APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 5 No. 5 (2024) 1463-1482
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

ISMS: Integrated Safety Management System

Anastazry Faidzli[1], Suziyanti Marjudi[1]*

[1] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: suziyanti@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2024.05.02.079

Article Info

Received: 12 June 2024
Accepted: 30 October 2024
Available online: 15 December 2024

Keywords

Safety management, HIRARC, incident reporting, construction, high-rise

Abstract

This study proposes the development and implementation of a web-based Integrated Safety Management System (ISMS). The proposed system can be used by industry that deals with hazards, risk or any other activities that can cause injury or harm. The system will also encompass some analytic tools such as incident reports, near misses and monthly safety analysis which can help the organization assess their construction site safety level. The case study for this project was conducted at the construction site located in Kuala Lumpur. The organization is currently facing operational challenges at one of their construction site. The core issue is their reliance on a conventional paper-based system for safety data management, which is further compounded by the small size of the on-site office, making efficient storage of essential safety documents, mandated for retention over a three-year period, quite challenging. Retrieving past documents for updates or compliance is also problematic. For the development of the proposed system, the Agile Model is selected as the system development methodology. It prioritizes continuous improvement based on customer feedback and adaptability to evolving requirements. During the system's implementation phase, the Laravel framework was used for the backend and TailwindCSS and JavaScript for the frontend. The result is a successful web-based Integrated Safety Management System (ISMS), effectively mitigating the challenges faced at construction site, significantly enhancing safety data management processes.

1. Introduction

An Integrated Safety Management System (ISMS) is essential for industries dealing with hazardous risks, particularly in construction. The system facilitates the management and storage of crucial safety data by safety workers at construction sites. This data includes Hazard Identification Risk Assessment Risk Control (HIRARC) forms, incident notifications, reports, and risk assessments, which are crucial for maintaining workplace safety and compliance with regulatory guidelines.

Physical storage of these documents often poses challenges, including accessibility and security. This is evident in the practices of Malaysian construction companies that struggle with managing HIRARC forms and incident documents. A web-based application for ISMS can address these challenges, offering efficient management and tracking of safety documents, which is a significant improvement over traditional physical storage method.

The primary goal of implementing such a system is to develop a more efficient, secure, and reliable method of managing safety data. The focus is on designing a HIRARC Information System that utilizes an object-oriented

This is an open access article under the CC BY-NC-SA 4.0 license.

approach, with the end product being a web-based application. This system aims to streamline processes, enhance data security, and improve overall safety management in the construction industry.

The ISMS is expected to transform safety management by providing a unified digital solution. This would not only improve data management and incident tracking but also ensure data integrity, leading to safer working conditions and more effective safety management practices. The adoption of such a system signifies a shift towards a more proactive and integrated approach to safety and risk management in construction and other high-risk industries.

2. Literature Review

This section explains the current safety documents management system, safety documents, HIRARC, management systems and comparison between existing system and proposed system.

2.1 Safety Documents Management System

The case study investigates the management of safety documents in high-risk industries, focusing on two companies with distinct challenges. Company A, a construction firm, struggles with the physical storage and retrieval of safety documents like HIRARC forms and incident reports, adhering to DOSH regulations. Company B, in the oil and gas sector, uses a distributed system but faces difficulties in integrating various safety management functions. The study proposes an Integrated Safety Management System (ISMS) to overcome these challenges, offering a digital, unified platform for efficient record-keeping, streamlined processes, and improved safety management compliance. This system aims to enhance safety documentation management, reduce risks, and ensure a safer working environment.

2.2 Safety Documents

Managing safety documents are utmost important in the construction industry, especially at high-rise construction sites where a significant number of accidents occur annually [1]. Proper management of these documents is essential for identifying hazards and determining control measures, thereby ensuring a safe working environment. Accidents, often sudden and unforeseen [2], can lead to injuries, damage, or loss, underscoring the need for effective safety management and documentation to prevent or minimize these incidents [3].

2.3 HIRARC

HIRARC, an acronym for Hazard Identification, Risk Assessment, and Control Determination, is a crucial work program in workplace safety management. It involves a systematic approach to identifying potential hazards in a workplace, assessing the associated risks, and implementing control measures to mitigate these risks and hazards [4]. Risk assessment, a key element of HIRARC, is part of a broader risk analysis framework. This framework consists of steps designed to recognize and evaluate all conceivable risks and issues that could negatively impact a business or enterprise. It's a continuous process, constantly updated to address emerging risks and ensure the safety and well-being of employees and the workplace [5].

2.4 Management Systems

A computerized management system is a digital tool that maintains a database of organizational maintenance operations, aiming to simplify work processes and enhance efficiency [6]. The system, centered around a specially designed database, facilitates record-keeping, tracking, and timely completion of tasks [7]. In the context of the Integrated Safety Management System at X Construction, administrators and staff can effectively manage HIRARC forms, incident notifications, incident reports, and monthly analyses through a tailored database, enabling data manipulation and retrieval within the system.

2.5 Review on Existing System

SafeSiteHq [11], SafetyCulture [12] and Synergi Life [13] are the three existing systems of safety management system.

2.6 SafeSiteHq

SafesiteHQ [11], established in 2014, was founded with a mission to enhance workplace safety. The company developed an easy-to-use safety management app with a well-organized dashboard displaying incidents summarization and weekly actions. The website allows users to access historical data easily. The HIRARC module simplifies the hazard reporting process, though there's a discrepancy regarding hazard resolution

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1463-1482

capabilities. The incident reporting module is commendable, offering five incident types, witness statements, and corrective actions, but lacks incident notifications.

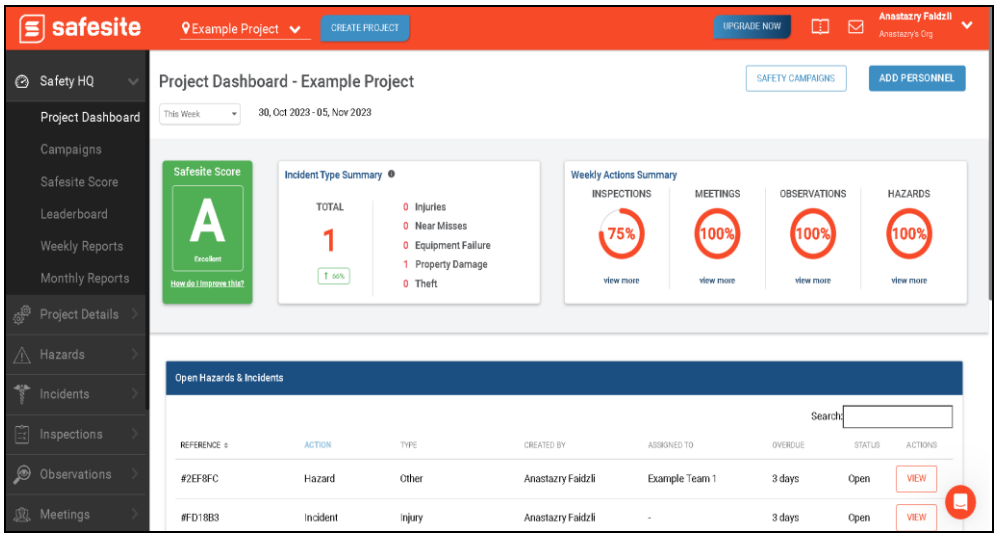


Fig. 1 *SafeSiteHq Dashboard*

2.7 SafetyCulture

SafetyCulture [12] is a workplace safety and operations solution, founded by Luke Anear. It began with the iAuditor app and has evolved into an all-in-one tool for safety, quality, and efficiency. The platform offers a mobile-first approach, a global reach, and offices in various cities. SafetyCulture's unique feature lies in using templates for HIRARC, incident reporting, and notifications, providing flexibility for different organizational safety documentation requirements. Users can choose templates from a public library or create their own, and approval from upper management, such as the Safety and Health Officer (SHO), may be required.

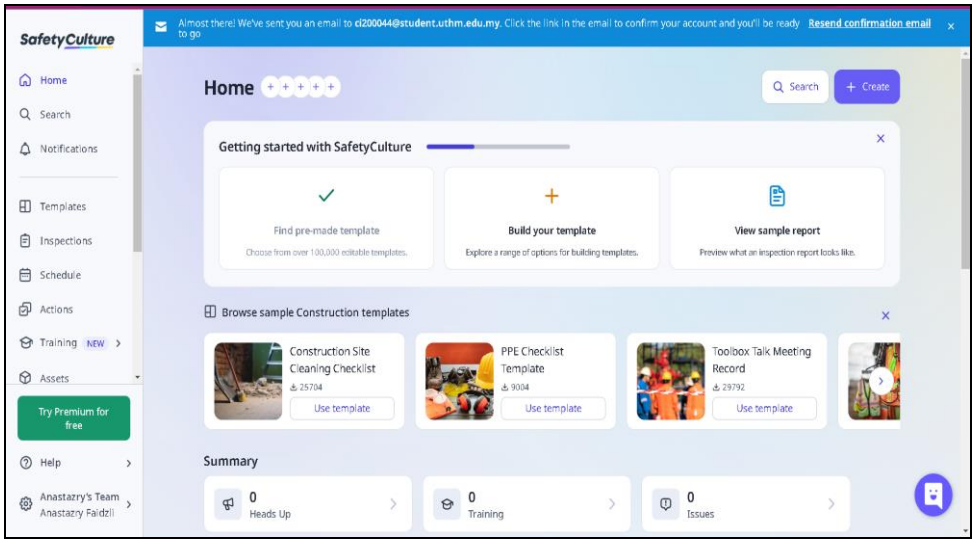


Fig. 2 *SafetyCulture's dashboard*

2.8 Synergi Life Risk Assessment

Synergi Life Risk Management [13] is a module in the DNV system that facilitates a transition from reactive to proactive risk management. The system offers benefits like process automation, compliance support, and alignment with industry best practices. The dashboard is simple and user-friendly, allowing quick case registration. The incident reporting module is efficient, categorizing incidents and utilizing scroll features for risk assessment. The system includes an action phase for incident investigation, and closed cases are clearly indicated.

