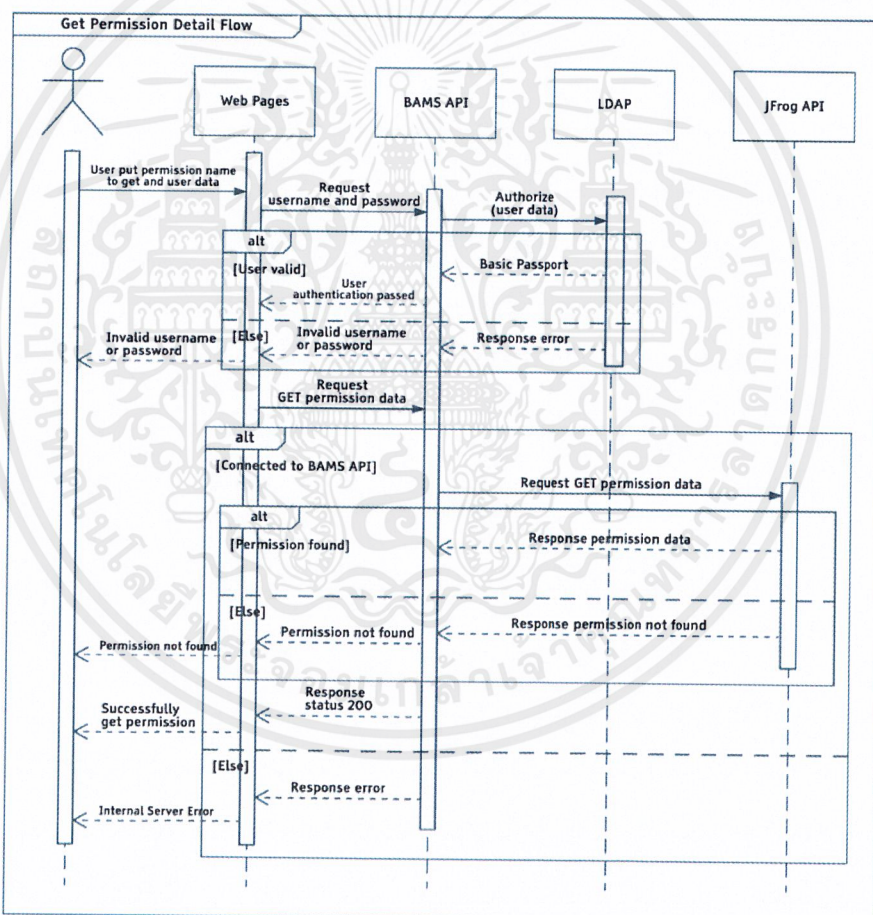
โดยทำการร้องขอในรูปแบบ DELETE จาก BAMS API ไปยัง JFrog API เมื่อลบเสร็จระบบจะทำการส่งผลกลับไปยัง BAMS API และส่งผลกลับในรูปแบบรหัส 200 แสดงผลกลับไปหาผู้ใช้งานว่าลบพื้นที่จัดเก็บข้อมูลเสร็จสิ้น ในทุกขั้นตอนของการลบพื้นที่จัดเก็บข้อมูล ถ้าเกิดข้อผิดพลาดจะทำการแสดงผลข้อผิดพลาดไปหาผู้ใช้งานทันที



รูปที่ 3.4 แผนภาพ Sequence Diagram แสดงการลบพื้นที่จัดเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ระบบการแสดงผลรายละเอียดสิทธิ์การจัดการของผู้ใช้งานในพื้นที่จัดเก็บข้อมูล โดยการทำงานจะเริ่มจากผู้ใช้งานเข้ามามายังหน้า User Interface และรับรายละเอียดจากผู้ใช้งานซึ่งประกอบไปด้วยข้อมูลของผู้ใช้และชื่อพื้นที่จัดเก็บข้อมูลที่ต้องการแสดงผลรายละเอียดสิทธิ์การจัดการ จากนั้นทำการส่งข้อมูลผู้ใช้เข้าไปที่เซิร์ฟเวอร์ของ BAMS API เพื่อทำการเรียก LDAP มาตรวจสอบข้อมูลผู้ใช้งานว่ามีข้อมูลผู้ใช้นั้นอยู่จริงหรือไม่ ถ้าไม่มีอยู่ก็จะแสดงผลกลับไปหน้าเว็บ แต่ถ้ามีอยู่ก็จะสามารถเริ่มทำการแสดงผลรายละเอียดสิทธิ์การจัดการของผู้ใช้งานได้ โดยทำการร้องขอในรูปแบบ GET จากหน้าเว็บไปยังเซิร์ฟเวอร์ BAMS API และทำการตรวจสอบว่าสิทธิ์การจัดการนี้มีอยู่จริงหรือไม่ โดยนำชื่อพื้นที่จัดเก็บข้อมูลทำการร้องขอในรูปแบบ GET ไปยัง JFrog API ถ้าหาสิทธิ์การจัดการไม่เจอจะแสดงผลกลับไปหาผู้ใช้งานว่าไม่เจอสิทธิ์การจัดการนี้ ถ้าเจอก็จะได้อข้อมูลมาจาก JFrog API ส่งคืนกลับไปยัง BAMS API และส่งผลกลับไปในรูปแบบรหัส 200 แสดงผลรายละเอียดสิทธิ์การจัดการกลับไปหาผู้ใช้



รูปที่ 3.5 แผนภาพ Sequence Diagram แสดงการแสดงผลรายละเอียดสิทธิ์การจัดการของผู้ใช้งานในพื้นที่จัดเก็บข้อมูล

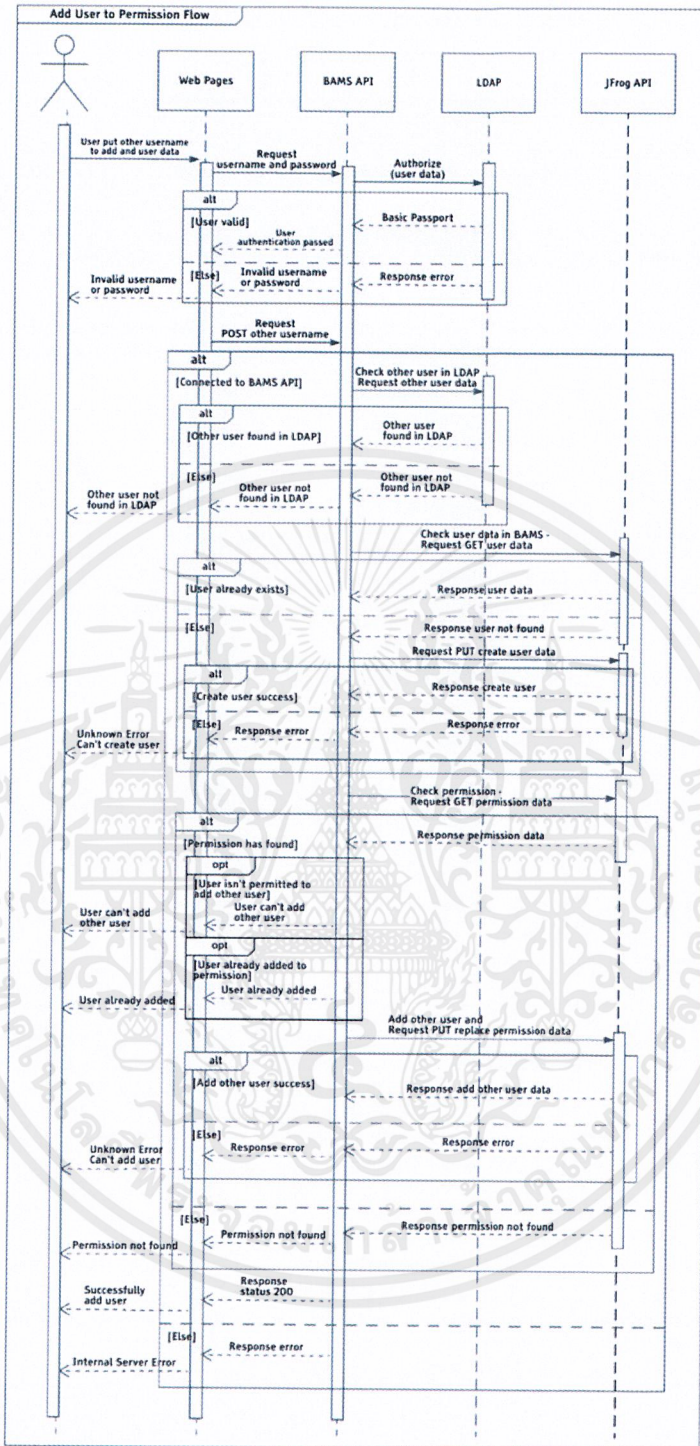
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ระบบการเพิ่มสิทธิ์การจัดการของผู้ใช้งาน โดยการทำงานจะเริ่มจากผู้ใช้งานเข้ามาฝั่งหน้า User Interface และรับรายละเอียดจากผู้ใช้งานซึ่งประกอบไปด้วยข้อมูลของผู้ใช้และชื่อผู้ใช้อื่นที่จะเพิ่มสิทธิ์การจัดการ จากนั้นทำการส่งข้อมูลผู้ใช้เข้าไปที่เซิร์ฟเวอร์ของ BAMS API เพื่อทำการเรียก LDAP มาตรวจสอบข้อมูลผู้ใช้นั้นมีข้อมูลผู้ใช้นั้นอยู่จริงหรือไม่ ถ้าไม่มีอยู่ก็จะแสดงผลกลับไปหน้า Web Pages แต่ถ้ามีอยู่ก็จะสามารถเริ่มทำการเพิ่มสิทธิ์การจัดการของผู้ใช้งานได้

