

INDEX

Page No

1. ABSTRACT	
2. INTRODUCTION	
The Organization's Profile	
Introduction about Project	
3. SYSTEM ANALYSIS	
System Definition	
Software Requirement Specification	
Feasibility Study	
4. SYSTEM DESIGN	
Design	
E-R Diagram	
DFD'S	
Use case Diagrams	
Class Diagram	
Sequence Diagrams	
Collaboration Diagrams	
Data Dictionary	
5. SYSTEM ENVIRONMENT	
Java, java Script	
HTML	
JSP	
Oracle	
6. SCREENS	
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.

ORGANISATION PROFILE

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

Present System:

In the present system all the information is maintained in the manual records to maintain details of the organization. This system contains details of different projects which are undergoing in the organization, and this becoming difficult to maintain details of all the projects, employees working in different projects and its reports at the same time this manual process is always time taking. So in order to reduce all the drawbacks of the manual system we are for newly automated system.

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 REQUIREMENT SPECIFICATION:

OPERATING PLATFORM : WINDOWS 98/2000/NT

DATA BASE : ORACLE 8

SOFTWARE : JDK 1.3, TOMCAT(web server)

BROWSER : INTERNET EXPLORER

FRONT END TOOL : HTML

LANGUAGE :**JAVA 2.0**

DOCUMENTATION TOOL :MS – WORD 2000

HARDWARE REQUIREMENT SPECIFICATION:

PROCESSOR : Pentium

SPEED : 233 MHz

MONITOR : SAMTRON

HARD DISK : 4.2 MB

RAM : 128 MB

MOUSE : Logitech

KEY BOARD : TVS

REQUIREMENT STUDY

The origin of most software systems is in the need of a client, who either wants to automate an 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, thing's like user interfaces are rarely modeled, whereas they frequently form a 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 there 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 exists 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 an important because if the specification study is not done properly then following design phase etc will go wrongly.

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 it finds errors. The testing strategies are of two types, the code testing and the specifications testing. In the code testing we examining the logic of the program. On the surface, code testing seems to be ideal methods for testing software, but no tall software errors are uncovered.

DESCRIPTION:

DESCRIPTION OF EXISTING SYSTEM:

In the existing system the project details, client detail group details, and time used for various levels of project development maintained in registers.

They maintained the following fields in register empcode, project code, project name, client code, actual hours, proposed hours, actual start date, proposed start date.

DESCRIPTION OF THE PROPOSED SYSTEM:

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. Where 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 reports, getting reports of the required requirements is very fast and easy in computerized system

DATA COLLECTION ON SITE OBSERVATION:

For the study of the project I went for the various software

Development organizations There I met the Technical Manager, project Manager, project Leader and programmers. There we got the Information about various levels of project development and time assigned for each phase.

The different functions are time sheet analysis, assign activity analysis. The data all about the above are stored in different registers (project list register, group register, client register, time register) in different departments. In manual process whenever project came they Maintain registers to store project details, client details, time proposed for each phase like analysis, design

According to the requirements we divided this manual process Into 4 phases:

Time Sheet : This is used to allow entering the time

Sheet details on the basis of actual start date vs. planned start date

And actual end date vs. planned end date it also maintains

Actual hours for each phase vs. planned hours for each phase

According to requirements this object is accessible to programmers, team leader, project manager.

Setup : This is used to install Project Time Analysis and Reporting System for specific Project. According to the requirements this Object is accessible to Project Manager.

Assign Activity: This is used for assigning various activities to different Employees in an organization. According to the requirements this is accessible to Project Manager and other employees.

FEASIBILITY STUDY

Investigation

For the study of the project, I went for the various software Development organizations There I met the Technical Manager, project Manager, project Leader and programmers . There we got the Information about various levels of project development and time assigned for each phase.

The different functions are time sheet analysis , assign activity analysis. The data all about the above are stored in different registers (project list register, group register, client register, time register) in different departments.

In manual process whenever project came they maintain registers to store project details, client details, time proposed for each phase like analysis, design.

According to the requirements we divided this manual process Into 4 phases:

Time Sheet: This is used to allow to enter the time sheet details on the basis of actual start date vs. planned start date and actual end date vs. planned end

date it also maintains actual hours for each phase vs. planned hours for each phase according to requirements this object is accessible to programmers, team leader, project manager.

Setup: This is used to install Project Time Analysis and Reporting System for specific Project. According to the requirements this is accessible to Project Manager.

Assign Activity: This is used for assigning various activities to different Employees in an organization. According to the requirements this is accessible to Project Manager.

After problem is clearly understood and Solutions are proposed the next step is to conduct the feasibility study, which is the part of the system analysis. The main objective of this study is to determine whether the proposed system is feasible or not. Solutions strategies are usually generated \without regard for feasibility because one cannot be both creative & critical at the same time. Hence feasibility analysis is a must to arrive at the most appropriate solution strategy. This feasibility analysis is done after the thorough study of the system.

The planning objective is achieved through a process of information discovery that leads to reasonable estimation.

Keeping the following aspect in mind the system analysis for the development of the project

- Identify the drawbacks of the existing system.

- Identify the need for conversion.

- Perform feasibility study.

- Identify hardware, software and the database requirements.

- Create a system definition that forms the foundation for the subsequent work.

Software scope:

The first activity in software project planning is the determination of software scope. Function and performance allocated to software should be assessed to establish a project scope. Software scope describes function, performance, constraints and reliability.

Estimation :

Software project planning estimate can do many ways, such as LOC (line of code)-based estimation, FP (function point) estimation. Process-based estimation. We should select best of one depending on the requirement.

Resource availability:

The resources, which are common to any system consist, of human effort, information and development resource. Developing any system any system without satisfactory resources is inappropriate and impossible.

The design development team for this project consists of three members who are responsible for software development and internal testing. Information assembly is to this project as with every other project. Various sources of Information are available.

Human resources:

The number of people required for a software development can be determined only after an estimate of development effort. The planner begins by evaluating scope and selecting the skills required to complete development.

Reusable software resources:

Existing software that can be acquired from a third party or that has been developed internally for past project.

Environmental resources:

The environment that supports the software, often called a *software engineering environment* (SEE). Hardware provides the platform that supports the tools required.

Operational feasibility:

It determines how much effort will go into educating, selling and training the user staff on a candidate system. People are inherently resistant to change and computers have been known to facilitate changes. An estimate should be made of how strong a reaction the user staff is likely to have towards the development of a computerized system.

Operational feasibility checks the operational scope of the system. The system under consideration should have enough operational reach. It is observed that the proposed system would provide a very interactive means to share information and have a far and wide reach. The proposed system would make the information more interactive. Thus operational feasibility of the proposed system is found to be high.

Technical feasibility:

Technical feasibility centers on the existing computer system and to what extent it can support the proposed addition. The benefits such as high accuracy, minimum response time and user friendliness of the proposed system outweighs cost for designing and installing the new system.

The level of technology comprises of the software tools, the development platform and the machine environment, which already exists within the organization. Since all the required tools and platform that were mentioned above were already available, the technical feasibility of this project is justified.

The procedure in the economic feasibility analysis is to determine the benefits and savings that we expected from a candidate system and compare them with cost. If benefits outweigh cost then design decision is made to design and

implement the system. The major disadvantages of the existing system are slow transmission of information between various locations and increased average waiting time. By the candidate system we can reduce processing charges and time of processing. By comparing the benefits with the cost of the proposed system benefits outweigh costs.

DESIGN PHASE

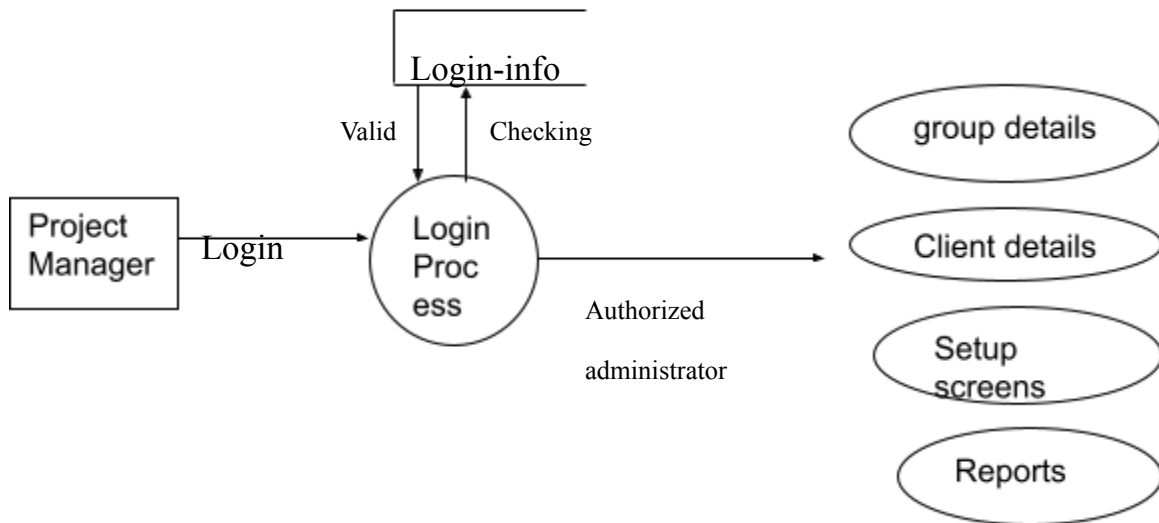
INTRODUCTION:

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 insufficient detail to permit its physical realization.

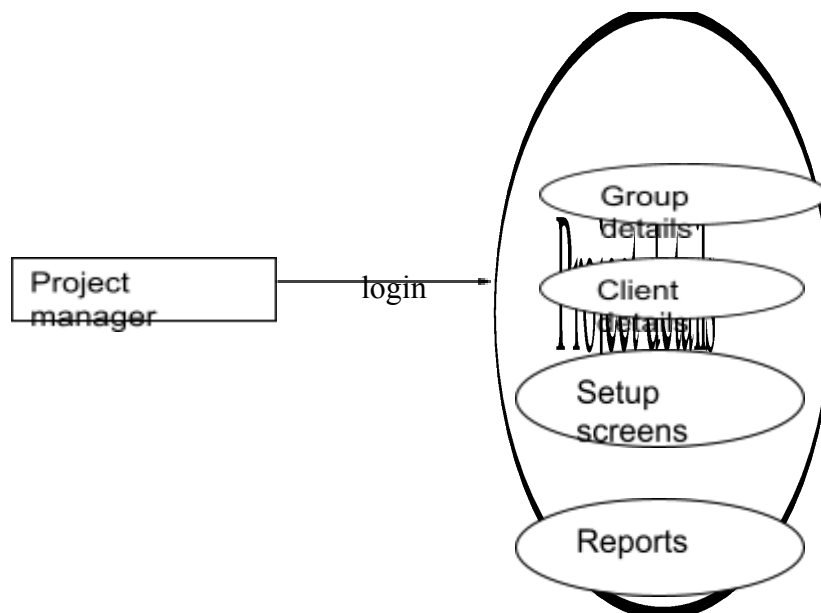
In the design phase the detailed design of the system selected in the study takes place. System design is stated by reviewing the study phase activities and making decisions as to which functions are to be performed by hardware, software or manually. This process through which the initial requirements 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.

