



ASHESI UNIVERSITY COLLEGE

LEGAL MANAGEMENT SOFTWARE

UNDERGRADUATE APPLIED PROJECT

B. Sc. Computer Science

Lydia Kemuma Kinyari

April 2016

Legal Management Software

APPLIED PROJECT

Applied Project submitted to the Department of Computer Science, Ashesi
University College in partial fulfilment of the requirements for the award of
Bachelor of Science degree in Computer Science

Lydia Kemuma Kinyari

April 2016

DECLARATION

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

.....

Candidate's Name:

.....

Date:

I hereby declare that preparation and presentation of this applied project were supervised in accordance with the guidelines on supervision of applied projects laid down by Ashesi University College.

Supervisor's Signature:

.....

Supervisor's Name:

.....

Date:

ACKNOWLEDGEMENT

The journey towards completion of this project would have been fierce, thanks to the special people who challenged and supported me along the way. First of all, I would like to express the deepest gratitude to my supervisor, Mr. Stephane Nwolley, for his expertise, valuable guidance and patience throughout the period.

I take this opportunity to thank all the faculty members of the Computer Science Department at Ashesi University for their help and encouragement.

I would also like to thank Abraham Odoi and Samuel Bentil of Nalo Solutions for their brilliant ideas especially at the brainstorming stage.

Finally I am highly indebted to Mr. Jones Nyachiro of Nyachiro Nyagaka & Co. Advocates for the inspiration to start this project.

ABSTRACT

As the world increasingly progresses towards a technology driven environment, the legal sector is not to be left behind. In the past years, clients got to know about lawyers through recommendation and at the same time lawyers had a challenge reaching out to their clients. Work was slow and tedious as research was conducted on dusty and old textbooks.

The basis of this project is to bridge the technology gap between the lawyers and advocates in Africa. Firms that deal with legal services will most likely handle huge quantum of inventory and have to make sure that these files are readily available at the point of request by the court. It has therefore become necessary for these firms to employ a method that will securely handle this large data and readily avail it to more than one person at a time even though they are in different locations. The goal of this project is to develop an application that will aid lawyers to access the files from wherever they are, and quickly generate reports.

ACRONYMS

PHP - Hypertext Preprocessor

HTML - Hypertext Markup Language

CSS – Cascading Style Sheet

UI –User Interface

AJAX – Asynchronous JavaScript and XML

API – Application Programming Interface

HTTP – Hypertext Transfer Protocol

ER – Entity Relationship

SQL – Structured Query language

JSON – Java Script Object Notation

TABLE OF CONTENTS

Chapter 1: Introduction	1
1.1 Background Research.....	1
1.2 Problem	2
1.3 Existing solution.....	4
1.4 Objectives	5
1.5 Overview of Report	5
Chapter 2: Requirements.....	7
2.1 Requirements Gathering.....	7
2.2 Users and Use Cases.....	8
2.3 Functional Requirements.....	9
2.4 Non-Functional Requirements	11
2.5 Software Quality Attributes.....	12
Chapter 3: Architecture and Design.....	13
3.1 Framework.....	14
3.2 Modules	15
3.3 Database architecture.....	18
3.4 Data Design	22
3.5 Class Diagrams	26
Chapter 4: Implementation	29

4.1	Description of Implementation.....	29
4.2	Major Tasks	29
4.3	Security.....	29
4.4	Software and Tools.....	30
4.5	Screenshots of Implementation	31
Chapter 5: Testing and Results		34
5.1	Development Testing	34
5.1.1	Unit Testing.....	34
5.1.2	Component Testing	35
5.1.3	System Testing.....	36
5.2	User Testing	36
Chapter 6: Conclusions and Recommendations		37
6.1	Conclusion.....	37
6.2	Challenges	37
6.3	Future work	38

LIST OF FIGURES

Figure 1.1 The state of Law firms' Registry office (a)	3
Figure 1.2 The state of Law firms' Registry office (b).....	3
Figure 1.3 The state of Law firms' Registry office (c)	4
Figure 2.1 use case diagram for the admin	8
Figure 2.2 use case diagram for the advocate	9
Figure 3.1 MVC diagram (tech, 2015).....	14
Figure 3.2 ER Diagram	18
Figure 3.3 Clients Table	19
Figure 3.4 Library Table	19
Figure 3.5 Library Table	20
Figure 3.6 Staff Table	21
Figure 3.7 Activity Diagram for the advocate	25
Figure 3.8 Activity diagram for the admin.....	25
Figure 3.9 Client Class.....	26
Figure 3.10 Staff Class	27
Figure 3.11 File Class	28
Figure 3.12 Library Class.....	28
Figure 4.1 Login Page	32
Figure 4.2 The Admin Home Page	32
Figure 4.3 View Files Page	33
Figure 4.4 Add File Page	33

Chapter 1: Introduction

1.1 Background Research

The different tasks endowed on lawyers have made legal practice an inevitable venture in today's Africa. Tasks such as providing advising services on legal rights and responsibilities, the representation of clients in civil and criminal cases and all other matters pertaining to business transactions, family advising and firm consulting.

Over the years, law firms have seen the need to incorporate technology in these practices. Lawyers from various parts of the world are adopting technologies for research, collaboration and communication (Ezor, 2014). The introduction of technology was a breakthrough for lawyers as it eliminated the out of office work limitation by helping lawyers handle large volume of cases whether in office or not.

Database technology for legal professionals has been in place since the 1980s. This was in the form of simple data field screens that were designed to track legal entity information. During this period, when attorneys/lawyers needed some information from the database, their request was channeled through the database gatekeepers for the different sections of the legal body such as the entity information, litigation, intellectual property, contracts and many more. However In the early 2000s the Web based software became a workable option for firms that were technologically aggressive (especially those in the western countries) (Weaver, 2016). Throughout this period the lawyers in the African context lagged behind due to the undeveloped technology in Africa.

1.2 Problem

According to Lexicata, the only cloud based client relationship management and client intake system designers exclusively for law firms, most law firms have adopted some technology into their processes already, but the acceptance of technology is still in its infancy (George, 2015). This is evident in the thousands of law firms found in Africa today. They have long been associated with piles of hard copy files with receipts flying all over the office. Neglecting the bad state of affairs, file bring ups and notifications have to be manually entered into a big office diary. The same office diary has to be duplicated so that more than one officer can access the information. This is a time and energy consuming process.

With the effort to find out the challenges that lawyers face, an independent research firm known as Activate, conducted research on South African Lawyers. This survey was conducted on behalf of LexisNexis. The survey covered over 160 law firms in that country. The report showed that, the smaller firms employed between one to eight *free earners* whose work was just registration, searching for files or compiling each of the lawyer's individual diary for synchronization purposes. This came as a surprise to the lawyers as they were unaware of the amount of money spent on employing relatively high numbers of subordinate staff (LexisNexis, 2015). This challenge is projected to most lawyers across the African continent.

Figure 1.1, 1.2 and 1.3 below show the state of the registry room of some of the law firms in Africa.



Figure 1.1 The state of Law firms' Registry office (a)



Figure 1.2 The state of Law firms' Registry office (b)



Figure 1.3 The state of Law firms' Registry office (c)

1.3 Existing solution

Currently, several applications have been developed to address the issue of low technology consumption in law firms but some of the lawyers still shy away from using technology in their practice even though these application have been tested and endorsed by the body of lawyers to improve the work efficiency and the profitability of the firm.

However, the attempts to incorporate technology in the legal sector have failed because of a few factors. Firstly, lawyers have been tagged pragmatists (Jacoby, 2015). This refers to them being able to see the positive effect of incorporating technology into their business as soon as possible. This is evident because word processing systems were quickly adapted to the law

firms as compared to management applications because these lawyers saw immediate value in changing their current style. This is also very practical because lawyers have cut their cost through the use of technology by eradicating a lot of their employees.

