# WEB APPLICATION FOR UNIVERSITY INTERNAL AUDIT SYSTEM

เว็บแอปพลิเคชันสำหรับระบบตรวจสอบภายในมหาวิทยาลัย


**BY**

**MR. SARUN JEARAWATTANAKUL 6288187**
**MR. KRITTAPAT DONSOM 6388120**
**MR. WARANAT DEEPRADUB 6388187**

**ADVISOR**

**LECT. PAGAPORN PENGSART**


**A Senior Project Submitted in Partial Fulfillment of the Requirements for**

**THE DEGREE OF BACHELOR OF SCIENCE (INFORMATION AND COMMUNICATION TECHNOLOGY)**


**Faculty of Information and Communication Technology**
**Mahidol University**

# ACKNOWLEDGEMENTS

This project focuses on the development of a web application designed for auditing university financial records to ensure accuracy and regulation. Traditionally, this auditing process has been performed manually, involving numerous paperwork and time consuming. However, nowadays technology has enabled us to create a web application audit system, making it more efficient and simple. These tools are essential due to the complexity and numerous data associated with university auditing finances documents. The goal for this project was to replace the traditional audit process with a more refined and effective solution, utilization of technology to enhance accuracy and efficiency.

The successful completion of this project can be attributed to the invaluable assistance and unwavering support extended by numerous individuals. We wish to extend our sincere gratitude to all those involved. Paramount to our achievements was the guidance and expertise generously provided by our advisor, LECT. PAGAPORN PENGSART. Additionally, we express our profound appreciation to the Faculty of Information Technology and Communication at Mahidol University, Thailand. The accomplishments of this project would not have been possible without the crucial support mentioned earlier.

| | | |
|---|---|---|
| Mr. Sarun | Jearawattanakul | 6288187 |
| Mr. Krittapat | Donsom | 6388120 |
| Mr. Waranat | Deepradub | 6388187 |

# WEB APPLICATION FOR UNIVERSITY INTERNAL AUDIT SYSTEM

MR. SARUN JEARAWATTANAKUL 6288187 ITCS/B
MR. KRITTAPAT DONSOM 6388120 ITCS/B
MR. WARANAT DEEPRADUB 6388187 ITCS/B

B.Sc. (INFORMATION AND COMMUNICATION TECHNOLOGY)

PROJECT ADVISOR: LECT. PAGAPORN PENGSART

## ABSTRACT

AUDITING MAHIDOL UNIVERSITY'S AUDIT TECHNIQUE HAS TRADITIONALLY BEEN A MANUAL PROCESS, REQUIRING A LOT OF PAPERWORK AND TIME CONSUMING. THE PURPOSE OF THIS PROJECT IS TO PROVIDE A WEB APPLICATION THAT FACILITATES THE AUDITING OF FINANCIAL RECORDS AT MAHIDOL UNIVERSITY IN ORDER TO ASSURE CORRECTNESS AND STANDARD WITH REGULATIONS. THE APPLICATION INCLUDES FEATURES FOR DATA STORAGE AND RETRIEVAL, AS WELL AS HANDLING FREQUENTLY USED DATA TO REDUCE CRITICAL PROCESSES, ALL DESIGNED TO HANDLE THE COMPLEXITY AND NUMEROUS OF UNIVERSITY AUDITING DOCUMENTS. OUR GOAL IS TO IMPROVE BOTH THE PRECISION AND RELIABILITY OF FINANCIAL AUDITS WHILE REDUCING COSTS AND SAVING TIME. MOREOVER, THE WEB APPLICATION IS DESIGNED TO BE USER-FRIENDLY AND EASY TO INTEGRATE INTO EXISTING AUDITING SYSTEMS. MOREOVER, THIS PROJECT SEEKS TO PROVIDE A MORE EFFECTIVE AND EFFICIENT AUDITING EXPERIENCE, HELPING TO ENSURE FINANCIAL INTEGRITY AND TRANSPARENCY.

**KEYWORDS : Web Application, Financial Audit, University Finances, Automation, Compliance, Real-time Data Analysis**

แอปพลิเคชันเว็บสำหรับระบบตรวจสอบภายในมหาวิทยาลัย

นาย ศรัณญ์ เจียรวัฒนกุล 6288187 ITCS/B
นาย กฤตพัฒน์ ดอนโสม 6388120 ITCS/B
นาย วรณัฐ ดีประดับ 6388187 ITCS/B

วท.บ. (เทคโนโลยีสารสนเทศและการสื่อสาร)

อาจารย์ที่ปรึกษาโครงการ: อาจารย์ ผกาพร เพ็งศาสตร์

บทคัดย่อ

การปฏิบัติงานตรวจสอบภายในของมหาวิทยาลัยมหิดล ในปัจจุบันการตรวจสอบและการประเมินผลการปฏิบัติการตรวจสอบภายในยังคงใช้วิธีการแบบดั้งเดิมซึ่งมีการทำเอกสารจำนวนมากทำให้ผู้ตรวจสอบใช้เวลามากและพบข้อผิดพลาดได้บ่อยครั้ง ดังนั้นการวางแผนและปฏิบัติการภายในผ่านทางเว็บแอพพลิเคชั่นจึงสามารถเข้ามาช่วยเพิ่มประสิทธิภาพมากกว่าวิธีการตรวจสอบแบบดั้งเดิม โดยวัตถุประสงค์ของโครงการนี้คือจัดทำเว็บแอปพลิเคชั่นที่อำนวยความสะดวกในการตรวจสอบบันทึกทางการเงินของมหาวิทยาลัยมหิดลเพื่อปฏิบัติตามกฎระเบียบและ ปฏิบัติการตรวจสอบและถูกต้องตามขั้นตอน เว็ปแอปพลิเคชันนี้มีคุณสมบัติสำหรับการจัดเก็บและเรียกค้นข้อมูลรวมถึงการจัดการข้อมูลที่ใช้บ่อยซึ่งช่วยลดขั้นตอนที่ยุ่งยากทั้งหมดนี้ออกแบบมาเพื่อจัดการกับความซับซ้อนและปริมาณของเอกสารการตรวจสอบของมหาวิทยาลัยเป้าหมายของเราคือการปรับปรุงทั้งความแม่นยำและความน่าเชื่อถือของการตรวจสอบทางการเงินในขณะเดียวกันก็ลดต้นทุนและประหยัดเวลานอกจากนี้ เว็บแอปพลิเคชันยังได้รับการออกแบบให้ใช้งานง่ายและง่ายต่อการรวมเข้ากับระบบการตรวจสอบที่มีอยู่นอกจากนี้โครงการนี้มุ่งหวังที่จะมอบประสบการณ์การตรวจสอบที่มีประสิทธิภาพและประสิทธิผลมากขึ้นช่วยให้มั่นใจในความสมบูรณ์ทางการเงินและความโปร่งใส

85 หน้า

# CONTENTS

# LIST OF TABLES

Page

# LIST OF FIGURES

Page

# CHAPTER 1

# INTRODUCTION

## 1.1 Motivation

The internal audit system is essential for every organization, including the university. Traditional auditing is now done manually and has several drawbacks that have caused disadvantageous impacts. Some examples of the negative effects include human error, inefficiency and time-consuming, limited scalability or struggle to handle increased volumes of data and audit tasks, difficulty in data analysis, the lack of time responding, and difficulty in collaboration. These are the motivations to use technology to make positive changes, improve workflows, reduce manual errors, and enhance the overall effectiveness of the web application for the university's internal audit system. The main focus of this project is to help auditors by creating a web application for the internal audit system of the university that is a more efficient, transparent, and responsive auditing process. Auditors can select Audit plans, access documentation, start auditing projects and data processing, and track and monitor the systems, and reports. Moreover, the purpose of these features is to help with auditor workflows and enhance the overall internal audit system. In addition to its technical features, this project provides an opportunity for collaboration and skill development. It encourages teamwork. The development of a web application for the university's internal audit system demonstrates our dedication to continuous improvement, innovation, and adapting to the evolving system. In the future, this web application for the university's internal audit system is an opportunity for the user to improve and add more features or even expand the ability to assist the auditor.

## 1.2 Problem Statement

- **Identification of Current Issues:** Begin by identifying and succinctly describing the key issues and challenges prevalent in the current internal audit system. This may include manual and time-consuming processes, decentralized data management, a lack of real-time insights, and the potential for errors in the auditing procedures.

- **Impact on Efficiency and Accuracy:** Delve into the consequences of the identified issues on the efficiency and accuracy of the internal audit processes. Discuss how the current system may result in delays, hinder timely decision-making, and potentially compromise the accuracy of audit findings. Emphasize the need for a more streamlined and automated approach to mitigate these challenges.

- **Transparency and Compliance Concerns:** Address any transparency and compliance concerns associated with the current system. Explore how a lack of centralized data may hinder transparency, making it challenging to ensure adherence to regulatory standards and internal policies. Highlight the importance of instilling confidence among stakeholders in the integrity of the audit processes.

- **Resource Allocation and Workload:** Discuss the allocation of human and financial resources required for the maintenance of the current system. Highlight any inefficiencies in resource utilization, such as excessive manual efforts and redundancies. Consider the impact on the workload of audit teams and how a streamlined web application could optimize resource allocation.
- **Adaptability to Technological Advancements:** Acknowledge the rapid evolution of technology and its implications for internal auditing. Discuss how the current system may struggle to keep pace with technological advancements, potentially resulting in outdated practices. Emphasize the necessity of a web application that is not only contemporary but also adaptable to future technological shifts.

## 1.3 Objectives of the Project

- To perform an internal audit used for reference and systematically raise the quality of the internal auditing of the university

## 1.4 Scope of the Project

- The scope of our project is to use a Web application to help create Audit plans.
- Web application It will facilitate users by organizing Audit plans. Documentation Verification notification Processing in various formats

## 1.5 Expected Benefits

- All internal auditors at the university perform their duties in accordance with guidelines appropriate to Mahidol University.
- All internal auditors at the university perform their duties. The results of the internal audit are of high quality according to internal audit standards. and able to respond to the needs of those involved, such as those who support various budgets and executives at all levels, in accordance with guidelines appropriate to Mahidol University.
- The internal audit process has appropriate steps. and can be continuously developed to be more efficient.
- The university's departments and agencies recognize all internal auditors at the university.

## 1.6 Organization of the Document

This document consists of 6 chapters including:

1. Introduction – Introduction contains 6 sections, including project motivation, problem statement, objective, scope of the project, expected benefits, and document organization.

2. Background – Background describes the background and related work of the project.
3. Analysis and Design – Analysis and Design describes the process of the project, methodology, and system architecture represented by a diagram, structure chart, data flow diagram, database design, and I/O design.
4. Implementation – The last chapter consists of five sections which are the conclusion, benefits, problems and limitations, challenges of the project, and future work.
5. Testing and Evaluation – This chapter discusses usability testing.
6. Conclusion – The last chapter consists of five sections which are the conclusion, benefits, problems and limitations, challenges of the project, and future work.

# CHAPTER 2

## BACKGROUND

The goal of the audit web application project is to develop a web audit application for Mahidol University to improve the efficiency and accuracy of financial auditing processes. Traditional manual audits are prone to human error, requiring significant time and resources. The feature of this web audit application will reduce paperwork and travel costs while ensuring compliance with ISO 19011 standards for auditing management systems. Key features include data storage and retrieval, comprehensive reporting capabilities, and a user-friendly interface. The application seeks to meet user expectations in developing the audit web application while following international auditing standards and meeting user requirements in developing the audit web application.

### 2.1User requirement

2.1.1 On the Audit side

1. Preparation of an audit plan
2. Long-term plan (4 years)
3. Annual plan

condition

- User is the importer of factors for planning.
- system is processed into long-term and annual plans.
- Users can adjust the map that the system has already processed (both long-term and yearly).

2.1.2 Preparation of project Audit documents

- Audit request letter
- Engagement Plan
- Schedule
- PowerPoint Open Audit
- Routing Slip

condition

- Let the system have a form according to 1.2.1 - 1.2.5, which has a key field for the system to fill out according to the information in the plan, but the user can be modified.
- In every fiscal year, let the user choose the annual plan (1.1.2) to proceed. Then set the time that the system will issue a request for Audit

  (1.2.1), such as 21 days before the start date, and be effective with all audit projects (audit job) in that plan. On the due date, give
  The system assigns the project code. Issue a request to check (1.2.1) delivered to the specified email and save the file in the job folder.

- Issue Engagement Plan, Schedule, and PowerPoint (1.2.2 - 1.2.4) at the Save File system in that job folder as well.
- User retrieves documents 1.2.2 - 1.2.4 and edit or increase data to print out.
- On the day of the Audit (as specified by the plan), the User's right to choose Audit Job to start checking and click "START"
- The system starts counting Audit Project Start Date (Fieldwork) and saves the data in Routing Slip (1.2.5).
- User identifies the status of the project that can be checked, such as in the process of field Audit. In the process of drafting reports, etc. by processing from the data in Routing Slip

2.1.3 Processing to randomize data for verification

condition

- Users can import data from outside, such as Excel files, reports from the MU-ERP system and others with similar characteristics, etc.
- Users can set random conditions such as maximum, minimum, number of items, etc.
- The system shows the sampling results along with the conditions that the user determines.
- The system stores that sampling data in the job folder/subfolder 'Paper'.

2.1.4 Write a paper from the Audit Program and write a summary of 5 audits condition

2.1.5 The system has an Audit Program database, which users can call all files along with the content in the file.

2.1.6 The use of the Audit Program (fill in the examination results) is characterized as filling through the menu.

In the menu 'Audit Program'

- User chooses the Audit Program (issue) to be used.
- Users can fill in the results and save only some parts, including being able to edit later until the User is in Charge Job or Super User (Director). Will click End Job by allowing the system to save data in the job folder / sub folder 'paper'
- Users can fill in the results by creating or adding tables.
- Users can attach files, documents from outside.
- In each sub-story in the Audit Program, each user can click to conclude that it is not an issue or an observation.

2.1.7 When User Save, the results entered in the 'Audit Program' menu are completed.

- Allow the user to call the Audit Program report (Edition) and export to save as a Word file (paper report) and can search for many old paper reports (Dimension), such as using the search command (search) according to the paper - subject Or according to the list of Audit projects (Job Folder) or use Drop Down List and can copy (copy) messages from both the system And external sources To paste (paste) in the menu 'Audit Program' for use in other Audit projects
- Let the data processing system be a list of issues and observations of the Audit project (Job) - by each issue / observation To specify the project code as a separate field - and User can call all list reports according to the period. And the specified conditions

2.1.8 Users can prepare a summary report of 5 fields by filling in additional details that should choose the topic of the big issue, sub-issue from the Drop Down List or search for the issue name from Master Data and should have a field to fill in more details. Can link with the paper And can choose the attachment from the paper (Master Data is the name of the topic, major issues, sub-issues and categories of detected items that have the same typical text (Pattern) or may be defined as a numerical code instead of the name of the topic for the benefit of further management and analysis of data)

2.1.9 Prepare both draft and complete audit reports.

- Let the system process data from the key field in 1.1, 1.2, 1.4.3, 1.4.5 to issue (draft) reports.
- Allow the user to adjust additional information, except the name of the topic, the big issue - sub-issue And the type of the detected category Because it will be defined as Master Data
- (In case it is necessary to edit, use the director's user privileges. Internal Audit Center)

- Let the user choose the attachment from the paper or summarize the results of 5 channels. When completed, the User Click 'NEXT' for the system to send email and link to the User.
- When the right to adjust the information, click 'NEXT' for the system to send and continue to the rights of the director.
- When the rights of the director Adjust the information, click 'NEXT' to allow the system to send email and link back to the user
- When there is no further correction, give User In Charge click 'FINISH'
- The system (draft) report 1 is a file, which the user can export as a Word file.
    - (Draft) Report 2 ((draft) Report 1 + Opinion of the Audit unit)
- When User Rights In Charge Receives Opinion Information From The Audit Unit To Fill In That Information To Add In (draft) Report 1
- The system issued (draft) report 2, which the user can print and export to save as a Word file.
- If there is an additional amendment, the User In Charge can redo 1.5.8. If not, click 'COMPLETE'. Complete report
- The system issued a complete Audit report by processing from (draft) report 2 and Users can still edit additional data and links and print and export to save as a Word file.
- **Notify the follow-up as scheduled And prepare a follow-up book with Action Sheet**
    - User specifies the due date to follow up the job. Log in with the email (s) that will allow the system to send notifications (In Charge Job and General Administration Officer)
    - Each job can be followed up no more than 2 times, the same process14 days before the due date, the system sends a notification to the specified email, along with the follow-up request and summary of the Audit report (Action Sheet) according to the template.
    - Users can edit the follow-up book. But can not solve the Action Sheet and can print and export to save as a Word file

2.1.10 Prepare an audit follow-up report.

- User enters the follow-up menu, selects the job to follow and selects the follow-up time.
- 1st Tracking
- The system displays the issue information according to the Action Sheet Job (1.6.2) and the User must not be able to edit the issue content.
- The system has a box for the user to fill in.
- Your name and date of follow
- 'Update status', which is 'can be closed' and 'cannot be closed' with accompanying reasons.

- If you select "cannot close" User and the system must repeat the process from 1.6.1 for the 2nd follow-up.
- The system processes and issues the 1st follow-up report according to the specified template.
- User Print and can export to save as a Word file.
- 2nd Tracking
- The information in the box according to 1.7.3 will be a new set of information. It is the information of "2nd update status".
- The system processes and issues the 2nd follow-up report according to the specified template.
- User Print and can export to save as a Word file.
- There is a box for User Rights of the Job. Click 'END' Job.
- Job information archive system **** No user can edit again ****

## Literature Review

### 2.2 Overview of web application for university internal audit system

Web applications must meet the international standard “ISO 19011 Guidelines for Auditing Management Systems” because nowadays many Agencies, whether public or private or some regulatory agencies, have used this standard in their own organization's audit management and have received satisfactory results. ISO 19011 standard is currently in its 3rd edition, updated in 2018. ISO 19011:2018 is designed to be used for both Internal and External Audit according to the Deming Cycle or PDCA (Plan Do Check ACT). It consists of the following main components principle of auditing (CI.4), Managing an audit programme (CI.5), Conducting an audit (CI.6), and Competence and evaluation of auditors (CI.7).

The principles outlined in Clause 4 of ISO 19011 serve as the foundation for all auditing activities. These principles, including integrity, fair presentation, and due professional care, are extensively discussed in the literature. Researchers highlight the importance of maintaining auditor independence, ensuring confidentiality, and embracing a risk-based approach to auditing. The integration of these principles into organizational culture and practices is a key focus for ensuring the credibility and reliability of audit outcomes.

ISO 19011's guidance on managing audit programs (Clause 5) has garnered attention in the literature for its role in enhancing organizational efficiency. Scholars emphasize the significance of establishing clear audit objectives, defining criteria, and ensuring adequate resources. The literature also explores the role of leadership in creating a culture that supports the audit program, fostering continual improvement and adaptability to changes in the organizational context.

Clause 6 of ISO 19011 provides detailed guidance on the process of conducting audits. Literature in this area discusses the practical application of audit methods, emphasizing the need for systematic and evidence-based approaches. Researchers highlight the value of communication during the audit process, both within the audit team and with auditees. Case studies and practical examples in the literature

demonstrate how organizations have successfully implemented ISO 19011's recommendations in their auditing processes.

The competence and evaluation of auditors, as outlined in Clause 7, are crucial for maintaining the effectiveness of auditing activities. The literature addresses the qualifications, skills, and personal attributes required for auditors. It also explores various methods of evaluating auditor competence, including training, experience, and ongoing professional development. The role of certification bodies in ensuring auditor competence is also discussed in the literature.

In conclusion, this literature review provides a comprehensive overview of the key components of ISO 19011:2018, focusing on the Principles of Auditing, Managing an Audit Program, Conducting an Audit, and Competence and Evaluation of Auditors. The body of literature underscores the significance of these components in establishing robust auditing processes, fostering continual improvement, and ultimately contributing to the overall success and sustainability of organizations.

**Related work**

In recent years, the increasing digitization of processes across various industries has prompted academic researchers and practitioners to explore the integration of web applications in the field of university internal audit systems. This literature review aims to examine the current state of research and development in the realm of web-based tools for enhancing the efficiency and effectiveness of internal audit processes within the academic setting.

**2.3 Integration of Web Technologies in Internal Auditing:**

The convergence of web technologies with internal auditing practices has been a subject of interest for researchers and professionals. Chen et al. (2018) highlighted the benefits of web-based systems in facilitating real-time data access, collaborative audit planning, and streamlined communication among audit team members. The utilization

of web applications has shown promise in overcoming traditional limitations associated with manual and paper-based audit processes[1].

**2.4 User Interface Design for Internal Audit Systems:**

The importance of an intuitive and user-friendly interface in web applications for internal audit systems has been emphasized in the literature. Johnson and Brown (2019) argued that a well-designed user interface contributes significantly to the adoption and effectiveness of internal audit tools[2]. Elements such as dashboard visualization and navigation mechanisms play a crucial role in enhancing the user experience.

**2.5 Security and Authentication Challenges:**

The integration of web applications in the sensitive domain of internal auditing necessitates a robust security framework. Smith and Patel (2020) examined the challenges associated with user authentication and data security in web-based internal audit systems[3]. The study emphasized the need for multi-factor authentication and encryption protocols to safeguard confidential audit information.

**2.6 Data Management and Analysis in Web-Based Audit Tools:**

Efficient data management and analysis capabilities are crucial components of a successful web application for internal audit systems. The work of Lee and Kim (2017) explored the integration of data analytics tools within web platforms to enhance the audit process[4]. Their research demonstrated how web applications can facilitate data-driven decision-making in internal auditing.

**Conclusion:**

This literature review provides a snapshot of the current research landscape regarding web applications for university internal audit systems. The integration of web technologies has shown promise in improving collaboration, user experience, and data-driven decision-making in the field of internal auditing within the academic context.

# CHAPTER 3

## ANALYSIS AND DESIGN

This chapter consists of the system architecture overview, the process analysis and design, the database analysis and design, the I/O design, and the current progress. We separate the process analysis and design into two sections, which are the data system structure chart and the flow diagram. Database analysis and design consists of three sections which are the ERdiagram, relational schema, and data dictionary. The I/O design consists of two sections which are the Interface design and the transition diagram.

### 3.1 System Architecture Overview

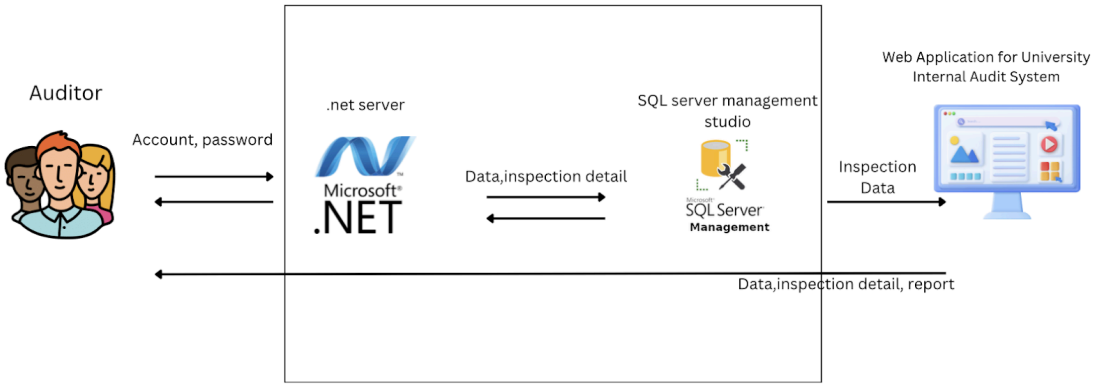



Figure 3.1: System architecture

The system architecture overview of this project primarily focuses on auditors. Our software framework utilizes .NET and Microsoft SQL Server Management for configuring, managing, and administering all database components. This setup is essential for developing a web application for the university's internal audit system.