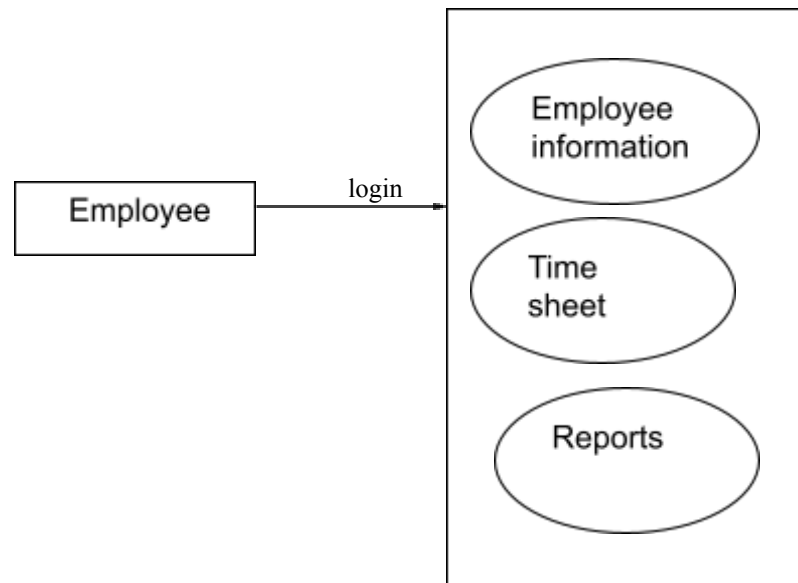
DFD'S



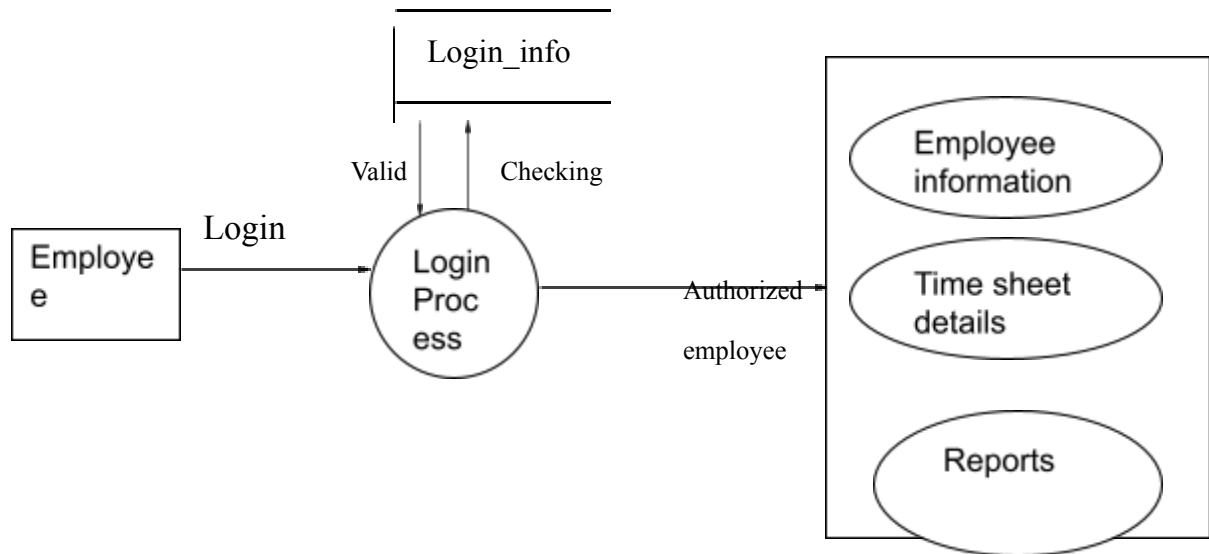
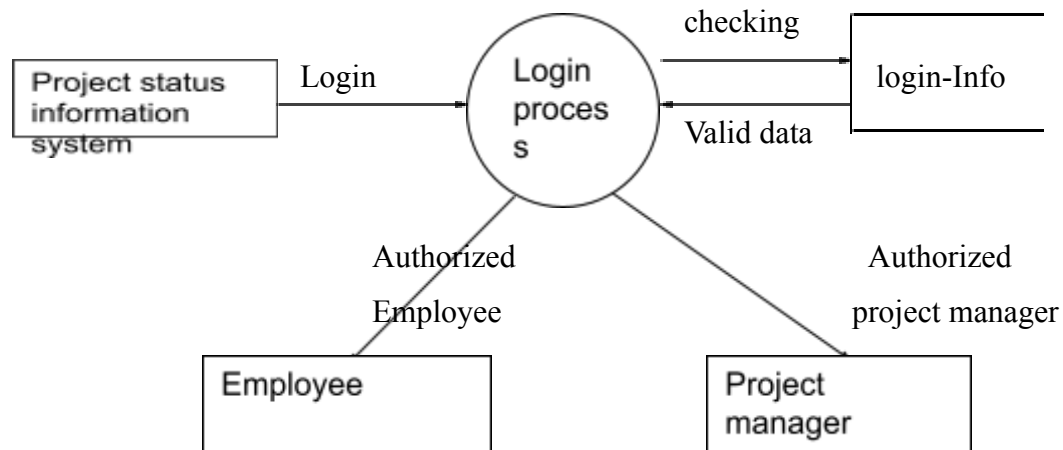
Context level DFD—Project Manager



Context level DFD –Employee



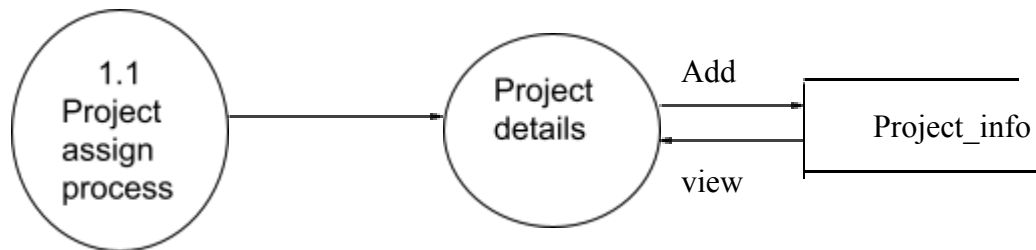
First-level DFD'S:



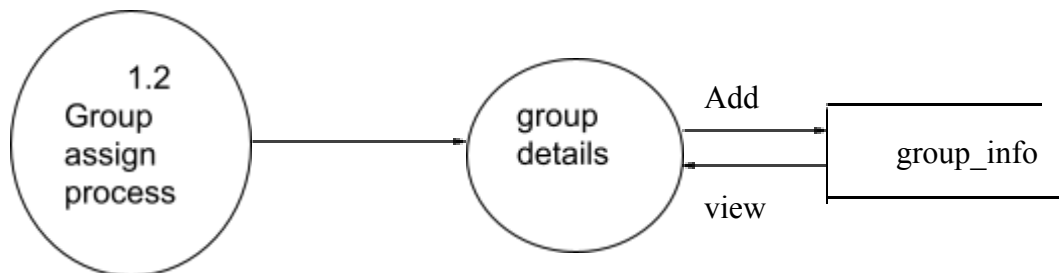
Level-II DFD –Project Manager:



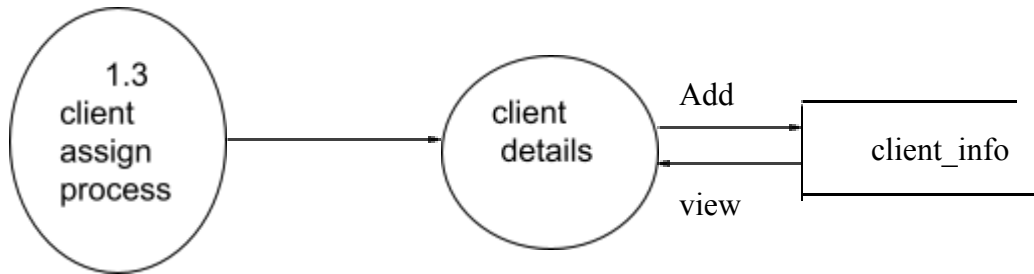
Project Assign



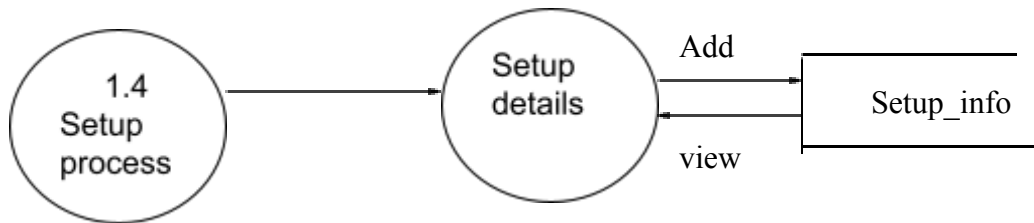
Assign Group



Project Client

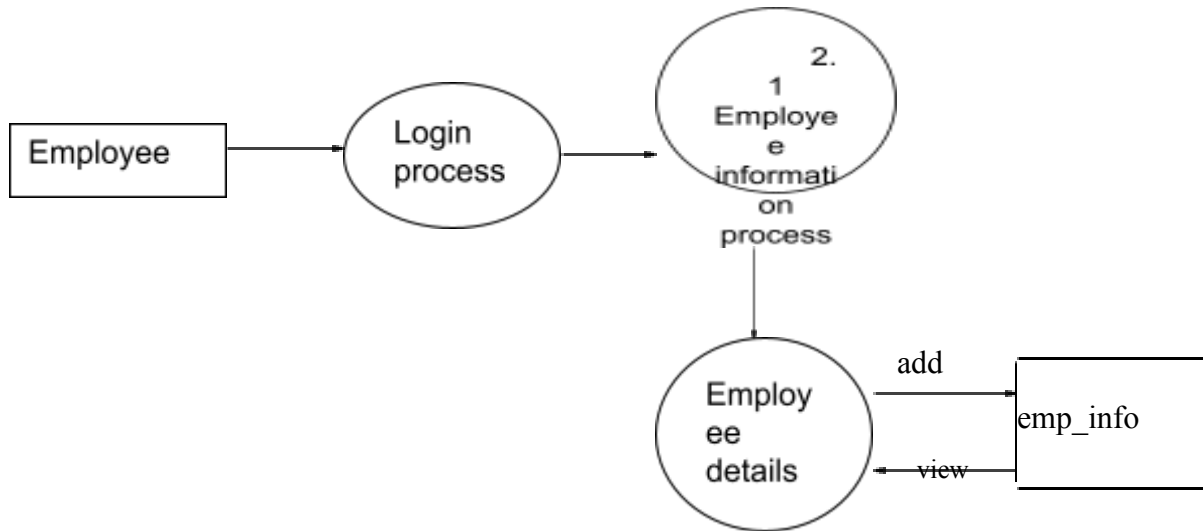


Setup Screens

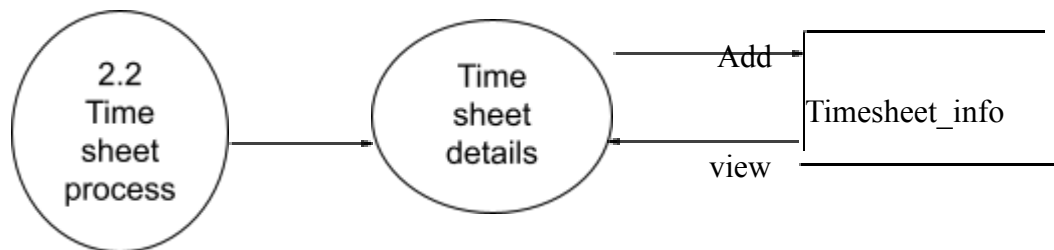


Level-II DFD –Employee

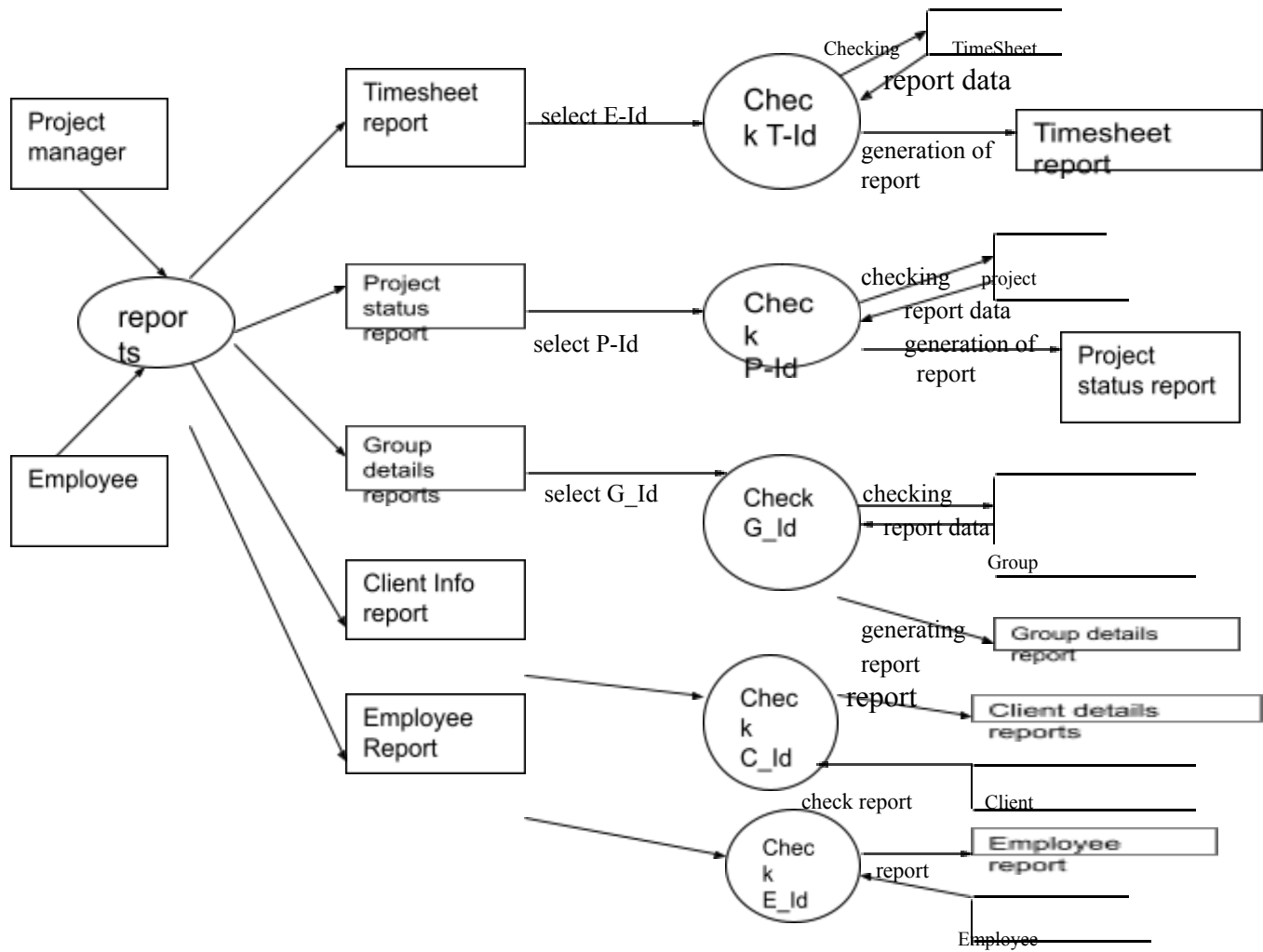
2.1 Employee information



2.2 Time Sheet Details

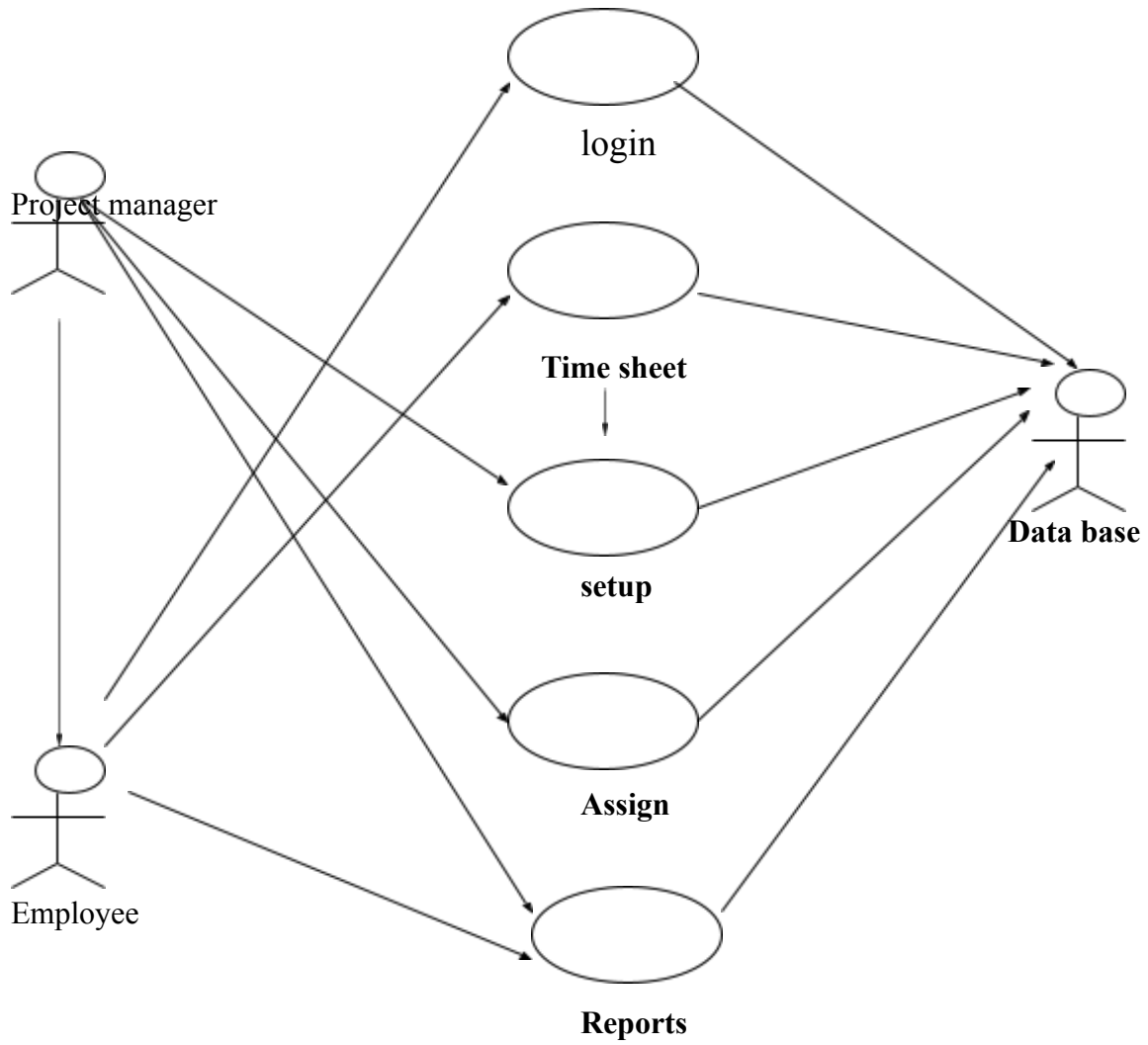


Reports



UML Diagrams

Use case diagrams



DATA DICTIONARY

FEATURES OF DATA DICTIONARY

The Volume of data in most Information system applications is substantially more than a single user can easily keep track of data dictionaries are an integral component of structural analysis, since data flow diagrams by themselves do not fully describe the subject of investigation. The data dictionary provides additional information about the system.

What is Data Dictionary?

A Data Dictionary a repository of the elements in a system. As the name suggests, these elements center around data and the way they are structured to meet user requirements and organization needs. In a data dictionary we will find a list of all elements composing the data flowing through a system. The major elements are data flows, data stores and process the Data Dictionary stores details and descriptions of these elements.

Why is a data dictionary important?

Analysts use data dictionaries for five important reasons.

1. To manage the detail in large systems.
2. To communicate a common meaning for all elements.
3. To document the features of the system.
4. To facilitate analysts of details in order to evaluate characteristics and determine where system changes would be made.
5. To locate errors and omissions in the system.

Manage Detail

Large systems have huge volumes of data flowing through them in the form of documents, reports and even. Similarly many different activities take place that use existing data or create new details. The best organized or most effective analysis use automated data dictionaries designed specifically for systems analysis and design.

Communicate Meaning

Data dictionaries assist in ensuring common meanings for system elements and activities. Data dictionaries record additional details about the data flow in a system; so that the persons who are involved may look up the description of data flows, data stores and processes. Document System Features

Documenting the features of an information system is the third reason for using the data dictionary systems. Features include the parts or components and the characteristics that distinguish

each. Having to formally describe, system features will produce a more complete understanding.

Facilitate Analysis

The Fourth reason for using data dictionaries is to determine whether new features are needed in a system or whether change of any type is in order. For any situation, system analysts will typically focus on the following system characteristics.

