IBIS MANAGEMENT SYSTEM

NURAIN ALEEYA BINTI CHE ZAHARUDIN

BACHELOR OF COMPUTER SCIENCE (SOFTWARE ENGINEERING) WITH HONOUR

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	: NURAIN ALEEYA BINTI CHE ZAHARUDIN
Date of Birth	
Title	: IBIS MANAGEMENT SYSTEM
Academic Session	: YEAR 3 SEMESTER 2 2022/2023

I declare that this thesis is classified as:

☐ CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐ RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☑ OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

_______________________ (Student's Signature)

_______________________ (Supervisor's Signature)

____
New IC/Passport Number
Date: 23/07/2023

TS DR ABDUL SAHLI BIN FAKHARUDIN
____
Name of Supervisor
Date: 23/07/2023

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor in Computer Science (Software Engineering) with Honour.

____

(Supervisor's Signature)

Full Name	: TS DR ABDUL SAHLI BIN FAKHARUDIN
Position	: Lecturer
Date	: 23/07/2023

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

____

(Student’s Signature)

Full Name	: NURAIN ALEEYA BINTI CHE ZAHARUDIN
ID Number	: CB20164
Date	: 12 JUNE 2023

IBIS Management System

Nurain Aleeya binti Che Zaharudin

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Computer Science (Software Engineering) with Honour

Faculty of Computing

UNIVERSITI MALAYSIA PAHANG

ACKNOWLEDGEMENTS

First and foremost, I would like to thank you to my supervisor, Dr. Sahli for the invaluable advice, and guidance throughout the project. Dr Sahli likely provided valuable feedback on my research and writing in order to finish the FYP. Next, I want to sincerely thank my parents for their unflagging love and support throughout my academic life. Their support and confidence in me have served as a constant source of inspiration and motivation. Without their love and support, I would not have been able to achieve this. I also want to express my gratitude to my friends for their encouragement and support. Their friendship has served as a constant source of motivation and inspiration, and their insights and criticism of my work have been priceless. I appreciate the many hours we spent talking about my project as well as the joy and encouragement they gave.

ABSTRAK

Sistem Pengurusan IBIS adalah sistem yang luas yang direka untuk mengurus pelbagai aspek organisasi, seperti pendaftaran, pengurusan kewangan, penyelenggaraan tugas-tugas, dan penjanaan laporan. Modul pendaftaran sistem ini membolehkan pengurusan pendaftaran pengguna baru yang efektif, termasuk pengumpulan data untuk butir-butir peribadi dan hubungan serta pengurusan akaun. Modul pengurusan kewangan membolehkan pengurusan pendapatan dan perbelanjaan, pemantauan transaksi kewangan, dan penciptaan laporan kewangan. Pengurusan tugas-tugas penyelenggaraan, termasuk penjadualan, pemantauan kemajuan, dan laporan, boleh dilakukan melalui modul aktiviti penyelenggaraan. Sistem juga mempunyai modul untuk mencipta laporan yang boleh digunakan untuk menganalisis pelbagai aspek bagaimana organisasi beroperasi, seperti prestasi kewangan, tugas-tugas penyelenggaraan, dan pendaftaran pengguna. Antara muka pengguna sistem Pengurusan IBI mudah dikendalikan dan menawarkan gambaran menyeluruh tentang operasi organisasi. Ia akan membantu meningkatkan prestasi organisasi secara keseluruhan, produktiviti dan kecekapan.

ABSTRACT

The IBIS Management system is an extensive system designed to manage various organisational aspects, such as registration, financial management, maintenance tasks, and report generation. The registration module of the system enables effective management of new user registration, including data collection for personal and contact details as well as account management. The financial management module enables the management of income and expenses, and the creation of financial reports. The management of maintenance tasks, including scheduling, tracking of progress, and reporting, is made possible by the maintenance activity module. The system also has a module for creating reports that can be used to analyse different aspects of how the organisation operates, such as its financial performance, upkeep tasks, and user registration. The user-friendly interface of the IBIS Management system makes it simple to navigate and offers a thorough overview of the organization's operations. It will support enhancing an organization's performance overall, productivity, and efficiency.

Table of Contents

CHAPTER 1 17

INTRODUCTION 17

1.1 Background Study 17

1.2 Problem Statement 18

1.3 Aim and Objective 19

1.4 Scope 19

1.5 Significant of this projects 19

CHAPTER 2 20

LITERATURE REVIEW 20

2.1 Introduction 20

2.2 Comparative Analysis 32

2.3 Proposed Project 35

2.4 Summary 35

CHAPTER 3 36

METHODOLOGY 36

3.1 Introduction 36

3.2 Methodology 37

3.3 Project Requirement 39

3.4 Propose Design 42

3.5 Data Design 55

3.6 Proof of Initial Concept 60

3.7 Testing Plan 68

3.8 Potential Use of Proposed Solution 69

3.9 Gantt Chart 70

CHAPTER 4 71

RESULTS AND DISCUSSION 71
4.1 Introduction 71
4.2 Implementation 71
4.3 Testing and Result 94
CHAPTER 5 99
CONCLUSION 99
5.1 Introduction 99
5.2 Project Constraint 99
5.3 Future Works 100
References 101

List of Figure

Figure 1 : Zoho Homepage 20
Figure 2 : Zoho Dashboard (People) 21
Figure 3 : Attendance Part (Check-in) 21
Figure 4 : Attendance Part (Check-out) 21
Figure 5 : Pending Job (Empty) 22
Figure 6 : Pending Job (Listed) 22
Figure 7: Add Pending Job 22
Figure 8 :Dashboard for Invoice Page 23
Figure 9 : Invoice Form (Create Invoice) 23
Figure 10 : Invoice 24
Figure 11 : Niagawan Homepage 25
Figure 12 : Niagawan Login Page 25
Figure 13 : Niagawan Dashboard 26
Figure 14 : Sales Invoice 26
Figure 15: Create New Invoice 27
Figure 16 : Invoice 27
Figure 17 : Register Employee Page 28
Figure 18 : List of User Page 28
Figure 19 : Attendance Page 29
Figure 20 : Checkin Page 29
Figure 21 : Check Out System 29
Figure 22 : Invoice Page 30
Figure 23 : Add New Invoice 30
Figure 24 : Email Template for Invoice 30
Figure 25 : Planning Interface 31
Figure 26: Project List 31
Figure 27: Create Project Interface 31
Figure 28 : Rapid Application Development 37
Figure 29 : Context Diagram of IBIS Management System 42
Figure 30 : Use case for IBI Management System 42
Figure 31 : Usecase of Manage Registration 43
Figure 32 : Flowchart for Administration 52
Figure 33 : Flowchart for Accountant 53
Figure 34: Flow Chart for Supervisor and Worker 54
Figure 35 : Entity Relationship Diagram of IBIS System 55
Figure 36 : Interface of Sign Page 60
Figure 37 : Interface of registration page 61
Figure 38 : Interface of Manage User page 62
Figure 39 : Interface of Dashboard Page 63
Figure 40 : Interface of Invoice Page 64
Figure 41 : Interface of Finance page 64
Figure 42 : Interface of Claim Page 65
Figure 43 : Interface of Maintenance Services Page 66

Figure 44 : Interface of Report Page 67
Figure 45 : Login Page 72
Figure 46: Login Codes 73
Figure 47 : Login codes 73
Figure 48 : Dashboard Page 74
Figure 49 : Dashboard Codes 74
Figure 50 : Dashboard Codes 75
Figure 51 : Dashboard Codes 75
Figure 52 : Manage User Page 76
Figure 53 : Manage User Codes 76
Figure 54 : Edit User Information 77
Figure 55 : Edit User Codes 78
Figure 56: Edit User Codes 78
Figure 57 : Delete User Alert 79
Figure 58 : Delete Codes 80
Figure 59 : My Profile Page 81
Figure 60 : My Profile Codes 82
Figure 61 : My Profile Codes 82
Figure 62 : Modal Popup Change Profile Picture 83
Figure 63 : Modal Profile Picture Codes 83
Figure 64 : Modal Profile Picture Codes 84
Figure 65 : Manage Claim Interface 85
Figure 66 : Add New Claim Interface 85
Figure 67 : Manage Invoice Interface 86
Figure 68 : Add New Invoice Interface 86
Figure 69 : Attendance Interface 87
Figure 70 : Check Out Interface 87
Figure 71 : Web Codes 88
Figure 72 : User Controller Codes 89
Figure 73 : User Controller Codes 89
Figure 74 : User Controller Codes 90
Figure 75 : Database ibis_ms 91
Figure 76 : Table Users 91
Figure 77: Table invoice 92
Figure 78 : Tables Jobs 92
Figure 79 : Table item 92
Figure 80 : Table Report 93
Figure 81 : Attendance table 93

List of Table

Table 1 : Summary of comparison two of existing system 32
Table 2 : Advantages and Disadvantages of existing system 33
Table 3 : Comparison System 35
Table 4 : Software Requirement 39
Table 5 : Hardware Requirement 39
Table 6 : : Use case of Manage Registration 43
Table 7 : : Use case of Manage Financial resource 45
Table 8 : : Use case of Maintenance and Report 47
Table 9 : Use case of Manage Attendance 49
Table 8 : : Use case of Invoice 50
Table 10 : Data Dictionary of User 56
Table 11 : Data Dictionary of Report 56
Table 12 : Data Dictionary of Invoice 57
Table 13 : Data Dictionary of Claim 57
Table 14 : Data Dictionary of Attendance 58
Table 16 : Data Dictionary of Job 58
Table 17 : Data Dictionary of Company 58
Table 18 : Data Dictionary of Item 59

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

1.	FYP	Final Year Project
2.	UMP	Universiti Malaysia Pahang
3.	IBIS	IBI Services
4.	UAT	User Acceptance Test
5.	RAD	Rapid Application Development
6.	IBIS-ms	IBIS Management System

CHAPTER 1
INTRODUCTION

1.1 Background Study

In today's era, various sectors of the economy are experiencing rapid expansion alongside the progress of technology within the country. The presence of advanced technology enables a nation to obtain information from around the globe, offering diverse perspectives through the internet. Consequently, it has been established that technology significantly facilitates the day-to-day activities of modern society, encompassing work, education, and other endeavors. Looking back, our nation used to lag significantly due to deteriorating internet infrastructure and numerous system and server failures(Europe House, 2023). However, it can now be affirmed that people throughout Malaysia can effortlessly access the internet using a wide range of devices such as phones, laptops, computers, and others, without encountering any problems.

With the progression of technology and the internet in our nation, numerous businesses and economic sectors have seized the opportunity to establish technology companies that cater to their specific organizational needs. A notable example of such systems is Niagawan.com, which has recently been developed. The proud owner of Niagawan.com is Niagawan Plus Sdn Bhd, located in Bayan Lepas, Penang.(*Niagawan | Penyelesaian Masalah Akaun Bisnes Anda*, n.d.) Having served 20,000 entrepreneurs in the country, this business, consisting of a team of 23 individuals, has achieved remarkable success. Notable clients, including Noor Arfa and DentaBay, have found value in utilizing the Niagawan system. The primary objective behind the development of this system was to provide a seamless and effective solution for managing sales business accounts, customer orders, and product inventory. Moreover, it offers valuable features for handling expenses, such as expense records and monitoring the company's cash flow(*About Journey - Niagawan*, n.d.).

Each business adopts its own distinctive approach to management. As you may have noticed, many businesses rely on a substantial workforce to oversee their operations. This is primarily because involving multiple individuals in each process is essential to ensure transparency, particularly in managing company finances. Research findings suggest that having a well-established system greatly facilitates disciplined and organized business operations for any organization. For instance, the current system enables accurate tracking of financial records by documenting every inflow and outflow of funds.

1.2 Problem Statement

IBIS Services, a contractor company specializing in supplying manpower for maintenance projects, faces several challenges in managing their workforce and crucial documents. The company's operations involve deploying a large number of workers to various job sites, necessitating efficient management and scheduling based on individual maintenance specialties. IBIS Services requires a comprehensive system that can streamline staff management and document handling, including reports, invoices, and more.

Currently, IBIS Services employees must complete manual, handwritten forms to confirm their attendance at work and obtain a manager's signature. This process becomes cumbersome as workers need to fill in details such as job site names, arrival and end times, and then scan and send the form to the duty supervisor. The supervisors, in turn, encounter difficulties in downloading and manually reviewing the submitted forms.

Additionally, IBIS Services employees frequently need to communicate with their superiors to inquire about available positions each morning. Obtaining timely information from the current site becomes crucial for effective operations. To avoid confusion, employees should be able to view assigned tasks through the developed system, with IBIS Services staff updating the Maintenance Services section the day before.

Regarding IBIS Services management, there are inefficiencies in accessing employee information and bank details, specifically when managing claims. Locating this information in the employee book consumes valuable time and can adversely impact employee finances. Moreover, staff members are required to individually review and verify each overtime form filled out by employees. Monthly payments for overtime work rely on the accuracy of these forms, submitted to management for verification. Additionally, the system needs to accommodate claims for advanced employee expenses incurred while purchasing necessary tools during the working period.

Addressing these challenges is crucial to enhance workforce management, streamline administrative processes, and ensure accurate financial management for IBIS Services. Thus, the development of a comprehensive system is imperative to overcome these obstacles and improve overall operational efficiency.

1.3 Aim and Objective

The aim of the IBIS-ms is to enhance efficiency and productivity by streamlining and automating processes. It seeks to optimize resource allocation, minimize manual tasks, and improve overall operational effectiveness. Additionally, the system aims to establish a structured framework that organizes and coordinates activities within the organization.

These Projects consist of three objectives which are:

1. To study the solution to a management problem.
2. To develop a web-based system called IBIS Services Management System.
3. To test a system with many functions that can be used by IBIS Company.

1.4 Scope

This project has several distinct scopes, some of which are listed below.

1. This system will be used by IBIS Company employees only.
2. There are five modules in this system whose primary purpose is to serve users.
3. The IBIS company itself is the source of all information pertaining to this system.
4. This system will be developed as a web-based system.
5. The structure of this system will be discussed between the developers and IBIS management staff.

1.5 Significant of this projects

Chapter 1: Discusses the introduction of the web-based system. This chapter will be covered the overview of the IBIS-ms, the problem statement, the goal and objective, and also the scope of the project.

Chapter 2: This chapter consists of a literature review of three existing systems that are related to management systems. It will briefly explain how to use the existing system one by one and it also explains the function that has in the existing system. There will be a comparison including the advantages and disadvantages between the existing system and the proposed system.

Chapter 3: This chapter will discuss the methodology used in developing the IBIS-ms. For example, this system will be developed using the RAD method. Besides, it also will be focused on the software and hardware used while developing this system.

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

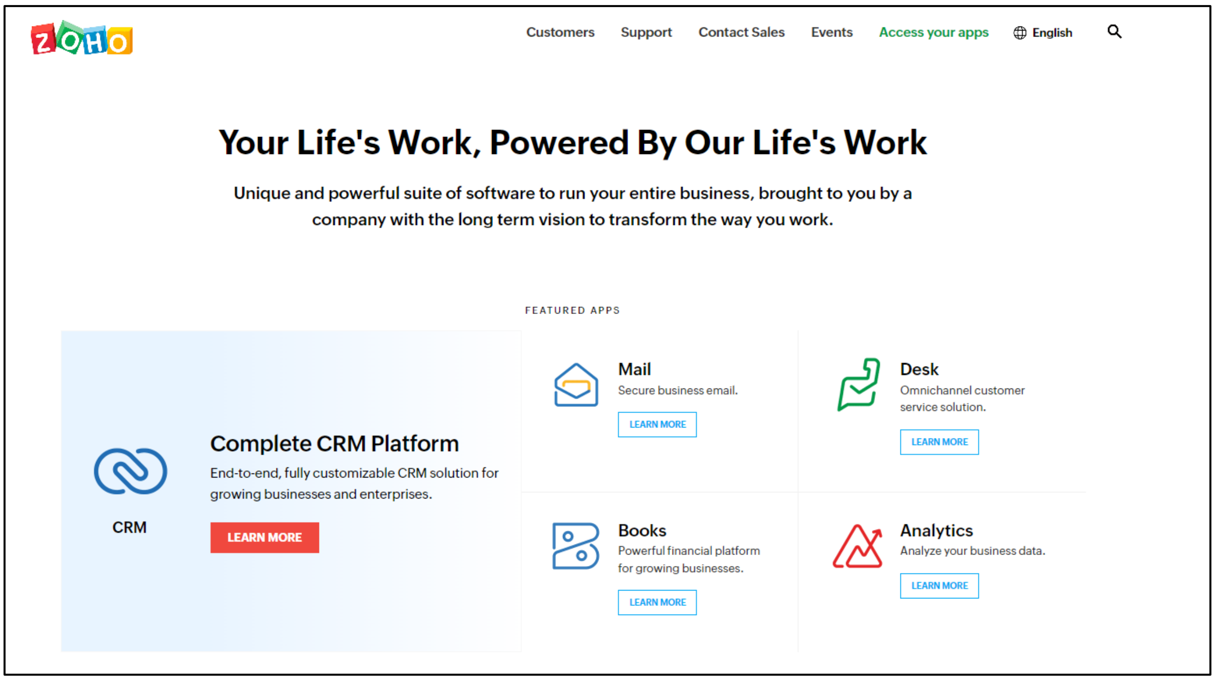
This chapter will be explained the project or existing system that is related to Company management System. The main purpose of this chapter is to review the existing system and explore its function have in that system. There are many management systems that can be used for the company to make their company well-managed on the internet. However, IBIS will be developed for a company named IBIS Services Sdn Bhd. It means this company wants to have its own system that they are not required to sign up for any current systems.

2.1.1 Review of Existing System

2.1.1.1 Zoho

- **Employee**

A web-based system called Zoho was created to simplify small business operations by helping them deal with issues like employee management for human resources and other issues. The ability to manage and issue basic invoices and receipts with various information entered is another significant benefit of using Zoho for sales and marketing. By using Zoho software to create and deliver marketing to the audience, all marketers can decrease the amount of time required to manage the business(*Zoho - Cloud Software Suite and SaaS Applications*, n.d.).



Your Life's Work, Powered By Our Life's Work

Unique and powerful suite of software to run your entire business, brought to you by a company with the long term vision to transform the way you work.

Complete CRM Platform
End-to-end, fully customizable CRM solution for growing businesses and enterprises.

Secure business email.

Omnichannel customer service solution.

Powerful financial platform for growing businesses.

Analyze your business data.

Figure 1 : Zoho Homepage

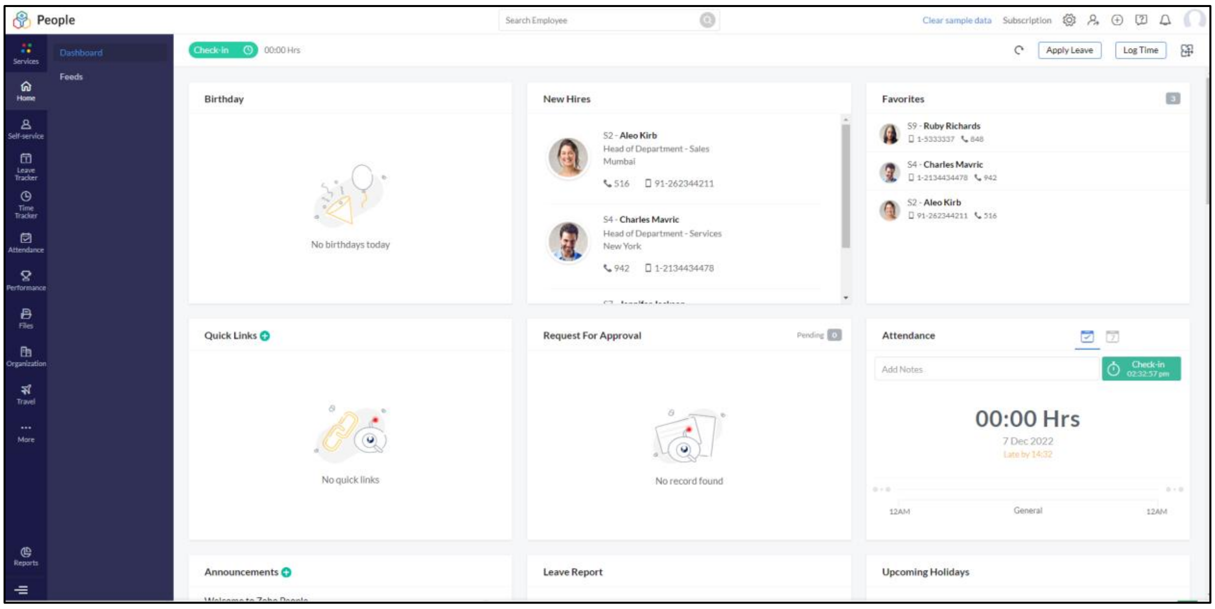


Figure 2 : Zoho Dashboard (People)

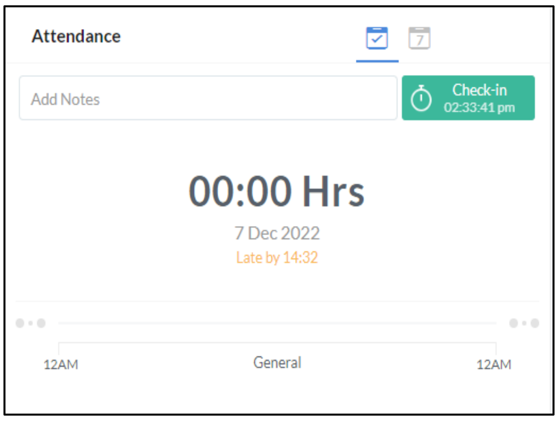


Figure 3 : Attendance Part (Check-in)

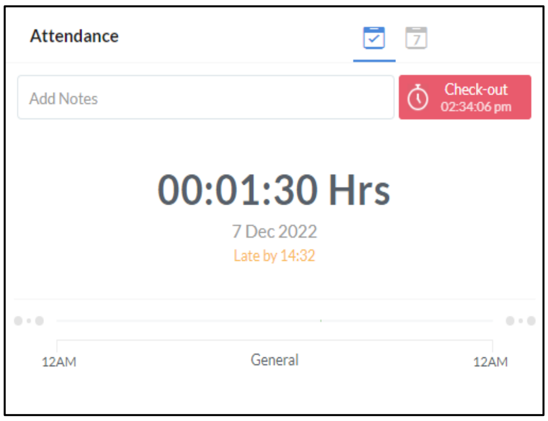


Figure 4 : Attendance Part (Check-out)

The Zoho Dashboard is shown in Figure 2 above and is made up of a number of sections, including Birthday, New Hires, Favourite, Quick Link, and others. The user can select from a wide variety of options in the left navigation on each page. Employees in human resources can click any button to go directly to the page they've chosen. The attendance for that staff member will be tracked when they click the check- in button in the attendance section. Figures 3 and 4 show how the button will change to the "check-out" option once the staff clicks it.

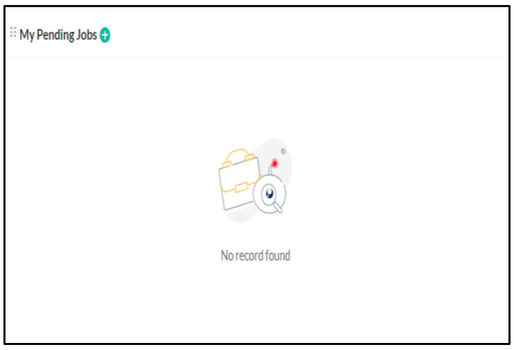
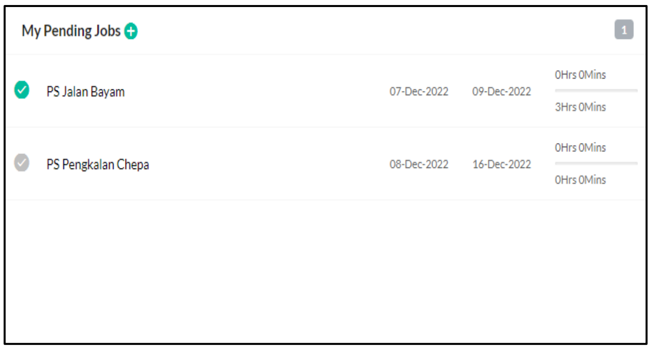
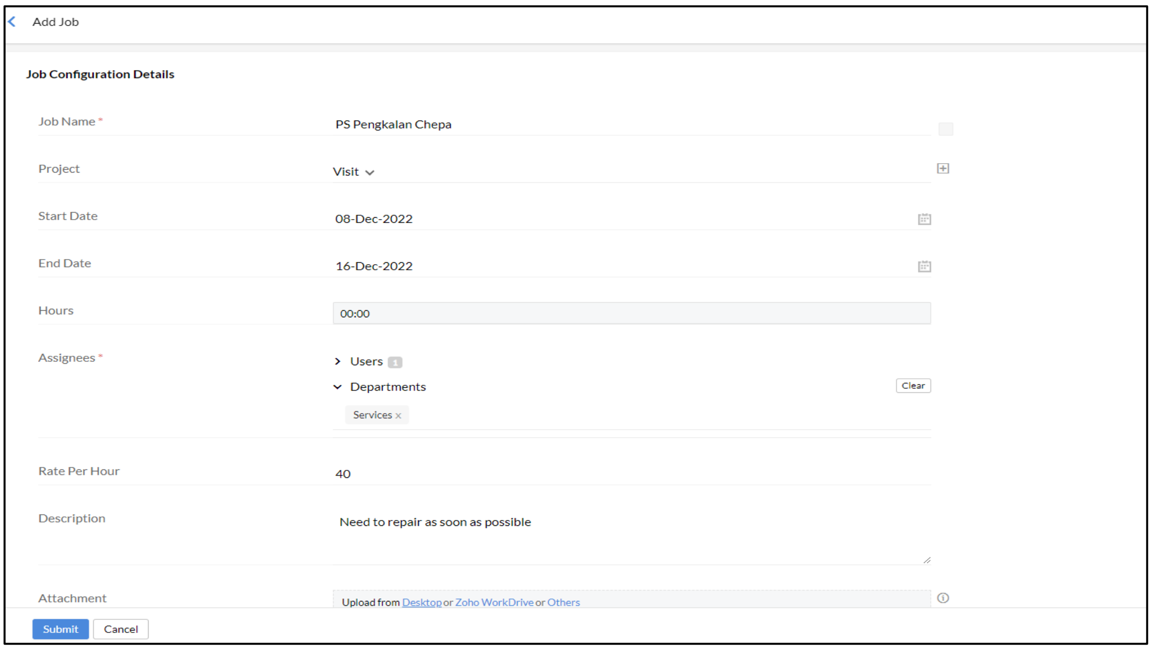


Figure 5 : Pending Job (Empty)


My Pending Jobs

✓	PS Jalan Bayam	07-Dec-2022	09-Dec-2022	0Hrs 0Mins / 3Hrs 0Mins
✓	PS Pengkalan Chepa	08-Dec-2022	16-Dec-2022	0Hrs 0Mins / 0Hrs 0Mins

Figure 6 : Pending Job (Listed)


Add Job

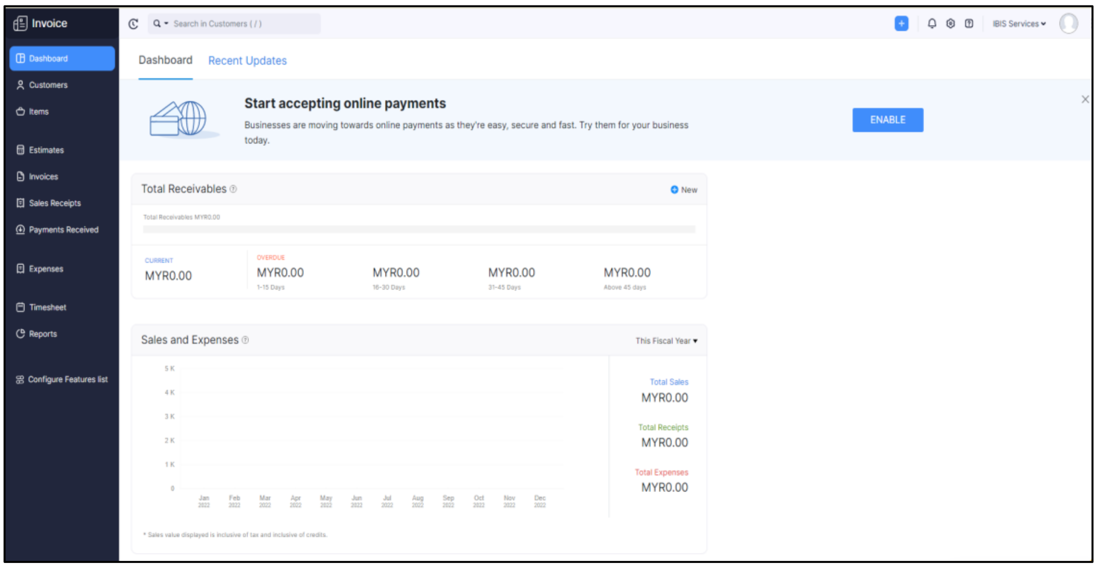
Job Configuration Details

Job Name *	PS Pengkalan Chepa
Project	Visit
Start Date	08-Dec-2022
End Date	16-Dec-2022
Hours	00:00
Assignees *	Users 1; Departments: Services
Rate Per Hour	40
Description	Need to repair as soon as possible
Attachment	Upload from Desktop or Zoho WorkDrive or Others

Figure 7: Add Pending Job

Figure 5 and 6 shows the “My Pending Jobs” sections that have the employee’s dashboard. On figure 5 shows no record found if there is no job assigned to the staff meanwhile if there is a job pending it will show as in figure 6. The employee can click on the tick button if the job is completed. It is because the completed job will not appear the next day. Besides, the form in figure 7 shows the information required when the employee wants to add the pending jobs. They need to enter the Job Name, Project, Start and End date, Assignees and etc. The employee needs to click submit button and the information will have appeared in the “My Pending Job’ section.

- **Company Staff (Accountant)**



Start accepting online payments

Businesses are moving towards online payments as they're easy, secure and fast. Try them for your business today.

Total Receivables

Total Receivables MYR0.00

CURRENT	OVERDUE			
MYR0.00	MYR0.00	MYR0.00	MYR0.00	MYR0.00
	1-15 Days	16-30 Days	31-45 Days	Above 45 days

Sales and Expenses

Total Sales MYR0.00

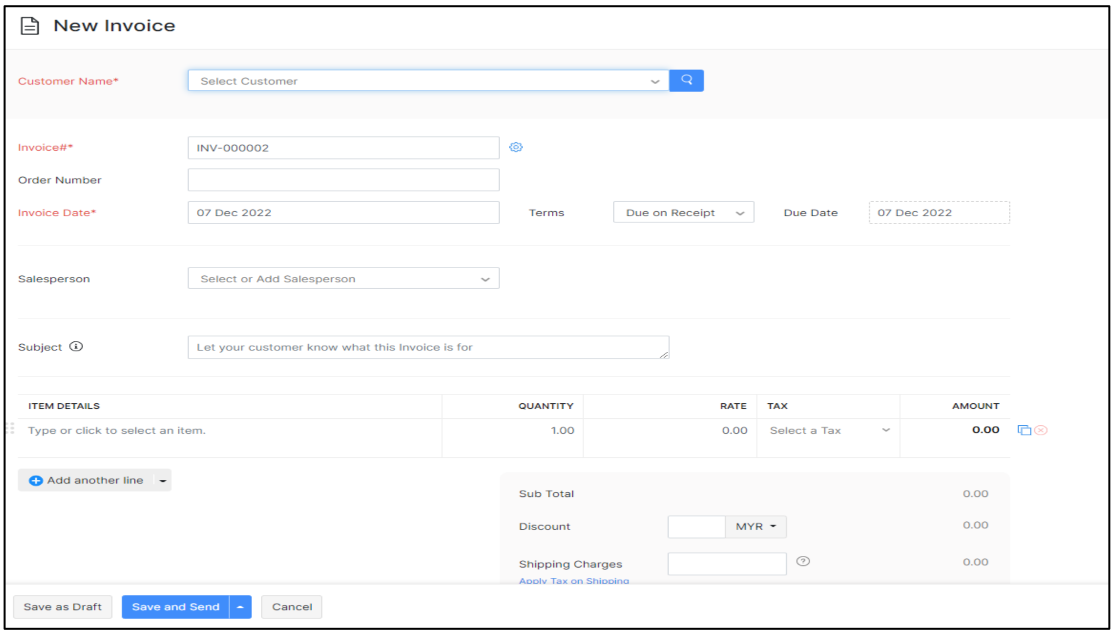
Total Receipts MYR0.00

Total Expenses MYR0.00

* Sales value displayed is inclusive of tax and inclusive of credits.

Figure 8 :Dashboard for Invoice Page

The invoice page will be used for company staff that consists including the Human Resources department and financial department. Because it contains business secrets like cash flow, expenses, and other financial information, this page is a view that only certain people can access. A breakdown of business expenses, including cash in and cash out, is shown in Figure 8. The functionality in the invoice part is displayed in the navigation bar on the left. The system will be able to produce reports, receipts, and invoices.



New Invoice

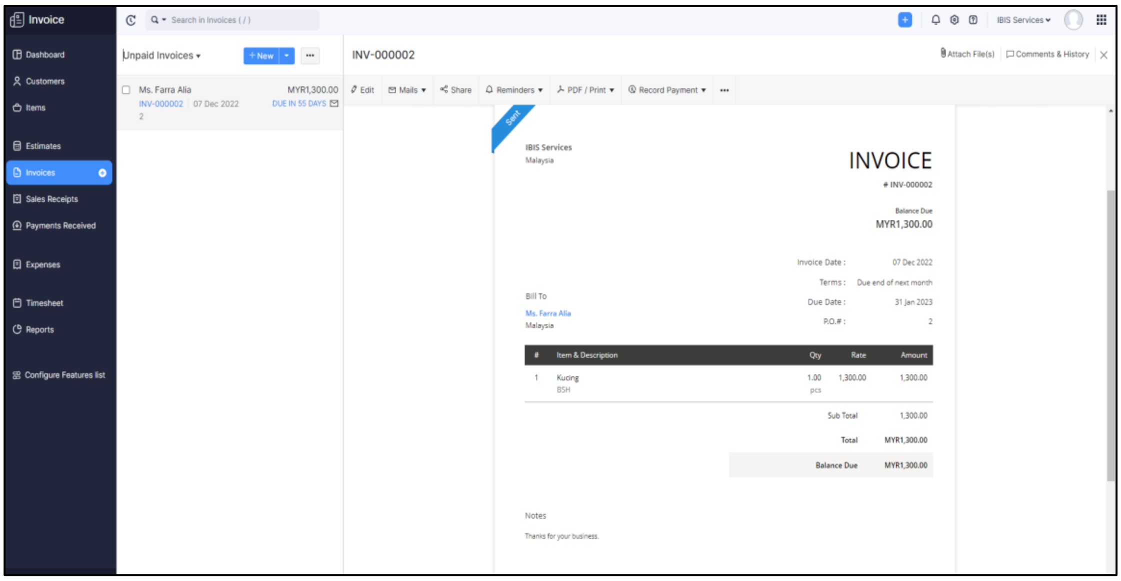
Customer Name*	Select Customer		
Invoice#*	INV-000002		
Order Number			
Invoice Date*	07 Dec 2022	Terms: Due on Receipt	Due Date: 07 Dec 2022
Salesperson	Select or Add Salesperson		
Subject	Let your customer know what this Invoice is for		

ITEM DETAILS	QUANTITY	RATE	TAX	AMOUNT
Type or click to select an item.	1.00	0.00	Select a Tax	0.00

Sub Total		0.00
Discount	MYR	0.00
Shipping Charges		0.00

Figure 9 : Invoice Form (Create Invoice)

Figure 9 depicts the form that must be completed in order to create an invoice for the client. Users must choose customers and enter all necessary data, including item details, price, quantity, and salesperson. The system will then automatically calculate and display the total price, and if the item has a discount, the user can also enter the discount at this point.



