

### General Guidelines:

- ◆ Library books should be used with great care.
- ◆ Tearing, folding, cutting of library books or making any marks on them is not permitted and shall lead to disciplinary action.
- ◆ Any defect noticed at the time of borrowing books must be brought to the library staff immediately. Otherwise the borrower may be required to replace the book by a new copy.
- ◆ The loss of LRC book(s) must be immediately brought to the notice of the Librarian in writing.

**Learning Resource Centre-JUIT**



**SP03005**



# **RAILWAY MANAGEMENT SYSTEM**

**Submitted in partial fulfillment of the  
requirements for the award of the degree of**

**BACHELOR OF TECHNOLOGY  
in  
COMPUTER SCIENCE ENGINEERING**

**By**

**AMRIT PAL SINGH-031237  
ASHISH AGRAWAL-031407  
'NACHIKETA SINGH-031411**



**MAY-2007**

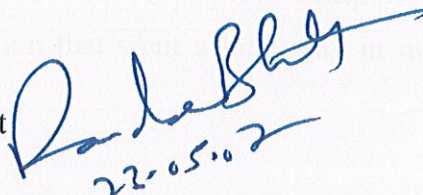
**DEPARTMENT OF COMPUTER SCIENCE  
JAYPEE UNIVERSITY OF INFORMATION  
TECHNOLOGY-WAKNAGHAT**



## CERTIFICATE

This is to certify that the work entitled, "RAILWAY MANAGEMENT SYSTEM" submitted by Amrit Pal Singh, Ashish Agrawal & Nachiketa Singh in partial fulfillment for the award of degree of Bachelor of Technology in INFORMATION TECHNOLOGY of Jaypee University of Information Technology has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Name of Supervisor: - Ravindra Bhatt



22-05-02



## ACKNOWLEDGMENT

First of all, we would like to thank our project supervisor Mr. Ravindra Bhatt, for his able guidance and support in the conception and development of this project. His suggestions in visualizing the project and sustained interest to attain the objective envisaged in the project are gratefully acknowledged.

A special thanks to Mr. S. P. Ghrera, HOD (Department of Computer Science) for his constant guiding vision and motivation that went a long way in materializing this project.

Amrit Pal

Amrit Pal  
(achish adnawal)

NSingh



## **TABLE OF CONTENTS**

### **1. INTRODUCTION**

1.1 History.....	11
1.2 Division of Indian Railways.....	14
1.3 Vision .....	16
1.4 Core Strengths.....	16

### **2. FEATURES OF PROJECT**

2.1 Real Time Updation.....	17
2.2 Multi- User.....	17
2.3 Multi- Hierarchy.....	17
2.4 Internet Connectivity.....	18
2.5 Static and Dynamic Nature.....	18

### **3. IMPLEMENTATION**

3.1 Microsoft SQL SERVER 2000.....	19
3.2 .NET.....	23
3.3 IIS.....	32
3.4 ASP.NET.....	33

### **4. ENTITIES / LIST OF TABLES**

4.1 Train Details.....	36
4.2 Fares.....	36
4.3 Source- Destination .....	37
4.4 Train Status.....	37



4.5	Train Schedule.....	37
4.6	Passenger Details.....	38
4.7	Universal Relations.....	38
4.8	Process Flow Diagram- "E-R Model".....	45
 <b>5. RESULT</b>		
5.1	Snap- Shots.....	48
 <b>6. CONCLUSION.....</b>		
		51
 <b>7. FUTURE SCOPE OF PROJECT.....</b>		
		52
 <b>8. REFERENCES.....</b>		
		53



## **LIST OF FIGURES**

### **Chapter 1:- INTRODUCTION**

- Figure 1.1 Railway Network of India.....13

### **Chapter 3: - IMPLEMENTATION**

- Figure 3.1 .NET Framework.....23
- Figure 3.2 Features of .NET.....24
- Figure 3.3 Easy Deployment of DLL HELD.....25
- Figure 3.4 Multitasking using .NET.....27
- Figure 3.5 Common Language Runtime 1.....27
- Figure 3.6 Common Language Runtime 2.....28
- Figure 3.7 Complete Pictured Working of .NET.....31
- Figure 3.8 IIS.....32
- Figure 3.9 ASP.NET Structure.....35
- Figure 3.10 How IIS handles on ASP file request.....36

### **Chapter 4: - ENTITIES**

- Figure 4.1 E-R MODEL.....45

### **Chapter 5: - RESULT**

- Figure 5.1 Home Page.....48
- Figure 5.2 Booking Ticket Page.....49
- Figure 5.3 Administrator Window.....50



## LIST OF ABBREVIATIONS

• ADO-	ActiveX Data Object
• API-	Application Protocol Interface
• ASP-	Active Server Pages
• BCNF-	Boyce Codd Normal Form
• CLR-	Common Language Run-Time
• DFD-	Data Flow Diagram
• DLL-	Dynamically Load Library
• ER-	Entity Relation
• FD-	Functional Dependencies
• HTML-	Hyper Text Markup Language
• HTTP-	Hyper Text transfer Protocol
• ID-	Identification
• IL-	Intermediate Language
• IIS-	Internet Information Services
• IP-	Internet Protocol
• IR-	Indian Railways
• JIT-	Just In Time
• JVM-	JAVA Virtual Machine
• NF-	Normal Form
• OS-	Operating System
• PFD-	Partial Functional Dependencies
• RDBMS-	Railway Database Management System
• VB-	Visual Basic
• VS-	Visual Studio
• WWW-	World Wide Web
• XML-	Extensive Markup Language



## **LIST OF TABLES**

### **Chapter 1:- INTRODUCTION**

- Table 1.1 Zones of Indian Railways.....14
- Table 1.2 Divisional Head Quarters.....15

### **Chapter 4: - ENTITIES / LIST OF TABLES**

- Table 4.1 Train Details.....36
- Table 4.2 Fares.....36
- Table 4.3 Source- Destination.....37
- Table 4.4 Train Status.....37
- Table 4.5 Train Schedule.....37
- Table 4.6 Passenger Details.....38
- Table 4.7 Functional Dependencies.....38
- Table 4.8 A.) 2<sup>nd</sup> Normal Form of P.F.D 1.....42
- Table 4.8 B.) 2<sup>nd</sup> Normal Form of P.F.D 2.....42
- Table 4.8 C.) 2<sup>nd</sup> Normal Form of P.F.D 3.....43
- Table 4.8 D.) 2<sup>nd</sup> Normal Form of P.F.D 4.....43
- Table 4.9 BCNF.....44



## **ABSTRACT**

This project uses ASP.Net as Front End and Microsoft SQL Server 2000 as Database (Back End). Our application is an ONLINE RAILWAY MANAGEMENT SYSTEM developed in ASP.NET to facilitate online Booking of Ticket, Enquiry related to Indian Railways etc..The site has been embellished with various CASCADED STYLE SHEETS (CSS).The various scripting languages like JAVA Script, VB Script have also been used. The portal provides the user with an online account which they can personalize. Once they have created their account they can log onto our website and search/surf. The portal provides the facility for online trade. The website provides secure, safe, reliable, convenient money transactions through cash, credit cards. The website also provides the Companies with facilities for marketing and advertisement of their products. It can be used as a platform by companies for selling their products. The companies can become the member by creating their account for payment purposes. The companies should have good marketing credibility and third party secure transaction system. The Website has been divided into various modules. A brief description of the various features incorporated in the project is covered in Chapter 2.



## **1. INTRODUCTION**

The railway enquiry system is a classical database management example. An obvious reason for its popularity in the computer fraternity is that railway itself is a fine example of human engineering. In the discussion on database classification, creation and population that follows, the ideas have been strongly typed to the domain of railway-enquiry.

