

**THE MAWEGO NATIONAL POLYTECHNIC**

**DEPARTMENT OF COMPUTING AND INFORMATICS**

**DIPLOMA IN INFORMATION AND COMMUNICATION TECHNOLOGY**

**PROJECT TITLE: AUTOMATED IT ASSET MAINTENANCE AND TRACKING SYSTEM**

**PRESENTED BY:** STEVE MARTIN OCHIENG OURE

**INDEX NUMBER:** 7311010066

**PAPER CODE:** 2920/308

**SUPERVISOR:** MR. REAGAN OMULLO

**PROJECT PROPOSAL SUBMITTED TO THE KENYA NATIONAL EXAMINATION COUNCIL IN PARTIAL FULFILMENT FOR THE AWARD OF DIPLOMA IN INFORMATION AND COMMUNICATION TECHNOLOGY**

**JULY 2025**

# DECLARATION

I, Steve Martin Ochieng Oure, declare that this research project is my work and it has never been represented in any University or College for an award.

Steve Martin Ochieng Oure

The Mawego National Polytechnic

Department of Computing and Informatics

Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Supervisor’s Approval

I, Mr. Reagan Omullo, confirm that this project titled "Automated IT Asset Maintenance and Tracking System” has been developed by Steve Martin Ochieng Oure under my supervision as the Supervisor.

Signature \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr/Mrs. Reagan Omullo

ICT Trainer

The Mawego National Polytechnic

# DEDICATION

I dedicate this research work to my parents Rose Okal and George Ngao for their unwavering support, encouragement, and prayers throughout this journey. Special dedication also goes to my instructors, classmates, and friends who stood by me during challenging times and inspired me to push forward. Their motivation gave me the strength to complete this project successfully.

# ACKNOWLEDGEMENT

I would like to express my sincere gratitude to the Almighty God for granting me the strength, good health, and resilience to undertake and complete this project. My heartfelt appreciation also goes to my supervisor, Mr. Reagan Omullo, for their guidance, constructive feedback, and patience throughout the development of this system. Special thanks to my family for their unwavering support and encouragement, especially during difficult moments. I also wish to thank my classmates and friends, for the valuable discussions, collaboration, and moral support that helped shape this project. Lastly, I am grateful to the Department of Computation and Informatics at The Mawego National Polytechnic for providing the resources and environment necessary for this project’s success.

Table of Contents

[DECLARATION ii](#_Toc202906773)

[DEDICATION iii](#_Toc202906774)

[ACKNOWLEDGEMENT iv](#_Toc202906775)

[LIST OF FIGURES viii](#_Toc202906776)

[ABSTRACT ix](#_Toc202906777)

[CHAPTER ONE: INTRODUCTION 10](#_Toc202906778)

[1.1 Introduction 10](#_Toc202906779)

[1.2 Background of the Problem 10](#_Toc202906780)

[1.3 Problem Statement 11](#_Toc202906781)

[1.4 Proposed Solution 12](#_Toc202906782)

[1.5 Justification 12](#_Toc202906783)

[1.6 General Objective 13](#_Toc202906784)

[1.7 Specific Objectives 13](#_Toc202906785)

[1.8 Research Questions 13](#_Toc202906786)

[CHAPTER TWO: LITERATURE REVIEW 15](#_Toc202906787)

[2.0 Introduction 15](#_Toc202906788)

[2.1 Overview of Literature Review 15](#_Toc202906789)

[2.2 User-Friendly Interface for IT Asset Tracking and Management 16](#_Toc202906790)

[2.3 Real-Time Tracking of IT Assets 17](#_Toc202906791)

[2.4 Maintenance Scheduling and History Tracking 17](#_Toc202906792)

[2.5 Securing IT Asset Information 18](#_Toc202906793)

[2.6 Generating Comprehensive Reports 19](#_Toc202906794)

[2.7 Enhancing Security and Access Control 20](#_Toc202906795)

[2.8 Conclusion 20](#_Toc202906796)

[CHAPTER THREE: RESEARCH METHODOLOGY 22](#_Toc202906797)

[3.0 Introduction 22](#_Toc202906798)

[3.1 Requirement Analysis 22](#_Toc202906799)

[3.1.1 Feasibility Study 22](#_Toc202906800)

[3.1.2 System Specifications 23](#_Toc202906801)

[3.2 System Design 25](#_Toc202906802)

[3.3 Implementation 25](#_Toc202906803)

[3.4 Testing 25](#_Toc202906804)

[3.5 Conclusion 26](#_Toc202906805)

[CHAPTER FOUR: SYSTEM ANALYSIS AND DESIGN 27](#_Toc202906806)

[4.0 Introduction 27](#_Toc202906807)

[4.1 Analysis of the Current System 27](#_Toc202906808)

[4.2 Requirements Analysis 28](#_Toc202906809)

[4.2.1 Types of Users and Their Requirements 28](#_Toc202906810)

[4.3 Functional Requirements 29](#_Toc202906811)

[4.3.1 User Authentication 29](#_Toc202906812)

[4.3.2 Asset Management 29](#_Toc202906813)

[4.3.3 Maintenance Scheduling 29](#_Toc202906814)

[4.3.4 Tracking 29](#_Toc202906815)

[4.3.5 Reporting 29](#_Toc202906816)

[4.3.6 User Management 29](#_Toc202906817)

[4.3.7 Notifications 29](#_Toc202906818)

[4.3.8 Context Diagram 30](#_Toc202906819)

[4.3.9 Level 0 Data Flow Diagram (DFD) 31](#_Toc202906820)

[4.3.3 Level 1 Data Flow Diagram (DFD) 32](#_Toc202906821)

[4.4 System Design 33](#_Toc202906822)

[4.4.1 Use Case Diagram 33](#_Toc202906823)

[4.4.2 System Flowcharts 34](#_Toc202906824)

[4.4.3 Activity Diagram 36](#_Toc202906825)

[4.4.4 Entity Relationship Diagram (ERD) 37](#_Toc202906826)

[4.5 Conclusion 38](#_Toc202906827)

[CHAPTER FIVE: SYSTEM IMPLEMENTATION 39](#_Toc202906828)

[5.1 Introduction 39](#_Toc202906829)

[5.2 User- Friendly Interface 39](#_Toc202906830)

[5.2.1 Login Module 39](#_Toc202906831)

[5.2.2 Dashboard Module 42](#_Toc202906832)

[5.2.3 Asset Management Module 46](#_Toc202906833)

[5.2.4 Maintenance Module 53](#_Toc202906834)

[5.2.5 Notification and Alerts Module 56](#_Toc202906835)

[5.2.6 User Module 58](#_Toc202906836)

[5.2.7 Settings Module 60](#_Toc202906837)

[5.3 Comprehensive Reports 62](#_Toc202906838)

[5.2.9 Database Structure 63](#_Toc202906839)

[5.3 Discussion 66](#_Toc202906840)

[5.4 Conclusion 67](#_Toc202906841)

[5.5 Recommendations 68](#_Toc202906842)

[REFERENCES 69](#_Toc202906843)

[APPENDICES 71](#_Toc202906844)

[APPENDIX 1: PROJECT BUDGET 71](#_Toc202906845)

[APPENDIX 2: PROJECT SCHEDULE (GANTT CHART) 72](#_Toc202906846)

# LIST OF FIGURES

[Figure 1Context diagram 30](#_Toc202906847)

[Figure 2 Data Flow diagram level 0 31](#_Toc202906848)

[Figure 3Level 1 DFD 32](#_Toc202906849)

[Figure 4. Use case diagram 33](#_Toc202906850)

[Figure 5 Asset Registration Flowchart 34](#_Toc202906851)

[Figure 6 Maintenance Scheduling Flowchart 35](#_Toc202906852)

[Figure 7 Activity Diagram 36](#_Toc202906853)

[Figure 8 E.R Diagram 37](#_Toc202906854)

[Figure 9 Login form 40](#_Toc202906855)

[Figure 10 Dashboard form 42](#_Toc202906856)

[Figure 11 Asset Management module 47](#_Toc202906857)

[Figure 12 Maintenance Module 53](#_Toc202906858)

[Figure 13 Alerts Module 57](#_Toc202906859)

[Figure 14 user Module 59](#_Toc202906860)

[Figure 15 Settings Module 61](#_Toc202906861)

[Figure 16 Asset Detail Report 62](#_Toc202906862)

[Figure 17 Maintenance report 62](#_Toc202906863)

[Figure 18 Database and Fields 63](#_Toc202906864)

[Figure 19 Database and Fields 64](#_Toc202906865)

[Figure 20 table Assets Structure 65](#_Toc202906866)

[Figure 21 Table Asset Sample Data 65](#_Toc202906867)

# ABSTRACT

This project presents the development of an Automated IT Asset Maintenance and Tracking System aimed at improving the management of IT assets within an organization. The system addresses challenges such as poor asset visibility, missed maintenance schedules, and lack of accountability by introducing features like a user-friendly interface, asset tracking, automated maintenance scheduling, and secure role-based access control. Developed using Visual Basic 6.0, the system enables activities, generate detailed reports, and ensure data integrity through user authentication. The solution integrates all modules within a unified dashboard for efficient navigation and access. This project demonstrates how automation and proper system design can streamline IT asset operations, reduce losses, and enhance decision-making in asset lifecycle management.

# CHAPTER ONE: INTRODUCTION

## 1.1 Introduction

In modern organizations, both IT assets such as computers, servers, printers, phones and network equipment play a crucial role in ensuring smooth operations. These assets are vital for maintaining day-to-day functions, and their management directly impacts efficiency, productivity, and cost-effectiveness. However, many organizations, struggle with inefficient tracking and maintenance of these assets. This project uses a case study of Sukari industries Ltd.

Sukari Industries, despite being an industrial leader, lacks a dedicated IT asset tracking system. The absence of such a system leads to challenges in maintaining, tracking, and optimizing the use of IT assets, which often results in unnecessary downtime, loss, and increased maintenance costs. Without proper monitoring of the life cycle and usage of these assets, resources are often underutilized or misplaced.

The Automated IT Asset Maintenance and Tracking System aims to provide a comprehensive solution to address these challenges. This system will automate the tracking of IT assets across their entire life cycle, including maintenance scheduling, status updates, and real-time location monitoring. By implementing such a system, organizations like Sukari Industries can reduce operational inefficiencies, prevent asset misplacement, and extend the useful life of its assets, ultimately leading to cost savings and better resource allocation.

## 1.2 Background of the Problem

Sukari Industries Ltd is a sugar milling company located in Homabay County, Ndhiwa Sub County. Sukari Industries Ltd relies heavily on IT devices to achieve its objectives.

The management of IT assets is a significant challenge for many organizations, especially in industrial environments like Sukari Industries, where operations depend on general IT equipment. Sukari Industries faces difficulties in efficiently tracking and maintaining its diverse IT assets, leading to frequent operational disruptions, equipment failures, and unnecessary maintenance costs. Without a proper asset tracking system, valuable time and resources are spent manually managing these assets, which affects overall productivity.

Currently, the organization relies on traditional methods, such as spreadsheets, stores records and manual records, to track its IT assets. This method is prone to human error, asset misplacement, and inefficiencies in maintenance scheduling. Without real-time updates on asset status or condition, Sukari Industries is unable to proactively address maintenance needs, which often results in downtime or unexpected equipment failures. Moreover, the lack of proper maintenance tracking increases the risk of assets being overused or neglected, shortening their lifespan.

A comprehensive Automated IT Asset Maintenance and Tracking System would address these issues by providing an organized, automated approach to managing assets. The system would streamline the tracking process, monitor asset performance, and automate maintenance schedules, ensuring assets are maintained efficiently and on time. This would minimize operational disruptions, optimize asset usage, and contribute to the overall success of the organization.

## 1.3 Problem Statement

In many organizations, the management and maintenance of IT assets is a significant challenge. The absence of an effective asset tracking system has resulted in inefficiencies in the management of these assets, leading to increased operational costs, asset misplacement, and unnecessary downtime. Currently, most organizations depend on manual methods such as spreadsheets and paper-based records to track and manage assets, which are prone to human error and inefficiencies.

The lack of a real-time, automated system for tracking IT assets has hindered

organizations from optimizing asset utilization and preventing equipment failures. Additionally, the absence of an integrated system for scheduling and tracking maintenance activities has led to missed or delayed maintenance, further affecting the performance and lifespan of critical assets. This problem not only increases operational costs but also reduces overall productivity, making it essential to implement an automated solution for IT asset tracking and maintenance.

## 1.4 Proposed Solution

To address the challenges identified, this project proposes the development of an Automated IT Asset Maintenance and Tracking System. The system will stream line the management of IT assets by providing a centralized platform where asset details including status, location, maintenance history, and assignment can be easily recorded and monitored. Through features such as automated maintenance scheduling, role-based access control, and real time reporting, the system will reduce manual workload, improve accountability, and minimize asset downtime. Built using visual Basic 6 and MS Access, the solution will be user friendly, secure, and suitable for deployment within small to medium sized institutions like hospitals or schools. This system aims to improve efficiency, ensure timely maintenance, and project asset data from unauthorized access or tampering.

