

รายงานการปฏิบัติงานสหกิจศึกษา

เรื่อง

การพัฒนาซอฟต์แวร์ไมโครเซอร์วิสเชิงพาณิชย์ ณ บริษัท เอสซีบี เทคเอกซ์

Commercial Microservices Software Developement at SCB TechX

ณ บริษัท เอสซีบี เทคเอ็กซ์ จำกัด มหาชน

เลขที่ 19 ถนนรัชดาภิเษก แขวงจตุจักร เขตจตุจักร กรุงเทพมหานคร

โดย

นายธนภัทร สมสิทธิ์ รหัส 640610639

รายงานนี้

เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

ปีการศึกษา 2567

หัวข้อโครงการ : การพัฒนาซอฟต์แวร์ไมโครเซอร์วิสเชิงพาณิชย์ ณ บริษัท เอสซีบี เทคเอกซ์
: Commercial Microservices Software Development at SCB TechX
โดย : นายธนภัทร สมสิทธิ์ รหัส 640610639
ภาควิชา : วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา : ผศ.ดร. ปฏิเวธ วุฒิสารวัฒนา
ปริญญา : วิศวกรรมศาสตรบัณฑิต
สาขา : วิศวกรรมคอมพิวเตอร์
ปีการศึกษา : 2567

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

..... หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์
(รศ.ดร. สันติ พิทักษ์กีนูกร)

คณะกรรมการสอบโครงการ

..... ประธานกรรมการ
(ผศ.ดร. ปฏิเวธ วุฒิสารวัฒนา)

..... กรรมการ
(ผศ.ดร. กานต์ ปทานุคม)

..... กรรมการ
(ผศ.ดร. นวदनย์ คุณเลิศกิจ)

หัวข้อโครงการ : การพัฒนาซอฟต์แวร์ไมโครเซอร์วิสเชิงพาณิชย์ ณ บริษัท เอสซีบี เทคเอกซ์
: Commercial Microservices Software Development at SCB TechX
โดย : นายธนภัทร สมสิทธิ์ รหัส 640610639
ภาควิชา : วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา : ผศ.ดร. ปฏิเวธ วุฒิสารวัฒนา
ปริญญา : วิศวกรรมศาสตรบัณฑิต
สาขา : วิศวกรรมคอมพิวเตอร์
ปีการศึกษา : 2567

บทคัดย่อ

ในรายงานนี้ ข้าพเจ้าได้สรุปประสบการณ์การปฏิบัติงานสหกิจศึกษา ณ บริษัท เอสซีบี เทคเอกซ์ จำกัด ซึ่งเป็นส่วนหนึ่งของหลักสูตรการศึกษาในระดับปริญญาตรี สาขาวิชาคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ การปฏิบัติงานมุ่งเน้นไปที่การพัฒนาซอฟต์แวร์ในโครงการ xPlatform และการสนับสนุนงานในทีม Fast Easy ในฐานะวิศวกรซอฟต์แวร์

นอกจากนี้ ข้าพเจ้ายังได้กล่าวถึงร่างขอบเขตงานกระบวนวิชาสหกิจศึกษา (TOR) ซึ่งเป็นแนวทางสำคัญในการปฏิบัติงาน โดยมีวัตถุประสงค์เพื่อให้การฝึกงานเป็นไปตามมาตรฐานที่มหาวิทยาลัยกำหนด

รายงานนี้จะนำเสนอรายละเอียดเกี่ยวกับประสบการณ์การทำงาน งานที่ได้รับมอบหมาย และผลลัพธ์ที่ได้จากการปฏิบัติงานในโครงการต่าง ๆ รวมถึงบทบาทของข้าพเจ้าในทีมพัฒนา ซึ่งจะช่วยให้ผู้ที่สนใจในสายงานนี้เข้าใจแนวทางและการดำเนินงานขององค์กรได้ดียิ่งขึ้น

กิตติกรรมประกาศ

การที่ข้าพเจ้าได้มาปฏิบัติงานสหกิจศึกษา ณ บริษัท เอสซีบี เทคเอ็กซ์ จำกัด ตั้งแต่วันที่ 4 เดือน มิถุนายน พ.ศ. 2567 ถึง วันที่ 25 เดือน ตุลาคม พ.ศ. 2567 ส่งผลให้ข้าพเจ้าได้รับ ความรู้และประสบการณ์ต่าง ๆ ที่มีค่ามากมาย สำหรับรายงานวิชาสหกิจศึกษาฉบับนี้ สำเร็จลงได้ด้วยดี จากความร่วมมือและสนับสนุนจากหลายฝ่าย ดังนี้

1. ศุภกร เนตรสุวรรณ ตำแหน่งวิศวกรซอฟต์แวร์ (นักศึกษาฝึกงาน)
2. สิริวิมล สุขสุคนธ์ ตำแหน่งวิศวกรซอฟต์แวร์
- 3.

และบุคคลท่านอื่น ๆ ที่ไม่ได้กล่าวนามทุกท่านที่ได้ให้คำแนะนำช่วยเหลือในการจัดทำรายงาน ข้าพเจ้าใคร่ขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่าน ที่มีส่วนร่วมในการให้ข้อมูล เป็นที่ปรึกษาใน การทำรายงานฉบับนี้จนเสร็จสมบูรณ์ ตลอดจนให้การดูแลและให้ความเข้าใจเกี่ยวกับชีวิตของการทำงาน จริง ข้าพเจ้าขอขอบคุณ ไว้ ณ ที่นี้

นายธนภัทร สมสิทธิ์
25 ตุลาคม 2567

สารบัญ

บทคัดย่อ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	จ
สารบัญตาราง	ฉ
1 บทนำ	1
1.1 ความสำคัญและที่มาของโครงการ	1
1.2 วัตถุประสงค์ของการศึกษา	1
2 ข้อมูลทั่วไปของบริษัท	2
2.1 สินค้าและบริการของบริษัท	2
2.2 งบแสดงฐานะการเงิน	3
2.3 งบกำไรขาดทุน	3
3 รายละเอียดเกี่ยวกับงาน	5
3.1 ร่างขอบเขตงานกระบวนวิชาสหกิจศึกษา (TOR)	5
3.2 ขั้นตอนการดำเนินงาน	5
3.3 ตำแหน่งที่ได้รับมอบหมาย	6
3.4 งานที่ได้รับมอบหมาย	6
3.4.1 พีเจอร์ xPlatform Change Runbook	6
3.4.2 การปรับปรุง Configurations การเข้าถึงฐานข้อมูลกับทีม Fast Easy	15
3.4.3 งานการบริหารข้อมูลผู้ใช้งานและสิทธิการใช้งานด้วย Keycloak	16
3.4.4 พีเจอร์ xPlatform Custom Library	18
3.4.5 พีเจอร์ xPlatform Documentation	18
3.5 โครงสร้างพื้นฐานและสถาปัตยกรรมของซอฟต์แวร์	19
3.5.1 โครงสร้างพื้นฐานและสถาปัตยกรรมของซอฟต์แวร์โปรเจค xPlatform	19
4 ผลของการทำงาน	22
4.1 สรุปผลการทำงานตามร่างขอบเขตงาน	22
4.2 สัดส่วนการทำงาน	22
4.3 ช่วงระยะเวลาการทำงาน	23
5 สภาพแวดล้อมของการทำงานในองค์กร	24
5.1 ก่อนที่จะปฏิบัติงานสหกิจศึกษา	24
5.2 เงินเดือนและสวัสดิการ	25
5.3 วัฒนธรรมองค์กร	26
5.3.1 กิจกรรมและการอบรม	26
5.3.2 ความคิดเห็นส่วนตัว	27
5.4 ข้อเสนอแนะในการสมัครงาน	27
5.5 ข้อเสนอจากบริษัท	27

สารบัญรูป

3.1	แผนผัง Change Runbook ที่ใช้จริง	7
3.2	แบบจำลองความสัมพันธ์ระหว่างข้อมูลของพีเจอร์ Change Runbook แบบย่อ	8
3.3	ตัวอย่างกราฟแสดงความสัมพันธ์ของ Change Runbook	9
3.4	ตัวอย่างการเปลี่ยนแปลงเวลาของ Activity ที่ 1	10
3.5	ตัวอย่างการเปลี่ยนแปลงเวลาของ Activity ที่ 2	11
3.6	ตัวอย่างการเปลี่ยนแปลงเวลาของ Activity ที่ 3	12
3.7	ตัวอย่างเปลี่ยนแปลง Activity แบบ Bypass	13
3.8	ตัวอย่างการลบ Activity	14
3.9	ตัวอย่างของ Change Runbook จากการ Export	15
3.10	Shared Database Configuration	16
3.11	แบบจำลองความสัมพันธ์ระหว่างข้อมูลภายใน Keycloak แบบย่อ	17
3.12	แบบจำลองการ Authorization กับ Authentication ผ่าน Keycloak	17
4.1	แผนภูมิรูปวงกลมแสดงสัดส่วนของ Story Points	23

สารบัญตาราง

2.1	โครงสร้างฐานะการเงินปี 2021 ถึงปี 2023	3
2.2	อัตราการเปลี่ยนแปลงโครงสร้างฐานะการเงินปี 2021 ถึงปี 2023	3
2.3	โครงสร้างงบกำไรขาดทุนปี 2021 ถึงปี 2023	3
2.4	อัตราการเปลี่ยนแปลงโครงสร้างงบกำไรขาดทุนปี 2021 ถึงปี 2023	4
4.1	ตารางแสดงสัดส่วนของ Story Points	22
4.2	ตารางแผนปฏิบัติงานสหกิจศึกษา	23
5.1	ตารางแสดงไหม้ไลน์การทำงาน	25

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของโครงการ

1.2 วัตถุประสงค์ของการศึกษา

1. ได้ประยุกต์ใช้ทักษะการพัฒนาซอฟต์แวร์ที่ในศึกษาในมหาวิทยาลัยและแหล่งอื่น ๆ บนโครงการที่อยู่ในอุตสาหกรรมซอฟต์แวร์จริง
2. ได้ประสบการณ์ในการพัฒนาเว็บแอปพลิเคชันในอุตสาหกรรม
3. ได้ศึกษาระบบการทำงานของการทำงานของผู้พัฒนาซอฟต์แวร์ รวมไปถึงเทคโนโลยีและอุปกรณ์ต่าง ๆ ที่ช่วยในการพัฒนาซอฟต์แวร์

บทที่ 2

ข้อมูลทั่วไปของบริษัท

2.1 สินค้าและบริการของบริษัท

บริษัท เอสซีบี เทคเอกซ์ มีความเชี่ยวชาญในการพัฒนานวัตกรรมและเทคโนโลยีที่สามารถตอบสนองความต้องการของลูกค้าในด้านการบริการอย่างครบวงจร โดยนำเสนอบริการที่หลากหลายซึ่งครอบคลุมตั้งแต่การให้คำปรึกษาไปจนถึงการพัฒนาโซลูชันทางเทคโนโลยี รวมถึงการวิเคราะห์ความต้องการของระบบ การออกแบบ การพัฒนาซอฟต์แวร์จนถึงการใช้งานจริง นอกจากนี้ บริษัทยังให้บริการด้านโครงสร้างพื้นฐานเทคโนโลยีและการประมวลผลบนระบบคลาวด์เพื่อสนับสนุนการดำเนินงานในยุคดิจิทัล

เพื่อสนับสนุนการตัดสินใจเชิงธุรกิจ บริษัทได้พัฒนาบริการจัดการข้อมูลที่สามารถสร้างข้อมูลเชิงลึกให้กับลูกค้า พร้อมทั้งมีบริการด้านความปลอดภัยทางไซเบอร์เพื่อปกป้องข้อมูลและระบบการดำเนินงานในสภาพแวดล้อมดิจิทัล