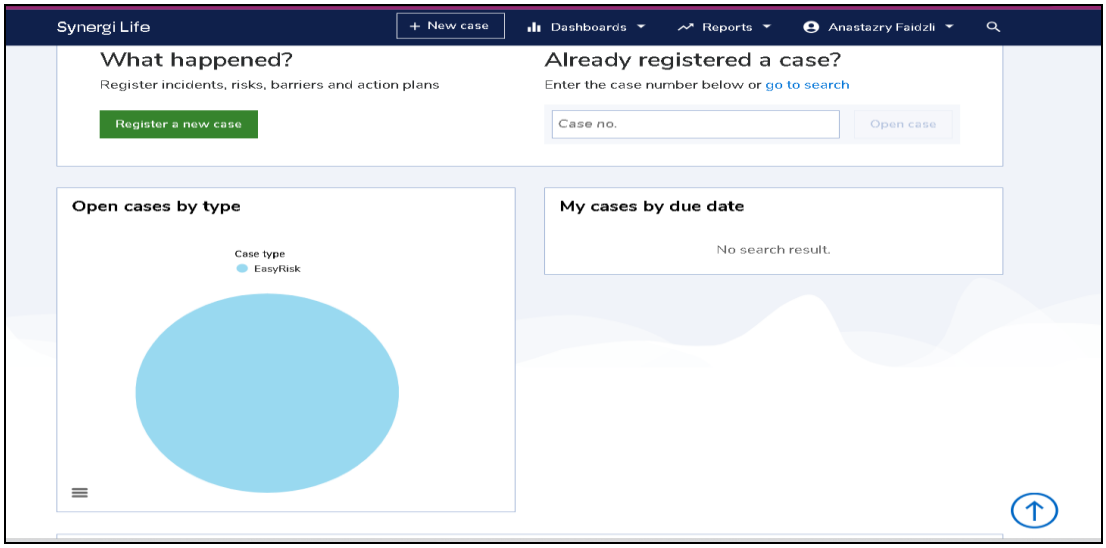


Fig. 3 *Synergi Life dashboard*

Table 1 shows the comparison between the existing system with the proposed system.

Table 1 *Comparison between the existing system with the proposed system*

Modules	SafesiteHQ	SafetyCulture	Synergi Life	ISMS
User authentication and access control module	Yes	Yes	Yes	Yes
HIRARC Management Module	Yes	Yes	No	Yes
Incident Reporting Module	Yes	Yes	Yes	Yes
Safety Analysis and Reporting Module	Yes	Yes	Yes	Yes
Management Approval and Assessment Module	No	No	No	Yes

3. Methodology

The Agile Model, chosen for developing the Integrated Safety Management System, combines iterative and incremental development. It emphasizes self-organizing, cross-functional teams collaborating to determine system needs and build it efficiently. The approach, known for its adaptability, enables developers to create high-quality products quickly and respond effectively to changes [8].

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1463-1482

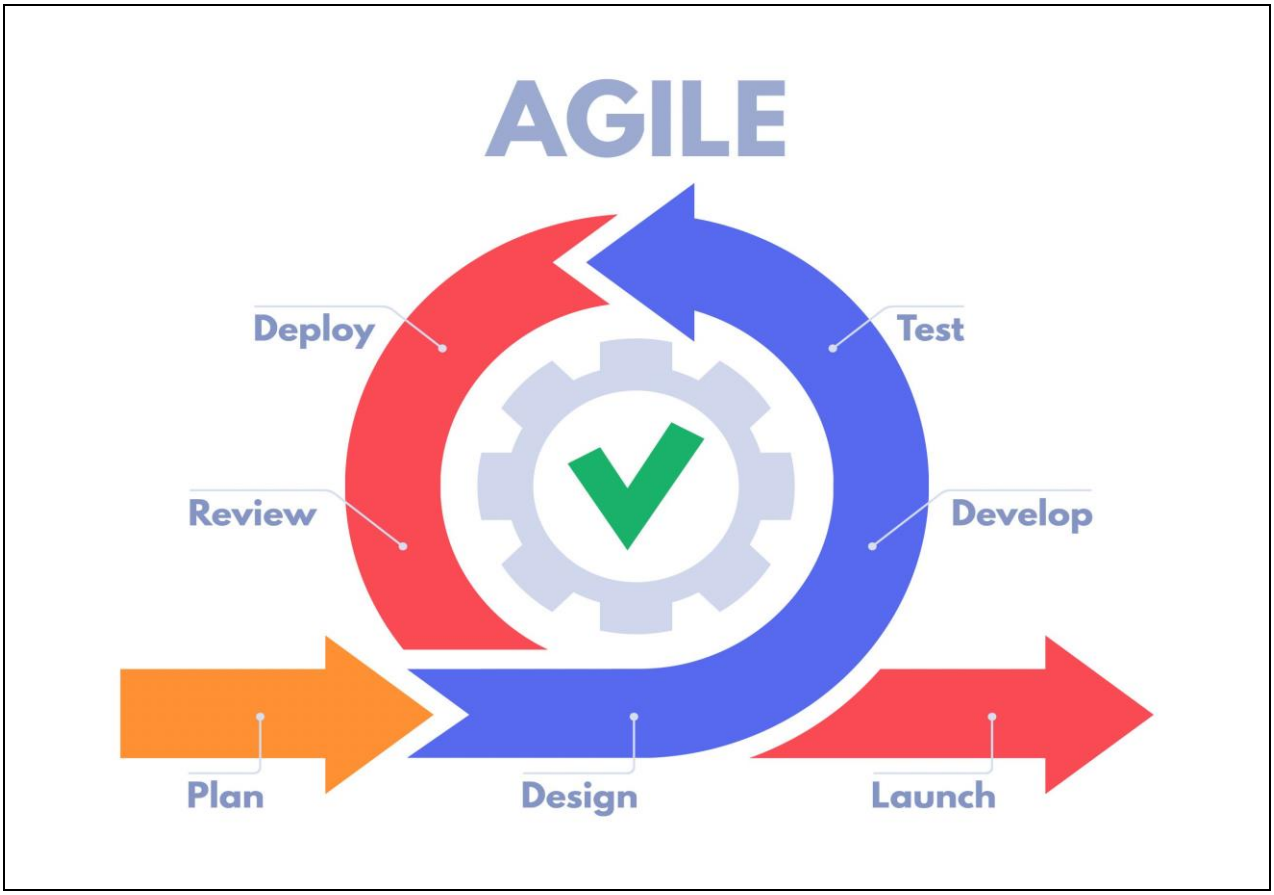


Fig. 4 *Agile Model*

Based on Figure 3, the Agile Model breaks down into five key phases: planning, analysis, design, implementation, and testing. Each phase has its own set of tasks to be completed. When all these main phases are done, it is called as an "iteration." If there are more updates needed for the system down the road, another iteration is conducted.

3.1 Planning Phase

In the Agile Model for system development, the planning phase is crucial. Problem statements needs to be identified and project objectives needs to be established. The scope, and significance also needs to be understood. Data is gathered through discussions with the safety committee and extensive research from various sources. Two methods, interviews with industries dealing with hazards and a literature review, are employed to collect information on existing systems. This data is vital for analyzing the proposed system's requirements.

3.2 Analysis Phase

In the analysis phase, the goal is to determine both functional and non-functional requirements for the proposed system. This involves examining X Construction's manual management system for safety documentation to understand existing procedures. A comprehensive analysis covers user types, modules, technology, and functionalities in existing systems, addressing identified problem statements. Comparative assessments guide the identification of requirements, considering insights from discussions with the owner and research conducted. The aim is to enhance safety documentation procedures through the proposed system.

3.3 Implementation Phase

In the Agile Model's implementation phase, the focus is on crafting and developing the proposed system's modules. The development involves coding and database construction using Visual Studio Code IDE, PHP for the back end, and MySQL for the database. The front end, representing the user interface, is created using HTML,

CSS, and JavaScript. This phase brings the proposed system to life in accordance with established requirements and designs.

3.4 Testing Phase

In the Agile Model's concluding testing phase, rigorous tests are conducted to ensure the seamless functioning of the developed proposed system. Functional testing aims to identify glitches or defects, and immediate debugging measures are taken to rectify them. User Acceptance Testing is also performed to ensure the proposed system meets user expectations before delivery. This phase is pivotal in validating the alignment of the system with functional requirements established during the analysis phase.

4. Analysis And Design

System analysis and design encompass system requirements, use-case diagrams, sequence diagrams, activity diagrams, class diagrams, flowcharts, system architecture, entity relationship diagrams (ERD), and user interface design.

4.1 Functional and Non-functional Requirements

Requirement analysis involves identifying user expectations and needs for a completely new or modified system. This analysis is pivotal in system development, serving as a cornerstone for the system's success. System requirements can be broadly categorized into two types: functional requirements, non-functional requirements.

Table 2 *Functional Requirements*

Modules	Functionality
User management	• The system should allow all users to login into the system • The system should only allow admin to create accounts for users. • The system should allow admin, supervisors and upper managements to manage and update their account profile.
HIRARC module	• The system should allow upper managements or supervisors to create, remove, update and read HIRARC data. • The system should include audit trails for HIRARC data.
Incident reporting	• The system should only allow upper managements or supervisors to create, remove, update and read incident reports. • The system should provide an email notification to all users except admins when a new incident was registered via incident notification forms.
Safety Analysis and Reporting	• The system should allow users to generate safety analysis reports. • The system should perform the calculation necessary in order to generate the safety analysis reports.
Management Approval and Assessment	• The system should only allow project managers to approve HIRARC forms. • The system should only allow Safety Health Supervisor to verify HIRARC forms.

