ACKNOWLEDGMENT

I express thanks and gratitude to Miss. B.JEHAN H.O.D M.B.A(IT), VIF COLLAGE OF ENGINEERING AND TECHNOLOGY for his encouraging support and guidance in carrying out the project.

I would like to express gratitude and indebtedness to Miss.B.JEHAN Faculty member, for his valuable advice and guidance without which this project would not have seen the light of the day.

I thank Mr. Sagar, Project guide, GALAXIE software solutions for providing us with an excellent project and guiding me in completing our project successfully. I would like to thank all the staff members of GALAXIE software solutions for their kind co-operation. I would like to thank my parents for being supportive all the time, and I am very much obliged to them.

SRIKANTH.T
(05D51E0034)

HUMAN RESOURCE MANAGEMENT SYSTEM

CONTENTS

Page nos.

1. INTRODUCTION

1.1 HUMAN RESOURCE MANAGEMENT SYSTEM

To develop a software application that supports the application specific to the HR automation in an intranet specific to a company there by allowing the integration of all the employees pertaining to that organization. To keep track of all the other departments related to that organization such as marketing, development etc.

To allow the HR of an organization to update the employee details when ever there is a change in the employee profile pertaining to that organization. To bring onto a string the employee specific suggestions and make them free to post their requirements to the HR thus bringing the organization more specific regarding the maintenance of the organization.

1.1.2 PURPOSE OF THE STYDY:

The HR center is a powerful application designed to allow companies to streamline their human resource tasks and manager their employees more efficiently

Employee and Company Information
Employee Time, Attendence, and Leave Request
HR Documentation Management (i.e. insurance forms, W-2's etc.)

The HR Center includes a comprehensive employee information database, work information, beneficiary information, and more for each employee. It comes standard with employee self-service access allowing employees to update their personal information, request time off or input their daily timesheet entries. It also has role –based access level control that is functionally based on whether a user is an employee, a manager, or an HR admistrators. With HR center managers and HR admistrators can manage an track

1.1.3 OBJECTIVE OF THE STUDY:

To develop a software application that supports Specific to the HR Automation in an intranet to a company there by allowing the interaction of all the employees pertaining to that organization. To keep track of all the other departments related to that organization like marketing, research etc.,

To allow the HR department of an organization to update the employee details when ever there is a change in the employee profile pertaining to that organization. To bring on to a string the employee specification suggestions and make them sure to post their requirements to the HR.

1.1.4 SCOPE OF THE STUDY:

Allow a for the creation of an application specification to the human resource that maintain the intranet automation of the HR software i.e., which contains the data related to the employee.

PROJECT OVERVIEW AND LIMITTIONS:

This project can be used to identify a employee in the organization. The project maintains the details of the entire employee. Each employee is given with different employee Id.

In this I can use the vb.net language. Back end is ms-access. Duration of this project is 3 months .i was discussed the total modules of HR department. Limits are there. In this project I was taken only HR department from organization.

2.0 COMPANY PROFILE

GALAXIE SOFTWARE SOLUTIONS

Galaxie Software Solutions is an IT solution provider for a dynamic environment where business and technology strategies converge. Their approach focuses on new ways of business combining IT innovation and adoption while also leveraging an organization's current IT assets. Their work with large global corporations and new products or services and to implement prudent business and technology strategies in today's environment.

Galaxie's range of expertise includes :

- Software Development Services
- Engineering Services
- Systems Integration
- Customer Relationship Management
- Product Development
- Electronic Commerce
- Consulting
- IT Outsourcing

We apply technology with innovation and responsibility to achieve two broad objectives:

- Effectively address the business issues our customers face today.
- Generate new opportunities that will help them stay ahead in the future.

This approach rests on:

- A strategy where we architect, integrate and manage technology services and solutions - we call it AIM for success.
- A robust offshore development methodology and reduced demand on customer resources.

- A focus on the use of reusable frameworks to provide cost and times benefits.

They combine the best people, processes and technology to achieve excellent results - consistency. We offer customers the advantages of:

Speed :

They understand the importance of timing, of getting there before the competition. A rich portfolio of reusable, modular frameworks helps jump-start projects. Tried and tested methodology ensures that we follow a predictable, low - risk path to achieve results. Our track record is testimony to complex projects delivered within and evens before schedule.

Expertise :

Our teams combine cutting edge technology skills with rich domain expertise. What's equally important - they share a strong customer orientation that means they actually start by listening to the customer. They're focused on coming up with solutions that serve customer requirements today and anticipate future needs.

A full service portfolio:

They offer customers the advantage of being able to Architect, integrate and manage technology services. This means that they can rely on one, fully accountable source instead of trying to integrate disparate multi vendor solutions

Services:

GSS is providing its services to Sain Medicaments Pvt. Ltd., Grace Drugs and Pharmaceuticals Pvt. Ltd., Alka Drugs and Pharmaceuticals Pvt. Ltd., Hi-tech Steels, Real Foods,Ravi Foods ,to name a few. With their rich expertise and experience in information technology they are in the best position to provide software solutions to distinct business requirements.

3.1 EXISTING SYSTEM

EXISTING SYSTEM

The HR Administration falls short of controlling the employee’s activities in analyzing his/her strengths and weakness. The decision for appraisal of assigning next project to the employee or to train him/her to enhance the skills – where lies with proper projection. He is not provided with the detailed project information done or to be assigned based on Application / Verticals.

DRAWBACKS IN EXISTING SYSTEM:

- Need of extra manual effort.
- It used to take much time to find any employee
- Not very much accurate.
- Danger of losing the files in some cases.

3.2 PROPOSED SYSTEM

Decision in assigning proper skillful hands for the project is an important issue in HR Module. The HR Administrator should report with the personal holding the necessary skills required for the project assignment. The decision in making analysis about the employee's skills is a prime important before booting in. The proposed system of HR Module is the right software to be incorporated into the Automation of HR Software for helping the organization needs with respect to skilful Human Resource.

The proposed system provides detail general information about the employee along with Educational, Certification, Skill and Project details. It enhances the HR Management in adding, viewing and updating employees' details and generates various reports regarding employee's skill and experience. Suggestions and Grievances posted by the employees are upheld for taking care of the necessary steps in forwarding company's obligation.

.

3.2.1 ADVANTAGES OF PROPOSED SYSTEM:

- Very fast and accurate.
- No need of any extra manual effort.
- No fever of data loss.
- Just need a little knowledge to operate the system.
- Doesn't require any extra hardware device.
- At last very easy to find the employees.

3.3 FEASIBILITY STUDY

Once the problem is clearly understood, the next step is to conduct feasibility study, which is high-level capsule version of the entered systems and design process. The objective is to determine whether or not the proposed system is feasible. The three tests of feasibility have been carried out.

- Technical Feasibility
- Economical Feasibility
- Operational Feasibility

TECHNICAL FEASIBILITY

In Technical Feasibility study, one has to test Whether the proposed system can be developed using existing technology or not. It is planned to implement the proposed system using java technology. It is evident that the necessary hardware and software are available for development and implementation of the proposed system. Hence, the solution is technically feasible.

ECONOMICAL FEASIBILITY

As part of this, the costs and benefits associated With the proposed system compared and the project is economically feasible only if tangible or intangible benefits outweigh costs. The system development costs will be significant. So the proposed system is economically feasible.

OPERATION FEASIBILITY

It is a standard that ensures interoperability
Without stifling competition and innovation among users, to the benefit of the public both in terms of cost and service quality. The proposed system is acceptable to users. So the proposed system is operationally feasible.

4. MODULE DESCRIPTION:

The list of modules incorporated with "**Human Resource Management System"** is

- Employee Info Module
- Administration Module
- Project Management Module
- Training Management Module
- HR Reports

This module deals with the management of the employee information such as the personal details-his name,qualification,skill,experience,login id,password,etc., Importance of modules in any software development side is we can easily understand what the system we are developing and what its main uses are. At the time of project we may create many modules and finally we combine them to form a system.

4.1 Employee Info Module

This module deals with the management of the employee information such as the personal details name, qualification, skill, experience, login id, password, etc., Importance of modules in any software development side is we can easily understand what the system we are developing and what its main uses are. At the time of project we may create many modules and finally we combine them to form a system person, so that it can be easily added to the database with any duplication of the data.

4.2 ADMINISTRATION MODULE:

This module deals with the management of the employee information such as the hiring of the eligible candidate, payments criteria, his personal information maintenance etc.

4.3 PROJECT MANAGEMENT MODULE:

This module deals with the management of the projects related with the employee like-projects that were past dealt, current projects in his account etc.

4.4 TRAINING MANAGEMENT MODULE:

This module deals with the training of the employee based on his experience and attendance monitoring. Also the information of the projects that need to be trained for the employees based on their experience and skills and the like.

4.5 HR REPORTS MODULE:

This module is specified for the purpose of the report generation for the HR on desired requests.

5. SYSTEM REQUIREMENTS

5.1 Software Specification

Language	:	VB.NET, ADO.NET.
Database	:	MS ACESS
Operating System	:	WindowsNT/95/98/2000
RAM	:	256MB

5.2 Hard ware Specification:

Processor	:	Intel P-III based system
Processor Speed	:	250 MHz to 833MHz
RAM	:	64MB to 256MB
Hard Disk	:	2GB to 30GB
Key Board	:	104 keys

6.LITERATURE SURVEY

Visual Basic.NET

Introduction to Windows Forms (Visual Basic.NET)

Windows Forms is the new platform for Microsoft Windows application development, based on the .NET Framework. This framework provides a clear, object-oriented, extensible set of classes that enable you to develop rich Windows applications. Additionally, Windows Forms can act as the local user interface in a multi-tier distributed solution. Windows Forms is a framework for building Windows client applications that utilize the common language runtime. Windows Forms applications can be written in any language that the common language runtime supports.

What Is a Form?

A form is a bit of screen real estate, usually rectangular, that you can use to present information to the user and to accept input from the user. Forms can be standard windows, multiple document interface (MDI) windows, dialog boxes, or display surfaces for graphical routines. The easiest way to define the user interface for a form is to place controls on its surface. Forms are objects that expose properties which define their appearance, methods which define their behavior, and events which define their interaction with the user. By setting the properties of the form and writing code to respond to its events, you customize the object to meet the requirements of your application.

As with all objects in the .NET Framework, forms are instances of classes. The form you create with the Windows Forms Designer is a class, and when you display an instance of the form at run time, this class is the template used to create the form. The framework also allows you to inherit from existing forms to add functionality or modify existing behavior. When you add a form to your project, you can choose whether it inherits from the **Form** class provided by the framework, or from a form you have previously created.

Additionally, forms are controls, because they inherit from the Control class.

Within a Windows Forms project, the form is the primary vehicle for user interaction. By combining different sets of controls and writing code, you can elicit information from the user and respond to it, work with existing stores of data, and query and write back to the file system and registry on the user's local computer.

Although the form can be created entirely in the Code Editor, it is easier to use the Windows Forms Designer to create and modify forms.

Some of the advantages of using Windows Forms include the following:

- **Simplicity and power:** Windows Forms is a programming model for developing Windows applications that combines the simplicity of the Visual Basic 6.0 programming model with the power and flexibility of the common language runtime.
- **Lower total cost of ownership:** Windows Forms takes advantage of the versioning and deployment features of the common language runtime to offer reduced deployment costs and higher application robustness over time. This significantly lowers the maintenance costs (TCO) for applications written in Windows Forms.

- **Architecture for controls:** Windows Forms offers an architecture for controls and control containers that is based on concrete implementation of the control and container classes. This significantly reduces control-container interoperability issues.
- **Security:** Windows Forms takes full advantage of the security features of the common language runtime. This means that Windows Forms can be used implement everything from an untrusted control running in the browser to a fully trusted application installed on a user's hard disk.
- **XML Web services support:** Windows Forms offers full support for quickly and easily connecting to XML Web services.
- **Rich graphics:** Windows Forms is one of the first ship vehicles for GDI+, a new version of the Windows Graphical Device Interface (GDI) that supports alpha blending, texture brushes, advanced transforms, rich text support, and more.
- **Flexible controls:** Windows Forms offers a rich set of controls that encompass all of the controls offered by Windows. These controls also offer new features, such as "flat look" styles for buttons, radio buttons, and check boxes.
- **Data awareness:** Windows Forms offers full support for the ADO data model.
- **ActiveX control support:** Windows Forms offers full support for ActiveX controls. You can easily host ActiveX controls in a Windows Forms application. You can also host a Windows Forms control as an ActiveX control.
- **Licensing:** Windows Forms takes advantage of the common language runtime enhanced licensing model.

- **Printing:** Windows Forms offers a printing framework that enables applications to provide comprehensive reports.
- **Accessibility:** Windows Forms controls implement the interfaces defined by Microsoft Active Accessibility (MSAA), which make it simple to build applications that support accessibility aids, such as screen readers.
- **Design-time support:** Windows Forms takes full advantage of the meta-data and component model features offered by the common language runtime to provide thorough design-time support for both control users and control implementers.

Crystal Reports

Crystal Reports for Visual Basic .NET is the standard reporting tool for Visual Basic.NET; it brings the ability to create interactive, presentation-quality content — which has been the strength of Crystal Reports for years — to the .NET platform.

With Crystal Reports for Visual Basic.NET, you can host reports on Web and Windows platforms and publish Crystal reports as Report Web Services on a Web server.

To present data to users, you could write code to loop through recordsets and print them inside your Windows or Web application. However, any work beyond basic formatting can be complicated: consolidations, multiple level totals, charting, and conditional formatting are difficult to program.

With Crystal Reports for Visual Studio .NET, you can quickly create complex and professional-looking reports. Instead of coding, you use the Crystal Report Designer interface to create and format the report you need. The powerful Report Engine processes the formatting, grouping, and charting criteria you specify.

Report Experts

Using the Crystal Report Experts, you can quickly create reports based on your development needs:

- Choose from report layout options ranging from standard reports to form letters, or build your own report from scratch.
- Display charts that users can drill down on to view detailed report data.
- Calculate summaries, subtotals, and percentages on grouped data.
- Show TopN or BottomN results of data.
- Conditionally format text and rotate text objects.

ACTIVE X DATA OBJECTS

In Visual Basic .Net, three data access interfaces are available: Active X Data Objects(ADO), Remote Data Objects (RDO) and Data Access Objects (DAO). These access interfaces are used to access the data from database.

Why use ADO?

Consistently accessing data within the enterprise is a challenge for today's business applications. ODBC provides the first step toward overcoming this challenge by enabling applications to access relational databases. However, as developers and system architects want to include nonrelational data sources and to work in environments such as the Internet, they encounter the dilemma of either developing their own data-access paradigms or working with application program interfaces (APIs) that are incompatible in the new environments. Microsoft® ActiveX® Data Objects (ADO) along with OLEDB solves this dilemma by providing a single model that works with all data sources in a variety of environments.

ADO provides consistent, high-performance access to data, whether you're creating a front-end database client or middle-tier business object using an application, tool, language, or even an Internet browser. ADO is the single data interface you need for developing 1- to *n*-tier client/server and Web-based, data-driven solutions.

This paper introduces ADO and the ADO programming model for application developers who are targeting Microsoft SQL Server™. Particular attention is given to taking advantage of SQL Server features with ADO, such as stored procedures and server cursors. The concepts presented in the sections titled "The ADO Object Model" and "Using ADO with Visual Basic, VBScript, Visual C++, and Java" are applicable to all ADO programmers.

ADO Overview

ADO was first introduced as the data access interface in Microsoft Internet Information Server (IIS). ADO is easy to use because it is called using a familiar metaphor: the Automation interface, available from just about any tool and language on the market today. Because of its popularity as an easy-to-use, lightweight interface to all kinds of data, and the growing need for an interface spanning many tools and languages, ADO is being enhanced to combine the best features of, and eventually replace, RDO and DAO, the data access interfaces in widest use today. ADO is in many ways similar to RDO and DAO. For example, it uses similar language conventions. ADO provides simpler semantics, which makes it easy to learn for today's developers.

ADO is designed to be the application-level interface to OLEDB, Microsoft's newest and most powerful data access paradigm. OLEDB provides high-performance access to any data source. Together ADO and OLEDB form the foundation of the Universal Data Access strategy. OLEDB enables universal access to any data. ADO makes it easy for developers to program. Because ADO is built on top of OLEDB, it benefits from the rich universal data access infrastructure that OLEDB provides.