1. **eKYC (Electronic Know Your Customer)** เป็นซอฟต์แวร์ที่พัฒนาขึ้นตามพระราชบัญญัติป้องกันและปราบปรามการฟอกเงิน ซึ่งกำหนดให้ธุรกิจที่เกี่ยวข้องกับการเงินและการลงทุนต้องดำเนินการระบบ KYC (Know Your Customer) ก่อนทำธุรกรรม ในอดีต การยืนยันตัวตนผู้ใช้บริการจะต้องใช้วิธีการกรอกเอกสาร ซึ่งอาจทำให้กระบวนการช้าและซับซ้อน เพื่อเพิ่มความสะดวกและความรวดเร็วในการยืนยันตัวตน บริษัทจึงได้สร้างซอฟต์แวร์นี้ขึ้น โดยนำเทคโนโลยีการสแกนใบหน้าและบัตรประชาชนมาใช้ ซึ่งช่วยให้กระบวนการยืนยันตัวตนสามารถดำเนินการได้อย่างรวดเร็ว
2. **xPlatform** ได้นำแนวทางปฏิบัติที่มีประสิทธิภาพใน DevOps มาประยุกต์ใช้ในการออกแบบแพลตฟอร์มอัตโนมัติ ในรูปแบบ Web Application ซึ่งช่วยให้ลดภาระการทำงานของทั้งทีมพัฒนาและทีมปฏิบัติการ โดยที่แพลตฟอร์มนี้มีฟีเจอร์ที่รองรับทุกขั้นตอนของวงจรซอฟต์แวร์ ตั้งแต่การพัฒนา การทดสอบ การปล่อยซอฟต์แวร์ ไปจนถึงการบำรุงรักษา การเฝ้าระวัง และการเพิ่มประสิทธิภาพ นอกจากนี้ยังช่วยสนับสนุนการทำงานร่วมกันแบบ Agile บนแพลตฟอร์มเดียว ซึ่งฟีเจอร์เหล่านี้ยังช่วยให้ PO/PM สามารถบริหารทีมและควบคุมงบประมาณของโครงการเพื่อลดค่าใช้จ่ายที่ไม่จำเป็น
3. **TechX Data Platform** เป็นแพลตฟอร์มที่ออกแบบมาเพื่อให้การจัดการข้อมูลเป็นเรื่องง่ายและครบวงจร โดยครอบคลุมทุกขั้นตอน ตั้งแต่การนำเข้าข้อมูล การจัดเก็บ การจัดการ การวิเคราะห์ จนถึงการรักษาความปลอดภัยของข้อมูล นอกจากนี้ยังมีบริการวิเคราะห์ข้อมูลขั้นสูงด้วยเทคโนโลยี Machine Learning ที่สามารถปรับแต่งให้เหมาะสมกับธุรกิจได้ทุกรูปแบบ ไม่ว่าจะเป็น Startup ธุรกิจ SME หรือองค์กรขนาดใหญ่ ซึ่งมีความต้องการด้านข้อมูลที่แตกต่างกันไป
4. บริษัทมีบริการให้คำปรึกษาและพัฒนาโซลูชันครบวงจร เพื่อตอบสนองความต้องการของลูกค้าและผู้มีส่วนเกี่ยวข้อง โดยใช้เฟรมเวิร์กกระบวนการคิดเชิงออกแบบ (Design Thinking) นอกจากนี้ บริษัทยังมีทีมวิศวกรซอฟต์แวร์และนักออกแบบประสบการณ์ผู้ใช้ (UX Designer) ที่มีความเชี่ยวชาญในการพัฒนาโซลูชันให้กลายเป็นผลิตภัณฑ์จริง
5. บริการโซลูชันด้านคลาวด์ที่เน้นความยืดหยุ่นและประสิทธิภาพในการจัดการโครงสร้างพื้นฐานทางไอที รวมถึงการย้ายข้อมูลและการรักษาความปลอดภัยบนคลาวด์ บริการเหล่านี้ครอบคลุมตั้งแต่การออกแบบสถาปัตยกรรมระบบ การจัดการทรัพยากรไอที จนถึงการเฝ้าระวังและปรับปรุงประสิทธิภาพของ

ระบบ ที่ทีมงานยังมีความเชี่ยวชาญในการบริหารจัดการระบบคลาวด์หลากหลายแพลตฟอร์ม (multi-cloud) และใช้กระบวนการที่เน้นความปลอดภัยในทุกขั้นตอน

2.2 งบแสดงฐานะการเงิน

ปี	จำนวน (ล้านบาท)		
	สินทรัพย์	หนี้สิน	ส่วนผู้ถือหุ้น
2021	1742	1241	501
2022	2351	821	1529
2023	1954	587	1367

ตารางที่ 2.1: โครงสร้างฐานะการเงินปี 2021 ถึงปี 2023

ปี	สินทรัพย์	หนี้สิน	ส่วนผู้ถือหุ้น
2021	0	0	0
2022	0.35	-0.34	2.05
2023	-0.11	-0.29	-0.17

ตารางที่ 2.2: อัตราการเปลี่ยนแปลงโครงสร้างฐานะการเงินปี 2021 ถึงปี 2023

บริษัท เอสซีพี เทคเอกซ์ ซึ่งเป็นบริษัทในเครือของธนาคารไทยพาณิชย์ มีสินทรัพย์ที่สูงมากสำหรับบริษัทใหม่ โดยในปีแรก (2021) บริษัทมีสินทรัพย์รวม 1742 ล้านบาท จากนั้นสินทรัพย์เพิ่มขึ้นเป็น 2351 ล้านบาท ในปี 2022 ในปีนี้ ส่วนของผู้ถือหุ้นเพิ่มขึ้นกว่า 1000 ล้านบาท จาก 501 ล้านบาทในปี 2021 เป็น 1529 ล้านบาท แม้ในปี 2023 สินทรัพย์จะลดลงเล็กน้อยมาอยู่ที่ 1954 ล้านบาท แต่ส่วนของผู้ถือหุ้นยังคงสูงอยู่ที่ 1367 ล้านบาท

ในด้านหนี้สิน บริษัทเริ่มต้นด้วยหนี้สิน 1241 ล้านบาทในปี 2021 ทำให้อัตราส่วนหนี้สินต่อทุนอยู่ที่ 2.43 เท่า จากนั้นบริษัทปรับลดระดับหนี้สินอย่างต่อเนื่อง ในปี 2022 หนี้สินลดลงเหลือ 821 ล้านบาท และในปี 2023 ลดลงเหลือ 587 ล้านบาท ส่งผลให้อัตราส่วนหนี้สินต่อทุนในปี 2023 อยู่ที่ 0.49 เท่า

2.3 งบกำไรขาดทุน

ปี	จำนวน (ล้านบาท)			
	กำไรสุทธิ	กำไรก่อนภาษี	รายจ่ายรวม	รายได้รวม
2021	350	438	1158	1596
2022	672	840	2329	3169
2023	246	308	1978	2289

ตารางที่ 2.3: โครงสร้างงบกำไรขาดทุนปี 2021 ถึงปี 2023

ปี	กำไรสุทธิ	กำไรก่อนภาษี	รายจ่ายรวม	รายได้รวม
2021	0	0	0	0
2022	0.92	0.92	1.01	0.99
2023	-0.63	-0.63	-0.15	-0.28

ตารางที่ 2.4: อัตราการเปลี่ยนแปลงโครงสร้างงบกำไรขาดทุนปี 2021 ถึงปี 2023

ในปีแรกของบริษัท มีผลกำไรสุทธิอยู่ที่ 350 ล้านบาท ขณะที่รายได้รวมอยู่ที่ 1596 ล้านบาท ซึ่งแสดงให้เห็นว่าบริษัทมีอัตรากำไรสุทธิที่ 0.219 เท่า เมื่อเปรียบเทียบกับรายได้รวมในปีถัดมาในปี 2022 อัตราการเพิ่มของรายได้รวมและรายจ่ายรวมของบริษัทได้เพิ่มขึ้นเท่าตัว โดยรายจ่ายรวมอยู่ที่ 2329 ล้านบาท และรายได้รวมอยู่ที่ 3169 ล้านบาท ซึ่งอัตรากำไรสุทธิในปีนั้นอยู่ที่ 0.212 เท่า แสดงให้เห็นว่าบริษัทยังคงรักษาอัตรากำไรสุทธิในระดับที่ใกล้เคียงกับปีแรก

ในปีถัดมา มีการลดอัตรารายจ่ายรวมและรายได้รวม โดยอัตราการลดของรายได้รวมนั้นเกือบสองเท่าของอัตราการลดรายจ่ายรวม ทำให้รายจ่ายรวมลดลงเหลือ 1978 ล้านบาท และรายได้รวมอยู่ที่ 2289 ล้านบาท ผลลัพธ์นี้ทำให้อัตรากำไรสุทธิของบริษัทตกลงเหลือเพียง 0.107 เท่า ซึ่งแสดงให้เห็นว่าบริษัทมีการเติบโตในปีแรกและปีที่สอง แต่ในปีที่สามกลับมีแนวโน้มการลดลงของอัตรากำไรสุทธิ อาจบ่งบอกถึงความท้าทายที่บริษัทเผชิญหรือเกิดจากการตัดสินใจลดขนาดหนี้สิน

บทที่ 3

รายละเอียดเกี่ยวกับงาน

บทนี้จะกล่าวถึงงานที่ได้รับมอบหมายในระหว่างการปฏิบัติงานสหกิจศึกษา ซึ่งประกอบด้วยร่างขอบเขตงาน (TOR) ที่กำหนดขึ้นเพื่อใช้เป็นแนวทางในการทำงาน รวมถึงคำอธิบายรายละเอียดของแต่ละงานและหน้าที่ความรับผิดชอบที่ได้รับจากทีมงานต่าง ๆ

3.1 ร่างขอบเขตงานกระบวนการวิชาสหกิจศึกษา (TOR)

ในช่วงระยะเวลาการปฏิบัติงานสหกิจ ได้มีโอกาสพัฒนา 2 โปรเจกต์หลักร่วมกับทีม xPlatform และทีม Fast Easy โดยดำเนินงานตามขั้นตอนการพัฒนาซอฟต์แวร์แบบเอจายล์ ซึ่งการทำงานถูกแบ่งออกเป็นงานย่อย ๆ เรียกว่า “การ์ด” โดยแต่ละการ์ดจะมีการกำหนดคะแนนความยากของงาน (Story Points) โดยมีการตั้งขอบเขตขั้นต่ำก่อนที่จะสำเร็จการปฏิบัติงานสหกิจศึกษาไว้ที่ 60 คะแนน

การพัฒนาโปรเจกต์ดังกล่าวต้องอาศัยทักษะหลากหลายด้าน เช่น การออกแบบฐานข้อมูล การวางแผนและพัฒนา API สำหรับผู้ใช้งาน การเรียกใช้ API จากซอฟต์แวร์อื่น ๆ การพัฒนาซอฟต์แวร์ การทดสอบซอฟต์แวร์ รวมถึงการใช้งานเครื่องมือสนับสนุนการพัฒนาซอฟต์แวร์ เช่น แพลตฟอร์ม Version Control แพลตฟอร์มจัดการโปรเจกต์ และสภาพแวดล้อมการพัฒนา (IDE)

3.2 ขั้นตอนการดำเนินงาน

โครงการทั้งหมดจะใช้ขั้นตอนการพัฒนาซอฟต์แวร์แบบเอจายล์ ซึ่งจะแบ่งช่วงการทำงานออกเป็นหลายวัฏจักร ซึ่งมีชื่อเรียกว่า Spring Cycle โดยที่แต่ละวัฏจักรนั้นจะมีช่วงเวลาการทำงานอยู่ 2 อาทิตย์ โดยมีวัตถุประสงค์เพื่อให้การพัฒนาซอฟต์แวร์มีความยืดหยุ่นต่อความต้องการของผู้มีส่วนเกี่ยวข้อง (Requirements) ที่มักจะมีการเปลี่ยนแปลงอยู่ตลอด ต่างจากขั้นตอนการพัฒนาแบบดั้งเดิมที่มีการวางแผนโครงการเพียงครั้งเดียว โดยในหนึ่งวัฏจักรนั้นจะสามารถแบ่งขั้นตอนการทำงานได้ออกเป็น 4 ขั้นตอนย่อย ได้แก่

