Forms Generator

By

Robert A Morrison

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Engineering Technology

University of Cincinnati
College of Applied Science

June 2006

Forms Generator

by

Robert A Morrison

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in Information Engineering Technology

_____	_____
Robert A Morrison	Date
_____	_____
Annu Prabhakar	Date
_____	_____
Patrick C. Kumpf, Ed.D. Interim Department Head	Date

Acknowledgements

Thanks to Rob Wolford for the underling database structure and mapping to the web interface, and extending my knowledge of SQL language.

Thanks to my wife for being patient and understanding of the amount of time dedicated to this project.

Table of Contents

List of Figures

Abstract

Forms Generator

The application provides an easy method to collect information over the internet. Regulations, wording and the information required for the grants funding is constantly changing. The tool makes creating and maintaining forms easy by letting an administrator of a funding program to create and maintain forms. Forms are created by entering a series of questions and the type of data expected for the answers. The questions the can be grouped and ordered on a page and sequenced over a multiple pages. The user's answers can control the flow of the application questions, reducing the burden on the end user answering unnecessary questions. The resultant application then can be printed for a hard copy and signatures. The data also can be extracted into a data table for further processing.

Forms Generator

Problem

Federal Home Loan Bank requires an efficient a way to collect data for grant and loan applications for housing, keep up with changing information requirements, and increase the availability of services to the public. Regulation changes require the wording of grant and loan application forms to change periodically. Not only will the wording change, but questions may need to be removed or added. The housing department wants to be more responsive to the changes. One of the more recent offers promised was loan aid to hurricane Katrina victims moving into the fifth banking district. While the offer was announced on the web site, the only way to apply was to download a word document and fill out the form. People applying for aid are required to have Microsoft's word processor. While word is useful for gathering information, the application must still be manually transcribed into a database.

Solution

The solution provides a method to create and modify forms for grant or loan applications, making the resulting applications available for end users to fill out over the Internet. Also, historical application information with the old system requires restoration of the original software in order to view old applications. By storing the question for the application and applicant data in a database, the system will also be able to provide historical application information. Through security, the software will also allow users different rolls in filling out the forms, allowing for a work flow to be embedded in the electronic document. The user's answers can control the flow of the application questions, reducing the burden on the end user by eliminating unnecessary questions.

User Profiles

The first user is anyone filling out an application. Given the user will not be familiar with applying for grants or loans for a housing project, the generated forms must be able to support the user with help, validate fields and provide popup text for special terms.

The next user profiles work at a member banks and is knowledgeable of the basic process. Their role is to approve the member's bank participation in the funding.

. The third user profile will use the application to extract the grant and loan applications for processing. The extraction process will make the data consumable by other systems and create PDF files. Reference: 1

The fourth user profile creates forms to collect application data.

Technical Elements

Federal Home Loan Bank standard operating environment uses Microsoft products. The forms generator project will use IIS version 6 on Windows 2003 Enterprise server. The web application will use .NET web framework with the application written in C#. The screens for the grant and loan application will be driven from questions stored in a Microsoft's SQL 2000 database server, with the resultant answers stored in the database. Submission of a grant or loan application will use Microsoft's SMTP to notify the next user in the processing of an application.

Design

In order to distinguish loan and grant applicants, the user must complete a registration process creating a unique user id and setting a password. The user will then be notified of the user ID and password by email. Then the applicant can answer questions on a form.

The size of the forms dictates that the forms have several layers of segmentation. A form will consist of sections. Each of the section titles will make up the left navigation of the menu. Within a section is a list of pages. A page corresponds to a single web page. Each page has a group of questions. One question on a page may be designated to control the flow to the next page. Each page may be associated with help text to help the user. The term can be automatically substituted into the web page allowing the applicant to view the definition without leaving the page. Answers to the questions can be collected with the following web controls.

- Radio Buttons
- Text Boxes
- Text Area
- Drop Down List Box
- Check Box

A data type is assigned to the answers. The data types available are STRING, INTEGER, MONEY, and DATETIME. All fields can have a required field validation. Text files will have more complex validations. The field will be validated so that conversion to data type chosen for the answer can be performed. Numbers can have an additional number range validation. String fields can validate the input using regular expressions. The

administrator tool will provide validations for some standard formats such as phone numbers, zip codes, email addresses, and social security numbers.

Database Design

The forms will be stored in Microsoft's SQL 2000 database. The database architecture fits the graphical user interface. A form has a hierarchical organization. Each form is divided into sections. The sections table in the database represents the left navigation displayed providing an entry point into the form. Each section has a list of pages. The page table determines the order of the displayed web pages. The flow of the pages can be controlled by selected answers. The order is controlled by selecting an answer type with only two options and also filling in the false link to the next page. One page may have one or more groups. The title of the group is displayed in bold text highlighting the group of questions. Groups have one or more questions. Each answer is stored in a row requiring an extraction process to take the answer row format into a single row for each application. The many layers of origination allow the form administrator to quickly organize a form.

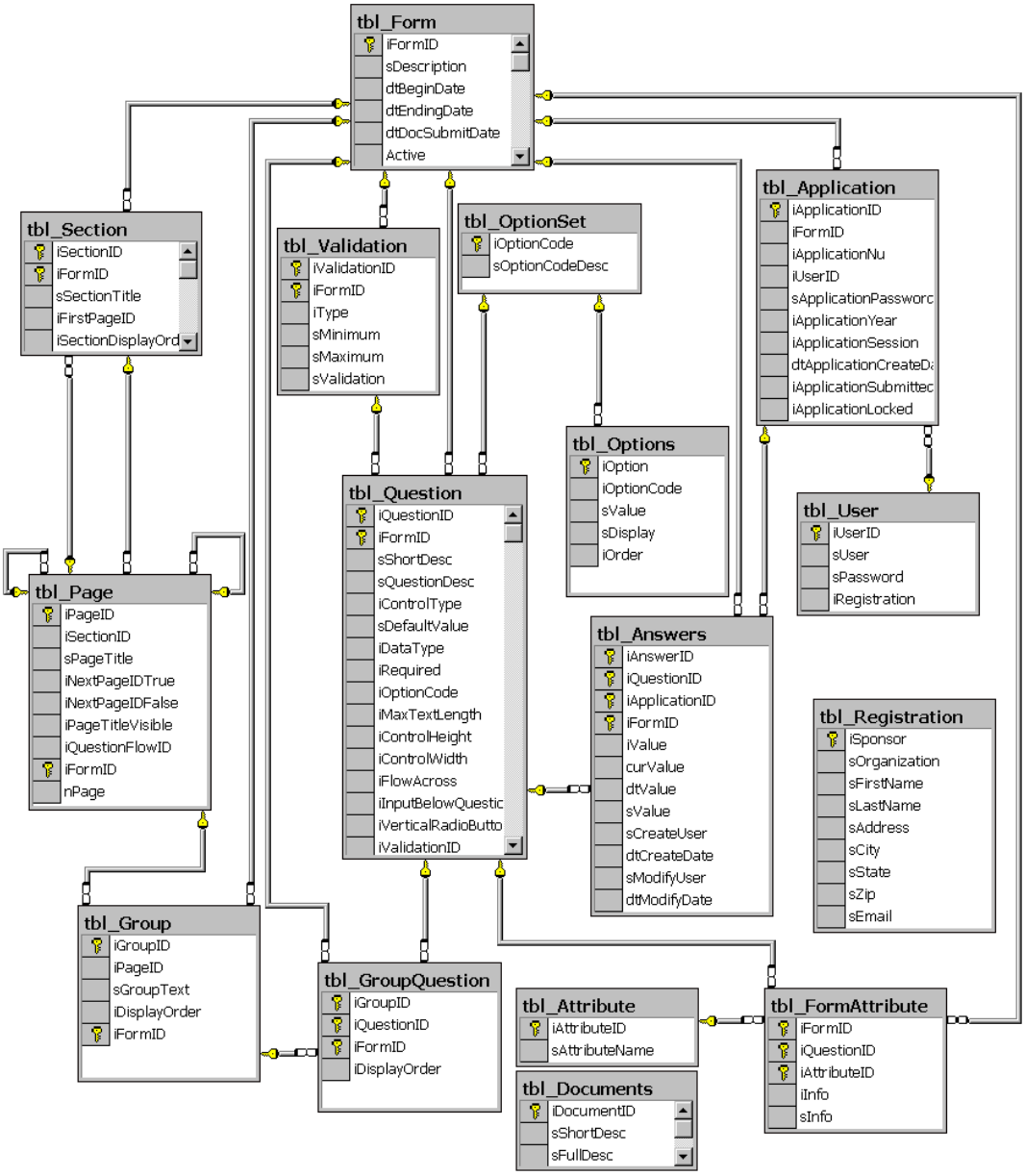


Figure 1 Database Diagram

In Figure 2 contains the database diagram. Sections correspond to left navigation buttons for an application. One unusual aspect of the database is the pages are link listed together to provide a method to navigate page to page depending on answers provided. Since the number of pages in a form is small the performance impact should be small. Note the relation is kept from most tables to the form table. This allows questions, sections, groups, pages and validations to be created and linked at a later time. The most

of the tables use merged keys with a reference to the form table to ease the coping of a form to a new form, by creating a duplicate record with the all of the fields duplicated except for the new form reference. The main structure of the database is hierarchical with each table representing level. The top level starts with the form. Next a form is broken into sections. The sections appear in the left navigation panel of the generated web page. Under the sections are pages. The page control flow from screen to screen. Pages contain groups. The group title accents a set of questions. Last groups contain a set of questions.

The validation table is used to create checks for ranges for most data types or regular expressions for more complex data types.

Work Flow

The completion of an application causes an email to be sent to the next user in the application process. E-mail messages consist of prepared messages with variables indicated with ${xyz} that can be substituted with information from the form to create a customized E-mail response.

A sponsor is a person who fills out a form. Upon completion the sponsor submits the application to a member bank. Then a member bank will certify its responsibilities for the project then submit the application to Federal Home Loan Bank.

