**3.2 Architectural Design**

**3.2.1 Data Flow Diagram**

A data flow diagram is a graphical technique that depicts data flow and transforms that are applied as data move from input to output. The DFD is used to represent increasing information flow and functional details. A Level 0 DFD also called a fundamental system model or context model represents the entire software elements as a single bubble with input and output indicated by incoming and outgoing arrows respectively. Additional process and information flow parts are represented in next level i.e., Level 1 DFD. Each of the processes represented at level 1 are sub functions of overall system depicted in the context model.

**Data flow diagram symbols:**

Source/Destination of Data

Data flow

Process

Storage

**3.3 Interface Design**

An interface design elements for the software tell how information flows into and out of the system and how it is communicated among the components as part of the architecture.

**3.3.1 Input Design**

Input design is the link between the information system and actors and those steps that are necessary to put transaction data into a usable form for processing data entry. Instructing the computer to read data from a written printed document can active the activity of putting data into the computer for processing or it can occur by keying data directly into the system. The design of input focusing on controlling the errors, avoid delay, and keeping the process simple. System analyst decides the following input design details

What data to input?

What medium to use?

How the data is arranged and coded?

In my project named faculty assessment corner & student counsel suite, I tried to include the following design constrains provided in the software engineering.

*1: Avoid scattering of fields in the forms*

In all forms of the software the textboxes (which provided to input some data), label

(which label the text boxes), combo box (list a set of values) etc all are arranged in a neat and well format. It provides a simple look to the pages. The buttons are placed at the bottom of the page and easily accessible to the user. The menu are arranged below the heading and at a minimum level of menus are arranged with pages. Menu provides the continuity to the pages.

*2: User only needs to enter a minimum amount of data*

All forms contain a minimum amount data, but most essentials. No page provides or wanted bulky of data. It provides more easiness to the user. It creates more the software to the end user. Also the operation continues by single click.

*3: Avoid confusion in the forms*

All forms have a well-defined menus and each menu name indicate its purpose. So the user can easily access various forms without confusion. Each form and its sub forms are well labeled . So the user can easily identify the forms and work on that.

**3.3.2 Output Design**

Designing computer should proceed in well thought out manner

The term output means any information produced by the information system weather printed or displayed. Output design is a process that involves designing necessary output that have to be used by various users according to requirement. The efficient intelligent output design should remove the system relationship with the users and help in decision making.

When designing the output, system analyst must accomplish the following:

Determine the information present

Decide whether to print, display the information and select output medium

Arrange information in acceptable format.

In my project, the outputs are in the form of reports. They are well format and it provides the output in a correct and neat format.

**3.4 Procedural Design**

The system has two modules. They are…

* Admin module
* Exhibitor module
* Company module
* User module

**1. Admin module**

Admin is the one who controls the organization. He can add,remove,approve

Exhibitor,company.

Following are the functions that can be performed by the admin

* Add landmark
* Approve expo
* Approve company
* Chat with companies and exhibitors
* Add notifications
* Add category and subcategory

**Add landmark**

Add landmark for conducting expo.lt means country ,district,state,place.

**Approve expo**

Exhibitor added expo will be approved by theadmin**.**

**Approve company**

Approve companies for the expo.

**Chat with companies and exhibitors**

Chat with companies and exhibitors and give instructions for expo.

**Add notifications**

Give necessary updates for the expo.

**Add category and subcategory**

Add category and subcategory for the expo.

**2. Exhibitor module**

Exhibitor is the one who conduct the expo. He is the overall control of the expo.

Following are the functions that can be performed by the exhibitor

* Register and login
* Add expos
* Invite companies
* Booth allocation
* Payment with companies
* Confirm ticket booking
* Chat with admin
* Add notifications

**Register and login**

Register new exhibitor and login to the expo.

**Add expos**

Add various expo for the exhibition.

**Invite companies**

Invite companies for the expo.

**Booth allocation**

Allocate admin accepted landmarks to the company.

**Payment with companies**

Receive payment from the company.