4.2 Use Case Diagram

A UML use case diagram is utilized to depict system/software requirements for a newly developed software program [9]. In the ISMS system, actors include Administrator, Supervisor, SHO and Project Manager. The administrator creates new users, while only Supervisor, SHO, and EHS managers can create HIRARC forms. Project Manager approves HIRARC forms. Incident notification is open to all users, but the responsibility lies with the supervisor. Monthly safety analysis is accessible to all users except the administrator.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1463-1482

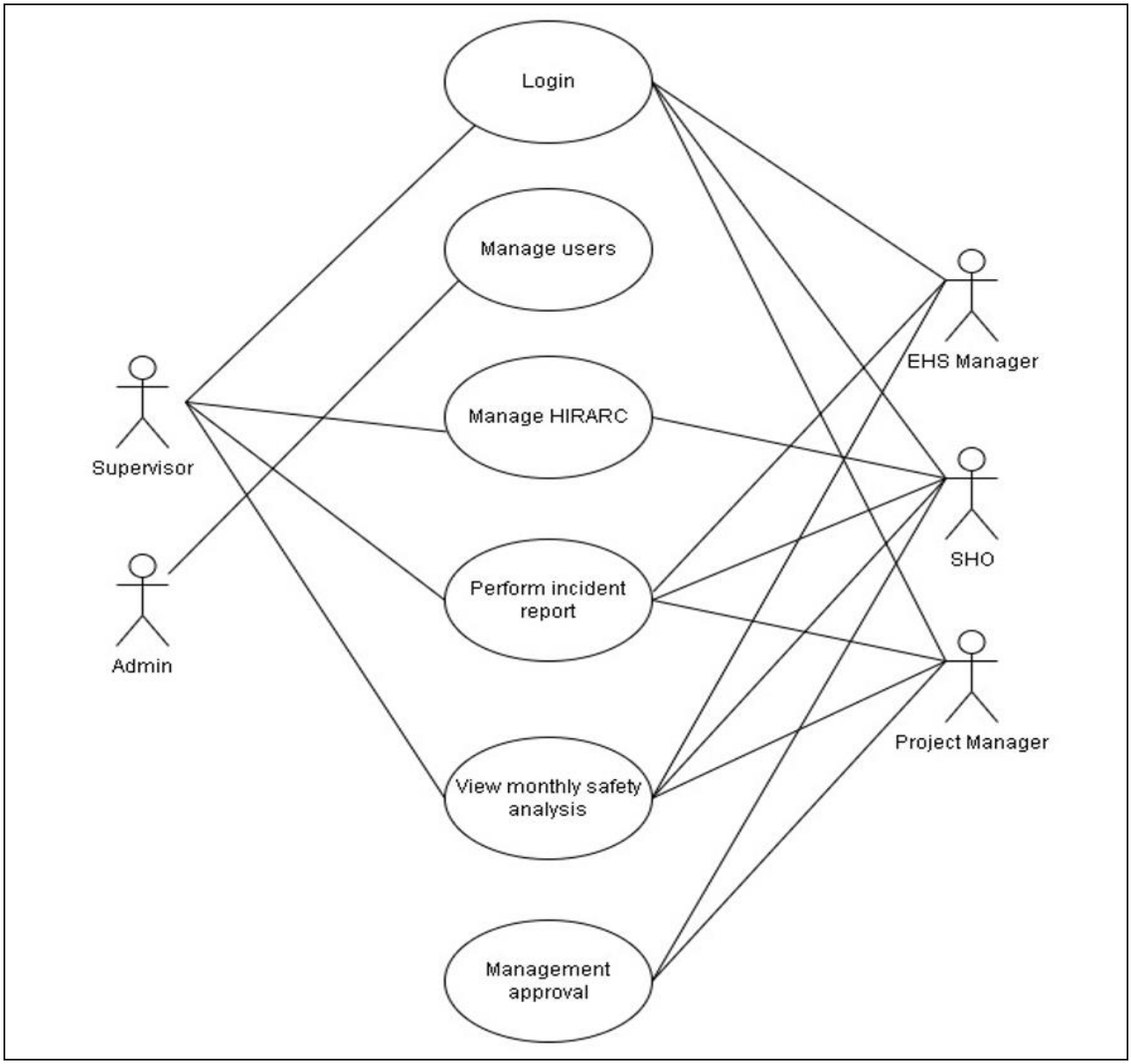


Fig. 5 *Use Case Diagram for ISMS*

4.3 Class Diagram

A class diagram delineates the attributes and operations associated with a class, along with the constraints imposed on the system [10]. It should be present in analysis and design to describe the attributes and operations of a class and also the constraints imposed on the proposed system. In Figure 5, fifteen classes are identified in the proposed systems which are User, Profile, Supervisor, SHO, Project Manager, ESH Manager, HIRARC, HIRARCreport, AuditTrail, MonthlySafetyAnalysis, IncidentNotification, InjuredPerson, Witness, and MonthlySafetyAnalysis.

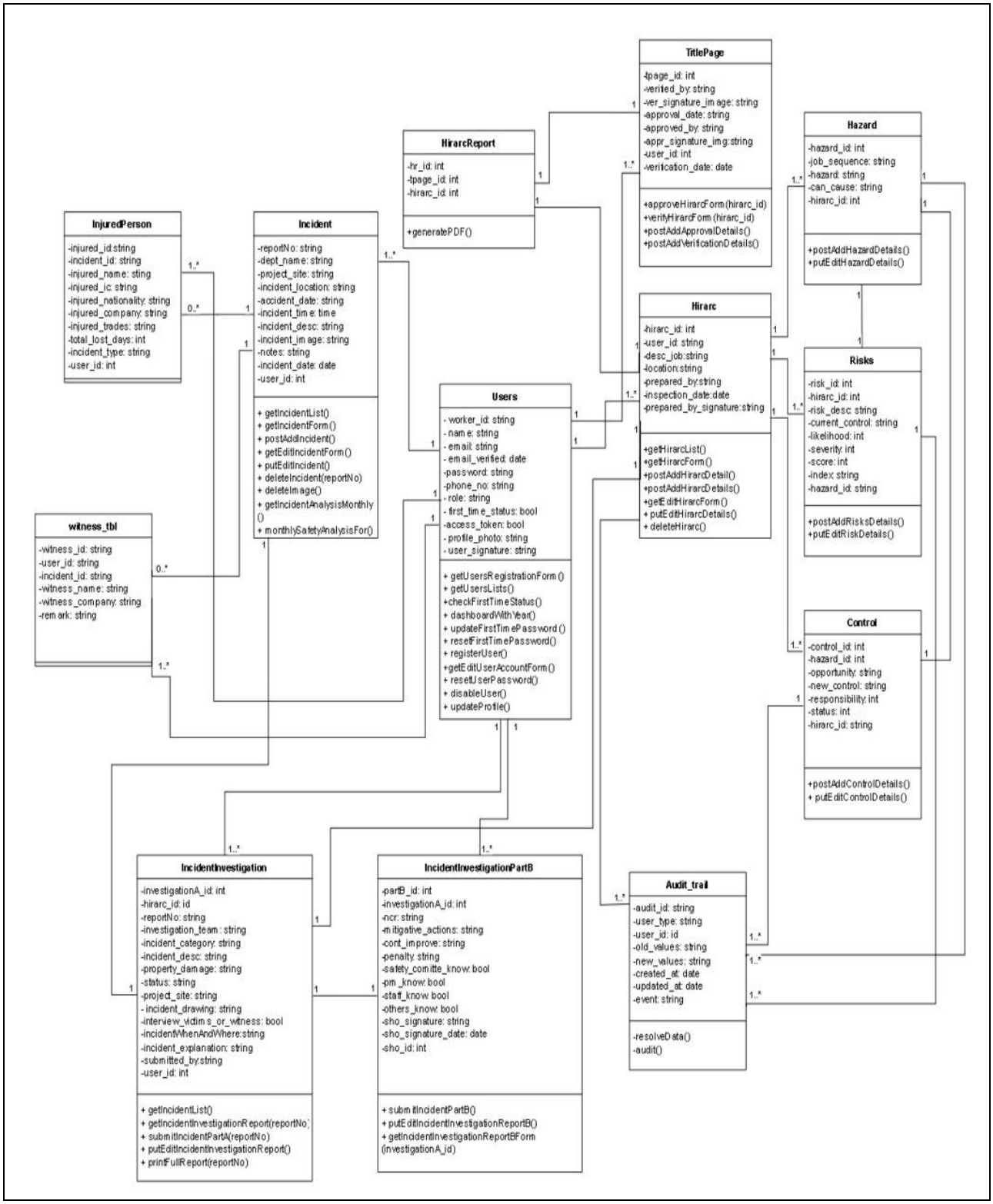


Fig. 6 *Class Diagram for ISMS*

4.4 System Architecture

System architecture serves as the conceptual model defining the views, structure, and behavior of a system, representing how it works and communicates with other components [9]. Figure 4.7 shows the system architecture of ISMS. All the users will interact with the ISMS and ISMS will interact with the database.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1463-1482

