



ក្រសួងអប់រំ យុវជន និងកីឡា



វិទ្យាស្ថានបច្ចេកវិទ្យាកម្ពុជា

ដេប៉ាតឺម៉ង់ ទេពកោសល្យព័ត៌មានវិទ្យា និងទំនាក់ទំនង

គម្រោងសញ្ញាប័ត្រវិស្វកម្ម

**ប្រធានបទ : កម្មវិធីបច្ចេកវិទ្យាគ្រប់គ្រងសន្លឹកសៀវភៅកម្រង
សម្រាប់**

និស្សិត : វ៉ែន ថុង

ឯកទេស : វិស្វកម្មព័ត៌មានវិទ្យា និងទំនាក់ទំនង

គ្រូទទួលបន្ទុក : ណុប ភារម្យ

ឆ្នាំសិក្សា : ២០២៣-២០២៤

**MINISTERE DE L'EDUCATION,
DE LA JEUNESSE ET DES SPORTS**

INSTITUT DE TECHNOLOGIE DU CAMBODGE

**DEPARTEMENT DE GENIE INFORMATIQUE ET
COMMUNICATION**

MEMOIRE DE FIND D'ETUDES

Titre : Système de gestion des stocks de livres

Etudiant : VEN Thon

Spécialité : Génie Informatique et Communication

Tuteur de stage : Mr. NOP Phearum

Année scolaire : 2023-2024



ក្រសួងអប់រំ យុវជន និងកីឡា វិទ្យាស្ថានបច្ចេកវិទ្យាកម្ពុជា



ដេប៉ាតឺម៉ង់ ទេពកោសល្យព័ត៌មានវិទ្យា និងទំនាក់ទំនង

គម្រោងសញ្ញាបត្រវិស្វកម្ម

បេសនិស្សិត: ច័ន ឌុន

កាលបរិច្ឆេទការពារនិក្ខេបបទ: ថ្ងៃទី ១០ ខែ កក្កដា ឆ្នាំ ២០២៤

អនុញ្ញាតឱ្យការពារគម្រោង

នាយកវិទ្យាស្ថាន: _____

ថ្ងៃទី _____ ខែ _____ ឆ្នាំ ២០២៤

ប្រធានបទ

: កម្មវិធីបច្ចេកវិទ្យាគ្រប់គ្រងសន្តិវិស័យកម្រង

សម្រាប់

សហគ្រាស

: ក្រសួងសេដ្ឋកិច្ច និង ហិរញ្ញវត្ថុ

ប្រធានដេប៉ាតឺម៉ង់

: លោក ឡែង ហេង _____

សាស្ត្រាចារ្យដឹកនាំគម្រោង

: លោក ណុប តារាម _____

អ្នកទទួលខុសត្រូវក្នុងសហគ្រាស: លោក ឌុន សីហា _____

រាជធានីភ្នំពេញ ឆ្នាំ២០២៤



**MINISTRE DE L'EDUCATION,
DE LA JEUNESSE ET DES SPORTS**



**INSTITUT DE TECHNOLOGIE DU CAMBODGE
DEPARTEMENT DE GENIE INFORMATIQUE ET
COMMUNICATION**

**MEMOIRE DE FIN D'ETUDES INGENIEUR
DE M. VEN Thon**

Date de soutenance: le 10 Juillet 2024

« Autorise la soutenance du mémoire »

Directeur de l'Institut: _____

Phnom Penh, le 2024

Titre : Système de gestion des stocks de livres
Etablissement du stage : Ministère de l'Economie et des Finances
Chef du département : Mr. LAY Heng _____
Tuteur de stage : Mr. NOP Phearum _____
Responsable de l'établissement : Mr. VON Seyha _____

PHNOM PENH, 2024

ACKNOWLEDGMENTS

As a fifth-year engineering student at the Institute of Technology of Cambodia, I must participate in a minimum three month for internship program. Additionally, it provides opportunity for me to apply the knowledge acquired during my academic studies to real work and get new experience, especially with technical skills, soft skills, or many more.

First of all, I would like to express my deepest gratitude to my parents who always give me all kinds of support, advice to me and thankfulness to the people who gave me support and involved in my academic journey.

H.E. Dr. PO Kimtho, Director of Institute of Technology of Cambodia for his excellent administration of the organization and good cooperation with the partner universities at the local, regional and international levels to improve the standard of the training of engineers and senior technicians.

Mr. LAY Heng, Head of the Department of Information and Communication Engineering at the Institute of Technology of Cambodia that provides students the opportunity to have the internship to improve the technical skill and his good management under his hierarchy at the department.

Mr. NOP Phearum, my academic supervisor, he spending time a lot of with me for the semester to provide the solution and idea for my internship as well. Therefore, it allows me to correct my mistakes through the process by meeting with him and complete this thesis success.

Mr. VON Seyha, he is my adviser for my internship, he always gave me the solution as the problem and idea of the process. His comments and correction welcome me as well, enabling me to finish my final dissertation.

All lectures in the Department of Information and Communication Engineering for the knowledge and all attribute key have shared me during my studies at Institute of Technology of Cambodia.

អត្ថបទសង្ខេប

ក្នុងអំឡុងពេលចុះកម្មសិក្សាឆ្នាំទី៥ នៃក្រសួងសេដ្ឋកិច្ចនិងហិរញ្ញវត្ថុអស់រយៈពេលបីខែកន្លះ ដោយចាប់ផ្តើមពីថ្ងៃទី ១៩ ខែ កុម្ភៈ ឆ្នាំ ២០២៤ រហូតដល់ថ្ងៃទី ៣១ ខែ ឧសភា ឆ្នាំ ២០២៤។ ចំពោះគម្រោងមួយនេះផងដែរត្រូវបានបង្កើតឡើង នៅក្នុងគោលបំណងគឺ រៀបចំកម្មវិធីបច្ចេកវិទ្យាគ្រប់គ្រងសន្និធិសៀវភៅកម្រងសម្រង់នេះឡើង ដើម្បីធានាបាននូវប្រសិទ្ធភាពក្នុងការគ្រប់គ្រងសន្និធិសៀវភៅកម្រងសម្រង់របស់លេខាធិការដ្ឋានក្រុមប្រឹក្សានីតិកម្ម។ ទន្ទឹមនេះ កម្មវិធីនេះ និងជួយពន្លឿននៃការរៀបចំរបាយការណ៍ប្រចាំខែ និងងាយស្រួលក្នុងការស្វែងរកទិន្នន័យកម្រងសម្រង់ផងដែរ។

ចំពោះគម្រោងមួយនេះត្រូវបានបង្កើតឡើងនូវប្រព័ន្ធគ្រប់គ្រងឧបករណ៍ស្នាដៃបែបឌីជីថលដ៏ល្អមួយ និងមានប្រសិទ្ធភាពខ្ពស់ដើម្បីអោយអ្នកប្រើប្រាស់អាចចូលទៅប្រើប្រាស់បាន ដែលប្រព័ន្ធមួយនេះផងដែរជួយពន្លឿននៃការរៀបចំចំនួនរបាយការណ៍ជាប្រចាំសប្តាហ៍ ប្រចាំខែ និងប្រចាំឆ្នាំ និងសម្រួលដល់ការពិនិត្យរបាយការណ៍វត្តមានបានគ្រប់ពេលពេល និងគ្រប់ទីកន្លែង។

Angular ដែលជា framework of JavaScript និង TypeScript ត្រូវបានយកមកប្រើប្រាស់ជាបច្ចេកវិទ្យាសម្រាប់បង្កើតជាគេហទំព័រមួយនេះ ចំណែក Nest JS ដែលជា framework របស់ Node JS និង MongoDB ត្រូវបានប្រើប្រាស់ក្នុងការគ្រប់គ្រងនិងរក្សាទុកទិន្នន័យមួយនេះ។ ក្នុងដំណើរការការបង្កើត គេហទំព័រ និង បច្ចេកវិទ្យាប្រព័ន្ធគ្រប់គ្រងសន្និធិសៀវភៅកម្រងសម្រង់ មានផលលំបាកច្រើនលើការសិក្សា framework និងការរៀបចំទិន្នន័យអោយបានត្រឹមត្រូវស្របតាមដំណើរការទាំងអស់ដែលបានកំណត់ក្នុងប្រព័ន្ធ។ ទន្ទឹមនេះដែរខ្ញុំបានទទួលនូវបទពិសោធន៍យ៉ាងច្រើនលើការប្រើប្រាស់ framework ការរៀបចំទិន្នន័យ និង របៀបស្វែងរកដំណោះស្រាយបញ្ហាបច្ចេកទេសផ្សេងៗ ពីគម្រោងមួយនេះ។

ក្នុងនិក្ខេបបទមួយនេះខ្ញុំផ្តោតទៅលើ ការរៀបរាប់និងរៀបចំអំពីដំណើរការ ក្នុងអំឡុងពេលធ្វើការអភិវឌ្ឍគម្រោងរបស់យើងខ្ញុំរួមបញ្ចូលទាំងការប្រមូលផ្តុំតម្រូវការ ការចនាប្រព័ន្ធការអភិវឌ្ឍន៍មូលដ្ឋានទិន្នន័យ ការសរសេរកូដការធ្វើតេស្តនិងការអនុវត្ត។ លើសពីនេះទៅទៀតបច្ចេកវិទ្យាប្រព័ន្ធគ្រប់គ្រងសន្និធិសៀវភៅកម្រងសម្រង់មួយនេះដើម្បីធានាបាននូវសុវត្ថិភាព ភាពងាយស្រួលក្នុងការកំណត់និងគ្រប់គ្រងគឺអ្នកប្រើប្រាស់ក្នុងប្រព័ន្ធ គឺតម្រូវមានការ Login ជាមួយ ឈ្មោះគណនី និងពាក្យសម្ងាត់ ដែលបានរៀបចំដោយ Admin។ គណនីអ្នកប្រើប្រាស់អាចមានច្រើនប្រភេទគឺ គណនីគ្រប់គ្រងប្រព័ន្ធ (Admin) គណនីអ្នករៀបចំលិខិត (Editor) គណនីពិនិត្យនិងសម្រួល (Basic) និងគណនីអ្នកអាន (Normal)។

RÉSUMÉ

Au cours de mon stage enrichissant de trois mois et demi au ministère de l'Économie et des Finances, du 19 février 2024 au 31 mai 2024, je me suis engagé dans un projet charnière centré autour de la création d'un système de sécurisation très efficace. Système de gestion des stocks de livres adapté à la base d'utilisateurs du ministère. Ce projet a également été mis en place dans le but de développer des programmes technologiques de gestion des stocks. Le livret assure la gestion efficace des inventaires, des livres et des devis du Secrétariat du Conseil législatif. Dans le même temps, ce programme accélérera la préparation des rapports mensuels et facilitera également la recherche des devis de données.

Pour l'objectif principal de ce projet, un système de gestion de flux de travail numérique robuste et très efficace a été développé, permettant aux utilisateurs d'y accéder. Ce système accélère la préparation des rapports hebdomadaires, mensuels et annuels et facilite les rapports de présence à tout moment et en tout lieu.

Dans le domaine du développement front-end, nous avons exploité Angular comme framework, associé à l'écosystème TypeScript. Pour le backend, nous avons exploité TypeScript, un framework Nest JS, en tandem avec MongoDB pour construire le système de gestion de l'inventaire des livres et stocker efficacement les données dans notre base de données. Naviguer dans les subtilités de ces technologies au cours du projet a posé de nombreux défis, exigeant une compréhension approfondie des cadres et une organisation habile des données en alignement avec les flux de travail complexes du système. Parallèlement, cette entreprise a enrichi mes compétences techniques, renforçant ma capacité à aborder diverses complexités survenues au cours du projet.

Dans le contexte de cette thèse, mon point central est la documentation méticuleuse et l'organisation systématique des processus de développement que nous avons traversés au cours de l'évolution du projet. De plus, la technologie utilisée dans ce livre Le système de gestion des stocks garantit la sécurité, la facilité de configuration des comptes utilisateur et la gestion au sein du système, nécessitant une authentification via un nom de compte et un mot de passe fournis par l'administrateur. Les comptes d'utilisateurs peuvent varier en type, notamment Administrateur, Éditeur, De base et Normal.

ABSTRACT

During my enriching three-and-a-half-month internship at the Legal Council of the Ministry of Economy and Finance, spanning from February 19, 2024, until May 31, 2024, I engaged in a pivotal project that centered around the creation of a highly efficient Securing Book Inventory Management System tailored to cater to the ministry's user base. This project was also established with the aim of developing inventory management technology programs. The booklet ensures the effective management of inventory, books, and quotes of the Secretariat of the Legislative Council. At the same time, this program will speed up the preparation of monthly reports and easy to find the data quotes as well.

For the primary objective of this project, a robust and highly efficient digital workflow management system has been developed, allowing users to access it. This system speeds up the preparation of weekly, monthly, and annual reports, and facilitates attendance reports anytime, anywhere.

In the realm of front-end development, we harnessed Angular as a framework, coupled with the TypeScript ecosystem. For the backend, we leveraged TypeScript, a Nest JS framework, in tandem with MongoDB to construct the Book Inventory management system and store data efficiently in our database. Navigating the intricacies of these technologies during the project posed numerous challenges, demanding an in-depth comprehension of the frameworks and the adept organization of data in alignment with the system's intricate workflows. Concurrently, this endeavor enriched my technical skill set, enhancing my ability to tackle various complexities that arose during the project.

Within the context of this thesis, my focal point is the meticulous documentation and systematic organization of the development processes that we traversed during the project's evolution. Furthermore, the technology utilized in this book Inventory management system ensures security, ease of user account setup, and management within the system, requiring authentication through an account name and password provided by the administrator. User accounts may vary in type, including Admin, Editor, Basic, and Normal.