1.4 Objectives

In the attempt to appeal to lawyers to apply technology in their practice, the Legal Management Software aims to increase the usage of cloud computing decreasing the need for staff to sit in a physical office, this also allows the access of files from anywhere in the world. The software also aims to improve the system such that it provides availability of online legal resources, the ability to consult documents while using the application.

This Web based software allows the users to access the information wherever they are and this can be through different platforms like the phone, tablet and personal computers. The software also supports unlimited users as different law firms have a varied number of employees and clients. Finally, the system is suitable for the African environment in terms of the database values and the interfaces.

Such a system is more responsive to clients as it does not only ease the law firms' work but also that of the client. It also has a greater flexibility bearing in mind that it performs a range of tasks that would otherwise be done by a number of applications.

1.5 Overview of Report

This section highlights the structure of this report. In the first chapter, the elements of the introduction are discussed. This is broken down into the background research and the existing solutions to the issue. From these the motivation gained to develop the system are also highlighted.

In the second chapter, the requirements of the software are discussed. These were gathered from interaction with the workers/stakeholders of law firms. Non-functional requirements are also included to ensure the efficiency of the software.

The third chapter covers the architecture and design of the project. The design of the database is also included.

Chapter 4 explains the implementation of the software. The implementation model, the tool and the languages used are highlighted here.

Chapter 5 illustrates the testing process and the results. This involves the challenges the users faced while testing the app and the positive test results as well.

The last chapter concludes the article and points out the recommendations and the way forward for this software. The features that could not be implemented during this version are also listed as future work.

Chapter 2: Requirements

This section describes the software requirements for the “Legal Management Software”.

2.1 Requirements Gathering

In order to gather the requirements, various stakeholders of the legal system were contacted. Observation of the workflow at various law firms also contributed to the requirements gathering.

Table 2.1 gives detail information about the requirement gathering process

Table 2.1 Requirements Gathering Plan

Resource	Information	Method
Stakeholders (lawyers and advocates)	An idea of how law firms are run from lawyers and advocates. What are some of the functional requirement they would like to be included in the system?	Called to book an appointment and arranged for a meeting with a couple of them
Software Engineers	What are some of the standards with regards to legal software development?	Met supervisor/lecturers and discuss
Supervisor	How to acquire skills required for the development?	Organize a meeting with my supervisor
Library	What work has been done related to legal management software?	Read academic journals in the library. Find out the policies with regard to legal software development and usage
Legal clients	What are some of the challenges they face with the law firm services	Used google forms and interviews

