Implementation

Overview

In this section, we intend to define how we implemented our system. The methods needed to develop and to deploy the project such as the hardware and software requirements as well as the technologies used associated with the justification of use are illustrated. In addition, the data and its classification and organization is highlighted. Moreover, the main procedures are explained. Finally, the input forms as well as the output forms of the project are demonstrated with snapshots.

11.1 Implementation

11.1.1 Introduction

The first step to initiate new activity or achievement is the hardest. As soon as we started to develop out project, it seemed to be difficult to finish it. Nevertheless, going through the implementation associated with taking the hazards empowered by the ambition of fulfilment, the project is done and has implemented that tasks we intended to work on. In the following sections, we intend to demonstrate and evaluate the tasks done including the database, coding, and interfaces. Moreover, we aim to classify these artifacts according to the MCV model (Model Controller Viewer)

11.1.2 Hardware and Software Requirements

11.1.2.1 for the Server

As a web application, there are certain software packages to be installed specially on the server. Based on these software packages, the hardware minimum specifications are defined. For this particular reason, we will start to describe the software needed for the server, which acts as a database, web and virtual server if needed, followed by the hardware minimum specifications.

11.1.2.1.1 Software:

Table 1 Software Requirements for the Server

Type of Software	Software Package
Server Operating System	Windows Server 2003 R2 SP2 64-bit Itanium Enterprise or equivalent.
Server Database Management System	SQL Server 2008 R2 Enterprise (64-bit)
Server Web Hosting System	Visual Studio 2010 .Net Framework 3.0
Virtualization Software	Microsoft Virtual PC 2007 SP1 OR Windows Virtual PC OR Windows Server 2008 Hyper-V OR Foundation Server 2010 RTM VPC
Virtual Image	Microsoft Visual Studio 2010 and Team

11.1.2.1.2 Hardware:

Table 2 Hardware Requirments for the Server

Component	Requirement
Processor Type	Itanium processor or faster.
Processor speed	2.6 GHz or more.
Memory	RAM: minimum of 4 GB, maximum of 2 TB.
Hard Disk	50 GB free space on a single drive/ partition

11.1.2.2 for the Web Client

As a web application used by clients, the users' hardware specifications are varied. We will try to demonstrate the minimum requirements of hardware by anticipating the hardware of the client that we used during the development stage of the life cycle of the project.

11.1.2.2.1 Software:

Table 3 Software Requirements for the client

Type of Software	Software Package
Operating System	Microsoft Windows XP Professional SP3/ Vista/ Windows 7 (Operating System)
Web browing System	.Net Framework 3.0 Any Web browser

11.1.2.2.2 Hardware:

Table 4 Hardware Requirments for the Client

Component	Requirements
Processor	Intel(R) CPU 1.83 GHz or higher
Memory	1.00 GB.

11.1.3 Technologies and Techniques

Some development and design frameworks and environment are used to implement BSIS system. In the following sections, we present each tool used in the system and some of its advantages.

11.1.3.1 Programming framework (Microsoft Visual Studio2010)

Microsoft Visual Studio is a Microsoft product that offer an integrated development environment (IDE). It provides design, development and debugging and deployment services. It can produce managed code as well as native code. Visual Studio contains a code editor supplemented with IntelliSense as well as code refactoring and code snippets. All of these cabaibilites aim to make the coding stage easier. The integrated debugger implements a source-level debugger as well as machine-level debugger. Visual Studio supports different programming languages which are combatible with the .NET Framework. Such as C, C++, VB.NET , C#, Python and others.

11.1.3.2 Programming Languages (C#)

We have the following requirements. A programming language to be simple, general-purpose, object-oriented programming language, providing support for software engineering programming language principles such as strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection. These requirements help for Software robustness, durability, and increase programmer productivity. Moreover, we need a light weight program that uses economical memory and processing power.

For our project we choose C# programming language. C# is generally cover our project requirement.

11.1.3.3 SQL

SQL refer to Structured Query Language. It is a database language designed for managing data in Relational Database Management Systems (RDBMS). SQL used in BSIS to write data query, create and update schema, and to control data access.

