

WEBSITE USING SILVERLIGHT AND ASP.NET MODULES

by

ARUN P. SHASTRY

B. E., Visveshwaraiah Technological University, Belgaum, Karnataka, India, 2004

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2009

Approved by:

Major Professor
Dr. Daniel Andresen

Abstract

Developing websites that provide the user with a rich experience has become one of the most important needs of the present day world wide web. This website has been implemented using Microsoft Silverlight 2.0 - one of the front runners among technologies used for creating applications with a rich user experience and ASP.NET.

The main objective of this project is to build parts of a website using Microsoft Silverlight – thus, showcasing the technology’s capabilities in improving user experiences on websites and also explore how these experiences are made possible from a programmer’s point of view. A part of the functionality of the website is also implemented as ASP.NET modules. The background code for both the parts is written in C#. NET.

Table of Contents

List of Figures	v
Acknowledgements	vii
CHAPTER 1 - Introduction	1
1.1 Aim	1
1.2 Scope.....	1
1.3 Need of the Application.....	2
1.4 Related Work and Problems	2
1.5 Microsoft Silverlight.....	2
1.5.1 Overview	2
1.5.2 XAML.....	3
1.5.3 Silverlight 2.0.....	4
1.6 Platform Specifications - Development.....	5
1.6.1 Hardware Specification.....	5
1.6.2 Software Specification	5
CHAPTER 2 - System Requirement Analysis.....	6
2.1 Information Gathering	6
2.2 System Feasibility.....	6
2.1.1 Economic Feasibility	6
2.1.2 Technical Feasibility	6
2.1.3 Behavioral Feasibility	7
CHAPTER 3 - System Analysis	8
3.1 Database Schema Diagram	8
3.2 Data Flow Diagram.....	9
3.3 State Transition Diagram	9
CHAPTER 4 - Design.....	11
4.1 Design Goals.....	11
4.2 Architectural Design	11
4.2.1 Architectural Context Diagram.....	12

4.2.2 Control Hierarchy	13
4.3 Procedural/Modular Approach	14
4.3.1 Home page Silverlight application.....	14
4.3.2 About page Silverlight application	15
4.3.3 Video page Silverlight application.....	15
4.3.4 Membership Registration Module	15
4.3.5 Program Reservation Module	16
CHAPTER 5 - Implementation.....	17
5.1 Database Implementation	17
5.2 User Interface Design and Implementation	17
5.2.1 Home menu application	18
5.2.2 About Us application	22
5.2.3 Video Player application.....	25
5.2.4 Membership Registration.....	27
5.2.5 Program Reservation.....	30
CHAPTER 6 - Testing.....	36
6.1 Unit Testing	36
6.2 Integration Testing.....	37
6.3 Validation Testing.....	38
6.4 White Box Testing.....	38
6.5 Performance Testing.....	39
6.5.1 Testing with JMeter	39
6.5.2 Test Results	40
6.5.3 Test Result Analysis	43
CHAPTER 7 - Results	44
CHAPTER 8 - Conclusion.....	45
8.1 Limitations	45
8.2 Future Work.....	45
References.....	46

List of Figures

Figure 1.1 Silverlight 2.0 Architecture	4
Figure 3.1 Database Schema Diagram.....	8
Figure 3.2 Context Level Data Flow Diagram.....	9
Figure 3.2 State Transition Diagram.....	10
Figure 3.2 Architectural Context Diagram	12
Figure 4.2 Control Hierarchy	13
Figure 5.1 Database Implementation using MS SQL Server 2008.....	17
Figure 5.2 Home page hosting the Silverlight application with buttons for navigation	19
Figure 5.4 Contacts user control	21
Figure 5.5 About page hosting the Silverlight application	22
Figure 5.6 History page of the Silverlight application.....	23
Figure 5.7 Indian Scene page of the Silverlight application.....	24
Figure 5.8 Silverlight Media Player.....	26
Figure 5.9 Membership Registration with Required Field Validation	27
Figure 5.10 Membership Registration with Regular Expression Validation.....	28
Figure 5.11 Membership Registration Confirmation.....	29
Figure 5.12 Available Programs	31
Figure 5.13 Program Reservation with Required Field Validation	31
Figure 5.14 Program Reservation with Regular Expression Validation.....	32
Figure 5.15 Program Reservation Confirmation.....	33
Figure 5.15 Program Reservation Failure.....	34
Figure 6.1 Summary Report for Home.aspx	41
Figure 6.2 Summary Report for About.aspx.....	41
Figure 6.3 Summary Report for Video.aspx	41
Figure 6.4 Summary Report for Memberships.aspx.....	41
Figure 6.5 Summary Report for Trekking.aspx	41
Figure 6.6 Summary Report for 100 users.....	41

Figure 6.7 Summary Report for 200 users	42
Figure 6.8 Summary Report for 300 users	42
Figure 6.9 Summary Report for 400 users	42
Figure 6.10 Summary Report for 500 users	42

Acknowledgements

I would like to express my gratitude to Dr. Daniel Andresen for his encouragement, guidance and advice during the course of my work on this project. Also, I would like to thank him for all the facilities and support that were required during the development of this application.

I would like to thank Dr. Gurdip Singh and Dr. Mitchell Neilsen for serving on my supervisory committee.

CHAPTER 1 - Introduction

1.1 Aim

The objective behind developing this application is to improve user experience while using a website by improving the user interface. In order to accomplish this, three modules on the website have been developed using the Microsoft web application framework Silverlight. The web application is an attempt to mimic a part of the functionalities available on the Youth Hostels Association of India (YHAI) website. The functionalities made available on the website currently are:

1. A convenient and easy to use web navigation interface as the home page.
2. A picture slideshow to make the home page more attractive.
3. Smart information display for static and short content like contact details.
4. An easy way of browsing through pages with static content without contacting the web server.
5. A custom video player for the videos hosted on the website.
6. A simple interface for a user to register as a member with the website. The user receives an instant membership number generated by the system.
7. A simple interface to view the available trekking programs and make a reservation if interested. Only registered members with a valid membership number can make reservations.

1.2 Scope

This application can be used to register as a member of the association online. Also, registered and confirmed members can also make reservations for the various trekking and other

programs that keep coming up with time online. Members and non-members alike can browse through the website's information, history, contact details and view videos.

1.3 Need of the Application

Improving the user experience with the help of technologies that provide a richer look and feel will make the website more attractive for browsing through. The functionalities of registering for memberships as well as making reservations for the various programs online will help the user to do so from the comfort of his/her home. This also provides the association with an opportunity to ensure that their programs reach a wider audience. There is the added possibility of finding advertisements for the website is further enhanced by using rich internet applications that are built using technologies like Silverlight.

1.4 Related Work and Problems

The user experience on the existing website for the Youth Hostels Association of India and many others could be improved by using new technologies while building them. Such an effort will not only improve the look and feel of the websites, but could also help in minimizing the number of times the server is contacted for information, thus improving the performance of the system and reducing the network traffic.

1.5 Microsoft Silverlight

1.5.1 Overview

Microsoft Silverlight is a cross-browser, cross-platform implementation of the .NET Framework which provides a retained mode graphics system and integrates multimedia, graphics, animations and interactivity into a single runtime environment. The user interfaces are declared in Extensible Application Markup Language (XAML) and programmed using a subset

of the .NET Framework. Silverlight textual content can be searched and indexed by search engines since it is represented as text in the form of XAML and not compiled.

Silverlight can playback media content of file types WMV, WMA and MP3 on all the supported browsers without need for having the Windows Media Player, the Windows Media Player ActiveX control or the Windows Media browser plug-in. Silverlight can read data from a database and update the display without interrupting the user by refreshing the whole page. Silverlight enables the loading of XML content dynamically. Using the technique which is consistent with conventional Ajax techniques, the manipulation of such content using a DOM interface, can be done. Any .NET language can be used to write the programming logic for Silverlight applications.

1.5.2 XAML

Extensible Application Markup Language (XAML) is a declarative markup language developed by Microsoft. It is based on XML and can be used to initialize structured values and objects. Vector graphics and animations are made possible by marking up in XAML. Visual design tools like Microsoft Expression Blend can be used to create and edit XAML files.

Alternatively, any standard text editor or a code editor can be used.

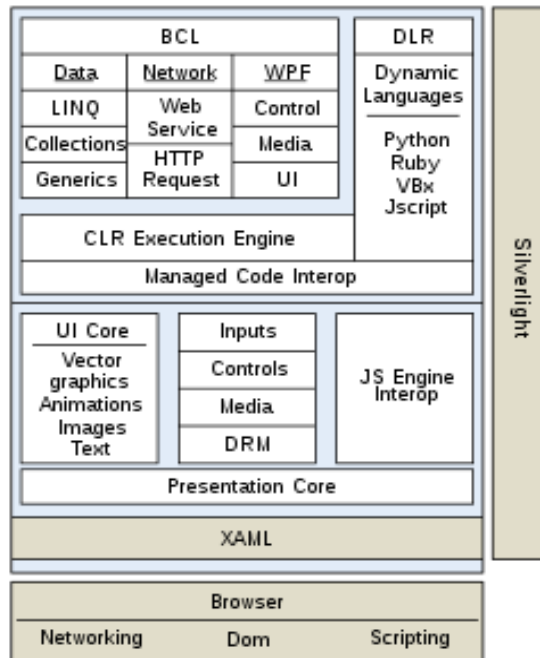
XAML can be used to create visible user interface (UI) elements in the declarative markup. A separate code-behind file is used to respond to the events and manipulate the objects declared in XAML. The elements in XAML map directly to the CLR (Common Language Runtime) object instances. The attributes in XAML map to the CLR properties and events on those objects. The key aspect of the technology is the reduced complexity needed for tools to process XAML because of its basis on XML.

