

**FINAL REPORT**

***Digital Lawyer Street***

**Submitted by**

*Anas Makki (2018-GCUF-058102)*

*Abdul Rahman (2018-GCUF-058166)*



**Department of Computer Science  
Govt. Graduate College Samanabad Faisalabad**

**2022**

**CERTIFICATE**

This is to certify that ***Anas Makki & Abdul Rehman*** bearing Registration No. ***2018-GCUF-058102 & 2018-GCUF-058166*** has completed the final project titled ***“Digital Lawyer Street”*** at the **Department of Computer Science**, **Govt. Graduate College Samanabad Faisalabad**, to fulfill the partialrequirement for the degree of ***BS - CS****.*

**Supervisor**

**Sir Ghulam Mustafa** **Signature: \_\_\_\_\_\_\_\_\_\_\_\_**

**Co-Supervisors**

**Member 1: Mam Sania** **Signature: \_\_\_\_\_\_\_\_\_\_\_\_**

**Member 2: Mam Qoseen** **Signature: \_\_\_\_\_\_\_\_\_\_\_\_**

**Member 3: Mam Ambreen** **Signature: \_\_\_\_\_\_\_\_\_\_\_\_**

**External Supervisor**

**Supervisor Name:** **Signature: \_\_\_\_\_\_\_\_\_\_\_\_**

**Project Coordinator Head of Department**

**Principal**

**DECLARATION**

The work reported in this project was carried out by me under the supervision of **Project Supervisor**, **Sir Ghulam Mustafa**, at **Government Graduate College Samanabad Faisalabad**.

I hereby declare that this project and the contents of the project are the product of my research and no part has been copied from any other written or published source (except the references, standard mathematical or genetics models/equation/formulas/protocol, etc.).

I further declare that this work has not been submitted for the award of any other degree/diploma.

The institution may take action if the provided information is found inaccurate at any stage.

**Name : Anas Makki**

**Registration No. : 2018-GCUF-058102**

**Name : Abdul Rahman Registration No. : 2018-GCUF-058166**

**ACKNOWLEDGEMENT**

He confided in me. I thank all who in one way or another contributed to the completion of this report. First, I thank ALLAH ALMIGHTY, most magnificent and most merciful, for all his blessings. Then I am so grateful to the Department of Computer Science of Govt. Graduate College Samanabad Faisalabad for making it possible for me to study here. My special and heartily thanks to my supervisor, Sir. Ghulam Mustafa encouraged and directed me. His challenges brought this work towards completion. It is with his supervision that this work came into existence. For any faults, I take full responsibility. I am also deeply thankful to my informants. I want to acknowledge and appreciate their help and transparency during my research. I am also so thankful to my fellow students whose challenges and productive critics have provided new ideas to the work. Furthermore, I also thank my family who encouraged me and prayed for me throughout the time of my research. May ALLAH Almighty richly bless all of you [Ameen]

**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 recommendations, and at the same time lawyers had a challenge reaching out to their clients. Development was slow and tedious as work was conducted on dusty and traditional files.

The basis of this project is to bridge the technology gap between lawyers and users in Pakistan. Lawyers that deal with legal services will most likely handle a huge quantum of inventory and have to make sure that these files are readily available when it is required. 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 at different locations.

The goal of this project is to develop an application that will aid lawyers to handle cases from wherever they are and quickly generate reports.

Overall, this project will help in searching for a user to find lawyers and vice versa. As a result, a better legal platform will come into being.

**Table of Contents**

[Chapter 1 1](#_Toc110884792)

[1.1 Purpose 1](#_Toc110884793)

[1.2 How it works 1](#_Toc110884794)

[1.2.1 Register Lawyer 1](#_Toc110884795)

[1.2.2 Evaluate Lawyers 1](#_Toc110884796)

[1.2.3 Feedbacks 1](#_Toc110884797)

[1.2.4 Queries 1](#_Toc110884798)

[1.2.5 Register Client 2](#_Toc110884799)

[1.2.6 Case Information 2](#_Toc110884800)

[1.2.7 Appointment 2](#_Toc110884801)

[1.2.8 Hire Staff 2](#_Toc110884802)

[1.3 Project Scope 2](#_Toc110884803)

[1.4 Project Planning 2](#_Toc110884804)

[1.5 Project Modules 4](#_Toc110884805)

[1.5.1 Admin Module: 4](#_Toc110884806)

[1.5.2 Lawyer Module: 4](#_Toc110884807)

[1.5.3 User (Client) Module: 4](#_Toc110884808)

[1.5.4 Team Member Module: 4](#_Toc110884809)

[Chapter 2 Background 5](#_Toc110884810)

[2.1 Background Research 5](#_Toc110884811)

[2.2 Existing Technology 5](#_Toc110884812)

[2.3 Area of Study 6](#_Toc110884813)

[2.4 Reason for the Project 6](#_Toc110884814)

[2.4.1 Accuracy of Information 6](#_Toc110884815)

[2.4.2 Efficiency 7](#_Toc110884816)

[2.4.3 Easy to Use 7](#_Toc110884817)

[2.4.4 Security 8](#_Toc110884818)

[2.4.5 User Friendly 8](#_Toc110884819)

[2.4.6 Time-Saving 9](#_Toc110884820)

[2.5 Objective of the project 9](#_Toc110884821)

[2.6 Methodology 10](#_Toc110884822)

[Chapter 3 System Requirement Analysis 11](#_Toc110884823)

[3.1 System Functional Requirements 11](#_Toc110884824)

[3.1.1 Project Interactivity Plan 11](#_Toc110884825)

[3.1.2 Signup Information 11](#_Toc110884826)

[3.1.3 Login Information 11](#_Toc110884827)

[3.1.4 Lawyer Record 11](#_Toc110884828)

[3.1.5 SRS Document (System Requirements Specification) 13](#_Toc110884829)

[3.2 Non-Functional Requirements of the system 13](#_Toc110884830)

[3.3 Hardware & Software Requirements 14](#_Toc110884831)

[3.3.1 Software Requirements 14](#_Toc110884832)

[3.3.2 Hardware Requirements 14](#_Toc110884833)

[3.4 Introduction of Tools 15](#_Toc110884834)

[3.4.1 VS Code 15](#_Toc110884835)

[3.4.2 HTML (Hyper Text Markup Language) 15](#_Toc110884836)

[3.4.3 CSS (Cascading Style Sheet) 16](#_Toc110884837)

[3.4.4 Bootstrap 16](#_Toc110884838)

[3.4.5 JavaScript 17](#_Toc110884839)

[3.4.6 JQuery 17](#_Toc110884840)

[3.4.7 PHP 18](#_Toc110884841)

[3.4.8 AJAX 18](#_Toc110884842)

[3.4.9 MySQL 19](#_Toc110884843)

[Chapter 4 System Design 20](#_Toc110884844)

[4.1 Use case – fully dressed 20](#_Toc110884845)

[4.1.1 Importance of Use Case Diagrams 20](#_Toc110884846)

[4.1.2 Use Case Diagram objects 20](#_Toc110884847)

[4.1.3 Use Case Diagram 22](#_Toc110884848)

[4.2 WBS fully dressed 26](#_Toc110884849)

[4.3 System Sequence diagram 27](#_Toc110884850)

[4.4 Class diagram 28](#_Toc110884851)

[4.5 ER (Entity Relationship Model) Model 30](#_Toc110884852)

[4.6 Data Flow Diagram 31](#_Toc110884853)

[4.6.1 Symbols and Notations Used in DFDs 31](#_Toc110884854)

[4.6.2 Data Flow Diagram of Digital Lawyer Street (DLS) 31](#_Toc110884855)

[Chapter 5 Testing & Implementation 34](#_Toc110884856)

[5.1 Testing Methods 34](#_Toc110884857)

[5.1.1 Black Box Testing 34](#_Toc110884858)

[5.1.2 White Box Testing 35](#_Toc110884859)

[5.1.2 Grey Box Testing 35](#_Toc110884860)

[5.2 Functional vs. non-functional testing 36](#_Toc110884861)

[5.3 Functional Testing 37](#_Toc110884862)

[5.3.1 Unit Testing 37](#_Toc110884863)

[5.3.2 Integration Testing 38](#_Toc110884864)

[5.3.3 System Testing 39](#_Toc110884865)

[5.3.4 Acceptance Testing 39](#_Toc110884866)

[5.4 Non- functional Testing 40](#_Toc110884867)

[5.4.1 Load testing 40](#_Toc110884868)

[5.4.2 Stress testing 41](#_Toc110884869)

[5.4.3 Endurance testing 42](#_Toc110884870)

[5.4.4 Spike testing 42](#_Toc110884871)

[5.4.5 Usability testing 42](#_Toc110884872)

[5.4.6 Compatibility testing 43](#_Toc110884873)

[5.5 Test Cases 44](#_Toc110884874)

[Chapter 6 User Manual 51](#_Toc110884875)

[6.1 Main/Home view 51](#_Toc110884876)

[6.2 Our Lawyers Page 51](#_Toc110884877)

[6.3 Services Page 52](#_Toc110884878)

[6.4 Testimonial Page 52](#_Toc110884879)

[6.5 About Us Page 53](#_Toc110884880)

[6.6 Contact Us Page 53](#_Toc110884881)

[6.7 Footer 53](#_Toc110884882)

[6.8 DLS Statistics 54](#_Toc110884883)

[6.9 Forget Password 54](#_Toc110884884)

[6.10 Lawyer Registration Page 55](#_Toc110884885)

[6.11 User Login 56](#_Toc110884886)

[6.12 User Dashboard 56](#_Toc110884887)

[6.13 Lawyer Dashboard 57](#_Toc110884888)

[6.14 Lawyer Requests 57](#_Toc110884889)

[6.15 List of Lawyers 58](#_Toc110884890)

[6.16 Database Connection 58](#_Toc110884891)

[6.17 Front End Coding 59](#_Toc110884892)

[6.18 Backend Coding 60](#_Toc110884893)

[6.19 Database 60](#_Toc110884894)

[6.20 Database Table Structure 61](#_Toc110884895)

**List of Tables**

[Table 1: Software Requirement Specification 13](#_Toc110884896)

[Table 2: Software Requirements 14](#_Toc110884897)

[Table 3: Hardware Requirements 14](#_Toc110884898)

[Table 4: WBS Levels 27](#_Toc110884899)

[Table 5: Test Case 1 44](#_Toc110884900)

[Table 6: Test Case 2 44](#_Toc110884901)

[Table 7: Test Case 3 45](#_Toc110884902)

[Table 8: Test Case 4 45](#_Toc110884903)

[Table 9: Test Case 5 46](#_Toc110884904)

[Table 10: Test Case 6 46](#_Toc110884905)

[Table 11: Test Case 7 47](#_Toc110884906)

[Table 12: Test Case 8 47](#_Toc110884907)

[Table 13: Test Case 9 48](#_Toc110884908)

[Table 14: Test Case 10 48](#_Toc110884909)

[Table 15: Test Case 11 49](#_Toc110884910)

[Table 16: Test Case 12 49](#_Toc110884911)

[Table 17: Test Case 13 50](#_Toc110884912)

[Table 18: Test Case 14 50](#_Toc110884913)

**List of Figures**

[Figure 1: SLDC Waterfall Model 3](#_Toc110884914)

[Figure 2: Accuracy 7](#_Toc110884915)

[Figure 3: Efficiency 7](#_Toc110884916)

[Figure 4: Easy to Use 8](#_Toc110884917)

[Figure 5: Security 8](#_Toc110884918)

[Figure 6: User Friendly 9](#_Toc110884919)

[Figure 7: Time Saving 9](#_Toc110884920)

[Figure 8: Visual Studio Code 15](#_Toc110884921)

[Figure 9: HTML (Hyper Text Markup Language) 16](#_Toc110884922)

[Figure 10: CSS (Cascading Style Sheet) 16](#_Toc110884923)

[Figure 11: Bootstrap 17](#_Toc110884924)

[Figure 12: JavaScript 17](#_Toc110884925)

[Figure 13: jQuery 18](#_Toc110884926)

[Figure 14: PHP 18](#_Toc110884927)

[Figure 15: AJAX 19](#_Toc110884928)

[Figure 16: MySQL 19](#_Toc110884929)

[Figure 17: Actor 21](file:///C:\xampp\htdocs\FYP-DLS\z-files\FYP-DLS-Report.docx#_Toc110884930)

[Figure 18: Use Case 21](file:///C:\xampp\htdocs\FYP-DLS\z-files\FYP-DLS-Report.docx#_Toc110884931)

[Figure 19: System 21](file:///C:\xampp\htdocs\FYP-DLS\z-files\FYP-DLS-Report.docx#_Toc110884932)

[Figure 20: Package 22](file:///C:\xampp\htdocs\FYP-DLS\z-files\FYP-DLS-Report.docx#_Toc110884933)

[Figure 21: Relationships 22](#_Toc110884934)

[Figure 22: User/Client Use Case 23](#_Toc110884935)

[Figure 23: Lawyer Use Case 24](#_Toc110884936)

[Figure 24: Admin Use Case 24](#_Toc110884937)

[Figure 25: Use Case Diagram Fully Dressed 25](#_Toc110884938)

[Figure 26: WBS (Work Breakdown Structure) 26](#_Toc110884939)

[Figure 27: Sequence Diagram 28](#_Toc110884940)

[Figure 28: Class Diagram 29](#_Toc110884941)

[Figure 29: ER (Entity Relationship Diagram) 30](#_Toc110884942)

[Figure 30: DFD Types 31](file:///C:\xampp\htdocs\FYP-DLS\z-files\FYP-DLS-Report.docx#_Toc110884943)

[Figure 31: Context Level DFD 31](#_Toc110884944)

[Figure 32: Level 1 DFD 32](#_Toc110884945)

[Figure 33: Level 2 DFD 33](#_Toc110884946)

[Figure 34: Black-box Testing 34](#_Toc110884947)

[Figure 35: White Box Testing 35](#_Toc110884948)

[Figure 36: Grey Box Testing 35](#_Toc110884949)

[Figure 37: Functional Testing 37](#_Toc110884950)

[Figure 38: Unit Testing 38](#_Toc110884951)

[Figure 39: Integration Testing 39](#_Toc110884952)

[Figure 40: System Testing 39](#_Toc110884953)

[Figure 41: Acceptance Testing 40](#_Toc110884954)

[Figure 42: Load Testing 41](file:///C:\xampp\htdocs\FYP-DLS\z-files\FYP-DLS-Report.docx#_Toc110884955)

[Figure 43: Stress Testing 41](#_Toc110884956)

[Figure 44: Compatibility Testing 43](file:///C:\xampp\htdocs\FYP-DLS\z-files\FYP-DLS-Report.docx#_Toc110884957)

[Figure 45: Home Screen 51](#_Toc110884958)

[Figure 46: Lawyers Page 51](#_Toc110884959)

[Figure 47: Services 52](#_Toc110884960)

[Figure 48: Testimonials 52](#_Toc110884961)

[Figure 49: About Us 53](#_Toc110884962)

[Figure 50: Contact Us 53](#_Toc110884963)

[Figure 51: Footer 53](#_Toc110884964)

[Figure 52: DLS Statistics 54](#_Toc110884965)

[Figure 53: Forget Password 54](#_Toc110884966)

[Figure 54: Lawyer Registration 55](#_Toc110884967)

[Figure 55: Login Area 56](#_Toc110884968)

[Figure 56: User Dashboard 56](#_Toc110884969)

[Figure 57: Lawyer Dashboard 57](#_Toc110884970)

[Figure 58: Lawyer List 58](#_Toc110884971)

[Figure 59: Database Connection 58](#_Toc110884972)

[Figure 60: Front End 59](#_Toc110884973)

[Figure 61: Back End 60](#_Toc110884974)

[Figure 62: Database 60](#_Toc110884975)

[Figure 63: Table Structure 61](#_Toc110884976)

Chapter 1 Introduction

## 1.1 Purpose

The current scenario used by the lawyers in maintaining heavy files works for keeping the records of their cases. **Digital Lawyer Street** is a new website for people to find lawyers via the use of the internet and also for the lawyers to maintain records of the cases they are dealing with. A lot of people need to contact their lawyers regularly or find appropriate lawyers for their cases. The goal of the project is to provide a flexible way for people to fulfill their requirements. The admin of the website can register/approve different Lawyers and each lawyer can in turn register his/clients and even his/her staff members. The lawyers and clients can work out their cases. This site provides a flexible way for lawyers and clients to handle their cases in a well-formatted way online through any part of Pakistan.

## 1.2 How it works

### 1.2.1 Register Lawyer

**Digital Lawyer Street** application hires lawyers in different practice areas. Admin verify requested lawyers on this system. On DLS registered lawyers receive the cases that clients apply or request and manage the case for whom clients hire.