Ms. Farra Alia MYR1,300.00
INV-000002 | 07 Dec 2022 DUE IN 55 DAYS
2

INV-000002

IBIS Services
Malaysia

INVOICE

INV-000002

Balance Due
MYR1,300.00

Invoice Date :	07 Dec 2022
Terms :	Due end of next month
Due Date :	31 Jan 2023
P.O.# :	2

Bill To
Ms. Farra Alia
Malaysia

#	Item & Description	Qty	Rate	Amount
1	Kucing BSH	1.00 pcs	1,300.00	1,300.00

Sub Total	1,300.00
Total	MYR1,300.00
Balance Due	MYR1,300.00

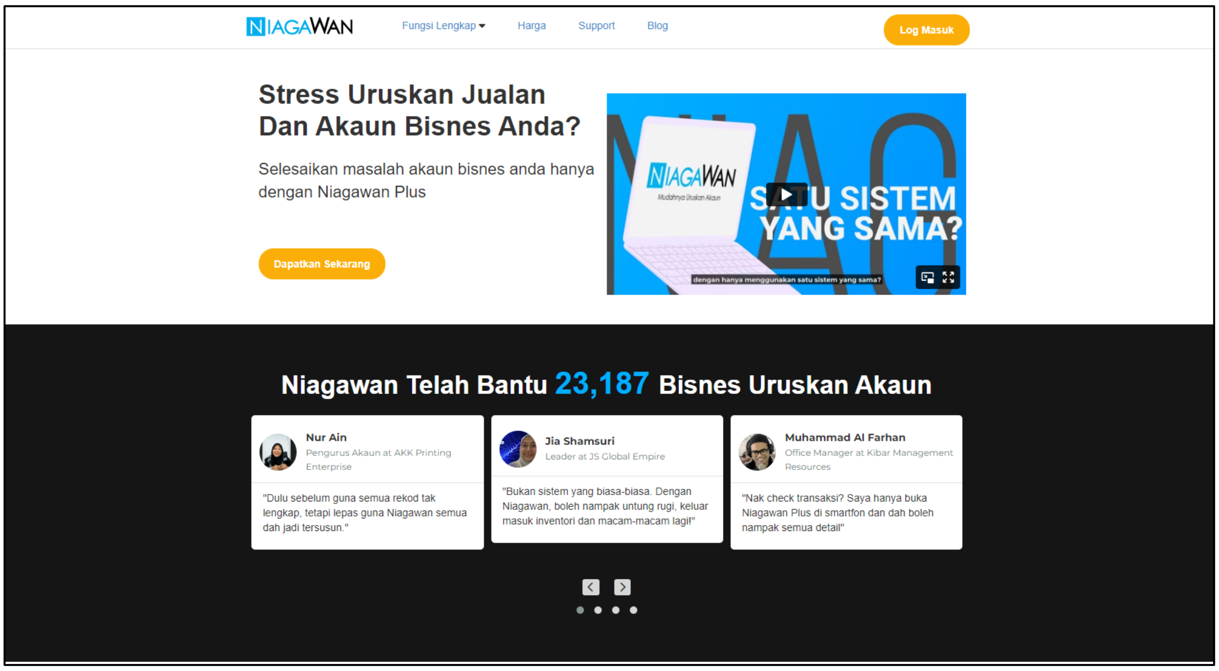
Notes
Thanks for your business.

Figure 10 : Invoice

The invoice that was generated by the system is displayed in Figure 10. This page will display all the information that we entered on the invoice form. The system will also display a section for emailing the customer with the invoice. The user only needs to click the send button because the system automatically recognizes the customer's email. The system has already set up the email template.

2.1.1.2 Niagawan

Small business owners use the website Niagawan to run their operations. Many business owners have been drawn to use this website because of its many features and appealing price offers. Sales and invoices, customer information management and storage, stock and inventory management, damage calculation, and account reports are some of the features offered by this system. With the help of the available cloud, this system can be used online and from any location. It has automatic data backup and is compatible with both laptops and smartphones. More than ten users may be registered by a single business, and employees may keep track of sales(*Niagawan - Accounting System for Non-Accountants*, n.d.).


NIAGAWAN Fungsi Lengkap Harga Support Blog Log Masuk

Stress Uruskan Jualan Dan Akaun Bisnes Anda?

Selesaikan masalah akaun bisnes anda hanya dengan Niagawan Plus

Dapatkan Sekarang

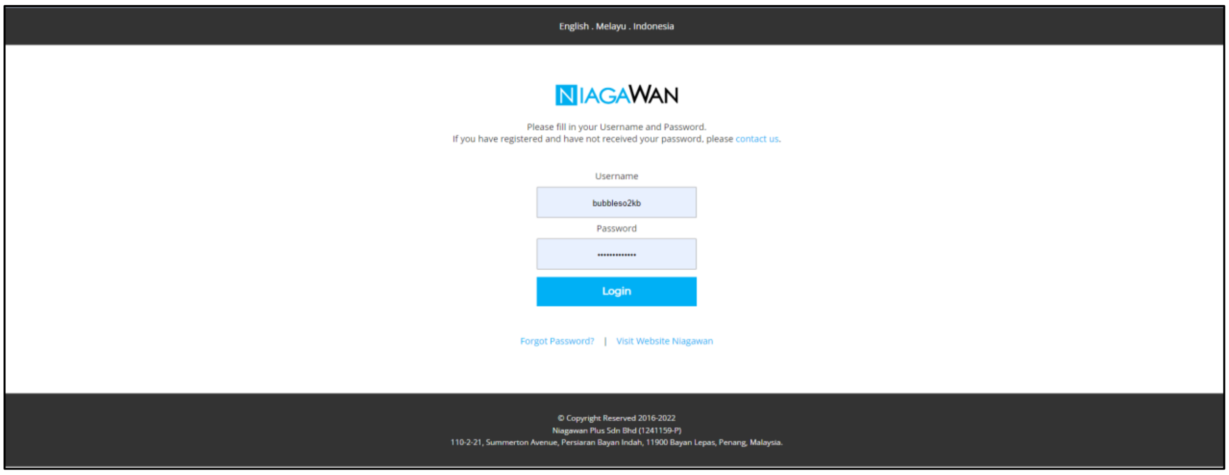
Niagawan Telah Bantu 23,187 Bisnes Uruskan Akaun

Nur Ain
Pengurus Akaun at AKK Printing Enterprise
"Dulu sebelum guna semua rekod tak lengkap, tetapi lepas guna Niagawan semua dah jadi tersusun."

Jia Shamsuri
Leader at JS Global Empire
"Bukan sistem yang biasa-biasa. Dengan Niagawan, boleh nampak untung rugi, keluar masuk inventori dan macam-macam lagi!"

Muhammad Al Farhan
Office Manager at Kibar Management Resources
"Nak check transaksi? Saya hanya buka Niagawan Plus di smartfon dan dah boleh nampak semua detail"

Figure 11 : Niagawan Homepage


English . Melayu . Indonesia

NIAGAWAN

Please fill in your Username and Password.
If you have registered and have not received your password, please contact us.

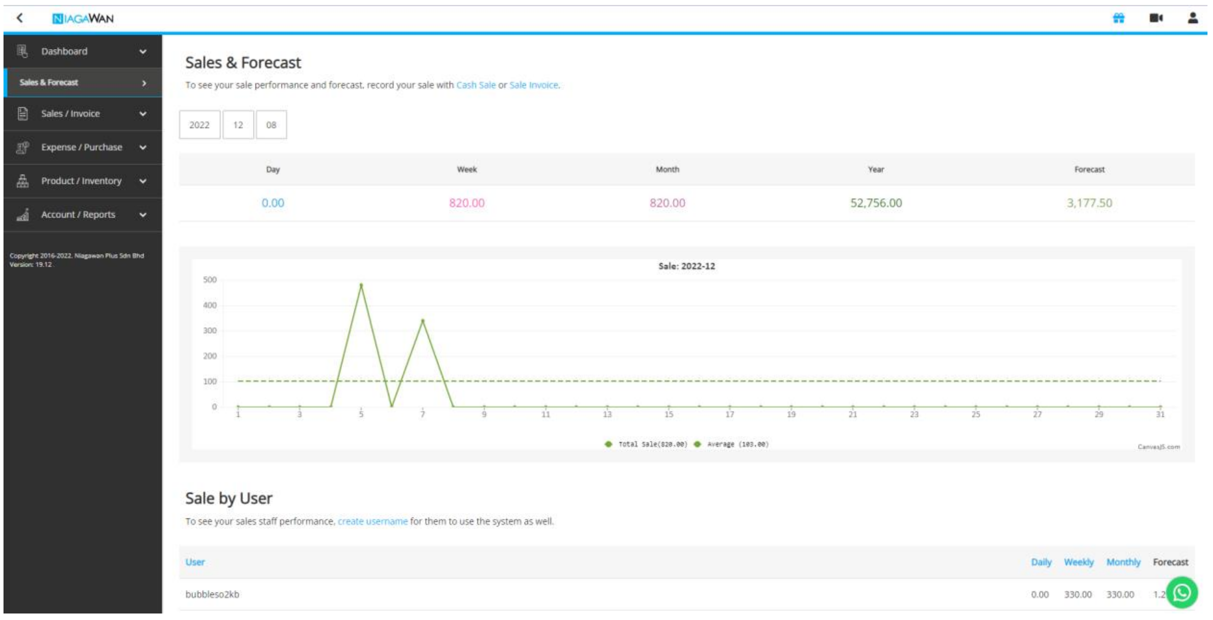
Username: bubbleso2kb
Password: ••••••••••••••

Login

Forgot Password? | Visit Website Niagawan

© Copyright Reserved 2016-2022
Niagawan Plus Sdn Bhd (1241159-P)
110-2-21, Summerton Avenue, Persiaran Bayan Indah, 11900 Bayan Lepas, Penang, Malaysia.

Figure 12 : Niagawan Login Page



Sales & Forecast

To see your sale performance and forecast, record your sale with Cash Sale or Sale Invoice.

2022 | 12 | 08

Day	Week	Month	Year	Forecast
0.00	820.00	820.00	52,756.00	3,177.50

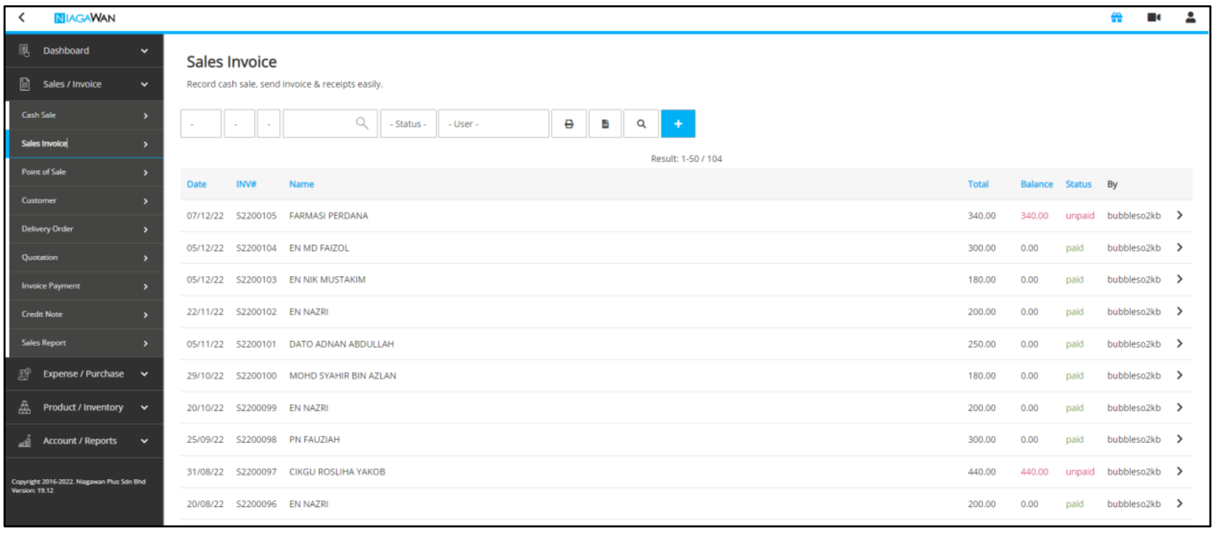
Sale by User

To see your sales staff performance, create username for them to use the system as well.

User	Daily	Weekly	Monthly	Forecast
bubbleso2kb	0.00	330.00	330.00	1.2

Figure 13 : Niagawan Dashboard

Figure 13 displays a graph or summary of the business's most recent month's sales. On this dashboard, companies can view the number of profits by day, week, month, or year. Users can also choose a specific date to view that day's total sales. The user can view the customer's choice of payment method—cash or transfer—by scrolling down to that section of the system. The navigation bar on the left side shows there are several functions for this system which is the system can generate the invoice, Receipts, Products and inventory, and reports.



Sales Invoice

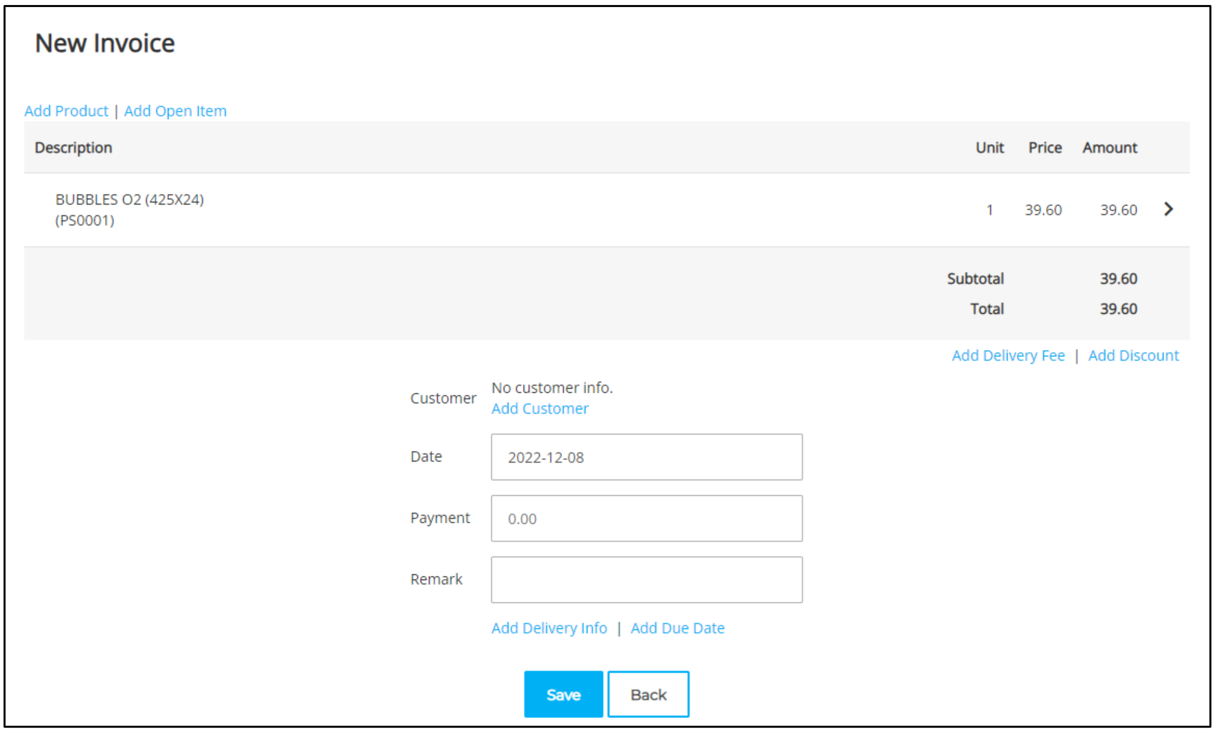
Record cash sale, send invoice & receipts easily.

Result: 1-50 / 104

Date	INV#	Name	Total	Balance	Status	By
07/12/22	S2200105	FARMASI PERDANA	340.00	340.00	unpaid	bubbleso2kb
05/12/22	S2200104	EN MD FAIZOL	300.00	0.00	paid	bubbleso2kb
05/12/22	S2200103	EN NIK MUSTAKIM	180.00	0.00	paid	bubbleso2kb
22/11/22	S2200102	EN NAZRI	200.00	0.00	paid	bubbleso2kb
05/11/22	S2200101	DATO ADNAN ABDULLAH	250.00	0.00	paid	bubbleso2kb
29/10/22	S2200100	MOHD SYAHIR BIN AZLAN	180.00	0.00	paid	bubbleso2kb
20/10/22	S2200099	EN NAZRI	200.00	0.00	paid	bubbleso2kb
25/09/22	S2200098	PN FAUZIAH	300.00	0.00	paid	bubbleso2kb
31/08/22	S2200097	CIKGU ROSLIHA YAKOB	440.00	440.00	unpaid	bubbleso2kb
20/08/22	S2200096	EN NAZRI	200.00	0.00	paid	bubbleso2kb

Figure 14 : Sales Invoice

The sales invoice that the company has issued is depicted in Figure 14. A list of clients who have received successful invoices for their purchases is shown in the figure. This page also shows the balance, total payment, and whether the invoice has been paid or not.


New Invoice

Add Product | Add Open Item

Description	Unit	Price	Amount
BUBBLES O2 (425X24) (PS0001)	1	39.60	39.60
Subtotal			39.60
Total			39.60

Add Delivery Fee | Add Discount

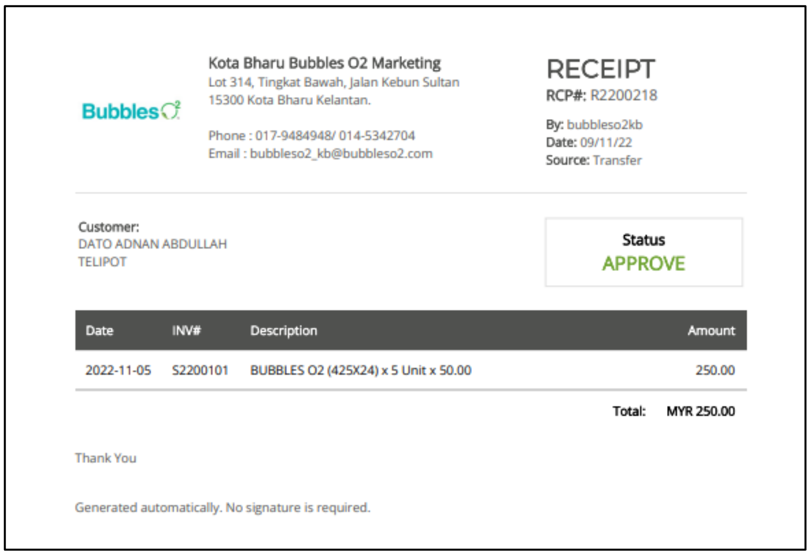
Customer	No customer info. Add Customer
Date	2022-12-08
Payment	0.00
Remark	

Add Delivery Info | Add Due Date

Save Back

Figure 15: Create New Invoice

The information required when a user wants to create a customer invoice is shown in Figure 15. Customers' information must be added before the data is entered. They can add products, and a list of the products this company offers is displayed in the pop-up. The invoice will show the product you've chosen. If a customer chooses the cash-on-delivery payment option, users can enter the amount of the payment.


Bubbles O2

Kota Bharu Bubbles O2 Marketing
Lot 314, Tingkat Bawah, Jalan Kebun Sultan
15300 Kota Bharu Kelantan.

Phone : 017-9484948/ 014-5342704
Email : bubbleso2_kb@bubbleso2.com

RECEIPT
RCP#: R2200218
By: bubbleso2kb
Date: 09/11/22
Source: Transfer

Customer:
DATO ADNAN ABDULLAH
TELIPOT

Status
APPROVE

Date	INV#	Description	Amount
2022-11-05	S2200101	BUBBLES O2 (425X24) x 5 Unit x 50.00	250.00
		Total:	MYR 250.00

Thank You

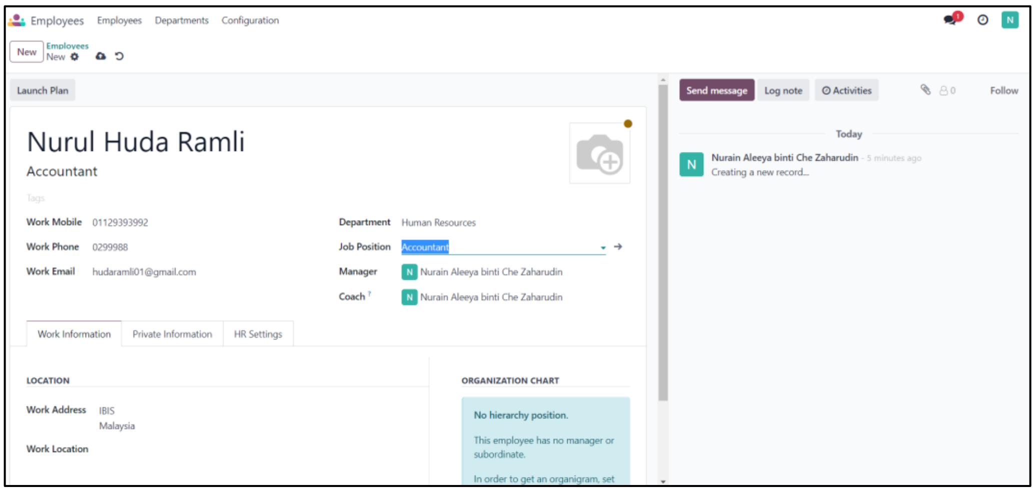
Generated automatically. No signature is required.

Figure 16 : Invoice

Figure 1 shows invoices generated by the Niagawan System. The information that has been entered before is displayed on the invoice including company details.

2.1.1.3 Odoo System

Onnet Consulting is committed to assisting businesses in their growth journey by incorporating technology systems into their daily operations(Tipalti Inc., 2022). Their system lead in supporting local SMEs' digital transformation endeavours, enabling them to enhance operational efficiency through the adoption of affordable business software solutions(Port Cities, n.d.). By having several module such as Accounting Module, Human Resources, Project Management and Reporting Analytics.


Nurul Huda Ramli
Accountant

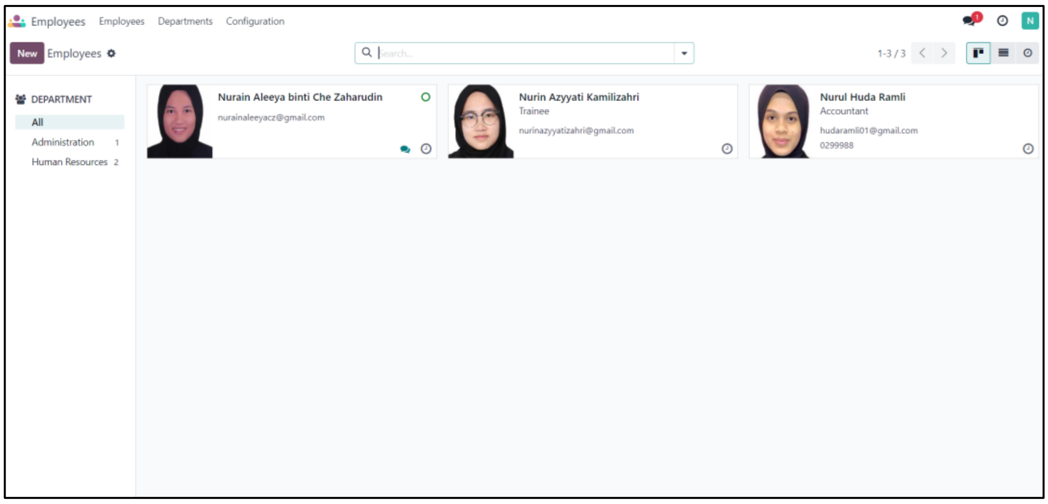
Work Mobile 01129393992
Work Phone 0299988
Work Email hudaramli01@gmail.com
Department Human Resources
Job Position Accountant
Manager Nurain Aleeya binti Che Zaharudin
Coach Nurain Aleeya binti Che Zaharudin

LOCATION
Work Address IBIS
Malaysia
Work Location

ORGANIZATION CHART
No hierarchy position.
This employee has no manager or subordinate.
In order to get an organigram, set

Today
Nurain Aleeya binti Che Zaharudin - 5 minutes ago
Creating a new record...

Figure 17 : Register Employee Page


Nurain Aleeya binti Che Zaharudin
nurainaleeyacz@gmail.com

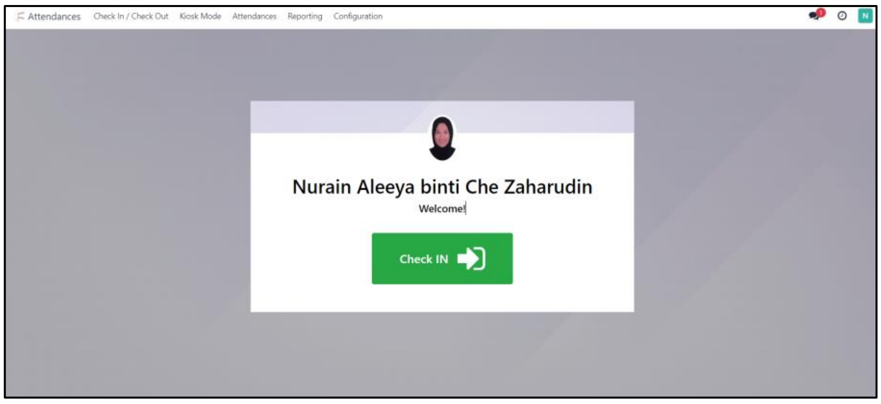
Nurin Azyyati Kamilizahri
Trainee
nurinazyyatizahri@gmail.com

Nurul Huda Ramli
Accountant
hudaramli01@gmail.com
0299988

DEPARTMENT
All
Administration 1
Human Resources 2

Figure 18 : List of User Page

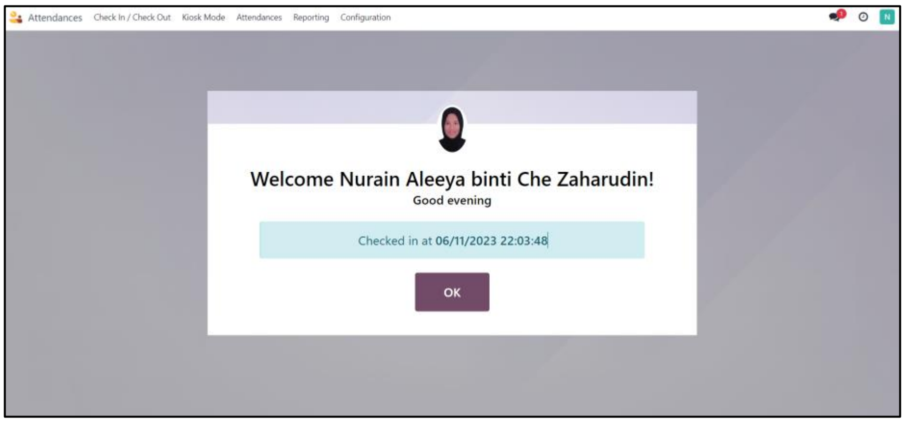
Based on figure 17 and 18, Odoo's user management functionality empowers administrators to create user accounts, define roles, and assign access levels. User groups and access rules ensure users have appropriate data and feature access. Administrators maintain data security and integrity through control over permissions, including read, write, create, and delete actions.



Nurain Aleeya binti Che Zaharudin

Welcome!

Figure 19 : Attendance Page

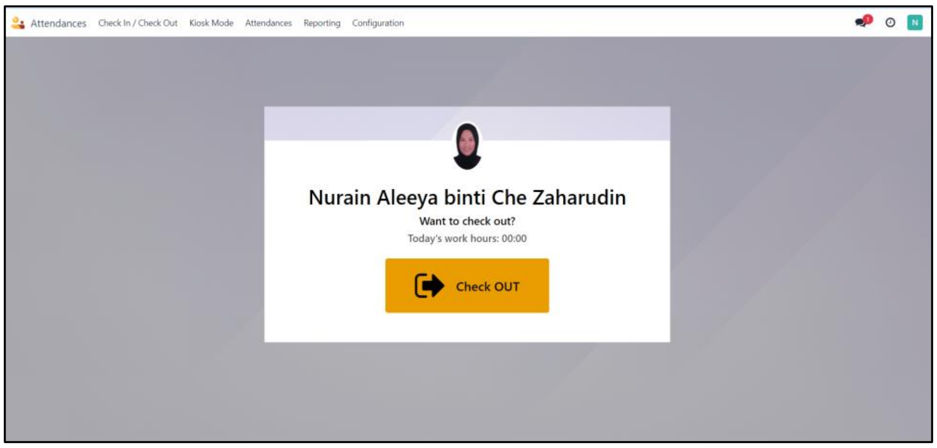


Welcome Nurain Aleeya binti Che Zaharudin!

Good evening

Checked in at 06/11/2023 22:03:48

Figure 20 : Checkin Page



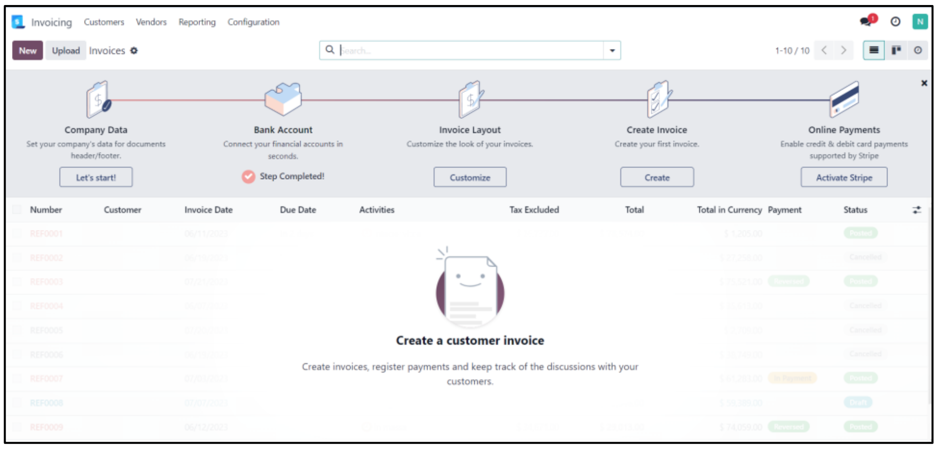
Nurain Aleeya binti Che Zaharudin

Want to check out?

Today's work hours: 00:00

Figure 21 : Check Out System

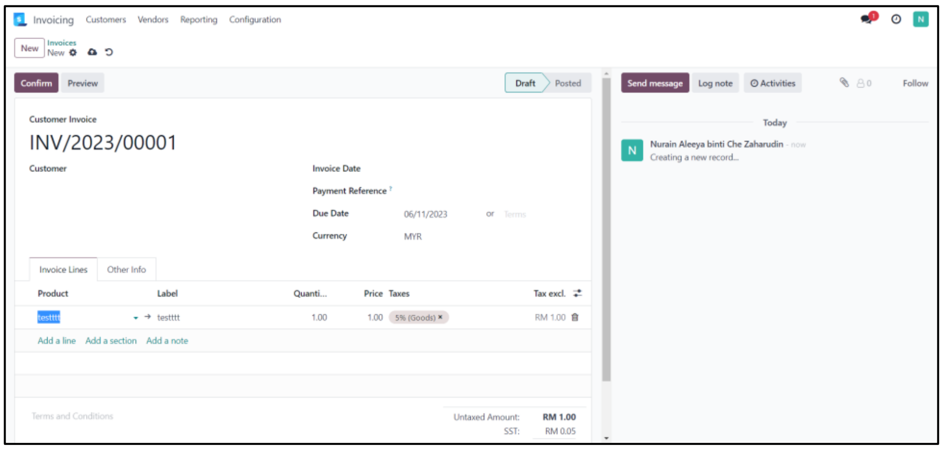
Based on Figures 19, 20, and 21, the attendance module in the Odoo system offers a user-friendly interface. Users can easily check in and out of work by clicking corresponding buttons. Check-in captures the date and time of arrival, while check-out records the departure. The module automatically calculates the duration between check-in and check-out, providing users with the total working period. This duration reflects the actual time spent at work during the session.



Create a customer invoice

Create invoices, register payments and keep track of the discussions with your customers.

Figure 22 : Invoice Page



Customer Invoice
INV/2023/00001

Customer

Invoice Date

Payment Reference

Due Date 06/11/2023 or Terms

Currency MYR

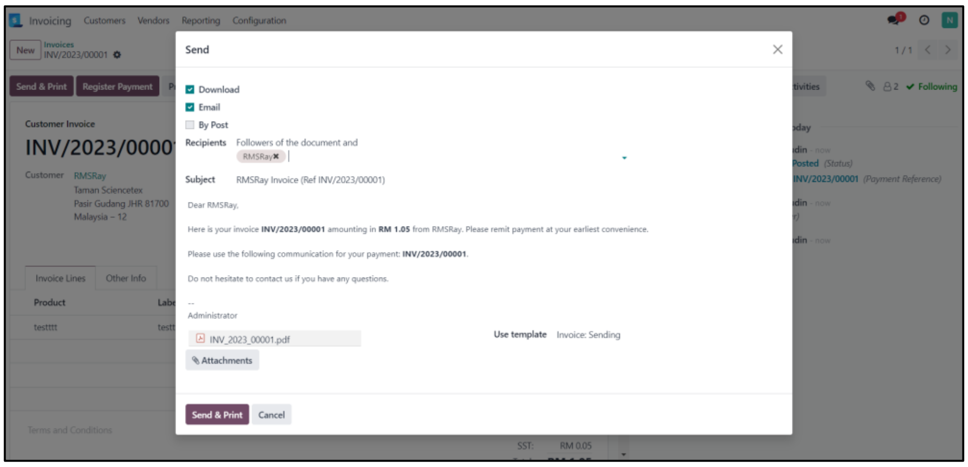
Product	Label	Quanti...	Price	Taxes	Tax excl.
testttt	testttt	1.00	1.00	5% (Goods)	RM 1.00

Untaxed Amount: RM 1.00
SST: RM 0.05

Today

Nurain Aleeya binti Che Zaharudin - now
Creating a new record...

Figure 23 : Add New Invoice



Send

☑ Download
☑ Email
☐ By Post

Recipients Followers of the document and RMSRay

Subject RMSRay Invoice (Ref INV/2023/00001)

Dear RMSRay,

Here is your invoice **INV/2023/00001** amounting in **RM 1.05** from RMSRay. Please remit payment at your earliest convenience.

Please use the following communication for your payment: **INV/2023/00001**.

Do not hesitate to contact us if you have any questions.

--
Administrator

INV_2023_00001.pdf

Use template Invoice: Sending

Customer Invoice
INV/2023/0000
Customer RMSRay
Taman Sciencetex
Pasir Gudang JHR 81700
Malaysia – 12

Figure 24 : Email Template for Invoice

The invoice module in the Odoo system offers users the ability to view a list of invoices, create new invoices, and update existing ones. Figures 22 and 23 depict the interface for adding a new invoice in Odoo. Additionally, Figure 23 displays the email template that users can customize before sending the invoice to clients. With the invoice module, users can conveniently access and manage invoices, ensuring accurate and up-to-date financial records.

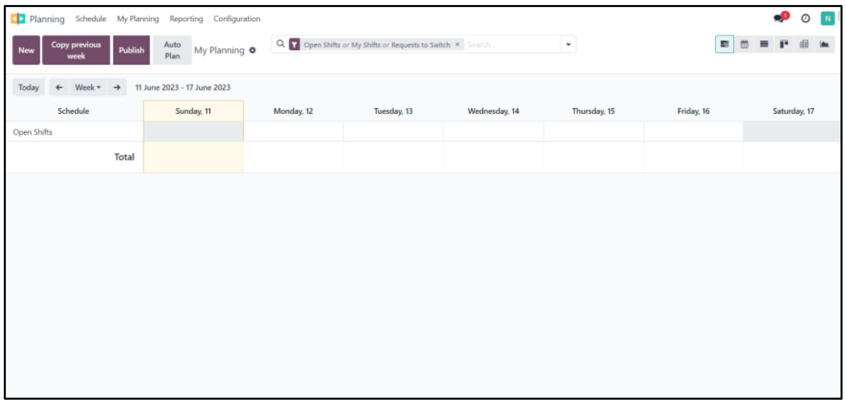


Figure 25 : Planning Interface

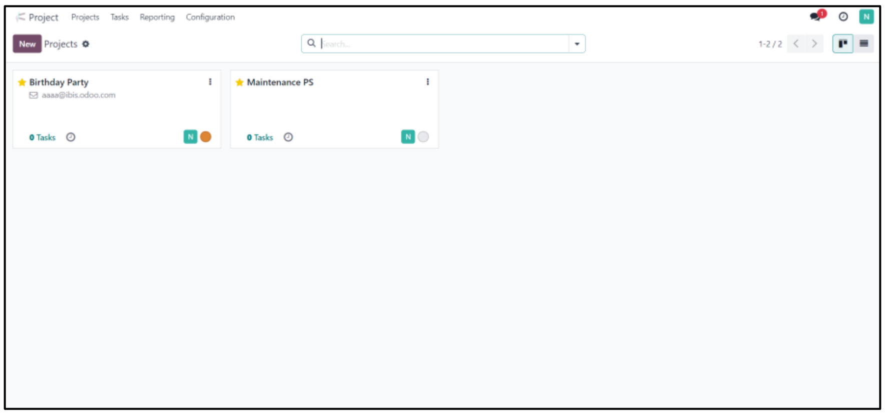


Figure 26: Project List

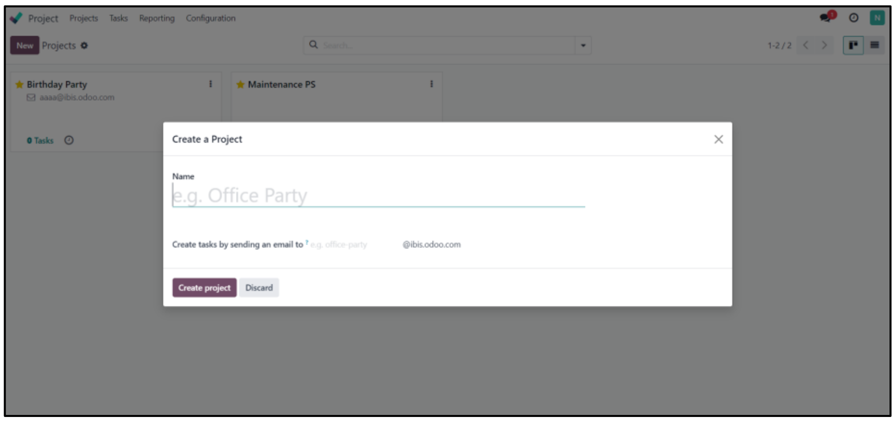


Figure 27: Create Project Interface

Users can leverage a calendar view, as showcased in Figure 25, to add planning events, set deadlines, assign tasks, and schedule project activities. This visual representation aids in effective resource allocation and timeline management. The system also offers a project list view, illustrated in Figure 27, where users can access and manage their created projects. From this interface, users can easily navigate between projects, view project details, and monitor progress. Then initiating a new project, users can utilize the interface in Figure 27 to input the project name and send project invitations via email.

