***S***

***YNOPSIS***

**THE TITLE OF THE PROJECT**

***MUSIC STORE MANAGEMENT***



**INTRODUCTION**

In today’s scenario computer is an important part day-to-day life. Every individual right

from a college student to a business tycoon utilizes the computer to browse the internet to

send email, chat, buy products via net etc.

This is inspired and encourage us to develop application software for every individual.

Our project entitle **“MUSIC STORE MANAGEMENT”** is designed that the Client’s requirement deals with opening an MUSIC STORE MANAGEMENT that will enable the client to mobilize its business through buying the different music related items in the website. It will enable a customer to have access to online shopping that is the site should be dynamic and more customer centric business procedure thereby maximizing the profit of the organization.

**“MUSIC STORE MANAGEMENT”** acts as a virtual showcase for a electronic music shop giving easy access to customers through login procedure, to interact with the **DATABASE**. The proposed solution will be developed using JSP for building the interfaces/Presentation Layer, EJB for business components/Logical Classes and Oracle8i as the database.

This is portal based automation project, which provides communication between the various users for music items such as music CD’s/Cassettes. User can purchase their needed music item anywhere, anytime just clicking their mouse. This project increases the interest of purchasing. User can easily finished their shopping through this shopping cart.

Screen design/Graphical User Interface:-

Graphical User Interface (GUI) that is straightforward and easy to navigate has been designed. This GUI provide various screens with appropriate incorporate icons, hyperlinks etc. to facilitate screen navigation and data entry. The user can easily add items to their cart, and they can easily remove items form the cart if they needed. The user has the ability to return to home page from any location within the application.

The following GUI form for user interaction can be conferred in the **“MUSIC STORE MANAGEMENT”** system

The **“MUSIC STORE MANAGEMENT”** is divided into three modules:

* + **Visitor module**
  + **User module**
  + **Administration module**

**Visitor:-**

If the client is a new user of the system then he performs the following tasks:

* New user can visit the home page of the online music site
* Search for a particular music item
* Cast his vote for a particular song
* View new releases
* For purchasing items, first the visitor has to click the register option in the client login area.
* The user must fill the registration form. The data are stored in to the database. It is not possible to leave any required data from the registration form.
* After that the visitor becomes a user and then he can enter into the shopping cart.
* View the Help page

Once the system authorizes the new client, then he can perform all the functions of the registered user.

**User: -**

If a registered user uses the following application then she/he has to perform the following tasks:

* First he/she has to login into the user interface which will help him/her to avail the services of the shopping cart.
* For the client to log in to the system it has to provide its username and password for authorization purpose.
* If the username and the password match with the entry in the database then the client enters the client shopping area. The client shopping area has three options to be utilized.
* The first option gives the client to search the list of music CD’s/Cassettes offered by the shop where the client can select the required one.
* The second option gives the client to add items to shopping cart.
* The third option gives the client to give feedback.
* The third option gives the client to logout from the site.

**Administrator:-**

The administrator has the following privileges:

* Delete a particular user from the database
* Add particular item to the inventory list
* Modify the details regarding a particular music item
* View the report, which includes sales reports for a specified day, previous week, and previous month.
* View messages and feedback from the customers

**FORMS:-**

We have different types of forms available for different module. These forms are listed and explained below.

**Visitor Module:**

This module has the following pages:

* The Home Page
* The Chart Toppers Page
* The Search Page
* The Vote Page
* The New Release Page
* The Register Page
* The Login Page

**TOOLS/ PLATFORM USED**

**HARDWARE SPECIFICATION**

Processor : Pentium-III

Memory : 128MB

Hard Disk : 20GB

Floppy Drive : 1.44MB

Monitor : 14’’ or above

Mouse : Standard Mouse

Key Board : 104 keys

**SOFTWARE SPECIFICATION**

Operating System  **:**  WINDOWS 2000 Server

Programming Language  **:** JAVA

Server Technologies  **:**  J2EE (JSP, SERVLETS, JDBC)

Application Server  **:** Jakarta Tomcat 4.1

Database Support  **:** ORACLE 8i

**ANALYSIS**

**DATA FLOW DIAGRAM**

**Data Flow:**

An arrow represents a data flow; it represents the path over which data travels in the system. A data flow can move between processes, flow into or out of data stores, to and from external entities.

**Bubbles (Process):**

A circle or bubble represents that transforms data from once form to another by performing some tasks with the data.

**Data store:**

A data store is a place where data is held temporarily from one transaction to the next or is stored permanently.