### **1.1 HISTORY**

**Indian Railways**, abbreviated as **IR**, is a Department of Government of India, under Ministry of Railways and is tasked with operating the rail network in India. The Ministry is headed by a cabinet rank Railways Minister, whereas, the department is managed by Railway Board. Indian Railways is NOT a corporate body, as many people think. However, of late, IR is trying to adopt a corporate management style. Indian Railways had, until recently, a monopoly on the country's rail transport. It is one of the largest and busiest rail networks in the world, transporting just over six billion passengers and almost 750 million tonnes of freight annually. IR is the world's largest commercial or utility employer, with more than 1.6 million employees, and is second only in total terms to the Chinese Army the railways traverse through the length and width of the country; the routes cover a total length of 63,140 km (39,462 miles). As of 2002, IR owned a total of 216,717 wagons, 39,236 coaches and 7,739 locomotives and ran a total of 14,444 trains daily, including about 8,702 passenger trains. Today the Indian Railways transports 5 billion passengers each year. Railways were first introduced to India in 1853. By 1947, the year of



India's independence, there were forty-two rail systems. In 1951 the systems were nationalized as one unit, becoming one of the largest networks in the world. Indian Railways operates both long distance and suburban rail systems.

Indian Railways operates 8,702 passenger trains and transports around five billion annually across twenty-six states and three union territories (Delhi, Puducherry (formerly Pondicherry) and Chandigarh). Sikkim and Meghalaya are the only states not connected.

The passenger division is the most preferred form of long distance transport in most of the country. A standard passenger train consists of eighteen coaches, but some popular trains can have up to 24 coaches. Coaches are designed to accommodate anywhere from 18 to 72 passengers, but may actually accommodate many more during the holiday seasons and on busy routes. The coaches in use are vestibules, but some of these may be dummied on some trains for operational reasons. Freight trains use a large variety of wagons. Each coach has different accommodation class; the most popular being the sleeper class. Up to nine of these type coaches are usually coupled. Air conditioned coaches are also attached, and a standard train may have between three and five air-conditioned coaches. Online passenger ticketing, introduced in 2004, is expected to top 100,000 per day by 2009.



## RAILWAY NETWORK OF INDIA

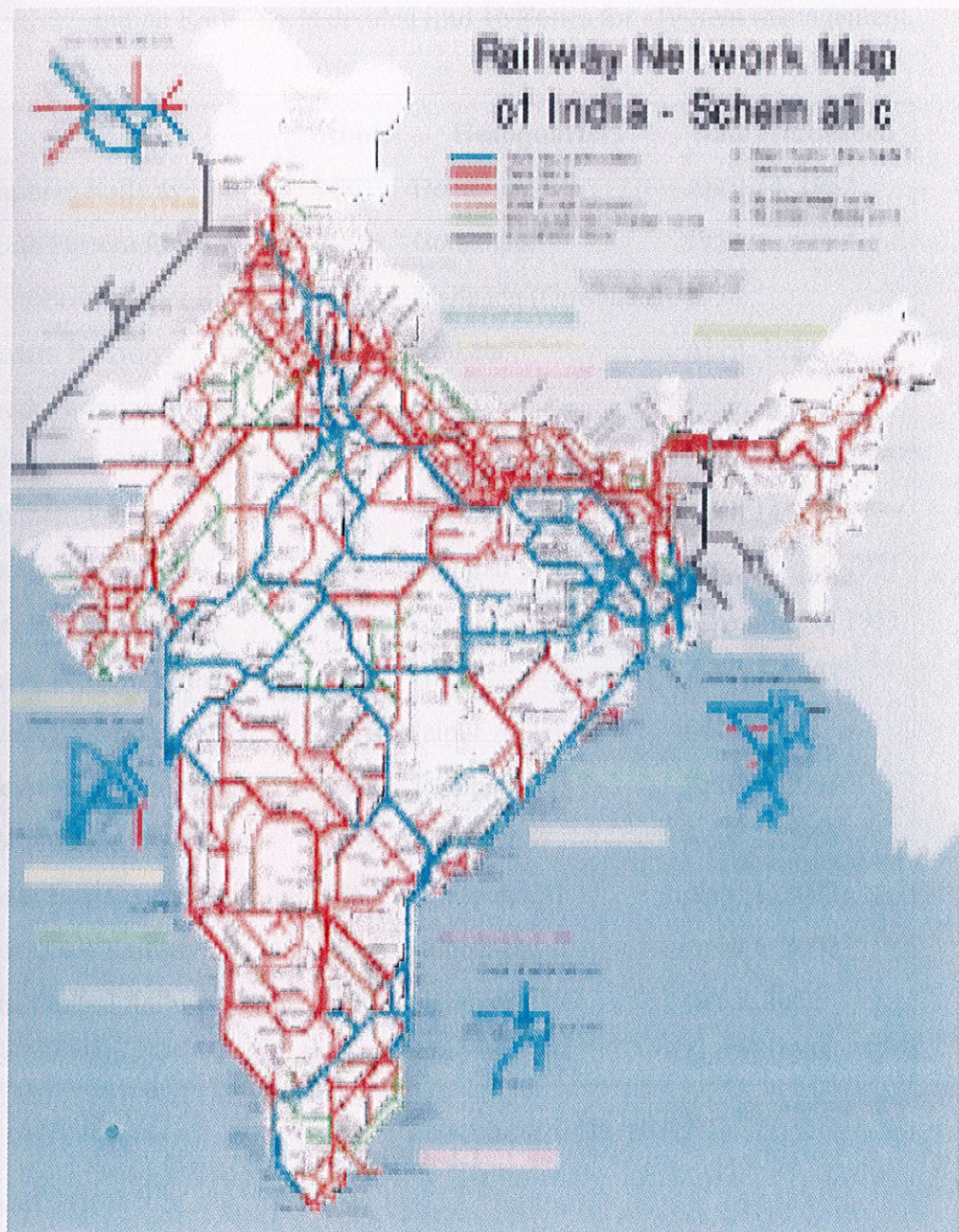


Figure No. 1.1



## 1.2 DIVISION OF INDIAN RAILWAYS

Indian Railways is divided into 16 Zones for efficient management.

No.	Name	Abbr.	Headquarters	Date established
1.	Northern Railway	NR	Delhi	April 14, 1952
2.	North Eastern Railway	NER	Gorakhpur	1952
3.	Northeast Frontier Railway	NFR	Maligaon (Guwahati)	1958
4.	Eastern Railway	ER	Kolkata	April, 1952
5.	South Eastern Railway	SER	Kolkata	1955,
6.	South Central Railway	SCR	Secunderabad	October 2, 1966
7.	Southern Railway	SR	Chennai	April 14, 1951
8.	Central Railway	CR	Mumbai	November 5, 1951
9.	Western Railway	WR	Mumbai	November 5, 1951
10.	South Western Railway	SWR	Hubli	April 1, 2003
11.	North Western Railway	NWR	Jaipur	Oct 1, 2002
12.	West Central Railway	WCR	Jabalpur	April 1, 2003
13.	North Central Railway	NCR	Allahabad	April 1, 2003
14.	South East Central Railway	SECR	Bilaspur, CG	April 1, 2003
15.	East Coast Railway	ECoR	Bhubaneswar	April 1, 2003
16.	East Central Railway	ECR	Hajipur	Oct 1, 2002
17.	Konkan Railway†	KR	Navi Mumbai	Jan 26, 1998

**Table No. 1.1**



Each zonal railway is made up of a certain number of divisions, each having a divisional headquarters. There are a total of sixty-seven divisions.

<b>Zonal Railway</b>	<b>Divisions</b>
Northern Railway	Delhi, Ambala, Firozpur, Lucknow, Moradabad
North Eastern Railway	Izzatnagar, Lucknow, Varanasi
Northeast Frontier Railway	Alipurduar, Katihar, Lumding, Rangia, Tinsukia
Eastern Railway	Howrah, Sealdah, Asansol, Malda
South Eastern Railway	Adra, Chakradharpur, Kharagpur, Ranchi
South Central Railway	Secunderabad, Hyderabad, Guntakal, Guntur, Nanded, Vijayawada
Southern Railway	Chennai, Madurai, Palghat, Tiruchchirapalli, Trivandrum, Salem
Central Railway	Mumbai, Bhusawal, Pune, Solapur, Nagpur
Western Railway	Mumbai Central, Baroda, Ratlam, Ahmedabad, Rajkot, Bhavnagar
South Western Railway	Hubli, Bangalore, Mysore
North Western Railway	Jaipur, Ajmer, Bikaner, Jodhpur
West Central Railway	Jabalpur, Bhopal, Kota
North Central Railway	Allahabad, Agra, Jhansi
South East Central Railway	Bilaspur, Raipur, Nagpur
East Coast Railway	Khurda Road, Sambalpur, Visakhapatnam
East Central Railway	Danapur, Dhanbad, Mughalsarai, Samastipur, Sonpur

**Table No. 1.2**



### **1.3 VISION**

- To be the most preferred and fastest growing IT services provider in the field of website development and various other IT related services.
- To raise the bar on service levels, with organization-wide commitment to building centers of excellence.