### 1.2.2 Evaluate Lawyers

Admin is responsible to evaluate the lawyers by checking and verifying their **License Numbers** or any other evaluation method.

### 1.2.3 Feedbacks

Clients on DLS can give feedback about our system. Visitors of DLS can get help using those feedback.

### 1.2.4 Queries

New visitors can ask any query about DLS. The admin will provide an answer to any query on WhatsApp or email.

### 1.2.5 Register Client

The user can send a hiring request to any lawyer. The lawyer can contact on Whatsapp or Email a client. After approval from the lawyer DLS stores the hired client’s information. The lawyer can manage client information.

### 1.2.6 Case Information

Lawyers can manage information about cases of hired clients.

### 1.2.7 Appointment

Lawyers can make the schedules for handling appointments with their clients, and notify them on WhatsApp.

### 1.2.8 Hire Staff

Hire the staff to support the respective lawyer.

## 1.3 Project Scope

The scope of this application is limited in a way that DLS does have not its chatting system. Moreover, only the admin can approve lawyers by approving the registration of the lawyer for this website. The admin will permit scrutiny or evaluation of details of the lawyer. Only lawyers can save technical details of their clients and staff after their registration. Further, the lawyer can maintain details of his client’s cases. Clients who are registered by the lawyer can view their case details as entered and managed by the lawyer. Using this website, non-member users can also find profiles of different lawyers registered with this website.

## 1.4 Project Planning

The purpose of this project is to hire lawyers online from anywhere in Pakistan. The goal of this section is to provide a set of recommendations that will help you plan an appropriately successful project. In this section, we used the life cycle model employed broadly at Microsoft. This model is a combination of iterative and waterfall life cycle models. In this model, five boundaries are defining a sequential set of milestones for the project. The phases, in order of execution, are as follows

* **Feasibility Study:** The proposed system is technically possible moreover there is no need for consideration over economical scrutiny because it is a final year project to get the degree.
* **Requirement analysis and specifications:** To make a user-friendly Online/Digital Lawyer System through which users can find lawyers easily.
* **Design**: Based on the functional requirements, physical design specifications are created and prototyping is conducted to verify design ideas and investigate the capabilities.
* **Coding and Unit Testing:** Using the design and functional specifications, the coding is done. Moreover, unit testing is also done by testing each module.
* **Integration and System Testing:** This is the process of testing the product to verify that it performs according to the specifications. Moreover, System is installed on the system to function.

**Maintenance:** The proposed system is ready to handle and adopt new changes and complexities according to changing environment of the world, such as policies, which change over time.

Over time, the customer may ask for new features or functions in the software, which will be fulfilled.

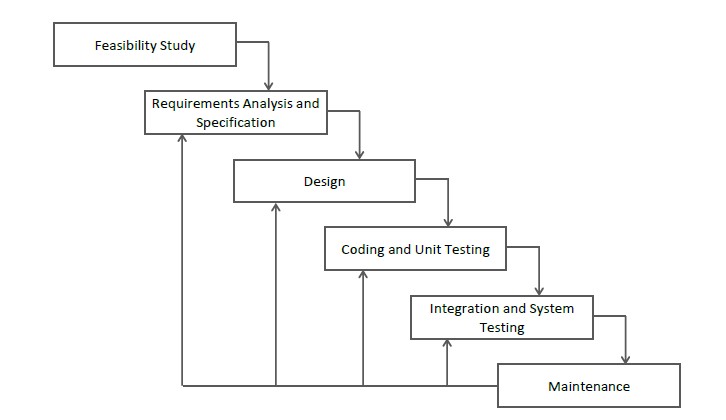
If any of the hardware and/or platform (such as operating system) of the target host changes, software changes are needed to keep adaptability.  


Figure 1: SLDC Waterfall Model

## 1.5 Project Modules

The entire project consists of 3 main modules, which are

* Admin Module
* Lawyer Module
* Client Module
* Team Member Module

### 1.5.1 Admin Module:

* Admin manages all lawyers.
* Admin verifies the lawyer whether he or she is legal or not.
* Admin can manage clients’ feedback.
* Admin can manage user queries.

### 1.5.2 Lawyer Module:

* Lawyers can accept or reject clients hiring requests.
* Lawyers can also register clients.
* Lawyers can manage case details.
* Lawyers can manage all appointments.
* Lawyers can manage clients' information
* Lawyers can add team members and assign them a task.

### 1.5.3 User (Client) Module:

* Client/User can hire a lawyer according to his/her choice.
* Client/User can see details of a hired lawyer.
* Client/User can contact hired lawyer.

### 1.5.4 Team Member Module:

* The team members can work on their assigned tasks and submit them.
* The team members can only see limited information provided by the lawyer.

# Chapter 2 Background

## 2.1 Background Research

Remember those times when you had to make phone calls to relatives or friends to find out the lawyers? When you had to hail a taxi by hand and rely on ads on walls to explore a better lawyer. When every single piece of information has to be found physically. And also a lawyer has to be hired physically by visiting the buildings or courts. That time was not a long time ago. Now a day the lifestyle of the people is different. People feel uncomfortable and time-consuming going to court for hiring a lawyer. So, online hiring saves a lot of time. Online hiring is usually available 24 hours a day and 7 days a week. So it is very convenient for them to hire a lawyer online. Technology has changed the law firm industry at an exponential rate in the last two decades.

An increasing amount of research has been conducted to understand the impacts of online finding lawyer development from the resident’s perspective. The driving force behind these phenomena can be attributed to the fact that the DLS has additionally played a vital role in social, cultural, and environmental impacts on people, destinations, and Pakistan.

The main purpose of this Web App is to facilitate the clients to find lawyers online because a customer cannot spend their precious time in markets trying to find out the best deal.

## 2.2 Existing Technology

Currently, several applications have been developed to address the issue of low technology consumption in law firms but some lawyers still shy away from using technology in their practice even though these applications 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 as pragmatists. 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. After all, these lawyers saw immediate value in changing their current style. This is also very practical because lawyers have cut their costs through the use of technology by eradicating a lot of their employees.

## 2.3 Area of Study

Technology nowadays plays a significant role in our lives. From simple machines to complex ones technology is used. This system is used by many lawyers in many cities to facilitate the users for online hiring them.

The law firm is one of the areas that can provide easiness in the community. To promote the DLS, the website must be accessible everywhere. A DLS website usually is a Lawyer to the client, whose main target is the general public. Websites can support all the core activities of a law firm. Like, such as managing appointments, adding cases, printing clients’ data, etc.

## 2.4 Reason for the Project

A Digital Lawyer Street website is a website that is dedicated to lawyers or advocates. The site may be focused on providing easiness to the user to find lawyers or advocates easily using the internet. Hundreds of new cases are registered in different courts in Pakistan, our aim is that people get an appointment online. My main motive is to promote the use of technology in Pakistan. The purpose of this project is to provide complete information about lawyers or advocates in a single click. Our priority will be our user and their legal requirements. If the online lawyer-finding industry emerges then the problem of rush in courts resolved too. Users will be able to find an appropriate lawyer with affordable fees and related skills.

### 2.4.1 Accuracy of Information

The accuracy of information or measurements is their quality of being true or correct, even in small details. The information of data organization in an information system can affect the speed, cost, and desired processing activities. Since all the calculation is done automatically, so the chances of error are very rare, which results in the accuracy of the system. The developed system provides a powerful searching mechanism. This can even search a minor record in less than a second because the stored data in a database is in accurate form due to validation checks. A computerized system’s accuracy is 100% while in manual it is 40%.



Figure 2: Accuracy

### 2.4.2 Efficiency

The developed system is greater efficient than the older one. Wrong data entry is impossible as different checks do not allow the entry of wrong data because of validations. The classification efficiency of the single-cycled classification model is intuitively worse than the multi-cycled model, as it requires more effort in collecting relatively comprehensive software measurement data to classify each unseen case. Software project managers must determine which type of classification model is best suited to their specific objectives since such a choice will significantly affect the accuracy and efficiency of the classification results.



Figure 3: Efficiency

### 2.4.3 Easy to Use

The source data input, review, and modification can be repeated any number of times on computer graphics. The power system graphics generation should be easy and efficient. The web app is very easy to use and operate.



Figure 4: Easy to Use

### 2.4.4 Security

Application security means many different things to many different people. Security should be explicitly at the requirement level. Security methods are under the high-level class and are considered to be an addition to the original software. The habits formed from initial programming can be for a long time. Security is considered a very critical issue for software systems. Software is itself a resource and thus must be afforded appropriate security. Software that is developed with security in mind is typically more resistant to both intentional attacks and unintentional failures.

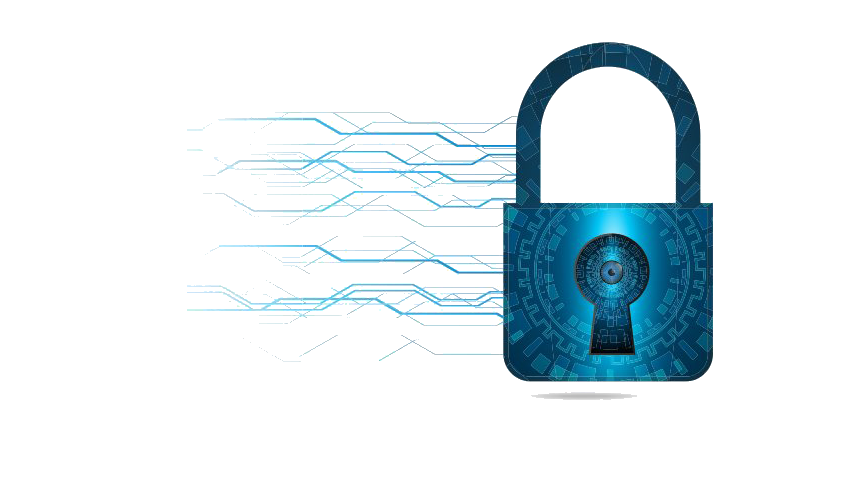


Figure 5: Security

### 2.4.5 User Friendly

The term user-friendly refers to anything that makes it easier for novices to use a computer. Menu-driven programs, for example, are considered more user-friendly than command-driven systems. Online help systems are another feature of user-friendly programs.

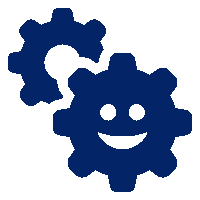


Figure 6: User Friendly

### 2.4.6 Time-Saving

Due to the high speed of processing, the proposed system takes less time to access information from the database, and as it is an online system so it is time saving than going physically to the office for lawyer hiring.



Figure 7: Time Saving

## 2.5 Objective of the project

In the attempt to appeal to lawyers to apply technology in their practice, the Digital Lawyer Street 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, and 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, tablets, 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 Pakistani 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 greater flexibility bearing in mind that it performs a range of tasks that would otherwise be done by several applications.

## 2.6 Methodology

The project consisted of the following components:

* This system provides the facility for the customer to hire a lawyer according to his/her choice.
* This system is also providing the facility of viewing lawyers’ profiles.
* The user may hire a lawyer from the relative court.
* After hiring client can get an appointment with the lawyer.
* Admin can approve and disapprove lawyers after scrutiny.
* Admin can also control full website through admin panel like practice areas, services, and testimonials.
* Admin can also give a response to the queries of users.

# Chapter 3 System Requirement Analysis

## 3.1 System Functional Requirements

The requirement of the system must be fulfilled for the proper working of the system. Such requirements describe system behavior under specific conditions and include the product features and functions that web & app developers must add to the solution. Such requirements should be precise both for the development team and stakeholders.

### 3.1.1 Project Interactivity Plan

The website is user-friendly due to third-party control. Users can save records and the system is easy to use through a user through user-friendly GUI.

### 3.1.2 Signup Information

Users must provide information in the form of their First name, Last name, Email, Password, and Confirm password then click on the signup button to register him/her.

### 3.1.3 Login Information

The user must provide a username and password to login into the system. If the admin wants to delete the user, he/she can do it by clicking the delete button in the user list visible.

### 3.1.4 Lawyer Record

Admin can delete and approve or view information of lawyer through admin panel.

**3.1.4.1 Add Lawyer**

Lawyers can be added by entering complete description data from the keyboard into fields and clicking on the register button. The following fields are required:

* + - * + First Name
        + Last Name
        + Education
        + Specialization
        + Experience
        + License Number
        + Law Firm Name
        + Email
        + Password
        + Phone Number
        + Address
        + Description
        + Upload Picture

**3.1.4.2 Update Record**

If the lawyer is going to update himself/herself following fields are required:

* + - * + First Name
        + Last Name
        + Education
        + Experience
        + Law Firm
        + Password
        + Phone Number
        + Twitter
        + Facebook
        + Instagram

**3.1.4.3 Delete Record**

If Admin wants to delete the user record, Admin can do it by clicking the delete button.

* + - * + Lawyer ID

### 3.1.5 SRS Document (System Requirements Specification)

Table 1: Software Requirement Specification

|  |  |
| --- | --- |
| **Sr. No.** | **Description** |
| SRS-01 | The system should be able to provide a login facility |
| SRS-02 | The system should be able to provide Sign Up facility |
| SRS-03 | The system should be able to Register New Lawyer |
| SRS-04 | The system should be able to Update Lawyer details |
| SRS-05 | The system should be able to Delete Lawyer details |
| SRS-06 | The system should be able to hire a client as a lawyer |
| SRS-07 | The system should be able to categorize lawyers based on specialization. |
| SRS-08 | The system should be able to provide a testimonial facility to users. |

## 3.2 Non-Functional Requirements of the system

The non-functional requirements elaborate a performance characteristic of a system. Non-functional requirements specify the quality of a system and are mostly related to the satisfiability of the user.

* Ensure the privacy of customers.
* Exclude unauthorized access.
* Reliability
* Flexibility
* User Safety
* User Friendly
* Satisfiability of the user

## 3.3 Hardware & Software Requirements

### 3.3.1 Software Requirements

Table 2: Software Requirements

|  |  |
| --- | --- |
| Operating system | window 7/8/8.1/10/11 |
| Database | MySQLi |
| Web Browser | Google Chrome / Firefox / Opera Browser |
| Web page style sheet | HTML, CSS, Bootstrap, JavaScript, Ajax and JQuery |
| Program code | PHP, MySQLi |

### 3.3.2 Hardware Requirements

Table 3: Hardware Requirements

|  |  |
| --- | --- |
| Internet Require | Yes |
| Main Memory | Minimum 1GB |
| CPU speed | 2.6GHz |
| Monitor | EGA/SVGA (display),800X600 24 bits true color |
| Keyboard | Standard keyboard |
| Mouse | Standard Mouse / Touchpad |

## 3.4 Introduction of Tools

### 3.4.1 VS Code

Visual Studio Code is a shareware cross-platform source code editor with a Python application programming interface (API). The main aim of using it is that It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses. It is also easy to use and I have been working on it for many years. Visual Studio Code can be downloaded and evaluated for free. There is no enforced time limit for the evaluation.

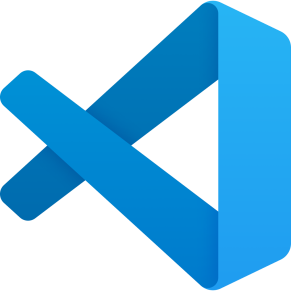


Figure 8: Visual Studio Code

### 3.4.2 HTML (Hyper Text Markup Language)

It is not possible to design or develop a web application without HTML. HTML is short for Hyper-Text Markup Language and is a language used to create electronic documents, especially pages on the World Wide Web that contains connections and hyperlinks to other pages.

HTML (Hypertext Markup Language) is the code that is used to structure a web page and its content. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables.



Figure 9: HTML (Hyper Text Markup Language)

### 3.4.3 CSS (Cascading Style Sheet)

With HTML we can only design structure or web page but with the help of CSS, we can style our web pages which attract the user. CSS stands for Cascading Style Sheets. It describes how HTML elements are to be displayed on the screen, on paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External style sheets are stored in CSS files. CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts and it allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language.



Figure 10: CSS (Cascading Style Sheet)

### 3.4.4 Bootstrap

Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing a responsive and mobile-first website. Bootstrap is a free collection of tools for creating websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation, and other interface components, as well as optional JavaScript extensions. Bootstrap is a framework to help you design websites faster and easier. It includes HTML and CSS-based design templates for typography, forms, Buttons, tables, navigation, modals, image carousels, etc. Here are some additional reasons to use Bootstrap: Bootstrap's responsive CSS adjusts to phones, tablets, and desktops.