**External Entity**:

Which defines a source or destination of system data also called an external entity. Based on the working process of the proposed system Data Flow Diagram (DFD), is a model, which gives the insight into the information domain and functional domain at the same time can be drawn using OMT symbols. DFD is refined into different levels. The more refined DFD is more details of the system are incorporated. In the process of creating a DFD, we decompose the system into different functional subsystems. The DFD refinement results in a corresponding refinement of data.

The DFD of the “MUSIC STORE MANAGEMENT”

Each break-up has been numbered as per the rule of DFD. Here we attempted to incorporate all the details of the system and still it requires further improvement since the entire system is under study.

0 LEVEL D.F.D.:-

This is the context level D.F.D. of the proposed system the whole system has been depicted in a single bubble, primary input and output has been carefully noted and depicted in the way so that information flow continuity should not be lost in the next level. The purposed system is shown as a whole process and the inputs and outputs are shown with incoming and outgoing arrow from the system.

Context Level Diagram

Valid

communication

Services

User

User

**0 LEVEL D.F.D. FOR “MUSIC STORE MANAGEMENT”**

1ST LEVEL D.F.D.:-

This D.F.D. shows all the processes together with all the data stores (tables). It shows the true data flow i.e. how data is actually flowing in the system. Data is coming from which table and going into which table is clearly shown by this DFD. This DFD is the main reference for the development of the system. After understanding the whole system, the application developer will fall back upon this DFD during the Development phase.

