

# *PROJECT STATUS INFORMATION SYSTEM*

*A Project developed for the degree of Masters in Computer Engineering at University of Bridgeport*



*Advisor  
Prof. Ausif Mahmood*

*Submitted by: Sreerama Arakapudi  
CPE 597B, Fall 2005  
Student ID#: 665136*

# **INDEX**

- 1. ABSTRACT**
- 2. INTRODUCTION**
- 3. SYSTEM ANALYSIS**
- 4. SYSTEM DESIGN**
- 5. SYSTEM ENVIRONMENT**
- 6. OUTPUT**
- 7. REPORTS**
- 8. SOFTWARE TESTING**
- 9. CONCLUSION**
- 10. BIBILOGRAPHY**

## ABSTRACT

The Project entitled "**Project Status Information System (PSIS)**" deals with the various levels of project development and will account for time used in analysis, design programming, testing and verification etc.

Information systems development projects range from one-person projects that take very little time and effort to multiple person, multi year efforts costing millions of dollars. The goal of Project Status Information System (PSIS) is to prevent projects from coming in late and going over budget.

Project Status Information System (PSIS) gives the management a clear picture of the usage of time by various projects i.e. utilized time and unutilized time. Every activity, no matter how small or large, requires use of the commodity called time. There is no substitute for time as there are substitutes for other resources. As such it is one of the most precious of resources. By analyzing the results provided by the software they might rectify the defects in utilizing time and take remedial actions.

Project Status Information System (PSIS) takes time sheet as input. The input may be in non-standard format differing from project to project.

Project Status Information System (PSIS) produces output in the form of reports. This output gives a clear picture of the time used at various levels of the project.

In Project Status Information System (PSIS) new project information is entered by the technical manager, based on the project information project manager will assign activities to employees who are working under him.

# **INTRODUCTION TO PROJECT STATUS INFORMATION SYSTEM (PSIS)**

## **Definition:**

Project Status Information System (PSIS) deals with the various levels of project development and will account for time used in analysis, design, programming, testing and verification etc.

## **Description:**

Project Status Information System (PSIS) gives the management a clear picture of usage of time by projects i.e. utilized time and unutilized time. By analyzing the results provided by the software they might rectify the defects in utilizing time and take remedial actions.

Project Status Information System (PSIS) takes time sheet reports as input. The input may be in non-standard format differing from project to project. Project Status Information System gives the individual report of project, which contains time used for various tasks, and also gives the picture of the unutilized time. In Project Status Information System (PSIS) new project information is entered by the technical manager, based on the project Information project manager will assign activities to employees who are working under him.

## **OBJECTIVES:**

When using this system to manage a project, you need to perform the following activities:

- Establish a project starting and ending date
- Enter activities and assign activity relationships
- Select a scheduling method to review project  
Information

**Proposed System:**

The newly constructed system will maintain all the above details at one place and depends upon the login constraints we are using it allows us to access the details of different phases of the system and all this system is constructed with different modules.

Project Status Information System takes time sheet as input. The input may be in non-standard format differing from project to project. Project Status Information System produces output in the form of reports. This output gives a clear picture of the time used at various levels of the project.

In Project Status Information System new project information is entered by the technical manager, based on the project information project manager will assign activities to employees who are working under him.

**Definition:**

Project Status Information System deals with the various levels of project development and will account for time used in analysis, design, programming, testing and verification etc.

**Description:**

Project Status Information System gives the management a clear picture of usage of time by projects i.e. utilized time and unutilized time. By analyzing the results provided by the software they might rectify the defects in utilizing time and take remedial actions.

Project Status Information System takes time sheet reports as input. The input may be in non-standard format differing from project to project. Project Status Information System gives the individual Report of project, which contains time used for various tasks, and also gives the picture of the unutilized time. In Project Status Information System new project information is entered by the technical manager, based on the project Information project manager will assign activities to employees who are working under him.

# **SYSTEM ANALYSIS**

## **SYSTEM DEFINITION**

### **Requirement Analysis:**

A requirement is a feature that must be included in the system. Before the actual design and implementation start, getting to know the system to be implemented is of prime importance.

Main emphasis should be on:

- The inputs to the system.
- The outputs expected from the system.
- The people involved in the working of the system.
- The volume of DATA (INPUTS) and the amount of INFORMATION (outputs) that will be involved.

With respect to the system itself, the following facts should be taking into consideration

The major processes involved.

- The main points of application.
- The processing rules for the collected data.
- The exceptions that may be present.
- The checks that should be in place in order to avoid wrong entries.

# SOFTWARE REQUIREMENT SPECIFICATION

## **Purpose:**

The purpose of this document is that it deals with various levels of project development and will account for time used in analysis, design, programming, testing and verification etc.

## **Scope:**

The module involved in giving the management the clear picture of the usage of time by projects i.e. utilized time and unutilized time. By analyzing the results provided by the software

## **Documentation Overview:**

This document has 3 major sections

- Section 1 provides an overview of entire software requirement specification
- Section 2 provides the product that will be produced It includes:
  - a. Product perspective
  - b. Product activity
  - c. User characteristics
  - d. General constraints
- Section 3 addresses the specific requirements of the system

It includes:

Functional requirements External interfaces requirements Performance requirements Design constraints Attributes

# SOFTWARE AND HARDWARE REQUIREMENTS

## SOFTWARE REQUIRMENT SPECIFICATION:

**OPERATING PLATFORM:** WINDOWS 98/2000/NT/XP

**DATA BASE** : ORACLE 9i

**SOFTWARE** : JDK 1.5, TOMCAT (web server)

**BROWSER** : INTERNET EXPLORER

**FRONT END TOOL** : HTML

**LANGUAGE** : JAVA 2.0

**DOCUMENTATION TOOL:** MS – WORD 2003

## HARDWARE REQUIREMENT SPECIFICATION:

**PROCESSOR** : Pentium M

**SPEED** : 1.73 GHZ

**HARD DISK** : 4.2 MB

**RAM** : 128 MB

## REQUIREMENT STUDY

The origin of most software systems is in the need of a client, who either wants to automate the existing manual system or desires a new software system. The software system itself is created by the developer finally the completed system will be used by the end user. Thus, there are three major parties interested in a new system: the client, the users, and the developer. The requirements for the system that will satisfy the needs of the clients and the concerns of the users have to be communicated to the developer. The problem is that the client usually does not understand software or the software development process, and the developer often does not understand the clients problem and application area. This causes a communication gap between the parties involved in the development project. A basic purpose of software requirements specification is to

bridge this communication gap .SRS is the medium through which the client and the user needs are accurately specified; indeed SRS forms the basic of software development. A good SRS should satisfy all the parties-something very hard to achieve and involves trade-offs and persuasion.

**The Requirement Process:**

The main reason of modeling generally focuses on the problem structure, not its external behavior. Consequently, things like user interfaces are rarely modeled, whereas they frequently form the major components of the SRS.

Similarly performance constraints, design constraints, standards compliance, recovery, etc. are specified clearly in the SRS because the designer must know about them to properly design the system.

To properly satisfy the basic goals, an SRS should have certain properties and should contain different type of requirements. A good SRS is [IEE87, IEE94]: complete if everything the software is supposed to do and the responses of the software to all classes of input data are specified in the SRS. Correctness and completeness go hand-in-hand; an SRS is unambiguous if and only if every requirement stated has one and only one interpretation, requirements often written in natural language.

