

Lending

*Tree*_(BESTOWSTEER)

Chapter 1

Abstract

Synopsis:

Most of the bank out-sources pre-loan process to loan agencies to reduce the burden and let the agencies pickup the information from customers and verify it before it is being forwarded to the actual bank for approval of loan.

Lending Tree is an interface which facilitates a customer to apply for a loan from on-line and to track the status from time-to-time along with aiding the loan approval agency to verify and accept/reject the customer file. Lending Tree is unique in such a way, it not only helps the customers but also the loan agency to check the pending, assign it to a departments, complete the formalities and procedures between the departments and arrive at decisions to very fact in addition to providing a transparency system for every one.

The customer can directly apply for a loan by selecting a bank and loan type from the list available. The application is received by loan agency who will have three departments- Pickup, Verifiaction and Legal. This system can be controlled by the administrator. First he will look at the application received and allot the application for a particular employee of pickup department. The employee will go and make a physical verification of the documents at the customers and receives the documents necessary for the loan. Then he logs into this system and forwards the application to the verification department which will verify the whereabouts of the person, his organization, his salary particulars etc. and then forwards the application with a status verified. Then application reaches the legal department. The legal department people will verify the builder details and when satisfied sends their report to the administrator.

The administrator or final approving authority views both types of reports, Viz, the reports from verification department and the report legal department. This will help him to take a decision regarding whether to forward it to the bank or not. The same is communicated to the customer.

The customer can at any time view the status of his application and can send any messages to the administrator and can get clarifications from him. Thus the lending tree s/w helps to simplify the loan system along with making the work easy.

There are 5 types of users who can get immense benefits from system:

- The customer - seeking the loan and information related to banks and loans
- The administrator of loan agency who will take track the decision of bank to approve or disapprove and also controls the overall system functionality
- The Pickup department users who picks up the details and documents from customers
- The verification department user who make a physical verification of the details submitted by the customer
- The legal department user who verifies the legality of the documents of the builder and construction.

Technologies Used: J2Se, Servlets, JSP, JDBC and Java Script.

Chapter 2

About The Organization

The **Techie Bits Private Limited** is a software product marketers firm, which associates its business process in developing tailor made products, and organizationally executing its sales, widely through a network of marketing executives. The marketing chaps find their way into the traditional business process environments, that exist in different firms, and specification ally acknowledge the clients for the necessity of the product what they are planning for the target business environment. The PC Tracker is one such product they are internationally targeting onto all such firms where there can be its implementation. Once the product base and its customer base are intact the system market the product with different customer associations. The system markets the product with different customer associations. The **Techie Bits Pvt Ltd** is a firm which has its applicational development products at the reach of the customers necessities with respect tot he demand of their implementatin. Out of many such products PC Tracker is one that is being targeted to hit the market in the coming days. The application that is being developed fully integrates itself with the standardized requirements that arise in any computerized environment where there is heavy usage of computers and the associated softwares. The generic maintenance upon the system becomes widely easier than the manual process.

Existing System:

Here the existing system is a manual one using which the banking agent can't maintain the effectively by sharing across different branches with proper security and can't track details easily. It doesn't provide proper co-ordination between different departments of the company. It doesn't allow the customer to check the status of his file in proper way which leads customer dis-satisfaction.

Doesn't provide faster and effective system

Doesn't provide good co-ordination between departments

Doesn't provide effective forwarding system to move the file from one level to another

Doesn't user-friendly interface

Difficulty in generating different reports as per the business requirement

Doesn't facilitate the services from online

Proposed System: The online automated system with web-based architecture can support issues like.

This system maintains the information related different departments and stored at a central DB, which leads easy accessibility and consistency

Interest rates of different banks and the other details are also available at the click of a mouse.

Customer can apply for a loan and track his file details from online.

The decision process is faster and more consistent

Provides good communication between two departments

Provides a facility to generate the reports very easily.

Chapter 3

FEASIBILITY REPORT: Feasibility study is the high level capsule version of the entire requirement analysis process. The objective of feasibility study is to determine whether the proposed system can be developed with available resources.

There are three steps to be followed for determining the feasibility study of proposed system.

Technical Feasibility

Operational Feasibility

Economical Feasibility

Technical Feasibility: It is concerned with hardware and software feasibility. In this study, one has to test whether the proposed system can be developed using existing technology or not. If new technology is required, what is the likelihood that it can be developed? The organization for which the system to be developed is not provided online services. Hence there is a requirement of new hardware and software technology for the deployment of proposed system. As per client requirements the system to be developed should have speed response because of fast change info, programming productivity, reliability, security, scalability, integration and availability. To meet these requirements I as a developer found jsp1.1 as a right choice because of its features platform independence, modularity and reusability.

Operational Feasibility: Operational feasibility determines whether the proposed system satisfied the user objectives and can be fitted in to current system operation. The proposed system “Lending Tree” can be justified as operationally feasible basing on the following.

The methods of processing and presentation are completely acceptable by the clients because they meet all the user and client requirements.

The clients have been involved during the preparation of requirement analysis and design process.

The system will certainly satisfy the user objectives and it will also enhance their capability.

The system can be best fitted into current operation and requires little training to both administrator and dealer. With the help of this system customer to place order requires simple data entry through forms provided. The proposed system is completely user friendly.

Economical Feasibility: This includes an evaluation of all incremental costs and benefits expected if proposed system is implemented. costs-benefit analysis which is to be performed during economical feasibility delineates costs for project development and weighs them against benefits of system. In this the proposed system replaces the manual process of receiving orders which benefits the organization to get more orders and good response. So developing this system is economically feasible to organization.

Application design

Users:

The major functionality of this product is divided into two categories.

1. Administrative User Functions.
2. Pickup Dept User Functions.
3. Verification Dept User Functions.
4. Legal Dept User Functions
- 5. Customer Functions**

Administrative User Functions: Administrators can perform the following task.

- Create/Update/Delete New Banks Info
- View the list of banks
- Create/Update/Delete New Departments.
- View List of Departments
- Create/Update/Delete Employee Info
- View the list of Banks
- Manage loans info related to different banks

- View the list applications and assign it to pickup dept employees
- Sending messages to customers
- View the pending applications
- Update the status of the application
- Generate reports

PickUp Dept User Functions: This user can perform the following task

- View the list of applications assigned to him
- Store the list of documents info that the customer has submitted
- Forward the application to verification

Verification Dept User Functions: This user can perform the following task

- View the list applications forwarded by different employees of Pickup Dept
- Verify the details
- Forward applications to Administrator

Legal Dept User Functions: This user can perform the following task

- Register Builder Info
- Store the documents info
- Generate APF no for the builder

Customer Functions: This user can perform the following task

- View the list of banks and their interest rates info
- Apply for loan in a bank
- Check the loan application status
- Messages

Number of Modules:

The system after careful analysis has been identified to present itself with the following modules:

1.Administrator Module: This module is responsible for coordinating the other modules. It allows the administrator create, update/delete and view the banks information and it allows admin to create create/update/delete and view different departments and it can create logins for different employees in each and every department and it can manage loan interest rates of different banks etc. It facilitates to view the new applicant details and assign it to different employees in pickup department initially. It allows the administrator to view customer application finally from verification department and builder details from legal department and then store bank final opinion in this application. It also allows to generate different reports for business analysis. It provides messaging facility for the administrator for communication.

2.PickUp Module: This module allows the pickup department to view their applications which are assigned to them, collect the documents according to the checklist and forward it to verification department.

