

A PROJECT REPORT ON

CRIME RECORDS MANAGEMENT SYSTEM

Submitted to *** University for the partial fulfillment of the
requirement for the**

Award of Degree for

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

Done by

Mr. /Miss. XXXXXX

XXXXX Institute of Management & Computer Sciences

Hyderabad

CERTIFICATE

This is to certify that Mr. XXXX, bearing Roll No. XXXXXXXXXXXX have developed Software project titled XXXXXXXX for **xxx SOFTWARE SOLUTIONS** as a partial Fulfillment for the award of the Degree of XXXXXXXX.

HEAD OF DEPARTMENT

PRINCIPAL
XXX institute of
Management &
Computer Sciences

EXTERNAL

ACKNOWLEDGMENT

My express thanks and gratitude and thanks to Almighty God, my parents and other family members and friends without whose uncontained support, I could not have made this career in XXXX.

I wish to place on my record my deep sense of gratitude to my project guide, **Mr. XXXXX, xxx Software Solutions, Hyderabad** for his constant motivation and valuable help through the project work. Express my gratitude to **Mr. XXXX**, Director of XXXXX **Institute of Management & Computer Sciences** for his valuable suggestions and advices through out the XXX course. I also extend my thanks to other Faculties for their Cooperation during my Course.

Finally I would like to thank my friends for their cooperation to complete this project.

XXXXXXX

ABSTRACT

ABSTRACT

The proposed system applies to all Police stations across the country and specifically looks into the subject of Crime Records Management. It is well understood that Crime Prevention, Detection and Conviction of criminals depend on a highly responsive backbone of Information Management. The efficiency of the police function and the effectiveness with which it tackles crime depend on what quality of information it can derive from its existing records and how fast it can have access to it.

It is proposed to centralize Information Management in Crime for the purposes of fast and efficient sharing of critical information across all Police Stations across the territory. Initially, the system will be implemented across Cities and Towns and later on, be interlinked so that a Police detective can access information across all records in the state thus helping speedy and successful completion to cases. The System would also be used to generate information for pro-active and preventive measures for fighting crime.

The project has been planned to be having the view of distributed architecture, with centralized storage of the database. The application for the storage of the data has been planned. Using the constructs of SQL server and all the user interfaces have been designed using the DOT Net technologies. The standards of security and data protective mechanism have been given a big choice for proper usage. The application takes care of different modules and their associated reports, which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff.

CONTENTS

1. Introduction

- 1.1 Introduction to Project
- 1.2 Organization Profile

2. System Analysis

- 2.1. Analysis Model
- 2.2. Existing System
- 2.3. Problem Statement
- 2.4. Proposed System

3. Software Requirement Specification

- 3.1. Product Overview
- 3.2. Hardware Requirements
- 3.3. Software Requirements
- 3.4. Performance Requirements

4. System Design

- 4.1. Introduction
- 4.2. Data flow Diagrams

5. Testing

6. Technical Notes

- 6.1. DOTNET Framework
- 6.2. C# Introduction and Overview

7. screens

8. Conclusion

9. Future Improvement

Introduction

1.1. Introduction to Project

Overview

The entire project has been developed keeping in view of the distributed client server computing technology, in mind. The specifications have been normalized up to 3NF to eliminate all the anomalies that may arise due to the database transaction that are executed by the general users and the organizational administration. The user interfaces are browser specific to give distributed accessibility for the overall system. At all proper levels high care was taken to check that the system manages the data consistency with proper business rules or validations. The authentication and authorization was crosschecked at all the relevant stages. The user level accessibility has been restricted into two zones namely. The administrative zone and the normal user zone.

Why new system?

- ☐ The system at any point of time can provide the details of the police station and the employees.
- ☐ The system at any point of time can provide the details of victims and the registered FIR's
- ☐ The system at any point of time can provide the details of evidence and their sequence
- ☐ The system at any point of time can provide the details of existing charge sheets and their statuses.

ANALYSIS

2.1. Analysis Model

The model that is basically being followed is the WATER FALL MODEL, which states that the phases are organized in a linear order. First of all the feasibility study is done. Once that part is over the requirement analysis and project planning begins. If system exists one and modification and addition of new module is needed, analysis of present system can be used as basic model.

The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model the sequence of activities performed in a software development project are: -

- Requirement Analysis
- Project Planning
- System design
- Detail design
- Coding
- Unit testing
- System integration & testing

Here the linear ordering of these activities is critical. End of the phase and the output of one phase is the input of other phase. The output of each phase is to be consistent with the overall requirement of the system. Some of the qualities of spiral model are also incorporated like after the people concerned with the project review completion of each of the phase the work done.

WATER FALL MODEL was being chosen because all requirements were known beforehand and the objective of our software development is the computerization/automation of an already existing manual working system.

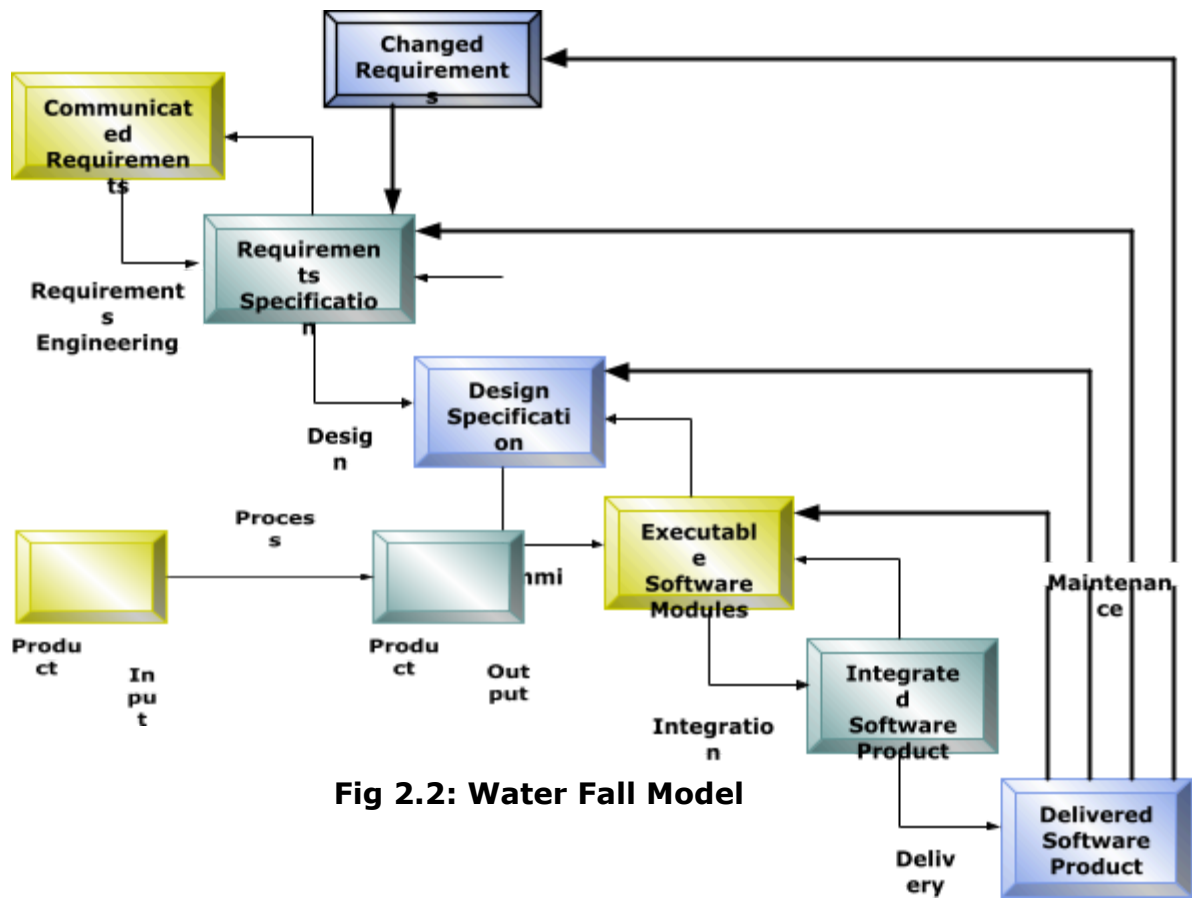


Fig 2.2: Water Fall Model

2.2 Existing System:

The existing system contains the about all the police stations that are registered as per the jurisdiction of the system. It also gets integrated with the employees who are working in these stations along with their designation.

2.3. Problem Statement:

The existing system doesn't have system security. That means, the user can login in to system any where in the world. But the data in this system is not for public. To avoid this problem, the proposed system is developed as MAC enabled website. That means, the user can access the website in that system only, so that we can avoid the information leakage problem.

2.4. Proposed System

The system after careful analysis has been identified to be presented with the following modules:

- **Police stations registration module:** This module maintains the information about all the police stations that are registered as per the jurisdiction of the system. It also gets integrated with the employees who are working in these stations along with their designation.
- **Victims FIR registration module:** This module maintains the information related to the first investigation report of the crime sequences that have taken place. The Fir registers all that a data that is necessary for the investigation to take place in proper length. It identifies the crime category and the crime nature.
- **Investigating evidence registration module:** This module makes a collection of information related to all the evidences that become categorically important under the normal sequence of the investigation, this module dynamically concentrates upon the changes that take place while the system of investigation is under process.

3. Software Requirement Specification

3.1. Overview

Purpose: The main purpose for preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and for determining the operating characteristics of the system.

Scope of the Development Project:

Database Tier: The concentration is applied by adopting the Oracle 9i Enterprise versions. SQL is taken as the standard query language. The overall business rules are designed by using the power of PL/SQL components like stored procedures stored functions and database triggers.

User Tier: The use interface is developed is a browses specific environment to have distributed architecture. The components are designed using HTML standards and Java server pages power the dynamic of the page design.

