

IMPLEMENTATION

- 5.1 Introduction
 - 5.2 Input / Output Design Implementation
 - 5.3 Database Implementation
 - 5.4 Code Module
 - 5.5 System Implementation
-

5.1 Introduction

This will discuss about implantation of the MLL system. Implementation of code module and database are the most critically concerned sections of this chapter.

5.2 Input / Output Design Implementation

The screen output design is an important design task in the development of the human computer interfaces. The success or failure of a system development effort rests on this element of the design. Poorly designed HCI lead to user frustration and disinterest in using the system, which leads to the system being abandoned shortly after it is deployed. The output design should be flexible to provide for the adaptation of differing user requirements for display of data. MLL system contains less input from subscriber but more output screens.

The following were taken into consideration when designing the screen outputs

- What are the contents of the screen output?
- The navigation of a page
- How to present information

Sample screen shots are provided in Annex - D

5.3 Database Implementation

The ERD developed in the structured analysis in Chapter 3 is used as a guide in designing the database structure using SQL Server 2000. This activity is consisted of identifying the tables the fields and the data types of the fields.

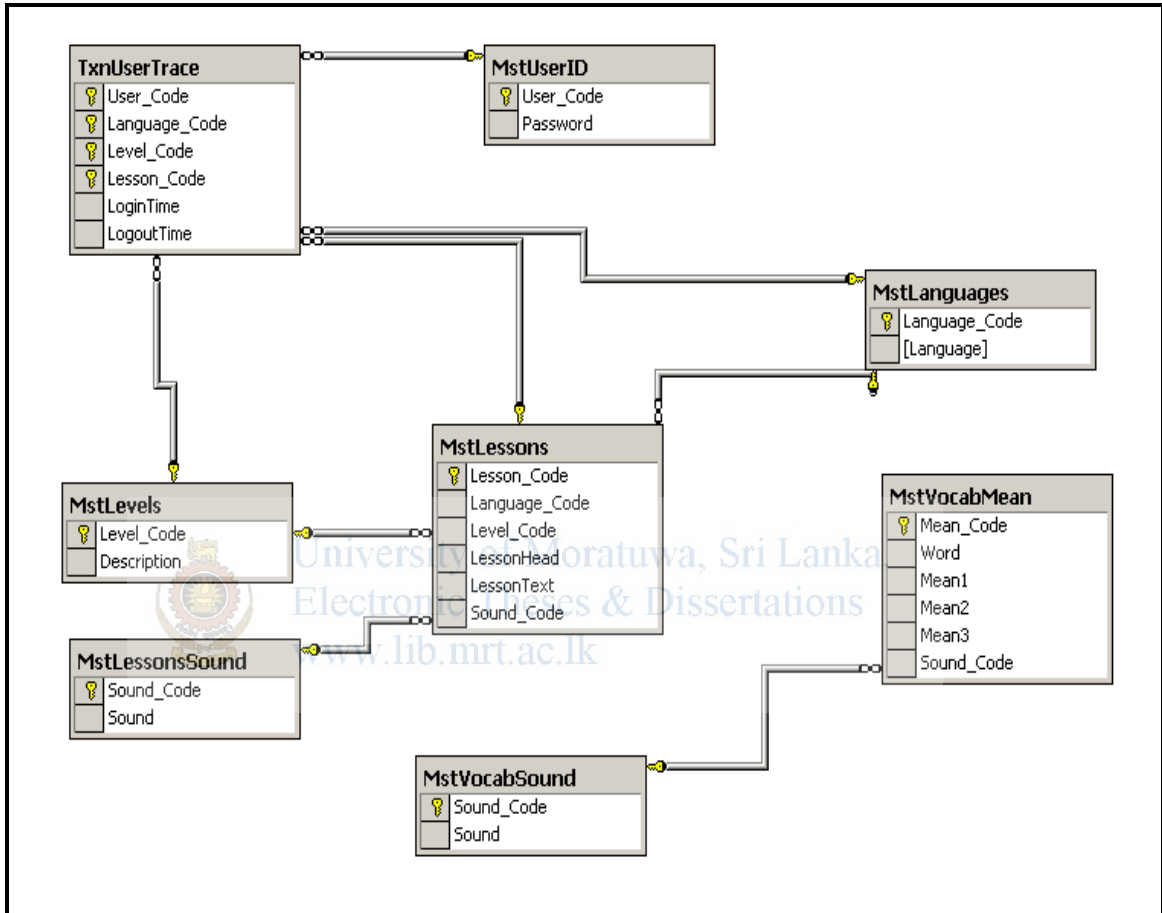


Figure 5.1– Database Implementation

Table Structures

Used table structures are given with descriptions.

