Family Nutrition Program

Report Sheet

CIS 690 – Implementation Project

Main URL: www.oznet.ksu.edu/sdev-mk/fnprs_new/welcome.aspx

Department of Computing and Information Sciences

Submitted by:

Mohamed Askar

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2004
Table of Contents

Table of Contents

1. Introduction
   1.1 Objective
   1.2 Target Audience

2. Overall Description
   2.1 System Architecture
   2.2 Database Design
   2.3 Functionalities
   2.4 Control Flow
   2.5 Improvements

3. Testing
   3.1 Microsoft Application Center Test
   3.2 ANTS Load Test
   3.3 Problems Faced
   3.4 Snap Shots
   3.5 References
1. **Introduction**

1.1 **Objective**

The main objective of this project is to convert the existing FNPRS website into a new one using ASP.NET. It also includes improving the user interface and adding some additional features, such as downloading the data directly into an Excel sheet.

1.2 **Target Audience**

This site will enable county agents throughout Kansas, to enter their program activity sheets for the Family Nutrition Program.

2. **Overall Description**

This project is mainly focused on increasing the speed of the FNPRS website. The current website, which was done using ASP, is very slow, especially on dial-up connections. The current application also takes a large amount of time to load.

The reporting for the Family Nutrition Program also took a large amount of time since the data from the table was collected manually and then fed into Excel sheets for calculation.

To avoid these problems, the new FNPRS website was created using ASP.NET. Special features were also created to download the data directly into an Excel sheet. The user can also view data for calculation in the website.

The user interface of the website was also improved tremendously. The user is now able to enter the program activity sheet quickly.
2.1 System Architecture

![System Architecture Diagram]

- **Front End** (HTML and C#): County Agent, Manage Agent session
- **Middleware** (.NET Framework): Add, Edit, Update, Print Program Activity Sheet
- **Back End** (Oracle 9i): Database
- **Admin**: Manage Admin session
  - Perform Admin activities


Technologies

a) Microsoft Visual Studio .NET

The .NET framework is the infrastructure for the new Microsoft .NET platform. It is a common environment for building, deploying, and running web services and web applications. It contains common class libraries, such as ADO.NET, ASP.NET and Windows Forms, to provide advanced standard services that can be integrated into a variety of computer systems.

It is language-neutral. Currently it supports C++, C#, Visual Basic, JScript, and COBOL. Third-party languages, such as Eiffel, Perl, Python, Smalltalk, and others, will also be available for building future .NET framework applications.

The new Visual Studio.NET is a common development environment for the new .NET framework. It provides a feature-rich application execution environment, simplified development, and easy integration between a number of different development languages.

b) Oracle 9i

The database server is Oracle 9i. It is robust and web-enabled. It offers easier fine tuning to serve the required purpose. It provides the most powerful database for secure, reliable, high-capacity Internet applications, data warehouses and e-commerce websites.

It is highly scalable. It scales to support the most demanding, high-throughput applications. It reduces the complexity of administration. It can manage the multimedia content directly. It provides complete data protection by means of Oracle Data Guard.

c) Microsoft Excel

Microsoft Excel allows one to create professional spreadsheets and charts. It performs numerous functions and formulas to assist us in our project.
2.2 Database Design

The program activity sheet table contains all the information about the program activity sheet. The pas_items table contains all the items which are placed in the program activity sheet. This table is linked with the program activity sheet table by means of pas_select table. The survey questions are stored in the survey_questions table. This table is linked with the program activity sheet table by means of survey_response table. The survey questions table is also linked to the pas_items table by means of survey_questions_link table. The user_counties table contains information about the user and his/her counties. The pas_counties_items table contains information about the id of each county.
2.3 Functionalities

The project is divided into three modules.

1) **Program Activity Sheet:**
   This module performs the following functions:
   a) The user can add the program activity sheet from the main page. The program activity sheet has a series of topics to be entered. They include:

   **General Information:**
   The user first enters the general information of the program activity sheet, i.e. date of activity, county, name of the person who delivered the activity, name of program site, and the participant category.