3.Verification Module: This module allows the employees of verification department employees to view the forwarded application from pickup department and check the details as per the documents and forward it to next level.

4.Legal Module: This module allows the employee of legal departments to check verify legal documents of the builder, verify the check list and then generate the APF no for the builder.

5.Customer Module: This module allows the customer to view the interest rates of the banks which we are dealing, apply for a loan, check the status of the loan at any point of time and communicate with the administrator if necessary. It allows messaging facility for communication.

Requirements at the Server-side:

Hard Ware requirements:

1. Intel Pentium processor family with a speed of 800 MHz. or above
2. 256 MB RAM
3. 40 GB above hard disk as per the requirements of the org

Hardware Requirements

Pentium IV processes architecture

1. **256 MB RAM.**
2. **40 GB Hard Disk Space.**
3. **Ethernet card.**

Software specification

- Windows XP as operating system
- Apache Tomcat-5.0 web server
- Oracle 10g as database server
- Jdk1.5.0 as Java Software

Chapter 4

Analysis

Report

SRS Document

Intended Audience And Reading Suggestions

The document is prepared keeping in view of the academic constructs of my Bachelors Degree / Masters Degree from university as partial fulfillment of my academic purpose the document procedure that has been followed by me, while the system was studied and developed. The general document was provided by the industry as a reference guide to understand my responsibilities in developing the system, with respect to the requirements that have been pinpointed to get the exact structure of the system as stated by the actual client.

The system as stated by my project leader the actual standards of the specification were desired by conducting a series of interviews and questionnaires. The collected information was organized to form the specification document and then was modeled to suite the standards of the system as intended.

Document Conventions:

The overall documents for this project use the recognized modeling standards at the software industries level.

- **ER-Modeling to concentrate on the relational states existing upon the system with respect to Cardinality.**
- **The Physical dispense, which state the overall data search for the relational key whereas a transactions is implemented on the wear entities.**
- **Unified modeling language concepts to give a generalized blue print for the overall system.**

Scope of The Development Project:

Database Tier: The concentration is applied by adopting the Oracle10g database. SQL is taken as the standard query language. The overall business

rules are designed by using the power of PL/SQL components like dynamic queries, stored procedures stored functions etc.

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

Data Base Connectivity Tier

The communication architecture is designed by concentrated on the standards of servlets and JSP. The database connectivity is established using the Java Database connectivity.

Role Of Oracle In Database

Oracle 10g is one of the many database services that plug into a client / server model. It works efficiently to manage resources, a database information, among the multiple clients requesting & sending.

Structured Query Language (SQL)

SQL is an inter-active language used to query the database and access data in database. SQL has the following features:

1. It is a unified language.
2. It is a common language for relational database
3. It is a non-procedural language.

Introduction To Oracle

ORACLE is a comprehensive operating environment that packs the power of a mainframe system into user microcomputer. It provides a set of functional programs that user can use as tools to build structures and perform tasks. Because application developed on oracle are completely portable to environment and then move it into a multi user platform. Users do not have to be an expert to appreciate ORACLE, but the better user understands the programmer, the more productivity and creativity you will use the tools it provides.

What is a Relational Database Management System

A relational database management system (RDBMS) can perform a wide array of tasks. It acts as a transparent interface between the physical storage and a logical presentation of data. It provides a set of more or less flexible and sophisticates tools for handling information. User can use this tool to:

- ✓ Define a database
- ✓ Query the database
- ✓ Add, edit and delete data
- ✓ Modify the structure of database

- ✓ Secure data from public access
- ✓ Communicate within the networks
- ✓ Export and Import data

Because it gives so much control over data, a relational DBMS can also serve as the foundation for products that generate application and extract data.

A Database Management system may be called fully relational if it supports:

1. Relational Databases and
2. A language that is at least as powerful as the relational algebra

Dr. E.F.CODDS'S RULES

E.F.Codd specified a set of rules that an RDBMS has to do following 12 rules

1. **Information rule:** All information in a relational database including table names, column names, and domain names is represented explicitly by values in tables. With this design speed and productivity will be improved.
2. **Guaranteed access rule:** using a combination of table name, a primary key value and a column name can access Every piece of data in a relational database. With this productivity is improved since there is no need to resort to using physical pointers of address and provides data independence.
3. **Systematic treatment of null values:** The RDBMS distinguishes between blanks nulls in records and handles such values in consistent manner that produces correct answers on comparison and calculations. With these users can distinguish results of queries/operations that involve nulls, zeroes and blanks.
4. **Active on-line catalog based on the relational model:** The description of a database and its contents are database tables and therefore can be queried on-line via the database language. With this the DBA's productivity is improved since the changes are common in catalog to all tables for all queries and reports.

5. **Comprehensive data sub-language rule:** An RDBMS language supports definitions of tables/views. Query and update data, integrity constraints, authorizations, definitions of transactions. With this user productivity is improved since there is a single approach for all database operations.
6. **View updating rule:** Any view that is theoretically updateable can be updated using RDBMS. With this data consistency is ensured since changes in the underlying tables are transmitted to the view their support.
7. **High-level insert, update and delete:** The RDBMS supports insertion, updating deletion at al table level. With this the performance is improved by optimizing the path for execution.
8. **Physical data independence:** The execution of adhoc requests and application programs are not affected by the changes in the physical data access methods. With this DBA can make changes in application programs or adhoc requests.
9. **Logical data independence:** Logical changes in tables and views do not require changes in the application programs or in the format of adhoc requests. With this databases can change and grow without changes in applications.
10. **Integrity independence:** Since constraints are stored in system catalog, they can be changed without changing application program. With this maintenance costs are reduced since only system catalog need to be changed.
11. **Distribution independence:** Application programs and adhoc requests are not affected by changes in the distribution of physical data. With this system reliability will be improved since application programs will work even if the programs and dare are shifted.
12. **Non-Subversion rule:** If the RDBMS has languages that access a record of a time it cannot be used to bypass integrity. With this data integrity is achieved since the system catalog is musts
 - a. Code defined a relationally complete set of operations taking one or more relation as their operands. The operators are two groups: the traditional set of operators union, intersection, difference and

Cartesian product and the special operators selection, projection, join and division.

b. A fully relational DBMS has to support the above operations.

The Oracle Environment

ORACLE is modular system that consists of the ORACLE database and several functional programs. ORACLE tools had four kinds of works:

- ❖ Database management
- ❖ Data access and manipulations
- ❖ Programming
- ❖ Connectivity

Database Management Tools

This is usually (known as RDBMS by ORACLE) includes the core programs of Oracle's database management system, the ORACLE database with its associated tables and views, which are stored in the Oracle's data dictionary and a group of helpful activities. The data dictionary stores information related to every facet of database system. User names, user access rights, table storage information and auditing data for the disaster recovery are all stored in the data dictionary.

Data Access and Manipulations

All of Oracle's data access and manipulation tools are firmly based on ANSI standard SQL. In Oracle, the tools that a user will use to access and manipulate data, has well as to design or use applications. Each provides separate point of entry and unique speech to the Oracle system.

SQL*PLUS allows direct access to database with SQL FORMS offer a user-friendly way to create and use forms. SQL*REPORT writer lets you to creates formatted output SQL*MENU provides a way for you to integrate your application menus.