- a) Nature of transaction
- b) Inquiries
- c) Output and Report generation

TABLE NAME: PROJECT

Column Name	Description	Data-Type(size)	Constraints
pro_code	Code of the Project	number	primary key
pro_name	Name of the Project	varchar2(15)	not null
pro_mana	Project Manager	varchar2(15)	not null
tec_mana	Technical Manager	varchar2(15)	not null
sta_date	Starting Date	varchar2(10)	not null
pro_sdate	Proposed starting date	varchar2(10)	not null
pro_edate	Proposed ending date	varchar2(10)	not null
act_sdate	Actual starting date	varchar2(10)	not null
act_edate	Actual ending date	varchar2(10)	not null
act_type	Activity type	varchar2(15)	not null
pro_hours	Proposed hours	number	not null
act_hours	Actual hours	number	not null

TABLE NAME: **PRO_GROUP**

Colum n Name	Description	Data-Type(size)	Constraints
gro_co de	Group code.	number	primary key,
gro_na me	Group Name	varchar2(20)	not null,
pro_sd ate	Proposed srarting date	varchar2(10)	not null,
pro_ed ate	Proposed ending date	varchar2(10)	not null,
pro_co de	Code of the project	number	references project(pro_code)

TABLE NAME: **Employee**

Column Name	Description	Data-Type(size)	Constraints
emp_code	Employee code	varchar2(10)	primary key
emp_name	Employee name	varchar2(25)	not null
password	Password	varchar2(10)	not null
Desg	Designation	varchar2(20)	not null
pro_code	Project code	number	references project(pro_code)
Acti	Type of the activity	varchar2(100)	not null

pro_code	Project code	number	references pro_group(gro_code)
----------	--------------	--------	-----------------------------------

TABLE NAME: **client**

Column Name	Description	Data-Type(size)	Constraints
cli_id	Client ID	number	primary key,
cli_name	Client name	varchar2(20)	not null,
cli_loca	Client location	varchar2(20)	not null,
Pro_code	Project code	number	references project(pro_code)

TABLE NAME: **time_analysis**

Column Name	Description	Data-Type(size))	Constraints
emp_code	Employee code	varchar2(10)	references employee(emp_code)
pro_code	Project code	number	references project(pro_code)

emp_name	Employee name	varchar2(25)	not null
pro_name	Project name	varchar2(20)	not null
pro_sdate	Project starting date	varchar2(10)	not null
star_date	Starting date	varchar2(10)	not null
act_sdate	Actual starting date	varchar2(10)	not null
act_etype	Actual ending date	varchar2(10)	not null
pro_hours	Proposed hours	number	not null
act_hours	Actual hours	number	not null
act_type	Activity type	varchar(15)	not null

TABLE NAME:setup

Column Name	Description	Data-Type(size)	Constraints
emp_code	Employee code	varchar2(10)	references

pro_code	Prpject code	number	employee(emp_code),
Desg	Designation	varchar2(15)	references
scr_rights	Screen rights for the employees	varchar2(20)	project(pro_code),
Priv	Privilizes to the employees	varchar2(20)	not null,

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 web's 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 a stack of magazines in front of us, we are likely to flip through them all. Web pages are like magazine pages, except that they are more 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. I still use WordPerfect 5.1. Some people still use CP/M. I have a friend who played Doom several hours a day for months.

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 in bringing Java to the web, have elegantly solved a much broader and more significant problem- how to develop network-capable

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 (o s), write software for that o s, and eventually migrate it to the other o s 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 o s 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 o s 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 its 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 AND MODULARITY

Application Java supports the partition of a large program into modules. Specifically, a Java program consists of number of classes. If the program is properly designed, these classes will reflect encapsulation and information hiding. In Java, grouping sets of related classes into packages. Could enhance encapsulation and information hiding.

JAVA AND STRUCTURED PROGRAMMING

Java supports the three control structures of structured programming. Unlike c and c++, there is no go statement in Java. There is essentially no need for a go because Java supports the labeled break and continue statements.

JAVA AND PORTABILITY

Portability is critical to success in the emerging world of networked applications and commerce. Java takes a multi programmed approach to this challenge. At the heart of this approach is the fact that the Java compiler generates byte codes that are intercepted at runtime. The fact that byte codes are generated is important because it avoids the problem of baring the binary code on a basic set of primitive types such as integers and floating point-which would be tied to a specific platform.

The byte code based system is important to writing a portable interpreter. The byte codes generated by the compiler are based on the specification of a Java Virtual Machine which, as its name suggests, is not a specific hardware platform but a machine implemented in software. The virtual machine is very similar to real C.P.U with its own instruction set, storage formats and registers. Since it is written in software, however it is portable.

JAVA AND HIGH PERFORMANCE

The reason why Java's portable solution is such a coup is that interpreted platforms have generally been very slow. Often their performance is so poor the systems based on interpreters have been unusable. Java's byte code system however, provides a "lean and mean" interpreted solution.

One of the key features that Java offers to improve performance is multithreading. Most interactive applications are characterized by large periods of time during which the user passes between actions to decide what to do next.

JAVA AND SECURITY

Unlike C++, Java does not support explicit pointer variables. Even without explicit pointer variables, is possible to access illegally if a byte code has been doctored appropriately. Java precludes this sort of security attack as well; the Java interpreter checks each byte code before interpreting it. There is still one loophole however, malefactor can modify the Java byte code interpreter to disable this byte code checking.

JAVA AND REALLABILITY

One way to ensure that a program is reliable is to implement it in a strongly typed language. In such a language, every variable has to be of a specific type, a variable can take on only those values has permitted for that type, and the only operations that can be performed on that variable are operations permitted for variables of that type. One consequences is that when a method is invoked within a strongly type language, the type of each actual parameter has to match that of the corresponding formal parameter.

JAVA AND CONSTANTS

Java is pure object-oriented language. Thus, it does not support constants as such. Instead, constants have to be defined as final variables within a class. It is some what confusing for a constant to be defend as a "variable".

Unlike C and C++, Java also does not support macros such as # define that can be used, in conjunction with a preprocessor, to declare the value of a constant. There is a good reason for this. Java applets are design to be independent programs loaded over the Internet. Java therefore does not support header files or other components that need to be available in conjunction with otherwise stand-alone programs.

JAVA AND OBJECT ORIENTATION

Java supports the five basic elements of Object-Oriented language, namely, classes, objects, inheritance, polymorphism, and dynamic binding. Java is a pure Object-Oriented language, it does not support the classical paradigm. Thus there are no functions or procedures, only methods or classes.

JAVA AND REUSABILITY

All Object-Oriented programming language supports reuse via inheritance of classes. In addition, mechanisms such as templates (in C++) and generics (in Adam) provide reuse at a higher level of abstraction. A template or generic is a module that is defined in terms of parameters. The generic is then instantiated for a particular set of parameters.

Java does not directly supports template of this kind. Java interfaces are similar to templates, but they are flexible and less powerful than templates. Nevertheless, they do provide a level of abstraction that can increases main trainability and reusability.

JAVA AND VISIBILITY

In Java, however, there are five types of access and the rules are more convoluted than in C++ because Java classes can be grouped into package of related classes, the simplest Visibility modifier is public, then it can be accessed from any where within the product. A protected attribute can be accessed by immediate derived classes. Private attribute cannot be accessed outside the class.

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 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. figure below shows a model of a single-tier application.

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. figure shows a model of a two-tier application.

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.PreparedS
tatement 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 are 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 effected 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 implemented 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 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 as 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.

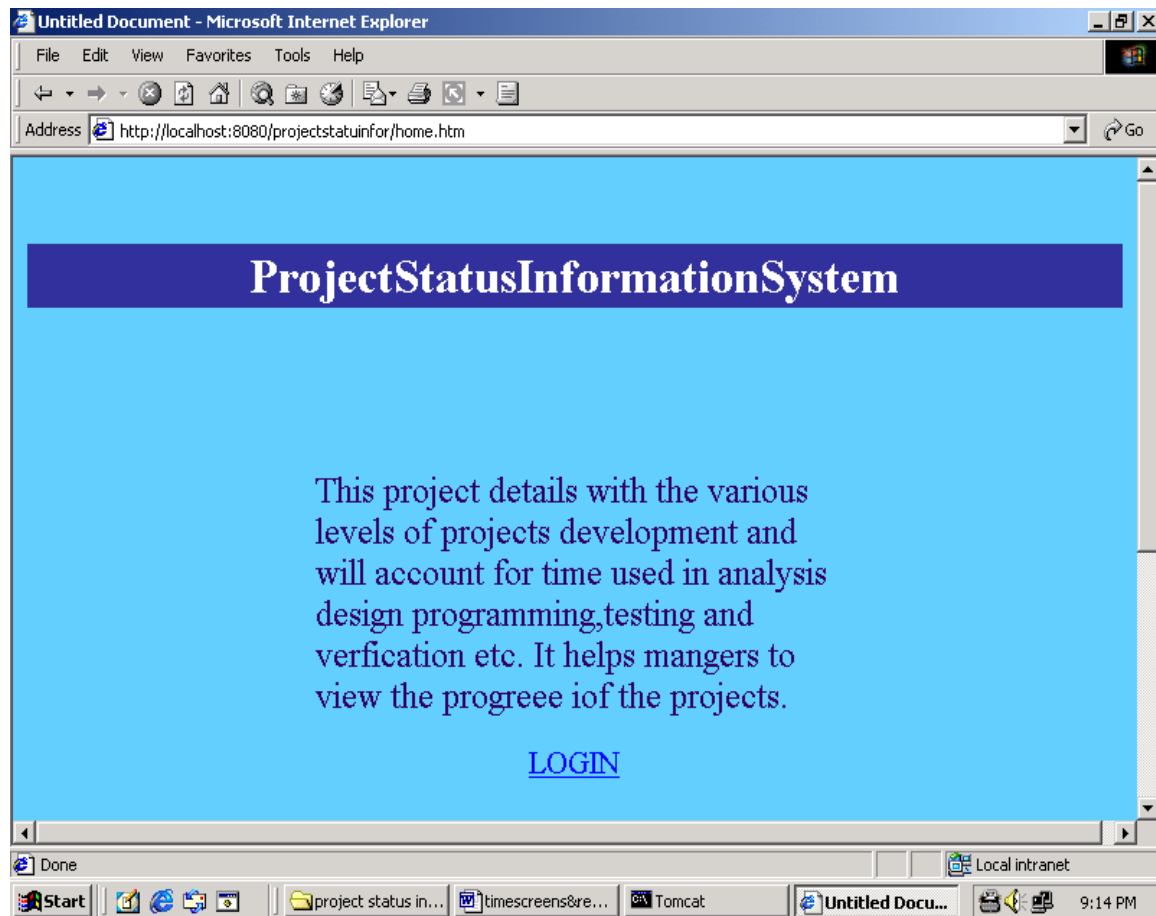
2) Oracle has several features that ensure integrity of user database. Database can be made secure by restrictions access to only authorized users using the tools available in ORACLE.

