

---

# ONLINE CASE MANAGEMENT SYSTEM FOR LAWYERS

---

**By**

**M. Shahid Sarwar**

**Supervised By**

**Memoona Afsheen Malik**

**Department of Computer Science**

**Quaid-I-Azam University**

**Islamabad**

**Session 2014-2018**

# Abstract

Online Case Management System for Lawyers (portfolio for firm) is a web-based application. This web-based application allows users to interact with each other on a single platform. Lawyer can easily manage and can maintain cases relevant information. Lawyer can check the status of different cases either it is pending, ongoing or closed. Lawyer can search the old cases easily by remembering the judge name or by client name and many more, the great search facility is provided to the lawyer and they can use searched cases information for the new cases when needed. Lawyer can easily find out details of opponent lawyer by considering the case details and can figure out the way with opponent lawyer for settlement when both parties got agreed on some terms and conditions. System can easily list cases under different courts added by lawyer. Clients can easily get appointment with the lawyer online and avoid future inconvenience. Lawyer can schedule reminders for their meetings with the clients. Lawyer can share the pictures of events which he /she had organized. Lawyer can share the news about their cases and about bar activities so that clients come to know about the reputation of lawyer. Lawyer can share the details of their staff with clients. Clients often find it difficult to find the office. By sharing the information about offices, clients can easily find out the office.

## Contents

Abstract .....	1
Chapter 1 .....	8
Introduction .....	8
1.1) Problem Definition.....	8
1.2) Proposed Solution .....	8
1.3) Scope .....	9
1.4) Objectives .....	9
1.5) Project Organization.....	10
1.5.1) Software Process Model .....	10
1.5.2) Roles and Responsibilities .....	10
1.5.3) Tools and Techniques.....	10
1.6) Project Management Plan.....	10
1.6.1) Project Libre plan .....	11
1.7) Giant Chart.....	12
1.8) Report Structure .....	13
Chapter 2 .....	14
Requirements Gathering and Analysis .....	14
2.1) Introduction.....	14
2.1.2) Stakeholders .....	14
2.1.3) Major Functions .....	14
2.1.4) Major Inputs.....	14
2.1.5) Major Outputs .....	14
2.1.6) Definitions, Acronyms and Abbreviations .....	15
2.2) Overall Description .....	15
2.2.1) Product Perspective .....	15
2.2.1.1) System Interfaces.....	15
2.2.1.2) User Interfaces.....	15
2.2.1.3) Software Interfaces .....	16
2.2.1.4) Hardware Interfaces.....	16
2.2.2) Product Functions.....	16

2.2.2.1) Request to login.....	16
2.2.2.2) Manage Cases Information .....	16
2.2.2.3) Manage Client’s information .....	16
2.2.2.4) Manage Staff Information.....	16
2.2.2.5) Manage Search Engine .....	16
2.2.2.6) Manage User’s Information .....	16
2.2.2.7) Share News about Cases .....	16
2.2.2.8) Request to get Appointment.....	17
2.2.2.9) Sharing of Event Details .....	17
2.2.2.10) Schedule Reminder for Appointments.....	17
2.2.2.11) Schedule Reminder for Events .....	17
2.2.3) User Characteristics.....	17
2.2.4) Constraints .....	17
2.2.5) Assumptions and Dependencies.....	17
2.3) Specific Requirements.....	17
2.3.1) Functional Requirements .....	17
2.3.2) Non-functional Requirements .....	18
2.3.2.1) Security .....	18
2.3.2.2) Reliability.....	18
2.3.2.3) Availability.....	18
2.3.2.4) Performance .....	18
2.3.2.5) Maintainability .....	18
2.4) List of Use Cases.....	18
2.4.1) Use Case Diagram.....	20
2.5) Use Case Descriptions.....	21
Use Case Details .....	21
2.5.1) Login.....	21
2.5.2) Add Case.....	22
2.5.3) View Case .....	23
2.5.4) Update Case .....	24
2.5.5) Delete Case .....	25

2.5.6) Search Case .....	26
2.5.7) Add Client's Data .....	27
2.5.8) View Client's Data .....	28
2.5.9) Update Client's Data .....	29
2.5.10) Delete Client's Data.....	30
2.5.11) Post News.....	31
2.5.12) Post Event Details.....	32
2.5.13) View News and Events Details .....	33
2.5.14) Schedule Reminder for Events .....	34
2.5.15) Request for Appointments .....	35
2.5.16) View Appointments.....	36
2.5.17) Response for Appointments .....	37
2.5.18) Schedule Reminder for Appointment.....	38
2.5.19) Add Staff Data.....	39
2.5.20) View Staff Data .....	40
2.5.21) Update Staff Data .....	41
2.5.22) Delete Staff Data .....	42
2.5.23) View Lawyer Details .....	43
2.5.24) Logout.....	44
2.6) Domain Model .....	45
2.7) Database Requirements .....	46
Chapter 3 .....	47
Software Design Description.....	47
3.1) Design Overview .....	47
3.1.2) Requirements Traceability Matrix.....	47
3.2) System Sequence Diagram .....	48
3.2.1) SSD login.....	48
3.2.2) Post News .....	49
3.2.3) Add Case.....	50
3.2.4) Add Client.....	51
3.2.5) View Case .....	52

3.2.6) Reminder for Appointments .....	53
3.2.7) View Appointments.....	54
3.3) Architecture Diagram.....	54
3.4) Interface design.....	56
3.4.1) Simple and Appealing .....	56
3.4.2) Responsive .....	56
3.5) Interfaces .....	56
3.5.1) Interface Login .....	56
3.5.2) Post Event .....	57
3.5.3) View Client .....	57
3.5.4) Add Case.....	58
3.5.5) Post News about Bar activities.....	59
3.6) Object and Actions (Sequence Diagram) .....	59
3.6.1) SD Login.....	60
3.6.2) SD Post News.....	60
3.6.3) SD Add Case.....	61
3.6.4) SD Add Client .....	62
3.6.5) SD Request for Appointment .....	63
3.6.6) SD View Case .....	64
3.6.7) SD Schedule Reminder for Appointment .....	65
3.7) Class Diagram.....	66
Chapter 4 .....	67
Implementation .....	67
4.1) System Definition .....	67
4.1.1) Database Server.....	67
4.1.2) Web Application .....	67
4.2) Development Tools .....	67
4.2.1) Framework .....	67
4.2.2) Language Selection .....	68
4.3) Why C#? .....	68
4.3.1) Platform Independent .....	68

4.3.2) Object Oriented .....	68
4.3.3) Robust and Secure .....	68
4.3.4) Distributed.....	68
4.4) Code Snippets .....	69
Chapter 5 .....	70
System Testing.....	70
5.1) Test Approach.....	70
5.2) Test Plan .....	70
5.1.1) Testing Tools and Environment .....	70
5.3) Test Cases .....	70
5.3.1) Login.....	71
5.3.2) Login (Alternative Scenario) .....	71
5.3.3) Post News .....	72
5.3.4) Post News (Alternative Scenario) .....	72
5.3.5) View Appointments.....	73
5.3.6) View Appointments (Alternative Scenario).....	73
5.3.7) Add Case.....	74
5.3.8) Add Case (Alternative Scenario).....	74
5.3.9) Add Client's data.....	75
5.3.10) Add Client's data (Alternative Scenario).....	75
Chapter 6 .....	76
Conclusions and future enhancements .....	76
Future Enhancements .....	76

## Content of Figures

Definitions Table 2.1.....	15
Figure 2.1 Use Case Diagram.....	20
Figure 2.2 Domain Model .....	45
Figure 2.3 ERD.....	46
Table of Matrix 3.1 .....	47
Figure 3.1 SSD Login .....	48
Figure 3.2 SSD Post News .....	49
Figure 3.3 SSD Add Case.....	50
Figure 3.4 SSD Add Client.....	51
Figure 3.5 SSD View Case.....	52
Figure 3.6 SSD Reminder For Appointments .....	53
Figure 3.7 SSD View Appointments.....	54
Figure 3.8 Architecture Diagram.....	55
Figure 3.9 Interface Login.....	56
Figure 3.10 Interface Schedule Event .....	57
Figure 3.11 Interface View Case .....	57
Figure 3.12 Interface Add Case .....	58
Figure 3.13 Interface Post News about Bar Activities.....	59
Figure 3.14 SD Login.....	60
Figure 3.15 SD Post News.....	60
Figure 3.16 SD Add Case.....	61
Figure 3.17 SD Add Client .....	62
Figure 3.18 SD Request for Appointment.....	63
Figure 3.19 SD View Case .....	64
Figure 3.20 SD Schedule Reminder For Appointment .....	65
Figure 3.21 Class Diagram .....	
Figure 4.1 Code Snap.....	69



# Chapter 1

## Introduction

Online Case Management System software will provide many facilities to lawyers. In this software, lawyer can easily maintain the record of their cases, and can know at a glance which matters individuals have been involved in and the status of those matters. They can access previous cases relevant information and use it for their new cases. Instead of looking into bulky files manually, with the help of this software, lawyers can search a case by keyword or title, client name, judge name etc.

### 1.1) Problem Definition

There exists some platform where lawyers can manage cases relevant information, but not all our desired needs don't get fulfill by those platforms. So, we are building our own software for this. Lawyers are working on multiple cases and they often find it difficult to distinguish between pending, ongoing and closed cases. For finding a case, they consider bulky files which are a time-consuming process. Lawyers face a lot of difficulties when they have to do a settlement between two parties because they don't have all details regarding case such as opponent lawyer details etc. There is no online as such source where lawyers can list cases under different courts. Meanwhile lawyers are working on different cases of different courts, so lawyers find it difficult to maintain the record of all cases and to find case details when needed. Clients often find it difficult to meet the lawyer and find his/her office. Because of hectic routine, lawyers often forgot their meetings with their clients. There is no online source where lawyers can share the pictures of events which were organized under his/her supervision and share the details about their cases such as how many cases they have won and names of those cases etc.

### 1.2) Proposed Solution

Hence a system has been proposed to be deployed in Quaid-E-Azam University, Islamabad named as "Case Management System" which is a web-based application. This web-based application allows users to interact with each other on a single platform. Lawyers can easily manage and can maintain cases relevant information. Lawyers can check the status of different cases either it is pending, ongoing or closed. Lawyers can search the old cases easily by remembering the judge name or by date and many more, the great search facility is provided to the lawyer and they can use searched cases information for the new cases when needed. Lawyer can easily find out details of opponent lawyer by considering the case details and can figure out the way with opponent

lawyer for settlement when both parties got agreed on some terms and conditions. Lawyers can easily list cases under different courts. Clients can easily get appointment with the lawyer online and avoid future inconvenience. Lawyers can schedule reminders for their meetings with the clients. Lawyers can share the pictures of events which he /she had organized. Lawyers can share the news about their cases so that clients come to know about the reputation of lawyer. Clients often find it difficult to find the office. By sharing the information about offices, clients can easily find out the office.

### **1.3) Scope**

This system is totally web-based. This web-based system is only for lawyers and clients. System will provide a platform to those lawyers, who are interested to maintain their cases repository online. This system provides a facility to save: -

- case relevant information
- manages client details
- list cases under different courts
- manages staff relevant details
- view lawyer's details
- view and can schedule appointments
- maintain gallery
- post information about their victories (news)
- This system also provides a facility to lawyers by providing a search option where lawyer can easily search a case by keyword, title, year etc.

### **1.4) Objectives**

Objective of this system is to save the time of users (lawyers, clients). Users can easily interact with each other. The tasks related to project can be done on single platform. The main objective of this project is to facilitate the users with different services are as follows: -

- It saves and manages cases (case no, case title, etc.) information.
- It saves and manages client's data.
- It saves and manages staff data.
- It lists cases under different courts.

- It posts news and events details.
- It schedules appointments and set reminders for them.
- It manages gallery.
- It provides extensive search engine for cases.

## **1.5) Project Organization**

A project organization is a structure that facilitates the coordination and implementation of project activities. Its main reason is to create an environment that fosters interactions among the team members with a minimum amount of disruptions, overlap and conflict.

### **1.5.1) Software Process Model**

I have used waterfall as software model for the completion of project due to following reasons: -

- Project requirements are clear
- Enough time to implement
- Each phase is completed first and then start second.

### **1.5.2) Roles and Responsibilities**

Administrator plays a major role here. Administrator can add case relevant information, client's data, and details of staff and can modify it. Administrator can search cases through extensive search engine and can share the details of events and can share news about cases. Administrator can see all activities carried out by users like getting an appointment. User can just view the staff details, office details and gallery and can request for appointment.

### **1.5.3) Tools and Techniques**

I have used Visual studio 2013 and language asp.net to implement Online Case Management System for Lawyers. To store the database, I used the SQL SERVER 2008 which is very powerful tool to store a large database.

## **1.6) Project Management Plan**

The project management plan is a formal, approved document used to manage project execution. The PMP documents the actions necessary to define, prepare, integrate and coordinate the various planning activities. The PMP defines how the project is executed, monitored, controlled and closed.

## 1.6.1) Project Libre plan

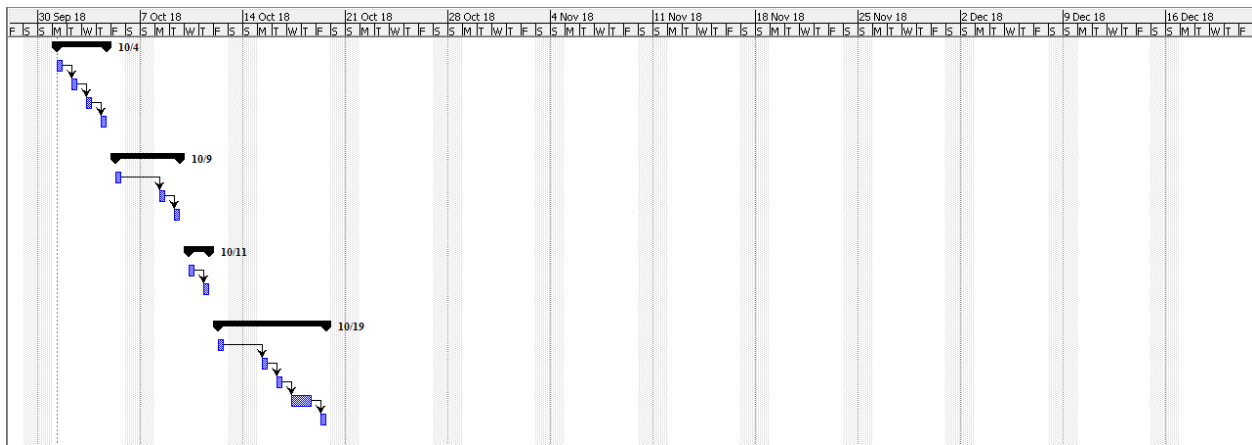
Name	Start	Finish	Manager	Status Date
Final Year Project	10/1/18 8:00 AM	1/9/19 5:00 PM	M Shehid Sarwar	2/11/19 5:00 PM

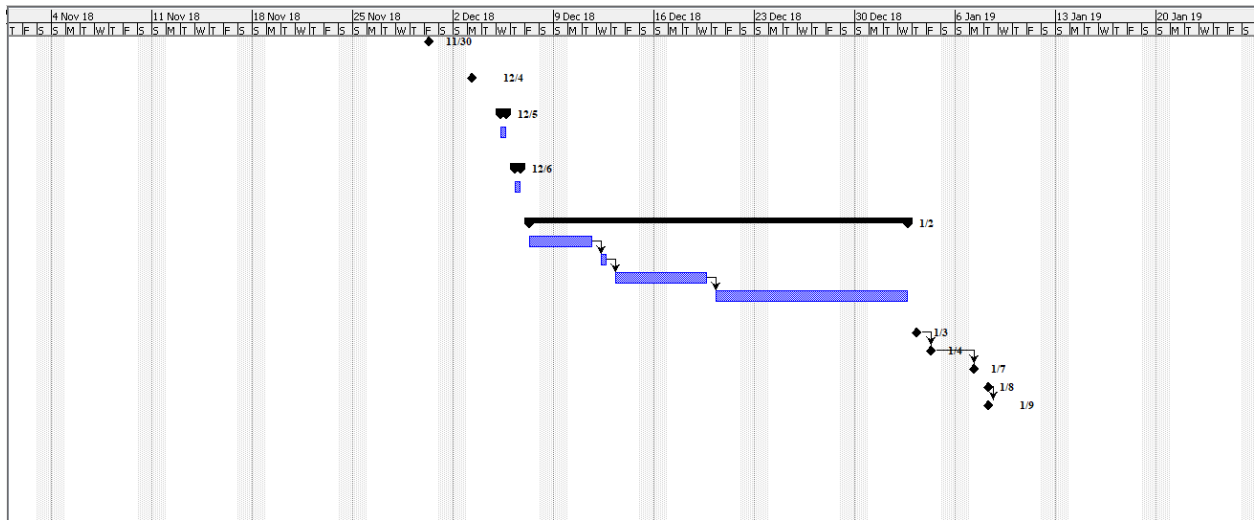
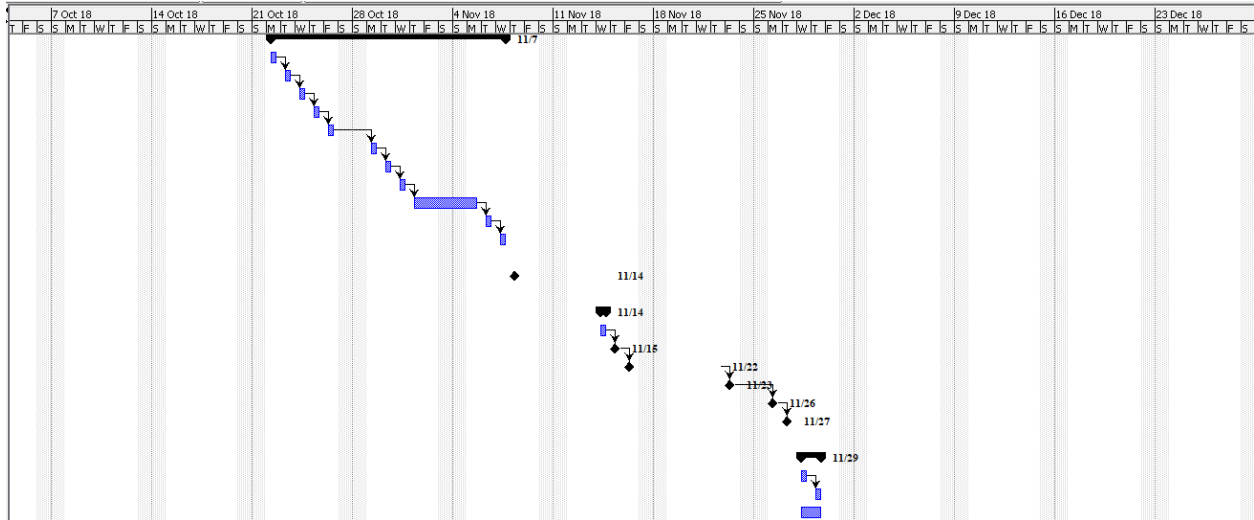
Complete plan is given below: -

