

**ZETECH UNIVERSITY**

**JUDICIAL CASE MONITORING SOFTWARE**

**NAME: CALEB DENZEIL SHIKHUTULI**

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

**SUPERVISOR: JOSEPHINE MAGU**

**A project proposal submitted to the department of Information Technology and Engineering for the partial fulfilment of the award in Bachelor of Science in Information Technology at Zetech University**

**04/01/2023**

**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: ……………………………………..

**Table of Contents**

Declaration…………………………………………………………………… i

Table of Contents…………………………………………………………….. ii

List of figures………………………………………………………………...iii

Abstract…………………………………………………………………………iv

CHAPTER ONE………………………………………………………………..5

Introduction……………………………………………………………………. 5

* 1. Background Information…………………………………………………....5
  2. Problem statement…………………………………………………………..7
  3. Proposed Solution…………………………………………………………...7
  4. Objectives…………………………………………………………………...8
     1. Main Objective…………………………………………………………...8
     2. Specific Objectives……………………………………………………….8
  5. Justification………………………………………………………….............8
  6. Scope………………………………………………………………………..9
     1. Limitations of the proposed system………………………………….…..9

CHAPTER TWO………………………………………………………………10

* 1. Introduction………………………………………………………………...11
  2. Theoretical review…………………………………………………………11
  3. Critique of the existing literature ………………………………………….12

2.0 Research gaps……………………………………………………………..12

2.1 Summary………………………………………………………………….12

CHAPTER THREE……………………………………….........................13

2.2 System Analysis and design…………………………………………….13

**LIST OF FIGURES**

**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. **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. **Problem Statement**

Records and information management have been gaining popularity and recognition from the public sector around the world as governments throughout the world 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. **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 to administer justice fairly as per the regulations of law.

* 1. **Objectives**

**1.4.1 Main Objective**

The main objective is to develop and implement an efficient case management system in Kenyan Courts that automates case recording.

**1.4.2 Specific Objectives**

1. Track cases.

2. To generate proper schedules for court cases.

3. To create a database to store, manage and back up case files and details.

4. To generate a system that will provide an overview of a case to the judge.

* 1. **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. **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.

* + 1. **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. **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. **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 (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).

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 (Gilbert, 2020).

* 1. **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.

This model is used in the research to ensure that the project is completed systematically and organized.

**2.4 Feasibility Study**

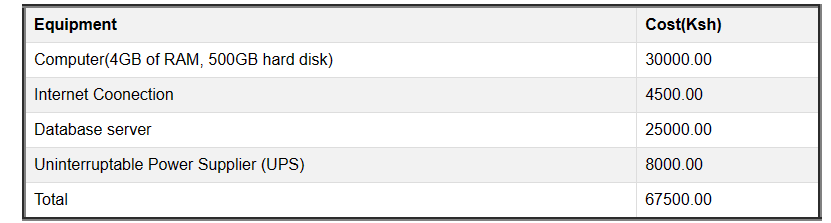
A feasibility study was done on the following areas to assess the viability of the project:

**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.



**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.

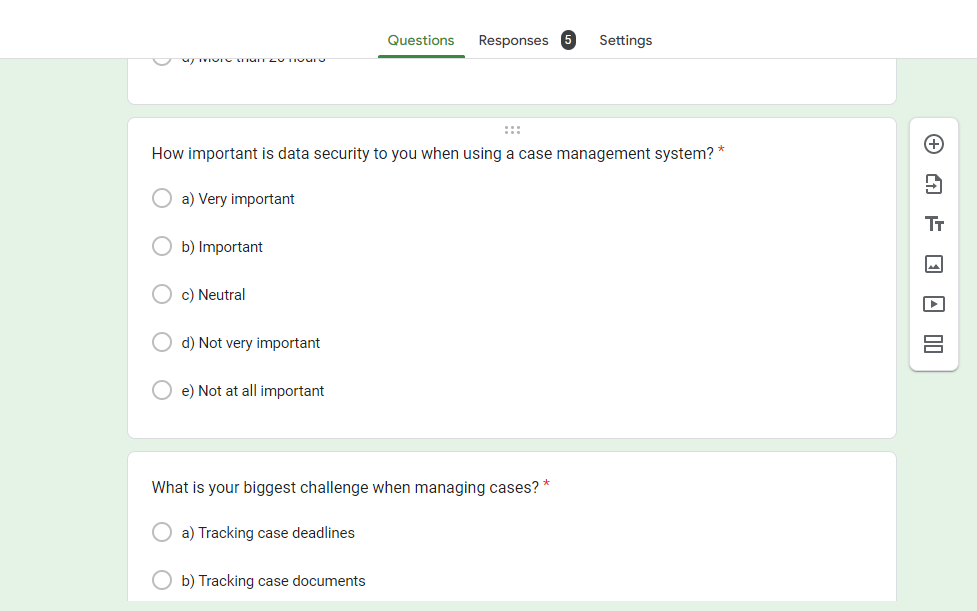
**Figure 1: Waterfall model**



**2.5 Data Collection**

A questionnaire was formulated and administered to a group of 5 clerks 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?

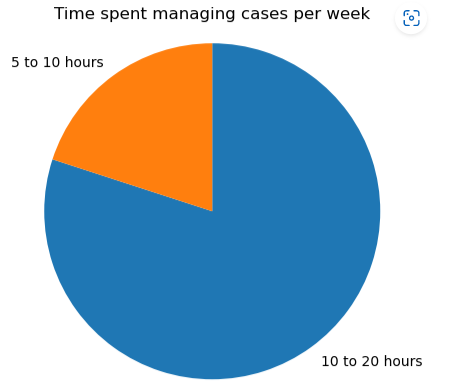
**Figure 2: Questionnaire form**



**2.6 Data and System Analysis.**

I downloaded the data from google docs as a CSV file and used jupyter notebooks for the data analysis. 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.