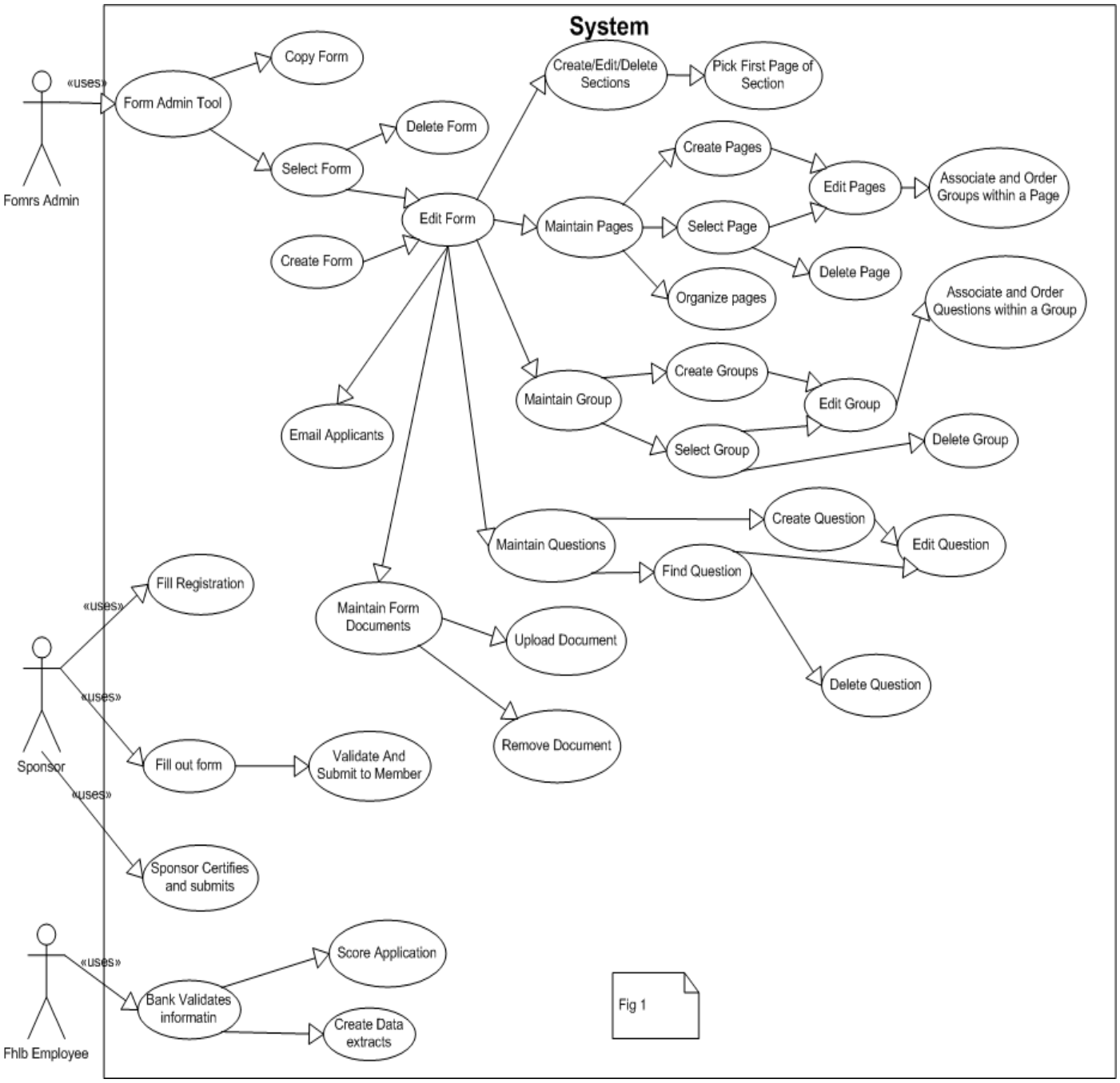
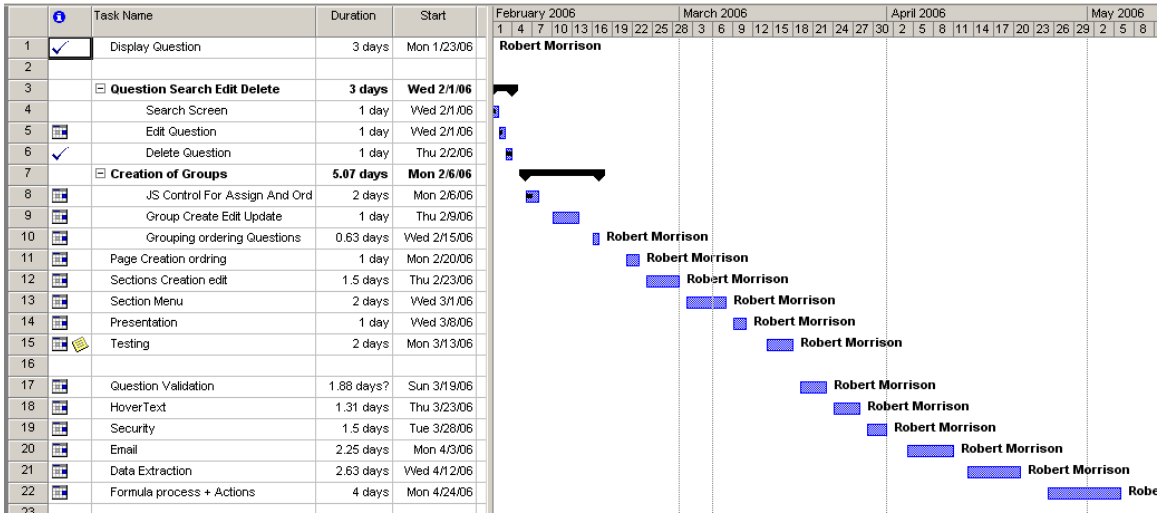


Figure 2 Use Case Diagram

In the above, Figure 1 diagrams the basic use cases for the forms generator. The use case for filling out a form is regulated by the form created by the admin tool. A created form would need its own use case diagram depending on the grant application. The forms admin organizes a series of questions into groups, pages and sections to create a form. The menu flow for creating a

form allows the user to enter at any point start working. Sections, pages, groups and questions can be created without the under lying elements. The forms administrator then can order the questions, groups, pages and sections to create the form for publication.

Schedule



	Task Name	Duration	Start
1	Display Question	3 days	Mon 1/23/06
2			
3	**Question Search Edit Delete**	**3 days**	**Wed 2/1/06**
4	Search Screen	1 day	Wed 2/1/06
5	Edit Question	1 day	Wed 2/1/06
6	Delete Question	1 day	Thu 2/2/06
7	**Creation of Groups**	**5.07 days**	**Mon 2/6/06**
8	JS Control For Assign And Ord	2 days	Mon 2/6/06
9	Group Create Edit Update	1 day	Thu 2/9/06
10	Grouping ordering Questions	0.63 days	Wed 2/15/06
11	Page Creation ordring	1 day	Mon 2/20/06
12	Sections Creation edit	1.5 days	Thu 2/23/06
13	Section Menu	2 days	Wed 3/1/06
14	Presentation	1 day	Wed 3/8/06
15	Testing	2 days	Mon 3/13/06
16			
17	Question Validation	1.88 days?	Sun 3/19/06
18	HoverText	1.31 days	Thu 3/23/06
19	Security	1.5 days	Tue 3/28/06
20	Email	2.25 days	Mon 4/3/06
21	Data Extraction	2.63 days	Wed 4/12/06
22	Formula process + Actions	4 days	Mon 4/24/06

Figure 3 Project Schedule

The project is capable of creating and formatting question to be represented over the internet. Validation for the input files offers a strong range of capabilities. The question can be mark as required. The end user filling out the form must answer the question. The data being entered can be restricted to integers, decimal, float, or a date. The entered data also can be range check. Input strings also can be validated against a regular expression and the error message can be customized for the regular expression to aid the user. The validation messages appear as tool tips if the user hovers over the input field.

I have completed the question edit, create search and edit screens and the code is functional. Creation of groups and ordering questions within groups is competed.

Creating pages and placing groups on a page and creating a display order is also fully functional. The user interface for page editing is weak. Representation of the page order is difficult because it lacks the ability to properly represent Question flow. The admin tool can create forms however issues of whether a form can be deleted needs to be answered. Once a form has answers for the questions deletion of the form removes grant and loan application information. The section edit and create can order the sections.

Budget

Development Environment

Development	Price	Personal Cost
MSDN Universal edition Includes 2003 .NET Visual Studio. SQL 2000 Developer	2,799	0.00
Dell Laptop • 3.0 Ghz Pentium IV • 1.0 gb of memory • 40 gb of hard disk • 100mbit Ethernet • Wireless • Windows XP Professional	2,400	0.00
Labor	8,224	
Total	$13,423	$0.00

Reference: 3

The development equipment is my laptop with the MSDN provided by the company. The development environment also requires 0 dollars. The company has purchased a universal subscription to Microsoft's MSDN providing the necessary

development tools and equipment. My time for the project is outside of the company time therefore the labor is free for the company.

Production Environment

Equipment	Cost	Actual
2.8 Ghz Dual Xeon	$2,129.00	0
120MB disk RAID SATA 4 x 40GB	0.00	0
1gigabit Ethernet.	0.00	0
4GB ECC memory	400.00	0
Windows 2003 Web	399.00	0
SQL Server Standard Edition	9998.00	0
Total	$11926.00	0

The production equipment already exists and is under minimal load. Actual cost to the company is 0 dollars. Currently the labor portion is being performed outside of company time. Current production environment utilization is low and has adequate capacity to handle the software.

Prototype

The prototype can render questions from the database. The user can enter the question and the data is stored in the database. The administration tool can edit questions which are reflected in the rendered form.