	⊙	Name	Duration	Start	Finish	Predecessors	Resource Names
1	⊙	<input type="checkbox"/> Project Planning	4 days?	10/1/18 8:00 AM	10/4/18 5:00 PM		
2		Intro to Project planning...	1 day?	10/1/18 8:00 AM	10/1/18 5:00 PM		
3		Intro to projects & alloca...	1 day?	10/2/18 8:00 AM	10/2/18 5:00 PM	2	
4		Development of the proj...	1 day?	10/3/18 8:00 AM	10/3/18 5:00 PM	3	
5		Review of Project planning	1 day?	10/4/18 8:00 AM	10/4/18 5:00 PM	4	
6	⊙	<input type="checkbox"/> Collect Requirements	3 days?	10/5/18 8:00 AM	10/9/18 5:00 PM		
7		Collect functional reqd	1 day?	10/5/18 8:00 AM	10/5/18 5:00 PM		
8		Collect non - functional r...	1 day?	10/8/18 8:00 AM	10/8/18 5:00 PM	7	
9		Meeting with customer	1 day?	10/9/18 8:00 AM	10/9/18 5:00 PM	8	
10	⊙	<input type="checkbox"/> Defining Requirments	2 days?	10/10/18 8:00 AM	10/11/18 5:00 PM		
11		Write it on paper	1 day?	10/10/18 8:00 AM	10/10/18 5:00 PM		
12		Meeting with customer	1 day?	10/11/18 8:00 AM	10/11/18 5:00 PM	11	
13	⊙	<input type="checkbox"/> Define Usecases	6 days?	10/12/18 8:00 AM	10/19/18 5:00 PM		
14		Write main, alternative s...	1 day?	10/12/18 8:00 AM	10/12/18 5:00 PM		
15		Identfy system	1 day?	10/15/18 8:00 AM	10/15/18 5:00 PM	14	
16		Draw use cases	1 day?	10/16/18 8:00 AM	10/16/18 5:00 PM	15	
17		Reviews of usecases	2 days?	10/17/18 8:00 AM	10/18/18 5:00 PM	16	
18		Meeting with customer	1 day?	10/19/18 8:00 AM	10/19/18 5:00 PM	17	
19	⊙	<input type="checkbox"/> Development of analys..	13 days?	10/22/18 8:00 AM	11/7/18 5:00 PM		
20		Draw a data model	1 day?	10/22/18 8:00 AM	10/22/18 5:00 PM		
21		Creates a model at cust...	1 day?	10/23/18 8:00 AM	10/23/18 5:00 PM	20	
22		Draw behavioural model	1 day?	10/24/18 8:00 AM	10/24/18 5:00 PM	21	
23		Draw State transaction ...	1 day?	10/25/18 8:00 AM	10/25/18 5:00 PM	22	
24		Draw System sequence ...	1 day?	10/26/18 8:00 AM	10/26/18 5:00 PM	23	
25		Draw Activity diagram	1 day?	10/29/18 8:00 AM	10/29/18 5:00 PM	24	
26		Draw State Chart diagram	1 day?	10/30/18 8:00 AM	10/30/18 5:00 PM	25	
27		Draw Domain model	1 day?	10/31/18 8:00 AM	10/31/18 5:00 PM	26	
28		Reviews of all models	3 days?	11/1/18 8:00 AM	11/5/18 5:00 PM	27	
29		Meeting with customer	1 day?	11/6/18 8:00 AM	11/6/18 5:00 PM	28	
30		Define language for coding	1 day?	11/7/18 8:00 AM	11/7/18 5:00 PM	29	

	📌	Name	Duration	Start	Finish	Predecessors	Resource Names
31	📌	Development SRS	5 days?	11/8/18 8:00 AM	11/14/18 5:00 PM		
32	📌	📄 Draw package diagram	1 day?	11/14/18 8:00 AM	11/14/18 5:00 PM		
33		Find UML elements	1 day?	11/14/18 8:00 AM	11/14/18 5:00 PM		
34		Draw Component diagram	1 day?	11/15/18 8:00 AM	11/15/18 5:00 PM	33	
35	📌	Identify Classes and objects	5 days?	11/16/18 8:00 AM	11/22/18 5:00 PM	34	
36		Draw Class diagram	1 day?	11/23/18 8:00 AM	11/23/18 5:00 PM	35	
37		Draw Sequence diagram	1 day?	11/26/18 8:00 AM	11/26/18 5:00 PM	36	
38		Create Design class	1 day?	11/27/18 8:00 AM	11/27/18 5:00 PM	37	
39	📌	📄 Make interface design	2 days?	11/28/18 8:00 AM	11/29/18 5:00 PM		
40		Input Interface	1 day?	11/28/18 8:00 AM	11/28/18 5:00 PM		
41		Output Interface	1 day?	11/29/18 8:00 AM	11/29/18 5:00 PM	40	
42	📌	Review for refinement o...	2 days?	11/28/18 8:00 AM	11/29/18 5:00 PM		
43	📌	Meeting with customer	1 day?	11/30/18 8:00 AM	11/30/18 5:00 PM		
44	📌	Database Connectivity	2 days?	12/1/18 8:00 AM	12/4/18 5:00 PM		
45	📌	📄 Define Classes	1 day?	12/5/18 8:00 AM	12/5/18 5:00 PM		
46		Make Objects	1 day?	12/5/18 8:00 AM	12/5/18 5:00 PM		
47	📌	📄 Interconnect classes	1 day?	12/6/18 8:00 AM	12/6/18 5:00 PM		
48		Make relations between ...	1 day?	12/6/18 8:00 AM	12/6/18 5:00 PM		
49	📌	📄 Coding	19 days?	12/7/18 8:00 AM	1/2/19 5:00 PM		
50		Use meaningful variables	3 days?	12/7/18 8:00 AM	12/11/18 5:00 PM		
51		Give comments after eac..	1 day?	12/12/18 8:00 AM	12/12/18 5:00 PM	50	
52	📌	Building front end	5 days?	12/13/18 8:00 AM	12/19/18 5:00 PM	51	
53	📌	Refining front end	10 days?	12/20/18 8:00 AM	1/2/19 5:00 PM	52	
54	📌	Testing the software	1 day?	1/3/19 8:00 AM	1/3/19 5:00 PM		
55	📌	Dry Run	1 day?	1/4/19 8:00 AM	1/4/19 5:00 PM	54	
56	📌	Testing by other software	1 day?	1/7/19 8:00 AM	1/7/19 5:00 PM	55	
57	📌	Validation	0 days?	1/8/19 8:00 AM	1/8/19 8:00 AM		
58	📌	Verification	2 days?	1/8/19 8:00 AM	1/9/19 5:00 PM	57	

## 1.7) Giant Chart





## 1.8) Report Structure

In the first chapter, I discussed the introduction of the project, the problem definition and solution. In the second chapter, I will discuss the major input and output major functionalities. In the third chapter, there is a sequence diagram, EERD and class diagram. In chapter four, there is a discussion of implementation: which tools and techniques are used. The chapter five contains the test cases. In the six chapter, I would conclude and discuss about future enhancements in project.

# Chapter 2

## Requirements Gathering and Analysis

This section of the document specifying the general factors that affects the product and its requirements, providing a background for the requirements of the software. It also describes the summary of the functions that software will perform. It also describes the user capabilities and their interest.

### 2.1) Introduction

Requirements gathering and analysis is very necessary before implementing this project. In this chapter, I discuss the functional and non-functional requirements, the stakeholders, the major functions, inputs and outputs, overall description of the product, use cases and domain model.

#### 2.1.2) Stakeholders

A person, group or organization that has an interest or concern in an organization. Lawyers and clients are stakeholders.

#### 2.1.3) Major Functions

Admin will register himself into the system by filling sign up form. After registration, admin can record cases relevant information. He can maintain client's repository. He can search for the cases by remembering judge name, date, year etc. He can add, update, and delete staff details. He can share news about his/her cases. He will send responses to the users on their personalized service requests (like I f a user request for an appointment and admin will respond to his request e.g. whether appointment is given or not). He can schedule reminder for appointment. He can share the details of events which is going to held under his/her supervision.

#### 2.1.4) Major Inputs

Major Input of the system which I implemented is that admin can login into the system by using his username and password. When the admin logs in into the system, admin can see the requests for appointments. Admin can share news about cases.

Admin can maintain client's profiles. Admin can record cases relevant information. Admin can search cases by extensible search engine. Admin can share the pictures of events.

#### 2.1.5) Major Outputs

Major output of the system is that system show the grid of menu where user can see lawyer details, his office details, news about his cases, his gallery which comprises of pictures about his events

and an appointment option where user can request for an appointment to admin. There will be login option for admin. After login, system will show different options to admin.

### **2.1.6) Definitions, Acronyms and Abbreviations**

<b>Acronyms</b>	<b>Abbreviations</b>
<b>User</b>	Guest Users (Clients)
<b>Admin</b>	Lawyer
<b>PMP</b>	Project Management Plan

**Definitions Table 2.1**

## **2.2) Overall Description**

Overall description I discuss the product features and operations performed by the system which I implemented. First system shows grid menu to the user which comprises of lawyer details, his office details, staff details, gallery and an appointment option. System also shows login option to admin. After login into the system, it shows different options to admin.

### **2.2.1) Product Perspective**

Case Management System is a standalone system. It serves only those users who have an internet connection. Admin can see the requests for appointments. Admin can share news about cases. Admin can maintain client's profiles. Admin can record cases relevant information. Admin can search cases by extensible search engine. Admin can share the pictures of events. Admin can schedule reminders for appointments. Admin can add, delete or update staff details.

#### **2.2.1.1) System Interfaces**

Case Management System is a web-based system. Case Management System uses a server which holds system database. The admin updates the database after logging in the system. The users access this system through laptop or computer using internet. After login, admin can perform different functions.

#### **2.2.1.2) User Interfaces**

In the user interface I describe how the user interacts with the system. The system which I implemented is web-based. To use this, admin must have authenticated id and password.



### **2.2.1.3) Software Interfaces**

This web-based system can access by any type of browser there is no restriction of web browser, but most preferred is to use the latest browsers. There is no restriction of the operating system. The user uses this system on any kind of operating system.

### **2.2.1.4) Hardware Interfaces**

There is no hard and tough hardware required to run our application. Even it can be run using processor with Windows XP. The website can be open in any browser using any device. But it is recommended to use better machine to run website perfectly. Hardware requirements for client and server side are:

## **2.2.2) Product Functions**

A Product functions provide a summary of the major functions that the software will perform. Some of the major functions which I implemented in the software are listed below:

### **2.2.2.1) Request to login**

The admin enters his id and password to use the application. Only authenticated users will be able to use the application.

### **2.2.2.2) Manage Cases Information**

Admin manages the records of different cases. Admin can add, delete, update and view the cases records.

### **2.2.2.3) Manage Client's information**

Admin manages the client's information. Admin can add, delete, update and view client's information.

### **2.2.2.4) Manage Staff Information**

Admin manages the staff information. Admin can add, delete, update and view the staff information.

### **2.2.2.5) Manage Search Engine**

Admin manages the search engine for cases. Admin can search case by remembering judge name, date, year and parties name.

### **2.2.2.6) Manage User's Information**

User's submits forms for appointments. Admin will approve user's requests for appointments.

### **2.2.2.7) Share News about Cases**

Admin can share the news about important cases like report sharing etc.

### **2.2.2.8) Request to get Appointment**

User can request for the appointment of lawyer by choosing date which suits him. Admin then approve the request the request of user for appointment. User will not register himself for request of appointment.

### **2.2.2.9) Sharing of Event Details**

Admin can share the details of events like which events are going to be held under his supervision.

### **2.2.2.10) Schedule Reminder for Appointments**

Admin can schedule the reminder for his appointments so that he can never forget about his meetings/appointments.

### **2.2.2.11) Schedule Reminder for Events**

Admin can schedule reminder for events so that he can never forget about events.

## **2.2.3) User Characteristics**

This is assumed that users know English language and can read and write it. Users must have knowledge of how to use a computer/laptop and related applications. And admin also must have knowledge of how to use computer/laptop and related applications.

## **2.2.4) Constraints**

To operate this system, the user must have a laptop/computer and the main thing is that it has an internet connection. Without internet connection, the user cannot operate this system.

## **2.2.5) Assumptions and Dependencies**

The system which I implemented is totally web-based system, the user must have internet connection to operate this system. Without the internet connection, the user cannot operate this system.

## **2.3) Specific Requirements**

There is no specific requirement of the implemented system. There are some functional and non-functional requirements.

### **2.3.1) Functional Requirements**

The system shall validate user id and password for admin. The system shall allow admin to add, modify and delete different cases records, staff details, and maintain client's data, and search cases by remembering judge name, year, parties name etc. The system shall allow admin to approve user's requests for appointments and can schedule reminders for appointments. The system shall allow admin to share news about cases and share the details about events. The system shall not store any information about problems.

## **2.3.2) Non-functional Requirements**

The non-functional requirements are those which are not directly involved in the system. The load time for admin interface screens shall take no longer than milliseconds. The login information shall be verified within a second. Queries shall return results within milliseconds.

### **2.3.2.1) Security**

This web-based Application can only use by lawyer which have valid id and password. The people who don't register like clients can only see the details about office, lawyer, and news about lawyer cases, gallery, and can request for appointment. If we authenticate the client's, then we can communicate with them on website and for that in future we can use encryption and decryption methods.

### **2.3.2.2) Reliability**

The system which I have implemented is very much reliable and responds well whenever user makes any sort of query. The system processes the query and sends back the exact results. The system does not send out any ambiguous results. The admin and users can rely on the system. The system will give accurate results.

### **2.3.2.3) Availability**

The system is available all the time. When the user makes any query, the database can send the results within milliseconds.

### **2.3.2.4) Performance**

The load time for user interface screen shall take no longer than milliseconds. The login information shall be verified within a second. Queries shall return results within milliseconds specific as possible.

