

CP Capstone Project Proposal

ข้อเสนอโครงการรวบยอดวิศวกรรมคอมพิวเตอร์ 1

เรื่อง

อักขราวิสุทธิ์

Akarawisut

โดย

สุชาครีย์ มณีปกรณ์ รหัสนิต 6230563321

รวีพร เอกคุณานนท์ รหัสนิต 6231353021

ขจรพงษ์ พิมพ์ม่วง รหัสนิต 6232002021

อาจารย์ที่ปรึกษา

รศ.ดร. โชติรัตน์ รัตนามัทธนะ

ที่ปรึกษาร่วม

ดร. วิชญ์ เนียรนาทตระกูล

รายงานนี้เป็นส่วนหนึ่งของวิชา 2110488 โครงการรวบยอดวิศวกรรมคอมพิวเตอร์

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ประจำปีการศึกษา 2565

บทคัดย่อ

อักษราวิสุทธิ์เป็นระบบตรวจสอบการลอกเลียนผลงานทางวิชาการที่ให้ผู้สนใจและต้องการจะตรวจสอบงานเขียนของตนเองว่ามีส่วนใดของงานมีความคล้ายคลึงกับงานที่ได้มีการตีพิมพ์ไปแล้วหรือไม่ ซึ่งระบบนี้มีการใช้อย่างแพร่หลายด้วยคุณสมบัติที่รองรับการตรวจงานภาษาไทย ทำให้เป็นที่ระบบนั้นเป็นที่ต้องการของมหาวิทยาลัยในประเทศไทยหลายแห่ง ส่งผลให้ปัจจุบันผู้ใช้ต้องรอผลลัพธ์จากระบบเป็นเวลานานเนื่องจากโครงสร้างของระบบนั้นไม่สามารถรองรับปริมาณคำขอที่หนาแน่นและเพิ่มสูงขึ้นจากความต้องการใช้งานได้ โครงการนี้มีจึงวัตถุประสงค์เพื่อศึกษาและพัฒนาเกี่ยวกับการย้ายบริการไปยังระบบคลาวด์ เพื่อแก้ไขปัญหาการขยายตัวของระบบ และปรับปรุงเวอร์ชันของชุดเทคโนโลยีที่ใช้ให้เป็นเวอร์ชันที่เป็นปัจจุบัน โดยขั้นตอนในการดำเนินงานในโครงการนี้ สามารถทำได้โดยการออกแบบโครงสร้างของระบบประกอบด้วย 3 ส่วน คือ Front-end website for submission and report view, Administrator Portal และ Portal for TDC (Dashboard) จากนั้นเขียนโปรแกรมบรรยายวิธีสร้างโครงสร้างดังกล่าวเพื่อให้คอมพิวเตอร์สามารถทำการสร้างระบบขึ้นมาได้อัตโนมัติ การพัฒนาระบบใหม่นี้จะส่งผลให้ระบบมีความเสถียรมากขึ้น มีฟังก์ชันครบถ้วนมากขึ้น ส่งผลต่อผู้ใช้งานของระบบ สามารถใช้งานระบบที่มีความน่าเชื่อถือ มีประสิทธิภาพมากยิ่งขึ้น ซึ่งระบบนี้จะถูกพัฒนาในเดือนมกราคมถึงพฤษภาคม พ.ศ.2566

Abstract

Akarawisut service allows users to upload reports, papers, or articles to websites, and then automatically checks them for plagiarism and emails the results. The services' usage has occasionally surged quickly due to its Thai language checking feature. The old workstation's resources were insufficient to handle the demand from users from various universities and general users, which put a severe stress on the entire system and caused a line of jobs in the queue to grow longer and longer over time. The project's objectives include migrating the Akarawisut service that is now run on an On-Premises server to be used with Cloud Services and updating the frameworks that make up the existing system's dependencies to recent releases that enable newly created features and functionalities. To achieve the objective, an expert redesigned a new architecture, divided it into three subunits, and then assigned each module to three individual developers. The assignments included Front-end websites for report viewing and submission, Administrator Portals, and TDC Portals (Dashboard). The developers then built Infrastructure as a Code (IAC), a technique that automatically builds the whole system from code alone. The new system functioned effectively thanks to its capacity for elastic scaling and workload distribution. The system should be able to manage several jobs being submitted simultaneously and should reduce costs when there are less submissions. The creation of the to-be system must take place between January and May of 2022.

สารบัญ

บทคัดย่อ	ก
Abstract	ข
บทที่ 1 บทนำ	1
ที่มาและความสำคัญ	1
วัตถุประสงค์ของโครงการ	5
ขอบเขตของโครงการ	5
ขั้นตอนและแผนการดำเนินโครงการ	6
ประโยชน์ที่คาดว่าจะได้รับ	7
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	8
ทฤษฎีที่เกี่ยวข้อง	8
1. Cloud Computing	8
2. Infrastructure as a Code (IAC)	9
บทที่ 3 แนวทางการพัฒนาโครงการ	10
บทที่ 4 ผลกระทบทางสังคมของโครงการ	13
บทที่ 5 บทสรุปและงานที่จะทำต่อไป	14
บทสรุป	14
งานที่จะทำต่อไป	14
บรรณานุกรม	15

สารบัญรูปภาพ

ภาพที่ 1 จำนวนครั้งที่ผลงานทางวิชาการถูกเข้าถึงออนไลน์ในมหาวิทยาลัยที่ใช้ระบบอักษรวิสุทธิ์	1
ภาพที่ 2 Diagram ของระบบปัจจุบัน	2
ภาพที่ 3 กราฟแสดงปริมาณการใช้งานตั้งแต่เริ่มต้นในระบบอักษรวิสุทธิ์ (จำนวนครั้งในการตรวจสอบเอกสาร)	3
ภาพที่ 4 กราฟแสดงปริมาณการใช้งานแบบเปรียบเทียบรายปีในแต่ละเดือนของระบบอักษรวิสุทธิ์	4
ภาพที่ 5 Diagram ของระบบที่จะพัฒนาต่อไป	10