OLEDB Overview

OLEDB is an open specification designed to build on the success of ODBC by providing an open standard for accessing all kinds of data throughout the enterprise. OLEDB is a core technology supporting universal data access. Whereas ODBC was created to access relational databases, OLEDB is designed for the relational and nonrelational information sources, such as mail stores, text and graphical data for the Web, directory services, and IMS and VSAM data stored in the mainframe. OLEDB components consist of data providers, which expose data; data consumers, which use data; and service components, which process and transport data (for example, query processors and cursor engines). These components are designed to integrate smoothly to help OLEDB component vendors quickly bring high-quality OLEDB components to market. OLEDB includes a bridge to ODBC to enable continued support for the broad range of ODBC relational database drivers available today.

OLEDB Providers

There are two types of OLEDB applications: consumers and providers. A consumer can be any application that uses or consumes OLEDB interfaces. For example, a Microsoft Visual C++® application that uses OLEDB interfaces to connect to a database server is an OLEDB consumer. The ADO object model that uses OLEDB interfaces is an OLEDB consumer. Any application that uses the ADO object model uses OLEDB interfaces indirectly through the

ADO objects. An OLEDB provider implements OLEDB interfaces; therefore, an OLEDB provider allows consumers to access data in a uniform way through a known set of documented interfaces. In a sense, an OLEDB provider is similar to an ODBC driver that provides a uniform mechanism for accessing relational data. OLEDB providers not only provide a mechanism for relational data but also for nonrelational types of data. Furthermore, OLEDB providers are built on top of Component Object Model (COM) interfaces that allow more flexibility; whereas ODBC drivers build on top of a C API specification.

Microsoft OLEDB SDK version 1.1 shipped two OLEDB providers: the ODBC Provider and sample text provider. The sample text provider is an example that demonstrates the implementation detail of an OLEDB provider. The ODBC Provider is an OLEDB provider for ODBC drivers. This provider enables consumers to use the existing ODBC drivers without having to implement new OLEDB providers to replace existing ODBC drivers. With OLEDB version 2.0, providers for SQL Server, Oracle data, and Microsoft Jet databases were added to the SDK. For more information about OLEDB and OLEDB providers, see the OLEDB section of the Microsoft Data Access

The ODBC Provider

The ODBC Provider maps OLEDB interfaces to ODBC APIs. With the ODBC Provider, OLEDB consumers can connect to a database server through the existing ODBC drivers in the following process: A consumer calls an OLEDB interface on the ODBC Provider. The ODBC Provider invokes corresponding ODBC APIs and sends the requests to an ODBC driver.

Because the ODBC Provider allows OLEDB consumers to use existing ODBC drivers, there may be some performance concern about the additional layer of the ODBC Provider on top of the existing ODBC driver manager. The design goal of the ODBC Provider is to implement all the functionality of the ODBC driver manager; therefore, the ODBC driver manager is not needed. However, the ODBC Provider still requires the ODBC Driver Manager to support connection pooling with ODBC applications.

The ADO Object Model

The ADO object model defines a collection of programmable objects that you can use in Visual Basic, Visual C++, Microsoft Visual Basic, Scripting Edition, Java, and any platform that supports both COM and Automation. The ADO object model is designed to expose the most commonly used features of OLEDB.

The ADO object model contains their objects:

- Connection
- Command
- Record set

The Connection Object

The **Connection** object allows you to establish a communication link with a data source. The **Connection** object provides a mechanism for initializing and establishing the connection, executing queries, and using transactions.

The underlying OLEDB provider used for connecting is not limited to the ODBC Provider; you can also use other providers for connecting. Specify a provider through the **Provider** property. If none is specified, MSDASQL (the ODBC provider) is the default provider used for the connection.

The Command Object

The **Command** object allows you to issue commands to the database. These commands can be, but are not limited to, query strings, prepared query strings, and associated parameters with query strings. The actual command language and features supported are dependent on the underlying provider for the database. The information and examples contained here focus on the Microsoft ODBC Provider that supports a wide variety of relational databases.

The Recordset Object

The **Recordset** object provides methods for manipulating result sets; it allows you to add, update, delete, and scroll through records in the result set. You can retrieve and update each record using the **Fields** collection and the **Field** objects. You can make updates on the **Recordset** object in an immediate or batch mode. When you create a **Recordset** object, a cursor is automatically opened.

Dynamic Cursor

Allows you to view additions, changes and deletions by other users, and allows all types of movement through the records that don't rely on bookmarks; Allows bookmarks if the provider supports them.

Key-set Cursor

Behaves like a dynamic cursor, except that it prevents you from seeing records that other users ad, and prevents access to records that other users delete. Data changes by other users will still be visible. It always supports bookmarks and therefore
Allows all types of movement through the Records.

Static Cursor

Provides a static copy of a set of records for you to use to find or generate reports. Always allows bookmarks and therefore allows all types of movement through the records. Additions, changes or deletions by other users will not be visible. This is the only type of cursor allowed when you open a client_side (ADO) records object.

Forward-only Cursor

Behaves identically to a dynamic cursor except that it allows you to scroll only forward through records. This improves performance in situation where you need to make only a single pass through a record.

Understanding the advantages of VISUAL BASIC

The following key points make Visual Basic an excellent development tool: Visual Basic applications are event – driven. Even driven means the user is in control of the application. The user generates a stream of events each time he or she clicks with the mouse or pressed a key on the keyboard. Your Visual Basic application responds to those events through the code you've written and attached to those events.

Visual Basic supports the principle of object-oriented design. This means that you can compartmentalize different aspects of your application as objects and develop and test those objects independently of the rest of the application. By modifying certain properties and invoking the methods of these objects, you exert a great deal of control over the user's interaction with the visual basic you've written. Although Visual Basic does not comply with all the concepts and principles behind the object – oriented development model supported by visual basic includes enough of these features to more than satisfy all but most ardent followers of object oriented programming

Microsoft has designed Visual Basic to be complete Windows Application Development system. This means that you're Visual Basic Application will look and behave like other.

Windows programs your users might work with. In other words, you Visual Basic applications will conform to the windows – 95 look and feel without any extra work on your part Unlike visual c++ and other development platforms you don't have to go to extreme measures to employ even the most sophisticated window features in your visual basic application.

Visual Basic is infinitely extensible through the use of Active X controls, dynamically linked libraries (DLL's) and add-ins. You can create these Active X controls, DLL's and add-ion with Visual basic 6 or buy them off the shelf from a large number of third party software vendors. In fact, one of the influences driving the rapid adoption of visual basic 6 is the desire of many developers to create active X controls and DLL's for use in other windows applications such as excel, word or Access.

Visual basic offers the fastest and easiest way to create application for windows. It is a complete set of tools that simplify raid application development for both experienced professional and novice window programmers. The visual in refers to the method used to create the GUI, rather than writing numerous line of code to describe the appearance and location of interface elements, pre built objects are dragged and dropped into place on screens. "BASIC" refers to the Basic programming Language VB evolved from the original Basic Language and how contains several hundred statements, functions, and Keyboard many of which are related directly to the windows GUI.

FEATURES

- Rapid – application development
- Database features to create fast, high performance applications and components
- Team development and scalability features with Microsoft visual Modeler, Visual Database Tools, Visual Source safe and SQL server.
- The ability to create reusable Active X components for traditional Client/server Architecture, Internet, and Microsoft Transaction Server.
- Native code compiles that can be optimized for speed, size and even for Pentium preprocessor's.
- A common programming language for all Microsoft Office applications. Visual basic is providing to be most powerful and flexible way of developing for GUI's as a window-programming tool Database Management as a front-end tool Internet as web programming tool.

Error Finding Methods

The debugging facility in the Visual Basic is also quite advanced. I have the immediate window where in at runtime I apply break and change the values in the variables. I can set break points and thus I come to know the flow and locations of errors in the code. Even as I type the code I have a runtime list that lists out the syntax and prevents us from making syntactical errors. I have used a lot of debugging methods in finding errors, there are different methods which exists but the one I frequently used was the line by line tracing of the program at runtime this is done by pressing F8 key, I also used the immediate window which shows the results of the expressions at the run time. Watch windows also helped me little. These facilities also helped Visual Basic 6.0 tool to be good.

7 SYSTEM DESGIN

7.1 DATA BASE DESGIN

DATABASE

DATABASE

A database is a set of data, organized for easy access. The database is an actual data; it is the database that you will be accessing when you need to retrieve data.

DATA DICTIONARY

The data dictionary is a set of tables Oracle uses to maintain information about the database. The data dictionary contains information about tables, indexes, clusters and so on.

DBA (DATABASE ADMINISTRATOR)

The DBA is the person responsible for the operation, configuration and performance of the database. The DBA is charged with keeping the database operating smoothly, ensuring that backups are done on regular basis (and that backups work), and installing new software. Other responsibilities might include planning for future expansion and disk space needs, creating databases and table spaces, adding users and maintaining security, and monitoring the database and retuning it as necessary. Large installations might have teams of DBA’s to keep the system running smoothly; alternatively, the task might be segmented among the DBA’s.

7.1.1UML DIAGRAMS

UNIFIED MODELING LANGUAGE

UML is the international standard notation for object-oriented analysis and design. The Object Management Group defines it. The heart of object-oriented problem solving is the construction of a model. The model abstracts the essential details of the underlying problem from its usually complicated real world. Several modeling tools are wrapped under the heading of the **UML**™, which stands for Unified Modeling Language™.

AN OVERVIEW OF UML:

The UML is a language for

- Visualizing
- Specifying
- Constructing
- Documenting

These are the artifacts of a software-intensive system. The three major elements of UML are

- The UML's basic building blocks
- The rules that dictate how those building blocks may be put together.
- Some common mechanisms that apply throughout the UML.

BASIC BUILDING BLOCKS OF THE UML:

The vocabulary of UML encompasses three kinds of building blocks:

- Things
- Relationships
- Diagrams

Things are the abstractions that are first-class citizens in a model.
Relationships tie these things together.
Diagrams group the interesting collection of things.

THINGS IN THE UML:

They are the abstractions that are first-class citizens in a model. There are four kinds of things in the UML

1. Structural things
2. Behavioral things.
3. Grouping things.
4. Annotational things.

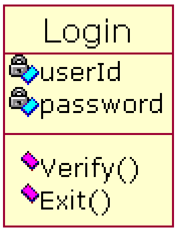
These things are the basic object oriented building blocks of the UML. They are used to write well-formed models.

STRUCTURAL THINGS:

Structural things are the nouns of the UML models. These are mostly static parts of the model, representing elements that are either conceptual or physical. In all, there are seven kinds of Structural things.

CLASS:

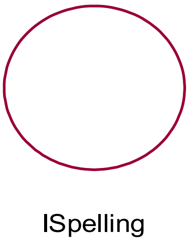
A class is a description of a set of objects that share the same attributes, operations, relationships, and semantics. A class implements one or more interfaces. Graphically a class is rendered as a rectangle, usually including its name, attributes and operations, as shown below.



INTERFACE:

An interface is a collection of operations that specify a service of a class or component. An interface describes the externally visible behavior of that element.

Graphically the interface is rendered as a circle together with its name.



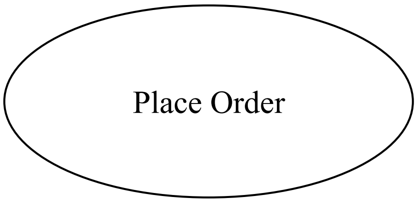
COLLABORATION:

Collaboration defines an interaction and is a society of roles and other elements that work together to provide some cooperative behavior that's bigger than the sum of all the elements. Graphically, collaboration is rendered as an ellipse with dashed lines, usually including only its name as shown below.

Chain of
Responsibili
ty

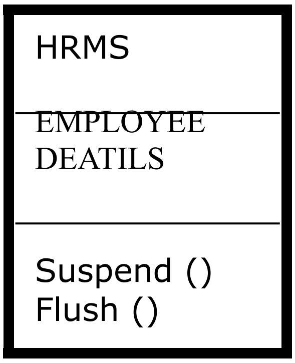
USE CASE:

Use case is a description of a set of sequence of actions that a system performs that yields an observable result of value to a particular thing in a model. Graphically, Use Case is rendered as an ellipse with dashed lines, usually including only its name as shown below.



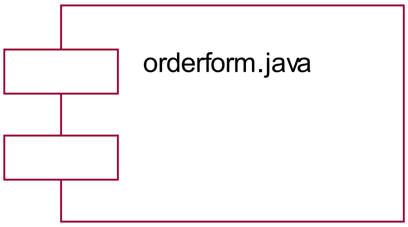
ACTIVE CLASS:

An active class is a class whose objects own one or more processes or threads and therefore can initiate control activity. Graphically, an active class is rendered just like a class, but with heavy lines usually including its name, attributes and operations as shown below.



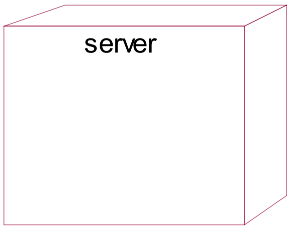
COMPONENT:

Component is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces. Graphically, a component is rendered as a rectangle with tabs, usually including only its name, as shown below.



NODE:

A Node is a physical element that exists at run time and represents a computational resource, generally having at least some memory and often, processing capability. Graphically, a node is rendered as a cube, usually including only its name, as shown below.



BEHAVIORAL THINGS:

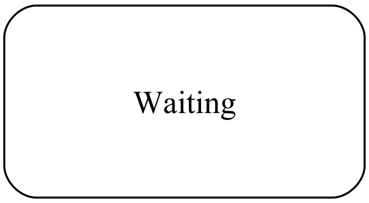
Behavioural Things are the dynamic parts of UML models. These are the verbs of a model, representing behaviour over time and space.

INTERACTION:

An interaction is a behavior that comprises a set of messages exchanged among a set of objects within a particular context to accomplish a specific purpose. Graphically, a message is rendered as a direct line, almost always including the name if its operation, as shown below.

STATE MACHINE:

A state machine is a behavior that specifies the sequence of states an object are an interaction goes through during its lifetime on response to events, together with its responses to those events. Graphically, a state is rendered as a rounded rectangle usually including its name and its sub-states, if any, as shown below.

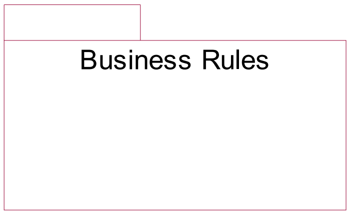


GROUPING THINGS:

Grouping things are the organizational parts of the UML models. These are the boxes into which a model can be decomposed.

PACKAGE:

A package is a general-purpose mechanism for organizing elements into groups.



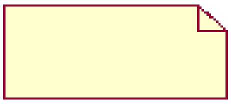
ANNOTATIONAL THINGS:

Annotational things are the explanatory parts of the UML models.

Note:

A note is simply a symbol for rendering constraints and comments attached to an element or a collection of elements.

Graphically a note is rendered as a rectangle with dog-eared corner together, with a textual or graphical comment, as shown below.



RELATIONSHIPS IN THE UML:

There are four kinds of relationships in the UML:

1. Dependency
2. Association
3. Generalization
4. Realization

1.DEPENDENCY:

This is relationship between two classes whenever one class is completely dependent on the other class. Graphically the dashed line represents it with arrow pointing to the class that it is being depended on.

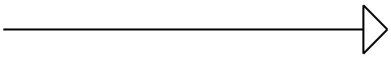
-------------------------->

2._ASSOCIATION:

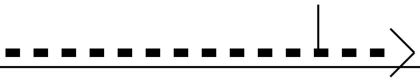
It is a relationship between instances of the two classes. There is an association between two classes if an instance of one class must know about the other in order to perform its work. In a diagram, an association is a link connecting two classes. Graphically it is represented by line as shown.

3. GENERALIZATION:

An inheritance is a link indicating one class is a super class of the other. A generalization has a triangle pointing to the super class. Graphically it is represented by line with a triangle at end as shown.



4. REALIZATION:



DIAGRAMS IN UML:

Diagrams play a very important role in the UML. There are nine kind of modeling diagrams as follows:

- Use Case Diagram
- Class Diagram
- Object Diagram
- Sequence Diagram
- Collaboration Diagram
- State Chart Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram

CLASS DIAGRAM:

Class diagrams are the most common diagrams found in modeling object-oriented systems. A class diagram shows a set of classes, interfaces, and collaborations and their relationships. Graphically, a class diagram is a collection of vertices and arcs.