โดยนำชื่อผู้ใช้อื่นทำการร้องขอในรูปแบบ POST จากหน้าเว็บไปยังเซิร์ฟเวอร์ของ BAMS API จากนั้นทำการตรวจสอบว่าชื่อผู้ใช้อื่นที่จะเพิ่มนี้มีตัวตนจริงหรือไม่ โดยนำชื่อผู้ใช้อื่นทำการร้องขอในรูปแบบ GET ไปยัง LDAP ถ้าหาชื่อผู้ใช้ไม่เจอจะแสดงผลกลับไปหาผู้ใช้ว่าไม่เจอชื่อผู้ใช้นี้ใน LDAP ถ้าเจอก็จะได้อ้อมูลมาจาก LDAP ส่งคืนกลับไปยัง BAMS API ว่ามีตัวตนอยู่จริง จากนั้นทำการตรวจสอบผู้ใช้ โดยนำข้อมูลผู้ใช้ทำการร้องขอในรูปแบบ GET ไปยัง JFrog API เพื่อตรวจสอบว่าข้อมูลผู้ใช้นี้เคยมีในระบบหรือไม่ ถ้าไม่เคยมีจะนำข้อมูลผู้ใช้ทำการร้องขอในรูปแบบ PUT ไปยัง JFrog API ทำการสร้างข้อมูลของผู้ใช้งานนั้น เพื่อครั้งต่อไปจะได้ไม่ต้องมาสร้างข้อมูลผู้ใช้ อีก แต่ถ้าเคยมีอยู่แล้วก็ข้ามขั้นตอนนี้ไปและส่งผลกลับไปยัง BAMS API

จากนั้นทำการตรวจสอบว่าสิทธิ์การจัดการนี้มีอยู่จริงหรือไม่ โดยนำชื่อพื้นที่จัดเก็บข้อมูลทำการร้องขอในรูปแบบ GET ไปยัง JFrog API ถ้าหาสิทธิ์การจัดการไม่เจอจะแสดงผลกลับไปหาผู้ใช้ว่าไม่เจอสิทธิ์การจัดการนี้ ถ้าเจอก็จะได้อ้อมูลมาจาก JFrog API ส่งคืนกลับไปยัง BAMS API จะทำการตรวจสอบว่าผู้ใช้สามารถทำการเพิ่มสิทธิ์ได้หรือไม่ ถ้าไม่มีจะแสดงผลกลับไปหาผู้ใช้ว่าไม่สามารถเพิ่มสิทธิ์การจัดการได้ แต่ถ้ามีจะทำการตรวจสอบอีกว่าเคยเพิ่มชื่อผู้ใช้นี้ไปหรือยัง ถ้ามีอยู่จะแสดงผลกลับไปหาผู้ใช้ว่ามีชื่อผู้ใช้ที่จะเพิ่มแล้ว แต่ถ้ายังไม่มีก็สามารถเพิ่มสิทธิ์การจัดการของผู้ใช้งานได้

จากนั้นทำการสร้างสิทธิ์การจัดการใหม่ โดยนำข้อมูลสิทธิ์การจัดการที่เพิ่มชื่อผู้ใช้เข้าไปแล้วทำการร้องขอในรูปแบบ PUT ไปยัง JFrog API เพื่อทำการแทนที่สิทธิ์การจัดการตัวเก่าที่ไม่มีชื่อผู้ใช้นั้น และส่งผลกลับไปในรูปแบบรหัส 200 แสดงผลกลับไปหาผู้ใช้ว่าเพิ่มสิทธิ์การจัดการของผู้ใช้งานเสร็จสิ้น ในทุกขั้นตอนของการเพิ่มสิทธิ์การจัดการของผู้ใช้งาน ถ้าเกิดข้อผิดพลาดจะทำการแสดงผลข้อผิดพลาดไปหาผู้ใช้งานทันที



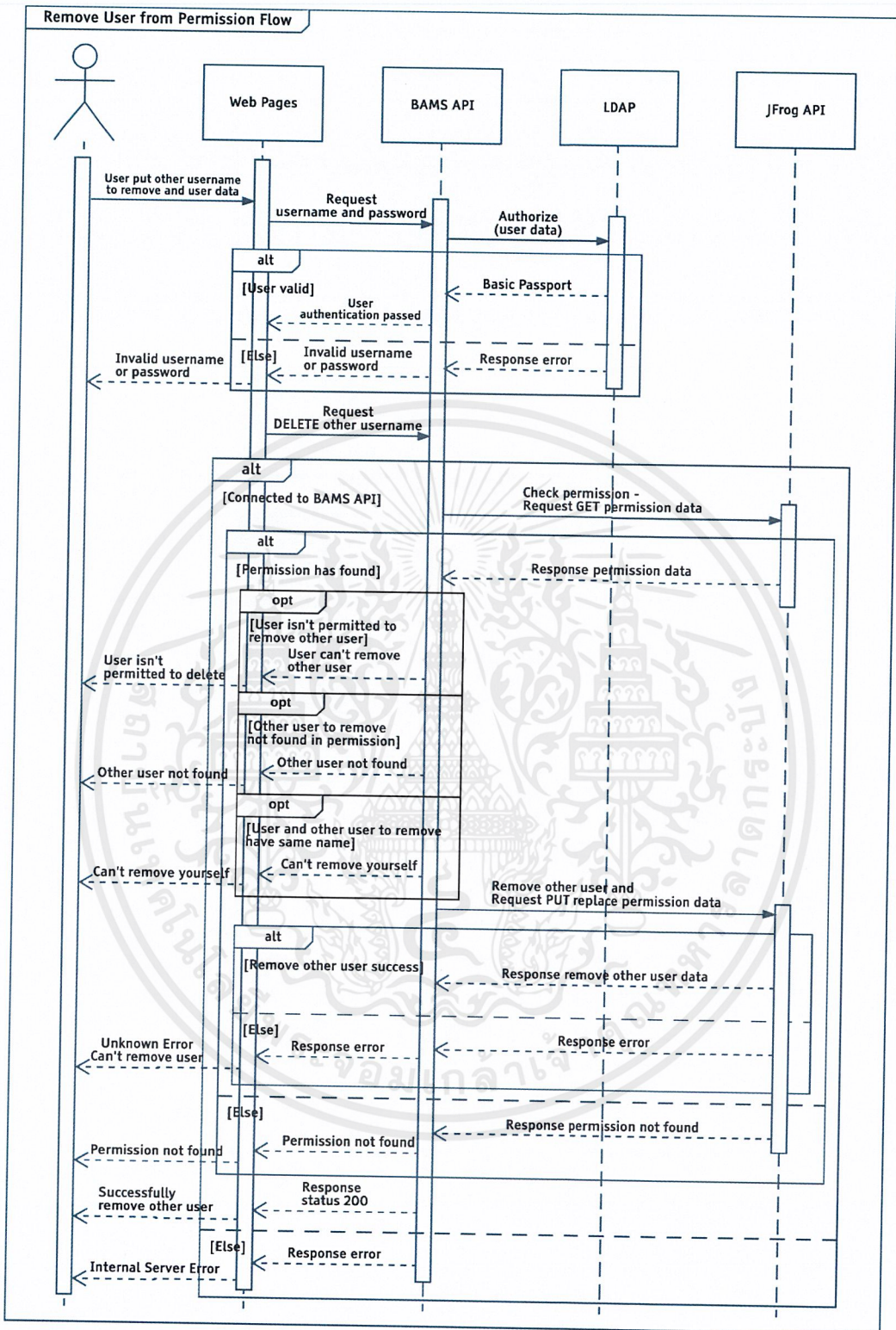
รูปที่ 3.6 แผนภาพ Sequence Diagram แสดงการเพิ่มสิทธิ์การจัดการของผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) ระบบการลบสิทธิ์การจัดการของผู้ใช้งาน โดยการทำงานจะเริ่มจากผู้ใช้งานเข้ามายังหน้า User Interface และรับรายละเอียดจากผู้ใช้งานซึ่งประกอบไปด้วยข้อมูลของผู้ใช้และชื่อผู้ใช้อื่นที่จะลบสิทธิ์การจัดการ จากนั้นทำการส่งข้อมูลผู้ใช้เข้าไปที่เซิร์ฟเวอร์ของ BAMS API เพื่อทำการเรียก LDAP มาตรวจสอบข้อมูลผู้ใช้นั้นมีข้อมูลผู้ใช้นั้นอยู่จริงหรือไม่ ถ้าไม่มีอยู่ก็จะแสดงผลกลับไปหน้า Web Pages แต่ถ้ามีอยู่ก็จะสามารถเริ่มทำการลบสิทธิ์การจัดการของผู้ใช้งานได้

โดยนำชื่อผู้ใช้ที่จะลบทำการร้องขอในรูปแบบ DELETE จากหน้าเว็บไปยังเซิร์ฟเวอร์ของ BAMS API จากนั้นทำการตรวจสอบว่าสิทธิ์การจัดการนี้มีอยู่จริงหรือไม่ โดยนำชื่อที่จัดเก็บข้อมูลทำการร้องขอในรูปแบบ GET ไปยัง JFrog API ถ้าหาสิทธิ์การจัดการไม่เจอจะแสดงผลกลับไปหาผู้ใช้งานว่าไม่เจอสิทธิ์การจัดการนี้ ถ้าเจอก็จะได้ข้อมูลมาจาก JFrog API ส่งคืนกลับไปยัง BAMS API จะทำการตรวจสอบว่าผู้ใช้สามารถทำการลบสิทธิ์ได้หรือไม่ ถ้าไม่มีจะแสดงผลกลับไปหาผู้ใช้งานว่าไม่สามารถลบสิทธิ์การจัดการได้ แต่ถ้ามีจะทำการตรวจสอบอีกว่ามีชื่อผู้ใช้ที่จะลบอยู่ในสิทธิ์การจัดการนี้หรือไม่ ถ้าไม่มีอยู่จะแสดงผลกลับไปหาผู้ใช้งานว่าไม่เจอชื่อผู้ใช้ที่จะลบแล้ว แต่ถ้ามีจะทำการตรวจสอบอีกว่าชื่อผู้ใช้ที่จะลบเหมือนกับชื่อผู้ใช้ของตัวเองหรือไม่ ถ้าเหมือนจะแสดงผลกลับไปหาผู้ใช้งานว่าไม่สามารถลบสิทธิ์การจัดการของตัวเองได้ ถ้าไม่เหมือนก็สามารถลบสิทธิ์การจัดการของผู้ใช้งานได้

จากนั้นทำการสร้างสิทธิ์การจัดการใหม่ โดยนำข้อมูลสิทธิ์การจัดการที่ลบชื่อผู้ใช้ออกไปแล้วทำการร้องขอในรูปแบบ PUT ไปยัง JFrog API เพื่อทำการแทนที่สิทธิ์การจัดการตัวเก่าที่มีชื่อผู้ใช้ที่ยังไม่ลบอยู่ และส่งผลลัพธ์ในรูปแบบรหัส 200 แสดงผลกลับไปหาผู้ใช้งานว่าลบสิทธิ์การจัดการของผู้ใช้งานเสร็จสิ้น ในทุกขั้นตอนของการเพิ่มสิทธิ์การจัดการของผู้ใช้งาน ถ้าเกิดข้อผิดพลาดจะทำการแสดงผลข้อผิดพลาดไปหาผู้ใช้งานทันที



