



TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES
COLLEGE OF ENGINEERING
ELECTRONICS ENGINEERING DEPARTMENT



**NBC PORTFOLIO OPTIMIZATION DATABASE SYSTEM BY USING
OPTICAL CHARACTER RECOGNITION AND MACHINE LEARNING**

Submitted By:

BONGCALES, RECO I.

CABELTES, KIERVY S.

OBEJAS, JORELEEN C.

RAUTO, LOUIE ANDRO T.

RUIZ, ELLISE MARY ANNE M.

August 2020

ACKNOWLEDGEMENT

The proponents of the study entitled “NBC Portfolio Optimization Database System by Using Optical Character Recognition and Machine Learning” would like to express their utmost gratitude to the following individuals who supported and helped them throughout the completion of this study.

First and foremost, the researchers would like to offer this endeavor to God Almighty for the blessings and guidance given to them to complete this research successfully.

The researchers would also like to appreciate their thesis adviser and Project Study Professor, Engr. Nilo M. Arago, for sharing his knowledge and expertise in bringing this study into success. The proponents are grateful for his dedication in guiding them all throughout the study.

The researchers would also like to acknowledge the ECE Department, for assisting them in gathering the files and documents needed, and for being patient and understanding throughout the duration of this study.

Last but not the least, the researchers would like to give their sincerest thanks to their families for providing unending support and love. The continuous prayers and care that guided them to this journey and chapter of their life.

Finally, they wouldn't have made it if not for these people beside them. Their million thanks to everyone!

ABSTRACT

The purpose of this study is to create a database system that will store the documents in a cloud-based storage with the help of a kiosk and will automatically assign points to the verified documents in accordance with NBC (National Budget Circular) no. 461. The system database is reconfigurable as the guidelines change every three years. The system uses an Optical Character Recognition to collect keywords for Machine Learning. These keywords are used to identify the type of document that is scanned. The project also includes a portfolio optimization wherein it gives notifications and suggestions that will help the faculty member to assess which category or areas they are lacking in accordance with the NBC CCE guidelines and pointing system. The kiosk will be deployed in the Electronics Engineering Department of the College of Engineering, Technological University of the Philippines that will help the faculty members in the monitoring of points aligned with the current pointing system of the NBC no. 461 PASUC CCE Guidelines.

TABLE OF CONTENTS

Title Page	i
Acknowledgement	ii
Abstract	iii
Table of Contents	iv
List of Figures and Tables	viii
Chapter 1 The Problem and Its Background	
Introduction	1
1.1 Background of the Study	1
1.2 Statement of the Problem	2
1.3 Objectives of the Study	2
1.3.1 General Objectives	2
1.3.2 Specific Objectives	3
1.4 Significance of the Study	3
1.5 Scope and Delimitations	3
Chapter 2 Review of Related Literature	
2.1 Conceptual Literature	5
2.1.1 Scanner	5
2.1.1.1 Types Of Scanner	5
2.1.2 Detachable Screen Laptop	8
2.1.3 Wi-Fi (Wireless Network)	9
2.1.3.1 How Does It Work	9
2.1.4 Cloud Storage	10
2.1.4.1 Types Of Cloud Storage	11
2.1.4.1.1 Public Cloud Storage	11
2.1.4.1.2 Private Cloud Storage	12
2.1.4.1.3 Hybrid Cloud Storage	12
2.1.5 Java Programming	13
2.1.6 Database Management System	14

2.1.7 Optical Character Recognition	14
2.1.8 Machine Learning	15
2.1.8.1 Types Of Machine Learning	15
2.1.9 Tesseract OCR	15
2.1.10 Natural Language Processing	16
2.1.10.1 Syntactic Analysis	16
2.1.10.2 Semantic Analysis	16
2.2 Related Studies	
2.2.1 JAVA Programming	17
2.2.1.1 A Software Architecture for Java Programming Learning Assistant System	17
2.2.1.2 Java technology in the design and implementation of web applications	17
2.2.2 Database Management System	18
2.2.2.1 A University Fixed Asset Database Information Management System Based on Internet of Things	19
2.2.2.2 Database programming using Java	19
2.2.3 Optical Character Recognition	20
2.2.2.1 OCR-Based Electronic Documentation Management System	20
2.2.2.2 Optical Character Recognition Based Intelligent Database Management System for Examination Process Control	20
2.2.4 Machine Learning	22
2.2.4.1 Optical Recognition of Digital Characters Using Machine Learning	22
2.2.4.2 Implementation of Optical Character Recognition Using Machine Learning	22

2.2.5 Tesseract OCR	23
2.2.5.1 Implementation of Optical Character Recognition using Tesseract with the Javanese Script Target in Android Application	23
2.2.5.2 An Overview of the Tesseract OCR Engine	23
Chapter 3 Methodology	
3.1 Project Research Design	25
3.1.1 Structural Framework	26
3.1.1.1 Kiosk Dimensions	27
3.1.2 Conceptual Framework	28
3.1.3 Input-Process-Output Chart	29
3.2 Project Development	30
3.2.1 Hardware Development	31
3.2.2 Software Development	31
3.2.3 Gathering of Materials and Equipment	31
3.2.3.1 SCANNER (HP DJET 3776)	31
3.2.3.2 ASUS Vivo book Flip TP301UJ	32
3.2.4 Guidelines.....	33
Chapter 4 Results and Discussion	
4.1 Technical Description of the Project	39
4.2 Structural Organization of the Project	39
4.3 Project Limitations and Capabilities	47
4.4 Project Assessment	48
4.4.1 File Gathering	48
Chapter 5 Summary of Findings, Conclusions And Recommendations	
5.1 Summary of Findings	49
5.2 Conclusion	49
5.3 Recommendations	50
APPENDIX A: SUMMARY OF EXPENSES	51
APPENDIX B: GANTT CHART	52

APPENDIX C:	PROGRAM CODES	55
References		172

List of Figures and Tables

Figure 2.1 Flatbed Scanner	6
Figure 2.2 Sheetfed scanner	6
Figure 2.3 Handheld scanner	7
Figure 2.4 Drum scanner	7
Figure 2.5 Photo scanner	7
Figure 2.6 Film scanner	8
Figure 2.7 Portable scanner	8
Figure 2.8 Detachable screen laptop	9
Figure 2.11 WI-FI (WIRELESS NETWORK)	10
Figure 2.13 CLOUD STORAGE	11
Figure 2.14 PUBLIC CLOUD STORAGE	12
Figure 2.15 PRIVATE CLOUD STORAGE	12
Figure 2.16 HYBRID CLOUD STORAGE	13
Figure 2.18 JAVA PROGRAMMING	14
Table 2.1 Related Studies about Java Programming	17
Table 2.2 Related Studies about Database Management System	19
Table 2.3 Related Studies about Optical Character Region	21
Table 2.4. Related Studies about Machine Learning	22
Table 5. Related Studies about Tesseract OCR	24
Figure 3.1.1 Structural Framework of NBC 461 Database System	26
Figure 3.1.1.1 Kiosk Dimensions	27
Figure 3.2 Block Diagram of NBC 461 Document Management System	28
Figure 3.3. IPO Chart of NBC 461 Document Management System	29
Figure 3.3. Software and Hardware Flowchart	30
Figure 3.4. HP 3776 SCANNER PRINTER	31
Figure 3.5. ASUS Vivo book Flip TP301UJ	32
Table 3.1 The PASUC CCE Guidelines	33
Figure 4.1 The Kiosk	39

Figure 4.2 Login Page	40
Figure 4.3 Sample Profile Page	41
Figure 4.4 Import PDF Page	41
Figure 4.5 Employee Credit View Page	42
Figure 4.6 Summary of Points Page	43
Figure 4.7 Detailed NBC CCE Pointing System Page	44
Figure 4.8 Criteria Lacking Points Page	45
Figure 4.9 Administrator Page	46
Figure 4.10 Training View Page	46
Table 4. File Gathering	47

CHAPTER 1

INTRODUCTION

According to Commission on Higher Education (CHED), there are 90 State Universities and Colleges in the Philippines and have a total of 50,310 members of the faculty as of 2018. The Department of Budget and Management issued NBC 461 with cooperation of Philippine Association of State Universities and Colleges (PASUC). The NBC 461 is a revision of the Position Classification Plan for Faculty Positions Embodied in National Compensation Circular (NCC) No. 69 to implement rules and regulation in accordance with the modified Common Criteria for Evaluation (CCE) for faculty positions. CCE is the primary guide for recruitment, classification, and promotion of a faculty. It consists of services and achievements that will determine the performance of a faculty member with the help of the point system. The rank and sub-rank of the faculty member depends on the points they accumulated throughout the cycle. The CCE also gives more emphasis on advancement and performance rather than on educational qualifications.

1.1 Background of the Study

Professor is a term used when a degree holder started to teach in college or university. However, a professor is also defined as a faculty member at an institution of higher education with the highest academic rank. Based on National Budget Circular No. 461 (NBC 461), the faculty members of the State Universities and Colleges (SUCs) must have at least 159 CCE points to be considered as a professor.

The documents accumulated by the faculty will have to go to the local and zonal evaluators. The local evaluator approves the documents submitted by the faculty and the zonal evaluator verifies it. After verifying, the current system will allocate points depending on the submitted documents. But the current system also has its drawback and that is not having a detailed summary of points and the overall process is conducted manually which can be tedious and time consuming. The current instrument also limits

the points the faculty can earn. With this limit, the faculty members stop gaining points once it reaches the maximum allowable points for a certain category.

With the help of a kiosk that can scan and store the documents, knowing the current standing of the faculty members can be easily determined without going to the local and zonal evaluator. With the Optical Character Recognition (OCR), the system can identify the content of the document by converting the scanned image to a digital version that can be read by the computer and organized according to its category. Once the documents are sorted to its category, the system will give its corresponding points and provide recommendations on the areas the faculty members are lacking thus, making the faculty more motivated to reach higher ranks in accordance with NBC 461.

1.2 Statement of the Problem

The purpose of this study is to develop a database system that will allocate points to the scanned document and store the documents in the cloud-based storage. The study aims to solve the following specific problems:

1. How will the proponents develop an application that will automatically allocate point/s on the scanned documents?
2. How will the proponents store and organize the documents of the faculty through the cloud-based storage?

1.3 Objectives of the Study

1.3.1 General Objectives

This primary objective of the study is to build an automated system for National Budget Circular 461 PASUC CCE Guidelines that can identify, categorize, and store documents, allocate points, and create user profiles.

1.3.2 Specific Objectives

Specifically, this study aims to:

1. To build a kiosk with an easy and simple user-interface.
2. To scan the documents using a sheet fed scanner.
3. To recognize the text and classify the documents using Tesseract OCR Java Language.
4. To create a database system connected to Google cloud storage and a web application using Java Language.
5. To automatically assign points to the documents verified by the system.
6. To test the functionality and deploy the kiosk in the ECE Department.

1.4 Significance of the Study

This study aims to develop a system that will provide a detailed allocation of points based on the current CCE point system that will help the faculty in identifying the total accumulated points for a specific component. The system will also provide recommendations in areas where the faculty members are lacking points to give assistance on certain criteria to improve and focus on. This will also help in easy monitoring of the current standing of the faculty. It also has a database management system where the faculty can sort the documents easily according to its category.

Moreover, the system can scan and store the documents to a cloud-based storage which can be accessed using computers or mobile phones through the Internet. It also has an optical character recognition (OCR) feature that can detect unique keyword/s that will help to categorize the scanned document.

This study will aid the faculty members speed up the process of checking and managing the points accumulated. The system will be a reliable tool for safekeeping of the documents in case of loss or damage and while also providing security to the user's account.

1.5 Scope and Delimitations

The system consists of hardware parts such as printer, fingerprint scanner, and laptop. The proponents used Tesseract for text and image recognition and Java as

programming language. Google cloud storage is used to store the documents uploaded in the system for security purposes and easy replication.

The system is limited only to scanning, saving, and storing of documents and allocation of points. Point allocation is limited only to the documents verified by the Records. Moreover, the documents stored in the system can only be accessed by an authorized faculty member using their username and password or by using the fingerprint scanner.

Lastly, the project will be implemented in the Electronics Engineering Department of Technological University of the Philippines - Manila and is exclusive for ECE faculty members.

CHAPTER 2

REVIEW OF RELATED LITERATURE AND STUDIES

This chapter covers the theories, principles, and studies that are useful in the conceptualization of the project in the development of the idea. This involves some technical terminology from past and current developed projects.

2.1 CONCEPTUAL LITERATURE

2.1.1 SCANNER

A scanner is an electronic device that captures and converts images from physical products to digital formats that can be stored on a desktop and viewed or altered using software applications. Various scanner kinds have distinct resolutions. The most cost-effective and reliable way to transmit images is through scanning in the field of electronic data transmission.

2.1.1.1 TYPES OF SCANNERS AND HOW DOES IT WORK

Scanners work by lighting the document and directing the reflected light (usually through a sequence of mirrors and glasses) to a component that is photosensitive. Scanners mostly use a sensing medium known as a loaded coupling device (CCD), which is an integrated electronic light-sensing circuit. Light-sensitive photo sites exhibited along the CCD transform levels of brightness into electronic signals that are then transformed into a digital image.

CCD is by far the most common light-sensing technology in contemporary scanners. Two other methods, CIS (Contact Image Sensor) and PMT (photomultiplier tube), are found in the low and high ends of the scanner sector respectively. CIS is a new technology that allows reduced and lighter scanners, but sacrifices dynamic range, field depth, and resolution. PMT-based

drum scanners produce high-quality images but have restricted library and archive scanning execution.

Figure 2.1 to 2.7 shows the different types of scanners.



Figure 2.1 Flatbed Scanner

(<https://www.amazon.in/Canon-CanoScan-FB620U-Flatbed-Scanner/dp/B00000JFKO>)



Figure 2.2 Sheetfed scanner

(<https://www.indiamart.com/proddetail/hp-scanjet-7000-s2-sheet-feed-scanner-11459201248.html>)



Figure 2.3 Handheld scanner

(<https://www.adesso.com/product/nuscan-7600tu-2d-antimicrobial-handheld-barcode-scanner/>)



Figure 2.4 Drum scanner

(<https://www.timlaytonfineart.com/blog/2019/1/25-off-your-first-drum-scanning-order-35mm-up-to-8x10-large-format>)



Figure 2.5 Photo scanner

(<https://imaging-superstore.co.uk/products/kodak-ps80>)



Figure 2.6 Film scanner

(https://www.gearbest.com/scanners/pp_009606908545.html)



Figure 2.7 Portable scanner

(<https://www.amazon.com/VuPoint-Solutions-Magic-Portable-Scanner/dp/B004EFXW6Q>)

2.1.2 DETACHABLE SCREEN LAPTOP

Detachable screen (or removable touchscreen) is a term frequently connected with hybrid phones from a laptop tablet, also called a convertible laptop. Typically, the device is a fully functional tablet that is attached to a docking device to provide a complete keyboard for users. The tablet becomes the laptop's "screen" when attached. In order to use the device as a touchscreen tablet, users can disconnect the

keyboard. The devices can be purchased as a tablet or advertised as a removable screen or convertible laptop.



Figure 2.8 Detachable screen laptop

(<https://www.consumerreports.org/cro/magazine/2014/11/are-detachable-computers-the-best-of-both-worlds/index.htm>)

2.1.3 WI-FI (WIRELESS NETWORK)

Wi-Fi is the name of a common wireless networking technology that provides high-speed wireless Internet and network links using radio waves. A popular misconception is that the word Wi-Fi is short for "wireless fidelity," but that is not the case. Wi-Fi is merely a marked sentence that implies IEEE 802.11x.

2.1.3.1 HOW DOES IT WORK

Using radio frequency (RF) technology, Wi-Fi networks have no physical wired link between sender and receiver— a frequency within the electromagnetic spectrum connected with radio wave propagation. When an antenna is provided with an RF current, an electromagnetic field is developed that can then propagate through space.

Every wireless network's cornerstone is an access point (AP). An access point's main task is to transmit a wireless signal that can be detected and "tuned" by PCs. Computers and devices must be fitted with wireless network adapters to connect to an access point and join a wireless network.



Figure 2.11 WI-FI (WIRELESS NETWORK)

(https://www.webopedia.com/TERM/W/Wi_Fi.html)

2.1.4 CLOUD STORAGE

Cloud storage is a service model where information is maintained, managed, remotely backed up and made accessible to customers via a network (usually the Internet). Users usually pay a monthly price per consumption for their cloud data storage. While the price per gigabyte has been pushed down radically, cloud storage companies have added operating expenses that can make the technology more costly than customers that have been negotiated for. Cloud security among customers remains to be a problem. Providers have attempted to address these concerns by constructing safety capacities into their services, such as encryption and authentication.

Cloud-based information is stored by a third-party cloud provider in logical pools across disparate commodity servers situated on site or in a data center. Using the RESTful API, a file and its associated metadata are stored as a single object by an object storage protocol and assigned an ID number. The user will present the ID to the scheme when content needs to be obtained and the content will be assembled with all its metadata, authentication and security.



Figure 2.13 CLOUD STORAGE

(https://www.nec.com/en/global/solutions/cloud/portfolio/images/storage_img01.jpg)

2.1.4.1 TYPES OF CLOUD STORAGE

There are three main cloud-based storage architecture models: public, private and hybrid.

2.1.4.1.1 PUBLIC CLOUD STORAGE

Public cloud storage facilities provide the most suitable multi-tenant storage environment for unstructured data. Data is stored in worldwide data centers with various areas or continents spreading storage information. Usually customers pay on a per-use basis comparable to the payment utility model. Amazon Simple Storage Service (S3), Amazon Cold Storage Glacier, Google Cloud Storage, Google Cloud Storage Nearline, and Microsoft Azure dominate this business industry.



Figure 2.14 PUBLIC CLOUD STORAGE

(<https://spotherld.com/2019/04/10/global-public-cloud-storage-service-market-regional-outlook-2019/>)

2.1.4.1.2 PRIVATE CLOUD STORAGE

Private cloud or on-site storage services provide a dedicated environment that is protected behind the firewall of an organization. Private clouds are suitable for customers who need to have their information customized and controlled more.

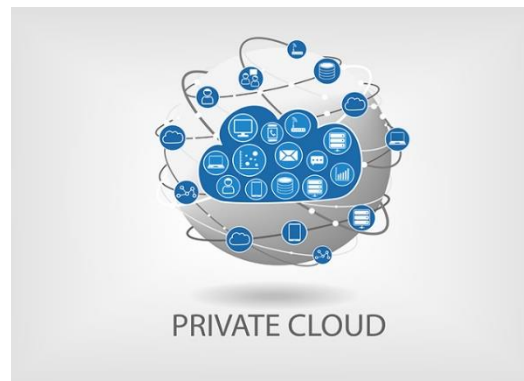


Figure 2.15 PRIVATE CLOUD STORAGE

(<https://storageservers.wordpress.com/2016/10/24/understanding-why-private-cloud-computing-works-for-your-business/>)

2.1.4.1.3 HYBRID CLOUD STORAGE

Hybrid cloud is a mixture of private cloud and public cloud services from third parties with orchestration between management

platforms. The model provides flexibility for companies and more choices for information deployment.

For instance, an organization could store actively used and organized information in a cloud on-site, and in a public cloud unstructured and archival information. In recent years, the hybrid cloud model has been embraced by a higher number of clients. Despite its advantages, there are technical, business, and management difficulties in a hybrid cloud. Private workloads, for instance, need to access and communicate with suppliers of public cloud storage, so compatibility and strong network connectivity are very significant considerations.

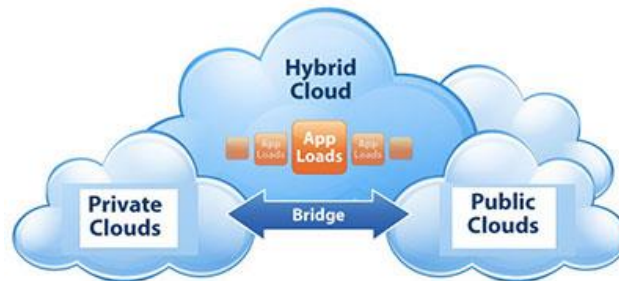


Figure 2.16 HYBRID CLOUD STORAGE

(<https://blog.inkjetwholesale.com.au/office-efficiency/top-cloud-storage-trend-of-2015/>)

2.1.5 JAVA PROGRAMMING

Java is a high-level programming language that Sun Microsystems developed but was later acquired by Oracle. Java was originally designed for developing programs for set-top boxes and handheld devices and was used later on as one of the best choices for creating web applications.



Figure 2.18 JAVA PROGRAMMING

(<https://icon-library.com/images/java-icon-image/java-icon-image-0.jpg>)

2.1.6 DATABASE MANAGEMENT SYSTEM

A database management system (DBMS) is used in accessing data as well as managing and creating databases. The DBMS is a system software that creates ways for both the users and programmers to create, update, read, manage data in a systematic manner. The DBMS ensures that the data is always organized and can be accessed easily by users. It acts as an interface between the database and the users.

2.1.7 OPTICAL CHARACTER RECOGNITION

OCR (optical character recognition) is used to recognize both printed or handwritten characters in text that are in digital images of originally physical documents. OCR involves examination of a text document and then converting it to code that can be processed. This is why it is also referred to as text recognition since it can translate characters into data or codes for processing of data into digital formats and machine-readable text.

A combination of hardware and software comprises OCR systems. Optical Scanner is one of the considered hardware tools that is used to copy or read text. Software is the one dealing with advanced processing and uses Artificial Intelligence (AI) to implement a better method such as identifying language or different styles of handwriting.

2.1.8 MACHINE LEARNING

Machine learning is a data analysis technique that automates the construction of analytical models. It is a subset of artificial intelligence centered on the concept that, with minimal human interference, devices can benefit from information, recognize trends and create choices.

2.1.8.1 TYPES OF MACHINE LEARNING

There are three types of Machine Learning Algorithms: Supervised Learning, Unsupervised Learning and Reinforcement Learning. Supervised Learning requires of a conditional variable to be selected from a specified collection of predictors (autonomous factors) then produces a feature using these number of factors that maps inputs to needed results. The learning method proceeds until a required amount of precision on the training data is achieved by the model.