Developer Responsibilities Overview:

The developer is responsible for:

- Developing the system, which meets the SRS and solving all the requirements of the system?
- Demonstrating the system and installing the system at client's location after the acceptance testing is successful.
- Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
- Conducting any user training that might be needed for using the system.
- Maintaining the system for a period of one year after installation.

3.2. Hardware Requirements:

- PIV 2.8 GHz Processor and Above
- RAM 512MB and Above
- HDD 20 GB Hard Disk Space and Above

3.3. Software Requirements:

- WINDOWS OS (XP / 2000 / 200 Server / 2003 Server)
- Visual Studio .Net 2005 Enterprise Edition
- Internet Information Server 5.0 (IIS)

3.4. Performance Requirements:

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

4. SYSTEM DESIGN

4. System design

4.1 Introduction

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirements have been specified and analyzed, system design is the first of the three technical activities - design, code and test that is required to build and verify software.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

4.2 Data Flow Diagrams

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of

data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this lead to the modular design.

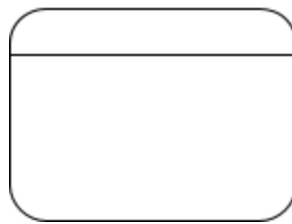
A DFD is also known as a "bubble Chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

4.2.1 Dfd Symbols:

In the DFD, there are four symbols

1. A square defines a source(originator) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows

3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data



Process that transforms data flow.



Source or Destination of data



Data flow



Data Store

4.2.2 Constructing a DFD:

Several rules of thumb are used in drawing DFD'S:

Process should be named and numbered for an easy reference. Each name should be representative of the process.

The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.

When a process is exploded into lower level details, they are numbered.

The names of data stores and destinations are written in capital letters.

Process and dataflow names have the first letter of each word capitalized.

A DFD typically shows the minimum contents of data store. Each data

store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out.

Missing interfaces redundancies and like is then accounted for often through interviews.

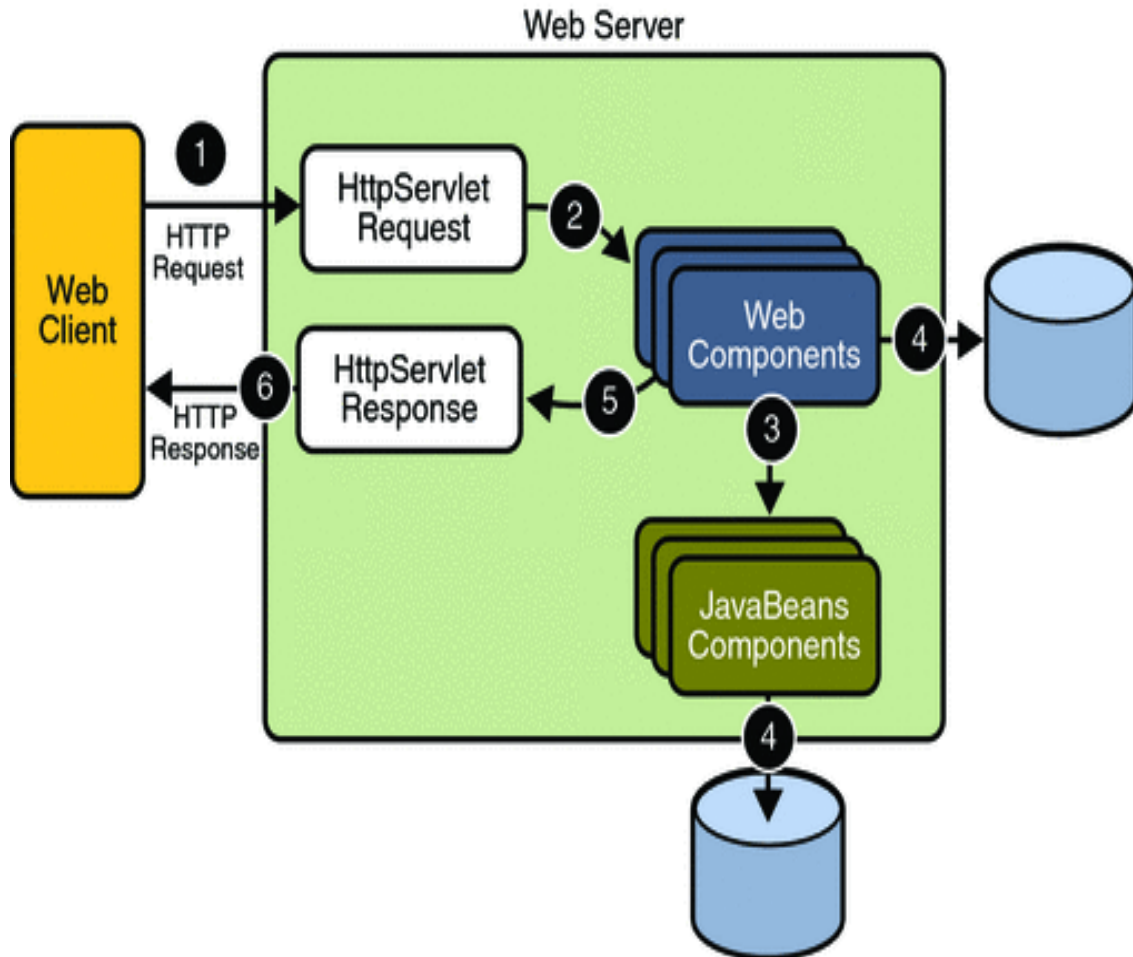
4.2.3 Silent Feature of DFD's

1. The DFD shows flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process whether the dataflow take place daily, weekly, monthly or yearly.
3. The sequence of events is not brought out on the DFD.

4.2.4 Data Flow:

- 1) A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The later is usually indicated however by two separate arrows since these happen at different type.
- 2) A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
- 3) A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produce some other data flow returns the original data into the beginning process.
- 4) A Data flow to a data store means update (delete or change).
- 5) A data Flow from a data store means retrieve or use. A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

Data Flow Diagram for our System



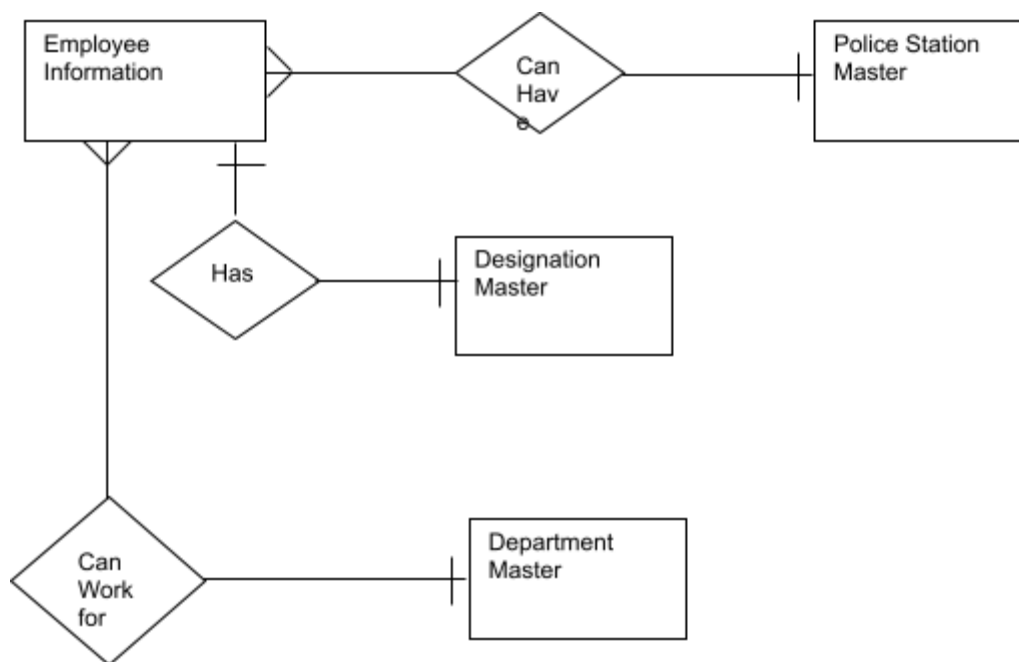
ER-Diagrams

- The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used

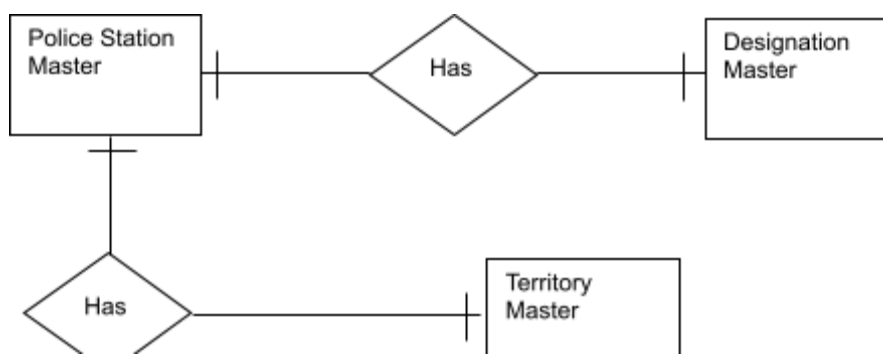
to conduct the data modeling activity the attributes of each data object noted in the ERD can be described using a data object description.

- The set of primary components that are identified by the ERD are
 - ◆ Data object
 - ◆ Relationships
 - ◆ Attributes
 - ◆ Various types of indicators.
- The primary purpose of the ERD is to represent data objects and their relationships.