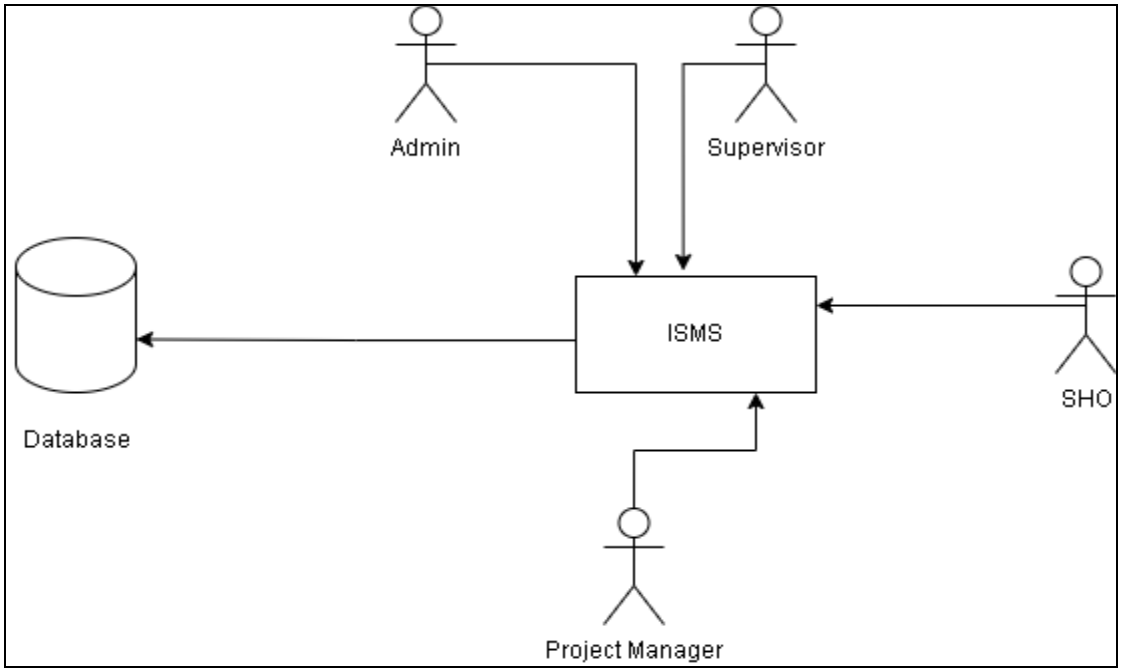


Fig. 7 *System Architecture Diagram for ISMS*

5. Results And Discussion

In this section, the result of implementation of Integrated Safety Management System are shown. This including the test cases result and the result from user acceptance testing.

5.1 Implementation

Based on **Fig 8**, a Laravel controller is used to code the account registration process. First of all, the code is written to receive a login request. Then, the system will authenticate the user to see if the user exists or not. If the user exists, it will create a new session ID to enhance security. And finally, it will redirect the user to the appropriate page.

```
class AuthenticatedSessionController extends Controller
{
    /**
     * Display the login view.
     */
    public function create(): View
    {
        return view('auth.login');
    }

    /**
     * Handle an incoming authentication request.
     */
    public function store(LoginRequest $request): RedirectResponse
    {
        $request->authenticate();

        $request->session()->regenerate();

        return redirect()->intended(RouteServiceProvider::HOME);
    }

    /**
     * Destroy an authenticated session.
     */
    public function destroy(Request $request): RedirectResponse
    {
        Auth::guard('web')->logout();

        $request->session()->invalidate();

        $request->session()->regenerateToken();

        return redirect('/');
    }
}
```

Fig. 8 *Account Login Source Code*

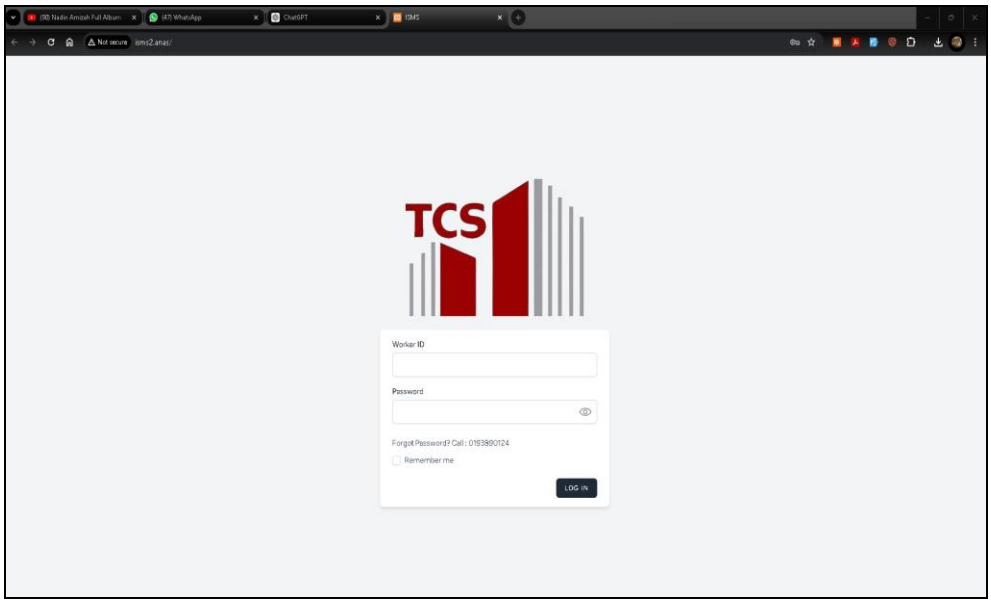


Fig. 9 *Account Login Interface*

The postAddHirarcDetails method as shown in **Fig 10**, handles adding new HIRARC details by first retrieving the current user's ID and creating a new Hirarc instance. It processes the prepared_by_signature input, saving the signature image if provided or using the user's existing signature. The method then sets various attributes on the Hirarc instance from the request data, saves the Hirarc instance, and associates it with a new HirarcReport. After saving both instances, it triggers a NewHirarcAdded event and returns a view named supervisor.hirarc-form with the new hirarc_id.

```php
public function postAddHirarcDetails(Request $request){
    $userId = auth()->user()->id;
    $hirarc = new Hirarc();
    if (isset($request['prepared_by_signature']) && !empty($request['prepared_by_signature'])){
        // dd('if');

        $base64ImageVerified = $request->input('prepared_by_signature');

        // Remove the data URL prefix (e.g., '')
        $base64ImageVerified = str_replace('', '', $base64ImageVerified);

        // Decode the base64-encoded image data
        $imageDataVerified = base64_decode($base64ImageVerified);

        // Generate unique file names for the images (e.g., using timestamp)
        $fileNameVerified = 'signature_prepared_by' . time() . '.png';

        // Specify the storage paths where the images will be saved
        $storagePathVerified = 'public/signatures/preparedby';
        // Ensure that the storage directories exist, create them if not
        if (!Storage::exists($storagePathVerified)) {
            Storage::makeDirectory($storagePathVerified);
        }
        // Store the image files in the specified storage paths
        Storage::put($storagePathVerified . '/' . $fileNameVerified, $imageDataVerified);
        // Get the full paths of the saved image files
        $filePathVerified = Storage::url($storagePathVerified . '/' . $fileNameVerified);

        $hirarc->prepared_by_signature = $filePathVerified;
    }else{
        $file_path = auth()->user()->user_signature;
        $hirarc->prepared_by_signature = $file_path;
    }
    $hirarc->desc_job = $request['desc_job'];
    $hirarc->location = $request['location'];
    $hirarc->prepared_by = $request['prepared_by'];
    // dd($request['inspection_date']);
    $hirarc->inspection_date = $request['inspection_date'];
    $tpage_id = $request['tpage_id'];
    // $hirarcReport = HirarcReport::where('tpage_id', $tpage_id)->first();
    $hirarcReport = new HirarcReport();

    $hirarc->save();

    $hirarc_id = $hirarc->hirarc_id;
    $hirarcReport->hirarc_id = $hirarc_id;
    $hirarcReport->save();
    event(new NewHirarcAdded($hirarc));
    // event(new NewHirarcAdded($hirarc));
    // // Redirect to the next page and pass the hirarc_id as a parameter
    // dd($hirarc_id);
    return view('supervisor.hirarc-form')->with('hirarc_id', $hirarc_id);
    // return redirect()->route('user.hirarc-form-view')->with('hirarc_id', $hirarc_id);
    // return redirect()->route('user.hirarc-form-view', ['hirarc_id' => $hirarc_id]);
```