   **Success Story:**
   He can enter the success story of the program activity sheet.

   **Behavioral Objectives:**
   He can select one or more objectives from the list. The behavioral objectives which appear in the list depend on the participant category, which is selected on the general information box.

   **Survey Questions:**
   He can enter the survey questions for the program activity sheet. There are three types of survey questions (i.e. K-2, 3-6, and adult). The survey questions that appear on the page depend both on the participant category and the behavioral objectives. He can enter the survey questions for as many as participants he wants. When he selects a list of questions for the first participant, the remaining participants get the same questions. He can edit the responses that were previously added. He can also view the responses that were added in a separate window.
Direct and Indirect Contact:
He can enter the direct and indirect contact of the program activity sheets.

Primary Curriculum:
He can select one of the primary curriculums from a list. He can otherwise enter a curriculum in the “other” box.

Supplementary Curriculum:
He can enter the supplementary curriculum.

Activity Topics:
He can select one or more activity topics from the list. He can otherwise enter a topic in the “other” box.

Programming Messages:
He can select one of the programming messages from a list. He can otherwise enter a message in the “other” box.

Number and Length of Lessons:
He can enter the number and the length of the lessons.

Collaborators:
There are subcategories under the collaborators. He can select multiple entries from each subcategory. There is also a “clear” button placed in order to unselect the wrong entries.

Marketing:
He can select the information from the list describing how and where he marketed the activity.
Characteristics of Audience:
He can select one or more of the characteristics of the audience.

Grade and Gender of Participants:
He can enter the grade and the gender of the participants. The participants number must be equal to the one entered in the “ethnicity” box. This box appears only if the participant is either K-2 or 3-6.

Age and Gender of Participants:
He can enter the age and the gender of the participants. The participants number must be equal to the one entered in the “ethnicity” box. This box appears only if the participant category is adult.

Ethnicity / Race of Participants:
He can select one or more ethnicity and the race of the participants. He can also enter the ethnicity which is not in the list.

He can also enter the fruit and vegetable message, food stamp message and the success barriers.

c) He can edit the program activity sheet by clicking the “edit” link. All the data which have been entered can be changed. There is also a “view all” option, so that he can verify the changes he has made on the program activity sheet.

d) He can copy his program activity sheet by clicking the “copy” link. The copy feature helps him to add a similar program activity sheet quickly.

e) He can also view the complete program activity sheet by clicking the “view” link. He can print the program activity sheet from the view page. Style sheets are cascaded to get the desired data in the printouts.
f) The program activity sheets can also be sorted according to the month so that he can view his sheets quickly.

2) **Management Site:**

This site performs the following functions:

a) The admin can add counties, objectives, contacts, curriculum, activity topics, collaborators, survey questions, survey responses, programming messages, marketing messages, ethnicity, age, and gender of the participants.

b) He can delete counties, objectives, contacts, curriculum, activity topics, collaborators, survey questions, survey responses, programming messages, marketing messages, ethnicity, age, and gender of the participants.

c) He can update the text of counties, objectives, contacts, curriculum, activity topics, collaborators, survey questions, survey responses, programming messages, marketing messages, ethnicity, age and gender of the participants.

d) He can make the counties, objectives, contacts, curriculum, activity topics, collaborators, survey questions, survey responses, programming messages, marketing messages, ethnicity, age, and gender invisible and visible to the normal users.

e) He can add the counties for which he wants to view the program activity sheet. He can also delete the counties from the list of selected counties.
3) **Reporting Site:**

From this site the user can download the data required for reporting from the database directly into an Excel sheet. The data can be also viewed in the webpage. The user can view all the data which was entered in the program activity sheet. The user can sort the data according to month and county. The user can also see the data for all counties by selecting the “All Counties” box.
2.4 Control Flow

a.) County Agent:

The county agent must enter his username and password to access the site. If he types in the incorrect data, he is again redirected to the same page. Once he enters the correct data, he can add the program activity sheet to his county alone. He cannot perform any operations for the other counties. He can log off once he is done with his activities.
b.) Admin

