การประยุกต์ใช้แนวคิด Clean Architecture สำหรับการพัฒนาระบบ Backend API
Applying the Concept of Clean Architecture for Backend API Development

ประภาวรินทร์ องอาจอุดมรัตน์[1], ษุภากร อนันต์[2], อลงกต กองมณี[3,*], ภานุวัฒน์ เมฆะ[4] และสมนึก สินธุปวน[5]
Praphawarin Ongardudomrat[1], Supakorn Anan[2], Alongkot Gongmanee[3,*], Panuwat Mekha[4] and Somnuek Sinthupuan[5]
สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้ เชียงใหม่ 50290 ประเทศไทย
* ผู้นิพนธ์ประสานงาน: อลงกต กองมณี อีเมล: alongkot@mju.ac.th

บทคัดย่อ:

กรมที่ดินได้พัฒนาระบบบริการค้นหาตำแหน่งรูปแปลงที่ดินผ่านระบบภูมิสารสนเทศทางอินเทอร์เน็ต (LandsMaps) เพื่อเพิ่มประสิทธิภาพการเข้าถึงข้อมูลที่ดินให้มีความถูกต้อง รวดเร็ว และโปร่งใส ทั้งสำหรับประชาชนและเจ้าหน้าที่ภายในหน่วยงาน เพื่อสนับสนุนการบริหารจัดการข้อมูล การติดตามผลการดำเนินงาน และการจัดทำรายงานเชิงวิเคราะห์ จึงได้มีการพัฒนาระบบโปรแกรมประยุกต์เพื่อสนับสนุนการทำงานของเจ้าหน้าที่ภายใต้ชื่อ LandsMaps Backoffice

บทความนี้มีวัตถุประสงค์เพื่อนำเสนอแนวทางการวิเคราะห์ ออกแบบ และพัฒนาระบบส่วนโปรแกรมประยุกต์หลังบ้าน (Backend API) ของ LandsMaps Backoffice โดยประยุกต์ใช้แนวคิด Clean Architecture (Martin, 2018) เพื่อจัดโครงสร้างระบบให้มีการแยกความรับผิดชอบขององค์ประกอบแต่ละส่วนอย่างชัดเจน สถาปัตยกรรมของระบบแบ่งออกเป็น 4 ชั้นหลัก ได้แก่ Domain Layer, Application Layer, Infrastructure Layer และ API Layer ซึ่งช่วยให้ระบบมีความเป็นโมดูล มีความยืดหยุ่น และเอื้อต่อการบำรุงรักษาและขยายระบบในอนาคต

ระบบได้รับการพัฒนาด้วยภาษา C# บนแพลตฟอร์ม .NET 8 Web API (Microsoft Corporation, 2024) โดยใช้ Oracle Database เป็นระบบจัดการฐานข้อมูลหลัก (Oracle Corporation, 2024) และประยุกต์ใช้กลไกการพิสูจน์ตัวตนด้วย JSON Web Token (JWT) เพื่อเสริมสร้างความมั่นคงปลอดภัยของข้อมูล กระบวนการทดสอบดำเนินการด้วยวิธี API Testing ผ่านเครื่องมือ Swagger UI และ Postman เพื่อประเมินความถูกต้องและความครบถ้วนของการให้บริการ

ผลการพัฒนาพบว่าระบบสามารถทำงานได้อย่างถูกต้องตามข้อกำหนด มีความครบถ้วนของฟังก์ชันการทำงาน และมีประสิทธิภาพเพียงพอสำหรับการนำไปใช้งานจริงภายในหน่วยงาน นอกจากนี้ยังสะท้อนให้เห็นถึงความเหมาะสมของการประยุกต์ใช้แนวคิด Clean Architecture ในการพัฒนาระบบสารสนเทศภาครัฐ อันช่วยยกระดับคุณภาพซอฟต์แวร์ในด้านโครงสร้าง ความยั่งยืน และความสามารถในการพัฒนาเพิ่มเติมในระยะยาว

คำสำคัญ: Clean Architecture; ระบบส่วนหลังบ้าน; .NET 8 Web API

Abstract:

The Department of Lands has developed an online Geographic Information System (GIS) for searching land parcel locations, known as LandsMaps, to enhance the accuracy, efficiency, and transparency of land information access for both the public and internal officers. To support data administration, operational monitoring, and analytical reporting, a backend application named LandsMaps Backoffice was subsequently developed.