รูปที่ 3.7 แผนภาพ Sequence Diagram แสดงการลบสิทธิ์การจัดการของผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ขั้นตอนการออกแบบการทดสอบระบบ

ในส่วนนี้จะเป็นการออกแบบวิธีการทดสอบเฉพาะส่วน (Unit testing) ในระบบต่างๆ ของ BAMS API เพื่อหาข้อผิดพลาดในการทำงานของทั้งระบบโดยสามารถแบ่งการทดสอบระบบได้ออกเป็น 7 ส่วนหลัก คือ 1) ระบบการสร้างพื้นที่จัดเก็บข้อมูล 2) ระบบการอัปเดตพื้นที่จัดเก็บข้อมูล 3) ระบบการลบพื้นที่จัดเก็บข้อมูล 4) ระบบการแสดงรายละเอียดสิทธิ์การจัดการของผู้ใช้งานในพื้นที่จัดเก็บข้อมูล 5) ระบบการเพิ่มสิทธิ์การจัดการของผู้ใช้งาน 6) ระบบการลบสิทธิ์การจัดการของผู้ใช้งาน 7) ระบบการเปิดเผยหรือปกปิดพื้นที่จัดเก็บข้อมูลของผู้ใช้งาน ซึ่งจะมีรายการทดสอบดังต่อไปนี้

- 1) ระบบ BAMS API สามารถใช้งาน LDAP ได้ผลอย่างถูกต้อง
- 2) เมื่อผู้ใช้งานทำการเข้าสู่ระบบ ในระบบ BAMS API จะต้องมียุทธศาสตร์ Token เก็บอยู่ในระบบ
- 3) ระบบการสร้างพื้นที่จัดเก็บข้อมูล สามารถสร้างได้อย่างถูกต้องครบถ้วนสมบูรณ์ ในแต่ละรูปแบบของพื้นที่จัดเก็บข้อมูล ตามที่ผู้ใช้งานได้กรอกข้อมูลไว้
- 4) ระบบการอัปเดตพื้นที่จัดเก็บข้อมูล สามารถจัดการเปลี่ยนแปลงค่าในพื้นที่จัดเก็บข้อมูลได้อย่างถูกต้อง ตามที่ผู้ใช้งานได้กรอกข้อมูลไว้
- 5) ระบบการลบพื้นที่จัดเก็บข้อมูล สามารถลบได้ครบถ้วนสมบูรณ์ ตามที่ผู้ใช้งานได้เลือกไว้
- 6) ระบบการแสดงรายละเอียดสิทธิ์การจัดการของผู้ใช้งานในพื้นที่จัดเก็บข้อมูล ต้องแสดงข้อมูลให้ครบถ้วน
- 7) ระบบการเพิ่มสิทธิ์การจัดการของผู้ใช้งาน ต้องให้ผู้ใช้งานคนใหม่ที่ถูกเพิ่มเข้าไปมีชื่ออยู่ในสิทธิ์การจัดการได้อย่างถูกต้อง
- 8) ระบบการลบสิทธิ์การจัดการของผู้ใช้งาน ต้องให้ผู้ใช้งานที่ต้องการจะนำออกจากสิทธิ์การจัดการ สามารถนำชื่อออกได้อย่างถูกต้อง
- 9) ระบบการเปิดเผยหรือปกปิดพื้นที่จัดเก็บข้อมูลของผู้ใช้งาน สามารถเปิดเผยหรือปกปิดพื้นที่จัดเก็บข้อมูลได้ตามความต้องการของผู้ใช้งานได้อย่างถูกต้อง

จากรายการทดสอบข้างต้นสามารถอธิบายรายละเอียดการทดสอบได้เป็นตารางการทดสอบดังนี้ 1) ตารางแสดงรายการทดสอบการใช้งาน LDAP ในระบบ BAMS API เป็นตารางที่ประกอบไปด้วยการทดสอบว่า LDAP สามารถพบข้อมูลของผู้ใช้งานหรือไม่ สามารถอธิบายการทดสอบได้ดังตารางที่ 3.1

ตารางที่ 3.1 การทดสอบการใช้งาน LDAP ในระบบ BAMS API

รายการทดสอบ	รายละเอียด
Should return LDAP data if username and password are valid	ถ้าพบข้อมูลของผู้ใช้งานใน LDAP BAMS API จะคืนค่าข้อมูล
Should have error handling if user not found	ถ้าไม่พบข้อมูลของผู้ใช้งานใน LDAP BAMS API จะมีการจัดการข้อผิดพลาดของระบบ
Should call LDAP with OU=Users	ระบบจะทำการค้นหาบัญชีประเภทผู้ใช้งาน
Should call LDAP with OU=Service Accounts	ระบบจะทำการค้นหาบัญชีประเภท Service Accounts

2) ตารางแสดงรายการทดสอบการเข้าสู่ระบบการใช้งานของ BAMS API เป็นตารางที่ประกอบไปด้วยการทดสอบว่า เมื่อผู้ใช้งานทำการเข้าสู่ระบบ ในระบบ BAMS API จะต้องมีค่า Token เก็บอยู่ในระบบ ซึ่ง Token จะเป็นแบบ JWT สามารถอธิบายการทดสอบได้ดังตารางที่ 3.2

ตารางที่ 3.2 การทดสอบการเข้าสู่ระบบการใช้งานของ BAMS API

รายการทดสอบ	รายละเอียด
Should get token	ถ้าผู้ใช้งานเข้าสู่ระบบอยู่ ระบบสามารถแสดงค่า Token ออกมาได้
Should return error code if username and password is invalid	ถ้าไม่พบข้อมูลของผู้ใช้งานในการเข้าสู่ระบบ ระบบจะทำการคืนค่าออกมาในรูปแบบรหัสข้อผิดพลาด
Should pass if token is valid	ผู้ใช้งานสามารถใช้งานระบบนี้ได้ ถ้าในระบบพบว่า มี Token อยู่
Should return error code if token is invalid	ผู้ใช้งานไม่สามารถใช้งานระบบนี้ได้ ถ้าในระบบไม่พบว่า มี Token อยู่ และจะทำการคืนค่าออกมาในรูปแบบรหัสข้อผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ตารางแสดงรายการทดสอบระบบการสร้างพื้นที่จัดเก็บข้อมูล เป็นตารางที่ประกอบไปด้วยการทดสอบว่า ระบบการสร้างพื้นที่จัดเก็บข้อมูลสามารถสร้างได้อย่างถูกต้องครบถ้วนและสมบูรณ์ ในแต่ละรูปแบบของพื้นที่จัดเก็บข้อมูล ตามที่ผู้ใช้งานได้กรอกข้อมูลไว้ ซึ่งในระบบนี้มีการทดสอบการทำงานย่อยในหลากหลายขั้นตอน สามารถอธิบายการทดสอบได้ดังตารางที่ 3.3

ตารางที่ 3.3 การทดสอบระบบการสร้างพื้นที่จัดเก็บข้อมูล

รายการทดสอบ	รายละเอียด
API Testing	
Users can create repository	ผู้ใช้งานสามารถสร้างพื้นที่จัดเก็บข้อมูลได้อย่างถูกต้อง
Users can create permission	ผู้ใช้งานสามารถสร้างสิทธิ์การจัดเก็บข้อมูลได้อย่างสมบูรณ์และถูกต้อง
Error Handling	
Check repository already exist	ระบบทำการทดสอบว่า มีพื้นที่จัดเก็บข้อมูลชื่อเดียวกันกับที่มีอยู่แล้ว
Users cannot create repository if repository name is wrong	ผู้ใช้งานไม่สามารถสร้างพื้นที่จัดเก็บข้อมูลได้ ถ้าชื่อของพื้นที่จัดเก็บข้อมูลผิดรูปแบบ
Users cannot create permission if user data has problem	ผู้ใช้งานไม่สามารถสร้างสิทธิ์การจัดเก็บข้อมูลได้ ถ้าข้อมูลผู้ใช้งานมีปัญหา

4) ตารางแสดงรายการทดสอบระบบการอัปเดตพื้นที่จัดเก็บข้อมูล เป็นตารางที่ประกอบไปด้วยการทดสอบว่า ระบบการอัปเดตพื้นที่จัดเก็บข้อมูลสามารถจัดการเปลี่ยนแปลงค่าได้อย่างถูกต้องตามที่ผู้ใช้งานได้กรอกข้อมูลไว้ ซึ่งในระบบนี้มีการทดสอบการทำงานย่อยในหลากหลายขั้นตอน สามารถอธิบายการทดสอบได้ดังตารางที่ 3.4

ตารางที่ 3.4 การทดสอบระบบการอัปเดตพื้นที่จัดเก็บข้อมูล

รายการทดสอบ	รายละเอียด
API Testing	
Check have repository to update	ระบบทำการทดสอบว่า ต้องมีพื้นที่จัดเก็บข้อมูลถึงจะทำการอัปเดตได้
Users can update repository	ผู้ใช้งานสามารถอัปเดตพื้นที่จัดเก็บข้อมูลตามที่เลือกไว้ได้อย่างถูกต้อง
Error Handling	
User is not permitted to update repository	ผู้ใช้งานไม่อนุญาตอัปเดตพื้นที่จัดเก็บข้อมูลนี้ได้ เนื่องจากผู้ใช้งานไม่มีสิทธิ์
Users cannot update repository if server has problem	ผู้ใช้งานไม่สามารถอัปเดตพื้นที่จัดเก็บข้อมูลได้ ถ้าเซิร์ฟเวอร์มีปัญหา

5) ตารางแสดงรายการทดสอบระบบการลบพื้นที่จัดเก็บข้อมูล เป็นตารางที่ประกอบไปด้วย การทดสอบว่า ระบบการลบพื้นที่จัดเก็บข้อมูลสามารถสร้างได้อย่างถูกต้องและสมบูรณ์ ตามที่ผู้ใช้งานได้เลือกไว้ ซึ่งในระบบนี้มีการทดสอบการทำงานย่อยในหลากหลายขั้นตอน สามารถอธิบายการทดสอบได้ดังตารางที่ 3.5

