វិទ្យាស្ថានជាតិប្រៃសណីយ៍ ទូរគមនាគមន៍ បច្ចេកវិទ្យាគមនាគមន៍ និងព័ត៌មាន
NATIONAL INSTITUTE OF POSTS, TELECOMMNUNICATIONS AND INFORMATION COMMUNICATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

ស្មាតអ៊ិនត្រាណេត

SMART INTRANET

A Thesis

In Partial Fulfillment of the Requirement for the Degree of

Bachelor Program of Computer Science

Submitted By:

Mr. OUCH Kongkea

Under the supervision of:

Mr. LY Rattana

October 2018

ACKNOWLEDGEMENT

First and foremost, I would like to dedicate my wholehearted thank and gratitude to **my mom and dad** for bringing me into this world and raising me up to be who I am today. Without your continuous support and motivation, I would not have made it this far to the point of almost graduating from a prestigious university and having interned at a world-class company.

I would like to express my sincere gratitude to His Excellency **Dr. SENG Sopheap**, President of **National Institute of Posts, Telecoms and ICT**, for essentially founding **NIPTICT** with a spectacular vision of creating new human resource driven by the application of entrepreneurship and technological innovation for a digitally developed Cambodia.

I would like to specially thank **Mr. CHOUR Porchourng**, Head of the Department of Computer Science, for his tireless effort in directing and effectively coordinating with everyone in the department with the single goal of achieving excellence for both students and lecturers.

I would like to extends my deepest gratitude to **Mr**. **LY Rottana** for being such a helpful and accommodating advisor during the internship program. Whenever I came across obstacle in internship, he was always there to help me find solution by trying to understand the problem from my perspective and then providing valuable advice from his experience. Moreover, I was able to perform work smoothly in an organized manner thanks to the knowledge of software engineering that he passionately imparted to me during the junior year.

I would like to greatly thank **Mr. HANG Sopheak** and **Mr. PEN Lymeng**, both senior web developers, for their invaluable supervising and mentoring me on what it takes to be a professional web developer that succeeds in this competitive technology-driven world. He willingly guided, assigned and trusted me with real projects of the company; thus, I was able to complete the required work in time and be equipped with hands-on knowledge and experience that are useful for my future career.

Last but not least, tremendous thanks and greetings to my fellow **CS3** classmates and lecturers for your support and gift of knowledge that have shaped me into who I am today.

មូលដ្ឋានសង្ខេប

របាយការណ៍នេះរៀបរាប់អំពីការងាររបស់ខ្ញុំក្នុងអំឡុងពេលចុះកម្មសិក្សារយៈពេល ៣ ខែ នៅក្រុមហ៊ុនស្មាតខូអិលធីឌី ក្នុងនាមជាអ្នកអភិវឌ្ឍន៍វេបអេបភ្លីខេសិន។ កម្មសិក្សានេះគឺជាផ្នែកមួយដ៏សំខាន់នៃកម្មវិធីសិក្សារបស់ដេប៉ាតឺម៉ង់វិទ្យាសាស្ត្រកុំព្យូទ័រនៅវិទ្យស្ថាននិបទិច។ កម្មសិក្សានេះបានចាប់ផ្តើមពីថ្ងៃទី ១៦ ខែកក្កដារហូតដល់ថ្ងៃទី ៥ ខែតុលាឆ្នាំ ២០១៨។ ប្រធានបទនៃកម្មសិក្សាគឺស្មាតអ៊ិនត្រាណែតដែលជាវេបសាយប្រើផ្នែកក្នុងដែលដើរតួជាបណ្តុំកម្មវិធីសម្រាប់ឲ្យនិយោជិកនៅក្រុមហ៊ុនស្មាតអាចចូលបើកមើលកម្មវិធី ប្រព័ន្ធ និងព័ត៌មានទូទៅរបស់ក្រុមហ៊ុន សម្រាប់ការប្រើប្រាស់រាល់ថ្ងៃ នៅកន្លែងធ្វើការ ដោយងាយស្រួលនិងឆាប់រហ័ស។ ប្រព័ន្ធអ៊ិនត្រាណែតដំបូងរបស់ស្មាតមានបញ្ហាធំៗជាច្រើន ដូចជាបទពិសោធន៍របស់អ្នកប្រើប្រាស់ និងការរក្សាកូដ ដូច្នេះគម្រោងនេះគឺជាការកែប្រែថ្មីនៃប្រព័ន្ធចាស់។ ដូច្នេះ ខ្ញុំត្រូវបានចាត់តាំងដោយលោក ហាង សុភ័ក្ត្រ ដើម្បីធ្វើការជាមួយអ្នកហាត់ការខាងអភិវឌ្ឍវេបម្នាក់ទៀត ព្រមទាំងបានផ្តល់នូវតម្រូវការសម្រាប់គម្រោងថ្មីនេះ។ ការទទួលខុសត្រូវចម្បងរបស់ខ្ញុំគឺ ការរចនានិងអភិវឌ្ឍបេកអិនវេប និងទំព័រអ្នករដ្ឋបាលនៃវេបសាយ ហើយខ្ញុំក៏បានជួយផ្នែកខ្លះនៃហ្វ្រុនអិនដែលការការទទួលខុសត្រូវរបស់ដៃគូខ្ញុំដើម្បីពន្លឿនការអភិវឌ្ឍគម្រោង។ ក្នុ ងអំឡុងពេលនៃគម្រោងនេះ ខ្ញុំបានប្រើប្រាស់ថាមពលពេញលេញរបស់ឡាវ៉ាវេល ដោយបានធ្វើការជ្រមុជយ៉ាងជ្រៅទៅក្នុងការអភិវឌ្ឍបេកអិនអេភីអាយ ព្រមទាំងបានអភិវឌ្ឍយ៉ាងសម្បើមទៅលើជំនាញដែលខ្ញុំខ្វះខាតពីមុនដូចជាឡាវ៉ាវេល ប៊ូតស្ត្រេប ចាវ៉ាស្ក្រីប និង ចេក្វឺរី និងក្រេបយកបច្ចេកវិទ្យាថ្មីៗដូចជាអាសឺសជាដើម។ លើសពីនេះទៅទៀតខ្ញុំមានបទពិសោធន៍ឃើញដោយផ្ទាល់លើវិធីសាស្ត្រស្តាំពិតៗដែលត្រូវបានអនុវត្តចំពោះការងារអភិវឌ្ឍវេប ហើយបានទទួលនូវជំនាញទន់ជាច្រើនដែលមានសារៈសំខាន់ចំពោះការរីកចម្រើននៃជំនាញវិជ្ជាជីវៈរបស់ខ្ញុំ។ ពិតណាស់ខ្ញុំប្រឈមនឹងឧបសគ្គមួយចំនួនក្នុងអំឡុងពេលកម្មសិក្សាដូចជាសម្ពាធខ្លាំង និងកង្វះចំណេះដឹងផ្នែកបច្ចេកទេស ប៉ុន្តែខ្ញុំបានយកឈ្នះឧបសគ្គទាំងនេះដោយជោគជ័យតាមការខិតប្រឹងប្រែងស្វែងរកដំណោះស្រាយប្រកបដោយភាពច្នៃប្រឌិត ហើយរៀនអ្វីថ្មីៗនិងសម្របខ្លួនខ្ញុំទៅនឹងបរិយាកាសការងារនិងការប្រាស្រ័យទាក់ទងប្រកបដោយភាពចម្រុះ។ នៅចុងបញ្ចប់នៃកម្មសិក្សា ខ្ញុំបានធ្វើចប់លើរាល់ការទទួលខុសត្រូវដែលបានចាត់តាំងឲ្យខ្ញុំ ហើយខ្ញុំសង្ឃឹមថាវេបសាយអ៊ិនត្រាណែតថ្មីនេះនឹងបង្កភាពងាយស្រួល និងបង្កើនផលិតភាពការងារដល់និយោជិកស្មាតក្នុងការប្រើប្រាស់ក្នុងការងាររាល់ថ្ងៃ។

Abstract

This report describes my work during the three-month internship at Smart Axiata Co., LTD as a web application developer. This internship is a crucial part of the Bachelor of Computer Science's curriculum at NIPTICT. The internship started form 16th-July to 5th-October 2018. The topic of the internship is Smart Intranet, an internal website that serves as an all-in-one portal for employees at Smart to access necessary company tools, systems and information for everyday use at work in a quick and convenient way. The first-version Smart Intranet had so many major problems notably user experience and code maintainability so this project is a revamp of the old one. So, I was assigned by Mr. HANG Sopheak to work with another web intern and provided with requirements for the new project. My main responsibility was to design and develop the backend and admin page of the website and I also helped involve with some of the frontend which is responsible by my partner to speed up the project development. Over the course of this project, I managed to harness the full power of Laravel by diving deeper into the API backend development, improve exponentially on the frontend skills I previously lacked in such as Laravel, Bootstrap, Javascript and JQuery and grab new technologies such as Axios. On top of that, I experienced realistic Scrum Methodology being applied to development work and gained many insightful soft skills which are vital to my professional growth. Indeed, I faced some challenges during the internship such as pressure and lack of technical knowledge but I successfully overcame them by consistently working hard to find creative solution and learn new things and adapting myself to the work environment and diverse communication. By the end of the internship program, I have completed all my assigned responsibility and I hope that the new Smart Intranet website will provide convenience and increase productivity in everyday work for all Smart employees.

ABBREVIATIONS AND SYMBOLS

HTML	:	Hyper Text Markup Language
CSS	:	Cascading Style Sheets
PHP	:	PHP: Hypertext Preprocessor
API	:	Application Program Interface
JSON	:	JavaScript Object Notation
HRIS	:	Human Resource Information System
NIPTICT	:	National Institute of Post, Telecoms & ICT
SQL	:	Standardized Query Language
GUI	:	Graphical User Interface
UX/UI	:	User Experience/User Interface
ORM:	:	Object Relational Mapping

TABLE OF CONTENTS

Table of Figures

I. INTRODUCTION

1. Presentation of the Internship

1.1. Objective

Having studied computer science at NIPTICT for three years, students are supposed to have gained a lot of practical skills and experience which are applicable for real-world enterprise work. Thus, NIPTICT requires the year-three students to utilize what they have learnt and conduct a three-month internship at a standardized and professional company. During this period, the goal is for them to realistically apply their talent and unleash their potential to contribute towards actual company project as well as grasp professional manners such as communication and work ethics and then get ready for the competitive world. This internship also essentially serves as a bridge to their senior year because after the completion of the internship they are required to hand in their thesis and defense regarding their internship in order to officially pass year three and get through to the final year.