## 1.5 Justification

The implementation of an Automated IT Asset Maintenance and Tracking System is crucial for organizations that rely on IT assets for their daily operations. By automating the asset tracking and maintenance processes, organizations can significantly reduce the inefficiencies associated with manual methods. The proposed system will provide real-time tracking of assets, ensuring that their locations, conditions, and usage are always updated and accessible.

The system will offer a more reliable and organized approach to asset management, ensuring that IT assets are maintained on schedule and remain in optimal working condition. This will prevent unexpected equipment failures, reduce downtime, and extend the lifespan of assets, ultimately leading to cost savings. In addition, automating these processes will free up valuable time and resources, allowing employees to focus on more strategic tasks.

Furthermore, with an automated system, organizations can easily generate reports on asset status, maintenance history, and usage patterns, which will assist in decision-making for asset procurement, replacement, and optimization. The proposed system will enhance operational efficiency, improve productivity, and provide greater control over IT assets, making it a valuable investment for any organization.

## 1.6 General Objective

To develop an automated system that tracks and monitors the usage, location, and condition of IT assets in real time preventing unexpected failure, reducing reliance on manual tracking and minimizing downtime.

## 1.7 Specific Objectives

1. To develop a user-friendly interface that allows administrators to easily track and manage the status, location, and condition of IT assets within the organization.
2. To integrate real-time tracking capabilities for monitoring the location and usage of IT assets throughout the organization, providing up-to-date information on asset status.
3. To implement a maintenance scheduling feature that automates reminders for routine maintenance and tracks the history of maintenance activities for each asset.
4. To generate comprehensive reports on asset status, usage, and maintenance history, enabling decision-makers to optimize asset utilization and plan for future upgrades or replacements.
5. To enhance security and access control by allowing only authorized personnel to access sensitive asset data and maintenance schedules, protecting the integrity of the system.

## 1.8 Research Questions

1. How can a user-friendly interface be designed to enable administrators to efficiently track and manage the status, location, and condition of IT assets within an organization?
2. What methods or technologies can be used to incorporate real-time tracking for monitoring the location and usage of IT assets within the organization?
3. How can an automated scheduling feature be implemented to manage and record routine IT asset maintenance activities effectively?
4. What kind of reporting activities should the system include to provide meaningful insights on asset usage, maintenance activities effectively?
5. Which security and access control mechanisms are most effective in limiting access to sensitive IT asset data and ensuring the integrity of the tracking system?

# CHAPTER TWO: LITERATURE REVIEW

## 2.0 Introduction

This chapter presents a review of relevant literature related to the development of an Automated IT Asset Maintenance and Tracking systems. The review is organized based on the specific objectives. Each section explores existing technologies, methodologies, and practices relevant to the objective in question. This helps in understanding the current trends, identifying gaps in existing systems, and providing a strong foundation for designing and implementing the proposed system.

## 2.1 Overview of Literature Review

The literature review explores key theoretical and practical contributions from past studies and scholarly work related to IT asset management systems. According to Gartner (2020) effective management of IT assets plays a critical role in ensuring operational efficiency, cost control, and long-term sustainability within organizations. Rouse (2021), emphasized that digital solutions, particularly asset tracking systems, improve visibility, automate maintenance cycles, and support informed decision-making.

In designing such systems, usability is paramount. A user-friendly interface not only enhances efficiency but also reduces errors and training time (Neilsen, 1994). Furthermore, modern systems often implement real-time tracking through technologies such as barcodes, RFID, or QR codes, enabling organizations to locate and monitor assets across various departments or locations (Zhang et al., 2019).

The literature also highlights the growing need for automated maintenance scheduling features. Systems that can notify users of upcoming service intervals or track repair histories are essential for prolonging asset lifecycles and reducing unexpected downtimes (Patel & Doshi, 2018). Generating detailed reports is another widely discussed component, as data-driven insights help organizations assess performance trends, predict asset failures, and budget for replacements (Sharma & Kumar, 2020).

Security remains a critical focus in IT asset systems. Role-based access controls, user authentication, and encryption are common best practices used to protect sensitive data from unauthorized access or misuse (Stallings & Brown, 2018). Many organizations implement multi-level user privileges to ensure that only authorized personnel can view or alter critical asset records.

Lastly, the importance of integrating asset management systems with existing platforms such as inventory, procurement, or financial software is considered as well. Integration ensures data consistency across departments, improves operational workflows, and reduces redundant data entry (Sambamurthy & Zmud, 2014). Although not always the primary objective in smaller deployments, integration is vital for scalability and long-term system effectiveness.

## 2.2 User-Friendly Interface for IT Asset Tracking and Management

A user-friendly interface is essential in ensuring that systems are easily navigable, reduce errors, and increase user satisfaction. For an IT asset management system, usability directly influences the efficiency of administrators in tracking, managing, and maintaining assets across an organization.

According to Nielsen (1993), usability in system design involves five key components: learnability, efficiency, memorability, error handling, and user satisfaction. These principles guide the development of interfaces that are intuitive and productive. When applied to IT asset management, they help ensure that users can quickly access critical information about asset status, location, and condition without requiring extensive training.

Shneiderman et al. (2016) emphasize the importance of visual clarity and logical layout in interface design. Incorporating structured menus, clearly labeled buttons, and intuitive workflows minimizes user confusion and facilitates faster operations, especially in administrative contexts where time and precision are critical.

Modern systems often adopt graphical user interfaces (GUIs) that support interaction through visual elements rather than command lines. As noted by Galitz (2007), well-designed GUIs not only enhance usability but also reduce cognitive load, which is especially beneficial when managing complex systems involving numerous IT assets.

Furthermore, responsive feedback such as confirmation messages, status updates, and alerts improves the communication between the system and its users. Such features are key in ensuring that administrators are constantly aware of system status and asset conditions, thus supporting informed decision-making.

By prioritizing usability, the system enhances operational efficiency, reduces the learning curve for new users, and ensures better compliance with asset management protocols.

## 2.3 Real-Time Tracking of IT Assets

Real-time tracking in IT asset management enables organizations to monitor the location, movement, and status of assets as events unfold. This functionality improves transparency, enhances operational efficiency, and minimizes asset loss or misuse.

The integration of real-time tracking tools, such as barcode scanning, RFID (Radio Frequency Identification), or network-based location tracking, helps ensure that IT administrators can access the most current data on any asset. As stated by Turban et al. (2018), real-time data availability supports timely decision-making, reduces downtime, and improves accountability in asset handling.

Real-time tracking ensures that any changes in an asset’s condition or location are immediately reflected in the system. This dynamic update mechanism is essential in large institutions where IT assets are frequently relocated or reassigned. For example, Gupta and Kohli (2006) note that timely asset visibility directly contributes to effective resource planning and audit readiness.

Additionally, real-time monitoring aids in identifying idle or underutilized assets. This insight allows organizations to reallocate resources where needed, thereby optimizing usage and reducing redundant purchases. Integrating this capability with automated alerts also facilitates faster response to issues such as unauthorized movement or overdue maintenance.

Although the implementation of advanced tracking technologies may require additional infrastructure or policy alignment, the long-term benefits in data accuracy and asset accountability justify the investment.

## 2.4 Maintenance Scheduling and History Tracking

Proper maintenance of IT assets is essential for ensuring long-term performance, reducing unexpected downtime, and extending equipment lifespan. A system that automates maintenance scheduling and keeps historical records streamlines this process and minimizes manual oversight.

Automated maintenance scheduling involves setting predefined intervals for servicing assets and generating timely reminders to alert responsible personnel. According to Kumar & Saini (2015), such automation minimizes human error, enhances reliability, and ensures that no asset is overlooked during critical maintenance windows.

Recording maintenance history provides a comprehensive audit trail for each asset. This includes details such as previous issues, servicing dates, parts replaced, and technicians involved. Having this information readily available supports better diagnostic accuracy and informed decision-making for future servicing needs or asset replacement.

Moreover, the maintenance log supports compliance with organizational policies or regulatory standards that require documentation of asset upkeep. Laudon and Laudon (2020) note that accurate records reduce risks associated with unplanned failures and improve asset accountability during audits.

This scheduling feature also contributes to financial planning by helping predict future maintenance costs and identifying patterns that may indicate systemic issues in certain asset categories.

While the implementation of a maintenance scheduling module requires input from both IT and asset management teams, its value in reducing downtime and enhancing operational efficiency is significant.

## 2.5 Securing IT Asset Information

Security is one of the most important aspects of any system that deals with sensitive data. In an organization, IT assets often contain or support access to critical information. Therefore, it is important to ensure that only authorized people can access or modify asset records.

To keep the system safe, different users can be assigned different roles. For example, an administrator may have access to all parts of the system, while a technician may only see the assets they maintain. This is known as role-based access control (RBAC). It ensures that users only see what they are allowed to do.

In addition, the system should have strong login security like using usernames and passwords. Each action a user takes like editing asset data or updating maintenance records should be recorded. This helps in tracking changes and holding users accountable.

According to Stallings and Brown (2018), systems that use access control and activity tracking improve both security and accountability. These controls reduce the chances of internal misuse and unauthorized access.

In a Visual Basic 6 (VB6) system, this can be done using login forms, user account tables in the database, and code that checks each user’s role before allowing access to specific forms or features.

Overall, protecting the data of IT assets helps maintaining trust, improves system reliability, and keeps the organization’s information safe from unauthorized access or tampering.

## 2.6 Generating Comprehensive Reports

One of the core functionalities of an effective IT asset management system is the ability to generate comprehensive and actionable reports. These reports serve as a vital tool for decision-makers, enabling them to analyze asset utilization, evaluate performance, and make strategic plans for maintenance, upgrades, or replacements.

Comprehensive reports consolidate data regarding asset status, location, usage frequency, and maintenance history. This consolidation facilitates performance tracking and helps identify underutilized or overburdened assets. For instance, if certain devices experience frequent breakdowns, reports can highlight these trends, prompting preventive action or timely replacement.

Moreover, usage analytics derived from reporting tools can assist in optimizing resource allocation. According to Laudon & Laudon (2020), reporting mechanisms within information systems empower managers with insights that lead to more efficient operations and better financial planning.

Maintenance history reports also play a critical role in compliance and audit processes. By maintaining a detailed log of service schedules, repairs, and costs, organizations can ensure accountability and justify expenditures. These records help minimize downtime and ensure that critical IT assets are always operational.

Additionally, customized reports can be generated for various stakeholders technical teams, finance departments, and top management each tailored to highlight the data most relevant to their roles. Advanced systems may allow filtering, exporting, and visualizing this information in charts or dashboards.

Ultimately, the ability to generate reports fosters a data-driven culture where asset management decisions are based on real-time evidence rather than assumptions.

## 2.7 Enhancing Security and Access Control

Security is a critical component in any information system, particularly one dealing with sensitive IT asset data. The Automated IT Asset Maintenance and Tracking System must ensure that access to data is restricted to authorized personnel to uphold confidentiality, integrity, and availability of asset information.

Access control mechanisms help prevent unauthorized users from modifying or viewing sensitive records. This can be achieved through techniques such as user authentication, role-based access control (RBAC), and activity logging. With RBAC, different users are assigned specific permissions based on their roles—administrators, technicians, and auditors may each have varying levels of access to the system’s functionalities and data.

According to Stallings & Brown (2018), well-structured access control models not only restrict unauthorized use but also help in accountability by logging user actions within the system. For example, if an asset record is modified, the system should log who made the change and when, enhancing traceability and reducing internal misuse.

The system should also incorporate measures such as strong password enforcement policies, encryption of stored and transmitted data, automatic session timeouts to protect against unattended terminals and user management interfaces for account creation, suspension, or deactivation.

Implementing these security protocols is essential in an organizational setting, where breaches in IT asset data can lead to operational disruptions, financial losses, or even legal implications.

In a VB6 environment, while modern encryption and security libraries may be limited, essential access control can still be implemented using structured login forms, user-level tables in the database, and conditional form loading based on user roles.

Ultimately, enhancing security and access control ensures that asset data remains protected, reliable, and tamper-proof, aligning with industry best practices in IT asset management.

## 2.8 Conclusion

This chapter has explored the key objectives that guide the development of the Automated IT Asset Maintenance and Tracking System. Each section aligned with a specific objective from creating a user-friendly interface, enabling real-time tracking, automating maintenance scheduling, generating insightful reports, to securing access to sensitive asset data. These elements are essential for improving how IT assets are managed, maintained, protected, and in a modern organizational environment. By reviewing literature practices related to each area, this chapter has laid the foundation for analyzing the current system and identifying gaps that the proposed solution aims to fill.