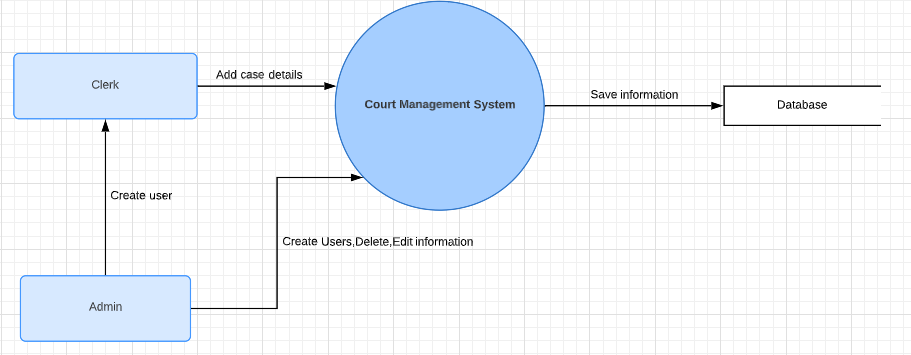
**DESIGN**

**2.9 System design**

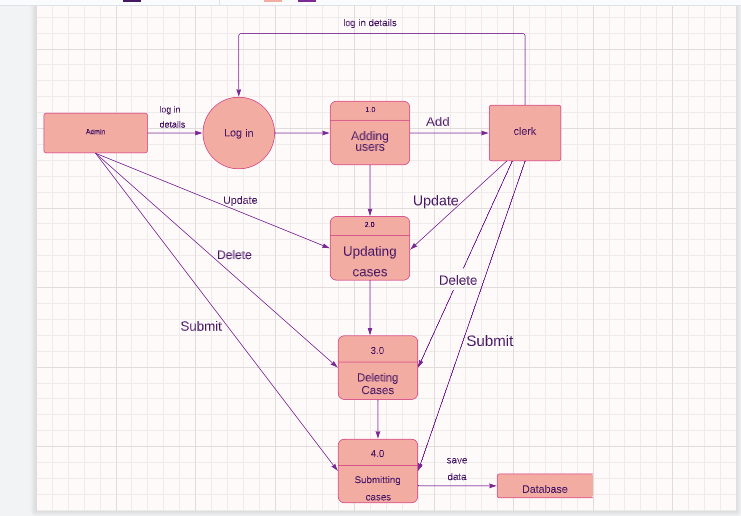
**2.9.1 Logical design**

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**

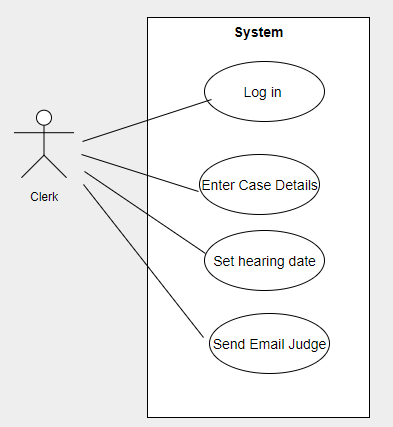
**Figure 6: DFD level 1**

****

**2.9.2 Use case**

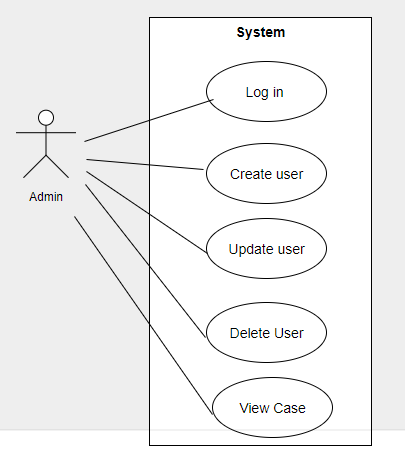
The court clerk in charge of a specific working desk will serve as the user of the system. They will enter the client’s information such as their name, and contacts, as well as the date and time of the recording. Additionally, they will enter the details of the case. Finally, they will set a hearing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

**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

**Figure 8: Admin Use Case**



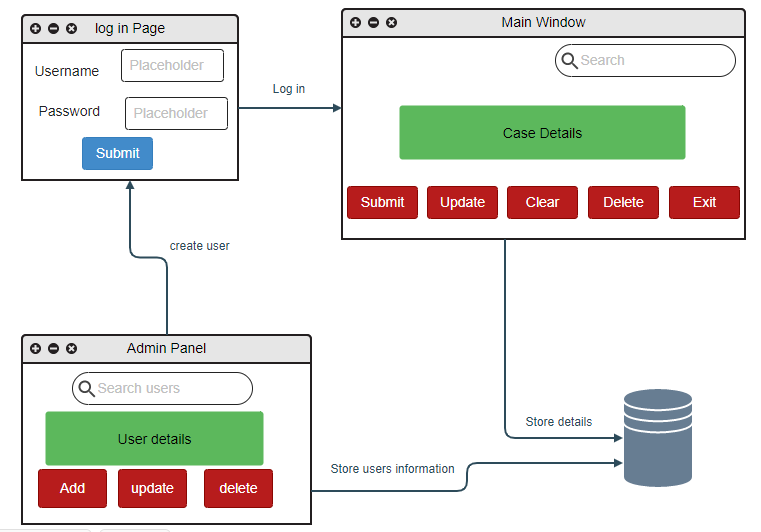
**2.9.2 Physical design**

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**



**3.0 System Architecture**

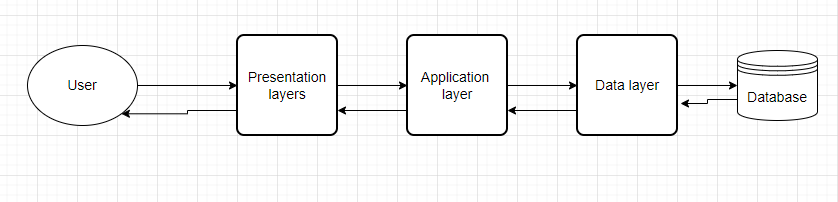
The presentation layer consists of the widgets and controls that the user interacts with such as buttons, this is the main interface with that the user will be interacting. For example, if a user clicks on a button to search for a case, the presentation layer would call a method in the application layer to execute the search and retrieve the results to display in the UI.

The 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**

**3.0.1 Proposed System Requirements**

**Software requirements**

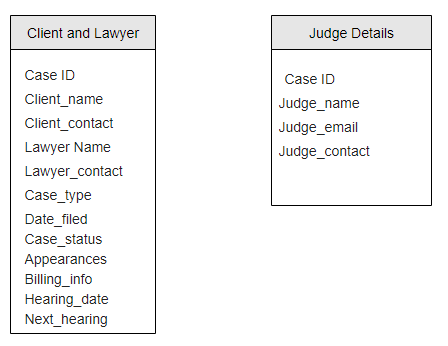
* Operating System: typically a modern version of Windows, Linux or macOS.
* Database: MySQL.
* Python 3.8 or higher to be installed
* Tkinter Library to be installed.