11.1.3.4 Relational Database Management Systems (RDBMS) (Microsoft SQL server 2008)

Microsoft SQL Server is a relational database management system developed by Microsoft. As a Relational Database Management Systems, the primary function is to store and retrieve data as requested by other applications. It can be accessed by other applications on the same computer or via the network including the Internet. There are many editions of Microsoft SQL Server aimed at different audiences and different workloads, ranging from small single-machine applications to large web-based applications associated with many concurrent users. Its primary query languages are T-SQL and ANSI SQL.

11.1.4 Procedures

In our project, we followed MCV (Model Controller Viewer). The model is embodied in the database. The controller is the code that connects the viewer with the database. Finally, the viewer is the interfaces. The following part explains each one.

11.1.4.1 The Model

Our database consists of seven tables illustrated below.

The events table is to store data about the events made by the system.

FRF-PC\SQLEXPRE...Reg - dbo.Events

Column Name	Data Type	Allow Nulls
Event_Id	bigint	☐
Event_Name	nvarchar(2000)	☑
Event_Type_Id	nvarchar(2000)	☑
Event_Logo	nvarchar(2000)	☑
Event_Brief	nvarchar(MAX)	☑
Event_Open	nvarchar(50)	☑
Event_End	nvarchar(50)	☑
IsDelete	bit	☑
Event_Detailes	nvarchar(MAX)	☑
holder_Id	int	☑
IsOpen	bit	☑
PLace_Id	int	☑
		☐

Figure 1Events Table

Workshops table is used to store data about the multiple workshops that are related to a particular event.

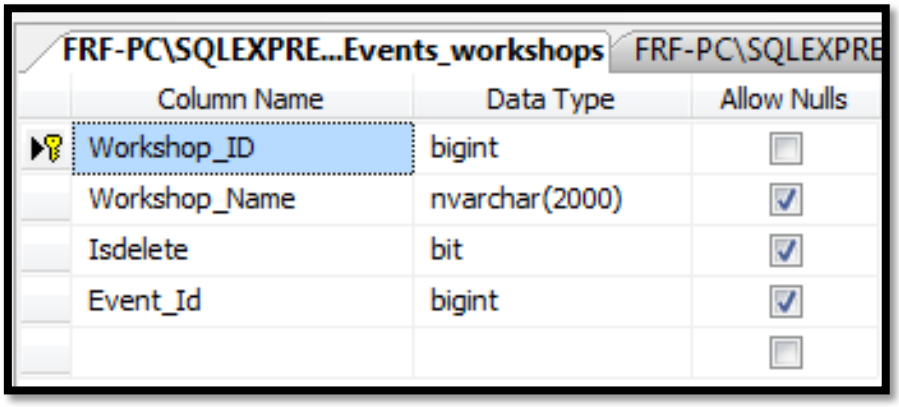

Figure 2 Event_WorkGroup Table

Holders table is to store the data about the users who make the events.

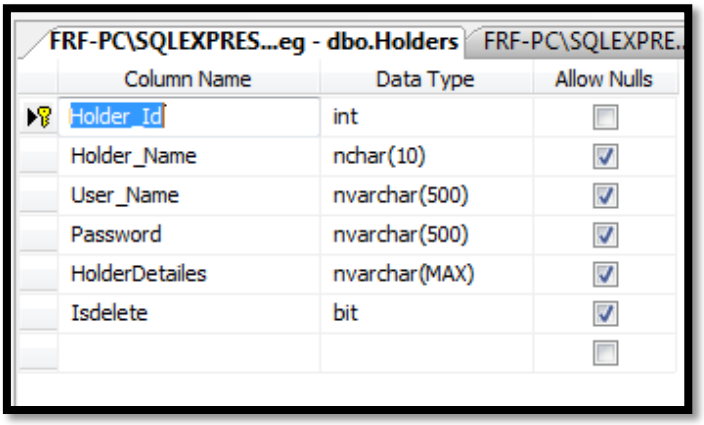

Figure 3 Holders Table

Place table is define the place of events and related workshops.

