**JUDICIAL CASE MONITORING SYSTEM**

**NAME: CALEB DENZEIL SHIKHUTULI**

**ADMISSION NO: BSCIT-05-0137/2020**

**A Project Report submitted to the Department of Information Technology in the School of Information Technology, Media Engineering in partial fulfilment of the requirements for the award of the degree of Bachelor in information technology of Zetech University**

**APRIL 2023**

**Declaration**

I declare that this research project on Judiciary Case Management System is my own original work. This work should not be edited, copyrighted, republished or any part of this publication without the permission or consent of the author. This work has never been submitted to this University or any other institution for the award of a Diploma, Degree, Master or PhD for examination and academic purposes.

NAME: CALEB DENZEIL SHIKHUTULI

ADMISSION NO: BSCIT-05-0137/2020

SIGNATURE: ……………………………………….

DATE: ………………………………………….

This research has been submitted to Zetech University for examination with my approval as Supervisor.

SUPERVISOR/LECTURER: MRS. JOSEPHINE MAGU

SIGNATURE: …………………………………..

DATE: ……………………………………..

# Dedication

I dedicate this research to my family and friends that have helped me to get where I am and for supporting me in finishing this project. Especially my father for pushing me even if I didn’t want to but I am grateful for everything.

# Acknowledgement

I would like to thank the Lord for giving me the strength to start and finish this project. Secondly, I sincerely express my gratitude to my supervisor Mrs. Josephine Magu for her continued support, guidance and encouragement. I offer my sincere appreciation to the University for providing me with the learning experience.

Contents

[Dedication 2](#_Toc131758003)

[Acknowledgement 3](#_Toc131758004)

[LIST OF FIGURES 6](#_Toc131758005)

[List of Tables 7](#_Toc131758006)

[ABSTRACT 8](#_Toc131758007)

[CHAPTER ONE 9](#_Toc131758008)

[INTRODUCTION 9](#_Toc131758009)

[1.1 Background Information 9](#_Toc131758010)

[1.2 Problem Statement 10](#_Toc131758011)

[1.3 Proposed Solution 11](#_Toc131758012)

[1.4 Objectives 11](#_Toc131758013)

[1.4.1 Main Objective 11](#_Toc131758014)

[1.4.2 Specific Objectives 11](#_Toc131758015)

[1.4.2 Research questions 11](#_Toc131758016)

[1.5 Justification 12](#_Toc131758017)

[1.6 Scope 12](#_Toc131758018)

[1.7 Limitations of the proposed system 12](#_Toc131758019)

[CHAPTER TWO 13](#_Toc131758020)

[1.7 Introduction 13](#_Toc131758021)

[1.8 Theoretical review 13](#_Toc131758022)

[1.9 Critique of the existing system. 14](#_Toc131758023)

[2.0 Research gaps 15](#_Toc131758024)

[2.1 Summary 15](#_Toc131758025)

[CHAPTER 3 16](#_Toc131758026)

[SYSTEM ANALYSIS AND DESIGN 16](#_Toc131758027)

[2.2 Introduction 16](#_Toc131758028)

[2.3 System Development Methodology 16](#_Toc131758029)

[2.4 Feasibility Study 17](#_Toc131758030)

[2.4.1 Technical feasibility 17](#_Toc131758031)

[2.4.2 Economic feasibility 18](#_Toc131758032)

[2.4.3 Operational feasibility 18](#_Toc131758033)

[2.5 Data Collection 18](#_Toc131758034)

[2.6 Data and System Analysis. 19](#_Toc131758035)

[2.7 System Requirements 21](#_Toc131758036)

[2.7.1 Functional requirements 21](#_Toc131758037)

[2.8 Functions of the Application 22](#_Toc131758038)

[DESIGN 22](#_Toc131758039)

[2.9 System design 22](#_Toc131758040)

[3.0 Logical design 22](#_Toc131758041)

[3.0.1 Use case 24](#_Toc131758042)

[3.1 Physical design 26](#_Toc131758043)

[3.2 System Architecture 27](#_Toc131758044)

[3.3 Proposed System Requirements 29](#_Toc131758045)

[3.4 Normalized Database 29](#_Toc131758046)

[3.5 Database Design 33](#_Toc131758047)

[CHAPTER 4 SYSTEM CODE GENERATION AND TESTING, CONCLUSIONS AND RECOMMENDATIONS 35](#_Toc131758048)

[3.6 Introduction 35](#_Toc131758049)

[3.6.1 Admin Page 35](#_Toc131758050)

[3.6.2 Log-in Page 36](#_Toc131758051)

[3.6.3 Case Entry Page 37](#_Toc131758052)

[3.6.4 Code generation 39](#_Toc131758053)

[3.7 Testing 40](#_Toc131758054)

[3.7.1 Usability Testing 41](#_Toc131758055)

[3.7.2 Graphical User Interface Testing 41](#_Toc131758056)

[3.7.2 Performance Testing 42](#_Toc131758057)

[3.8 Summary 43](#_Toc131758058)

[3.9 Conclusion 43](#_Toc131758059)

[4.0 Limitation 43](#_Toc131758060)

[4.1 Recommendation 44](#_Toc131758061)

[References 45](#_Toc131758062)

# LIST OF FIGURES

Figure 1: Waterfall model

Figure 2: Questionnaire form

Figure 3: Pie Chart

Figure 4: Bar Chart

Figure 5: DFD level 0

Figure 6: DFD level 1

Figure 7: Clerk use case diagram

Figure 8: Admin Use case

Figure 9: Physical design

Figure 10: System Architecture

Figure 11: Unnormalized form

Figure 12: First Normal form

Figure 13: Second Normal form

Figure 14: Third Normal form

Figure 15: Client Information Table

Figure 16: Legal Information Table

Figure 17: Admin Page

Figure 18: Log-in details

Figure 19: Log in Page

Figure 20: Case Entry Page

Figure 21: Case Details email

Figure 22: Usability results

Figure 23: GUI results

Figure 24: Performance results

# List of Tables

Table 1: Economic feasibility

# ABSTRACT

In many global justice systems, access to justice is a major concern. The key role of technology in legal systems is to be a potential catalyst for access to justice, particularly in terms of enhancing the justice sector. Registration, case identification and case tracking are significant duties related to court operations.

The purpose of the Judiciary Case Monitoring Software is to record and keep track of cases, investigations or incidents that require action or resolution.

Cases back in the day used to be documented manually on paper and this led to having huge loads of paperwork and it was very cumbersome. Even today some courts still use the old system of recording cases.

The Judiciary cases management system is supposed to provide real-time status on cases whether they are pending or not, case ID, record the type of cases etc. All of that information is stored in a secure database. The system is supposed to be user-friendly, fast and cost-effective

From the information collected on cases, analysis can be done. The system will keep track of new crimes, who committed them, and the results of ongoing investigations, which will free up the judge's time and lessen their workload.

# CHAPTER ONE

# INTRODUCTION

## 1.1 Background Information

The Judiciary is a branch of the government that is responsible for its legal systems and that consists of all the judges in its court of law. Under Chapter 10, Article 159 of the Constitution of Kenya. It is established as the independent custodian of justice in Kenya. Its main or primary objective is to exercise the judicial authority given to it, by the people of Kenya**.** Under the Judiciary, there are courts that are governed and controlled by judges who preside over different cases and lawyers who partake in defending their clients. A lot of paperwork is done before a case is presented in a court and that is where the Judiciary Case Management System comes into use.

The Judiciary ensures that the laws are upheld by the citizens. They work to ensure the protection of human rights and freedoms as envisioned in the Constitution. The courts are divided into a hierarchy structure starting from the highest, the Supreme Court, to Kadhi’s court, which is the lowest.