Testing

- Creating a form
- Create, Edit, Delete Questions
- Create, Edit, Delete Groups
- Create, Edit, Delete Pages
- Create, Edit, Delete Sections
- Ordering sections
- Putting groups on pages and setting the order of display.
- Grouping and ordering the display of the questions.
- Placing and Grouping questions on a page.
- Ordering Pages within a section and setting branching functionality
- Check the different types Question inputs Text box, Text Area, Radio Buttons, Check boxes, and Drop down list boxes.
- Check the different data types allowed Integer Money date, and String
- Check Validation for the question.
- Check the registration process.
- Check a Questionnaire created by the form admin tool.

The following test matrix is use to help check how the program renders controls and effects visual appearance.

Test Matrix

Options Set	Controls	Radio Button	Drop Down Box	Check Box	Text Box
Question flow across Input Below Radio Direction					
Question flow across Input Below Radio Direction	X				
Question flow across Input Below Radio Direction	X				
Question flow across Input Below Radio Direction	X X				
Question flow across Input Below Radio Direction	X				
Question flow across Input Below Radio Direction	X X				
Question flow across Input Below Radio Direction	X X				
Question flow across Input Below Radio Direction	X X X				

Deliverables

Provide an easy to use forms generator to gather information over the web.

1.0 Forms admin tool

1.1 Create Edit Forms

1.2 Create Edit Questions

1.3 Group and order Questions.

1.4 Organizing Questions

1.5 Provide Question Flow

1.6 Set Validation on questions.

1.7 Create help information.

1.8 Create Popup hints.

1.9 Create edit and order Sections.

2.0 Web presence for the generated forms.

2.1 The web site will provide a registration screen.

2.2 Registration will email User ID and Passwords to the registered user.

2.3 Registered user will be able to login and make multiple applications.

2.4 Forms will have the ability to validate data.

2.5 Pop up tips can be provided for highlighted key words.

3.0 Create an example form show various capabilities of the form generator.

4.0 Provide Registration for user applications

5.0 Setup an email system to demonstrate operational capabilities.

6.0 Security

6.1 Application registration

6.2 Member Bank Employee login

6.3 FHLB Employee login

7.0 Extraction tool to flatten the data

7.1 Each answer is stored in an individual row requiring a pivot for extract for use by other databases.

8.0 Deployment instructions.

Create a user manual for the Admin tool

Proof of Design

The user can easily create a form to present over the internet. By composing series of question and placing them on a page to gather information. The design allows creation of very large forms by breaking the form into smaller more manageable pieces. The parts of the form can be manipulated as larger set of questions. The application to create the forms has some quick navigation features to move through the form to quickly find next pages, groups or questions.

The architecture of the interface allows the public user of the form natural feel from the form. The form sections are listed on the left navigation of the page. The header reflects the forms name the user is filling out. Validation can be control what the user enters into the web page reducing the number of mistakes filling out a form. All control types except check box can be marked as required by the creators of the form. Text boxes can be setup to check the type of data and if the enter value is in a range. For string inputs such as email a regular expression can be used. The flow control of the pages prevents the end user from seeing unnecessary pages, reducing burden on the end user.

The registration process works to identify who entered provided data form the form and known working email address. The user is emailed the password before they are allowed to fill out form.

The end form is rendered as a PDF file for printing and hard copy version of the created on line applications. Also the program retains all of the information entered by the user. The user can also make multiple applications, for their projects.

Performance testing reveled the application has a large dependence on the database access for performance so a radio button the form has been activated was added

to the forms page. This radio button once set can not be turned off and indicates to the software the form will not be changed allowing the information for the form to be cached. Doing this nearly tripled the speed of the application question page to nearly 500 TPS on a dual xenon system, also dropped the load on the database by nearly 70 percent.

Conclusion

The project is successful at delivering the basic objectives. The interface delivers an easy to use mechanism to create forms to be presented over the internet.

Improvements on the interface for organizing pages into sections should have used a method similar to the question grouping without allowing the ordering mechanism . Also representing a list of pages with decisions controlling the flow of pages was difficult to represent.

At the Tech Expo several attendees had useful features which can be added at a future date to the application. One suggestion is being able to link forms together. Certain criteria on forms being filled out automatically require other forms. Overall I have learned allot about how .NET and C# web development

Appendix

The entire purpose of the program is to create forms to collect information from grant applicants through a web browser. The structure of the form is a tree. The form consists of sections each section has a list of pages. Each page contains a group of questions. In order to construct a form, the user must lay out a plan of how the form is to be divided up into pieces. First the administrator of the form creates the necessary questions. Then the user organizes the questions into groups. The order to be presented to the applicant can be changed. Answers for questions can be restricted to a set of answer using radio buttons or drop down list box. Text answers can be restricted to money, dates, integers or strings like email addresses. Also the field can be marked as required. Answers can be restricted to a range for numbers or dates. Pages are ordered with a link list and the sequence of pages can be controlled through binary type questions. After completing the form and the content has been verified by the person creating the form go back to the form screen and select the activate radio button and save. The form will now be active. The last step is to place a link on the web page to navigate to the form.

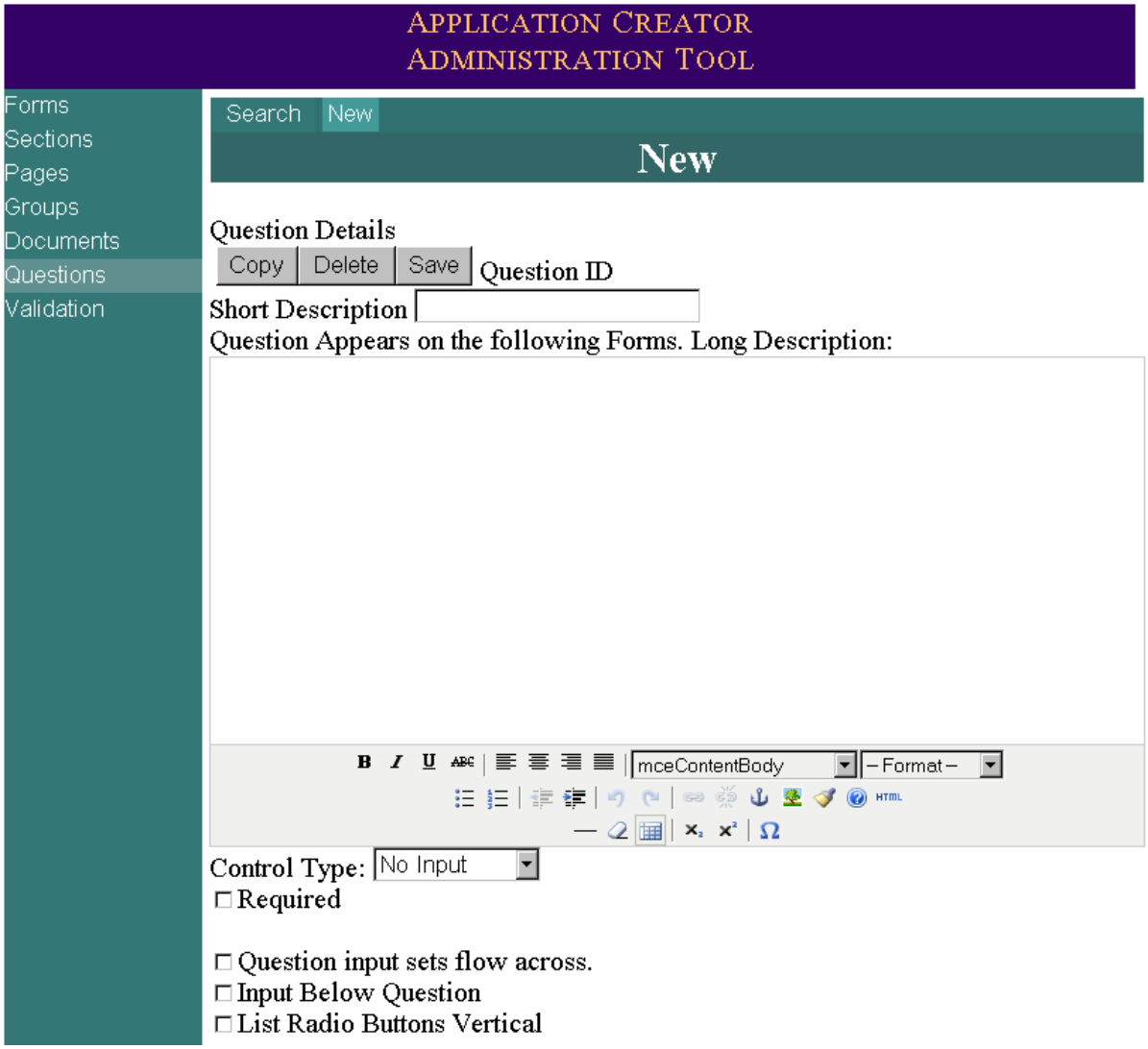


Figure 4 New Question Screen

The short description is a label used for identifying questions in the question search or listing the questions for the group editing. The long description appears on the web form posing a question to the applicant. The control type of no control defined is used when the user wants to place instructions on the screen without soliciting an answer. Radio button and drop down list boxes restrict answers to a known set of answers. The text box control is for retrieving single line of data from the user and can apply data validation to

the given answer. The text area control type is used to solicit essay type answers. Input below the question if checked will place the control on the next line below the question. The break between controls only affects the radio buttons, if the check the radio buttons will display vertically.

