

**PROJECT REPORT ON
ADVOCATE CASE DIARY**

**SUBMITTED BY
TAGADGHAR MANOJKUMAR MOTILAL
Exam Seat No: 5801**



**UNIVERSITY OF MUMBAI
DEPARTMENT OF INFORMATION TECHNOLOGY**

PROJECT GUIDE

Prof. Ansari Wahid

PADHMSHREE ANNASAHEB JADHAV BHARTIYA SAMAJ UNNATI SIKSHAN MANDAL's

BNN COLLEGE, BHIWANDI

(Arts, Commerce & Science)

BHIWANDI, District: THANE-421 305

2013-2014



PADHMSHREE ANNASAHEB JADHAV BHARTIYA SAMAJ UNNATI SIKSHAN MANDAL's
BNN COLLEGE, BHIWANDI
(Arts, Commerce & Science)
BHIWANDI, District: THANE-421 305

CERTIFICATE

This is to certify that **Tagadghar Manojkumar Motilal [S.No: 5801]** student of Third Year Bachelor of Science (Information Technology) of BNN College has successfully completed the Project work titled "**Advocate Case Diary**" in partial fulfillment in this educational year of University of Mumbai.

This Project is the record of authentic work carried out during the academic year 2013-14. It is further certified that he has completed all required phases of the Project.

Date: ___/___/2014

Place: Bhiwandi

Project Guide
(Prof. Ansari Wahid)

H.O.D

Principal

Internal Examiner

External Examiner

Contents

Synopsis	8-12
1. Introduction	13-19
1.1 Objective and Scope of the Project	13
1.2 Theoretical Background	14
1.3 Definition of Problem	19
2. System Analysis and Design User Requirements	20-37
2.1 System Analysis and Design User Requirements	20
2.1.1 Analysis	20
2.1.2 Design	20
2.2 About Advocate Case Diary	21
2.3 Software Requirements	22
2.4 System Planning (Gantt chart)	23
2.5 Detailed Life Cycle of the Project	24
2.5.1 Entity Relationship Diagram	25
2.5.2 Data Flow Diagram	28
2.5.3 Use Case Diagram	32
2.5.4 Class Diagram	35
2.5.5 Collaboration Diagram	36
2.5.6 Activity Diagram	37
3. Methodology Adopted	38-43
3.1 Model used in the system	39
3.2 System Implementation	41
3.3 Details of Hardware and Software	41
3.3.1 Hardware Requirements	41
3.3.2 Software Requirements	41
4. User/Operational Manual	44-47
4.1 User/Operational Manual	45
4.2 Step to install SQL Server 2005	45
4.3 Step to install Visual Studio 2010	46
4.4 Installation steps for project	47
4.5 Execution steps for project	47
5. GUI Interface	48-79
5.1 GUI Interface	48
6. Methodology used for testing	80-82
6.1 Methodology used for testing	81
6.1.1 Testing	81
6.2. Testing Objective	81
6.3 Strategies used for software testing	81
6.3.1 Login form validation	81
6.3.2 Incomplete details validation	81

7. Cost and Benefit Analysis	83-84
7.1 Cost and benefit analysis	84
7.1.1 Need of the cost benefit analysis	84
7.1.2 Cost Estimation	84
7.1.3 Benefit Analysis	84
8. System Maintenance	85-87
8.1 System Maintenance	86
8.1.1. Backup	86
8.1.2. Malware Management	86
8.1.3. File System Maintenance	87
8.2 Limitations	87
9. Annexure	88-94
9.1 Brief Background of Database System	89
9.2 Data Dictionary	90
9.3 Table List	92
10. Conclusion and Bibliography	95-98
Conclusion	96
10.1 List of abbreviations, Figures	96
10.1.1. Abbreviation	96
10.1.2. Figures	98
10.2 Reference	98
10.2.1. Bibliography	98
10.2.2. Website Used	98

ACKNOWLEDGEMENT

This Project, being a maiden experience for us, has been a big confidence booster. Contributions of the following individuals / persons are noteworthy. It gives me immense pleasure to acknowledge the innumerable guidance extended to me by Advocate Mr. Patil . I extend my sincere thanks to him who helped me in getting the details of the process in court management. I would like to thank my classmates for their help in ASP.NET.

In this endeavor I acknowledge the valuable contribution and expert guidance of our internal project guide **Prof. Ansari Wahid Sir** of Department of Information Technology, B.N.N. college of Arts, Science and Commerce, Bhiwandi, Thane-421 305. His frank suggestions for improvements & innovative ideas have inspired us.

I am greatly thankful to our Principal. U.D.Kadam Sir for his continued and unending support. I also thank Prof. Dr. Bhamare Sir Head of Information Technology Department for granting me permission to work on this project.

We wish to express our gratitude to the staff of I.T. department for their co-operative & considerate approach. Last but not least, our sincere thanks to College who gives us a great opportunity to grape the knowledge.

Thanking you all.

Yours sincerely

Tagadghar Manojkumar Motilal
TYBSc(IT)-5801

DECLARATION

I hereby declare that the project work entitled “Advocate Case Diary” submitted to the Mumbai University is a record of an original work done by me under the guidance of Prof. Wahid Sir and this project work has not performed on the basis for the award of any degree or diploma / associate ship/fellowship and similar project if any.

Signature

Tagadghar Manojkumar Motilal

Synopsis

1. Project Title: Advocate Case Diary

❖ Introduction:

This Software is very useful for Managing the daily cases diary for Lawyers/ Advocates. This is of software, which store information about an advocate's clients, opponents, cases no., Previous Date, Current Date, Next Date, court name. You can filter the data according to case no, Parties, next date etc. You can also maintain the files stored on the locations of your hard disk for individual client. The software is very easy to operate and light on system tool to maintain information about Clients, Cases, Hearings, Rulings, Accounts, books, periodicals etc.

Software manages day-to-day activity and builds a comprehensive client/case/matter database and history that can help improve every aspect of your practice. You will have immediate access to tools for case status and information, document management and assembly, calendaring and docketing and contact management. The system also comes with a number of standard reports that can be accessed easily by users. Track time, manage receivables etc. For quick reference provision of many reports is also made. It also provides information about daily schedule tasks. Software is dedicated to advocates/ lawyers to help them in maintaining their offices.

2. Objectives:

This software is very easy to use. The main goal is reducing paper work. It helps us to recollect the information about the previous and future case information.

3. Features of the Software:

Information need only be entered once and is available wherever and whenever you need it.

More importantly, it all works together in the way you would expect, so there's a natural workflow to everything you do.

- Daily reports, overviews and practice reminders

- Create separate files on each client matter
- A Notepad pad for centralized note taking
- Provide information about daily schedule tasks.
- Maintain their Diary of Cases & Clients

Case Entry –

Information need only be entered once. You have to just fill up required information of your Client/Case, Further you have to only make entries of Next Date and Stage and remaining will be take care by this software. When case is decided, you can select option of decide cases so that further it will be not shown in your Diary.

Personal Information –

You have to make entries relating to your personal information can also add your photograph.

Diary –

You can see cases of an Specific date or between any dates and many more information as detailed below. Next date of Hearing and Case Stage can be entered directly from Diary.

Next Hearing Date Entry –

This entry can be made directly into Diary or by using Special Feature which shows all cases where Next Hearing date not entered.

Cases without next date of Hearing –

A special feature in diary is which shows you all cases where no next date of hearing entered. You can directly enter next date and case stage.

Case Stage Entry –

Facility to know progress stage of case, this entry made while entering Hearing Date in Diary.

Individual Case Details –

Facility to see All Hearing Dates and other details of an individual case provided.

All cases of a Client –

Facility to see details of All Cases of a Client with hearing dates provided.

Court Wise Cases –

Diary will show you list of cases for a Specific Court.

Active Cases –

You have option in case details to make a Case Active or Decided. New case is automatically selected option of Active Case.

Decided Cases –

When you select this option for a case in case details, than that Case is not shown in Diary. You can any time deselect this option and Case can become Active.

4. Facilities:

- 1. Printouts:** You can take Printouts of any information provided by this Software
- 2. PDF File:** Any information provided by this software can be converted to PDF File. This file can be than given and Viewed by any person who did not own this software.
- 3. Email:** PDF File can be easily Email to any person.
- 4. Sorting:** Any Information provided by Diary can be Sorted (A to Z or Z to A)
- 5. Backup:** Facility to take backup of data entered by you.
- 6. Recovery:** Facility to Recover backup data is also provided.

5. Hardware Requirement:

- 4 GB of Hard Disk
- 1 GB RAM(minimum)

6. Software Requirement:

- Operating System:- windows XP(onwards)
- Front End:- Microsoft Visual Studio 2010
- Back End:- Microsoft SQL server 2005

Resources

The resources used for designing this project are Asp.Net and SQL.

- i. **Visual C# .NET:** An object-oriented programming language from Microsoft. It is the .NET version of the Visual c# programming language. Like all .NET languages, ASP.NET uses the Common Language Runtime (CLR) for program execution. ASP.NET is substantially different from traditional Visual C#, which has been the most popular language for developing Windows applications.
- ii. **SQL:** Computer programming language used for retrieving records or parts of records in databases and performing various calculations before displaying the results. SQL is particularly suitable for searching relational databases. It has a formal, powerful syntax and is able to accommodate logical operators. Its sentence-like structure resembles natural language except that its syntax is limited and fixed.

7. Proposed architecture

- i. The system should be developed in such a way that it handles the needs of customers.
- ii. It should be developed in such a way that new users can also handle operations of the system.
- iii. It should have user friendly interface & very easy structure, so that it can be understood by the end users.
- iv. It should increase the efficiency of the system with reduced cost. Also the system should take very few inputs from the users.
- v. To accomplish above features we will use the "**client-server architecture**".

8. Testing technology to be used

The main objective of system testing is to find the errors or bugs in the system and to see whether the system fulfills expectations of the user. Three types of testing were carried out.

i. Unit testing

After coding of each form or module over it was tested with test data, output was compared with the expected one.

ii. Integration testing

As the complete system was ready in pieces scattered all over they were integrated together and vigorous integration testing is carried out.

iii. System testing

System Testing was carried out by one of the person who manages a threat where he confirms everything is cleared and is working fine.

9. Contribution made by project

- i. Whenever the administrator wants' detail information i.e. how many customer buy the products, how many products are sold, what is the availability of the stock etc. Then he doesn't have to depend on the manager or any employee of that Advocate Store. Immediately he can get all information.
- ii. Whenever stock of the product is reduce to particular level, automatically administrator get this information and he gives order to supplier.

CHAPTER 1

Introduction

1.1 Objectives and Scope of the Project

This Software is very useful for Managing the daily cases diary for Lawyers/ Advocates. This is of software, which store information about an advocate's clients, opponents, cases no., Previous Date, Current Date, Next Date, court name. You can filter the data according to case no, Parties, next date etc. You can also maintain the files stored on the locations of your hard disk for individual client. The software is very easy to operate and light on system tool to maintain information about Clients, Cases, Hearings, Rulings, Accounts, books, periodicals etc.

Software manages day-to-day activity and builds a comprehensive client/case/matter database and history that can help improve every aspect of your practice. You will have immediate access to tools for case status and information, document management and assembly, calendaring and docketing and contact management. The system also comes with a number of standard reports that can be accessed easily by users. Track time, manage receivables etc. For quick reference provision of many reports is also made. It also provides information about daily schedule tasks. Software is dedicated to advocates/ lawyers to help them in maintaining their offices.

Objectives

This software is very easy to use. The main goal is reducing paper work. It helps us to recollect the information about the previous and future case information.

Preliminary investigation

a) Existing System:-

- In the existing system the details of the client entered manually
- A case diary is used for maintained the details of client.
- All day to day work is maintained in different registers.
- There is no maintenance of the records of payment.
- It is difficult to explain the history of the cases.
- There is all paperwork.

b) Limitations of the present system:-

The present system is the manual system. And it is difficult to work with it. Following are the major drawbacks.

- Entry of new cases done through forms and are stored in files.
- The information regarding the case and clients is maintained in diary.
- There is no proper maintenance of case details and the payments.
- It is time consuming.
- It is not user friendly.

C) Advantage of Proposed System:-

a) TIME SAVING :-

Since all the client details and related information are stored in the system, it enables insignificant reduction in the number of registration taken up for the storage and retrieval of information.

b) ACCURACY IN RELIABILITY :-

Problems related with data loss and inconsistency can be eliminated, thus making the system more accurate and reliable.

c) INFORMATION MAINTENANCE :-

It can now be an easy task as it removes the large registration from the scene.

d) AVAILABILITY OF REPORTS:-

Reports can be generated manually.

1.2 Theoretical Background

.NET is both a business strategy from Microsoft and its collection of programming support for what are known as Web services, the ability to use the Web rather than your own computer for various services. Microsoft's goal is to provide individual and business users with a seamlessly interoperable and Web-enabled interface for applications and computing devices and to make computing activities increasingly Web browser-oriented. The .NET platform includes servers; building block services, such as Web-based data storage; and device software. It also includes Passport, Microsoft's fill-in-the-form-only-once identity verification service.

The .NET platform was designed to provide:

- The ability to make the entire range of computing devices work together and to have user information automatically updated and synchronized on all of them
- Increased interactive capability for Web sites, enabled by greater use of XML (Extensible Markup Language) rather than HTML
- A premium online subscription service, that will feature customized access and delivery of products and services to the user from a central starting point for the management of various applications, such as e-mail, for example, or software, such as Office .NET
- Centralized data storage, which will increase efficiency and ease of access to information, as well as synchronization of information among users and devices
- The ability to integrate various communications media, such as e-mail, faxes, and telephones
- For developers, the ability to create reusable modules, which should increase productivity and reduce the number of programming errors

According to Bill Gates, Microsoft expects that .NET will have as significant an effect on the computing world as the introduction of Windows. One concern being voiced is that although .NET's services will be accessible through any browser, they are likely to function more fully on products designed to work with .NET code.

The full release of .NET is expected to take several years to complete, with intermittent releases of products such as a personal security service and new versions of Windows and Office that implement the .NET strategy coming on the market separately. Visual Studio .NET is a development environment that is now available. Windows XP supports certain .NET capabilities.

1.3Definition of Problem

1. Our project is one of the best choices for all the traditional users of Advocate store.
2. The main objective of our project is to design & develop a user friendly system which helps not only to the customers of the Advocate Store but also to the employees, suppliers by making their job easier.
3. We know that, in all over India there are many Advocate Stores where work is done manually. To manually handle any system was very difficult task.
4. Being an IT student, we know the importance of the computerization. Because of computerization the work of any system becomes very easy.
5. Through this computerization we can help the people to increase the growth of Advocate Store system & also reduce their hectic schedule.