Contents:

Class Diagrams commonly contain the following things:

Classes

Interfaces

Collaborations

Dependency, generalization and association relationships

USE CASES DIAGRAM:

Use Case diagrams are one of the five diagrams in the UML for modeling the dynamic aspects of systems(activity diagrams, sequence diagrams, state chart diagrams and collaboration diagrams are the four other kinds of diagrams in the UML for modeling the dynamic aspects of systems). Use Case diagrams are central to modeling the behavior of the system, a sub-system, or a class. Each one shows a set of use cases and actors and relationships.

COMMON PROPERTIES:

A Use Case diagram is just a special kind of diagram and shares the same common properties, as do all other diagrams- a name and graphical contents that are a projection into the model. What distinguishes a use case diagram from all other kinds of diagrams is its particular content.

Contents

Use Case diagrams commonly contain:

Use Cases

Actors

Dependency, generalization, and association relationships

Like all other diagrams, use case diagrams may contain notes and constraints. Use Case diagrams may also contain packages, which are used to group elements of your model into larger chunks. Occasionally, you will want to place instances of use cases in your diagrams, as well, especially when you want to visualize a specific executing system.

INTERACTION DIAGRAMS

An Interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them. Interaction diagrams are used for modeling the dynamic aspects of the system.

A sequence diagram is an interaction diagram that emphasizes the time ordering of the messages. Graphically, a sequence diagram is a table that shows objects arranged along the X-axis and messages, ordered in increasing time, along the Y-axis and messages, ordered in increasing time, along the Y-axis.

Contents

Interaction diagrams commonly contain:

Objects

Links

Messages

Like all other diagrams, interaction diagrams may contain notes and constraints.

SEQUENCE DIAGRAMS:

A sequence diagram is an interaction diagram that emphasizes the time ordering of the messages. Graphically, a sequence diagram is a table that shows objects arranged along the X-axis and messages, ordered in increasing time, along the Y-axis.

Typically you place the object that initiates the interaction at the left, and increasingly more sub-routine objects to the right. Next, you place the messages that these objects send and receive along the Y-axis , in order of increasing time from top to the bottom. This gives the reader a clear visual cue to the flow of control over time.

Sequence diagrams have two interesting features:

1. There is the object lifeline. An object lifeline is the vertical dashed line that represents the existence of an object over a period of time. Most objects that appear in the interaction diagrams will be in existence for the duration of the interaction, so these objects are all aligned at the top of the diagram, with their lifelines drawn from the top of the diagram to the bottom.

2. There is a focus of the control. The focus of control is tall, thin rectangle that shows the period of time during which an object is performing an action, either directly or through the subordinate procedure. The top of the rectangle is aligns with the action; the bottom is aligned with its completion.

Contents

Sequence diagrams commonly contains

Objects

Object Life Line

Focus of Control

ACTIVITY DIAGRAM

An Activity Diagram is essentially a flow chart showing flow of control from activity to activity. They are used to model the dynamic aspects of as system. They can also be used to model the flow of an object as it moves from state to state at different points in the flow of control.

An activity is an ongoing non-atomic execution with in a

State machine. Activities ultimately result in some action, which is made up of executable atomic computations that result in a change of state of distinguishes a use case diagram from all other kinds of diagrams is its particular content.

Contents

Activity diagrams commonly contain:
Fork
Start & End Symbol

STATE CHART DIAGRAMS

A state chart diagram shows a state machine. State chart diagrams are used to model the dynamic aspects of the system. For the most part this involves modeling the behavior of the reactive objects. A reactive object is one whose behavior is best characterized by its response to events dispatched from outside its context. A reactive object has a clear lifeline whose current behavior is affected by its past.

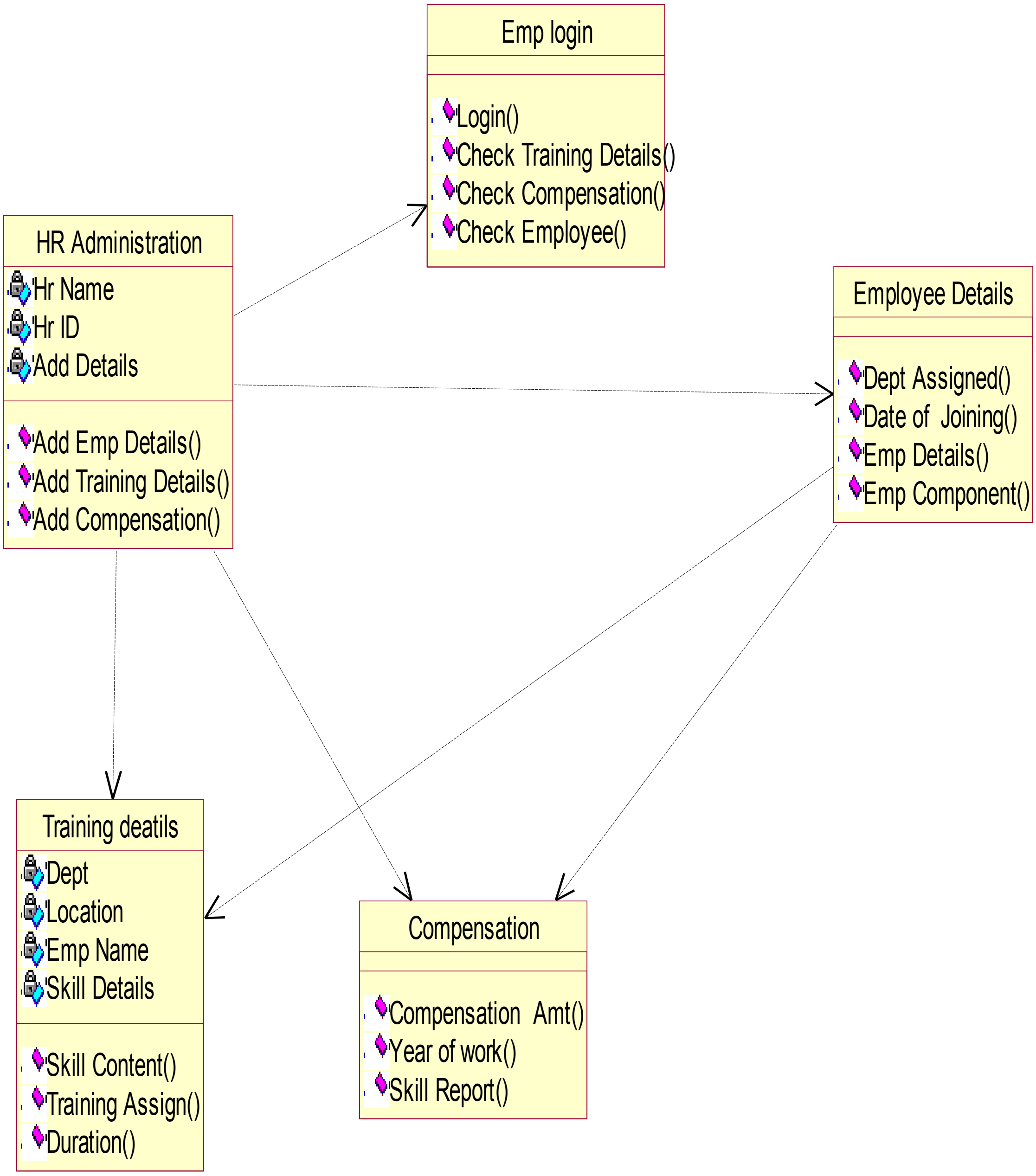
A state chart diagram show a state machine emphasizing the flow of control from state to state. A state machine is a behavior that specifies the sequence of states an object goes through during its lifetime in response to events together with its
Response to those events. A state is a condition in the life of the object during which it satisfies some conditions, performs some activity or wait for some events. An event is a specification of a significant occurrence that has a location in time and space.
Graphically a state chart diagram is a collection of vertices and arcs.

Contents:

State chart diagram commonly contain:
Simple states and Composite states.
Transitions, including events and actions.