With the occurrence of Information Communication Technology (ICT). The promotion of justice has been improved greatly. With the possibility to focus on the technical advancement of people who are aware of their rights and obligations and who have a high degree of information technology literacy, opportunities and problems have emerged. ICT has provided management and operational support allowing courts to more effectively manage the magnitude and complexity of their growing caseloads. If ICT is properly used, it can be a significant tool to improve transparency, efficiency and the standards of services provided by courts.

Managing records is an important aspect and function in courts. Among the places where Records management is crucial and serves as the foundation for all other tasks, the justice system is strengthened. It has been noted that among other difficulties, the Judiciary has a significant difficulty of imposition caused by inappropriate placement, loss and harm. Considering this issue, Due to the difficulties with the manual methods, it has been challenging to find justice in this crucial government branch, or else there has been a substantial delay in the process of acquiring paperwork. This is the rationale behind why there should be significant interest in automating the court system.

The courts have a lot of paperwork. The system that I am going to create will record new crimes, ease the process of investigation and reduce the judge’s work during the trial

## 1.2 Problem Statement

Records and information management have been gaining popularity and recognition from the public sector worldwide as governments worldwide use information and communication technology to manage the administration of their corporate records. The current case management system is an e-filling website and requires a person to only submit the case document. This was developed back in 2020 during the Covid pandemic because cases could not be handled physically. It was launched on 1st July 2020 (Victor, 2020). Sometimes the files being submitted can be large and it can be cumbersome for Judges to go through and remember all of the details. I thought to myself, what if I was to create a system that breaks down some of the highlighting details in a document and sends it to the judge which would make things easier? The judge would have an overview of the case before even getting to the actual case.

## 1.3 Proposed Solution

This research seeks to develop case monitoring software using Python to break down important parts of a case document which would reduce the time spent on hearings. By doing this the judge will know what a case file actually entails without actually going through the whole document. Tracking cases will become much easier. Furthermore, the current file system will be improved for better judicial services and help judges administer justice fairly per the law's regulations.

## 1.4 Objectives

### 1.4.1 Main Objective

The main objective is to develop and implement an efficient case management system in Kenyan Courts specifically the lower courts, Subordinate courts that automates case recording.

### 1.4.2 Specific Objectives

1. Track cases by Case ID and Status to provide real-time updates on case progress.

2. Generate proper schedules to efficiently assign court cases and optimize court proceedings.

3. Implement an email-based system to provide judges with a comprehensive overview of each case to facilitate informed decision-making.

4. Create a secure and reliable database system for storing and managing case files and details to ensure easy access and retrieval.

### 1.4.2 Research questions

1. How much time is spent in handling cases in a week?

2. What is the biggest challenge when it comes to handle cases?

## 1.5 Justification

I am developing a case monitoring system to ease the burden on judges and clerks by reducing the amount of time spent on handling cases. This system will be highly beneficial and advantageous to judges particularly when a case requires prior historical references from other judgements. The system will be of advantage to court personnel and administrators by pointing out the deficiencies in the manual system by users. Clients will not complain about missing files because all of their information will be stored in a database. Court cases will be planned out by the system automatically and hearings will take a shorter time. Furthermore, the project will provide other researchers with ample information to create more applicable or sophisticated programs or enhancements.

## 1.6 Scope

This project is based on developing a graphical user interface using python to automate the case management system. To be specific I will use the Tkinter module to create the Interface. The system will be able to run on any computer as long as it meets the minimum requirements. It will be used by clerks to assist judges with case hearings by recording and scheduling It is focused on recording case details to help judges and clerks in the subordinate courts presiding over cases.

## Limitations of the proposed system

1. The training cost. The employees have to be trained on how to use the system which will cost the judiciary both time and money.

2. Maintenance cost**.** The system will require modifications, upgrades and installation which will be costly but the results will be a worthwhile undertaking.

3. Incompatibility with the primary system.

# CHAPTER TWO

## 1.7 Introduction

The use of a case management system refers to the use of technology in the administration of justice. It aims to improve the efficiency and effectiveness of the legal system by automating various processes, including case tracking and effective scheduling. This literature review will provide an overview of the theoretical and empirical studies related to the case management system, including a review of the conceptual framework, the hypothesized variables and a critique of the existing literature.

## 1.8 Theoretical review

In recent years there has been a growing interest in the use of JCMS in courts. Studies have shown that the integration of technology in courts has improved the speed and accuracy of the case managing process, leading to increased efficiency and effectiveness in the court system. The judiciary is necessary for the impartial administration of justice and no civilized can be conceived in the absence of an effective, independent and impartial system for the administration of justice. (Asghar, 2018).

Another important aspect of the case management system is the use of best practices and evidence-based decision-making. This involves the use of data and research to inform the development of JCMS and to evaluate their effectiveness. For example, research has shown that the use of evidence-based decision-making can improve the speed and accuracy of case processing and reduce the risk of errors and delays (Monika, 2020).

The field of electronic filing has progressed significantly in recent years. Research conducted by legal experts suggests that the use of electronic case filing has the capacity to significantly improve the services provided by courts. It is easy to locate various electronic filing initiatives that are currently underway (Nyambane,n.d.).

Studies have shown for example in Europe that technology is rapidly evolving affecting the rule of law lawyers will undergo more significant changes in under two decades than they have in the previous systems have already become capable of predicting judicial verdicts with machine learning as case management software and other tools, which can improve the speed and accuracy of case processing. For example, automating routine tasks, tracking case deadlines, and reducing the risk of filing saves time and resources and shows that successful implementation of the JCMS requires careful planning, and adequate training and between the court and users technology has always been met with resistance. Other services such as e-filing make virtual proceedings possible but there are such as digital filing. (Strikaitė-Latušinskaja,2023).

Some countries are actually integrating their court systems with other technologies for example machine learning models that help with making court decisions from mining and extracting judicial documents. This is being done in Europe and USA. ( Alatrista-Salas, Nunez-del-Prado& Francia, 2022)

One of the key components of JCMS is the use of technology, such as case management software and other tools, which can improve the speed and accuracy of case processing. For example, automating routine tasks, tracking case deadlines, and reducing the risk of errors and delays. Other services such as e-filing save time and resources and make virtual proceedings possible. The study shows that successful implementation of the JCMS requires careful planning, adequate training and between the court and users. The use of sophisticated technology has always been met with resistance but there are indications that such tools will strengthen Judicial decision-making such as digital filing (Zalnieriute & Bell, 2020).

## 1.9 Critique of the existing system.

The existing literature on JCMS has provided valuable insights into the impact of technology on-court performance. However, there is a need for further research to address the gaps in the current knowledge base. In particular, there is a need for studies that examine the long-term impact of JCMS on-court performance and the sustainability of the technology over time. On-court also a need for research on the integration of JCMS with other court systems and the impact of this on court performance.

Additionally, more research needs to be done on the potential risks and challenges associated with the JCMS. While the literature provides some insights into the potential benefits of JCMS, there is limited research on the potential risks and challenges that need to be addressed. This limits our understanding of the full impact of JCMS on the justice system.

## 2.0 Research gaps

Despite the significant progress made in the area of JCMS, there are gaps in the existing literature. There is a need for further research on the long-term impact of JCMS on court performance and the sustainability of the technology over time, as well as the integration of JCMS and other court systems.

## 2.1 Summary

In conclusion, the literature review provides an overview of the recent studies on JCMS and its impact on-court performance. The review highlights the importance of having a technology-driven solution in courts, careful planning and training in the successful implementation of the JCMS. Future research can focus on addressing the research gaps to further contribute on the understanding of the long-term impact of JCMS

# CHAPTER 3

# SYSTEM ANALYSIS AND DESIGN

## 2.2 Introduction

This chapter focuses on the analysis and design of the Subordinates’ Case Management System. This chapter is a critical phase in the project as it provides an in-depth understanding of the system requirements, analysis of collected data, and development of the system specifications. The purpose of the chapter is to outline the approach, methodology, and tools used in the systems development process.