Oracle Supports Applications Development

SQL*FORMS is an excellent user-friendly tool for quickly creating forms. User can start with extremely simple default forms or use the full screen painting function to create detailed screens for accessing and updating multiple tables and for

controlling and editing the data as is entered. In SQL FORMS, ORACLE provides unique control devices called TRIGGERS to influence user action on a field before, during and after data input. Those triggers can execute SQL commands, native SQL FORMS commands, or external procedural language sub routines from within a form.

These forms are an advanced fourth-generation tool that will adapt to your requirements extremely well. With SQL*MENU you can link all of the forms. Programs and queries are easily maintained with the secured menu structures. Oracle Uses The SQL Command Set

With SQL * Menu you can link all forms, programs and queries in easily maintained secure structures.

Oracle provides a SQL command set that is close to the ANSI standard ORACLE has added extensive report-formatting commands to extend the direct SQL language our capabilities and to delay the needs for alternative report formatting techniques. Statistical, Arithmetic string date/time functions are also included.

ORACLE has with limitation. The Single-user does not include all of the tools available on the hardware platforms. Also. ORACLE is relatively expensive and more complex than more single user, PC-basis database managers. ORACLE delivers a comprehensive package that allows for unlimited growth.

Oracle Gives You Security and Control ORACLE has several features that ensure the integrity of user database. If an interruption occurs in processing, a rollback can reset the database to a point before the disaster. If a restore is necessary, ORACLE has a roll forward command for creating a database to its most recent save point. Oracle provides users with several functions for serving data. Grant and Revoke commands limited access to information down to the row and column levels. Views are valuable future for limiting access to the primary tables in the database.

Oracle Performs Competitively

ORACLE has been constantly improved to perform competitively on the largest database because RDBMS has been hampered by a reputation for slow access

time. ORACLE had to prove itself continuously and so the unique feature of clustering techniques for storing data on the disk or another performance gained. Additional functions help control complex database installations. The active data dictionary, which automatically update and logs modification to the database provide documentation data off loading form the modification process. Finally, ORACLE stores the DBMS kernel in extended memory, so more main memory is available for the application.

Programming Tools

One of the most important categories of tools available from oracle is its series of programming interface. This precompiled software provides a convenient and easy to use method of incorporating ORACLE SQL statement in high level programming language. Current oracle can interface with COBOL and 'C'.

Connective Tools

Connectivity tools make oracle available to network and to other database managers. SQL * Star is group of products including SQL *NET and SQL * connect that allows user to use stores on remote machines. A micro based version of oracle called network station oracle, user may SQL * NET, instead of oracle RDBMS. SQL * Connection does the translating interactively without any modification or translations.

Fourth Generation Languages

The fourth generation languages were created to overcome the problem of third generation languages, and these 4GL's are generally referred to as high productivity languages.

Objectives Of Fourth Generation Languages

To speed up the application building process

- ☐ To make the application building process
- ☐ To minimize the debugging problems
- ☐ To generate bug free code from high level of expressions of requirements.
- ☐ To make languages easy to use and understand

All these make the end users solve their own problems and pit computers to work.

Characteristics of Fourth Generation Languages

- ✓ Simple Query facilities/ language
- ✓ Complex query and updating language
- ✓ Report generators
- ✓ Graphic languages
- ✓ Decision support languages
- ✓ Application generations
- ✓ Specification language
- ✓ Very high level language
- ✓ Parameterized application language
- ✓ Application language

Properties of Fourth Generation Languages

- ◆ Easy to use
- ◆ Employees a database management system directly
- ◆ Requires significantly fewer instructions than third generation language
- ◆ Intelligent default assumptions make abort what the use wants possible
- ◆ Easy to understand and maintain
- ◆ Enforces and encourages structured code
- ◆ Subset can learnt not by non-technical users in a short period



Client Server Technologies

Client Server

Over view:

With the varied topic in existence in the fields of computers, Client Server is one, which has generated more heat than light, and also more hype than reality. This technology has acquired a certain critical mass attention with its dedication conferences and magazines. Major computer vendors such as IBM and DEC, have declared that Client Servers is their main future market. A survey of DBMS magazine revealed that 76% of its readers were actively looking at the client server solution. The growth in the client server development tools from \$200 million in 1992 to more than \$1.2 billion in 1996.

Client server implementations are complex but the underlying concept is simple and powerful. A client is an application running with local resources but able to request the database and relate the services from separate remote server. The software mediating this client server interaction is often referred to as MIDDLEWARE.

The typical client either a PC or a Work Station connected through a network to a more powerful PC, Workstation, Midrange or Main Frames server usually capable of handling request from more than one client. However, with some configuration server may also act as client. A server may need to access other server in order to process the original client request.

The key client server idea is that client as user is essentially insulated from the physical location and formats of the data needs for their application. With the proper middleware, a client input from or report can transparently access and manipulate both local database on the client machine and remote databases on one or more servers. An added bonus is the client server opens the door to multi-vendor database access indulging heterogeneous table joins.

What is a Client Server

Two prominent systems in existence are client server and file server systems. It is essential to distinguish between client servers and file server systems. Both provide shared network access to data but the comparison ends there! The file server simply provides a remote disk drive that can be accessed by LAN applications on a file by file basis. The client server offers full relational database

services such as SQL-Access, Record modifying, Insert, Delete with full relational integrity backup/ restore performance for high volume of transactions, etc. the client server middleware provides a flexible interface between client and server, who does what, when and to whom.

Why Client Server

Client server has evolved to solve a problem that has been around since the earliest days of computing: how best to distribute your computing, data generation and data storage resources in order to obtain efficient, cost effective departmental and enterprise wide data processing. During mainframe era choices were quite limited. A central machine housed both the CPU and DATA (cards, tapes, drums and later disks). Access to these resources was initially confined to batched runs that produced departmental reports at the appropriate intervals. A strong central information service department ruled the corporation. The role of the rest of the corporation limited to requesting new or more frequent reports and to provide hand written forms from which the central data banks were created and updated. The earliest client server solutions therefore could best be characterized as "SLAVE-MASTER".

Time-sharing changed the picture. Remote terminal could view and even change the central data, subject to access permissions. And, as the central data banks evolved in to sophisticated relational database with non-programmer query languages, online users could formulate adhoc queries and produce local reports without adding to the MIS applications software backlog. However remote access was through dumb terminals, and the client server remained subordinate to the Slave\Master.

Front end or User Interface Design

The entire user interface is planned to be developed in browser specific environment with a touch of Intranet-Based Architecture for achieving the Distributed Concept.

The browser specific components are designed by using the HTML standards, and the dynamism of the designed by concentrating on the constructs of the Java Server Pages.

Communication or Database Connectivity Tier

The Communication architecture is designed by concentrating on the Standards of Servlets and Enterprise Java Beans. The database connectivity is established by using the Java Data Base Connectivity.

The standards of three-tire architecture are given major concentration to keep the standards of higher cohesion and limited coupling for effectiveness of the operations.

Features of The Language Used

In my project, I have chosen *Java* language for developing the code.

About Java

Initially the language was called as "oak" but it was renamed as "Java" in 1995. The primary motivation of this language was the need for a platform-independent (i.e., architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- ☐ Java is a programmer's language.
- ☐ Java is cohesive and consistent.

- Except for those constraints imposed by the Internet environment, Java gives the programmer, full control.

Finally, Java is to Internet programming where C was to system programming.

