

Regis University

ePublications at Regis University

Regis University Student Publications
(comprehensive collection)

Regis University Student Publications

Spring 2006

Design and Implementation of a Web-Based Lead Management System Using Oracle and Coldfusion Technologies

James Darschewski
Regis University

Follow this and additional works at: <https://epublications.regis.edu/theses>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Darschewski, James, "Design and Implementation of a Web-Based Lead Management System Using Oracle and Coldfusion Technologies" (2006). *Regis University Student Publications (comprehensive collection)*. 399.

<https://epublications.regis.edu/theses/399>

This Thesis - Open Access is brought to you for free and open access by the Regis University Student Publications at ePublications at Regis University. It has been accepted for inclusion in Regis University Student Publications (comprehensive collection) by an authorized administrator of ePublications at Regis University. For more information, please contact epublications@regis.edu.

Regis University
School for Professional Studies Graduate Programs
Final Project/Thesis

Disclaimer

Use of the materials available in the Regis University Thesis Collection ("Collection") is limited and restricted to those users who agree to comply with the following terms of use. Regis University reserves the right to deny access to the Collection to any person who violates these terms of use or who seeks to or does alter, avoid or supersede the functional conditions, restrictions and limitations of the Collection.

The site may be used only for lawful purposes. The user is solely responsible for knowing and adhering to any and all applicable laws, rules, and regulations relating or pertaining to use of the Collection.

All content in this Collection is owned by and subject to the exclusive control of Regis University and the authors of the materials. It is available only for research purposes and may not be used in violation of copyright laws or for unlawful purposes. The materials may not be downloaded in whole or in part without permission of the copyright holder or as otherwise authorized in the "fair use" standards of the U.S. copyright laws and regulations.

Design and Implementation of a Web-Based Lead Management System Using Oracle and ColdFusion Technologies

by

James M. Darschewski
James.Darschewski@gmail.com

A Project Report submitted in partial fulfillment of the requirements for the degree of
Master of Science in Computer Information Technology

School for Professional Studies
Regis University
Denver, Colorado

April 30, 2006

An Abstract of a Project/Practicum Report Submitted to Regis University School for Professional Studies in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Information Technology

Design and Implementation of a Web-Based Lead Management System Using Oracle and ColdFusion Technologies

by

James Darschewski

April 30, 2006

This project focuses on developing a Web-based lead management application to facilitate Affordable Residential Communities (ARC) in managing their customers more efficiently and providing them with a method of storing customer data in a central repository. ARC has sales staff that are distributed throughout the United States, thereby creating a need for an application that can be used where an Internet connection is available. The application was developed using a combination of ColdFusion and Oracle, however, the overall success of the application does not depend on the technology incorporated, rather it is dependant on a successful design and an easy to use interface. Research on the different methodologies available for developing a Web-based application, the effect of different design factors on the users of the application, and ultimately how the design can determine the success of the application was performed.

Acknowledgments

I would like to acknowledge my wife, Tricia, and my 2 children, Jimmy and Mary Sheridan. I dedicate this degree to them for providing me with the motivation to complete this project, as well as the MSCIT program.

Table of Contents

Chapter 1 - Introduction	1
Statement of the problem	1
Relevance, significance or need for the project.....	2
Barriers and/or issues	3
Research questions to be answered	4
Limitations and scope of the project	4
Summary.....	5
Chapter 2 – Review of Literature and Research.....	7
Overview of all literature and research sources.....	7
<i>Introduction.....</i>	<i>7</i>
<i>Web Application Technologies</i>	<i>7</i>
<i>Relational Database Technologies</i>	<i>9</i>
<i>CRM</i>	<i>11</i>
<i>Server-Side Language Technologies.....</i>	<i>13</i>
<i>Application Methodologies</i>	<i>15</i>
<i>Web Site Design</i>	<i>21</i>
<i>Security</i>	<i>22</i>
<i>Conclusion</i>	<i>24</i>
Research methods to be used	25
Literature and research that is relevant to the project	25
Summary of what is known and unknown about the topic.....	27
The contribution this project will make to the field	27
Chapter 3 – Methodology	29
Life Cycle models followed.....	29
Specific procedures	31
<i>Application Development.....</i>	<i>31</i>
<i>Lead Assignment</i>	<i>33</i>
<i>Application Coding</i>	<i>34</i>
Formats for presenting results/deliverables	34
Review of deliverables	35
Resource requirements.....	36
Outcomes	37
Summary.....	38
Chapter 4 – Project History	40
How the project began.....	40
<i>The Application.....</i>	<i>48</i>
<i>The Database</i>	<i>62</i>
How the project was managed	64
Significant events/milestones in the project.....	65
Changes to the project plan	66
Evaluation of whether or not the project met project goals	67
What went right and what went wrong in the project	68
Project variables and their impact on the project	69
Findings / analysis results.....	71
<i>Application Analysis</i>	<i>71</i>

<i>Data Analysis</i>	72
Summary of results	75
Chapter 5 – Lessons Learned and Next Evolution of the Project	77
Lessons learned from the project experience	77
What would have been done differently in the project	79
Whether or not the project met initial project expectations	80
Next stage for the project if it were continued	80
Conclusions / recommendations	81
Summary	82
Works Cited	84
Appendix A	87
Appendix B	88
Appendix C	89
Annotated Bibliography	105

List of Figures

Figure 1 - Web Application Components (Chen and Heath, 2005).....	9
Figure 2 – Waterfall Methodology (Fioravanti, 2005 p. 73).	15
Figure 3 - Spiral Model (Boehm, 1988).....	17
Figure 4 - Prototyping Methodology (Fioravanti, 2005 p. 78)	18
Figure 5 - Three Menu Styles Studied; simple selection menu (Left), global and local navigation aid menu (Middle), and pull-down menu (Right) (Yu et al., 2001).....	21
Figure 6 - Possible assignment of a potential sales lead (Darschewski, 2006).....	34
Figure 7 - Required Lead Information (Darschewski, 2006).....	41
Figure 8 - Lead Management Organization Chart. Blue dashed lines indicate who leads can be assigned to, while solid lines indicate assignment of leads by superiors (Darschewski, 2006).	42
Figure 9 - Job roles and job role creation hierarchy (Darschewski, 2006)	44
Figure 10 - Assignment of leads hierarchy represented in table form (Darschewski, 2006)	44
Figure 11 - Pull-down menu system with admin job role (Darschewski, 2006)	46
Figure 12 - Welcome screen of the application (Darschewski, 2006).....	48
Figure 13 - Login Page (Darschewski, 2006).	48
Figure 14 - Forgot Password Page (Darschewski, 2006).....	49
Figure 15 - Forgot Password Page (Darschewski, 2006).....	49
Figure 16 - Activate Password E-mail (Darschewski, 2006).....	49
Figure 17 - Initial Login Page (Darschewski, 2006).....	50
Figure 18 - Create Profile Page (Darschewski, 2006).	51
Figure 19 - Edit Agent Profile Selection Page (Darschewski, 2006).	51
Figure 20 - Edit Agent Profile Data Page (Darschewski, 2006).....	52
Figure 21 - Edit Your Profile Page (Darschewski, 2006).....	52
Figure 22 - Delete Profile Page (Darschewski, 2006).	53
Figure 23 - Change Password Page (Darschewski, 2006).....	53
Figure 24 - Enter Lead Page (Darschewski, 2006).....	54
Figure 25 - Show Leads Page (Darschewski, 2006).....	55
Figure 26 - A Subordinate Agents Leads (Darschewski, 2006).	55
Figure 27 - Detailed Lead Information (Darschewski, 2006).....	56
Figure 28 - Lead Documents Page (Darschewski, 2006).	57
Figure 29 - Enter Out Of Market Lead (Darschewski, 2006).....	57
Figure 30 - Search Leads Page (Darschewski, 2006).	58
Figure 31 - Search leads Results Page (Darschewski, 2006).....	58
Figure 32 - Close Lead Page (Darschewski, 2006).....	59
Figure 33 - Search Lead Documents (Darschewski, 2006).	59
Figure 34 - Search Lead Documents Result Page (Darschewski, 2006).	60
Figure 35 - Add Marketing Source Page (Darschewski, 2006).....	60
Figure 36 - Edit Marketing Source Selection Page (Darschewski, 2006).	61
Figure 37 - Example Lead Management Report Flow (Darschewski, 2006).	62
Figure 38 - Send Message Page (Darschewski, 2006).....	62
Figure 39 - ARC Move-Ins Less ARC Move-Outs (Darschewski, 2006).....	72
Figure 40 - Percent of ARC Home sites occupied (Darschewski, 2006).....	73
Figure 41 - ARC Home Sales (Darschewski, 2006).....	74

List of Tables

Table 1 - Howcroft & Carroll Methodology (Howcroft & Carroll, 2000).....19

Chapter 1 - Introduction

Statement of the problem:

In business a key to being successful is being able to identify who your customers are and evaluating their needs. Most businesses are not only comprised of a single customer; therefore, there is a need to retain customer information and to track the customer from initial contact to the point of sale. While there are numerous *off the shelf* products that can store customer data, often times they do not easily allow for modifications and many of these products would require some sort of wide area network (WAN), which could incur extensive costs, in order for highly distributed sales people to access the application.

As of this writing, Affordable Residential Communities (ARC) is currently the largest owner and operator of manufactured home communities in the United States, managing over 57,600 home sites in more than 278 properties across 24 states. Prior to the implementation of this application, ARC used Microsoft Excel spreadsheets to track customer inquiries and possible sales leads. However, using this method did not provide an easy way for ARC to assign leads to the sales staff nor did it provide a way for the sales staff to track and monitor the lead until the point of sale. The Excel spreadsheets were also used as traffic logs for ARC's communities which were sent to district level and regional level personnel for evaluation.

ARC needed an efficient system in place which would eliminate the need for the Excel based system and one which would facilitate in the assigning, tracking and reporting on possible leads who want to move into one of their communities or who want to take part in one of the housing options ARC offers. The Excel spreadsheets were the

only tool upper-level personnel had to report how much traffic was moving through a community and it was only by combining and manipulating these spreadsheets that they were able to see the traffic at a district or regional level. Currently, ARC uses Oracle Report Server to generate reports based on data stored in Oracle databases which the company relies on for its day to day operations, its financial reporting, and its property management data; it is in the reporting system that the lead management reports were to be integrated.

Throughout this paper this author will touch on the lead tracking and assignment problems encountered by ARC and document the design, development and implementation of a customer relationship management (CRM) tool that can be used by numerous sales people via an Internet connection and a Web browser.

Relevance, significance or need for the project:

At ARC there existed a void between the sales staff and potential customers. Since leads were being tracked using Excel or other tools by individual sales people, it was difficult for the corporate office and the sales managers to see what happened to those leads if the sale person no longer worked for ARC. By providing the sales staff with tools to help them retain, assign and track possible sales leads, this project attempts to fill the void that existed between the sales staff, the customer, and ARC's upper management. The Internet is a powerful source of information, but it can also be used as a *portal* to a corporate intranet that can host Web-based applications. In this project, the Internet was used as a *portal*, a means of connecting all of ARC's sales people, allowing them to store pertinent customer information in a central repository in an Oracle database.

ARC needed this repository for all of its possible sales leads in order to develop new marketing and sales initiatives that would better suit the needs of its customer base. The inherent need for a central repository of potential customers, along with the fact that sales people are not always working out of a single office, reinforced the need for a Web-based CRM solution.

Barriers and/or issues:

During the research, design and implementation of this project this author encountered the following barriers and issues:

1. *Time Management:* The coordination of ample time to meet with management of ARC to work out design specifics of the application, as well as finding suitable time to work, code, research and write this paper.
2. *Commitment:* This project took the commitment of both this author as well as the staff of ARC. There were many parts of the CRM application that required testing and feedback. If ARC were not able to commit to providing the proper staff to aide in the testing and debugging of the application then it would not have been successful.
3. *Research skills of the author:* This author has limited experience in authoring a research publication of this magnitude and in this format. To overcome this barrier it was necessary to incorporate the use of the Internet as well as the knowledge and experience of colleagues.

Research questions to be answered:

Upon reviewing the research literature used for this project the following questions were answered:

1. Can a viable Web-based solution be created with the technology currently in use at ARC?
2. Is it feasible to use the Internet as a suitable backbone for accessing this application?
3. Can ARC's sales be expected to increase by providing the sales staff with a tool they can all use?
4. Will this application allow ARC to collect the data they need so marketing and sales events can be steered towards the needs of the customers?
5. Is a custom built Web-based application a better solution than an *off the shelf* product?

Limitations and scope of the project:

The scope of this project was limited to providing ARC a company-wide solution for tracking and assigning sales leads. The application was designed specifically for ARC and was by no means meant to be used as an off the shelf product, due to the fact that the needs of ARC were unique in terms of the information they wanted to collect and the business model they used. This application's design was based on the specifications given to this author, and both tested and accepted by ARC's employees.

Summary:

Affordable Residential Communities is a small to mid-sized company with approximately 1500 employees and of those 1500 employees there are roughly 150 sales personnel, which do not include the community managers who can also sell homes. The sales staff at ARC did not have a consistent way of tracking potential customers. Some of the sales staff used Microsoft Excel to track customers while others used a pen and paper or some other proprietary method. ARC was looking for a cost-effective solution that would allow its sales staff to be able to enter potential customer information and track those customers. The solution ARC needed would not only allow for the input of potential customers, but would also automatically assign a customer to a sales agent if the person who entered the customer was not a member of the sales staff. ARC believed that a custom built CRM tool would not only help in creating future marketing initiatives targeted at their specific customer base, but also felt it would help maintain good relationships with current, as well as future customers. ARC also believed they would see an increase in the sales, leases and rentals of its manufactured homes.

ARC has remote offices throughout the United States. All of these offices have an Internet connection and most of them utilize a broadband connection. ARC wanted to leverage the power of the Internet by using it as a *portal* in such a way that their sales staff would be able to input customer data on a web page and have that data stored in a database at their corporate office. Ideally ARC wanted to make use of the technology they had in place to build and host this application. This technology included an Oracle 9i relational database management system (RDBMS) solution running on Linux, Microsoft Internet Information Services (IIS) 5.0, and ColdFusion MX.

While this project followed standard research methodologies, as well as standard software development life cycle (SDLC) procedures, the author anticipated several obstacles. One of these obstacles was in the management of time due to the fact that this author was not the only individual involved in this project and would have to accommodate and respect the time restrictions of the staff at ARC. ARC needed to take time away from members of their sales staff to test individual portions of the application to make sure it was progressing and developing while maintaining the requirements set forth. Another obstacle was for ARC to fully commit to this project and follow it through to the end. If they were to lose focus of what they were trying to gain by having a central location for customer information then the project was bound to fail. Lastly, the author of this project never attempted a project of this magnitude. Therefore, finishing the application within the timeline given could have proven to be difficult.

Chapter 2 – Review of Literature and Research

Overview of all literature and research sources:

Introduction:

Woojung Suh, author of the book *Web Engineering: Principles and Techniques*, states that “since the advent of the Web, every aspect of our lives and organizational activities has changed dramatically” (p. vi). The World Wide Web has become extremely important and has had a significant impact in business where the use of Web technologies has rapidly increased (Standing, 2002; Suh 2005). Many businesses view Web technologies as a vital tool in their day-to-day operations and they have made every effort to expand the development, use and maintenance of Web-based applications (Suh 2005). Expanding on Suh’s view of Web technologies, John Artz (1996) believes that corporate Web applications, specifically those using multimedia hypertext software based on the TCP/IP protocol suite, provide corporations easy access to a multitude of “corporate information resources for internal and external users on a wide variety of platforms in geographically distributed locations” (p. 64).

Web Application Technologies:

Web applications are built using a number of different technologies that work with each other to produce the desired product. Hassan and Holt (2005), along with Westman (2002), advocate that Web applications are not just made up of static hyper-text markup language (HTML) pages but also include dynamic pages. Suh (2005) expands on this idea defining a Web application as “any application program that runs on the Internet or corporate intranets and extranets” (p. 77). Moreover, Suh (2005) suggests three types

of Web applications: static Web documents, interactive Web applications and complex Web-based database systems.

Static Web pages are usually written entirely in HTML while dynamic, or active pages, are similar in structure to static pages but are written in languages such as VBScript, Active Server Pages (ASP), Java, JavaScript (Hassan & Holt, 2005), ColdFusion, or PHP (Westman, 2002), which contain active control code. In dynamic Web pages the HTML tags control *how* the content will be displayed on the users screen while the control code offers more complex functionality, depending on the framework chosen, to determine *what* to display (Hassan & Holt, 2005; Brooks-Bilson, 2003; McNeil & Robin, 2000).

There are numerous frameworks one can use to develop applications for the Web but they all operate in a similar fashion. If a client browser requests a dynamic Web page, that page is preprocessed by the Web server. The control code in the page is executed and the result of the executed code is combined with the HTML tags in the page and the result is displayed to the user's browser (Hassan & Holt, 2005).

According to Chen and Heath (2005) there are five major components to a database backed Web application. Figure 1 shows how these components act together to deliver a dynamically generated Web page. The Web server and application server are often times one and the same, where the Web server runs the software that compiles the HTML code to be displayed on the client's browser and the application server is responsible for managing session state, enforcing business rules and processing logic (Chen and Heath, 2005). The database server is the component that provides access to and storage of data pertaining to the application and often times has database

management capabilities (Chen and Heath, 2005). When a client browser requests a page from the Web server the server retrieves the requested data, if any, from the database and then the server compiles the underlying programming language on that page which subsequently is delivered to the client browser in a readable form.

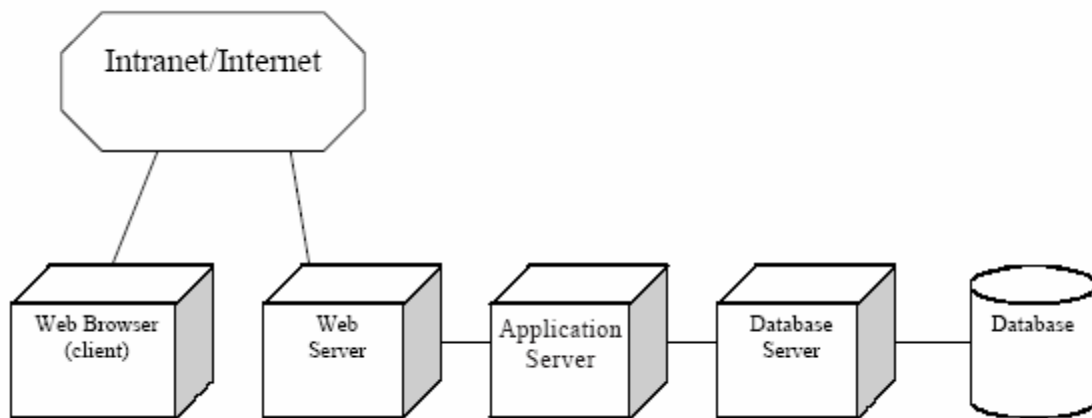


Figure 1 - Web Application Components (Chen and Heath, 2005).

