CL-BSCSD-21-10_ST20197917_Lahiru Sampath_Dissertation.pdf

by Dassanayake Mudiyanselage Lahiru Sampath Dassanayake

Submission date: 14-Oct-2021 07:12PM (UTC+0100)

Submission ID: 160731390

File name: 80234_Dassanayake_Mudiyanselage_Lahiru_Sampath_Dassanayake_CL-BSCSD-21-10_ST20197917_Lahiru_Sampath_Dissertation_1200330_889838368.pdf (6.53M)

Word count: 13709

Character count: 86616

CARDIFF METROPOLITAN UNIVERSITY

Faculty of Information Technology

CIS6002 Software Engineering – Final Project Dissertation

[ADVANCED MULTIFUNCTIONAL COURT CASES MANAGEMENT SYSTEM (CCMS) FOR SUPREME CORT LAWYERS BY USING LATEST LARAVEL PHP WEB FRAMEWORK]

Submitted on Sunday, 17th October 2021

By

[Dassanayake Mudiyanselage Lahiru Sampath Dassanayake]

(Student ID – [ST20197917] – [CL/BSCSD/21/10])

This project thesis is submitted in partial fulfillment of the requirements for the degree of

Bsc (Hons) Software Engineering

Abstract

Every procedure has been automated as a result of the technological revolution, which has shifted from manual to digital techniques. The digitalization of business operations inside a legal firm is the focus of this study. The mass of disorganized work and file cabinets is the primary problem that led to the proposal of an automated process for attorneys. The system will include features such as storing distinct client case data for each part of the client case.

The system is designed to integrate a court case management system with calendar reminders and alerts (notifications), data backup, case management, and appointment management, among other features. All of the aforementioned capabilities will be stored in a data base, allowing for always-on access via the application. However, just arranging tasks and filing cabinets isn't enough to maximize efficiency and effectiveness. It's helpful to give search searches for previous comparable instances and how they were handled.

The goal of this project is to demonstrate how digitizing work in a advocate's chambers will reduce the number of manual interactions while simultaneously improving the efficiency and efficacy of the manual working process.

The Agile Methodology is what I utilized to create the project. Since the project must interact directly with customers and users, this approach was used to ensure that we, the developers, knew exactly what the users wanted and how they wanted the system to work based on their input after each iteration.

At the end of project there will be a complete court case management system suited with good functions and features as a result of this software development project.

Table of Contents

Table of Figures

List of Tables

Chapter 01: Introduction

1.1 Background Study

Lawyers play a vital role in the Sri Lankan government sector, helping to strike a balance between truth and justice in order to maintain societal stability. In the event that any citizen of the globe is subjected to an injustice, that citizen has the duty to raise concerns about his or her rights in response to the misbehavior or unlawful conduct of the other party. As a result, attorneys, regardless of their race, color, or nationality, do all in their power to ensure that they get justice. Despite the fact that they put out all of their efforts to get justice, there is no digitalized appropriate method for attorneys to manage their job things, allowing them to operate more efficiently. A digitalized court case management system will be essential for attorneys in the future.

1.2 Problem Statement

The time and effort used by Sri Lankan attorneys in submitting a lawsuit is particularly inefficient, since they must separate all of the correct papers into different files before filing the case. A file must be opened for each and every client who enters a case, and all data and information, papers, pieces of evidence, hearings, and other items must be physically collected and filed by the appropriate lawyer for each and every client.

Furthermore, if any changes are made to those papers, the lawyer must locate each file via the cabinet rack, get the correct file from the shelf, and do each change manually on the documents. A advocate must also keep several files if she or he is managing a large number of cases for one particular client. As a result, the administration of numerous documents for a single customer increases the complexity and exhaustion of the lawyer's job. If a client requests an update, the attorneys must search through all of the client's files to determine which document the up to date should be performed on by reviewing the case facts. Apart from that, the manual administration of the advocate's business procedure keeps them in the conventional period and keeps everybody busy updating & constantly repeating or writing down the case numbers, case kinds, case stages, future court dates, status, and so on.

It is common for lawyers to be faced with circumstances such as re-filing matters that they previously handled months or years earlier. Sometimes a case may not be active for a long period of time, yet the relevant information may be lost, inaccessible, or impossible to locate as a result of the large number of file cabinets that have been placed around the chamber.. As a result, appropriate preservation and long-term protection for case files are critical since such points may be needed as evidence in future instances. According to the authors of the article, "Evans, Nina Price, and James," new technologies must be used in legal companies to be effective. It is their opinion that there is extra pressure on legal firms and a proliferation of legal data resources. The truth is that "information overload and the simple act of recognizing and maintaining these assets has become a problem for legal businesses." (Evans & Price, 2017)

1.3 Project Purpose

This project's main purpose is to develop a well-functional advanced court case management system. By using this system, supreme court lawyers will be able to manage their court cases, clients, and much more legal information. With this application, lawyers can keep the case records at their fingertips.

1.4 Project Overview

1.4.1 Project Scope

The CCMS is a system that helps lawyers to manage their clients' cases online by accessing online databases through the system application. Moreover, this system provides facilities to manage clients, appointments, expenses, etc. The system notifications system will notify today's hearing cases and pending cases. With this application, lawyers can keep the client's legal data at their fingertips.

Moreover, lawyers will be able to manage their clients' work by storing their legal data and information in a database, allowing access to their case files at any time in any location when they are in essential very simply & easily. Through this system application, the court lawyers can keep separate records for each customer & even store several cases for the same customer.

This CCMS will be a internet based system application built using the Laravel MVC architecture. Moreover, this system will be based on the Laravel web framework.

1.4.2 Project Aim

It is the primary goal of this project to emphasize the importance of a CCMS for advocates to manage their clients' cases by storing their case files and data in an online database, allowing them to access their legal data at any time and from any location when they are in need, in a simple and straightforward manner. The lawyer may use this method to keep distinct case files for each customer & even store several court cases for the same customer in one place. Other than that, a lawyer can manage his or her clients, appointments, team members, income, vendors, and expenses as well. Another important consideration is that a lawyer should be able to back up all of his or her data just in case.

1.4.3 Main Objective

Incorporate digitalization into the practice of law so that attorneys may save time and effort by not having to repeat tasks over and over again.

1.4.4 Specific Objectives

1. Replace lawyers' manual working system with a new digitized system. It will also be more efficient and effective than conventional filling techniques since this technology automates them.
2. Develop a well-functional advanced court case management system and fix international and local existing systems by adding new advanced features and functions to fulfill the users' requirements.

1.4.4 Project Deliverables

Lawyers may handle their data with the system's help. To begin, each lawyer must create an account and log in. Legal professionals will be able to organize their cases by client, establishing different folders and files for each one. Lawyers will be reminded of upcoming hearings and ongoing cases via the system, which will create a notification system. These alerts will be produced on their own accord. On a daily calendar, the system keeps track of future customer appointments.

Chapter 02: Literature Review

2.1 Introduction

Despite the fact that Sri Lankan lawyers haven't quite embraced the use of software or any other applications to maintain their documents more efficiently through the use of a court case management system, some international lawyers have taken the first steps toward digitalization as part of the technological renaissance. Foreign legal firms use a variety of different systems in their applications. Cilo, My-Case, File-vine, E-File Cabinet, Cilo-Fox, App4-Legal, "Best-Case," Pro-Law, and a slew of other systems are among those that are available. However, these software tools are not very useful in the Sri Lankan environment, where attorneys practice.

Specifically, their functional needs have been focused on paying & invoicing, cost monitoring, and time tracking, among other things. However, in the case of a Sri Lankan advocate & a customer, the price for the advocate is determined by the practitioner's skillful patterns as well as the connections that exist among the customer & the attorney.

However, with the proposed system, the major features is focused on file and case information and data management, as in a Sri Lankan law firm, the most pressing problem is inefficiency and ineffectiveness, as evidenced by the large number of files crammed into small office spaces in the country.

2.2 Similar Existing Software Reviews

When examining each of the software programs that have been discussed previously, each one has its own set of characteristics that distinguishes it from the others. Sometimes a single or a group of apps with similar but distinct characteristics have been accepted. When comparing the applications "Clio," "My-Case," "File-vine," "e-File-Cabinet," "Case-Fox," "App4-Legal," "Best-Case," and "Pro-Law," it is important to note that they all share functionalities such as legal data management, time tracking, payment services, invoicing & billing, monitoring of expenses in parallel with time usage, integration with Dropbox Cloud Storage, MS Office Excel plugin, auditing, & so on. These applications have also been Although the proposed system will include all of the above-mentioned features (with the exceptions of billing, invoicing, time and expense tracking), it will also include additional features such as a live calendar to track tasks, a notification system, income management, expense management, adding venders, and the ability to download and print case files.

The results of the search revealed that several online apps have offered the function of legal data management for attorneys, and that the most frequently used and famous software for managing lawyers' work is "My-Case," "Clio," "eFile-Cabinet," "File Gate," "Action-step," and other similar programs. (“ABA Legal Technology Buyers Guide”)

2.3 Similar Project Reviews

A system that is based on document watermarking, cryptography, and access control has been suggested by Zhang, Diao, & Wen (2018), who have described methods of integrating security for legal documents & have developed a system that is based on document watermarking, cryptography, and access control. Several hypotheses, like the one advanced by "Zhang," claim that "watermarks are implanted in files by the original author who created them." Until the watermark is implanted in all of the files, they will be unable to be read or written to. Consequently, this will ensure that only authorized users will be able to access or write to the files. Second, users are classified into various security tiers, which are described in more detail below. Users with varying degrees of security have varying amounts of permissions to act on files. The symmetrical key method is being used to secure the file that has been implanted with the 1st watermark as the third degree of protection, according to the specification. Unauthorized users may be prevented from reading the file as a result of this. The public

password algorithm generates a key pair for distribution. The passwords are used in the encryption and decryption of files.

Kang, Chen, & Zhan (2012) have identified data files as a fundamental concept that contributes to the development of a standard method of management practice. It says that "the system is a simple database management system that assists managers in attempting to solve daily activities & relieving work efficiency; improvement in working eficiency and system level; creation of an institutionalized, standardized & scientific management platform; guidance in developing management features; guidance in developing management capabilities" according to the paper "the database helps achieve the following objectives:

Authors Kang et al (2012) claim that the system was built using Microsoft V.S. 2004 development kit, C# programming language and SQL Server 2005 dbms (database management systems). Additionally, there are many capabilities offered by contemporary software providers like My-Case, Clio, and eFile-Cabinet. (n.d.) says that the main functions include monitoring of bills and expenses, settlement calculator, control of closure dates, and sophisticated reporting. More typical features like handling cases and documents, invoicing and collecting from customers, as well as managing duties for the day, week, or month are also available. Trajectories are also managed for the day, week, or month.

There are other components inside a single system that are called Documents management; Capital allocation; Work & records; Group skills, and so on, according to the architecture established by the Mishra, Alok, and Deepti, Mishra teams. In addition, Alok, & Mishra, Deepti created architecture identifies components inside a single platform as, The modules may have different names, but they all have the same goal: to make the work of lawyers simpler. When it comes to printing letters, notices, and reports, the system allows "automated creation and updating of case data," "tracking the availability of files," & "sending a fax to customers with deadline and schedule," according to the documentation. (Mishra & Mishra, 2011).

2.4 Project Related Article Reviews

Aside from the evaluations and comparison of the systems based on their use that have already been done, several other technological experts have suggested the development of a management system for lawyers. In their paper, T. Plessis & A.S.A. Toit highlighted the importance of the above declaration, writing, "Multiple research findings have shown that

advances in information and communications technology (ICT) are beginning to transform the methodologies that lawyers have used to access & process information in order to produce legal facilities to the clients" (Plessis & Toit, 2015).

More importantly, the article emphasizes the need of attorneys having structured access to the data produced on their customers while they are actively involved in the case management process. On top of that, since past notes and case information may be helpful in subsequent instances, the orderly organization of the earlier notes & case data must be readily available at any time (Plessis and Toit, 2015).

The "Aschner" file management system introduced in 2010 featured a method to establishing a generic document categorization covering broad subject categories and differentiating between file classes mostly on retention requirements, which is relevant to this context in terms of file management. Other advantages include the ability for users to add relevant information to on file directory labels for local retrieval requirements, as well as the ability to save money and learn the system more quickly than complicated schemes (Aschner, 2010).

Additionally, Taylor and Aschner, the system's designers, emphasized that "a dbms should be more functional than it should be organizational," and that "a dbms should be more operational than it should be organizational" (Aschner, 2010).

'West' and 'Thomson' research ideas indicate that dms enable law firms to reuse part of their products. To top it all off, some experts believe that DMS are a better option for increasing work effectiveness and production. (Management systems for law firms, 2013).

As further explained by Alok Mishra & Deepti Mishra (2011), digitization of legal work reduces costs, increases efficiency, and provides a competitive edge over others who have hectic and disorganized work schedules, particularly in the context of litigation.

(Classification, 2012) proposed a system in which all papers relating to a single client are kept in the cloud and where both the advocate who produced them and the customer who got them may quickly and effectively access them from any place at any time, evaluate them, or change them. Attorneys in Sri Lanka are currently devoting greater attention to the shift to new technology settings as a result of these qualities To paraphrase "Van Ooarscot," it's because clerks and judges accustomed to performing their work by hand with paperwork are now confronted with court work that is becoming digitized.

Consequently, that article suggests that, in order to satisfy system users, the system should offer equitable features that will assist attorneys in improving the quality of their job while ensuring optimal time management than was previously the case (Van Oorschot, 2014).

Aside from the aforementioned primary goals of file management, attorneys encounter a slew of difficulties during their careers as attorneys. Lawyers are often faced with circumstances such as refiling court cases that they handled months or even years ago. Because of the large number of file cabinets that have been put in the chamber, it is often not a matter of how much longer it will take, but the information may be missing, inaccessible, or impossible to locate. The appropriate preservation and long-term protection of case files are thus critical since those elements may be required as evidence in future instances. The authors of the article, "Evans, Nina Price, James," emphasize the need of using modern technologies in law firms in their paper. As according them, there is increased pressure on law firms, as well as an increase in the proliferation of legal data resources. Actually, "information overload, as well as the simple act of recognizing and maintaining these properties, becomes a problem for law firms" (Evans & Price, 2017).

Increasingly, technological advancements are prompting legal firms to use management systems and other technologies. When it comes to customer, case-related and other business information, cloud storage capacity enables info to be kept centrally & accessible from anywhere, including work or home with substantial charge savings. Advocates can access digital papers in court because of the portability of their devices" (Evans & Price, 2017).

The authors of the article, "Legal Documents Handle Using Big Data," Siddharth, Aarthi, Athreya, & Balaji, explain how they came up with their solution. They have taken note of the problems raised above and have suggested the idea of data mining via the usage of the "Hadoop" method, which has been endorsed by the research community. There are court cases where a specific order or affiliation must be visible at the most basic level, to gather data that could be the tipping point needed to win a specific case, & also as an alibi for any voice statements given during a particular session, they argue that the technical be of organizing such data can be beneficial" (Athreya, Aarthi, Siddharth, & Balaji, 2018).

The natural resources, such as details of the case or documents covering every aspect of a specific case, are collected & compiled in a organized way, according to (Aarthi et al., 2018), and then those documents are studied using the Sql Language and some basic Apache Spark programming, as described in (Aarthi et al., 2018). After the analysis is completed, the

unrefined files are saved in a cluster for further use. All of the steps in the data analysis process are meticulously documented in a sql database referred to as metadata. This data may be utilized for historical trend analysis, as well as for backup purposes in the event of information loss. Along with the gathering of original information, saving, arranging, & updating the data's safety is also necessary, since each data set contains "a private set of data about others about their personal lives" that must be kept confidential.

Aside from that, attorneys' tacit knowledge should be documented so that they may refer to it when repeating points in subsequent precedents in future cases, or when it may be required in their everyday job. What this means is that "experienced attorneys develop forms that contain extensive comments & practical remarks that serve as templates for the kinds of papers that are often needed in the firms everyday routine" (Plessis & Toit, 2015).

As a result, it is obvious that each and every record and form is critical in the practice of law. As a result, data should be collected in a systematic manner in order to minimize the stress associated with their hectic work schedules. The issues raised above clearly demonstrate how important it is for attorneys to have well-organized and easy-to-access documentation. If you have easy access, there should be no data breaches or security worries regarding the company's personal information or case details since you have it. As the author (Scheffer, 2017) points out, "the document is not just a distracting medium, but it is also an essential member of legal discourse, and it gives the customer a voice in that discussion." As a consequence, protecting private information should be of the utmost significance. The main worries attorneys may have in terms of systems, according to Plessis & Toit, showed a high degree of worry linked to information privacy, computer viruses that damage data, and the validity of information" (Plessis & Toit, 2015).

Consequently, as previously said, systems have been suggested to be created, and some of these systems are being developed with the capabilities that have been recognized as being answers to the current business issues. Because technology is so omnipresent, persons are becoming more cautious about each single new item that enters the mainstream. Customers have become more demanding of excellence, effectiveness, and efficiency in any service or business they participate in as a result of this increased attentiveness. In the words of Evans and Price (2017), "Customers are frequently ahead of attorneys in terms of adopting new technology, & that they also have better access to legal material that is easily accessible on the web."

Aside from that, several nations have embraced the use of information technology in their legal systems. When it comes to use of the information & communication technologies (ICTs) in the courtroom, transparency and efficacy are two of the most often cited benefits. It has increased the accessibility of information as well as the possibility of judicial judgments. (Filho et al., 2019).

In so many national judiciaries, court automating is not a new phenomenon; nevertheless, the breadth and degree of development of court automation varies considerably, even among the most sophisticated industrialized nations. The integration & digitization of case data, court case management, legal details management, as well as the electronic transmission and reception of documents have only been tried in a few nations up to this point. Many courts claim to have made some progress, but just a handful have really done so. (2013)) (Greenwood and Bockweg, 2012).

2.5 Chapter Summary

This chapter we were discussed all about the literature review related to existing works and studies to understand moreover about CCMS. This chapter basically related several studies which are; Introduction, Similar Existing Software Reviews, Similar Project Reviews, Project Related Article Reviews.

2.6 Review on Similar Systems That Exists

Table 1. Review on Similar Systems That Exists

Main Functions & Features	Similar Applications	Proposed System
Advanced dashboard	Have basic dashboards	✔
Live daily calendar to track future appointments	✘	✔
Clients management	✔	✔

Court cases management	✔	✔
Add tasks to daily calendar	✘	✔
Clients appointments management	✔	✔
Multiple members support	✔	✔
Income management	✘	✔
Expense management	✘	✔
Legal cases/client's data and information backup option	✘	✔
Pending case notifications	✘	✔
Today's hearing case notifications	✘	✔
Appointment notifications	✔	✘
Time tracking	✔	✘
Client's case reports download & print option	✘	✔
Mailing system	✔	✔
System settings section	✘	✔

Chapter 03: Analysis and Requirements

3.1 Introduction

The methods that were utilized to accomplish the project's goals are discussed in detail in this chapter. The document also primarily illustrates the methods that may be used to collect user needs and specifications.

3.2 Development Methodology and Approach

It was necessary to carry out the project development approach in two stages. "The perspective of Sri Lankan attorneys about the establishment of a court case management system, as well as the existing difficulties they have while keeping large amounts of files, data, and pressing needs" were the objectives of Phase One of the study. The second phase of project development included an examination of the systems that had been established by early developers, as well as a study of the relevant literature and works of art. As a result of its adaptability, the agile approach is being utilized in the system's development process.

The system was developed using the agile development approach, which was used throughout the process. The Agile method offers an incremental & iterative approach to system design, as opposed to the waterfall model, which assumes that software development proceeds methodically from start to finish from the beginning. This approach provides the user with regular and early chances to examine the system & make choices about the project, as well as to make modifications to the project.

3.2.1 Development Approach Stage 01: Requirements Gathering

The needs for the system application were initially collected via telephone interviews with a sample of attorneys, which served as the first stage. In addition, a Microsoft form was sent among the attorneys in order to assess the feasibility of digitizing the lawyers' work in the setting of Sri Lanka.

3.2.2 Development Approach Stage 02: Analysis

Second, the information collected was evaluated to determine whether or not the attorneys preferred the digitization of the manual method and whether or not they were likely to switch to a fully digitized system. According to the findings of the study, their primary point is that traditional document and legal data administration makes them inefficient & time-consuming. They also point out that the traditional technique of file & legal data administration wastes a significant amount of advocate time in looking for case files via each shelf & then checking through every page of the file they use. Moreover, it has been determined that the vast majority of them are preoccupied with the current systems, making it inconvenient for them to participate. In addition, a literature review was carried out in order to get an understanding of the current systems.

3.2.3 Development Approach Stage 03: Design

The design of the so was drawn up as the third stage, in order to ensure that it met the needs of the lawyer. UML diagrams and a system architecture were drawn in order to put the system design into practice. The system architecture was created to represent how attorneys and team members connect with the servers, as well as how each lawyer and team member collaborates on the internet. Attorneys and team members may join through a data connection, and lawyers & team members can keep their legal data via an online database, according to the system architecture that has been developed. The program was created to provide access to almost any device that is compatible with any OS on the market.

3.3 Requirements Elicitation for Proposed System

3.3.1 Questionnaire in the Online Survey (Microsoft Form)

Table 2. Questionnaire in the Online Survey (Microsoft Form)

Questions	Answers (Provided)
1. Are you facing issues while using manual court case management?	• Yes • No

2. If yes, what are the issues that you're facing while using manual case management?	• Time and dates management issues • Case files and data misplacements • Difficulties to find exact case file from files cabinets • Client's information management issues • Other management issues (appointments, income, expenses)
3. Do you think digitalized case management system will solve those issues?	• Yes • No
4. Would you like to work with digitalized management system?	• Yes • No
5. If yes, what are the main functions (features) do you prefer or want from digitalized court case management system?	• Client's case management • Client's information management • Future tasks management • Client's appointments management • Appointments tracking calendar • Other managements (income, expenses, venders, team) • Data backup option

3.3.2 Online Survey (Microsoft Form) Results

Table 3. Online Survey (Microsoft Form) Results

Question	Answers (Provided)	Count
Q01	• Yes	09
	• No	01
Q02	• Time and dates management issues	09
	• Case files and data misplacements	09
		09

	• Difficulties to find exact case file from files cabinets • Client's information management issues • Other management issues (appointments, income, expenses)	08 07
Q03	• Yes • No	09 00
Q04	• Yes • No	09 00
Q05	• Client's case management • Client's information management • Future tasks management • Client's appointments management • Appointments tracking calendar • Other managements (income, expenses, venders, team) • Data backup option	09 09 09 09 09 08 08

3.3.3 Interviews Overall Result

The final overall result according to the age of the Sri Lankan advocates' analysis interview result is shown in the following Figure 1.

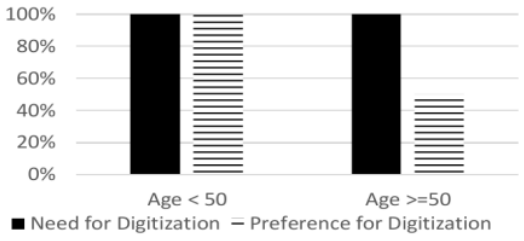


Figure 1. Overall Result From Interviews

3.4 Requirements Determination

A requirement is a formal description of a system's functionality that is expressed in formal terms. It includes conditions pertaining to the overall performance and functioning of the whole system. It is possible to divide functional needs into two categories: functional requirements & non-functional requirements, respectively.

3.4.1 Functional Requirements

- **Admin's (Lawyer)**

System Login: Lawyers should login to the system to access the system. The system will inform the lawyer if the username or password is invalid or if the system will give access permission to the lawyer.

Recover Password: If somehow the lawyer wants to recover the password, the system will send the password recovery email to the lawyer's email address. Lawyers can use this password recovery email to replace the new password with a forgotten password. This password recovery email will only be available for one time.

Profile Management: A lawyer can manage his or her account profile by editing or changing the profile picture, first and last name, email address, mobile number, registration number, associated name, address, zip code, country, state, and city.

Change Password: A lawyer can change a password by providing the old password.

System Logout; Lawyers can logout of the system using the upper dropdown menu 'Logout' button. After that, the system will redirect you to the login portal.