### 3.2 System Structure Chart

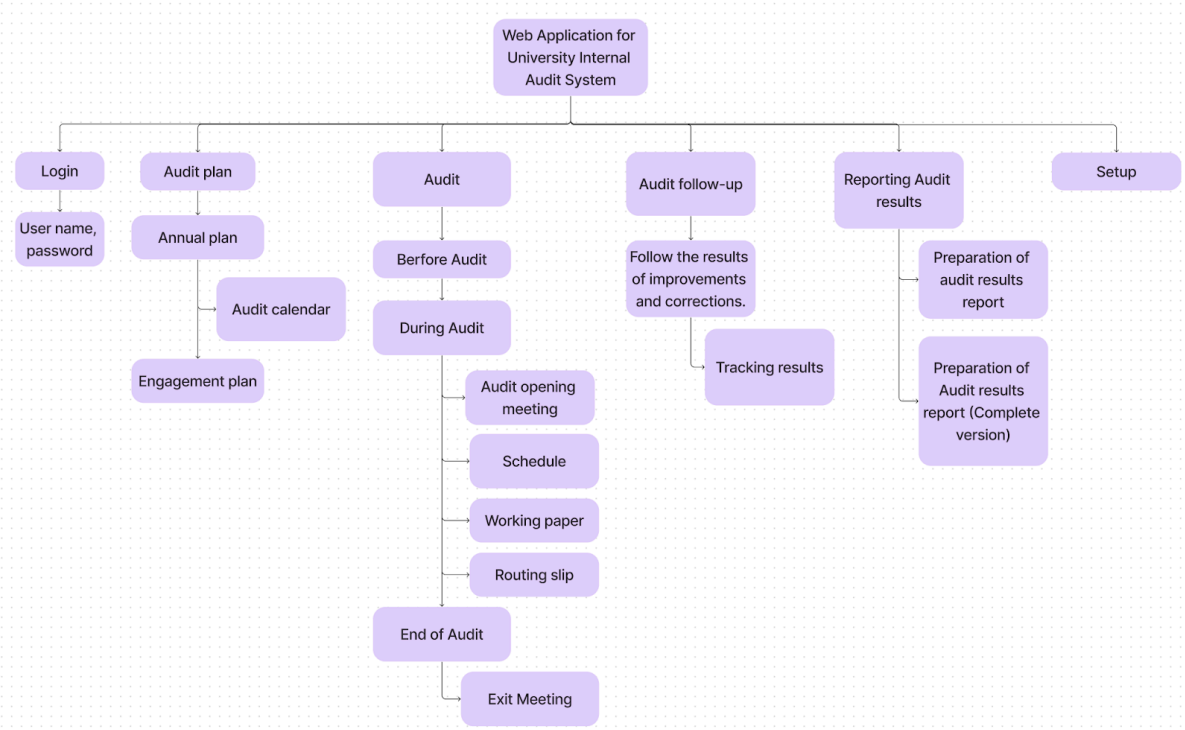



Figure 3.2: Structure chart

The structure chart of this web application for the university internal audit system is divided into six modules: login, audit plan, audit, audit follow-up, reporting audit results, and setup.

#### 3.2.1 Login

The module is a login module.

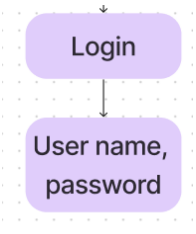



Figure 3.2.1: Login module

This login module contains the following components including username and password.

### 3.2.2 Audit plan

This module is an Audit plan. This module includes an annual plan and engagement plan.

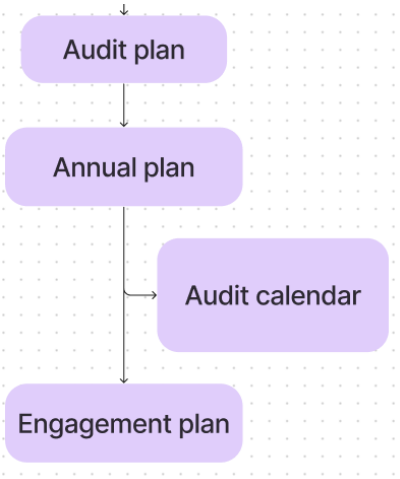



Figure 3.2.2: Audit plan module

The audit plan module contains the following components, including annual plan, which consists of a subcomponent known as audit calendar, and engagement plan.

### 3.2.3 Audit

The module is the Audit module. The module include the Audit procedures including before Audit, during audit, and end of audit

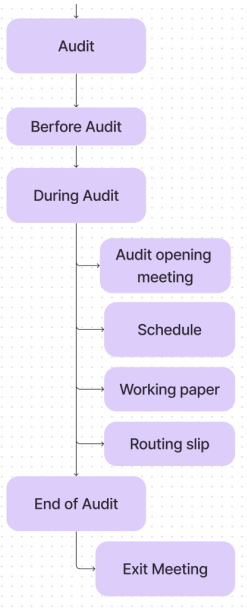



Figure 3.2.3: Audit module

The audit module contains the following components: before audit, during audit, which consists of a subcomponent known as audit opening meeting, schedule, working paper, and the last component, exit meeting.

### 3.2.4 Audit follow-up

The module is the audit follow-up. This module includes following the results of improvements and corrections, and tracking results.

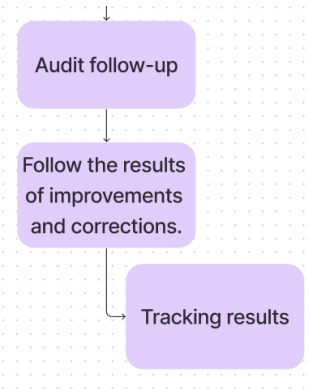



Figure 3.2.4: Audit follow-up module

The audit follow-up module contains the following components: audit follow-up, which consists of a subcomponent known as tracking results.

### 3.2.5 Reporting Audit results

The module is a reporting audit results. This module includes preparation of audit results report, and preparation of audit results report (complete version).

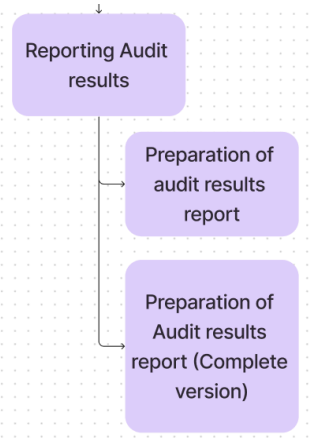



Figure 3.2.5: Reporting Audit results module

The reporting audit results module contains the following components: Preparation of audit results report, and preparation of audit results report(complete version).

### 3.2.6 Audit setup

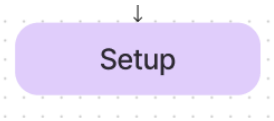



Figure 3.2.6: Audit setup

The reporting audit results module contains the following components: preparation of audit results report, and preparation of audit results report(complete version).

The detailed description of each subsystem is shown below:

Login Module

- User Login Component

Audit plan Module

- Before Audit Component
- During Audit Component
- End of Audit Component

Audit follow-up Module

- Follow the results of improvements and corrections Component

Reporting Audit result Module

- Preparation of audit results report Component
- Preparation of Audit results report (Complete version) Component

Setup Module

- Setup Component

**Process Analysis and Design**

### 3.3 Data Flow Diagram

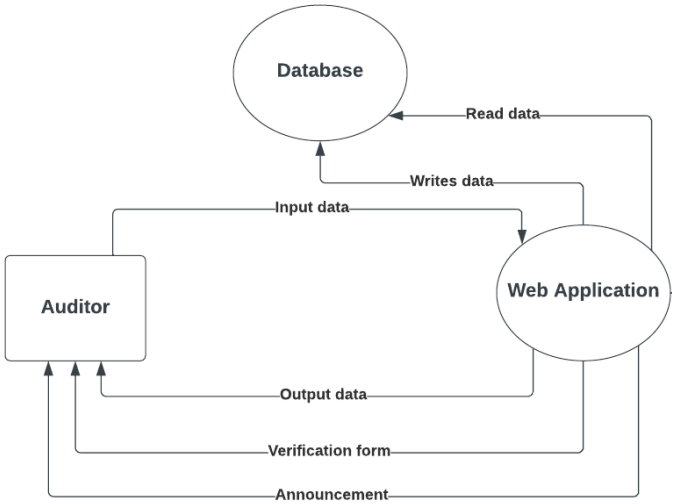



Figure 3.3:Data flow diagram Lv0

This level 0 data flow diagram shows the flow of the web application system for auditors. Initially, auditors send their login information to the system, and the system will retrieve their data from the database for authentication. Once logged in, auditors can input, edit, and delete audit data. The system updates the database with the auditors' inputs.

Following the audit process, auditors receive outputs from the system, which retrieves data from the database. Auditors then receive verification forms and announcements from the system based on the processed data.

### 3.4 Data Flow Diagram Level.1

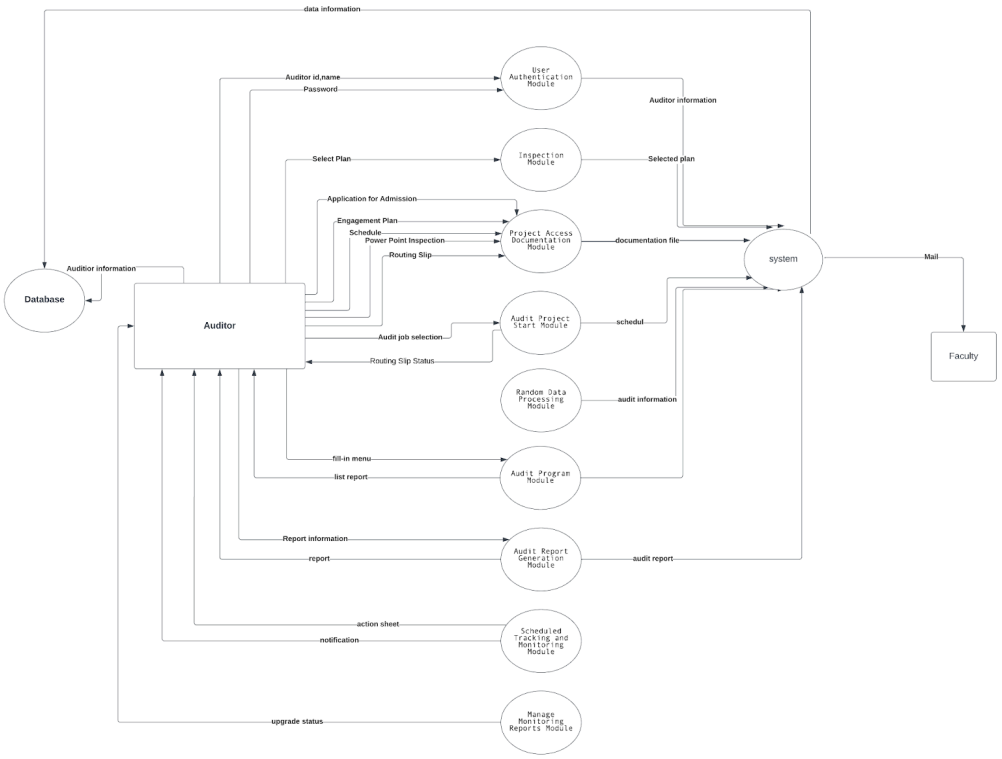



Figure 3.4:Data flow diagram Lv1

This level 1 data flow diagram shows the flow of the web application system for auditors, containing the following information.

**Login**

This login will request auditor information including username and password will be sent to the system and then stored into the database.

**Audit Plan**

The audit plan will request auditor information, including annual plan data, and engagement plan data, then it will be sent to the system and then stored into the database.

**Audit**

The audit will request auditor information including before audit data, during audit data ,then it will be sent to the system and then stored into the database.

**Audit follow-up**

The audit follow-up will request auditor information including before audit audit follow-up data, then it will be sent to the system and then stored into the database.

**Audit Report**

The reporting audit results will request auditor information including department name, the topic or the subject of the audit, the date, and the status of each faculty ,then it will be sent to the system and then stored into the database.

### 3.5 Use Case Diagram

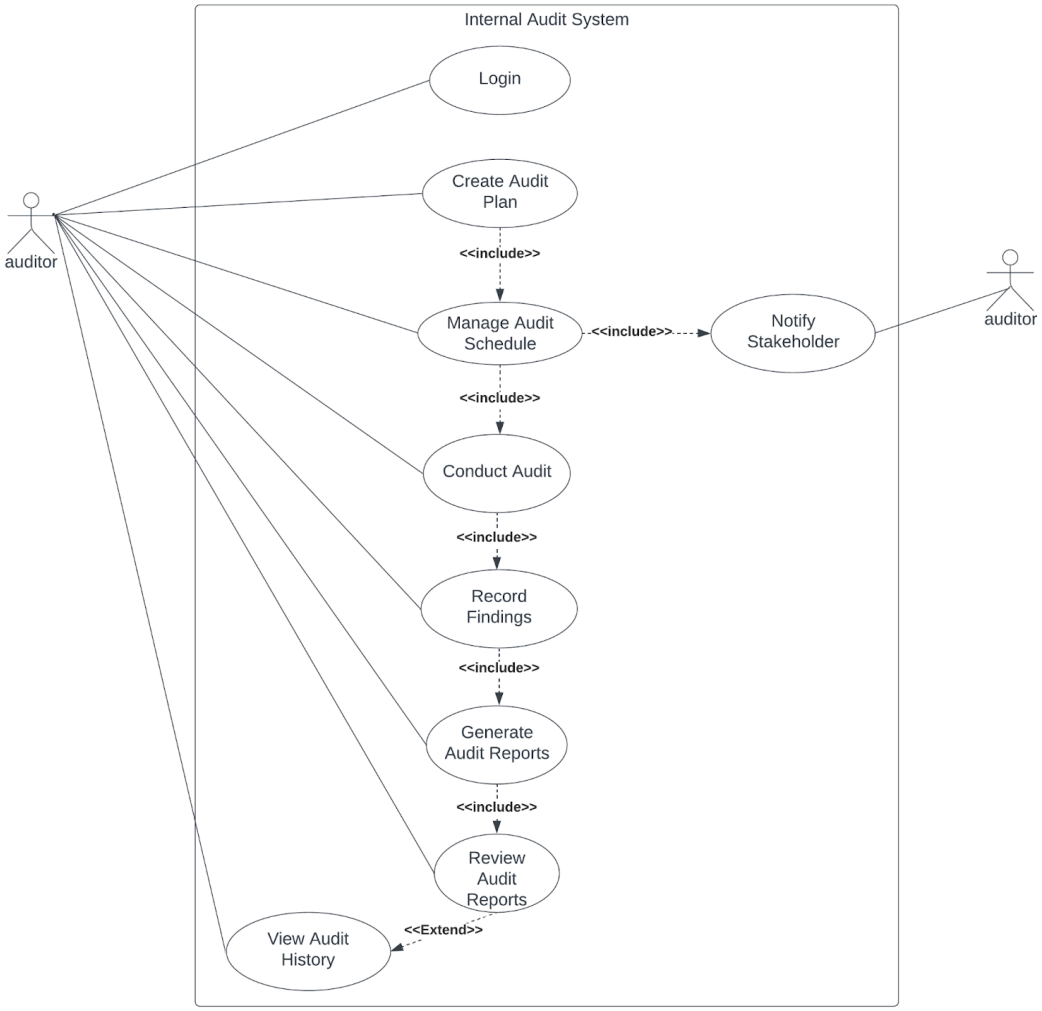



Figure 3.5: Use Case Diagram

This use case diagram for the web application system designed for auditors at Mahidol University encompasses several key functionalities. Auditors can log in by inputting their information, which are validated and stored in the system's database. They can then create, update, or review audit plans, including engagement plan details, which are saved for future reference. During audits, auditors input data before and during the audit process, which is transmitted to the system and stored securely. Moreover, auditors can conduct follow-up reports by entering data into the system. Finally, auditors can recieve audit reports by specifying department names, audit topics, dates, and faculty statuses. This use case diagram highlights the system's role in facilitating

efficient auditing processes while ensuring data integrity and compliance with auditing standards

**Data Stores**

This section describes the data stores that exist in the data flow diagram and consists of the Data Store Name, Description, Inbound Data, and Outbound Data.

Table 3.6: List of all Data Stores

| No. | Data Store Name | Type | Description |
|---|---|---|---|
| 1 | Board_id | int | Unique identifier for a board. |
| 2 | BoardName | nvarchar | Name of the Board. |
| 3 | Inspection_id | int | Unique identifier for inspection. |
| 4 | InspectionName | nvarchar | Name of the inspection. |
| 5 | InspectionDateStart | datetime2 | Start date of inspection. |
| 6 | InspectionDateEnd | datetime2 | End date of inspection. |
| 7 | InspectionObjective | nvarchar | Objective of inspection. |
| 8 | Status | nvarchar | Status of the inspection. |
| 9 | Schedule_id | int | Unique identifier for a schedule. |
| 10 | ScheduleDetail | nvarchar | Detail of the schedule that has an inspection. |
| 11 | ScheduleDoc | nvarchar | The name of the document for the inspection. |
| 12 | ScheduleDate | nvarchar | The date that has been inspected. |
| 13 | ScheduleTime | nvarchar | The time that has the inspection. |
| 14 | RoutingSlip_id | int | Unique identifier for RoutingSlip. |
| 15 | Detail | nvarchar | The detail in the routing slip. |
| 16 | InspectionPlan | nvarchar | The Plan for the day that has an inspection. |
| 17 | WorkPlan | nvarchar | The work plan for the inspection day. |
| 18 | Comment | nvarchar | The comment for the auditing. |
| 19 | User_id | int | Unique identifier for user. |

| No. | Data Store Name | Type | Description |
|---|---|---|---|
| 20 | Username | nvarchar | The username for logging in to the website. |
| 21 | Password | nvarchar | The password for logging in to the website. |
| 22 | Name | nvarchar | The name of the user. |
| 23 | WorkCalendar_id | int | Unique identifier for work calendar. |
| 24 | Project | nvarchar | The Project is to be shown on the work calendar. |
| 25 | ProcessingTime | nvarchar | The processing time of the project. |
| 26 | Quantity | nvarchar | The quantity of the coordinator of the project. |
| 27 | Coordinator | nvarchar | The name of the coordinator of the project. |
| 28 | Issue | nvarchar | The project is an issue. |
| 29 | WorkPaper_id | int | Unique identifier of the work paper. |
| 30 | Activity | nvarchar | The control activity of the project. |
| 31 | WorkPaperProcedure | nvarchar | The procedure for auditing. |
| 32 | InspectionResult | nvarchar | The result of the audit. |
| 33 | Issue | nvarchar | The result of the audit is an issue. |
| 34 | Not_issue | nvarchar | The result of the audit is not an issue. |
| 35 | Remark | nvarchar | The remark for the auditing. |
| 36 | Engagement_id | int | Unique identifier for engagement plan. |
| 37 | EngagementProcedure | nvarchar | The procedure of the engagement plan. |
| 38 | Inspector_Name | nvarchar | The inspector's name. |
| 39 | Working_Hour | nvarchar | The working hours of auditing. |
| 40 | WP_Code | nvarchar | The W/P code. |

### 3.6 Database Analysis and Design

ER-Diagram

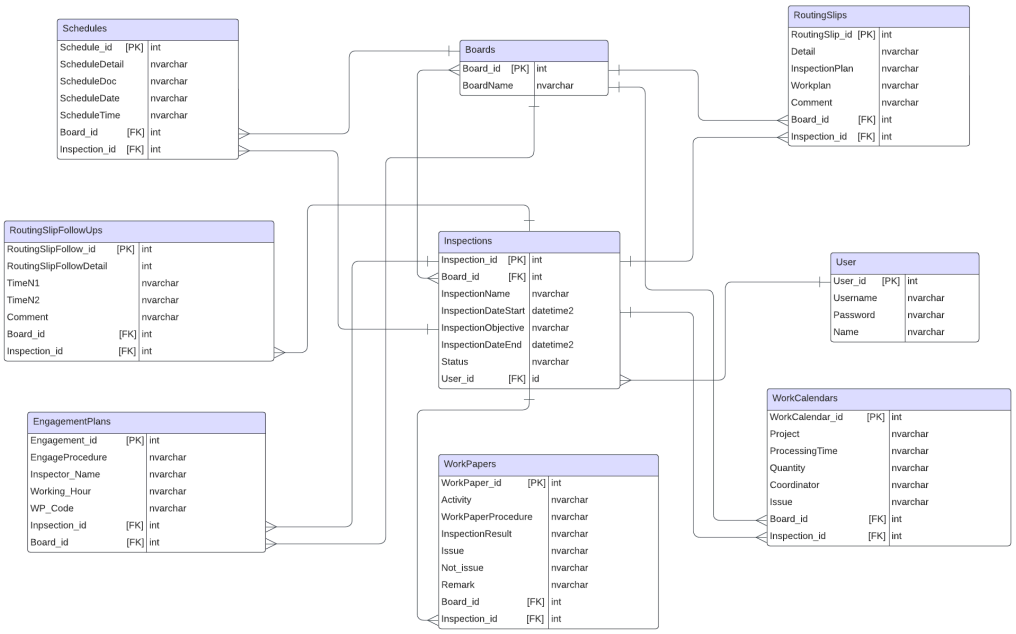



Figure 3.6: ER-Diagram

The figure above shows the er-diagram system consists of nine entities:
schedule, routing slip, routing slip follow-up, boards, user, work calendars, engagement plans, working paper, and inspection.

The user entity has a user_id as the PK, which means each user cannot have the same username.

The schedule entity has a schedule_id as the PK, which means each schedule cannot have the same schedule_id.

The routing slip entity used routingslip_id as the PK, which means that every routing slip cannot have the same routingslip_id.

The routing slip follow-up entity used routingslipfollowup_id as the PK, which means that every routing slip follow-up cannot have the same routingslip_id.

The boards entity used boards_id as the PK, which means that every boards cannot have the same boards_id.

