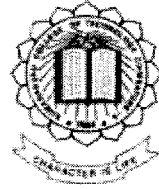




P-1912



On Demand Video Retrieval Using SOAP

By

Santhosh.S

Reg. No. 71204621036

Of

**KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE**

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements

for the award of the degree

of

MASTER OF COMPUTER APPLICATIONS

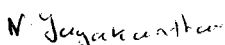
July, 2007




P-1912

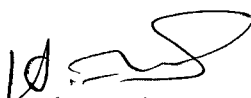
Bonafide Certificate

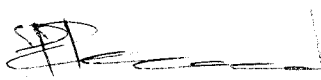
Certified that this project report titled **On Demand Video Retrieval using SOAP** is the bonafide work of **Mr.Santhosh.S** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.


Project Guide


Head of the Department

Submitted for the University Examination held on 03/07/2007


Internal Examiner


External Examiner 3/7/07



ILOGIX
INNOVATIVE SOLUTIONS

21/06/2007

TO WHOMSOEVER IT MAY CONCERN

This is to inform you that Mr. Santhosh. S. (Reg. No. 71204621036) final year Master of Computer Applications student of Kumaraguru College of Technology, Coimbatore has successfully completed his project work title "On Demand Video Retrieval using SOAP" in our organization during 26th December 2006 to 25th May 2007.

We wish him all success for his future endeavors

For Ilogix

Loganathan. D
H.R. Manager

Acknowledgement

I express my grateful thanks to our beloved principal, **Dr. Joseph V Thanikal**, Kumaraguru College of Technology, Coimbatore, for giving me an opportunity to take up this project.

I express my deep sense of gratitude to **Dr.S.Thangasamy**, Professor and Dean of Computer Science and Engineering department and **Dr.M.Gururajan**, Professor and Head of Department of Computer Applications for extending their help in providing all the facilities at college for the successful completion of the project.

I would like to express my deepest and sincere gratitude to **Mr.N.Jayakanthan**, Lecturer Department of Computer Applications for his guidance, support, cooperation and valuable suggestions during the course of this project.

I also thank **Mr.A.Muthukumar** Assistant Professor and course co-ordinator Computer applications, for providing all the desired documents and details regarding the various aspects necessary to the proper functioning of the system.

SYNOPSIS

This project entitled as **“On Demand Video Retrieval using SOAP”** is to provide live video streaming using Simple object access protocol.

There are many forms of rights management for digital media. But there continues some flaws in maintaining the accessibility and the limits they should be restricted. And when you look it as a web-based application, the problem of restricting unauthorized users also increases. For this we maintain different servers, the license server, which looks after the rights management, and the media server where the resources such as audio/video digital data are stored. The clients are identified by their licenses and these licenses are subjected to the expiry date and limited to resource. Here, only the view concept is used, this is because once the user has downloaded the media, the owner loses the control over the information which may be subjected to misuse.

Hence we prevent downloading of digital data. The client can view movie or listen music only on-line. These processes involve registration of users, selecting the privileges, assignment of password, setting the expiry data, receiving the license cost via card. The main concept is that neither the producer nor the consumer gets affected. They must be privileged with their promised rights. Different types of privilege are given for the client to choose from. The License Server acts as patrol between the client and the Media Server. As and when the client sends request to the Media Server, it checks the privilege from the License Server.

Table of Contents

Topic	Page No
Abstract	iii
List of Tables	VI
List of Figures	vii
1. Introduction	1
1.1 Organization Profile	1
1.2 Project Overview	2
2. System Analysis	8
2.2 Existing System	8
2.3 Proposed System	9
3. Development Environment	11
3.1 Hardware Requirements	11
3.2 Software Requirements	11
3.3 Programming Environment	12
3.3.1. Overview of .NET and Asp.Net	12
3.3.2 Overview of Servlets and JSP	17
3.3.3 Overview of MS-Access	23
4. System Design and Development	26
4.1 Design Process	26
4.1.1 Input Design	26
4.1.2 Output Design	27
4.2 Data Flow Diagram	29
4.3 Data Base Design	33

5. System Testing And Implementation	35
5.1 System Testing	35
5.2. White-Box Testing	36
5.3. Black-Box Testing	36
5.4. Quality Assurance	36
5.5. System Implementation	37
5.6. System Maintenance	38
6. Conclusion and Future Enhancement	39
6.1 Conclusion	39
6.2 Scope Of The Future Development	39
Reference	40
Appendices	41

List of Tables

Table Description	Page No
Table 4.3.1 Vendor Entry	33
Table 4.3.2: Audio List	33
Table4.3.3: Video List	33
Table4.3.4: Authentication	33
Table4.3.5: License Number	34
Table4.3.6: Package Details	34
Table4.3.7: Registration	34
Table4.3.8 :Transaction Details	34

List of Figures

	Figure Description	Page No
Figure 1.2	DRM Architecture	3
Figure 2.2	SOAP Description	9
Figure 3.3.3.1	Overview of Servlets	17
Figure 3.3.3.2	Servlet Life Cycle	17
Figure 3.3.3.3	Access Models -I	20
Figure 3.3.3.4	Access Models -I I	20
Figure 4.2.1.	DFD-Level 0	30
Figure 4.2.2.	DFD-Level 1	31
Figure 4.2.3.	DFD-Level 2	32
Screen Layouts		
Dot Net Screens		
	Bank server	41
	License server	46
Java Screens		
	Application Server	49
	Media Server	51

Chapter- 1

INTRODUCTION

1.2 Organization Profile:-

The **ILOGIX** is a software development firm providing technology solutions to a diverse customer. GTS an organization committed to share knowledge and provide quality in its core competencies and in a way nurture talent for the future endeavors. We believe that we would be able to offer a great support for Customers through our time tested and mature processes and proven global delivery model.

ILOGIX as a team has the prowess to have a clear vision and realize it too. As a statistical evaluation, the team GTS has more than 80,000 hours of expertise in providing real-time solutions in the fields of **Embedded Systems, DSP, Networking with C & C++, Client Server Technologies with Java, J2EE/J2ME, VB & VC++ and Operating System concepts with LINUX.**

Team ILOGIX always has one fundamental aspect clear in mind and it's about change, information age has one thing to remain for sure i.e. change and ILOGIX is a full blooded info-age complaint institution.

Our Vision &Our Mission

“Dreaming a vision is possible and realizing it is our goal.” Our mission is to partner with our customers, help them manage change and drive significant improvements in their business processes through the effective use of technology.

1.2 Project Overview:-

Digital Rights Management poses one of the greatest challenges for content communities in this digital age. Traditional rights management of physical materials benefited from the materials' physicality as this provided some barrier to unauthorized exploitation of content. However, today we already see serious breaches of copyright law because of the ease with which digital files can be copied and transmitted.

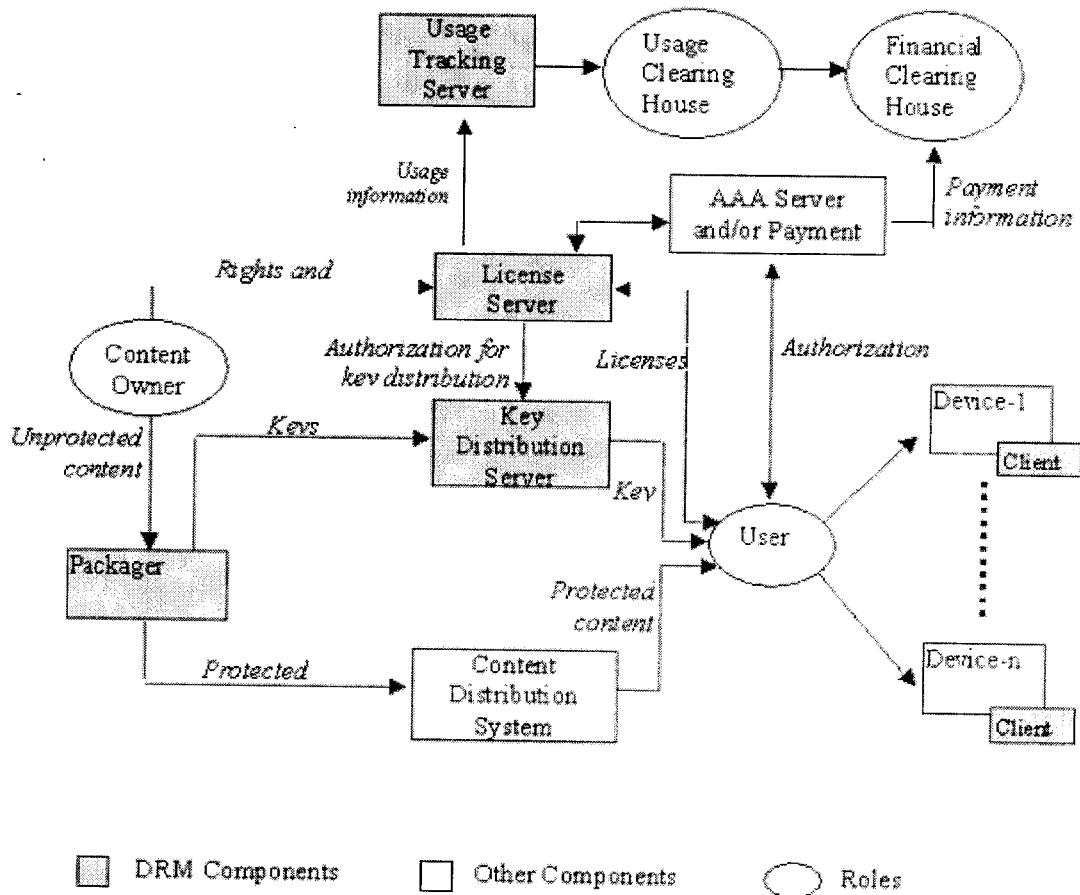
Previously, Digital Rights Management (DRM) focused on security and encryption as a means of solving the issue of unauthorized copying that is, lock the content and limit its distribution to only those who pay. This was the first-generation of DRM, and it represented a substantial narrowing of the real and broader capabilities of DRM. The second-generation of DRM covers the description, identification, trading, protection, monitoring and tracking of all forms of rights usages over both tangible and intangible assets including management of rights holder's relationships.

Additionally, it is important to note that DRM is the "digital management of digital rights". That is, DRM manages *all* rights, not only the rights applicable to permissions over digital content. In designing and implementing DRM systems, there are two critical architectures to consider. The first is the Functional Architecture, which covers the high-level modules or components of the DRM system that together provide an end-to-end management of rights. The second critical architecture is the Information Architecture, which covers the modeling of the entities within a DRM system as well as their relationships. (There are many other architectural layers that also need to be considered, such as the Conceptual, Module, Execution, and Code layers but these architectures will not be discussed in this article.

This project issues the Functional and Information Architecture domains and provides a Summary of the current state of DRM technologies and information architectures.

Objectives of the project:-

Figure 1.2. DRM Architecture



Component description:-

The following components are a part of the basic DRM system. All these components are not necessary in some cases and the architecture may be changed slightly without any efficiency loss.

- **Content owner:** The content owner provides content to the packager. The content owner also provides contracts and rights to the license server. These determine how the content is to be used by a given user.