Relational Database Technologies:

An important part of developing a Web-based application is how to store the data that will be used or collected. There are numerous ways one can store the application data, however, Westman (2002) suggests the use of a good RDBMS. RDBMS software allows an organization to centralize and efficiently manage their data while also providing access to that data by other programs (Laudon & Laudon cited in Liao, 2005). According to Lemieux (1996) a database system should consist of the following components which need to interact together to ensure the meaning of the database system is preserved:

- People.
- A definition of data structures and their inherent constraints.

- The expression of the data structures as real objects.
- The definition of a set of inference rules that may be used to derive facts from pure data.
- The definition of explicit constraints.
- The definition of a set of business functions.
- Software components: database engine and operating system.
- Appropriate hardware.

The previous list is very general and does not specifically outline what a relational database should consist of, but rather shows the components that need to interact in a database system. In his book *Database Systems*, Beynon-Davies (2004) writes about the 12 rules set forth by Dr. E.F Codd, a researcher for IBM and considered by many to be the father of relational databases (Loshin, 2001), that describes a true relational database system. Using these rules Beynon-Davies (2004) scored Oracle's RDBMS a 10 out of a possible 12. He states that the Oracle product uses tables as its primary data structure, it utilizes primary as well as foreign key constructs, uses null characters to represent missing or incomplete data, offers an on-line catalog and it uses SQL as its defined interface (Beynon-Davies, 2004).

There are numerous database systems available that would provide a sufficient backend to a Web-based application. Microsoft Access, a DBMS originally designed for the desktop market and non-sophisticated users, could potentially be used for Web-based applications but is limited in the features it offers. Access lacks a fully-formed method of transaction management, is limited in terms of the volume of data it can manage satisfactorily, and lacks a system catalog which is essential for good database

administration (Beynon-Davies, 2004). Microsoft also has another database product available called SQL Server which is a “comprehensive database platform providing enterprise-class data management with integrated business intelligence tools (“What is SQL Server” para. 1)”. Other enterprise level databases include DB2, Informix and Sybase. If affordability is a concern there are free or open source databases available, which include FrontBase, MySQL and PostgreSQL. These open source databases may not contain all of the features found in an enterprise level DBMS; however, they may still be sufficient for building powerful Web-based applications (Westman, 2002).

CRM:

Bob Thompson (2002), CEO of CustomerThink Corp and founder of CRMGuru.com, writes that “customer relationship management (CRM) is a business strategy to select and manage the most valuable customer relationships” (para. 4). Furthermore, he proposes that “CRM requires a customer-centric business philosophy and culture to support effective marketing, sales and service processes” (Thompson, 2002 para. 4). Increasingly, sales people are being asked to use new technologies, such as CRM tools or sales force automation (SFA) tools, to increase their sales and overall efficiency (Buehrer, Senecal, Bolman, Pullins, 2005). Roh, Ahn and Han (2005) agree with Buehrer et al., while also expanding on their ideas by noting that many firms have utilized CRM systems for improved business intelligence, better decision making, enhanced customer relations, and good quality of services and product offerings.

Often when a CRM project is started, it is common for organizations to expect a substantial payback, increased revenue, reduced cost, loyal customers, real-time customer

information and satisfied CRM users (Roh et al., 2005). However, implementing a CRM system that is highly effective may carry a high initial cost (Sohn and Lee, 2005) due to system equipment and a commitment of dedicated resources and services (Roh et al., 2005). For this reason it may be necessary for organizations to perform a cost of ownership (COO) analysis on such systems, evaluating the purchase cost of the system, the operating costs and utilization over its lifetime (Sohn and Lee, 2005).

Roh et al. (2005) propose that an important starting point for effective CRM implementations and deployments is an enterprise-wide understanding of the factors that lead to CRM success: process fit, information quality and system support.

Process fit. Roh et al. (2005) suggest that to leverage marketing and sales efforts it is imperative for the CRM system to be designed around an extensive understanding of the CRM process, thereby impeding the CRM system initiatives and can be imperative in the success of the system.

Customer information quality. A key to the success of a CRM system is to know your customers; however, merely gathering customer information is not enough. A successful CRM system should be able to make use of the customer data through customer information analytics (Roh et al., 2005). Customer information analytics is more than just mining the data about a customer from the CRM system, rather, it allows users of the system to build insight into customer and market behaviors (Roh et al., 2005) while allowing organizations to make better decisions in retail marketing and appropriately adapting to changing customer needs (Chen, Chiu & Chang, 2005).

System support. A review of the literature suggests that in terms of system support there are multiple factors that can determine if the CRM system will be

successful which relate to the system itself. Swanson (1974, cited in Roh et al., 2005) indicates that reliability, response time, and ease of terminal use are important. Hamilton and Chervany (1981, cited in Roh et al., 2005) emphasize data accuracy, reliability, completeness, system flexibility and ease of use. Seddon (1997, cited in Roh et al., 2005) believes that the consistency of the user interface, quality of documentation, and sometimes the quality and maintainability of the program code are important factors in the success of the system.

The literature indicates that Chen et al. (2005) believe that a successful CRM system will allow an organization to make more informed decisions regarding retail-marketing. However, the Internet is changing the focus of marketing from *supplier centric* to *customer centric* and, with this reverse marketing, the customer becomes the starting point for marketing activities (Sharma & Sheth, 2004).

Server-Side Language Technologies:

Brooks-Bilson (2003) writes that ColdFusion is designed for rapid application development (RAD) through the use of HTML-like tags which make operations such as connecting to a mail server or performing queries on a database much simpler. ColdFusion is available for numerous operating systems (Linux, Solaris, Microsoft Windows, and IBM AIX) and Web servers (Microsoft Internet Information Services, Netscape Enterprise Server, Apache, iPlanet Enterprise Server and ColdFusion's standalone Web server) (Brooks-Bilson, 2003). Churvis, Helms, Arehart & Churvis (2003) agree with Brooks-Bilson (2003) that ColdFusion is one of the easiest of the Web-programming languages to use. ColdFusion can interface with many different

technologies, for instance, Java objects, custom tags written in C++ or Java, JSP pages and tag libraries, and the operating system. This interfacing is what makes ColdFusion easily extensible (Churvis et al., 2003). ColdFusion possesses the ability to communicate with several types of databases allowing for Web-based access to data stored in popular RDBMS systems like Oracle, Microsoft SQL Server and MySQL (Brooks-Bilson, 2003).

PHP Hypertext Preprocessor (PHP), like ColdFusion, is a server-side language because it actually runs on a Web server (Sklar, 2004). PHP is similar to ColdFusion in other ways as well. Instructions in a PHP page cause the PHP interpreter on the Web server to output the page (Sklar, 2004); just as the ColdFusion application running on a Web server interprets and outputs ColdFusion Markup Language (CFML) pages. PHP distributions are available for Linux, as well as Microsoft Windows, Mac OS X, Solaris, and other flavors of UNIX. Database connections are also possible with PHP which allow for communications with the same database systems that ColdFusion can communicate with; though it may not be as easy to initiate these connections. Unlike ColdFusion, PHP is an open source project, therefore, there are not any licensing fees or any other kind of fees involved in using it (Sklar, 2004).

ASP.NET or Active Server Pages (ASP) is a server-side language, like both PHP and ColdFusion. It is developed by Microsoft and based on the .NET Framework and the Common Language Runtime (Liberty & Hurwitz, 2005). According to Microsoft (“.NET Framework Developer’s Guide”, n.d.), “ASP.NET is supported on Windows 2000, Windows XP Professional, and the Windows Server 2003 family for both client and server applications” (para. 1). ASP.NET is freely downloadable on Microsoft’s website, but requires Internet Information Services as its Web server (“.NET Framework

Developer's Guide", n.d.). Dissimilar to ColdFusion and PHP, ASP.NET must be programmed in a language supported by the .NET Common Language Specification; C#, Visual Basic .NET, Managed C++ and Jscript.NET (Liberty & Hurwitz, 2005). In ASP.NET the code may be separate from the HTML and rather than the code being interpreted it is compiled; this allows for the creation of "larger, easier to scale and easier to maintain websites" (Liberty & Hurwitz, 2005 p. 4).

Application Methodologies:

Chen and Heath (2005) state that "Web application development lacks standards and structured methodologies" (p. 77). By making some slight modifications, the Waterfall and Prototyping methodologies can still be used for Web-based application development (Chen and Heath, 2005); however, the adaptations of the existing methodologies are still based on the software development life cycle (SDLC).

The traditional Waterfall model divides project development into five phases, to be executed one after the other (Figure 2), as well as a phase before and a phase after to complete the cycle (Fioravanti, 2005).

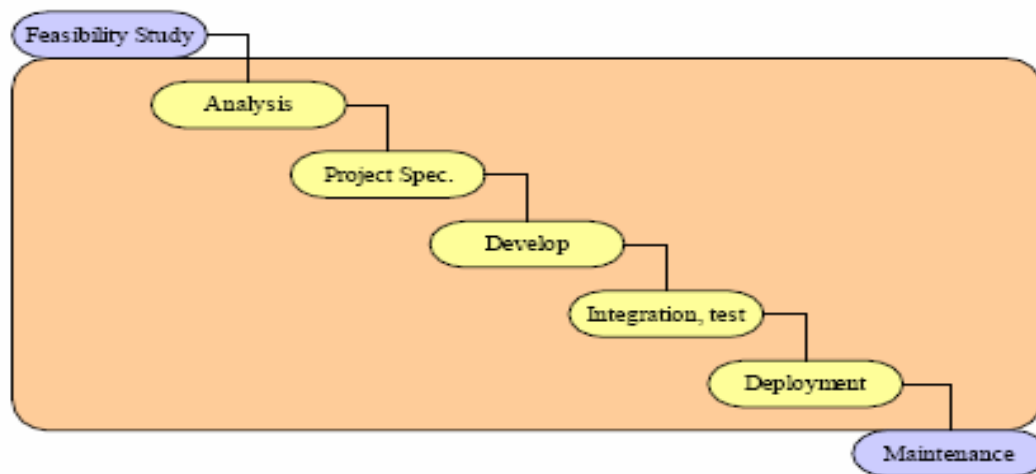


Figure 2 – Waterfall Methodology (Fioravanti, 2005 p. 73).

Powell, Jones and Cutts (1998, cited in Chen and Heath, 2005) suggested a modified Waterfall model that consists of *whirlpools*. Their model is still made up of the same initial phases of the Waterfall model, however, the *whirlpools* are actually the analysis phase and the project specification phase of the traditional model which iterate multiple times so the developer can gain a better understanding of the users' requirements. While it is important to have a full understanding of each of the phases of this model in developing an application, an in depth discussion of these phases is beyond the scope of this paper.

The Spiral methodology often finds its place in product development in medium and large sized companies (Fioravanti, 2005). Fioravanti (2005) believes this is the case because it is “formal enough to match quality certification needs of companies and is also agile enough to avoid compelling people to draw all the analyses and specifications in advance” (p. 81). According to Fioravanti (2005) the Spiral model is an iterative approach to developing the final product. Boehm (1988) does a better job at explaining how the model works. He describes it as a risk-driven approach to the software process rather than strictly a specification-driven or prototype-driven process.

Boehm's (1988) illustration, Figure 3, shows the radial dimensions as the total incurred costs involved in accomplishing the steps to date, while the angular dimensions represent the progress made in completing each cycle of the spiral. It is Fioravanti's (2005) opinion that the Spiral model is best used slightly modified, in which the requirements and specifications for the entire system are defined first, thereby being revised at each iteration, and at each iteration a subset of the system is built while those

his research; the first being the throw-away prototype and the second is referred to as a *real* life cycle.

The throw-away version of the Prototyping Life Cycle is commonly used in rapid application development (RAD) environments typically when the user interface has significant bearing on the project (Fioravanti, 2005). However, the throw-away process cannot really be considered a life cycle because there is only a single phase after, which is used for “analysis and specification of the real project” (Fioravanti, 2005 p. 77).

According to Fioravanti (2005) the Prototyping Life Cycle, consisting of five phases and presented in Figure 4, can advantageously be applied in the development of Web interfaces and similar applications, because the users steadily see the growth of the project and the evolution of it toward the final product. The Prototyping Life Cycle is composed of two main iterative processes. The first of these processes centers on the requirements (fast project, prototyping, and requirement review) and the second is among the requirement review and engineered product (Fioravanti, 2005).

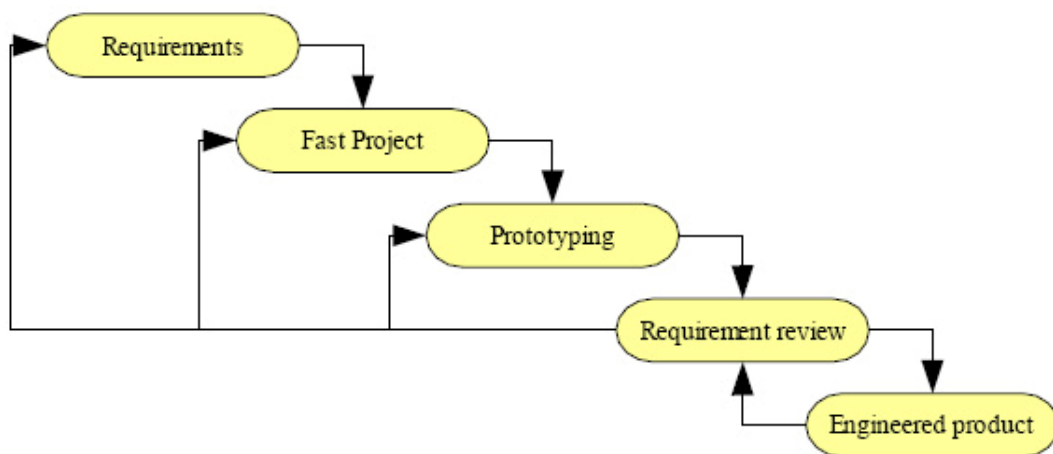


Figure 4 - Prototyping Methodology (Fioravanti, 2005 p. 78)

Craig Standing (2002) observes that while there are numerous products to document Web sites, as well as monitor Web site activity, they typically are not

incorporated into a collection of tools for the development process; and similarly the efficiency and effectiveness of Web sites cannot be determined the same way as traditional applications. Through their research, Howcroft & Carroll (2000) argue that while there have been numerous attempts at a Web development methodology, they are nothing more than *best practices* as to how a Web site should look and feel. A review of the literature reveals that some proposed methodologies for developing Web-based applications can prove to be useful models to abide by during development.

One such methodology is the Intranet Design methodology which is focused on the development of an intranet application (Lee cited in Standing, 2002). This methodology consists of multiple stages which, similar to other methodologies, begin with a feasibility study. The following four stages are focused on design issues while the final two stages cover implementation and testing (Standing, 2002). Standing (2002) points out that the primary features of this methodology are the meta-information structure, a method to organize information in abstract nodes with their links, and the information structure, which defines the local structure of these nodes using a variety of attributes.

Building on traditional application methodologies Howcroft & Carroll (2000) developed their own methodology for developing Web-based applications. Their model contains four phases which can be further divided into steps as seen in Table 1.

Table 1 - Howcroft & Carroll Methodology (Howcroft & Carroll, 2000)

Phase	Description	Step	Description
Analysis	This phase focuses around the development of a Web strategy as well as the analysis of how this strategy may be achieved.	<i>Development of a Web Strategy</i>	Creation of a Strategic Planning Document which identifies the following (Robson cited in Howcroft & Carroll, 2000): <ul style="list-style-type: none"> • Where the organization wishes to be • Where the organization is currently • Implementation plan of how to get from the current position to the desired one
		<i>Defining Objectives</i>	Creation of the Objectives Document which outlines the objectives of the site.

		<i>Objectives Analysis</i>	Analysis of the Objectives Document focusing on analysis of technology, information, skills, users, cost and risk. A new Objectives Document is then created post-analysis.
Design	Iterative phase that focuses on the design of the actual Web site with emphasis on the fact that the site is likely to change due to new technologies and requirements.	<i>Design</i>	Development of a Design Document focusing on the information design and the graphical design. Information design can include database design, functions, stored procedures and other scripts. Graphical design is the "look and fell" of the site.
		<i>Design Testing</i>	This step focuses on testing the design of the system to see if it will satisfy the goals set forth in the Analysis phase.
Generation	Phase which focuses on the generation of the Web site concept from the previous phases.	<i>Resource Selection</i>	This step deals with the resources necessary for the site to operate. Resources can include hardware, software and personnel.
		<i>Design Review</i>	During this step the Design Document from the Design phase is evaluated to the resources from the Resource Selection step. The Design Phase and Resource Selection will need to be revisited if there are compatibility issues between the resources and the design.
		<i>Code Generation and Installation</i>	Installation of software, database connections and posting the site onto the Web server may all be performed during this step.
		<i>Testing</i>	The testing step could be one of the more difficult steps. The site will need to be tested on multiple platforms and browser to ensure compatibility of code.
Implementation	The most simplistic but most important of the four phases which is ongoing throughout the lifecycle of the web site.	<i>Implementation</i>	Implementation focuses on notifying the pertinent users that the site is now available.
		<i>Maintenance</i>	Throughout the life of the Web site it will need to be maintained so that information and links on the site are current and updated in a timely manner.
		<i>Objectives Review</i>	This step is an on going process for Web developers to review new technologies and to assess them with respect to the requirements of the Analysis phase.

In this methodology the steps involved in the Design phase are iterative and will repeat until a refined Design Document is achieved. Also in this methodology, there is iteration between the Analysis phase and the Design phase to put aside any objectives which may be time consuming or expensive to develop (Howcroft & Carroll, 2000).

While the Spiral, Prototyping, and Waterfall methodologies are acceptable for use in creating an application not developed for the Web with a few modifications many of the phases within them can be applied to Web-based application development. The Intranet Design Methodology and the Howcroft & Carroll methodology were created

with Web-based applications in mind and could have been used in the development of this project, however, these methodologies are not as well known as standard development methodologies and therefore may not be as proven.

Web Site Design:

Chen & Yen (2004) believe that imperative to the success of a Web site, and to deliver the best user experience, are proper design and several essential features. Of critical importance to a Web site is how users will navigate through the multitude of pages linked on that site. Yu, Roh & Han (2001) performed a study on three different types of menu systems as seen in Figure 5.



Figure 5 - Three Menu Styles Studied; simple selection menu (Left), global and local navigation aid menu (Middle), and pull-down menu (Right) (Yu et al., 2001).

The results of the study showed that the pull-down menu design was more beneficial for users than the other two style while performing search related tasks. Yu et al. (2001) attribute this to the fact that the users are able to navigate to lower levels of the site without first navigating to pages that contain the links to the pages below them. While performing browsing tasks the results showed that users preferred the global and local navigation style (Yu et al., 2001).

