

การพัฒนาระบบบันทึกและตรวจสอบประวัติการทำผิดระเบียบเพื่ออนุญาต เข้าสอบเป็นกรณีพิเศษ กรณีศึกษา มหาวิทยาลัยนอร์ทกรุงเทพ

Special Examination Access History Management System of North Bangkok University

สณัฐภูมิ วิทย์ไพศาล¹, จิรวินัญญ์ ดีเจริญชิตพงษ์²

¹คณะเทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล, มหาวิทยาลัยนอร์ทกรุงเทพ, กรุงเทพมหานคร, 10220

²คณะเทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล, มหาวิทยาลัยนอร์ทกรุงเทพ, กรุงเทพมหานคร, 10220

บทคัดย่อ

ระบบบันทึกและตรวจสอบประวัติการทำผิดระเบียบเพื่ออนุญาตเข้าสอบเป็นกรณีพิเศษ กรณีศึกษา มหาวิทยาลัยนอร์ทกรุงเทพ ถูกพัฒนาขึ้นเพื่ออำนวยความสะดวกในการทำงานของคณะกรรมการคุมสอบส่วนกลาง โดยใช้วงจรชีวิตการพัฒนาระบบ (SDLC) แบบ Waterfall Model ประกอบด้วย 7 ขั้นตอน ได้แก่ Planning, Requirement & Analysis, System Design, System Development, System Testing, System Deployment และ Maintenance

ฟังก์ชันการทำงานประกอบไปด้วยการบันทึกประวัติการทำผิดระเบียบก่อนเข้าสอบของนักศึกษา, การสรุปผลจำนวนนักศึกษาที่มาขอใบอนุญาตเข้าห้องสอบกรณีพิเศษ, การเพิ่มข้อมูลส่วนต่าง ๆ ในฐานข้อมูล โดยระบบได้ถูกพัฒนาจากโปรแกรม Visual Studio Code และมีการเชื่อมต่อฐานข้อมูลด้วย MongoDB

ประชากรตัวอย่าง คือ คณาจารย์ภายในมหาวิทยาลัยนอร์ทกรุงเทพ โดยสุ่มตามคณะภายในมหาวิทยาลัยนอร์ทกรุงเทพ จำนวน 30 คน โดยมีทั้งหมด 7 คณะ ได้แก่ คณะบริหารธุรกิจ คณะเทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล คณะศิลปศาสตร์ คณะรัฐศาสตร์ คณะนิเทศศาสตร์ คณะศึกษาศาสตร์ และคณะพยาบาลศาสตร์ ใช้ Google Form ในการเก็บข้อมูลแบบสอบถามและการประเมินคุณภาพของระบบ

คำหลัก: ผิดระเบียบ, ห้องสอบ, ระบบบันทึก, การตรวจสอบประวัติ

Abstract

The Disciplinary Record and Special Examination Authorization System: A Case Study of North Bangkok University was developed to facilitate and streamline the operations of the Central Examination Committee. The system was built following the Systems Development Life Cycle (SDLC) Waterfall Model, which encompasses seven essential stages: Planning, Requirement & Analysis, System Design, System Development, System Testing, System Deployment, and Maintenance.

The core functionalities of the system include recording student disciplinary history regarding regulatory violations prior to exams, summarizing the total number of students requesting special permission to enter examination rooms, and managing various data entries within the database. From a technical perspective, the system was developed using Visual Studio Code and utilizes MongoDB for database management.

The target population for this study consisted of faculty members at North Bangkok University. A sample of 30 participants was selected through random sampling across seven faculties, including the Faculty of Business Administration, Information Technology and Digital Innovation, Liberal Arts, Political Science, Communication Arts, Education, and Nursing. Data collection and system quality evaluation were conducted through questionnaires distributed via Google Forms.

Keywords: Violation, Examination room, Recording system, Background check

ความเป็นมาและความสำคัญของปัญหา

ในช่วงเวลาสอบแต่ละครั้งจะมีนักศึกษาที่ทำผิดระเบียบ เช่น ไม่มีเนคไท ไม่มีเข็มขัด กางเกงผิดระเบียบ เป็นต้น โดยการที่ให้เข้าห้องสอบนั้นต้องมีใบอนุญาตเข้าห้องสอบกรณีพิเศษซึ่งออกโดยกรรมการคุมสอบส่วนกลาง ซึ่งทำหน้าที่เป็นใบขออนุญาตเข้าห้องสอบในกรณีที่นักศึกษาทำผิดระเบียบโดยแสดงให้เห็นกับกรรมการคุมสอบเพื่ออนุมัติให้เข้าห้องสอบ (ธันยรัตน์ ชัยวันดี และ ณัฐพัชญ์ ศรีราชจันทร์, 2558)