The admin can enter the username and password and log on into the site. He can perform various functions from the “welcome” page. He can view the program activity sheets for all the counties. He can log off once he is done with his activities.
2.5 Improvements

a) Speed:

The speed of the website has increased tremendously. In the old website the user took half an hour to enter one program activity sheet, but in the new website the user takes only 10 minutes to enter a program activity sheet. Oracle functions were implemented to perform certain checks, which increased the speed of the application. Oracle Packages and Procedures were used extensively in the project. Procedures are compiled code in the database, which executes much faster than a sql statement executed in the code side.

b) Reduction of Code:

The code was reduced to a large extent. For example the management site contained 22 asp pages. This was reduced to a single page in .NET with user controls. This helps the application to load fast. Dynamic placeholders were used to load the control which is only necessary.

c) Improved User Navigation:

The user interface was improved with the help of the features given by .NET. Datagrids were used extensively to improve the user interface.

d) Storing View State In State Server:

The view state is used by server controls to manage their state across individual web requests. In this application the view state of the page is stored in the state server. This helps the page to load faster, since the size of the page is reduced tremendously. For example, the “Add Program Activity” page was 222 kb without storing the view state in the state server. But after storing the view state the page size was reduced to 122 kb.

e) Reporting Sheets:

The FNPRS reporting side was implemented only in the new site. It saves a lot of time because the report data is directly fed into Excel sheets. The report data can also be viewed in the website.
The site was tested in all counties throughout Kansas. Most of them contained only dial-up connections. The site worked faster than the previous site. The users were able to enter their program activity sheets more quickly and easily.

f) Cascading Style Sheets:

Style Sheets were used in the project. Style Sheets were cascaded to print the program activity sheet directly from the web page.

e) Database Tables Reduction:

The previous application contained as many as 15 tables. It was reduced to eight tables in the new application.
3. Testing

3.1 Microsoft Application Center Test

The application center test enables the gathering of performance information and making capacity decisions on the web applications. Tests were created to simulate several users’ simultaneously requesting pages from the web application. These simulations helped to determine the stability, speed, and responsiveness of the application.

The test result showed that even when the number of users increased, the application was stable. It also proved that there were no HTTP, DNS, or socket errors. There were no connection or timeout errors.

![Microsoft Application Center Test](image-url)
3.2 ANTS Load Test

ANTS Load is a tool which helps to predict a web application’s behavior and performance. It works on all .NET applications. It can generate the scripts automatically. It has the ability to test XML web services. It finds out the fastest and the slowest objects in the web application. This helped to eliminate the slowest unnecessary objects in the web application. It also finds the fastest and the slowest pages in the web application. This helped to make the slowest page a little bit faster.
Steady running 5

8/18/2004 6:32:16 PM

- 5 virtual clients

Summary information by page

This section shows you the average timings for each web page for this test. You can use this to find the fastest, and slowest, pages in your web application.

Note that all timings are in milliseconds

<table>
<thead>
<tr>
<th>#</th>
<th>Page</th>
<th>Time to connect</th>
<th>Time to first byte</th>
<th>Time to last byte</th>
<th>Bytes received</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><a href="http://www.ozenet.ksu.edu/sdev-mk/fnprs_new/Welcome.aspx">http://www.ozenet.ksu.edu/sdev-mk/fnprs_new/Welcome.aspx</a></td>
<td>6</td>
<td>25</td>
<td>111</td>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td><a href="http://www.ozenet.ksu.edu/sdev-mk/fnprs_new/Default.aspx/?id=user_county">http://www.ozenet.ksu.edu/sdev-mk/fnprs_new/Default.aspx/?id=user_county</a> 5</td>
<td>30</td>
<td>120</td>
<td>35</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><a href="http://www.ozenet.ksu.edu/sdev-mk/fnprs_new/Par_Add.aspx?status=PRL">http://www.ozenet.ksu.edu/sdev-mk/fnprs_new/Par_Add.aspx?status=PRL</a></td>
<td>7</td>
<td>77</td>
<td>121</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

Summary information by object

This section shows you the average timings for each object for this test. You can use this to find the fastest, and slowest, objects in your web application.