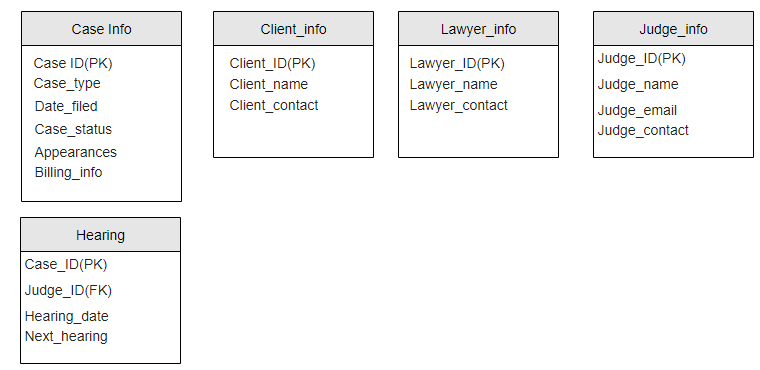
**Hardware requirements**

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

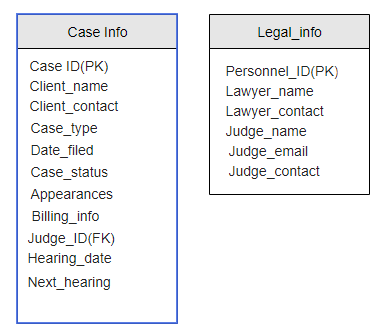
**3.1 Normalized Database**

**Figure 11: Unnormalized form**

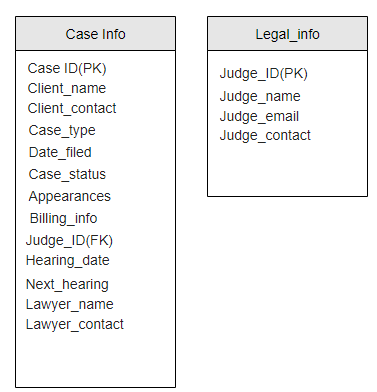


**Figure 12: First Normal Form **

**Figure 13: Second Normal Form**



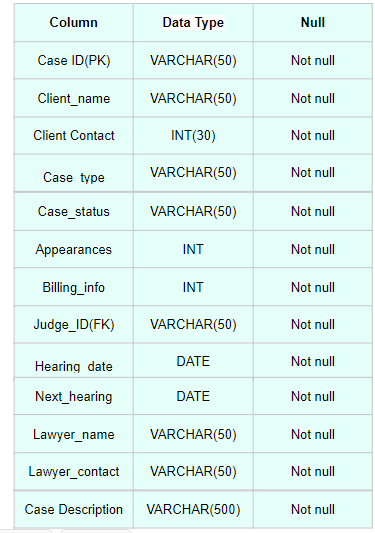
**Figure 14: Third Normal Form**



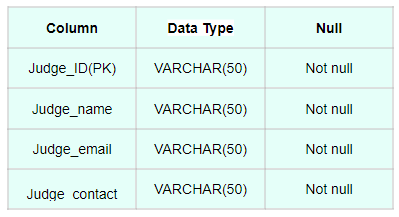
**3.2 Database Design**

The database will have two tables: Client and legal information

**Figure 15: Client information Table**



**Figure 16: Legal information Table**

****

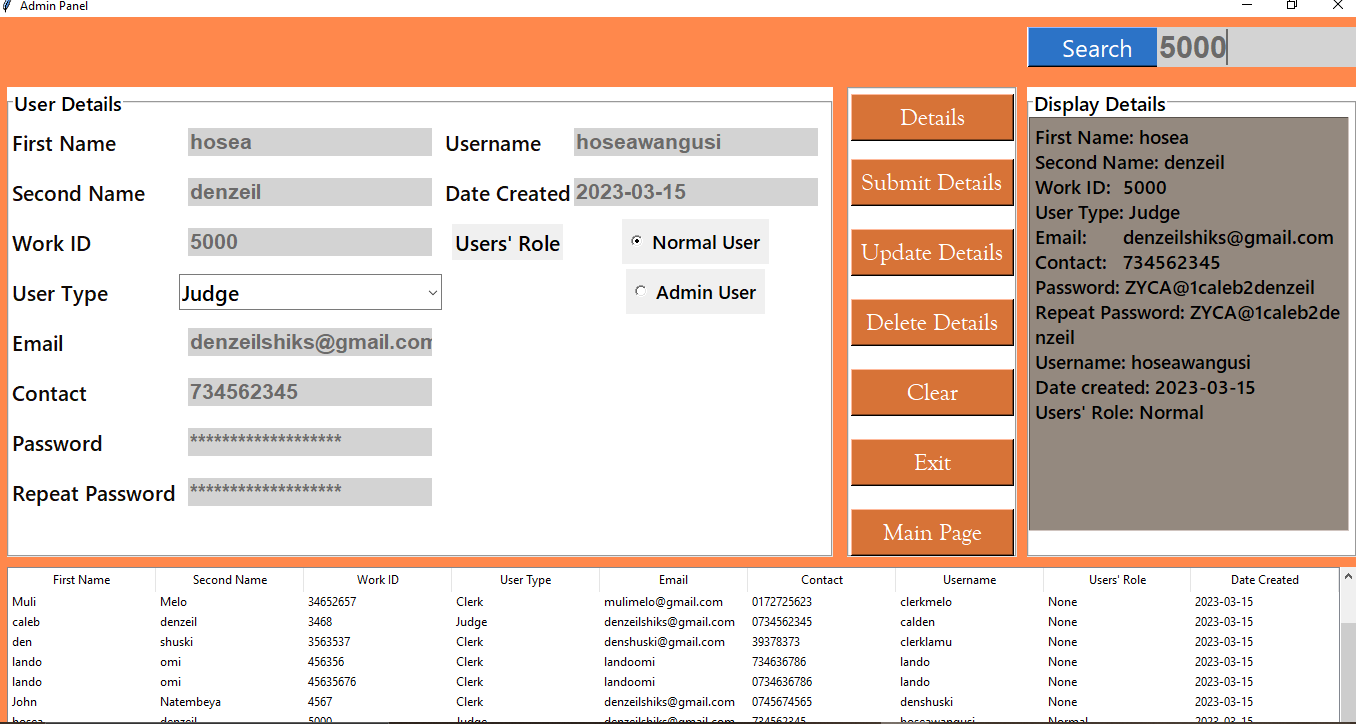