# CHAPTER THREE: RESEARCH METHODOLOGY

## 3.0 Introduction

This chapter presents the methodology that will be employed in the development of the Automated IT Asset Maintenance and Tracking System. The selected model, the Waterfall Model, follows a linear sequence of phases: Requirement analysis, System design, Implementation, Testing, Deployment, and Maintenance. Each phase builds upon the previous, ensuring structured progress from problem identification to solution deployment.

## 3.1 Requirement Analysis

In this project, requirement gathering was conducted through personal observation and referencing best practices found in literature. Specific requirements were identified, such as user login functionality, asset registration and maintenance scheduling. Both functional and non-functional requirements were documented clearly and served as a blueprint for system’s structure and capabilities.

### 3.1.1 Feasibility Study

A feasibility study is conducted to determine whether the proposed Automated IT Asset Maintenance and Tracking System can be practically implemented using the available resources, technology, and within the constraints of time and budget. This study examined four major areas of feasibility: technical, economical, operational, and legal feasibility.

#### 3.1.1.1 Technical Feasibility

The proposed system can be developed using Visual Basic 6 (VB6), a reliable and tested programming environment that supports the creation of user-friendly desktop applications. Since most organizations already use Windows-based system, VB6 is compatible and appropriate. Basic networking and a relational database such as MS Access will also be used, all of which are available and, manageable with the developer’s current skill set.

#### 3.1.1.2 Economic Feasibility

This system is cost-effective because it can be developed using free or already available software tools (VB6, MS Access). No need for new hardware or expensive licenses makes the project economically viable. Additionally, by automating asset tracking and maintenance reminders, the system will reduce losses from misplaced equipment and missed maintenance schedules, saving money in the long term.

#### 3.1.1.3 Operational Feasibility

The proposed system will solve key problems such as manual tracking of IT assets, lack of accountability, and inefficient maintenance scheduling. It is easy to use and will require minimal training for staff. Since the system aligns with how asset data is already managed manually, transitioning will be smooth and the benefits will be quickly felt in day-to-day operations.

#### 3.1.1.4 Legal Feasibility

The system will store sensitive IT asset information, and appropriate access control will be implemented to ensure compliance with standard data protection practices. Only authorized users will have access to critical data, ensuring that confidentiality and data integrity are maintained.

In conclusion, the feasibility study indicates that the project is practical, affordable, and beneficial. All the factors examined support the successful development and deployment of the system.

### 3.1.2 System Specifications

This section outlines the technical and functional requirements for developing the Automated IT Asset Maintenance and Tracking System. These specifications serve as the foundation for system design, ensuring that the final product meets user expectations and operates effectively within the intended environment.

#### 3.1.2.1 Hardware Requirements

The system will run on standard desktop computers used within an organization. Below are the minimum and recommended hardware requirements:

Minimum Requirements:

* Processor: Intel Pentium 4 or equivalent.
* RAM: 1GB.
* Hard Disk: 100 MB free space.
* Display: SVGA Monitor (1024x768 resolution).
* Input Devices: Keyboard and Mouse.

Recommended Requirements:

* Processor: Intel Core i3 or higher.
* RAM: 4GB or more.
* Hard Disk: 500MB free space.
* Network Interface Card (for LAN connectivity).
* UPS for power backup.

#### 3.1.2.2 Software Requirements

The system will be developed using Visual Basic 6, with a Backend database of Microsoft Access. The following software will be used:

* Operating System: Windows 10.
* Development environment: Microsoft Visual Basic 6.0.
* Database Management: Microsoft Access.
* Antivirus Software ensure security of system files.

#### 3.1.2.3 Functional Requirements

The system should be able to perform the following core functions:

* User login and authentication.
* Asset registration for adding new IT assets with details.
* Tracking of asset location and condition.
* Generation of reports; status, maintenance and usage.
* Role-based access control; admin technician and viewer.
* Alert and notification system for due maintenance.

#### 3.1.2.4 Non-Functional Requirements

* Usability: The system will have a user-friendly interface suitable for non-technical users.
* Reliability: The system should operate consistently without crashing or data loss.
* Security: Only authorized users can log in, and all user actions are logged.
* Maintainability: The system should be easy to and fix in case of errors.
* Performance: The system should respond quickly to user inputs and generate reports efficiently.

These specifications ensure that the system will be robust, efficient, and tailored to the organization’s IT asset management needs.

## 3.2 System Design

This stage focused on designing the overall system architecture, identifying the system’s modules, user interfaces, database structures, and how various components interact with one another. The design process ensured that the system would be modular, scalable, and maintainable.

## 3.3 Implementation

The implementation phase involved the actual development and construction of the Automated IT Asset Maintenance and Tracking System using Visual Basic 6.0 as the programming language and Microsoft Access as the backend database. This phase translated the system design into a working software application.

During Implementation forms were created for user login and dashboard. The interface was organized using frames within the main dashboard form, which allowed users to navigate between modules launching new windows. This helped maintain a clean, consistent user experience.

## 3.4 Testing

The testing phase was carried out to ensure that the Automated IT Asset Maintenance and Tracking System functioned as intended and met the specific objectives outlined in the initial stages of the project. Testing focused on identifying errors, verifying functionalities, and evaluating the overall performance of the system.

User Acceptance Testing (UAT) was simulated by assuming the roles of the different users-admin, technician, and asset user.

## 3.5 Conclusion

In summary, this chapter has presented the methodology used in developing the Automated IT Asset Maintenance and Tracking System, guided by the waterfall Model. The study covered the stages of the waterfall model. Although deployment and maintenance were not fully reached due to academic constraints, the completed stages provided a solid framework for building a functional system. The feasibility assessment and system specifications confirmed that the project was both practical and achievable within the available resources.

# CHAPTER FOUR: SYSTEM ANALYSIS AND DESIGN

## 4.0 Introduction

This chapter presents the system analysis and design of the automated IT Asset Maintenance and Tracking System. The analysis involves understanding the current system, identifying its weakness, and outlining the requirements for proposed system. The design aspect includes functional modeling and the use of diagrams, data flow diagrams, use case diagrams. These tools help visualize how the system will operate and interact with users and data.

## 4.1 Analysis of the Current System

The current system for managing IT assets in most organizations including Sukari Industries Ltd, is largely manual and paper-based. Asset information such as the item description, serial numbers, condition, and location is often recorded in logbooks or spreadsheets, which are updated periodically by hand. In some cases, the records may be incomplete, inconsistent, or misplaced, leading to difficulties in asset tracking and maintenance.

When an IT asset requires maintenance, the process is typically initiated by physically reporting the issue to the IT department or writing it in a maintenance request book. There is often no systematic follow-up, resulting in delays, overlooked tasks, or redundant maintenance efforts. Additionally, there is no real-time visibility into asset locations, no alerts for scheduled maintenance, and no automated reports to assist in decision-making.

Security is also a concern, as asset data can be accessed by unauthorized individuals or altered without audit trails. The absence of role-based access control increases the risk of data manipulation or loss.

Overall, the current system is inefficient prone to human error, lacks transparency, and is no scalable as the organization grows. These shortcomings have necessitated the development of an automated system to streamline IT asset maintenance and tracking.

## 4.2 Requirements Analysis

To design an effective Automated IT Asset Maintenance and Tracking System, it’s essential to identify the requirements of the system based on the various types of users who will interact with it. These requirements define what the system must do and how it should behave, starting with a breakdown of user types and their needs.

### 4.2.1 Types of Users and Their Requirements

#### 4.2.1.1 Administrator

Has the highest level of access.

Can add, update, and delete asset records.

Manages user accounts and roles.

Views all reports, logs, and system statistics.

Sets maintenance schedule and policies.

#### 4.2.1.2 IT Technician

Views assigned maintenance tasks.

Updates asset condition after inspection or repair.

Logs maintenance history and remarks.

May request new assets or flag assets for requirement.

#### 4.2.1.3 Auditor

Has read-only access to system reports and logs.

Monitors asset usage, movements, and depreciation.

Evaluates maintenance efficiency and asset life cycle trends.

#### 4.2.1.4 Standard User

Can view assets assigned to them.

Can report asset or request maintenance.

Views maintena5nce status of assets they are responsible for.

## 4.3 Functional Requirements

Functional requirements describe the specific behavior and functions the system must support. These outline what the Automated IT Asset Maintenance and Tracking System should do from a user’s perspective.

Here are the core functional requirements:

### 4.3.1 User Authentication

The system must allow users to log in using a valid username and password.

The system should validate user roles and grant access accordingly.

### 4.3.2 Asset Management

Admins can add, update, or delete IT asset records.

Users can view assigned assets and their details including status, location and last maintenance.

### 4.3.3 Maintenance Scheduling

Admins can set routine maintenance schedules.

Technicians can view due tasks and update maintenance reports.

### 4.3.4 Tracking

The system updates the current location of each asset.

Movement of assets between departments is logged.

### 4.3.5 Reporting

Generate reports on asset status, usage history, and maintenance logs.

Export to PDF.

### 4.3.6 User Management

Admins can create, update, suspend, or delete user accounts

Role-based access is enforced: Admin, Technician, Auditor, Department User.

### 4.3.7 Notifications

System should alert technicians of upcoming maintenance.

Users are notified when their reported issues are resolved.

### 4.3.8 Context Diagram

The context diagram below shows the entire system as a single process and its interaction with external entities.

Figure 1Context diagram

Department User

Technician

Auditor/

Management

Administrator

### 4.3.9 Level 0 Data Flow Diagram (DFD)

This diagram shows the entire system as a single process, the entities that access the data that flows to and from the database.

Figure 2 Data Flow diagram level 0

D4 Notifications DB

D3 Maintenance Records

D2 Assets DB

D1 Users DB

Asset User

Administrator

Technician

### 4.3.3 Level 1 Data Flow Diagram (DFD)

This diagram breaks the system into major processes, showing how data flows between them and the external entities.

Technician

Login credentials

Login credential

D3. Maintenance Records

D4. Reports Archive

D2. Asset Database

D1. User Accounts

Figure 3Level 1 DFD

Department User

Auditor/

Manager

Administrator

# 4.4 System Design

### 4.4.1 Use Case Diagram

A Use Case Diagram is used to show how different users interact with the system. It visually outlines the system’s main functionalities from the user’s point of view.

Department User

Technician

Figure 4. Use case diagram

Administrator

### 4.4.2 System Flowcharts

These system Flowcharts shows how data and processes flow through the system from input to output. They outline the major steps involved in managing IT assets in the proposed system.

No

Is User Admin

Enter Asset Details.

Save to Database

Login

Yes

Figure 5 Asset Registration Flowchart

System sends reminder Notification Auto

Confirmation Message

Enter Maintenance schedule info

Select Asset

View Asset list

Login Admin or Technician

Figure 6 Maintenance Scheduling Flowchart

### 4.4.3 Activity Diagram

An Activity Diagram visualizes the sequence of activities involved in using the system. It shows the flow of control from one activity to another. It is useful for understanding how users interact with the system to perform tasks like asset registration, maintenance, or report generation.

Else

System saves data to database

System logs user Activity

User Submits Maintenance details

User views Maintenance details

User Selects an Asset

User Updates and Add New details

Role check

Display asset list

If authorized

System authenticates User

User Logs in

Figure 7 Activity Diagram

Technician

ADMIN

### 4.4.4 Entity Relationship Diagram (ERD)

An Entity Relationship Diagram (ERD) is used to model the logical structure of the database. It shows entities, their attributes, and relationship between them.

M

1

M

1

1

M

Perfom

assigned

M

1

Login\_Log

User

Maintance\_Record

IT\_Asset

Figure 8 E.R Diagram

## 4.5 Conclusion

This chapter has outlined the system analysis and design of the proposed Automated IT Asset Maintenance and Tracking System. It began by evaluating the shortcomings of the current manual system, highlighting inefficiencies such as lack of real-time tracking, delayed maintenance, and weak security controls. A comprehensive requirement specification was developed based on different user roles, guiding the functional needs of the system. Various modeling tools including context diagrams, data flow diagrams, flowcharts, activity diagrams, use case diagrams, and the entity-relationship diagram were used to visualize system functionality and database structure. These designs form the foundation for the actual system implementation, ensuring that it is efficient, user-friendly, and secure.

# CHAPTER FIVE: SYSTEM IMPLEMENTATION

## 5.1 Introduction

The implementation of the Automated IT Asset Maintenance and Tracking System marks the culmination of the system development process. The system is developed using Visual Basic 6.0 (VB6) as the front-end interface and Microsoft Access as the backend database. It is designed to manage the life cycle of IT assets within an organization, including their registration, maintenance, tracking and reporting.

The goal of this chapter is to present the actual working system, showing how the various modules interact to meet the defined objectives. Screenshots of the user interface are provided to illustrate the appearance and functionality of the system. Additionally, code snippets for key functions are included to offer insights into the logic behind important operations, such as login validation, asset management, and maintenance scheduling.

The database structure is also highlighted to show how information is organized and stored. The implementation has been guided by the system requirements identified earlier and aligns with the proposed design architecture.