- **Packager:**The packager can, if it is needed, compress the data, encrypt and watermarking it. The packager gets content as input and outputs keys to the key distribution server and protected content to the content distribution system.
- **Key distribution server:**This server gets keys as input from the packager and outputs these to the users. The license server obtains authorization for key distribution.
- **License server:**The license server gets contracts and rights from the content owner, and outputs licenses to users based on contracts, rights and the input from the AAA (authorization, authentication and access control) server and/or payment. The license server then provides authorization information to the key distribution server.

Content distribution system

This system may distribute the content over the Internet as streaming or download, distribute over physical media or broadcast.

- **Users, devices and clients:**Rights can be associated with users or a device, depending on the implementation, but users association is preferred. Super-distribution (distribution of protected content between users)may be supported.
- **AAA:**The server carries out authorization, authentication and access control. On top or instead, the user may have to pay a license fee for the service. After completion, this server signals to the license server to send licenses to the user.
- **Usage tracking server:**This server gets usage information from the license server, and outputs to the usage clearinghouse.

- **Usage clearinghouse:**The usage clearinghouse keeps information of the usage of the content and is responsible for the distribution of that information to all involved parties (content owner, financial clearinghouse etc).
- **Financial clearinghouse:**It enables financial transactions to be carried out. This covers the collection and contracted distribution to all involved parties (content owner etc).

Project Modules:-

License Server:- The license server provides the full authentication and act as a front end browser. The license server serves the following applications.

- User registration
- Package selection
- Payment validation
- Issuing license to clients

Media Server:- The media server stores the resources such as audio or video digital data. The media server serves the following applications

- Maintains media list (audio or video)
- Media files storage
- Interfacing with streaming server

Streaming Server:- The streaming server will process the selected media into stream. The streaming server serves the following applications

- Processing selected media to stream
- Identifying tracks to be transmitted
- Starting a new streaming process

Bank Server:- The bank server looks after the account transactions. The bank server serves the following applications

- User registration
- Payment transactions
- Creating web services

Implementation of SOAP:- In real-time implementation, the licensing server might manage by different vendors. To utilize the service of that vendor through media server will be difficult when the programming language and the operating platform differ from this implementation. To overcome this situation, SOAP is used to bridge the gap between cross language and cross platform.

Process Description:-

Process between Client and Application server : User name, license number & password are verified by the application server and the client's request is forwarded to the streaming server.

Process between license server & bank server : Account number of the client and the License server, are provided to the bank server. The Bank server performs the transaction.

Process between Client and license server : User's details, card number and password are provided by the client to the License server. Based on the client's choice license server issues the license after performing some sequence of processes.

Process between Application server & streaming server : Application server redirects the client's selected media file to the streaming server. Streaming server in turn converts the media file in to RTP streams and provides it back to the client.

Process between Application server & bank server : Application server sends its account number and license server's account number to the bank server .Bank server in turn performs the transaction process after validation of both the servers account.

Process between Streaming server & Client : RTP streamed media files are provided to the client. Client views those streamed media file using an JMF player which is embedded in client's browser.

Scope of the System:-

DRM is a technology that protects the digital content with rights and encryption. Digital content can be digital video, audio, e-books, documents, applications etc. In a DRM system digital content can be distributed securely between parties with the rights connected to the content. The rights are described in a special kind of mark up language (XML). There currently are not any standard specified for DRM by any standardization agency, hence there are big interoperability problems between the different implementations. There are some initiatives for standardization underway and this is necessary to facilitate for all parties.

The user may receive the protected content using any of the methods of delivery supported by the content distribution system. The user then accesses the AAA server for authorization, authentication and access control and/or may have to pay a fee. The AAA server then signals the license server to deliver the license to the user and also authorize the key distribution server to send the keys to the user. The key distribution server delivers the keys when the user begins to use the content. A DRM client is needed in the display device for the decryption and usage control of the content.

Chapter- 2

SYSTEM ANALYSIS

2.1 Existing System:-

Many companies have had problems protecting their digital belongings from being compromised and misused in an illegal way. Secret documents have been copied, printed or emailed. Digital audio, video and e-books have been copied, distributed illegally over the net and software has been hacked so it can run unlicensed. Companies have lost money because of this. To prevent this several DRM methods have been applied:

- **Digital watermarking:** Copyright and other source information is hidden inside the document or audio/video file without the user's knowledge. If a copy is made, the information will still remain in that copy. This technique will not prevent piracy or restrict use, but it can prove authorship and track copies to the original owner.

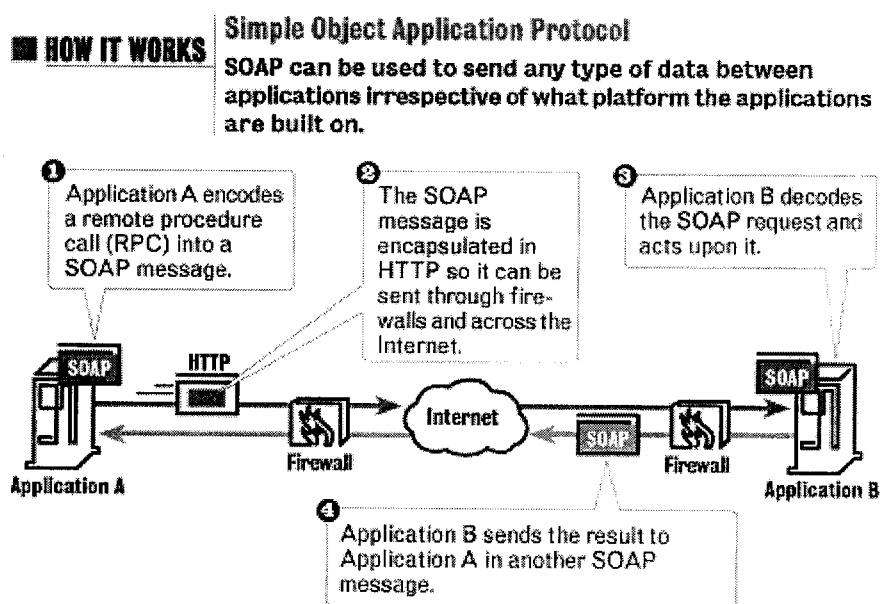
- **Physical copy protection:**A physical property of the medium is needed or a requirement of a specific piece of hardware attached to a user's computer, like a USB or smart card.

- **Encryption:** The content is encrypted and the distribution of keys or certificates is made separately. In some cases a third-party validation is required.

- **Product activation:**The product comes with an activation/identification code that has to be registered with the publisher before it can run properly. These DRM new techniques to manage the rights on digital content are needed. Most new initiatives are based on certificate-based protection and watermarking plus distribution of rights for each file and user.

2.2 Proposed System:-

Figure 2.2 SOAP Description



SOAP is a communication protocol that facilitates the interaction between application and web services located on remote computers. It stands for Simple Object Application Protocol. It can be used to develop a communications protocol that was easy to use and Flexible enough to apply to changes in the industry, and that could provide the ever-elusive interoperability. web services are a technology that allows programmers to create large-scale distributive Systems efficiently. SOAP, the Simple Object Access Protocol, is a protocol that allows the exchange of structured data between peers in a decentralized, distribution environment the structure of the data being exchanged is specified by an XML scheme.

The fact that SOAP messages in XML makes SOAP messages portable, because XML is a portable, system independent way of representing data. By representing data using XML, you can access data from legacy system as well as share your data with other enterprises. The data integration offered by XML also makes this technology a natural for web based computing such as web services. Firewalls can recognize SOAP packets on their content type (text/xml_SOAP) and can filter messages based on information exposed in the SOAP message header.

The SOAP specification describes a set of conventions for exchanging XML messages. As such, it forms a natural foundation for web services that also need to exchange information encoded in XML. although any two partners could define their own protocol for carrying on this exchange, having a standard such as SOAP allows developers to build the generic pieces that support this exchange. These pieces might be software that adds functionality to the basic SOAP exchange, or might be tools that administer SOAP messaging, or might even comprise parts of an operating system that supports SOAP processing. once this support is put in place, other developers can focus on creating the web services themselves.

SOAP is a simple XML-based protocol to let applications exchange information over HTTP.

- SOAP stands for Simple Object Access Protocol
- SOAP is a communication protocol
- SOAP is for communication between applications
- SOAP is a format for sending messages
- SOAP is designed to communicate via Internet
- SOAP is based on XML

Chapter- 3

3. DEVELOPMENT ENVIRONMENT

3.1.Hardware Specification

SERVER:-

- Processor : Pentium
- Speed : 933 MHz
- Hard Disk : 10GB
- RAM : 128 MB

CLIENT:-

- Processor : Pentium
- Speed : 933 MHz
- Hard Disk : 10GB
- RAM : 32 MB

3.2.Software Specification

- Operating System : Windows 2000
- Front end : J2EE,.NET
- Database : Ms-Access
- Web Browser : IIS Server
- Server : Apache Tomcat Server 1.4



P-1912

3.3. Programming Environment

3.3.1. NET Overview: -

.NET is the Microsoft's development model in which software becomes platform and device independent and data becomes available over the Internet. The .Net Framework is the infrastructure of .NET. .NET is built from the group up on open architecture. .NET is a platform that can be used for building and running the next generation of Microsoft Windows and Web applications.

The goal of the Microsoft .NET platform is to simplify web development. The .Net Framework provides the foundation upon which application and XML web services are build and executed the unified Nature of the .Net Framework means that all applications, whether they are windows applications, web applications are XML web services are developer by using a common set tools and code, and are easily integrated with one another.

The .NET Framework consists of:

- **The Common Language Runtime.** The runtime handles runtime services, including language integration, security and memory management. During development, the runtime provides features that are needed to simplify development.
- **Class Libraries.** Class libraries provide reusable code for most common task, including data access, XML web service development, web and windows forms.

What Problems does .Net solve?

The .Net Framework was developed to overcome several limitations that developers have to deal with when developing web applications and it makes strong use of the Internet as means for solving these limitations. Even with the advent of a global, easily accessible network of sharing information (The Internet), few application works on more than one type of client or the ability to seamlessly interact with other applications.

This limitation leads to two major problems that developers must confront:

- Developers typically have to limit their scope.
- Developers spent the majority of their time rewriting applications to work on each type of platform and client rather than spent their time designing new applications.

The .Net Framework solves the preceding two problems by providing the runtime, which is language-independent and platform independent, and by making use of the industry-standard XML. Language independence in .Net allows developers to build an application in any .Net-based language and know that the web application will work on client that supports .Net. XML web services use XML to send data, there by ensuring that any XML-capable client can received that data. Since XML is an open standard, most modern clients such as computer operating systems, cellular telephones, personal digital assistants (PDA's), and game from consoles, can accept XML data.

The .NET Framework Components:

The .NET Framework provides the compile time and runtime foundation to build and run .Net-based applications. The .NET Framework consists of different components that help to build and run.

Net-Based Applications:

- Platform substrate
- Application Service
- .Net Framework Class Library
- Common Language Runtime
- Microsoft ADO.NET
- ASP .NET
- XML Web services
- User Interfaces
- Languages

Benefits of using .NET Framework:

The benefits of using the .Net Framework for developing application include:

- **Based on Web standards and practices:**The .Net framework fully supports existing Internet technologies, including HTML, HTTP, XML, SOAP and other Web standards.
- **Design using unified application models:**The functionality of a .Net class is available from any .Net compatible languages are programming model. Therefore, the same piece of code can be used by windows applications, web applications and XML web services.
- **Easy for developers to use :** The .NET Framework provides the unified type system, which can be used by any .NET-compatible language. In the unified type system, all language elements are objects. These objects can be used by any .Net applications written in any .NET-based language.
- **Extensible classes :**The hierarchy of the .Net Framework is not hidden from the developer. You can access and extend .Net classes through inheritance

Introduction To Asp.Net

To create dynamic web pages by using server-side scripts. Microsoft has introduced ASP. ASP.NET is the .NET version of ASP. ASP.NET is a standard HTML file that contains embedded server-side scripts.