**CHAPTER 4 SYSTEM CODE GENERATION AND TESTING, CONCLUSIONS AND RECOMMENDATIONS**

**3.3 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.

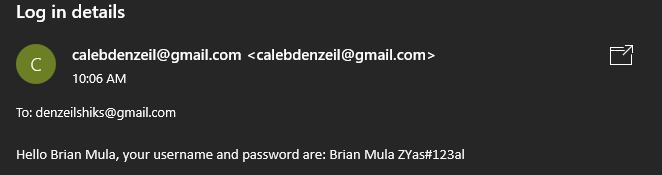
**3.3.1 Admin Page**

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 17: Admin Page**

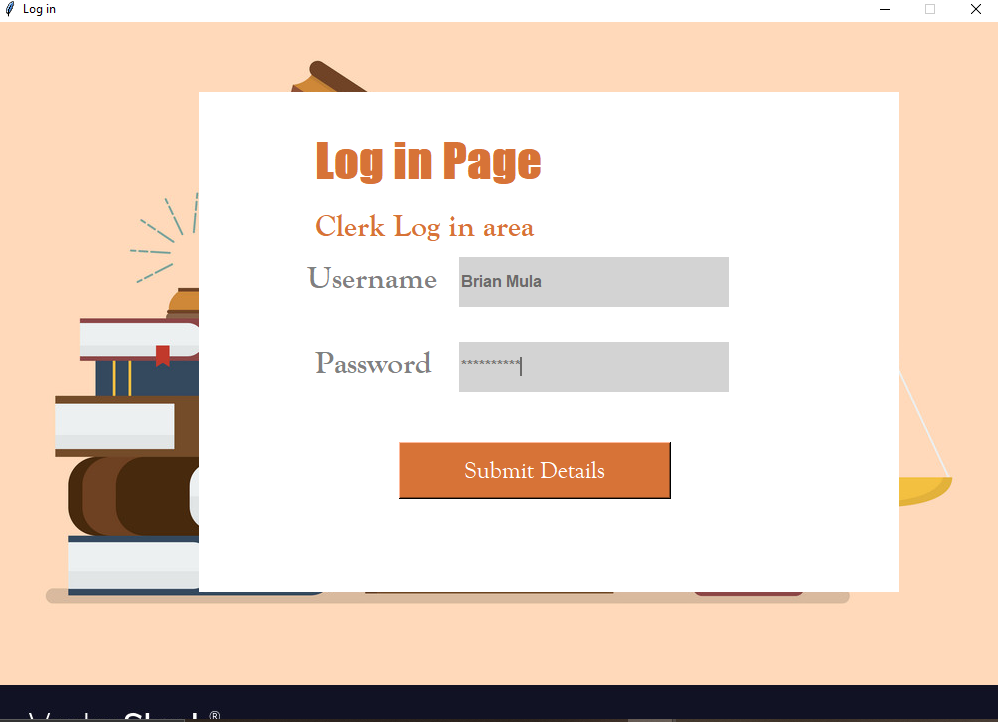
The login details will be sent via email using an API called SMTP, this API connects the system and Google. The user can be an admin or a normal user. After submitting the information to the database. The users’ information will be displayed at the bottom and right.