## 5.2 User- Friendly Interface

### 5.2.1 Login Module

The Login Module is the entry point to the system that handles user authentication. It checks the entered username and password against stored credentials in the Access database. If login is successful, the user is redirected to the dashboard and their role is captured.

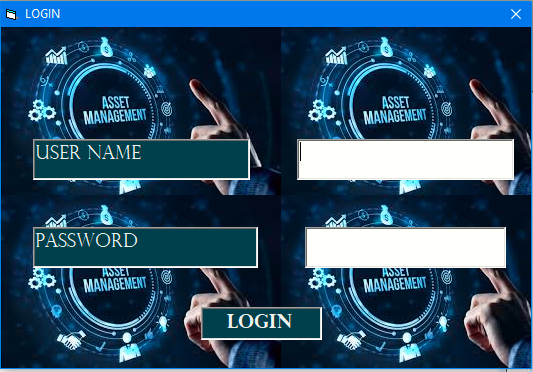


Figure 9 Login form

Private Sub lblLogin\_Click()

Dim Name As String

Dim Pass As String

Dim max\_num As Integer

Dim CurrentTime As Date

Dim username As String

Dim computername As String

username = Environ("USERNAME")

computername = Environ$("COMPUTERNAME")

Name = txtName.Text

Pass = txtPass.Text

DatUser.Refresh

DatLog.Refresh

max\_num = 0

If Name = "" Or Pass = "" Then

MsgBox "Enter All Login Credentials to Login!", vbExclamation

Exit Sub

End If

If Not DatLog.Recordset.BOF Then

DatLog.Recordset.MoveFirst

End If

If Not DatUser.Recordset.BOF Then

DatUser.Recordset.MoveFirst

End If

Do Until DatLog.Recordset.EOF

If DatLog.Recordset.Fields("LogID") > max\_num Then

max\_num = DatLog.Recordset.Fields("LogID")

End If

DatLog.Recordset.MoveNext

Loop

Do Until DatUser.Recordset.EOF

If DatUser.Recordset.Fields("Name") = Name And DatUser.Recordset.Fields("Password") = Pass Then

LogID = max\_num + 1

Role = DatUser.Recordset.Fields("Role")

CurrentUserID = DatUser.Recordset.Fields("UserID")

frmDash.Caption = Role & ": " & Name

DatLog.Recordset.AddNew

DatLog.Recordset.Fields("LogID") = LogID

DatLog.Recordset.Fields("LoginTime") = Now

DatLog.Recordset.Fields("ComputerName") = computername

DatLog.Recordset.Fields("UserID") = CurrentUserID

DatLog.Recordset.Update

Load frmDash

frmDash.Show

DatUser.Recordset.Edit

DatUser.Recordset.Fields("Status") = "Online"

DatUser.Recordset.Update

MsgBox "WELCOME!", vbInformation

Unload Me

Exit Sub

End If

DatUser.Recordset.MoveNext

Loop

MsgBox "Log in Failed: Wrong Password Or User Name!", vbExclamation

End Sub

### 5.2.2 Dashboard Module

The Dashboard Module acts as a control center of the Automated IT Asset Maintenance and Tracking System. Once a user logs in, they are redirected to this interface. It provides a user –specific view depending on the access level or role, whether Admin, Technician or Asset user. It has got the content panel where other modules contents are displayed.



Figure 10 Dashboard form

Private Sub Form\_Load()

DatAssets.Refresh

DatUsers.Refresh

frameUser.Visible = False

DatMaintenance.Refresh

DatNotification.Refresh

frameIssueAssets.Visible = False

frameNotifications.Visible = False

cmbMStatus.AddItem "Maintained"

cmbMStatus.AddItem "Pending"

cmbMStatus.AddItem "Missed"

cmbMStatus.AddItem "Completed"

cmbMStatus.AddItem "Scheduled"

DTPicker1.Value = Now

If Role = "AssetUser" Then

lblInterval.Visible = False

txtInterval.Visible = False

lblMAssign.Visible = False

DatAssets.RecordSource = "SELECT \* FROM Assets WHERE UserID=" & CurrentUserID

DatUsers.RecordSource = "SELECT \* FROM User WHERE UserID=" & CurrentUserID

lblDate.Caption = "Maintenance Date"

cmbStatus1.Enabled = False

DTPicker1.Enabled = False

cmbDepartment.Enabled = False

cmbDepartment.Text = ""

cmbStatus1.Text = ""

txtUserID.Text = CurrentUserID

txtUserID.Enabled = False

txtAssetID.Text = "SELECT"

route = True

lblIssue.Caption = "SUBMIT"

If Not DatAssets.Recordset.BOF Then

DatAssets.Recordset.MoveFirst

End If

If Not DatMaintenance.Recordset.BOF Then

DatMaintenance.Recordset.MoveFirst

End If

Do Until DatAssets.Recordset.EOF

If DatAssets.Recordset.Fields("UserID") = CurrentUserID Then

txtAssetID.AddItem DatAssets.Recordset.Fields("AssetID")

End If

DatAssets.Recordset.MoveNext

Loop

Do Until DatMaintenance.Recordset.EOF

DatAssets.Recordset.MoveFirst

Do Until DatAssets.Recordset.EOF

If DatMaintenance.Recordset.Fields("AssetID") = DatAssets.Recordset.Fields("AssetID") And DatAssets.Recordset.Fields("UserID") = CurrentUserID Then

cmbMMaintenanceID.AddItem DatMaintenance.Recordset.Fields("MaintenanceID").Value

End If

DatAssets.Recordset.MoveNext

Loop

DatMaintenance.Recordset.MoveNext

Loop

End If

'Admin

If Role = "Admin" Then

lblDescription.Visible = False

txtDescription.Visible = False

cmbStatus1.AddItem "Assigned"

cmbStatus1.AddItem "Available"

cmbStatus1.AddItem "Maintenance"

cmbStatus1.AddItem "Retired"

cmbDepartment.AddItem "IT Department"

cmbDepartment.AddItem "Human Resource"

cmbDepartment.AddItem "Automotive"

cmbDepartment.AddItem "Finance"

cmbUDepartment.AddItem "IT Department"

cmbUDepartment.AddItem "Human Resource"

cmbUDepartment.AddItem "Automotive"

cmbUDepartment.AddItem "Finance"

cmbURole.AddItem "Admin"

cmbURole.AddItem "AssetUser"

cmbURole.AddItem "Technician"

If Not DatAssets.Recordset.BOF Then

DatAssets.Recordset.MoveFirst

End If

Do Until DatAssets.Recordset.EOF

txtAssetID.AddItem DatAssets.Recordset.Fields("AssetID")

cmbMAssetID.AddItem DatAssets.Recordset.Fields("AssetID")

DatAssets.Recordset.MoveNext

Loop

If Not DatUsers.Recordset.BOF Then

DatUsers.Recordset.MoveFirst

End If

Do Until DatUsers.Recordset.EOF

If DatUsers.Recordset.Fields("Role") = "Technician" Then

cmbMTechnicianID.AddItem DatUsers.Recordset.Fields("UserID")

cmbATechnicianID.AddItem DatUsers.Recordset.Fields("UserID")

End If

DatUsers.Recordset.MoveNext

Loop

If Not DatMaintenance.Recordset.BOF Then

DatMaintenance.Recordset.MoveFirst

End If

Do Until DatMaintenance.Recordset.EOF

If DatMaintenance.Recordset.Fields("TechnicianID") = 0 Then

cmbMMaintenanceID.AddItem DatMaintenance.Recordset.Fields("MaintenanceID").Value

End If

DatMaintenance.Recordset.MoveNext

Loop

If Not DatNotification.Recordset.BOF Then

DatNotification.Recordset.MoveFirst

End If

Do Until DatNotification.Recordset.EOF

If DatNotification.Recordset.Fields("Status") = "Unread" Then

cmbAAssetID.AddItem DatNotification.Recordset.Fields("AssetID")

lblAlerts.BackColor = &H40C0&

End If

DatNotification.Recordset.MoveNext

Loop

End If

'Technician

If Role = "Technician" Then

lblReply.Visible = False

DatNotification.Refresh

lblDescription.Caption = "REQUEST ITEM"

lblIssue.Caption = "Request"

cmbStatus1.AddItem "Maintenance"

lblMAssign.Caption = "SUBMIT"

DatMaintenance.Refresh

If Not DatAssets.Recordset.BOF Then

DatAssets.Recordset.MoveFirst

End If

If Not DatMaintenance.Recordset.BOF Then

DatMaintenance.Recordset.MoveFirst

End If

Do Until DatMaintenance.Recordset.EOF

If Not DatAssets.Recordset.BOF Then

DatAssets.Recordset.MoveFirst

End If

Do Until DatAssets.Recordset.EOF

If DatMaintenance.Recordset.Fields("TechnicianID") = CurrentUserID Then

If DatMaintenance.Recordset.Fields("AssetID") = DatAssets.Recordset.Fields("AssetID") Then

txtAssetID.AddItem DatAssets.Recordset.Fields("AssetID")

cmbMMaintenanceID.AddItem DatMaintenance.Recordset.Fields("MaintenanceID").Value

End If

End If

DatAssets.Recordset.MoveNext

Loop

DatMaintenance.Recordset.MoveNext

Loop

If Not DatNotification.Recordset.BOF Then

DatNotification.Recordset.MoveFirst

End If

Do Until DatNotification.Recordset.EOF

If DatNotification.Recordset.Fields("UserID") = CurrentUserID Then

cmbAAssetID.AddItem DatNotification.Recordset.Fields("AssetID")

End If

DatNotification.Recordset.MoveNext

Loop

End If

End Sub

### 5.2.3 Asset Management Module

The Asset Management Module enables administrators to register, view, and assign IT assets to specific users or departments. It enables technicians to view details of assets they are maintaining. This module enables regular asset users view make inquiry about assets assigned to them. It is central to the system’s functionality and ensures that asset records are accurately maintained and up to date. The Module is hosted within a frame on the dashboard’s content panel.



Figure 11 Asset Management module

Private Sub lblAssets\_Click(Index As Integer)

frameMaintenance.Visible = False

frameNotifications.Visible = False

frameUser.Visible = False

If Role = "Admin" Then

PopupMenu mnuAssets

Exit Sub

End If

If frameIssueAssets.Visible = True Then

frameIssueAssets.Visible = False

Else

frameIssueAssets.Visible = True

End If

End Sub

Private Sub cmbAAssetID\_Click()

DatUsers.Refresh

DatNotification.Refresh

If Not DatAssets.Recordset.BOF Then

DatAssets.Recordset.MoveFirst

End If

If Not DatNotification.Recordset.BOF Then

DatNotification.Recordset.MoveFirst

End If

Do Until DatNotification.Recordset.EOF

DatAssets.Recordset.MoveFirst

Do Until DatAssets.Recordset.EOF

If DatAssets.Recordset.Fields("AssetID") = DatNotification.Recordset.Fields("AssetID") Then

DatUsers.RecordSource = "SELECT \* FROM User WHERE UserID=" & DatAssets.Recordset.Fields("UserID").Value

txtAAssetName.Text = DatAssets.Recordset.Fields("AssetName")

txtAUserID.Text = DatAssets.Recordset.Fields("UserID")

txtAUserName.Text = DatUsers.Recordset.Fields("Name")

txtATitle.Text = DatNotification.Recordset.Fields("Title")

DTPicker3.Value = DatNotification.Recordset.Fields("NotificationID")

txtAMessage.Text = DatNotification.Recordset.Fields("Message")

DatNotification.Recordset.Edit

DatNotification.Recordset.Fields("Status") = "Read"

DatNotification.Recordset.Update

lblAlerts.BackColor = &H404000

End If

DatAssets.Recordset.MoveNext

Loop

DatNotification.Recordset.MoveNext

Loop

End Sub

Private Sub cmdPrint\_Click()

Dim AssetID As Long

AssetID = Val(InputBox("Enter The Asset Id "))

Adodc1.RecordSource = "SELECT \* FROM Assets WHERE AssetID=" & AssetID

Adodc1.Refresh

DataReport1.Caption = "ASSET DETAILS"

Set DataReport1.DataSource = Adodc1

DataReport1.Show

End Sub

Private Sub lblAlerts\_Click()

frameIssueAssets.Visible = False

frameMaintenance.Visible = False

frameUser.Visible = False

If frameNotifications.Visible = False Then

frameNotifications.Visible = True

Else

frameNotifications.Visible = False

End If

End Sub

Private Sub lblIssue\_Click()

Dim AssetName As String

Dim Manufacturer As String

Dim status As String

Dim serialNo As String

Dim Department As String

Dim purchaseDate As Date

Dim UserID As Long

Dim max\_id As Long

Dim DeptID As Long

Dim num As Integer

Dim maxid As Long

max\_id = 0

If Role = "Admin" Then

DatMaintenance.Refresh

DatUsers.Refresh

DatAssets.Refresh

DatDepartment.Refresh

num = Val(txtInterval.Text)