1. ระยะเวลาการวางแผน หรือ ระยะเวลาออกแบบโปรแกรมตาม Requirements โดยขั้นตอนนี้จะมีการคาดการณ์ความยากของแต่ละชิ้นงานเรียกว่า Story Points ผู้พัฒนาที่ได้รับชิ้นส่วนของงานใด ก็จะสามารถ Story Points ของงานทั้งหมด ซึ่งสำหรับหลาย ๆ ทีมในบริษัท เอสซีบี เทคโนโลยี Story Points นี้จะถูกกำหนดให้มีค่าเป็นเลขจำนวนฟีโบนัชชี และประมาณค่าดังกล่าวเป็นจำนวนของวันที่ใช้ในการพัฒนาชิ้นส่วนงานนั้น ๆ
2. ขั้นตอนการพัฒนาโปรแกรมตามชิ้นส่วนของงานที่ได้รับมอบหมาย
3. การทดสอบโปรแกรม
4. การปรับปรุงแก้ไขโปรแกรม

โดยจะใช้ Jira และ Confluence เป็นแพลตฟอร์มจัดการโครงการ ตรวจสอบขั้นตอนและรายงานสถานะขั้นตอนของชิ้นส่วนโครงการ และใช้ GitLab เป็นแพลตฟอร์มควบคุมเวอร์ชัน

นอกจากนี้ ซอฟต์แวร์ที่ได้พัฒนานั้นจะมีสถาปัตยกรรม Microservices ทั้งหมด กล่าวคือ ตัวของซอฟต์แวร์นั้นจะประกอบไปด้วยหลาย ๆ ชิ้นส่วนเรียกว่า Services ที่จะมีหน้าที่ประมวลผลส่วนหนึ่งของฟังก์ชัน

การทำงานของหมดของตัวซอฟต์แวร์ และจะต้องถูกออกแบบให้มีความเชื่อมโยงกับ Services อื่น ๆ ให้น้อยที่สุดตามเท่าที่จำเป็น ทั้งนี้ เนื่องจากแต่ละ Service ของตัวซอฟต์แวร์สามารถรองรับการใช้งานแต่ละฟังก์ชันได้ต่างกัน สำหรับกรณีที่บางฟังก์ชันในซอฟต์แวร์ถูกใช้งานมากกว่าฟังก์ชันอื่น ๆ ฝั่งของการ Deployment จะสามารถเพิ่มจำนวนของ Services นั้น ๆ ได้

3.3 ตำแหน่งที่ได้รับมอบหมาย

ระหว่างการฝึกปฏิบัติงานสหกิจศึกษา ได้รับมอบหมายให้ทำงานในตำแหน่ง Software Engineer ภายใต้สังกัด CTO โดยร่วมงานกับทีม xPlatform Developer ซึ่งมีหน้าที่พัฒนาเว็บไซต์ที่เป็นส่วนต่อระหว่างผู้ใช้กับการ Integration กับ Services ต่าง ๆ ในโปรเจก xPlatform นอกจากนี้ยังมีทีมอื่น ๆ ที่มีบทบาทสำคัญในการบริหารและพัฒนาโปรเจก เช่น ทีม xPlatform Platform Engineer ทีม QA และทีม DevOps

3.4 งานที่ได้รับมอบหมาย

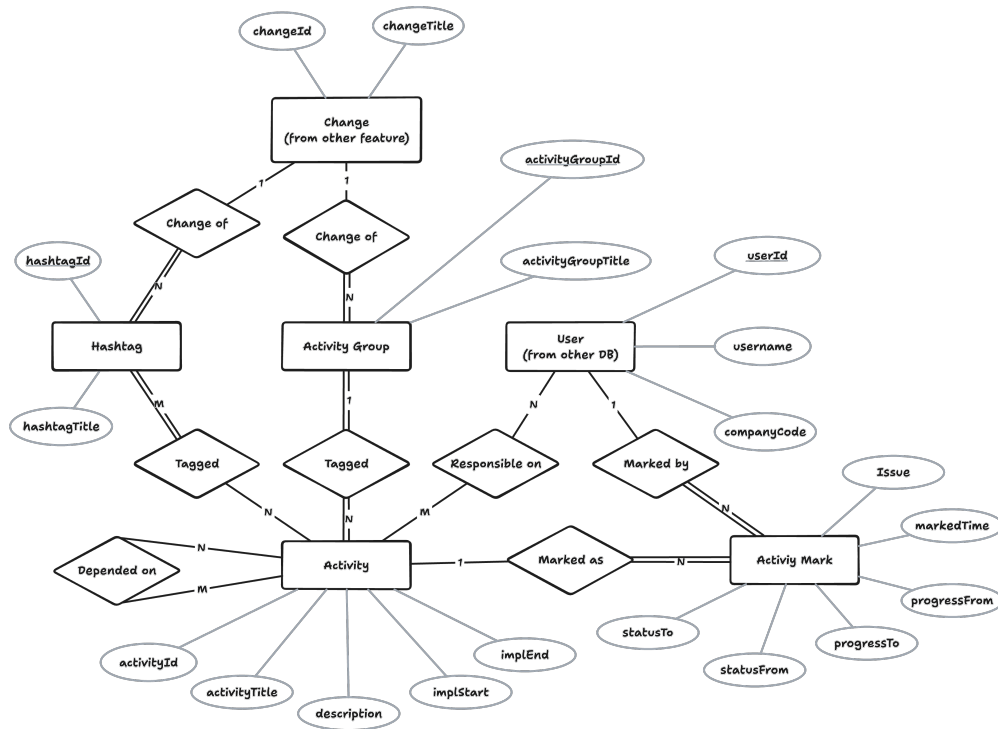
งานที่ได้รับมอบหมายในระหว่างการปฏิบัติงานสหกิจศึกษานั้น ส่วนมากเป็นงานที่ทำร่วมกับทีม xPlatform แต่ก็มีงานที่ร่วมกับทีมของ Fast Easy อยู่เป็นบางครั้ง ซึ่งเป็นทีมที่พัฒนาซอฟต์แวร์ที่เป็น product ของบริษัท เอสซีบี เทคเอ็กซ์ ทั้งคู่ งานที่ได้รับมอบหมายมีความหลากหลายและครอบคลุมหลายด้านของการพัฒนาซอฟต์แวร์ ตั้งแต่การออกแบบ การพัฒนา การทดสอบ ไปจนถึงการปรับปรุงประสิทธิภาพของระบบ

3.4.1 ฟีเจอร์ xPlatform Change Runbook

ในขั้นตอนของการพัฒนาซอฟต์แวร์นั้น หากผู้พัฒนาจำเป็นต้องปรับเปลี่ยน configuration ของระบบต่าง ๆ ที่เกี่ยวข้อง ผู้พัฒนาไม่มีสิทธิในการปรับเปลี่ยนโดยตรง เช่น ขั้นตอนการ deploy แต่ละส่วนของระบบ การปรับเปลี่ยนสิทธิการเข้าถึงข้อมูล หรือการจัดการความปลอดภัยของระบบ การเปลี่ยนแปลงเหล่านี้จะต้องดำเนินการผ่านการแจ้งให้พนักงานในแผนกอื่น ๆ ที่มีสิทธิในการเข้าถึง เช่น DevOps หรือแผนกผู้บริหารฐานข้อมูล

ในกรณีนี้ Runbook จะเป็นเอกสารที่มีรายละเอียดของขั้นตอนการดำเนินงาน เพื่อให้ผู้ที่มีสิทธิในการเข้าถึงสามารถดำเนินการเปลี่ยนแปลงเหล่านี้ได้อย่างถูกต้องตามขั้นตอนที่กำหนด เช่น การ deploy ซอฟต์แวร์ใหม่หรือการอัปเดต configuration การมี Runbook ช่วยลดความเสี่ยงจากความผิดพลาดในการทำงาน

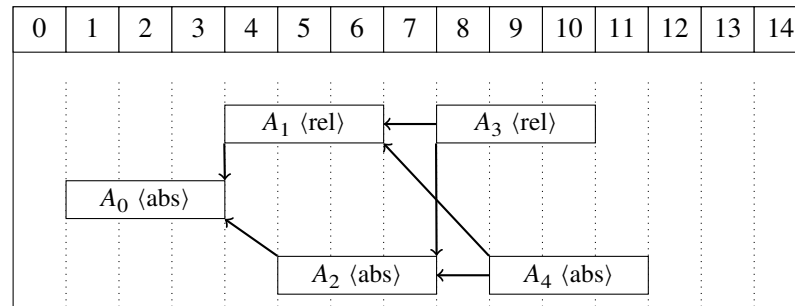
การทำงานแต่ละขั้นตอนดังกล่าวจะมีชื่อเรียกว่า Activity รายงานขั้นตอนการทำงานที่จะแจ้งแผนกต่าง ๆ นั้นจะมีชื่อว่า Runbook โดยที่ขั้นตอนดังกล่าวนี้โดยปกติจะทำการรวมกับการเปลี่ยนแปลงเวอร์ชันของซอฟต์แวร์ที่จะเรียกว่า Change หรือที่มักจกเป็นที่รู้จักกันว่า Release โดยปกติแล้ว การขั้นตอนการเขียน Runbook นั้นจะลงเองด้วยมือ ซึ่งเป็นเรื่องที่ค่อนข้างเสียเวลามาก และสามารถเกิดข้อผิดพลาดขณะการเขียนได้ง่าย เราจึงได้สร้างฟีเจอร์ Change Runbook เพื่อช่วยให้นักพัฒนาซอฟต์แวร์สามารถรายงานขั้นตอนการทำงานได้สะดวกขึ้น



รูปที่ 3.2: แบบจำลองความสัมพันธ์ระหว่างข้อมูลของฟีเจอร์ Change Runbook แบบย่อ

2. ผู้ที่จะสามารถเปลี่ยนแปลงข้อมูล (Update) หรือลบ (Delete) Activity ได้ จะเป็นผู้ที่สร้าง Activity นั้น ๆ หรือ Product Manager กับ Product Owner (PO & PM)
3. ในแต่ละ Activity จะสามารถเปลี่ยนแปลงสถานะการทำงานหรือความก้าวหน้าของงานได้ ซึ่งการทำเช่นนี้จะมีเรียกว่าการ Marking โดยที่ผู้ที่จะสามารถ Mark ได้จะเป็นเพียงแค่ผู้ที่ทำหน้าที่รับผิดชอบ (Responsible people) หรือผู้ใช้ที่มีหน้าที่เป็น PO & PM ซึ่งผู้ Mark จะสามารถระบุโน้ต หรือว่า Issue ที่เกี่ยวข้องกับการเปลี่ยนแปลงนั้นได้
4. ในแต่ละ Activity จะสามารถแบ่งวิธีการหนดเวลาได้เป็น 2 รูปแบบ ได้แก่ Absolute กับ Relative โดยที่
 - (a) Absolute Activity คือ Activity ที่ในขณะที่ถูก Create หรือ Update นั้น ผู้ใช้งานจะต้องระบุเวลาเริ่มต้นและเวลาจบของงาน โดยที่เวลาเริ่มต้นของ Activity ดังกล่าวดังต้องมาหลังเวลาจบของทุก ๆ Dependency (Constraint)
 - (b) Relative Activity คือ Activity ที่ในขณะที่ถูก Create หรือ Update นั้น ผู้ใช้จะระบุเพียงแค่ระยะเวลาการทำงานของ Activity นั้น ๆ โดยที่เวลาเริ่มต้นกับเวลาจบนั้นจะขึ้นอยู่กับ Constraint กล่าวคือ เวลาเริ่มต้นของ Activity นั้น ๆ จะเท่ากับ Constraint เสมอ ซึ่งหมายความว่าทุก ๆ Relative Activity จะจำเป็นต้องมีอย่างน้อย 1 Dependency
5. ในการ Update Activity นั้น อาจเกิดกรณีที่ Activity นั้นเป็น Dependency ของ Activity ตัวอื่น ๆ ได้ ซึ่งเวลาการทำงานของ Activity ดังกล่าวนั้นจะจำเป็นต้องเปลี่ยนไปอัตโนมัติตามกฎหมายดังนี้