1.2. Duration

The internship at Smart Axiata started from 16th July 2018 and ended on 5th October 2018, so the duration is approximately 3 months which suits the internship's requirement.

2. Presentation of the Organization

2.1. Background

After the successful acquisition of StarCell and merger with Hello Axiata, Smart is now part of Axiata Group Berhad, one of the largest telecommunications groups in Asia. The company's workforce consists of more than 1000 local and foreign experts. Smart is committed to its customers, employees and the people of Cambodia, in delivering its promise of enriching their lives through world-class networks, exceptional digital experiences and through significant corporate social responsibility engagements. Smart

aspires to become Cambodia's Digital Champion, while playing an active role in socio-economic growth.

2.2. Services and Activities

Figure 1: Smart Axiata's Logo

Smart Axiata Co., Ltd., Cambodia's leading mobile telecommunications operator, currently serves 8 million subscribers under the 'Smart' brand. Smart is at the forefront of mobile technology advancement in Cambodia. Smart was the first network to introduce 4G LTE in 2014, 4G+ in 2016 and 4G+ with HD Voice (VoLTE) in early 2017. In mid-2017, Smart introduced cutting-edge 4.5G, manifesting its data leadership position in Cambodia. Smart also provides 2G, 2.5G, 3G and 3.75G mobile services as well as international roaming across more than 190 countries. Its extensive nationwide network coverage stretches to more than 98% of the Cambodian population.

The company is also rapidly transforming itself into a digital lifestyle brand, having introduced many innovative offerings and lifestyle entertainment value propositions. This includes various international partnerships, with brands as diverse as Universal Music, Apple, Facebook and iflix, as well as digital services including SmartLuy, Smart Insurance, SmartPay, Smart Music and SmartNas.

Smart sees itself also as an integral part of the society with a mission far beyond pure business. While Smart's core mission is connecting people which by its own generates tremendous benefits for the society and economy of Cambodia, Smart has defined Education, Sport and Green Environment as its fields of engagement. Smart collaborates with UNESCO and Ministry of Education to counter illiteracy, helping 92,000 Cambodians across the nation to read and write during 2015.

2.3. Address and Contact

Smart Axiata is reachable through the following means:

- ❖ Address : No. 464A Monivong Blvd, Sangkat Tonle Bassac, Khan Chamkarmorn, Phnom Penh, Cambodia
- ❖ Email : info@smart.com.kh / 888@smart.com.kh
- ❖ Telephone : 888 / (+855) 10 200 888
- ❖ Website : www.smart.com.kh

2.4. Organization Chart

As Smart Axiata is a big company, only relevant structure related to my department, DevOps, will be illustrated and some specific information will be blurred due to the nature of company confidentiality.

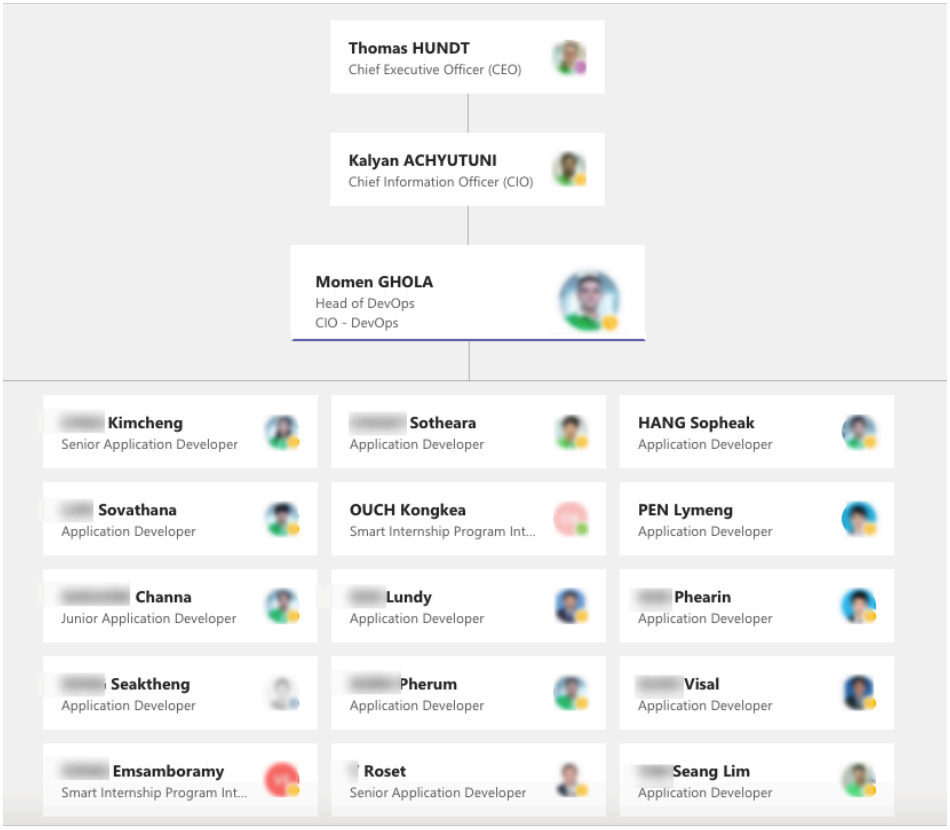

Figure 2: Organization Chart

II. PRESENTATION OF THE PROJECT

1. General Presentation of the Project

The title of the project is “Smart Intranet” which is one of the projects I involved in during the 3-month internship. It is a complete redesign and rewrite of the first Smart Intranet, an internal Smart’s website that acts as a portal for employees at Smart to access essential company tools, systems and information for daily use at work in a quick and convenient manner. Smart Intranet is targeted for company desktops and laptops, so it will only have web platform.

2. Problematic

The original Smart Intranet website is practical but lack in user experience, code maintainability and lack of admin panel.

For user experience, the majority of Smart’s employees complain that the website is not quite user-friendly and poorly designed. It takes them so many steps to access what they want and sometimes they can’t find the desired feature.

In order to manage the content of the website, direct SQL and GUI access of the database must be performed because there is no admin panel page and this is quite complex and inefficient.

On the other hand, about code maintainability, the original website’s backend was built using ASP.NET by previous developers and the new developers are not knowledgeable in this technology plus the company prefers new technology that is easy to maintain for new developers.

3. Objective

To address these problems, Smart's application department decides to create a new Smart Intranet website from scratch to replace the old one. The purpose is to provide a more user-friendly version with same functionality as the old one but with more intuitive interface for employees, a maintainable codebase for newer developers and a management panel for administrator as the original one doesn't have.

4. Development Methodology

To effectively initiate a project, it is essential to carefully choose a well-suited and effective type of methodology. Hence, Scrum methodology was specifically selected due to the nature of its flexibility and ease in adoption method as it supports a great variety of different types of project.

Agile Software Development with Scrum is usually considered as a methodology. In such methodology, the project manager plays the role of the Scrum Master. Each scrum lasts the period of one week and we carry out a small meeting about the process of each developer for approximately 10 minutes every morning.

Each task was assigned by project manager and was split into sprint in total that one sprint took about 2 weeks. We always have our stand-up meeting every morning around 5-10 minutes to discuss about our previous tasks, challenges, solutions, struggles and the planning for upcoming tasks. The following table demonstrates my plan during the 6 weeks of Smart Intranet project.

Activity/Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
Study requirement						
Study new technology						
DB design and analysis						
Implementation						
Testing						

Figure 3: Project Schedule

5. Task breakdown

The table below shows the tasks that we have divided, which is specifically the work breakdown structure in our project development. There are two people in the development team which is me and the other developer whose name cannot be disclosed, so the project manager split the tasks accordingly based on our expertise.

Task	Role
User Authentication	Ra My
Integration of Yammer and HRIS	
Core Values	
Partners	
Guidelines	
Forms	Kongkea OUCH
Policies	
Manage Partners	
Manage Policies	
Manage Forms	
Manage Guidelines	

Figure 4: Task breakdown

III. ANALYSIS AND GENERAL CONCEPT

1. Requirements

To get a better picture of the objective of the project, we need to list out and briefly elaborate each of all the requirements of the project.

1.1. Functional Requirements

The functional requirements are the core functionalities needed to be implemented in order to make the web application works. The details of these functionalities are as following:

- **User Authentication**

  In order to access Smart Intranet, every user must first log in with the correct email and password which are uniquely provided for every employee. If the credential entered is correct, then the user will be logged into Smart Intranet website as well as HRIS and Yammer which are integrated into the Smart Intranet website. Upon logout of Smart Intranet, the user will be signed out of HRIS and Yammer as well.

- **Middleware Security**

  Normal type user will be directed to Smart Intranet home if they try to click back to log in or type out a site route that doesn't exist and not accessible to them. Admin type user will have access to the admin page while normal type user can only browse typical features.

- **Integration of Yammer and HRIS**

  Yammer is a social networking site for corporation and Smart uses it for internal social communication. HRIS, human resource information system, is Smart internal website that allows employees to check and manage details of their HR information. Making these two sites automatically accessible and combined into Smart Intranet is convenient for users as they can save time without logging in individual website of Yammer and HRIS and can even use several other Intranet features at the same

time. Hence, they are linked and automatically logged in when user logs in the Intranet and ready for use at a fingertip.

- **Core Values**

  Smart has strong core values and they are incorporated into the core values page of Smart Intranet, making it convenient for employees to access for inspiration at any time. There are five main core values since the foundation and they are likely to remain for a long time, so this page is static as required.