### **2.3.2.5) Maintainability**

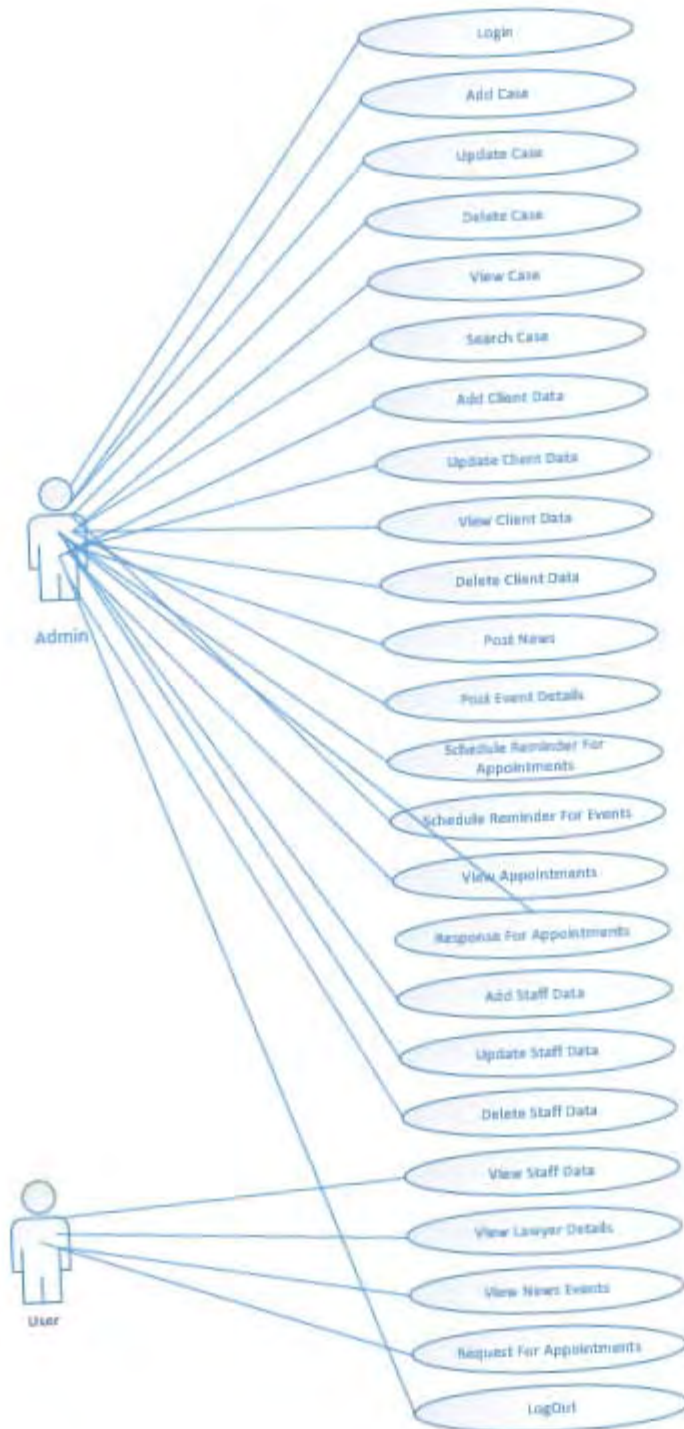
The database may crash at any certain time due to virus or an operating system failure. Therefore, it is required to keep back up of all data on the database to avoid any data loss.

## **2.4) List of Use Cases**

1. Login.
2. Add Case.
3. Update Case.
4. Delete Case.
5. View Cases.
6. Search Cases.

7. Add Client's Data.
8. Update Client's Data.
9. Delete Client's Data.
10. View Client's Data.
11. Post News.
12. View News.
13. Schedule Reminder for Events.
14. Schedule Reminder for Appointments.
15. Request for Appointment.
16. Response for Appointment.
17. View Appointments.
18. Add Staff Data.
19. Update Staff Data.
20. Delete Staff Data.
21. View Staff Data.
22. View Lawyer Details.
23. Post Event Details
24. Log out

## 2.4.1) Use Case Diagram



**Fig 2.1 Use Case Diagram**

## 2.5) Use Case Descriptions

The use-case diagram is a graphical representation of a user's interaction with the system. Use-case diagram can portray different types of users of a system and ways that interact with the system. A list of use-cases mentioned in use-case diagram are described in detail, so that we can look more precisely that how user can interact with the system to perform tasks.

### Use Case Details

#### 2.5.1) Login

This use case describes how the admin login into the system, the admin selects login button. The login screen appears, the admin enters the name and password. The system will check the admin name and password. The admin will log in.

<b>Name</b>	<b>UC1: Login</b>
<b>Primary Actor</b>	Administrator
<b>Pre-Conditions</b>	Administrator is registered on system.
<b>Post-Conditions</b>	Administrator will be logged in.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Administrator selects login option.</li><li>2. System asks to enter id and password.</li><li>3. Administrator adds id and password.</li><li>4. System will check id and password.</li><li>5. Administrator will be logged in.</li></ol>
<b>Alternative Scenario</b>	A*. System fails at any time. <ol style="list-style-type: none"><li>1. System rolls back all changes made by administrator.</li><li>3a. ID or password is incorrect.<ul style="list-style-type: none"><li>• System prompts admin to enter correct information.</li></ul></li></ol>

## 2.5.2) Add Case

This use-case describes how admin add case. Admin selects option of add case, dropdown menu appears about which type of case admin wants to add and then after selection of desired case, a form will appear. Admin fills the form and then select add button, case will be added.

<b>Name</b>	<b>UC2: Add Case</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin will successfully add new case.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin selects the option of add case.</li><li>2. System shows the drop-down menu listing different types of court cases.</li><li>3. Admin selects one option from drop-down menu.</li><li>4. System shows the form for add case.</li><li>5. Admin fills the form for adding case.</li><li>6. System saves the new case record in database.</li></ol>
<b>Alternative Scenario</b>	A*. System fails at any time. <ol style="list-style-type: none"><li>1. System rolls back all changes made by admin.</li><li>5. Admin not fills the form properly.<ol style="list-style-type: none"><li>a. System ask the user to fill the form properly.</li></ol></li></ol>

### 2.5.3) View Case

This use-case describes how admin view the case. Admin selects view case option; drop-down menu appears which shows, which kind of case you want to view. Admin then selects desired case and system will display it to admin.

<b>Name</b>	<b>UC3: View Case</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin will successfully view the desired case.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin selects the option of view case.</li><li>2. System shows drop-down menu listing different types of cases.</li><li>3. Admin selects desired option for viewing case.</li><li>4. System shows the list of different cases.</li><li>5. Admin selects the desired case for viewing.</li><li>6. System shows the details of that case.</li></ol>
<b>Alternative Scenario</b>	A*. System fails at any time. <ol style="list-style-type: none"><li>1. Admin selects unwanted case option in hurry.<ol style="list-style-type: none"><li>a. So, admin must get back to choose the required case.</li></ol></li></ol>



## 2.5.4) Update Case

This use case describes how user updates the desired case. Admin search for desired case then admin view the desired case, there will be an option of update case information. Admin selects edit option, system will display the form so admin according to his need can modify case details by making changes in form and then he can press update button for saving changes.

<b>Name</b>	<b>UC4: Update Case</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin will successfully update the desired case information.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin search the case through search engine.</li><li>2. Admin selects the desired case.</li><li>3. System shows case details of selected case and an update option.</li><li>4. Admin selects edit option</li><li>5. System displays form in which admin can modify the case details.</li><li>6. Admin selects update button and changes will be saved in database.</li></ol>

## 2.5.5) Delete Case

This use case describes how admin deletes case. Admin will search the case which he wants to delete. Admin selects the case; there will be a delete option. Admin choose delete option and system will delete that case information from database.

<b>Name</b>	<b>UC5: Delete case</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin will successfully delete the case.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin search the case through search engine.</li><li>2. Admin selects the desired case.</li><li>3. System show the details of desired case and there will be a delete option.</li><li>4. Admin selects delete option.</li><li>5. System will delete that case from database.</li></ol>
<b>Alternative Scenario</b>	A*. System fails at any time. <ol style="list-style-type: none"><li>1. Admin misspells while searching case.<ol style="list-style-type: none"><li>a. System prompts to admin that entered case doesn't exist. Enter correct details.</li></ol></li></ol>

## 2.5.6) Search Case

This use case describes how admin search case. Admin enters case details for searching a case. System shows that case.

<b>Name</b>	<b>UC6: Search case</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin successfully got required case.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin selects search option.</li><li>2. System displays drop-down menu.</li><li>3. Admin choose desired option.</li><li>4. Admin enters details for searching.</li><li>5. System shows the desired case.</li></ol>
<b>Alternative Scenario</b>	A*. System fails at any time. <ol style="list-style-type: none"><li>1. Admin misspells while searching case.<ol style="list-style-type: none"><li>a. System prompts to admin that entered case doesn't exist. Enter correct details.</li></ol></li></ol>

## 2.5.7) Add Client's Data

This use case describes how system adds client's data. Admin selects add client data option. System shows the form. Admin fills the form by providing Client's data. Admin selects save option and client's data will be saving in database.

<b>Name</b>	<b>UC6: Add Client's data</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin will successfully add client's data.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin selects client's option.</li><li>2. System displays a screen which shows add client's data.</li><li>3. Admin selects add client's data.</li><li>4. System shows the form for adding client's data.</li><li>5. Admin fills the form and selects save option.</li><li>6. System saves client's data in database.</li></ol>
<b>Alternative Scenario</b>	<p>A* System fails at any time.</p> <p>5.Admin doesn't fill the required fields.</p> <ol style="list-style-type: none"><li>a. System prompts the admin to fill the required details.</li></ol>

## 2.5.8) View Client's Data

This use case describes how admin view client's data. Admin selects client's option. System shows a drop-down menu listing different scenarios. Admin choose View Client's data; system shows client's data in a list.

<b>Name</b>	<b>UC8: View client's data</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin successfully view the client's data.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin selects Client's option.</li><li>2. System shows drop-down menu listing different scenarios.</li><li>3. Admin choose the required option like viewing client's data.</li><li>4. System shows the client's data.</li></ol>
<b>Alternative Scenario</b>	A* System fails at any time.

## 2.5.9) Update Client's Data

This use case describes how user can update client's data. Admin selects client's option. System shows a listing indicating an option of view client's data. Admin selects view option. System lists client's data. Admin choose specific client's data, there will be an edit option. Admin selects edit option. System displays form and admin can modify the form according to his needs.

<b>Name</b>	<b>UC9: Update Client's data</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin successfully update client's data.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. Admin selects client's option.</li> <li>2. System shows drop-down menu listing a view options.</li> <li>3. Admin selects specific view option.</li> <li>4. System lists different client's data.</li> <li>5. Admin selects specific client's data and There will be an edit option.</li> <li>6. Admin selects edit option.</li> <li>7. System displays form.</li> <li>8. Admin make changes in that form according to his needs.</li> <li>9. Admin selects update button and data will be updated in database.</li> </ol>
<b>Alternative Scenario</b>	A* System fails at any time.

## 2.5.10) Delete Client's Data

This use case describes how user can delete client's data. Admin selects client's option. System shows a listing indicating an option of view client's data. Admin selects view option. System lists client's data. Admin choose specific client's data, there will be a delete option. Admin selects delete option. System will delete selected client's data from database.

<b>Name</b>	<b>UC10: Delete Client's data</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin successfully delete the client's data.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin selects client's option.</li><li>2. System shows drop-down menu listing a view options.</li><li>3. Admin selects specific view option.</li><li>4. System lists different client's data.</li><li>5. Admin selects specific client's data and there will be a delete option.</li><li>6. Admin selects delete option and system delete selected client's data from database.</li></ol>
<b>Alternative Scenario</b>	A*. System fails at any time.

## 2.5.11) Post News

This use case describes how admin post news. Admin selects post option. System displays form. Admin fills the form and press share option, news will be posted.

<b>Name</b>	<b>UC11: Post News</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin successfully post the news.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin selects post news option.</li><li>2. System shows a form.</li><li>3. Admin fills the form, and press share button, news will be posted.</li><li>4. And system saves that news in database.</li></ol>
<b>Alternative Scenario</b>	<p>A* System fails at any time.</p> <ol style="list-style-type: none"><li>3, Admin makes mistakes while filling form.<ol style="list-style-type: none"><li>a. System prompts admin to enter correct details.</li></ol></li></ol>



## 2.5.12) Post Event Details

This use case describes how admin post event details. Admin selects post event details option. System shows a form to admin. Admin fills that form and selects share option. Details of event will be posted.

<b>Name</b>	<b>UC12: Post Event Details</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin successfully post event details.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin selects post event details option.</li><li>2. System displays form to admin.</li><li>3. Admin fills that form and there will be a share button option.</li><li>4. Admin selects share button option and details will be posted.</li><li>5. System also saves that details in database.</li></ol>
<b>Alternative Scenario</b>	<p>A*. System fails at any time.</p> <ol style="list-style-type: none"><li>3, Admin makes mistake while filling form or left some fields empty<ol style="list-style-type: none"><li>a. System prompts the admin to enter correct details.</li></ol></li></ol>

## 2.5.13) View News and Events Details

This use case describes how user can view news and event details. User come on site's main page, system automatically display news and event details in a slide show but both separately.

<b>Name</b>	<b>UC13: View News and Events Details</b>
<b>Primary Actor</b>	Guest User
<b>Pre-Conditions</b>	Guest user must have internet connection to access this site.
<b>Post-Conditions</b>	Guest user have successfully view news and events details.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User visits the site.</li><li>2. User view the slide show for news events.</li></ol>
<b>Alternative Scenario</b>	A* System fails at any time. <ol style="list-style-type: none"><li>a. User internet connection fails.</li></ol>

## 2.5.14) Schedule Reminder for Events

This use case describes how admin schedule reminder for appointments. Admin selects schedule reminder option. System displays a form. Admin fills that form and selects schedule option. Form's data will be saved in database. Admin get notify before 1 day for event.

<b>Name</b>	<b>UC14:</b> Schedule reminder for event
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin successfully schedule reminder for events.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin selects schedule reminder for event option.</li><li>2. System displays a form.</li><li>3. Admin fills the form and press schedule button.</li><li>4. Reminder will be scheduled.</li><li>5. Admin get notify before a day about event.</li></ol>
<b>Alternative Scenario</b>	A* System fails at any time <ol style="list-style-type: none"><li>3, Admin left some blanks unfilled.<ol style="list-style-type: none"><li>a. System prompts the user to enter correct details.</li></ol></li></ol>

## 2.5.15) Request for Appointments

This use case describes how user request for an appointment. User selects apply for appointment option. System displays the form. User fills the form and press apply button. System saves the data in database. Users request will be forwarded to admin.

<b>Name</b>	<b>UC15:</b> Request for appointment
<b>Primary Actor</b>	User
<b>Pre-Conditions</b>	User must have internet connection to access site.
<b>Post-Conditions</b>	User successfully make request for appointment.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User visits the site.</li><li>2. User selects request for appointment option.</li><li>3. System displays form to user.</li><li>4. User fills in the form.</li><li>5. System saves users form data and forward user request to admin.</li></ol>
<b>Alternative Scenario</b>	A*. System fails at any time. <ol style="list-style-type: none"><li>4, User makes mistakes by filling form.<ol style="list-style-type: none"><li>a. System prompts the user to enter correct details.</li></ol></li></ol>

## 2.5.16) View Appointments

This use case describes how admin view appointments. Admin selects view appointments option. System displays user request for appointments, and scheduled appointments.

<b>Name</b>	<b>UC16: View Appointments</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin successfully view appointments.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1) Admin selects appointments option.</li><li>2) System list down the user requests for appointments And scheduled appointments.</li><li>3) Admin selects the desired appointment for checking out the details of that appointment.</li></ol>
<b>Alternative Scenario</b>	A* System fails at any time.

## 2.5.17) Response for Appointments

This use case describes how admin response to user appointments. Admin selects view appointments option, system displays a list of user's appointments and then admin selects a request of appointment. Admin either approve it or reject it.

<b>Name</b>	UC17: Response for Appointments
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin successfully respond to user's requests.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin selects view appointment option.</li><li>2. System displays a list of user's requests for appointments.</li><li>3. Admin selects specific user request for appointment.</li><li>4. Admin responds to selected request.</li><li>5. Admin either approve or discard user request.</li><li>6. System saves the accepted requests in database.</li></ol>
<b>Alternative Scenario</b>	A*- System fails at any time.

## 2.5.18) Schedule Reminder for Appointment

This use case describes how admin schedule reminder for appointment. Admin selects appointment option. System displays the list of approved appointments by admin. Admin schedule reminder for specific appointment. System notify the admin before a day for his scheduled appointments.

<b>Name</b>	<b>UC18:</b> Schedule reminder for appointment
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin successfully schedule reminder.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin selects appointments option.</li><li>2. System displays a list of approved appointments by admin.</li><li>3. Admin selects an appointment and then schedule reminder for it.</li><li>4. System notify the admin before a day for his scheduled appointment.</li></ol>
<b>Alternative Scenario</b>	A* System fails at any time.

## 2.5.19) Add Staff Data

This use case describes how admin add staff data. Admin selects Staff option; drop-down menu appears which shows different options. Admin selects add staff data option. System displays a form; Admin fills the form. Staff data will be saved in database.

<b>Name</b>	<b>UC19: Add staff data</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin successfully add staff data.
<b>Main-Scenario</b>	<ol style="list-style-type: none"><li>1. Admin selects staff option.</li><li>2. System display drop-down menu</li><li>3. Admin selects add staff data option.</li><li>4. System displays form.</li><li>5. Admin fills the form to add staff data.</li><li>6. Admin press add button to add staff data.</li><li>7. System saves form data in database.</li></ol>
<b>Alternative Scenario</b>	A* System fails at any time.  <ol style="list-style-type: none"><li>5, Admin makes mistakes while filling form.<ol style="list-style-type: none"><li>a. System prompts the user to enter correct details.</li></ol></li></ol>



## 2.5.20) View Staff Data

This use case describes how admin view staff data. Admin selects staff option; system displays drop-down menu. Admin selects view staff data option; system displays a list of staff data.