### **1.4 CORE STRENGTHS**

- Entrepreneurial Ability
- Serving single handedly to the requirements of users.



## **2. FEATURES OF PROJECT**

The typical railway reservation system has the following features:

### **2.1 REAL TIME UPDATION**

Owing to the nature of its purpose, the RDBMS system is a real time DBMS system. The system never goes offline and a complete mirroring of the constant changes should reflect in the system. For example, the act of cancellation of any train should reflect in the RDBMS.

### **2.2 MULTI-USER**

The RDBMS supports the multi-user functionality. The same system is used in all the enquiry counters at different stations. For example two people with same destination and source locations may make two very different queries at the same time because of their diverse personal requirements.

### **2.3 MULTI-HIERARCHY**

The rights and permissions of the various end-users (again railway employees) vary depending upon the dimension of railway-enquiry, they are taking care of. For example, an employee handling a normal enquiry counter shall have no access and rights to query on the database of goods trains. even when their information is very similar to the one on passenger trains.



## **2.4 INTERNET-CONNECTIVITY**

A recent addition to the RDBMS stable is the support for Internet functionality. With a user-friendly interface, the railway enquiry system is now available to end users (passengers) through the Internet. This provides another dimension of complexity to the RDBMS system with access to the database provided to “machines”.

## **2.5 STATIC AND DYNAMIC NATURE**

The RDBMS is a healthy mix of both static and dynamic relations. Few relations are almost static in nature; like the “trains-running” sub-schema which change only rarely. On the other hand a sub-schema like “scheduled platform number and scheduled arrival/departure for a train” changes from minute to minute.



### **3. IMPLEMENTATION**

#### **3.1 MICROSOFT SQL SERVER 2000**

In this Project we used Relational Databases to maintain the record of Train, Passengers, Reserved Tickets, Stations etc. all related information to Indian Railways. SQL SERVER 2000 helps us to create, access and to interact with those relational Databases in a faster, easier and efficient manner. Some of the basic commands we used are covered below.

MICROSOFT SQL SERVER 2000 deals with

- A) Data types
- B) Primary Key, Foreign Key
- C) Relational Database
- D) SQL Queries

Microsoft® SQL Server™ 2000 includes powerful features to support international operations and environments. Extensive distinguishing features make SQL Server 2000 a compelling database product and applications platform.

#### **A) Data types**

An attribute that specifies what type of information can be stored in a column, parameter, or variable. System-supplied data types are provided by SQL Server; user defined data types can also be created.

E.g. char, varchar, text, nchar, nvarchar, ntext, datetime



## **B) Primary Key, Foreign Key**

### **Primary Key:**

A column or set of columns that uniquely identify all the rows in a table. Primary keys do not allow null values. No two rows can have the same primary key value therefore; a primary key value always uniquely identifies a single row. More than one key can uniquely identify rows in a table, each of these keys is called a candidate key. Only one candidate can be chosen as the primary key of a table; all other candidate keys are known as alternate keys. Although tables are not required to have primary keys, it is good practice to define them. In a normalized table, all of the data values in each row are fully dependent on the primary key. For example, in a normalized employee table that has EmployeeID as the primary key, all of the columns should contain data related to a specific employee.

### **Foreign Key:**

The column or combination of columns whose values match the primary key (PK) or unique key in the same or another table. Also called the referencing key.

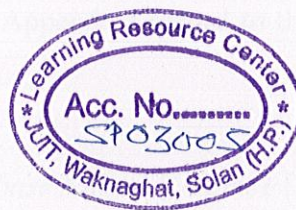
## **C) Relational Database:**

A system that organizes data into related rows and columns. SQL Server is a relational database management system (RDBMS).



**Relationship:** A link between tables that references the primary key in one table to a foreign key in another table. The relationship line is represented in a database diagram by a solid line if referential integrity between the tables is enforced, or a dashed line if referential integrity is not enforced for INSERT and UPDATE transactions. The endpoints of a relationship line show a primary key symbol to denote a primary key-to-foreign key relationship, or they show an infinity symbol to denote the foreign key side of a one-to-many relationship.

Applications based on an RDBMS can offer better performance and scalability.



#### **D) SQL Queries:**

End users seldom access data warehouse data directly using Structured Query Language (SQL) queries. Analytical SQL queries can be quite complex, requiring database expertise to create correctly. The volume of data in a data warehouse is often so large that sophisticated SQL techniques are needed to achieve useful performance. A SQL query that joins three or four dimension tables to a fact table containing millions of rows and uses aggregate functions such as SUM to summarize and group the results can impose a significant load on any relational database and often yields performance that is not acceptable for online analysis. The use of SQL queries is one of the oldest methods of accessing data. Microsoft® SQL Server™ 2000 provides sophisticated query processing and optimization techniques.

General SQL Queries:

select, insert, update, or delete statements.



The SELECT statement retrieves rows by specifying the columns of interest, the scope (the set of files) for the search, and the search criteria. If the results must be ordered, you can add an additional ORDER BY Clause to return the rows in ascending or descending order.

**SELECT LIST**  
**FROM CLAUSE**  
**WHERE CLAUSE**  
**ORDER BY**

The Insert Statement Appends a record to the end of a table that contains the specified field values.

**INSERT INTO** *Database Name1* [(*fname1* [, *fname2*, ...])] **VALUES** (*eExpression1* [, *eExpression2*, ...])

The Update Statement Updates records in a table with new values.

**UPDATE** *DatabaseName1 Table Name1* **SET** *Column\_Name1* = *eExpression1* , *Column\_Name2* = *eExpression2* ...  
**WHERE** *FilterCondition1* AND | OR *FilterCondition2* ...

The Delete Statement Marks records for deletion.

**DELETE FROM** *Database Name Table Name*  
**WHERE** *FilterCondition1* AND | OR *FilterCondition2* ...



### 3.2 .NET

We have used .NET Framework in this Project, which very efficiently and easily interacts with any Script (VB, ASP, C, C++ etc.) and gives high productivity results. The concept of Multi- Users, Multi- Processing etc. in our Project came into existence only after the use of .NET. The features .NET incorporates like Reusability of data reduced much of complexity and bulk of the Project. Other notable features are covered below.

#### 3.2.1 THE .NET FRAMEWROK

*Man's greatest ability is not his ability to read, write or think. His greatest ability is to dream. Because unless one has dreams they cannot come true. The .NET framework is the result of combined dreams of programmers of last decade or so.....*

The .NET framework is a high productivity standards – based, multi – language application execution environment that handles essential plumbing chores and eases deployment. It provides an application execution environment that manages memory, addresses versioning issues and improves the reliability, Scalability and security of the application. The .NET framework consists of several parts including Common Language Runtime and a rich set of class libraries. Web services are also provided that releases developers from the burden of building everything themselves.

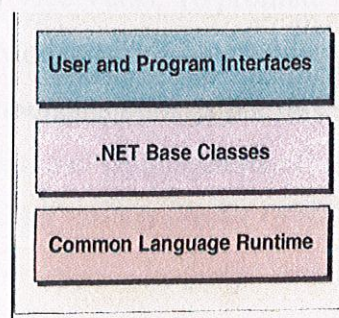
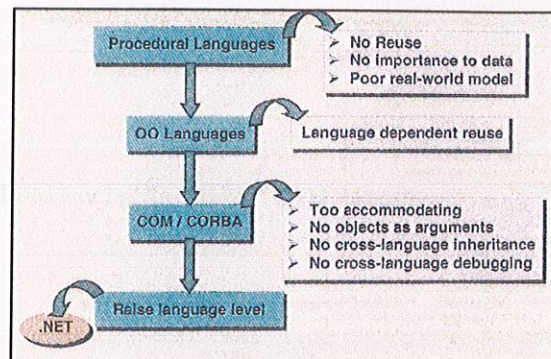


Figure 3.1



### 3.2.2 FEATURES OF .NET



**Figure 3.2**

#### **Reusability**

In Procedural languages like C, reuse was limited to calling the same library functions from different programs. In Object Oriented languages like C++ and Java also reusability was limited to inheritance i.e. these languages promoted reuse by letting the user inherit the features of one class into another and then provide additional features in the new class. This was possible even if the source code of the first class was not available. This reuse was a big step forward. However the reuse was restricted within the language. Thus a Java class was not of much use to a C++ programmer and Vice Versa. To promote language independent reuse Component Object Model (COM) was introduced. COM was a binary specification and components built as per this specification could be used not only by the conventional languages like C or C++ but also by scripting languages like VBScript or JavaScript. COM didn't supported cross – language inheritance or cross – language debugging. Instead of letting all kinds of languages to use COM components and requiring a lot of plumbing code to be written in the process, .NET adopts a different



philosophy. In .NET the capabilities of the language itself are raised in such a manner that they can use COM components without being required to write lot of plumbing code. Here language interoperability is possible with great ease. .NET allows languages to be integrated with each other. For example it is possible to create a class in C++ that derives from a class implemented in Visual Basic. Hence .NET defines reusability in true sense.