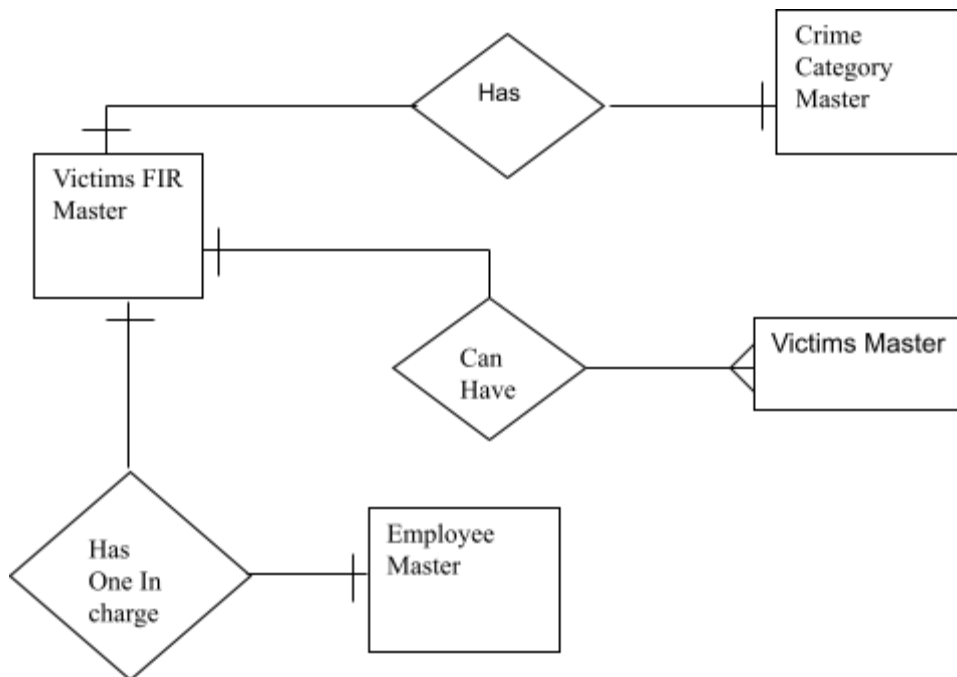
Employee Information



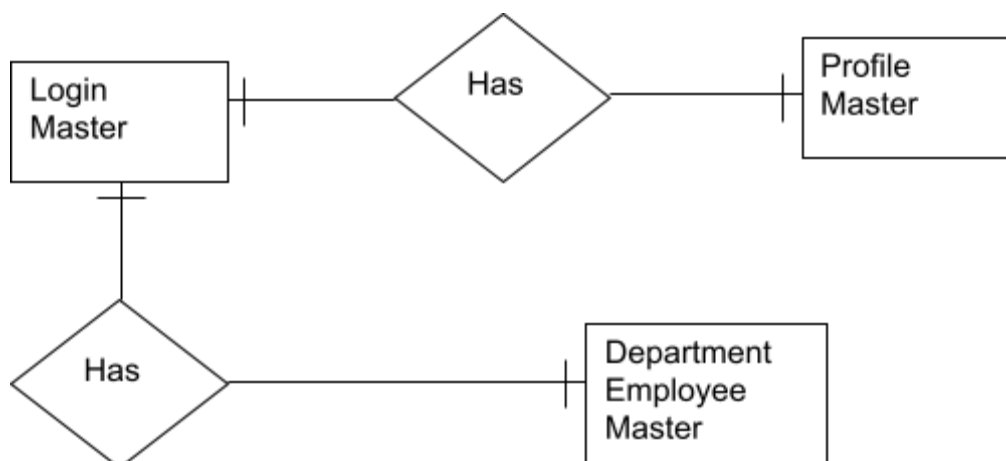
Police Station Master



Victims FIR Master



Login Master



Unified Modeling Language Diagrams

- The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.
- A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.
- **User Model View**
 - i. This view represents the system from the users perspective.
 - ii. The analysis representation describes a usage scenario from the end-users perspective.

Structural model view

- ◆ In this model the data and functionality are arrived from inside the system.
- ◆ This model view models the static structures

Behavioral Model View

- ◆ It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View

- ◆ In this the structural and behavioral as parts of the system are represented as they are to be built.

Environmental Model View

In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are

- ◆ UML Analysis modeling, which focuses on the user model and structural model views of the system.
- ◆ UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

Use Case Diagrams

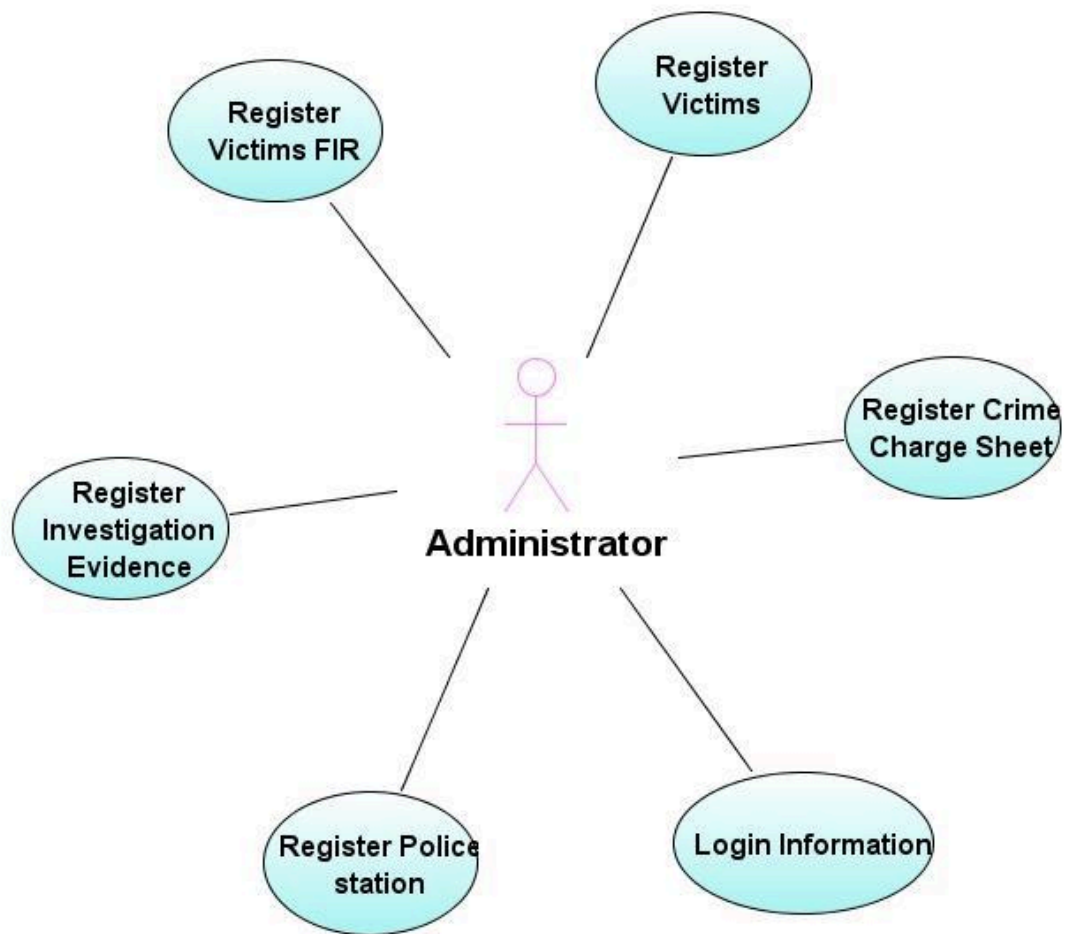
The actors who have been identified in the system are as follows:

1. **Investigating officer**
2. **Administrator**
3. **Writer**

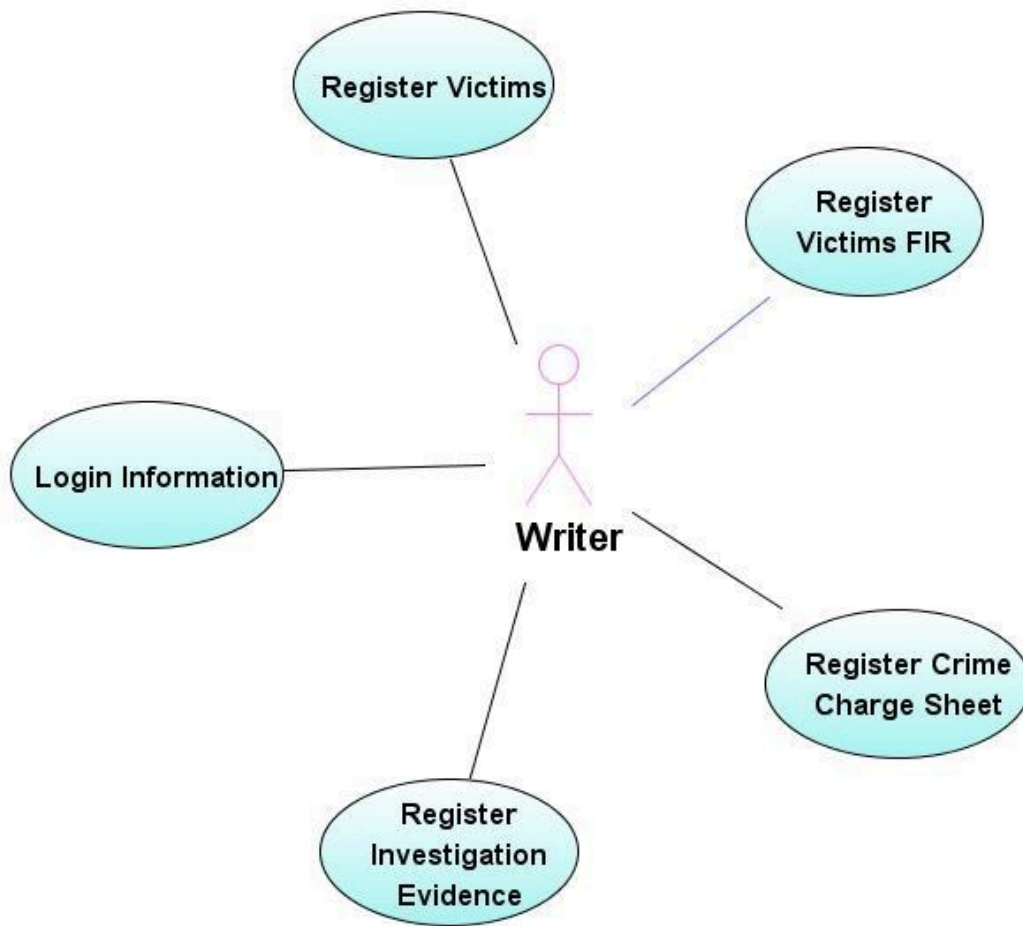
Investigating officer: He is the actor who can practically work upon the existing data in the police station only for view purpose.