System Dashboard: The dashboard of the system will show the summary of data and some more important information related to the current day. like cases, appointments and linked them to their pages. It also shows the 'Case Filling No' and 'Appointments' in the daily calendar.

System Notifications; There are two different notifications on the menu; today's hearing case notifications and pending case notifications. Under today's hearing case notifications, lawyers will get notification for the cases that have today's hearing date. Under pending case notifications, lawyers will get notification for cases that have pending case dates. These two

types of notifications will appear on the upper menu. Lawyers can see notification details by clicking on those notifications.

Clients' Management: In the "Client Section", lawyers can see all the details about their clients. A lawyer can add his or her clients by clicking on the "add client" button. Fields to be entered by a lawyer for any client are first name, middle name, last name, gender, email, mobile number, alternate mobile number, address, country, state, city, reference name, reference mobile number.

Also, more people can be added by clicking on the checkbox "Add more people", and there is a choice between a single advocate and multiple advocates for that particular person. First name, middle name, last name, mobile number, address are the fields of a single advocate person, and first name, last name, mobile number, address, and advocate name are the fields of a multiple advocate person.

In 'Action', lawyers can see all the details about a client's history, client hearing history, and client account history. When clicking on a client, the details of that particular client will be shown. These details will include: name, mobile number, alternate mobile number, reference name, reference mobile number, email, address, city, state, country.

When clicking on the "Case" button, the cases of that particular client will be shown and details will be like "Case Detail," "Court Detail," "Next hearing date," "Case Status," etc.

When clicking on the 'Account' button, the account of that particular client will be shown and details like Invoice No., Client name, Total amount, Due amount, Status etc. will be included. When clicking on the 'Action' button, you can add a payment and see the payment history of the client.

Court Cases Management: In the "case section", lawyers can see all the details about a case and add a case through the "add case" button. There are basically four sub sessions in the case section, which are: running, important, no-board cases, and Archived Cases. The client's name, registration number, case type, court, court no, magistrate, petitioner and respondent names, next date, case status, assigned to, and edited by will be displayed in the list of the cases. Also, the user can sort the cases according to the next date filter.

There are a few fields to be entered for adding cases. Once all the mandatory fields of this form are added, the case of that particular client will be added in the Running cases section. If the lawyer has marked any case as important, then all those cases will be listed in the Important

Cases section. If any case is declared as a No Board Case, then those cases will be listed in the No Board Case section. Those cases which are dismissed or closed for any reason will be displayed in the Archived Cases section of the Case section.

The fields to be entered by the lawyer for any case are Client Details, Case Details, FIR Details, Court Details, and Task Assigned. In the client detail section, add the client's name, respondent's name, etc. In the case detail section, add case no., case type, filling no., etc. In the FIR detail section, add the FIR no., FIR date, etc. In the court detail section, add the court no., court type, etc. In the task assigned section, select the lawyer and also the task assigned to a particular user (team member).

In the 'Action menu', when clicking on 'view', the lawyer (user) can see three menus like "case details," "hearing history," and "case transfer history." When clicking on "view" in the action bar, a particular case will be opened and the user can see the case details like case type, filling number, filling date, registration number, CNR number, first hearing date, court no., judge, etc. When clicking on history, the user can see a particular case's history. In that you can see the history of hearing dates and etc. When clicking on transfer, the user can see a particular case's transfer history. In that, you can see the registration number, transfer date, court number and judge, and so on.

When clicking on the 'Add Next Date' button, the user can add the next hearing date and change the case status. When clicking on the 'Case Transfer' button, the user can transfer a case from the current court to another court with the transfer date.

Lawyer Tasks Management: In the 'Task Section', lawyers can see all the details about a task and add a task through the 'Add task' button. In the 'Task Section', a list of added tasks is displayed. Details like task name, start date, members, status, etc.

Client's Appointments Management: In the 'Appointment Section', users can see all the details about an appointment and add an appointment through the 'Add appointment' button. An advocate can list out or add his appointments with his existing clients or new clients by adding all the details of the appointment. Also, users can sort the appointments according to the date filter.

For adding any appointment, the user has a choice between adding a new client or the existing client. If a user chooses a new client, he needs to add the client's name; if a user chooses an

existing client, he needs to select a client from a given list of clients. Other fields like mobile number, date, time, and notes (optional) have to be filled in to add an appointment.

When an appointment is canceled, then admin can change the status of the appointment to something like "cancelled by client" or "cancel by advocate".

Employees (Team Members): There are two sections available: roles and team members. In the "role section", lawyers can see all the details about a role and add a role through the "Add role" button. After adding roles, all the roles added by the lawyer are listed here. Details such as role name, action, and details are displayed. When clicking the "Add role" button, lawyers can add a role name and a role description. After creating a role, a lawyer can assign permissions from the action menu.

In the 'Team Member Section', lawyers can see all the details about a team member and add a team member through the 'Add team member' button. After adding team members, all the team members added by the lawyer will be listed. Details like team member name, email, contact number, etc.

To add team members (employees), fields like profile picture, first name, last name, email, mobile number, address, zip code, country, state and city, role should be filled up by the main lawyer.

The main lawyer can give different system access credentials to his or her team members by adding "Role Permissions". System access credentials will be dependent on team members' roles. A lawyer can choose access types for each section (appointments, expenses, etc.). Such as view, add, edit, delete.

Lawyer Income Management is divided into two sections: services and invoices. In the "service section", lawyers can see all the details about services and add services through the "Add service" button. After adding services, all the services added by the lawyer will be listed. Details like name, amount, status, and action are displayed.

A lawyer can add services through the 'Add Service' button (like fees, etc.) and a lawyer can use this service at the time of creating an invoice.

In the 'invoice Section', lawyers can see all the details about the invoice and add the invoice through the 'Add invoice' button. After adding invoices, all the invoices added by the lawyer

will be displayed. Details are invoice number, client name, total amount, due amount, paid amount, status, and action.

A user can create an invoice for a lawyer's client by filling in all the details. A user can add invoice due payments when he or she receives due payments from a client. By clicking on the 'Action' button, the user can see the invoice payment history.

Add Venders: In the "vendor section", users can see all the details about a vendor and add them through the "Add vendor" button. After adding venders, all the vendors added by the user will be displayed. Details such as vendor name, mobile number, status, and action are displayed.

When clicking on 'Add vendor', fields like company name, first name, last name, email id, mobile number, alternate number, address, country, state, city, GSTIN, and PAN will be shown and the user (lawyer) can fill them out.

Expense Management: A user can choose between two sections: expense type and expense. can add a new expense type through the 'Add Expense' button (like a phone bill, stationary, etc.), and the user can use this expense at the time of creating the expense. After adding expenses, all expenses added by the user will be displayed. Details are vendor name, invoice number, total amount, due amount, paid amount, status, and action.

General Settings; Using this section user (lawyer) will able to add or change settings such as case types, court types, courts, case status, judge, tax, general settings, database backup.

- **Team Members' (Employees)**

Team member's functionalities will be depend on their role given by lawyer. As an example, accountant can access the income management, venders, expenses management. These type of access permissions should be given by the lawyer. Lawyer can add more than two team members if he or she wants.

- **Team Member 01: Accountant**

Account can manage lawyer's income, office expenses and can add vendors to the system. Moreover, can access tax settings as well.

- **Team Member 02: Personal Assistant**

Personal Assistant can manage clients and client's appointments.

3.4.2 Nonfunctional Requirements

Nonfunctional requirement define how a system should act and what bounds there are on its functionality.

System Performance: The system can handle large database and advanced features without reducing system performance.

System Availability: The system will be accessible every day & lawyers can access the system anytime without any delay.

System Usability: The system usable for all Sri Lankan and foreign lawyers. This system very easy to use and handle.

System Reliability: The system has little system failure occurrence & low risk. And will not take considerable time to correct it.

System Accuracy: The system will work precisely without high failure or fault.

System Security: each system user (lawyer and team) is required to login. The system allow to lawyers and team with assigned usernames and passwords. The system will be designed to make it impossible for unauthorized people to login without valid usernames or password.

3.4.3 Hardware and System Requirements

- Electronics – laptop or desktop pc, keyboard and mouse
- RAM – minimum 2GB or 4GB
- Screen Resolution – minimum 1920x1080
- CPU – minimum Core i3 or i5
- Hard Disk Space – minimum 500MB
- Internet connection

Chapter 04: Feasibility Study

4.1 Financial Feasibility

Fidelity in terms of finances A hosting fee may be charged for CCMS due to the fact that it is an internet-based program. Considering that the system does not include any multimedia data transmission, the bandwidth needed for the functioning of this web application is very minimal. Using freeware software standards, the system will be built. The prospective users will not be charged anything. A fee will be charged for bug repairs and other maintenance activities. Sri Lankan supreme court attorneys and other lawyers will be the prospective market area at the outset of the project. The consumers will get a slew of advantages, in addition to the financial costs. It is apparent from the above that the CCMS project is financially viable.

4.2 Technical Feasibility

Feasibility in terms of technology Project CCMS is a fully-featured web-based software solution. PHP, SQL, JSP, CSS, SCSS, SASS, and the PHPStorm IDE are the primary technologies and tools used with CCMS. Diagramming tools like as Visio and Draw.io are also used in conjunction with the CCMS. Each of the techniques is publicly accessible, & the technical skill required and manageable. The time constraints of software development and the ease with which these technologies may be implemented are in sync with one another. After a successful trial run on a web hosting service space (XAMPP 127.0.0.1), the web application will be moved to a premium web hosting space with adequate bandwidth for further deployments. Because this program does not include any multimedia features, the bandwidth needed by this software is very minimal. CCMS is technically viable, as shown by the fact that the project is technically feasible.

4.3 Resource Feasibility

Resource feasibility Resources that are essential for the CCMS project contains, Programming electronic device such as laptop, free hosting space, Programming and development tools (free). So it's vibrant that the project CCMS has the essential resource feasibility.

4.4 Time Feasibility

Time feasibility mean the time it takes to complete this system, the system operationally started on 29/07/2021 and will be completed on 09/10/2021.

Table 4. Time Feasibility

Task	Activity	Duration pay week	Percentage (%)
A	Analyzing	02 weeks	15%
B	Designing	01 week	10%
C	Coding	03 weeks	50%
D	Implementing	02 weeks	25%

Chapter 05: Risk Analysis

Table 5. List of Identified Risks

Risk Index	Identified Risks
R01	Achieve the project aim within the given time period.
R02	Complete project main objective 100%
R03	Complete project specific objective number one 100%
R04	Complete project specific objective number two 100%
R05	Provide project deliverables.
R06	Continue development approach and methodology.
R07	Gather the required requirements for the project.
R08	Analyze the gathered requirements correctly.
R09	Step into designing the system according to the users' requirements.
R10	Design system mockups/wireframes according to users' requirements.
R11	Manage hardware issues while dealing with system development.
R12	Develop front-end of the system according to mockups and wireframes.
R13	Choose best and most suitable design pattern for future design and development.
R14	Choose best and most suitable API for system design and development.
R15	Use those design patterns and API correctly.

R16	Fix bugs and issues during the development process.

Table 6. Probability Definitions

Probability (%)	Description
0 - 20	Very unlikely the risk will occur
21 - 40	Unlikely the risk will occur
41 - 60	Even likelihood the risk will occur
61 - 85	Likely the risk will occur
86 - 100	Very likely the risk will occur

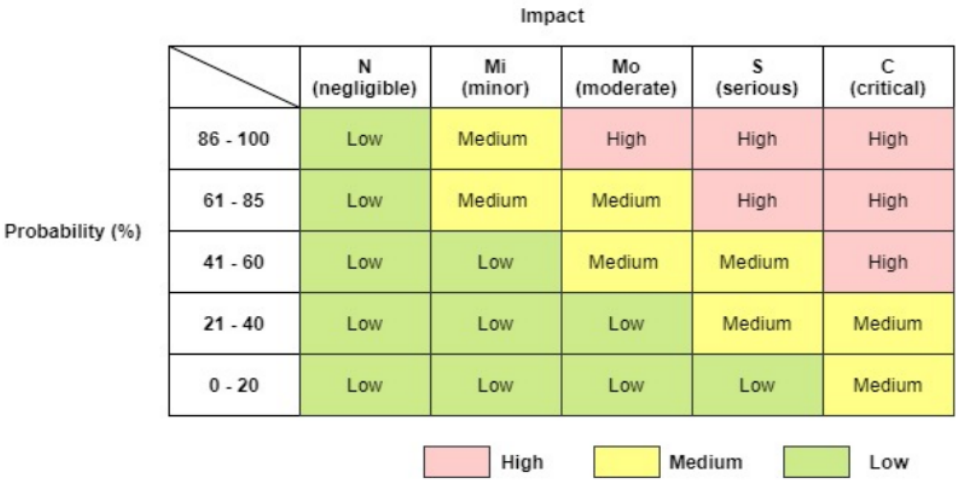


Figure 2. Risk Matrix

Table 7. Analyzed Risk Matrix for Identified Risks

Risk Index	Probability (%)	Impact	Risk Level
R01	10	C	Medium
R02	50	Mi	Low
R03	50	Mi	Low
R04	50	Mi	Low
R05	15	C	Medium
R06	55	Mi	Low
R07	55	Mi	Low

R08	60	Mi	Low
R09	86	Mo	High
R10	80	Mo	Medium
R11	84	Mo	Medium
R12	82	Mo	Medium
R13	87	Mi	Medium
R14	80	Mo	Medium
R15	59	Mi	Medium
R16	87	Mo	High

Chapter 06: System Design

6.1 System Architecture Diagram

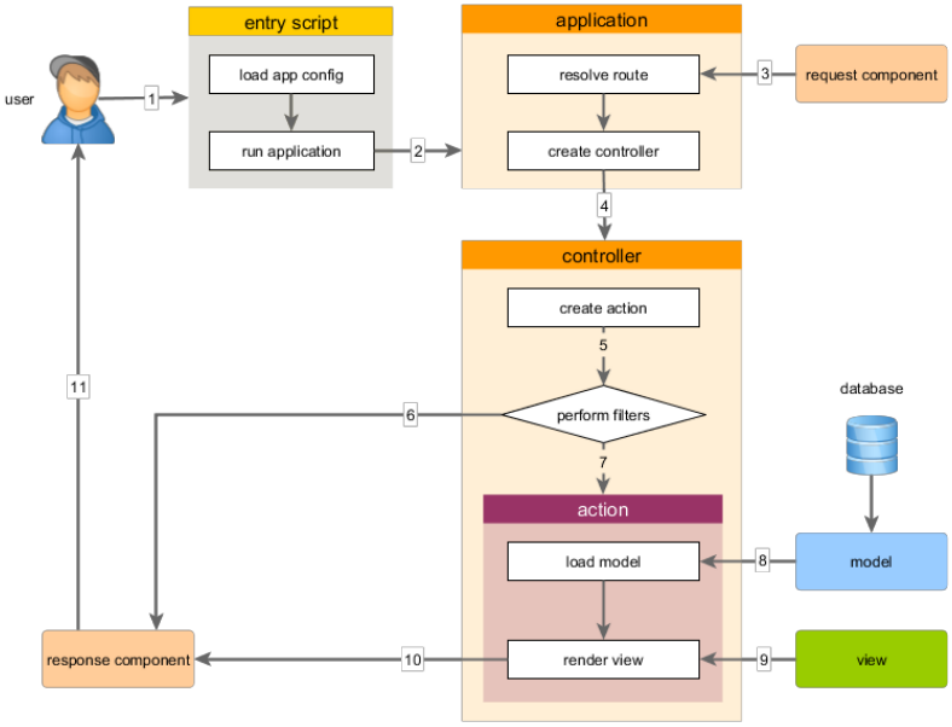


Figure 3. System Architecture Diagram

6.2 UML Diagrams – ER Diagram

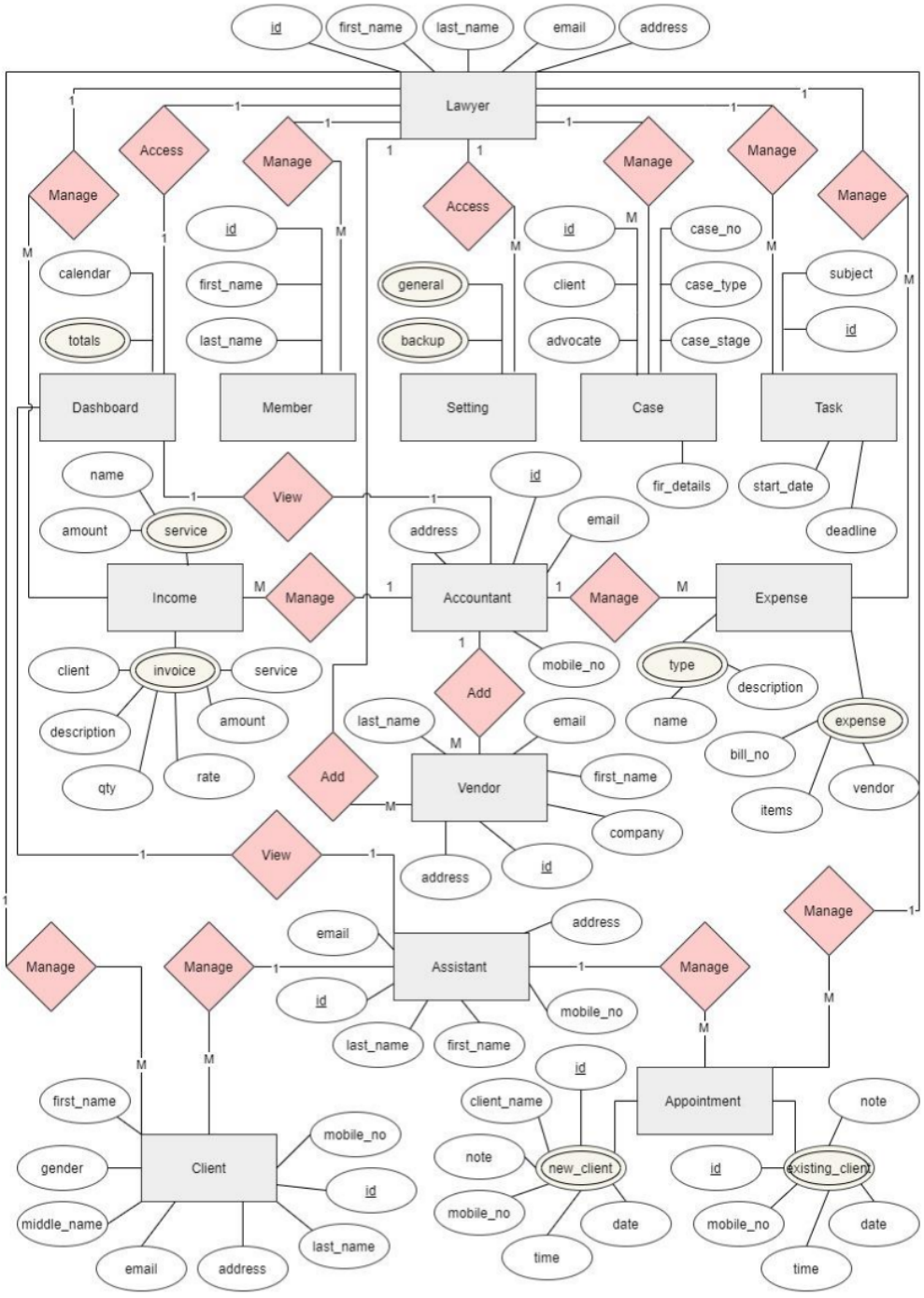
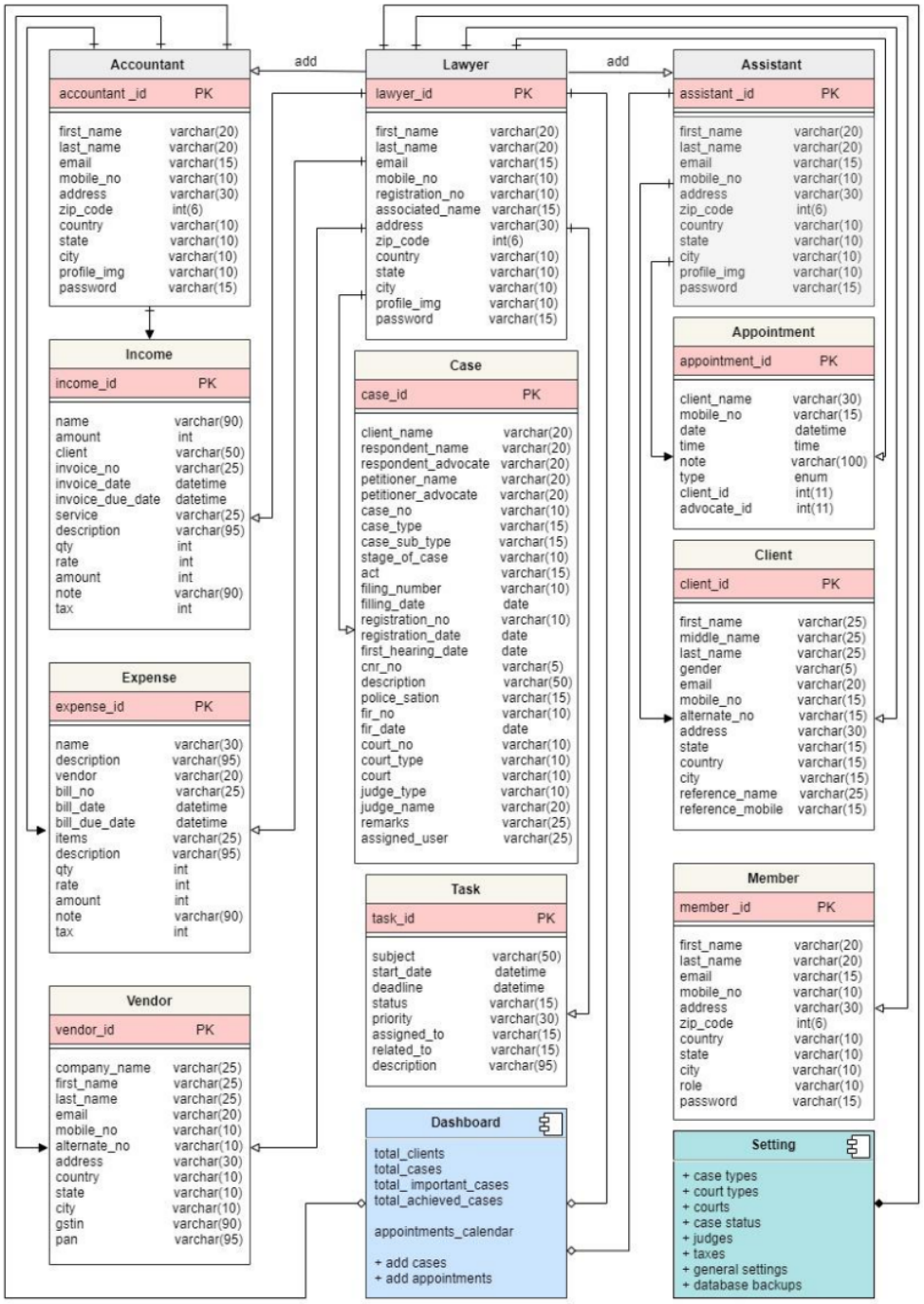