1.5.3 Silverlight 2.0

Silverlight 2.0 implements the same full Common Language Runtime version as .NET Framework 3.0. This makes it possible to execute programs written in any .NET language. The markup files (.xaml files) in any Silverlight 2.0 applications have an associated code-behind file (.xaml.cs files in case of C#). This file can be used to manipulate the Silverlight application as well as the page which hosts the Silverlight control programmatically. The .xaml file and the corresponding code file are both compiled into .NET assemblies. These are then compressed using ZIP and are stored into files with the “.xap” extension.

Some of the features of Silverlight 2 include extensible controls, XML Web Services, networking components and LINQ APIs. This class library is a subset of the .NET Framework's Base Class Library (BCL). Since Silverlight code runs in a sandbox, the invocation of platform APIs is prevented.

Figure 1.1 Silverlight 2.0 Architecture



The .NET Framework in Silverlight enables the use of shapes, documents and media and animation objects. The classes for collections, reflection, regular expressions, string handling and data access are provided by the .NET Base Class Library. Silverlight also fully supports LINQ to Objects and expression trees. Object serialization for data persistence is also supported. In addition to XML, Silverlight can also handle data in RSS or JSON format. For working with XML data, the .NET BCL provides enhanced support. Silverlight 2 also supports asynchronous programming with the help of the threading libraries.

Silverlight applications can communicate using HTTP, or at the lower socket level by using the networking support provided by Silverlight. Silverlight includes classes for accessing data over any web services that are based on XML, Windows Communications Framework services and also ADO.NET data services. The Dynamic Language Runtime (DLR) provides support for the compilation and execution of scripting languages like JScript.

1.6 Platform Specifications - Development

1.6.1 Hardware Specification

Processor: Pentium IV 2.8 GHz

RAM: 2 GB

1.6.2 Software Specification

Operating Systems: Windows XP/Server 2003

Platform: .NET Framework

IDE: Visual Studio 2008 with Silverlight tools installed

Silverlight SDK and Silverlight developer runtime

Database: MS SQL Server 2008

CHAPTER 2 - System Requirement Analysis

2.1 Information Gathering

A major part of gathering information to start this project was to learn about Silverlight. Many articles on the internet were extensively studied to understand the development of rich internet applications using Silverlight. Knowledge about creating Silverlight applications, hosting them on .aspx and .html pages was gained. Visual layout management, using WCF web services to interact with databases, using user controls to eliminate the need for loading pages from the server, creating video player – all these using Silverlight were learned.

Having to work on the ASP.NET modules that deal with membership registration and program reservation warranted a study of the interaction between ASP.NET using Visual C# and MS SQL Server.

2.2 System Feasibility

The system feasibility can be divided into the following sections:

2.1.1 Economic Feasibility

The project is economically feasible as the only cost involved is having a computer with the minimum requirements mentioned earlier. For the users to access the application, browse through, registering as members and making reservations for programs, the only cost involved will be in getting access to the Internet.

2.1.2 Technical Feasibility

To deploy the application, the technical requirements are:

- Operating Systems: Win XP/Server 2003

- Platform: .Net Framework, Silverlight 2.0 runtime and IIS
- Database: MS SQL Server 2008

For Users:

- A Silverlight compatible internet browser like Microsoft Internet Explorer
- An internet connection

2.1.3 Behavioral Feasibility

The application requires no special technical guidance and all the pages available on the website are self explanatory. Instructions are provided where required. A user can read these instructions at any point of time.

CHAPTER 3 - System Analysis

After analyzing the requirements, the functionality of the web application can be depicted with the diagrams: the database schema diagram, data flow diagram and the state transition diagram. These were the basis for figuring out the database schemas, the flow of information through the website and the various states it could be in.

3.1 Database Schema Diagram

The following diagram depicts the different database tables that are needed to implement the functionalities of the website.

Figure 3.1 Database Schema Diagram

The figure displays three database table schemas. Each table is presented in a window-like format with a title bar and a scrollable table area. The 'Programmes' table has a primary key on ProgramID. The 'ProgramBooking' table has a foreign key on ProgramID. The 'Memberships' table has a primary key on MembershipID.

Column Name	Data Type
ProgramID	int
ProgramName	varchar(100)
Level	varchar(20)
Availability	int

Column Name	Data Type
ProgramID	int
MembershipNumber	int
Name	varchar(50)
Address	varchar(50)
PhoneNumber	varchar(20)
ReportingDate	date

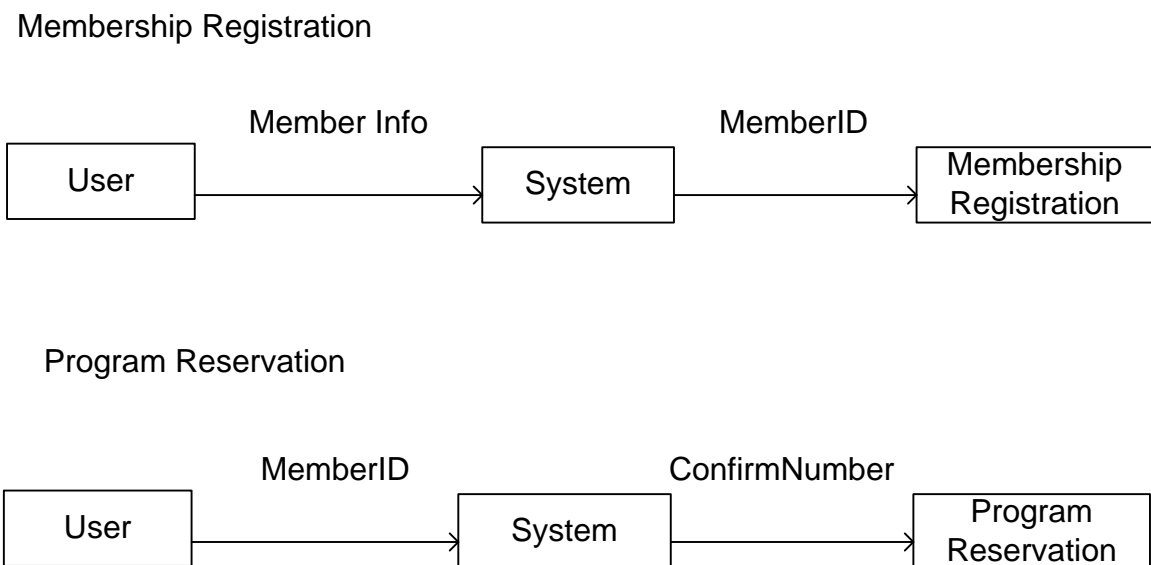
Column Name	Data Type
MembershipID	int
MembershipType	varchar(50)
MemberName	varchar(50)
Gender	varchar(10)
Email	varchar(50)
PhoneNumber	varchar(50)
DOB	date
Address	varchar(50)
State	varchar(50)

The information about the registered members is stored in the “Memberships” table. The current programs are stored in the “Programmes” table. When a member makes a reservation online, that information is stored in the “ProgramBooking” table.

3.2 Data Flow Diagram

The following diagram depicts the two different data flows that occur in the website. The user provides the required information to the system while registering as a member. The system generates a unique membership number to the user. In case of Program Reservation, the user provides his membership number to the system. If it is valid, the system creates a reservation for the user and generates a confirmation number.

Figure 3.2 Context Level Data Flow Diagram

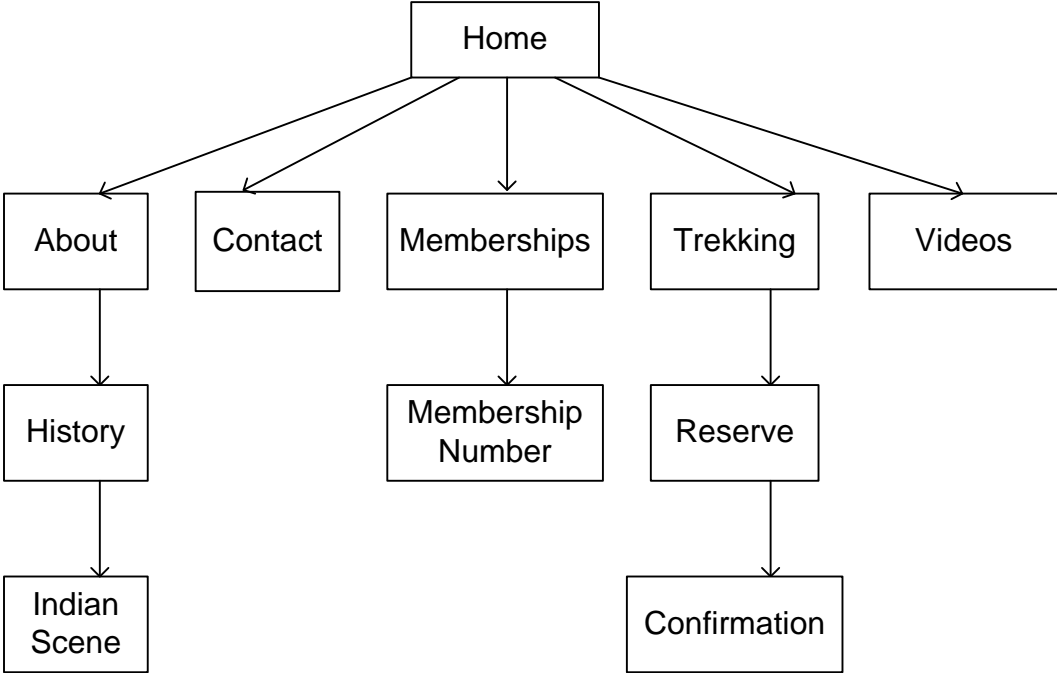


3.3 State Transition Diagram

The following diagram depicts the flow through the website along with the different states that it could be in. The basic flow is taken from the YHAI website. However, some changes are made to utilize the power of Silverlight. The branch from the Home page leading to

the Indian Scene is implemented as a single Silverlight application. So, no new pages are requested from the server. The Contact is implemented as a user control on the Silverlight application that implements the Home page interface. The Videos page is added to showcase the capability in Silverlight to build video players. The membership registration and the trekking are functionally similar.