Unsupervised Learning is used for population clustering in separate organizations, which is commonly used for customer segmentation in distinct organizations for particular action. There is no dependent variable to predict and estimate in this algorithm. The Reinforcement Learning instructs the computer to create particular choices. It operates by trial and error subjected to a setting and continuously trains a particular device. This device teaches from prior knowledge and attempts to obtain the best necessary understanding to make precise choices.

2.1.9 Tesseract OCR

Tesseract is an open-source OCR engine that was developed at HP between 1984 and 1994. Tesseract is very easy to implement, and subsequently isn't overly powerful. It's mainly used for reading computer generated text on black and white images, which is done with decent accuracy. Google's Tesseract engine, we built an extremely simple app that accepts an image through a form, extracts the textual contents from it and returns it to the submitted image.

2.1.10 NATURAL LANGUAGE PROCESSING

Natural Language Processing (NLP) is a branch of artificial intelligence that uses the natural language to deal with the interaction between computers and humans. It is the technology that help computers to understand, interpret and manipulate the natural language of humans in a manner that is valuable. The main methods used to obtain Natural Language Processing are syntactic analysis and semantic analysis.

2.1.10.1 Syntactic Analysis

Syntax relates to word structure in a sentence that makes grammatical meaning. In NLP, syntactic analysis is used to evaluate the alignment of the natural language with the grammar regulations. There are 7 syntax techniques that can be used: Lemmatization, Morphological Segmentation, Word Segmentation, Part-of-speech tagging, Parsing, Sentence-breaking and Stemming. Lemmatization entails reducing the different inflected types of a sentence into a single form. Morphological segmentation involves separating words into parts called morphemes. Word segmentation is dividing a big part of continuous text into separate units. Part-of-speech tagging includes recognizing the part of the expression for each sentence. Parsing includes the grammatical study of the phrase given. Sentence breaking involves putting the limits of sentences on a big section of document. Stemming: includes slicing to the root form of the inflected phrases.

2.1.10.2 Semantic Analysis

Semantics refers to the meaning of the text conveyed in a sentence. It involves the application of computer algorithms to understand the meaning and interpretation of words and the structure of sentences. Named entity recognition (NER) is one of the techniques used in semantic analysis. It involves identifying parts of a text that can be identified and classified into preset groups. Examples of such groups include lists of individuals and lists of locations. Another technique is the Word Sense Disambiguation that involves giving meaning to a context-based word. Natural language generation

use databases to obtain semantic plans and transform them into human language.

2.2 Related Studies

2.2.1 JAVA Programming

The Java programming language was primarily used for Internet-based applications. It was developed by Sun Microsystems in the early 1990s. Java is a simple, efficient, general-purpose language that was originally designed for embedded network applications running on multiple platforms. Java is a portable, object-oriented, interpreted language.

2.2.1.1 A Software Architecture for Java Programming Learning Assistant System

Nobuya Ishihara et al developed a web application Java Programming Learning System (JPLAS). A software architecture for JPLAS to avoid redundancy. The number of code files is compared with that of the previous implementation, and the number of additional files is examined for two new functions.

2.2.1.2 Java technology in the design and implementation of web applications

This study shows the development of Web applications in the Java programming language. The paper shows the advantages of using Java in web applications and the methods to implement when using it.

Table 2.1 Related Studies about Java Programming

TITLE	AUTHOR	METHODOLOGY	FINDINGS
A Software Architecture for Java Programming	Nobuya Ishihara, Nobuo Funabiki1, Minoru	Web-based Java Programming	Students have experienced programming for

Learning Assistant System	Kuribayashi and Wen-Chung Kao	Learning System (JPLAS). JPLAS provides four problems with different levels, namely, element fill-in blank problem, value trace problem, statement fill-in-blank problem, and code writing problem, to cover students at different learning stages.	practical systems that have been used in Java programming courses in universities
Java technology in the design and implementation of web applications	Sead Mašović, Muzafer Saračević, Hamza Kamberović, Mensura Kudumović	Web application through three-layer architecture using Java Servlet technology and Java Server Pages	Using three-layer architecture for creating Web pages greatly facilitates their maintenance because changes of one layer do not require changing the other layers, which makes the application easy to transfer.

2.2.2 Database Management System

DBMS can be the most effective way in terms of providing multiple users a centralized view of information or data that can be accessed in a controlled manner

from multiple locations. It is important that the method used in the project is reliable to avoid data infringement.

2.2.1.1 A University Fixed Asset Database Information Management System Based on Internet of Things

Zhijian Yu et al used barcode technology in Database Information Management System providing financial data to office systems and campus network interface. This work applied barcode technology in realizing the technology that is based on the internet of things.

2.2.1.2 Database programming using Java

This study focuses on database programming using Java. A database programming using Java for the Microsoft Access DBMS. The Java programming language has the potential of working with different databases like SQL Server, Oracle, Informix, Sybase, Microsoft Access, and others. Java uses the JDBC (Java Database Connectivity) tool to work with the database.

Table 2.2 Related Studies about Database Management System

TITLE	AUTHOR	METHODOLOGY	FINDINGS
A University Fixed Asset Database Information Management System Based on Internet of Things	Zhijian Yu, Chengyang Yuan, Ke Zheng	Database information management systems use barcode technology as the means for realizing technology of the internet of things.	Provides financial data, office system, and campus network platform interface and realize asset information sharing

Database programming using Java	M. Swain, J.A. Anderson, R. Korrapati, N.K. Swain	Introduces the concept of a JDBC-ODBC bridge for Microsoft Access, which is a useful system for understanding and teaching database programming.	The concepts of Database Management systems using Java are illustrated.
---------------------------------	---	--	---

2.2.3 Optical Character Recognition

Optical Character Recognition will be used to convert physical documents to digital formats that will let the user to save soft copy of documents and certificates for future use. This will recognize characters in the scanned document to be inputted in the system.

2.2.2.1 OCR-Based Electronic Documentation Management System

Khalaf S. Alkhalaf et al created a web application that enables users to upload a scanned document in PDF format that is editable if the information scanned is not well recognized. This work used an optical character recognition software that has a very high accuracy in distinguishing Arabic language.

2.2.2.2 Optical Character Recognition Based Intelligent Database Management System for Examination Process Control

Mehdi Rizvi et. al used OCR in examination process control and created a database system for examination process control. The tasks include image alignment, thresholding, blurring, noise reduction, and segmentation.

Table 2.3 Related Studies about Optical Character Region

TITLE	AUTHOR	METHODOLOGY	FINDINGS
OCR-Based Electronic Documentation Management System	Khalaf S. Alkhalaf, Abdulelah I. Almishal, Anas O. Almahmoud, and Majed S. Alotaibi	Creating a software by making a simple web application that allows users to upload a PDF scanned document in Arabic language. Then, based on the OCR engine, the software analyzes the document based on positions. After recognizing desired information, it will be shown as an editable field to allow the user to correct some information that might be not well recognized.	The software developed has very high accuracy rate in recognizing the Arabic language
Optical Character Recognition Based Intelligent Database Management System for Examination Process Control	Mehdi Rizvi, Hasnain Raza, Shan Jaffry and Shahab Tahzeeb	The technology used OpenCV for pre-processing. The tasks include image alignment, thresholding, blurring, noise reduction and segmentation to make the image of the first page of examination copy understandable for digit recognition algorithms.	The work has obtained the efficiency of 98.5% in recognition of digits from the images of examination copies.

2.2.4 Machine Learning

2.2.4.1 Optical Recognition of Digital Characters Using Machine Learning

Dr Sunanda Dixit et al created a model that utilizes OCR using machine learning and trained with various images. This study used an image as an input then processed the image and used the recognized characters for machine learning.

2.2.4.1 Implementation of Optical Character Recognition Using Machine Learning

This research focuses on how OCR is one of the main tools used to train machines. The paper shows the implementation of OCR functions that is accompanied by machine learning algorithms.

Table 2.4. Related Studies about Machine Learning

TITLE	AUTHOR	METHODOLOGY	FINDINGS
Optical Recognition of Digital Characters Using Machine Learning	Dr Sunanda Dixit, Bharath M, Amith Y, Goutham M L, Ayappa K, Harshitha D	The recognition system is input an image of any format (jpeg, png, etc). This is done either through scanning an image from any of the digital scanners or by loading it from the internal storage. Image processing using python.	This model is able to recognize texts in optical form. Input can be fed to this model either through scanning printed text or a digital image. Texts will be printed in a text box and each character is read out by the voice synthesizer function.
Implementation of Optical Character Recognition Using Machine Learning	Vishal Chourasia, Sanjay Silakari, Rajeev Pandey	The investigations in this direction focus on three important preprocessing tasks	The accuracy of the proposed system is 85-90%

		i.e., detection and removal of horizontal and vertical lines, detection of scratched words in pre-printed documents and printed and handwritten text classification.	Approximately from Around 25 images used for simulation. The accuracy of system is calculated on the basis of recognized characters.
--	--	--	--

2.2.5 Tesseract OCR

Tesseract can recognize more than 100 languages out of the box and still continuously grow. Tesseract is the most accurate open-source engine available on the market.

2.2.5.1 Implementation of Optical Character Recognition using Tesseract with the Javanese Script Target in Android Application

Abdul Robby G. et al proposed a study that utilizes the Tesseract OCR. The authors used three training methods in reading handwritten Javanese characters. The first one is using a separate bounding box to interpret the main body of the scripts and the phonetic symbols. Second, using the same bounding box to interpret both the main body and the phonetic symbols and, third, the combination (a hybrid) of first and second methods.

2.2.5.2 An Overview of the Tesseract OCR Engine

Ray Smith utilizes Tesseract OCR and classified the calculation process into two: class pruner and configuration. The author explored the Tesseract OCR functions and identified the weakness and strengths of the program.

Table 5. Related Studies about Tesseract OCR

TITLE	AUTHOR	METHODOLOGY	FINDINGS
Implementation of Optical Character Recognition using Tesseract with the Javanese Script Target in Android Application	Abdul Robby G., Antonia Tandra, Imelda Susanto, Jeklin Harefa, Andry Chowanda,	This research proposes three training methods. First, using a separate bounding box to interpret the main body of the scripts and the phonetic symbols. Second, using the same bounding box to interpret both the main body and the phonetic symbols and, third, the combination (a hybrid) of first and second methods.	The research has high accuracy in reading handwritten Javanese characters.
An Overview of the Tesseract OCR Engine	Ray Smith	Classification in two steps: class pruner and configuration. Each feature fetches, from a coarsely quantized 3-dimensional lookup table, a bit-vector of classes that it might match, and the bit-vectors are summed over all the features.	The key strength of Tesseract OCR is the unusual choice of features. Its weakness is the use of a polygonal approximation as input to the classifier instead of the raw outlines.

CHAPTER 3

METHODOLOGY

The purpose of this study is to improve the document management system of the Electronics Engineering Department which provides a detailed allocation of points of the current CCE point system that will help the faculty in identifying the total accumulated points for a specific component. The researchers will build an automated system for National Budget Circular 461 which can identify, categorize, and store documents, points, and user profiles by Optical Character Recognition (OCR) using Tesseract OCR Java Language and these files will be stored to a Cloud-Based Storage. This chapter presents the systematic procedures and investigation of methods used in the development and implementation of the project.

3.1 Project Research Design

The researchers will build an automated system for National Budget Circular 461 that will be used for the accreditation purposes and document management system of the Electronics Engineering Department, College of Engineering of the Technological University of the Philippines - Manila. The system is designed to scan documents using a sheet fed scanner; to store and manage scanned documents in a Google cloud storage; to sort documents using Tesseract OCR; and to assign points using Java programming language. Additionally, the system is expected to provide recommendations on certain criteria where the users are lacking points. It also uses Secure File Transfer Protocol (SFTP) for facilitating data access and data transfer.

3.1.1 Structural Framework

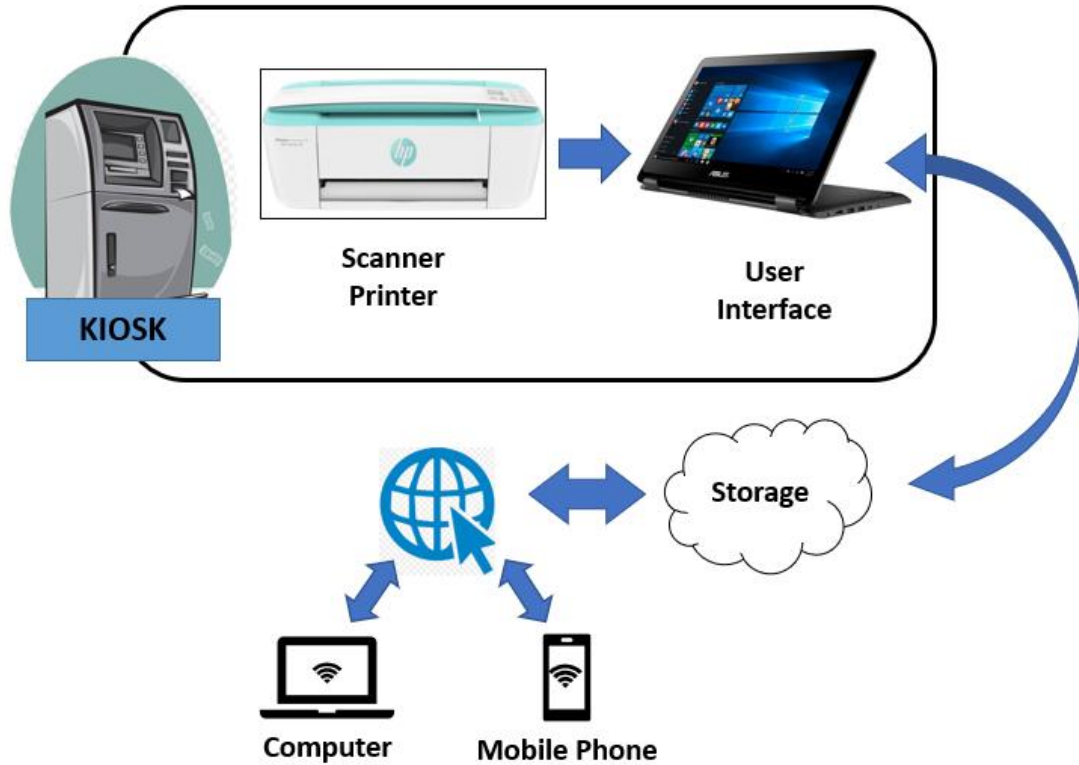


Figure 3.1. Structural Framework of NBC 461 Database System

Figure 3.1 shows the structural framework of NBC 461 Document Management System consisting of Kiosk which serves as the housing of the system and contains laptop, scanner, and fingerprint scanner. The scanner will convert the hard copy of the document into an image and the user needs to save it as pdf. Then, the user will have to log in to the system installed in the user interface (laptop) to upload the file. The uploaded file will be stored in cloud storage and exclusive for respective accounts registered on the system. The application is accessible through the Internet using computer, laptop, or mobile phone.

3.1.1.1 KIOSK DIMENSIONS

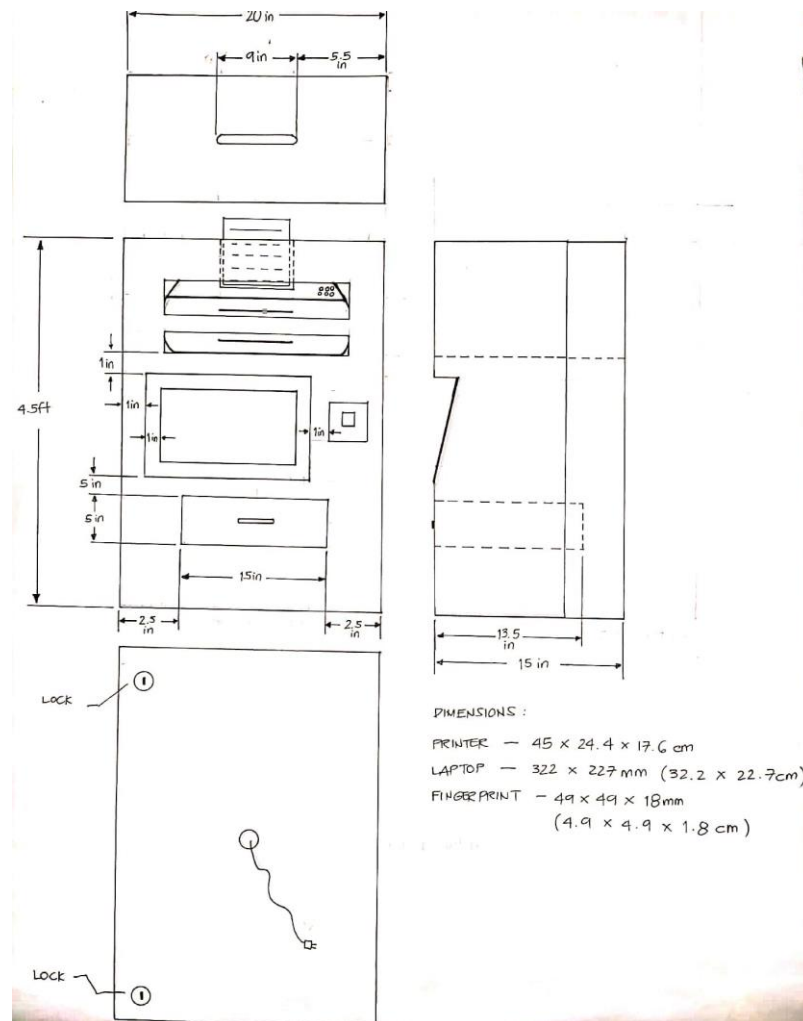


Figure 3.2 Kiosk Dimensions

3.1.2 Conceptual Framework

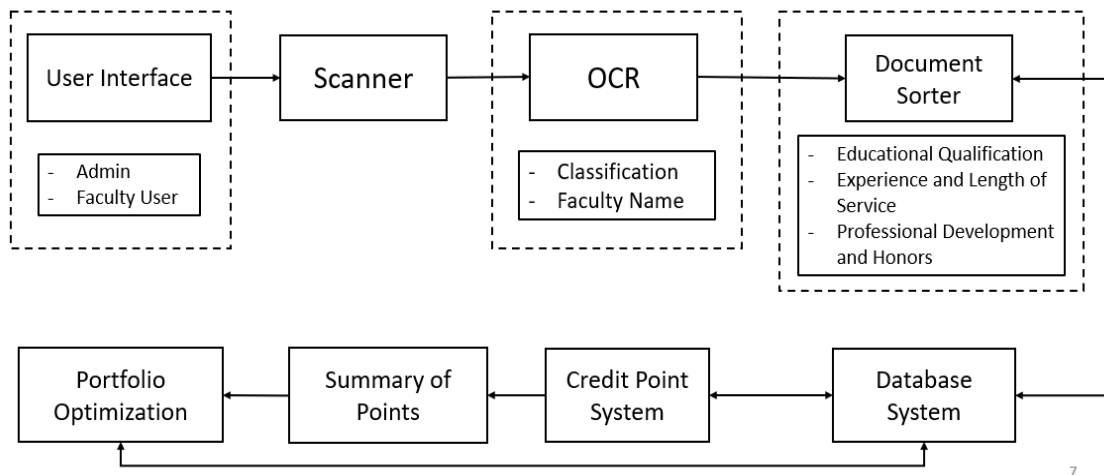


Figure 3.3 Block Diagram of NBC 461 Document Management System

Figure 3.3 shows the block diagram of the system. First, the user/admin must be logged in to access the machine (kiosk) and the machine is using Secure File Transfer Protocol (SFTP) in facilitating data access and data transfer. Next, insert documents to the Scanner and convert to softcopy documents (image). After scanning, Optical Character Recognition (OCR) only identifies the keywords in certain documents. Then, the Document Sorter will automatically categorize the documents into three: Educational Qualification, Experience and Length of Service, and Professional Development Achievement and Honors, and compute the corresponding points. The Database System is a storage of keywords, credential points and softcopy documents (image). The Credit Point System is interconnected to the Database System. Summary of Points has a display button indicating all accumulated points and can be printed through a PDF (Portable Document File) preview. Lastly, Portfolio Optimization provides recommendations/suggestions to the users on what areas are lacking.

3.1.3 Input-Process-Output Chart

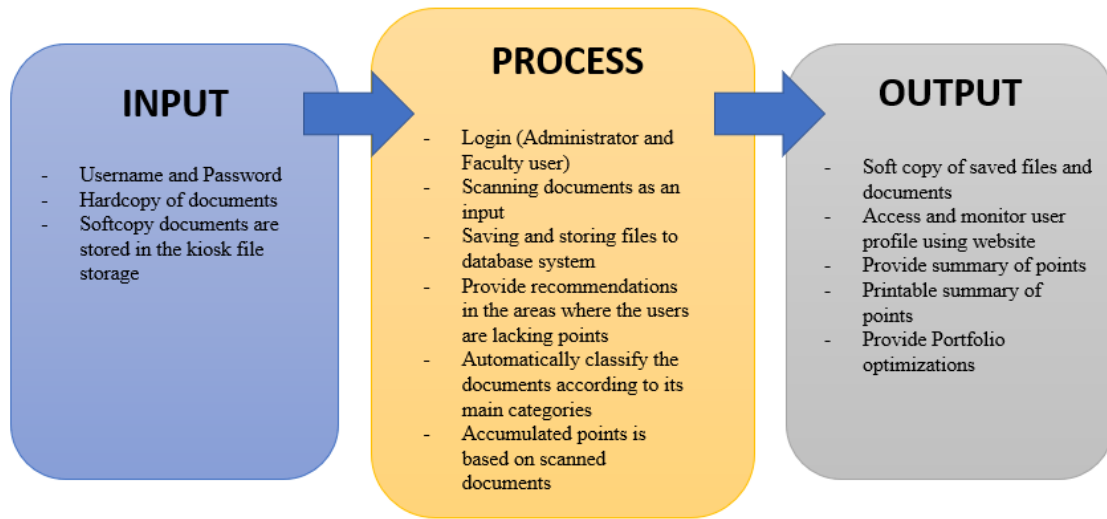


Figure 3.4. IPO Chart of NBC 461 Document Management System

Figure 3.4 shows the IPO Chart of NBC 461 Document Management System. The user needs to convert the document into an image by scanning it using the sheetfed scanner. Once converted to image, the user can upload and store the image by logging in to the system using their username and password.