## 2.3 System Development Methodology

The methodology used is water-fall development. The methodology used is a structured approach to creating, modifying, maintaining, and managing a software system. The goal of the development methodology is to improve the efficiency, consistency and quality of the software while ensuring the software meets the user’s needs.

The Systems Development Methodology (SDM) is a framework used to develop and manage software systems. It provides a structured and systematic approach to software development, covering all aspects of the development process from planning to maintenance.

The Waterfall model is a linear and sequential approach to software development. The model is divided into several phases, and each phase must be completed before the next one can begin. The phases include requirements gathering, analysis, design, implementation, testing, deployment and maintenance. Each phase has its own set of deliverables and the outputs of one phase serve as the inputs for the next. In fact, the first model which was widely used in the software industry is the waterfall model (“What is SDLC waterfall model”, 2023).

This model is used in the research to ensure that the project is completed systematically and organized. The research methodology that will be used is qualitative.

**Figure 1: WaterFall Model**



## 2.4 Feasibility Study

A feasibility study was done on the following areas to assess the viability of the project and if it is going to be successful or not:

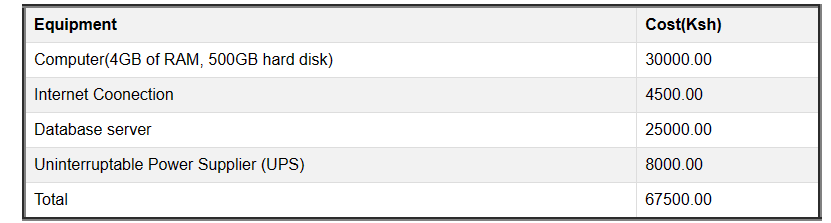
### 2.4.1 Technical feasibility

The technical feasibility of the court system will depend on the availability of the necessary hardware and software. The system will require a computer with a minimum of 4GB of RAM, 500GB hard disk, and a processor speed of at least 2.0GHz. The system will also require the installation of the Python programming language, the Tkinter graphical user interface, and a database management system like MySQL. These requirements are easily met by most modern computers.

### 2.4.2 Economic feasibility

The economic feasibility assessment involved examining the costs and benefits of the project. This included a review of the potential financial benefits of the system, as well as the costs associated with its development, implementation, and maintenance. The costs of developing and implementing the system, including hardware and software, were estimated at ksh.67500.

**Table 1: Economic feasibility**

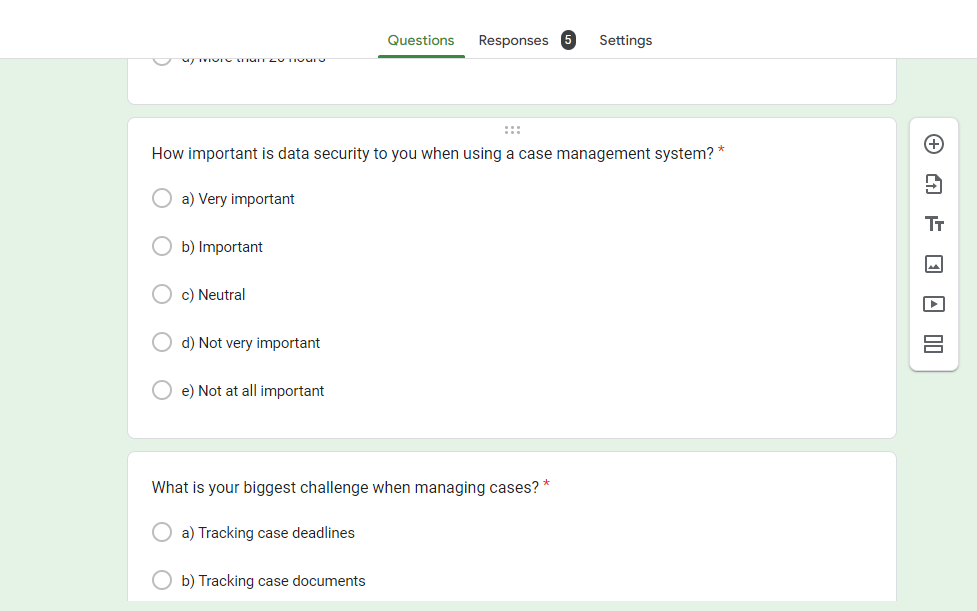


### 2.4.3 Operational feasibility

The operational feasibility assessment involved evaluating whether the system can be integrated into the existing business operations and whether it is acceptable to the stakeholders. It was determined that the system would be easily integrated into existing workflows and that users would be able to quickly learn and use the system. Training and support will be offered to users during the transition period.

## 2.5 Data Collection

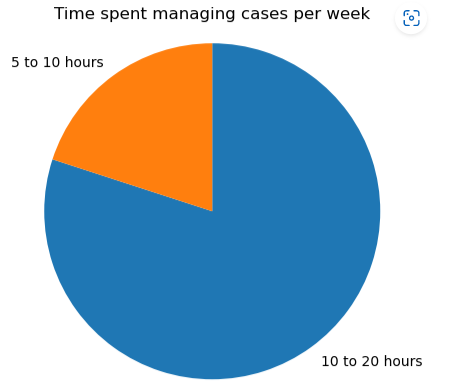
The type of questionnaire used was an online one, In this type, respondents are sent the questionnaire via email or other online mediums. This method is generally cost-effective and time-efficient (Bhat, 2023) .A questionnaire was formulated and administered to a group of 5 clerks in Thika Law courts. and all of them responded to the questionnaire. The number of clerks would have been higher but each court has very few of them. I chose to use a questionnaire because it would be easy for me to analyze the data and my responders would remain anonymous and they would be likely to provide honest answers. Some of the questions that were in the questionnaire are: how much time do they spend handling cases on a typical week, and what is their biggest challenge when it comes to handling cases?

**Figure 2: Questionnaire form**  

## 2.6 Data and System Analysis.

In this section, the data that was obtained from the clerks will be analyzed in order to come up with a system design(logical and physical) based on the data. I downloaded the data from google docs as a CSV file and used jupyter notebooks for the data analysis. The  Jupyter Notebook App is a server-client application that allows editing and running notebook documents via a web browser. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing and computational journalism(“JupyterLab: A Next-Generation Notebook Interface”, 2023). Just by looking at the data using visual assessment. All the clerks handle 5 to 10 cases in a week. From the pie chart below, it is clear that most of the clerks spend 10 to 20 hours on their cases.

**Figure 3: Pie Chart**



All the clerks agreed that their current system does experience delays sometimes and the biggest challenge that they are experiencing is tracking case deadlines.

**Figure 4: Bar Chart**



From the data obtained, a majority of the people using the current system are unsatisfied.

## 2.7 System Requirements

### 2.7.1 Functional requirements

The system should be able to add create, updateand delete user accounts from the database. An admin panel will be made for this function. The users will be able to also add, update and delete cases

The system will have a mechanism for assigning judges cases and informing them of the cases they’re supposed to handle

A mechanism for tracking case deadlines and status will be implemented. Furthermore, to avoid unauthorized entry, the system must include a method of authentication. The system should allow users to search for cases based on different criteria such as case number, defendant name, and case type.

## 2.8 Functions of the Application

The Judicial Case Monitoring Software will be used by a clerk to record case details of a particular client, the name, Judge in charge of that case, type of case, lawyer or attorney in charge of that case, the status of the case, the date that it was recorded and whether any evidence has been presented. This information will be submitted and recorded in a database.

An email will be sent to the judge to inform him/her of the case details and when to appear in court for the hearing. The application will be able to track cases from the beginning to the end. A mechanism shall be implemented to assign cases to judges. The clerk will also be able to create, update and delete case files. Furthermore, they will be an admin panel for adding users.

The system will have an authentication mechanism to prevent unauthorized from

accessing the interface.