**Figure 18: Log-in details**

**3.3.2 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.

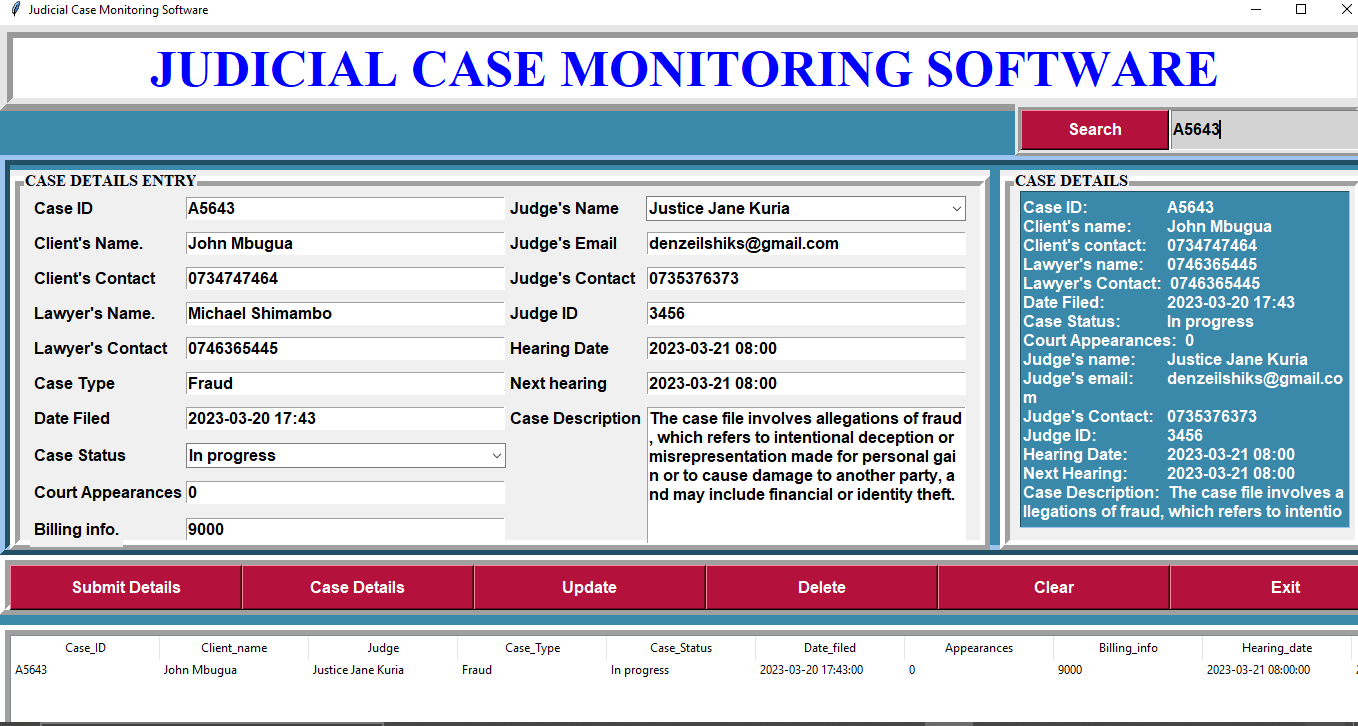


**Figure 19: Log-in Page**

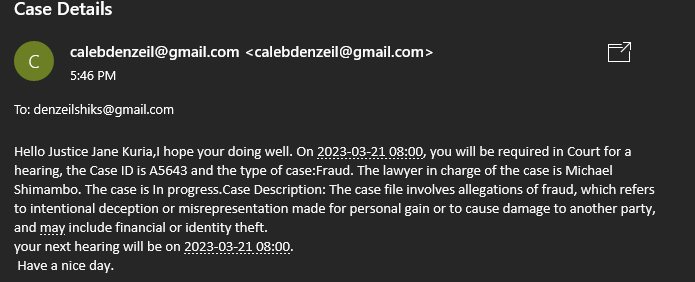
**3.3.3 Case Entry Page**

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 20: Case Entry Page**



**Figure 21: 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 updated for example the client has shown up twice in the court room, 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.

**3.3.4 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*

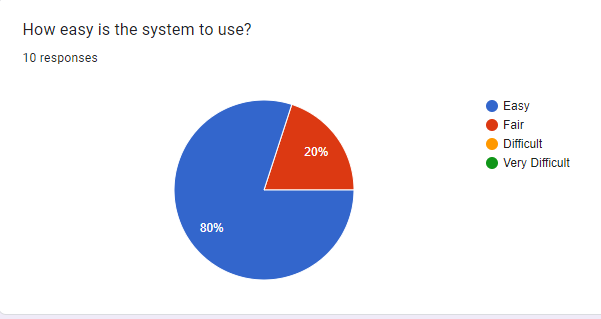
**3.4 Testing**

Usability, GUI and Performance testing were done to ensure the system is functional and meets the expected requirements.

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.

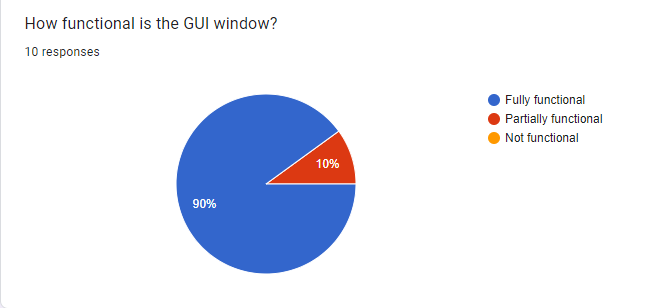
10 people used the system and after that, I asked them to fill out a questionnaire. This was their response according to how easy the system is to use. 80% of the users said that the system is easy to use while 20% fair.



**Figure 22: Usability Testing**

**Graphical User Interface Testing**

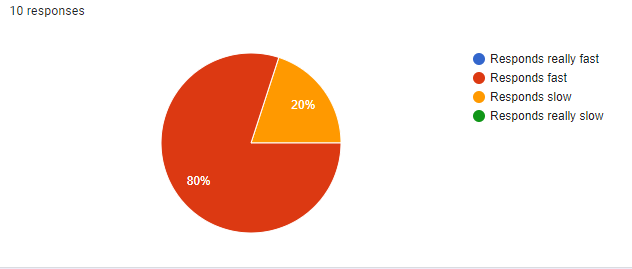
The graphical user interface of the system was tested and most of the users were conversant with how the system works. 90% of the users agreed that the system is fully functional while 10% partially functional.



**Figure 23: GUI Testing**

**Performance Testing**

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 to the 20%, it responded slowly.



**3.5 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.

**3.6 Conclusion**

I was able to solve almost all the problems by creating a system that is able to send a summary of the case file to the judge thus reducing the time to go through the whole document. Cases can be tracked easily. I was able to create a database to store, manage and back up case files and details. Hearing dates can be scheduled for the client and the Judge.

**3.7 Limitation**

Creating the case monitoring software was not easy and some of the challenges that I encountered were during the collection of data. It was difficult to get in touch with some of the clerks because most of them are extremely busy and they are not to find. I had to try another way to be able to collect data from them and luckily I was able to. Another challenge that I encountered was during the coding phase, implementing some of the functionalities was not easy for example, I was trying to hide the delete button for non-admin personnel.

**3.8 Recommendation**

I was able to solve most of the problems but I know more challenges would arise in the future. I was able to come up with this system and most probably someone else will come up with a better solution to the upcoming challenges. I recommend any improvement to be done on this system to provide a better experience for the users.

**References**