ตารางที่ 3.5 การทดสอบระบบการลบพื้นที่จัดเก็บข้อมูล

รายการทดสอบ	รายละเอียด
API Testing	
Check have repository to delete	ระบบทำการทดสอบว่า ต้องมีพื้นที่จัดเก็บข้อมูลถึงจะทำการลบได้
Users can delete repository	ผู้ใช้งานสามารถลบพื้นที่จัดเก็บข้อมูล ตามที่เลือกไว้ได้อย่างถูกต้อง
Error Handling	
User is not permitted to delete repository	ผู้ใช้งานไม่อนุญาตลบพื้นที่จัดเก็บข้อมูลนี้ได้ เนื่องจากผู้ใช้งานไม่มีสิทธิ์
Users cannot delete repository if server has problem	ผู้ใช้งานไม่สามารถลบพื้นที่จัดเก็บข้อมูลได้ ถ้าเซิร์ฟเวอร์มีปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) ตารางแสดงรายการทดสอบระบบการแสดงผลละเอียดสิทธิ์การจัดการของผู้ใช้งานในพื้นที่จัดเก็บข้อมูล เป็นตารางที่ประกอบไปด้วยการทดสอบว่า ระบบแสดงผลละเอียดนี้สามารถแสดงได้อย่างถูกต้องและครบถ้วน สามารถอธิบายการทดสอบได้ดังตารางที่ 3.6

ตารางที่ 3.6 การทดสอบระบบการแสดงผลละเอียดสิทธิ์การจัดการของผู้ใช้งานในพื้นที่จัดเก็บข้อมูล

รายการทดสอบ	รายละเอียด
API Testing	
Check have repository to show detail	ระบบทำการทดสอบว่า ต้องมีพื้นที่จัดเก็บข้อมูลถึงจะทำการแสดงผลละเอียดได้
Users can get detail from repository	ผู้ใช้งานได้รับการแสดงผลละเอียดสิทธิ์การจัดเก็บข้อมูลได้อย่างถูกต้อง
Error Handling	
Users cannot get detail repository if server has problem	ผู้ใช้งานไม่สามารถได้รับการแสดงผลละเอียดสิทธิ์การจัดเก็บข้อมูลได้ ถ้าเซิร์ฟเวอร์มีปัญหา

7) ตารางแสดงรายการทดสอบระบบการเพิ่มสิทธิ์การจัดการของผู้ใช้งาน เป็นตารางที่ประกอบไปด้วยการทดสอบว่า ระบบการเพิ่มสิทธิ์การจัดการของผู้ใช้งาน ต้องให้ผู้ใช้งานคนใหม่ที่ถูกเพิ่มเข้าไปมีชื่ออยู่ในสิทธิ์การจัดการได้อย่างถูกต้อง ซึ่งในระบบนี้มีการทดสอบการทำงานย่อยในหลากหลายขั้นตอน สามารถอธิบายการทดสอบได้ดังตารางที่ 3.7

ตารางที่ 3.7 การทดสอบระบบการเพิ่มสิทธิ์การจัดการของผู้ใช้งาน

รายการทดสอบ	รายละเอียด
API Testing	
Check LDAP user data to add permission	ระบบทำการทดสอบว่า ข้อมูลผู้ใช้งานใหม่ที่ถูกเพิ่มเข้าสิทธิ์การจัดการต้องมีอยู่ในข้อมูลของ LDAP
Users can add user to permission	ผู้ใช้งานสามารถเพิ่มผู้ใช้งานใหม่เข้าไปในสิทธิ์การจัดการได้

ตารางที่ 3.7 การทดสอบระบบการเพิ่มสิทธิ์การจัดการของผู้ใช้งาน (ต่อ)

รายการทดสอบ	รายละเอียด
Error Handling	
Check add user exist in permission	ระบบทำการทดสอบว่า มีชื่อของผู้ใช้งานใหม่ที่จะทำการเพิ่มอยู่ในสิทธิ์การจัดการแล้ว
User is not permitted to add permission	ผู้ใช้งานไม่อนุญาตให้เพิ่มผู้ใช้งานใหม่เข้าไปในสิทธิ์การจัดการนี้ได้ เนื่องจากผู้ใช้งานไม่มีสิทธิ์
Users cannot add permission if server has problem	ผู้ใช้งานไม่สามารถเพิ่มผู้ใช้งานใหม่เข้าไปในสิทธิ์การจัดการได้ ถ้าเซิร์ฟเวอร์มีปัญหา

8) ตารางแสดงรายการทดสอบระบบการลบสิทธิ์การจัดการของผู้ใช้งาน เป็นตารางที่ประกอบไปด้วยการทดสอบว่า ระบบการลบสิทธิ์การจัดการของผู้ใช้งาน ต้องให้ผู้ใช้งานที่ต้องการจะนำออกจากสิทธิ์การจัดการ สามารถนำชื่อออกได้อย่างถูกต้อง ซึ่งในระบบนี้มีการทดสอบการทำงานย่อยในหลากหลายขั้นตอน สามารถอธิบายการทดสอบได้ดังตารางที่ 3.8

ตารางที่ 3.8 การทดสอบระบบการลบสิทธิ์การจัดการของผู้ใช้งาน

รายการทดสอบ	รายละเอียด
API Testing	
Check have user permission to remove user	ระบบทำการทดสอบว่า ต้องมีชื่อของผู้ใช้งานเดิมอยู่ ถึงจะทำการลบออกจากสิทธิ์การจัดการได้
Users can remove user from permission	ผู้ใช้งานสามารถลบผู้ใช้งานเดิมออกจากสิทธิ์การจัดการได้
Error Handling	
User is not permitted to remove permission	ผู้ใช้งานไม่อนุญาตให้ลบผู้ใช้งานเดิมออกจากสิทธิ์การจัดการนี้ได้ เนื่องจากผู้ใช้งานไม่มีสิทธิ์
User cannot remove yourself	ผู้ใช้งานไม่สามารถลบชื่อตัวเองออกจากสิทธิ์การจัดการได้
Users cannot remove permission if server has problem	ผู้ใช้งานไม่สามารถลบผู้ใช้งานเดิมออกจากสิทธิ์การจัดการนี้ได้ ถ้าเซิร์ฟเวอร์มีปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9) ตารางแสดงรายการทดสอบระบบการเปิดเผยหรือปกปิดพื้นที่จัดเก็บข้อมูลของผู้ใช้งาน สามารถเปิดเผยหรือปกปิดพื้นที่จัดเก็บข้อมูลได้ตามความต้องการของผู้ใช้งานได้อย่างถูกต้อง สามารถอธิบายการทดสอบได้ดังตารางที่ 3.9

ตารางที่ 3.9 การทดสอบระบบการเปิดเผยหรือปกปิดพื้นที่จัดเก็บข้อมูลของผู้ใช้งาน

รายการทดสอบ	รายละเอียด
Users can change private repository to public repository	ผู้ใช้งานสามารถปรับเปลี่ยนพื้นที่จัดเก็บข้อมูลแบบปกปิดเป็นแบบเปิดเผยได้
Users can change public repository to private repository	ผู้ใช้งานสามารถปรับเปลี่ยนพื้นที่จัดเก็บข้อมูลแบบเปิดเผยเป็นแบบปกปิดได้
Users cannot change repository if server has problem	ผู้ใช้งานไม่สามารถเปลี่ยนแปลงพื้นที่จัดเก็บข้อมูลได้ ถ้าเซิร์ฟเวอร์มีปัญหา

3.5 ขั้นตอนการพัฒนาระบบ

จากขั้นตอนศึกษาและการรวบรวมข้อมูล ออกแบบการทำงานของระบบ ออกแบบส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชัน จนถึงขั้น ตอนการออกแบบหน้าจอบริการแอปพลิเคชันสำหรับผู้ใช้งาน ทำให้ผู้พัฒนาทราบถึงวิธีการและขั้นตอนต่างๆในการพัฒนา ซึ่งในการพัฒนาส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟคแบบไบนารี จะใช้โปรแกรมหลักในการพัฒนาคือโปรแกรม Visual studio code ภาษาที่ใช้ในการพัฒนาส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชัน (API) คือภาษา TypeScript ส่วนสำหรับการทำ Unit test จะใช้ Mocha.js, Chai, Sinon.js เป็น Framework ส่วนสำหรับหน้าจอบริการส่วนติดต่อผู้ใช้ (User Interface) จะใช้ React.js เป็น Framework สำหรับการทำให้หน้าเว็บแอปพลิเคชันซึ่งจะมีส่วนที่ประกอบไปด้วยภาษา HTML, CSS และ JavaScript

เมื่อทำการพัฒนาในส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชัน และ เว็บแอปพลิเคชัน ให้สามารถใช้งานได้สมบูรณ์แล้วจะต้องมีการนำไปติดตั้งระบบบนเซิร์ฟเวอร์ โดยใช้เครื่องมือคือ Jenkins ช่วยในการจัดการการติดตั้งระบบบนเซิร์ฟเวอร์

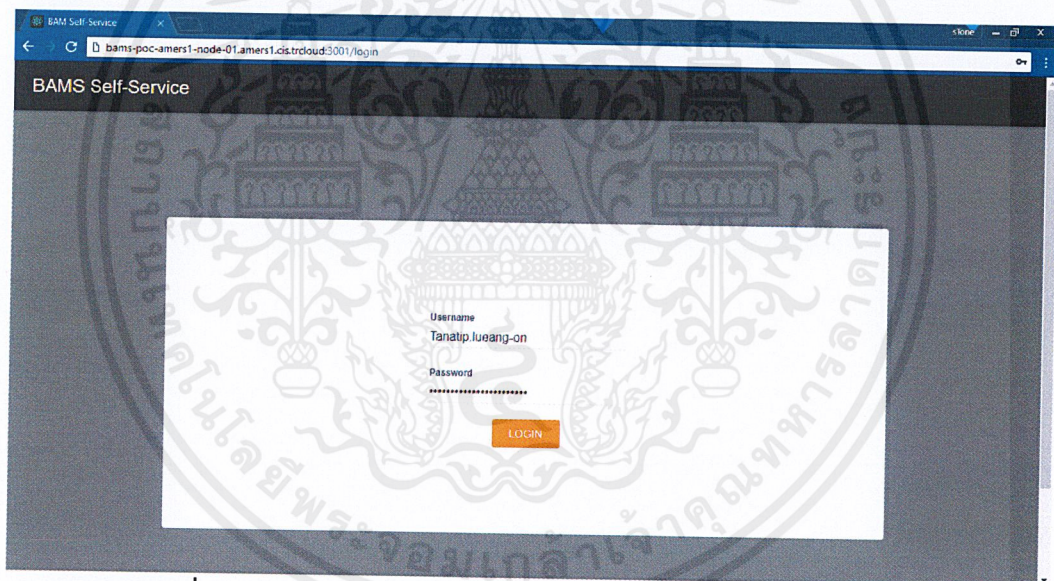
บทที่ 4

ผลการพัฒนาระบบ

เนื้อหาในส่วนนี้จะกล่าวถึงผลจากการพัฒนาเว็บแอปพลิเคชันและส่วนติดต่อเซอร์วิสของแอปพลิเคชันของส่วนการบริการตนเองของระบบการจัดการอาร์ติแฟกต์แบบไบนารี ซึ่งทั้งสองส่วนนั้นจะมีการนำมาใช้ร่วมกัน โดยจะทดสอบการใช้งานจากผลของการนำ Source code ที่ทำการพัฒนาไปติดตั้ง (Deploy) ลงบนเซิร์ฟเวอร์ ซึ่งการสรุปผลการพัฒนาระบบจะถูกแบ่งเป็นหัวข้อ โดยจะแบ่งออกเป็นหัวข้อต่าง ๆ ดังนี้