AssetName = frmDash.txtAssetName.Text

Manufacturer = txtManufacturer.Text

status = cmbStatus1.Text

maxid = 0

serialNo = txtSerialNumber.Text

Department = cmbDepartment.Text

purchaseDate = DTPicker1.Value

UserID = Val(txtUserID.Text)

If AssetName = "" Or num = Null Or num = 0 Or Manufacturer = "" Or status = "SELECT" Or serialNo = "" Or Department = "SELECT" Or status = "" Or Department = "" Or UserID = Null Or UserID = 0 Then

MsgBox "All Fields Are Required", vbExclamation

Exit Sub

End If

DatDepartment.RecordSource = "SELECT \* FROM Department WHERE DepartmentName='" & Department & "'"

If Not DatMaintenance.Recordset.BOF Then

DatMaintenance.Recordset.MoveFirst

End If

Do Until DatMaintenance.Recordset.EOF

If DatMaintenance.Recordset.Fields("MaintenanceID") > maxid Then

maxid = DatMaintenance.Recordset.Fields("MaintenanceID")

End If

DatMaintenance.Recordset.MoveNext

Loop

If purchaseDate > Now Then

MsgBox "Enter a valid date or set Date and time", vbCritical

Exit Sub

End If

If route = False Then

DatMaintenance.RecordSource = "SELECT \* FROM Maintenance WHERE AssetID=" & DatAssets.Recordset.Fields("AssetID")

DatAssets.Recordset.Edit

DatAssets.Recordset.Fields("AssetName") = AssetName

DatAssets.Recordset.Fields("Manufacturer") = Manufacturer

DatAssets.Recordset.Fields("Status") = status

DatAssets.Recordset.Fields("SerialNumber") = serialNo

DatAssets.Recordset.Fields("DepartmentID") = DatDepartment.Recordset.Fields("DepartmentID")

DatAssets.Recordset.Fields("PurchaseDate") = purchaseDate

DatAssets.Recordset.Fields("UserID") = UserID

DatAssets.Recordset.Update

DatMaintenance.Recordset.Edit

DatMaintenance.Recordset.Fields("MaintenanceInterval") = num

DatMaintenance.Recordset.Fields("NextMaintenance") = DateAdd("d", num, purchaseDate)

DatMaintenance.Recordset.Update

MsgBox "Asset Of id:" & DatAssets.Recordset.Fields("AssetID") & " Updated successfully", vbInformation

txtAssetName.Text = ""

txtManufacturer.Text = ""

txtSerialNumber.Text = ""

txtUserID.Text = ""

txtAssetID.Text = ""

Exit Sub

End If

DatMaintenance.Recordset.AddNew

DatMaintenance.Recordset.Fields("MaintenanceID") = maxid + 1

DatMaintenance.Recordset.Fields("AssetID") = i

DatMaintenance.Recordset.Fields("MaintenanceInterval") = num

DatMaintenance.Recordset.Fields("NextMaintenance") = DateAdd("d", num, purchaseDate)

DatMaintenance.Recordset.Fields("TechnicianID") = 0

DatMaintenance.Recordset.Update

DatAssets.Recordset.AddNew

DatAssets.Recordset.Fields("AssetName") = AssetName

DatAssets.Recordset.Fields("Manufacturer") = Manufacturer

DatAssets.Recordset.Fields("Status") = status

DatAssets.Recordset.Fields("SerialNumber") = serialNo

DatAssets.Recordset.Fields("DepartmentID") = DatDepartment.Recordset.Fields("DepartmentID")

DatAssets.Recordset.Fields("PurchaseDate") = purchaseDate

DatAssets.Recordset.Fields("UserID") = UserID

DatAssets.Recordset.Fields("AssetID") = i

DatAssets.Recordset.Update

MsgBox "Asset Issuance Success!", vbInformation

Unload Me

Load frmDash

frameIssueAssets.Visible = True

frmDash.Caption = Role

frmDash.Show

txtAssetName.Text = ""

txtManufacturer.Text = ""

txtSerialNumber.Text = ""

txtUserID.Text = ""

txtAssetID.Text = ""

End If

If Role = "AssetUser" Then

DatNotification.Refresh

If txtAssetName.Text = "" Or txtManufacturer = "" Then

MsgBox "Choose Asset To Report", vbExclamation

Exit Sub

End If

If txtDescription.Text = "" Then

MsgBox "Enter Description!", vbExclamation

Exit Sub

End If

If Not DatNotification.Recordset.BOF Then

DatNotification.Recordset.MoveFirst

End If

Do Until DatNotification.Recordset.EOF

If DatNotification.Recordset.Fields("NotificationID") > max\_id Then

max\_id = DatNotification.Recordset.Fields("NotificationID")

End If

DatNotification.Recordset.MoveNext

Loop

DatNotification.Recordset.AddNew

DatNotification.Recordset.Fields("Title") = txtAssetName.Text & " Issue"

DatNotification.Recordset.Fields("Message") = txtDescription.Text

DatNotification.Recordset.Fields("Status") = "Unread"

DatNotification.Recordset.Fields("NotificationDate") = Now

DatNotification.Recordset.Fields("NotificationID") = max\_id + 1

'DatNotification.Recordset.Fields("UserID") = CurrentUserID

DatNotification.Recordset.Fields("AssetID") = txtAssetID.Text

DatNotification.Recordset.Update

MsgBox "Issue Reported Successfully!", vbInformation

Unload Me

Load frmDash

frmDash.Caption = Role

frmDash.Show

txtAssetName.Text = ""

txtManufacturer.Text = ""

txtSerialNumber.Text = ""

txtUserID.Text = ""

txtAssetID.Text = ""

End If

If Role = "Technician" Then

If txtAssetName.Text = "" Or txtManufacturer = "" Or txtAssetID.Text = 0 Or txtAssetID.Text = "" Or txtDescription.Text = "" Then

MsgBox "Choose Asset To Report", vbExclamation

Exit Sub

End If

DatNotification.Recordset.AddNew

DatNotification.Recordset.Fields("Title") = txtAssetName.Text & "Issue"

DatNotification.Recordset.Fields("Message") = txtDescription.Text

DatNotification.Recordset.Fields("Status") = "Unread"

DatNotification.Recordset.Fields("NotificationDate") = Now

DatNotification.Recordset.Fields("NotificationID") = max\_id + 1

'DatNotification.Recordset.Fields("UserID") = CurrentUserID

DatNotification.Recordset.Fields("AssetID") = txtAssetID.Text

DatNotification.Recordset.Update

MsgBox "Requested Successfully!", vbInformation

Unload Me

Load frmDash

frmDash.Caption = Role

frmDash.Show

txtAssetName.Text = ""

txtManufacturer.Text = ""

txtSerialNumber.Text = ""

txtUserID.Text = ""

txtAssetID.Text = ""

End If

End Sub

### 5.2.4 Maintenance Module

The Maintenance Module in the Automated IT Asset Maintenance and Tracking System facilitates the assignment, tracking, and monitoring of maintenance tasks. It accommodates three roles:

Admin: Assigns technicians to specific maintenance tasks.

Technician: Views and updates their assigned tasks, including remarks and completion status.

Asset User: Can view their asset’s maintenance status and history.

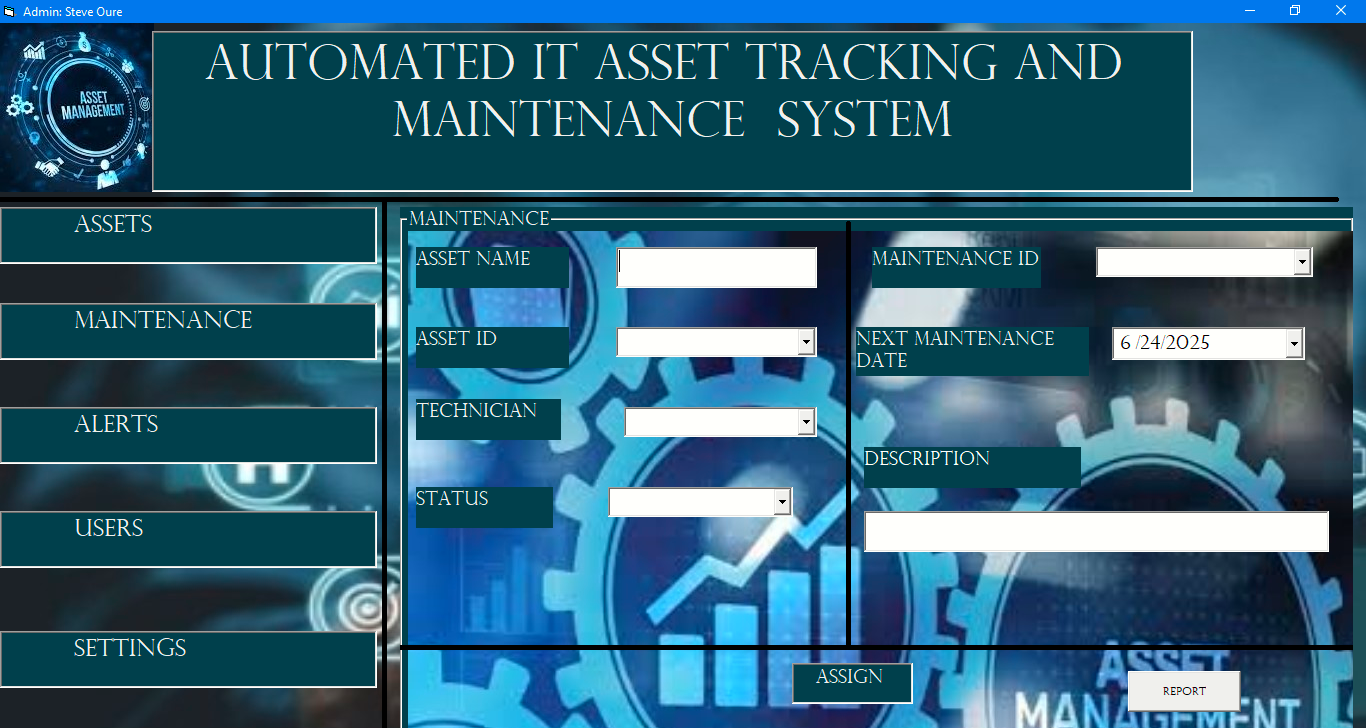


Figure 12 Maintenance Module

Private Sub lblMaintenance\_Click(Index As Integer)

frameIssueAssets.Visible = False

frameNotifications.Visible = False

frameUser.Visible = False

If frameMaintenance.Visible = False Then

frameMaintenance.Visible = True

Else

frameMaintenance.Visible = False

End If

End Sub

Private Sub cmbMAssetID\_Click()

If Not DatAssets.Recordset.BOF Then

DatAssets.Recordset.MoveFirst

End If

Do Until DatAssets.Recordset.EOF

If DatAssets.Recordset.Fields("AssetID") = Val(cmbMAssetID.Text) Then

txtMAssetName.Text = DatAssets.Recordset.Fields("AssetName")

End If

DatAssets.Recordset.MoveNext

Loop

If Not DatMaintenance.Recordset.BOF Then

DatMaintenance.Recordset.MoveFirst

End If

Do Until DatMaintenance.Recordset.EOF

If DatMaintenance.Recordset.Fields("AssetID") = Val(cmbMAssetID.Text) Then

cmbMMaintenanceID.Text = DatMaintenance.Recordset.Fields("MaintenanceID")

cmbMTechnicianID.Text = DatMaintenance.Recordset.Fields("TechnicianID")

If Not DatMaintenance.Recordset.Fields("Status") = Null Then

cmbMStatus.Text = DatMaintenance.Recordset.Fields("Status")

End If

If Not DatMaintenance.Recordset.Fields("Description") = Null Then

txtDescription.Text = DatMaintenance.Recordset.Fields("Description")

End If

DTPicker1.Value = DatMaintenance.Recordset.Fields("NextMaintenance")

End If

DatMaintenance.Recordset.MoveNext

Loop

End Sub

Private Sub cmbMMaintenanceID\_Click()

'DatMaintenance.RecordSource = "SELECT \* FROM Maintenance WHERE MaintenanceID=" & Val(cmbMMaintenanceID.Text)

' DatAssets.RecordSource = "SELECT \* FROM Assets WHERE AssetID=" &

If Not DatAssets.Recordset.BOF Then

DatAssets.Recordset.MoveFirst

End If

If Not DatMaintenance.Recordset.BOF Then

DatMaintenance.Recordset.MoveFirst

End If

Do Until DatMaintenance.Recordset.EOF

DatAssets.Recordset.MoveFirst

Do Until DatAssets.Recordset.EOF

If cmbMMaintenanceID.Text = DatAssets.Recordset.Fields("AssetID") And DatMaintenance.Recordset.Fields("AssetID") = DatAssets.Recordset.Fields("AssetID") Then

txtMAssetName.Text = DatAssets.Recordset.Fields("AssetName")

cmbMAssetID.Text = DatAssets.Recordset.Fields("AssetID")

If Not DatMaintenance.Recordset.Fields("Status") = Null Then

cmbMStatus.Text = DatMaintenance.Recordset.Fields("Status")