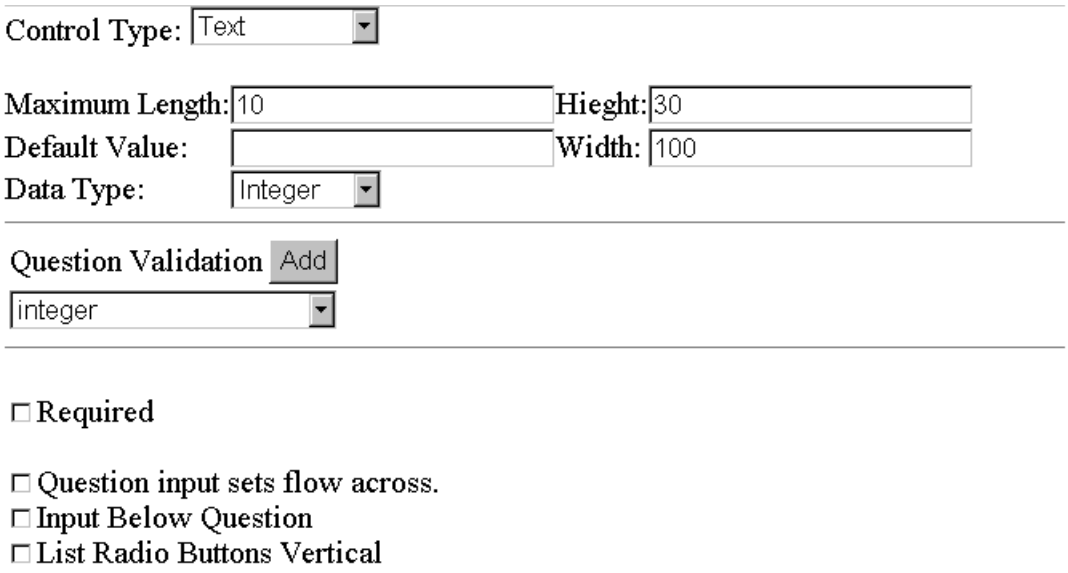


Figure 5 Question Detail Screen

The textbox control box has more options, which appear when selected. The maximum length controls the number of characters allowed in the field. The default value is the initial displayed value. The height and width control the dimensions of the text box control. The data type determines how the data is verified. The available data types are integer, currency, date and string. The program automatically will enforce validation according to data type. The user has the option of creating additional validations. The existing validations appear in the drop down list box.

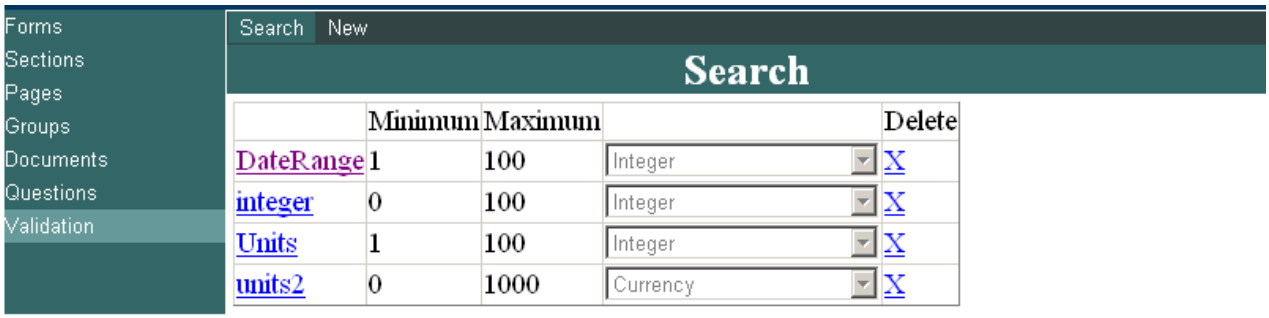


Figure 6 Validation List Screen

Validation:

The validation screen shows a list of validation for the form. The search validation screen allows the user to click on the link in the first column to edit the validation. Clicking the X in the last column will result in the validation to be removed.

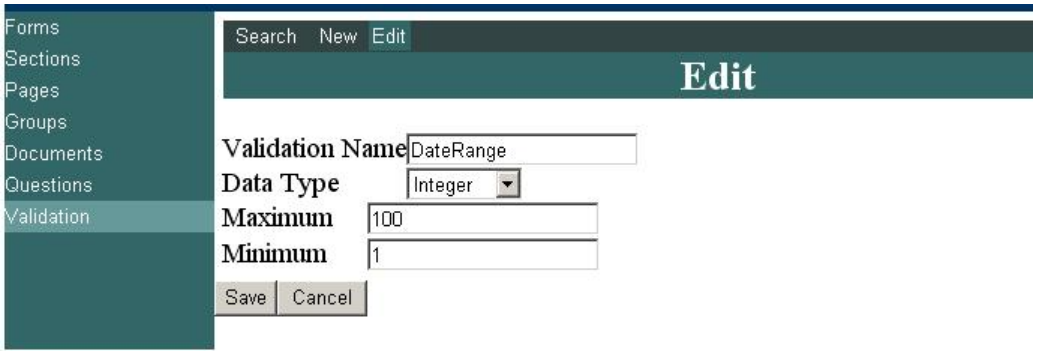


Figure 7 Validation Edit Screen

On the edit validation screen the user sets the name of the validation and the data type. After the data type is set the user can set a range of valid values.

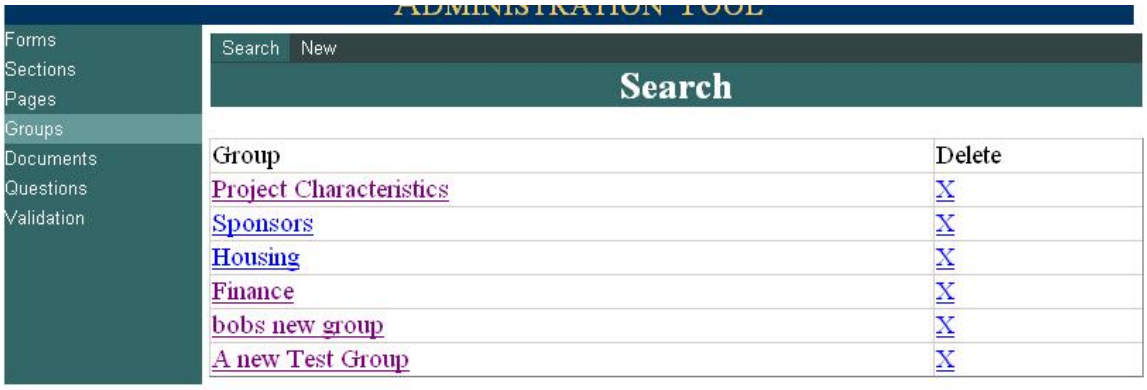


Figure 8 Group List Screen

The group list shows the list of groups of questions. The X under delete will delete the group without removing the questions contained within the group. Clicking on the group name link will take the user to the group edit screen. The new menu item will take the edit screen to create a new group.

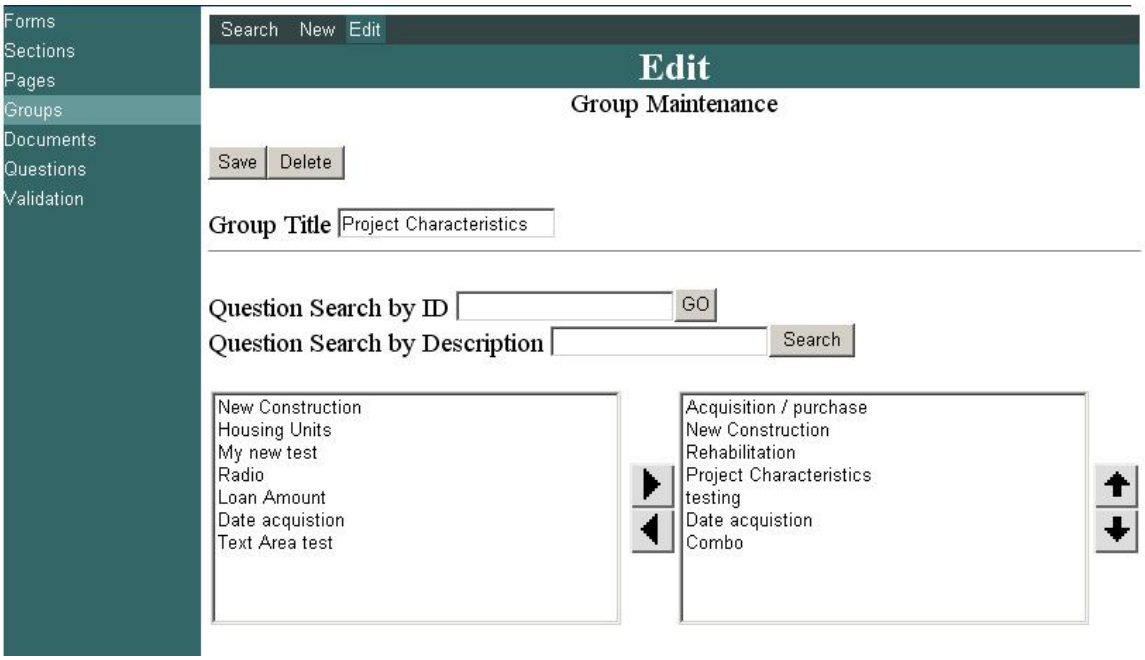


Figure 9 Group Edit Screen

The group edit screen sets the title of the group which will appear on the form in a bold font. The two lower list boxes contain questions. The list box on the left contains questions available to be placed in the group. The user can use the question search to reduce the number of questions in the left list box. In order to place a question in a group highlight the question on the left and click on the right arrow. To remove a question mark it in the right list box and click the arrow pointing to the left. To set the order of the questions mark the question and use the up or down arrows to move the question. The save button will store the changes to the group.

Figure 10 Page List Screen

The page list shows a list of pages with in a form. Clicking the X will delete the page but will not remove groups or question related to the page.