After logging in, the user will have to choose between administrator and faculty. The administrator has the authority to customize the point allocation if there are any changes or revisions in the NBC guidelines. Meanwhile, the faculty is restricted only to upload, download, and delete documents in their respective accounts. The system will automatically classify the uploaded document according to its main category based on the text and keywords identified by the Optical Character Recognition (OCR). Point/s will be given to the document based on its category. The system will now compute the overall points for each category and provide recommendations for categories with low points.

Summary of points will be shown in the user's account and the user has the option to download and print it.

3.2 Project Development

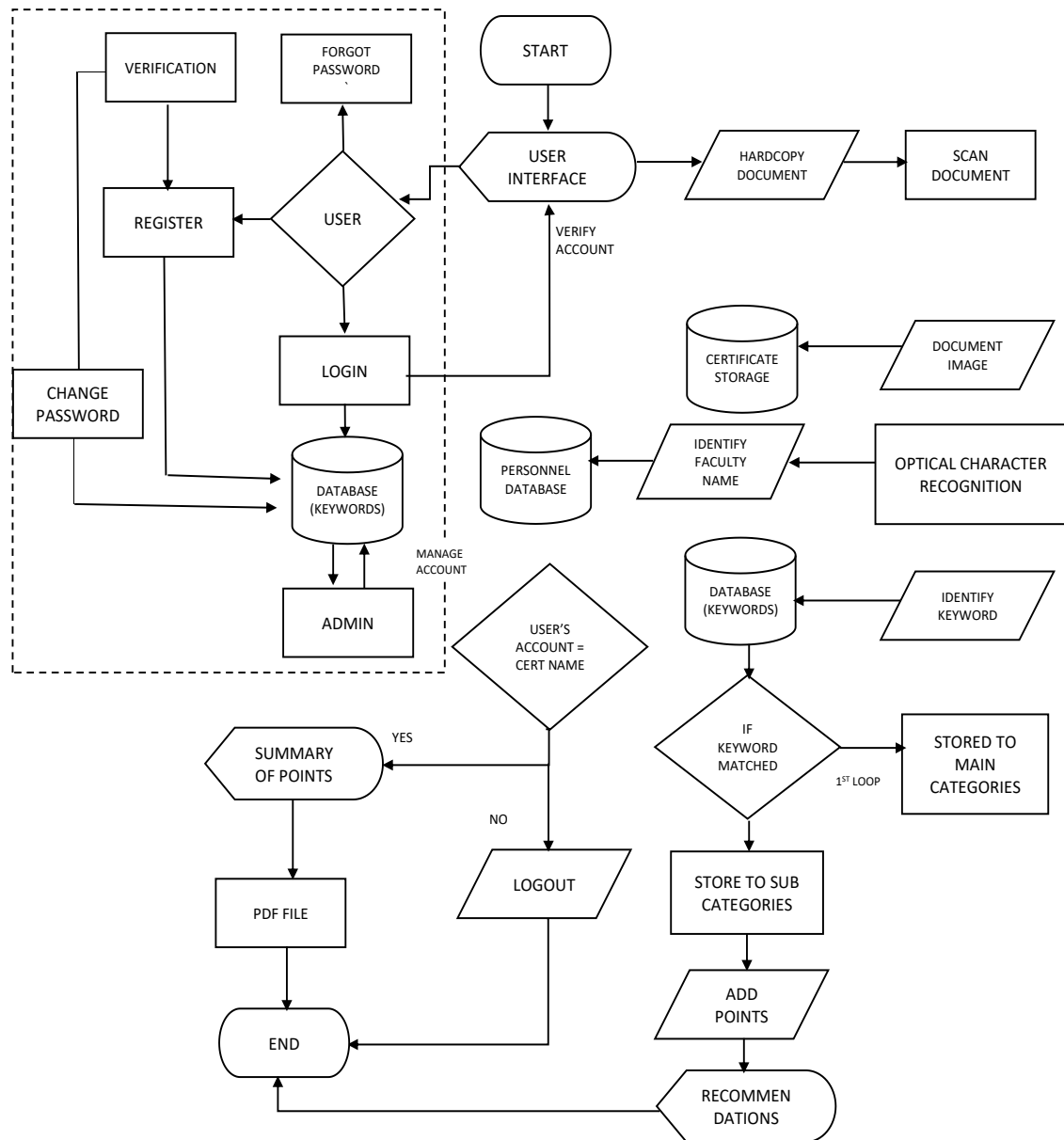


Figure 3.5. Software and Hardware Flowchart

Figure 3.5 shows all the processes that happen inside the system depending on the function the user intends to do.

3.2.1 Hardware Development

The system consists of hardware components such as laptop, sheetfed scanner, and fingerprint scanner placed in a kiosk. The laptop serves as the user interface of the system in which the application is installed. The Sheetfed Scanner is HP 3776 scanner printer which will be used to scan the documents.

3.2.2 Software Development

The application is divided into smaller and basic program which are then regrouped to create a whole complex program. The programming language to be used for the application is Java.

3.2.3 Gathering of Materials and Equipment

The needed materials and tools for the development of the project were gathered from different sources.

3.2.3.1 SCANNER (HP DJET 3776)



Figure 3.6. HP 3776 SCANNER PRINTER

(<https://store.hp.com/th-en/default/hp-deskjet-ink-advantage-3776-all-in-one-printer-t8w39b.html>)

The HP 3776 All-in one scanner printer is compatible with Google Cloud services and gets connected quickly and starts printing faster with easy set-up from smartphone or tablet. Easily scan, edit, save, and transfer scanned documents.

3.2.3.2 ASUS Vivo book Flip TP301UJ



Figure 3.7. ASUS Vivo book Flip TP301UJ

(<https://www.notebookcheck.net/Asus-VivoBook-Flip-TP301UJ-DW008T.177260.0.html>)

ASUS Vivo book Flip TP301UJ allows the user to do more in less time with an energy efficient battery. The display size of 13.3 inches is the biggest size for typical notebook computers. Powered by an Intel Core i5 6200U, it provides the user the platform to do work on the Windows operating system. With its specifications of 8GB RAM (Random Access Memory) and 1TB hard disk capacity accompanied by the large screen size, it is a good choice to serve as the main screen for the kiosk to be developed.

3.2.4 Guidelines

Table 3.1 shows the NBC no. 461 PASUC CCE Guidelines that is the criteria for the categorization and automatic pointing of the documents

Table 3.1 The PASUC CCE Guidelines

1.0 Education Qualification		(85 pts.)
1.1 Highest relevant academic degree or educational attainment.	1.1.1 Doctorate	85 pts.
	1.1.2 Master's Degree	65pts.
	1.1.3 LLB and MD	65pts
	1.1.4 Diploma Course (Above a Bachelor's Degree)	55pts.
	1.1.5 Bachelor's Degree a. Four years b. Exceeding four years	45pts. 45 plus 5 pts. For every year over 4 yrs.
	1.1.6 SPECIAL COURSES a. 3- years post-secondary course b. Special Courses (Non- degree)	30 pts. 25 pts.
1.2 Additional equivalent degree earned related to the present position	1.2.1 Master's Degree	4.0
	1.2.2 Bachelor's Degree	3.0
1.3 Additional credits earned (maximum of 10 pts)	1.3.1 For every 3-unit credit earned towards an approved higher degree course	1.0
2.0 Experience and Length of Service		(25pts)
2.1 Academic Experience	2.1.1 For every year of full-time academic service in a state institution of higher learning (1.0)	
	2.1.2. For every year of full-time academic service in an institution of higher learning other than SUCs, CHED-Supervised and TESDA Schools; service in a public or private research institution (0.75)	

2.2 Administrative Experience For every full-time year of administrative experience as: <ul style="list-style-type: none"> a. President b. Vice President c. Dean/Director/School Superintendent d. Principal/Supervisor/Department Chairperson/ Head of Unit 		3.0 2.5 2.0 1.5
2.3.1 For every year of relevant full-time professional and technical experience as: <ul style="list-style-type: none"> a. Manager/Entrepreneur/Consultant b. Supervisor/Head of Unit c. Rank and File 		1.50 1.0 0.5
2.3.2 For every year of experience in the public and private basic institution: <ul style="list-style-type: none"> a. Cooperating Teacher b. Basic Education Teacher 		0.75 0.50
3.0 Professional Development Achievement and Honors		90pts
3.1 Discoveries, patented inventions, innovations, publications and other creative works (maximum of 30 points)	3.1.1. For every cost and time saving innovation, patented invention and creative work as well as discovery of an educational, technical, scientific and/or cultural value.	2 to 7 pts
	3.1.2. For every published book: original, edited or compiled, copyrighted/ published within the last ten years, 2nd editions and succeeding editions will be credited like the original book if there is a major revision of the contents of the book evidenced by the granting of new copyright and new ISBN. <ul style="list-style-type: none"> a. As author/s b. As reviewer c. As translator d. As editor 	3-7 1-4 1-4 1-3

	e. As compiler	1-2
	<p>3.1.3 For every scholarly research/ monograph/Educational technical articles in a technical/Scientific/professional journal including electronic and digital journals included in the lists of CHED, ISI, Harvard, SCOPUS and other journals of sterling reputation for international and national. Local journals refer to institutional research-based publications</p> <p>a. International b. National c. Local</p>	<p>5 3 2</p>
	3.1.4 For every instructional manual/ audio-visual	1 (max. of 10pts)
3.2 For expert services, training and active participation in professional/technical activities (maximum of 30 points)	3.2.1 Training and seminars	(maximum of 10 points)
	<p>3.2.1.1 For every training course with a duration of at most one month not to exceed the full credit (P=No. of days/30)</p> <p>a. International b. National c. Local</p>	<p>5 3 2</p>
	3.2.1.2 For participation in conferences, seminars, workshops (must be relevant to one's assignment/field)	
	3.2.2 Expert service rendered	(maximum of 20 points)

	<p>3.2.2.1 For serving as a short-term consultant /expert in an activity of an educational, technological, professional scientific or cultural nature (foreign or local) sponsored by government or other agencies.</p> <p>International 5 National 3 Local 2</p>	
	<p>3.2.2.2 For services rendered as trainer, coordinator, lecturer, resource person or guest speaker in conferences, workshops, and/or training courses for professionals</p> <p>International 5 National 3 Local 2</p>	
	<p>3.2.2.3. For expert services as adviser in doctoral dissertation, master's and undergraduate thesis, or their equivalents as requirements for the completion of academic programs (maximum of 10 points)</p> <p>Doctoral dissertation 1.00 Master's thesis 0.50 Undergraduate thesis 0.25</p>	
	<p>3.2.2.4. For certified services as member of the Board of Examiners in the Professional Regulations Commission (PRC) or in the Civil Service Commission (CSC)</p>	1.00
	<p>3.2.2.5. For expert services in accreditation/ quality assurance work as member of the Board of Director, Accreditor, Member of the Technical Committee or Consultant Group in regional or national agencies</p>	1.00

	3.2.2.6 For every year of expert service as testing officer/assessor in trade skills certification	1.00
	3.2.2.7. For every year of services as coach /trainer of the students in official activities and adviser of accredited students organization not to exceed 1 point per year	1.00
3.3. Membership in professional organizations/honor societies and honor received (maximum of 10 pts.)	<p>3.3.1. For current individual membership in relevant professional organization(s)</p> <p>a. Learned society</p> <p>Full member 2</p> <p>Associate member 1</p> <p>b. Honor society 1.0</p> <p>c. Scientific society 1.0</p> <p>d. Professional Officer 1.0</p> <p>Member 0.5</p>	
	<p>3.3.2. For academic honors earned:</p> <p>3.3.2.1. Undergraduate Degree</p> <p>a. Summa cum Laude 5</p> <p>b. Magna cum Laude 3</p> <p>c. Cum Laude 1</p> <p>3.3.2.2. Graduate Degree</p> <p>a. Highest Honors/ With Distinction. Or equivalent 3</p>	
	<p>3.3.3. Scholarship/ Fellowship. This may be degree or non- degree granting.</p> <p>International, competitive</p> <p>Doctorate 5</p> <p>Master's 4</p> <p>Non- degree 3</p> <p>International, non- competitive</p> <p>Doctorate3</p>	

	Master's2 Non-degree.....1 National/Regional, competitive Doctorate.....3 Master's.....2 Non-degree.....1 National/ Regional, non-competitive Doctorate.....2 Master's.....1 Local, competitive or Non-competitive.....1	
3.4 Awards of distinction received in recognition of achievement in relevant areas of specification/profession and/or assignment of faculty concerned:	International	5
	National/ Regional	3
	Local	2 (institution-wide)
3.5 Community Outreach (maximum of 5 points)	3.5.1 For every year of participation in service-oriented projects in the community	1
3.6. Professional examinations	3.6.1 For every relevant licensure and other professional examinations passed (maximum of 10 pts.)	
	a. Engineering, Accounting, Medicine, Law, Teacher's Board, etc	5
	b. Career Executive Service Officers Examination / Career Service Executive Examination...	3
	c. Seaman Certificate; Master Electrician/Master Plumber Certificate, etc.; Plant Mechanic Certificate; Professional Radio Operator Certificate...	2
	d. IT proficiency certification...	2
	National Certificates (NC)/Trade skill Certificates...	1/level

Chapter 4

RESULTS AND DISCUSSION

This chapter presents the project analysis, project structure, results, and discussion. This chapter shows the results gathered after testing the system.

4.1 Technical Description of the Project

The proponents developed a system that can sort and assign points to the documents based on the PASUC CCE pointing system. The system is composed of laptop, sheetfed scanner, fingerprint scanner, and an application with Optical Character Recognition (OCR) feature developed using Java programming language.

4.2 Structural Organization of the Project



Figure 4.1 The Kiosk

Figure 4.1 shows the kiosk. The 13.3-inch laptop serves as the user interface for the kiosk. The sheetfed scanner printer is at the top shelf covered by an acrylic glass. It also has three compartments that can be used for storage of other materials needed such as bond paper.

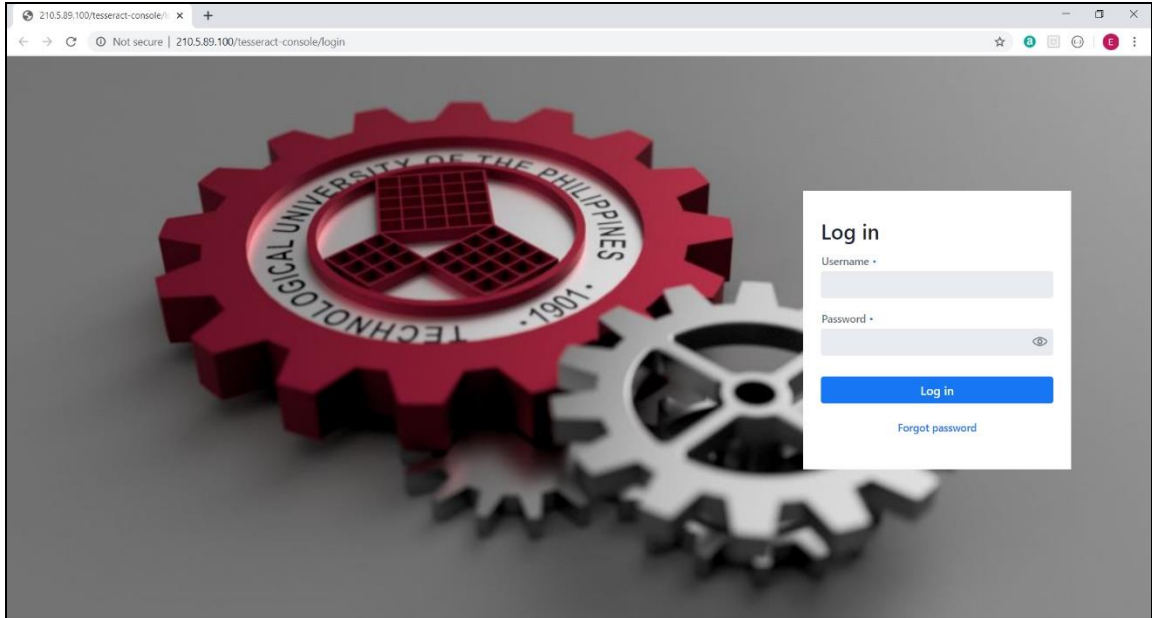


Figure 4.2 Login Page

Figure 4.2 shows the login page of the system where the user can enter their credentials such as username and password. The “username” will be the name of the faculty member and the default password is “1”. The user can reset their password in this page by clicking the “Forgot password”.

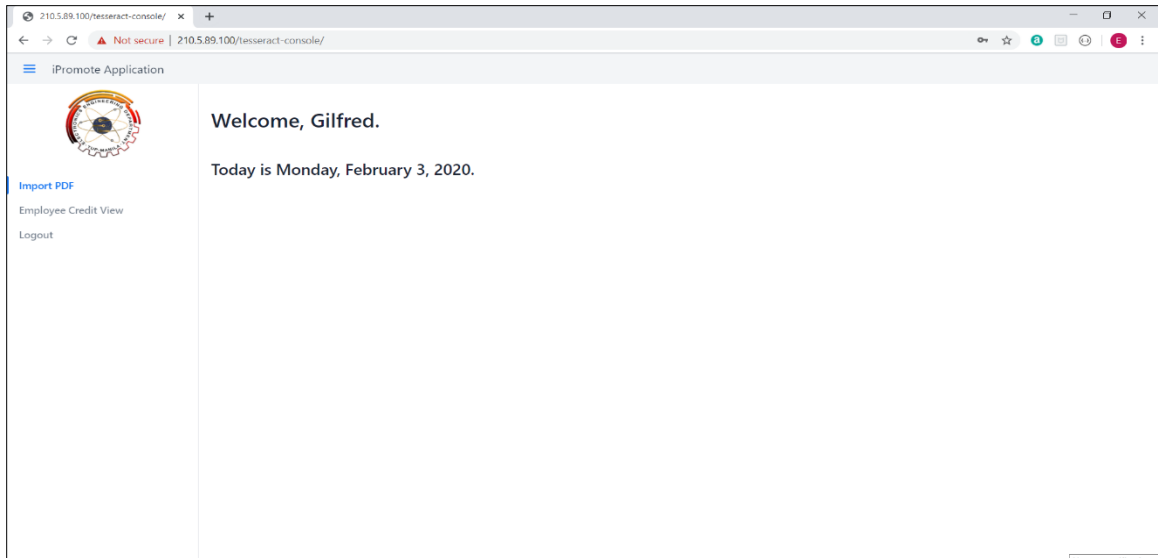


Figure 4.3 Sample Profile Page

Figure 4.3 shows the profile page that displays the information about the user. It also shows other tabs that can be navigated by the user such as the import PDF, employee credit view and the log out.

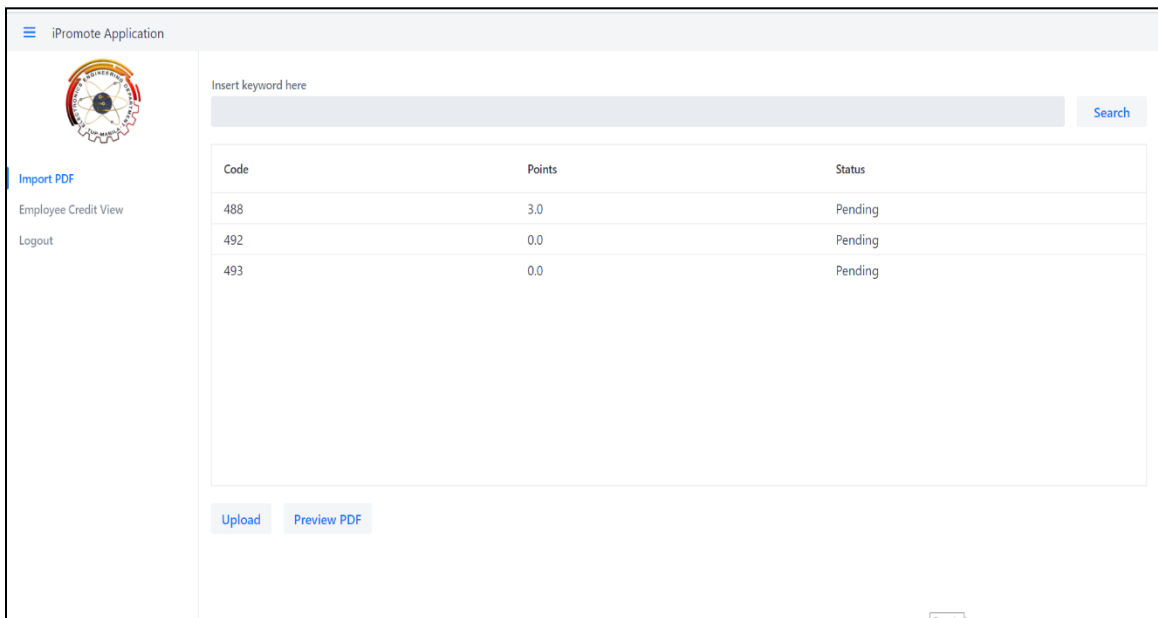


Figure 4.4 Import PDF Page

Figure 4.4 shows the feature where the user can import documents and certificates. This page allows the user to preview the uploaded file in Portable Document Format (PDF). It also shows the credit points once a document or certificate is uploaded.

Category	Target Points	Credit Points
1.0 Educational Qualification	85	9.0
2.0 Experience and Length of Service	25	0.0
3.0 Professional Development Achievement and Honors	90	0.0

Figure 4.5 Employee Credit View Page

Figure 4.4 shows the Employee Credit View page where the three main criteria for evaluation is displayed. The three main criteria are Educational Background, Experience and Length of Service, and Professional Development, Achievement and Honors. This page also shows the target maximum points per criteria and the credit points earned by the user.