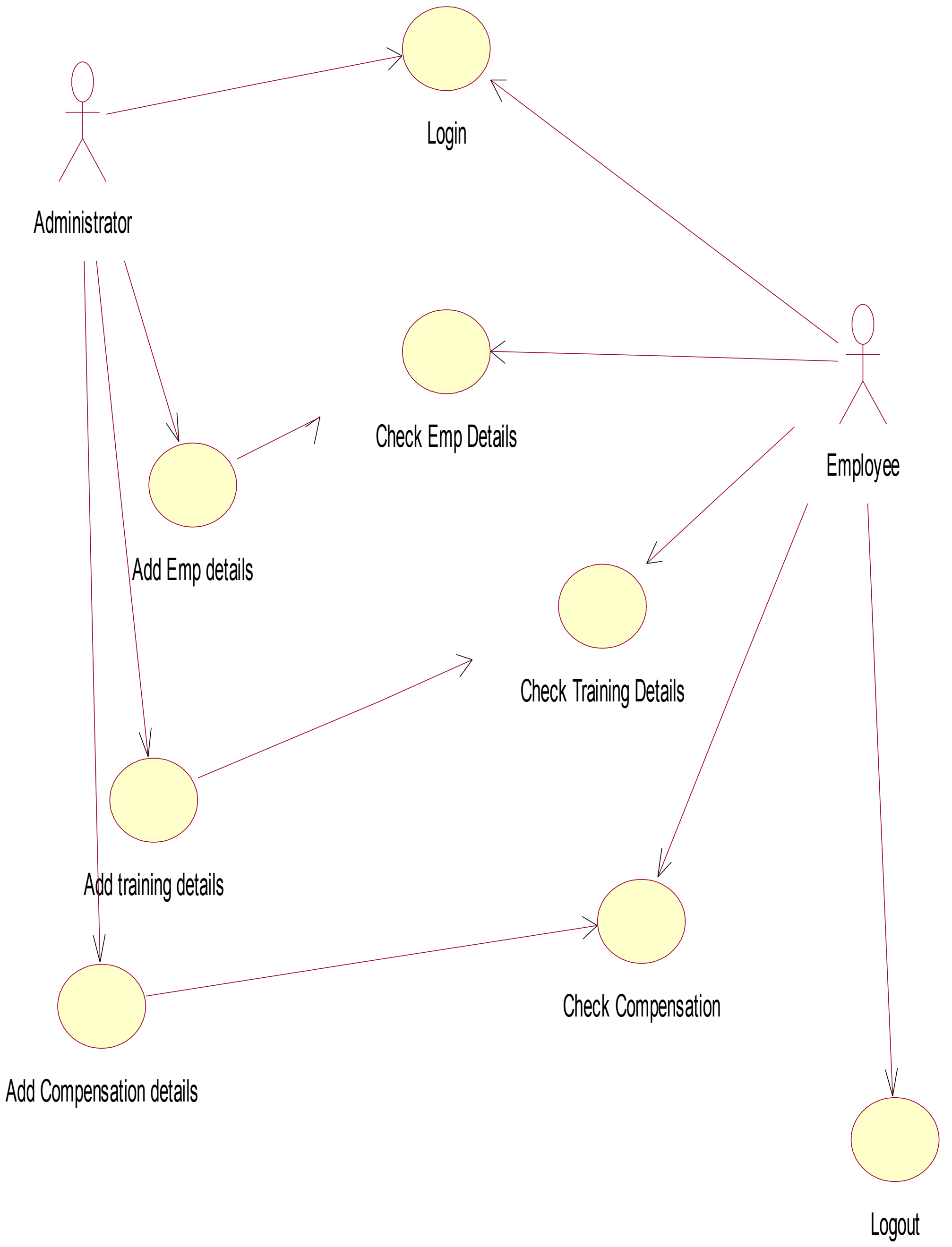
7.2 DATA BASE DESIGN

CLASS DIAGRAM

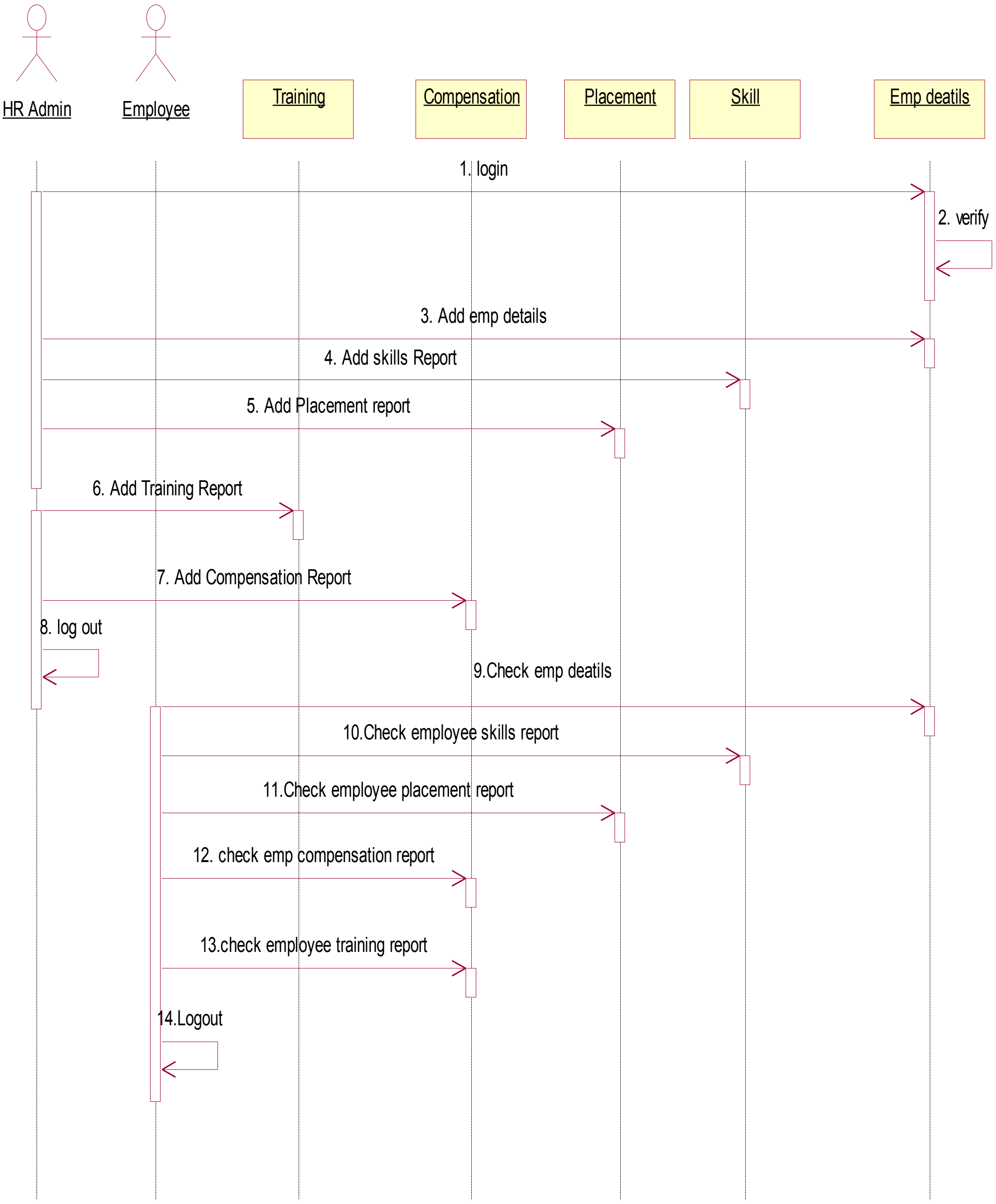


OBJECT DIAGRAM

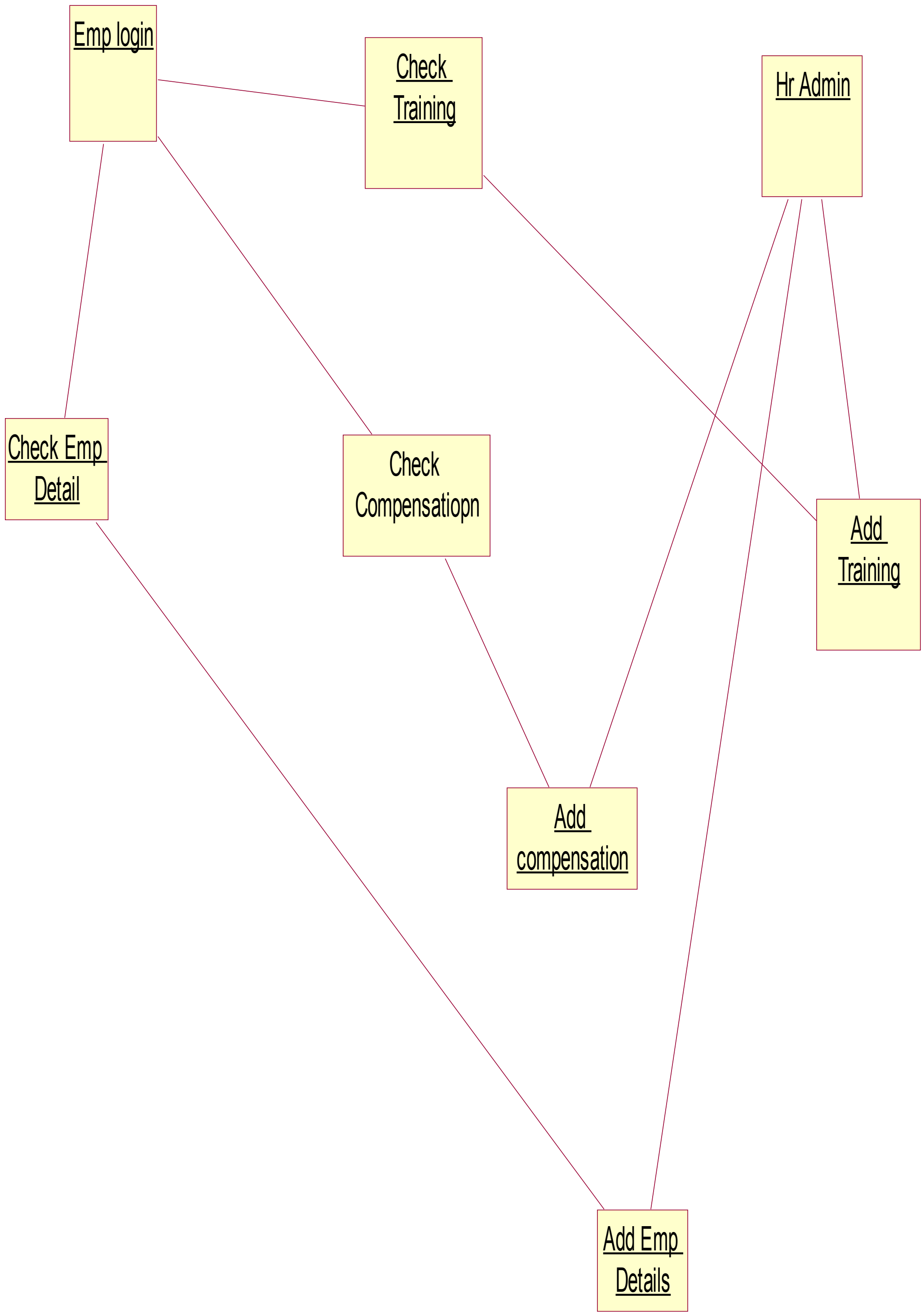
USE CASE DIAGRAM



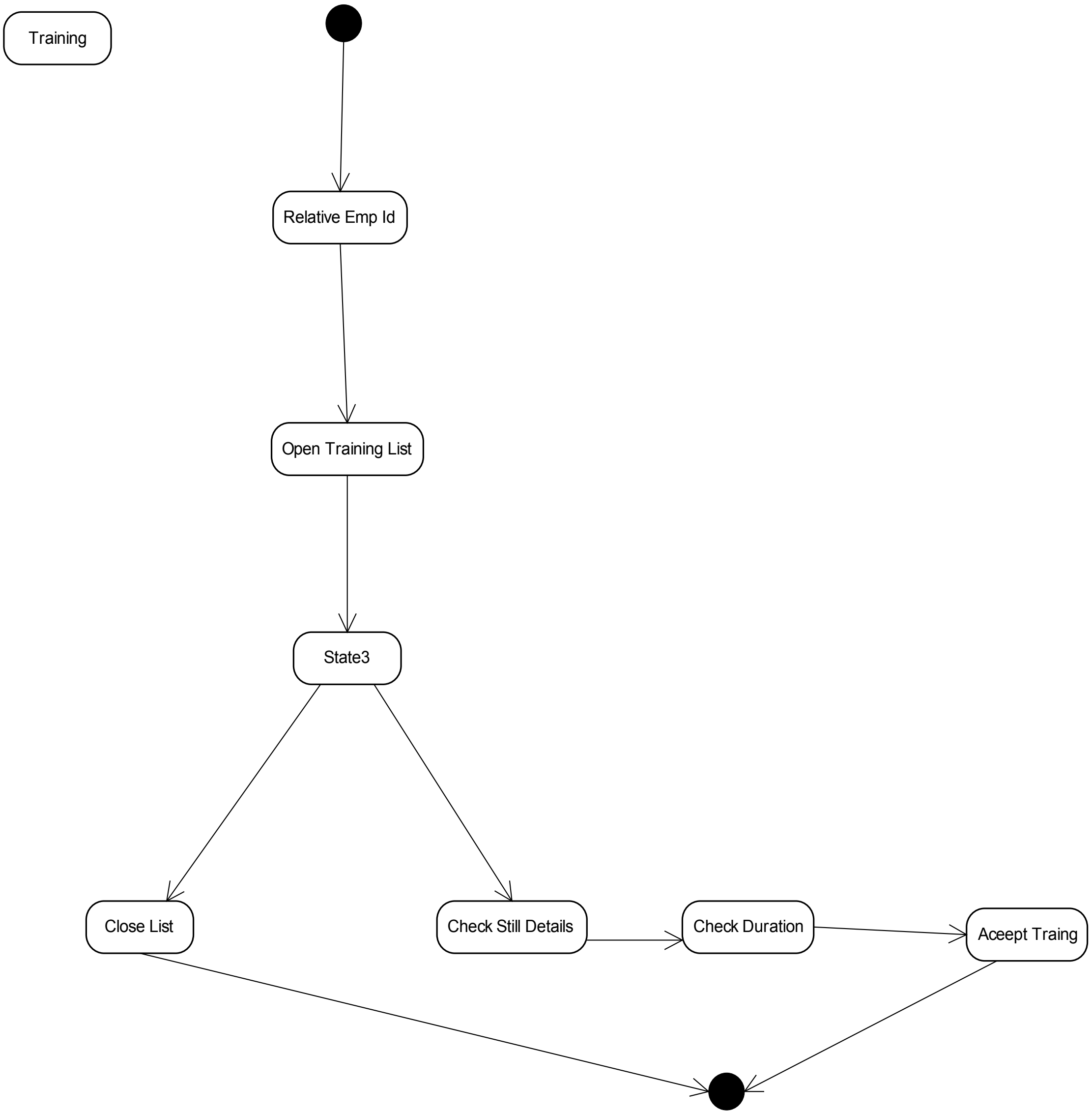
SEQUENCE DIAGRAM



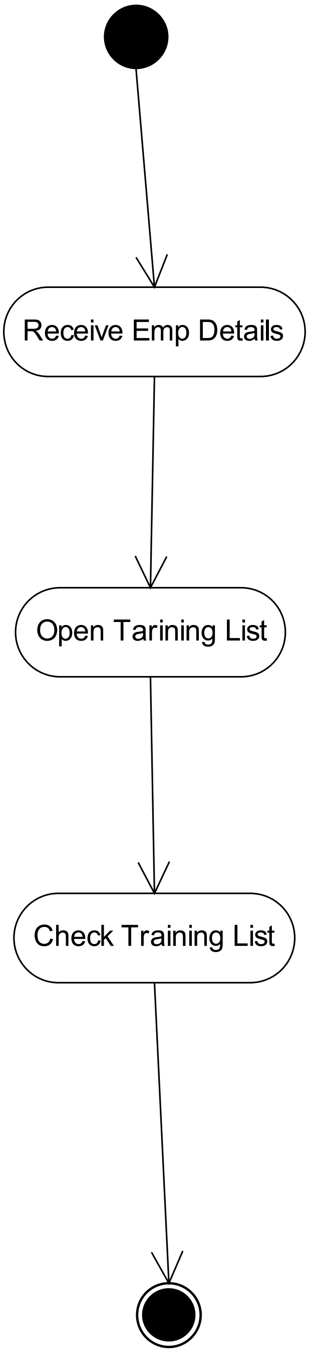
COLLABORATION DIAGRAM

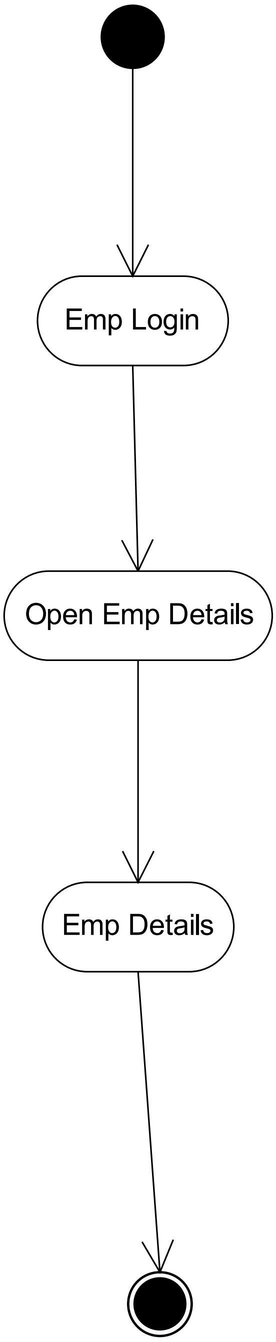


STATE CHART DIAGRAM

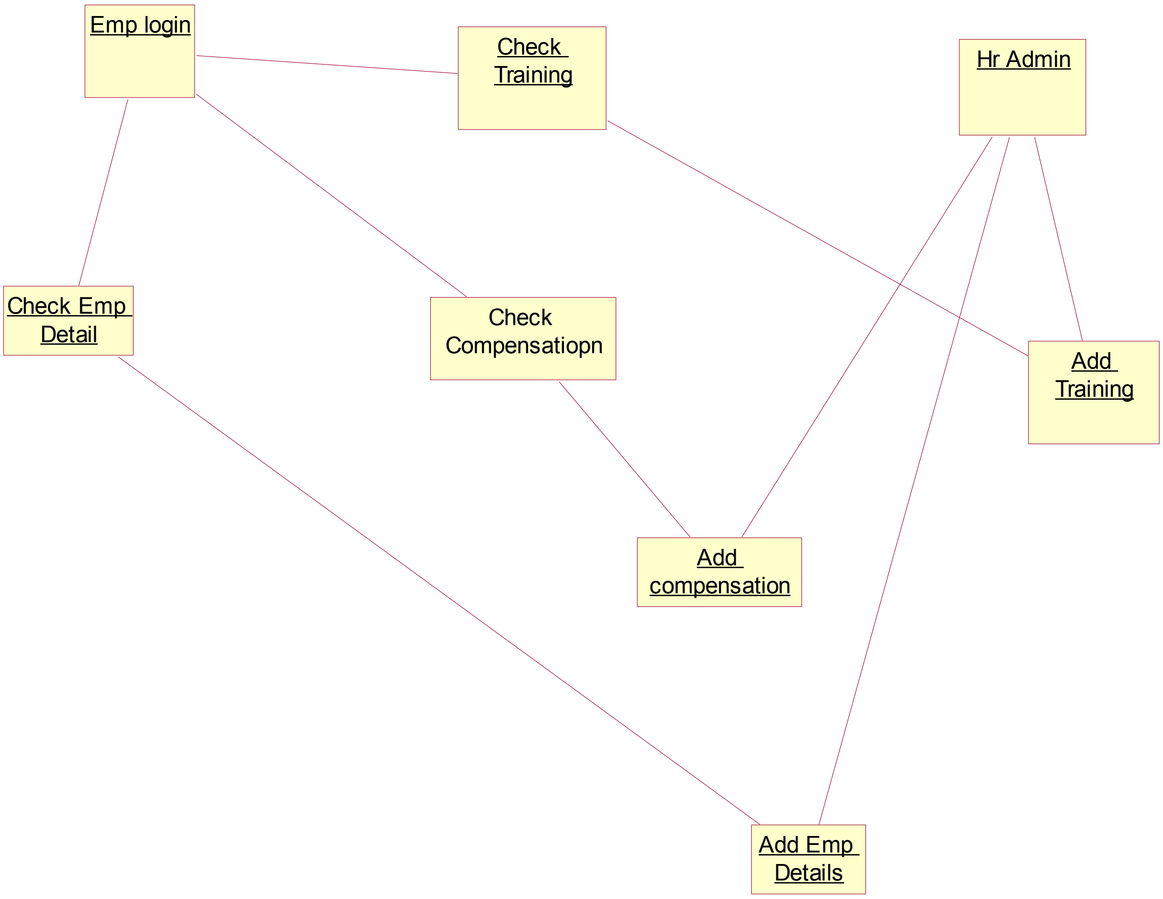


ACTIVITY DIAGRAM

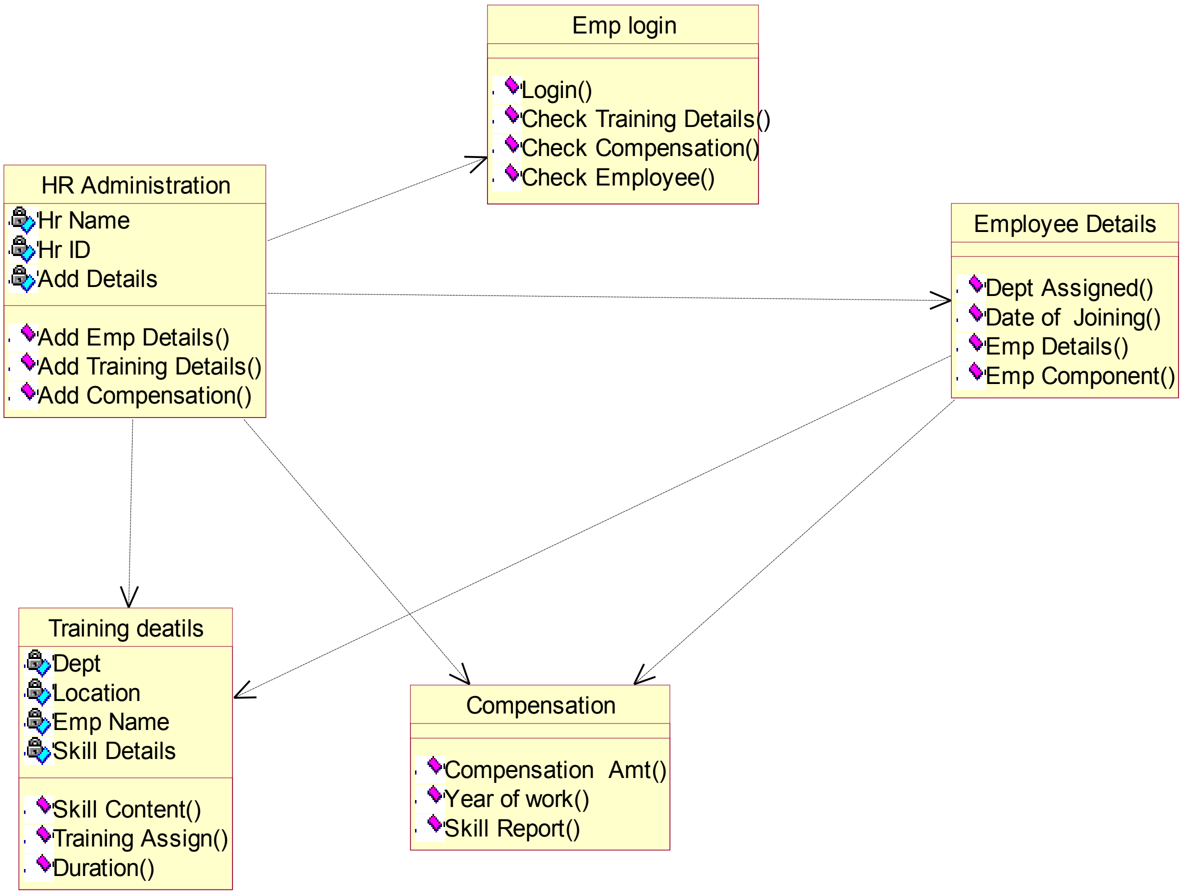




COMPONENT DIAGRAM



CLASS DIAGRAM



7.2.1 DATA FLOW DIAGRAMS

DATA FLOW DIAGRAMS:

A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also know as a data flow graph or a bubble chart.