ปัจจุบันระบบการจัดเก็บข้อมูลการอนุญาตเข้าห้องสอบยังคงใช้รูปแบบการบันทึกความผิดระเบียบลงบนกระดาษ ซึ่งก่อให้เกิดข้อจำกัดสำคัญด้านประสิทธิภาพ ทั้งความยากลำบากในการสืบค้นประวัติเพื่อตรวจสอบความผิดซ้ำในอนาคต ความล่าช้าในขั้นตอนการพิจารณาอนุมัติที่ต้องรอการลงนามจากประธานหรือผู้แทนกรรมการกลางแบบเป็นลายลักษณ์อักษร ตลอดจนการขาดระบบรวบรวมข้อมูล

ในภาพรวม ทำให้ไม่สามารถสรุปรายงานสถิติการกระทำผิดระเบียบของนักศึกษาทั้งในการสอบกลางภาค และปลายภาคได้อย่างถูกต้องและครบถ้วน

จากปัญหาดังกล่าว ทางผู้จัดทำได้มีแนวคิดในการพัฒนาระบบบันทึกและตรวจสอบประวัติการทำผิดระเบียบเพื่ออนุญาตเข้าสอบเป็นกรณีพิเศษ กรณีศึกษา มหาวิทยาลัยนอร์มัลกรุงเทพฯขึ้นมาเพื่อให้คณะกรรมการคุมสอบส่วนกลางสามารถรอกข้อมูลต่าง ๆ บนใบอนุญาตการเข้าห้องสอบกรณีพิเศษลงในระบบและสามารถตรวจสอบประวัติย้อนหลังที่ได้ทำการบันทึกไว้ในระบบ ทำให้สามารถบริหารจัดการได้อย่างสะดวกรวดเร็วมากขึ้นด้วย (สนั่น หวานแท้, 2553)

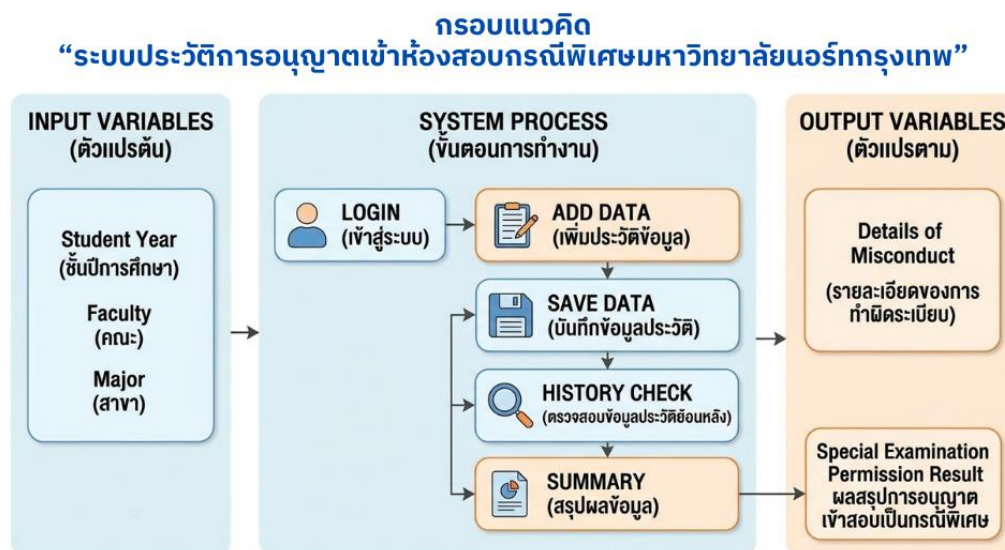
วัตถุประสงค์ของงานวิจัย

1. เพื่อออกแบบและพัฒนาระบบดิจิทัลสำหรับการบันทึกและตรวจสอบประวัติการทำผิดระเบียบเข้าสอบ
2. เพื่อเพิ่มประสิทธิภาพและลดระยะเวลาในกระบวนการพิจารณาอนุมัติการเข้าห้องสอบกรณีพิเศษ
3. เพื่อสร้างระบบรายงานสถิติการกระทำผิดระเบียบสำหรับการวางแผนและปรับปรุงการทดสอบ

