



The Development of Maha Mas Medic Services Sdn.Bhd Employee Management System

Mohamad Irfan Abdul Muthalif¹, Ruhaya Ab. Aziz^{1*}

¹ *Fakulti Sains Komputer dan Teknologi Maklumat,*

Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: ruhaya@uthm.edu.my

DOI: <https://doi.org/10.30880/aitcs.2025.06.01.071>

Article Info

Received: 30 July 2024

Accepted: 18 June 2025

Available online: 30 June 2025

Keywords

Management, Salary, Web-Based,
Work Schedule

Abstract

Maha Mas Medic Services Sdn. Bhd., an ambulance provider in Selangor and Kuala Lumpur, faces inefficiencies and resource wastage using Microsoft Excel for staff management. Issues include time-consuming processes, overstaffing, missed schedules, and potential salary calculation errors. To address these, a web-based Employee Management System is proposed. The objectives are to analyze and design the system using an object-oriented approach, develop it as a web-based application, and test its functionality and usability through alpha and beta testing. The development process follows a prototyping model, using tools like Visual Studio Code and phpMyAdmin, aiming to streamline operations, enhance accuracy, and improve overall efficiency. The system gives administrators control over user accounts, staff, schedules, salaries, and ambulances, while allowing staff to view schedules, check salaries, and request changes. Notifications will alert users to schedule updates and requests. This system is expected to significantly improve operational efficiency and accuracy.

1. Introduction

The "Employee Management System" project aims to streamline staff information, schedules, and salaries, offering an adaptable software solution for small and medium-sized companies to manage personnel information efficiently. An adaptable software solution for small and medium-sized companies facilitates personnel information management, crucial for organizational assets[1]. Effective work schedules, managed by administrators, ensure smooth business operations, while regular shift checks aid in effective workforce management. Accurate management of staff data and salaries is crucial for legal compliance, fair evaluations, and efficient resource allocation. Transparent salary management fosters trust and satisfaction, aiding recruitment and retention efforts, forming a foundational approach for organizational success.

At Maha Mas Medic Services Sdn. Bhd., an provider of ambulance services in Selangor and Kuala Lumpur, administrators currently use traditional methods for scheduling work, staff data management, and salary calculations. These methods have led to increased risks of errors. including time-consuming processes, overstaffing, missed schedules, and potential salary calculation errors. Issues such as poor print quality can cause schedule misinterpretations, and manual calculations are prone to human error, resulting in financial losses and significant time spent on rectifications. Additionally, inadequate data backups and the growing number of staff pose risks to data security. Failure to address these issues could lead to anxiety and organizational chaos, potentially disrupting business processes. Therefore, a system is needed to manage these problems, ensuring smoother and more efficient staff management.

The system is planned to help the team manage work schedules, staff data, and salaries at Maha Mas Medic Services Sdn. Bhd., located in Sg. Choh, Rawang. Designed as a web-based system, it allows access for both staff and administrators with a good internet connection. This system includes the modules listed in Table 1:

Table 1: Modules and functions

Module	Function
Login	The administrator and staff should be able to login to their own accounts when correct username and password that been given by administrator and can be able to logout anytime.
Manage user account	The administrator should be able to manage their user account by updating the username and password of the staff
Manage staff information	The administrator should be able to view, add, update or delete information or data of any staff.
Manage work schedule	The administrator should be able to create, update or delete a work schedule in this system.
Request change	The staff should be able to view and request for changing their shift in the work schedule created. The administrator should be able to view and approves or reject any request.
Manage salary	The administrator should be able calculate salary for staff and update status. Staff only view the status .
Generate scheduling report	1. The system should be able to generate a report about the work schedule created. 2. The system should be able to generate a report about the staff salary of the month.
Manage Ambulance Information	The administrator should be able to manage available ambulance that needed to assign in work schedule. Staff can only view the ambulance that been assigned.

The system includes modules for login, user account management, staff information management, work schedule management, change requests, salary management, scheduling report generation, and ambulance information management. Administrators can manage user accounts, create and update work schedules, and handle staff information and salary calculations. Staff can view schedules, check salary statuses, and request shift changes. Notifications will alert users to schedule updates and requests. The development process follows a prototyping model, using tools like Visual Studio Code and phpMyAdmin, aiming to streamline operations, enhance accuracy, and improve overall efficiency. The expected outcome is an efficient and effective Employee Management System that minimizes errors and optimizes staff management processes.

2. Related Work

2.1 Background of Maha Mas Medic Services Sdn.Bhd

Maha Mas Medic Services Sdn. Bhd., founded by Dr. Abdul Saleem Hj. Said Mohamad in 2004, provides emergency and non-emergency medical services across Malaysia. Located at 17, Jalan Batu Muda Tambahan 1, Kuala Lumpur, the company employs 25 to 30 staff members specializing in medical care, management, and ambulance driving. Effective employee management is crucial for the company's success [3]. Currently, the administrator uses Microsoft Excel to manage staff information, work schedules, and salaries, which is time-consuming and less efficient. To improve security and usability, the employee management system needs an upgrade to integrate work schedules, information, and salary management into a single system.

2.2 Employee Management System

Employees management is built on ideas and procedures developed to increase worker motivation, productivity, and performance [4]. Clearly, an employee management system, which also known as human resource management system (HRMS), is a tool that integrates ideas and procedures with the goal of assisting employees in performing at their best and achieving company's goals.

Employee management system is widely used in any organization in this century. It is clearly showing the important of the system in the organization. Various industries have implemented this employee management system to suit their specific needs and operational requirements. An employee management system is a special kind of system made to keep track of employee details and how work gets done in a business. Relevant employee details can not only be designed to be held secret from public access but can also be configured to be kept private from someone other than the software administrator or the organization’s leader. Therefore, having a computerized employee management system in a company is a big improvement. It gets rid of doing things manually, which saves a lot of time and makes tasks more efficient. This system holds important information like employee details, and leave records. Having this data in the system helps the organization to make decisions that affect how it operates.

Dissanayake and Dias [5] state that the static websites evolved by incorporating with server-side development languages like PHP or JAVA and the client-side processing languages such as JavaScript. With the help of server-side and client-side application components, the website will evolve to web-based system, which is also known as web application, that is capable in processing and producing the information. Löfberg and Molin [6] state that there are a few advantages gained by using web-based system. Firstly, web-based system is platform independent since the system can be accessed through a web browser regardless of the user’s operative system. Besides, the system can be updated and deployed easily as the user has a stable internet connection.

