7-19-2011

DATALOT

Niyati Goyal
Purdue University, ngoyal@purdue.edu

PURDUE
UNIVERSITY

Purdue University West Lafayette, Indiana

College of Technology

DATALOT

In partial fulfillment of the requirements for the
Degree of Master of Science in Computer and Information Technology

A Directed Project Report

By

Niyati Goyal

2011/06/30

Committee Member	Approval Signature	Date
Professor Alka R. Harriger, Chair	Alka R Harriger	7/5/11
Dr. John A. Springer	John A. Springer	6-30-2011
Dr. Eric T. Matson		6/30/2011

DATALOT

Simplify Data Management

A customizable web tool to facilitate collection and management of assessment data for research projects involving human subjects

A Directed Project

Submitted to the Faculty

of

Purdue University

by

NIYATI GOYAL

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

August 2011

Purdue University

West Lafayette, Indiana

TABLE OF CONTENTS

Page

ABSTRACT

Efficient data management is core to every research project, especially research involving human subjects. Some projects are tied to their participants (human subjects) over long periods and involve various stages of data collection in the form of surveys. These projects may also involve a series of presentations and sessions. Collection, organization and management of data at every stage are very critical to these types of research studies. Although there are a number of tools available to support each of these tasks, having a single tool to handle all of these can facilitate the process for the investigators or program managers. The purpose of this project is to provide a single, customizable data collection and management solution for such projects. This online tool provides a unified web interface for the researchers or program administrators, thus serving many purposes like organizing participant data, managing information about the sessions/presentations, creating online web surveys to collect assessment data, and viewing the collected data as reports. It also allows the administrators to download data in the form of CSV files for archiving and analyzing purposes.

CHAPTER 1. INTRODUCTION

1.1. Problem Statement

Purdue University is home to several research projects in diverse areas. Many of these projects involve human participants who visit the campus and attend structured sessions and presentations. These projects are designed to study certain quantitative or qualitative differences among participants based on their completion of program sessions.

Measuring and evaluating participant differences is an integral aspect of the research studies of these projects. Researchers can perform evaluation by collecting assessment data at various points during the course of a program. Conclusions are drawn from the results obtained through data analysis. Apart from collecting assessment data, the program manager also needs to manage all data associated with different participants, sessions and the surveys before and after the assessment data is collected. A single tool which will allow the program managers to group and manage the data as well as review it will considerably reduce the management overhead involved in administrating such programs.

There are several methods of collecting data for such programs. The most commonly used methods are web, email, or paper surveys. Web surveys provide numerous benefits and overcome the limitations of the other types. Web surveys provide greater access to respondents and higher control over how the user can answer a question, for instance, validating the format of a response by the participant. For studies involving a large number of participants, web surveying eliminates the need to physically publish and distribute paper copies of surveys. It also saves the time and effort required to transcribe received surveys and minimize data entry errors associated with this task. Web based surveys also provide the ability to present survey information in formats that were previously difficult to achieve (Schmidt, 1997). Moreover, the data can be directly exported to any desirable computer readable format like a CSV file, Excel sheets or be saved in a database. It can also be directly exported to statistical data analysis software programs like SAS or SPSS (Wright, 2005).

At present several online tools like Google docs and Survey Monkey are available to anybody who wants to conduct an online survey. However, they are not designed to serve the unique data collection needs of research projects involving human participants. These tools provide a solution for collecting online participant assessments via online surveys but do not provide a solution for organizing and managing the collected data. Also, the data collected through these tools is not tied to individual participant or program sessions. Projects which require collecting and monitoring responses of individual participants will need to adhere to paper and pencil surveys and organize the data in required formats after the task of data entry or invest time and resources in developing a custom data collection and management tool. These commercial tools do not provide any control of the data to the participants after they submit it. This tool provides a higher level of control to the administrators as well as participants over their data. Customer support for these products is not so well established and most of the companies charge extra to provide extended customer support like consultation and training (Wright, 2005).

In the case of commercial tools the survey administrators do not have access to the data servers. The data collected by commercially available tools resides on external servers and poses security concerns in terms of confidentiality and private participant data (O'Neill, 2004). This project developed to serve the needs of researchers at Purdue University to collect timely assessment data and at the same time allow them to manage participants and the session information through a single web interface. The tool is hosted on a Purdue University server, which means the data will be secured under all the protocols followed by the university to secure data on their servers with timely backup.

1.2. Significance

Data collection is an integral part of every research project. Research involving external human participants may require physical attendance of several sessions and presentations as part of the project. Evaluating the effectiveness of these sessions is an important aspect of the project. Evaluation involves several stages of data collection such as daily feedback. Timely and error free data collection is critical to the success and validity of these research studies. Also managing the data about the participants attending these sessions, monitoring individual participant responses and at the same time managing the session information are important aspects of such a research program. The program data about the participants and different sessions constitutes the research program and the collected assessment data is tied to it in several ways. For example the assessment data might need to be analyzed and organized based on various sessions or different kinds of participant groups.

Traditional data collection methods that employ one-on-one interviews or paper and pencil surveys increase the time required for the research and also impose an additional overhead of data entry before the data can be analyzed. Furthermore, potential errors may be introduced during the data entry phase, leading to unreliable results.

The World Wide Web (WWW), which has gained popularity as an information resource since the early 1990s (Commercenet, 1995), provides an efficient solution for the data collection needs of researchers. Publishing surveys online reduces costs in terms of both time and money over conventional methods. Web surveys can be administered simultaneously for the entire group, and the data can be directly stored into databases. This approach completely eliminates the need for an extra data entry step for the administrators, thus reducing the potential of common errors associated while entering the data manually. Additionally, web surveys can be intelligently tailored to be interactive and provide customized feedback, thus increasing the motivation levels of respondents (Schmidt, 1997).

Although web surveys offer significant gains by saving time and money, these advantages can only be realized by implementing carefully designed web surveys. Poorly designed software can lead to problems like missing data, duplicate submissions, security issues with confidential data, and unacceptable or incorrect data format associated with web forms.

Traditional web surveys are designed using HTML (Hypertext Markup Language) documents processed by CGI scripts. Design of secure and efficient user interactive web survey requires an innate understanding of HTML as well as CGI scripts (Schmidt, 1997). Most often research projects in non technical fields lack the required expertise to develop such surveys. There are several customizable survey tools available on the Internet. But, the use of these tools raises security concerns of data leakage, and there are additional costs associated with these products. These tools do not provide an interface for the program administrators to manage program participant information, manage sessions and link specific surveys for each of the different program sessions and monitor individual participant responses. The tool developed as a part of this project provides a generic, customizable web data management and assessment interface that is specifically suited to the data needs of research projects involving human participants who attend on campus programs would be an efficient data collection and management solution for university researchers.

1.3. Assumptions

The following are the assumptions in the study:

1. Every Program will have only one survey administrator at the beginning. The administrator can further add more users as administrators of the program.
2. Every research program is assumed to have a group of participants of the type *Presenter*. These individuals are part of the program team and lead the sessions or discussions during the program.
3. The program administrators can only be members of Purdue University with valid Purdue career account IDs. Program administrators will be using their Purdue IDs as their username to register into the system and for further access to the system.

4. The tool is designed to serve the data collection needs of research studies which involve human participants who visit the campus and attend programs in person (as opposed to virtually).
5. The surveys may be administered at any point, including before, during, or after the program. The administrators will have control to set the survey activation and deactivation time.
6. Only program administrators or their designees may manage the surveys.
7. Because the surveys are assumed to be conducted in the presence of the presenters or administrators, the study does not consider the issues of response rates and sampling bias associated with remotely conducted online surveys.
8. This tool does not allow collection of anonymous surveys. The participants need to be registered into the system by their program administrators. They can only access the system by using the username and password generated by the system. Administrators can thus monitor participant responses and follow up as required.
9. Sampling of the target population is assumed to be taken into consideration during the participant selection. The web tool is assumed to have no effect on the sampling of the target population.
10. The data collected by the use of this tool will be the property of the research program administrator.
11. The data will only be available to be modified or updated by the administrator or any individual authorized by the administrator.

1.4. Delimitations

The following delimitations have limited the scope of this study:

1. The tool only allows participants to be defined per program. A single participant cannot belong to two different programs. If an individual is part of two programs than s/he would be treated as two

different participants. S/he will have a unique, assigned participant username/password for each program.

2. Program administrators are treated as individual entities recognized by their Purdue email IDs. These email IDs serve as their usernames. If they are added as administrators for other programs, the new program is just added to their list of programs in the database.
3. Participants cannot save their survey responses without completing the mandatory questions. The survey status is then shown as C*ompleted* on their survey list. Participants are however allowed to return and edit their responses as long as the survey is kept active by the administrator.
4. A one-to-many relationship is defined between sessions and surveys. Each session can have multiple surveys associated with it, but a survey can only belong to one session. So if different sessions need to use the same survey questionnaires the program administrators should still create a new survey for each session and copy the questionnaire. Every survey is registered with a separate unique survey ID.
5. The participant ID field provided for participant information is to help the program administrators identify their own participants. Different programs can have different schemes of assigning IDs to their participants. These program participant IDs are however not the unique IDs used by the system. They are like any other non unique details such as the first name of a participant. If a program administrator uses a duplicate ID for two different participants, the system will allow it.

1.5. Limitations

The following are the limitations of the study:

1. After the initial development of the tool on local machine, space was needed on Purdue University's ECN server for web hosting. A delay in getting the approval for the server space impacted the implementation schedule and the ability to conduct usability testing.
2. The number of users who can simultaneously access the application is limited by the technical specification of the database type and server capacity.
3. The types of questions and the number and levels of validation supported in case of survey administration is limited by the technical design specification of the tool.
4. The level of data security achieved through this tool is restricted to the security provided by the Purdue University server on which it runs and not on a third party, outside server.

1.6. Definitions

Administrator – A type of user who owns/manages one or more programs.

Assessment data– The data collected by the administrators through this system to evaluate a program by means of surveys.

CGI script - Web applications involve sending of HTML documents to the users' web browser. The browser collects user input and sends it back to the server. The server has no information about the format of incoming data. The data first needs to be passed to separate programs for processing. These programs are called Common Gateway Interface scripts and can be written in any programming language (Schmidt, 1997).

CSV (Comma separated value) - This file is created by separating related field values by commas and a line break after each group of related values. These files can be easily opened in Microsoft Excel and manipulated.