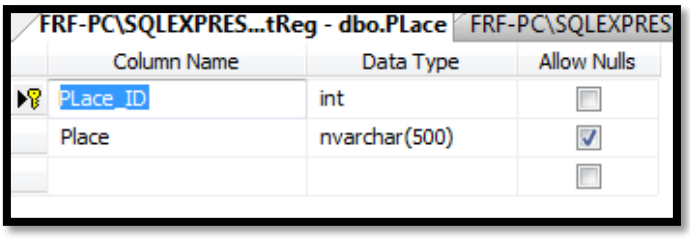

Figure 4 Place Table

Sponsors table is to store data about the sponsors of an event.

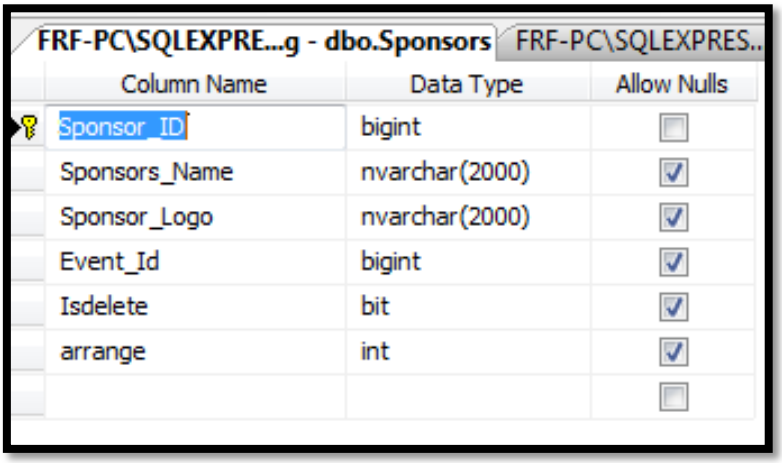

Figure 5 Sponsor Table

Permision table is to store data about the permission of the screens. It used mainly by the Controller.

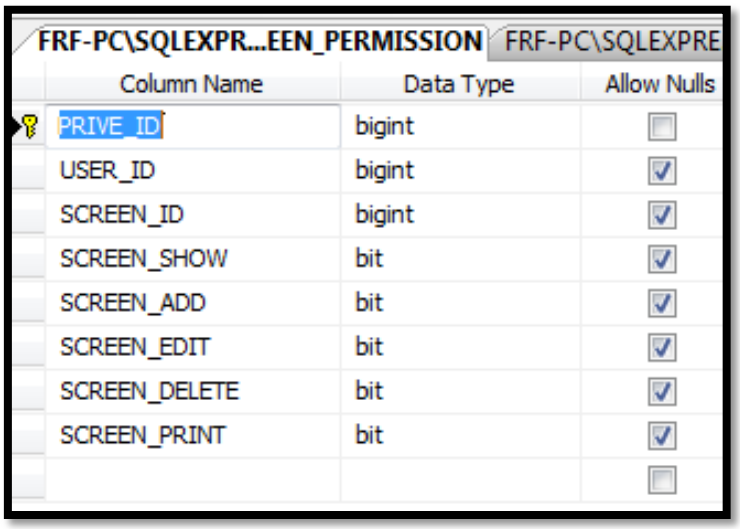

Figure 6 Screen_Permission Table

The user data stores information about the system users and their profile.

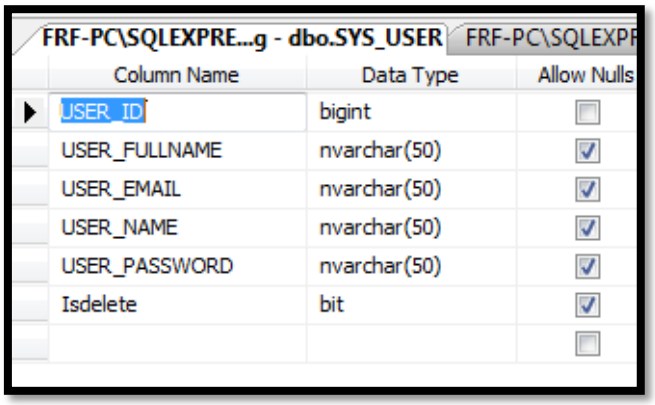

Figure 7 System User Table

The visitor workshop table is used to store data about non members who attend the events' workshops.

