UNIVERSITI TEKNIKAL MALAYSIA MELAKA

MULTIMEDIA DATABASE

BITP3353

SEM 2

SESSION 2024/2025

MAINTENANCE REPORT SYSTEM WITH IMAGE-BASED EVIDENCE

GROUP MEMBERS:

NAME	MATRICS NUMBER
AFIFAH SYAZA HUDA BINTI AHMAD	B032210273
NUR FATHEHAH BINTI MOHD ARIS	B032210139
NUR AMIERA BADRIESYIA BINTI ZAINAL	B032210379
NUR ELESA AQILAH BINTI ABDUL RAHMAN	B032310231
NUR LYANA ATHIRAH BINTI MOHD FADLY	B032420116

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Introduction of this project

In any modern organization, from a university campus to a corporate building, the effective management of facilities is crucial for ensuring both operational continuity and safety. However, the conventional methods for reporting maintenance issues, which often depend on cumbersome paperwork, word-of-mouth instructions, or scattered email threads, are frequently inefficient. Such outdated processes can result in significant delays, misunderstandings between staff and maintenance teams, and a lack of reliable records. Consequently, minor problems (e.g., leaking pipe or broken chairs) can be overlooked, potentially escalating into more severe and costly failures.

To address these shortcomings, this project proposes the development of a Maintenance Report System with Image-Based Evidence. This web-based system is designed to enhance the entire maintenance reporting workflow. It will enable facilities staff to submit reports that include not only textual descriptions, but also vital multimedia evidence, such as photographs and short video clips of the issue. By providing administrators with clear, visual information, the system aims to eliminate ambiguity, facilitate faster diagnosis, and enable more accurate allocation of repair resources. Ultimately, the goal is to establish a transparent, accountable, and highly efficient communication bridge between the individual reporting issues and the administration responsible for resolving them.

1.2 Project Motivation

The motivation for this project is rooted in a common frustration, which is the inefficiency of traditional facility management. In many organizations, the process for a staff member to report a simple issue (e.g., broken light, leaking faucet) is surprisingly difficult. It can involve tracking down the right person, filling out paper forms, or sending an email to a general inbox where it might get overlooked. This

hassle often discourages staff from reporting minor issues promptly, which leaves administrators without a clear, real-time picture of their facility's condition.

We realized that a dedicated system allowing a user to simply snap a photo or record a quick video of an issue could fundamentally change the reporting process. It would not only make reporting faster and more convenient but also significantly improve the accuracy of the information received. Therefore, our primary driver for this project is to build a practical, user-friendly tool that closes this communication gap and delivers tangible benefits like faster repair times, better record-keeping, and a less frustrating experience for everyone involved.

1.3 Problem statements

The current manual or semi-automated methods for reporting maintenance issues are plagued by several key problems that this project aims to solve:

1. Inefficient and slow reporting process
   Reliance on paper forms, phone calls, or emails creates significant delays. Reports can be misplaced, forgotten, or not reach the correct person on time, hindering a swift response.

2. Ambiguity and misinterpretation issues
   Text-only descriptions are often subjective and can be misinterpreted. A report stating the "air conditioner is not working" lacks crucial details. Is it making a noise? Leaking water? Not turning on at all? This ambiguity forces administrators to perform a preliminary inspection, wasting their valuable time.

3. Poor documentation and archiving
   Manual records are difficult to store, search, and analyse. This makes it challenging to identify recurring problems, track the maintenance history of specific assets, or generate reports for auditing and planning purposes.

1.4 Objective of this project

To address the problems identified above, this project has the following primary objectives:

1. To develop a web-based system for staff to submit and for administrators to manage maintenance reports.

2. To enable users to upload multimedia evidence, specifically images and short video clips, along with textual descriptions, to provide clear and comprehensive details of an issue.

3. To provide an administrative dashboard that allows for efficient viewing of all submitted reports and updating of repair statuses.

1.5 Multimedia data and relevance

This project is fundamentally a multimedia database application, and the integration of specific media elements is crucial to its success. The primary multimedia data types to be used are images and short video clips, supported by status alerts.

- Images

An image serves as powerful, unambiguous evidence. A photograph can instantly convey the exact location and natural physical defect, such as broken pieces of equipment or leaking parts, or piping. This visual proof eliminates the guesswork inherent in text-only reports and helps administrators assess the urgency and required resources more accurately, often without needing an initial site visit.

- Short video clips

Videos are indispensable for capturing issues that involve motion, sound, or intermittent behaviour. For example, a flickering light, an air conditioner making disruptive noise, or a dripping faucet cannot be fully represented by a

static image. A short video clip provides dynamic context that is crucial for diagnosing such problems correctly.

- Status alerts

While not a traditional media type, status alerts are a key component of the system's interactive experience. These alerts provide real-time feedback to the user, confirming report submission and providing updates on its progress. This use of interactive feedback is a form of media that closes the communication loop and enhances user engagement.

The relevance of these media data is directly tied to the project's core objective of improving clarity and efficiency. As supported by research facilities management, visual data significantly reduces diagnostic time and miscommunication. Our system builds on this principle by integrating both image and video to create a rich, informative, and actionable reporting tool.

CHAPTER 2: PROJECT SCOPE

2.1 Introduction of this project scope

The scope of this project is precisely defined to ensure the development of a functional and effective system within a feasible time frame. The maintenance report system with image-based evidence is designed primarily as a specialized tool for the reporting and initial management of facility maintenance issues. The core focus is on bridging the communication gap between those who identify problems and the administrators who manage the response.

The system's scope encompasses two distinct user roles and is structured into several key functional modules. The boundaries of the project are set to include user authentication, multimedia-enhanced report submission, status tracking and administrative oversight. Any functionality beyond this, such as financial tracking, inventory management for the repair parts, or detailed technician scheduling, is considered outside the scope of this project.