Similar to Yu et al. (2001), Huizingh (2000) sees the importance of navigation in Web sites. His study of 651 Web sites found four different navigation structures; a tree, a tree with a home page button, a tree with some horizontal links and finally an extensive network. The tree structure is one in which a page is linked to one or more pages at the next level and to only one page at the previous level (Huizingh, 2000). Results showed that 60% of the Web sites studied had a simple tree structure or a tree supplemented with a *back to home page* button. The results also indicated that the navigational structure is dependent on the industry for which the page was designed (Huizingh, 2000).

Security:

Soh & Young (1998) note that the open design of the Internet has led to a substantial lax in system security, which can lead to the possibility of aggressive attacks on local area networks (LANS) and wide area networks (WANS). Chen & Heath (cited in Suh, 2005) agree stating that due to the open operating environment of the Internet, security is a major concern. Westman (2002) believes that developers should address security holes at three levels; the system level, the database level and the browser/server communication level. Web site developers can use a variety of methods to accommodate for the lapse in security on the Internet at these different levels. At the browser/server communication level a developer may implement Secure Sockets Layer (SSL) which provides for the encryption of data, the use of digital certificates for authentication and the assurance that messages are not altered through message integrity (Soh & Young, 1998). A number of system level security deficiencies can be resolved by network administrators through the use of firewalls and/or proxies (Soh & Young, 1998) and by

making sure their servers, especially those exposed to the Internet, are properly patched (Westman, 2002; Alapati, 2003, sec. 1, par. 27).

Given that many businesses use databases for the storage of critical information, and new requirements of both laws and governments require increased security of this data, it is important to take the necessary actions to ensure security (Fernandez-Medina & Piattini, 2005). Beynon-Davies (2004) suggests that the primary way of securing database systems is by defining a set of authorized users of the entire system, or more commonly, parts of the database. In order to prevent hacking attempts on default user accounts of the RDBMS, Oracle Corporation recommends that all default user accounts are locked and expired, with the exception of the SYS and SYSTEM account (Alapati, 2003, sec. 1, par. 3). Alapati (2003, sec. 1, pars. 4, 5, 7, 27) also suggests the following to secure a database:

- Do not hard-code user passwords in scripts, but rather use a password file.
- Force users to change passwords in a timely fashion.
- Perform frequent checks of the audit trail for signs of logins as SYSDBA and unsuccessful attempts to log in to the database.
- Keep current on new security vulnerabilities and patch accordingly.

While securing the perimeter of a network with a firewall will protect the database from external threats, using these suggestions is a good step in protecting the database from risks from the network inside the firewall (Alapati, 2003, sec. 1, par. 1, 22).

Conclusion:

Developing a Web-based application can be of significant value for an organization. The Internet has had a profound impact on businesses where Web technology usage has increased (Standing, 2002; Suh 2005). Through the use of server-side scripting languages, such as ColdFusion, ASP.NET or PHP, it is now easier for organizations to use the Internet to host Web-based applications. Organizations often use some sort of RDBMS system to store critical data and Web technologies allow them Web-based access to their data (Brooks-Bilson, 2003).

Many of the server-side scripting languages provide for rapid development of Web-based applications. Since the Web is a new medium for application development, there are few software development methodologies dedicated to developing Web-based applications. Howcroft & Carroll (2000) presented a four phase methodology designed for Web applications. Traditional methodologies, for example the Waterfall methodology, the Spiral methodology and the Prototyping methodology, may be adapted for Web-based applications.

Critical to the success and acceptance of a Web site and application is the design, especially in the area of navigation and menu design (Chen & Yen, 2004; Huizingh, 2000; Yu et al., 2001).

Due to the open design of the Internet there exists a lax in security (Soh & Young, 1998) so measures need to be taken to insure that the application and its database (data) are secure against risks both internal and external to the organization.

Research methods to be used:

The research methods that were used for this project included the use of acceptable journals to gain an understanding of what makes a Web-based application successful. The research was also used to understand what was needed for a CRM system to be successful while also taking into account the cost of ownership of the system. The research provided the basis for developing a working Web-based CRM system that was coded using ColdFusion as the server-side language, and that stored and retrieved data relevant to the system in an Oracle 9i backend database. From the information gathered an informed decision about the technology to be used was made. The necessary software was installed on development servers and tuned according to best practices. After the CRM project was completed and put into the production environment, an analysis of the application and its related data was performed to determine if the use of the CRM system had a positive effect on sales and if the system adequately allowed the sales staff to track their potential customers.

Literature and research that is relevant to the project:

There is not a clearly defined methodology for developing Web-based applications, and for that reason the book *Web Engineering: Principles and Techniques* by Woojung Suh provided an abundance of information about the process of creating Web-based applications, focusing on how the traditional development methodologies can be applied. The methodology proposed by Howcroft & Carroll (2000) provided a vast amount of information; because their methodology was developed strictly for Web-based

applications, they covered the necessary phases, as well as steps within those phases, to develop a system that is well thought out by those using and requesting it.

One of the more helpful resources was the book *Programming ColdFusion MX* by Rob Brooks-Bilson. This book contains all of the functions and tags built into the ColdFusion language and it shows how to use them with examples. The book also covers a variety of other useful topics for building a ColdFusion application, such as creating user-defined functions, custom tags (using C++ and Java), maintaining session state, security, and database connectivity topics.

Research performed by Yu et al. (2001) on how different menu styles effect the usability of a Web page was insightful, as often times users do not know how they want a Web page to look, they just know what they want it to do. Using the results of this study one can make an assumption about they type of menu system that would be the most beneficial in a Web-based application.

The study presented in the article by Roh et al. (2005) is valuable because it provides insight into how to measure the success of a CRM system. Success in the design and implementation of a CRM system could potentially result in an increase in sales, allow for better informed marketing decisions (Chen et al., 2005), and an increase in customer loyalty (Roh et al., 2005), which is essential for this project.

Alapati (2003) provided practical ideas into how to properly secure a database. Many organizations use a database in some way or another and while he specifically targets an Oracle RDBMS, the concepts can be applied to other systems as well. New laws are making it a requirement for certain organizations to abide by strict guidelines

concerning the security of their data and their network as a whole, which gives value to the works by Fernandez-Medina & Piattini (2005), and Westman (2002).

Summary of what is known and unknown about the topic:

As a result of the research performed it was determined that there is an abundance of literature available both in print and online on the topic of server-side scripting languages. ColdFusion and ASP.NET are developed by large companies who produce their own detailed documentation, while PHP, an open source application, relies on the development community to update and contribute to its documentation.

Some of the literature went into depth about CRM tools, as well as using marketing to retain customers, however, relevant research could not be found as to whether or not a custom built Web-based CRM solution provided a benefit, either in cost or customer retention, over a product that is used *off the shelf*.

A search for journal articles on the topic of software development methodologies revealed that there is a great deal of existing literature which goes into depth about traditional development methodologies, but is limited on the topic of methodologies created specifically for the development of a Web-based application.

The contribution this project will make to the field :

Internet use and its availability have grown throughout the years, therefore, it provides another medium in which businesses and organizations can use to reach its customers and also provides businesses and organizations a way to extend the workplace. This project demonstrated how the Internet, along with Web technologies, was used as a

host for applications of which users of the system did not have to be located in a corporate office or be part of a WAN. The project focused on the steps involved in designing the backend database, securing the data, and securing the application. It also showed how to build an application for the Web when there is not a traditional methodology to follow which meets the requirements set out by those who had the initial concept for the system.

Chapter 3 – Methodology

Life Cycle models followed :

This project consisted of developing a Web-based application that was used as a CRM tool. Since an application was designed, coded, and implemented it was fitting to use the Software Development Life Cycle (SDLC) and a somewhat modified ancillary model, the Prototyping Life Cycle which was discussed earlier in this paper and depicted in Figure 4. According to Fioravanti (2005), there are two Prototyping Life Cycles that one can use in application development. The first is called the throw-away prototype and is usually used in a rapid application development (RAD) project/environment and the second is the *real* life cycle in which follows more methodical development (Fioravanti, 2005). While this project needed to be developed rather rapidly the throw-away prototype just did not seem to be as suitable for the project as the true Prototyping Life Cycle. The phases of this life cycle allowed for rapid development, but also, for the flexibility of going back to previous phases and building on itself; similar to the Spiral Life Cycle.

1. *Requirements*: The application was built from the requirements given to this author by the *owner* of the project at ARC. The requirements gathering phase consisted of meeting with management personnel at ARC to gain a thorough understanding of the goals they wanted to accomplish with the application. These requirements were then used throughout the project and were revisited during development of the individual project *modules*. Once all of the requirements were met the application was deemed complete.

2. *Fast Project*: The first time this phase was encountered the primary focus was for ARC to define in detail how the user interface should look and feel to the users of the application. From these recommendations, this author proposed an interface and menu style that suited ARC's needs. Also, this interface implemented the requirements specified in the first requirements gathering (Fioravanti 2005).
3. *Prototyping*: In the prototyping phase actual coding began. The database was designed, constraints were put in place on table columns and the coding of certain stored procedures and functions applicable to the first requirements phase were developed. Fioravanti (2005) states that in this phase the coding approach resembles that of RAD in which visual tools improve development speed and also allow the generation of pseudo-code. For this application the use of Macromedia HomeSite+ and Quest Software's Toad for Oracle was a fundamental aide in its development.
4. *Requirements Review*: The requirements review phase was visited multiple times throughout the project. During this phase ARC looked at the prototype that was developed based on their initial requirements, and subsequently approved or denied particular *modules* that were completed. Often times during this phase new requirements were added, which, eventually, when the necessary changes were made, moved the application from a prototype to a finished product.
5. *Engineered Product*: The engineered product was the culmination of all the revisions to the prototype and the end result of the Prototyping Life Cycle. The engineered product had been previously tested for any bugs and to ensure that all required functionality was implemented. For this project the engineered product

was the custom built application that the sales staff of ARC were able to use to track their customers.

Specific procedures:*Application Development:*

There were numerous specific procedures that were used in developing this application. The procedure for application development at ARC is probably similar to that of other companies. A project is first presented to the Vice President (VP) of Information Technology (IT) who then will assign that project to a member of their staff; typically the project will not have a requirements document with it. The person who is assigned to the project will then meet with the owner of the project at ARC to make sure they understand it, and to gather the requirements. After the project is developed it is moved to a test environment where user testing is performed. Users will report any bugs that were found back to the person who was developing the application so they can fix the bugs in the development environment and once fixed move those fixes to the test environment. When all testing is completed the application is moved to the production environment.

In order to move an application to the production environment at ARC a *Move to Production* form will be filled out by the developer. This document will be comprised of the exact steps to put the application into production. In the *Move to Production* form, the developer is required to indicate any changes that will need to be made to the ColdFusion environment, including data sources, directory mappings, and any Java classes that might be used in the application. This document will also contain the

location of any SQL scripts that will need to be run in order to create the database objects. Finally, the document will contain a listing of the files or directories that will need to be copied to the production server, as well as the path to which they will need to be copied.

Typically, the procedure for reporting bugs in the test environment is that the users will generate a screen shot of the error and indicate the steps they took to produce the error so that the developer can duplicate the problem. Once the problem is identified the developer will make the necessary fixes in the development environment and then those changes will be implemented to the test environment for further testing. When an application is in the production environment the procedure for reporting bugs is slightly different. If a user finds a bug in the production environment they are required to complete a *Trouble Report* form. This form requires the users to indicate the steps to reproduce the error and to indicate if any recent changes may have resulted in this bug. Upon completing the form and giving it to the developer, the problem is fixed in the development environment, tested in the test environment and then moved to the production environment.

When an application is moved to the production environment often times there will need to be changes or improvements to it. In order to have a developer work on those changes the owner of the project is required to complete a *Change Request* form. The purpose of this form is so the developer knows what changes to make to the application. When the change is complete it gets treated as if it were a separate project, therefore, the developer would complete the *Move to Production* form in order to implement the change in the production environment.

Lead Assignment:

The procedure ARC envisioned for assigning leads to their sales staff can be seen in Figure 6. If a lead is unassigned (A), then the ARC Call Center staff would be allowed to assign that lead to any of the sales staff (B), available within the geographic location of the lead. If the lead is assigned to a Sales Agent, or a member of the Community Staff (C), and that person does not act on the lead, while there are currently no other Sales Agent or Community Staff available who could have taken the lead, then the lead will flow up the hierarchy to the Sales Manager (D). The Sales Manager can then assign that lead to one of their subordinates (2). If a Sales Manager is not available in that geographic area then the lead will flow to the Marketing Manager (3). The Marketing Manager in the area can assign the lead to any of their subordinates (5), however, if there is not a Marketing Manager in the area then the lead flows to the Regional Sales Vice President (VP) (6). The Regional Sales VP, similar to the other managers, can assign the lead to any of the staff below them in the hierarchy. At any point in the flow a manager can take the lead for themselves (1, 4, 7) instead of assigning it to one of their subordinates.

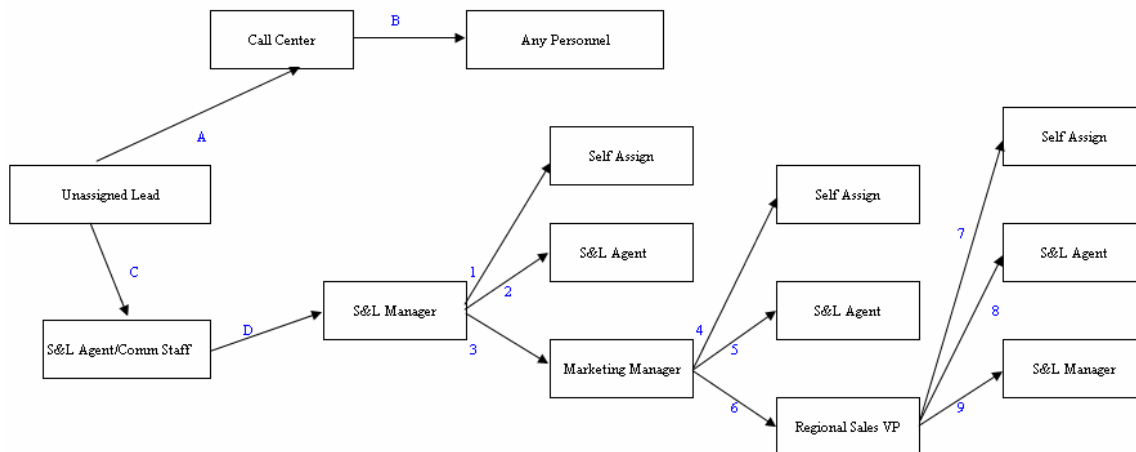


Figure 6 - Possible assignment of a potential sales lead (Darschewski, 2006).

Application Coding:

The coding of the Website was done using Macromedia HomeSite+ using CFML as the language, JavaScript to perform some client-side form validation and to generate the pull-down menu, and Cascading Style Sheets (CSS). It made sense during development to use a separate ColdFusion page (CFM) for every link in the application, rather than combining multiple pages into one page.

Formats for presenting results/deliverables:

The results and deliverables of this project are presented through a series of screenshots of the application modules and the modules' sections. In addition, an analysis of ARC sales reports was performed to see if the implementation of the CRM application had any effect on company sales and overall occupancy. The findings of these reports are represented in graphical and chart form so that they are more meaningful for the readers.

Review of deliverables:

The deliverables are presented in a logical order based on how the application was generally used by the sales staff. The following areas of the application will be covered.

Security:

- Login
- Forgot password section
- Profile/User creation section
- Edit agents profile
- Edit your profile
- Delete profile
- Change password

Home Page:

- Welcome section
- User dashboard

Leads:

- Enter lead
- Show leads
- Create sales documents
- Enter out of market lead

Search:

- Search leads
- Search lead documents

Marketing:

- Add marketing source
- Edit marketing source

Reporting, etc.:

- Reports
- Send Message

Also presented will be an analysis of sales and occupancy reports for the first nine months the application was fully used by the ARC sales staff, as well as the same nine months one year prior to implementation.

Resource requirements:

There are three environments for an application at ARC; development, test, and production. A single server is used as the development Web server and an additional server houses both the development databases and test databases. ARC's test environment consists of a single Web server, however, the test environment uses the same server as the development environment for its databases. The production environment consists of a Web server and a separate database server. The servers used in all environments are nearly identical to each other, in terms of hardware and software installed, to avoid any conflicts that could occur if they were different. ARC owns all of the servers that were used in the development of the application. Microsoft Windows 2000 Service Pack 4 is the operating system installed on the Web servers used in the project, while the operating system on the database servers is Red Hat Linux Advanced Server Version 2.1; all operating system software licenses are the property of ARC.

When the project began ColdFusion 5 Standard Edition was the version that ARC was running for which they had purchased the proper licenses. As the project progressed it was decided by the Vice President of Information Technology to migrate to ColdFusion MX. The reason for the migration was for better support of other applications ARC already had running in their production environment.

The database used for the project was Oracle 9i. ARC was already running this version of Oracle for their data warehouse and is the owner of the software licenses.

A company owned laptop along with licensing of Quest Software's Toad and Macromedia's HomeSite+ were also used in the coding of the application and the development of the database tables, functions, procedures and other database objects.

Outcomes:

A report of ARC's occupancy numbers and their net resident move-ins were analyzed for the nine months beginning January 2005, as well as the same period one year earlier. The CRM application was put into the test environment at ARC in early October 2004 so that the entire system could be tested as a *whole* application. After fixing some bugs found in the test environment, by early December, the application was moved to the production environment allowing the sales staff around the country to be trained on how to properly use the application. By January 2005, the application was believed to be used by all sales staff to track their potential customers. The charts depicting the net move-ins, the percentage of occupied home sites, and the total home sales for the first nine months of 2005, and the same nine months one year earlier, can be seen later in this paper.

Aside from the impact of using the CRM application, a major outcome was the application itself. ARC finally had a reliable, fast, easy to use application for their sales staff to track potential leads. The application had the potential to increase sales, as well as the potential to increase the efficiency of ARC's sales staff; allowing them more time to focus on customer service.

Summary:

The Prototyping Life Cycle, based on the SDLC, was the best candidate of all the methodologies considered to develop the Web-based CRM application for ARC. During its development, this project progressed through the five stages of this life cycle; requirements, fast project, prototyping, requirements review, and engineered product.

The project was developed using the specific procedures ARC used for developing other applications. Initially, the developer is to meet with the owner of the project at ARC to gather the requirements for the project. Once the requirements are known development can begin. When the application, or parts of the application, is ready for testing it is moved from the development environment to the test environment. Any bugs users may find will be reported to the developer so they can be resolved. When the users believe all bugs have been resolved then the application is moved to the production environment using the *Move to Production* form to outline the steps involved in the move.

ARC already owned the proper licenses for Oracle and the operating systems that would be necessary for the application. They also had all the necessary hardware requirements in place for the application to run. ARC purchased the proper ColdFusion

MX licenses, as well as the other software required for the development of the system, and all hardware that was required for the application to run.

The major outcome of the project was the final application, however, equally impressive are the results of the data analysis, discussed later in this paper, that were produced by having the sales staff at ARC use the application for tracking potential customers.

Chapter 4 – Project History

How the project began:

The project was presented to this author by way of management at ARC.