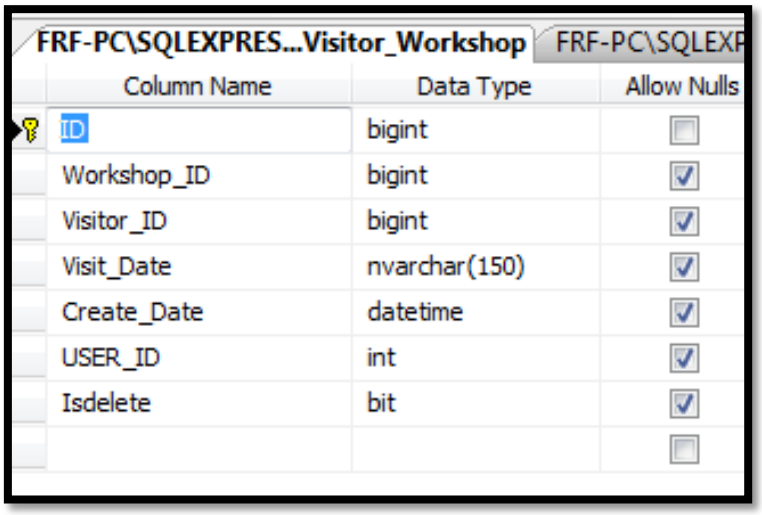

Figure 8 Visitor Workshop Table

Visitors table define the data of the visitors of the website who attend events.

FRF-PC\SQLEXPRES... dbo.VR_Visitors | FRF-PC\SQLEXPR

Column Name	Data Type	Allow Nulls
Visitor_ID	bigint	☐
Visitor_Name	nvarchar(500)	☑
Visitor_Gender	int	☑
Nick_Name	nvarchar(500)	☑
Visitor_Age	int	☑
Country_Name	nvarchar(500)	☑
Side_Name	nvarchar(500)	☑
Job_Name	nvarchar(500)	☑
Visitor_Mobile	nvarchar(50)	☑
Visitor_Phone	nvarchar(50)	☑
Visitor_Fax	nvarchar(50)	☑
Visitor_POBox	nvarchar(50)	☑
Visitor_ZipCode	nvarchar(50)	☑
Visitor_City	nvarchar(50)	☑
Visitor_Country	nvarchar(50)	☑

Figure 9 Visitors Table

Screen table stored information about the screens. The controller mainly uses it.

FRF-PC\SQLEXPRE...- dbo.Sys_SCREEN | FRF-PC\SQLEXPRE...g

Column Name	Data Type	Allow Nulls
SCREEN_ID	bigint	☐
SCREEN_NAME	nvarchar(50)	☑
IsDelete	bit	☑
		☐

Figure 10 Screen Table

11.1.4.2 The Controller

The controller code is responsible to connect the database with the interface according the business requirements of the system. It is a C# class. It is illustrated below.

11.1.4.2.1 ConnectionString:

To create a connection with the database.

```
Public Shared ConnectionString As String =
WebConfigurationManager.ConnectionStrings("EventRegConnectionS
tring").ConnectionString
```

11.1.4.2.2 Quizconection function:

To begin a transaction after opening a connection with the database.

```
Public Quizconection As New SqlConnection(ConnectionString)
```

11.1.4.2.3 comitttrans function:

To commit a transaction after opening a connection with the database.

```
Sub comitttrans()
```

11.1.4.2.3 rollbacktrans function:

To rollback a transaction and close the connection after opening a connection with the database.

```
Sub rollbacktrans()
```

11.1.4.2.3 Bind function:

To store the data in a datareader and load the data to the appropriate controls.

```
Sub Bind(ByVal listControl As ListControl, ByVal tableIndex As String, Optional ByVal header As String = "0")
```

11.1.4.3 The Viewer

The viewer is the interface of the project which are illustrated in the following section.