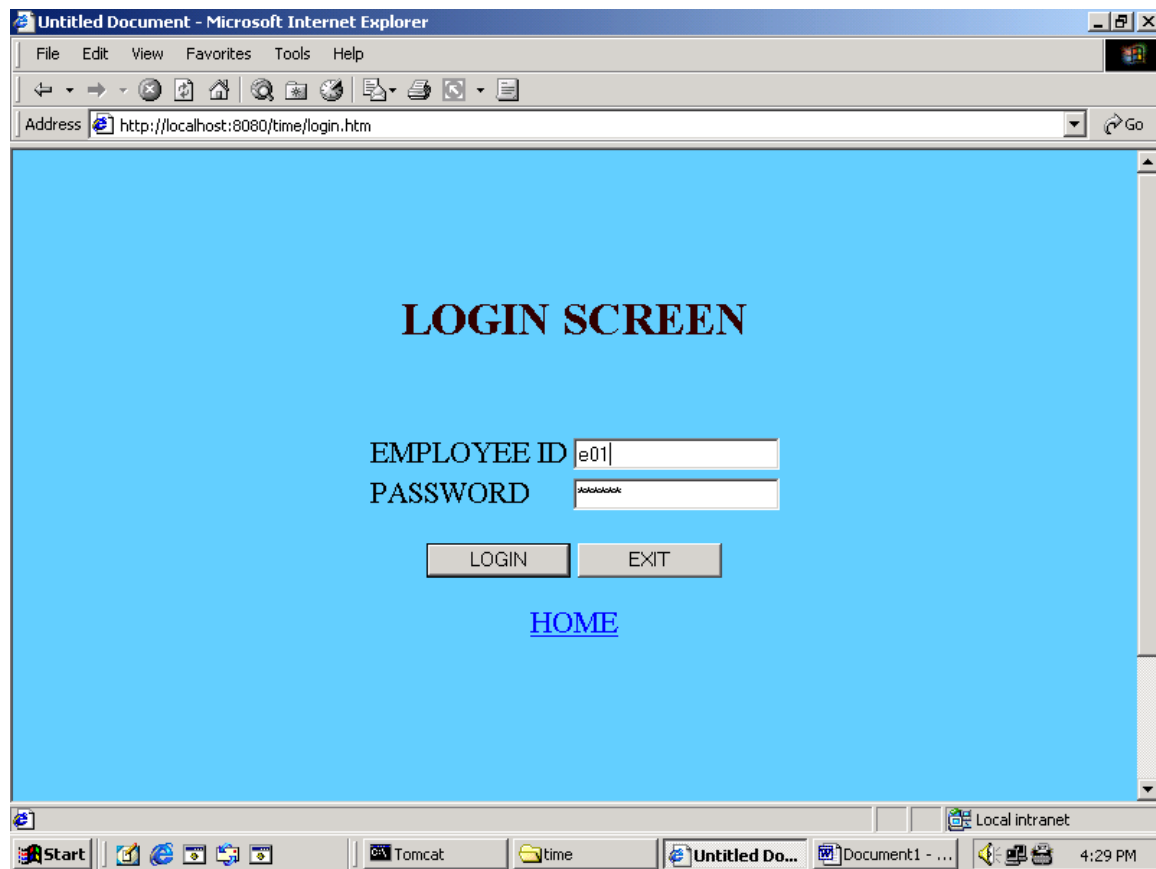
3) Oracle can perform effectively on large database .It offers unique clustering technique for storing data on the disk.

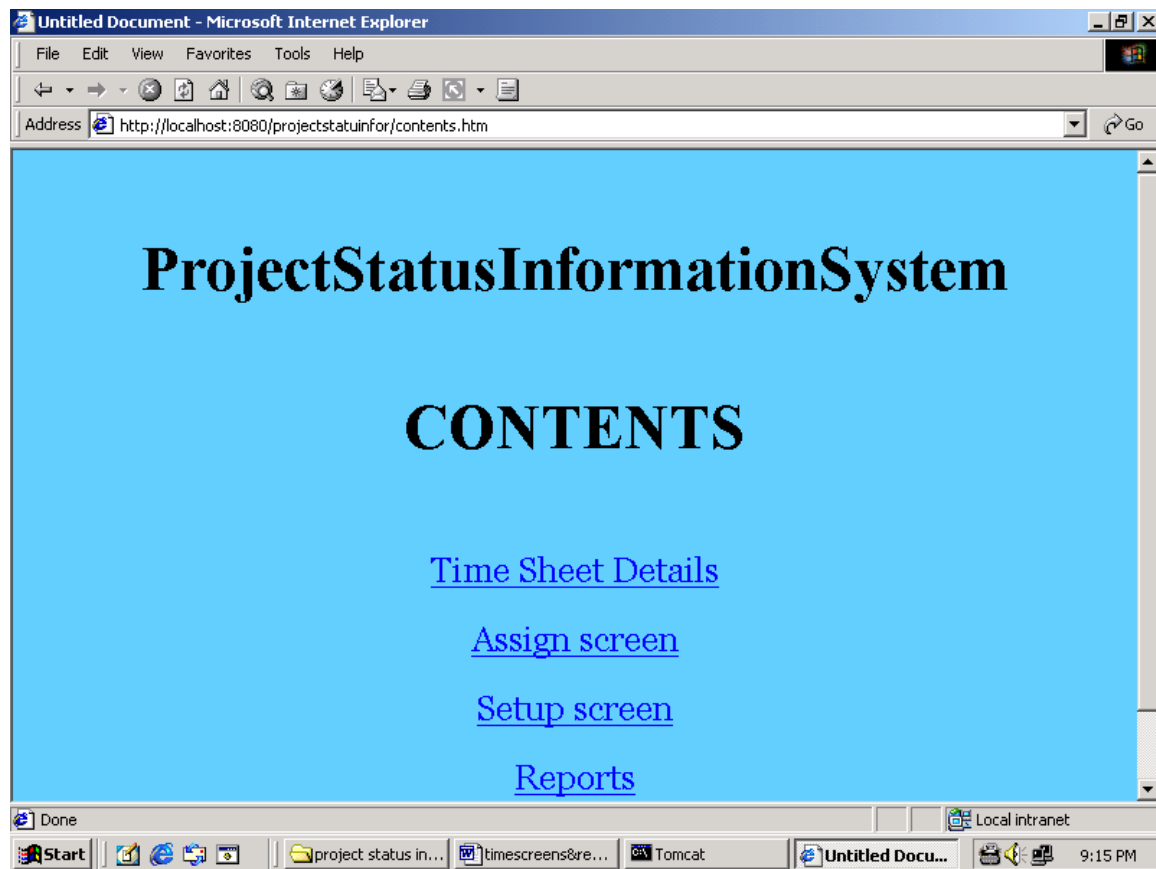
ORACLE database works effectively in an Intranet/Internet environment.

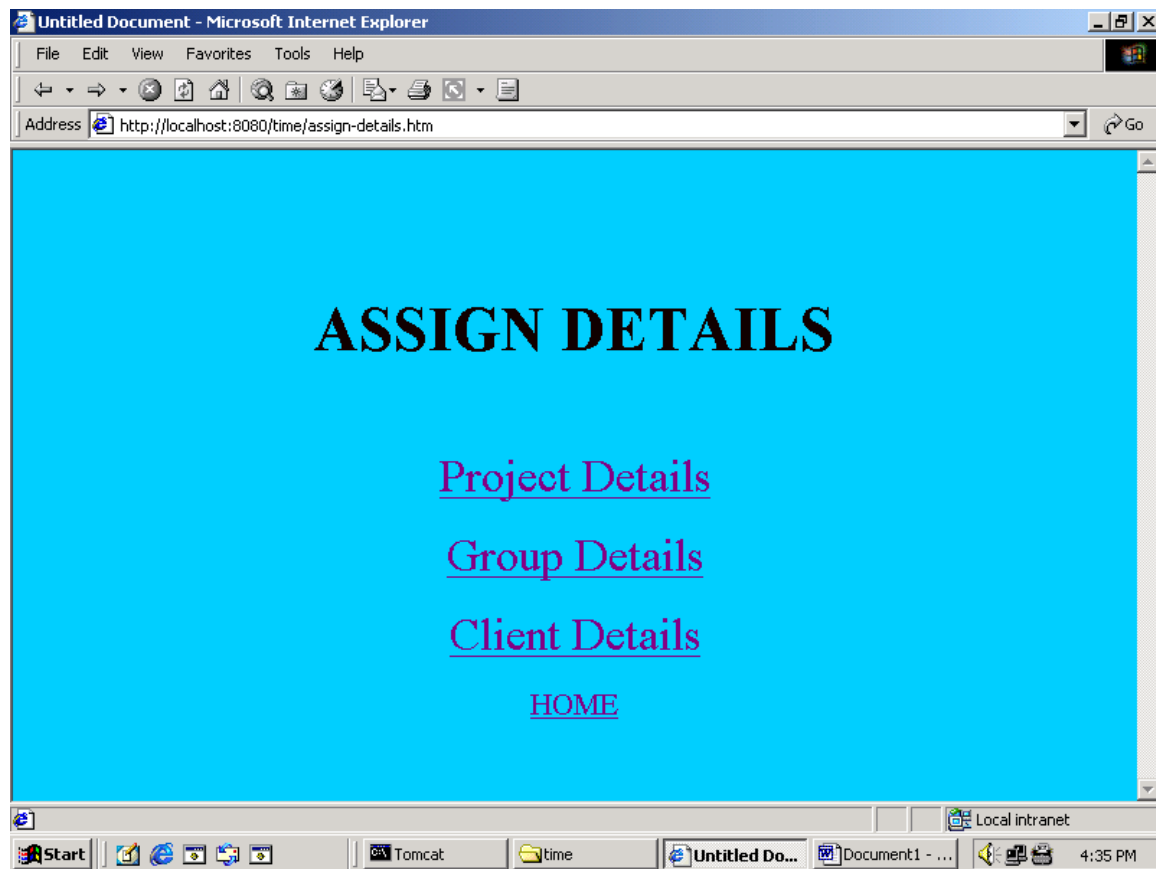
Since the Internet applications demands the data to be secure, ORACLE can be chosen as the right tools for maintaining databases.

SCREENS & REPORTS









Untitled Document - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://localhost:8080/time/project-details.htm> Go