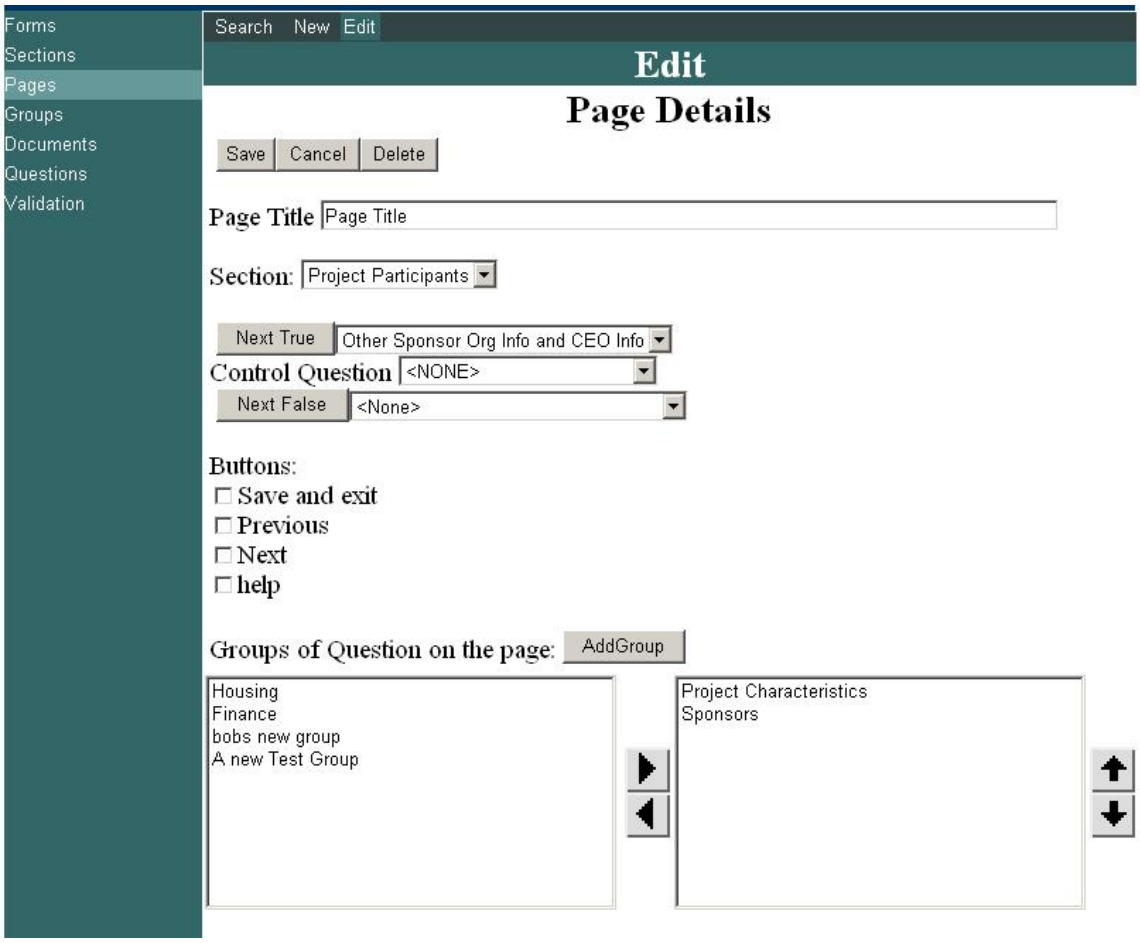


Figure 11 Edit Page Screen

The page edit associates a page with a section of the form. The page title names the page. The next true button will go to the edit page for the next page. The drop down list box next to the next true button sets the next page. If the control question is not set, the next true page is always the next page. If the control question is set and the question is true, the next true question will be the next page. If the question is false, the next false page will be the next page. The button to the left of the drop down list box with the text “next false” will bring up the next page if the answer is false. The check boxes controls if the buttons appear on the form. At the bottom there are

two list boxes. The list on the left is the available groups. Marking the group on the left and clicking the right arrow places the group into the page. Marking the group on the right list box and clicking the left arrow will remove the group from the page. The order the groups appear on the page can be changed by marking the group and moving the group using the up or down arrows. Clicking the save button commits the changes. Clicking the cancel will take the screen back to the page list. If the page next is set to <none> the next page will be the first page in the next section.

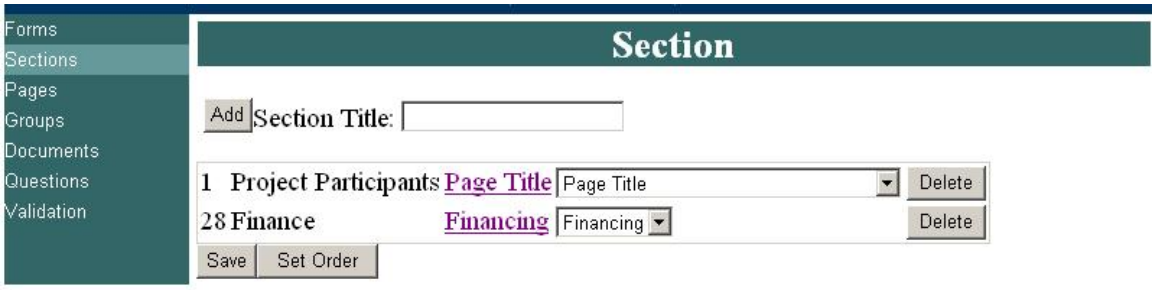


Figure 12 Section Add / List Screen

The section table breaks the form up into manageable pieces. A form must have at least on section. To add a section fill in the title and click the add button. To delete a section, click the delete button next to the section you want to delete. The page link can be followed to the page edit for that page. To set the first page for the section use the drop down list box in the section you want to change and press save. To change the order of the section press set order.

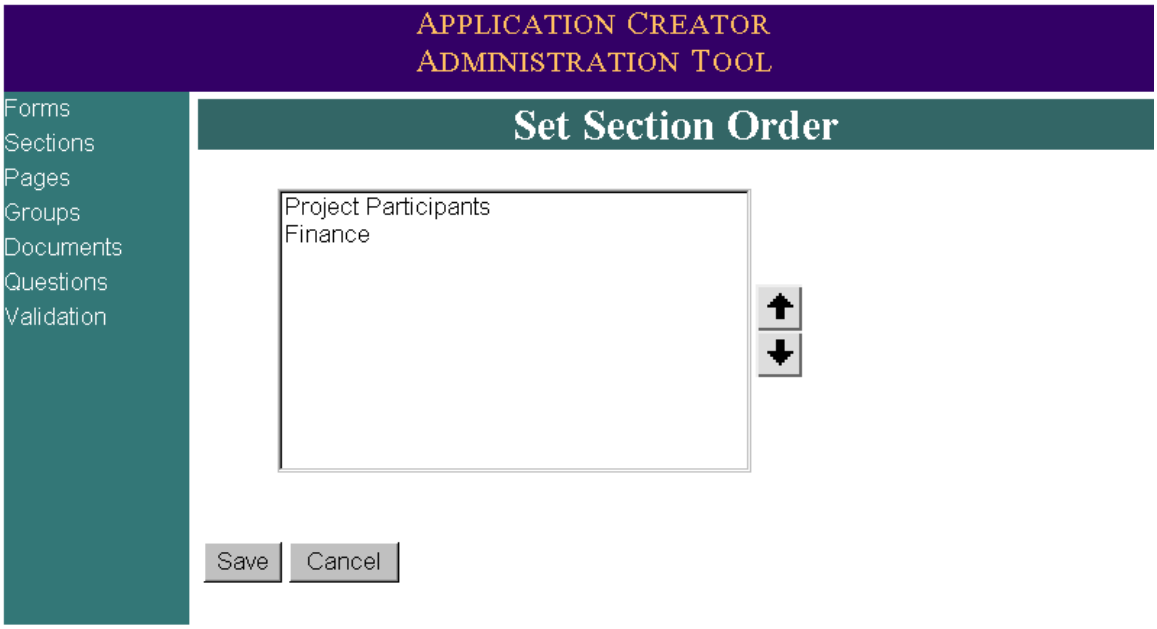


Figure 13 Set Section Order Screen

On the section reorder screen select the section and use the up and down arrows to change the order of the sections. Click the save button to commit the changes. Click the cancel button to go back to the section menu.

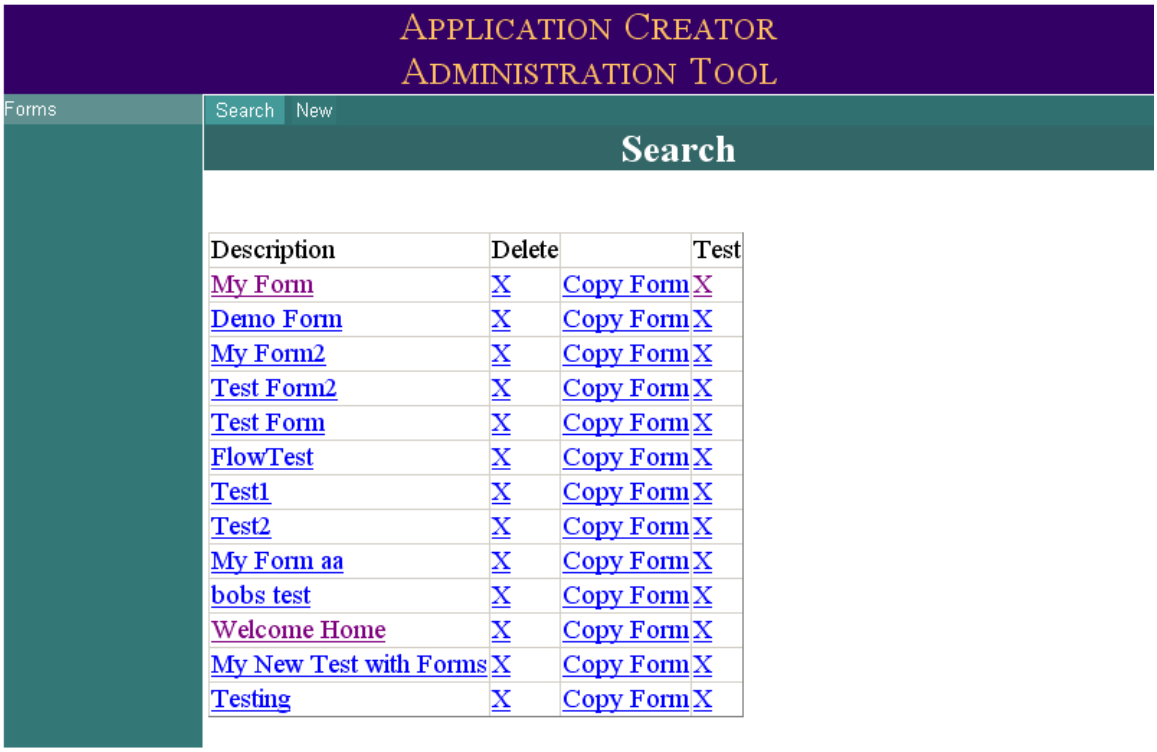


Figure 14 Form List Screen

The administration tool will provide a list of forms. In web screen figure 14, the list links to each individual form. A link is provided to copy the form. Once a form is made public it should not be changed. In order to change a previous form the administrator will copy the old form to which the modifications are made. The copy will reset the active flag and allow modification of the form. The list also provides for a way to delete forms. Forms which have been published and have non test applications for the form can not be deleted. The test column provides a link for the designer to test the form.