An SRS is verifiable if and only if every stated requirement is verifiable. A requirement is verifiable if there exist some cost-effective process that can check whether the final software meets those requirements. An SRS is consistent if there is no requirement that conflicts with another.

Writing an SRS is an iterative process. Even when the requirements of a system are specified, they are later modified as the needs of the client change. Hence an SRS should be easy to modify. An SRS is traceable if the origin of each of its requirements is clear and if it facilitates the referencing of each requirement in future development [EEE87].

One of the most common problems in requirement specification is when some of the requirements of the client are not specified. This necessitates addition and modifications to the requirements later in the development cycle, which are often expensive to incorporate.

## **PROJECT SCHEDULE STUDY PHASE**

In the study phase we do the preliminary investigation and determine the system requirements. We study the system and collect the data to draw the dataflow diagrams. We follow the methods like questions and observation to find the facts that are involved in the process. This is important because if the specification study is not done properly then following design phase etc will go wrong.

## **DESIGN PHASE**

In this design phase we design the system making use of study phase and the data flow diagrams. We make use the general access methods for designing. We consider the top down approach. In the design phase we determine the entities and their attributes and the relationships between the entities. We do both logical and the physical design of the system.

## **DEVELOPMENT PHASE**

In the development phase we mostly do the coding part following the design of the system. We follow modular programming for development and after development and after developing each and every module we do the unit testing followed by the integration testing.

## **IMPLEMENTATION PHASE**

The last phase of the project is the implementation phase. Quality assurance is the primary motive in this phase. The quality assurance is the review of software products and related documentation for completeness, correctness, reliability and maintainability. The philosophy behind the testing is to finds errors. The testing strategies are of two types, the code testing and the specifications testing. In the code testing we examined the logic of the program. On the surface, code testing seems to be ideal methods for testing software, but not all software errors are uncovered.

## **DESCRIPTION:**

By manual system projects, clients, project time details information is stored in different registers, while retrieving the information it is every time taking process to search the information in each register as it is very fast through the computerized Project Time Analysis and *Reporting System*.

In manual system after each and every transaction the updating in the registers is little bit different in computerized system. The process of updating is automatic according to the new system. Human errors can be avoided due to the less manual involvement.

The system provides different report and getting reports of the required requirements is very fast and easy in computerized system.

## **DESIGN PHASE INTRODUCTION**

Design is the first step in the development of any engineered product or system. It may be defined as the process of applying various techniques and principles for the purpose of defining a device, a processor or a system with insufficient detail to permit its physical realization.

In the design phase the detailed design of the system selected in the study takes place. System design is stated by reviewing the study phase activities and making decisions as to which functions are to be performed by hardware, software or manually. These processes through which the initial requirements are specified in the study phase are translated into a representation of the software. Initially, the representation depicts holistic view of the software. Subsequent refinement leads to a design representation, which is very close to that of the source code. The design phase recommendations are presented as reports to the user. The following steps are followed in designing software.

- a. The statements of functions desired from software to be developed are recognized.
- b. The database design steps are applied to identify the various entities and relationships between them.
- c. General access methods are then applied to the conceptual schema to get the logical model, which is then further transformed into relational tables.
- d. The concept of software engineering is then applied to get the logical architecture of the system comprising of different modules, each containing all related functions.

## SOFTWARE ENVIRONMENT

### WHY CHOOSE "JAVA":

In its brief existence, the WORLD WIDE WEB has evolved into a truly global Information space. Not only does it offer information on any subject, it provides its citizens with the power to globally publish information on any subject and at minimal cost. The massive international participation in the web has resulted in the creation of many web pages that are not only informative, but also entertaining. This entertainment value has further fueled the webs growth and has led to browsing becoming an international pastime.

Browsing involves scanning web pages for interesting information, following useful links to other pages, and repeating the process until we come across something that makes us temporarily stop and focus sometimes we mutter "HMM" or "That's interesting!" and create a bookmark. Then, eventually, we move on.

The tendency to move on and continue browsing is natural.

We usually don't read the same book over and over. If we have stock of magazines in front of us, we are likely to flip through them all. Web pages are like magazine pages, except that they are available, usually free and have more "next" pages to which to turn computer programs are different. They are active, while books, magazines, and web pages are static or passive. People do use programs over and over.

This difference between active computer program and passive web pages is what makes JAVA an attractive addition to the web. When we click on a web page containing a JAVA APPLET, we don't just read it, listen to it, or watch it we interact with it. Interactivity is the difference between a program and a page, and JAVA has brought dynamic, interactive content to the web.

JAVA'S rapidly growing popularity is due to the web. But Java's inherent power does not come from the fact that it is a web programming language. The talented software engineers at Sun brought Java to the web, have elegantly solved a much broader and more significant problem how to develop network capable of windowing software that will run on almost any 32-bit computer and operating system.

The Modern software developers faces enormous headaches and challenges when he tries to develop software that is portable to Microsoft windows, X window systems, motif, Macintosh, and OS/2 windowing and networking environments. The usual approach is to pick a target operating system (OS), write software for that OS, and eventually migrate it to the other OS platforms. This usually involves great expense in terms of labor and software development resources. It also results in the sacrifice of features that are difficult to support across multiple OS platforms.

Java fulfills the software developer's dream of being able to write software in a single language that will support windowing and networking on all major OS platforms, without specified tailoring or even compilation. But Java's attractiveness does not end here. Sun's engineers were thoughtful enough to develop in java a language and runtime system that art simple, compact, Object Oriented, extendible, and secure. And then they gave it away for free.

## **ABOUT JAVA**

Java is a programming language, a runtime system, a set of development tools, an application programming interface (API). The relationship between these elements is depicted in figure.

The Java API contains predefined software packages with numerous platform independent "hooks" into the native windowing and networking capabilities of the host operating system. The Java API provides a single common API across all operating system to which Java is ported.

The keys to Java's portability are its run time system, and it's API. The run time system is very compact, evolving from earlier Sun efforts to build a software platform for consumer electronics. Because this platform was not designed around any existing microprocessor, it was built from scratch to be simple and efficient. The fact that it was not tied to a given hardware architecture enabled it to be architecture neutral. The Simple, efficient, compact and architectural neutral nature of the runtime system allows it to be highly portable and still provide effective performance.

The powerful windowing and networking features included in the Java API make it easier for programmers to develop software that is both attractive and platform independent. For example, Adam is a programming language that is highly standardized and supported on most operating systems. Yet Adam applications are not very portable. This is because Adam does not come with a common API that supports windowing and networking on all platforms. Java differs from Adam and all other programming languages in that there is one universal, but powerful, Java API for all operating systems platforms. That is why Java is the most portable language.

## **JAVA DATABASE CONNECTIVITY**

### **Introduction:**

The database is the most important component of a company's information services infrastructure. It is heart of the applications on which a company depends for its survival. Any programming language must be able to provide an application with access to these databases if it is to be considered a serious programming language.

The issues surrounding database access are often very difficult; other languages use either proprietary APIs specific to individual databases or complex universal APIs such as ODBC. Before starting any program there must be a need to use through data modeling and database design.

## **DATA BASE ARCHITECTURE**

There are three types of architecture: ONE TIER ARCHITECTURE, TWO-TIERED and THREE-TIERED.

### **ONE TIER ARCHITECTURE**

The application and the data reside together logically. These are not usually database programs. The logic and its data reside together.

### **TWO-TIERED**

The application resides in a different logical location than the data. These are usually database applications. Most Client/Server applications fit into this category.