### EASY DEPLOYMENT: Solution To DLL Hell

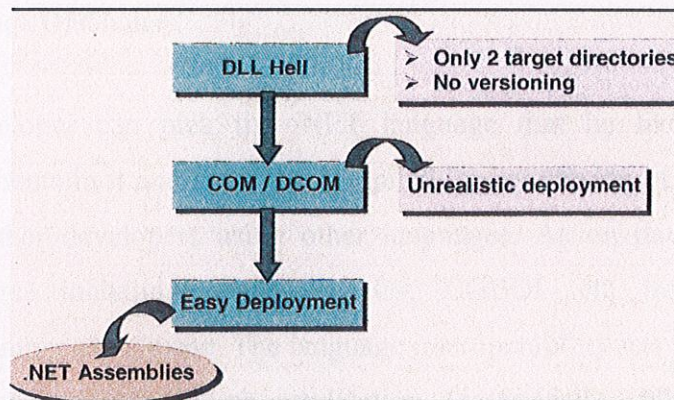


Figure 3.3

DLLs are created with a motive to share some code between more than one application. Hence while installing them instead of placing them in the application directory they are installed in either windows directory or windows\system directory. This a very useful feature of DLLs but the problem crops up when a newer version of the DLL is released. If the newer version is copied onto the older version the applications sharing the older version might break and stop functioning. Thus the two DLLs have to have a different name if the two versions are to coexist. This versioning problem is popularly known as **DLL HELL**



COM overcame this problem by storing two different version of a same DLL in two different directories. But the solution was not satisfactory because the COM component, the interface (through which it interacted with DLL) and the class ID of DLL were all in different locations and if any of these went bad, the component used to be rendered useless. .NET has come up with a different deployment idea called 'Assembly' which gives a solution to DLL HELL in that the class ID has knowledge of previous class ID and so the new assembly needs not to be sent to every client. The applications which used older version will continue using older version and other applications can use newer version without any conflict.

### **Language Of Choice**

A developer can pick the .NET language that he likes most, write components in it and share the compiled binary version of his component with other developers using other languages. As on date there are 28 languages including VB, C++, C#, COBOL etc from which the programmer can choose. The language interoperability lets a user develop different modules of an application in different .NET compatible languages.

### **Structured Runtime**

DOS was a single tasking system that was unrealistic in meeting the ever growing demands as it could run only one program in memory. As a solution Windows emerged and was appreciated a lot as it allowed users to run more than one programs in memory. But as the use of windows went on, the problems that came to light were. Memory leaks poor error handling and costly and dangerous IPC.



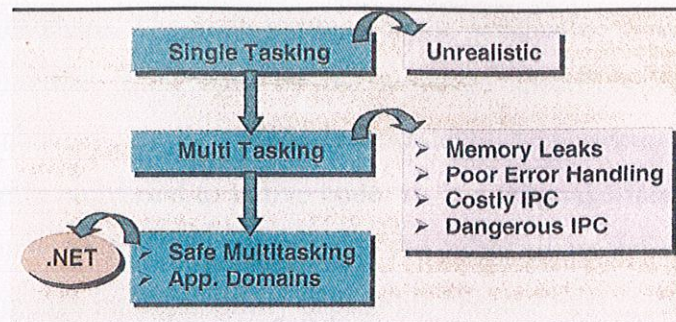


Figure 3.4

.NET overcomes these limitations by using a concept called Application Domains.

### The Common Language Runtime

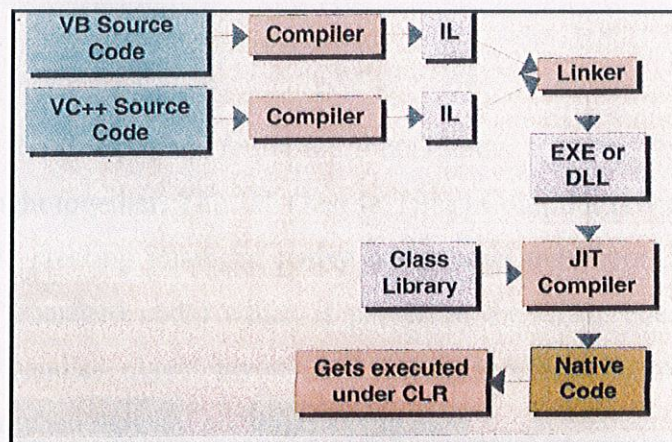


Figure 3.5

A runtime is an environment in which programs are executed. If a user wants to run an application built using any language his machine must have the runtime of that language installed. To avoid such problems



.NET introduced a single CLR (Common Language Runtime) that all the .NET languages share. The code written for the application is converted into IL (Intermediate Language) by the language specific compiler as shown in the above figure. IL is CPU independent set of instructions that can be efficiently converted to native code. IL is a language independent code. The course of further execution of the program is shown in the figure. The language interoperability is possible because of IL's as an IL is machine and language independent.

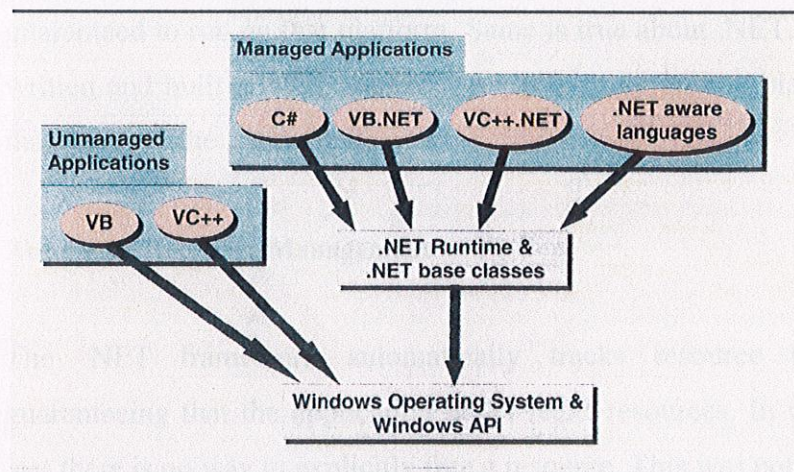


Figure 3.6

On execution, the IL code and requested functionality from the .NET base classes is brought together. The JIT (Just In Time) Compiler processes the combined code creating managed native code (code that targets the run time is called managed code) which is passed to the CLR. The runtime automatically handles object layout and manages references to objects releasing them when they are no longer being used.



### **3.2.3 WHY .NET MAKES SENSE?**

.NET is the design tool of future as it has the following advantages: -

#### **Platform Independence**

Any java program runs under a JVM (Java Virtual Machine). So for any new platform if JVM exists then the java program is guaranteed to run on that platform. Same is true about .NET. Once written and built a .NET application can execute on any platform that supports the .NET CLR.

#### **Automatic Resource Management**

The .NET framework automatically tracks resource usage, guaranteeing that the application never leaks resources. In fact in .net there is no way to explicitly free a resource. This was not the case before where a programmer had to explicitly allocate and free resources.

#### **Security**

Traditional OS security is based on restricting/granting access based on user accounts and user role. This is a fairly reliable model but it assumes that all code is equally trustworthy .with increasing reliance on mobile code such as web scripts, internet applications downloads there is a need of finer control. .NET code access security model delivers this control.