End If

If Not DatMaintenance.Recordset.Fields("Description") = Null Then

txtMDescription.Text = DatMaintenance.Recordset.Fields("Description")

End If

DTPicker2.Value = DatMaintenance.Recordset.Fields("NextMaintenance")

If DTPicker2.Value < Now Then

cmbMStatus.Text = "Missed"

End If

Exit Sub

End If

DatAssets.Recordset.MoveNext

Loop

DatMaintenance.Recordset.MoveNext

Loop

End Sub

Private Sub Command1\_Click()

Adodc1.RecordSource = "SELECT \* FROM Maintenance WHERE AssetID=" & Val(cmbMAssetID.Text)

Adodc1.Refresh

Set DataReport2.DataSource = Adodc1

DataReport2.Caption = "MAINTENANCE REPORT"

DataReport2.Show

End Sub

Private Sub lblMAssign\_Click()

If Role = "Admin" Then

If cmbMTechnicianID.Text = "" Or cmbMMaintenanceID.Text = "" Then

MsgBox "Enter Required Fields To Assign", vbExclamation

Exit Sub

End If

DatMaintenance.Recordset.Edit

If Not txtMDescription.Text = "" Then

DatMaintenance.Recordset.Fields("Description") = txtMDescription

End If

If Not cmbMStatus.Text = "" Then

DatMaintenance.Recordset.Fields("Status") = cmbMStatus.Text

End If

DatMaintenance.Recordset.Fields("TechnicianID") = cmbMTechnicianID

DatMaintenance.Recordset.Update

MsgBox "Task Assigned", vbInformation

Unload Me

Load frmDash

frmDash.Caption = Role

frameMaintenance.Visible = True

frmDash.Caption = Role

frmDash.Show

End If

If Role = "Technician" Then

If cmbMAssetID.Text = "" Or txtMDescription.Text = "" Or cmbMMaintenanceID.Text = "" Or cmbMStatus.Text = "" Then

MsgBox "Enter data to Proceed", vbExclamation

Exit Sub

End If

DatMaintenance.Recordset.Edit

DatMaintenance.Recordset.Fields("Description") = txtMDescription

DatMaintenance.Recordset.Fields("Status") = cmbMStatus.Text

DatMaintenance.Recordset.Fields("TechnicianID") = CurrentUserID

DatMaintenance.Recordset.Update

MsgBox "Task Assigned", vbInformation

Unload Me

frmDash.Caption = Role

frameMaintenance.Visible = True

frmDash.Show

End If

End Sub

### 5.2.5 Notification and Alerts Module

The Notification and Alerts Module is responsible for alerting users about key events and updates related to IT assets and maintenance activities. Notification help ensure timely action, reduce asset downtime, and enhance user accountability.

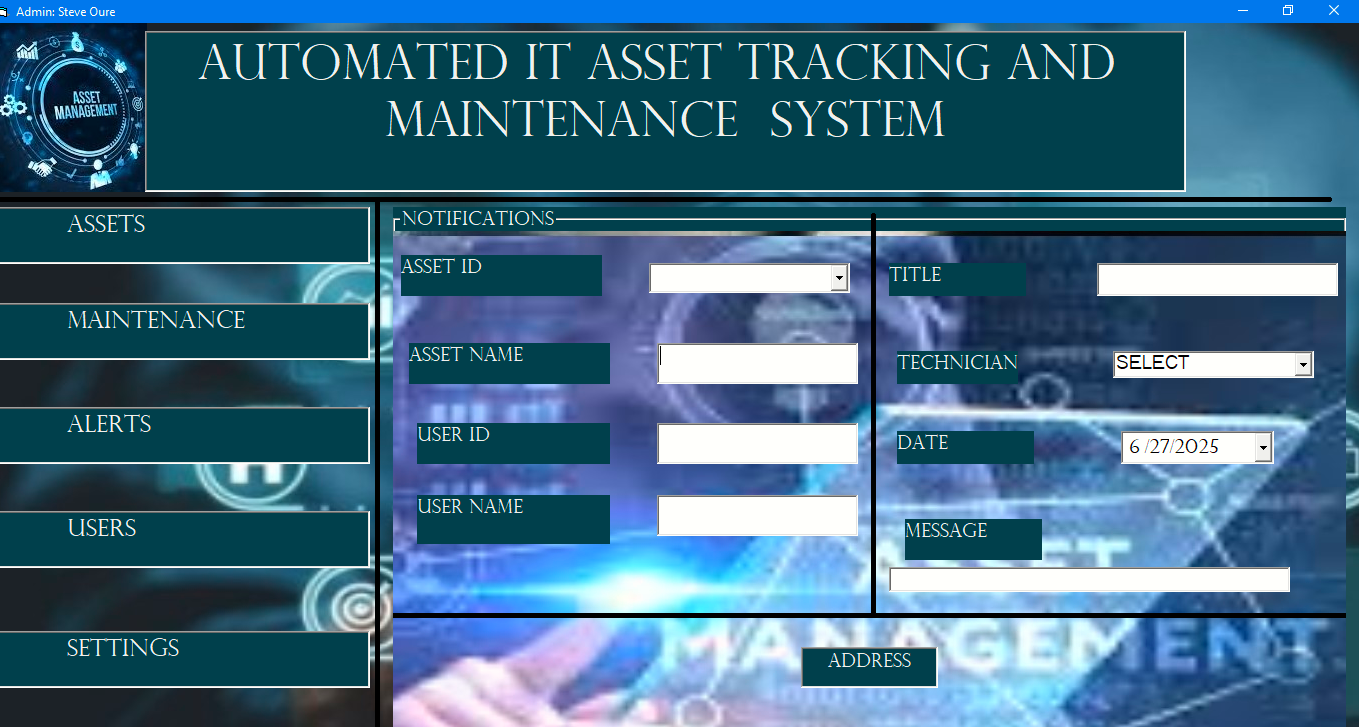


Figure 13 Alerts Module

Private Sub lblAlerts\_Click()

frameIssueAssets.Visible = False

frameMaintenance.Visible = False

frameUser.Visible = False

If frameNotifications.Visible = False Then

frameNotifications.Visible = True

Else

frameNotifications.Visible = False

End If

End Sub

Private Sub cmbAAssetID\_Click()

DatUsers.Refresh

DatNotification.Refresh

If Not DatAssets.Recordset.BOF Then

DatAssets.Recordset.MoveFirst

End If

If Not DatNotification.Recordset.BOF Then

DatNotification.Recordset.MoveFirst

End If

Do Until DatNotification.Recordset.EOF

DatAssets.Recordset.MoveFirst

Do Until DatAssets.Recordset.EOF

If DatAssets.Recordset.Fields("AssetID") = DatNotification.Recordset.Fields("AssetID") Then

DatUsers.RecordSource = "SELECT \* FROM User WHERE UserID=" & DatAssets.Recordset.Fields("UserID").Value

txtAAssetName.Text = DatAssets.Recordset.Fields("AssetName")

txtAUserID.Text = DatAssets.Recordset.Fields("UserID")

txtAUserName.Text = DatUsers.Recordset.Fields("Name")

txtATitle.Text = DatNotification.Recordset.Fields("Title")

DTPicker3.Value = DatNotification.Recordset.Fields("NotificationDate")

txtAMessage.Text = DatNotification.Recordset.Fields("Message")

DatNotification.Recordset.Edit

DatNotification.Recordset.Fields("Status") = "Read"

DatNotification.Recordset.Update

lblAlerts.BackColor = &H404000

End If

DatAssets.Recordset.MoveNext

Loop

DatNotification.Recordset.MoveNext

Loop

End Sub

### 5.2.6 User Module

The User Module is a core component of the system that facilitates secure and role-based access control. It allows the administrator to manage system users by adding, updating or deactivating accounts, while ensuring that each user is granted appropriate permissions based on their role.

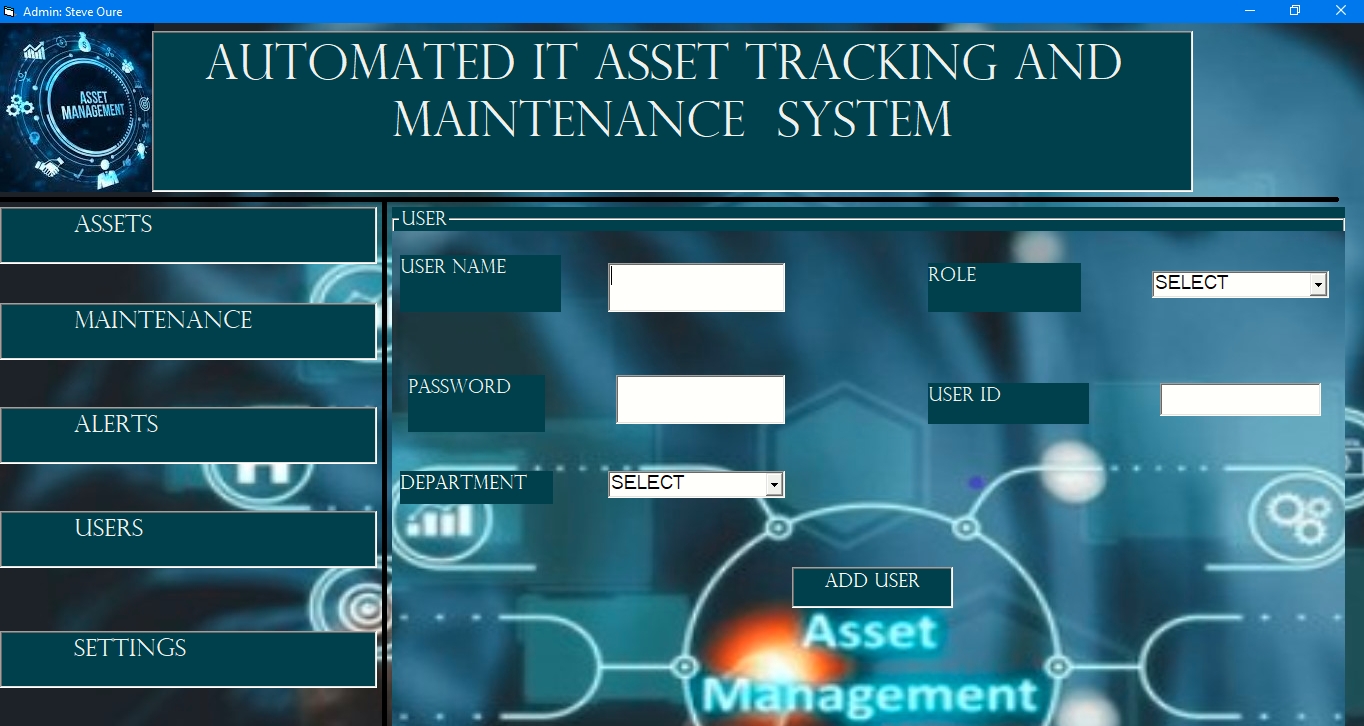


Figure 14 user Module

Private Sub txtUUserName\_Change()

If frameUser.Caption = "USER" Then

If txtUUserName.Text = "" Then

txtUUserID.Text = ""

Exit Sub

End If

Dim maxid As Long

maxid = 0

DatUsers.Refresh

If Not DatUsers.Recordset.BOF Then

DatUsers.Recordset.MoveFirst

End If

Do Until DatUsers.Recordset.EOF

If DatUsers.Recordset.Fields("UserID") > maxid Then

maxid = DatUsers.Recordset.Fields("UserID")

End If

DatUsers.Recordset.MoveNext

Loop

txtUUserID.Text = maxid

Exit Sub

End If

End Sub

Private Sub lblUUser\_Click()

If txtUPassword.Text = "" Then

MsgBox "Enter Password!", vbExclamation

Exit Sub

End If

Dim departmentid As Long

If Role = "Admin" And frameUser.Caption = "USER" Then

If txtUUserName = "" Or txtUPassword.Text = "" Or cmbURole.Text = "" Or cmbURole.Text = "SELECT" Then

MsgBox "All Fields Are Required", vbExclamation

Exit Sub

End If

If Not DatDepartment.Recordset.BOF Then

DatDepartment.Recordset.MoveFirst

End If

Do Until DatDepartment.Recordset.EOF

If DatDepartment.Recordset.Fields("DepartmentName") = cmbUDepartment.Text Then

departmentid = DatDepartment.Recordset.Fields("DepartmentID")

End If

DatDepartment.Recordset.MoveNext

Loop

DatUsers.Recordset.AddNew

DatUsers.Recordset.Fields("Name") = txtUUserName.Text

DatUsers.Recordset.Fields("Password") = txtUPassword.Text

DatUsers.Recordset.Fields("Status") = Offline

DatUsers.Recordset.Fields("Role") = txtURole.Text

DatUsers.Recordset.Fields("UserID") = txtUUserID.Text

DatUsers.Recordset.Fields("DepartmentID") = departmentid

DatUsers.Recordset.Update

MsgBox "User Added Successfully", vbInformation

txtUUserName.Text = ""

txtUPassword.Text = ""

Exit Sub

End If

DatUsers.Recordset.Edit

DatUsers.Recordset.Fields("Password") = txtUPassword.Text

DatUsers.Recordset.Update

MsgBox "Password Changed!", vbInformation

End Sub

### 5.2.7 Settings Module

The Settings Module empower system users to manage their personal account settings, specifically the ability to change their login password securely. This Module enhances user control and overall system security by allowing regular updates to authentication credentials.