<b>Name</b>	<b>UC20: View staff data</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin successfully view staff details.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin selects staff option.</li><li>2. System displays droop-down menu listing different categories.</li><li>3. Admin selects View staff data option.</li><li>4. System displays staff details in a list.</li></ol>
<b>Alternative Scenario</b>	A* System fails at any time.

## 2.5.21) Update Staff Data

This use case describes how admin update staff data. Admin selects staff option; system displays drop-down menu. Admin selects view staff data. System displays a list of staff data. Admin selects specific staff data, there will be edit option. Admin selects edit option, System displays a form and then admin makes changes according to his needs in a form and then press update button. Updated data will be saved in a database.

<b>Name</b>	<b>UC21: Update Staff Data</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin successfully updated the staff data.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. Admin selects staff option.</li> <li>2. System displays drop-down menu.</li> <li>3. Admin selects view staff option.</li> <li>4. System displays a list of staff data.</li> <li>5. Admin selects a specific staff data.</li> <li>6. System displays a form.</li> <li>7. Admin make changes in the form.</li> <li>8. Admin press update button and updated data will be saved in database.</li> </ol>
<b>Alternative Scenario</b>	<p>A* System fails at any time.</p> <p>7, Admin makes mistakes while changing the form.</p> <ol style="list-style-type: none"> <li>a. System prompts the user to enter correct details.</li> </ol>

## 2.5.22) Delete Staff Data

This use case describes how admin delete staff data. Admin selects staff option; system displays drop-down menu. Admin selects view staff data. System displays a list of staff data. Admin selects specific staff data, there will be delete option. Admin selects delete option; system delete specific staff data from the database.

<b>Name</b>	<b>UC22: Delete Staff Data</b>
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin successfully delete specific staff data.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin selects staff option.</li><li>2. System displays drop-down menu.</li><li>3. Admin selects view staff option.</li><li>4. System displays a list of staff data.</li><li>5. Admin selects a specific staff data.</li><li>6. Admin press delete button.</li><li>7. System delete specific data from Database.</li></ol>
<b>Alternative Scenario</b>	A* System fails at any time.

## 2.5.23) View Lawyer Details

This use case describes how user view the lawyer details. User visits the site and then selects about lawyer option. System displays information about lawyer and his office.

<b>Name</b>	<b>UC23: View lawyer Details</b>
<b>Primary Actor</b>	User
<b>Pre-Conditions</b>	User must have internet connection to access this site.
<b>Post-Conditions</b>	User successfully view the lawyer details.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User visits the site.</li><li>2. User selects view lawyer option.</li><li>3. System displays lawyer's details.</li></ol>
<b>Alternative Scenario</b>	A*. System fails at any time.

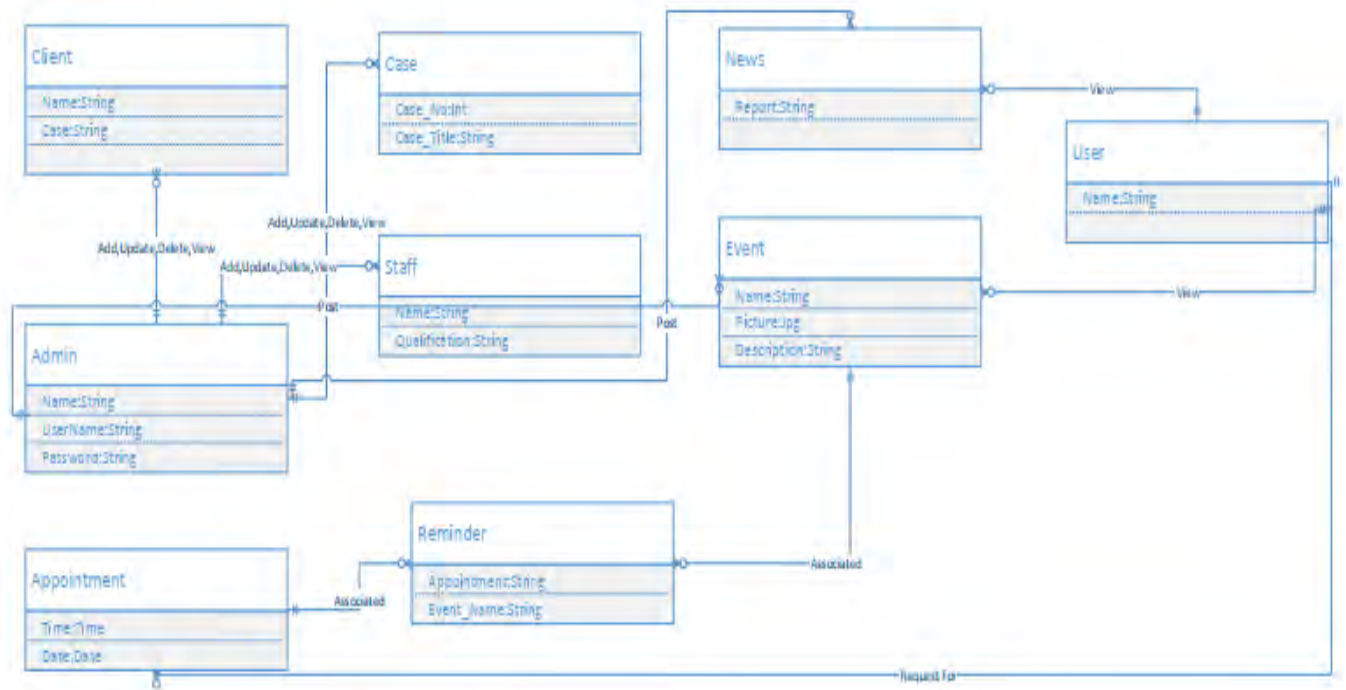
## 2.5.24) Logout

This use case scenario tells how the admin logout from the system. Admin selects the option of logout and system shows main screen.

<b>Name</b>	UC24: Logout
<b>Primary Actor</b>	Admin
<b>Pre-Conditions</b>	Admin is authenticated and logged in.
<b>Post-Conditions</b>	Admin successfully logout from system.
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. Admin clicks logout button.</li><li>2. System shows the message you logout successfully from system.</li><li>3. System shows the main screen.</li></ol>
<b>Alternative Scenario</b>	<p>A*. System fails at any time.</p> <ol style="list-style-type: none"><li>a. System rolls back all changes made by admin.</li></ol>

## 2.6) Domain Model

Domain model is an object model of problem domain. It is based on real world classes/concepts and their relationships, that is used to identify the relationships among all the entities within the scope of problem domain.



**Figure 2.2 Domain Model**

**Notations used in diagram: -**

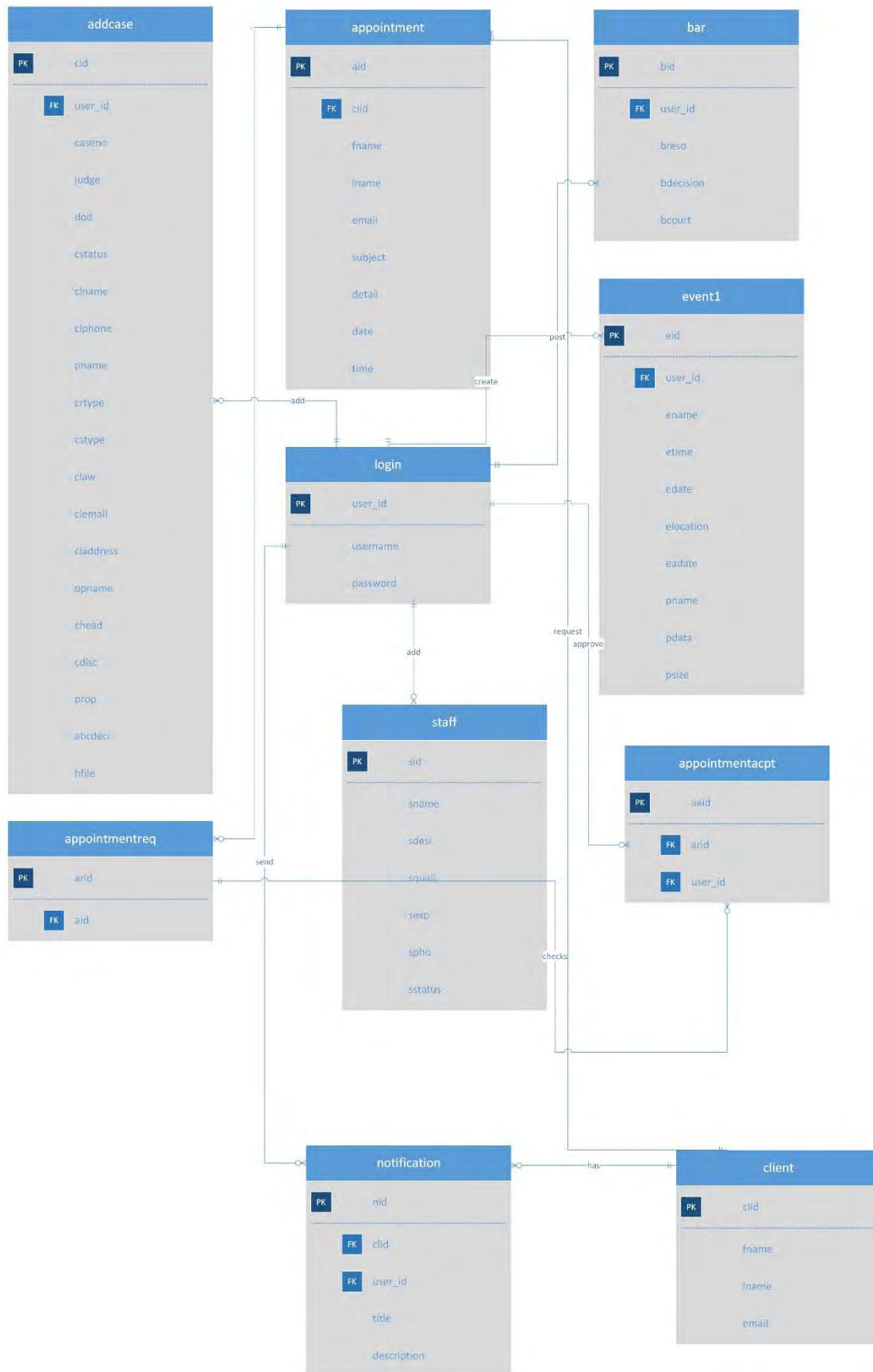


Crow foot's database notation and it means 1.1



It's also Crow foot's database notation and it means 0 to many.

## 2.7) Database Requirements



46 Figure 2.3 ERD

# Chapter 3

## Software Design Description

This chapter specifies the general factors that can affect the product and its requirements, providing a background for the requirements of the software. To describe the key concepts of the problem domain and data items used in our system, the following models have been used system sequence diagram, sequence diagram and class diagram.

Software Design Description (SDD) is a representation of a system that how the users can interact with the system. This can be shown with the help of diagrams. The SSD shows that the system is completely fulfill the requirement.

### 3.1) Design Overview

In the design overview, we can describe the system at architecture level and the complete structure of the system. In this we can explain how the user interacts with the system. In the class diagram, we can show that how the classes can interact with each other define the relationship. Sequence diagram that is an interaction diagram, which shows the sequence of messages interacting with objects [3].

#### 3.1.2) Requirements Traceability Matrix

Requirement Id	Requirement Name	Sequence Diagram	Test Case	System Sequence Diagram
UC:1	Login	Fig 3.4.1	5.3.1	Fig 3.2.1
UC:2	Add case	Fig 3.4.3	5.3.7	Fig 3.2.3
UC:5	View Case	Fig 3.4.6		Fig 3.3.5
UC:7	Add Client	Fig 3.4.4	5.3.9	Fig 3.3.4
UC:11	Post News	Fig 3.4.2	5.3.3	Fig 3.3.2
UC:14	Reminder for Appointment	Fig 3.3.7		Fig 3.3.6



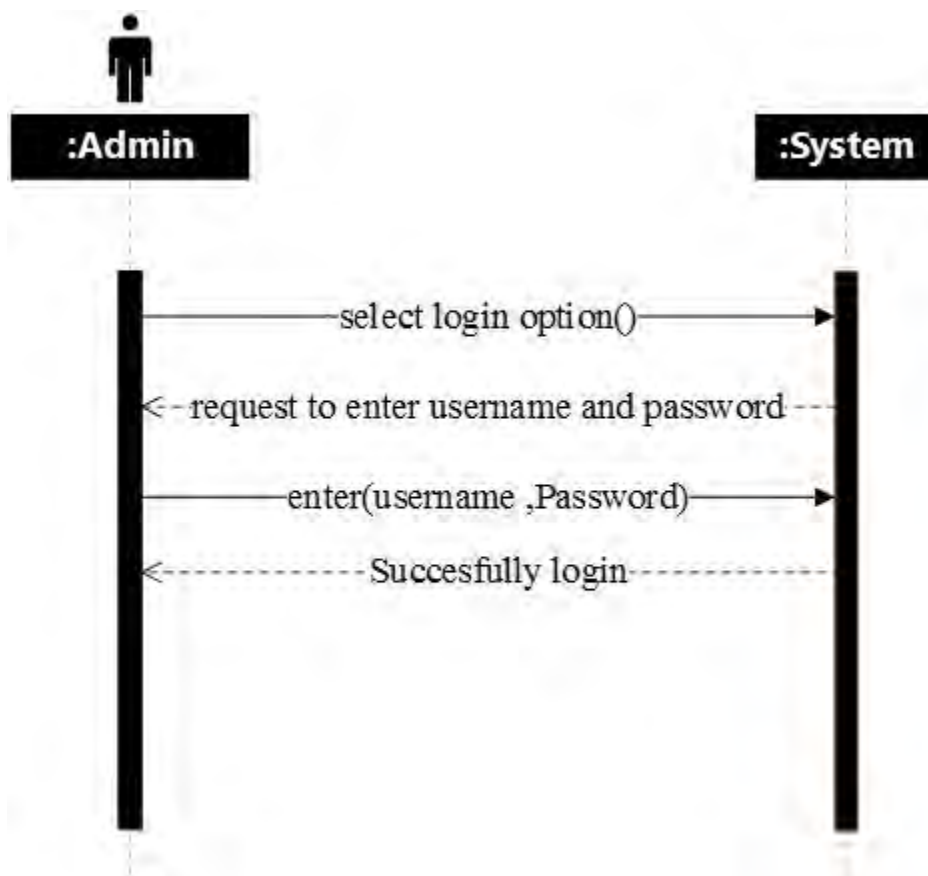
UC:17	View Appointments		5.3.5	Fig 3.3.7
-------	----------------------	--	-------	-----------

**Table of Matrix 3.1**

### 3.2) System Sequence Diagram

A system sequence diagram illustrates input and output events related to our system. System is treated as a black box and the emphasis of the diagram is events that are generated by system for a particular scenario of a use-case.