The work calendars entity used workcalendars_id as the PK, which means that every work calendars cannot have the same workcalendars_id.
The engagement plans entity used engagementplans_id as the PK, which means that every engagement plans cannot have the same engagementplans_id.
The working paper entity used workingpaper_id as the PK, which means that every working paper cannot have the same workingpaper_id.
The inspection entity used inspection_id as the PK, which means that every inspection cannot have the same inspection_id.
the end of the stage. The following details will be provided regarding each entity's attributes:

**schedule entity:**

- schedule_id ( PK) : The id to identify the schedule.
- scheduleDetail : Used to indicate the detail of the schedule table.
- scheduleDoc : Used to specify the name of the required document in the schedule.
- scheduleDate : Used to specify the schedule date.
- scheduleTime : Used to specify the schedule time.
- Board_id(FK) : Used to indicate which board is for this schedule.
- inspection_id(FK) : Used to indicate which inspection subjects are for this schedule table.

**routing slip entity:**

- routingslip_id ( PK): The number to identify the routing slip.
- Detail: Used to indicate the detail of routing slip.
- InspectionPlan: Used to indicate the inspection plan detail of routing slip.
- Workplan : Used to indicate the work plan detail of routing slip.
- Comment : Used to indicate the feedback of routing slip.

**routing slip follow-up entity:**

- routingslipfollowup_id ( PK) : The number to identify the routing slip follow-up.
- routingslipFollowDetail: Used to indicate the detail of routing slip follow-up.
- TimeN1: Used to indicate the detail of the first time of routing slip follow-up.

- TimeN2: Used to indicate the detail of the second time of routing slip follow-up.
- Comment: Used to indicate the detail of the feedback of routing slip follow-up.
- Board_id(FK) : Used to indicate which board are for this routing slip follow-up
- inspection_id(FK) : Used to indicate which inspection subjects are for this routing slip follow-up table.

**boards entity:**

- boards_id ( PK): The number to identify the boards.
- BoardName: the name of the board added to the table.

**user entity:**

- user_id ( PK): The number to identify the users.
- Username: the username of the user who registered this account.
- Password: the password of the user who registered this account.
- Name: the account name of the user who registered this account.

**work calendars entity:**

- workcalendars_id ( PK): The number to identify the work calendars.
- Project: Used to specify the project name of the work calendars.
- ProcessingTime: Used to specify the processing time of work calendars.
- Quantity: Used to specify the quantity that is required.
- Coordinator: Used to specify the auditor who's responsible for this matter.
- Issue: Used to indicate the work calendar's objective.
- Board_id(FK) : Used to indicate which boards are for this work calendar.
- inspection_id(FK) : Used to indicate which inspection subjects are for this work calendars table

**engagement plans entity:**

- engagementplans_id ( PK): The number to identify the engagement plan.
- EngageProcedure: Used to store engagement procedure data.
- Inspector_Name: Used to store the inspector's name.
- Working_Hour: Used to store the number of hours of the procedures.
- WP_Code: Used to store wp code of the procedures.
- Board_id(FK) : Used to indicate which board are for this engagement plans

- inspection_id(FK) : Used to indicate which inspection subjects are for this engagement plans table.

**working paper entity:**

- workingpaper_id ( PK): The number to identify the working paper.
- Activity: Used to indicate the working paper activity.
- WorkPaperProcedure: Used to specify the name of the required document in the working paper.
- inspectionResult: Used to specify the inspection result.
- issue: Used to specify the details of issues that need to be resolved.
- Not_issue: Used to specify the details of issues that no need to be resolved.
- Remark: Used to specify the remark topic of the working paper.
- Board_id(FK) : Used to indicate which boards are for this working paper entity.
- inspection_id(FK) : Used to indicate which inspection subjects are for this working paper table.

**inspection entity:**

- inspection_id ( PK): The number to identifies the inspection
- Board_id(FK) : Used to indicate which boards are for this inspection entity.
- User_id(FK) : Used to indicate which inspection subjects are for this inspection table.
- inspectionName: Used to indicate the inspection subject.
- inspectionDatestart: Used to indicate the inspection date start.
- inspectionObjective: Used to indicate the inspection objective.
- inspectionDateEnd: Used to indicate the inspection date end.
- Status: Used to indicate the inspection status.

### 3.7 Relational Schema

This section describes the attributes of the tables in the database. The attribute notation is shown below.

- **Attributes** – which are bold and underlined are the Primary Keys
- *Attributes* – which are Italic are the Foreign Keys
- ***Attributes*** – which are bold, italic, and underlined are both Primary Keys and Foreign Keys

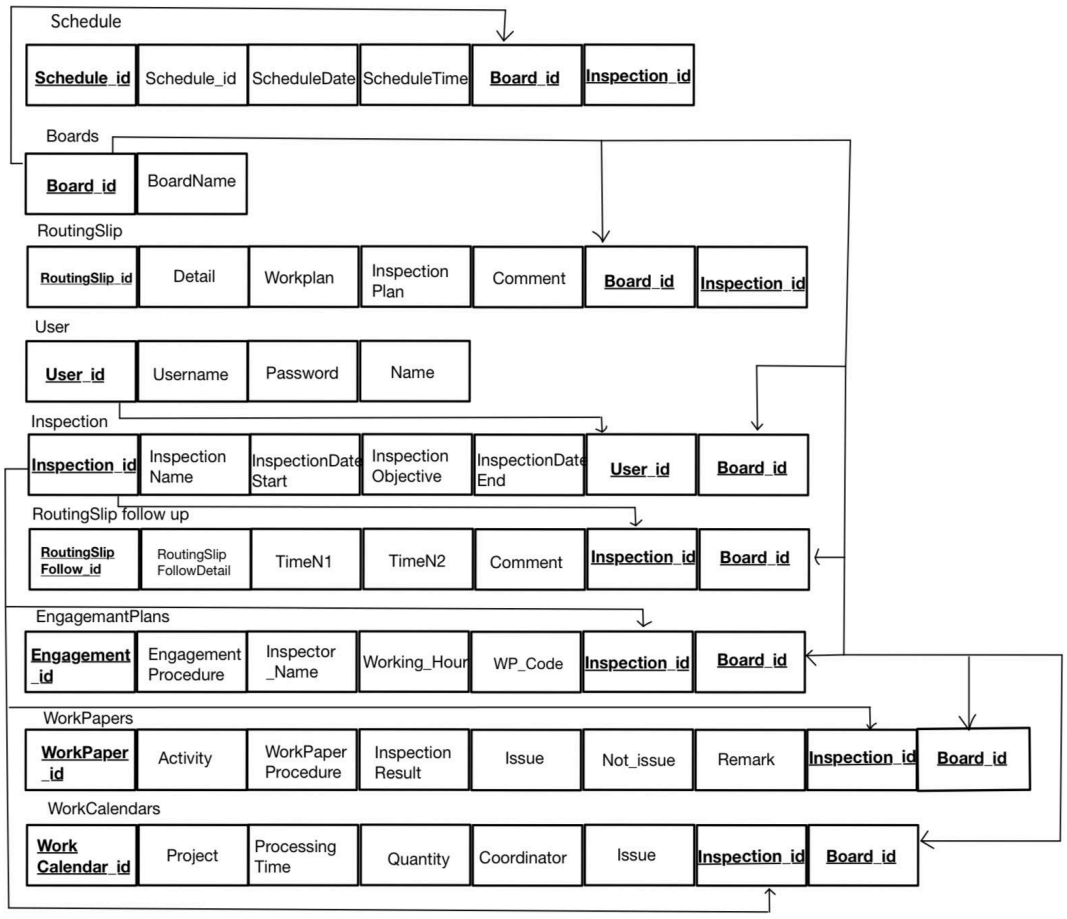



Figure 3.7: Relational schema

The relational schema for Mahidol University's web audit application helps the auditor with the auditing process through web application. The Schedule Entity captures schedule details, while the Routing Slip and Routing Slip Follow-up Entities manage routing slips and follow-ups. The Boards Entity stores board information, and the User Entity manages user accounts. The Work Calendars Entity tracks project

details, and the Engagement Plans Entity details engagement procedures. The Working Paper Entity captures activities and results, and the Inspection Entity manages inspection details. This schema ensures data integrity, enhancing audit efficiency, accuracy, and compliance.

Tables in this system can be divided into 3 groups as follows:

- Master File Table
- Base File Table
- Transaction File Table

Table 3.7: List of all Tables in Our System Database

| Table# | Table Name | Table Type | Description |
|---|---|---|---|
| 1. | User | Master File | This table stores the user information |
| 2. | Audit | Base File | This table stores the Audit plan of the Audit Web Application |
| 3. | EngagementPlans | Transaction File | This table stores the information of the procedure, auditor, and Time of Engagement Plan. |
| 4. | Boards | Transaction File | This table stores the information of the Board. |
| 5. | Schedules | Transaction File | This table stores the information on the schedule for having an audit. |
| 6. | RoutingSlips | Transaction File | This table stores the information of the detailed plan for the audit. |
| 7. | WorkingPapers | Transaction File | This table stores the auditor, comment, remark of the audit |
| 8. | RoutingSlipFollowUp | Transaction File | This table stores the information on the follow-up of the audit schedule. |

| Table# | Table Name | Table Type | Description |
|---|---|---|---|
| 9. | WorkCalendars | Transaction File | This table stores the information of the Calendar to save the audit calendar data. |

**Interface Design**

### 3.8 Login page

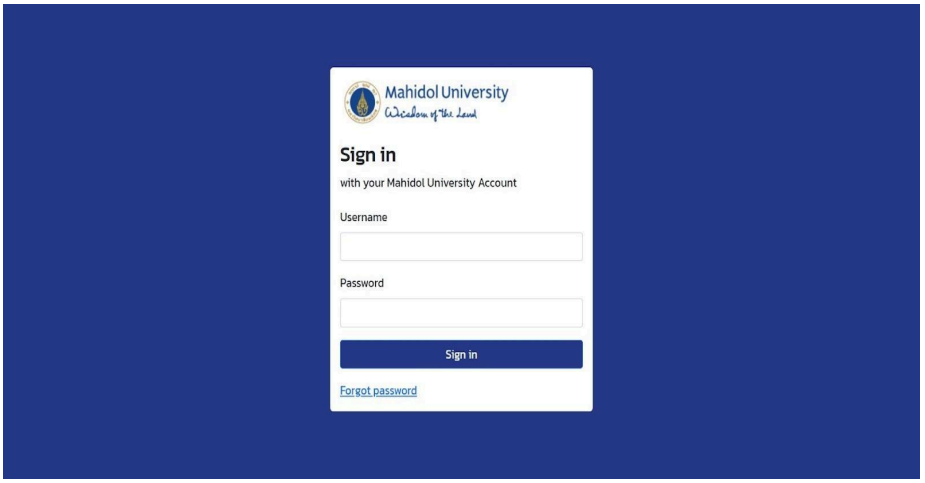



Figure 3.8 :Login page

This page allows auditors to log in by entering their username and password. By providing their credentials, auditors can securely access the system to perform their auditing tasks. The login process ensures that only authorized personnel can gain entry, thereby maintaining the security and integrity of the financial records and audit data.

### 3.8.1 Setting page

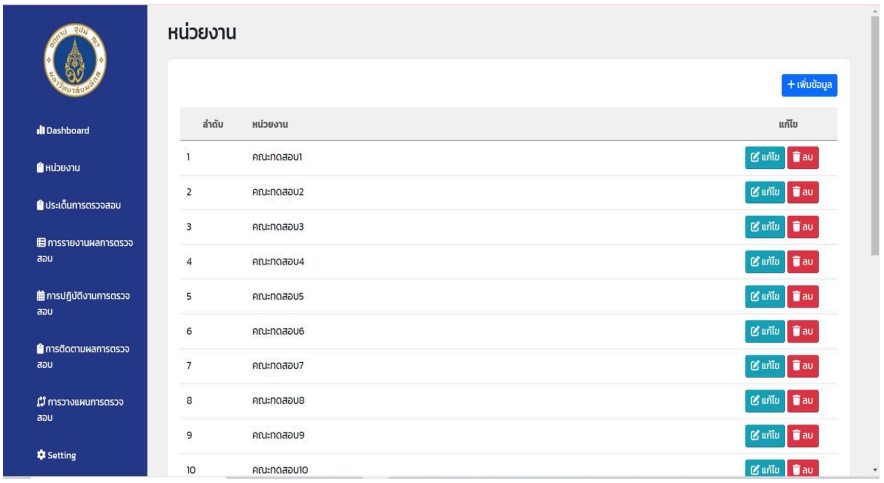



Figure 3.8.1 : Setting page

This settings page provides auditors with the functionality to manage the various faculties and departments that will undergo auditing. Auditors have the ability to add new faculties or departments, edit existing ones, and delete those that are no longer relevant. This feature ensures that the list of entities to be audited is always up-to-date and accurately reflects the current auditing requirements.

### 3.8.2 Audit management page

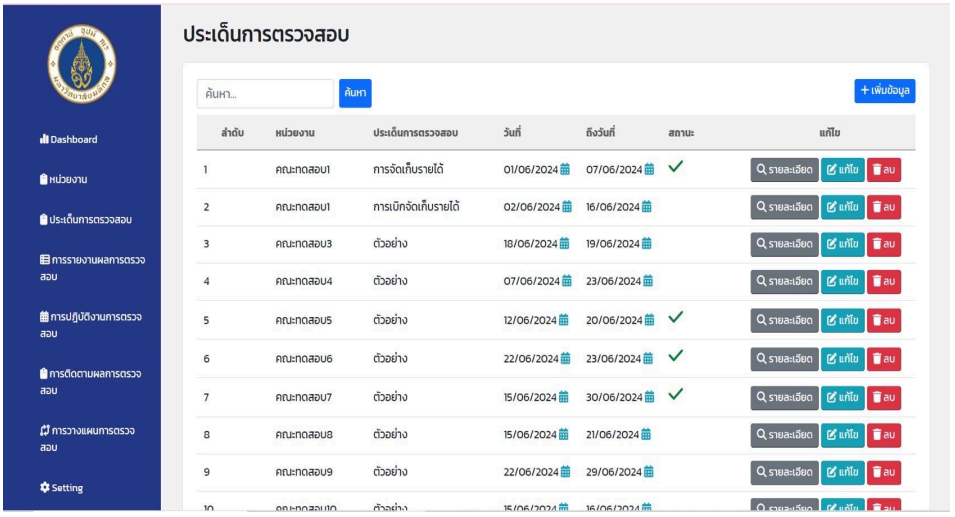



Figure 3.8.2: Audit management page

This audit management page focuses on the topic or the subject that the auditor will be auditing in various faculty. This page collects information such as the faculty

of department name, the topic or the subject of the audit, the date, and the status of each faculty. Moreover, Auditors have the ability to add new faculties or departments, edit existing ones, and delete those that are no longer relevant in the audit detail section as shown in figure 3.9.3 below.

### 3.8.3 Add audit detail page

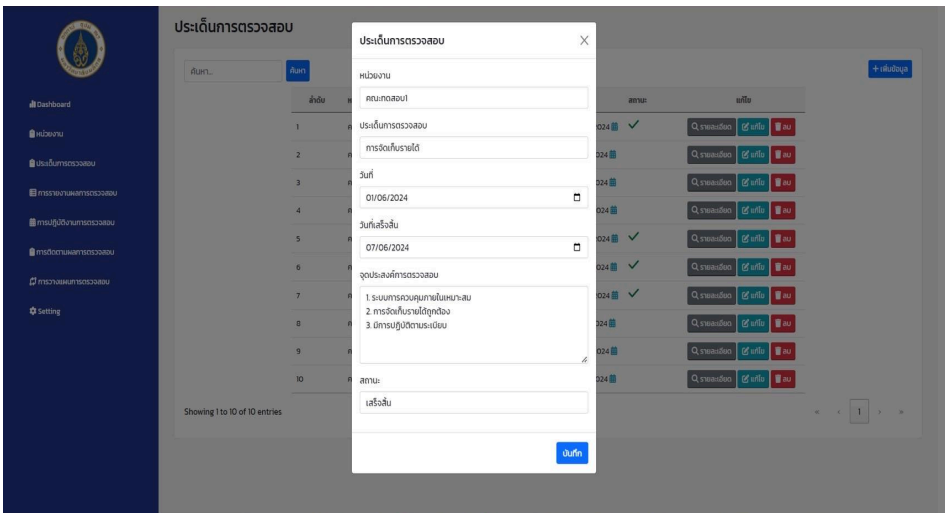



Figure 3.8.3 : Add audit detail page

### 3.8.4 Routing slip

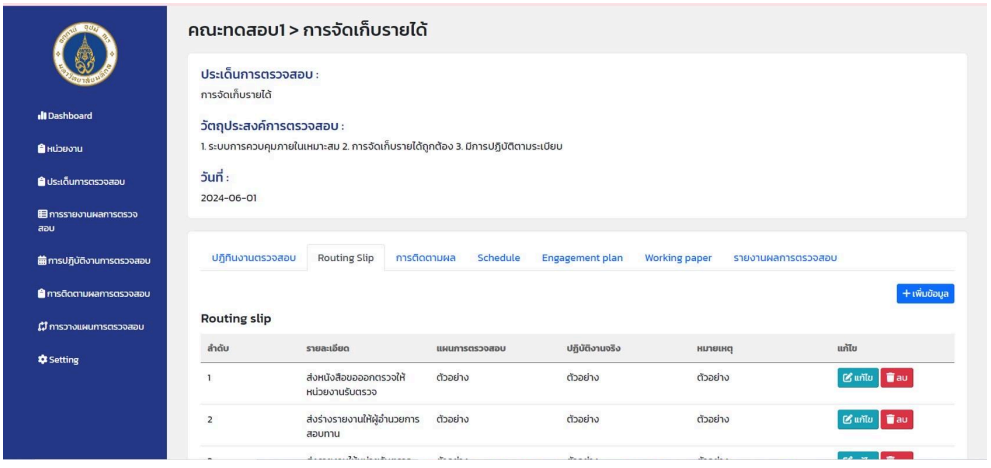



Figure 3.8.4 : Routing slip page

This routing slip page focuses on the details of the audit and monitors the progress of the audit. Users are enabled to input their routing slip data into the table, with the ability to add, edit, and delete entries as needed.

### 3.8.5 Routing slip follow-up

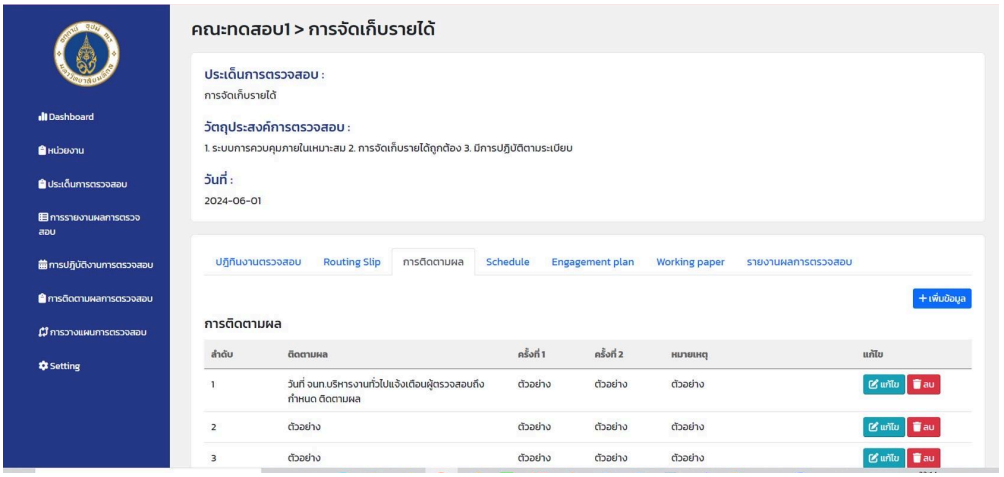



Figure 3.8.5 : Routing slip follow-up page

This routing slip follow-up page focuses on tracking the details and progress of the audit. Users can input their routing slip data into the table and have the ability to add, edit, and delete entries as needed.

### 3.8.6 Engagement plan

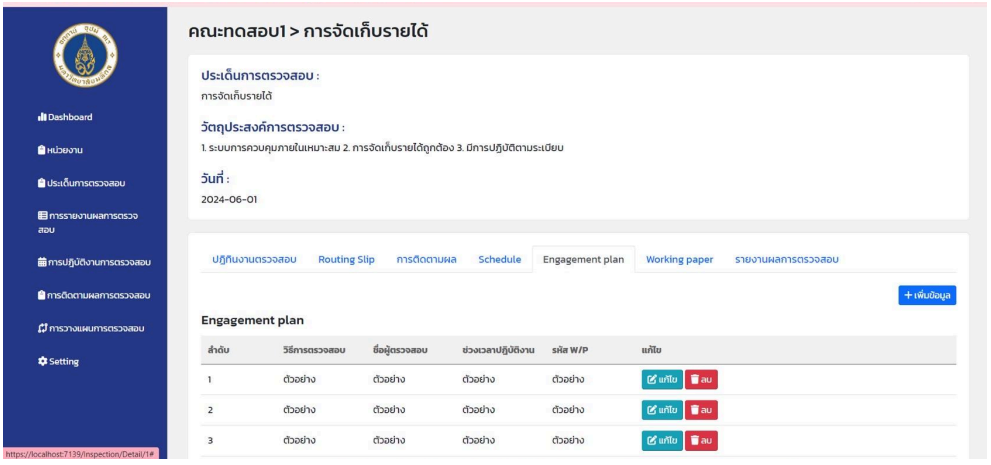



Figure 3.8.6 : Audit detail engagement plan page

This engagement plan page focuses on details of the list of auditors and the methods of audit inspection. Users can input their engagement plan data into the table, with the ability to add, edit, and delete entries as needed.

### 3.8.7 Schedule

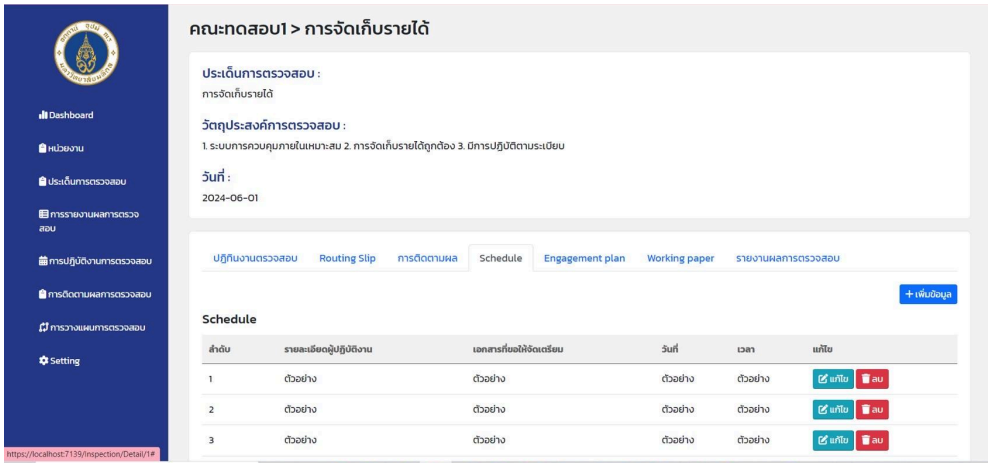



Figure 3.8.7 : Schedule page

This schedule page focuses on the operation details and the list of documents required for audit inspection. Users can input their schedule data into the table, with the ability to add, edit, and delete entries as needed.