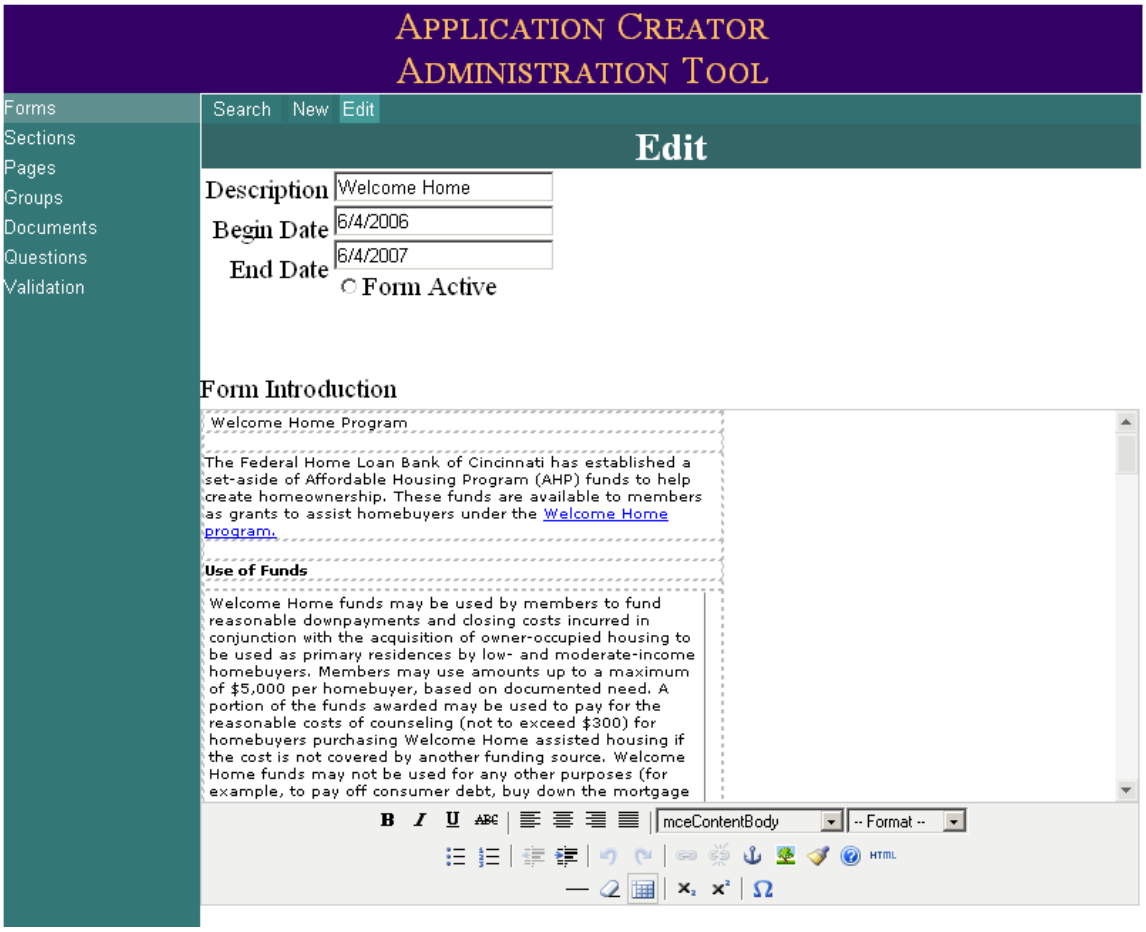


Figure 15 Edit Form Screen

Following the link for the form in figure 15, the begin date sets the first day the form is available for the public. The web site will remove the form from the web site after the end date. From the form edit screen the form administrator can navigate to items in the left navigation menu. The Form Active radio button once set cannot be reset. The button indicates the form is ready for web access. Activation turns on caching for the form reducing the burden on the database server. The form introduction allows the form administrator to create an introduction page describing the application to the public.

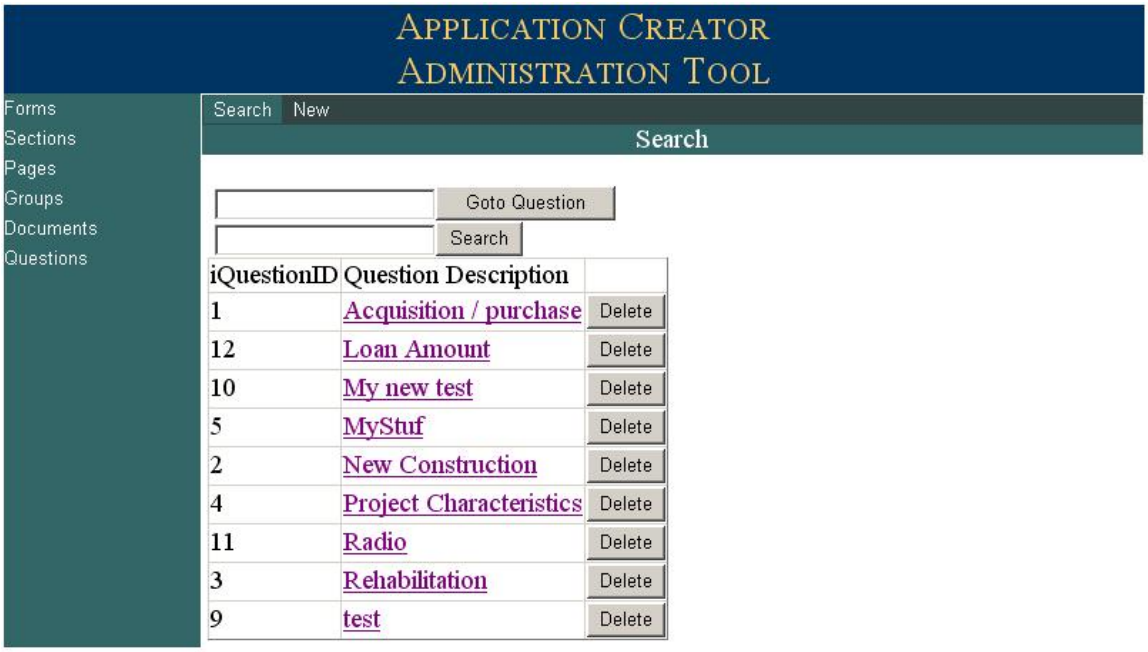


Figure 16 Question List Screen

All lists share a general pattern as in Figure 16. If the list of objects is large a search mechanism is provided. The result is a list of links to the underlying object with the option to delete the object. The program is a hierarchal tree of objects with the form representing the top-level object. Each form can consist of several sections. Sections render navigation buttons at the left of the form as in figure 17, providing a shortcut entry into long forms. Each section has a list of pages. A page can have one or more groups of questions. The group name appears as a bolded title. In figure 13 Project Characteristics is a group title.