Limitations Of Existing System

1. **Time consumption:** As the records are to be manually maintained it consumes a lot of time.
2. **Paper Work:** Lot of paper work is involved as the records are maintained in the files and registers.
3. **Storage Requirements:** As files and registers are used the storage space requirement is increased.
4. **Less Reliable:** Use of papers for storing valuable data information is not at all reliable.
5. **Accuracy:** As the system is in manual there are lot many chances of human errors. These cause errors in calculating mechanism or maintaining product and supplier data in registers.
6. **Difficulty in keeping new records:** It is difficult for keeping all the new entries of the products, their available stock, new suppliers.

This system is developed with an aim to solve above -mentioned problems.

Scope of the project

- Keeping all records of the client and court cases along with the personal details.
- Maintaining day to day information about client and cases.
- Maintaining all sorts of records in database.
- Storing information about the client payment transaction.
- Storing information about the old cases and court hearings.
- Provide search facility for searching desired data.
- Provides reports of all transactions.

Feasibility study

The following feasibility is considered for the project in order to ensure that the project is variable and is does not have any major obstructions.

- Economic feasibility
- Technical feasibility
- Behavioral feasibility
- Problem analysis

1. Economical feasibility:

Economical-feasibility is an evaluation of development cost weighed against the benefit from the developed system.

The existing resources are efficient for developing the system. No extra software is required.

2. Technical feasibility:

Economical-feasibility is a study of function, performance and constraints that may affect the ability to achieve an acceptable system. Technical analysis is being with assessment of the technical viability of the proposed system. During technical analysis we have to evaluate the merits of the system concept while, at the same time collecting additional information about performance, reliability, maintainability and predictability.

3. Behavioral feasibility:-

People are interested resistant to change and computers are known to facilitate changes. Every department welcomes the idea of computerization, but the resistance will be from the operations who are involved in the manual system.

4. Problem analysis:-

The first of the two basic activities performed during the requirement phase is analyzing the problem. Problem analysis is done to obtain a clear understanding of the needs of the clients and the users and what exactly desired from the software.

CHAPTER 2

System Analysis

and

Design User

Requirement

2.1 System Analysis and Design User Requirements

2.1.1. Analysis

This stage should report on what the user (client) wants the project should do. The candidate should draw a list of general and specific objectives of the project. The goal of requirement analysis is to understand such different aspects as the requirements of the problem, its context, and how it fits in the client's organization. All the errors in the analysis phase should be removed before design.

In this project a detailed analysis of various plans and policies has been done. The system developed calculates the premium for five policies by taking the input from the user i.e. Age, sum assured and term of the policy and gets the tabular premium from the database and performs calculation and displays the result. User can enter information through textboxes and the system validates it and displays the output.

2.1.2. Design

It includes important things like as below:

- a) What the interface will look like – use of various controls and so on
- b) The files needed - their structure, organization and processing
- c) How the data will be validated
- d) The procedures needed

The goal of the design process is to find the best possible design. The most important criterion to judge a design is verifiability, reliability and maintainability.

In this project the interface used is user friendly and the front-end used is asp.net and backend used is SQL. Since SQL is inbuilt in ASP.NET 2005 there is no requirement of SQL separately. The two users of the project are the insurance agent and a guest user. The insurance agent has a different login where as there is no login for the guest user. All the modules are displayed to the insurance agent, only policy information and premium calculator is visible to the guest user. The project is designed with the intention of simplifying the work of the insurance agent.

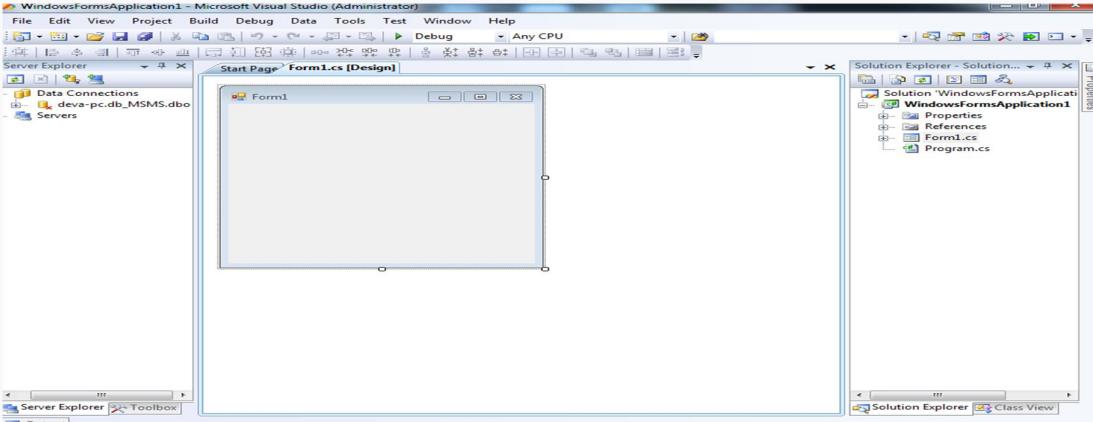
2.2 About Advocate Case Diary

In this Advocate Case Diary Client makes enquiry to the employee of the Case Details and appointments. Then the employee checks for the availability of the dates. If the dates are available then the meetings and court related information is gathered and fixed. This process is going on and lastly the bill is generated according to the payment that is needed to be done for the advocate. Also they can change the information of the client (if necessary).

Again there is an Administrator. He or she is also having some responsibility regarding to the Advocate Management System Software. His/ Her work is to calculate salary of the employee working under the Head Advocate and also be done with the job of attendance. And for security purpose they can change their password. The employees cannot access the information regarding to the administrator. This is done for the security purpose.

2.3 Software Requirement

- Visual Studio 2010 (Also contains inbuilt SQL)

Visual c#	
	
Paradigm	Structured, imperative, object-oriented, declarative
Appeared in	2002 (last revised 2013)

Designed by	Microsoft Corporation
Typing discipline	Dynamic, strong, both safe and unsafe nominative
Major implementations	.NET Framework, Mono
Dialects	Microsoft Visual Studio .NET, .NET 2003, 2005, 2008, 2010, 2014
Influenced	None

Figure (2.1): Microsoft Visual Studio Details.

- **SQL Server 2005**

The advent of the .NET Framework has brought with it some major changes, and database access is certainly no exception. Although ADO has been around since 1996, with the arrival of ASP, its latest incarnation present in the .NET Framework - ADO.NET - really does represent its coming-of-age. We'll be covering connecting to a SQL Server database, executing queries, calling stored procedures, filtering data, and reflecting changes in a database - and point you in the direction of further resources for more in-depth discussions on particular topics.

2.4 System Planning (Gantt chart)

A Gantt chart is a graphical representation of the duration of tasks against the progression of time. A Gantt chart is a useful tool for planning and scheduling projects.

A Gantt chart is helpful when monitoring a project's progress. A Gantt chart allows you to assess how long a project should take. A Gantt chart lays out the order in which tasks need to be carried out. A Gantt chart helps manage the dependencies between tasks. A Gantt chart allows you to see immediately what should have been achieved at a point in time. A Gantt chart allows you to see how remedial action may bring the project back on course.

A Gantt chart is constructed with a horizontal axis representing the total time span of the project, broken down into increments (for example, days, weeks, or months) and a vertical axis representing the tasks that

make up the project (for example, if the project is outfitting your computer with new software, the major tasks involved might be: conduct research, choose software, install software). Horizontal bars of varying lengths represent the sequences, timing, and time span for each task. Using the same example, you would put "conduct research" at the top of the vertical axis and draw a bar on the graph that represents the amount of time you expect to spend on the research, and then enter the other tasks below the first one and representative bars at the points in time when you expect to undertake them. The bar spans may overlap, as, for example, you may conduct research and choose software during the same time span. As the project progresses, secondary bars, arrowheads, or darkened bars may be added to indicate completed tasks, or the portions of tasks that have been completed. A vertical line is used to represent the report date.

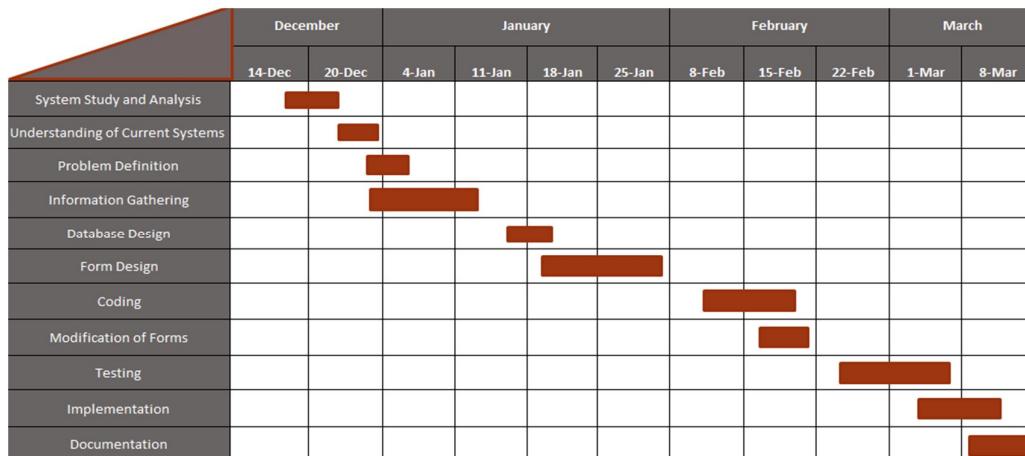


Figure (2.2): Gantt chart

- **The steps which are carrying out according to the Gantt chart are as follow:**
 - i. System study and analysis phase is started on 14/12/2013 and continued up to 20/12/2013. In this phase analysis of the project is done.
 - ii. Understanding current system phase is started on 20/12/2013 and continued up to 04/1/2014. In this phase actual understanding of the contents used for the project is done.

- iii. Problem definition phase is started on 20/12/2013 and continued up to 04/1/2014. In this phase all the problems arise are defined.
- iv. Information gathering phase is started on 20/12/2014 and continued up to 11/1/2014. In this phase all the related information is gathered.
- v. Database design phase is started on 11/01/2014 and continued up to 18/01/2014. In this phase database design is done.
- vi. Form design phase is started on 18/01/2014 and continued up to 25/01/2014. In this phase form design is done.
- vii. Coding phase is started on 08/01/2014 and continued up to 15/02/2014. In this phase coding is done.
- viii. Modification of forms phase is started on 08/02/2014 and continued up to 15/03/2014. In this phase modification and the design of the form is done.
- ix. Testing phase is started on 22/03/2014 and continued up to 01/03/2014. In this phase testing is done.
- x. Implementation phase is started on 01/03/2014 and continued up to 08/03/2014. In this phase actual implementation is done.
- xi. Documentation phase is started on 01/03/2014 and continued up to 08/03/2014. In this phase documentation is done.

2.5 Detailed Life Cycle

2.5.1. ERD (Entity Relationship Diagram)

i. All entities with attributes.

An **entity** is a “thing” or “object” in the real world that is distinguishable from all set of objects. An entity set is a set of entities of the same type that share same properties, or attributes. An entity is represented by a set of attributes.

ii. E-R Diagram.

E-R diagram is a relationship between two entity sets. E-R diagram can express the overall structure of a database graphically. E-R diagrams are simple and clear.

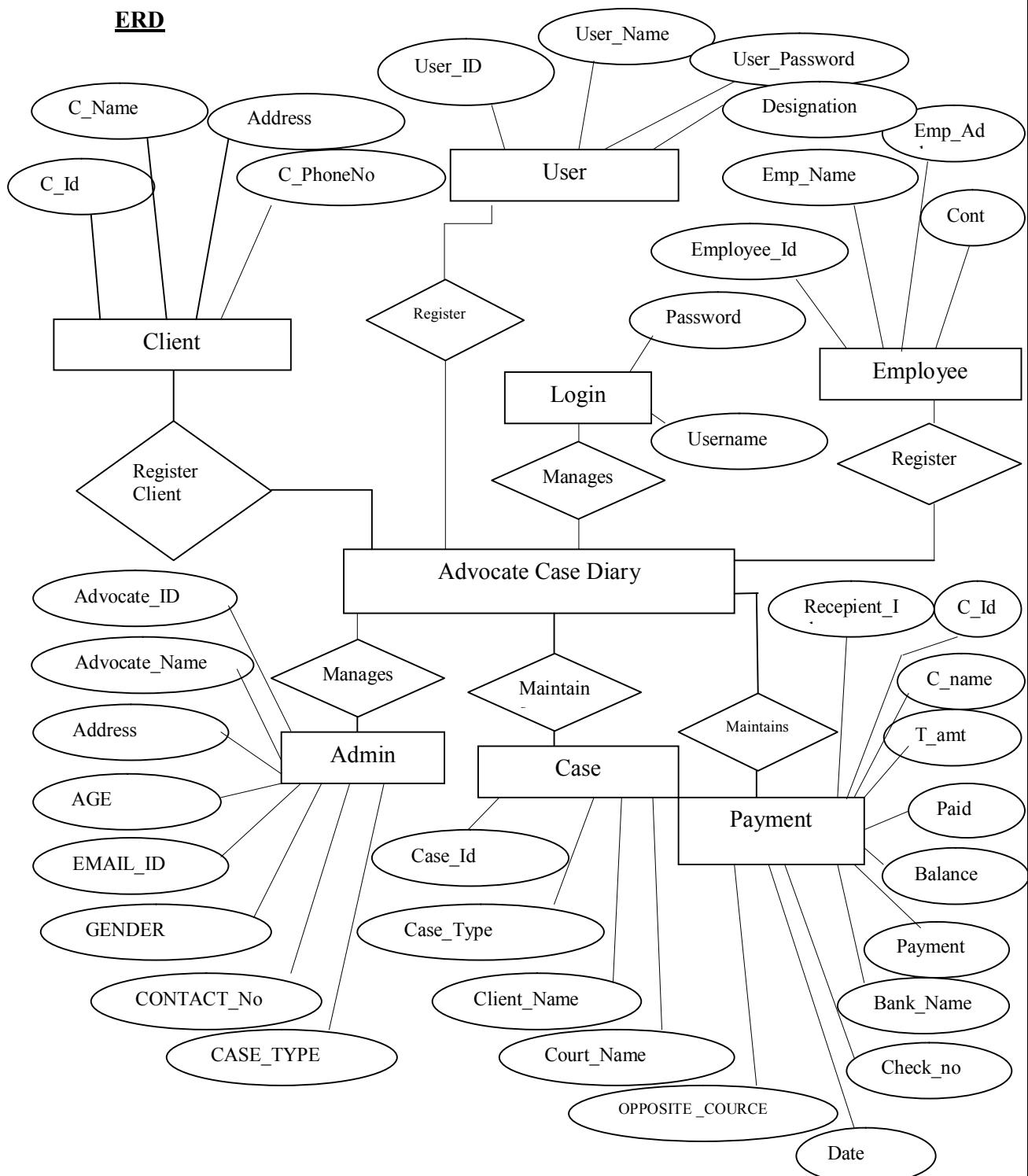
Symbols used:
E-R diagram consists of the following major components:

Sr no	Symbol	Name	Description
1		Rectangles	Represent entity sets
2		Ellipses	Represent attributes
3		Diamonds	Represent relationship sets
4		Lines	Link attributes to entity sets and entity sets to relationship sets
5		Double ellipses	Represent multivalued attributes
6		Double lines	Indicate total participation of an entity in a relationship set
7		Double Rectangle	Represent weak entity set.