TABLE OF CONTENTS

ACKNOWLEDGMENTS.....	I
អត្ថបទសង្ខេប	II
RÉSUMÉ.....	III
ABSTRACT	IV
TABLE OF CONTENTS.....	V
LIST OF FIGURES.....	VIII
LIST OF TABLES	IX
ABBREVIATION LIST	X
INTRODUCTION.....	1
I.GENERAL PRESENTATION.....	2
1.1. Presentation of the Organization	2
1.1.1. History of the Company	2
1.1.2. Services	2
1.1.3. Hierarchy	3
1.1.4. Contact Information	3
1.2. Internship Project Presentation	4
1.2.1. Structure	4
1.2.2. Problems Overview	4
1.2.3. Statement of Problem	5
1.2.4. Objective	5
1.3. Schedule	6
1.4. Detailed Project Description	8
1.4.1. Scope of Study.....	8
1.4.2. Limitation of Study	8

1.4.3. Hardware and Operation Systems Used.....	9
II.ANALYSIS AND SPECIFICATION OF REQUIREMENT	10
2.1. Case Study of Requirement	10
2.1.1. Functional Requirements.....	10
2.1.2. Non-functional Requirements	22
2.2. Constrained System	22
2.2.1. Constrained Techniques	22
2.2.2. Constrained Time	23
2.3. Use Case.....	24
2.4. Conception of Database.....	28
III.TECHNOLOGY AND DESIGN	29
3.1. Choice of Technology	29
3.1.1. Programming Language	29
3.1.2. Framework Technology	31
3.1.3. Tool.....	32
3.2. System Design	35
3.2.1. Physical Architecture	35
3.2.2. Logical Architecture.....	36
3.2.3. Sequential Diagram	37
IV.PROJECT IMPLEMENTATION	39
4.1. Project Setup	39
4.1.1. Environment setup.....	39
4.1.2. Frontend setup project	39
4.1.3. Backend setup project	41
4.2. Implementation	43
4.2.1. Recipients Module.....	43
4.2.2. Search Module.....	49
4.2.3. Manage Offer Module	50
4.2.4. Telegram Integration Module.....	50

V.SUMMARY AND CONCLUSION	51
5.1. Summary	51
5.1.1. Complete Work	51
5.1.2. Incomplete Work.....	52
5.2. Strong Point.....	53
5.3. Weak Point	53
5.4. Experiences.....	53
5.5. Difficulties.....	54
5.6. Perspective.....	54
5.7. Conclusion	55
References.....	55
APPENDIX.....	55

LIST OF FIGURES

Figure 1: LCS Logo	2
Figure 2: The Legal Council Organization.....	3
Figure 3: MEF Location.....	4
Figure 4: Use Case Admin Diagram	24
Figure 5: Use Case Editor Diagram	25
Figure 6: Basic Use Case Diagram	26
Figure 7: Use Case Normal Diagram	27
Figure 8: Database Design	28
Figure 9: HTML Logo	29
Figure 10: CSS Logo.....	29
Figure 11: JavaScript Logo	30
Figure 12: SCSS Logo	30
Figure 13: Angular Logo.....	31
Figure 14: Nest JS Logo.....	31
Figure 15: TypeScript logo	32
Figure 16: VS Code Logo	32
Figure 17: GitLab Logo.....	33
Figure 18: Postman Logo	33
Figure 19: Figma Logo.....	34
Figure 20: Swagger Logo.....	34
Figure 21: MongoDB Logo.....	35
Figure 22: Physical Architecture.....	35
Figure 23: Logical Architecture	36
Figure 24: Sequential Diagram User Login	37
Figure 25: Sequential Diagram Crud on User.....	38
Figure 26: Frontend Structure	40
Figure 27: Backend Structure.....	42
Figure 28: Activity Crud on Recipient Type.....	44
Figure 29: Activity Crud on Ministry	45
Figure 30: Activity Diagram of Search	49

LIST OF TABLES

Table 1: Planning Project	6
Table 2: Admin Role	10
Table 3: Editor Role	15
Table 4: Editor Role	17
Table 5: Normal Role.....	20
Table 6: Table of Tasks Completed	51
Table 7: Table of Tasks Incomplete	52

ABBREVIATION LIST

API	: Application Programming Interface
CSS	: Cascading Style Sheets
DICE	: Department of Information and Communication Engineering
HIML	: Hypertext Markup Language
SCSS	: Sassy Cascading Style Sheets
VS Code	: Visual Studio Code

INTRODUCTION

Upon reaching their fourth year at the Department of Information and Communication Engineering (DICE) within the Institute of Technology of Cambodia (ITC), students are mandated to engage in an internship with a company or educational institution. This internship serves as the foundation for their subsequent thesis defense. Consequently, students must compile a comprehensive thesis report detailing the projects they undertook during this internship. It is imperative that the chosen project topic not only aligns with the internship experience but also possesses the requisite substance to be effectively defended during their thesis presentation.

An internship holds great importance for students as it offers a multitude of benefits. It provides a practical platform for students to apply their academic knowledge and gain real world experience. Internships enable students to develop essential professional skills, such as communication, teamwork, and problem-solving. They also expose students to the industry they are interested in, allowing them to understand its dynamics, trends, and best practices. Internships provide valuable networking opportunities, enabling students to establish connections with professionals in their field. Additionally, internships enhance a student's resume, making them more marketable to future employers. Overall, internships play a pivotal role in shaping a student's career trajectory by providing valuable learning experiences, industry insights, and a competitive edge in the job market.

I. GENERAL PRESENTATION

1.1. Presentation of the Organization

1.1.1. History of the Company



Figure 1: LCS Logo

The Legal Council of the Ministry of Economy and Finance was established by Prakas No. 290 MEF.PrK dated 15 March 2013 on the Establishment of the Legal Council of the Ministry of Economy and Finance. It was subsequently revised and finally replaced by Sub Decree No. 75 dated 25 May 2017 on the revision of Sub-Decree No. 488 dated October 16, 2013, regarding the organization and functioning of the Ministry of Economy and Finance, and Sub-Decree No. 76 dated June 13, 2018, concerning the organization and functioning of the Legal Council of the Ministry of Economy and Finance To ensure the legitimacy, consistency, and harmony in drafting legal documents, as well as to enhance the efficiency and effectiveness of management, the Ministry of Economy and Finance implements legal documents within its jurisdiction and contributes to the implementation of management reform programs in public finance.

1.1.2. Services

- ❖ **Vision:** To make the national budget management more effective and efficient.
- ❖ **Mission:** To ensure the sustainability, transparency, and effectiveness of public financial management and public service delivery.

1.1.3. Hierarchy

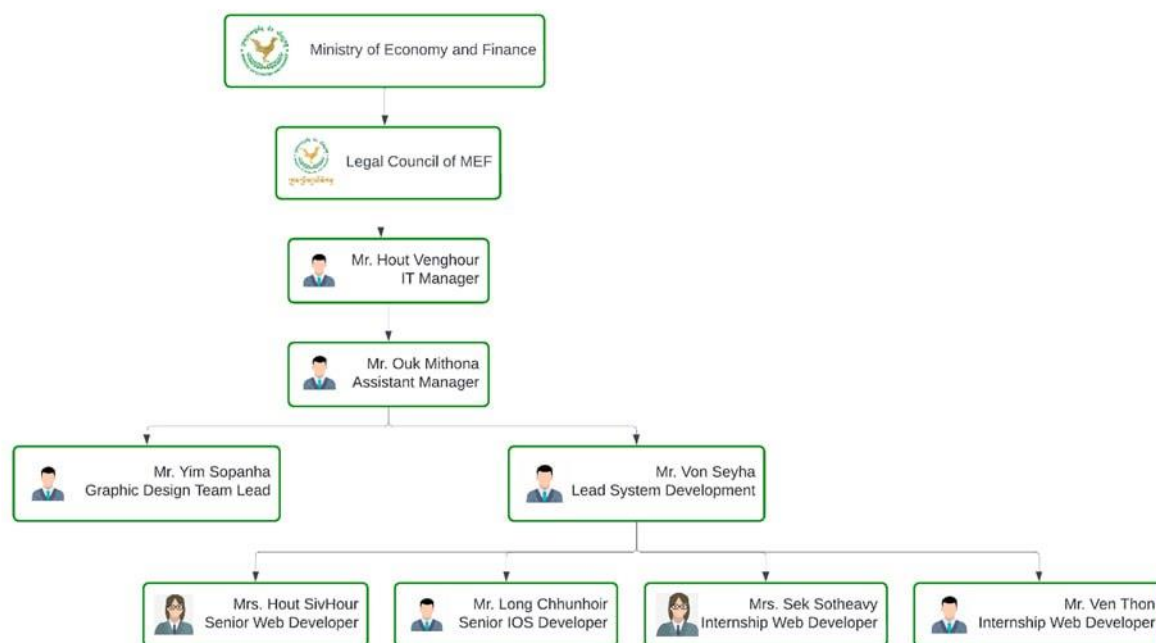


Figure 2: The Legal Council Organization

1.1.4. Contact Information

This is information and map of Ministry of Economy and Finance (MEF).

Address: St 92, Sangkat Wat Phnom Khan Doun Penh, Phnom Penh

Telephone: +855 97 25 28 358

Telegram: +855 97 25 28 358

E-mail: legalcouncilmef@gmail.com

Website: <https://www.legalcouncilmef.com/km>

Facebook Page: <https://www.facebook.com/LegalCouncilMEF/>

Map: Show on Figure 3

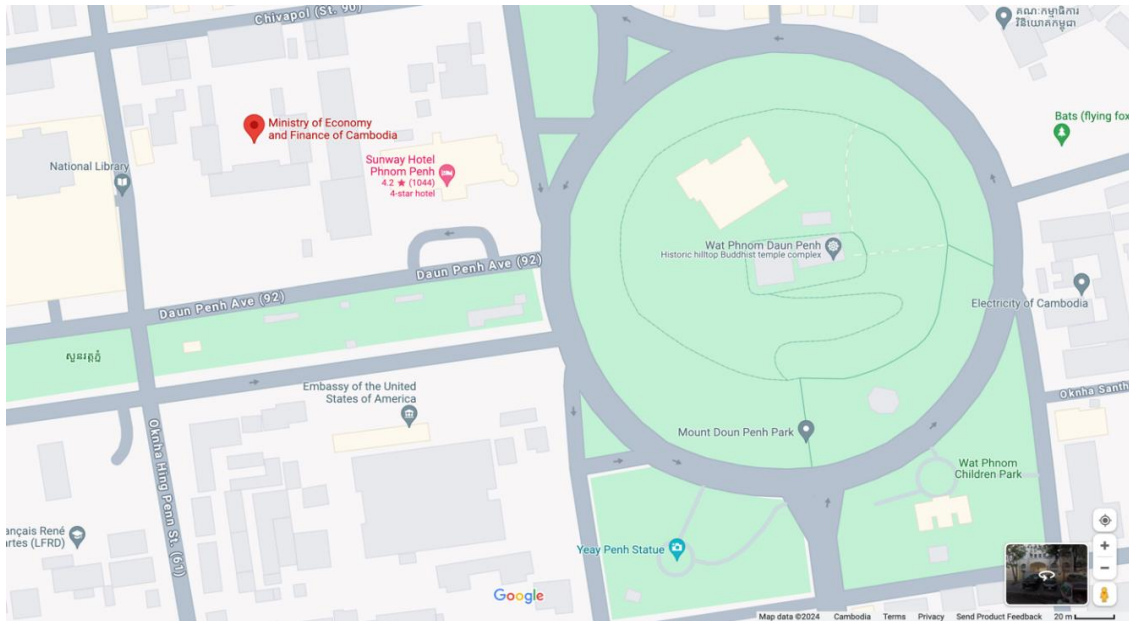


Figure 3: MEF Location

1.2. Internship Project Presentation

For the internship during three months at the Legal Council of Ministry of Economy and Finance, I work on project called "Book Inventory Management System", that is the system that ensure efficiency in managing the inventory of quotations within the Secretariat of the Legal Council and Additionally, this software will facilitate the monthly preparation of reports and make it easier to search for specific data within the document collection.

1.2.1. Structure

The internship and project development are guided and assisted by:

- Academic Supervisor: **Mr. NOP Phearum**
- Project Advisor: **Mr. VON Seyha**
- Developer **Mr. VEN Thon, Ms. SEK Sotheavy**

1.2.2. Problems Overview

The Secretariat of the Legal Council currently faces challenges in efficiently managing the inventory of quotations. The existing system lacks effectiveness and requires a more streamlined approach to enhance inventory management. Additionally, the process of preparing reports and searching for specific data within the document collection is time consuming and cumbersome. To address these issues, a Book Inventory Management system is proposed. The system aims to improve efficiency in managing quotations, automate manual operations using the DigiTech approach, and provide regular monthly and annual reports in a confidential manner. The key objectives of this project include:

- Develop a comprehensive management system based on the DigiTech approach to facilitate efficient manual operations.
- Generate monthly and annual reports that contain relevant information while ensuring confidentiality.
- Regularly provide reports on a monthly and yearly basis, enabling effective monitoring and evaluation.
- Designate specific locations for accessing and distributing the reports, ensuring convenient availability.

1.2.3. Statement of Problem

The Secretariat of the Legal Council is currently facing challenges in effectively managing the inventory of quotations. The existing system lacks efficiency and requires a more streamlined approach to improve inventory management processes. Additionally, the manual operations involved in preparing reports and searching for specific data within the document collection are time-consuming and labor-intensive. These issues hinder the Secretariat's ability to efficiently track and manage quotations, resulting in potential delays, errors, and difficulties in accessing relevant information.

Furthermore, the lack of a dedicated software solution specifically designed for book inventory management exacerbates these challenges.