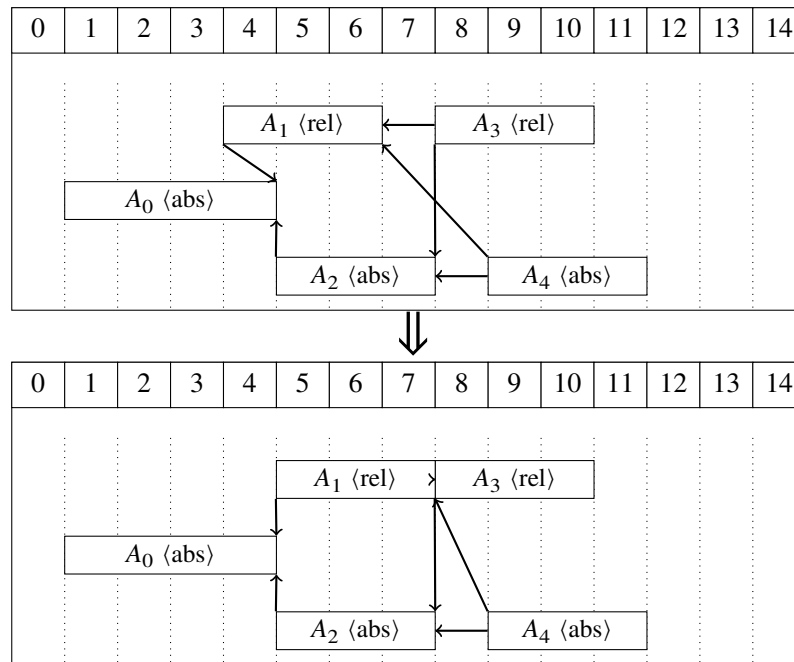
- (a) หาก Constraint ของ Absolute Activity ถูกเลื่อนไปอยู่หลัง Activity นั้น เวลาในการทำงานของ Activity จะถูกเลื่อนตามไปอยู่หลัง Constraint โดยผู้ใช้สามารถเลือกที่จะ Bypass Absolute Activity เพื่อไม่ให้เวลาการทำงานของ Activity เปลี่ยนแปลง แต่จะทำให้ความเป็น Dependency ของ Activities ที่เสร็จหลังก่อนที่ Absolute Activity จะเริ่ม นั้นถูกยกเลิก
- (b) Relative Activity จะต้องเปลี่ยนเวลาใหม่ถ้าหาก Constraint เปลี่ยน



รูปที่ 3.3: ตัวอย่างกราฟแสดงความสัมพันธ์ของ Change Runbook

จากตัวอย่าง Runbook ด้านบนจะแบ่งได้ออกเป็น 5 Activities โดยที่ A_1, A_2 จะมี A_0 เป็น Dependency ส่วน A_3, A_4 จะมี A_1, A_2 เป็น Dependency แสดงว่า Constraint ของ A_1, A_2 จะเป็นเวลาจบของ A_0 ส่วน A_3, A_4 จะมี Constraint เป็นเวลาจบของ A_2 เนื่องจากว่า A_1, A_3 เป็น Activity แบบ Relative ทั้งสอง Activities จึงถูกทำงานตาม Constraint เสมอ แต่ว่า A_2, A_4 ซึ่งเป็น Absolute จะทำงานช่วงไหนก็ได้หลังจาก Time Constraint

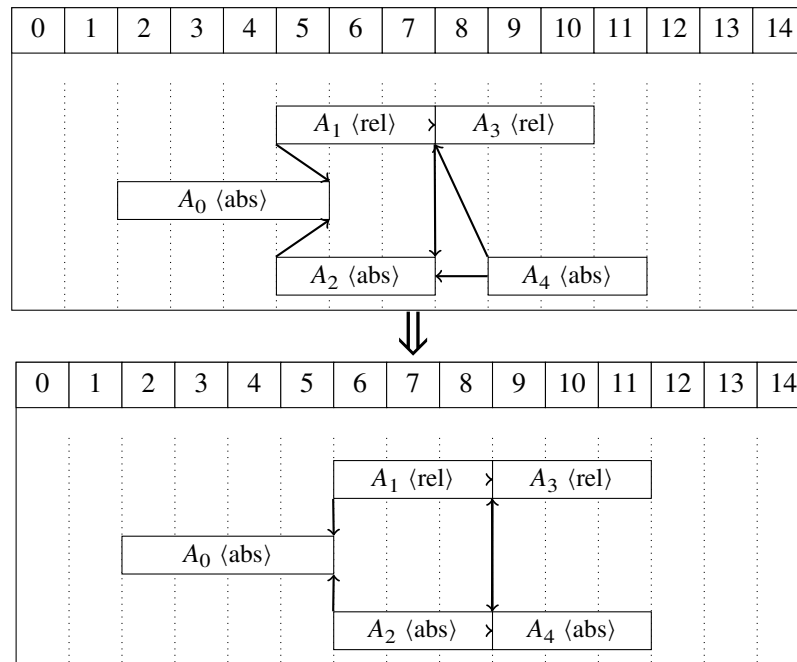
ตัวอย่างที่ 1: A_0 ถูก Update ให้จบเวลาที่ 4



รูปที่ 3.4: ตัวอย่างการเปลี่ยนแปลงเวลาของ Activity ที่ 1

จะสังเกตได้ว่าเนื่องจากเวลาจบของ A_0 นั้นได้ถูกเปลี่ยนแปลงไป Constraint ของ A_1, A_2 ก็เปลี่ยนแปลงไปตามเช่นกัน แต่เนื่องจาก A_2 เป็น Absolute Activity ที่ยังอยู่หลัง Constraint จึงเลยไม่ได้ถูกกำหนดเวลาใหม่ ในขณะที่ A_1 จะต้องเปลี่ยนแปลงเวลาตาม Constraint ใหม่เนื่องจากเป็น Relative Activity

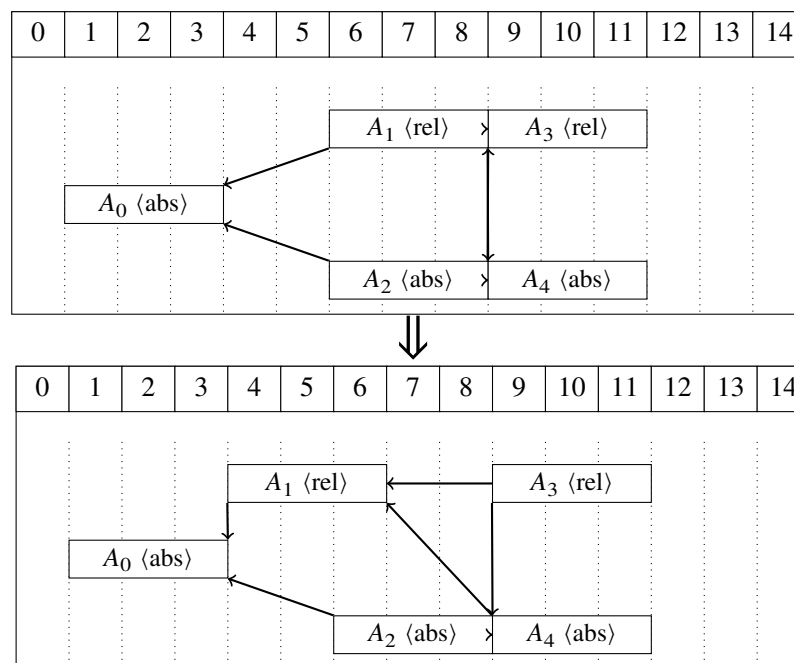
ตัวอย่างที่ 2: A_0 ถูก Update ไปอีก 1 ช่วงเวลา



รูปที่ 3.5: ตัวอย่างการเปลี่ยนแปลงเวลาของ Activity ที่ 2

ต่างจากตัวอย่างก่อนหน้านี้ **Constraint** ของ A_2 ถูกเปลี่ยนแปลงให้ทำงานหลัง A_2 จะเริ่มต้น จึงเลยต้องเปลี่ยนเวลาการทำงานตามด้วย นอกจากนี้ **Constraint** ของ A_3, A_4 ก็เปลี่ยนแปลงไปเช่นเดียวกัน ส่งผลให้เวลาทำงานของ A_3 ถูกเปลี่ยนแปลงไป

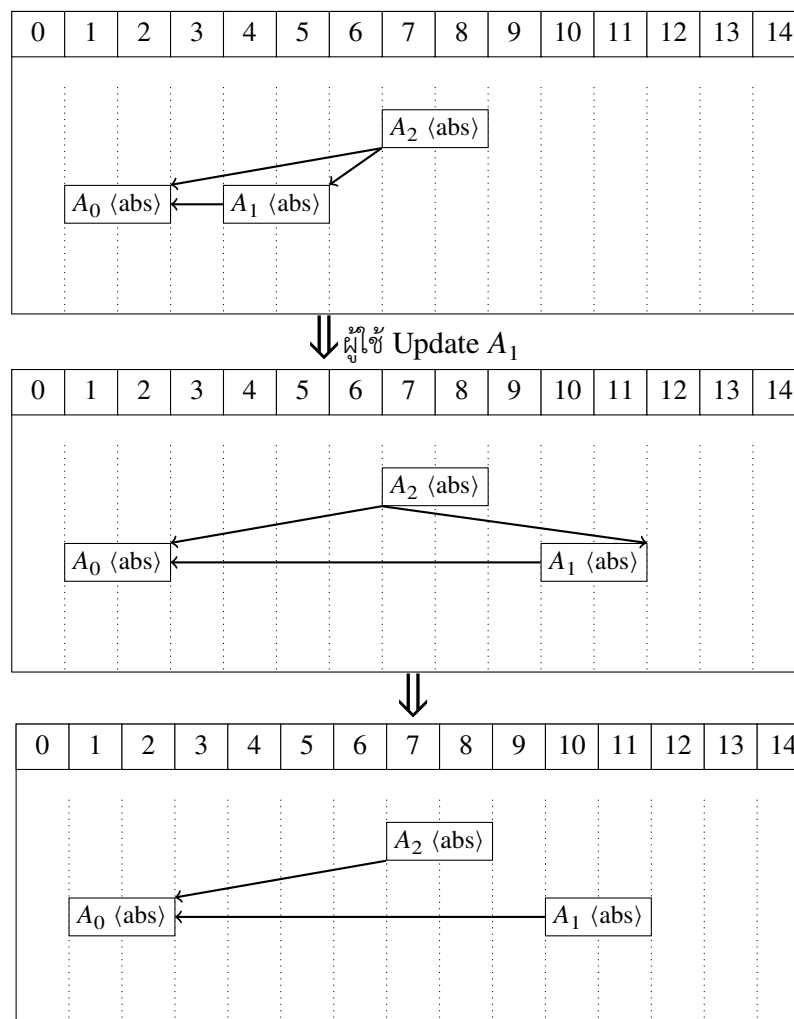
ตัวอย่างที่ 3: A_0 ถูก Update เปลี่ยนไปเป็นเวลาการทำงานเดิม



รูปที่ 3.6: ตัวอย่างการเปลี่ยนแปลงเวลาของ Activity ที่ 3

จะเห็นได้ว่า Constraint ของ A_1, A_2 ได้ถูกเปลี่ยนแปลงไป แต่ว่ามีแค่ A_1 เท่านั้นที่ถูกเปลี่ยนแปลงเวลาการทำงานตาม เนื่องจาก A_1 เป็น Relative Activity ที่ต้องทำงานหลังจาก Constraint ทันที ในขณะที่ A_2 ซึ่งเป็น Absolute Activity ทำงานตอนไหนก็ได้หลังจาก Constraint