Figure 3.3 State Transition Diagram



CHAPTER 4 - Design

4.1 Design Goals

The design of the website is the sum total of the design of the following:

1. Design of a database which holds the member information, reservations and generates membership and reservation confirmation numbers dynamically.
2. Design of a structure which helps the user to navigate from one part of the application to the other at any point of time.
3. Design of a Silverlight application that implements the functionality of the home page and provides a means for navigation.
4. Design of a Silverlight application that implements the functionality of displaying the “About” information as well as being able to access related information.
5. Design of a Silverlight application that implements the functionality of playing videos on the website.
6. Design of a Silverlight user control that implements the functionality of displaying the contact information by using a pop-up effect from the home page.
7. Design of a picture slideshow for the home page.

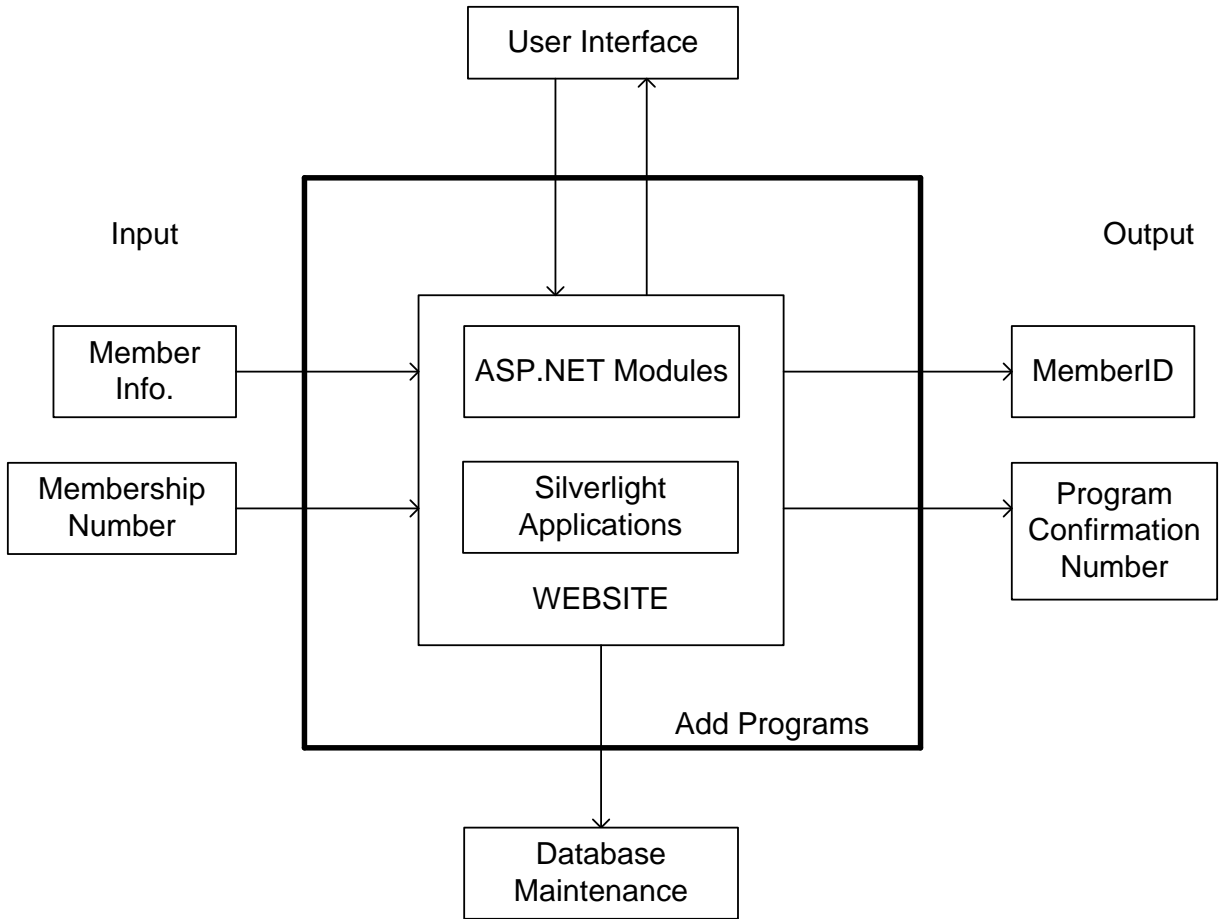
4.2 Architectural Design

In the following Architectural Context Diagram, the information provided to and received from the ‘Website’ is identified. The arrows represent the information received or generated by the Web Application. The closed boxes represent the set of sources and sinks of information.

In the system, we can observe that the user interacts with the application through a graphical user interface. The inputs to the system are the member information provided by the user at the time of registration as well as the membership number provided at the time of making a reservation for a program. Also, the output is in the form of the membership number and the program reservation confirmation number that are based on the inputs. There are some static outputs which do not change over the execution of the project such as the information displayed in the About section as well as the Contact details. Other than this, the maintenance of the system can be done by a website administrator which could be the addition, updates or deletion of the programs.

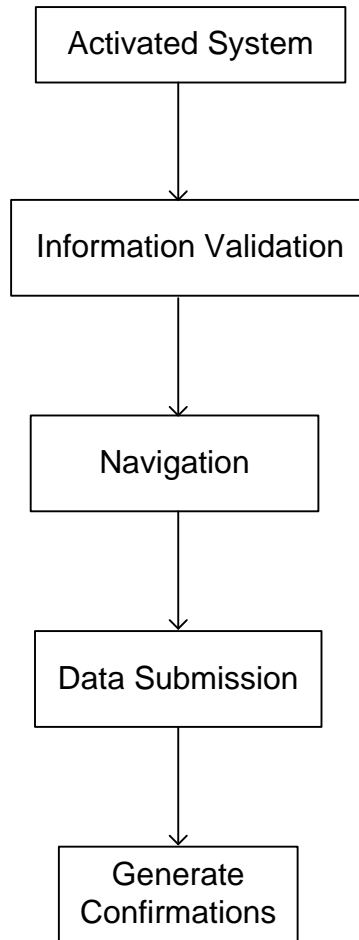
4.2.1 Architectural Context Diagram

Figure 3.1 Architectural Context Diagram



4.2.2 Control Hierarchy

Figure 4.2 Control Hierarchy



4.3 Procedural/Modular Approach

The following “modules” have been designed as a part of the website. Each of the Silverlight applications is considered as a single separate module.

4.3.1 Home page Silverlight application

This is the application that loads when the website is first launched. The Home page (.aspx) file hosts this application. It provides the interface for the user to select a particular “place” on the website by clicking a button. The different buttons lead to different “modules” in the website. A picture slideshow is embedded to make the interface more attractive. It also has a button which displays Contact information when clicked. This display is a Silverlight user control housed in this application.

4.3.2 About page Silverlight application

This application is a multi-page Silverlight application. It is loaded when the user navigates to the “About” page. The first page that is displayed has text that describes the “about” of the YHAI. The button at the bottom of this page navigates the user to the “History” page when clicked. This page is a Silverlight page and a part of this same application. This page has two different buttons at the bottom. The left button takes the user back to the “About” page whereas the right button takes the user to the next page which holds more details about the history of YHAI when clicked. This page has a button to go back to the “History” page when clicked. The number of pages in this application could be increased if needed based on the amount of information.

4.3.3 Video page Silverlight application

This application can play any videos that are supported by Silverlight. It is very useful for a website of this nature to be able to provide the users with media content. The video player in the application could be customized as required using Silverlight controls in the XAML file and/or programmatically.

4.3.4 Membership Registration Module

This ASP.NET module includes a web form that facilitates in procuring the user information for the purpose of registration as a member with the association. It also carries out data validation before passing it on to the database. If the data is not validated, the form does not get submitted and no registration is recorded in the database. If the registration is successful, the user is navigated to the next page which displays his membership number.

4.3.5 Program Reservation Module

This ASP.NET module includes a web page that displays all the programs currently open for reservation as a list. From this list, the user can select a program for making a reservation. Once the selection is done, the user is navigated to a page which asks for a membership number. If the user is a registered member, his/her membership number can be entered. After this, the user is navigated to a page that displays the message about whether the reservation was successful or not depending on whether the previously entered membership number was a valid one or not.

CHAPTER 5 - Implementation

5.1 Database Implementation

The database is implemented based on the database schema design discussed earlier using MS SQL Server 2008 Express edition. The following figure shows the different tables created in the database.