Figure (2.3): Symbol notation for the E-R Diagram

Figure (2.4): E-R Diagram

ERD

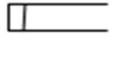


2.5.2 DFD (Data Flow Diagram)

A data flow diagram (DFD) is a graphical system model that shows all of the main requirements for an information system in one diagram: inputs and outputs, processes, and data storage. A DFD describes what data flows rather than how it is processed. Everyone working on a development project can see all aspects of the system working together at once with DFD. That is one reason for its popularity. The DFD is also easy to read because it is graphical model.

The DFD is mainly used during problem analysis. End Users, management, and all information systems workers typically can read and interpret the DFD with minimal training.

Symbol Used:

Sr No.	Symbol	Name	Description
1		Entity	An Entity is source or Destination of a data flow which is outside the area of study. For eg. Customer, Student etc.
2		Process	A Process shows Transformation or manipulation of data flows within the system
3		Dataflow	A data flow shows the flow of information from its source to its destination. A data flow is represented by a line, with arrowheads showing the direction of flow.
4		Datafile or Datastore	A data store is a holding place for information within the system.

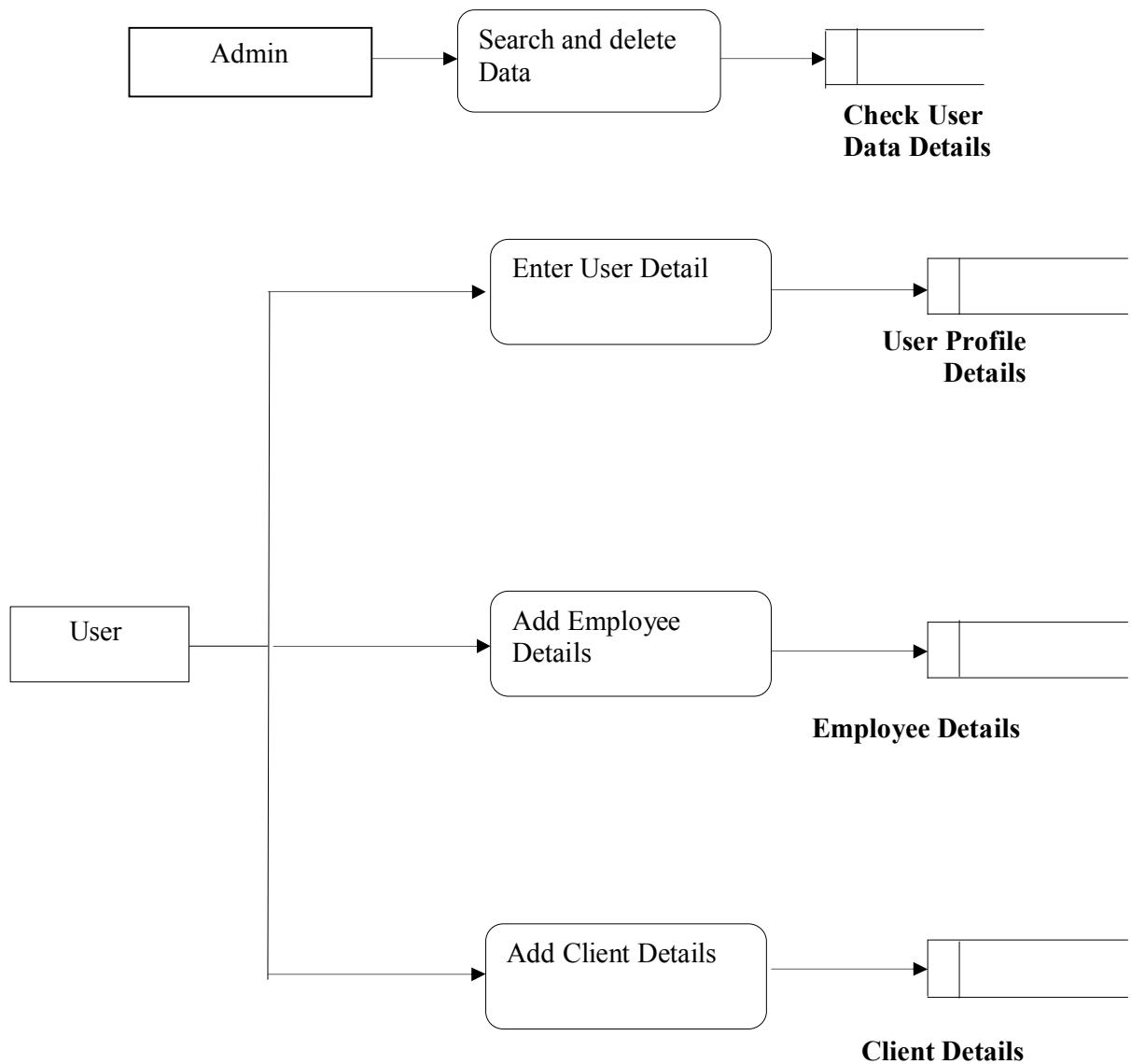


Figure (2.6) DFD diagram

i. Context level data flow diagram

The context diagram is useful for showing boundaries. The system scope is defined by what is represented within single process and what is represented as an external agent. External agents that supply or receive data from the system are outside of the system scope. Everything else is inside the system scope.

Data stores are not usually shown on the context diagram because all of the system's data stores are considered to be within the system scope. The context diagram is simply the highest-level DFD. It is also called as Level 0 DFD.

The context diagram provides a good overview of the scope of the system, showing the system in “context” but it does not show any detail about the processing that takes place inside the system.

2.5.2 CONTEXT LEVEL DAIGRAM

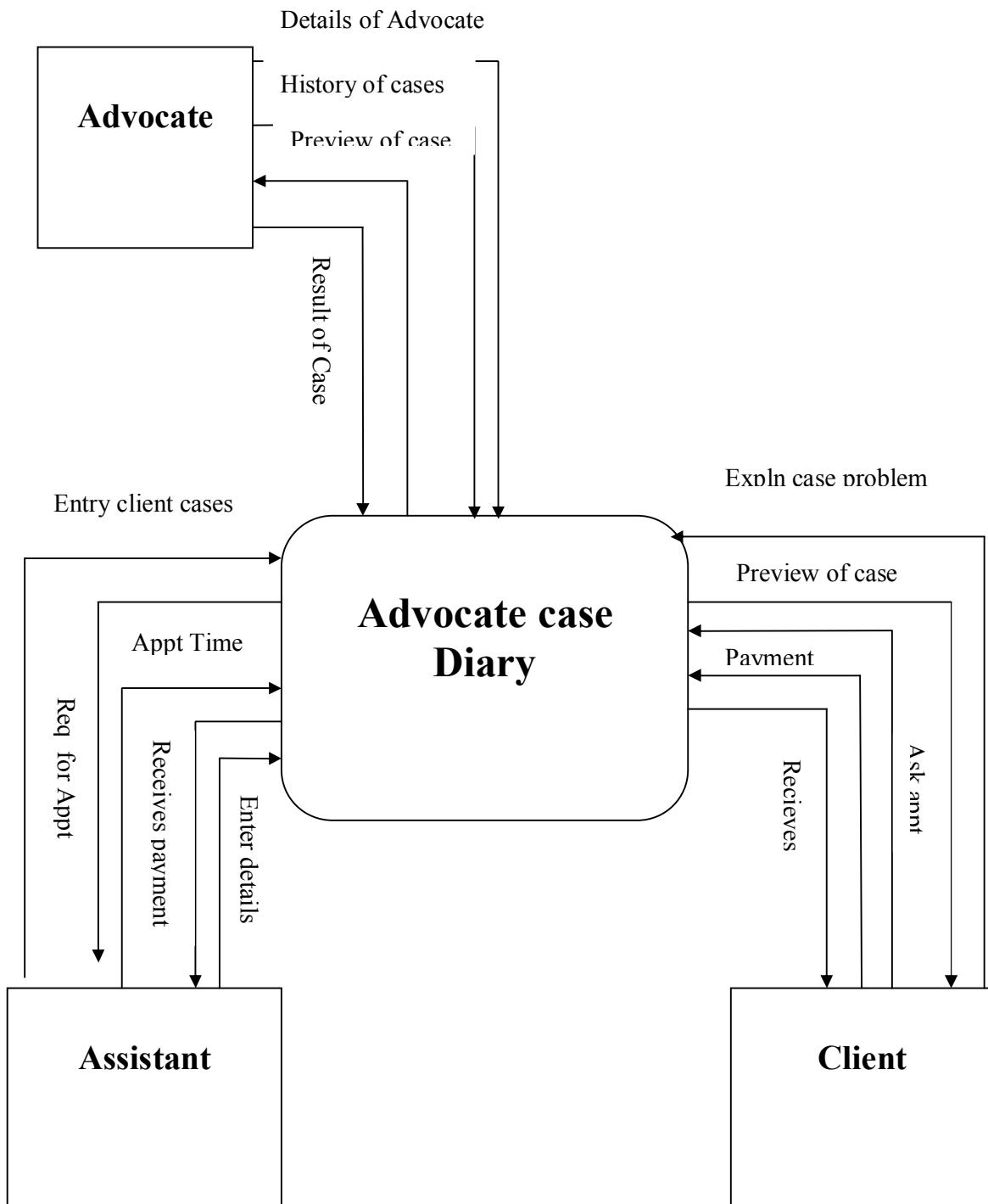
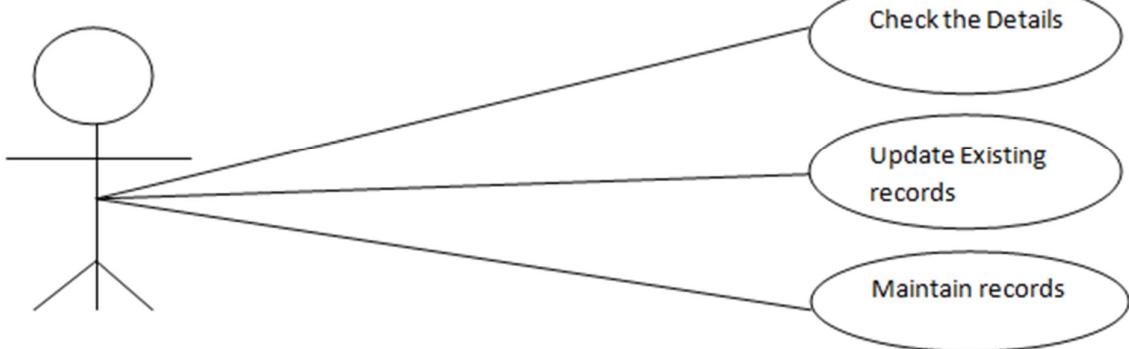