- **Guidelines**

  This page is important for new employees who need to understand about the company guidelines. All guidelines are shown as grid style in a single page and each has a button for download or view of respective document. User shall be able to do live search of guideline name and sort by category of department smoothly without having to refresh the page.

- **Policies**

  Every employee must abide by and understand Smart policies in order to avoid breaking any rule and causes consequence to oneself or the company. This is why it is highly required for a page with all company policies to be included in Intranet. Same as guidelines, this page contains all policies of Smart and they are all downloadable as well as viewable in document form. Plus, user can perform real-time search and sort by department in a convenient manner. This feature page has the same interface as guidelines.

- **Forms**

  In case any employee needs to request for a specific purpose, the protocol of the company is to submit a form in hard copy. Hence, all forms are stored and accessible for view/download on Intranet as well as searchable and sortable without having to reload. This feature page has the same interface as guidelines.

- **Partners**

  Smart has a lot of partners with tons of discounts or promotions. Hence, all partners' information is also accessible on Intranet. Again, user can search by keyword title and sort by partner's name in an immediate fashion. There will be a table that shows all discount of each partner.

- **Manage Guidelines**

  This feature is accessible for admin type user only. Here, admin has the power to perform CRUD operation on guideline items. All guidelines are shown in table style on a single page and they are also can be searched and sorted real time in order to make it easy for admin to locate specific guideline. Admin can also view path of document and download/preview document of any desired guideline. Then, admin can create new, edit and even do multiple delete of any guideline. After any CRUD task, the items on the page will be updated without having to reload the whole page.

- **Manage Forms**

  Only admin can access this and, just like manage guidelines, the admin can do any CRUD operation onto the form in real time with no need for full page reload. In order to locate form, admin can do live search and sort by department as it is rather quicker than browsing for specific one. This feature's page has the same table interface as manage guidelines.

- **Manage Partners**

  Admin performs CRUD of partners on a modal dialog of the same page without having entering new page. On add/edit dialog, admin can add/edit info and logo of partner and then add/edit multiple discounts or promotions.

- **Manage Policies**

  Just like aforementioned management features, admin has the ability to do CRUD on policies as well with the same style as manage guidelines. Real-time

functionality is also used in this feature to ensure smooth convenience and efficiency.

1.2. Non-functional Requirements

Apart from the functional requirements, it is crucial to consider as well about the non-functional requirements as it would contribute largely to the enhancement of the application quality and usability of our system. The non-functional requirements include:

- **UX/UI**

  The original intranet has severe flaw in UX/UI as reported by so many users. Hence, the new intranet's UX/UI must be simple, intuitive and user-friendly so that users can navigate on our system smoothly and painlessly.

- **Performance**

  All steps in developing the new intranet must always give regard to the impact of performance because It is vital to make our system run quickly and efficiently without any errors.

- **Maintainability**

  Due to the fact that the old intranet suffers in maintainability as a result of the usage of an unfamiliar code for new developers, the Smart Intranet must be designed and programmed in a way that is easy and hassle-free for the next developers to maintain.

- **Scalability**

  The system architecture must be carefully designed in regard to the scalable nature so that new changes or adjustments can be performed in the future seamlessly and effectively.

2. Analysis of the Project

2.1. System Users

In this application, there are only 2 types of users which are:

- **Administrator:** can perform all operations including viewing HRIS, Yammer, Core Values, Guidelines, Policies, Forms and Partners, and managing these features as well.
- **Staff:** can only has view and browse HRIS, Yammer, Core Values, Guidelines, Policies, Forms and Partners.

2.2. Use Case Diagram

- **Staff**

  As described in section 2.1, staff user can only access HRIS and Yammer and browse core features but staff is required to log in first to perform these taks

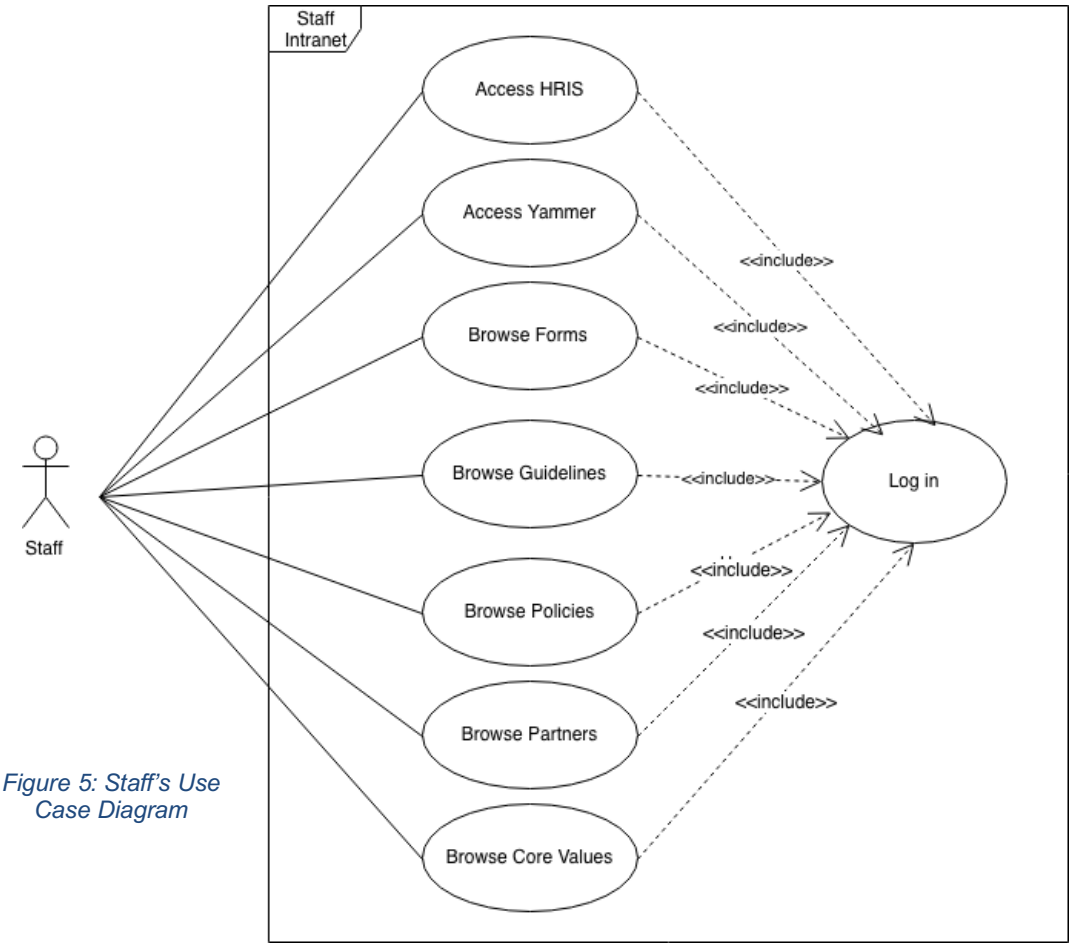

Figure 5: Staff's Use Case Diagram

- **Admin**

  As demonstrated in section 2.1, admin user has the power to access CRUD features on top of the features accessible by staff but also needs to log in with admin credential.

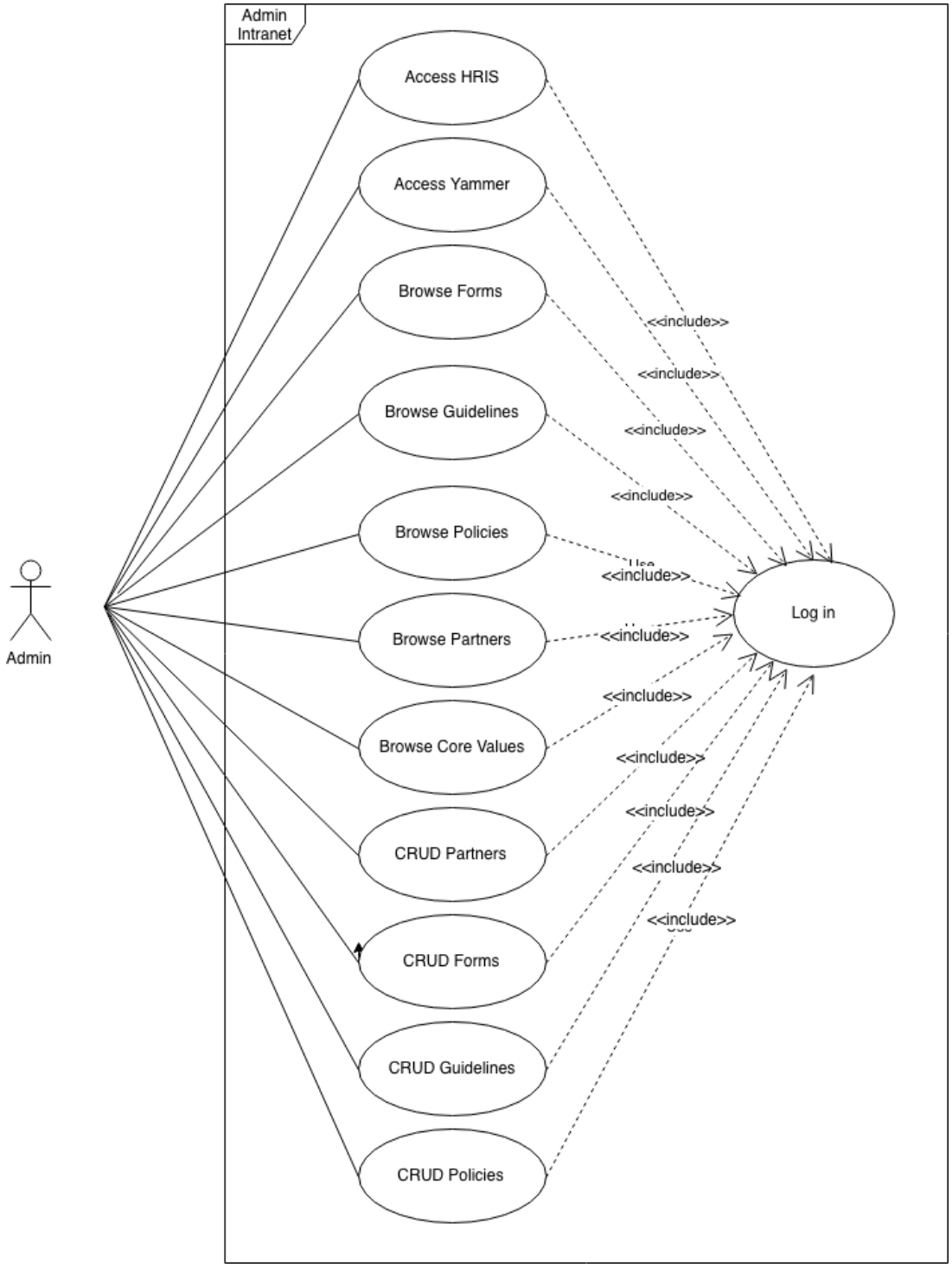

Figure 6: Admin's Use Case Diagram

2.3. Activity Diagram

For activity diagram, there will be one each for the flow for the two type of users which are administrator and for the important sub-flows. These activity diagrams will take into consideration the requirements of section 2.1.