Figure 5.1 Database Implementation using MS SQL Server 2008



5.2 User Interface Design and Implementation

The user interface of the website has been designed using Microsoft Visual Studio 2008 (with Microsoft Silverlight tools installed). The controls provided by ASP.NET as well as Silverlight are used as necessary. The following sections detail the user interface implementation of the different “modules” mentioned earlier.

When a Silverlight application is created in VS 2008, by default, two files namely, Page.xaml and App.xaml are contained in the project along with their associated code-behind class files (depending on the programming language chosen). By default, the Page.xaml file is the initial control that is loaded when the application is launched. The App.xaml file can be used to declare application resources like styles that are shared by pages across the application. When a Silverlight application is built, VS 2008 will compile the code and the XAML file into a standard .NET assembly file, and then package it with any static resources like images or static files into a .xap file. VS 2008 can automatically add test html and aspx pages to the project in order to host the Silverlight application. We can also host an application in any of our other pages by including the particular .xap file.

One of the most important limitations faced during the user interface design was the non-availability of a visual design tool. The tools provided by Microsoft, the Deep Zoom Composer and the Expression Blend are proprietary and were not available for free. As a result, most of the design had to be done textually using the xaml markup language.

The lack of sufficient information about certain errors and exceptions generated by Silverlight along with the fact that it is a relatively new technology made the whole experience challenging and the development complex.

5.2.1 Home menu application

This application involved the designing of the interface by specifying the controls in the application's Page.xaml file. The button controls were used to provide the functionality of navigation. The design of the buttons themselves was created as a template in the App.xaml file. The shape of the buttons was designed using the "ellipse" control and the color was created by using the gradient control. The logic for navigation based on button clicks was written in the

code behind file in C#.NET. Whenever a button is clicked, the browser is directed to the corresponding URI. The background of the page is designed using Silverlight color gradient control.

A picture slideshow is embedded into the application on the home page. The pictures in the slideshow are the constituents of a “ListBox” – a Silverlight control. Silverlight also provides a library that could be used for tracking time. The pictures are displayed one after the other serially. Once the last picture in the list is reached, the picture displayed is set to the first picture in the list. The time interval for changing the picture could be set programmatically. In this application, it is set at 1 second. The slideshow is in operation as long as the home application is running.

The following screenshot depicts the Home page user interface.

Figure 5.2 Home page hosting the Silverlight application with buttons for navigation



Youth Hostels Association of India

ISO 9001:2000 Certified Organization



[Home](#) | [About Us](#) | [Memberships](#) | [Hostels](#) | [Videos](#) | [Trekking](#) | [Links](#)



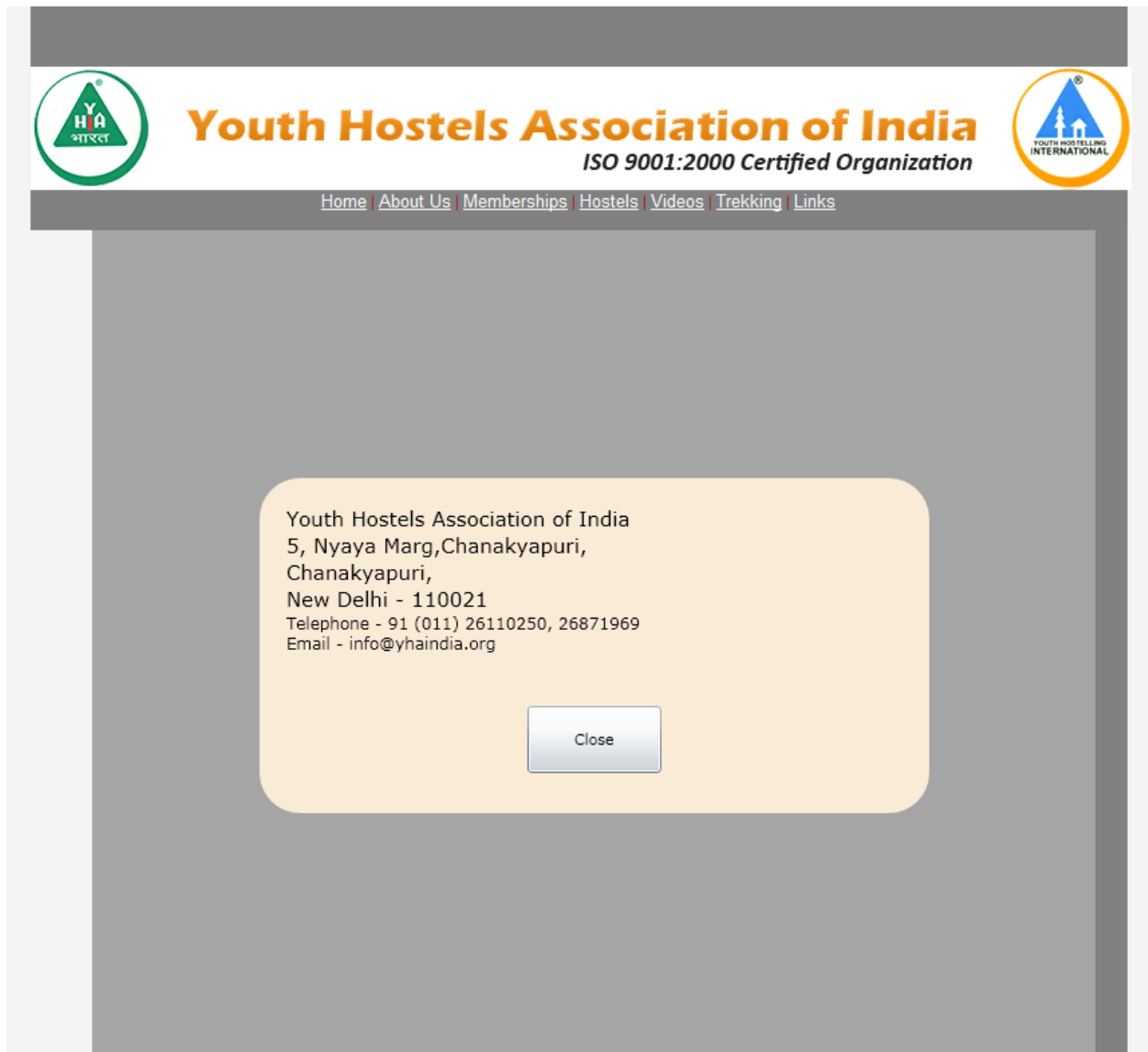
When the “Contact Us” button is clicked, a user control is displayed that contains the relevant information. It appears on top of the other content on the page and ensures that the user cannot do anything else with the page until the contact details user control is closed.

The design of the user control involves adding a new Silverlight user control to the project. It uses a rectangle control which is configured to stretch to take up all of the available space. Its background color is set to transparent. A second border control is then layered on top

of this rectangle control which takes up a fixed width. This control holds the contact details as well as a close button. The logic for closing this is written in the code-behind file of the user control.

The following screenshot depicts the Contact user control that is displayed when the Contact Us button is clicked.

Figure 5.3 Contacts user control



5.2.2 About Us application

This application allows a user to switch from one page to another within itself, thus eliminating the need to contact the server to fetch a new page as is the case in classical web applications. This approach intends to exploit the fact that every “page” in a Silverlight application is actually a user control and that one user control can contain another as its “content”. The architecture for building multiple page applications in Silverlight needs a user control page (the container), which will be empty. This will then be filled programmatically with the contents of other pages (the contents) depending on the user input. Each of the “contents” pages is created only when there is a request and then each can fill up the “container” as necessary.

The following screenshot depicts the About page. This is the first page that is displayed when the About application loads from the About.aspx page. Clicking on the button at the bottom navigates the user to the History page in the same application.

Figure 5.4 About page hosting the Silverlight application



Youth Hostels Association of India

ISO 9001:2000 Certified Organization



[Home](#) | [About Us](#) | [Memberships](#) | [Hostels](#) | [Videos](#) | [Trekking](#) | [Links](#)