Following the phases of the Prototyping Life Cycle it was necessary to meet with ARC management, herein known as the *owner* of the project, on a few occasions to gather the requirements of the project. Upon the initial meeting with the owner of the project it was clear that the application needed to adhere to the following requirements:

- The application must be accessible to the sales staff of ARC via the Internet and not by means of third party applications such as Citrix Metaframe or Microsoft Windows Terminal Server.
- The application must contain its own user database and not rely on the network security (Microsoft's Active Directory) of ARC.
- The main purpose of the application is to gather information about the housing needs of potential customers of ARC. The pertinent customer information ARC would like to collect can be seen in Figure 7. This figure is the data collection screen as envisioned by ARC.

Navigation Bar

HOME
LEADS
ENTER CUST.
REPORTS
SEARCH

Community Location: Housing needed within:
 Lead Origination: Date Entered:
 First Name:*
 Last Name:*
 Address:*
 City:* State:* Zip Code:*
 Phone Number (Home):* Phone Number (Work):
 E-Mail Address:
 Best time to contact: Language Preference:
 Lead follow up/ Closure

Interested In:
 Lease to Own: ☐ Finance: ☐ Cash Sale: ☒
 Lease: ☐ Lot Rent: ☐

Action:

Customer Notes
 10/1/2004 4:51 PM
 Customer Data Entered
 A note that the "Customer Data Entered" pops up automatically when one of the "Action" buttons are clicked

Add Customer Notes

Figure 7 - Required Lead Information (Darschewski, 2006).

- A hierarchy needs to be established that defines the *roles* of the users of the application. The hierarchy will be used to determine who has authority over which users in terms of assigning and viewing sales leads.
- Marketing staff will need to be able to add and edit marketing sources (i.e. specific newspapers, advertisements, etc.) to the application through some kind of interface.
- *Roles* that are superior in the organization chart (Figure 8) should be allowed to create and edit users that are their subordinates and certain *roles* will be allowed to create any kind of user.

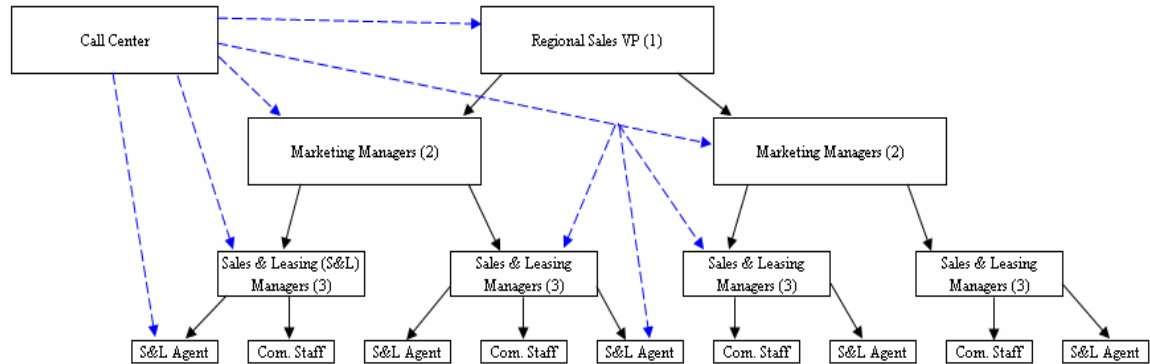


Figure 8 - Lead Management Organization Chart. Blue dashed lines indicate who leads can be assigned to, while solid lines indicate assignment of leads by superiors (Darschewski, 2006).

- Once a sales lead is entered the sales person should have a method of creating an appointment for the lead and have that appointment information readily available.
- If the person who entered the lead has staff that is subordinate to them on the organization chart then they will have the option to automatically have the lead assigned to a member of their staff, or an option to manually assign the lead to a member of their staff.
- If a sales lead is not accessed within a 24 hour period then that lead becomes stale and is automatically reassigned to a new sales agent within the same geographic location.
- Sales staff will need to be notified via e-mail when a new lead is assigned to them.

The preceding requirements were the foundation for the business logic of the application. The next step of the project involves the fast project phase of the Prototyping Life Cycle.

After obtaining the requirements of the project it was necessary to determine which technologies to use to develop the application. Because ARC was currently using

ColdFusion for their internal intranet applications it was fitting that this project be developed using that server-side programming language. Although ARC was using ColdFusion alternative programming languages were researched to see if they would be more suitable for the development of the application. Microsoft's ASP.NET, as well as PHP, were potential candidates for developing this application. This author had some experience programming in all three of these languages, but using ColdFusion made the most sense for ARC for the simple fact that its IT staff would be able to better maintain and modify the application after the project has ended.

Once the development language was decided it was necessary to decide on a RDBMS system. As was the case with the development language, ARC was currently using Oracle 9i for other database applications so it made sense to use that system for this project. To accommodate the needs of the application a new schema was created where the tables, stored procedures, functions and other database objects required by the application were created.

The database administrator (DBA) at ARC insisted on using the same database that ARC uses for most of their other Web-based applications, as well as their data warehouse. In this database a new schema was created, referred to as *leadmanagement*, for the CRM application. When designing the database this author attempted to normalize the tables as much as possible with emphasis on storing related data in the same tables, while also concentrating on eliminating the possibility of redundant data, which is the suggested database design strategy.

The majority of the table structures were relatively simple. There were, however, some difficulties expressing the sales staff's organizational hierarchy and job roles in

table form, as well as the hierarchy of assigning leads. This issue was resolved by creating a table with multiple columns; the ID column would contain a unique ID, the NAME column would hold the logical name of the job role, while the remaining columns would hold the job role ID's which the named job role can create (CCx columns) accounts for, i.e. a Sales and Leasing Agent can only create accounts in the system for Sales and Leasing Agents and Community Managers, this is depicted in Figure 9.

I ▲	NAME	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10
1	CALL CENTER MANAGER	2	5	6	7						
2	CALL CENTER AGENT										
3	VP OF SALES	4	5	6	7						
4	MARKETING MANAGER	5	6	7							
5	SALES AND LEASING MANAGER	6	7								
6	SALES AND LEASING AGENT										
7	COMMUNITY MANAGER										
8	ADMIN	1	2	3	4	5	6	7	8		

Figure 9 - Job roles and job role creation hierarchy (Darschewski, 2006)

The method used for the hierarchy of assigning leads is similar to the one used in the JOB_ROLE_RIGHTS table; this is depicted in table form in Figure 10. The JRRID column is a foreign key to the JOB_ROLE_RIGHTS table's ID column. The ALx (where x is a number from 1 to 10) columns refer to the job roles that the JRRID column can assign sales leads to.

ID	JRRID	AL1	AL2	AL3	AL4	AL5	AL6	AL7	AL8	AL9	AL10
1	1	3	4	5	6	7					
2	2	3	4	5	6	7					
3	3	4	5	6	7						
4	4	5	6	7							
5	5	6	7								
8	8	3	4	5	6	7					

Figure 10 - Assignment of leads hierarchy represented in table form (Darschewski, 2006)

All of the tables were created with a database tool called TOAD by Quest Software. This tool is a Windows based application for creating tables, functions, stored procedures, and other database objects. After the tables were created a SQL script was generated that was used to deploy the tables to the test and production environments. The other tables and their relationships can be seen in Appendix A; the final entity relationship diagram (ERD) was the result of revisiting the requirements phase and adding any new requirements ARC may have added to increase the usability and functionality of the application.

The owner of the project at ARC provided Microsoft Excel spreadsheets which included crude depictions of how they believed the application should look and feel to the users. Additionally the file contained details as to how the application should *flow* or proceed through the process of tracking a lead from first contact until the lead is considered dead or closed.

The design of the actual Web page user interface was somewhat based on the crude drawings provided as well as the Internet site of ARC. As Chen and Yen (2004) and Yu et al. (2001) pointed out, the success of a Web site is based on proper design, which includes navigation, and of which menu style is extremely important. For this application it was decided to use the pull-down menu approach for navigation between the *modules* (Home, Leads, Search, Profile, Marketing and Reporting, etc.) of the application, seen in Figure 11. The menu system is restricted by job role so that not every salesperson is allowed access to all items in the modules.

Home	Leads	Search	Profile	Marketing	Reporting, etc.
	Enter Lead	Search Leads	Create Profile	Add Marketing Source	Reports
	Show Leads	Search Lead Documents	Edit Your Profile	Edit Marketing Source	Send Message
	Enter Out Of Market Lead		Edit Agents Profile		
			Delete Profile		
			Change Password		

Figure 11 - Pull-down menu system with admin job role (Darschewski, 2006)

As the Prototyping Life Cycle progressed the requirements review phase was encountered multiple times, particularly after each module was coded. This was done so that the owner of the project at ARC could comment on the prototype in regard to the functionality of the application and so they could determine if anything was missing. Often times new requirements were added to the project and the prototype had to be revised.

One added requirement of interest was that the user needed a way to reset his password if it was forgotten. The solution to this requirement was that the system would generate a new password for the user, send him an e-mail which had a link in it to activate the password, and after the password was activated the user would be allowed to log into the system to change his assigned password. Another requirement was that the application had to allow for the population of Portable Document Format (PDF) documents with specific information about the lead for which the document was being created. These documents were typically legal documents for the sale, leasing and financing of homes. The prototype was moved to the test environment during this phase for testing by ARC employees and they noted any bugs with the system so those problems could be resolved.

After numerous revisions the owner of the project at ARC approved the application and they had a final engineered product to be moved to the production environment for real world use. If any bugs were found with the system at this stage the author had to fix the bugs in the development environment, then the fixes were moved to the test environment for testing by certain individuals, in particular those who found the bugs, and finally the fixes were moved to the production environment.

To implement the system to the test and production environments new schemas needed to be created by the DBA in those environments. In addition, the Web pages that were developed for the system were copied to a designated folder in the Web root directory on these servers. In order to make the application separate from other Internet applications at ARC, a sub domain name of ARC's parent domain was created, as well as a virtual directory in Internet Information Server that pointed to the location of the application files. To create the database objects on the test and production servers multiple structured query language (SQL) scripts were created; one for the tables, one for the functions and procedures, one for the other database objects, including sequences, triggers, views and custom data types, and finally one for the static data of the database. The scripts were then executed in these environments by the DBA. Equally vital to the implementation of the final product was the creation of a data source connection. The datasource was created in the ColdFusion Administrator to allow ColdFusion to access the application's database.

The following section will provide details of the application. This will be done through the use of screen shots of the individual modules.

The Application:

When a user first enters the universal resource locator (URL) of the application into their browser they are presented with the Welcome screen in Figure 12, instructing them to sign in.



Figure 12 - Welcome screen of the application (Darschewski, 2006).

Security/Profile - The security portion of the application is composed of multiple screens and also consisted of a few stored procedures. The initial step in creating a user for the application is for an administrator to log in as seen in Figure 13.

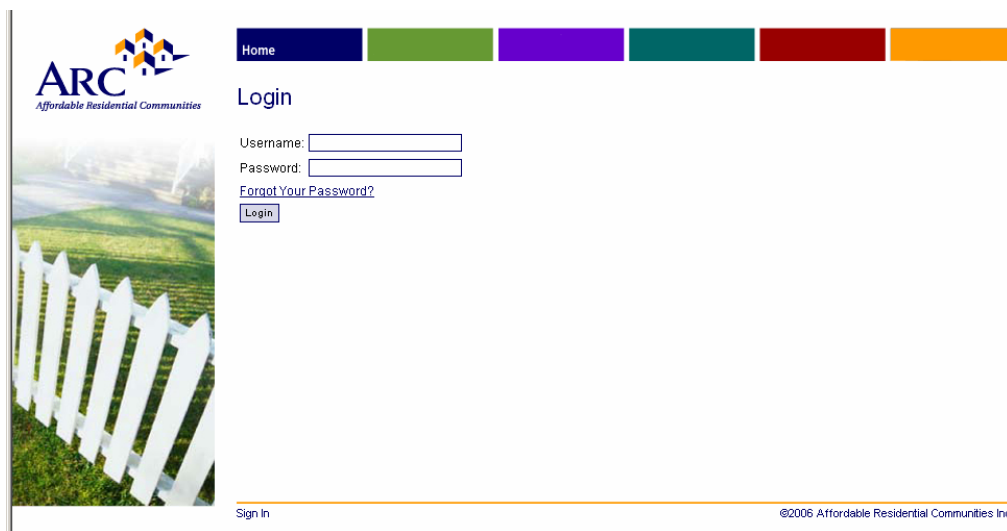


Figure 13 - Login Page (Darschewski, 2006).

If a user forgets their password they can click on the *Forgot Your Password?* link and are taken to a page where they can enter their username and the e-mail address that was input when their profile was created. This is illustrated in Figure 14.

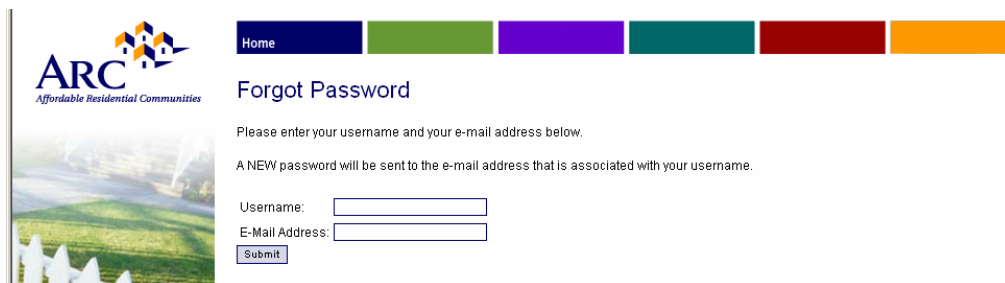


Figure 14 - Forgot Password Page (Darschewski, 2006).

Upon submitting the form the user is presented with the page seen in Figure 15. In addition they will receive an e-mail, Figure 16, which informs them on how to activate a temporary password to gain access to the application.

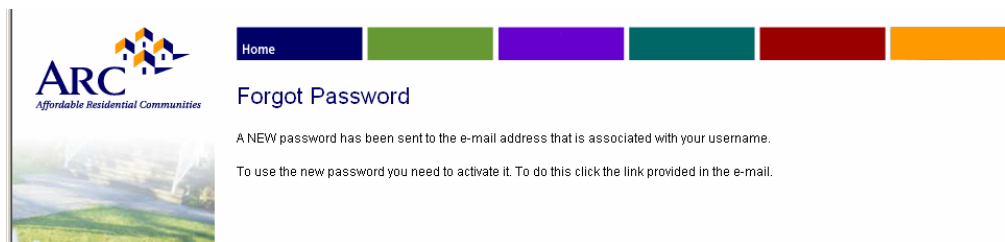


Figure 15 - Forgot Password Page (Darschewski, 2006).

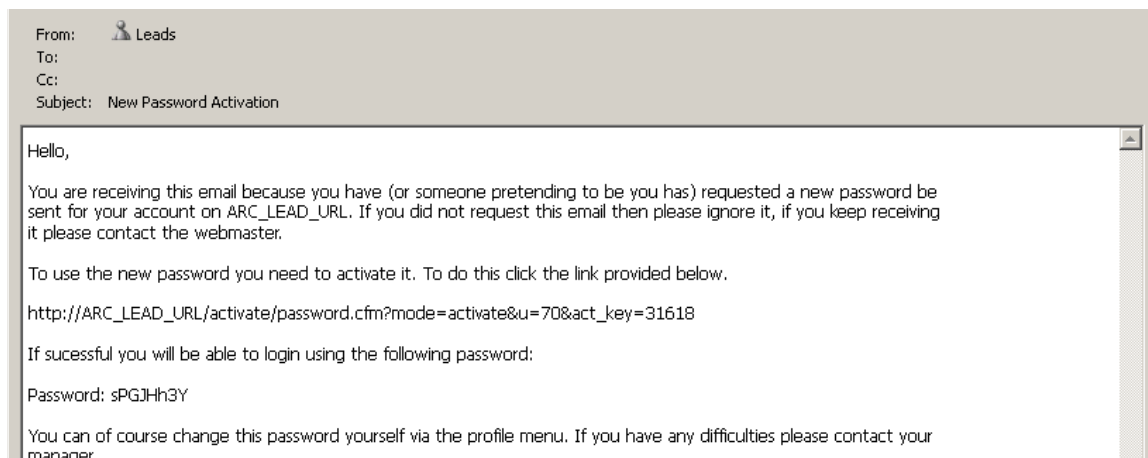


Figure 16 - Activate Password E-mail (Darschewski, 2006).

After successfully logging into the application the user sees an expanded menu system along the top of the screen and is presented with the Message and Appointment page, Figure 17. Depending on the users *role* in the system the menu items will be limited.

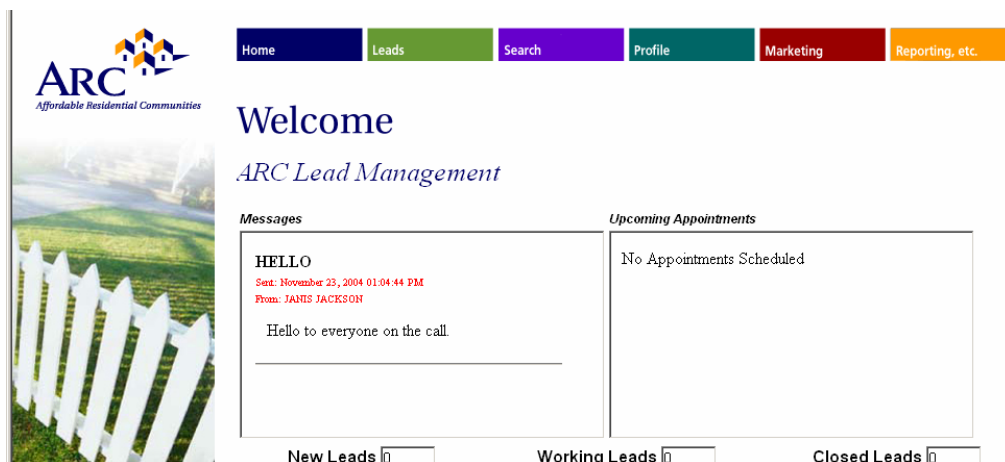


Figure 17 - Initial Login Page (Darschewski, 2006).

The Create Profile section, under the Profile menu item, is where the administrator can create other users of the application. This page can be seen in Figure 18. The user is required to enter the information that is identified by a red asterisk. It is important that the correct job role and location access is given so that a user is not inadvertently given more, or less, access than their job requires. When the form is submitted the user information is stored in the database. In this application user passwords will not be stored as clear text in the database, but rather in an encrypted form.

ARC
Affordable Residential Communities

Home Leads Search Profile Marketing Reporting, etc.