2.2 Comparative Analysis

Table 1 : Summary of comparison two of existing system

Element	Zoho System	Niagawan System	Odoo System
Web application categories	✓ Dynamic web-based system	✓ Dynamic web-based system	✓ Dynamic web-based system
Design	✓ Attractive ✓ User-Friendly ✓ Simple and easy to use	✓ Very Simple ✓ Minimalist ✓ Easy to understand	✓ Responsive Design ✓ Customizability ✓ Modular Design
Metaphor (Language)	✓ Common	✓ Simple	✓ Common
Features	✓ Human Resource function ✓ Leave Tracker ✓ Attendance ✓ Performance ✓ Organization ✓ Travel Claim ✓ Files ✓ Reports ✓ Calendar ✓ Expenses ✓ Invoice and receipt ✓ Customer database	✓ Billing & Invoice ✓ Customer Database ✓ Company Management ✓ Company Letterhead ✓ Real-time data ✓ Real-time report ✓ Statistic ✓ Product & Inventory.	✓ Attendance ✓ Add New User ✓ Update Profile Information ✓ Invoicing ✓ Email Invoice ✓ Template Invoice ✓ Sales and expenses ✓ Create Project ✓ Planning ✓ Project List
Programming Language	✓ HTML ✓ PHP ✓ CSS ✓ Javascript	✓ HTML ✓ PHP ✓ CSS ✓ Javascript	✓ HTML ✓ PHP ✓ CSS ✓ Javascript

Module	✓ Registration ✓ Manage Product ✓ Manage Customer ✓ Manage Organization ✓ Manage Calendar ✓ Manage report & Invoice ✓ Manage Profile	✓ Registration ✓ Manage Product ✓ Manage Customer ✓ Manage Inventory ✓ Manage Report	✓ Registration ✓ Manage Invoice ✓ Manage User ✓ Manage Attendance ✓ Manage Report ✓ To-Do ✓ Manage Inventory ✓ Manage Calendar

2.2.3 Comparison of the existing system

Table 2 : Advantages and Disadvantages of existing system

System	Advantages	Disadvantages
Zoho System	• Users can distinguish between each role in this system. For instance, only certain related roles are able to access some modules. • The Zoho system has a comprehensive dashboard, which makes it very interesting to use. Users can, for instance, view a summary of their activities only on the dashboard, including attendance, pending jobs, approval requests, announcements, and more. • By selecting the Email section on the invoice page, an invoice can be sent to customer’s email using this system.	• When evaluating statistics, the image and graph appear to be improperly organized and may be confusing. • There is no effective inventory and product module. For instance, the user must enter the quantity and item for the invoice.

Niagawan	• The dashboard displays a list of the company's expenses. The company will find it simpler to track cash flow each month. • The font and symbol used are similar to that of another system, so users can use them without consulting the system owner. • By selecting the WhatsApp button on the invoice page, an invoice can be sent to WhatsApp using this system.	• Do not have calendar management where the user cannot select the available date. • It is only used to facilitate sales. It means that employee management cannot be done with it. • Low quality; the free system has many fake buttons inside and a poor-quality system overall.
Odoo System	• Odoo has an intuitive, user-friendly interface that is simple to use. Employees can more easily learn and use the system since it offers a consistent user experience across all modules. Thus, training time is cut down, and user adoption is increased. • Odoo provides robust reporting and analytics capabilities, allowing businesses to gain valuable insights into their operations. Users can generate customizable reports, create dashboards, and visualize data in real-time, helping in decision-making and performance tracking. • Odoo's modular design allows businesses to choose and customize specific modules to fit their needs, resulting in a flexible and personalized system.	• Odoo's user interface may lack consistency across different modules or functionalities. This can result in variations in the user experience, which may require additional training or cause confusion for users transitioning between different areas of the system. • Some users may find that the overall design aesthetic of Odoo's interface lacks modern or visually appealing elements. The system's design may appear dated compared to other contemporary software solutions, which could affect user satisfaction and engagement.

2.2.4 Comparison between the existing system and the proposed system.

Table 3 : Comparison System

Features	IBIS Management System	Zoho System	Niagawan System	Odoo System
Registration and Login	✓	✓	✓	✓
Manage Product and Inventory	X	X	✓	X
Manage Invoice and receipt	✓	✓	✓	✓
Manage Calendar	X	✓	X	✓
Manage profile	✓	✓	X	✓
Manage Customer	✓	✓	✓	✓
Claim Overtime and Allowance	✓	✓	X	X
Attendance and Leave	✓	✓	X	✓

2.3 Proposed Project

The improvement or changes to the project, program and application development, coding, unit, integration and system testing are all split down in the development phase.

2.4 Summary

This chapter explores three existing systems: Zoho, Niagawan, and Odoo, highlighting their respective advantages, drawbacks, and similarities to the proposed IBIS Management System. By reviewing relevant literature, this chapter aims to provide guidance and inspiration to developers working on the development of the IBIS Management System

CHAPTER 3
METHODOLOGY

3.1 Introduction

The development process for this IBIS system's methodology will be covered in this chapter. The methodology used in this development process is crucial to ensuring client satisfaction. There are numerous methodologies for the software development life cycle, but Rapid Application Development was selected for the IBIS system because it has an appropriate phase for system changes. Project requirements like functional requirements, non-functional requirements, constraints, and limitations are also explained in this chapter(*Rapid Application Development (RAD): Full Guide | Creatio,* n.d.).

The software company will gather all of the information required to develop the system during this phase. A client interview is conducted as part of the information collection process to learn the requirements they want to use in the system. It is recommended to create a simple design to satisfy the client's request so that there won't be any changes made during the development process. It is immutable because the Software Requirement Specification is required to contain all requirements in one document (SRS).

The development team has many responsibilities during this phase because the process of creating the system will start. The SRS for this project, as stated in phase I, must be adhered to by the system that will be developed. The client and the developer have made all of the agreements for the requirements in this system. The system analyst plays a key role in this phase as well by inspecting the system that was created using SRS. To satisfy the client's request, the system must be shown to the system analyst and the client after the developer is ready to create it(Matthew Martin, 2023). If the system developed does not meet the client's recorded requirements as listed on the SRS, the client might also comment. According to the client's comments, the developer needs to fix any bugs as soon as possible. This phase will be repeated until the system is finished and meets all the specifications.

Usually, the system is prepared for testing once the entire development process has been completed. User Acceptance Test is a form that is included with the system that will be tested. The client will receive instructions on how to use the system during this phase. The system is typically tested by a designated software tester prior to the training. Only minor changes are permitted during this phase. Only when the project manager and developer are in agreement can significant changes be made. This is so that if there are significant changes, the cost will go up since the agreement was made in phase I. After the client is satisfied with the testing performed on this system, the company will typically send this system and the UAT document to a third party to test the functionality of the system, including entering data into the database and determining whether or not all of the button's function. When there are no bugs left, the testing phase will be over(Lucid Content Team, 2023).

The built product will be made live during this phase, which is the implementation phase. When all the procedures are completed, the system will be made available online so that the client can use it independently without assistance from the developer. Potential users will only receive user manuals from the developer as instructions on how to use the IBIS Management system.

3.2 Methodology

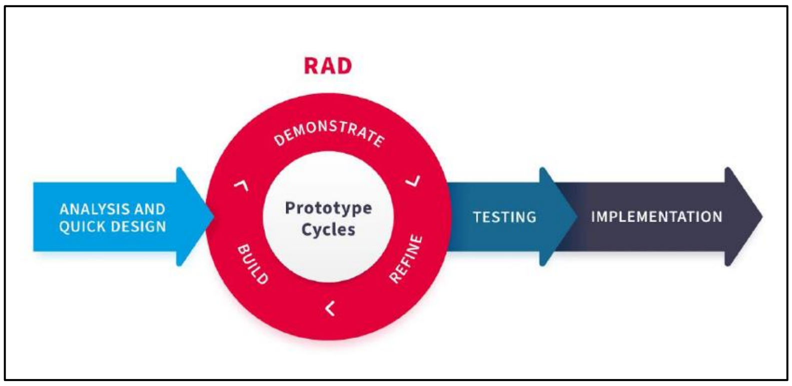


Figure 28 : Rapid Application Development

Phase I: Analysis and Quick Design

During the analysis and quick design phase of Phase 1, a meeting was conducted with IBI Services Sdn Bhd, the client, to gather their requirements. The meeting involved discussions about the challenges faced by both the management team and employees. The client expressed the need for a system that could assist in various task management aspects, such as employee registration, claim management, invoice management, maintenance, and reporting. To visualize the system's flow, prototypes and designs were sketched out. The requirements and functional specifications were documented and agreed upon by both the client and developers, ensuring a clear understanding of the project scope and objectives.

Phase II: Prototype Cycles

During Phase II, the prototype cycles of the IBIS-ms development, the developers commence the implementation based on the requirements discussed in the first phase. To ensure the satisfaction of the IBI Services team, developers conduct a demonstration and record all the necessary information for approval. The client provides feedback, and the developers brainstorm ideas to enhance the system's visual appeal and user-friendliness. The Manage Registration module is completed relatively quickly, taking only 1-2 weeks due to its general and familiar interface. However, the invoice and claim modules encounter some issues that require testing by the client to ensure the correctness of the flow and processes. The system's design adheres to the requirements specified in the document from Phase 1. After discussions with the IBIS team, the developers proceed to re-code and resolve any identified bugs, aiming to achieve a smooth system. The IBIS-ms is developed using the PHP Laravel framework, along with HTML, CSS, and JavaScript as the primary languages utilized in the development process.

Phase III: Testing

During the testing phase, the developers first conduct internal testing to ensure the system runs smoothly without any bugs. After this self-testing, a User Acceptance Test (UAT) is conducted to update the client on the system's progress. Our team facilitates a UAT session with the client, providing training on how to effectively utilize the system. Additionally, a third-party tester is invited to evaluate the system's performance especially for claim and invoice modules. The testing process continues until the system is deemed ready for use, ensuring that all functionalities, including buttons, databases, and forms, are working properly and can be easily utilized by the end-users.

Phase IV: Implementing

After receiving approval and satisfaction from the IBIS team, the system is deployed to the Indah host server, indicating its readiness for use by end-users. In this phase, the IBIS team takes ownership of the system and begins utilizing it for their daily operations. During the initial stages of system usage, the IBIS team can seek assistance and support if needed. The developers are available to provide guidance and address any queries or concerns that may arise during the early adoption of the system.

3.3 Project Requirement

3.3.1 System Requirement

The two types of system requirements that are used in the development of IBIS Management Systems will be covered in this section: software requirements and hardware requirements.

Table 4 : Software Requirement

Software	Description
Visual Studio Code	To write a source code for the web application
Xampp	Xampp is a local host to connect the devices and the website before deploying on the server.
Figma	Figma is used for prototype purposes. The demonstration of the system will be in Figma including testing the button and design of a system.
Draw.io	Draw.io is a platform to draw the use case for this system. It also uses for dialogue diagrams and etc.
Microsoft Office	Microsoft word will be used for documentation purposes and Microsoft Excel will be used for client feedback during the testing phase.

Table 5 : Hardware Requirement

Hardware	Description
Laptop	Asus A45U: To write source code and document for this project.
Monitor	HP 22inch: To display design and Web-based system.
Mouse	HP Z500u: To point and move the object during the design phase.

3.3.2 Functional requirement

- **Manage Registration and Profile**
    - The system allows all the users to make an account and login into the system.
    - The system allows the user to edit their profile such as their personal information and the password of the system.

- **Manage User**
    - The system allows the Human Resources to register new account for employee.
    - The system allows the Human Resources to view the user for the IBIS-ms.

- **Manage Attendance**
    - The system allows the user to submit their attendance by click check in button.
    - The system allows the user to submit check out time by click the checkout button.
    - The system allows the user to view their attendance report.

- **Manage Maintenance & Report**
    - The system allows Supervisor to assign the work location for general workers.
    - The system allows the user to view the Job List.
    - The system allows the user to submit a site visit report to get approval from the supervisor.

- **Manage Invoice and Company**
    - The system allows the accountant to make an invoice for the customer.
    - The system provides the capability for users to add companies, facilitating easier selection of the company for invoice purposes by accountants.

3.3.3 Non- Functional Requirement

- **Security**

  IBIS Services Sdn Bhd is in charge of all the data in this system. In order to protect the information and records of employees and the company, this system has high security. Viruses and the leakage of sensitive information are difficult with a high-security system. Users who are not registered cannot use this system and the system will detect the location when the user is signed in.
- **Performance**

  This system will load pages quickly to maintain user satisfaction. This system can be accessed by users with reliable internet in an estimated 3 seconds per page. When there are many users online at once, the hosting used might require a sizable quota to prevent system outages.
- **Usability**

  The IBIS Management System should feature a well-crafted user interface that combines visual appeal, organization, and intuitiveness. It should effectively present information in a clear and comprehensible manner, employing suitable navigation, menus, and controls. Ease of learning is crucial, with an aim to minimize the learning curve for new users. Accomplishing this requires the provision of clear instructions, tooltips, contextual help, and a logical task flow. Error mitigation is another important aspect, with the system designed to minimize errors through the implementation of appropriate feedback, error prevention mechanisms, and unambiguous error messages. In the event of errors, the system should handle them gracefully, enabling users to recover seamlessly without encountering data loss or confusion.

3.3.4 Constraint and Limitation

- IBIS Management System can be used with internet access only.
- A total of 200 users can simultaneously use the IBIS Management System.
- IBIS Management System is only available for web-based systems.

3.4 Propose Design

3.4.1 Context Diagram

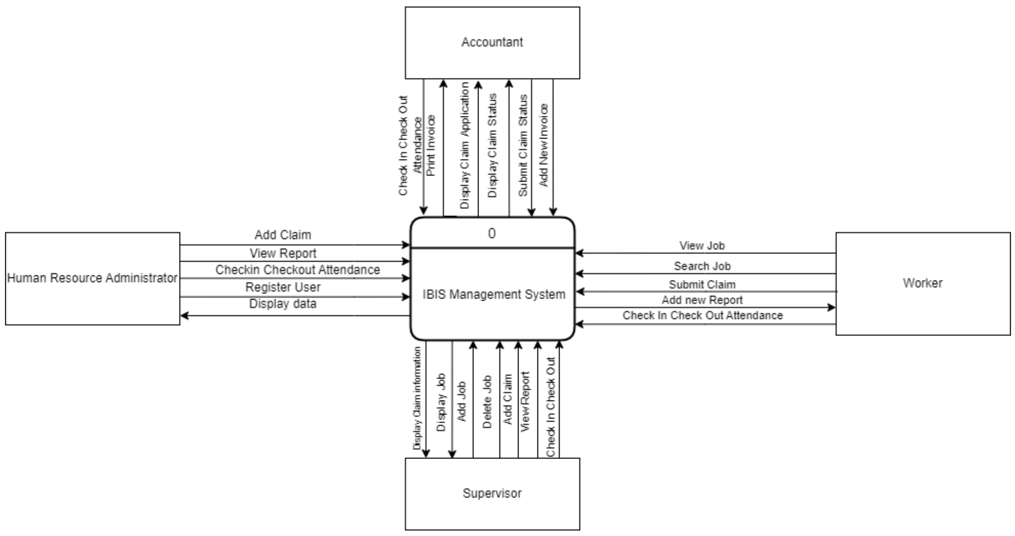


Figure 29 : Context Diagram of IBIS Management System

3.4.2 Use Case Diagram

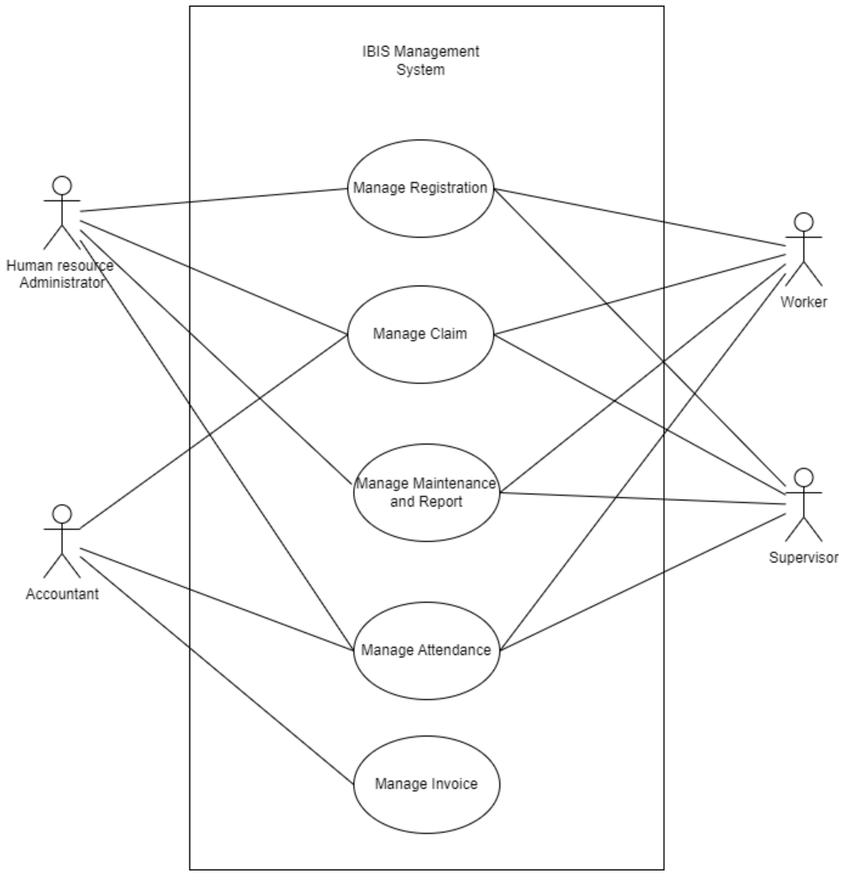


Figure 30 : Use case for IBI Management System

3.4.3 Use Case Description

3.4.1.1 Manage Registration

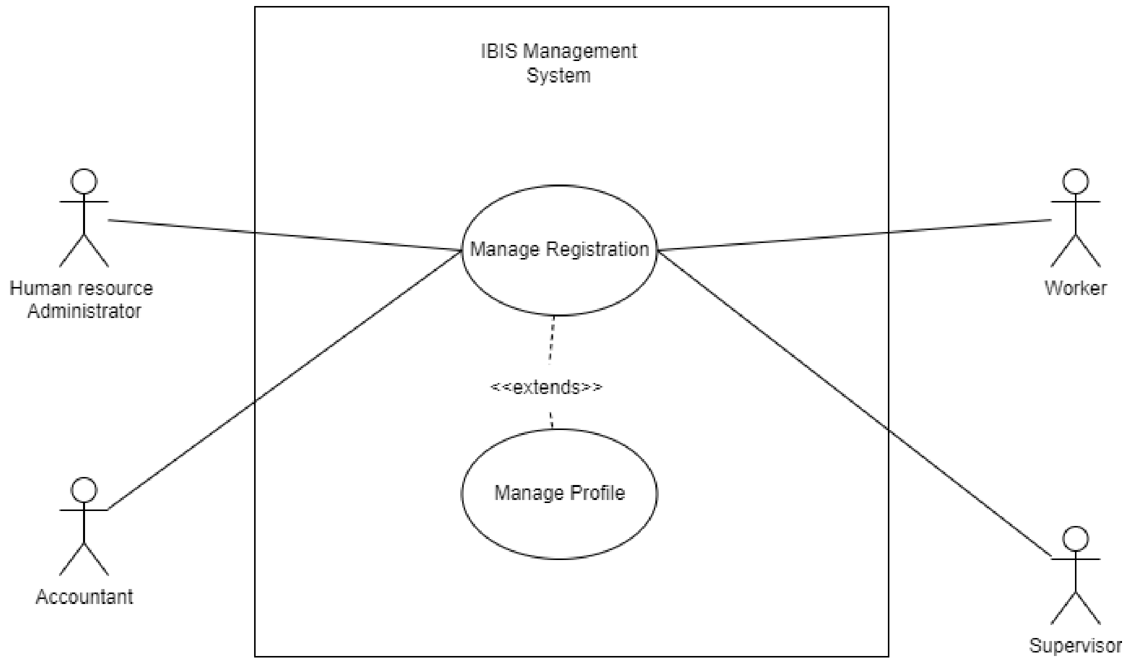


Figure 31 : Usecase of Manage Registration

Table 6 : : Use case of Manage Registration

Use Case Name	IBI-UC-100
Brief Description	This use case allows user to register and login into the system by using email and password. User also can manage their profile in this system.
Actor	1. Human resources 2. Accountant 3. Workers 4. Supervisor
Pre-Condition	User must not have an account for this system

Basic Flow	[Register] [Human Resource Administrator] 1. The use case begins when the user login into the system. 2. The system shows Manage User page. 3. User need to click <<Add new user>> to register a new user. 4. The system shows registration form. 5. User need to fill in all the required information. 6. User clicks <<Register>> button 7. The data will be stored in database. 8. Successful message displayed. 9. The use case end. [Manage profile] [All actor] 1. The use case begins when the user login into the system. 2. The system shows the dashboard. 3. User clicks <<My Profile>> button on the top navigation. 4. User can edit their profile. 5. User need to click <<submit>> button. 6. The data will be changed in database. 7. Successful message displayed 8. The use case end.
Alternative Flow	None
Exception Flow	E1 – Username and Email already registered. 1. System display error message. 2. User need to fill in another information 3. The use case continues with step 3 basic flow [Register]
Post Condition	Users can access and login into the system. The profile also can be updated.
Constraint	Not applicable

3.4.1.2 Manage Financial Resources

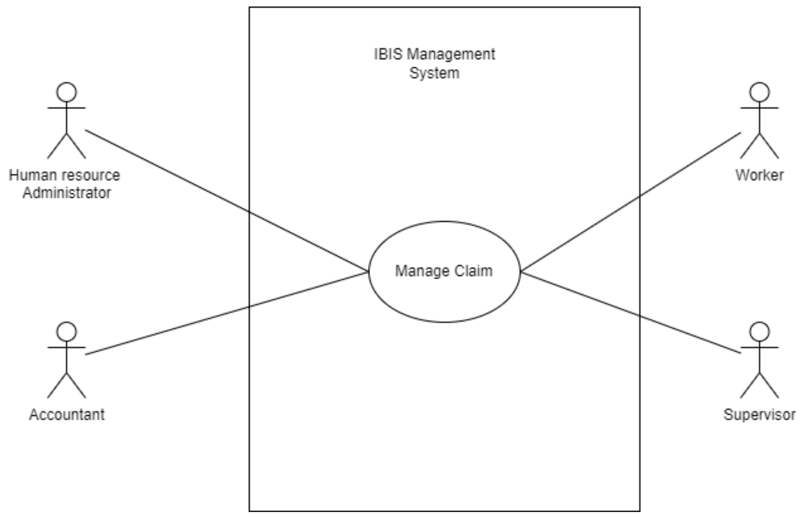


Table 7 : : Use case of Manage Financial resource

Use Case Name	IBI-UC-200
Brief Description	This use case allows user to manage claim including Apply Claim, Review Claim, Approve Claim.
Actor	1. Human resources 2. Accountant 3. Workers 4. Supervisor
Pre-Condition	User must login into the IBIS Management System.
Basic Flow	1. The use case begins when the user login into the system. 2. The system shows Dashboard page. 3. User need to click <<Claim>> in the side navigation bar. 4. The system shows Claim page. 5. The graph and chart that related to company financial displayed. 6. User need to choose the option below: i) [A1: View] ii) [A2: Add New Claim] 7. The use case end.

Alternative Flow	[A1: View] 1. The alternative flow begins when the accountant clicks <<View>> button. 2. The system shows claim information that need to be reviewed by the Supervisor. 3. User select Reviewed/Successful for status. 4. The system displays the popup message. 5. The status changed to 'Approve' in the status section. 6. The use case end. [A2: Add New claim] 1. The alternative flow begins when the accountant clicks <<Add New Button>> 2. The system shows claim form that need to be filled in by the user. 3. User clicks <<Submit>> button. 4. The system displays the popup message. 5. The use case end.
Exception Flow	None
Post Condition	Approval Claim has been done. User can print the document.
Constraint	Not applicable

3.4.1.3 Manage Maintenance and services

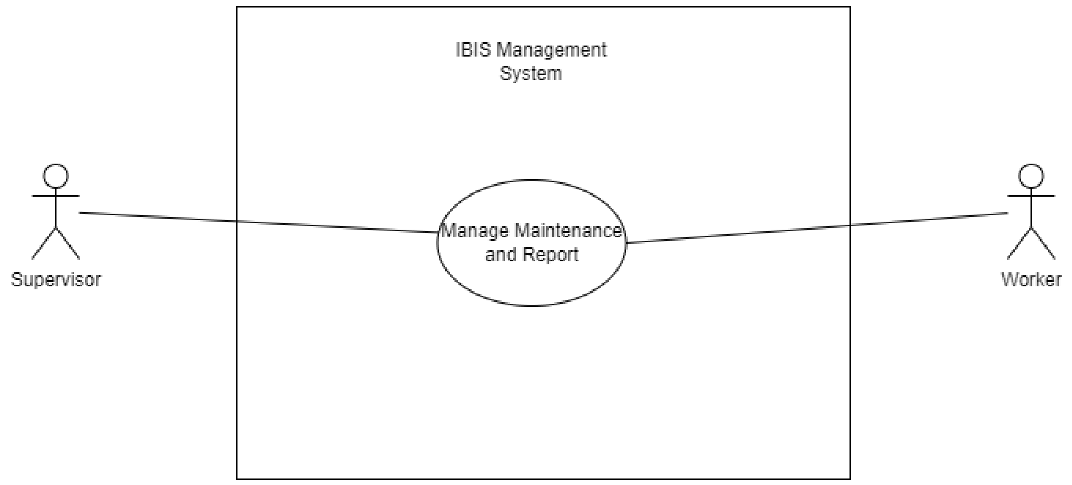


Table 8 : : Use case of Maintenance and Report

Use Case Name	IBI-UC-300
Brief Description	This use case allows user to manage maintenance and report in the IBIS management system. It allows the user to add Job, view Job and update Job. User also can Add New Report.
Actor	1. Workers 2. Supervisor
Pre-Condition	User must login into the IBIS Management System.
Basic Flow	1. The use case begins when the user login into the system. 2. The system shows Dashboard page. 3. User need to click <<Maintenance and Report>> in the side navigation bar. 4. The system shows Maintenance and Report page. 5. User have three option to choose: i) [A1: Add New Job] ii) [A2: Delete Job] ii) [A3: Add New Report] 6. The use case end.

Alternative Flow	[A1: Add New Job] 1. The alternative flow begins when the Supervisor click <<Add New Job>> button. 2. The system shows Task information form page. 3. User fill in all the required information 4. User click <<Submit>> button. 5. Task saved in database. 6. The system displays the popup message. 7. User clicks <<Ok>> button 8. The use case end. [A2: Delete] 1. The alternative flow begins when the user click <<Delete>> button. 2. The system shows confirmation message. 3. User clicks <<Yes>> button 4. The use case end. [A1: Add New Report] 1. The alternative flow begins when the Supervisor click <<Add New Report>> button. 2. The system shows Task information form page. 3. User fill in all the required information 4. User click <<Submit>> button. 5. Task saved in database. 6. The system displays the popup message. 7. User clicks <<Ok>> button 8. The use case end
Exception Flow	None
Post Condition	Status for available task will be changed to pending task. The worker's name will be displayed on the selected task.
Constraint	Not applicable

3.4.1.4 Manage Attendance

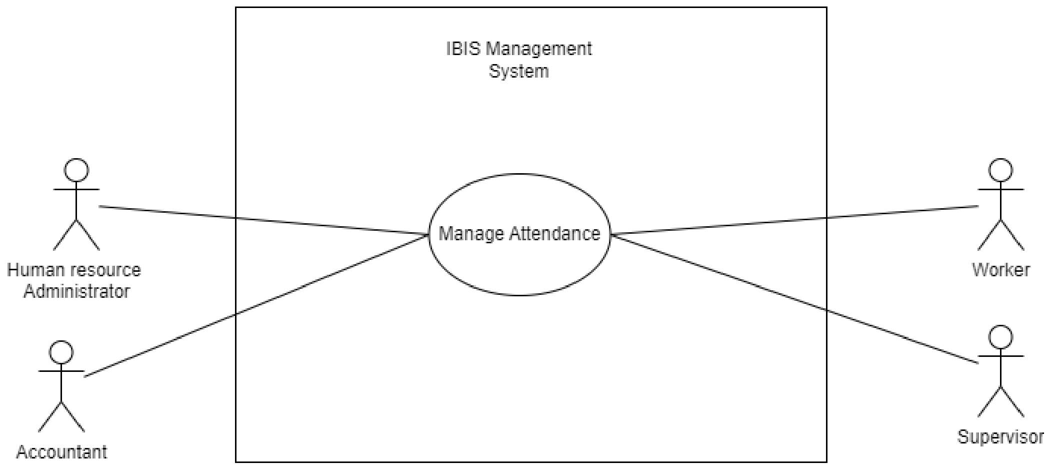


Table 9 : Use case of Manage Attendance

Use Case Name	IBI-UC-400
Brief Description	This use case allows user to manage their attendance.
Actor	1. Human resources 2. Accountant 3. Workers 4. Supervisor
Pre-Condition	User must login into the IBIS Management System.
Basic Flow	1. The use case begins when the user needs to click <<Attendance>> in the side navigation bar. 2. The system shows attendance page. 3. The system requests the data from the database. 4. User clicks <<Check-In>> button. 5. User can view the attendance report 6. The use case end.
Alternative Flow	None
Exception Flow	None
Post Condition	1. Activity report has generated by the system.
Constraint	Not applicable

3.4.1.3 Manage Invoice

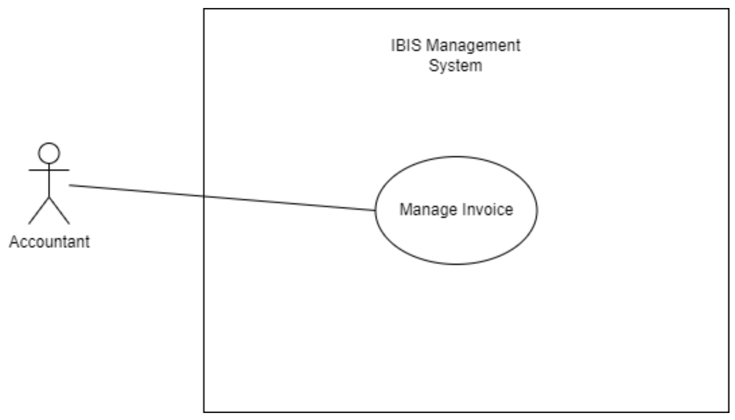


Table 10 : : Use case of Invoice

Use Case Name	IBI-UC-500
Brief Description	This use case allows user to manage invoice. The user can create, view, and delete the invoice.
Actor	1. Accountant
Pre-Condition	User must login into the IBIS Management System.
Basic Flow	1. The use case begins when the user login into the system. 2. The system shows Dashboard page. 3. User need to click <<Invoice>> in the side navigation bar. 4. The system shows Invoice page. 5. User have one option to choose: i. [A1: Add New Invoice] 6. The use case end.
Alternative Flow	[A1: Add New Invoice] 1. The alternative flow begins when the accountant clicks <<Add New Invoice>> button. 2. The system shows Task information form page. 3. User fill in all the required information 4. User click <<Submit>> button. 5. Task saved in database. 6. The system displays the popup message. 7. User clicks <<Print in PDF>> button 8. The user need to click <<Print>>