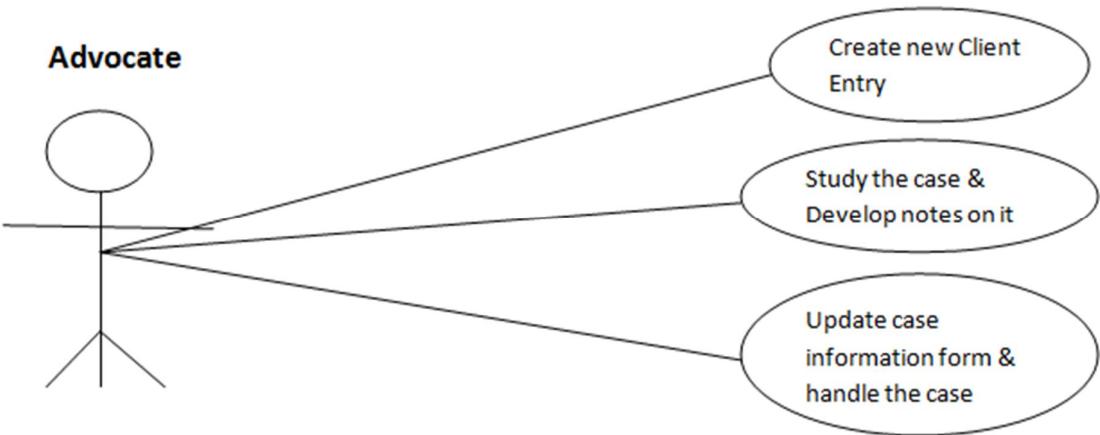
Figure (2.7) Context Level DFD Diagram

2.5.3 USE CASE DIGRAM

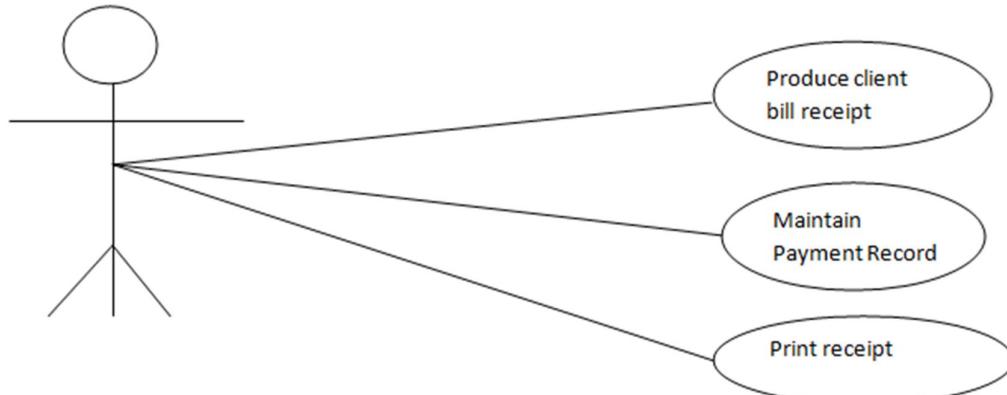
Assistant



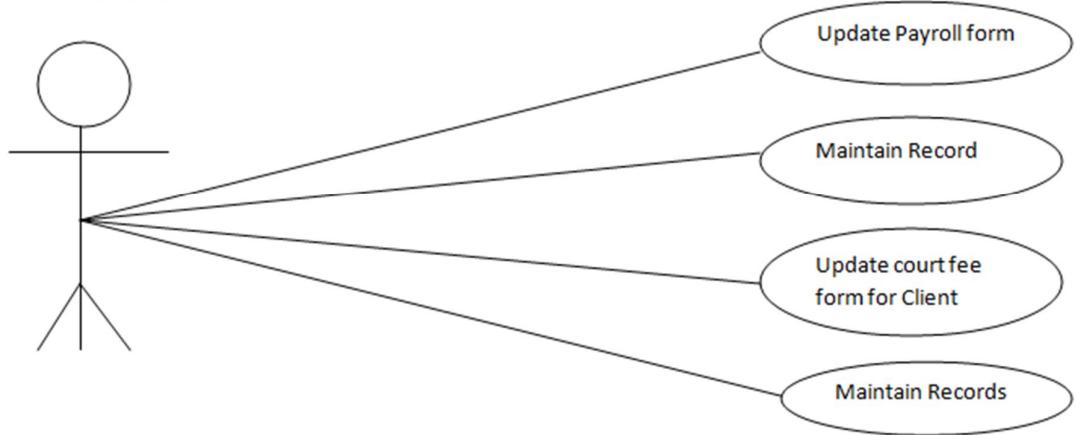
Advocate



Assistant



Advocate



Assistant

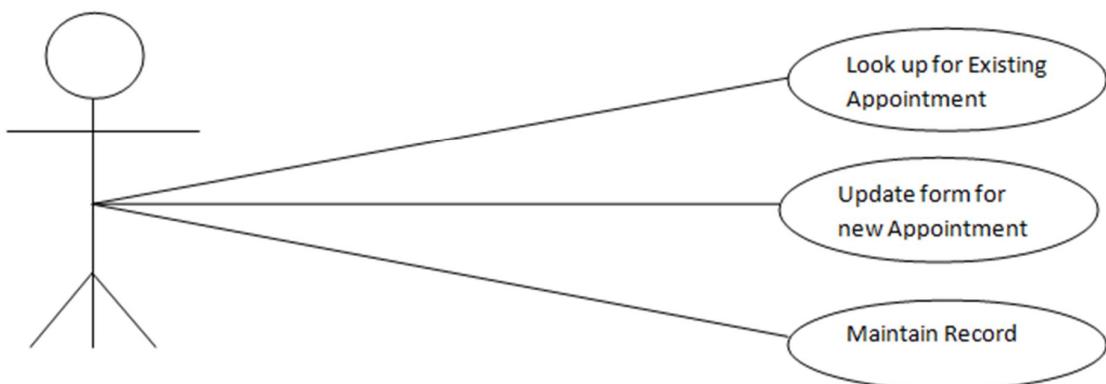


Figure (2.8) use case diagram

2.5.4 CLASS DIGRAM

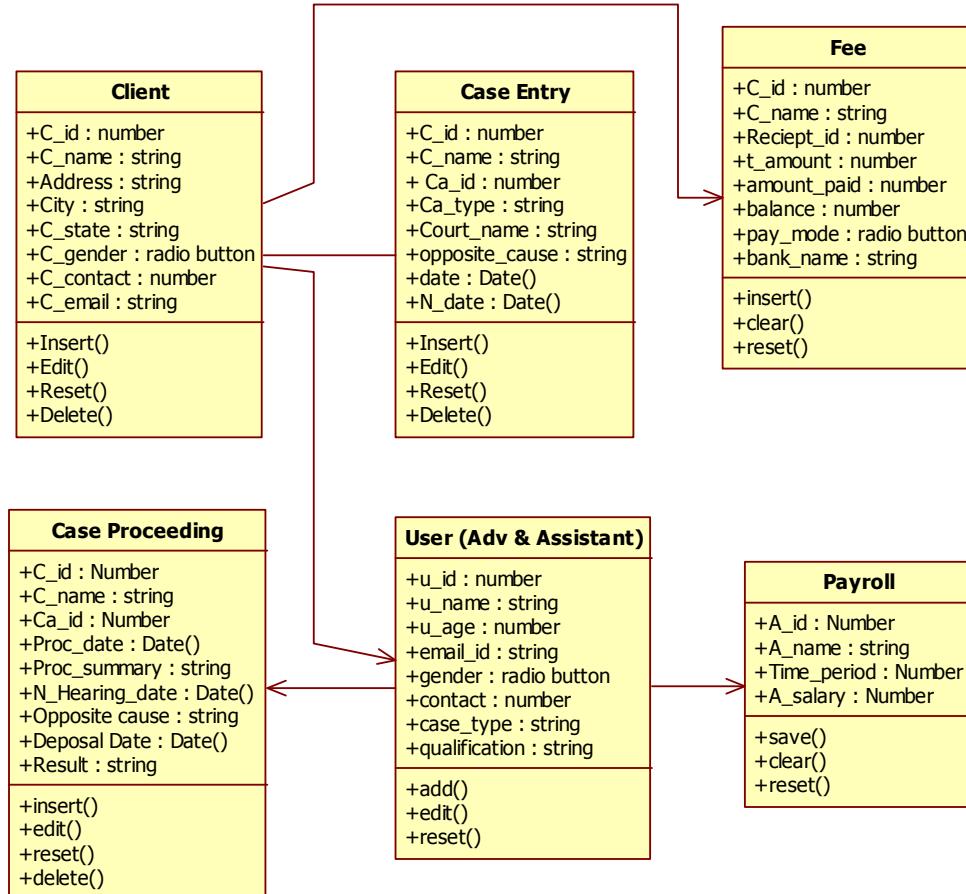


Figure (2.9) Class Diagram

2.5.6 Collaboration DIGRAM

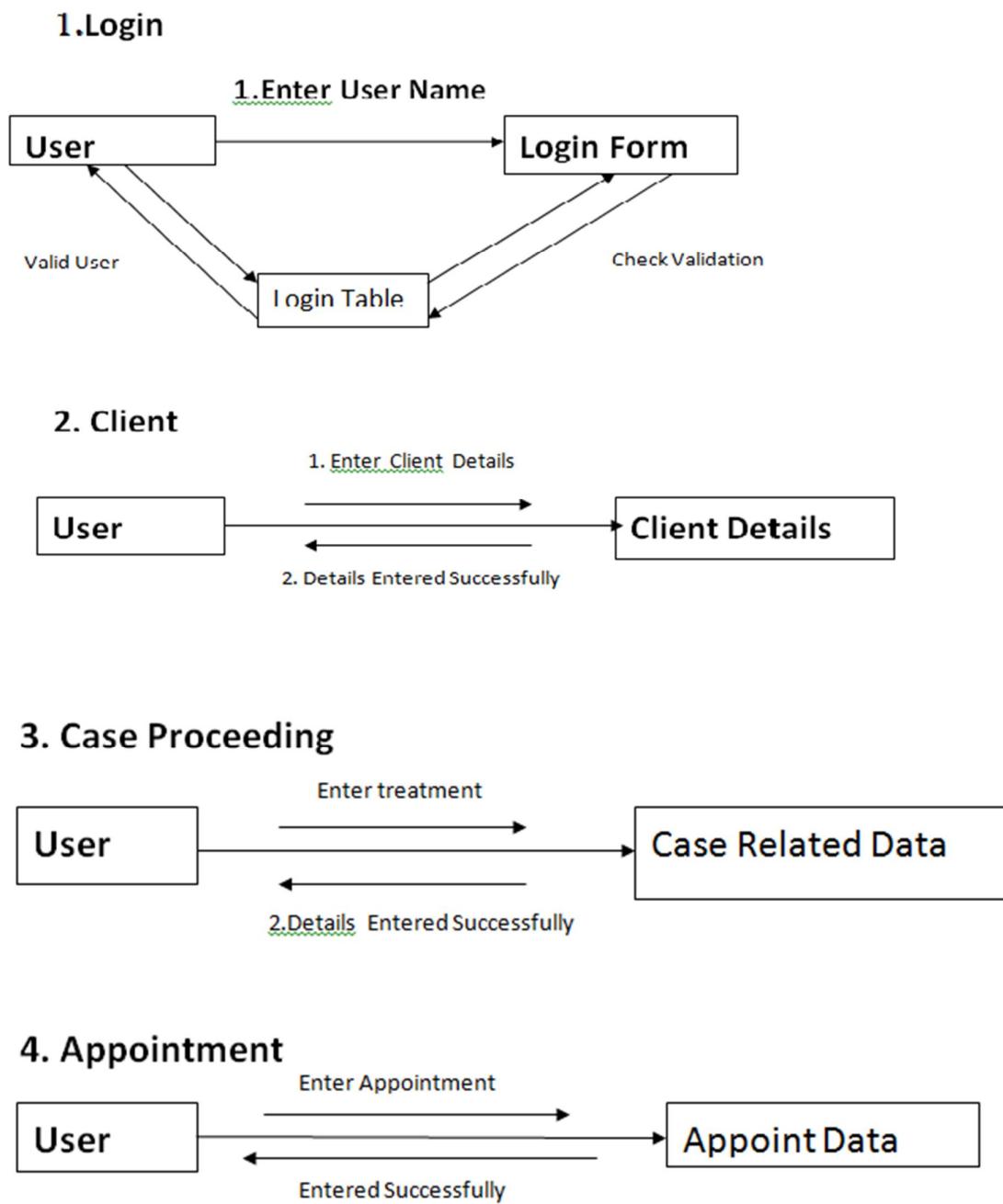


Figure (2.10) Collaboration Diagram

2.5.7 Activity Diagram

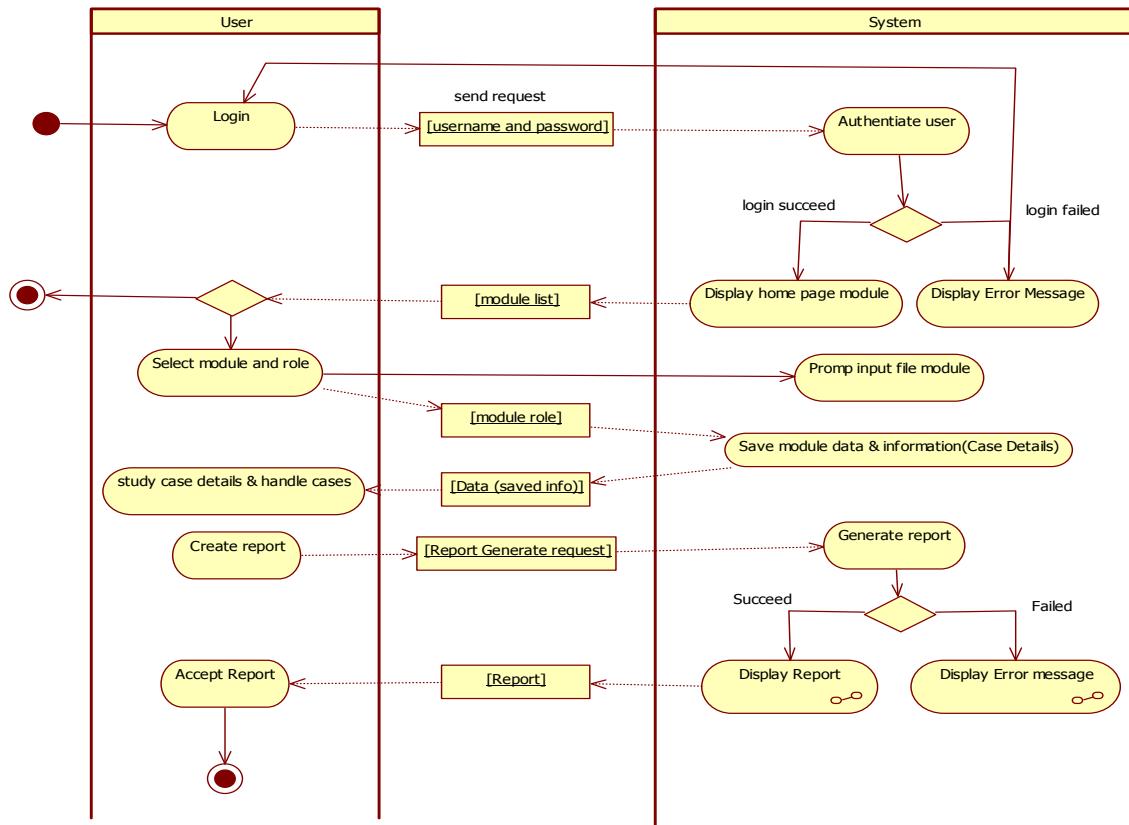


Figure (2.11) Activity Diagram

CHAPTER 3

Methodology Adopted, System Implementation and Details of hardware and software

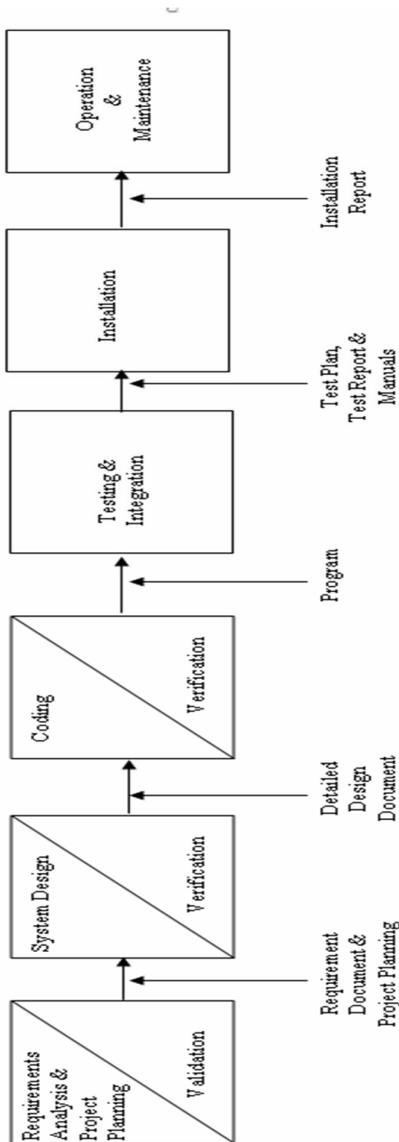
3.1 PROCESS MODEL – SOFTWARE ENGINEERING

In the development of software we have used **Waterfall Model**, the linear sequential mode. This model encompasses the following activities:

a) Analysis Phase:

System Analysis:-

This refers to the gathering of system requirements, with the goal of determining how these requirements will be accommodated in the system.



b) System Design Phase:

This is actually a multistep process. In this we tried to focus on some distinct attributes of a program like data structure, software architecture, interface representations and algorithmic detail. In this we tried to translate requirements into representation of the software which can be assessed for quality before coding begins. In the verifications, I have tried to ensure that the design is satisfying the requirements and is of good quality. I have tried to find out if there is any misinterpretation of specified any requirements.

c) Code Generation Phase:

In this phase, we translated design of a system into code which can be compiled and executed. In this phase we have done actual coding for all forms. In this we tried to produce simple program which are clear to understanding and modify.