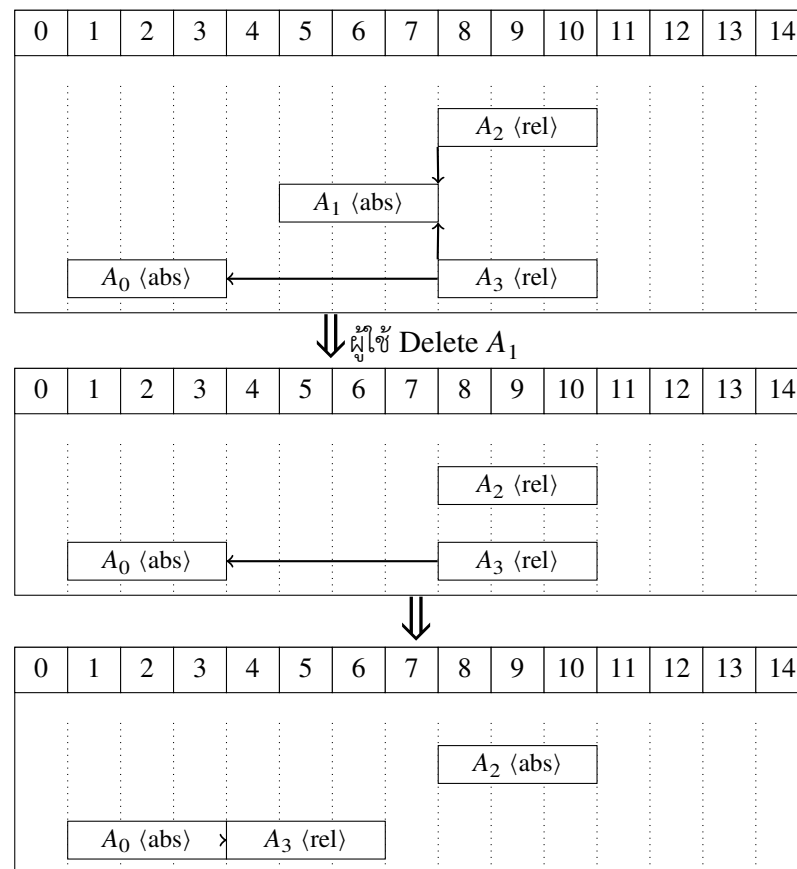
ตัวอย่างที่ 4: การ Bypass Activity



รูปที่ 3.7: ตัวอย่างเปลี่ยนแปลง Activity แบบ Bypass

จากตัวอย่างข้างต้นจะเห็นได้ว่า A_1 เป็นตัวที่จะทำงานหลังจาก A_2 ทั้ง ๆ ที่เป็น Dependency ของ A_2 แต่ว่าเนื่องจากผู้ใช้งานที่ตัดสินใจให้ Bypass เวลาการทำงานของ A_2 จึงไม่เปลี่ยนแปลงตาม A_1 และ A_1 ก็ออกจากการเป็น Dependency ของ A_2 เพื่อไม่ให้ผิดหลักเกณฑ์ของ Constraint

6. ในการ Delete Activity นั้น ถ้าหากตัวที่กำลังถูกลบอยู่เป็น Dependency ตัวเดียวของ Relative Activity Activity นั้นจะถูกโปรโมทให้เป็น Absolute Activity แทน

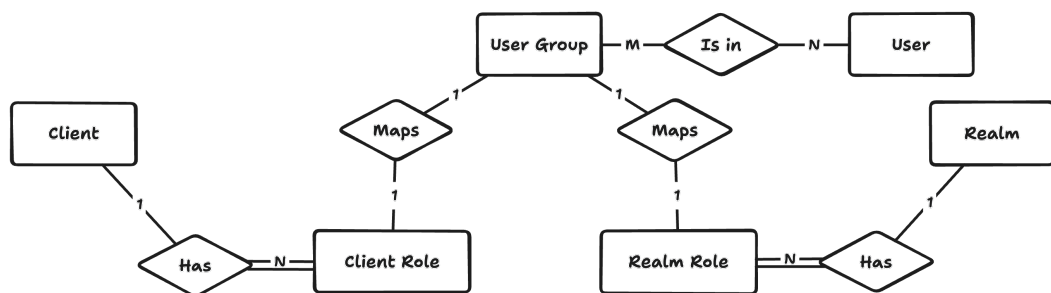


รูปที่ 3.8: ตัวอย่างการลบ Activity

- จากตัวอย่างข้างต้นจะเห็นได้ว่า A₂ นั้นได้ถูกโปรโมทเป็น Absolute Activity หลังจากที่ A₁ ถูก Delete เนื่องจากว่า Relative Activity จำเป็นต้องมี Dependency ส่วนในกรณีของ A₃ ก็จะถูกเลื่อนเวลาการทำงาน เนื่องจาก Constraint ได้ถูกเปลี่ยนแปลง
7. ผู้ใช้สามารถดึงข้อมูล (Import) จากไฟล์ประเภท CSV ได้ โดยที่ผู้ใช้จะสามารถทำการ Map header ให้เป็นแต่ละ Field ของ Activity
 8. ผู้ใช้สามารถดึงข้อมูลของ Issues จากเว็บไซต์ Jira ในการสร้าง Activity ได้ โดยที่ผู้ใช้งานจะสามารถคัดเลือกข้อมูล (Query) ได้อยู่สองวิธี
 - (a) การ Query แบบ Basic: ผู้ใช้จะต้องระบุ โค้ดของโปรเจกต์ Label ของ Issue และ ประเภทของ Issue
 - (b) การ Query ด้วย Jira Query Language (JQL) ซึ่งเป็นภาษาที่ช่วยในการค้นหาข้อมูลใด ๆ ก็ตามภายในเว็บไซต์ของ Jira

โดยที่วิธีการดึงข้อมูลนี้จะแตกต่างกันตาม Type ของ Field ที่กำลังถูกดึง นอกจากนี้ Activity ที่ถูกดึงมา จะสามารถลิงก์กลับไปบนหน้าเว็บเพจของ Issue นั้น ๆ บน Jira ได้ด้วยเช่นกัน

9. การ Import จากแหล่งใดก็ตามจะได้ประเภทการกำหนดเวลาแบบ Absolute เสมอ เนื่องจากการ Import จะไม่สามารถระบุ Dependency ได้ ผู้ใช้จะสามารถเพิ่ม Dependency ด้วยการ Update ที่หลัง
10. ผู้ใช้จะสามารถส่งออกข้อมูล (Export) ของ Runbook ออกเป็น Microsoft Excel Workbook ได้ โดยที่จะแบ่งออกเป็นอยู่ 2 Worksheets ตามประเภทของ Activity



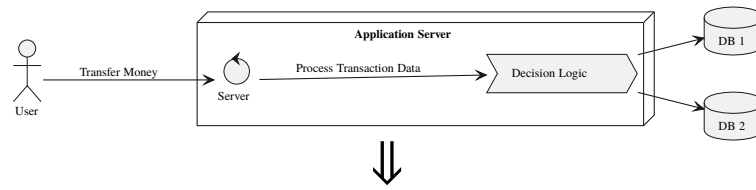
รูปที่ 3.9: ตัวอย่างของ Change Runbook จากการ Export

3.4.2 การปรับปรุง Configurations การเข้าถึงฐานข้อมูลกับทีม Fast Easy

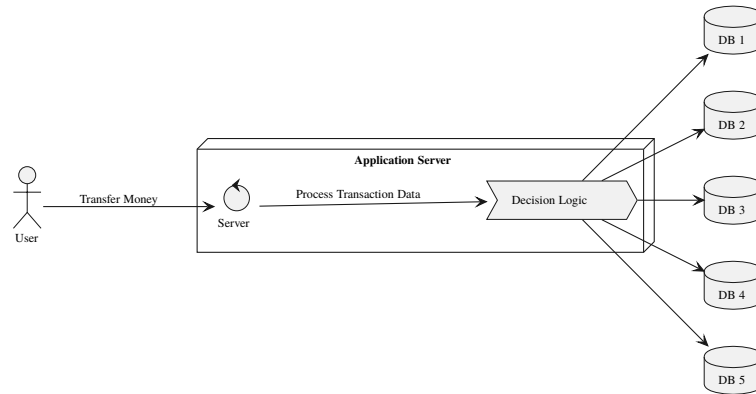
ในการจัดเก็บข้อมูลผู้ใช้งานแอปพลิเคชันธนาคารนั้น มีปริมาณข้อมูลจำนวนมาก โดยเฉพาะในส่วนของการบันทึกและค้นหาข้อมูลการทำธุรกรรม ซึ่งอาจส่งผลให้เกิดความล่าช้าได้ เพื่อแก้ไขปัญหานี้ ทีมพัฒนาได้ทำการแยกฐานข้อมูลออกเป็นหลาย Instances เพื่อช่วยลดระยะเวลาในการอ่านและเขียนข้อมูล ซึ่งกระบวนการนี้มีลักษณะคล้ายกับการทำ Indexing หรือ Shared Databases

เดิมทีฐานข้อมูลถูกจัดเก็บอยู่ใน 2 Instances และทางทีมมีแผนจะขยายเป็น 5 Instances เพื่อเพิ่มประสิทธิภาพในการประมวลผล จึงต้องมีการปรับเปลี่ยน Configurations เพื่อให้ซอฟต์แวร์สามารถจัดการการอ่านและเขียนข้อมูลระหว่าง Instances

Before Configuration:



After Configuration:



รูปที่ 3.10: การเปลี่ยน Configuration การเข้าถึงฐานข้อมูล

3.4.3 งานการบริหารข้อมูลผู้ใช้งานและสิทธิการใช้งานด้วย Keycloak

ในโปรเจกต์ xPlatform เดิมทีจะมีการจัดเก็บข้อมูลผู้ใช้และ Credential ภายในฐานข้อมูลของระบบเพียงอย่างเดียว ซึ่งทำให้เกิดความไม่สะดวกสำหรับผู้ใช้งานที่อยากเข้าถึง แอปพลิเคชันอื่น ๆ ที่อยู่ภายใต้การดูแลของบริษัทลูกค้า Integrated ไว้กับ xPlatform อย่างเช่น Jenkins ArgoCD Nexus หรือ SonarQube การที่ผู้ใช้ต้องล็อกอินซ้ำหลายครั้งกับแอปพลิเคชันต่าง ๆ ส่งผลให้เกิดประสบการณ์การใช้งานที่ไม่ราบรื่นและมีความซับซ้อนมากขึ้น ดังนั้น เพื่อแก้ไขปัญหานี้ เราจึงได้นำ Keycloak มาใช้ในการทำหน้าที่เป็น Identity Provider ทำให้แต่ละแอปพลิเคชันสามารถ Authenticate และ Authorize ผู้ใช้ผ่านบริการของ Keycloak เพียงอย่างเดียว

Keycloak เป็นซอฟต์แวร์ที่ทำหน้าที่ในการจัดการ Authentication และ Authorization ซึ่งมีจุดเด่นที่สำคัญในการทำ Single Sign-On (SSO) โดยช่วยให้ผู้ใช้ล็อกอินเพียงครั้งเดียวและสามารถเข้าถึงแอปพลิเคชันต่าง ๆ ได้โดยไม่ต้องทำการล็อกอินซ้ำ ซึ่งช่วยลดความซับซ้อนในการใช้งานและเพิ่มความสะดวกให้กับผู้ใช้เป็นอย่างมาก นอกจากนี้ Keycloak ยังทำให้การรวม Identity Provider หลาย ๆ ตัวเข้าด้วยกันเป็นเรื่องง่ายขึ้น