Figure 4. Entity Relationship Diagram

6.3 UML Diagrams – Class Diagram



26

Figure 5. Class Diagram

6.4 UML Diagrams – Use Case Diagram

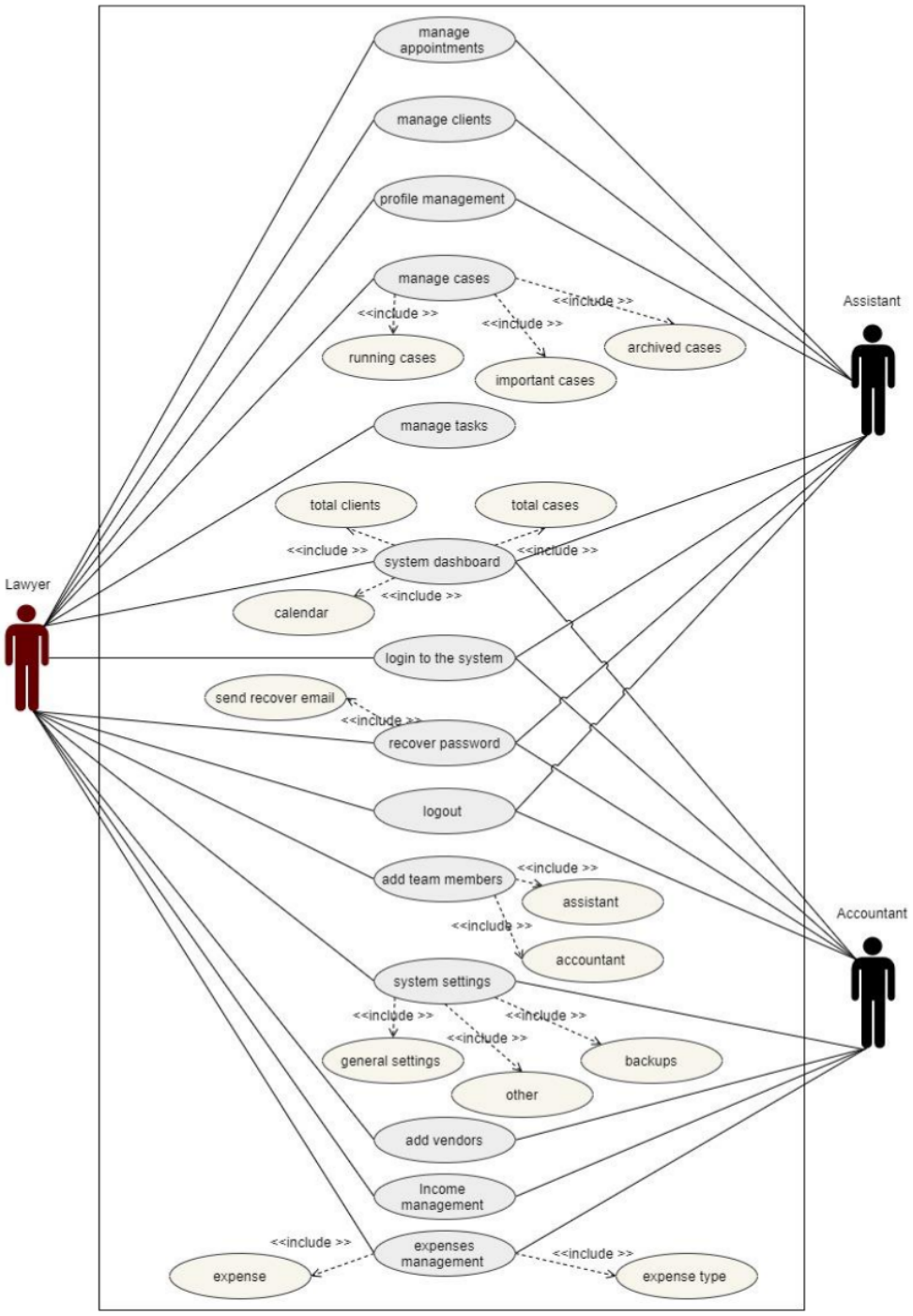


Figure 6. Use Case Diagram

6.5 UML Diagrams - Sequence Diagram

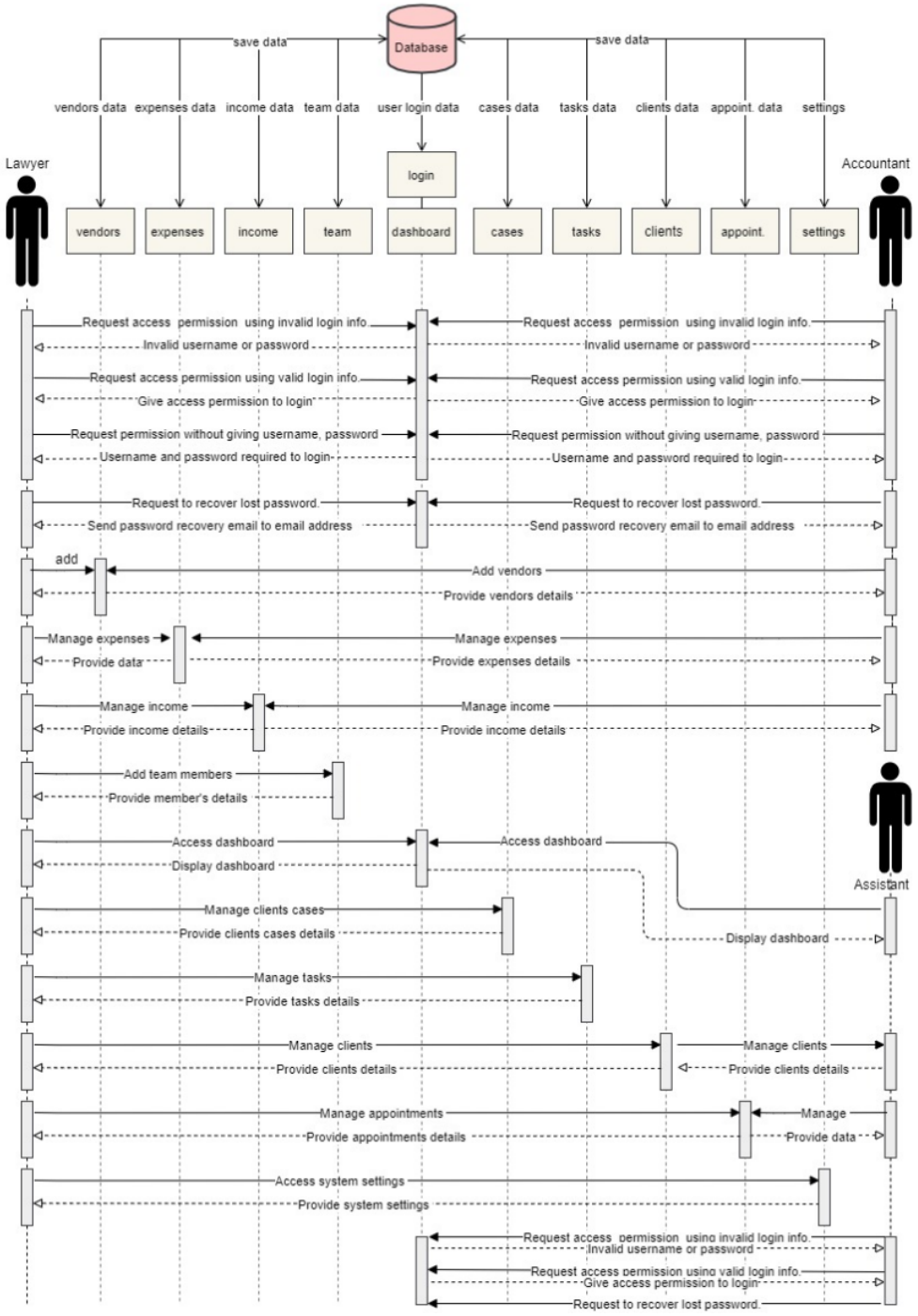


Figure 7. Sequence Diagram

6.6 System UI Wireframes & Mockups

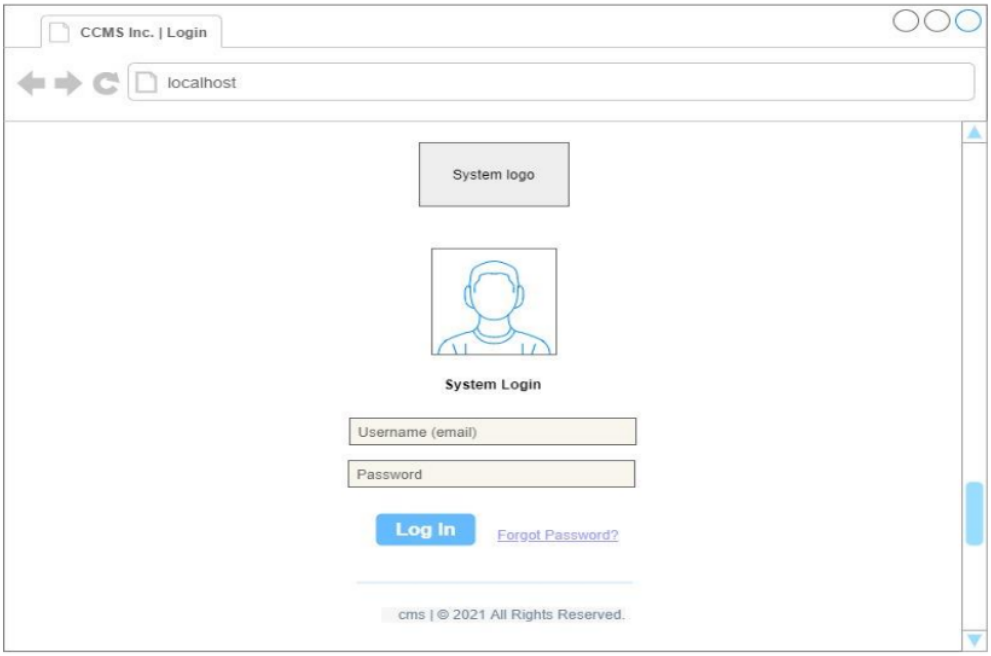


Figure 8. Wireframes & Mockups - Admin/Team Members Login Page

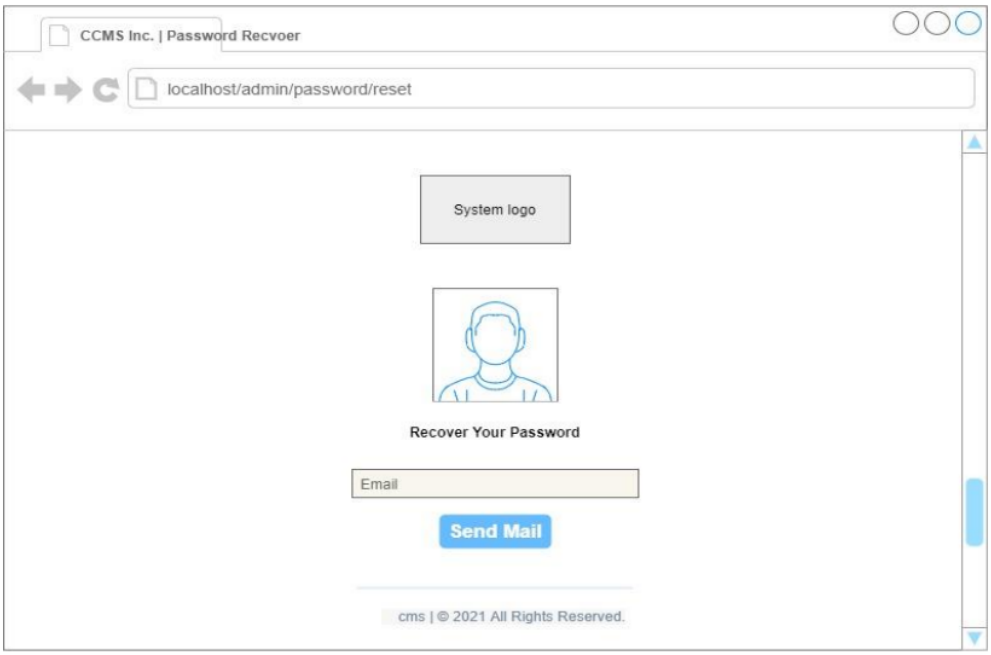


Figure 9. Wireframes & Mockups - Password Recover Page

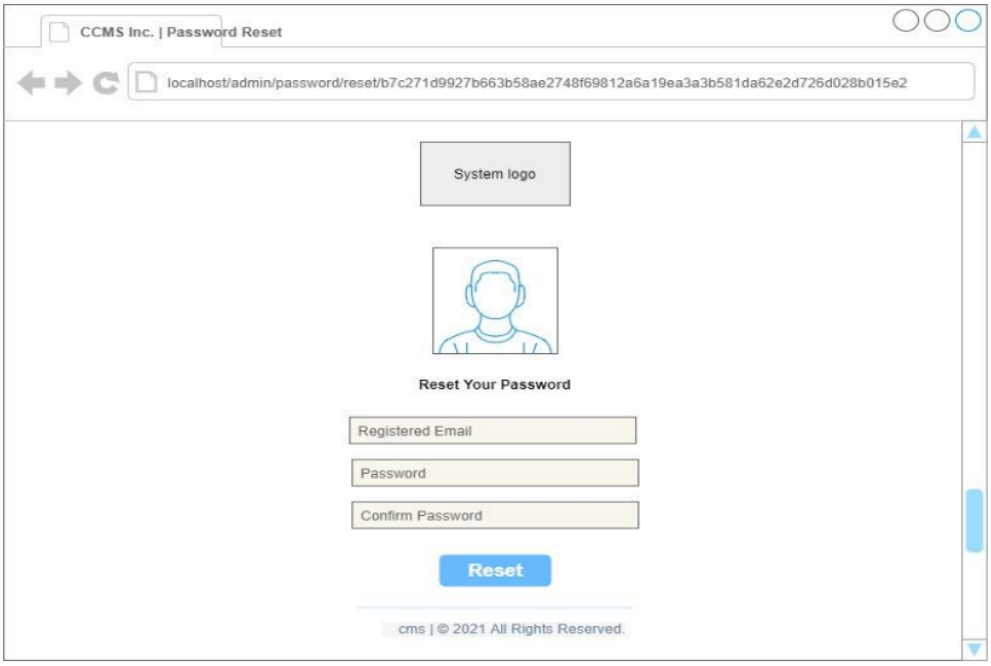


Figure 10. Wireframes & Mockups - Password Reset Page

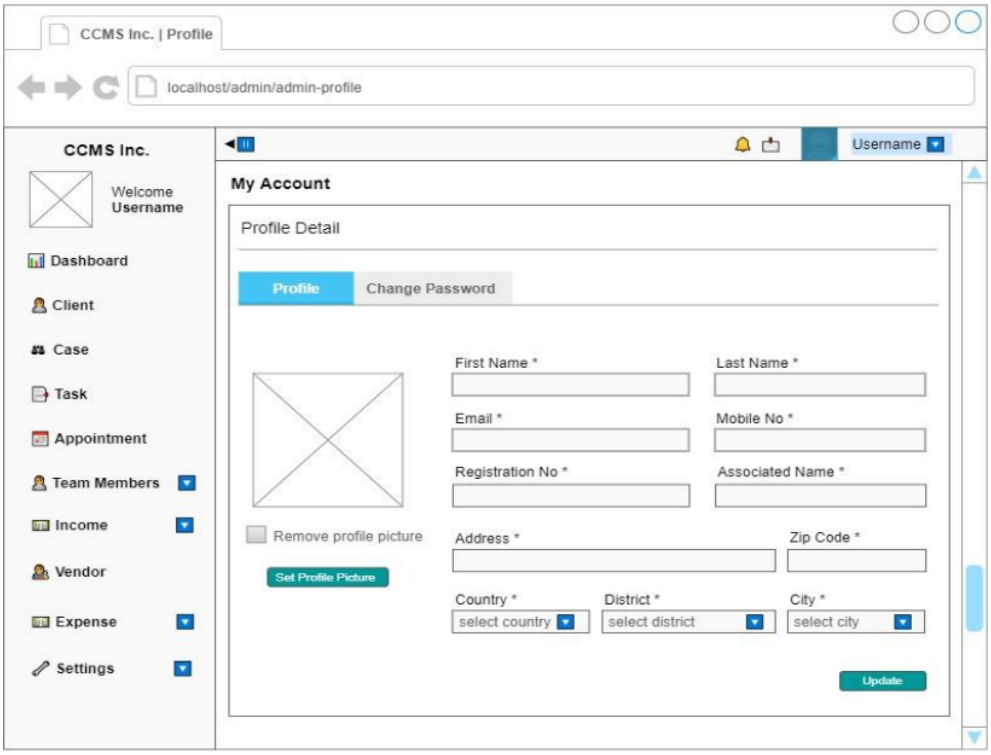


Figure 11. Wireframes & Mockups - Profile Update (Admin)

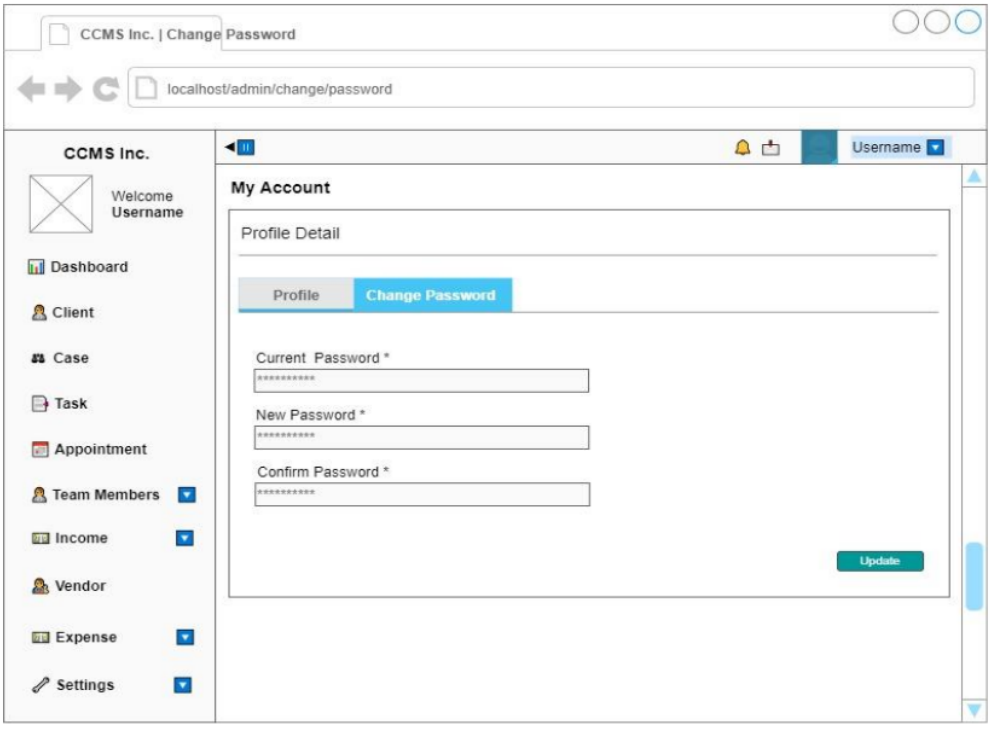


Figure 12. Wireframes & Mockups - Password Change Page (Admin)