This paper presents the analysis, design, and development of the Backend API for the LandsMaps Backoffice system by applying the Clean Architecture approach (Martin, 2018). This architectural paradigm emphasizes a clear separation of responsibilities among system components. The system architecture is structured into four primary layers: Domain Layer, Application Layer, Infrastructure Layer, and API Layer, thereby enhancing modularity, flexibility, and long-term maintainability while supporting future scalability.

The system was implemented using C# on the .NET 8 Web API platform (Microsoft Corporation, 2024), with Oracle Database as the primary database management system (Oracle Corporation, 2024). JSON Web Token (JWT) was adopted as the authentication mechanism to strengthen data security. System validation was conducted through API testing using Swagger UI and Postman to evaluate functional correctness and service completeness.

The results demonstrate that the system operates in accordance with the specified requirements, provides complete functional coverage, and delivers sufficient performance for practical deployment within the organization. Furthermore, the findings indicate that the Clean Architecture approach is well-suited for public-sector information system development, contributing to improved software structure, sustainability, and long-term extensibility.

Keywords: Clean Architecture; Backend API; .NET 8 Web API

บทนำ:

กรมที่ดินได้พัฒนาระบบให้บริการค้นหาตำแหน่งรูปแปลงที่ดินผ่านระบบภูมิสารสนเทศทางอินเทอร์เน็ต (LandsMaps) เพื่ออำนวยความสะดวกแก่ประชาชน หน่วยงานภาครัฐ และภาคธุรกิจในการเข้าถึงข้อมูลด้านที่ดินได้อย่างรวดเร็วและมีประสิทธิภาพ อีกทั้งยังสามารถสนับสนุนการบูรณาการข้อมูลร่วมกับหน่วยงานที่เกี่ยวข้อง เช่น องค์กรปกครองส่วนท้องถิ่น และหน่วยงานภาครัฐอื่น ๆ ที่ต้องใช้ข้อมูลเชิงพื้นที่ในการดำเนินงาน อย่างไรก็ตาม ระบบ LandsMaps มีผู้ใช้งานจำนวนมากและมีการแลกเปลี่ยนข้อมูลสารสนเทศอย่างต่อเนื่อง ส่งผลให้เกิดข้อมูลจำนวนมากและมีความซับซ้อนในการบริหารจัดการ ดังนั้นจึงจำเป็นต้องมีระบบสนับสนุนการปฏิบัติงานภายใน ที่มีประสิทธิภาพ เพื่อรองรับการบริหารจัดการสิทธิ์การเข้าถึงข้อมูล การตรวจสอบการทำงานของระบบ และการประสานการทำงานระหว่างหน่วยงานที่เกี่ยวข้อง

ด้วยเหตุนี้ ระบบ LandsMaps Backoffice จึงถูกพัฒนาขึ้นเพื่อทำหน้าที่เป็นศูนย์กลางในการบริหารจัดการผู้ใช้งานและควบคุมความมั่นคงปลอดภัยของระบบในภาพรวม อย่างไรก็ตาม การพัฒนาระบบสารสนเทศระดับองค์กร มักเผชิญกับความท้าทายด้านความซับซ้อนของซอร์สโค้ด ซึ่งมีแนวโน้มเพิ่มขึ้นตามขนาดและขอบเขตของระบบ ส่งผลให้เกิดข้อจำกัดในการบำรุงรักษาระบบ การทดสอบระบบ ตลอดจนความยากลำบากในการปรับเปลี่ยนหรือพัฒนาเทคโนโลยีในอนาคต (Martin, 2018)

ดังนั้น โครงงานวิจัยนี้จึงมุ่งเน้นการออกแบบและพัฒนาระบบส่วนโปรแกรมประยุกต์ฝั่งหลังบ้าน สำหรับระบบบริหารจัดการสิทธิ์ผู้ใช้งาน โดยใช้โครงร่างการพัฒนา .NET 8 (Microsoft Corporation, 2024) เป็นเทคโนโลยีหลัก และประยุกต์ใช้แนวคิด Clean Architecture เป็นแนวทางในการออกแบบสถาปัตยกรรมซอฟต์แวร์ เพื่อช่วยลดการพึ่งพาระหว่างองค์ประกอบของระบบ ส่งเสริมการแยกหน้าที่ความรับผิดชอบอย่างชัดเจน และเพิ่มประสิทธิภาพในการบำรุงรักษาและพัฒนาระบบในระยะยาว อันจะนำไปสู่การยกระดับมาตรฐานการพัฒนาซอฟต์แวร์ภาครัฐให้มีความยั่งยืนและสามารถรองรับการขยายตัวของระบบในอนาคตได้อย่างมีประสิทธิภาพ