- **Admin's Flow**

  In this flow, admin can either choose to access staff's features or admin's features and switch back and forth. If any info or CRUD feature is selected, it will call to the server to return the necessary data.

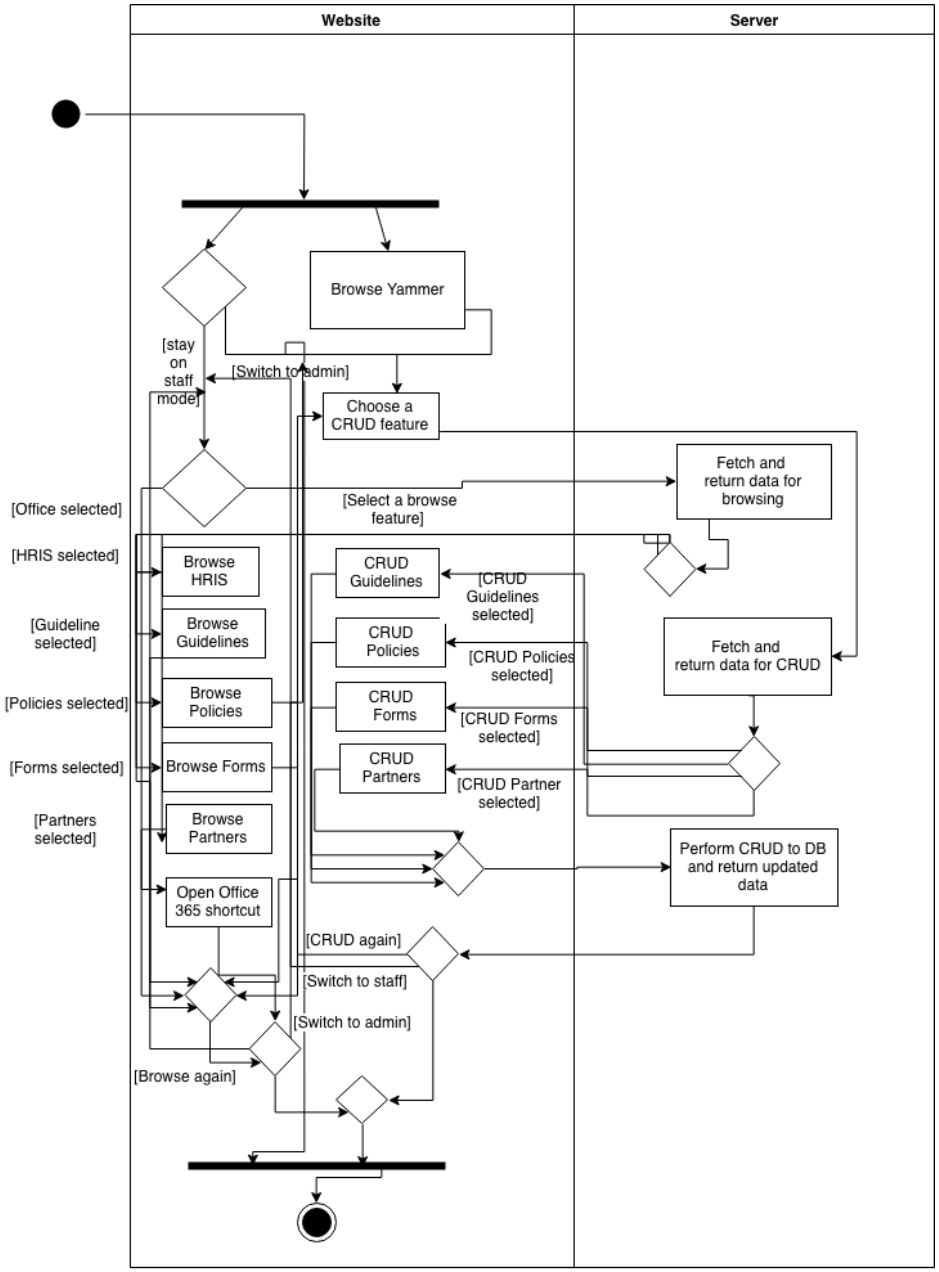

Figure 7: Admin's Flow Activity Diagram

- **Staff's Flow**

  Similar to the admin's flow minus the CRUD features, staff can only access the information features such as HRIS, Core Values, Guidelines, Forms, Partners and Policies. On top of that, Yammer loads once and stay afloat on the right sidebar and can be browsed at all time. Whenever the info feature that has data stored on server is selected, it will call to the server so that it can fetch and return the required data back for use on frontend.

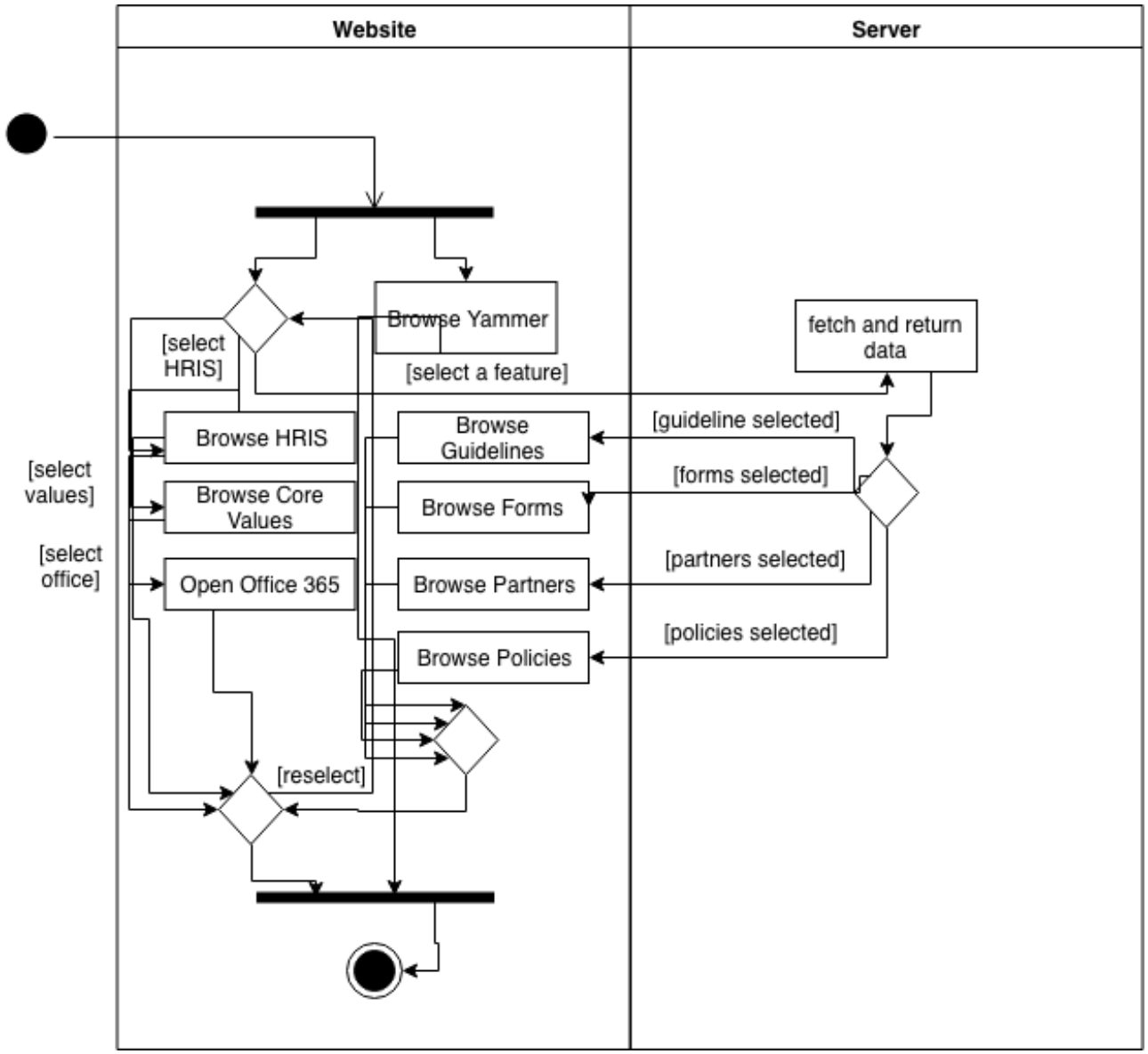

Figure 8: Staff's Flow Activity Diagram

- **Edit Flow**

  This flow is a crucial part of all CRUD features and acts as a blueprint in CRUD implementation. In order to successfully edit an item, admin has to make change to any info field and change new file. If all conditions are met, the form submit will carry out request to server to update and save new file to the server. The page will update automatically without full reload if update is successful.