Figure 11: Bootstrap

### 3.4.5 JavaScript

JavaScript is a text-based programming language used both on the client side and server side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user. JavaScript can calculate, manipulate and validate data.



Figure 12: JavaScript

### 3.4.6 JQuery

JQuery is a JavaScript Library. It greatly simplifies JavaScript programming. JQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. (jquery.com).The purpose of using jQuery is to make it much easier to use JavaScript on your website. JQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish and wraps them into methods that you can call with a single line of code.



Figure 13: jQuery

### 3.4.7 PHP

The purpose of using PHP is that it is easier than other server-side technologies, PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. PHP is a server-side scripting language that is used to develop Static websites Dynamic websites or Web applications. PHP stands for Hypertext Preprocessor, which earlier stood for Personal Home Pages. PHP scripts can only be interpreted on a server that has PHP installed.



Figure 14: PHP

### 3.4.8 AJAX

Changing data in select fields is not possible without AJAX. Update a web page without reloading the page. Ajax Request data from a server - after the page has loaded. It receives data from a server - after the page has loaded. It sends a data server - in the background.



Figure 15: AJAX

### 3.4.9 MySQL

* + - * + MySQL is a freely available open-source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing, and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use

1.15.10 Server-Side Programming: PHP will be used for server-side programming. The 'i' in **MySQLi** stands for Improved. Therefore, this is also known as the improved version of MySQL.

* + - * + MySQLi is an open-source relational database management system (RDBMS) in July 2013, it was the world's second most widely used RDBMS, and the most widely used open-source client-server RDBMS.
        + MySQL is the old database driver, and MySQLi is the improved driver. MySQLi can be done procedural and object-oriented whereas MySQL can only be used procedurally. MySQLi also supports prepared statements which protect from SQL Injection.



Figure 16: MySQL

# Chapter 4 System Design

## 4.1 Use case – fully dressed

A use case is a description of a system’s behavior from a user’s standpoint. For system developers, this is a valuable tool: requirements from a user’s point of view. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

### 4.1.1 Importance of Use Case Diagrams

As mentioned before use case diagrams are used to gather a usage requirement of a system. Depending on your requirement you can use that data in different ways. Below are a few ways to use them.

* + - * **To identify functions and how roles interact with them** – The primary purpose of using case diagrams.
      * **For a high-level view of the system** – Especially useful when presenting to managers or stakeholders. You can highlight the roles that interact with the system and the functionality provided by the system without going deep into the inner workings of the system.
      * **To identify internal and external factors** – This might sound simple but in large complex projects a system can be identified as an external role in another use case.

### 4.1.2 Use Case Diagram objects

Use case diagrams consisting of 4 objects.

* + - * Actor
      * Use case
      * System
      * Package

**4.1.2.1 Actor**

An actor in a UML Use Case Diagram is any entity (person, organization, or external system) that performs a role in one given system. In a use case diagram, an actor interacts with a use case.



Figure 17: Actor

**4.1.2.2 Use case**

A use case in a UML Use Case Diagram gives a visual representation of distinct business functionalities in a system.

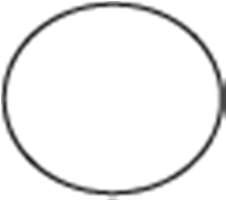


Figure 18: Use Case

**4.1.2.3 System**

A system in a UML Use Case Diagram is a rectangle spanning all the use cases in the system that defines the scope of your system. Anything within the box represents functionality that is in scope and anything outside is not. Note that the actors in the system are outside the system.



Figure 19: System

**4.1.2.4 Package**

A package object in a UML Class and Use Case Diagram provides the ability to group classes and/or interfaces that are either similar or related. Grouping these design elements in a package element provides for better readability of UML diagrams, especially complex diagrams.

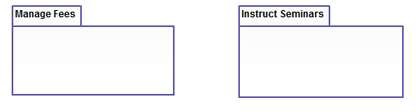


Figure 20: Package

**4.1.2.5 Relationship**

The relationship is an association between the use case and the actor.

There are several use cases relationships:

* Association
* Extend
* Generalization
* Include

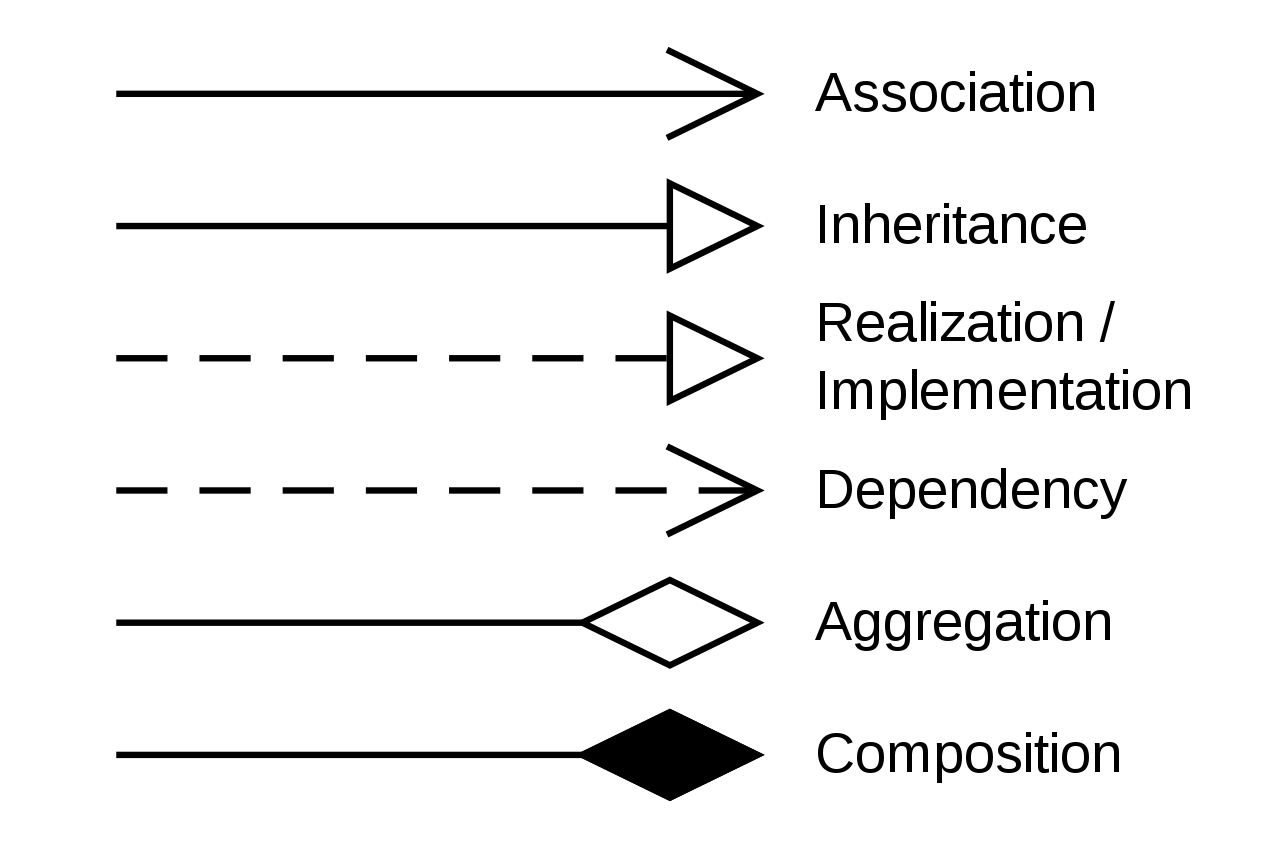


Figure 21: Relationships

### 4.1.3 Use Case Diagram

In the rare case, you are performing the use case analysis of some authentication or user management software, such as an SSO solution, in which the business value for the user is really to get logged into some protected systems and perform activities.

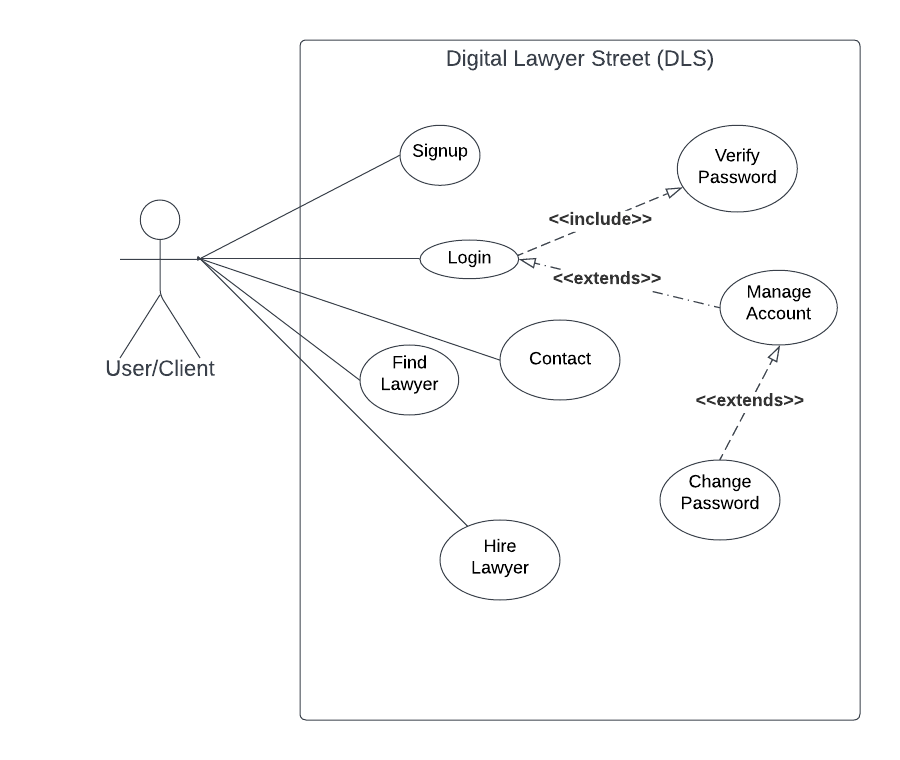


Figure 22: User/Client Use Case

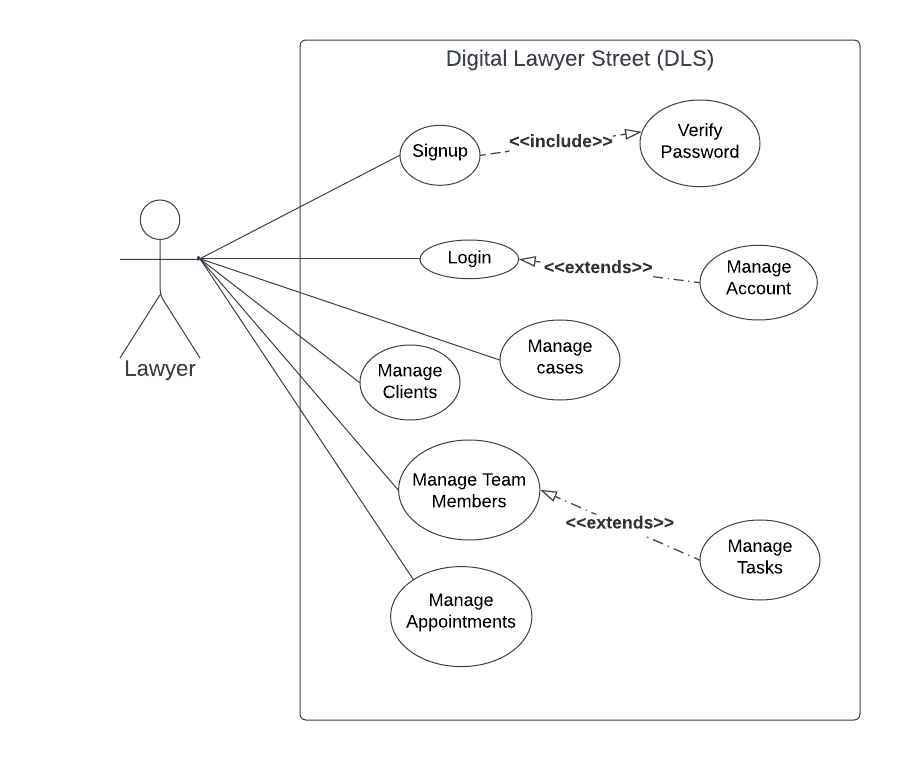


Figure 23: Lawyer Use Case

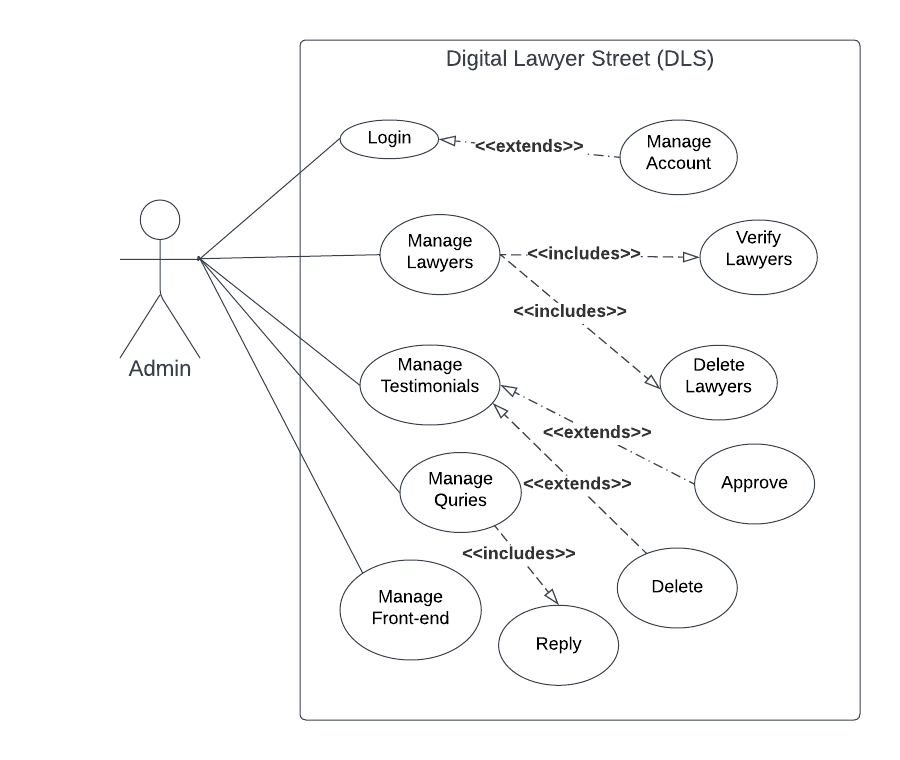


Figure 24: Admin Use Case

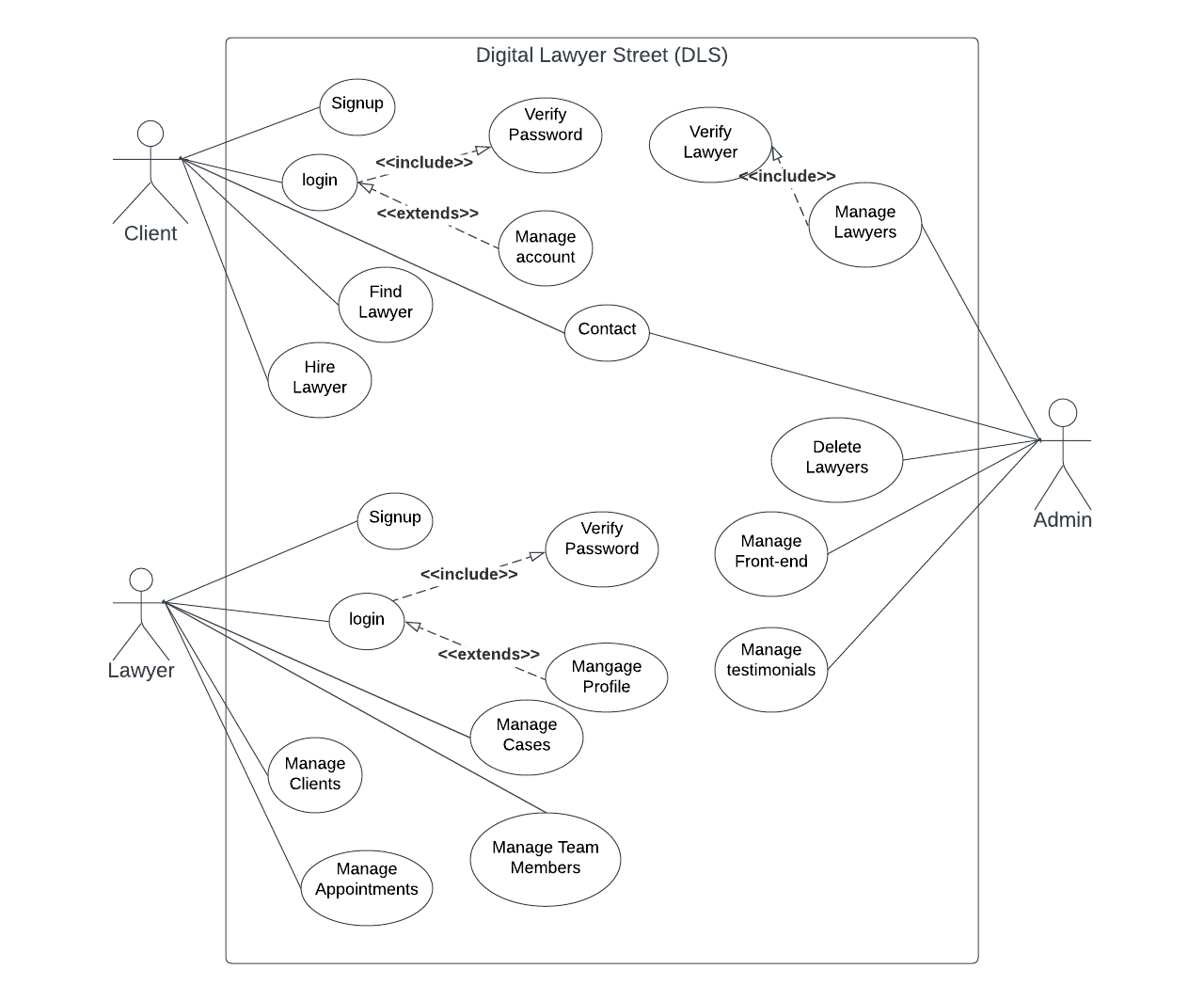


Figure 25: Use Case Diagram Fully Dressed

**Pre-Conditions**

* + - * + The user must be logged in to hire a lawyer.
        + However, any user can find a lawyer even without a login.