#### 3.2.1) SSD login



**Figure 3.1 SSD Login**

### 3.2.2) Post News

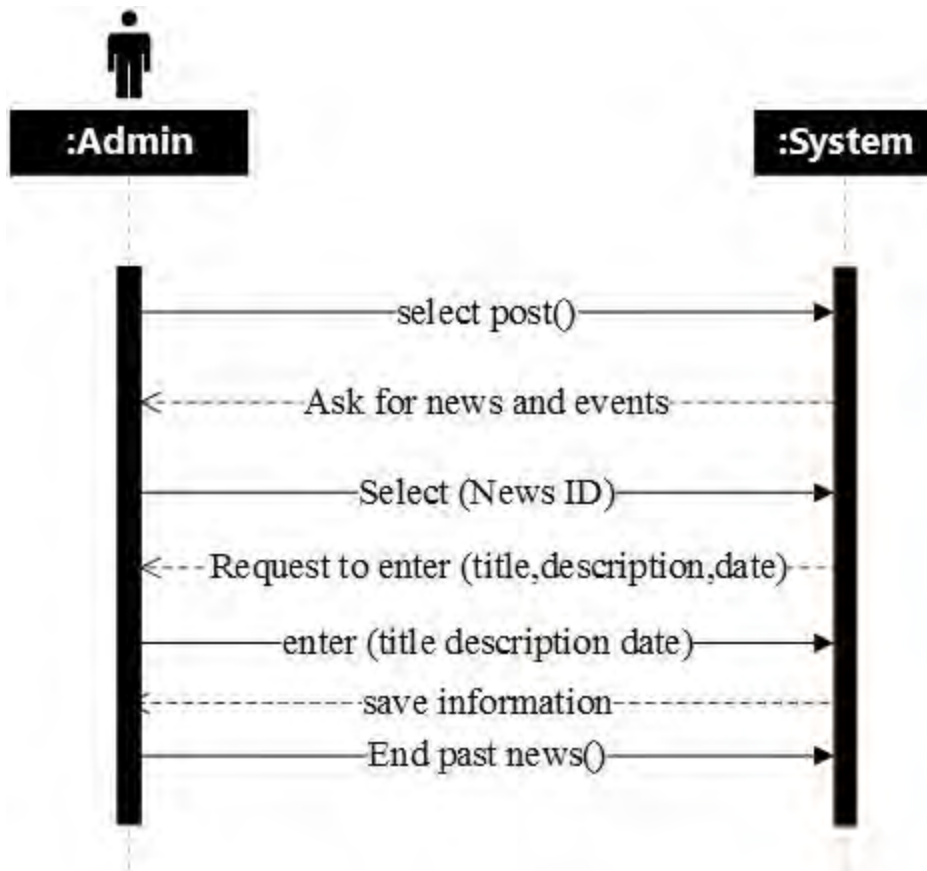


Figure 3.2 SSD Post News

### 3.2.3) Add Case

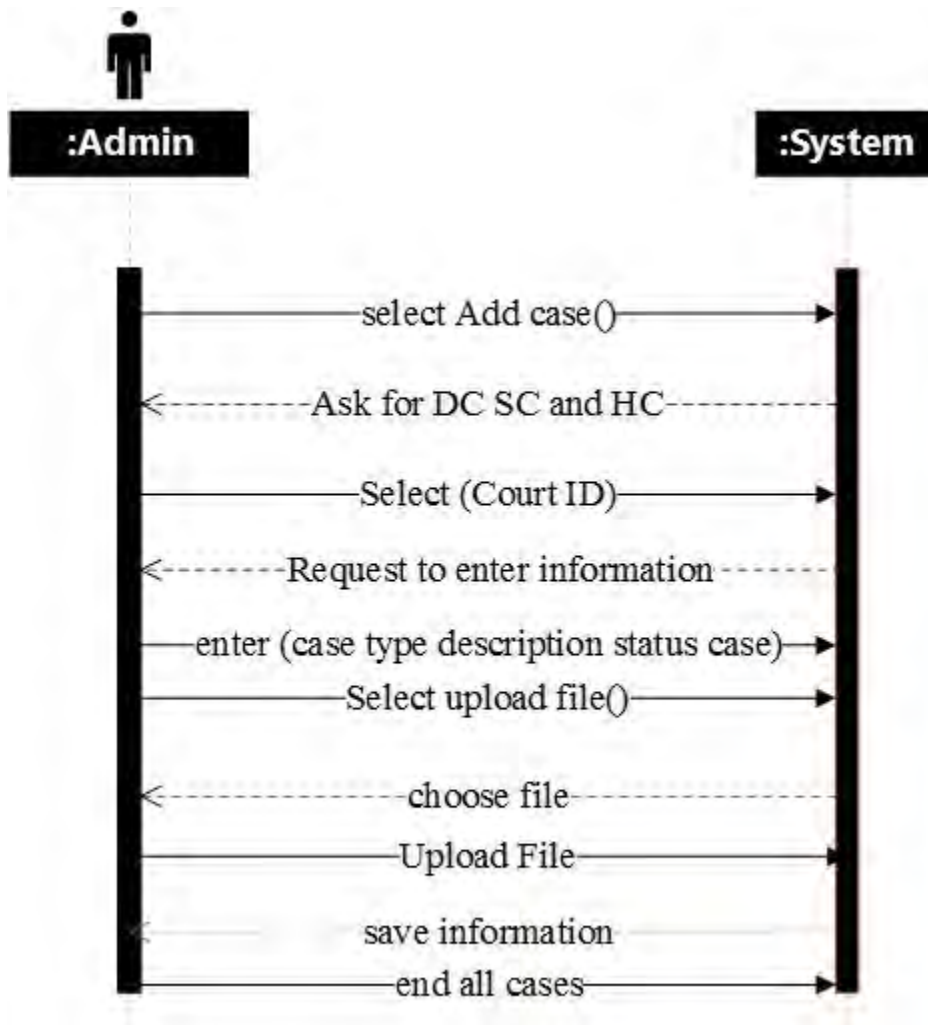


Figure 3.3 SSD Add Case

### 3.2.4) Add Client

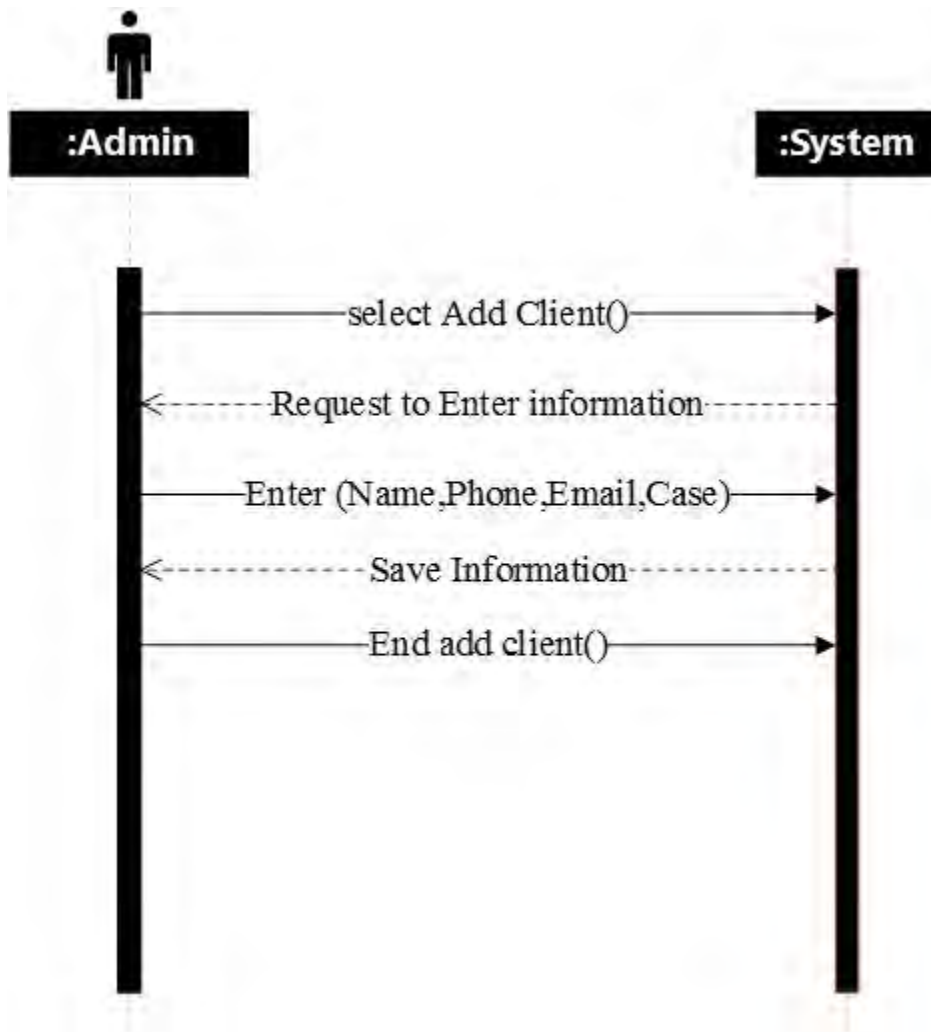


Figure 3.4 SSD Add Client

### 3.2.5) View Case

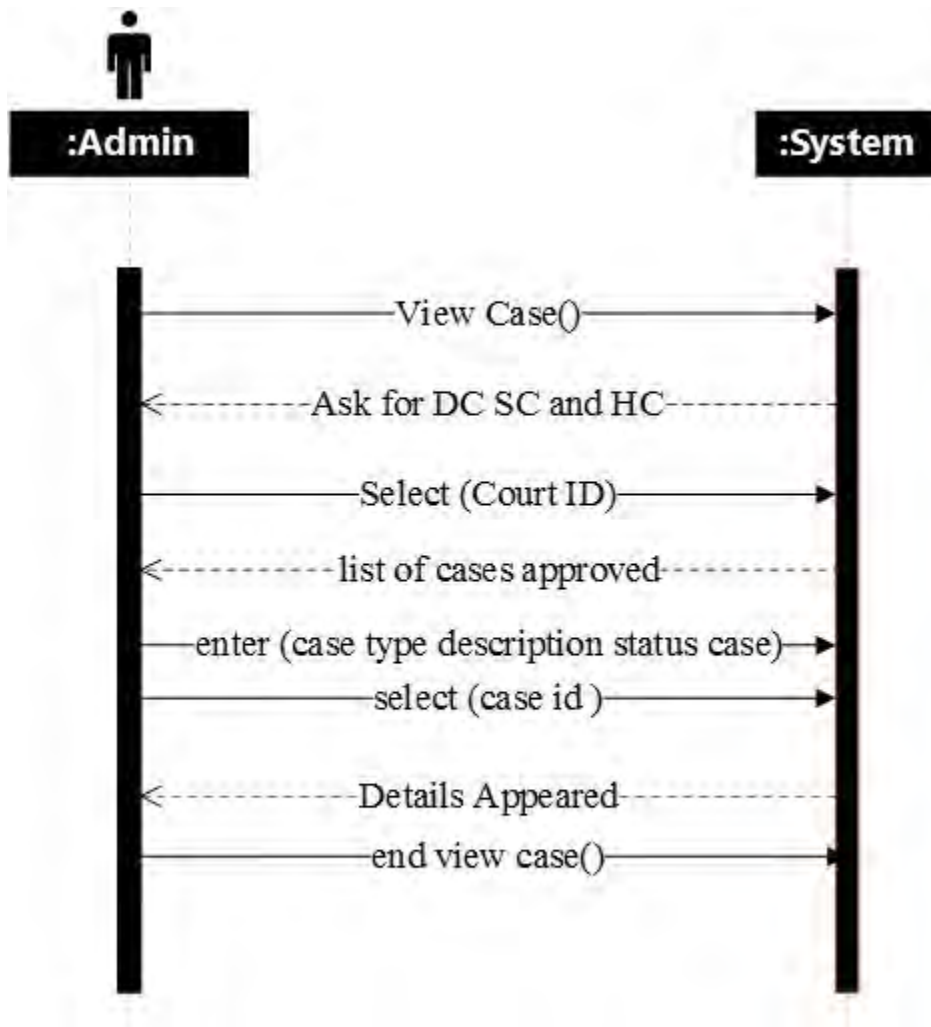


Figure 3.5 SSD View Case

### 3.2.6) Reminder for Appointments

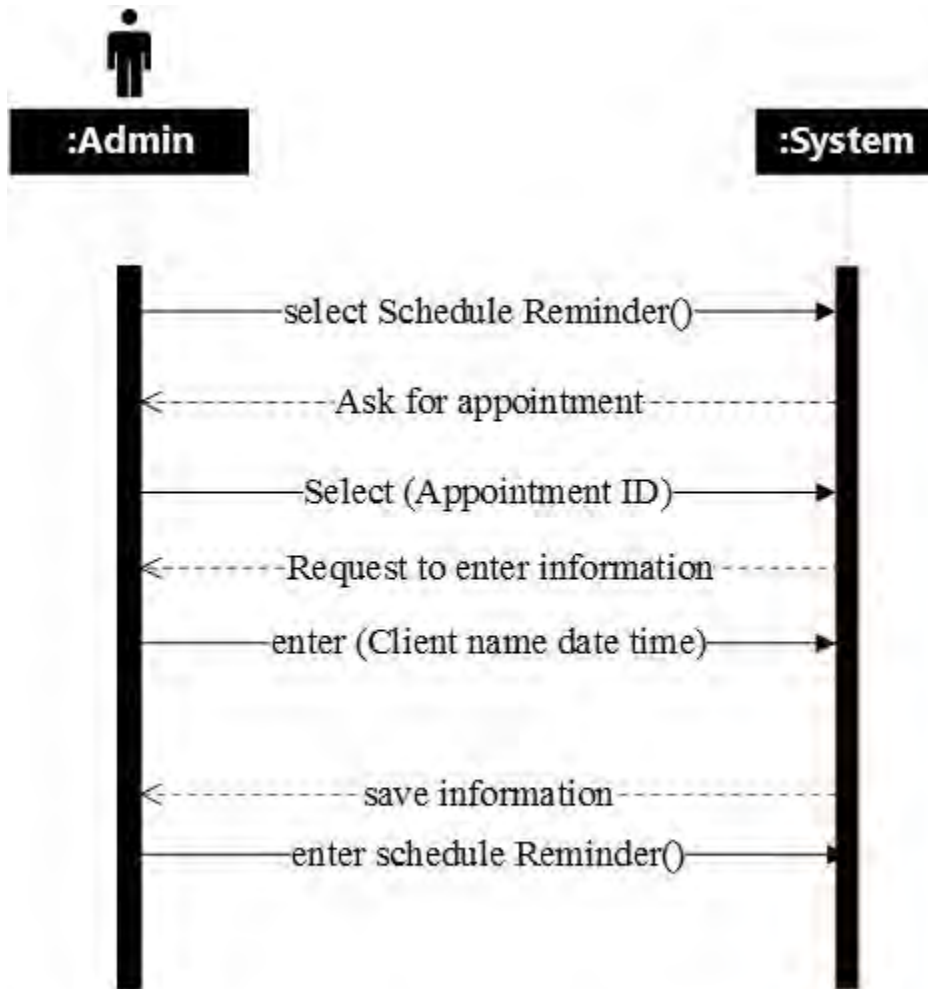


Figure 3.6 SSD Reminder for Appointments

### 3.2.7) View Appointments

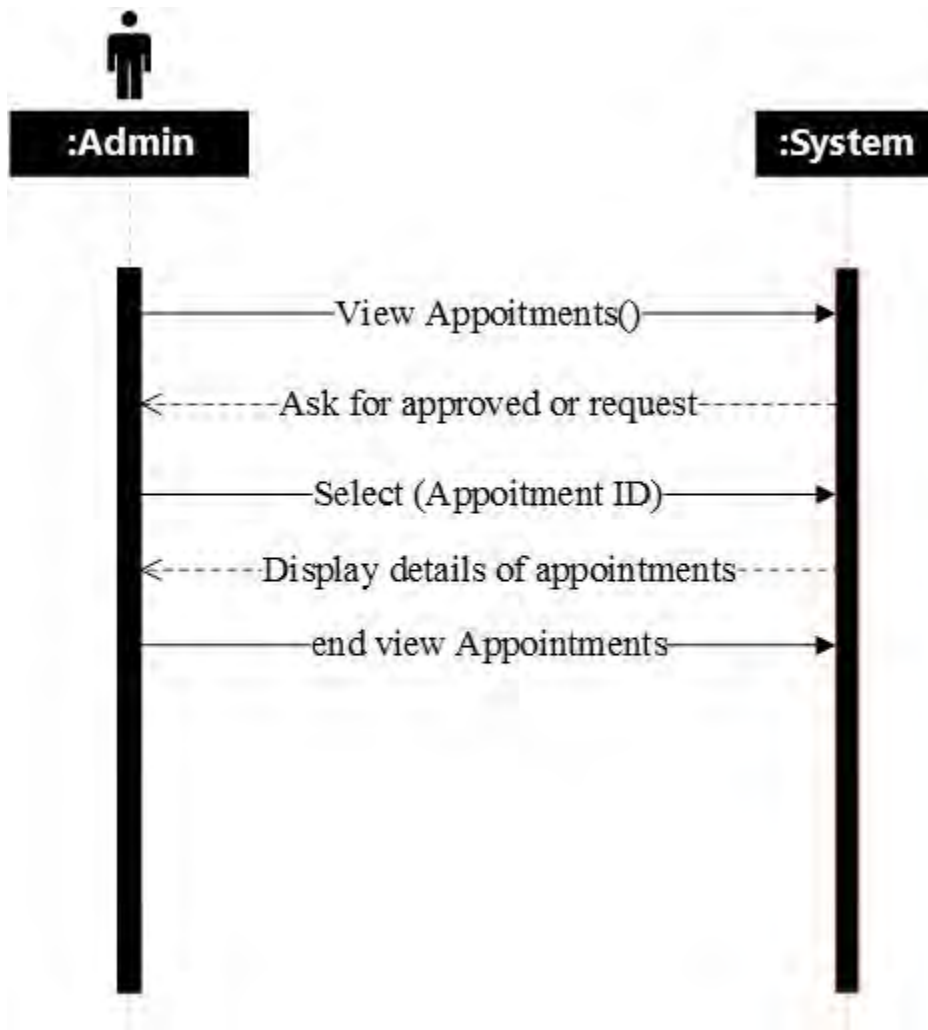
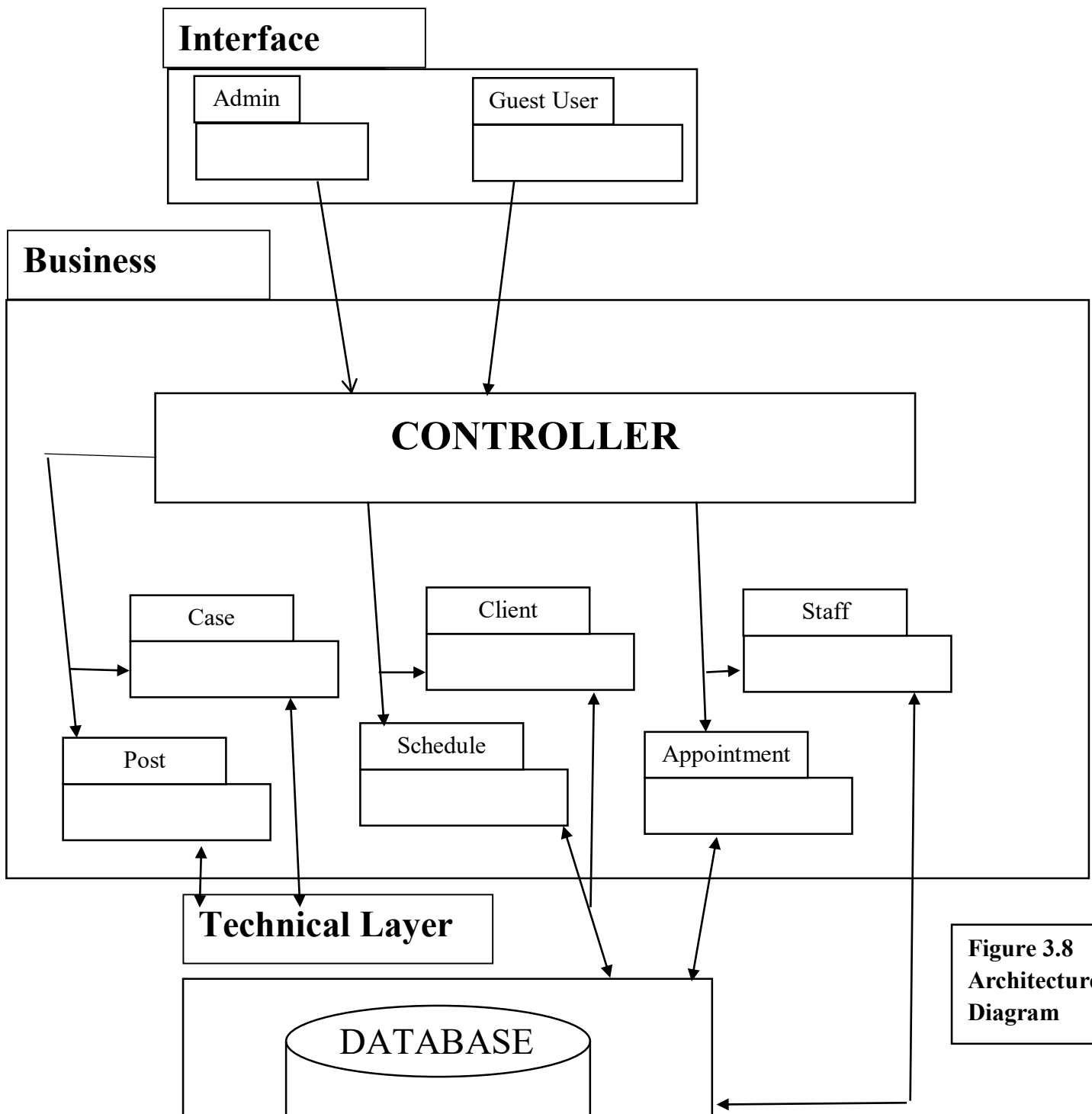