- ASP.NET enables you to access information from data sources, such as back-end database and text files that are stored on a web server or a computer that is accessible to a web server.
- ASP.NET enables you to use a set of programming code called templates to create HTML documents. The advantage of using template is that you can dynamically insert content retrieved from data sources, such as back-end database and text-files, into an HTML document before the HTML document is displayed to users. Therefore, the information need not be changed manually as and when the content s retrieved from data source change.
- ASP.NET also enables you to separate HTML design from the data retrieval mechanism. Therefore changing the HTML deign does not affect the program that retrieve data from the databases.

Features of Asp.Net:-

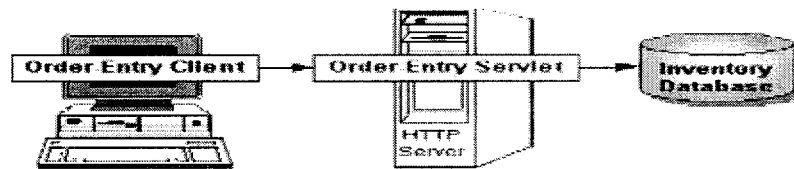
- **Power and Flexibility** :ASP.NET applications are based on Common Language Runtime (CLR). Therefore, the powerful and flexibility of the .NET platform is available enable you to ensure that the .NET Framework class library, messaging and data access solutions are seamlessly over the web. ASP.NET is also language-independent. Therefore, you can choose any .NET language to develop your application.

- **Compiled Code:**Code written in ASP.NET is compiled and not interpreted. This makes ASP.NET applications faster to execute than other server- side scripts that are interpreted, such as scripts written in a previous of ASP.
- **Enriched Tool Support :**The ASP.NET Framework is provided with a rich toolbox and designer in VS.NET IDE (Visual Studio .NET integrated development environment). Some of the features of this powerful tool are the WYSIWYG (What You See Is What You Get) editor, drag-and-drop server controls and automatic deployment.
- **Simplicity :**ASP.NET enables you to build user interfaces that separate application logic from presentation content. In addition, CLR simplifies application development by using managed code services, such as automatic reference counting and garbage collection. Therefore, ASP.NET makes it easy to perform common tasks ranging from submission and client authentication to site configuration and deployment.
- **Manageability :**ASP.NET enables you to manage Web application by storing the configuration information in an XML file. You can open the XML file in the visual Studio .NET IDE.
- **Scalability :** ASP.NET has been designed with scalability in mind. It has features that help improve performance in a multiprocessor environment.
- **Security :**ASP.NET provides a number of options for implementing security and restricting user access to a web application. All these options are configured within the configuration file.

3.3.3. Overview of Servlets

Servlets are modules that extend request/response-oriented servers, such as Java-enabled web servers. For example, a servlet might be responsible for taking data in an HTML order-entry form and applying the business logic used to update a company's order database.

Figure 3.3.3.1 Overview of Servlets

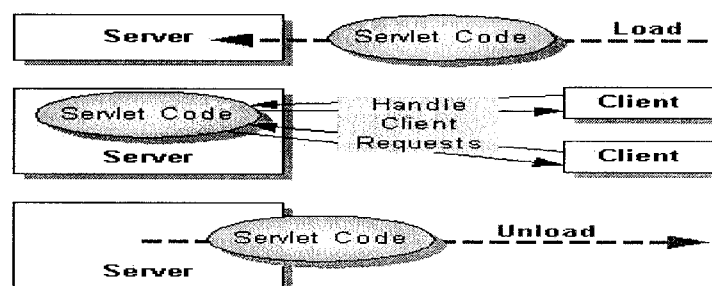


Servlets are to servers what applets are to browsers. Unlike applets, however, servlets have no graphical user interface. Servlets can be embedded in many different servers because the servlet API, which you use to write servlets, assumes nothing about the server's environment or protocol. Servlets have become most widely used within HTTP servers; many web servers support Java Servlet technology.

Servlet Life Cycle

- A server loads and initializes the servlet
- The servlet handles zero or more client requests
- The server removes the servlet
(some servers do this step only when they shut down)

Figure 3.3.3.2 Servlet Life Cycle



- **Initializing a Servlet:**-When a server loads a servlet, the server runs the servlet's init method. Initialization completes before client requests are handled and before the servlet is destroyed. Even though most servlets are run in multi-threaded servers, servlets have no concurrency issues during servlet initialization. The server calls the init method once, when the server loads the servlet, and will not call the init method again unless the server is reloading the servlet. The server can not reload a servlet until after the server has destroyed the servlet by running the destroy method.
- **Interacting with Clients:**-After initialization, the servlet is able to handle client requests. This part of the servlet life cycle was handled in the previous lesson.
- **Destroying a Servlet:**-Servlets run until the server destroys them, for example, at the request of a system administrator. When a server destroys a servlet, the server runs the servlet's destroy method. The method is run once; the server will not run the destroy method again until after the server reloads and reinitializes the servlet. When the server calls the destroy method, another thread might be running a service request. The Handling Service Threads at Servlet Termination lesson shows you how to provide a clean shutdown when there could be long-running threads still running service requests.

Other Uses for Servlets:-

Here are a few more of the many applications for servlets:

- **Allowing collaboration between people.** A servlet can handle multiple requests concurrently, and can synchronize requests. This allows servlets to support systems such as on-line conferencing.
- **Forwarding requests.** Servlets can forward requests to other servers and servlets. Thus servlets can be used to balance load among several servers that

mirror the same content, and to partition a single logical service over several servers, according to task type or organizational boundaries

Overview of Java Server^(TM) Pages:-

Java Server^(TM) 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 only eases maintenance headaches, it also allows web team members to focus on their areas of expertise. Now, web page designers can concentrate on layout, and web application designers on programming, with minimal concern about impacting each other's work.

- **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 interactions with associated components. The JSP engine for a particular server might be built-in or might be provided through a 3rd-party add-on. As long as the server on which you plan to execute the Java Server Pages supports the same specification level as that to which the file was written, no changes should be necessary as you move your files from server to server. Note, however, that instructions for the setup and configuration of the files may differ between files.
- **Composition:-** 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. Support for Enterprise Java Beans components will likely be added in a future release. As the default scripting language, Java Server Pages use the Java programming language. This means that scripting on the server side can take advantage of the

full set of capabilities that the Java programming language offers. Support for other scripting languages might become available in the future.

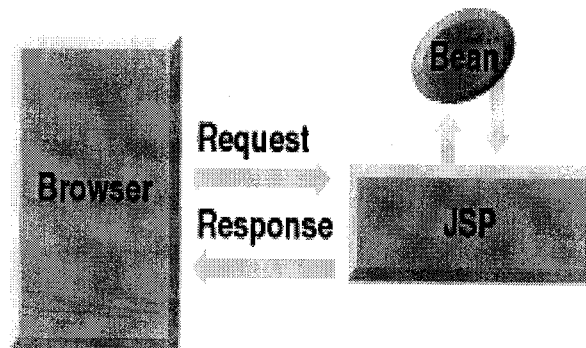
- **Processing:**-A Java Server Pages file is essentially an HTML document with JSP scripting or tags. The use of components is not required. The Java Server Pages file has a .jsp extension to identify it to the server as a Java Server Pages file. Before the page is served, the Java Server Pages syntax 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. Because it is standard HTML, the dynamically generated response looks no different to the client browser than a static response.

Access Models

A Java Server Pages file may be accessed in at least two different ways:

- A client request comes directly into a Java Server Page.

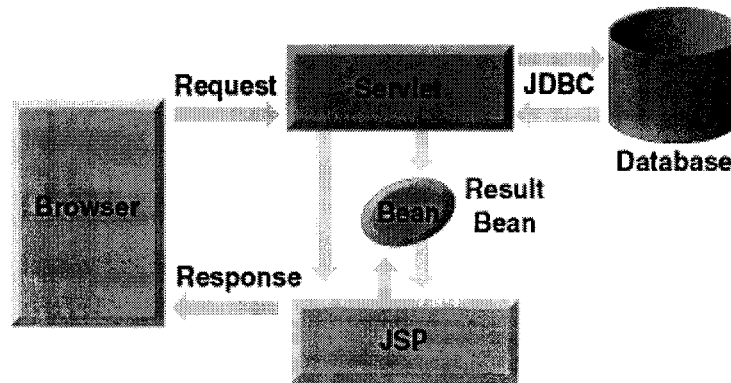
Figure 3.3.3.3



In this scenario, suppose the page accesses reusable JavaBean components that perform particular well-defined computations like accessing a database. The result of the Bean's computations, called *result sets* are stored within the Bean as properties. The page uses such Beans to generate dynamic content and present it back to the client.

- A request comes through a servlet.

Figure 3.3.3.4



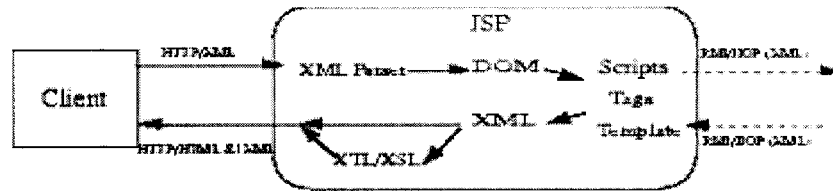
The servlet generates the dynamic content. To handle the response to the client, the servlet creates a Bean and stores the dynamic content (sometimes called the result set) in the Bean. The servlet then invokes a Java Server Page that will present the content along with the Bean containing the generated from the servlet. There are two APIs to support this model of request processing using Java Server Pages. One API facilitates passing context between the invoking servlet and the Java Server Page. The other API lets the invoking servlet specify which Java Server Page to use. .

How to Choose Between Access Models

- If a graphical interface (GUI) is necessary to collect the request data--use a Java Server Pages file.
- If the request and request parameters are otherwise available to the servlet, but the results of the servlet processing requires a graphical interface to present them--use a Java Server Pages file.
- If presentation layout is minimal (will not require very many println lines in your servlet ode) and you don't need to make that presentation logic available to a customer or your webpage designer, then a Servlet might suffice.

Using XML with JSP:

Figure 3.3.3.5



The JavaServer Pages technology is an ideal way to describe processing of XML input and output. Simple XML generation can be done by just writing the XML as static template portions within the JSP. Dynamic generation will be done through JavaBeans components, Enterprise JavaBeans components, or via custom tags that generate XML. Similarly, input XML can be received from POST or QUERY arguments and then sent directly to JavaBeans components, Enterprise JavaBeans components, or custom tags, or manipulated via the scripting. There are two attributes of JSP that make it specially suited for describing XML processing. One is that XML fragments can be described directly in the JSP text either as templates for input into some XML-consuming component, or as templates for output to be extended with some other XML fragments. Another attribute is that the tag extension mechanism enables the creation of specific tags and directives that are targeted at useful XML manipulation operations.

Features in JSP 1.0

The JSP 1.0 enables a tag extension mechanism for the creation of custom tags but such a mechanism will not appear until a later version of the specification.