2.3 Comparison Of Existing System

In this section, the proposed system is compared to three existing systems: Workday, MintHCM, and OrangeHRM. The comparison, shown in Table 2, uses simple symbols ✓ and ✗ to indicate whether the systems support certain features. The features are classified into three main categories: operating platform, user experience, and management functions.

Table 2: comparison of 3 Existing system with proposed system

Categories	Feature/System	A	B	C	D
Operating Platform	Android / iOS	/	/	/	/
	Windows	/	/	/	/
User Experience (UX)	Advertising banner pop-up	x	x	x	x
	Free to use	x	x	x	/
	Paid to use	/	/	/	
Management Functions	Login	/	/	/	/
	Staff information management	/	/	/	/
	Payroll / salary	/	x	x	/
	Performance	/	/	/	/
	Work Schedule/ Time Table	x	/	x	/
	Generate report	/	x	/	/
	Notification	/	/	/	/
	Leave management	x	/	/	/
	Attendance	/	/	/	
	Recruitment	x	/	/	x

A- Workday
 B- MintHCM
 C- OrangeHRM
 D- Proposed System

/ - Available
 x- Not Available

Table 2 compares four systems, the proposed system, Workday, MintHCM, and OrangeHRM. All include login, employee management, performance, and notification modules. MintHCM and OrangeHRM lack a salary module, while Workday and OrangeHRM lack work schedule functionality. Only MintHCM cannot generate reports. Workday lacks staff leave request features, and the proposed system lacks an attendance module.

Recruitment features are absent in the proposed system and Workday. The systems vary in compatibility with Android/iOS and Windows. User experience differs by the presence of ads and whether the system is free or paid. Key management functions include login, staff information management, payroll, performance, work schedules, report generation, notifications, and attendance tracking.

3. Methodology/Framework

The prototyping model is selected to assist project development of this Employee Management System. Prototyping model implements the main phases that present in the system development life cycle process, including five phases, which are planning phase, analysis phase, design phase, the implementation phase and testing phase. The prototypes will be improved according to the stakeholders' needs as the system is under development [4]. Moreover, the prototyping model is iterative that enable the developed prototypes to be improved to fulfil the requirements of the stakeholders [7]. Every phase must fulfil the requirement specification. There are total of five phases from the prototype model. As shown in Table 3, each phase has its own assignment and output that need to produce during the entire project development. Besides that, the output had been completed within the specific days that have been given.

Table 3: Software development activities and their task

Phase	Task	Output
Planning	– Proposed the project	– Project proposal
	– Determine the project schedule, activities and output	– Develop Gantt chart
	– Identify problems, objectives and scope	
Analysis	Drive requirements:	– Problem identification and data collection through an interview
	– User requirement	– Software & hardware requirements for developer
	– System development requirement	– Software & hardware requirements for user
	– System requirement	
Design	– Design database	– Use case diagram for the system
	– Design interface	– Class Diagram for database design
	– System Architecture	– Sequence diagram for system flow
Implementation	– Develop the system modules that connect with database	– Proposed System
Testing	– Alpha testing & Beta Testing	– Error found and fixed, ensure that users are satisfied with the system
Prototype Iteration 1	– Collect feedback from stakeholder	– Improved system prototype
	– Repeat analysis, design and implementation	
Prototype Iteration 2	– Repeat analysis, design and implementation	– Finalised system prototype
	– Collect feedback from stakeholder	

4. Analysis and Design

As shown in Table 4, Functional requirement analysis is a process that show the developer the functionalities of the modules for the proposed system.

Table 4: Functionalities requirement for the system

No	Modules	Functionalities
1	Login Module	<ol style="list-style-type: none"> 1) The system should allow the users to login to the system using username and password given. 2) The system should redirect users to their respective main page once successful login.
2	Manage User Account Module	<ol style="list-style-type: none"> 1) The system should allow the administrator to change their username and password. 2) The system should allows the administrator to change their username and password. 3) The system should save the changes made in the database.
3	Manage Staff Information Module	<ol style="list-style-type: none"> 1) The system should display the staff details. 2) The system should allow the administrator to add, update or delete staff's information. 3) The system should save the changes made by the administrator in database.
4	Manage Work Schedule Module	<ol style="list-style-type: none"> 1) The system should display the work schedule details. 2) The system should allow the administrator to add new details, update the details or delete details in the work schedule. 3) The system should save the changes made by the administrator in database. 4) The system should display the work schedule details to the staffs.
5	Request Change Module	<ol style="list-style-type: none"> 1) The system should allow the staff to make a request on changing the working shift that have assigned to them. 2) The system should save the request made and can be displayed to administrator. 3) The system should allow the administrator to update the status of the request
6	Ambulance Management Module	<ol style="list-style-type: none"> 1) The system should allow display the ambulance details and status. 2) The system should allows the administrator to add new details, update the details or delete details about the ambulance. 3) The system should save the changes made by the administrator in database.
7	Manage Salary Module	<ol style="list-style-type: none"> 1) The system should display the salary details and status. 2) The system should allow the administrator to add new details, update the details or delete details of the salary. 3) The system should calculate the salary that inserted by administrator and save the changes made by the administrator in database. 4) The system should display the salary details and status to the staffs.
8	Generate Report Module	<ol style="list-style-type: none"> 1) The system should able to generate a work schedule report after the administrator selected a schedule 2) .The system should be able to generate a salary report after the administrator selected the month. 3) The system should save the changes made by the administrator in database.

As shown in Table 5, Non-functional requirement analysis is a process that show the developer the capabilities and constraints of the proposed system.

Table 5: Non-functionalities requirement for the system

No	Requirement	Description
1	Security	1) Users can only access their own account by using valid username and password.
2	Operational	1) The system should able to be accessed on any web browser. 2) The system should be user friendly.
3	Performance	1) .The system should be able for use anytime. 2) The interaction between user and the system should not be more than 5 seconds.

4.1 Use Case Diagram

Figure 1, shows the use case diagram that represents the overall activity of the new system. The purpose of use case diagram is to capture the dynamic aspect of a system. The actors of this diagram are users, and administrator.