4.1 หน้าจอแสดงผลส่วนติดต่อผู้ใช้งานเว็บแอปพลิเคชัน

ผู้ใช้งานต้องทำการเข้าสู่ระบบก่อนที่ระบบจะนำผู้ใช้งานเข้าสู่หน้าหลัก ซึ่งในส่วนนี้ถ้าหากกระบวนการจัดการอาร์ติแฟกต์แบบไบนารีนั้นเกิดปัญหาไม่สามารถเชื่อมต่อได้ ก็จะมีการแจ้งเตือนในส่วนนี้



รูปที่ 4.1 หน้าเว็บแอปพลิเคชัน ในส่วนของหน้ายืนยันการเข้าสู่ระบบ

เมื่อผู้ใช้งานเข้าสู่ระบบ ระบบจะนำผู้ใช้งานไปสู่หน้าจอหลัก ซึ่งเมื่อผู้ใช้งานเข้ามาในช่วงแรกจะมีการโหลดข้อมูล ที่จำเป็นที่จะต้องใช้ในการจัดการที่จัดเก็บข้อมูล ได้แก่ Application Short Name, Repository ที่มีข้อมูลอยู่แล้วภายในระบบ โดยในระหว่างที่ระบบกำลังทำการโหลดข้อมูลผู้ใช้งานอยู่นั้นจะไม่สามารถใช้งานการจัดการต่าง ๆ ในระบบได้ ซึ่งระยะเวลาที่ใช้ในการโหลดข้อมูลจะใช้เวลาไม่นาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหน้าต่าง ๆ ของเว็บแอปพลิเคชันจะมีการแบ่งส่วนผู้ใช้เป็นส่วนหลัก โดยทุกส่วนจะปรากฏในทุกหน้าของเว็บแอปพลิเคชัน ซึ่งส่วนนั้นหลักประกอบไปด้วย

- ส่วนบนสุดของเว็บจะมีการแบ่งส่วนที่มุมขวา เพื่อแสดงชื่อของผู้ที่ใช้งานระบบอยู่ในขณะนั้นและปุ่มสำหรับให้ผู้ใช้กดออกจากระบบ ส่วนของมุมด้านซ้ายจะแสดงชื่อของระบบที่ของเว็บ

- ส่วนถัดลงมาจะเป็นส่วนที่แสดงถึงเมนูหลัก โดยจะแบ่งเป็นส่วนย่อยดังนี้

- Repository จัดการที่จัดเก็บจัดเก็บข้อมูล

- Permission จัดการเรื่องการกำหนดสิทธิ์ให้ผู้ใช้

- ส่วนถัดลงมาอีกจะเป็นส่วนของเมนูย่อย โดยจะมีความสัมพันธ์กับส่วนของเมนูหลัก เมนูย่อยจะเปลี่ยนไปตามที่ผู้ใช้เลือกเมนูหลักด้านบน ซึ่งเมนูย่อยจะเป็นส่วนที่ให้ผู้ใช้เลือกว่าจะจัดการทำอะไรกับส่วนของเมนูหลัก ที่ผู้ใช้เลือก



BAMS Self-Service Development Guide --> [Here !!!](#)

รูปที่ 4.2 หน้าจอหลักเว็บแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

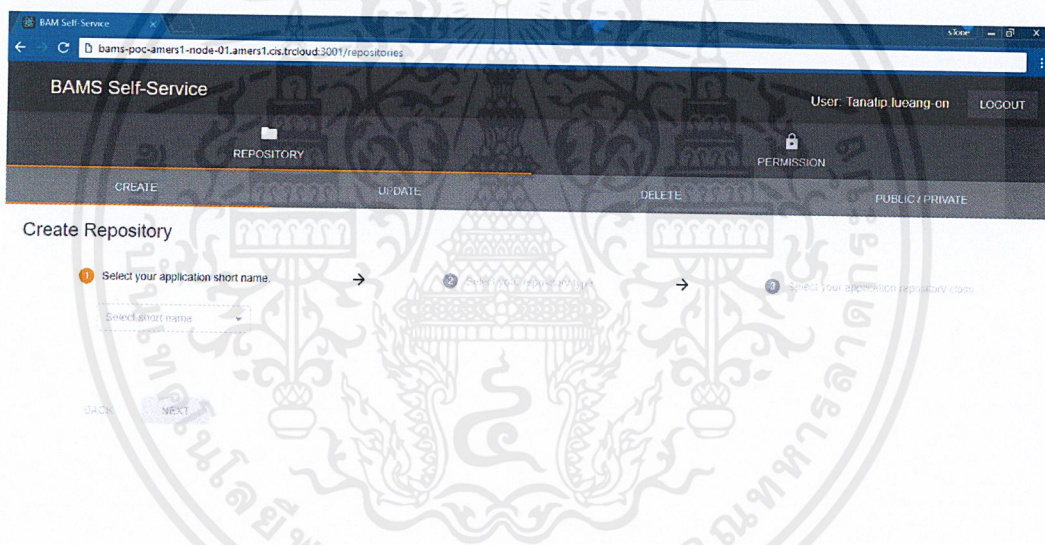
การใช้งานของผู้ใช้ในแต่ละฟังก์ชันของระบบจะกำหนดให้ผู้ใช้กรอกข้อมูลที่ระบบต้องการนำไปใช้ โดยจะกำหนดให้ผู้ใช้ทำตามลำดับขั้นตอน ตามส่วนหน้าจของผู้ใช้งาน ที่ผู้พัฒนาได้ออกแบบไว้ โดยผู้ใช้งานจะต้องกรอกข้อมูลที่ระบบต้องการให้ครบ จึงสามารถกดเข้าไปใช้งานในแต่ละขั้นต่อไปได้

ข้อมูลบางส่วนที่จำเป็นต้องใช้จะมีการดึงข้อมูลในระบบมาให้ผู้ใช้เลือกผู้ใช้สามารถเลือกใช้งานได้เลยโดยไม่ต้องทำการกรอกข้อมูลเข้าไปเอง อีกทั้งผู้ใช้งานยังสามารถกรอกข้อมูลเพื่อค้นหาข้อมูลที่ต้องการได้อีกด้วย

ในส่วนของหน้าจอ เมื่อเลือกเมนูหลักเป็นการจัดการที่จัดเก็บข้อมูล (Repository) จะมีส่วนของเมนูย่อยได้แก่

- Create ส่วนจัดการสร้าง Repository

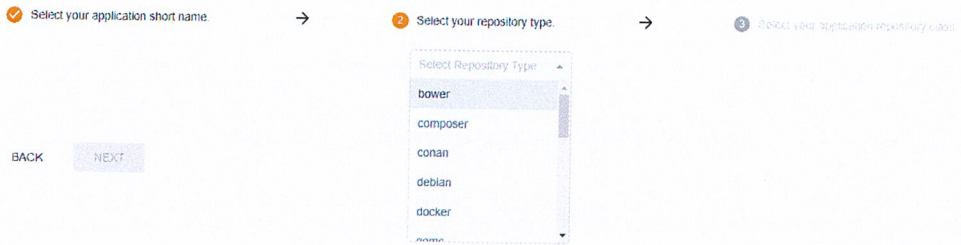
โดยจะมีส่วนข้อมูลที่ต้องการกรอกเพื่อการใช้งานได้สมบูรณ์ได้แก่ Application Short Name, Repository Type, Repository Class โดยข้อมูลทั้งสามจะถูกกำหนดไว้อยู่แล้วจากระบบ



รูปที่ 4.3 ส่วนของหน้าจอ สร้างที่จัดเก็บข้อมูล (Create Repository)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

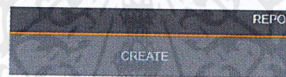
Create Repository



รูปที่ 4.4 ส่วนการเลือกข้อมูล ในขั้นตอนการสร้างที่จัดเก็บข้อมูล

- Update จัดการปรับปรุงให้ Repository ใช้งานได้ โดยการปรับเปลี่ยนการตั้งค่าของ Package Type ของ Repository (ในกรณีที่ใช้งานไม่ได้)
- Delete จัดการลบ Repository

โดยทั้ง Update และ Delete Repository ทั้งสองส่วนนี้นั้นจะมีส่วนข้อมูลที่ต้องกรอกเพื่อการใช้งานได้สมบูรณ์เหมือนกัน ได้แก่ ชื่อที่จัดเก็บข้อมูล ที่มีอยู่แล้วภายในระบบ (Repository Name)



Delete Repository

Select your repository name.