Importance of Java to the Internet

Java has had a profound effect on the Internet. This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the Server and the Personal computer. They are: Passive information and Dynamic active programs. The Dynamic, Self-executing programs cause serious problems in the areas of Security and probability. But, Java addresses those concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

Java can be used to create two types of programs

Applications and Applets: An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++. Java's ability to create Applets makes it important. An Applet is an application designed to be transmitted over the Internet and executed by a Java –compatible web browser. An applet is actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can react to the user input and dynamically change.

Features Of Java

Security

Every time you that you download a "normal" program, you are risking a viral infection. Prior to Java, most users did not download executable programs frequently, and those who did scanned them for viruses prior to execution. Most users still worried about the possibility of infecting their systems with a virus. In addition, another type of malicious program exists that must be guarded against. This type of program can gather private information, such as credit card numbers, bank account balances, and passwords. Java answers both these concerns by providing a "firewall" between a network application and your computer.

When you use a Java-compatible Web browser, you can safely download Java applets without fear of virus infection or malicious intent.

Portability

For programs to be dynamically downloaded to all the various types of platforms connected to the Internet, some means of generating portable executable code is needed. As you will see, the same mechanism that helps ensure security also helps create portability. Indeed, Java's solution to these two problems is both elegant and efficient.

The Byte code

The key that allows the Java to solve the security and portability problems is that the output of Java compiler is Byte code. Byte code is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for byte code.

Translating a Java program into byte code helps makes it much easier to run a program in a wide variety of environments. The reason is, once the run-time package exists for a given system, any Java program can run on it.

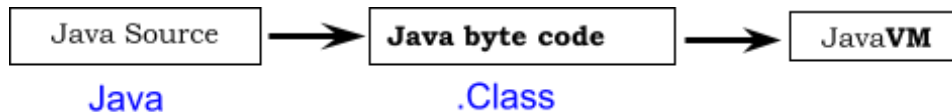
Although Java was designed for interpretation, there is technically nothing about Java that prevents on-the-fly compilation of byte code into native code. Sun has just completed its Just In Time (JIT) compiler for byte code. When the JIT compiler is a part of JVM, it compiles byte code into executable code in real time, on a piece-by-piece, demand basis. It is not possible to compile an entire Java program into executable code all at once, because Java performs various run-time checks that can be done only at run time. The JIT compiles code, as it is needed, during execution.

Java, Virtual Machine (JVM)

Beyond the language, there is the Java virtual machine. The Java virtual machine is an important element of the Java technology. The virtual machine can be embedded within a web browser or an operating system. Once a piece of Java code is loaded onto a machine, it is verified. As part of the loading process, a

class loader is invoked and does byte code verification makes sure that the code that's has been generated by the compiler will not corrupt the machine that it's loaded on. Byte code verification takes place at the end of the compilation process to make sure that is all accurate and correct. So byte code verification is integral to the compiling and executing of Java code.

Overall **Description**



Picture showing the development process of JAVA Program

Java programming uses to produce byte codes and executes them. The first box indicates that the Java source code is located in a .Java file that is processed with a Java compiler called javac. The Java compiler produces a file called a .class file, which contains the byte code. The .Class file is then loaded across the network or loaded locally on your machine into the execution environment is the Java virtual machine, which interprets and executes the byte code.

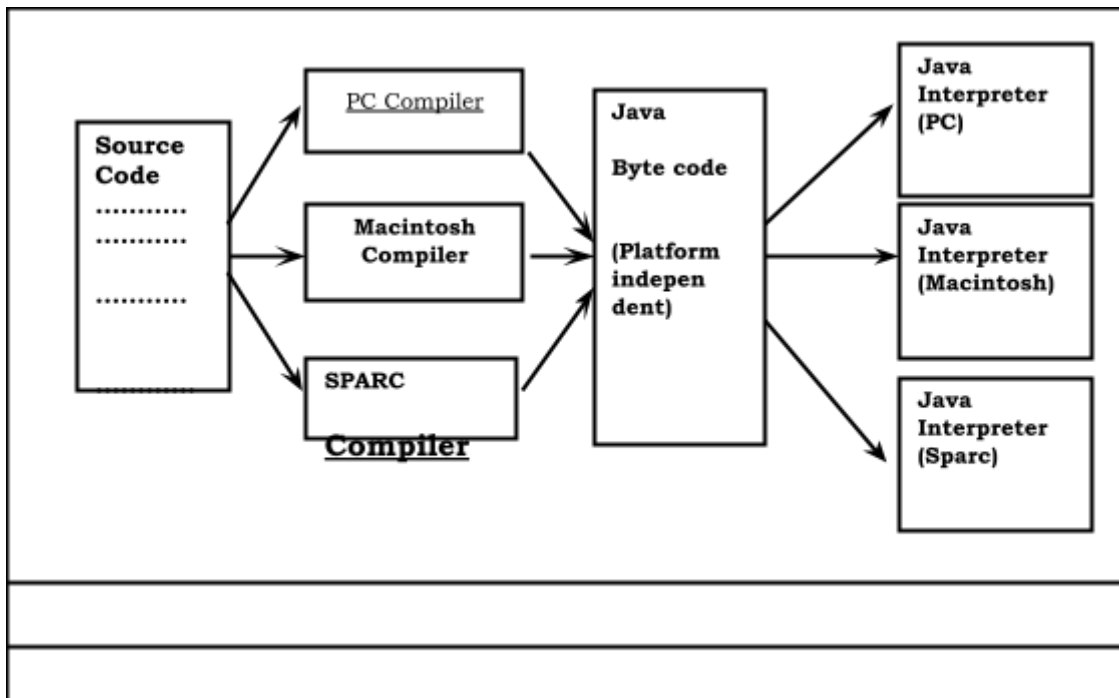
Java Architecture

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

Compilation of code

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed to execute the byte code. The JVM is created for overcoming the issue of portability. The code is written and compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

Compiling and interpreting Java Source Code



During run-time the Java interpreter tricks the bytecode file into thinking that it is running on a Java Virtual Machine. In reality this could be a Intel Pentium Windows 95 or SunSARC station running Solaris or Apple Macintosh running system and all could receive code from any computer through Internet and run the Applets.

Simple

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ programmer, learning Java will be even easier. Because Java inherits the C/C++ syntax and many of the object oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

Object-Oriented

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank slate. One outcome of this was a clean usable, pragmatic approach to objects. The object

model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

Robust

The multi-platform environment of the Web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and run time.

Java virtually eliminates the problems of memory management and deallocation, which is completely automatic. In a well-written Java program, all run time errors can –and should –be managed by your program.

*Servlets, JSP, JDBC & HTML
Technologies*

SERVLETS

Introduction

The Java web server is JavaSoft's own web Server. The Java web server is just a part of a larger framework, intended to provide you not just with a web server, but also with tools. To build customized network servers for any Internet or Intranet client/server system. Servlets are to a web server, how applets are to the browser.

About Servlets

Servlets provide a Java-based solution used to address the problems currently associated with doing server-side programming, including inextensible scripting solutions, platform-specific APIs, and incomplete interfaces.

Servlets are objects that conform to a specific interface that can be plugged into a Java-based server. Servlets are to the server-side what applets are to the client-side - object byte codes that can be dynamically loaded off the net. They differ from applets in that they are faceless objects (without graphics or a GUI component). They serve as platform independent, dynamically loadable, plugable helper byte code objects on the server side that can be used to dynamically extend server-side functionality.

For example, an HTTP Servlets can be used to generate dynamic HTML content. When you use Servlets to do dynamic content you get the following advantages:

- ☐ They're faster and cleaner than CGI scripts
- ☐ They use a standard API (the Servlets API)
- ☐ They provide all the advantages of Java (run on a variety of servers without needing to be rewritten).