2.2 Users and Use Cases

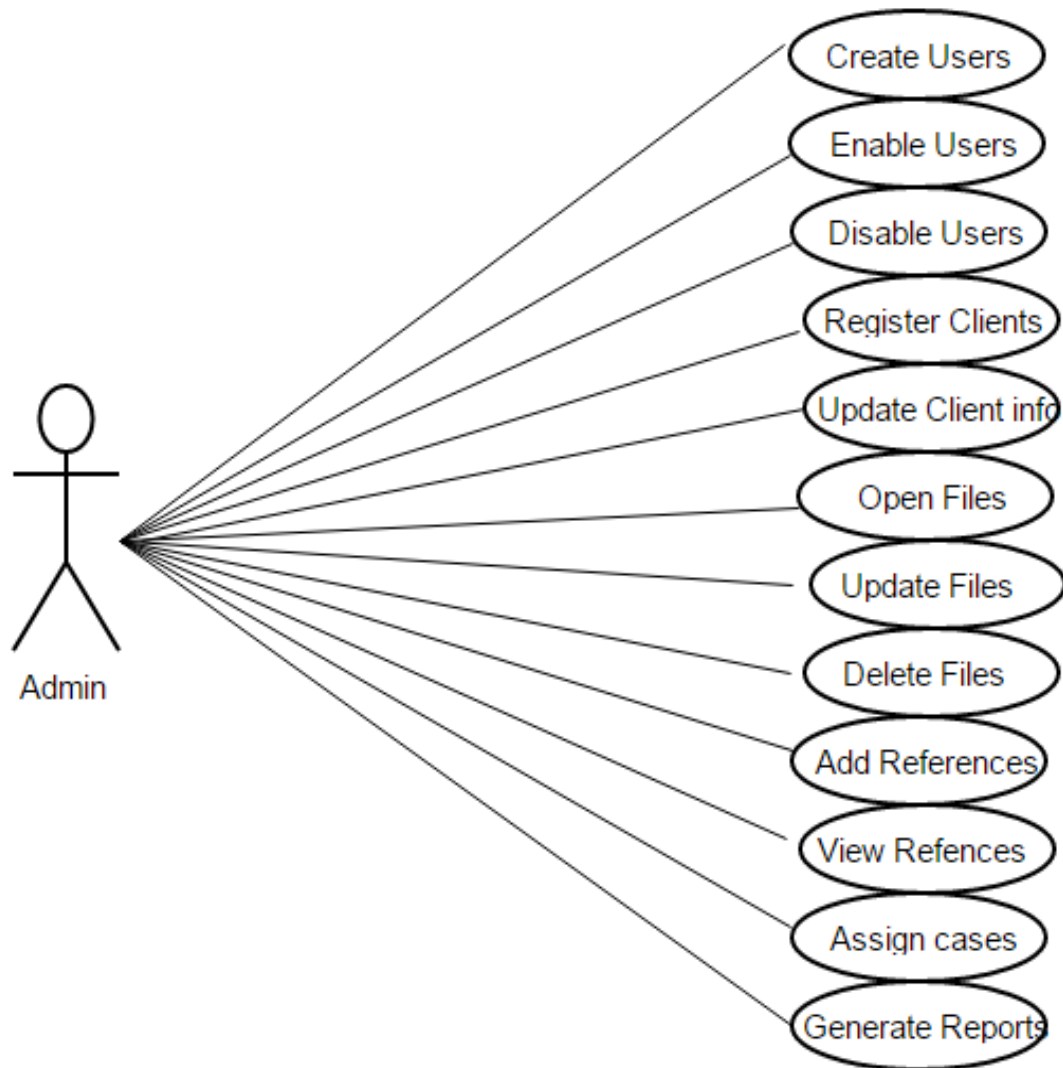


Figure 2.1 use case diagram for the admin

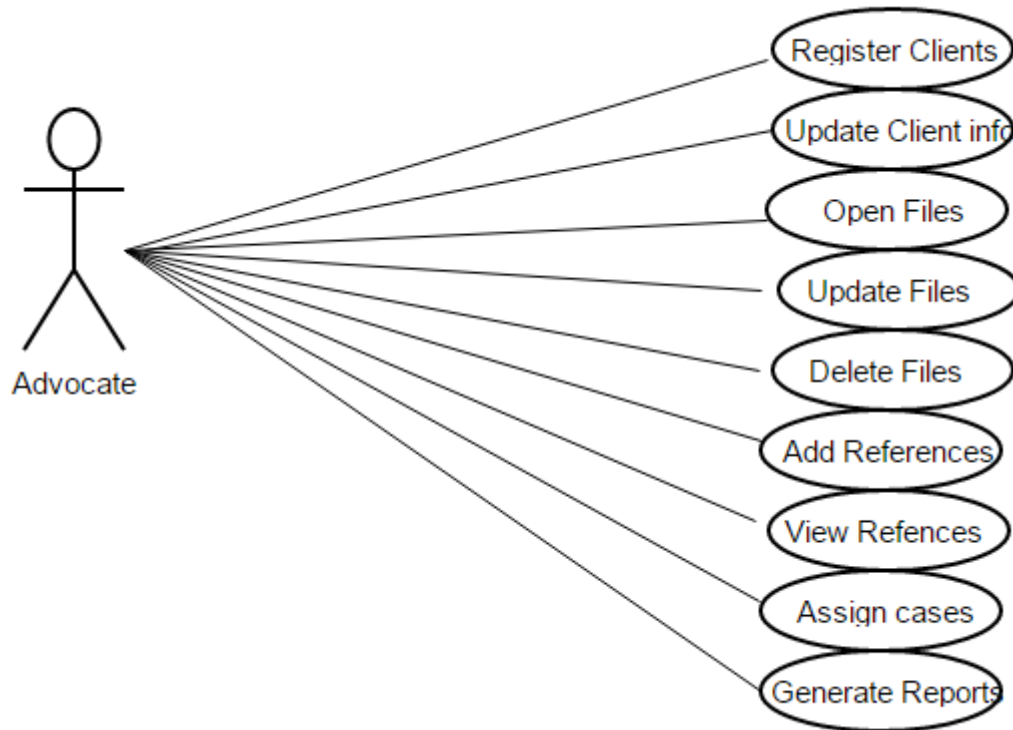


Figure 2.2 use case diagram for the advocate

Upon successful gathering of requirements, they were then grouped into functional and non-functional requirements.

2.3 Functional Requirements

The functional requirements were further subdivided into the specific functionalities of the software as below:

Manage files/cases:

- i. **Add a file**-The system allows the user to add new files to the database. This happens when a new case or client has been recruited.
- ii. **View files**-The system displays a list of files to the user and the user selects the number of files to be displayed at a time as desired.

- iii. **Search for a file**-The system also allows users to search for a given file. This can be achieved using any of the file attributes displayed to the user.
- iv. **Update a file**-Whenever a case has been heard in court or a new document has been discovered, the system allows the user to update the existing documents.
- v. **Delete a file**-In the event that the wrong data has been entered the user deletes the data but needs to provide confirmation that he really needs to delete the data

Manage clients:

- i. **Add a client**-The software allows the user to register new clients.
- ii. **View clients**-The user views the clients' information in a list and can alter it according to preference.
- iii. **Search for a client**-The system also allows users to search for a given client. This can be achieved using the client details displayed on the interface.
- iv. **Update a client information**-Whenever a client changes his/her details like contact and physical address the user updates this information accordingly.
- v. **Delete a client record**-a client can be deleted from the system but in the event of deleting confirmation is done to ensure the deletion of the record.

Staff/Account Management:

- i. **Create Staff Account**-The system allows the user (manager) to create a new user. This happens when a new staff has been recruited.
- ii. **View Staff Account**-The system also allows the manager to view the accounts of his employees and at the same time the employees can view the details of their individual accounts.

- iii. **Search staff**-the manager can search for a specific staff using their token or name.
- iv. **Update an account**-In the event that some details of a given staff have changed the manager updates these details accordingly. Again, prompts are made to ensure that the updated details are correct.
- v. **Disable user**-Whenever a staff leaves the firm the manager disables and re-enables the user upon return.

Manage Resource Materials:

- i. **Add a resource**-The users of the system are allowed to add new resource materials to the database.
- ii. **View resources**-The system displays the available resource material to the user. The user selects one of the materials to view.
- iii. **Delete a resource**-In the event that the wrong file has been added as a resource material the user deletes the file but needs to provide confirmation before the action is taken the data.

2.4 Non-Functional Requirements

Product Requirements

- i. **Performance Requirements** - In order for the system to work efficiently, the device needs to be connected to the internet. This is to enable the user to access the updated version of the details in the server.
- ii. **Security Requirements** – The passwords of the users are securely stored in an encrypted format. The system automatically logs the user out in the case that there is a period of inactivity. The clients' information needs to be confidential to all the staff.

- iii. **Software Quality Attributes** - The software has been tested to ensure all features are working as stated in the requirement document.

Organizational Requirements

- i. Only users who work with the organization and directly deal with the cases are allowed to use the software.
- ii. Multiple authenticated users can access the software at the same time.

External Requirements

- i. Ethical - Clients' information is confidential to those who require access to it and this information may not be shared with any other firm or person.

2.5 Software Quality Attributes

- i. **Correctness** - the legal management software behaves according to the stated functional requirements.
- ii. **Reliability** - the user is able to rely on the software to efficiently manage their files and clients information and prepare the required reports.
- iii. **Robustness** - when the user puts a wrong input, the software detects and alerts the user of the error.
- iv. **Maintainability and capability to evolve** – The legal management software has been developed such that there is room for modification and improvement so that other versions can be made.
- v. **Understandability** – The source code has been well documented so that other people can work on it without any difficulty.

Chapter 3: Architecture and Design

The success of any application or system depends on the architecture pattern used. The architecture of this application encompasses a set of significant decisions about the organization of the software system. These patterns not only guided the designing and developing of the application but also determined how various component of the system interact with each other. Security is one of the important requirements considered in this software project especially in the legal world due to the confidentiality of the information handled, thus the intention to combine the MVC and the repository. The repository approach is vital for the project as the software needed to have a centralized database which would be updated by various users of the system. The MVC architecture is also important because of the necessity to separate the three components i.e. the model, view and the controller for easier code management. Using the MVC would make it easier for the developer to change either the view or the controller without affecting the other. Since each of the components was clearly defined and articulated, testing the code was a lot easier.

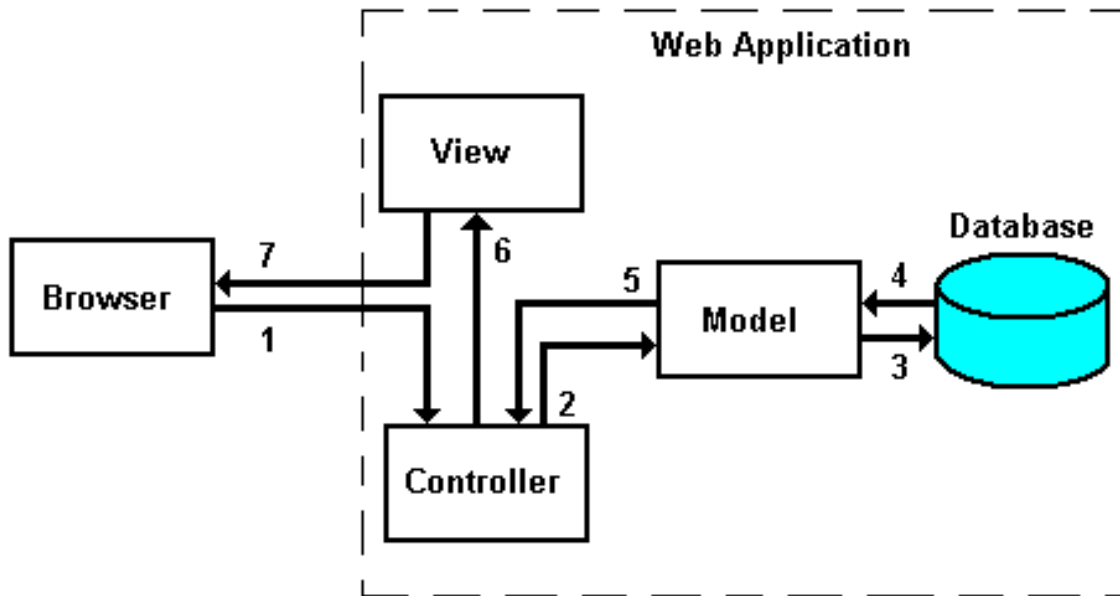


Figure 3.1 MVC diagram (tech, 2015)

Figure 3.1 illustrates the MVC architecture and the flow of information in applications built according to this architecture. In step 1, the user clicks on a button or send a form that submits information to the URL. In step 2, the data is passed to the controller and requests are sent to the model. In step 3 and 4 the requests are processed and data is either read or written to the database. The result is then posted to the controller in step 5 which further passes the data to the view in step 6. The view formats and sends the output to the browser in step 7 (tech, 2015).