### **THREE-TIERED**

In a three-tiered system, the application resides in a different logical location than the logic of the application and the data.

To put it another way, the client software makes a call to a remote service. That remote service is responsible for interacting with the data and responding to the client. The client has no knowledge of how and where the data is stored. All it knows about is the remote service has no knowledge of the clients that will be calling it. It only knows about the data.

## **ODBC**

Open Database Connectivity, or ODBC, is a standard developed by Microsoft Corporation. ODBC is an application program interface for accessing data in a standard manner from an abundance of data sources regardless of their type. If the data source is ODBC compliant, program can talk to it.

ODBC drivers are available for almost every major database vendor. Using ODBC the user can communicate with the database through any front-end tool like C, C++, and JAVA.

## **JDBC INTERFACES**

JDBC defines eight interfaces that must be implemented by a driver in order to be JDBC-compliant:

```
java.sql.Driver java.sql.Connection java.sql.Statement  
java.sql.PreparedStatement java.sql.CallableStatement java.sql.ResultSet  
java.sql.ResultSetMetaData java.sql.DatabaseMetaData
```

### **Java. SQL. Driver Interface:**

Driver is essentially a connection factory. The Driver Manager uses a Driver to determine whether it can handle a given URL. If one of the Drivers in its list can handle the URL, that Driver should create a connection object and return it to the Driver Manager. Because an application is rarely concerned with this interface Java. SQL. Connection Interface.

A Connection is a single database session. As such, it stores state information about the database session it manages and provides the application with Statement, preparedStatement or CallableStatement objects for making calls during the session Java. SQL. Statement Interface.

A Statement is an unbound SQL call to the database. It is generally a simple UPDATE, DELETE, INSERT, or SELECT Statement in which no columns must be bound to Java data.

A Statement provides methods for making such calls and returns to the application the results of any SELECT Statement or the no of rows affected by an UPDATE, DELETE, or INSERT Statement.

Statement has the subclass preparedStatement, which is in turn subclass by Callable Statement. A preparedStatement is a precompiled database call that requires with parameters to be bound. An example of a prepared Statement is a stored procedure call that has no OUT or INOUT parameters. For stored procedures with OUT or INOUT parameters, an application should use the Callable Statement Interface. Java. SQL. Result Set Interface.

An application gets data returned by a SELECT query through the implementation of the Java. SQL. Result set Interface. Specifically, the Result Set object enables an application to retrieve sequential rows of data returned from a previous SELECT call.

The Result Set provides a multitude of methods that enable you to retrieve a given row as any data type to which it makes sense to convert it. For example, if you have a data stored in the database as a date/time, you can retrieve it through the getString () method and use it has a String.

#### **Callable Statement:**

Callable Statement is used to execute SQL stored procedures. JDBC provides a stored procedure SQL escape that allows stored procedures to be called in a standard way for all Rebases'.

A Callable Statement may return a Result Set or Multiple Result Sets. Multiple Result Sets are handled using operations inherited from the Statement Connection.

A Connection represents a session with a specific database. Within the context of a connection, SQL statements are executed and results are returned. A Connection's database is able to provide information describing its tables, its supported SQL grammar,

its stored procedures, the capabilities of this connection, etc. This information is obtained with the getMetaData method. Prepared Statement

A SQL Statement is pre-compiled and stored in a prepared Statement object. This object can then be used to efficiently execute this Statement multiple times.

Some prepared statements return multiple results; the execute method handles these complex statements as well as the simpler form of statements handled by executeQuery and executeUpdate.

**Time Stamp:**

Set a parameter to a java.sql.TimeStamp value. The driver converts this to a SQL TIMESTAMP value when it sends it to the database. Parameters:

Parameter Index-the first parameter is 1, the second is 2 ... X - the parameter Value Throws: SQL Exception if a database- access error occurs.

**ABOUT JAVA SCRIPT**

Java script is Netscape's cross-platform, object-based scripting language for client and server applications. There are two types of java script:

Navigator java script, also called client-side java script

Live wire java script, also called server-side java script

Netscape navigator 2.0 can interpret java script statements embedded in an html page. When navigator requests such a page, the server sends the full length of the document, including html and execute the java script, producing the results that the user sees.

**Features of java script:**

Java script is interpreted by the client. Java script is object based. It uses built in, extensible objects, but no classes or inheritance. Java script code is integrated with and embedded in html. In java script variable data types are not declared. The-striking feature of java script is Dynamic binding .The object references are checked at runtime. It cannot be automatically written to hard disk.

**Script tag:**

The <script> tag is an extension to HTML that can enclose any number of java script statements.

`<Script language = "JavaScript version">Java script statements; </script>` A document have multiple script tags.

## **ABOUT HTML**

Hyper text markup language is used to create static web pages. It is a set of markup tags used to format text and include other data formats in a hypermedia documents so that the web browsers can interpret and display them. Basic HTML tags:

Html documents can be said to be made of html elements, which are pieces of code with in mark up tags. Some examples of elements are Heads, Tables, Paragraphs and Lists. Some elements may include attribute, which is additional information that is included inside the start tag.

An Html document starts with `<Html >` tag and ends with `</html>` tag. Html documents contain two parts namely Head and Body. Head:

It contains the descriptions of the Html page. `<HEAD> <Title>...</Title> </HEAD>`

Body:

This is large part of document, which contains the content of the document to be displayed in the text area of browser. `<BODY>`

`</BODY>`

We can also create links to another web pages using anchor tags.

`<a href="file\...\\">.....</a>`

We can also use form tag for specifying the attribute as action.

`<form action="..\ A">.....</form>`

By using all the tags we create static web pages. Dhtml is used for creating dynamic WebPages. It is used for making styles.

## **ORACLE:**

ORACLE is a relational Database management system that has a transparent interface between the physical storage and logical presentation of data and provides the user with a set of flexible & sophisticated tools to perform the operations basing on data and the data structures.

Oracle is a modular system that consists of the Oracle database and several functional packages.

Oracle tools do 4 major kinds of work:

- Database Management
- Data access & manipulation
- Programming

For handling information the user can use these tools to:

- Define a database
- Query a database
- Add, update, edit & delete data
- Modify the structure of database
- Export & Import data.
- Connectivity between back-end & front-end

**Database Management tools:** This usually includes the core programming of oracle's database management system .The oracle database with its associated tables & views are stored in the oracle data dictionary. The data dictionary stores information related to every fact of the database system

**Data access & manipulation tools:** ORACLE uses the SQL command set for the purpose of data access & its manipulation. Structured Query Language (SQL) includes all commands, which are used for data access.

Reason for **choosing ORACLE:**

1) Applications developed on Oracle are completely portable to other versions of programming. Programmers create a complex application in a single user environment and then move it to multi-user platform.