We have used dynamic method to verify the code. We have executed program on some test data and output of the program examined to determine if there are any error present. I have read the code carefully to detect any discrepancies between the design specification and the actual implementation.

d) Testing:

Testing plays a critical role in quality assurance for software. Due to limitations of the verification methods for the previous phase, design and requirement faults also appear in the code. Testing is used to detect these errors, in addition to the errors introduced during the coding phase.

3.2 System Implementation

In computer science, an implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system. Many implementations may exist for a given specification or standard. For example, web browsers contain implementations of World Wide Web Consortium-recommended specifications, and software development tools contain implementations of programming languages.

In the IT Industry, implementation refers to post-sales process of guiding a client from purchase to use of the software or hardware that was purchased. This includes Requirements Analysis, Scope Analysis, Customizations, Systems Integrations, User Policies, User Training and Delivery.

Software Implementations involve several professionals that are relatively new to the knowledge-based economy such as Business Analysts, Technical Analysts, Solutions Architects, and Project Managers.

3.3 Details of Hardware and Software

3.3.1. Hardware Requirements:

- Pentium 300 MHz or faster recommended
- Best display resolution 1024x786
- Display with more than 256-colors recommended (High Color)
- 32 MB of RAM
- 15 MB free hard drive space recommended
- Windows XP/2007 or a newer 32-bit Windows operating system.

3.3.2. Software Requirements:

- Visual Studio 2010
- SQL server 2005 or higher

3.3.2.1. ASP.NET 2010

Visual c# is a programming language based on the original DOS language called BASIC (Beginners' All-purpose Symbolic Instruction Code). VISUAL STUDIO 2010, an offshoot of Visual c#, is an object-oriented programming language based on VB that is implemented using the Microsoft .NET framework. The basic syntax of the Visual c# language remains unchanged in VISUAL STUDIO 2010, but includes additional features such as structured exception handling and short circuited expressions to enhance the infrastructure of the programming language. Visual c# programmers supporting ASP.NET indicate that language constructs and user interface features in the newer system have caused programming issues within the original VB system. Developers using ASP.NET recognize that the upgraded VB system incorporates contemporary object oriented programming paradigms in a more stable environment than was originally thought possible.

Visual c# .NET is a programming language that can be used to create window forms or web-app applications and create programming based in any object-oriented programming language (OOP), ADO.NET, Multithreading or Windows Services. ASP.NET programming utilizes concepts connected to string encryption, forms inheritance, regular expressions and deployment.

3.3.2.2. SQL SERVER

The advent of the .NET Framework has brought with it some major changes, and database access is certainly no exception. Although ADO has been around since 1996, with the arrival of ASP, its latest incarnation present in the .NET Framework - ADO.NET - really does represent its coming-of-age. We'll be covering connecting to a SQL Server database, executing queries, calling stored procedures, filtering data, and reflecting changes in a database - and point you in the direction of further resources for more in-depth discussions on particular topics.

In order to use ADO.NET, we'll need to import two namespaces:

[Visual c#]
Imports System.Data
Imports System.Data.SqlClient

In order to establish a connection with SQL Server, we use the SqlConnection object. There are also generic OleDbConnection and OdbcConnection objects present in the System.Data.OleDb and System.Data.Odbc namespaces for use with other data sources such as Access (NB: support for ODBC was added in .NET 1.1). This would work in exactly the same way as illustrated here, but as Microsoft were kind enough to provide a data access class specifically optimized for SQL server, we might as well take advantage of it! If you're using Access, then you can literally replace every occurrence of *SqlSomething*. With *OleDbSomething*.

Opening a connection is very simple:

[Visual c#]

```
'create a new SqlConnection object with the appropriate connection string  
con As New SqlConnection  
' open the connection  
con.Open()  
// do some operations here...  
// close the connection  
con.Close()
```

with the connection string usually taking this form:

```
con.ConnectionString = "Data Source=.;Initial Catalog=advo  
;Integrated Security=True"
```

The idea behind connection pooling is simple - instead of incurring a large amount of overhead each time a connection to database server is established and closed, once a connection has been opened, it remains open for the lifetime of the process, available to be used again if needed. Pooling database connections can significantly enhance the performance and scalability of your application. The .NET data providers automatically pool connections for you. However, to take maximum advantage of these, you should take note of the following:

- Connections are only pooled if they have the same connection string; so ensure this is always constant.
- When finished with a SqlConnection object, call Dispose () or Close () to release the connection back to the pool.
- In order to keep the maximum number of connections available, you should keep connections open for a short as period as possible - remembering that thanks to connection pooling, re-opening a database connection will incur little overhead.

Once open, you can't do much with a SqlConnection object on its own - other than close the connection again, and query its connection status using the Connection State property, so we'll move on to how we go about querying the database.

CHAPTER 4

User/Optional Manual

4.1 User / Operational Manual

This is a guide for the user, several pages in length that explains how to use your system. It should include screen displays; refer to error messages, which may crop up and how to recover from these errors.

- 1) This is a guide for the user of this system.
- 2) This contains all the relevant screen display that will help the user to understand the system.
- 3) There are proper uses of comments that will help the user to understand in case of error.
- 4) Proper validation is done and message box are prompted to the user before entering data into the database.
- 5) The comments describe what exactly the code block does.

4.2 Step to install SQL Server 2005

- i. Insert SQL Server 2005 CD
- ii. You will get a SQL Server 2005 wizard screen from that chooses “SQL Server 2005 components”.
- iii. Choose “Install Database Server”.
- iv. Choose the **Next** button for default setting.
- v. On the following screen, enter your name (or initials).
- vi. Choose the defaults from the screens.
- vii. On the following screen choose “Use the same account for each service. Auto starts SQL Server service” and “Use the local system account”.
- viii. Choose the defaults from the screens.
- ix. From the following screen choose “Ignore the security threat warning, leave the password blank” (if this were a “production” system you obviously should choose a password but for our purposes it is not necessary).
- x. On the following screen, leave the first box unchecked and check the 2nd box.
- xi. Take the defaults from the following screens.
- xii. You will see the following after the files are installed. Since this is a new installation and is only for learning purposes don’t worry about what the message says.
- xiii. Press Finish.
- xiv. To start using SQL Server, Start the “SQL Server Enterprise Manager”.

4.3 Step to install Visual Studio 2010

- i. Any pre requirement prior to the installation will be checked and verified by the setup wizard. You need to have the VS 2010 DVD. Insert the DVD into the DVD drive, the Autorun will be executed, displaying the following Windows form. Click the View ReadMe button to read the readme information. Then click the Install Visual Studio 2010 link to start the installation.
- ii. The setup wizard will start copying needed files into a temporary folder. Just wait.
- iii. In the welcome setup wizard page you can enable the tick box to send your setup experience to Microsoft if you want. In this case we just leave it unchecked. Just wait for the wizard to load the installation components.
- iv. Click the Next button to go to the next step.
- v. The setup wizard will list down all the required components need to be installed. Any already installed components will also be mentioned. Notice that VS 2010 (version 8.x) needs .NET Framework version 3.5. Key in the Product key and accept the license terms. Then click the Next button.
- vi. In the installation type, as usual we have three choices: **Default, Full or Custom**. In this case we select the Full installation type and accept the default installation path given. You can change the installation path and the required space for every installation type also displayed when we select it.
- vii. In this case, select the Full and click the Install button. Full installation required around 4.3 GB of space. The installation starts. Just wait and see the step-by-step, Visual Studio 2010 components being installed.

4.4 Installation steps for project

1. Install Visual Studio 2010 on your machine.
2. Install SQL Server 2005 (recommended) on your machine.
3. Copy Advocate Case Diary folder in your 'E' drive.
4. Establish database connectivity as illustrated below:
 - 4.1 Attach the Advocate Case Diary database present in 'Database' folder with 'local windows NT server'.

4.5 Execution steps for the project

1. There are two ways to open the project.
 - A)
 1. Start the Visual Studio 2010.
 2. Click on File menu → Open → File submenu → Select appropriate path as (E:/Advocate Case Diary/ icon).
 - B)
 1. Open My Computer → Choose E drive → Click on Advocate Case Diary folder → Double click on the icon.
 2. To run this project press F5 key of keyboard or Choose Debug menu → Start Debugging.
 3. Splash screen will be displayed first and then Login form will be displayed.
 4. Choose Administrator and Password is 'admin'.

CHAPTER 5

GUI Interface

3.5.5 DATA MODULES AND THEIR DESCRIPTION

1) Login Page :