Note that all timings are in milliseconds

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Time to connect</th>
<th>Time to first byte</th>
<th>Time to last byte</th>
<th>Bytes received</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GET <a href="http://www.ozenet.ksu.edu/sdev-mk/fnprs_new/images/back.jpg">http://www.ozenet.ksu.edu/sdev-mk/fnprs_new/images/back.jpg</a></td>
<td>0</td>
<td>45</td>
<td>8</td>
<td>769</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>GET <a href="http://www.ozenet.ksu.edu/sdev-mk/fnprs_new/images/spacer.gif">http://www.ozenet.ksu.edu/sdev-mk/fnprs_new/images/spacer.gif</a></td>
<td>0</td>
<td>11</td>
<td>32</td>
<td>4200</td>
<td>79</td>
</tr>
<tr>
<td>3</td>
<td>GET <a href="http://www.ozenet.ksu.edu/aspnet_client/system_web/1_1_4722/WebltValidation.js">http://www.ozenet.ksu.edu/aspnet_client/system_web/1_1_4722/WebltValidation.js</a></td>
<td>0</td>
<td>11</td>
<td>23</td>
<td>4722</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>GET <a href="http://www.ozenet.ksu.edu/sdev-mk/fnprs_new/Par_Add.aspx?status=PRL">http://www.ozenet.ksu.edu/sdev-mk/fnprs_new/Par_Add.aspx?status=PRL</a></td>
<td>0</td>
<td>11</td>
<td>23</td>
<td>4722</td>
<td>5</td>
</tr>
</tbody>
</table>

Summary information by metric

This section shows the overall distribution of metrics for your web application. The charts show what proportion of timings fall into which intervals.

Web Request

Time to connect

[Bar chart showing distribution of time to connect]
3.3 Problems Faced

- I was building the application with a fast internet connection. Hence I could not know how the application would work on a dial-up connection until county agents tested it themselves.
- I used JavaScript to open and close a window for the survey forms. The size of the window was fine in my computer, but some of the agents could not view the entire window. The size of the window varied according to the screen settings in the computer.
### 3.4 Snap Shots
### Behavioural Objectives

<table>
<thead>
<tr>
<th>ID</th>
<th>Behavioral Objective</th>
<th>Question</th>
<th>Behavioral Objectives</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>Participants will balance the food they eat with physical activity</td>
<td>After today, how often will you wash your hands before you work with food?</td>
<td>606, 607, 609, 610</td>
<td>Update</td>
</tr>
<tr>
<td>601</td>
<td>Participants will use safe food handling, preparation</td>
<td>After today, how often will you wash your hands before you work with food?</td>
<td>606, 607, 609, 610</td>
<td>Update</td>
</tr>
<tr>
<td>602</td>
<td>Participants will choose and prepare nutritious meals</td>
<td>After today, how often will you wash your hands before you work with food?</td>
<td>606, 607, 609, 610</td>
<td>Update</td>
</tr>
<tr>
<td>603</td>
<td>Participants will choose and prepare nutritious meals</td>
<td>After today, how often will you wash your hands before you work with food?</td>
<td>606, 607, 609, 610</td>
<td>Update</td>
</tr>
<tr>
<td>604</td>
<td>Participants will choose and prepare nutritious meals</td>
<td>After today, how often will you wash your hands before you work with food?</td>
<td>606, 607, 609, 610</td>
<td>Update</td>
</tr>
<tr>
<td>605</td>
<td>Participants will choose and prepare nutritious meals</td>
<td>After today, how often will you wash your hands before you work with food?</td>
<td>606, 607, 609, 610</td>
<td>Update</td>
</tr>
</tbody>
</table>
3.5 References


[3] Oracle Pages (http://www.imano.com/content/oracle9i.aspx)