### **Less Plumbing Code**

VS.NET development environment eliminates a lot of plumbing code required by WIN 32 or COM applications. .NET provides wizards liberal use of drag and drop functionality and standard classes that developers can use to create powerful applications quickly

### **.NET Base Classes:-**

.NET supplies a library of base classes that developers can use to implement applications quickly. These classes include:-

- classes and types related to basic database management.
- classes to debug an application and to trace the execution.
- classes that allow reading/writing to files and other data streams.
- classes to perform common mathematical operations.
- classes to inspect metadata of an assembly.
- classes that enable security capabilities.



## THE COMPLETE PICTURE

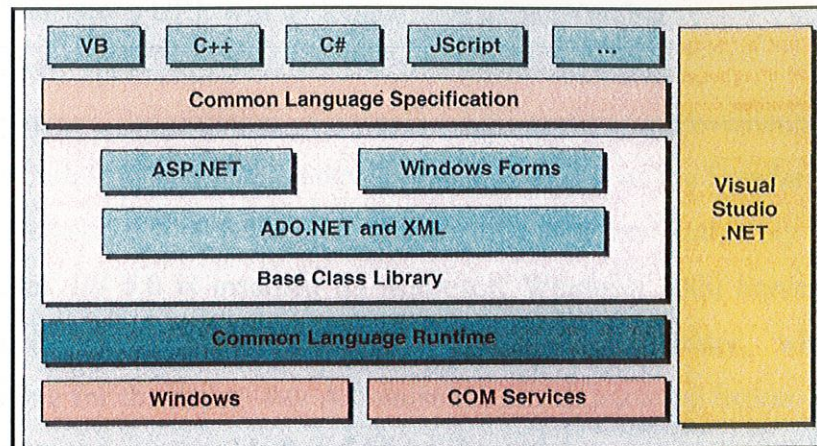


Figure 3.7

The above picture shows the complete scenario of working of .NET and how various components interact with each other under the development environment. The languages at the top of hierarchy are increasing with time so as to make the choice of languages available more vast.



### 3.3 IIS

We have designed our Project using a Designing Tool named Visual Studio 2005. The basic use of IIS :-

- In order to ensure that our web application are available even when Visual Studio is not running.
- To allow users on other computers in the network to run our Web application.
- Web applications URL will no longer need a Port Number

IIS (Internet Information Services) 5.0 sports improvements for administration and application development. Building on the foundation of its predecessor, IIS 5.0 takes performance, reliability, and scalability to the next level. IIS 5.0 is installed on Microsoft Windows 2000 Server and Advanced Server by default. The System Administrator, who is responsible for the day-to-day Administration of Commerce Server 2000, is typically also responsible for administering IIS 5.0

We use the IIS snap-in to manage Web site access permissions and security, performance, directories and paths, the identification of Web site names, and Internet Protocol (IP) addresses and ports.

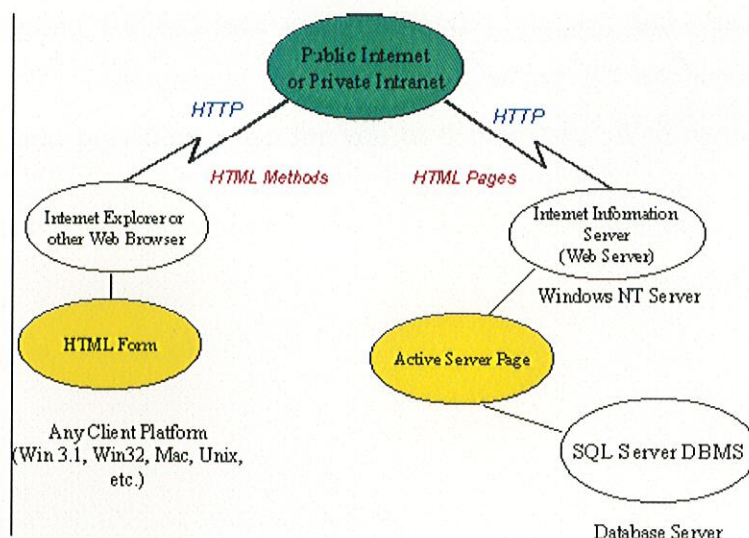


Figure 3.8



### 3.4 ASP.NET

Visual Studio .NET allows you to create applications that leverage the power of the WWW (World Wide Web). This includes everything from a traditional Website that serves HTML pages, to fully featured business applications that run on an intranet or the Internet, to sophisticated business-to-business applications providing Web-based components that can exchange data using XML. A Visual Studio Web application is built around ASP.NET. ASP.NET is a platform — including design-time objects and controls and a run-time execution context — for developing and running applications on a Web server.

ASP.NET in turn is part of the .NET Framework, so that it provides access to all of the features of that framework. For example, you can create ASP.NET Web applications using any .NET programming language (VB, C#, Managed Extensions for C++, and many others) and .NET debugging facilities. You access data using ADO.NET. Similarly, you can access operating system services using .NET Framework classes, and so on. ASP.NET Web applications run on a Web server configured with Microsoft IIS. However, you do not need to work directly with IIS. You can program IIS facilities using ASP.NET classes, and Visual Studio handles file management tasks such as creating IIS applications when needed and providing ways for you to deploy your Web applications to IIS.



### 3.4.1 ELEMENTS OF ASP.NET WEB APPLICATIONS

Creating ASP.NET Web applications involves working with many of the same elements you use in any desktop or client-server application. These include:

- **Project management features** When creating an ASP.NET Web application, you need to keep track of the files you need, which ones need to be compiled, and which need to be deployed.
- **User interface** Your application typically presents information to users; in an ASP.NET Web application, the user interface is presented in Web Forms pages, which send output to a browser. Optionally, you can create output tailored for mobile devices or other Web appliances.
- **Components** Many applications include reusable elements containing code to perform specific tasks. In Web applications, you can create these components as XML Web services, which makes them callable across the Web from a Web application, another XML Web service, or a Windows Form, for example.
- **Data** Most applications require some form of data access. In ASP.NET Web applications, you can use ADO.NET, the data services that are part of the .NET Framework.
- **Security, performance, and other infrastructure features** As in any application, you must implement security to prevent unauthorized use, test and debug the application, tune its performance, and perform other tasks not directly related to the application's primary function. The following diagram provides an overview of how the pieces of ASP.NET Web



applications fit together and fit into the broader context of the .NET Framework.