### 3.8.8 Working paper

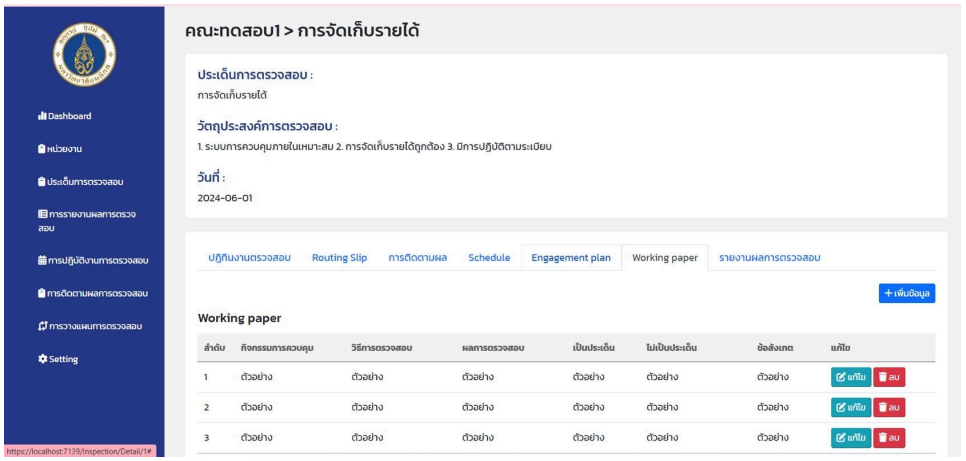



Figure 3.8.8 : Working paper page

The working paper page focuses on documenting inspection results, indicating whether issues are identified or not. Users can input their working paper data into the table, with the ability to add, edit, and delete entries as necessary.

### 3.8.9 Audit report

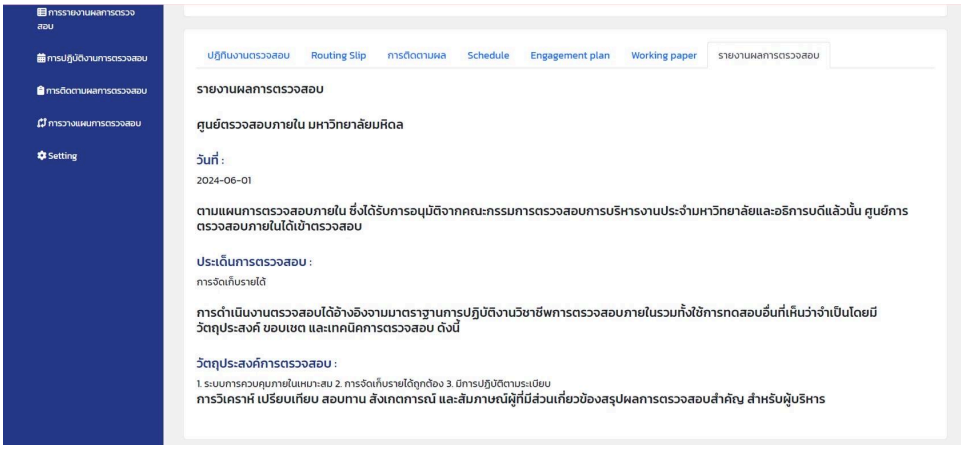



Figure 3.8.9 : Audit detail Report page

The Audit report page generates a comprehensive report for auditors based on department name, audit topic, and audit date configured in the audit management page.

### 3.9 Transition Diagram

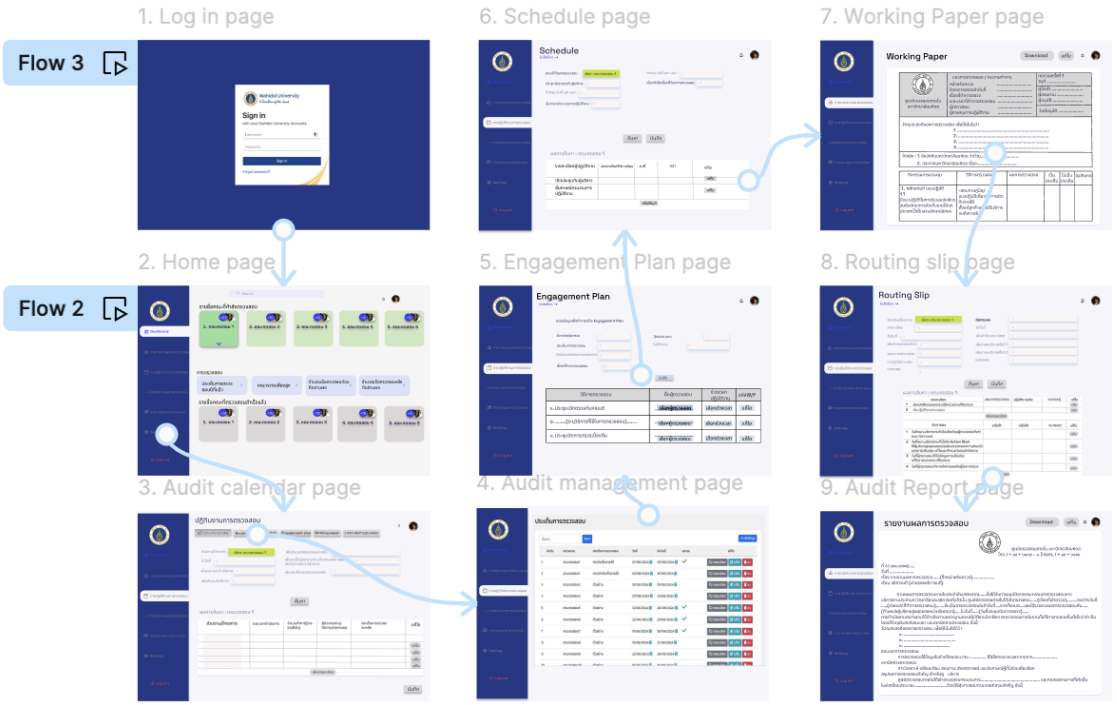



Figure 3.9 : Audit web application transition diagram

This transition diagram shows the sequence of the process of how to audit. The first thing the auditor needs to log in, and then they will see the home page. Secondly, auditors have to set an audit calendar. To do that, the auditor can click on the navigation bar to access the audit calendar page. Thirdly, the auditor then chooses the objective of the audit on the audit management page. Next, after setting the objective, the auditor can use the engagement plan page to store and edit their data in the table. Next, the auditor can set their schedule on the schedule page. After that, the auditor can input their data on the working paper page. Next, the auditor can use the routing slip page to check the progress of the audit. Lastly, auditors can see their audit report on the audit report page.

# CHAPTER 4

## IMPLEMENTATION

In this section, we provide the details of the specifications of the computer of our web server (Table 4.1), editor (Table 4.2), system environment (Table 4.3), framework (Table 4.4), and database management system (Table 4.5). For more information will be discussed in the next section below.

### 4.1 Hardware and System Environment

- Programming and Scripting Tools
    - SQL server management studio

      This is a tool used in Database Engine to create tables to store data. There will be SQL Server Integration Service (SSIS) for preparing data, SQL Server Analysis Service (SSAS) for making Data Models, SQL Server Reporting Service (SSRS) for doing. Summary report including accessing, configuring, managing, administering, and developing
    - .Net framework 4.8.1

      This is a platform used to develop software by connecting databases to computer network connections to make web applications.
    - Visual studio 2022

      This is an IDE (Integrated Development Environment) program used to develop software and web applications.
    - Bootstrap

      It is a tool used for designing websites to be suitable for devices by using HTML, CSS, JS to develop tools for designing website pages (Front-end component library).

- Components
    - Browser
    - Chrome

      Chrome is a website used to display the appearance of web applications.
- Equipment
    - Notebook

Table 4.1: Specifications of the computer of our web server

| Table Name | Description |
|---|---|
| CPU | 4 Core |
| RAM | 8 GB |
| Storage | 1 TB |
| Operating System | Window 10 |

Table 4.2: Editor

| Software Name | Version | Purpose |
|---|---|---|
| Visual Studio 2022 | 17.9.6 | To perform coding |

Table 4.3: Database Management System

| Software Name | Version | Purpose |
|---|---|---|
| SQL Server Management Studio | 19.3 | Used to store data and run the database service on the web server. |

4

| Software Name | Version | Purpose |
|---|---|---|
| C# | 11 | Using a programming language to write frontend and backend. |

Table 4.5: Framework

| **Library name** | **Version** | **Purpose** |
|---|---|---|
| .Net framework | 4.8.1 | This feature enables software developers to create applications in one programming language, ensuring compatibility with code written in other languages. |
| Bootstrap 5 | 5.3 | Making web development faster and easier. |

In this section, we use the architecture design for the standard .net framework(Table 4.1) to discuss our website implementation. For more information about architecture is shown in the list below.

## 4.2 Website Implementation

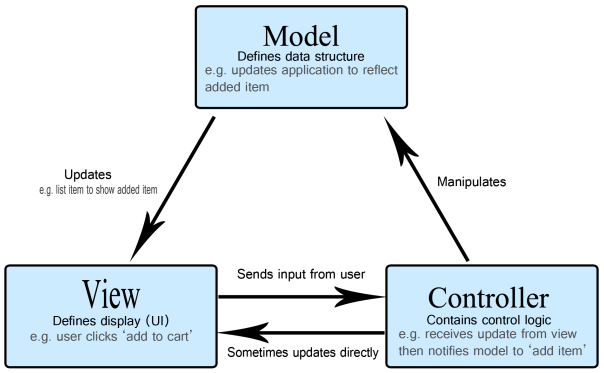
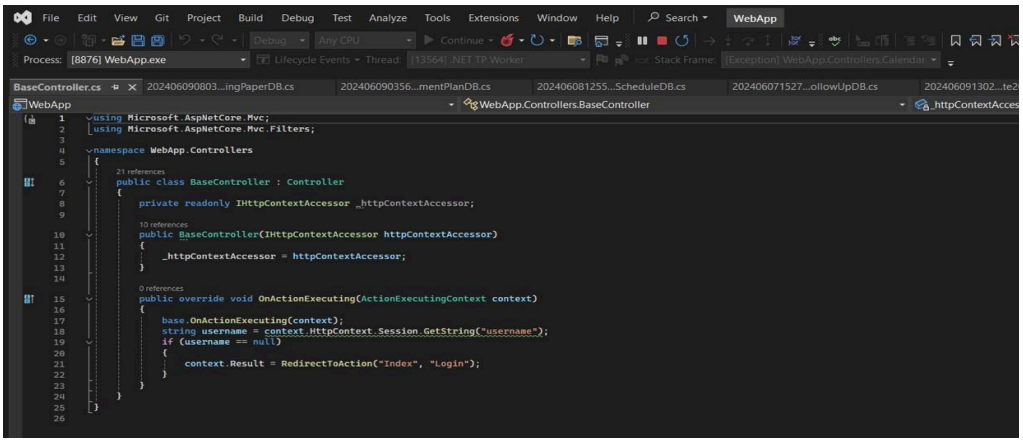



Figure 4.2: Architecture Design for standard .net framework (MVC)

## 4.3 Controller

A controller is responsible for controlling the way that a user interacts with an MVC application. A controller contains the flow control logic for an ASP.NET MVC application. A controller determines what response to send back to a user when a user makes a browser request. Figure 4.3.1, Figure 4.3.2, Figure 4.3.3, Figure 4.3.4, Figure 4.3.5, Figure 4.3.6, Figure 4.3.7, Figure 4.3.8, Figure 4.3.9, Figure 4.3.10, Figure 4.3.11, Figure 4.3.12, Figure 4.3.13 represent a code of controller for each page of web application.

**Base controller**

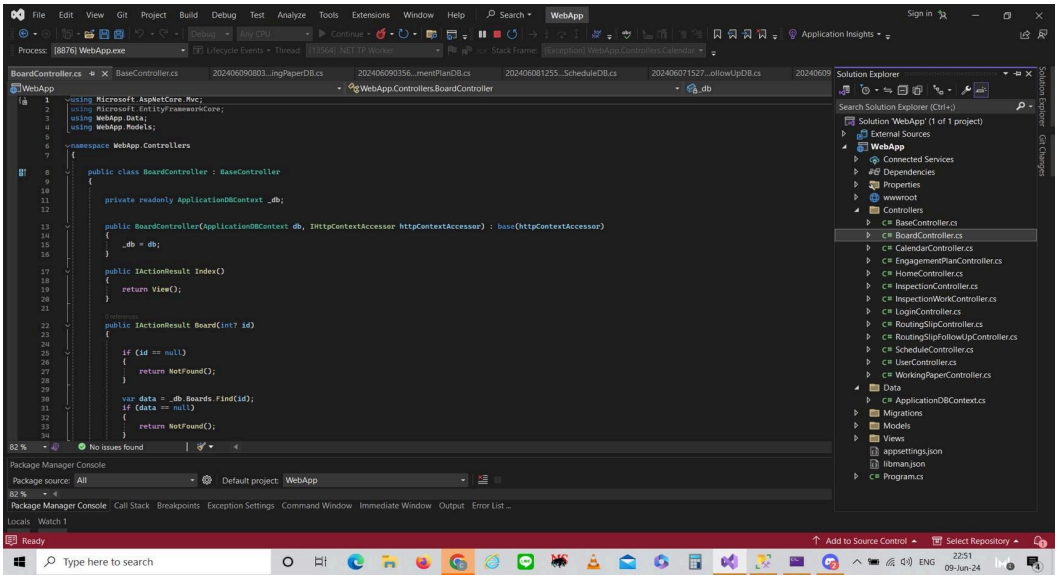


```
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Filters;

namespace WebApp.Controllers
{
    21 references
    public class BaseController : Controller
    {
        private readonly IHttpContextAccessor _httpContextAccessor;

        10 references
        public BaseController(IHttpContextAccessor httpContextAccessor)
        {
            _httpContextAccessor = httpContextAccessor;
        }

        0 references
        public override void OnActionExecuting(ActionExecutingContext context)
        {
            base.OnActionExecuting(context);
            string username = context.HttpContext.Session.GetString("username");
            if (username == null)
            {
                context.Result = RedirectToAction("Index", "Login");
            }
        }
    }
}
```

Figure 4.3.1: Base controller

The figure shows BaseController class is part of an authorization filter implemented using the on action executing method. It checks if a user is authenticated by looking for a session variable named "username". If the user is not logged in.

**Board controller**



```
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using WebApp.Data;
using WebApp.Models;

namespace WebApp.Controllers
{
    public class BoardController : BaseController
    {
        private readonly ApplicationDBContext _db;

        public BoardController(ApplicationDBContext db, IHttpContextAccessor httpContextAccessor) : base(httpContextAccessor)
        {
            _db = db;
        }

        public IActionResult Index()
        {
            return View();
        }

        public IActionResult Board(int? id)
        {
            if (id == null)
            {
                return NotFound();
            }

            var data = _db.Boards.Find(id);
            if (data == null)
            {
                return NotFound();
            }
```

Figure 4.3.2: Board controller

This figure shows the method to find a board with the specified ID from the database using the Find method of the db.Boards property. If a board is found, the code snippet you provided doesn't show what the method does with it. However, if no board is found, the method returns a NotFound result.

**Calendar controller**

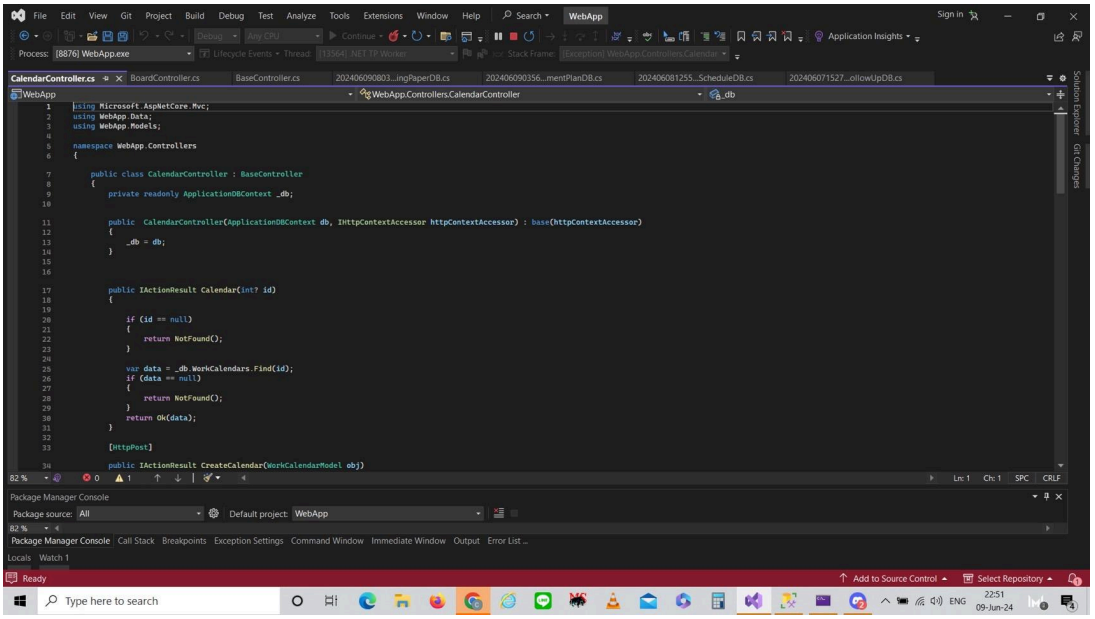



Figure 4.3.3:Calendar controller

This figure shows the method of managing calendars. The Calendar method can retrieve a calendar by its ID from the database, and the CreateCalendar method can create a new calendar record.

**Engagement plan controller**

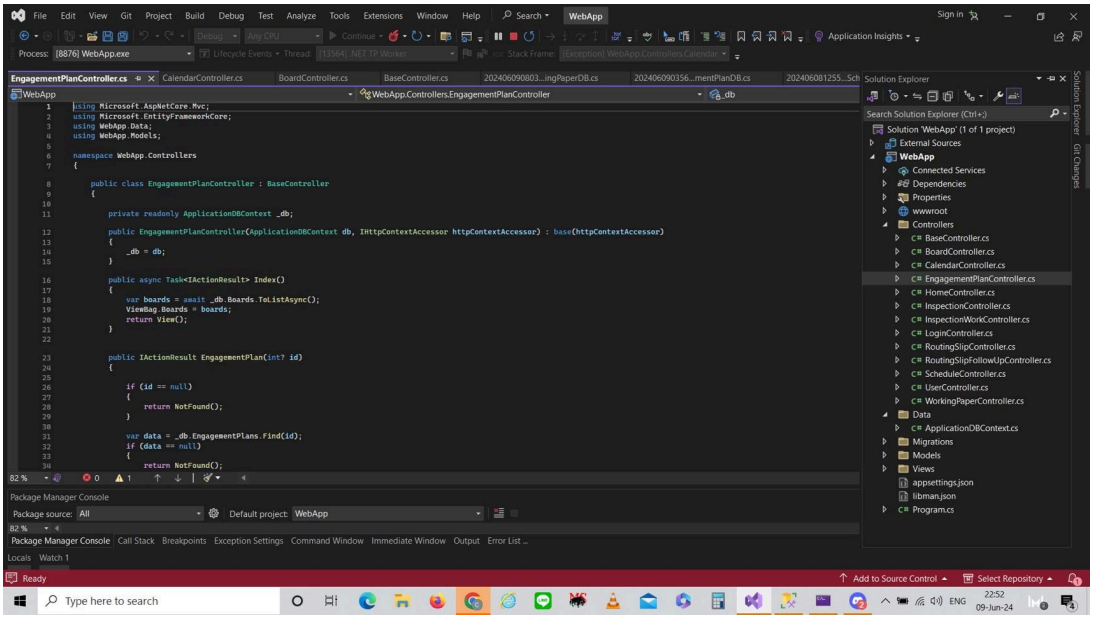



Figure 4.3.4:Engagement plan controller

The figure shows a EngagementPlanController class is responsible for handling HTTP requests related to engagement plans, and it inherits some common functionality from the base controller class.

**Home controller**

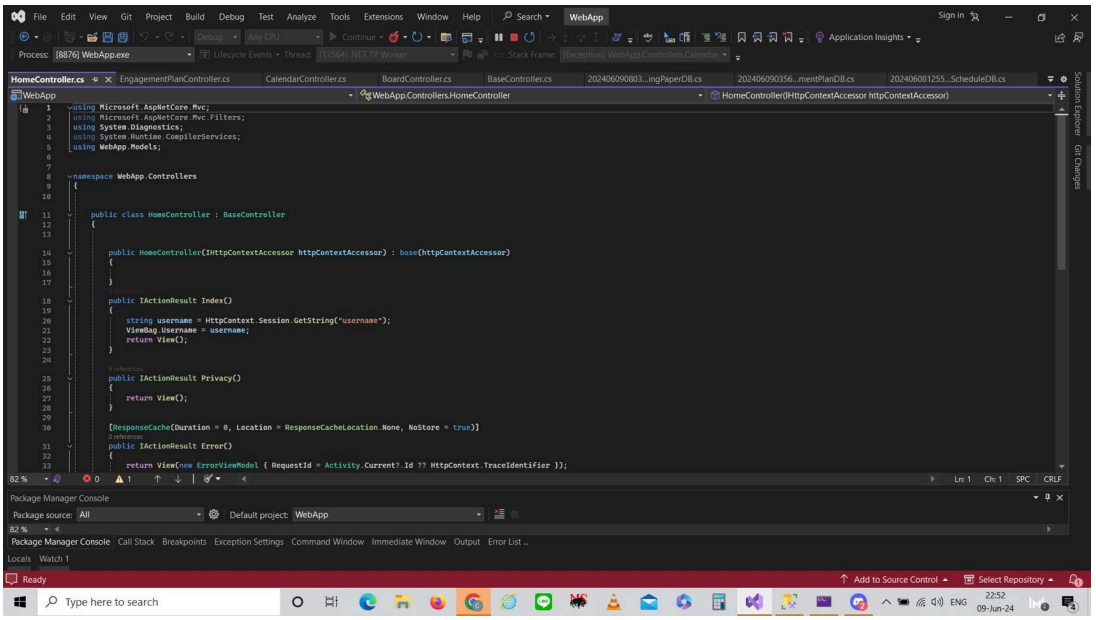



Figure 4.3.5:Home controller

The figure shows a partial definition of the ScheduleDB class within the Models folder. This class likely serves as a data model for representing schedule information.The class name is HomeController, and it inherits from the BaseController class. In ASP.NET Core MVC, controllers handle HTTP requests from the web browser.