ประโยชน์ที่คาดว่าจะได้รับ

1. เพิ่มประสิทธิภาพของระบบบริหารจัดการการสอบและลดขั้นตอนการทำงานที่ของคณะกรรมการคุมสอบส่วนกลาง
2. เพิ่มประสิทธิภาพในการตรวจสอบและติดตามพฤติกรรมการทำผิดระเบียบของนักศึกษา
3. ช่วยสำหรับการตัดสินใจให้ผู้บริหารใช้รายงานสถิติการกระทำผิดระเบียบที่หามาตรการป้องกันและปรับปรุงกระบวนการทำงานเพื่อการตัดสินใจ

กรอบแนวคิด



ภาพ 1 กรอบแนวคิด

กระบวนการทำงานเริ่มต้นจากคณะกรรมการคุมสอบส่วนกลางต้องเข้าสู่ระบบก่อนจะทำการเพิ่มข้อมูลประวัติการทำผิดระเบียบ หลังจากนั้นทำการกรอกข้อมูลต่าง ๆ ที่อยู่บนใบอนุญาตเข้าห้องสอบกรณีพิเศษลงในระบบ พอทำการกรอกข้อมูลครบถ้วนก็บันทึกข้อมูลลงในระบบ ถ้าต้องการค้นหาข้อมูลประวัติย้อนหลังก็สามารถตรวจสอบได้ทันที และสามารถสรุปผลข้อมูลภาพรวมว่าคณะและสาขาไหน, ชั้นปีการศึกษาไหนมีนักศึกษาทำผิดระเบียบจำนวนกี่คน ตัวแปรนำ คือ ชั้นปี, คณะและสาขา และตัวแปรตาม คือ รายละเอียดการทำผิดระเบียบ

วิธีดำเนินการวิจัย

สำหรับวิธีการดำเนินการวิจัยเรื่อง “การพัฒนาบบบันทึกและตรวจสอบประวัติการทำผิดระเบียบเพื่ออนุญาตเข้าสอบเป็นกรณีพิเศษ กรณีศึกษา มหาวิทยาลัยนอร์ทกรุงเทพ” ใช้วงจรชีวิตการพัฒนาบระบบ (SDLC “Software Development Life Cycle”) แบบ Waterfall Model ซึ่งเป็นกรอบงานที่ใช้ในการวางแผน, พัฒนา, ทดสอบ, ปรับใช้และปรับปรุงระบบ โดยแบ่งโครงสร้างออกเป็น 7 ขั้นตอนหลัก ซึ่งช่วยให้การพัฒนาบระบบเป็นระเบียบและมีประสิทธิภาพ (Shirley Radack, 2009) โดยขั้นตอนต่าง ๆ มีดังนี้

ขั้นตอนที่ 1: Planning

เป็นการวางแผนในการพัฒนาซอฟต์แวร์ในขั้นตอนนี้จะวางแผนในงานวิจัยว่าจะใช้ระยะเวลาในการพัฒนาบระบบเท่าไร, วางแผนขอบเขตการศึกษา, วางแผนการใช้เครื่องมือในการพัฒนา

ระบบว่าใช้อะไรบ้าง, จำนวนบุคลากรที่จะใช้พัฒนาระบบ, แบ่งหน้าที่เพื่อทำส่วนที่เกี่ยวข้องกับการพัฒนาระบบ, วางแผนเพื่อเก็บ Requirement, Design Software และการทดสอบซอฟต์แวร์ การมีข้อมูลที่ละเอียดจะช่วยให้การวางแผนงานได้อย่างแม่นยำจะช่วยลดความเสี่ยงของการพัฒนาซอฟต์แวร์และสามารถพัฒนาสำเร็จตามวัตถุประสงค์ หลังจากที่ทำวางแผนแล้ว จะได้ดังนี้

1. ระยะเวลาในการพัฒนาระบบ: มิถุนายน 2568 ถึง กุมภาพันธ์ 2569
2. ขอบเขตการศึกษา: ภายในมหาวิทยาลัยนอร์ทกรุงเทพ
3. เครื่องมือในการพัฒนาระบบ: คอมพิวเตอร์ส่วนบุคคลที่มีสเปคดังนี้
 - 3.1 หน่วยประมวลผล 6 คอร์ 12 เธรด
 - 3.2 แรมขนาด 16 กิกะไบต์
 - 3.3 การ์ดจอค่าย Nvidia Geforce GTX 1650
 - 3.4 โปรแกรมที่ใช้ในการพัฒนามีดังนี้
 - 3.4.1 Visual Studio Code
 - 3.4.2 MongoDB
 - 3.4.3 React (HTML)
 - 3.4.4 Typescript
 - 3.4.5 Express JS
 - 3.4.6 Vite
 - 3.4.7 Jotai
 - 3.4.8 Tailwind CSS
4. จำนวนบุคลากรในการพัฒนาระบบ: 1 คน