# DESIGN

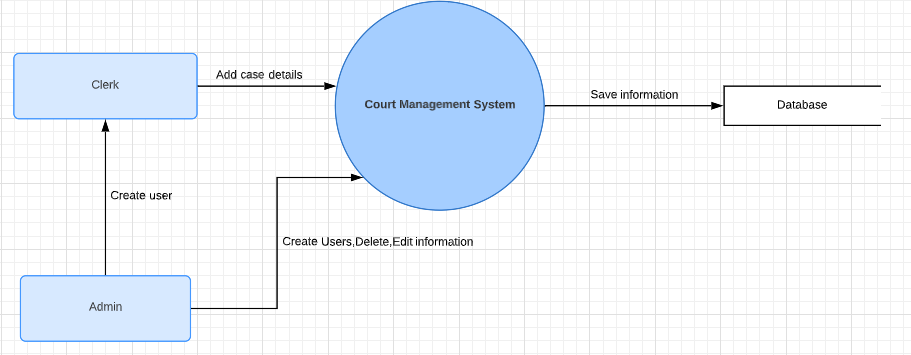
## 2.9 System design

System development involves modifying or creating a system in a way that brings about changes in the processes, practices, and methodologies used for developing it. As a result, it is essential to adopt a methodical approach to managing system requirements and design methodology. The system design will cover logical and physical design. The logical design represents the abstract data flow, while the physical design represents the system’s input and output processes. The system design (Pedamkar, .n.d.)

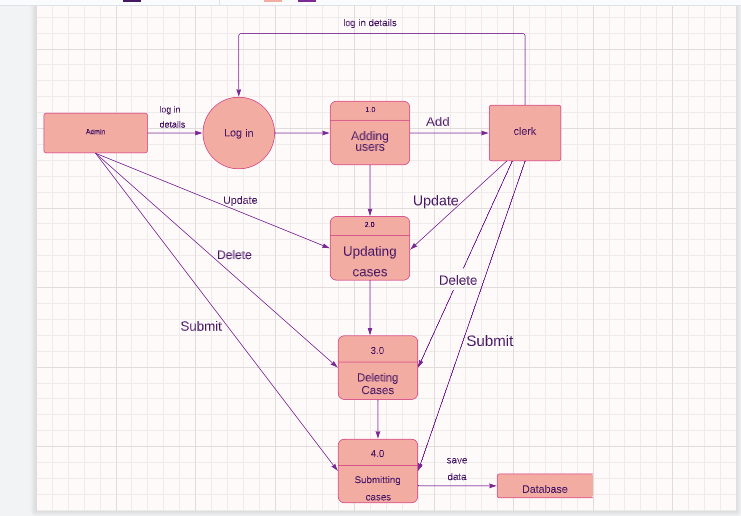
## 3.0 Logical design

The court’s management system is supposed to help clerks with recording cases in an efficient way that will help the judge with administering justice during court trials. The interface will be made using python specifically the Tkinter module and the database used will be MySQL. The diagrams below depict how the users will interact with the system and how the information will flow.

Figure 2 shows the logical flow of data in the system.

**Figure 5: DFD level 0**

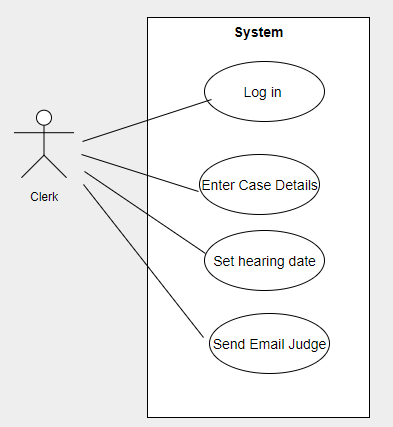
**Figure 6: DFD level 1**

****

### 3.0.1 Use case

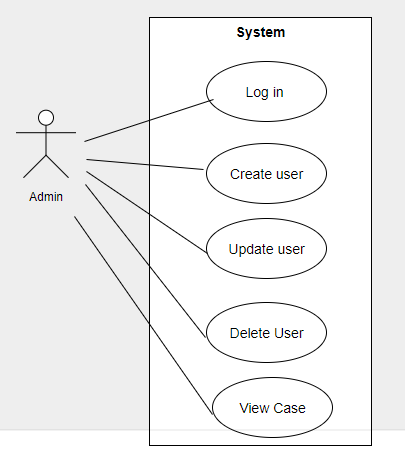
The court clerk in charge of a specific working desk will serve as the user of the system. They will enter the client’s information such as their name, and contacts, as well as the date and time of the recording. Additionally, they will enter the details of the case. Finally, they will set a hearing date for the case. The clerk must provide a password for authentication before being allowed to enter any details into the system. The clerk will have to note down the case details such as the case id, the judge and lawyer in charge of the case, the client, the judge’s details and a description of the case which will be recorded to the database.

**Figure 7: Clerk use case Diagram**



An administrator will oversee tasks like adding, deleting creating updating users. Also adding cases to the database. The administrator will be of great use and help when adding new judges and clerks into the system. The admin can also log in to the main interface to delete cases, update or delete. The admin will have to create an account for the clerk in order to log in to the main interface. Depending on the privileges given to the clerk for example they can be set to be an admin or normal user.

**Figure 8: Admin Use Case**



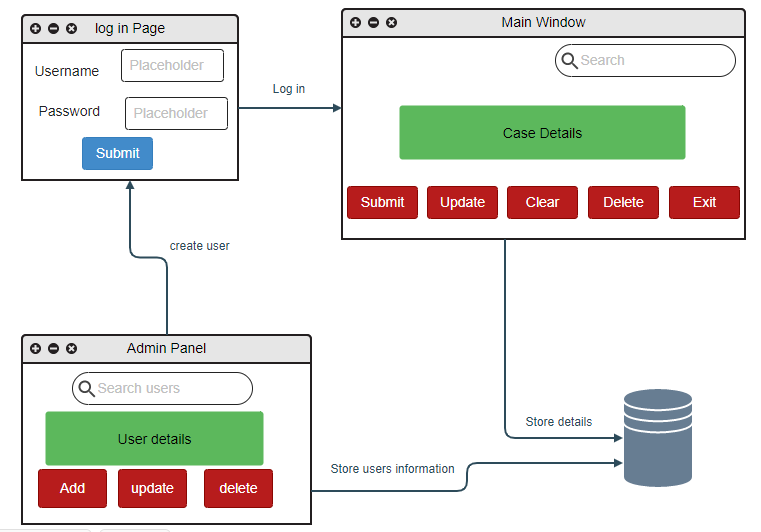
## 3.1 Physical design

The physical design of the case monitoring software using Tkinter involves creating a user-friendly interface that enables users to interact with the software system. This interface should be intuitive and easy to navigate, allowing users to quickly access the different features and functions of the system.

To achieve these objectives, the physical design will be developed using a modular approach. This approach involves dividing the system into smaller components or modules, each of which performs a specific function. Each module will be developed separately, using Tkinter to create the necessary graphical components and linking them together to create the complete system

The physical design depicts how the court management system will look and I used mock-up forms for that depiction.

**Figure 9: Physical design**



## 3.2 System Architecture

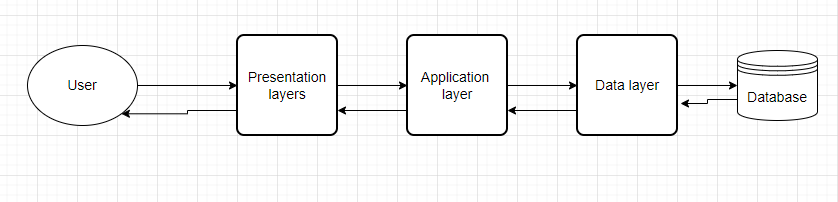
The presentation layer consists of the widgets and controls that the user interacts with such as buttons, this is the main interface with that the user will be interacting. For example, if a user clicks on a button to search for a case, the presentation layer would call a method in the application layer to execute the search and retrieve the results to display in the UI. The purpose of system architecture activities is to define a comprehensive solution based on principles, concepts, and properties logically related to and consistent with each other(Faisandier, Roedler & Adcock, 2022).