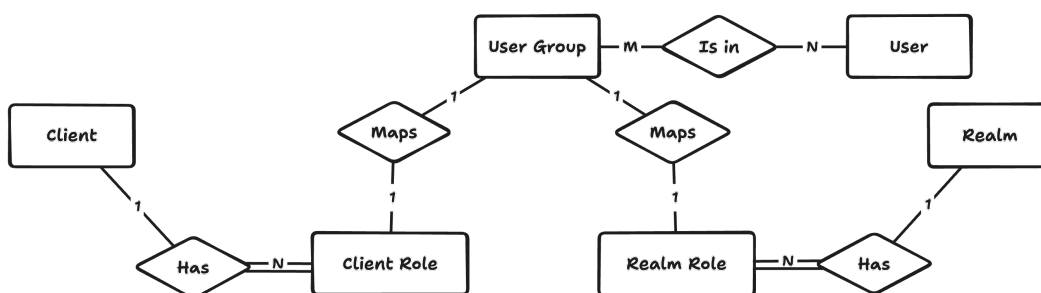
ในการจัดการข้อมูลผู้ใช้งาน ระบบของ Keycloak จะทำการแบ่ง Entity ออกเป็น 5 ประเภทหลัก ๆ ดังนี้

1. **User:** เป็นตัวแทนของผู้ใช้งานแต่ละคนในระบบ
2. **User Group:** คือกลุ่มของผู้ใช้งาน ซึ่งช่วยให้ง่ายต่อการกำหนดสิทธิการใช้งานต่าง ๆ โดยสามารถกำหนดสิทธิให้กับกลุ่มผู้ใช้ได้แทนการตั้งค่าเป็นรายบุคคล
3. **Client:** คือตัวแทนของ Services ต่าง ๆ เช่น Jenkins หรือแม้แต่ xPlatform เอง

4. **Role:** เป็นชุดสิทธิการใช้งานที่ระบุถึงการเข้าถึงหรือการทำงานที่กลุ่มผู้ใช้สามารถทำได้ โดย Role จะแบ่งออกเป็นสองประเภท ได้แก่ **Client Role** ซึ่งเชื่อมโยงกับบริการเฉพาะ และ **Realm Role**
5. **Realm:** คือ การแบ่งพื้นที่ในระบบออกเป็นส่วน ๆ เพื่อจัดการ **User Group**, **Role** และ **Client** ให้เกาะกันเป็นกลุ่ม

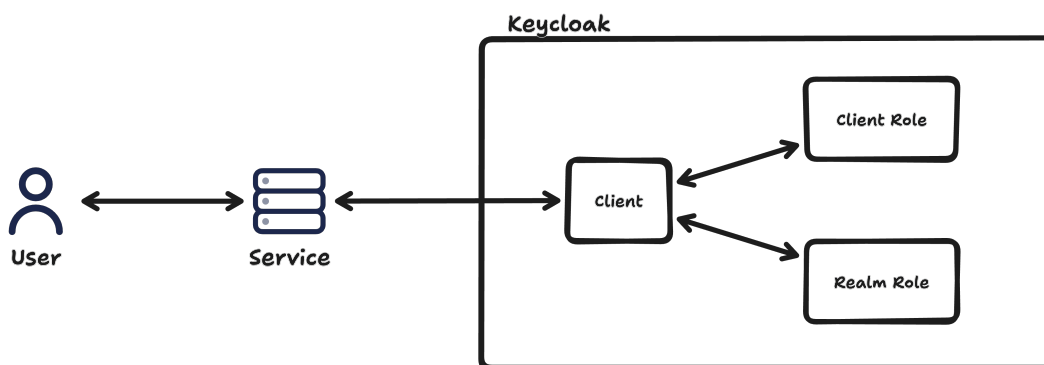
นอกจากนี้ xPlatform มีการแบ่งหน้าที่ความรับผิดชอบออกเป็น 3 ระดับ ดังนี้:

1. **ระดับ Company:** จะมีการสร้าง **Realm** และ **Client** สำหรับแต่ละบริษัท
2. **ระดับ Project:** จะมีการกำหนด **Client Roles**, **Realm Roles** และ **User Groups** สำหรับแต่ละโปรเจกต์ เพื่อควบคุมการเข้าถึงและการกำหนดสิทธิของผู้ใช้งาน
3. **ระดับ User:** มีหน้าที่สร้างผู้ใช้งานในแต่ละ **Realm** และเชื่อมโยงผู้ใช้เหล่านั้นเข้ากับ **User Group** ที่เกี่ยวข้องกับโปรเจกต์ เมื่อมีการเพิ่มผู้ใช้งานในโปรเจกต์



รูปที่ 3.11: แบบจำลองความสัมพันธ์ระหว่างข้อมูลภายใน Keycloak แบบย่อ

หมายเหตุ: ทุก ๆ Entity ในแบบจำลองความสัมพันธ์ ยกเว้น **Realm** นั้น มีความสัมพันธ์กับ **Realm** แบบ *many-to-one* แต่เส้นแสดงความสัมพันธ์ถูกตัดทอนออกเพื่อให้สามารถทำความเข้าใจได้ง่ายขึ้น แต่ละ Services สามารถ Authenticate และ Authorize แต่ละผู้ใช้งานตามแผนผังต่อไปนี้



รูปที่ 3.12: แบบจำลองการ Authorization กับ Authentication ผ่าน Keycloak

ชุดสิทธิการใช้งานถูกแบ่งออกเป็น 2 ประเภทหลัก ได้แก่ **Realm Role** และ **Client Role** โดยการเลือกใช้งานแต่ละประเภทขึ้นอยู่กับ **Service** ที่พร้อมจะ **Integrate** กับ **Keycloak** อย่างไร ตัวอย่างเช่น **Jenkins** และ **ArgoCD** จะอ้างอิงสิทธิการใช้งานจาก **Realm Role** ในขณะที่ **Nexus** และ **SonarQube** จะอ้างอิงจาก **Client Role** แทน

3.4.4 ฟังก์ชัน xPlatform Custom Library

ในระบบ **xPlatform** ผู้ใช้งานสามารถรวมโมดูลหลาย ๆ ส่วนเข้าด้วยกันเพื่อสร้างซอฟต์แวร์ขนาดใหญ่ขึ้น โดยโมดูลเหล่านี้มีบทบาทสำคัญในการเสริมสร้างความสามารถของระบบ ซอฟต์แวร์เหล่านี้สามารถแบ่งออกเป็น 4 ประเภทหลัก ได้แก่

1. **Service:** ส่วนที่ทำหน้าที่ประมวลผลหลัก ซึ่งมักจะเรียกว่าแอปพลิเคชัน เป็นส่วนที่ผู้ใช้งานหรือระบบอื่น ๆ สามารถสื่อสารหรือทำงานร่วมกับซอฟต์แวร์ได้
2. **Infrastructure:** ส่วนโครงสร้างพื้นฐานที่ช่วยสนับสนุนการทำงานของ **Service** เช่น ฐานข้อมูล พื้นที่จัดเก็บข้อมูล พื้นที่แคช หรือบริการที่ช่วยในกระบวนการจัดการข้อมูลต่าง ๆ เช่น **Data Factory**
3. **Secret:** ส่วนที่เกี่ยวข้องกับการจัดเก็บข้อมูลที่มีความอ่อนไหว เช่น รหัสผ่านของผู้ใช้งาน โทเคน API หรือข้อมูลที่ต้องการความปลอดภัยสูง การจัดการกับข้อมูลเหล่านี้จะต้องมีการเข้ารหัสและการควบคุมการเข้าถึงอย่างเข้มงวดเพื่อป้องกันการรั่วไหลของข้อมูล
4. **Custom Library:** ส่วนของโค้ดที่ประกอบด้วยฟังก์ชันหรือ **Definition** ที่ถูกใช้งานบ่อย เพื่อให้ **Services** หลายตัวสามารถใช้งานร่วมกันได้โดยไม่ต้องเขียนโค้ดซ้ำ ๆ ในแต่ละ **Service** การใช้ **Custom Library** ช่วยลดความซ้ำซ้อนและป้องกันการเกิด **Code Duplication** ซึ่งเป็นประโยชน์อย่างยิ่งสำหรับซอฟต์แวร์ที่ใช้สถาปัตยกรรมแบบ **Microservice**

ก่อนหน้านี้ **xPlatform** มีเพียง 3 ส่วนประกอบหลัก ได้แก่ **Service Infrastructure** และ **Secret** ซึ่งสามารถตอบโจทย์การใช้งานในระดับหนึ่ง ทีมพัฒนาได้มอบหมายงานให้พัฒนาส่วนของ **Custom Library** เพิ่มเติม ส่วนประกอบนี้ถูกออกแบบมาเพื่อให้การใช้งานฟังก์ชันที่ซับซ้อนมีความคล่องตัวมากขึ้น และช่วยในกระบวนการพัฒนาซอฟต์แวร์รวดเร็วขึ้นและลดภาระงานในการจัดการโค้ดในแต่ละ **Service**

การพัฒนาส่วนของ **Custom Library** นั้นมีหลายขั้นตอนที่คล้ายคลึงกับ **Service** โดยเริ่มจากการสร้าง **Repository** ใหม่สำหรับจัดเก็บโค้ด และการสั่ง **Build** โค้ด ซึ่งเปรียบเสมือนการแปลงโค้ดให้อยู่ในรูปแบบที่คอมพิวเตอร์สามารถเข้าใจและประมวลผลได้ (คล้ายกับการ **Compile** ในภาษาโปรแกรม เช่น **C++** หรือ **Java**) อย่างไรก็ตาม **Custom Library** แตกต่างจาก **Service** ตรงที่ว่าเมื่อพัฒนาเสร็จสิ้นแล้วจะไม่มี **Deploy** เนื่องจาก **Custom Library** ไม่จำเป็นต้องเปิดให้ระบบอื่น ๆ แต่จะถูกฝังภายในโค้ดของ **Service** ต่าง ๆ ที่เรียกใช้งานแทน

3.4.5 ฟังก์ชัน xPlatform Documentation

ฟังก์ชันนี้เป็นฟังก์ชันที่ถูกพัฒนาขึ้นมาเพื่อค้นหาและดึงข้อมูลจากไฟล์ **Markdown** ที่อยู่ใน **Repositories** ต่าง ๆ อย่าง **Service** หรือ **Custom Library** บนแพลตฟอร์ม **GitLab** ฟังก์ชันนี้จะช่วยให้สามารถค้นหาไฟล์ **Markdown** เมื่อฟังก์ชันนี้ค้นพบไฟล์ **Markdown** ที่เกี่ยวข้องแล้ว ระบบจะทำการดึงเนื้อหาภายในไฟล์

เหล่านั้นมาแสดงในหน้าเว็บเพจ ทั้งนี้เพื่อช่วยให้ผู้ใช้งานสามารถอ่าน Documentation ของแต่ละ Repository ผ่านเว็บไซต์ของ xPlatform ได้โดยไม่ต้องจำเป็นต้องเข้าไปอ่านใน GitLab

3.5 โครงสร้างพื้นฐานและสถาปัตยกรรมของซอฟต์แวร์

3.5.1 โครงสร้างพื้นฐานและสถาปัตยกรรมของซอฟต์แวร์โปรเจกต์ xPlatform

xPlatform เป็นโปรเจกต์ Full-Stack ที่ใช้สถาปัตยกรรมแบบ Microservice

1. หลังบ้าน

- (a) **ExpressJS (Typescript):** เป็นเฟรมเวิร์คสำหรับการพัฒนาเว็บแอปพลิเคชันฝั่งเซิร์ฟเวอร์ที่ใช้ Node.js โดย ExpressJS มีความยืดหยุ่นสูงและสามารถสร้าง API ได้อย่างรวดเร็ว ส่วนการใช้ Typescript ช่วยเพิ่มความปลอดภัยในการเขียนโค้ดด้วยการตรวจสอบประเภทของข้อมูล (Type Checking) ทำให้โค้ดมีความเสถียรและลดข้อผิดพลาด
- (b) **Apollo GraphQL:** เป็นไลบรารีสำหรับการใช้งาน GraphQL ซึ่งเป็นภาษาสำหรับการดึงข้อมูลจากเซิร์ฟเวอร์ โดย Apollo GraphQL สามารถช่วยให้ฝั่งเรียกใช้ API มีความสะดวกมากขึ้น โดยเฉพาะเมื่อใช้ในสถาปัตยกรรมไมโครเซอร์วิส ซึ่งช่วยให้สามารถจัดการการดึงข้อมูลจากบริการหลาย ๆ แหล่ง
 - i. การควบคุมประเภทข้อมูล (Controlled Type) GraphQL ช่วยให้สามารถกำหนดประเภทของข้อมูลที่ส่งและรับได้อย่างชัดเจน ซึ่งช่วยลดข้อผิดพลาดในการส่งข้อมูลที่ไม่ตรงตามประเภทที่กำหนด และเพิ่มความปลอดภัยในการใช้งาน TODO: reference apollo Type ยกตัวอย่าง

```
getPosts(getPostsQuery: GetPostsQuery!): GetPostsResult!
```

```
input GetPostsQuery {  
  ownerId: String!  
  searchQuery: String  
  afterDate: Date  
}
```

```
type GetPostsResult {  
  posts: [Post]!  
  owner: User  
}
```

```
type User {  
  userId: String!  
  username: String!  
  posts: [Post]!  
  comments: [Comment]!
```

```

}

type Post {
  postId: String!
  content: String!
  owner: User
  replies : [Comment]!
}