Administrator: He is the actor who has the full-length potentiality and privilege to carry out transactions upon the system. He is authorized to maintain consistency within the information.



Writer: He is the actor who can enter all the details of the crime or evidence. Once entered cannot be edited. Only the administrator can edit or delete the record from the database.



Use case Description:

<i>Use case name</i>	Login Information
<i>Participating actors</i>	Administrator, Investigator, Writer
<i>Flow of events</i>	Provides username and password
<i>Entry Condition</i>	Users must know the username and password
<i>Exit condition</i>	User successfully logged into the system
<i>Quality Requirements</i>	Should provide proper error messages while login into the system.
<i>Use case name</i>	Register Victims
<i>Participating actors</i>	Administrator, Writer
<i>Flow of events</i>	User will enter the Victims information
<i>Entry Condition</i>	User should know the details of the victim
<i>Exit condition</i>	Victim details are successfully inserted into the system.
<i>Quality Requirements</i>	Display proper error messages while insertion.
<i>Use case name</i>	Register Victims FIR
<i>Participating actors</i>	Administrator, Writer

<i>Flow of events</i>	User will register the FIR
<i>Entry Condition</i>	User should know the details of the FIR
<i>Exit condition</i>	FIR details are successfully inserted into the system.
<i>Quality Requirements</i>	Display proper error messages while insertion.

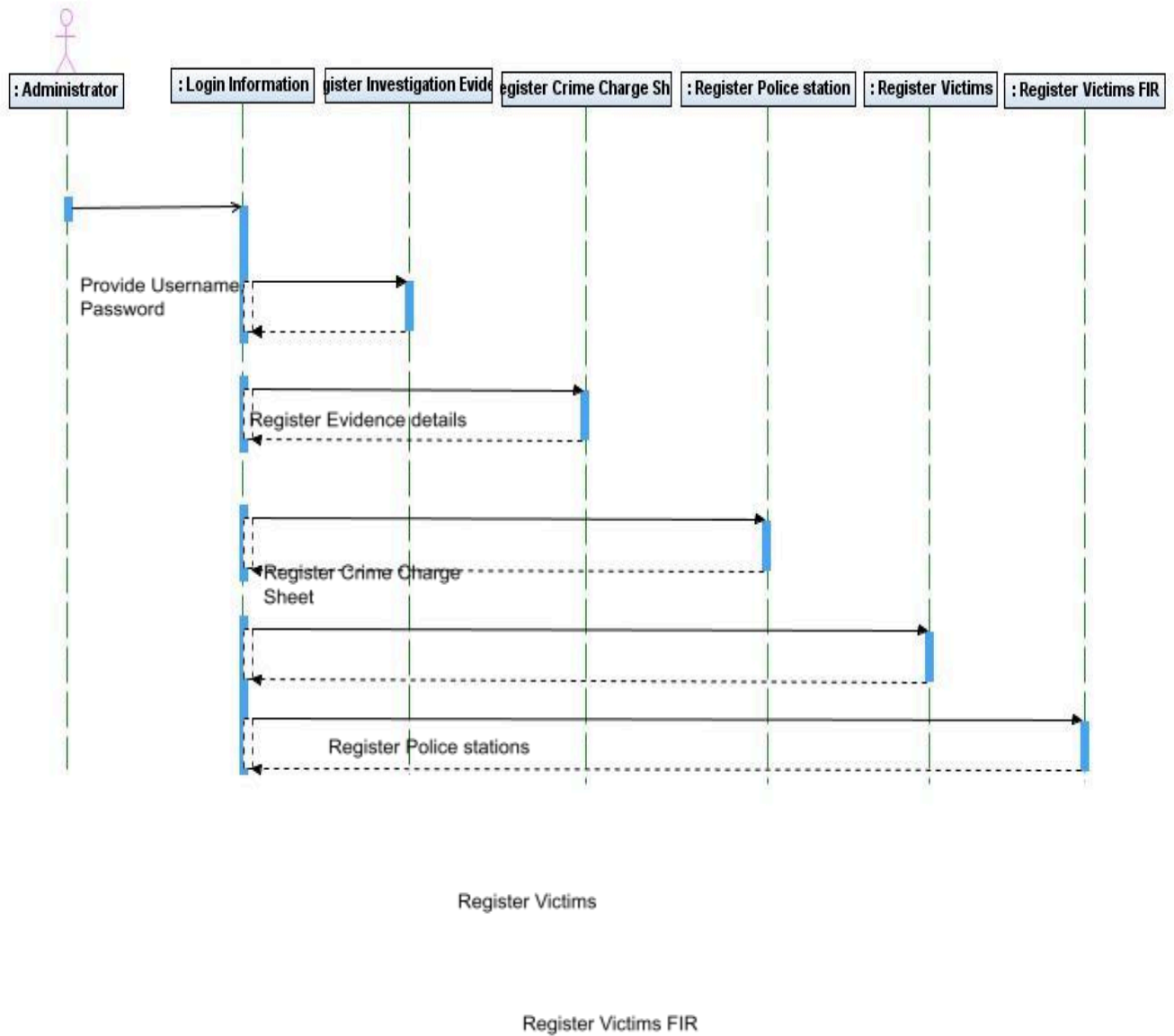
<i>Use case name</i>	Register Crime charge sheet
<i>Participating actors</i>	Administrator, Writer
<i>Flow of events</i>	User will register the crime charge sheet
<i>Entry Condition</i>	User should know the details of charge sheet.
<i>Exit condition</i>	Charge sheet details are successfully inserted into the system.
<i>Quality Requirements</i>	Display proper error messages while insertion.

<i>Use case name</i>	Register Investigation Evidence
<i>Participating actors</i>	Administrator, Writer
<i>Flow of events</i>	User will register the investigation evidence
<i>Entry Condition</i>	User should know the details of evidence.

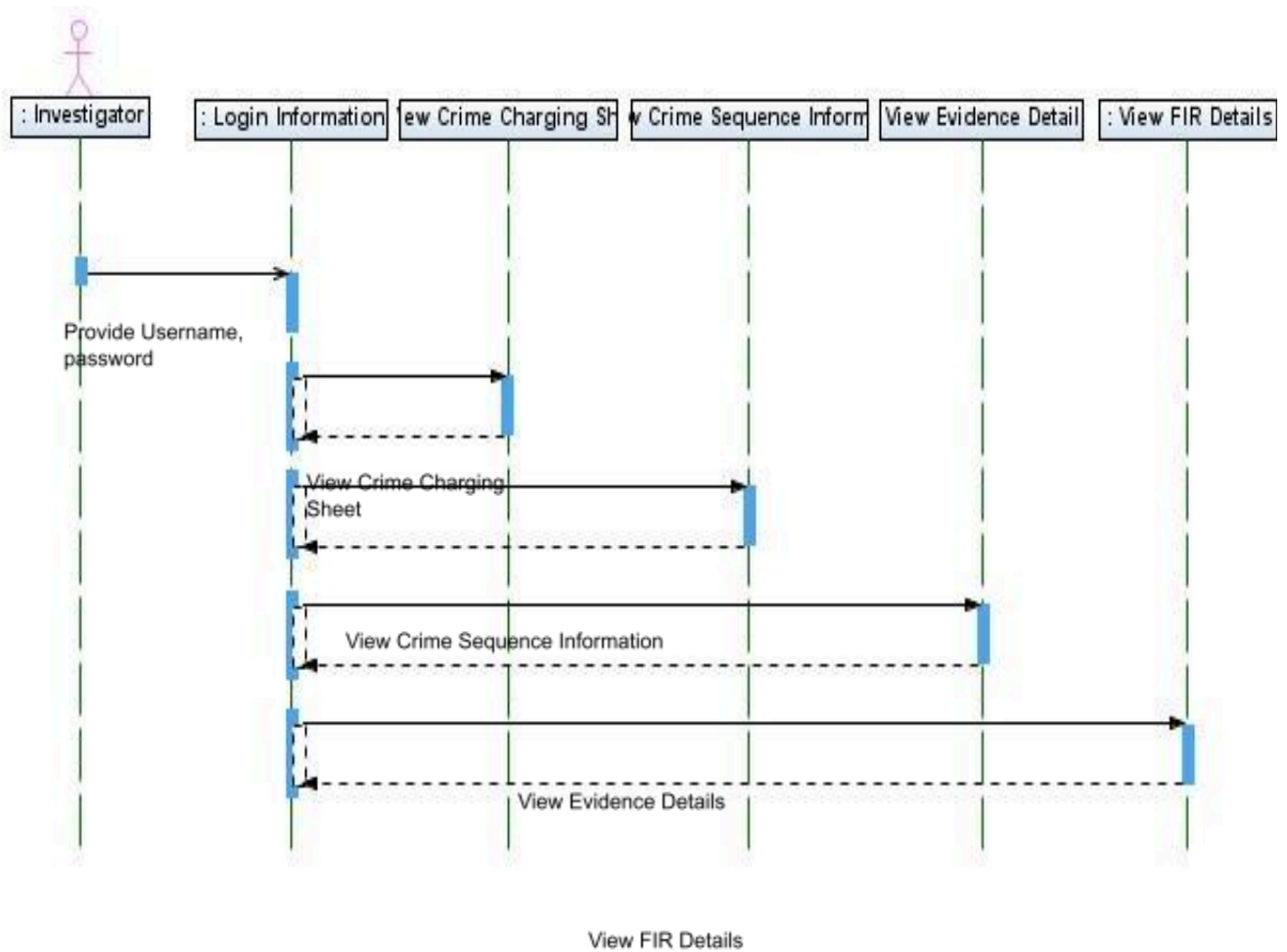
<i>Exit condition</i>	Evidence details are successfully inserted into the system.
<i>Quality Requirements</i>	Display proper error messages while insertion.
<i>Use case name</i>	Register Police Station
<i>Participating actors</i>	Administrator
<i>Flow of events</i>	User will register the police station.
<i>Entry Condition</i>	User should know the details of police station.
<i>Exit condition</i>	Police station details are successfully inserted into the system.
<i>Quality Requirements</i>	Display proper error messages while insertion.
<i>Use case name</i>	View all crime details
<i>Participating actors</i>	Investigator
<i>Flow of events</i>	User can view all the crime details.
<i>Entry Condition</i>	Display the details of crime and evidences.
<i>Exit condition</i>	Evidence and crime details are successfully displayed.
<i>Quality Requirements</i>	N/A