HTML (Hypertext Markup Language) - A standard document format that consists of annotated, textual content to describe the static content of web pages.

Human Subjects – When a research study involves humans as part of the experiments and tests for the study.

Internet – It is a system of interconnected global computer networks to share data and information.

IRB (Institutional Review Board) - This is a unit within a university that oversees compliance of any research study involving human subjects appropriately safeguard the privacy and interest of the human subjects involved in the study.

Module – A set of features grouped by functionality. E.g. a group of features which helps managing participants in the system is a module.

Program – The complete set of activities of a project involving human subjects, sessions and presentations.

Report – View of subset of assessment data as identified by the user.

RIA (Rich Internet Application) - RIAs separate the presentation and interaction layer form the server and aim at minimizing the data transfer between the server and the client (Preciado, Linaje, Sanchez, Comai , 2005)

Server – Any combination of hardware and software that manage access to centralized resources/services in a network.

Server side program - A server side program is software which runs on the server and not on the client machine.

Session – A single activity that is part of a larger program that the participants (human subjects) attend. A set of sessions when combined together form a program.

User – Individual allowed to login into the tool and access it is considered to be a user.

WAI (Web Accessibility Initiative) – It develops series of standards and guidelines to address the issue of essential components of web accessibility.

Web Assessment – An analysis of the effect of a particular session, program or course by participants who complete surveys and feedback forms deployed on the World Wide Web.

Web Services –Web applications that are designed in such a way so as to support interoperability between various machines and networks.

1.7. Summary

This section provided an overview of the research work, including significance, problem statement and definitions. The next section outlines the motivations for the study. Also, it provides an overview of the current methods for data collection in research studies. The literature review also includes design considerations along with the technical challenges in designing of such data intensive web applications.

CHAPTER 2. LITERATURE REVIEW

This section talks about the related work done by other researchers. It provides the background for the work done by the author.

2.1. Motivation for Using a Web Based Tool

This section delves into the advantages, disadvantages and issues associated with online surveys and the costs and problems with the commercially available web tools.

World Wide Web (WWW) has virtually affected every aspect of society over the last two decades. Research surveys are one of the numerous areas which have undergone massive transformation with the advent of the Internet (Solomon, 2001). The ability to be able to reach a large population at the same time is a strong motivation for the use of the web in survey research.

Web surveys offer a great potential to devise effective means of conducting surveys and data collection in varied fields. Kay and Johnson (1999) identified over 2,000 Web-based surveys in 59 diverse areas in an informal search of Yahoo. Several organizations have used web survey tools for the process of recruitment and selection since the early 90s (Berner, 1994; Schmitt, 1997).

Web surveys have not only been increasingly used for surveying in large commercial organizations but also shown to be influencing the health care and bio-medical researchers. Marleen, Gelder, Reini and Roeleveld (2010) discuss the potential of web based questionnaires in epidemiological research. The article discusses the advantages, disadvantages and current application of web surveys in epidemiological studies. The web offers a promising mode of data collection in the area. The study concludes that theoretically, web-based data collection can be considered as an alternative or complementary mode of data collection in epidemiological research studies. Web-based survey research is also being widely used in the areas of social sciences and educational research. The increasing popularity of web-based surveying can be attributed to the obvious advantages offered by online tools over traditional survey methods.

Web-based surveys significantly reduce time and cost of conducting a survey at the same time, eliminating the potential errors associated with the task of manual data entry (Medin, Roy, Ann, 1999).

A study conducted by Stanton (1998) performed an empirical assessment of web surveys. Two population samples were chosen to answer the same set of questions. One sample was asked to fill out the paper and pencil survey while the other sample completed a web survey. Analysis of the two sets of data using statistical tools proved that the data collected over the Internet had fewer missing values than the paper and pencil data.

The use of web based surveys in behavioral research has been reported to yield better quality data by reducing the errors due to the inclusion of explanatory material and prompts on the online surveys. Web surveys provide a more structured interactive survey to the respondents thus reducing the instances of erroneous entry (Rhodes, Bowie, Hergenrather, 2003).

2.2. Developing Web Based Assessment Instruments

Most web surveys are developed using HTML forms. Program code known as common gateway interface (CGI) scripts may be used to process the user input, including verification of the acceptability of the data as well as copying the data into organized computer files for later data analysis. Several packages like the Microsoft's FrontPage and Macromedia's ColdFusion are HTML development packages that provide HTML editors as well as automatically generate the CGI scripts (Solomon, 2001).

Web Survey Methodology has been a significant area of research. –The WebSM website (http://websm.org) provides access to a collection of the publicly available research in the area and has over 700 bibliographical units. The site also lists numerous commercial web survey software packages available for researchers willing to invest both money and time (Manfreda, Batagelj, Vehovar, 2002). A study by Wright (2005) compiled a list of the 20 most prominent software packages and web survey services available to the researchers. Table 2.1 lists 20 packages, along with their web addresses, pricing and features.

Table 2.1 Comparison of online survey software and services (Source: Wright, 2005)

Company Name	URL	Features	Pricing	Service Limitations
Active Websurvey	http://www.activewebsoftwares	Unlimited surveys; software automatically generates HTML codes for survey forms	Information unavailable on website	Customer required to purchase software; limited to 9 question formats
Apian Software	http://www.apian.net/	Full service web design and hosting available	$1195 up to $5995 depending on number of software users; customer charged for technical support	Customer required to purchase software
CreateSurvey	http://www.createsurvey.com	Standard features; educational discount	$99 a month for unlimited surveys and responses; free email support	Survey housed on company server for a set amount of time
EZSurvey	http://www.raosoft.com	Unlimited surveys; mobile survey technology available; educational discount	$399 for basic software; additional software is extra; telephone training is $150 an hour	Customer required to purchase software
FormSite	http://www.formsite.com	Weekly survey traffic report; multiple language support	$9.95 up to $99.95 per month depending on desired number of response	Survey housed on company server for only a set amount of time; limited number of response per month
HostedSurvey	http://www.hostedsurvey.com	Standard features; educational discount	Charge is per number of responses; first 250 response are free, then around $20 every 50 responses.	Survey housed on company server for only a set amount of time
InfoPoll	http://www.infopoll.net/	Standard features; Software can be downloaded for free	Information unavailable on website; limited customer support; training available for a fee	Software can be downloaded free, but works best on InfoPoll server; customers appear to be charged for using InfoPoll server
InstantSurvey	http://www.netreflector.com	Standard features; supports multimedia	Information unavailable on website; free 30 day trial	Survey housed on company server for only a set amount of time
KeySurvey	http://www.keysurvey.com	Online focus group feature; unlimited surveys	$670 per year for a basic subscription; free 30 day trial	Survey housed on company server for only a set amount of time; limited to 2000 responses

Company Name	URL	Features	Pricing	Service Limitations
Perseus	http://www.perseus.com	Educational discount; mobile survey technology available	Information unavailable on website; free 30 day trial	Survey housed on company server for only a set amount of time
PollPro	http://www.pollpro.com	Standard features; unlimited surveys	$249 for single user; access to PollPro server is an additional fee	Customer required to purchase software
Quask	http://www.quask.com	Supports multimedia	$199 for basic software; access to Quask server for an additional fee	Customer required to purchase software; more advanced features only come with higher priced software
Ridgecrest	http://www.ridgecrestsurveys.com	Standard features; educational discount	$54.95 for 30 days	Survey housed on company server for only a set amount of time; limited to 1000 responses for basic package
SumQuest	http://www.sumquest.com/	Standard features; user guidebook for creating questionnaire available	$495 to purchase software; free unlimited telephone support	Customer required to purchase software
SuperSurvey	http://www.supersurvey.com	Standard features	$149 per week for basic package.	Survey housed on company server for only a set amount of time; 2000 response per week limit
SurveyCrafter	http://www.surveycrafter.com	Standard features; educational discount	$495 for basic software package; free and unlimited technical support	Customer required to purchase software
SurveyMonkey	http://www.surveymonkey.com	Standard features; unlimited surveys	$20 a month for a basic subscription; free email support	Survey housed on company server for a set amount of time; limited to 1000 initial responses
SurveySite	http://www.surveysite.com	Company helps with all aspects of survey design, data collection and analysis; online focus group feature	Information unavailable on website	Company staff rather than customer create and conduct survey
WebSurveyor	http://www.websurveyor.com	Standard features; unlimited surveys	$1,495 per year for software license	Customer required to purchase software
Zoomerang	http://www.zoomerang.com	Standard features; educational discount	$599 for software	Customer required to purchase software

The web tools listed in Table 2.1 above can be classified into two major categories based on their current features. One category is the online survey software packages which are only computer programs and researchers use their own computers to create and conduct online surveys. These packages usually include features like customer support, server space, data tracking and analysis options. The second category encompasses a wider range of services including the design, online questionnaire development along with the data analysis services. The price of the survey varies depending on the requirements of the study and the features offered by the businesses (Wright, 2005).

Apart from the pricing and design, ethics, security and control are important aspects of web surveys which need to be considered in research involving human subjects. A study on e-research by Nosek, Banaji, and Greenwald (2002) discusses these issues in detail with regards to psychological research and proposes possible solutions and guidelines. Although the study focuses on psychological web research, the proposed solutions and guidelines can be extended and applied in other research areas. The study points out the ethical issues regarding adequate informed consent and debriefing, and potential loss of anonymity or confidentiality. The author suggests creation of a debriefing page with sufficient FAQs and also a leave the study button for every page. Inclusion of a consent page and assent page at the beginning of the surveys would also be a possible solution to guarantee informed participant consent requirements. A built-in template which could be suitably edited by the administrators would be an ideal solution to these issues and would considerably reduce preparation time for the researchers.

As pointed in the above tables there are several limitations to using the commercially available survey tools. Some companies only allow researchers to host the survey for a set amount of time. There might be an extra cost for keeping the survey for an extended period of time. The funding resources of the research project might not be sufficient and cause an unwanted hurdle towards the completion of research study (Wright, 2005).

The study by Wright (2005) described in the above tables clearly points out that the online available survey tools are too generic in terms of the wide range of survey types they target and do not necessarily focus on the data collection needs of university research projects. Researchers also need to pay for these commercially available tools. Most of the tools mentioned above come in packages and researchers need to pay for the complete package even if they intend to utilize all the features of the package. Also, it is often seen that as the number of offered features of a tool increase, its usability goes down. This concept applied to technology in general is known as feature fatigue (Thompson, Hamilton, Rust, 2005). A web tool which will focus on assessment needs of research studies would be an ideal solution to save time and money of the researchers. Because the project focuses on developing a data intensive web tool, the following section reviews some literature on design and techniques in the areas of web and data modeling.