- JSP standard directives
- JSP standard actions
- Script language declarations, scriptlets and expressions

A future version of the JSP specification will also add additional features to enhance the use of JSP pages in a J2EE platform.

3.3.4. Overview of MS-ACCESS:

About designing a database

Good database design ensures that your database is easy to maintain. You store data in tables and each table contains data about only one subject, such as customers. Therefore, you update a particular piece of data, such as an address, in just one place and that change automatically appears throughout the database. A well-designed database usually contains different types of queries that show the information you need. A query might show a subset of data, such as all customers in London, or combinations of data from different tables, such as order information combined with customer information.

Customers Table		
Customer ID	Company Name	City
BSBE V	B's Beverages	London
EAST C	Eastern Connection	London
HANAR	Hansri Carnes	Rio de Janeiro

Orders Table		
Order ID	Required Date	Customer ID
10931	21-Apr	HANAR
10943	05-Apr	BSBE V
10987	25-Apr	EAST C

London Orders for April			
Order ID	Required Date	Company Name	City
10943	05-Apr	B's Beverages	London
10987	25-Apr	Eastern Connection	London

The results you want from your database the forms and data access pages you want to use, and the reports you want to print don't necessarily provide clues about how you should structure the tables in your database, because you often base forms, reports, and data access pages on queries instead of tables.

Before you use Microsoft Access to actually build tables, queries, forms, and other objects, it's a good idea to sketch out and rework your design on paper first. You can also examine well-designed databases similar to the one you are designing, or you can open the Northwind sample database and then open the Relationships window to examine its design.

Determine the purpose of your database

As you determine the purpose of your database, a list of information you want from the database will begin to emerge. From that, you can determine what facts you need to store in the database and what subject each fact belongs to. These facts correspond to the fields (columns) in your database, and the subjects that those facts belong to correspond to the tables.

- Talk to people who will use the database. Brainstorm about the questions you and they would like the database to answer.
- Sketch out the reports you'd like the database to produce.
- Gather the forms you currently use to record your data.

Determine the fields you need in the database

Each field is a fact about a particular subject. For example, you might need to store the following facts about your customers: company name, address, city, state, and phone number. You need to create a separate field for each of these facts. When determining which fields you need, keep these design principles in mind:

- Include all information you need. Store information in the smallest logical parts.
- Don't create fields for data that consists of lists of multiple items
- Don't include derived or calculated data (data that is the result of an expression)
- Don't create fields that are similar to each other

Determine the tables you need in the database

Each table should contain information about one subject. Your list of fields will provide clues to the tables you need. For example, if you have a HireDate field, its subject is an employee, so it belongs in the Employees table. You might have a table for Customers, a table for Products, and a table for Orders.

Determine which table belongs to each field

In this respect, a table in a Microsoft Access database differs from a table in a flat file database such as a spreadsheet. When each piece of information is stored only once, you update it in one place. This is more efficient, and it also eliminates the possibility of duplicate entries that contain different information.

- Add the field to only one table.
- Don't add the field to a table if it will result in the same information appearing in multiple records in that table. If you determine that a field in a table will contain a lot of duplicate information, that field is probably in the wrong table.

Determine the relationships between tables

Now that you've divided your information into tables and identified primary key fields, you need a way to tell Microsoft Access how to bring related information back together again in meaningful ways. To do this, you define relationships between tables. You may find it useful to view the relationships in an existing well-designed database such as the Northwind sample database.

Refine your design

After you have designed the tables, fields, and relationships you need, it's time to study the design and detect any flaws that might remain. It is easier to change your database design now than it will be after you have filled the tables with data. Use Microsoft Access to create your tables, specify relationships between the tables, and enter enough sample data in your tables so you can test your design. To test the relationships in your database, see if you can create queries to get the answers you want. Create rough drafts of your forms and reports and see if they show the data you expect. Look for unnecessary duplications of data and eliminate them.

Chapter- 4

4. SYSTEM DESIGN AND DEVELOPMENT

4.1 Design Process

System design provides the understandings and procedural details necessary for implementing the system recommended in the system study. Emphasis is on the translating the performance requirements into design specifications. The design phase is a transition from a user-oriented document (System proposal) to document oriented to the programmers or database personnel. System Design goes through two phases of development:

- Logical Design
- Physical Design

4.1.1. Input Design

The input design is the link that ties the information system into the user's world. Input specifications describe the manner in which data enters the system for processing. Input design features can ensure the reliability of the system and produce results from accurate data, or they can result in the production of erroneous information.

- Developing specifications and procedures for data preparation.
- Steps necessary to put data into a usable form for processing.
- Data entry, the activity of putting data into the computer processing.

Objectives of Input Design

1. Controlling the amount of input required
2. Avoid delay
3. Avoiding errors in data
4. Avoiding extra steps
5. Keeping the process simple

In the input design users enter the name without space. The name must not contain any special character, but the name with the underscore and dots is permissible. The user will enter the proper mail id in the textbox. The user must specify the @ and dot symbol. The new user wants to register with this site he/she must fill all the fields. The user must enter the password. The password should be six characters. Only the registered users only enter into the site.

4.1.2. Output Design

Output design is one of the most important features of the information system. When the output is not of good quality the user will be averse to use the newly designed system and may not use the system. Various types of outputs required can be listed as below:

- External outputs, whose destination is outside the organization
- Internal outputs, whose destination is with the organization
- Operational outputs, whose use is purely with in the computer department e.g., program-listing etc.
- Interactive outputs which involve the user are communicating directly with the computer, it is particularly important to consider human factor when designing computer outputs.
- End user must find outputs easy to use and useful to their jobs, without quality output, user may find the entire system unnecessary and avoid using it.

In the output design there are lots of links to interact with this page. The user can go to any page. The links are very user-friendly. In the output design paging concept is also used to display the information. In this we can declare how many items will display in the particular page. The number of pages will display under the table. The page number also has a link to view that page. The next information view also via the link.

Other important factors that were taken into consideration are:

- The End user, who will use the output.
- The actual usage of the planned information.
- The information that is necessary for presentation when and how often output and their format is needed.
- While designing output for project based Attendance Compilation System, the following aspects of outputs designing were taken into consideration.
- The outputs (i.e., well formatted table outputs in the screen itself) designed are simple to read and interpret.
- Format of each output was another important point taken into consideration. Output media, for each output appropriate media is decided whether it will be displayed on screen or will be taken to printer or both.
- Other output design related specification, i.e., how frequently the outputs will be generated, how many pages or sheets approximately it will keep up, what is its planned use and output distribution to users are also taken into account.

Output contents originate from these sources:

- Retrieval from a data source.
- Transmission from a process or system activity.
- Directly from an input source.

4.2 Data Flow Diagram

The Data Flow diagram is a graphic tool used for expressing system requirements in a graphical form. The DFD also known as the “bubble chart” has the purpose of clarifying system requirements and identifying major transformations that to become program in system design. Thus DFD can be stated as the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. The DFD consists of series of bubbles joined by lines. The bubbles represent data transformations and the lines represent data flows in the system. A DFD describes what data flow is rather than how they are processed, so it does not depend on hardware, software, data structure or file organization.

Rules Used For Constructing a DFD

Process should be named and numbered for easy reference. Each name should be representative of the process. The direction of flow is from top to bottom and from left to right that is data flow should be from source to destination. When a process is exploded into lower level details, they are numbered. The name of the data stores, sources and destinations are written in capital letters. Process and data flow names have the first letter of each word capitalized.

The DFD is particularly designed to aid communication. If it contains dozens of process and data stores it gets too unwieldy. The rule of the thumb is to explode the DFD into a functional level, so that the sublevel does not exceed 10 processes. Beyond that, it is best to take each function separately and expand it to show the explosion in a single process. If a user wants to know what happens within a given process, then the detailed explosion of that process may be shown.

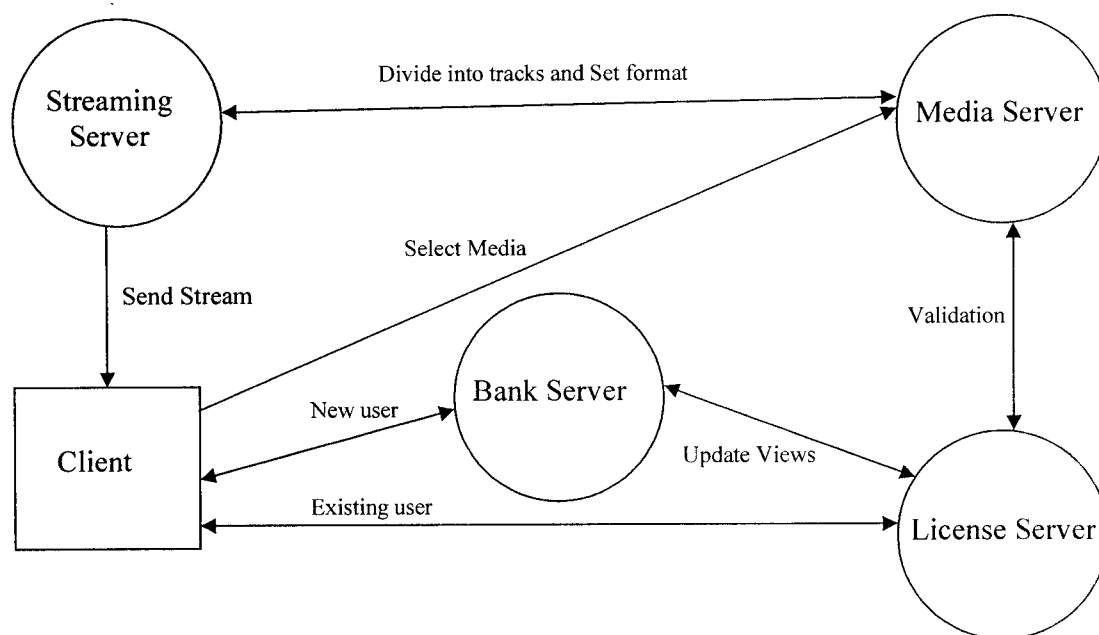
Figure 4.2.1.Level 0:

Figure 4.2.2.Level 1:

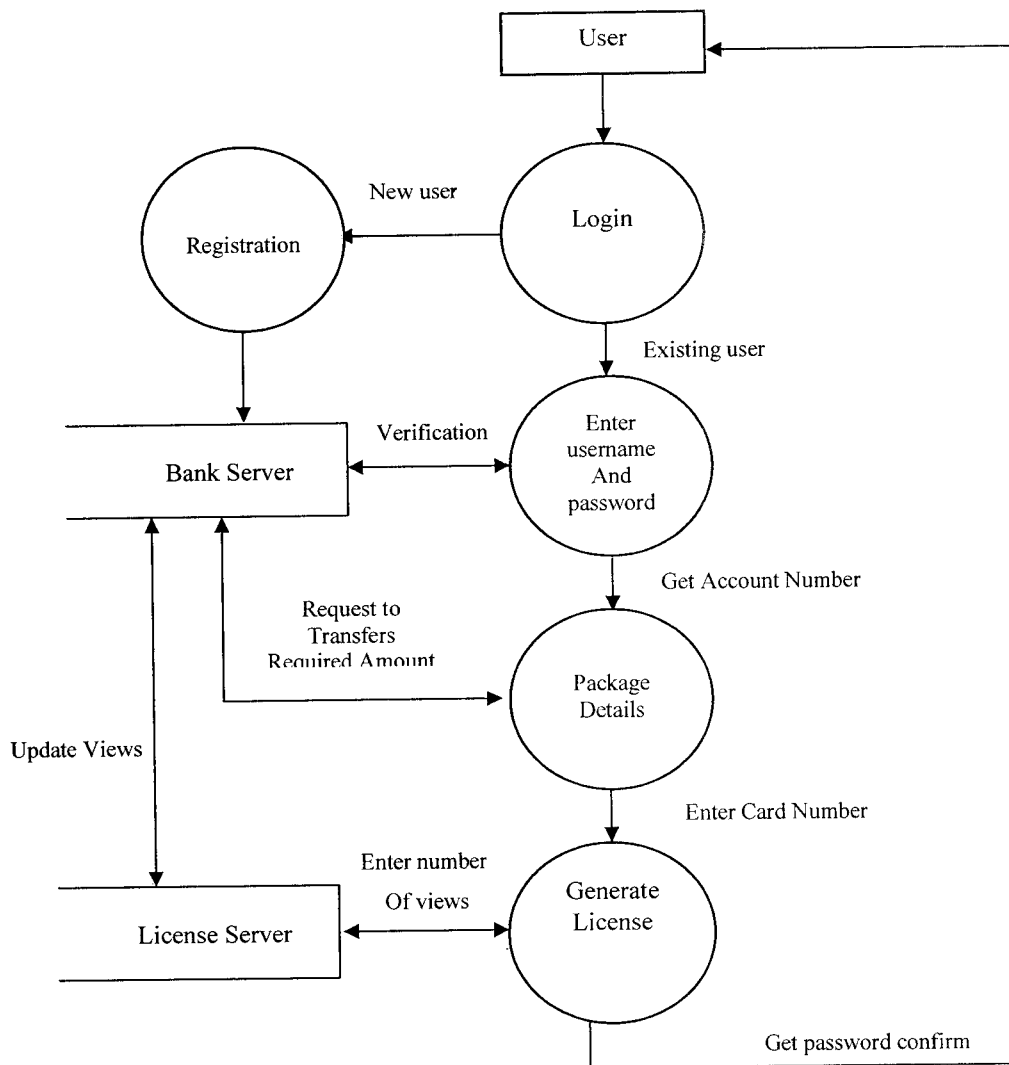
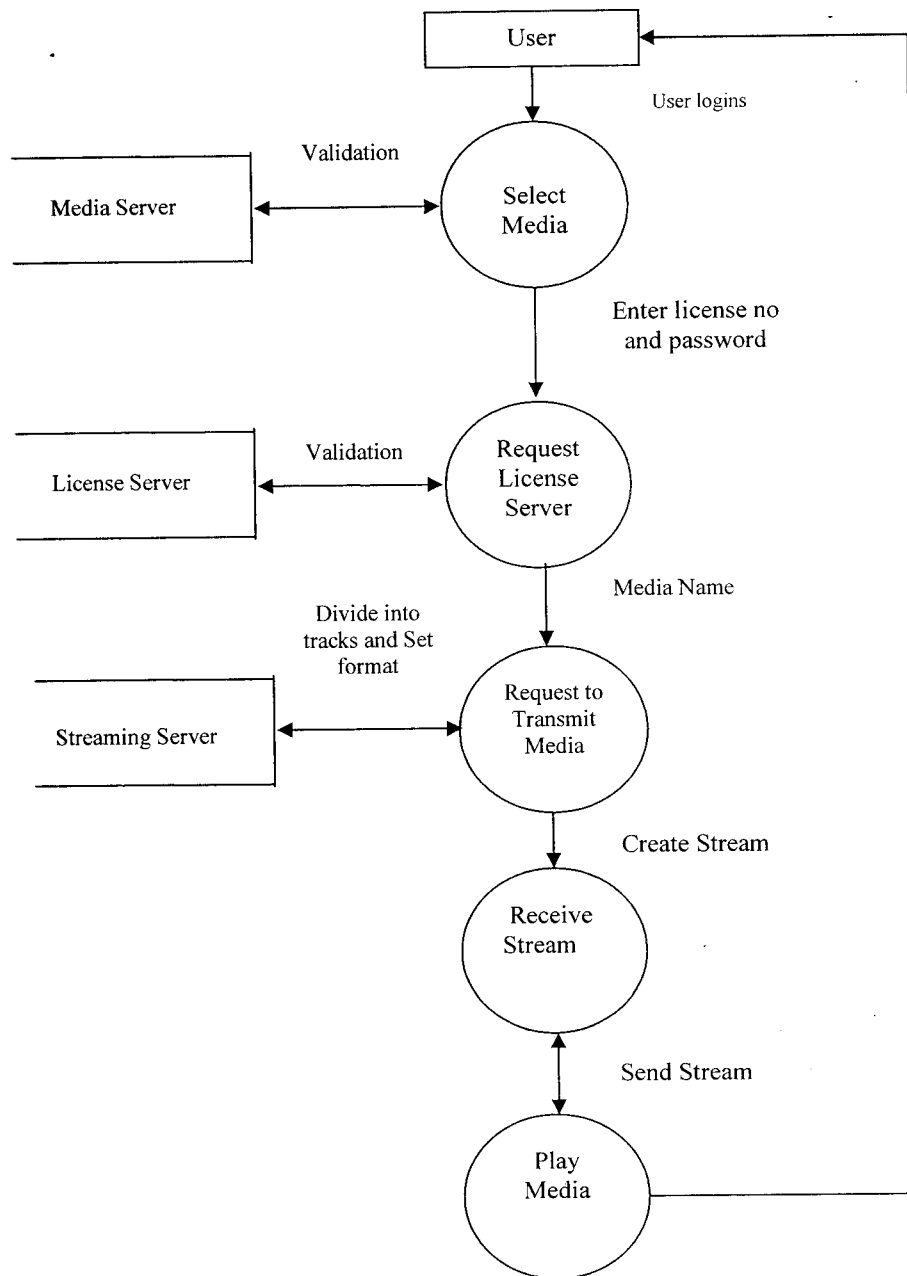


Figure 4.2.3.Level 2:



4.3 Data Base Design

Table Design: Java

Table4.3.1: Vendor Entry

Primary Key: ID

FIELD NAME	FIELD TYPE	DESCRIPTION
ID	Number(5)	Company ID
Com-Name	Varchar(10)	Company Name
Con-Person	Varchar(10)	Contact Person
Con-No	Number(7)	Contact Number
Email-Id	Varchar(15)	Email ID
Web-addr	Varchar(20)	Website Address
Acc no	Number(5)	Account Number
Acc Holder	Varchar(10)	Account Holder
Cost-per-view	Number(5)	Cost-per-View
Usr-Name	Varchar(10)	User Name
Pass-wd	Varchar(10)	Password
Date		Issuing Date
Views	Number(2)	No of views

Table4.3.2: Audio List

Primary Key: IDA

FIELD NAME	FIELD TYPE	DESCRIPTION
ID1	Number(5)	Id no1
ID2	Number(5)	Id no 2
F-name	Varchar(10)	File Name
Aud-name	Varchar(10)	Audio Name
Web-addr	Varchar(10)	Website Address

Table4.3.3: Video List

Primary Key: IDV

FIELD NAME	FIELD TYPE	DESCRIPTION
ID1	Number(5)	Id no1
ID2	Number(5)	Id no2
F-name	Varchar(10)	File Name
Mov-name	Varchar(10)	Movie Name
Web-addr	Varchar(10)	Website Address

Table Design:-.NET

Table4.3.4: Authentication

FIELD NAME	FIELD TYPE	DESCRIPTION
Usr-name	Varchar(10)	User name
Pwd	Varchar(10)	Password
Mode	Varchar(10)	Normal or New user

Table4.3.5: License Number

FIELD NAME	FIELD TYPE	DESCRIPTION
Lic-no	Number(10)	License number

Table4.3.6: Package Details

FIELD NAME	FIELD TYPE	DESCRIPTION
Pac-id	Varchar(10)	Package id
Mode	Varchar(10)	Audio or video
Valid	Number(10)	Validity period
No/. of items	Number(10)	Number of items
Package	Varchar(10)	Package name
Cost	Number(10)	Cost

Table4.3.7: Registration

FIELD NAME	FIELD TYPE	DESCRIPTION
Usr-name	Varchar(10)	User Name
Addr	Varchar(10)	Address
Pincode	Number(10)	Pincode
Con-No	Number(10)	Phone no
City	Varchar(10)	City
State	Varchar(10)	State
Cre-no	Number(10)	Credit card no
Holder name	Varchar(10)	Holder name

Table4.3.8 :Transaction Details

FIELD NAME	FIELD TYPE	DESCRIPTION
Pac-name	Varchar(10)	Package name
No of items	Number(10)	No/. of items
Cost	Number(10)	Cost
Cre-no	Number(10)	Credit card no
Holder-name	Varchar(10)	Holder name
Accno	Number(10)	Account number
Particulars	Varchar(10)	Package name & no/. of packages
Exp-date	Number(10)	Expiry date
Lic-no	Number(10)	License number

Chapter- 5

5.SYSTEM TESTING AND IMPLEMENTATION

5.1. System Testing

Testing is a process of checking whether the developed system is working according to the original objectives and requirements. System should be tested experimentally with the test data so as to ensure that the system works according to the required specification. When the system is found working, test it with the actual data and check the performance.

- **Module Testing:** This is also known as unit testing it focuses on the verification of the smallest unit of software design of the module. In this each module was found to be working satisfactory as per the expected output of the module.
- **Integration Testing:** Integration testing address the issues associated with the dual problems of verification and program construction. The main objective in this testing process is to take unit tested modules and build a program structure that has been dictated by design.
- **Validation Testing:** At the end of integration testing, software is completely assembled as a package, interfacing errors have been uncovered and correction testing begins. Since this system is used internally by the organization, user acceptance test is not given much importance.
- **Output Testing:** In output testing each test has a different purpose all the work should verify that all the system elements have been properly integrated and perform allocated functions. No system could be useful if it does not produce the required output. Here, output testing is done by checking whether the data being updated through database is coming in the correct portion and position of the website template.
- **Code Testing:** There are many strategies of testing the code of newly developed system. The first is to check, to see the logic involved and then the correctness. Tests are conducted based upon sample data, and the limits and ranges of attributes in screen entries were tested.

5.2. White-Box Testing:

White-box testing, sometimes called glass-box testing, is a test case design method, that uses the control structure of the procedural design to derive test cases. Using this testing we can derive test cases such that

- Guarantee that all independent paths within a module have been exercised
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds
- Exercise internal data structures to assure their validity

5.3. Black-Box Testing:

Black box testing focuses on the functional requirements of the software. That is, black box testing enable the software engineer to derive sets of input conditions that will fully exercise all function requirements for a program. It is not an alternative to white-box testing. Rather, it is a complementary approach that is likely to uncover a different class of errors now white-box methods. Black-box testing attempts to find errors in the following categories.

- Incorrect or missing functions.
- Interface errors.
- Performance errors, and
- Initialization and terminal errors.

5.4. Quality Assurance

The quality assurance is a planned and systematic pattern of action that are required to ensure quality in software. The implication of software is that many different constituencies in an organization have software quality assurance responsibility. Software engineers, project manager, customer, sales people and the individuals who serve within QA group.