	9. The use case end.
Exception Flow	None
Post Condition	New Invoice Added
Constraint	Not applicable

3.4.4 Flowchart

i) Flowchart for Human Resources Administration

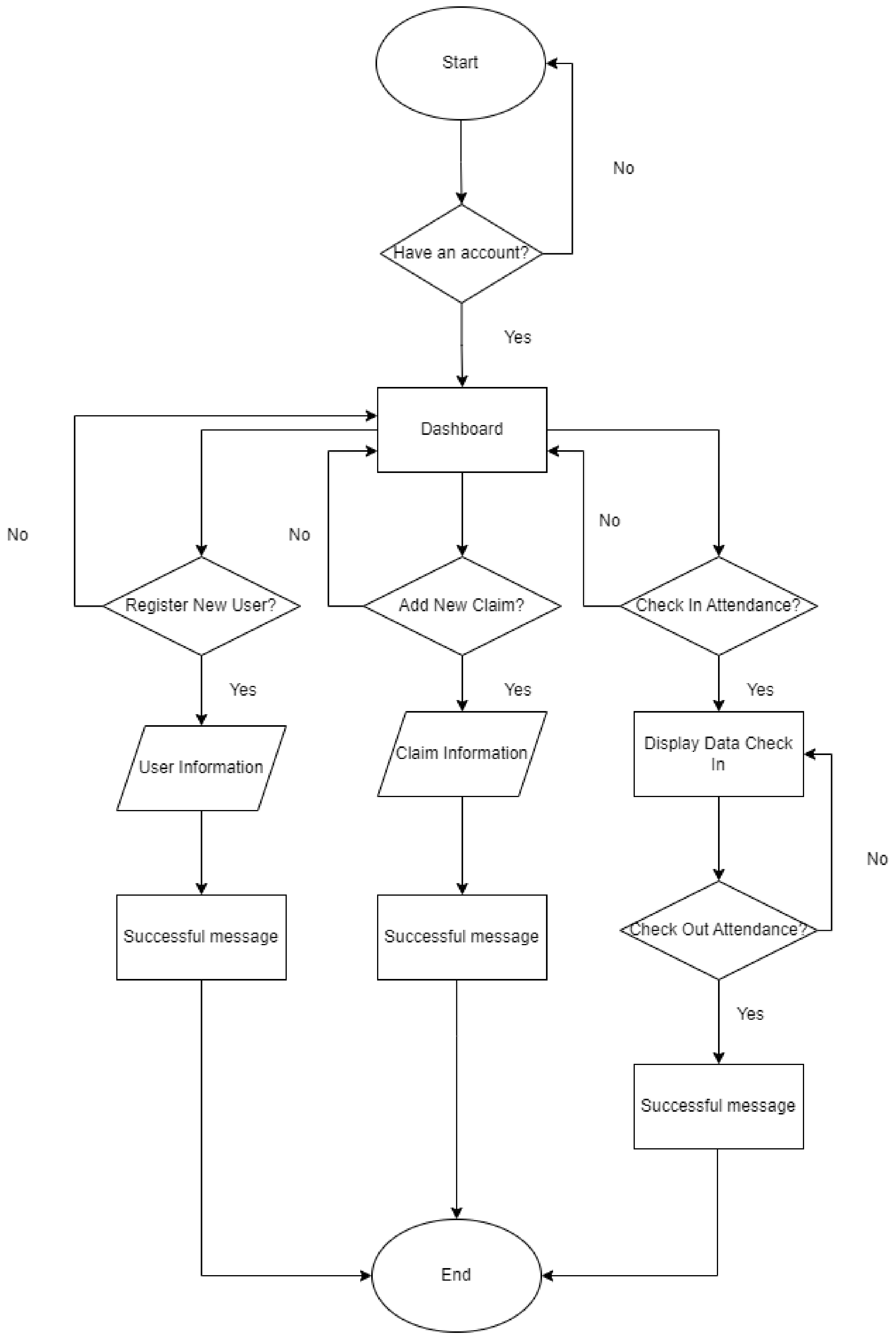


Figure 32 : Flowchart for Administration

ii) Flowchart for Accountant

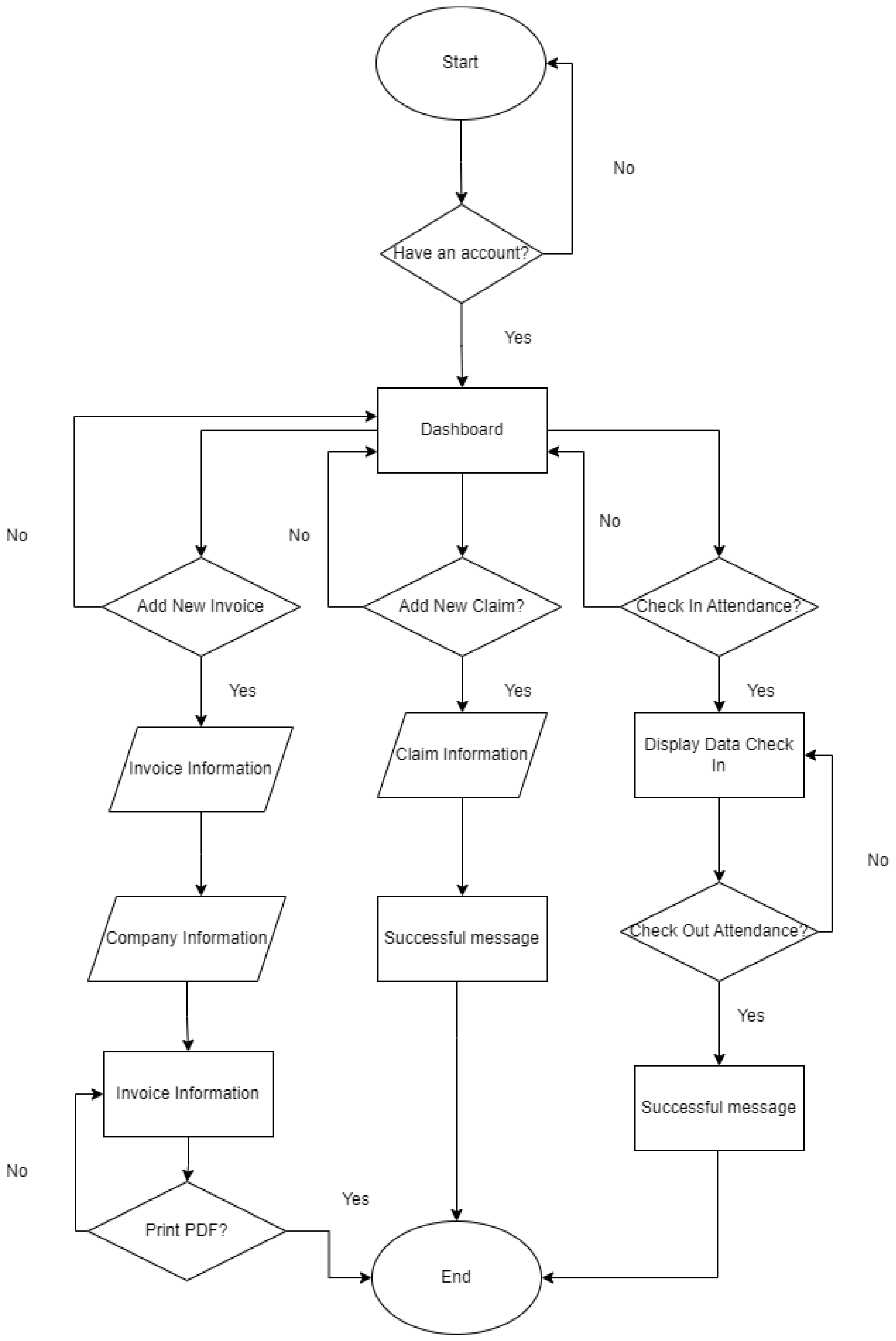


Figure 33 : Flowchart for Accountant

iii) Flowchart for Worker and Supervisor

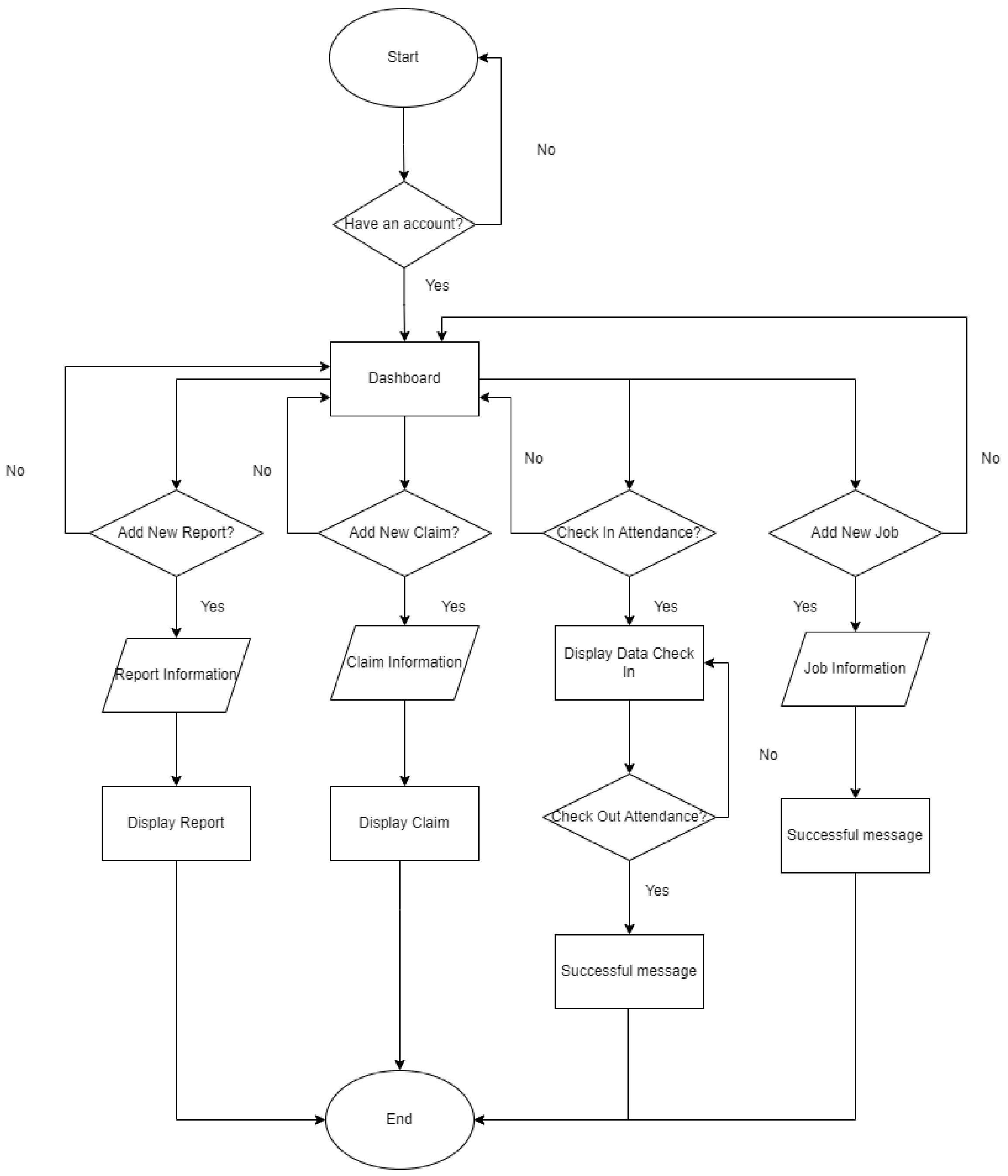


Figure 34: Flow Chart for Supervisor and Worker

3.5 Data Design

3.5.1 Entity Relation Diagram

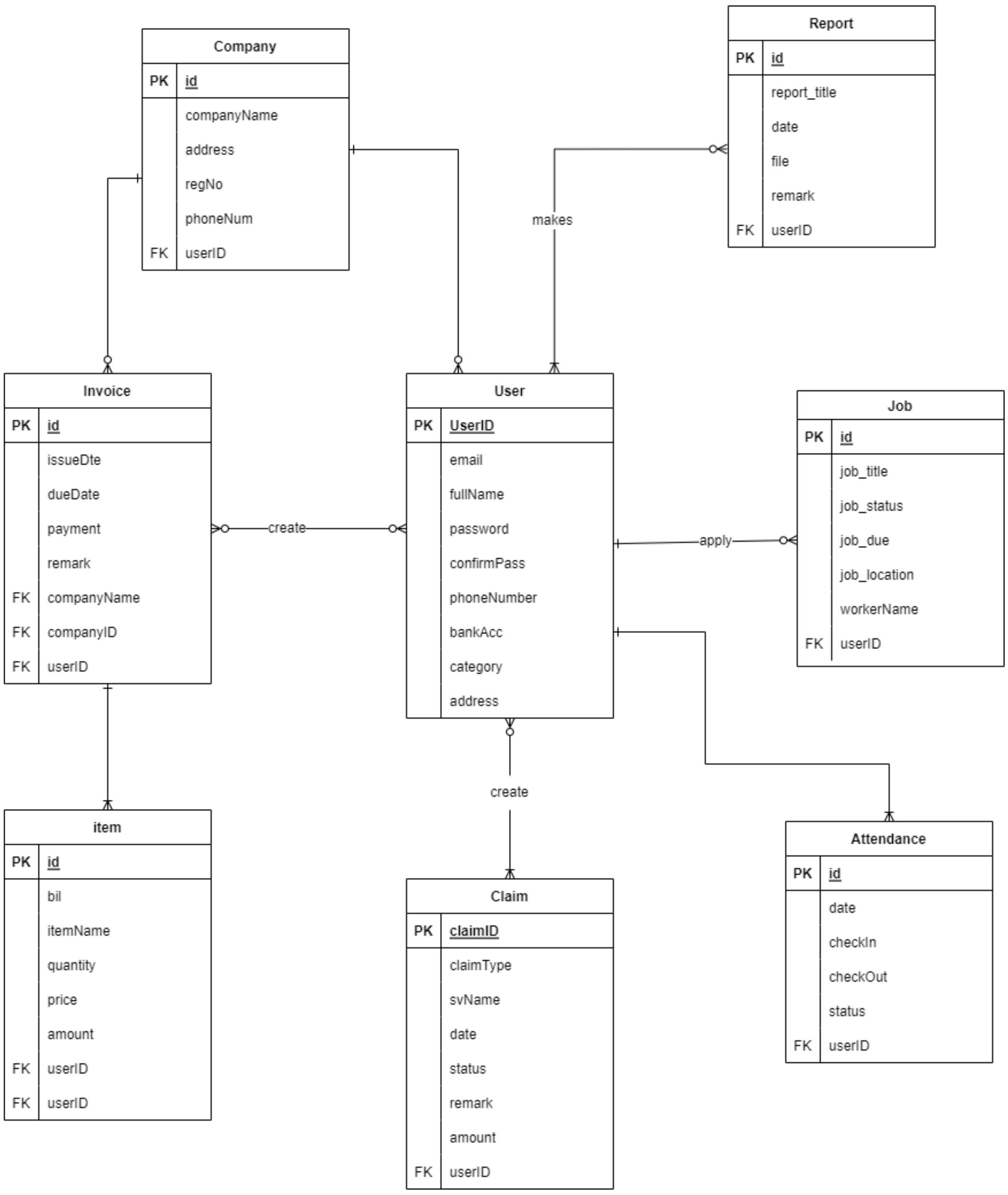


Figure 35 : Entity Relationship Diagram of IBIS System

3.5.2 Data Dictionary

3.5.2.1 User

Table 11 : Data Dictionary of User

Field Name	Description	Data Type	Constraint
id	User ID	INT	PK
email	User email	VARCHAR (255)	
fullname	User Fullname	VARCHAR (255)	
password	User Password	VARCHAR (255)	
confirmPass	User Confirm Password	VARCHAR (255)	
phoneNumber	User Phone Number	VARCHAR (255)	
bankAcc	User bankAcc	VARCHAR (255)	
category	User Category	VARCHAR (255)	
address	User password	VARCHAR (255)	

3.5.2.2 Report

Table 12 : Data Dictionary of Report

Field Name	Description	Data Type	Constraint
id	Report ID	INT	PK
UserID	User ID	INT	FK
report_title	Report title	VARCHAR (255)	
file	Report file	VARCHAR (255)	
remark	Report Information	VARCHAR (255)	
date	Date of issued	DATE	

3.5.2.3 Invoice

Table 13 : Data Dictionary of Invoice

Field Name	Description	Data Type	Constraint
id	Invoice ID	INT	PK
UserID	User ID	INT	FK
companyID	Company ID	INT	FK
companyName	Company Name	VARCHAR (255)	FK
issueDate	Date of issued	DATE	
dueDate	Due Date	DATE	
payment	Type of Payment for the Invoice	VARCHAR (255)	
remark	Invoice Description	VARCHAR (255)	

3.5.2.4 Claim

Table 14 : Data Dictionary of Claim

Field Name	Description	Data Type	Constraint
claimID	Claim ID	INT	PK
UserID	User ID	INT	FK
claimType	Claim Type	VARCHAR (255)	
Status	Status Claim	VARCHAR (255)	
remark	remark	VARCHAR (255)	
amount	Total of Claim	DOUBLE	

3.5.2.5 Attendance

Table 15 : Data Dictionary of Attendance

Field Name	Description	Data Type	Constraint
id	Attendance ID	INT	PK
UserID	User ID	INT	FK
checkIn	Check in time	TIMESTAMP	
checkOut	Check out time	TIMESTAMP	
status	User Availability	VARCHAR (255)	

3.5.2.6 Job

Table 16 : Data Dictionary of Job

Field Name	Description	Data Type	Constraint
id	Job ID	INT	PK
UserID	User ID	INT	FK
jobTitle	Job Title	VARCHAR (255)	
jobStatus	Job Status	VARCHAR (255)	
jobDue	Due Date	DATE	
workerName	Worker Name Assigned	VARCHAR (255)	

3.5.2.7 Company

Table 17 : Data Dictionary of Company

Field Name	Description	Data Type	Constraint
id	Company ID	INT	PK
UserID	User ID	INT	FK
companyName	Company Name	VARCHAR (255)	
address	Company Address	VARCHAR (255)	
regNo	Company Registration No	VARCHAR (255)	
phoneNum	Company No	VARCHAR (255)	

3.5.2.8 Item

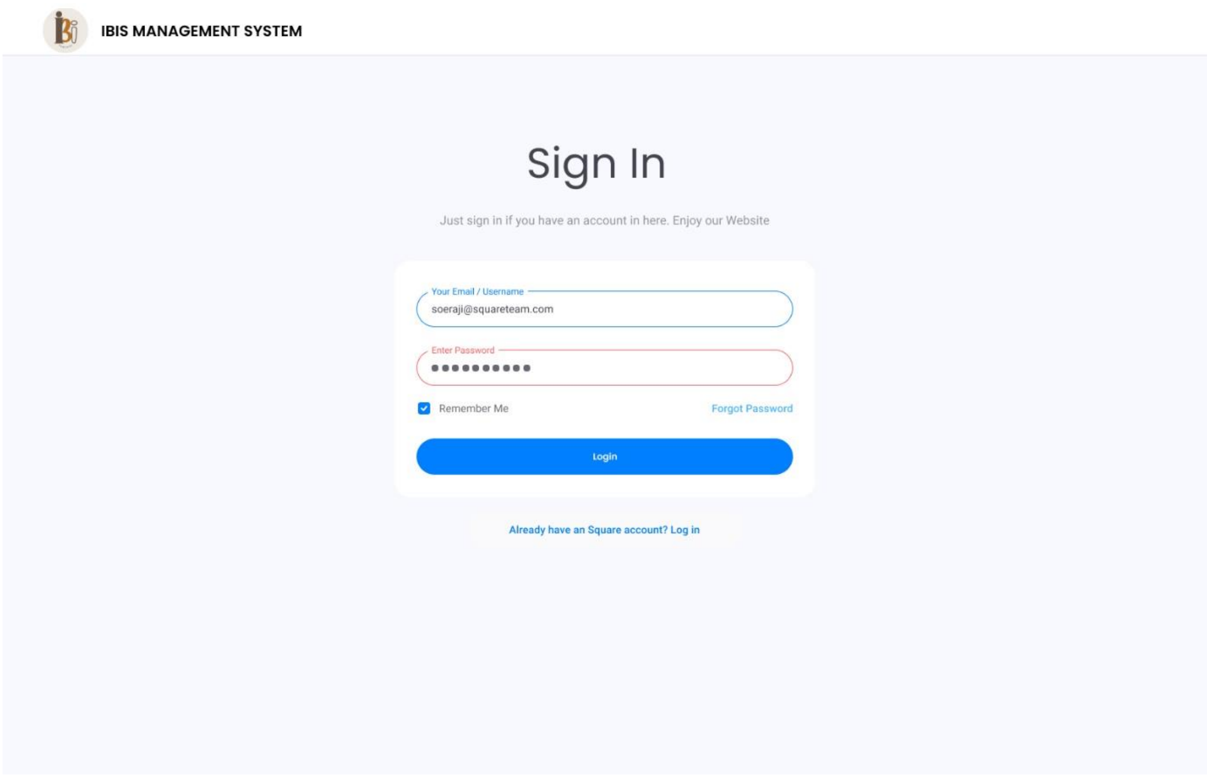
Table 18 : Data Dictionary of Item

Field Name	**Description**	**Data Type**	**Constraint**
id	Item ID	INT	PK
UserID	User ID	INT	FK
bil	Number of Item	VARCHAR (255)	
itemName	Item Name	VARCHAR (255)	
quantity	Item Quantity	VARCHAR (255)	
price	Item Price	DOUBLE	
amount	Item Amount	DOUBLE	

3.6 Proof of Initial Concept

3.6.2 Prototype

3.6.2.1 Sign In Page


IBIS MANAGEMENT SYSTEM

Sign In

Just sign in if you have an account in here. Enjoy our Website

Your Email / Username: soeraji@squareteam.com

Enter Password: ●●●●●●●●●●

Remember Me

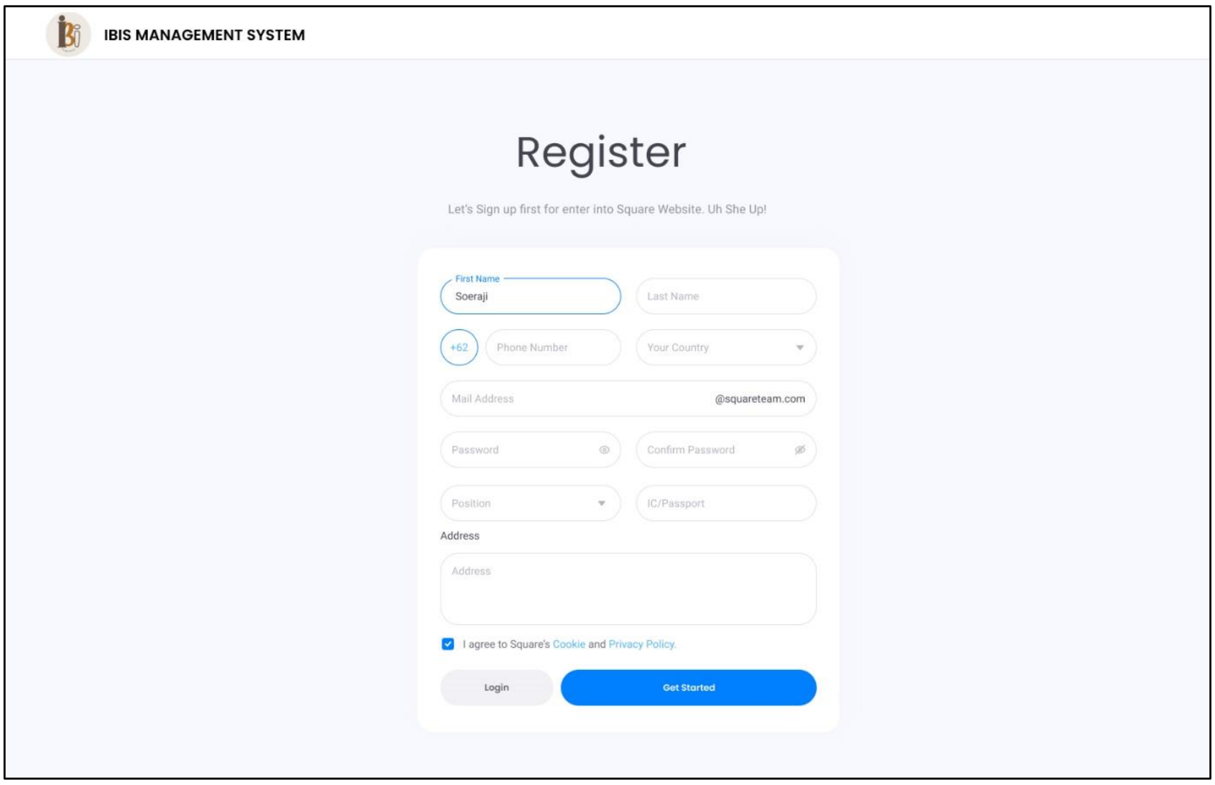
Forgot Password

Login

Already have an Square account? Log in

Figure 36 : Interface of Sign Page

3.6.2.2 Register Page



IBIS MANAGEMENT SYSTEM

Register

Let's Sign up first for enter into Square Website. Uh She Up!

First Name: Soeraji

Last Name

+62 | Phone Number | Your Country

Mail Address @squareteam.com

Password | Confirm Password

Position | IC/Passport

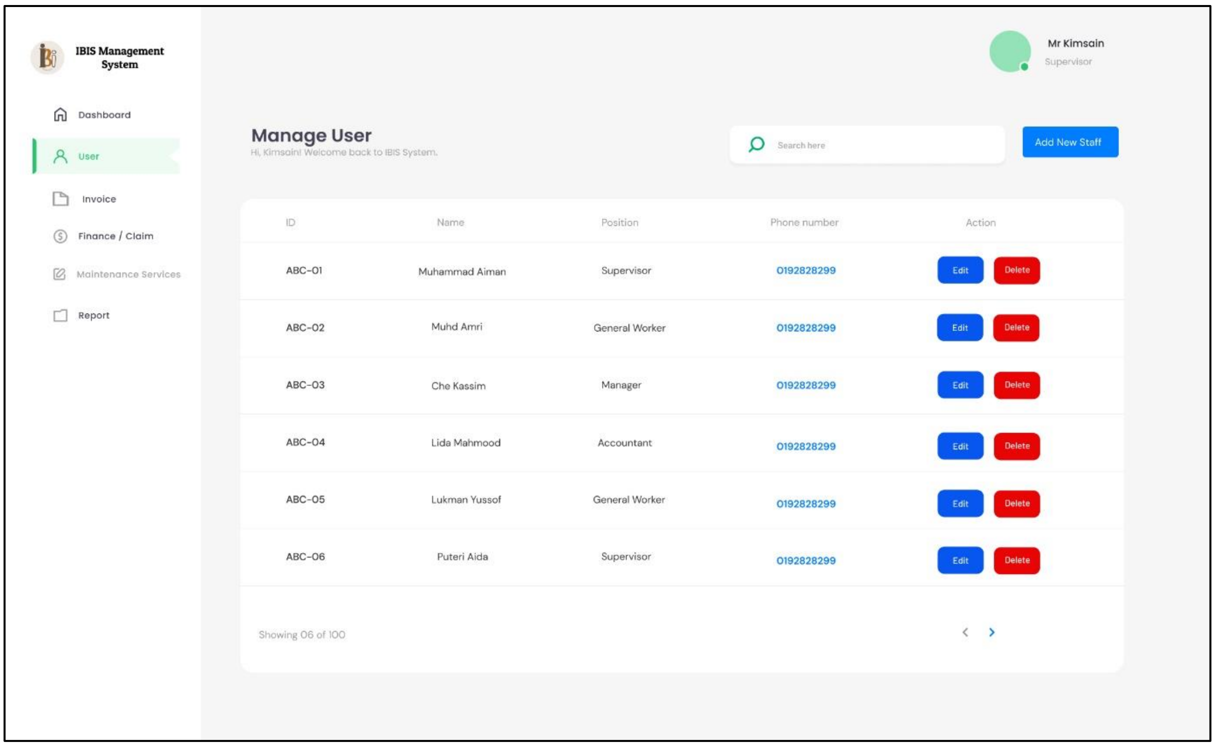
Address

Address

☑ I agree to Square's Cookie and Privacy Policy.

Figure 37 : Interface of registration page

3.6.2.3 Manage User Page



Manage User

Hi, Kimsain! Welcome back to IBIS System.

ID	Name	Position	Phone number	Action
ABC-01	Muhammad Aiman	Supervisor	0192828299	Edit Delete
ABC-02	Muhd Amri	General Worker	0192828299	Edit Delete
ABC-03	Che Kassim	Manager	0192828299	Edit Delete
ABC-04	Lida Mahmood	Accountant	0192828299	Edit Delete
ABC-05	Lukman Yussof	General Worker	0192828299	Edit Delete
ABC-06	Puteri Aida	Supervisor	0192828299	Edit Delete

Showing 06 of 100

Figure 38 : Interface of Manage User page

3.6.2.4 Dashboard Page

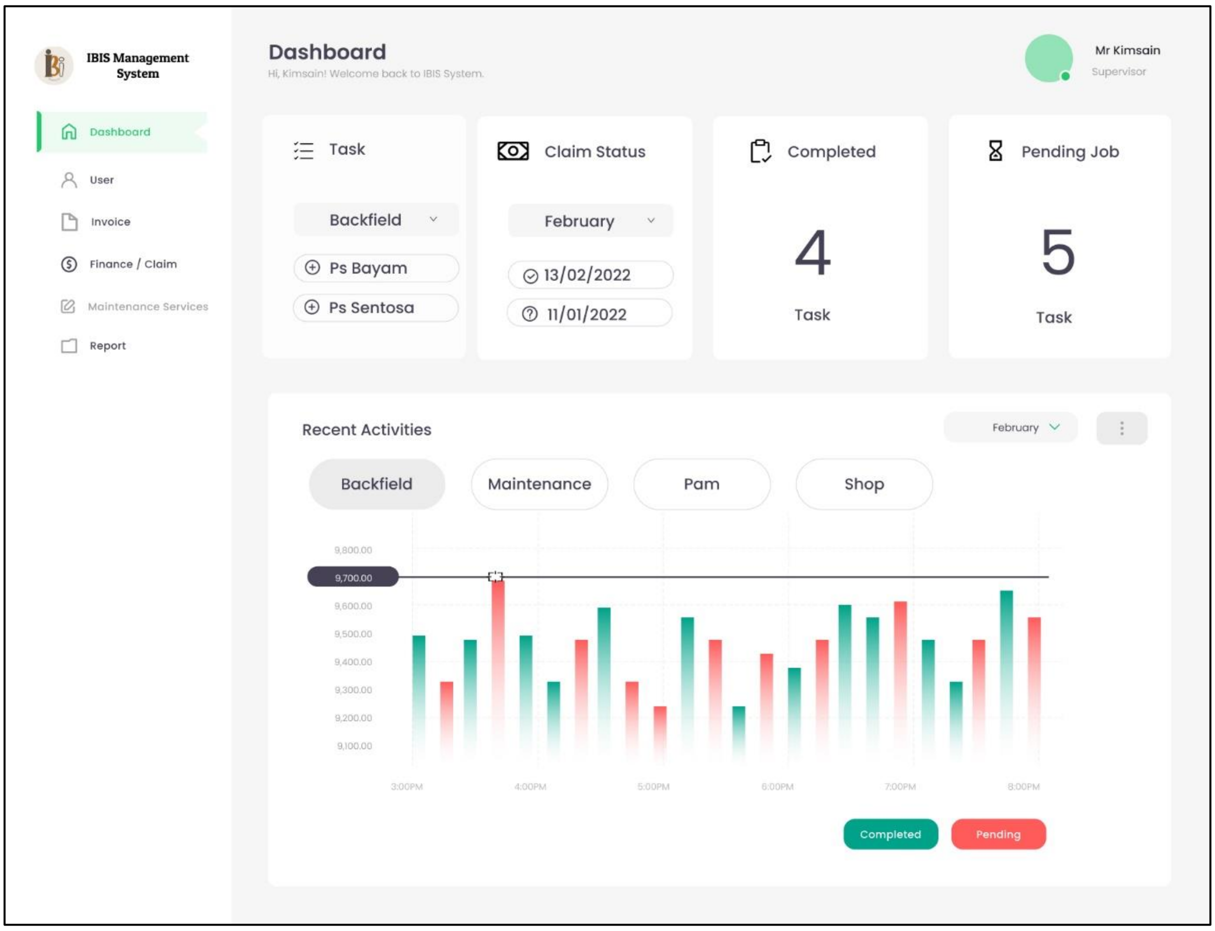
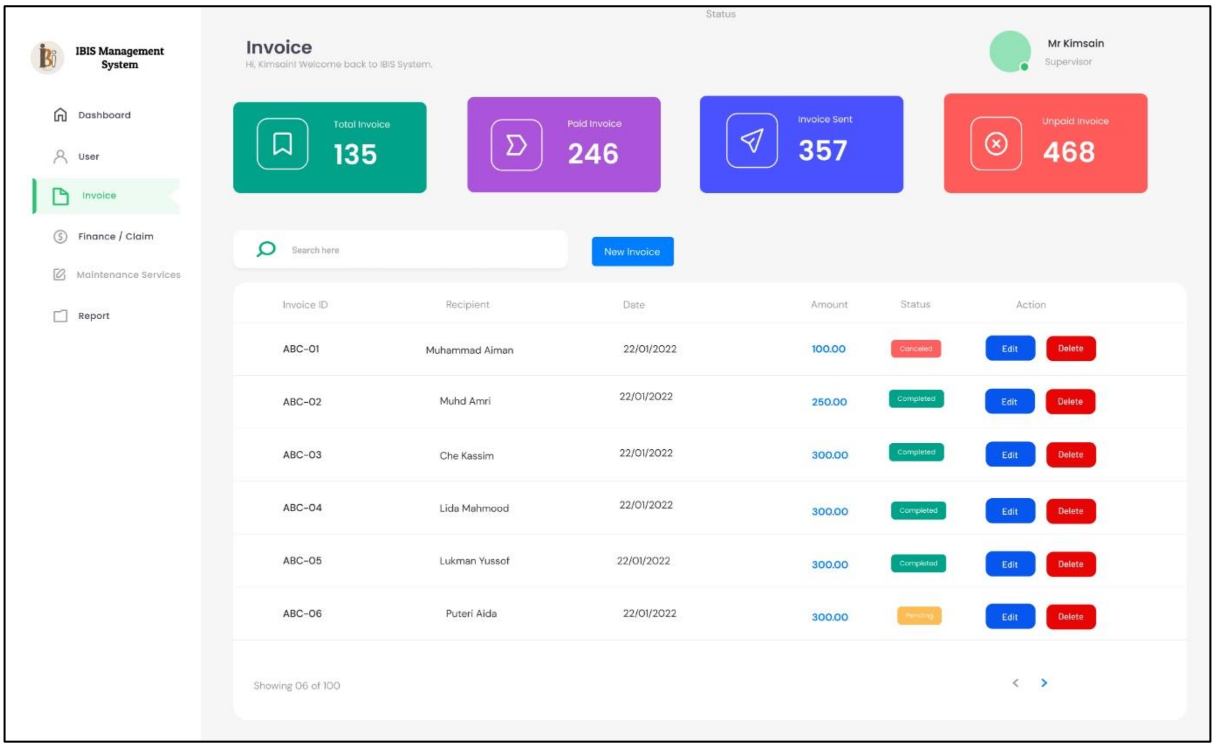


Figure 39 : Interface of Dashboard Page

3.6.2.5 Invoice Page



IBIS Management System

Dashboard
User
Invoice
Finance / Claim
Maintenance Services
Report

Status

Invoice
Hi, Kimsain! Welcome back to IBIS System.

Mr Kimsain
Supervisor

Total Invoice 135
Paid Invoice 246
Invoice Sent 357
Unpaid invoice 468

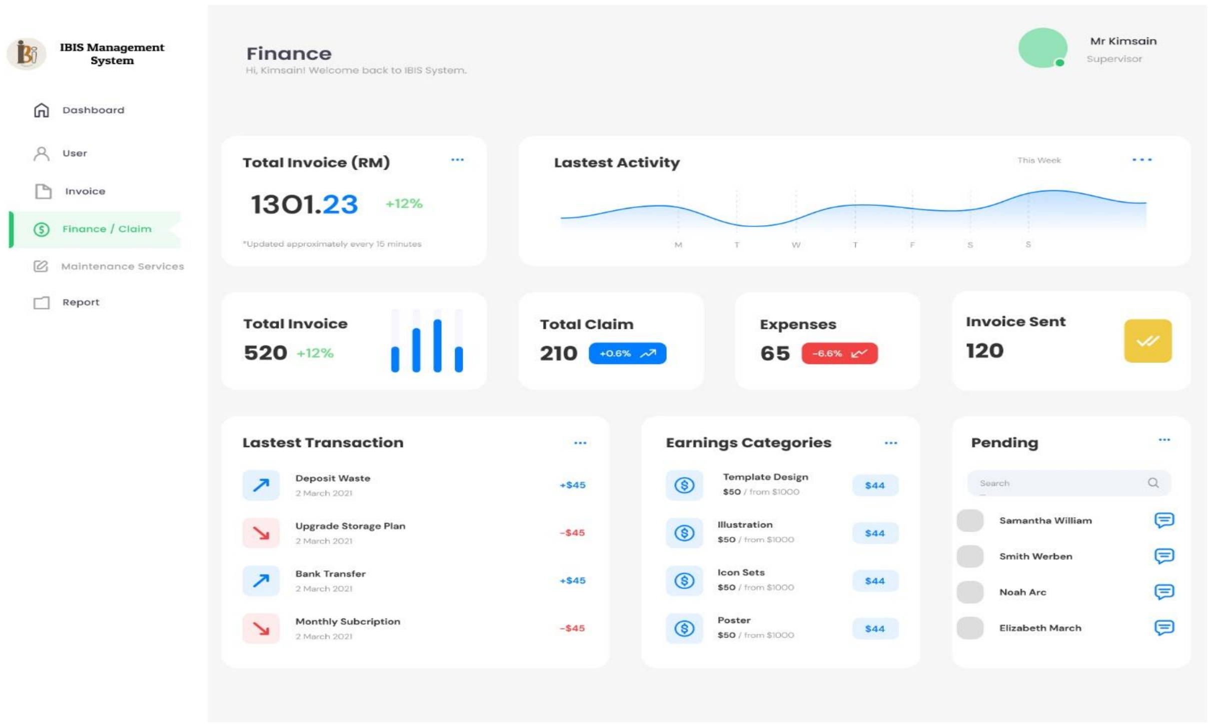
Search here
New Invoice

Invoice ID	Recipient	Date	Amount	Status	Action
ABC-01	Muhammad Aiman	22/01/2022	100.00	Canceled	Edit Delete
ABC-02	Muhd Amri	22/01/2022	250.00	Completed	Edit Delete
ABC-03	Che Kassim	22/01/2022	300.00	Completed	Edit Delete
ABC-04	Lida Mahmood	22/01/2022	300.00	Completed	Edit Delete
ABC-05	Lukman Yussof	22/01/2022	300.00	Completed	Edit Delete
ABC-06	Puteri Aida	22/01/2022	300.00	Pending	Edit Delete

Showing 06 of 100

Figure 40 : Interface of Invoice Page

3.6.2.6 Finance Page



IBIS Management System

Dashboard
User
Invoice
Finance / Claim
Maintenance Services
Report

Finance
Hi, Kimsain! Welcome back to IBIS System.