ขั้นตอนที่ 2: Requirement & Analysis

ในขั้นตอนนี้จะทำการเก็บรวบรวม Requirement จากผู้มีส่วนเกี่ยวข้องเพื่อเข้าใจความต้องการจากผู้ใช้งานและการทำงานของระบบ โดยอาจจัดทำอยู่ในรูปแบบ Prototyping เพื่อให้ผู้ใช้งานสามารถเข้าใจการทำงานของระบบได้ง่ายขึ้น หลังจากการรวบรวม Requirement เสร็จแล้วตรวจสอบ Requirement เพื่อใช้ในการ Design Software และประเมินความเสี่ยงและข้อจำกัดระบบที่อาจเกิดขึ้นได้ หลังจากที่ได้เก็บ Requirement แล้วสามารถสรุปได้ดังนี้ คือ ระบบสามารถตรวจสอบประวัติการทำผิดระเบียบและอนุมัติโดยประธานหรือผู้แทนได้ทันที เพื่อให้ให้นักศึกษานำผลการอนุมัติไปแสดงต่อกรรมการคุมสอบ โดยกรรมการคุมสอบต้องดำเนินการอนุมัติรับทราบในระบบให้เสร็จสิ้นก่อน จึงจะอนุญาตให้นักศึกษาเข้าห้องสอบได้ตามขั้นตอน

ขั้นตอนที่ 3: System Design

นำ Requirement ที่ได้มาทำการออกแบบ เช่น Design สถาปัตยกรรมที่จะใช้ (Architects), ส่วนต่อประสานผู้ใช้งาน (User Interface), ภาษาที่จะใช้ในการพัฒนาซอฟต์แวร์, Design ฐานข้อมูล Security และ Network เป็นต้น ผลลัพธ์ที่ได้จากขั้นตอนนี้จะได้ System Design Specification ซึ่งสามารถให้ผู้มีส่วนเกี่ยวข้องได้ Review และสามารถให้คำแนะนำได้

ขั้นตอนที่ 4: System Development

หลังจากที่ได้ออกแบบ System Design แล้วขั้นตอนต่อไปคือการพัฒนาตัวระบบโดยมีขั้นตอนดังนี้

4.1 เริ่มพัฒนา Frontend โดยใช้ HTML (Ankush Sharma and Aakanksha, 2018), Vite และ Typescript (Gavin Bierman, Martin Abadi and Mads Torgersen, 2014), ในการขึ้นโครงสร้างหลักเพื่อกำหนดองค์ประกอบหลักของหน้าจอรระบบ เช่น การจัดวางตำแหน่ง Components ต่าง ๆ ในระบบ ตัวอย่างเช่นปุ่มกดต่าง ๆ ของระบบ

4.2 นำ Tailwind CSS มาใช้ในการตกแต่งหน้าระบบ เช่น สี, ขนาด, น้ำหนักของ ฟอนต์ตัวอักษร, การสร้างกรอบตารางข้อมูล (Sari Azhariyah and Muhammad Mukhlis, 2024)

4.3 นำ Jotai มาใช้สำหรับการอัปเดตข้อมูลในแต่ละจุดของระบบมีประสิทธิภาพสูง และไม่ซับซ้อน ช่วยให้มีความรวดเร็วในการค้นหาข้อมูล (Kim Tran, 2023)

4.4 หลังจากพัฒนา Frontend เสร็จแล้ว ขั้นตอนต่อไปเป็นการพัฒนา Backend ของระบบ โดยใช้ MongoDB และ Express JS ในการพัฒนา Backend ของระบบ

4.4.1 สร้าง Table ใน MongoDB สำหรับเก็บข้อมูลต่าง ๆ เช่น ชื่อ-นามสกุล ของนักศึกษา, รายวิชา, คณะและสาขาวิชา, รายละเอียดการฝึกระเบียบ (Cornelia GyÖrödi, Robert GyÖrödi, George Pecherle and Andrada Olah, 2015)

4.4.2 สร้าง API สำหรับการรับส่งข้อมูลระหว่าง Frontend และ Backend ใน Express JS

4.5 หลังจากพัฒนาส่วนของ Backend เสร็จแล้ว ทำการเชื่อมต่อระหว่าง Frontend และ Backend เข้าด้วยกัน