**Primary Scenario**

* The user will open the lawyer menu item from the Main Menu.
* The user will click on the desired lawyer profile or view profile button to view lawyer details.
* The information about the specific lawyer is loaded.
* Users can check different lawyers’ details in the same process.
* Users can hire a lawyer.
* Users can contact a hired lawyer on any media platform

**Secondary Scenario**

* Users can cancel the hiring request at any time.

**Post Condition**

* The user can fix an appointment with a lawyer now.

## 4.2 WBS fully dressed

* + - Work breakdown structure allows you to decompose a project into small manageable sections, also known as deliverables. Clearly defined deliverables are easy to assign, allow you to accurately estimate needed time and resources, and track their completion.
    - Work breakdown structure (WBS) in project management is a method for completing a complex, multi-step **project**. It's a way to divide and conquer large projects to get things done faster and more efficiently. The goal of a WBS is to make a large project more manageable.

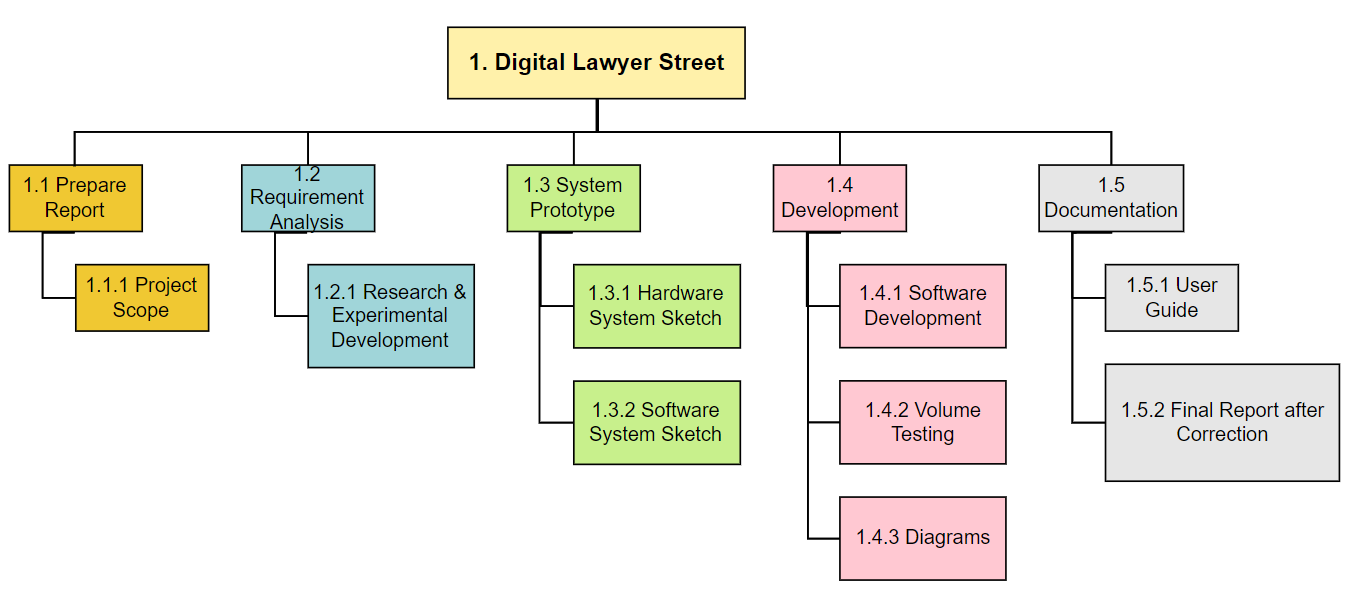


Figure 26: WBS (Work Breakdown Structure)

Table 4: WBS Levels

|  |  |  |
| --- | --- | --- |
| **Level 1** | **Level 2** | **Level 3** |
| 1. Digital Lawyer Street | 1.1 Prepare Report | 1.1.1 Project Scope |
| 1.2 Requirement Analysis | 1.2.1 Research & Experimental Development |
| 1.3 System Prototype | 1.3.1 Hardware & System Sketch 1.3.2 Software System Sketch |
| 1.4 Development | 1.4.1 Software Development 1.4.2 Volume Testing 1.4.3 Diagrams |
| 1.5 Documentation | 1.5.1 User Guide 1.5.2 Final Report after correction |

## 4.3 System Sequence diagram

* + - Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of collaboration.
    - Sequence Diagrams are time focused and they show the order of the interaction visually by using the vertical axis of the diagram to represent the time what messages are sent and when.
    - A sequence diagram simply depicts the interaction between objects in sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram.
    - Sequence diagrams describe how and in what order the objects in a system function.

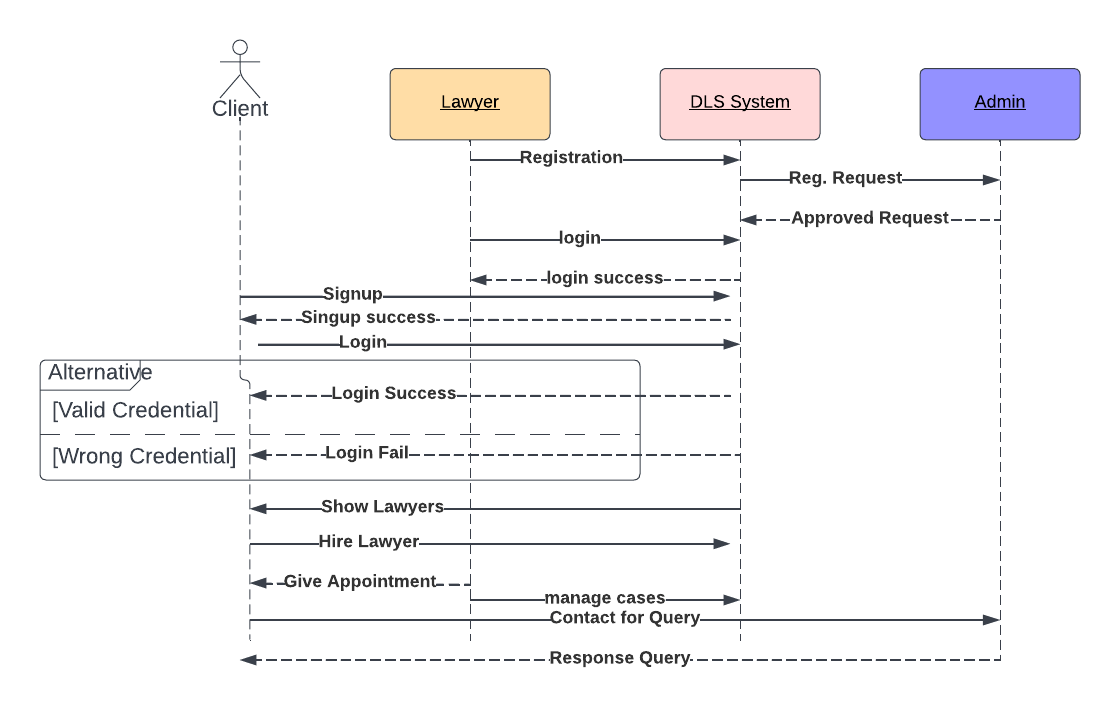


Figure 27: Sequence Diagram

## 4.4 Class diagram

* + - A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.
    - The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. The class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints.
    - Class diagrams are the best way to illustrate a system's structure in a detailed way, showing its attributes, operations as well as inter-relationships.

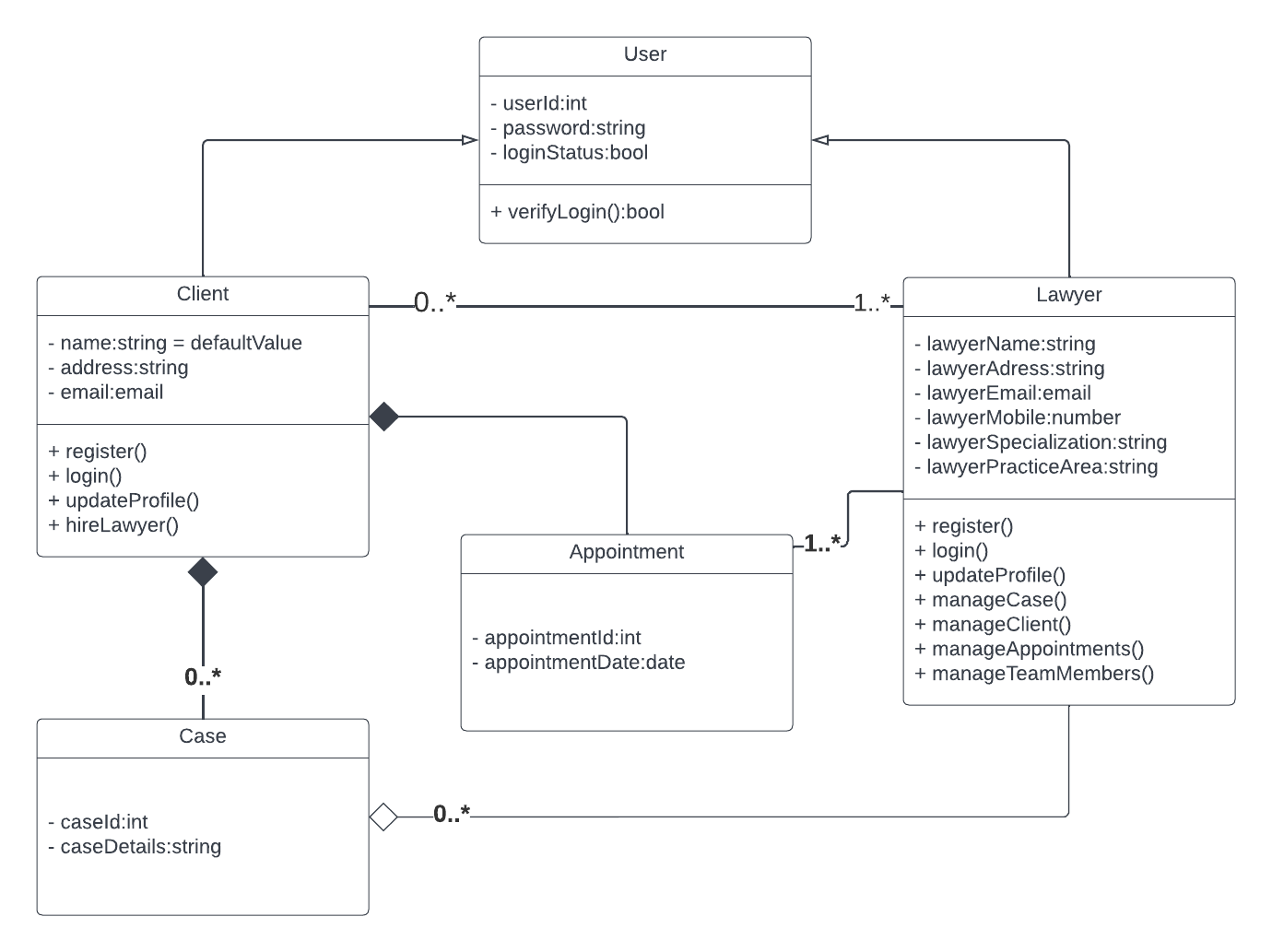


Figure 28: Class Diagram

## 4.5 ER (Entity Relationship Model) Model

* + - **Entity Relationship Model** (ER Modeling) is a graphical approach to database design. It is a high-level data model that defines data elements and their relationship to a specified software system. An ER model is used to represent real-world objects. An Entity is a thing or object in the real world that is distinguishable from the surrounding environment.
    - The three main components of the ER Model are entities, attributes, and relationships. In ERM terms, an entity is a "thing" within the organization that we want to keep information about, such as a customer, employee, or course.

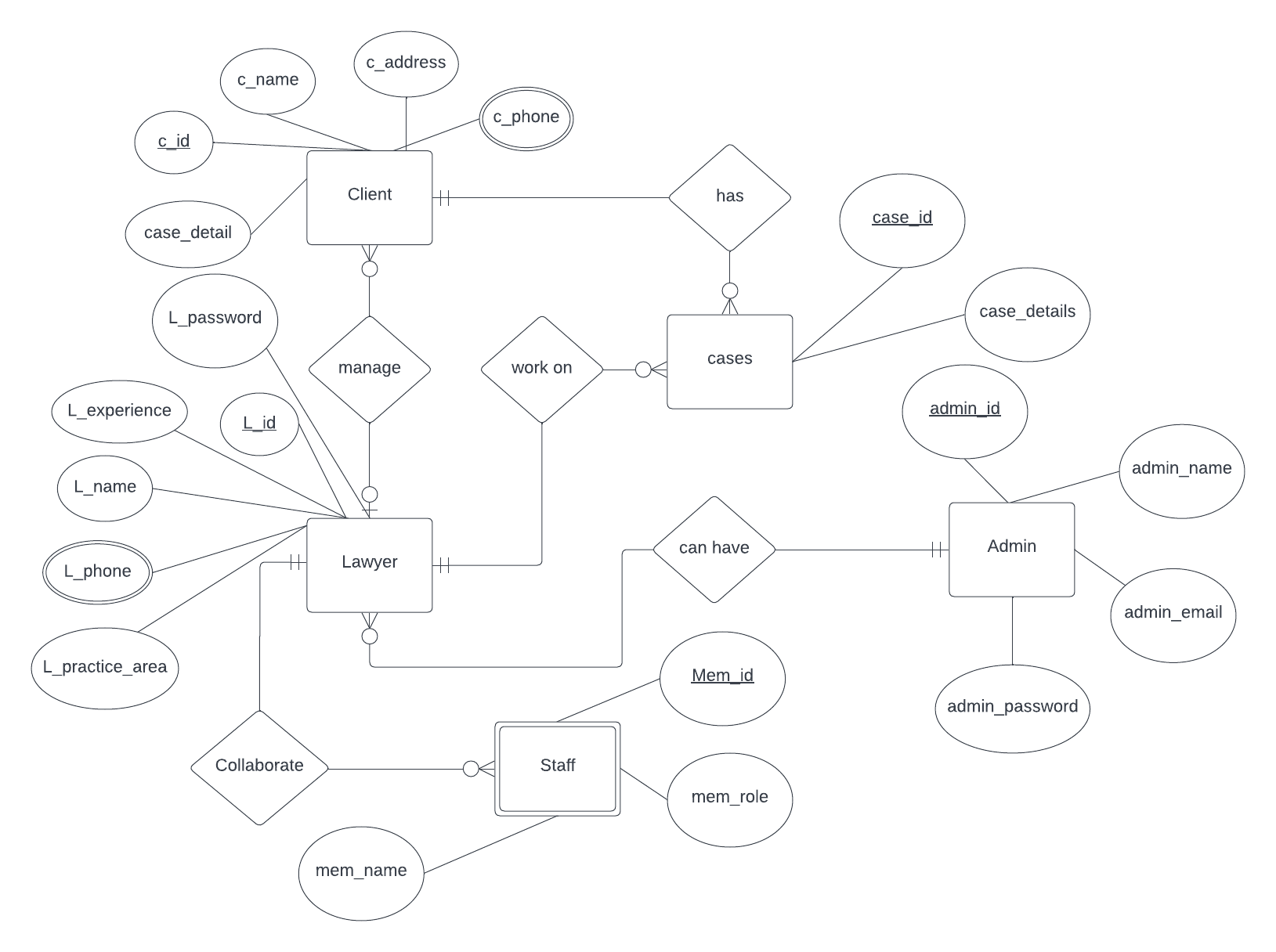


Figure 29: ER (Entity Relationship Diagram)