Valid

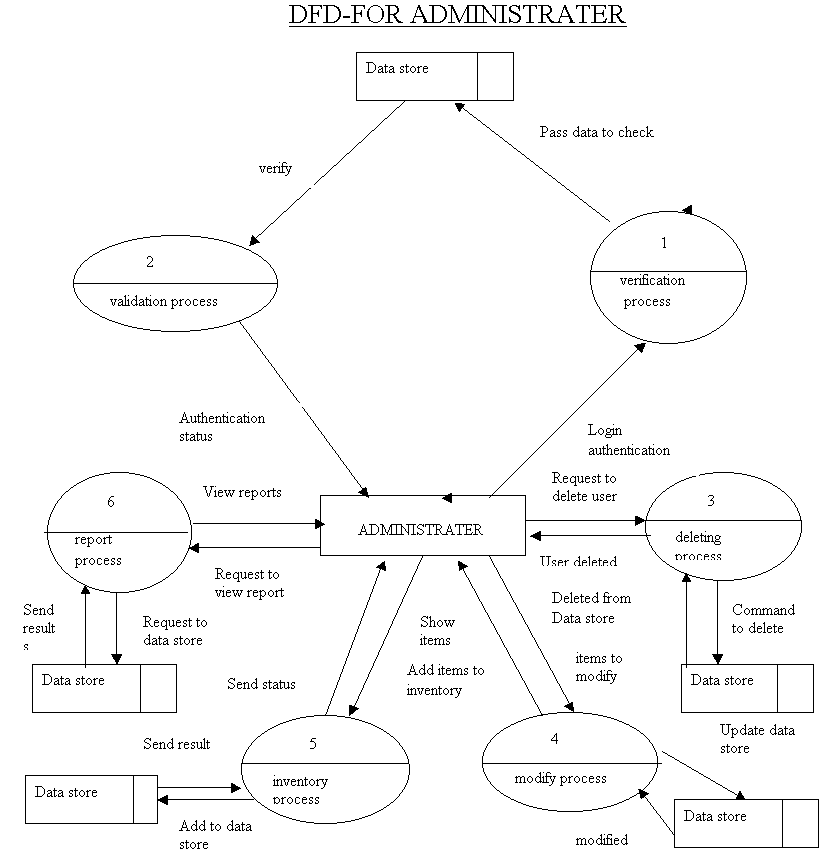
Login

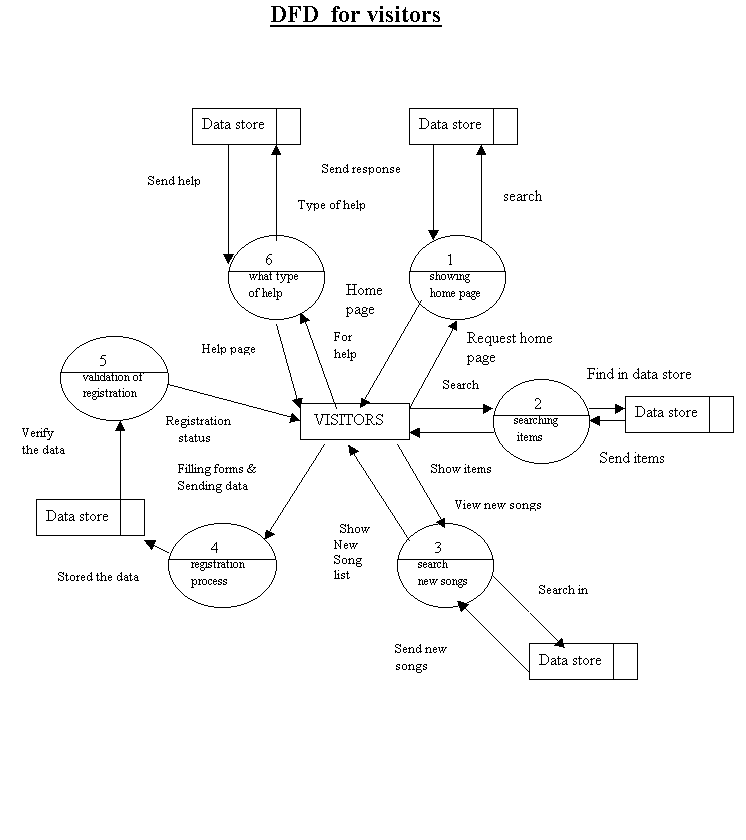
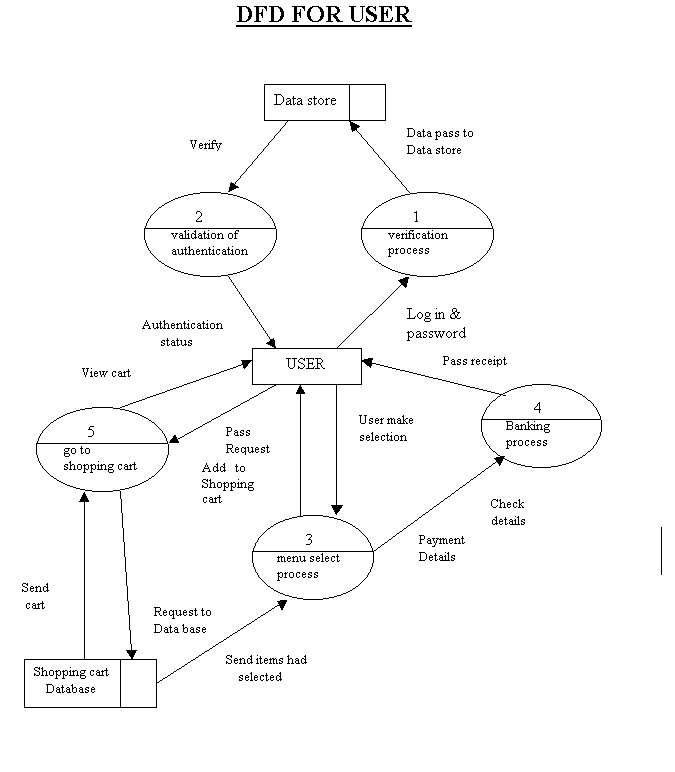
Valid

User

Invalid

Valid

******



**ER-Diagram**

Enter

Login

Information

Feed back

Master

Sale info

Reply

User information

Registration

Purchases

Reply info

Sales master

Vote

Vote info

Item master

[**Database Design**](http://databasesearch.blogspot.com/2010/04/database-fundamentals.html)

Database files are the key source of information into the system. It is the process of designing database files which are the key source of information to the system. The files should be properly designed and planned for collection, accumulation, editing the required information.

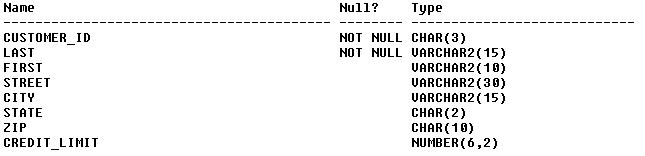
The objectives of the file design are to provide effective auxiliary storage and to contribute to the overall the efficiency of the computer program component of the system. In concepts of database design, there are two types of data – physical data and logical data.

Physical data is that which is written on those pieces of paper. Logical data are those, which are calculated based on some of the retrieved data in a certain sequence in summary form. In a computer-based data processing system, separation of physical and logical data provides the same advantages.

**The Following are Basic Facts about databases.**

|  |  |
| --- | --- |
| **Database Terms** | |
| * Attribute * Cardinality * Data Dictionary * DBMS Engine * Design Tools * Attribute's Domain * Entity * Entity Class * Father of Relational Databases * Foreign Key | * Hierarchy of Data Elements * Meta Data * Overhead Data * Primary Key * Relation * Relational Database * Runtime Subsystem * Schema * Transactions * User Data |

**ATTRIBUTE**  
An attribute is another word for field. In spreadsheet language it would be a cell. It is a place in a database table to store one piece of data of a given type. For example an attribute designated to hold a last\_name, could hold "Smith", but should not hold "Amy Smith".  
  