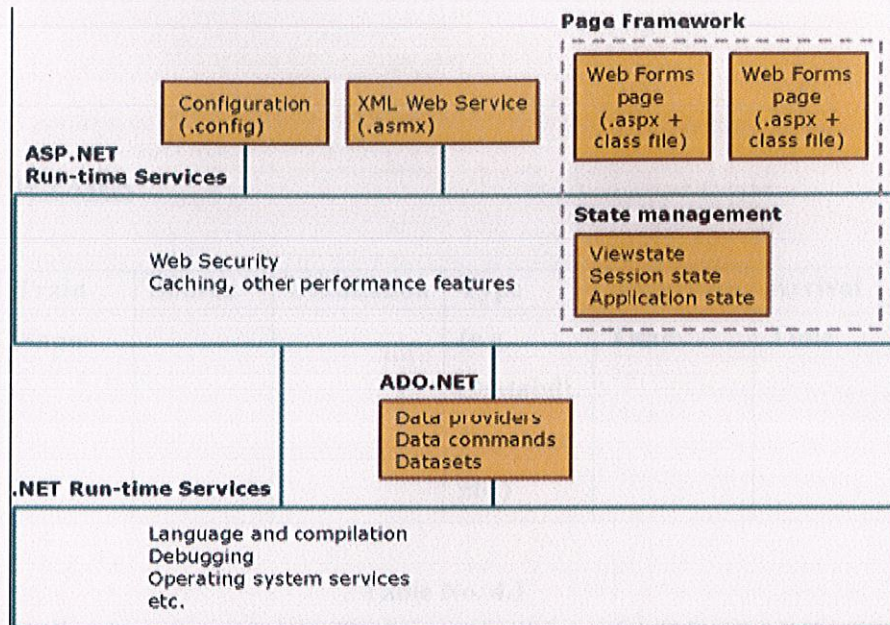


Figure 3.9

### How IIS handles an ASP file request

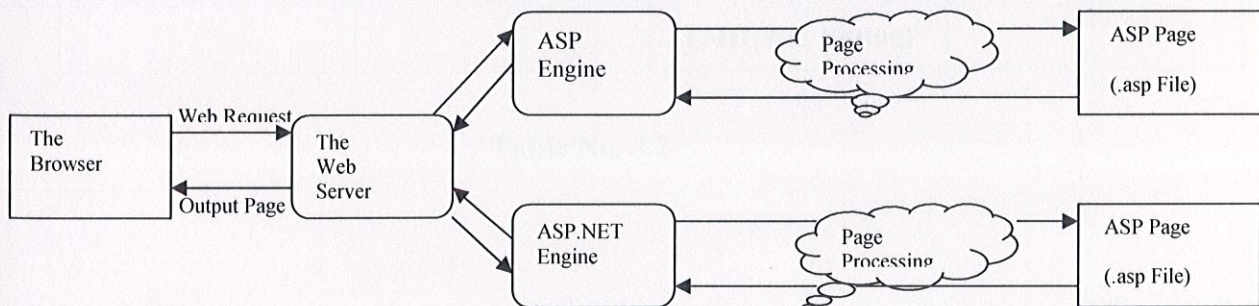


Figure 3.10



#### **4. ENTITIES / TABLES USED**

The Database would be filled with values according to following Tables and Attributes of them:-

##### **4.1 TRAIN DETAILS**

<b>Train No.</b>	<b>Train Name</b>	<b>Source</b>	<b>Destination</b>	<b>Type (e.g. Shatabdi, Rajdhani etc.)</b>	<b>Departure Time</b>	<b>Arrival Time</b>	<b>No. of Stoppages</b>

**Table No. 4.1**

##### **4.2 FARES**

<b>Distance (in Kms.)</b>	<b>Distance- Fare</b>	<b>Class (e.g. Sleeper, AC-III, EE, Sitting)</b>	<b>Reservation Charge</b>

**Table No. 4.2**



#### 4.3 SOURCE- DESTINATION

Source	Destination	Distance	Train No.	Sttopage Time
--------	-------------	----------	-----------	---------------

Table No. 4.3

#### 4.4 TRAIN STATUS

Train No.	Date	I-Class	AC-1	AC-2	Sleeper	Unreserved	Tatkal
-----------	------	---------	------	------	---------	------------	--------

Table No. 4.4

#### 4.5 TRAIN SCHEDULE

Train No.	Stop No.	Stop Name	Stoppage Time	Distance (from source stop )	Time of Arrival	Platform No.
-----------	----------	-----------	---------------	------------------------------	-----------------	--------------

Table no. 4.5



#### 4.6 PASSENGER DETAILS

PNR	Train No.	Passenger Name/s	Gender	Age	Address	Boarding Station	Destination Station	Date of Boarding	Class	Coach No.	Seat No.	Status
-----	-----------	------------------	--------	-----	---------	------------------	---------------------	------------------	-------	-----------	----------	--------

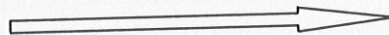
Table No. 4.6

#### 4.7 UNIVERSAL RELATIONS

##### 4.7.1 FUNCTIONAL DEPENDENCIES

The functional dependencies shall provide the platform on which our process of decomposition to normalize the universal table is based.

- **Partial** - Redundancy of elements, due to which anomalies like Update, Insert, Delete are tracked.
- **Transitive** – An attribute B of a relation R is transitively dependent on attribute A of R if it is functionally dependent on an attribute C which in turn is functionally dependent on A or any proper sub-set of A.



Train No.		Train Name
Train No.		Type
Train No.		Departure Time
Train No.		Arrival Time
Train No.		No. of Stoppages
Train No.		Source
Train No.		Destination
Distance		Distance- Fare



<b>Class</b>		<b>Reservation Charge</b>
<b>Destination, Train No.</b>		<b>Stoppage Time</b>
<b>Source, Destination, Train No.</b>		<b>Distance</b>
<b>Train No.</b>		<b>I-Class Seats</b>
<b>Train No.</b>		<b>II-Class Seats</b>
<b>Train No.</b>		<b>Tatkal Seats</b>
<b>Train No., Stop No.</b>		<b>Stop Name</b>
<b>Train No., Stop Name</b>		<b>Stop No.</b>
<b>Train No., Stop No.</b>		<b>Stoppage Time</b>
<b>Train No., Stop Name</b>		<b>Stoppage Time</b>
<b>Train No., Stop No.</b>		<b>Distance</b>
<b>Train No., Stop Name</b>		<b>Distance</b>
<b>Train No., Stop No.</b>		<b>Time of Arrival</b>
<b>Train No., Stop Name</b>		<b>Time of Arrival</b>
<b>Train No., Stop No.</b>		<b>Platform No.</b>
<b>Train No., Stop Name</b>		<b>Platform No.</b>
<b>Train No., Distance</b>		<b>Stop No.</b>
<b>Train No., Distance</b>		<b>Stop Name</b>
<b>Train No., Distance</b>		<b>Platform No.</b>
<b>Train No., Distance</b>		<b>Time of Arrival</b>
<b>Train No., Distance</b>		<b>Stoppage Time</b>
<b>PNR</b>		<b>Passenger Name</b>
<b>PNR</b>		<b>Train No.</b>
<b>PNR</b>		<b>Passenger Address</b>
<b>PNR</b>		<b>Gender</b>
<b>PNR</b>		<b>Age</b>
<b>PNR</b>		<b>Boarding Station</b>
<b>PNR</b>		<b>Destination</b>
<b>PNR</b>		<b>Date of Boarding</b>



PNR		Seat No.
PNR		Coach No.
PNR		Status
PNR		Class
Train No., Coach No., Seat No., Date, Target Station		PNR
Train No.		Train Name
Train No.		Type
Train No.		Departure Time
Train No.		Arrival Time
Train No.		No. of Stoppages
Train No.		Source
Train No.		Destination

**Table No. 4.7**

#### **4.7.2 SUB-SCHEMAS**

- 1.) Train Name, Train No., Source, Destination, Type, Departure time, Arrival time, No. Of Stoppages.

**Primary key:** Train No. completely determines all attributes.

- 2.) Distance, Distance-Fare, Class, Reservation-Charge.

**Primary key:** Distance, Class

- 3.) Train No., Source, Destination, Distance, Stoppage Time.



**Primary key:** Train No., Source, Destination

- 4.) Train No., I-Class, AC-I, AC-II, Sleeper, Unreserved.

**Primary key:** Train No.

- 5.) Train No., Stop Name, Stop Number, Stoppage Time, Time of Arrival, Platform No., and Distance.

**Primary key:** Train No., Stop Number

- 6.) PNR, Train No., Passenger Name, Gender, Age, Address, Boarding Station, Target Station, Date of Boarding, Class, Coach No., Seat Number, Status (Reserved, Cancelled etc.)

**Primary key:** PNR

#### 4.7.3 ANALYSIS

The analysis of the given relations and derived Sub- Schemas using the properties:

1. Loss less if  $r_1 \rightarrow r_2 \rightarrow r_1$  or  $r_1 \rightarrow r_2 \rightarrow r_2$
2. F.D. preserving if  $f = f_1 \cup f_2 \dots \cup f_n$  Then  $f = f$  or  $f^+ = f^+$

Reveals that they are loss less and F.D. preserving.

The Normalization of schemas is shown here:



- **1 NF:**

The attributes of sub-schemas 1, 2, 3, 4, 5, 6 are atomic in nature and cannot be further divided into sub-fields. Hence, the given schema of relational schemes is in 1 NF.

- **2 NF:**

(1.) is a single attribute, primary key relation. Hence, it is free of any P.F.D

(2.) has the P.F.D's given as:-

A.)

Distance, Class	Distance- Fare
Distance	Distance- Fare

**Table No. 4.8 A**

B.)