**Audit controller**

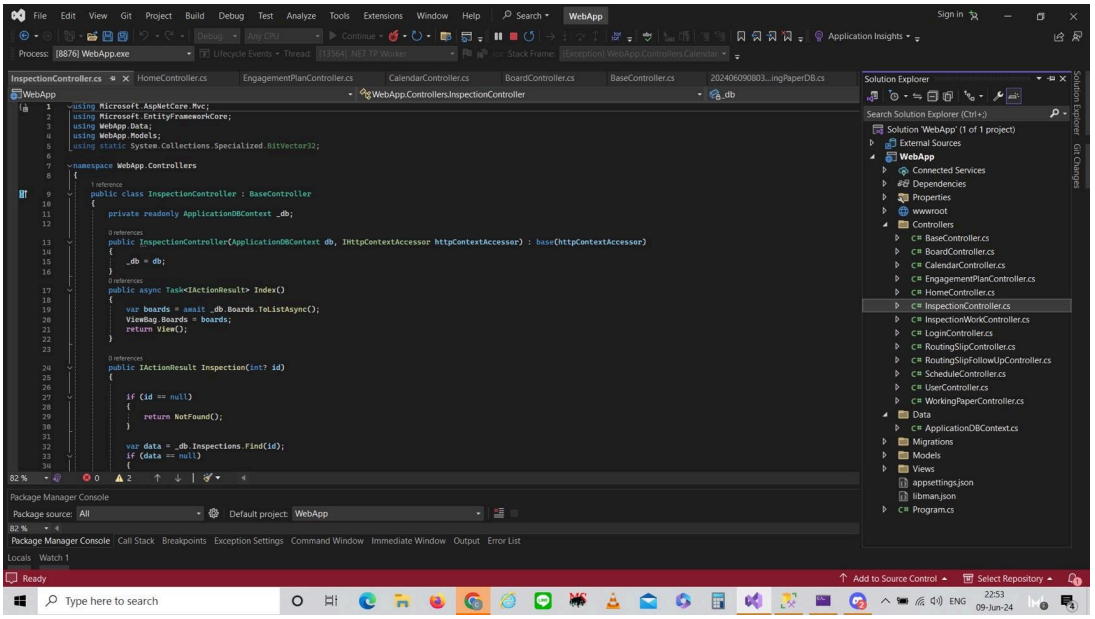



Figure 4.3.6:Audit controller

This figure defines an AuditController class that inherits from a base controller (BaseController) and handles functionality related to inspections. It retrieves inspection data from a database and interacts with views to display or potentially manage inspections.

**Audit work controller**

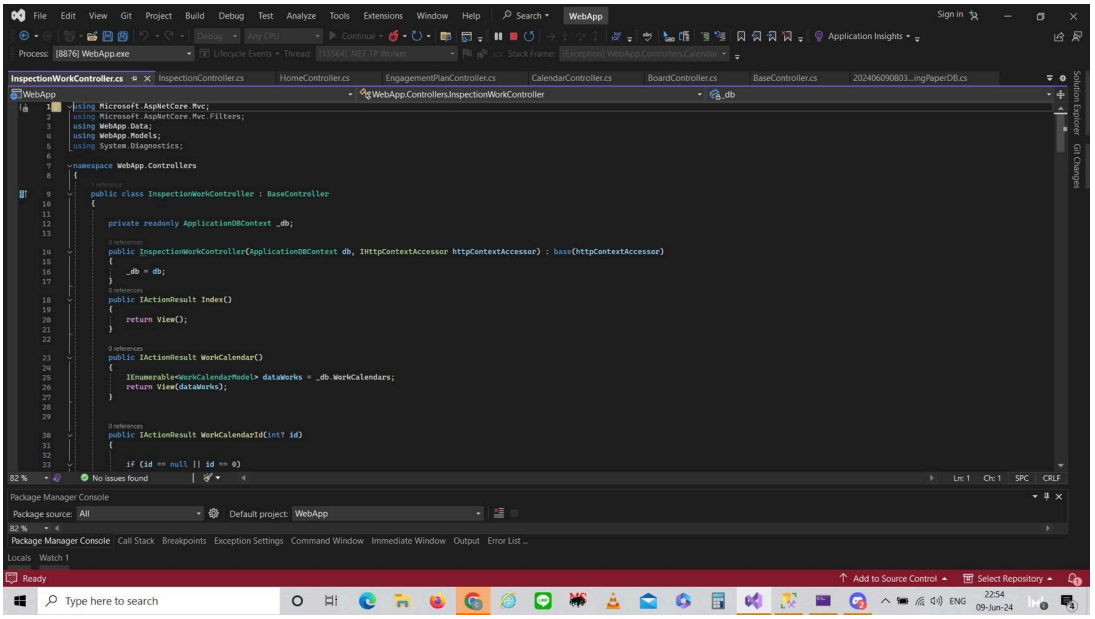



Figure 4.3.7:Audit work controller

This figure defines a AuditWorkController class that inherits from a base controller (BaseController) and handles functionality related to work calendars. It retrieves work calendar data from a database and interacts with views to display them.

**Login controller**

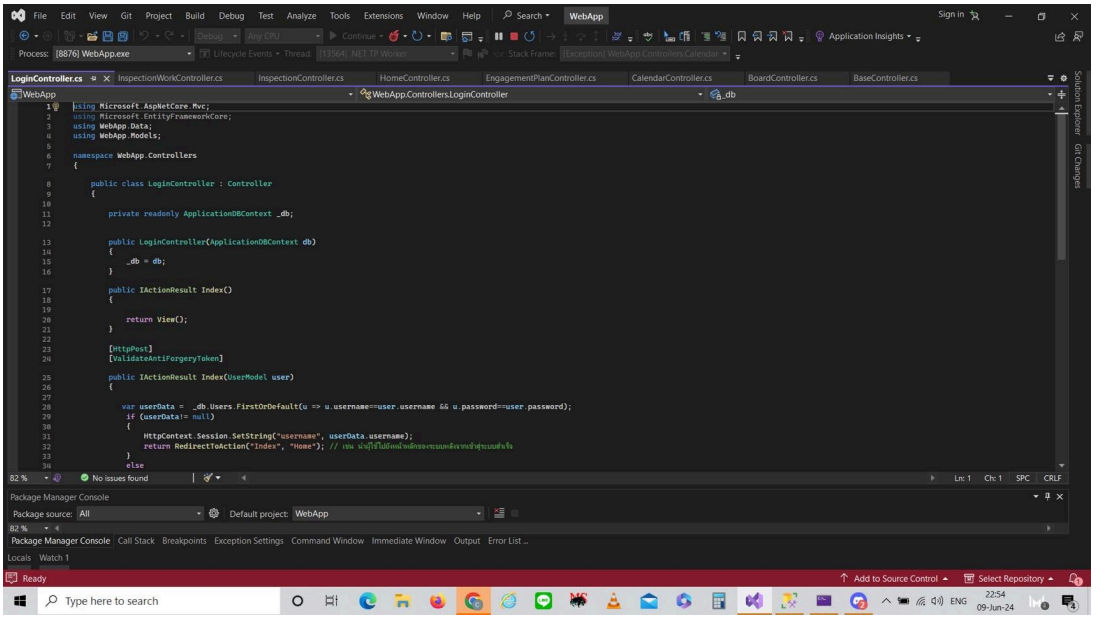



Figure 4.3.8:Login controller

This figure defines a LoginController class that handles user login functionality. It likely uses a db context to interact with a database (potentially for user authentication) and redirects to different views based on login success or failure.

**Routing slip controller**

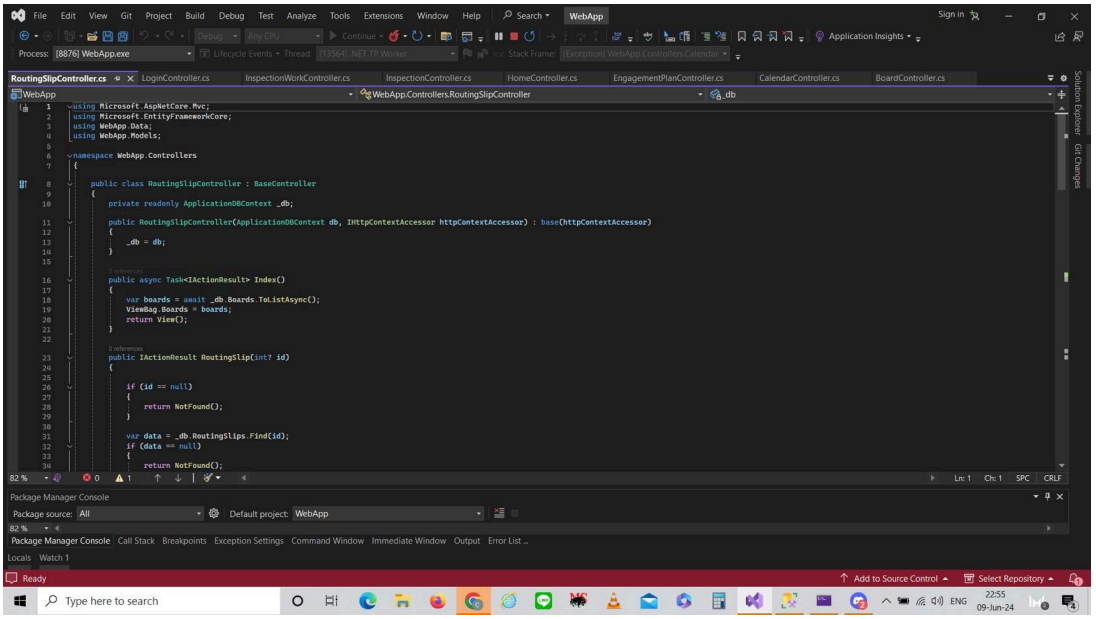



Figure 4.3.9:Routing slip controller

This figure defines a SlipController class that inherits from a base controller (BaseController) and handles functionality related to routing slips. It retrieves routing slip data from a database and interacts with views to display them (potentially listing all slips or viewing a specific slip).

**Routing slip follow up controller**

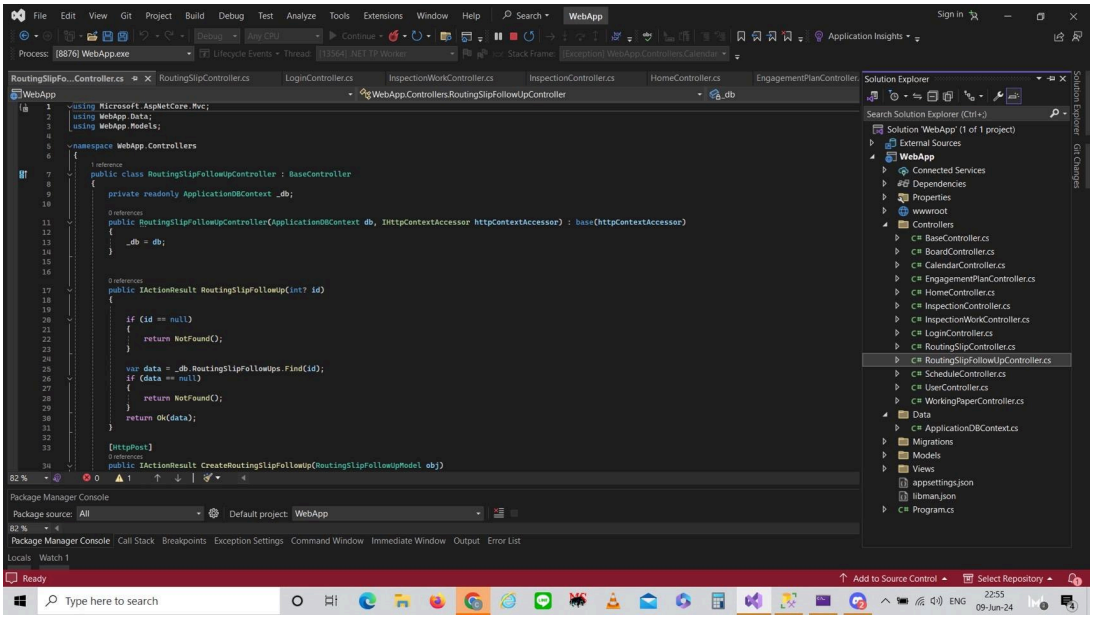



Figure 4.3.10:Routing slip follow up controller

This figure defines a RoutingSlipFollowUpController class that likely deals with following up on routing slips. It retrieves routing slip follow-up data from a database and potentially interacts with views to display or manage follow-up information.

**Schedule controller**

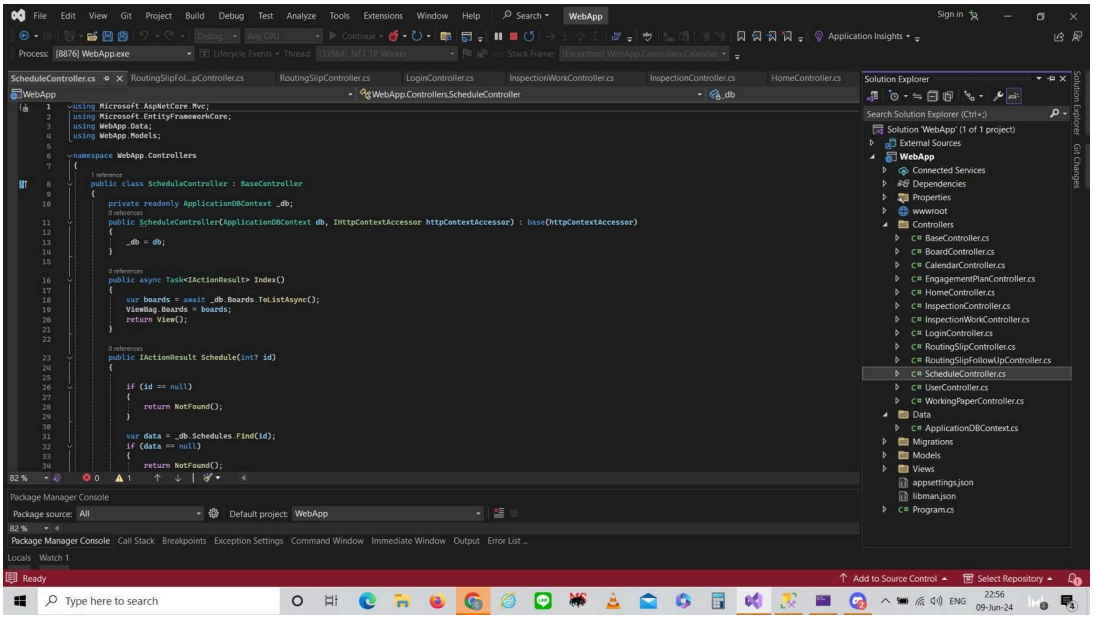



Figure 4.3.11:Schedule controller

This figure defines a ScheduleController class that inherits from a base controller (BaseController) and handles scheduling functionality, likely related to inspections. It retrieves schedule data from a database and interacts with views to display or manage schedules.

**User controller**

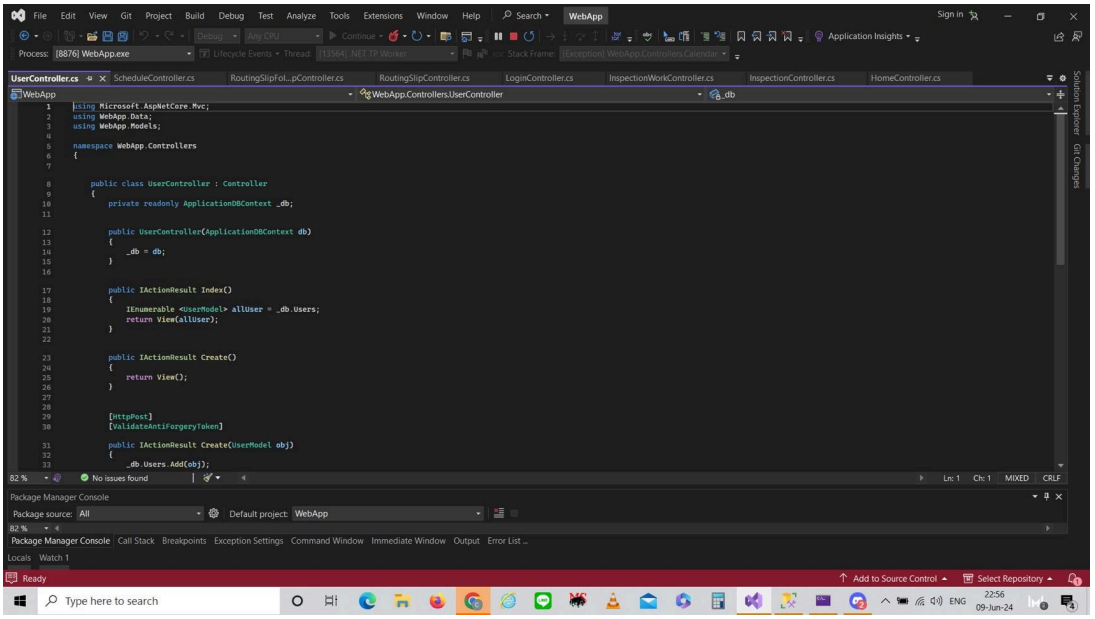



Figure 4.3.12:User controller

This figure defines a UserController class that handles user management functionalities. It retrieves user data from a database and interacts with views to display or manage users.

**Working paper controller**

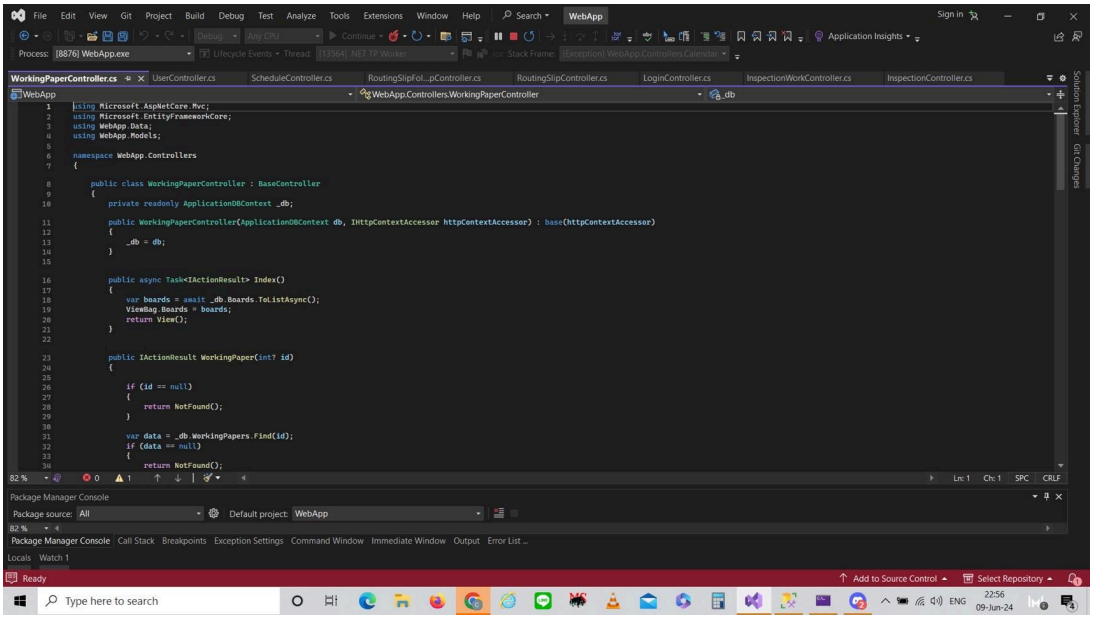



Figure 4.3.13:Working paper controller

This figure defines a controller class for an ASP.NET Core MVC application that handles requests related to working papers. The Index method retrieves a list of boards from a database, and the WorkingPaper method retrieves a specific working paper by ID.

## 4.4 View

The view layer provides the UI necessary to interact with the application. It includes components needed to display the data and enables users to interact with that data.Figure 4.4.1, Figure 4.4.2, Figure 4.4.3, Figure 4.4.4, Figure 4.4.5, Figure 4.4.6, Figure 4.4.7, Figure 4.4.8, Figure 4.4.9, Figure 4.4.10 represent a code of view layer for each page of web application.

## Board view

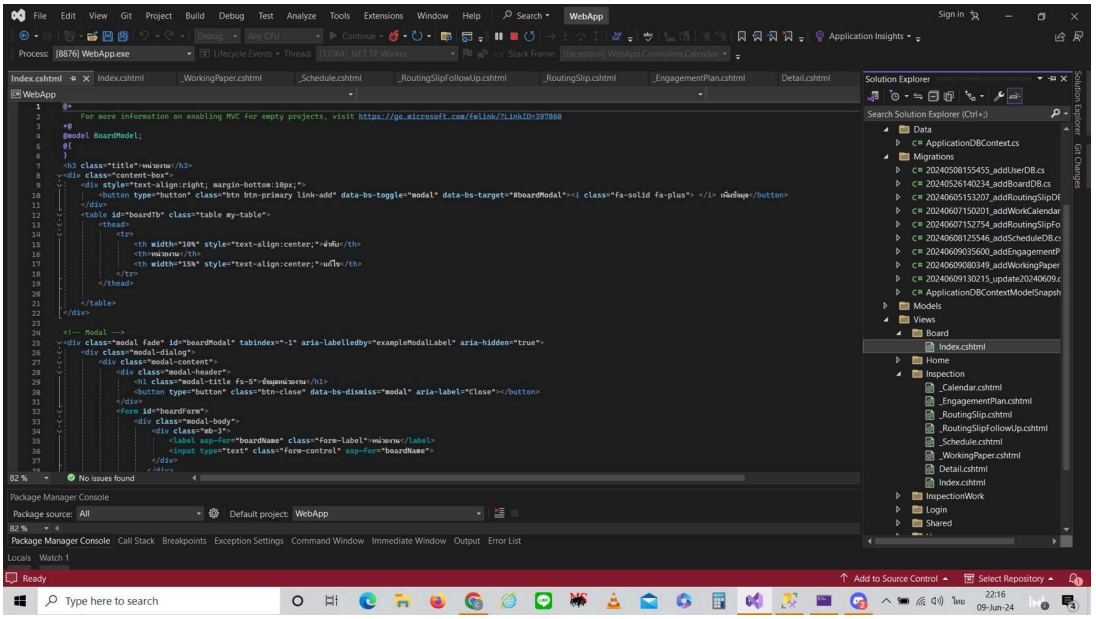


Figure 4.4.1: Board view

The Board view in the web application displays a list of faculties or departments. It presents information related to academic units within the organization, allowing users to view and manage faculty details efficiently through the application's interface.

## Calendar view

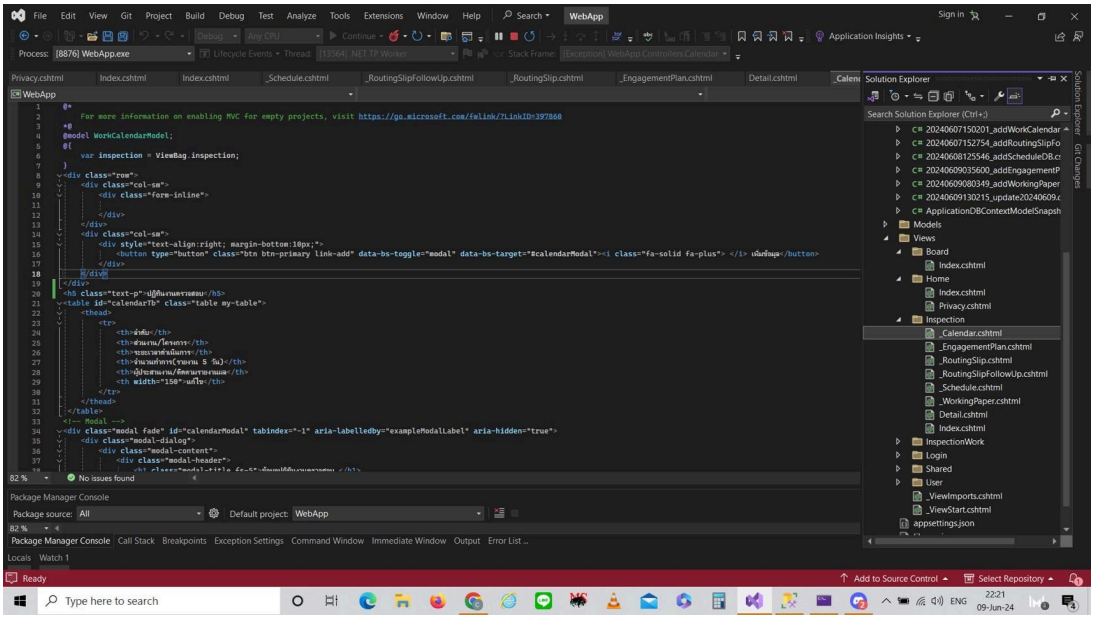


Figure 4.4.2: Calendar view

The Calendar view in the web application presents a table showing faculty names, durations, and auditors' names. This view provides a structured overview of scheduled activities and audits, facilitating organized management and planning within the application.