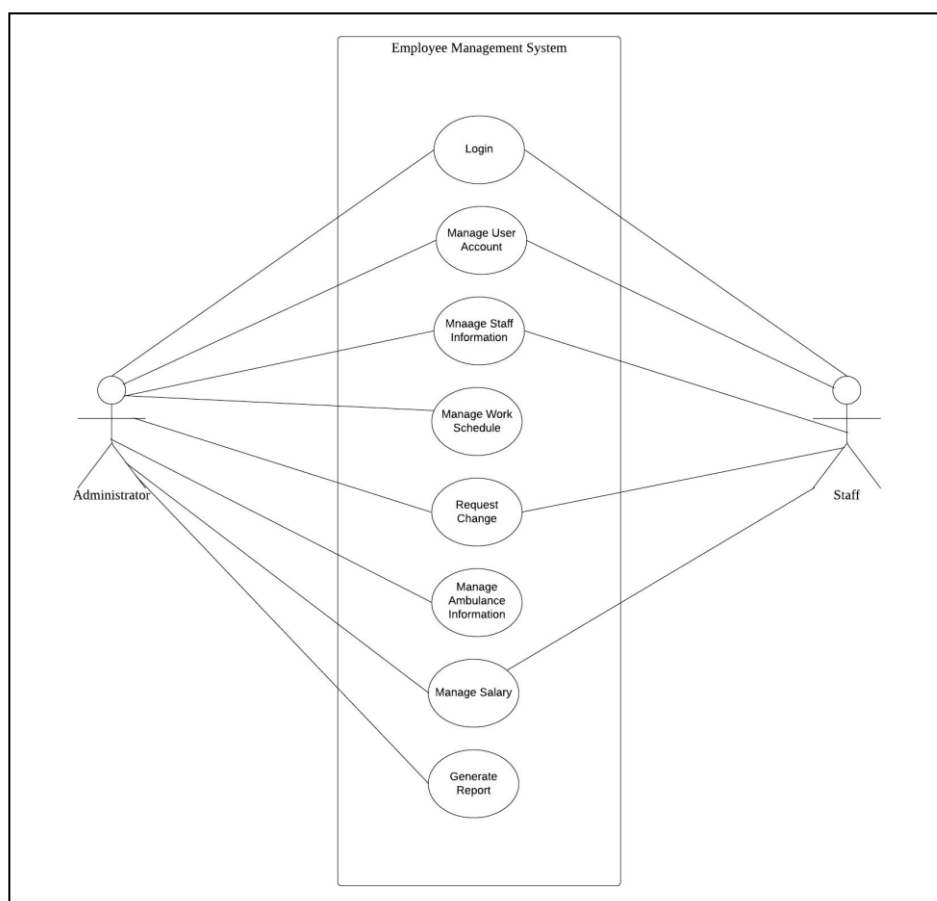


Figure 1: Use case Diagram For Employee Management System

From Figure 1, the use case is the action the actor perform when interacting with the system. The actors are the users of the system show are the administrator and the staff.

The login use case covers the login process such as user authentication and authorization. The manage user account is about managing the users account such as username and password. The manage staff

information use case covers view, add, update and delete the staff information. This can only be performed by the Administrator and can only view by staff.

The manage Work Schedule use case covers about create work schedule for the staffs, update and delete the schedules. This can be performed only by the administrator and can only view by the staff. The request change use case is about managing the request change that been requested by the staff according their schedule and the status can be changed by the administrator only.

The manage ambulance information use case covers about the management of the ambulance which the administrator only can perform it by add, update or delete any ambulance information. The manage salary use case covers about the administrator managing the staff salary calculations such as crate, update and delete and the staff can view the salary created. The generate report use case covers about generating the overall work schedule and salary report.

The sequence diagrams and activity diagrams for each use case are attached in **Appendix A**.

The class diagram shows the static view of the structure of the system and how the system components interact with each other. The class diagram for the employee management system is shown in Figure 2.

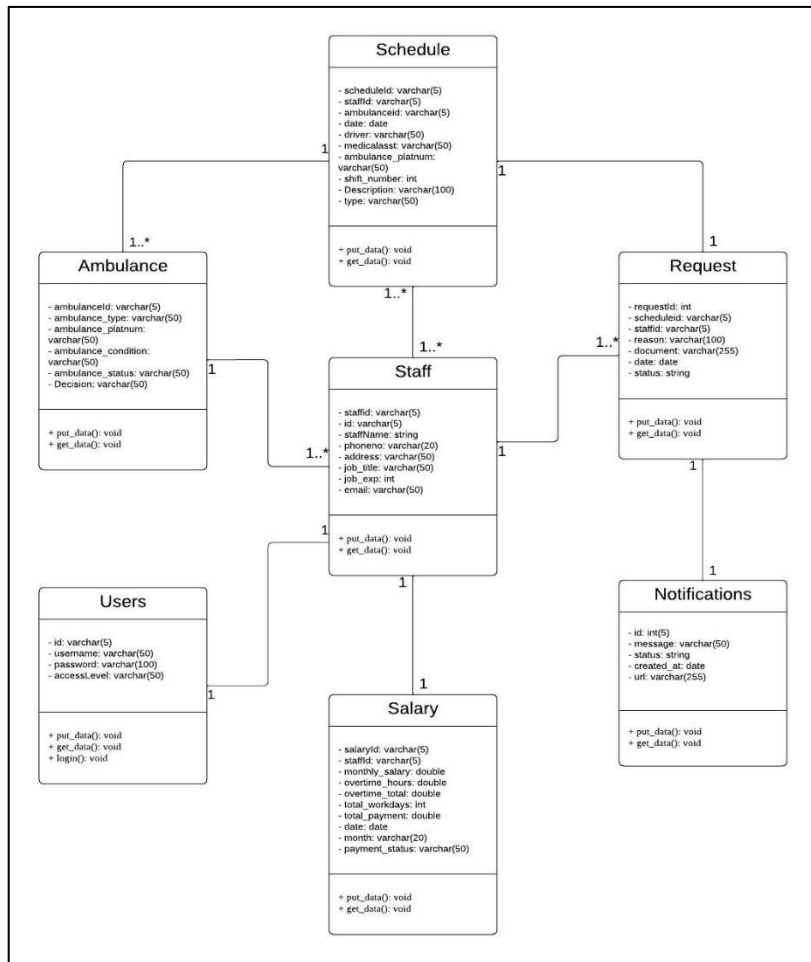


Figure 2: Class Diagram

4.2 System Design

The system architecture is the whole overview of how the information system components are organized and interact with each other. The system was developed based on the client-server architecture. Figure 3 shows the system architecture for the system.