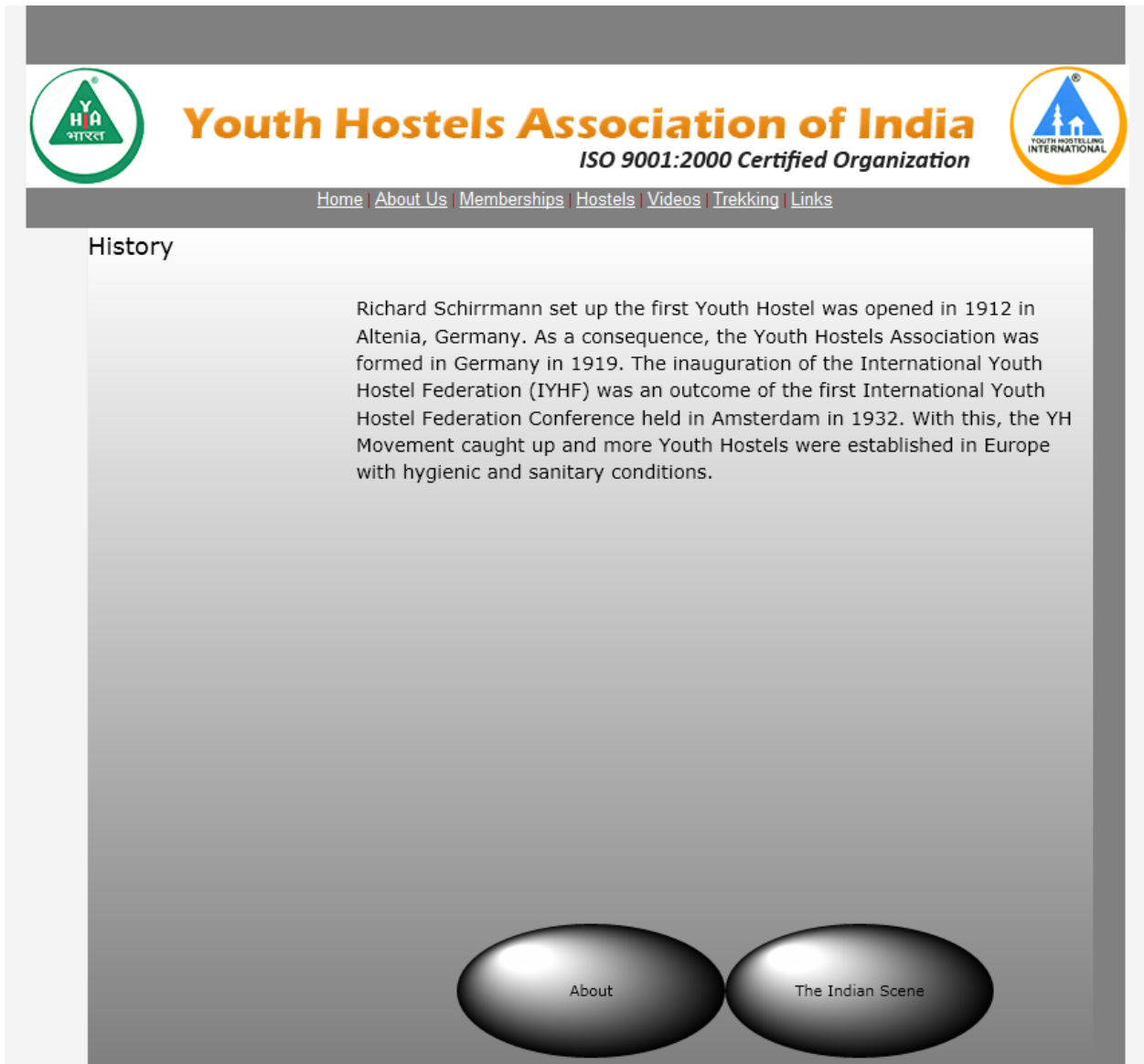
About Us

Youth Hostelling as a movement has not taken long to establish in the world. During the course of the last almost eighty years, it has spread from country to country and spanned all the five continents. The movement is based on two basic urges in the youth - love for the unknown, which includes desire for adventure and the desire to make friends. Since the movement started in Germany in the second decade of last century, there was very little opportunity for young people of limited means to travel at will and see and enjoy the beautiful places in their own country and the world around. The virtues of travel have been acknowledged in all ages and in all climes. Its hazards and uncertainties and costs used to be too great to allow the common youth to get as much benefit from it as was desired. The development of the means of communication and the establishment of youth hostels in more beautiful and attractive spots in many countries has made the opportunity of travel at a nominal cost far more easily available for youth.



The following screenshot depicts the History page of the Silverlight application hosted on the About.aspx page. Clicking on the left button at the bottom navigates the user back to the About page whereas the right button navigates the user to the Indian Scene page in the same application.

Figure 5.5 History page of the Silverlight application



The following screenshot depicts the Indian Scene page of the Silverlight application hosted on the About.aspx page. Clicking on the button at the bottom navigates the user back to the History page in the same application.

Figure 5.6 Indian Scene page of the Silverlight application



Youth Hostels Association of India

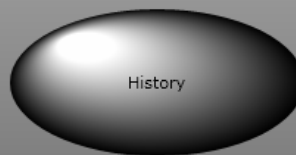
ISO 9001:2000 Certified Organization



[Home](#) | [About Us](#) | [Memberships](#) | [Hostels](#) | [Videos](#) | [Trekking](#) | [Links](#)

The Indian Scene

The hostelling principles had their roots in the ancient Indian wisdom in the form of resident places of learning, which imparted conventional and non-formal system of education. The modern Youth Hostel Movement has thus a close resemblance with our age old pilgrim idea. Education was not considered complete in olden days till the disciple was sent out on a pilgrimage. This meant a hard and strenuous journey to sacred spots all over the country. The pilgrim centres were located in such beautiful places that a journey to them by itself was an elevating experience. All the adventures and hardships encountered on the journey strengthened and disciplined the mind. Meeting people at road and in shelters like dharamshalas and rest houses in a simple friendly way with different backgrounds and cultures marching on road for the same goal, helped to break down the barriers and broadened the mind and liberated the spirit. It has been said that the pilgrim centres, more than anything else, have maintained the oneness of India. All this conforms to the youth hostelling ideals in better and spirit.



5.2.3 Video Player application

This application is created to play videos on the website. Currently, it needs a media file supported by Silverlight to be placed in the file system. The “MediaElement” control is the key which is responsible for playing media files. A button placed on the page, when clicked, loads the Uri of the media file into the Silverlight application. This functionality of programmatically determining the media file to be loaded can be used to have multiple buttons that can load different media files to implement a playlist for the media player.

The toggle button control is used to play/pause the video programmatically. Also, the slider control is used to show the progress of the video in the player. It also includes a timer that is displayed next to the slider to indicate the time duration as well as the progress of the video in terms of time.

The following screenshot depicts the Silverlight application playing a video that is hosted in the Video.aspx page. It has a media player that can play files supported by Silverlight.

Figure 5.7 Silverlight Media Player



5.2.4 Membership Registration

The implementation of this module involved the design of a simple web forms interface. The drop down menu has a static list of items. The rest of the fields are used to gather user input. If the data input is valid, in the code-behind, it is extracted from the various fields programmatically. Then, a connection to the database is established and the gathered data is stored into the appropriate table. The return value from the database is caught and displayed on a page after redirection.

The following screenshot depicts the membership registration form developed in ASP.NET along with the required field validation controls.