2.2 Scope of the user

The system is designed to be used by three primary groups of users, each with distinct roles and access privileges that correspond to their function within the maintenance workflow.

2.2.1 Facilities staff

This user group represents the primary reporters of the system. They are general staff, employees, lecturers, or any individual within the organization who is authorized to report a maintenance issue. They do not require administrative privileges. Their role is to identify and document problems they encounter in the facility. Their capabilities within the system are focused on submission and tracking, and include:

- Login
  Securely access the system using their unique credentials

- Submit a maintenance report
  Create and submit a new report, including a title, a detailed text description or via speech recognition and the location of the issue.

- Upload multimedia evidence
  Attach images or short video clips to their report to provide a clear,visual context of problem

- View own submitted reports
  Access a personal history of the reports they have submitted to check their status..

2.2.1 Administrators

This user group consists of individuals responsible for overseeing and managing the maintenance process, such as a facilities manager, maintenance supervisor, or administrative head. They have elevated permissions to manage all incoming reports and control the workflow. Their role is to review, prioritize, and update the status of issues. Their capabilities include all the functions of a facilities staff member, plus:

- View all maintenance reports
  Access a dashboard displaying all reports submitted by all users.

- Update repair status
  Change the status of any report (e.g., ‘Pending’ to ‘Assigned’). This action is the primary trigger for the status alerts.

- Review report history
  Access the full history of a report, including when it was submitted, who updated it, and when status changed.

- staff account management
  admin can manage staff whether they want to add staff account , delete staff account or edit staff account and know who user still has an active account.
- technician account management
  admin can manage technicians whether they want to add technician account delete technician account or edit technician account and know who user still has an active account.

2.2.2 Technician

This user group is composed of technical personnel assigned to resolve the reported maintenance issues. They are responsible for inspecting, diagnosing, and addressing facility problems as assigned by the administrator. Technicians have operational access limited to the reports that are specifically assigned to them. Their capabilities within the system focus on task management and status reporting, and include:

- Login

  Securely access the system using their individual technician credentials.

- View assigned reports

  Access a list of maintenance reports that have been assigned to them by the administrator. Each report includes a description, location, multimedia attachments, and relevant details necessary for completing the repair task.

- Update repair status

  Provide real-time updates on the progress of their assigned tasks such as changing status from "In Progress" to "Completed". These status updates will notify the original reporter of the maintenance issue.

- Add resolution notes

  Document the solution or repair steps taken to resolve the issue. This ensures transparency and allows future reference in the report history.

- Technician dashboard

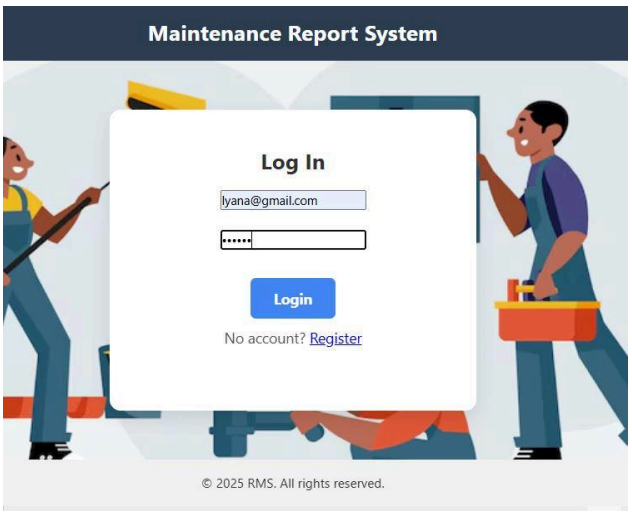
  Technicians can always monitor and manage all assigned maintenance tasks including quick tools to view assignments,archived reports,and maintenance guide.

2.3 Modules of the project

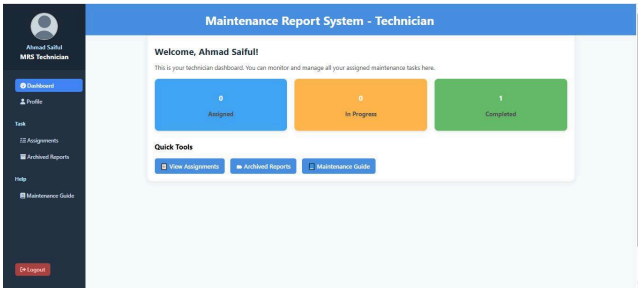
2.3.1 User authentication module

This module is fundamental to the system's security and ensures that users can only access features appropriate to their role. Its primary function is to manage user access. It will handle the login process for both facilities staff, administrators and technicians authenticating their credentials against the database. It will also enforce role-based access control, preventing regular staff from accessing the administrative dashboard and its associated functions.