Mr Kimsain
Supervisor

Total Invoice (RM)
1301.23 +12%
*Updated approximately every 15 minutes

Lastest Activity
This Week

Total Invoice 520 +12%
Total Claim 210 +0.6%
Expenses 65 -6.6%
Invoice Sent 120

Lastest Transaction

Deposit Waste	2 March 2021	+$45
Upgrade Storage Plan	2 March 2021	-$45
Bank Transfer	2 March 2021	+$45
Monthly Subcription	2 March 2021	-$45

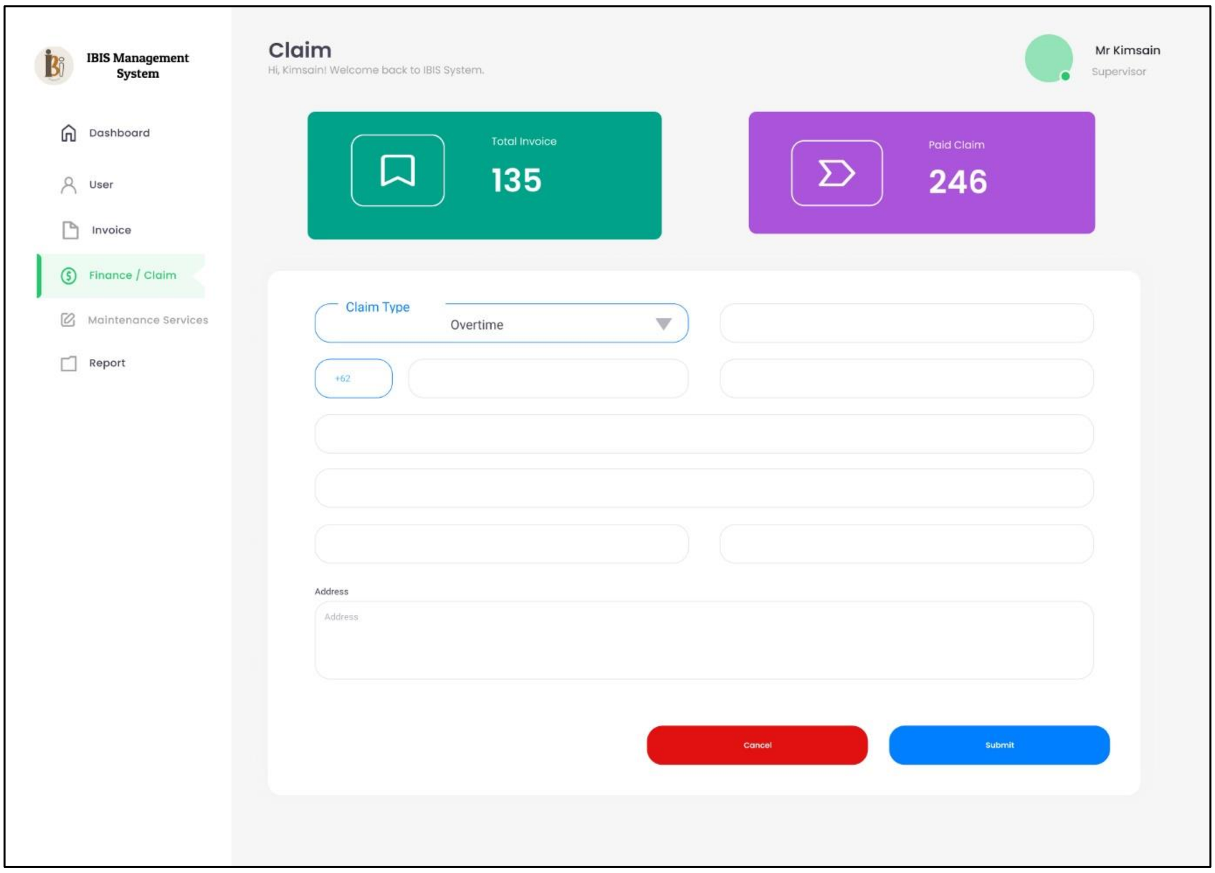
Earnings Categories

Template Design	$50 / from $1000	$44
Illustration	$50 / from $1000	$44
Icon Sets	$50 / from $1000	$44
Poster	$50 / from $1000	$44

Pending
Search
Samantha William
Smith Werben
Noah Arc
Elizabeth March

Figure 41 : Interface of Finance page

3.6.2.7 Claim Page



Claim

Hi, Kimsain! Welcome back to IBIS System.

Mr Kimsain

Supervisor

Total Invoice 135

Paid Claim 246

Claim Type: Overtime

+62

Address

Address

Figure 42 : Interface of Claim Page

3.6.2.8 Maintenance and Services Page

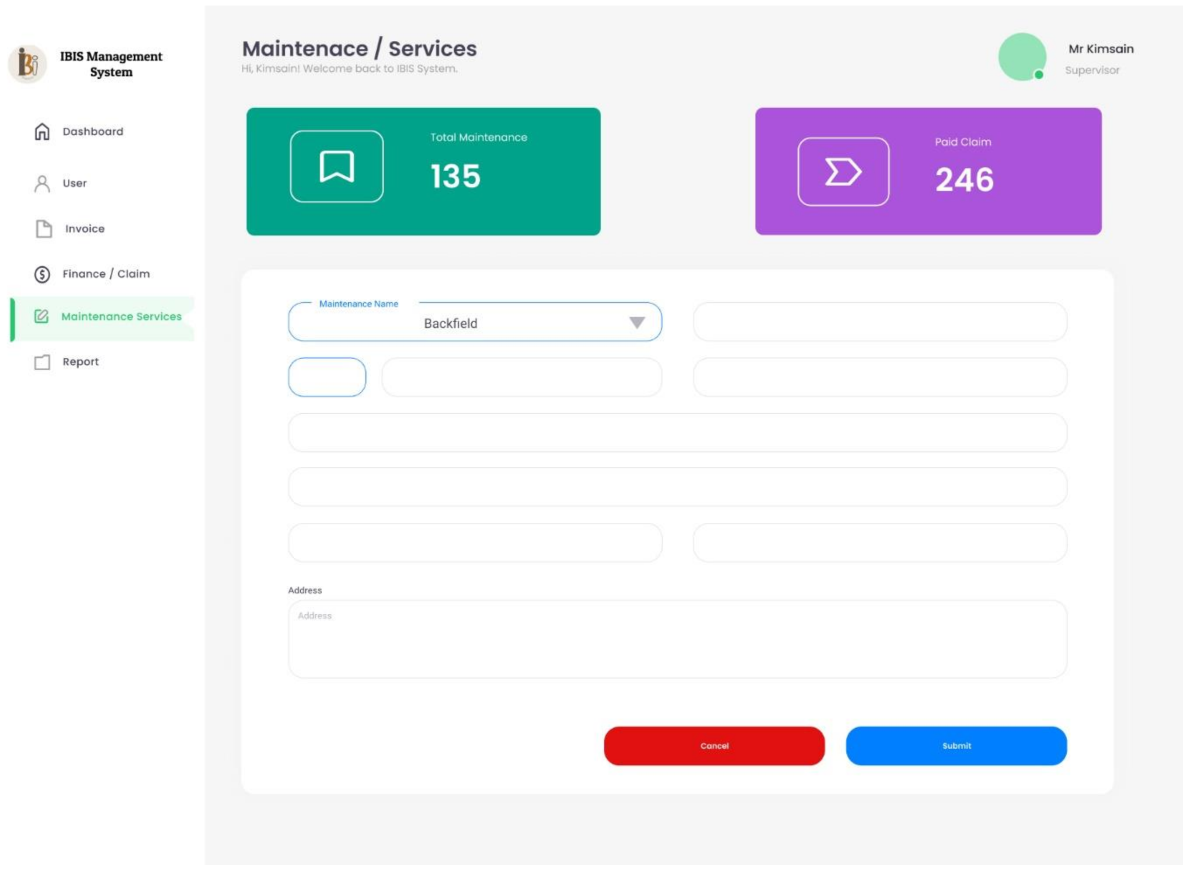


Figure 43 : Interface of Maintenance Services Page

3.6.2.3 Report Page

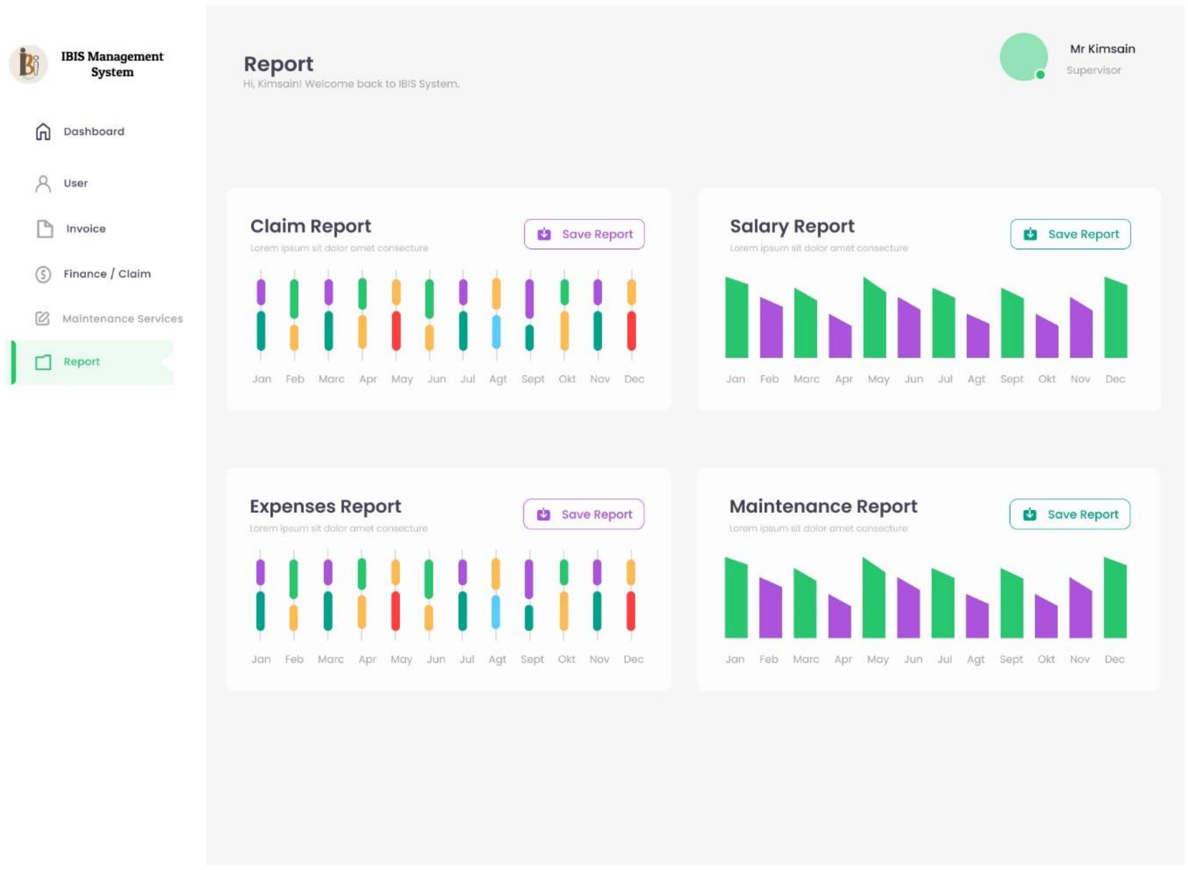

IBIS Management System
Dashboard
User
Invoice
Finance / Claim
Maintenance Services
Report
Report
Hi, Kimsain! Welcome back to IBIS System.
Mr Kimsain
Supervisor
Claim Report
Lorem ipsum sit dolor amet consecture
Save Report
Salary Report
Lorem ipsum sit dolor amet consecture
Save Report
Expenses Report
Lorem ipsum sit dolor amet consecture
Save Report
Maintenance Report
Lorem ipsum sit dolor amet consecture
Save Report
Jan Feb Marc Apr May Jun Jul Agt Sept Okt Nov Dec

Figure 44 : Interface of Report Page

3.7 Testing Plan

IBI Management system (IBIS) is used User Acceptance Test (UAT) for testing plan. This testing will be handled to test the functionality including the button and requirements. UAT will be conduct by the end user or third party to ensure all the button, form and data displayed is meet the user requirements.

No.	Module	Activities	Status		Comments
1.	Registration	Navigation bar	Yes ( )	No ( )	
		Button	Yes ( )	No ( )	
2.	Login	User registration	Yes ( )	No ( )	
		User login	Yes ( )	No ( )	
		User logout	Yes ( )	No ( )	
		Password retrieval	Yes ( )	No ( )	
3.	Register	New user registration	Yes ( )	No ( )	
		New user login	Yes ( )	No ( )	
		New user logout	Yes ( )	No ( )	
		Password retrieval	Yes ( )	No ( )	
4.	Manage Financial Resources	View dashboard	Yes ( )	No ( )	
		Add Claim button	Yes ( )	No ( )	
		Claim Form	Yes ( )	No ( )	
		Submit Form	Yes ( )	No ( )	
		Edit Form	Yes ( )	No ( )	
		Delete Form	Yes ( )	No ( )	
		View Chart and graph	Yes ( )	No ( )	
5.	Manage Maintenance and Services	View dashboard	Yes ( )	No ( )	
		Add new task	Yes ( )	No ( )	
		Add information	Yes ( )	No ( )	
		Update information	Yes ( )	No ( )	
		Delete information	Yes ( )	No ( )	
6.	Generate Report	View dashboard	Yes ( )	No ( )	
		View Report	Yes ( )	No ( )	
		Add new report	Yes ( )	No ( )	
		Review report	Yes ( )	No ( )	

This test has been performed by:

Name	: ______________________________
Signature	: ______________________________
Date	: ______________________________

3.8 Potential Use of Proposed Solution

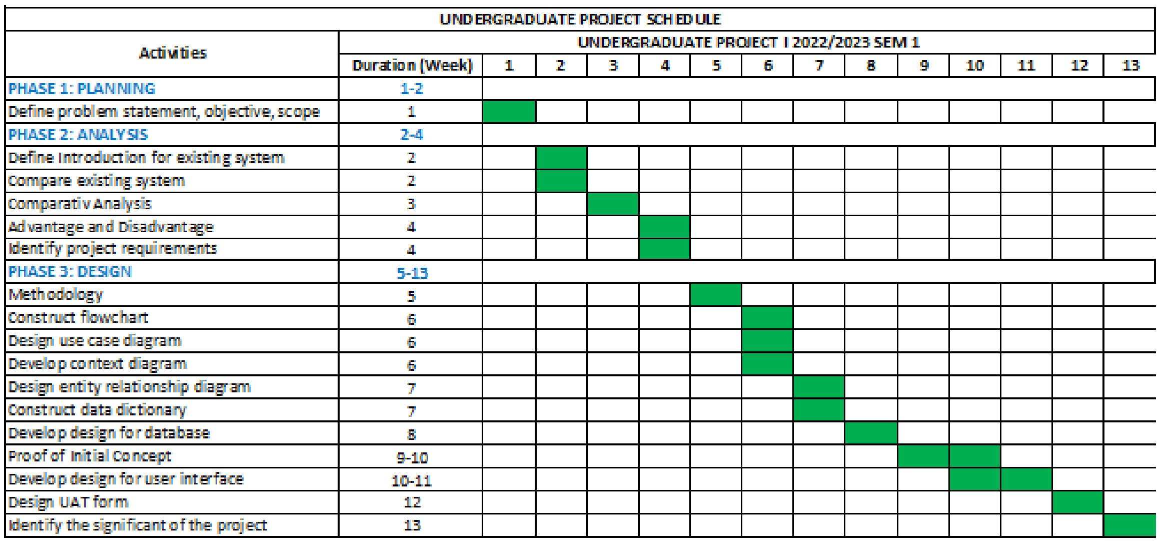
An IBIS Management System is a structure used within an organisation to organise and manage resources. By offering a framework for organising, directing, and controlling activities, it can be used to increase productivity, efficiency, and overall performance. Besides, the functionalities in the IBIS will ease the staff including the human resources administration and accountant to manage the report, finance part and to monitor the other workers.

IBIS provides several modules that will assist the user in managing their work as below:

1. IBIS provides maintenance and service module so that the user can add new task and update the task in that section. It will ease the supervisor to arrange the schedule for workers and notify them by using this module.
2. IBIS provides Financial Resources module that benefits the human resources administrator and account in managing the company finance including the report and company expenses.

In order to complete the IBI Management System in a much more organised and systematic manner, IBIS will serve as and develop into the best system for all users.

3.9 Gantt Chart



UNDERGRADUATE PROJECT SCHEDULE														
Activities	UNDERGRADUATE PROJECT I 2022/2023 SEM 1													
	Duration (Week)	1	2	3	4	5	6	7	8	9	10	11	12	13
PHASE 1: PLANNING	1-2													
Define problem statement, objective, scope	1	🟩												
PHASE 2: ANALYSIS	2-4													
Define Introduction for existing system	2		🟩											
Compare existing system	2		🟩											
Comparativ Analysis	3			🟩										
Advantage and Disadvantage	4				🟩									
Identify project requirements	4				🟩									
PHASE 3: DESIGN	5-13													
Methodology	5					🟩								
Construct flowchart	6						🟩							
Design use case diagram	6						🟩							
Develop context diagram	6						🟩							
Design entity relationship diagram	7							🟩						
Construct data dictionary	7							🟩						
Develop design for database	8								🟩					
Proof of Initial Concept	9-10									🟩	🟩			
Develop design for user interface	10-11										🟩	🟩		
Design UAT form	12												🟩	
Identify the significant of the project	13													🟩

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter will provide an explanation of the process and methodology utilized for the deployment of the IBIS Management System. A detailed description of the system flow will be given from its initial stages to its completion. Moreover, in this chapter also provide a user manual to offer the guidance to users on how to operate the finished system. It will compromise step-by-step instruction accompanied by screenshots and explanation to ensure the user understand the system. In order to achieve the best possible performance of the system, a form for User Acceptance Test (UAT) will be created to evaluate both functional and non-functional aspects of the system. The users will be invited to share their experience and provide feedback regarding the system. It is worth emphasizing that conducting various tests is crucial to conduct a comprehensive evaluation of the system's performance. This ensures that the system fulfils all the necessary requirements and operates according to expectations. The UAT form will play a significant role in this process by providing valuable information about the system's performance and identifying areas that require improvement.

4.2 Implementation

The system's implementation involved the use of several web-based programming languages, namely PHP, CSS, and Javascript. Visual Studio Code software was utilized to write source code for each interface design during the development process. XAMPP was employed to set up a temporary server for the demo, before the finalized system is deployed into a real server. The web-server is responsible for connecting the web to the database, which is MySQL, as this connection is critical for proper system functioning and data storage and retrieval. The IBIS-ms encompasses four types of users, namely Human Resource, Accountant, Supervisor, and Workers, and the interfaces developed for the system adhered to the client's requirements with respect to user types. This implies that every user has their unique interface with distinct functions within the system.

4.2.1 User Manual and Development Progress

This section will offer a detailed description of each interface included in the IBIS-ms, with the aid of step-by-step instructions to assist users in effectively utilizing the system. The interfaces are designed to be user-friendly and engaging, incorporating straightforward instructions and easy-to-use functionalities.

4.2.1.1 Login

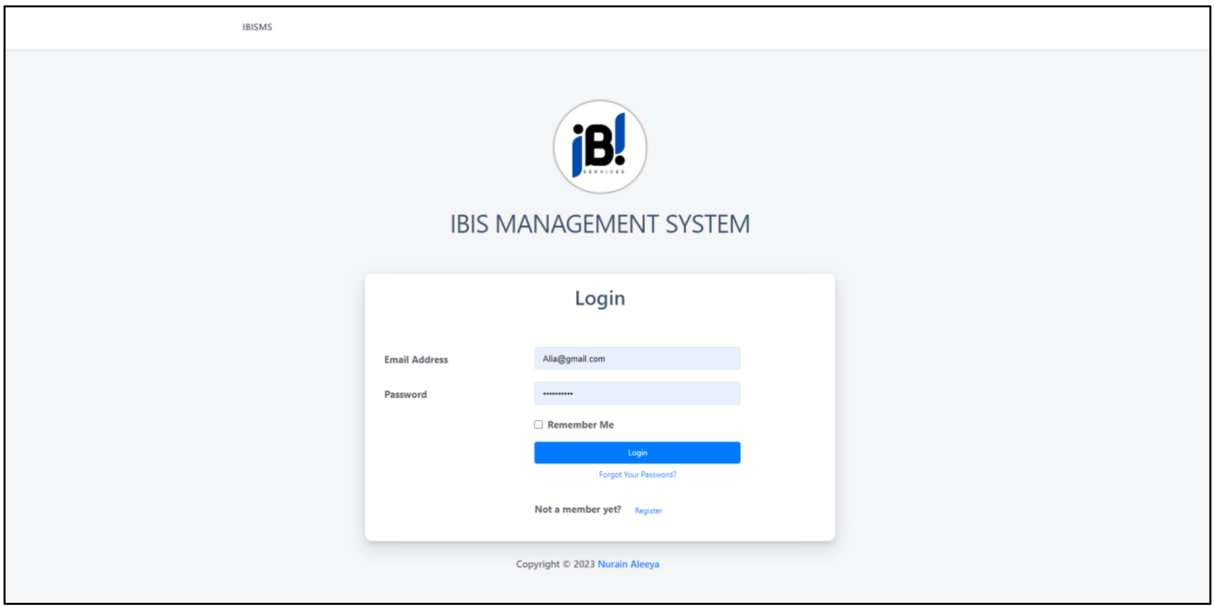
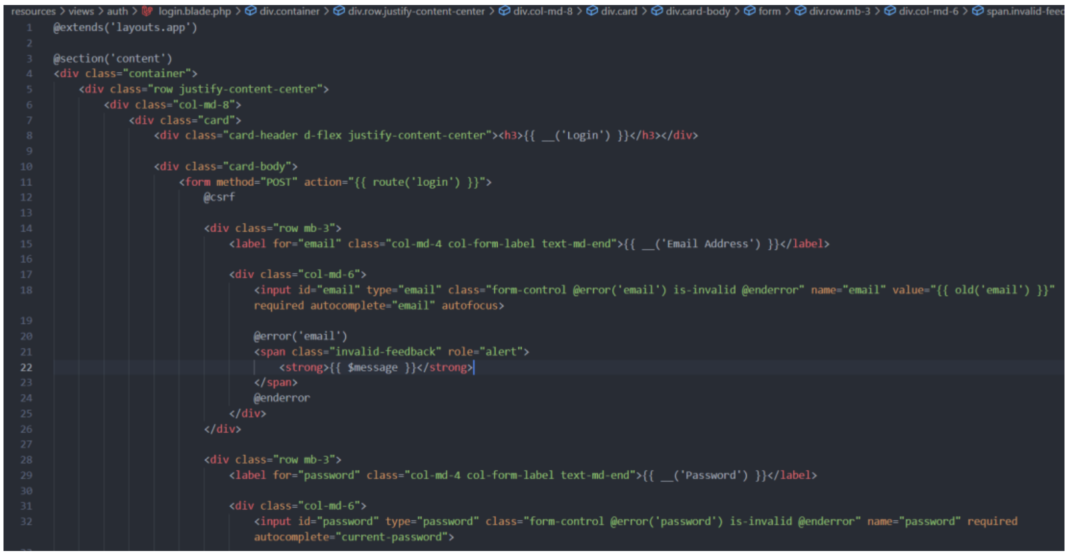


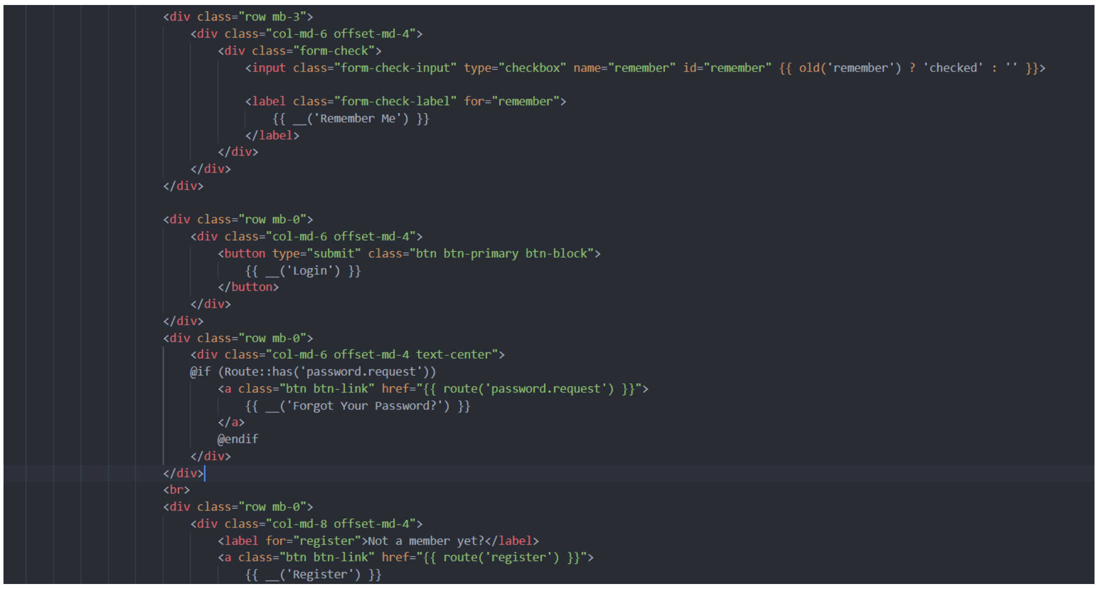
Figure 45 : Login Page

The login page for IBIS Management System serves as the primary gateway for users to access the system's features and functions. The page typically requires users to input their email address and password, which The HR Manager has previously registered with the system. Upon inputting valid login credentials, the user gains access to the system's interfaces and functionalities, which are unique to their designated user type. For instance, an HR Manager will have access to different interfaces and functions compared to an Accountant or a Supervisor. The login page is designed to ensure the security and integrity of the system, preventing unauthorized access and maintaining confidentiality of the user's data. Additionally, it allows the system to track and record each user's actions and activities within the system, enabling efficient auditing and monitoring.



```
resources > views > auth > login.blade.php > div.container > div.row.justify-content-center > div.col-md-8 > div.card > div.card-body > form > div.row.mb-3 > div.col-md-6 > span.invalid-feed
@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row justify-content-center">
        <div class="col-md-8">
            <div class="card">
                <div class="card-header d-flex justify-content-center"><h3>{{ __('Login') }}</h3></div>

                <div class="card-body">
                    <form method="POST" action="{{ route('login') }}">
                        @csrf

                        <div class="row mb-3">
                            <label for="email" class="col-md-4 col-form-label text-md-end">{{ __('Email Address') }}</label>

                            <div class="col-md-6">
                                <input id="email" type="email" class="form-control @error('email') is-invalid @enderror" name="email" value="{{ old('email') }}"
                                required autocomplete="email" autofocus>

                                @error('email')
                                <span class="invalid-feedback" role="alert">
                                    <strong>{{ $message }}</strong>
                                </span>
                                @enderror
                            </div>
                        </div>

                        <div class="row mb-3">
                            <label for="password" class="col-md-4 col-form-label text-md-end">{{ __('Password') }}</label>

                            <div class="col-md-6">
                                <input id="password" type="password" class="form-control @error('password') is-invalid @enderror" name="password" required
                                autocomplete="current-password">
```

Figure 46: Login Codes



```
                        <div class="row mb-3">
                            <div class="col-md-6 offset-md-4">
                                <div class="form-check">
                                    <input class="form-check-input" type="checkbox" name="remember" id="remember" {{ old('remember') ? 'checked' : '' }}>

                                    <label class="form-check-label" for="remember">
                                        {{ __('Remember Me') }}
                                    </label>
                                </div>
                            </div>
                        </div>

                        <div class="row mb-0">
                            <div class="col-md-6 offset-md-4">
                                <button type="submit" class="btn btn-primary btn-block">
                                    {{ __('Login') }}
                                </button>
                            </div>
                        </div>
                        <div class="row mb-0">
                            <div class="col-md-6 offset-md-4 text-center">
                            @if (Route::has('password.request'))
                                <a class="btn btn-link" href="{{ route('password.request') }}">
                                    {{ __('Forgot Your Password?') }}
                                </a>
                                @endif
                            </div>
                        </div>
                        <br>
                        <div class="row mb-0">
                            <div class="col-md-8 offset-md-4">
                                <label for="register">Not a member yet?</label>
                                <a class="btn btn-link" href="{{ route('register') }}">
                                    {{ __('Register') }}
```

Figure 47 : Login codes

Figure above shows the code provided is a HTML code that represents the login page for IBIS management system. The page includes a form that allows the user to input their email address and password that have been registered to log into the system. The form has a "Remember Me" checkbox for the user to select if they want the system to remember their login information. The page also has a link for users who have forgotten their password, which directs them to the password reset page. Additionally, the page includes a link for users who are not yet registered to become a member of the system. The design of the page is responsive and user-friendly. The page is written in Laravel framework using PHP, with bootstrap for styling the HTML elements.

4.2.1.2 Dashboard

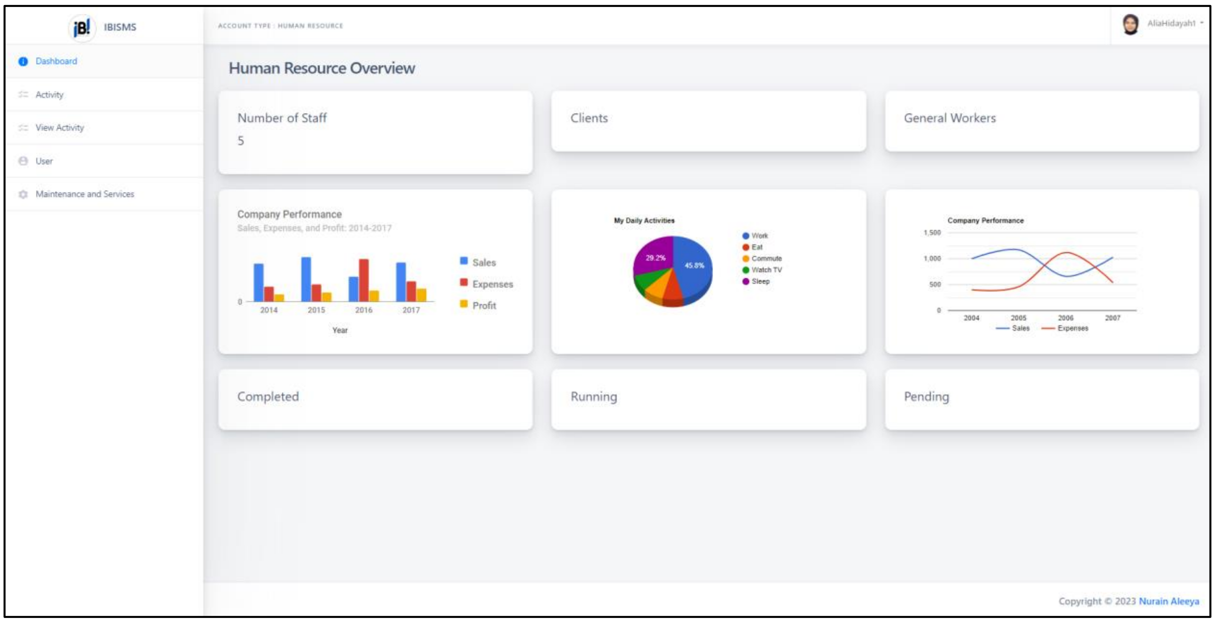
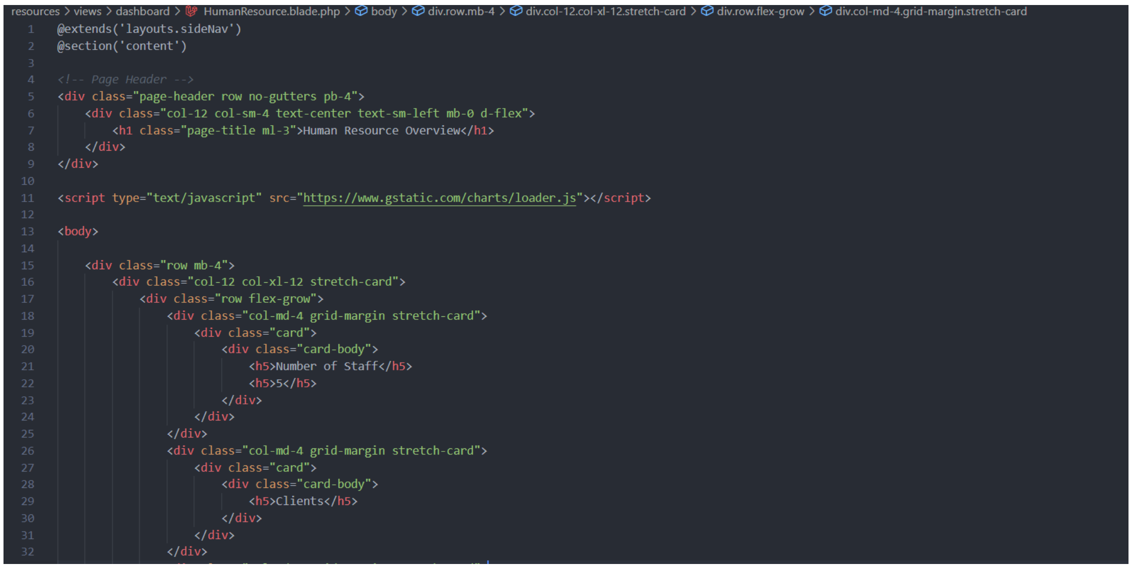


Figure 48 : Dashboard Page

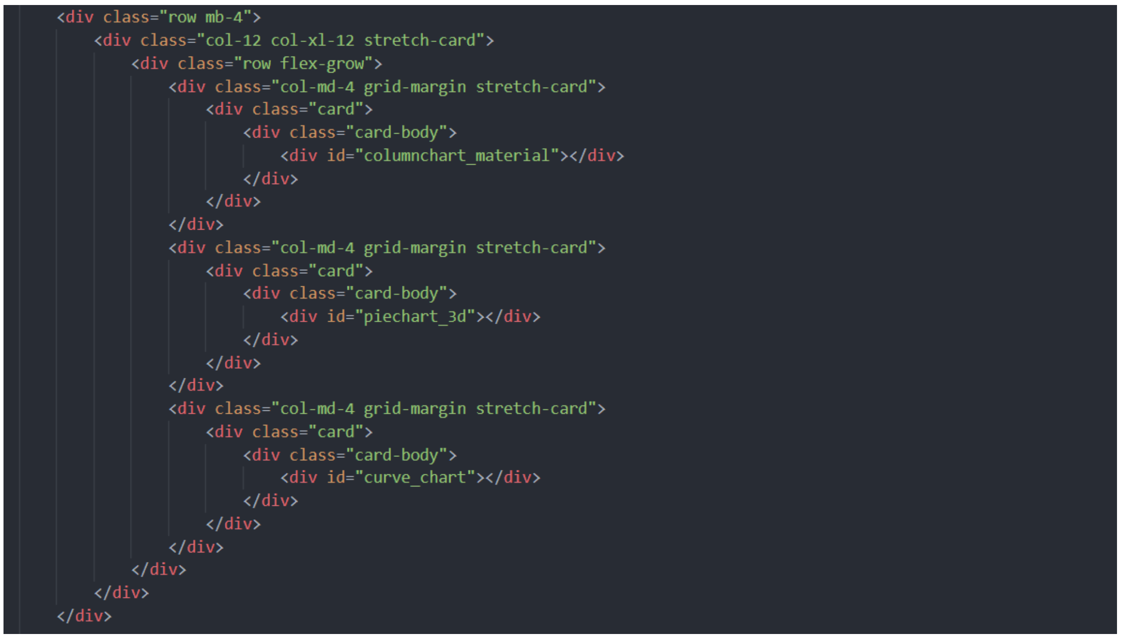
The dashboard interface of the IBIS management system is designed to display important company information in an easy-to-understand manner. The dashboard contains various graphical representations, such as bar graphs, pie charts, and line charts, that show company data in an intuitive way. For example, the bar graph may display information about the number of employees in each department, while the pie chart may show the distribution of company expenses across different categories. The line chart may display the company's financial performance over time, showing trends in revenue and expenses.



resources > views > dashboard > HumanResource.blade.php > body > div.row.mb-4 > div.col-12.col-xl-12.stretch-card > div.row.flex-grow > div.col-md-4.grid-margin.stretch-card