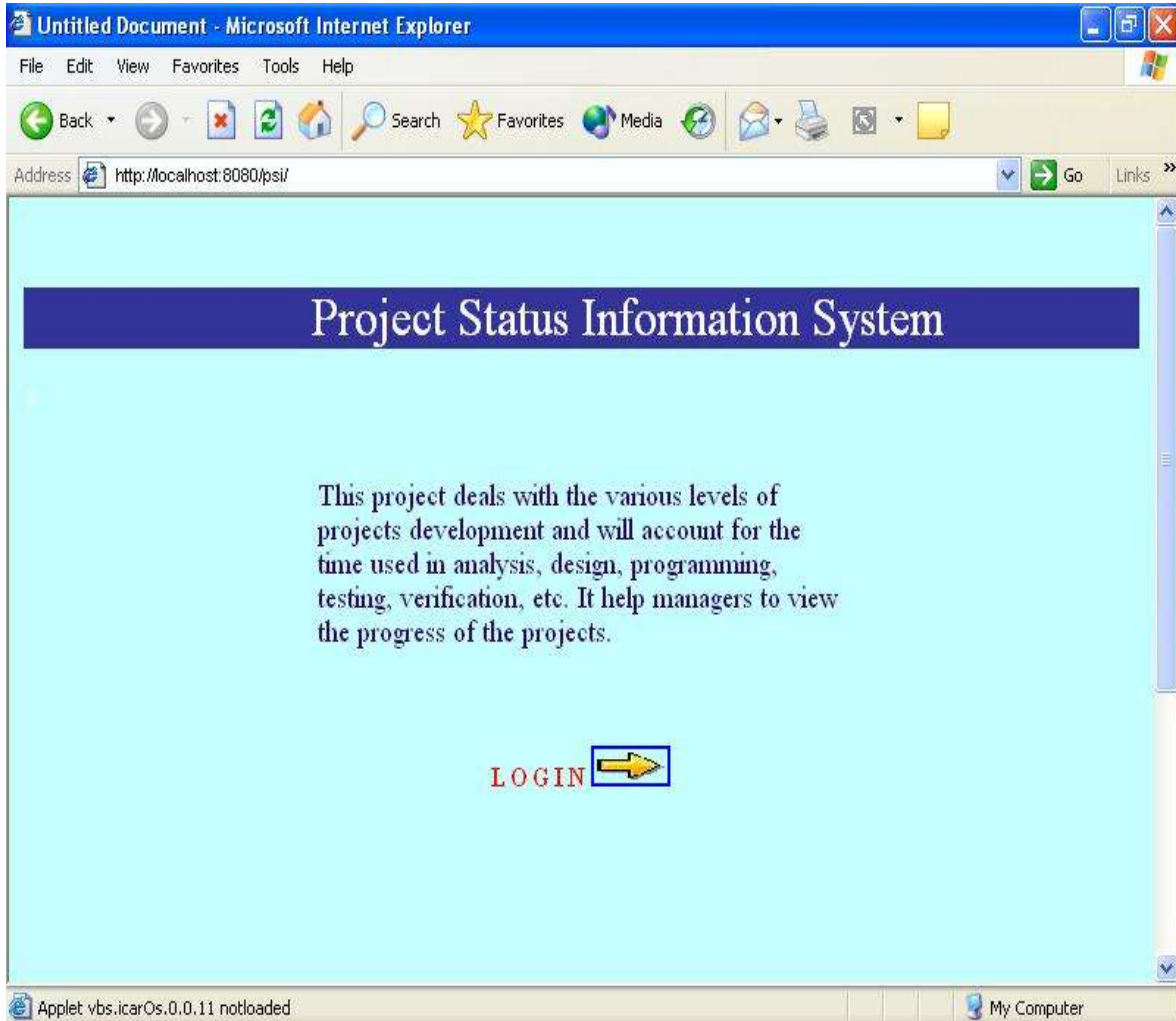
2) Oracle has several features that ensure integrity of user database. Database can be made secure by restrictions access to only authorized users using the tools available in ORACLE.

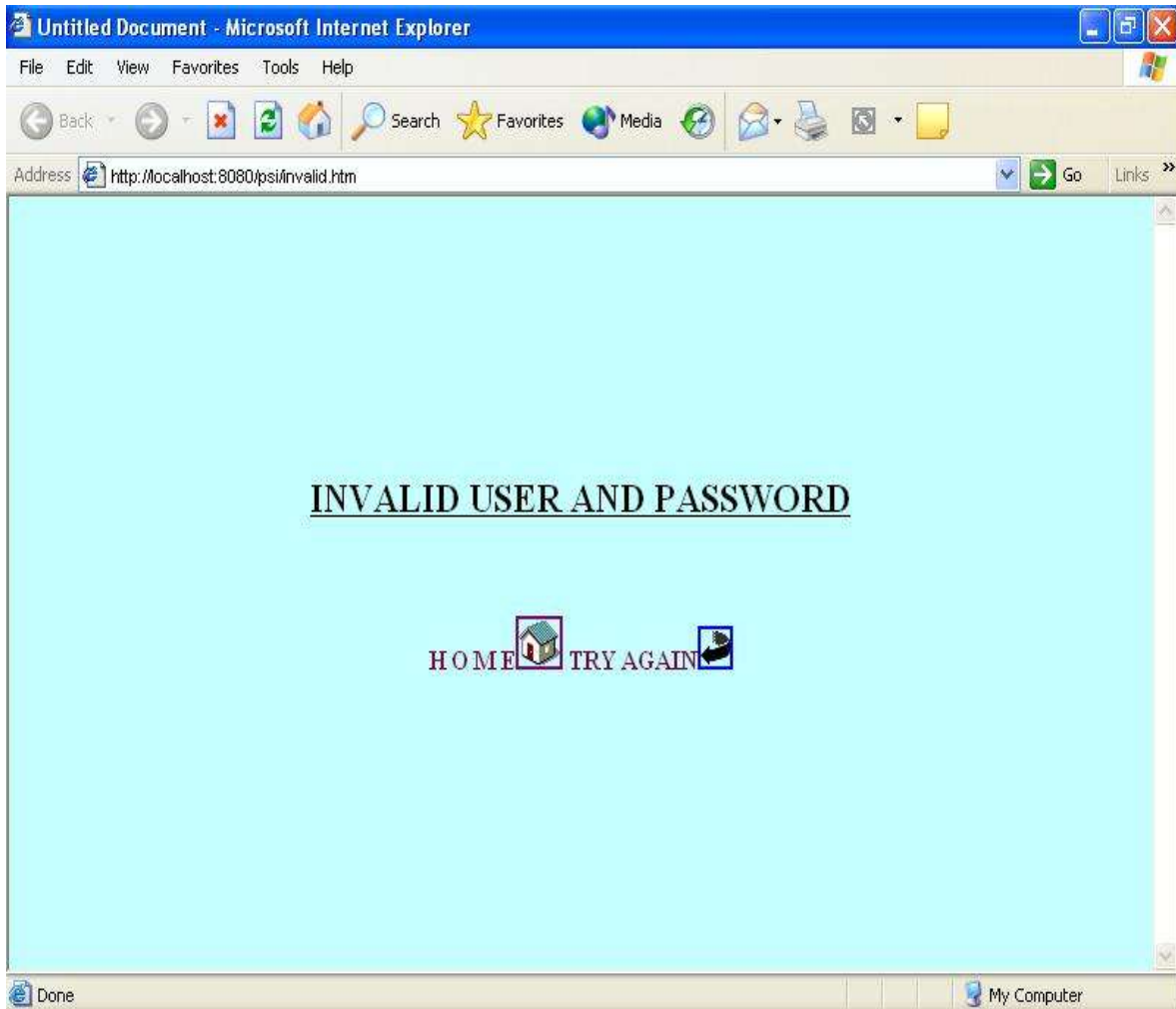
3) Oracle can perform effectively on large database. It offers unique clustering technique for storing data on the disk.