The second layer is the application layer, which contains the system's business logic. This layer defines rules and processes for managing judicial cases, such as creating new cases, scheduling a hearing or updating cases. The application layer interacts with the presentation layer to receive user input and display output and also communicates with the data access layer to retrieve and store data. For example, when a user submits a form to create a new case, the application layer would validate the input and then call a method in the data access layer to store the case information in the database.

The data access layer is responsible for retrieving and storing data. This layer interacts with the application layer to receive and provide data and interacts with the database layer to store and retrieve data from the database. In Tkinter, the data access layer might use Python's built-in SQL connector module to interact with a MySQL database. For example, when the application layer needs to retrieve a list of cases for a specific judge, it would call a method in the data access layer to execute an SQL query and return the results.

The database layer includes the DBMS used to store and manage the data for the court system. The database layer communicates with the data access layer to provide data storage and retrieval capabilities. For example, when the data access layer needs to store case information in the database, it would execute an SQL INSERT statement to add the data to the appropriate table.

**Figure 10: System Architecture**

****

## 3.3 Proposed System Requirements

**Software requirements**

* Operating System: typically a modern version of Windows, Linux or macOS.
* Database: MySQL.
* Python 3.8 or higher to be installed
* Tkinter Library to be installed.

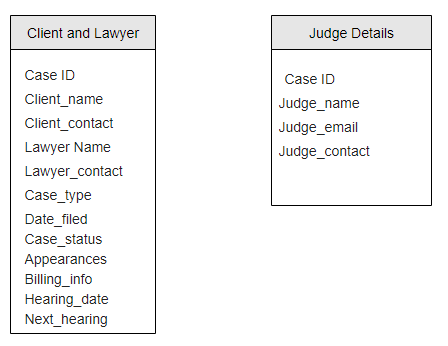
**Hardware requirements**

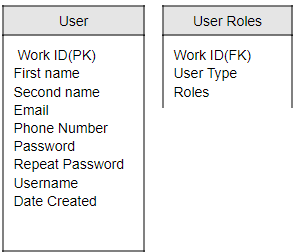
* Processor: Intel Core i3 or higher
* RAM: 4GB or higher
* Hard Disk Space: 500GB or higher
* Display: 1280x768 resolution or higher

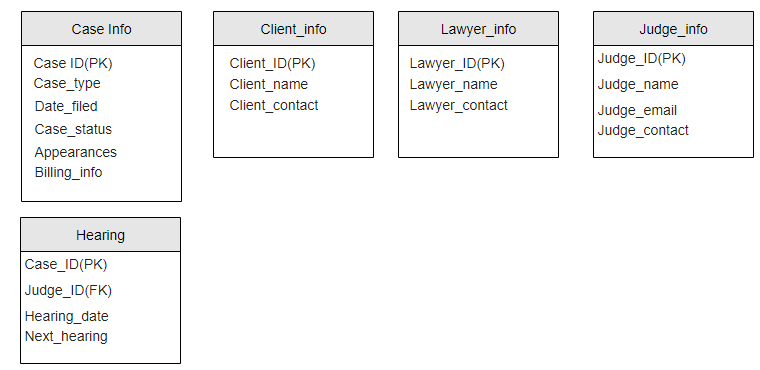
## 3.4 Normalized Database

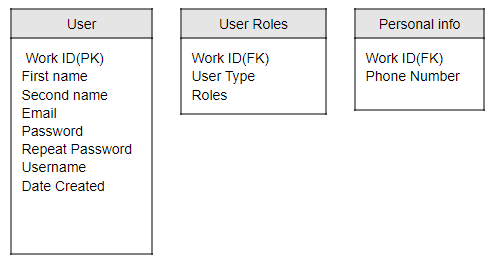
In this section, normalization of the tables that are required will be done. Starting from the first to the third which is the final form.

**Figure 11: Unnormalized form**

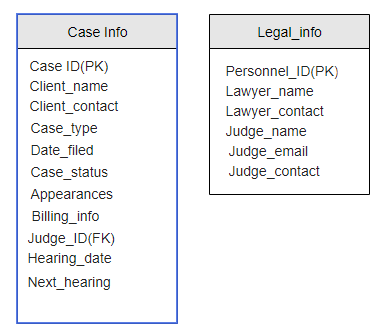


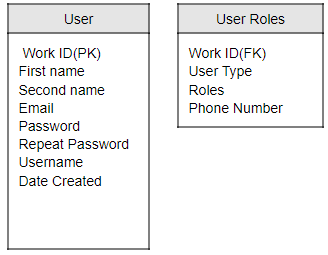


**Figure 12: First Normal Form **

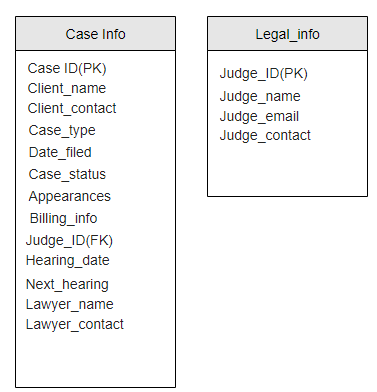
****

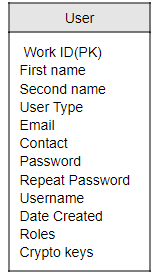
**Figure 13: Second Normal Form**



****

**Figure 14: Third Normal Form**

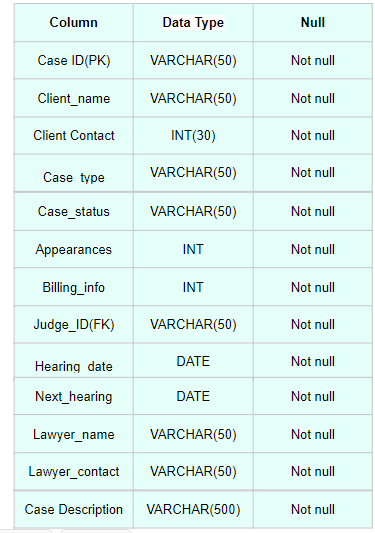




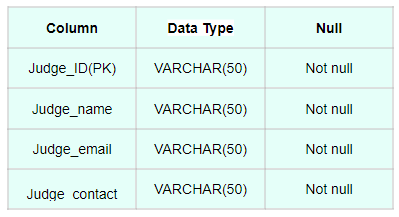
## 3.5 Database Design

The database will have three tables, client, legal and admin information.

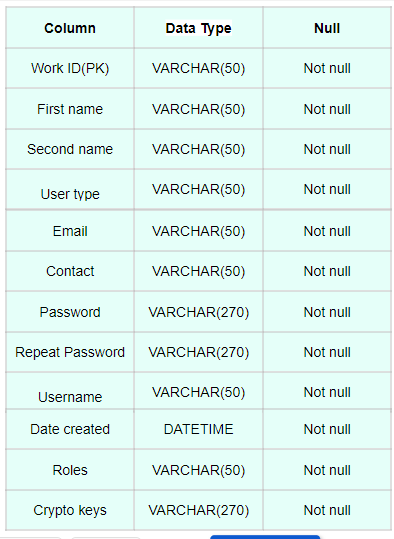
**Figure 15: Client Information Table**



**Figure 16: Legal information Table**

****

**Figure 17: Admin information**

****

The above tables will be created and used in the database for storing information. The Relational database management system that will be used is MySQL. It works on the client-server model. The SQL language will be used to interact with the database from the GUI window for submitting, updating and deletion of information. MySQL is easy to use because you can modify the source to meet your expectations and don’t need to pay anything for this freedom. Another reason is because of its high performance no matter the amount of data that is being processed. (Richard, 2023)