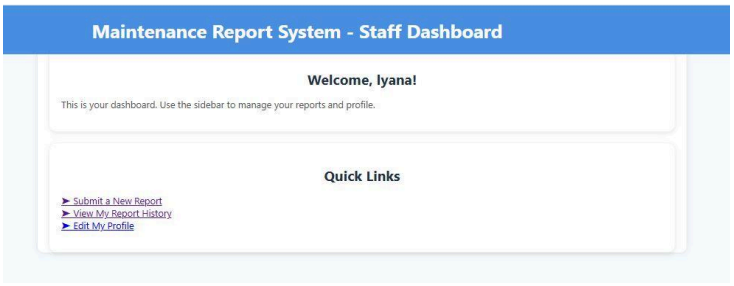
- login



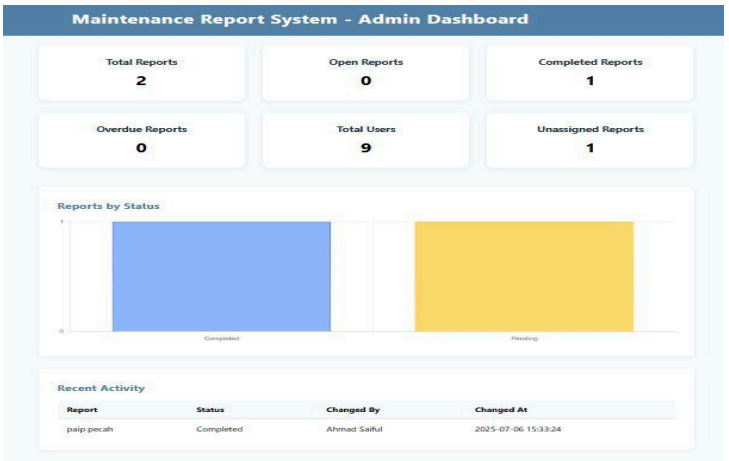
- Technician homepage



- staff homepage



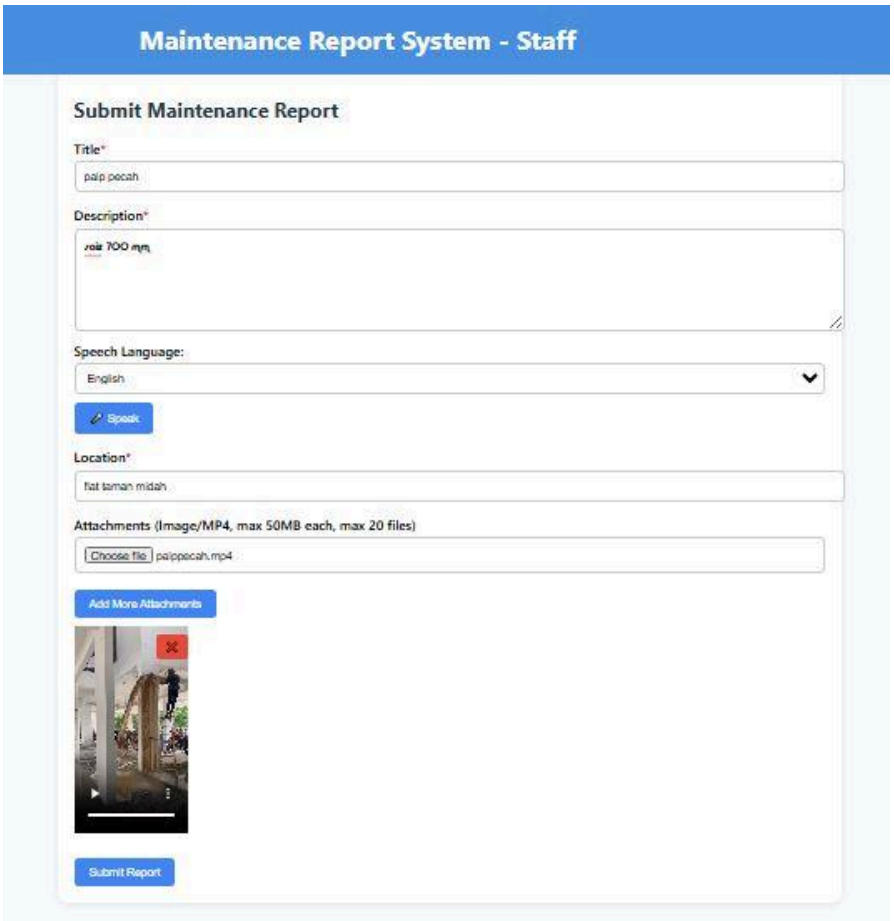
- admin homepage



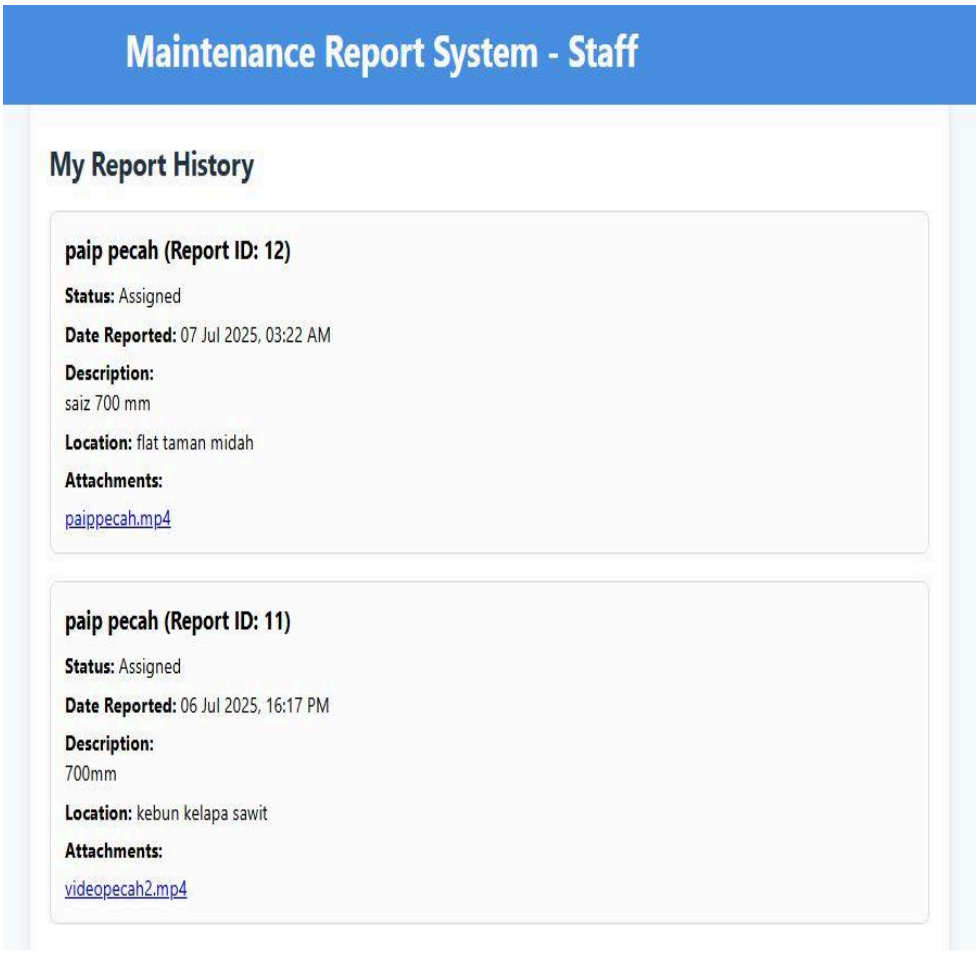
2.3.2 Report submission module

This is the core module for the facilities staff user. It provides the interface for creating and submitting a new maintenance request. This module will include a user-friendly form with fields for a report title, a detailed description, and the issue's location. The most critical feature of this module is the functionality to upload multimedia files (images and short videos) from the user's device, which will then be associated with the textual report and also have speech recognition features.