CONTEXT DIAGRAM:

The top-level diagram is often called a *"context diagram"*. It contains a single process, but it plays a very important role in studying the current system. The context diagram defines the system that will be studied in the sense that it determines the boundaries. Anything that is not inside the process identified in the context diagram will not be part of the system study. It represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows respectively.

TYPES OF DATA FLOW DIAGRAMS:

Data Flow Diagrams are of two types as follows:

(a)Physical DFD

(b)Logical DFD

1. PHYSICAL DFD:

Structured analysis states that the current system should be first understand correctly. The physical DFD is the model of the current system and is used to ensure that the current system has been clearly understood. Physical DFDs shows actual devices, departments, and people etc., involved in the current system

2. LOGICAL DFD:

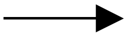
Logical DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system's structure charts.

BASIC NOTATION:

The Basic Notation used to create a DFD's are as follows:

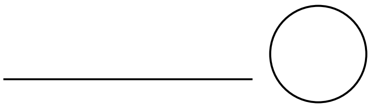
DATAFLOW:

Data move in a specific direction from an origin to a destination.



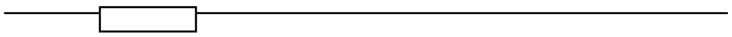
PROCESS

People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.__



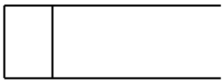
SOURCE:

External sources or destination of data, which may be People, programs, organizations or other entities.



DATA STORE:

Here data are stored or referenced by a process in the System

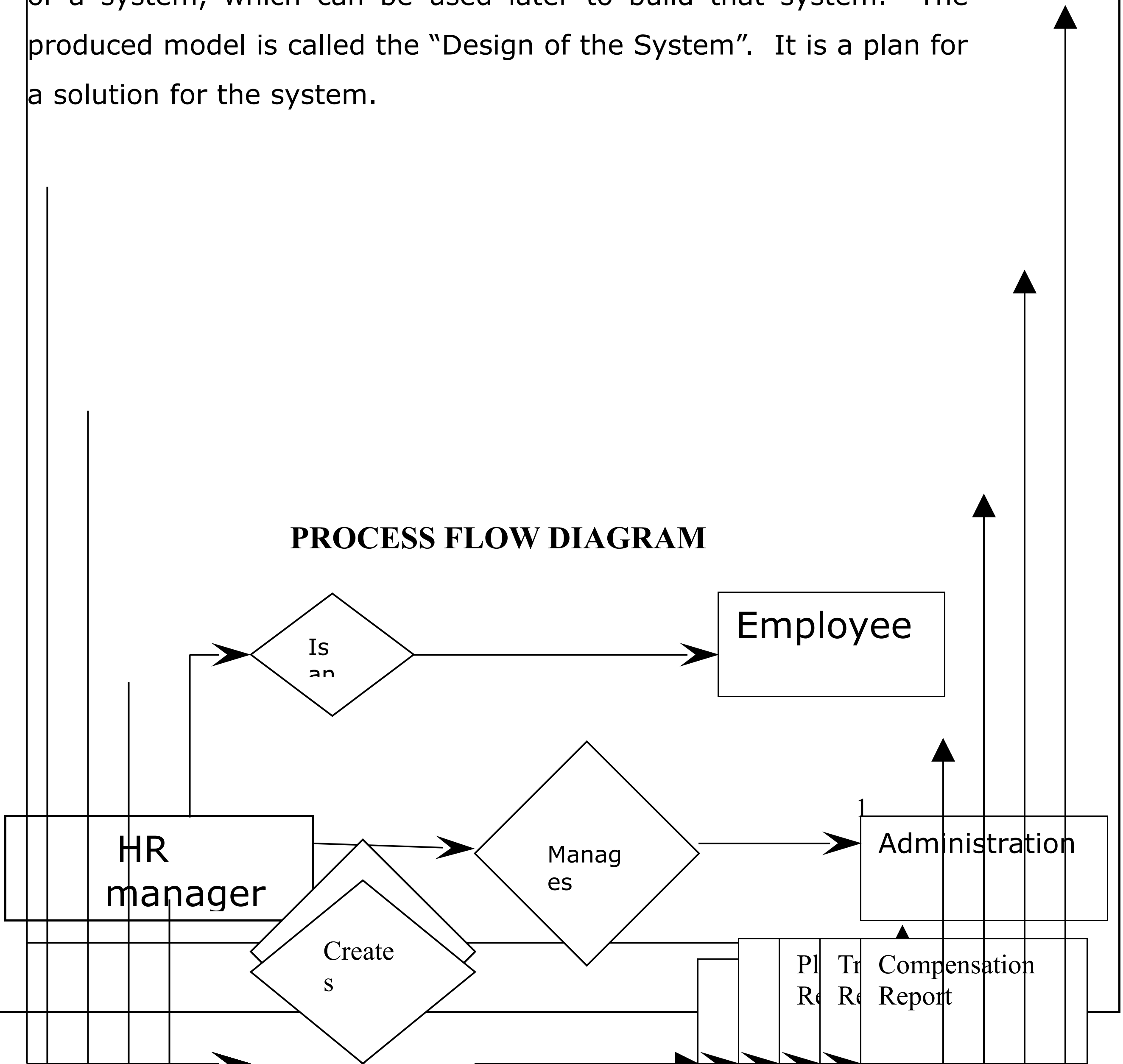


DESIGN:

Design is the first step in moving from problem domain to the solution domain. Design is essentially the bridge between requirements specification and the final solution.

The goal of design process is to produce a model or representation of a system, which can be used later to build that system. The produced model is called the "Design of the System". It is a plan for a solution for the system.

PROCESS FLOW DIAGRAM

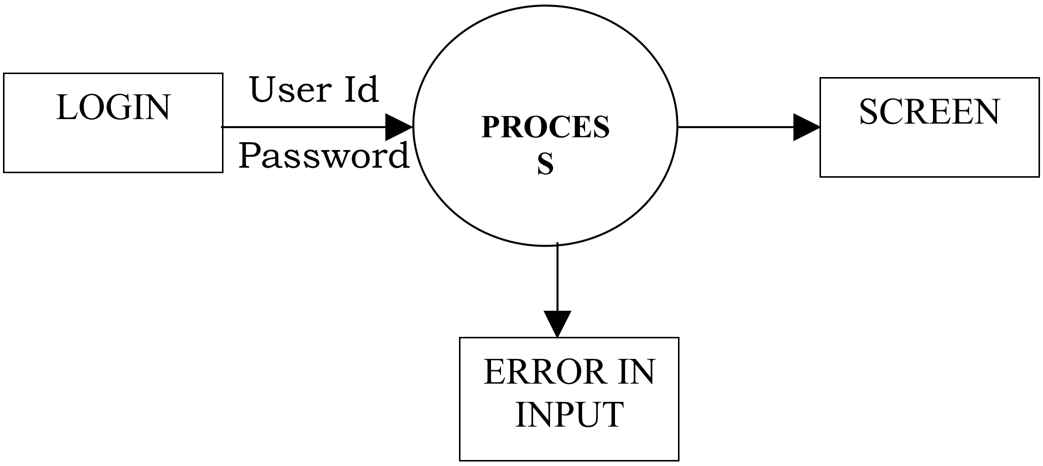


CONTEXT FLOW DIAGRAM

Description:

Context Flow Diagram gives us the complete details about the inputs and outputs for a given system. In the above system the main task is to identify a criminal face. So, the operator and eyewitness are the inputs to our system and criminal face is desired output.

LOGIN PROCESS

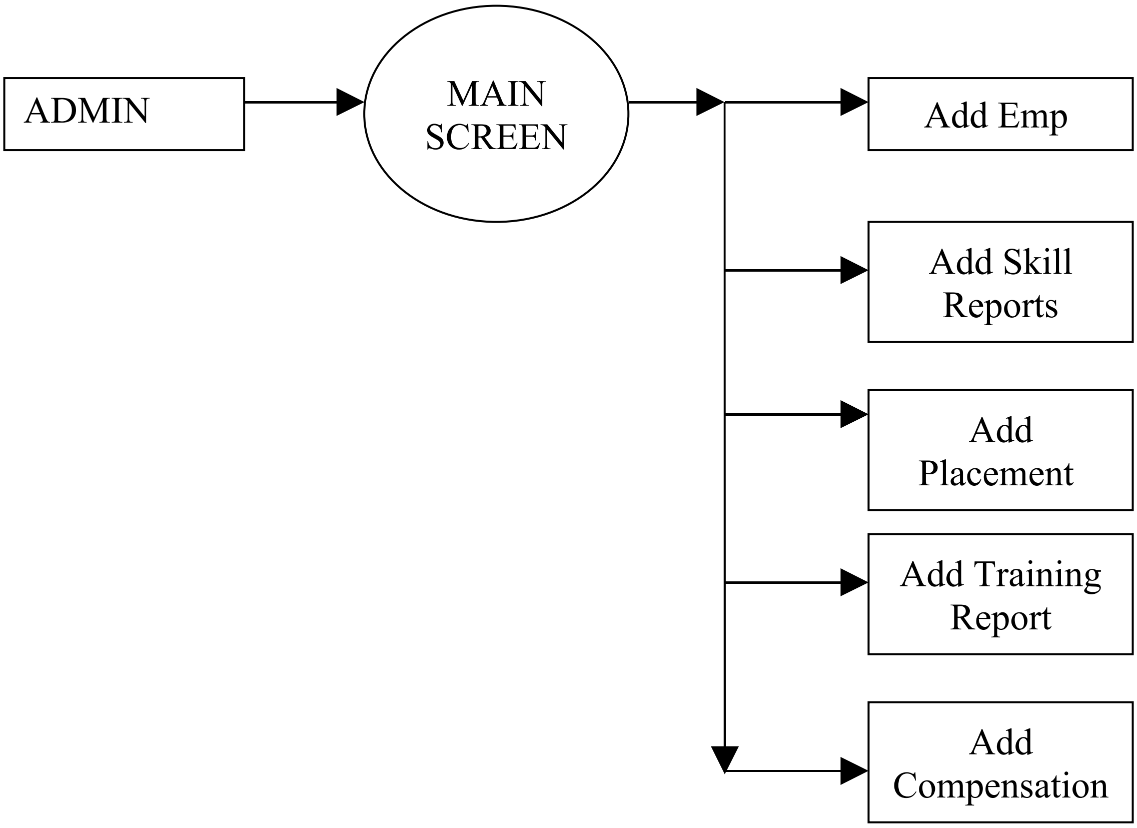


Level-1

Description:

The inputs to the process are User Id and Password given by the developer to allow the software available for the Admin environment. After giving the inputs the details, checks whether the entered ones are valid are not. It displays screen if match occurs otherwise error message if they are not matched.

MAIN SCREEN PROCESS

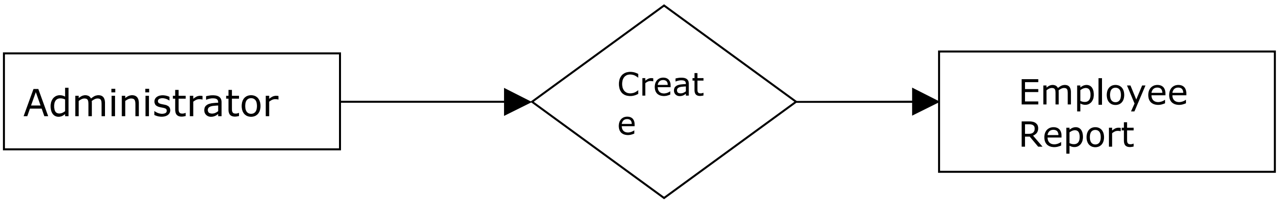


Level -2

Description:

This process mainly explains the different screens that are available for the admin. Here the selection of the screen depends on the admin and he can select whatever screen he wants. The different screens that are available are Add Emp details, Add skill Report, Add placement report, Add Training Report, Add Compensation Report.

Add Employee Details

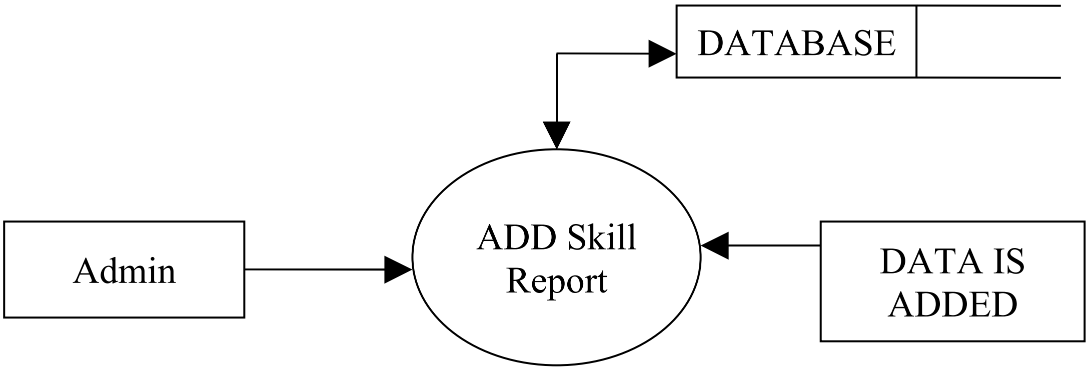


Level-3

Description:

This process clearly illustrates adding the details of the Employee such as name, age, gender, location, address, state and city along with his EMP Id. These details are being added to the database, if any error is generated then it will be prompted to the admin otherwise we get message data is successfully added.

Add Skill Report

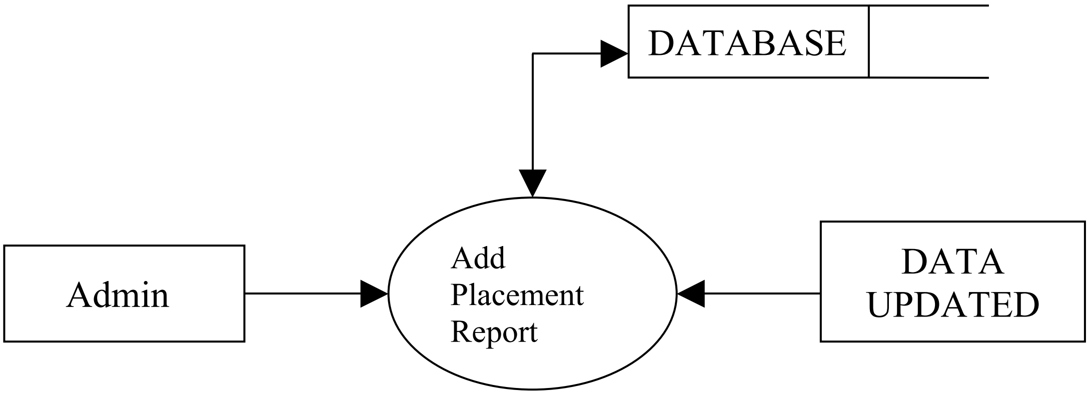


Level-4

Description:

This process clearly illustrates adding the details of the Employees skill Report such as name, qualification, experience, department, projects handled, current projects and skills along with his EMP Id. These details are being added to the database, if any error is generated then it will be prompted to the admin otherwise we get message data is successfully added.