Figure 3.7 SSD View Appointments

### 3.3) Architecture Diagram

Architecture Diagram is used to represent the components of system and interaction between them. System under discussion is based on “Three Tier” Architectural pattern. Three-tier architecture is a software architecture pattern in which the user interface (presentation), functional process logic

(business rules), computer data storage and data access are developed and maintained as independent modules. Interacting between components of system is shown in diagram. Double arrows represents the interaction from both sides. Similarly, single arrow represents one-way interaction. The singular quality of a three-tier architecture is the separation of the application logic into a distinct logical middle tier of software. The interface tier is relatively free of application processing; windows or web pages' forward task requests to the middle tier. The middle tier communicates with the back-end storage layer. It is possible to make changes on the presentation level without affecting the other two (business or data access layer). As each tier is independent it is possible to use different sets of developers since the client doesn't have direct access to the database. Business logic are more secure.



**Figure 3.8**  
**Architecture**  
**Diagram**



## 3.4) Interface design

### 3.4.1) Simple and Appealing

The interface is simple to use, a naïve user can also use it very easily and efficiently to get there required information.

### 3.4.2) Responsive

The interface is responsive this means it can adjust its size according to device's display Like (small screens, large screens) etc.

## 3.5) Interfaces

### 3.5.1) Interface Login

This picture will show the interface of login screen. Which is used in the project. It can contain two field of username and password.

**ADMIN LOGIN**

Username

Passsword

LOGIN

**Figure 3.9 Interface Login**

### 3.5.2) Post Event

This picture will show the interface of post event. Which is used in the project. It can contain field of event name, date, time, location, alert date and upload image.

The screenshot shows a web application interface with a sidebar on the left and a main content area. The sidebar has a 'Home' button and a list of menu items: Staff, Schedule, Event, Appointment, Post, News, About Case, Bar Activity, Event, View, and Staff. The main content area displays an 'Event Form' with the following fields: EventName, Time, Date, Location, and Alert Date. There are also 'UPLOAD IMAGE' and 'CANCEL' buttons at the bottom of the form.

Figure 3.10 Interface Schedule Event

### 3.5.3) View Client

This picture will show the interface of view client. Which is used in the project. It can show the list of added clients.

The screenshot shows a web application interface with a sidebar on the left and a main content area. The sidebar has a 'Home' button and a list of menu items: News, About Case, Bar Activity, Event, View, Staff, Cases, Clients, Pages, Update Credentials, and Register. The main content area displays a table with the following data:

Case	Client Name	Client email	Case type
1	Ali	Ali@gmail.com	Family
58	Wasi	Wasi@gmail.com	Civil

There is a 'REFRESH' button above the table. The URL in the browser is localhost:35398/admindash/lawfirmfyp/clientclick.aspx.

Figure 3.11 Interface View Case

### 3.5.4) Add Case

This picture will show the interface of add case. Which is used in the project. It can show fields of case number, parties name, judge name, court type, date of decision, case type, case status, case law, client email, client name, phone number and client address, opponent name and choose file.

The screenshot displays the 'Add Cases' form within a web application. The interface is organized into a sidebar on the left and a main content area on the right. The sidebar contains navigation links: Dashboard, Add, Cases, Staff, Schedule, Event, Appointment, Post, News, About Case, and Bar Activity. The main content area is titled 'Add Cases' and features the following fields:

- Case No: Text input field
- Judge Name: Text input field
- Court Type: Dropdown menu (currently set to 'Supreme Court')
- Parties Name: Text input field
- Date of Decision: Text input field with a date format hint 'mm/dd/yy'
- Case Type: Dropdown menu (currently set to 'Civil')
- Case Status: Dropdown menu (currently set to 'Open')
- Case Law: Text input field
- Client Name: Text input field
- Client Email: Text input field
- Client Phone: Text input field
- Client Address: Text input field
- Opponent name: Text input field
- Upload file: Section containing a 'Choose File' button and the text 'No file chosen'
- INSERT: A prominent green button for submitting the case

Figure 3.12 Interface Add Case

### 3.5.5) Post News about Bar activities

This picture will show the interface of posting news for bar activities. Which is used in the project. It contains fields of resolution, decision, and court. It can show the option of post for posting the news.