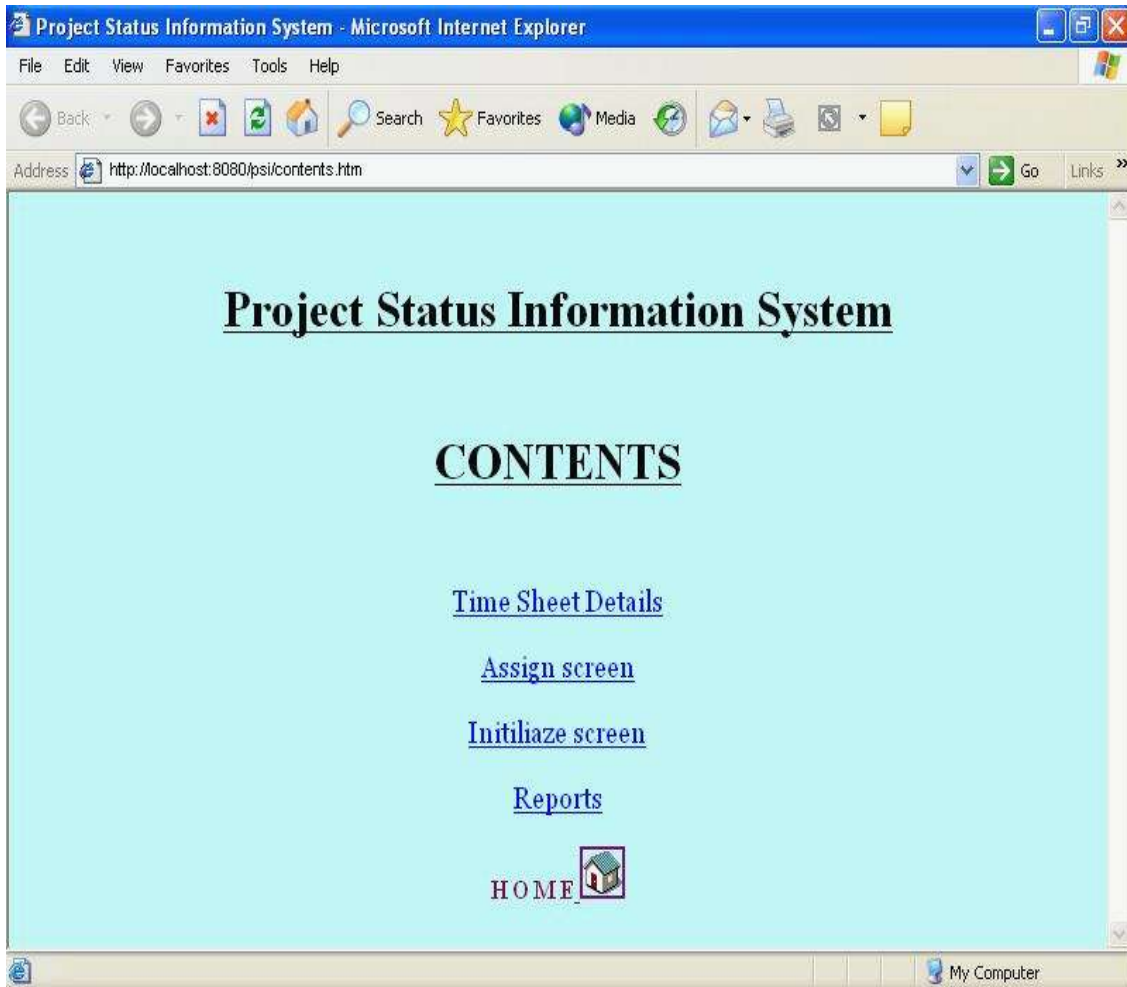
ORACLE database works effectively in an Intranet/Internet environment.

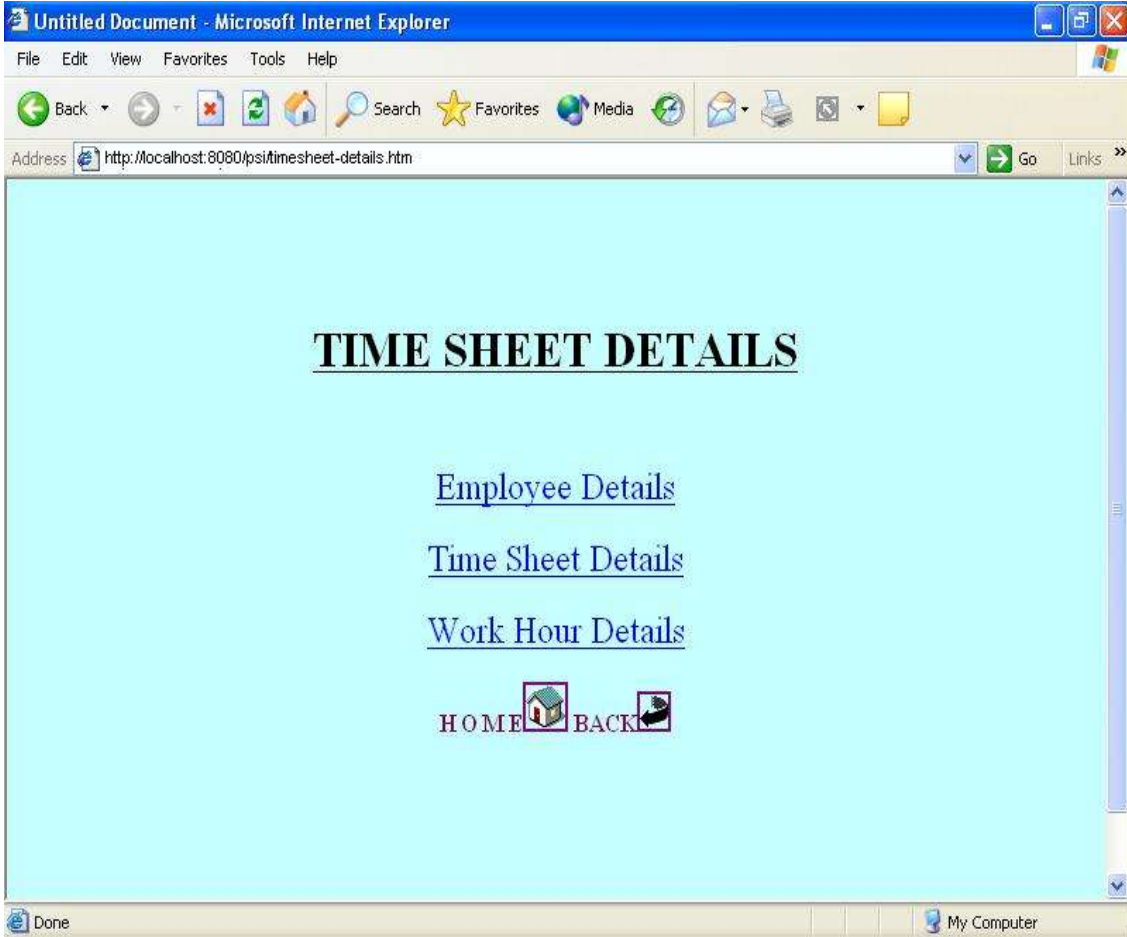
Since the Internet applications demands the data to be secure, ORACLE can be chose as the right tool for maintaining databases.