Add Placement Report

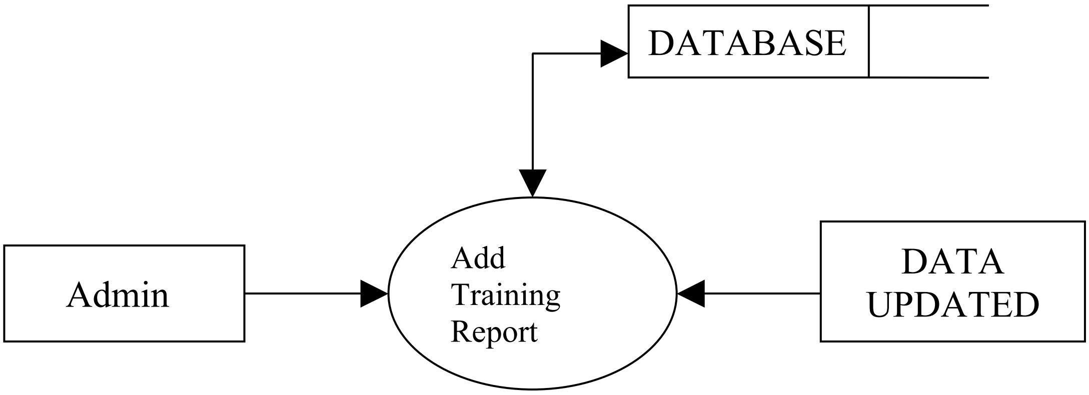


Level-5

Description:

This process clearly illustrates adding the details of the Employees placement Report such as name, employee status, reporting date, regulatory region, country, company control unit, department, supervisor id, location and along with his EMP Id. These details are being added to the database, if any error is generated then it will be prompted to the admin otherwise we get message data is successfully added.

TRAINING REPORTS

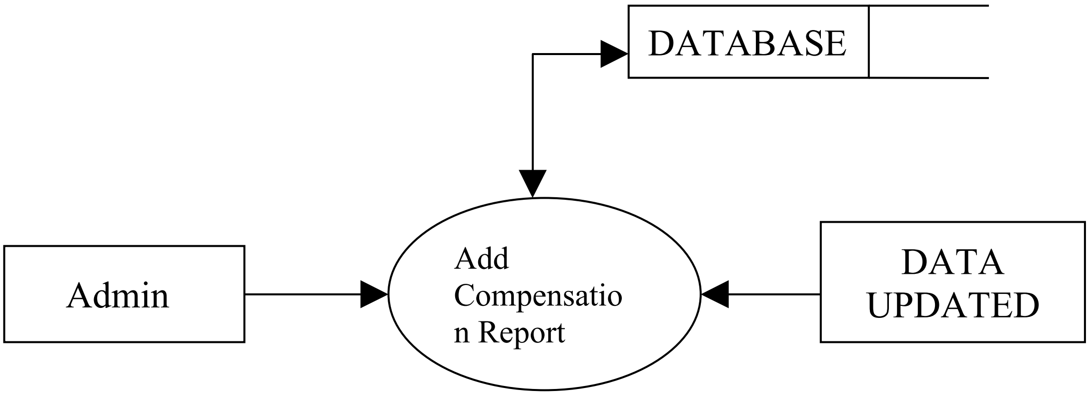


Level-6

Description:

This process clearly illustrates adding the details of the Employees Training Report such as name, project name, training department, skill report, start date, end date, department, and remarks and along with his EMP Id. These details are being added to the database, if any error is generated then it will be prompted to the admin otherwise we get message data is successfully added.

COMPARISON PROCESS



Level-7

Description:

This process clearly illustrates adding the details of the Employees Compensation Report such as name, salary range, and annual income, tax of income, loans, present annual income, facilities, and insurance and along with his EMP name. These details are being added to the database, if any error is generated then it will be prompted to the admin otherwise we get message data is successfully added.

7.2.2DATABASE TABLES

Login Table

user_login

user_name	password	user_type
admin	admin	Administrator
kishore	kishore	
sree	sree	employee
tarun	tarun	employee
vamsi	vamsi	employee

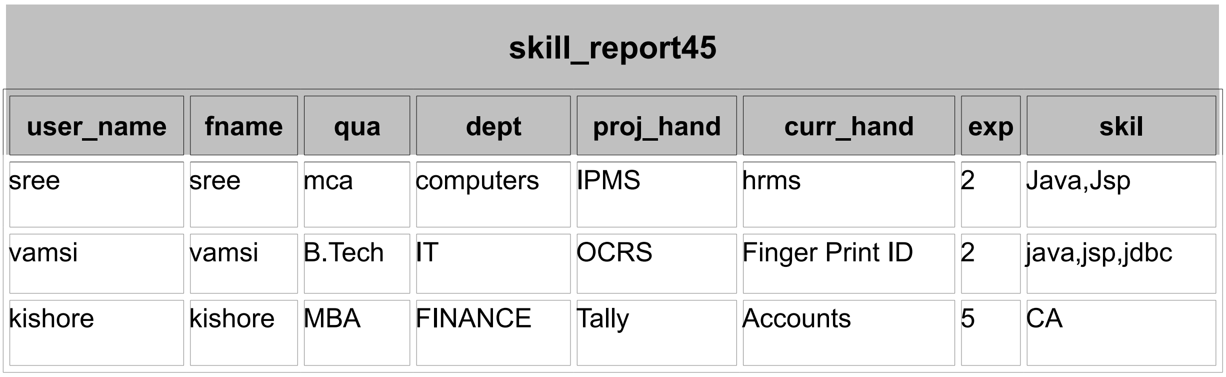
add_emp

user_name	password	id	first_name	last_name	qualification	department	date_of birth	age	sex	address	date_of_joining	supervisor_id_
kishore	kishore	1004	kishore	kumar	MBA	FINANCE	18-April	21	male	yousfguda	22may	employee
sree	sree	1001	sree	kumar	mca	computers	20/10/1984	21	male	DSNR	22/06/2005	employee
tarun	tarun	1003	tarun	kumar	B.Tech	computers	15/03/1985	20	male	yousfguda	16/03/2006	employee
vamsi	vamsi	1002	vamsi	Krishna	B.Tech	computers	20/05/1984	21	male	chandanagar	22/06/2005	employee

Add Employee Detai

Add

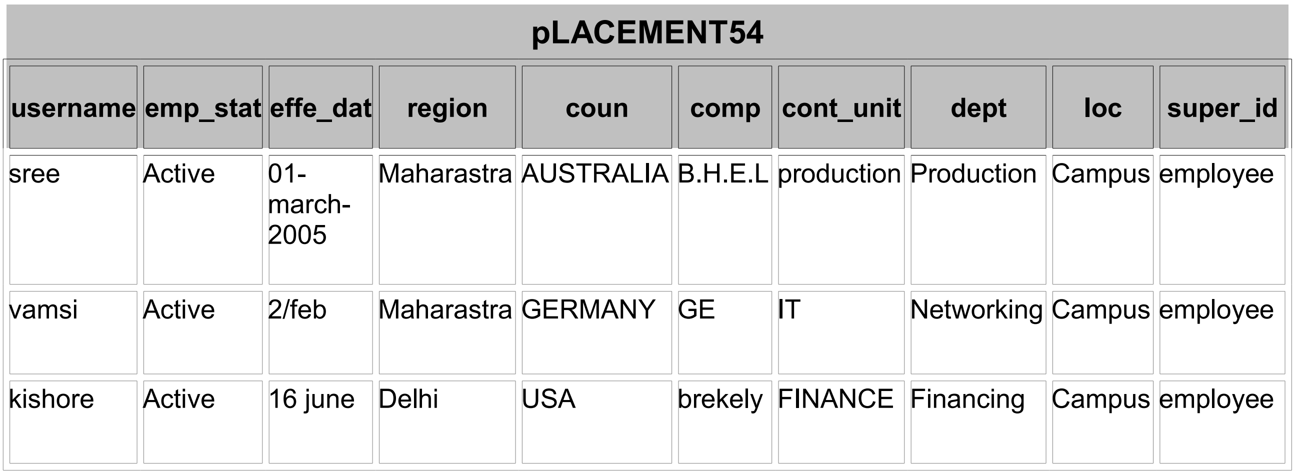
Skill Report



skill_report45

user_name	fname	qua	dept	proj_hand	curr_hand	exp	skil
sree	sree	mca	computers	IPMS	hrms	2	Java,Jsp
vamsi	vamsi	B.Tech	IT	OCRS	Finger Print ID	2	java,jsp,jdbc
kishore	kishore	MBA	FINANCE	Tally	Accounts	5	CA

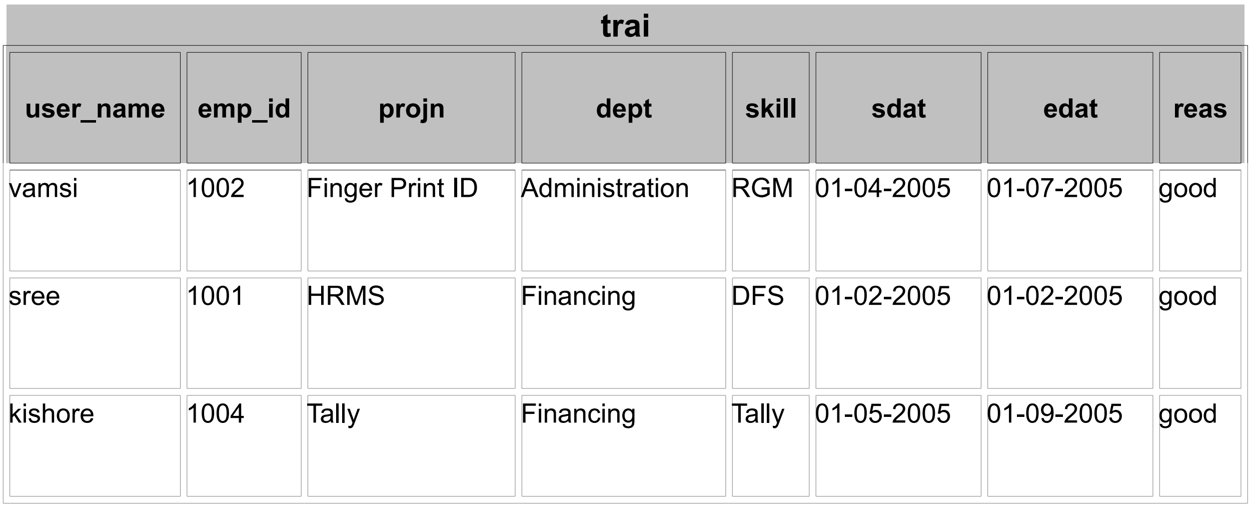
Add Placement Report



pLACEMENT54

username	emp_stat	effe_dat	region	coun	comp	cont_unit	dept	loc	super_id
sree	Active	01-march-2005	Maharastra	AUSTRALIA	B.H.E.L	production	Production	Campus	employee
vamsi	Active	2/feb	Maharastra	GERMANY	GE	IT	Networking	Campus	employee
kishore	Active	16 june	Delhi	USA	brekely	FINANCE	Financing	Campus	employee

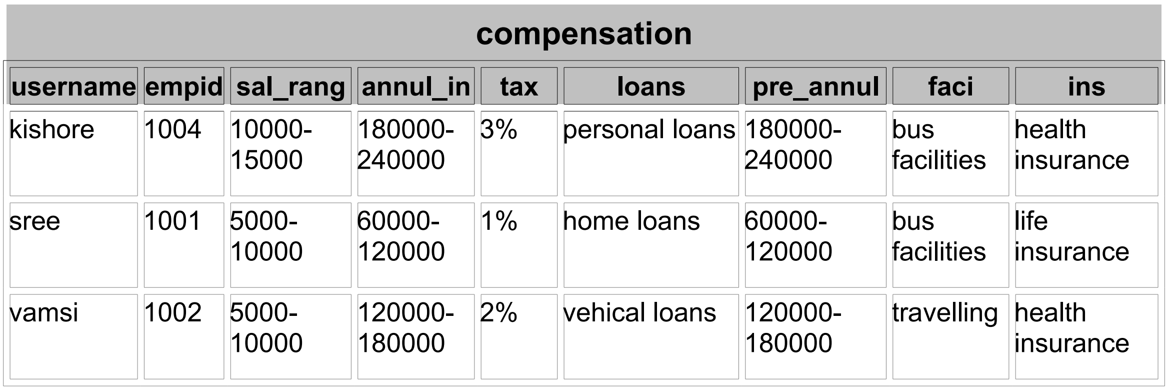
Add Training Report



trai

user_name	emp_id	projn	dept	skill	sdat	edat	reas
vamsi	1002	Finger Print ID	Administration	RGM	01-04-2005	01-07-2005	good
sree	1001	HRMS	Financing	DFS	01-02-2005	01-02-2005	good
kishore	1004	Tally	Financing	Tally	01-05-2005	01-09-2005	good

Add Compensation Report



compensation

username	empid	sal_rang	annul_in	tax	loans	pre_annul	faci	ins
kishore	1004	10000-15000	180000-240000	3%	personal loans	180000-240000	bus facilities	health insurance
sree	1001	5000-10000	60000-120000	1%	home loans	60000-120000	bus facilities	life insurance
vamsi	1002	5000-10000	120000-180000	2%	vehical loans	120000-180000	travelling	health insurance

DATA BASE DESIGN TABLES

LOGIN FORM

S NO	FIELD NAME	DATA TYPE
1	USER NAME	Varchar2(20)
2	Pass word	Varchar2(20)

EMPLOYEE DETAILS TABLES

SNO	FIELD NAME	DATA TYPE
1	Employee id	Varchar2(20)
2	First name	Varchar2(20)
3	Last name	Varchar2(20)
4	qualification	Varchar2(20)
5	department	Varchar2(20)
6	Date of birth	date
7	age	Number(3)
8	sex	Char(1)
9	address	Varchar2(20)
10	Date of joining	date
11	Supervisor id	Varchar2(20)

EMPLOYEE SKILL REPORT DATA TABLE

SNO	FIELD NAME	DATA TYPE
1	First name	Varchar2(20)
2	qualification	Varchar2(20)
3	department	Varchar2(20)
4	Project handled	Varchar2(20)
5	Current project	Varchar2(20)
6	experience	Number(3,2)
7	Skills	Varchar2(20)

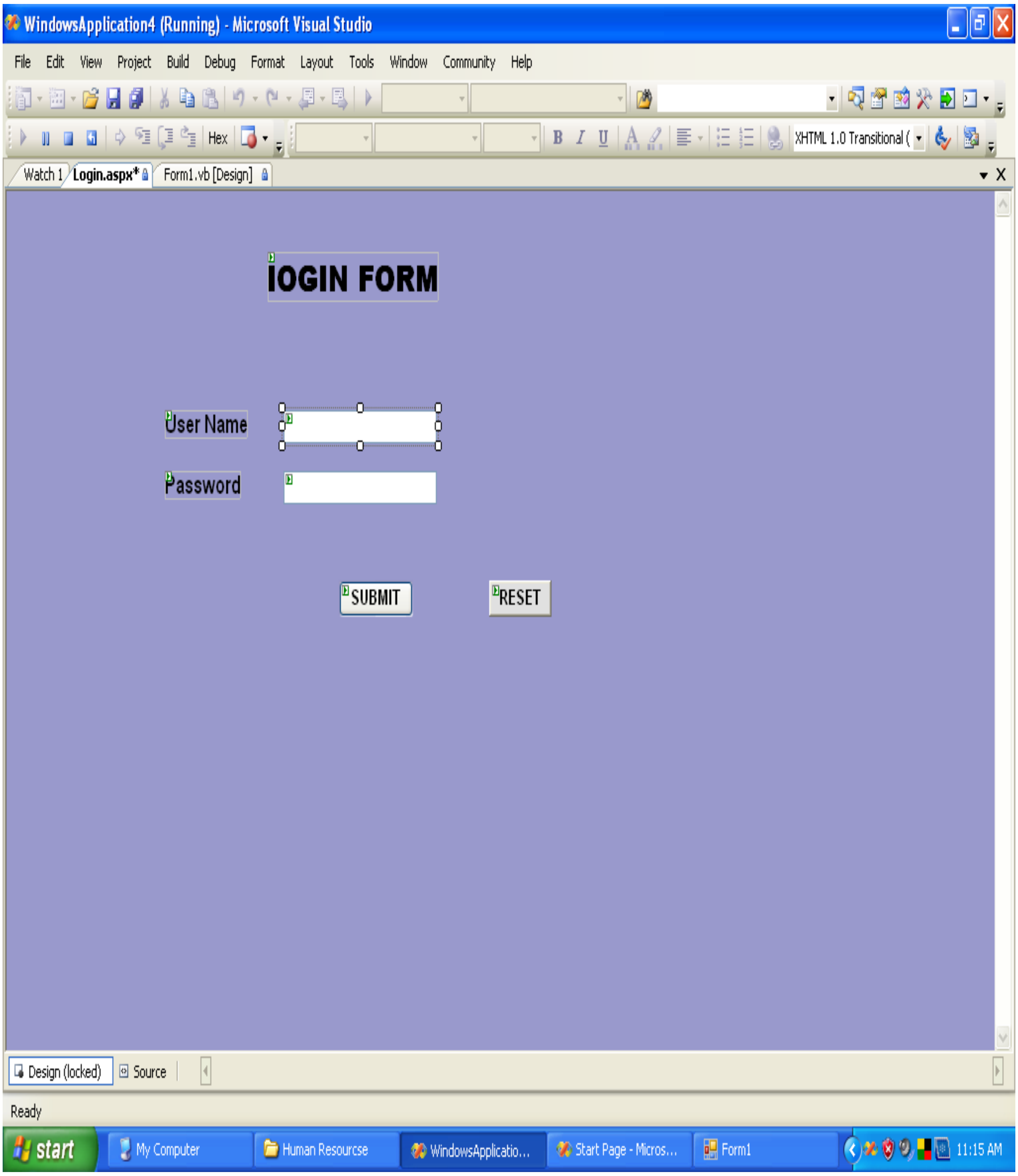
EMPLOYEE PLACEMENT REPORT DATA TABLE

SNO	FIELD NAME	DATA TYPE
1	USER NAME	Varchar2(20)
2	Empl_status	Varchar2(20)
3	Eff_date	Date
4	region	Varchar2(20)
5	country	Varchar2(20)
6	company	Varchar2(20)
7	Count-unit	Varchar2(20)
8	dept	Varchar2(20)
9	location	Varchar2(20)
10	Super id	Varchar2(20)