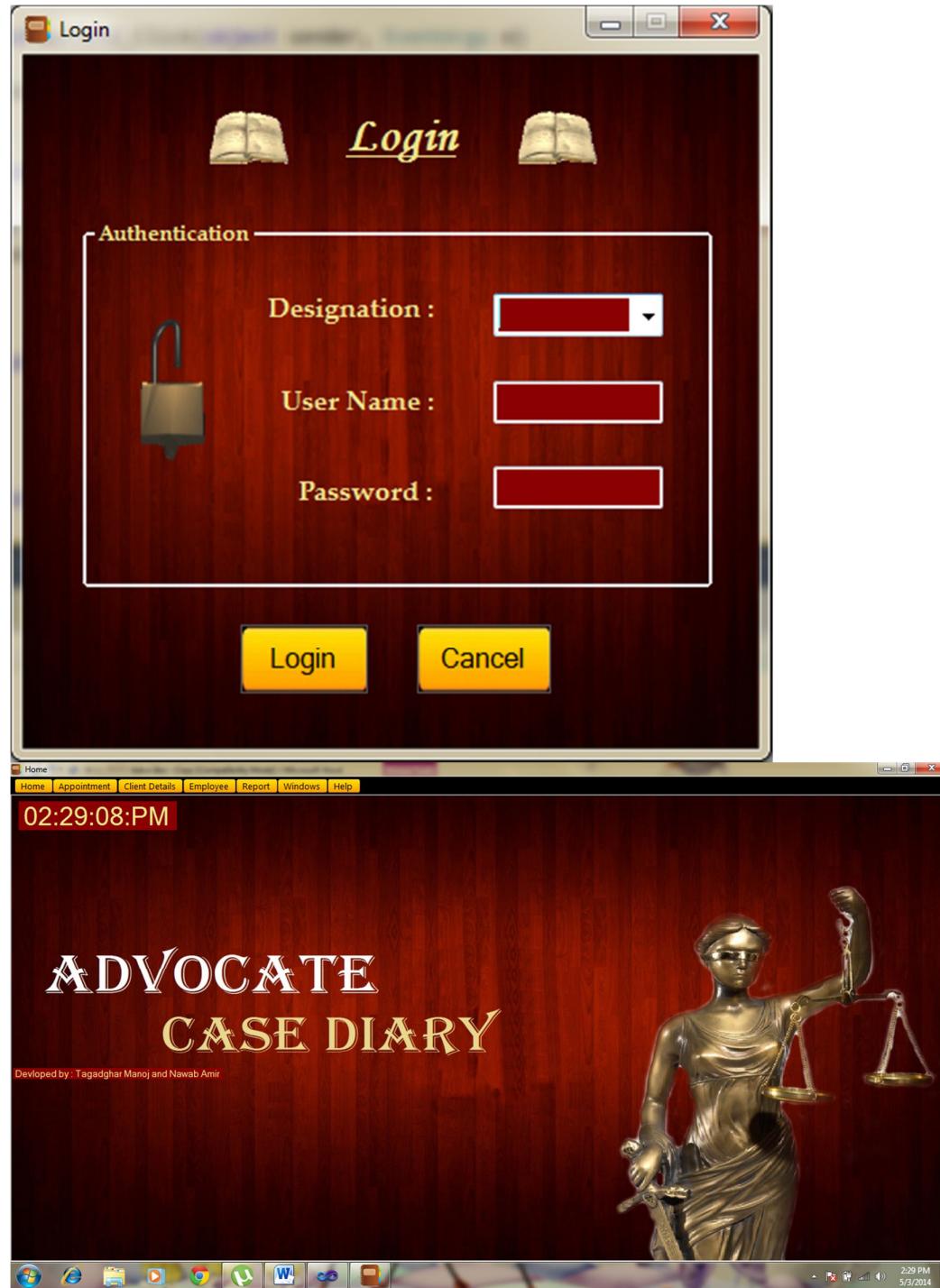


Figure (5.1) Main Page

2) Advocate Details



Figure (5.2) Advocate Details

3) Search Cases



Figure (5.3) Search Cases

4) Appointment Entries



Figure (5.4) Appointment Entries Form

5) Client Entry



Figure (5.5) Client Entry Form

6) Case Entry



Figure (5.6) Case Entry Form

7) Proceeding Case Entry



Figure (5.7) Proceeding Case Entry Form

8) Payment



Figure (5.8): Payment Form

9) Installments Payment



Figure (5.9) Installments Payment Form

10) Update Client Details

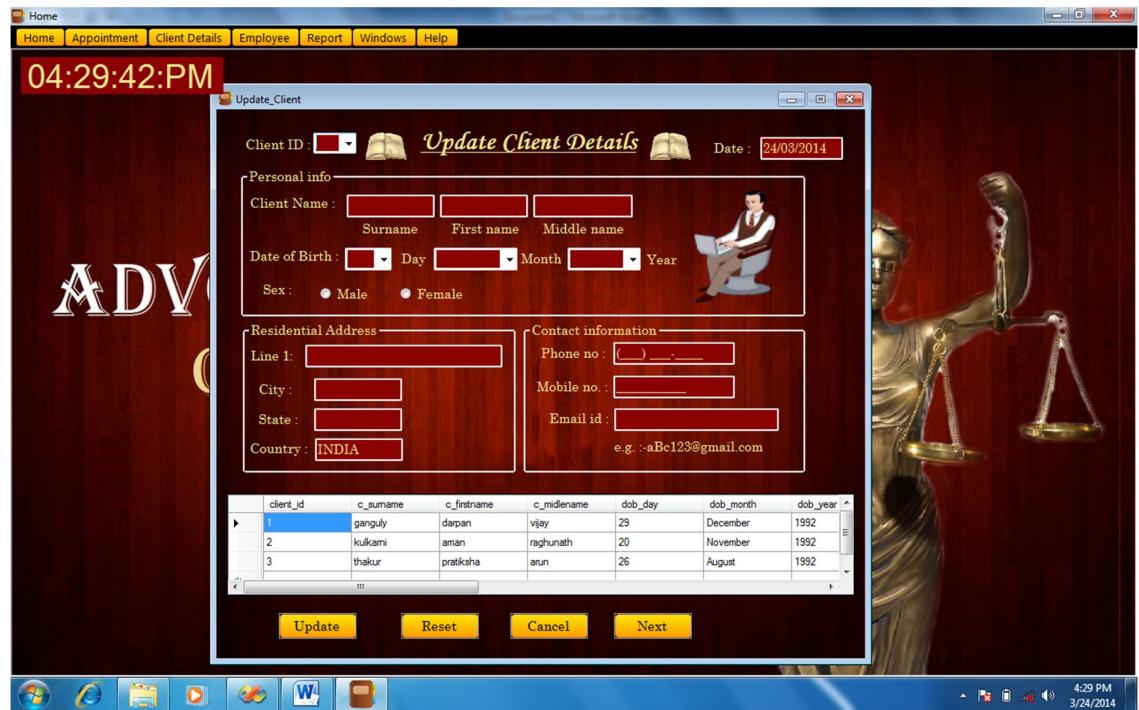


Figure 5.10) Update Client Details Form

11) Updating Case Information

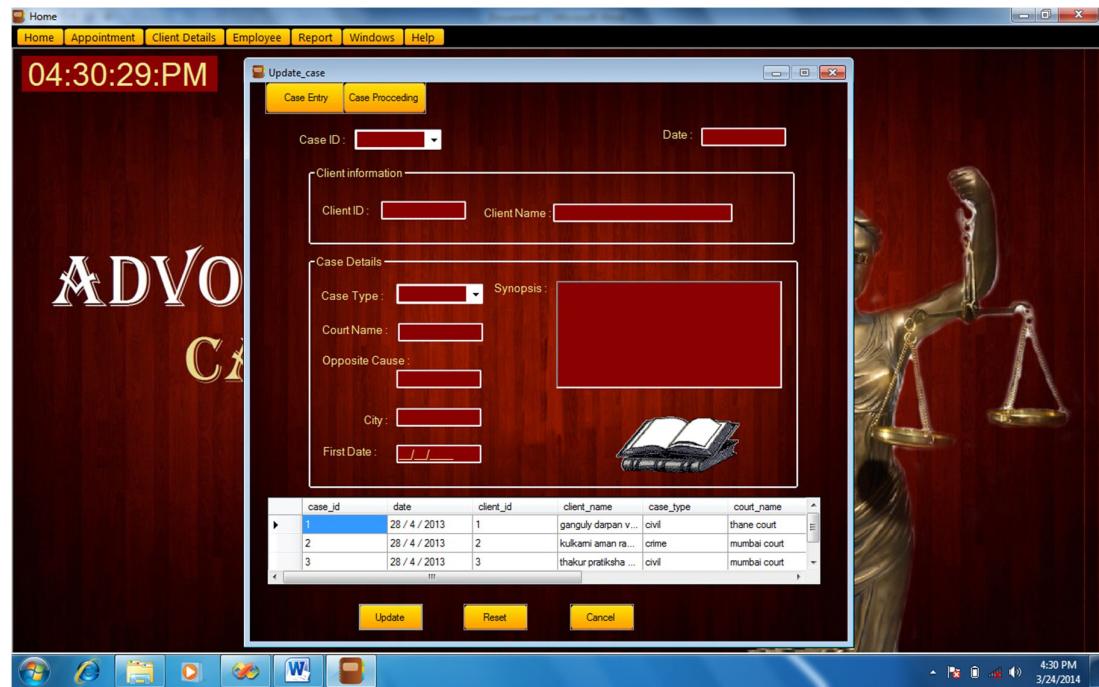


Figure (5.11) Updating Case Information Form

12) Search Client Details



Figure (5.12) Search for client cases Form

13) Employee Menu Bar

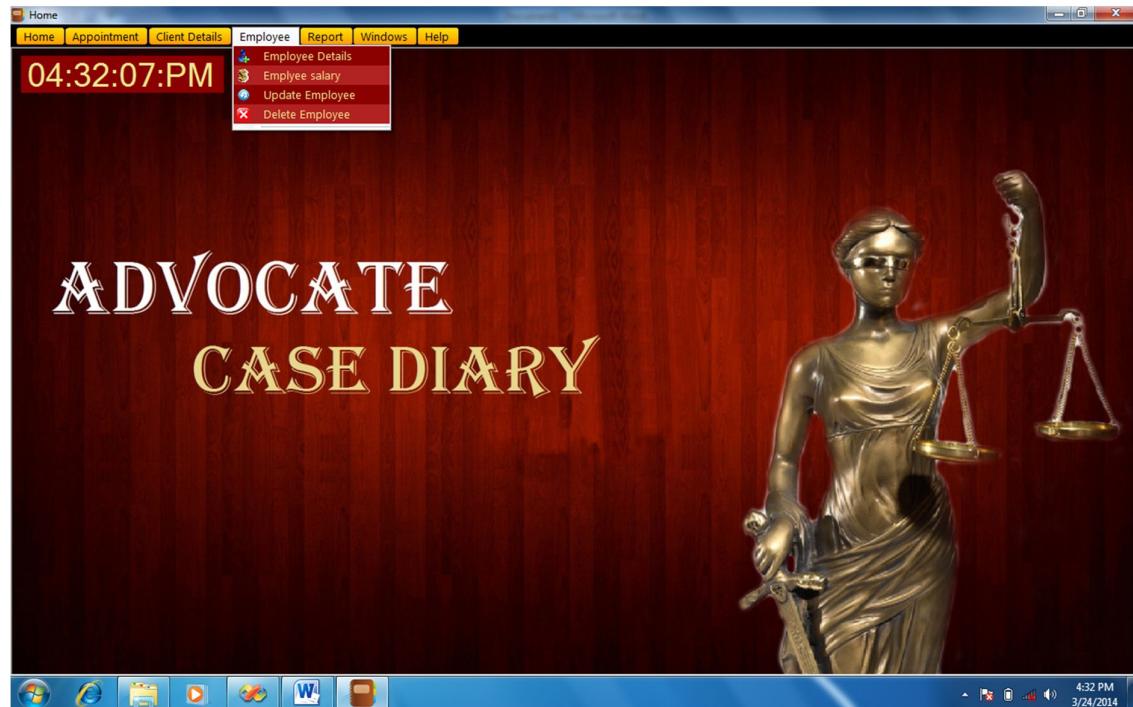


Figure (5.13) Employee Menu Bar

14) Employee Details



Figure (5.14) Employee Details Form

15) Employee Payment



Figure (5.15) Employee Payment Form

16) Employee Update

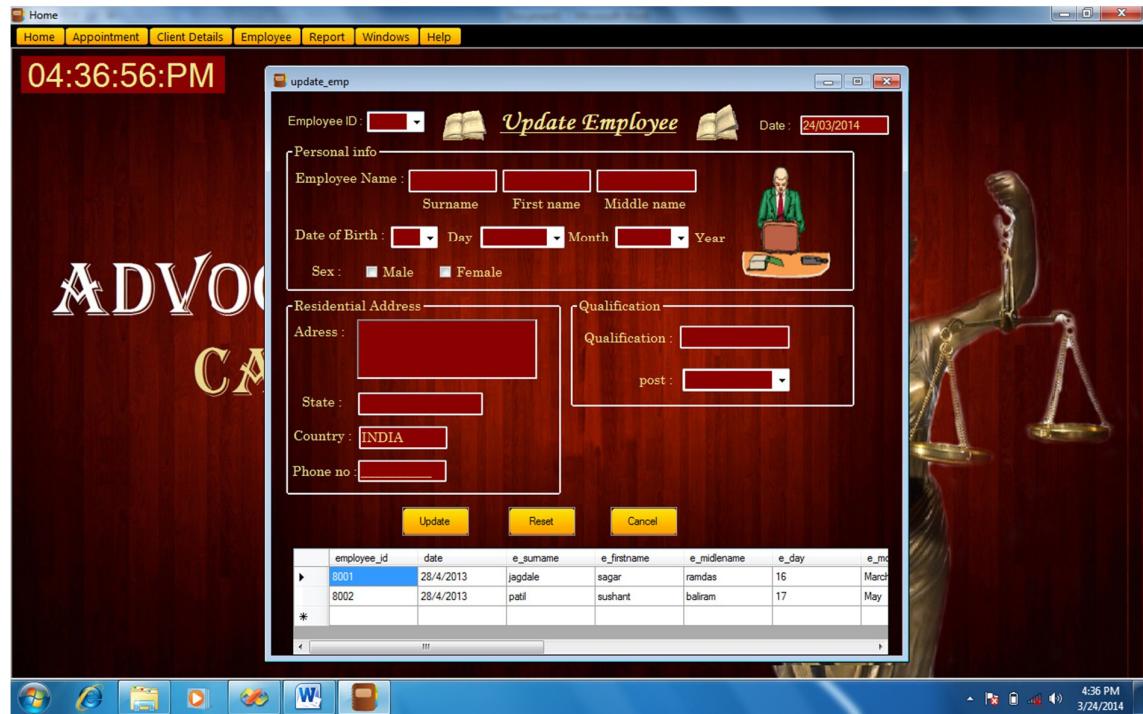


Figure (5.16) Employee Update Form

17) Search and Delete Employee Details



Figure (5.17) Search and Delete Employee Details Form

18) Report Menu Bar

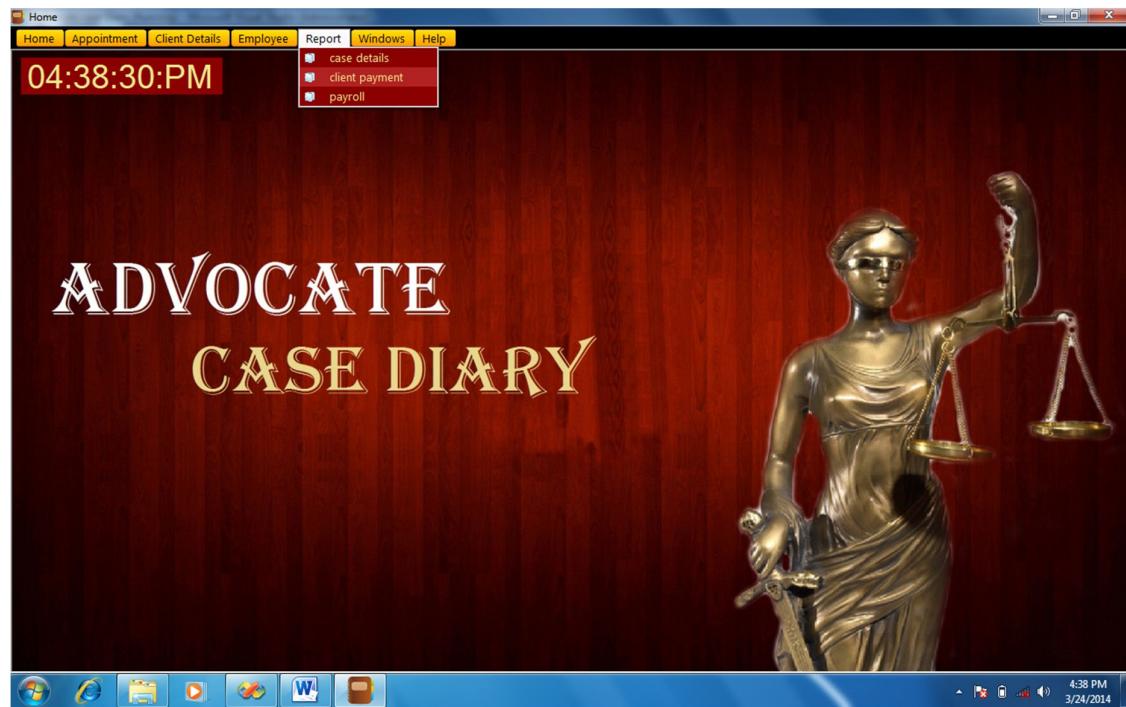


Figure (5.18) Report Menu Bar

19) Case Report



Figure (5.19) Case Report Form

20) Payment Report

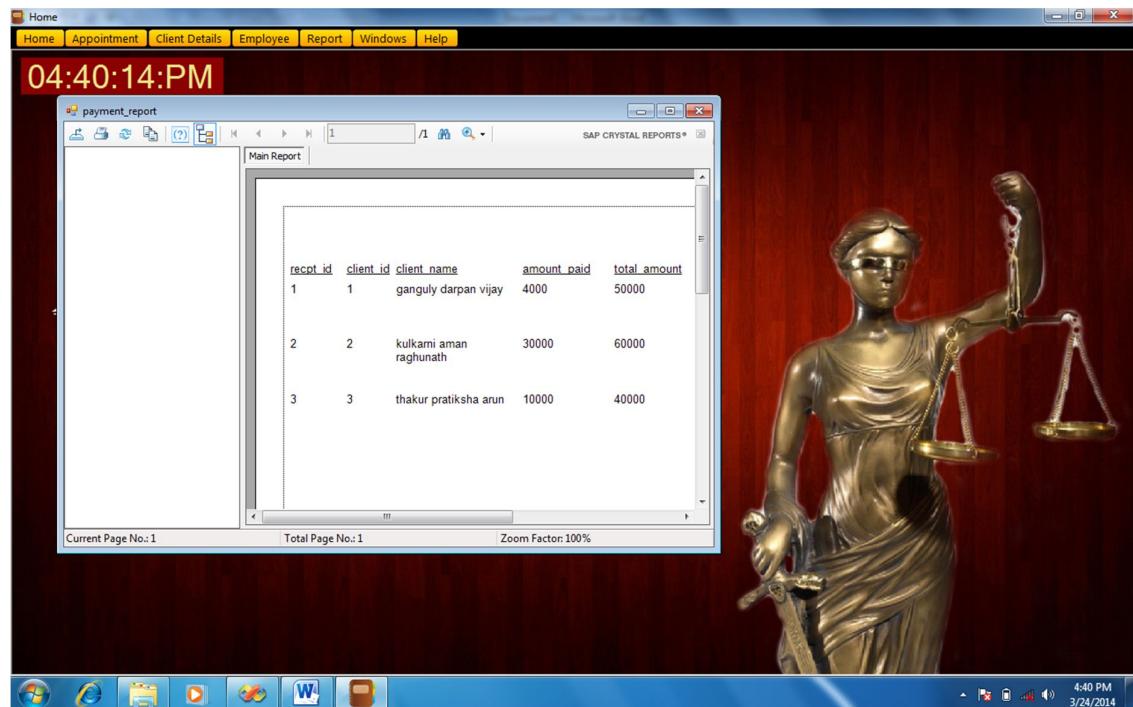


Figure (5.20) Payment Report

21) Payroll Report

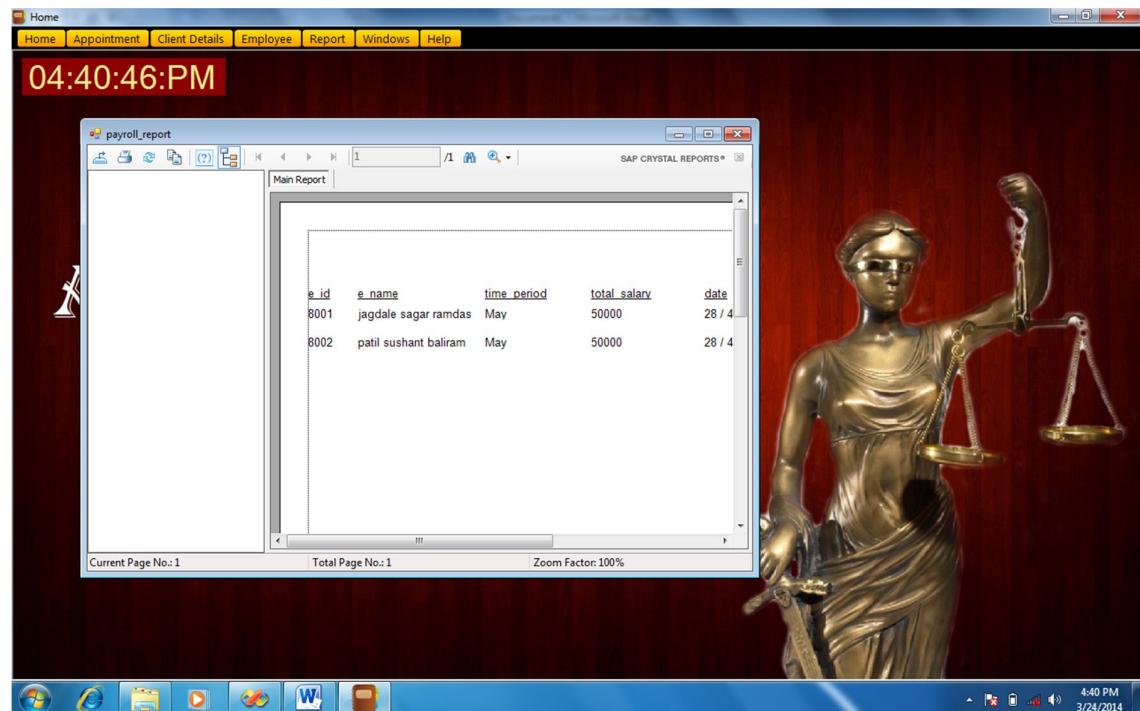


Figure (5.21) Payroll Report Form

22) Windows Menu Bar

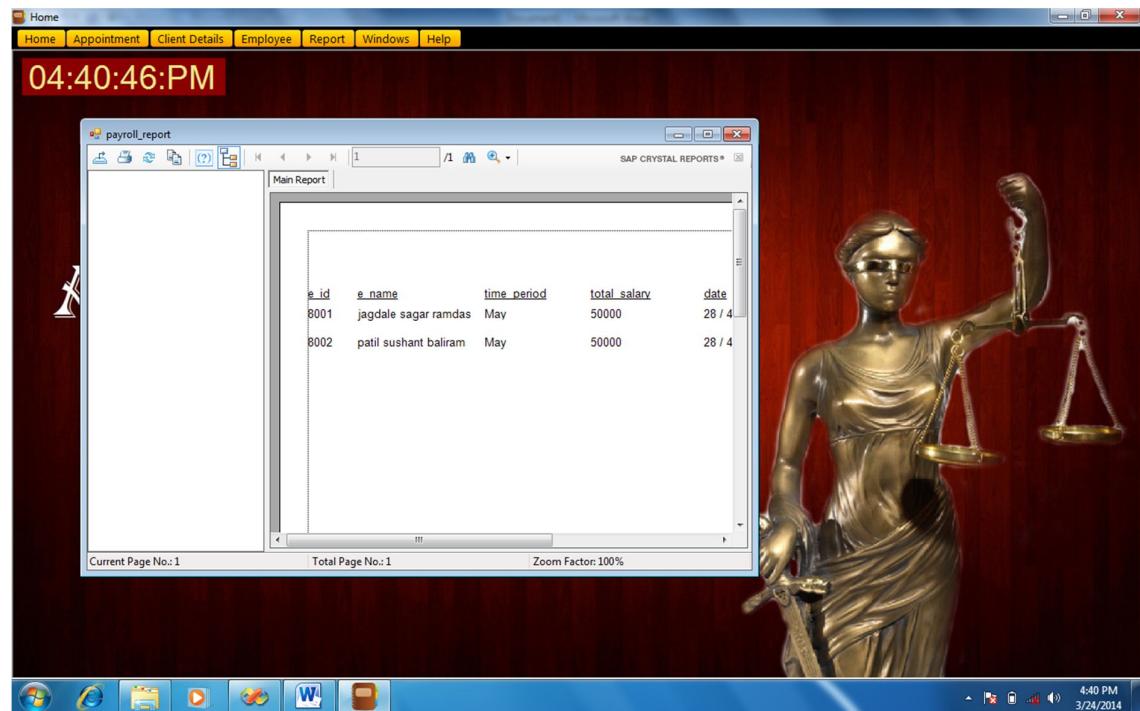


Figure (5.22) Windows Menu Bar

23) Create User Account



Figure (5.23) Create User Account Form

24) Change Password



Figure (5.24) Change Password Form

25) Help Menu Bar

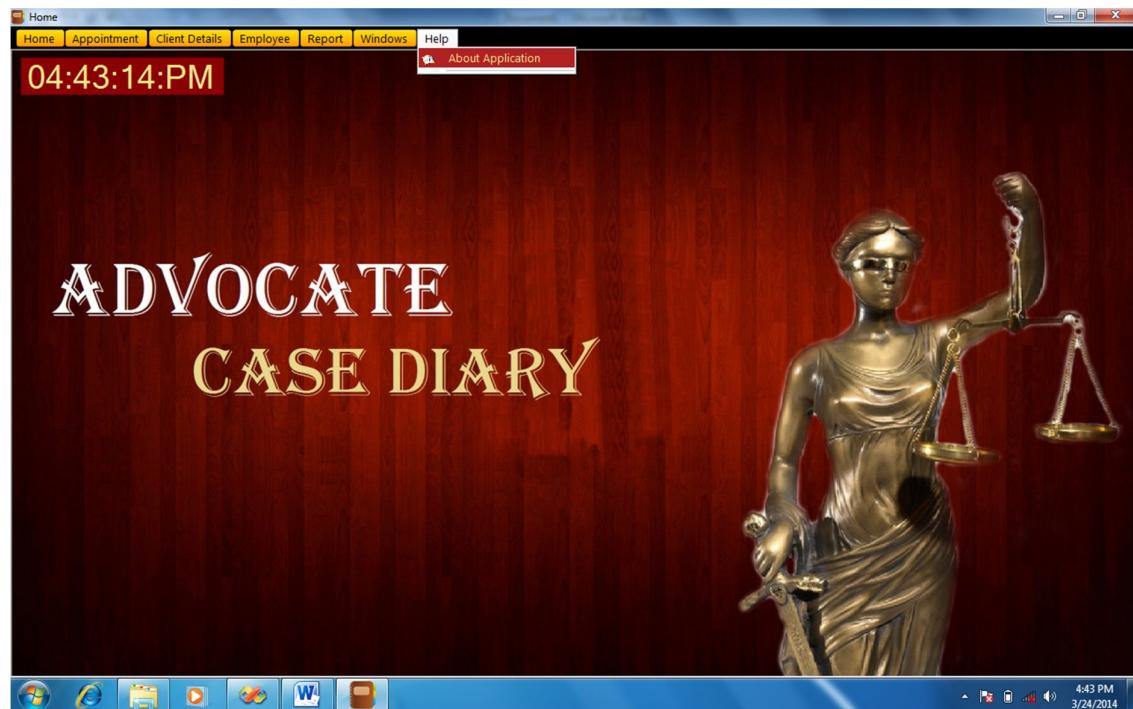


Figure (5.25) Help Menu Bar

26) About Application



Figure (5.26) About Application Form