วิธีการวิจัยและการพัฒนา:

การพัฒนาระบบหลังบ้าน ดำเนินการโดยอ้างอิงวงจรการพัฒนาซอฟต์แวร์ และใช้หลักการ Clean Architecture ซึ่งแบ่งโครงสร้างระบบออกเป็น 4 ชั้นหลัก ดังแสดงในภาพที่ 1 ดังนี้

1) Domain Layer

เป็นชั้นแกนกลางของระบบ ทำหน้าที่กำหนดแบบจำลองข้อมูลทางธุรกิจ (Business Entities) และกฎเกณฑ์การทำงานหลักของระบบ (Business Rules) ชั้นนี้ถูกออกแบบให้มีความเป็นอิสระจากเทคโนโลยีภายนอกและไม่อ้างอิงไปยังชั้นอื่น เพื่อรักษาความบริสุทธิ์ของตรรกะทางธุรกิจ และเอื้อต่อการนำไปใช้ซ้ำหรือพัฒนาต่อยอดในอนาคต

2) Application Layer

เป็นชั้นที่รับผิดชอบการจัดการกระบวนการทำงานของระบบ (Use Cases) โดยทำหน้าที่ประสานการทำงานระหว่าง Domain Layer และชั้นอื่น ๆ ของระบบ ผ่านองค์ประกอบสำคัญ เช่น Service, Data Transfer Object (DTO) และ Interface เพื่อกำหนดลำดับขั้นตอนของการประมวลผลข้อมูลตามตรรกะทางธุรกิจที่กำหนดไว้ใน Domain Layer

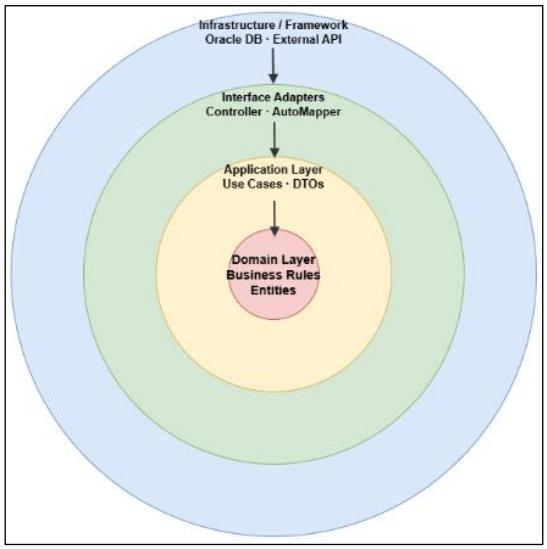
3) Infrastructure Layer

เป็นชั้นที่ทำหน้าที่เชื่อมต่อกับเทคโนโลยีภายนอก เช่น ระบบฐานข้อมูล หรือบริการจากระบบอื่น โดยทำหน้าที่จัดการกระบวนการจัดเก็บและเรียกใช้ข้อมูลของระบบ ทั้งนี้การออกแบบในชั้นดังกล่าวช่วยแยกส่วนการทำงานด้านเทคโนโลยีออกจากตรรกะหลักของระบบ เพื่อให้สามารถปรับเปลี่ยนเทคโนโลยีในอนาคตได้โดยไม่กระทบต่อโครงสร้างธุรกิจหลัก

4) API Layer

เป็นชั้นที่ทำหน้าที่เป็นจุดเชื่อมต่อระหว่างระบบกับผู้ใช้งานหรือระบบภายนอก โดยทำหน้าที่รับคำร้องขอ และส่งผลลัพธ์การประมวลผลกลับไปยังผู้ใช้งานในรูปแบบของบริการโปรแกรมประยุกต์ (API)