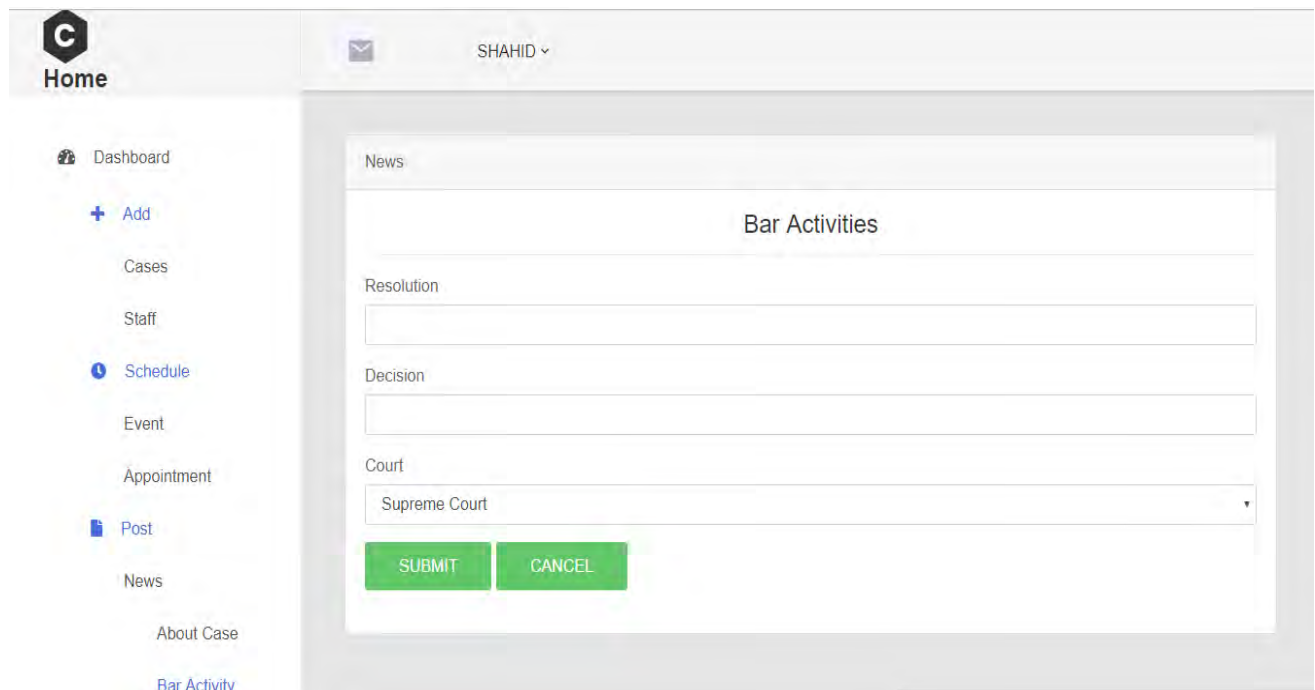


Figure 3.13 Interface Post News About Cases

### 3.6) Object and Actions (Sequence Diagram)

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

### 3.6.1) SD Login

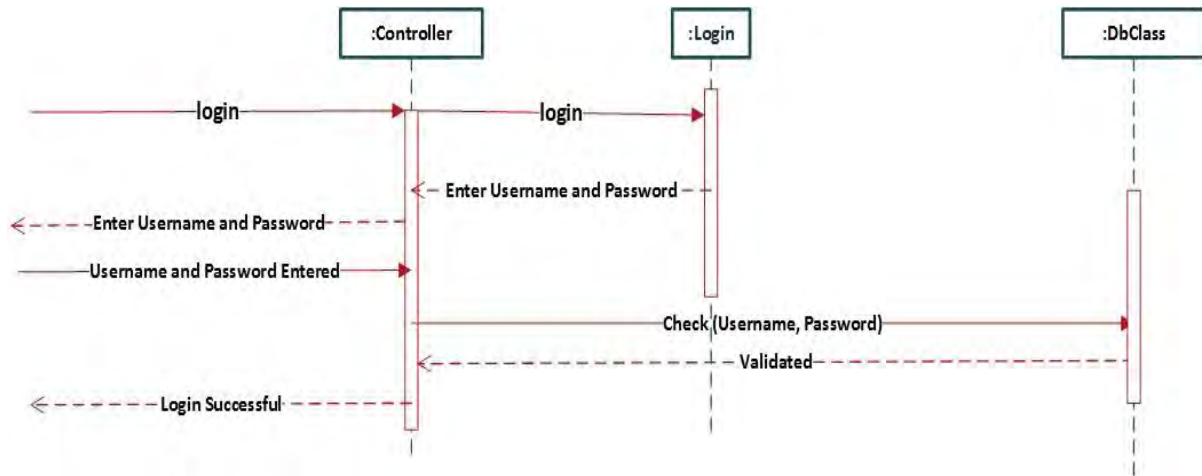


Figure 3.14 SD Login

### 3.6.2) SD Post News

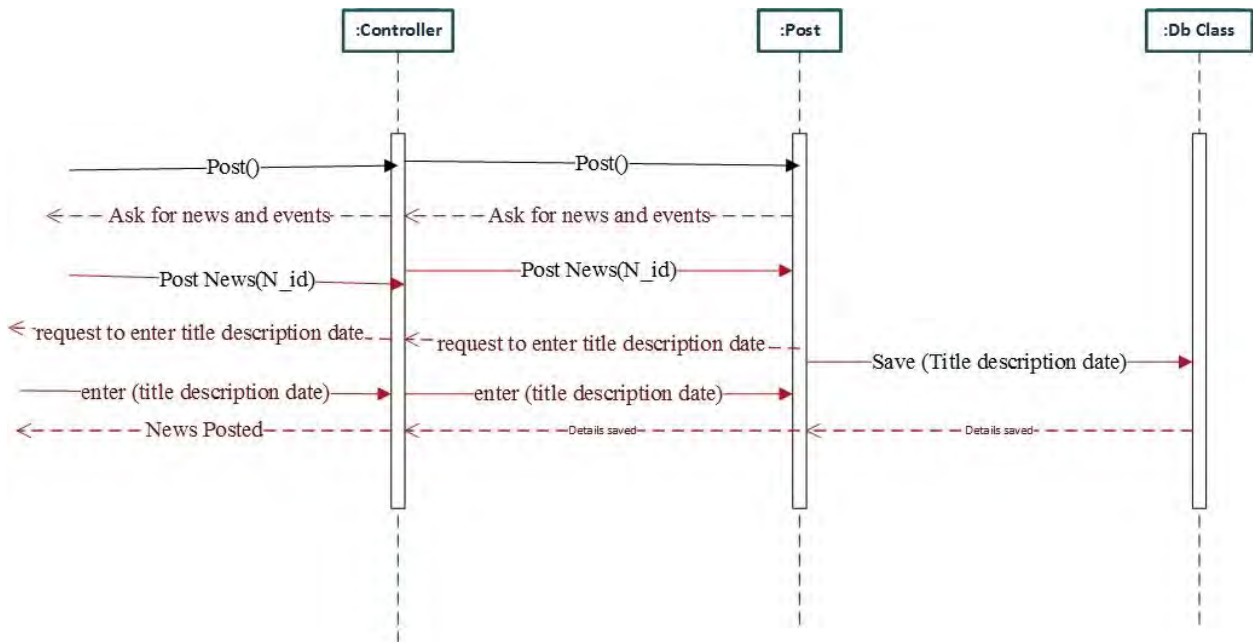


Figure 3.15 SD Post News

### 3.6.3) SD Add Case

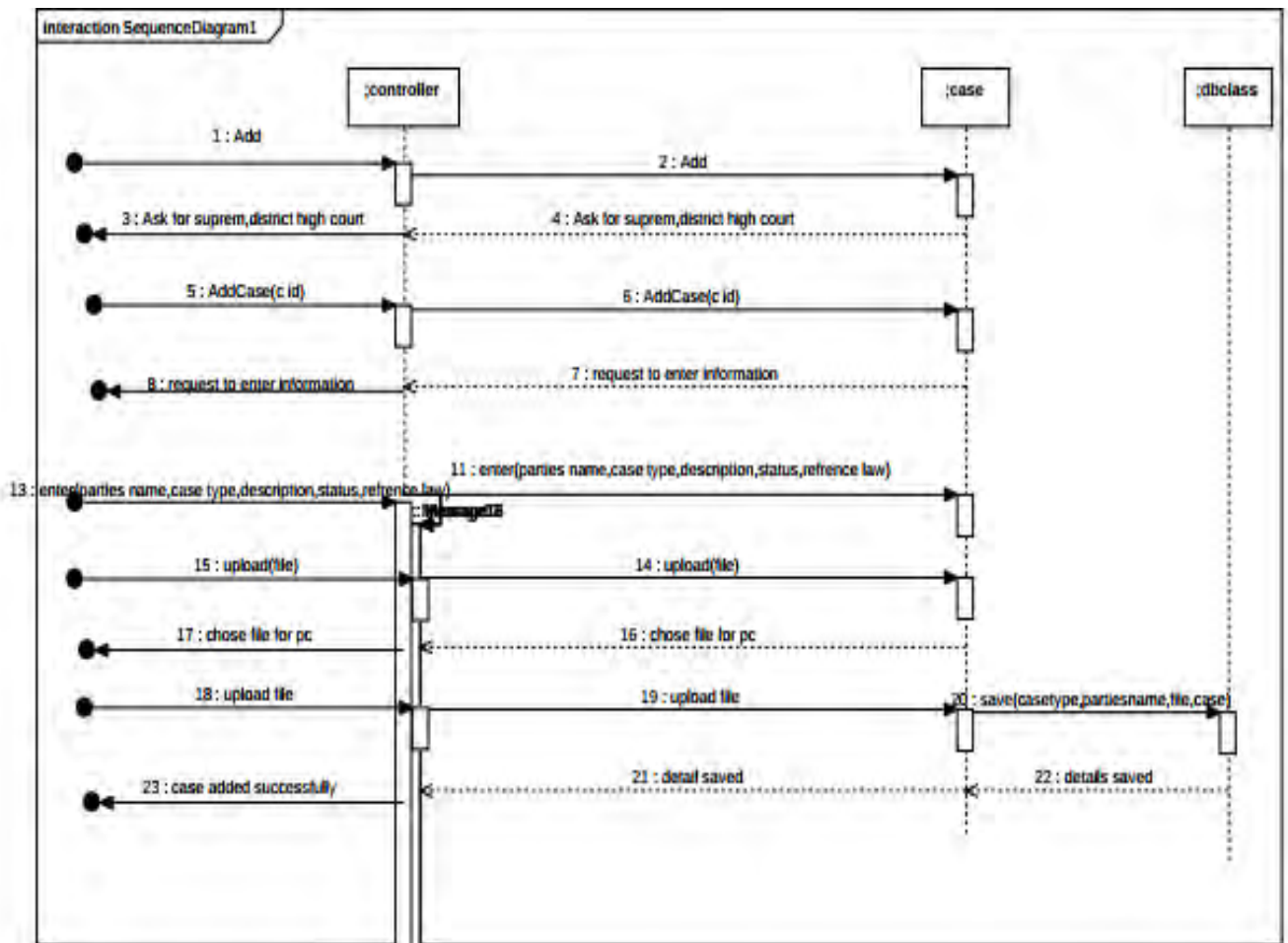


Figure 3.16 SD Add Case

### 3.6.4) SD Add Client

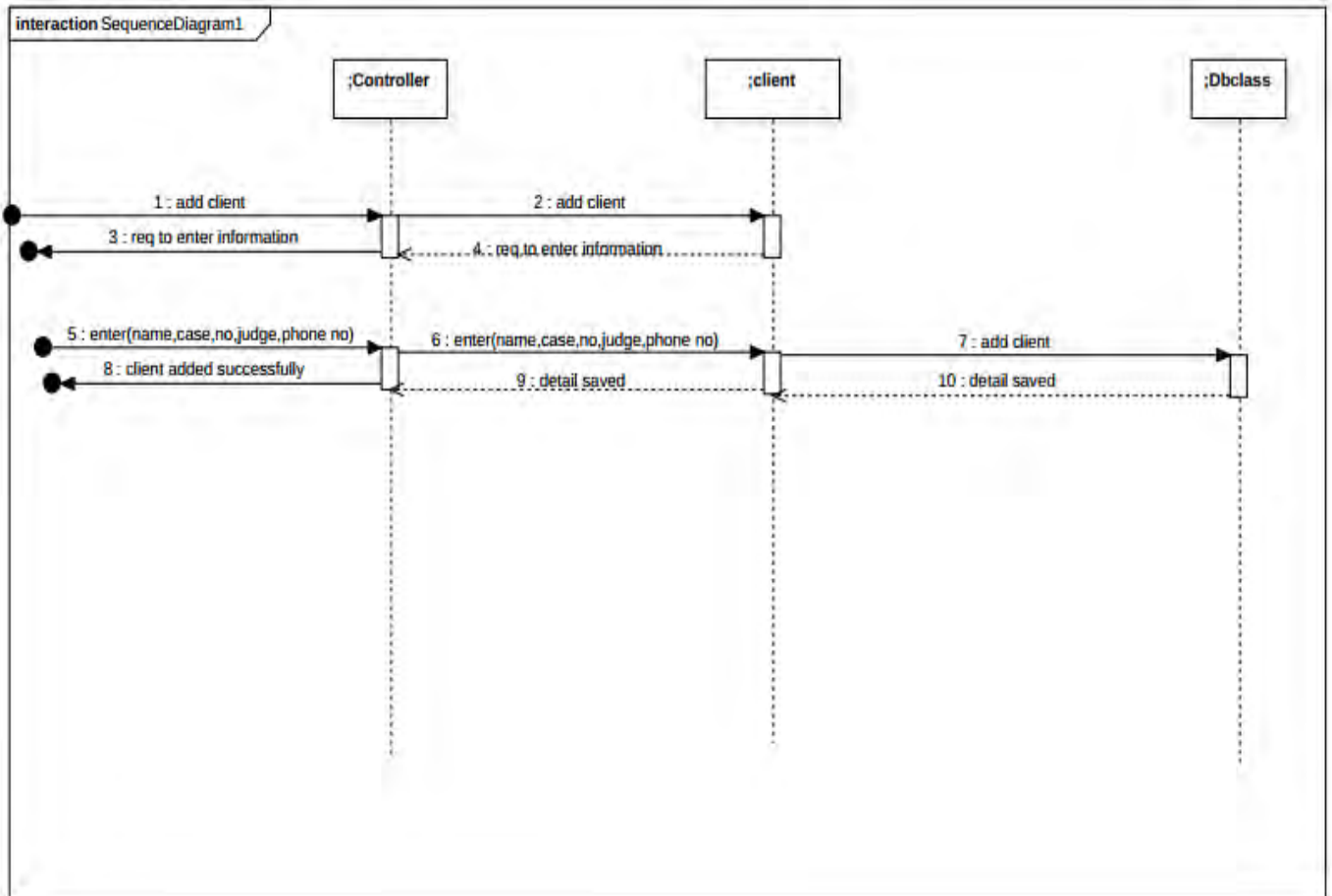


Figure 3.17 SD Add Client

### 3.6.5) SD Request for Appointment

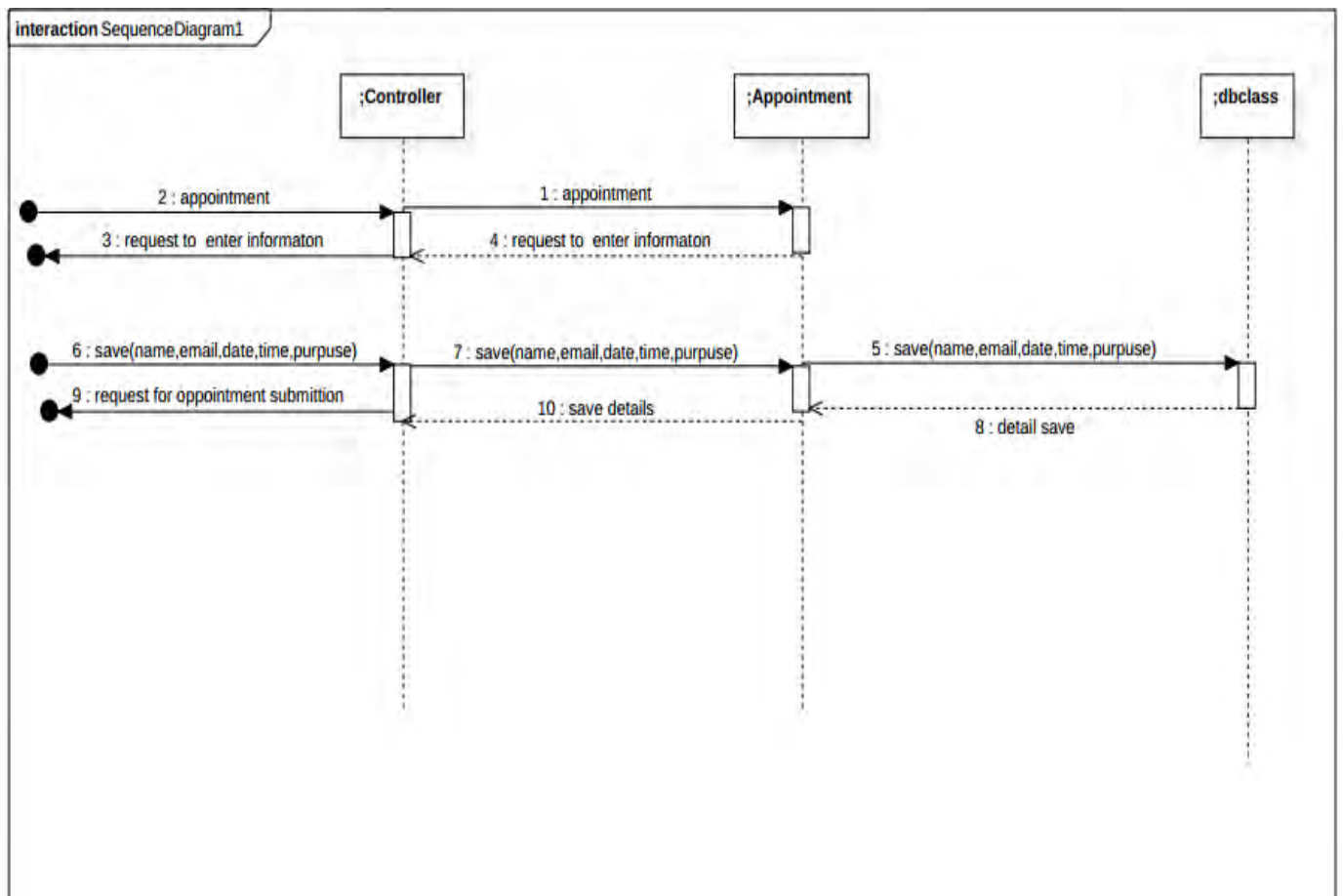


Figure 3.18 SD Request for Appointment



### 3.6.6) SD View Case

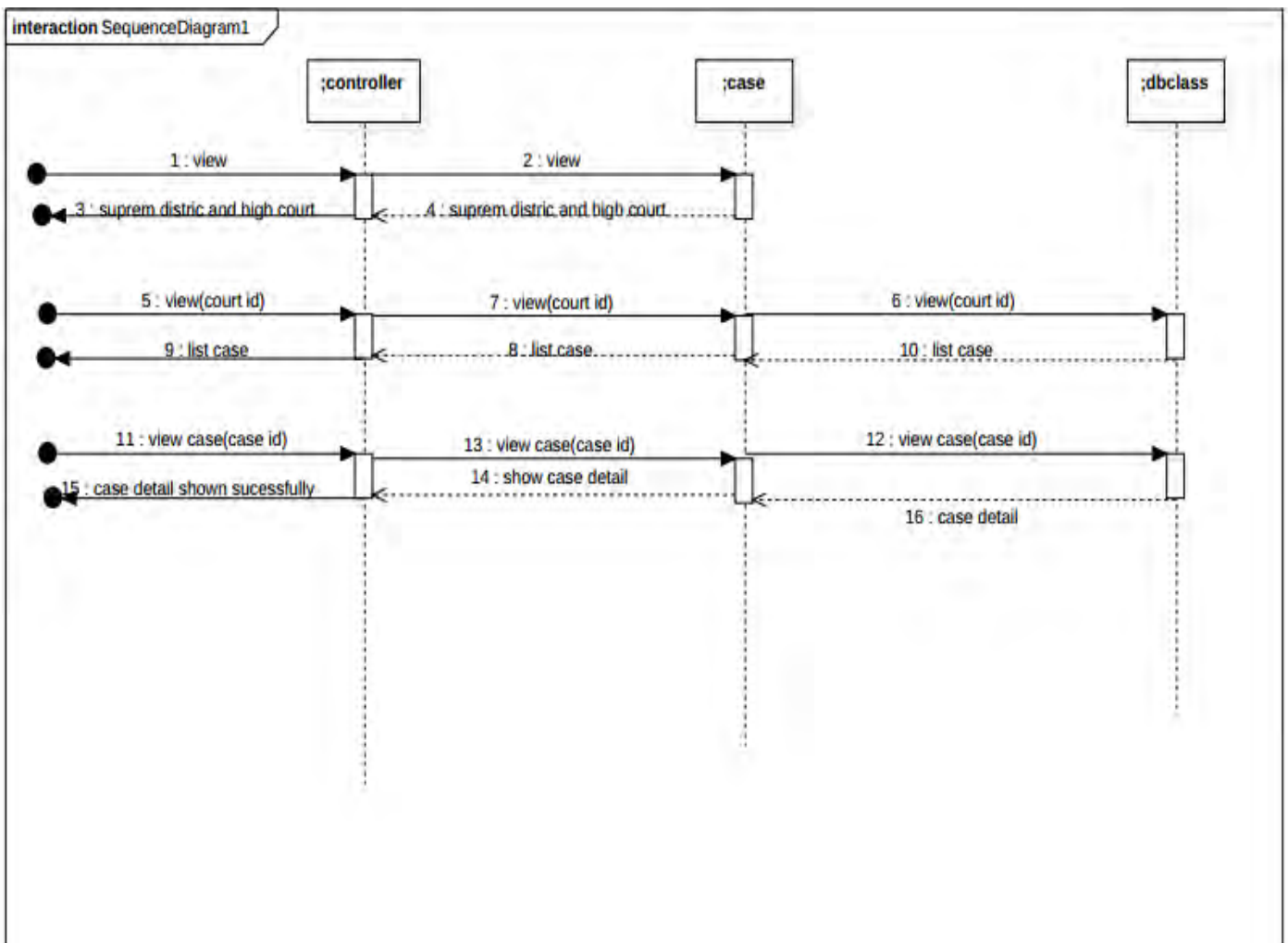


Figure 3.18 SD View Case

### 3.6.7) SD Schedule Reminder for Appointment

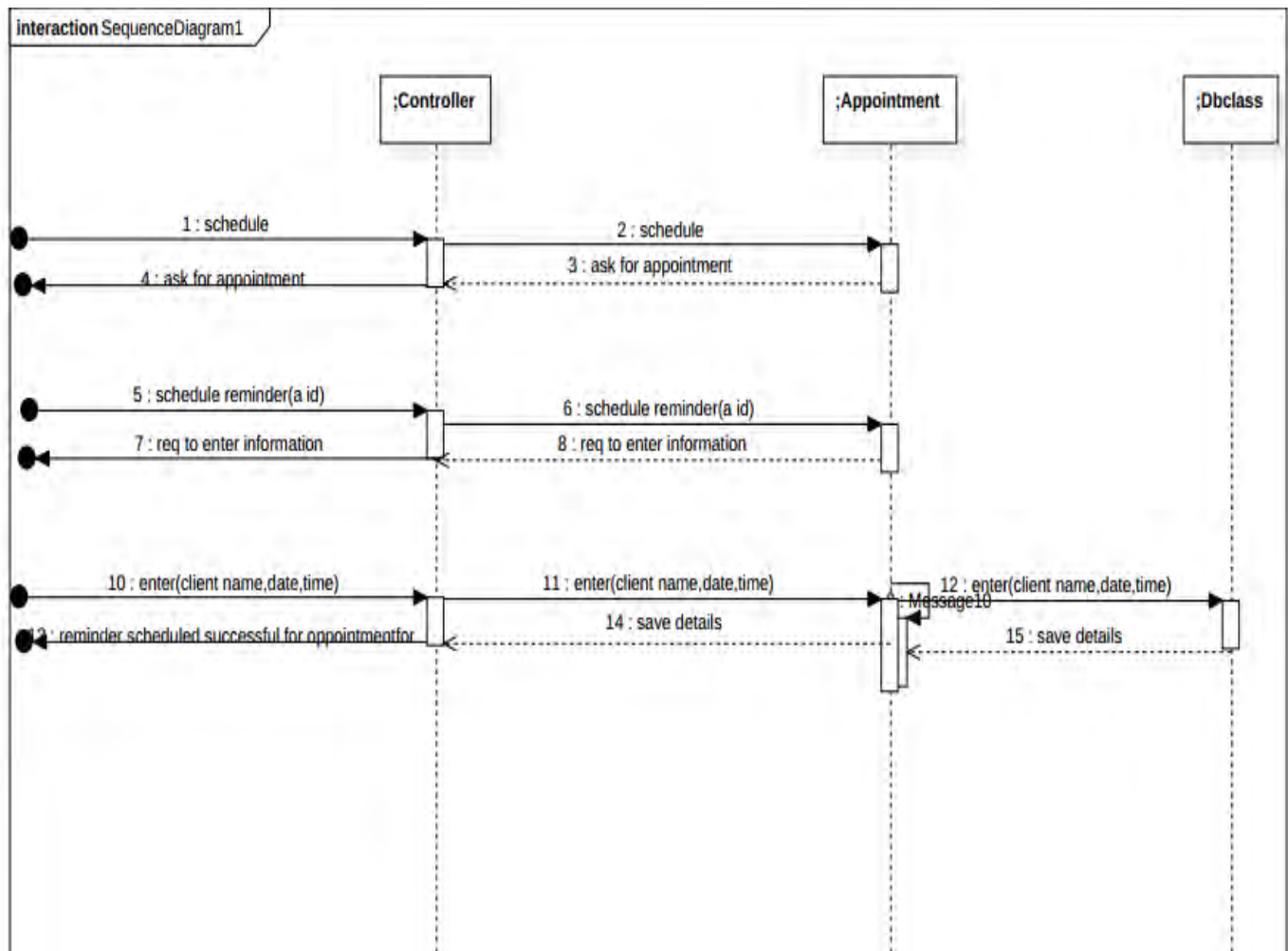


Figure 3.19 SD Schedule Reminder for Appointment

### 3.7) Class Diagram

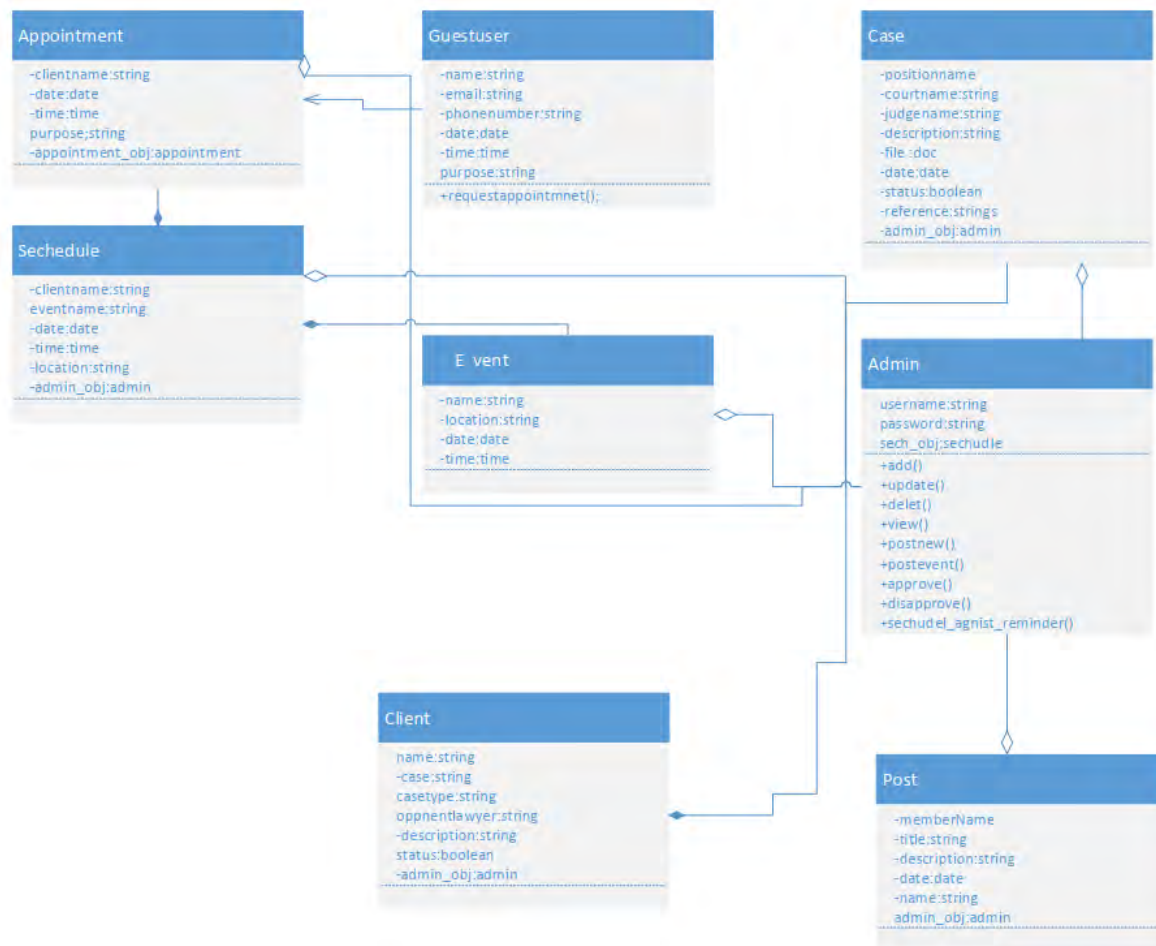
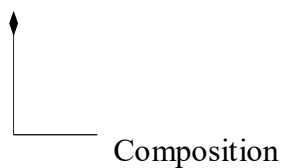


Figure 3.20 Class Diagram

Notations used in diagram: -



# Chapter 4

## Implementation

After the design phase, the implementation phase comes. This chapter is related to system implementation. The chapter mentions the tools, framework, platforms and database used to develop the application. In this phase we decide how to implement our design and which techniques to use. At last some interfaces are provided to visualize the application.

### 4.1) System Definition

System will mainly be composed of two parts

- Database Server
- A Web Application

#### 4.1.1) Database Server

System requires a Database Server which is used to store all data about cases, clients, staff, reminders, appointments and other required data for the case management system.

#### 4.1.2) Web Application

System consists of a web application which will be used as UI (User Interface). Admin will be able to access data from server using this web application. User can request for an appointment. Admin can approve user's appointments and can make them clients.

### 4.2) Development Tools

#### 4.2.1) Framework

The system is developed using visual Studio Integrated Development Environment (IDE).

##### Visual studio

Visual Studio is the official Integrated Development Environment (IDE) for web platform developing.

## 4.2.2) Language Selection

Building a web app comes down to two major languages: C# and Asp.net. C# is the language used in web, learning Asp.net for the design of the app.

## C Sharp

C# was designed to have the look and feel of the C++ language, but it is simpler to use than C++ and enforces an object-oriented programming model. C# can be used to create complete applications that may run on a single computer or be distributed among servers and clients in a network. It can also be used to build a small application module. The C# programming language requires the presence of a software platform in order for compiled programs to be executed.

## 4.3) Why C#?

Following are the main features

### 4.3.1) Platform Independent

C# Language is Platform Independent means program of C# is Easily transferable because after Compilation of C# program bytes' code will be created then we have to just transfer the Code of Byte Code to another Computer This is not necessary for computers having same Operating System in which the code of the C# is Created and Executed After Compilation of the C# Program We easily Convert the Program of the C# the another Computer for Execution.

### 4.3.2) Object Oriented

We Know that is purely OOP language that is all the code of the C# language is written into the classes and objects so for This feature java is most popular language because it also supports code reusability, maintainability etc.

### 4.3.3) Robust and Secure

The Code of C# is Robust and first checks the reliability of the code before execution when we trying to convert the higher data type into the lower. Then it checks the demotion of the code the It will warn a user to not to do this, so it is called as Robust.

### 4.3.4) Distributed

C# is Distributed Language Means because the program of C# is compiled onto one machine can be easily transferred to machine and Executes them on another machine because facility of Bytes Codes So C# is Specially designed for Internet Users which uses the Remote Computers for Executing their Programs on local machine after transferring the Programs from Remote Computers or either from the internet.

## 4.4) Code Snippets

```
namespace LAWFIRM.admindash.lawfirmfyp
{
    1reference
    public partial class login : System.Web.UI.Page
    {
        SqlCommand cmd = new SqlCommand();
        SqlConnection con = new SqlConnection();
        SqlDataAdapter sda = new SqlDataAdapter();
        DataSet ds = new DataSet();

        0references
        protected void Page_Load(object sender, EventArgs e)
        {
            string conn = ConfigurationManager.ConnectionStrings["lf_con"].ConnectionString;

            con.ConnectionString = conn;
            con.Open();
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            cmd.CommandText = "select* from login where username='" + TextBox1.Text + "'and password='" + TextBox2.Text + "'";
            cmd.Connection = con;
            sda.SelectCommand = cmd;
            sda.Fill(ds, "Login");
            if (ds.Tables[0].Rows.Count > 0)
            {
                Label2.Text = "data is found";
                Session["USER_ID"] = TextBox1.Text;
                Response.Redirect("~/admindash/lawfirmfyp/dashboard2.aspx");
            }
            else
            {
                Label2.Text = "data not found";
            }
        }
    }
}
```

Figure 4.1 Code Snap for getting login into system

# Chapter 5

## System Testing

This chapter illustrates the test approach which I used in this project, testing tools and environment, and the test cases.

### 5.1) Test Approach

A test approach is the test strategy implementation of a project and defines how testing would be carried out. Test approach has two techniques: -

**Proactive:** - An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.

**Reactive:** - An approach in which the testing is not started until after design and coding are completed.

### 5.2) Test Plan

#### 5.1.1) Testing Tools and Environment

The testing environment which I use is black box testing in which the internal structure design implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. This method is named so because the software program, in the eyes of the tester, is like a black box inside which one cannot see. This method attempts to find errors in the following categories incorrect or missing functions interface errors.

### 5.3) Test Cases

System consists of a web application which will be used as UI (User Interface). Admin will be able to access data from server using this web application. User can request for an appointment. Admin can approve user's appointments and can make them clients.

### 5.3.1) Login

This test case will describe the success scenario of user login. The user enters the username and password. The system matches the username and password from the database. If it is match the user login successfully.

ID	T001
Description	Admin can login into the system.
Tester	Admin
Setup	Register admin with id <a href="mailto:shahid88@gmail.com">shahid88@gmail.com</a> and password 6995950.
Instructions:-	<ol style="list-style-type: none"><li>1. Enter Id <a href="mailto:shahid88@gmail.com">shahid88@gmail.com</a></li><li>2. Enter password 6995960</li><li>3. Press login button</li></ol>
Expected Result	Admin with id <a href="mailto:shahid88@gmail.com">shahid88@gmail.com</a> should be logged into the system.
Oracle	<b>Pass</b>

### 5.3.2) Login (Alternative Scenario)

This test case will describe the alternative scenario of user login. The user enters the username and password. The system matches the username and password from the database. If it is not match the user login cannot login into the system.

ID	T002
Description	Admin cannot logged into the system.
Tester	Admin
Setup	Register admin with id <a href="mailto:shahid88@gmail.com">shahid88@gmail.com</a> and password 6995960.
Instructions	<ol style="list-style-type: none"><li>1. Enter id <a href="mailto:shahid88@gmail.com">shahid88@gmail.com</a></li><li>2. Enter password 6995960.</li><li>3. Press login button</li></ol>