```

ในตัวอย่างนี้ API ที่ชื่อว่า `getPosts` จะรับข้อมูลเข้าในรูปของ `GetPostsQuery` และจะส่งข้อมูลกลับในรูปของ `GetPostsResult` โดยที่เครื่องหมาย `[]` หมายถึงข้อมูลประเภทอาร์เรย์ และ เครื่องหมาย `!` หมายถึงฟิลด์นั้น ๆ เป็น **non-nullable** ซึ่งหมายความว่าฟิลด์ต้องมีค่าภายในทุกครั้งที่มีการส่งข้อมูล ไม่สามารถเป็น `null` ได้

- ii. ฝั่งเรียกใช้ API สามารถเลือกข้อมูลที่เป็นต่อการใช้งานได้ยืดหยุ่น นอกจากนี้ยังสามารถเลือกข้อมูลภายในที่เป็น **Object** ได้ เช่น ผู้ใช้ที่เรียกใช้งาน `getPosts` จะสามารถเลือกข้อมูลใด ๆ ใน `GetPostsQuery` ได้ตามต้องการ นอกจากนี้ยังสามารถค้นหาข้อมูลอื่น ๆ ภายใน `posts` และ `owner` ได้อีกด้วย **TODO: reference apollo Resolver**

ยกตัวอย่าง

```

getPosts {
  posts {
    owner {
      username
    }
  }
  owner {
    posts {
      content
    }
  }
}

```

ในตัวอย่างนี้ ผู้ใช้งานขอค้นหาข้อมูล `posts` และ `owner` จากข้อมูลประเภท `GetPostsResult`

- A. ฟิลด์ `posts` จะดึงข้อมูลเกี่ยวกับผู้โพสต์ (`owner`) และในส่วนของ `owner` จะทำการค้นหาชื่อผู้ใช้ (`username`)
- B. ฟิลด์ `owner` จะเรียกดูโพสต์ (`post`) ทั้งหมดที่ผู้ใช้นั้นสร้างขึ้น และภายในแต่ละโพสต์จะค้นหาเนื้อหาที่เกี่ยวข้อง
- iii. การรวม API ด้วย **Apollo Federation** คือการสร้าง **service** ที่สามารถ **reroute requests** ไปยัง **service** อื่น ๆ ซึ่งหมายความว่าผู้ใช้สามารถส่ง **API request** ไปยัง **service**

ใด ๆ ก็ได้ผ่าน router TODO: reference Apollo Federation

- iv. Apollo Federation ยังมีพีเจอร์ Reference Resolver ซึ่งช่วยในการค้นหาข้อมูลของ Objects ที่อยู่ใน service ที่แตกต่างกัน ตัวอย่างเช่น “post” service มีการเก็บข้อมูลของทุก ๆ โพสต์และคอมเมนต์ แต่ข้อมูลของผู้ใช้แต่ละคนจะถูกจัดการโดย “user” service แทน Apollo Federation จะช่วยให้ใน API getPost ที่ถูกจัดการโดย “post” service สามารถไปส่ง userId ให้ router ไปจัดการการดึงข้อมูล user จาก “user” service ต่อ โดยอัตโนมัติ TODO: reference resolveReference

2. ตัวอย่างโครงสร้างพื้นฐานที่ใช้พัฒนา

- (a) **MySQL:** เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ ที่ใช้ภาษา SQL ในการสืบค้นและจัดการข้อมูล ซึ่ง MySQL เป็นระบบที่เหมาะสมสำหรับการจัดเก็บข้อมูลที่มีโครงสร้างชัดเจน และสามารถรองรับการทำงานร่วมกับแอปพลิเคชันต่าง ๆ
- (b) **Jenkins:** เป็นเครื่องมือสำหรับการทำ Continuous Integration (CI) และ Continuous Delivery (CD) ซึ่งช่วยให้การพัฒนาและการปรับปรุงซอฟต์แวร์เป็นไปอย่างต่อเนื่องและอัตโนมัติ Jenkins สามารถทำการทดสอบและดีพลอยโค้ดได้อย่างรวดเร็ว ลดเวลาและข้อผิดพลาดในการพัฒนา
- (c) **Keycloak:** เป็นซอฟต์แวร์ที่ทำหน้าที่ในการ Authentication ซึ่งมีจุดเด่นในการทำ Single Sign-On (SSO) ที่ช่วยให้ผู้ใช้สามารถล็อกอินเพียงครั้งเดียวและเข้าถึงแอปพลิเคชันอื่น ๆ ได้โดยไม่ต้องทำการล็อกอินซ้ำ

บทที่ 4

ผลของการทำงาน

4.1 สรุปผลการทำงานตามรางขอบเขตงาน

ตลอดระยะเวลาการทำงานที่บริษัท เอสซีบี เทคเอกซ์ ข้าพเจ้าได้สะสม Story Points รวมทั้งหมด xx คะแนน ซึ่งมากกว่าข้อกำหนดขั้นต่ำที่ระบุในรางขอบเขตงานที่ 60 Story Points อย่างไรก็ตาม ข้าพเจ้าไม่ได้มีโอกาสทำโปรเจกให้กับทีม Fast Easy ตามที่คาดหวังไว้ เนื่องจากทั้งสองทีมเห็นตรงกันว่าการย้ายข้าพเจ้าไปทำงานในทีม Fast Easy อาจไม่เหมาะสมมากนัก เนื่องจากข้าพเจ้าและเพื่อนร่วมฝึกงานยังไม่มีประสบการณ์กับโค้ดเบสของทีมนั้น ในขณะที่มีประสบการณ์ในการพัฒนาบนโปรเจก xPlatform มากพอที่จะเริ่มงานอื่น ๆ ได้เลย

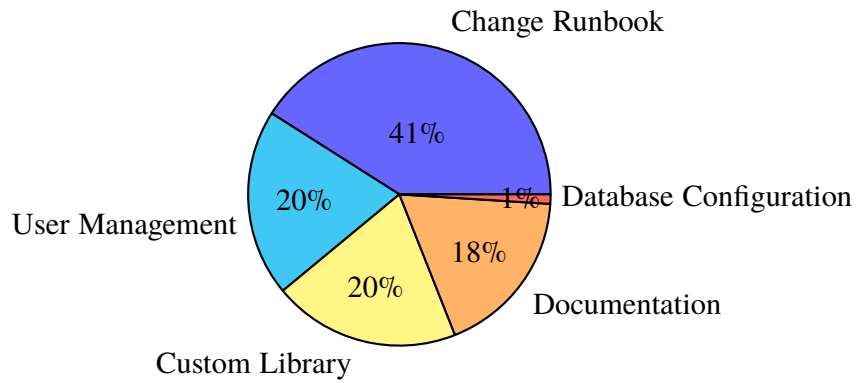
อย่างไรก็ตาม ข้าพเจ้ายังคงได้ทำงานร่วมกับทีม Fast Easy ในระดับเล็กน้อย ส่วนงานที่เหลือจะเป็นงานย่อยขนาดกลางในทีม xPlatform ตามที่ได้ระบุไว้ในบทก่อนหน้านี้

4.2 สัดส่วนการทำงาน

Story Points ส่วนใหญ่จะอยู่ที่ฟีเจอร์ xPlatform Change Runbook ซึ่งมีคะแนนรวมถึง 41 Story Points เนื่องจากโครงการนี้เป็นโครงการใหญ่ตามที่กำหนดในรางขอบเขตงาน ในโครงการจะมีการแบ่งงานออกเป็นงานย่อย โดยเฉลี่ยแต่ละงานจะมีคะแนนประมาณ 3 Story Points ส่วนชิ้นงานขนาดปานกลางและขนาดเล็กที่เหลือจะสะสมคะแนนรวมกันได้ทั้งหมด xx Story Points รายละเอียดคะแนนความยากของแต่ละงานจะถูกบันทึกไว้ในภาคผนวก

	Story Points	อัตราส่วน
Change Runbook	41	41%
User Management	20	41%
Custom Library	20	41%
Documentation	20	41%
Database Configuration	1	5%

ตารางที่ 4.1: ตารางแสดงสัดส่วนของ Story Points



รูปที่ 4.1: แผนภูมิรูปวงกลมแสดงสัดส่วนของ Story Points

4.3 ช่วงระยะเวลาการทำงาน

ข้าพเจ้าได้เริ่มต้นทำงานที่พีเจเออร์ xPlatform เป็นระยะเวลาประมาณ 3 เดือน โดยในช่วงเวลาดังกล่าว ข้าพเจ้าได้ทำงานร่วมกับทีม Fast Easy เป็นระยะเวลา 1 สัปดาห์หลังจากเสร็จสิ้นโปรเจกต์แรก และในระยะเวลาที่เหลือข้าพเจ้าได้มีส่วนร่วมในการทำงานในโปรเจกต์ขนาดกลางกับทีม xPlatform ซึ่งจะได้แผนปฏิบัติงานสหกิจศึกษาดังนี้

	2024															
	ก.ค.				ส.ค.				ก.ย.				ต.ค.			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Change Runbook																
DB Configuration																
Documentation																
Custom Library																
User Management																
Others																

ตารางที่ 4.2: ตารางแผนปฏิบัติงานสหกิจศึกษา

บทที่ 5

สภาพแวดล้อมของการทำงานในองค์กร

5.1 ก่อนที่จะปฏิบัติงานสหกิจศึกษา

ก่อนที่จะได้ปฏิบัติงานสหกิจศึกษานั้น ข้าพเจ้าได้ยื่นสมัครไป 3 บริษัท ได้แก่ บริษัท อะแวร์ คอร์ปอเรชั่น จำกัด ในตำแหน่ง Java Developer บริษัท เอสซีบี เทคเอกซ์ ในตำแหน่ง Software Engineer และบริษัท ที.ซี.ซี. เทคโนโลยี จำกัด ในตำแหน่ง Software Engineer

โดยที่แต่ละบริษัทจะมีขั้นตอนการสมัครงานดังนี้

1. บริษัท อะแวร์ คอร์ปอเรชั่น จำกัด:

- (a) เริ่มต้นด้วยการส่งจดหมายสมัครงานทางอีเมลถึงฝ่ายบุคคล โดยในอีเมลต้องแนบ ใบรับรองผลการศึกษา (Transcript) ประวัติย่อ (Resume) และวิดีโอแนะนำตัวเอง
- (b) ทำการสอบสัมภาษณ์ออนไลน์เกี่ยวกับการแก้ปัญหาเบื้องต้น และความรู้พื้นฐานเกี่ยวกับภาษา Java ใช้เวลาทั้งหมด 1 ชั่วโมง
- (c) เข้าร่วมสัมภาษณ์สหกิจศึกษา โดยจะสอบถามเกี่ยวกับโปรเจกต์ที่เคยทำ และให้อธิบายคำตอบจากการสอบครั้งก่อน ใช้เวลาในการสัมภาษณ์ประมาณ 1 ชั่วโมง