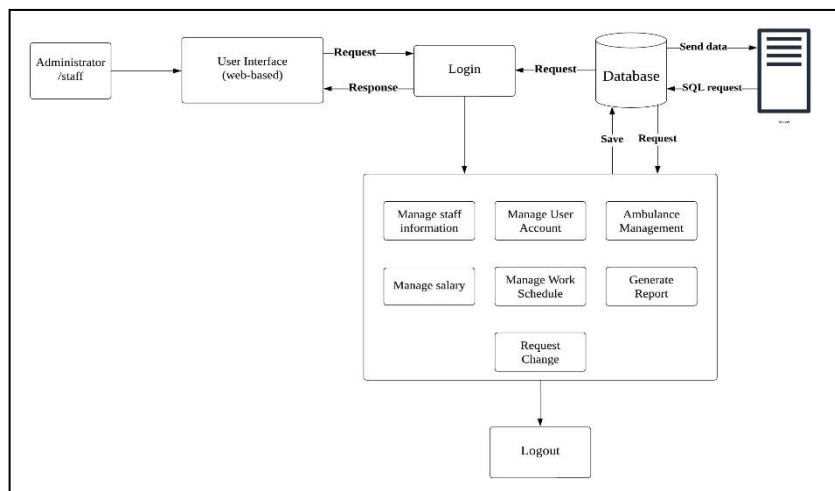


Figure 3: System Architecture

The database is designed to hold and manage the data of the system. The database schema which describes the entities that hold the different data in the database is shown.

- i. **Users**(id, username, password, accessLevel)
- ii. **Staff** (staffId, id, staffName, phoneno, adresss, job_title, job_exp, email)
- iii. **Schedule**(scheduleId, staffId, ambulanceId, date, driver, medicalasst, ambulance_platnum, shift_number, Description, type)
- iv. **Salary**(salaryId, staffId, monthly_salary, overtime_hours, overtime_total, total_workdays, total_payment, date, month, payment_status)
- v. **Ambulance**(ambulanceId, ambulance_type, ambulance_platnum, ambulance_condition, ambulance_status, Decision)
- vi. **Request**(requestId, staffId, scheduleId, reason, document, date, status)
- vii. **Notification**(id, message, status, created_at, url)

The user interfaces of the systems are designed to give the user good experience when using the system. Figure 4 shows an example of user interface design for the login page.



Figure 4: Login Page Interface Design

5. Result and Discussion

5.1 Implementation

The employee management system is a web-based application with a front end developed using PHP. The system includes a login module for user authentication, where users must enter a valid username and password

to gain access. If the credentials match, the login is successful, otherwise, it fails. Figure 5 illustrates the user interface for the login form along with its validation code segment that checks the entered username and password.



Fig.5 Login Module: (a) Interface; (b) Coding Segment

The Manage User Account module allows administrators to add, update, or delete user account information for both administrators and staff. The system updates the user list after any changes. Administrators can also search for user information using related data. The interface for this module is shown in Figure 6.

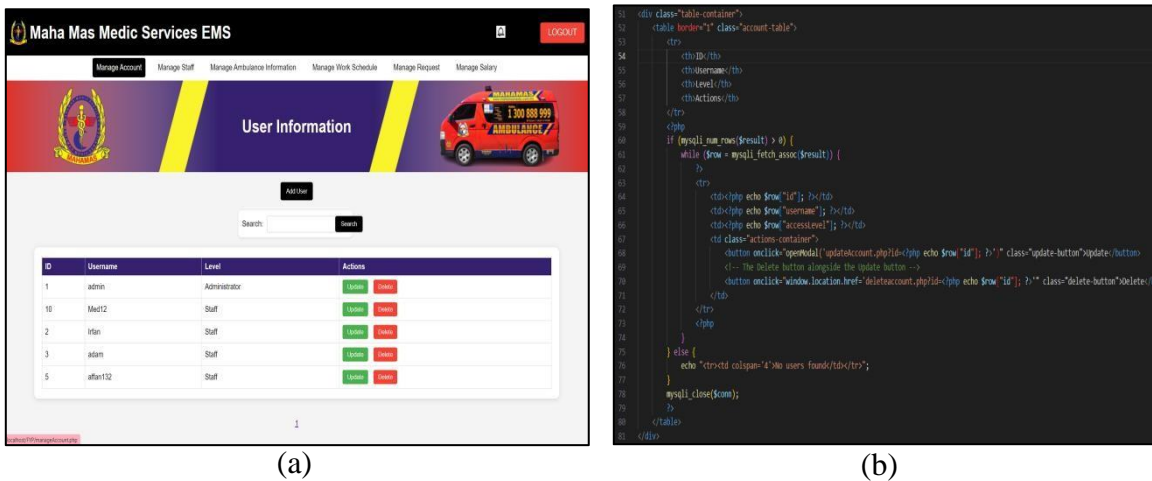


Fig.6 Manage User Account Module: (a) Interface; (b) Coding Segment

The Manage Staff Information module allows administrators to add, update, or delete staff information, with the system updating the staff list accordingly. Administrators can search for staff details using related information. Staff members can view their own information on their personal page. The interface for this module is shown in Figure 7.