Expected Result	<ol style="list-style-type: none"><li>1. Admin with id <a href="mailto:shahid88@gmail.com">shahid88@gmail.com</a> should not be logged into the system.</li><li>2. There is an error in email or password.</li></ol>
Oracle	<b>Fail</b>



### 5.3.3) Post News

This use case tells us the success scenario how admin can post news. The system shows the form to admin. If admin successfully fill the form, then admin successfully post the news.

ID	T003
Description	News will be posted for guest users.
Tester	Student, Teacher, Faculty.
Setup	Login as Admin
Instructions:-	<ol style="list-style-type: none"><li>1. Login as admin with id <a href="mailto:shahid88@gmail.com">shahid88@gmail.com</a></li><li>2. Enter the title of news.</li><li>3. Then enter the description for news.</li><li>4. Enter that date when you are going to post news.</li><li>5. Press post button.</li></ol>
Expected Results	News will be posted.
Oracle	<b>Pass</b>

### 5.3.4) Post News (Alternative Scenario)

This use case tells us the alternative scenario how admin can post news. The system shows the form to admin. If admin miss any field while filling form, then admin cannot post the news.

ID	T004
Description	News will not be posted for guest users.
Tester	Student, Teacher, Faculty
Setup	Login as admin.
Instructions	<ol style="list-style-type: none"><li>1. Login as admin with id <a href="mailto:shahid88@gmail.com">shahid88@gmail.com</a>.</li><li>2. Enter the title of news.</li><li>3. Enter the description for news.</li><li>4. Enter the date, you are posting news.</li><li>5. Press post button.</li></ol>
Expected Result	News will not be posted. <ol style="list-style-type: none"><li>1. There is an internet problem.</li><li>2. Required fields are not filled properly.</li></ol>
Oracle	<b>Fail</b>

### 5.3.5) View Appointments

This use case tells admin that how admin can see the appointments successfully. If the admin selects carefully appointment type, then admin can successfully view the appointments.

ID	T005
Description	Appointments list will be shown to admin.
Tester	Student, Teacher, Faculty
Setup	Login as admin.
Instructions	<ol style="list-style-type: none"><li>1. Login as admin with id <a href="mailto:shahid88@gmail.com">shahid88@gmail.com</a></li><li>2. Selects appointments option.</li><li>3. Choose approved or requests option from dropdown.</li><li>4. Appointments will be shown to admin according to his chosen option.</li></ol>
Expected Results	List of appointments will be shown to admin.
Oracle	<b>Pass</b>

### 5.3.6) View Appointments (Alternative Scenario)

This use case tells admin that how admin can see the appointments successfully. If the admin hasn't select appointment type, then admin can successfully view the appointments.

ID	T006
Description	Admin cannot view appointments.
Tester	Student, Teacher, Faculty
Setup	Login as admin.
Instructions	<ol style="list-style-type: none"><li>1. Login as admin with id <a href="mailto:shahid88@gmail.com">shahid88@gmail.com</a></li><li>2. Selects appointments option.</li><li>3. Choose approved or requests option from dropdown.</li><li>4. Appointments will be shown to admin according to his selection.</li></ol>
Expected Result	List of appointments will not be shown. <ol style="list-style-type: none"><li>1. There will be an internet error.</li><li>2. Admin haven't chose option from dropdown.</li></ol>
Oracle	<b>Fail</b>

### 5.3.7) Add Case

This use case describes the success scenario of how admin can add case into system. Admin selects case category then selects desired type and then form appears to admin. Admin fill the form and press add button and case will be added into the system.

ID	T007
Description	Admin successfully adds case into system,
Tester	Student, Teacher, Faculty
Setup	Login as Admin
Instructions:-	<ol style="list-style-type: none"><li>1. Login as admin</li><li>2. Admin selects case option.</li><li>3. Then admin choose the desired type of court for which he wants to add case.</li><li>4. Admin fills the form and upload file.</li><li>5. Press add button.</li></ol>
Expected Result	Case will be added into the system successfully.
Oracle	<b>Pass</b>

### 5.3.8) Add Case (Alternative Scenario)

This use case describes the alternative scenario of how admin can add case into system. Admin selects case category then selects desired type and then form appears to admin. Admin left some field blank in the form and press add button and case will not be added into the system.

ID	T008
Description	Case will not be added into the system.
Tester	Student, Teacher, Faculty
Setup	Login as admin
Instructions:-	<ol style="list-style-type: none"><li>1. Login as admin with id <a href="mailto:shahid88@gmail.com">shahid88@gmail.com</a></li><li>2. Admin selects case option.</li><li>3. Then admin choose the desired type of court for which he wants to add case.</li><li>4. Admin fills the form incorrectly and use redundant information.</li><li>5. Press add button.</li></ol>
Expected Result	Case will not be added into system.

Oracle	<b>Fail</b>
--------	-------------

### 5.3.9) Add Client's data

This use case describes the success scenario of how admin adds client data into the system. Admin selects client option. System shows the form, admin fills the forms and successfully saves client's data into the system.

ID	T009
Description	Admin successfully adds client's data into the system
Tester	Student, Teacher, Faculty
Setup	Login as admin
Instructions:-	<ol style="list-style-type: none"> <li>1. Login as admin with id <a href="mailto:shahid88@gmail.com">shahid88@gmail.com</a></li> <li>2. Admin selects client option.</li> <li>3. Admin fills the form correctly.</li> <li>4. Press add button.</li> </ol>
Expected Results	Client's data will be added successfully
Oracle	<b>Pass</b>

### 5.3.10) Add Client's data (Alternative Scenario)

This use case describes the success scenario of how admin adds client data into the system. Admin selects client option. System shows the form, admin left some field blank and cannot saves client's data into the system.

ID	T010
Description	Admin cannot add client's data into the system.
Tester	Student, Teacher, Faculty
Setup	Login as admin
Instructions:-	<ol style="list-style-type: none"> <li>1. Login as admin with id <a href="mailto:shahid88@gmail.com">shahid88@gmail.com</a></li> <li>2. Admin selects client option.</li> <li>3. Admin fills the form and left some fields blank.</li> <li>4. Press add button</li> </ol>
Expected Result	Client's data will not be added into the system.
Oracle`	<b>Fail</b>

# Chapter 6

## Conclusions and future enhancements

The main objective of this application is to facilitate the lawyer with different services like it can save and manages cases, client's data, staff data. By this, it gets easier for lawyer to list cases under different courts. Lawyer posts news and events for clients. Lawyer schedules reminder against appointments and events for avoiding future inconvenience. Cases and client's information can be accessed on a single click, because of extensive search engine provided to lawyer about cases and clients.

Client can get appointments with lawyer easily via internet instead of personally visiting the lawyer. Client can easily locate the lawyer office though google map by getting directions. Client can view news and events which lawyer had shared for clients.

### Future Enhancements

It is possible to enhance the application in future. Here are some features that can be added to application:

- Clients will be able to get register on site and meanwhile can check the status of their ongoing case by means of progress report shared by lawyer.
- After multiple lawyers' registration into system, a lawyer can allocate his case to other lawyer and can also share case relevant information with him.
- By client registration, lawyer and client can easily communicate instead of physical meeting.

## References

- [1] Chapter 2. Software Requirement Specifications Software Engineering A lifecycle approach by P. Mohapatra
- [2] Chapter 6. Use cases and Functional Requirement Applying UML and Patterns by Craig Larman
- [3] Chapter 13. Design Concepts and Principles, Software Engineering A Practitioner's Approach by R.S. Pressman
- [4] Chapter 14.5 Class Diagrams Software engineering A lifecycle approach by P. Mohapatra  
Software Engineering A Practitioner's Approach by R.S. Pressman,
- [5] Chapter 8. Software Testing Ian\_Sommerville \_Software\_Engineering\_9<sup>th</sup>\_edition