Technological University of the Philippines
 Ayala Boulevard, Ermita, Manila

Name of Faculty: _____

College/Campus: _____

Present Rank: _____

Department: _____

PASUC Common Criteria for Evaluation of Faculty
NBC 461
SUMMARY OF POINTS
 _____ Cycle covering the Period of _____ to _____

Major Components	Maximum Points	Previous Point as of	Additional Point as of	Total
1.0 Educational Qualification	85	0	9	9
2.0 Experience and Length of Service	25	0	0	0
3.0 Professional Development Achievement and Honors	90	0	0	0
TOTAL	200	0	9	9

Local Evaluation Committee:

Review Committee:

Chairman

Date

Member

Date

Member

Date

Member

Date

Member

Date

Chairman

Date

Member

Date

Member

Date

Member

Date

Member

Date

Figure 4.6 Summary of Points Page

Figure 4.5 displays the summary of points acquired by the user for the main categories. The user will have to print this document and submit it to the local and review committee for verification and signature.

Technological University of the Philippines					
Ayala Boulevard, Ermita, Manila					
PASUC Common Criteria for Evaluation of Faculty					
NBC 461					
		Evaluator's Rating		Verifier's Rating	
		Raw Score	Equivalent points	Raw Score	Equivalent points
		RS	EP	RS	EP
1.0	Educational Qualification	Points : 85.0			
1.1	Highest relevant academic degree or educational attainment				
N.B In case of research-work graduate program, certified true copy of diploma and periodic evaluation of research from the supervisor/adviser shall be presented.					
1.1.1	Doctorate	85.0	0.0	0.0	0.0
1.1.2	Master's Degree	65.0	0.0	0.0	0.0
1.1.3	LLB and MD	65.0	0.0	0.0	0.0
N.B.: MD is considered a doctorate if holder is licensed and teaching in a College of Medicine. LLB is considered a Master's Degree if holder is a bar passer and teaching professional subject.					
1.1.4	Diploma Course (Above a Bachelor's Degree)	55.0	0.0	0.0	0.0
N.B.: Diploma in line with the field of specialization					
1.1.5	Bachelor's Degree				
N.B.: Graduates of DVM, Engineering and DDM programs are considered bachelor's degree holders with additional points for years beyond 4					
a.	Four years	45.0	0.0	0.0	0.0
b.	Exceeding four years	50.0	0.0	0.0	0.0
1.1.6	SPECIAL COURSES				
a.	3-years post secondary course	30.0	0.0	0.0	0.0
b.	Special Courses (Non- degree)	25.0	0.0	0.0	0.0
1.2	Additional equivalent degree earned related to the present position				
1.2.1	Master's Degree	4.0	0.0	0.0	0.0

Figure 4.7 Detailed NBC CCE Pointing System Page

Figure 4.7 shows the detailed NBC CCE pointing system and the points acquired by the user for the main criteria and subcategories. It also shows the maximum points for each subcategory.

Technological University of the Philippines
 Ayala Boulevard, Ermita, Manila

PASUC Common Criteria for Evaluation of Faculty
NBC 461

		Evaluator's Rating		Verifier's Rating	
		Raw Score	Equivalent points	Raw Score	Equivalent points
		<u>RS</u>	<u>EP</u>	<u>RS</u>	<u>EP</u>
1.0	Educational Qualification	Points : 85.0			
1.1 Highest relevant academic degree or educational attainment N.B In case of research-work graduate program, certified true copy of diploma and periodic evaluation of research from the supervisor/adviser shall be presented.					
1.1.1 Doctorate		85.0	0.0	0.0	0.0
1.1.2 Master's Degree		65.0	0.0	0.0	0.0
1.1.3 LLB and MD		65.0	0.0	0.0	0.0
N.B.: MD is considered a doctorate if holder is licensed and teaching in a College of Medicine. LLB is considered a Master's Degree if holder is a bar passer and teaching professional subject.					
1.1.4 Diploma Course (Above a Bachelor's Degree)		55.0	0.0	0.0	0.0
N.B.: Diploma in line with the field of specialization					
1.1.5 Bachelor's Degree					
N.B.: Graduates of DVM, Engineering and DDM programs are considered bachelor's degree holders with additional points for years beyond 4					
a. Four years		45.0	0.0	0.0	0.0
b. Exceeding four years		50.0	0.0	0.0	0.0
1.1.6 SPECIAL COURSES					

Figure 4.8 Criteria Lacking Points Page

Figure 4.8 shows the criteria lacking points page. This page will show the user the subcategories in the guidelines with insufficient points. This will help the user to focus on acquiring point/s and monitor these categories.

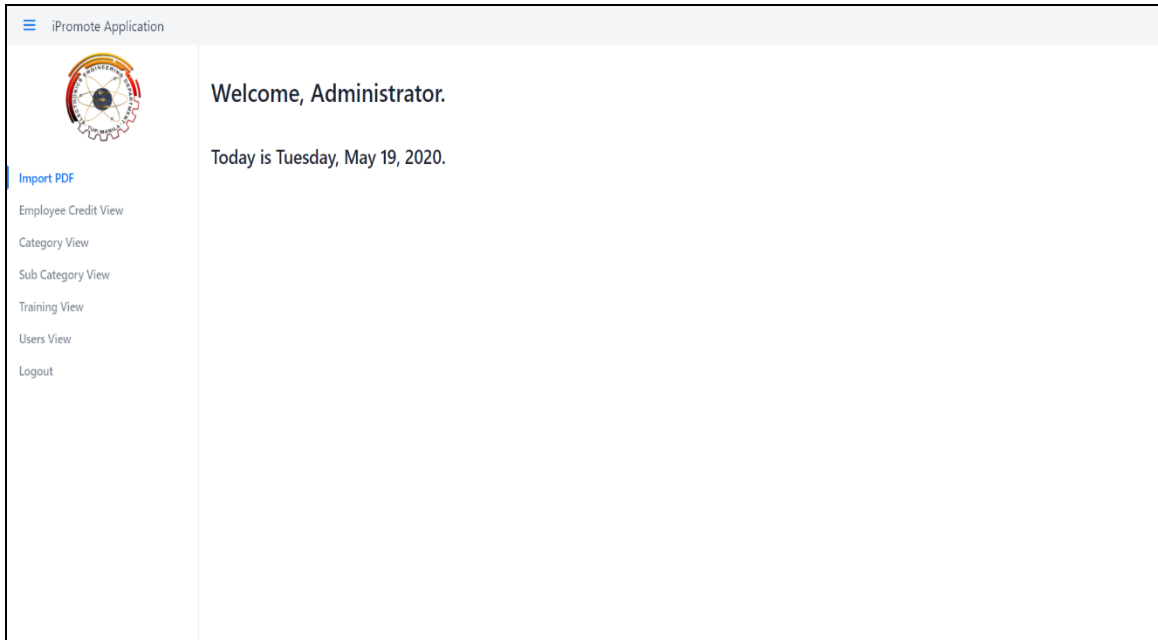
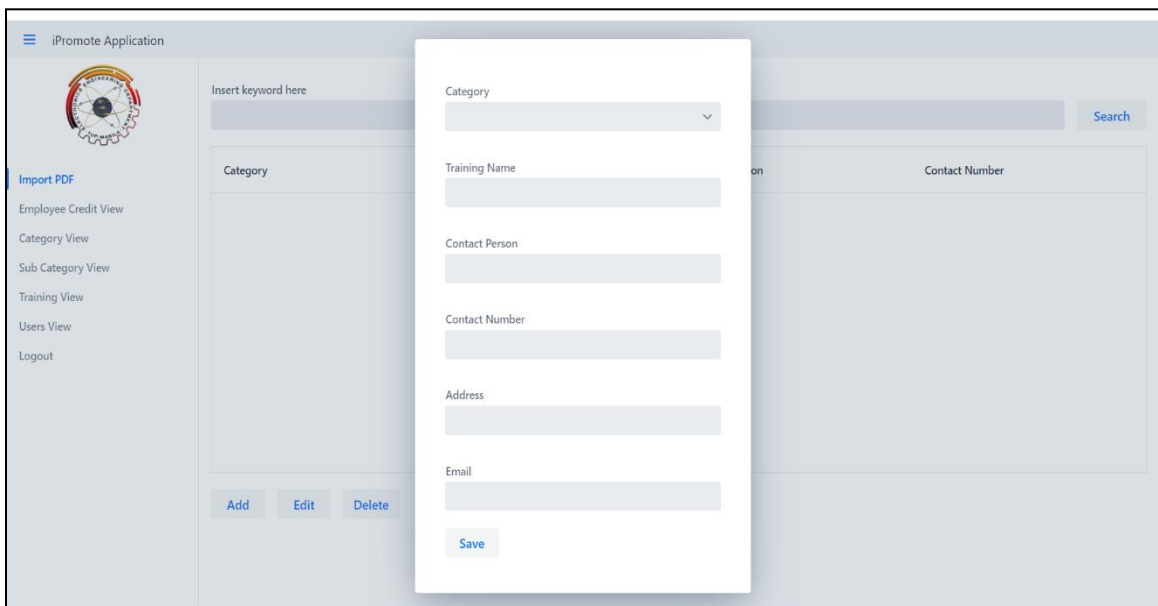


Figure 4.9 Administrator Page

Figure 4.10 shows the administrator page. In this page, the administrator can control and modify the points based on the latest version of NBC 461 guidelines. The administrator can also add list of training per category in the training view tab.



4.10 Training View Page

Figure 4.10 shows the Training View page where the administrator can add future trainings, seminars, conferences, and other important events that faculty members can join.

4.3 Project Limitations and Capabilities

This study focused on developing a system that will help the faculty members in their evaluation in accordance with the NBC PASUC CCE pointing system. This will serve as a management system for the faculty members in which they can store, sort, and monitor their documents and credit points needed to evaluate their ranking.

The system was able to scan the documents using the HP 3776 sheetfed scanner. The user was able to view the documents by logging in to the application using their username and password. Manual uploading of the document in the user account was also successful. The Tesseract OCR was able to recognize keyword/s from the uploaded documents and the program was able to classify and assign points to the documents that are in landscape orientation and with plain background and text.

The system also provides printable outputs such as summary of points, detailed NBC CCE pointing system, detailed allocation of points acquired by the user per main criteria and subcategories, and report of criteria with insufficient points. The system will not add points if the user has already reached the maximum points on those specific criteria, but the user can still upload documents for future use. The user can also import documents using their mobile phone and computer.

For its limitations, the system is a bit faulty in reading characters since documents and certificates vary in designs, orientation, and styles resulting to the inaccurate allocation of points. The system can only detect maximum of two keywords per document. The application can only read documents that are in landscape orientation.

4.4 Project Assessment

4.4.1 File Gathering

Table 4.1 shows the number of documents accumulated for each category. For testing purposes, the proponents gathered a total of 237 documents that includes certificates of participations and attendance, diplomas, etc.

Table 4. File Gathering

Major Components	Category	Subcategory	Number of Accumulated Documents
1. Educational Qualification	1.1	1.1.2	4
		1.1.5	2
		1.2.2	1
2. Experience and Length of Service	2.3	2.3.2	1
3. Professional Development and Honors	3.1	3.1.1	2
		3.1.2	2
	3.2	3.2.1.2	169
		3.2.2.2	30
		3.2.2.3	5
		3.2.2.5	1
		3.2.2.7	7
	3.3	3.3.1	12
	3.6		1
Total			237

CHAPTER 5

SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary of Findings

The proponents were able to develop an automated system for National Budget Circular 461. The system can scan documents and store it in a cloud storage. It can also sort documents based on the keyword/s detected by the Tesseract OCR and can assign points according to the category of the document. The system has an application that can be navigated by the faculty and administrator to monitor their points for evaluation and promotion. The system can also produce printable outputs such as summary of points, detailed NBC CCE pointing system, detailed allocation of points acquired by the user per main criteria and subcategories, and report of criteria with insufficient points. Moreover, the system is also configurable since the NBC 461 criteria changes every three years.

However, the system can only function accurately on the documents with plain text and background and in landscape orientation.

5.2 Conclusion

Based on the data gathered, the proponents were able to conclude the following:

1. The system can do its primary functions: scan, save, and store documents on the database system.
2. The application is reconfigurable by the administrators since the NBC 461 guidelines changes every three years.
3. The system can automatically assign points to the sorted documents.
4. The administrator can manage the users and check if their documents are valid or not.
5. The system can only detect maximum of two keywords for each document.
6. The accumulated points for each category can be viewed and monitored by the user in his/her account.

5.3 Recommendations

The recommendations below are from the proponents' self-evaluation of the program.

1. Improve the accuracy of the system's ability to read characters as well as recognize characters in portrait orientation.
2. Improve the accuracy of the system in assigning points to some of the subcategories.
3. Increase the number of keywords the system can detect for faster categorization.

APPENDIX A: SUMMARY OF EXPENSES

Component	No. of Units	Unit Price	Price
HP Printer DJET 3776 Sea Grass	1	Php 4,260.00	Php 4,260.00
ASUS Vivo book Flip TP301UJ	1	Php 19,000.00	Php 19,000.00
Got It Fingerprint Reader Arduino Mega 2560	1	Php 2,650.00	Php 2,650.00
Customized Kiosk	1	Php 18,000.00	Php 18,000.00
TOTAL			Php 43, 910 .00

APPENDIX B: GANTT CHART

Shows the timetable for the research process flow.

ACTIVITIES	JUNE	JULY	AUGUST	SEPTEMBER	NOVEMBER	DECEMBER	JANUARY	FEBRUARY	MARCH	APRIL	MAY	JUNE	JULY	AUGUST
SHARING IDEAS														
RESEARCHING AND VISUALIZING OF THE CHOSEN TOPIC														
TOPIC DEFENSE														
BRAINSTORMING OF THE PROJECT STUDY TITLE														
CHAPTER 1														
CHAPTER 2														
CHAPTER 3														
TITLE DEFENSE														
INITIAL LAYOUT OF THE PROJECT														

CANVAS SING EQUIPME NT														
ACQUIRI NG TOOLS AND APPS														
GATHERI NG PERSON AL FILES FROM THE FACULT Y														
PROGRA MMING														
PURCHA SE OF EQUIPME NT														
PROGRES S DEFENSE														
TESTING AND DATA GATHERI NG														
PRODUC TION OF PROJECT TRAININ G MANUAL														

FOR THE COMPLE TE INSTRUC TIONS AND SPECIFIC ATIONS FOR THE FUTURE TRAININ GS														
TRAININ G FOR THE BENEFICI ARY														
PRE- FINAL DEFENSE														
CHAPTE R 4														
CHAPTE R 5														
ONLINE SYMPOSI UM														

APPENDIX C: PROGRAM CODES

Program

TesseractApp.java

```
package com.tesseract;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

@SpringBootApplication
public class TesseractApp extends SpringBootServletInitializer {

    public static void main(String[] args) {
        SpringApplication.run(TesseractApp.class, args);
    }
}
```

TesseractConfig.java

```
package com.tesseract;

import org.springframework.context.annotation.Configuration;
import org.springframework.data.jpa.repository.config.EnableJpaAuditing;
import org.springframework.transaction.annotation.EnableTransactionManagement;

@Configuration
@EnableJpaAuditing
@EnableTransactionManagement
public class TesseractConfig {
}
}
```

ServletInitializer.java


```

package com.tesseract;

import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

public class ServletInitializer extends SpringBootServletInitializer {

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder
application) {
        return application.sources(TesseractApp.class);
    }
}

```

Domain Package

BaseEntity.java

```

package com.tesseract.domain;

import java.time.LocalDateTime;

import javax.persistence.EntityListeners;
import javax.persistence.MappedSuperclass;
import javax.persistence.Version;

import org.springframework.data.annotation.CreatedDate;
import org.springframework.data.annotation.LastModifiedDate;
import org.springframework.data.jpa.domain.support.AuditingEntityListener;

import lombok.AllArgsConstructor;
import lombok.EqualsAndHashCode;
import lombok.Getter;

```

```
import lombok.NoArgsConstructor;
import lombok.Setter;
import lombok.ToString;

@MappedSuperclass
@Getter
@Setter
@AllArgsConstructor
@NoArgsConstructor
@ToString
@EqualsAndHashCode
@EntityListeners(AuditingEntityListener.class)
public abstract class BaseEntity {

    @Version
    private int version;

    @CreatedDate
    private LocalDateTime createdAt;

    @LastModifiedDate
    private LocalDateTime lastModifiedDate;

}
```

Category.java

```
package com.tesseract.domain;

import javax.persistence.Entity;
```

```

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
/**
 *
 * @author Edison Ray I. Tañala
 *
 */
@Entity
@Getter
@Setter
@Builder
@AllArgsConstructor
@NoArgsConstructor
@EqualsAndHashCode(callSuper = true)
public class Category extends IdEntity {

    private String name;

    private int points;

    @Override
    public String toString() {
        return name;
    }

    public Object getName() {
        // TODO Auto-generated method stub
        return null;
    }
}

```

```

        public void setName(String value) {
            // TODO Auto-generated method stub

        }
    }
}

```

Document.java

```

package com.tesseract.domain;

import javax.persistence.Entity;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

/**
 *
 * @author Edison Ray I. Tañala
 *
 */
@Entity
@Getter
@Setter
@Builder
@AllArgsConstructor
@NoArgsConstructor
@EqualsAndHashCode(callSuper = true)
public class Category extends IdEntity {

```

```

        private String name;

        private int points;
        @Override
        public String toString() {
            return name;
        }
        public Object getName() {
            // TODO Auto-generated method stub
            return null;
        }
        public void setName(String value) {
            // TODO Auto-generated method stub
        }
    }
}

```

IdEntity.java

```

package com.tesseract.domain;

import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.MappedSuperclass;

import lombok.AllArgsConstructor;
import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import lombok.ToString;

@MappedSuperclass

```

```

@Getter
@Setter
@AllArgsConstructor
@NoArgsConstructor
@ToString
@EqualsAndHashCode(callSuper = true)
public abstract class IdEntity extends BaseEntity {

    @Id
    @GeneratedValue
    private Long id;

}

```

Privilege.java

```

package com.tesseract.domain;

import javax.persistence.Entity;
import javax.persistence.Id;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import lombok.ToString;

@Entity
@Getter
@Setter

```

```

@Builder
@AllArgsConstructor
@NoArgsConstructor
@ToString
@EqualsAndHashCode
public class Privilege {

    @Id
    private String code;

    private String description;

    public String getCode() {
        // TODO Auto-generated method stub
        return null;
    }

}

```

Role.java

```

package com.tesseract.domain;

import javax.persistence.Entity;
import javax.persistence.Id;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

```

```

import lombok.ToString;

@Entity
@Getter
@Setter
@Builder
@AllArgsConstructor
@NoArgsConstructor
@ToString
@EqualsAndHashCode
public class Role {

    @Id
    private String code;

    private String description;

}

```

RolePrivilege.java

```

package com.tesseract.domain;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.ManyToOne;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.EqualsAndHashCode;

```



```

import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import lombok.ToString;

@Entity
@Getter
@Setter
@Builder
@AllArgsConstructor
@NoArgsConstructor
@ToString
@EqualsAndHashCode
public class RolePrivilege {

    @Id
    @GeneratedValue
    private Long id;

    @ManyToOne
    private Privilege privilege;

    @ManyToOne
    private Role role;

    public Privilege getPrivilege() {
        // TODO Auto-generated method stub
        return null;
    }
}

```

SubCategory.java

```
package com.tesseract.domain;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.ManyToOne;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

/**
 *
 * @author Edison Ray I. Tañala
 *
 */
@Entity
@Getter
@Setter
@Builder
@AllArgsConstructor
@NoArgsConstructor
@EqualsAndHashCode(callSuper = true)
public class SubCategory extends IdEntity {
    @Column(length = 2048)
    private String name;
```

```

@ManyToOne
private Category category;
private String seriesNo;
private double points;
private String tag;
@Column(length = 4096)
private String remarks;
private boolean withPoints;
@Column(length = 2048)
private String documents;

```

```

@ManyToOne
private SubCategory parent;
private int sequence;
private int level;

```

```

@ManyToOne
private SubCategorySummary subCatSummary;
@Override
public String toString() {
    return name;
}
public CharSequence getTag() {
    // TODO Auto-generated method stub
    return null;
}
public Object getPoints() {
    // TODO Auto-generated method stub
    return null;
}
public Category getCategory() {

```

```

        // TODO Auto-generated method stub
        return null;
    }

    public SubCategorySummary getSubCatSummary() {
        // TODO Auto-generated method stub
        return null;
    }

    public Object getName() {
        // TODO Auto-generated method stub
        return null;
    }

    public void setName(String value) {
        // TODO Auto-generated method stub

    }

    public void setCategory(Category value) {
        // TODO Auto-generated method stub

    }

    public void setTag(String value) {
        // TODO Auto-generated method stub

    }

    public void setPoints(int intValue) {
        // TODO Auto-generated method stub

    }
}

```

SubCategorySummary.java

```
package com.tesseract.domain;
```

```

import javax.persistence.Column;
import javax.persistence.Entity;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

/**
 *
 * @author Edison Ray I. Tañala
 *
 */
@Entity
@Getter
@Setter
@Builder
@AllArgsConstructor
@NoArgsConstructor
@EqualsAndHashCode(callSuper = true)
public class SubCategorySummary extends IdEntity {
    @Column(length = 2048)
    private String name;

    private String seriesNo;
    private double points;
    private String tag;

```

```

        private boolean        boldStyle;

        private int sequence;
        private int level;

        @Override
        public String toString() {
            return name;
        }
    }
}

```

Training.java

```

package com.tesseract.domain;

import javax.persistence.Entity;
import javax.persistence.ManyToOne;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

/**
 *
 * @author Edison Ray I. Tañala
 *
 */
@Entity
@Getter

```

```

@Setter
@Builder
@AllArgsConstructor
@NoArgsConstructor
@EqualsAndHashCode(callSuper = true)
public class Training extends IdEntity {

    private String name;
    private String trainer;
    private String contactPerson;
    private String contactNumber;
    private String address;
    private String email;
    @ManyToOne
    private Category category;
    private String status;
    @Override
    public String toString() {
        return name;
    }
    public Category getCategory() {
        // TODO Auto-generated method stub
        return null;
    }
    public Object getName() {
        // TODO Auto-generated method stub
        return null;
    }
    public Object getContactNumber() {
        // TODO Auto-generated method stub
        return null;
    }