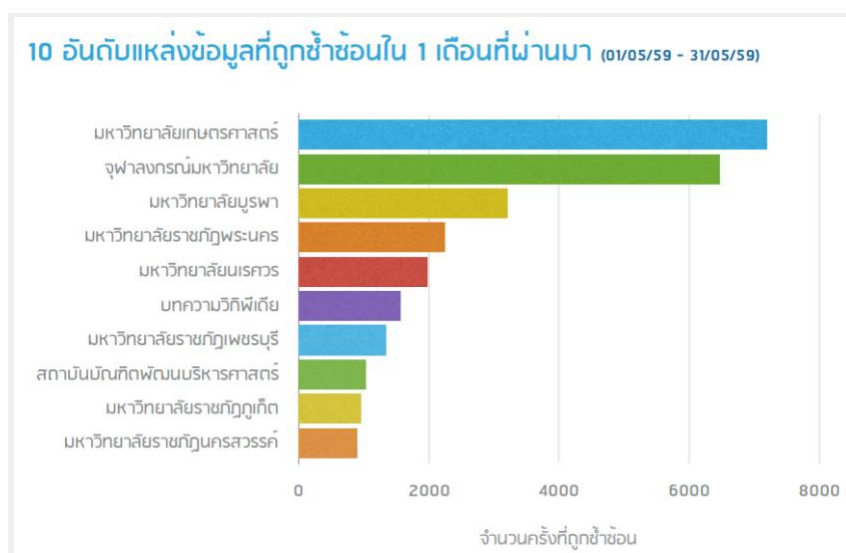
สารบัญตาราง

ตารางที่ 1 แผนการดำเนินงาน	6
----------------------------	---

บทที่ 1 บทนำ

ที่มาและความสำคัญ

อักขราวิสุทธิ์ เป็นระบบตรวจสอบการลอกเลียนผลงานทางวิชาการ โดยจุฬาลงกรณ์มหาวิทยาลัย ร่วมกับ บริษัท อินสปิกา จำกัด (INSPICA CO.,LTD.) เริ่มจัดทำโครงการนี้ตั้งแต่ปลายปี พ.ศ. 2555 ในขณะที่เริ่มต้น ต้องการจะแก้ไขปัญหาการลอกเลียนผลงานในแวดวงการศึกษา และระบบที่ใช้ตรวจสอบการลอกเลียนผลงานทางวิชาการที่ใช้อยู่เดิม เช่น Turnitin ซึ่งเป็นระบบจากต่างประเทศมีข้อจำกัดที่สามารถตรวจสอบงานผลงานทางวิชาการได้ดีเฉพาะภาษาอังกฤษเท่านั้น หากตรวจผลงานทางวิชาการภาษาไทยจะเกิดข้อผิดพลาดได้สูง อีกทั้งยังเป็นระบบที่เสียค่าใช้จ่ายในการใช้งาน จึงต้องการพัฒนาระบบตรวจสอบการลอกเลียนผลงานทางวิชาการสำหรับคนไทย เพื่อให้คนไทยได้ใช้งานระบบตรวจสอบที่มีประสิทธิภาพสำหรับภาษาไทย และไม่เสียค่าใช้จ่าย ปัจจุบันมีสถาบันอุดมศึกษาที่ร่วมลงนามบันทึกข้อตกลงความร่วมมือทางวิชาการกว่า 150 สถาบัน [1][2][3]

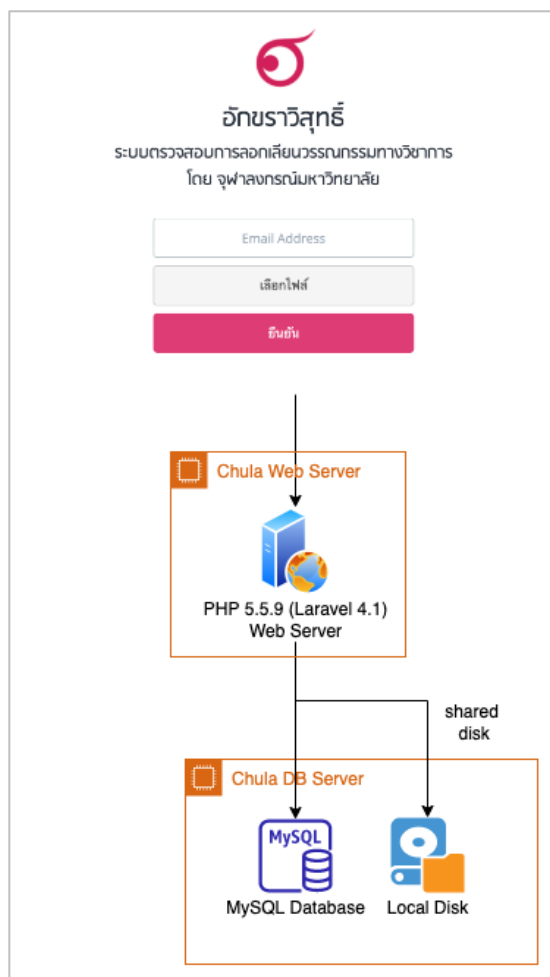


ภาพที่ 1 จำนวนครั้งที่ผลงานทางวิชาการถูกซ้ำซ้อนภายในมหาวิทยาลัยที่ใช้งานระบบอักขราวิสุทธิ์