Add Profile

Profile Information

Job Role: *

First Name: *

Last Name: *

Address1: *

Address2:

City: * State: * Zipcode: *

Phone # (home): *

E-Mail: * Bilingual: *

Login Information

Username: *

Password: *
(More than 6 characters)

Confirm Password: *

Access Rights

Allowed Leads? *

Location Access: * Choices

Montgomery, AL
Mobile, AL
Fayetteville/Springdale, AR
Inland Empire, CA
Front Range, CO

Allowed Access

Birmingham, AL
Huntsville, AL

Submit Profile Clear Profile

Figure 18 - Create Profile Page (Darschewski, 2006).

After a user is created the application allows for certain job roles to edit the profile information of their subordinates. If the user's job role allows, this is done by choosing the Edit Agents Profile item from the Profile menu. Figure 19 shows that the user is required to pick one of their subordinates from the drop down list and after the person is picked, their information can be edited as seen in Figure 20. Once the agents information has been edited and submitted, those changes are immediately updated in the database.

ARC
Affordable Residential Communities

Home Leads Search Profile Marketing Reporting, etc.

Edit Agent Profile

Please choose an agent to edit:

--- Select ---

AMY WHITT
MARSHALL WIGGINS
PAMELA WIGGINS
JAY WILLIAMS
TAMMY WILSON
TONI WIMMER
TABETHA WINCHELL
ROSE WISMEG
GREG WRIGHT
MADDIE WRIGHT
TONY ZUCHINSKI

Figure 19 - Edit Agent Profile Selection Page (Darschewski, 2006).

ARC
Affordable Residential Communities

Home Leads Search Profile Marketing Reporting, etc.

Edit Agent Profile

Profile Information

Job Role:* ADMIN

First Name:* JAMES

Last Name:* DOE

Address1:* 800 GRANT ST

Address2: SUITE 900

City:* DENVER State:* CO Zipcode:* 80203

Phone # (home):* 3031231234

E-Mail:* jamesd@arcemail.com Bilingual:* Yes

Access Rights

Allowed Leads?* No

Location Access:* Select Choices

Allowed Access

- West Region
- South Central Region
- East Region
- North Central Region

Update Profile Cancel

Sign Out ©2006 Affordable Residential Communities Inc.

Figure 20 - Edit Agent Profile Data Page (Darschewski, 2006).

Additionally, the Profile menu allows users to edit certain fields of their own profile. This is available to them by clicking on the Edit Your Profile item. The fields that the user is able to edit can be seen in Figure 21.

ARC
Affordable Residential Communities

Home Leads Search Profile Marketing Reporting, etc.

Edit Your Profile

Profile Information

First Name:* JAMES

Last Name:* DOE

Address1:* 800 GRANT ST

Address2: SUITE 900

City:* DENVER State:* CO Zipcode:* 80203

Phone # (home):* 3031231234

E-Mail:* JAMESD@ARCEMAIL.COM Bilingual:* Yes

Update Profile

Sign Out ©2006 Affordable Residential Communities Inc.

Figure 21 - Edit Your Profile Page (Darschewski, 2006).

Throughout the life of the application users will need to be removed from the system, therefore, the Delete Profile item allows a person with the proper rights to remove these users. The person deleting the profile first needs to select the profile they

want to delete. When the profile is deleted from the system the person who is initiating the deletion has the option to take the leads of the deleted user. This option is only available if the person performing the deletion has the ability to have leads assigned to them. If they do not have the ability to have leads assigned to them then the leads are automatically reassigned to other agents in that geographic location; this is seen in Figure 22. The deleted user is not really deleted from the system, rather a flag is set in the database indicating that the user is disabled and their password is set to something different.

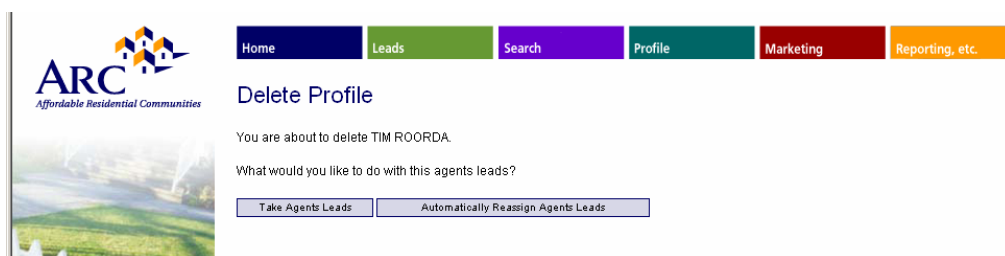


Figure 22 - Delete Profile Page (Darschewski, 2006).

The final item in the Profile menu is the ability for users to change their password. The application does not require password changes, but does provide the ability for users to maintain their own passwords. This feature can be seen in Figure 23.

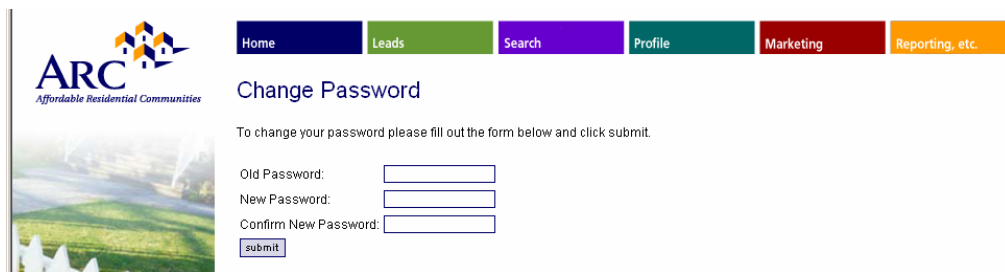


Figure 23 - Change Password Page (Darschewski, 2006).

Leads – The leads section of the application is where a logged in user enters lead information or updates the information of one of their leads already in the system. To enter a new lead the user clicks the Enter Lead item in the Lead menu and is presented with the form seen in Figure 24.

The screenshot shows the 'Enter Lead' form in the ARC Affordable Residential Communities application. The form is divided into several sections:

- Navigation Bar:** Home, Leads, Search, Profile, Marketing, Reporting, etc.
- Lead Information:**
 - Community Location:
 - Market:
 - Lead Origination:
 - Date:
 - Housing Need:
- Personal Information:**
 - First Name:
 - Last Name:
 - Address:
 - City:
 - State:
 - Zipcode:
 - Phone Number (Home):
 - Phone Number (Work):
 - E-Mail Address:
 - Best time to contact:
 - Language Preference:
- Interested In:**
 - Bedrooms:
 - Bathrooms:
- Lease Options:**
 - Lease To Own: ☐
 - Finance: ☐
 - Cash Sale: ☐
 - Lease: ☐
 - Lot Rent: ☐
- Buttons:** Auto Assign Lead, Manually Assign Lead, Clear Form

Figure 24 - Enter Lead Page (Darschewski, 2006).

The user is only allowed to see the locations that they are assigned to in the community location drop down, as well as the markets for the locations they are assigned to in the market drop down. When entering a lead, certain fields are required; these fields are indicated by a red asterisk. Depending on the *role* of the person logged into the application the user can auto assign the lead or manually assign the lead. If the user is able to accept leads then they are able to assign the lead to themselves.

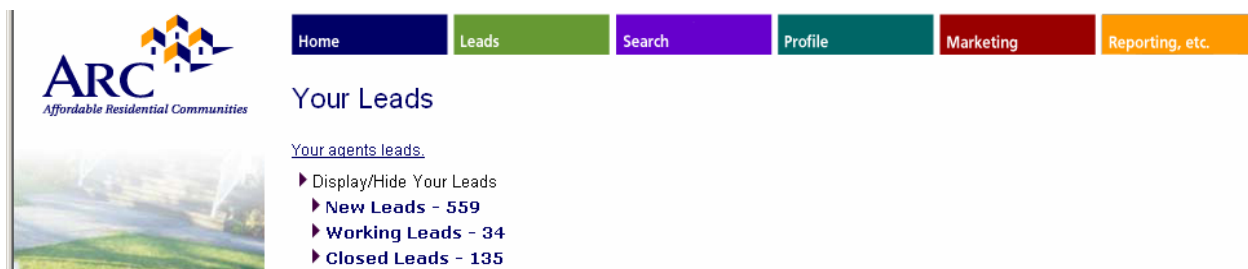


Figure 25 - Show Leads Page (Darschewski, 2006).

By selecting the Show Leads item the user can see all the leads currently assigned to them, seen in Figure 25. Figure 25 also shows that the particular person logged in is a manager and is allowed to see the leads of their subordinates. Seeing the leads of a subordinate is accomplished by clicking the *Your Agents Leads* link. The leads are grouped into three type; New Leads, Working Leads, and Closed Leads. To see the leads in each of these groups the user can click the triangle to the left of the respective heading. Figure 26 shows the expanded leads of a subordinate agent.

Leads of MARIA ACOSTA

► Display/Hide Leads

▼ **Working Leads - 8**
Click a lead to retrieve the record.

Appt.	Name	Location	Interest	Last Contact	Housing Need
	TOMAS SANCHEZ	Friendly Village, GA	Lease,Lease To Own,Finance	2005-03-16 16:12:32.0	Immediate
	WELLENTON ISURIETA	Friendly Village, GA	Lease To Own,Finance	2005-03-19 17:17:39.0	Immediate
	JUANA ORTIZ	Friendly Village, GA	Finance,Lease To Own	2005-03-19 17:22:37.0	Immediate
	JUAN MIGUEL DEL VALLE	Friendly Village, GA	Finance	2005-03-19 17:19:16.0	Immediate
	MANUEL TUNJAMA	Friendly Village, GA	Lease To Own	2005-03-19 17:24:11.0	Within 1 Month
	ALJANDRO EDITH VILLA	Friendly Village, GA	Lease To Own,Finance	2005-03-19 17:26:31.0	Immediate
	MICHEAL DIAZ	Friendly Village, GA		2005-03-19 17:26:10.0	Immediate
	MEDIUN ESPINO	Friendly Village, GA		2005-03-19 17:26:49.0	Immediate

▼ **Closed Leads - 9**
Click a lead to retrieve the record.

Name	Location	Marketing Source	Closing Date	Closing Action
PEDRO VURGOS	Friendly Village, GA	TV	2005-03-16 16:20:57.0	No Land Option Available
ISRAEL DORANA	Friendly Village, GA	TV	2005-03-16 16:19:12.0	Lost to stick built
VINNEZ PEBRAZA	Friendly Village, GA	NEWSPAPER	2005-02-22 16:41:29.0	Lost to stick built
JUAN MATANCIAS	Friendly Village, GA	TV	2005-03-16 16:31:10.0	No inventory available
MARTIN FERNANDEZ	Friendly Village, GA	REFERRAL	2005-07-07 10:13:09.0	Unqualified Lead
JOSE GUZMAN	Friendly Village, GA	TV	2005-07-07 10:30:49.0	Disconnected Phone Number
SANTOS ESPINOZA	Friendly Village, GA	TV	2005-07-07 10:11:58.0	Unqualified Lead
LETICIA TREJO	Friendly Village, GA	TV	2005-05-04 14:39:47.0	Duplicate Lead
LETICIA TREJO	Friendly Village, GA	TV	2005-05-06 11:43:41.0	Duplicate Lead

Go [back](#) to the other agents leads.

Figure 26 - A Subordinate Agents Leads (Darschewski, 2006).

When the user clicks on a lead the detailed information for that lead is displayed and can be updated. Also from this screen sales documents can be generated based on the information that is currently in the system for the lead. The detailed lead information

is illustrated in Figure 27. Since the user who is viewing this lead information is in a higher *role* that user is allowed to reassign the lead if he or she finds that is necessary.

ARC
Affordable Residential Communities

Home Leads Search Profile Marketing Reporting, etc.

Lead Information

Lead Information

Date Entered: March 19, 2005 Date Updated: March 08, 2006

Community Location: Friendly Village, GA Housing Need: Immediate

Market: --Select--

Lead Origination: TV

First Name: WELLENTON

Last Name: ISURI

Address: N/A

City: LAWRENCEVILLE State: GA Zipcode: 30093

Phone Number (Home): 1231231234 Phone Number (Work):

E-Mail Address:

Best time to contact: Morning Language Preference: Spanish

Interested In

Bedrooms: 3

Bathrooms: 2

Lease To Own: ☒ Finance: ☒ Cash Sale: ☐ Lease: ☐ Lot Rent: ☐

Appointment Information

Set Appointment? No

Update Lead Lead Documents Reassign Lead Close Lead

Current Notes

03/19/2005 05:17:37 PM LEAD
UPDATED 03/19/2005 05:17:37 PM will
follow-up on Monday 03/19/2005
08:55:23 AM may be contacted at any
time, number is a cell. 03/19/2005

Add Notes

Figure 27 - Detailed Lead Information (Darschewski, 2006).

When the user selects the option to create sales documents they are presented with the page seen in Figure 28. Upon choosing a sales program the user is then prompted to enter the unique identifier of the home for which they want to prepare the documents. The user is then presented with a screen that verifies the information that will be transferred to the document, and, if the sales program requires a home price, the user can enter the pricing information. When all required information is completed and the form is submitted the user can then pull the final sales document, which presents itself as a portable document format (PDF) file, complete with the customer information.

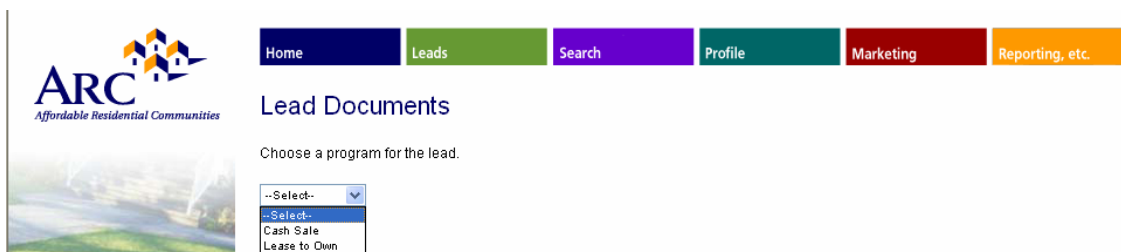


Figure 28 - Lead Documents Page (Darschewski, 2006).

If the user of the application happens to come across a customer who currently lives in one area but is looking to relocate they have the option to enter that lead by going to the Enter Out Of Market Lead item; Figure 29 shows this page. The procedure for entering an out of market lead is similar to the standard lead entry page; the main difference being that the user only has the option to auto assign the lead, and all locations and markets are available in the drop downs.

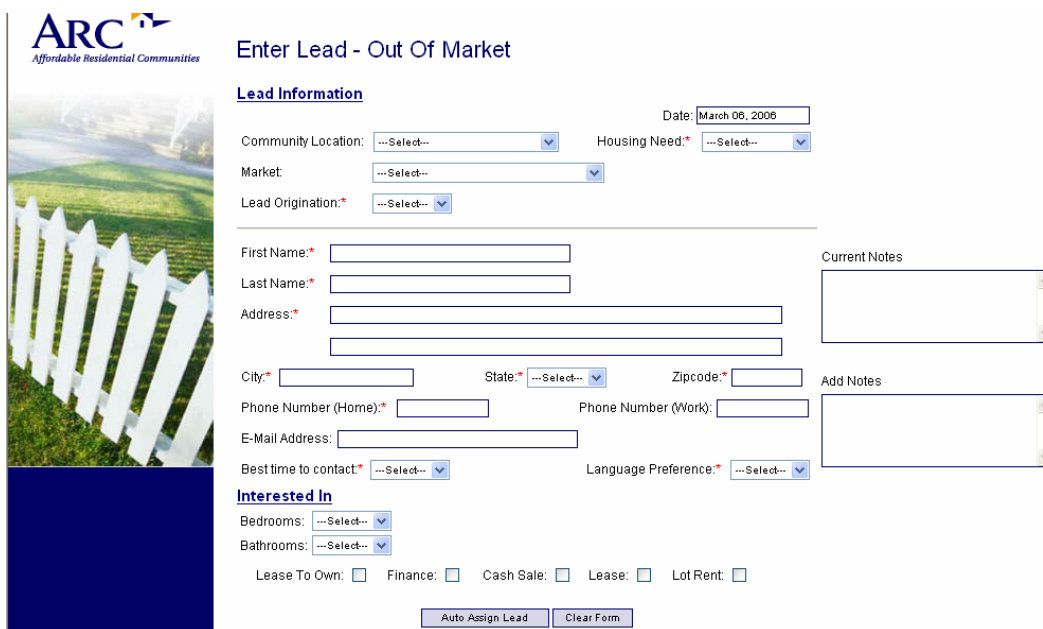


Figure 29 - Enter Out Of Market Lead (Darschewski, 2006).

Search – The search portion of the application does exactly what the name implies; it is there for the user to either run a search for one of their specific leads or to search for the documents created for a specific lead. Figure 30 shows the page a user sees if they want to search for one of their leads. The user can enter some or all of the fields depending on how detailed they want the search to be. They can also use the wildcard character (%) to search on a portion of the leads first or last name.

Figure 30 - Search Leads Page (Darschewski, 2006).

The results of the search can be seen in Figure 31. It lists any leads that *belong* to the user, *or*, if the user is in a certain job role it also shows the leads of the user's subordinates. When the results of the search are displayed, the user can click on a lead and then edit the information for that lead or close the lead.

Customer Name	Assigned To	Status	Location	Entered
JERRY JONES	MICHELE LANGFORD	ACTIVE	Gainesville, FL	12/01/2004
EVAN JONES	SHELIA BREWMINGTON	ACTIVE	Suburban	01/11/2005
EVAN JONES	SHELIA BREWMINGTON	ACTIVE	Suburban	02/09/2005

Figure 31 - Search leads Results Page (Darschewski, 2006).

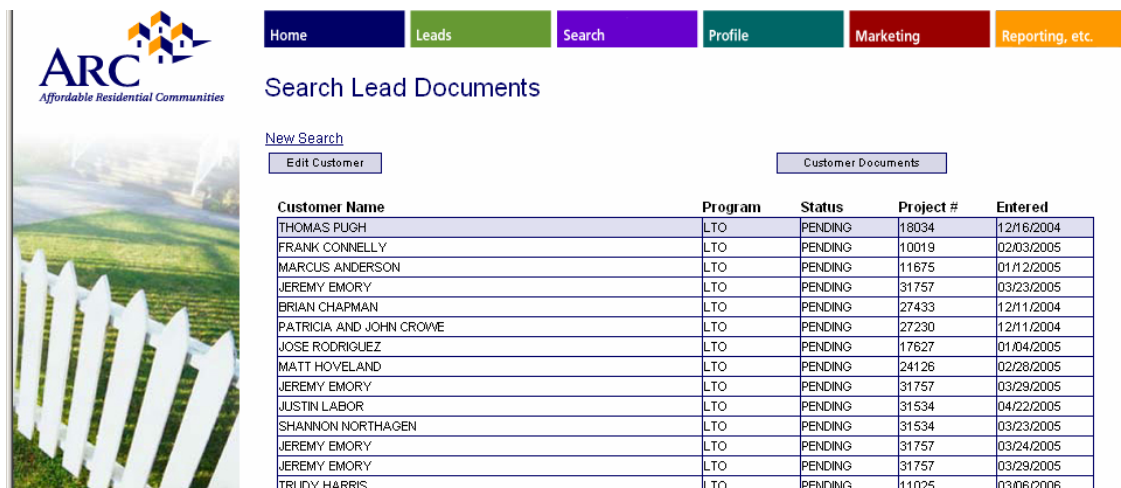
When the user closes a lead they are presented with the screen in Figure 32. On this screen the user is required to choose a reason for closing the lead and is urged to enter any additional information as to why the lead was closed.