- report form submission



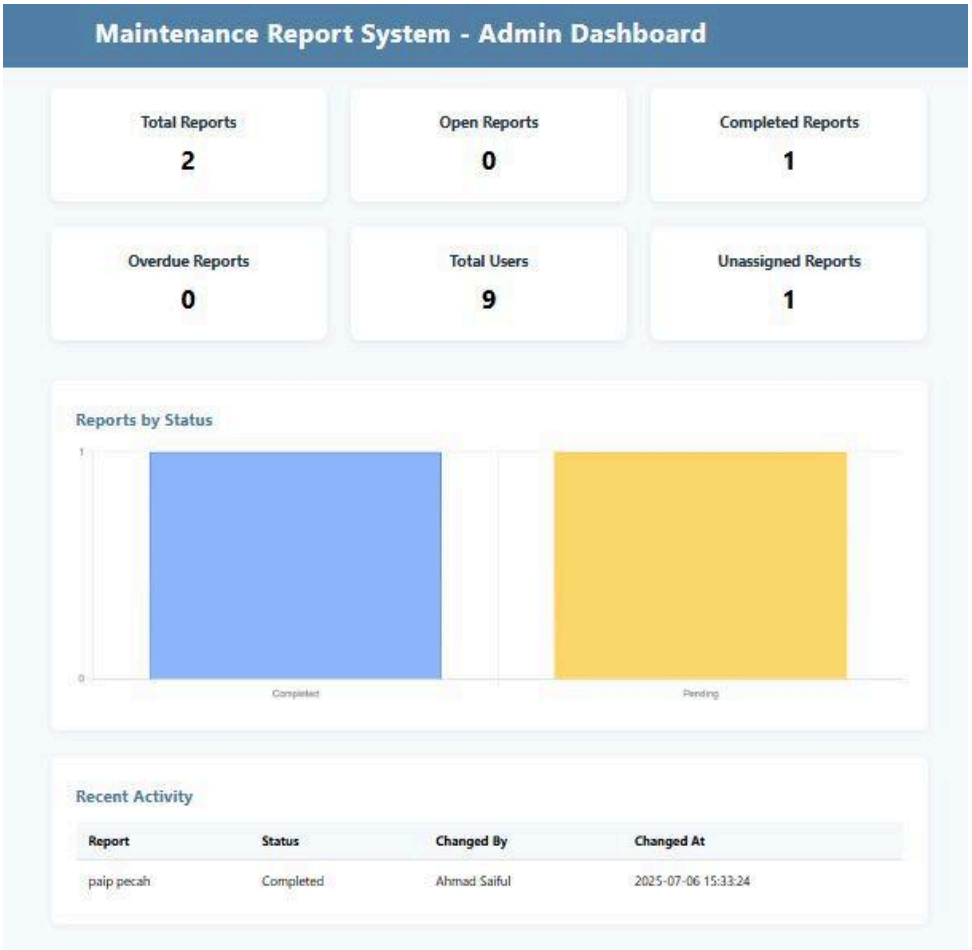
- view report history



2.3.2 Administrative and Management Module

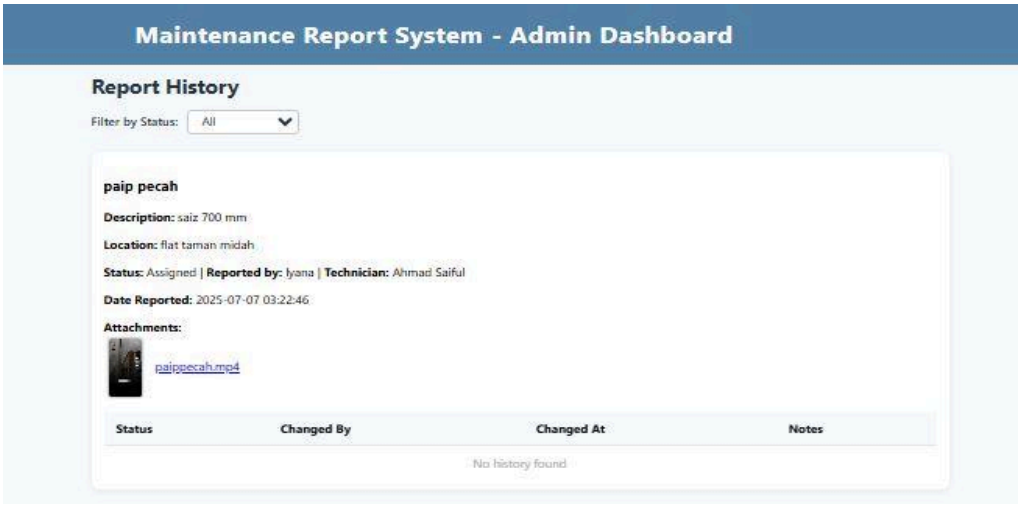
This module serves as the central command center for the administrator. It is presented as an administrative dashboard that provides a comprehensive, real-time view of all maintenance reports in the system. From this dashboard, the administrator can view report details, including any attached media evidence. A key function within this module includes the ability to update the status of a report and to archive reports that have been resolved, ensuring the active queue remains manageable. This module is essential for the management and operational control of the maintenance workflow.