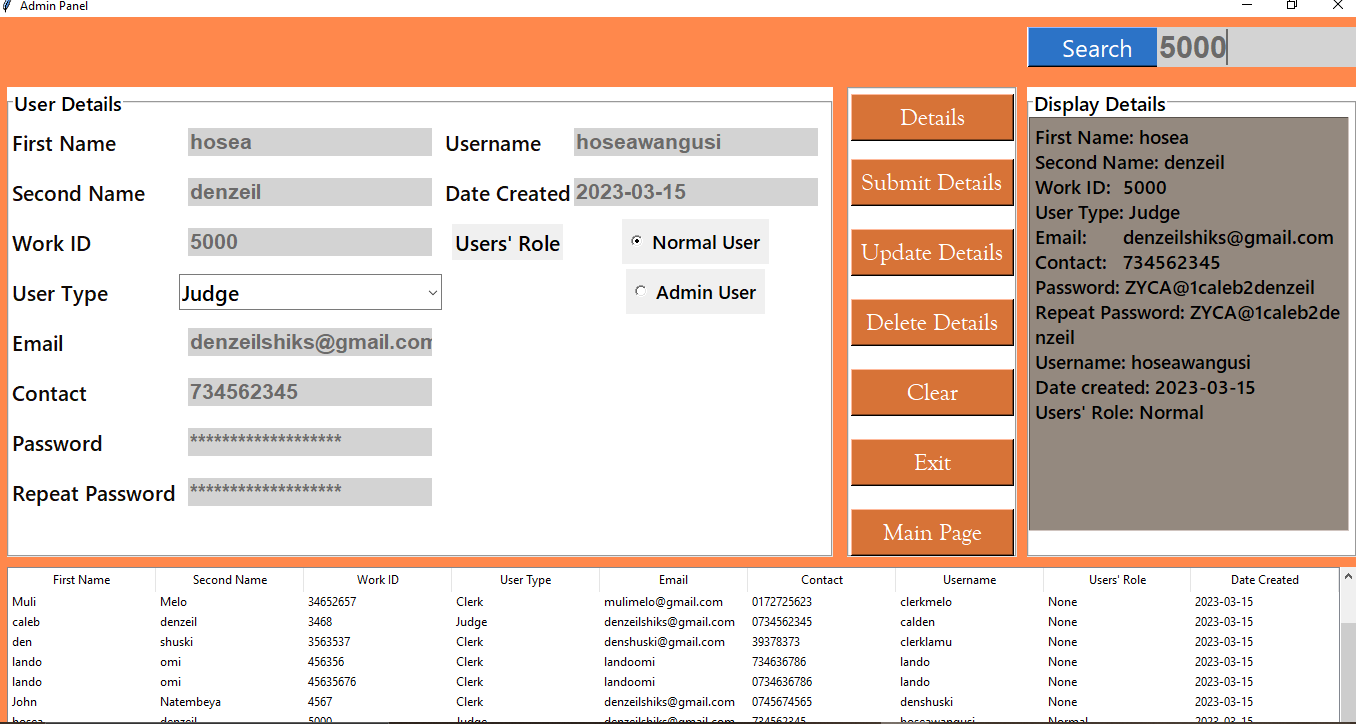
# CHAPTER 4 SYSTEM CODE GENERATION AND TESTING, CONCLUSIONS AND RECOMMENDATIONS

## 3.6 Introduction

The coding of Judicial case monitoring software was coded using Python to be specifically the Tkinter module to create the Graphical User Interface and the database was designed using MySQL workbench. I created three GUI windows that will be used to carry out the system’s designed tasks: The log-in page, Admin page and Case entry page. This chapter involves portraying the actual system code, software integration and testing.

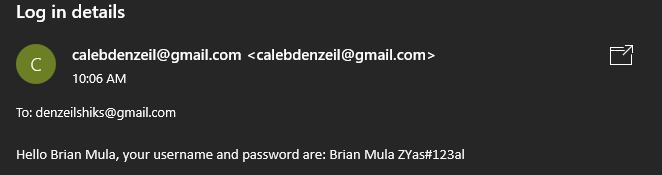
### 3.6.1 Admin Page

Before a user is allowed to access the system, the admin has to create the user’s details and log information. Not just anyone can access the system, a username and password have to be created for a specific user before being allowed to access the system. Some of those details are First and second names, work ID, email, and contact. Also, the day on which the details were created. The user’s information will be recorded in to a database and it can updated or deleted.

****

**Figure 18: Admin Page**

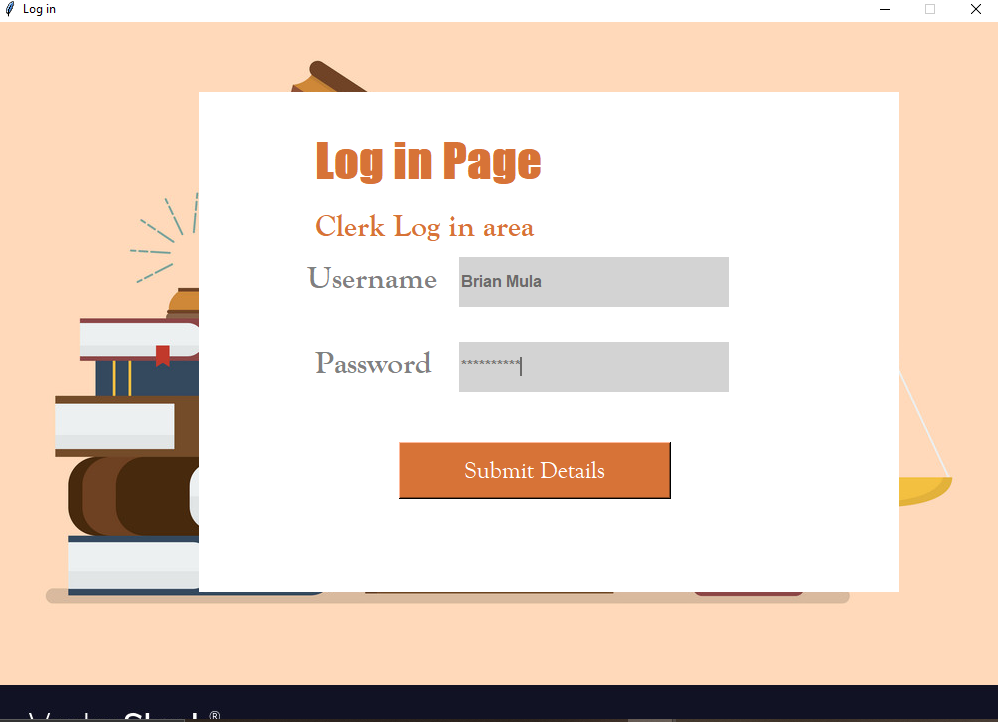
The login details will be sent via email using an API called SMTP, this API connects the system and Google. The user can be an admin or a normal user. After submitting the information to the database. The users’ information will be displayed at the bottom and right.



**Figure 19: Log-in details**

### 3.6.2 Log-in Page

After the users’ account has been created, they can be able to log in as either an admin or a normal user. If the user is an admin, he will be taken to the admin page or if normal, he will be redirected to the main page which is the Case Entry page. The login page requires a username and password. The password that will be entered and retrieved from the database will be encrypted and that is by hashing. Hashing is one-way encryption and it cannot be reverse-engineered, that is why I used it for encryption. Both the encrypted passwords will be compared before granting the user access to the system.