5.5. System Implementation

Implementation is process of converting a new or revised system design into an operational one. The first task is implementation planning that is deciding on the methods and time scale to be adopted. The proper implementation involves conversion of existing clerical files to computer media and hence these files as they are get converted. Then the actual changeover from the existing system to the new system takes place.

System implementation phase is concerned with translation of the design specifications into the source code and internal documentation so that debugging, testing and modification are made easier. Goal can be achieved by making the source code clear and straightforward. Simplicity, clarity and elegance are the hallmarks of good programs.

The changeover plays a vital role, which checks the developed tool for the following requirements, and then only the developed tool will be accepted by the users. Once the system has been developed, the system has to be tested and if no bugs found, then it is implemented. The changeover can take place only when the system has been proved to the satisfaction of the system analysis and other implementation activities have been completed. User managers are satisfied with the results of the system, test, staff training and reference manuals.

A good test cannot solve a bad program and testing can never prove a program is right. A good test case designs that detects many errors is alarming. Testing requires skill and knowledge. If the system is approved to be error free it can be implemented. Implementation includes proper training to the end user. The implemented software should be maintained for prolonged running of the software. For every dollar spent developing software, we have spent at least two dollar maintaining it.

5.6. System Maintenance

Maintenance is the main sigma of system development. It holds the software industry combative, tying up programming resources. Analysts and programmer spend far more time maintaining programs than they do writing them.

There are other problems as well are:

- Maintenance is not as rewarding as exciting as development system. It is perceived neither skill not experience.
- Users are not fully cognizant of the maintenance problem or its high cost.
- Few tools and techniques are available for maintenance.
- A good test plan is lacking.
- Standards, procedures, and guidelines are poorly defined and enforced.
- There is practically no maintenance manager job classification in the MIS field.
- Programs are often maintained without care for structure and documentation.
- There are minimal standards for maintenance,
- Programmers expect that they will not be in their current commitment by the time their programs go into the maintenance cycle.

Chapter- 6

6.CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion

The **ON DEMAND VIDEO RETRIEVAL** has been developed using JMF, JAVA1.4, APACHE TOMCAT SERVER, Simple Object Access Protocol and MS – ACCESS 2000. All the requirements were satisfied by this project. It is effective and satisfies the user and vendor needs. The system has been done in connected facilities. The system is flexible to accommodate future changes. The implementation that has been done is step-by-step process. Each module has been developed and tested. The system was tested with all possible samples of data and was done with high degree of accuracy and user friendliness. Only when the user controls, the system is made run and therefore it should be made to run automatically when the user switches on the computer. Details amount modules and restricting functionalities can be added.

6.2.Scope of the future development

DRM allows owners and creators of digital content to specify how it should be used and at the same time makes sure that specified rights are followed. There has not been any working way of doing this before and DRM shows much promise on enforcing rights for digital content. The main positive aspects for the home users are the secure storage of data, the secure execution environment for applications and a wider range of cheaper digital media. The main concept is that neither the producer nor the consumer gets affected. They must be privileged with their promised rights. Different types of privilege are given for the client to choose from. The license server acts as patrol between the client and the media server. As and when the client sends request to the media server, it checks the privilege from the license server. In real-time implementation, the licensing server might manage by different vendors. To utilize the service of that vendor through media server will be difficult. When the programming language and the operating platform differ from this implementation. To overcome this situation SOAP is used to bridge the gap between cross-language and cross-platform.

BIBLIOGRAPHY

Books:-

1. Ernesto Diamni - **Fine Grained Access Control for SOAP E-Services**,
EEE Publications, 1993.
2. Erickson - **Enhanced Attribution for Networked Copyright
Management**, TM Publication, 1987.
3. John S.Erickson – **A Mediation Approach to DRM Technology**, AST
Publication, 1989.
4. Cliff Binstock – **The XML Schema Complete Reference**, AXT
Publishers, 1999.

Websites:-

www.omg.org

www.microsoft.com

www.lunasil.com

www.java.sun.com

www.tutorialmanaicas.com

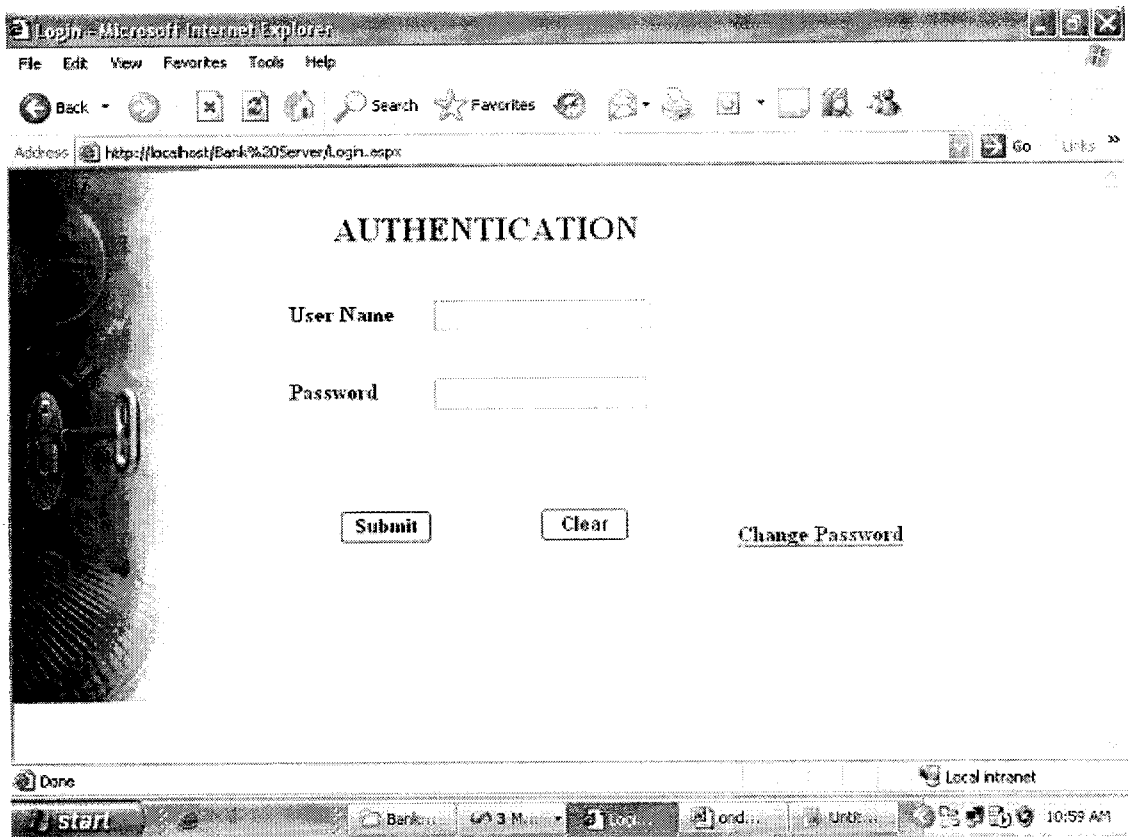
APPENDICES

SCREEN LAYOUTS

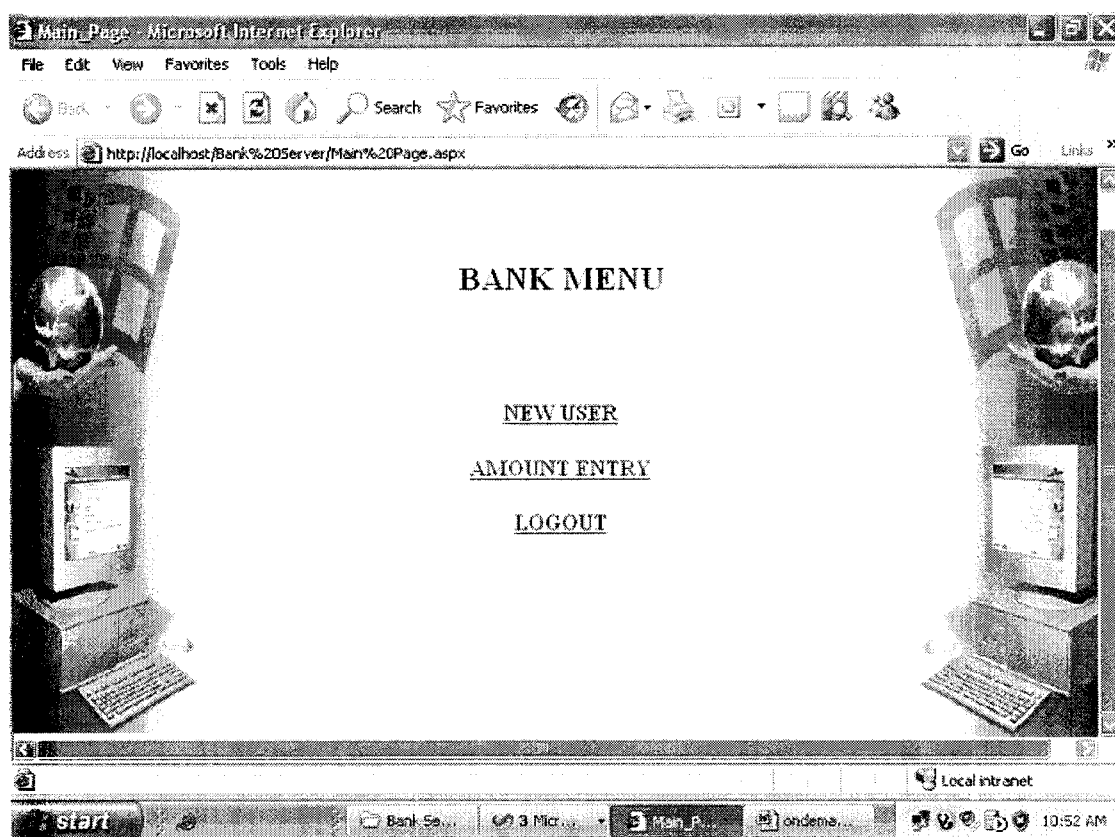
Dot Net Screens

BANK SERVER

Login Screen



Menu Screen



Registration Screen

Registration - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Go Links

Address <http://localhost/Bank%20Server/Registration.aspx>

Company Address	Ashok nager, Chennai	Date	1/11/2007
------------------------	-------------------------	-------------	-----------