1.2.4. Objective

The purpose of Steel is to establish this system of book inventory management system to ensure efficiency in managing the inventory of quotations within the Secretariat of the Legal Council. Additionally, this software will facilitate the monthly preparation of reports and make it easier to search for specific data within the document collection.

The main objectives of this project are as follows:

- We created a comprehensive management system for manual operations based on the DigiTech approach.
- Provide reports on a monthly and annual basis in a confidential manner.
- Reports are provided regularly, both on a monthly and a yearly basis and at designated

1.3. Schedule

As shown in the table above that showed the planning of my project during the internship period for the implementation of the Book Inventory Management System, following the Agile methodology with the time table planning.

Table 1: Planning Project

Tasks	Weeks											
	1	2	3	4	5	6	7	8	9	10	11	12
Learn Technology												
Define the project scop												
Mockup UI												
Analyze Requirement												
Development												
Testing												
Report												

For the development, testing and document as well I had to spend 12 weeks to develop the project that we spend time from fist week to twelve weeks. The work for each on the table planning is described below:

- **Learn new technology:** During my internship, I got my project and I had to do research and learn about Angular Framework and Nest JS with typescript like how to set up project, usage of this framework such as how to route each page. I did the research in document, watched many tutorials and offered some help from my team works. Therefore, I could catch up new technology fast and be able to learn and fix all my problems during the development process.

- **Mockup UI:** first I have to discussion with my advisor about UI of admin what we do, what we needed function and variable the on my project. After that I started to draw UI about of admin.
- **Analyze requirement:** In the third week, I started to analysis about the project about the design and the main functionalities that are required by project manager. We have discussed about the requirement inside the meeting about what were the main requirements and optional requirements inside the project before starting it.
- **Development:**
 - From seven to eleventh week, I started implementing code using Angular with typescript to build user interfaces. Before implementing, we have analyzed and divide the task plan in our project.
 - From third to eleventh week, I started implementing code using Nest JS with typescript to build Api and I tested flow of API by using Postman and Swagger to make sure it works well before connecting it with front- end.
 - After building UI of each page, I checked its responsiveness to make sure it looks well in each device.
 - In case, there are new requirement or feedback from my project manager, I had to fix it, double check on my UI design and the flow again.
- **Testing:** I did test part when I almost completed my user interfaces like responsiveness, accuracy routing to right destination page and checked whether its performance went smoothy when switching the pages while routing pages. If something went, I needed to implement again to make the website run well. Addition to that, I need to make sure it is connected well and go along well with flow of back- end.
- **Report:** I started to write the report to draft each step that I have done during development process and to make sure I would have enough time to correct my thesis report as well.

1.4. Detailed Project Description

1.4.1. Scope of Study

When I participated in a web development internship, the scope of my study included various aspects of web development such as front-end design, back-end programming, database management, and web application security. I also gained experience in using different frameworks and tools commonly used in the industry.

- **Frontend:** I focused on frontend development using HTML, CSS, SCSS, and JavaScript. I also utilized the Angular framework with TypeScript to build the frontend of web applications.
- **Backend:** I used Nest JS with TypeScript to build robust and scalable APIs. Nest JS is a progressive Node.js framework that leverages TypeScript to create highly testable, maintainable, and scalable applications. It follows the modular architecture pattern, making it easy to manage and organize the codebase. Additionally, I used MongoDB for data storage, utilizing the Mongoose library for object data modeling (ODM) to interact with the database. This setup allowed me to define schemas, perform CRUD operations, and handle data validation efficiently.
- **Full stack:** We combined the front-end and back-end. To connect the API from the back-end to the front-end Angular application
- **UX and UI design:** Web design, including layout, typography, color theory, and user experience (UX) design. I also learned about design tools like Figma.
- **Version Control System:** Using version control systems like Git. I learned how to collaborate with other developers using platforms like GitLab, and gained an understanding of branching, merging, and resolving conflicts.

1.4.2. Limitation of Study

- **Complexity of Technology Stack:** One of the primary limitations stemmed from the complexity of integrating multiple technologies within the project. While Angular provided robust frontend capabilities, integrating it seamlessly with Nest JS for backend logic and MongoDB for data storage proved to be challenging. This complexity led to longer development cycles and increased debugging efforts.
- **Learning Curve:** Another significant limitation was the steep learning curve associated with the chosen technologies, especially for team members who were new to Angular, Nest JS, or MongoDB. Acquiring proficiency in these technologies required extensive

training and experimentation, which impacted the project's timeline and resource allocation.

- **Scalability Concerns:** As the project progressed, scalability concerns emerged, particularly concerning MongoDB's scalability limitations. While MongoDB offers flexibility and ease of use for certain applications, its scalability may become a bottleneck for large-scale inventory management systems with high volumes of data and concurrent users.
- **Performance Optimization:** Achieving optimal performance proved to be a challenge, especially when handling complex queries or large datasets in MongoDB. Despite Nest JS's efficiency in handling asynchronous operations, optimizing database queries and minimizing response times required careful consideration and fine-tuning.
- **Deployment Complexity:** Deploying the application in a production environment posed challenges due to the intricacies of deploying Angular frontend, Nest JS backend, and MongoDB database components. Coordinating deployment strategies, managing dependencies, and ensuring compatibility across different environments required meticulous planning and coordination.

1.4.3. Hardware and Operation Systems Used

This project is conducted by using some devices with operating systems such as:

- **Hardware:** We used Laptop
- **Operating System:** We used Ubuntu and Digital Ocean was chosen as the primary operation systems for the server due to its stability and compatibility with software stack.

II. ANALYSIS AND SPECIFICATION OF REQUIREMENT

2.1. Case Study of Requirement

A case study of requirements for a web development internship involves understanding and documenting the specific goals and expectations of the intern and the organization providing the internship opportunity. It helps set clear expectations and defines the scope of work for the intern.

- **Frontend:** We need to learn foundational web technologies like HTML, CSS, and SCSS, along with JavaScript. Additionally, we should master the Angular framework with TypeScript to build dynamic, scalable, and maintainable frontend applications.
- **Backend:** We used Nest JS with TypeScript and MongoDB to build a robust backend API. Nest JS, a progressive Node.js framework, enabled us to structure our application with a modular architecture and leverage TypeScript for strong typing. By integrating MongoDB with Mongoose, we efficiently managed our data models and created scalable endpoints for user management.

2.1.1. Functional Requirements

The main functional requirements are those requirements that need to be designed and developed. It is very important to define because they must match the objective of the project.

2.1.1.1. Admin Role

An admin role has comprehensive access and permissions within the system, enabling them to perform all tasks such as creating, editing, and deleting data across the entire platform, ensuring full control over system management and maintenance.

Table 2: Admin Role

Admin			
Module	Feature	Description	Responsibility
User	Authentication and Authorization	Ensuring secure access to the system and controlling user permissions based on their roles.	Sotheavy
	Manage User	The admin can create, read, updating and delete user on the system.	

User	Manage Profile	The admin can update profile information, such as contact detail, preference and setting.	Sotheavy
	Manage User Role	The admin can assign roles to users, update role permission and retrieving role details	
Recipient	Manage Recipient	The admin can create, read, update and delete recipient on the system.	VEN Thon
	Manage Recipient Type	The admin can create, read, update and delete recipient type on the system.	
	Manage Ministry	The admin can create, read, update and delete ministry on the system.	
	Manage Staff	The admin can create, read, update and delete staff members on the system.	
	Manage Public Institute	The admin can create, read, update and delete pubic institute on the system.	
	Manage Sector	The admin can create, read, update and delete sector information on the system.	
	Manage Department	The admin can create, read, update and delete department detail on the system.	
	Manage Office	The admin can manage office withing organization including, create, read, update and delete on the system.	

Recipient	Manage Position	The admin can manage position withing organization including, create, read, update and delete position on the system.	VEN Thon
	Manage Book Store	The admin can create, read, update and delete book store on the system.	
Stock	Manage Book	The admin can create, read, update and delete book record on the system.	Sotheavy
	Manage Book Type	The admin can create, read, update and delete book type on the system.	
	Manage Author	The admin can create, read, update and delete author information on the system.	
	Manage Accessories	The admin can create, read, update and delete accessories record on the system.	
	Manage Publication	The admin can create, read, update and delete publication detail on the system.	
	Manage Publication Record	The admin can create, read, update and delete publication record on the system.	
	Manage Unit	The admin can create, read, update and delete unit on the system.	
	Manage Check Status	The admin can create, read, update and delete check status on the system.	

Stock	Manage Temporarily Delete Item	The admin can delete temporarily deleted item record on the system.	Sotheavy
	Manage Delete Permanently Item	The admin can delete permanently deleted item record on the system.	
Dashboard	Calculate Total	Calculating the total number of various entities, such as books, accessories, recipients, books-in-stock, accessories-in-stock, public institute, staffs, book-on-bookstore and accessories-on-bookstore.	Sotheavy
	Display latest Recipient	To displaying the latest recipients, such as the latest 5 recipients.	
	Latest Order Item	For displaying the latest order items, such as the latest 10 order items.	
Report	Download Report	For downloading reports of data by date, week, month, quarter, semester, year, and date interval. Includes reports such as, Book on Bookstore Report, Book provided to Public Institute Report, Book provided to staff Report, Joint book report form.	Sotheavy
	Export Report	For exporting reports in Excel, CSV, and PDF formats.	
Telegram Integration	Telegram Bot Alert	For integrating with Telegram bot to alert the Inventory Manager about important events or updates.	VEN Thon

Search	Search	Search any keyword of each list manage on the system.	VEN Thon
	Advance search on book and accessories	Advance search on book and accessories module. This search provides endpoints for conducting complex searches based on keywords, date intervals, years, book-type, publication, check status and sort.	
	Advance search on book order	Advance search on book order. This search provides endpoints for conducting complex searches based on keywords, date intervals, years, book-type and recipient.	
Manage Offer	Manage Book Order	This functionality enables users to manage book orders within the system. Users can view, create, update, and delete book orders as needed.	VEN Thon
	Manage Book Order Detail	This functionality allows users manage the details of book orders, as the individual books included in order and their respective quantities	

2.1.1.2. Editor Role

An Editor has full access to all parts of the system, allowing them to edit, update, and delete content. However, they do not have permission to create new user accounts, maintaining a clear distinction from admin capabilities.

Table 3: Editor Role

Editor			
Module	Feature	Description	Responsibility
User	Authentication and Authorization	Ensuring secure access to the system and controlling user permissions based on their roles.	Sotheavy
Recipient	Manage Recipient	The editor can create, read, update and delete recipient on the system.	VEN Thon
	Manage Recipient	The editor can create, read, update and delete recipient on the system.	
	Manage Ministry	The editor can create, read, update and delete ministry on the system.	
	Manage Staff	The editor can create, read, update and delete staff members on the system.	
	Manage Public Institute	The editor can create, read, update and delete pubic institute on the system.	
	Manage Sector	The editor can create, read, update and delete sector information on the system.	
	Manage Department	The editor can create, read, update and delete department detail on the system.	

Recipient	Manage Office	The editor can manage office withing organization including, create, read, update and delete on the system.	VEN Thon
	Manage Position	The editor can manage position withing organization including, create, read, update and delete position on the system.	
	Manage Book Store	The editor can create, read, update and delete book store on the system.	
Stock	Manage Book	The editor can create, read, update and delete book record information on the system.	Sotheavy
	Manage Book Type	The editor can create, read, update and delete book type information on the system.	
	Manage Author	The editor can create, read, update and delete author information on the system.	
	Manage Accessories	The editor can create, read, update and delete accessories information on the system	
	Manage Publication	The editor can create, read, update and delete publication information on the system	
	Manage Publication Record	The editor can create, read, update and delete publication record on the system	

Stock	Manage Unit	The editor can create, read, update and delete unit on the system	Sotheavy
	Manage Check Status	The editor can create, read, update and delete check status on the system	
	Manage Temporarily Delete Item	The editor can delete temporarily deleted item record on the system	
	Manage Delete Permanently Item	The editor can delete permanently deleted item record on the system.	
Manage Offer	Manage Book Order	The editor can create, read, update and delete book order information on the system	VEN Thon
	Manage Book Order Detail	The editor can create, read, update and delete book order item on the system	

2.1.1.3. Basic Role

After successfully logging in, the basic user has the ability to create and view content exclusively within the management system; however, they are not granted the privilege to generate new user accounts or assign user roles.

Table 4: Basic Role

Basic			
Module	Feature	Description	Responsibility
User	Authentication and Authorization	Ensuring secure access to the system and controlling user permissions based on their roles	Sotheavy

Recipient	Manage Recipient	The basic can create and view recipient information on the system	VEN Thon
	Manage Recipient Type	The basic can create and view recipient type information on the system	
	Manage Ministry	The basic can create and view ministry information on the system	
	Manage Staff	The basic can create and view staff members on the system	
	Manage Public Institute	The basic can create and view public institute information on the system	
	Manage Sector	The basic can create and view sector information on the system	
	Manage Department	The basic can create and view department information on the system	
	Manage Office	The basic can create and view u office on the system	
	Manage Position	The basic can create and view position on the system	
	Manage Book Store	The basic can create and view book store on the system	

Stock	Manage Book	The basic can create and view book record on the system	Sotheavy
	Manage Book Type	The basic can create and view book type on the system	
	Manage Author	The basic can create and view author information on the system	
	Manage Accessories	The basic can create and view accessories record on the system	
	Manage Publication	The basic can create and view publication detail on the system	
	Manage Publication Record	The basic can create and view publication record on the system	
	Manage Unit	The basic can create and view unit on the system	
	Manage Check Status	The basic can create and view check status on the system	
Manage Offer	Manage Book Order	The basic can create and view book order on the system	VEN Thon
	Manage Book Order Detail	The basic can create and view book order item on the system	

2.1.1.4. Normal Role

After a successful login, the normal user is restricted to viewing content solely within the management system and does not possess the authority to create new user accounts or assign user roles.