ขั้นตอนที่ 5: Testing

การทดสอบระบบจะใช้วิธีการทดสอบระบบแบบ Black Box (N. Srinivas and J. Dondeti, 2012) เน้นการตรวจสอบฟังก์ชันการทำงานของระบบโดยไม่ต้องสนใจ Code หรือโครงสร้าง ภายในของโปรแกรมนั้น ผู้ทดสอบจะต้องใส่ข้อมูลเข้าไปและตรวจสอบผลลัพธ์ที่ได้ว่าตรงตามที่คาดหวังหรือไม่ เทคนิคนี้ช่วยให้สามารถระบุข้อบกพร่องและปัญหาการทำงานของระบบได้อย่างมีประสิทธิภาพ แนวทางที่ใช้ในการวิจัยนี้มีดังนี้:

- การทดสอบฟังก์ชัน (Functional Testing): การออกแบบการทดสอบจากมุมมองโดยรวมเพื่อตรวจสอบฟังก์ชันและความสามารถของระบบ โดยทดสอบว่าฟังก์ชันที่พัฒนานั้นทำงานตามที่คาดหวังหรือไม่ ในที่นี้จะทดสอบฟังก์ชันต่าง ๆ ภายในระบบ เช่น การเข้าสู่ระบบ, การเพิ่มข้อมูล, การค้นหาข้อมูล เป็นต้น

- การทดสอบเข้า-ออกข้อมูล (Input/Output Testing): ขั้นตอนนี้ทดสอบว่าระบบสามารถรับข้อมูลเข้าและแสดงผลลัพธ์ออกมาได้อย่างถูกต้องตามที่คาดหวัง โดยใช้ชุดข้อมูลที่กำหนดไว้ เช่นการแสดงผลประวัติของนักศึกษาแต่ละคนว่าแสดงครบถ้วนหรือไม่

- การทดสอบประสิทธิภาพ (Performance Testing): การประเมินประสิทธิภาพและความสามารถของระบบในสถานการณ์ต่าง ๆ เช่น การทดสอบความเร็วในการโหลดหน้าเว็บหรือประสิทธิภาพของฟังก์ชันที่ใช้งานบ่อย

- การทดสอบความปลอดภัย (Security Testing): ขั้นตอนนี้เกี่ยวข้องกับการทดสอบระบบเพื่อตรวจหาจุดอ่อนที่อาจนำไปสู่อาชญากรรมไซเบอร์, การแฮคหรือการรั่วไหลของข้อมูลเพื่อให้แน่ใจว่าระบบนั้นปลอดภัยต่อการใช้งาน

ขั้นตอนที่ 6: Deployment

เมื่อพัฒนาระบบจนเสร็จพร้อมใช้งานก็ทำการ Deploy ระบบให้ผู้ใช้งานทั่วไปใช้ระบบ โดยในขั้นตอนนี้จะเริ่มมีการแสดงความคิดเห็นต่อระบบที่ได้พัฒนาเพื่อนำมาปรับปรุงพัฒนาระบบต่อไป

ขั้นตอนที่ 7: Maintenance

เมื่อระบบได้ผ่านการทดสอบแล้วก็จะมีการ Deploy เพื่อให้ผู้ใช้งานได้ใช้งานระบบจริง ในระหว่างที่ผู้ใช้งานระบบอยู่นั้นก็อาจจะเกิดปัญหาต่าง ๆ ได้ เช่น ต้องการ Feature เพิ่มเติม, ระบบทำงานไม่เป็นไปตามที่ความคาดหวัง, ระบบมี Bug หรือการ Deploy ครั้งแรกอาจมีข้อจำกัดอะไรบางอย่าง เมื่อเกิดปัญหาเหล่านี้เกิดขึ้นก็ต้องมีการแก้ไข, ปรับปรุงระบบเพื่อให้ผู้ใช้งานพึงพอใจมากที่สุด ซึ่งในการแก้ไขก็อาจจะกลับไปเริ่มในขั้นตอนที่ 1 ถึง 6 ใหม่ (New Development Cycle)