Fig. 10 *Add Hirarc Source Code*

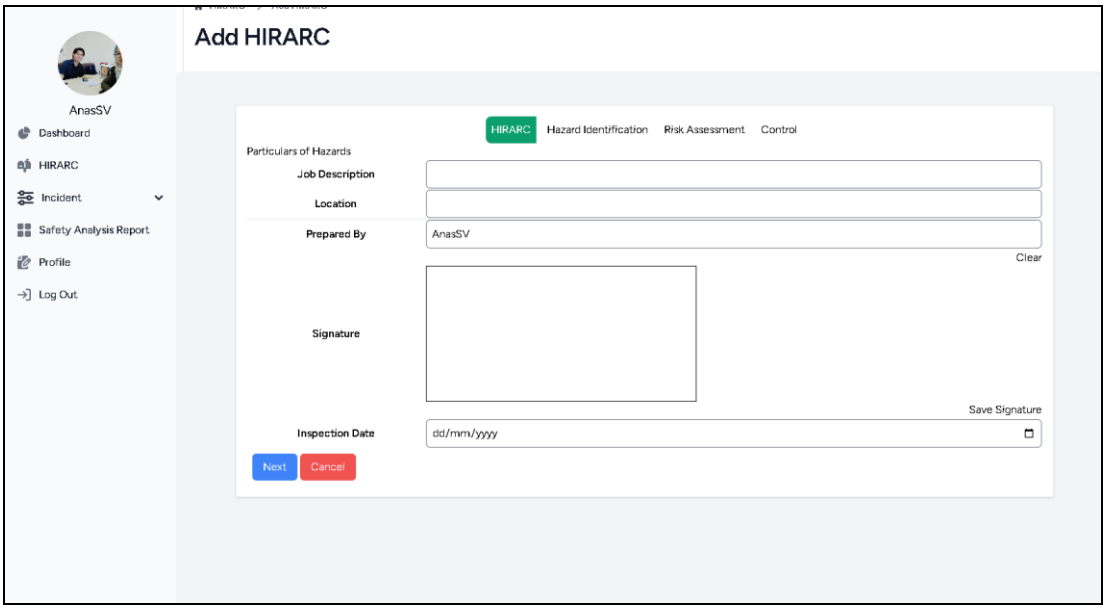


Fig. 11 *Add Hirarc Source Code*

Fig 12 shows the getIncidentList method retrieves all incidents and users, then iterates through each incident to fetch associated images from storage. It constructs URLs for these images, assigns the first image and all images to the incident object, and also attaches the user's name to each incident based on the user ID. Finally, it sets breadcrumb and heading variables and passes the incidents data to the supervisor.incidents-list view which is displayed at **Fig 13**.

```php
class IncidentController extends Controller
{
    public function getIncidentList(){
        $incidents = Incident::all();
        $users = User::all();
        foreach ($incidents as $incident) {
            $folderName = $incident->incident_image;

            // Get all files in the folder
            $files = Storage::disk('public')->files($folderName);

            // Initialize an array to store image URLs
            $incidentImages = [];

            // Loop through files and add to the array
            foreach ($files as $file) {
                // Prepend 'storage/' to the file path to make it accessible from the web
                array_push($incidentImages, asset('storage/' . $file));
            }
            // Assign the first image and all images to the incident object
            $incident->firstImage = count($files) > 0 ? $files[0] : null;
            $incident->incidentImages = $incidentImages;
            $user = User::find($incident->user_id);
            if($user){
                $incident->user_name = $user->name;
            }
        }
        $breadcrumb1 = "Incident";
        $headings = "Incidents List";
        return view('supervisor.incidents-list', compact('incidents', 'breadcrumb1', 'headings'));
    }
```

Fig. 12 *Incident List Source Code*

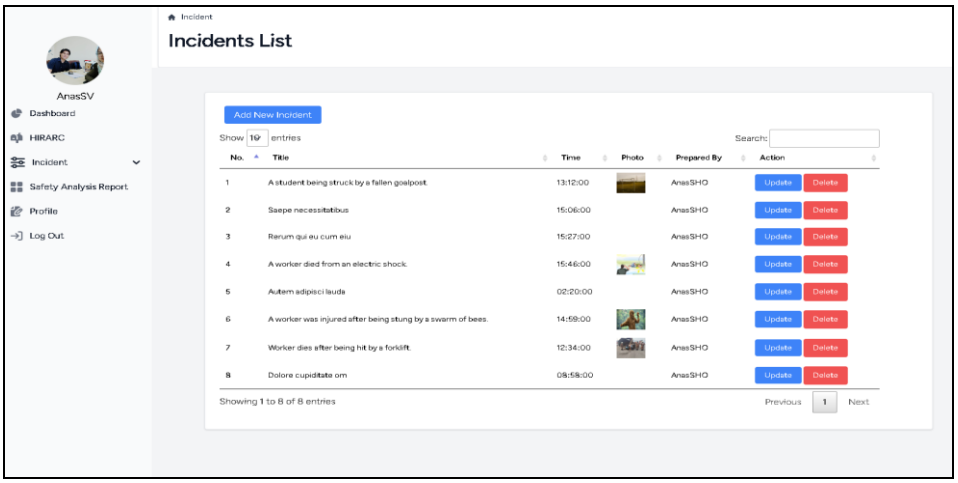


Fig. 13 *Incident List Interface*

Fig. 14 shows the monthlySafetyAnalysisFor function generates a safety analysis report for a specified month, aggregating data from the start of the year up to the selected month. It processes incident records to calculate metrics such as the number of fatality cases, lost time cases, non-lost time cases, near-miss cases, total days lost, and computes frequency and severity rates. The results are formatted and truncated to include only the most recent three months, along with a count of recorded incidents by users. This data, along with breadcrumb and heading information, is then passed to a view at **Fig. 15** for display.

```
public function monthlySafetyAnalysisFor(Request $request)
                $recordedBy[] = $username;

            }
        }

    }


    // Calculate total days lost for the current month
    foreach ($injuredPeople as $injuredPerson) {
        $totalDaysLost += $injuredPerson->total_lost_days;
    }
    // dd(count($recordedBy));
    // Calculate safety metrics for the current month
    // dd($fatalityCases);

    $numOfFatalityCasesCurrMonth = count($fatalityCases);

    $numOfLostTimeCasesCurrMonth = count($lostTimeCases);
    $numOfNonLostTimeCasesCurrMonth = count($nonLostTime);
    $frequencyRateCurrMonth = ($numOfLostTimeCasesCurrMonth * 1000000) / $totalManHoursWorked;
    $severityRateCurrMonth = ($totalDaysLost * 1000000) / $totalManHoursWorked;
    $numOfNearMissCasesCurrMonth = count($nearMissCases);
    $frequencyRateCurrMonth = number_format($frequencyRateCurrMonth, 2);

    // Format severity rate to 2 decimal places
    $severityRateCurrMonth = number_format($severityRateCurrMonth, 2);

    // Store safety analysis data for the current month in the result array
    $monthlySafetyAnalysis[] = [
        'month' => $currentDate->format('Y-m'),
        'numOfFatalityCases' => $numOfFatalityCasesCurrMonth,
        'numOfLostTimeCases' => $numOfLostTimeCasesCurrMonth,
        'numOfNonLostTimeCases' => $numOfNonLostTimeCasesCurrMonth,
        'totalDaysLost' => $totalDaysLost,
        'frequencyRate' => $frequencyRateCurrMonth,
        'severityRate' => $severityRateCurrMonth,
        'numOfNearMissCases' => $numOfNearMissCasesCurrMonth,
        'totalManHoursWorked' => $totalManHoursWorked,
        'averageWorkersPerDay' => $averageWorkersPerDay,
    ];
}
$recordedCounts = array_count_values($recordedBy);

for($i = 0 ; $i < (count($monthlySafetyAnalysis)) ; $i++){
    $monthlySafetyAnalysis[0]['numOfFatalityCases'] = $monthlySafetyAnalysis[0]['numOfFatalityCases'] + $monthlySafetyAnalysis[$i]['numOfFatalityCases'];
    $monthlySafetyAnalysis[0]['numOfLostTimeCases'] = $monthlySafetyAnalysis[0]['numOfLostTimeCases'] + $monthlySafetyAnalysis[$i]['numOfLostTimeCases'];
    $monthlySafetyAnalysis[0]['numOfNonLostTimeCases'] = $monthlySafetyAnalysis[0]['numOfNonLostTimeCases'] + $monthlySafetyAnalysis[$i]['numOfNonLostTimeCases'];
    $monthlySafetyAnalysis[0]['numOfNearMissCases'] = $monthlySafetyAnalysis[0]['numOfNearMissCases'] + $monthlySafetyAnalysis[$i]['numOfNearMissCases'];
    $monthlySafetyAnalysis[0]['totalDaysLost'] = $monthlySafetyAnalysis[0]['totalDaysLost'] + $monthlySafetyAnalysis[$i]['totalDaysLost'];
}

$monthlySafetyAnalysis[0]['frequencyRate'] = ($monthlySafetyAnalysis[0]['numOfLostTimeCases'] *1000000) / $totalManHoursWorked;
$monthlySafetyAnalysis[0]['severityRate'] = ($monthlySafetyAnalysis[0]['totalDaysLost'] *1000000) / $totalManHoursWorked;
$monthlySafetyAnalysis[0]['frequencyRate'] = number_format($monthlySafetyAnalysis[0]['frequencyRate'], 2);

// Format severity rate to 2 decimal places
$monthlySafetyAnalysis[0]['severityRate'] = number_format($monthlySafetyAnalysis[0]['severityRate'], 2);
$monthlySafetyAnalysis[1] = $monthlySafetyAnalysis[count($monthlySafetyAnalysis) - 2];
$monthlySafetyAnalysis[2] = $monthlySafetyAnalysis[count($monthlySafetyAnalysis) - 1];
$monthlySafetyAnalysis = array_slice($monthlySafetyAnalysis, 0, 3);
// dd($monthlySafetyAnalysis);

// dd($recordedCounts);
// Pass the truncated array to the view
$breadcrumb1 = "Monthly Analysis";
$headings = "Monthly Safety Analysis";
return view('supervisor.incident-analysis.incident-analysis-monthly')
->with('monthlySafetyAnalysis', $monthlySafetyAnalysis)
->with('recordedCounts', $recordedCounts)
->with('breadcrumb1', $breadcrumb1)
->with('headings', $headings);
```