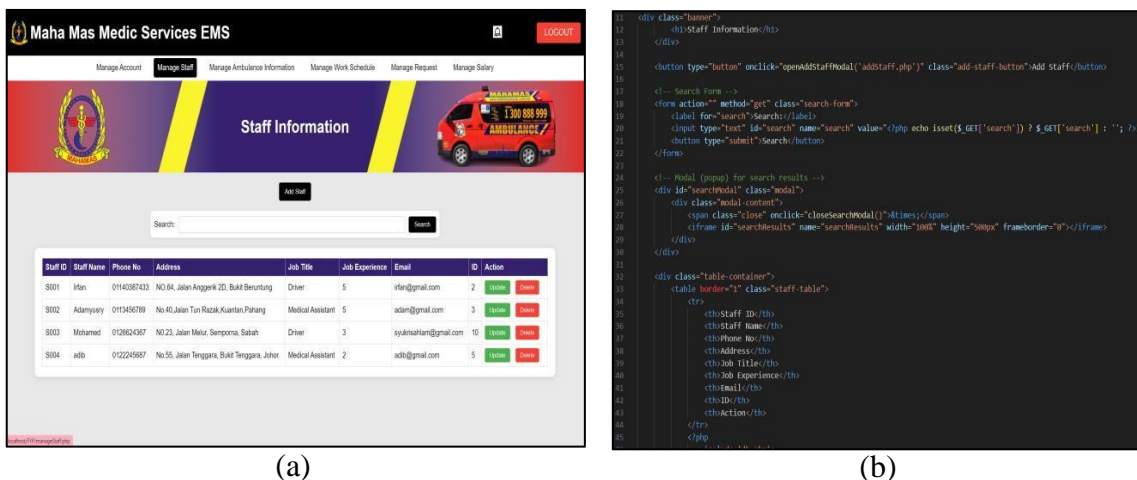


Fig.7 Manage Staff Information Module: (a) Interface; (b) Coding Segment

The Manage Schedule module allows administrators to add, delete, or update work schedules, with changes automatically reflected in the schedule list. Administrators can search schedules by date, and on the Add Schedule page, input required fields and click "Add Schedule" to display the new schedule. On the Update Schedule page, they modify fields and click "Update Schedule" to show the changes. For deletions, administrators verify information before clicking "Delete Schedule." Staff can view updated schedules on their page. The interface for this module is shown in Figure 8.

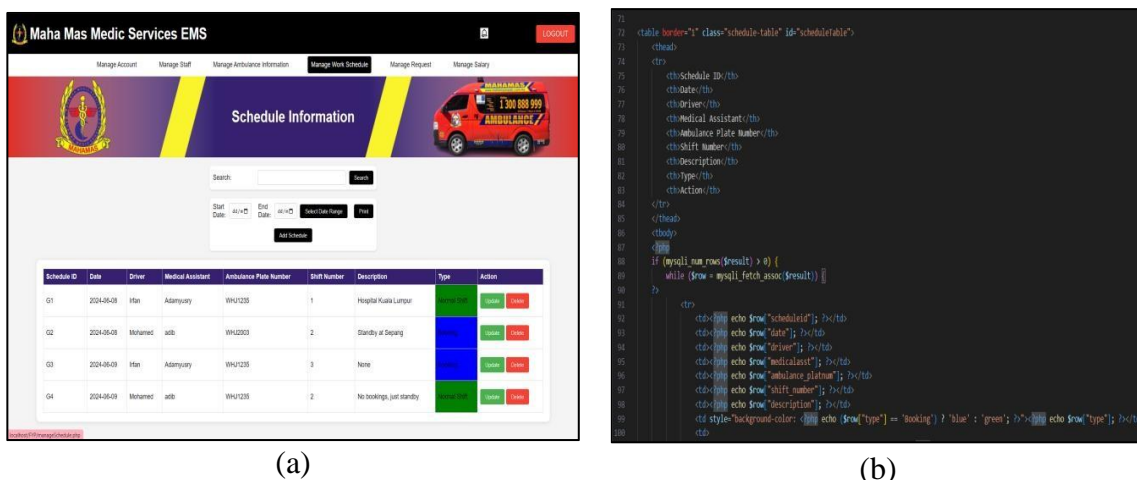
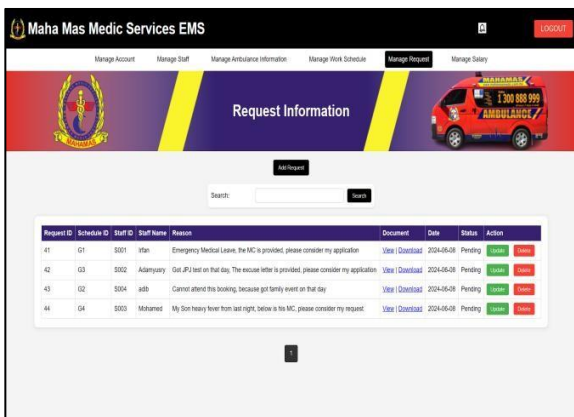


Fig.8 Manage Work Schedule: (a) Interface; (b) Coding Segment

The Request Change in Working Shift module allows staff to send shift change requests to the manager by selecting a reason, providing a description and document, and clicking "Submit." Staff can view and search their requests and statuses in the "Request Change Made" tab. Administrators can view, search, approve, or reject requests by staff name, reason, and document in the Manage Request Page, with the system updating the status accordingly. The interface for this module is shown in Figure 9.



(a)

```

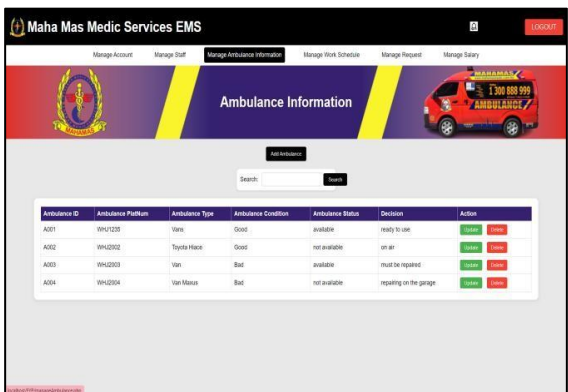
1 <title>Request Information</title>
2 <link rel="stylesheet" href="requestable.css"> <!-- Link to combined CSS file -->
3 </head>
4 <body>
5 <php include "header.php">
6 <div class="banner">
7 <h1>Request Information</h1>
8 </div>
9
10 <!-- Update the button to call the JavaScript function -->
11 <button type="button" onclick="openAddRequestModal('addreq.php');" class="add request button">Add Request</button>
12
13 <form action="" method="get" class="search form">
14 <input type="text" value="Search" />
15 <input type="submit" value="Search" />
16 </form>
17
18 <div class="table container">
19 <table border="1" class="request-table">
20 <tr>
21 <th>Request ID</th>
22 <th>Schedule ID</th>
23 <th>Staff ID</th>
24 <th>Staff Name</th>
25 <th>Reason</th>
26 <th>Document</th>
27 <th>Date</th>
28 <th>Status</th>
29 <th>Action</th>
30 </tr>
31 <tr>
32 <td>41</td>
33 <td>01</td>
34 <td>S01</td>
35 <td>Ivan</td>
36 <td>Emergency Medical Leave. The MC is provided, please consider my application</td>
37 <td>View</td>
38 <td>2024-05-08</td>
39 <td>Pending</td>
40 <td><button type="button" class="green">View</button> <button type="button" class="red">Delete</button></td>
41 </tr>
42 <tr>
43 <td>42</td>
44 <td>03</td>
45 <td>S02</td>
46 <td>Admynova</td>
47 <td>Out of list on that day. The excuse letter is provided, please consider my application</td>
48 <td>View</td>
49 <td>2024-05-08</td>
50 <td>Pending</td>
51 <td><button type="button" class="green">View</button> <button type="button" class="red">Delete</button></td>
52 </tr>
53 <tr>
54 <td>43</td>
55 <td>02</td>
56 <td>S04</td>
57 <td>add</td>
58 <td>Cannot attend this booking, because got family event on that day</td>
59 <td>View</td>
60 <td>2024-05-08</td>
61 <td>Pending</td>
62 <td><button type="button" class="green">View</button> <button type="button" class="red">Delete</button></td>
63 </tr>
64 <tr>
65 <td>44</td>
66 <td>04</td>
67 <td>S03</td>
68 <td>Mahamad</td>
69 <td>My Son heavily fever from last night, below is the MC, please consider my request</td>
70 <td>View</td>
71 <td>2024-05-08</td>
72 <td>Pending</td>
73 <td><button type="button" class="green">View</button> <button type="button" class="red">Delete</button></td>
74 </tr>
75 </table>
76 </div>