**Confirm ticket booking**

Confirm the tickets that are booked by the user.

**Chat with admin**

Chat with admin about the companies and expo.

**Add notifications**

Add instructions to the user and companies.

**3. company module**

Company is one who participating in the expo and one who sales their product in the expo.

* Following are the functions that can be performed by the company
* Registration and login
* View expo details
* View booth details and reserve for booth
* Payment
* View user booking and reserve the items
* Add details about the products

**Registration and login**

Register companies for the expo and login.

**View expo details**

View the expo details. Which type expo is done. And the procedures for the expo.

**View booth details and reserve for booth**

View the allocations for the expo and reserve them for the expo.

**Payment**

Payment for the expo.

**View user booking and reserve the items**

View user bookings and reserve the items the user will be purchased.

**Add details about the products**

Add detail description about the products that the company was launched.

**4. user module**

User is the one who purchase product from the expo.

Following are the functions that can be performed by the user.

* Register and login
* View expos
* Select and book items
* View selected item,cancellation,payament
* Ticket booking and payment
* Download ticket
* Chat with exhibitor

**Register and login**

Register and login to the expo.

**View expos**

View the available expos.

**Select and book items**

Select the needed items from the company and book them.

**View selected item,cancellation,payament**

View the ordered items . if any cancellation is needed that will be done. Done the payments for the products **.**

**Ticket booking and payment**

Ticket will be booked for the expo and done its payment.

**Download ticket**

Download the tickets for the expo.

**Chat with exhibitor**

Chat with the exhibitor about the expo.

**4. CODING**

**4.1. Description of Software Used**

**Microsoft.net**

.NET is an integral part of many applications running on Windows and provides common functionality for those applications to run.Microsoft.NET is prefabricated infrastructure for solving common problems in internet applications. Microsoft.NET has been getting enormous amount of publicity lately, even for this industry Microsoft. NET is an add-on-run-time environment that runs on windows 2000 operating system. Later versions of .NET will probably be made part of the operating system, the U.S. department of justice willing.

**Microsoft.net framework**

The .NET Framework includes the runtime and compiles time services required to run a .NET application. Compile time is when the developer is compiling the source. Runtime is when the compiled code is executing in memory. At the center of the runtime execution of .NET code is the Common Language Runtime (CLR). The CLR is a virtual machine that runs as a process on the computer on which it is installed

**COMMON LANGUAGE RUNTIME**

The Common Language Runtime (CLR) serves as the execution environment for the .NET Framework. The CLR is responsible for managing the compiled code of .NET applications, which can be written in different languages including VB, C#, Java, and Perl. The cross-language integration is achieved through the two major components of CLR: Intermediate Language and Metadata.

Intermediate Language (IL) is an assembly language that runs on almost any type of CPU. IL achieves this versatility by using stacks to handle

functions that normally occur in registers. As managed code, IL is just-in-time (JIT) compiled when .NET applications are executed. JIT compilers convert IL into machine language that is specific to the host CPU. [During runtime, JIT compilers have the luxury of choosing stacks, registers or other stores to implement IL stacks.] Various JIT compilers are provided by the CLR, making it possible for different computer architectures to execute IL. Unlike other assembly languages, IL integrates high-level concepts which make CLR code more robust. As a high level language, IL is strongly typed and uses the ideas behind structured-exception handling, deployment support, and component interaction. Thus a range of software can run on the .NET Framework as long as the compiler can produce IL. Metadata, the second component of the CLR, is a description of the implemented code. The Metadata is responsible for ensuring that the CLR executes the code securely. To prevent modules of software from breaking type definitions, Metadata stores information regarding classes, methods, and types. Registers are no longer required to keep track of information because relevant data is stored with the compiled code or IL. By housing information on classes and registrations, Metadata allows the CLR to function more efficiently since programs are less likely to get hung up on version and inheritance dependencies.