2.3. Web and Data Modeling Techniques

This project developed a web assessment tool to serve the data collection needs of research projects involving human subjects; therefore, the following section examines the research in the area of modeling data intensive web applications. In recent years, there has been exponential growth in the amount of information available over the web. With the increase in the amount of data over the web, there is a simultaneous need to optimize the methodologies to model, process and present the data on web. This has led to significant research in the area of developing efficient methodologies and tools for modeling of web applications and data for such data intensive web applications.

The study conducted by Fraternali (1999) models the development of data intensive web applications as a balance between Information Systems development and hypermedia authoring. The study involved survey and analysis of available web development tools over a set of parameters like lifecycle coverage, process automation, abstraction, reuse, architecture and usability. The study showed that development of a web application and the selection of tools depends on the underlying business rules and data needs of the application being developed.

The study carried out by Ceri, Fraternali and Matera (2002) investigated the development of a conceptual model for data intensive web applications. It analyzed the use of the Web Modeling Language (WebML) as a basis of conceptual modeling language. It describes the use of WebML to abstract data intensive websites into skeletons of data and hypertext diagrams. The study classifies a website in the following two orthogonal conceptual dimensions:

- Data Model
- Hypertext Model

Each dimension is further decomposed into skeletons of core concepts, access concepts, interconnection concepts, and content management concepts. The data model for WebML relies on the use of entity relationship (ER) model for data modeling. The study analyzed one of the several approaches towards modeling and design of data and web applications.

There is another approach of model driven development of data intensive web applications. The study by Ceri, Daniel, Matera and Facca (2007) presents a framework for conceptual modeling for context-aware, multichannel web applications. The basis of the study is to show that high level modeling constructs can drive the application development process by automatic code generation. The basis of the design is separation of data and hypertext. Context is identified as data and users and groups are represented as first class citizens in the application data store. The study covers the need of context to be adaptive for personalization, which is an important feature for many e-commerce sites. It also introduces an XML specification for the data operations and design which can be further included as part of WebML.

There has been extensive research in the use of WebML as a standard modeling language for websites. The research has led to various approaches towards the extension of WebML for serving the design and modeling needs of data intensive web sites. The study conducted by Bongio, Ceri, Fraternali and Maurino (2000) looked into enhancing the WebML by adding data entry and operation units. The study proposed addition of data entry units and operation units to the existing WebML architecture. The addition of data entry unit to

WebML enhances its capability to support handling of data and the operation unit helps in processing of data.

Another study conducted by Ceri, Matera, Rizzo and Demalde (2007) introduces the idea of content accessibility as the focus for designing modeling tools for data driven web sites. The existing WAI guidelines focus on presentation accessibility rather than the management and modeling of content. The study suggests identifying core content areas of a web site and modeling them as web marts a concept similar to data marts in data warehousing environment.
Ceri and Fraternali (2003) explored the architectural challenges faced during the design of data intensive web sites and proposed several solutions. They analyzed different aspects and specifications of WebML and proposed a web tool called Web-Ratio. The challenges of a modular design and architecture were handled by suggesting trade-offs for several design needs. Web-Ratio addresses the areas of MVC architecture, scaling, managing presentation, caching and optimization which are the major problem areas in modeling a data intensive web application. Web Ratio also applies the concept of CASE tools and automatic software generation tools. The study demonstrates the relation between modeling and architectural issues.

Another area of research is the design of web services. Web services aims at designing of reusable web sites offered as a service and can be used by registered users. A study conducted by Brambilla, Ceri, Fraternali, Acerbis and Bongio (2005) focused on using the model driven approach towards design of service-enabled web applications. The study evaluates web services as a tool for integrating process intensive web applications driven by business rules and domain specification with data intensive web applications. The approach classifies various components into WebML based service units. These units then interact via WebML workflow primitives. The architectural implementation is done using the CASE tool web ratio. The methodology is applied and evaluated for developing three complex applications. The study clearly demonstrates the use of high level language, conceptual data modeling is a good tool for integrating and designing complex data driven web applications.

A study by Andrews, Kappe, Maurer (2000) explores the maintenance of large scale information systems on the web. The paper describes a new distributed large scale information managing system Hyper-G. Hyper-G combines the following features into one:

- Intuitive top down hierarchical navigation
- Associative hyperlinks
- Content searches

Hyper-G is an optimum solution for maintaining large information bases. It also optimizes the search and presentation aspects of a web site.

Another related research area in the field of web is the various web engineering methodologies that have evolved over time. All methodologies serve some or all of the desired features for designing web applications. There is a new category of classification of web applications called the Rich Internet Application (RIA). RIAs provide a higher level of interaction and presentation for a web site. There is a set of features which are demanded by RIA web sites and the existing web, multimedia and hypermedia technologies do not support all of these features. The study by Preciado, Linaje, Sanchez and Comai (2005) describes the special characteristics which define a RIA and evaluates different existing methodologies in web, multimedia and hypermedia against a set of comparison parameters and statistically evaluates the results for each of them. The results of the study showed that there is no existing methodology which supports all of the features of RIA. Also, only the recent model-driven methodology supports future extension RIA features. The study demonstrates the need for further research and development in the area of web, multimedia and hypermedia technologies.

The study by Navathe (1992) delves into the evolution of data modeling. It evaluates the basic characteristics of various data models like relational, semantic and object oriented modeling techniques with a focus on application development. It describes the advantages of mathematical modeling inherent in a relational model. The expressiveness of a relational model in relational algebra makes it simple and easy for query optimization. It reviews the features of relational model which make it a flat model. This led to the evolution of semantic data models. Semantic data model focuses on conceptual model rather than the underlying DBMS. Entity relation (ER) model is an example of a semantic data model. The differences between the relational and ER model have been clearly explained with an example of a data model for an employee database. The article also reviews other data models like the object-oriented models and their characteristics pointing out the similarities between the semantic data models and O-O data models. It briefly defines the idea of Active and Dynamic databases.

Codd (1982) derived the relational data model based on the principles of relational algebra. The study focuses on the role of relational database systems in increasing the productivity of data processing application programs and end users. The article describes three principal reasons for the failure of traditional DBMS systems to boost the productivity of application programs. It describes the objectives of achieving data independence, communicability and the set processing as the motivation for the evolution and study of relational model. It describes the relational model in detail with an emphasis on its manipulative part and the integrity part. It describes various aspects of the relational processing capability which makes relational model a productivity booster for application programs.

Another study by Blaha, Premerlani and Rumbaugh (1998) focuses on the object oriented database technology. It introduces the idea of Object Modeling technique (OMT) which is based on the principles of semantic databases. The methodology incorporates the application of OMT to relational database design. The methodology categorizes the methodology into three levels of representation:

- High
- Middle
- Low

High level identifies the logical data model, middle level identifies DBMS independent tables, and low level identifies the actual database definition commands. OMT application was performed by two different people on a chemical engineering and an electrical engineering problem. The study shows that OMT is more intuitive, expressive, and extensible provides a useful level of abstraction and promotes database integrity and integration over the existing ER or LRDM approaches.

Hammer and McLeod (1981) studied the semantic database model. SDM schema provides specification of the information that the database will contain. It also provides the Database Administrators (DBAs) a basic methodology that can be used to build a logical database model.

Designing a data intensive web application requires a clear identification of data needs and application needs. The article introduces the conceptual and model driven approaches to web development of data intensive web applications. It also analyzes the different studies and approaches for web modeling and data modeling.

Web modeling approaches include the model driven approach based on the use of WebML as the language for web modeling. WebML uses the entity relation (ER) model for data modeling of web applications. There has been research in the areas of extending and improving WebML for data entry and operations by introducing data entry units and operation units in the WebML architecture.

2.4. Summary

This chapter provided an overview of the available literature on the significance of web assessments in research studies as well as the problems and costs associated with the commercially available survey tools which provides the motivation to develop the tool. It reviewed the issues concerning the design and development of such a tool along with a comparative study of the available tools and their features and limitations. It also discussed the web modeling and data modeling techniques as the project focuses on development of a data intensive web tool for collecting and managing research assessments for human subjects.

Data modeling section concentrated on the various types of existing basic data models like relational, semantic and object oriented. A study reviewed the characteristics of relational model and its importance in boosting the productivity of application programs. The evolution and features of semantic model identified the weaknesses of relational model being flat in nature and less expressive. The implementation of OBT (Object oriented technique) highlighted the weaknesses in the entity relation (ER) model and the areas of improvement.

All the research in the area of data modeling for data intensive web application inherently focuses on web methodologies and web modeling. The data modeling focuses on the database techniques like relational, semantic and object oriented.

CHAPTER 3. DESIGN FRAMEWORK

This section describes the functional and technical design specifications of the tool developed by the author followed by the methodology used for performance evaluation.

3.1. Functional and Technical Design

DataLot is designed to allow researchers to create, collect and manage their research program data, and assessment data submitted by the participants via a single web interface. The following sub sections describe in detail the functional and technical design specifications of the tool.

3.1.1. Functional Design Specifications

DataLot is designed to allow program administrators to create, collect and manage their program data. This tool allows the administrators to setup assessments for the program participants. The participants than login into the system and answer any of the assessments they are required to submit as part of the research study. Because the tool is accessible to three types of users -- program administrators, participants and the program presenters -- it has the following user views:

1. Admin View
2. Participant View
3. Presenter View

Each of these views is comprised of different modules. All the different modules and their functionalities are described in the following sub sections.

3.1.1.1. Admin View

Admin View is the view of the tool visible to the program administrators when they login into the system with their administrator credentials. This view allows the administrator to setup the program sessions, surveys, participants and view/download reports of the submitted assessments. Each of the modules in the Admin View with different pages and their dependencies and functionalities will be discussed in detail below.