จากภาพจะเห็นได้ว่าปัญหาการลอกเลียนผลงานในแวดวงศึกษานั้นเป็นไปได้ง่าย โดยมีสาเหตุจากการรับจ้างทำวิทยานิพนธ์ ความเข้าใจผิดเรื่องการลอกเลียนเรื่องผลงานว่าไม่ใช่เรื่องผิด การลอกทำได้ง่าย และขาดความเข้าใจเรื่องการละเมิดลิขสิทธิ์การลอกเลียนผลงาน และการอ้างอิงผลงานผู้อื่นอย่างถูกวิธี [1]

การลอก สามารถเป็นได้หลายวิธี เช่น ลอกทั้งหมด ตัดปะะ คั่นหาและแก้ไขเป็นคำ เรียบเรียงรูปประโยคใหม่ และอื่นๆ โดยระบบของอักขราวิสุทธิ์สามารถตรวจสอบข้อความที่คล้ายกันด้วยการตรวจสอบ

อ้างอิงจากฐานข้อมูลของระบบ ซึ่งมีข้อมูลจากวิทยานิพนธ์ของมหาวิทยาลัยต่าง ๆ วิทยานิพนธ์จากสำนักงานคณะกรรมการการอุดมศึกษา (สกอ.) และบทความที่สามารถสืบค้นได้จากอินเทอร์เน็ต [1]



ภาพที่ 2 Diagram ของระบบปัจจุบัน

สำหรับระบบที่ใช้อยู่ในปัจจุบัน ผู้ใช้สามารถเข้าถึงบริการได้ผ่าน Web browser ซึ่งระบบใช้งานง่าย เข้าใจง่าย เพียงอัปโหลดเอกสารโดยคลิกปุ่ม “เลือกไฟล์” และปุ่ม “ยืนยัน” ระบบอักษราวิสุทธิ์ในปัจจุบันรองรับการตรวจสอบไฟล์เอกสาร 3 สกุล ได้แก่ .doc .docx และ .pdf [4] หลังเสร็จสิ้นการตรวจสอบ ผู้ใช้จะได้รับรายงานผลซึ่งแสดงอัตราส่วนการเข้าช้อนของเอกสาร เทียบกับชุดข้อมูลจากฐานข้อมูลของระบบ โดยสามารถตรวจสอบได้ผ่าน Email และ Web browser เช่นกัน

การให้บริการของระบบอักษราวิสุทธิ์ในปัจจุบัน พัฒนาอยู่บนเครื่องแม่ข่ายของจุฬาลงกรณ์มหาวิทยาลัย ซึ่งเรียกว่า On-Premise Server กล่าวคือ ต้องมีเครื่องคอมพิวเตอร์เปิดทำงานอยู่ตลอดเวลา และเชื่อมต่อกับอินเทอร์เน็ตของมหาวิทยาลัย เมื่อต้องการเพิ่มทรัพยากรของเครื่องแม่ข่ายเพื่อรองรับปริมาณงานที่เพิ่มขึ้น จำเป็นต้องซื้อเครื่องเวิร์คสเตชันมาเพิ่มเป็นเครื่องใหม่

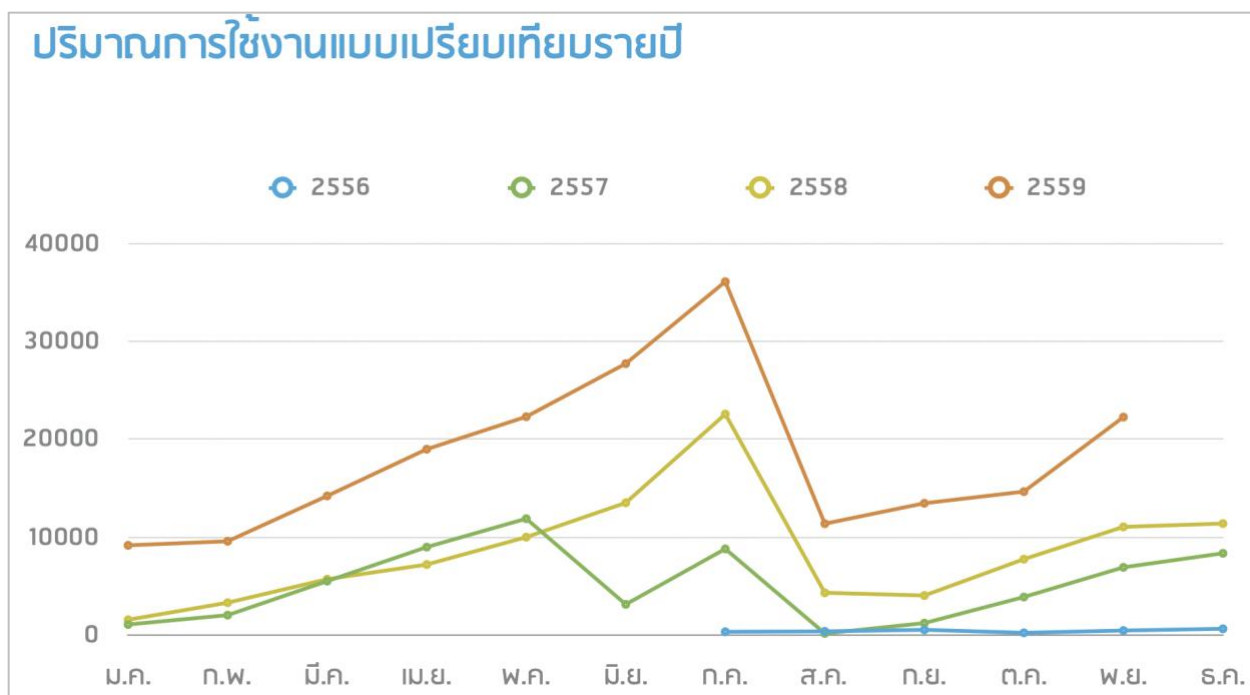
ไม่สามารถเลือกเพิ่มทรัพยากรเท่าที่จำเป็นต้องใช้ซึ่งประหยัดค่าใช้จ่ายมากกว่าและใช้ทรัพยากรได้อย่างเต็มที่ ซึ่งเป็นข้อจำกัดในการเพิ่มหรือลดทรัพยากรสำหรับ