- Admin dashboard



- Assign report based on technician's specification.

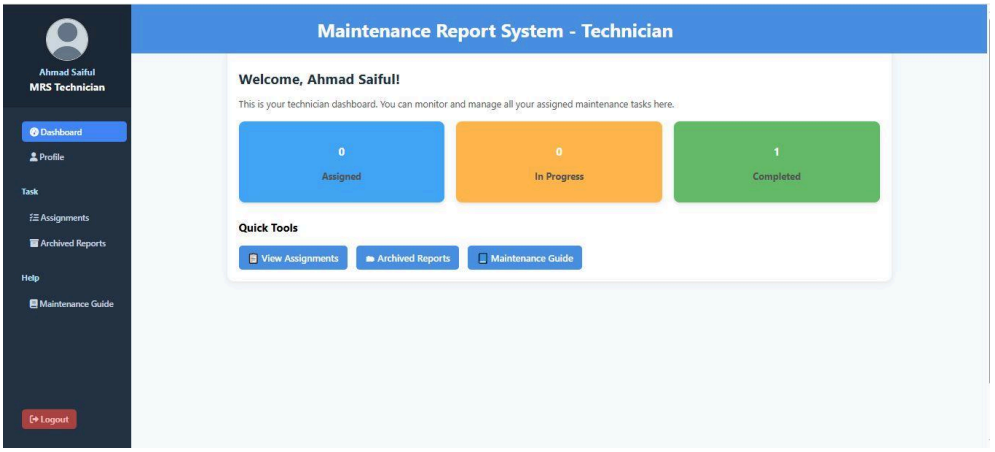
- View report history



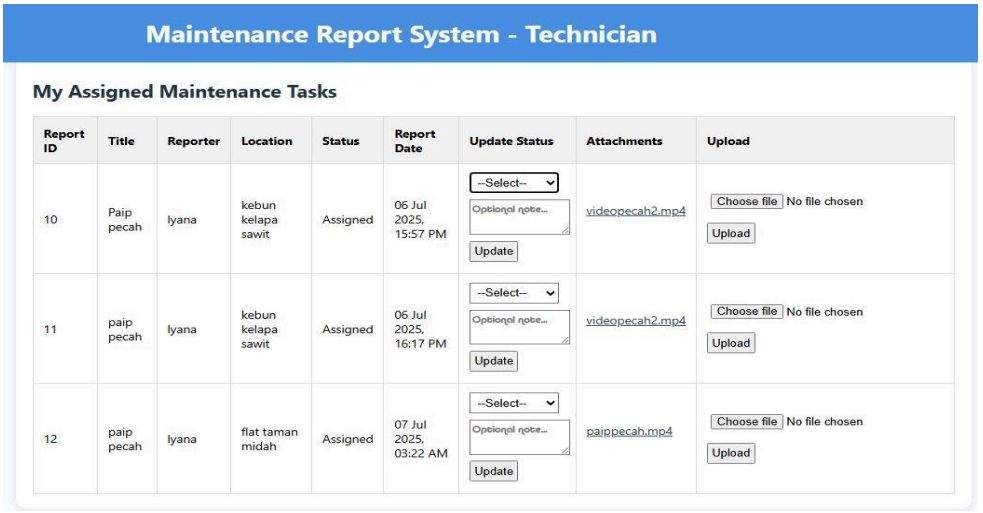
2.3.3 Technician Module

Technician Module serves as a central hub for technicians to efficiently manage their assigned tasks within a building complaint system. It allows technicians to receive work assignments in real-time, review detailed complaint information (such as issue descriptions, location, and attached media), and update the status of each task as it progresses—from "In Progress" to "Completed".

- Technician dashboard for overview their daily workload



- View assigned report for update status,view details of report like attachment media and the location from reporter and update picture after complete work.



CHAPTER 3: ENTITY-RELATIONSHIP DIAGRAM

3.1 Entity-relationship diagram (ERD) description

The database is the backbone of the maintenance report system with image-based evidence. Its design is crucial for ensuring data integrity, scalability, and efficient retrieval of information. The entity-relationship diagram (ERD) visually represents the logical structure of the database, detailing the entities (tables), their attributes (columns), and the relationships between them.

3.1.1 Explanation

The ERD for this project, shown below, consists of five core entities: SYS_USER, USER_ROLE, USER_REPORT, ATTACHMENT, and REPORT_HIST. This structure is designed to capture all necessary information, from user roles and report submissions to the multimedia evidence and status history associated with each report.