Table 5: Normal Role

Normal			
Module	Feature	Description	Responsibility
User	Authentication and Authorization	Ensuring secure access to the system and controlling user permissions based on their roles	Sotheavy
Recipient	Manage Recipient	The normal can view recipient information on the system	VEN Thon
	Manage Recipient Type	The normal can view recipient type information on the system	
	Manage Ministry	The normal can view ministry information on the system	
	Manage Staff	The normal can view staff members on the system	
	Manage Public Institute	The normal can view public institute information on the system	
	Manage Sector	The normal can view sector information on the system	

Recipient	Manage Department	The normal can view department information on the system	VEN Thon
	Manage Office	The normal can view u office on the system	
	Manage Position	The normal can view position on the system	
	Manage Book Store	The normal can view book store on the system	
Stock	Manage Book	The normal can view book record on the system	Sotheavy
	Manage Book Type	The normal can view book type information on the system	
	Manage Author	The normal can view author information on the system	
	Manage Accessories	The normal can view accessories record on the system	
	Manage Publication	The normal can view publication information on the system	
	Manage Publication Record	The normal can view publication record on the system	
	Manage Unit	The normal can view unit information on the system	
	Manage Check Status	The normal can view check status on the system	

Manage Offer	Manage Book Order	The normal can view book order on the system	VEN Thon
	Manage Book Order Detail	The normal can view book order item on the system	

2.1.2. Non-functional Requirements

Non-functional requirements refer to aspects of a system that focus on its qualities and characteristics rather than specific functions. Here are the corrected descriptions of the important non-functional requirements you mentioned:

- **Clean code:** It is important to ensure that our code is clean and well-organized, facilitating future development by other individuals.
- **Performance:** Ensuring that the website operates smoothly and efficiently, without any errors, is of utmost importance.
- **Stability:** The website should run reliably and consistently, without encountering errors, slowdowns, or data loss.
- **Security:** Strengthening website security is crucial to prevent unauthorized access and data breaches.
- **Maintenance:** Detecting and addressing errors or mistakes promptly enhances the overall quality and reliability of the application.

2.2. Constrained System

2.2.1. Constrained Techniques

Constrained Techniques in the context of a web development internship typically refers to working within certain limitations or constraints when developing web applications or websites. These constraints can arise from various factors, and as an intern, it's essential to understand and adapt to them. Here are some common constraints I might encounter during a web development internship:

- **Technology Stack:** The ministry may have a specific set of technologies or programming languages they prefer to use. As an intern, I need to learn and work with these technologies even if I have experience with others.
- **Accessibility:** Ensuring that web applications are accessible to all users, including those with disabilities, can be a constraint. I need to follow accessibility standards and guidelines.

- **Team Collaboration:** Working within a team can also bring constraints in terms of communication, collaboration tools, and version control systems.

2.2.2. Constrained Time

Constrained time in the context of a web development internship refers to the limitations or pressures related to time that you might experience while working on projects or tasks within the company. Time constraints are common in professional settings, and they can impact how you plan, execute, and deliver your work. Here are some aspects of constrained time you might encounter during a web development internship:

- **Project Deadlines:** web development projects have specific deadlines that must be met. I'll need to manage my time effectively to ensure that I can complete tasks and deliverables on schedule.
- **Meetings and Collaborative Work:** Meetings and collaborative sessions with team members or stakeholders can consume a significant portion of my time. Balancing these activities with actual development work requires effective time management.
- **Code Reviews and Feedback:** The time allocated for code reviews and incorporating feedback from senior developers or team leads is another aspect where time constraints may apply.

2.3. Use Case

The use case diagram is important for defining and validating the required features in our application. Each use case box represents an important module that is implemented and contains other features within it. There are four types of users: Admin, Editor, Basic and Normal. While all user required must be log in before accessing to dashboard.

❖ Admin Use case

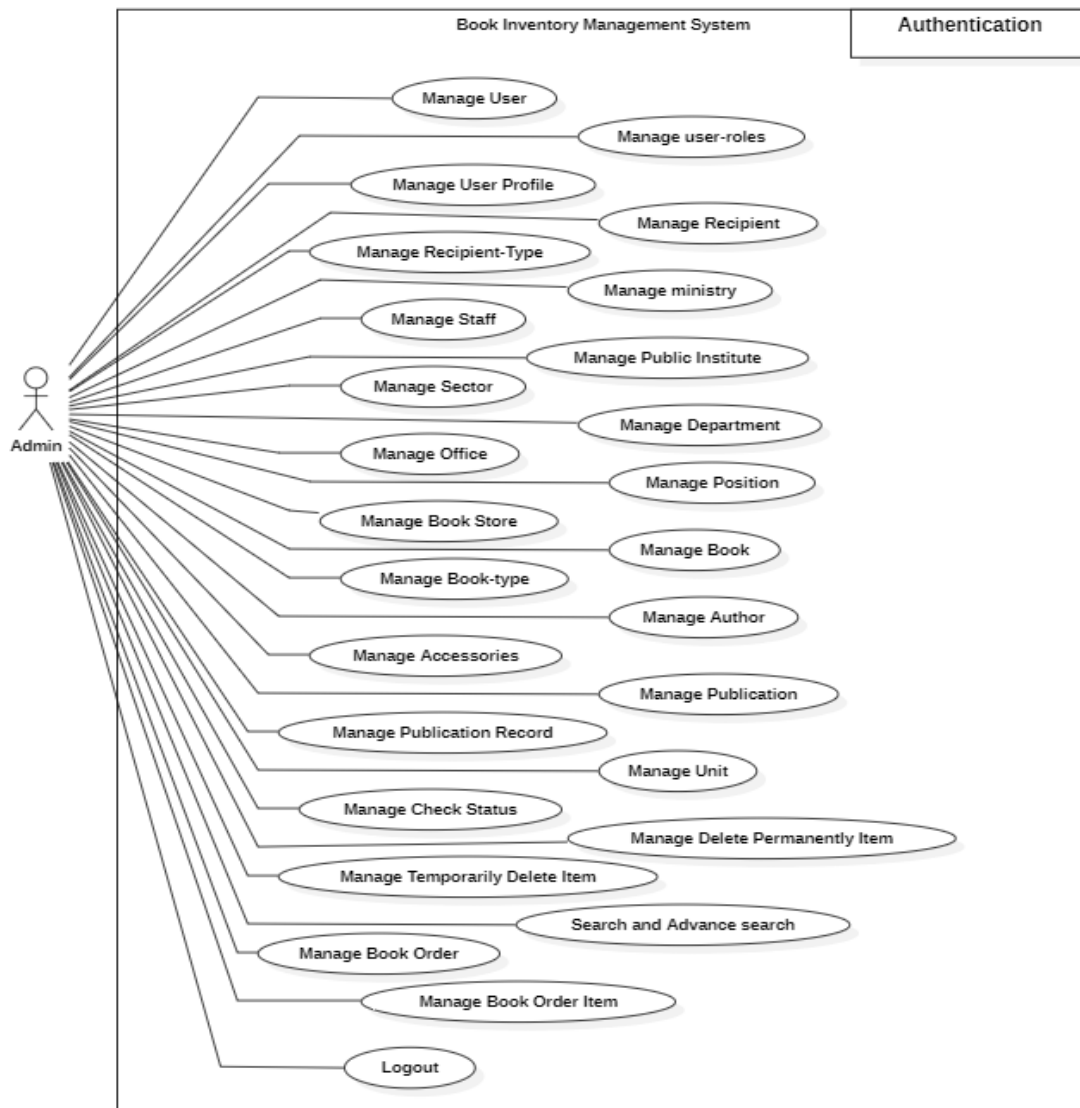


Figure 4: Use Case Admin Diagram

The use case above describes some functionality that the admin can interact with the web application.

- Admin after login success can to do all functionalities on the system such as all manage function.

❖ Editor Use case

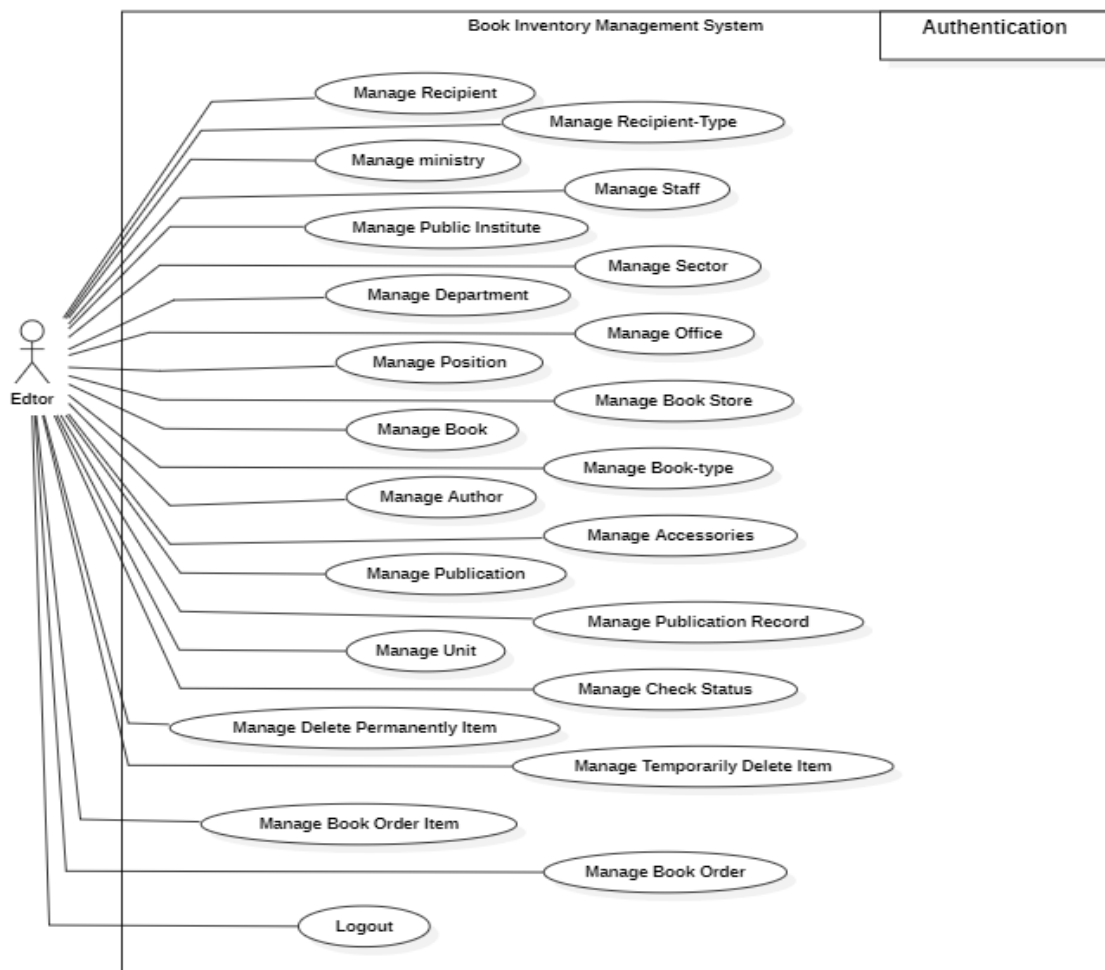


Figure 5: Use Case Editor Diagram

The use case above describes some functionality that the editor can interact with the web application.

- For editor after login success can do all manage functionalities on the system, but cannot create new user with role on the system.

❖ Basic Use case

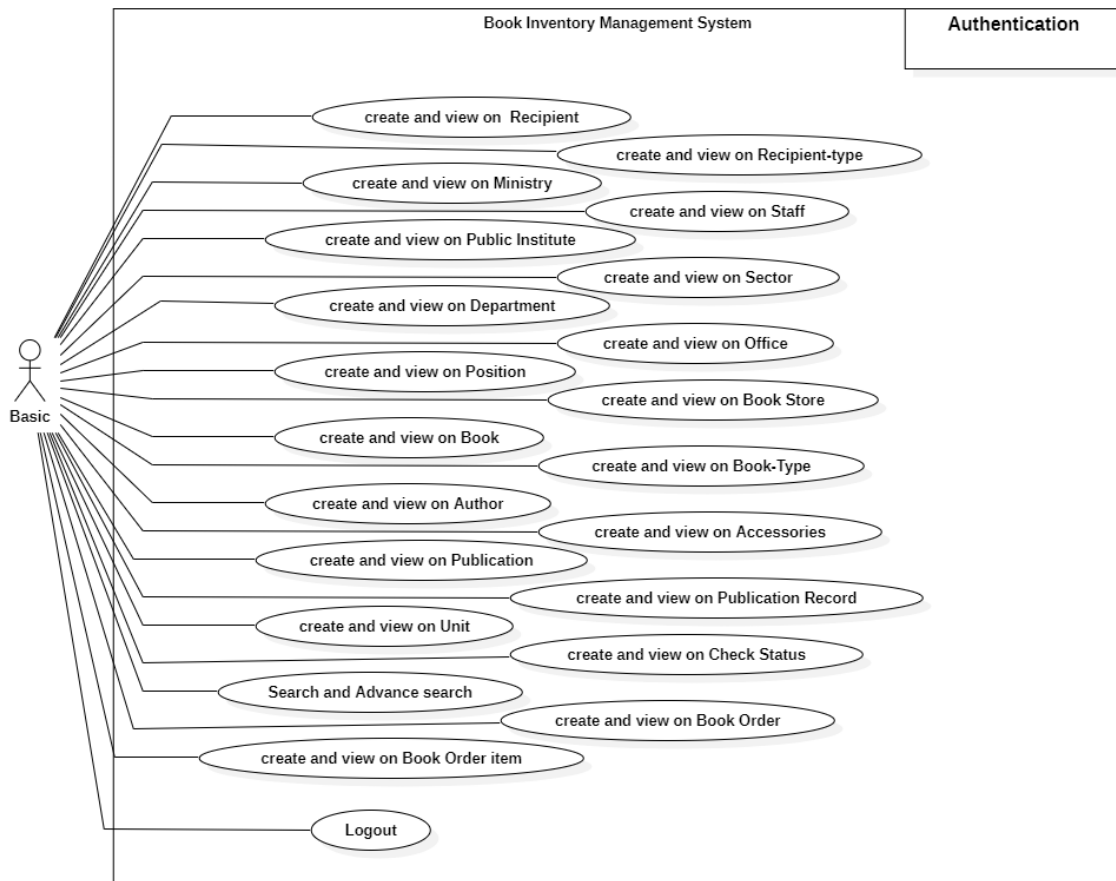


Figure 6: Basic Use Case Diagram

The use case above describes some functionality that the basic can interact with the web application.