Fig. 14 *Incident Analysis Source Code*

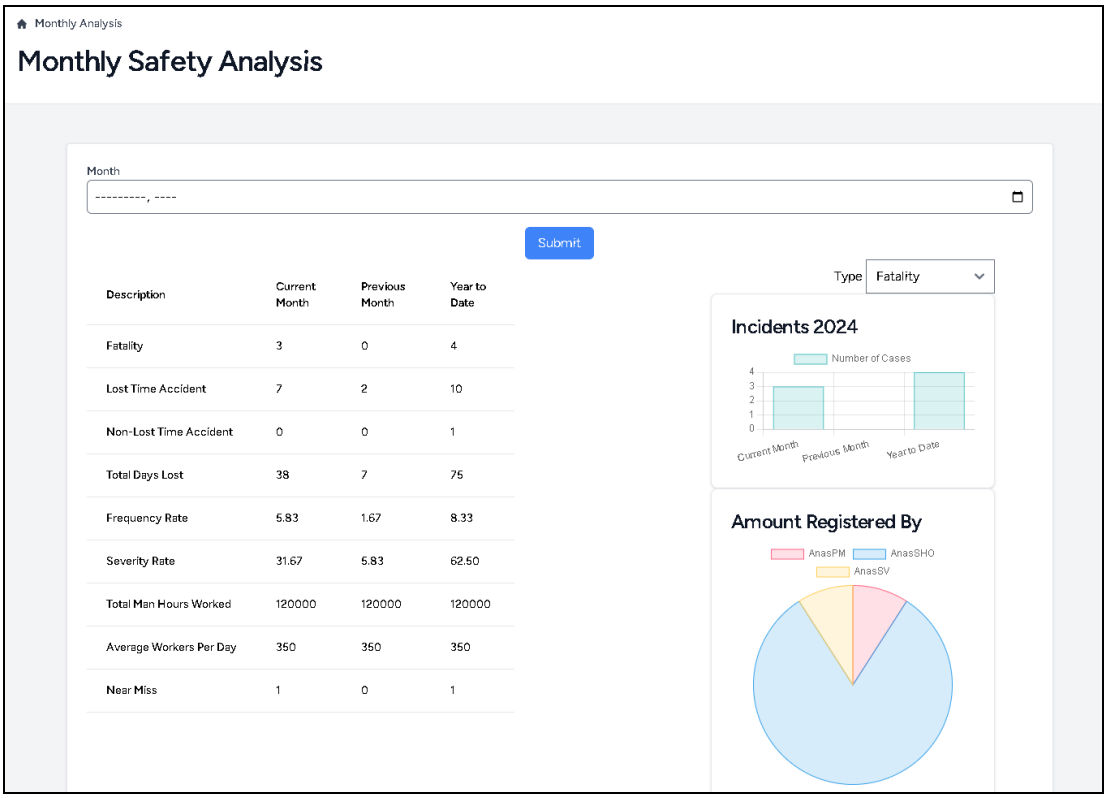


Fig. 15 *Incident Analysis User Interface*

Fig. 16 shows the postAddApprovalList which is used to fetch approval data from **Fig. 17**. The upper managements signature are required whenever they want to verify or approve forms. The signature is saved and will be fetched when the user want to download the completed HIRARC form.

```
public function postAddApprovalDetails(Request $request){

        // Specify the storage paths where the images will be saved
        $storagePathApproved = 'public/signatures/approvedby';

        // Ensure that the storage directories exist, create them if not

        if (!Storage::exists($storagePathApproved)) {
            Storage::makeDirectory($storagePathApproved);
        }

        // Store the image files in the specified storage paths
        Storage::put($storagePathApproved . '/' . $fileNameApproved, $imageDataApproved);

        // Get the full paths of the saved image files
        $filePathApproved = Storage::url($storagePathApproved . '/' . $fileNameApproved);
    } else {
        $filePathApproved = $request->input('approved_by_signature');
    }

    $hirarc_id = $request['hirarc_id'];
    $hirarcReport = HirarcReport::where('hirarc_id', $hirarc_id)->first();

    if($hirarcReport){
        // dd('comom');
        $titlePage = TitlePage::where('tpage_id', $hirarcReport->tpage_id)->first();
        if($titlePage){
            $titlePage->appr_signature_img = $filePathApproved;
            $titlePage->approval_date = $request['approval_date'];
            $titlePage->approved_by = $request['approved_by'];
            $titlePage->user_id = $userId;
            $titlePage->save();
            $hirarcReport->tpage_id = $titlePage->tpage_id;
            $hirarcReport->save();


        }else{
            $titlePage = new TitlePage();
            if($request->input('approved_by_signature')){

            }
            $titlePage->appr_signature_img = $filePathApproved;
            $titlePage->approval_date = $request['approval_date'];
            $titlePage->approved_by = $request['approved_by'];
            $titlePage->user_id = $userId;
            $titlePage->save();
            $hirarcReport->tpage_id = $titlePage->tpage_id;
            $hirarcReport->save();

        }
    }else{
        // dd('takda');
        $hirarcReport = new HirarcReport();
        $titlePage = new TitlePage();
        $titlePage->appr_signature_img = $filePathApproved;
        $titlePage->approval_date = $request['approval_date'];
        $titlePage->approved_by = $request['approved_by'];
        $titlePage->user_id = $userId;
```

Fig. 16 *HIRARC Approval Source Code*

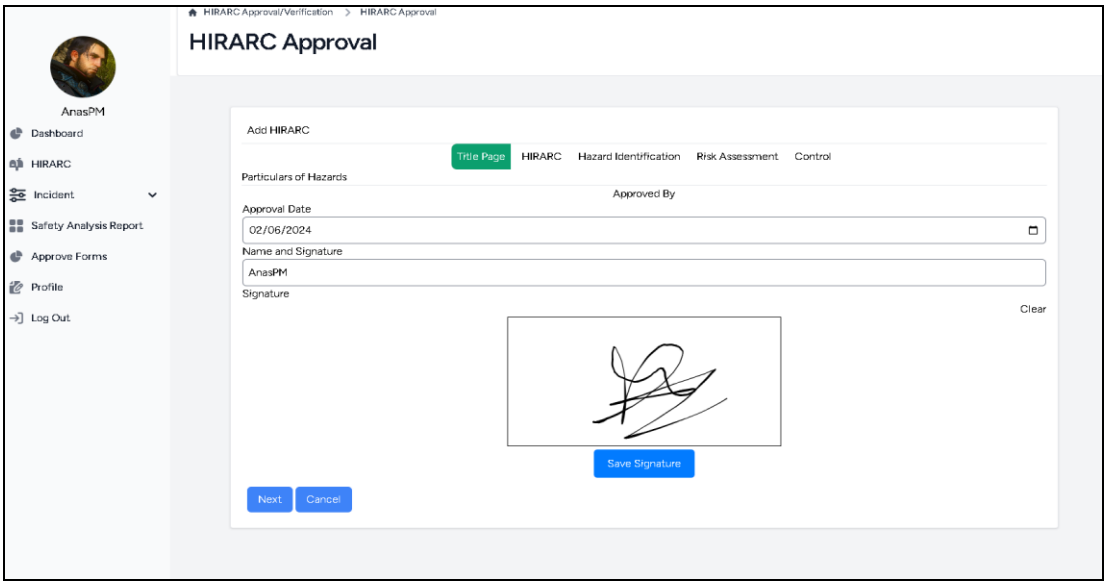


Fig. 17 *HIRARC User Interface*

Fig. 19 shows the Incident investigation form (Part A). This function submitIncidentPartA handles the submission of an incident investigation report. It begins by receiving the data submitted from **Fig. 18**. And then extracting and validating input data from the request, ensuring the investigation team leader is a registered user and that a HIRARC (Hazard Identification, Risk Assessment, and Risk Control) exists. It checks that mandatory fields like incident category, property damage, investigation findings, and status are provided. If validation fails, it redirects back with error messages. It handles file uploads for project site and incident drawings, storing them in specific directories. If validation passes and files are processed, it attempts to save the incident investigation data to the database. If successful, it redirects to a list page with a success message; otherwise, it redirects back with an error message indicating the failure reason.

```
class IncidentInvestigationController extends Controller
    public function submitIncidentPartA(string $reportNo, Request $request){
            $errors['status'] = 'Status cannot be empty!';
        }


        // If there are any errors, redirect back with errors
        if (!empty($errors)) {
            return redirect()
                ->route('incident-investigation-form', ['reportNo' => $reportNo])
                ->with('error', 'There are validation errors')
                ->withErrors($errors);
        }

        // Handle file uploads
        if ($request->hasFile('project_site')) {
            $folderName = 'incident-investigation/project_site/project_site_' . $reportNo . '_';
            $projectSiteFolderName = $folderName;

            foreach ($request->file('project_site') as $file) {
                $filename = $request->reportNo . '_' . uniqid() . '.' . $file->getClientOriginalExtension();
                $path = $file->storeAs($folderName, $filename, 'public');
            }
        }

        if ($request->hasFile('incident_drawing')) {
            $folderNameDrawing = 'incident-investigation/incident_drawing/incident_drawing' . $reportNo . '_';
            $incidentDrawingFolderName = $folderNameDrawing;

            foreach ($request->file('incident_drawing') as $file) {
                $filenameDrawing = $request->reportNo . '_' . uniqid() . '.' . $file->getClientOriginalExtension();
                $path = $file->storeAs($folderNameDrawing, $filenameDrawing, 'public');
            }
        }

        try {
            $investigation = new IncidentInvestigation();
            $investigation->hirarc_id = $request->hirarc_id;
            $investigation->reportNo = $request->reportNo;
            $investigation->investigation_team = json_encode($request->investigation_team);
            $investigation->incident_category = $request->incident_category;
            $investigation->incidentWhenAndWhere = $request->incidentWhenAndWhere;
            $investigation->incident_desc = $request->incident_desc;
            $investigation->property_damage = $request->property_damage;
            $investigation->investigation_findings = $request->investigation_findings;
            $investigation->status = $request->status;
            $investigation->incident_explanation = $request->incident_explanation;
            $investigation->project_site = $projectSiteFolderName;
            $investigation->incident_drawing = $incidentDrawingFolderName;
            if($request->interview_victims_or_witness){
                $investigation->interview_victims_or_witness = ($request->interview_victims_or_witness === "on") ? 1 : $request->interview_victims_or_witness;
            }
            $investigation->save();

            return redirect()->route('incident-investigation-list')->with('success', 'Incident investigation data saved successfully!');
        } catch (Exception $e) {
            return redirect()
                ->route('incident-investigation-form', ['reportNo' => $reportNo])
                ->with('error', 'Failed to save incident investigation data: ' . $e->getMessage())
                ->withErrors(['database' => 'An error occurred while saving the data.']);
        }
    }
```