On-Premise Server



ภาพที่ 3 กราฟแสดงปริมาณการใช้งานตั้งแต่เริ่มต้นในระบบอักษราวิสุทธิ (จำนวนครั้งในการตรวจสอบเอกสาร)

จากกราฟจะเห็นว่าจำนวนการใช้งานระบบนั้นสูงขึ้นอย่างต่อเนื่อง ตั้งแต่เปิดใช้งานระบบเมื่อเดือนมกราคม 2556 จนถึงเดือนพฤศจิกายน 2559 [1] และยังคงเพิ่มขึ้นเรื่อย ๆ ทำให้การให้บริการของระบบช้าลงอย่างมาก เนื่องจากระบบเดิมไม่รองรับจำนวนงานที่ถูกส่งมาตรวจสอบในปริมาณที่เกินกว่าการประเมินในตอนต้น อีกทั้งยังไม่สามารถเพิ่มทรัพยากรของระบบได้ทันกับปริมาณการใช้งานที่เพิ่มขึ้นอย่างรวดเร็ว



ภาพที่ 4 กราฟแสดงปริมาณการใช้งานแบบเปรียบเทียบรายปีในแต่ละเดือนของระบบอักษรวิสุทธิ์

จากกราฟจะเห็นได้ว่าการใช้งานของระบบจะเพิ่มขึ้นเป็นอย่างมากในเดือนกรกฎาคม และมีการใช้งานลดน้อยลงทันทีในเดือนสิงหาคม [1] ซึ่งสำหรับระบบเก่า หากจะเตรียมพร้อมระบบให้สามารถรองรับการใช้งานมากในช่วงเดือนกรกฎาคม ในช่วงเดือนสิงหาคมที่การใช้งานค่อนข้างต่ำนั้นระบบจะสูญเสียทรัพยากรไปโดยเปล่าประโยชน์

ด้วยสาเหตุสองประการ คือ จำนวนการใช้งานที่สูงขึ้น และความไม่สม่ำเสมอของการใช้งาน จึงทำให้เราเลือกใช้งานการประมวลผลบนคลาวด์ซึ่งตอบโจทย์ปัญหาที่ได้กล่าวไปข้างต้น ทั้งในเรื่องการรองรับปริมาณงานได้มาก และความยืดหยุ่นการเพิ่มและลดขนาดของระบบตามปริมาณการใช้งานที่แตกต่างกันในแต่ละช่วงเวลา ช่วยประหยัดค่าใช้จ่ายและดูแลระบบด้วยมาตรฐานที่กำหนดโดยผู้ให้บริการคลาวด์

อีกปัญหาที่พบในระบบ คือ ระบบที่ปัจจุบันใช้อยู่ ไม่ได้ถูกพัฒนาอย่างต่อเนื่อง ทำให้ชุดเทคโนโลยีที่ใช้เป็นเวอร์ชันเก่า บางฟีเจอร์ไม่รองรับแล้วในปัจจุบัน ทำให้ระบบไม่สามารถทำงานได้อย่างครบถ้วน มีบางฟังก์ชันที่หยุดใช้งานไป จึงจำเป็นต้องปรับปรุงชุดเทคโนโลยีให้เป็นเวอร์ชันที่ใหม่ขึ้น

วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาและทำความเข้าใจเกี่ยวกับ System Architecture ในระดับการใช้งานจริง
2. เพื่อพัฒนาระบบใหม่ให้มีความยืดหยุ่น สามารถเพิ่มและลดทรัพยากรการทำงานตามความหนาแน่นของการใช้งานได้ดี (Scaling) ลดปัญหาระบบไม่พร้อมใช้งานในช่วงเวลาที่มีความหนาแน่นของการใช้งานมาก
3. เพื่อปรับปรุงชุดของเทคโนโลยีที่ใช้ให้เกิดการบำรุงรักษา (Maintainability) ได้สะดวกขึ้น ง่ายต่อการดูแลและพัฒนาต่อ
4. เพื่อเพิ่ม Testing environment จากแต่เดิมที่มีเพียงแค่ Production environment

ขอบเขตของโครงการ

1. โอนย้ายบางส่วนของระบบอักขรวิสุทธิ จาก On-Premise ไปอยู่บนแพลตฟอร์มคลาวด์ของ Amazon Web Services (AWS)
2. อัปเดตเวอร์ชันของชุดเทคโนโลยีที่ใช้อยู่เดิม สู่เวอร์ชันที่เป็นปัจจุบันมากขึ้น

ประโยชน์ที่คาดว่าจะได้รับ

1. ระบบที่ได้รับการขึ้นระบบแบบ On-Cloud จะสามารถเพิ่ม-ลดทรัพยากรได้สะดวก และรวดเร็วกว่าระบบเดิมที่เป็นอยู่แบบ On-Premise (มี Scalability)
2. ระบบใหม่จะใช้ชุดเทคโนโลยีในเวอร์ชันที่ใหม่ขึ้นและมีความเสถียรแล้ว สามารถรองรับฟีเจอร์ใหม่ๆ ที่พัฒนาขึ้นมาสำหรับใช้บนเวอร์ชันปัจจุบัน มีฟังก์ชันให้เลือกใช้หลากหลายยิ่งขึ้น สะดวกต่อการพัฒนาและดูแลต่อ (Maintainability) อีกทั้งยังเพิ่มความปลอดภัย (Security) และความเชื่อมั่นในระบบ (Reliability) จากผู้ให้บริการคลาวด์ (Cloud Provider)
3. ระบบมีความน่าเชื่อถือมากขึ้น ผู้ใช้งานระบบโดยเฉพาะผู้ที่ต้องการตรวจสอบการลอกเลียนผลงานทางวิชาการสามารถใช้งานได้ด้วยคามมั่นใจ ลดปัญหาที่เกิดขึ้นกับบัณฑิตวิทยาลัยของสถาบันอุดมศึกษาที่ต้องใช้งานระบบ