Attractiveness of Servlets

There are many features of Servlets that make them easy and attractive to use.

These include:

- ☐ Easily configured using the GUI-based Admin tool
- ☐ Can be loaded and invoked from a local disk or remotely across the network.
- ☐ Can be linked together, or chained, so that one Servlets can call another Servlets, or several Servlets in sequence.

- ☐ Can be called dynamically from within HTML pages, using server-side include tags.
- ☐ Are secure - even when downloading across the network, the Servlets security model and Servlets sandbox protect your system from unfriendly behavior.

Advantages of the Servlet API

One of the great advantages of the Servlet API is protocol independence. It assumes nothing about:

- ☐ The protocol being used to transmit on the net
- ☐ How it is loaded
- ☐ The server environment it will be running in

These qualities are important, because it allows the Servlet API to be embedded in many different kinds of servers. There are other advantages to the Servlet API as well. These include:

- ☐ It's extensible - you can inherit all your functionality from the base classes made available to you.
- ☐ it's simple, small, and easy to use.

Features of Servlets

- ☐ Servlets are persistent. Servlet are loaded only by the web server and can maintain services between requests.
- ☐ Servlets are fast. Since Servlets only need to be loaded once, they offer much better performance over their CGI counterparts.
- ☐ Servlets are platform independent.
- ☐ Servlets are extensible. Java is a robust, object-oriented programming language, which easily can be extended to suit your needs
- ☐ Servlets are secure.
- ☐ Servlets can be used with a variety of clients.

Loading Servlets

Servlets can be loaded from three places

From a directory that is on the CLASSPATH. The CLASSPATH of the JavaWebServer includes service root/classes/ which is where the system classes reside.

From the <SERVICE_ROOT /Servlets/ directory. This is **not** in the server's classpath. A class loader is used to create Servlets from this directory. New Servlets can be added - existing Servlets can be recompiled and the server will notice these changes.

From a remote location. For this a code base like http: // nine.eng / classes / foo / is required in addition to the Servlets class name. Refer to the admin GUI docs on Servlet section to see how to set this up.

Loading Remote Servlets

Remote Servlets can be loaded by:

1. Configuring the [Admin Tool](#) to setup automatic loading of remote Servlets
2. Setting up [server side include](#) tags in .shtml files
3. Defining a [filter chain](#) configuration

Invoking Servlets

A Servlet invoker is a Servlet that invokes the "service" method on a named Servlet. If the Servlet is not loaded in the server, then the invoker first loads the Servlet (either from local disk or from the network) and then invokes the "service" method. Also like applets, local Servlets in the server can be identified by just the class name. In other words, if a Servlet name is not absolute, it is treated as local.

A client can invoke Servlets in the following ways:

- ☐ The client can ask for a document that is served by the Servlet.
- ☐ The client (browser) can invoke the Servlet directly using a URL, once it has been mapped using the Servlet Aliases section of the admin GUI.
- ☐ The Servlet can be invoked through server side include tags.
- ☐ The Servlet can be invoked by placing it in the Servlets/ directory.
- ☐ The Servlet can be invoked by using it in a filter chain.

JAVASCRIPT

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then updates the browser's display accordingly

Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client side programming since most of the browsers supports it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags

```
<SCRIPTS>..  
</SCRIPT>.
```

```
<SCRIPT LANGUAGE = "JavaScript">
```

```
JavaScript statements
```

```
</SCRIPT>
```

Here are a few things we can do with JavaScript :

- ☐ Validate the contents of a form and make calculations.
- ☐ Add scrolling or changing messages to the Browser's status line.
- ☐ Animate images or rotate images that change when we move the mouse over them.
- ☐ Detect the browser in use and display different content for different browsers.
- ☐ Detect installed plug-ins and notify the user if a plug-in is required.

We can do much more with JavaScript, including creating entire application.

JavaScript Vs Java

JavaScript and Java are entirely different languages. A few of the most glaring differences are:

- ☐ Java applets are generally displayed in a box within the web document; JavaScript can affect any part of the Web document itself.
- ☐ While JavaScript is best suited to simple applications and adding interactive features to Web pages; Java can be used for incredibly complex applications.

There are many other differences but the important thing to remember is that JavaScript and Java are separate languages. They are both useful for different things; in fact they can be used together to combine their advantages.

ADVANTAGES

- ☐ JavaScript can be used for Sever-side and Client-side scripting.
- ☐ It is more flexible than VBScript.
- ☐ JavaScript is the default scripting languages at Client-side since all the browsers supports it.

Hyper Text Markup Language

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or

emphasized works that load to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.

HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Basic HTML Tags :

<!-- -->	Specifies comments
<A>.....	Creates hypertext links
.....	Formats text as bold
<BIG>.....</BIG>	Formats text in large font.
<BODY>...</BODY>	Contains all tags and text in the HTML document
<CENTER>...</CENTER>	Creates text
<DD>...</DD>	Definition of a term
<DL>...</DL>	Creates definition list
...	Formats text with a particular font
<FORM>...</FORM>	Encloses a fill-out form
<FRAME>...</FRAME>	Defines a particular frame in a set of frames
<H#>...</H#>	Creates headings of different levels
<HEAD>...</HEAD>	Contains tags that specify information about a document
<HR>...</HR>	Creates a horizontal rule
<HTML>...</HTML>	Contains all other HTML tags
<META>...</META>	Provides meta-information about a document
<SCRIPT>...</SCRIPT>	Contains client-side or server-side script
<TABLE>...</TABLE>	Creates a table
<TD>...</TD>	Indicates table data in a table
<TR>...</TR>	Designates a table row
<TH>...</TH>	Creates a heading in a table

ADVANTAGES

- ☐ A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
- ☐ HTML is platform independent.
- ☐ HTML tags are not case-sensitive.

Java Database Connectivity

What Is JDBC?

JDBC is a Java API for executing SQL statements. (As a point of interest, JDBC is a trademarked name and is not an acronym; nevertheless, JDBC is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API.

Using JDBC, it is easy to send SQL statements to virtually any relational database. One can write a single program using the JDBC API, and the program will be able to send SQL statements to the appropriate database. The combinations of Java and JDBC lets a programmer write it once and run it anywhere.

What Does JDBC Do?

Simply put, JDBC makes it possible to do three things:

- ☐ Establish a connection with a database
- ☐ Send SQL statements
- ☐ Process the results.

JDBC versus ODBC and other APIs

At this point, Microsoft's ODBC (Open Database Connectivity) API is that probably the most widely used programming interface for accessing relational databases. It offers the ability to connect to almost all databases on almost all platforms.

So why not just use ODBC from Java? The answer is that you can use ODBC from Java, but this is best done with the help of JDBC in the form of the JDBC-ODBC Bridge, which we will cover shortly. The question now becomes "Why do you need JDBC?" There are several answers to this question:

1. ODBC is not appropriate for direct use from Java because it uses a C interface. Calls from Java to native C code have a number of drawbacks in the security, implementation, robustness, and automatic portability of applications.
2. A literal translation of the ODBC C API into a Java API would not be desirable. For example, Java has no pointers, and ODBC makes copious use of them, including the notoriously error-prone generic pointer "void *". You can think of JDBC as ODBC translated into an object-oriented interface that is natural for Java programmers.
3. ODBC is hard to learn. It mixes simple and advanced features together, and it has complex options even for simple queries. JDBC, on the other hand, was designed to keep simple things simple while allowing more advanced capabilities where required.
4. A Java API like JDBC is needed in order to enable a "pure Java" solution. When ODBC is used, the ODBC driver manager and drivers must be manually installed on every client machine. When the JDBC driver is written completely in Java, however, JDBC code is automatically installable, portable, and secure on all Java platforms from network computers to mainframes.