กลุ่มประชากรตัวอย่าง

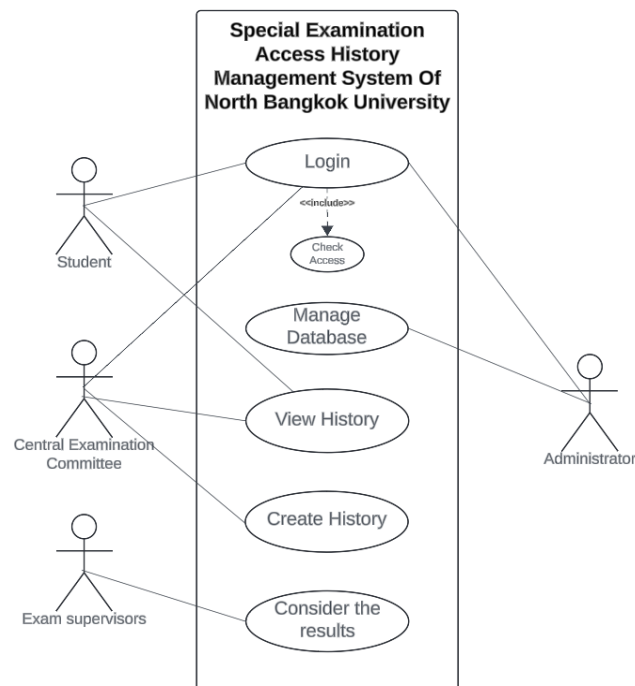
การวิจัยนี้จะใช้กลุ่มประชากรตัวอย่าง จำนวน 30 คน โดยเป็นคณาจารย์ภายในมหาวิทยาลัยนอร์ทกรุงเทพ โดยวิธีการสุ่มคือสุ่มตามคณะภายในมหาวิทยาลัย โดยมีทั้งหมด 7 คณะ ได้แก่ คณะบริหารธุรกิจ คณะเทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล คณะศิลปศาสตร์ คณะรัฐศาสตร์ คณะนิเทศศาสตร์ คณะศึกษาศาสตร์ และคณะพยาบาลศาสตร์

ผลการวิจัย

ในงานวิจัยนี้ ผู้จัดทำนำเสนอผลการวิจัยที่แบ่งออกเป็น 3 ส่วน ได้แก่

ส่วนที่ 1: การวิเคราะห์และออกแบบระบบ

การวิเคราะห์และออกแบบระบบประวัติการอนุญาตเข้าห้องสอบกรณีพิเศษ โดยใช้แผนภาพ Use Case Diagram เพื่อแสดงการออกแบบระบบดังภาพ 2



ภาพ 2 Use Case Diagram

คำอธิบายกรณีการใช้งานสำหรับระบบประวัติการอนุญาตเข้าห้องสอบกรณีพิเศษ (Dr. S.S.Mule and Mr.Yashwant Waykar, 2015) มีดังนี้

1. Login

ผู้ใช้ทั่วไปและผู้ดูแลระบบ: สามารถเข้าสู่ระบบได้โดยใช้ชื่อบัญชีผู้ใช้และรหัสผ่าน

2. Check Access

ระบบ: ระบบจะทำการตรวจสอบสิทธิ์การเข้าถึงของระบบ โดยแบ่งเป็น 2 ระดับ คือ ระดับผู้ใช้ทั่วไปและระดับผู้ดูแลระบบ

3. Manage Database

ผู้ดูแลระบบ: สามารถจัดการข้อมูลต่าง ๆ ภายในฐานข้อมูล เช่น เพิ่ม แก้ไขและลบข้อมูลส่วนต่าง ๆ เพื่อเตรียมความพร้อมของระบบสำหรับส่วนของผู้ใช้ทั่วไป

4. View History

ผู้ใช้ทั่วไป: สามารถดูประวัติย้อนหลังได้

5. Create History

ผู้ใช้ทั่วไป: สามารถสร้างประวัติการอนุญาตเข้าห้องสอบได้

6. Consider the result

ผู้ใช้ทั่วไป: สามารถพิจารณาการอนุมัติการขอเข้าห้องสอบได้

ส่วนที่ 2: การพัฒนาระบบ



ภาพ 3 และ 4 ออกแบบระบบในส่วนของผู้ใช้งานทั่วไปผ่านโปรแกรม Figma



ภาพ 5 และ 6 ออกแบบระบบในส่วนของผู้ดูแลผ่านโปรแกรม Figma

ส่วนที่ 3: การประเมินประสิทธิภาพของระบบประวัติการอนุญาตเข้าห้องสอบกรณีพิเศษ

ในการประเมินระบบนั้นจะใช้ค่าเฉลี่ย (Mean) และค่าเบี่ยงเบนมาตรฐาน (Standard Deviation / S.D.) การคำนวณค่าเฉลี่ยมาจากเกณฑ์การประเมินด้านต่าง ๆ เช่น การเข้าสู่ระบบ, ความสะดวกในการสร้างข้อมูล, ความถูกต้องของการเรียกดูข้อมูล, การประเมินการออกแบบระบบ, ประสิทธิภาพในการจัดการข้อมูล, ความถูกต้องของการแบ่งระดับสิทธิ์การเข้าถึงและความปลอดภัยในการป้องกันการโจมตี โดยเกณฑ์คะแนนจะแบ่งเป็นตามนี้ (Dr. S.S.Mule and Mr.Yashwant Waykar, 2015):