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้อง

1. Cloud Computing

Cloud Computing คือการใช้ฮาร์ดแวร์และซอฟต์แวร์เพื่อส่งมอบบริการผ่านเครือข่าย (โดยทั่วไปคืออินเทอร์เน็ต) ด้วยการประมวลผลแบบคลาวด์ผู้ใช้สามารถเข้าถึงระบบจัดเก็บข้อมูลและใช้แอปพลิเคชันจากอุปกรณ์ใดก็ได้ที่สามารถเข้าถึงอินเทอร์เน็ตได้ [5][6]

การใช้ Cloud Computing มีข้อดีหลายประการ ช่วยเพิ่มศักยภาพทั้งในด้านของความเร็ว ความยืดหยุ่น และความปลอดภัย รวมไปถึงยังสามารถช่วยลดต้นทุนและประหยัดค่าใช้จ่าย เนื่องจากส่วนใหญ่เป็นการใช้งานแบบ “Pay as you go” กล่าวคือ หากใช้บริการเท่าใด จะจ่ายเงินเท่าที่ใช้ไป ไม่เหมือนกับการซื้อ Server มาติดตั้งที่องค์กรเอง จะเสียค่าใช้จ่ายสูงในการติดตั้งครั้งแรก และค่าใช้จ่ายรายเดือนในการดูแลรักษาระบบ

การจะถือว่าบริการนี้เป็น Cloud Computing ได้ ต้องพิจารณา 3 ส่วน [7][8][9][10][11]

1. คุณลักษณะ (Essential characteristics)

1.1. การบริการที่สั่งการได้เอง (On-demand self-service)

ผู้ใช้สามารถกำหนดและเลือกใช้ทรัพยากรได้ด้วยตนเอง โดยไม่จำเป็นต้องติดต่อกับเจ้าหน้าที่ของผู้ให้บริการ

1.2. การเข้าถึงระบบเครือข่ายแบบกว้างขวาง (Broad network access)

ผู้ใช้บริการต้องเข้าถึงแพลตฟอร์มของผู้ให้บริการจากที่ใดก็ได้จากอุปกรณ์มาตรฐาน เช่น โทรศัพท์มือถือ แท็บเล็ต และคอมพิวเตอร์

1.3. การรวมทรัพยากรไว้ด้วยกัน (Resource pooling)

ทรัพยากรและการประมวลผลของผู้ใช้ถูกรวบรวมไว้ที่ศูนย์กลาง ซึ่งคือผู้ให้บริการคลาวด์ และผู้ใช้หลายคนใช้งานทรัพยากรเดียวกันร่วมกันได้

1.4. ความยืดหยุ่นและรวดเร็ว (Rapid elasticity)

สามารถปรับเปลี่ยนทรัพยากรได้ทันที โดยไม่มีข้อจำกัดด้านปริมาณและระยะเวลา

1.5. บริการที่วัดผลได้ (Measured service)

สามารถวัดปริมาณการใช้งาน และเรียกเก็บค่าใช้จ่ายตามการใช้งานจริง

2. ประเภทของบริการ (Services models)

2.1. On-Premises - ระบบ IT ที่ตั้งอยู่ในสถานที่ของเจ้าของระบบ และมีการดูแลรักษาระบบด้วยตัวเอง

2.2. IaaS – บริการบนคลาวด์ซึ่งจ่ายตามการใช้งานสำหรับบริการต่างๆ เช่น พื้นที่เก็บข้อมูล ระบบเครือข่าย และการจำลองเสมือน

2.3. PaaS - เครื่องมือฮาร์ดแวร์และซอฟต์แวร์ที่มีอยู่ทางอินเทอร์เน็ต มักใช้เพื่อพัฒนาแอปพลิเคชัน

2.4. SaaS - ซอฟต์แวร์ที่พร้อมใช้งาน เป็นบริการแอปพลิเคชันบนระบบคลาวด์

3. รูปแบบการใช้งาน Deployment models)

3.1. Private Cloud - การใช้งานภายในสำหรับองค์กร ผู้ใช้งานเป็นผู้บริหารจัดการระบบ
ปรับเปลี่ยนระบบต่าง ๆ ได้ด้วยตนเอง ผู้ให้บริการมีหน้าที่ติดตั้งและดูแลรักษาเท่านั้น

3.2. Public Cloud - เป็นบริการเข้าถึงข้อมูลผ่านเครือข่ายสาธารณะ
โดยปกติจะใช้งานผ่านผู้ให้บริการซึ่งให้บริการผ่านเครือข่ายสาธารณะ