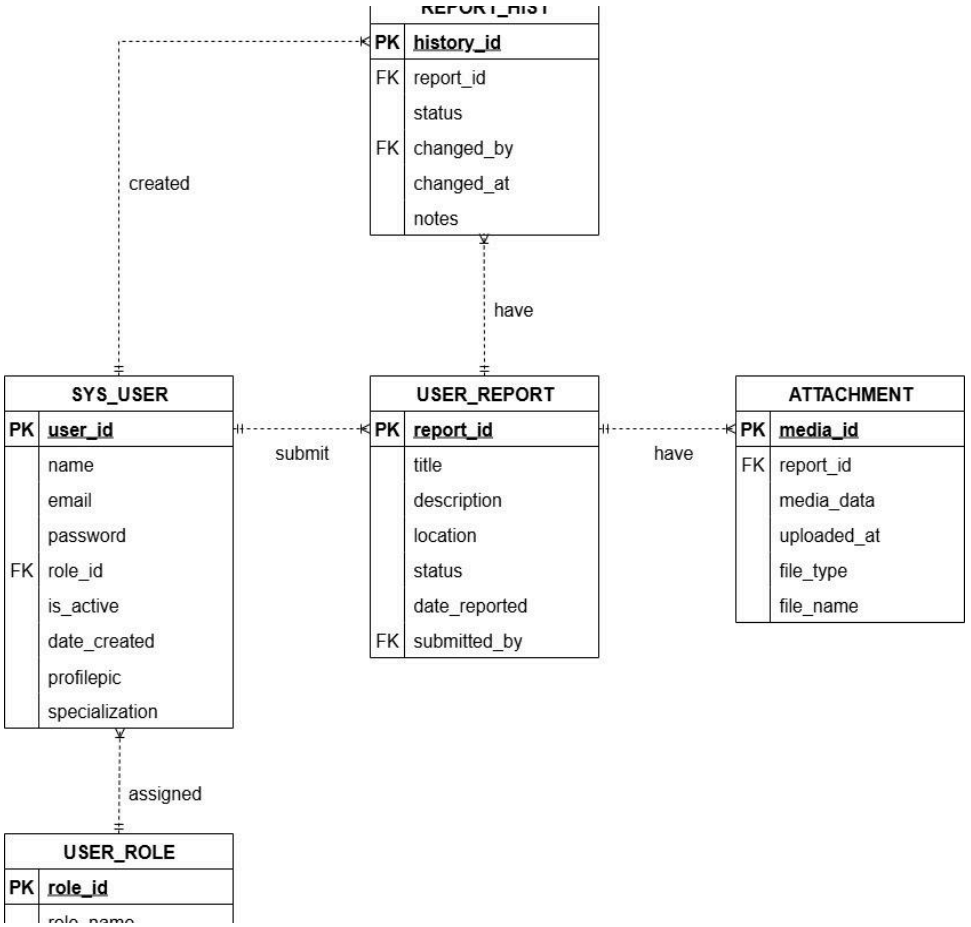


Figure 3.1: Entity-relationship diagram (ERD)

Business rules:

The design of the ERD is governed by a set of business rules that dictate how the data is related and stored:

1. Each user in SYS_USER must be assigned exactly one role from USER_ROLE. One role can be assigned to many users.

2. A registered SYS_USER can submit one or many maintenance reports (USER_REPORT). Each report is submitted by exactly one user.

3. Each USER_REPORT has a corresponding history of history status changes, which are recorded in the REPORT_HIST table. A report can have many records.

4. Each USER_REPORT can have one or more ATTACHMENT (multimedia evidence). Each attachment belongs to exactly one report.

5. Each record in REPORT_HIST must be linked to the SYS_USER (typically an administrator) who made the status change.

Data dictionary:

The following tables define the attributes for each entity in the database

field name	data type	descriptions	constraints
user_id	int	unique identifier for each user	PK
name	varchar(128)	full name of the user	NOT NULL
email	varchar(128)	user's email, used for login	UNIQUE, NOT NULL
password	varchar(256)	hashed password	NOT NULL
role_id	int	links to user's role	FK (USER_ROLE)
is_active	tinyint(1)	indicates if the user account is active	DEFAULT TRUE
date-created	datetime	datetime when the user account is created	
specialization	varchar(50)		NULLABLE

Table 1: SYS_USER data dictionary

field name	data type	descriptions	constraints
role_id	int	unique identifier for each role	PK
role_name	varchar(128)	name of the role	UNIQUE, NOT NULL

Table 2: USER_ROLE data dictionary

field name	data type	descriptions	constraints
report_id	int	unique identifier for each maintenance report	PK
title	varchar(128)	a brief title for the report	NOT NULL
description	text	a detailed description of the issue	
location	text	location of the issue	NOT NULL
status	varchar(64)	current status of the report	DEFAULT (PENDING)
date_reported	datetime	datetime of when the report is submitted	
submitted_by	int	id of the user who submitted the report	FK (SYS_USER)

Table 3: USER_REPORT data dictionary

field name	data type	descriptions	constraints
media_id	int	unique identifier for each media file	PK
report_id	int	id of the report this attachment belongs to	FK (USER_REPORT)
media_data	longblob	media file itself or path file	NOT NULL
uploaded_at	datetime	datetime of when the media is uploaded	
file_type	varchar(255)	type of file	
file_name	varchar(100)	original name of the uploaded file	

Table 4: ATTACHMENT data dictionary

field name	data type	descriptions	constraints
history_id	int	unique identifier for each history entry	PK
report_id	int	id of the report this history belongs to	FK (USER_REPORT)
status	varchar(64)	new status was set	
changed_by	int	id of user who changed the status	FK (SYS_USER)
changed_at	datetime	datetime when the status changed	
notes	text	optional notes from the admin about the status changed	NULLABLE

Table 5: REPORT_HIST data dictionary

3.2 Metadata examples

Metadata, or “data about data”, is essential for describing and managing the information within our system. It can be categorized into several types. Below are the examples demonstrating associative, navigational, and schema metadata relevant to this project.

3.1.1 Associative Metadata

Associative metadata links data from different tables to provide meaningful context. This is typically achieved using JOIN operations in SQL.

- Example 1: Retrieving all reports submitted by a specific user.

  This query associates a user with all the reports they have created.