Sequence Diagram:

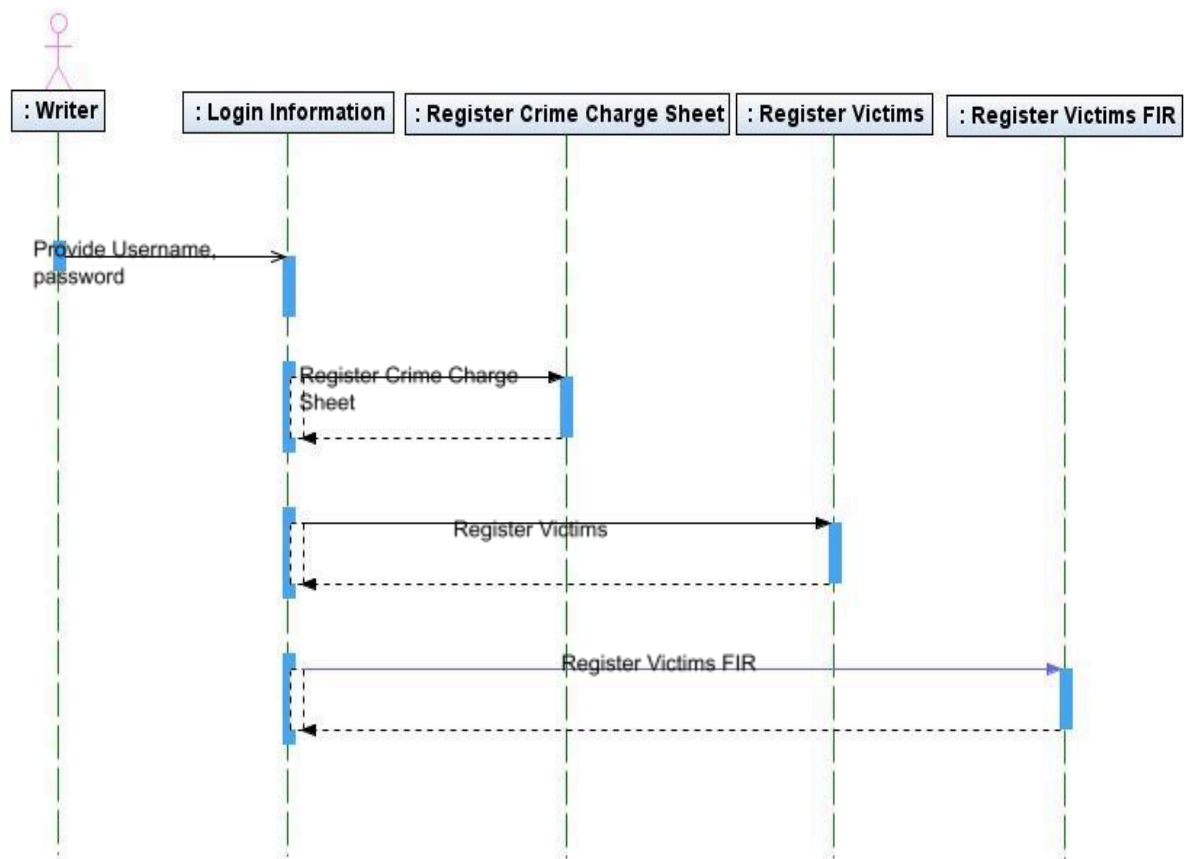
Administrator:



Investigator:



Writer:



TECHNICAL NOTES

5.1 Dot Net Framework:

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code.

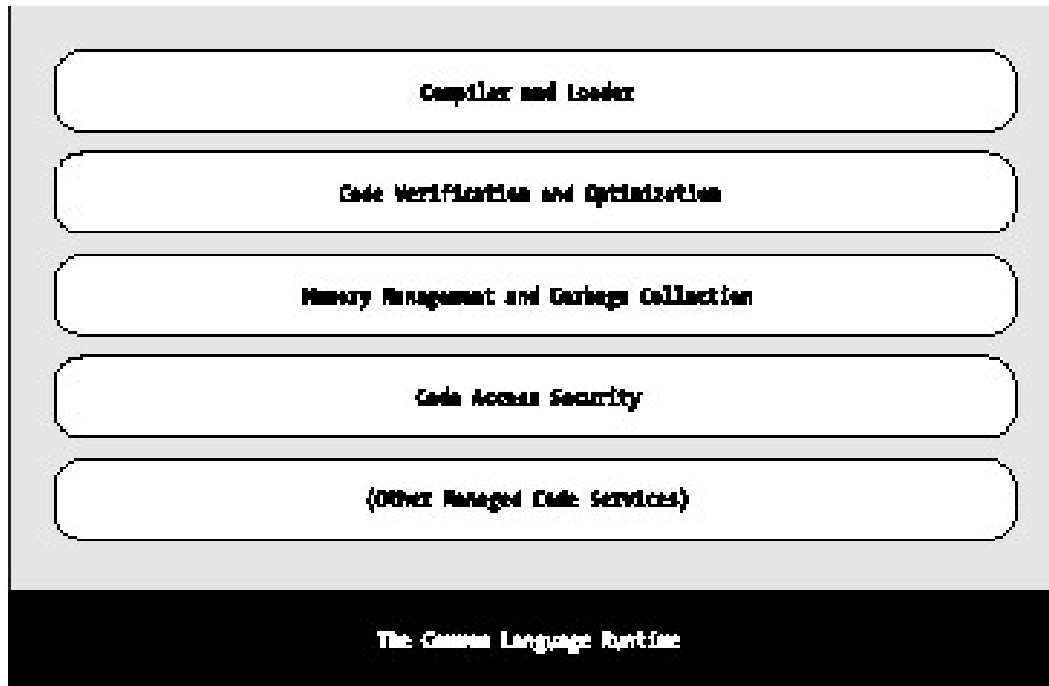
.NET Framework in context

The following sections describe the main components and features of the .NET Framework in greater detail.

Features of the Common Language Runtime

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

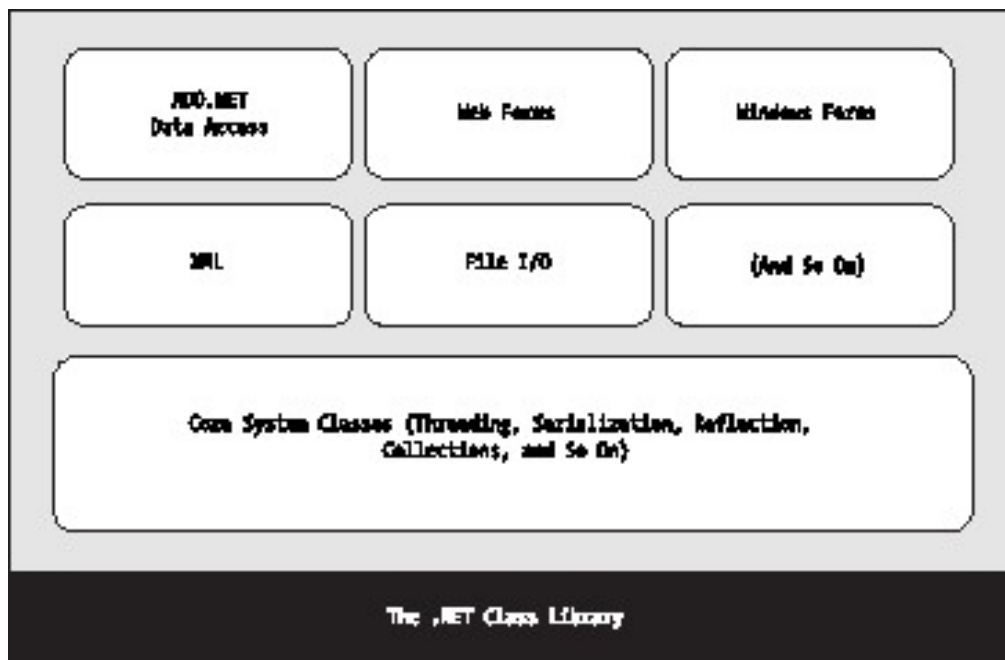


NET Framework Class Library

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework. For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework. As you would expect from an object-oriented class library, the .NET Framework types enable you to accomplish a range of common programming tasks, including tasks such as string management, data collection, database connectivity, and file access. In addition to these common tasks, the class library includes types that support a variety of

specialized development scenarios. For example, you can use the .NET Framework to develop the following types of applications and services:

- ☐ Console applications.
- ☐ WindowsGUIApplications (Windows Forms).
- ☐ ASP.NET applications.
- ☐ XML Web services.
- ☐ Windows services.