ภายใต้แนวคิดของ Clean Architecture การไหลของข้อมูลภายในระบบจะเริ่มจาก API Layer ซึ่งรับคำร้องขอจากผู้ใช้งาน จากนั้นคำร้องขอจะถูกส่งต่อไปยัง Application Layer เพื่อประมวลผลตามกรณีการใช้งานของระบบ โดยใช้กฎเกณฑ์ทางธุรกิจที่กำหนดไว้ใน Domain Layer และหากมีการจัดการข้อมูล ระบบจะติดต่อกับ Infrastructure Layer เพื่อดำเนินการกับฐานข้อมูล ก่อนส่งผลลัพธ์ย้อนกลับผ่าน Application Layer และ API Layer ไปยังผู้ใช้งานตามลำดับ



ภาพที่ 1 โครงสร้างสถาปัตยกรรม Clean Architecture ของระบบ LandsMaps Backoffice

ผลการทดลองและการประเมินผล:

จากการออกแบบและพัฒนาระบบ LandsMaps Backoffice ภายใต้แนวคิด Clean Architecture พบว่าระบบสามารถดำเนินงานได้อย่างมีประสิทธิภาพทั้งในด้านความครบถ้วนของฟังก์ชันและโครงสร้างสถาปัตยกรรมของซอฟต์แวร์

ในด้านความสมบูรณ์ของฟังก์ชันการทำงาน ระบบสามารถให้บริการ Backend API จำนวน 37 Endpoints เพื่อรองรับการจัดการข้อมูลจาก 19 ตารางในฐานข้อมูล และรองรับการควบคุมสิทธิ์การเข้าถึงข้อมูลตามแนวคิด Role-Based Access Control โดยแบ่งผู้ใช้งานออกเป็น 3 ระดับ ได้แก่ ผู้ดูแลระบบส่วนกลาง, หน่วยงานองค์กรปกครองส่วนท้องถิ่น(อปท.) และหน่วยงานภายนอก ทั้งนี้การพิสูจน์ตัวตนและการควบคุมการเข้าถึงระบบถูกดำเนินการผ่านกลไก JSON Web Token (JWT) ซึ่งช่วยเพิ่มความมั่นคงปลอดภัยในการเข้าถึงบริการของระบบ

ในด้านสถาปัตยกรรมซอฟต์แวร์ ผลการวิเคราะห์โครงสร้างการพึ่งพาของซอร์สโค้ด พบว่าระบบสามารถรักษาหลักการ Dependency Rule ของ Clean Architecture ได้อย่างถูกต้อง กล่าวคือ การอ้างอิงระหว่างโมดูลภายในระบบมีทิศทางเข้าสู่ Domain Layer ซึ่งเป็นแกนกลางของตรรกะทางธุรกิจ ส่งผลให้ส่วนที่เกี่ยวข้องกับเทคโนโลยีภายนอก เช่น ชั้น Infrastructure ถูกแยกออกจากตรรกะธุรกิจอย่างชัดเจน แนวทางดังกล่าวช่วยลดการยึดติดระหว่างองค์ประกอบของระบบ และเพิ่มความสามารถในการบำรุงรักษาและพัฒนาระบบในระยะยาว

สำหรับการทดสอบการทำงานของระบบ ได้ดำเนินการผ่านเครื่องมือสำหรับทดสอบบริการ API เช่น Swagger และ Postman ผลการทดสอบพบว่าระบบสามารถประมวลผลคำร้องขอที่เกี่ยวข้องกับการบริหารจัดการผู้ใช้งาน การจัดเก็บบันทึกการทำงานของระบบ และการจัดการข้อมูลประกาศข่าวสารได้อย่างถูกต้องและมีเสถียรภาพ นอกจากนี้ ระบบยังสามารถรองรับการทำงานในสภาพแวดล้อมจำลองของการปฏิบัติงานจริงภายในกรมที่ดินได้อย่างมีประสิทธิภาพ

โดยสรุป ผลการทดลองแสดงให้เห็นว่าแนวทาง Clean Architecture สามารถช่วยเพิ่มความเป็นระเบียบของโครงสร้างซอฟต์แวร์ ลดความซับซ้อนของการพึ่งพาระหว่างโมดูล และสนับสนุนการพัฒนาระบบสารสนเทศภาครัฐที่ต้องการความยืดหยุ่นและสามารถขยายระบบได้ในอนาคต

สรุปผล:

การศึกษาและพัฒนาระบบ LandsMaps Backoffice ในครั้งนี้แสดงให้เห็นว่าการประยุกต์ใช้เทคโนโลยี .NET C# ร่วมกับแนวคิดสถาปัตยกรรม Clean Architecture มีความเหมาะสมต่อการพัฒนาระบบสารสนเทศระดับองค์กรของภาครัฐ โดยโครงสร้างสถาปัตยกรรมดังกล่าวช่วยแยกส่วนความรับผิดชอบขององค์ประกอบภายในระบบอย่างชัดเจน ลดการพึ่งพาระหว่างโมดูล และเพิ่มความยืดหยุ่นในการพัฒนาและบำรุงรักษาระบบในระยะยาว ผลการพัฒนาระบบแสดงให้เห็นว่าระบบสามารถรองรับการบริหารจัดการสิทธิ์ผู้ใช้งานและการควบคุมการเข้าถึงข้อมูลได้อย่างมีประสิทธิภาพ อีกทั้งยังช่วยสร้างโครงสร้างซอร์สโค้ดที่มีความเป็นระเบียบ สามารถรองรับการปรับปรุงเทคโนโลยีหรือการขยายฟังก์ชันการทำงานของระบบในอนาคตได้อย่างเหมาะสม แนวทางดังกล่าวจึงถือเป็นแนวทางหนึ่งที่สามารถสนับสนุนการพัฒนาระบบสารสนเทศภาครัฐให้มีความยั่งยืนและสามารถรองรับการเปลี่ยนแปลงของเทคโนโลยีได้ในระยะยาว

กิตติกรรมประกาศ:

โครงงานคอมพิวเตอร์เรื่อง “โครงการเชื่อมโยงระบบให้บริการค้นหาตำแหน่งรูปแปลงที่ดินด้วยระบบภูมิสารสนเทศทางอินเทอร์เน็ตกับแพลตฟอร์มกลางทะเบียนทรัพย์สิน” สำเร็จลุล่วงไปได้ด้วยดี ด้วยความร่วมมือและการสนับสนุนจากหลายภาคส่วน ผู้จัดทำโครงงานขอขอบพระคุณ บริษัท เซ็นส์ อินโฟ เทค จำกัด ที่ให้ความอนุเคราะห์ด้านเทคนิคและการพัฒนาระบบ ตลอดจนประสานความร่วมมือกับ กรมที่ดิน ในการให้ข้อมูลและแนวทางด้านการจัดการสินทรัพย์ภาครัฐ ซึ่งมีส่วนสำคัญอย่างยิ่งต่อความสำเร็จและความสมบูรณ์ของโครงงาน (บริษัท เซ็นส์ อินโฟ เทค จำกัด, 2568)

ขอขอบพระคุณ อาจารย์อลงกต กองมณี ประธานกรรมการที่ปรึกษาโครงงาน ที่ได้ให้คำแนะนำ ดูแล และให้ความช่วยเหลืออย่างใกล้ชิดตลอดระยะเวลาการดำเนินโครงงาน รวมถึงการตรวจสอบและให้ข้อเสนอแนะอันเป็นประโยชน์ เพื่อปรับปรุงผลงานให้มีคุณภาพดียิ่งขึ้น

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ภานุวัฒน์ เมฆะ และ ผู้ช่วยศาสตราจารย์ ดร.สมนึก สินธุปวน กรรมการโครงงาน ที่ได้ให้เกียรติเป็นคณะกรรมการตรวจสอบความถูกต้องและความครบถ้วนของเนื้อหา พร้อมทั้งให้ข้อเสนอแนะที่มีคุณค่าอย่างยิ่งต่อการพัฒนาโครงงานให้สมบูรณ์ยิ่งขึ้น ผู้จัดทำโครงงานขอแสดงความขอบคุณมา ณ โอกาสนี้

เอกสารอ้างอิง

บริษัท เซ็นส์ อินโฟ เทค จำกัด. (2568). โครงงานระบบ LandsMaps Backoffice สำหรับกรมที่ดิน. เอกสารภายในโครงการ LandsMaps Backoffice, กรมที่ดิน.

Martin, R. C. (2018). Clean architecture: A craftsman's guide to software structure and design. Prentice Hall.

Microsoft Corporation. (2024). ASP.NET Core documentation. https://learn.microsoft.com/en-us/aspnet/core

Oracle Corporation. (2024). Oracle Database 21c documentation. https://docs.oracle.com/en/database/oracle/oracle-database/21