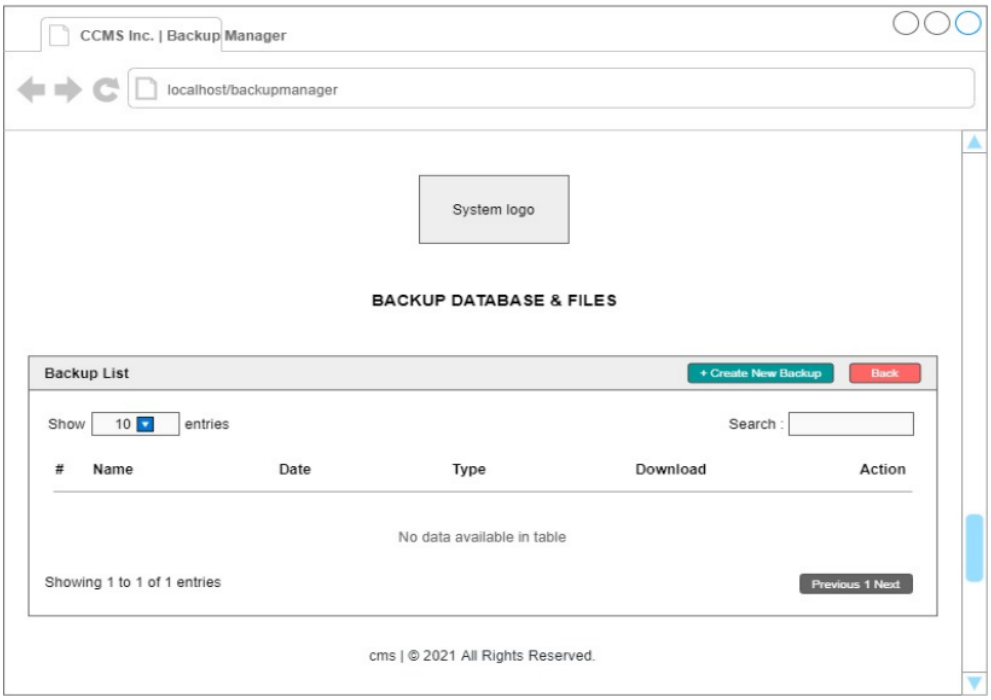


Figure 13. Wireframes & Mockups - Database and Files Backup Page

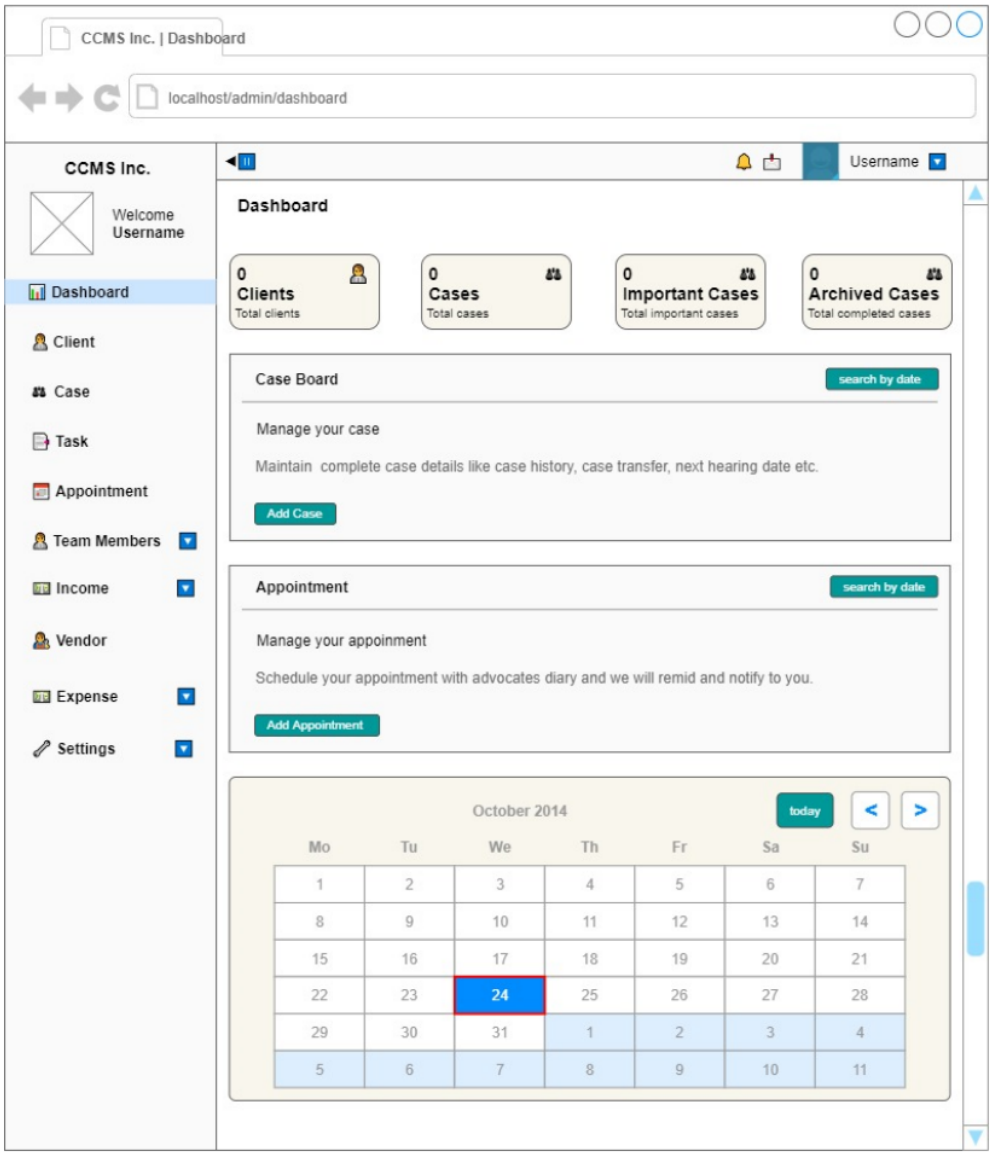


Figure 14. Wireframes & Mockups - System Dashboard

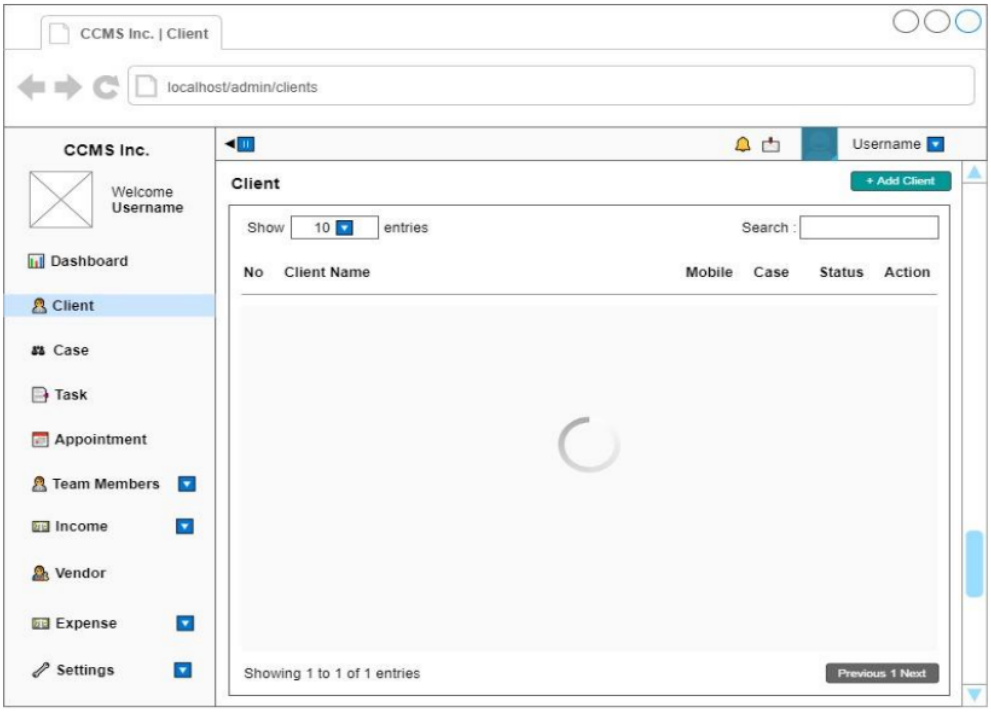


Figure 15. Wireframes & Mockups - Clients List Page

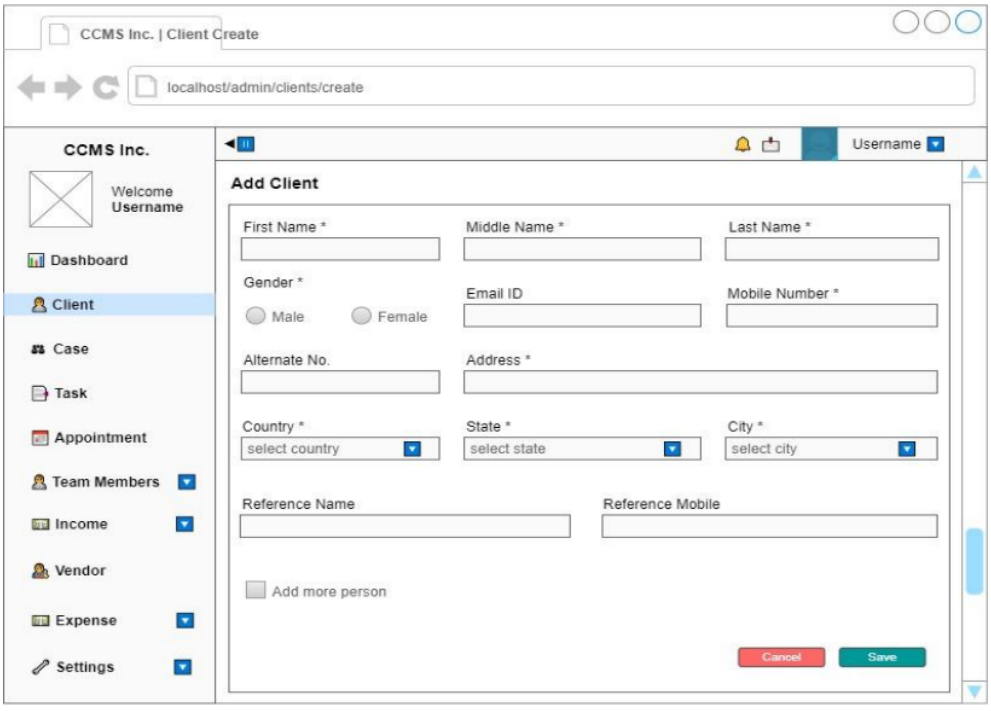


Figure 16. Wireframes & Mockups - Add Client Page

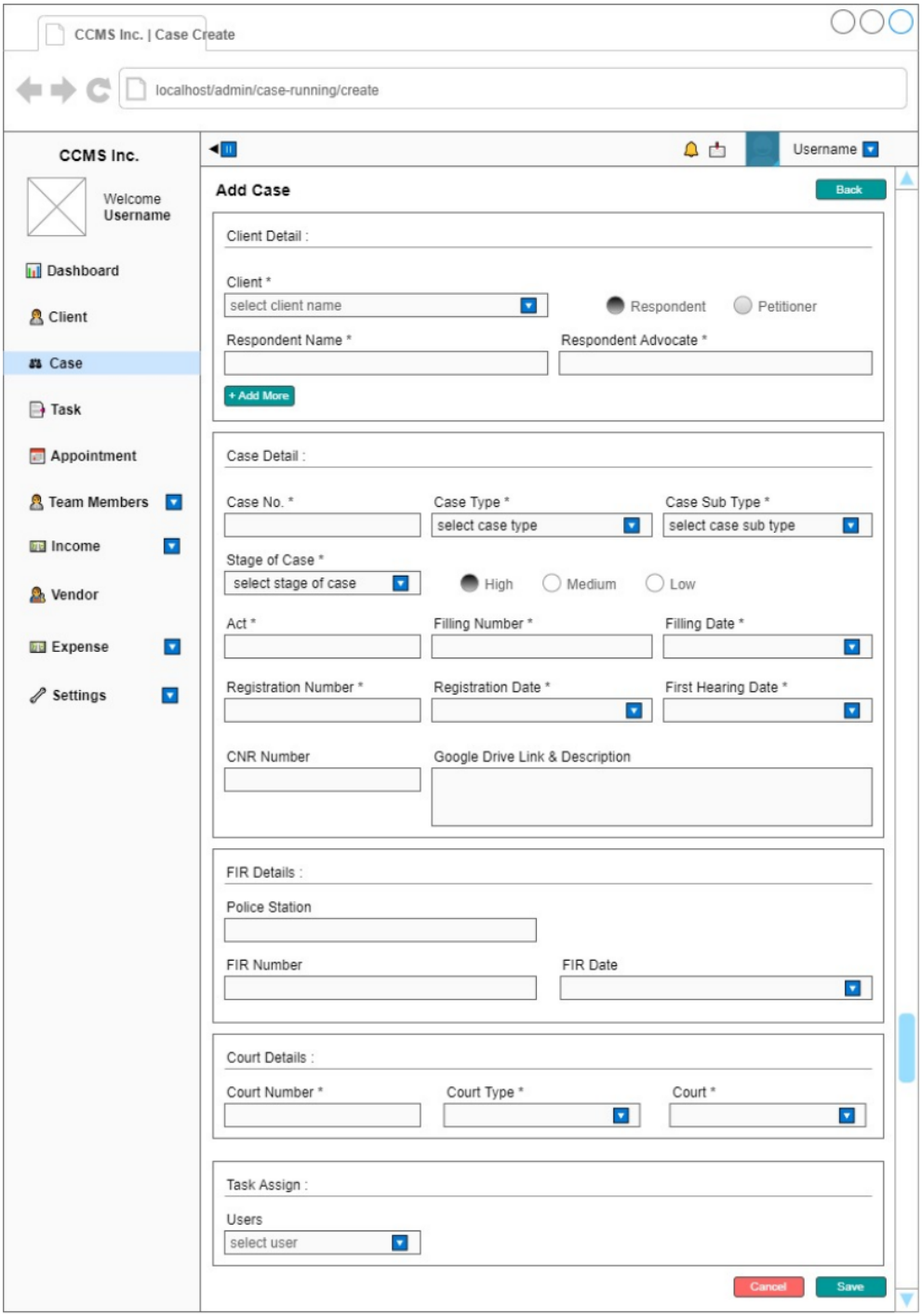


Figure 17. Wireframes & Mockups - Add Case Page

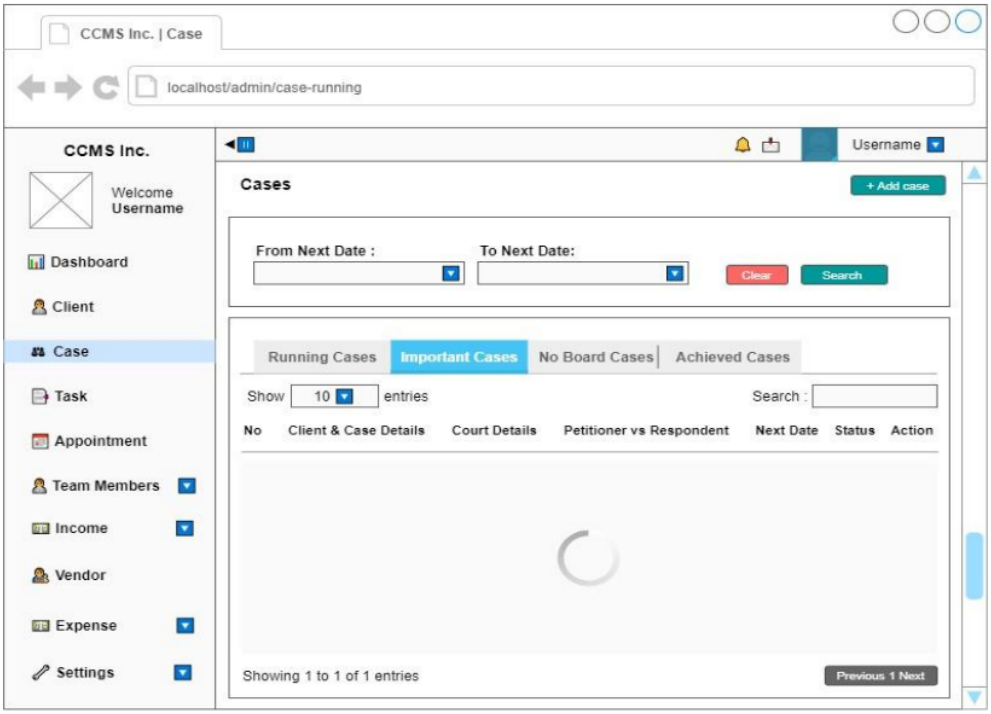


Figure 18. Wireframes & Mockups - Case List Page

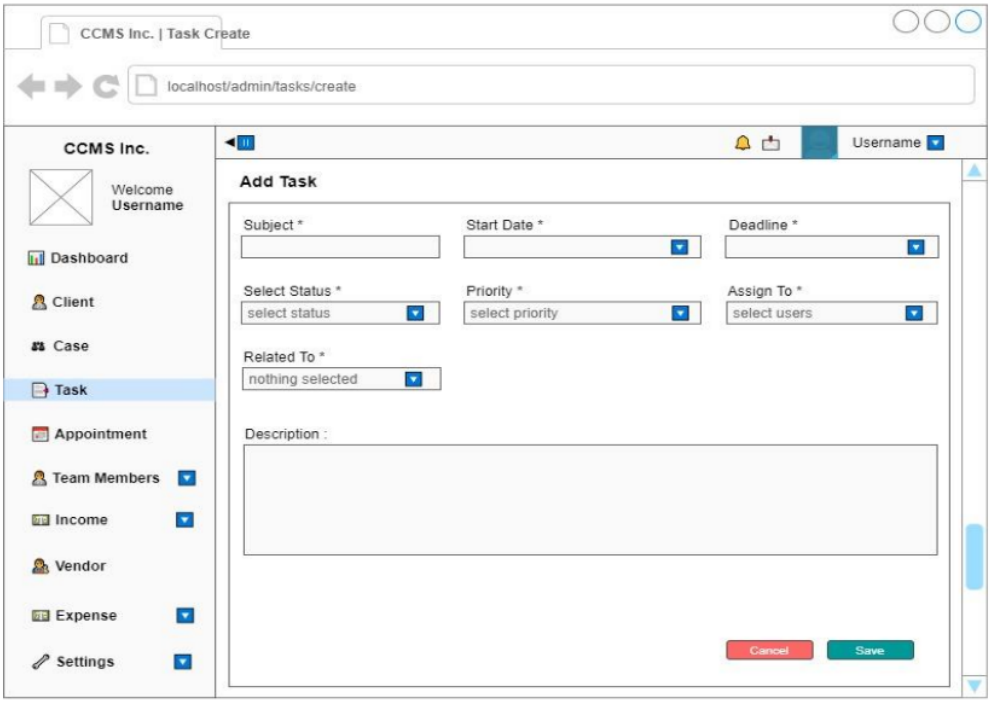


Figure 19. Wireframes & Mockups - Add Tasks Page

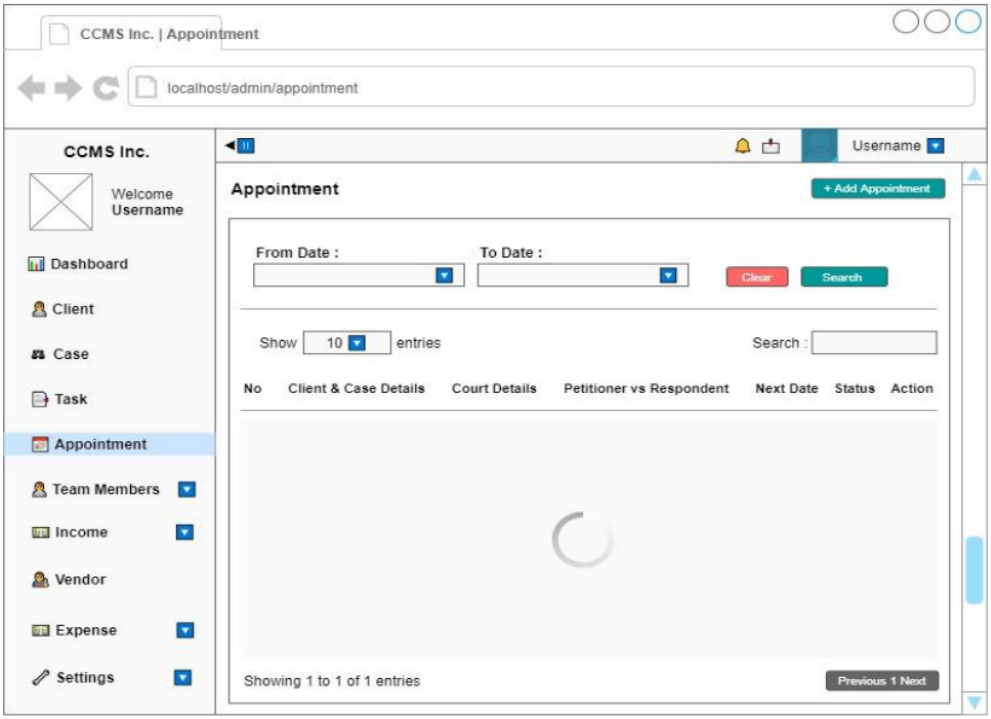


Figure 20. Wireframes & Mockups - Appointments List Page

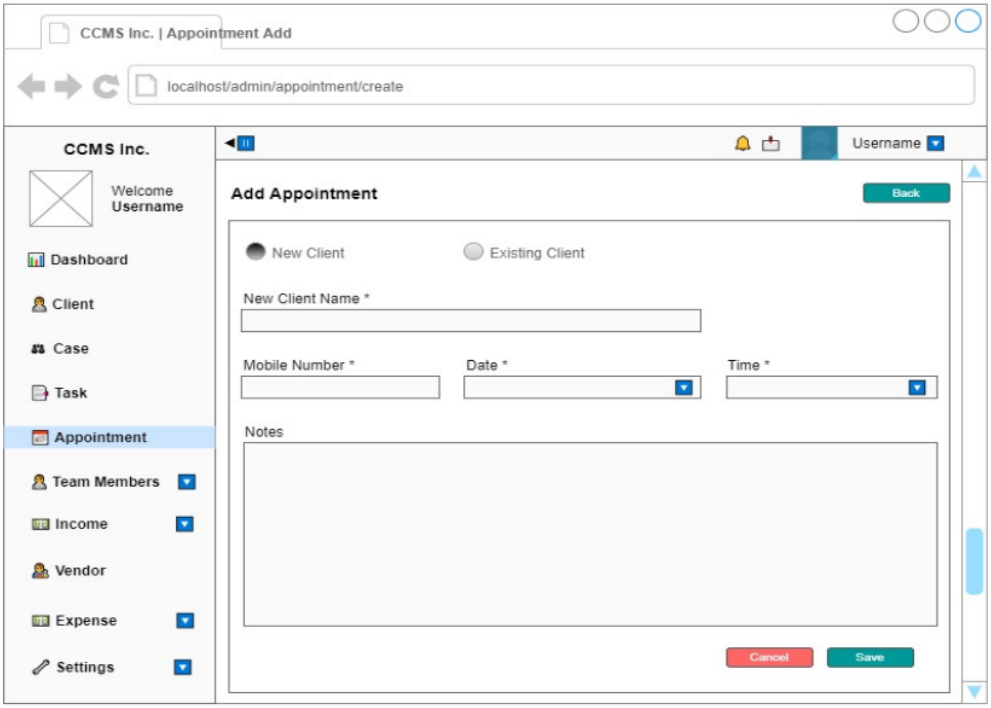


Figure 21. Wireframes & Mockups - Add Appointments Page

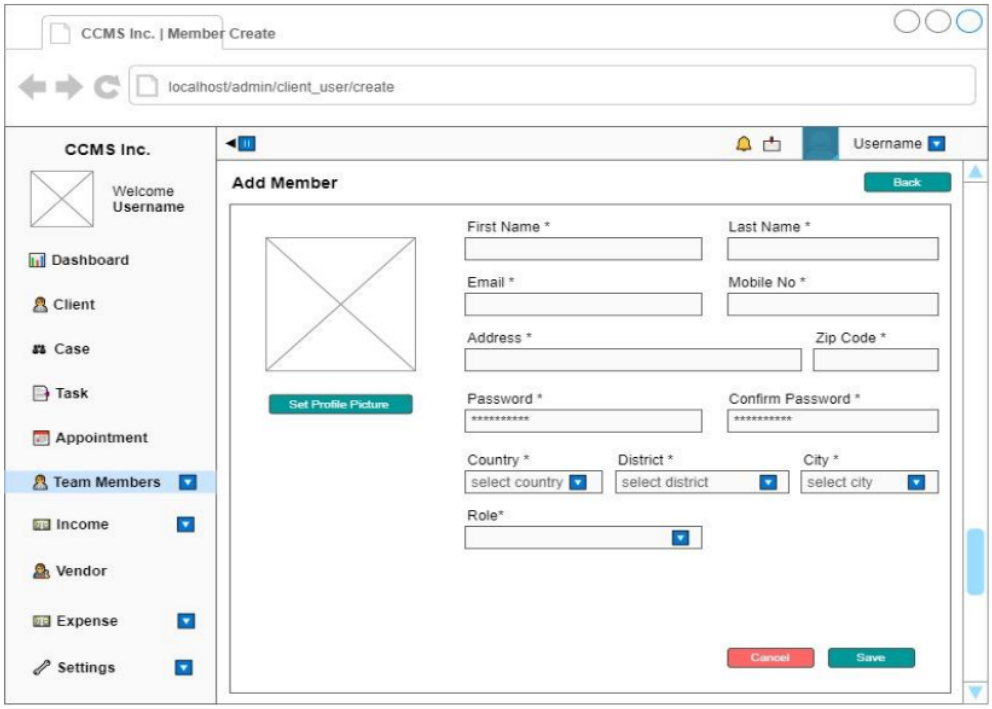


Figure 22. Wireframes & Mockups - Add Team Members Page

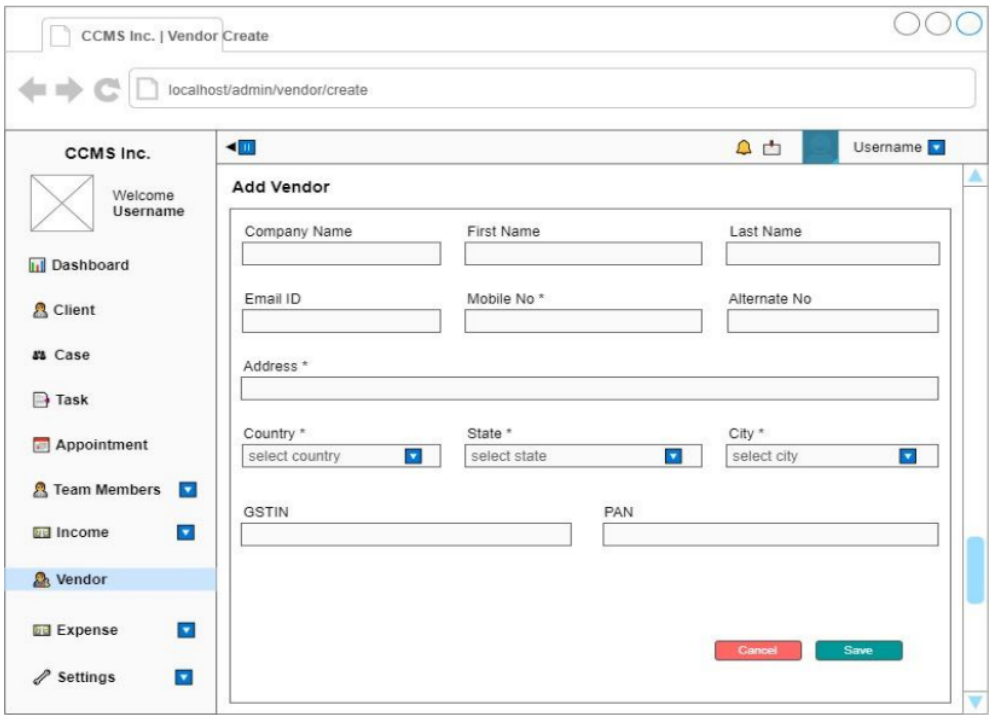


Figure 23. Wireframes & Mockups - Add Vendors Page

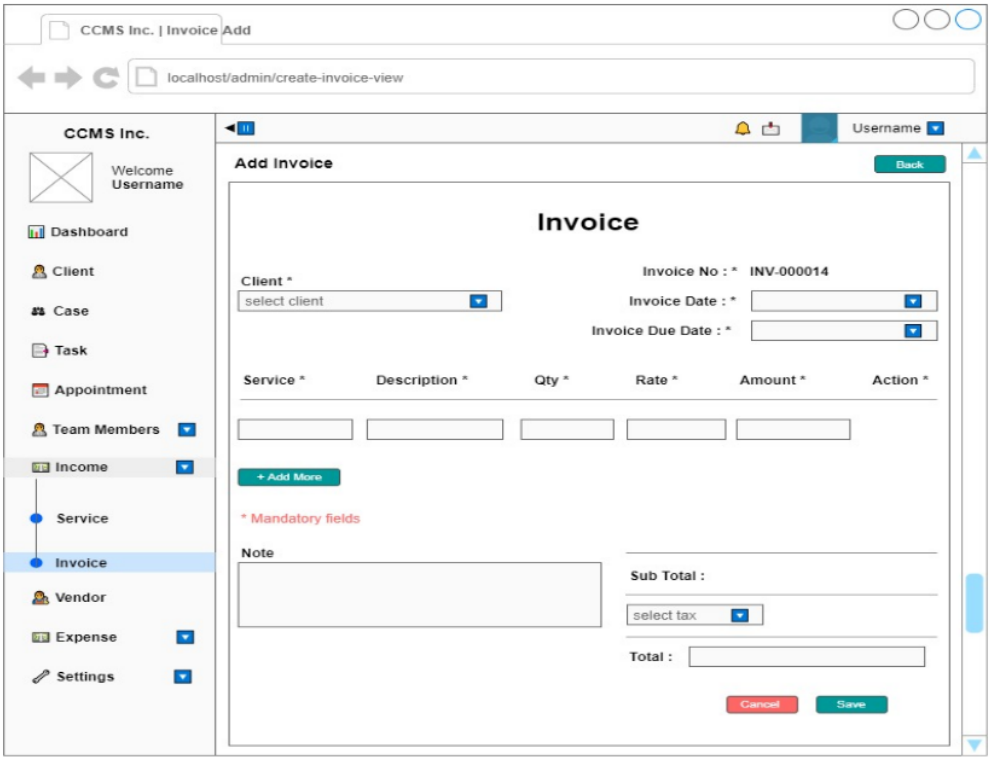


Figure 24. Wireframes & Mockups - Add Invoice Page

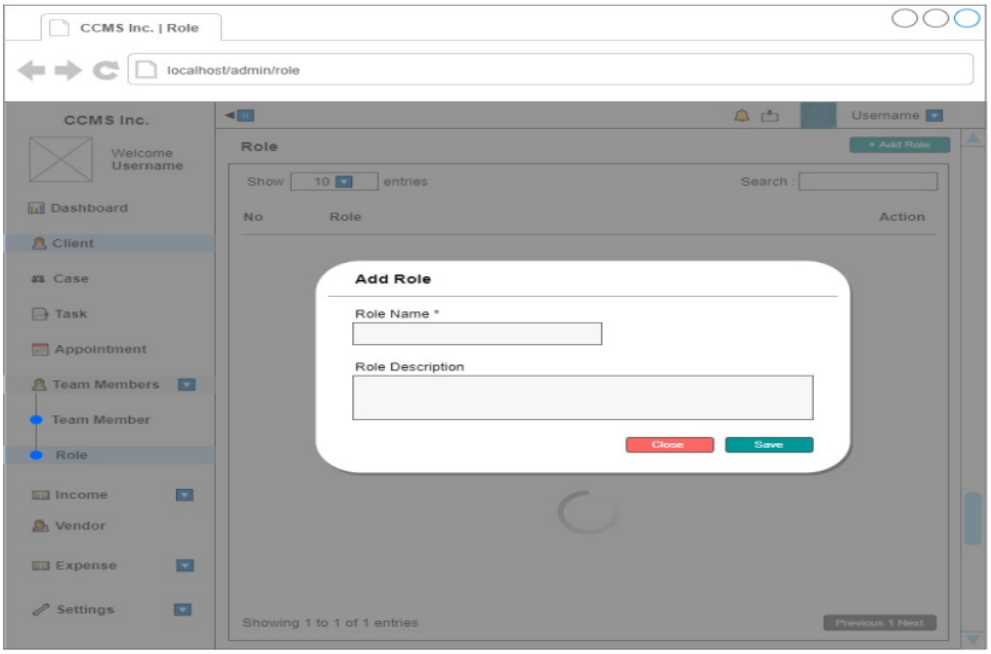


Figure 25. Wireframes & Mockups - Add Roles Page

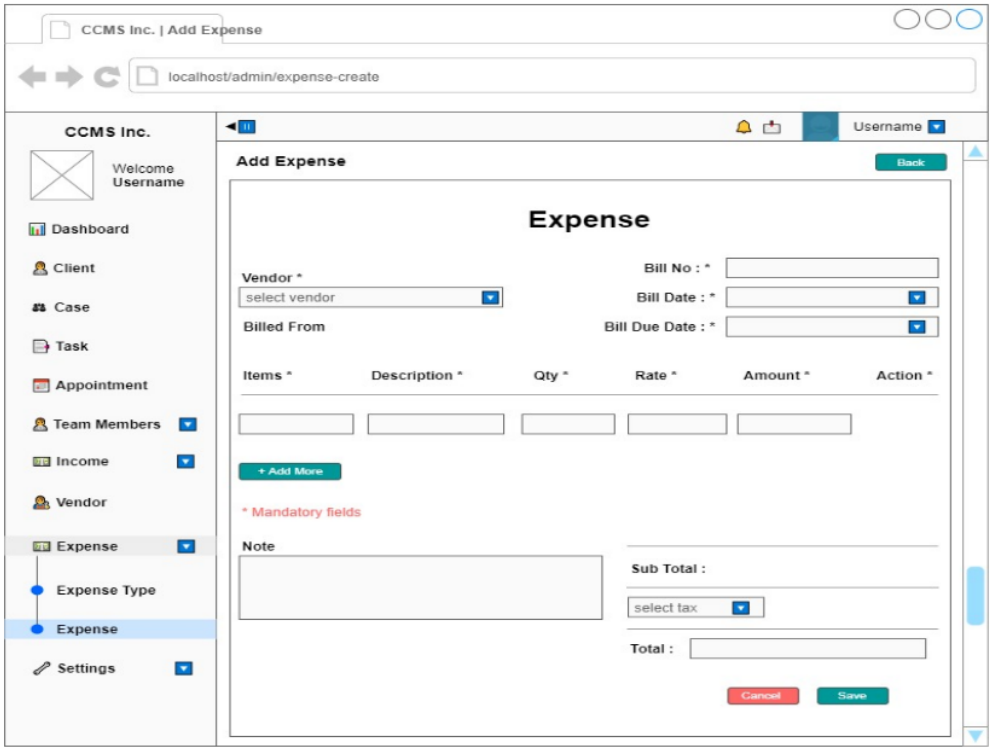


Figure 26. Wireframes & Mockups - Add Expenses Page

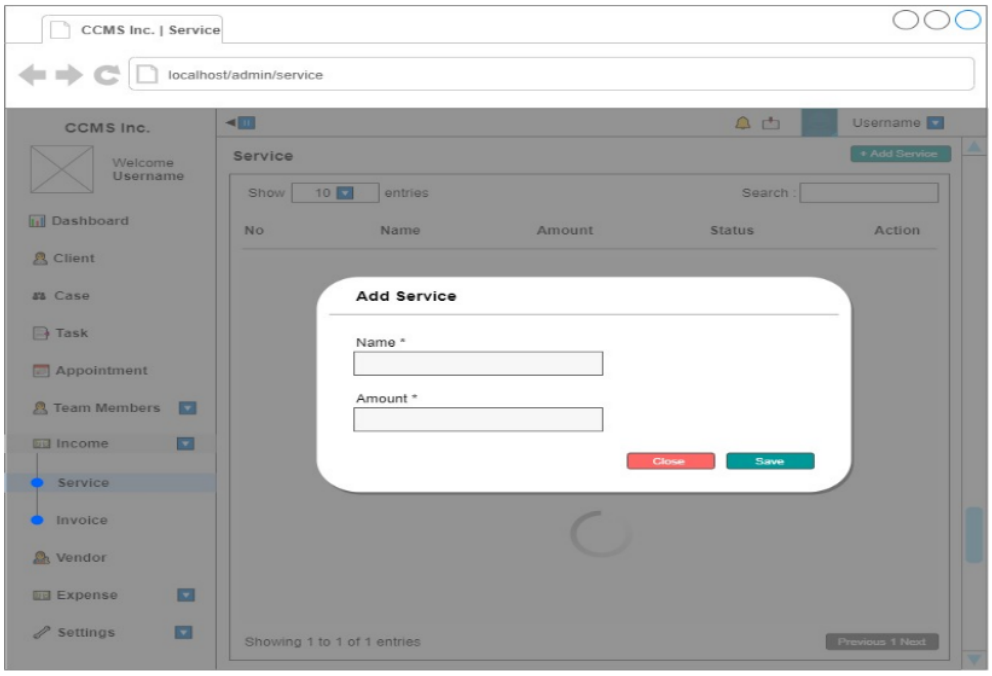


Figure 27. Wireframes & Mockups - Add Services Page

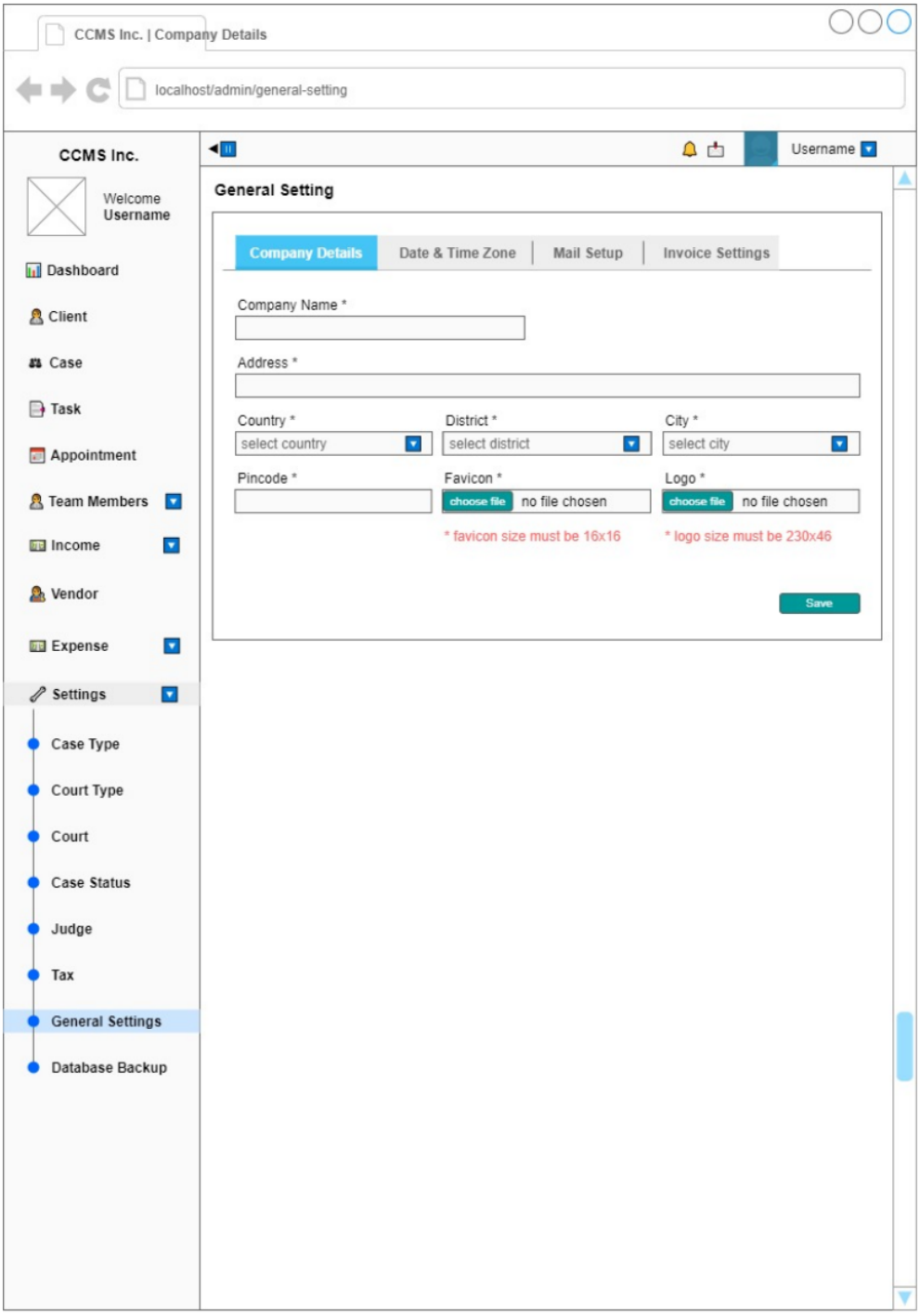


Figure 28. Wireframes & Mockups - Add Company Details Page

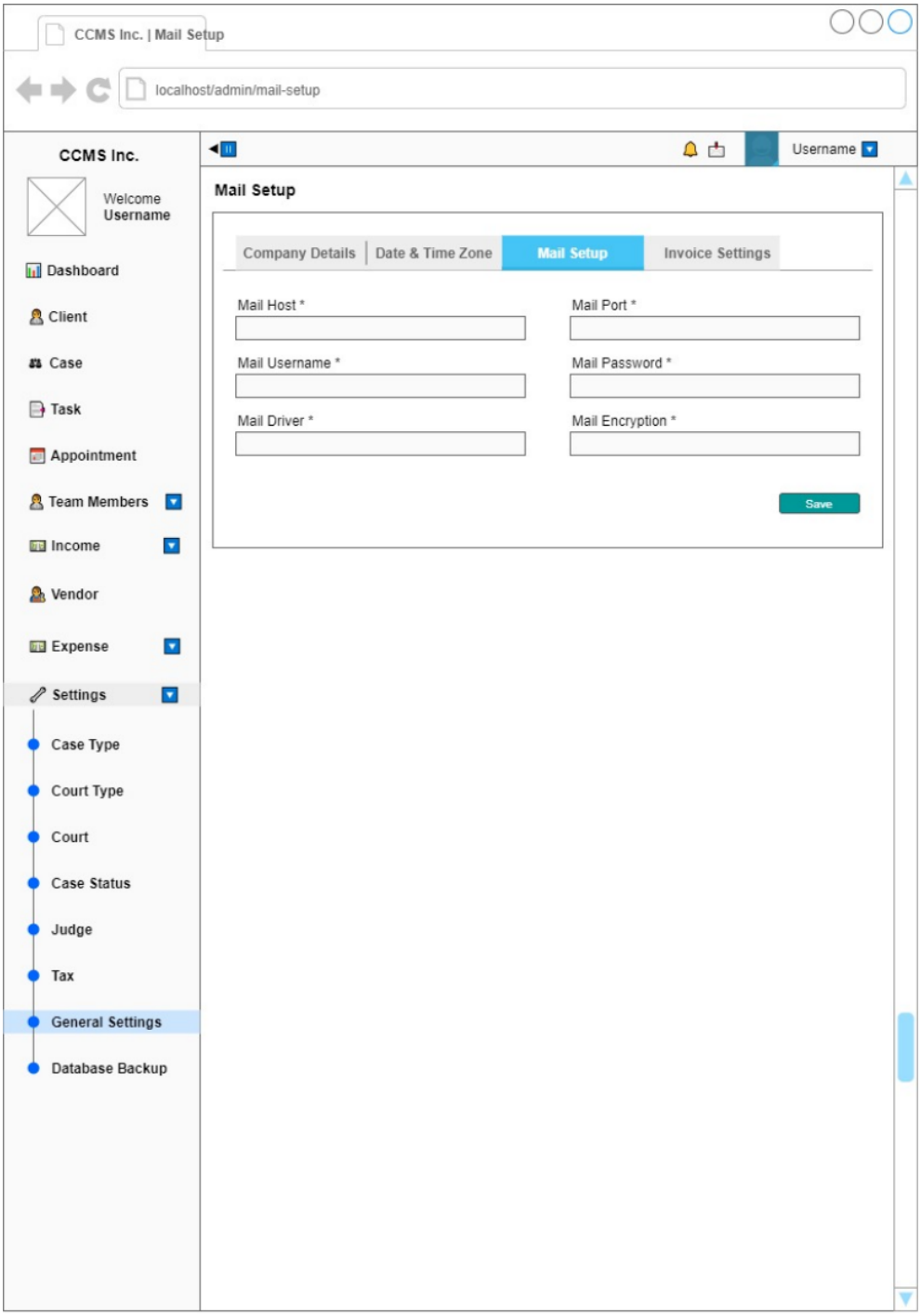


Figure 29. Wireframes & Mockups - Add Mail Settings Page

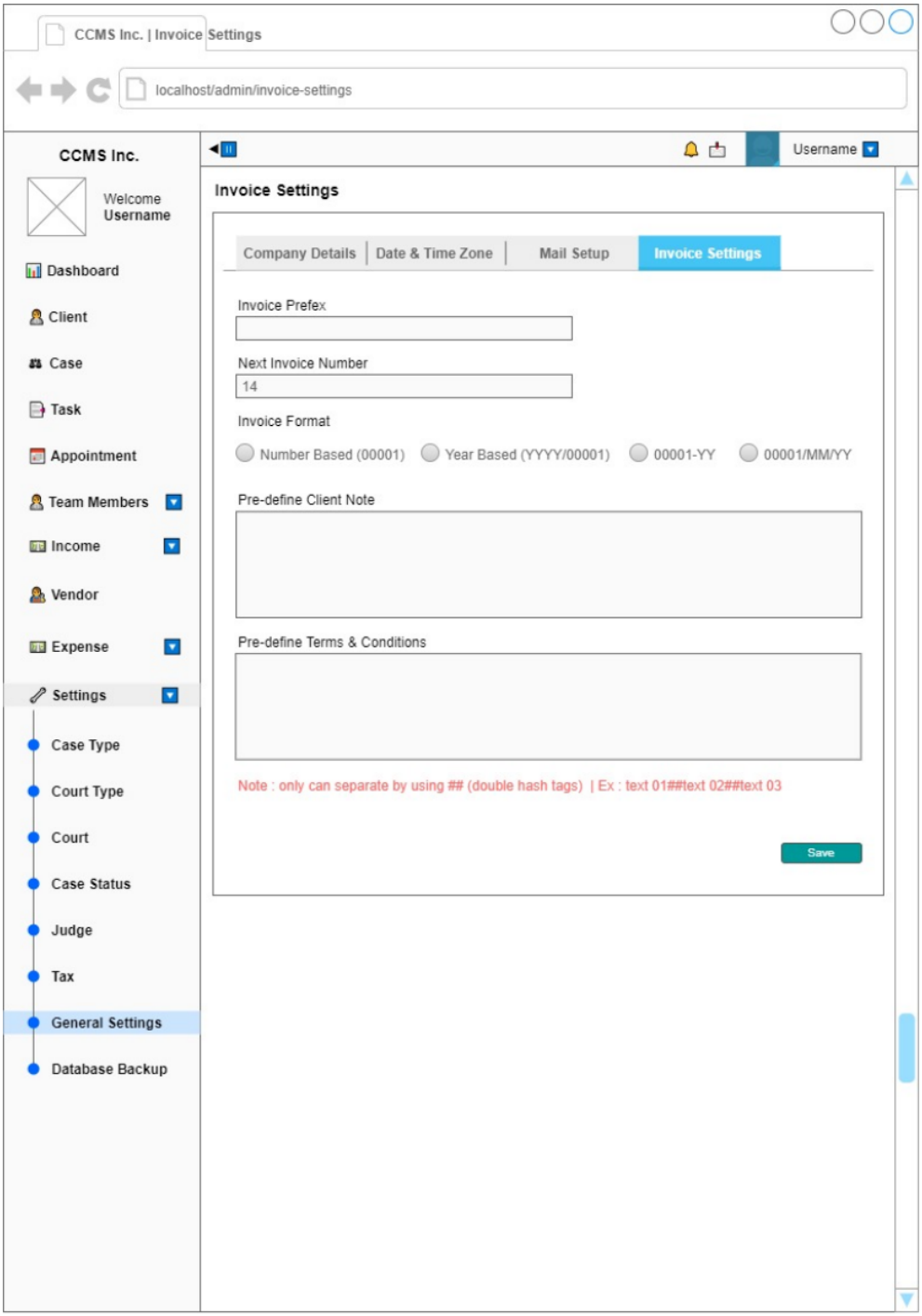


Figure 30. Wireframes & Mockups - Add Invoice Settings Page

Chapter 07: System Implementation

7.1 System Development Tools

- **Operating system - Windows 7 or Windows 10:** We need operating system to run all the required software and Microsoft Windows offers variety of resources for web development and design.
- **Web browser – Firefox browser developer edition:** Firefox web browser is a one of the fastest web browser we can choose. Firefox provide best developer tools latest features such as CSS grid support and framework support etc.
- **Offline Web server – XAMPP:** XAMPP is an open source free web server and it allows to run web applications offline, on a local web server on our computers. Apache web server, which is helped to build the local server, & MySQL database management system which we can use as a DB language for PHP web application.