Fig. 18 *Incident Investigation Source Code*

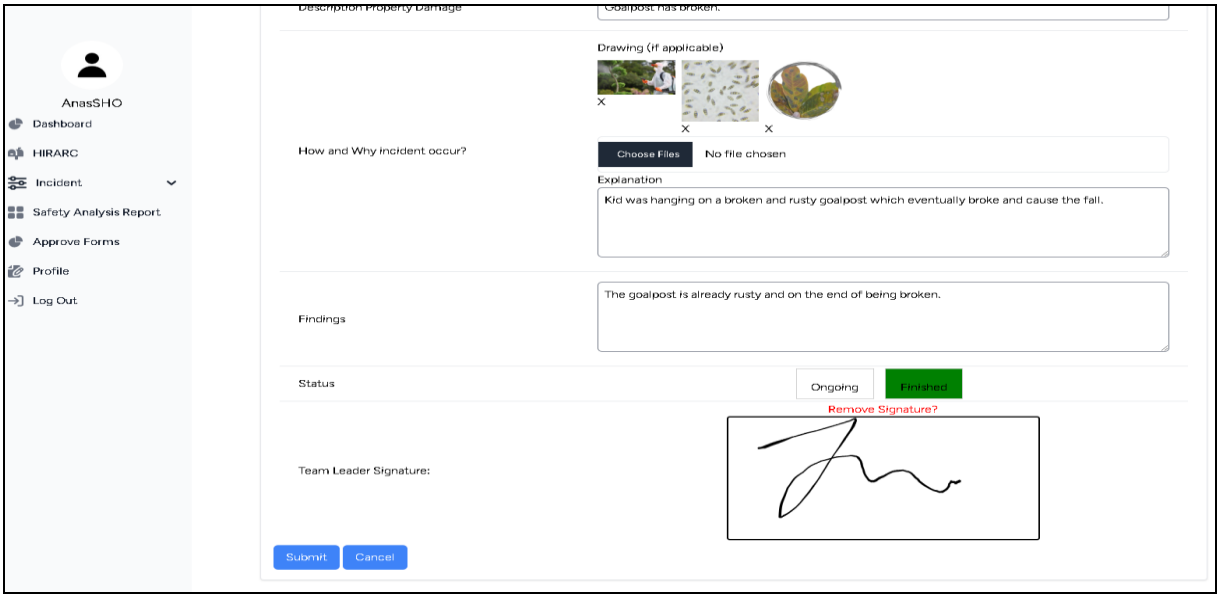
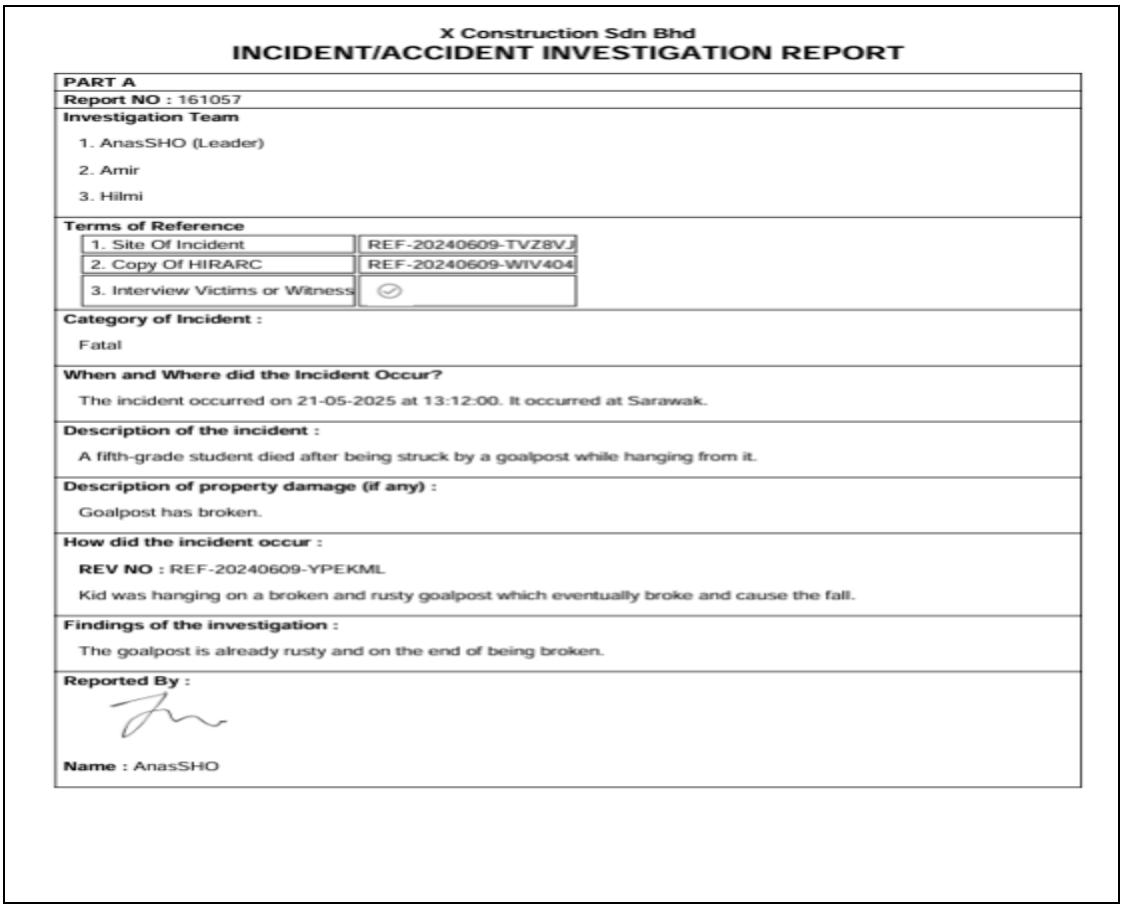


Fig. 19 *Incident Investigation User Interface*

Next, **Fig. 20** shows the incident investigation report. The report is only accessible once incident investigation report A and B are completed and signed by the SHO and Team leader.



X Construction Sdn Bhd

INCIDENT/ACCIDENT INVESTIGATION REPORT

PART A

Report NO : 161057

Investigation Team

1. AnasSHO (Leader)
2. Amir
3. Hilmi

Terms of Reference

1. Site Of Incident	REF-20240609-TVZ8VJ
2. Copy Of HIRARC	REF-20240609-WIV404
3. Interview Victims or Witness	✓

Category of Incident :

Fatal

When and Where did the Incident Occur?

The incident occurred on 21-05-2025 at 13:12:00. It occurred at Sarawak.

Description of the incident :

A fifth-grade student died after being struck by a goalpost while hanging from it.

Description of property damage (if any) :

Goalpost has broken.

How did the incident occur :

REV NO : REF-20240609-YPEKML

Kid was hanging on a broken and rusty goalpost which eventually broke and cause the fall.

Findings of the investigation :

The goalpost is already rusty and on the end of being broken.

Reported By :

Name : AnasSHO

Fig. 20 *Incident Investigation Report*

5.2 Testing

On the developed modules, functional testing is carried out. The purpose of this testing is to make sure the developed system functions exactly as the user needs. Test cases provide instructions on how to run tests on a program, system, or application. A test case is a distinct set of steps or directives that a tester must adhere to verify the functionality of a particular feature of a product or application.

Table 3 *Test Case for User Management Module*

Test Case ID	Test Cases	Expected Outcome	Actual Outcome	Pass/Fail
M1-1	To check whether administrator can register for an account	The user should be able to create for an account	The user has successfully created for an account	Pass
M1-2	To check whether an administrator and other user can login into the system and directed correctly.	The user should be able to login into the system and directed correctly	The user has successfully logged into the system and directed correctly.	Pass
M1-3	To check whether the system will restrict login whenever a wrong credential is entered	The system should restrict login when an incorrect credentials has been entered	The system restricted the login when an incorrect or no credentials has been entered	Pass
M1-4	To check whether the administrator could update the existing users' information	The administrator should be able to update the existing users' information	The administrator has successfully updated the users' information	Pass
M1-5	To check whether the administrator could disable existing user	The administrator should be able to disable existing user	The administrator has successfully updated disable existing user	Pass

Table 4 *Test Case for HIRARC Module*

Test Case ID	Test Cases	Expected Outcome	Actual Outcome	Pass/Fail
M2-1	To check whether supervisor and upper managements can insert a new HIRARC data.	Supervisor and upper managements should be able to insert a new HIRARC data.	Supervisor and upper managements have successfully created a new HIRARC data.	Pass
M2-2	To check whether supervisor and upper managements can search a specific HIRARC data with keywords.	Supervisor and upper managements should be able to search HIRARC data.	Supervisor and upper managements have successfully searched a specific HIRARC data.	Pass
M2-3	To check whether supervisor and upper managements can edit an existing HIRARC data.	Supervisor and upper managements should be able to edit an existing HIRARC data	Supervisor and upper managements have successfully edited a new HIRARC data.	Pass

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1463-1482

Table 4 *Test Case for HIRARC Module (cont)*

Test Case ID	Test Cases	Expected Outcome	Actual Outcome	Pass/Fail
M2-4	To check whether supervisor and upper managements can view changes made to an existing HIRARC data.	The administrator should be able to view changes made to an existing HIRARC data.	Supervisor and upper managements have successfully viewed changes made to an HIRARC data.	Pass