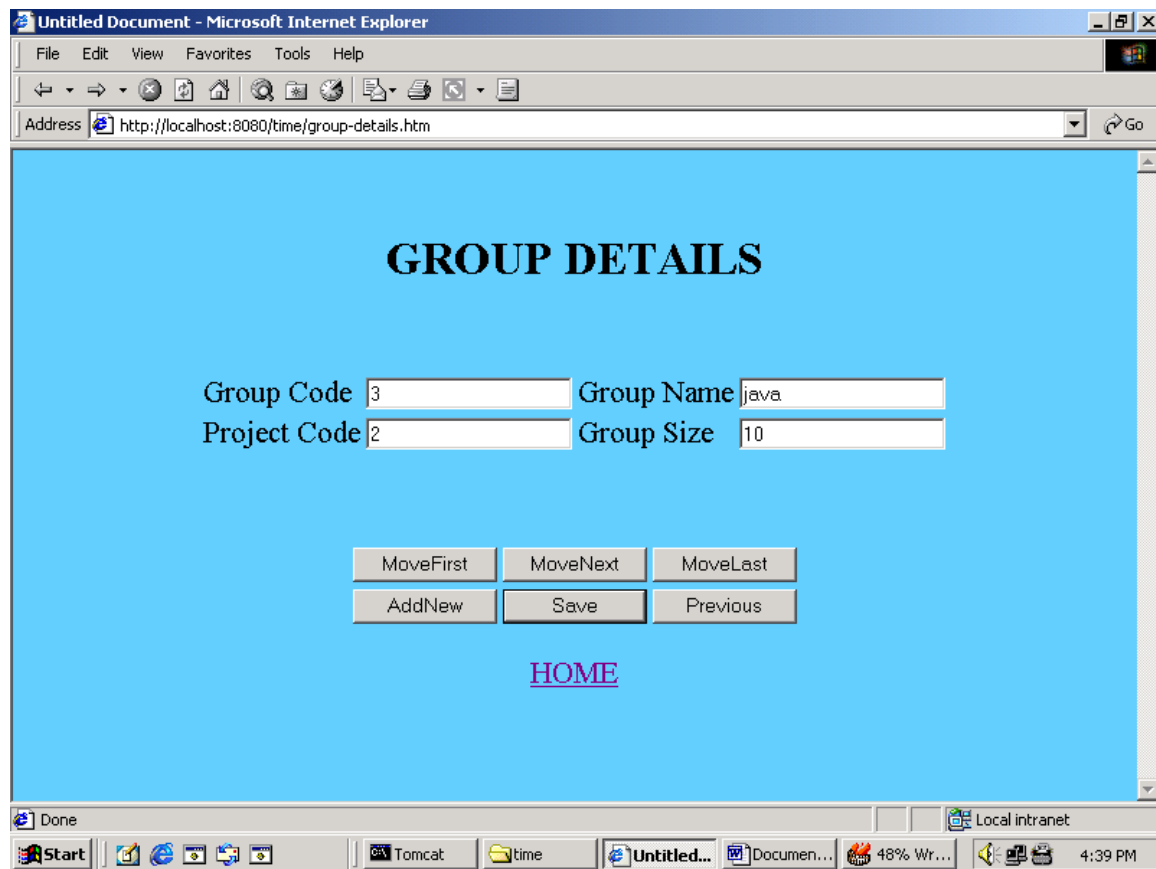
Project Details

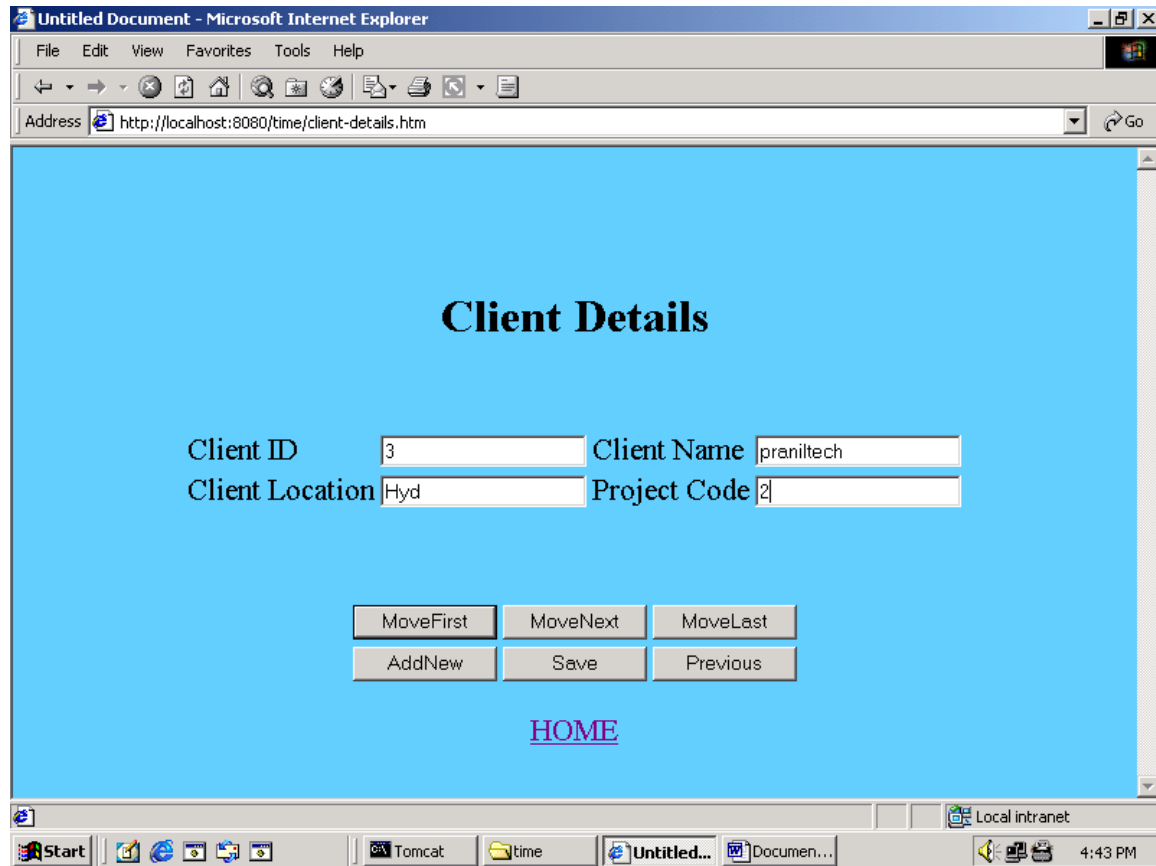
Project Code	e03	Status Date	02-may-2004
Project Name	TimeTrackingSystem	Proposed Start Date	11-dec-2003
Project Manager	vittal	Proposed End Date	22-feb-2004
Team Leader	venkat	Proposed Hours	100
Actual StartDate	11-dec-2003	Actual Hours	75
Actual End Date	22-feb-2004		

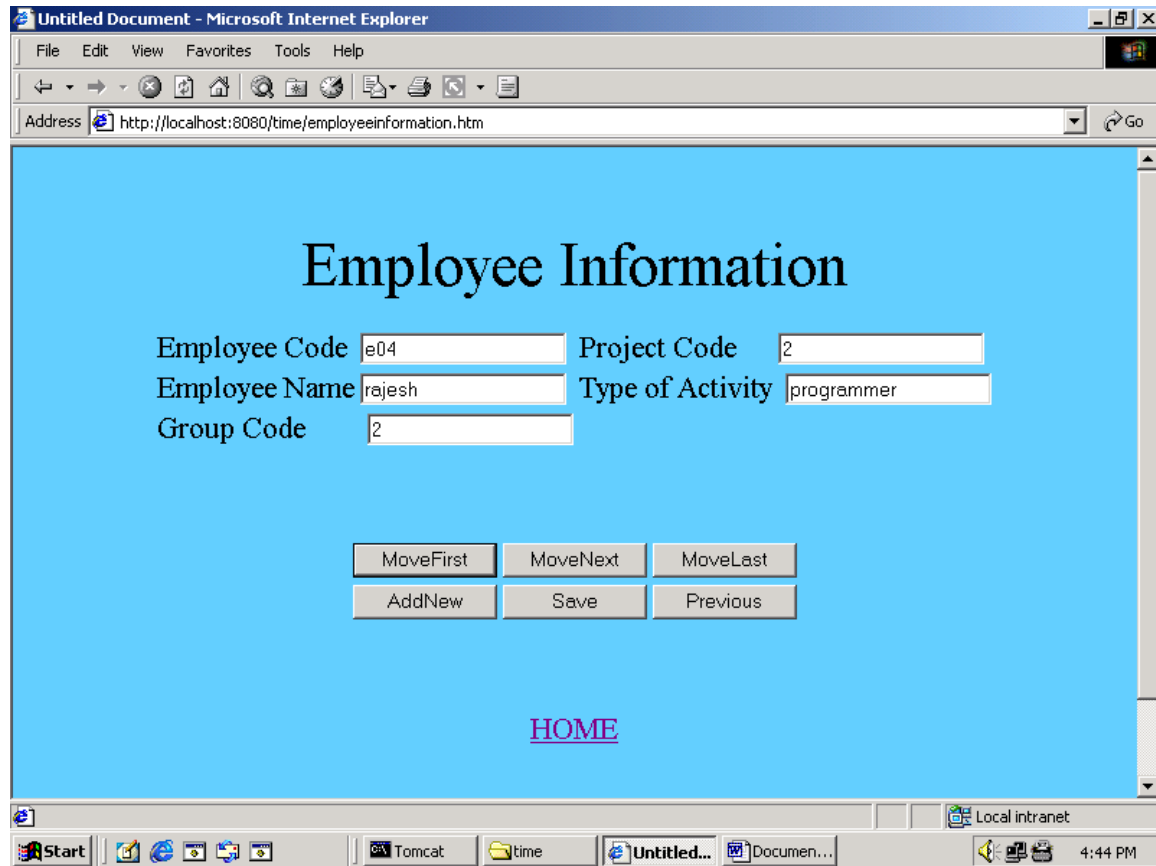
[HOME](#)

Done Local intranet

Start Tomcat time Untitled Do... Document1 - ... 4:33 PM







Untitled Document - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://localhost:8080/time/time.htm> Go

TimeSheet Details

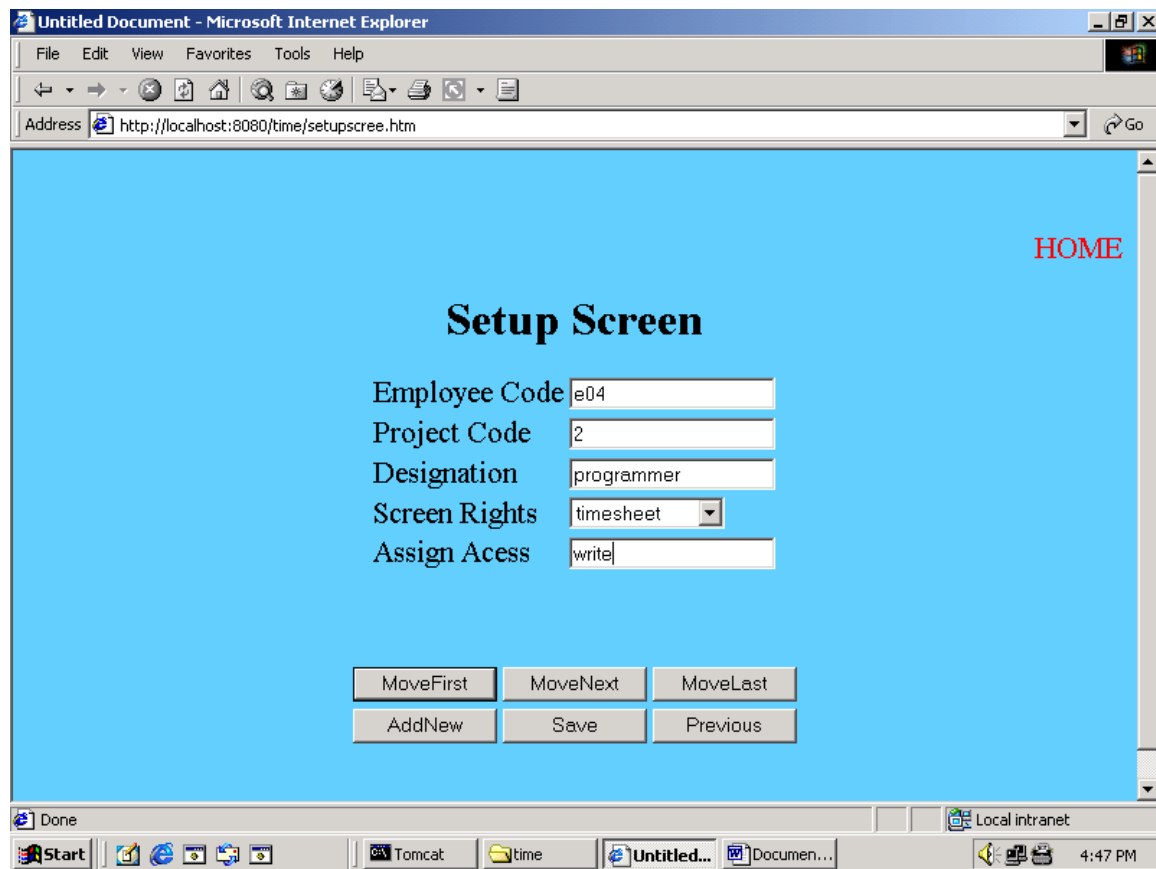
Employee Code	e04	Project Code	2
Proposed Start Date	11-dec-2003	Proposed Hours	100
EmployeeName	rajesh	Actual Hours	75
Actual Start Date	11-dec-2003	ProjectName	TimeTracking
Actual End Date	22-feb-2004		
TargetDate	22-feb-2004		