**Language(c#)**

Microsoft Visual C# is a powerful but simple language aimed primarily at developers creating Applications by using the Microsoft .NET Framework. It inherits many of the best features of C++ and Microsoft Visual Basic, but few of the inconsistencies and anachronisms, resulting in a cleaner and more logical language. The advent of C# 3.0 has seen several important new features added to the language, including Generics, Iterates, and anonymous methods. By design, C# is the programming language that most directly reflects the underlying CLI.

Most of its intrinsic types correspond to value-types implemented by the Common Language Infrastructure (CLI) framework. However, the language specification does not state the code generation requirements of the compiler: that is, it does not state that a C# compiler must target a Common Language Runtime, or generate Common Intermediate Language (CIL), or generate any other specific format. Theoretically, a C# compiler could generate machine code like traditional compilers of C++ or FORTRAN

**Features of c#**

• C# language is intended to be a simple, modern, general-purpose, object-oriented programming language.

• The language, and implementations thereof, should provide support for software engineering principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection. Software robustness, durability, and programmer productivity are important.

• The language is intended for use in developing software components

Suitable for deployment in distributed environments.

• Source code portability is very important, as is programmer portability, especially for those programmers already familiar with C and C++.

Support for internationalization is very important.

• C# is intended to be suitable for writing applications for both hosted and embedded systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions.

Although C# applications are intended to be economical with regard to memory and processing power requirements, the language was not intended

to compete directly on performance and size with C or assembly language.

**Microsoft SQL Server 2008**

Microsoft SQL Server 2008 provides the Microsoft Windows Server System integrated server software with a database platform for the next generation of connected, scalable, and reliable enterprise applications. The breadth and depth of innovation in this version is in response to t he needs of customers. This white paper is targeted to database administrators, to give you an understanding of the new features in and capabilities of SQL Server 2005. From many enhancement of existing features, to an entirely new security model, database administrative is now more productive and in tune with needs of the administrator.

At the core of SQL Server 2008 are new infrastructure application capabilities. SQL Service Broker is a distributed application framework that provides a new form of scalability and reliability for asynchronous message delivery. Though not new, Microsoft SQL server Notification Services, Reporting Services, and SQL Server Mobile Edition (formerly called SQL Server CE) are all greatly enhanced in SQL Server 2005.

SQL (Structured Query Language) is a database computer language designed for the retrieval and management of data in relational database scheme creation and modification, and database object access control management.

**4.2 Coding Principle**

The input to the coding phase is the design document. During coding phase, modules identified in the design document are coded according to the module specification. Objectives of coding phase are, to transform design into code and unit test the code.

**Coding Guidelines**

Code should be easy to understand.

Don’t take pride in cryptic code.

Code should be well documented.

Comments should be present.

Functions should be small.

Do not use Go-to statement.

Do not use the same variable for multiple purposes.

**5. SYSTEM TESTING**

Testing is a methodology for evaluating the project. The good test has a high probability of finding errors. Testing is generally two types.

***White box testing***

White-box testing is also called as glass-box testing, is a test case design method that goes to the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that

Guarantee that all independent paths within a module have been exercised at least once.

Exercise all logical decision on their true and false sides.

Execute all loops at their boundaries and within their operational sides.

Exercise internal data structure to ensure their validity

White box testing was successfully conducted on our system. All independent paths within a module have been executed at least once and all logical decisions have been exercised on their true and false sides.

***Black box testing***

Black-box testing is also called as behavioral testing, focuses on the functional requirement of the software. It is a complementary approach that is likely uncover a different class of errors than white box methods. Black box testing attempts to find errors in the following categories

Incorrect or missing functions.

Interface errors.

Error on data structures or external database access.

Behavior or performance errors.

Initialization and termination errors.

Black box testing was successfully conducted on our system. The system was divided into a number of modules and testing was conducted on each module. We have tested the system for incorrect or missing functions, interface and performance errors.