As soon as the administrator accesses the DataLot URL, s/he will be presented with a login page asking for his/her credentials. First time users of DataLot can create a new administrator account via the *Create Account* link provided at the bottom of the page. Administrators can only use their Purdue email ID as username for registering into the system. Once the administrator registers and logs in into the system s/he will be redirected to his/her personal *Home* page. The *Home* page of the administrator provides a list of all the programs for the administrator. A single administrator can be running multiple programs and s/he can manage all of his/her program data from this single interface. Administrators can create a new program from the *Add new Program* link or *edit* or *delete* the existing programs. Administrator can than choose to manage the details of individual program by clicking on the program title link in the table. This link redirects the administrator to the program specific pages and modules. Once the administrator enters a program, s/he will only see details of the selected program. The program menu bar at the top right corner (see Figure 3.1) helps the administrator to access all of the different modules of the system related to the specific program.

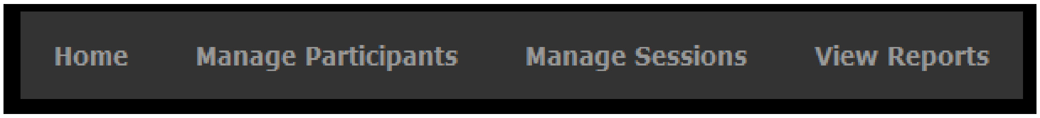


Figure 3.1

The *Home* tab in the menu bar navigates the user back to the list of administrator programs. The other three modules of the Admin View are described below.

3.1.1.1.1. Manage Participants

The first step in setting up a program is to setup the information about the program participants. The second menu item in the menu bar shown in Figure 3.1 – *Manage Participants* navigates the administrator to different pages which capture the details about the program participants. This module captures all the data associated with all kinds of participants involved in the research program.

The first step in setting up the participant information is defining the participant type information. The first sub menu item under the *Manage* P*articipant* tab – *Manage Participant Type* navigates the administrator to the page where s/he can setup the participant types for the program. Every program can have various types or categories of users participating in the program. For example a program might include capturing differences on perception of a particular session by male and female participants. The kind of assessments for each group will be different. In this case *male* and *female* are two different participant types associated with the program. Usually the sessions and presentations attended by different groups are also different. The type of assessments collected for these different groups may also differ. The *Manage Participant Type* page allows the administrator to assign the different categories (participant types) of users that will be participating in the program. Every participant that the administrator uploads into the system must belong to at least one or more of these categories. There is a default participant type called *Presenter*. Every program is assumed to have *Presenter* participants who lead/help lead specific sessions and discussions for the program.

Once the user sets up the participant type information for the program, s/he should proceed to upload the participant information which belongs to the different participant type setup above. The second sub-menu under the *Manage Participants* tab – *Manage Participant Info* navigates the administrator to the page for managing individual/multiple participant details. The page shows the administrator a list of participants and their information, which is already uploaded into the system. The administrator can add/edit/delete individual participant information or choose to upload a CSV file in the format specified on the page for uploading multiple participant data at once. If the administrator chooses to upload individual participant information they must select the participant type of the individual. The participant type checkboxes are populated from the participant type information provided by the administrator on the *Manage Participant Type* page.

Once the administrator uploads the participant information into the system, the system generates *usernames* and *passwords* for each of these participants. All this data is listed into the table which is populated with the uploaded data. The administrator can download this data to the CSV format by using the *Export to CSV* button below the table. The administrator can then pass on the login information to the respective participants via mail merge or any other means. This information is important as it will be required by the participants to enter into the system and answer the surveys and assessments they are required to submit for the research study.

A program can be managed by multiple administrators. Once an administrator registers into the system s/he can add other administrators to the program who will all share the same administrator privileges. The administrator can do this by the third sub menu item under the *Manage Participant* tab – *Manage Admin Info*. This sub menu navigates the administrator to a page which lists all the administrators of the program. The administrator can choose to add new administrators of the program, manage his/her own account information like name and password information, or remove the administrator privileges of other users. The primary administrator cannot however choose to remove herself/himself as the program administrator. This feature was included to avoid situations when the current administrator is the only administrator of the program, and removing the administrator for the program would result in all the program data being lost due to the absence of an administrator.

When an administrator adds another administrator to the program, the system checks the username (Purdue email ID) to verify if the user is already registered as an administrator user into the system. If the administrator is already registered than the current program is added to his/her list of programs on his/her

Home page. If the user is not already registered into the system a new administrator user is created and the program is added to his/her list of programs.

3.1.1.1.2. Manage Sessions

The second step in setting up this system to use for the program is setting up the various sessions and presentations which are part of the research program. The third item in the program menu bar in Figure 3.1 – *Manage Sessions* navigates the administrator to the page where s/he can manage all the information related to the sessions of the program. The top half of the page shows a list of all the session data already uploaded into the system if any. Every session is assumed to be attended by one or more participant types of the program. For example one session might be attended by the *male* participants for a program which can be one type of participants in the program. The administrator can setup the basic session information like the session title and description, the start date and time, and end date and time. The list of presenters who will be leading the session is populated from the participant uploaded as *Presenter* on the participant information page. The administrator can select one or more of these individuals as presenter for a particular session. The administrator also selects the type of participants who will be attending a particular session.

The administrator can choose to upload individual session data by using the *Add new Session* link at the top of the page or upload a CSV file in the specified format under the *Manage Multiple Sessions* subheading. The administrator can also edit/delete the session data by using the *edit* and *delete* links beside each of the session rows in the table.

A program may be required to collect assessment data for each session of the program via surveys. Beside each session information is the list of surveys associated with the session. If there are no surveys associated with the session a *No surveys* message is displayed in red. Using the *Create Survey* link located beside every session, administrator can navigate to the page where s/he can design/create survey for the session.

Once the administrator navigates to the survey creation page s/he will be asked to fill in the survey information in two parts – The basic survey details like the title, description, activation date and time and the deactivation date and time. Activation date and time is the time after which the survey will be active for the participants to answer and similarly the survey will no longer be available for the participants to answer after the deactivation date and time. Activation and deactivation date and time are defaulted to be the session start and end date and time. The administrator can however choose to change these values to be different from the session values. The administrator also chooses one or more type of participants who are attending the session who will be asked to answer these surveys. The administrator can therefore administer different surveys for different types of participants attending the same session if needed.

The second step is to create the questionnaire for the survey. The administrator can choose to copy the questionnaires from the existing surveys by using the *copy survey* functionality provided on the page. After copying the questions, administrator can choose to make any specific changes to the survey and save it as the survey for this session. The panel at the top of each question on the survey creation page lets the administrator choose the question attributes for each question. S/he can select from the list of question types, force responses for mandatory questions by checking the *Force Response* checkbox and also choose to provide custom validation message for each mandatory question.

The administrator will be allowed to edit the survey details by clicking on the survey title link on the *Manage Sessions* page. The administrator can come back and edit the basic survey details even after some of the participants already submitted the survey. Thus if the administrator decides to increase the time for which the survey is active s/he can do that even after it has been submitted by some of the participants. However, if the administrator chooses to change the survey questionnaire after the survey answers have been submitted by some of the participants than the already submitted responses of the participants will be permanently deleted from the system and the new edited survey will appear as *Not attempted* for participants when they login into the system. This is to avoid any kind of data inconsistencies into the system. The administrator needs to be careful if s/he decide on editing the survey questionnaire after it has been active and submitted by a few participants. After the administrator saves the survey details s/he will be redirected to the *Manage Sessions* page where s/he can see the survey listed in the table.

The administrator can also download the program session data by using the *Export to CSV* button as the bottom of the page.

3.1.1.1.3. View Reports

After the administrator sets up the participants, sessions and surveys for the program, the tool is ready to be used for collecting assessment data by means of the surveys created for the sessions. The data collection will be covered in more details in the Participant View section of this report. After the participants submit their responses for the surveys administered for them the *View Reports* module allows the administrator to view the results of the surveys as well as monitor the status of who all submitted the surveys.

The first sub menu item under the *View Reports* tab in the program sub menu in Figure 3.1 is the *View survey status* page. This link navigates the administrator to the page where s/he can choose to view the status of individual surveys submitted or not submitted by the participants in real time. The administrator needs to select a specific program session from the dropdown list of program sessions. Once the administrator selects the session, a list of all the surveys created for that session is populated in the survey dropdown. As soon as the administrator selects a survey from the survey dropdown s/he can view the submission status report of that survey for each participant. The tool also provides the administrator the following criterions to filter the status report:

1. Show All
2. Submitted

3. Not Submitted

The survey status report also includes a column with the participant email, which was uploaded by the administrator on the participant information page. The administrator can download this status report in the form of a CSV file, and use it for sending a follow up email to the participants who did not respond to the survey. This feature can help programs increase their response rates from participants when this data might be critical to the results of the research study.

The second item in the sub menu under the *View Reports* tab is *View survey reports* navigates the administrator to the page where s/he can view results of the responses submitted by the participants. The administrator can download the results in the CSV file format and use them for further analysis and reporting purposes. S/he can also view them as tables generated on the page and filter them based on different filter criterions.

This completes all the modules associated with the Admin View of the tool. The following section describes the functional specifications of the participant and presenter views.

3.1.1.2. Participant View

Participants are the human subjects of the research study. These individuals usually attend sessions and presentations designed by the researcher and are then asked to submit the assessments in the form of the survey created by the program administrators. When the administrator uploads participant information using the *Manage Participants* module of Admin View, the system generates unique *usernames* and *password* for each of the participants. Participants can then login into the system using these credentials provided by their program administrator.

When a participant logs in into the system s/he is redirected to his/her *Home* page. The Participant View *Home* page lists all the active surveys for that participant, depending on his/her participant type and the sessions attended. This eliminates the risk of participants choosing to answer the surveys for the sessions they did not attend. The table also lists some basic information about the session with which the survey is associated to give them some clues to recognize the session. The table also shows the status of the surveys for an individual participant. If the participant submitted the survey by answering at least the required answers, then the responses are saved into the system and the survey status is *Completed*. Participants can however choose to go back and edit the responses as long as the survey is active. The surveys which they have not attempted even once show the status of *Not Attempted* in red, thus indicating that the response is pending. When the participant clicks on the *Respond* link for a particular survey s/he is navigated to the online survey with the set of questions as designed by the administrator. Participants will be prompted for error if they do not submit responses for the required questions with the custom message if provided by the administrator or a generic system message, thus making sure that all the mandatory questions are answered.

3.1.1.3. Presenter View

Program presenters are the individuals associated with the program who lead the sessions or discussions associated with the program. Presenters are still considered as program participants with some special privileges. A participant can be a session attendee for some sessions and a presenter for others. A presenter can log in into the system using the credentials provided to the administrator on uploading the presenter participant information into the system. Once a presenter participant logs in into the system s/he is navigated to the Presenter View *Home* page. This page is the same as the *Home* page for any other type of participants. If the presenter was also an attendee for some sessions, then the home page will list the surveys s/he is required to respond to like it is for other participants.