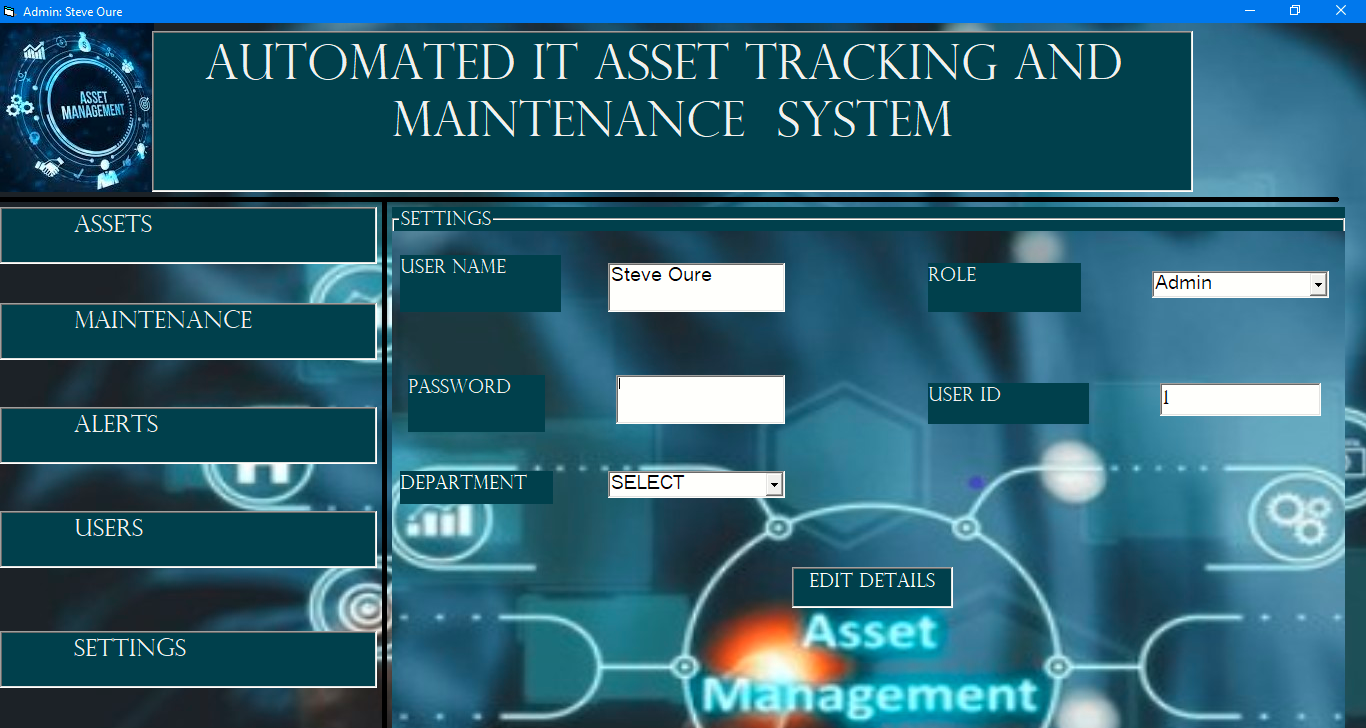


Figure 15 Settings Module

Private Sub lblUUser\_Click()

If txtUPassword.Text = "" Then

MsgBox "Enter Password!", vbExclamation

Exit Sub

End If

Dim departmentid As Long

If Role = "Admin" And frameUser.Caption = "USER" Then

If txtUUserName = "" Or txtUPassword.Text = "" Or cmbURole.Text = "" Or cmbURole.Text = "SELECT" Then

MsgBox "All Fields Are Required", vbExclamation

Exit Sub

End If

If Not DatDepartment.Recordset.BOF Then

DatDepartment.Recordset.MoveFirst

End If

Do Until DatDepartment.Recordset.EOF

If DatDepartment.Recordset.Fields("DepartmentName") = cmbUDepartment.Text Then

departmentid = DatDepartment.Recordset.Fields("DepartmentID")

End If

DatDepartment.Recordset.MoveNext

Loop

DatUsers.Recordset.AddNew

DatUsers.Recordset.Fields("Name") = txtUUserName.Text

DatUsers.Recordset.Fields("Password") = txtUPassword.Text

DatUsers.Recordset.Fields("Status") = Offline

DatUsers.Recordset.Fields("Role") = txtURole.Text

DatUsers.Recordset.Fields("UserID") = txtUUserID.Text

DatUsers.Recordset.Fields("DepartmentID") = departmentid

DatUsers.Recordset.Update

MsgBox "User Added Successfully", vbInformation

txtUUserName.Text = ""

txtUPassword.Text = ""

Exit Sub

End If

DatUsers.Recordset.Edit

DatUsers.Recordset.Fields("Password") = txtUPassword.Text

DatUsers.Recordset.Update

MsgBox "Password Changed!", vbInformation

End Sub

### 5.3 Comprehensive Reports

The Reports Module is integrated into various parts of the system, enabling users to generate, view and print meaningful insights derived from the IT asset records. These reports support data-driven decision-making and improve visibility across the organization’s lifecycle. Below are asset details report and maintenance report respectively.

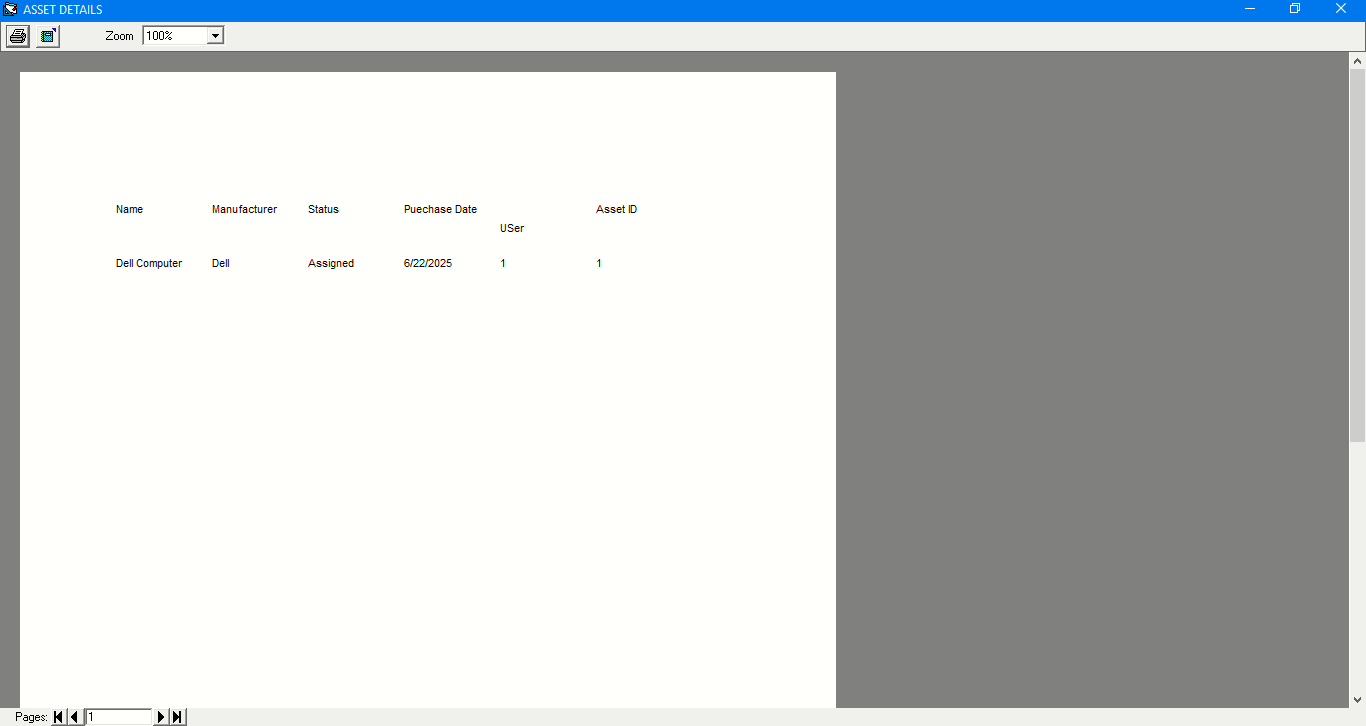


Figure 16 Asset Detail Report

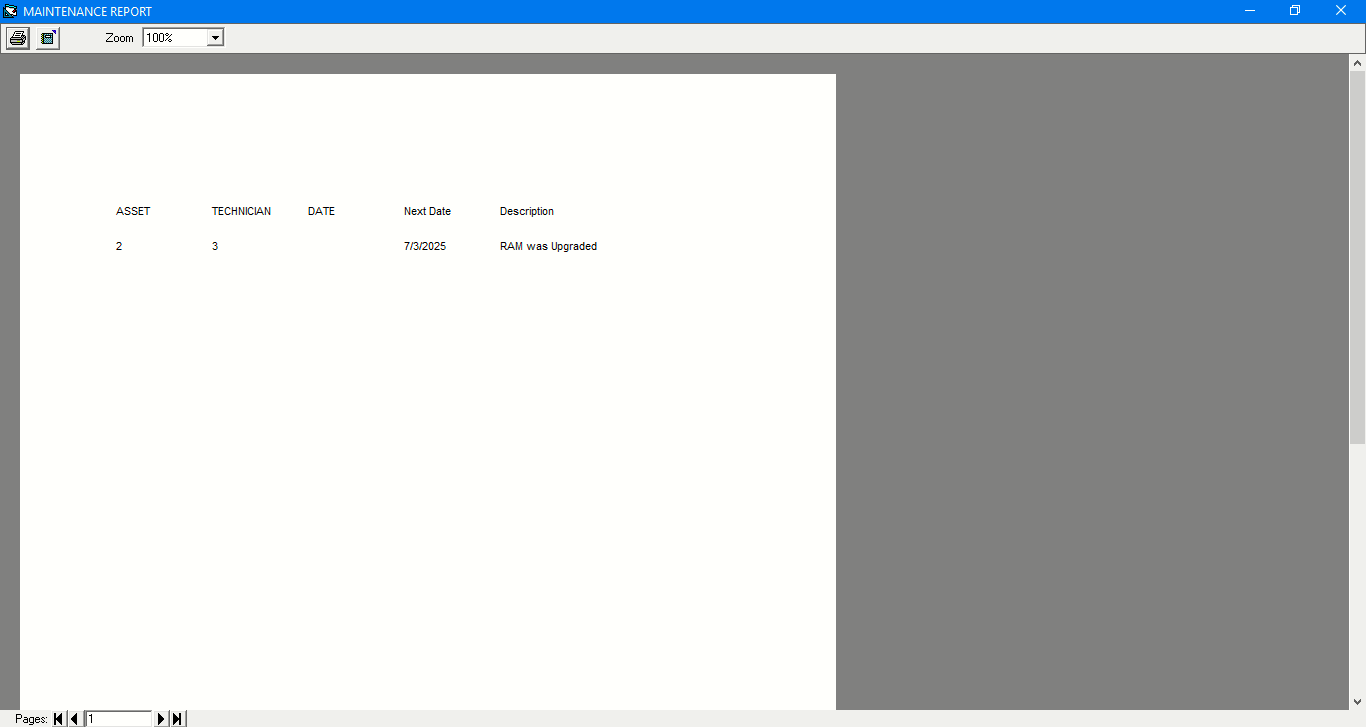


Figure 17 Maintenance report

### 5.2.9 Database Structure

The system uses Microsoft Access as the database backend to store and manage all information related to IT assets, users, maintenance schedules, alerts, and login activity. The integration between the VB6 frontend and the database is done using ADO(Active X Data Objects) and Data Control components, which allow seamless data interaction and navigation between forms and tables. The image below shows the database tables and their fields.