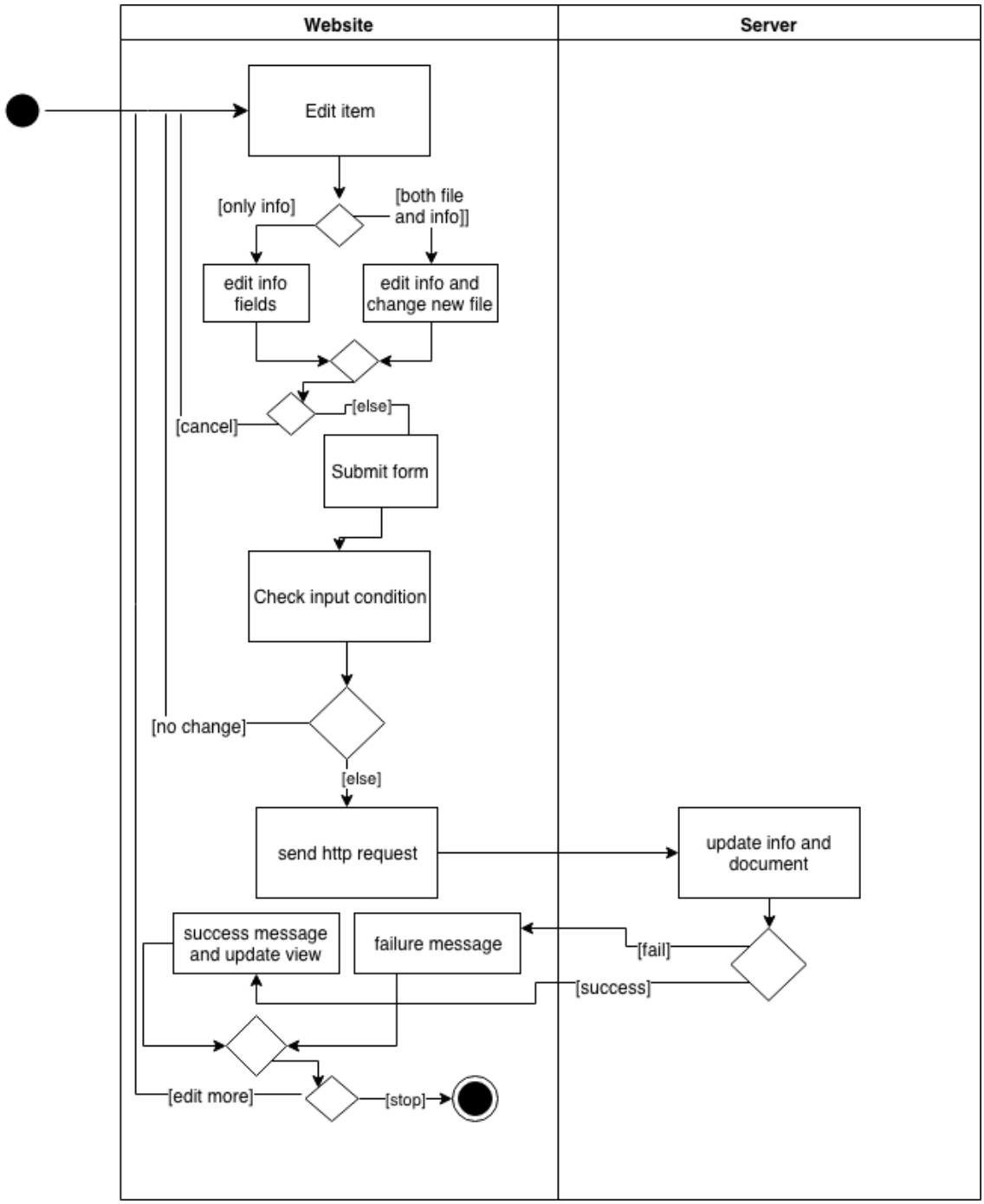

Figure 9: Edit Flow Activity Diagram

- **Create Flow**

  The create flow is also an essential part of CRUD features and acts as a blueprint for CRUD implementation. To create successfully, admin needs to input all form fields and choose file. Then, form submission shall be carried out followed by condition check and if it's met a request will be sent to server to perform the creation of the item.

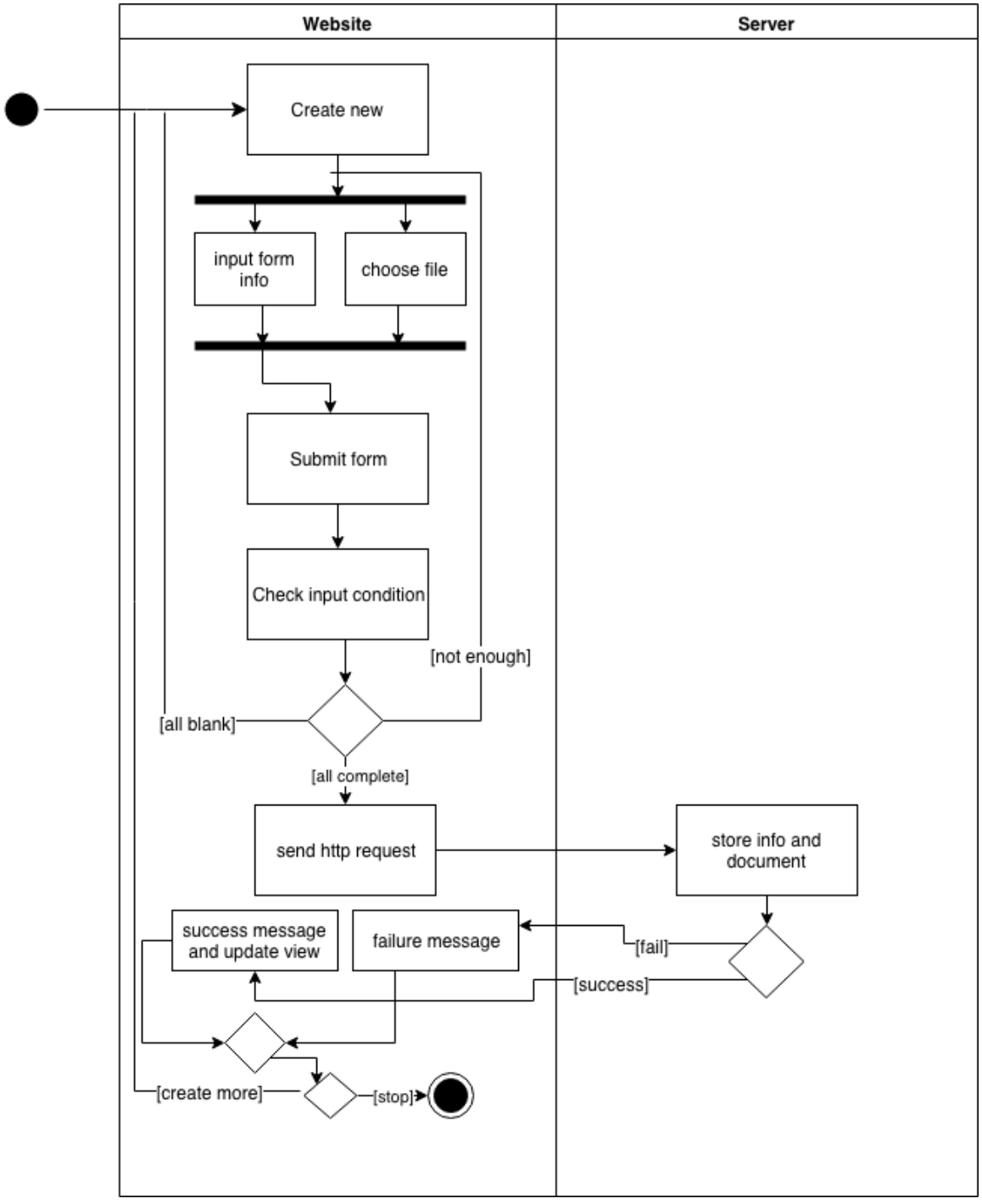

Figure 10: Create Flow Activity Diagram

- **Delete Flow**

  The delete flow is the backbone of deleting in any CRUD features. Admin can perform single delete or multiple delete but multiple delete requires select on at least one item. If the delete decision is confirmed, it shall request the server to perform deletion of corresponding info and file(s).