**Engagement plan view**

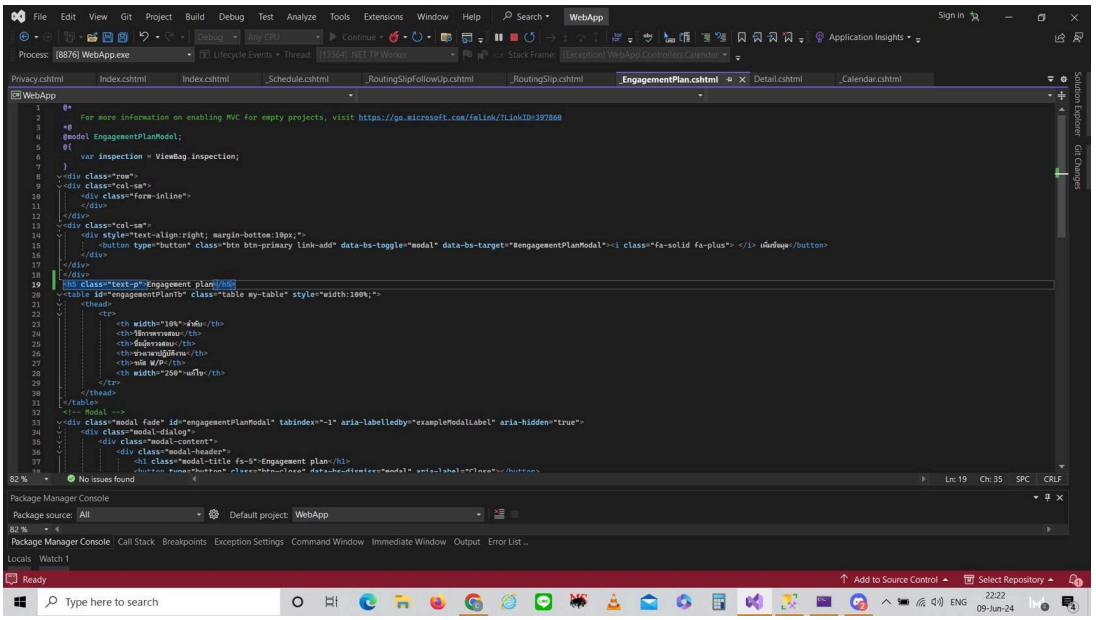


Figure 4.4.3: Engagement plan view

The Engagement Plan view in the web application displays information regarding auditing procedures and plans. It provides details such as engagement procedures, inspector names, and work hours allocated to each procedure. This view enables users to manage and monitor auditing engagements effectively through the application's interface.

**Audit Routing Slip view**

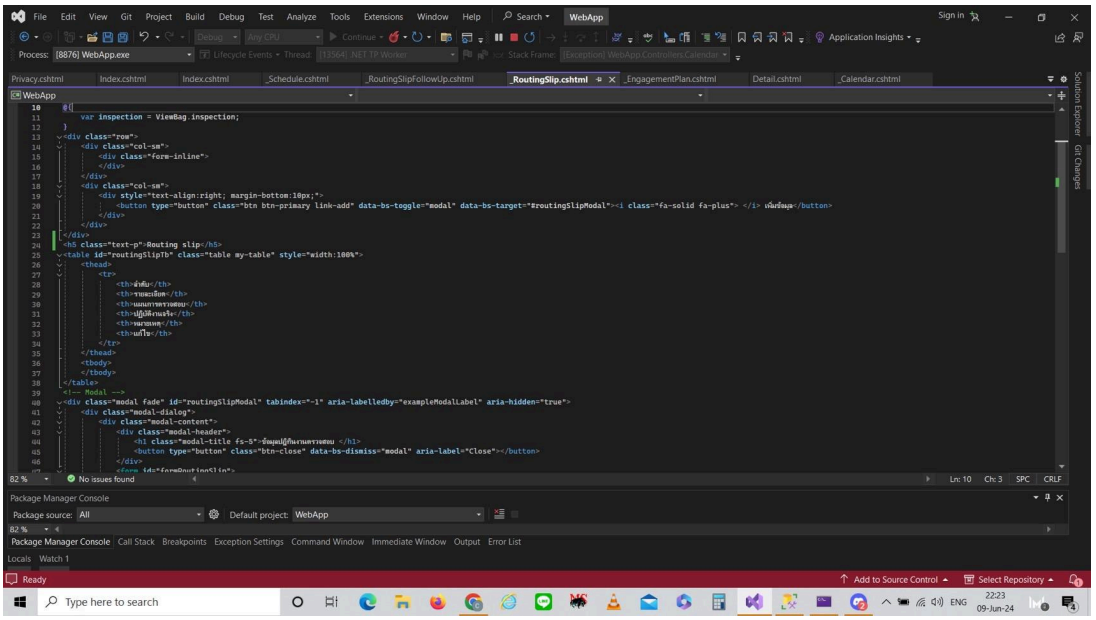


Figure 4.4.4: Audit Routing Slip view

The Audit Routing Slip view in the web application provides an interface to manage and track routing slips used during audits. It includes fields for detailing audit plans, work plans, and comments related to auditing activities. This view allows auditors to input, edit, and delete routing slip data as needed, ensuring thorough documentation and follow-up throughout the auditing process.

**Routing Slip FollowUp view**

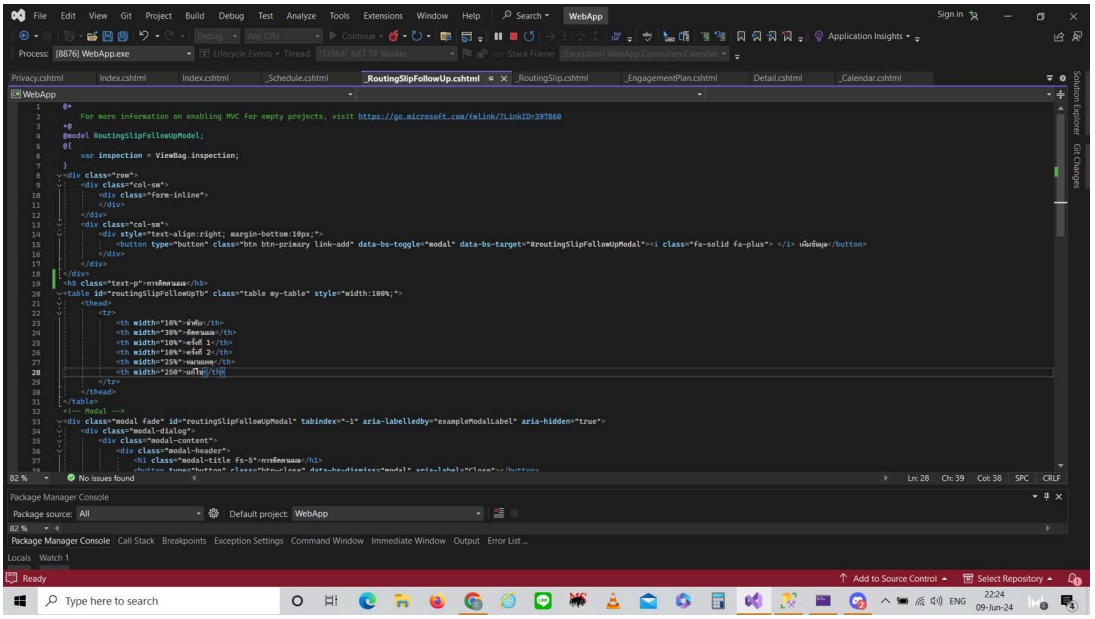


Figure 4.4.5: Routing Slip FollowUp view

The Routing Slip Follow-Up view in the web application facilitates the monitoring and management of follow-up activities after audits. It allows users to input details such as follow-up actions, timelines (TimeN1 and TimeN2), and comments related to audit findings. This view enables auditors to track and update follow-up progress effectively, ensuring that audit recommendations are addressed and resolved in a timely manner.

**Schedule view**

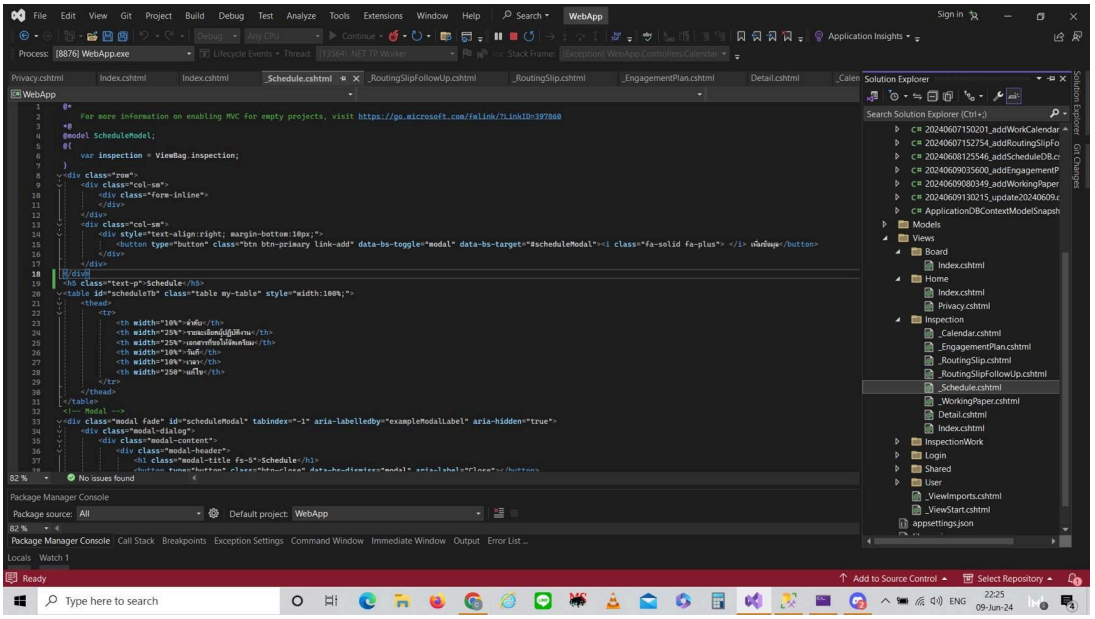


Figure 4.4.6: Schedule view

The Schedule view in the web application provides a structured display of scheduled activities related to audits. It includes fields for faculty names, durations, and auditors' names, offering a clear overview of planned audit schedules. This view allows users to manage and coordinate audit activities efficiently, ensuring that audits are conducted according to predefined timelines and with assigned auditors.

**Working paper view**

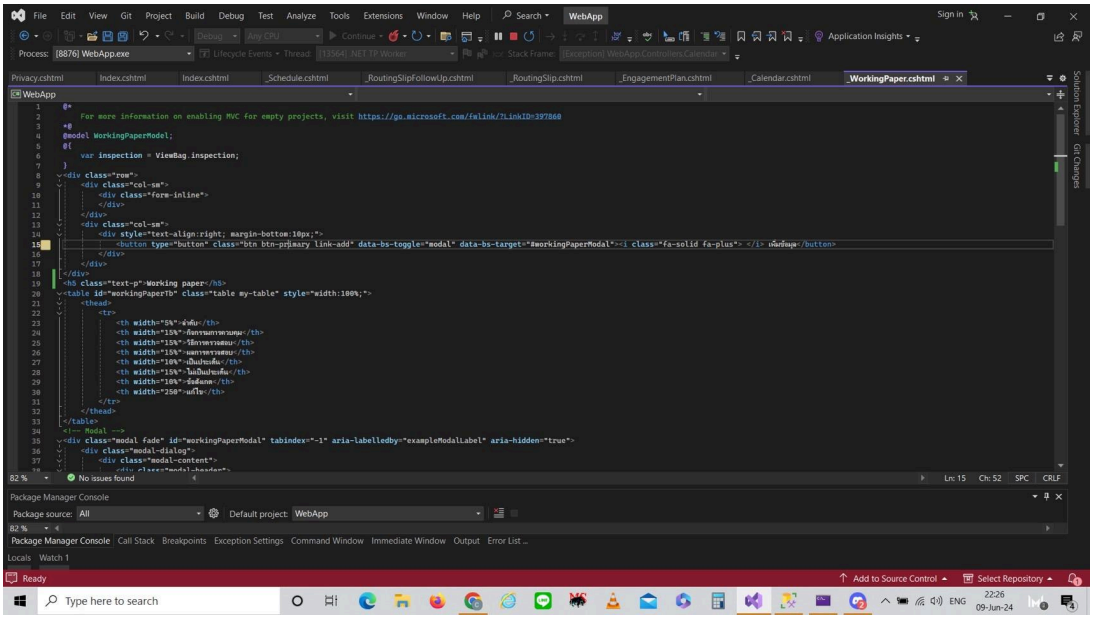


Figure 4.4.7: Working paper view

The Working Paper view in the web application presents a detailed overview of audit-related documentation and procedures. It includes fields for documenting audit activities, specifying required documents, recording inspection results, detailing identified issues, and providing remarks or additional notes. This view facilitates comprehensive tracking and management of audit processes, ensuring thorough documentation and adherence to auditing standards.

**Detail view**

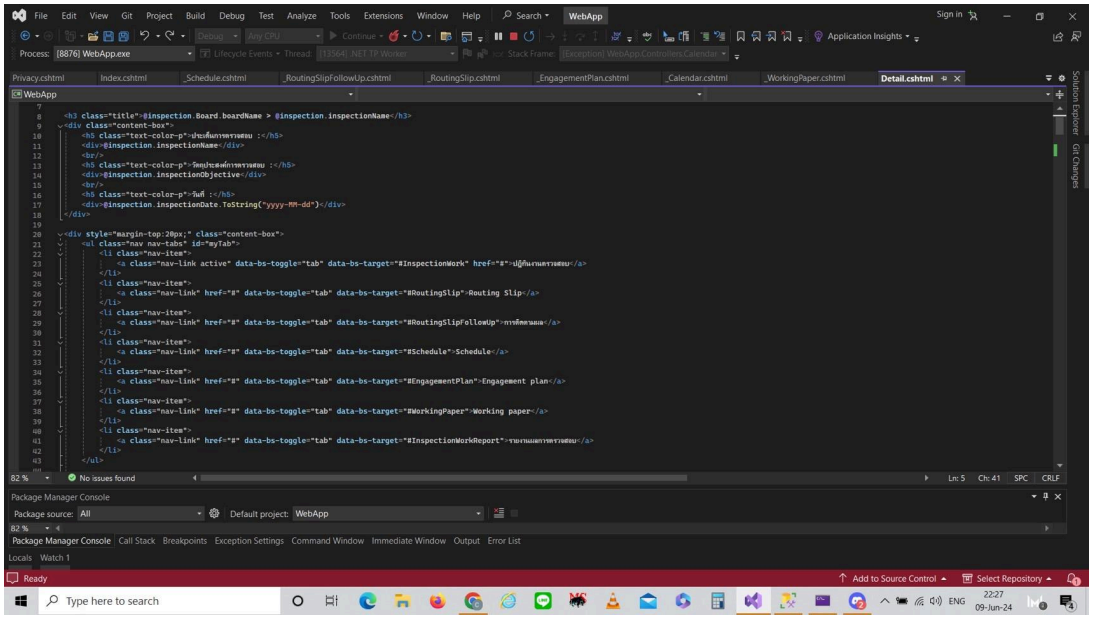


Figure 4.4.8: Detail view

The Detail view in the web application provides a focused display of specific information related to various aspects of auditing. It allows users to view detailed data such as audit plans, inspection details, routing slip information, schedules, and other pertinent audit-related data. This view enables auditors and administrators to access comprehensive information quickly and efficiently, decision making and ensuring thorough oversight of audit processes.

**Index view**

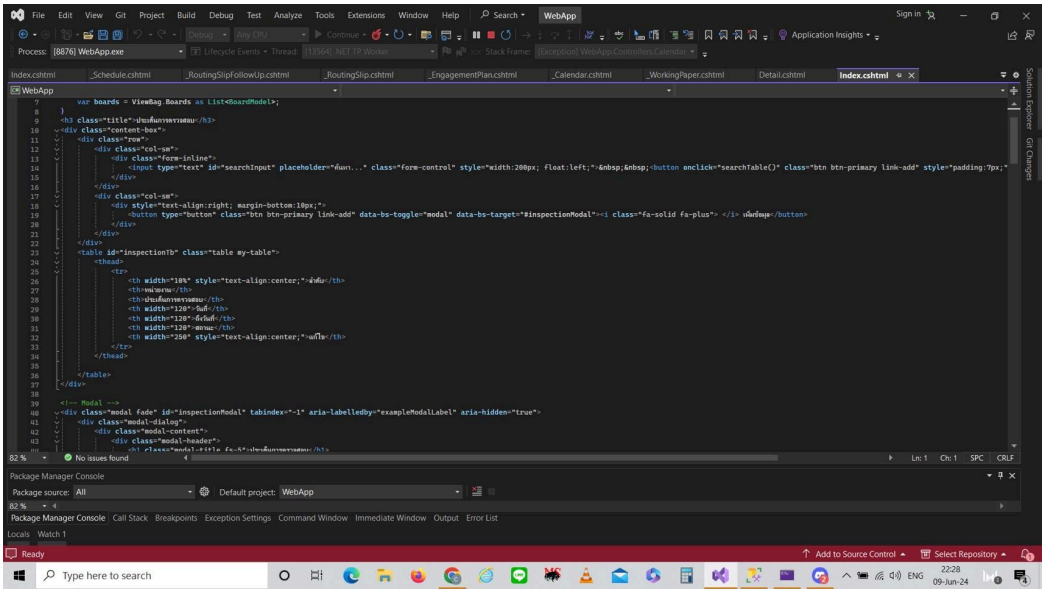


Figure 4.4.9: Index view

The Index view in the web application serves as the main landing page or dashboard that provides an overview of essential information and functionalities. It typically includes summarized data and quick links to key sections such as audits, schedules, reports, and user management. The Index view aims to offer a centralized hub where users can navigate and access various features of the auditing system efficiently. It serves as a starting point for users to interact with and manage different aspects of the audit process.

**Login view**

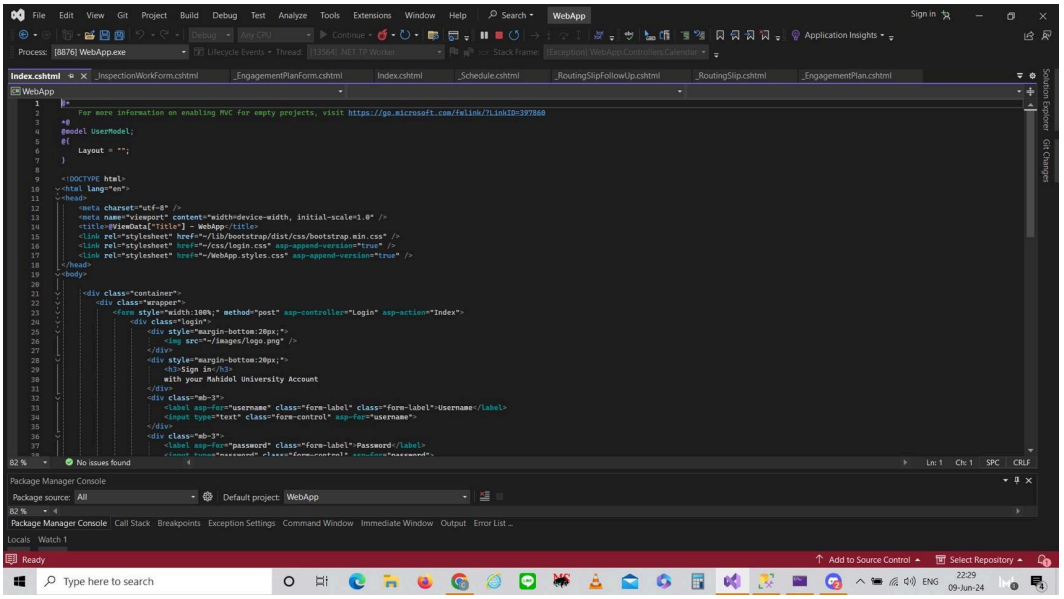


Figure 4.4.10: Login view

The Login view is where users enter their username and password to securely access the web application. Successful authentication grants them access to the system's features. It serves as the starting point for users to manage audit-related tasks and information.

## 4.5 Model

This section is about storing various information entered on each page. Each page stores different class and object and other information which includes collecting information of Board, Engagement plan, Error view, Audit, Routing Slip follow-up, Routing slip, Schedule, User, Working Calendar, and Workingpape.

### Board model

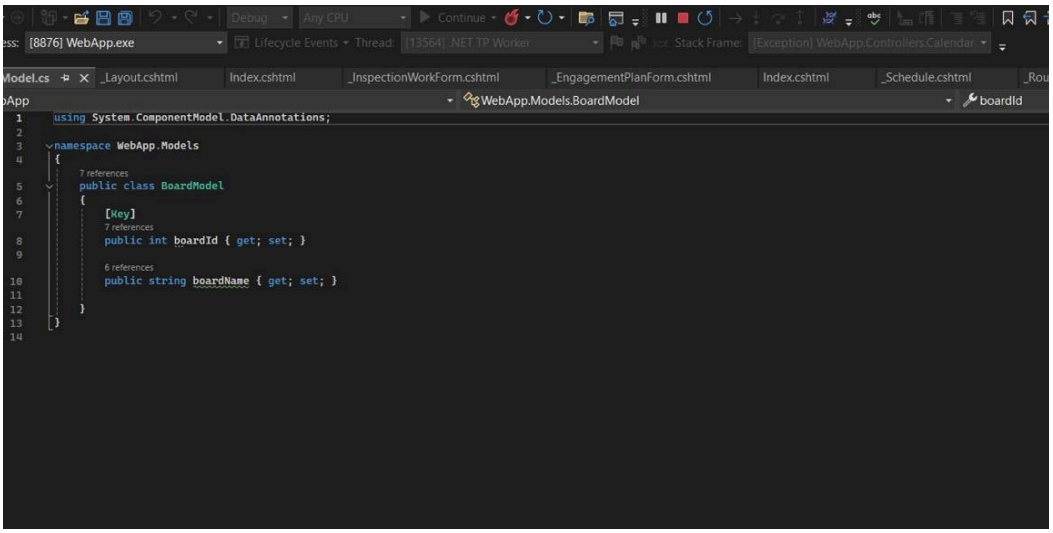



Figure 4.5.1: Board model

The Board model primarily stores information related to faculty names or departments. It is designed to capture and manage data regarding various academic units within the organization, facilitating organized and structured storage of this essential information.