- **Code editor and IDE – Microsoft VS Code and PHPStorm:** VS Code is a best code editor with support for development operations such as code debugging, building and editing. VS Code provide just the tools a developer needs for a quick development. PhpStorm supports to get around system code more efficiently & save time when working with PHP projects. Jump to a method & function or variable definition in just one click, or we search for its usages.
- **Version control software – GitHub Desktop:** GitHub Desktop is a software that enables to interact with GitHub using a GUI instead of the command prompt command or a web browser. Using this software we can push requests, pull requests, and also can clone remote repositories with the help of GitHub Desktop.

7.2 System Framework

Laravel is one of the best open-source PHP web framework, which is robust & easy to understand. Laravel follows model-view-controller design pattern. Laravel reuses the existing components of different frameworks which helps in creating a web application. The web application thus designed is more structured and pragmatic.

Why Laravel Chosen?

Laravel offers a rich set of functionalities which incorporates the basic features of PHP frameworks like CodeIgniter, Yii & other programming languages like Ruby on Rails. Laravel has a very rich set of features which will boost the speed of web development. This framework saves a lot time if we are planning to develop a system from scratch. Moreover, a web application built in Laravel is secure & prevents several web attacks.

Laravel MVC Architecture

Model: a place to store all business logic that deals with the actual data.

View: a place to store all design logics, like HTML, and interpolating data values.

Controller: a place to store all the code that allows the application to work (i.e. handles an incoming request, gets the data from the model, give it to the view, renders the view, outputs it to the web server)

Generally MVC is considered a "tightly-coupled" architecture, meaning that for each abstract domain object (like an account, a user, a product, etc.) we have a single model, a single controller, and a number of views. All requests start with the controller and end with the controller. It's in the name "controller". The view should never talk to the model, and the model should never talk to the view.