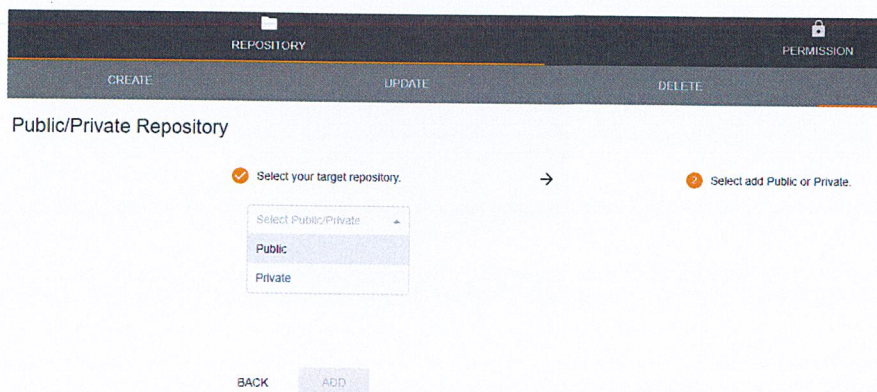
Select repository

bams.bower.local
git.gems.local
bams.composer.local
bams.yum.local
bams.gems.local
bams.rpm.local

รูปที่ 4.5 ส่วนการเลือกข้อมูล ในขั้นตอนการลบที่จัดเก็บข้อมูล

- Public/Private จัดการให้ผู้ใช้สามารถเห็น Repository นั้น ๆ ใน BAMS

โดยส่วนนี้จะมีส่วนข้อมูลที่ต้องกรอกเพื่อการใช้งานได้สมบูรณ์ได้แก่ ชื่อที่จัดเก็บข้อมูล ที่มีอยู่แล้วภายในระบบ (Repository Name) และ การเลือกระหว่าง เปิดเผย(Public) หรือ ปกปิด (Private)

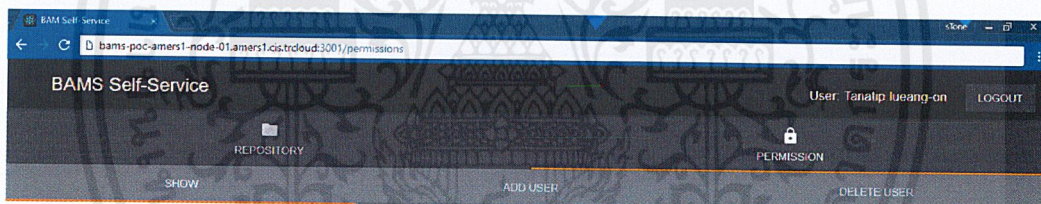


รูปที่ 4.6 ส่วนการเลือกข้อมูล ในขั้นตอนการตั้ง Public/Private ให้ที่จัดเก็บข้อมูล

ในส่วนของหน้าจอ เมื่อเลือกเมนูหลักเป็นการจัดการเรื่องการกำหนดสิทธิให้ผู้ใช้ (Permission) จะมีส่วนของเมนูย่อยได้แก่

- Show แสดงชื่อผู้ใช้ที่มีสิทธิใน Repository

โดยส่วนนี้นั้นจะมีส่วนข้อมูลที่ต้องกรอกเพื่อการใช้งานได้สมบูรณ์ได้แก่ ชื่อที่จัดเก็บข้อมูล ที่มีอยู่แล้วภายในระบบ



Show Permission

Select your repository name.

bams bower local

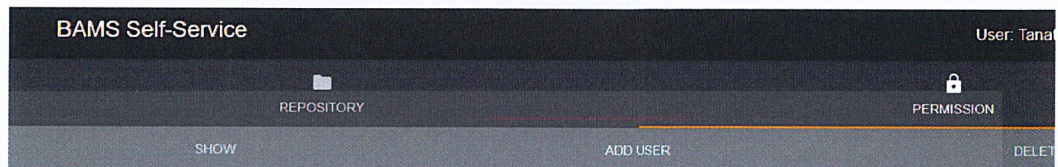
SHOW

รูปที่ 4.7 ส่วนของหน้าจอ แสดงสิทธิของผู้ใช้ (Show)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Add User เพิ่มสิทธิให้ผู้ใช้ใน Repository
- Delete User ลบสิทธิของผู้ใช้ใน Repository

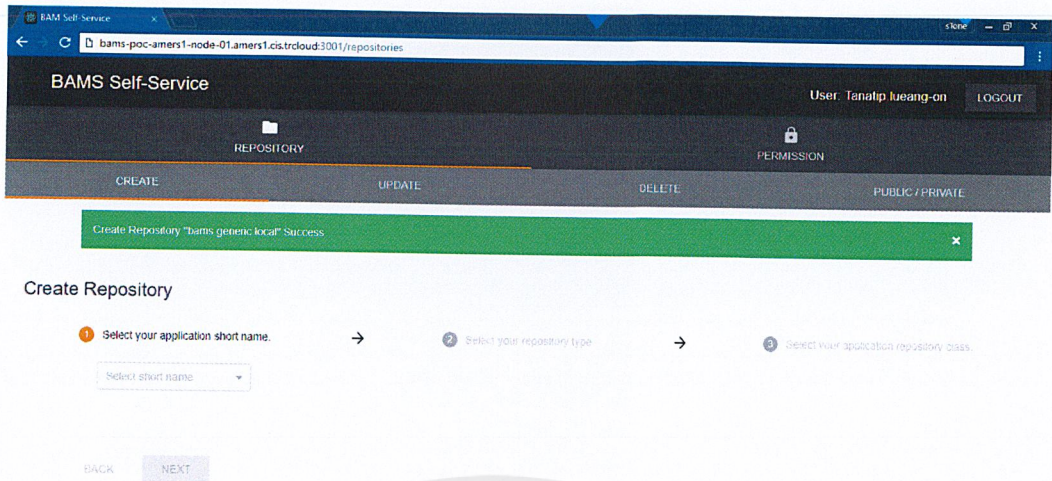
โดยทั้ง Add และ Delete User ทั้งสองส่วนนี้นั้นจะมีส่วนข้อมูลที่ต้องกรอกเพื่อการใช้งานได้สมบูรณ์เหมือนกัน ได้แก่ ชื่อที่จัดเก็บข้อมูล ที่มีอยู่แล้วภายในระบบ และ ชื่อของผู้ใช้งาน ที่มีอยู่ในระบบของภายในองค์กร



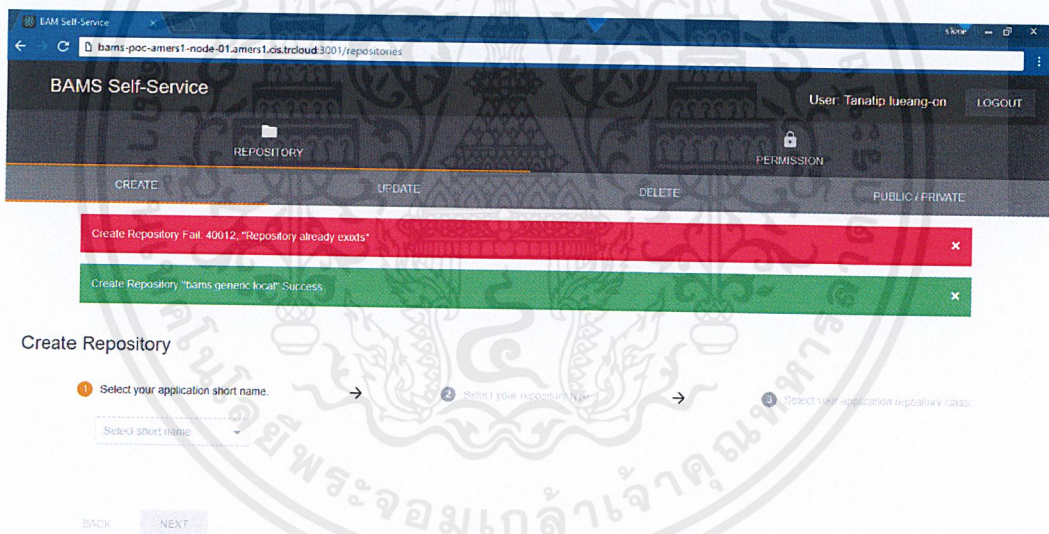
Add user to Permission

รูปที่ 4.8 ส่วนของหน้าจอ เพิ่มสิทธิของผู้ใช้ (Add User)

เมื่อผู้ใช้ ใช้งานส่วนต่าง ๆ ที่มีในระบบ เมื่อระบบประมวลผลเสร็จจะส่งข้อมูลกลับมา แสดงผลทางหน้าจอของผู้ใช้งาน โดยจะแสดงให้เห็นให้ผู้ใช้งานรู้ว่าการทำงาานนั้นสำเร็จหรือไม่ และถ้าไม่สำเร็จ สาเหตุเกิดจากความผิดพลาดของอะไร ซึ่งจะแสดงผลค้างไว้ ถ้าหากผู้ใช้ต้องการปิดสามารถปิดได้โดยการกดที่ปุ่มกากบาท ที่กรอบของสถานะนั้น ๆ โดยถ้ามีการสั่งใช้งานส่วนอื่น สถานะล่าสุดจะแสดงโดยจะไปต่อที่ด้านบนของสถานะก่อนหน้า



รูปที่ 4.9 ส่วนของหน้าจอเมื่อมีการใช้งานฟังก์ชัน เมื่อมีการประมวลผลสำเร็จ



รูปที่ 4.10 ส่วนของหน้าจอเมื่อมีการใช้งานฟังก์ชัน เมื่อมีการประมวลผลไม่สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดสอบเฉพาะส่วนของ BAMS API

เพื่อทดสอบการทำงานเฉพาะส่วน (Unit testing) ของ BAMS API ให้เป็นไปตามความต้องการของผู้ใช้งาน โดยการทดสอบดังกล่าวจะช่วยยืนยันว่าการทำงานของ ระบบนั้นเป็นไปตามที่ได้ตกลงกับผู้ใช้งานก่อนที่จะได้เริ่มต้นพัฒนาระบบ BAMS ในส่วนอื่น ๆ การทดสอบเฉพาะส่วนของ BAMS API นั้นจะมีรายละเอียดดังต่อไปนี้

1) ตารางแสดงผลการทดสอบการใช้งาน LDAP ในระบบ BAMS API แสดงผลการทดสอบที่เกี่ยวข้องกับ LDAP เนื่องจากข้อมูลของผู้ใช้งานนั้นมีส่วนเกี่ยวข้องกับ LDAP ซึ่งเก็บข้อมูลที่เป็นชื่อของผู้ใช้และรหัสผ่าน เพื่อให้ได้ข้อมูลของผู้ใช้งานที่ถูกต้อง แล้วสามารถระบุตัวตนของผู้ใช้งานได้ว่ามีอยู่จริง ซึ่งสามารถอธิบายรายละเอียดได้ดังตารางที่ 4.1

ตารางที่ 4.1 ผลการทดสอบการใช้งาน LDAP ในระบบ BAMS API

Test Case	Result
Should return LDAP data if username and password are valid	PASS
Should have error handling if user not found	PASS
Should call LDAP with OU=Users	PASS
Should call LDAP with OU=Service Accounts	PASS