3.1 Framework

In developing this software, Bootstrap template was used as the front-end framework. Bootstrap was easily accessible and it provided a good framework that was used to implement the project.

Below is a list of reasons Bootstrap was used in the implementation of the project

- 1) **Speed of development-** Rather than coding from scratch, the utilization of ready-made blocks by Bootstrap made the development process fast and easy.
- 2) **Responsiveness:** In the research conducted on the users of this software, there was a realization that many of them have smartphones that they basically use for everything. There was need to create a responsive website that would suit the needs of these users and Bootstrap offers everything we need to implement this website.
- 3) **Ease of customization:** In using Bootstrap, the developer had the ability to pick and choose the features that are needed and ignore the rest. This is easily accomplished using the Bootstrap customizing page.

3.2 Modules

The software entails different modules as indicated below:

1. **File Management-** This module allows the user to manage the files which contain the documents related to cases. New files can then be added to the repository whenever a new case comes up. There will be a prompt to enter the file number, the date, the case number and the parties involved and upload the scanned documents. The user is also able to update the documents related to a specific file once more documents are added, replaced or removed. When a wrong file or document has been added to the system they can be removed.
2. **Client Management-** This module allows the user to add a new client whenever a new client walks into the law firm. The user is required to know the client's name, the client's contact information and their location. The client's details such as the contact

information, location and other attributes are updated when and if necessary and finally the details can be deleted as desired.

- 3. Document Management-** Important legal documents can be managed by this module. Such documents include commonly referenced documents like the human rights documents and others that the lawyers deem important. The user uploads these documents by specifying their names and other required fields.
- 4. Case Management-**Case management involves keeping track of active cases. In this module the system matches the files to the clients and the user is able to open new cases by filling in the case number, case type, and the clients involved. In addition to this the progress of the case is monitored and in case of any hearing in court the files are updated. Once the case has been resolved the user should be able to mark the case as resolved/closed. The cases should always be updated with the last action.
- 5. Calendar Management-** This synchronizes all the events in the law firm such as appointments, upcoming hearings and other meetings with the calendar preferably google calendar. The user of the system is able to know the important activities of the day by looking at the calendar which is updated from the database.
- 6. Email Management-** This module deals with sending notifications and reminders. Reminders are sent once a client has made a payment and also to remind the client to make a payment when he has delayed.
- 7. Financial management-** Whenever a client makes any payment it is reflected in the system so that an electronic receipt can be sent to both the client and the lawyer.

8. Human Resource Management- The manager is able to create users. The secretaries will have the same functionality as the manager except the ability to create and view other users once they have been added.

9. Report Generation- This module generates various reports from the information in the system. For example the active cases, file bring ups and closed files.

These different modules and their respective features as defined were compiled to form the structure of the project.

3.3 Database architecture

The diagram below represents the ER diagram for the software.

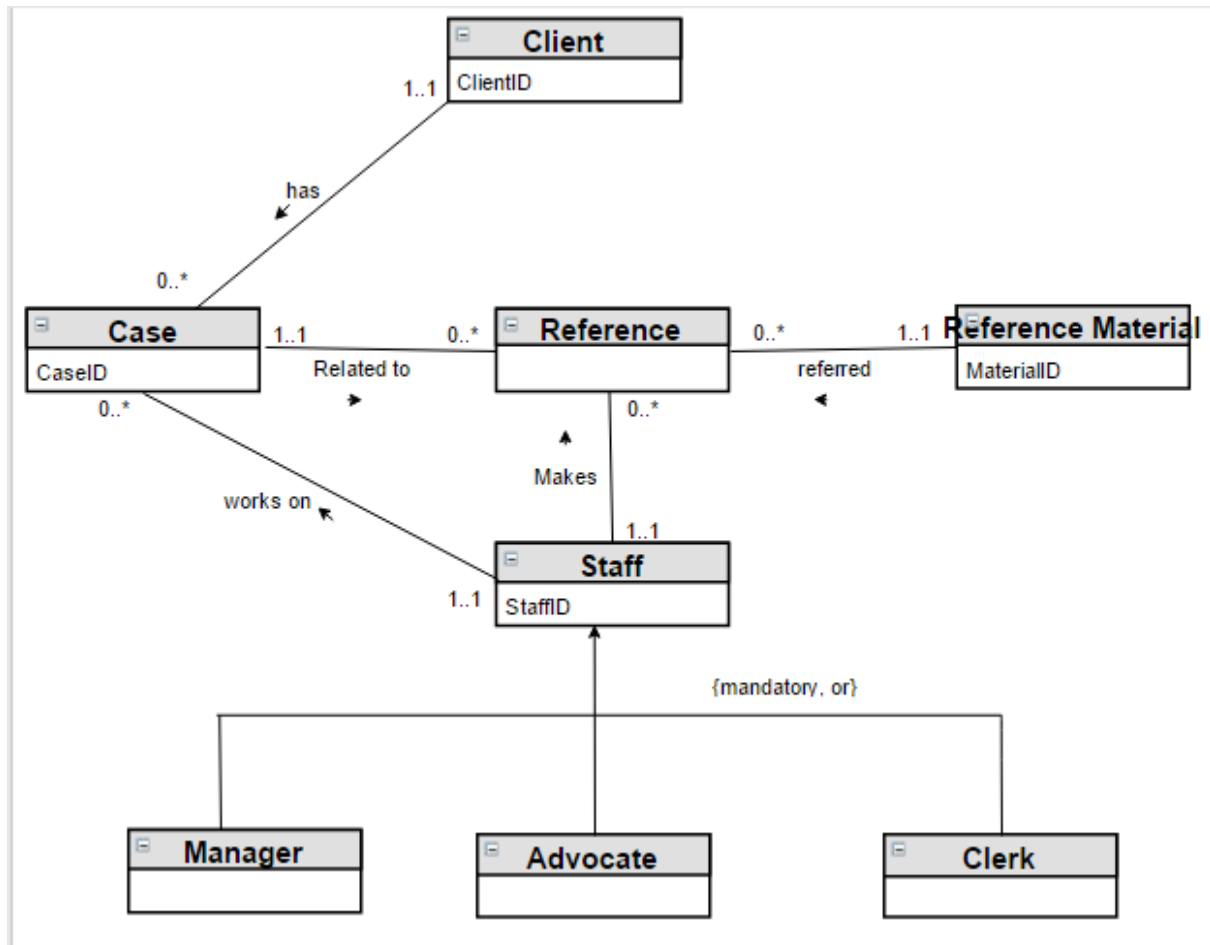


Figure 3.2 ER Diagram

In the construction of the Legal Management Software the following entities were needed:

- Clients-This table represents the entity that holds clients information.

v legal.clients	
🔑	id : int(11)
📄	name : varchar(50)
📄	address : varchar(50)
📄	code : varchar(20)
📄	town : varchar(20)
📄	telephone : varchar(20)
📄	email : varchar(50)
📄	website : varchar(100)