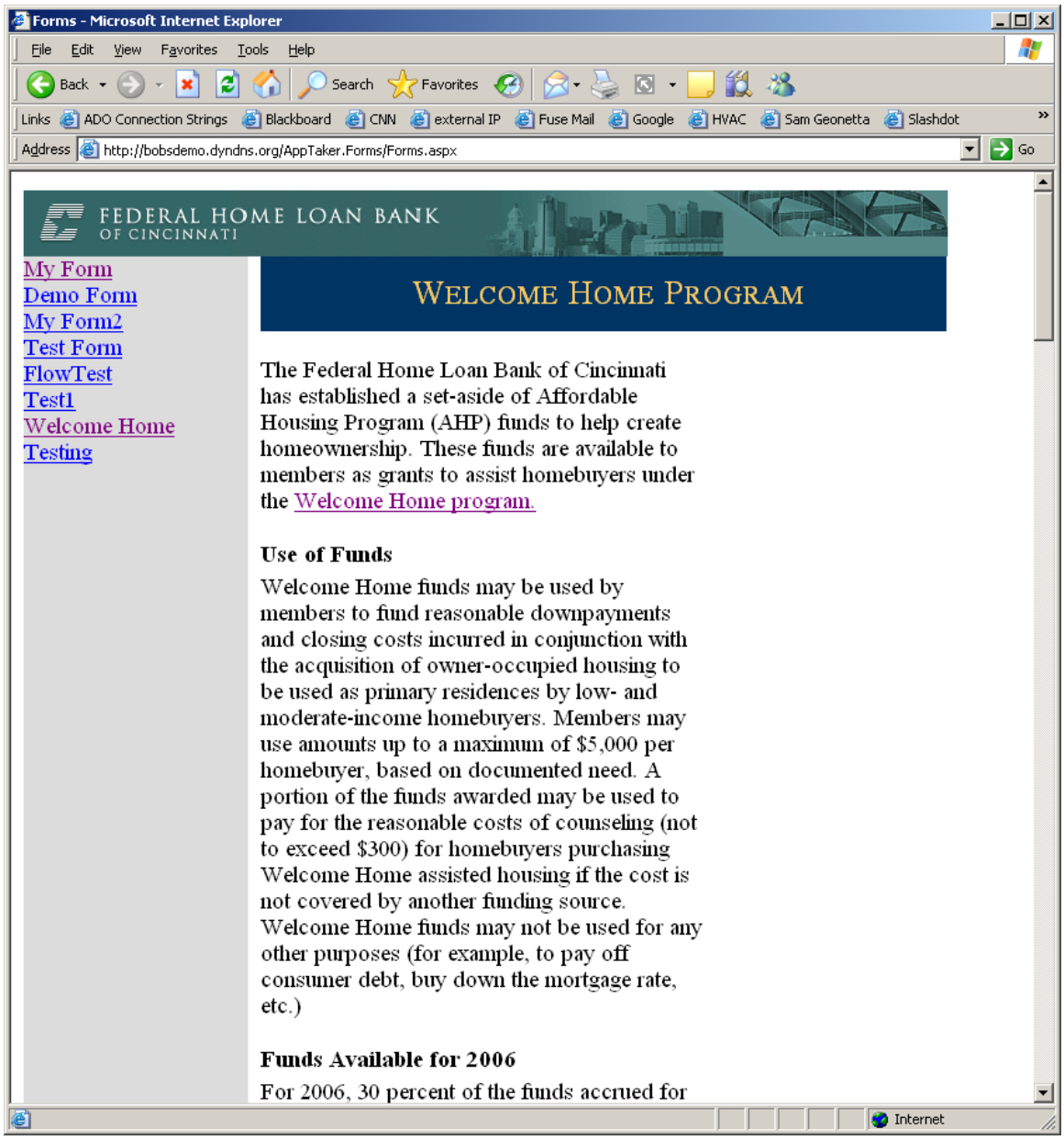


Figure 17 Form Introduction

The form administration tool can be used to create an introduction to the form. The above was created by edit the introduction description to the form. The editor allows enough flexibility to create an HTML formatted introduction the online form.

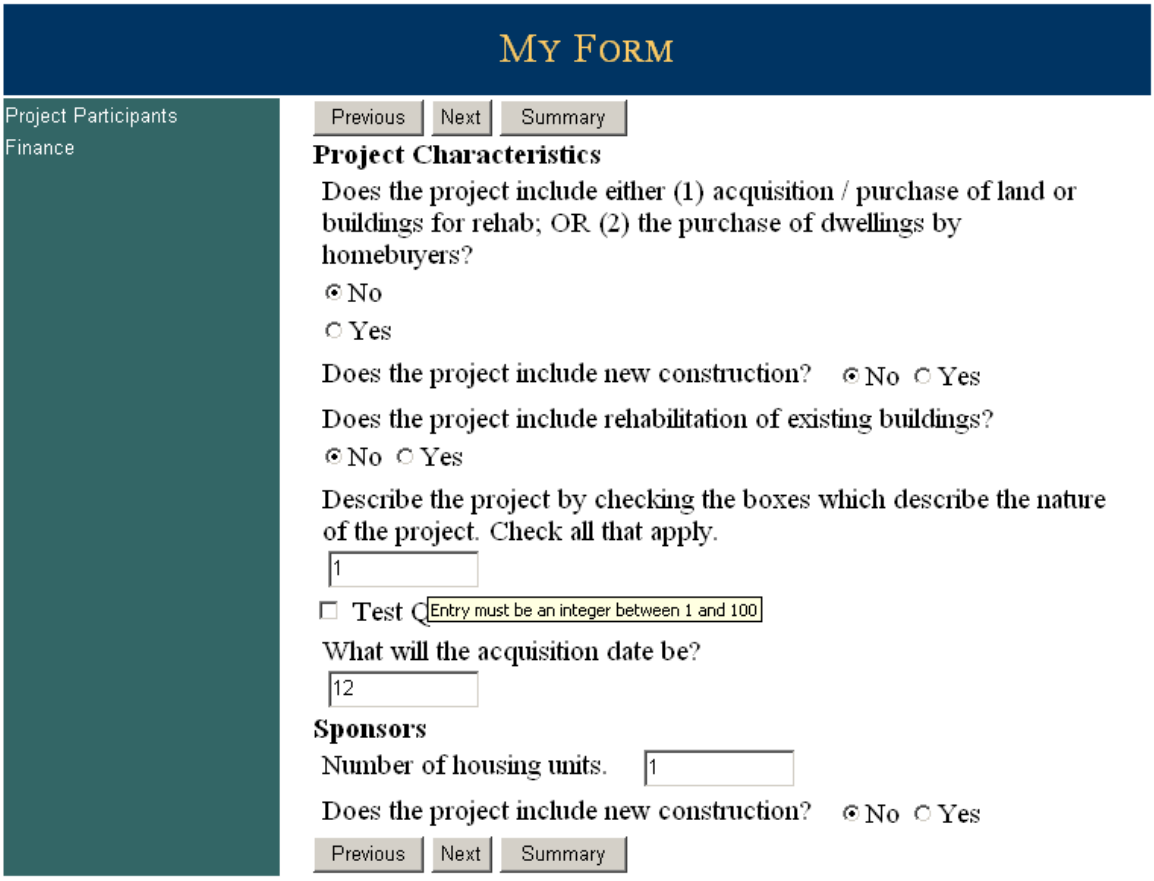
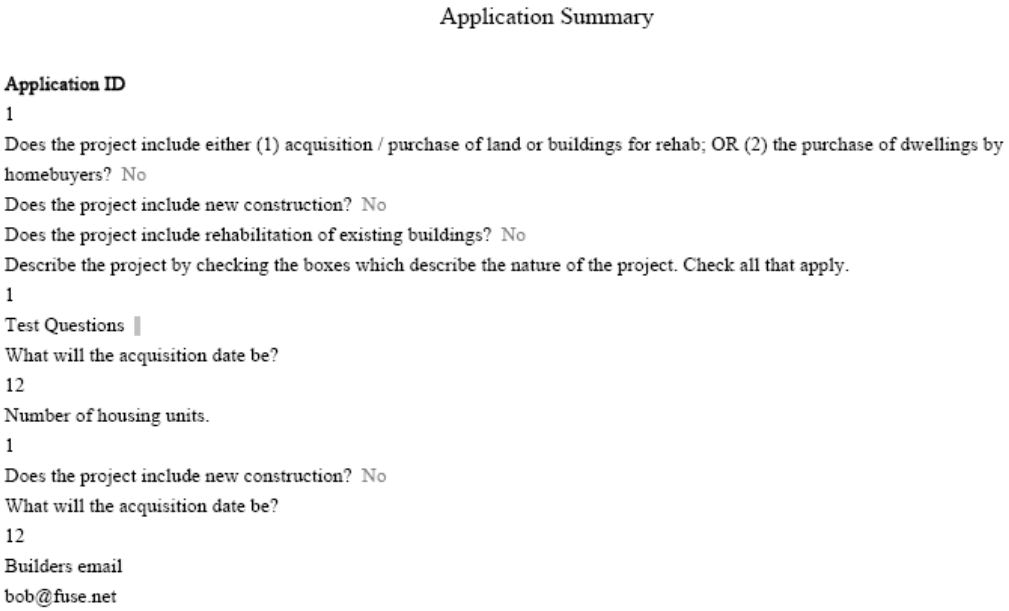


Figure 18 Generated Form

Figure 17 shows a form created by the administration tool. Note the bold titles are the group headings. The buttons next and previous navigate the pages. The summary button is used to download and print a hard copy of the form. If the user hovers over a control with validation a pop up tool tip will appear helping the user to fill in the form with the appropriate data.


Application Summary

Application ID
1
Does the project include either (1) acquisition / purchase of land or buildings for rehab; OR (2) the purchase of dwellings by homebuyers? No
Does the project include new construction? No
Does the project include rehabilitation of existing buildings? No
Describe the project by checking the boxes which describe the nature of the project. Check all that apply.
1
Test Questions
What will the acquisition date be?
12
Number of housing units.
1
Does the project include new construction? No
What will the acquisition date be?
12
Builders email
bob@fuse.net

Figure 19 PDF Output of a Simple Form

Figure 19 shows the question edit page. Here the user can compose their question. The control for the edit box is from http://www.moxiecode.com/ with a LGPL license. Reference: 4. The product is for internal use, which is within the licensing of the MCE tiny editor agreement. The data queried by the form can be collected using a variety of web controls such as a text box, text area, radio buttons, check boxes, or drop down list boxes. A method for validating data will be provided for the question. Basic validation types include numbers, ranges, and strings with specific formations such as phone numbers.

Questions can also provide primitive positioning of the control with relation to the question, by providing breaks after a question or after a control. Break between controls is only for the radio buttons, which will cause the buttons to be stacked vertically.