Figure 5.8 Membership Registration with Required Field Validation



Membership Registrations

* Mandatory fields

Membership Type*

Name* * A name is required

Gender* Male
 Female

* Please select a gender

Email* * An email is required

Phone* * A phone number is required

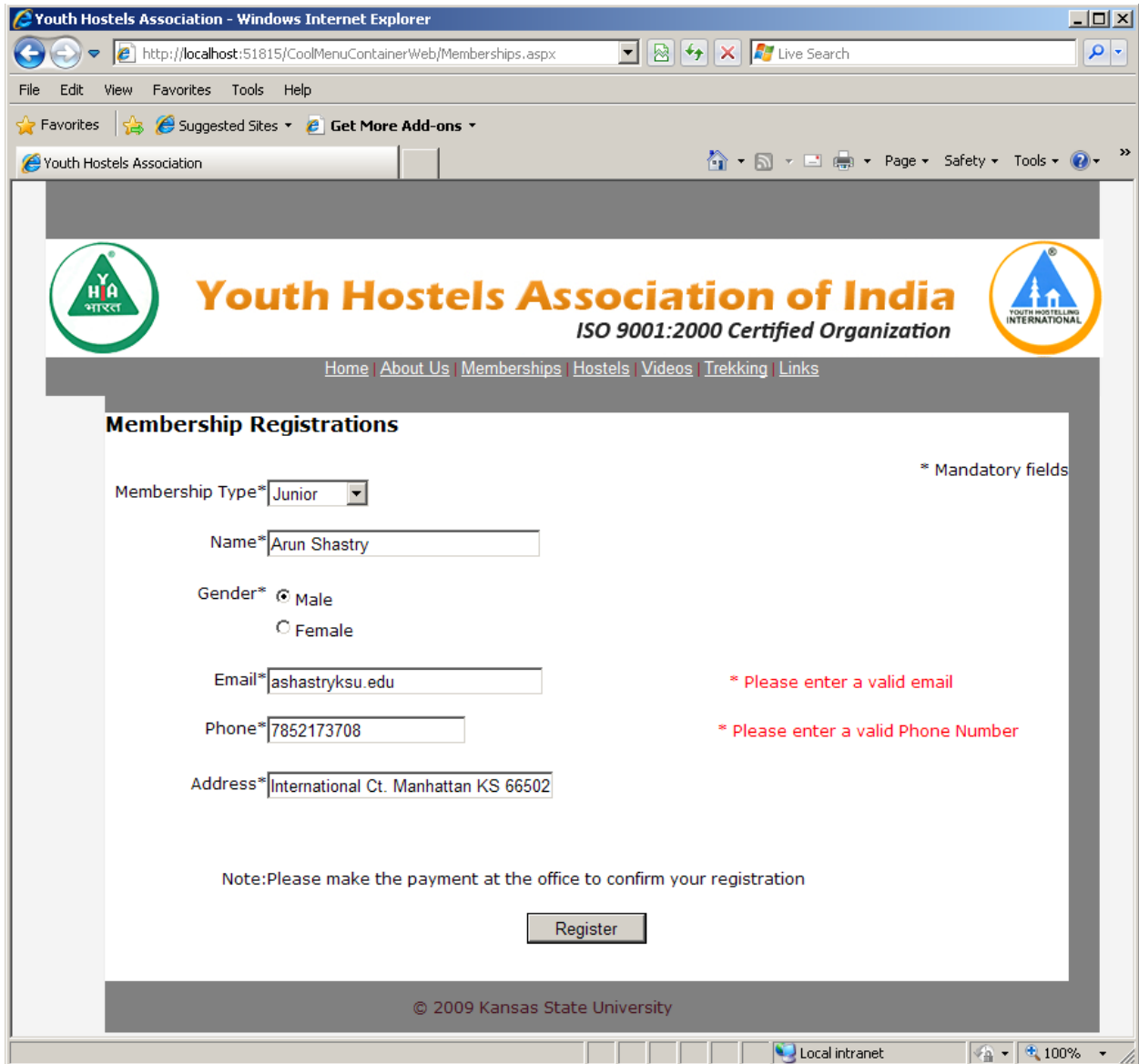
Address* * An address is required

Note: Please make the payment at the office to confirm your registration



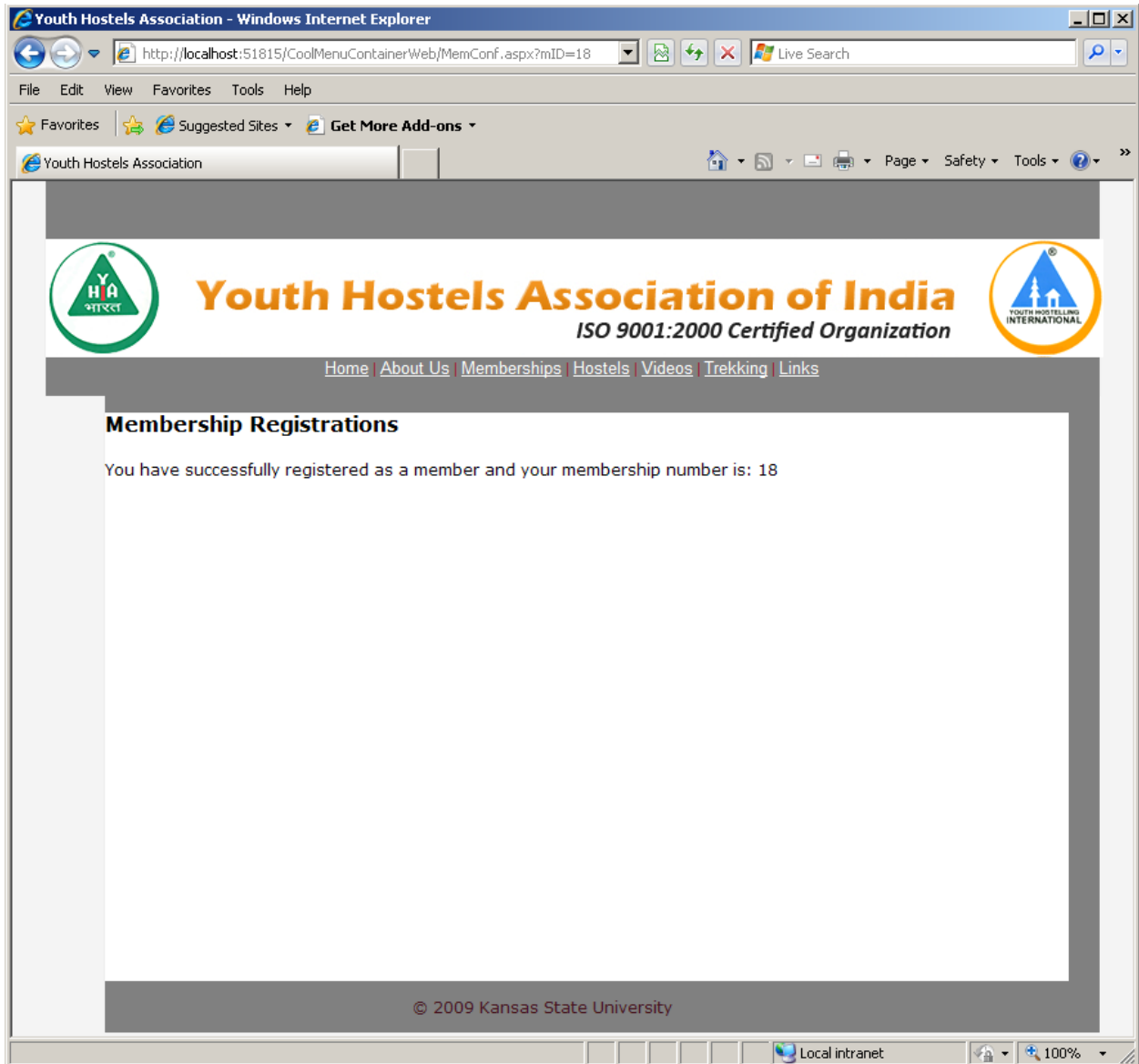
The following screenshot depicts the membership registration form along with the regular expression validation controls for the email and phone number fields.

Figure 5.9 Membership Registration with Regular Expression Validation



The following screenshot depicts the confirmation and an accompanying number after membership registration.

Figure 5.10 Membership Registration Confirmation



5.2.5 Program Reservation

The implementation of this module involved the use of two web pages. The first one displays the available programs with the help of a “datalist” control to which data is bound from the database. The user selection from the first page is passed on to the redirected page where the relevant program name is displayed along with a text box that prompts for the membership number.

The following screenshot depicts the list of programs available from the association.

Figure 5.11 Available Programs

The screenshot displays the homepage of the Youth Hostels Association of India. At the top, there are two logos: the YHA India logo on the left and the Youth Hostelling International logo on the right. The main heading is "Youth Hostels Association of India" in orange, with "ISO 9001:2000 Certified Organization" below it. A navigation menu includes links for Home, About Us, Memberships, Hostels, Videos, Trekking, and Links.

ProgramName	ProgramID	Reserve Online
NATIONAL BIKING EXPEDITION GOA 2009	2000	Select
NATIONAL HIMALAYAN SANDAKPHU-GURDUM TREKKING EXPEDITION-09	2001	Select
NATIONAL BIKING EXPEDITION GOA 2009	2002	Select
NATIONAL SAILING EXPEDITION GOA 2009	2003	Select
NATIONAL TREKKING EXPEDITION GOA 2009	2004	Select
NATIONAL FAMILY ADVENTURE CAMPING EXPEDITION GOA 2009	2005	Select

Four images are displayed on the right side of the page: a cave temple, a boat on a river, a group of people in yellow jackets holding signs, and a person on a motorcycle in a mountainous landscape.

The following screenshot depicts the program booking page with required field validation for membership number.

Figure 5.12 Program Reservation with Required Field Validation



Program Booking

Program Name: NATIONAL BIKING EXPEDITION GOA 2009

Membership Number:

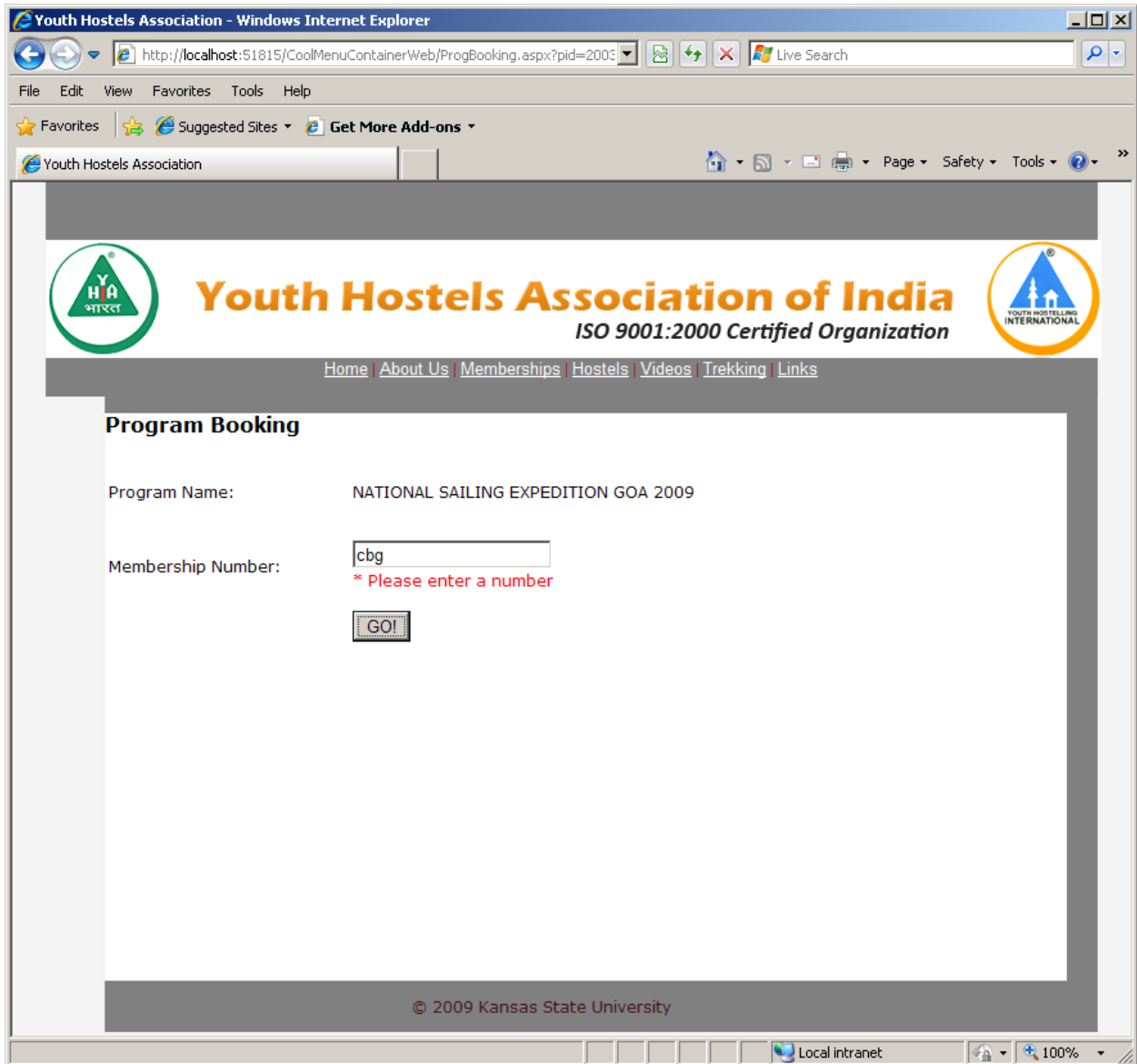
* Membership Number is required for Program Booking

GO!