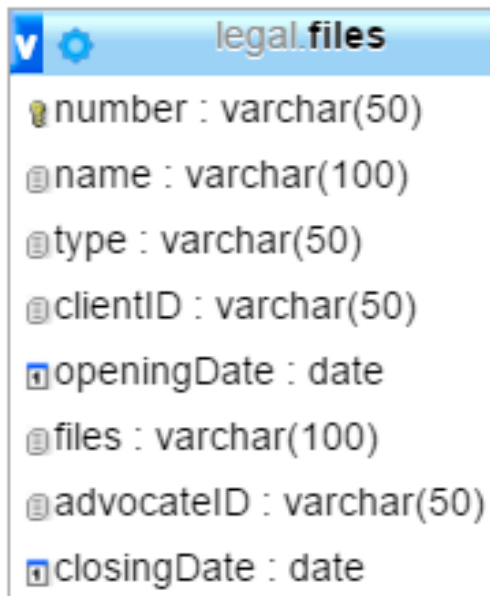
Figure 3.3 Clients Table

- Library-This table represents the entity that holds the reference materials.

v legal.library	
🔑	number : varchar(50)
📄	name : varchar(50)
📄	type : varchar(50)
📅	uploadDate : date
📄	files : varchar(100)
📄	advocateID : varchar(50)

Figure 3.4 Library Table

- Files-This table represents the entity that holds the cases/files.



The image shows a screenshot of a database table definition window. The title bar at the top is blue and contains a 'v' icon, a gear icon, and the text 'legal.files'. Below the title bar, the table's columns and data types are listed. Each column name is preceded by a small icon: a key icon for the primary key 'number', and a document icon for the other columns. The columns are: 'number : varchar(50)', 'name : varchar(100)', 'type : varchar(50)', 'clientID : varchar(50)', 'openingDate : date', 'files : varchar(100)', 'advocateID : varchar(50)', and 'closingDate : date'.

legal.files	
number	: varchar(50)
name	: varchar(100)
type	: varchar(50)
clientID	: varchar(50)
openingDate	: date
files	: varchar(100)
advocateID	: varchar(50)
closingDate	: date

Figure 3.5 Library Table

- Staff- this table represents the entity that holds the staff information.

legal.staff	
☰	surname : varchar(30)
☰	othernames : varchar(50)
🔑	Id : varchar(50)
☰	type : varchar(30)
☰	gender : varchar(10)
📅	dateOfBirth : date
📅	dateOfEmployment : date
☰	status : varchar(20)
☰	address : varchar(20)
☰	code : varchar(20)
☰	town : varchar(20)
☰	officeNo : varchar(30)
☰	mobileNo : varchar(30)
☰	email : varchar(30)
☰	kin : varchar(50)
☰	relation : varchar(50)
☰	telephone : varchar(30)
☰	nssf : varchar(30)
☰	nhif : varchar(30)
☰	kra : varchar(30)
☰	bank : varchar(30)
☰	branch : varchar(30)
☰	account : varchar(30)
☰	secretCode : varchar(100)
☰	pword : varchar(50)

Figure 3.6 Staff Table

3.4 Data Design

Table 3.1 below shows the data used to implement the software, their data type, whether null data is allowed and the description of the data.

Table 3.1 Data Dictionary

Table Name	Field Name	Data Type	Allow Null	Description
Client table	Id	int	No	Stores the id of the client as an auto increment value
	Name	varchar	No	Stores the name of the client; could be a company as well
	Address	varchar	Yes	The first line of an address
	Code		Yes	The postal code of the address
	Town	varchar	Yes	The town related to the address
	Telephone	varchar	Yes	The client's telephone number
	Email	varchar	Yes	The client's email address
	Website	varchar	Yes	The client's website URL
Staff table	Surname	varchar	No	The surnames of the staff
	Other names	varchar	No	Other names for the staff
	Id	varchar	No	The national ID or passport number of the staff
	Type	varchar	No	The type of employer whether advocate, clerk, manager or partner
	Gender	varchar	No	The gender of the staff
	Date of birth	date	No	The date of birth of the staff
	Date of employment		No	The date of employment of the staff
	Status	varchar	No	The type of employment whether permanent or temporary
	Address	varchar	Yes	The home address of the staff

	Code	varchar	Yes	The area code related to the address
	Town	varchar	Yes	The town related to the address
	Office Number	varchar	No	The office number assigned to the staff
	Mobile Number	varchar	Yes	The personal mobile number of the staff
	Email	varchar	No	The email address of the staff
	Next of Kin	varchar	No	The next of kin of the staff
	Relationship	varchar	No	The relationship of the next of kin to the staff example son, brother, friend
	Kin's telephone	varchar	No	The next of kin's telephone number
	Nssf number	varchar	No	The National Social Security number of the staff
	Nhif number	varchar	No	The National Health insurance number of the staff
	KRA	varchar	No	The Revenue authority number of the staff
	Bank	varchar	No	The bank where the staff's salary will be disbursed
	Branch	varchar	No	The branch related to the staff's account
	Account	varchar	No	The account number of the staff
	Secret code	varchar	No	The secret code generated when the staff has been added to the system
	Password	varchar	yes	The password set by the staff upon logging in
Files	Number	varchar	No	The file number. This may include the court and case number
	Name	varchar	No	The name of the file example client A versus person B

	Type	varchar		The type of case example family, criminal or dispute
	Client ID	varchar	No	The clients' id as selected from the system when the client has been selected
	Opening date	date	no	The date the case was reported or the file was opened
	Files	varchar	yes	The path/directory to the uploaded files related to the case
	Advocate ID Closing date	varchar	No	The id of the advocate responsible for that cases/file
Library	Number	varchar	No	The id of the reference material(the ISBN number)
	Name	varchar	No	The name of the reference material
	Type	varchar	No	The type of reference example family, conveyancing, dispute or criminal
	Upload date	date	No	The date that the reference material was uploaded
	Files	varchar	No	The path/directory to the uploaded file of the reference material
	Advocate ID	varchar	no	The id of the staff who uploaded the reference material

Figure 3.8 and 3.9 below show the activity diagrams for the user of the system.

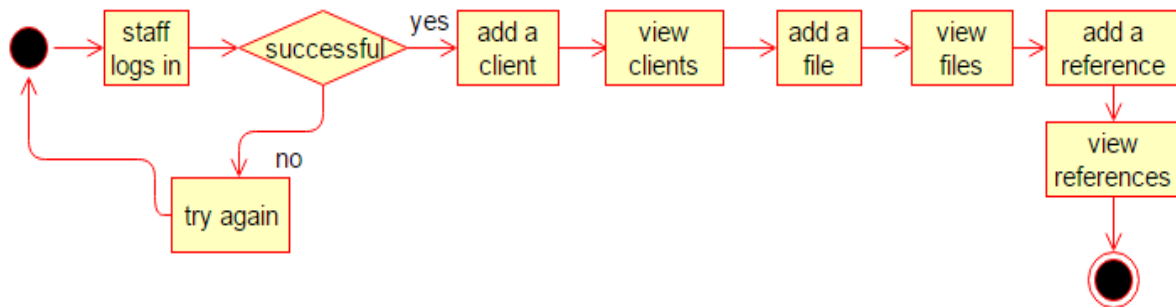


Figure 3.7 Activity Diagram for the advocate

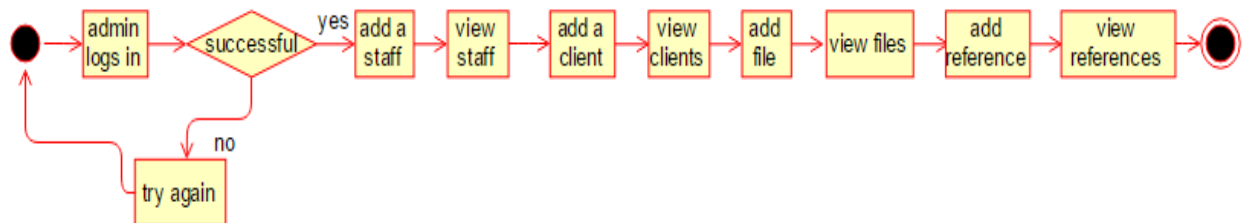


Figure 3.8 Activity diagram for the admin

3.5 Class Diagrams

- The Client class is responsible for all the activities that involve the client. Here is the class diagram