27) Client Entry



Figure (5.27) Validations for completing fields

28) Client Entry

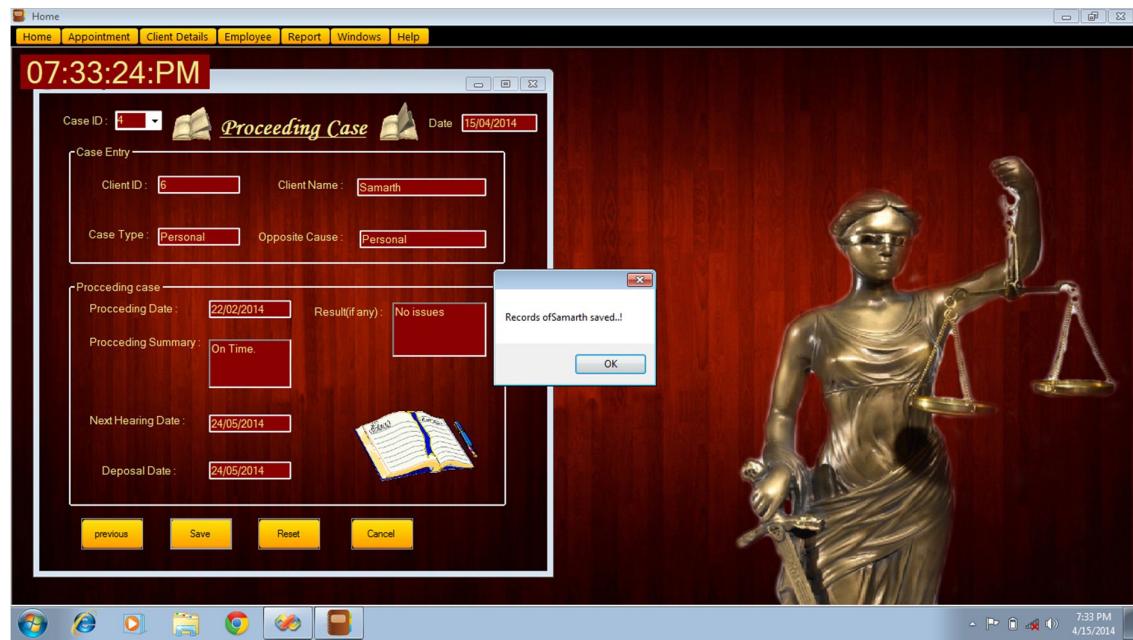


Figure (5.28) Client Entry Validation for Saved Record

29) Employee Payment

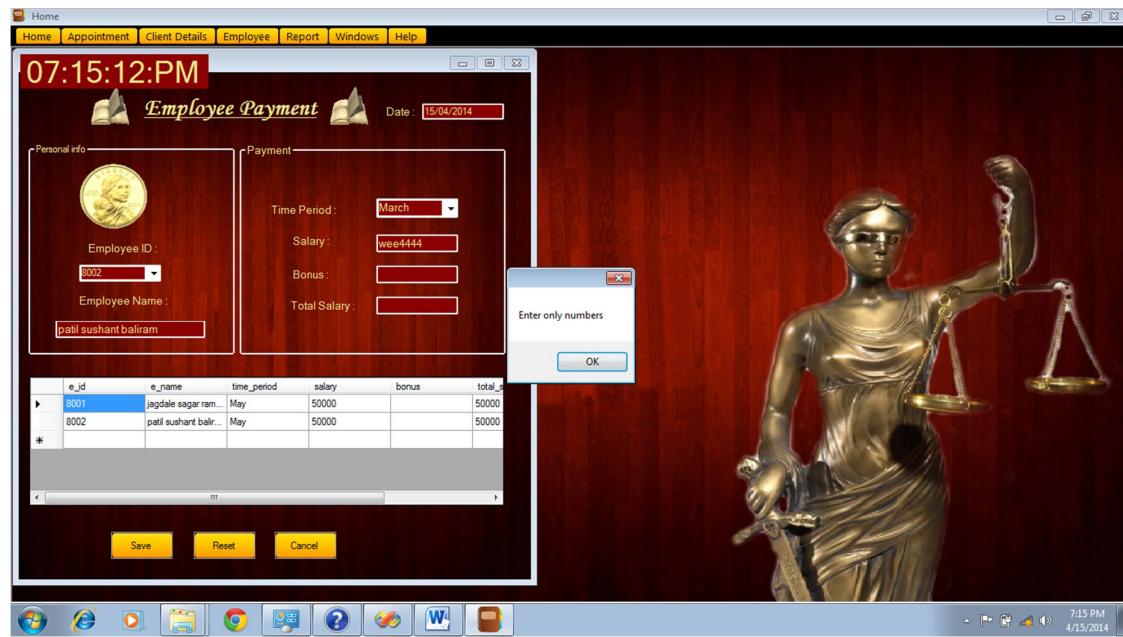


Figure (5.29) Validation For numbers

30) Validation



Figure (5.30) Validation

31) Valid Id Entry



Figure (5.31) Valid Id entry message box

CHAPTER 6

Methodology used for testing

6.1 TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, coding.

6.2 Testing objectives

1. Testing is a process of executing a program with the intent of finding an error.
2. A good test case is the one that has high portability of finding an as-yet undiscovered error.
3. A successful test is one that uncovers an as-yet discovered error.

The main objective here will be to design test cases to uncover different classes of errors and to do so with minimum amount of time and efforts. If testing is conducted successfully it will uncover errors in the software. Another advantage is that it demonstrates that software functions appear to be working according to the specifications and performance requirement have been met.

But Testing cannot show the absence of defects it can show only that software errors are present.

6.3 Strategies used for software testing

The software engineering process is viewed as spiral as shown:

Unit testing begins at the vortex of the spiral and concentrates at the each unit of the software as implemented in the source code. Unit testing assures each module tested individually functions properly as a unit.

Integration testing focuses on the design and construction of the software architecture. It is a systematic technique for constructing a program structure while conducting tests to uncover errors associated with interfacing. The objective is to take unit-tested module and build a program structure that has been specified as design.