```

```

    }
    public Object getContactPerson() {
        // TODO Auto-generated method stub
        return null;
    }
    public void setName(String value) {
        // TODO Auto-generated method stub

    }
    public void setCategory(Category value) {
        // TODO Auto-generated method stub

    }
    public void setAddress(String value) {
        // TODO Auto-generated method stub

    }
    public void setContactPerson(String value) {
        // TODO Auto-generated method stub

    }
    public void setContactNumber(String value) {
        // TODO Auto-generated method stub

    }
    public void setEmail(String value) {
        // TODO Auto-generated method stub

    }
    public String getAddress() {
        // TODO Auto-generated method stub

```



```

        return null;
    }
    public String getEmail() {
        // TODO Auto-generated method stub
        return null;
    }
}

```

User.java

```

package com.tesseract.domain;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.UniqueConstraint;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import lombok.ToString;

@Entity
@Table(uniqueConstraints = @UniqueConstraint(columnNames = { "username" }))
@Getter
@Setter
@Builder

```

```

@AllArgsConstructor
@NoArgsConstructor
@ToString
@EqualsAndHashCode
public class User {

    @Id
    @GeneratedValue
    private Long id;

    private String username;

    private String name;

    private String email;

    @Builder.Default
    private boolean active = true;

    private boolean admin;

    private String passwordHash;

    public String getPasswordHash;

    public String getName() {
        // TODO Auto-generated method stub
        return null;
    }

    public String getPasswordHash() {

```

```

        // TODO Auto-generated method stub
        return null;
    }

    public Object getId() {
        // TODO Auto-generated method stub
        return null;
    }

    public void setPasswordHash(String encode) {
        // TODO Auto-generated method stub

    }

    public void setAdmin(boolean b) {
        // TODO Auto-generated method stub

    }
}

```

UserCredit.java

```

package com.tesseract.domain;

import javax.persistence.Entity;
import javax.persistence.ManyToOne;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.EqualsAndHashCode;
import lombok.Getter;

```

```

import lombok.NoArgsConstructor;
import lombok.Setter;

/**
 *
 * @author Edison Ray I. Tañala
 *
 */
@Entity
@Getter
@Setter
@Builder
@AllArgsConstructor
@NoArgsConstructor
@EqualsAndHashCode(callSuper = true)
public class UserCredit extends IdEntity {

    @ManyToOne
    private Category category;

    private double points;

    @ManyToOne
    private User user;

    @Override
    public String toString() {
        return user.getName();
    }

    public void setCategory(Category cat) {
        // TODO Auto-generated method stub

```

```

    }
    public void setUser(User user2) {
        // TODO Auto-generated method stub

    }
    public Document getCategory() {
        // TODO Auto-generated method stub
        return null;
    }
    public Object getPoints() {
        // TODO Auto-generated method stub
        return null;
    }
    public void setPoints(double accumulatedPoints) {
        // TODO Auto-generated method stub

    }
}

```

UserCreditDetail.java

```

package com.tesseract.domain;

import javax.persistence.Entity;
import javax.persistence.ManyToOne;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

```

```

/**
 *
 * @author Edison Ray I. Tañala
 *
 */
@Entity
@Getter
@Setter
@Builder
@AllArgsConstructor
@NoArgsConstructor
@EqualsAndHashCode(callSuper = true)
public class UserCreditDetail extends IdEntity {

    @ManyToOne
    private SubCategory subCategory;

    private double points;
    private double points1;
    private double points2;
    private double points3;
    private double points4;

    @ManyToOne
    private User user;

    @Override
    public String toString() {
        return user.getName();
    }

    public void setSubCategory(SubCategory cat) {

```

```

        // TODO Auto-generated method stub

    }

    public void setUser(User user2) {
        // TODO Auto-generated method stub

    }

    public char[] getId() {
        // TODO Auto-generated method stub
        return null;
    }

    public double getPoints1() {
        // TODO Auto-generated method stub
        return 0;
    }

    public void setPoints1(double accumulatedPoints) {
        // TODO Auto-generated method stub

    }
}

```

UserCreditSummary.java

```

package com.tesseract.domain;

import javax.persistence.Entity;
import javax.persistence.ManyToOne;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.EqualsAndHashCode;
import lombok.Getter;

```

```

import lombok.NoArgsConstructor;
import lombok.Setter;

/**
 *
 * @author Edison Ray I. Tañala
 *
 */
@Entity
@Getter
@Setter
@Builder
@AllArgsConstructor
@NoArgsConstructor
@EqualsAndHashCode(callSuper = true)
public class UserCreditSummary extends IdEntity {

    @ManyToOne
    private SubCategorySummary subCategorySummary;

    private double points1;

    @ManyToOne
    private User user;

    @Override
    public String toString() {
        return user.getName();
    }

    public void setSubCategorySummary(SubCategorySummary cat) {
        // TODO Auto-generated method stub
    }
}

```



```

    }
    public void setUser(User user2) {
        // TODO Auto-generated method stub

    }
    public char[] getId() {
        // TODO Auto-generated method stub
        return null;
    }
    public double getPoints1() {
        // TODO Auto-generated method stub
        return 0;
    }
    public void setPoints1(double accumulatedPoints) {
        // TODO Auto-generated method stub

    }
}

```

UserLogin.java

```

package com.tesseract.domain;

import java.time.LocalDateTime;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.ManyToOne;

import lombok.AllArgsConstructor;
import lombok.Builder;

```

```

import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import lombok.ToString;

/**
 * @author Franklin Chua
 *
 */
@Entity
@Getter
@Setter
@Builder
@AllArgsConstructor
@NoArgsConstructor
@ToString
@EqualsAndHashCode
public class UserLogin {

    @Id
    @GeneratedValue
    private Long id;

    private String deviceCode;

    @ManyToOne
    private User user;

    @Builder.Default
    private LocalDateTime loginDate = LocalDateTime.now();

```

```
}
```

UserRole.java

```
package com.tesseract.domain;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.ManyToOne;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import lombok.ToString;

/**
 * @author Franklin Chua
 *
 */
@Entity
@Getter
@Setter
@Builder
@AllArgsConstructor
@NoArgsConstructor
@ToString
@EqualsAndHashCode
public class UserRole {
```

```

        @Id
        @GeneratedValue
        private Long id;

        @ManyToOne
        private User user;

        @ManyToOne
        private Role role;

        public Object getRole() {
            // TODO Auto-generated method stub
            return null;
        }
    }
}

```

Exception Package

InvalidPasswordException.java

```

package com.tesseract.exception;

@SuppressWarnings("serial")
public class InvalidPasswordException extends RuntimeException {

}

```

UserNotFoundException.java

```

package com.tesseract.exception;

@SuppressWarnings("serial")

```

```
public class UserNotFoundException extends RuntimeException {  
  
}
```

Layout Package

ExportUtil.java

```
package com.tesseract.layout;  
  
import java.io.IOException;  
import java.sql.Connection;  
import java.sql.SQLException;  
import java.util.HashMap;  
import java.util.Map;  
  
import javax.sql.DataSource;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.ApplicationContext;  
import org.springframework.core.io.Resource;  
import org.springframework.stereotype.Service;  
  
import com.tesseract.domain.User;  
  
import net.sf.jasperreports.engine.JRException;  
import net.sf.jasperreports.engine.JasperExportManager;  
import net.sf.jasperreports.engine.JasperFillManager;  
import net.sf.jasperreports.engine.JasperPrint;  
  
@Service  
public class ExportUtil {
```

```

    @Autowired
    private DataSource ds;

    @Autowired
    private ApplicationContext appContext;

    public byte[] generate(User o) {
        Resource res = appContext.getResource("classpath:reports/tesseract-
summary.jasper");
        if(res.exists()) {
            try {
                Connection conn = ds.getConnection();
                JasperPrint jPrint =
JasperFillManager.fillReport(res.getInputStream(), init(o), conn);
                byte[] data =
JasperExportManager.exportReportToPdf(jPrint);
                conn.close();
                return data;
            } catch (JRException | SQLException e) {
                System.out.println("JRE");
                e.getMessage();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        } else {
            System.out.println("No file found");
        }
        return null;
    }

    public byte[] generateDetail(User o) {

```

```

        Resource res = appContext.getResource("classpath:reports/tesseract-
detail.jasper");
        if(res.exists()) {
            try {
                Connection conn = ds.getConnection();
                JasperPrint jPrint =
JasperFillManager.fillReport(res.getInputStream(), init(o), conn);
                byte[] data =
JasperExportManager.exportReportToPdf(jPrint);
                conn.close();
                return data;
            } catch (JRException | SQLException e) {
                System.out.println("JRE");
                e.getMessage();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        } else {
            System.out.println("No file found");
        }
        return null;
    }

    private Map<String, Object> init(User entity) {
        Map<String, Object> params = new HashMap<String, Object>();
        params.put("ID", entity.getId());
        return params;
    }

    private Map<String, Object> init(User entity, double total) {
        Map<String, Object> params = new HashMap<String, Object>();
        params.put("ID", entity.getId());

```

```

        params.put("TOTAL", total);
        return params;
    }

    public byte[] generateDetailSummary(User o, double total) {

        Resource res = appContext.getResource("classpath:reports/tesseract-
detail-summary.jasper");

        if(res.exists()) {
            try {
                Connection conn = ds.getConnection();

                JasperPrint jPrint =
JasperFillManager.fillReport(res.getInputStream(), init(o,total), conn);
                byte[] data =
JasperExportManager.exportReportToPdf(jPrint);
                conn.close();
                return data;
            } catch (JRException | SQLException e) {
                System.out.println("JRE");
                e.getMessage();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        } else {
            System.out.println("No file found");
        }
        return null;
    }

    public byte[] generateDetailDanger(User user) {

```



```

Resource res = appContext.getResource("classpath:reports/tesseract-danger.jasper");
    if(res.exists()) {
        try {
            Connection conn = ds.getConnection();

            JasperPrint jPrint =
JasperFillManager.fillReport(res.getInputStream(), init(user), conn);
            byte[] data =
JasperExportManager.exportReportToPdf(jPrint);
            conn.close();
            return data;
        } catch (JRException | SQLException e) {
            System.out.println("JRE");
            e.getMessage();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    } else {
        System.out.println("No file found");
    }
    return null;
}
}

```

MainAppLayout.java

```

package com.tesseract.layout;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

```

```

import com.tesseract.domain.User;
import com.tesseract.view.CategoryView;
import com.tesseract.view.DocumentView;
import com.tesseract.view.LoginView;
import com.tesseract.view.SubCategoryView;
import com.tesseract.view.TrainingView;
import com.tesseract.view.UserCreditView;
import com.tesseract.view.UsersView;
import com.vaadin.flow.component.applayout.AppLayout;
import com.vaadin.flow.component.applayout.DrawerToggle;
import com.vaadin.flow.component.dependency.CssImport;
import com.vaadin.flow.component.html.Div;
import com.vaadin.flow.component.html.Image;
import com.vaadin.flow.component.html.Label;
import com.vaadin.flow.component.tabs.Tab;
import com.vaadin.flow.component.tabs.Tabs;
import com.vaadin.flow.component.tabs.Tabs.Orientation;
import com.vaadin.flow.component.tabs.TabsVariant;
import com.vaadin.flow.router.AfterNavigationEvent;
import com.vaadin.flow.router.AfterNavigationObserver;
import com.vaadin.flow.router.BeforeEnterEvent;
import com.vaadin.flow.router.BeforeEnterObserver;
import com.vaadin.flow.router.RouterLink;
import com.vaadin.flow.server.VaadinSession;
import com.vaadin.flow.theme.Theme;
import com.vaadin.flow.theme.lumo.Lumo;
import com.vaadin.flow.theme.material.Material;

/**
 *
 * @author Edison Ray I. Tañala

```

```

*
*/
@SuppressWarnings("serial")
//@StyleSheet("/styles/styles.css")
@CssImport("./styles/styles.css")
//@Theme(value =Lumo.class, variant = Lumo.DARK)
public class MainAppLayout extends AppLayout implements AfterNavigationObserver,
BeforeEnterObserver{

    protected static Logger logger =
LoggerFactory.getLogger(MainAppLayout.class);

    private Tabs tabs;
    private Div divDrawer;
    public MainAppLayout() {
        divDrawer = new Div();
        tabs = new Tabs();
        tabs.setOrientation(Orientation.VERTICAL);
        tabs.addThemeVariants(TabsVariant.LUMO_SMALL);

        addToNavbar(new DrawerToggle(), new Label("iPromote Application"));
        Image image = new Image("images/logo.png", "logo");
        image.addClassName("drawer-logo");
        addToDrawer(image);
        addToDrawer(divDrawer);
    }

    @Override
    public void afterNavigation(AfterNavigationEvent event) {
        // TODO Auto-generated method stub
        divDrawer.removeAll();
    }
}

```

```

        tabs.removeAll();
        User user = (User) VaadinSession.getCurrent().getAttribute("user");

        tabs.add(new Tab(new RouterLink("Import PDF",
DocumentView.class)));
        tabs.add(new Tab(new RouterLink("Employee Credit View",
UserCreditView.class)));

        if(user.isAdmin()) {
            tabs.add(new Tab(new RouterLink("Category View",
CategoryView.class)));
            tabs.add(new Tab(new RouterLink("Sub Category View",
SubCategoryView.class)));
            tabs.add(new Tab(new RouterLink("Training View",
TrainingView.class)));
            tabs.add(new Tab(new RouterLink("Users View",
UsersView.class)));
        }

        tabs.add(new Tab(new RouterLink("Logout", LogoutView.class)));
        divDrawer.add(tabs);
    }

    @Override
    public void beforeEnter(BeforeEnterEvent event) {
        // TODO Auto-generated method stub
        divDrawer.removeAll();
    }
}

```

Repository Package

CategoryRepository.java

```
package com.tesseract.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import com.tesseract.domain.Category;

public interface CategoryRepository extends JpaRepository<Category, Long> {
}
```

DocumentRepository.java

```
package com.tesseract.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;

import com.tesseract.domain.Document;
import com.tesseract.domain.User;

public interface DocumentRepository extends JpaRepository<Document, Long> {

    @Query("Select u from Document u where u.user = :user and (:tag is null or :tag = "
        + " or u.tag like concat('%',:tag,'%'))")
    List<Document> findAll(@Param("tag") String tags,
                           @Param("user") User user);

    Document findByData(byte[] data);

    Document findByFilename(String filename);
}
```

```
}
```

PrivilegeRepository.java

```
package com.tesseract.repository;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import com.tesseract.domain.Privilege;
```

```
public interface PrivilegeRepository extends JpaRepository<Privilege, String> {  
}
```

RolePrivilegeRepository.java

```
package com.tesseract.repository;
```

```
import java.util.List;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import com.tesseract.domain.Role;
```

```
import com.tesseract.domain.RolePrivilege;
```

```
public interface RolePrivilegeRepository extends JpaRepository<RolePrivilege, Long> {  
    List<RolePrivilege> findByRole(Object object);  
}
```

RoleRepository.java

```
package com.tesseract.repository;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import com.tesseract.domain.Role;
```

```
public interface RoleRepository extends JpaRepository<Role, String> {  
}
```

SubCategoryRepository.java

```
package com.tesseract.repository;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import com.tesseract.domain.SubCategory;
```

```
public interface SubCategoryRepository extends JpaRepository<SubCategory, Long> {  
}
```

SubCategorySummaryRepository.java

```
package com.tesseract.repository;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import com.tesseract.domain.SubCategorySummary;
```

```
public interface SubCategorySummaryRepository extends  
JpaRepository<SubCategorySummary, Long> {  
}
```

TrainingRepository.java

```
package com.tesseract.repository;
```

```
import java.util.List;
```

```

import org.springframework.data.jpa.repository.JpaRepository;

import com.tesseract.domain.Category;
import com.tesseract.domain.Training;

public interface TrainingRepository extends JpaRepository<Training, Long> {
    List<Training> findByCategory(Category category);
}

```

UserCreditDetailRepository.java

```

package com.tesseract.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;

import com.tesseract.domain.SubCategory;
import com.tesseract.domain.User;
import com.tesseract.domain.UserCreditDetail;

public interface UserCreditDetailRepository extends JpaRepository<UserCreditDetail,
Long> {
    @Query("Select u from UserCreditDetail u where u.user = :user")
    List<UserCreditDetail> findAll(@Param("user") User user);
    UserCreditDetail findBySubCategoryAndUser(SubCategory category, User user);
}

```


UserCreditRepository.java

```
package com.tesseract.repository;
```

```
import java.util.List;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import org.springframework.data.jpa.repository.Query;
```

```
import org.springframework.data.repository.query.Param;
```

```
import com.tesseract.domain.Category;
```

```
import com.tesseract.domain.User;
```

```
import com.tesseract.domain.UserCredit;
```

```
public interface UserCreditRepository extends JpaRepository<UserCredit, Long> {
```

```
    @Query("Select u from UserCredit u where u.user = :user")
```

```
    List<UserCredit> findAll(@Param("user") User user);
```

```
    UserCredit findByCategoryAndUser(Category category, User user);
```

```
    @Query("Select u from UserCredit u where u.user = :user and u.id in(99,100) and  
u.category.points = 85")
```

```
    List<UserCredit> findAllIn(@Param("user") User user);
```

```
}
```

UserCreditSummaryRepository.java

```
package com.tesseract.repository;
```

```
import java.util.List;
```

```

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;

import com.tesseract.domain.SubCategorySummary;
import com.tesseract.domain.User;
import com.tesseract.domain.UserCreditSummary;

public interface UserCreditSummaryRepository extends
JpaRepository<UserCreditSummary, Long> {

    @Query("Select u from UserCreditSummary u where u.user = :user")
    List<UserCreditSummary> findAll(@Param("user") User user);

    UserCreditSummary
findBySubCategorySummaryAndUser(SubCategorySummary
subCategorySummary,User user);

}

```

UserLoginRepository.java

```

package com.tesseract.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import com.tesseract.domain.UserLogin;

public interface UserLoginRepository extends JpaRepository<UserLogin, Long> {

}

```

UserRepository.java

```
package com.tesseract.repository;

import java.util.List;
import java.util.Optional;

import org.springframework.data.jpa.repository.JpaRepository;
import com.tesseract.domain.User;

public interface UserRepository extends JpaRepository<User, Long> {
    User findByUsername(String username);
    List<User> findByActive(boolean active);
}
```

UserRoleRepository.java

```
package com.tesseract.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import com.tesseract.domain.User;
import com.tesseract.domain.UserRole;

public interface UserRoleRepository extends JpaRepository<UserRole, Long> {
    List<UserRole> findByUser(User user);
}
```

Service Package

CategoryService.java

```

package com.tesseract.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.tesseract.domain.Category;
import com.tesseract.repository.CategoryRepository;

@Service
public class CategoryService {
    @Autowired
    private CategoryRepository repo;
    public void save(Category entity) {
        repo.save(entity);
    }
    public List<Category> findAll() {
        return repo.findAll();
    }
    public void delete(Category entity) {
        repo.delete(entity);
    }
}

```

DocumentService.java

```

package com.tesseract.service;

import java.util.List;

```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
```

```
import com.tesseract.domain.Document;
import com.tesseract.domain.User;
import com.tesseract.repository.DocumentRepository;
```

```
@Service
```

```
public class DocumentService {
    @Autowired
    private DocumentRepository docRepo;
    public void save(Document doc) {
        docRepo.save(doc);
    }
    public List<Document> findAll(String tags, User user) {
        return docRepo.findAll(tags, user);
    }
    public Document findByData(byte[] data) {
        return docRepo.findByData(data);
    }
    public Document findByFilename(String filename) {
        return docRepo.findByFilename(filename);
    }
}
```

SubCategoryService.java

```
package com.tesseract.service;
```

```
import java.util.List;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```

import org.springframework.stereotype.Service;

import com.tesseract.domain.SubCategory;
import com.tesseract.repository.SubCategoryRepository;

@Service
public class SubCategoryService {
    @Autowired
    private SubCategoryRepository repo;
    public void save(SubCategory entity) {
        repo.save(entity);
    }
    public List<SubCategory> findAll() {
        return repo.findAll();
    }
    public void delete(SubCategory entity) {
        repo.delete(entity);
    }
}

```

SubCategorySummaryService.java

```

package com.tesseract.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.tesseract.domain.SubCategorySummary;
import com.tesseract.repository.SubCategorySummaryRepository;

```

```

@Service
public class SubCategorySummaryService {
    @Autowired
    private SubCategorySummaryRepository repo;
    public void save(SubCategorySummary entity) {
        repo.save(entity);
    }
    public List<SubCategorySummary> findAll() {
        return repo.findAll();
    }
    public void delete(SubCategorySummary entity) {
        repo.delete(entity);
    }
}

```

SystemService.java

```

package com.tesseract.service;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import com.tesseract.domain.User;
import com.tesseract.domain.UserLogin;
import com.tesseract.exception.InvalidPasswordException;
import com.tesseract.exception.UserNotFoundException;
import com.tesseract.repository.UserLoginRepository;

```

```

import com.tesseract.repository.UserRepository;

@Service
public class SystemService {
    @Autowired
    private UserRepository userRepo;
    private PasswordEncoder passwordEncoder = new BCryptPasswordEncoder();
    @Transactional
    public User login(String username, String password, String deviceCode) {
        // validate username
        User user = userRepo.findByUsername(username);
        if (user == null) {
            throw new UserNotFoundException();
        }
        // validate password
        if (!passwordEncoder.matches(password, user.getPasswordHash())) {
            //      saveDeviceLogin(deviceCode, user, LoginResult.FAILURE);
            throw new InvalidPasswordException();
        }
        //      saveDeviceLogin(deviceCode, user, LoginResult.SUCCESS);
        return user;
    }
    @Transactional
    public User login(String username, String password) {
        // validate username
        User user = userRepo.findByUsername(username);
        if (user == null) {
            throw new UserNotFoundException();
        }
        // validate password
        if (!passwordEncoder.matches(password, user.getPasswordHash())) {

```



```

        throw new InvalidPasswordException();
    }
    return user;
}
}

```

TrainingService.java

```

package com.tesseract.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.tesseract.domain.Category;
import com.tesseract.domain.Document;
import com.tesseract.domain.Training;
import com.tesseract.repository.TrainingRepository;

@Service
public class TrainingService {
    @Autowired
    private TrainingRepository repo;

    public void save(Training entity) {
        repo.save(entity);
    }
    public List<Training> findAll() {
        return repo.findAll();
    }
    public List<Training> findByCategory(Category category){

```