  SQL QUERY:

  ```
  SELECT u.name, r.report_id, r.title, r.status
  FROM SYS_USER u
  JOIN USER_REPORT r ON u.user_id = r.submitted_by
  WHERE u.email = 'afifah@gmail.com';
  ```

  Example result:

name	report_id	title	status
afifah binti nasir	2	pintu terkunci	Resolved
afifah binti nasir	4	leaking pipe in the toilet	Submitted

- Example 2: Finding all media files for a specific report

  This query associates a report with its multimedia evidence

  SQL QUERY:

  SELECT r.title, a.file_name, a.file_type, a.uploaded_at

  FROM USER_REPORT r

  JOIN ATTACHMENT a ON r.report_id = a.report_id

  WHERE r.report_id = 2;

  Example result:

title	file_name	file_type	uploaded_at
pintu terkunci	null	null	2025-06-10 02:09:4

- Example 3: Displaying the full history of a report, including admin's name

  This query associates a report's history with the name of the administrator who made the change.

  SQL QUERY:

  SELECT h.changed_at, h.status, h.notes, u.name AS admin_name

  FROM REPORT_HIST h

  JOIN SYS_USER u ON h.changed_by = u.user_id

  WHERE h.report_id = 2

  ORDER BY h.changed_at;

Example result:

changed_at	status	notes	admin_name
2025-06-10 02:16:40	In Progress	Report updated via admin panel	afifah binti nasir

3.1.2 Navigational Metadata

Navigational metadata helps users find their way through a collection of data. In our system, a user might first find a report and then "navigate" to its associated evidence. This query simulates that path.

- Example 1: Finding the evidence for a report title "leaking pipe in the toilet"

  This demonstrates navigating from a known report detail to its associated files.

  SQL QUERY:

  SELECT a.file_name, a.file_type

  FROM ATTACHMENT a

  WHERE a.report_id = (SELECT report_id FROM USER_REPORT WHERE title = 'leaking pipe in toilet');

  Example result:

file_name	file_type
leak_pipe.jpg	image.jpeg

3.1.3 Schema Metadata

Schema metadata is data about the database structure itself, such as table definitions, column types, and constraints. This is useful for developers and database administrators.

- Example 1: Describing the structure of the USER_REPORT table.

  This query retrieves the schema information for a specific table, showing its columns and their properties.

  SQL QUERY (MySQL):

  DESCRIBE USER_REPORT;

  Example result:

filed	type	null	key	default	extra
report_id	int(11)	no	PRI	null	auto_increment
title	varchar(128)	no		null	
description	text	no		null	
location	text	no		null	
status	varchar(64)	default		Pending	
date_reported	datetime	default		current_timestamp	
submitted_by	int	no	MUL	null	

CHAPTER 4: SYSTEM MANUAL

4.1 Instructions of using the project

This chapter serves as a comprehensive manual for the maintenance report system with image-based evidence. It is divided into two main sections. The first section provides technical instructions for developers or evaluators on how to set up the project environment, download the source code from its GitHub repository, and launch the application locally. The second section is a user guide that explains how the intended users, facilities staff and administrators can navigate and interact with the system's features to perform their respective tasks.

4.2 Installation and setup guide (from GitHub)

This guide outlines the necessary steps to deploy and run the project from its source code. The project is hosted on GitHub, and a working internet connection is required to clone the repository.

4.2.1 Prerequisites and required software

Before proceeding with the installation, ensure the following software is installed on your local machine:

- Git: A version control system required to clone the project repository to GitHub.
- Web server environment: A local server stack such as XAMPP. This provides Apache (web server), MySQL (database), and PHP (server-side scripting language).

- Web browser: A modern web browser like Google Chrome, or Microsoft Edge to access applications.

- Code editor: A text editor such as Visual Studio Code or Sublime Text for editing configuration files if necessary.

4.2.2 Step 1: Clone the repository and set up files

First, you need to download the project source code from GitHub and place it in the web server's directory.

1. Open your command line interface (CLI) or Git Bash.

2. Navigate to your web server's public directory. For XAMPP, this is typically C:\XAMPP\HTDOCS.

3. Clone the repository using the following command:
   git clone https://github.com/your-username/capybara.git

4. This will create a new folder named capybara inside your htdocs directory containing all the projects files.

4.2.3 Step 2: Database setup

The system requires a database to store user information, reports, and media metadata.

1. Start the Apache and MySQL services from your XAMPP control panel.

2. Open your web browser and navigate to http://localhost/phpmyadmin/.

3. Create a new database. Click on “new” in the left sidebar, enter a database name (complain), and click “create”.

4. Select the newly created database. Click on the “Import” tab.

5. Click “Choose File” and locate the .sql database file provided within the cloned project folder.

6. Click “Go” at the bottom of the page to execute the SQL script. This will create all the necessary tables (SYS_USER, USER_REPORT, etc.) and populate them with any initial data (e.g., user roles).

4.2.4 Step 3: Configuration and launch

Finally, connect the application code to your newly created database.

1. Using your code editor, navigate to the project folder (htdocs/capybara) and find the database configuration file (e.g., conn.php, config.php, or db_connect.php).

2. Open the file and update the database connection details (hostname, username, password, and database name) to match your local setup.
   //Example
   $db_host = ‘localhost’;
   $db_user = ‘mmdb’;
   $db_pass = ‘mmdb1’;
   $db_name = ‘complain’;

3. Save the configuration file.

4. You can now launch the application by navigating to http://localhost/capybara/ in your web browser

4.3 User guide for application

This section explains how to use the deployed application from the perspective of the two main user roles.

4.3.1 For Facilities Staff: Reporting an issue

Facilities staff are the primary users who will be submitting reports.