2) ตารางแสดงผลการทดสอบการเข้าสู่ระบบการใช้งานของ BAMS API เป็นแสดงผลการทดสอบที่เกี่ยวข้องกับ JWT เมื่อผู้ใช้งานทำการเข้าสู่ระบบ ในระบบ BAMS API ทำให้ JWT นั้นทำการสร้าง Token มาเพื่อเหมือนเป็นตัวแทนของการใช้งานของผู้ใช้งานนั้น ๆ และเก็บอยู่ในระบบ ถ้าทำการออกจากระบบ Token ของผู้ใช้นั้น ๆ จะถูกลบทิ้ง ซึ่งสามารถอธิบายรายละเอียดได้ดังตารางที่ 4.2

ตารางที่ 4.2 ผลการทดสอบการเข้าสู่ระบบการใช้งานของ BAMS API

Test Case	Result
Should get token	PASS
Should return error code if username and password is invalid	PASS
Should pass if token is valid	PASS
Should return error code if token is invalid	PASS

3) ตารางแสดงผลการทดสอบระบบการสร้างพื้นที่จัดเก็บข้อมูล เป็นแสดงผลการทดสอบต่างๆของระบบการสร้างพื้นที่จัดเก็บข้อมูล ในการสร้างจะมีการทำงานที่หลากหลายขั้นตอน เช่น การสร้างพื้นที่จัดเก็บข้อมูลในชื่อตามที่ผู้ใช้งานกำหนด การสร้างสิทธิ์การจัดการของผู้ใช้งานหรือเป็นเจ้าของพื้นที่ จัดเก็บข้อมูลนั้น ๆ ก่อนที่ทำการสร้างนั้นจะต้องมีการทดสอบว่าพื้นที่จัดเก็บข้อมูลนั้นไม่เคยถูกสร้างขึ้นมาก่อนหรือเซิร์ฟเวอร์ของ JFrog API ต้องไม่เกิดปัญหา สามารถอธิบายรายละเอียดได้ดังตารางที่ 4.3

ตารางที่ 4.3 ผลการทดสอบระบบการสร้างพื้นที่จัดเก็บข้อมูล

Test Case	Result
API Testing	
Users can create repository	PASS
Users can create permission	PASS
Error Handling	
Check repository already exist	PASS
Users cannot create repository if repository name is wrong	PASS
Users cannot create permission if user data has problem	PASS

4) ตารางแสดงผลการทดสอบระบบการอัปเดตพื้นที่จัดเก็บข้อมูล เป็นแสดงผลการทดสอบต่างๆของระบบการอัปเดตพื้นที่จัดเก็บข้อมูล การอัปเดตพื้นที่จัดเก็บข้อมูลนั้นจะต้องทำการเปลี่ยนแปลงค่าให้ถูกประเภทของพื้นที่จัดเก็บข้อมูล ถ้าอัปเดตผิดประเภทข้อมูลในพื้นที่จัดเก็บข้อมูลอาจเกิดความผิดพลาดได้ ก่อนที่ทำการอัปเดตจะต้องมีการทดสอบว่าพื้นที่จัดเก็บข้อมูลนั้น ต้องมีอยู่จริงหรือเซิร์ฟเวอร์ของ JFrog API ต้องไม่เกิดปัญหา ซึ่งสามารถอธิบายรายละเอียดได้ดังตารางที่ 4.4

ตารางที่ 4.4 ผลการทดสอบระบบการอัปเดตพื้นที่จัดเก็บข้อมูล

Test Case	Result
API Testing	
Check have repository to update	PASS
Users can update repository	PASS
Error Handling	
User is not permitted to update repository	PASS
Users cannot update repository if server has problem	PASS

5) ตารางแสดงผลการทดสอบระบบการลบพื้นที่จัดเก็บข้อมูล เป็นแสดงผลการทดสอบต่างๆของระบบการลบพื้นที่จัดเก็บข้อมูล การลบพื้นที่จัดเก็บข้อมูลนั้นจะต้องทำการระบุชื่อพื้นที่จัดเก็บข้อมูลให้ถูกต้องและผู้ใช้งานที่ทำการลบต้องมีสิทธิ์ในการลบ ไม่ใช่จะสามารถลบพื้นที่จัดเก็บข้อมูลของใครก็ได้ ก่อนที่ทำการลบจะต้องมีการทดสอบว่าพื้นที่จัดเก็บข้อมูลนั้นต้องมีอยู่จริงหรือเซิร์ฟเวอร์ของ JFrog API ต้องไม่เกิดปัญหา ซึ่งสามารถอธิบายรายละเอียดได้ดังตารางที่ 4.5

ตารางที่ 4.5 ผลการทดสอบระบบการลบพื้นที่จัดเก็บข้อมูล

Test Case	Result
API Testing	
Check have repository to delete	PASS
Users can delete repository	PASS
Error Handling	
User is not permitted to delete repository	PASS
Users cannot delete repository if server has problem	PASS

6) ตารางแสดงผลการทดสอบระบบการแสดงรายละเอียดสิทธิ์การจัดการของผู้ใช้งานในพื้นที่จัดเก็บข้อมูล เป็นแสดงผลการทดสอบของระบบว่า เมื่อผู้ใช้งานจะต้องการดูข้อมูลในพื้นที่จัดเก็บข้อมูลนั้น ๆ ข้อมูลที่แสดงรายละเอียดออกมา ต้องถูกต้องและครบถ้วน ก่อนที่ทำการแสดงจะต้องมีการทดสอบว่าพื้นที่จัดเก็บข้อมูลนั้นต้องมีอยู่จริงหรือเซิร์ฟเวอร์ของ JFrog API ต้องไม่เกิดปัญหา ซึ่งสามารถอธิบายรายละเอียดได้ดังตารางที่ 4.6

ตารางที่ 4.6 ผลการทดสอบระบบการแสดงรายละเอียดสิทธิ์การจัดการของผู้ใช้งานในพื้นที่จัดเก็บข้อมูล

Test Case	Result
API Testing	
Check have repository to show detail	PASS
Users can get detail from repository	PASS
Error Handling	
Users cannot get detail repository if server has problem	PASS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) ตารางแสดงผลการทดสอบระบบการเพิ่มสิทธิ์การจัดการของผู้ใช้งาน เป็นแสดงผลการทดสอบต่าง ๆ ของระบบเพิ่มสิทธิ์การจัดการ การเพิ่มสิทธิ์นั้นจะต้องทำการระบุชื่อผู้ใช้งานใหม่ที่จะเพิ่มให้ถูกต้องและผู้ใช้งานที่ทำการเพิ่มต้องมีสิทธิ์ในการเพิ่ม ไม่ใช่ว่าจะสามารถเพิ่มสิทธิ์ผู้ใช้งานใหม่ให้ใครก็ได้ ก่อนที่ทำการเพิ่มสิทธิ์จะต้องมีการทดสอบว่าผู้ใช้งานใหม่นั้นต้องมีอยู่จริงโดยทดสอบจาก LDAP และชื่อที่ทำการเพิ่มนั้นต้องไม่ซ้ำ หรือเซิร์ฟเวอร์ของ JFrog API ต้องไม่เกิดปัญหา ซึ่งสามารถอธิบายรายละเอียดได้ดังตารางที่ 4.7

ตารางที่ 4.7 ผลการทดสอบระบบการเพิ่มสิทธิ์การจัดการของผู้ใช้งาน

Test Case	Result
API Testing	
Check LDAP user data to add permission	PASS
Users can add user to permission	PASS
Error Handling	
Check add user exist in permission	PASS
User is not permitted to add permission	PASS
Users cannot add permission if server has problem	PASS

8) ตารางแสดงผลการทดสอบระบบการลบสิทธิ์การจัดการของผู้ใช้งาน เป็นแสดงผลการทดสอบต่าง ๆ ของระบบลบสิทธิ์การจัดการ การลบสิทธิ์นั้นจะต้องทำการระบุชื่อผู้ใช้งานเดิมที่จะทำการลบให้ถูกต้องและผู้ใช้งานที่ทำการลบต้องมีสิทธิ์ในการลบ ไม่ใช่ว่าจะสามารถลบสิทธิ์ผู้ใช้งานเดิมของใครก็ได้ ก่อนที่ทำการลบสิทธิ์จะต้องมีการทดสอบว่าผู้ใช้งานเดิมที่ทำการลบต้องมีอยู่จริงในสิทธิ์การจัดการนั้น ๆ และชื่อที่ทำการลบต้องห้ามเป็นชื่อตนเอง หรือเซิร์ฟเวอร์ของ JFrog API ต้องไม่เกิดปัญหา ซึ่งสามารถอธิบายรายละเอียดได้ดังตารางที่ 4.8

ตารางที่ 4.8 ผลการทดสอบระบบการลบสิทธิ์การจัดการของผู้ใช้งาน

Test Case	Result
API Testing	
Check have user permission to remove user	PASS
Users can remove user from permission	PASS
Error Handling	
User is not permitted to remove permission	PASS
User cannot remove yourself	PASS
Users cannot remove permission if server has problem	PASS

9) ตารางแสดงผลการทดสอบระบบการเปิดเผยหรือปกปิดพื้นที่จัดเก็บข้อมูลของผู้ใช้งาน เป็นแสดงผลการทดสอบว่าผู้ใช้งานที่ต้องการจะทำการปกปิดหรือเปิดเผยพื้นที่จัดเก็บข้อมูลของตัวเองให้กับผู้ใช้งานคนอื่นนั้น สามารถทำให้มีการเปลี่ยนแปลงไปมากันได้ ในการเปลี่ยนแปลงพื้นที่จัดเก็บข้อมูลเซิร์ฟเวอร์ของ JFrog API ต้องไม่เกิดปัญหา ซึ่งสามารถอธิบายรายละเอียดได้ดังตารางที่ 4.9

ตารางที่ 4.9 ผลการทดสอบระบบการเปิดเผยหรือปกปิดพื้นที่จัดเก็บข้อมูลของผู้ใช้งาน

Test Case	Result
Users can change private repository to public repository	PASS
Users can change public repository to private repository	PASS
Users cannot change repository if server has problem	PASS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการพัฒนาระบบและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากการพัฒนา เว็บแอปพลิเคชันและส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชันของ ระบบการจัดการอาร์ติแฟกต์แบบไบนารี (Binary Artifact Management System: BAMS) สามารถสรุปผลการดำเนินงานได้ดังนี้