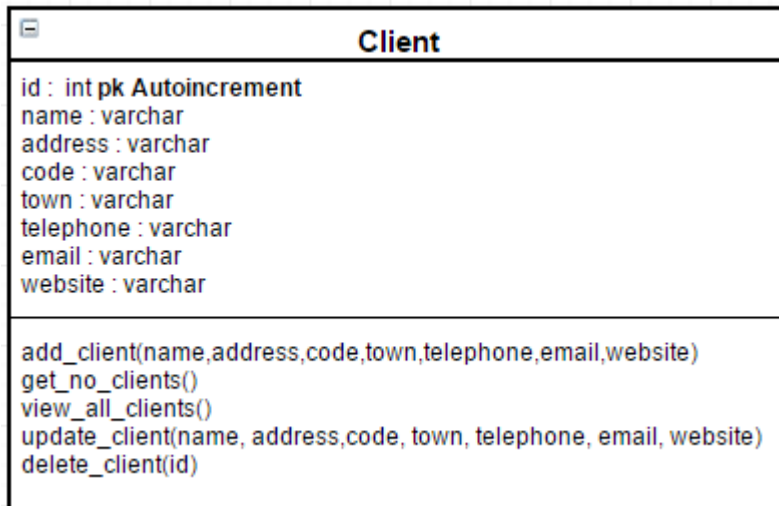


Figure 3.9 Client Class

- Staff class

Staff
id : varchar pk surname : varchar othernames : varchar type : varchar gender : varchar dateOfBirth : date dateOfEmployment : date status : varchar address : varchar code : varchar town : varchar officeNo : varchar mobileNo : varchar email : varchar kin : varchar relation : varchar telephone : varchar nssf : varchar nhif : varchar kra : varchar bank : varchar branch : varchar account : varchar secretcode : varchar pword : varchar
add_staff(surname,othernames,id,type,gender,dateOfBirth,dateOfEmployment,status,address,code,town,officeNo,mobileNo,email,kin,relation,telephone,nssf,nhif,kra,bank,branch,account) get_no_staff() view_all_staff() staff_login(email,pword)

Figure 3.10 Staff Class

- File class

File
number : varchar pk name : varchar type : varchar clientID : int fk openingDate : date files : varchar advocateID : varchar fk
add_file(number,name,type,clientID,openingDate,files,advocateID) get_no_files() get_no_open_files() get_no_closed_files() view_all_files() view_one_file(number) update_File(number,name,type,clientID,openingDate,files,advocateID)

Figure 3.11 File Class

- Library Class

Library
number : varchar pk name : varchar type : varchar uploadDate : date files : varchar advocateID : varchar fk
add_resource(number,name,type,uploadDate,files,advocateID) get_no_resources() view_all_resources() view_one_resource(number)

Figure 3.12 Library Class

Chapter 4: Implementation

This section describes how the Legal Management System was implemented. A brief description of the major tasks involved in the implementation, the overall resources needed to support the implementation effort (such as hardware, software, facilities, materials, and personnel) are highlighted. The plan was developed during the design phase and was updated during the development phase. The outline shows the structure of the implementation.

4.1 Description of Implementation

The implementation of this software followed the MVC approach mainly because the three models needed to be separated from each other so that the backend components could be adapted for other law firms. The application issues commands to the controller which is the PHP aspect of the code. The command alters the model which is the database aspect and the resulting change is displayed on the view HTML, JS and CSS.

4.2 Major Tasks

The system facilitates the management of files in the database by allowing the user to add, update and delete files. This feature has the highest precedence because hard copy documents are hard to manage especially if they are not well arranged.

The software is mainly accessed through a web interface and as such multiple users are allowed to login. The managers can then create new accounts for the subordinate staff and can also update their details when necessary. In the case that the employee leaves the firm, the manager is able to disable the login details of that employee.

4.3 Security

- **System Security Features**

The system has a login page where all the users have to authenticate themselves before they can access the system.

- **Security during Implementation**

During implementation the computers used for the development was connected to the Ashesi server which is a secured sever.

4.4 Software and Tools

During the implementation of this software various tools and software were employed as below;

- 1 Front End: HTML, CSS, JAVASCRIPT
- 2 Back End: AJAX, PHP
- 3 Database: MYSQL

In this section the reasons as to why these platforms are suitable for the software are highlighted.

- **HTML:** HTML is a markup language that is used for the front end of the web design. HTML was used because of its high speed and loading time. Also from research, all browsers support HTML. HTML is the base language that the JavaScript, CSS and PHP codes will plug in. HTML provides the content that CSS styles, JavaScript enhances while PHP links to the database.
- **PHP:** PHP over the years has become a very popular language for web development. PHP was chosen because it is easy to use and includes simple formats. Also, PHP offers incomparable usability and efficiency when used for website development. PHP when used to develop websites includes fast data processing features. PHP is also compatible

across all platforms such as windows, UNIX, etc. Also there is a lot of documentation on the internet and PHP can connect to MySQL.

- **CSS:** Cascading Style Sheet was used for describing the presentation that was written in HTML. Thus the primary aim of using CSS was to style the project and make it usable and look presentable to the users of the system.
- **JAVASCRIPT:** JavaScript was basically used to create a responsive page since using only HTML would mean the page is static. JavaScript would be used to make the URL responsive by sending requests to AJAX.
- **AJAX:** AJAX was used to make request to the other classes in the system.
- **JSON** –This was used to store and format data before sending it across the software interfaces
- **MYSQL:** MYSQL was used to implement the creation of databases in the system. This is because it could handle large volumes of data.
- **Apache Server (Xampp):** Xampp was used because it enabled the software to run and implement the database both locally and on the server.
- **Libraries:** In this project, Angular JS was an important library, this is because Angular JS framework enables the developer to extend HTML by acting as the controller portion of the MVC model

4.5 Screenshots of Implementation

Here are a few screen shots of the implementation.

This is the log in page for both the admin and the advocate

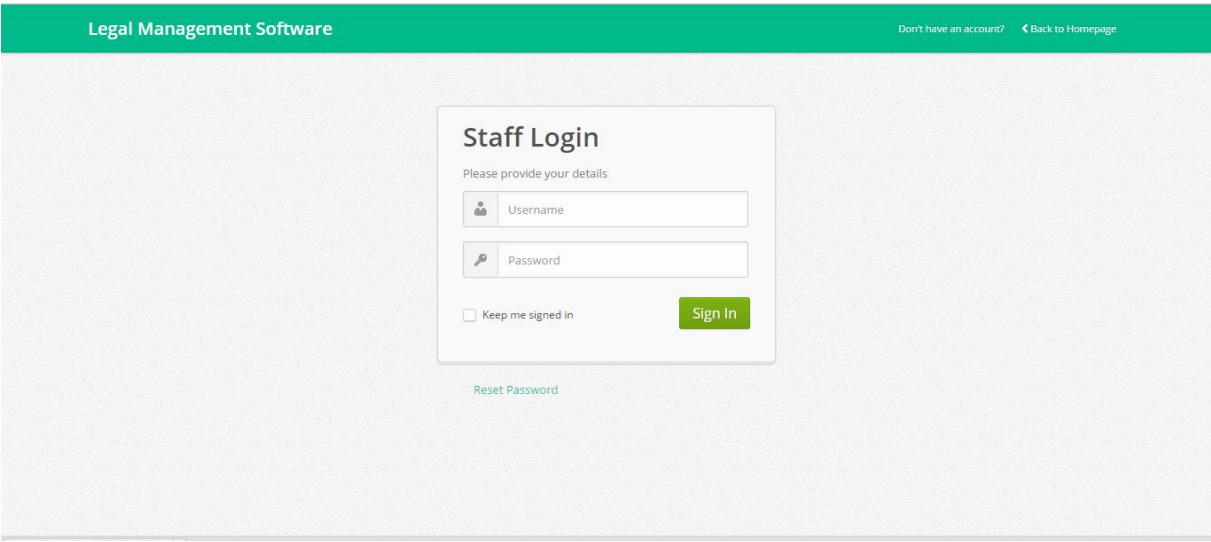


Figure 4.1 Login Page

This is the home page of the application.