- 4.50-5.00 หมายถึง “ดีมาก”
- 3.50-4.49 หมายถึง “ดี”
- 2.50-3.49 หมายถึง “ปานกลาง”
- 1.50-2.49 หมายถึง “พอใช้”
- 1.00-1.49 หมายถึง “ควรปรับปรุง”

รายการประเมิน	ผลการประเมิน		
	\bar{X}	S.D.	แปลผล
1. การเข้าสู่ระบบ	3.25	0.23	ดี
2. ความสะดวกในการสร้างข้อมูล	3.78	0.36	ดี
3. ความถูกต้องของการเรียกดูข้อมูล	3.5	0.57	ดี
4. การประเมินการออกแบบระบบ	3.7	0.58	ดี
5. ประสิทธิภาพในการจัดการข้อมูล	3.28	0.45	ดี
6. ความถูกต้องของการแบ่งระดับสิทธิ์การเข้าถึง	3.31	0.39	ดี
7. ความปลอดภัยในการป้องกันการโจมตี	3.35	0.50	ดี
ประสิทธิภาพโดยรวม	3.45	0.48	ดี

จากตาราง 1 สามารถเห็นได้ว่าประสิทธิภาพโดยรวมของระบบประวัติการอนุญาตเข้าห้องสอบกรณีพิเศษถูกประเมินผล โดยมีค่าเฉลี่ย (\bar{X}) เท่ากับ 3.55 และส่วนเบี่ยงเบนมาตรฐาน (SD) เท่ากับ 0.43 เมื่อแยกประสิทธิภาพตามหมวดหมู่:

- การเข้าสู่ระบบ (Log In) มีคะแนนเฉลี่ย 3.25 และ S.D. เท่ากับ 0.23
- ความสะดวกในการสร้างข้อมูล (Ease of data creation) มีคะแนนเฉลี่ย 3.78 และ S.D. เท่ากับ 0.36
- ความถูกต้องของการเรียกดูข้อมูล (Accuracy of data retrieval) มีคะแนนเฉลี่ย 3.5 และ S.D. เท่ากับ 0.57
- การประเมินการออกแบบระบบ (System design evaluation) มีคะแนนเฉลี่ย 3.7 และ S.D. เท่ากับ 0.58

- ประสิทธิภาพในการจัดการข้อมูล (Efficiency in data management) มีคะแนนเฉลี่ย 3.28 และ S.D. เท่ากับ 0.45
- ความถูกต้องของการแบ่งระดับสิทธิ์การเข้าถึง (The accuracy of access level classification) มีคะแนนเฉลี่ย 3.31 และ S.D. เท่ากับ 0.39
- ความปลอดภัยในการป้องกันการโจมตี (Security in preventing attacks) มีคะแนนเฉลี่ย 3.35 และ S.D. เท่ากับ 0.50

สรุปผลการวิจัย

ปัจจุบันระบบสามารถรองรับฟังก์ชันการทำงานพื้นฐานตามวัตถุประสงค์หลักได้เป็นอย่างดี อย่างไรก็ตาม ในส่วนของการออกแบบส่วนติดต่อผู้ใช้งาน (User Interface) ยังคงอยู่ในขั้นตอนการปรับปรุงเพื่อให้มีความสมบูรณ์และเป็นมิตรต่อผู้ใช้งานมากยิ่งขึ้น แต่โดยภาพรวมระบบยังคงสามารถปฏิบัติงานได้ตามเกณฑ์ที่กำหนด

ข้อจำกัดในการดำเนินงานคืองานวิจัยนี้มีทรัพยากรบุคคลในด้านการพัฒนาระบบเพียง จำนวน 1 ท่าน ส่งผลให้ระยะเวลาในการดำเนินการและพัฒนาฟังก์ชันในบางส่วนเกิดความล่าช้ากว่าแผนงานที่วางไว้เล็กน้อย

อภิปรายผล

การศึกษานี้มีวัตถุประสงค์เพื่อออกแบบและพัฒนาระบบบันทึกและตรวจสอบประวัติการทำผิดระเบียบเพื่ออนุญาตเข้าสอบเป็นกรณีพิเศษ กรณีศึกษา มหาวิทยาลัยนอร์ทกรุงเทพ ผู้จัดทำได้ทำการออกแบบและพัฒนาระบบที่เรียกว่าระบบประวัติการอนุญาตเข้าห้องสอบกรณีพิเศษ มหาวิทยาลัยนอร์ทกรุงเทพ โดยใช้กรอบวงจรชีวิตการพัฒนาระบบ (SDLC) โดยครอบคลุมขั้นตอนต่อไปนี้: Planning, Requirement & Analysis, System Design, System Development, System Testing, System Deployment และ Maintenance ซึ่งสอดคล้องกับการวิจัยของ Henocho Juli Christanto และ Yerik Afrianto Singgalen ซึ่งการศึกษานี้แสดงให้เห็นว่าการใช้กรอบวงจรชีวิตการพัฒนาระบบ (SDLC) สามารถพัฒนาระบบได้อย่างมีประสิทธิภาพและเรียบง่าย