```

(b)

Fig. 9: Request Change: (a) Interface; (b) Coding Segment

The Manage Ambulance Information module allows administrators to add, delete, or update ambulance records, with changes reflected automatically. Administrators fill in fields and click "Add Ambulance" to add, "Update Ambulance" to update, or "Delete" to remove records. This ensures the ambulance list is always current. The interface is shown in Figure 10.



(a)

```

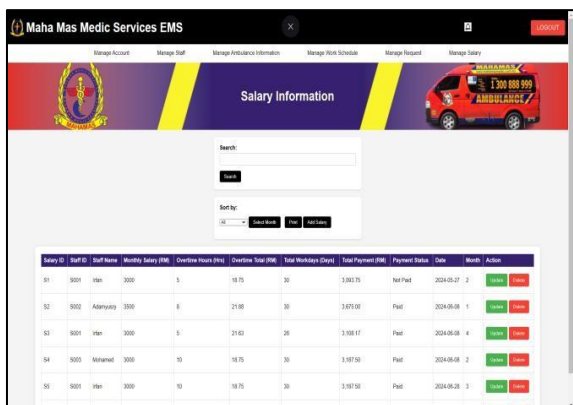
1 <table border="1" class="ambulance-table">
2 <tr>
3 <th>Ambulance ID</th>
4 <th>Ambulance PlatNum</th>
5 <th>Ambulance Type</th>
6 <th>Ambulance Condition</th>
7 <th>Ambulance Status</th>
8 <th>Decision</th>
9 <th>Action</th>
10 </tr>
11 <tr>
12 <td>A001</td>
13 <td>WPK1235</td>
14 <td>Van</td>
15 <td>Good</td>
16 <td>available</td>
17 <td>ready to use</td>
18 <td><button type="button" class="green">View</button> <button type="button" class="red">Delete</button></td>
19 </tr>
20 <tr>
21 <td>A002</td>
22 <td>WPK002</td>
23 <td>Toyota Hiace</td>
24 <td>Good</td>
25 <td>not available</td>
26 <td>on air</td>
27 <td><button type="button" class="green">View</button> <button type="button" class="red">Delete</button></td>
28 </tr>
29 <tr>
30 <td>A003</td>
31 <td>WPK003</td>
32 <td>Van</td>
33 <td>Bad</td>
34 <td>available</td>
35 <td>must be repaired</td>
36 <td><button type="button" class="green">View</button> <button type="button" class="red">Delete</button></td>
37 </tr>
38 <tr>
39 <td>A004</td>
40 <td>WPK004</td>
41 <td>Van Maxus</td>
42 <td>Bad</td>
43 <td>not available</td>
44 <td>repairing on the garage</td>
45 <td><button type="button" class="green">View</button> <button type="button" class="red">Delete</button></td>
46 </tr>
47 </table>

```

(b)

Fig.10 Manage Ambulance Information : (a) Interface; (b) Coding Segment

The Manage Salary module allows administrators to add, delete, or update salary records, with changes automatically reflected in the list. Administrators fill in fields and click "Add Salary" to add, "Update Salary" to update, or "Delete" to remove records. Staff can view their salary details and print pay slips from their page. The interface is shown in Figure 11.



(a)

```

1 <div class="name">
2 <h2>Salary Information</h2>
3 </div>
4 <form action="" method="get" class="search-form" target="_self">
5 <input type="text" value="Search" />
6 <input type="submit" value="Search" />
7 </form>
8 <form action="" method="get" class="set-form" target="_self" style="margin-top: 10px;">
9 <input type="text" value="Set by:" />
10 <input type="submit" value="Set by:" />
11 <input type="text" value="Month:" />
12 <input type="submit" value="Month:" />
13 </form>
14 <table border="1" class="salary-table" id="salaryTable">
15 <tr>
16 <th>Salary ID</th>
17 <th>Staff ID</th>
18 <th>Staff Name</th>
19 <th>Monthly Salary (RM)</th>
20 <th>Overtime Hours (hr)</th>
21 <th>Overtime Total (RM)</th>
22 <th>Total Workdays (Dpd)</th>
23 <th>Total Payment (RM)</th>
24 <th>Payment Status</th>
25 <th>Date</th>
26 <th>Month</th>
27 <th>Action</th>
28 </tr>
29 <tr>
30 <td>S1</td>
31 <td>S001</td>
32 <td>Ivan</td>
33 <td>3000</td>
34 <td>5</td>
35 <td>15,075</td>
36 <td>30</td>
37 <td>2,042.75</td>
38 <td>Not Paid</td>
39 <td>2024-05-27</td>
40 <td>5</td>
41 <td><button type="button" class="green">View</button> <button type="button" class="red">Delete</button></td>
42 </tr>
43 <tr>
44 <td>S2</td>
45 <td>S002</td>
46 <td>Admynova</td>
47 <td>3000</td>
48 <td>5</td>
49 <td>21,180</td>
50 <td>30</td>
51 <td>3,675.00</td>
52 <td>Paid</td>
53 <td>2024-05-08</td>
54 <td>1</td>
55 <td><button type="button" class="green">View</button> <button type="button" class="red">Delete</button></td>
56 </tr>
57 <tr>
58 <td>S3</td>
59 <td>S001</td>
60 <td>Ivan</td>
61 <td>3000</td>
62 <td>5</td>
63 <td>21,180</td>
64 <td>20</td>
65 <td>3,100.17</td>
66 <td>Paid</td>
67 <td>2024-05-08</td>
68 <td>4</td>
69 <td><button type="button" class="green">View</button> <button type="button" class="red">Delete</button></td>
70 </tr>
71 <tr>
72 <td>S4</td>
73 <td>S003</td>
74 <td>Mahamad</td>
75 <td>3000</td>
76 <td>10</td>
77 <td>18,175</td>
78 <td>30</td>
79 <td>3,187.50</td>
80 <td>Paid</td>
81 <td>2024-05-08</td>
82 <td>3</td>
83 <td><button type="button" class="green">View</button> <button type="button" class="red">Delete</button></td>
84 </tr>
85 <tr>
86 <td>S5</td>
87 <td>S001</td>
88 <td>Ivan</td>
89 <td>3000</td>
90 <td>10</td>
91 <td>18,175</td>
92 <td>30</td>
93 <td>3,187.50</td>
94 <td>Paid</td>
95 <td>2024-05-28</td>
96 <td>3</td>
97 <td><button type="button" class="green">View</button> <button type="button" class="red">Delete</button></td>
98 </tr>
99 </table>