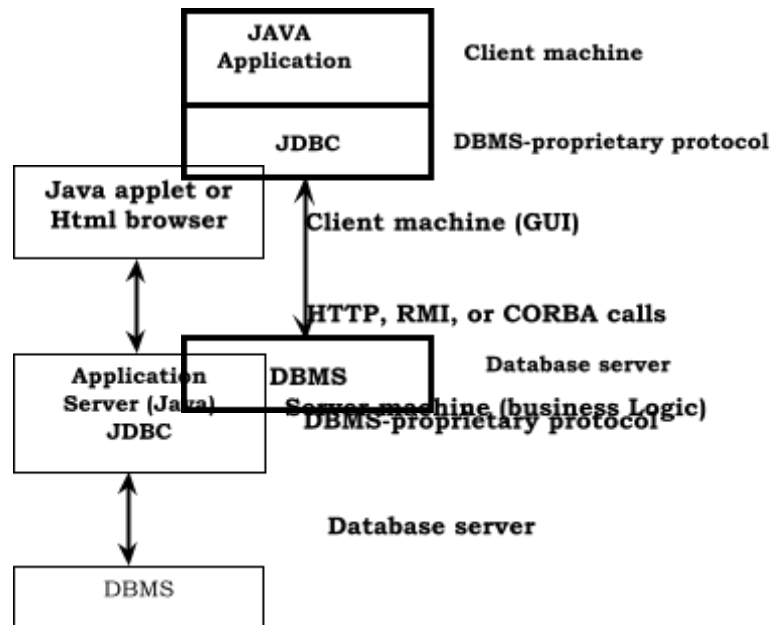
Two-tier and Three-tier Models

The JDBC API supports both two-tier and three-tier models for database access.

In the two-tier model, a Java applet or application talks directly to the database. This requires a JDBC driver that can communicate with the particular database management system being accessed. A user's SQL statements are delivered to the database, and the results of those statements are sent back to the user. The database may be located on another machine to which the user is connected via a network. This is referred to as a client/server configuration, with the user's

machine as the client, and the machine housing the database as the server. The network can be an Intranet, which, for example, connects employees within a corporation, or it can be the Internet.

In the three-tier model, commands are sent to a "middle tier" of services, which



then send SQL statements to the database. The database processes the SQL statements and sends the results back to the middle tier, which then sends them to the user. MIS directors find the three-tier model very attractive because the middle tier makes it possible to maintain control over access and the kinds of updates that can be made to corporate data. Another advantage is that when there is a middle tier, the user can employ an easy-to-use higher-level API which is translated by the middle tier into the appropriate low-level calls. Finally, in many cases the three-tier architecture can provide performance advantages.

Until now the middle tier has typically been written in languages such as C or C++, which offer fast performance. However, with the introduction of optimizing compilers that translate Java byte code into efficient machine-specific code, it is becoming practical to implement the middle tier in Java. This is a big plus, making it possible to take advantage of Java's robustness, multithreading, and security features. JDBC is important to allow database access from a Java middle tier.

JDBC Driver Types

The JDBC drivers that we are aware of at this time fit into one of four categories:

- ☐ JDBC-ODBC bridge plus ODBC driver
- ☐ Native-API partly-Java driver
- ☐ JDBC-Net pure Java driver
- ☐ Native-protocol pure Java driver

JDBC-ODBC Bridge

If possible, use a Pure Java JDBC driver instead of the Bridge and an ODBC driver. This completely eliminates the client configuration required by ODBC. It also eliminates the potential that the Java VM could be corrupted by an error in the native code brought in by the Bridge (that is, the Bridge native library, the ODBC driver manager library, the ODBC driver library, and the database client library).

What Is the JDBC- ODBC Bridge?

The JDBC-ODBC Bridge is a JDBC driver, which implements JDBC operations by translating them into ODBC operations. To ODBC it appears as a normal application program. The Bridge implements JDBC for any database for which an ODBC driver is available. The Bridge is implemented as the

sun.jdbc.odbc Java package and contains a native library used to access ODBC. The Bridge is a joint development of Intersolv and JavaSoft.

Java Server Pages (JSP)

Java server Pages is a simple, yet powerful technology for creating and maintaining dynamic-content web pages. Based on the Java programming language, Java Server Pages offers proven portability, open standards, and a mature re-usable component model .The Java Server Pages architecture enables the separation of content generation from content presentation. This separation not eases maintenance headaches, it also allows web team members to focus on their areas of expertise. Now, web page designer can

concentrate on layout, and web application designers on programming, with minimal concern about impacting each other's work.

Features of JSP

Portability:

Java Server Pages files can be run on any web server or web-enabled application server that provides support for them. Dubbed the JSP engine, this support involves recognition, translation, and management of the Java Server Page lifecycle and its interaction components.

Components

It was mentioned earlier that the Java Server Pages architecture can include reusable Java components. The architecture also allows for the embedding of a scripting language directly into the Java Server Pages file. The components currently supported include Java Beans, and Servlets.

Processing

A Java Server Pages file is essentially an HTML document with JSP scripting or tags. The Java Server Pages file has a JSP extension to the server as a Java Server Pages file. Before the page is served, the Java Server Pages syntax is parsed and processed into a Servlet on the server side. The Servlet that is generated outputs real content in straight HTML for responding to the client.

Access Models:

A Java Server Pages file may be accessed in at least two different ways. A client's request comes directly into a Java Server Page. In this scenario, suppose the page accesses reusable Java Bean components that perform particular well-defined computations like accessing a database. The result of the Beans computations, called result sets is stored within the Bean as properties. The page uses such Beans to generate dynamic content and present it back to the client.

In both of the above cases, the page could also contain any valid Java code. Java Server Pages architecture encourages separation of content from presentation.

Steps in the execution of a JSP Application:

1. The client sends a request to the web server for a JSP file by giving the name of the JSP file within the form tag of a HTML page.
2. This request is transferred to the JavaWebServer. At the server side JavaWebServer receives the request and if it is a request for a jsp file server gives this request to the JSP engine.
3. JSP engine is program which can understands the tags of the jsp and then it converts those tags into a Servlet program and it is stored at the server side. This Servlet is loaded in the memory and then it is executed and the result is given back to the JavaWebServer and then it is transferred back to the client.

JDBC connectivity

The JDBC provides database-independent connectivity between the J2EE platform and a wide range of tabular data sources. JDBC technology allows an Application Component Provider to:

- Perform connection and authentication to a database server
- Manager transactions
- Move SQL statements to a database engine for preprocessing and execution
- Execute stored procedures
- Inspect and modify the results from Select statements.

Purpose

The generated application is the first version upon the system. The overall system is planned to be in the form of web-based architecture with homogeneous database platform. The major objective of the overall system is to keep the following components intact.

□ System consistency □ System integrity □ Overall security of data □ Data reliability and Accuracy □ User friendly UI both at HR Level and PM levels □ Considering the fact of generality and clarity □ To cross check that the system overcomes the hurdles of the version specific standards

Overall Description

Productive Perspective: This software has been developed as software for bank agents which helps to automate and track the activities in the agent consultancy. It also provides an interface for the customers to interact with Lending Tree Agent from online.