- For editor after login success can do all manage functionalities on the system, but cannot create new user with role and cannot edit, delete information on the system.

❖ Normal Use case

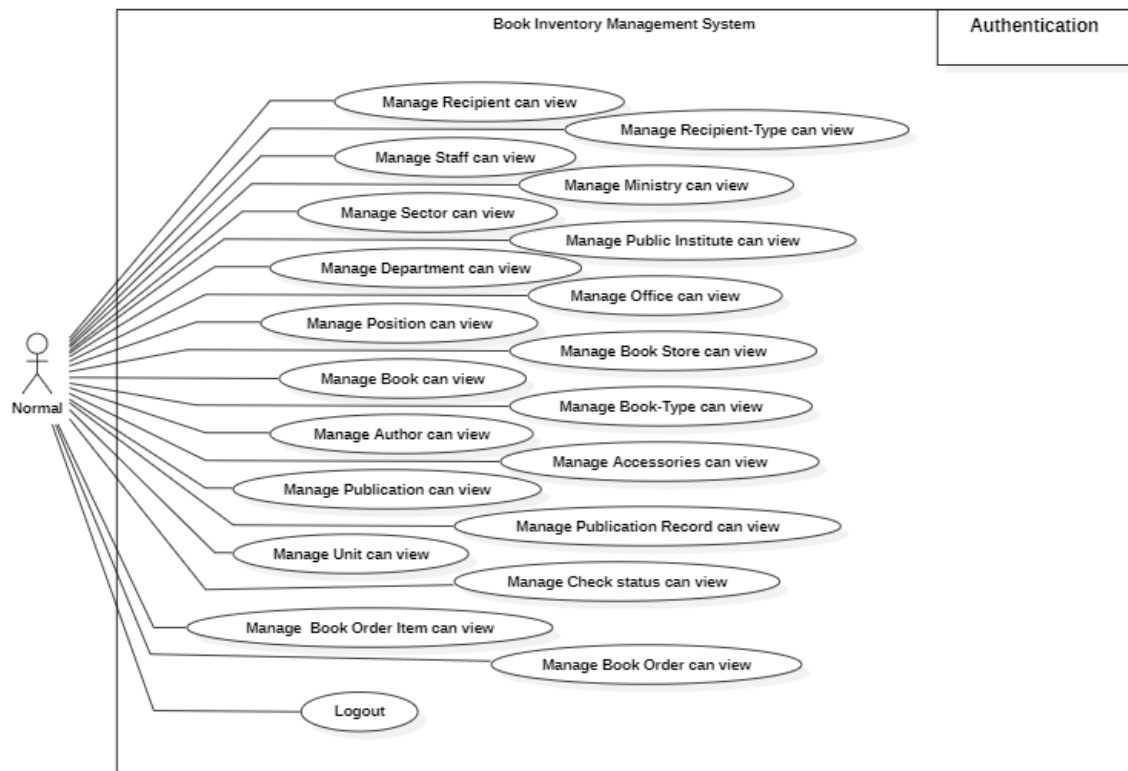


Figure 7: Use Case Normal Diagram

The use case above describes some functionality that the normal can interact with the web application.

- For normal after login success access to dashboard can view data only on the system.

2.4. Conception of Database

This section provides a simple introduction to the details of database design. Inside, you will discover a clear picture of our application's database, showing how different pieces of information are connected and explaining what's inside each table.

This is the overall database schema in our system that makes the system work. The figure below shows all the table that we use in this system, there are 22 tables. It's included the Users, User roles, Recipient Type, Recipients, Publications, Author, Book type, Unit, Accessories, Publication Record, Status Check, Public Institute, Ministry, Staff, Sector, Book Store, Department, Office, Position, Book Order, Book Order Items.

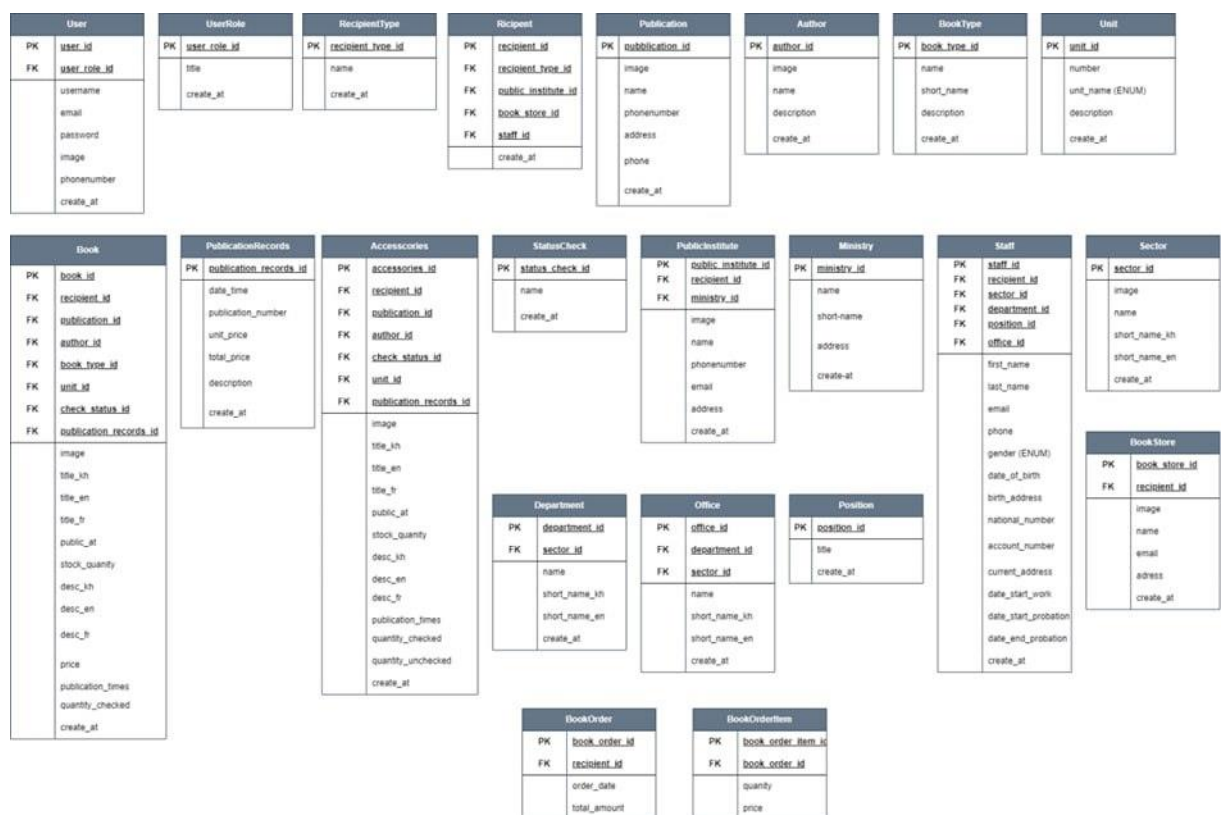


Figure 8: Database Design

III. TECHNOLOGY AND DESIGN

3.1. Choice of Technology

3.1.1. Programming Language

For develop this project we used some language such as HTML CSS JS SCSS:

HTML stands for Hypertext Markup Language. It is the standard markup language used for creating and structuring web pages. HTML uses a set of markup tags or elements to describe the structure and content of a web document.



Figure 9: HTML Logo

CSS is standing Cascading Style Sheet language used for style web page and describing the look and formatting of a document written in HTML. CSS separates the presentation a document from its structure, allowing web designers control the visual appearance of web pages.



Figure 10: CSS Logo

JS stands for JavaScript. It is a high-level programming language primarily used for adding interactivity and dynamic behavior to web pages. JavaScript is a versatile language that runs in web browsers, allowing developers to create interactive elements, handle events, manipulate the document object model (DOM), and communicate with servers.

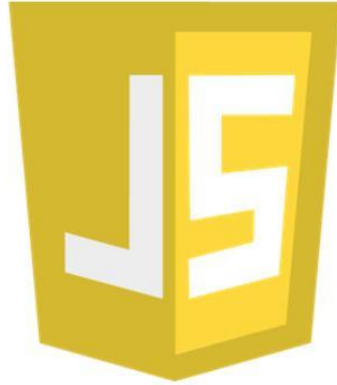


Figure 11: JavaScript Logo

SCSS (Sassy Cascading Style Sheets) is one of two syntaxes available for the popular CSS preprocessor Sass (Syntactically Awesome Stylesheets). It can be used to style the visual elements of a webpage, including buttons, sliders, images, color schemes, fonts, themes, and layouts. As a true superset of CSS, all valid CSS is also valid SCSS.



Figure 12: SCSS Logo

3.1.2. Framework Technology

We are used some framework to develop my project such as Angular and Nest JS.

Angular is a platform and framework for building single-page client applications using HTML and TypeScript and is a robust, full-featured framework for building dynamic and responsive web applications. Its component-based architecture, TypeScript integration, and comprehensive toolset make it a popular choice for developing modern, scalable web applications.



Figure 13: Angular Logo

Nest JS is a versatile and powerful framework for building server-side applications. Its use of TypeScript, modular architecture, and rich set of features make it an excellent choice for developing modern, scalable, and maintainable backend systems. By leveraging the principles and design patterns inspired by Angular, Nest JS offers a familiar and productive development experience for those already acquainted with Angular.



Figure 14: Nest JS Logo

TS (TypeScript): TypeScript (TS) is a strongly typed, object-oriented, compiled language developed and maintained by Microsoft. It is a superset of JavaScript, meaning any valid JavaScript code is also valid TypeScript code. TypeScript adds static typing, classes, and interfaces to JavaScript, providing a more robust development experience. TypeScript enhances JavaScript development by providing a robust type system, improved tooling, and features that support the development of large-scale and maintainable applications.



Figure 15: TypeScript logo

3.1.3. Tool

We are used many tools to create this project such as:

Short for **Visual Studio Code**, we used for write code this project and it is a free and opensource code editor developed by Microsoft. It has gained significant popularity among developers due to its extensive features, cross-platform compatibility, and a large ecosystem of extensions.



Figure 16: VS Code Logo

GitLab used for store code and it is a web-based platform that provides hosting services for version control using Git. It offers a centralized location for developers to collaborate, share, and manage their software projects. GitLab combines code hosting, version control, issue tracking, and collaboration features, making it a popular choice for individual developers and teams. GitLab essentials like repositories, branches, commits and pull request.



Figure 17: GitLab Logo

Postman is a popular collaboration platform and API development tool used by developers to design, test, and document APIs (Application Programming Interfaces). It provides a user-friendly interface that simplifies the process of working with APIs and enhances the efficiency of API-related tasks.



Figure 18: Postman Logo

Figma is used for draw UI on my project and a cloud-based design and prototyping tool used for creating user interfaces (UI), user experience (UX) designs, and interactive prototypes. It provides a collaborative environment for designers, developers, and stakeholders to work together on design projects in real-time.



Figure 19: Figma Logo

Swagger UI is an open-source tool that allows developers to visualize and interact with an API's resources without having any of the implementation logic in place. It provides a user-friendly interface for exploring and testing the endpoints of a RESTful API, making it easier for developers to understand the capabilities and behaviors of the API.



Figure 20: Swagger Logo

MongoDB is a versatile and powerful NoSQL database that provides flexibility, scalability, and high performance for modern application development. Its document-oriented approach and dynamic schema make it a great choice for a wide range of use cases, from content management to real-time analytics and IoT applications. With rich query capabilities, robust indexing, and support for complex transactions, MongoDB is a solid choice for developers looking to build scalable and responsive applications



Figure 21: MongoDB Logo

3.2. System Design

System architecture is the concept model that defined the structure, behavior and other views of a system. There are two components is physical and logical architecture and following describes the detail of process about those architecture in our website.

3.2.1. Physical Architecture

Physical architecture describes about the external architecture such as devices used in our website in order to get it up and running.



Figure 22: Physical Architecture

The physical architecture of a system plays a crucial role in ensuring the reliable and efficient functioning of the software application. It defines the distribution and arrangement of the system's components, including the client web browser, web server, database server, and

file system, among others. The physical architecture is essential because it addresses the practical aspects of deploying and running the system in a real- world environment.

- **Client side:** Users access the system through a web browser. They interact with the system by sending HTTP requests to the web server.
- **Web Server:** The web server receives the HTTP requests from clients and processes them. It uses Nest JS, which is a JavaScript runtime environment, along with an instructs framework and makes communication with the database server.
- **Database Server (MongoDB):** This is the server that stores and manages the application's data. In this case, the database server is using MongoDB, a popular NoSQL database. The web server communicates with the database server to retrieve, store, or update data as needed to fulfill the client's requests.

3.2.2. Logical Architecture

The logical architecture as shown and describes the idea of how the struts framework structure to build the web application with Nest JS as API and Angular as Front-end.