### 4.6 Data Flow Diagram

* + - Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and report generation. Data flow diagrams can be divided into logical and physical.
    - There are two types of DFD:

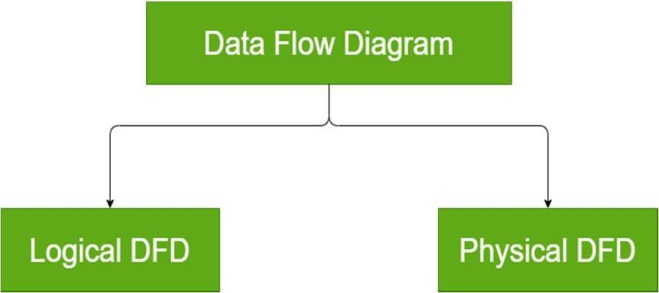


Figure 30: DFD Types

### 4.6.1 Symbols and Notations Used in DFDs

Two common systems of symbols are named after their creators:

* + - * Yourdon and Coad
      * Yourdon and DeMarco
      * Gane and Sarson

### 4.6.2 Data Flow Diagram of Digital Lawyer Street (DLS)

**Context level DFD – 0 Level DFD**

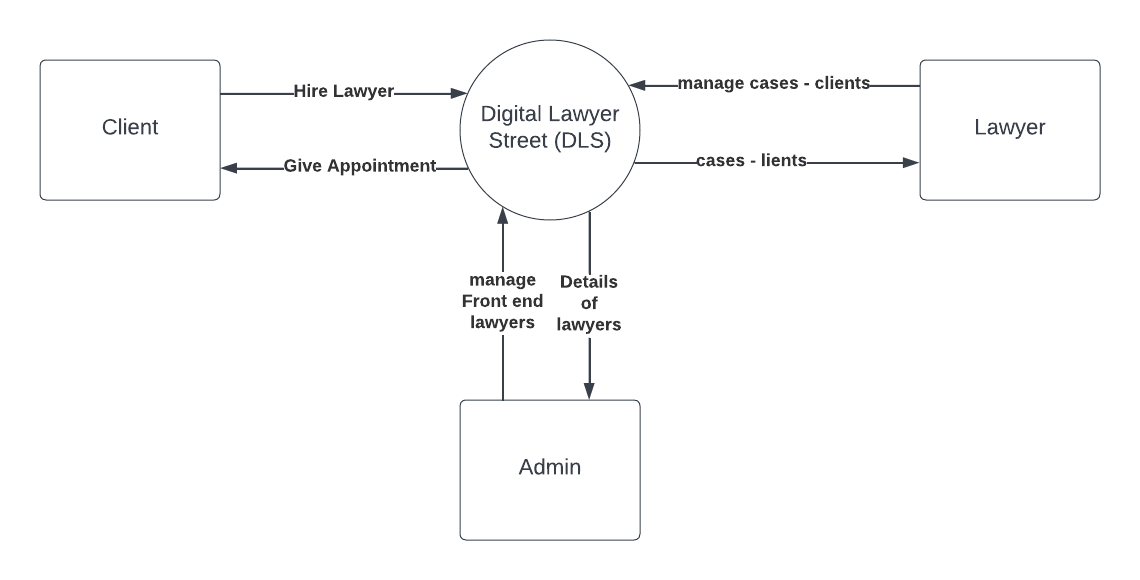
****

Figure 31: Context Level DFD

**LEVEL 1 DFD**

Level 1 DFD contains more details than level 0 DFD but less information than Level 2 DFD. It describes the flow more deeply.

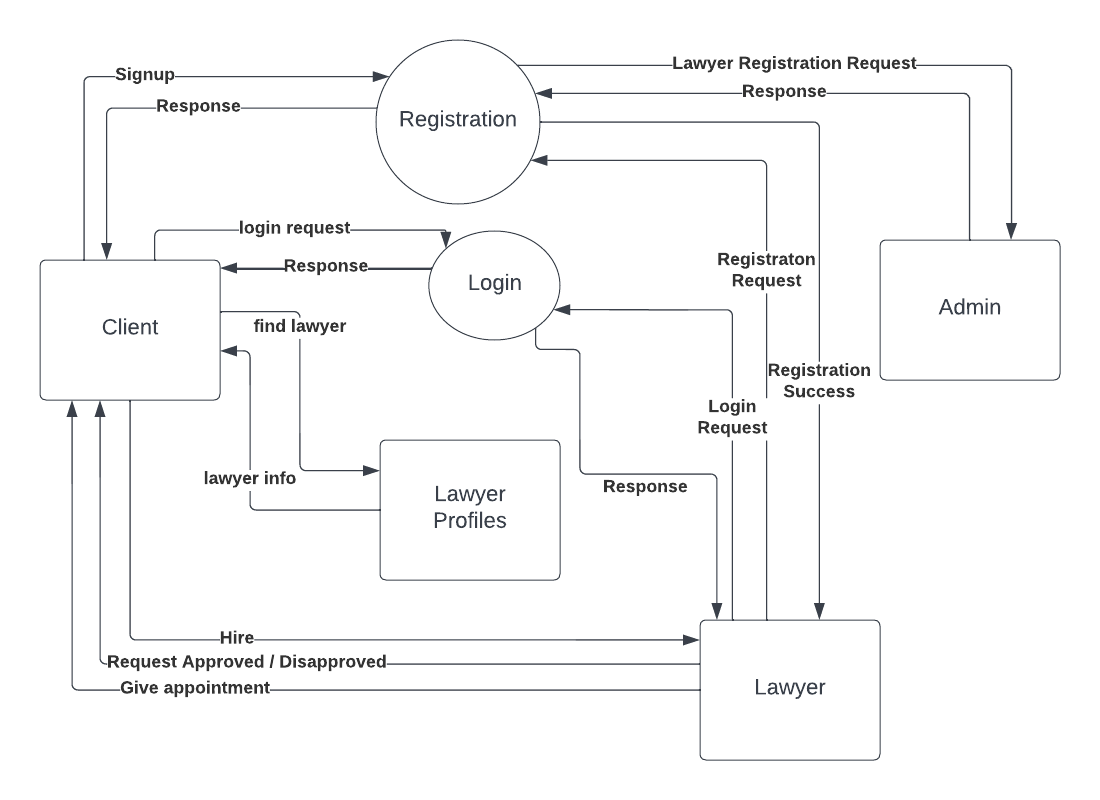


Figure 32: Level 1 DFD

**LEVEL 2 DFD**

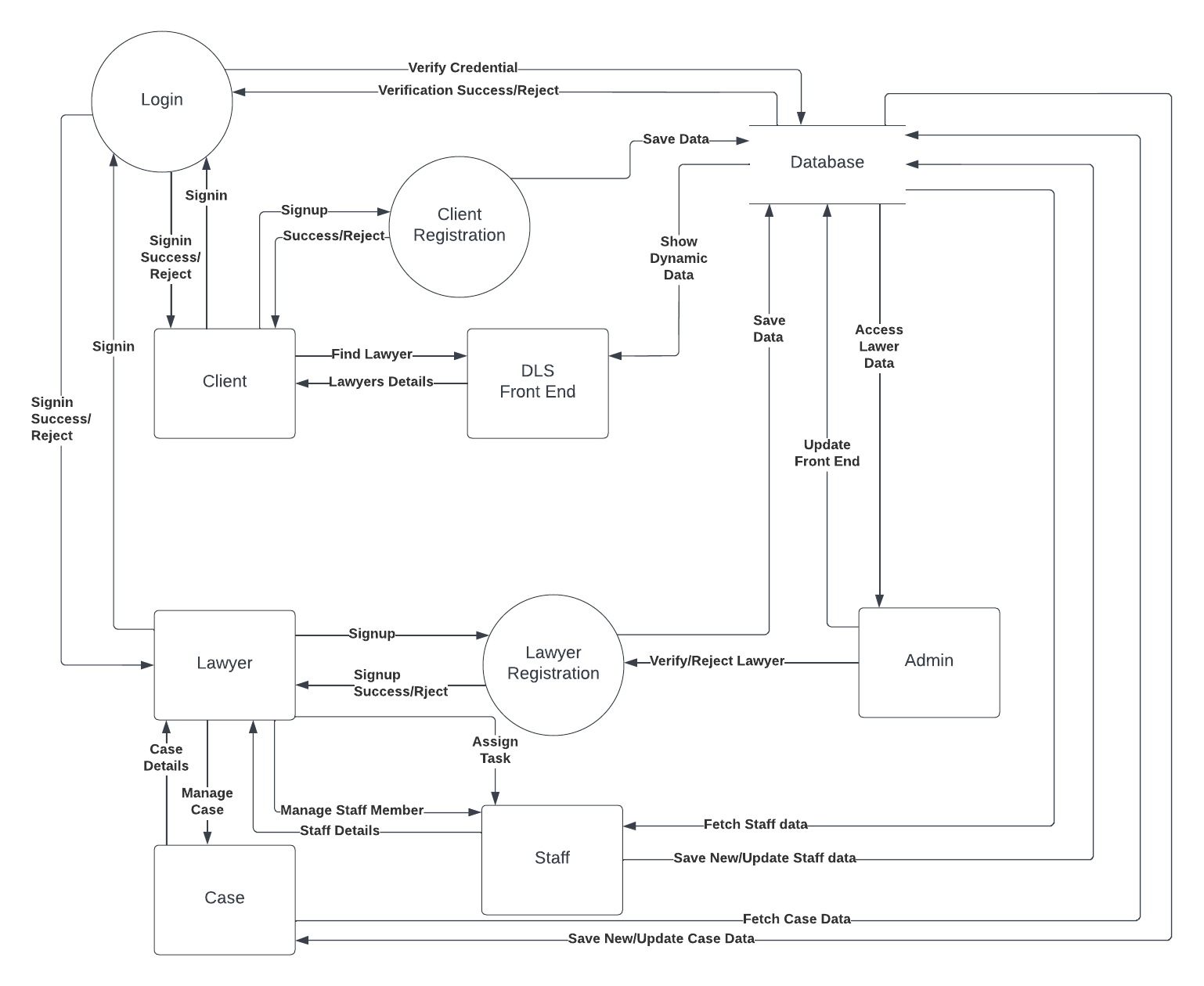
****

Figure 33: Level 2 DFD

# Chapter 5 Testing & Implementation

## 5.1 Testing Methods

System Testing methods are the various strategies or approaches used to test an application to ensure that it behaves and looks as expected. These encompass everything from front to backend testing, including unit and system testing.

### 5.1.1 Black Box Testing

Black box testing is a method of software testing that examines the functionality of an application without peering into its internal structure or working. This is also called behavioral testing. This method of test can be applied virtually to every level of software testing: unit, integration, system, and acceptance.

While testing we didn’t take care of the source code of the system, our primary focus is testing system functions and non-functional functionality.

* Mission or mistaken functions
* Inappropriate errors
* Flaws in data structures or exterior database access
* Behavior or performance errors
* Initialization and termination errors

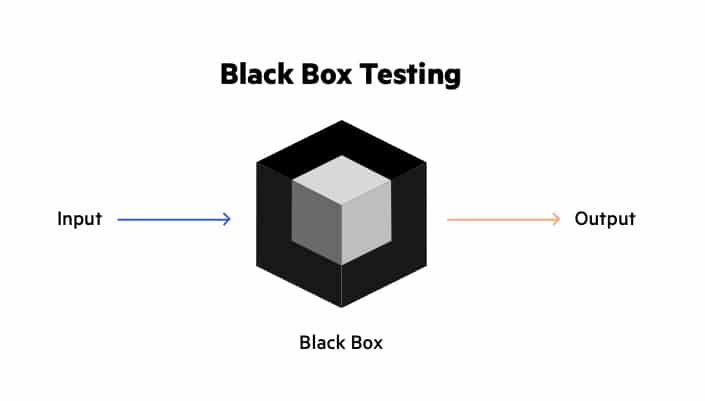


Figure 34: Black-box Testing

### 5.1.2 White Box Testing

This is also known as Glass Box, Clear Box, and Open Box testing. White Box testing is a method of software testing that tests the internal structures or workings of an application, as opposed to its functionality. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases.

The white box testing method applies to the following levels

* Unit Testing (For testing path within the unit)
* Integration Testing (For Testing paths between paths)
* System Testing (Testing Subsystem)

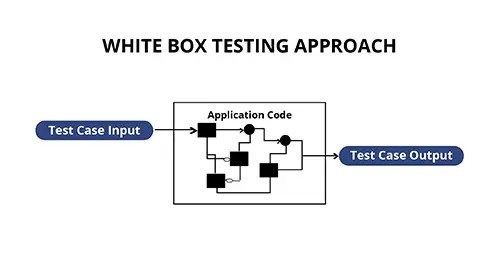


Figure 35: White Box Testing

### 5.1.2 Grey Box Testing

Gray box testing is a software testing method that is a combination of the black box testing method and white box testing method.

Gray box testing is a technique to test the software product or application with partial knowledge of the internal workings of an application. In this process, context-specific errors that are related to web systems are commonly identified. It will increase the testing coverage by concentrating on all of the layers of any complex system.

Gray box testing is a software testing method, which is a combination of both white box testing and the black box testing method.

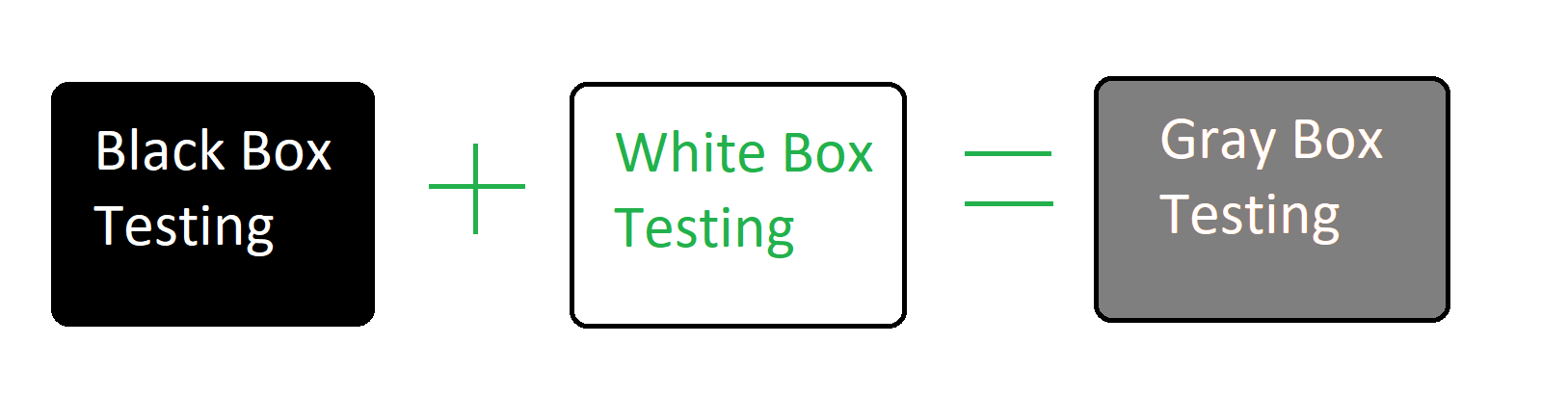


Figure 36: Grey Box Testing

## 5.2 Functional vs. non-functional testing

The goal of utilizing numerous testing methodologies in your development process is to make sure your software can successfully operate in multiple environments and across different platforms. These can typically be broken down between functional and non-functional testing. Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using use cases provided by the design team or business analyst. These testing methods are usually conducted in order and include:

* + - Unit testing
    - Integration testing
    - System testing
    - Acceptance testing

Non-functional testing methods incorporate all test types focused on the operational aspects of a piece of software. These include:

* + - Performance testing
    - Security testing
    - Usability testing
    - Compatibility testing

The key to releasing high-quality software that can be easily adopted by your end users is to build a robust testing framework that implements both functional and non-functional software testing methodologies.