<p>1. MstLanguages</p> <p>MstLanguages will hold the language which users have to choose.</p>
--

MstLanguages	
PK	Language Code
	Language

- Language_Code – Unique id for a language
- Language – The description of language

Sample data

Language_Code	0001	0002	0003
Language	English	French	German

Annex – C contain the rest definitions in the MLL Database

When working with a system that has a large number of users, speed of a query is a crucial factor to consider. Although speed of a query depends on various factors, creating proper indexes for the querying relation base on the query can play a big role to improve the speed of the query response time. When considering the above fact, so many queries are performed based on the user code, index have been created as shown in Figure 5.2

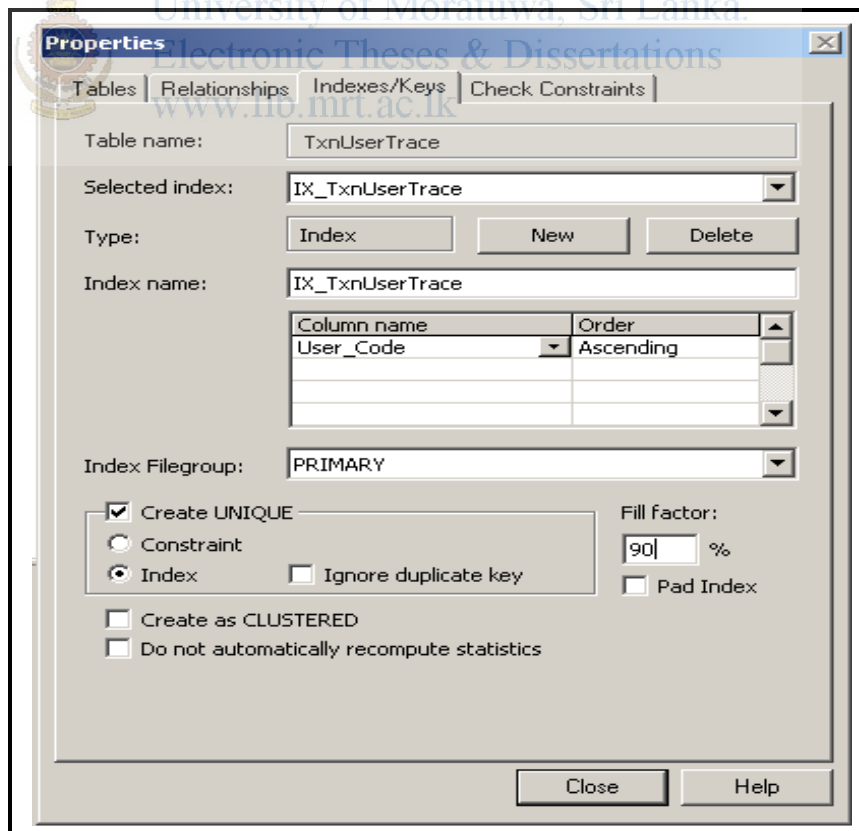


Figure 5.2– Manage Index and Properties

To access data first, it is needed to create a *SqlCommand* object and set its properties such as command text; parameters etc. In addition, assign it to the appropriate command properties in the *SqlDataAdapter* object. Then using that adapter it is able to execute the commands and get the information. Here the data adapter handles the resource handling part such as connection pooling; dispose the connection. so we do not need to destroy the connection created for the query, the adaptor does it on behalf of us.

5.4 Code Module

The system design used in the MLL is a layered model, which is the client server three-tier system architecture of a browser front-end, a web-server in the middle tier and a databases server at the backend shown in Figure 5.2.