Figure 32 - Close Lead Page (Darschewski, 2006).

Users of this application are able to create sales documents for their leads. The Search Lead Documents item allows the users to find the documents which they created. This search interface is as intuitive as the Search Leads interface. The user can choose a community to search, or they can search all communities, and can search on a specific document type, or all types of documents. Figure 33 depicts this interface.

Figure 33 - Search Lead Documents (Darschewski, 2006).

The results of the search will either indicate that no records were found or will be similar to the results seen in Figure 34. The user can then click on the lead, click on the Customer Documents button and be redirected to a page where they can choose the specific document they want to print, pre-filled with the lead's information.



Customer Name	Program	Status	Project #	Entered
THOMAS PUGH	LTO	PENDING	18034	12/16/2004
FRANK CONNELLY	LTO	PENDING	10019	02/03/2005
MARCUS ANDERSON	LTO	PENDING	11675	01/12/2005
JEREMY EMORY	LTO	PENDING	31757	03/23/2005
BRIAN CHAPMAN	LTO	PENDING	27433	12/11/2004
PATRICIA AND JOHN CROWE	LTO	PENDING	27230	12/11/2004
JOSE RODRIGUEZ	LTO	PENDING	17627	01/04/2005
MATT HOVELAND	LTO	PENDING	24126	02/28/2005
JEREMY EMORY	LTO	PENDING	31757	03/29/2005
JUSTIN LABOR	LTO	PENDING	31534	04/22/2005
SHANNON NORTHAGEN	LTO	PENDING	31534	03/23/2005
JEREMY EMORY	LTO	PENDING	31757	03/24/2005
JEREMY EMORY	LTO	PENDING	31757	03/29/2005
TRUDY HARRIS	LTO	PENDING	11025	03/06/2006

Figure 34 - Search Lead Documents Result Page (Darschewski, 2006).

Marketing – The marketing portion of the application is used mainly as an administrative tool; allowing certain job roles the ability to add or remove certain marketing promotions from the system.

The user can use the screen in Figure 35 to add a marketing source to the system. The user must pick the market of where the promotion is taking place or they have the option to apply this promotion to all markets. When the user completes this form, and if the source was marked as an active source, then it is immediately available to the sales staff for use.



Source Information

Market:

Lead Origination Source:

Media Source Type:

Is this an active source?

Figure 35 - Add Marketing Source Page (Darschewski, 2006).

Since certain users are allowed to add marketing sources it is fitting that they have a place to edit the sources which they entered. Figure 36 is the Edit Marketing Source page. The user will select one of the marketing sources that were added, and upon selecting it they will be brought to a screen similar to Figure 35, but with the form fields already populated with information of the selected marketing source.

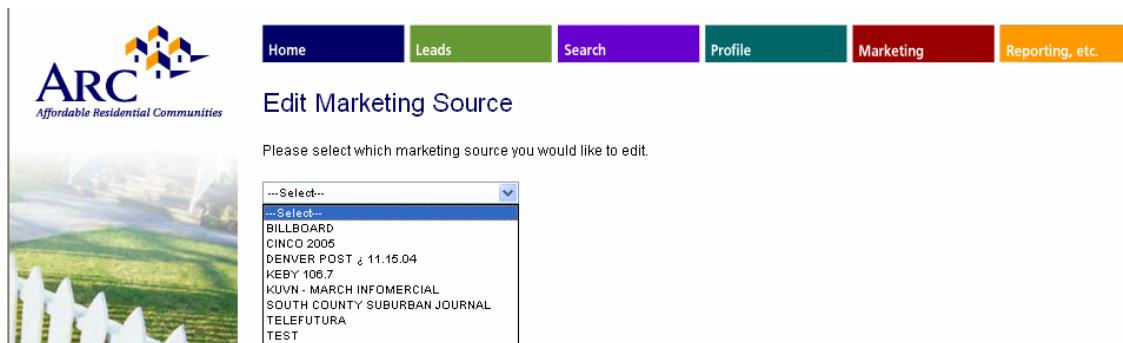


Figure 36 - Edit Marketing Source Selection Page (Darschewski, 2006).

Reporting, etc – The reporting portion of the application is important for managers who use the application. It allows them to report on different aspects of the leads entered into the application, and also allows them to communicate with their subordinates by sending messages to them, who, in turn, will see them when they log into the application.

The Reports item allows users to run reports on the data collected by the application. These reports are housed in ARC's reporting application which is an independent system. The reports allow managers to make informed decisions about how their sales staff is doing, and also to help them predict how many homes should be sold, leased or financed within a given period, based on past performance of their district or region.

Figure 37 shows just one of the reports managers could run using ARC's reporting system. The user running the report first chooses the type of report, in this case

Lead Activity, then they would pick a location for the data they want to report on, whether or not they want to see report details or just a summary, and finally they select the date range for the data they want to report on.

Lead Management:

- Lead Activity
 - District Managers
 - Wyoming District
 - Detail

FROM thru TO

Figure 37 - Example Lead Management Report Flow (Darschewski, 2006).

To communicate in a manner exclusive of e-mail the Send Message item is available to privileged users. Figure 38 shows that the user merely needs to enter the title of the message they want to send and the message itself, and upon submission the message is immediately available to the user's subordinates.

Figure 38 - Send Message Page (Darschewski, 2006).

The Database:

Aside from the interface there are aspects of this application which occur *behind the scenes* in the backend database. A sequence was created for the purpose of generating the unique IDs in the PROFILE table, the LEADS table and the

FORGOT_PWD table. In addition, nearly three dozen functions were created along with 11 stored procedures and four packages. Of these objects there are a few of particular interest, while the others were created to perform supportive functionality of the application, such as performing tax calculations.

The APP_USER_SECURITY package is of interest because often times user's passwords are stored in the databases as clear text which could have security implications. The APP_USER_SECURITY package contains a function that uses the DBMS_OBFUSCATION_TOOLKIT.MD5 function to hash a password based on the input of the user's username and their password. The package body also contains a function to check whether or not the user entered a valid password. This is done by calling the GET_HASH function again and verifying that the hashed password stored in the table is the same as the result of hashing the password entered.

There were a few stored procedures created to aid in the automatic assignment of sales leads. The P_LEADASSIGN procedure was developed for automatically assigning leads that were directly entered into the system through the application's Web interface or the XML upload interface. This procedure can take one parameter as input, the ID of the lead (LID). Based on this LID, and through the use of other functions that were developed, the procedure determines what sales staff is available to take over that lead for the geographic area the lead is located in. The procedure identifies if the lead is Spanish speaking and if it is then priority is given to bilingual sales staff. If there are multiple salespeople available to take the lead then the procedure uses a round robin approach to determine whom the lead should go to. The round robin approach is based on the time the salesperson last received a lead. The salesperson that has the longest time span

between lead assignments is awarded the lead. This procedure also takes into account the job role hierarchy, starting from the bottom of the hierarchy and working upward.

The P_LEADREASSIGN procedure is very similar to the P_LEADASSIGN procedure except the P_LEADREASSIGN procedure excludes the salesperson that let the lead get stale. Also the P_LEADREASSIGN procedure is scheduled to run automatically every hour to scan the system for leads that have not been accessed in 24 hours. Upon finding a stale lead the procedure would perform its search for another salesperson to assign the lead.

Both the P_LEADREASSIGN procedure and the P_LEADASSIGN procedure incorporate a MAIL_NOTIFICATION procedure that was developed to allow for automatic notification of new leads via e-mail to the salesperson that the lead was assigned to. The notification e-mail includes all pertinent information for the lead, including a running history of any notes that may have been entered, so that the new salesperson can see what may have been done with the lead and what type of housing option they are interested in.

How the project was managed:

Managing this project ended up being a joint effort between this author and the staff at ARC. The initial timeline for the completion of the application was eight weeks, so it was within that time frame that development and implementation of the CRM application had to be completed. However, since members of ARC's staff performed the testing and quality assurance of the application by individual module, it was vital that each module met a specific deadline. The importance of this was so that the bugs found

in a module or changes to a module were needed, these changes could be fixed when time permitted. Bugs found during the testing of the application could have potentially impacted meeting the deadlines for subsequent modules, depending on the severity of the bug. The timeframe for the entire project, as well as each individual module, can be seen in Appendix B.

Significant events/milestones in the project:

This project presented a number of significant milestones. One of the major milestones in this project occurred near the beginning. This was when a complete understanding of the business processes, business logic and goals that ARC was trying to achieve, by implementing the application and by developing the application rather than using an *off the shelf* CRM solution, was attained.

Another major milestone occurred upon the completion of the design of the database for the system. It was imperative for the application to have an efficient database design which was normalized as much as possible because of the amount of data that could potentially be stored in this database. An un-normalized database could have proven to be detrimental to the application due to the effects it could have had on its performance.

During the development of this project one of the more challenging issues was programmatically routing leads that were entered into the system by members of the call center. These leads needed to be routed to the correct agent based on the location of the lead, as well as the language preference of the lead. What made this a challenge was that a round robin approach was needed to allow each of the agents who could receive the

lead an equal opportunity. This logic of assigning leads also had to take into account that there were regional or district personnel who could be in the same geographical location of the lead, but would not be allowed to receive it because their profile prohibits them from receiving leads. It was a major milestone when the stored procedures that performed these tasks were finally completed and working as intended.

Each of the modules that were developed for this application presented their own challenge. When all of the modules were completely developed and tested it was a significant milestone in this project, the reason being was that this indicated the project was nearing completion and soon it would be used by the sales staff in *real world* situations.

Since all of the testing and suggested changes to the prototype were being performed in conjunction with the development of the other modules, the most significant event of this project was when the application was moved from the test environment to the production environment, available for use by all of the sales staff at ARC.

Changes to the project plan:

Change is inevitable especially when dealing with developing a piece of software. The changes to the project plan for this project were minimal. The most significant change occurred when the owner of the project at ARC decided that they needed a way to upload leads that they may receive from an outside source. These leads were provided to ARC in the form of an extensible markup language (XML) file. An administrator of the lead management application was to be able to go to a specific page in which they would be able to browse for the XML file they wanted to upload. Upon submitting the file a

stored procedure would parse the file and insert the data into the necessary tables. This change was made after the initial testing of the application and while it did not have an impact on the *go live* date for the entire company to use the application, this piece of the project still needed to go through the proper testing and debugging. The testing of the upload feature pushed the implementation of the entire application from the test environment into the production environment out a couple days.

Evaluation of whether or not the project met project goals:

The primary goal of this project was to develop a Web-based CRM application for the sales staff at ARC so that they would be able to manage and act on their sales leads more efficiently. Since the implementation of the application more than 19,000 sales leads have been entered into the application. The fact that so many leads were entered into the system shows that the application is being used frequently for the management of potential customers, and therefore, achieved the goals and expectations of the project. An additional goal of the application was that it was to be available to the sales staff regardless of their geographic location. This goal was accomplished due to the fact that the application was developed as a Web-based application, which allowed for the sales staff located throughout the United States to access it via the Internet.

What went right and what went wrong in the project: Throughout the life of the project the number of things that went right outnumbered the number of things that went wrong, however, when things went wrong the solution was not straightforward.

The overall design of the database was fairly simple, with the exception of the job roles and the business logic by which users can create other users, therefore, it can be said that things went right in that respect. The database contained numerous tables and constraints, but having normalized it and having thoroughly thought through its design before actually scripting it helped to simplify this process.

The coding of the application was in line with expectations also. Throughout the project it was expected that the majority of the application would use HTML tags, JavaScript and CSS along with underlying ColdFusion logic embedded in the application pages. However, trying to put too much of the business logic into the ColdFusion code rather than into database objects proved to be problematic resulting in an application that, at times, would run slowly. This issue was addressed by moving some of that logic from the ColdFusion pages to the backend database, but some of the code had to stay in the front-end pages. As a result, certain areas of the application still are not performing as efficiently as possible and could be improved.

Another suboptimal event was that the code was insufficiently modularized to allow for efficient code reuse. If the code was modularized the application would allow for easier future additions and modifications. By not modularizing the code nothing is being taken from the application, however, it is proper development practice to use reusable code whenever possible.

Using the Internet site of ARC as a *template* for this application was an excellent decision. The pull-down menu system allowed for restricted access to different modules based on a user's roles, and the layout was well received by both the users and management at ARC.

One of the most notable areas where the project did not go as planned was in the communication between this author and ARC regarding deadlines. A project plan was developed so that specific modules would be designed on a schedule. The problem was getting ARC to dedicate the staff needed to thoroughly test these modules and to report any bugs that were found. By not allocating proper resources in terms of personnel to the testing of the project, ARC increased the risk of not having the application developed in the timeframe they wanted.

Another area where the project went wrong was in failing to thoroughly test the speed of the application at its maximum usage. Having failed to do this the application began to slow a few months following implementation. ARC was not willing to put the application through aggressive testing due to the urgency of implementing it into the production environment. Their thought was to allow users to do performance testing in the production environment and then revisit any issues that may arise. Final testing of an application in the production environment is not the suggested way of performing *stress* testing.

Project variables and their impact on the project:

The most significant variable that impacted this project was the speed of the application after it was deployed to the production environment and in use by hundreds of

sales staff throughout the United States. In the first nine months of the application being in use, more than 15,000 leads were entered into the system, which averages to approximately 55 leads per day. Despite the fact that the application was designed to handle a large number of records, being able to test it with that number of records was unrealistic. Approximately four months into the application's use there was a noticeable difference in the speed when compared to the speed when it was first deployed. As more and more leads and users were added to the system the queries and logic to determine what users had access to other users and what leads users can see began to take its toll. For the average user this did not have as much of an impact as it did for those users who were in management roles in the system. These users, since they had subordinates they could assign leads to and subordinate leads they could look at, would experience a wait time of about ten to fifteen seconds when accessing the pages that would find their subordinates. While this is not a long time to wait it was too long for these users. This issue was resolved through better use of indexes on the backend database and through the use of hints in the front-end SQL queries.

Another variable that impacted the project was the underestimation of the learning curve that would be involved for both the users and administrators of the system. Having developed the system, this author believed that the interface was intuitive and that the application flowed consistently. The initial testing of the application with a small group of users also revealed its intuitive design. However, once the application was deployed there were multiple problems with users not knowing where to go to perform certain tasks. This may have been partly due to the minimal training ARC provided, but could also be attributed to the user interface. The impact of this issue was that users were

hesitant to use the application until they were forced to do so by their superiors. There was a strong push by management to have all sales staff use this application, not just so they can track their leads but so management can see what marketing initiatives could be used to increase sales. If the staff did not use this system then the ultimate result was to terminate their employment.

The CRM application began as an idea someone had at ARC. Nobody at that time knew the impact it would have on ARC's business or how much data would actually be collected by having the sales staff use this application. This project presented itself with minimal variables that could have an impact. Yet, while there will always be unknowns developing an application of this magnitude, the satisfaction comes in dealing with those unknowns in the proper manner. In this project the proper manner in dealing with application issues was to fix the problems that the users found while testing. As far as the design and training issues that occurred, ARC management ultimately accepted the application and liked the layout, they just did not do enough to properly train their staff on how to use it.

Findings / analysis results:

Application Analysis:

Overall the actual CRM application had positive effects for ARC. As more and more sales staff started to use the system they saw the potential that it had to assist them in tracking their customers. Management at ARC also saw the potential of the application, as they now had a central repository for their customers and an application that could be modified and expanded if they needed. ARC could have used a third party

CRM application that would have also allowed them to store data, but it may not have allowed for the flexibility of a custom built application.

Data Analysis:

Using ARC's reporting system sales reports were generated for the periods of January, 2005 through September, 2005 as well as the same period one year prior. Using these reports charts were made using Microsoft Excel to show the sales trends.

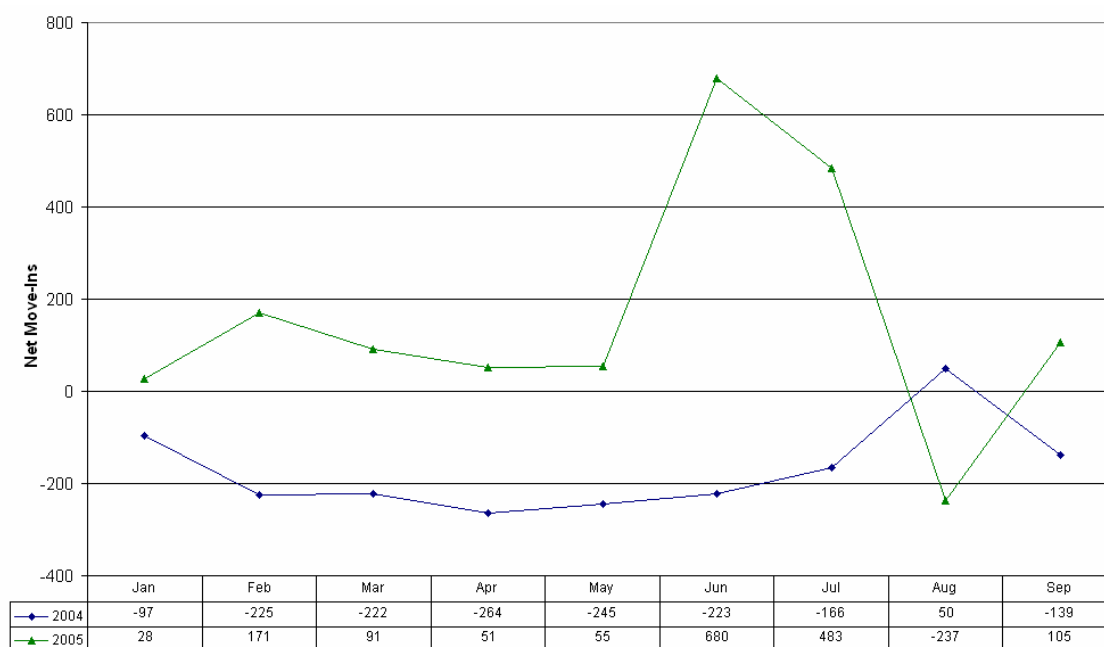


Figure 39 - ARC Move-Ins Less ARC Move-Outs (Darschewski, 2006).

Figure 39 shows the number of residents that moved in minus the number of residents that moved out for the given time period. It can be seen that during the period of January, 2005 through the end of September, 2005 the number of move-ins was greater than the number of move outs for the same period one year earlier. This increase in residents may be attributed to the implementation of the application to track potential sales, but it may also be the result of a stronger push to increase occupancy by

management. There was not any data available through ARC that would have indicated other factors that may have attributed to this increase in resident move-ins. However, the non-modular housing market was booming during this timeframe, which typically affects the modular home industry in an inverse way causing more residents to move out.