Basic structure of the system

- Maintains and manages the banks with which this agent has tied up.
- Maintains and manages different departments in the organization
- Maintains and manages the employees in different departments.
- Maintains and manages the Loans information in different banks
- Maintains and customer information and customer loan information.

Database design

Database Tables: The total number of database tables that were identified to build the system is 6. The major part of the Database is categorized as

1. Transactional components: The Transactional components are useful in recording the transactions made by the system. All the projects information was handled by these components

2. Data Dictionary components: These components are used to store the major information like Employee details, project details, requirement details, availability details, allocations details etc.

3. General components: These components are used to store the general information like login information etc.

Database Tables:

To create the application the following tables are used

Banks:	Contains details of Banks
Dept:	Contains details of different departments
Emp:	Contains employee information
Customers:	Contains customers information
Customer_Loan_Details:	Contains customers loan information
Customers_Verification:	Contains customer verification details info
Bank_Loan:	Contains Banks loan information
Admin:	Contains admin details
Messages :	Contains messages information

Banks:

Column Name	Type (size)	Constraint
BankID	Number (12)	Primarykey
BankName	Varchar2 (12)	
Branch	Varchar2 (12)	
Address	Varchar2 (12)	

Dept:

Column name	Type (size)	Constraint
DeptID	Number (12)	Primarykey
DeptName	Varchar2 (12)	

Emp:

Column name	Type (size)	Constraint
EmpID	Number (12)	Primary key
EName	Varchar2 (12)	
Design	Varchar2 (14)	
Dept	Varchar2 (13)	
Address	Varchar2 (12)	
Phone	Number (15)	
Email	Varchar (25)	

Customers:

Column name	Type (size)	Constraint
CustName	Varchar2 (25)	
CustAddress	Varchar2 (50)	
Qualification	Varchar2 (30)	
Occupation	Varchar2 (30)	
CompanyBS	Varchar2 (30)	
ComAddress	Varchar2 (50)	
ContactNo	Number (20)	PrimaryKey
Salary	Number (20)	
Designation	Varchar2 (40)	
Gender	Varchar2 (10)	
State	Varchar2 (30)	
AssignTo	Varchar2 (30)	
EmailID	Varchar2 (30)	
BuilderName	Varchar2 (30)	
BankName	Varchar2 (30)	
CDate	Date	
SanctionDate	Date	

Customer_Loan_Details

Column name	Type (size)	Constraint
CustName	Varchar2 (30)	
Email	Varchar2 (30)	
Ags	Date	
Wo	Varchar (30)	
Ps	Varchar (30)	
Ol	Int (11)	
Ap	Varchar (50)	
Pi	Varchar (50)	
Itr	Decimal (18,0)	
Pan	Varchar (50)	

Customers_Verification:

Column name	Type (size)	Constraint
CustName	Varchar2 (30)	
Email	Varchar (30)	
Phone	Number (12)	
Ap	Varchar2 (12)	
Cd	Varchar2 (12)	
Pe	Varchar2 (13)	

Messages:

Column name	Type (size)	Constraint
MsgID	Number (20)	PrimaryKey
From	Varchar2 (50)	
To	Varchar2 (50)	
MsgSubject	Varchar2 (50)	
Msg	Varchar2 (100)	
CDate	Date	

Admin:

Column name	Type (size)	Constraint
LName	Varchar2 (20)	PrimaryKey
Password	Varchar2 (20)	

Bank_Loans:

Column name	Type (size)	Constraint
BankName	Varchar2 (20)	
LoanType	Varchar2 (20)	
InterestRate	Number (10,5)	
LoanID	Number(10)	PrimaryKey

Builder_Details:

Column name	Type (size)	Constraint
BuilderName	Varchar2 (20)	
OffAddress	Varchar2 (30)	
RegNo	Number (20)	PrimaryKey
PanNumber	Varchar2 (20)	
Phone	Number (12)	
Email	Varchar2 (50)	
Website	Varchar2 (50)	
LP	Varchar2 (20)	
PNo	Varchar2 (20)	
Noc	Varchar2 (20)	
BP	Varchar2 (20)	
EC	Varchar2 (20)	
APF No	Number (30)	
Reason	Varchar2 (100)	

Cust:

Column name	Type (size)	Constraint
CName	Varchar2 (40)	PrimaryKey
Cpwd	Varchar2 (20)	

Chapter 5

Design

Document

Design Document

- The entire system is projected with a physical diagram which specifics the actual storage parameters that are physically necessary for any database to be stored on to the disk. The overall systems existential idea is derived from this diagram.
- The relation upon the system is structure through a conceptual ER-Diagram, which not only specifics the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.

ER-Diagrams

- The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the date modeling activity the attributes of each data object noted is the ERD can be described resign a data object descriptions.
- The set of primary components that are identified by the ERD are
 - ◆ Data object
 - ◆ Relationships
 - ◆ Attributes
 - ◆ Various types of indicators.
- The primary purpose of the ERD is to represent data objects and their relationships.

Unified Modeling Language Diagrams

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

Structural model view

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

Behavioral Model View

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

Implementation Model View

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

Environmental Model View

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

UML Diagrams:

Chapter 6

Program Design Language

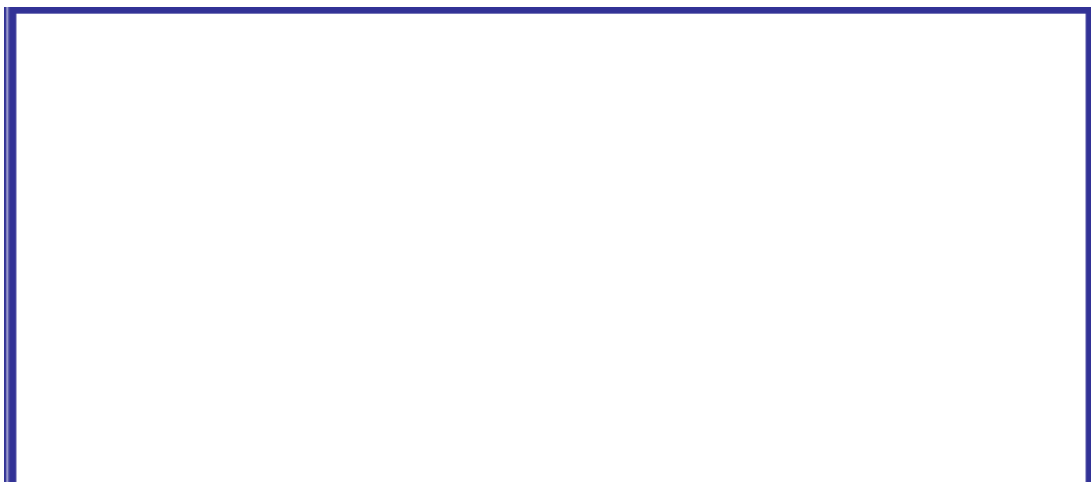
- The program design language is also called as structured English or pseudopodia. PDL is a generic reference for a design language PDL looks like a modern language. The difference between PDL and real programming language lies in the narrative text embedded directly within PDL statements.

The characteristics required by a design language are:

- ☐ A fixed system of keywords that provide for all structured constructs data declaration and modularity characteristics.
- ☐ A free syntax of natural language that describes processing features.
- ☐ Data declaration facilities that should include both simple and complex data structures.
- ☐ Subprogram definition and calling techniques that support various nodes of interface description.