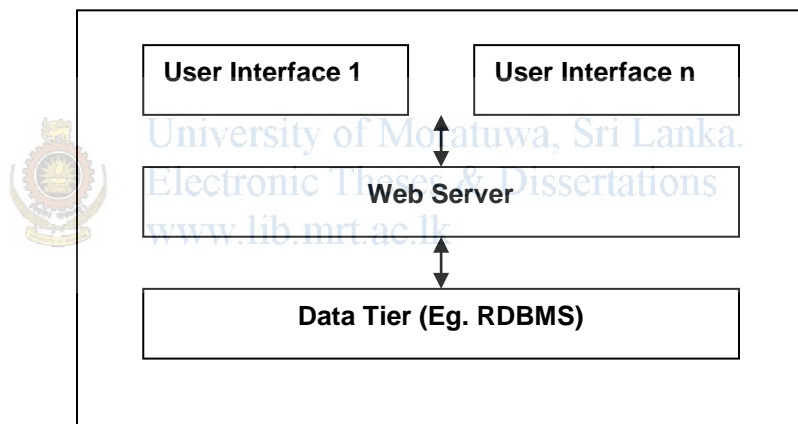


Figure 5.3– Coding Structure

In the three-tier client server, architecture the user, using a web browser located at the client end requests web pages from device. The web server takes these requests and passes them on to the database as SQL queries. The SQL server processes the queries and sends the results set back to the web server. This data is processed as specified in the script and the results served back to the user as WML/HTML pages for display depending on the client's web browser in the device.

The ASPX script is executed in the web server is called server side scripting and is not visible to the client. This protects the script from tampering by users and protects

the processing functionality of the application. Sometimes it is not advisable to perform all data validations on the server as they tie up the web server system resources leading to a drop in system performance. To overcome this problem there should be a client side script, which is executed by the browser-scripting engine. The MLL application makes use of server side script for its processing functionality.

In view of the interactivity that is required in a web-enabled application such as the MLL it is required to modularize the code and distribute it among several interconnected web pages. This results in having to address the problem of maintaining state between web pages. This is done by the use of session variables (like global variables) and local variables, which are populated by transferring data required.