1. Login
    a. Navigate to the application's login page
    b. Enter your assigned email and password
    c. Click the "Login" button to access your user dashboard

2. Submit a new maintenance report
    a. From your dashboard, click on the submit new report button
    b. You will be directed to a form. Fill in the required fields:
        i. Title: A short, clear title for the issue (e.g., Air Conditioner is not cooling").
        ii. Location: The specific location of the problem (e.g., "MPD2")
        iii. Description: A detailed explanation of the issue.

3. Upload image or video evidence
    a. On the same form, use the “Upload Files” or “Attach Evidence” button.
    b. Select one or more images or short videos from your device that clearly show the problem.
    c. Once all the details and files are added, click “Submit Report”.

4. View submitted reports and status
    a. After logging in, you can access a “Reports” page.
    b. This page will list all the reports you have submitted and their current status (e.g., Submitted, Pending, In Progress, or Resolved)

4.3.2 For Administrators: Managing Reports

Administrators have full oversight of the system and are responsible for managing the reports.

1. Login and access the admin dashboard
    a. Log in using your administrator credentials
    b. Upon successful login, you will be directed to the Admin Dashboard, which provides a complete overview of all reports submitted to the system.

2. Review new reports and evidence
    a. The dashboard will list all reports, typically with the newest ones at the top.
    b. Click on any report to view its full details, including the description, location, and the name of the staff who submitted it.

c. Click on the attached media files to view the images or play the video evidence to better understand the issue.

3. Update report status
    a. From the report detail view, you will have options to manage the report.
    b. Use the dropdown menu or buttons to change the status (e.g., Pending to In Progress).
    c. You can add notes in the text box to provide more information about the update (e.g., “Technician assigned, eta 2 hours”).
    d. Click “Update Status”.

4. Archive resolved reports
    a. Once the repair is complete, change the status to resolved
    b. Resolved reports will be moved to an archive using the “Archive” button. This will keep the dashboard clean and focused on active issues while preserving the record for future reference.

4.3.3 For Technicians: Handling Assigned Reports

Technicians are responsible for resolving issues assigned by administrators. They receive maintenance reports, update progress, and document the repair process.

1. Login and access technician dashboard
    a. Navigate to the login page and enter your technician credentials.
    b. After a successful login, you’ll be directed to the Technician Dashboard.
    c. The dashboard will display all reports assigned to you, where you can monitor and manage all your assigned maintenance

tasks including those marked as In Progress, Completed and Assigned.

2. View assigned reports and details
    a. From the sidebar navigation, click on “Task”, then select “Assignment” or just click on view assignments.
    b. This page will list all reports assigned to you:
        i. Report Title and Description
        ii. Exact Location of the issue
        iii. The reporter (staff) ID
        iv. Update status progress
        v. Any attached images or videos for better context

3. Update report progress
    a. After inspecting the problem on-site, update the report status:
        i. For example, change from “Assigned” to “In Progress”
    b. You can add notes to explain the situation or your next steps (e.g., “Parts ordered, repair scheduled tomorrow”)
    c. Click “Update Status” to save the changes

4. Upload completion proof
    a. Once the repair is completed, you may upload images or short video clips as proof.
    b. Use the “Choose Files” button in the report view to upload media evidence then click button upload.

5. Archive resolved reports
    a. After completing the task, change the status to “Completed” and it will redirect to Archived Reports.
    b. This will remove the report from the active assignment list and keep your dashboard clean while keeping a permanent record of the job.

CHAPTER 5: CONCLUSION

5.1 Summary of the project

The maintenance report system with image-based evidence was developed to address the inherent inefficiencies and ambiguities found in traditional facility maintenance reporting. The project's primary goal was to replace outdated, manual processes with enhanced digital platforms. By leveraging web technologies, the system successfully created a modern workflow where facilities staff can submit detailed reports enriched with crucial visual evidence, such as images and short videos. This provides administrators with the clear, actionable information needed to diagnose issues accurately, prioritize tasks effectively, and manage the entire repair functional prototype that effectively demonstrates the value of integrating multimedia into facility management.

5.2 Achievement of the project

Throughout the development process, the project remained focused on its core objectives. We are pleased to report that all the primary objectives set out in Chapter 1 have been successfully met:

1. A web-based system was developed, providing distinct interfaces for both facilities staff (for submission) and administrators (for management), thereby creating a single source of truth for all maintenance issues.

2. The system's core feature, enabling users to upload multimedia evidence (images and videos), was successfully implemented. This directly addresses the problem of report ambiguity.

3. An administrative dashboard was designed and implemented, allowing administrators to view all submitted reports, inspect evidence, and manage the workflow by updating the status of each report.

5.3 Project strength

The developed system exhibits several key strengths that make it a significant improvement over traditional methods:

- Enhanced clarity and accuracy: The use of image and video evidence is the system's most significant strength. It virtually eliminates the guesswork involved in interpreting text-only reports, leading to faster and more accurate diagnostics.

- Improved efficiency: By digitizing and centralizing the reporting process, the system drastically reduces the time and effort required to report an issue and for an administrator to process it.

- Increased transparency and accountability: Both staff and administrators have clear visibility into a report's lifecycle. The status alerts and history log ensure that reporters have a clear audit trail for every action taken.

5.4 Final conclusion

In conclusion, the maintenance report system with image-based evidence projects has successfully demonstrated that a well-designed multimedia database application can revolutionize a critical organizational workflow. By transforming the maintenance reporting process from an ambiguous, manual task into an efficient, transparent, and visually-driven one, the system provides tangible value to both staff and administrators.

REFERENCES

THIRUGNANASAMBANDAM, S. *et al.* (no date) *BUILDING REPAIRS AND MAINTENANCE*, *BUILDING REPAIRS AND MAINTENANCE*. https://www.annamalaiuniversity.ac.in/studport/download/engg/civil%20and%20structural/Building%20Repairs%20and%20Maintenance.pdf.