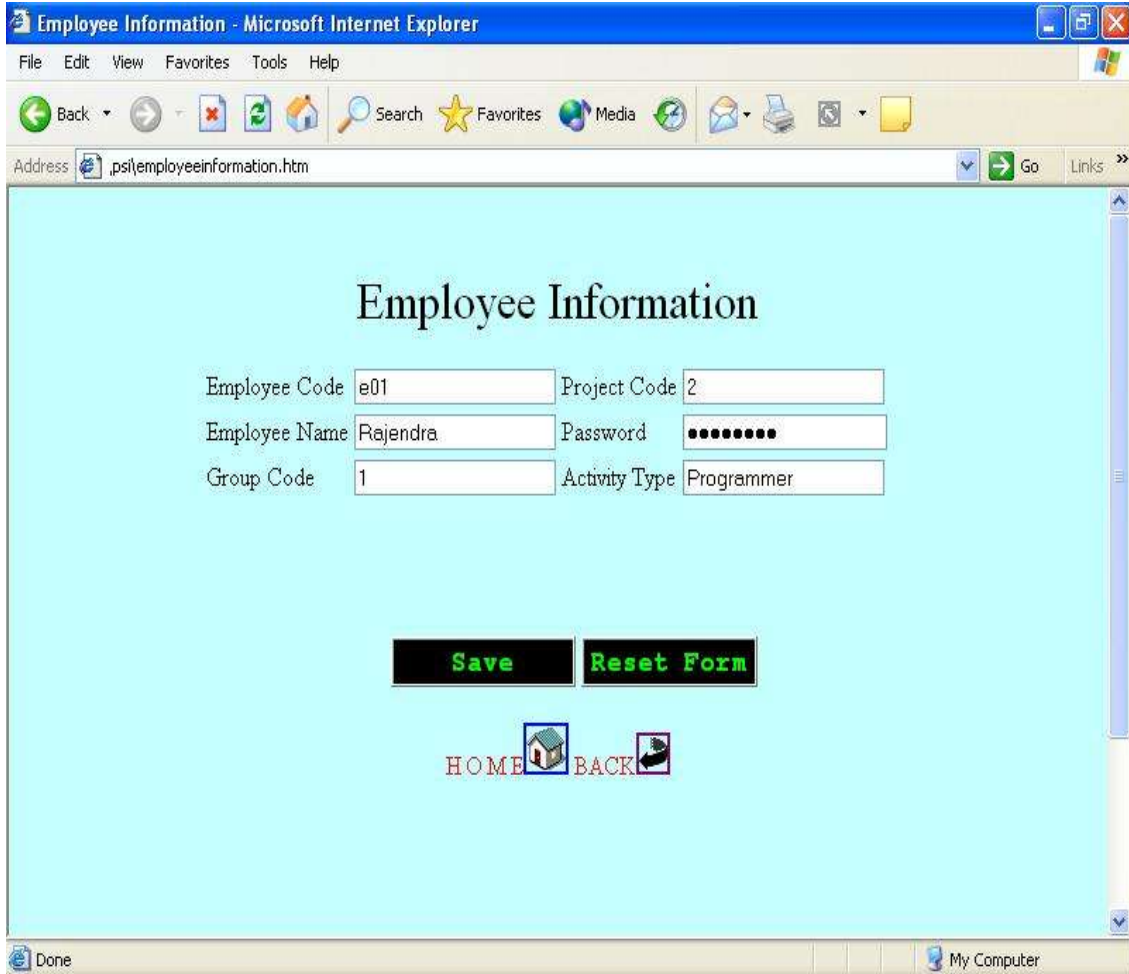
## OUTPUT:











C:\Program Files\Apache Group\Tomcat 4.1\webapps\psitime.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Stop Refresh Home Search Favorites Media Print Mail