**CARDINALITY**  
Cardinality is a way to express minimum value and maximum value which are governed by the business rules. Cardinality refers to the required number of instances an entity must have in order to make the in a relationship in order for it to be valid. Minimum cardinality then for a one to many relationship would be one. Minimum cardinality for a basketball team would be 5, or you would be forced to forfeit the game. Maximum cardinality is the maximum number of entities which can occur in a relationship in order for it to be valid. In a one to one relationship the maximum cardinality would also be one. For a baseball team, during the normal season, the maximum cardinality would be 25 active players on the roster.   
  
**THE DATA DICTIONARY**

A database is self describing. By this we mean it documents itself through table structure outputs. One of the components of the data dictionary is the table data type layout.   
   
You can easily see how the fields are defined. The data types, lengths of the fields, and if they can be null or not. This is just one example of the data dictionary information provided by a DBMS.   
  
**THE DBMS ENGINE**

This is a component of the DBMS (Database Management System) which is the intermediary between the design tools and run-time sub-systems and the data. The DBMS engine receives requests form the other two components, which is presented in column and row format, translates them into commands which are passed to the operating system in order to provide read and write functions to the disk.

**THE DESIGN TOOLS**

One of the three components of a DBMS. This subsystem provides the tools to assist users and programmers in creating and modifying components of the database. such components are, Tables, Queries, Reports, and User Forms. Many DBMS products provide a programming environment to create databases which perform very complex tasks.

**Attribute's DOMAIN**

The domain of an attribute is the range of data it can contain. This is not to say the attribute can contain the entire range at one time. An attributes contents must be atomic, meaning they must be of a single bit of information about the theme of the record. For example an attribute named "JOB\_TITLE", from the EMPLOYEES table, could contain values from "Machine Operator", "Driver", "Foreman","Shift Manager", all the way up to "President". It can only hold one of these per record at a time. An attribute designated for "JOB\_TITLE" cannot hold any other type data, such as Salary, or Date\_Of\_Hire. Can you imagine having to look for the Date\_Of\_Hire somewhere in a table, but having no specified place? You might as well be searching text files again.  
  
**ENTITY**  
A entity is something that someone wants to track. An employee for example. It is basically the subject for a table. You gather data about the employee, you run queries to find out information about them, track their time, vacation, sick days etc... Therefore, an entity is very much the same as a record in a table.relational databases was E.F. Codd, who worked for IBM at that time. He published a paper titled "A Relational Model of Data for Large Shared Databanks" in June of 1970.   
  
**ENTITY CLASS**

A entity class is a collection of entities, as defined by their structure. There are usually many entities in an entity class, all of the same structure and type. In my mind, an entity class is the table which contains the entities.

**Father of Relational Databases.**

**FOREIGN KEY**

A foreign key is the same data field and type which is linked to a primary key in a corresponding table. For example in a transaction table The Customer\_ID would be the Foreign Key field. The Foreign Key is used to look up the Customer\_ID in the customer table where the Customer\_ID is the primary key.

**The Hierarchy of Data Elements in:**

* A file processing system
  1. Bits->Bytes or Characters->Fields->Records->Files
* A Database system
  1. Bits->Bytes or Characters->Fields->Records->Files+Metadata + Indexes + App MetaData.

**META DATA**

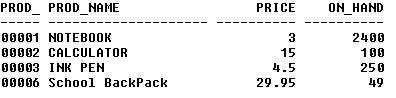
Meta data is the data about data. In the above example concerning the DATA DICTIONARY. Is an example of meta data. It is the self describing part of a database. Information such as the table names, user names, data types, and field sizes are all meta data, describing the database.  
  
**OVERHEAD DATA**

Overhead data is that which the system uses for itself. Indexes for example are overhead data. This is because the system uses indexes to speed searches, and to aid in joins. The overhead part comes in that this data also consumes processing time, and resources. Each time you update a table, the index must also be updated, which takes a bit of processing time but it also speeds up the search capability. You have to decide if the price in resources is worth the benefit of speed in creating and maintaining an index.

**PRIMARY KEY**

The primary key is that field, or fields, which by itself, or together uniquely identify each row in a table. The Primary Key is usually indexed, in some systems that is required. The primary key is normally the field or combined fields by which joins are linked. All data within each row or record should be dependent on the  entirety of the primary key. Primary Keys are used to normalize data tables.