Distance, Class	Reservation- Charge
Class	Reservation- Charge

**Table No. 4.8 B**

Where the primary key is Distance, Class



Hence we represent (2.) as

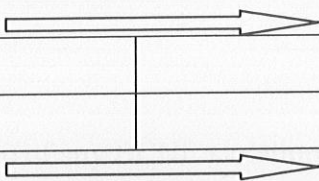
(2-i) distance, class

(2-ii) distance, distance fare

(2-iii) class, reservation charge

(3.) has P.F.D's given as: -

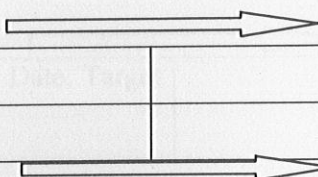
A.)



Train No., Source	Distance
Train No.	Distance

**Table No. 4.8 C**

B.)



Train No., Destination	Stoppage Time
Train No.	Stoppage Time

**Table No. 4.8 D**

Hence, decomposing 3 as

(3-i) Train No., Source, Destination, Distance

(3-ii) Train No., Destination, Stoppage Time



5, 6 is free of P.F.D's

Now the schema is in 2 NF

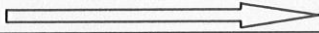
- **3 NF:**

(1.), (2.), (3.), (4.), (5.), (6.) does not have any Transitive Dependency.

- **BCNF**

(1.), (2.), (3.), (4.), (5.) are free of any BCNF violation.

In (6.), PNR is the Primary key and the under said dependency violates the BCNF



Train No., Coach No., Seat No., Date, Target Station	PNR
--	-----

**Table No. 4.9**

Hence we break 6 as:

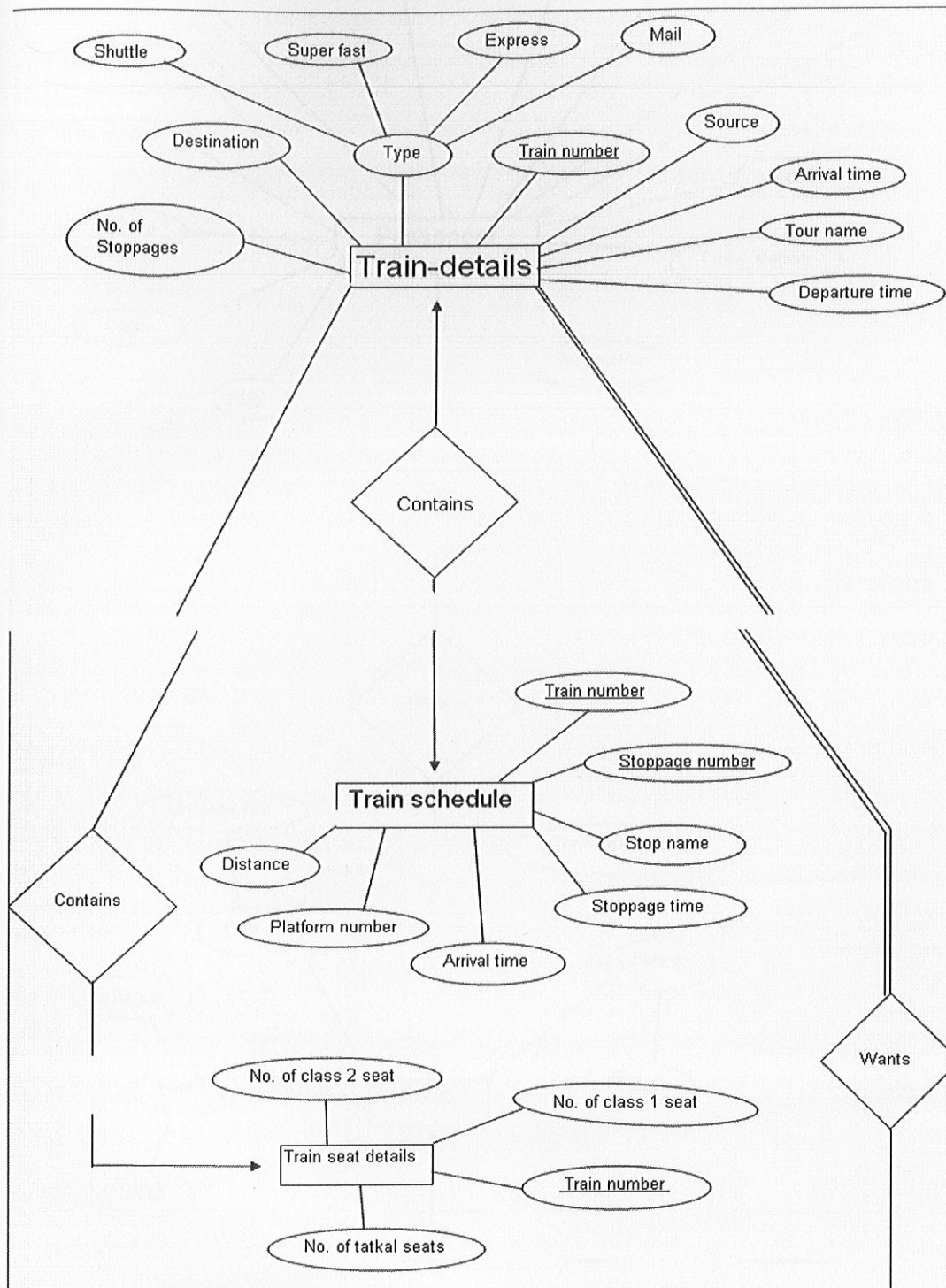
6-i) PNR, Seat Number, Date, Target Station, Seat Number, Coach no.

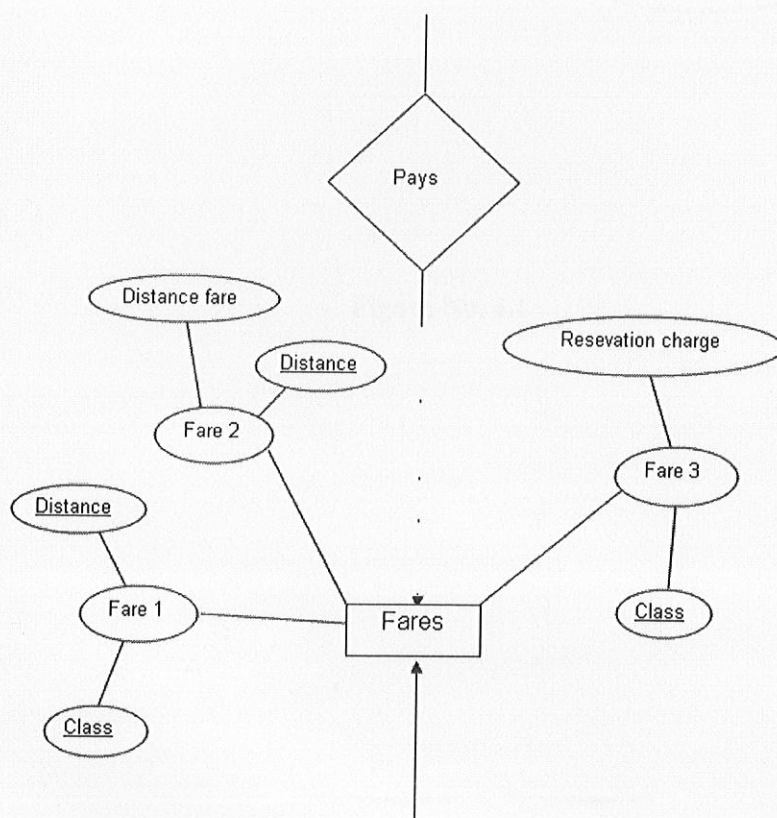
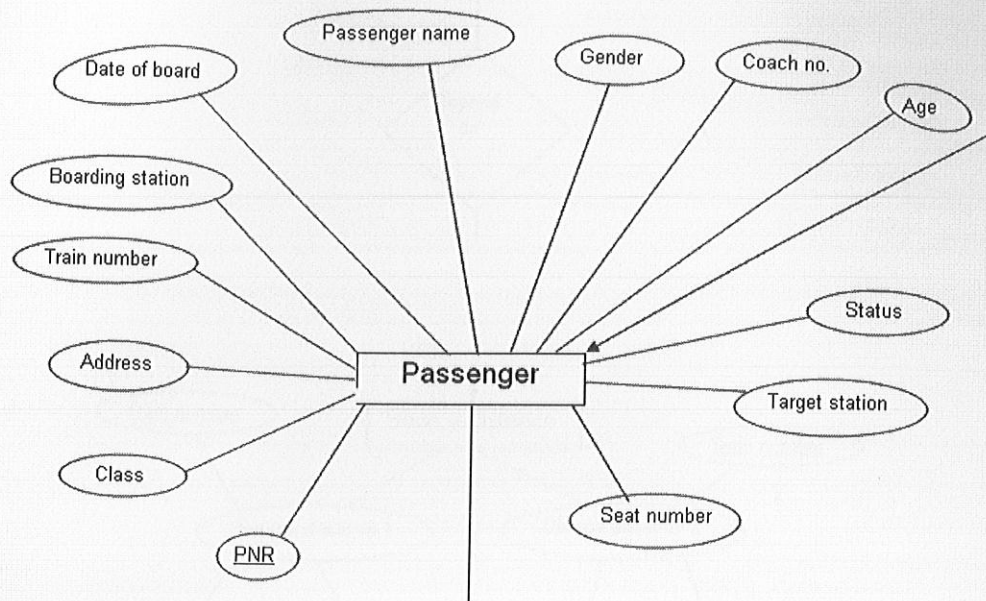
6-ii) Passenger Name, Gender, Age, Address, Boarding Station, Status