2. บริษัท เอสซีบี เทคเอกซ์

- (a) ส่งแบบฟอร์ม Google Form ซึ่งประกอบไปด้วยข้อมูลส่วนตัวอย่างละเอียด เช่น ประวัติการศึกษา ผลการเรียน Resume และตำแหน่งที่สนใจในการทำงาน
- (b) ทำการสอบออนไลน์เกี่ยวกับการแก้ไขปัญหเบื้องต้นผ่านแพลตฟอร์ม HackerRank
- (c) เข้าร่วมสัมภาษณ์สหกิจศึกษา ซึ่งในระหว่างการสัมภาษณ์นี้จะมีการสอบถามเกี่ยวกับโปรเจกต์ที่เคยทำ รวมถึงการให้ผู้สมัครอธิบายคำตอบจากการสอบครั้งก่อน

ทั้งบริษัท อะแวร์ คอร์ปอเรชั่น จำกัด และบริษัท เอสซีบี เทคเอกซ์ ได้ติดต่อข้าพเจ้าเพื่อเข้าฝึกงานอย่างเป็นทางการ ส่วนบริษัท ที.ซี.ซี. เทคโนโลยี จำกัด ได้ติดต่อครั้งแรกหลังจากที่ข้าพเจ้ายืนยันการฝึกปฏิบัติงานสหกิจศึกษาที่บริษัท เอสซีบี เทคเอกซ์ ทำให้ข้าพเจ้าไม่มีโอกาสในการสมัครงานกับบริษัทดังกล่าว

วันที่	เหตุการณ์
13 ม.ค. 2567	ส่งจดหมายสมัครบริษัท อะแวน์
16 ม.ค. 2567	บริษัท อะแวน์ ส่งจดหมายเชิญชวนทดสอบความรู้ออนไลน์
18 ม.ค. 2567	กรอกแบบฟอร์มสมัครบริษัท เอสซีพี เทคเอกซ์
30 ม.ค. 2567	ทดสอบความรู้ออนไลน์กับบริษัท อะแวน์
30 ม.ค. 2567	บริษัท เอสซีพี เทคเอกซ์ ส่งจดหมายเชิญชวนทดสอบความรู้ออนไลน์
31 ม.ค. 2567	ทดสอบความรู้ออนไลน์กับบริษัท เอสซีพี เทคเอกซ์
2 ก.พ. 2567	บริษัท เอสซีพี เทคเอกซ์ ส่งจดหมายเชิญสัมภาษณ์
2 ก.พ. 2567	บริษัท อะแวน์ ส่งจดหมายเชิญสัมภาษณ์
8 ก.พ. 2567	สัมภาษณ์กับบริษัท เอสซีพี เทคเอกซ์
9 ก.พ. 2567	สัมภาษณ์กับบริษัท อะแวน์
12 ก.พ. 2567	บริษัท อะแวน์ติดต่อแสดงความยินดีเข้าโปรแกรมฝึกงาน
13 ก.พ. 2567	บริษัท เอสซีพี เทคเอกซ์ติดต่อแสดงความยินดีเข้าโปรแกรมฝึกงาน

ตารางที่ 5.1: ตารางแสดงไทม์ไลน์การทำงาน

5.2 เงินเดือนและสวัสดิการ

พนักงานทุกคนจะได้รับเงินเดือนเริ่มต้นที่ 30,000 บาท โดยเงินเดือนจะปรับขึ้นทุกปีตามผลการปฏิบัติงาน สำหรับนักศึกษาฝึกงานหรือสหกิจศึกษา จะได้รับเบี้ยเลี้ยงวันละ 500 บาท คิดเป็นประมาณ 10,000 บาท ต่อเดือน

นอกจากเงินเดือนและเบี้ยเลี้ยงแล้ว พนักงานยังได้รับสิทธิประโยชน์เพิ่มเติมดังนี้:

1. พนักงานแต่ละคนจะได้รับ MacBook Pro สำหรับใช้งานตลอดระยะเวลาการทำงาน โดยต้องส่งคืนเมื่อสิ้นสุดการจ้างงาน
2. อาหารเข้าฟรีทุกวันทำงาน
3. ในแต่ละปีพนักงานสามารถลาพักร้อน 15 วัน ลาเดือนเกิด 1 วัน ลากิจ 5 วัน และลาป่วย 30 วัน โดยสามารถสะสมวันลาพักร้อนข้ามปีได้สูงสุดถึง 5 วัน นอกจากนี้พนักงานสามารถลาบวช 105 วัน ซึ่งไม่เปลี่ยนแปลงในแต่ละปี
4. นักศึกษาฝึกงานจะได้รับประกันชีวิตจำนวน 5000 บาท ประกันอุบัติเหตุ 150000 บาท และความคุ้มครองค่ารักษาพยาบาล (AME) อีก 5000 บาท โดยทั้งหมดเป็นแบบประกันกลุ่ม
5. พนักงานทั่วไปจะได้รับประกันชีวิตจำนวน 1000000 บาท ประกันอุบัติเหตุจำนวน 1000000 บาท และประกันสุขภาพตามแผนที่พนักงานเลือก
6. บริการรถรับส่งพนักงานระหว่างสำนักงานและสถานที่ต่าง ๆ สองรอบต่อวัน:
 - (a) รอบเช้า: 6:30 – 9:00 น.
 - (b) รอบเย็น: 17:20 – 20:00 น.

จุดขึ้น-ลงรถมี 4 สถานที่:

- (a) อาคาร SCB Plaza West
- (b) BTS หมอชิต (ทางออก 2)
- (c) MRT สวนจตุจักร (ทางออก 4)
- (d) Big C วงศ์สว่าง

5.3 วัฒนธรรมองค์กร

บริษัทใช้รูปแบบการทำงานแบบไฮบริด 8 ชั่วโมงต่อวัน 9:00 - 18.00 น. มีพักเบรกรับประทานอาหารเที่ยง 1 ชั่วโมง โดยพนักงานจะเข้ามาทำงานที่สำนักงาน 2 วันต่อสัปดาห์ และทำงานทางออนไลน์อีก 3 วัน ส่วนวันที่ต้องเข้ามาทำงานที่สำนักงานจะมีการกำหนดให้แต่ละทีมผลัดกันมา

5.3.1 กิจกรรมและการอบรม

บริษัทมีการจัดกิจกรรมที่เปิดให้พนักงานทุกคนสามารถเข้าร่วมได้ เพื่อส่งเสริมการการพัฒนาทักษะต่าง ๆ ที่เกี่ยวข้องกับการทำงาน กิจกรรมเหล่านี้ยังช่วยสร้างความสัมพันธ์ระหว่างพนักงานและผู้บริหาร โดยมีตัวอย่างกิจกรรมดังต่อไปนี้

1. **Knowledge Sharing:** กิจกรรมที่ให้โอกาสที่ผู้มีความรู้และประสบการณ์มาร่วมพูดคุยและแบ่งปันความรู้ใหม่ ๆ เช่น เทคนิคการออกแบบ UX/UI ให้ประสบความสำเร็จ วิธีป้องกันตัวเองจาก Office Syndrome และการดูแลสุขภาพร่างกาย รวมถึงเคล็ดลับในการเพิ่มความสุขและสร้างชีวิตที่มีความสุขยิ่งขึ้นในที่ทำงาน
2. **Town Hall:** กิจกรรมที่ CEO จะมาพบปะพูดคุยและตอบคำถามจากพนักงานในบริษัท โดยกิจกรรมนี้จะจัดขึ้นทุก ๆ 2 เดือน
3. กิจกรรมวิ่งมาราธอนเฉลิมฉลองครบรอบ 3 ปี

สำหรับนักศึกษาฝึกงานจะมีการอบรมที่จัดขึ้นโดยแผนก Human Resources เพื่อเสริมสร้างความรู้ที่สำคัญต่อการทำงาน นอกเหนือจากการพัฒนาซอฟต์แวร์อย่างเดียว โดยกิจกรรมเหล่านี้จะช่วยให้นักศึกษาได้เรียนรู้ทักษะใหม่ ๆ และเข้าใจเกี่ยวกับการทำงานในสภาพแวดล้อมจริง

1. **Welcome New Intern:** กิจกรรมแนะนำบริษัท เอสซีบี เทคเอกซ์ และสถานที่ทำงานที่ SCBX Park Plaza เพื่อให้ นักศึกษาฝึกงานได้รู้จักกับองค์กรและสภาพแวดล้อมการทำงาน
2. **Sharing Session:** กิจกรรมที่พนักงานจากแต่ละตำแหน่งมาร่วมแบ่งปันประสบการณ์และรายละเอียดเกี่ยวกับงานที่ทำในตำแหน่งนั้น ๆ ให้ นักศึกษาฝึกงานได้เรียนรู้
3. **Tips on Building a Strong Profile:** กิจกรรมอบรมการเขียนเรซูเม่ การจัดการโปรไฟล์โซเชียลมีเดียที่น่าสนใจ และการสร้างการเชื่อมต่อกับบุคคลอื่น
4. **Agile & Cloud & AI Skills:** กิจกรรมอบรมเกี่ยวกับการใช้งาน Azure Cloud Services และเทคนิค Agile ที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์
5. **Intern Checking:** กิจกรรมผ่อนคลายสำหรับนักศึกษาฝึกงานให้ได้มีโอกาสสร้างสัมพันธ์และพูดคุยกัน

6. **Design Thinking:** กิจกรรมอบรมการคิดเชิงออกแบบ (Design Thinking) สำหรับการพัฒนาซอฟต์แวร์

7. **Communication Skills:** กิจกรรมอบรมเกี่ยวกับทักษะการสื่อสารและการพูดคุยกับผู้อื่น

5.3.2 ความคิดเห็นส่วนตัว

TODO

5.4 ข้อเสนอแนะในการสมัครงาน

ในส่วนของทีม xPlatform Developer นั้น สมาชิกในทีมทุกคนมีทักษะแบบ Full-stack และมีความรู้ในการรัน Jenkins Jobs ด้วยเช่นกัน โดยทักษะที่สำคัญประกอบด้วย:

1. **Front-end:** ต้องมีความรู้ในการใช้ React NextJS และ Apollo Client
2. **Back-end:** ควรมีประสบการณ์ในการพัฒนาโปรเจกแบบ Microservice ใช้ Typescript และ ExpressJS ได้อย่างคล่องแคล่ว รวมถึงมีความรู้ใน Apollo GraphQL

อย่างไรก็ตาม มันไม่มีไม่จำเป็นต้องเชี่ยวชาญทุกด้าน ตัวอย่างเช่น ข้าพเจ้าไม่มีประสบการณ์ในการเขียน backend ด้วย Typescript ไม่เคยพัฒนา Microservice หรือไม่เคยใช้งาน GraphQL มาก่อน แต่สิ่งที่ผู้สมัครควรที่จะให้ความสำคัญคือการแสดงให้เห็นถึงความรู้ในเครื่องมือและเทคโนโลยีต่าง ๆ ในอุตสาหกรรมซอฟต์แวร์ รวมถึงความพร้อมที่จะเรียนรู้สิ่งใหม่ ๆ

5.5 ข้อเสนอจากบริษัท

ทางบริษัทไม่ได้ให้ข้อเสนอการทำงานทันทีหลังจากที่นักศึกษาจบการศึกษาในมหาวิทยาลัย อย่างไรก็ตามจากการพูดคุยกับผู้ที่เกี่ยวข้อง มีการกล่าวว่าผู้ที่ผ่านการปฏิบัติงานสหกิจศึกษาหรือฝึกงานกับบริษัทมีโอกาสค่อนข้างสูงที่จะได้รับการพิจารณาในการรับเข้าทำงานหลังจากสำเร็จการศึกษา แม้ว่าจะยังไม่มีมารับรองอย่างเป็นทางการในเรื่องนี้