**Engagement plan model**

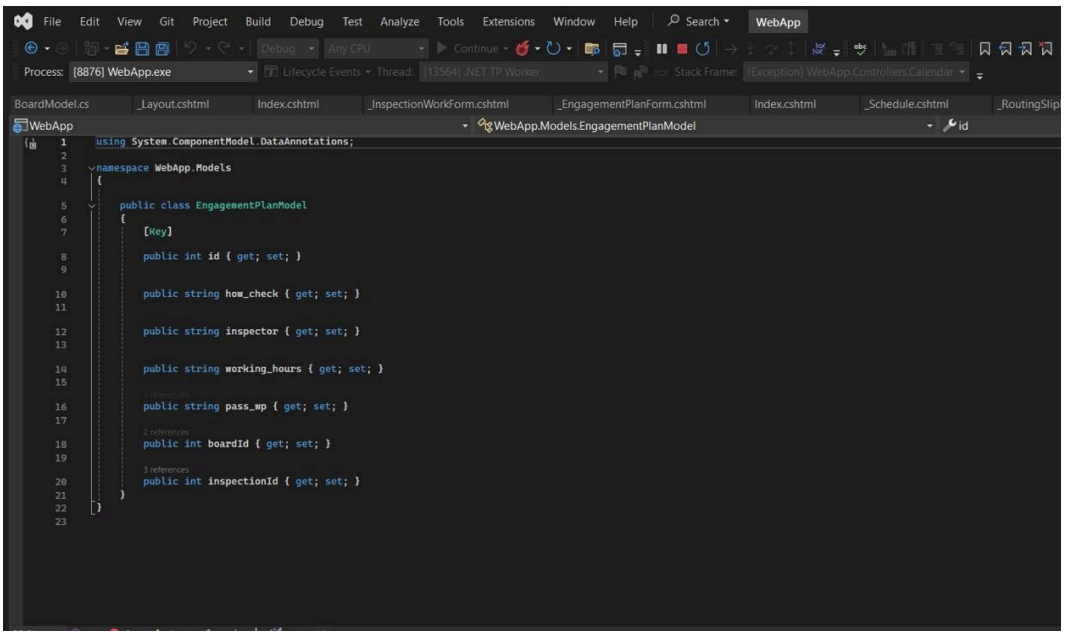



Figure 4.5.2:Engagement plan model

The Engagement Plan model is used to store all data related to the engagement plan procedures. It stores and manages information concerning the specific plans and procedures followed during auditing engagements, ensuring clarity and organization in the process.

**Error view model**

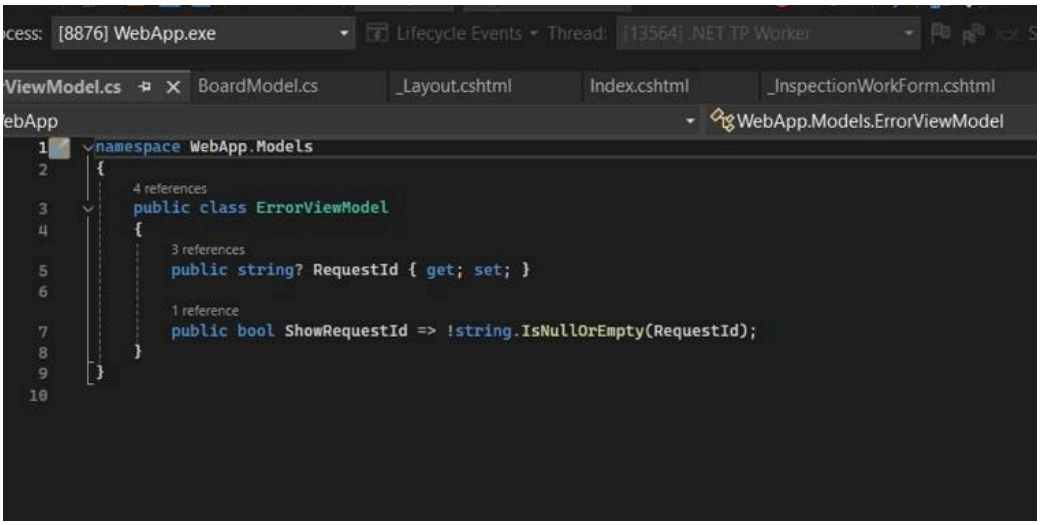



Figure 4.5.3: Error view model

The Error View model specifically contains and displays the request ID. It focuses on managing and presenting information related to individual request IDs, ensuring clarity and ease of access within the application.

**Audit model**

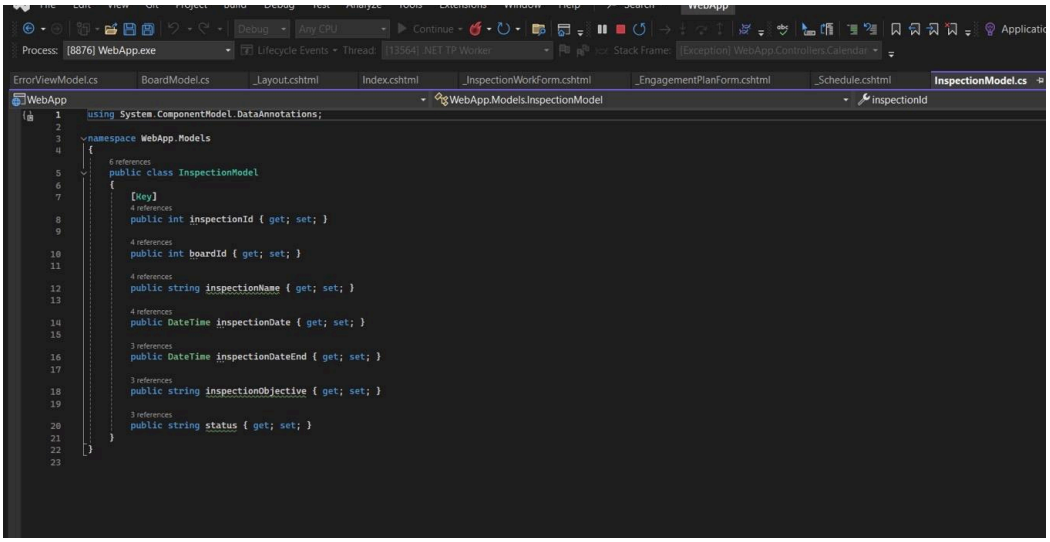
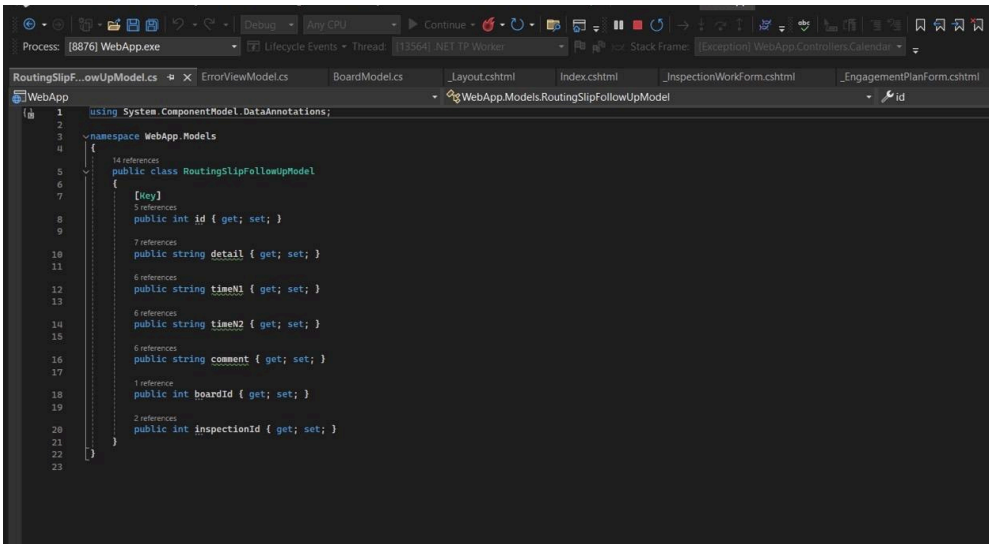



Figure 4.5.4: Audit model

The Audit model encompasses essential fields such as inspection ID, board ID, inspection name, inspection date start, inspection date end, inspection objective, and status. It serves to manage and track details pertaining to inspections conducted within the system, ensuring comprehensive oversight and management of audit processes.

**Routing Slip follow-up model**

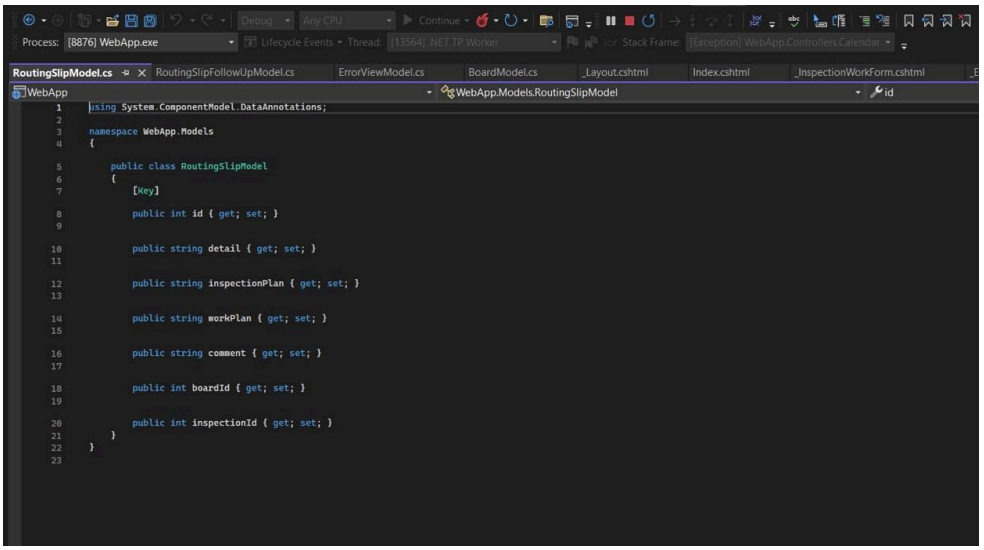


```
using System.ComponentModel.DataAnnotations;

namespace WebApp.Models
{
    14 references
    public class RoutingSlipFollowUpModel
    {
        [Key]
        5 references
        public int id { get; set; }

        7 references
        public string detail { get; set; }

        6 references
        public string timeN1 { get; set; }

        6 references
        public string timeN2 { get; set; }

        6 references
        public string comment { get; set; }

        1 reference
        public int boardId { get; set; }

        2 references
        public int inspectionId { get; set; }
    }
}
```

Figure 4.5.5: Routing Slip follow-up model

The routing slip follow-up model is used to store all data related to the routing slip follow-up procedures. It stores and manages information concerning the specific plans and procedures.

**Routing slip model**



```
using System.ComponentModel.DataAnnotations;

namespace WebApp.Models
{
    public class RoutingSlipModel
    {
        [Key]

        public int id { get; set; }


        public string detail { get; set; }


        public string inspectionPlan { get; set; }


        public string workPlan { get; set; }


        public string comment { get; set; }


        public int boardId { get; set; }


        public int inspectionId { get; set; }
    }
}
```

Figure 4.5.6: Routing slip model

The routing slip model is used to store all data related to the routing slip procedures. It stores and manages information concerning the specific procedures.

**Schedule model**

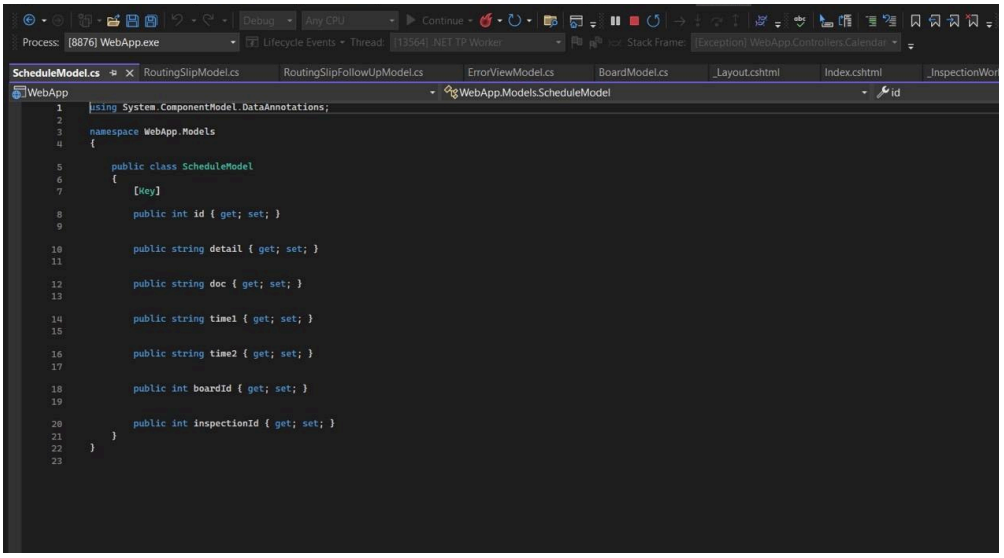



Figure 4.5.7: Schedule model

The schedule model is used to store all data related to the schedule procedures. It stores and manages information concerning the specific schedule.

**User model**

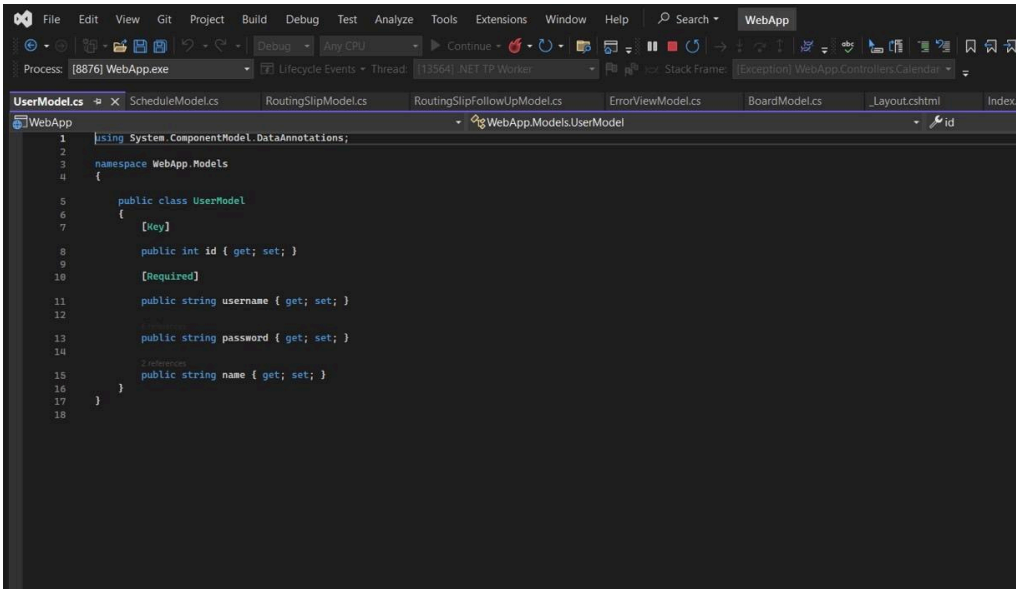



Figure 4.5.8: User model

The User model includes fields for ID, username, password, and name. It is designed to store and manage user credentials and personal information within the application, ensuring secure authentication and personalized user interactions.

**Working Calendar model**

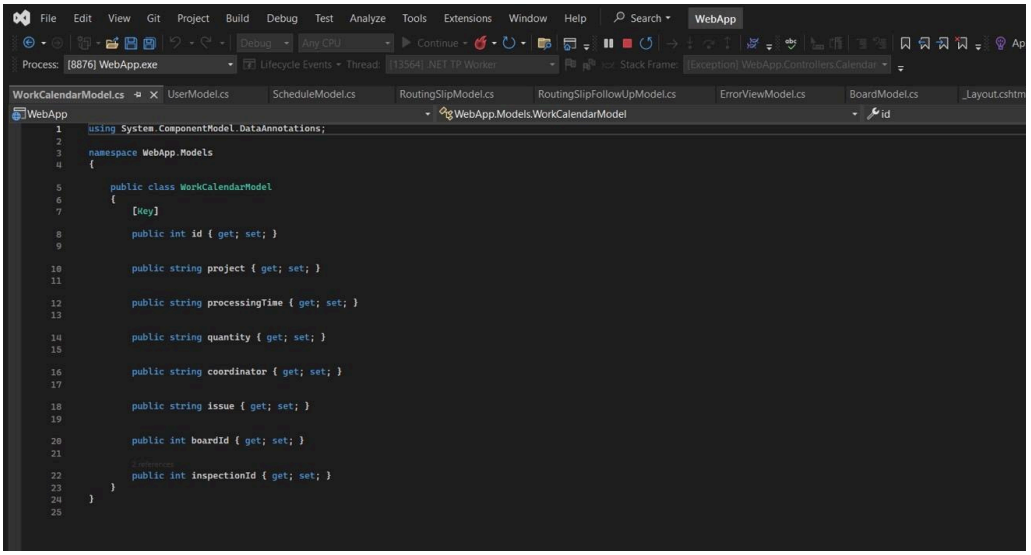



Figure 4.5.9: Working Calendar model

The Working Calendar model stores data related to the working calendar within the application activities to ensure projects are executed efficiently and on schedule.

**WorkingPaper model**

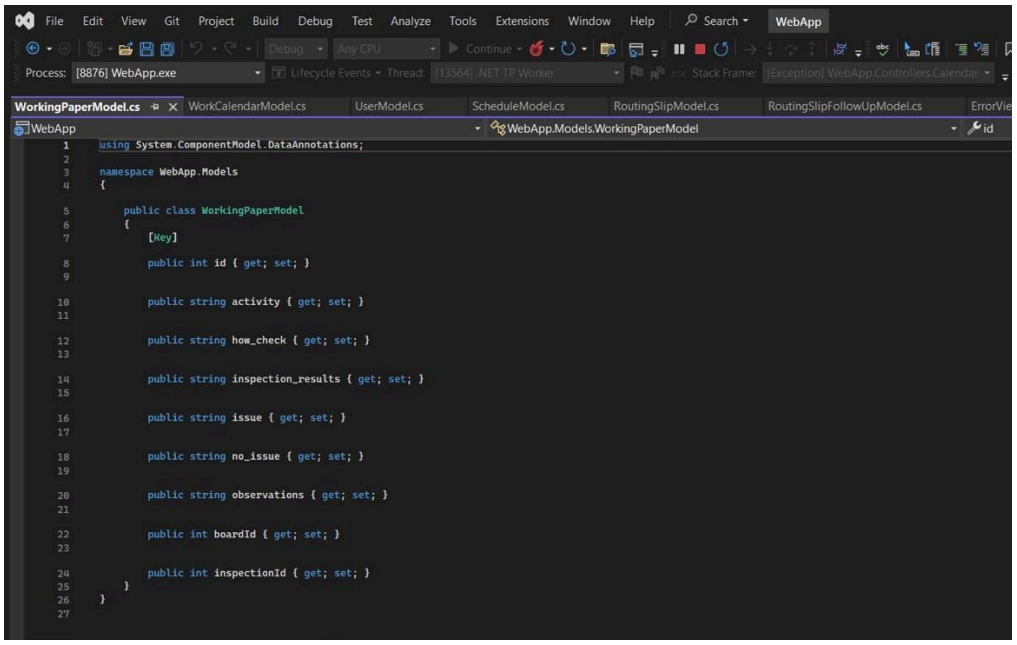



Figure 4.5.10: Workingpaper model

The Working Paper model stores data related to audit activities within the application. It manages information about audit procedures, documentation requirements, inspection results, issues identified, and any additional remarks. This model facilitates comprehensive tracking and management of audit-related data throughout the auditing process

### 4.6 Migrations

This section pertains to migrating data from different pages Board, Engagement Plan, Error View, Audit, Routing Slip Follow-up, Routing Slip, Schedule, User, Working Calendar, and Working Paper into corresponding database tables. Migration involves converting these entities into database schema using Entity Framework migrations. This ensures data consistency and facilitates efficient data storage and retrieval aligned with application needs.

### Add user migrations

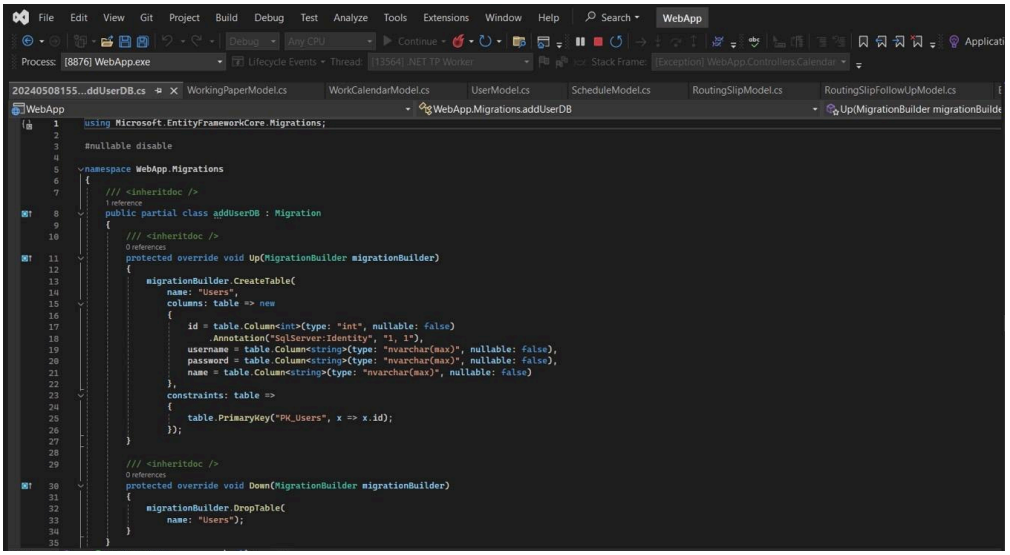



Figure 4.6.1: Add user migrations

This figure shows the DbContext class for an ASP.NET Core MVC application. The DbContext class provides access to the database and the code snippet shows that it includes a DbSet for a User model with an Id property as the primary key.