Hence, BCNF is achieved. **THE DATABASE TABLES ARE NORMALISED**

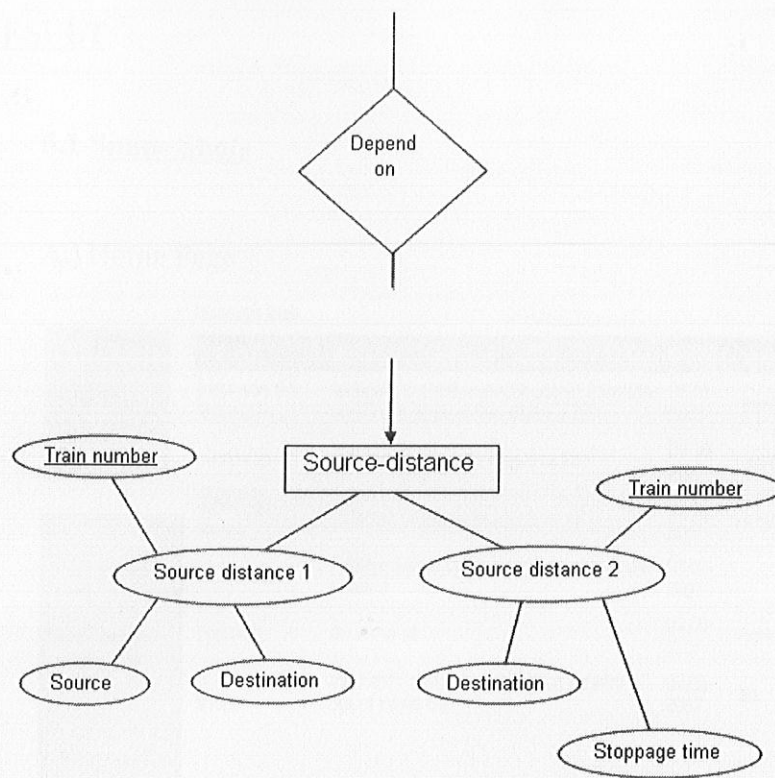


#### 4.8 PROCESS FLOW DIAGRAM: "E-R Model"









**Figure No. 4.1**



## 5. RESULT

### 5.1 Snap- Shots

#### A.) Home Page

[Home](#)  
[Train List](#)  
[Book A Ticket](#)  
[Cancelation](#)  
[PNR Status](#)

Site Stats

Total Clients	3
Total Employee	5
Total Hits	134
Your Hits	133

Message Box

Subject	message	Dated	publisher
India won the first match..	India win the cricket match against Bangladesh and Dhoni declared Man of the match	05-11-2007	naveen
World cup	Pakistan Lost to Ireland by 3 wickets.	03-18-2007	sumit
World cup shokes	India lost to bangladesh by 5 wickets	03-18-2007	sumit
Bobs Murder	Pakistan Teams Coach has mudered by some bookies!	03-05-2007	
Welcome	Welcome All	03-05-2007	naveen
World cup Shocks	India and Pakistan Two big Asian Country is out of World Cup.	03-05-2007	E1

Log In


User Name:

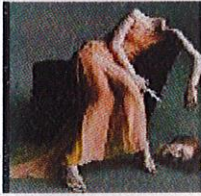
Password:


UserType

[Forgot Password?](#)

Advertisement







[Home](#)   [Contact Us](#)   [Created By](#)

Figure 5.1



## B.) Running View of HOW To BOOK A TICKET

**Home**

**Train List**

**Book A Ticket**

**Cancellation**

**PNR Status**

**Book Tickets**

PNR Number

Dated

Train Number

Train Name

Total Seats

Available Seats

From

To

Departure Time

Arrival Time

Distance

Cost

Seat Booked

Total Cost

Pay Mode

Status

**Site Stats**

Total Clients	3
Total Employee	5
Total Hits	134
Your Hits	133

[Home](#)

[Contact Us](#)

[Created By](#)

Figure 5.2



### C.) View of Administrator Window

Welcome : ashish
[Sign Out](#)

[Home](#)
[Admin Panel](#)
[User Master](#)
[Seats](#)
[Distance](#)
[Locations](#)
[Train List](#)
[Advertisement](#)
[Posts Master](#)
[Sign Out](#)

Subject	message	publisher	dated	Verified
World cup	Pakistan Lost to Ireland by 3 wickets.	sumit	3/18/2007 12:00:00 AM	Yes
World cup shokes	India lost to bangladesh by 5 wickets	sumit	3/18/2007 12:00:00 AM	Yes
World cup Shocks	India and Pakistan Two big Asian Country is out of World Cup.	E1	3/5/2007 12:00:00 AM	Yes
Bobs Murder	Pakistan Teams Coach has muderred by some bookies!		3/5/2007 12:00:00 AM	Yes
Welcome	Welcome All	naveen	3/5/2007 12:00:00 AM	Yes
India won the first match..	India win the cricket match against Banglades and Dhoni declared Man of the match	naveen	5/11/2007 12:00:00 AM	Yes

[Verify](#)
[Delete](#)

Subject

Message

To

All

▼

[Post](#)

[Home](#)
[Contact Us](#)
[Created By](#)

**Figure 5.3**



## **6. CONCLUSION**

To conclude with, the Project has vital scenarios and application in 21<sup>st</sup> century world. It is of mass and public use. This software throws light on one of the most important features of concurrent accessing once it is installed in the network. The framework of the software we designed can easily be implemented on many other Database systems through some modifications. The tools used in the Project are latest and couple with the modern technology. So any new requirement in the Project would be very easily and efficiently implemented without use of any new specification or software. This can be judged by the easy application of it over a network. On completion of this Project we finally got thorough with the concepts of Database Management, Server- Client Mechanism in a Network. As an individual one can learn but with planning and teamwork we can implicate our knowledge and learning's towards development and success.



## **7. FUTURE SCOPE OF PROJECT**

Railway System is a Department which is seen in nearly all the countries of world. So any new Technology, Feature or Attribute brought up in this field would surely have a bright future. Our Project is constrained with Indian Sub- Continent, but the basic idea of development and adding up with features lie the same. The web application based Railway Management System of ours not only serves users with the aide of just enquiring of trains or getting a ticket done (for example), but it also incorporates the features like News Updates, Advertisements, Latest up comings in the world, personalized user accounts etc. Henceforth the Future Scope of this Project can be, an in-built Search Engine using which users can harness information from any part of Internet. This way an individual can search/ surf net along with using the Railway Management System and that too by opening just one site. So this will reduce the complexity of opening too many sites individually.



## **8. REFERENCES**

- [www.indianrail.gov.in](http://www.indianrail.gov.in)
- [www.wikipedia.com](http://www.wikipedia.com)
- [www.google.com](http://www.google.com)
- [www.iretc.com](http://www.iretc.com)
- **ASP.NET 2.0 A Developer's Notebook By Wei- Meng Lee.**
- **Beginning ASP.NET 2.0 in VB 2005 By Matthew Mac Donald.**
- **Database System Concepts By Silberchatz, Korth and Sudershan.**