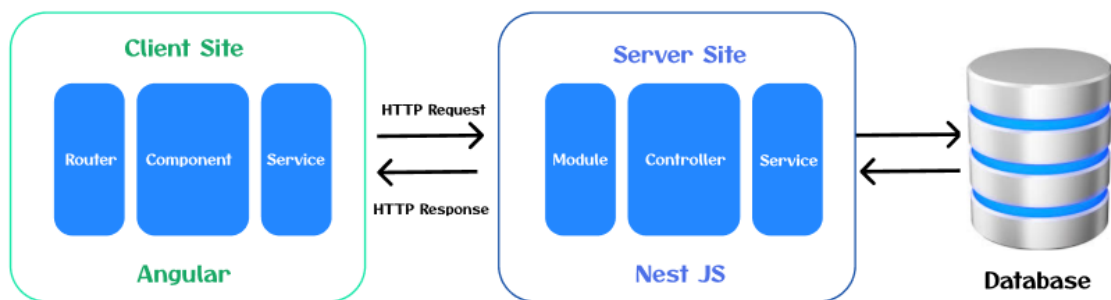


Figure 23: Logical Architecture

- ❖ **Nest JS** is a Node.js framework used to build efficient, powerful, and scalable server-side applications. It follows a modular architecture, allowing for flexibility and ease of upkeep. Some key components of the Nest JS architecture include:
 - **Module:** Nest JS applications are organized into self-contained modules, enhancing flexibility, maintainability and help in organizing and separating different parts of the application.
 - **Controller:** The controller component acts as an intermediary between the client and the backend logic. It receives requests from the client, processes them, and coordinates the appropriate actions within the system. It handles the business logic, validates inputs, invokes necessary operations on the service, and prepares responses to be sent back to the client.

- **Service:** Contain the business logic and are injected into controllers.
- ❖ **Angular** is a front-end framework used to build rich and interactive single-page applications. The architecture of an Angular application typically includes the following components:
 - **Route:** define the end point of the page.
 - **Component:** Building blocks of an Angular application, representing the view layer. They are encapsulated, loosely coupled, and have defined inputs and outputs.
 - **Service:** can communicate with external APIs or backend services through HTTP requests or WebSocket connections.

3.2.3. Sequential Diagram

- ❖ User login

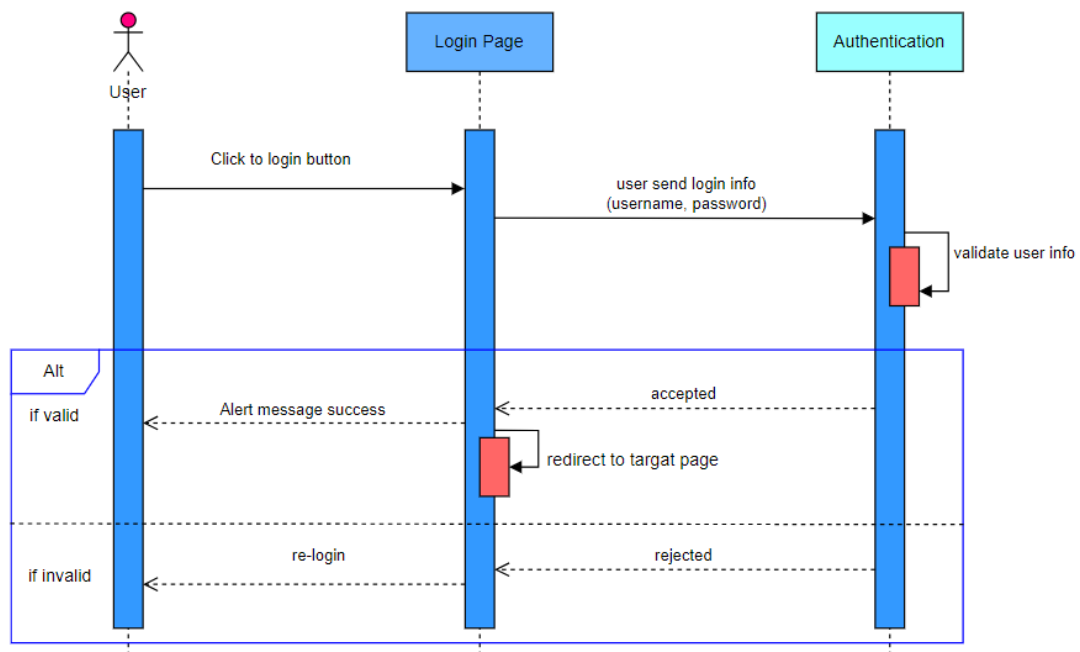


Figure 24: Sequential Diagram User Login

- User request to server for login page and fill username and password to login.
- The next validation will check the username and password. If the username and password are correct, it will allow the user to log in successfully with an alert message and direct them to the home page of the dashboard. Moreover, if the username and password are not correct, it will not allow the user to access the dashboard and display an alert message stating that the username or password is incorrect.

❖ Crud on manage User

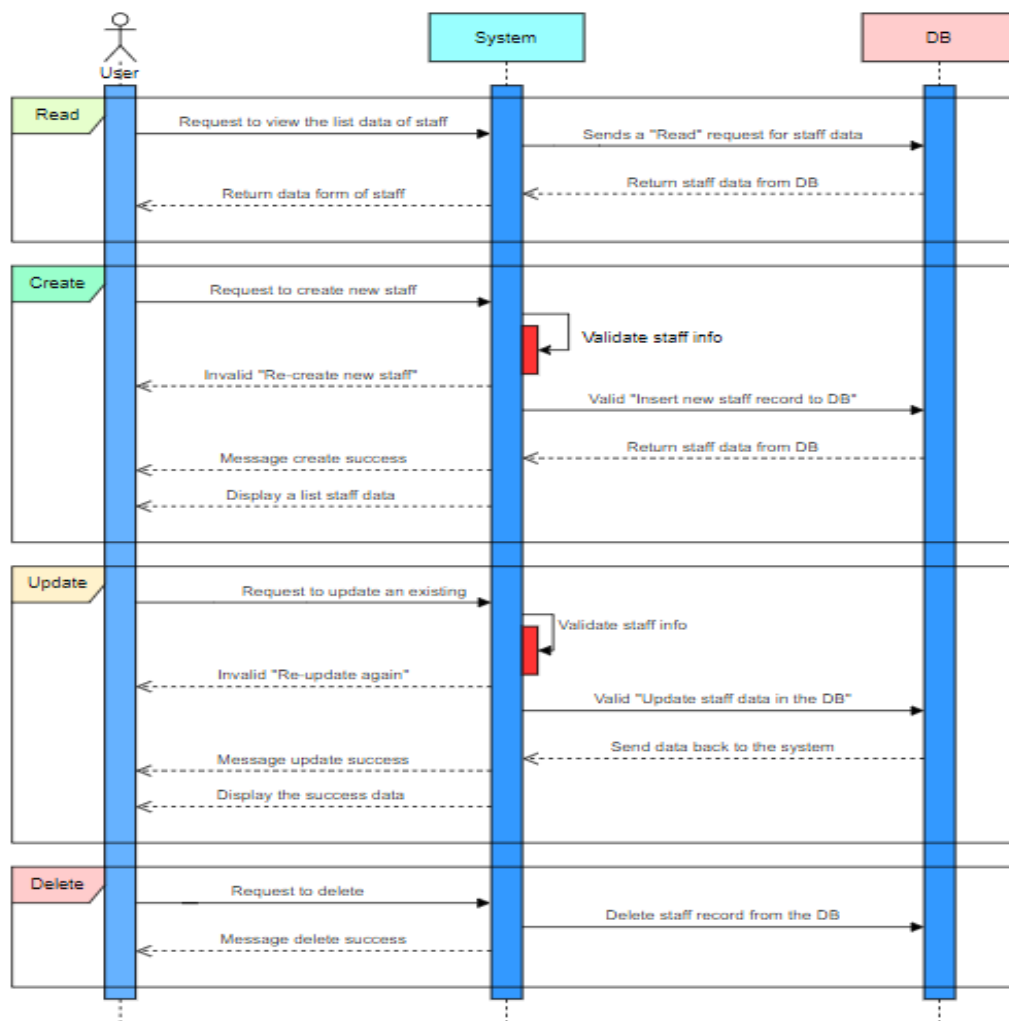


Figure 25: Sequential Diagram Crud on User

- Crud (create, read or view, update, delete), operations for user management is a common practice in software development. Here are some reasons why CRUD operations are useful when managing:
 - Create: The "Create" operation allows you to add new user to the system. When managing user, you need the ability to create news and store their information in a database.
 - Read Or View: The "Read" operation enables you to retrieve user information from the system. It allows you to view user details, or any other relevant information associated with each user. Reading user data is necessary for displaying user profiles, generating reports, or performing various tasks related to user management.

- Update: The "Update" operation allows you to modify user details. User information can change over time, such as updating or modifying their profile information. The ability to update user data ensures that the system stays up-to-date with the latest information.
- Delete: The "Delete" operation enables you to remove users from the system. There may be scenarios where you need to deactivate or delete a user account, either due to inactivity, policy violations, or other reasons. The delete operation allows you to remove the user's data and ensure it is no longer accessible within the system.

IV. PROJECT IMPLEMENTATION

This section, detail the implementation of the project during the internship. I explain how to set up tools and technology, project implementation, and installment.

4.1. Project Setup

The project setup for the Book Inventory Management system involves several steps to prepare the development environment and establish the necessary infrastructure. Firstly, the front-end of the system is built using Angular, a popular JavaScript framework.

On the other hand, the back-end of the system is developed using Nest JS framework known for its elegance and simplicity. The developer sets up the Nest JS development environment by installing Nest JS CLI.

4.1.1. Environment setup

To set up the required environment and tools for my project, follow these steps:

- Install Nest JS version 10.3.2
- Install the IDE. In my case, I choose IntelliJ IDEA and Visual Studio Code for the code editor.
- Install the Postman or swagger UI for testing API.
- Install MongoDB for storing data and manage the database.

4.1.2. Frontend setup project

Step 1: create a folder on the desktop and navigate to the folder through terminal

Step 2: Run command line on the terminal

npm install -g @angular/cli

Step 3: After that we have select some option in terminal section to our project and then run command line on the terminal.

`. cd project name`

`. npm install or yarn install`

Step 4: Now navigate to our app folder through command line. Following this picture show the project structure:

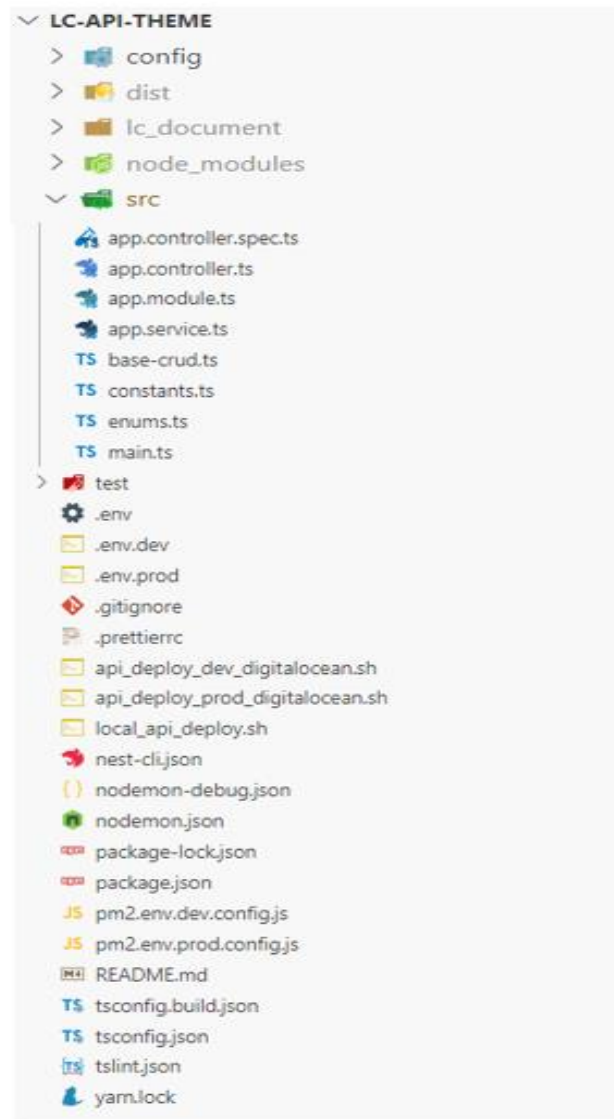


Figure 26: Frontend Structure

Project structure specifications:

- **Node modules folder:** This folder contains all the project's dependencies, including Angular itself and third-party libraries on this project.
- **Src folder:** This folder contains the source code on Angular and the main module of the application.
- **Core folder:** This folder might contain core services, modules, or components that are fundamental to the application's functionality.
- **Pages folder:** This folder contains components representing different pages or views of the application.
- **Shared folder:** This folder often holds component, service that share the others button action on the component.
- **App-routing.module.ts:** is route on the web application to routing configuration determining how different URL paths map to specific components.
- **App.component.html:** This file contains the template for the application's main component.
- **App.module.ts:** This file defines the main module of the application, including its components, services, and dependencies.
- **Assets:** This folder stores static assets like images, fonts, or other files that are not compiled by Angular.

Step 5: Run our project by using the command line *ng serve*

After run command line on your terminal, you will be seeing URL below you can click URL to access web browser ([http://localhost: 4200](http://localhost:4200)).

4.1.3. Backend setup project

Step 1: create a folder on the desktop and navigate to the folder through terminal

Step 2: Run command line on the terminal

npm install -g @nestjs/cli

Step 3: After that we have select some option in terminal section to our project and then run command line on the terminal.