11.2 I/O Screens

In this section, we will show the snapshot of main input & output screens.

First the user visitors our website on

http://c.com.crowfoot.arvixe.com/interface/Default.aspx

The main screen appears. It contains the main project design. In the upper part of the page there is the project logo in Arabic on the right hand side and English on the left hand side. Below it, the main menu bar which contain the main links to guide the user to navigate to his used functions. Other services offered are handy icon to share the event on social network and printing tool. These parts will be available in each page for ease of use

On the main screen, there is a welcoming message that identifies the aim of the program as well as its main functions. AT the the bottom, the last 10 updated events are located.

Figure 11Main screen of the System

To start using the system, the user have to login. The user enters the username and password. Errors such as invalid username and/or password are caught.

Figure 12 Login Screen for the System

For new users, they have to register in the system. It is a simple registration form that collects information about the name, nationality, the user name he/she want to use, city, phone no, mail, gender, year of birth, cell phone no. and password.

Figure 13 New User Registration Screen

After login, the users' control panel appears. It guides the user to fulfill the needed functions

Figure 14 User control panel Screen

The event inquiry screen displays the available events.

Figure 15 Inquiry events Screen

When the user click on any event from the previous screen the show events details screen is displayed. It contains every piece of information available on the event with availability to share the event on the social network.

Figure 16 Events Details Screen

The workshop screen shows the workshops associated with the selected event.

Figure 17 Workshops Screen

Organizers screens shows information about the sponsors of the selected event.

Figure 18 Organizers Screen

New organizer screen enables to add information about the organizer of events. It include information such as organizer name, username, password to be entered

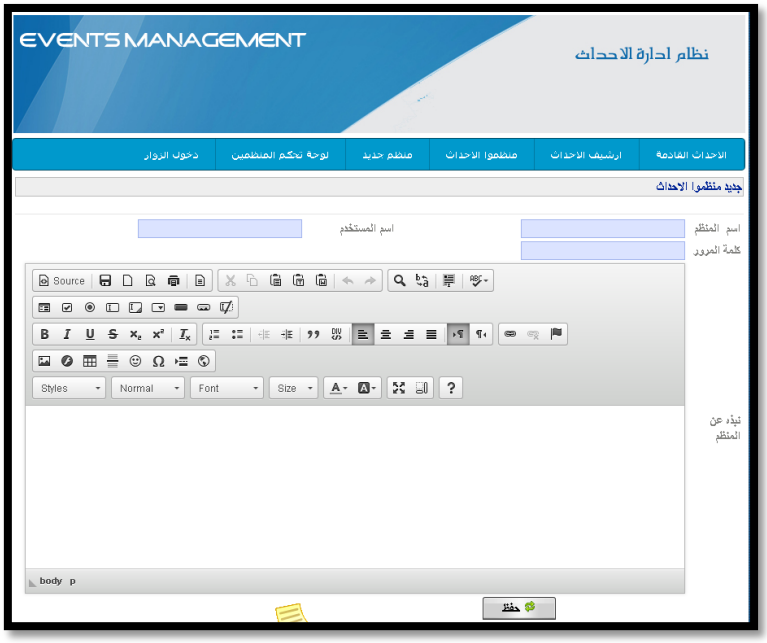

Figure 19 New organizer Screen

For the organizers, they have to log in and define that they are organizers by choosing the checkbox that illustrated this feature.

Figure 20 Organizers Login Screen

After login, the next screen shoot shows the organizer control panel. It leads the organizer to his/her implement the required functions for example to create events or update an existing one or to check for archived one.