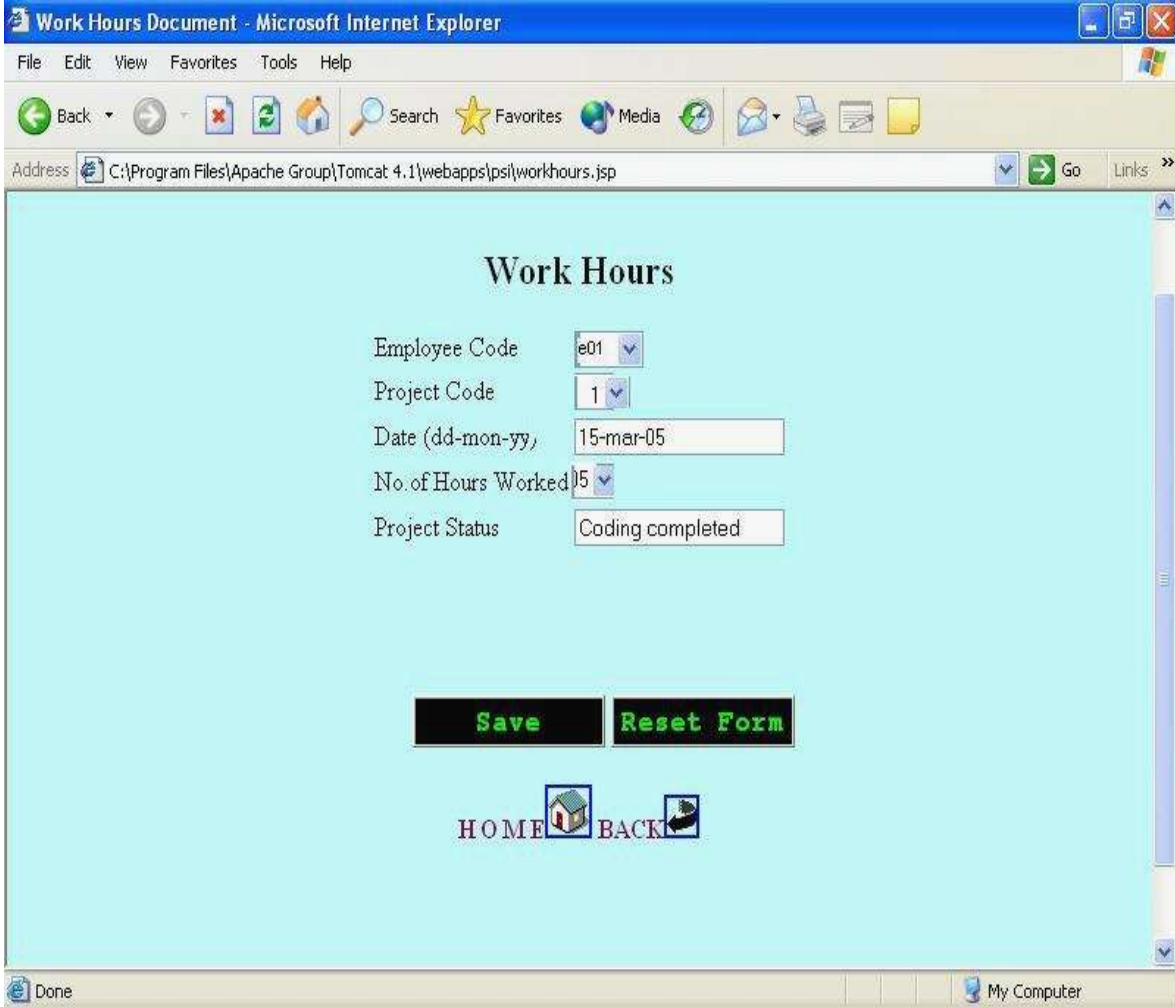
Address \psitime.jsp Go Links >>

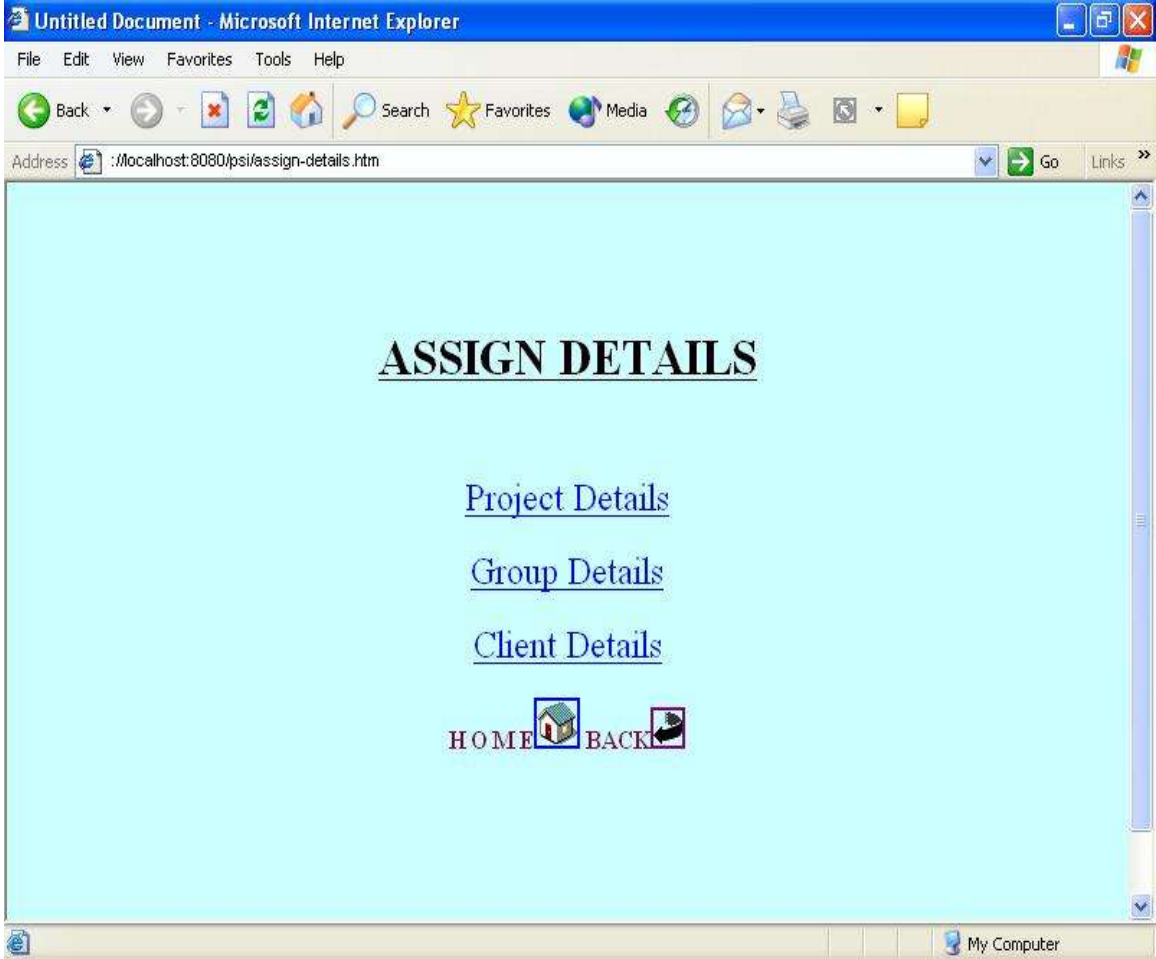
## TimeSheet Details

Project code	<input type="text" value="1"/>	Employee Code	<input type="text" value="je"/>
Proposed Start Date	<input type="text" value="12-jan-05"/>	Proposed Hours	<input type="text" value="100"/>
EmployeeName	<input type="text" value="Rajesh"/>	Actual Hours	<input type="text" value="75"/>
Actual Start Date	<input type="text" value="12-jan-05"/>	ProjectName	<input type="text" value="HRMS"/>
Actual End Date	<input type="text" value="9-apr-05"/>		

HOME BACK

Done My Computer





Untitled Document - Microsoft Internet Explorer



File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites Media Print

Address [/psij/project-details.htm](#) Go Links

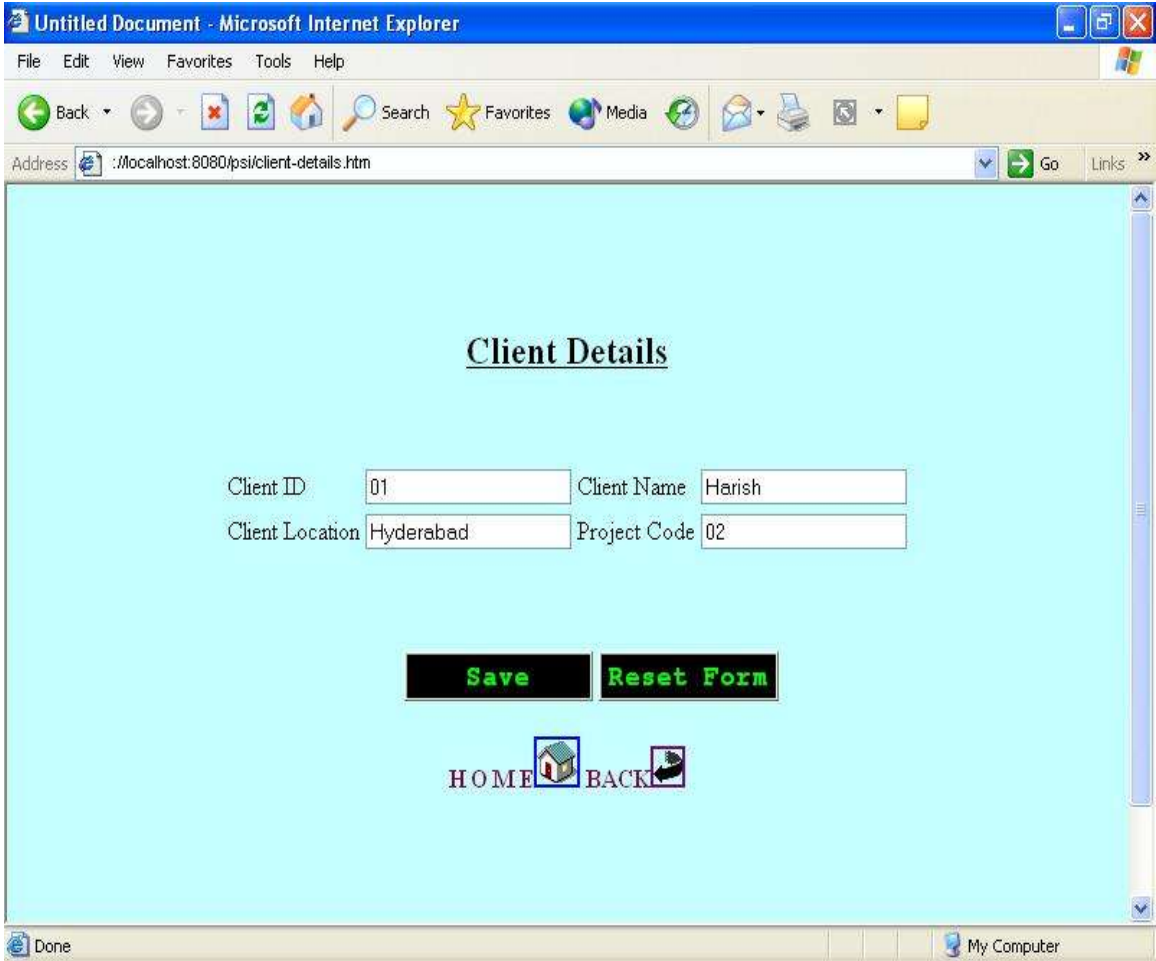
## Project Details

Project Code	<input type="text" value="02"/>	Project Name	<input type="text" value="HRMS"/>
Proposed Start Date	<input type="text" value="12-jan-05"/>	Project Manager	<input type="text" value="Ravi"/>
Proposed End Date	<input type="text" value="12-apr-05"/>	Team Leader	<input type="text" value="Kumar"/>
Proposed Hours	<input type="text" value="240"/>	Actual StartDate	<input type="text" value="12-jan-05"/>
Actual End Date	<input type="text" value="9-apr-05"/>		