. cd project name

. npm install or yarn install

Step 4: Now navigate to our app folder through command line. Following this picture show the project structure

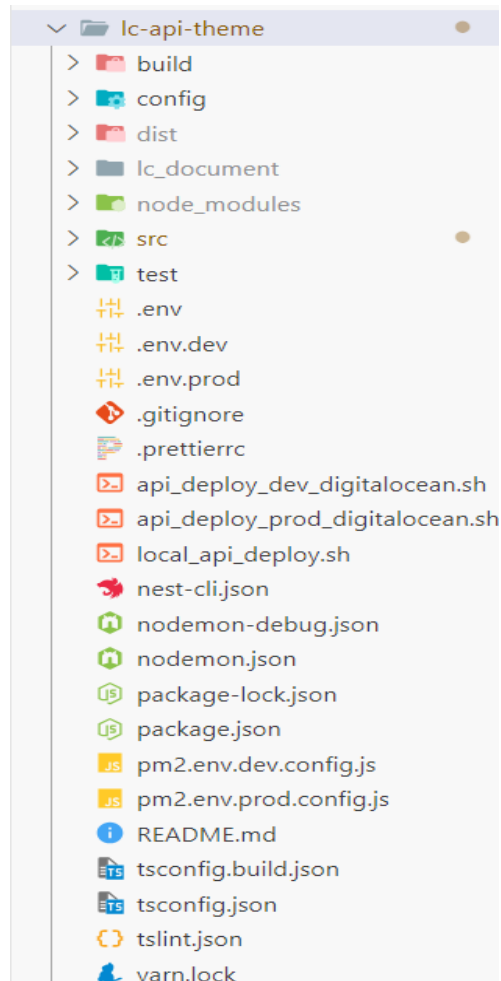


Figure 27: Backend Structure

Project structure specifications:

- **LC document folder:** Tts is folder to store all image when admin upload image to the system.
- **Node modules folder:** This folder contains all the project's dependencies, including Nest JS itself and third-party libraries.

- **Src folder:** It is folder contains the source code directory on the nest JS application and folder schema.
- **App.controller.ts:** This file defines the controller for the application, which handles incoming requests and routes them to the appropriate services.
- **App.module.ts:** This file defines the main module of the application, including its controllers, services, and dependencies.
- **App.service.ts:** This file defines the service for the application, which handles the application's business logic.
- **Main.ts:** This file bootstraps the Nest JS application, setting up the initial environment and starting the application.
- **Env:** It is file connect to MongoDB such as name database, username, password and root database.
- **Package.json:** This file defines the project's metadata, dependencies, and scripts.
- **Tsconfig.json:** This file contains configuration settings for the TypeScript compiler.

Step 5: Run our project by using the command line ***nest start - -watch***

4.2. Implementation

To implement front-end or user interface according to all required functions is not an easy task to do. Some flows are easy to develop but some flows are difficult to develop and take some time to deal with. However, it was quite challenging to solve all those problems and gain new experience in web development. Below are some functions that I have overcome with and some descriptions about them.

4.2.1. Recipients Module

4.2.1.1. Manage Recipient

- After login success users can access directly to dashboard and can manage recipient.
- So, the next admin and editor can perform CRUD operations on the recipient. For creating a recipient, input all the data on form create related joint with table recipient type, table public institute, table book store and table staff to create a new recipient and click the submit button to finalize the process. If the data is valid, the system will save the data and display a success alert message. The list data option allows viewing all data on the web page. For editing data, click the edit button and navigate to the form containing the data you want to edit. After editing the data, click the save button to finalize the process and store the data in the database and alert message. Lastly, for deleting a recipient, click the delete button to finalize the process and remove the data from the database and alert message.

4.2.1.2. Manage Recipient Type

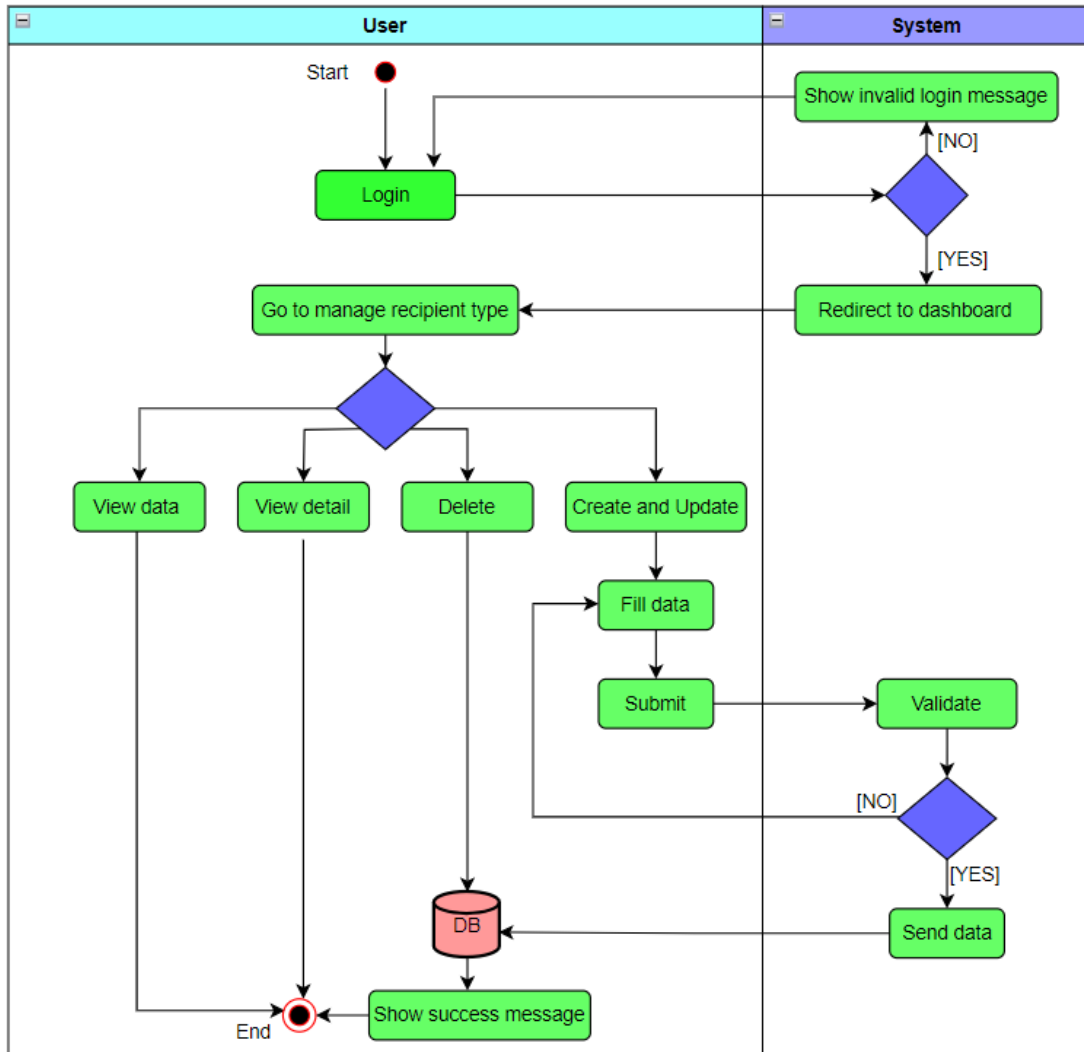


Figure 28: Activity Crud on Recipient Type

- After login success users can access directly to dashboard and can manage recipient type.
- So, the next admin can perform CRUD operations on the recipient type. For creating a recipient type, input all the data on form create to create a new recipient type and click the submit button to finalize the process. If the data is valid, the system will save the data and display a success alert message. Once a recipient type with a particular name is created in the database, it cannot be created again. The list data option allows viewing all data on the web page. For editing data, click the edit button and navigate to the form containing the data you want to edit. After editing the data, click the save button to

finalize the process and store the data in the database and alert message. Lastly, for deleting a recipient type, click the delete button to finalize the process and remove the data from the database and alert message.

4.2.1.3. Manage Ministry

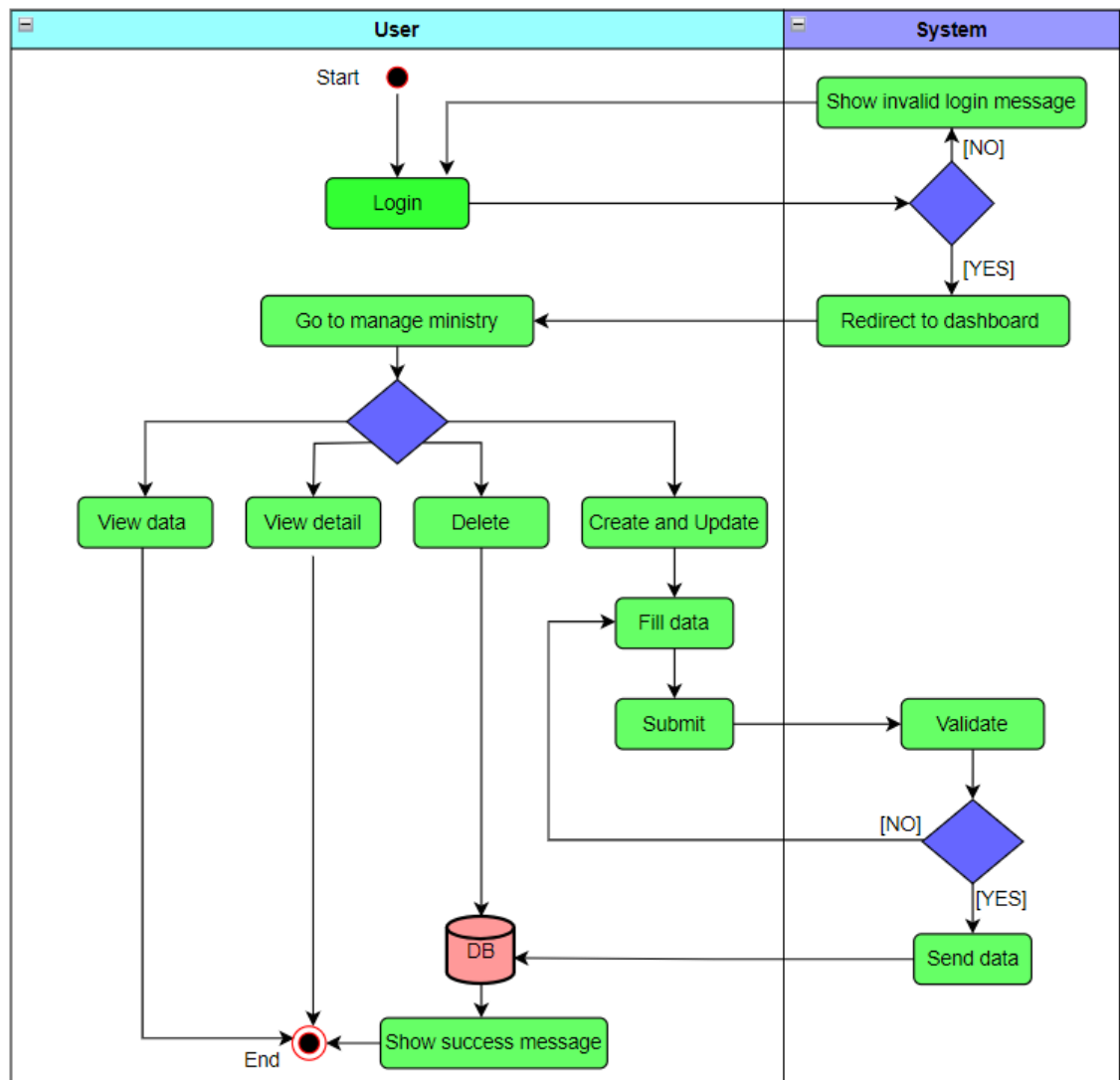


Figure 29: Activity Crud on Ministry

- Firstly, we need to login access to admin dashboard and check validation the user login. If user input username or password correct can redirected to the Dashboard and fail the system displays an error message indicating an invalid login.
- Secondly, upon successful login, administrators gain access to the "User role Management" feature within their profile. They can the process to input of all data of the create new ministry and click the Submit button to finalize the process. If data valid

system going to save data and alert message success. When create new ministry if name of ministry has done on database is cannot create.

4.2.1.4. Manage Staff

- After login success users can access directly to dashboard and can manage staff.
- So, the next admin and editor can perform CRUD operations on the staff. For creating a staff, input all the data on form create related joint with table recipient, table sector, table department, table position and table office to create a new staff and click the submit button to finalize the process. If the data is valid, the system will save the data and display a success alert message. Once a staff with a particular email is created in the database, it cannot be created again. The list data option allows viewing all data on the web page. For editing data, click the edit button and navigate to the form containing the data you want to edit. After editing the data, click the save button to finalize the process and store the data in the database and alert message. Lastly, for deleting a staff, click the delete button to finalize the process and remove the data from the database and alert message.

4.2.1.5. Manage Public Institute

- After login success users can access directly to dashboard and can manage public institute.
- So, the next admin and editor can perform CRUD operations on the public institute. For creating a public institute, input all the data on form create related joint with table recipient and table ministry to create a new public institute and click the submit button to finalize the process. If the data is valid, the system will save the data and display a success alert message. Once a public institute with a particular email is created in the database, it cannot be created again. The list data option allows viewing all data on the web page. For editing data, click the edit button and navigate to the form containing the data you want to edit. After editing the data, click the save button to finalize the process and store the data in the database and alert message. Lastly, for deleting a public institute, click the delete button to finalize the process and remove the data from the database and alert message.