**A RELATION**

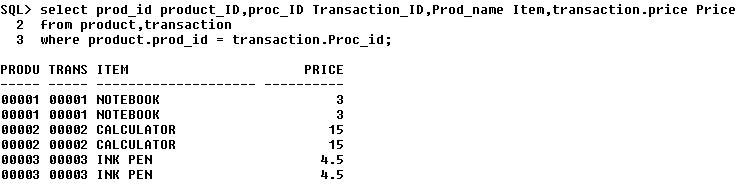
This is a table which, as one of its attributes has a unique identifier for each of it's records, also known as a primary key. In most cases, the primary key is indexed to enhance performance of the system by speeding the lookup capabilities of the DBMS.  
  
In the above example you can see the Product\_ID is unique for each item. This entire table, with the unique field is called a relation.

**A RELATIONAL DATABASE**

Relational database we entails some forms of data relationship. It gets its name for it's relation to other tables within the database. A relational database is set up so that the a key is presented in two or more tables. In one table it will be the primary key, however, in the other table it will be the foreign key. Where the primary key matches the foreign key is where the relationship occurs. You may have a one to one relation where only one of each key members can be present in each table. You may also have a one to many relationship, where only one member can exist in one table but many occurrences can be present in the other table. An finally you can have a many to many relationship, where many occurrences can be present in both tables. Below is an example of a one to many relationship. There is only one occurrence of the Product\_id in the products table, But many occurrences can exist in the transaction table.layout of the tables, attribute types and sizes, which fields are indexed, the relationships, domains, and business rules concerning a database. It is the design from which the database as well as its application programs were built. In a nutshell, the schema encompasses everything about the database.entire function. For example a sales transaction. You need the following.

**THE RUNTIME SUBSYSTEM**

This subsystem processes the application components that are developed using the design tools. For example Access has a runtime component that links data to forms, and reports. This is just part of the DBMS. The user or the developer need be concerned with how it works. When a given form is opened the runtime subsystem opens the required tables extracts the data and displays it to the user. There is also a component that facilitates the read and write requests for the applications.

**SCHEMA**  
The SCHEMA is the design of the database, and why it was created. The schema is the   
  
  
  
**TRANSACTIONS**  
Transactions are a group of sql statements which work together to perform an

1. A statement to add a record to the transaction table.
2. A statement update the Inventory Table.
3. A statement update the customer table. If necessary.
4. A statement to commit the data.

One of two things MUST happen. All of these statements must work together to accomplish their goal, or none of the statements work. That is the key to transaction processing, all or nothing. Log files are kept by the system to record what has been accomplished so, in the event something goes wrong, we know where to start. This is a way of maintaining the integrity of our data.   
  
**USER DATA**  
User data is just as the name implies. User data is the data which the user enters into the database tables.

**A COMPLETE structure**

**Analysis Report:**

System analysis is the first step towards the software building process. The purpose of system analysis is to understand the system requirements, identify the data, functional and behavioral requirements and building the models of the system for better understanding of the system.

In the process of system analysis one should first understand that, what the present system, what it does, is how it works (i.e. processes). After analyzing these points we become able to identify the problems the present system is facing. Upon evaluating current problems and desired information (input and output to the system), the analyst looks towards one or more solutions. To begin with, the data objects, processing functions, and behavior of the system are defined in detail. After this models, from three different aspects of the system-data, function and behavior. The models created during the system analysis process helps in better understanding of data and control flow, functional processing, operational behavioral and information content.

The proposed system contains the following main processes: -

**PROJECT DESCRIPTION**

**PROJECT MODULES:**

The proposed website will have its main page and will be mainly divided into partially dependent and partially independent modules as

**Visitor Module:**

This module has the following pages:

* The Home Page
* The Chart Toppers Page
* The Search Page
* The Vote Page
* The New Release Page
* The Register Page
* The Login Page

**Home Page:**

The Home page of the music application will be the first page to be displayed when a person visits the music application. The page will display a welcome message and, in addition, it will display various menus to facilitate navigation through the application.

**The Chart Toppers Page:**

The Chart Toppers page will display a list of the most popular songs. The popularity of the song will be decided based on the voting at the Vote Page

**The Search Page:**

The Search Page will enable users to perform a search of songs based on the song categories like song name, singer name or album name.

**The Vote Page:**

The Vote page of the music application will enable the user to vote for his favorite song.

**The Latest Releases Page:**

The Latest Releases page of the music application will display a list of the latest songs released during the last month.