### 

## 5.3 Functional Testing

Functional Testing consists of the following testing techniques.

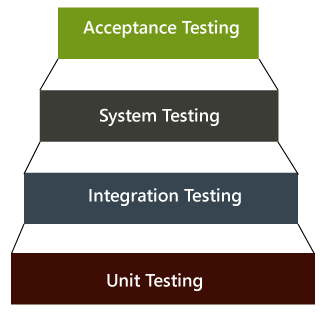


Figure 37: Functional Testing

### 5.3.1 Unit Testing

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Unit testing can be conducted manually, but automating the process will speed up delivery cycles and expand test coverage.

Developers in a test-driven environment will typically write and run the tests before the software or feature is passed over to the test team. Unit testing can be conducted manually, but automating the process will speed up delivery cycles and expand test coverage.

Unit testing will also make debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process Test Left is a tool that allows advanced testers and developers to shift left with the fastest test automation tool embedded in any IDE.

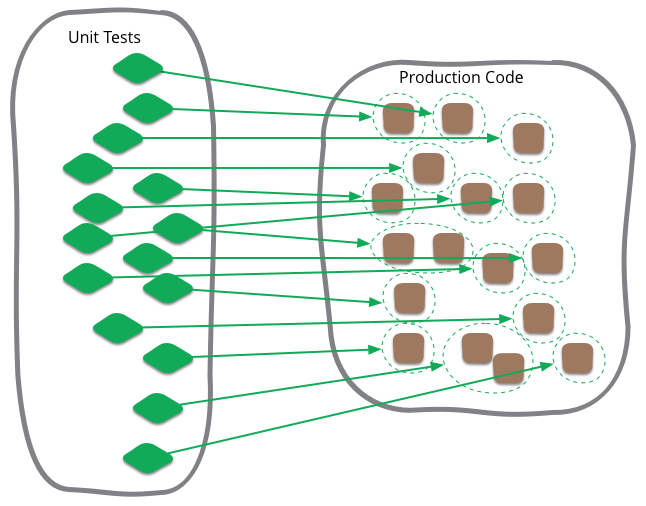


Figure 38: Unit Testing

### 5.3.2 Integration Testing

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as a group through integration testing to ensure whole segments of an application behave as expected (the interactions between units are seamless). These tests are often framed by user Scenarios, such as logging into an application or opening files. Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests.

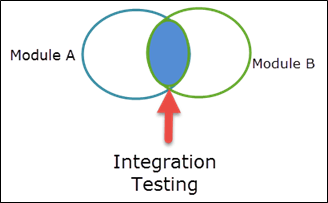


Figure 39: Integration Testing

### 5.3.3 System Testing

System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of the software is tested from end to end and is typically conducted by a separate testing team than the development team before the product is pushed into production.

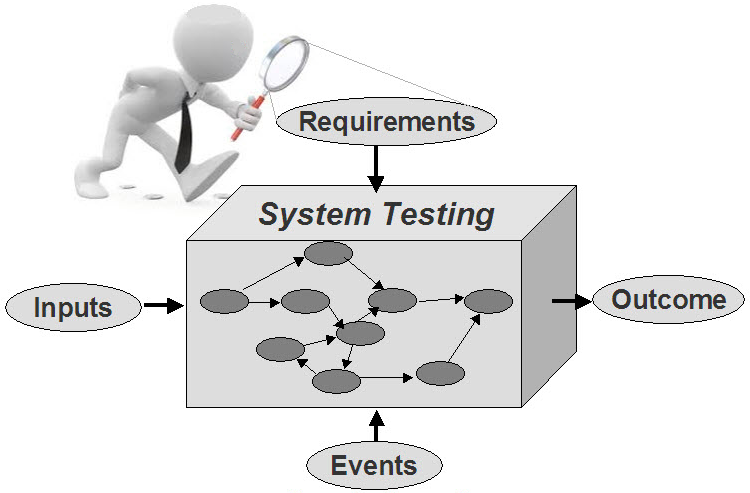


Figure 40: System Testing

### 5.3.4 Acceptance Testing

Acceptance testing is the last phase of functional testing and is used to assess whether or the final piece of software is ready for delivery. This requires the product to be tested both internally and externally, meaning you’ll need to get it into the hands of your end users for beta testing along with those of your QA team. **Acceptance tests** are carried out upon reception of new devices or equipment. The **purpose of acceptance testing** is to verify that the correct devices have been delivered according to the contract conditions and technical specifications, and in some cases, whether the equipment is properly installed. This requires the product to be tested both internally and externally, meaning you’ll need to get it into the hands of your end users for beta testing along with those of your QA team.

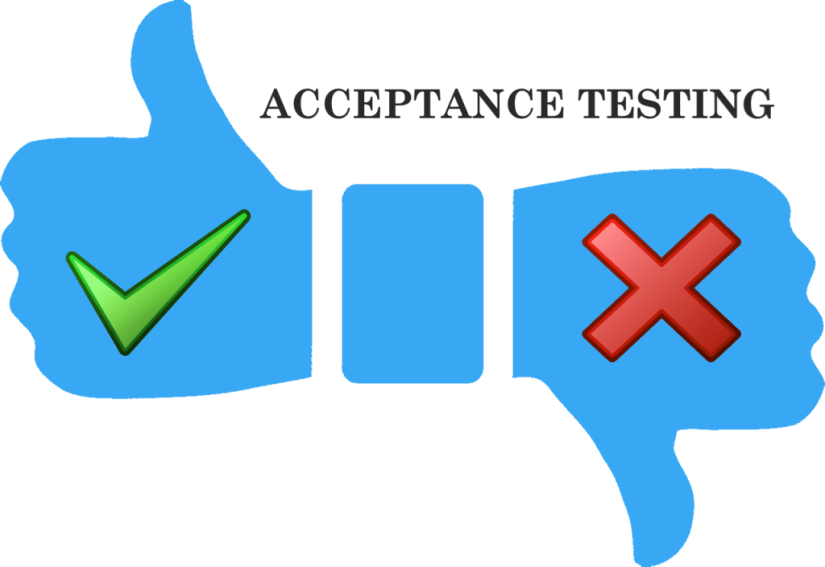


Figure 41: Acceptance Testing

## 5.4 Non- functional Testing

Performance testing is a non-functional testing technique used to determine how an application will behave under various conditions. The goal is to test its responsiveness and stability in real user situations. It is designed to test the readiness of a system as per non-functional parameters which are never addressed by functional testing. An excellent example of a non-functional test would be to check how many people can simultaneously login into software. The goal is to test its responsiveness and stability in real user situations. Performance testing can be broken down into four types:

### 5.4.1 Load testing

Load testing is the process of putting increasing amounts of simulated demand on your software, application, or website to verify whether or not it can handle what it’s designed to handle. When the load is increased beyond normal usage patterns, testingthe system's performance at exceptionally high or peak loads, is known as stress testing. Load testing is performed to find out the upper limit of the system or application.



Figure 42: Load Testing

### 5.4.2 Stress testing

Stress testing takes this a step further and is used to gauge how your software will respond at or beyond its peak load. The goal of stress testing is to overload the application on purpose until it breaks by applying both realistic and unrealistic load scenarios. With stress testing, you’ll be able to find the failure point of your piece of software. **Stress testing** refers to a type of **testing** that is so harsh, that it is expected to push the program to failure.

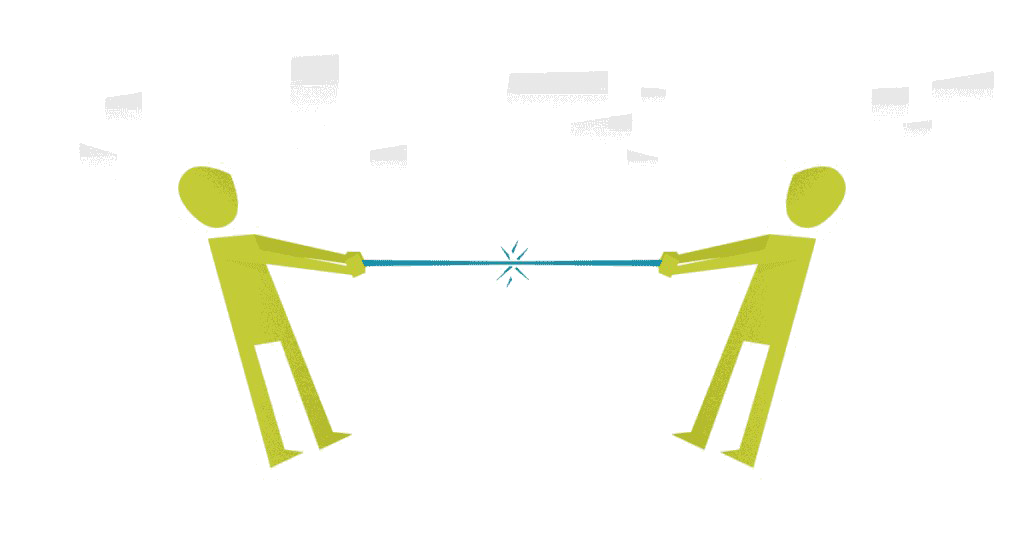


Figure 43: Stress Testing

### 5.4.3 Endurance testing

Endurance testing**,** also known as soak testing, is used to analyze the behavior of an application under a specific amount of simulated load over longer amounts of time. The goal is to understand how your system will behave under sustained use, making it a longer process than load or stress testing (which are designed to end after a few hours). A critical piece of endurance testing is that it helps uncover memory leaks.

### 5.4.4 Spike testing

Spike testing is a type of load test used to determine how your software will respond to substantially larger bursts of the concurrent user or system activity over varying amounts of time. The goal is to purposefully find loopholes and security risks in the system that could result in unauthorized access to or the loss of information by probing the application for weaknesses. There are multiple types of this testing method, each of which is aimed at verifying six basic principles of security:

* + - * + Integrity
        + Confidentiality
        + Authentication
        + Authorization
        + Availability
        + Non-repudiation

### 5.4.5 Usability testing

Usability testing is a testing method that measures an application’s ease of use from the end-user perspective and is often performed during the system or acceptance testing stages. The goal is to determine whether or not the visible design and aesthetics of an application meet the intended workflow for various processes, such as logging into an application.

### 5.4.6 Compatibility testing

Compatibility testing is used to gauge how an application or piece of software will work in different environments. It is used to check that your product is compatible with multiple operating systems, platforms, browsers, or resolution configurations. The goal is to ensure that your software’s functionality is consistently supported across any environment you expect your end users to be using.

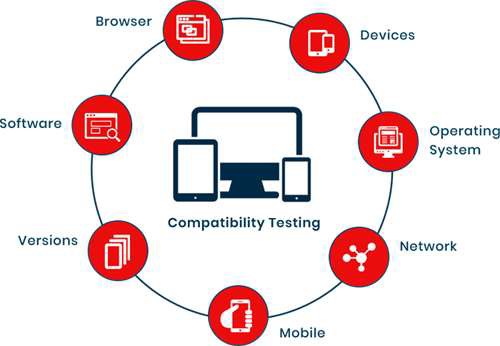


Figure 44: Compatibility Testing

## 

## 5.5 Test Cases

A test case is the set of steps that need to be done to test a specific function of the software. They are developed for various scenarios so that testers can determine whether the software is working the way it should and producing the expected results.

Table 5: Test Case 1

|  |
| --- |
| **Test Case ID**: **Test 1** **Test Engineer: Ghulam Mustafa** |
| **Objective**: Verify that system initialize successfully. **Product**: Digital Lawyer Street **Environment**: Web Application **Pre-Requisite**: None |
| **Methods**: App Initialization |
| **Comments**: Passed |

Table 6: Test Case 2

|  |
| --- |
| **Test Case ID**: **Test 2** **Test Engineer: Ghulam Mustafa** |
| **Objective**: Verify that system runs successfully **Product**: Digital Lawyer Street **Environment**: Web Application **Pre-Requisite**: System Initialize |
| **Methods**: Launch The Application  Insert data |
| **Comments**: Passed |

Table 7: Test Case 3

|  |
| --- |
| **Test Case ID**: **Test 3** **Test Engineer: Ghulam Mustafa** |
| **Objective**: Verify that Admin / Lawyer / User login successfully **Product**: Digital Lawyer Street **Environment**: Web Application **Pre-Requisite**: Login page display |
| **Methods**: Launch the application  Enter Login Credentials |
| **Comments**: Passed |

Table 8: Test Case 4

|  |
| --- |
| **Test Case ID**: **Test 4** **Test Engineer: Ghulam Mustafa** |
| **Objective**: Verify that data is inserted  **Product**: Digital Lawyer Street **Environment**: Web Application **Pre-Requisite**: Input Forms |
| **Methods**: Open Specific Web Page  Insert data in databases |
| **Comments**: Passed |

Table 9: Test Case 5

|  |
| --- |
| **Test Case ID**: **Test 5** **Test Engineer: Ghulam Mustafa** |
| **Objective**: Verify that data is retrieved  **Product**: Digital Lawyer Street **Environment**: Web Application **Pre-Requisite**: database must have data |
| **Methods**: Open Specific Web Page  Retrieve data in the table |
| **Comments**: Passed |

Table 10: Test Case 6

|  |
| --- |
| **Test Case ID**: **Test 6** **Test Engineer: Ghulam Mustafa** |
| **Objective**: Verify that data is updated  **Product**: Digital Lawyer Street **Environment**: Web Application **Pre-Requisite**: database must have data |
| **Methods**: Open Specific Web Page  Update data using forms |
| **Comments**: Passed |

Table 11: Test Case 7

|  |
| --- |
| **Test Case ID**: **Test 7** **Test Engineer: Ghulam Mustafa** |
| **Objective**: Approve new lawyer by admin  **Product**: Digital Lawyer Street **Environment**: Web Application **Pre-Requisite**: New lawyer signup |
| **Methods**: Open Lawyers Requests Web Page  Approve lawyer |
| **Comments**: Passed |

Table 12: Test Case 8

|  |
| --- |
| **Test Case ID**: **Test 8** **Test Engineer: Ghulam Mustafa** |
| **Objective**: Delete new lawyer by admin  **Product**: Digital Lawyer Street **Environment**: Web Application **Pre-Requisite**: New lawyer signup |
| **Methods**: Open Lawyers Requests Web Page  Delete new lawyer by admin |
| **Comments**: Passed |

Table 13: Test Case 9

|  |
| --- |
| **Test Case ID**: **Test 9** **Test Engineer: Ghulam Mustafa** |
| **Objective**: Disapprove lawyer by admin  **Product**: Digital Lawyer Street **Environment**: Web Application **Pre-Requisite**: Lawyer must be registered |
| **Methods**: Open Lawyers Web Page  Disapprove lawyer by admin |
| **Comments**: Passed |

Table 14: Test Case 10

|  |
| --- |
| **Test Case ID**: **Test 10** **Test Engineer: Ghulam Mustafa** |
| **Objective**: Approve user request for hiring **Product**: Digital Lawyer Street **Environment**: Web Application **Pre-Requisite**: Lawyer must be login |
| **Methods**: Open Client Request Web Page  Approve user by a lawyer |
| **Comments**: Passed |

Table 15: Test Case 11

|  |
| --- |
| **Test Case ID**: **Test 11** **Test Engineer: Ghulam Mustafa** |
| **Objective**: Do Whatsapp and Email API work or not **Product**: Digital Lawyer Street **Environment**: Web Application **Pre-Requisite**: Admin/Lawyer/Client must be login |
| **Methods**: Open Web Page contain Chat option  Approve user by the lawyer |
| **Comments**: Passed |

Table 16: Test Case 12

|  |
| --- |
| **Test Case ID**: **Test 12** **Test Engineer: Ghulam Mustafa** |
| **Objective**: Whether Services provided by Admin or Lawyer correct or not **Product**: Digital Lawyer Street **Environment**: Web Application **Pre-Requisite**: None |
| **Methods**: Interface is checked one by one. |
| **Comments**: Passed |

Table 17: Test Case 13

|  |
| --- |
| **Test Case ID**: **Test 13** **Test Engineer: Ghulam Mustafa** |
| **Objective**: Security of DLS **Product**: Digital Lawyer Street **Environment**: Web Application **Pre-Requisite**: None |
| **Methods**: Checking whether is it possible to login into any other account with someone else credentials. |
| **Comments**: Passed |

Table 18: Test Case 14

|  |
| --- |
| **Test Case ID**: **Test 14** **Test Engineer: Ghulam Mustafa** |
| **Objective**: DLS can handle multiple requests **Product**: Digital Lawyer Street **Environment**: Web Application **Pre-Requisite**: None |
| **Methods**: Checking whether the system is crashed or not when multiple requests are at the same time. |
| **Comments**: Passed |

# Chapter 6 User Manual

## 6.1 Main/Home view

A home page is the main web page of a website. The term also refers to one or more pages always shown in a web browser, when the application starts up. In this way, it is also known as the start page.