```

        return repo.findByCategory(category);
    }
    public void delete(Training entity) {
        repo.delete(entity);
    }
    public List<Training> findByCategory(Document category) {
        // TODO Auto-generated method stub
        return null;
    }
}

```

UserCreditDetailService.java

```

package com.tesseract.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.tesseract.domain.SubCategory;
import com.tesseract.domain.User;
import com.tesseract.domain.UserCreditDetail;
import com.tesseract.repository.UserCreditDetailRepository;

@Service
public class UserCreditDetailService {
    @Autowired
    private UserCreditDetailRepository repo;
    @Autowired
    private SubCategoryService subService;
    public void save(UserCreditDetail entity) {
        repo.save(entity);
    }
}

```

```

    }
    public List<UserCreditDetail> findAll(User user) {
        return repo.findAll(user);
    }
    public void createCreditDetail(User user) {
        for(SubCategory cat : subService.findAll()) {
            UserCreditDetail uc = new UserCreditDetail();
            uc.setSubCategory(cat);
            uc.setUser(user);
            repo.save(uc);
        }
    }
    public UserCreditDetail findBySubCategory(SubCategory sub,User user) {
        return repo.findBySubCategoryAndUser(sub,user);
    }
}

```

UserCreditService.java

```

package com.tesseract.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.tesseract.domain.Category;
import com.tesseract.domain.SubCategory;
import com.tesseract.domain.User;
import com.tesseract.domain.UserCredit;
import com.tesseract.repository.UserCreditRepository;

```

```

@Service
public class UserCreditService {
    @Autowired
    private UserCreditRepository repo;
    @Autowired
    private SubCategoryService subService;
    @Autowired
    private CategoryService catService;
    public void save(UserCredit entity) {
        repo.save(entity);
    }
    public List<UserCredit> findAll(User user) {
        return repo.findAll(user);
    }
    public List<UserCredit> findAllIn(User user) {
        return repo.findAllIn(user);
    }
    public void createCredit(User user) {
        for(Category cat : catService.findAll()) {
            UserCredit uc = new UserCredit();
            uc.setCategory(cat);
            uc.setUser(user);
            repo.save(uc);
        }
    }
    public UserCredit findByCategory(Category category,User user) {
        return repo.findByCategoryAndUser(category,user);
    }
}

```

UserCreditSummaryService.java

```

package com.tesseract.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.tesseract.domain.SubCategorySummary;
import com.tesseract.domain.User;
import com.tesseract.domain.UserCreditSummary;
import com.tesseract.repository.UserCreditSummaryRepository;

@Service
public class UserCreditSummaryService {

    @Autowired
    private UserCreditSummaryRepository repo;
    @Autowired
    private SubCategorySummaryService subService;
    public void save(UserCreditSummary entity) {
        repo.save(entity);
    }
    public List<UserCreditSummary> findAll(User user) {
        return repo.findAll(user);
    }
    public void createCreditSummary(User user) {
        for(SubCategorySummary cat : subService.findAll()) {
            UserCreditSummary uc = new UserCreditSummary();
            uc.setSubCategorySummary(cat);
        }
    }
}

```

```

        uc.setUser(user);
        repo.save(uc);
    }
}

public UserCreditSummary findBySubCategorySummary(SubCategorySummary
sub,User user) {
    return repo.findBySubCategorySummaryAndUser(sub,user);
}
}

```

UserService.java

```

package com.tesseract.service;

import java.util.ArrayList;
import java.util.List;
import java.util.Optional;
import java.util.Set;
import java.util.TreeSet;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.cache.annotation.Cacheable;
import org.springframework.data.domain.Sort;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.stereotype.Service;

import com.tesseract.domain.Privilege;
import com.tesseract.domain.Role;
import com.tesseract.domain.RolePrivilege;
import com.tesseract.domain.User;
import com.tesseract.domain.UserRole;

```

```

import com.tesseract.exception.UserNotFoundException;
import com.tesseract.repository.RolePrivilegeRepository;
import com.tesseract.repository.UserRepository;
import com.tesseract.repository.UserRoleRepository;

@Service
public class UserService {
    @Autowired
    private UserRepository userRepo;
    @Autowired
    private UserRoleRepository userRoleRepo;
    @Autowired
    private RolePrivilegeRepository rolePrivRepo;

    private PasswordEncoder passwordEncoder = new BCryptPasswordEncoder();
    @Autowired
    private UserCreditService userCreditService;
    @Autowired
    private UserCreditDetailService userCreditDetailService;
    @Autowired
    private UserCreditSummaryService userCreditSummaryService;
    @Cacheable("users")
    public User findById(long id) {
        return
userRepo.findById(id).orElseThrow(UserNotFoundException::new);
    }
    @Cacheable("users")
    public List<User> findAll() {
        return userRepo.findAll(Sort.by("name"));
    }
    public User save(User user, CharSequence password) {

```

```

        User u;
        if (password != null) {
            user.setPasswordHash(passwordEncoder.encode(password));
        }
        u = userRepo.save(user);
        if(userCreditService.findAll(user).size() ==0) {
            userCreditService.createCredit(user);

        }
        if(userCreditDetailService.findAll(user).size() ==0) {
            userCreditDetailService.createCreditDetail(user);

        }
        if(userCreditSummaryService.findAll(user).size() ==0) {
            userCreditSummaryService.createCreditSummary(user);
        }
        return u;
    }

    public void changePassword(User user, CharSequence password) {
        Optional<User> o = userRepo.findById((Long) user.getId());
        if (o.isPresent()) {
            User u = o.get();
            u.setPasswordHash(passwordEncoder.encode(password));
            userRepo.save(user);
        }
    }

    public void delete(User user) {
        userRepo.delete(user);
    }

    public User findByUsername(String username) {
        return userRepo.findByUsername(username);
    }

```



```

    }

    public List<String> getUserRoles(User user) {
        List<UserRole> userRoles = userRoleRepo.findByUser(user);
        List<String> roles = new ArrayList<String>();
        for (UserRole userRole : userRoles) {
            roles.add(userRole.getRole().hashCode(), null);
        }
        return roles;
    }

    public List<String> getUserPrivileges(User user) {
        List<UserRole> userRoles = userRoleRepo.findByUser(user);
        Set<String> privileges = new TreeSet<String>();
        for (UserRole userRole : userRoles) {
            for (Privilege privilege : getRolePrivileges(userRole.getRole())) {
                privileges.add(privilege.getCode());
            }
        }
        return new ArrayList<String>(privileges);
    }

    public List<Privilege> getRolePrivileges(Object object) {
        // This is a candidate for in-memory caching.
        List<RolePrivilege> rolePrivileges = rolePrivRepo.findByRole(object);
        List<Privilege> privileges = new ArrayList<Privilege>();
        for (RolePrivilege rolePrivilege : rolePrivileges) {
            privileges.add(rolePrivilege.getPrivilege());
        }
        return privileges;
    }
}

```

View Package

CategoryView.java

```
package com.tesseract.view;

import java.util.List;

import javax.annotation.PostConstruct;

import org.springframework.beans.factory.annotation.Autowired;

import com.tesseract.domain.Category;
import com.tesseract.domain.UserCredit;
import com.tesseract.layout.MainAppLayout;
import com.tesseract.service.CategoryService;
import com.vaadin.flow.component.Text;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.dialog.Dialog;
import com.vaadin.flow.component.grid.Grid;
import com.vaadin.flow.component.html.Anchor;
import com.vaadin.flow.component.orderedlayout.HorizontalLayout;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.component.textfield.TextField;
import com.vaadin.flow.router.Route;

@Route(value = "category", layout = MainAppLayout.class)
public class CategoryView extends VerticalLayout {

    private static final long serialVersionUID = -7349714164334467197L;

    @Autowired
    private CategoryService service;
```

```

        private Grid<Category> grid;
        private HorizontalLayout layoutActions;
        private Button buttonAdd;
        private Button buttonPrint;
        private Button buttonApprove;
        private TextField textTag;
        private Button buttonSearch;
        private HorizontalLayout layoutFilter;
        private Dialog updateDialog = new Dialog();
        private Dialog detailDialog = new Dialog();

        private Anchor anchor;
//        @Autowired
//        private ExportUtil exportUtil;
        public CategoryView() {
            grid = new Grid<>(Category.class);
            add(grid);
        }
        @PostConstruct
        private void init() {
            grid.removeAllColumns();
            grid.addColumn(o ->
o.getName()).setHeader("Name").setAutoWidth(true);
            grid.setWidthFull();
//            date = new DatePicker("Date");
//
            textTag = new TextField("Insert keyword here");
            textTag.setWidth("100%");
//            comboStatus = new ComboBox<String>("Status",statusUtil.getStatus());
//            comboStatus.setValue("Pending");
            buttonSearch = new Button("Search");

```

```

buttonSearch.setWidth("100px");
buttonSearch.addClickListener(e->{
    loadData();
});
buttonAdd = new Button("Add");
buttonAdd.addClickListener(event -> {
    Dialog d = new Dialog();
    VerticalLayout content = new VerticalLayout();
    Category entity = new Category();
    TextField txtName= new TextField("Name");
    Button save = new Button("Save");
    save.addClickListener(e->{
        entity.setName(txtName.getValue());
        service.save(entity);
        d.close();
        loadData();
    });
    content.add(txtName,save);
    d.add(content);
    d.open();
});
Button edit = new Button("Edit");
edit.addClickListener(event -> {
    Dialog d = new Dialog();
    VerticalLayout content = new VerticalLayout();
    Category entity = grid.getSelectedItems().iterator().next();
    TextField txtName= new TextField("Name");
    Button save = new Button("Save");
    if (entity != null) {
        txtName.setValue((String) entity.getName());
    }
});

```

```

        save.addClickListener(e->{
            entity.setName(txtName.getValue());
            service.save(entity);
            d.close();
            loadData();
        });
    }
    content.add(txtName,save);
    d.add(content);
    d.open();
});

Button delete = new Button("Delete");
delete.addClickListener(event -> {
    Category entity = grid.getSelectedItems().iterator().next();
    service.delete(entity);
    loadData();
});

layoutActions = new HorizontalLayout();
layoutActions.add(buttonAdd,edit,delete);
add(layoutActions);

layoutFilter = new HorizontalLayout();
layoutFilter.add(textTag,buttonSearch);
layoutFilter.setDefaultVerticalComponentAlignment(Alignment.END);
layoutFilter.setWidthFull();
addComponentAtIndex(0, layoutFilter);

loadData();

```

```

    }

    private void loadData() {

        List<Category> list = service.findAll();
        grid.setItems(list);
    }

}

```

DocumentView.java

```

package com.tesseract.view;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.InputStream;
import java.util.List;

import javax.annotation.PostConstruct;

import org.apache.pdfbox.multipdf.Splitter;
import org.apache.pdfbox.pdmodel.PDDocument;
import org.springframework.beans.factory.annotation.Autowired;

import com.tesseract.domain.Document;
import com.tesseract.domain.SubCategory;
import com.tesseract.domain.SubCategorySummary;

```

```
import com.tesseract.domain.User;
import com.tesseract.domain.UserCredit;
import com.tesseract.domain.UserCreditDetail;
import com.tesseract.domain.UserCreditSummary;
import com.tesseract.layout.MainAppLayout;
import com.tesseract.service.DocumentService;
import com.tesseract.service.SubCategoryService;
import com.tesseract.service.UserCreditDetailService;
import com.tesseract.service.UserCreditService;
import com.tesseract.service.UserCreditSummaryService;
import com.tesseract.service.UserService;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.dialog.Dialog;
import com.vaadin.flow.component.grid.Grid;
import com.vaadin.flow.component.html.Anchor;
import com.vaadin.flow.component.notification.Notification;
import com.vaadin.flow.component.notification.NotificationVariant;
import com.vaadin.flow.component.orderedlayout.HorizontalLayout;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.component.textfield.TextField;
import com.vaadin.flow.component.upload.Upload;
import com.vaadin.flow.component.upload.receivers.MemoryBuffer;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.StreamRegistration;
import com.vaadin.flow.server.StreamResource;
import com.vaadin.flow.server.VaadinSession;

import net.sourceforge.tess4j.Tesseract;
```

```

@Route(value = "documents", layout = MainAppLayout.class)
public class DocumentView extends VerticalLayout {

    /**
     *
     */
    private static final long serialVersionUID = -7349714164334467197L;

    @Autowired
    private DocumentService service;

    private Grid<Document> grid;

    private HorizontalLayout layoutActions;
    private Button buttonAdd;
    private Button buttonPrint;
    private Button buttonApprove;

    private TextField textTag;
    //private ComboBox<String> comboStatus;
    private Button buttonSearch;
    private HorizontalLayout layoutFilter;

    private Dialog updateDialog = new Dialog();
    private Dialog detailDialog = new Dialog();

    private Anchor anchor;

```



```

    @Autowired
    private SubCategoryService subService;

    @Autowired
    private UserCreditService userCreditService;

    @Autowired
    private UserCreditDetailService detailService;

    @Autowired
    private UserCreditSummaryService summaryService;

    public DocumentView() {
        grid = new Grid<>(Document.class);
        add(grid);
    }

    @PostConstruct
    private void init() {
        grid.removeAllColumns();
        grid.addColumn(o -> o.getId()).setHeader("Code").setAutoWidth(true);
        grid.addColumn(o ->
o.getPoints()).setHeader("Points").setAutoWidth(true);

        grid.addColumn(o ->
o.getStatus()).setHeader("Status").setAutoWidth(true);

        grid.setWidthFull();

        //         date = new DatePicker("Date");

```

```

//
        textTag = new TextField("Insert keyword here");
        textTag.setWidth("100%");
//
        comboStatus = new ComboBox<String>("Status",statusUtil.getStatus());
//
        comboStatus.setValue("Pending");
        buttonSearch = new Button("Search");
        buttonSearch.setWidth("100px");
        buttonSearch.addClickListener(e->{
            loadData();
        });
        buttonAdd = new Button("Upload");
        buttonAdd.addClickListener(event -> {
            Dialog d = new Dialog();

            MemoryBuffer buffer = new MemoryBuffer();

            Upload upload = new Upload(buffer);
            upload.setMaxFileSize(100000000);

            upload.addSucceededListener(e -> {
                onOCR( buffer.getInputStream(),buffer.getFileName());
            //buffer.get
//
                Component component =
                createComponent(event.getMIMEType(),
                //
                    event.getFileName(), buffer.getInputStream());
                //  showOutput(event.getFileName(), component, );
            });
            d.add(upload);
            d.open();
//
            onOCR();
//
            GMPChecklistWindow window = new GMPChecklistWindow();

```

```

//          updateDialog = window.init();
//          window.EditWindow("GMP Checklist Details");
//
window.load(grid.getSelectedItems().iterator().next(),gmpService,this);
//          updateDialog.open();
//      });

buttonApprove = new Button("Approve");
buttonApprove.addClickListener(event -> {
//          if (grid.getSelectedItems().size() > 0) {
//              User user = (User)
VaadinSession.getCurrent().getAttribute("user");
//
service.Complete(grid.getSelectedItems().iterator().next(),user);
//          loadData();
//      }
//      });
Button button = new Button("Preview PDF", event -> {
    boolean isCheckPassed = true;
    if (!isCheckPassed) {
        Notification.show("Unfortunately you can not download this file");
    } else {
        Document so = grid.getSelectedItems().iterator().next();
        StreamResource resource = new StreamResource("Sample.pdf",
            () -> new ByteArrayInputStream(so.getData()));

        StreamRegistration registration =
VaadinSession.getCurrent().getResourceRegistry().registerResource(resource);

```

```

        UI.getCurrent().getPage().setLocation(registration.getResourceUri());
    }
});

```

```

layoutActions = new HorizontalLayout();
layoutActions.add(buttonAdd,button);
add(layoutActions);

```

```

layoutFilter = new HorizontalLayout();
layoutFilter.add(textTag,buttonSearch);
layoutFilter.setDefaultVerticalComponentAlignment(Alignment.END);
layoutFilter.setWidthFull();
addComponentAtIndex(0, layoutFilter);

```

```

loadData();

```

```

}

```

```

private void onOCR(InputStream inputStream, String filename) {

```

```

//      File convFile = new File("C:\\testing\\s.aquino.pdf");
      Tesseract t = new Tesseract();

```

```

//      t.setDatapath("C:\\Tess4J\\tessdata");

```

```

t.setDatapath("/usr/share/tesseract-ocr/tessdata");

```

```

try {

```

```

    PDDocument document = PDDocument.load(inputStream);

```

```

    Splitter splitter = new Splitter();

```

```

List<PDDocument> pages = splitter.split(document);
System.out.println("start");
int x=0;
for(PDDocument page:pages) {
    System.out.println("pages");
    Document doc = new Document();
    File file = File.createTempFile("sample", ".pdf");
    page.save(file);
    //page.close();
    String tag = t.doOCR(file);
    System.out.println(x++);
    System.out.println(tag);
    ByteArrayOutputStream baos = new
ByteArrayOutputStream();
    page.save(baos);
    page.close();

    doc.setTag(tag);
    User user = (User)
VaadinSession.getCurrent().getAttribute("user");
    boolean checking = false;
    Document dData= service.findByFilename(filename);

    //    if(dData == null) {

    for(SubCategory sub : subService.findAll()) {
        if(tag.toLowerCase().contains(sub.getTag())) {
            System.out.println("detected tag");
            //
            if(tag.toLowerCase().contains(user.getUsername())) {
                checking = true;

```

```

System.out.println("detected name");

System.out.println("sub:"+sub.getTag());

doc.setPoints(sub.getPoints());
UserCredit uc =
userCreditService.findByCategory(sub.getCategory(),user);
int totalPoints = (int)
uc.getCategory().getPoints();

double accumulatedPoints =

uc.getPoints()+doc.getPoints();

if(totalPoints > accumulatedPoints) {

uc.setPoints(accumulatedPoints);

}else if(totalPoints <=
accumulatedPoints) {

uc.setPoints(totalPoints);
}
userCreditService.save(uc);
saveUserCreditDetail(sub,user,doc);

saveUserCreditSummary(sub.getSubCatSummary(),user,doc);
//}

}

}
//}

doc.setFilename(filename);
doc.setData(baos.toByteArray());

doc.setStatus("Pending");

```

```

        //if(checking) {
            System.out.println("user:"+user.getName());
            doc.setUser(user);
            service.save(doc);
        //}

    }
    document.close();
} catch (Exception e){
    e.printStackTrace();
}

loadData();

}

private void saveUserCreditSummary(SubCategorySummary sub, User user,
Document doc) {
    UserCreditSummary uc =
summaryService.findBySubCategorySummary(sub, user);
    System.out.println("Summary");
    System.out.println(uc.getId());
    double totalPoints = uc.getPoints1();
    System.out.println(doc.getPoints());
    double accumulatedPoints = uc.getPoints1()+doc.getPoints();

    //    if(totalPoints > accumulatedPoints) {
        uc.setPoints1(accumulatedPoints);
    //    }else if(totalPoints <= accumulatedPoints) {
        //        uc.setPoints1(totalPoints);
    }
}

```

```

//      }
        summaryService.save(uc);
    }

private void saveUserCreditDetail(SubCategory sub, User user, Document doc) {
    UserCreditDetail uc = detailService.findBySubCategory(sub, user);
    System.out.println("Detail");
    System.out.println(uc.getId());
    double totalPoints = uc.getPoints1();
    double accumulatedPoints = uc.getPoints1()+doc.getPoints();
    System.out.println(doc.getPoints());
    //if(totalPoints > accumulatedPoints) {
        uc.setPoints1(accumulatedPoints);
    //}else if(totalPoints <= accumulatedPoints) {
    //    uc.setPoints(totalPoints);
    //}
    detailService.save(uc);
}

public void loadData() {
    User user = (User) VaadinSession.getCurrent().getAttribute("user");
    List<Document> list = service.findAll(textTag.getValue(),user);
    grid.setItems(list);
}

}

```

HomeView.java

```

package com.tesseract.view;

```



```

import java.time.LocalDate;
import java.time.format.DateTimeFormatter;

import com.tesseract.domain.User;
import com.tesseract.layout.MainAppLayout;
import com.vaadin.flow.component.html.H2;
import com.vaadin.flow.component.html.H3;
import com.vaadin.flow.router.AfterNavigationEvent;
import com.vaadin.flow.router.AfterNavigationObserver;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;

@SuppressWarnings("serial")
@Route(value = "", layout = MainAppLayout.class)
public class HomeView extends ProtectedView implements AfterNavigationObserver {

    private H2 labelWelcome;
    private H3 labelToday;

    public HomeView() {
        labelWelcome = new H2();
        labelToday = new H3();
        add(labelWelcome, labelToday);
    }

    @Override
    public void afterNavigation(AfterNavigationEvent event) {
        User u = (User) VaadinSession.getCurrent().getAttribute("user");
        if (u != null) {
            labelWelcome.setText("Welcome, " + u.getName() + ".");
        }
    }
}

```

```

        }
        labelToday.setText("Today is " +
LocalDate.now().format(DateTimeFormatter.ofPattern("EEEE, MMMM d, yyyy"))) +
".");
    }
}

```

LoginView.java

```

/**
 *
 */
package com.tesseract.view;

import org.springframework.beans.factory.annotation.Autowired;

import com.tesseract.domain.User;
import com.tesseract.exception.InvalidPasswordException;
import com.tesseract.exception.UserNotFoundException;
import com.tesseract.service.SystemService;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.dependency.CssImport;
import com.vaadin.flow.component.html.Div;
import com.vaadin.flow.component.login.LoginForm;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;

@SuppressWarnings("serial")
@Route(value = "login")

```

```

@CssImport("./styles/styles.css")
//@CssImport("./styles/styles.css")
public class LoginView extends VerticalLayout {

    private Div loginDiv;
    private LoginForm loginForm;

    @Autowired
    private SystemService systemService;

    public LoginView() {
        loginForm = new LoginForm();
        loginDiv = new Div(loginForm);
        loginDiv.addClassName("login-center");

        add(loginDiv);
        addClassName("login-page");
        setHeightFull();

        loginForm.addLoginListener(event -> {
            try {
                User u = systemService.login(event.getUsername(),
event.getPassword());