SNO	FIELD NAME	DATA TYPE
1	USER NAME	Varchar2(20)
2	Salary range	Number(10)
3	Annual income	Number(10)
4	Tax of income	Number(10)
5	loans	Varchar2(20_
6	Present annual income	Number(12)
7	facilities	Varchar2(20)
8	insurance	Varchar2(20)

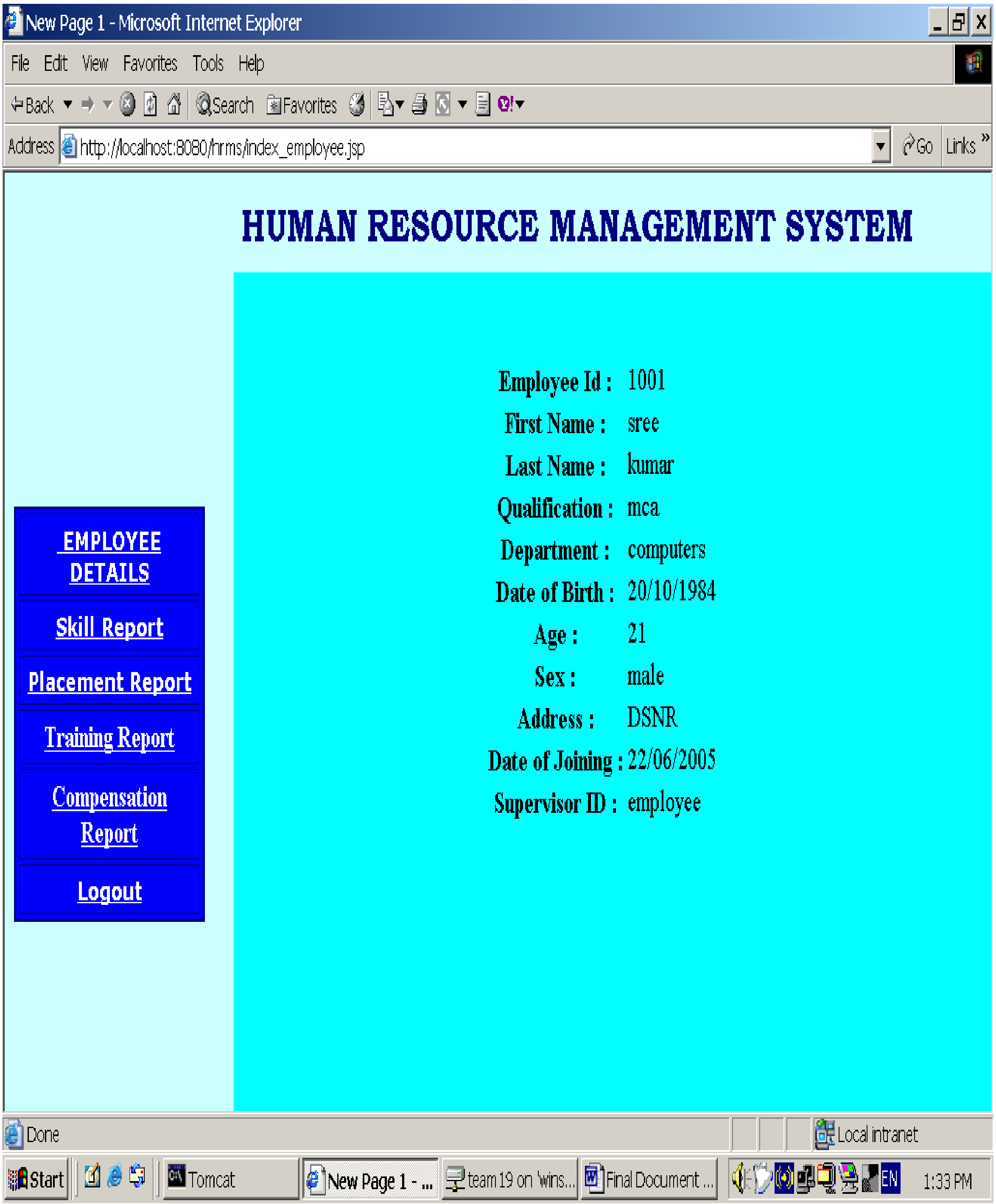
SNO	FIELD NAME	DATA TYPE
1	First name	Varchar2(20)
2	qualification	Varchar2(20)
3	department	Varchar2(20)
4	Project handled	date
5	Current project	date
6	experience	Number(3,2)
7	Skills	Varchar2(20)

7.2.3 LOGIN SCREEN


WindowsApplication4 (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Format Layout Tools Window Community Help
Hex
XHTML 1.0 Transitional (
Watch 1
Login.aspx*
Form1.vb [Design]
lOGIN FORM
User Name
Password
SUBMIT
RESET
Design (locked)
Source
Ready
start
My Computer
Human Resourcse
WindowsApplicatio...
Start Page - Micros...
Form1
11:15 AM

MAIN SCREEN

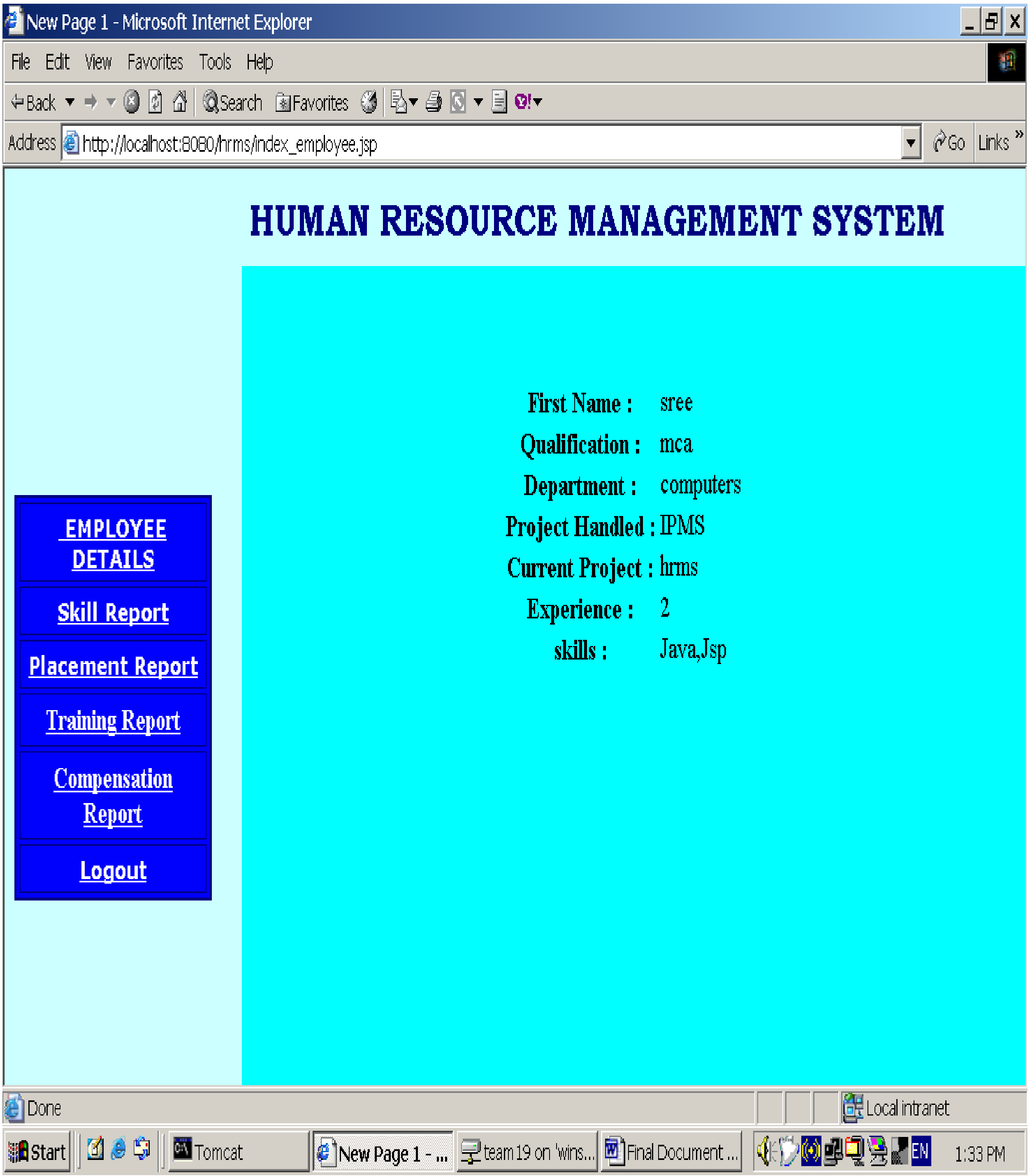
View employee details


New Page 1 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/hrms/index_employee.jsp
Go Links
EMPLOYEE DETAILS
Skill Report
Placement Report
Training Report
Compensation Report
Logout
Done
Local intranet
Start Tomcat New Page 1 - ... team19 on 'wins... Final Document ... EN 1:33 PM

HUMAN RESOURCE MANAGEMENT SYSTEM

Employee Id :	1001
First Name :	sree
Last Name :	kumar
Qualification :	mca
Department :	computers
Date of Birth :	20/10/1984
Age :	21
Sex :	male
Address :	DSNR
Date of Joining :	22/06/2005
Supervisor ID :	employee

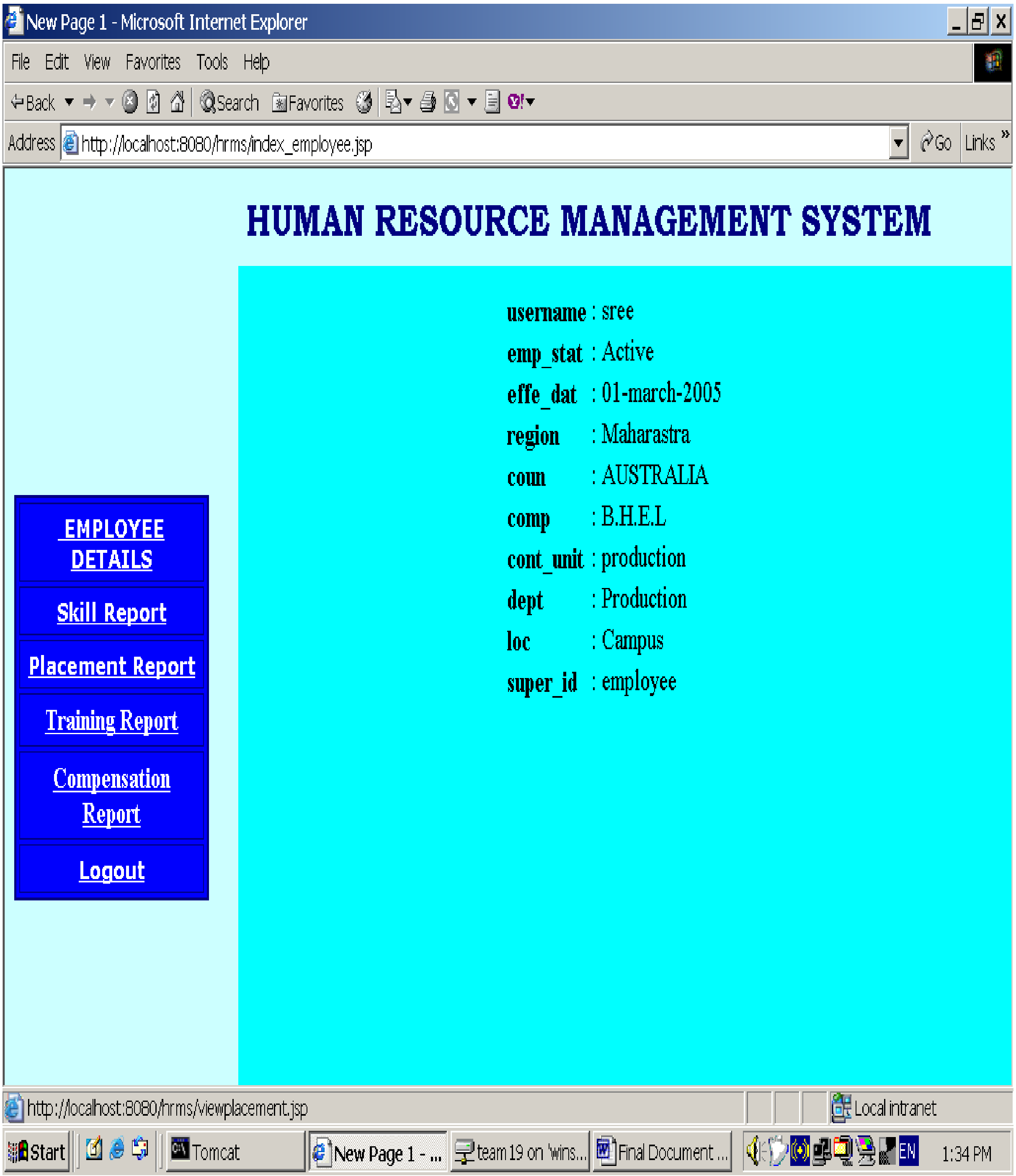
View employee skill report



HUMAN RESOURCE MANAGEMENT SYSTEM

First Name :	sree
Qualification :	mca
Department :	computers
Project Handled :	IPMS
Current Project :	hrms
Experience :	2
skills :	Java,Jsp