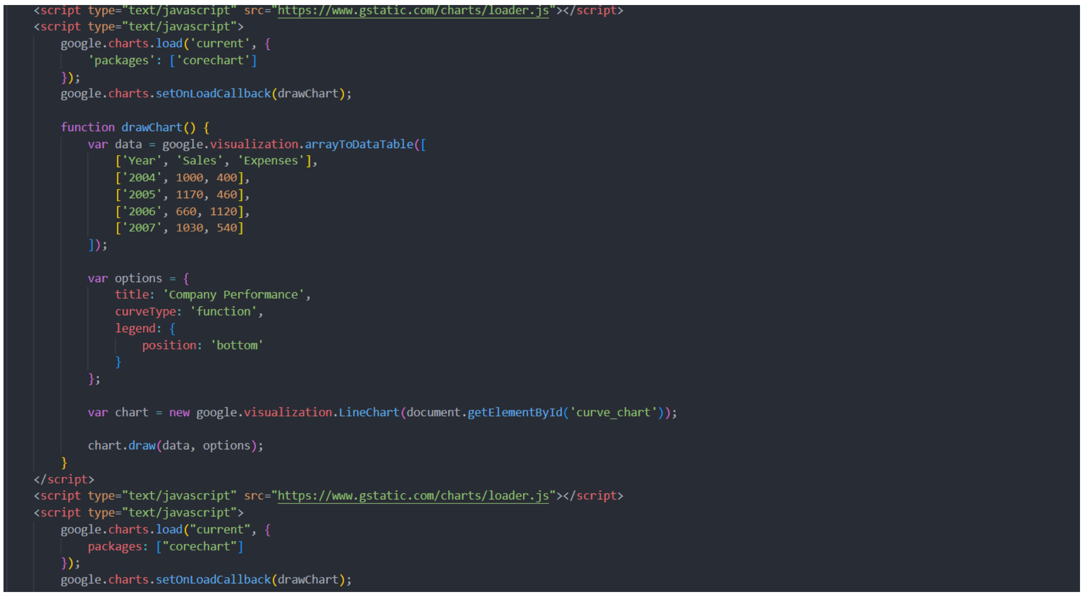
```
@extends('layouts.sideNav')
@section('content')

<!-- Page Header -->
<div class="page-header row no-gutters pb-4">
    <div class="col-12 col-sm-4 text-center text-sm-left mb-0 d-flex">
        <h1 class="page-title ml-3">Human Resource Overview</h1>
    </div>
</div>

<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>

<body>

    <div class="row mb-4">
        <div class="col-12 col-xl-12 stretch-card">
            <div class="row flex-grow">
                <div class="col-md-4 grid-margin stretch-card">
                    <div class="card">
                        <div class="card-body">
                            <h5>Number of Staff</h5>
                            <h5>5</h5>
                        </div>
                    </div>
                </div>
                <div class="col-md-4 grid-margin stretch-card">
                    <div class="card">
                        <div class="card-body">
                            <h5>Clients</h5>
                        </div>
                    </div>
                </div>
```

Figure 49 : Dashboard Codes



```html
<div class="row mb-4">
    <div class="col-12 col-xl-12 stretch-card">
        <div class="row flex-grow">
            <div class="col-md-4 grid-margin stretch-card">
                <div class="card">
                    <div class="card-body">
                        <div id="columnchart_material"></div>
                    </div>
                </div>
            </div>
            <div class="col-md-4 grid-margin stretch-card">
                <div class="card">
                    <div class="card-body">
                        <div id="piechart_3d"></div>
                    </div>
                </div>
            </div>
            <div class="col-md-4 grid-margin stretch-card">
                <div class="card">
                    <div class="card-body">
                        <div id="curve_chart"></div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

Figure 50 : Dashboard Codes

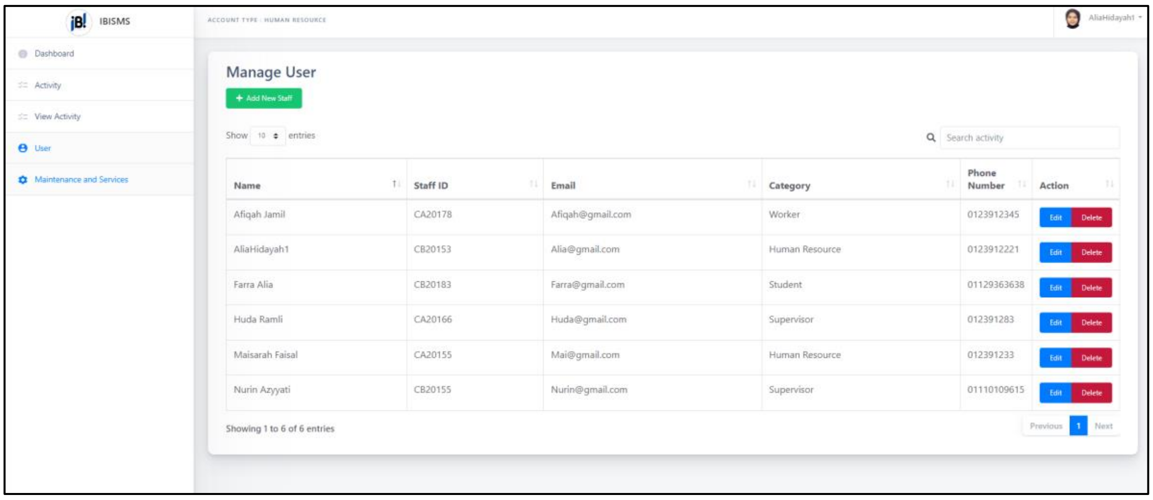


```html
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
    google.charts.load('current', {
        'packages': ['corechart']
    });
    google.charts.setOnLoadCallback(drawChart);

    function drawChart() {
        var data = google.visualization.arrayToDataTable([
            ['Year', 'Sales', 'Expenses'],
            ['2004', 1000, 400],
            ['2005', 1170, 460],
            ['2006', 660, 1120],
            ['2007', 1030, 540]
        ]);

        var options = {
            title: 'Company Performance',
            curveType: 'function',
            legend: {
                position: 'bottom'
            }
        };

        var chart = new google.visualization.LineChart(document.getElementById('curve_chart'));

        chart.draw(data, options);
    }
</script>
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
    google.charts.load("current", {
        packages: ["corechart"]
    });
    google.charts.setOnLoadCallback(drawChart);
```

Figure 51 : Dashboard Codes

The given code is a web page content that shows an overview of human resources using Google Charts API. The page has a header with the title "Human Resource Overview". The body contains several sections, each with a row that contains three cards showing different pieces of information. The first row contains three cards showing the number of staff, clients, and general workers. The second row contains three cards showing charts created using the Google Charts API. The first card shows a column chart, the second card shows a 3D pie chart, and the third card shows a curve chart. The third row contains three cards showing the number of completed, running, and pending tasks.

4.2.1.3 Manage User



Manage User

+ Add New Staff

Show 10 entries

Search activity

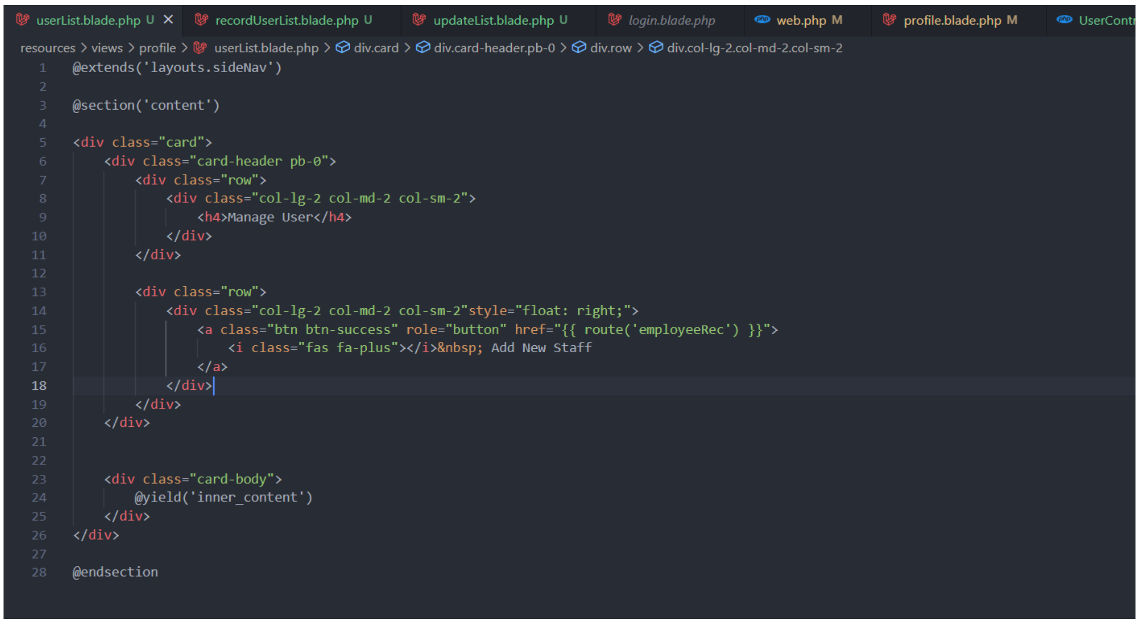
Name	Staff ID	Email	Category	Phone Number	Action
Afiqah Jamil	CA20178	Afiqah@gmail.com	Worker	0123912345	Edit Delete
AliaHidayah1	CB20153	Alia@gmail.com	Human Resource	0123912221	Edit Delete
Farra Alia	CB20183	Farra@gmail.com	Student	01129363638	Edit Delete
Huda Ramli	CA20166	Huda@gmail.com	Supervisor	012391283	Edit Delete
Maisarah Faisal	CA20155	Mai@gmail.com	Human Resource	012391233	Edit Delete
Nurin Azyyati	CB20155	Nurin@gmail.com	Supervisor	01110109615	Edit Delete

Showing 1 to 6 of 6 entries

Previous 1 Next

Figure 52 : Manage *User Page*

Figure 8 shows the list of user that has been registered in the system. The Human Resource Manager can click add new staff button register new employee into the system. They alco can click edit and delete button to edit user's information and delete from the system.

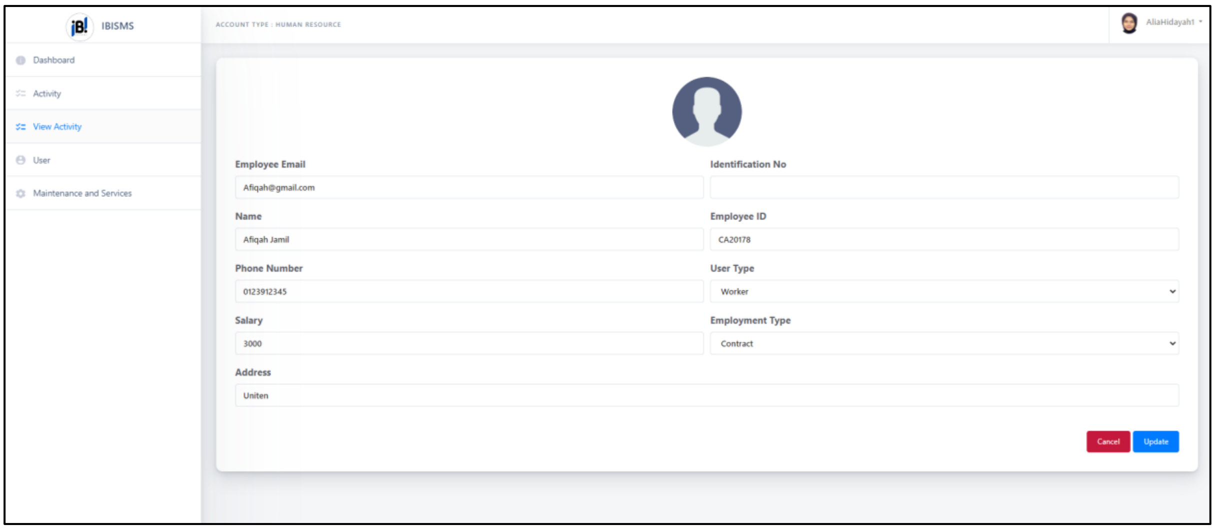

userList.blade.php U
recordUserList.blade.php U
updateList.blade.php U
login.blade.php
web.php M
profile.blade.php M
UserContr
resources > views > profile > userList.blade.php > div.card > div.card-header.pb-0 > div.row > div.col-lg-2.col-md-2.col-sm-2

```
@extends('layouts.sideNav')

@section('content')

<div class="card">
    <div class="card-header pb-0">
        <div class="row">
            <div class="col-lg-2 col-md-2 col-sm-2">
                <h4>Manage User</h4>
            </div>
        </div>

        <div class="row">
            <div class="col-lg-2 col-md-2 col-sm-2"style="float: right;">
                <a class="btn btn-success" role="button" href="{{ route('employeeRec') }}">
                    <i class="fas fa-plus"></i>&nbsp; Add New Staff
                </a>
            </div>
        </div>
    </div>


    <div class="card-body">
        @yield('inner_content')
    </div>
</div>

@endsection
```

Figure 53 : Manage User Codes

This code is a blade template for managing user information. It extends the "layouts.sideNav" template and includes a card header with a title and a button to add a new staff member. The card body includes a yield statement that allows for the inclusion of content from child templates.

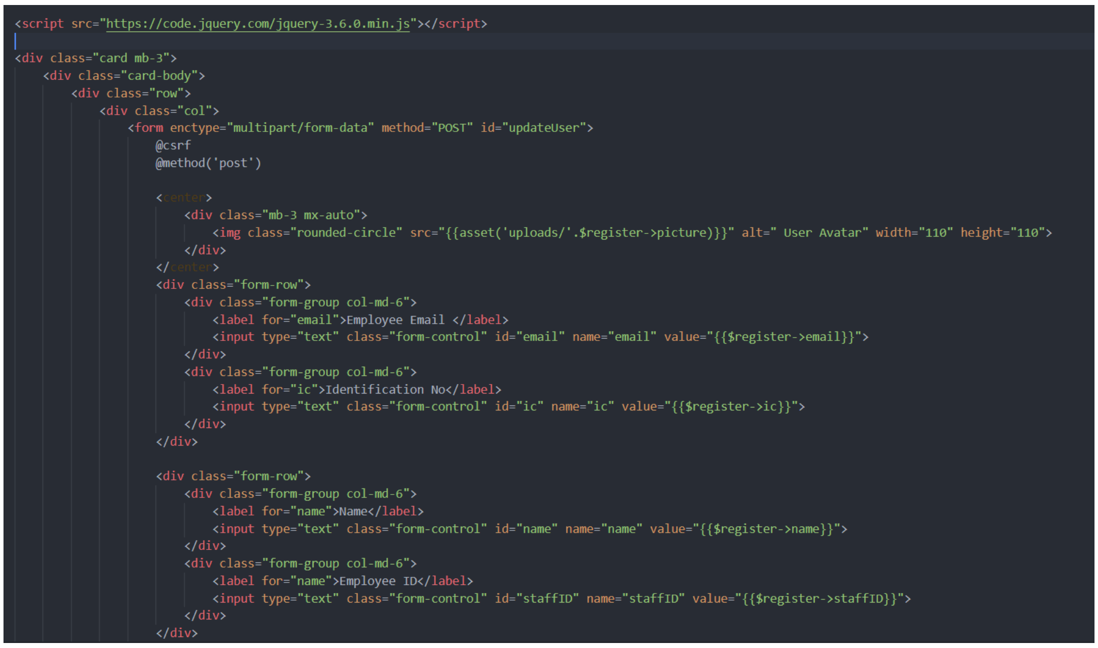
4.2.1.3.1 Manage User (Edit User’s Profile)



Employee Email	Identification No
Afiqah@gmail.com	
Name	Employee ID
Afiqah Jamil	CA20178
Phone Number	User Type
0123912345	Worker
Salary	Employment Type
3000	Contract
Address	
Uniten	

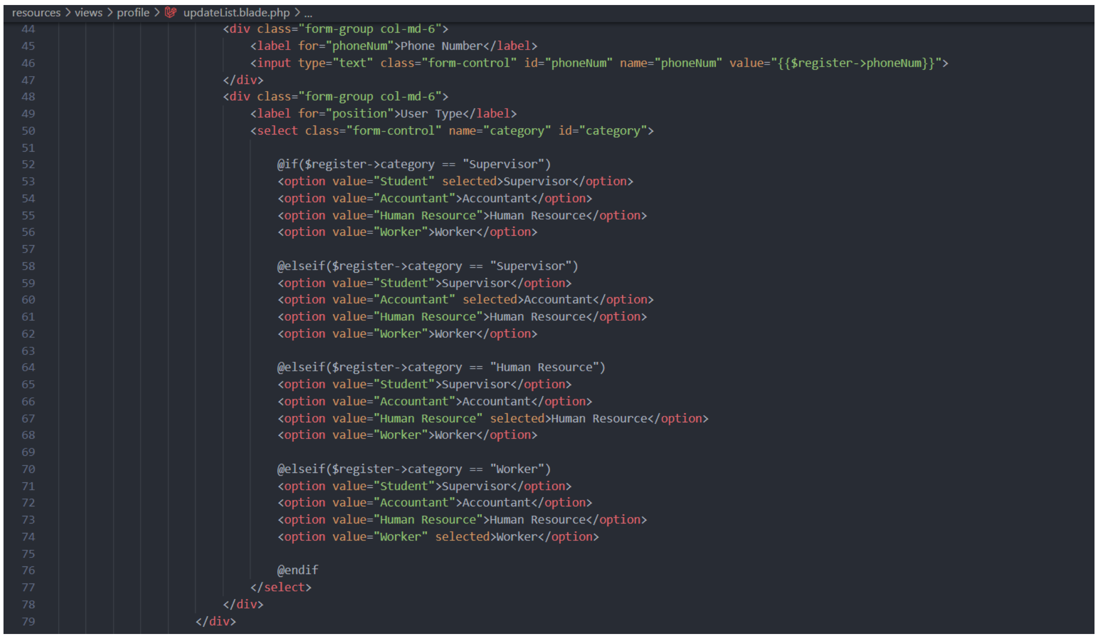
Figure 54 : Edit User Information

Figure 10 shows an edit page for an employee information form allows users to modify the information that was previously entered in the form. This page typically displays the existing employee information in editable fields, such as name, address, phone number, email, job title, department, and other relevant details. Users can make changes to any of the fields and save the updated information once they are done. The save button on the edit page triggers a process that updates the employee's information in the database, which makes the new information available for other users to view. For example, the form should verify that required fields are not left blank, input data is of the correct format, and that all necessary data is provided before allowing the user to save the changes.



```html
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<div class="card mb-3">
    <div class="card-body">
        <div class="row">
            <div class="col">
                <form enctype="multipart/form-data" method="POST" id="updateUser">
                    @csrf
                    @method('post')

                    <center>
                        <div class="mb-3 mx-auto">
                            <img class="rounded-circle" src="{{asset('uploads/'.$register->picture)}}" alt=" User Avatar" width="110" height="110">
                        </div>
                    </center>
                    <div class="form-row">
                        <div class="form-group col-md-6">
                            <label for="email">Employee Email </label>
                            <input type="text" class="form-control" id="email" name="email" value="{{$register->email}}">
                        </div>
                        <div class="form-group col-md-6">
                            <label for="ic">Identification No</label>
                            <input type="text" class="form-control" id="ic" name="ic" value="{{$register->ic}}">
                        </div>
                    </div>

                    <div class="form-row">
                        <div class="form-group col-md-6">
                            <label for="name">Name</label>
                            <input type="text" class="form-control" id="name" name="name" value="{{$register->name}}">
                        </div>
                        <div class="form-group col-md-6">
                            <label for="name">Employee ID</label>
                            <input type="text" class="form-control" id="staffID" name="staffID" value="{{$register->staffID}}">
                        </div>
                    </div>
```

Figure 55 : Edit User Codes



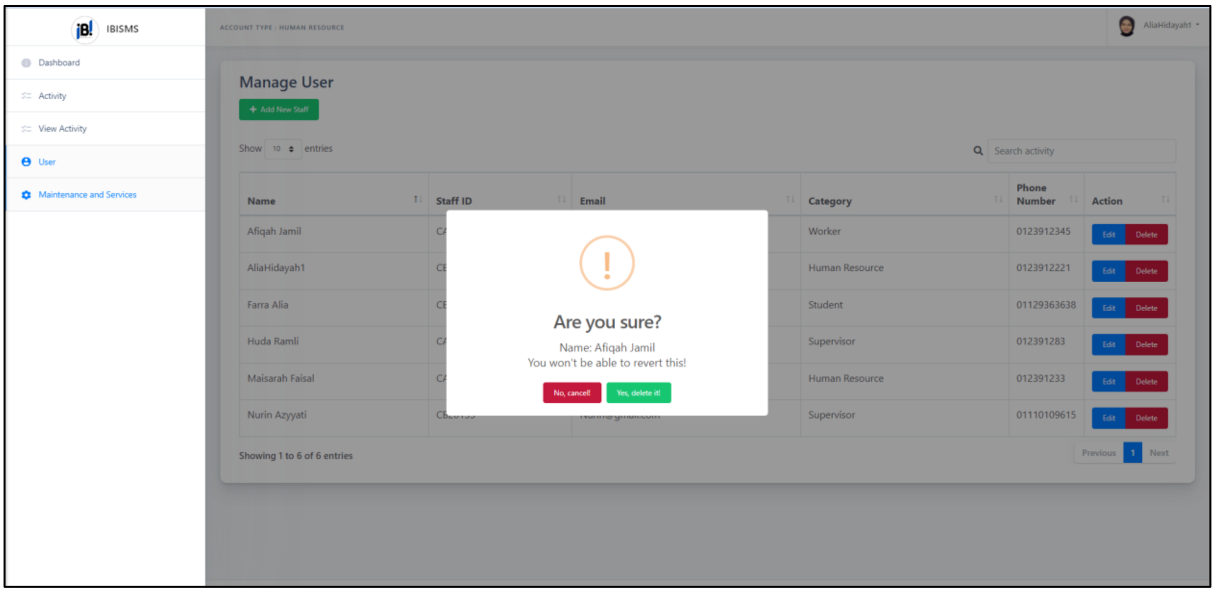
resources > views > profile > updateList.blade.php > ...

```html
                        <div class="form-group col-md-6">
                            <label for="phoneNum">Phone Number</label>
                            <input type="text" class="form-control" id="phoneNum" name="phoneNum" value="{{$register->phoneNum}}">
                        </div>
                        <div class="form-group col-md-6">
                            <label for="position">User Type</label>
                            <select class="form-control" name="category" id="category">

                                @if($register->category == "Supervisor")
                                <option value="Student" selected>Supervisor</option>
                                <option value="Accountant">Accountant</option>
                                <option value="Human Resource">Human Resource</option>
                                <option value="Worker">Worker</option>

                                @elseif($register->category == "Supervisor")
                                <option value="Student">Supervisor</option>
                                <option value="Accountant" selected>Accountant</option>
                                <option value="Human Resource">Human Resource</option>
                                <option value="Worker">Worker</option>

                                @elseif($register->category == "Human Resource")
                                <option value="Student">Supervisor</option>
                                <option value="Accountant">Accountant</option>
                                <option value="Human Resource" selected>Human Resource</option>
                                <option value="Worker">Worker</option>

                                @elseif($register->category == "Worker")
                                <option value="Student">Supervisor</option>
                                <option value="Accountant">Accountant</option>
                                <option value="Human Resource">Human Resource</option>
                                <option value="Worker" selected>Worker</option>

                                @endif
                            </select>
                        </div>
                    </div>
```

Figure 56: Edit User Codes

Figure 11 and 12 shows the code that shows a form to update user information. The form displays the current user information and allows the user to modify some fields. The form includes user email, identification number, name, employee ID, phone number, user type, salary, employment type, and address. The form is submitted using the POST method to a Laravel route called **updateUser**. The route accepts a parameter **$register->id**, which is used to identify the user being updated. The form includes two buttons: **Cancel** and **Update**. The **Cancel** button redirects the user to the previous URL.

4.2.1.3.2 Manage User (Delete User's Profile)



Manage User

Show 10 entries

Name	Staff ID	Email	Category	Phone Number	Action
Afiqah Jamil	CA		Worker	0123912345	Edit Delete
AliaHidayah1	CE		Human Resource	0123912221	Edit Delete
Farra Alia	CE		Student	01129363638	Edit Delete
Huda Ramli	CA		Supervisor	012391283	Edit Delete
Maisarah Faisal	CA		Human Resource	012391233	Edit Delete
Nurin Azyyati	CB		Supervisor	01110109615	Edit Delete

Showing 1 to 6 of 6 entries

Are you sure?

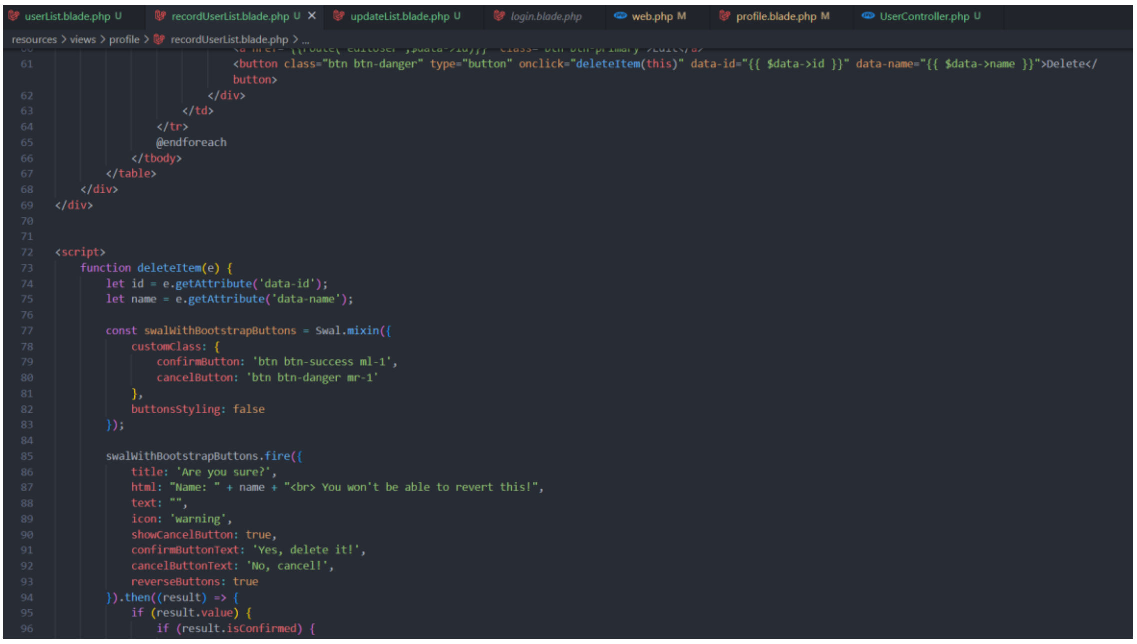
Name: Afiqah Jamil

You won't be able to revert this!

No, cancel! Yes, delete it!

Figure 57 : Delete User Alert

Figure 13 shows a delete user alert is a prompt that appears on a screen when a user attempts to delete a piece of data or information, such as a user profile or account. The purpose of the alert is to ensure that the user intends to delete the data and to prevent accidental or unintentional deletion. Typically, the alert will ask the user if they are sure they want to delete the data, and will provide two buttons for the user to click: "Yes" or "No". Clicking "Yes" will proceed with the deletion, while clicking "No" will cancel the action and return the user to the previous screen.

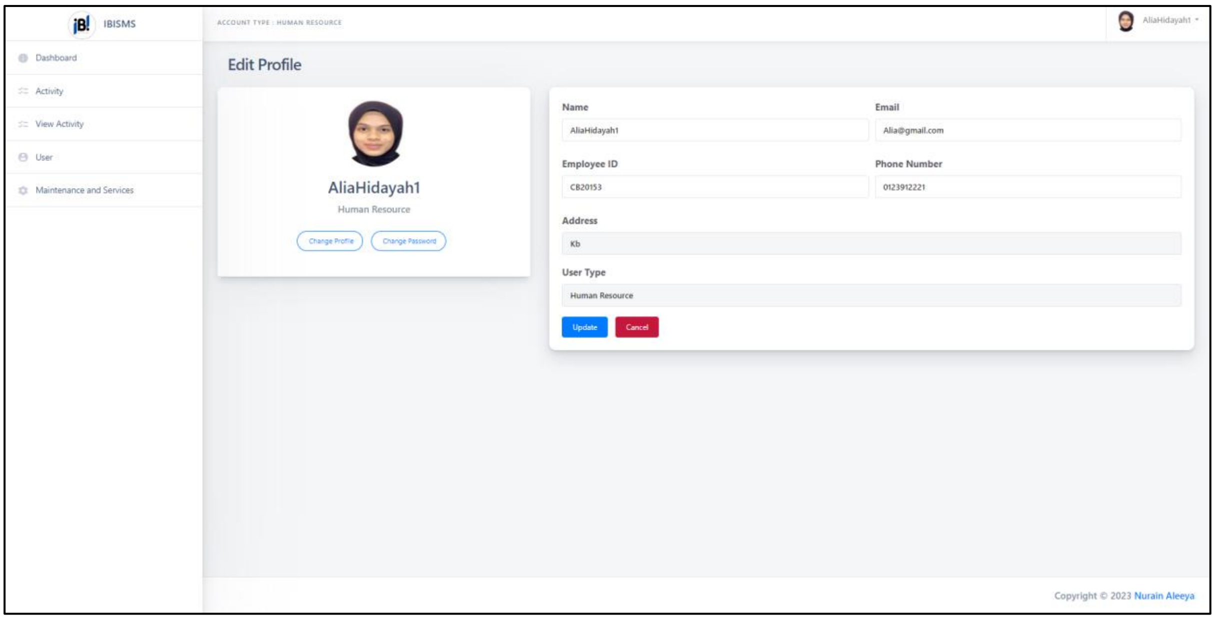


```
                        <button class="btn btn-danger" type="button" onclick="deleteItem(this)" data-id="{{ $data->id }}" data-name="{{ $data->name }}">Delete</
                        button>
                    </div>
                </td>
            </tr>
            @endforeach
        </tbody>
    </table>
</div>
</div>


<script>
    function deleteItem(e) {
        let id = e.getAttribute('data-id');
        let name = e.getAttribute('data-name');

        const swalWithBootstrapButtons = Swal.mixin({
            customClass: {
                confirmButton: 'btn btn-success ml-1',
                cancelButton: 'btn btn-danger mr-1'
            },
            buttonsStyling: false
        });

        swalWithBootstrapButtons.fire({
            title: 'Are you sure?',
            html: "Name: " + name + "<br> You won't be able to revert this!",
            text: "",
            icon: 'warning',
            showCancelButton: true,
            confirmButtonText: 'Yes, delete it!',
            cancelButtonText: 'No, cancel!',
            reverseButtons: true
        }).then((result) => {
            if (result.value) {
                if (result.isConfirmed) {
```

Figure 58 : Delete Codes

Figure 14 shows a block of code written in HTML and JavaScript. It creates a table with data and a button to delete a user account. When the user clicks on the delete button, a pop-up alert is triggered asking the user to confirm the deletion. The pop-up shows the name of the user account and a warning that the action cannot be reverted. The user can either click on the "Yes, delete it!" button to proceed with the deletion or "No, cancel!" to cancel the action. If the user clicks "Yes, delete it!", an AJAX request is sent to the server to delete the user account. If the deletion is successful, the table row containing the deleted user account is removed from the table.

4.2.1.4 My Profile



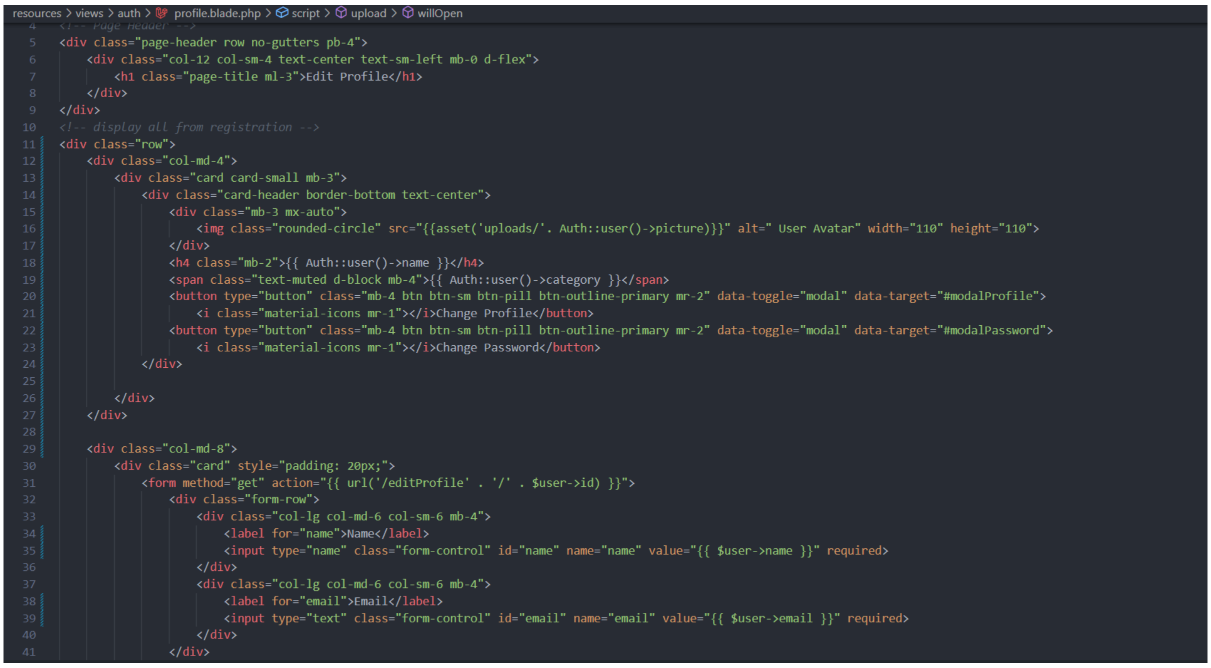
Edit Profile

AliaHidayah1
Human Resource

Name: AliaHidayah1
Email: Alia@gmail.com
Employee ID: CB20153
Phone Number: 0123912221
Address: Kb
User Type: Human Resource

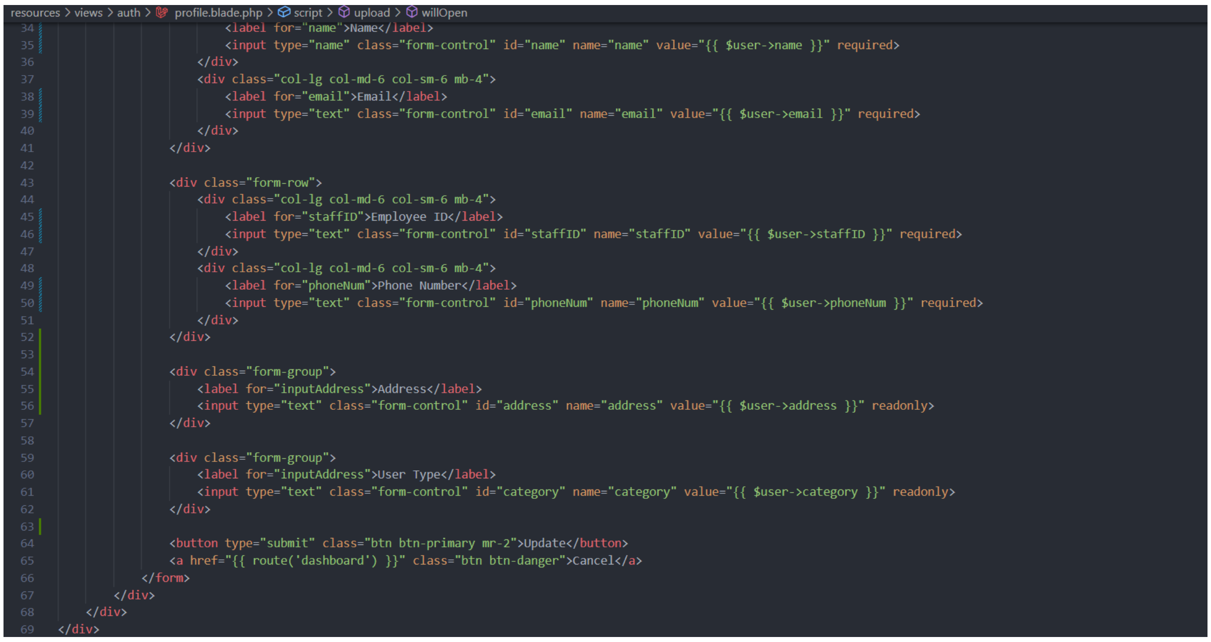
Figure 59 : My Profile Page

Figure 15 shows the "My Profile" interface displays the user's personal information such as name, email, employee ID, phone number, address, and position. The user can easily view and update their information as needed. Additionally, the interface includes options for changing the user's profile picture and password. By clicking on the "Change Profile Picture" button, the user can upload a new profile picture to replace the current one. Similarly, the "Change Password" button allows the user to update their login credentials to ensure the security of their account. These features provide a convenient and user-friendly way for users to manage their personal information and security settings within the application.