```

(b)

Fig.11 Manage Salary: (a) Interface; (b) Coding Segment

The Generate Report module allows administrators to create work schedule reports for a specified date range and salary reports for any chosen month, with the option to print individual pay slips for staff members. The system displays the report for the administrator and provides a PDF download option. The interface for this module is shown in Figure 12.

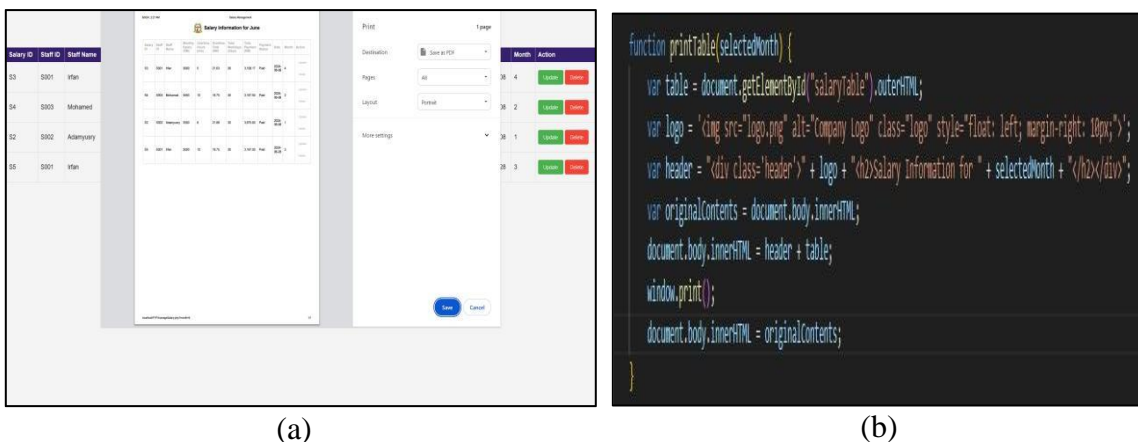


Fig.12 Generate Report: (a) Interface; (b) Coding Segment

5.2 Testing

Two types of testing are carried out which functionality testing and user acceptance testing. In functionality testing, the test cases are derived based on the functional requirement determined. Table 6 to table 14 showsthe List of test cases according the modules. It traces the system requirements and specifications. It is used especially during testing to determine whether the system requirements are met or not. This assists in testing to tracing the requirements to make sure all the requirements are being tested.

Table 6: List of test cases for Login Module

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass / Fail)
SRS_REQ_100	TEST_100	Login	
SRS_REQ_101	TEST_100_001	The system shall allow the user to login to the system.	Pass
SRS_REQ_102	TEST_100_002	The system shall be able to handle exceptions.	Pass
SRS_REQ_103	TEST_100_003	The system shall only allow the user with correct username and password to login to the system.	Pass
SRS_REQ_104	TEST_100_004	While exception occur, the system shall return to the previous state	Pass
SRS_REQ_105	TEST_100_005	The system shall be able to inform users to re-enter their username and password	Pass

Table 7: List of test cases for Manage User Account Module

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass / Fail)
SRS_REQ_200	TEST_200	Manage User Account	

SRS_REQ_403	TEST_400_003	The system shall allow the administrator to delete the information of existing work schedule.	Pass
SRS_REQ_404	TEST_400_004	The system shall allow the administrator to update the information of existing work schedule.	Pass
SRS_REQ_405	TEST_400_005	The system shall display the details of work schedule to the staff.	Pass
SRS_REQ_406	TEST_400_006	The system shall not allow the staff to alter the data of work schedule.	Pass
SRS_REQ_406	TEST_400_007	The system shall only proceed to next process if all the required text fields are filled in.	Pass

Table 10: List of test cases for Request Change Module

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass / Fail)
SRS_REQ_500	TEST_500	Request Change	
SRS_REQ_501	TEST_500_001	The system shall allow the staff to send the change request.	Pass
SRS_REQ_502	TEST_500_002	The system shall allow the administrator to view the detail of change request.	Pass
SRS_REQ_503	TEST_500_003	The system shall allow the administrator to update the status of the change request.	Pass

Table 10: List of test cases for Request Change Module(cont)

SRS_REQ_504	TEST_500_004	The system shall only proceed to next process if all the required text fields are filled in.	Pass
SRS_REQ_505	TEST_500_005	The system shall not allow the staff to alter the data of existing request table.	Pass

Table 11: List of test cases for Manage Ambulance Information Module

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass / Fail)
SRS_REQ_600	TEST_600	Manage Ambulance Information	
SRS_REQ_601	TEST_600_001	The system can display the ambulance list to the administrator.	Pass
SRS_REQ_602	TEST_600_002	The system can display the ambulance list to the administrator.	Pass