3.3. Hybrid Cloud - เป็นการผสมผสานกันระหว่าง Private Cloud และ Public Cloud
โดยการนำข้อดีของแต่ละบริการมาใช้งานร่วมกัน

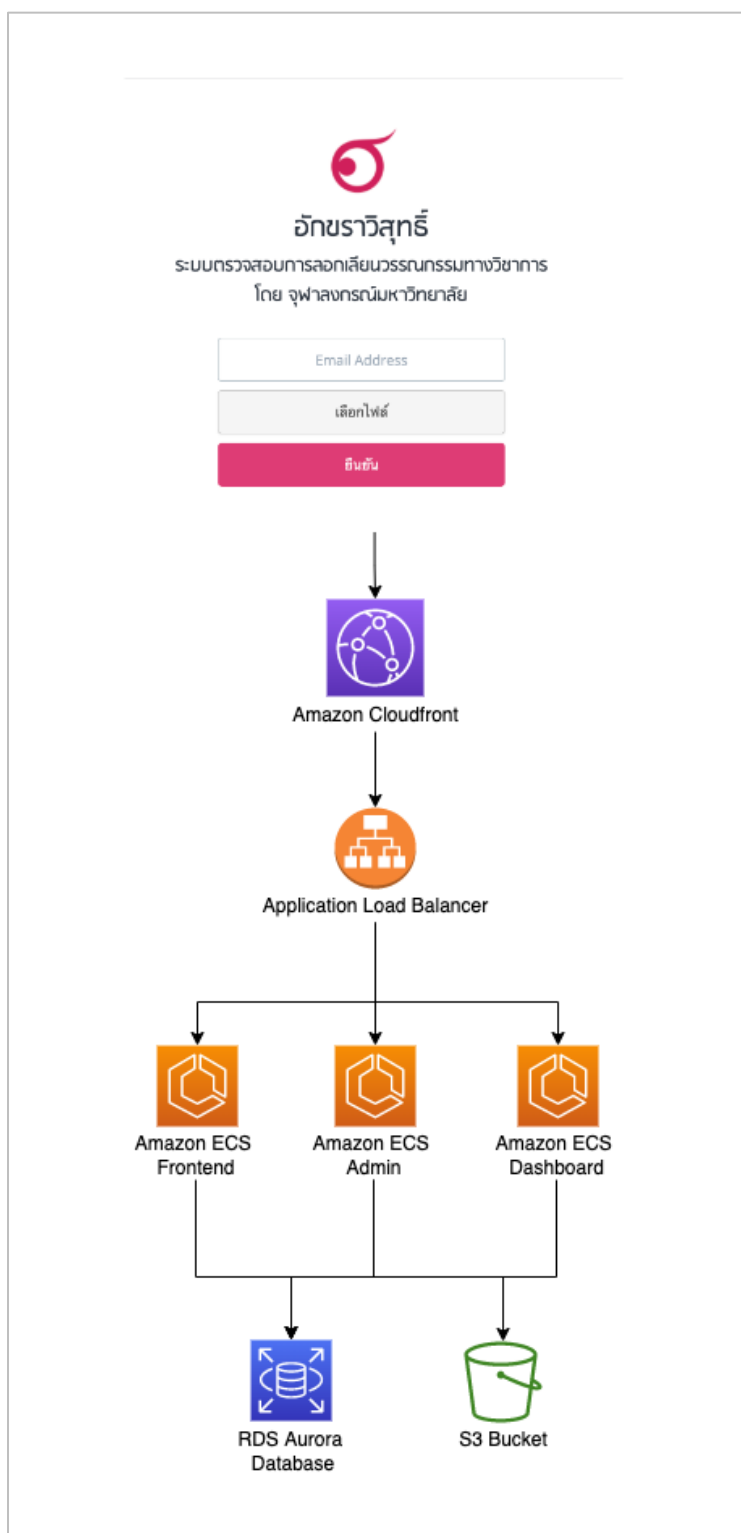
3.4. Community Cloud - บริการ Cloud แบบกลุ่ม ประกอบไปด้วย Private Cloud ตั้งแต่ 2 ผู้ให้บริการ
ขึ้นไป ซึ่งมีการกำหนดข้อตกลงและแชร์ข้อมูลร่วมกัน

2. Infrastructure as a Code (IAC)

ในอดีตนั้น การ Deployment ของระบบ ๆ หนึ่ง ผู้ดูแลระบบ (Administrator) จำเป็นต้องเข้าไป
ปรับตั้งค่าบนแต่ละเครื่องด้วยตนเอง (Manual configuration) แล้วค่อยเชื่อมต่อ (Integrate) แต่ละเครื่อง
ย่อยเข้าหากัน ซึ่งเป็นกระบวนการที่มีความซับซ้อน สูญเสียเวลา และเสี่ยงต่อความผิดพลาดเป็นอย่างมาก
การเข้ามาของแนวคิด DevOps ที่ให้ความสำคัญกับการทำงานร่วมกันระหว่างผู้ดูแลระบบและผู้พัฒนา ทำให้เกิดเครื่องมือที่ช่วยในการเตรียม Environment ทั้ง Development, Testing และ Production นั่นคือ
Infrastructure as a code (IAC)

โดยแนวคิดการทำ IAC นั้น คือ การเปลี่ยนให้ขั้นตอนต่าง ๆ ของการ Deploy ระบบ ให้เป็นแบบ
Automation โดยมี Code เป็นตัวกำหนดค่าต่าง ๆ ในการ Deploy ทำให้การติดตั้ง Infrastructure และการ
ตั้งค่าทุกอย่างอยู่ในรูปแบบของ Code เพื่อให้คอมพิวเตอร์ทำตามคำสั่งตามที่ผู้พัฒนาได้ตั้งค่าเอาไว้ ทำให้มี
ความแม่นยำและตรวจสอบได้ [12] ตัวอย่างเครื่องมือที่ใช้ในการทำ IAC เช่น Terraform, Ansible,
CloudFormation เป็นต้น