Validation testing takes care of the requirements established as part of the software requirements analysis are validated against the software that has been constructed. It is said to be successful when the software functions in a manner that can be reasonably expected by the customer.

System testing tests software and other system elements as a whole. These tests fall outside the scope of software engineering process and are not conducted solely by the software developer.

CHAPTER 7

Cost and Benefit

Analysis

7.1 Costs and Benefit Analysis

7.1.1 Need of cost benefit analysis

Why should you do a cost-benefit analysis for your project? IT projects frequently over promise and under deliver. Executive managers have become aware of this performance issue and the cost-benefit analysis is their guarantee that the project team has carefully evaluated the project before commencement, studying the whole life cycle costs and the expected benefits.

7.1.2 Cost Estimation

For a given set of requirements it is desirable to know how much it will cost to develop the software to satisfy a given requirements, and how much time development will take. The cost of a project is a function of many parameters. Foremost among them is the size of the project. Other factors that affect the cost are programmer ability, experience of the developers in the area, complexity of the project, and reliability requirements. It is also due to the requirements of software, hardware and human resources.

Example:

Cost required for the project is to install the software and hardware requirements. Software may include installing SQL Server on the system. Cost due to the time taken for completion of the project which can be around 5 months. A Gantt chart given in the beginning helps to understand this in a better way.

7.1.3 Benefit Analysis

Example:

In the old system the work was done manually, so the cost of handling the system was increasing. The cost mainly includes the charges for registry maintenance, receipt books, files, etc. To reduce the costs the new system was proposed.

Positive aspects of the designed system which contributed to the benefit analysis are fast and easy storage of all information. It was also easy to retrieve any required details as fast as possible. There is no need for maintaining receipt books. The new system is very beneficial than the old one because the system is fully automated.

CHAPTER 8

System Maintenance

8.1 System Maintenance

- 1) There is no need of in depth knowledge of ASP.net and SQL. Basic knowledge of maintaining any software is required.
- 2) The system can be modified as and when required.
- 3) Proper training of how to handle the software is given.
- 4) The user password can be changed so as to protect the database, since database is viewed only in insurance agent section and not in guest section.
- 5) Validation is done properly and even if the user is not entering proper data it can be viewed and deleted from the database.
- 6) Comments are given in code section so that the insurance agent can understand what exactly the code does.
- 7) Backup of database should be maintained by writing the important data to CD on a daily or weekly basis.

The basic routine maintenance tasks are:

- Data backup
- Malware management
- File system maintenance

8.1.1. Backup

To backup is to create a redundant copy, so that if anything should happen to the original file, you have recourse to the backup. The process can be as simple as copying files to diskettes, but this soon becomes a problem where files are too big for diskette, where there are too many files, or where too many diskettes are required.

A better solution is to use an archiver (such as WinZip) or a backup utility to create a single compressed file from a collection of data files, and to split this over as many diskettes as required. This uses fewer diskettes and allows large files to be backed up even if the file is larger than a diskette can hold.

For large data sets, you may need to use a bulk storage medium such as tape, Zip disk, CDR or similar. These are generally faster and more reliable than diskettes.

In my system the reports can be exported anywhere and can be saved.

8.1.2. Malware Management

There's more on safe computing and malware. Malware includes viruses, worms, trojans, and increasingly invasive commercial applications, and management has several parts:

- Risk avoidance and evaluation - choice of applications and system setup
- Risk avoidance and evaluation - user education and safe computing practice
- Risk detection and destruction - choice and use of antivirus software
- Keeping abreast of malware - antivirus updates and ongoing user education

Simply running an antivirus utility is not enough, even if it is kept up to date.

For best performance, you can use on-demand rather than on-access antivirus scanners - but this requires the user to know when to use this, and act accordingly.

Updating an antivirus generally involves these steps:

- Go to antivirus vendor's web site via (say) Internet Explorer
- Navigate to the download section of the site
- Download any updates that are relevant, noting where these are saved
- Extract files from downloaded archive to the antivirus program directory

Some Windows-based antivirus utilities may automate this process to some extent, by accessing the Internet directly from within the program. You should check for updates at least once a week, and make sure your antivirus data files do not become more than a month out of date.

8.1.3. File System Maintenance

Much can be done during system setup to improve the survivability, maintainability and recoverability of the file system and its data, as discussed on the data management page. Thereafter, there are three tasks required on a regular basis:

- Check that sufficient free space is available; ideally 50M+ on C: volume
- Check the file system for errors, and manage these
- Defragment the file system once it is known to be error-free

The tools used here are Windows Explorer (or its "My Computer" incarnation), ScanDisk, and Defrag. If free space is low, you can clear .TMP files from the Windows base directory.

8.2 Limitations

- No new products can be added into the system. You can just increase and decrease the stock of the currently available product.
- Also the data to be stored may require data entry to be done.

CHAPTER 9

Annexure

DATA STRUCTURE

1) Admin database

Column Name	Data Type	Allow Nulls
d	varchar(50)	<input type="checkbox"/>
advocate_name	varchar(MAX)	<input type="checkbox"/>
address	varchar(MAX)	<input type="checkbox"/>
age	varchar(50)	<input type="checkbox"/>
E_mail_ID	varchar(MAX)	<input type="checkbox"/>
gender	varchar(50)	<input type="checkbox"/>
phone_no	varchar(50)	<input type="checkbox"/>
case_type	varchar(MAX)	<input type="checkbox"/>
qualification	varchar(MAX)	<input type="checkbox"/>

2) Appointment Database

Column Name	Data Type	Allow Nulls
app_id	varchar(50)	<input type="checkbox"/>
c_surname	varchar(MAX)	<input type="checkbox"/>
c_firstname	varchar(MAX)	<input type="checkbox"/>
c_midlename	varchar(MAX)	<input type="checkbox"/>
type	varchar(MAX)	<input type="checkbox"/>
subject	varchar(MAX)	<input type="checkbox"/>
remind_date	varchar(MAX)	<input type="checkbox"/>
remind_time	varchar(MAX)	<input type="checkbox"/>

3) Client Appointment

Column Name	Data Type	Allow Nulls
client_id	varchar(50)	<input type="checkbox"/>
c_surname	varchar(MAX)	<input type="checkbox"/>
c_firstname	varchar(MAX)	<input type="checkbox"/>
c_middlename	varchar(MAX)	<input type="checkbox"/>
dob_day	varchar(50)	<input type="checkbox"/>
dob_month	varchar(50)	<input type="checkbox"/>
dob_year	varchar(50)	<input type="checkbox"/>
sex	varchar(50)	<input type="checkbox"/>
Adress_line	varchar(MAX)	<input type="checkbox"/>
city	varchar(MAX)	<input type="checkbox"/>
state	varchar(MAX)	<input type="checkbox"/>
country	varchar(50)	<input type="checkbox"/>
phone_contact	varchar(50)	<input checked="" type="checkbox"/>
mobile_contact	varchar(50)	<input checked="" type="checkbox"/>
email_id	varchar(50)	<input type="checkbox"/>
date	varchar(50)	<input type="checkbox"/>

4) Employee Details

Column Name	Data Type	Allow Nulls
employee_id	varchar(50)	<input type="checkbox"/>
date	varchar(50)	<input type="checkbox"/>
e_surname	varchar(MAX)	<input type="checkbox"/>
e_firstname	varchar(MAX)	<input type="checkbox"/>
e_middlename	varchar(MAX)	<input type="checkbox"/>
e_day	varchar(50)	<input type="checkbox"/>
e_month	varchar(50)	<input type="checkbox"/>
e_year	varchar(50)	<input type="checkbox"/>
gender	varchar(50)	<input type="checkbox"/>
address	varchar(MAX)	<input type="checkbox"/>
state	varchar(MAX)	<input type="checkbox"/>
country	varchar(MAX)	<input type="checkbox"/>
phone_no	varchar(50)	<input type="checkbox"/>
qualification	varchar(MAX)	<input type="checkbox"/>
post	varchar(MAX)	<input type="checkbox"/>

5) Entry Details

Column Name	Data Type	Allow Nulls
case_id	varchar(50)	<input type="checkbox"/>
date	varchar(50)	<input type="checkbox"/>
client_id	varchar(50)	<input type="checkbox"/>
client_name	varchar(MAX)	<input type="checkbox"/>
case_type	varchar(MAX)	<input type="checkbox"/>
court_name	varchar(MAX)	<input type="checkbox"/>
opposite_cause	varchar(MAX)	<input type="checkbox"/>
stage_no	varchar(MAX)	<input type="checkbox"/>
next_date	varchar(MAX)	<input type="checkbox"/>
synopsis	varchar(MAX)	<input type="checkbox"/>

6) Installments Database

Column Name	Data Type	Allow Nulls
recpt_id	varchar(50)	<input type="checkbox"/>
client_id	varchar(50)	<input type="checkbox"/>
client_name	varchar(MAX)	<input type="checkbox"/>
amount_paid	varchar(50)	<input type="checkbox"/>
total_amount	varchar(50)	<input type="checkbox"/>
balance	varchar(50)	<input type="checkbox"/>
paymode	varchar(50)	<input type="checkbox"/>
check_no	varchar(50)	<input checked="" type="checkbox"/>
bank_name	varchar(MAX)	<input checked="" type="checkbox"/>
date	varchar(50)	<input type="checkbox"/>

7) Login Database

Column Name	Data Type	Allow Nulls
id	varchar(MAX)	<input type="checkbox"/>
designation	varchar(50)	<input type="checkbox"/>
username	varchar(50)	<input type="checkbox"/>
password	varchar(50)	<input type="checkbox"/>

8) Payment Database

Column Name	Data Type	Allow Nulls
receipt_id	varchar(50)	<input type="checkbox"/>
client_id	varchar(50)	<input type="checkbox"/>
client_name	varchar(MAX)	<input type="checkbox"/>
amount_paid	varchar(50)	<input type="checkbox"/>
total_amount	varchar(50)	<input type="checkbox"/>
balance	varchar(50)	<input type="checkbox"/>
paymode	varchar(50)	<input type="checkbox"/>
check_no	varchar(50)	<input checked="" type="checkbox"/>
bank_name	varchar(MAX)	<input checked="" type="checkbox"/>
date	varchar(50)	<input type="checkbox"/>

9) Payroll Database

Column Name	Data Type	Allow Nulls
e_id	varchar(50)	<input type="checkbox"/>
e_name	varchar(MAX)	<input type="checkbox"/>
time_period	varchar(50)	<input type="checkbox"/>
salary	varchar(50)	<input type="checkbox"/>
bonus	varchar(50)	<input checked="" type="checkbox"/>
total_salary	varchar(50)	<input type="checkbox"/>
date	varchar(50)	<input type="checkbox"/>

10) Proceeding Entry Database

Column Name	Data Type	Allow Nulls
case_id	varchar(50)	<input type="checkbox"/>
date	varchar(50)	<input type="checkbox"/>
client_id	varchar(50)	<input type="checkbox"/>
client_name	varchar(MAX)	<input type="checkbox"/>
case_type	varchar(MAX)	<input type="checkbox"/>
opposite_cause	varchar(MAX)	<input type="checkbox"/>
procceding_date	varchar(MAX)	<input type="checkbox"/>
procceding_summary	varchar(MAX)	<input type="checkbox"/>
next_hdate	varchar(50)	<input type="checkbox"/>
deposal_date	varchar(50)	<input type="checkbox"/>
result	varchar(MAX)	<input type="checkbox"/>

CHAPTER 10

Conclusion

&

Bibliography

Conclusion

- i. Automation is the big factor of our system. It reduces the work load of the administrator & employees of the system & also helps them to increase the profit margin of the Advocate Store system.
- ii. Thus, the '**ADVOCATE CASE DIARY**' is an effective & efficient system fulfilling the needs required by any customer.

10.1 List of abbreviations, Figures

10.1.1. Abbreviation

- frm: Form
- cbo: Combo Box
- btn : Button
- grp : Group Box
- pic : Picture Box
- txt : Text Box
- lbl : Label
- dtp : Date Time Picker
- mnu : Menu Item
- lst : List View

10.1.2. Figures

- Figure (2.1) Microsoft Visual Studio Details
- Figure(2.2)Gantt chart
- Figure (2.4) Symbol notation for the E-R Diagram
- Figure (2.5) E-R Diagram
- Figure (2.6) DFD Diagram
- Figure (2.7)context level Diagram
- Figure(2.8) use case Diagram
- Figure(2.9) Class Diagram
- Figure(2.10) Collaboration Diagram
- Figure(2.11) Activity Diagram
- Figure(5.1)Main Page
- Figure(5.2) Advocate Details
- Figure(5.3) Search Cases
- Figure(5.4) Appointment Entries Form
- Figure(5.5) Client Entry Form
- Figure(5.6)Case Entry Form
- Figure(5.7) Proceeding Case Entry Form
- Figure(5.8) Payment Form
- Figure(5.9) Installment Payment Form
- Figure(5.10) Update Client Details Form
- Figure(5.11) Update Case Information Form
- Figure(5.12) Search For Client Cases Form
- Figure(5.13) Employee Menu Bar
- Figure(5.14) Employee Details Form
- Figure(5.15) Employee Payment Form
- Figure(5.16)Employee Update Form
- Figure(5.17)Search and Delete details Form
- Figure(5.18)Report Menu Bar
- Figure(5.19)Case Report Form
- Figure(5.20)Payment Report
- Figure(5.21)Payroll Report
- Figure(5.22)Windows Menu Bar
- Figure(5.23)Create User Account Form
- Figure(5.24)Change Password Form
- Figure(5.25)Help Menu Bar
- Figure(5.26)About Application Form
- Figure(5.27)Validation For Completing Fields
- Figure(5.28)Client Entry Validation For Saved Records
- Figure(5.29)Validation For Numbers
- Figure(5.30)Validation
- Figure(5.31)Valid ID entry Message Box

10.2 Reference

10.2.1. Bibliography

- Mastering Visual Studio .NET , Jon Flanders, Ian Griffiths, Chris Sells,
Publisher: O'Reilly,
Edition: Third.
- Software Engineering, Roger Pressman,
Publisher: Gillies, Alan,
Edition: First.
- Database System Concepts, Silberschatz, Korth, Sudarshan,
Publisher: Fairley, Richard,
Edition: Third.
- The Complete Reference SQL ,Groff Weinberg,
Publisher: Grundgeiger, Dave,
Edition: Second.

10.2.2. Website Used

- <http://www.google.com> on 15/12/2013
- <http://www.sqldts.com> on 20/01/2014
- <http://www.support.microsoft.com> on 18/02/2014
- <http://www.msdn.microsoft.com> on 20/02/2014
- <http://en.wikipedia.org/wiki/diary> 27/02/2014