**The Register Page:**

The register page will be used by visitors to register with the application. Visitors will need to provide information such as username, password, address, credit card details during registration.

**The Login Page:**

To logon to the music application, visitors will provide logon information in the Login page. Whether the visitor is a user or administrator is determined based on the user name provided by the visitor. Respective home pages for the users and administrators are displayed.

**User Interface Module:-**

The user interface module contains the following pages:

* The User Home Page
* The Shopping Cart Page
* The Wish list Page
* The Feedback Page
* The Buy Page

**The User Home Page:**

The User Home page is displayed to a user when the user logon to the music application.

**The Shopping Cart Page:**

The Shopping Cart contains the items that a user has selected for buying. The Shopping cart page of the music application will display a list of CDs, Cassettes and other music items selected by the user. Users can add the items to their wish list by clicking the check box next to the items and then clicking on the Add to Wish list button. They can also specify the quantity of the items.

**The Buy Page:**

To purchase music item, a user needs to select songs based on a search criterion. A user can search for items based on song categories and select the songs to purchase on the Buy Page.

**The Wish list Page:**

A user can move the items in the shopping cart to a wish list for later purchase. For example, if the credit limit of a user is exhausted and the user has an item in the shopping cart that he has inclined to purchase, he can put the item on the wish list and purchase the item later. The music application allows the user to view his wish list.

**The Feedback Page:**

To enable users to send feedback to the eMusic World site, there is a feedback page.

**Administrator Module:-**

The administrator Module has the following pages:

* The Delete User Page
* The Add Item Page
* The Modify Item Page
* The Report Page
* The View Wish list Page
* The View Messages Page

**The Delete User Page:**

An administrator can delete a user by using the Delete User page.

**The Add Item Page:**

The add Item page enables administrator to add items to inventory.

**The Modify Item Page:**

This page enables the administrator to manage and modify item details in the inventory.

**The Report Pages:**

The Music application will allow the administrator to view sales reports. The sales reports can be created for the previous week, previous month or for any particular date selected by the administrator.

**The Users Wish list Page:**

This page enables the application developers to view the wish list of the users.

**The View Messages Page:**

The users can send feedback about the web site by using the Feedback Page The administrator need to be able to view these messages so that they can act on the user’s suggestions. Therefore to enable administrators to view feedback from users, a View Message page has been created.

**DATABASE TABLE**

**Table Structure:**

There will be 7 tables in the **MUSIC STORE MANAGEMENT** The normalized form of the table with their structure is described as under: -

**The User Info Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Size** |
| **Request ID** | **Int** | **4** |
| **User name** | **Varchar** | **20** |
| **First name** | **Varchar** | **20** |
| **Middle name** | **Varchar** | **20** |
| **Last name** | **Varchar** | **20** |
| **Date of Birth** | **Date time** | **10** |
| **E-mail** | **Varchar** | **25** |
| **Address\_line1** | **Varchar** | **50** |
| **Address\_line2** | **Varchar** | **50** |
| **City** | **Varchar** | **25** |
| **State** | **Varchar** | **25** |
| **Credit card** | **Char** | **16** |
| **Credit card Type** | **Char** | **25** |

**The Feedback Master Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Size** |
| **User name** | **Var char** | **20** |
| **Message** | **Var char** | **50** |
| **Msg date** | **Date time** | **8** |

**The Item Master Table**

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data Type** | **Size** |
| **Item Code** | **Char** | **8** |
| **Title** | **Varchar** | **50** |
| **Rate** | **Float** | **50** |
| **Item Desc** | **Varchar** | **50** |
| **Singer** | **Varchar** | **50** |
| **Qty on hand** | **Int** | **50** |
| **Type** | **Varchar** | **50** |
| **Release Date** | **Date time** | **8** |

# 

# The Login Info Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Size** |
| **User name** | **Varchar** | **20** |
| **Password** | **Varchar** | **20** |
| **Secret Question** | **Varchar** | **20** |
| **Secret Answer** | **Varchar** | **20** |
| **Role** | **Char** | **10** |

**The Reply Info Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Size** |
| **Reply date** | **Date time** | **8** |
| **Reply msg** | **Varchar** | **50** |
| **Reply user name** | **Varchar** | **20** |

**The Sales Master Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Size** |
| **Sale Date** | **Date time** | **8** |
| **Item Code** | **Char** | **8** |
| **Sale qty** | **Int** | **4** |
| **Rate** | **Money** | **8** |
| **User name** | **Varchar** | **20** |

**The Vote Info Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Size** |
| **Item Code** | **Char** | **8** |
| **Number votes** | **Int** | **4** |

***PROCESS LOGIC OF EACH MODULE***

**Visitor Module:**

This module has the following pages:

* The Home Page
* The Chart Toppers Page
* The Search Page
* The Vote Page
* The New Release Page
* The Register Page
* The Login Page

**Home Page:**

The Home page of the music application will be the first page to be displayed when a person visits the music application. The page will display a welcome message and, in addition, it will display various menus to facilitate navigation through the application.