Amount Details

Deposit Amount 12000

The Credit Card Number is 11111111127 The User Id is Aneesh3380SOP

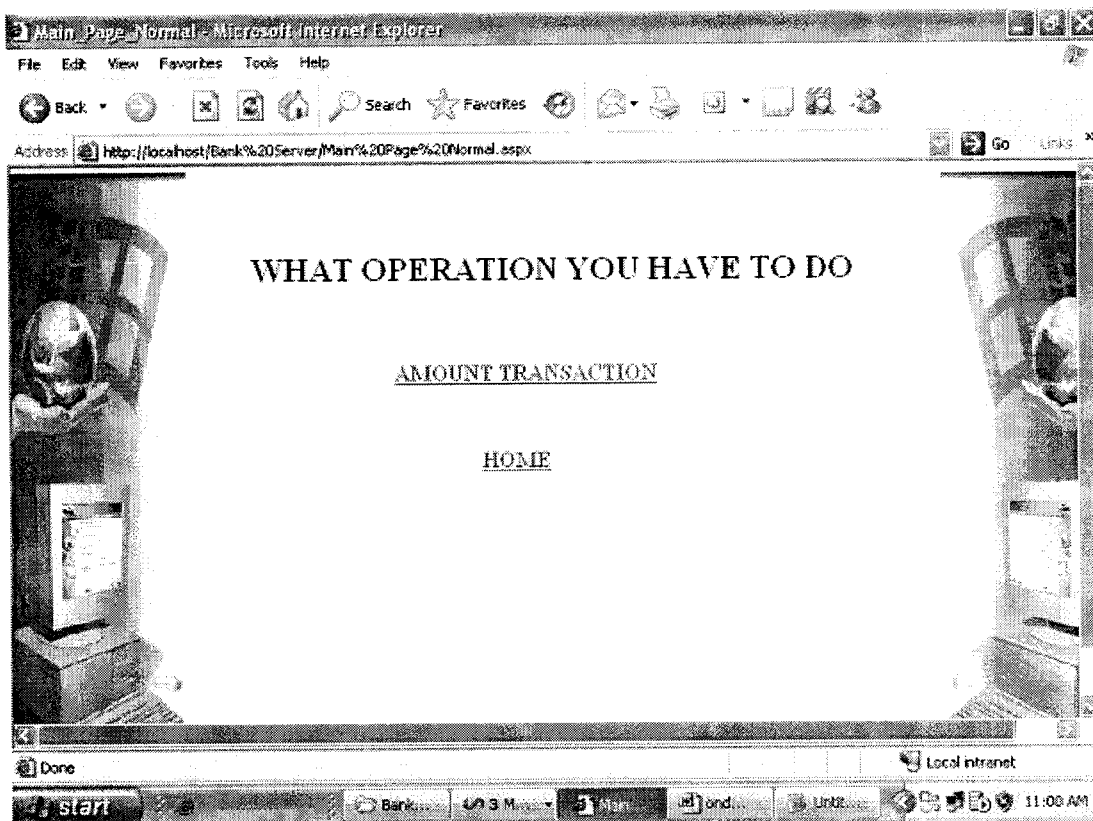
The Password is UV70LW Record Inserted

[Home](#)

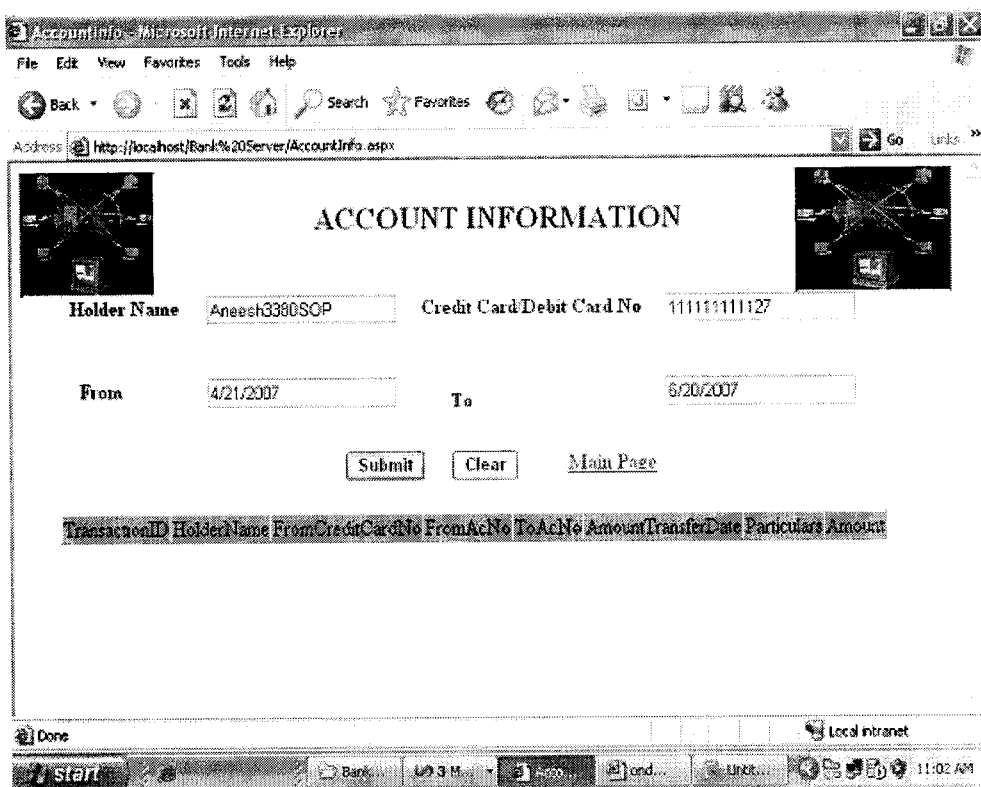
Done Local intranet

start Bank S... 3 Mic... 3 regist... 2 endem... 10:56 AM

Amount Transaction Screen

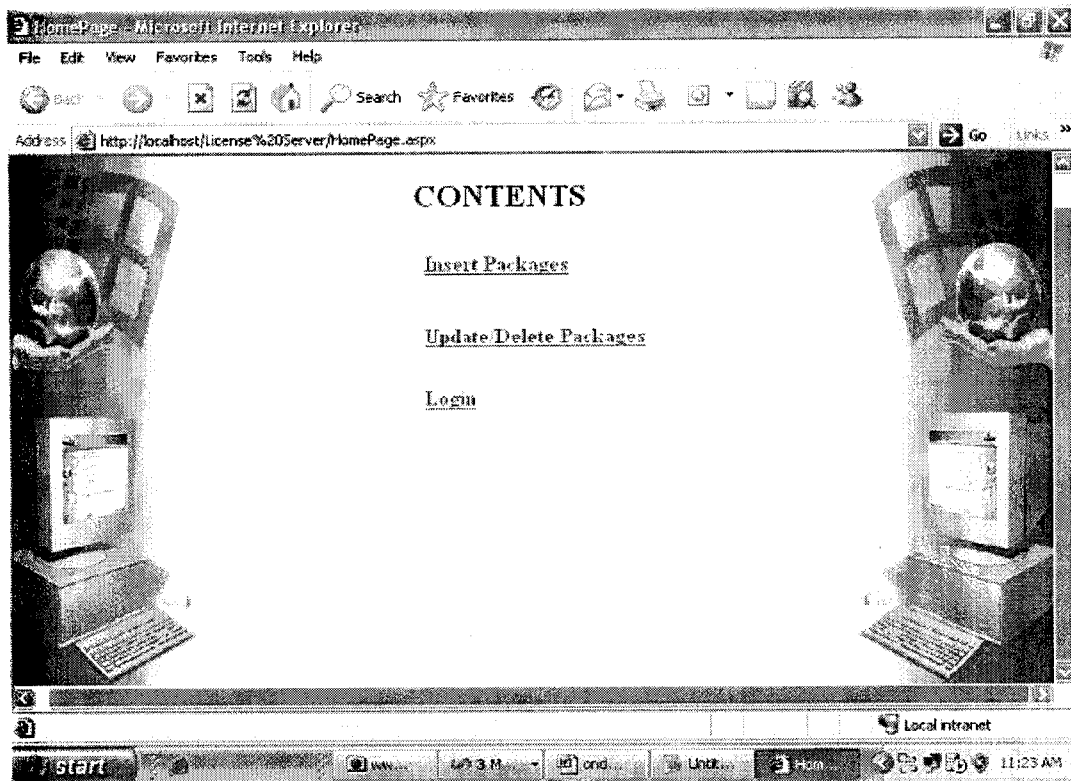


Account Information Screen



LICENSE SERVER

Content Screen



Registration Screen

The screenshot shows a Microsoft Internet Explorer browser window displaying a registration form. The address bar shows the URL: `http://localhost/License%20Server/RegistrationForm.aspx`. The form contains the following fields and values:

User Name	Aneesh3380SOP	Phone No	98756
Address	Akshaya Nagar	City	Dindigul
		State	TamilNadu
Pincode	624005		

Below the address fields, there is a section titled "Credit Card Details" with the following information:

Credit Card No	111111111127	The License Number is	123456789017
Holder Name	Aneesh	The Password is	SO63IU

At the bottom right of the form area, a message states: "Successfully license number inserted". The browser's status bar at the bottom shows "Done" and "Local intranet". The taskbar at the very bottom includes a "start" button and several open applications.

Package Screen

PackageDetails - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Stop Refresh Home Search Favorites Print Mail Stop

Address http://localhost/License%20Server/PackageDetails.aspx

PACKAGE DETAILS

Video Audio

Select	NoofItems	Ambio	Validity	NumberofItems	Package	Cost
<input type="radio"/>	1	AUDIO	6 MONTH	12	SANIA MIRZA	100
<input type="radio"/>	2	AUDIO	1 MONTH	12	shiva	12
<input type="radio"/>	3	AUDIO	4 MONTH	12	CUMMA	12
<input type="radio"/>	4	AUDIO	9 MONTH	100	sachin tendulkar	1000
<input type="radio"/>	5	AUDIO	3 MONTH	350	spl1	650