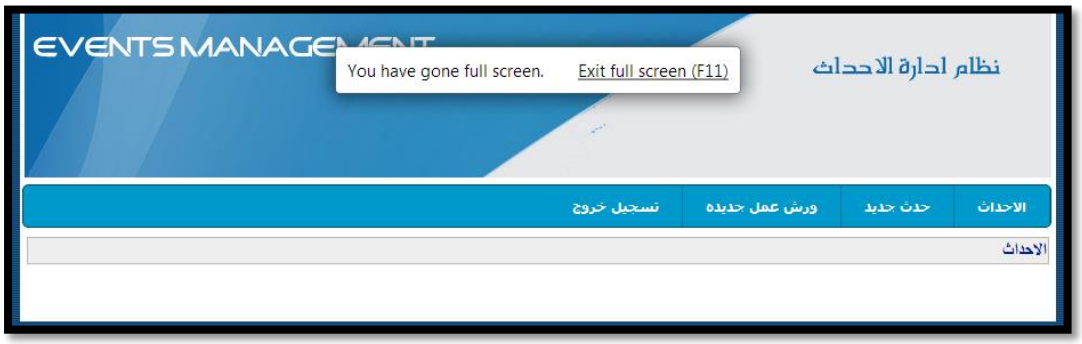

Figure 21 Organizers Control Panel Screen

Organizers' event screen show the events created by the organizer.

Figure 22 Organizer's Events Screen

New organizer after creating an event can add a workshop to it

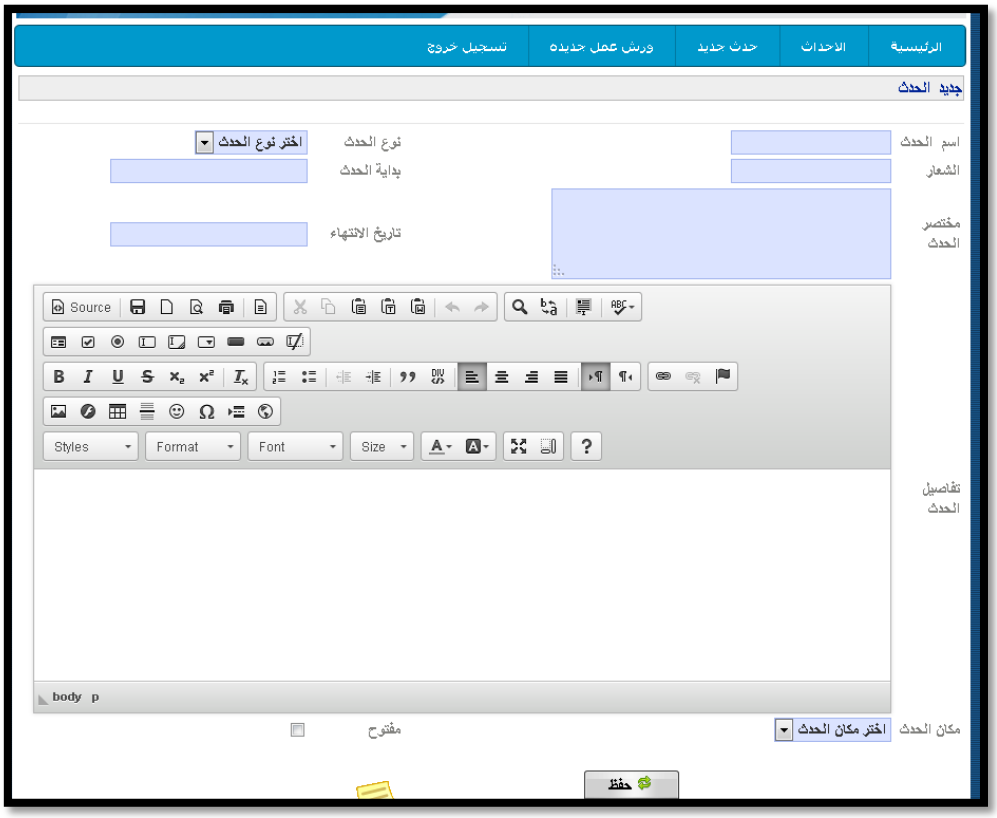

Figure 23 Organizer Add Workshop Event Screen

Then the organizer can associate the workshop with a previously made event.

Figure 24 Organizer Associate Workshop and Event

Conclusion

In this sections, we demonstrated the implementation of our project. First, we illustrated the hardware and software requirements for the server and the client. Then we introduced the technologies used. After that, we illustrated the model, which is the database implementation. Next, we introduced the controller code to connect the database with the screens according the business functions of the system. Finally, we demonstrated the main screens of input and output of our project.