                VaadinSession.getCurrent().setAttribute("user", u);
                UI.getCurrent().navigate(HomeView.class);
            } catch (InvalidPasswordException e) {
                loginForm.setError(true);
            } catch (UserNotFoundException e) {
                loginForm.setError(true);
            }
        });
    }
}

```

```

        }
    });
}
}

```

LogoutView.java

```

/**
 *
 */
package com.tesseract.view;

import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.router.AfterNavigationEvent;
import com.vaadin.flow.router.AfterNavigationObserver;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;

@SuppressWarnings("serial")
@Route(value = "logout")
public class LogoutView extends VerticalLayout implements AfterNavigationObserver {

    @Override
    public void afterNavigation(AfterNavigationEvent event) {
        VaadinSession.getCurrent().setAttribute("user", null);
        UI.getCurrent().navigate(LoginView.class);
    }
}

```

ProtectedView.java

```
/**
 *
 */
package com.tesseract.view;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import com.tesseract.domain.User;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.router.BeforeEnterEvent;
import com.vaadin.flow.router.BeforeEnterObserver;
import com.vaadin.flow.server.VaadinSession;

@SuppressWarnings("serial")
public abstract class ProtectedView extends VerticalLayout implements
BeforeEnterObserver {

    private static Logger logger = LoggerFactory.getLogger(ProtectedView.class);

    @Override
    public void beforeEnter(BeforeEnterEvent event) {
        User u = (User) VaadinSession.getCurrent().getAttribute("user");
        logger.info("u: " + u);
        if (u == null) {
            // re-route to login page
            event.forwardTo(LoginView.class);
        }
    }
}
```

```

        }
    }

}

```

SetupView.java

```

/**
 *
 */
package com.tesseract.view;

import javax.annotation.PostConstruct;

import org.springframework.beans.factory.annotation.Autowired;

import com.tesseract.domain.User;
import com.tesseract.service.UserService;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.formlayout.FormLayout;
import com.vaadin.flow.component.html.Label;
import com.vaadin.flow.component.notification.Notification;
import com.vaadin.flow.component.notification.NotificationVariant;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.component.textfield.PasswordField;
import com.vaadin.flow.component.textfield.TextField;
import com.vaadin.flow.data.binder.Binder;
import com.vaadin.flow.data.binder.ValidationException;
import com.vaadin.flow.router.Route;

```

```

@SuppressWarnings("serial")
@Route(value = "setup")
public class SetupView extends VerticalLayout {

    @Autowired
    private UserService userService;

    private FormLayout layoutForm;
    private TextField username;
    private TextField name;
    private TextField email;
    private PasswordField password;

    private Binder<User> binder;
    private User bean;

    private Button buttonSave;

    private Button buttonInitializeConditionMonitoring;

    public SetupView() {
        username = new TextField("Username");
        name = new TextField("Name");
        email = new TextField("Email");
        password = new PasswordField("Password");
        password.setRequired(true);

        layoutForm = new FormLayout();
        layoutForm.add(username, name, email, password);
    }
}

```

```

        buttonSave = new Button("Add");
        buttonSave.addClickListener(event -> {
            try {
                if (password.getValue().isEmpty()) {
                    Notification.show("Please complete the required
fields.").addThemeVariants(NotificationVariant.LUMO_ERROR);
                    return;
                }
                binder.writeBean(bean);
                userService.save(bean, password.getValue());

                bean = new User();
                binder.readBean(bean);

                Notification.show("Administrator account added
successfully.").addThemeVariants(NotificationVariant.LUMO_SUCCESS);
            } catch (ValidationException e) {
                Notification.show("Please complete the required
fields.").addThemeVariants(NotificationVariant.LUMO_ERROR);
            }
        });

        add(new Label("Adminstrator Account"), layoutForm, buttonSave);

        binder = new Binder<>(User.class);
        binder.forMemberField(username).asRequired();
        binder.forMemberField(name).asRequired();
        binder.bindInstanceFields(this);
    }

```



```

        @PostConstruct
        private void init() {
            bean = new User();
            bean.setAdmin(true);
            binder.readBean(bean);
        }
    }
}

```

SubCategoryView.java

```

package com.tesseract.view;

import java.util.List;

import javax.annotation.PostConstruct;

import org.hibernate.event.internal.OnLockVisitor;
import org.springframework.beans.factory.annotation.Autowired;

import com.tesseract.domain.Category;
import com.tesseract.domain.SubCategory;
import com.tesseract.layout.MainAppLayout;
import com.tesseract.service.CategoryService;
import com.tesseract.service.SubCategoryService;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.combobox.ComboBox;
import com.vaadin.flow.component.dialog.Dialog;
import com.vaadin.flow.component.grid.Grid;

```

```

import com.vaadin.flow.component.html.Anchor;
import com.vaadin.flow.component.orderedlayout.HorizontalLayout;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.component.textfield.NumberField;
import com.vaadin.flow.component.textfield.TextField;
import com.vaadin.flow.router.Route;

@Route(value = "sub-category", layout = MainAppLayout.class)
public class SubCategoryView extends VerticalLayout {

    /**
     *
     */
    private static final long serialVersionUID = -7349714164334467197L;

    @Autowired
    private SubCategoryService service;

    private Grid<SubCategory> grid;

    private HorizontalLayout layoutActions;
    private Button buttonAdd;
    private Button buttonPrint;
    private Button buttonApprove;

    private TextField textTag;

    private Button buttonSearch;

```

```

private HorizontalLayout layoutFilter;

private Dialog updateDialog = new Dialog();
private Dialog detailDialog = new Dialog();

@Autowired
private CategoryService catService;

private Anchor anchor;

// @Autowired
// private ExportUtil exportUtil;

public SubCategoryView() {
    grid = new Grid<>(SubCategory.class);
    add(grid);
}

@PostConstruct
private void init() {
    grid.removeAllColumns();
    //grid.addColumn(o ->
o.getCategory().getName()).setHeader("Category").setAutoWidth(true);
    grid.addColumn(o ->
o.getName()).setHeader("Name").setAutoWidth(true);
    grid.addColumn(o->
o.getPoints()).setHeader("Points").setAutoWidth(true);
    grid.addColumn(o ->
o.getTag()).setHeader("Keywords").setAutoWidth(true);

```

```

        grid.setWidthFull();

//        date = new DatePicker("Date");
//
        textTag = new TextField("Insert keyword here");
        textTag.setWidth("100%");
//        comboStatus = new ComboBox<String>("Status",statusUtil.getStatus());
//        comboStatus.setValue("Pending");
        buttonSearch = new Button("Search");
        buttonSearch.setWidth("100px");
        buttonSearch.addClickListener(e->{
            loadData();
        });
        buttonAdd = new Button("Add");
        buttonAdd.addClickListener(event -> {
            Dialog d = new Dialog();
            d.setWidth("400px");
            VerticalLayout content = new VerticalLayout();
            SubCategory entity = new SubCategory();
            TextField txtName= new TextField("Name");
            txtName.setSizeFull();

            ComboBox<Category> cat = new
ComboBox<Category>("Category");
            cat.setItems(catService.findAll());
            cat.setSizeFull();
            NumberField textPoints = new NumberField("Points");
            textPoints.setSizeFull();
            TextField textTag= new TextField("Keywords");
            textTag.setSizeFull();

```

```

        Button save = new Button("Save");
        save.addClickListener(e->{
            entity.setName(txtName.getValue());
            entity.setCategory(cat.getValue());
            entity.setTag(textTag.getValue());
            entity.setPoints(textPoints.getValue().intValue());
            service.save(entity);
            d.close();
            loadData();
        });
        content.add(cat,txtName,textPoints,textTag,save);
        d.add(content);
        d.open();

    });

    Button edit = new Button("Edit");
    edit.addClickListener(event -> {
        Dialog d = new Dialog();
        d.setWidth("400px");
        VerticalLayout content = new VerticalLayout();
        SubCategory entity = grid.getSelectedItems().iterator().next();
        TextField txtName= new TextField("Name");
        txtName.setSizeFull();

        ComboBox<Category> cat = new
ComboBox<Category>("Category");
        cat.setItems(catService.findAll());
        cat.setSizeFull();
        NumberField textPoints = new NumberField("Points");
        textPoints.setSizeFull();

```

```

        TextField textTag= new TextField("Keywords");
        textTag.setSizeFull();

        Button save = new Button("Save");
        if(entity != null) {
            txtName.setValue((String) entity.getName());
            cat.setValue(entity.getCategory());
            textTag.setValue((String) entity.getTag());
            textPoints.setValue(Double.valueOf((String)
entity.getPoints()));

            save.addClickListener(e->{
                entity.setName(txtName.getValue());
                entity.setCategory(cat.getValue());
                entity.setTag(textTag.getValue());
                entity.setPoints(textPoints.getValue().intValue());
                service.save(entity);
                d.close();
                loadData();
            });
        }

        content.add(cat,txtName,textPoints,textTag,save);
        d.add(content);
        d.open();

    });

    Button delete = new Button("Delete");
    delete.addClickListener(event -> {

        SubCategory entity = grid.getSelectedItems().iterator().next();

```

```

        service.delete(entity);
        loadData();
    });

    layoutActions = new HorizontalLayout();
    layoutActions.add(buttonAdd,edit,delete);
    add(layoutActions);

    layoutFilter = new HorizontalLayout();
    layoutFilter.add(textTag,buttonSearch);
    layoutFilter.setDefaultVerticalComponentAlignment(Alignment.END);
    layoutFilter.setWidthFull();
    addComponentAtIndex(0, layoutFilter);

    loadData();

}

private void loadData() {

    List<SubCategory> list = service.findAll();
    grid.setItems(list);

}

}

```

TrainingView.java

```

package com.tesseract.view;

import java.util.List;

```

```

import javax.annotation.PostConstruct;

import org.springframework.beans.factory.annotation.Autowired;

import com.tesseract.domain.Category;
import com.tesseract.domain.SubCategory;
import com.tesseract.domain.Training;
import com.tesseract.layout.MainAppLayout;
import com.tesseract.service.CategoryService;
import com.tesseract.service.SubCategoryService;
import com.tesseract.service.TrainingService;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.combobox.ComboBox;
import com.vaadin.flow.component.dialog.Dialog;
import com.vaadin.flow.component.grid.Grid;
import com.vaadin.flow.component.html.Anchor;
import com.vaadin.flow.component.orderedlayout.HorizontalLayout;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.component.textfield.NumberField;
import com.vaadin.flow.component.textfield.TextField;
import com.vaadin.flow.router.Route;

@Route(value = "training", layout = MainAppLayout.class)
public class TrainingView extends VerticalLayout {
    private static final long serialVersionUID = -7349714164334467197L;

    @Autowired
    private TrainingService service;

```



```

        private Grid<Training> grid;

        private HorizontalLayout layoutActions;
        private Button buttonAdd;
        private Button buttonPrint;
        private Button buttonApprove;

        private TextField textTag;

        private Button buttonSearch;
        private HorizontalLayout layoutFilter;

        private Dialog updateDialog = new Dialog();
        private Dialog detailDialog = new Dialog();

        @Autowired
        private CategoryService catService;

        private Anchor anchor;

//        @Autowired
//        private ExportUtil exportUtil;

        public TrainingView() {
            grid = new Grid<>(Training.class);
            add(grid);
        }

        @PostConstruct

```

```

private void init() {

    grid.removeAllColumns();
    grid.addColumn(o ->
o.getCategory().getName()).setHeader("Category").setAutoWidth(true);
    grid.addColumn(o ->
o.getName()).setHeader("Name").setAutoWidth(true);
    grid.addColumn(o -> o.getContactPerson()).setHeader("Contact
Person").setAutoWidth(true);
    grid.addColumn(o -> o.getContactNumber()).setHeader("Contact
Number").setAutoWidth(true);

    grid.setWidthFull();

//    date = new DatePicker("Date");
//
    textTag = new TextField("Insert keyword here");
    textTag.setWidth("100%");
//    comboStatus = new ComboBox<String>("Status",statusUtil.getStatus());
//    comboStatus.setValue("Pending");
    buttonSearch = new Button("Search");
    buttonSearch.setWidth("100px");
    buttonSearch.addClickListener(e->{
        loadData();
    });
    buttonAdd = new Button("Add");
    buttonAdd.addClickListener(event -> {
        Dialog d = new Dialog();
        d.setWidth("400px");
        VerticalLayout content = new VerticalLayout();
        Training entity = new Training();

```

```

        TextField txtName= new TextField("Training Name");
        txtName.setSizeFull();
        TextField txtPerson= new TextField("Contact Person");
        txtPerson.setSizeFull();
        TextField txtNumber= new TextField("Contact Number");
        txtNumber.setSizeFull();
        TextField txtAddress= new TextField("Address");
        txtAddress.setSizeFull();
        TextField txtEmail= new TextField("Email");
        txtEmail.setSizeFull();
        ComboBox<Category> cat = new
ComboBox<Category>("Category");
        cat.setItems(catService.findAll());
        cat.setSizeFull();


        Button save = new Button("Save");
        save.addClickListener(e->{
            entity.setName(txtName.getValue());
            entity.setCategory(cat.getValue());
            entity.setAddress(txtAddress.getValue());
            entity.setContactPerson(txtPerson.getValue());
            entity.setContactNumber(txtNumber.getValue());
            entity.setEmail(txtEmail.getValue());
            service.save(entity);
            d.close();
            loadData();
        });

        content.add(cat,txtName,txtPerson,txtNumber,txtAddress,txtEmail,save);
        d.add(content);

```

```

        d.open();
    });
    Button edit = new Button("Edit");
    edit.addClickListener(event -> {
        Dialog d = new Dialog();
        d.setWidth("400px");
        VerticalLayout content = new VerticalLayout();
        Training entity = grid.getSelectedItems().iterator().next();
        TextField txtName= new TextField("Training Name");
        txtName.setSizeFull();
        TextField txtPerson= new TextField("Contact Person");
        txtPerson.setSizeFull();
        TextField txtNumber= new TextField("Contact Number");
        txtNumber.setSizeFull();
        TextField txtAddress= new TextField("Address");
        txtAddress.setSizeFull();
        TextField txtEmail= new TextField("Email");
        txtEmail.setSizeFull();
        ComboBox<Category> cat = new
ComboBox<Category>("Category");
        cat.setItems(catService.findAll());
        cat.setSizeFull();

        Button save = new Button("Save");
        if(entity != null) {
            txtName.setValue((String) entity.getName());
            cat.setValue(entity.getCategory());
            txtAddress.setValue(entity.getAddress());
            txtPerson.setValue((String) entity.getContactPerson());
            txtNumber.setValue((String) entity.getContactNumber());

```

```

        txtEmail.setValue(entity.getEmail());
        save.setOnClickListener(e->{
            entity.setName(txtName.getValue());
            entity.setCategory(cat.getValue());
            entity.setAddress(txtAddress.getValue());
            entity.setContactPerson(txtPerson.getValue());
            entity.setContactNumber(txtNumber.getValue());
            entity.setEmail(txtEmail.getValue());
            service.save(entity);
            d.close();
            loadData();
        });
    }

    content.add(cat,txtName,txtPerson,txtNumber,txtAddress,txtEmail,save);
    d.add(content);
    d.open();
});

Button delete = new Button("Delete");
delete.setOnClickListener(event -> {

    Training entity = grid.getSelectedItems().iterator().next();
    service.delete(entity);
    loadData();
});

layoutActions = new HorizontalLayout();
layoutActions.add(buttonAdd,edit,delete);
add(layoutActions);

```

```

        layoutFilter = new HorizontalLayout();
        layoutFilter.add(textTag,buttonSearch);
        layoutFilter.setDefaultVerticalComponentAlignment(Alignment.END);
        layoutFilter.setWidthFull();
        addComponentAtIndex(0, layoutFilter);
        loadData();
    }
    private void loadData() {

        List<Training> list = service.findAll();
        grid.setItems(list);
    }
}

```

UserCreditView.java

```

package com.tesseract.view;

import java.io.ByteArrayInputStream;
import java.util.List;

import javax.annotation.PostConstruct;

import org.springframework.beans.factory.annotation.Autowired;

import com.tesseract.domain.Document;
import com.tesseract.domain.Training;
import com.tesseract.domain.User;
import com.tesseract.domain.UserCredit;
import com.tesseract.layout.ExportUtil;
import com.tesseract.layout.MainAppLayout;

```

```

import com.tesseract.service.TrainingService;
import com.tesseract.service.UserCreditService;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.dialog.Dialog;
import com.vaadin.flow.component.grid.Grid;
import com.vaadin.flow.component.html.Anchor;
import com.vaadin.flow.component.notification.Notification;
import com.vaadin.flow.component.orderedlayout.HorizontalLayout;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.component.textfield.TextField;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.StreamRegistration;
import com.vaadin.flow.server.StreamResource;
import com.vaadin.flow.server.VaadinSession;

```

```

@Route(value = "UserCredit", layout = MainAppLayout.class)

```

```

public class UserCreditView extends VerticalLayout {

```

```

    private static final long serialVersionUID = -7349714164334467197L;

```

```

    @Autowired

```

```

    private UserCreditService service;

```

```

    private Grid<UserCredit> grid;

```

```

    private HorizontalLayout layoutActions;

```

```

    private Button buttonAdd;

```

```

    private Button buttonPrint;

```

```

    private Button buttonApprove;

```

```

private TextField textTag;

private Button buttonSearch;
private HorizontalLayout layoutFilter;

private Dialog updateDialog = new Dialog();
private Dialog detailDialog = new Dialog();


private Anchor anchor;

@Autowired
private ExportUtil exportUtil;

@Autowired
private TrainingService trainingService;

private double total;

public UserCreditView() {
    grid = new Grid<>(UserCredit.class);
    add(grid);
}

@PostConstruct
private void init() {
    grid.removeAllColumns();
    grid.addColumn(o ->
o.getCategory().getName()).setHeader("Category").setAutoWidth(true);

```



```

        grid.addColumn(o -> o.getCategory().getPoints()).setHeader("Target
Points").setAutoWidth(true);
        grid.addColumn(o -> o.getPoints()).setHeader("Credit
Points").setAutoWidth(true);

        grid.setWidthFull();

//        date = new DatePicker("Date");
//
        textTag = new TextField("Insert keyword here");
        textTag.setWidth("100%");
//        comboStatus = new ComboBox<String>("Status",statusUtil.getStatus());
//        comboStatus.setValue("Pending");
        buttonSearch = new Button("Search");
        buttonSearch.setWidth("100px");
        buttonSearch.addClickListener(e->{
            loadData();
        });
        buttonAdd = new Button("Training Avail");
        buttonAdd.addClickListener(e->{
            Dialog d= new Dialog();
            d.setWidth("500px");
            VerticalLayout content = new VerticalLayout();
            content.setSizeFull();
            Grid<Training> gridTraining = new Grid<>(Training.class);
            gridTraining.removeAllColumns();
            gridTraining.addColumn(o ->
o.getCategory().getName()).setHeader("Category").setAutoWidth(true);
            gridTraining.addColumn(o -> o.getName()).setHeader("Training
Name").setAutoWidth(true);

```

```

        gridTraining.addColumn(o ->
o.getContactPerson()).setHeader("Contact Person").setAutoWidth(true);
        gridTraining.addColumn(o ->
o.getContactNumber()).setHeader("Contact Person").setAutoWidth(true);
        gridTraining.addColumn(o ->
o.getAddress()).setHeader("Address").setAutoWidth(true);
        gridTraining.addColumn(o ->
o.getEmail()).setHeader("Email").setAutoWidth(true);

        gridTraining.setWidthFull();

        content.add(gridTraining);
        d.add(content);
        UserCredit so = grid.getSelectedItems().iterator().next();
        if (so != null) {
            List<Training> list =
trainingService.findByCategory(so.getCategory());
            gridTraining.setItems(list);
            d.open();
        }

    });

    Button button = new Button("Print Summary", event -> {
        boolean isCheckPassed = true;
        if (!isCheckPassed) {
            Notification.show("Unfortunately you can not download this file");
        } else {
            User user = (User)
VaadinSession.getCurrent().getAttribute("user");
            StreamResource resource = new StreamResource("Summary.pdf",
                () -> new ByteArrayInputStream(exportUtil.generate(user)));

```

```

        StreamRegistration registration =
VaadinSession.getCurrent().getResourceRegistry().registerResource(resource);
        UI.getCurrent().getPage().setLocation(registration.getResourceUri());
    }
});

```

```

Button buttonDetail = new Button("Print Detail", event -> {
    boolean isCheckPassed = true;
    if (!isCheckPassed) {
        Notification.show("Unfortunately you can not download this file");
    } else {
        User user = (User)
VaadinSession.getCurrent().getAttribute("user");
        StreamResource resource = new StreamResource("Detail.pdf",
            () -> new
ByteArrayInputStream(exportUtil.generateDetail(user)));

```

```

        StreamRegistration registration =
VaadinSession.getCurrent().getResourceRegistry().registerResource(resource);
        UI.getCurrent().getPage().setLocation(registration.getResourceUri());
    }
});

```

```

Button buttonDetailSummary = new Button("Print Detail Summary",
event -> {
    boolean isCheckPassed = true;

```

```

        if (!isCheckedPassed) {
            Notification.show("Unfortunately you can not download this file");
        } else {
            User user = (User)
VaadinSession.getCurrent().getAttribute("user");

            StreamResource resource = new StreamResource("Detail.pdf",
                () -> new
ByteArrayInputStream(exportUtil.generateDetailSummary(user,total)));

            StreamRegistration registration =
VaadinSession.getCurrent().getResourceRegistry().registerResource(resource);
            UI.getCurrent().getPage().setLocation(registration.getResourceUri());
        }
    });

    Button buttonDetailDanger = new Button("Print Lack Points", event -> {
        boolean isCheckedPassed = true;
        if (!isCheckedPassed) {
            Notification.show("Unfortunately you can not download this file");
        } else {
            User user = (User)
VaadinSession.getCurrent().getAttribute("user");

            StreamResource resource = new StreamResource("detail-lack-
points.pdf",
                () -> new
ByteArrayInputStream(exportUtil.generateDetailDanger(user)));

```