**The Chart Toppers Page:**

The Chart Toppers page will display a list of the most popular songs. The popularity of the song will be decided based on the voting at the Vote Page

**The Search Page:**

The Search Page will enable users to perform a search of songs based on the song categories like song name, singer name or album name.

**The Vote Page:**

The Vote page of the music application will enable the user to vote for his favorite song.

**The Latest Releases Page:**

The Latest Releases page of the music application will display a list of the latest songs released during the last month.

**The Register Page:**

The register page will be used by visitors to register with the application. Visitors will need to provide information such as username, password, address, credit card details during registration.

**The Login Page:**

To logon to the music application, visitors will provide logon information in the Login page. Whether the visitor is a user or administrator is determined based on the user name provided by the visitor. Respective home pages for the users and administrators are displayed.

**User Interface Module:-**

The user interface module contains the following pages:

* The User Home Page
* The Shopping Cart Page
* The Wish list Page
* The Feedback Page
* The Buy Page

**The User Home Page:**

The User Home page is displayed to a user when the user logon to the music application.

**The Shopping Cart Page:**

The Shopping Cart contains the items that a user has selected for buying. The Shopping cart page of the music application will display a list of CDs, Cassettes and other music items selected by the user. Users can add the items to their wish list by clicking the check box next to the items and then clicking on the Add to Wish list button. They can also specify the quantity of the items.

**The Buy Page:**

To purchase music item, a user needs to select songs based on a search criterion. A user can search for items based on song categories and select the songs to purchase on the Buy Page.

**The Wish list Page:**

A user can move the items in the shopping cart to a wish list for later purchase. For example, if the credit limit of a user is exhausted and the user has an item in the shopping cart that he has inclined to purchase, he can put the item on the wish list and purchase the item later. The music application allows the user to view his wish list.

**The Feedback Page:**

To enable users to send feedback to the eMusic World site, there is a feedback page.

**Administrator Module:-**

The administrator Module has the following pages:

* The Delete User Page
* The Add Item Page
* The Modify Item Page
* The Report Page
* The View Wish list Page
* The View Messages Page

**The Delete User Page:**

An administrator can delete a user by using the Delete User page.

**The Add Item Page:**

The add Item page enables administrator to add items to inventory.

**The Modify Item Page:**

This page enables the administrator to manage and modify item details in the inventory.

**The Report Pages:**

The Music application will allow the administrator to view sales reports. The sales reports can be created for the previous week, previous month or for any particular date selected by the administrator.

**The Users Wish list Page:**

This page enables the application developers to view the wish list of the users.

**The View Messages Page:**

The users can send feedback about the web site by using the Feedback Page The administrator need to be able to view these messages so that they can act on the user’s suggestions. Therefore to enable administrators to view feedback from users, a View Message page has been created.

**SOFTWARE REQUIREMENT SPECIFICATION**

The **software requirement specification** **(SRS)** is very important part of the software building process, which describes the actual user level requirement from technical point of view. I.e. what the user exactly wants? The objective of preparing the software requirement specification is to represent the requirements of the software in such a manner that ultimately leads to successful software implementation. It is the result of the analysis process of the software development. It should contain all the data the software is going to process, the function it will provide, and the behavior it will exhibit.

This Software Requirements Specifications (SRS) is defined in IEEE Std. 830-1993, IEEE Recommended Practice for Software Requirements Specifications. The synopsis is prepared in the way to fulfill almost all the points needed in S.R.S.

**TESTING PROCESS TO BE USED**

Quality assurance is an important step in software engineering. This overlaps with all the phases of development right from the requirement analysis. This quality requirement of the software system must be clearly extracted during the requirement analysis and all the subsequent phases should be made biased to that, the final testing will become trivial and less expensive.

There are number of quality parameters like correctness, accuracy, reliability, robustness, efficiency, effectiveness, reusability, maintainability etc.. The state of requirement of each of these parameters will vary depending upon the name and domain of the application. The testing should be done at the end of all development steps. Even though the final testing and verification are inevitable for better life and functionality of the software.

The different software testing approaches and methods like white box testing and black box testing. The major phases in testing are design of test plan, setting up test case and test candidate and test procedure, testing and correction. This is a cycle process and the software will circulate through all the steps till it attends the required quality. The testing is carried in the following steps.