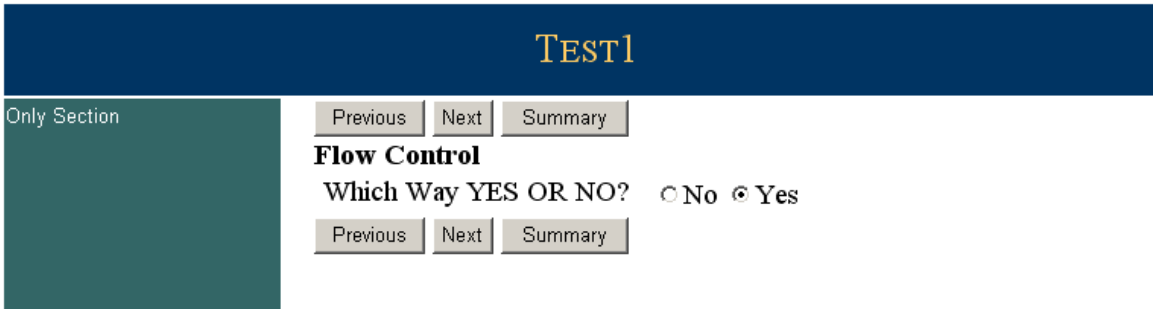


Figure 20 Question Flow

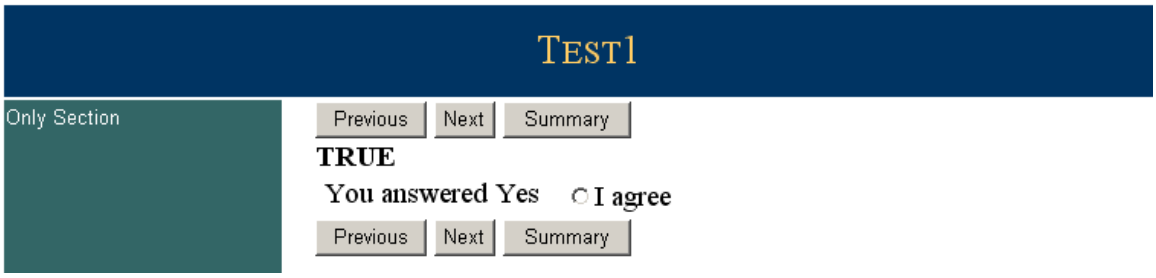


Figure 21 Flow True

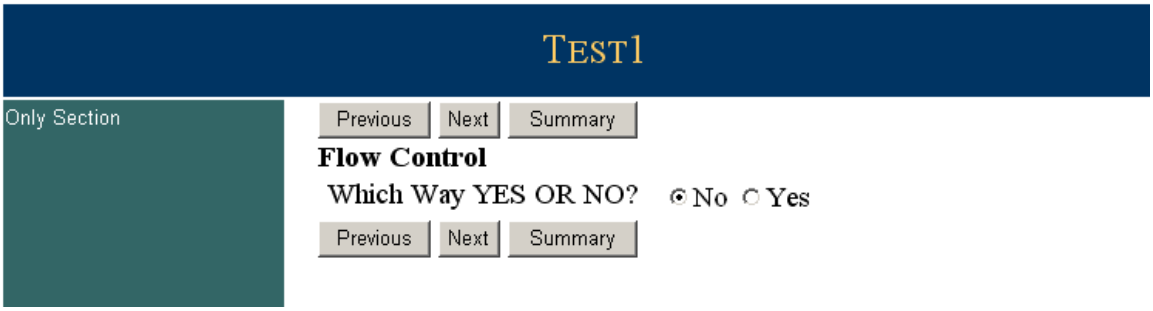


Figure 22 Question Flow 2

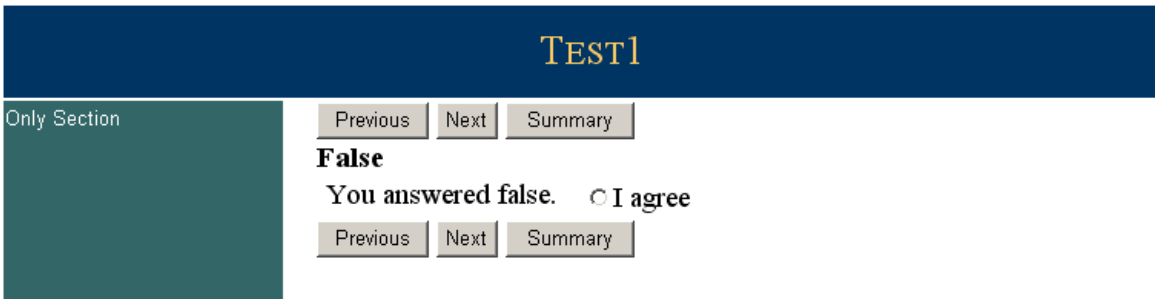


Figure 23 Flow False

In figures 20-23 show an example of how a question can control the flow of the question asked.

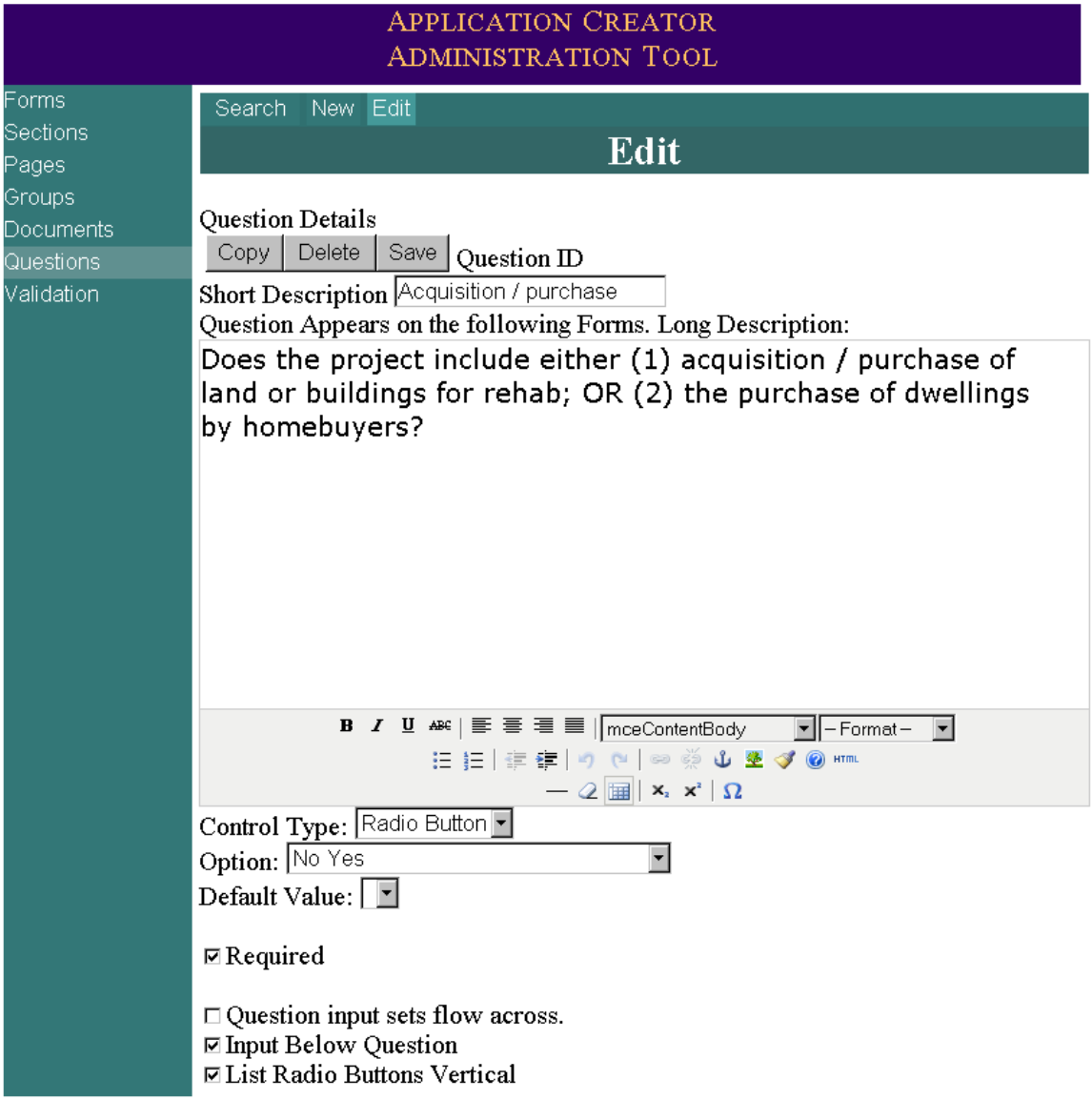


Figure 24 Question Edit Screen

NOTES

Install

1. Prerequisites
    a. Microsoft's SQL Server 2000 Service Pack 4 installed.
    b. Microsoft's IIS Server with .NET 1.1 installed.
2. Create the database AppTaker2 on the SQL server.
3. Restore the database backup file AppTaker2.bak to the database.
4. Create User for the database and assign the user as DB owner of the AppTaker2 database. The application uses integrated security through the aspnet user in the default web.config files for the web applications.
5. Extract from the zip file the three directories AppTaker, AppTaker.Forms and AppTaker.Controls_1 into any directory accessible by the web server.
6. In the IIS administration tool create a virtual directory for AppTaker and point the directory to the same path used for extraction. Then create an application name for AppTaker.
7. In the IIS administration tool create a virtual directory for AppTaker.Forms and point the directory to the same path used for extraction. Then create an application name for AppTaker.Forms.
8. The web application for AppTaker.Forms is public and needs to be setup with anonymous access.
9. The web application for AppTaker is the administration tool. The application uses IIS integrated security to authenticate users. So setup a group of Form administrators and give the group access to the AppTaker

application. Then add any person which is allowed to create and edit forms.

10. Modify the database connection string to point to the required database.
11. It is up to the end user to provide entry links in to the administration form. The administration start page is FormSelect.aspx.
12. The initial entry into a create form is left up the sites designers. As a help the public site will provide a list of available forms. The site developers can grab this link to provide a seamless integration into the site. The Forms.aspx in the AppTaker.Forms application will list all of the forms in which the current date is between the dates set for the application taker.
13. Copy the tiny_mce directory from the CD to directory accessible by the URL http://<SITE>/jscripts.
14. For further development of the application you will need to create a directory from the root called LIB and copy the iTextSharp.dll into the directory. Also unzip the project files into you're my projects. Note if the place the application right off the web root you should not have to modify any of the development files, however if placed into a deeper directory or on a different machine I would recommend editing the build files manually and placing the correct URL to point to the source files.

References

1. ADOBE http://www.adobe.com/enterprise/pdfs/managing_archiving.pdf. March 2006

2. eXMill http://www.insoft-tech.com/exmill.htm March 2006

3. iTextSharp http://sourceforge.net/projects/itextsharp May 8

4. Microsoft development reference http://msdn.microsoft.com November 2005

5. Moxiecode http://www.moxiecode.com/ March 2006