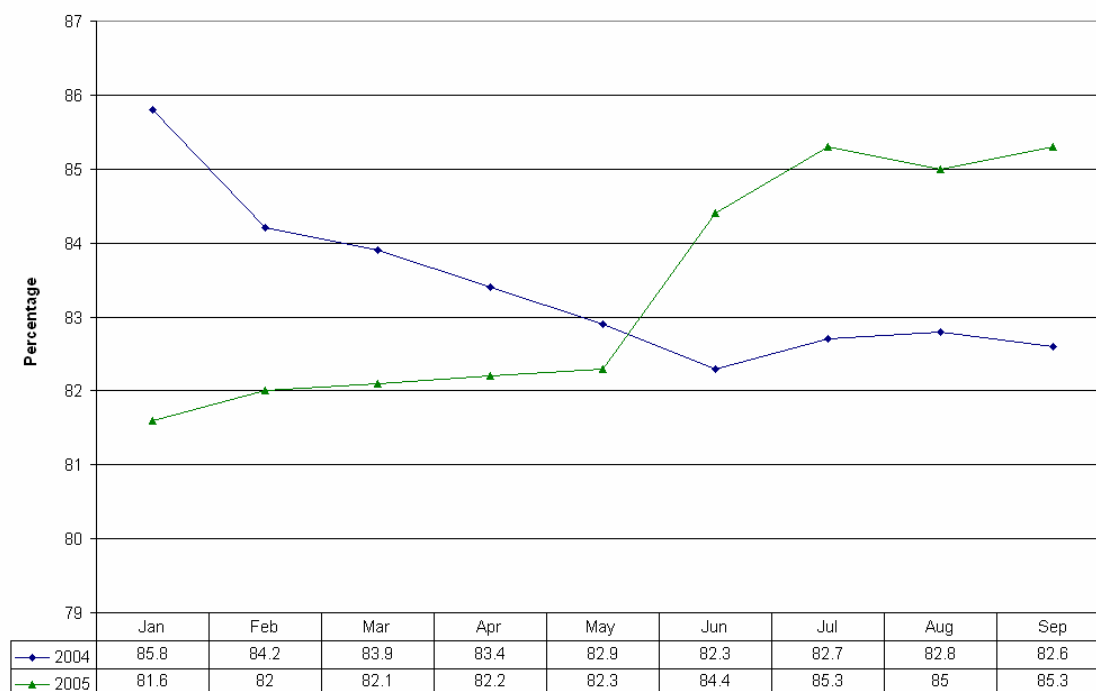


Figure 40 - Percent of ARC Home sites occupied (Darschewski, 2006).

Figure 40 depicts the percentage of home sites that ARC operated that are occupied. The chart indicates that the percentage of sites occupied in 2004 was slowly decreasing as the year progressed. In 2005 the percentage of occupied homes began to rise again. The decrease in occupancy in 2004 could be attributed to a large acquisition of poorly performing properties that had low occupancy to begin with. However, in September, 2005 the percentage of occupied sites begins to approach the beginning percentage of occupied sites in January, 2004. Again, this could be attributed to the sales staff having access to the lead tracking application or other factors, such as increased

incentives to the sales staff to increase occupancy and other promotional programs initiated by ARC. According to the data in the database approximately 20% of all leads that were closed moved into an ARC community, and while this may not seem like an outstanding number, the reason for the application was to be able to track potential customers and mine customer data, not to just increase occupancy.

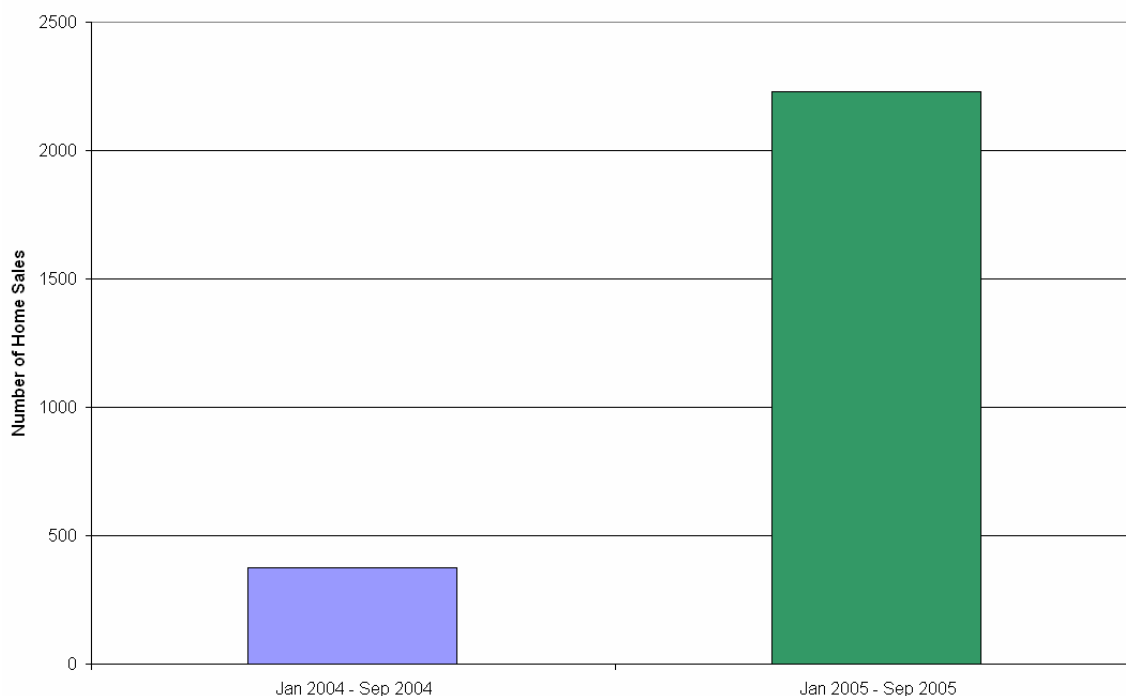


Figure 41 - ARC Home Sales (Darschewski, 2006).

Finally, Figure 41 is a graph showing the number of home sales for the given period. The number of home sales in the first nine months of 2004 was dramatically lower than the same period one year later. In fact, in 2005 it appears that the number of homes sold was more than four times that of the same period in 2004. During 2005 ARC had a strong push to sell homes and there were benefits to the sales staff to sell. This could attribute for such a significant increase in the amount of home sales, but it could also be due to the fact that the sales force now had a system in place to aide them in their sales.

Summary of results:

This project began with a need for a tool to reliably track potential sales leads at ARC. Since the employees that had a need for this application were dispersed throughout the United States it was imperative that it be available over the Internet. The end result of the project was a fully functional lead management application. This application allowed its users to input potential customer information, create sales documents for these customers, search the backend database for their customers and ultimately to report on the data. The application also allowed for the majority of the application's maintenance to be performed by privileged users of the system. These users were allowed to add marketing sources, create users, and upload leads from third party XML files.

Using ARC's reporting system and looking at the raw table data, it was determined that the nine months following deployment of the application, ARC's net resident move-ins had significantly increased compared with the same period one year earlier. It was also shown that the percentage of occupied home sites that ARC manages also increased following a steady decrease one year prior. A closer look at the data in the database revealed that approximately 20% of the leads that were closed actually moved into an ARC community. Finally, the number of home sales that occurred in the first nine months post-deployment had nearly quadrupled when compared to the same time one year prior.

Providing ARC with a suitable application to track their sales leads proved to be a vital tool for their business. No longer did ARC need to rely on using Microsoft Excel spreadsheets to track potential customers, and no longer did they have inconsistent data throughout the company. Management is now able to collect the data they require, store

it in a central repository, and mine the data they feel is important to generate sales prediction reports, and market analysis reports.

Chapter 5 – Lessons Learned and Next Evolution of the Project

Lessons learned from the project experience:

Throughout the course of this project a number of valuable lessons were learned. One of the most significant lessons was how to take a concept for an application and, by using the SDLC, turn that concept into a working application. This author also learned that while there may not be a defined methodology for building a Web-based application there are methodologies available that can be adapted for Web development. This may be of utmost importance to ARC since many of their internal applications are Web-based and by seeing the success of using such a methodology, they may strive to implement its usage as part of their standard development procedures.

Another important lesson learned was how to effectively work with management in setting goals, getting them to allocate resources, in terms of both personnel, hardware and software, and in communicating ideas to them in a way that was clear and concise. It was important in this application to set goals, in terms of timeframe, for each of the individual modules of the application. This not only helped to keep ARC focused on the project, but showed them that progress was being made which generated excitement among management. Equally important was communicating with ARC about the ideas this author had in regard to the design of the application. If the application was designed strictly by the conceptual screens given then the application probably would not have been as successful, if at all.

Learning about the effect of different menu structures that can be used on Web pages, and how those menu structures can notably change the usability of a Web page, was an important lesson for this project. This knowledge allowed for making a well

informed decision on the type of menu that should be used in this application. If the wrong type of menu structure was used the users may not have felt as comfortable using the application and the project could have been a failure. Of all comments made about the application, there were never any negative comments by the users about the menu system.

It was important in this project to perform backend database processing to aide in the application's speed and to perform certain steps of the business logic of the application. This application allowed the author to expand his knowledge of PL/SQL through the need of stored procedures and functions. Implementing this code as inline ColdFusion code would have been more difficult. During this project this author was also able to learn how to retrieve XML data from the Internet, parse that data with Oracle and insert the data into the appropriate tables.

Probably the most important lesson learned throughout this project was about perseverance. Often times a situation was encountered where this author did not know how to proceed any further, however, by taking a step back and revisiting the problem a solution would manifest itself. Perseverance also played a part due to the fact that this author had a timeline to follow in terms of developing the application, and while at times it was difficult to work on the application it needed to be done. A specific instance of when perseverance played a part in the project was when ARC added the requirement of specific users being able to upload the leads via an XML file. This was something that this author had never done before and really had no idea where to begin. Through the examination of numerous code samples and by searching the Internet, the answer on how to programmatically do this with ColdFusion and Oracle was found.

What would have been done differently in the project:

If the opportunity to do the project over again presented itself there are number of things that would have been done differently. One area where changes would have been made is in the processing of portions of the SQL code. Much of the SQL, such as the code to determine what role a user is part of or what locations that user has access to, is done within the front end ColdFusion page. This was done because at that time it was the only way this author knew how to code the application, however, through experience he has learned that the majority of times it is more efficient to allow the database processing to be performed in the database rather than in the application. This could have been accomplished by creating more stored procedures and functions in the database and calling those objects from within the ColdFusion code.

For an application this size it should have been a requirement to provide external documentation to aide in future changes or revisions rather than only documenting the code in line. The development of user documentation should also have been a priority in this project because that documentation would have been useful during user training, as well as during the testing of the application.

Since the number of leads that could be added to the system might be in the hundreds of thousands, it would have been a good idea to develop more administrative functionality into the system. The ability for certain users to delete a lead from the system, maybe a lead that was entered twice or a lead that was entered because one user was training another on how to use the application, is an example of the types of administrative tasks that could have been added. Another administrative task that could have been put into the hands of the users is the ability to completely remove a user from

the system that was incorrectly entered. As the application operates now the DBA at ARC needs to go into the database to run a function that can completely remove a user from the system.

Whether or not the project met initial project expectations:

The initial expectation of this project was to develop an application for ARC which would allow their employees to have a better tool for tracking their sales leads. When the application was deployed for all users, a feeling of accomplishment was felt by both this author and the management at ARC. The expectation to gain a better understanding of the SDLC to develop a Web-based application was also anticipated. Finally, expanding this author's knowledge of Oracle was also expected. This occurred in the development of the application due to the number of stored procedures and functions that had to be created, as well as the creation of a user-defined data type, and using those objects in the application.

Next stage for the project if it were continued:

This project could easily be continued and evolved from its initial code base. If ARC wanted to begin collecting additional personal information from its customers, such as social security number, then the next stage of this project would involve the implementation of Secure Sockets Layer (SSL). SSL would encrypt the data being transmitted over the Internet and may provide reassurance to ARC's customers that their vital information is not going to fall into the wrong hands. The use of SSL, however, could affect the speed of the application. If SSL were to be used more of the business

logic that is currently in the front end ColdFusion pages would need to be moved to the backend database for processing. Even if SSL was not implemented, migrating some of the front end code and having it run on the backend, in Oracle, would also help to improve the performance of the application.

A major development hurdle, which may need to be addressed in the next stage of this application, involves allowing the sales agents to be able to access the application from their Blackberry or any other handheld device. This would require ARC to re-code multiple parts of the application to maximize its speed and allow it to be easily read on such a small display. Allowing this application to be used on a handheld device may make the sales staff more responsive to checking in on the leads they currently have and to future leads that are assigned to them.

As a final point, the ability for customers to be able to enter in their own information at one of the sales locations, at a kiosk or even as a form external to this application, might also be in the future of this application. This could minimize the traffic at the call center where the majority of leads are entered either from phone calls, external web forms or e-mails to the call center.

Conclusions / recommendations:

Based on the success of this application it can be concluded that to implement a CRM solution a company does not necessarily need to buy an application *off the shelf* that was designed to accommodate all business models. Through the use of ColdFusion and Oracle a lead management application was developed which had the potential to help the sales staff with tracking their customers, and additionally, in the sale of homes.

ColdFusion and Oracle technologies were used for this project primarily because they were already in use at ARC and they were fully licensed to use them. This application could have been developed using open source technologies like PHP and MySQL or by using a Microsoft solution of ASP.net, C# and SQL Server. Regardless of the technology used, the concepts behind developing a Web-based application in house, as opposed to purchasing an application *off the shelf*, remain the same.

Summary:

Completing this project and this application taught this author about the process of developing a database backed Web application. Working on the application from the initial concept through the final implementation to the production environment allowed for the use of the Prototyping Life Cycle for development. Working on this project in a business environment helped to develop this author's skills of meeting with users to understand what their needs are, as well as meeting with management to develop a timeline for completion of the application. Having a timeline to follow and being the sole developer of the application also helped to refine project management skills.

Although the application was a success there are a few areas of the project which could have been done differently. More backend database processing to alleviate overhead on the front end of the application is one of these areas of improvement. Improvements can also be made in providing better user and training documentation, and the addition of more maintenance related functionality built into the application so that it can be a truly self-contained system.

Although the project met initial expectations there is room for improvement in evolving the application into a more sophisticated system. Such improvements may include the integration of SSL into the application to secure customer information, the development of a version of the site to be used on a Blackberry device or handheld, and possibly the development of a section of the site for use in a kiosk or on a third party web page. These improvements may not necessarily affect the functionality of the application but could allow the users of the application to be more efficient.

The use of ColdFusion and Oracle technologies is powerful enough to develop a successful Web-based lead management system. There are numerous combinations of server-side languages and databases, both fee-based and open source, that could have produced similar results, but since these technologies were already in use at ARC it made sense to take advantage of them. Having created a custom application, ARC is now able to easily modify and add features to the application so it can evolve with their business model.

The use of this application will fill the void ARC had in keeping a central repository of their potential sales leads. Having this information in a database will also allow management at ARC to mine the data and use it as a guide in making more informed decisions about the purchasing habits of their customers and the success, or lack of success, of certain marketing campaigns. The application will also allow for the sales staff to work more efficiently and organized.

Works Cited

- .NET Framework Developer's Guide* (n.d.). Retrieved, January 17, 2006 from <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconaspnetplatformrequirements.asp>
- Alapati, Sam R. (2003). *Expert Oracle 9i Database Administration*. New York, NY: Apress.
- Artz, John M. (1996). A top-down methodology for building corporate Web applications. *Internet Research*, 6, 64-74.
- Beynon-Davies, Paul (2004). *Database Systems*. New York, NY: Palgrave Macmillan.
- Boehm, Barry W. (1988). A spiral model of software development and enhancement. *Computer*, 21(5), 21-42.
- Brooks-Bilson, Rob (2003). *Programming ColdFusion MX, 2nd Edition*. Sebastopol, CA: O'Reilly & Associates, Inc..
- Buehrer, Richard E., Senecal, Sylvain, Pullins, Ellen Bolman (2005). Sales force technology usage – reasons, barriers, and support: an exploratory investigation. *Industrial Marketing Management*, 34, 389-398.
- Chen, Kuanchin, Yen, David C. (2004). Improving the quality of online presence through interactivity. *Information & Management*, 42, 217-226.
- Chen, Mu-Chen, Chiu, Ai-Lun, Chang, Hsu-Hwa (2005). Mining changes in customer behavior in retail marketing. *Expert Systems with Applications*, 28, 773-781.
- Fioravanti, Fabrizio (2005). *Skills for managing rapidly changing IT projects*. Hershey, PA: Idea Group Publishing.
- Fernández-Medina, Eduardo, Piattini, Mario (2005). Designing secure databases. *Information and Software Technology*, 47, 463-477.
- Hassan Ahmed E., Holt, Richard C. (2004). A lightweight approach for migrating Web frameworks. *Information and Software Technology*, 47, 521-532.
- Huizingh, Eelko K.R.E. (2000). The content and design of web sites: an empirical study. *Information & Management*, 37, 123-134.
- Lemieux, Jacques R. (1996). Using RAD tools to develop secure client/server applications. *Computers & Security*, 15, 289-295.