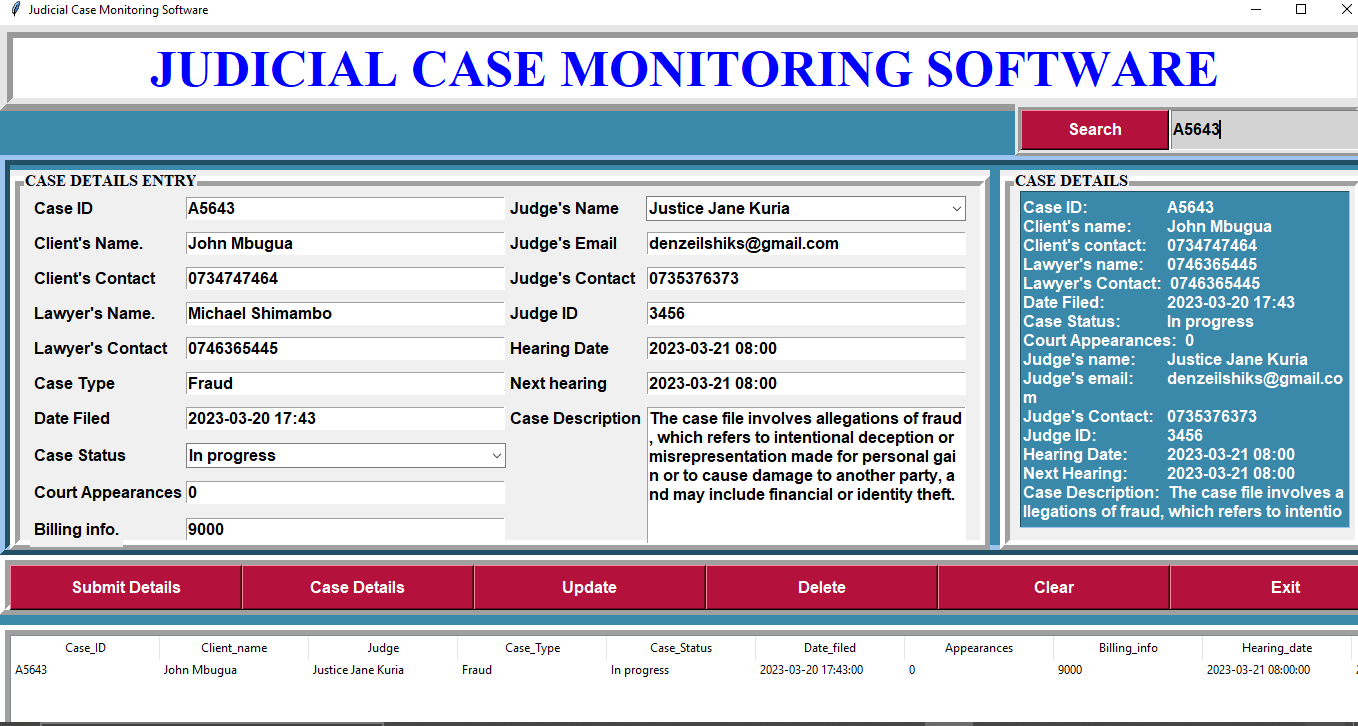


**Figure 20: Log-in Page**

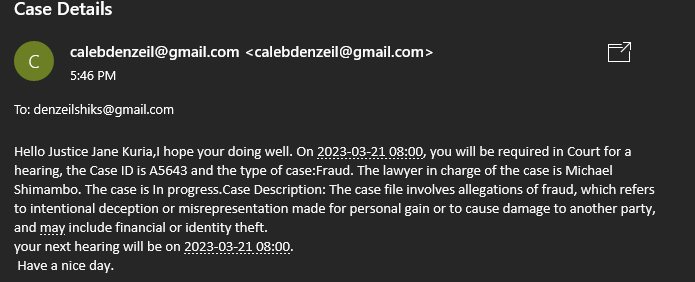
### 3.6.3 Case Entry Page

After being given access, the user will be taken to the main for entering cases. The user will be able to record cases. Each case file requires a case ID, the client’s first and second name, the lawyer/Attorney in charge of that case, and both the client and lawyer’s contacts. Some of the fields that are critically required are the Case type and Case status, it can be pending, approved, on hold or resolved. The clerk will be able to track a case using the Case status. The court appearances that the client has made and billing information. The name of the Judge in charge of the case has to be recorded including their email, contact and ID number. A hearing date has to be set for the case and also the next hearing. All these details have to be submitted to the database and an email will be sent to the Judge containing some of the case details. This will save the judge time by sending them the required details of a case file instead of going through a whole document with a lot of pages.

**Figure 21: Case Entry Page**



**Figure 22: Case Details email**



After submitting the case details, an email that looks like the one in Figure 21 will be sent to the judge. The case details can be updated for example the client has shown up twice in the courtroom, the number of appearances and the next hearing has to be updated. If the user is an admin, he can be able to delete the case details.

### 3.6.4 Code generation

The code below is a function that is used to submit information to the database.

  def submit(self):

            if self.caseid\_string.get()== " " or self.appearances\_string == " ":

                messagebox.showerror('Error','All fields are required')

            else:

*#establish a connection*

                    db=mysql.connector.connect(

                    host="localhost",

                    user="root",

                    password="1caleb2denzeil",

                    database="Judiciary"

                     )

*#create a cursor*

                    cursor=db.cursor()

                    try:

                        casestatus=self.my\_object.casestatus\_string.get()

                        judgename=self.my\_object.judgename\_string.get()

                        judge\_tables="INSERT INTO judge\_information(Judge\_ID,Judge\_name,Judge\_email,Judge\_contact) VALUES (%s,%s,%s,%s)"

                        judge\_values=(self.judge\_stringid.get(),judgename,self.judgeemail\_string.get(),self.judgecontact\_string.get())

                        cursor.execute(judge\_tables,judge\_values)

                        sql\_tables="INSERT INTO client\_information(Case\_ID,Client\_name,Client\_Contact,Case\_type,Case\_status, Appearances, Billing\_ksh, Judge\_ID, Hearing\_date, Next\_hearing, Laywer\_name, Laywer\_contact,Date\_filed) VALUES(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"

                        client\_values=(self.caseid\_string.get(),self.clientname\_string.get(),self.clientcontact\_string.get(),self.casetype\_string.get(), casestatus ,self.appearances\_string.get(),self.billing\_string.get()

                                      ,self.judge\_stringid.get(),self.hearingdate\_string.get(),self.nexthearing\_string.get() ,

                                      self.lawyername\_string.get(), self.lawyercontact\_string.get(),self.datefiled\_string.get())

                        cursor.execute(sql\_tables,client\_values)

*#Insert into judge information*

                        db.commit()

                        messagebox.showinfo("Success","Case Details inserted successfully")

                    except mysql.connector.Error as error:

                        messagebox.showerror("Error",str(error))

                    finally:

                        cursor.close()

                        self.my\_object.fetch\_data()

                        db.close()

*#establish a connection*

## 3.7 Testing

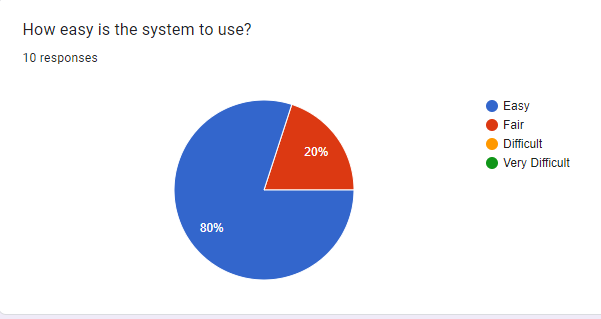
Usability, GUI and Performance testing were done to ensure the system is functional and meets the expected requirements. The system must undergo testing to ensure that the app is working in the manner in which it was intended (Pearson, 2015).

This involved testing components of the system to ensure functionality and a logical flow of data from one unit to the next. The system was tested several times to ensure that there were no errors during the execution of the program and that information submitted from one phase to the next was successful.

Functionalities have been implemented to handle errors during execution to be specific to python exception handlers. A lot of time was spent during coding cause of the many functionalities that had to be implemented. Usability testing was done to determine the ease of the system and this was the feedback given back.