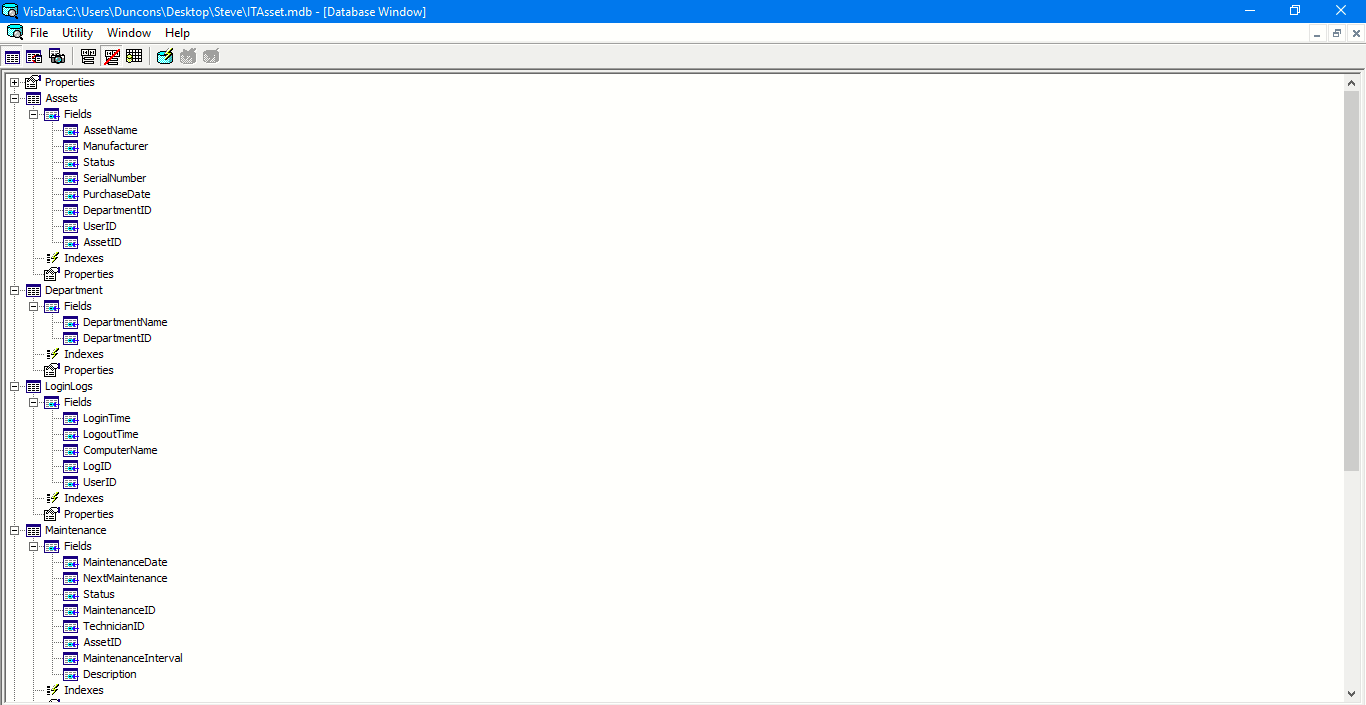


Figure 18 Database and Fields

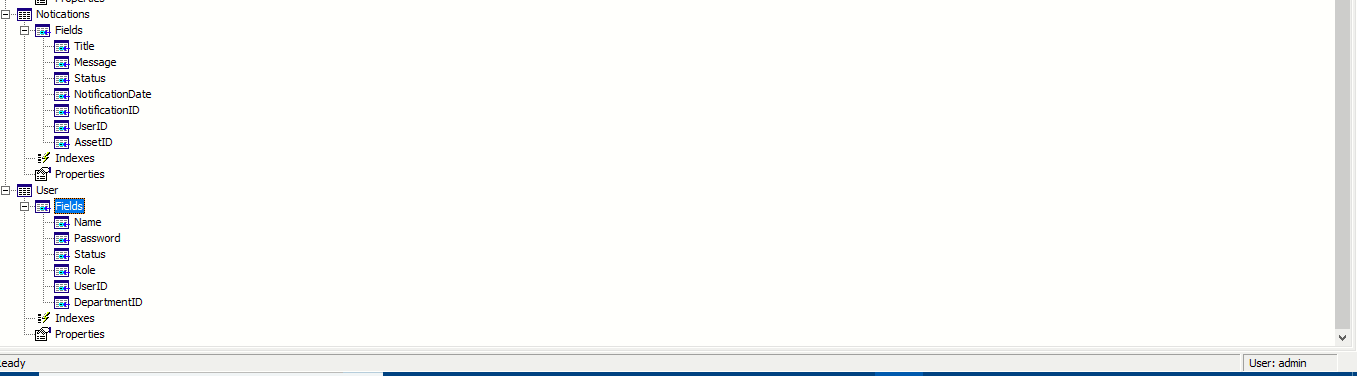


Figure 19 Database and Fields

The following image shows table assets structure and sample data respectively.

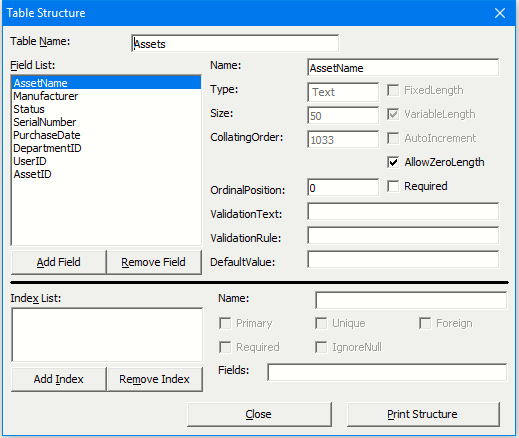


Figure 20 table Assets Structure



Figure 21 Table Asset Sample Data

## 5.3 Discussion

The development and implementation of the Automated IT Asset Maintenance and Tracking System mark a significant step toward modernizing the way IT assets are managed and maintained within an organization. By automating asset tracking, user management, maintenance scheduling, and reporting functionalities, the system offers a structured, secure, and efficient alternative to traditional manual methods, which are often prone to errors, inefficiencies, and inconsistencies.

One of the key highlights of the system is its modular architecture, achieved through the use of frames embedded in a single main form. This design choice helps maintain a unified dashboard experience, where different modules are accessible without navigating between multiple forms. It simplifies system navigation, especially for users with limited technical expertise, and keeps the interface intuitive and user-friendly.

The login and authentication module ensures that only authorized users can access the system, thereby enhancing security and user accountability. With each user assigned a specific role; admin, technician, or Asset User, access rights are restricted according to predefined permissions. The system also logs user login times and logout user times providing valuable audit trails.

In the assets module, the system allows administrators to register new IT assets, track asset details, and assign them to users across departments. Each asset record contains critical information such as acquisition date, serial number, last maintenance date, and Status. This data helps in making informed decisions on asset utilizations, upgrades, or replacements. Additionally, asset user can view the IT resources assigned to them, fostering a sense of responsibility and accountability.

The Maintenance module plays a vital role in proactive asset management. Administrators can assign technicians to specific maintenance tasks, set due dates, and track progress. Technicians, upon logging in, view only their assigned tasks, and asset users can check the status of their devices currently under maintenance. This module helps reduce unplanned downtimes and ensures that maintenance activities are conducted in a timely manner.

The alerts and notification module enhances communication between system users. Alerts are triggered for maintenance, asset expiration, and user enquiries. The Users module facilitates user creation and management. Admins can add new users, assign them roles, and link them to specific departments. The settings module allows each user to change their password, reinforcing security and user autonomy.

Reports are seamlessly embedded across relevant modules and allow printing of asset details, maintenance logs, and user logs. These reports can be used for compliance purposes, internal evaluations, or planning meetings.

From a technical perspective, the use of Visual Basic 6(VB6) and Microsoft Access provides a solid base for local deployment in environments that may lack advanced infrastructure. Although VB6 is outdated compared to modern development environments, it still provides the tools necessary to build a fully functional desktop application with robust data handling capabilities when paired with Access.

Despite the achievement, some challenges we encountered. Data environment registration error was returned when trying to use the data environment. This forced the use of the ADODC ADO data control tool and manually linking it to a data report in code to facilitate the generation of reports.

In conclusion, the Automated IT Asset Maintenance and Tracking System meets its intended goals effectively. It improves asset visibility, ensures scheduled maintenance, streamlines user access, and enhances data security. While there is room for future enhancements, especially in terms of real-time tracking and cloud management in local institutional setups.

## 5.4 Conclusion

The Automated IT Asset Maintenance and Tracking System effectively addresses the challenges of managing IT assets manually. It provides a centralized platform for registering, monitoring, and maintaining assets, ensuring timely servicing and accountability. By automating key processes and enforcing role-based access, the system enhances efficiency, reduces errors, and improves asset visibility across the organization of integrating technology into asset management.

## 5.5 Recommendations

To improve the system’s adaptability in larger organizational settings, future development should focus on integrating the system with other enterprise tools such as inventory, procurement, and finance systems. Such integrations would streamline operations across departments and eliminate redundancy in data entry. Additionally, the system could benefit from centralized data management, especially if scaled across multiple branches, allowing for synchronized asset monitoring from a central point.

Another key area of improvement is system accessibility. While the current VB6-based desktop implementation is functional, developing a web-based or hybrid version of the system would allow users to access it remotely. This would be especially useful for technicians working in different locations or administrators managing assets across various departments. A mobile-friendly version could also enhance convenience and increase system usage in the field.

Security features should also be strengthened further. Future improvements may include multi-factor authentication, encryption of sensitive asset data, and enhanced logging of user actions. This would better safeguard the system against unauthorized access and data breaches. In addition, automating system backups and incorporating disaster recover strategies would improve system resilience.

Finally, enhancing the system’s user experience should be a priority. Features such as graphical dashboards, printable asset cards, and notification systems via email or SMS would make the system more interactive and efficient. Feedback forms or an inbuilt support module could help gather user input and inform future upgrades. These additions would ensure the system remains user-friendly, effective, and aligned with evolving technological trends.

# REFERENCES

Galitz, W. O. (2007). *The essential guide to user interface design: An introduction to GUI design principles and techniques* (3rd ed.). Wiley.

Gupta, M., & Kohli, A. (2006). Enterprise resource planning systems and its implications for operations function. Technovation, 26(5–6), 687–696. https://doi.org/10.1016/j.technovation.2004.10.005

Kumar, A., & Saini, M. (2015). A study of maintenance management in IT asset lifecycle. *International Journal of Engineering Research and Applications*, 5(6), 45–50.

Laudon, K. C., & Laudon, J. P. (2020). *Management information systems: Managing the digital firm* (16th ed.). Pearson Education.

Nielsen, J. (1993). *Usability engineering*. Academic Press.

Nielsen, J. (1994). Heuristic evaluation. In J. Nielsen & R. L. Mack (Eds.), *Usability inspection methods* (pp. 25–62). John Wiley & Sons.

Patel, P., & Doshi, K. (2018). A review on preventive maintenance scheduling using predictive analytics. *International Journal of Computer Applications,* 181(4), 14–18. https://doi.org/10.5120/ijca2018917242

Rouse, M. (2021). IT asset management (ITAM). *TechTarget*. https://www.techtarget.com/searchcio/definition/IT-asset-management-ITAM

Sambamurthy, V., & Zmud, R. W. (2014). Managing IT for digital transformation: The role of enterprise architecture. *MIS Quarterly Executive, 13(2)*, 77–90.

Sharma, V., & Kumar, R. (2020). IT asset management using analytics: A case study. *International Journal of Advanced Computer Science and Applications,* 11(5), 123–129.

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: *Strategies for effective human-computer interaction (6th ed.)*. Pearson.

Stallings, W., & Brown, L. (2018). *Computer security: Principles and practice (4th ed.)*. Pearson Education.

Turban, E., Volonino, L., & Wood, G. (2018). *Information technology for management: Advancing sustainable, profitable business growth* (10th ed.). Wiley.

Zhang, Q., Li, X., & Wang, Y. (2019).RFID-based IT asset management system design and implementation. *Journal of Computer and Communications, 7*(8), 20–29. https://doi.org/10.4236/jcc.2019.78003

Gartner. (2020). *Market guide for IT asset management tools*. Gartner Research. <https://www.gartner.com>

# APPENDICES

## APPENDIX 1: PROJECT BUDGET

|  |  |  |
| --- | --- | --- |
| Item Description | Cost (KES) | Remarks |
| HP EliteBook Laptop | 27000 | Purchased specifically for system development. |
| Internet for Research | 2000 | Estimated for browsing, tutorials and documentation access. |
| Printing and Binding | 1200 | Final project report printing and hardcopy binding. |
| Laptop Charger Replacement | 1500 | Unexpected expense after original charger became faulty. |
| Software Tools; VB6, MS Access | 0 | Freely available |
| Total | 31700 |  |

## APPENDIX 2: PROJECT SCHEDULE (GANTT CHART)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | Jan | Feb | Mar | Apr | May | Jun | Jul |
| Project topic selection and approval |  |  |  |  |  |  |  |
| Requirement gathering and analysis |  |  |  |  |  |  |  |
| Literature review |  |  |  |  |  |  |  |
| System design |  |  |  |  |  |  |  |
| Development(Coding in VB6,MS Access DB) |  |  |  |  |  |  |  |
| Testing and debugging |  |  |  |  |  |  |  |
| Final documentation |  |  |  |  |  |  |  |
| Project submission and presentation |  |  |  |  |  |  |  |