บทที่ 3 แนวทางการพัฒนาโครงการ



ภาพที่ 5 Diagram ของระบบที่จะพัฒนาต่อไป

ระบบที่จะพัฒนาในโครงการนี้ ประกอบด้วย 3 ส่วน คือ

1. Front-end website for submission and report view

ใช้สำหรับอัปโหลดไฟล์ที่ต้องการตรวจสอบการลอกเลียนวรรณกรรม และส่งผลการตรวจสอบผ่านทางอีเมล เพื่อระบุส่วนของข้อความที่มีการซ้ำซ้อน คัดลอก จัดแปลงมาจากเอกสารอื่นที่อยู่ในฐานข้อมูลที่มีในระบบ เน้นเรื่องความถูกต้องและความสวยงามในการแสดงผล

2. Administrator Portal

เพื่อให้ผู้ดูแลระบบสามารถจัดการข้อมูลทั้งหมดในระบบได้ ประกอบด้วย

- Submissions - submission ทั้งหมดที่ถูกส่งเข้ามาในระบบและข้อมูลที่เกี่ยวข้องกับ submission ซึ่งมีทั้งการแสดงผลบนเว็บไซต์ และสามารถดาวน์โหลดไฟล์จากระบบได้
- Organization - ดูแลหน่วยงานทั้งหมดในระบบ และข้อมูลเกี่ยวกับหน่วยงาน โดยสามารถเพิ่ม ลด และแก้ไขหน่วยงานได้
- Repositories - สำหรับดูไฟล์ที่เกี่ยวข้องกับหน่วยงานนั้น ๆ
- Applications - สามารถเพิ่ม API key เพื่อให้ผู้ร่วมพัฒนาระบบในหน่วยงานที่ทำ MOU ไว้ สามารถพัฒนาระบบสำหรับส่ง Submission ผ่าน API ได้
- Import Data - เปรียบเสมือนช่อง upload เอกสารเพิ่มในฐานข้อมูลของระบบ
- Users - แสดงข้อมูลผู้ใช้งานในระบบ

3. Portal for TDC (Dashboard)

ใช้สำหรับแสดงผล Dashboard เพื่อให้สำนักงานคณะกรรมการการอุดมศึกษา (สกอ.) สามารถดูข้อมูล Overview ของวิทยานิพนธ์ทั้งหมดที่ได้รับการตรวจสอบได้ ได้แก่ สถิติการใช้งาน จำนวนข้อมูลที่มีในระบบ จำนวนวิทยานิพนธ์ที่มีความซ้ำซ้อน

โดยเมื่อผู้ใช้ ต้องการส่งข้อมูลเข้ามาในระบบ จะต้องผ่านองค์ประกอบ คือ

1. AWS CloudFront เพื่อช่วยเพิ่ม Availability ของ Website
2. Application Load Balancer เพื่อจัดการแบ่งงานภายในระบบให้กับทรัพยากรต่าง ๆ ภายในระบบ
เมื่อมีทรัพยากรที่ใช้ทำงานมากกว่า 1 ชิ้น ให้ทุกทรัพยากรทำงานโดยเท่าเทียมกัน ไม่คงค้างอยู่ที่
จุดใดจุดหนึ่งมากเกินไป
3. RDS Aurora Database สำหรับเป็นฐานข้อมูลของระบบ
4. S3 Bucket ใช้สำหรับการเก็บข้อมูลของงานเขียนที่ผู้ใช้อัปโหลดเข้ามา

ระบบที่จะพัฒนา จะใช้ Amazon Elastic Container Service (Amazon ECS) ในการสร้างและทดสอบระบบ และมีการทำ version control ด้วย GitHub

ในระหว่างการพัฒนา จะใช้เครื่องมือเพื่อช่วยเหลือในการทำงาน ได้แก่

- | | |
|--------------------|---|
| 1. Slack | สำหรับการสอบถาม แลกเปลี่ยนข้อมูล พูดคุยกับคนในทีม |
| 2. Microsoft Teams | สำหรับการประชุม |
| 3. Google Drive | สำหรับจัดเก็บไฟล์เพื่อจัดทำเอกสาร |
| 4. Microsoft Word | สำหรับจัดทำเอกสาร |

บทที่ 4 ผลกระทบทางสังคมของโครงการ

โครงการนี้เกิดขึ้นจากความต้องการของบริษัท อินสปิกา จำกัด (INSPIKA CO.,LTD.) เพื่อโอนย้ายบางส่วนของระบบอักษราวิสุทธิ์ จาก On-Premise ไปยังบริการคลาวด์ของ Amazon AWS และอัปเดตเวอร์ชันของชุดเทคโนโลยีที่เคยใช้อยู่เดิมแล้ว โดยยังคงใช้ชุดเทคโนโลยีเดิม โดยแอปพลิเคชันนี้จะส่งผลกระทบต่อผู้มีส่วนได้ส่วนเสีย (Stakeholder) 4 กลุ่ม ดังนี้

1. ผู้ที่ต้องการตรวจสอบงานเขียน

ภายใต้กลุ่มนี้ ประกอบด้วย ผู้เรียน อาจารย์ในมหาวิทยาลัย นักวิจัย และบุคคลภายนอก เมื่อระบบมีความเสถียรมากขึ้น สามารถใช้ระบบได้อย่างลื่นไหล และระบบจะสามารถประมวลผลได้รวดเร็วมากยิ่งขึ้นจากการขยายตัวตามความหนาแน่นของการใช้งาน จึงทำให้ได้รับผลลัพธ์จากระบบได้เร็วยิ่งขึ้น

2. เจ้าหน้าที่มหาวิทยาลัย

ลดภาระงานที่เกิดจากการที่ระบบเกิดข้อผิดพลาดหรือระบบขัดข้องจากผู้ใช้งานที่มากเกินไป อีกทั้งยังสามารถดูข้อมูลการใช้งานระบบที่เกิดจากมหาวิทยาลัยของตนเองได้โดยไม่ต้องเป็นกังวลเพราะระบบที่มีความเสถียรมากขึ้น

3. เจ้าหน้าที่ สกอ.