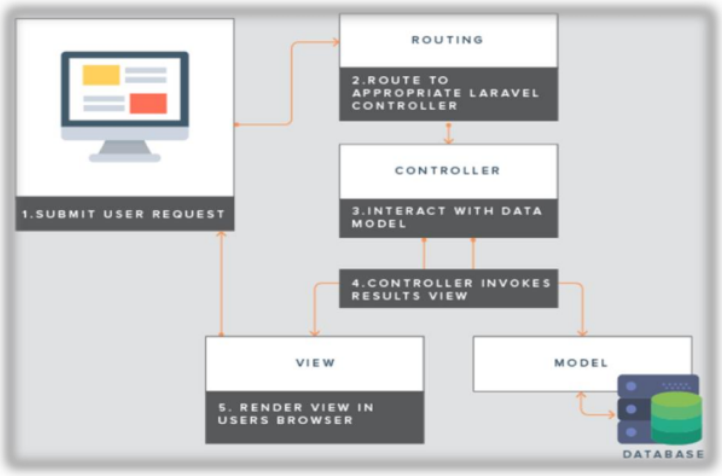


Figure 31. Laravel MVC Architecture

7.3 System User Interfaces

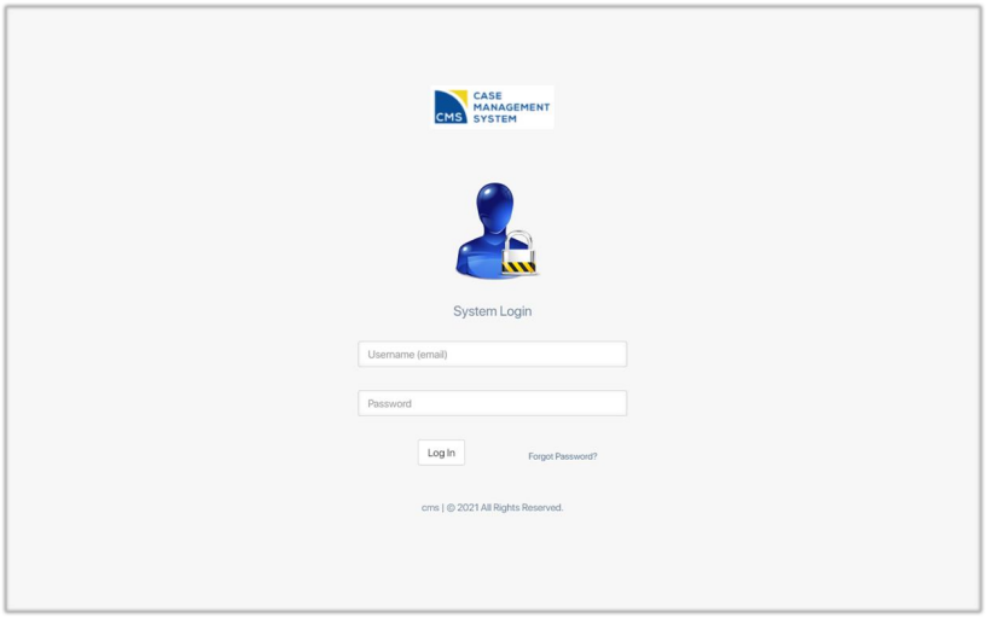


Figure 32. System Login Page

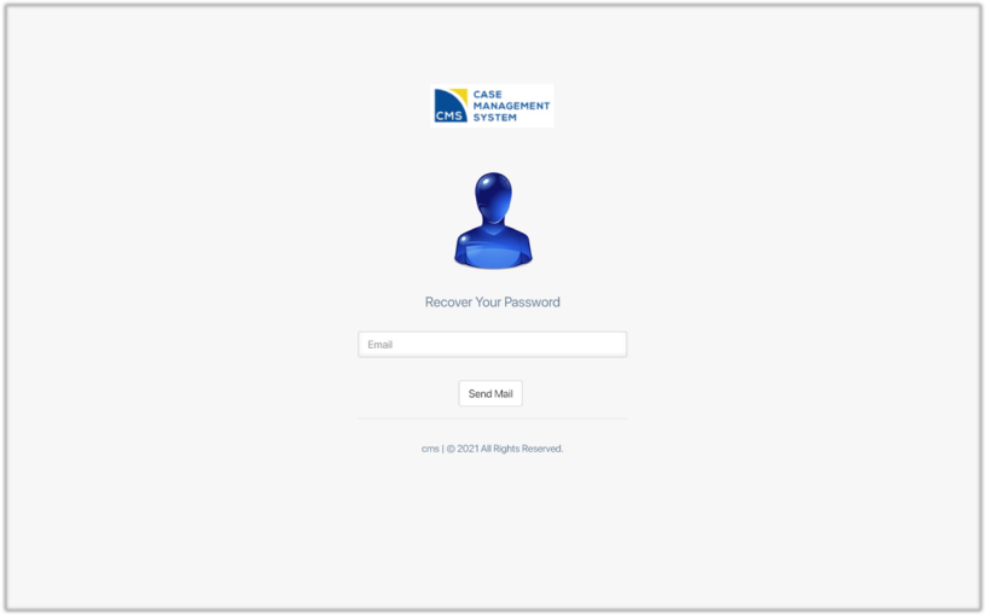


Figure 33. Password Recover Page

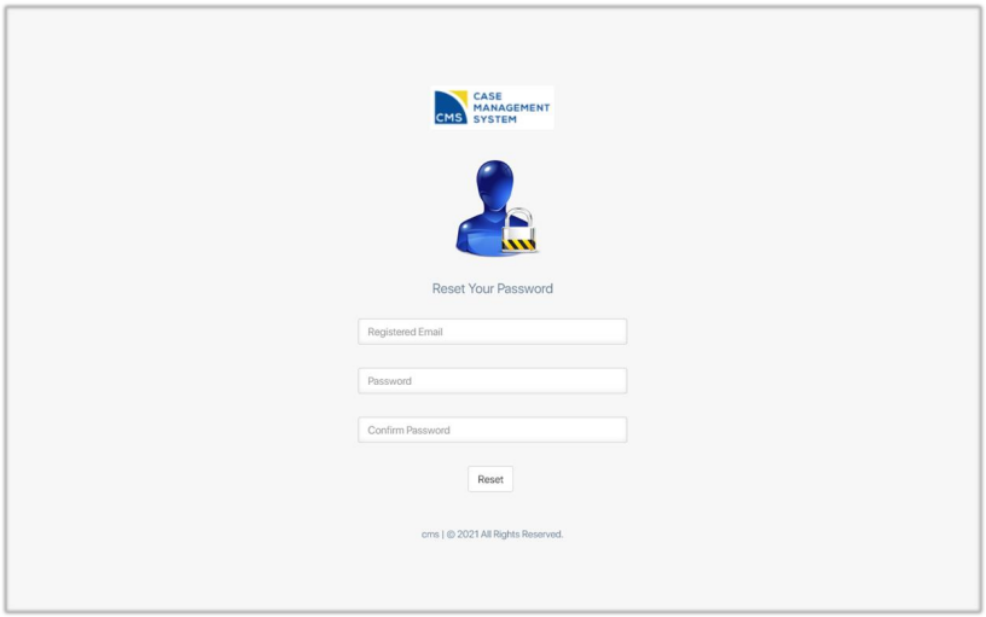


Figure 34. Password Reset Page

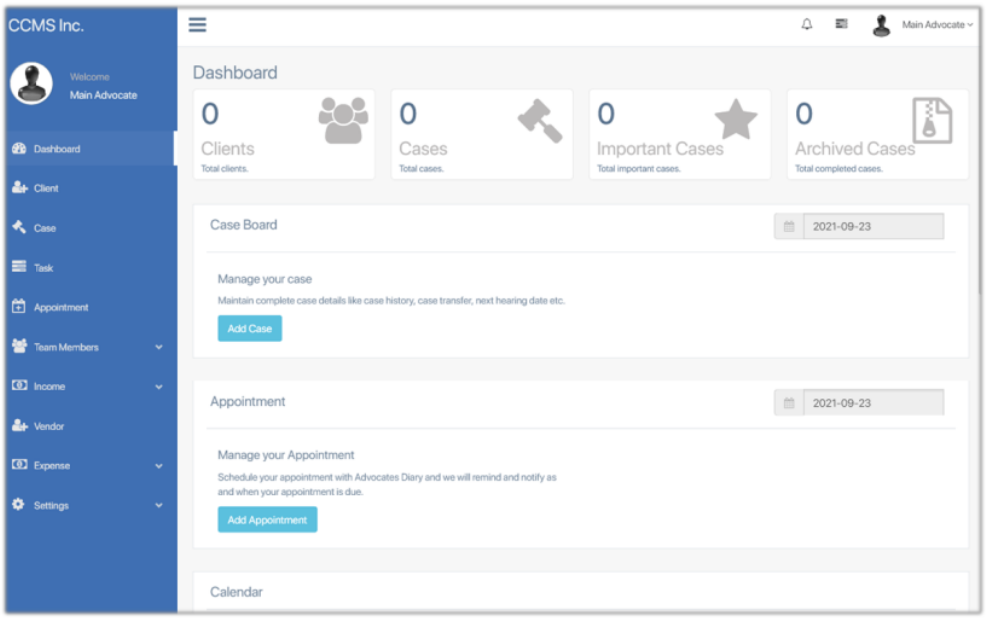


Figure 35. System Dashboard

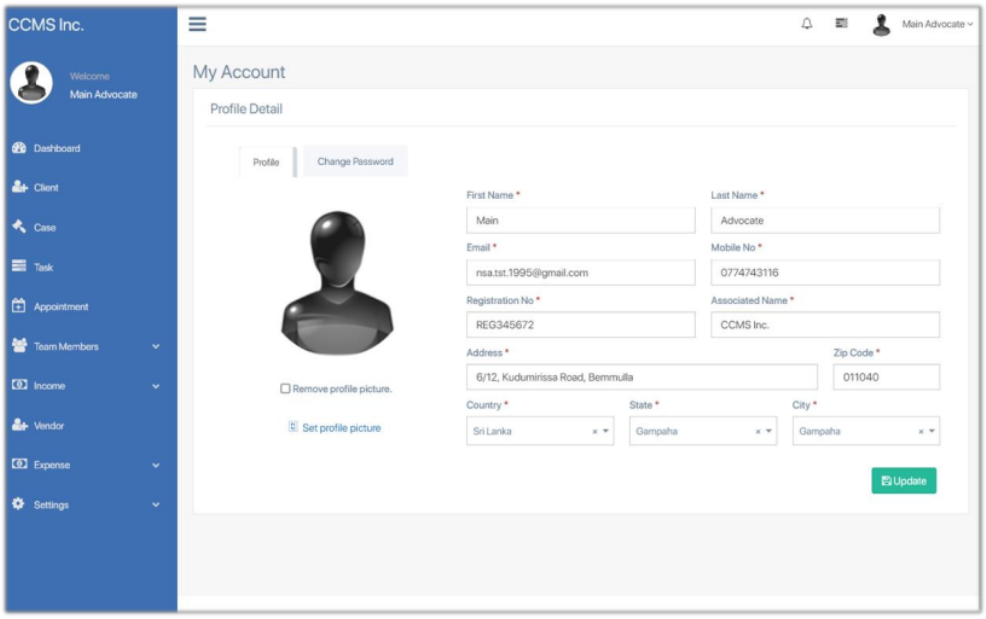


Figure 36. Admin Profile Update Page

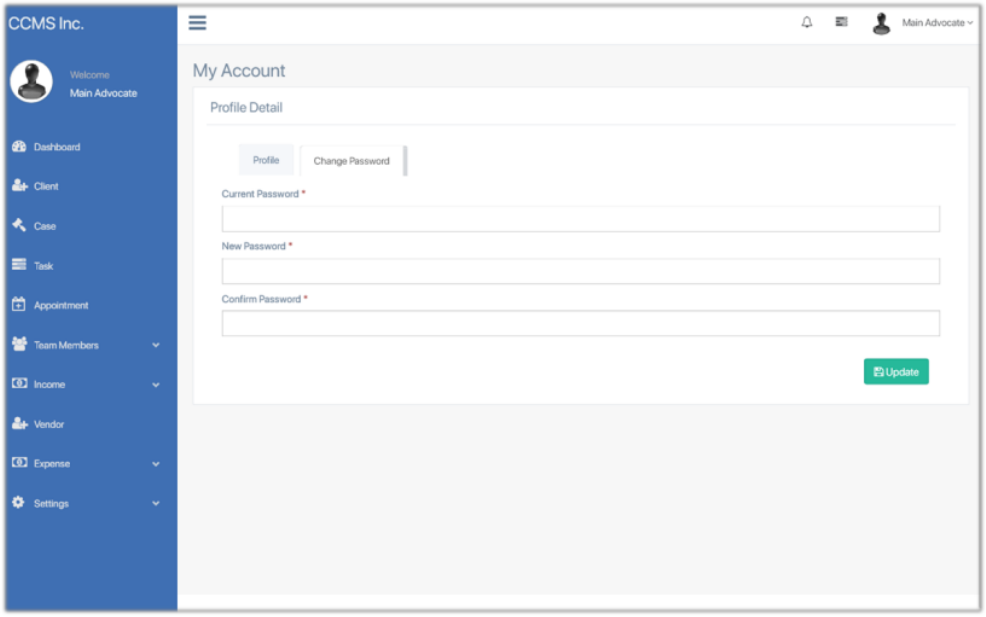


Figure 37. Password Change Page (Admin)

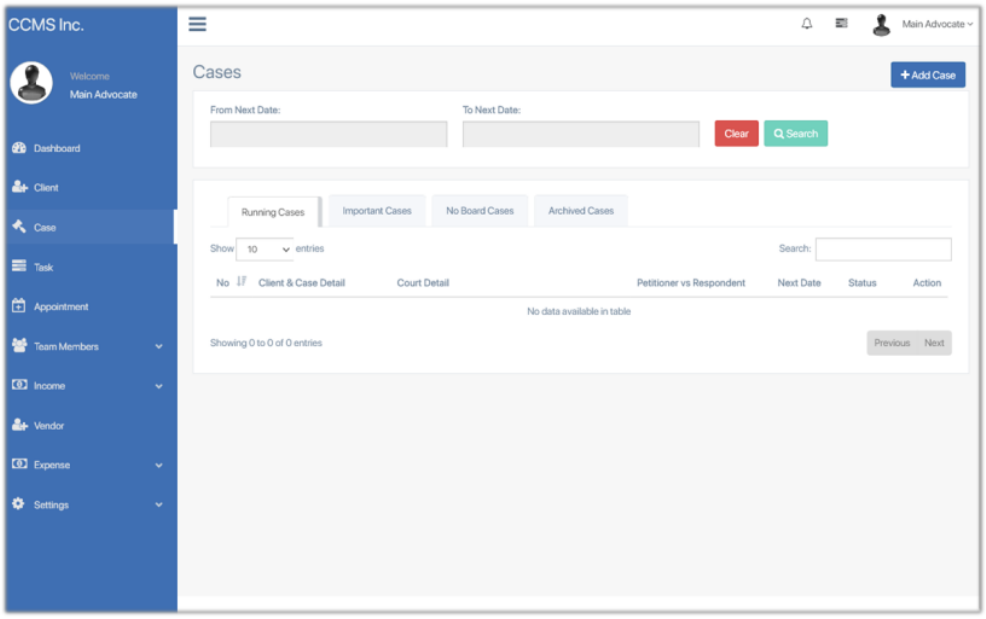


Figure 38. Cases List Page

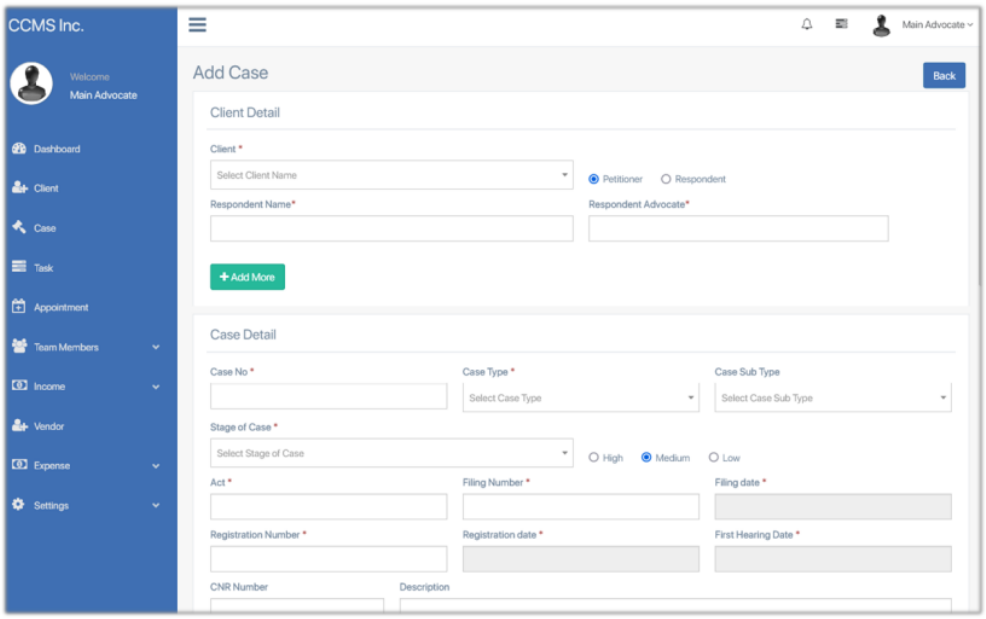


Figure 39. Add Cases Page

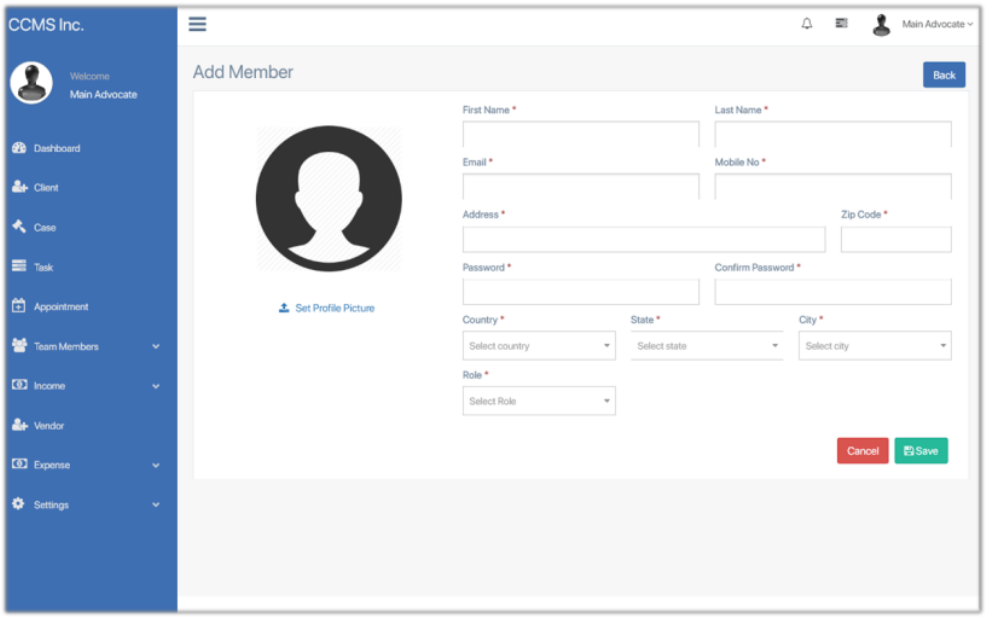


Figure 40. Add Team Members Page

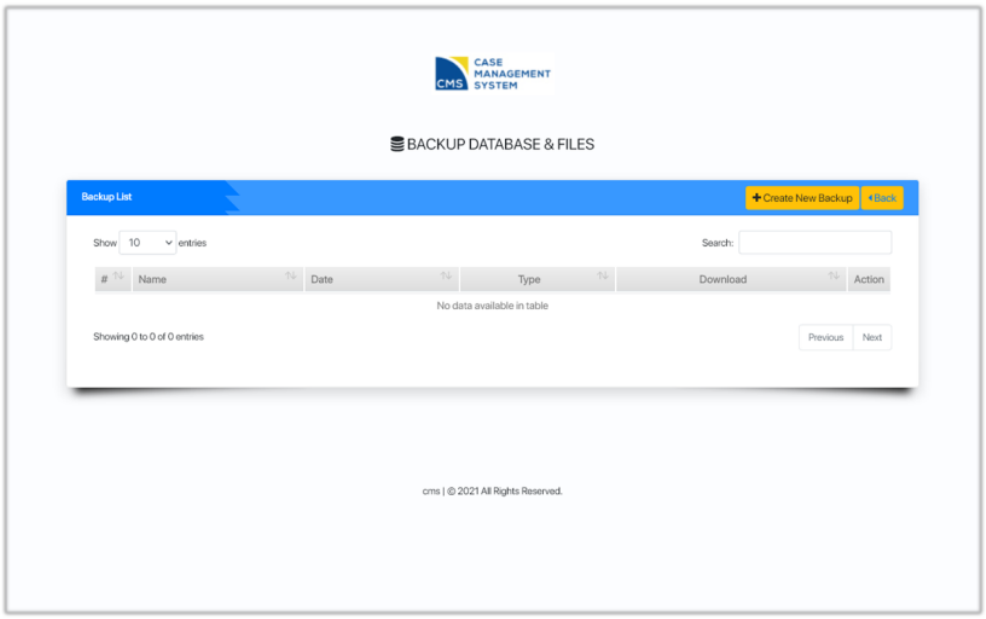


Figure 41. Database & Files Backup Page

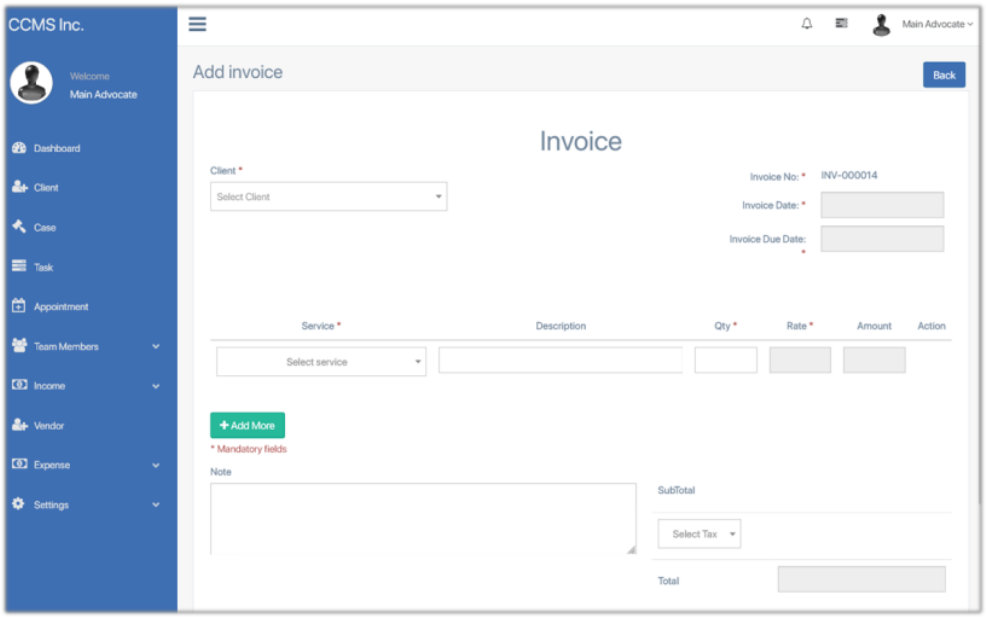


Figure 42. Add Invoices Page

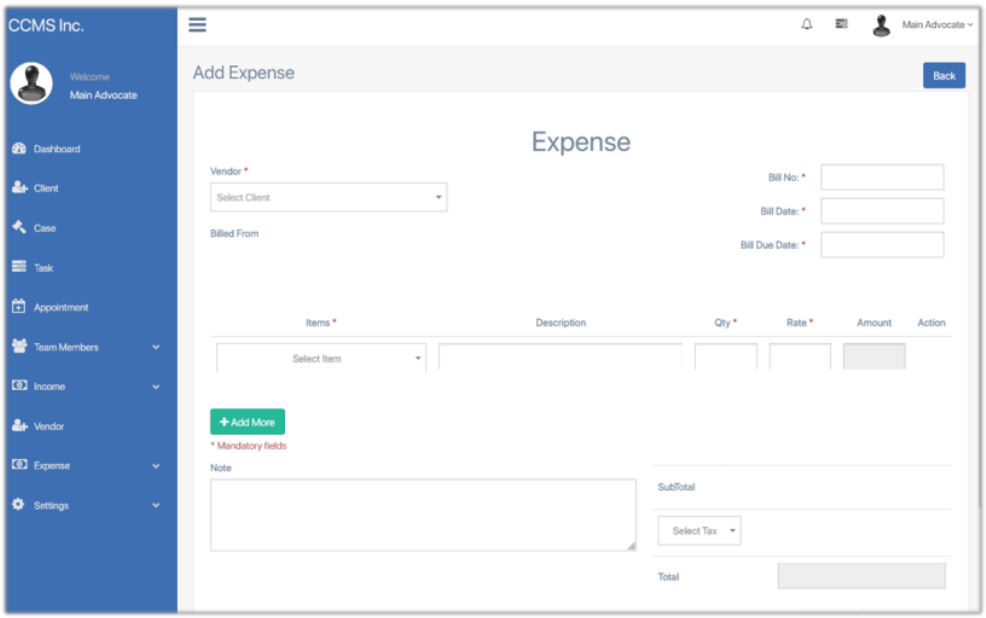


Figure 43. Add Expenses Page

7.4 Special Code Parts

```php
protected function schedule(Schedule $schedule)
{
    $schedule->command( command: 'reset:database')->daily();
    $schedule->command( command: 'backupmanager:create')->daily();
}
```

Figure 44. Backup Schedule Code

This protected schedule function will use to define the application' schedules. To setup or create automatic backups daily, we will use **backupmanager:create**. Although backup manager package provides GUI interfaces to manage backups, following commands also can use according to our needs.