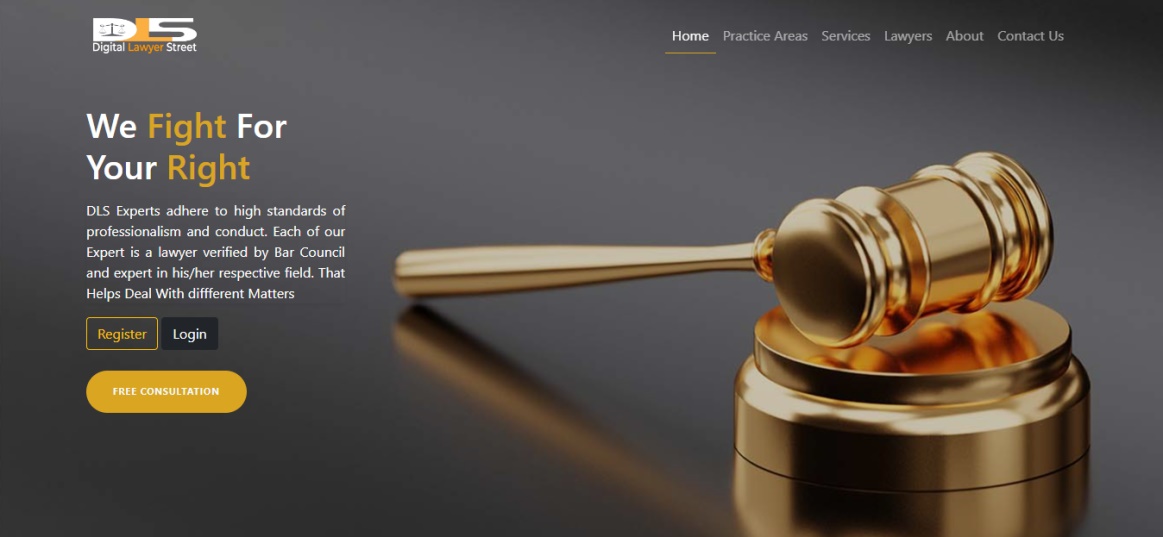


Figure 45: Home Screen

## 6.2 Our Lawyers Page



Figure 46: Lawyers Page

## 6.3 Services Page

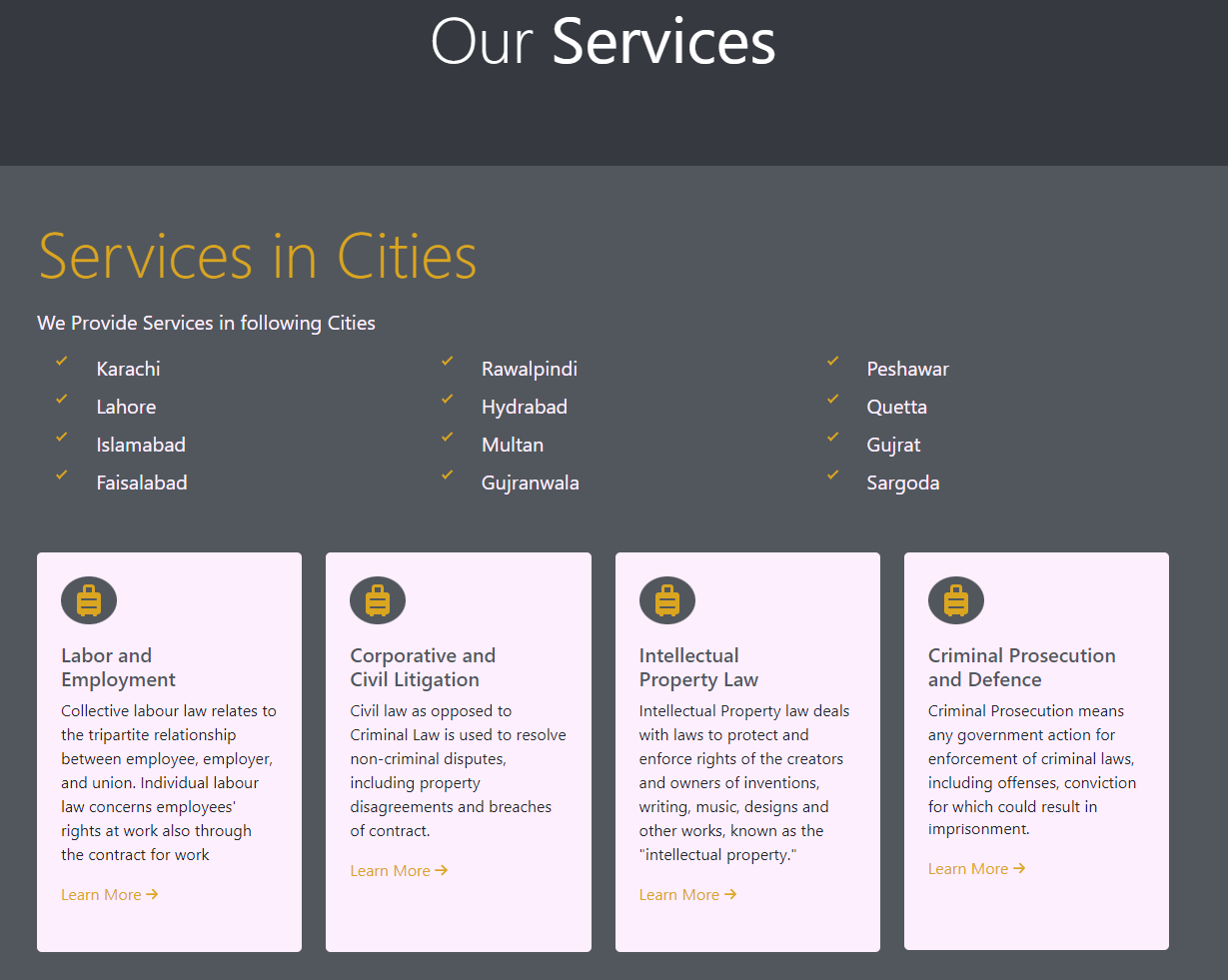


Figure 47: Services

## 6.4 Testimonial Page

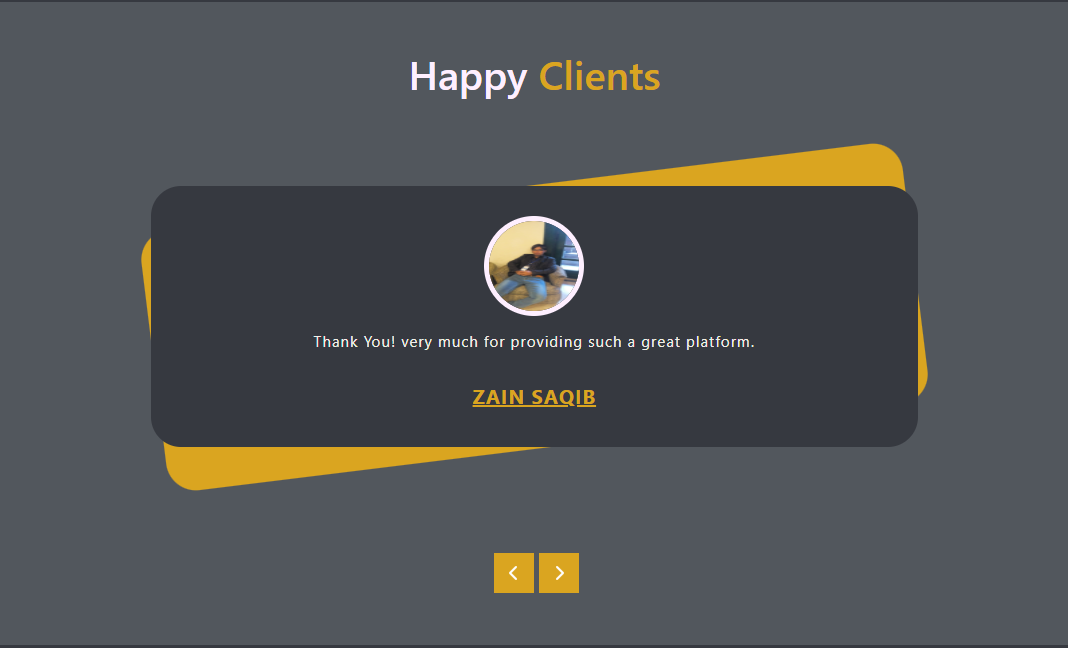


Figure 48: Testimonials

## 6.5 About Us Page



Figure 49: About Us

## 6.6 Contact Us Page

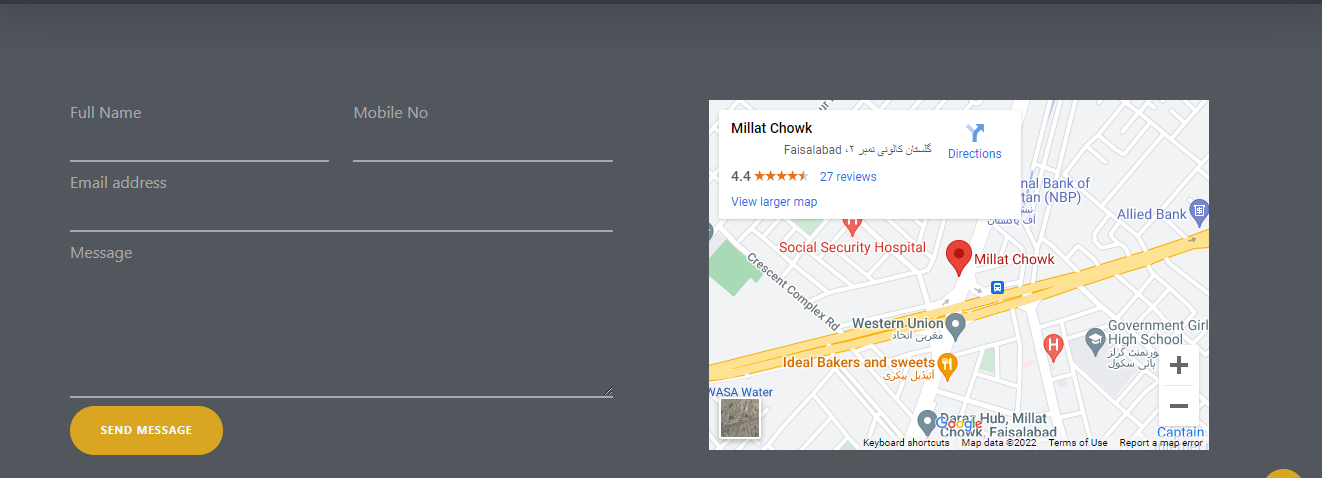


Figure 50: Contact Us

## 6.7 Footer

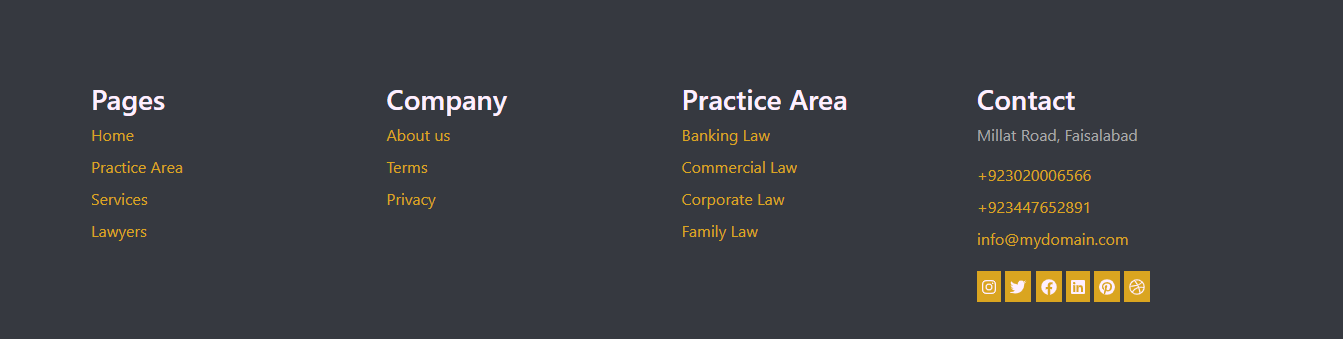


Figure 51: Footer

## 6.8 DLS Statistics



Figure 52: DLS Statistics

## 6.9 Forget Password

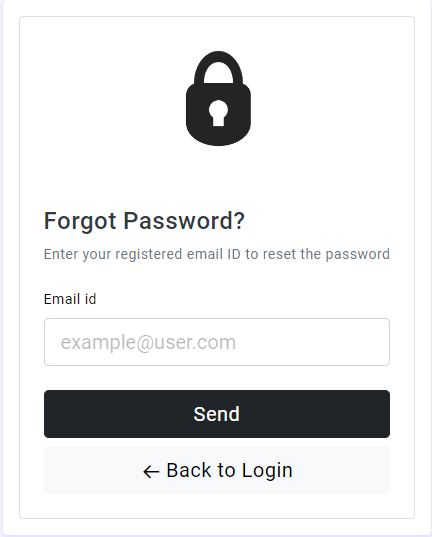


Figure 53: Forget Password

## 6.10 Lawyer Registration Page

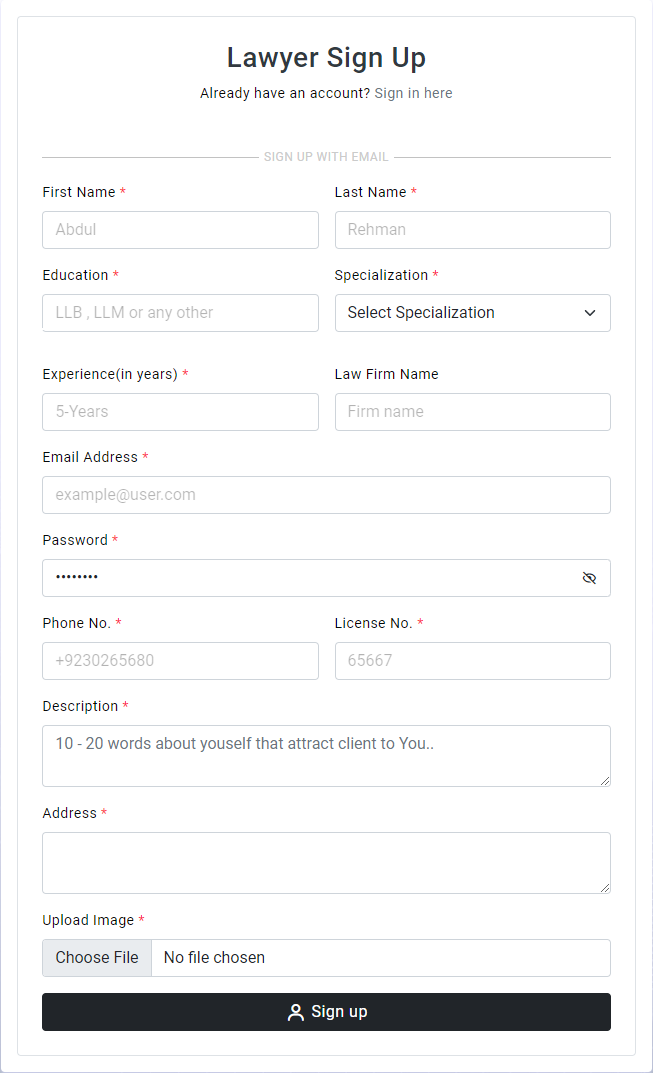


Figure 54: Lawyer Registration

## 6.11 User Login

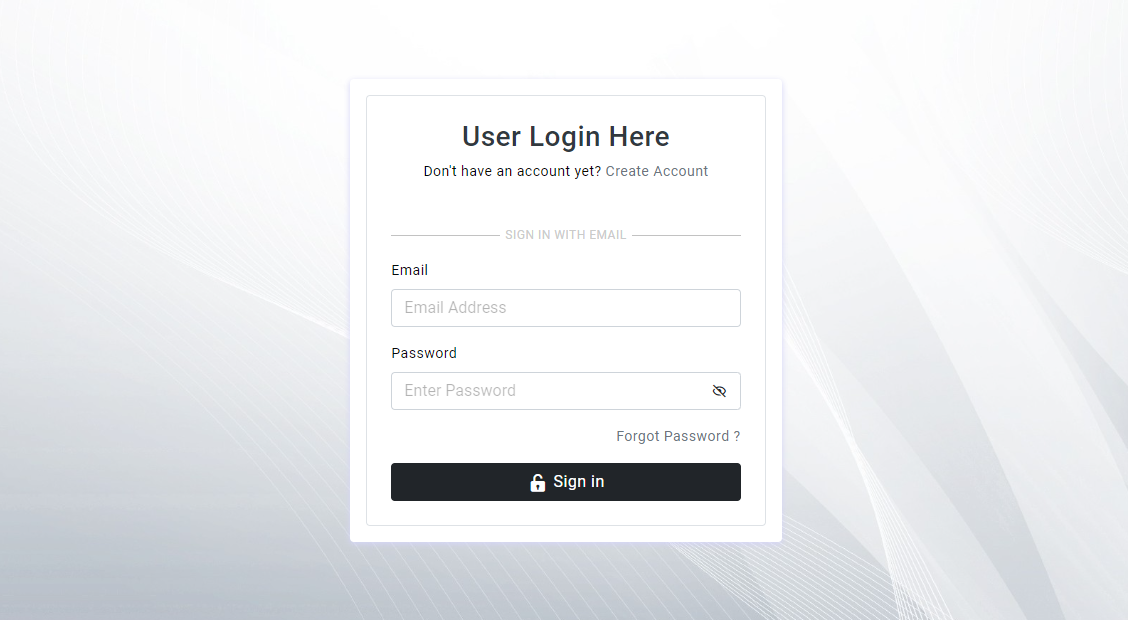