Client Application Development

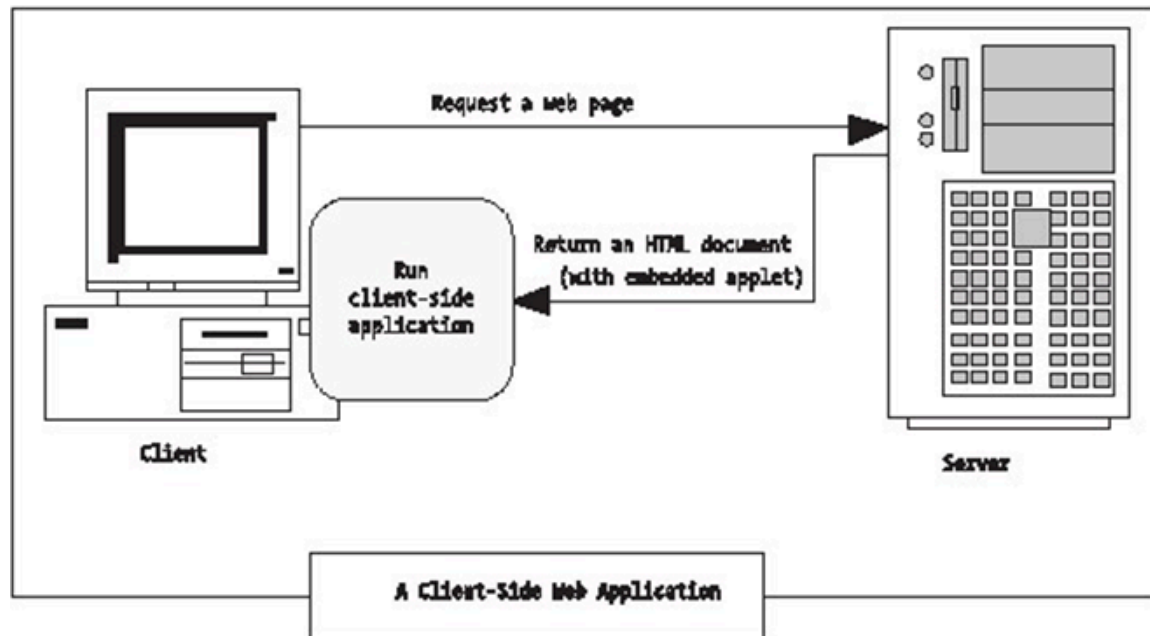
Client applications are the closest to a traditional style of application in Windows-based programming. These are the types of applications that display windows or forms on the desktop, enabling a user to perform a task. Client applications include applications such as word processors and spreadsheets, as well as custom business applications such as data-entry tools, reporting tools, and so on. Client applications usually employ windows, menus, buttons, and other GUI elements, and they likely

access local resources such as the file system and peripherals such as printers. Another kind of client application is the traditional ActiveX control (now replaced by the managed Windows Forms control) deployed over the Internet as a Web page. This application is much like other client applications: it is executed natively, has access to local resources, and includes graphical elements.

In the past, developers created such applications using C/C++ in conjunction with the Microsoft Foundation Classes (MFC) or with a rapid application development (RAD) environment such as Microsoft® Visual Basic®. The .NET Framework incorporates aspects of these existing products into a single, consistent development environment that drastically simplifies the development of client applications.

The Windows Forms classes contained in the .NET Framework are designed to be used for GUI development. You can easily create command windows, buttons, menus, toolbars, and other screen elements with the flexibility necessary to accommodate shifting business needs.

For example, the .NET Framework provides simple properties to adjust visual attributes associated with forms. In some cases the underlying operating system does not support changing these attributes directly, and in these cases the .NET Framework automatically recreates the forms. This is one of many ways in which the .NET Framework integrates the developer interface, making coding simpler and more consistent.

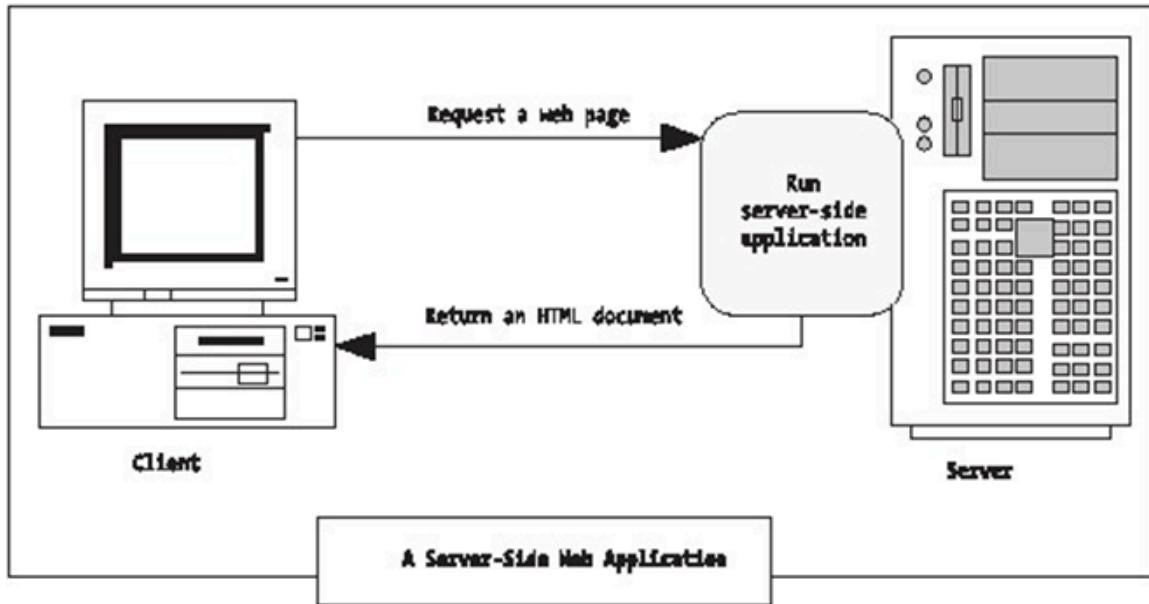


Server Application Development

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server.

This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server.

The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and SQL Server can perform standard operations while your application logic executes through the managed code.



Server-side managed code

ASP.NET is the hosting environment that enables developers to use the .NET Framework to target Web-based applications. However, ASP.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using managed code. Both Web Forms and XML Web services use IIS and ASP.NET as the publishing mechanism for applications, and both have a collection of supporting classes in the .NET

5.2. C# INTRODUCTION AND OVERVIEW

For the past two decades, C and C++ have been the most widely used languages for developing commercial and business software. While both languages provide the programmer with a tremendous amount of fine-grained control, this flexibility comes at a cost to productivity. Compared with a language such as Microsoft® Visual Basic®, equivalent C and C++ applications often take longer to develop. Due to the complexity and long cycle times associated with these languages, many C and C++ programmers have been searching for a language offering better balance between power and productivity.

There are languages today that raise productivity by sacrificing the flexibility that C and C++ programmers often require. Such solutions constrain the developer too much (for example, by omitting a mechanism for low-level code control) and provide least-common-denominator capabilities. They don't easily interoperate with preexisting systems, and they don't always mesh well with current Web programming practices.

The ideal solution for C and C++ programmers would be rapid development combined with the power to access all the functionality of the underlying platform. They want an environment that is completely in sync with emerging Web standards and one that provides easy integration with existing applications. Additionally, C and C++ developers would like the ability to code at a low level when and if the need arises.

Microsoft Introduces C#

The Microsoft solution to this problem is a language called C# (pronounced "C sharp"). C# is a modern, object-oriented language that enables programmers to quickly build a wide range of applications for the new Microsoft .NET platform, which provides tools and services that fully exploit both computing and communications. Because of its elegant object-oriented design, C# is a great choice for architecting a wide range of components-from high-level business objects to system-level applications. Using simple C# language constructs, these components can be converted into XML Web services, allowing them to be invoked across the Internet, from any language running on any operating system. More than anything else, C# is designed to bring rapid development to the C++ programmer without sacrificing the power and control that have been a hallmark of C and C++. Because of this heritage, C# has a high degree of fidelity with C and C++. Developers familiar with these languages can quickly become productive in C#.

Productivity and Safety

The new Web economy-where competitors are just one click away-is forcing businesses to respond to competitive threats faster than ever before. Developers are called upon to shorten cycle times and produce more incremental revisions of a program, rather than a single monumental version. C# is designed with these considerations in mind. The language is designed to help developers do more with fewer lines of code and fewer opportunities for error.

Embraces emerging Web programming standards

The new model for developing applications means more and more

solutions require the use of emerging Web standards like Hypertext Markup Language (HTML), Extensible Markup Language (XML), and Simple Object Access Protocol (SOAP). Existing development tools were developed before the Internet or when the Web as we know it today was in its infancy. As a result, they don't always provide the best fit for working with new Web technologies. C# programmers can leverage an extensive framework for building applications on the Microsoft .NET platform. C# includes built-in support to turn any component into an XML Web service that can be invoked over the Internet-from any application running on any platform. Even better, the XML Web services framework can make existing XML Web services look just like native C# objects to the programmer, thus allowing developers to leverage existing XML Web services with the object-oriented programming skills they already have. There are more subtle features that make C# a great Internet programming tool. For instance, XML is emerging as the standard way to pass structured data across the Internet. Such data sets are often very small. For improved performance, C# allows the XML data to be mapped directly into a struct data type instead of a class. This is a more efficient way to handle small amounts of data.

Eliminates costly programming errors

Even expert C++ programmers can make the simplest of mistakes-forgetting to initialize a variable, for instance-and often those simple mistakes result in unpredictable problems that can remain undiscovered for long periods of time. Once a program is in production use, it can be very costly to fix even the simplest programming errors.