- เว็บแอปพลิเคชันและส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชัน ทั้งสองส่วนสามารถทำงานร่วมกันได้ เมื่อนำระบบไปติดตั้งบนเซิร์ฟเวอร์
- เว็บแอปพลิเคชันและส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชัน สามารถติดต่อเข้าไปจัดการกับระบบการจัดการอาร์ติแฟกต์แบบไบนารี ในเวอร์ชันใหม่ได้
- เว็บแอปพลิเคชันและส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชัน ถูกออกแบบมาให้สอดคล้องกับความต้องการ ของการปรับปรุงรูปแบบของ โครงสร้างภายในของระบบที่ต่างไป
- การใช้งานของระบบเป็นไปตามรูปแบบและข้อตกลงภายในองค์กร

5.2 ข้อจำกัดของระบบ

จากความสามารถของโปรแกรมระบบที่ได้กล่าวมาข้างต้นนั้นมีความสามารถบางอย่างที่ผู้พัฒนาเห็นว่ายังมีข้อจำกัดดังนี้

- ระบบสามารถนำไปติดตั้งและใช้งานได้ในระบบเซิร์ฟเวอร์ขององค์กร แต่ระบบนั้นยังมีปัญหาในเรื่องของความเสถียร เนื่องจากมีปัญหาที่เกิดจากการเชื่อมต่อกับ LDAP ทำให้ บางช่วงเวลาส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชันที่ติดตั้งอยู่ภายในระบบเซิร์ฟเวอร์ขององค์กรหยุดการทำงานลง ซึ่งสามารถแก้ไขได้โดยการเริ่มการทำงาน ส่วนติดต่อเซิร์ฟเวอร์ของแอปพลิเคชันที่นำไปติดตั้งภายในระบบเซิร์ฟเวอร์ใหม่

5.3 ข้อเสนอแนะ

เนื่องจากระบบนี้เป็นระบบที่มีการพัฒนามาจากเวอร์ชันเก่าที่มีมาก่อน ทางผู้พัฒนามีข้อเสนอแนะดังนี้

- ในส่วนของการจัดการที่จัดเก็บข้อมูล การจัดการประเภทของแพ็คเกจในรูปแบบต่าง ๆ นั้นมีเงื่อนไขของการจัดการที่แตกต่างกันออกไป ดังนั้นจึงต้องมีการศึกษาเพิ่มเพื่อหาข้อจำกัดในแต่ละประเภทของแพ็คเกจที่ใช้ภายในองค์กรเพิ่มเติม
- สามารถเพิ่มการปรับปรุงในส่วนของ Source code ให้มีความสะดวกและเข้าใจง่าย ในการพัฒนาและตรวจสอบปัญหาเมื่อนำไปใช้งานจริง

เอกสารอ้างอิง

- Puttasak Tantisuttivet. 2559. ทำความรู้จักกับ Agile และ Scrum [Online].
เข้าถึงได้จาก : <https://brandinside.asia/agile-and-scrum-for-new-business/>
- Wikipedia. 2560. Binary repository manager [Online].
เข้าถึงได้จาก : https://en.wikipedia.org/wiki/Binary_repository_manager
- Saixiii. 2560. RESTful คืออะไร REST คือ การสื่อสารแลกเปลี่ยนข้อมูลผ่าน webservice [Online].
เข้าถึงได้จาก : <https://saixiii.com/what-is-restful>
- Nuttavut Thongjor. 2559. ออกแบบ REST API ยังไงดี? [Online].
เข้าถึงได้จาก : <https://www.babelcoder.com/blog/posts/rest-api-design>
- Yoav Landman. 2560. JFrog Artifactory User Guide [Online].
เข้าถึงได้จาก : <https://www.jfrog.com/confluence/>
- TheHoiStory. 2560. ลองเล่นๆกับ TypeScript [Online].
เข้าถึงได้จาก : <https://medium.com/@thehoistoryลองเล่นๆกับ-typescript-983d810d0986>
- Ae Eiei. 2559. มาทดสอบ API ด้วย PostMan กันเถอะ [Online].
เข้าถึงได้จาก : <https://medium.com/@nest12345/มาทดสอบ-api-ด้วย-postman-กันเถอะ-c750a09c86ce>
- Designil. 2559. React คืออะไร? ไขข้อสงสัยสำหรับมือใหม่ [Online].
เข้าถึงได้จาก : <https://www.designil.com/react-คืออะไร.html>
- Babelcoder. 2559. สอนสร้าง Isomorphic Application ด้วย React.js และ Redux [Online].
เข้าถึงได้จาก : <https://www.babelcoder.com/blog/series/react-redux-isomorphic-series>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

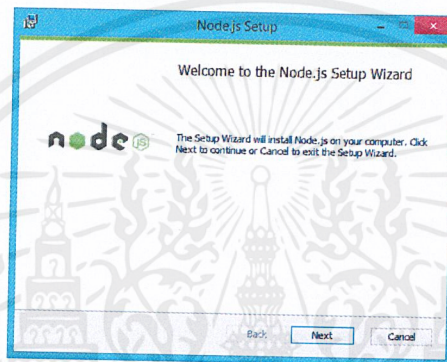
ภาคผนวก ก

การเตรียมเครื่องมือเพื่อใช้ในการพัฒนาระบบ

ก.1 ติดตั้ง Node.js

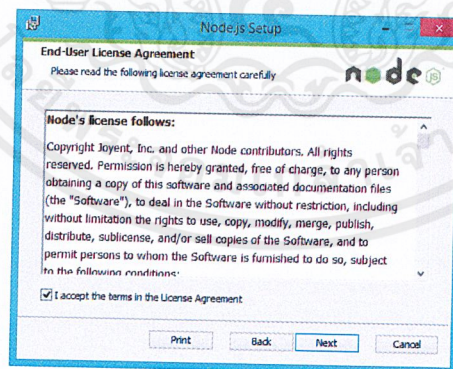
1) ดาวน์โหลดตัวติดตั้ง Node.js จาก <https://nodejs.org/en/download/> โดย Node.js ที่ใช้งานนั้นจะเป็น Node.js LTS Version

2) เมื่อดาวน์โหลดตัวติดตั้งเสร็จสิ้นแล้วให้ทำการเปิดตัวติดตั้ง Node.js เมื่อมีหน้าต่างขึ้นตามรูปด้านล่างแล้วให้ทำการกด Next



รูปที่ ก.1 หน้าSetup ของ Node.js

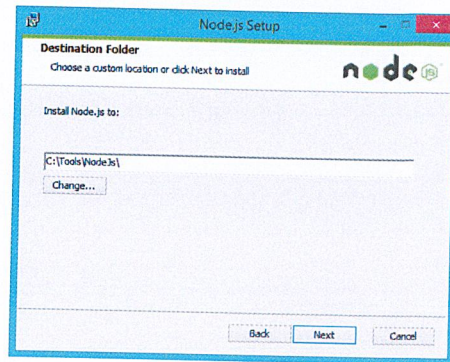
3) เลือกรายการ I accept the terms in the License Agreement และ คลิกที่ปุ่ม Next



รูปที่ ก.2 หน้ายืนยันข้อตกลง ของ Node.js

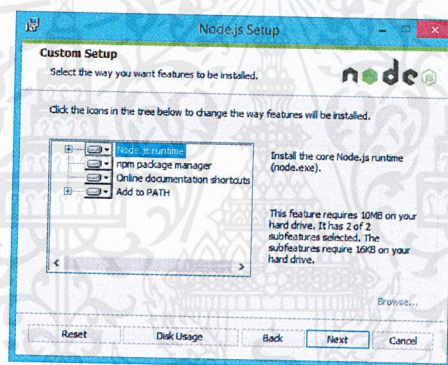
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) เลือกตำแหน่งติดตั้ง Node.js และ คลิกที่ปุ่ม Next



รูปที่ ก.3 หน้าเลือกตำแหน่งติดตั้ง ของ Node.js

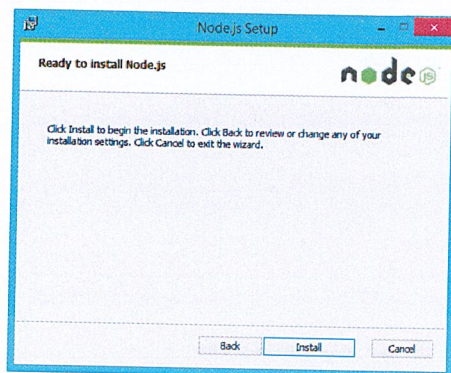
5) แสดงรายการโปรแกรมที่ได้หลังติดตั้ง Node.js สำเร็จ และ คลิกที่ปุ่ม Next



รูปที่ ก.4 หน้าแสดงรายการโปรแกรม ของ Node.js

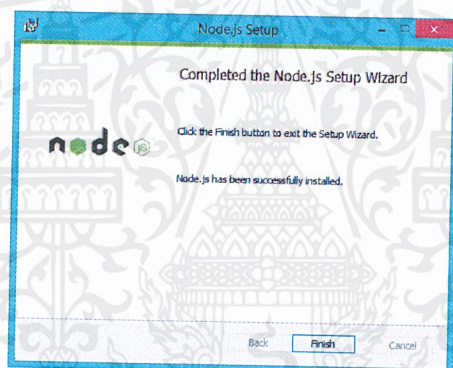
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) คลิกที่ปุ่ม Install และ รอระบบติดตั้ง Node.js



รูปที่ ก.5 หน้าติดตั้ง ของ Node.js

7) ระบบติดตั้ง Node.js สำเร็จแล้ว สุดท้ายกดปุ่ม Finish เพื่อปิดหน้าจอ

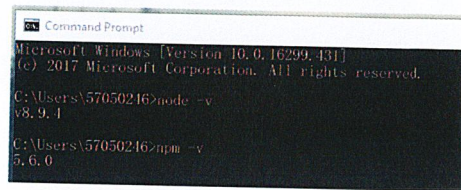


รูปที่ ก.6 หน้าสิ้นสุดการติดตั้ง ของ Node.js

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) ทดสอบการติดตั้ง Node.js เพื่อใช้งานคำสั่ง npm โดย เปิดโปรแกรม Command-Lite Tools บนระบบปฏิบัติการ Windows

9) พิมพ์คำสั่ง `node -v` หรือ `npm -v` หากสำเร็จ จะแสดงผลดังรูป



```

Command Prompt
Microsoft Windows [Version 10.0.16299.431]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\57050216>node -v
v8.9.1

C:\Users\57050216>npm -v
5.6.0
  
```

รูปที่ ก.7 ทดสอบการติดตั้ง ของ Node.js



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้