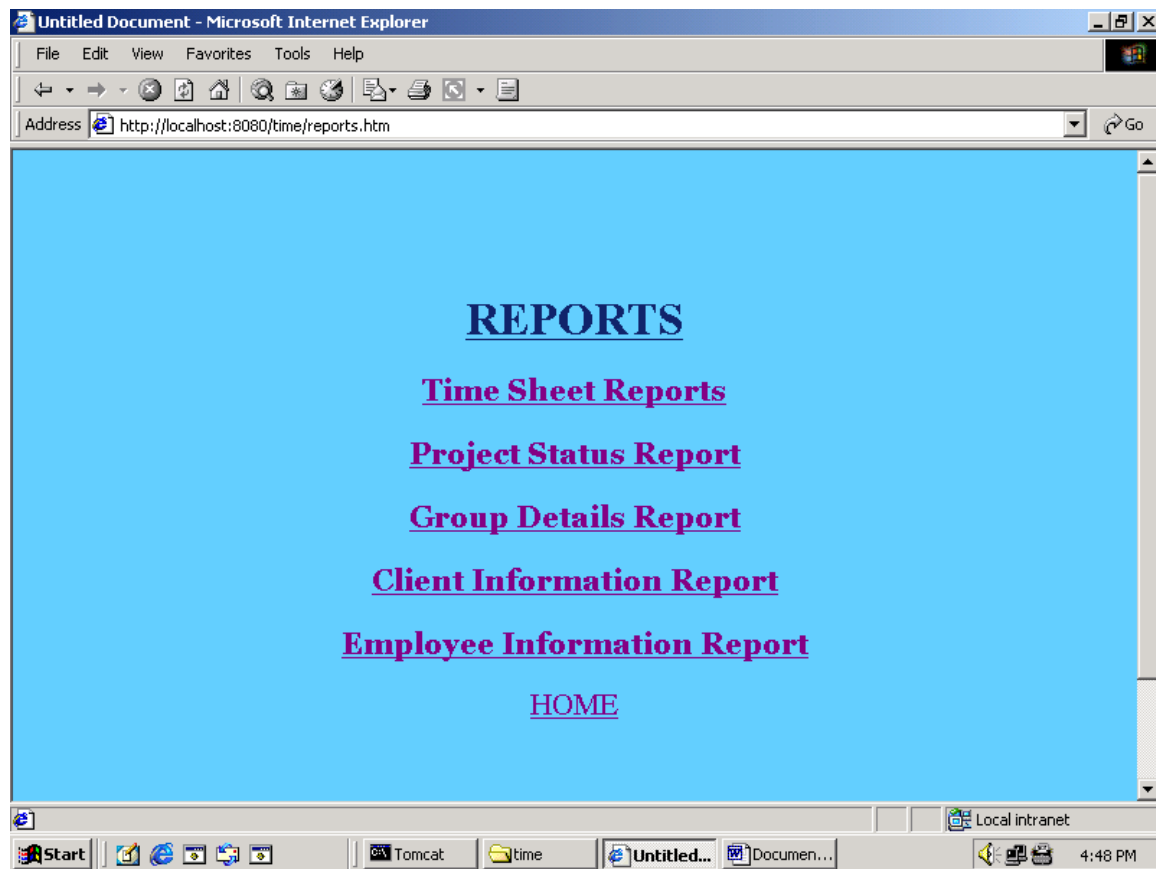
MoveFirst MoveNext MoveLast

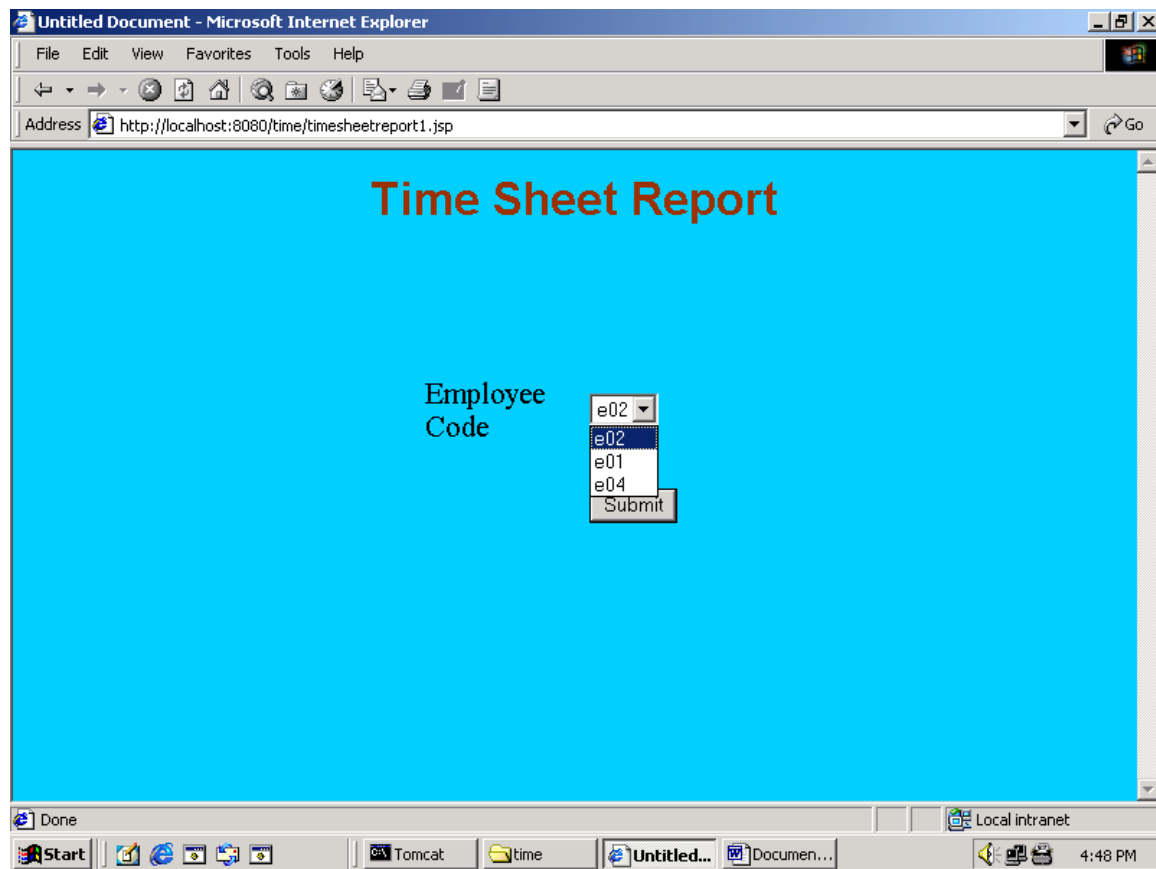
AddNew Save Previous

Done Local intranet

Start Tomcat time Untitled... Documen... 4:46 PM







Untitled Document - Microsoft Internet Explorer

File Edit View Favorites Tools Help

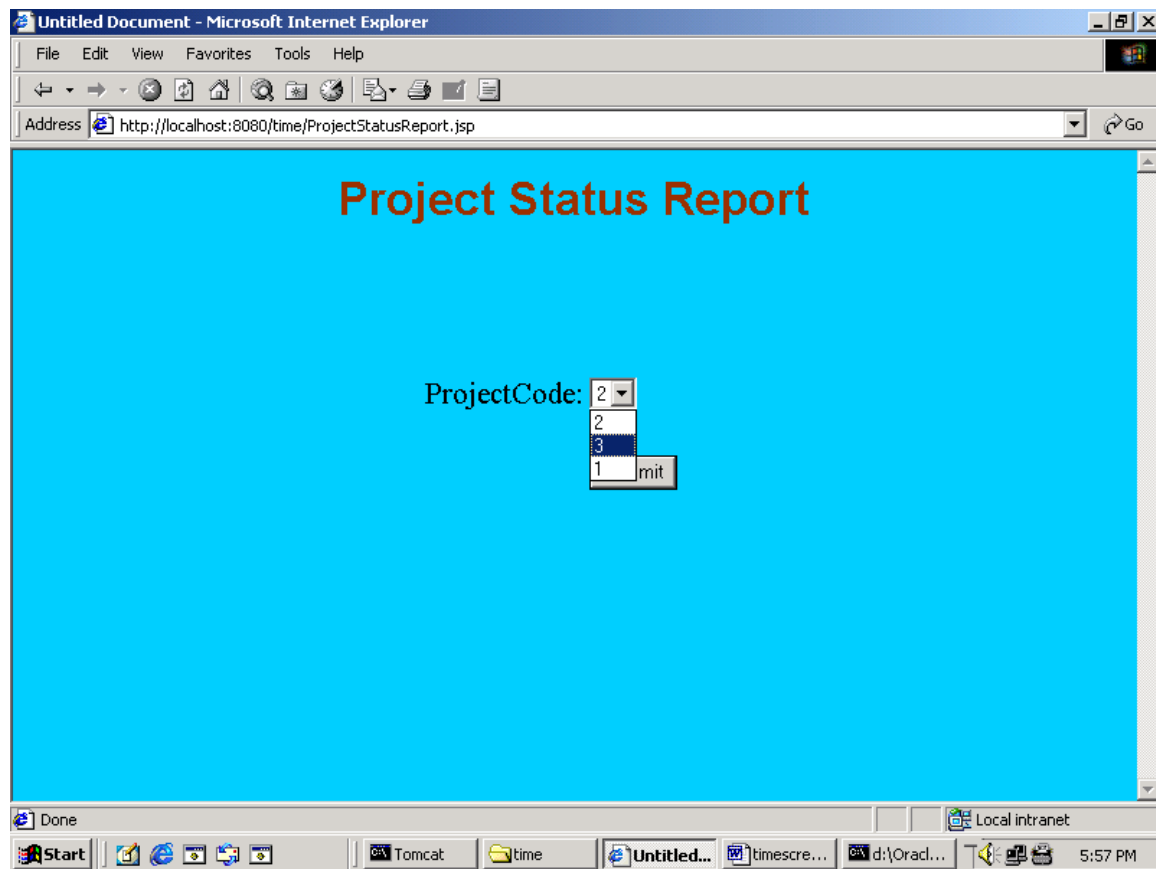
Address <http://localhost:8080/time/TimeSheetReport.jsp> Go

TimeSheet Report

ProjName	PStartDate	TargetDate	AStartDate	AEndDate	PHours	AHours
timetracking	2004-05-02 00:00:00	2004-06-02 00:00:00	2004-05-02 00:00:00	2004-06-02 00:00:00	400	380