**TEST CASES**

**Login form**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test scenario** | **Expected result** | **Observed result** | **result** |
| 1. | Enter wrong user name and password | Display login form again with invalid message. | Message displayed. | Pass |
| 2. | Enter correct user name and wrong password | Display login form again with invalid message. | Message displayed. | Pass |
| 3. | Enter correct user name and correct password of admin or exhibitor or user or company for student. | Admin or user or company or exhibitor can login to the system. | Login to their pages. | Pass |
| 4. | Press login button without filling the user name and password. | Display warning message to fill the details. | Warning message displayed | Pass |

**Register exhibitor**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No.** |  |  |  |  | | **Test scenario** | **Expected result** | **Observed result** | **result** |
| 1 | Form displayed | Display form with all controls. | Form loaded with all controls. | Pass |
| 2 | Click register button without data | Display warning message to fill the details. | Warning message displayed | Pass |
| 3 | Click register button with data | Display message saved success fully and clear the fields | Message displayed and all fields cleared | Pass |
| 4 | Click cancel button | |  |  | | --- | --- | |  | Exit the form | | Exited | Pass |

**Register company**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No.** |  |  |  |  | | **Test scenario** | **Expected result** | **Observed result** | **Result** |
| 1 | Form displayed | Display form with all controls. | Form loaded with all controls. | Pass |
| 2 | Click register button without data | Display warning message to fill the details. | Warning message displayed | Pass |
| 3 | Click register button with data | Display message saved success fully and clear the fields | Message displayed and all fields cleared | Pass |
| 4 | Click cancel button | |  |  | | --- | --- | |  | Exit the form | | Exited | Pass |

**Register user**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No.** |  |  |  |  | | **Test scenario** | **Expected result** | **Observed result** | **result** |
| 1 | Form displayed | Display form with all controls. | Form loaded with all controls. | Pass |
| 2 | Click register button without data | Display warning message to fill the details. | Warning message displayed | Pass |
| 3 | Click register button with data | Display message saved success fully and clear the fields | Message displayed and all fields cleared | Pass |
| 4 | Click cancel button | |  |  | | --- | --- | |  | Exit the form | | Exited | Pass |

**6. SYSTEM IMPLEMENTATION**

Implementation is the process of deploying the new system in the operational environment. Proper implementation is essential to provide a reliable system to meet the organizational requirement. There are four types of implementation methods. They are Direct Changeover, Phased Implementation, Parallel Run and Pilot Approach. The most commonly used implementation methods are Pilot Approach and Parallel Run.

The implementation method I used to implement faculty assessment corner and student counsel suite is Parallel Run. i.e., the new system will work parallel to the existing system. The new system will replace the existing system completely after six months.

Faculty assessment corner and student counsel suite is developed as a desktop application, as usual some desktop development technologies are used in the implementation of the project. The language I selected to program this software is ASP.Net.The reason I selected ASP.Net is that is a simple and powerful language that especially developed to create desktop application.

Technologies used in the development of the software are:

* Programming language - ASP.Net
* DBMS - SQL Server 2008
* Development tool - Visual Studio 2010
* Development platform - Windows 8.1

The front end is ASP.Net and back end is SQL server 2008. The system developed on Visual studio 2012 development tool in Windows 7 operating system.

**7. CONCLUSION**

The project faculty assessment corner and student counsel suite is for computerizing the working of the college. The software takes care of all the requirement of College and is capable to provide easy and effective storage of information related to students and faculty details, feedback system and mentoring reports. This desktop application offers several features, which is very useful for the administrator, faculties and students. This project allows different actors of the system to work within the same environment. Using the facility and flexibility of ASP.Net and SQL server, the software has been developed in a neat and simple manner, thereby reducing the actor’s work.

Faculty assessment corner and student counsel suite providing following features to its users:

A user with minimum knowledge can be able to operate the system easily.

The software is developed with modular approach

All modules in the system have been tested with valid and invalid data.

Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

**FUTURE ENHANCEMENT**

As a future venture, it is suggested to make some changes to provide more services and to provide information at right time in right manner.

**They are:**

All the data cards for the data collections can be computerized.

This software can be converted to a web application and used by all college for their needs**.**

Using this software can easily find out the information related to students and faculties.