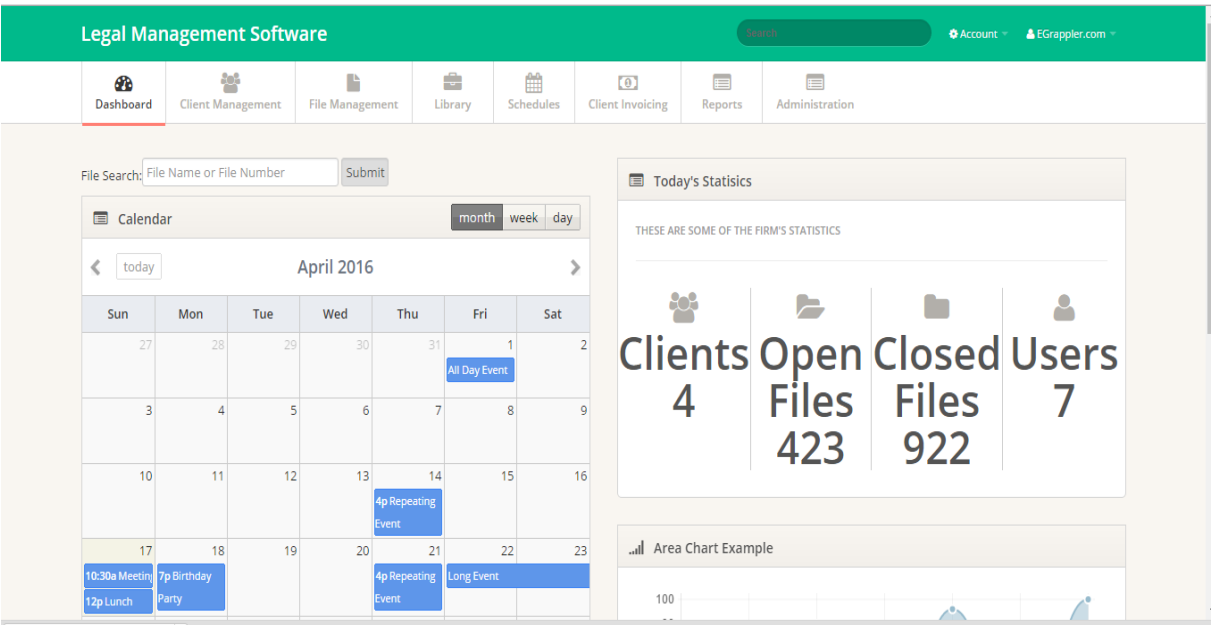


Figure 4.2 The Admin Home Page

Below is the File management screen where the user views the list of files and select to view the full details related to the file.

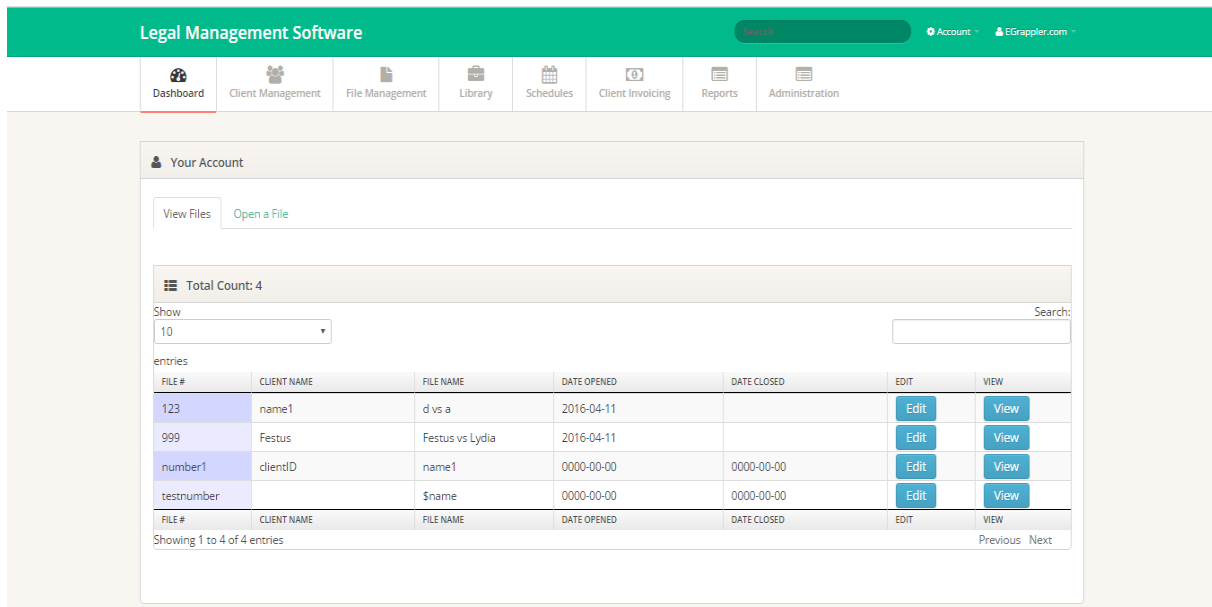


Figure 4.3 View Files Page

The figure below is a screen for opening a new file when a new case comes up.