**4.1 Unit testing**

Unit testing focuses verification effort on the smallest unit of software design the module. Using the details design description as a guide, important control paths are tested to uncover errors within boundary of the boundary of the module. The relative complexity of tests and the errors detected as a result is limited by the constrained scope established for unit testing.

Unit testing is normally considered an adjacent to coding steps. After source level code has been developed, reviewed, and verified for correct syntax, unit test case design begins. A review of design information provides guidance for establishing test cases that are likely to uncover error in each case of the categories discussed above. Each test case should be coupled with a set of expected results.

**4.3 Integration testing**

Integration testing is systematic technique for constructing the program structure while at the same time conducting test to uncover error associated with interfacing .The objective is to take unit tested modules and build a program structure that has been dictated by design.

There is often a tendency to attempt no incremental integration; that is to construct the program using a “big bang “approach. The entire modules are combined in advance. The entire program is tested as whole and chaoses usually result! A set of error is encountered. Correction is difficult because the isolation of cause is complicated by the vast expanse of entire program. Once errors are corrected, new ones appear and process continues in a seemingly endless loop.

**SECURITY MECHANISMS**

**TESTING & VALIDATION**

The approach of web app testing adopts the basic principle for all software testing and applies a strategy and tactics that have been recommended for object oriented system .the following steps summarizes the approach:

1. T**he content model for the web app is reviewed to uncover errors.**

This is like copy-editing.

**2. The design model for the web application as reviewed to uncover**

**navigation error.**

Use cases derived as part of the analyst activity, allow a web Engineer exercise each usage scenario against the architectural and navigational design. These non-executable test help uncover error in navigation.

3. **Selected processing component and web page is unit tested**.

When web apps are considered, the concepts of the unit changes, each web page encapsulated in itself content navigation link as well as script, form and applet (processing element). It is not always possible or practical to test each of these characteristic individually.

4. **The architecture is constructed and integration tests are conducted**.

The strategy for integration testing depends upon the architecture that has been chosen for the web application.

5**. The assembled web application is tested for overall functionality and content delivery**.

Like conventional validation, the validation of web based systems and application focuses on user visible action and user recognizable output from the system. To assist in the derivation of validation tests the tester should draw upon use cases the use cases provides a scenario that has high likelihood of uncovering errors in user interaction requirement

.

6. **The web application is implemented in a variety of different environmental configuration and is tested for compatibility with each configuration.**

7. **The web application is tested by controlled and monitored population of end user.**

Finally with the modular concept inside the application it is being also tested for its Reliability:

The system reliability will be insured through data integrity rules built into the database at the backend and the system rules built into the front-end application. The system will take assurance from the user before making any changes permanent.

**MAINTAINABILITY:**

The system has been designed taking care of modularity. Faults in the system can be traced to modules.

**VALIDATION CHECKS**

This will be as such to maintain consistent and persistent information on the web when most of the time the project has to deal with uploads so a minor error will down the impression of the company.

**Scope for Future Enhancement**

As changes are always necessary in future it applies to software development also but these changes should be appreciable in nature. These appreciable changes will make the software to fight for its survival in the competitive market. Hence it is necessary to think about the future enhancements at present.

The system ‘MUSIC STORE MANAGEMENT’ will fulfill the entire requirement of the clients. The system is developed according to the present requirements of the company. The system is developed as easy as possible for the sake of end users.

One drawback of my system is that the client cannot view, search and purchase music according to a particular language option .By the next time I would like to add this facility.

By the next time I would like to add two more modules: Purchase Module and Accounting Module. Purchase Module deals with purchasing activities of music related items. Accounting Module deals with all accounting activities such as billing, ledger preparation, balance sheet preparation, profit and loss account preparation etc.

In the present system transaction is through a particular bank or through money orders. In future I would like to make it through credit cards. Credit card validation techniques are needed for that.

The developed software for the organization is flexible and it can be made to run on all kinds of platforms. The system is error free and highly portable. It can be implemented in any servers in the Internet providing an easy access to the clients. It also has more options of the future developments.

**LIMITATION OF THE PROJECT**

* Can’t cover the financial data of the company.
* Can’t cover the windows interface with this project.

**Bibliography**

**Serial No. Title**

**1. Servlet Programming 2. Java Server Programming**

**3. Java Server Pages**

**4. Oracle8i**

**A. Online Resources**

* + - **http://www.google.com**
    - **http://oracle.com**
    - **http://Redbooks.IBM.com**
    - **http://javaguru.com**
    - **http://www.sun.com**