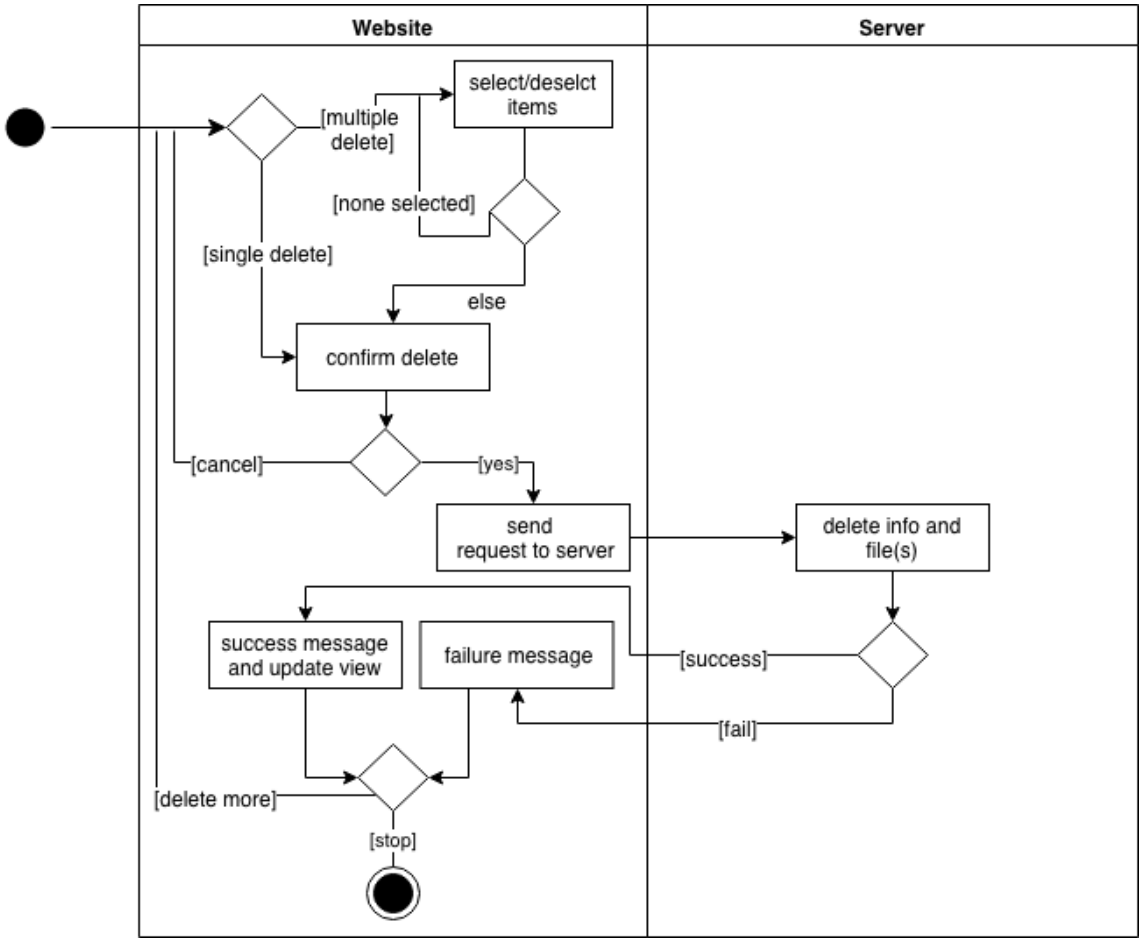

Figure 11: Delete Flow Activity Diagram

2.4. Database Design

For the analysis and design of the database, I choose to draw class diagram first which closely resembles the enhanced ER Diagram in order to represent the requirements and complexities of the databases. This class diagram will describe the structure of the intranet system by showing the system's classes, their attributes, methods and the relationships among objects as reflected by the requirements.

In abstract explanation, staff inherits all attributes and methods from user but the methods are limited unlike admin who is extended on top of staff and can also perform management tasks. Admin can perform CRUD on many items of the below features and staff can only browse them. The function of Core Values function is not given a place in database as it is static as required. Same goes for HRIS, Yammer and Office 365 as they are just links to other websites/apps so no structure required. One department contains many feature items and partner items are part of a bigger group.

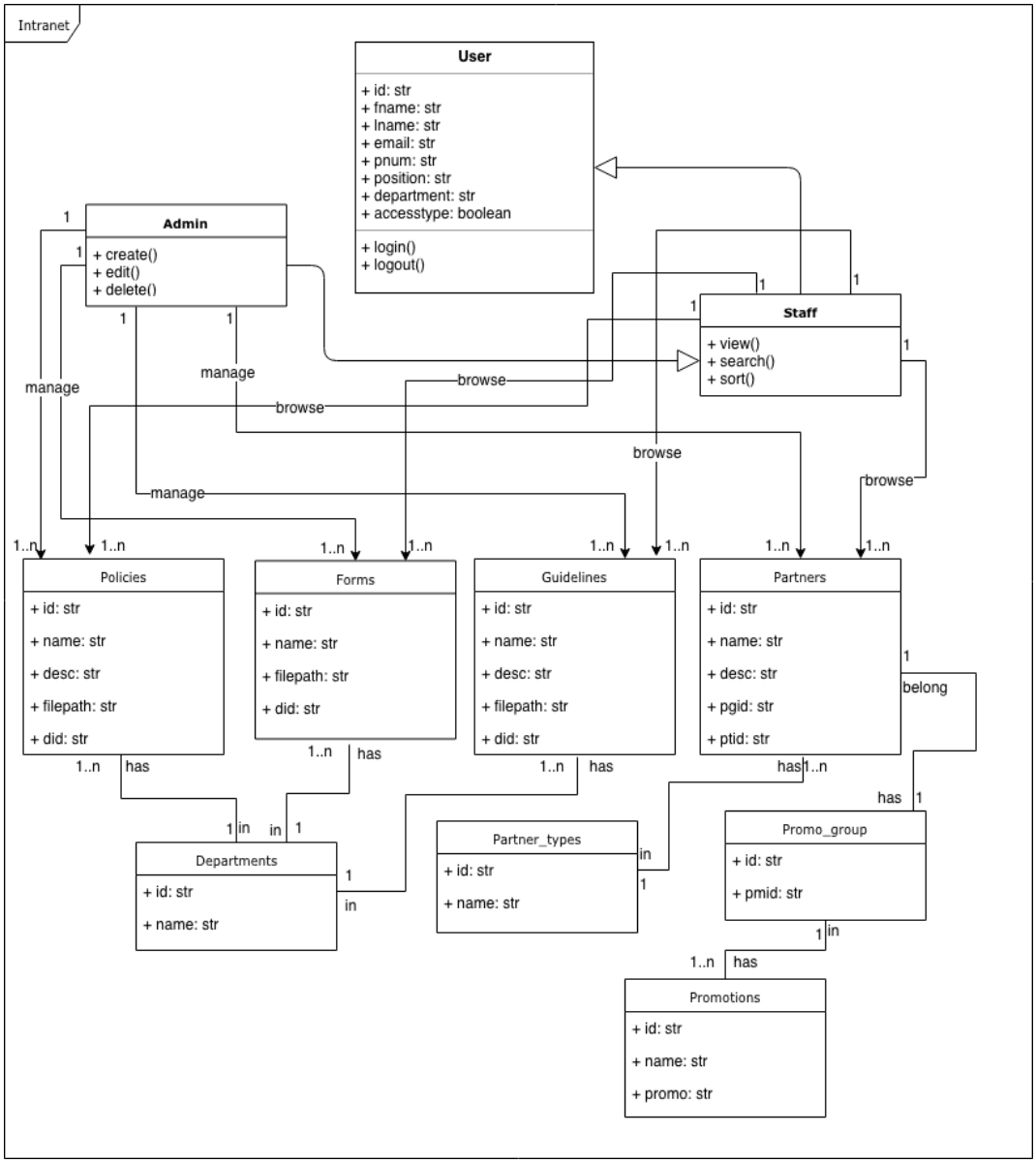

Figure 12: Intranet Class Diagram

IV. DETAIL CONCEPT

1. Choice of Technology

First and foremost, a software product must meet all the requirements of the customer or end-user. Also, the cost of developing and maintaining the software should be low. The development of software should be completed in the specified time-frame.

1.1. Web Framework

Figure 13: Laravel Logo

Hence, Laravel is our solution for creating Smart Intranet application with beautiful code and excellent maintainability. Laravel is an open-source PHP framework, which is robust and easy to understand. It follows a model-view-controller design pattern. Laravel reuses the existing components of different frameworks which helps in creating a web application. The web application thus designed is more structured and pragmatic.

1.2. Languages and Libraries

In order to achieve effective outcome, several languages and libraries were used as following:

Figure 14: HTML Logo

- **HTML:** Hypertext Markup Language, is a standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on World Wide Web pages.

Figure 15: CSS Logo

- **CSS**: is a simple design language intended to simplify the process of making web page presentable by define how to display HTML elements.

Figure 16: JavaScript Logo

- **JavaScript**: is a scripting language designed to add interactivity to HTML pages and create dynamic web.

Figure 17: jQuery Logo

- **jQuery:** is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.

Figure 19: Bootstrap Logo

- **Bootstrap:** is a sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development that uses HTML, CSS and Javascript.

Figure 18: AdminLTE Logo

- **AdminLTE:** beset open source admin dashboard & control panel theme which provides a range of responsive, resusable, and commonly used components.

Figure 21: PHP Logo

- **PHP:** is a server scripting language, and a powerfull tool for making dynamic and interactive web pages.

Figure 20: SweetAlert2 Logo

- **SweetAlert2:** is a beautiful, responsive, customizable and accessible (WAI-ARIA) replacement for JavaScript's popup boxes.

1.3. Tools

The following tools help me in the project development in an effective and smooth manner:

Figure 22: VS Code Logo

- **Visual Studio Code:** is a code editor redefined and optimized for building and debugging modern web apps and cloud applications.

Figure 23: HeidiSQL Logo

- **HeidiSQL:** is a useful and reliable tool designed for web developers using the popular MySQL server, Microsoft SQL databases and PostgreSQL.

Figure 24: Postman Logo

- **Postman:** is the essential toolchain for API developers to share, teset, document and monitor APIs.

Figure 25: Gitlab Logo

- **Gitlab:** is a web-based repository manager that lets teams collaborate on code, duplicate code to safely create and edit new projects, then merge finished code into existing projects.

Figure 26: Chrome Logo

- **Google Chrome:** is a powerful web browser that gives amazing browsing speed and contains concrete developer tool is great for debugging code.

2. Architecture of the Application

2.1. Physical Architecture

The physical architecture of the web application below illustrates the general concept of the web application that explains how the web application functions and the physical components of the web application as below.

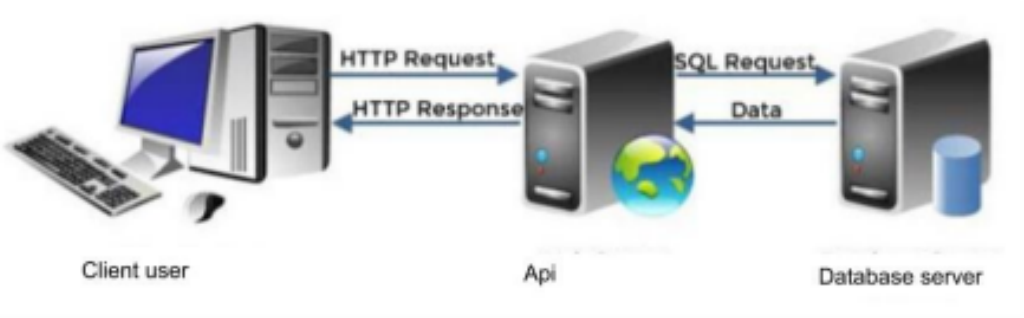

Figure 27: Physical Architecture of Web Application

As shown above, there are 3 major components which are:

- Client browser: represents the client web browser
- API: represents the server
- Database server: represents the data storage server

The web application is an application that is stored in the location of a web server and the client can access it by using the web browser. Upon the web server receives the request, it will communicate with the database server to fetch data that the client desires. Then, the web server will return the requested data back so that the client browser can interprets those data and show to the user.