Figure 55: Login Area

## 6.12 User Dashboard

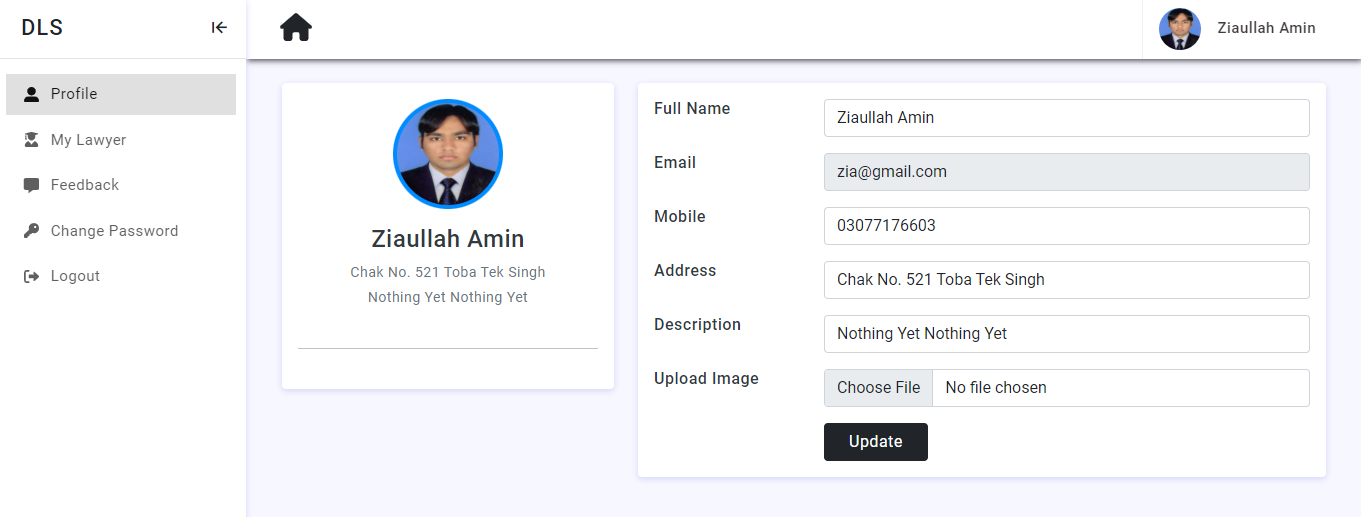


Figure 56: User Dashboard

## 6.13 Lawyer Dashboard

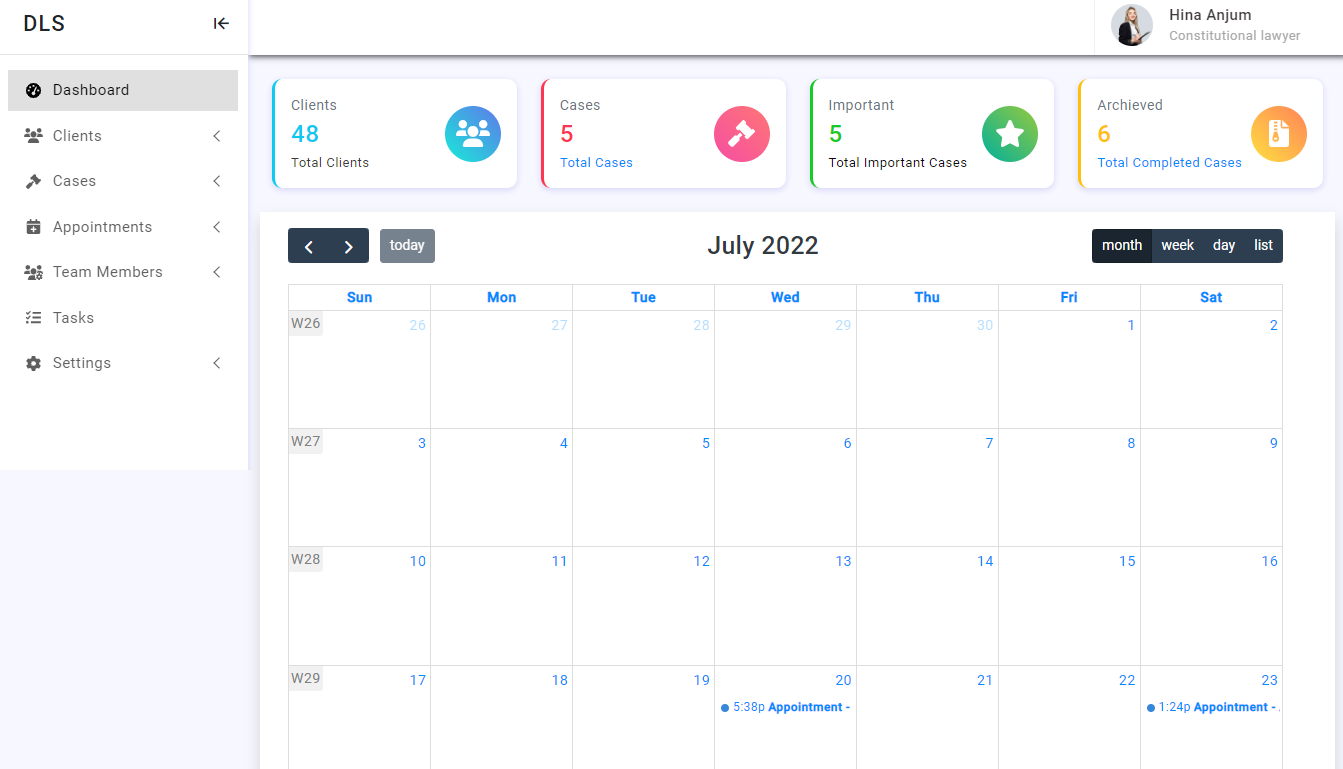
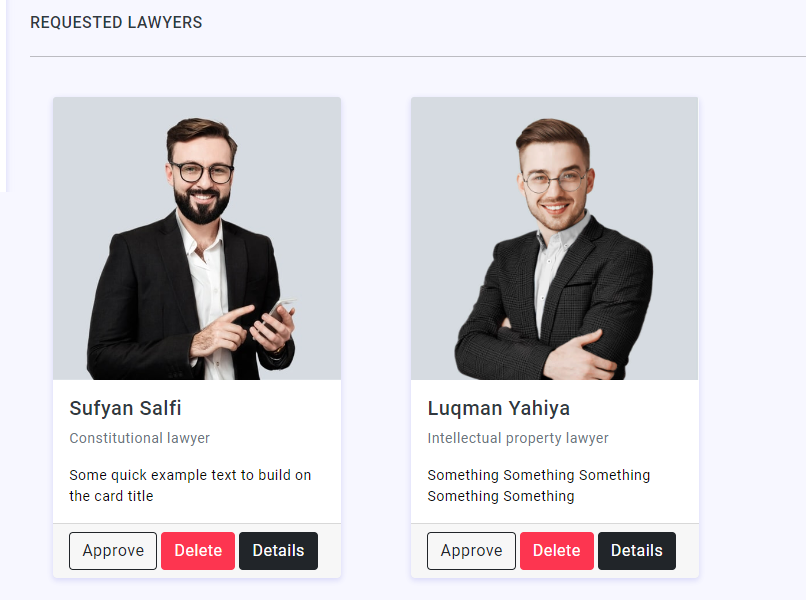


Figure 57: Lawyer Dashboard

## 6.14 Lawyer Requests



## 6.15 List of Lawyers

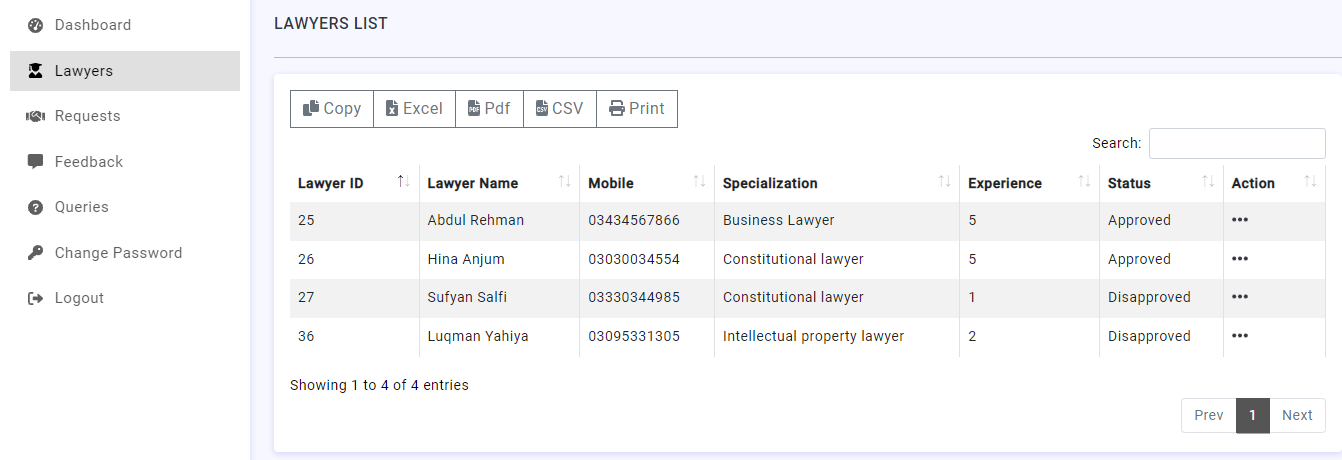


Figure 58: Lawyer List

## 6.16 Database Connection

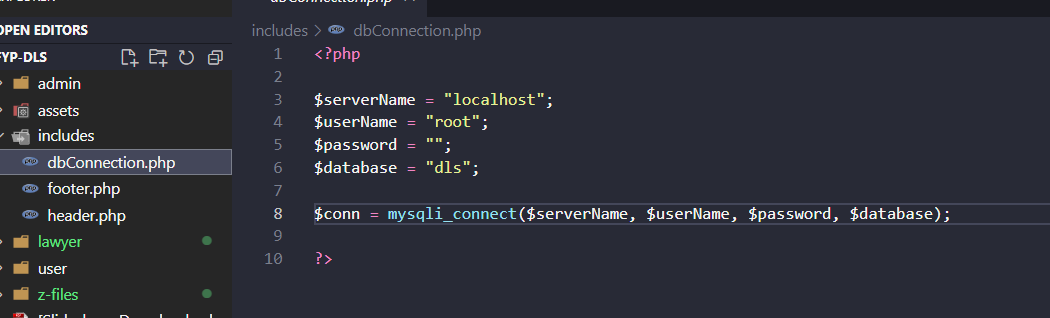


Figure 59: Database Connection

## 6.17 Front End Coding



Figure 60: Front End

## 6.18 Backend Coding

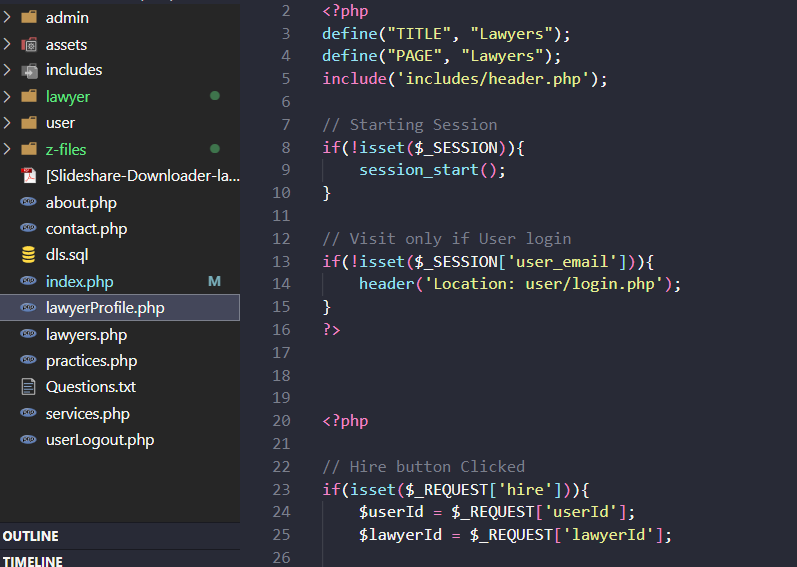


Figure 61: Back End

## 6.19 Database

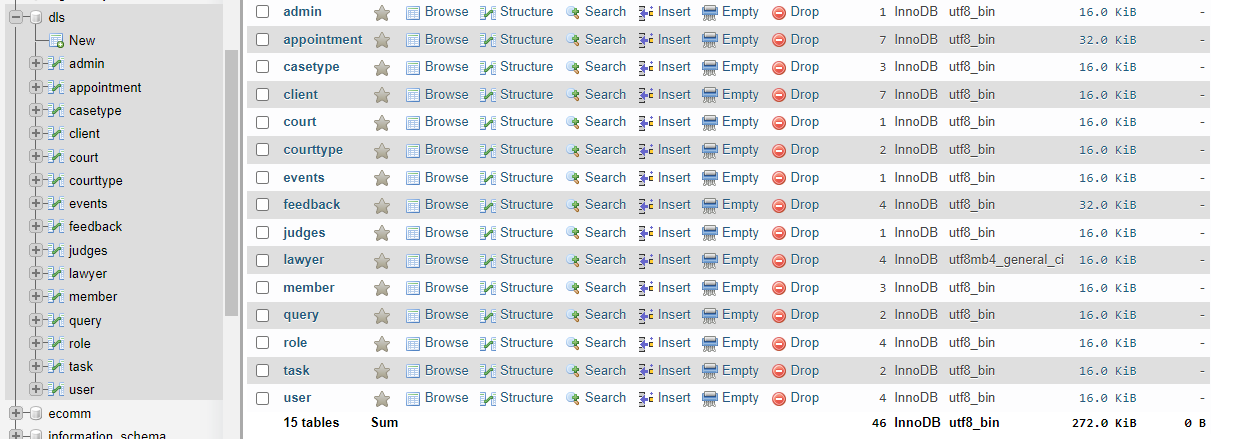


Figure 62: Database

## 6.20 Database Table Structure

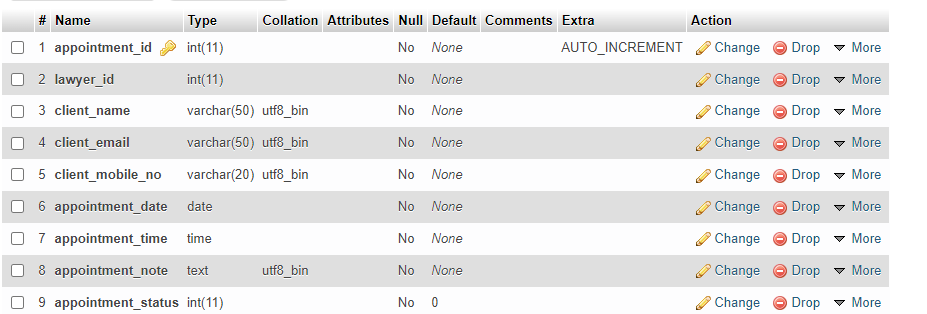


Figure 63: Table Structure