The only difference between the Presenter View and the Participant View is that the presenter can view the results of the surveys associated with the sessions for which they were the presenters. The presenter cannot view any participant detail on the reports. Presenters can only view anonymous responses to the survey questionnaires. They can view the reports by choosing the *View survey reports* tab in the top right menu which appears on the *Home* page.

Some programs also have external evaluators who need to review the results of surveys for all the sessions, but they are not allowed to view the participant names and IDs. After the surveys are deactivated, administrators can add the evaluators as presenters for each of the sessions. Then, when the evaluators log in, they can view anonymous results of all the sessions and review the results. Another alternative is that the administrators can download the reports without the participant names and ID and share it with the evaluators.

3.1.2. Technical Design Specifications

This section describes the technical design specifications of the system in detail. The initial development of the tool was on the local 64bit windows 7 machine using the free download mini Apache server development package called WAMP server 2.1 which the following configuration:

1. Apache 2.2.17
2. PHP Version 5.3.4
3. MySQL (phpMyAdmin are preinstalled for managing the database)
    a. MySQl server version 5.1.53
    b. MySQL client version- mysqlnd 5.0.7-dev - 091210 - $Revision: 304625$

After the development was completed on the local system, space was requested on the Purdue Engineering web cluster (ECN) which supports the PHP Apache server for web deployment. 10GB of space was requested on the MySQL database on ECN. The technical specifications of both are below:

1. Apache 2.2.17
2. PHP Version 5.3.4
3. MySQL version 5.1

As indicated by the technical specifications of the system above, the tool was developed using several technologies that are described in Table 3.1

Table 3.1 Technical Design

Technology	Purpose
PHP	Used as a scripting language above ASP.Net or Java because of the ease of use and a higher level of control it provides to the programmer. PHP script can be easily embedded into the HTML pages thus providing an easy handle to dynamically control the web pages without making a server trip. Also, there is a higher level of support available for integrating Ajax and Jquery within the PHP code.
HTML and Jquery1.5 and other Jquery libraries	For designing the web pages and handling the client side validations.
Ajax XHR	Requests for some of the calls to the server in order to avoid page refresh for each request and also reducing the time required for completing the server requests.
CSS template	Used for designing the development version of the tool was downloaded from the free CSS templates provided by OS Templates on the web.
XML	Used for the survey designing and editing modules described in the Admin View. XML was chosen for the flexibility it offered by an XML in terms of scalability and design for capturing information. Also PHP, JQuery and Ajax provide a set of inbuilt libraries for fast and efficient parsing of XML data.
Firefox browser plug-in Firebug 1.7.2	Used for client side debugging during development.

The following sub section will describe the detailed database and code design of the project.

3.1.2.1. Database Design

DataLot's database design employs thirteen tables. Figure 3.2 shows the Entity Relationship Diagram for the physical database design of the system.

Each of the tables in Figure 3.2 is described further in Table 3.2. The database was designed to be relational so as to reduce any redundancies and inconsistencies in the data. Foreign key constraints were created to avoid inconsistent addition or deletion of data.

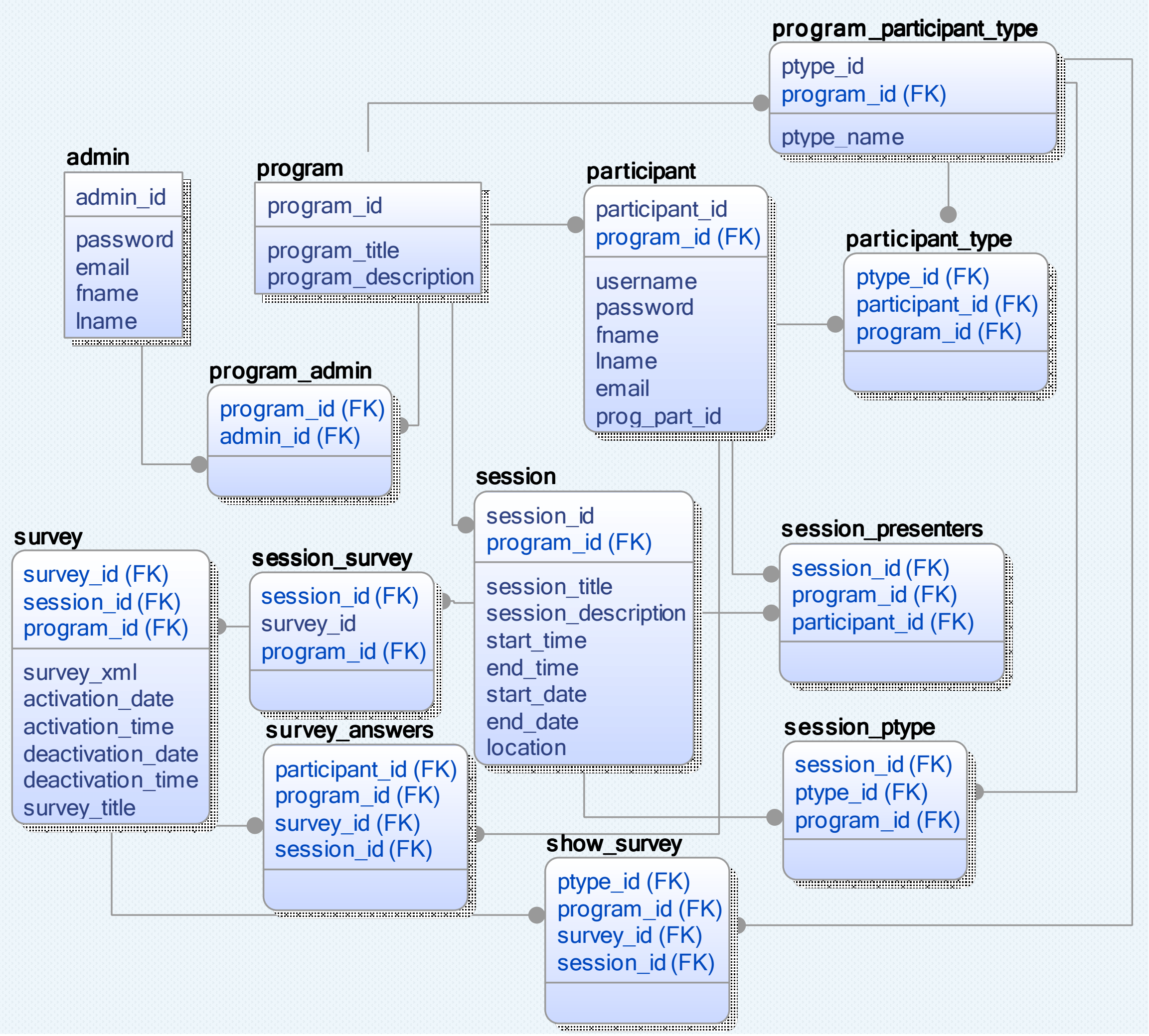


Figure 3.2 Database design

Table 3.2 Database Tables

Table	Description
Admin	This table holds all the information related to the registered administrators in the system.
Program	This table holds the details about the programs created by administrator users.

Table	Description
program_admin	Each administrator user can have multiple programs. This table holds the list of all programs for each administrator user.
program_participant_type	This table holds the participant type information defined for each individual program.
Participant	This table holds information about all the users of the system other than the administrator. Every user other than the administrator is considered as the participant.
participant_type	This table holds the participant type information for each participant registered into the system.
Session	This table holds information about all the session associated with the programs.
session_ptype	This table holds information about the type of participants attending each session associated with a program.
session_presenters	This table holds information about the participant who are the presenters for individual sessions.
Survey	This table holds information about the surveys created by administrator users in the system.
session_survey	This table holds information about the survey and the associated sessions with the surveys.
show_survey	This table holds information about the participant types who are required to respond to individual surveys.
survey_answers	This table holds the participant responses to the surveys.

3.1.2.2. Software Design and Organization

This section describes the software design and code organization of the project.

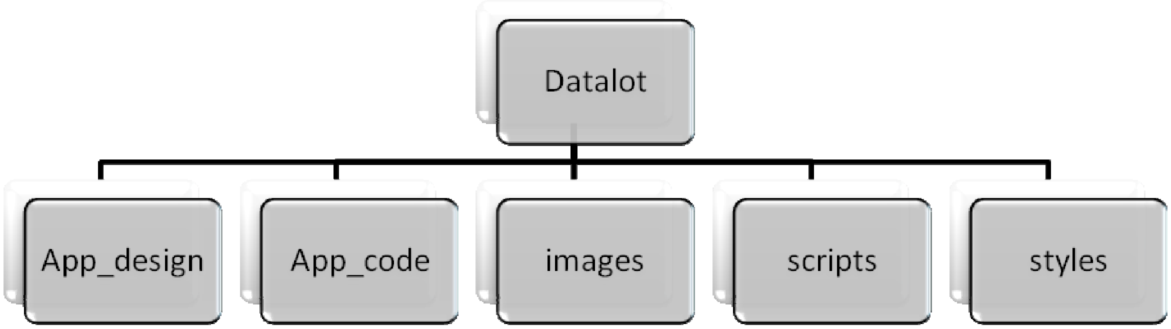


Figure3.3 Folder Structure

The folder organization shown in Figure 3.3 above is explained in Table3.3.

Table 3.3 Folder Structure

Folder	Description
Datalot	This is the parent folder which holds all the subfolders.
App_design	This folder holds all the PHP files which generate the HTML to render the pages.
App_code	This folder holds all the PHP files which handle the application logic, database connections and utility methods.
images	This folder holds all the images used by the tool.
scripts	This folder holds all the Javascript files used in the tool for client application logic.
styles	This folder holds all the CSS files used for styling the pages.