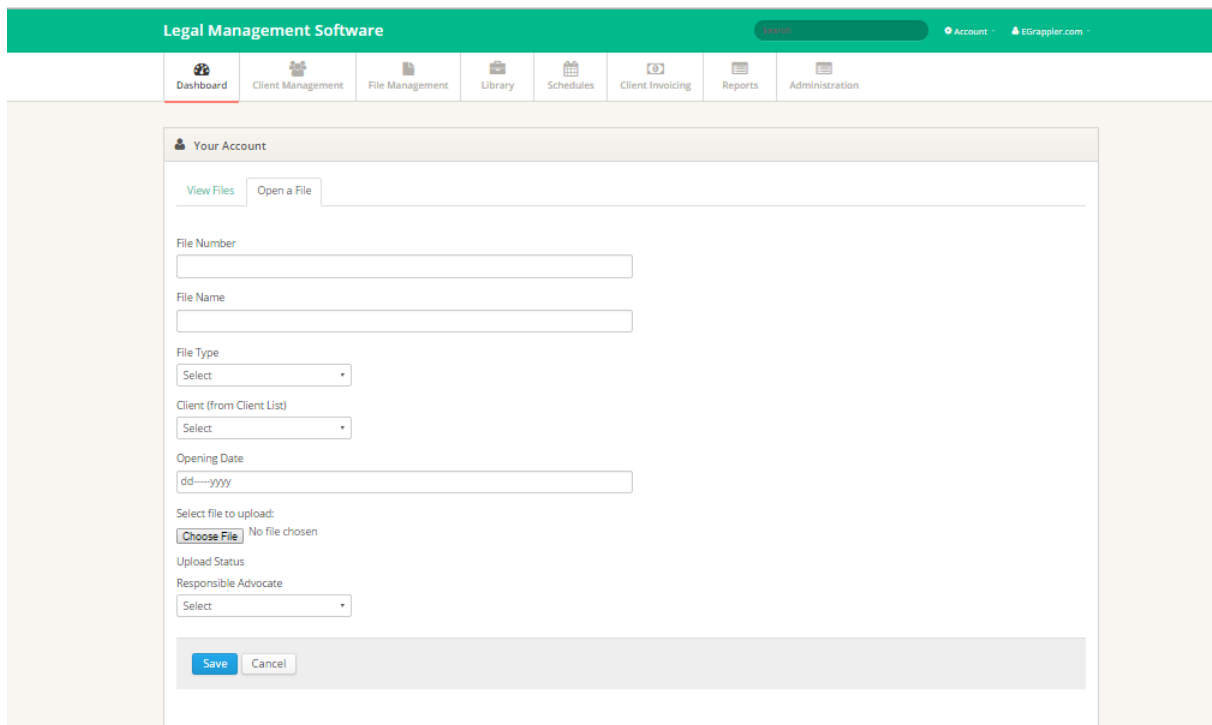


Figure 4.4 Add File Page

Chapter 5: Testing and Results

During and after implementation of the software some form of testing was done to ensure that the components worked as expected. The testing approach employed involved the development testing. This testing was to ensure that the software meets the user requirements and to detect undesirable behavior.

5.1 Development Testing

Development testing was used during the implementation stages of development. This involves continually testing the functionality of the components as the implementation proceeds. Development testing is divided into three components; unit, component and system testing.

5.1.1 Unit Testing

In the unit testing various methods of the main functionality were tested. Once the function was implemented, it was immediately tested before it was integrated into the rest of the system. In this testing, the correct values were used with expected outputs and also wrong values with expected error messages. The wrong values included data with wrong datatypes, inserting duplicate values, omission of data and searching/deleting values that are not in the database.

The SQL statements were also tested by running them directly on phpmyadmin before adding them to the code.

The functions to add and view were included in the test. The following code is a sample test code from the client class.

```

<?php
class test extends PHPUnit_Framework_TestCase{
function testclient(){
include("client.php");
$test = new client();

echo $this->assertTrue($test-
>add_client("testName","1111","2222","testTown","+233542616910","tes
t@gmail.com","www.test.com"));

echo $this->assertTrue($test->get_no_clients());
echo $this->assertTrue($test->view_all_clients());

echo $this->assertTrue($test-
>update_client("testNameUpdate","11112222","22221111","testTownupdat
e","+233540000000","testupdate@gmail.com","www.testupdate.com"));
}
}
?>

```

After the test the results showed that the functions worked correctly.

5.1.2 Component Testing

This is testing of individual modules of the software. For instance the whole file management process, the client management process, the reference management process and the staff management were tested. The login was also part of the component testing.

Since the unit testing had already been done, it was quite easy to test the component testing. During component testing, a challenge related to redirecting the page to the home page after logging in was uncovered and fixed.

5.1.3 System Testing

This involves testing the entire system to ensure that all components are working as expected. This came a lot easier because of the unit test and the component test that I had already done.

After implementation, the functionality of the system was tested to ensure that the added functionality did not affect the existing/already working part.

5.2 User Testing

User testing was done after implementation where a third party was asked to run the software and any challenges were observed. The testers logged into the system as different users of the system and tested to ensure the smooth running of the processes.

The admin account was tested by creating an account and after logging the user was allowed to perform the tasks as specified in the user requirements. The admin also created an account for a staff. The admin also tried to log in with a wrong password and as expected there was a prompt that the password was wrong.

The tester also logged in as the staff created by the admin. Once the tester logged in the feature for staff management was unavailable as that was only granted to the admin. The staff was able to perform other tasks as expected by the user requirements in Chapter 2. Once again the staff tried to log in with wrong password and there was a prompt with denial of service since the password was wrong.

Chapter 6: Conclusions and Recommendations

6.1 Conclusion

Throughout the development of this project I have been able to learn new concepts as well as strength my skills in other tools such as HTML, AJAX, JavaScript, Bootstrap, php and MYSQL.

After testing the software the following functions were proved to be working as expected.

- ✓ New clients can be registered into the software
- ✓ The clients can be viewed in a well formatted table
- ✓ The admin can add a new staff into the database
- ✓ The admin can view a table containing the details of the other staff
- ✓ The users can add/open a new file/case
- ✓ The user can sort/search for a given file from the database using any of the parameters
- ✓ The user can view the details of the a given case
- ✓ The user can upload/add a reference material which is accessible to all the user of the system
- ✓ The user can refer to the reference materials at any point on time
- ✓ Reports on clients and/or files can be generated instantly

The legal management software will go a long way in helping the law firms to effectively manage their documents.

6.2 Challenges

Malicious hackers or technical glitches can open up lawyer's cloud storage to breaches that could lead to violation of the obligation of confidentiality. So the users should not rely on

the cloud storage as the only primary location of the clients' files. This can be addressed by saving a copy of the files in the cloud in the local storage. The files could also be encrypted before uploading them to the cloud (Ezor, 2014).

6.3 Future work

Due to the constraint in time I highly recommend that in the next versions of the Legal Management software the following features be implemented.

- More security features should be implemented so that the information is secure.
- The software should be linked to a calendar engine for synchronization and easy planning.
- The email engine should also be incorporated so that the clients can be directly contacted on the progress of their cases.
- A module should be added such that after collection of large amounts of data, the system can apply data mining techniques to predict the probability of winning the case.

References

Ezor, J. I. (2014, January 1). *The Risks of Innovation Through Technology in Legal Practice*.

Retrieved March 19, 2016, from Law Practice Today:

http://www.americanbar.org/content/newsletter/publications/law_practice_today_home/lpt-archives/2014/january14/the-risks-of-innovation-through-technology-in-legal-practice.html

George, A. (2015, August 20). *4 Key Differences Between Today's Law Firms and Law Firms in the Future*. Retrieved September 21, 2015, from Lexicata:

<https://lexicata.com/blog/4-key-differences-between-todays-law-firms-and-law-firms-in-the-future/>

Groot, B. (2014, June 20). *New trends facing lawyers*. Retrieved September 21, 2015, from SAFLII (Southern African Legal Information Institute):

<http://www.saflii.org/za/journals/DEREBUS/2014/109.html>

Jacoby, C. (2015, January). *Deconstructing the Myth of Low Technology Adoption in Law Firms*, Volume 16, Number 1. Retrieved September 21, 2015, from Legal Solutions:

<https://info.legalsolutions.thomsonreuters.com/signup/newsletters/practice-innovations/2015-jan/article1.aspx>

LexisNexis. (2015, March 1). *Survey on South Africa's small law firms*. Retrieved April 1, 2016, from De Rebus: <http://www.derebus.org.za/survey-on-south-africas-small-law-firms/>

tech. (2015, October 21). Retrieved March 31, 2016, from Zytrax:

<http://www.zytrax.com/tech/lang/ruby/ror.html>

Weaver, C. (2016, March/April). Creating a Functional Ecosystem. *Facts and Findings* , 30-

31. Retrieved March 19, 2016, from

<https://www.nala.org/sites/default/files/Technobits-March-April-2016-Facts-Findings.pdf>