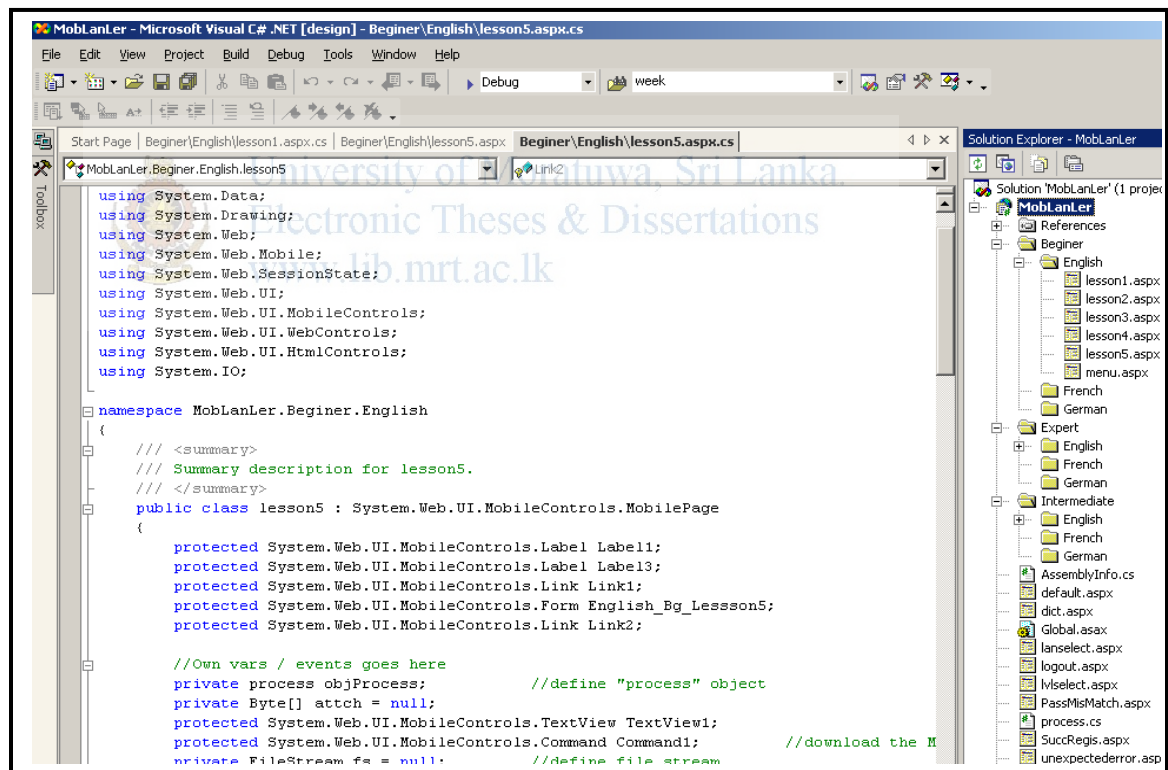


Figure 5.4– Coding Environment

Handling sessions is very important part in any form of a web application. It has to follow two steps, define the session variable at the *Session_Start()* event in the *Global.aspx* page and assign a value to it when submit the request at the web form.

When the user clicks the Submit button and if it is a valid request, and if all the information is correctly entered, request is processed by the system. If any error occurred while submitting, it is captured using a try-catch block and the error information passes to the user.

Sample of application code provided in Annex – C

5.5 System Implementation

After completed the coding part, system was deployed on a web server. Sample screen shots are shown in Figure 5.5



Figure 5.5– Working Application

Chapter Summary

Chapter – Five discussed implementation of the system. Coding structure and database implementation were concerned in this chapter. MLL is a three-tire application and coding compiled .aspx files will reside in web server and database is implemented on SQL server 2000 machine with win2K OS. Next chapter discusses about testing of the MLL system.

SYSTEM TESTING

- 6.1 Introduction
 - 6.2 Methodology
 - 6.3 Results
-

6.1 Introduction

Testing of the system is carried out as a parallel activity together with system development and implementation. Each web page formed a module in the code model of this application. Therefore, each page, which is written in C# code, had to be tested in isolation. When the pages are strung together integration. Testing was also carried out to ensure that the outputs of one web page is compatible and consistent with the inputs required by another related web page written in C# code. Therefore, module testing and integration are carried out during system testing of the MLL.

A test plan is designed by defining the input and the output required and comparing it with the actual output. This approach is also known as the Black Box Testing technique.

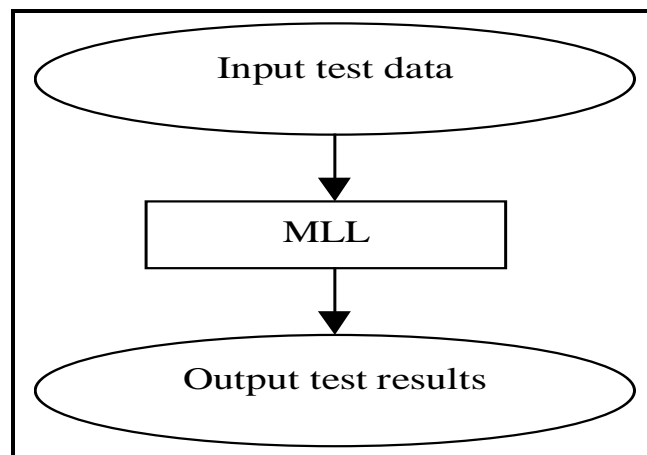


Figure 6.1– Black Box Testing

6.2 Methodology

The testing (i.e. test plan) of entire system from the software developer perspective, can be broken down to unit, module and system testing and therefore the test plan for the entire MLL can be broken down to the following components:

- Testing of the Process
 - Subscriber login
 - Navigation
 - Click events
 - Sound file download
- Code Testing
 - According to the guide line documentation
 - Database objects
- System Test
 - According to the SRS / Administration Part

Test Plan

The test strategy consists of a series of different tests that will fully exercise the MLL. The primary purpose of these tests is to uncover the systems limitations and measure its full capabilities. A list of the various planned tests and a brief explanation of them follows below.