- **backupmanager:create** to creates backup of files and/or database.
- **backupmanager:list** to shows list of backups taken.
- **backupmanager:restore** to restores a backup already taken.

```php
// define disk options
'disk' => 'local', // any disk from config/filesystems.php like local, ftp, s3, etc
'backup_path' => 'backups',

// backup files name suffix
'backup_file_date_suffix' => date( format: 'M-d-Y'),

// define number of days old backup files will be deleted before new backup
'delete_old_backup_days' => 10
```

Figure 45. Define Disk Options Code

By default backup manager package saves backups to **local** disk but we can use built-in feature of laravel filesystem to save backups to other disks too. Let's say we want to upload to different server for which we have ftp credentials, we need to update those ftp credentials into laravel's **config/filesystems.php** file under **ftp** disk setting. Once we have done that, in backup manager config file (**config/backupmanager.php**) specify our disk to be **ftp** instead of **local**.

How Restore is verified: even though there is no 100% way to verify restores, yet for files we create and verify restore feature by putting some contents into **backup-verify** file before and after restore. Similarly, we verify database restore by putting some contents into **verifybackup** table before and after restore. In both cases, contents of that file and database table are different at the time of backup and restore.

```
'disks' => [

    'local' => [
        'driver' => 'local',
        'root' => storage_path( path: 'app'),
    ],

    'public' => [
        'driver' => 'local',
        'root' => storage_path( path: 'app/public'),
        'url' => env( key: 'APP_URL').'/storage',
        'visibility' => 'public',
    ],

    's3' => [
        'driver' => 's3',
        'key' => env( key: 'AWS_ACCESS_KEY_ID'),
        'secret' => env( key: 'AWS_SECRET_ACCESS_KEY'),
        'region' => env( key: 'AWS_DEFAULT_REGION'),
        'bucket' => env( key: 'AWS_BUCKET'),
        'url' => env( key: 'AWS_URL'),
    ],

],
```

Figure 46. Configure File System Disks Code

Here we may configure as many filesystem "disks" as we wish, and we may even configure multiple disks of the same driver. Defaults have been setup for each driver as an example of the required options.

To run figure 45 and 46 codes, we need PHP >= 5.6, Laravel 5, **mysql** as database restore source, **mysqldump** as database backup source, *tar* as files backup and restore source, **zcat** to extract database achieve.

7.5 System Design Patterns

7.5.1 Facades Design Pattern

The Facade pattern is an object-oriented programming design pattern often utilized. A facade is a class that wraps a more complicated library to make it easier to use. It's also possible to utilize the Facade approach to offer a consistent and well-crafted API to a number of complicated and badly built APIs.

Essentially, a Laravel facade is a class that acts as a static front end to the container's internal services. The documentation claims that these facades act as a middleman for accessing the container's internal functionality.

Every service in the container has its own name, as we all know. There are two ways to get to a service in a Laravel application: either App::make() or the helper app().

Facade classes are used by Laravel to make services more accessible to developers. If we were to use a facade class, we would just have to write the following lines of code to achieve the same result: someService:: methodName();

All services in Laravel have a class called facade. These facade classes build on the Illuminate/Support package's basic Facade class. Only the getFacadeAccessor function, which returns the container's service name, needs to be implemented.

Facade is referred to as someService in the above syntax. methodName really refers to a method in the container's original service. Looking at this syntax outside of Laravel, we can see that there is a static method called methodName exposed by a class named someService ().

Importing Laravel facades is necessary since they are PHP classes. When we access a class by its fully-qualified name, PHP automatically loads it thanks to namespaces and autoloading support. The use directive in PHP allows you to alias classes. SomeServiceFacade:SomeMethod();

However, this must be done in every script that calls for a facade class of that kind. Laravel uses an alias loader to handle facade aliasing on its own.

How Laravel Aliases the Facades ?

The app.php configuration file, located in the /config directory, has an aliases array that stores all alias names. Each alias identifier is linked to a fully qualified class name, as can be seen by looking at the array. As a result, we're free to give our façade class any name we choose.

7.5.2 Command Design Pattern

The command design pattern is a very common pattern in Laravel, there are many examples that make use of the command pattern, for example we can think of keyboard clicks as commands, console commands, etc.

In the context of OOP the command pattern enable us to map certain class operations to commands. This enable us to decouble the invoker object from the object that implements the actual operation. Let's imagine that you have a class calculator, this class contains common calculator operation like (Addition, Subtraction, Multiplication, Division). Now to utilize the command pattern we have to make a class (command) for each operation so we will have **AddCommand**, **SubtractCommand**, and so on.

Command Pattern Key Components;

Receiver Class: This class contains the actual operations and their implementations.

Command Interface: This interface needs to be implemented by every concrete command class, and contains just one method execute which calls the receiver class operation.

Concrete Commands: Those classes implement the command interface execute method, for example the AddCommand in the calculator.

Command Invoker: This is an utility class to set commands and execute them.

7.6 Laravel REST API

Interoperability among computer systems on the internet may be achieved via the use of REST or RESTful web services. Requesting systems may use a consistent and specified set of stateless procedures to access and modify textual representations of online resources. Also in computer programming, an application programming interface (API) is a set of subroutine definitions, protocols and tools for building applications. A good API makes it easier to develop a program by providing all the building blocks, which are then put together by the programmer. Therefor a RESTful API is an application program interface that uses HTTP requests to GET, PUT, POST and DELETE data.

7.7 API Route

```php
Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});
```

Figure 47. API Route

7.8 Main Web Routes

```php
Route::group(['prefix' => 'admin'], function () {
Route::get('/login', 'AdminAuth\LoginController@showLoginForm')->name('login');
Route::post('/login', 'AdminAuth\LoginController@login');
Route::post('/logout', 'AdminAuth\LoginController@logout')->name('logout');

Route::get('/register', 'AdminAuth\RegisterController@showRegistrationForm')->name('regist
Route::post('/register', 'AdminAuth\RegisterController@register');

Route::post('/password/email', 'AdminAuth\ForgotPasswordController@sendResetLinkEmail')->n
Route::post('/password/reset', 'AdminAuth\ResetPasswordController@reset')->name('password.
Route::get('/password/reset', 'AdminAuth\ForgotPasswordController@showLinkRequestForm')->n
Route::get('/password/reset/{token}', 'AdminAuth\ResetPasswordController@showResetForm');
```

Figure 48. Admin Authentication Web Routes

```
//Dashboard
Route::resource('/dashboard', 'DashBordController');
Route::post('/dashboard', 'DashBordController@index');
Route::get('/ajaxCalander', 'DashBordController@ajaxCalander');
Route::post('dashboard-all-caseList', 'DashBordController@dashboardAllCaseList');
Route::post('dashboard-appointment-list', 'DashBordController@appointmentList')->n
Route::get('downloadCaseBoard/{date}', 'DashBordController@downloadCaseBoard');
Route::get('printCaseBoard/{date}', 'DashBordController@printCaseBoard');
```

Figure 49. Dashboard Web Routes

```
Route::resource('clients', 'ClientController');
Route::post('clients/data-list', 'ClientController@ClientList')->name('clients.list');
Route::post('clients/data-status', 'ClientController@changeStatus')->name('clients.stat
Route::post('check_client_email_exits', 'ClientController@check_client_email_exits')->n
Route::get('client/case-list/{id}', 'ClientController@caseDetail')->name('clients.case-
Route::get('client/account-list/{id}', 'ClientController@AccountDetail')->name('clients
```

Figure 50. Client Web Routes

```
//--------------------------Case Running-----------------//
Route::resource('case-running', 'CaseRunningController');
Route::post('allCaseList', 'CaseRunningController@allCaseList');
Route::get('select2Case', 'CaseRunningController@select2Case')->name('select2Case');
Route::get('case-list/{id}', 'CaseRunningController@caseListByClientId');
Route::post('client/client_case_list', 'CaseRunningController@client_case_list')->name(
Route::post('allCaseList', 'CaseRunningController@allCaseList');
Route::get('/case-nb', 'CaseRunningController@caseNB');
Route::get('/case-important', 'CaseRunningController@caseImportant');
Route::get('/case-archived', 'CaseRunningController@caseArchived');
Route::post('allCaseHistoryList', 'CaseRunningController@allCaseHistoryList');
Route::get('addNextDate/{case_id}', 'CaseRunningController@addNextDate');
Route::get('restoreCase/{case_id}', 'CaseRunningController@restoreCase');
Route::post('case-next-date', 'CaseRunningController@caseNextDate');
Route::get('/getNextDateModal/{case_id}', 'CaseRunningController@getNextDateModal')->nam
Route::get('/getChangeCourtModal/{case_id}', 'CaseRunningController@getChangeCourtModal
Route::get('/case-history/{case_id}', 'CaseRunningController@caseHistory');
Route::get('/case-transfer/{case_id}', 'CaseRunningController@caseTransfer');
Route::get('/getCaseImportantModal/{case_id}', 'CaseRunningController@getCaseImportantMo
Route::post('allCaseTransferList', 'CaseRunningController@allCaseTransferList');
Route::post('changeCasePriority', 'CaseRunningController@changeCasePriority');
Route::post('transferCaseCourt', 'CaseRunningController@transferCaseCourt');
Route::get('case-running-download/{id}/{action}', 'CaseRunningController@downloadPdf');
```

Figure 51. Case Running Web Routes

7.9 Version Controlling

In this project I have used Git as a version control system and git helped me to track and manage my source code. I have used GitHub as a cloud-based hosting service and GitHub let me manage my project git repository. These are the steps that I have used for software version control.

Create a repository on GitHub : As the first step of version control I made a GitHub repository for my project. This repository will contain all the project directory and files that I have created.

Initializing the Git repository : I initialized my Git repository to make my project directory officially a Git repository.

Adding new files to Git repository : I added all my project files to the repository using Git commands.

Make Git commitments : I committed my current files in the bashed directory. There will be more commitments that are based on source code changes and upgrades.

Reviewing changes : I reviewed every change that I have made after every commitment that I made.

Chapter 08: Testing and Validation

8.1 Introduction

Laravel apps comes with built-in testing platform called PHPUnit. In fact, support for testing with PHPUnit is included a phpunit.xml file that contains the application testing configurations. The system's tests directory holds two directories which are Feature & Unit. Unit tests are tests that focus on a very small, isolated portion of application code. In fact, most unit tests probably focus on a single method.

System feature tests, test a larger portion of system code, including how several objects interact with each other or even a full HTTP request to a JSON endpoint. Generally, most of system tests are feature tests. When running tests, Laravel will automatically set the configuration environment to testing because of the environment variables defined in the phpunit.xml file.

Laravel also automatically configures the session and cache to the array driver while testing, meaning no session or cache data will be persisted while testing.

Moreover, in this project test cases testing using PHPUnit package and test case files created by using Laravel Dusk pakage. Testing results will be output by PhpStorm as a HTML file.

8.2 Testing Tools and Technologies

8.2.1 PHPUnit Testing Framework

PHPUnit is the unit testing framework that used for run system testings. It is an example of the xUnit architecture for testing frameworks that invented with SUnit framework. We can find mistakes in system committed code quickly and assert that no code regression has occurred in other parts of the code base. Much like other unit testing frameworks, PHPUnit uses assertions to verify that the behavior of the specific component.

8.2.2 Laravel Dusk

Dusk is based on the open source tools ChromeDriver and Facebook Php-webdriver which makes it simple to use without the need to experience the intense procedure of setting up Selenium. One of Dusk features also inlcudes the ability to wait for a condition to be true at the frontend, before executing tests. For example it could wait for a JavaScript component or CSS selector to load before taking any action. In this project we used Laravel dusk to generate test case files and run those test cases using PHPUnit framework.

8.2.3 PHPStorm Test Result Export Tool

This tool helps to export test results generated by Laravel Dusk as a HTML file or XML file. We can specify the target filename (file name), and directory (folder) to export test outputs.

8.3 Test Plan

Table 8. Test Plan

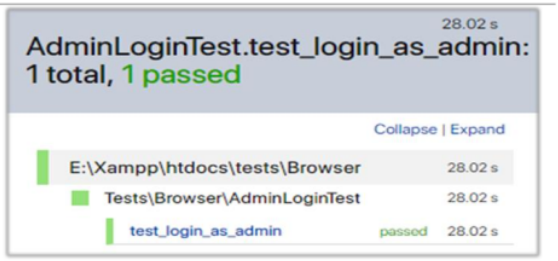
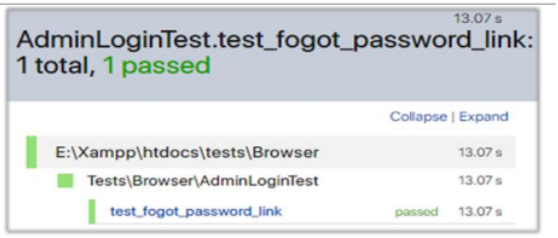
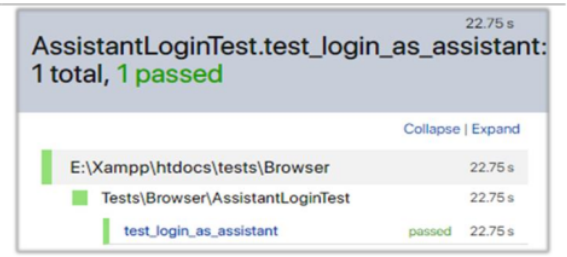
Case ID	Test Name	Test Steps
TC01	Login page view test	1. Assign to visit http://localhost/admin/login 2. Assert to see 'System Login'
TC02	Password recover page view test	1. Assign to visit http://localhost/admin/password/reset 2. Assert to see 'Recover Your Password'
TC03	Database backup page view test	1. Assign to visit http://localhost/backupmanager 2. Assert to see 'BACKUP DATABASE & FILES'
TC04	Admin login test	1. Assign to visit http://localhost/admin/login 2. Type correct admin username (email) 3. Type correct admin password 4. Press <login> button 5. Assert path http://localhost/admin/dashboard
TC05	Forgot password link test	1. Assign to visit http://localhost/admin/login 2. Click on <Forgot Password?> link 3. Assert path http://localhost/admin/password/reset
TC06	Assistant login test	1. Assign to visit http://localhost/admin/login 2. Type correct assistant username (email) 3. Type correct assistant password 4. Press <login> button 5. Assert path http://localhost/admin/dashboard
TC07	Accountant login test	1. Assign to visit http://localhost/admin/login 2. Type correct accountant username (email) 3. Type accountant correct admin password 4. Press <login> button

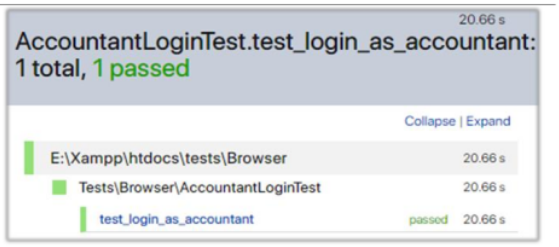
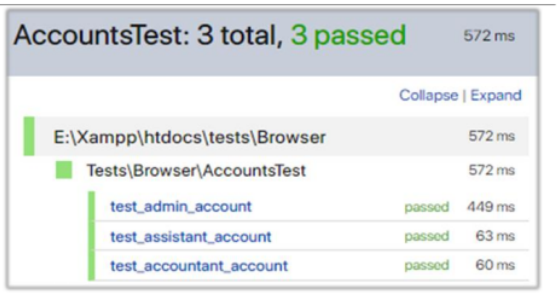
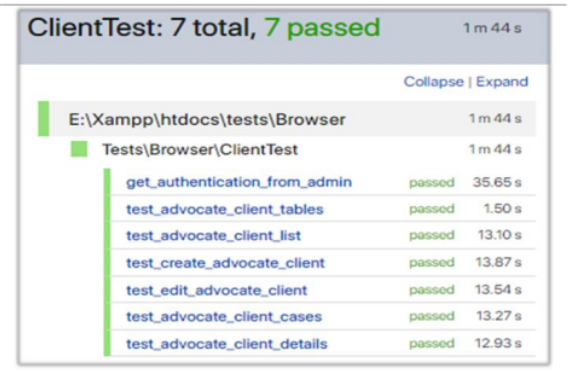
		5. Assert path http://localhost/admin/dashboard
TC08	Admin account test	1. Assert database has admins table 2. Assign admin's email, password, user type 3. Check whether admin account active or not
TC09	Assistant account test	1. Assert database has admins table 2. Assign assistant's email, password, user type 3. Check whether assistant account active or not
TC10	Accountant account test	1. Assert database has admins database table 2. Assign accountant's email, password, user type 3. Check whether accountant account active or not
TC11	Client section test	1. Test advocate_clients cms database table 2. Test client section
TC12	Case section test	1. Test case_logs, case_parties_involves, case_transfers cms database tables 2.Test case section
TC13	Task section test	1. Test tasks, task_members database tables 2. Test task section
TC14	Appointment section test	1. Test appointments cms database datable 2. Test appointment section
TC15	Members test	1. Test task_members cms database table 2. Test members section
TC16	Members roles test	1. Test roles cms database table 2. Test members roles section
TC17	Advocate services test	1. Test services cms database table 2. Test advocate services
TC18	Advocate invoice test	1. Test invoices cms database table 2. Test advocate invoices
TC19	Vendors section test	1. Test vendors cms database table 2. Test vendors section
TC20	Expense test	1. Test advocate expenses cms database table

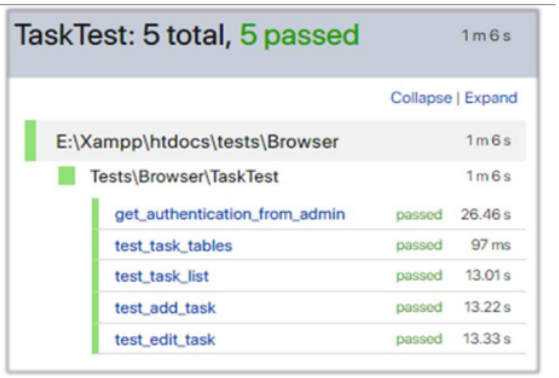
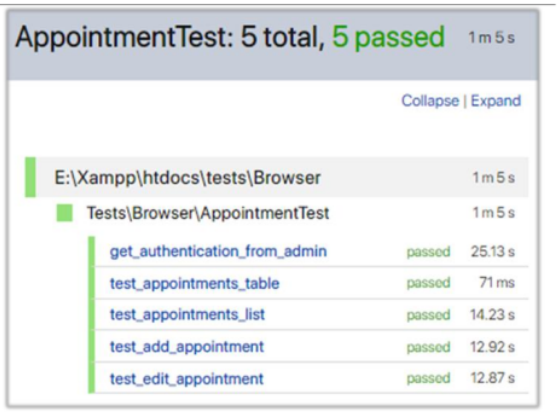
		2. Test advocate expense
TC21	Expense types test	1. Test expenses_items cms database table 2. Test expense types
TC22	System settings test	1. Test all the system settings tables at database 2. Test system settings section

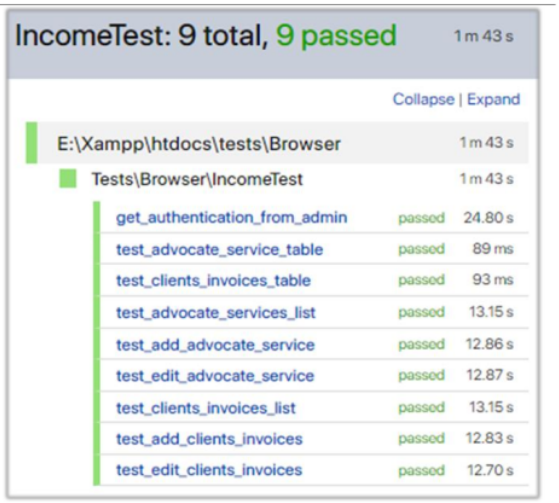
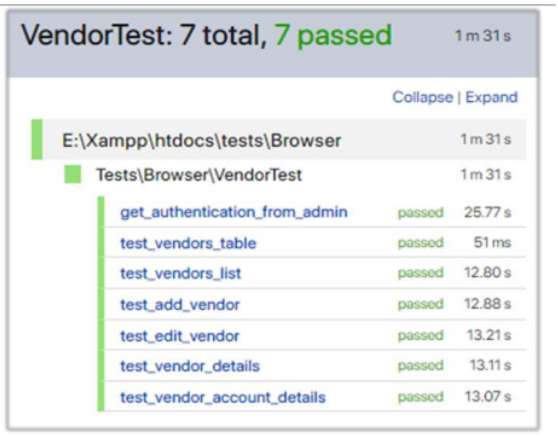
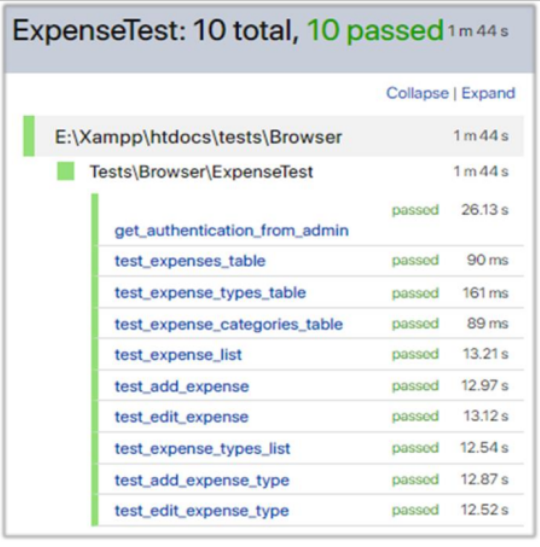
8.4 Automated Test Cases and Results

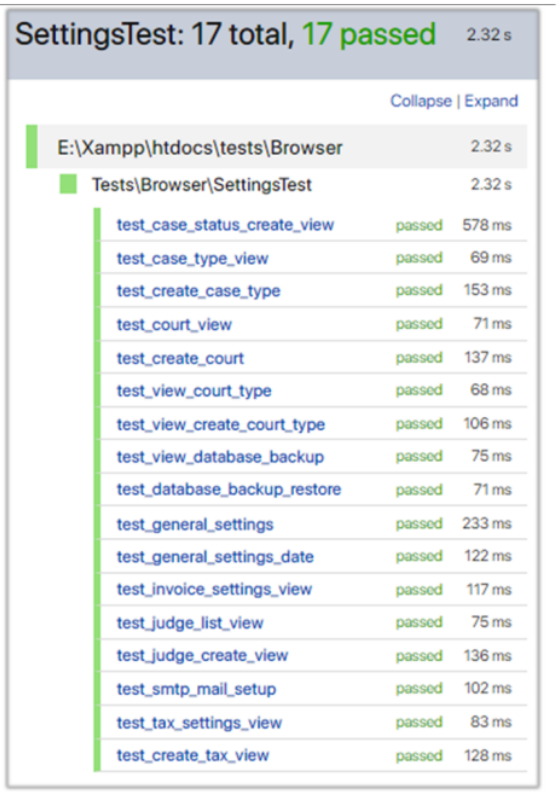
Table 9. Test Cases & Results

Test Case	Tested Data	Test Results
TC01 **TC02** **TC03**	▪ System login page contents ▪ Link request page contents ▪ Data backup page contents	
TC04	▪ Admin username ▪ Admin password	
TC05	▪ <Forgot Password?> link	
TC06	▪ Assistant username ▪ Assistant password	

TC07	▪ Accountant username ▪ Accountant password	
TC08 **TC09** **TC10**	• Admin profile data • Assistant profile data • Accountant profile data ✓ Full name ✓ Address ✓ Email address ✓ Mobile no.	
TC11	• Advocate clients data ✓ Full name ✓ Email address ✓ Permanent address ✓ Mobile number ✓ Work place address ✓ Work place number	
TC12	• Client's case data ✓ Running cases ✓ Important cases ✓ No board cases ✓ Archived cases ✓ Court details ✓ Petitioner ✓ Respondent	

TC13	• Assigned task data ✓ Task name ✓ Related to ✓ Assigned Date ✓ Deadline ✓ Members	
TC14	• Appointments data ✓ Client name ✓ Mobile number ✓ Appointment date ✓ Appointment time ✓ Appointment status	
TC15 **TC16**	• Team members data • Members roles data ✓ Member name ✓ Email address ✓ Contact number ✓ Member role	