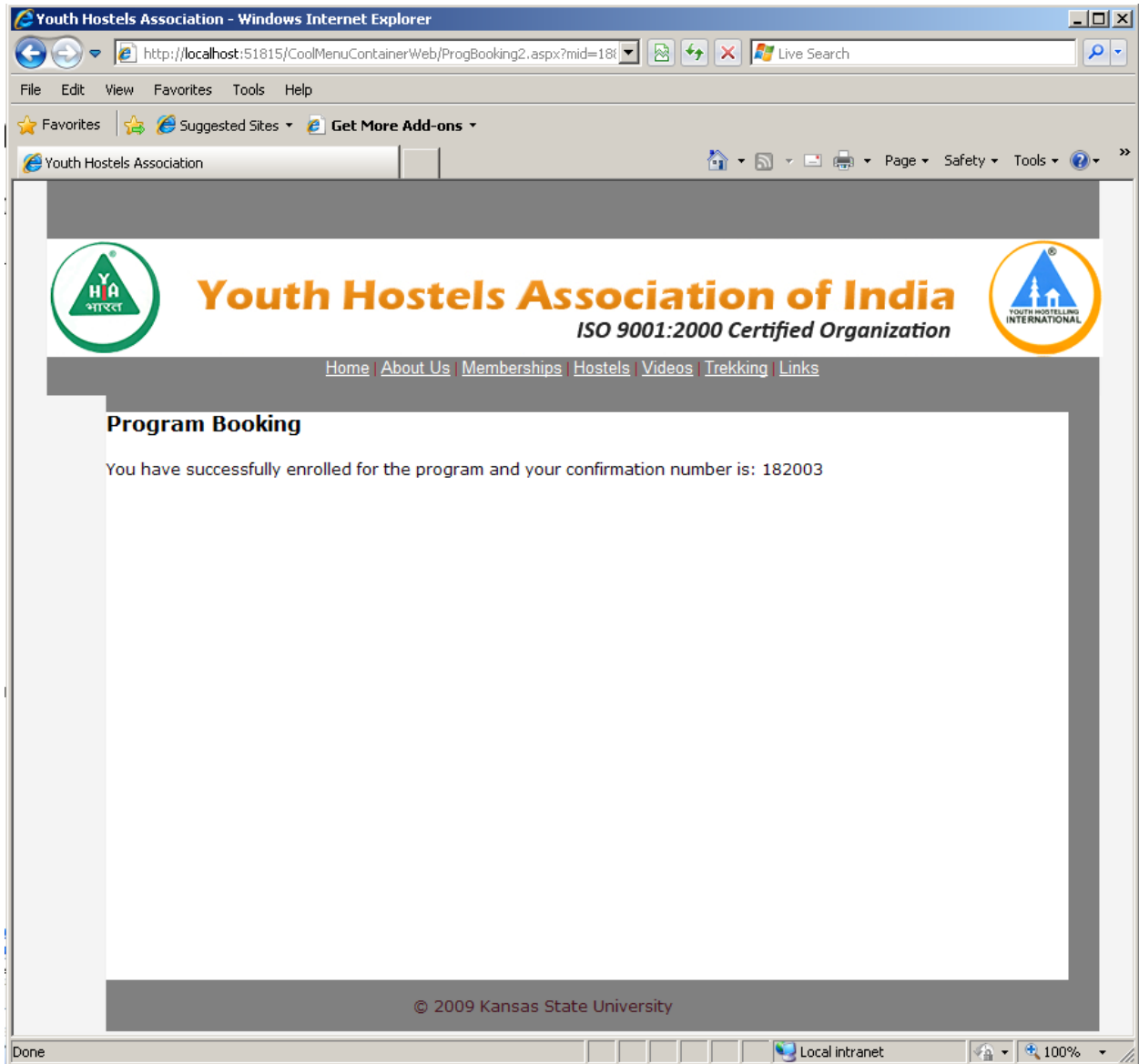
The following screenshot depicts the program booking page with regular expression validation for membership number.

Figure 5.13 Program Reservation with Regular Expression Validation



The following screenshot depicts the program confirmation page with the confirmation number.

Figure 5.14 Program Reservation Confirmation



The following screenshot depicts the program confirmation failure.

Figure 5.15 Program Reservation Failure

Windows Internet Explorer browser window showing the Youth Hostels Association of India website. The address bar displays the URL: <http://localhost:51815/CoolMenuContainerWeb/ProgBooking2.aspx?mid=89>. The page title is "Youth Hostels Association".

The website header features the Youth Hostels Association of India logo (left) and the Youth Hostels International logo (right). The main heading is "Youth Hostels Association of India" with the text "ISO 9001:2000 Certified Organization" below it. A navigation menu includes links for [Home](#), [About Us](#), [Memberships](#), [Hostels](#), [Videos](#), [Trekking](#), and [Links](#).

The main content area is titled "Program Booking" and displays the message: "Sorry, the membership number you entered does not match our records. Please try again."

The footer contains the copyright notice: "© 2009 Kansas State University".

The browser status bar at the bottom shows "Local intranet" and a zoom level of "100%".

CHAPTER 6 - Testing

Software testing is the process of running any software system with the intention of uncovering errors. It provides assurances for the quality of software and represents a final review after the other phases of creating software, like specification, design, code generation etc.

The website has been tested on Internet Explorer 8 and Mozilla Firefox 3.0.12.

6.1 Unit Testing

Unit testing is the effort to verify the smallest unit of software designs i.e. a software component or a module for correctness. Unit testing is a dynamic method for verification, where program is actually compiled and executed. Unit testing is performed in parallel with the coding phase. Unit testing tests units or modules not the whole software.

Each Silverlight application and ASP.NET module has been individually tested. During the development process, testing was carried out in parallel. All kinds of input and control flows within the Silverlight applications were tracked for ascertaining that the working was as expected.

The application implementing the home page was tested to see if the button clicks were leading to proper navigation. The picture slideshow was tested by changing the number of pictures displayed as well as changing the time intervals at which the pictures are changed. The user control that displays the contact details was tested to see if it completely disappeared from the view of the user when it was closed.

The application implementing the about page was tested to see if the displayed text was always in the expected form. The buttons on all the three different pages of this application were tested to see if they resulted in navigation that was consistently as expected.

The application that plays the video on the server was tested to see if the buttons on the user interface functioned as intended. It was tested to ensure that the video was not played until the user clicks the relevant button.

The module implementing the functionality of membership registration was tested to ensure that the data was being saved into the database. The required field validators and regular expression validators that are used to validate the data on the form were tested individually by supplying data that were expected to exercise them. The database was checked to ensure that the member information was stored in the table once the registration was completed successfully and not stored otherwise.

The module implementing the functionality of program registration was tested to ensure that the page displaying the programs only displays those programs that are available. The correctness of the user selection of a program was tested by displaying the name of the program on the page that is loaded after the selection is made. The page that prompts for the membership number was tested by supplying invalid membership numbers to ensure that the registration was not being accepted by the system. Also, the required field validator for membership number was tested. The regular expression validator on this page was tested by providing non-numerical data in the membership number field. . The database was checked to ensure that the program reservation information was stored in the table once the reservation was completed successfully and not stored otherwise.

6.2 Integration Testing

Integration testing is the process of testing a software system consisting of different modules for problems that might result from the interaction of its constituent modules. What

constitutes integration testing should be derived from the system specification. It could initially be done on an integrated prototype with minimum configuration.

After unit testing, the Silverlight applications and the ASP.NET modules were integrated into a system to form the website. Once these components were put together, the website was tested to ensure that there were no problems arising from their interactions and they worked together as expected.

6.3 Validation Testing

It is the process by which final assurances could be provided that a software system meets all the functional and behavioral requirements. Black box testing techniques are used.

As explained earlier in the section 6.1, during unit testing, test cases have been used to uncover possible errors in the functionalities and behavior of the ASP.NET modules as well as the Silverlight applications. After these were put together to form the website, the system was tested to ensure that the navigations occur as expected even when the browser back and next buttons are used.

6.4 White Box Testing

White box testing involves knowledge about the internal working of a piece of software. Tests can be conducted to ensure that all the internal functions happen as expected and also the different components and control flow paths are adequately exercised. Test cases are developed to include inputs that exercise specific logical paths.

As mentioned earlier, during unit testing, effort has been made to ensure that invalid inputs are provided, inputs that exercise boundary conditions as well as cause different logical paths to be exercised are used and the system responds to this as expected by writing out suitable error messages at appropriate places.