PDL syntax should include constructs for subprogram definition, interface description data declaration techniques for structuring, conditions constructs, repetition constructs and I/O constructs.

PDL can be extended to include keywords for multitasking and/or concurrent processing interrupt handling, interposes synchronization the application design for which PDL is to be used should dictate the final form for the design language.



Testing

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also.

Psychology of Testing

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn't work. Testing is the process of executing a program with the intent of finding errors.

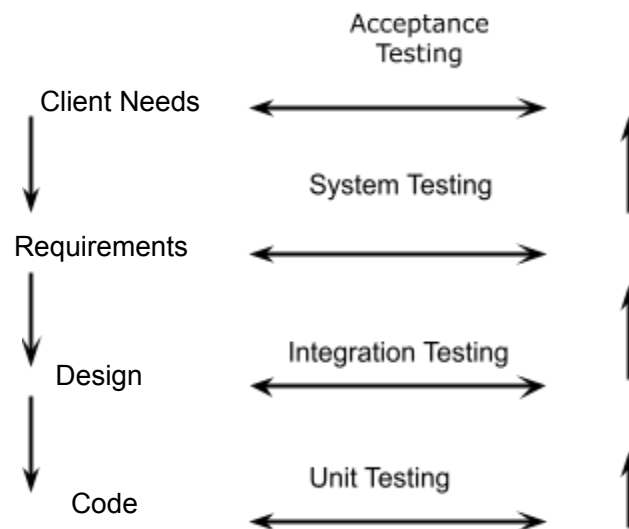
Testing Objectives

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

- ☐ Testing is a process of executing a program with the intent of finding an error.
- ☐ A successful test is one that uncovers an as yet undiscovered error.
- ☐ A good test case is one that has a high probability of finding error, if it exists.
- ☐ The tests are inadequate to detect possibly present errors.
- ☐ The software more or less confirms to the quality and reliable standards.

Levels of Testing

In order to uncover the errors present in different phases we have the concept of levels of testing. The basic levels of testing are as shown below...



System Testing

The philosophy behind testing is to find errors. Test cases are devised with this in mind. A strategy employed for system testing is code testing.

Code Testing:

This strategy examines the logic of the program. To follow this method we developed some test data that resulted in executing every instruction in the program and module i.e. every path is tested. Systems are not designed as entire nor are they tested as single systems. To ensure that the coding is perfect two types of testing is performed or for that matter is performed or that matter is performed on all systems.

Types Of Testing

- **Unit Testing**
- **Link Testing**

Unit Testing

Unit testing focuses verification effort on the smallest unit of software i.e. the module. Using the detailed design and the process specifications testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins.

In this project each service can be thought of a module. There are so many modules like Login, HWAdmin, MasterAdmin, Normal User, and PManager. Giving different sets of inputs has tested each module. When developing the module as well as finishing the development so that each module works without any error. The inputs are validated when accepting from the user.

In this application developer tests the programs up as system. Software units in a system are the modules and routines that are assembled and integrated to form a specific function. Unit testing is first done on modules, independent of one another to locate errors. This enables to detect errors. Through this errors resulting from interaction between modules initially avoided.

Link Testing

Link testing does not test software but rather the integration of each module in system. The primary concern is the compatibility of each module. The Programmer tests where modules are designed with different parameters, length, type etc.

Integration Testing

After the unit testing we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing

interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interactions.

In this project integrating all the modules forms the main system. When integrating all the modules I have checked whether the integration effects working of any of the services by giving different combinations of inputs with which the two services run perfectly before Integration.

System Testing

Here the entire software system is tested. The reference document for this process is the requirements document, and the goal is to see if software meets its requirements.

Here entire 'ATM' has been tested against requirements of project and it is checked whether all requirements of project have been satisfied or not.

Acceptance Testing

Acceptance Test is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behavior of the system; the internal logic of program is not emphasized.

In this project 'Network Management Of Database System' I have collected some data and tested whether project is working correctly or not.

Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once. The testing phase is an important part of software development. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

White Box Testing

This is a unit testing method where a unit will be taken at a time and tested thoroughly at a statement level to find the maximum possible errors. I tested

step wise every piece of code, taking care that every statement in the code is executed at least once. The white box testing is also called Glass Box Testing.

I have generated a list of test cases, sample data. which is used to check all possible combinations of execution paths through the code at every module level.

Black Box Testing

This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather getting into details at statement level. Here the module will be treated as a block box that will take some input and generate output. Output for a given set of input combinations are forwarded to other modules.

Criteria Satisfied by Test Cases

- 1) Test cases that reduced by a count that is greater than one, the number of additional test cases that much be designed to achieve reasonable testing.
- 2) Test cases that tell us something about the presence or absence of classes of errors, rather than an error associated only with the specific test at hand.

S No	Test Case	Action taken
1	Enter Wrong username and password for Admin at the login form.	Displays error message
2	Enter correct username and password for Admin at the login form	Displays Home Page
3	Create users of departments Pickup, Verifiaction and Legal	New Users of these departments will be created
4	Register the customer info	Customer Info has to be registered
5	Create a loan information related to a bank	New Loan Info has to be stored Related to that bank
6	Register a customer	It registers a new customer
7	Login as customer with correct username and password.	Displays customer home page
8	Select a bank and get the interest rates	Displays Interests Rates
9	Apply for the loan	Posts the application
10	View the loan applications by logging as admin	Displays new application
11	Forward the application to Pickup from admin	Forwards to an employee in Pickup Dept
12	Forward the application to verification Dept from Pickup dept	Forwards to Verification Dept
13	Forward the application to Admin Dept from Verification Dept	Forwards to Admin
14	Enter Builder Info	Creates APF No
15	Update the status of the application by admin finally	Updates Status
16	Check the status	Displays current status of the application
17	Send a message to a user	It Sents a message



Installation

- The database as it is developed by Oracle10g can be installed only by creating the tables in scott/tiger user with the help of LendingTree.sql file.
- Using Web components like JSP and servlets needs proper deployment on a top of a webserver called Apache Tomcat
- Copy the database driver inside common/lib of Tomcat
- Start the Server
- Open a browser and give the URL to access the application

*Conclusions &
Recommendations*
Chapter 9

Conclusions And Recommendations

The entire project has been developed and deployed as per the requirements stated by the bank agents consultancy, it is found to be bug free as per the testing standards that are implemented. It helps Bank Agents and Customers of Lending Tree and provides effective communication between them to provide the services very promptly with proper procedures. Any specification-untraced errors will be concentrated in the coming versions, which are planned to be developed in near future. The system at present does not take care payments info through this application since it requires payment gateway. Our company is planning to integrate in future and do it as an enhancement.

Bibliography:

References for the Project Development were taken from the following Books and Web Sites.

Oracle

PL/SQL Programming by Scott Urman

SQL complete reference by Livion

JAVA Technologies

JAVA Complete Reference

Java Script Programming by Yehuda Shiran

Mastering JAVA Security

JAVA2 Networking by Pistoria

JAVA Security by Scotl oaks

Head First EJB Sierra Bates

J2EE Professional by Shadab siddiqui

JAVA server pages by Larne Pekowsley

JAVA Server pages by Nick Todd

HTML

HTML Black Book by Holzner

JDBC

Java Database Programming with JDBC by Patel moss.

Software Engineering by Roger Pressman