ระบบใหม่จะปรับปรุงในส่วนของการแสดงผล เพิ่มรูปแบบของการแสดงผล เห็นข้อมูลเชิงลึกของการใช้งานระบบได้ง่าย ละเอียด และหลากหลายมากขึ้น

4. ผู้ดูแลระบบ

สามารถใช้ฟังก์ชันของระบบได้อย่างครบถ้วน เนื่องจากระบบใหม่แก้ปัญหาเกี่ยวกับ Support ของเครื่องมือที่ใช้พัฒนา ทำให้มีการดูแลที่สะดวกมากยิ่งขึ้น

บทที่ 5 บทสรุปและงานที่จะทำต่อไป

บทสรุป

โครงการอักษรวิสุทธิ์ มีขอบเขตในการทำงานแบ่งเป็น 3 ส่วน ได้แก่ ส่วน Front-end website for submission and report view สำหรับผู้ที่ต้องการตรวจสอบงานเขียนสามารถอัปโหลดงานเขียนสำหรับตรวจสอบได้ ส่วน Administrator Portal สำหรับผู้ดูแลระบบสามารถควบคุมระบบและจัดการข้อมูลภายในระบบได้ง่ายขึ้น และส่วน Portal for TDC (Dashboard) สำหรับเจ้าหน้าที่ สกอ. สามารถเห็นข้อมูลเชิงลึกของงานเขียนในระบบและการใช้งานของระบบได้ง่ายขึ้น โดยโครงการอักษรวิสุทธิ์ในระยะก่อนหน้าได้มีการใช้งานจริงมาก่อนแล้ว

ในปัจจุบัน กำลังอยู่ในการออกแบบโครงสร้างของระบบ เพิ่มเติมในส่วนของรายละเอียดของส่วนย่อยต่าง ๆ เพื่อสามารถพัฒนาทั้ง 3 ส่วนต่อไปได้ ทั้ง ส่วน Front-end website for submission and report view, Administrator Portal, และ Portal for TDC (Dashboard)

งานที่จะทำต่อไป

- ออกแบบโครงสร้างของระบบ (System Architecture) เพิ่มเติมในส่วนของรายละเอียด จากระบบที่ออกแบบไว้แล้วเบื้องต้น
- ศึกษาวิธีการเขียนที่ดี (Best practice) ของชุดโปรแกรมที่เลือกใช้ในการพัฒนา
- เริ่มการทำงานแต่ละส่วนตามแผนที่ได้วางไว้
- ทดสอบระบบก่อนนำไปสู่ Production
- ส่งมอบงาน

บรรณานุกรม

- [1] บริษัท อินส์ไปก้า จำกัด. (2559). เอกสารประกอบการบรรยายเกี่ยวกับระบบอักษรวิสุทธิ์. มหาวิทยาลัยราชภัฏราชชนครินทร์. <http://grad.rru.ac.th/akarawisut/Akarawisut05.pdf>
- [2] บริษัท อินส์ไปก้า จำกัด. (ม.ป.ป.). เกี่ยวกับระบบอักษรวิสุทธิ์. Akarawisut. <https://www.akarawisut.com/>
- [3] บริษัท อินส์ไปก้า จำกัด. (ม.ป.ป.). รายชื่อสถาบันอุดมศึกษาที่ร่วมลงนามบันทึกข้อตกลงความร่วมมือทางวิชาการเรื่อง "การตรวจสอบการลอกเลียนงานวรรณกรรมด้วยระบบอักษรวิสุทธิ์" กับ จุฬาลงกรณ์มหาวิทยาลัย. Akarawisut. <https://www.akarawisut.com/clients.html>
- [4] บริษัท อินส์ไปก้า จำกัด. (ม.ป.ป.). คำถามที่พบบ่อยในระบบอักษรวิสุทธิ์. Akarawisut. <https://www.akarawisut.com/faq.html>
- [5] *Cloud Computing คืออะไร?*. (ม.ป.ป.). Lenovo. <https://www.lenovo.com/th/th/faqs/laptop-faqs/what-is-cloud-computing>
- [6] Chanakan Budrak. (2565, 6 กันยายน). *Cloud Computing คือ อะไร เทคโนโลยีสุดล้ำที่ช่วยให้ธุรกิจของคุณเติบโต*. Openlandscape. <https://blog.openlandscape.cloud/what-is-cloud-computing>
- [7] Cloud Computing. (2565, 3 พฤศจิกายน). In *Wikipedia*. https://en.wikipedia.org/wiki/Cloud_computing
- [8] *About CloudApps*. (ม.ป.ป.). CloudApps. <https://totcloud.com/faq-cloud>
- [9] Admin Netway. (2564, 3 มีนาคม). *Cloud Computing คืออะไร*. NetWay. <https://netway.co.th/kb/blog/cloud-managed-services/cloud-computing-คืออะไร>
- [10] Parichat Phothiin. (2565, 27 กุมภาพันธ์). *เปรียบเทียบความแตกต่าง IaaS vs. PaaS vs. SaaS*. Monster Connect. <https://monsterconnect.co.th/iaas-vs-paas-vs-saas/>
- [11] *On-Cloud vs On-Premise ต่างกันอย่างไร*. (ม.ป.ป.). Coraline. <https://www.coraline.co.th/single-post/on-cloud-vs-on-premise>
- [12] รู้จักกับ *Infrastructure as Code* อีกแนวคิดทางด้าน IT Infrastructure ที่ผู้ดูแลระบบและนักพัฒนาต้องรู้จักเอาไว้. (2559, 12 กุมภาพันธ์). TechTalkThai. <https://www.techtalkthai.com/introduce-infrastructure-as-code/>