### 3.7.1 Usability Testing

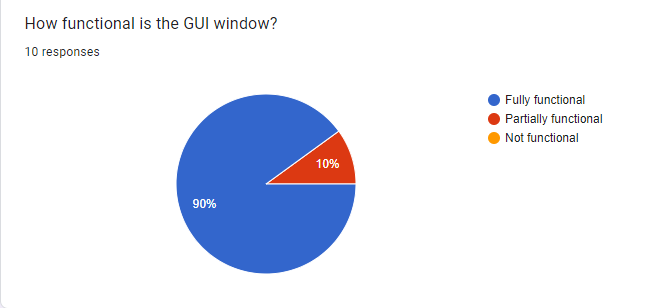
Usability testing is the practice of testing how easy a design is to use with a group of representative users. It usually involves observing users as they attempt to complete tasks and can be done for different types of designs (“Usability Testing”, .n.d.). 10 people used the system and after that, I asked them to fill out a questionnaire. This was their response according to how easy the system is to use. 80% of the users said that the system is easy to use while 20% fair.



**Figure 23: Usability results**

### 3.7.2 Graphical User Interface Testing

GUI testing refers to testing the functions of an application that are visible to a user(“GUI testing guide”, n.d.). GUI Testing is a process of testing the application’s graphical user interface to ensure proper functionality as per the specifications. It involves checking the application components like buttons, icons, checkboxes, colour, menu, windows etc (“GUI Testing Tutorial: A Complete User Interface (UI) Testing Guide”, 2023).The graphical user interface of the system was tested and most of the users were conversant with how the system works. 90% of the users agreed that the system is fully functional while 10% partially functional.

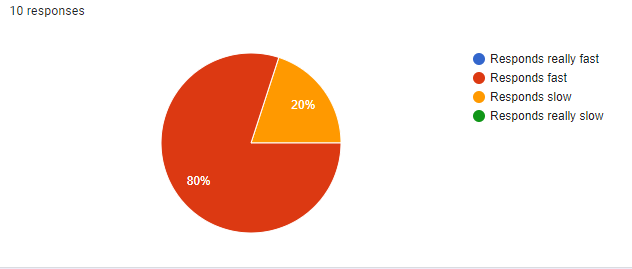


**Figure 24: GUI results**

### 3.7.2 Performance Testing

Performance testing is a form of software testing that focuses on how a system running the system performs under a particular load (“The Ultimate Guide to Performance Testing”,2023). The users tested the performance and responsiveness of the system and this was their response. 80% of the users agreed that the system responds fast while 20%, responded slowly.

**Figure 25: Performance results**



## 3.8 Summary

Technology is ever-changing and after developing this system, the time spent on updating and tracking cases will be reduced. The judge will be able to know the case details of a certain file through an email without him going through the actual document. Cases will be stored in a secure database and updating them will be quite easy. The system can work both online and offline.

## 3.9 Conclusion

In conclusion, the objectives of this project were successfully implemented. The system now allows for the tracking of cases based on their unique case ID and status. Proper schedules can be generated for court cases, ensuring that all parties involved are informed and prepared. The creation of a database has enabled the storage, management, and backup of case files and details, reducing the risk of data loss. Additionally, the system generates an overview of a case to the judge via email, providing a more efficient and streamlined process for case management. Overall, the successful implementation of these objectives has resulted in an improved Judicial Case Monitoring System that is more efficient and effective in managing court cases.

## 4.0 Limitation

Creating the case monitoring software proved to be a challenging task, with several obstacles encountered during the data collection process. One of the main challenges faced was the difficulty in reaching out to some of the clerks, who were often preoccupied with their busy schedules, making it hard to establish contact with them. Another way to be able to collect data from them and luckily was viable. During the coding phase, the developer encountered yet another challenge in implementing some of the system functionalities. For instance, hiding the delete button for non-admin personnel proved to be a daunting task. Despite these challenges, the developer managed to overcome them and create a functional and reliable case-monitoring software

## 4.1 Recommendation

The researcher was able to solve most of the problems encountered during the creation of the system but acknowledges that there may be more challenges in the future. The researcher believes that the current system is a solution to the identified problems, but acknowledges that someone else may come up with a better solution to address future challenges. The researcher recommends that any future improvements should be made to enhance the system's user experience.

# References

Juma,G.(2020). Embracing electronic court case management systems. Retrieved from https://www. Dlapiperafrica .com/senegal/ insights/ view.html?item=a5e63fe2-ceb3-11eb-99ca-393367b20614

Mayo, A. A.(2018). Role of Technology in Modernizing Judicial System.docx .Retrieved from [Role of Technology in Modernizing Judicial System.docx | Asghar Mayo - Academia.edu](https://www.academia.edu/38321106/Role_of_Technology_in_Modernizing_Judicial_System_docx)

Nyambane, E. O.(n.d.). Court management system project proposal. Retrieved from [court management system project proposal | kcau bscit - Academia.edu](https://www.academia.edu/8747468/court_management_system_project_proposal?email_work_card=view-paper)

Strikaitė-Latušinskaja, G.(2023,January 20). The Rule of Law and Technology in the Public Sector. Retrieved from [https://www.researchgate.net/publication/ 367285135\_THE\_RULE\_OF\_LAW\_AND\_TECHNOLOGY\_IN\_THE\_PUBLIC\_SECTOR](https://www.researchgate.net/publication/%20367285135_THE_RULE_OF_LAW_AND_TECHNOLOGY_IN_THE_PUBLIC_SECTOR)

Alcántara Francia, O. A., Nunez-del-Prado, M., & Alatrista-Salas, H. (2022,October 11). Survey of Text Mining Techniques Applied to Judicial Decisions Prediction. Retrieved from [https://www.researchgate.net/publication/ 364506976\_Survey\_of\_Text\_Mining\_Techniques\_Applied\_to\_Judicial\_Decisions\_Prediction](https://www.researchgate.net/publication/%20364506976_Survey_of_Text_Mining_Techniques_Applied_to_Judicial_Decisions_Prediction)

Zalnieriute, M., & Bell, F. (2020,January). Technology and the Judicial Role. Retrieved from [https://www.researchgate.net/publication/337582545 \_Technology\_and\_the\_Judicial\_Role](https://www.researchgate.net/publication/337582545%20_Technology_and_the_Judicial_Role)

The Ultimate Guide to Performance Testing and Software Testing: Testing Types, Performance Testing Steps, Best Practices, and More. (2023,March,10). Retrieved from <https://stackify.com/ultimate-guide-performance-testing-and-software-testing/>

GUI testing Guide .(.n.d.). Retrieved from [https://www.ranorex.com/ resources/testing-guides/gui-testing/](https://www.ranorex.com/%20resources/testing-guides/gui-testing/)

Faisandier, A., Roedler, G., & Adcock, R. (2022,October 31). System Architecture. Retrieved from <https://sebokwiki.org/wiki/System_Architecture>

What is SDLC Waterfall Model (2023,February 23). Retrieved from <https://www.softwaretestinghelp.com/what-is-sdlc-waterfall-model/>

Bhat, A. (2023). Questionnaires: The ultimate guide, advantages & examples. Retrieved from <https://www.questionpro.com/blog/what-is-a-questionnaire/>

Pedamkar, P. (.n.d.). What Is System Design? Retrieved from <https://www.educba.com/what-is-system-design/>

Richard, B. (2023,January 31). What is MySQL: MySQL Explained For Beginners. Retrieved from <https://www.hostinger.com/tutorials/what-is-mysql>

JupyterLab: A Next-Generation Notebook Interface(2023, March). Retrieved from <https://jupyter.org/>

Pearson, L. (2015, September 11) The Four Levels of Software Testing. Retrieved from <https://www.seguetech.com/the-four-levels-of-software-testing/>

Usability Testing (.n.d.). Retrieved from <https://www.interaction-design.org/literature/topics/usability-testing>

GUI Testing Tutorial: A Complete User Interface (UI) Testing Guide (2023, February 11). Retrieved from <https://www.softwaretestinghelp.com/gui-testing/>