The modern design of C# eliminates the most common C++ programming errors. For example:

- Garbage collection relieves the programmer of the burden of manual memory management.
- Variables in C# are automatically initialized by the environment.

- Variables are type-safe.

The end result is a language that makes it far easier for developers to write and maintain programs that solve complex business problems.

Better mapping between business process and implementation

With the high level of effort that corporations spend on business planning, it is imperative to have a close connection between the abstract business process and the actual software implementation. But most language tools don't have an easy way to link business logic with code.

For instance, developers probably use code comments today to identify which classes make up a particular abstract business object.

The C# language allows for typed, extensible metadata that can be applied to any object. A project architect can define domain-specific attributes and apply them to any language element-classes, interfaces, and so on. The developer then can programmatically examine the attributes on each element. This makes it easy, for example, to write an automated tool that will ensure that each class or interface is correctly identified as part of a particular abstract business object, or simply to create reports based on the domain-specific attributes of an object. The tight coupling between the custom metadata and the program code helps strengthen the connection between the intended program behavior and the actual implementation.

Extensive interoperability

The managed, type-safe environment is appropriate for most enterprise applications. But real-world experience shows that some applications continue to require "native" code, either for performance reasons or to interoperate with existing application programming interfaces (APIs). Such scenarios may force developers to use C++ even when they would prefer to use a more productive development environment.

C# addresses these problems by:

- Including native support for the Component Object Model (COM) and Windows.- based APIs.

2.3 ADO. Net:

ADO .NET stands for activeX data object.

Dataset is an in-memory content of the ADO. Net and is disconnected architecture of ADO. Net.

The connection between back end (database) and front end (dataset) is called as data adaptor. In back end creates a table with table name and column and required data types.

Main database used in. Net

- ✓ Oracle
- ✓ Sql server
- ✓ Ms access

To establish connection in Sql.

```
[SqlConnection cn = New SqlConnection (connection string);]
```

As

```
"Userid= sa;
```

```
Password=rvrjc;
```

```
Database=rentalsystem.mdf;
```

```
Server=local host."
```

To establish connection in oracle

```
[Oracle Connection cn = New Oracle (connection string);]
```

As

```
"User id= sa;  
Password=rvrjc;  
Database=rentalsyatem.mdf;  
Server=local host.";
```

Command object:**Syntax**

```
[Sql Command<obj> = New Sql Command ("Sql Statements", cn)]
```

Methods if command object in ADO. Net:

1. ExecuteNonQuery();
2. ExecuteScalar();
3. ExecuteReader();
4. ExecuteXmlReader();

To open connection the following method is used

```
[Cn. Open;]
```

```
Cn. Open ();
```

```
Sql Command;
```

```
Command Type;
```

```
Command<object> <method ()>;
```

Execute Non Query:

It is a method in ADO. Net used to perform Non-Query action.

Example insert, delete, update.

Execute Scaler:

It is a method in ADO. Net used to perform retrieve a single entity from the back end.

Example select status from products where productid = 1234.

Execute Reader:

It is a method in ADO. Net used to perform retrieve the data from back End.

Example Select *from EMP

Execute Xml Reader:

It is a method in ADO. Net used to perform retrieve the xml data in the .Net applications.

2.4 ASSEMBLY:

Logical collection of one or more .exe and .dll files is called as assembly and the compile source code of .Net architecture. There are two types of assemblies are categories.

- ☐ Single assemblies and,
- ☐ Multi file assemblies.

.EXE: Self execution files and not reusable.

.DLL: Not self execution files and reusable.

Assemblies of single file Content:

1. Manifest- metadata of assembly
2. Type metadata-data about assembly class
3. Msil-microsoft integrated language
4. Resource- external resource.

Microsoft integrated language is divided into four parts according to the versions:

- ☐ Version 6.0 major
- ☐ Version 3.0 minor
- ☐ Version 0.0 build
- ☐ Version 1.0 revision

Assemblies of multi file Content:

1. Manifest- metadata of assembly
2. Type metadata-data about assembly class
3. Msil-microsoft integrated language

These are myassembly.dll

1. Metadata
2. MSIL-Microsoft integrated language

Before Microsoft intermediate language (MSIL) can be executed, it must be converted by a .NET Framework just-in-time (JIT) compiler to native code, which is CPU-specific code that runs on the same computer architecture as the JIT compiler.

Rather than using time and memory to convert all the MSIL in a portable executable (PE) file to native code, it converts the MSIL as it is

needed during execution and stores the resulting native code so that it is accessible for subsequent calls.

The runtime supplies another mode of compilation called install-time_code generation. The install-time code generation mode converts MSIL to native code just as the regular JIT compiler does, but it converts larger units of code at a time, storing the resulting native code for use when the assembly is subsequently loaded and executed.

As part of compiling MSIL to native code, code must pass a verification process unless an administrator has established a security policy that allows code to bypass verification. Verification examines MSIL and metadata to find out whether the code can be determined to be type safe, which means that it is known to access only the memory locations it is authorized to access.

2.5 Validation controls and page navigation:

Validation controls:

Validation is the process of making sure that the user enters correct information into a form. Validation controls is provide by the. Net framework in the client browser if the browser supports java script and DHTML and checks the data that user entered before sending it to the server. The entire validation take place in the browser and nothing is sent back to the server. If the browser does not support DHTML and scripting then is done on the server. All validation controls in the. Net frameworks are derived from the Base Validator class.

The validation controls that are provided by the. Net frameworks are as following:

1. Required Field Validator
2. Compare Validator
3. Range Validator
4. Regular Expression Validator
5. Custom Validator

Required Field Validator:

Simple of all, Required Field Validator makes sure that the user enters data into a form. For example, on a registration form you might want your users to enter their data of in a textbox. If they leave field empty, this validation control will display an error.

Notable property of the Required Field Validator is the initial value property, which set as an initial value in the control

Compare Validator:

Comparison Validations are used to compare the entered by the user into a control (Textbox, for example) with the value entered into another control for with a constant value. We indicate the control to validate by setting the Control to validate property and if we want to compare a specific control with an other control we need set the Control to compare property to specify the Control to compare with.

Range Validator:

Range Validators are used to test if the value of a control is inside a specified range of values.

The three property of this control are the Control To Validate property, which contains the control of validate and maximum and minimum values which hold the maximum and minimum values of the valid range.

Regular Expression Validator:

Regular Expression Validators are used to check if the value in a control matches a pattern defined by the regular expression. Notable property for this control is the Validation Expression property, which allow us to select a predefined expression which we want to match with the data enter in a control.

Custom Validator:

Custom Validator is used to our own Validation for the data in a control. For example, you can check the value entered by a user is even or odd, with this control which is not possible with any of the above mentions validation controls. You write the script for the validation for JavaScript or

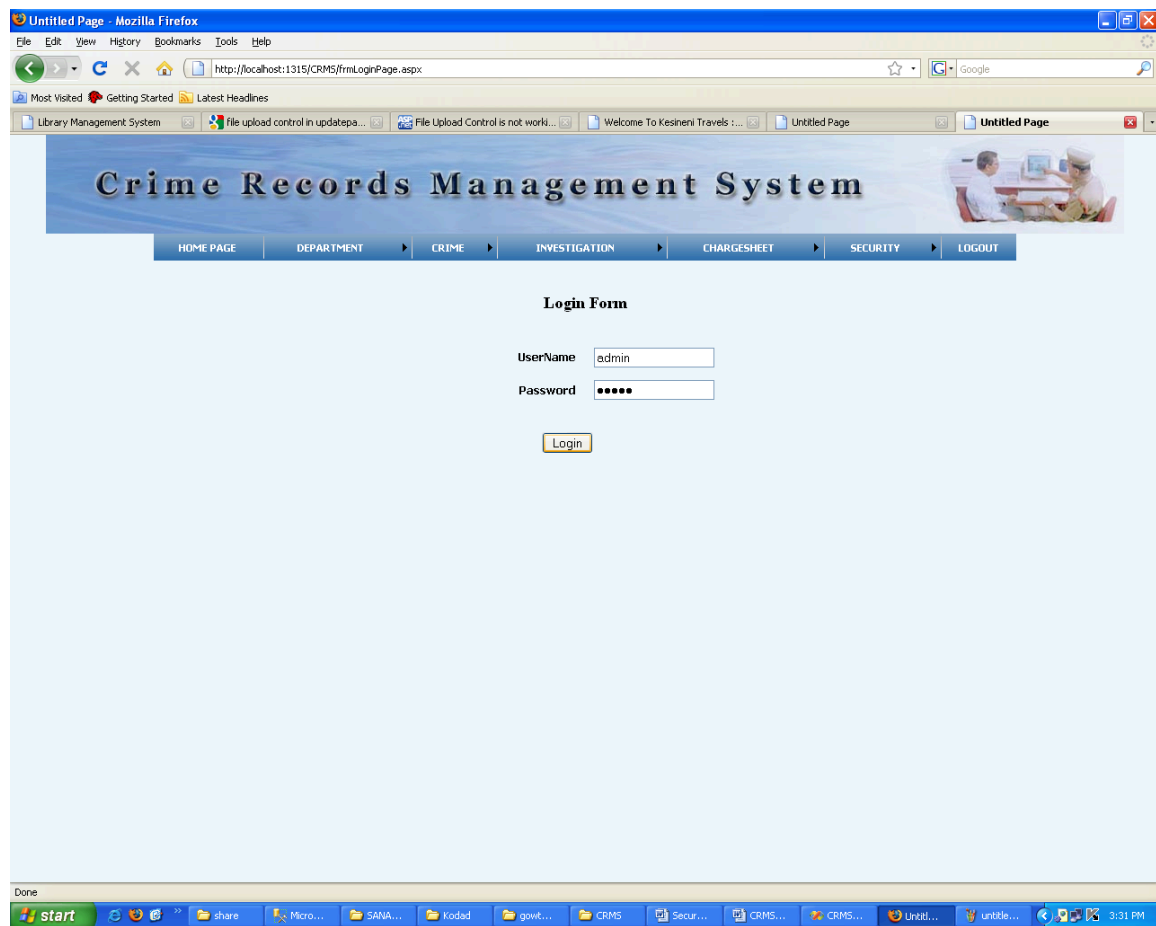
vbscript and associate that script function to the Client Validation Function property of this control