## Add board migration

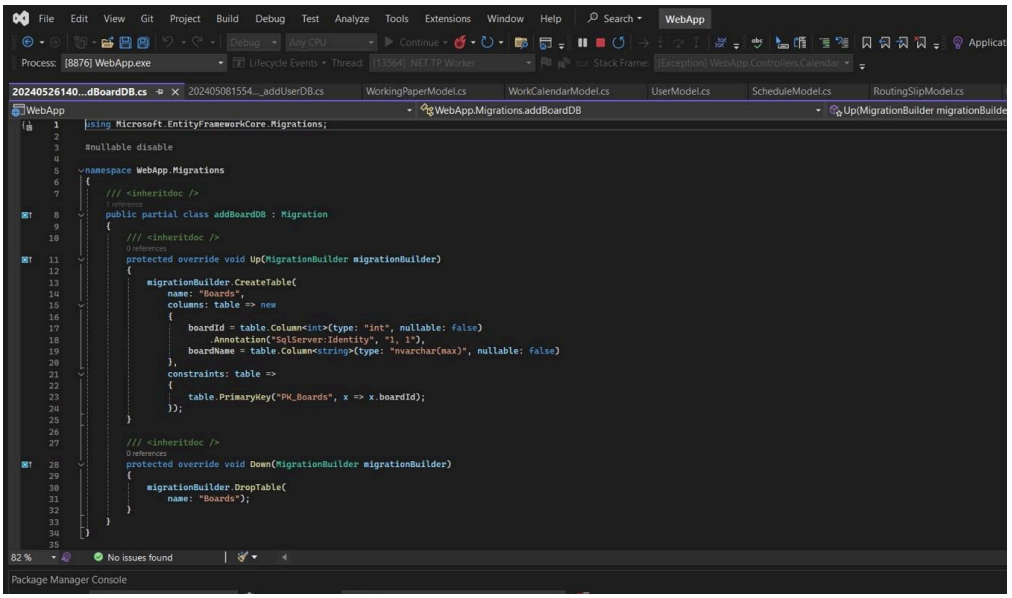



Figure 4.6.2: Add board migrations

This figure is part of a migration that defines a table to display a list of users and store a collection of users and displays their name and role in a table format.

## Add routing slip migrations

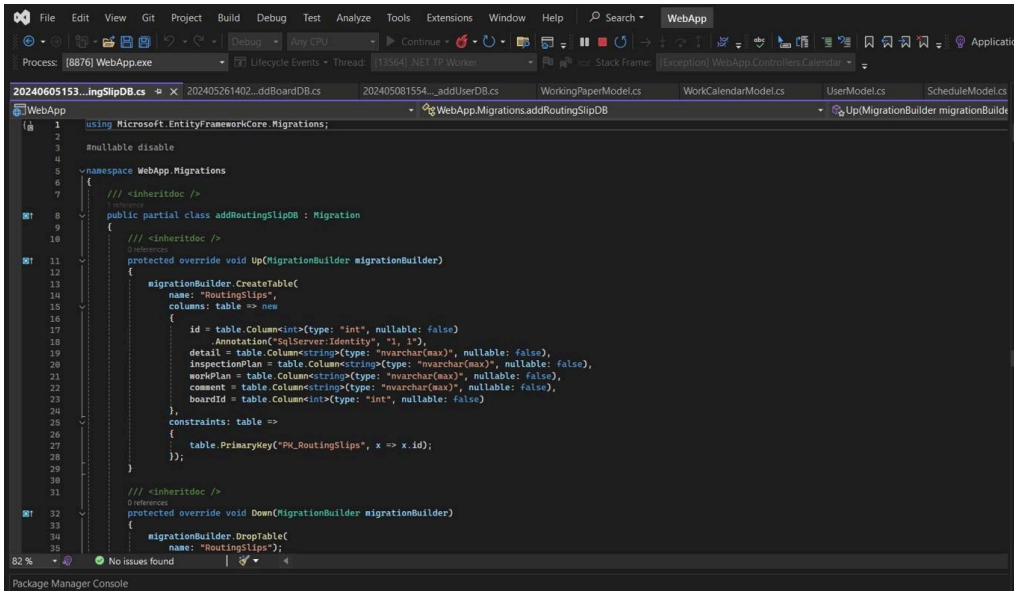



Figure 4.6.3: Add routing slip migrations

This figure is part of a migration in an ASP.NET Core MVC application that adds a new table named WorkCalendars to the database. The table is used to store work calendar information, which includes  a schedule ID, day of work, start time, end time, and a description.

**Add work calendar migrations**

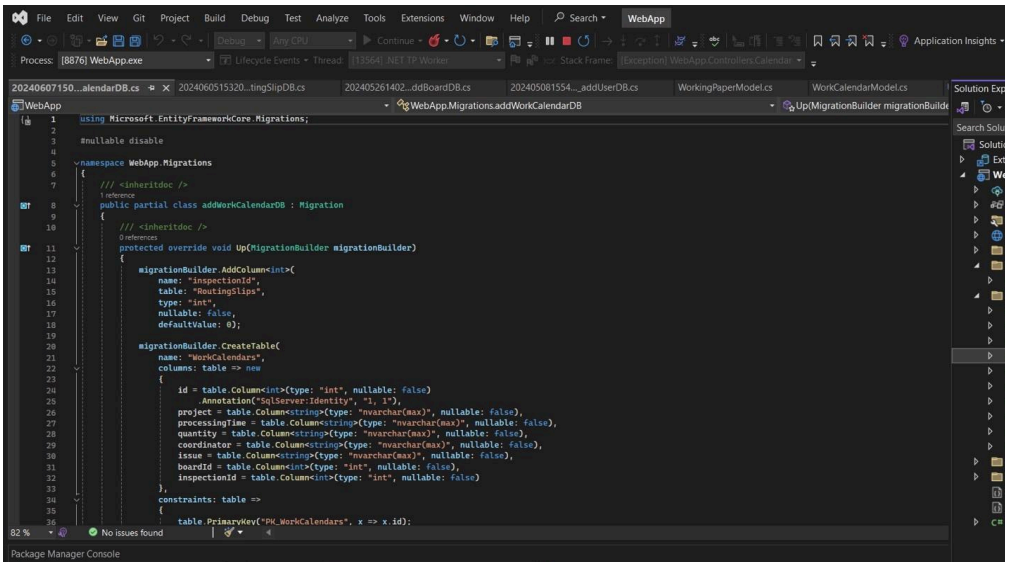



Figure 4.6.4: Add work calendar migrations

This figure is a part of a migration in an ASP.NET Core MVC application that adds a work calendar. The collection of strings is passed from the controller to the view using the ViewData property.

**Add routing slip follow-up migrations**

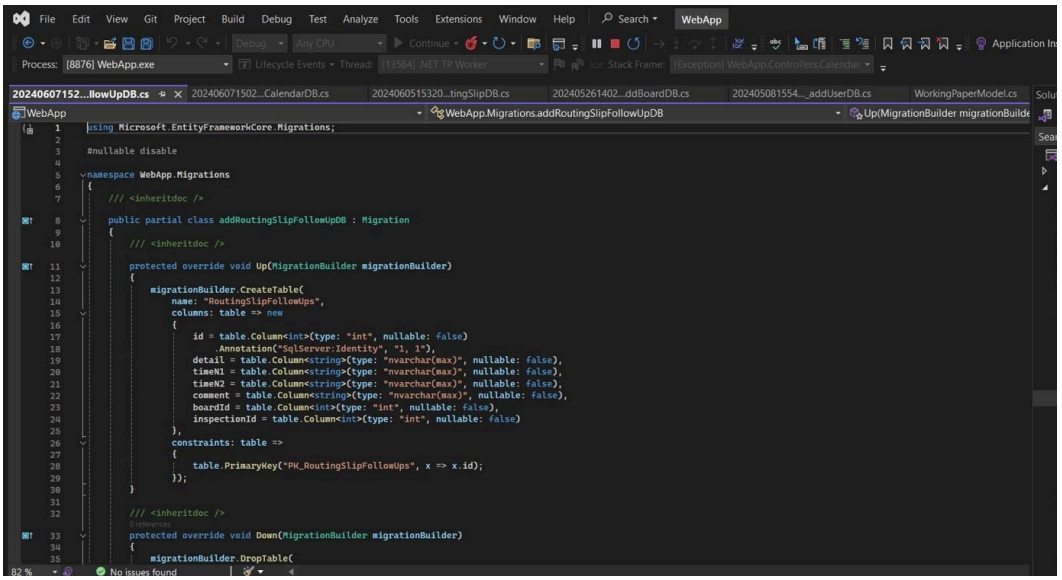
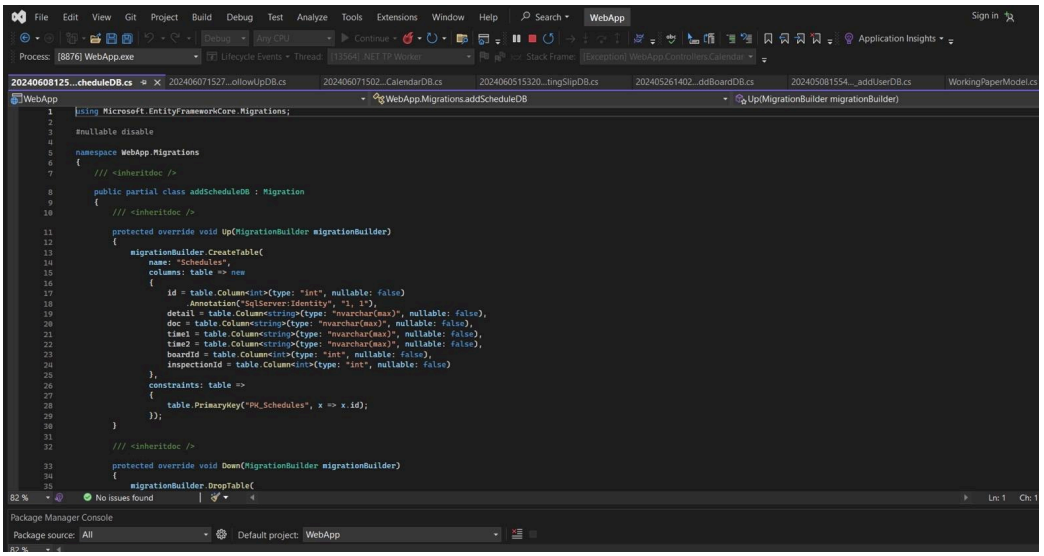



Figure 4.6.5: Add routing slip follow-up migrations

This figure is a part of a migration in an ASP.NET Core MVC application that adds a new table named RoutingslipFellowups to the database. The table is used to store routing slip follow ups, which include information such as the routing slip ID, follow up by, follow up at, description, and associated board and inspection IDs.

**Add schedule migrations**

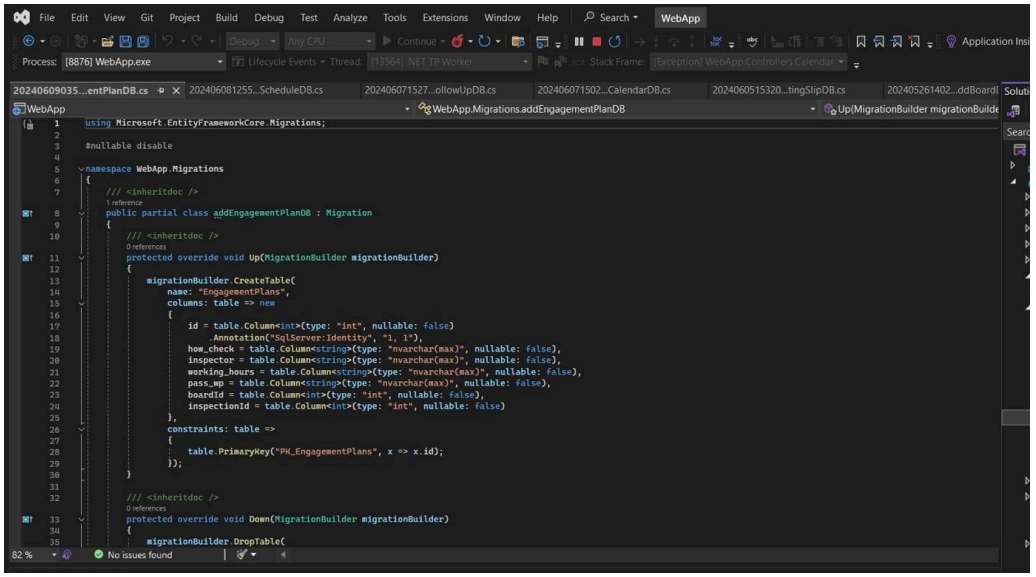


```
using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace WebApp.Migrations
{
    /// <inheritdoc />
    public partial class addScheduleDB : Migration
    {
        /// <inheritdoc />
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.CreateTable(
                name: "Schedules",
                columns: table => new
                {
                    id = table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1"),
                    detail = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    doc = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    time1 = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    time2 = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    boardId = table.Column<int>(type: "int", nullable: false),
                    inspectionId = table.Column<int>(type: "int", nullable: false)
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_Schedules", x => x.id);
                });
        }

        /// <inheritdoc />
        protected override void Down(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.DropTable(
```

Figure 4.6.6: Add schedule migrations

This figure defines a table to display a list of scheduled elements that define the header cells for the table. The header contains text labels for the schedule details (ID, Detail, Doc, Time, BoardId, InspectionId).

**Add engagement plan migrations**



```
using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace WebApp.Migrations
{
    /// <inheritdoc />
    public partial class addEngagementPlanDB : Migration
    {
        /// <inheritdoc />
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.CreateTable(
                name: "EngagementPlans",
                columns: table => new
                {
                    id = table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1"),
                    how_check = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    inspector = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    working_hours = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    pass_wp = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    boardId = table.Column<int>(type: "int", nullable: false),
                    inspectionId = table.Column<int>(type: "int", nullable: false)
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_EngagementPlans", x => x.id);
                });
        }

        /// <inheritdoc />
        protected override void Down(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.DropTable(
```

Figure 4.6.7: Add engagement plan migrations

This figure shows a migration in an migration in an ASP.NET Core MVC application that adds a new table named EngagementPlans to the database. The table is used to store engagement plan records, which include information such as the name, checklist, description, and due date.

**Add working paper migrations**

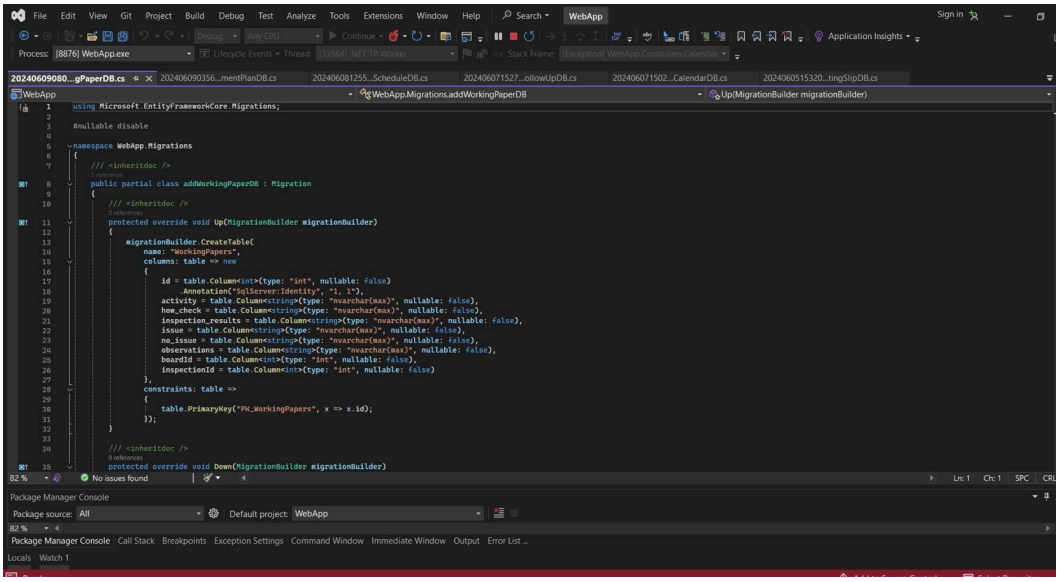



Figure 4.6.8: Add working paper migrations

This figure shows a migration in an ASP.NET Core MVC application that adds a new table named WorkingPapers to the database. The table is used to store working paper records, which include information such as the activity, inspection results, issues, observations, and associated board and inspection IDs.

# CHAPTER 5

## TESTING AND EVALUATION

### 5.1 Unit Tests

For the unit tests, we selected some important and critical processes for formal unit testing. The selected processes include:

- Audit work calendar
- Routing slip
- Schedule

### 5.2 Test Performed on Audit work calendar

Table 5.2: Audit work calendar

| Operation Performed | Condition Tested | Actual Result |
|---|---|---|
| Add a new Audit work to the calendar | To ensure that the new event is successfully added to the calendar | The event is successfully added to the calendar |
| Edit the data in the table | To edit the data in the Audit work calendar | The data is successfully edited and saved |
| Display the data in the table to the user | The data in the table should be displayed to the user | The data in the table is successfully displayed in the table |

### 5.3 Test Performed on Routing slip

Table 5.3: Routing slip

| Operation Performed | Condition Tested | Actual Result |
|---|---|---|
| Add a new routing slip data to the table | To ensure that the new data is successfully added to the routing slip | The data is successfully added to the routing slip |
| Edit the data in the table | To edit the data in the routing slip table | The data is successfully edited and saved |
| Display the data in the table to the user | The data in the table should be displayed to the user | The data in the table is successfully displayed in the table |

### 5.4 Test Performed on Schedule

Table 5.4: Schedule

| Operation Performed | Condition Tested | Actual Result |
|---|---|---|
| Add a new routing slip data to the table | To ensure that the new data is successfully added to the routing slip | The data is successfully added to the routing slip |
| Edit the data in the table | To edit the data in the routing slip table | The data is successfully edited and saved |

| | | |
|---|---|---|
| Display the data in the table to the user | The data in the table should be displayed to the user | The data in the table is successfully displayed in the table |

### 5.5 System Integration Test

This activity is performed after the system is completely integrated. The purpose of this testing is to check whether the system can operate correctly according to the required functions or not.

**Test Scenario**

In order to test all functional aspects of the system thoroughly, we had set up a test scenario which consisted of phases as shown below.

- System initialization
- User authentication
- Data input and editing
- System integration
- Performance and scalability

Moreover, the test scenario can be used as a user guideline because it covers all the steps necessary in order to use our system. The details of each phase are shown in the next section.

#### 5.5.1 System initialization

1. Start the system.
2. Verify that all essential modules load without errors.
3. Check system logs for any initialization issues.

#### 5.5.2 User authentication

1. Enter valid user credentials.

2. Verify successful login.
3. Test login failure with invalid.

### 5.5.3 Data input and editing

1. Input new data into the system.
2. Edit existing data record.
3. Verify that changes are saved correctly.

### 5.5.4 System integration

1. Integrate with external databases or APIs.
2. Test data exchange between systems.
3. Confirm compatibility and data integrity.

### 5.5.5 Performance and scalability

1. Perform load testing with simulated user traffic.
2. Measure response times and system stability.
3. Test scalability by increasing user load and data volume.

# CHAPTER 6

## CONCLUSION

Overall, this project demonstrates the potential of web applications to self-modify applications by combining advanced features and functionality, such as using code knowledge with auditing. The web applications have achieved important goals and greatly improved the efficiency and fluency of the audit process.

### 6.1 Benefits

#### 6.1.1 Benefits to Project Developers

- The development of this metaverse education platform has allowed project developers to showcase our innovation and technical expertise in merging content with VR technology.
- Developers have gained hands-on experience in addressing technical issues, refining user interfaces, and optimizing content delivery.
- Integrating the program workflow bridge between Unity and Blender has provided project developers with a streamlined workflow and enhanced collaboration capabilities.

#### 6.1.2 Benefits to Users

- Users benefit from an immersive and interactive learning experience that goes beyond traditional methods, fostering deeper engagement and understanding of historical concepts.
- Users gain exposure to culturally diverse historical settings and perspectives
- Users can retain historical knowledge more effectively and develop a deeper comprehension of historical narratives.
- The interactive challenges and quests within the platform encourage users to apply critical thinking and problem-solving skills in a historical context.

- The platform supports multi-user environments, enabling collaborative learning experiences where users can exchange ideas and insights with peers.

**6.2 Problems and Limitations**

- The project may face challenges related to VR hardware requirements. Based on limitations of Oculus Quest 2, some scripts in Spatial do not support VR headset now. For example, quest tasks do not show on the user's UI.
- Creating historically accurate and engaging content for the platform may pose challenges in terms of research, content curation, and balancing educational value with technological innovation.
- The challenges of using spatial scripts in VR tools are complex, including debugging and optimizing script performance.

**6.3 Future Work**

- Expanding content and features. There is potential to expand the platform's content with additional historical periods, events, and interactive features to offer users a broader range of learning experiences.
- Conducting further research and evaluation studies can provide insights into the platform's effectiveness in achieving educational objectives
- Future work could focus on refining the VR platform based on enhancing features, content, and usability to meet evolving educational needs.

# REFERENCES

[1]Chen, J., Smith, A., & Wang, L. Enhancing Internal Audit Efficiency through Web-Based Collaboration Tools. [Journal of Accounting Technology] 2018 [cited 2024 Jan 01]; 22(4): 567-584

[2] Author, B., *Title*. Edition ed. Series Title, ed. S. Editor. Vol. Volume. Year, City: Publisher. Number of Pagers.[Accessed: 1-Apr-2024]

[3] Johnson, M., & Brown, R. User Interface Design Best Practices for Internal Audit Web Applications. [International Journal of Human-Computer Interaction] 2019 [cited 2024 Jan 01]; 35(2): 189-204

[4] Smith, K., & Patel, S. Ensuring Security in Web-Based Internal Audit Systems. [Journal of Information Security] 2020 [cited 2024 Jan 01]; 12(3): 321-337

[5] Lee, S., & Kim, Y. Web-Based Data Analytics for Internal Audit: A Case Study. [Journal of Information Systems] 2017 [cited 2024 Jan 01]; 19(1): 45-62

[6] Mahidol. Title. [Online]. Available: https://op.mahidol.ac.th/ia/wp-content/uploads/2017/05/a1Laboratory-manuals210959.pdf. [cited 2024 Apr 17].

[7] Mahidol. มาตรฐาน ISO 19011: มาสนับสนุนการบริหารงานตรวจสอบอย่างไร_post.pdf. [Online]. Year Last Update Date. Available from: https://acpro-std.tfac.or.th/test_std/uploads/files/มาตรฐานสอบบัญชี/มาตรฐาน%20ISO%2019011%20มาสนับสนุนการบริหารงานตรวจสอบอย่างไร_post.pdf. [cited 2024 Apr 17].

# BIOGRAPHIES

**NAME** Mr. Sarun Jearawattanakul

**INSTITUTIONS ATTENDED** , :TriamudomsuksaPattanakarn School 2019
High School Diploma
Mahidol University, 2024:
Bachelor of Science (ICT)

**NAME** Mr. Krittapat Donsom

**INSTITUTIONS ATTENDED** Yothinburana School 2021
High School Diploma
Mahidol University, 2024:
Bachelor of Science (ICT)

**NAME** Mr. Waranat Deepradub

**INSTITUTIONS ATTENDED** Joseph Upatham School 2021
High School Diploma
Mahidol University, 2024:
Bachelor of Science (ICT)