2.2. Logical Architecture

As we use Laravel as the framework to build this project, the following will elaborate about its MVC architecture:

- Model: represents your data structures and normally contains functions that interact with your database to retrieve your objects' information.
- View: is basically the information that is presented to user after being manipulated by the controller.

- Controller: act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output.

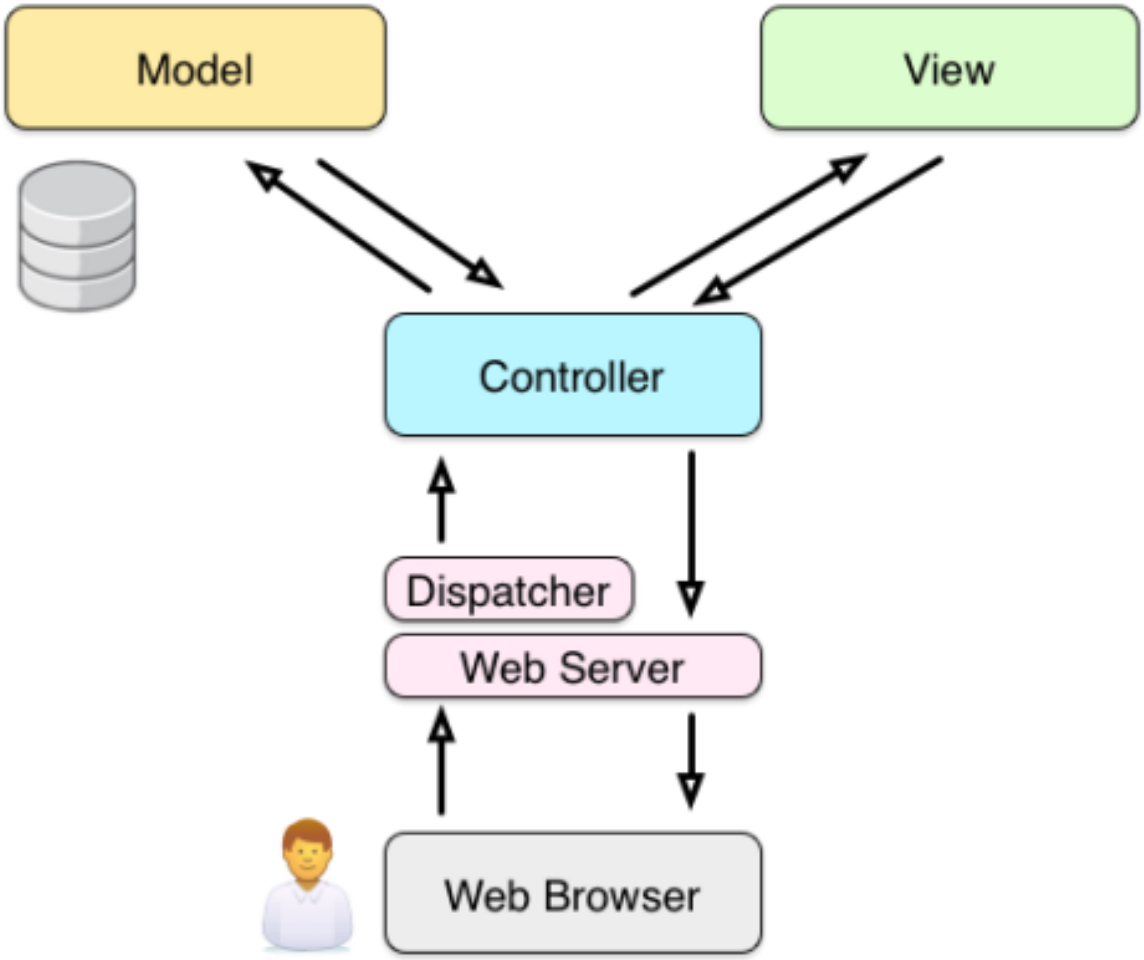

Figure 28: Laravel Architecture

V. IMPLEMENTATION

In this section, I will thoroughly explain how the project is implemented. However, since there are various functions in this project, I will only cover the important points. Notice that due to the nature of confidentiality of my company, I cannot highlight any code or exact screenshot of the project to explain so I will do my best to explain by relating to the flows of the project requirements and diagrams in previous sections as I have mostly already explained how they work already.

1. Structure of the Application

The application structure of my project is basically the structure of folders, sub-folders and files included in a Laravel project. Once I create a Laravel project in VS Code, I get an overview of the application structure and it includes various sub-folders and files, hence the description of folders and files, along with their functional aspects is given below.

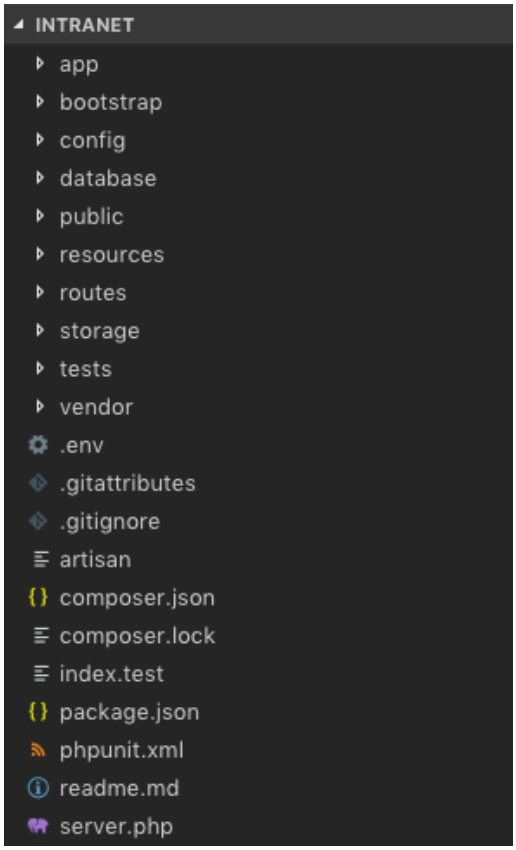

Figure 29: Structure of the Application

- **App:** is the main part of the application, all the major parts of the code we develop resides in this app folder. It contains Controller files, Model files, Routing file, Request files, handler files and many more, the meat of the application lives here in the app directory.
- **Bootstrap:** This folder encloses all the application bootstrap scripts. It contains a sub-folder namely cache, which includes all the files associated for caching a web application. You can also find the file **app.php**, which initializes the scripts necessary for bootstrap.
- **Config:** includes various configurations and associated parameters required for the smooth functioning of a Laravel application.
- **Database:** includes various parameters for database functionalities which are seeds that contains the classes used for unit testing database, migrations that helps in queries for migrating the database used in the web application and factories that is used to generate large number of data records.
- **Public:** is the root folder which helps in initializing the Laravel application and includes .htaccess that gives the server configuration, JavaScript and CSS which are assets and index.php that is requires for the initialization of the web application.
- **Resources:** contains the files which enhances the web application which are assets that include files such as LESS and SCSS that are required for styling the web application, lang contains configuration for localization or internationalization and view that are the HTML files or templates which interact with end users and play a primary role in MVC architecture.
- **Storage:** stores all the logs and necessary files which are needed frequently when Laravel is running and includes app that contains files that are called in succession, framework that contains sessions, cache and views which are called frequently and lastly logs that store all exceptions and error logs.
- **Tests:** contains all the unit test cases.
- **Vendor:** includes all the composer dependencies.

2. Web Application Configuration

- **Updating dependencies**

Laravel uses Composer to manage its dependencies. So, before running the project, I have to update Composer to the latest version in order to maintain good performance and avoid any potential error.

- **Config the database**

On top of that, since my project is a dynamic web application, it is essential that it is correctly connected with the database that resembles Figure 12. Hence, I have to config database by accessing the .env file and enter the right database credentials. Here, in this file, I can also config other things such as application name and especially making sure that the app key exists or else I must generate it with command.

- **Seeding the database**

In case there's mistake in losing data and I need to rebuild the database, I generate seeder classes for all tables in the database to account for the data there so that I can rollback and re-run all of my migrations thus getting my data back.

- **Associating tables with models**

Each database table has a corresponding "Model" which is used to interact with that table and the relationship between database tables are also depicted in model. This is extremely useful for retrieving and performing CRUD features as it removes the hassle of writing long query.

- **Route prefixing**

Referring to figure 5 and 6 which are essentially about the use case of admin and staff, we see that there are several different pages to access. So, this means we have to create separate routes for all those pages with prefix accordingly to the user type and actions too.

3. Feature Implementation

3.1. User Authentication

Referring to figure 5 and 6, both users need to login to access the inside features. In implementation, first of all, the user visits the homepage and clicks login and will be prompted a dialog to login. Then, they have to provide email and password but the email has to be valid email so that they can click submit login. Here, Axios sends http request to the company API and while waiting for response the screen has spinning wheel overlay and user can't do anything.

Upon response, the spin will stop and the result can go two ways. If the server confirms the credential, it will return JSON info of user such as name, phone number, department, position and access type whether the user is admin or staff and start login session as well as automatically connected to the user's HRIS and Yammer by API request.

If the server can't find any existed credential, it will immediately send failure code back which will be read by Axios's post-condition and then it will flash sweet alert dialog that the email or the password don't match. Back to the server confirming the credential, if the user has admin access, after login success, the page will shift to the intranet normal home interface that has a button to switch to admin mode. But, if it's staff user, the switch to admin option won't show and the staff user can only access normal intranet home.