```
resources > views > auth > profile.blade.php > script > upload > willOpen
<!-- Page Header -->
<div class="page-header row no-gutters pb-4">
    <div class="col-12 col-sm-4 text-center text-sm-left mb-0 d-flex">
        <h1 class="page-title ml-3">Edit Profile</h1>
    </div>
</div>
<!-- display all from registration -->
<div class="row">
    <div class="col-md-4">
        <div class="card card-small mb-3">
            <div class="card-header border-bottom text-center">
                <div class="mb-3 mx-auto">
                    <img class="rounded-circle" src="{{asset('uploads/'. Auth::user()->picture)}}" alt=" User Avatar" width="110" height="110">
                </div>
                <h4 class="mb-2">{{ Auth::user()->name }}</h4>
                <span class="text-muted d-block mb-4">{{ Auth::user()->category }}</span>
                <button type="button" class="mb-4 btn btn-sm btn-pill btn-outline-primary mr-2" data-toggle="modal" data-target="#modalProfile">
                    <i class="material-icons mr-1"></i>Change Profile</button>
                <button type="button" class="mb-4 btn btn-sm btn-pill btn-outline-primary mr-2" data-toggle="modal" data-target="#modalPassword">
                    <i class="material-icons mr-1"></i>Change Password</button>
            </div>

        </div>
    </div>

    <div class="col-md-8">
        <div class="card" style="padding: 20px;">
            <form method="get" action="{{ url('/editProfile' . '/' . $user->id) }}">
                <div class="form-row">
                    <div class="col-lg col-md-6 col-sm-6 mb-4">
                        <label for="name">Name</label>
                        <input type="name" class="form-control" id="name" name="name" value="{{ $user->name }}" required>
                    </div>
                    <div class="col-lg col-md-6 col-sm-6 mb-4">
                        <label for="email">Email</label>
                        <input type="text" class="form-control" id="email" name="email" value="{{ $user->email }}" required>
                    </div>
                </div>
```

Figure 60 : My Profile Codes

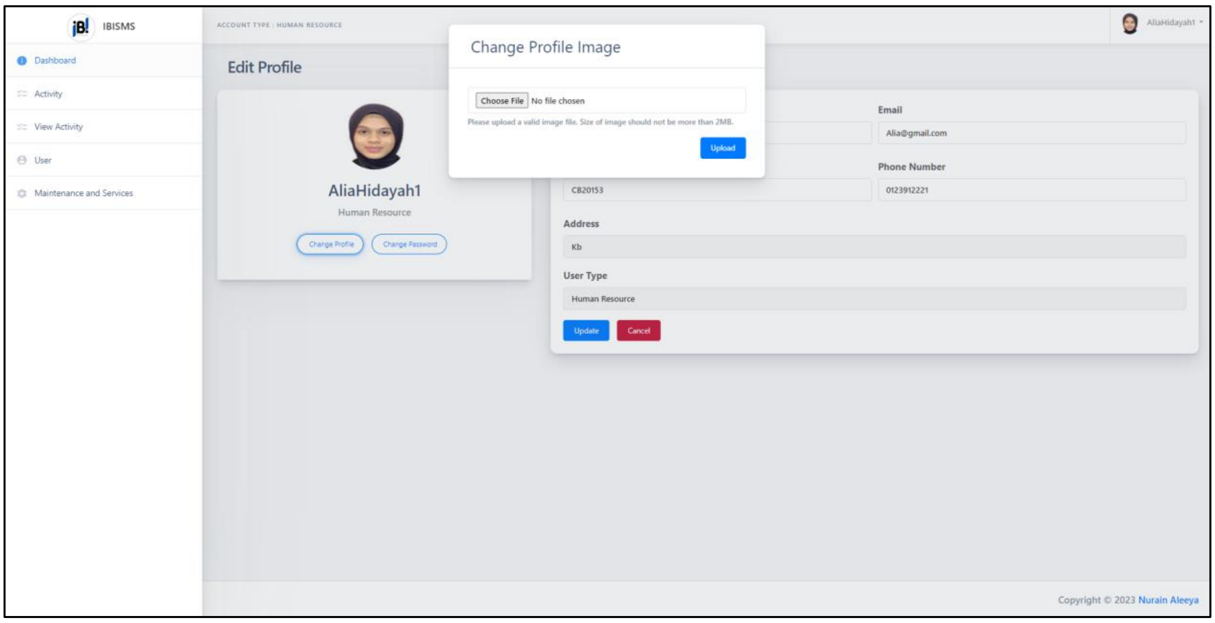


```
resources > views > auth > profile.blade.php > script > upload > willOpen
                        <label for="name">Name</label>
                        <input type="name" class="form-control" id="name" name="name" value="{{ $user->name }}" required>
                    </div>
                    <div class="col-lg col-md-6 col-sm-6 mb-4">
                        <label for="email">Email</label>
                        <input type="text" class="form-control" id="email" name="email" value="{{ $user->email }}" required>
                    </div>
                </div>

                <div class="form-row">
                    <div class="col-lg col-md-6 col-sm-6 mb-4">
                        <label for="staffID">Employee ID</label>
                        <input type="text" class="form-control" id="staffID" name="staffID" value="{{ $user->staffID }}" required>
                    </div>
                    <div class="col-lg col-md-6 col-sm-6 mb-4">
                        <label for="phoneNum">Phone Number</label>
                        <input type="text" class="form-control" id="phoneNum" name="phoneNum" value="{{ $user->phoneNum }}" required>
                    </div>
                </div>

                <div class="form-group">
                    <label for="inputAddress">Address</label>
                    <input type="text" class="form-control" id="address" name="address" value="{{ $user->address }}" readonly>
                </div>

                <div class="form-group">
                    <label for="inputAddress">User Type</label>
                    <input type="text" class="form-control" id="category" name="category" value="{{ $user->category }}" readonly>
                </div>

                <button type="submit" class="btn btn-primary mr-2">Update</button>
                <a href="{{ route('dashboard') }}" class="btn btn-danger">Cancel</a>
            </form>
        </div>
    </div>
</div>
```

Figure 61 : My Profile Codes

The code provided shows a form for editing user profile information. The form includes input fields for the user's name, email, employee ID, phone number, and read-only fields for the user's address and user type. The form includes a submit button to update the user's information and a cancel button to return to the dashboard without making changes. The form uses the GET method to send the updated information to the server, which will handle the update logic. The form does not include fields for changing the user's profile picture or password.

4.2.1.4.1 My Profile (Change Profile Picture)



AliaHidayah1
Human Resource

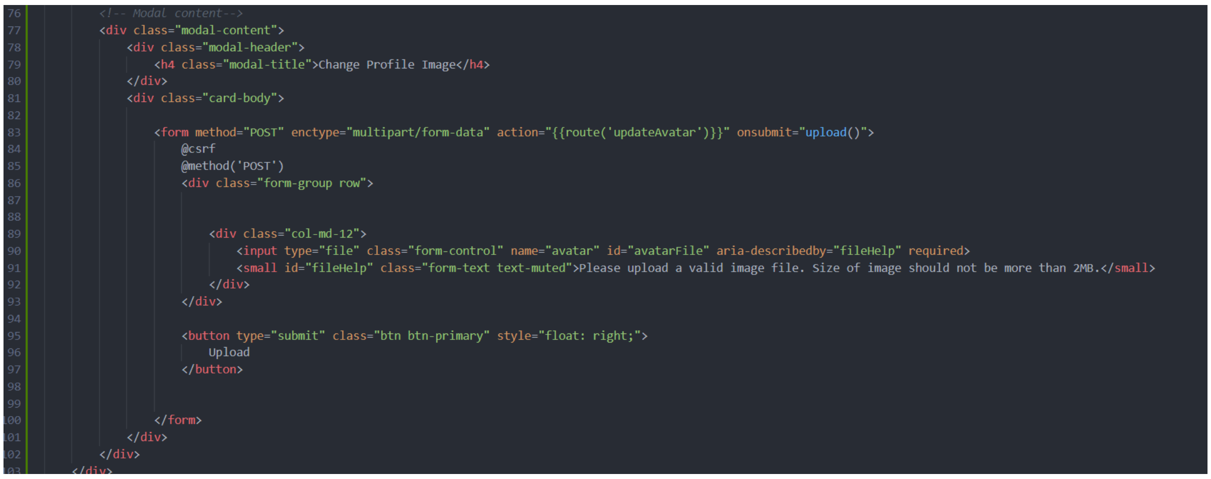
No file chosen
Please upload a valid image file. Size of image should not be more than 2MB.

Email
Alia@gmail.com
Phone Number
0123912221
CB20153
Address
Kb
User Type
Human Resource

Copyright © 2023 Nurain Aleeya

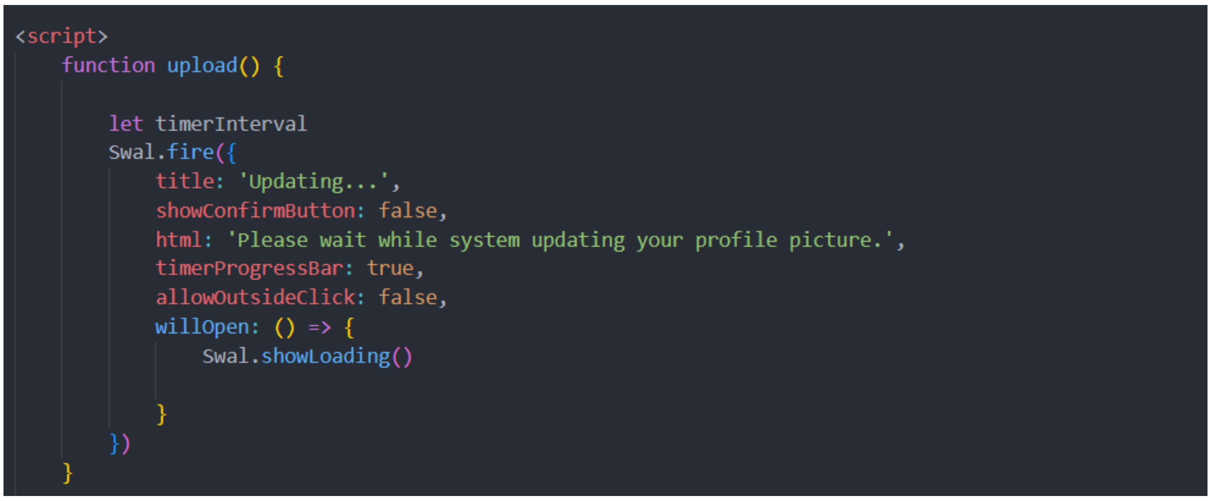
Figure 62 : Modal Popup Change Profile Picture

Figure 18 shows the interface displays a modal window that allows the user to change their profile picture. The modal contains a header with a title "Change Profile Picture" and a close button. The body of the modal displays the current profile picture along with a "Choose File" button, allowing the user to select a new image from their device. Once an image is selected, a preview of the image is shown, and the user can click the "Upload" button to upload the new profile picture.



```
        <!-- Modal content-->
        <div class="modal-content">
            <div class="modal-header">
                <h4 class="modal-title">Change Profile Image</h4>
            </div>
            <div class="card-body">

                <form method="POST" enctype="multipart/form-data" action="{{route('updateAvatar')}}" onsubmit="upload()">
                    @csrf
                    @method('POST')
                    <div class="form-group row">


                        <div class="col-md-12">
                            <input type="file" class="form-control" name="avatar" id="avatarFile" aria-describedby="fileHelp" required>
                            <small id="fileHelp" class="form-text text-muted">Please upload a valid image file. Size of image should not be more than 2MB.</small>
                        </div>
                    </div>

                    <button type="submit" class="btn btn-primary" style="float: right;">
                        Upload
                    </button>


                </form>
            </div>
        </div>
    </div>
```

Figure 63 : Modal Profile Picture Codes

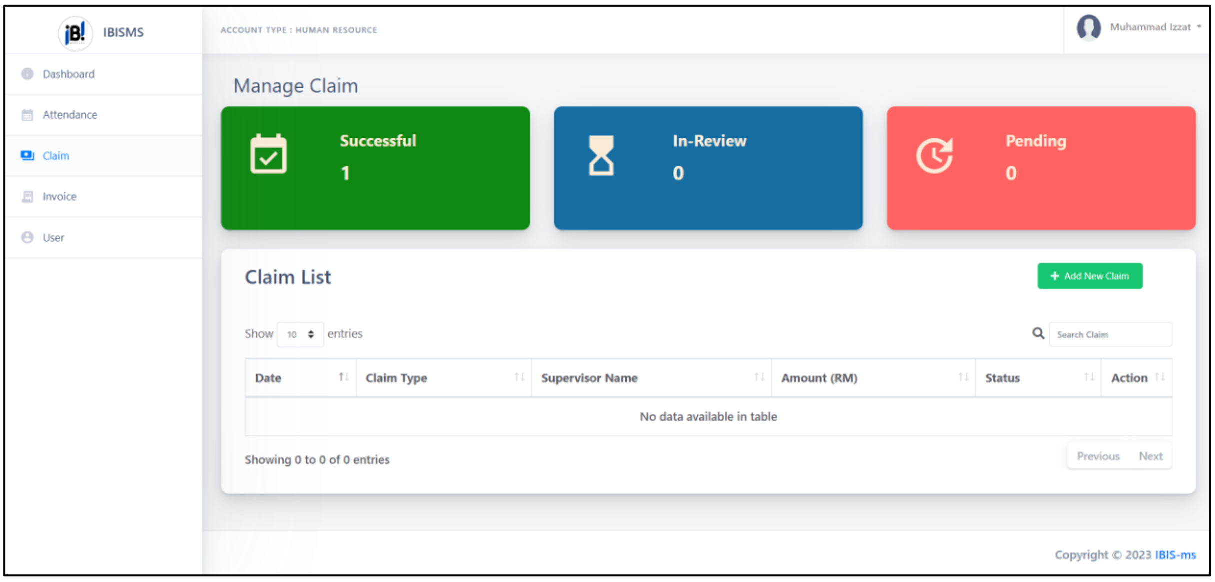


```
<script>
    function upload() {

        let timerInterval
        Swal.fire({
            title: 'Updating...',
            showConfirmButton: false,
            html: 'Please wait while system updating your profile picture.',
            timerProgressBar: true,
            allowOutsideClick: false,
            willOpen: () => {
                Swal.showLoading()

            }
        })
    }
```

Figure 64 : Modal Profile Picture Codes

Figure 19 shows code is for a modal that allows users to change their profile picture. It contains a header with the title "Change Profile Image" and a form with an input field for selecting a file to upload. The form has an attribute "enctype" set to "multipart/form-data" to allow file uploads. It also includes a CSRF token and a method attribute set to "POST". The user can select a file from their local system and then click on the "Upload" button to upload the file. There is also a small text below the input field providing information about the file size limit. Besides, figure 20 shows This is a JavaScript function that creates a loading animation using SweetAlert2 library when the user submits the form to change their profile picture. The function is triggered by the form's onsubmit event and displays a modal dialog with the title "Updating..." and a message "Please wait while system updating your profile picture." The dialog has a progress bar that indicates the upload progress. It also disables the user from clicking outside of the dialog by setting "allowOutsideClick" to false.

4.2.1.5 Manage Claim



Manage Claim

Successful 1

In-Review 0

Pending 0

Claim List

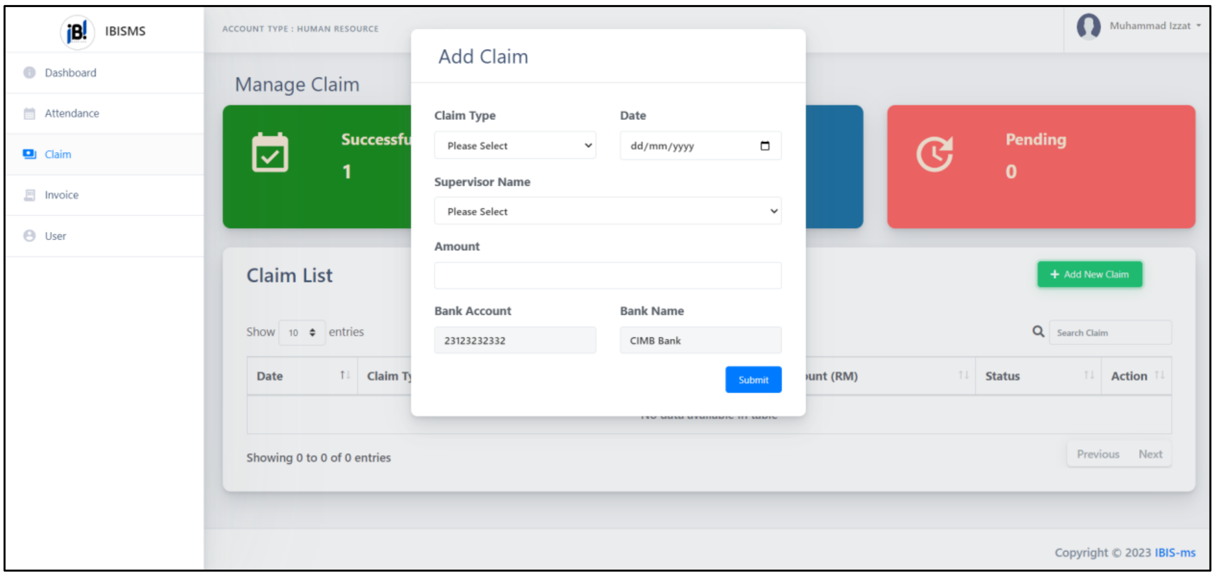
Show 10 entries

Date	Claim Type	Supervisor Name	Amount (RM)	Status	Action
No data available in table					

Showing 0 to 0 of 0 entries

Copyright © 2023 IBIS-ms

Figure 65 : Manage Claim Interface



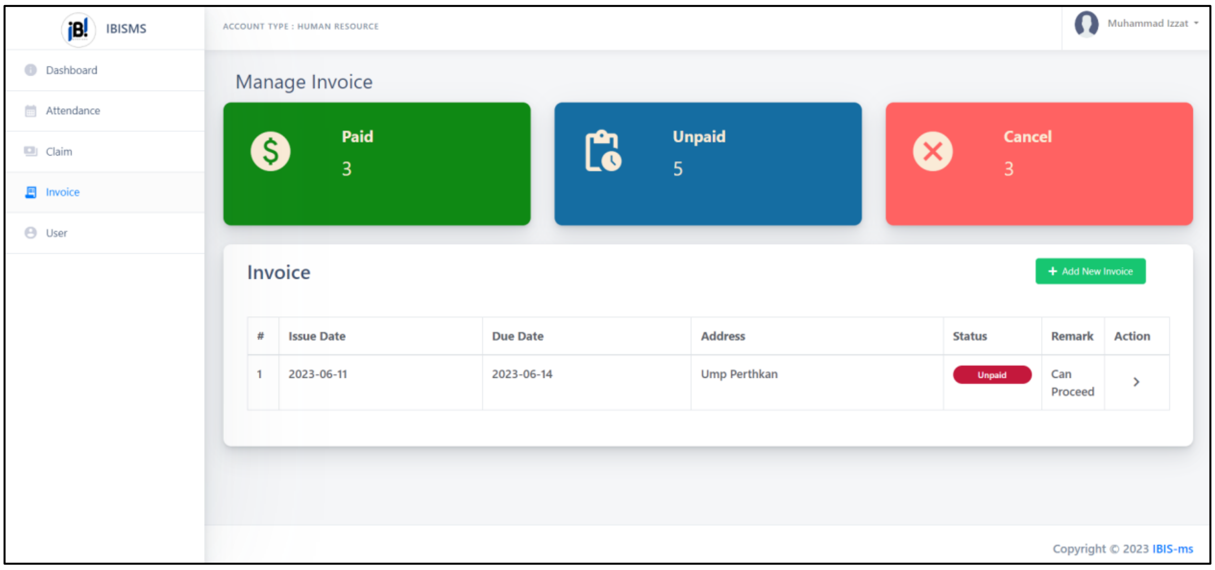
Add Claim

Claim Type: Please Select	Date: dd/mm/yyyy
Supervisor Name: Please Select	
Amount:	
Bank Account: 23123232332	Bank Name: CIMB Bank

Figure 66 : Add New Claim Interface

The Manage Claim interface in the IBIS Company provides users with a platform to submit claims for Overtime, Fuel, and Medical expenses. Figure 64 displays the claim dashboard, where users can track the status of their claims, whether they are approved, in review, or pending. To submit a claim, users are required to enter specific information in the claim form. This includes selecting the claim type (Overtime, Fuel, or Medical), entering the date of the expense, providing the supervisor's name for approval, and specifying the amount being claimed. Once all the necessary information is entered, users can click the submit button to finalize their claim submission.

4.2.1.6 Manage Invoice



Manage Invoice

Paid 3

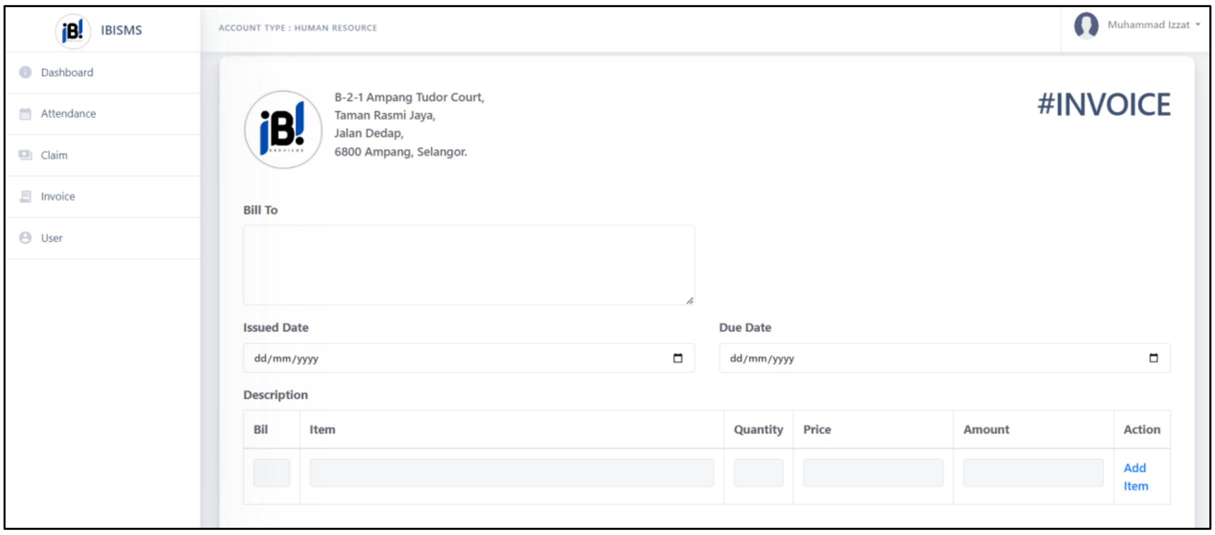
Unpaid 5

Cancel 3

Invoice

#	Issue Date	Due Date	Address	Status	Remark	Action
1	2023-06-11	2023-06-14	Ump Perthkan	Unpaid	Can Proceed	›

Figure 67 : Manage Invoice Interface



B-2-1 Ampang Tudor Court,
Taman Rasmi Jaya,
Jalan Dedap,
6800 Ampang, Selangor.

#INVOICE

Bill To

Issued Date: dd/mm/yyyy

Due Date: dd/mm/yyyy

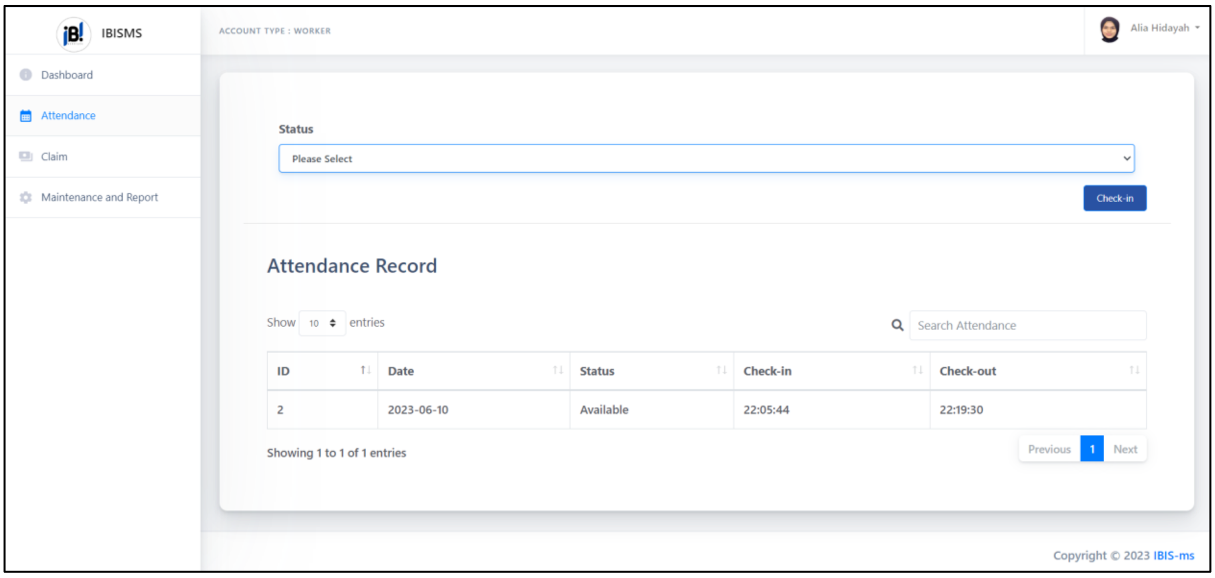
Description

Bil	Item	Quantity	Price	Amount	Action
					Add Item

Figure 68 : Add New Invoice Interface

The Manage Invoice interface displayed in the figure above provides a comprehensive view of the submitted invoices. The interface consists of a list of invoices that have been submitted and a dedicated form for creating new invoices. Accountants can utilize the form to generate invoices by entering essential details such as the company name, address, date, due date, and itemized information. Upon entering this information, the system will generate the invoice in PDF format, making it easily accessible and printable

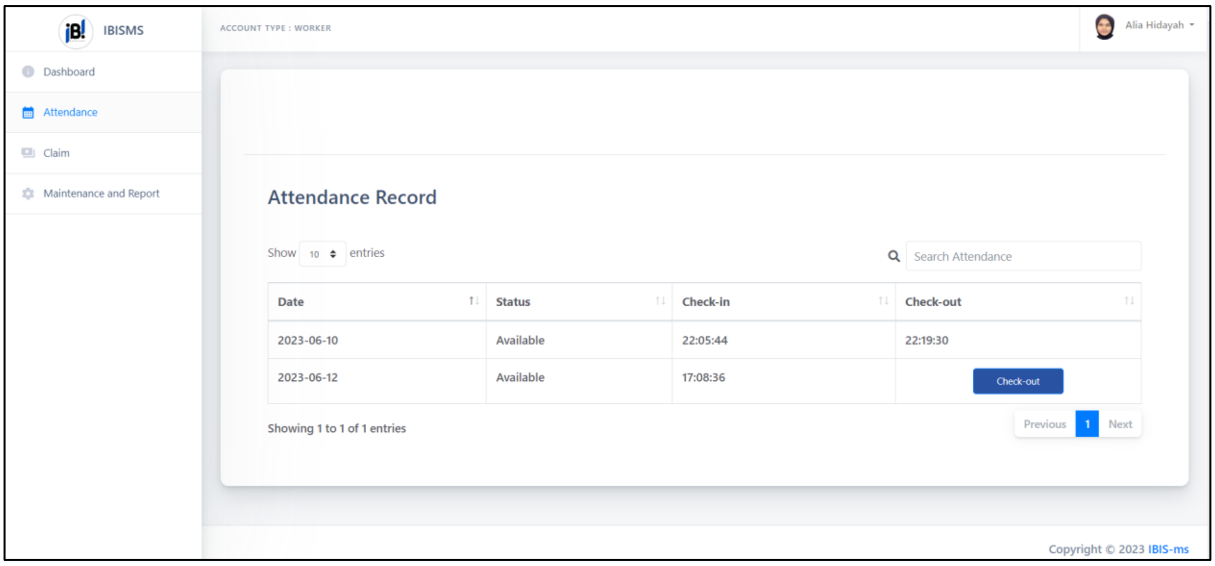
4.2.1.6 Manage Attendance



ID	Date	Status	Check-in	Check-out
2	2023-06-10	Available	22:05:44	22:19:30

Showing 1 to 1 of 1 entries

Figure 69 : Attendance Interface



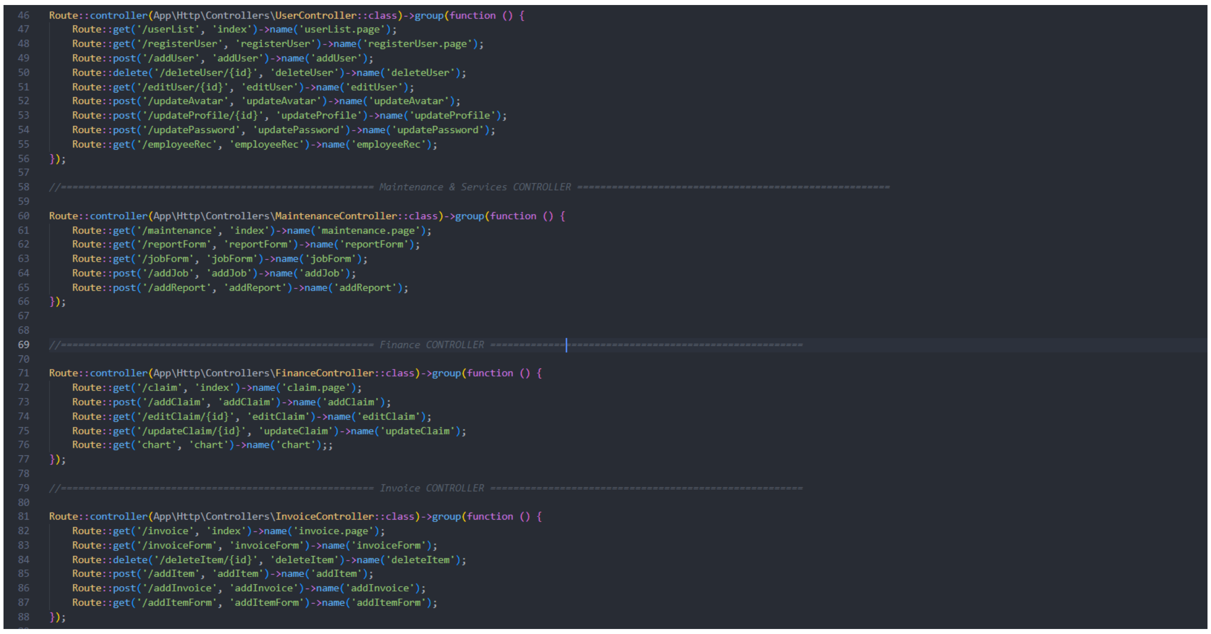
Date	Status	Check-in	Check-out
2023-06-10	Available	22:05:44	22:19:30
2023-06-12	Available	17:08:36	Check-out

Showing 1 to 1 of 1 entries

Figure 70 : Check Out Interface

The Manage Attendance module in IBIS-ms is designed for use by all actors within the system. Users have the ability to record their attendance by selecting their status as either "On Site" or "Available" and clicking the "Check-In" button. The system will automatically update the attendance record with the corresponding date and time. When it is time to check out, users can simply click the "Check-Out" button. The system will record the date and time of the check-out, and this information will be displayed in the attendance record table.

4.2.1.7 Web.php

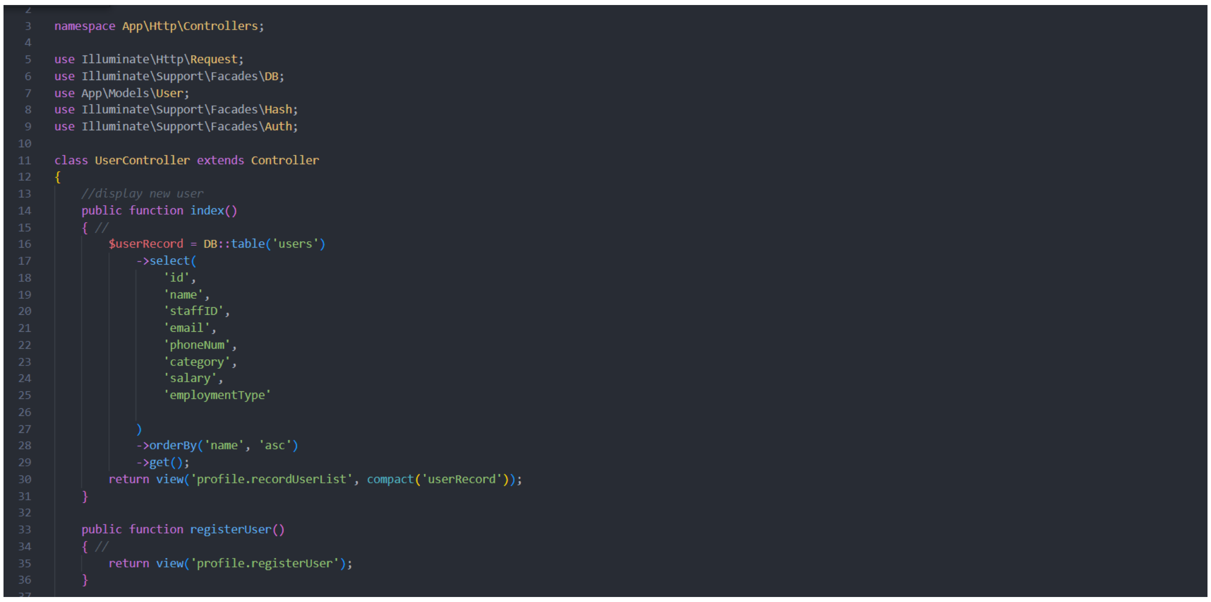


```php
Route::controller(App\Http\Controllers\UserController::class)->group(function () {
    Route::get('/userList', 'index')->name('userList.page');
    Route::get('/registerUser', 'registerUser')->name('registerUser.page');
    Route::post('/addUser', 'addUser')->name('addUser');
    Route::delete('/deleteUser/{id}', 'deleteUser')->name('deleteUser');
    Route::get('/editUser/{id}', 'editUser')->name('editUser');
    Route::post('/updateAvatar', 'updateAvatar')->name('updateAvatar');
    Route::post('/updateProfile/{id}', 'updateProfile')->name('updateProfile');
    Route::post('/updatePassword', 'updatePassword')->name('updatePassword');
    Route::get('/employeeRec', 'employeeRec')->name('employeeRec');
});

//====================================================== Maintenance & Services CONTROLLER ======================================================

Route::controller(App\Http\Controllers\MaintenanceController::class)->group(function () {
    Route::get('/maintenance', 'index')->name('maintenance.page');
    Route::get('/reportForm', 'reportForm')->name('reportForm');
    Route::get('/jobForm', 'jobForm')->name('jobForm');
    Route::post('/addJob', 'addJob')->name('addJob');
    Route::post('/addReport', 'addReport')->name('addReport');
});


//====================================================== Finance CONTROLLER ======================================================

Route::controller(App\Http\Controllers\FinanceController::class)->group(function () {
    Route::get('/claim', 'index')->name('claim.page');
    Route::post('/addClaim', 'addClaim')->name('addClaim');
    Route::get('/editClaim/{id}', 'editClaim')->name('editClaim');
    Route::get('/updateClaim/{id}', 'updateClaim')->name('updateClaim');
    Route::get('chart', 'chart')->name('chart');;
});

//====================================================== Invoice CONTROLLER ======================================================