Done Local intranet

Start Tomcat time Untitled... timescre... d:\Orac... 5:57 PM



Untitled Document - Microsoft Internet Explorer

File Edit View Favorites Tools Help

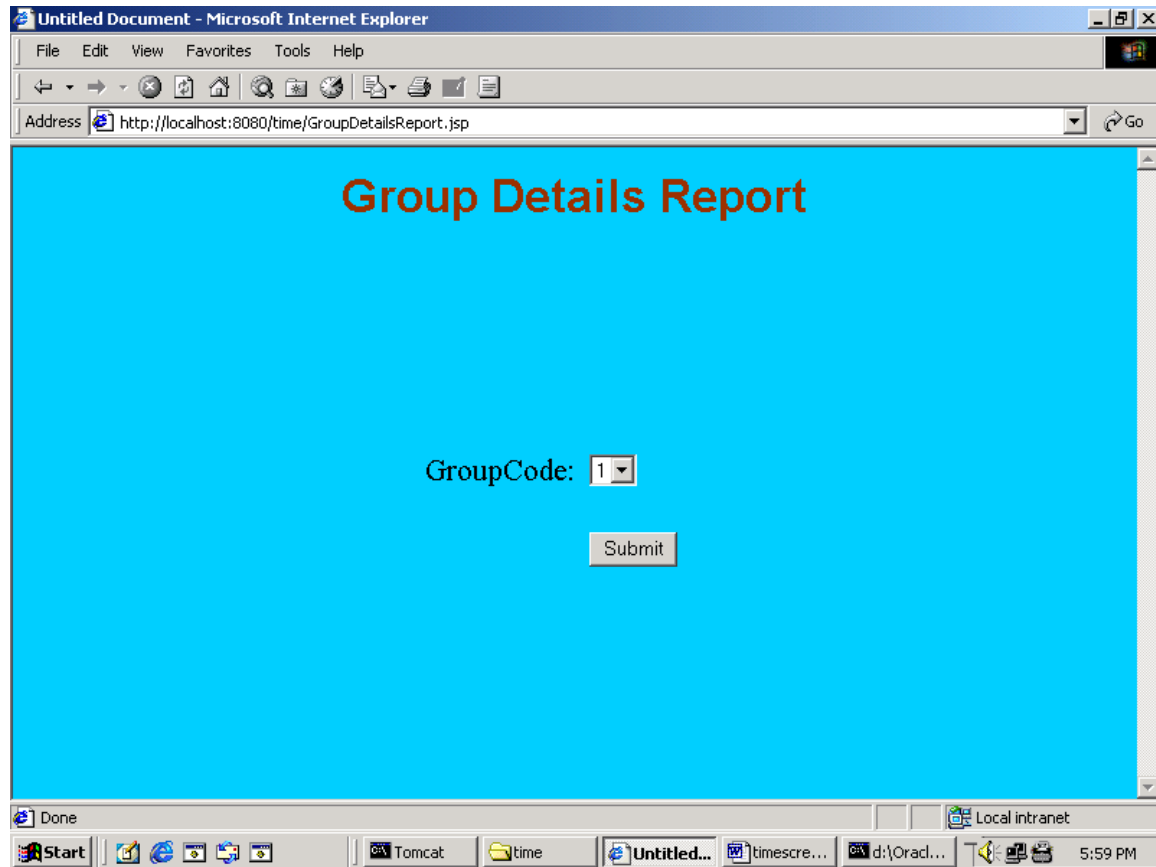
Address <http://localhost:8080/time/getPReport.jsp> Go

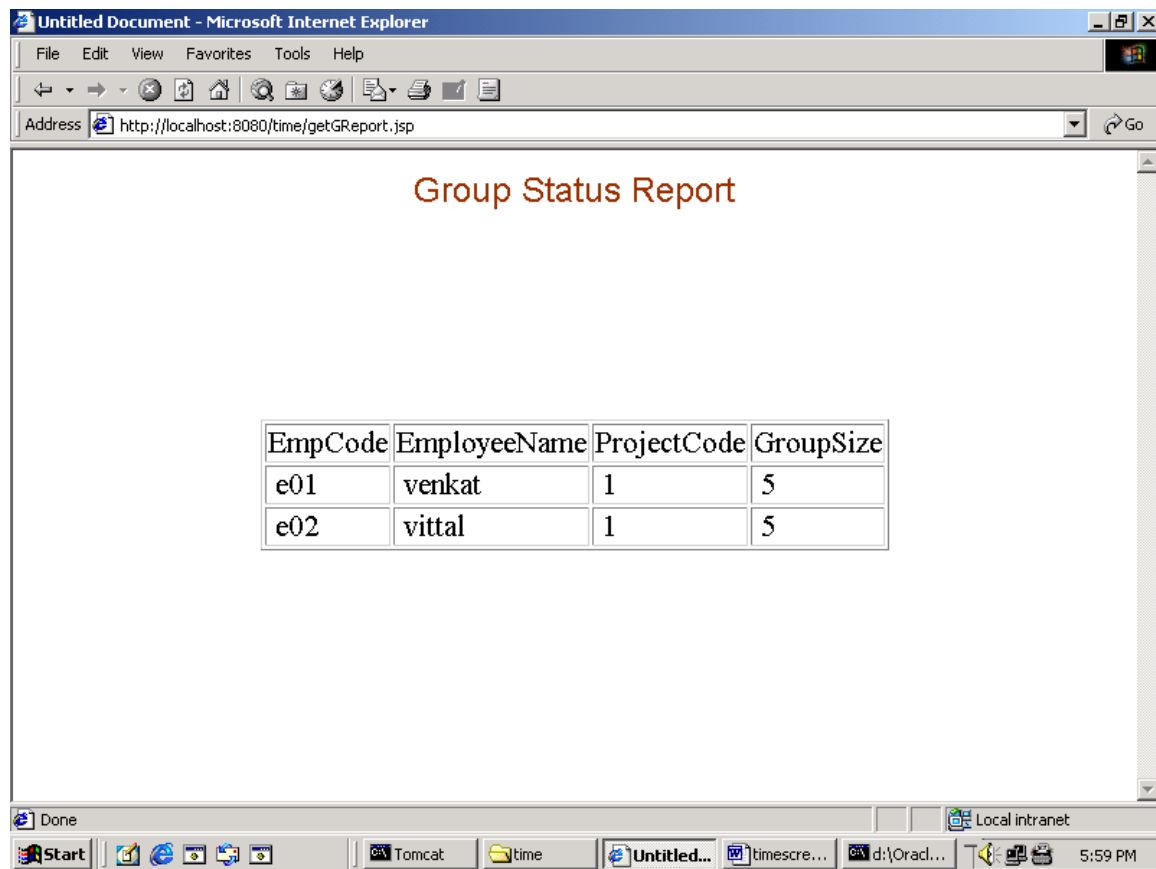
Project Status Report

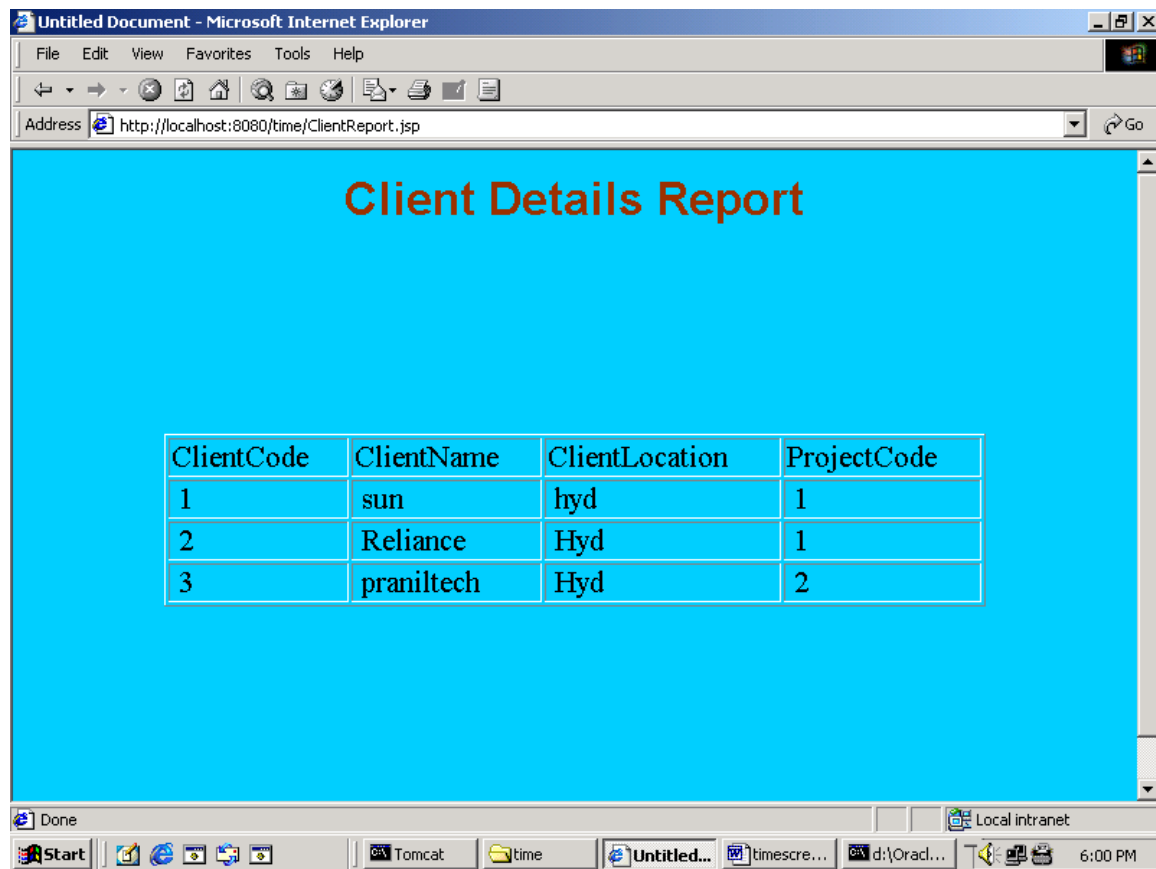
EmpCode	PManager	Activity	PStartDate	PEndDate	AStartDate	AEndDate	StatusDate
e04	vittal	programmer	2004-05-02 00:00:00	2004-06-02 00:00:00	2004-05-02 00:00:00	2004-06-02 00:00:00	2004-01-02 00:00:00

Done Local intranet

Start Tomcat time Untitled... timescre... d:\Orad... 5:58 PM







Untitled Document - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://localhost:8080/time/EmployeeReport.jsp> Go

Employee Report

EmpCode	EmpName	ProjectCode	Activity	GroupCode
e01	venkat	1	programmer	1
e02	vittal	1	designer	1
e04	rajesh	2	programmer	2

Done Local intranet

Start Tomcat time Untit... Docu... d:\Or... 5:01 PM

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.

We have tested the entire individual! Programs first, after having

Successful results in the individual! Program testing we moved further for the integration. We have combined some programs and then tested it, after having good results; we have combined all the programs together and started for system testing.

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 sestet and its original objective, current specifications, and system documentation. Analysts try to fm 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 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 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

Implementation literally means to put into effect or to carry out. 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 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 did 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 this 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