The contents of the App_design and App_code folders are described in details in Table 3.4

Table 3.4 Code Files

File	Description
App_design	
login.php	This file is rendered as the starting point for any of the users to access the tool. On click of *Login* user is redirected to login_submit.php file located in the App_code folder where the user credentials are authenticated and the session variables for the user depending on the type of user are set. The user is then redirected to the respective user view pages depending on the type of user.
create_admin.php	This page is rendered when a new administrator user registers into the system.
admin_programs.php	This is the *Home* page for the Admin View of the tool. This page checks if the user is an administrator by including admin_check.php file which redirects the user to the login page if administrator session parameters are not set. If the user is verified as an administrator user, this page renders the list of programs associated with the particular administrator.
admin_program_welcome.php	After the administrator user selects a program from the *Home* page s/he is redirected to this page. It is the entry point into an individual program. The top navigation menu holds the different modules associated with Admin View of the program as shown in Figure 2.1.The content of the page includes help notes explaining the functionalities and features provided by each of the menu items in the top navigation.
participant_type.php	The administrator is navigated to this page when they choose the first sub menu item under the *Manage Participant Type* link in the top menu bar. This page renders participant type information associated with the program.

File	Description
participant_info.php	This page is rendered when the administrator user selects the second sub menu item in the program menu bar. This page renders the participant information uploaded for the program into the system by the administrator. It also provides interface to add, edit or delete any of the participant information into the system. It also provides a file upload control for uploading the CSV files for multiple participant data upload.
manage_admin.php	This page is rendered when the administrator user selects the third sub menu item under the participant tab in the program menu. This page allows the administrator to add or remove program administrators as well as edit their own account information.
manage_sessions.php	This page is rendered when the administrator user selects the *Manage Sessions* tab in the program menu. This page lists down all the session information uploaded into the system by administrator users. It also allows them to add, edit or delete any of the session information. They can also upload multiple session information at once by uploading a CSV upload control on this page.
create_survey.php	This file is rendered when the administrator user selects the *Create Survey* link located beside each session row on the manage session.php page. This page allows administrator user to create surveys for the participants attending the session.
edit_survey.php	This page is rendered when the administrator user selects any of the existing surveys on the manage sessions page to modify.
report_status.php	This page is rendered when the administrator user selects the first sub menu under the *View Reports* tab in the program menu bar. This page renders the status of the surveys submitted or not submitted by the participants for the surveys selected by the administrator.

File	Description
manage_reports.php	This page is rendered when the administrator user selects the second sub menu item under the *View Reports* tab in the program menu. This page renders reports based on various filter criterions specified by the administrator user.
header.php	This page holds the header HTML as well as any other important includes which are generic over all the pages. This page is included on all the program pages for the Admin View of the tool. Any changes made to the header page are reflected across all the Admin View pages at once.
header_home.php	This is the header page for the administrator program list page. This does not contain the program menu bar specific to a particular administrator program.
header_home_pres.php	This page is included as header for the pages on Presenter View of the tool.
footer.php	This page holds the footer HTML for the tool. It is included in all the pages across the tool.
manage_reports_pres.php	This page is rendered in the Presenter View when the presenter selects the *View survey reports* tab in the menu bar. This page renders the anonymous results of the survey for the individual presenter sessions.
participant_home.php	This page is rendered when a participant logs into the system. This page lists all the surveys the participant is required to answer.
participant_survey.php	This page is rendered when the participant chooses to respond to a survey from the participant_home.php page.
App_code	
login_submit.php	This page is used for running the scripts to authenticate user credentials, setup the session parameters and redirect to the respective page based on the user type.
logout.php	This page is used to run the scripts required to logout any user from the application. It deletes all the session parameters and redirects the user to the login.php page.

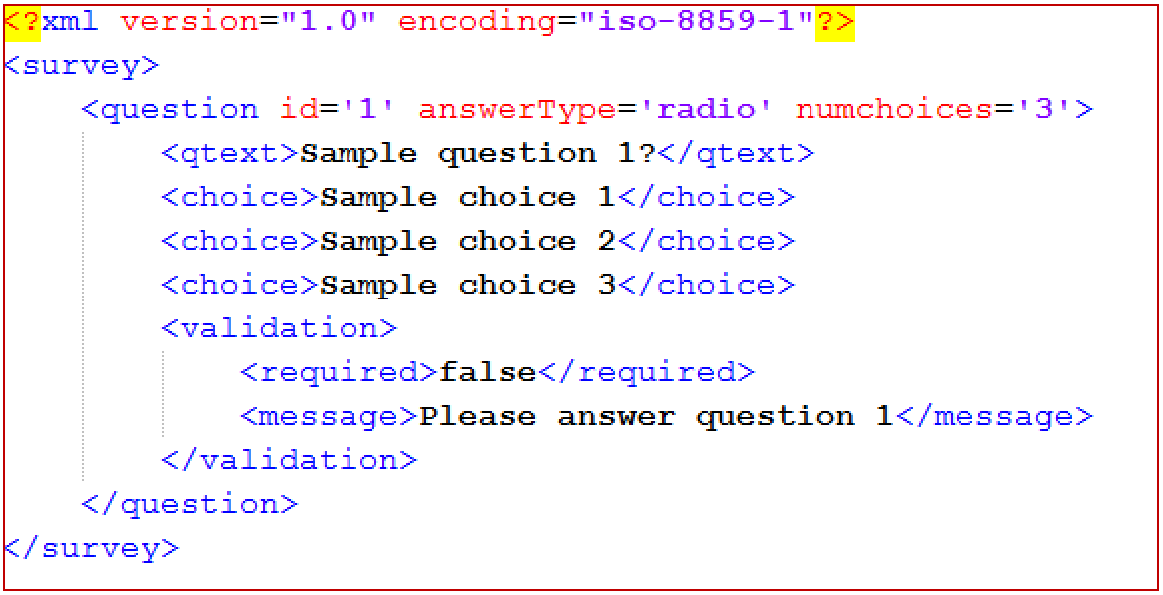
File	Description
utility.php	This file contains utility methods which are used across all the files in the project. This also includes the dbConnection.php file which is required for setting up a connection to the database. This file in included on each of the pages in App_design folder.
dbConnection.php	This file includes the script which connects to the database using the database credentials. If the database of the project changes than only the credentials in this file needs to be changed.
admin_check.php	This file is used for authenticating an administrator user on each of the pages in the Admin View of the tool. If the administrator session parameters are not set than the user is redirected to the login page.
admin_programs_code.php	This file holds application logic required for rendering the admin_programs.php page. This file connects to the database to provide data to the admin_programs.php page.
create_admin_code.php	This file holds the application logic required for rendering the create_admin.php page.
create_survey_code.php	This file holds the application logic required for rendering the create_survey.php page.
edit_survey_code.php	This file holds the application logic required for rendering the edit_survey.php page.
participant_type_code.php	This file holds the application logic required for rendering the participant_type.php page.
participant_info_code.php	This file holds the application logic required for rendering the participant_info_code.php page.
manage_admin_code.php	This file holds the application logic required for rendering the manage_admin_code.php page.
manage_sessions_code.php	This file holds the application logic required for rendering the manage_session_code.php page.
report_status_code.php	This file holds the application logic required for rendering the report_status_code.php page.
manage_reports_code.php	This file holds the application logic required for rendering the manage_reports.php page.
manage_reports_pres_code.php	This file holds the application logic required for rendering the manage_reports_pres.php page.

File	Description
participant_home_code.php	This file holds the application logic required for rendering the participant_home.php page.
participant_survey_code.php	This file holds the application logic required for rendering the participant_survey.php page.

3.1.2.2.1. Design of Survey Module

Survey creation and administration is an important aspect of this project. Program administrators collect assessment data by administering surveys for each session. While designing the survey creation module for the Admin View the most challenging question was to account for variable lengths of different surveys. Also the types of questions and several other question attributes needed to be stored and recreated when requested. As the survey module was the most dynamic module in the application it was not feasible to use the database for storing each question and its attributes in different tables and query each of them frequently. Also, the use of databases will limit the scalability to the number of attributes to the survey questions. The use of XML for storing the survey design and answers was an optimal solution to allow for the dynamically changing survey questions and question attributes. The XML are then stored as LONG TEXT into the database. The xml retrieved from the database is parsed using JQuery and rendered as HTML for the participants to answer the questionnaire. Participant answers are again embedded into the survey question XML and the new answer XML is saved into the database tables. When generating the survey reports survey XML's are parsed using PHP's simpleXML parsing methods. PHP simpleXML provides fast and easy to use parsing tools for parsing XML data. XML support provided by JQuery and PHP makes use of XML for designing an optimum solution.

The design of the survey XML is explained with a sample question XML in Figure 3.3. The <survey> tag encapsulates all the information required to render the complete survey to the user. Inside the survey tag there are several <question></question>. These tags hold all the information about a particular survey question. These tags are repeated within the <survey></survey> tags for each question present in the survey. Thus every time the survey administrator adds a question a new <question></question> tag holding the information about that particular question is created.



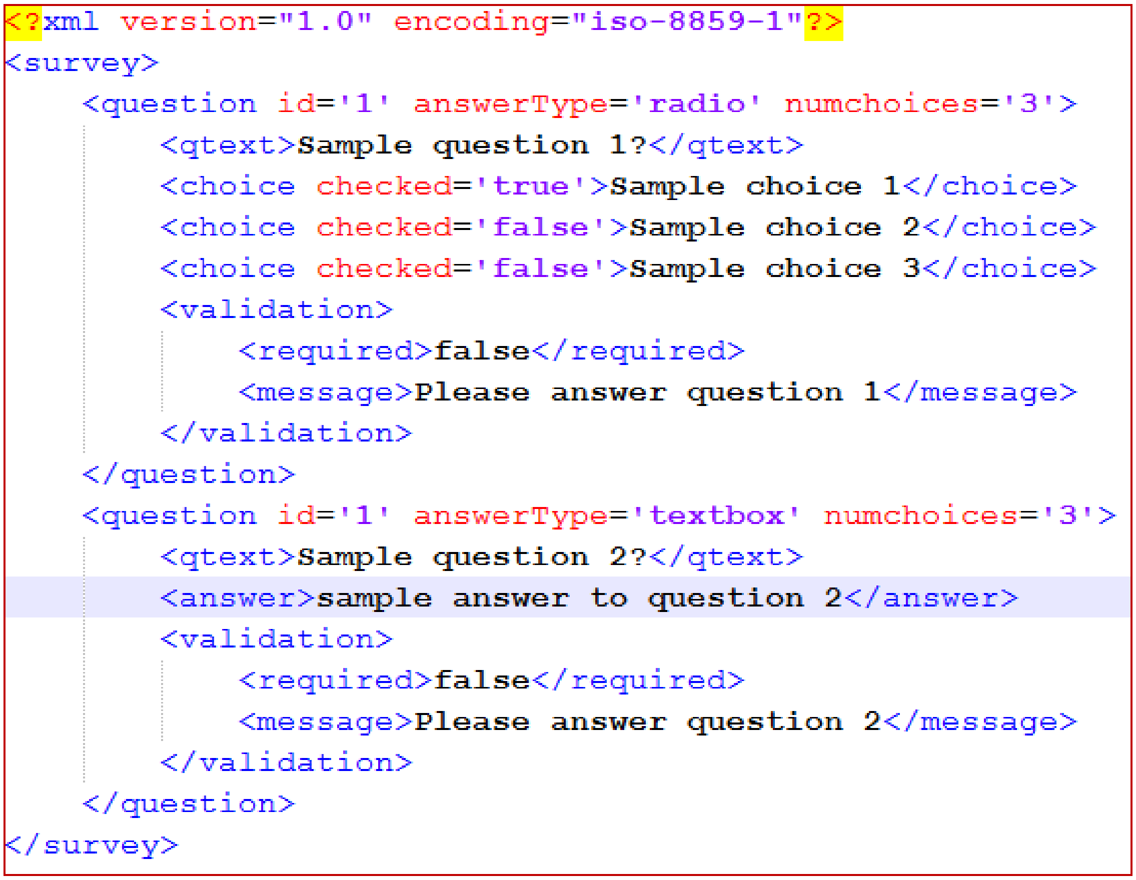
```
<?xml version="1.0" encoding="iso-8859-1"?>
<survey>
    <question id='1' answerType='radio' numchoices='3'>
        <qtext>Sample question 1?</qtext>
        <choice>Sample choice 1</choice>
        <choice>Sample choice 2</choice>
        <choice>Sample choice 3</choice>
        <validation>
            <required>false</required>
            <message>Please answer question 1</message>
        </validation>
    </question>
</survey>
```