HOME  BACK 

Done My Computer





# SOFTWARE TESTING

## TESTING AND IMPLEMENTATION:

In this the different modules of a system are integrated using an integration plan. The integration plan specifies the steps and the order in which modules are combined to realize the full system. After each integration step, the partially integrated system is tested. The primary objective of integration testing is to test the module interface.

An important factor that guides the integration plan is the module dependency graph. The module dependency graph denotes the order in which different modules call each other. A structure chart is a form of a module dependency graph. Thus, by examining the structure chart the integration plan can be developed based on any of the following approaches:

- Big-bang approach.
- Top-down approach.
- Bottom-up approach.
- Mixed approach Bottom-up Integration Testing

In this approach, each subsystem is tested separately and then the full system is tested. A subsystem might consist of many modules, which communicate among each other through well-defined interfaces. The primary purpose of testing each subsystem is to test the interfaces among various modules making up the subsystem. Both control and data interface are tested. A principal advantage of bottom-up integration testing is that several disjoint subsystems can be tested simultaneously. A disadvantage of bottom-up testing is the complexity that occurs when the system is made up of large number of small subsystems.

## SYSTEM TESTING:

Once we are satisfied that all the modules work well in themselves and there are no problems, we do in to how the system will work or perform once all the modules are put together. The main objective is to find discrepancies between the purpose and its original objective, current specifications, and system documentation. Analysts try to form moulds that have been designed with different specifications, which could cause incompatibility.

At this stage the system is used experimentally to ensure that all the requirements of the user are fulfilled. At this point of the testing takes place at different levels so as to ensure that the system is free from failure. Persons, who have never worked with the system before, so that the feedback we get is free from bias, mostly perform testing.

Testing is vital to the success of the system. System testing makes a logical assumption that whether all parts of the system are correct. Initially the system was given to the user for entry validation and was provided at each and every stage so that the user is not allowed to enter unrelated data. The training is given to user about how to make an entry.

While implementing the system it was observed that the user was initially resisting the change, however the system being the need of the hour and user friendly, the fear was overcome. Entering live data of the past months records was little tedious, prior to the actual day-to-day transactions.

The best test made on the system was whether it produces the correct outputs. All the outputs were checked out and were found to be correct. Feedback sessions were conducted and the suggested changes given by the user were made before the acceptance test. Finally the system is being accepted and made to run with live data. System tests are designed to validate a fully developed system with a view to assuring that it meets its requirements. There are three main kinds of system testing:

- Alpha Testing.
- Beta Testing.
- Acceptance Testing.

**Alpha Testing:** This refers to the system testing that is carried out by the test team with the organization.

**Beta Testing:** This refers to the system testing that is performed by a select group of friendly customers.

**Acceptance Testing:** This refers to the system testing that is performed by the customer to determine whether or not to accept the delivery of the system.

## **IMPLEMENTATION:**

The system implementation phase of the software deals with the translation of the design specifications into the source code. The ultimate goal of the implementation is to write the source code and the internal documentation so that it can be verified easily. The code and documentation should be written in a manner that eases debugging, testing and modification. System flowcharts, sample run on packages, sample output etc. is the part of the implementation.

An effort was made to satisfy the following goals in order specified.

- Minimization of Response Time.
- Clarity and Simplicity of the Code.
- Minimization of Hard-Coding.
- Minimization of the Amount of Memory Used.

Various types of bugs were discovered while debugging the modules. These ranged from logical errors to failure on account of various processing cases.

## **DOCUMENTATION:**

Documentation is a method of communication. A satisfactory documentation of the system should be objective, factual and complete. Thus its adequacy is not determined by format, length, volume or complexity. In documentation, there are no uniform standards that are applicable to all system projects. Documentation is essential to the development, implementation and operation of any system. Documentation is necessary as it helps in maintaining the system and also acts as a reference for the user.

Embedding comments in the executable portion of the code do proper documentation of each module. To enhance the readability of the comments, indentation, parenthesis, blank lines and spaces, proper lineation of the loops were used around the block of comments. Care was also taken to use descriptive names of tables, fields, modules, forms etc. The proper use of indentation, parenthesis, blank lines and spaces were also ensured during coding to enhance the readability of the code.

## **MAINTENANCE:**

Maintenance activities involve making enhancements to software products, adapting products to new environments, and correcting problems.

Software product enhancement may involve providing new functional capabilities, improving user displays and modes of interaction, upgrading external documents and internal documentation, or upgrading external documents and internal documentation, or upgrading the performance characteristics of a system. Adaptation of software to a new environment may involve moving the software to a different machine. Problem correction involves modification and revalidation of software to correct errors.

Maintenance activities consume a large portion of the total life cycle budget. Software Maintenance accounts for 70 percent of total software life-cycle costs. Maintenance includes 60 percent of maintenance budget for enhancement, and 20 percent each for adaptation and correction. The primary product attributes that contribute to software maintainability are clarity, modularity, and good internal documentation of the source code, as well as appropriate supporting documents.

Analysis activities during software maintenance involve understanding the scope and effect of a desired change, as well as the constraints on making the change.

Design during maintenance involves redesigning the product to incorporate the desired changes. The changes must then be implemented.

Internal documentation of the code must be updated, and new test cases must be designed to access the adequacy of the modification. Also the supporting documents must be updated to reflect the changes. Updated versions of the software must then be distributed to various customer sites, and configuration control records for each site must be updated.

Failure to recognize the true cost of a small change in the source code is one of the most significant problems in software maintenance.

## **CONCLUSION**

The project "Project Time Analysis and Reporting System" aims at accomplishing the task of allowing the project manager to maintain the project details. It also helps in maintaining the time details of each project.

The system provides a graphical user interface, which helps all the employees to know the project details.

It also generates reports, which gives detailed information about the clients of the company, different groups their size including the team leaders.

Future enhancements for this project can be also created using Bar-charts by which the performance of each project can be better analyzed and by using the resource allocation can be done efficiently.

## **BIBLIOGRAPHY**

1. JAVA-2 Complete-Reference

Author: Patrice Norton & Herbert Child Publisher: Tata McGraw Hill

2. System Analysis & Designing

Author: James A. Senn

Publisher: Tata McGraw Hill

3. Software Engineering Concepts

Author: Fairley

Publisher: (TATA MC-GRAW Hill publication, Third Edition)

4. Software engineering principles

Author: Pressman

5. Java Server Pages

Author: James good will Pubishentechmedia