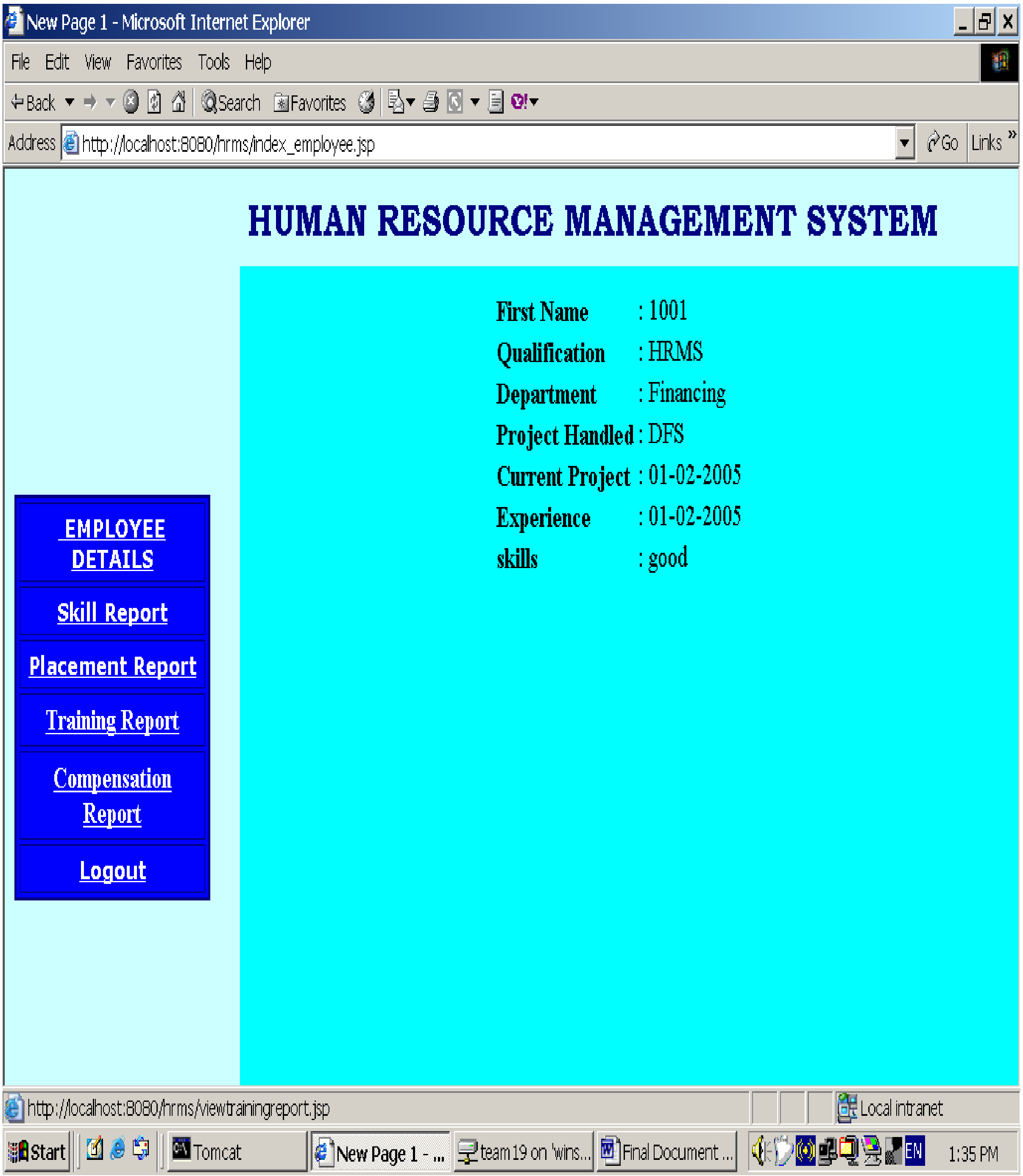
View employee placement report


New Page 1 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/hrms/index_employee.jsp
Go Links
EMPLOYEE DETAILS
Skill Report
Placement Report
Training Report
Compensation Report
Logout
http://localhost:8080/hrms/viewplacement.jsp
Local intranet
Start Tomcat New Page 1 - ... team 19 on 'wins... Final Document ... EN 1:34 PM

HUMAN RESOURCE MANAGEMENT SYSTEM

username : sree
emp_stat : Active
effe_dat : 01-march-2005
region : Maharastra
coun : AUSTRALIA
comp : B.H.E.L
cont_unit : production
dept : Production
loc : Campus
super_id : employee

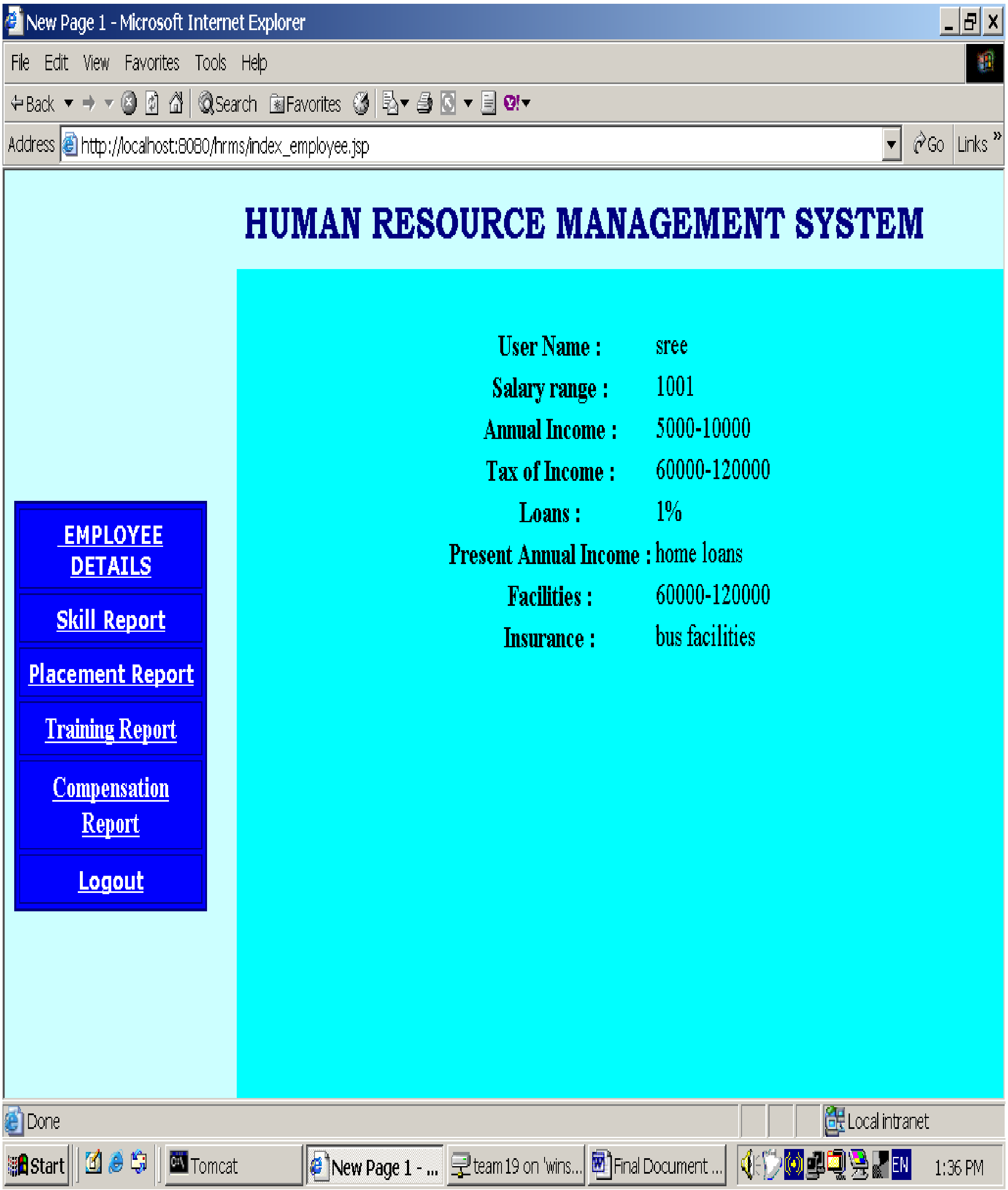
View employee training report


New Page 1 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/hrms/index_employee.jsp
Go Links
EMPLOYEE DETAILS
Skill Report
Placement Report
Training Report
Compensation Report
Logout
http://localhost:8080/hrms/viewtrainingreport.jsp
Local intranet
Start Tomcat New Page 1 - ... team19 on 'wins... Final Document ... EN 1:35 PM

HUMAN RESOURCE MANAGEMENT SYSTEM

First Name	: 1001
Qualification	: HRMS
Department	: Financing
Project Handled	: DFS
Current Project	: 01-02-2005
Experience	: 01-02-2005
skills	: good

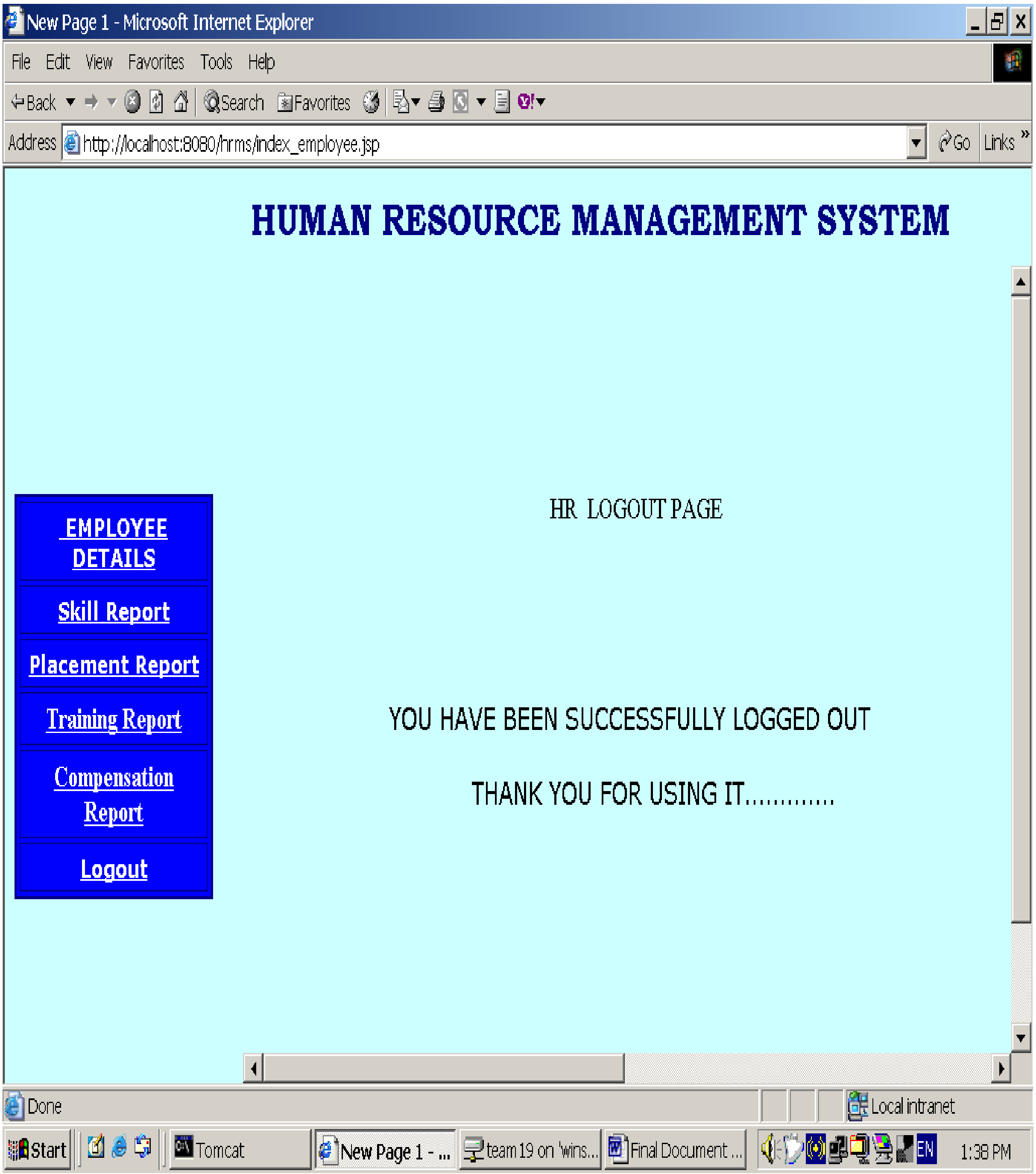
View employee compensation report



HUMAN RESOURCE MANAGEMENT SYSTEM

User Name :	sree
Salary range :	1001
Annual Income :	5000-10000
Tax of Income :	60000-120000
Loans :	1%
Present Annual Income :	home loans
Facilities :	60000-120000
Insurance :	bus facilities

Employee logout page


New Page 1 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/hrms/index_employee.jsp
Go Links
Done
Local intranet
Start Tomcat New Page 1 - ... team 19 on 'wins... Final Document ...
EN 1:38 PM

HUMAN RESOURCE MANAGEMENT SYSTEM

EMPLOYEE DETAILS

Skill Report

Placement Report

Training Report

Compensation Report

Logout

HR LOGOUT PAGE

YOU HAVE BEEN SUCCESSFULLY LOGGED OUT

THANK YOU FOR USING IT.............

8 TESTING PHASE

The testing phase involves the testing of the developed system using various kinds of data. An elaborated testing of data is prepared and a system is tested using the test data. While testing, errors are noted and corrections remade, the corrections are also noted for future use.

SYSTEM TESTING

Testing is a set of activities that can be planned in advance and conducted systematically. The proposed system is tested in parallel with the software that consists of its own phases of analysis, implementation, testing and maintenance. Following are the tests conducted on the system.

UNIT TESTING

During the implementation of the system each module of the system was tested separately to uncover errors with in its boundaries. User interface was used as a guide in the process.

MODULE TESTING

A module is composed of various programs related to that module. Module testing is done to check the module functionality and interaction between units within a module.

It checks the functionality of each program with relation to other programs within the same module. It then tests the overall functionality of each module.

INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing. The objective is to take unit-tested module and build a program structure that has been dictated by design.

ACCEPTANCE TESTING

The software has been tested with the realistic data given by the client and produced fruitful results. The client satisfying all the requirements specified by them has also developed the software within the time limitation specified. A demonstration has been given to the client and the end-user giving all the operational features.

IMPLEMENTATION PHASE

The implementation is the final and important phase. It involves User training, system testing and successful running of the developed system. The users test the developed system when changes are made according to the needs. The testing phase involves the testing of the developed system using various kinds of data. An elaborate testing of data is prepared and system is tested using the tests data.

Implementation is the stage where theoretical design turned into a working system. Implementation is planed carefully to propose system to avoid unanticipated problems. Many preparations involved before and during the implementation of proposed system. The system needed to be plugged in to the organization's network then it could be accessed from anywhere, after a user logins into the portal. The tasks that had to be done to implement the system were to create the database tables in the organization database domain. Then the administrator was granted his role so that the system could be accessed.

The next phase in the implementation was to educate the system. A demonstration of all the functions that can be carried out by the system was given to examination department person, who will make extensive use of the system.

9 CONCLUSION

10 BIBLIOGRAPHY

BOOKS REFERRED

The following books were used extensively for the project development and implementation.

1.”**VB.N ET** Developer’s Guide ” Syngress Publishing, Inc. Rockland,

by Cameron Wakefield

Henk-Evert Sonder

Wei Meng Lee

2. “ **ASP.NET** Database Programming” Hungry Minds, Inc.

Publishing Company Limited.

By Jason Butler and Tony Caudill

WEBSITES REFERRED

The following links were searched and exploited extensively for the project development and implementation.

1 www.syngress.com/solutions

2 www.hungryminds.com

Project Report

A STUDY ON

HUMAN RESOURCE MANAGEMENT SYSTEMS

AT

GALAXIE SOFTWARE SOLUTIONS

Submitted in partial fulfillment for the award of
Master of Business Administration (IT)

Submitted by
SRIKANTH.T
(H.T.No: 05D51E0034)

VIF COLLEGE OF ENGG. & TECHNOLOGY
AFFILIATED TO JNT UNIVERSITY
HYDERABAD
2005-2007

DECLARATION

I here by declare that the project titled **"HUMAN RESOURCE MANAGEMENT SYSTEM"** done at **GALAXIE SOFTWARE SOLUTIONS**, HYDERABAD, submitted by me as part of partial fulfillment for the award of the "**Master of Business Administration**" at. **VIF COLLEGE OF ENGG. & TECHNOLOGY** Affiliated to JNT University is a record of bonafied work done by me.

Place: **(SRIKANTH.T)**

Date: **(05D51E0034)**

Guide’s Certificate

This is to certify that the project work titled “**HUMAN RESOURCE MANAGEMENT SYSTEM”** done at **GALAXIE SOFTWARE SOLUTIONS, HYDERABAD** of **VIF COLLEGE OF ENGG &TECH**, moinabad, under by guidance in partial fulfillment of the requirement for the award of MBA (human resource management) degree.

Place: **Miss.B.JEHAN,**

Date: **Faculty member.**

CERTIFICATE

This is to certify that by **Mr. SRIKANTH.T** Bearing **roll no: 05D51E0034** a Bonfide student of MBA, IV Semester**, VIF COLLEGE OF ENGG &TECH**, moinabad has successfully completed her project work titled "**RECRUITMENT AND SELECTION" in** partial fulfillment of the requirement for the award of the degree of MBA (IT) of JNTU under the guidance of **Miss B.JEHAN** faculty member **VIF COLLEGE OF ENGG &TECH**, moinabad.

Place:

Date PRINCIPAL