```

        StreamRegistration registration =
VaadinSession.getCurrent().getResourceRegistry().registerResource(resource);
        UI.getCurrent().getPage().setLocation(registration.getResourceUri());
    }
});

layoutActions = new HorizontalLayout();
layoutActions.add(buttonAdd);
add(layoutActions);

layoutFilter = new HorizontalLayout();

layoutFilter.add(button,buttonDetail,buttonDetailDanger,buttonDetailSummary);
layoutFilter.setDefaultVerticalComponentAlignment(Alignment.END);
layoutFilter.setWidthFull();
addComponentAtIndex(0, layoutFilter);
loadData();

}
private void loadData() {
    total = 0;
    User user = (User) VaadinSession.getCurrent().getAttribute("user");
    List<UserCredit> list = service.findAll(user);
    for(UserCredit uc :list) {
        total +=uc.getPoints();
    }
    grid.setItems(list);
    System.out.println(service.findAllIn(user).size());
}
}

```

UsersView.java

```
package com.tesseract.view;

import javax.annotation.PostConstruct;

import org.springframework.beans.factory.annotation.Autowired;

import com.tesseract.domain.User;
import com.tesseract.layout.MainAppLayout;
import com.tesseract.service.UserService;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.dialog.Dialog;
import com.vaadin.flow.component.formlayout.FormLayout;
import com.vaadin.flow.component.grid.Grid;
import com.vaadin.flow.component.html.Label;
import com.vaadin.flow.component.notification.Notification;
import com.vaadin.flow.component.notification.NotificationVariant;
import com.vaadin.flow.component.orderedlayout.HorizontalLayout;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.component.textfield.PasswordField;
import com.vaadin.flow.component.textfield.TextField;
import com.vaadin.flow.data.binder.Binder;
import com.vaadin.flow.data.binder.ValidationException;
import com.vaadin.flow.router.AfterNavigationEvent;
import com.vaadin.flow.router.AfterNavigationObserver;
import com.vaadin.flow.router.Route;

@SuppressWarnings("serial")
@Route(value = "users", layout = MainAppLayout.class)
public class UsersView extends ProtectedView implements AfterNavigationObserver {
```

```

@Autowired
private UserService userService;

private Grid<User> gridUsers;

private FormLayout layoutForm;
private TextField username;
private TextField name;
private TextField email;
private PasswordField password;

private Binder<User> binder;
private User bean;

private Button buttonSave;
private Button buttonRefresh;
private Button buttonChangePassword;

private Dialog dialogChangePassword;
private PasswordField fieldNewPassword;
private Button buttonConfirmChangePassword;

public UsersView() {
    gridUsers = new Grid<>(User.class);
    gridUsers.setColumns("username", "name", "email");

    username = new TextField("Username");
    name = new TextField("Name");
    email = new TextField("Email");
    password = new PasswordField("Password");
    password.setRequired(true);
}

```

```

layoutForm = new FormLayout();
layoutForm.add(username, name, email, password);

buttonSave = new Button("Add");
buttonSave.addClickListener(event -> {
    try {
        if (password.getValue().isEmpty()) {
            Notification.show("Please complete the required
fields.").addThemeVariants(NotificationVariant.LUMO_ERROR);
            return;
        }
        binder.writeBean(bean);
        userService.save(bean, password.getValue());
        bean = new User();
        binder.readBean(bean);
        loadUsers();
        Notification.show("User added
successfully.").addThemeVariants(NotificationVariant.LUMO_SUCCESS);
    } catch (ValidationException e) {
        Notification.show("Please complete the required
fields.").addThemeVariants(NotificationVariant.LUMO_ERROR);
    }
});

buttonRefresh = new Button("Refresh");
buttonRefresh.addClickListener(event -> {
    loadUsers();
});

buttonChangePassword = new Button("Change Password");
buttonChangePassword.addClickListener(event -> {

```



```

        if (gridUsers.getSelectedItems().size() > 0) {
            dialogChangePassword.open();
        }
    });
    add(gridUsers, new HorizontalLayout(buttonRefresh,
buttonChangePassword), layoutForm, buttonSave);
    binder = new Binder<>(User.class);
    binder.forMemberField(username).asRequired();
    binder.forMemberField(name).asRequired();
    binder.bindInstanceFields(this);
    dialogChangePassword = new Dialog();
    fieldNewPassword = new PasswordField("New Password");
    buttonConfirmChangePassword = new Button("Confirm");
    buttonConfirmChangePassword.addClickListener(event -> {
        if (fieldNewPassword.getValue().isEmpty()) {
            Notification.show("Password is
required.").addThemeVariants(NotificationVariant.LUMO_ERROR);
            fieldNewPassword.focus();
            return;
        }
        User u = gridUsers.getSelectedItems().iterator().next();
        userService.changePassword(u, fieldNewPassword.getValue());
        dialogChangePassword.close();
        Notification.show("Password changed
successfully.").addThemeVariants(NotificationVariant.LUMO_SUCCESS);
    });
    VerticalLayout v = new VerticalLayout();
    v.setPadding(false);
    v.add(new Label("Change Password"), fieldNewPassword,
buttonConfirmChangePassword);
    dialogChangePassword.add(v);

```

```
}  
@PostConstruct  
private void init() {  
    bean = new User();  
    binder.readBean(bean);  
}  
private void loadUsers() {  
    gridUsers.setItems(userService.findAll());  
}  
@Override  
public void afterNavigation(AfterNavigationEvent event) {  
    loadUsers();  
}  
}
```

RECO ILLUSTRISIMO BONGCALES

Address : College View Park Subd., Illustrisimo Compound Puro 4 Zone 8 Brgy.

Cupang Antipolo City

Mobile No. : (63)9485455229

Email Add. : Bongcalesreco03@gmail.com



PERSONAL VITAE

Date of Birth : April 3, 1998
Age : 22
Place of Birth : Marikina
Marital Status : Single
Religion : Roman Catholic
Height : 5'4
Weight : 50 kgs.
Citizenship : Filipino
Language spoken : Filipino and English

QUALIFICATION AND SKILLS

Programming Languages

- Basic Java Programming, PHP and Python Programming
- Basic C and C++ Programming
- SQLite Database Management

Hardware

- Electronic Devices Troubleshooting
- Personal Computer Troubleshooting and Services
- Local Area Network Connection Troubleshooting

Software

- Circuit Wizard (Circuit Simulation)
- Visual Studio
- Eclipse IDE
- Microsoft Office
-

EDUCATIONAL ATTAINMENT

➤ **Technological University of the Philippines**

Ayala Blvd, Ermita, Manila.

Bachelor of Science in Electronics Communication Engineering

2017 to Present

➤ **Technological University of the Philippines**

Ayala Blvd, Ermita, Manila.

Electronics Communication Engineering Technology

2014 to 2017

➤ **Cupang National High School**

Purok 2 Zone 8 Cupang Antipolo City

2010-2014

➤ **Cupang Elementary School**

Purok 2 Zone 8 Cupang Antipolo City

2004-2010

TRAININGS AND LEADERSHIP

➤ **PhilKo Ubins**

November 2017 to March 2018

Intern

KIERVY SAILE CABELTES

Address : Blk 9, Lot 6, Villa Nova ave. Villa Nova Subd. Bgry. Nagkaisang
Nayon, Novaliches Quzon City

Mobile No. : (63)9653683232

Email Add. : kiervycblts08@gmail.com



PERSONAL VITAE

Date of Birth : January 8, 1996

Age : 23

Place of Birth : Manila

Marital Status : Single

Religion : Christian Baptist

Height : 5'6

Weight : 50 kgs.

Citizenship : Filipino

Language spoken : Filipino and English

QUALIFICATION AND SKILLS

- Proficient in NI Multisim, and MATLAB
- Basic knowledge in Programming and Photoshop
- Public speaker
- Know how to lead people

EDUCATIONAL ATTAINMENT

- Technological University of the Philippines
Ayala Blvd, Ermita, Manila.
Bachelor of Science in Electronics Communication Engineering
2016 to Present
- Technological University of the Philippines
Ayala Blvd, Ermita, Manila.
Electronics Communication engineering technology
2013 to 2016

- Novaliches High School
Brgy. San Agustin T.S Crus subd. Novaliches, Quezon City
2009-2013
- Nagkaisang Nayon Elementary School
Brgy. Nagkaisang Nayon Novaliches, Quezon City
2003-2009

TRAININGS AND LEADERSHIP

- **Worship Generation**
May 2018 to Present
Social Media Ambassador
- **Institute of Electronics Engineers of the Philippines -Manila**
May 2018 to May 2019
Vice Chairman for External Affairs
- **Organization of Electronics Engineering Students**
June 2018 to April 2019
Vice President for External Affairs
- **ABS-CBN LINGKOD KAPAMILYA FOUNDATION INC.**
May to October, 2015
On the Job Training

JORELEEN C. OBEJAS

Address : 0007 Blk. 11, Durian St., Golden Acres Subd., Talon
Uno, Las Piñas City
Mobile No. : (63)9610738493
Email Add. : j.cobejas19@gmail.com



PERSONAL VITAE

Date of Birth : November 19, 1998
Age : 21
Place of Birth : Pangasinan
Marital Status : Single
Religion : Born Again
Height : 5'2"
Weight : 42 kg
Citizenship : Filipino
Language spoken : Filipino and English

QUALIFICATION AND SKILLS

- Proficient in NI Multisim and MATLAB
- Basic knowledge in Programming
- Finish tasks on time
- Determined to work hard

EDUCATIONAL ATTAINMENT

- Technological University of the Philippines
Ayala Blvd, Ermita, Manila
Bachelor of Science in Electronics Communication Engineering
2015 to Present
- Las Piñas City National Science High School
Carnival Park St., BF Resorts Village, Las Piñas City
2011-2015
- Almanza Elementary School

Almanza Uno, Las Piñas City
2005-2011

TRAININGS AND LEADERSHIP

- **Institute of Electronics Engineers of the Philippines -Manila**
2017-2020
Member
- **Organization of Electronics Engineering Students**
2015-2020
Member

LOUIE ANDRO TABANGAY RAUTO

Address : Blk 9B, Lot 7, Phase 2, Area 3 A/B Dagat-dagatan, Malabon City

Mobile No. : (63)9984636304

Email Add. : androrauto@gmail.com



PERSONAL VITAE

Date of Birth : June 5, 1997
Age : 23
Place of Birth : Manamoc, Cuyo, Palawan
Marital Status : Single
Religion : Roman Catholic
Height : 5'6
Weight : 45 kgs.
Citizenship : Filipino
Language spoken : Filipino and English

QUALIFICATION AND SKILLS

- Has a basic knowledge in troubleshooting and repairing electronic system.
- Has a basic knowledge in Multisim, PCB Wizard, MATLAB, Simulink, Python programming language, Java programming language and Octave.
- Can work productively in an independent or a team environment.
- Able to learn new tasks easily and quickly.

EDUCATIONAL ATTAINMENT

➤ Technological University of the Philippines

Ayala Blvd, Ermita, Manila.

Bachelor of Science in Electronics Communication Engineering

2017 to Present

➤ Technological University of the Philippines

Ayala Blvd, Ermita, Manila.

Electronics Communication Engineering Technology

2014 to 2017

➤ **Manamoc National High School**

Manamoc, Cuyo, Palawan

2010-2014

➤ **Manamoc Elementary School**

Manamoc, Cuyo, Palawan

2004-2010

TRAININGS AND LEADERSHIP

➤ **Tricom Dynamics Inc.**

November 2017 to March 2018

Intern

➤ **Pantronics International Corporation - Intern**

February 2020 to March 2020

ELLISE MARY ANNE M. RUIZ

Address : 2029 A-415 Rizal Avenue Sta. Cruz, Manila
Mobile No. : (63)9052565639
Email Add. : ellisemaryanneruiz@gmail.com



PERSONAL VITAE

Date of Birth : September 16, 1999
Age : 20
Place of Birth : Manila
Marital Status : Single
Religion : Roman Catholic
Height : 54
Weight : 55 kgs.
Citizenship : Filipino
Language spoken : Filipino and English

QUALIFICATION AND SKILLS

- Proficient in NI Multisim, MATLAB, and Octave
- Basic knowledge in Programming and Photoshop
- Highly determined and dedicated towards work
- Work well with others and have good communication skills

EDUCATIONAL ATTAINMENT

- Technological University of the Philippines
Ayala Blvd, Ermita, Manila.
Bachelor of Science in Electronics Communication Engineering
2016 to Present
- Benigno S. Aquino National High School
San Nicolas Poblacion Concepcion Tarlac
2011-2015
- Talimundoc San Miguel Elementary School
Brgy. Talimundoc San Miguel Concepcion Tarlac

2005 - 2011

TRAININGS AND LEADERSHIP

➤ **Organization of Electronics Engineering Students**

2016 to Present

Member

➤ **Institute of Electronics Engineering of the Philippines - Manila**

2017 to 2019

Member

➤ **NASA International Space Apps Challenge**

2018

References:

- Abdul Robby G., A. T. (2019). Implementation of Optical Character Recognition using Tesseract with the Javanese Script Target in Android Application. 4th International Conference on Computer Science and Computational Intelligence 2019 (ICCSCI), 12-13 September 2019, 7.
- Adesso Inc. (2019). *Nuscan™ 7600TU 2D Antimicrobial Handheld Barcode Scanner - Adesso Inc :: Your Input Device Specialist ::*. [online] Available at: <https://www.adesso.com/product/nuscan-7600tu-2d-antimicrobial-handheld-barcode-scanner/> [Accessed 17 Jul. 2019].
- Alkhalaf, K. (2014). OCR-Based Electronic Documentation Management System. International Journal of Innovation, Management and Technology, 5(6). doi:10.7763/ijimt.2014.v5.560
- Amazon.com. (2019). [online] Available at: <https://www.amazon.com/VuPoint-Solutions-Magic-Portable-Scanner/dp/B004EFXW6Q> [Accessed 17 Jul. 2019].
- Android Central. (2019). *How to use FolderSync to manage all of your media*. [online] Available at: <https://www.androidcentral.com/how-sync-your-personal-music-and-other-files-foldersync> [Accessed 17 Jul. 2019].
- Anon, (2019). [online] Available at: <https://www.tacit.dk/foldersync/> [Accessed 17 Jul. 2019].
- Anon, (2019). [online] Available at: <https://www.amazon.in/Canon-CanoScan-FB620U-Flatbed-Scanner/dp/B00000JFKO> [Accessed 17 Jul. 2019].
- Anon, (2019). [online] Available at: https://www.gearbest.com/scanners/pp_009606908545.html [Accessed 17 Jul. 2019].
- Bangera, J. (2019). *Top 9 Cloud Storage Trends Of 2015 - Inkjet Wholesale Blog*. [online] Inkjet Wholesale Blog. Available at:

<https://blog.inkjetwholesale.com.au/office-efficiency/top-cloud-storage-trend-of-2015/> [Accessed 17 Jul. 2019].

Christensson, P. (2012, April 19). Java Definition. Retrieved 2020, Aug 3, from <https://techterms.com>

Consumerreports.org. (2019). *Are Detachable Computers the Best of Both Worlds?* - *Consumer Reports*. [online] Available at: <https://www.consumerreports.org/cro/magazine/2014/11/are-detachable-computers-the-best-of-both-worlds/index.htm> [Accessed 17 Jul. 2019].

Dr Sunanda Dixit, B. M. (2018). Optical Recognition of Digital Characters Using Machine. *International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)*, 8.

Explain that Stuff. (2019). *How does a hard drive work?*. [online] Available at: <https://www.explainthatstuff.com/harddrive.html> [Accessed 17 Jul. 2019].

Goddess, C., Jobs, J., Opinion, N. and Communities, C. (2019). *How Developers use Python Programming Language*. [online] TechGig. Available at: <https://content.techgig.com/how-developers-use-python-programming-language/articleshow/67886849.cms> [Accessed 17 Jul. 2019].

Google.com. (2019). *foldersync app - Google Search*. [online] Available at: https://www.google.com/search?q=foldersync+app&rlz=1CAZZAC_enPH746PH750&source=lnms&tbn=isch&sa=X&ved=0ahUKEwi4vrTQsLHjAhWVdHAKHeqvBtkQ_AUIESgC&biw=1366&bih=641#imgrc=SBNR0r_6u1scYM [Accessed 17 Jul. 2019].

Howard Austerlitz, in *Data Acquisition Techniques Using PCs (Second Edition)*, 2003

Imaging-Superstore. (2019). *Kodak PS80*. [online] Available at: <https://imaging-superstore.co.uk/products/kodak-ps80> [Accessed 17 Jul. 2019].

indiamart.com. (2019). *Hp Scanjet 7000 S2 Sheet Feed Scanner*. [online] Available at: <https://www.indiamart.com/proddetail/hp-scanjet-7000-s2-sheet-feed-scanner-11459201248.html> [Accessed 17 Jul. 2019].

Lifewire. (2019). *A USB Port Is One of the Most Useful Features on Computers and Phones*. [online] Available at: <https://www.lifewire.com/what-is-a-usb-port-818166> [Accessed 17 Jul. 2019].

Lifewire. (2019). *A USB Port Is One of the Most Useful Features on Computers and Phones*. [online] Available at: <https://www.lifewire.com/what-is-a-usb-port-818166> [Accessed 17 Jul. 2019].

Lifewire. (2019). *What Is a Router (Residential Gateway) and How Does It Work?*. [online] Available at: <https://www.lifewire.com/what-is-a-router-2618162> [Accessed 17 Jul. 2019].

M. J., Dr. (2018, October 15). A Simple Introduction to Natural Language Processing. Retrieved July 31, 2019, from <https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32>

M. Swain, J. A. Anderson, R. Korrapati and N. K. Swain, "Database programming using Java," Proceedings IEEE SoutheastCon 2002 (Cat. No.02CH37283), Columbia, SC, USA, 2002, pp. 220-225, doi: 10.1109/SECON.2002.995590.

Networking, C., Connectors, C., USB Cables, H. and lot, D. (2019). *USB 2.0 Extension Cable USB Male to Female Lead Cord USB Extender Wire lot / eBay*. [online] eBay. Available at: <https://www.ebay.co.uk/itm/USB-2-0-Extension-Cable-USB-Male-to-Female-Lead-Cord-USB-Extender-Wire-lot-/122903971809> [Accessed 17 Jul. 2019].

Nobuya Ishihara, N. F.-C. (2017). A Software Architecture for Java Programming Learning Assistant System. International Journal of Computer & Software Engineering, 7.

Ohlsson, V. (2016). Optical Character and Symbol Recognition using Tesseract. 77.

Preservationtutorial.library.cornell.edu. (2019). *Digital Imaging Tutorial - Image Creation*. [online] Available at:
<http://preservationtutorial.library.cornell.edu/technical/technicalB-02.html>
[Accessed 17 Jul. 2019].

Python.org. (2019). *What is Python? Executive Summary*. [online] Available at:
<https://www.python.org/doc/essays/blurb/> [Accessed 17 Jul. 2019].

Rizvi, Mehdi & Raza, Hasnain & Tahzeeb, Shahab & Jaffry, Shan. (2019). Optical Character Recognition Based Intelligent Database Management System for Examination Process Control. 500-507. 10.1109/IBCAST.2019.8667127.

Rouse, M. (2015, January). What is a Database Management System (DBMS)? - Definition from WhatIs.com. Retrieved July 16, 2019, from
<https://searchsqlserver.techtarget.com/definition/database-management-system>

Rouse, M. (n.d.). What is OCR (optical character recognition)? - Definition from WhatIs.com. Retrieved from
<https://searchcontentmanagement.techtarget.com/definition/OCR-optical-character-recognition>

Smith, R. (2007). An Overview of the Tesseract OCR Engine. Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2.
doi:10.1109/icdar.2007.4376991

State Universities and Colleges Faculty Student Ratio. (2018, June 8). Retrieved July 16, 2019, from <https://ched.gov.ph/wp-content/uploads/2018/07/State-Universities-and-Colleges-SUCs-Faculty-Student-Ratio.pdf>

SearchStorage. (2019). *What is cloud storage? - Definition from WhatIs.com*. [online] Available at: <https://searchstorage.techtarget.com/definition/cloud-storage> [Accessed 17 Jul. 2019].

Spotherd.com. (2019). *Global Public Cloud Storage Service Market Regional Outlook 2019 – Spot Herld*. [online] Available at:

<https://spotherld.com/2019/04/10/global-public-cloud-storage-service-market-regional-outlook-2019/> [Accessed 17 Jul. 2019].

Storage Servers. (2019). *Understanding why Private Cloud Storage works for your business*. [online] Available at:
<https://storageservers.wordpress.com/2016/10/24/understanding-why-private-cloud-computing-works-for-your-business/> [Accessed 17 Jul. 2019].

Techopedia.com. (2019). *What is a Hard Disk Drive (HDD)? - Definition from Techopedia*. [online] Available at:
<https://www.techopedia.com/definition/5288/hard-disk-drive> [Accessed 17 Jul. 2019].

Techopedia.com. (2019). *What is a Scanner? - Definition from Techopedia*. [online] Available at: <https://www.techopedia.com/definition/30441/scanner> [Accessed 17 Jul. 2019].

Thilakarathne, Ashan. (2019). Tesseract: Simple Java Optical Character Recognition.

Tim Layton Fine Art. (2019). *25% Off Your First Drum Scanning Order (35mm up to 8x10 Large Format)*. [online] Available at:
<https://www.timlaytonfineart.com/blog/2019/1/25-off-your-first-drum-scanning-order-35mm-up-to-8x10-large-format> [Accessed 17 Jul. 2019].

Vishal Chourasia, Sanjay Silakari, Rajeev Pandey, (2018). Implementation of Optical Character Recognition Using Machine Learning. *International Journal of Computer Sciences and Engineering*, 6(6), 1350-1356.

Webopedia.com. (2019). *What is a Detachable Screen Laptop? Webopedia Definition*. [online] Available at:
<https://www.webopedia.com/TERM/D/detachable-touchscreen.html> [Accessed 17 Jul. 2019].

Webopedia.com. (2019). *What is Wi-Fi (Wireless)? Webopedia Definition*. [online]
Available at: https://www.webopedia.com/TERM/W/Wi_Fi.html [Accessed 17
Jul. 2019].

Yu, Z., Yuan, C., & Zheng, K. (2018). A University Fixed Asset Database
Information Management System Based on Internet of Things. 2018 2nd IEEE
Advanced Information Management,Communicates,Electronic and Automation
Control Conference (IMCEC). doi:10.1109/imcec.2018.8469407