6.5 Performance Testing

Performance testing is an essential phase before any web application is deployed successfully. It could prove very crucial to know how the web application and the web server itself would behave as the number of users visiting a website increases. In order to subject a web application to this kind of a scenario, testing tools that can simulate the work load on a web server, by producing numerous user requests can be used. These tools allow us to use a single computer to simulate a large number of virtual users, concurrently requesting predefined pages of the application. The communication protocols between a web browser and a web server are simulated exactly by these requests.

The website has been tested for performance using the Apache JMeter. It is an open source, Java based desktop software designed to load test and measure performance of client/server software such as a web application. It can be used to simulate heavy loads on a server, network or object to test its strength or to analyze overall performance under different load types.

6.5.1 Testing with JMeter

The following steps were taken to create different tests using JMeter:

- A Thread Group element was added. This is used to specify the number of users to simulate, number of requests they can send and the time intervals between two requests.
- A HTTP Request Defaults element was added. This is used to specify the default settings for the HTTP requests like the server name and port number.

- Following the previous step, HTTP Request elements were added. Each of these is used to specify a separate page to be requested by typing in the specific path for the page.
- The final step was to add one or more Listener elements which are responsible for storing all of the results of the HTTP requests and presentation of the data.

6.5.2 Test Results

Each of the web pages of Home.aspx, About.aspx, Video.aspx, Memberships.aspx and Trekking.aspx were tested separately by setting the number of users to 1000. The following screenshots depict the results. In each of the following results, the table headings have the following meanings (taken from the JMeter User Manual at [5]):

- Label - The name of the test.
- Samples - The number of users (in this case).
- Average - The average elapsed time of a set of results
- Min - The lowest elapsed time for the samples with the same label
- Max - The longest elapsed time for the samples with the same label
- Std. Dev. - the Standard Deviation of the sample elapsed time
- Error % - Percent of requests with errors
- Throughput – number of requests per second. The time is calculated from the start of the first sample to the end of the last sample. This includes any intervals between samples, as it is supposed to represent the load on the server.
- Kb/sec - The throughput measured in Kilobytes per second
- Avg. Bytes - average size of the sample response in bytes. (in JMeter 2.2 it wrongly showed the value in kB)

Figure 6.1 Summary Report for Home.aspx

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
Home page	1000	1142	3	2787	487.38	3.00%	120.4/sec	440.16	3742.4
TOTAL	1000	1142	3	2787	487.38	3.00%	120.4/sec	440.16	3742.4

Figure 6.2 Summary Report for About.aspx

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
About Page	1000	1815	35	3377	705.84	1.60%	132.9/sec	488.73	3765.0
TOTAL	1000	1815	35	3377	705.84	1.60%	132.9/sec	488.73	3765.0

Figure 6.3 Summary Report for Video.aspx

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	1000	1162	35	3795	525.70	0.00%	144.2/sec	532.98	3786.0
TOTAL	1000	1162	35	3795	525.70	0.00%	144.2/sec	532.98	3786.0

Figure 6.4 Summary Report for Memberships.aspx

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
Memberships Page	1000	2221	2	4717	1420.35	7.90%	130.3/sec	709.89	5578.5
TOTAL	1000	2221	2	4717	1420.35	7.90%	130.3/sec	709.89	5578.5

Figure 6.5 Summary Report for Trekking.aspx

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
Trekking	1000	16114	2080	26780	6409.75	0.00%	35.1/sec	202.33	5901.0
TOTAL	1000	16114	2080	26780	6409.75	0.00%	35.1/sec	202.33	5901.0

Tests were created to simulate the load on the server when users are requesting for any of the pages in parallel by including the different HTTP Requests in the same test plan. The following tables depict the results for 100, 200, 300, 400 and 500 users.

Figure 6.6 Summary Report for 100 users

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
Home	100	235	15	491	150.48	0.00%	81.4/sec	302.11	3802.0
About	100	530	54	1204	335.12	0.00%	41.7/sec	154.69	3797.0
Video	100	941	25	1572	432.88	0.00%	27.6/sec	101.97	3786.0
Memberships	100	1392	165	2045	433.03	0.00%	19.9/sec	114.67	5903.0
Trekking	100	1779	851	2355	313.38	0.00%	14.8/sec	85.12	5901.0
TOTAL	500	975	15	2355	659.75	0.00%	67.2/sec	304.58	4637.8

Figure 6.7 Summary Report for 200 users

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
Home	200	652	7	2051	543.79	0.00%	77.1/sec	286.38	3802.0
About	200	1187	40	2073	527.83	0.00%	47.9/sec	177.63	3797.0
Video	200	1788	26	3488	649.04	0.00%	26.2/sec	97.00	3786.0
Memberships	200	2682	69	4081	1034.87	0.00%	20.6/sec	118.72	5903.0
Trekking	200	2638	1551	4997	745.45	0.00%	17.1/sec	98.77	5901.0
TOTAL	1000	1790	7	4997	1076.64	0.00%	82.6/sec	374.18	4637.8

Figure 6.8 Summary Report for 300 users

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
Home	300	945	194	1517	346.79	0.00%	114.0/sec	423.36	3802.0
About	300	1518	840	2121	314.35	0.00%	67.6/sec	250.48	3797.0
Video	300	1929	1429	2335	191.92	0.00%	52.4/sec	193.64	3786.0
Memberships	300	3022	1841	4960	961.58	0.00%	32.4/sec	187.04	5903.0
Trekking	300	4180	2390	5224	630.35	0.00%	28.0/sec	161.63	5901.0
TOTAL	1500	2319	194	5224	1281.88	0.00%	98.6/sec	446.36	4637.8

Figure 6.9 Summary Report for 400 users

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
Home	400	1461	561	2568	517.33	0.00%	109.6/sec	406.78	3802.0
About	400	2666	1703	3019	257.52	0.00%	72.6/sec	269.14	3797.0
Video	400	2843	2646	3110	45.16	0.00%	59.9/sec	221.36	3786.0
Memberships	400	3174	2826	5101	527.51	0.00%	44.6/sec	257.29	5903.0
Trekking	400	6770	3271	8627	1295.56	0.00%	30.2/sec	173.92	5901.0
TOTAL	2000	3383	561	8627	1913.51	0.00%	92.4/sec	418.68	4637.8

Figure 6.10 Summary Report for 500 users

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
Home	500	1877	399	3389	776.13	0.00%	116.7/sec	433.34	3802.0
About	500	3400	3061	3775	104.11	0.00%	73.1/sec	271.13	3797.0
Video	500	3571	3471	3930	64.43	0.00%	68.0/sec	251.58	3786.0
Memberships	500	4058	3686	4976	271.34	0.00%	56.7/sec	326.61	5903.0
Trekking	500	9072	4541	11991	1892.09	0.00%	30.4/sec	175.21	5901.0
TOTAL	2500	4396	399	11991	2617.92	0.00%	90.1/sec	408.07	4637.8

6.5.3 Test Result Analysis

From the tables in Figures 6.1 to 6.5, when the pages were requested separately, the response from the web server was in the order of less than 5 seconds. The numbers indicate that the request for `Trekking.aspx` consumed the most time because the page involves getting data into a datalist from the database.

From the tables in Figures 6.6 to 6.10, when the website was subjected to parallel requests to multiple pages, the numbers indicate that the responses from the web server took longer with increase in the number of users. Even at 500 requests in parallel requesting for each of the pages (for a sum of 2500 requests), there were no errors reported.

The performance testing of the website was conducted locally. Since the machine was also running other programs like the test tool itself along with the web server, the performance on a machine that runs only the web application is expected to be better. However, when considering a remote server-client scenario, factors like network traffic and bandwidth could degrade the time required to receive responses from the server.

CHAPTER 7 - Results

The website runs as intended. The three Silverlight applications developed load as expected when the page hosting them is requested from the server. The Silverlight user control loads without any delay as expected. The navigation through the Silverlight pages in the About application is as expected. The video player behavior is consistent with expected behavior.

The membership registrations are done correctly and the member information is correctly inserted into the database. In the program reservation module, the website checks for the validity of the member number from the database before confirming and generating a number.

CHAPTER 8 - Conclusion

This website is designed to showcase how parts of any website can be turned into interfaces that provide a rich experience to the users using technologies like Microsoft Silverlight. This could also mean faster responsiveness in cases like the “About” and “Contacts” where content is displayed as part of the same application reducing network traffic and requests to the server. The online facilities for membership registration and program reservations could help in the increase of people joining the association and taking part in the programs. The video player facility would attract potential members much better than a website without the facility.

8.1 Limitations

Since a Silverlight application is loaded in entirety, if the users are not interested in some of the pages, it could be wastage of memory, especially if such pages hold huge amounts of data.

Another limitation is that a member has to make a payment in person at the office to confirm his registration or program reservation.

8.2 Future Work

The website can be modified to incorporate some form of online payment like using a credit card. A user interface could be provided for an administrator to add, update and delete programs. Also, the Silverlight applications could be designed better to improve the look and feel of the website. The video player could provide the users with more options, perhaps like www.youtube.com.

References

- [1] <http://en.wikipedia.org/wiki/Silverlight>
- [2] <http://en.wikipedia.org/wiki/XAML>
- [3] www.silverlight.net
- [4] www.msdn.com
- [5] http://jakarta.apache.org/jmeter/usermanual/component_reference.html