ข้อเสนอแนะ

ในระยะถัดไป จะดำเนินการพัฒนาความสมบูรณ์ของระบบและปรับปรุงประสิทธิภาพเชิงเทคนิค เช่น การพัฒนาฟังก์ชันเพิ่มเติมที่ไม่ได้ใส่เข้ามาในระบบ และ ปรับปรุงฟังก์ชันเดิมให้ดียิ่งขึ้น

กิตติกรรมประกาศ

คณะผู้จัดทำขอแสดงความขอบคุณต่อคณาจารย์และบุคลากรทุกท่าน สำหรับคำแนะนำและการสนับสนุนที่มีคุณค่าตลอดการดำเนินงาน ความอนุเคราะห์ของท่านมีส่วนสำคัญยิ่งในการขับเคลื่อนการศึกษานี้ให้สำเร็จลุล่วงตามวัตถุประสงค์

เอกสารอ้างอิง

- ฉันทรัตน์ ชัยวันดี และ ณัฐพัชญ์ ศรีราชจันทร์. (2558). *การพัฒนาระบบบันทึกความผิดฝ่ายกิจการนักเรียนโรงเรียนกรรมสถานศึกษา จังหวัดสุพรรณบุรี*. [วิทยานิพนธ์การศึกษามหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม]. <https://publication.npru.ac.th/jspui/handle/123456789/499>
- สนั่น หวานแท้. (2553). *การพัฒนาระบบสารสนเทศเพื่อการจัดเก็บและการสืบค้น สำหรับการบริหารงานบุคคล คณะอักษรศาสตร์ มหาวิทยาลัยศิลปากร*. [วิทยานิพนธ์การศึกษามหาบัณฑิต สาขาวิชาเทคโนโลยีการศึกษา มหาวิทยาลัยศิลปากร]. <https://sure.su.ac.th/xmlui/handle/123456789/7800>
- Ankush Sharma and Aakanksha. (2018). *Introduction to HTML (Hyper Text Markup Language) - A Review Paper*. [Chandigarh University Gharuan, India]. <https://www.ijsr.net/getabstract.php?paperid=ART20182355>
- Cornelia GyÖrödi, Robert GyÖrödi, George Pecherle and Andrada Olah. (2015). *A comparative study: MongoDB vs. MySQL*. [Department of Computer Science and Information Technology, University of Oradea, Romania]. <https://ieeexplore.ieee.org/document/7158433>
- Dr. S.S.Mule and Mr.Yashwant Waykar. (2015). *ROLE OF USE CASE DIAGRAM IN SW DEVELOPMENT*. [Department of Management Science Dr.B.A.M.University, Aurangabad]. <http://bit.ly/4bBLhFt>
- Gavin Bierman, Martin Abadi and Mads Torgersen. (2014). *Understanding TypeScript*. [Microsoft]. https://link.springer.com/chapter/10.1007/978-3-662-44202-9_11
- Henoch Juli Christanto and Yerik Afrianto Singgalen. (2023). *Analysis and Design of Student Guidance Information System through Software Development Life Cycle (SDLC) and Waterfall Model*. [Information System Department, Faculty of Engineering, Atma Jaya Catholic University of Indonesia, Jakarta, Indonesia]. <https://journal-isi.org/index.php/isi/article/view/645>

- Kim Tran. (2023). *State Management in React*. [Vaasa University of Applied Sciences VAMK, Finland]. <https://www.theseus.fi/handle/10024/798988>
- N. Srinivas, and J. Dondeti. (2012). *Black box and white box testing techniques-a literature review*. [International Journal of Embedded Systems and Applications (IJESA)]. <https://bit.ly/3NboMiw>
- Sari Azhariyah and Muhammad Mukhlis. (2024). *Framework CSS: Tailwind CSS Untuk Front-End Website Store PT. XYZ*. [Politeknik Negeri Subang, Subang]. <https://bit.ly/47Xl4zU>
- Shirley Radack. (2009). *THE SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)*. [Computer Security Division Information Technology Laboratory National Institute of Standards and Technology]. <https://www.nist.gov/publications/system-development-life-cycle-sdlc>