Output Screens

Home page Screen:



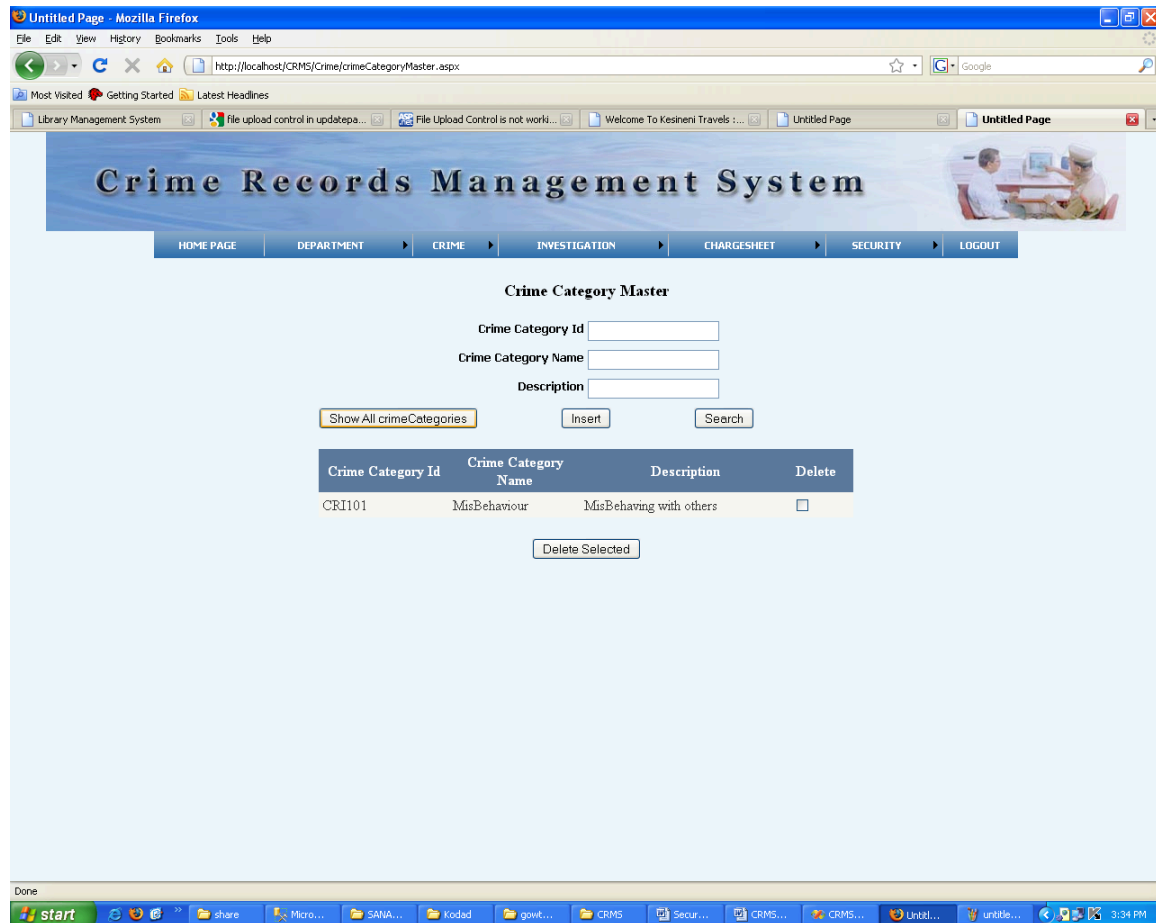
Login Page:



Designation page:



Crime category master:



Crime Charge Sheet Master:

Untitled Page - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/CRMS/ChargeSheet/frmCrimeChargeSheetMaster.aspx

Most Visited Getting Started Latest Headlines

Library Management System file upload control in updatepa... File Upload Control is not worki... Welcome To Kesinani Travels :... Untitled Page

Crime Records Management System

HOME PAGE DEPARTMENT CRIME INVESTIGATION CHARGESHEET SECURITY LOGOUT

Crime Charge Sheet Master

Charge Sheet No

Charge Sheet Date

Charge Sheet Details

Fir No

Produced Court No

Charge Sheet Filled By

Show All Insert Search

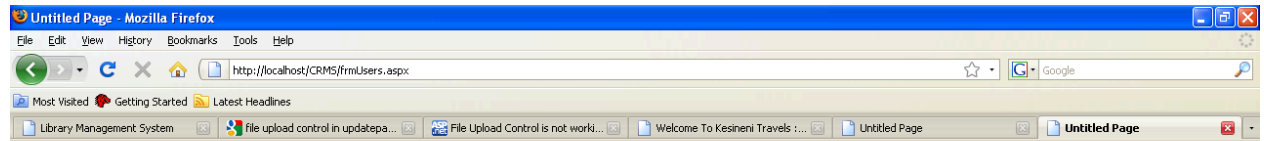
Charge Sheet No	Charge Sheet Date	Charge Sheet Details	FIR No	Produced Court Name	Charge Sheet Filed By	Delete
CHR102	2/6/2009	jdfdsj dfvbjksdb	INVFIR101	prd102	me	☐

Delete Selected

Done

start share Micro... SANA... Kodad gowit... CRMS Secur... CRMS... CRMS... Untitl... untitle... 3:34 PM

User details:



Crime Records Management System



[HOME PAGE](#) | [DEPARTMENT](#) | [CRIME](#) | [INVESTIGATION](#) | [CHARGESHEET](#) | [SECURITY](#) | [LOGOUT](#)

UserID	User Name	MAC Address	Select
1	admin		☐
6	gowtham	001CC011455C	☐
11	pradeep	001CC011455C	☐
1			

[Add](#) | [Edit](#) | [Delete](#)



User access denied page:



TESTING

6. Testing

1. The process of executing a system with the intent of finding an error.
2. Testing is defined as the process in which defects are identified, isolated, subjected for rectification and ensured that product is defect free in order to produce the quality product and hence customer satisfaction.
3. Quality is defined as justification of the requirements
4. Defect is nothing but deviation from the requirements
5. Defect is nothing but bug.
6. Testing --- The presence of bugs
7. Testing can demonstrate the presence of bugs, but not their absence
8. Debugging and Testing is not the same thing!
9. Testing is a systematic attempt to break a program or the AUT

Testing Methodologies:

- Black box Testing: is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application.

Usually Test Engineers are involved in the black box testing.

- White box Testing: is the testing process in which tester can perform testing on an application with having internal structural knowledge.

Usually The Developers are involved in white box testing.

- Gray Box Testing: is the process in which the combination of black box and white box tonics are used.

6.1 STLC (Software Testing Life Cycle)

Test Planning:

- Test Plan is defined as a strategic document which describes the procedure how to perform various testing on the total application in the most efficient way.
- This document involves the scope of testing,
- Objective of testing.
- Areas that need to be tested.
- Areas that should not be tested.
- Scheduling Resource Planning.

Types of Testing:

- Regression Testing: is one of the best and important testing. Regression testing is the process in which the functionality, which is already tested before, is once again tested whenever some new change is added in order to check whether the existing functionality remains same.
- Re-Testing: is the process in which testing is performed on some functionality which is already tested before to make sure that the defects are reproducible and to rule out the environments issues if at all any defects are there.
- Static Testing: is the testing, which is performed on an application when it is not been executed.ex: GUI, Document Testing
- Dynamic Testing: is the testing which is performed on an application when it is being executed.ex: Functional testing.
- Alpha Testing: it is a type of user acceptance testing, which is conducted on an application when it is just before released to the customer.
- Beta-Testing: it is a type of UAT that is conducted on an application when it is released to the customer, when deployed in to the real time environment and being accessed by the real time users.
- Installation Testing: it is the process of testing in which the tester try to install or try to deploy the module into the corresponding environment by following the guidelines produced in the deployment document and check whether the installation is successful or not.

Conclusion

Conclusions /Project Summary

The **Crime Records Managing System** is a web-based application for primarily providing training to the employees who provide customized solutions to meet organizational needs.

This application software has been computed successfully and was also tested successfully by taking "test cases". It is user friendly, and has required options, which can be utilized by the user to perform the desired operations.

The software is developed using Java as front end and Oracle as back end in Windows environment. The goals that are achieved by the software are:

- ✓ Instant access.
- ✓ Improved productivity.
- ✓ Optimum utilization of resources.
- ✓ Efficient management of records.
- ✓ Simplification of the operations.
- ✓ Less processing time and getting required information.
- ✓ User friendly.

FUTURE IMPROVEMENT

- This System being web-based and an undertaking of Cyber Security Division, needs to be thoroughly tested to find out any security gaps.
- A console for the data centre may be made available to allow the personnel to monitor on the sites which were cleared for hosting during a particular period.
- Moreover, it is just a beginning; further the system may be utilized in various other types of auditing operation viz. Network auditing or similar process/workflow based applications...

BIBLIOGRAPHY

- **FOR .NET INSTALLATION**

www.support.microsoft.com

- **FOR DEPLOYMENT AND PACKING ON SERVER**

www.developer.com

www.15seconds.com

- **FOR SQL**

www.w3schools.com

- **FOR ASP.NET**

[Asp.Net 3.5 Unleashed](#)

www.msdn.microsoft.com/net/quickstart/aspplus/default.com

www.asp.net

www.fmexpense.com/quickstart/aspplus/default.com