System Test

The system tests were focused on the behaviour of the system. User scenarios were executed against the system as well as screen mapping and error message testing. Overall, the system tests test the integrated system and verify that it meets the requirements defined in the requirements document.

Performance Test

Performance test was conducted to ensure that the system's response times meet the user expectations and does not exceed the specified performance criteria.

During these tests, response times were measured under heavy stress and/or volume.

Security Test

Security tests determined how secure the MLL is. The tests verified that an unauthorized user access to confidential data is prevented. Allowing different users to log in and verify the available user options with the user rights list did this.

Following format is used to measure the system output in Black Box testing.

Test Case No	01			
FORM	Login Form			
FUNCTIONAL SPECIFICATION:	User Authentication			
TEST OBJECTIVE	To check whether the entered User name and Password are valid or Invalid			
TEST DATA	USER Name: PASSWORD:			
Step No	Steps	Data	Expected Results	Actual Results
1				
2				
3				
4				
5				

Table 6.1– Test Document Format

User Acceptance Test

Once the MLL is ready for implementation, subscribers can logon to the system and browse lessons.

6.3 Results

Actual Test Results

Test Case No	01			
FORM	Login Form			
FUNCTIONAL SPECIFICATION:	User Authentication			
TEST OBJECTIVE	To check whether the entered User name and Password are valid or Invalid			
TEST DATA	USER Name: KG0045 PASSWORD: q12			
Step No	Steps	Data	Expected Results	Actual Results
1	Enter User Name and press LOGIN Button	User Name= A001	Should Display Message "User not recognize"	Display Message "User not recognize"
2	Enter Password and press LOGIN Button	Password= test	Should Display Message "User not recognize"	Display Message "User not recognize"
3	Enter user Name and Password and press LOGIN Button	USER = A001 AND Password = test1	Should Display Message "User not recognize"	Display Message "User not recognize"
4	Enter user Name and Password and press LOGIN Button	USER = XYX AND Password = COES	Should Display Warning Message Box "Invalid User name or Password"	Display Warning Message Box "Invalid User name or Password"
5	Enter user Name and Password and press LOGIN Button	USER = XYZ AND Password = XYZ	Should Display Warning Message Box "Invalid User name or Password"	Display Warning Message Box "Invalid User name or Password"
6	Enter user Name and Password and press LOGIN Button	USER = " " AND Password = " "	Should Display Warning Message Box "Enter User ID and Password"	Display Warning Message Box "Enter User ID and Password"
7	Enter User Name and Password and press LOGIN Button	USER = A001 AND Password =test	Should navigate to lanselect.aspx page.	Navigate to lanselect.aspx page.

Test Case No	02			
FORM	Language Select Form			
FUNCTIONAL SPECIFICATION:	User Can Select Option			

TEST OBJECTIVE	To check whether the User can select desired language			
TEST DATA	Select any Option			
Step No	Steps	Data	Expected Results	Actual Results
1	Press the Go button without selecting a option	Default selection will be English	Should Display lvlselect.aspx for English	Display lvlselect.aspx for English
2	Press the GO button After selecting a language option	Select a option	Should Display lvlselect.aspx for selected language	Display lvlselect.aspx for selected language

Test Case No	03			
FORM	Manu Form			
FUNCTIONAL SPECIFICATION:	User Selection of the desired lesson			
TEST OBJECTIVE	To check whether the User can go forward by clicking the menu items			
TEST DATA	Click on menu items			
Step No	Steps	Data	Expected Results	Actual Results
1	Select option "Lesson One"	Select option	Should Display the Lesson One with Download option for sound file and Move Next Lesson option	Display the Lesson1.aspx file with options
1	Select option "Language Menu"	Select option	Move backward to the Language selection page (lanselect.aspx)	Move backward to the lanselect.aspx
2	Select option "Back"	Select option	Should navigate to the lvlselect.aspx for selection for level	Move backward to the lvlselect.aspx.

Chapter Summary

Chapter – Six discussed testing following functions of the system. Display of information, security features, error messages and admin functions.