4.2.1.6. Manage Sector

- After login success user can access to dashboard directly.
- So, the next admin can perform CRUD operations on the sector. For creating a sector, input all the data on form create to create a new sector and click the submit button to finalize the process. If the data is valid, the system will save the data and display a success alert message. Once a sector with a particular name is created in the database, it cannot be created again. The list data option allows viewing data on the web page. For editing data, click the edit button and navigate to the form containing the data you want to edit. After editing the data, click the save button to finalize the process and store the data in the database and alert message. Lastly, for deleting a sector, click the delete button to finalize the process and remove the data from the database and alert message.

4.2.1.7. Manage Department

- After login success users can access directly to dashboard and can manage department.
- So, the next admin and editor can perform CRUD operations on the department. For creating a department, input all the data on form create related joint with table sector to create a new department and click the submit button to finalize the process. If the data is valid, the system will save the data and display a success alert message. Once a department with a particular name is created in the database, it cannot be created again. The list data option allows viewing all data on the web page. For editing data, click the edit button and navigate to the form containing the data you want to edit. After editing the data, click the save button to finalize the process and store the data in the database and alert message. Lastly, for deleting a department, click the delete button to finalize the process and remove the data from the database and alert message.

4.2.1.8. Manage Office

- After login success users can access directly to dashboard and can manage office.
- So, the next admin and editor can perform CRUD operations on the office. For creating an office, input all the data on form create related joint with table sector and table department to create a new office and click the submit button to finalize the process. If the data is valid, the system will save the data and display a success alert message. Once an office with a particular name is created in the database, it cannot be created again. The list data option allows viewing all data on the web page. For editing data, click the edit button and navigate to the form containing the data you want to edit. After editing

the data, click the save button to finalize the process and store the data in the database and alert message. Lastly, for deleting an office, click the delete button to finalize the process and remove the data from the database and alert message.

4.2.1.9. Manage Position

- After login success users can access directly to dashboard and can manage position.
- So, the next admin can perform CRUD operations on the position. For creating a position, input all the data on form create to create a new position and click the submit button to finalize the process. If the data is valid, the system will save the data and display a success alert message. Once a position with a particular title is created in the database, it cannot be created again. The list data option allows viewing all data on the web page. For editing data, click the edit button and navigate to the form containing the data you want to edit. After editing the data, click the save button to finalize the process and store the data in the database and alert message. Lastly, for deleting a position, click the delete button to finalize the process and remove the data from the database and alert message.

4.2.1.10. Manage Book Store

- After login success users can access directly to dashboard and can manage book store.
- So, the next admin and editor can perform CRUD operations on the book store. For creating a book store, input all the data on form create related joint with table recipient to create a new book store and click the submit button to finalize the process. If the data is valid, the system will save the data and display a success alert message. Once a book store with a particular name is created in the database, it cannot be created again. The list data option allows viewing all data on the web page. For editing data, click the edit button and navigate to the form containing the data you want to edit. After editing the data, click the save button to finalize the process and store the data in the database and alert message. Lastly, for deleting a book store, click the delete button to finalize the process and remove the data from the database and alert message.

4.2.2. Search Module

4.2.2.1. Search

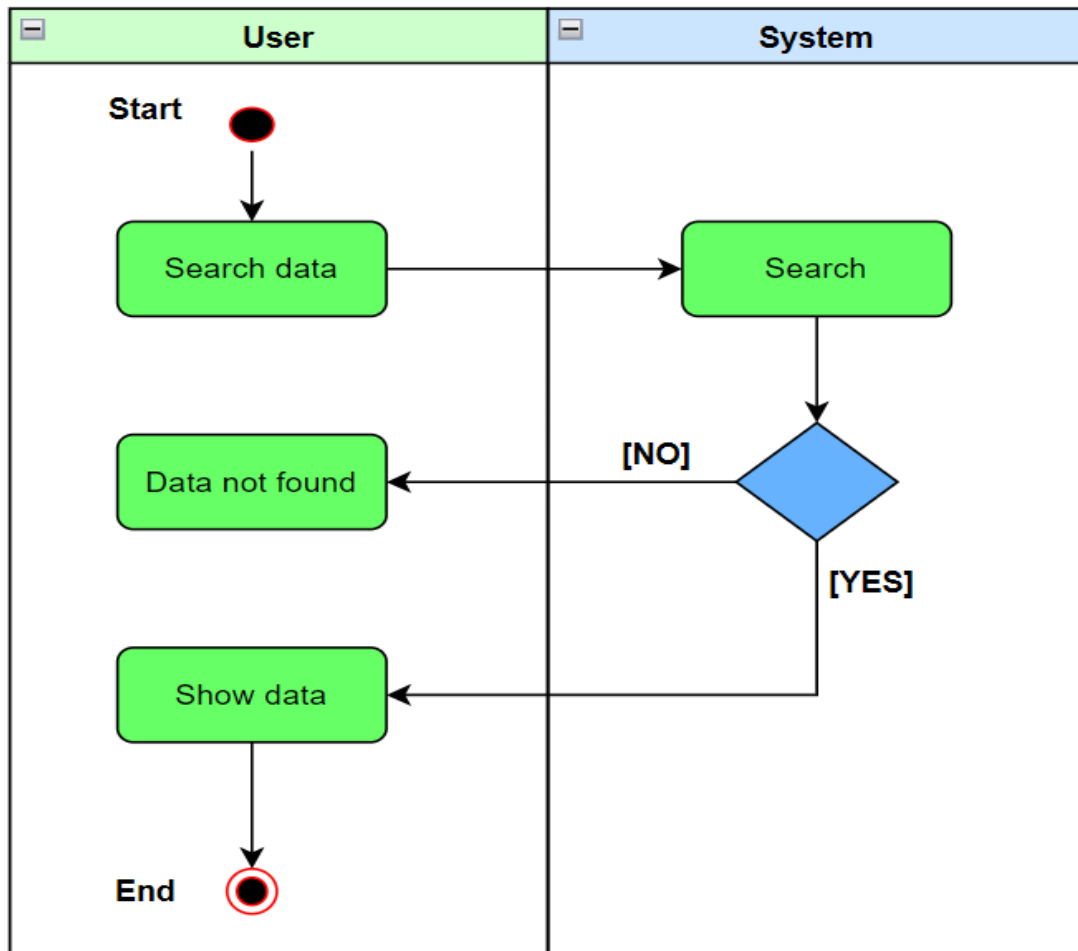


Figure 30: Activity Diagram of Search

- After login success users can access directly to dashboard and can view list data.
- The dashboard contains many lists of data, each of which can be managed. If a user wants to know more about the information, they can enter data in the search button. The search will then finalize the process and display all data matching the keyword.

4.2.2.2. Search Advance

- The advance search to do on module book, accessories and book order.
- The dashboard contains many lists of data, each of which can be managed. If a user wants to know more about the information, user can searches based on keyword date intervals, years and short list character, they can enter data in the search button. The search will then finalize the process and display all data matching the keyword.

4.2.3. Manage Offer Module

4.2.3.1. Manage Book Order

- After login success users can access directly to dashboard and can manage book order.
- So, the next admin and editor can perform CRUD operations on the book order. For creating a book order, input all the data on form create related joint with table recipient to create a new book order and click the submit button to finalize the process. If the data is valid, the system will save the data and display a success alert message. The list data option allows viewing all data on the web page. For editing data, click the edit button and navigate to the form containing the data you want to edit. After editing the data, click the save button to finalize the process and store the data in the database and alert message. Lastly, for deleting a book order, click the delete button to finalize the process and remove the data from the database and alert message.

4.2.3.2. Manage Book Order Item

- After login success users can access directly to dashboard and can manage book order item.
- So, the next admin and editor can perform CRUD operations on the book order item. For creating a book order item, input all the data on form create related joint with table book order to create a new book order item and click the submit button to finalize the process. If the data is valid, the system will save the data and display a success alert message. The list data option allows viewing all data on the web page. For editing data, click the edit button and navigate to the form containing the data you want to edit. After editing the data, click the save button to finalize the process and store the data in the database and alert message. Lastly, for deleting a book order item, click the delete button to finalize the process and remove the data from the database and alert message.

4.2.4. Telegram Integration Module

4.2.4.1. Telegram Bot Alert

Integrating with a Telegram bot to alert the Inventory Manager about important events or updates in the system can significantly enhance the efficiency and responsiveness of inventory management. By providing real-time notifications, the bot ensures that the Inventory Manager is immediately informed of critical changes, such as low stock levels, incoming shipments, or system errors.

V. SUMMARY AND CONCLUSION

5.1. Summary

5.1.1. Complete Work

After finishing the internship program at Ministry Economy and finance, the result of the project Book Inventory Management System will be illustrated in the table below:

Table 6: Table of Tasks Completed

Tasks	Status	Responsibility
Authentication and Authorization	Completed	SEK Sotheavy
Manage User	Completed	SEK Sotheavy
Manage Profile	Completed	SEK Sotheavy
Manage User Role	Completed	SEK Sotheavy
Manage Recipient	Completed	VEN Thon
Manage Recipient Type	Completed	VEN Thon
Manage Ministry	Completed	VEN Thon
Manage Staff	Completed	VEN Thon
Manage Public Institute	Completed	VEN Thon
Manage Sector	Completed	VEN Thon
Manage Department	Completed	VEN Thon
Manage Office	Completed	VEN Thon
Manage Position	Completed	VEN Thon
Manage Book Store	Completed	VEN Thon
Manage Book	Completed	SEK Sotheavy
Manage Book Type	Completed	SEK Sotheavy
Manage Author	Completed	SEK Sotheavy

Manage Accessories	Completed	SEK Sotheavy
Manage Publication	Completed	SEK Sotheavy
Manage Publication Record	Completed	SEK Sotheavy
Manage Unit	Completed	SEK Sotheavy
Manage Check Status	Completed	SEK Sotheavy
Calculate Total	Completed	SEK Sotheavy
Search and advance search	Completed	VEN Thon
Manage Book Order	Completed	VEN Thon
Manage Book Order Detail	Completed	VEN Thon

5.1.2. Incomplete Work

On my project, we have many tasks completed but we have some tasks no complete. There are some points uncomplete in have to improve on the table below such as:

Table 7: Table of Tasks Incomplete

Tasks	Status	Responsibility
Manage Temporarily Delete Item	InProgress	SEK Sotheavy
Manage Delete Permanently Item	InProgress	SEK Sotheavy
Download Report	InProgress	SEK Sotheavy
Export Report	InProgress	SEK Sotheavy
Telegram Bot Alert	InProgress	VEN Thon
Display lates Recipient	InProgress	SEK Sotheavy
Latest Order Item	InProgress	SEK Sotheavy

5.2. Strong Point

In this internship program, I have worked hard on the project and have successfully conducted so many strong points as follows:

- Authentication and validation work efficiently
- Most of the functionalities are very responsive
- Loading speed is very fast
- Performance is acceptable
- Comprehensible and attractive UI
- Easy control of the system
- Can control data of user

5.3. Weak Point

Even though the system was developed with many strong points, there are still some weak points that need improvement and further development. These include:

- Efficient storage of file images can upload all image and extension.
- Enhancing configuration options to facilitate user-friendly configuration.

5.4. Experiences

Honestly, three months of internship is quite a short period, and acquiring knowledge. Following are experiences that I have received from my internship including soft skills and hard skills:

❖ Firstly, I have acquired hard skills such as:

- Learning and practicing new technologies in real project with Angular, Nest JS and MongoDB as technologies in the process of full stack development.
- Researching about using new technologies, debugging and finding solution to the problems related to coding and implementing.
- Full stack development
- Apply new technology
- Time management
- Self-learning
- Database management
- Working independence

❖ Secondly, I have developed soft skills such as:

- Asking for help when needed with my colleagues on my team.
- Communicate with the team's project.
- Solving difficulties in development process by taking and giving suggestions during weekly meetings.
- Dealing with problems with multiple solution and working as a team in a real work environment.

Therefore, I have gained valuable experience and knowledge from this internship that will help me in my future career. In conclusion, I am grateful for this opportunity and I appreciate the guidance and support from my mentors and colleagues.

5.5. Difficulties

During this internship, I encountered numerous difficulties and problems, with the primary issue being the limitation of work analytics. I have had to learn and adapt to a new project that is entirely unfamiliar to me. This marks the first time I have embarked on solo development, and I find it exceedingly challenging. When faced with problems, I often have to rely solely on my solutions, which may not always yield satisfactory results. As a result, I frequently find myself seeking assistance from my project advisor to make necessary changes and seek guidance. Regarding many issues during the implementation, it is also a valuable experience that I get from this internship.

5.6. Perspective

This project is a long-term development since this project have many functions and requirement. During my three months internship, I am able to solve some main problems to develop the website that are proposed by the manager. Currently, some functions have been completed as we will complete it in the next version. I am very interested in Book Inventory Management System project because I like some content, and I can get to know about to design website. I gain a lot of knowledge and new experiences at work place like group work, work with self- responsible like managing the time, doing research and how to study and adapt with new technology faster to save time for development. If I have enough time to develop on this project, I have some desired perspectives to fulfill the web application as the following.

5.7. Conclusion

The internship is finally very vital for students to gain social and academic skills. After finishing the internship, I become fully aware of the fact that the knowledge we have learned in class is like a foundation for us to adapt to every new thing that we need in our future job. Furthermore, this internship program allows me to get more experience, I can practice the use of the methodology of project management and how to use it properly. I am sure that all of these experiences will be the most valuable for my future career.

References

[1] Introduction to Nest JS. (n.d.). Introduction to Nest JS. Retrieved May 30, 2024, from

<https://docs.nestjs.com/>

[2] Introduction to Angular. (n.d.). Introduction to Angular. Retrieved May 30, 2024, from

<https://angular.io/docs>

[3] Introduction and explain of type script. Retrieved Jun 12, 2024 from

https://www.youtube.com/watch?v=2pZmKW9-I_k&list=PL4cUxeGkcC9gUgr39Q_yD6v-bSyMwKPUI&index=2

APPENDIX