3.2. Core Values

This is a static page that contains Smart's five main core values. Each is can be toggled to show or hide its detail as it uses the bootstrap button's functionality.

3.3. Forms, Guidelines, Policies and Partners

The reason that I group these features together to explain is because each of them is a page that follows the same blueprint of figure 8. They are accessible by both staff and admin user and can be selected through the left side bar menu or the main interface.

As depicted in figure 8, when user selects one of these pages, it will be routed to controller that returns the corresponding view with data. The data then is displayed to the user in the

form of bootstrap grid box which each content has title and description and view/download document button. There is actually only one code module with blade template-based condition to dynamically replicate more boxes based on the total of elements returned by the data. Each module also stores the document path in the form of html data attribute which is used to dynamically load each document when the user clicks view/download. Here, when the view/download button is clicked, it will get the document path from the data attribute and use it to fetch file from database and preview it on a bootstrap modal if it's pdf file or else it will be automatically downloaded.

There is a bootstrap search box and category selection on the top of page which can be used to perform live search and categorize of the loaded elements. Both search and categorize functions are written using jQuery and both functions are reused everywhere. For the search function, on key pressed, it will start iterating the elements to compare each of search input key with the title and description of each element. If there is any match, the content will be kept on display, but if none of title and description match with the search input, the element will be hidden. It also takes into account the currently selected category by checking another condition that compares the hidden category attribute of each element with the selected category. If the match search doesn't belong to the chosen category, it will not show either. Same goes for the categorize function, when the user select any dropdown category, it will first check with the search input to see if the elements in the selected category has any matched title or description. If the condition is not satiated, it will be kept hidden. Overall, search and categorize work together dynamically and dependently to create a smooth and real-time experience for user.

3.4. Management of Forms, Guidelines, Policies and Partners

Again, similar to section 3.3, the management features also follow a uniformed code blueprint that incorporates the create, edit and delete's flow as illustrated in figure 9, 10 and 11 respectively. These features are only accessible to admin user by switching to admin mode on the intranet home. And if staff users try to access by manually typing any of the management-related route, the middleware rule will redirect them to homepage.

As illustrates in figure 7, when a management page is selected, it will retrieve corresponding data from the server and return to view in the form of bootstrap table. Similar

to the grid box generation in section 3.3, there is a blade template logic in the view to iterate and multiply the table row in accordance with the total of data returned. Then, each row also has title, category and file path thank to the dynamic blade statement of variable. Each row also has hidden id in its data attribute which is useful for CRUD to locate the right item. There is also checkbox next to each row and a select all checkbox as well that will tick every checkbox, which is useful for multiple delete. Moreover, each row has action column which contain buttons for edit, single delete and view/download document.

Focusing on CRUD side, I refer to the figure 9, 10 and 11 as foundation for implementation here. For creating new, as shown in figure 10, there is a "add new" button on the management page and when it is clicked a form dialog will pop up and requires to fill and choose file. Only pdf and Microsoft office format files are allowed by including the format in the accept attribute of the input file. If all are complete and the admin presses submit, a create request is sent through the api route to the controller and then the saving process will be performed. All info like title and description will be stored to database using Eloquent model and method and the file will be manually saved to storage as well with the original name and id attached as new name by using the storeAs method. If the creation succeeds, it will response code 200 that will be read by the post condition of Axios. Success sweetalert pop up will be flashed if it receives success code or else it will flash failure message. As for the edit and delete, the sweetalert success and failure message pop up condition also applies. Now onto edit, when clicked, it gets its id and information from the data attributes that were created when the page loads data. Then, a modal pops up with all the available form fields and the fetched information will fill those fields which are ready for edit. Admin shall change any info and click submit. A condition will check if there is change in information and, if there isn't, the modal will close without sending any request. But if there is change, then Axios will be used to send the edited information to the API to perform update operation. The controller will retrieve the sent data and save them to the database using eloquent method. If there is file, the method storeAs will be utilized to upload and store on the Laravel file storage and its path is saved to database along other info. After the update operation, the controller shall return a success response back. Axios will check if the response is success, hence, success or failure message will pop up depending on the response. Moreover, for the delete function, on click, it will fetch id from

data attribute and pop up dialog to confirm delete. Admin can pick no and the dialog is close and nothing happens or select yes so that Axios will send delete request along the selected ID to API to perform the removal of the desired item. Just like create and edit, it will return success code and Axios will be waiting to check it to see if the pop up message should show success or failure. Last but not least, for the multiple delete function to work, the admin must check at least one of the table rows. If there is none selected and the admin press the multiple delete button, it will show warning that no item is selected. Whether admin check or uncheck item, the item id will be added to or removed from an array variable from a condition with handler attached to checkbox. This is possible thanks to javascript convenient array methods. Empty array means none is selected so if the array is not empty and admin perform the delete, a confirm delete dialog pops up for admin to decide. When the admin clicks yes, Axios will send request with the array to api and the multiple delete controller function will iterate over the array and delete related info from the database one by one using Eloquent method. And if the item associates with file, the controller will get its file path and use storage delete method to remove the file. Notice that after every successful CRUD task, the whole page doesn't have to reload to update data as Axios post-condition will partially reload the part of table rows only and it is quite smooth and effective.

In management feature, it also uses the live search/categorize algorithm for filtering table elements. The search bar and category dropdown situate right on top of the table and stays float when the admin browse the page, making it quick and convenient for admin to find desired table item. Basically, how it works is the same as explained in the live/search/categorize of section 3.3.

VI. CONCLUSION

1. Results

Having tried my best, I manage to complete all of the functions assigned to me and even give a hand to my partner and help out his tasks. Here are the completes and non-completed functionality:

1.1. Completed Functionality

As assigned in the requirements of figure 4, I complete the functions I am responsible for which are:

- Forms
- Policies
- Manage Forms
- Manage Partners
- Manage Policies
- Manage Guidelines

All above are single page applications that user can perform any operations in one place without having to reload or redirect to another page. For forms and policies, user now can find any content immediately using the live search and categorize function and can even preview the file without having to download. Overall, I can safely say it is smoother, cleaner and more user-friendly than the original Intranet that lacks in intuitive design and performance.

1.2. Non-completed Functionality

I have completed my part but the only part that is none done yet is Partners that is responsible by my intern colleague. The partners page has issue as its UI is still buggy and it doesn't perform reliably.

2. Strong Points and Weak Points of the Web Application

Nothing is perfect, so same goes for any web application. It is uniquely designed so it has its strong points and weak points.

2.1. Strong Points

The newly renovated Smart Intranet has several strong points:

- It surpasses the original Smart Intranet's features and performance.
- Super intuitive and user-friendly interface.
- Easy to maintain as it's completely rebuilt from the ground up with Laravel, beautiful and dynamic web framework.
- A platform that contains several single-page applications making it easy and fast for users to access and perform any tasks
- Great scalability as it adopts admin-style interface and can be added with more applications

2.2. Weak Points

Due to time constraint, I only focused on practical code so some code is redundant and not clean which requires to be optimized for reusability and scalability. Moreover, the UI is clean and acceptable already but it can still be improved to look more intuitive.

3. Difficulties

Life is a struggle and full of difficulties, but that only makes us stronger. Throughout my period of the internship, I faced several challenges both technically and personally.

When I started the internship, I have limited knowledge in Laravel and the project is to primary use Laravel, so I got a problem here. I acted immediately by commit myself to an intensive self-training of Laravel through thorough research and practice. It was frustrated at times when the code didn't work like I expected to, but I keep persevering and eventually I managed to pull through achieving tremendous practical knowledge Laravel and making me ready for the real-world application.

Prior to the internship, I never engaged in full-time job from 8 to 6. It was quite exhausting and I have trouble adjusting myself to this new lifestyle. Sometimes I felt like it was too much and stressing and it got better as time passed. When I talked to more people and started to treat my workplace like a home, my work fatigue was gone and I was more passionate and energetic than ever. Thanks to my supportive supervisor and colleagues who were always there for me.

4. Experiences

This internship program is a great kickstart into the professional world of software development to me. I have learnt both practical and professional skill and knowledge. Practically, as I stated in the difficulty, I managed to enhance my Laravel knowledge and on top of that I've learnt more jQuery and applied it fluently in order to manipulate the frontend DOM. I even applied software engineering theories to my project and experienced as well as improved on project management. Professionally, I have learnt the spirit of teamwork and communication as the formula success of software development. I exchanged useful information with my partner and we accounted for our responsibility that was assigned to us. Working at Smart also taught me work ethics and integrity as these will sustain trust and drive our career forward. Overall, the experience and skills I have inherited from this internship will act as a foundation for a successful future as a software developer and entrepreneur.

5. Perspective

Throughout the internship, what interests me the most is the magic of API that facilitates the communication between various software components. I received a lot of help from my supervisor about API as I haven't mastered it yet, so I am determined to practice more and enhance my API skill. Especially, I am looking forward to learn Node.js API as it's quite popular and in great demand in Cambodia. At the same time, I also want to learn Angular and enhance my frontend skill and eventually become a successful full-stack web developer.

6. Conclusion

After finishing my internship, I have gained a lot of practical skill and soft skill from my colleagues and self-development. The advice, the insight and the knowledge I have received from my work environment are highly valuable and appreciated. To me, Smart is not just an ordinary workplace – it is a home to passionate and driven people from several countries who do not discriminate each other but instead share experience to support each other. I hope the new Smart Intranet project will be useful in everyday use for all Smart employees so that they can save time and be more productive. Last but not least, I would like thank NIPTICT and Smart for making this internship possible and always be the force behind the drive of innovation. What I have obtained from this internship has enhanced me both professionally and practically thus getting me ready for the professional world and I cannot be thankful enough for everyone who has supported me until now.

REFERENCES

- Eloquent ORM and API Resource

  https://laravel.com/docs/5.7/eloquent

- Blade template

  https://laravel.com/docs/5.7/blade

- jQuery documentation

  https://api.jquery.com/

- Bootstrap documentation

  https://getbootstrap.com/docs/4.1/