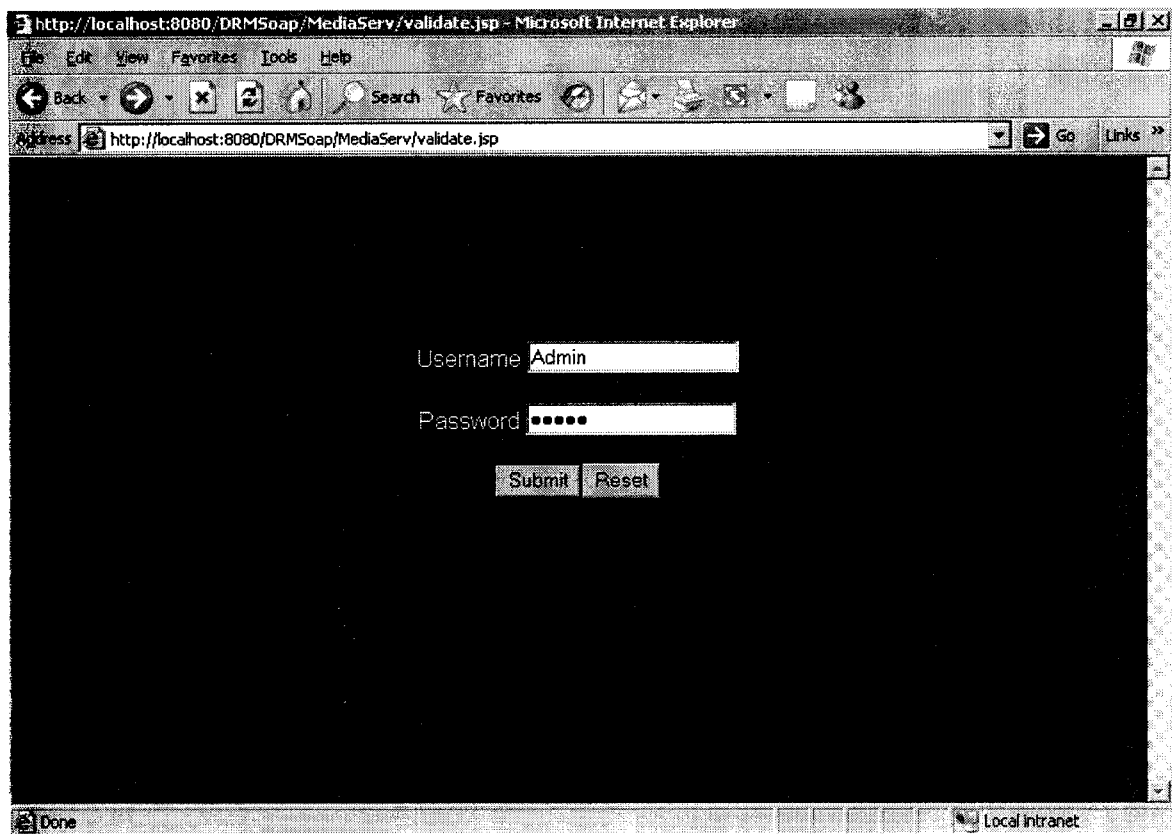
Done Local intranet

start 6/9 3 M... on... Unk... 11:27 AM

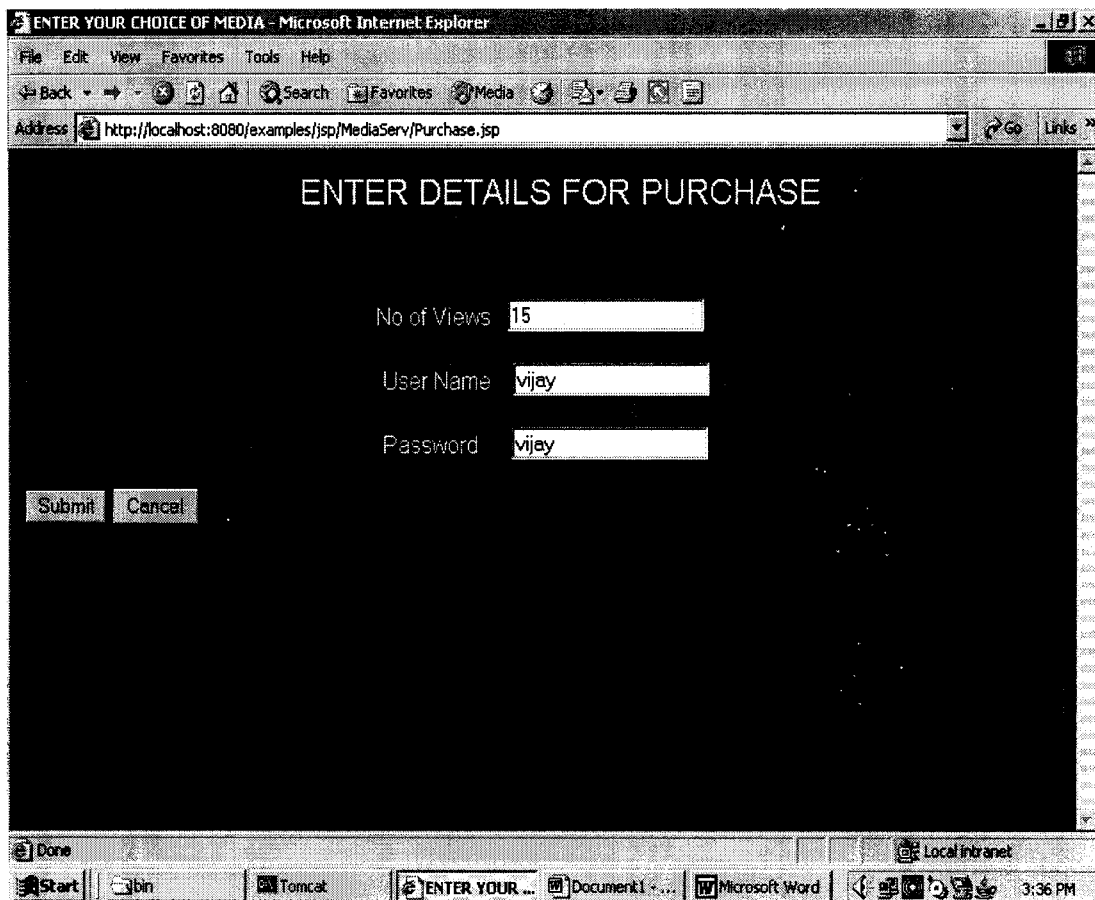
Java Screens

APPLICATION SERVER

Admin Screen



Purchase Screen



The image shows a screenshot of a Microsoft Internet Explorer browser window. The title bar reads "ENTER YOUR CHOICE OF MEDIA - Microsoft Internet Explorer". The address bar contains the URL "http://localhost:8080/examples/jsp/MediaServ/Purchase.jsp". The main content area has a black background with the text "ENTER DETAILS FOR PURCHASE" in white. Below this text are three input fields: "No of Views" with the value "15", "User Name" with the value "vijay", and "Password" with the value "vijay". At the bottom left of the form area are two buttons: "Submit" and "Cancel". The Windows taskbar at the bottom shows the Start button, several open applications (bin, Tomcat, ENTER YOUR..., Document1..., Microsoft Word), and the system tray with the time "3:36 PM" and "Local intranet" indicator.

ENTER YOUR CHOICE OF MEDIA - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media

Address <http://localhost:8080/examples/jsp/MediaServ/Purchase.jsp> Go Links

ENTER DETAILS FOR PURCHASE

No of Views

User Name

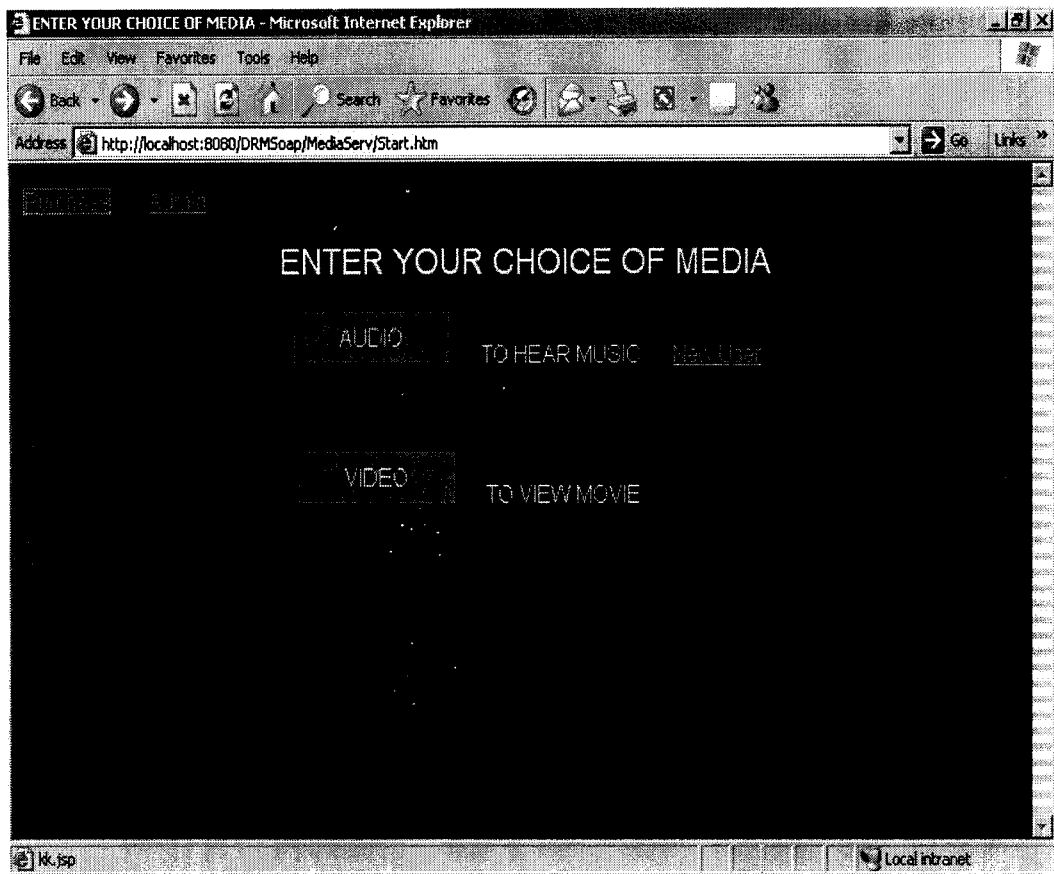
Password

Done Local intranet

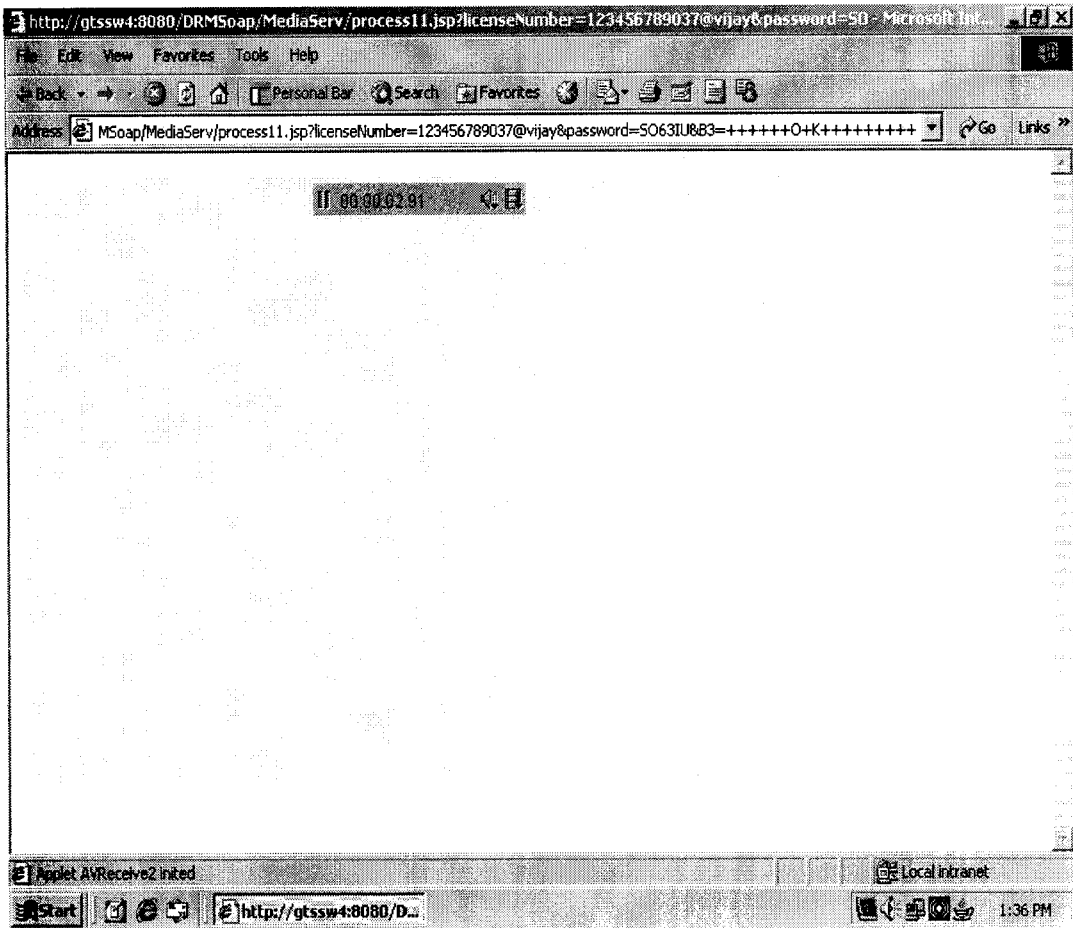
Start bin Tomcat ENTER YOUR ... Document1 ... Microsoft Word 3:36 PM

MEDIA SERVER

Home Screen



Audio Screen



Video Screen