TC17 **TC18**	• Advocate services data • Clients invoices data ✓ Advocate service ✓ Service charge ✓ Invoice number ✓ Client number ✓ Total amount ✓ Paid amount ✓ Due amount	
TC19	• Vendors data ✓ Vendor name ✓ Mobile number ✓ Email address ✓ Company address	
TC20 **TC21**	• Expenses data • Expenses types data ✓ Expense types ✓ Invoice no. ✓ Company name ✓ Total amount ✓ Paid amount ✓ Due amount ✓ Payment status	

TC22	• System settings data ✓ Case types settings ✓ Court types settings ✓ Courts settings ✓ Case status types ✓ Judges settings ✓ Taxes settings ✓ General settings ✓ Office settings ✓ Date & time zone ✓ Mail setup (smtp) ✓ Invoice settings ✓ Database backup	

8.5 Testing Metrics

Table 10. Testing Metrics

Metric No.	Testing Metrics	Data Retrieved
M01	Total test cases written per requirement	24
M02	No. of test cases failed	00
M03	No. of test cases blocked	02
M04	No. of test cases passed with warning	05
M05	No. of test cases passed without any issue	17
M06	Total no. of test assertions executed	44
M07	Total no. of defects identified during test case development	10
M08	No. of invalid defects count	02
M09	No. of critical defects count	00

M10	No. of high defects count	01
M11	No. of medium defects count	04
M12	No. of low/lowest defects count	03
M13	Total no. of main functions covered during testing	15
M14	Total no. of statements covered during testing	13
M15	Total no. of branches covered during testing	13
M16	Total no. of condition covered during testing	12
M17	Total no. of line covered during testing	1620

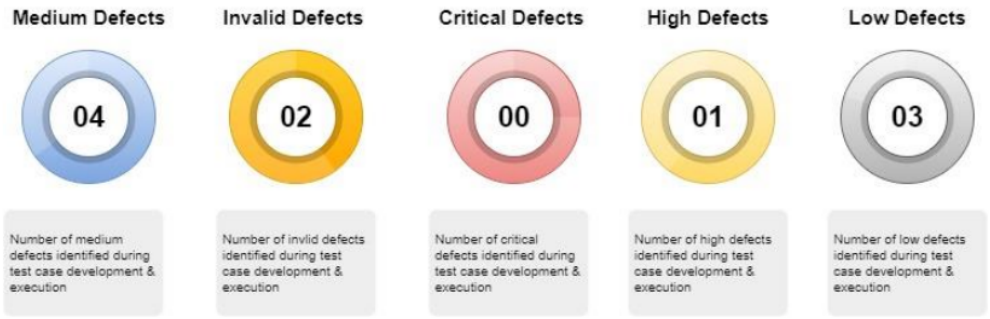


Figure 52. Defects Density

8.6 End User Validation Results

Table 11. End User Validation Results

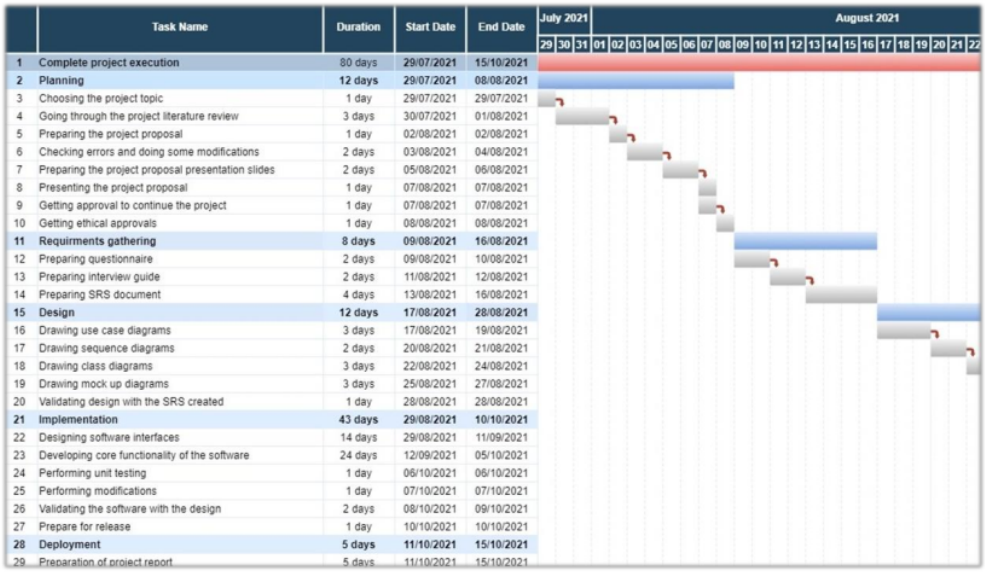
System Main Functionalities & Features	Strongly Agree	Agree	Disagree	Need Improvement(s)
01. This case management system has satisfied security features to prevent online attacks	☐	☒	☐	☐
02. This system has well functional data backup feature to backup and restore legal data just in case	☒	☐	☐	☐
03. System notification system functional without any technical issues	☒	☐	☐	☐

04. Advocate daily calendar reminders working smoothly without any delays	☐	☒	☐	☐
05. Daily appointments and court cases functional to access without using side panel	☐	☒	☐	☐
06. Advocate can manage client's cases and able to download case reports as a pdf file	☒	☐	☐	☐
07. Advocate can add team members and advocate can change members system access privileges according to member's roles	☒	☐	☐	☐
08. Advocate can manage his or her income and expenses using this system	☒	☐	☐	☐
09. Advocate can generate invoice for his or her clients and able to download as a pdf file	☒	☐	☐	☐
10. This system own system settings section that helps to change system settings without a system maintainer	☐	☒	☐	☐
11. This system has user friendly interface that anyone can handle and control	☐	☐	☐	☒
12. All the system functionalities and features running without any technical issues	☐	☒	☐	☐
13. Advocates will able to solve manual management issues through this system	☒	☐	☐	☐
14. I can verify this system as a good system to manage all needs of an advocate	☐	☒	☐	☐

8.7 End User Feedbacks & Comments

- System interface user friendly but font size needs to be increased to regular size.
- All the system functions and features worked well. Some pages take some time to load and respond.
- Better if we can upload and view case documents as an extra feature. But Google drive links will be better enough to cover that extra feature.
- I am pretty sure this system will help to manage our workload and this system will be able to fix manual management issues that we are currently facing.

9.0 Work Breakdown Structure



	Task Name	Duration	Start Date	End Date
1	**Complete project execution**	80 days	29/07/2021	15/10/2021
2	**Planning**	**12 days**	29/07/2021	08/08/2021
3	Choosing the project topic	1 day	29/07/2021	29/07/2021
4	Going through the project literature review	3 days	30/07/2021	01/08/2021
5	Preparing the project proposal	1 day	02/08/2021	02/08/2021
6	Checking errors and doing some modifications	2 days	03/08/2021	04/08/2021
7	Preparing the project proposal presentation slides	2 days	05/08/2021	06/08/2021
8	Presenting the project proposal	1 day	07/08/2021	07/08/2021
9	Getting approval to continue the project	1 day	07/08/2021	07/08/2021
10	Getting ethical approvals	1 day	08/08/2021	08/08/2021
11	**Requirments gathering**	**8 days**	09/08/2021	16/08/2021
12	Preparing questionnaire	2 days	09/08/2021	10/08/2021
13	Preparing interview guide	2 days	11/08/2021	12/08/2021
14	Preparing SRS document	4 days	13/08/2021	16/08/2021
15	**Design**	**12 days**	17/08/2021	28/08/2021
16	Drawing use case diagrams	3 days	17/08/2021	19/08/2021
17	Drawing sequence diagrams	2 days	20/08/2021	21/08/2021
18	Drawing class diagrams	3 days	22/08/2021	24/08/2021
19	Drawing mock up diagrams	3 days	25/08/2021	27/08/2021
20	Validating design with the SRS created	1 day	28/08/2021	28/08/2021
21	**Implementation**	**43 days**	29/08/2021	10/10/2021
22	Designing software interfaces	14 days	29/08/2021	11/09/2021
23	Developing core functionality of the software	24 days	12/09/2021	05/10/2021
24	Performing unit testing	1 day	06/10/2021	06/10/2021
25	Performing modifications	1 day	07/10/2021	07/10/2021
26	Validating the software with the design	2 days	08/10/2021	09/10/2021
27	Prepare for release	1 day	10/10/2021	10/10/2021
28	**Deployment**	**5 days**	11/10/2021	15/10/2021
29	Preparation of project report	5 days	11/10/2021	15/10/2021

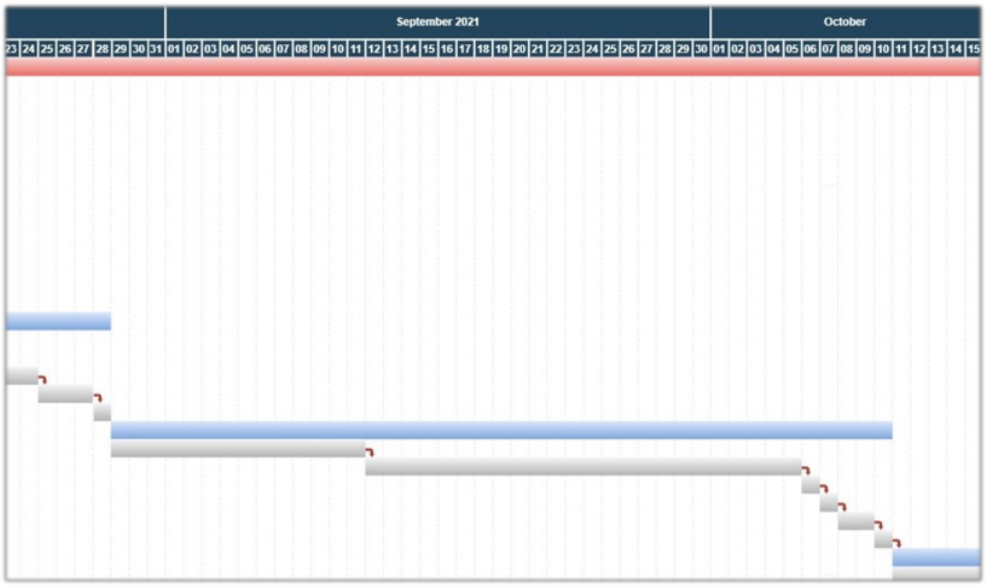


Figure 53. Work Breakdown Structure

10.0 Conclusion

There are a number of issues with the legal company, including an excessive workload, a lack of order and structure, and a dearth of effective management methods for handling client information. Consequently, the goal of this project is for a more sophisticated technique to digitalize the administration of file work by enabling the lawyer to work efficiently. The goal is to improve data management while also saving time by using online collaboration.

What I did in this project was create a program tailored to the needs of a Sri Lankan advocate working in a Sri Lankan legal system. I did my best to see that this project met its stated objectives, and in retrospect, it's obvious that the initiative was a success.

10.1 Importance of the Project

Before the start of the project there were issues with manual case management that were used by advocates. Especially legal data management issues. This software development project will help to keep those manual case management issues away. Moreover this project will help to

manage all the lawyers related stuff such as client payments management, clients appointments management etc.

10.2 Project Achievements and Results

Before the start of the project I was aimed to develop a complete digitalized case management system with other helpful managements as well. Such as appointment management, income management etc. At the end of the development project I was able to complete the main aim of the project as an achievement. Moreover project objectives were achieved at the end of the project.

10.3 Problems Faced During the Project Timeframe

There was one major problem that I have faced during the project timeframe was fixing bugs that occurred during development. I got few major bugs during backup function development. Anyway I was able to fix those bugs after a little research over the internet.

10.4 Limitations

- Some of the system icons (Font Awesome) and designs not supported for every browser and systems UI alignments can be changed through mobile web browsers.
- Some of the system animations are not supported by old browsers.
- Some pages take a while to load some times.

10.5 Future Recommendations

- I plan to add additional case documents upload options as a future development.
- I plan to increase the range of application so that advocates clients' also can access the system to book appointments or maybe see their case details such as next case hearing date, venue etc.

11.0 References List

Aarthi, S., Siddharth, S., Athreya, V. and Balaji, P. (2018). Legal Files Management System Using Big Data. [online] IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/8723904 [Accessed 28 Aug. 2021].

www.capterra.com. (n.d.). App4Legal vs CaseFox 2021 - Feature and Pricing Comparison on Capterra. [online] Available at: https://www.capterra.com/legal-case-management-software/compare/146976-121682/App4Legal-vs-CaseFox [Accessed 28 Aug. 2021].

Aschner, K. (2010). Files Management in the Pre-Paperless Office. Journal of Information Systems Management, 3(4), pp.78–81. Available at: https://www.tandfonline.com/doi/abs/10.1080/07399018608965279 [Accessed 28 Aug. 2021].

www.capterra.com. (n.d.). Best Legal Case Management Software 2021 | Reviews of the Most Popular Tools & Systems. [online] Available at: https://www.capterra.com/legal-case-management-software/ [Accessed 28 Aug. 2021].

buyersguide.americanbar.org. (n.d.). Clio - ABA Legal Technology Buyers Guide. [online] Available at: https://buyersguide.americanbar.org/sites/clio/3092/Document+Management. [Accessed 28 Aug. 2021].

GetApp. (n.d.). Clio vs eFileCabinet. [online] Available at: https://www.getapp.co.uk/compare/91415/103257/clio/vs/efilecabinet# [Accessed 28 Aug. 2021].

buyersguide.americanbar.org. (n.d.). Filevine - ABA Legal Technology Buyers Guide. [online] Available at: https://buyersguide.americanbar.org/sites/filevine/8930/Document+Management [Accessed 28 Aug. 2021].

Kang, D., Xing, H., Wu, S., Chen, J. and Zhan, J. (2012). Design of University Teacher's Files Management System. [online] IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/6405848 [Accessed 28 Aug. 2021].

Anon, (n.d.). Knowledge-management systems for law firms. [online] Available at: https://patents.google.com/patent/US8126818B2/en [Accessed 29 Aug. 2021].

Du Toit, A. and Du Plessis, T. eds., (2015). Survey of information and knowledge management in South African law firms. [online] ResearchGate. Available at: https://www.researchgate.net/publication/264086685_Survey_of_information_and_knowledge_management_in_South_African_law_firms [Accessed 29 Aug. 2021].

Mishra, A. and Mishra, D. eds., (2011). A Legal Business Information System: Implementation Process Context. [online] ResearchGate. Available at: https://www.researchgate.net/publication/259148863_A_Legal_Business_Information_System_Implementation_Process_Context [Accessed 29 Aug. 2021].

Oorschot, I.V. ed., (2014). Seeing the Case Clearly: File-Work, Material Mediation, and Visualizing Practices in a Dutch Criminal Court. [online] ResearchGate. Available at: https://www.researchgate.net/publication/267099780_Seeing_the_Case_Clearly_File-Work_Material_Mediation_and_Visualizing_Practices_in_a_Dutch_Criminal_Court [Accessed 29 Aug. 2021].

Zhang, H., Diao, J. and Wen, Q.-Y. eds., (2018). Secure Files Management System in Intranet. [online] ResearchGate. Available at: https://www.researchgate.net/publication/4347280_Secure_Files_Management_System_in_Intranet [Accessed 29 Aug. 2021].

Filho, F. (2019). The use of ICT in Brazilian Courts. Electronic Journal of e-Government, [online] 7. Available at: https://academicpublishing.org/index.php/ejeg/article/download/512/475 [Accessed 29 Aug. 2021].

Greenwood, J.M. and Bockweg, G. (2012). Insights to Building a Succesful E-filing Case Management Service: U.S. Federal Court Experience. International Journal for Court Administration, 4(2), p.2. Available at: https://www.iacajournal.org/articles/abstract/10.18352/ijca.74/ [Accessed 29 Aug. 2021].

Arda Kılıçdağı and Halil İbrahim Yılmaz (2014). Laravel design patterns and best practices : enhance the quality of your web applications by efficiently implementing design patterns in Laravel. Birmingham, Uk: Packt Publishing Ltd.

Toptal Engineering Blog. (n.d.). Laravel API Tutorial: How to Build and Test a RESTful API. [online] Available at: https://www.toptal.com/laravel/restful-laravel-api-tutorial [Accessed 23 Sep. 2021].

laravel.com. (n.d.). Installation - Laravel - The PHP Framework For Web Artisans. [online] Available at: https://laravel.com/docs/8.x [Accessed 23 Sep. 2021].

12.0 Appendix

12.1 Survey Model (Online – Microsoft Form)

Online Survey

Advanced Multifunctional Court Cases Management System

* Required

1. Lawyer Name: *

2. Age: *

3. Experience(Years) & Specialization: *

* Please answer following questions

4. Are you facing issues while using manual court case management? *

- ◯ Yes
- ◯ No

5. If yes, what are the issues that you're facing while using manual case management?

- ☐ Time and dates management issues
- ☐ Case files and data misplacements
- ☐ Difficulties to find exact case file from files cabinets
- ☐ Client's information management issues
- ☐ Other management issues (appointments, income, expenses)

6. Do you think digitalized case management system will solve those issues? *

- ◯ Yes
- ◯ No

7. Would you like to work with digitalized management system? *

- ◯ Yes
- ◯ No

8. If yes, what are the main functions (features) do you prefer or want from digitalized court case management system? *

- ☐ Clients case management
- ☐ Client's information management
- ☐ Future tasks management
- ☐ Client's appointments management
- ☐ Appointments tracking calendar
- ☐ Other managements (income, expenses, venders, team)
- ☐ Data backup option

Figure 54. Survey Model (Online - Microsoft Form)

12.2 Database Tables

admins	court_cases	payment_mades
admin_password_resets	court_types	payment_receiveds
admin_role	dumps	permissions
advocate_clients	expenses	permission_role
all_taxes	expenses_items	roles
appointments	expense_cats	services
case_logs	general_settings	settings
case_members	invoices	sql_backups
case_parties_involves	invoice_items	states
case_statuses	invoice_settings	tasks
case_transfers	judges	task_members
case_types	mailsetups	users
cities	migrations	user_password_resets
client_parties_invoives	newsletters	vendors
countries	notifications	verifybackup
courts	password_resets	zone

Figure 55. Database Tables

12.3 Project Log Sheets

Table 12. Project Log Sheets

34

PLS01

Student's Name: D M L S DASSANAYAKE		**Cardiff Number**: ST20197917
Date: 29/08/2021	**Meeting No**: 01	**Intake**: 21/ 2021
Project Title: Advanced Multifunctional Court Cases Management System (CCMS) For Supreme Court Lawyers By Using Latest Laravel PHP Web Framework.		

PLS02

Student's Name: D M L S DASSANAYAKE **Cardiff Number**: ST20197917

Date: 02/09/2021 **Meeting No**: 02 **Intake**: 21/ 2021

Project Title: Advanced Multifunctional Court Cases Management System (CCMS) For Supreme Court Lawyers By Using Latest Laravel PHP Web Framework.

PLS03

Student's Name: D M L S DASSANAYAKE **Cardiff Number**: ST20197917

Date: 09/09/2021 **Meeting No**: 03 **Intake**: 21/ 2021

Project Title: Advanced Multifunctional Court Cases Management System (CCMS) For Supreme Court Lawyers By Using Latest Laravel PHP Web Framework.

PLS04

Student's Name: D M L S DASSANAYAKE **Cardiff Number**: ST20197917

Date: 09/09/2021 **Meeting No**: 04 **Intake**: 21/ 2021

Project Title: Advanced Multifunctional Court Cases Management System (CCMS) For Supreme Court Lawyers By Using Latest Laravel PHP Web Framework.

PLS05

Student's Name: D M L S DASSANAYAKE **Cardiff Number**: ST20197917

Date: 23/09/2021 **Meeting No**: 05 **Intake**: 21/ 2021

Project Title: Advanced Multifunctional Court Cases Management System (CCMS) For Supreme Court Lawyers By Using Latest Laravel PHP Web Framework.

PLS06

Student's Name: D M L S DASSANAYAKE **Cardiff Number**: ST20197917

Date: 04/10/2021 **Meeting No**: 06 **Intake**: 21/ 2020

Project Title: Advanced Multifunctional Court Cases Management System (CCMS) For Supreme Court Lawyers By Using Latest Laravel PHP Web Framework.

PLS07

Student's Name: D M L S DASSANAYAKE		**Cardiff Number**: ST20197917
Date: 11/10/2021	**Meeting No**: 07	**Intake**: 21/ 2020
Project Title: Advanced Multifunctional Court Cases Management System (CCMS) For Supreme Court Lawyers By Using Latest Laravel PHP Web Framework.		

PLS08

Student's Name: D M L S DASSANAYAKE		**Cardiff Number**: ST20197917
Date: 13/10/2021	**Meeting No**: 08	**Intake**: 21/ 2020
Project Title: Advanced Multifunctional Court Cases Management System (CCMS) For Supreme Court Lawyers By Using Latest Laravel PHP Web Framework.		

13.0 Glossary of Terms

Table 13. Glossary of Terms

Addition	The action or process of adding something to something else.
Subtraction	The process or skill of taking one number or amount away from another.
Multiplication	The process or skill of multiplying.
Division	The action of separating something into parts or the process of being separated.
Facade	A deceptive outward appearance.
Vendor	A person or company offering something for sale, especially a trader.
Expense	The cost incurred in or required for something.
Advocate	A professional pleader in a court of justice.
Feasibility	The state or degree of being easily or conveniently done.
Repository	A place where or receptacle in which things are or may be stored.
Commit	Transfer something to (a state or place where it can be kept or preserved).
Aliases	An alternative name or label that refers to a file, command, address, or other item, and can be used to locate or access it.

14.0 List of Abbreviation

Table 14. List of Abbreviation

SASS	Syntactically awesome style sheets
SCSS	Syntactically cascading style sheets
HTML	Hypertext markup language
MVC	Model view controller
IDE	Integrated development environment
CSS	Cascading style sheets
JSP	Java server pages
TC	Test case
DB	Database
CCMS	Court case management system
API	Application programming interface
REST	Representational state transfer
OOP	Object-oriented programming
SQL	Structured query language
PHP	Hypertext preprocessor
FTP	File transfer protocol
GSTIN	Goods and service tax identification number
PAN	Permanent account number
CNR	Case number record
FIR	First information report
UML	Unified modeling language
OS	Operating system
ICT	Information and communications technology
PLS	Project log sheet

CL-BSCSD-21-10_ST20197917_Lahiru Sampath_Dissertation.pdf

ORIGINALITY REPORT

11%	10%	1%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	global.oup.com Internet Source	2%
2	ir.kdu.ac.lk Internet Source	1%
3	laravel.com Internet Source	1%
4	www.tutorialspoint.com Internet Source	1%
5	scotch.io Internet Source	1%
6	Submitted to Ghana Technology University College Student Paper	<1%
7	adelekand.wordpress.com Internet Source	<1%
8	en.wikipedia.org Internet Source	<1%
	Submitted to University of Greenwich	

9	Student Paper	<1%
10	Submitted to Swinburne University of Technology Student Paper	<1%
11	www.aaccu.coop Internet Source	<1%
12	fsegitlab.wlv.ac.uk Internet Source	<1%
13	Submitted to General Sir John Kotelawala Defence University Student Paper	<1%
14	ebookee.org Internet Source	<1%
15	elearning.symphonytech.com Internet Source	<1%
16	www.encyclopedia.com Internet Source	<1%
17	Submitted to University of Northumbria at Newcastle Student Paper	<1%
18	Submitted to Southern University And A & M College Student Paper	<1%
19	adun.edu.ng Internet Source	

		<1%
20	Submitted to University of Bradford Student Paper	<1%
21	Submitted to University of Westminster Student Paper	<1%
22	hdl.handle.net Internet Source	<1%
23	Submitted to University of Portsmouth Student Paper	<1%
24	Submitted to International School of Management and Technology (ISMT), Nepal Student Paper	<1%
25	Submitted to Glasgow Caledonian University Student Paper	<1%
26	Submitted to Staffordshire University Student Paper	<1%
27	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
28	Submitted to Kingston University Student Paper	<1%
29	Submitted to University of Southampton Student Paper	<1%

30	repository.cardiffmet.ac.uk Internet Source	<1%
31	support.builderdesigns.com Internet Source	<1%
32	eacop.com Internet Source	<1%
33	kristenpoborsky.com Internet Source	<1%
34	m-mebel.by Internet Source	<1%
35	www.xsonly.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		