Table 5 *Test Case for Incident Module*

Test Case ID	Test Cases	Expected Outcome	Actual Outcome	Pass/Fail
M3-1	To check whether supervisor and upper managements can insert a new incident data.	Supervisor and upper managements should be able to insert a new incident data.	Supervisor and upper managements have successfully created a new incident data.	Pass
M3-2	To check whether supervisor and upper managements can search a specific incident data with keywords.	Supervisor and upper managements should be able to search incident data.	Supervisor and upper managements have successfully searched a specific incident data.	Pass
M3-3	To check whether supervisor and upper managements can edit an existing incident data.	Supervisor and upper managements should be able to edit an existing incident data	Supervisor and upper managements have successfully edited a new incident data.	Pass
M3-4	To check whether supervisor and upper managements can delete an existing incident data.	Supervisor and upper managements should be able to delete an existing incident data.	Supervisor and upper managements have successfully deleted an incident data.	Pass
M3-5	To check whether supervisor and upper managements can receive an email notification about a new incident.	Supervisor and upper managements should be able to receive an email notification about a new incident.	Supervisor and upper managements have successfully received an email on an incident data.	Pass

Table 6 *Test Case for Monthly Safety Analysis Module*

Test Case ID	Test Cases	Expected Outcome	Actual Outcome	Pass/Fail
M4-1	To check whether supervisor and upper managements can view monthly safety analysis.	Supervisor and upper managements should be able to view safety analysis data.	Supervisor and upper managements have successfully viewed a safety analysis data.	Pass
M4-2	To check whether the calculations made are correct.	Supervisor and upper managements should be able to view correct data.	Supervisor and upper managements have verified that the data are correctly calculated.	Pass

Table 6 *Test Case for Monthly Safety Analysis Module (cont)*

Test Case ID	Test Cases	Expected Outcome	Actual Outcome	Pass/Fail
M4-3	To check whether supervisor and upper managements can view a visualisation of incidents by type and the person that registered it.	Supervisor and upper managements should be able to view a visualisation of incidents by type and the person that registered it.	Supervisor and upper managements have successfully viewed a visualisation of incidents by type and registered by.	Pass

Table 7 *Test Case for Management Approval Module*

Test Case ID	Test Cases	Expected Outcome	Actual Outcome	Pass/Fail
M5-1	To check whether upper managements can view all HIRARC forms and its approval and verification status.	Supervisor and upper managements should be able to view all HIRARC forms and its approval and verification status.	Supervisor and upper managements have successfully viewed all HIRARC forms and its approval and verification status.	Pass
M5-2	To check whether Project Manager can approve HIRARC forms.	Project Manager should be able to approve HIRARC forms.	Project Manager have approved HIRARC forms.	Pass
M5-3	To check whether SHO can verify HIRARC forms	SHO should be able to verify HIRARC forms.	SHO have verified HIRARC forms	Pass
M5-4	To check whether upper managements can download completed HIRARC report in PDF format.	Upper managements should be able to download completed HIRARC report in PDF format.	Upper managements have downloaded HIRARC report in PDF format.	Pass

5.3 User Acceptance Test

During the User Acceptance Testing (UAT) phase of software development, the system is assessed by end users to determine if it satisfies the objectives and requirements. This is the final phase of testing before the system is put into production. The purpose of UAT is to verify that the system meets the criteria and is both functional and user-friendly. In this testing round, only the shop owner and a staff member were involved. The system was tested by the supervisor, SHO and project manager and a friend who was currently working as a software developer was acting as the administrator. Comments and suggestions from the user acceptance test were gathered and will be utilized to enhance the system in the future. **Fig. 21** shows the bar chart of the UAT result.

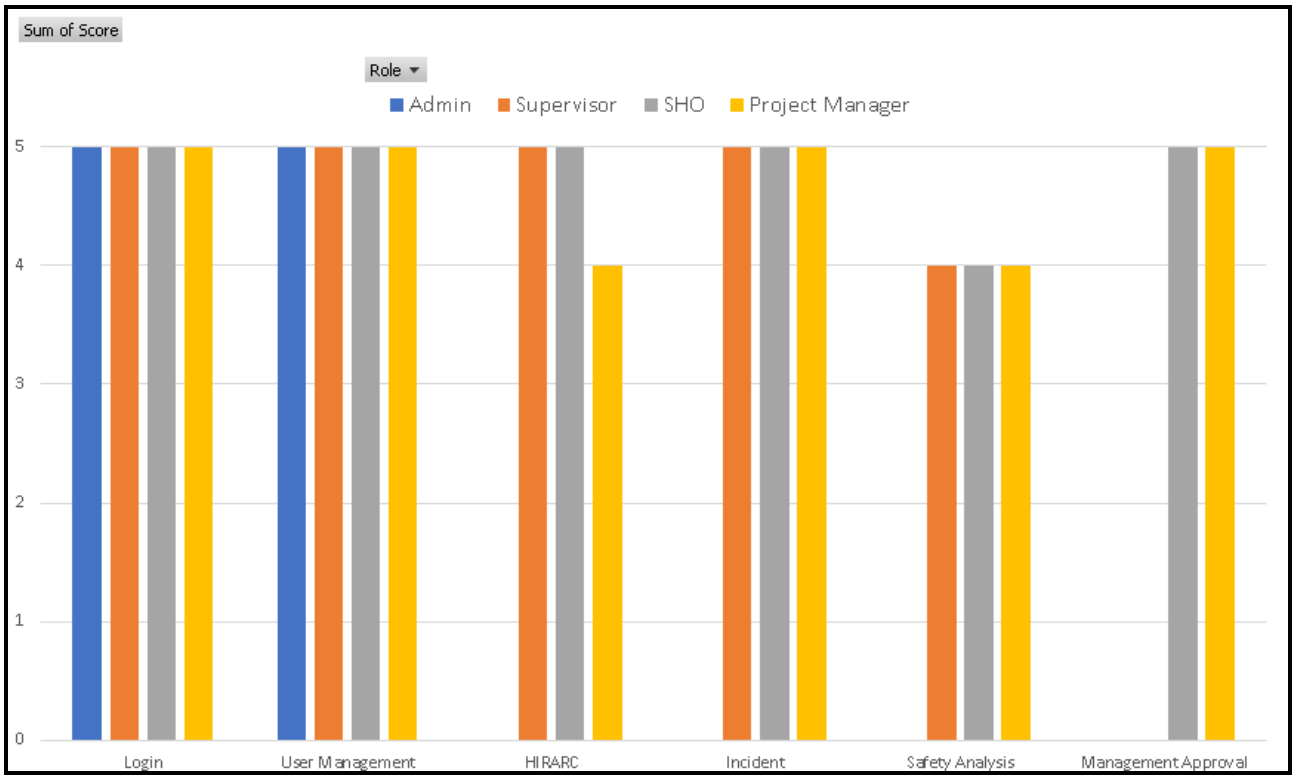


Fig. 21 *UAT Results*

6. Conclusion

In summary, the development of Integrated Safety Management System (ISMS) offers significant advantages for construction and hazard-prone companies by digitizing information, facilitating easy access to safety forms and incident details, enabling safety analysis, and ensuring accountability through audit trails and digital signatures. However, the system has limitations, such as the lack of hazard control measure recommendations, dependency on data from human resources, the time-consuming process of data migration, a learning curve for non-digital-savvy employees, limited customization, and the need for regular maintenance. Future enhancements should focus on integrating hazard control recommendations, improving data integration with human resources, implementing better data migration tools, providing extensive user training, expanding customization options, and ensuring regular updates. Despite its limitations, the ISMS has been successfully developed and holds potential for further improvement to better meet company needs and enhance efficiency.

Acknowledgement

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

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

[1] Francis K.W. Wong et al., "Findings from a research study of construction safety in Hong Kong," Journal of Engineering, Design and Technology, vol. 7(2), no. 130–142, pp. 1–15, 2009.
[2] Julie Anderson, "Accident Definition," Britannica.

[3] Jona Tarlengco, “Incident Report Guide: All You Need To Know,” https://safetyculture.com/topics/incident-report/.
[4] B. Aninditya, D. Dinariana, and F. Suryani, “IMPLEMENTATION OF HIRARC METHOD ON ERECTION GIRDER WORK OF SOUTH JAPEK II TOLL ROAD CONSTRUCTION PROJECT PACKAGE 3.” [Online]. Available: https://injurity.pusatpublikasi.id/index.php/in
[5] SafetyCulture, “Risk Assessment,” https://safetyculture.com/topics/risk-assessment/.
[6] William W. Cato and R. Keith Mobley, “Definition of a CMMS.”.
[7] Terry Wireman, Computerized Maintenance Management Systems. Industrial Press Inc, 1994.
[8] McCormick, M. (2012). "Waterfall vs. Agile Methodology." MPCS, N/A.
[9] Visual Paradigm, "What is Use Case Diagram?" Visual Paradigm, [Online]. Available: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/.
[10] TutorialsPoint. "UML - Class Diagram," Tutorialspoint. [Online]. Available: https://www.tutorialspoint.com/uml/uml_class_diagram.htm.
[11] Safesite, "Best-in-Class Safety Management System & Safety App," Safesite. [Online]. Available: https://safesitehq.com/. [Accessed: 11-Jun-2024].
[12] SafetyCulture, "Improve Workplace Operations," SafetyCulture. [Online]. Available: https://safetyculture.com/. [Accessed: 11-Jun-2024].
[13] DNV, "Synergi Life Risk Management," DNV. [Online]. Available: https://www.dnv.com/software/services/qhse/synergi-life-risk-management-index/. [Accessed: 11-Jun-2024].