Route::controller(App\Http\Controllers\InvoiceController::class)->group(function () {
    Route::get('/invoice', 'index')->name('invoice.page');
    Route::get('/invoiceForm', 'invoiceForm')->name('invoiceForm');
    Route::delete('/deleteItem/{id}', 'deleteItem')->name('deleteItem');
    Route::post('/addItem', 'addItem')->name('addItem');
    Route::post('/addInvoice', 'addInvoice')->name('addInvoice');
    Route::get('/addItemForm', 'addItemForm')->name('addItemForm');
});
```

Figure 71 : Web Codes

In Laravel, web.php is a file located in the routes directory that handles HTTP requests from web browsers and returns views or performs actions. The file contains a Route facade that allows defining routes using HTTP verbs and a URL pattern that maps to specific controller methods. Web routes can include parameters, middleware, and named routes. Middleware adds authentication, authorization, or other functionality to routes. Named routes provide a convenient way to generate URLs for a given route by referencing its name instead of its URL pattern.

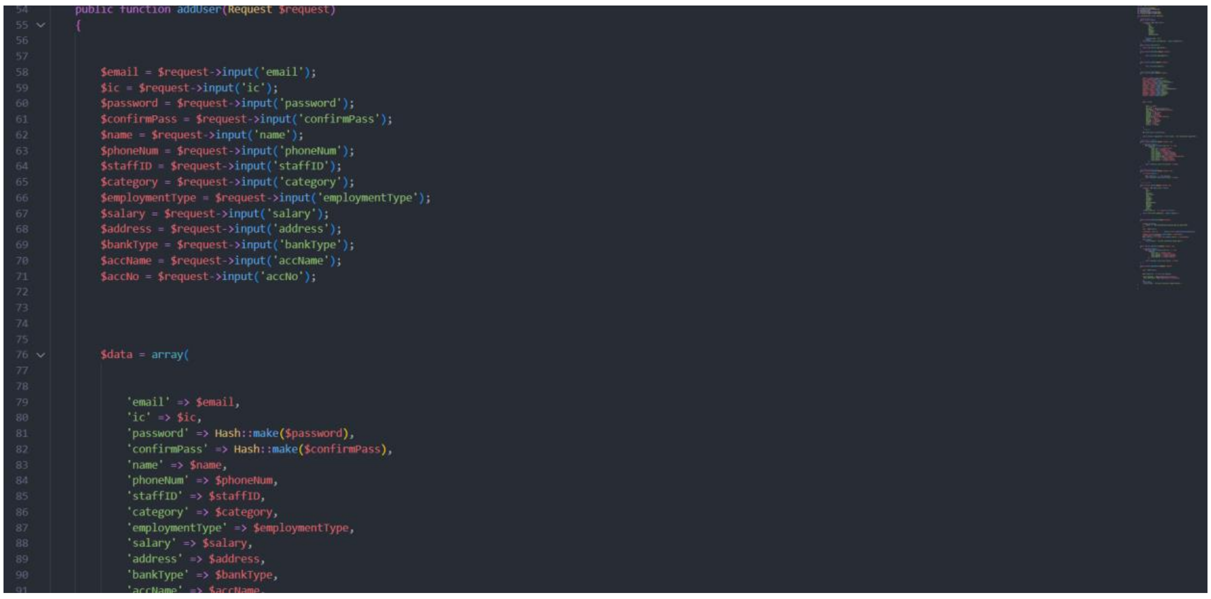
4.2.1.8 UserController.php



```php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use App\Models\User;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Auth;

class UserController extends Controller
{
    //display new user
    public function index()
    { //
        $userRecord = DB::table('users')
            ->select(
                'id',
                'name',
                'staffID',
                'email',
                'phoneNum',
                'category',
                'salary',
                'employmentType'

            )
            ->orderBy('name', 'asc')
            ->get();
        return view('profile.recordUserList', compact('userRecord'));
    }

    public function registerUser()
    { //
        return view('profile.registerUser');
    }
```

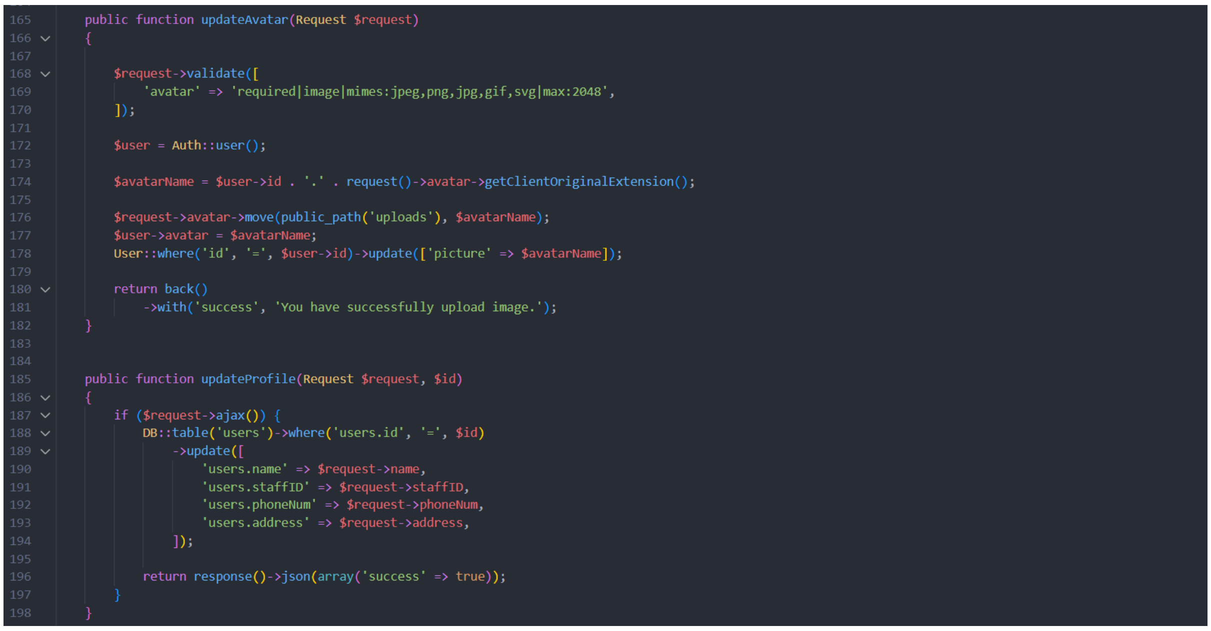
Figure 72 : User Controller Codes

The figure above displays the controller for the "index" function, which handles retrieving data from the database and displaying it on the user's page.The "index" controller is responsible for retrieving relevant data from the database, typically based on specific criteria or filters. This data could include information from various tables, such as attendance records, claims, invoices, or other relevant data



```php
    public function addUser(Request $request)
    {


        $email = $request->input('email');
        $ic = $request->input('ic');
        $password = $request->input('password');
        $confirmPass = $request->input('confirmPass');
        $name = $request->input('name');
        $phoneNum = $request->input('phoneNum');
        $staffID = $request->input('staffID');
        $category = $request->input('category');
        $employmentType = $request->input('employmentType');
        $salary = $request->input('salary');
        $address = $request->input('address');
        $bankType = $request->input('bankType');
        $accName = $request->input('accName');
        $accNo = $request->input('accNo');




        $data = array(


            'email' => $email,
            'ic' => $ic,
            'password' => Hash::make($password),
            'confirmPass' => Hash::make($confirmPass),
            'name' => $name,
            'phoneNum' => $phoneNum,
            'staffID' => $staffID,
            'category' => $category,
            'employmentType' => $employmentType,
            'salary' => $salary,
            'address' => $address,
            'bankType' => $bankType,
            'accName' => $accName,
```

Figure 73 : User Controller Codes



```php
    public function updateAvatar(Request $request)
    {

        $request->validate([
            'avatar' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
        ]);

        $user = Auth::user();

        $avatarName = $user->id . '.' . request()->avatar->getClientOriginalExtension();

        $request->avatar->move(public_path('uploads'), $avatarName);
        $user->avatar = $avatarName;
        User::where('id', '=', $user->id)->update(['picture' => $avatarName]);

        return back()
            ->with('success', 'You have successfully upload image.');
    }


    public function updateProfile(Request $request, $id)
    {
        if ($request->ajax()) {
            DB::table('users')->where('users.id', '=', $id)
                ->update([
                    'users.name' => $request->name,
                    'users.staffID' => $request->staffID,
                    'users.phoneNum' => $request->phoneNum,
                    'users.address' => $request->address,
                ]);

            return response()->json(array('success' => true));
        }
    }
```

Figure 74 : User Controller Codes

In Laravel, a controller is a PHP file located in the app/Http/Controllers directory that handles user requests and defines application logic. A controller typically contains a set of methods, each of which corresponds to a specific route defined in the web.php file. When a user makes a request to a particular route, the corresponding method in the controller is executed. Controllers can interact with models to retrieve and manipulate data from a database, and they can return responses in various formats, such as views, JSON, or file downloads. Controllers can also make use of middleware to perform tasks such as authentication, authorization, or input validation.

4.2.2 Database Design

This section concentrates on the testing of the IBIS system using the MySQL database. The database is called "ibis_ms" and is responsible for storing all the data entered by users. In order to capture and track all user activities, different tables have been designed within the database. These tables are specifically created to store all the relevant information pertaining to the system, ensuring that all user data is accurately recorded and available.

4.2.2.1 Database for IBIS Management System

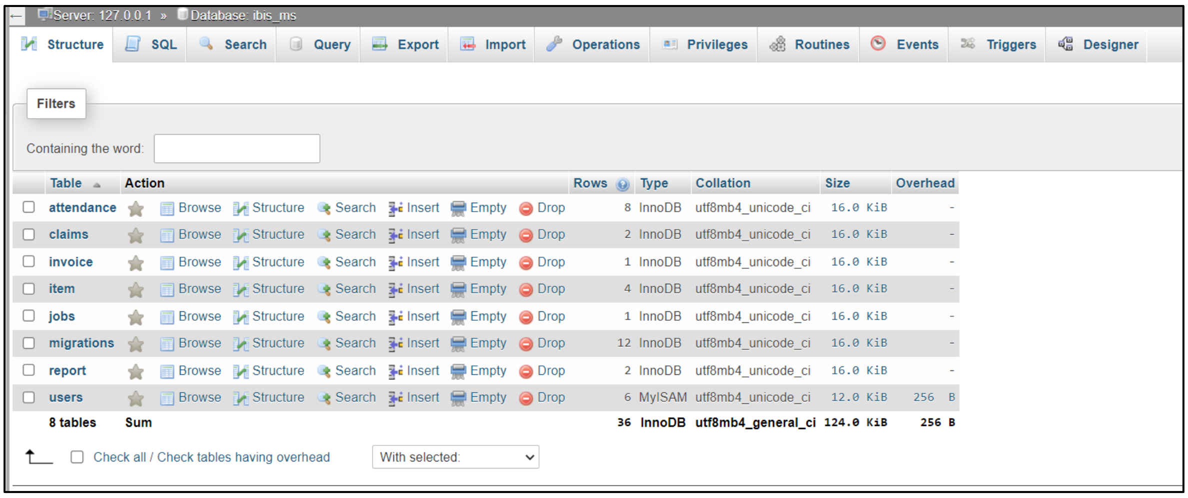
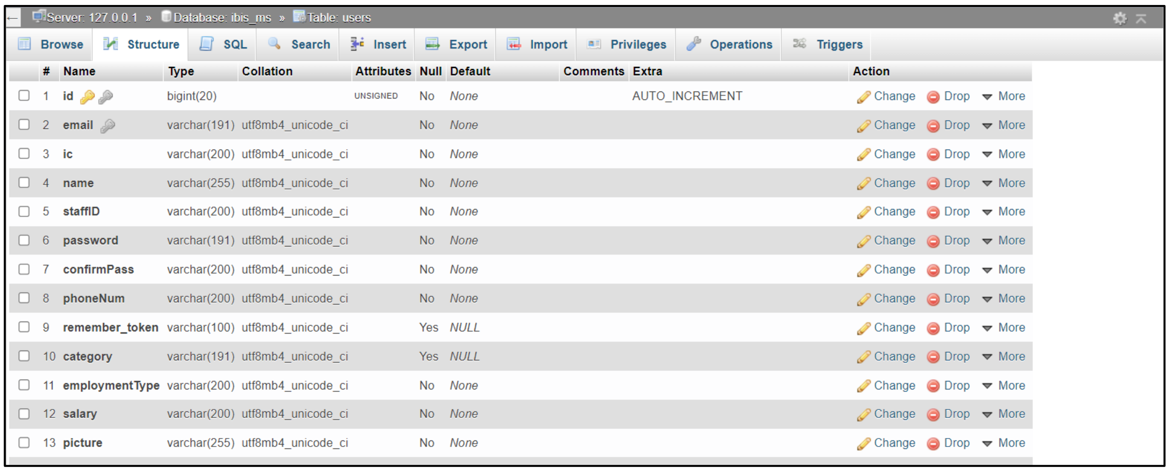


Table	Action							Rows	Type	Collation	Size	Overhead
attendance	☆	Browse	Structure	Search	Insert	Empty	Drop	8	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
claims	☆	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
invoice	☆	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
item	☆	Browse	Structure	Search	Insert	Empty	Drop	4	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
jobs	☆	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
migrations	☆	Browse	Structure	Search	Insert	Empty	Drop	12	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
report	☆	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
users	☆	Browse	Structure	Search	Insert	Empty	Drop	6	MyISAM	utf8mb4_unicode_ci	12.0 KiB	256 B
8 tables	Sum							36	InnoDB	utf8mb4_general_ci	124.0 KiB	256 B

Figure 75 : Database ibis_ms

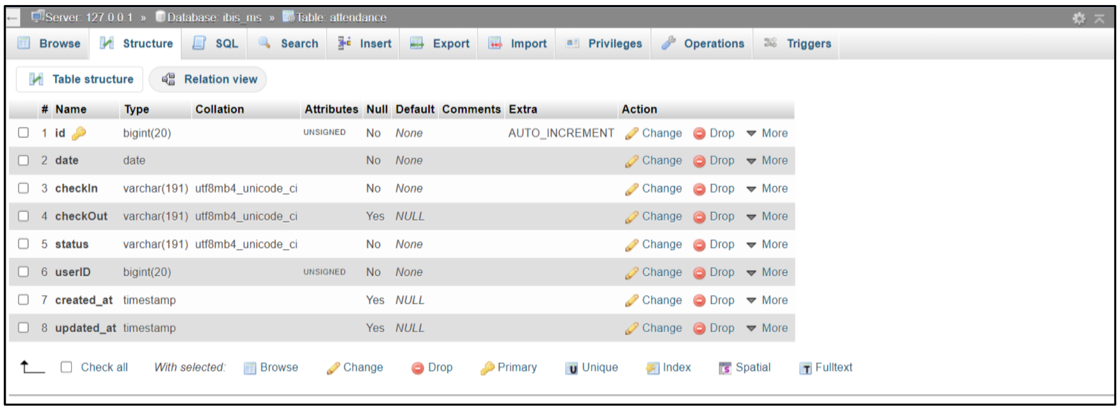
Figure above shows the database for IBIS Management System that named as ibis_ms. Its content 8 tables such as attendance, claims, invoice, item, jobs, migrations, report and users.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	id	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change	Drop	More
2	email	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change	Drop	More
3	ic	varchar(200)	utf8mb4_unicode_ci		No	*None*			Change	Drop	More
4	name	varchar(255)	utf8mb4_unicode_ci		No	*None*			Change	Drop	More
5	staffID	varchar(200)	utf8mb4_unicode_ci		No	*None*			Change	Drop	More
6	password	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change	Drop	More
7	confirmPass	varchar(200)	utf8mb4_unicode_ci		No	*None*			Change	Drop	More
8	phoneNum	varchar(200)	utf8mb4_unicode_ci		No	*None*			Change	Drop	More
9	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	*NULL*			Change	Drop	More
10	category	varchar(191)	utf8mb4_unicode_ci		Yes	*NULL*			Change	Drop	More
11	employmentType	varchar(200)	utf8mb4_unicode_ci		No	*None*			Change	Drop	More
12	salary	varchar(200)	utf8mb4_unicode_ci		No	*None*			Change	Drop	More
13	picture	varchar(255)	utf8mb4_unicode_ci		No	*None*			Change	Drop	More

Figure 76 : Table Users

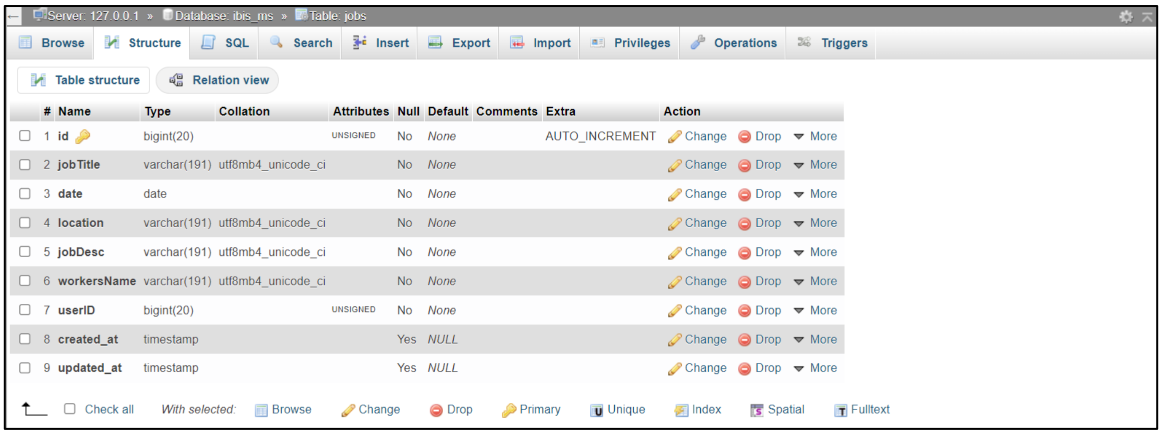
The figure above shows the table of the users. It consists of 13 attributes for manage user information.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
2	date	date			No	*None*			Change Drop More
3	checkIn	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
4	checkOut	varchar(191)	utf8mb4_unicode_ci		Yes	*NULL*			Change Drop More
5	status	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
6	userID	bigint(20)		UNSIGNED	No	*None*			Change Drop More
7	created_at	timestamp			Yes	*NULL*			Change Drop More
8	updated_at	timestamp			Yes	*NULL*			Change Drop More

Figure 77: Table invoice

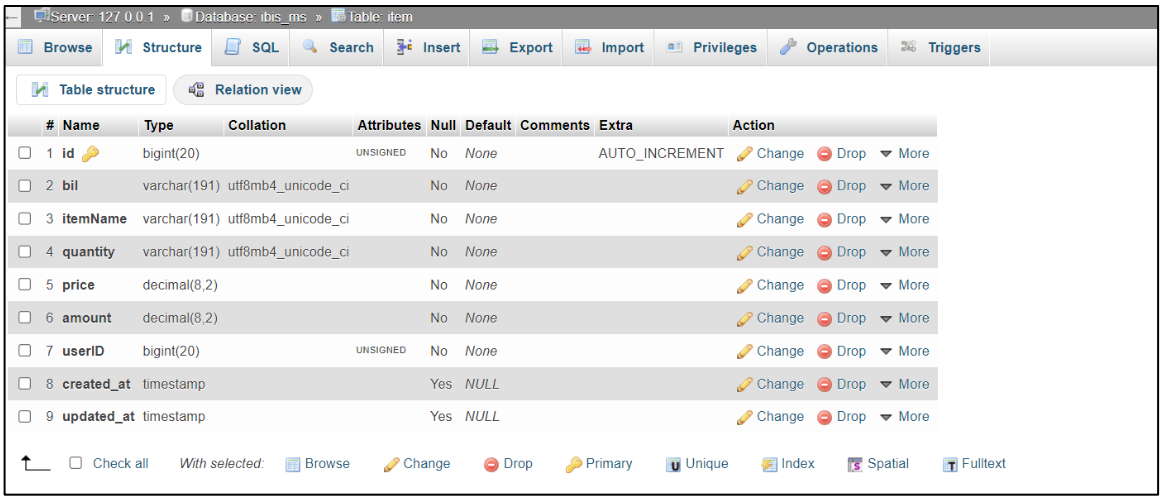
The figure above shows the table of the invoice. It consists of 8 attributes for manage invoice information.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
2	jobTitle	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
3	date	date			No	*None*			Change Drop More
4	location	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
5	jobDesc	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
6	workersName	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
7	userID	bigint(20)		UNSIGNED	No	*None*			Change Drop More
8	created_at	timestamp			Yes	*NULL*			Change Drop More
9	updated_at	timestamp			Yes	*NULL*			Change Drop More

Figure 78 : Tables Jobs

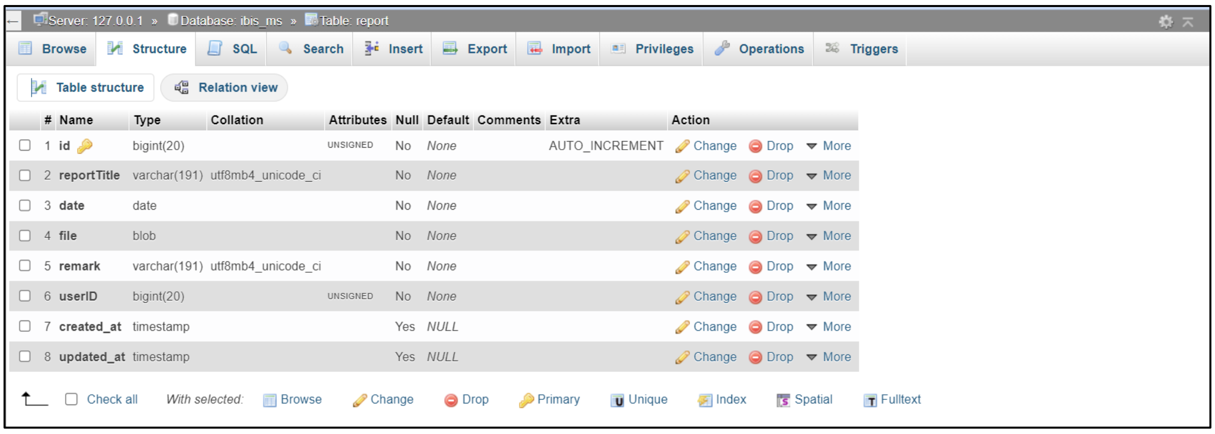
The figure above shows the table of the Jobs. It consists of 9 attributes for manage jobs.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop More
2	bil	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
3	itemName	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
4	quantity	varchar(191)	utf8mb4_unicode_ci		No	*None*			Change Drop More
5	price	decimal(8,2)			No	*None*			Change Drop More
6	amount	decimal(8,2)			No	*None*			Change Drop More
7	userID	bigint(20)		UNSIGNED	No	*None*			Change Drop More
8	created_at	timestamp			Yes	*NULL*			Change Drop More
9	updated_at	timestamp			Yes	*NULL*			Change Drop More

Figure 79 : Table item

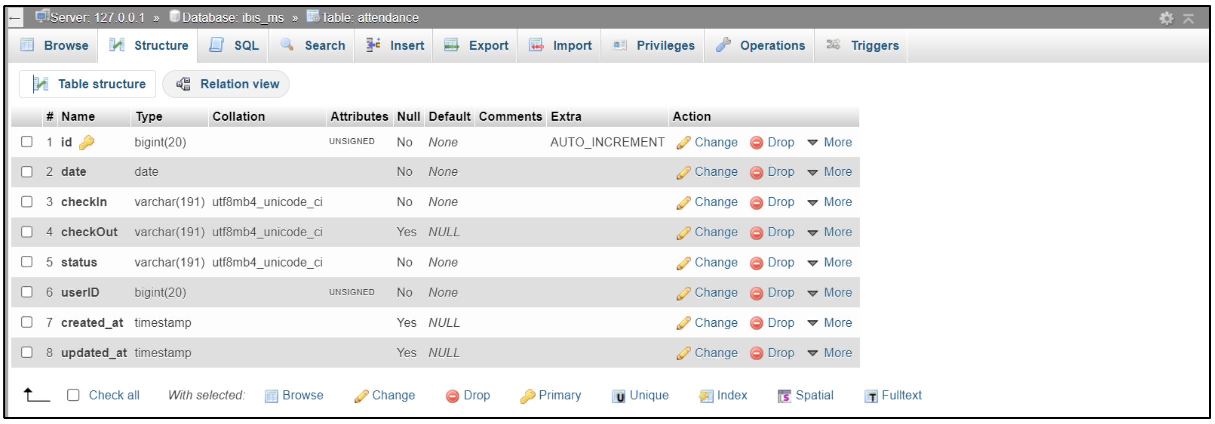
The figure above shows the table of the items. It consists of 9 attributes for manage item for the invoice purpose.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	reportTitle	varchar(191)	utf8mb4_unicode_ci		No	None			Change Drop More
3	date	date			No	None			Change Drop More
4	file	blob			No	None			Change Drop More
5	remark	varchar(191)	utf8mb4_unicode_ci		No	None			Change Drop More
6	userID	bigint(20)		UNSIGNED	No	None			Change Drop More
7	created_at	timestamp			Yes	NULL			Change Drop More
8	updated_at	timestamp			Yes	NULL			Change Drop More

Figure 80 : Table Report

The figure above shows the table of the Report. It consists of 8 attributes for manage report for. It can store the data that has been submitted by the worker to the supervisor.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	date	date			No	None			Change Drop More
3	checkIn	varchar(191)	utf8mb4_unicode_ci		No	None			Change Drop More
4	checkOut	varchar(191)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
5	status	varchar(191)	utf8mb4_unicode_ci		No	None			Change Drop More
6	userID	bigint(20)		UNSIGNED	No	None			Change Drop More
7	created_at	timestamp			Yes	NULL			Change Drop More
8	updated_at	timestamp			Yes	NULL			Change Drop More

Figure 81 : Attendance table

The figure above shows the table of the attendance. It consists of 8 attributes for manage attendance to record all the attendance data.

4.3 Testing and Result

This section explains how the system is tested to find any problems or areas that need to be improved so that it operates properly and satisfies all intended requirements. The UAT form, which will be broken down into various modules for testing, will be used by testers. Testers will offer feedback on the system's performance and any problems they run into while using it based on the user acceptance test. They will also make recommendations for enhancing the system's effectiveness and efficiency for users.

No.	Module	Activities	Status		Comments
1.	Manage Registration	Navigation bar	Yes ( )	No ( )	
		User registration	Yes ( )	No ( )	
		User login	Yes ( )	No ( )	
		User logout	Yes ( )	No ( )	
		Change Password	Yes ( )	No ( )	
		Edit Profile	Yes ( )	No ( )	
		Edit User	Yes ( )	No ( )	
		Delete User	Yes ( )	No ( )	
		Change Profile Picture	Yes ( )	No ( )	
2.	Manage Claim	View dashboard	Yes ( )	No ( )	
		Add Claim button	Yes ( )	No ( )	
		Claim Form	Yes ( )	No ( )	
		Submit Form	Yes ( )	No ( )	
		Edit Form	Yes ( )	No ( )	
3.	Manage Invoice	View dashboard	Yes ( )	No ( )	
		Add New Invoice	Yes ( )	No ( )	
		View invoice status	Yes ( )	No ( )	
		Add Company	Yes ( )	No ( )	
		Edit Company	Yes ( )	No ( )	
4.	Manage Attendance	View dashboard	Yes ( )	No ( )	
		Check In Button	Yes ( )	No ( )	
		Check Out Button	Yes ( )	No ( )	
		View Report	Yes ( )	No ( )	
5.	Manage Maintenance and Report	View Dashboard	Yes ( )	No ( )	
		Add New Job	Yes ( )	No ( )	
		Delete Job	Yes ( )	No ( )	
		Add New Report	Yes ( )	No ( )	
		View Report	Yes ( )	No ( )	

This test has been performed by:

Name	: ______________________________
Signature	: ______________________________
Date	: ______________________________

No.	Module	Activities	Status		Comments
1.	Manage Registration	Navigation bar	Yes (✓)	No ( )	
		User registration	Yes (✓)	No ( )	
		User login	Yes (✓)	No ( )	
		User logout	Yes (✓)	No ( )	
		Change Password	Yes (✓)	No ( )	
		Edit Profile	Yes (✓)	No ( )	
		Edit User	Yes (✓)	No ( )	
		Delete User	Yes (✓)	No ( )	
		Change Profile Picture	Yes (✓)	No ( )	
2.	Manage Claim	View dashboard	Yes (✓)	No ( )	
		Add Claim button	Yes (✓)	No ( )	
		Claim Form	Yes (✓)	No ( )	
		Submit Form	Yes (✓)	No ( )	
		Edit Form	Yes (✓)	No ( )	
3.	Manage Invoice	View dashboard	Yes (✓)	No ( )	
		Add New Invoice	Yes (✓)	No ( )	
		View invoice status	Yes (✓)	No ( )	
		Add Company	Yes (✓)	No ( )	
		Edit Company	Yes (✓)	No ( )	
4.	Manage Attendance	View dashboard	Yes (✓)	No ( )	
		Check In Button	Yes (✓)	No ( )	
		Check Out Button	Yes (✓)	No ( )	
		View Report	Yes (✓)	No ( )	
5.	Manage Maintenance and Report	View Dashboard	Yes (✓)	No ( )	
		Add New Job	Yes (✓)	No ( )	
		Delete Job	Yes (✓)	No ( )	
		Add New Report	Yes (✓)	No ( )	
		View Report	Yes (✓)	No ( )	

This test has been performed by:

Name	: MAISARAH BT. FAISAL
Signature	: [illegible]
Date	: 18/6/2023

No.	Module	Activities	Status		Comments
1.	Manage Registration	Navigation bar	Yes (✓)	No ( )	
		User registration	Yes (✓)	No ( )	
		User login	Yes (✓)	No ( )	
		User logout	Yes (✓)	No ( )	
		Change Password	Yes (✓)	No ( )	
		Edit Profile	Yes (✓)	No ( )	
		Edit User	Yes (✓)	No ( )	
		Delete User	Yes (✓)	No ( )	
		Change Profile Picture	Yes (✓)	No ( )	
2.	Manage Claim	View dashboard	Yes (✓)	No ( )	
		Add Claim button	Yes (✓)	No ( )	
		Claim Form	Yes (✓)	No ( )	
		Submit Form	Yes (✓)	No ( )	
		Edit Form	Yes (✓)	No ( )	
3.	Manage Invoice	View dashboard	Yes (✓)	No ( )	
		Add New Invoice	Yes (✓)	No ( )	
		View invoice status	Yes (✓)	No ( )	
		Add Company	Yes (✓)	No ( )	
		Edit Company	Yes (✓)	No ( )	
4.	Manage Attendance	View dashboard	Yes (✓)	No ( )	
		Check In Button	Yes (✓)	No ( )	
		Check Out Button	Yes (✓)	No ( )	
		View Report	Yes (✓)	No ( )	
5.	Manage Maintenance and Report	View Dashboard	Yes (✓)	No ( )	
		Add New Job	Yes (✓)	No ( )	
		Delete Job	Yes (✓)	No ( )	
		Add New Report	Yes (✓)	No ( )	
		View Report	Yes (✓)	No ( )	

This test has been performed by:

Name	: NUR AWA HIDAYAH BINTI ROHAYA UDIN
Signature	: [illegible]
Date	: 18/6/2023

No.	Module	Activities	Status	Comments
1.	Manage Registration	Navigation bar	Yes (✓) No ( )	
		User registration	Yes (✓) No ( )	
		User login	Yes (✓) No ( )	
		User logout	Yes (✓) No ( )	
		Change Password	Yes (✓) No ( )	
		Edit Profile	Yes (✓) No ( )	
		Edit User	Yes (✓) No ( )	
		Delete User	Yes (✓) No ( )	
		Change Profile Picture	Yes (✓) No ( )	
2.	Manage Claim	View dashboard	Yes (✓) No ( )	
		Add Claim button	Yes (✓) No ( )	
		Claim Form	Yes (✓) No ( )	
		Submit Form	Yes (✓) No ( )	
		Edit Form	Yes (✓) No ( )	
3.	Manage Invoice	View dashboard	Yes (✓) No ( )	
		Add New Invoice	Yes (✓) No ( )	
		View invoice status	Yes (✓) No ( )	
		Add Company	Yes (✓) No ( )	
		Edit Company	Yes (✓) No ( )	
4.	Manage Attendance	View dashboard	Yes (✓) No ( )	
		Check In Button	Yes (✓) No ( )	
		Check Out Button	Yes (✓) No ( )	
		View Report	Yes (✓) No ( )	
5.	Manage Maintenance and Report	View Dashboard	Yes (✓) No ( )	
		Add New Job	Yes (✓) No ( )	
		Delete Job	Yes (✓) No ( )	
		Add New Report	Yes (✓) No ( )	
		View Report	Yes (✓) No ( )	

This test has been performed by:

Name	: NUR AFIQAH BT MOHAMMED JAMIL
Signature	: [illegible]
Date	: 18 JUN 2023

No.	Module	Activities	Status		Comments
1.	Manage Registration	Navigation bar	Yes (✓)	No ( )	
		User registration	Yes (✓)	No ( )	
		User login	Yes (✓)	No ( )	
		User logout	Yes (✓)	No ( )	
		Change Password	Yes (✓)	No ( )	
		Edit Profile	Yes (✓)	No ( )	
		Edit User	Yes (✓)	No ( )	
		Delete User	Yes (✓)	No ( )	
		Change Profile Picture	Yes (✓)	No ( )	
2.	Manage Claim	View dashboard	Yes (✓)	No ( )	
		Add Claim button	Yes (✓)	No ( )	
		Claim Form	Yes (✓)	No ( )	
		Submit Form	Yes (✓)	No ( )	
		Edit Form	Yes (✓)	No ( )	
3.	Manage Invoice	View dashboard	Yes (✓)	No ( )	
		Add New Invoice	Yes (✓)	No ( )	
		View invoice status	Yes (✓)	No ( )	
		Add Company	Yes (✓)	No ( )	
		Edit Company	Yes (✓)	No ( )	
4.	Manage Attendance	View dashboard	Yes (✓)	No ( )	
		Check In Button	Yes (✓)	No ( )	
		Check Out Button	Yes (✓)	No ( )	
		View Report	Yes (✓)	No ( )	
5.	Manage Maintenance and Report	View Dashboard	Yes (✓)	No ( )	
		Add New Job	Yes (✓)	No ( )	
		Delete Job	Yes (✓)	No ( )	
		Add New Report	Yes (✓)	No ( )	
		View Report	Yes (✓)	No ( )	

This test has been performed by:

Name	: Farra Alia binti Rosli
Signature	: [illegible]
Date	: 18 JUNE 2023

CHAPTER 5
CONCLUSION

5.1 Introduction

In this chapter, the project outcome is summarized, highlighting the successful completion of the IBIS-ms system and its alignment with the initial project objectives and requirements. It emphasizes how the system meets the needs and expectations of the end users and contributes to streamlining and optimizing the management processes within the organization. The chapter also acknowledges the constraints and limitations faced during the development of the IBIS-ms system. These constraints may include resource limitations, time constraints, technological limitations, or any other factors that may have impacted the system's development or functionality. By recognizing these constraints, the project team can reflect on lessons learned and identify areas for improvement in future projects. Furthermore, the chapter explores future possibilities for the IBIS Management System. It discusses potential enhancements and expansions that could be implemented to further improve the system's functionality, usability, and effectiveness. This could involve integrating additional features, incorporating new technologies, or adapting the system to meet evolving user requirements.

5.2 Project Constraint

The constraint of time plays a crucial role in the development of the IBIS Management System. It refers to the specific deadline or time frame within which the system needs to be completed, including all phases of development, testing, and deployment. By efficiently managing the project timeline, prioritizing tasks, fostering effective communication, and employing appropriate project management methodologies, the project team can navigate the time constraint successfully. This allows for the timely development, testing, and deployment of the IBIS Management System, ensuring its availability to end-users within the predefined time frame.

Besides, the development of the IBIS Management System may encounter technical constraints and compatibility requirements that dictate the choice of programming languages, frameworks, or platforms to be utilized. These constraints could arise from existing infrastructure, integration requirements with other systems, or compatibility with hardware and software components.

5.3 Future Works

The IBIS Management System has been developed based on the specific requirements outlined by the IBIS Company. As part of future works, there is potential to expand the system by incorporating additional modules that cater to various aspects of company management. These modules can address areas such as inventory management, project tracking, resource allocation, and financial management, among others. Furthermore, considering the increasing use of mobile devices, developing a mobile application for the IBIS Management System can enhance accessibility for end users. This would enable users to access the system conveniently from their smartphones or tablets, providing them with on-the-go access to important features and information. By continuously improving and expanding the system's functionality, as well as adapting it for mobile platforms, the IBIS Management System can evolve to better serve the needs of the company and its users in the future.

References

About Journey - Niagawan. (n.d.). Retrieved December 9, 2022, from https://niagawan.com/my/about-journey-2/

Europe House. (2023). *Understanding the Impact of Technology on Business*. https://www.arnia.com/understanding-the-impact-of-technology-on-business/

Lucid Content Team. (2023). *4 Phases of Rapid Application Development Methodology*. https://www.lucidchart.com/blog/rapid-application-development-methodology

Matthew Martin. (2023). *RAD MODEL*. https://www.guru99.com/what-is-rad-rapid-software-development-model-advantages-disadvantages.html

Niagawan - Accounting System for Non-Accountants. (n.d.). Retrieved December 9, 2022, from https://app.niagawan.com/login/?login=1

Niagawan | Penyelesaian Masalah Akaun Bisnes Anda. (n.d.). Retrieved December 8, 2022, from https://niagawan.com/lp/

Port Cities. (n.d.). *Third-Party Odoo Integrations*. https://portcities.net/en_GB/contact

Rapid Application Development (RAD): Full Guide | Creatio. (n.d.). Retrieved December 9, 2022, from https://www.creatio.com/rapid-application-development

Tipalti Inc. (2022). *The Total Guide to Odoo ERP*. https://tipalti.com/erp-integrations/odoo-erp/#:~:text=Odoo ERP system is enterprise,an ERP solution when combined.

Zoho - Cloud Software Suite and SaaS Applications. (n.d.). Retrieved December 9, 2022, from https://www.zoho.com/