Table 11: List of test cases for Manage Ambulance Information Module(cont)

SRS_REQ_603	TEST_600_003	The system shall allow the administrator to delete the information of existing ambulance.	Pass
SRS_REQ_604	TEST_600_004	The system shall allow the administrator to update the information of existing ambulance.	Pass
SRS_REQ_605	TEST_600_005	The system shall only proceed to next process if all the required text fields are filled in.	Pass

Table 12: List of test cases for Manage Salary Module

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass / Fail)
SRS_REQ_700	TEST_700	Manage Salary Information	
SRS_REQ_701	TEST_700_001	The system can display the staff salary list to the administrator.	Pass
SRS_REQ_702	TEST_700_002	The system shall allow the administrator to add the salary of staff.	Pass
SRS_REQ_703	TEST_700_003	The system shall allow the administrator to delete the salary of existing staff.	Pass
SRS_REQ_704	TEST_700_004	The system shall allow the administrator to update the salary of existing staff.	Pass
SRS_REQ_705	TEST_700_005	The system shall only proceed to next process if all the required text fields are filled in.	Pass

Table 13: List of test cases for Generate Report Module

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass / Fail)
SRS_REQ_800	TEST_800	Generate Report	
SRS_REQ_801	TEST_800_001	The system shall display a daily work schedule and monthly salary report.	Pass
SRS_REQ_802	TEST_800_002	The system shall allow the administrator to download the report in PDF form.	Pass
SRS_REQ_803	TEST_800_003	The system shall only allow the report of existing work schedule or current salary transactions to be created.	Pass

Table 14: List of test cases for Logout Module

Software Requirement Specification	Test Case ID	Test Case Description	Test Result (Pass / Fail)
SRS_REQ_900	TEST_900	Logout	
SRS_REQ_901	TEST_900_001	The system can allowed the user to logout.	Pass

Overall, 40 test case are tested. Table 15 shows overall test case result.

Table 15: Overall Test Case Result

Test Case ID	Total Test Case	Total Passed
TEST_100	5	5
TEST_200	5	5
TEST_300	5	5
TEST_400	6	6
TEST_500	5	5
TEST_600	5	5
TEST_700	5	5
TEST_800	3	3
TEST_900	1	1
	40	40

Table 15 shows that all 40 test cases are passed. This indicates the system is able to perform the functionalities that meet the system functional requirements.

Next, user acceptance testing is carried out with the expected user. The aspects being tested are the functionality of the system, usability of the system, and user interface design of the system. There is one user who tested the system as an Administrator and two users who tested the system as staff. The bar graph in Figure 13 shows the user satisfaction with the system.

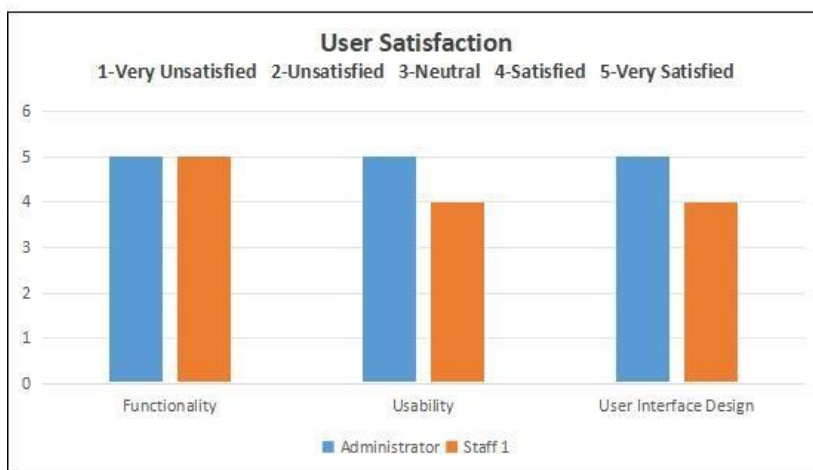


Fig.13: User Satisfaction

Figure 13 ,Both of the user indicates high user satisfaction with the system's functionality, improving staff management effectively. The Administrator finds the system's usability easy and straightforward, while the staff suggests a multi-language toolkit for further enhancement. Regarding user interface design, the Administrator is very satisfied, but the Staff suggests a slightly larger and clearer font size for improved readability.

6. Conclusion

In this project, the development of the Employee Management System for Maha Mas Medic Services Sdn. Bhd., designed to efficiently manage employee information, schedules, and salaries. The system integrates staff

availability, job roles, and shift requirements, improving operational efficiency and reducing errors. Managers can schedule employees, manage data, and process salaries effectively. Employees can request shift changes or leave, with managers responding efficiently, ensuring transparency and good communication. The project successfully improves employee management. Future enhancements will further streamline operations and support growth.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

References

- [1] A.S.SyedNavaz, A.S.SyedFiaz, C.Prabhadevi, V.Sangeetha, S.Gopalakrishnan, "Human Resource Management System", IOSR Journal of Computer Engineering (IOSR-JCE), Volume 8, Issue 4 (Jan. – Feb. 2013) Page 62- 71.
- [2] M. Menz et al., "Corporate Strategy and the Theory of the Firm in the Digital Age," Journal of Management Studies, vol. 58, no. 7, pp. 1695–1720, 2021, Available: <https://onlinelibrary.wiley.com/doi/10.1111/joms.12760>
- [3] D. R. Manna, "Strategic Aspects Of The Importance Of Employee Management", JDM, vol. 3, no. 1, pp. 1–6, Jan. 2008.
- [4] Susanto, A. & Meiryani. (2019). System Development Method with The Prototype Method. International Journal of Scientific & Technology Research, 8(7), pp. 141- 144.
- [5] N. R. Dissanayake and G. K. A. Dias, "Rich Web-based Applications: An Umbrella Term with a Definition and Taxonomies for Development Techniques and Technologies," International Journal of Future Computer and Communication, vol. 7, no. 1, pp. 14-20, 2018.
- [6] M. Löfberg and P. Molin, "Web vs. Standalone Application- A maintenance application for Business Intelligence," M. S. thesis, School of Engineering Blekinge Institute of Technology, 2005.
- [7] P. Saranya et al., "Comparative Study of Software Development Methodologies," International Research Journal of Engineering and Technology, vol. 4, no. 5, pp. 172-179, 2017.