Figure 3.3 Survey Question XML

The survey XML is created using Jquery when the administrator has completed editing the survey questions and decides to save it. Every question is identified by the id attribute present in the <question> tag. Also the type of answer for the question is stored in answerType attribute of <question> tag. In the case of radio buttons and checkboxes, the answer may have multiple choices. The number of choices is saved in numchoices attribute. The text of the question is stored between the <qtext></qtext> tags followed by the value of different answer choices between the <choice></choice> attributes.

The administrator may incorporate validation for specific survey questions. All the information about the validation is saved within the <validation></validation> tags. The current design of the survey supports validation of forcing a response for individual questions and a custom validation message. Information about both is saved within the validation tag and the child tags <required></required> and <message></message>. As is evident from this design the tool can be extended to include more validation features. These features can be included within the validation tag to add further support in the future.

When the program participants logs in to the system to answer survey questionnaires they are presented with the HTML which is generated by parsing the above XML. After the participant add their responses and submit the survey another XML similar in design to the question XML is generated with participant answers embedded in it. This answer XML is then saved into the database against the participant ID and survey ID and used later for generating reports. The design of the answer XML is shown in Figure 3.4. The answer XML in figure 3.4 shows the snapshot of two questions submitted by the participants. The first question with an answerType of radio has multiple choices as answer. The <choice></choice> tag has an added attribute checked='true' or checked ='false'. This attribute indicates the participant selection for the question. The second question is of the type textbox and has an <answer></answer> tag embedded after the <qtext></qtext> tag. The <answer></answer> tag holds the participant answer in case of text answer type questions.



```
<?xml version="1.0" encoding="iso-8859-1"?>
<survey>
    <question id='1' answerType='radio' numchoices='3'>
        <qtext>Sample question 1?</qtext>
        <choice checked='true'>Sample choice 1</choice>
        <choice checked='false'>Sample choice 2</choice>
        <choice checked='false'>Sample choice 3</choice>
        <validation>
            <required>false</required>
            <message>Please answer question 1</message>
        </validation>
    </question>
    <question id='1' answerType='textbox' numchoices='3'>
        <qtext>Sample question 2?</qtext>
        <answer>sample answer to question 2</answer>
        <validation>
            <required>false</required>
            <message>Please answer question 2</message>
        </validation>
    </question>
</survey>
```

Figure 3.4 Answer XML

3.2. Summary

This chapter described the framework of the tool. The functional design specification sub sections covered the functional use cases for each of the user views supported by the system. The technical design specifications explained in detail the code organization and design of the project.

CHAPTER 4. METHODOLOGY AND DISCUSSION

This chapter describes the design and methods used to evaluate the performance statistics and usability analysis of the tool. The tool was evaluated on the basis of the overall average time required to access different web pages on the tool with sufficient data by simultaneous group of users. Performance in terms of rendering a particular web page is crucial for usability of the tool when multiple participants are accessing it at the same time. One such situation arose when a customized web tool was created for collecting assessments from participants for the SPIRIT project. Approximately 100 participants were using the tool simultaneously to answer the survey questionnaire. The tool design was not optimum to support multiple requests, so the program administrators had to use different methods to collect assessments when the tool crashed under the load.

Performance testing DataLot for approximately 200 multiple requests provided an estimate of the performance when the tool is used for collecting assessments from multiple participants simultaneously. At the same time because the tool is designed for the administrators to manage and review their research data, a usability survey could help in improving the tool for future use.

A usability survey was conducted to evaluate the usability of the tool by the project administrators who might be the potential users of the tool in the future. The results of the usability survey could provide insights into how the tool may be improved for it to better serve administrators’ needs so that it can be used in the future. The performance of the tool in terms of how quickly different pages are rendered is also directly related to the usability.

4.1. Performance Statistics

Performance of the tool was evaluated by calculating the average time required for individual data pages to load for both the Admin View and the Participant View. The performance of the tool was to be tested for multiple requests to various web pages in the tool. There are different methods to create multiple requests from different sources. Multiple, simultaneous machines can be used to trigger simultaneous requests to the server, thus recreating a typical program scenario. There are several online tools that are available which simulate the above scenario by triggering multiple asynchronous requests from a single machine. Although these requests are not simultaneous because they are triggered from a single machine, the requests are still asynchronous and the scenario is very close to the typical case of parallel requests.

Fiddler, a free web debugging and the performance evaluation tool, was used to simulate the web traffic and record the performance statistics. Fiddler is an HTTP debugging proxy that logs all HTTP traffic between the host computer and the Internet. Fiddler inspects all HTTP traffic with incoming or outgoing data. Fiddler simulates asynchronous multiple requests to a page and provides various performance statistics. Fiddler is also used by Microsoft to build their own web debugging tools like Microsoft Fiddler Power Toy (Lawrence, 2005), and they also developed an Add-On called neXpert for Fiddler for some advanced debugging options. The requests created by Fiddler are not parallel but are asynchronous, thus the performance statistics obtained are worse than those with parallel requests.

Fiddler was used to simulate the web traffic to different data pages for both of the user views, and the performance statistics were documented. The performance tests were conducted with a machine connected to a modem (6KB/s) and using the Firefox and Google Chrome browsers. The results for both browsers were very close. The figures obtained through the tests are very close estimates of the true numbers but the performance of the tool can vary slightly, depending on the network connection, and the browser used.

4.1.1. Procedure

The following steps were followed to setup the test data into the system to perform the testing of the tool:

1. After the initial development of the tool on local machine, web and data space was requested from Purdue ECN to host the web tool. 10GB of web space and 10GB of table space were allocated on the Purdue engineering web cluster. The web tool and the database were deployed on the engineering cluster. The tool is now available at the following URL for the users to try out and use: https://engineering.purdue.edu/~ngoyal/Datalot/App_design/login.php?
2. A special administrator account was created for uploading the test data. The account credentials are:

   Username: performance@purdue.edu

   Password: test123
3. On the administrator's *Home* page, 10 programs were created for this administrator.
4. For *Program1*, 10 participant types were created on the *Manage Participant Types* page.
5. After the creation of program participant types, 200 participants were uploaded using the CSV upload method provided on the page.
6. 10 program administrators were created on the *Manage Admin Info* page.
7. After setting up all the participant data, program sessions were created on the *Manage Sessions* page. 50 sessions were created for this program by uploading the data using the CSV upload provided on the web page.
8. For the first 10 sessions, the participant type and session presenter data was edited using the *edit* link in each row.
9. New surveys were created for the first 10 sessions using the *Create Survey* link.
10. One of the surveys was created with 50 questions in it.
11. Other surveys were created by creating one survey with 20 questions and then using the *copy* functionality to copy the existing program surveys for each session.
12. The last module in the Admin View is the *View Reports* module. For testing this module, the survey with 50 questions was answered by

logging into the system as a participant. In order to create data for 200 participants, each answering the survey with 50 questions, a PHP script file was used to insert the answer data of one participant for all the remaining 199 participants. The script file for the same is located in the root folder on the server named *answerInsert.php*. The call to this file inserts 200 rows into the database with answers for each participant.

13. At this point all the data has been uploaded into the system for testing both the user views of the system.

The data uploaded into the system for the performance testing is summarized in the Table 4.1 below:

Table 4.1 Test Data

Attributes	**Number**
Admin	PERFORMANCE@PURDUE.EDU
Programs	10
Participant Types	10
Participants	200
Sessions	50
Surveys	10
Survey questions	50

4.1.2. Admin View

For each of the Admin View web pages, Fiddler was used to simulate the request for 50 administrators. Overall time elapsed for a single request for each of the data intensive pages was noted. The single request was reissued asynchronously by using Fiddler and the average performance statistics for multiple asynchronous requests to complete a round trip to US West Coast (Modem - 6KB/sec) were noted. The results of all the Admin View pages are summarized in Table 4.2. The average time to load for most of the administrator pages is less than 12 seconds, except for the *Manage Participant Info and View survey reports* page takes approximately 31 seconds to load. Not all the programs will have 200 participants, so the processing time should be faster. The

View survey reports page requires processing of each of the 200 participant XMLs with 50 questions each. The time to load this page for a single administrator request is approximately 47 seconds. These numbers are for a survey of 50 questions, which may not be very common. The time required for processing 50 asynchronous requests to the *View survey reports* page is around 71 seconds. Because most of the administrators will not be accessing the same page at the same time and these numbers are on the higher side to test for the worst case scenario, typical usage of the tool is expected to be more responsive.

Table 4.2 Statistics of Admin View data intensive web pages

Page Name	**Data on the page**	**Time elapsed for a single request**	**Average estimated time for each 50 asynchronous requests**
Admin Home	Lists down 10 programs for the administrator	0.574s	1.92s
Manage Participant Types	Lists down 10 participant for program1	0. 383s	2.16s
Manage Participant Info	Lists down data uploaded for 200 participants	1.426s	31.02s
Manage Admin Info	List down the details of 10 other program administrator.	0.476s	1.7s
Manage Sessions	List details of 50 sessions	1.007s	11.64s
View survey status	Status of 200 participants for a survey	1.103s	8.58s
View survey report	Responses of 200 participants for each of the 50 survey questions	46.469s	70.92s

4.1.3. Participant View

For each of the participant web pages, Fiddler was used to simulate the request for 200 and 500 participants. Overall time elapsed for a single request for each of the data intensive page was noted followed by the average performance statistics for multiple asynchronous requests calculated based on the estimates provided by Fiddler for a round trip request to US West Coast (Modem - 6KB/sec). The results of all the participant view pages are summarized in the Table 4.3. As can be seen the time required to recreate an HTML survey page with 50 questions for 200 simultaneous participants is less than 3 seconds, which makes the tool efficient in such scenarios.

Table 4.3 Statistics of Participant View data intensive web pages

Page Name	**Data on the page**	**Time elapsed for a single request**	**Average estimated time for each 200 asynchronous requests**
Participant Home	Lists down 10 surveys for the participant	0.267s	1.05s
Answer a survey page	A survey with 50 questions is created	0. 208s	2.75s

4.2. Usability Survey

An IRB approved usability survey to assess the usability of the tool for the project administrators involved in research involving data collection from human participants was conducted. The results of the usability survey are listed below. These results allowed us to assess the usability and design of the tool as well as estimate the usefulness of the tool to support data collection and management. Feedback received from in the results of the usability survey can be used to improve the design and features provided by the tool.

4.2.1. Usability Survey Results

The Usability survey with a small video tutorial for the tool was sent out on 22nd June 2011 to the 25 prospective users of the tool. Purdue public research database was used to find researchers involved in projects involving human participants. As of June 28, 2011, only one response has been received. The result of the response shows satisfactory performance and likelihood of using this tool in future for data management of research projects. Follow up emails were sent to some of the potential users but no responses were received as of 4th July, 2011.

4.3. Discussion

The tool was successfully implemented and deployed on the space provided on the engineering web cluster. The functional and technical specification of the tool were designed to accommodate common use cases for data collection and management needs of research involving human participants. The features provided by the survey design module for the survey administrator are limited but cover all the basic design features required to administer a simple survey to capture information. The XML design of the survey module allows for easy implementation of more complex and new features in the future without any changes in the database design. The XML for the surveys is only fetched once at the page load and then processed using JQuery scripting on the client side. This reduces the number of server trips every time the survey administrator adds or deletes a new question to the survey.

Use of Ajax XHR requests during the implementation of the code helped in minimizing the page refresh for every single request to the server. Most of the GET requests of the server on all the pages were handled using AJAX. This improves the performance of the system.

In the Admin View of the application, CSV file upload functionality to upload large number of participants or sessions help avoiding the task of submitting a request for every single entry. At the same time, administrators are also provided the functionality to download that data in CSV formats for archiving or mailing purposes. The CSV upload parses the CSV file and provides the administrator with very specific error messages in case of any errors. Also, the CSV download is implemented using a JQuery library which converts the HTML table data into CSV file format and submits the request to the server. This avoids the server side processing of the HTML tables.

Performance evaluation of the tool for the most data intensive web pages with a load of 50 simultaneous administrators indicates that the average time for most of the requests is less than 5 seconds. The most data intensive page - *View survey reports* takes around 60-70 seconds on an average to process all the XML's and display results. These figures are competitive enough to make the tool usable in future for real projects.

The project files are burned on the CD with the instructions to setup the project as well as the database scripts. The CD is submitted along with the final project of the report.

4.4. Recommendations for Future Work

The deliverable of this project was a functional generic tool which is usable for real projects. The features and functionalities provided by the tool are focused on serving the needs of program administrators. As this is a software product there is always a scope to increase the number of features provided. Below are some of the suggestions to improve the features and usability of the tool:

1. The design and look of the tool is currently based on a free web template. It can further be improved by a professional designer to make it more appealing and usable.
2. The number of features provided in the survey creation module can be increased to support more involved question types.
3. Adding the drag and drop feature to move the questions will also make it more usable.
4. Allowing including page breaks within a survey for surveys with large number of questions will be a helpful feature.
5. The types of custom validations for the survey questionnaire can be improved.

6. Support for generating quantitative reports for Likert scale questions like overall average score etc. can be included.
7. The CSV format only supports yyyy-mm-dd type of date format which is not a default type in Excel. Support of different date formats will be helpful for the program administrator.
8. The survey report status page can be improved to include a mail client so that the administrators can send out the follow up emails to the participants for completing the pending surveys.
9. The *View status report* page allows the administrator to view reports based on a session survey. So for viewing the status of all the survey, the administrators need to select the session and then the surveys every time. A broader filter which will allow the administrators to view result status of all the surveys at one go will be helpful.

REFERENCES

Agrawal, R., Gupta, A., & Sarawagi, S. (1997).Modeling Multidimensional Databases. *Data Engineering, Proceedings,13th International Conference.*

Andrews, K., Kappe, F., & Maurer, H. (2000) Serving Information to the Web with Hyper-G. *Institute for Information Processing and Computer Supported New Media (IICM), Graz University of Technology,* 27(6), 919-926.

Berner R. (1994, August 6). An update for resumes: Software lets computer do the choosing. *Patriot Ledger,* p. 25.

Blaha, M., Premerlani, W., & Rumbaugh, J. (1998) Database design using object-oriented methodology. *Communications of the ACM*, 31(4), 414-427.

Bongio, A., Ceri, S., Fraternali, P., & Maurino, A. (2000). Modeling data entry and operations in WebML. *Workshop Web and Databases* (WebDB 00), Springer-Verlag, Berlin, 2000, 201-214.

Brambilla, M., Ceri, S., Fraternali, P., Acerbis, R., & Bongio,A. (2005) Model-driven Design of Service-enabled Web Applications. *International Conference on Management of Data,* 2005, 851-856.

Ceri, S., Daniel, F., Matera, M., & Facca, F. (2007) Model-driven development of context-aware Web applications. *ACM Transactions on Internet Technology(TOIT),* 1(2).

Ceri, S., Fraternali, P., & Matera, M. (2002) Conceptual modeling of Data Intensive Web Application. *Internet Computing, IEEE,* 6(4), 20-30.

Ceri, S., & Fraternali, P. (2003) Architectural issues and solutions in the development of Data-Intensive Web Applications. *Proceedings of CIDR'03, Asilomar.*

Ceri, S., Matera, M., Rizzo, F., & Demalde, V. (2007) Designing Data Intensive Web Applications for Content Accessibility using Web Marts. *Communications of the ACM,* 50(4), 55-61.

Codd, E. (1982) Relational Database: A practical foundation of productivity. *Communication of the ACM,* 25(2), 109-117.

Commercenet. (1995, October). *The CommercenetlNielsen Internet demographics survey.*Palo Alto, CA: Author.

David J. Solomon,(2001). Conducting Web-based Surveys.*Practical Assessment, Research & Evaluation,*7(19).

Fraternali, P. (1999) Tools and approaches for developing data-intensive Web applications: A survey. *ACM Comput. Surv. 31*, 3 (Sept.), 227–263.

Gelder V. M.M., Bretveld, R.W., Roeleveld, N.(2010, December 1). Web-based Questionnaires: The Future in Epidemiology**.** *American Journal of Epidemiology*,172, 1292-1298.

Hammer, M., & McLeod, D. (1981). Database Description with SDM: A Semantic Database Model. *ACM Transactions on Database Systems (TODS),* 6(3), 351-386.

Kaye B.K. & Johnson T.J. (1999) .Research Methodology: Taming the Cyber Frontier. *Social Science Computer Review*, 17, 323-337.

Lawrence, E. (2005). Fiddler Power Toy – Part1: HTTP Debugging. Retrieved June 20, 2011 from http://msdn.microsoft.com/en-us/library/bb250446(v=vs.85).aspx.

Manfreda, K. L., Batagelj, Z. and Vehovar, V. (2002). Design of Web Survey Questionnaires: Three Basic Experiments. *Journal of Computer-Mediated Communication*, 7(0). doi: 10.1111/j.1083-6101.2002.tb00149.x

Medin, C., Roy, S. & Ann, T. (1999) World Wide Web versus mail surveys: A comparison and report. Paper presented at *ANZMAC99 Conference, Marketing in the Third Millennium*, Sydney, Australia.

Navathe, S. (1992) Evolution of data modeling for databases. *Communications of ACM,* 35(9), 112-123.

O'Neill, B. (2004). Collecting research data online: Implications for Extension professionals. *Journal of Extension*, 42(3).

Preciado, J.C., Linaje, M., Sanchez, F.,& Comai, S. (2005). Necessity of methodologies to model Rich Internet Applications*. Web Site Evolution, 2005. (WSE 2005) Seventh IEEE International Symposium,* 2005, 7-13.

Rhodes S, Bowie D, Hergenrather K(2003). Collecting behavioural data using the world wide web: considerations for researchers**.***Journal of Epidemiology and Community Health* ,57,68-73.

Schmidt, W. C. (1997). World-Wide Web survey research: Benefits, potential problems, and solutions. *Behavior Research Methods, Instruments, & Computers*, 29, 274–279.

Schmitt CH. (1997, March 2). Behind the wave: consequences of the digital age. *San Jose Mercury News,* 1S-5S.

Stanton, J. M. (1998). An empirical assessment of data collection using the Internet. *Personnel Psychology*, 51(3), 709–726.

Thompson, D. V., Hamilton, R. W. & Rust, R. T. (2005a) Feature fatigue: When product capabilities become too much of a good thing. *Journal of Marketing Research* 42(4), 431–42.

Wright, K. B. (2005). Researching Internet-Based Populations: Advantages and Disadvantages of Online Survey Research, Online Questionnaire Authoring Software Packages, and Web Survey Services. *Journal of Computer-Mediated Communication*, 10: 00. doi: 10.1111/j.1083-6101.2005.tb00259.x

Yang J., Karlapalem K., & Li Q. (1997). Algorithms for materialized view design in data warehousing environment. *In:Proc. of VLDB. Athens, Greece,* 136–145.