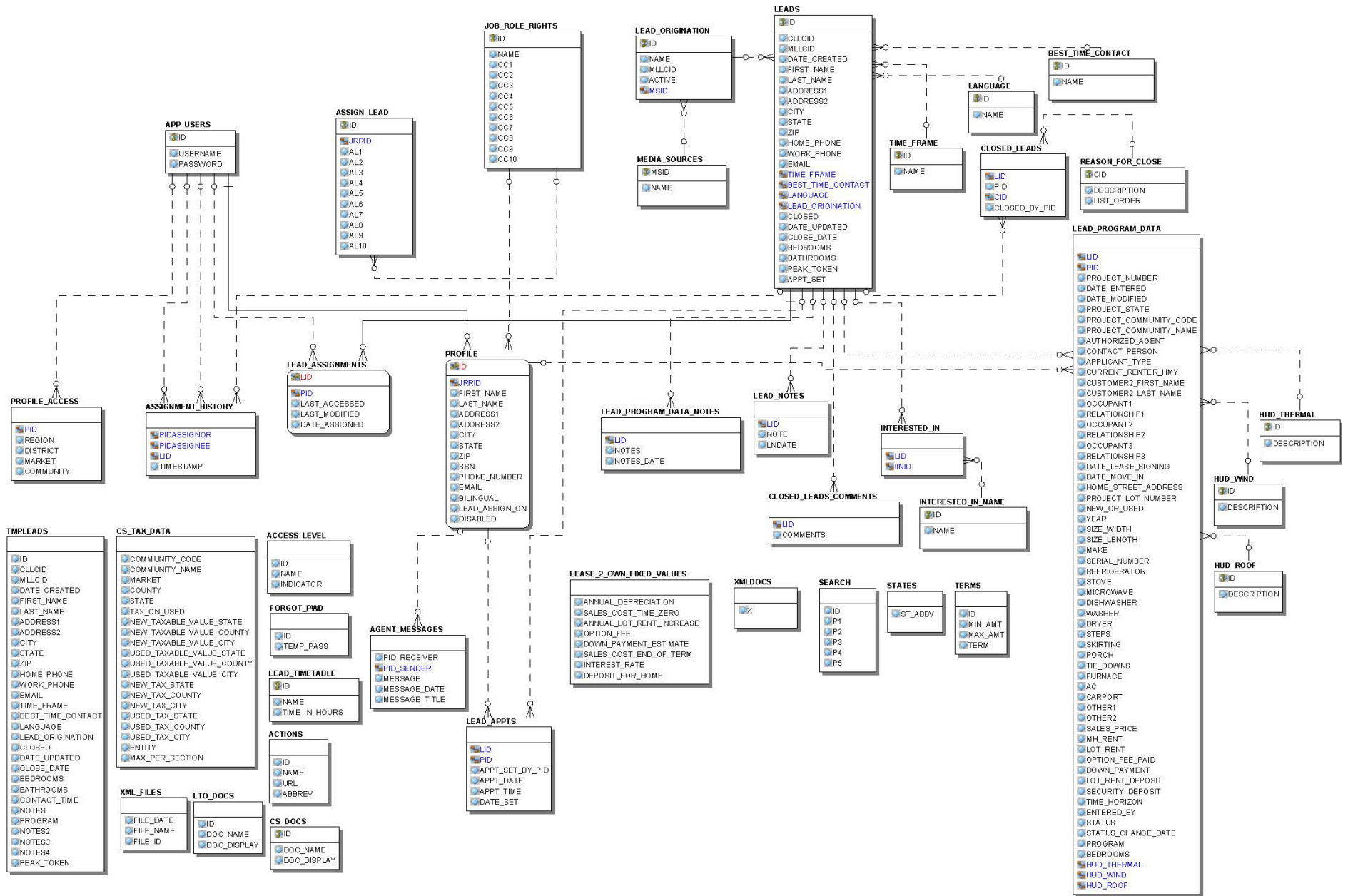
- Liao, Shu-Hsien (2005). Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert Systems with Applications*, 28, 93–103.
- Liberty, Jesse, Hurwitz, Dan (2005). *Programming ASP.NET*. O'Reilly & Associates: Sebastopol, CA.
- Loshin, Paul (2001, January 8). *Relational Databases*. Retrieved January 16, 2006, from <http://www.computerworld.com/databasetopics/data/story/0,10801,55918,00.html>
- McNeil, Sara G., Robin, Bernard R. (2000). Facilitating collaboration, knowledge construction and communication with Web-enabled databases. *WebNet 200 World Conference on the WWW and Internet Proceedings* 7p.
- Roh, Tae Hyup, Ahn, Cheol Kyung, Han, Ingoo (2005). The priority factor model for customer relationship management system success. *Expert Systems with Application*, 28, 641-654.
- Sharma, Arun, Sheth, Jagdish N. (2004). Web-based marketing: the coming revolution in marketing thought and strategy. *Journal of Business Research*, 57, 696-702.
- Sklar, David (2004). *Learning PHP 5*. Sebastopol, CA: O'Reilly Media, Inc..
- Soh, B.C., Young, S. (1998). Network system and world wide web security. *Computer Communication*, 20, 1431-1436.
- Sohn, So Young, Lee, Ji Soo (2005). Cost of ownership model for a CRM system. *Science of Computer Programming*, 60, 68-81.
- Standing, Craig (2002). Methodologies for developing Web applications. *Information and Software Technology*, 44, 151-159.
- Suh, Woojong (2004). *Web engineering: principles and techniques*. Hershey, PA: Idea Group Publishing.
- Thompson, Bob (2002). *What is CRM?* Retrieved December 29, 2005, from http://crm guru.custhelp.com/cgi-bin/crmguru.cfg/php/enduser/std_adp.php?p_faqid=416&p_created=1018758641&p_sid=YVT1iiYh&p_lva=&p_sp=cF9zcmNoPSZwX3NvcnRfYnk9JnBfZ3JpZHNvcnQ9JnBfcm93X2NudD0xMzAyJnBfcHJvZHM9JnBfY2F0cz0mcF9wdj0mcF9jdj0mcF9zZWZyY2hfdHlwZT1hbN3ZXJzLnNIYXJjaF9ubCZwX3BhZ2U9MQ**&p_li=&p_topview=1

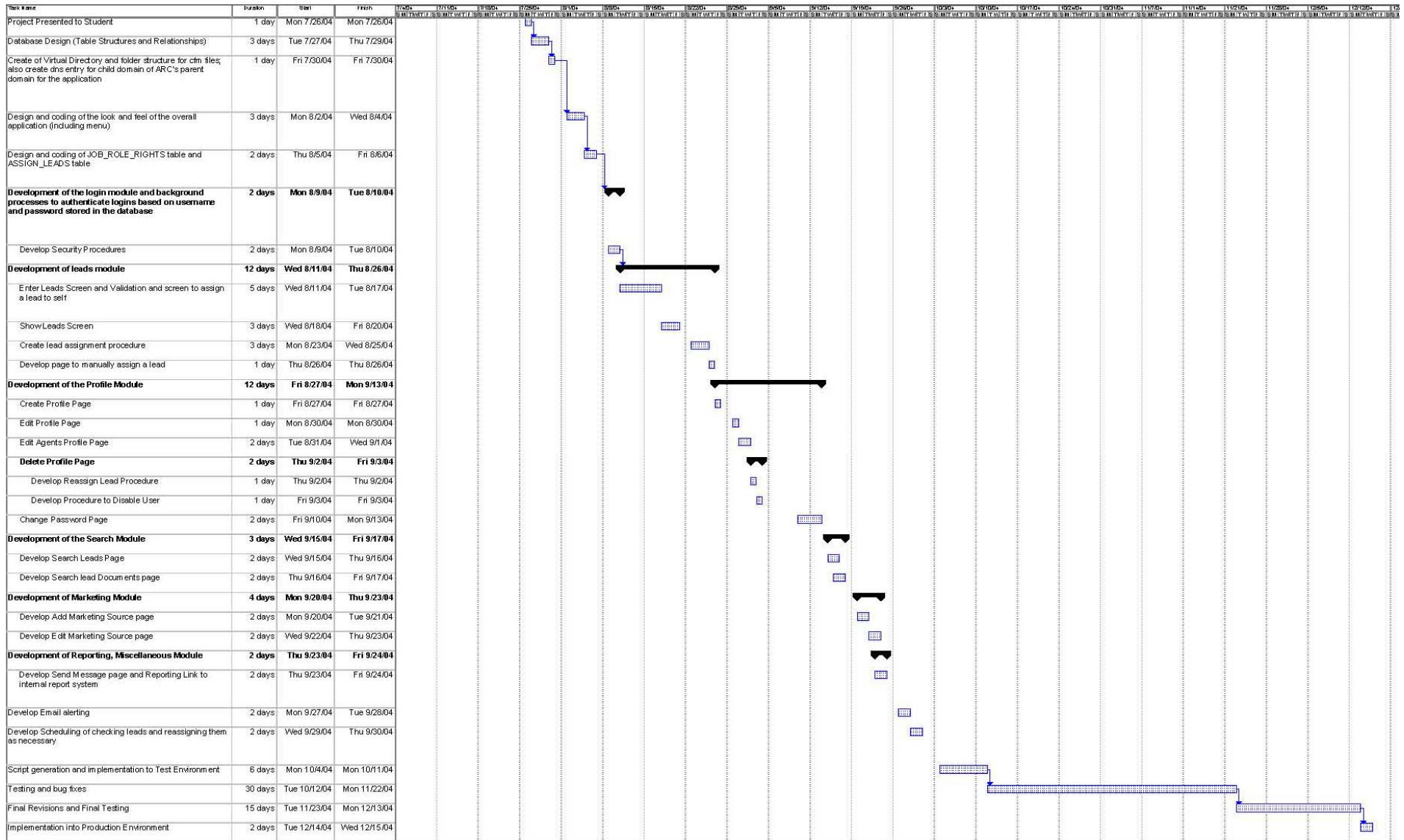
What is SQL Server 2005? (n.d.). Retrieved December 29, 2005, from
<http://www.microsoft.com/sql/prodinfo/overview/what-is-sql-server.msp>

Westman, Stephan (2002). Building database-backed Web applications: process and issues. *Information Technology and Libraries*, 21 i2, 63-64.

Yu, Byeong-Min, Seak-Zoon Roh, and Sungwook Han (2001). The effect of different menu styles on the user's perception and performance on the WWW. *Annual Topics on Distance Learning Conference*.

Appendix A





Appendix C

Design and Implementation of a Web-Based Lead Management System Using Oracle and ColdFusion Technologies

James Darschewski
Regis University
MSCIT Professional Project
Draft 1 – March 12, 2006

Agenda

- ❑ Project Introduction
- ❑ Web & Database Technologies
- ❑ Web Methodologies and Design Considerations
- ❑ Project History
- ❑ Lessons Learned
- ❑ Next Evolution of the Project
- ❑ Conclusion
- ❑ Questions and Answers



Project Introduction

- ❑ In business a key to being successful is being able to identify who your customers are and evaluating their needs
- ❑ ARC manages over 67,000 home sites in more than 300 properties



ARC's Need For A Lead Management System

- ❑ Fill the void between sales staff and the customer as well as the sales staff and ARC's upper management
- ❑ Highly mobile and distributed sales staff
- ❑ Operations in 29 states
- ❑ Need to be able to assign leads
- ❑ Need for a central repository to store customer information
- ❑ Increase customer retention



The Internet as a Portal

- ❑ Can be used to allow employees access to Intranet applications
- ❑ Can be used to connect remote locations
- ❑ Many technologies available to implement a Web-based application



The World Wide Web

- ❑ “...Since the advent of the Web, every aspect of our lives and organizational activities has changed dramatically” (Suh, 2005)
- ❑ Vital tool for many businesses
- ❑ Web-based applications built with different technologies



Web Technologies

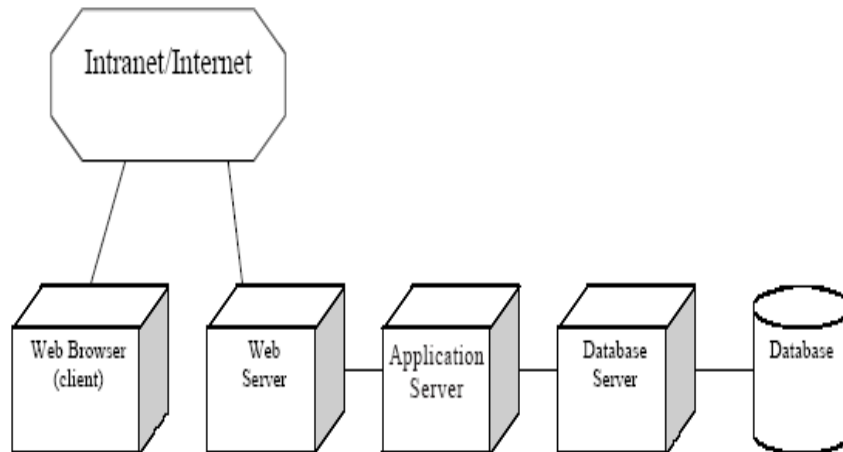
- Web applications are not just made up of static hyper-text markup language (HTML) pages but also include dynamic pages
- Dynamic pages usually written in a combination of HTML and any of the following Server-Side Languages:
 - Active Server Pages (ASP)
 - Java Server Pages (JSP)
 - PHP
 - ColdFusion



Web Technologies Cont.

- Three types of Web applications
 - Static Web documents
 - Interactive Web applications
 - Complex database backed Web-based applications

Components of a Database backed Web Application



Server-Side Language Used

- ❑ ColdFusion
 - Already in use at ARC
 - Uses HTML-like tags
 - Easy to learn
 - Can interface with Java objects, JSP pages and tag libraries
 - Can make use of custom tags written in Java or C++



Database Technologies

- Storage of data is important for most Web applications
- Relational Database Management System (RDBMS) is the suggested method of storing data
- Numerous database choices available for the data storage
 - Microsoft Access
 - Microsoft SQL Server
 - FrontBase, mySQL and PostgreSQL
 - Oracle



Database Used

- Oracle 9i
 - Very robust database
 - Already in use at ARC for their data warehouse
 - Fully licensed by ARC
 - Integrates well with ColdFusion



Web Methodologies

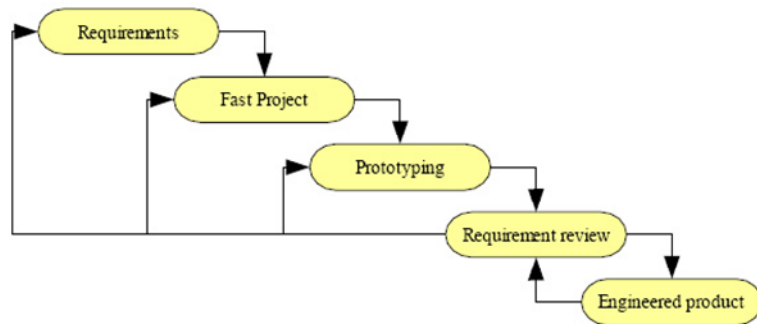
- Web application development lacks standards and structured methodologies (Chen & Heath, 2005)
- Traditional Methodologies
 - Waterfall
 - Spiral
 - Prototyping



Web Methodologies Cont.

- Methodologies targeted at Web applications
 - Intranet Design methodology
 - Howcroft & Carroll methodology
- Methodology used for the design of the Lead Management application
 - Prototyping Life Cycle

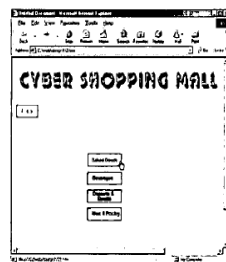
Prototyping Life Cycle



Design Considerations

- Study performed on three different menu styles

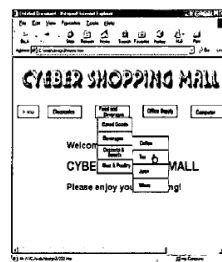
A



B



C

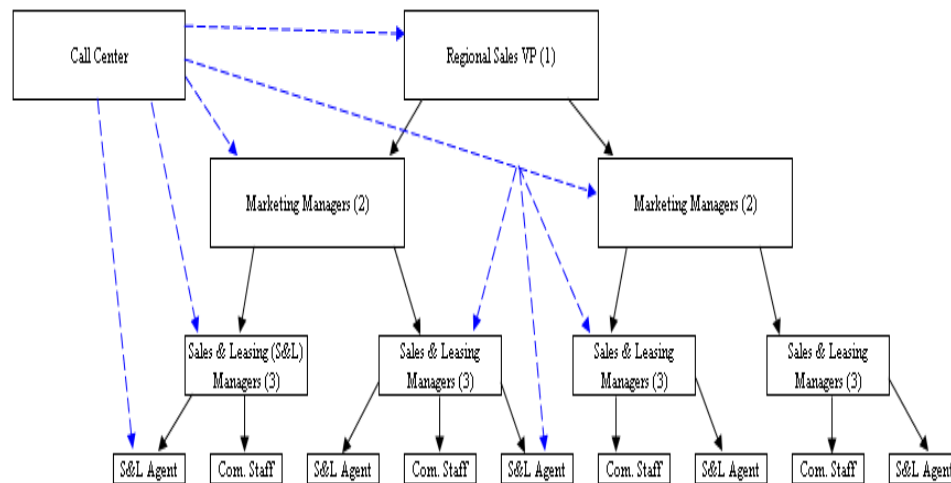


Project History

□ Application Requirements

- Accessible to the sales staff of ARC via the Internet
- Contain its own user database
- Hierarchy of roles
- Automatic or manual lead assignment
- Automatic re-assignment of *stale* leads
- E-mail notification
- XML upload of leads

Hierarchy of Roles





Application Development

- First Steps
 - Server-Side Language
 - ColdFusion
 - RDBMS
 - Oracle 9i
 - “LeadManagement” schema created
 - Sub domain of ARC’s parent domain was created for the application



Development Details

- Design of a normalized database
 - Prevent redundant data
- Programming the application
 - Design modeled after ARC’s Internet site, menu study, and visualizations provided by the owner of the project

ARC's Design Idea

Navigation Bar

HOME
LEADS
ENTER CUST.
REPORTS
SEARCH

Community Location: Housing needed within:
 Lead Origination: Date Entered:
 First Name:
 Last Name:
 Address:
 City: State: Zip Code:
 Phone Number (Home): Phone Number (Work):
 E-Mail Address:
 Best time to contact: Language Preference:
 Lead follow up/Close

Interested in:
 Lease to Own: ☐ Finance: ☐ Cash Sale: ☒
 Lease: ☐ Lot Rent: ☐

Actions:
 CASH SALE LEASE TO OWN FINANCE SALE LEASE LOT LEASE RESIDENT DATA
 SET APPOINTMENT PRINT LEAD SHEET NOT INTERESTED

Customer Notes
 10/1/2004 4:51 PM
 Customer Data Entered
 A note that the "Customer Data Entered" page is automatically generated when one of the "Action" buttons are clicked.

Add Customer Notes
 POST

Final Design

ARC
Affordable Residential Communities

Home Leads Search Profile Marketing Reporting, etc.

Enter Lead

Lead Information

Date:
 Community Location: Housing Need:
 Market:
 Lead Origination:

First Name:
 Last Name:
 Address:
 City: State: Zipcode:
 Phone Number (Home): Phone Number (Work):
 E-Mail Address:
 Best time to contact: Language Preference:

Interested in
 Bedrooms:
 Bathrooms:
 Lease to Own: ☐ Finance: ☐ Cash Sale: ☐ Lease: ☐ Lot Rent: ☐

Current Notes
 10/1/2004 4:51 PM
 Customer Data Entered
 A note that the "Customer Data Entered" page is automatically generated when one of the "Action" buttons are clicked.

Add Notes

Auto Assign Lead Manually Assign Lead Clear Form

Final Design

- Pull-down menu

Home	Leads	Search	Profile	Marketing	Reporting, etc.
	Enter Lead	Search Leads	Create Profile	Add Marketing Source	Reports
	Show Leads	Search Lead Documents	Edit Your Profile	Edit Marketing Source	Send Message
	Enter Out Of Market Lead		Edit Agents Profile		
			Delete Profile		
			Change Password		

Application Flow

- Admin will create a user profile
- User will login via login screen
- User is presented with the “Welcome” screen which contains appointments they scheduled and messages from their manager
- User can now enter a lead, edit a lead, search their leads or create sales documents for the lead



Application Features

- ❑ Automatic notification of new leads and reassigned leads to the user
- ❑ Ability to edit their own user information
- ❑ Users are restricted to the menu items they have access to



Application Features Cont.

- ❑ Users are assigned to specific geographic regions and users are only able to be assigned leads in their given region
- ❑ Ability for manager to send a message to all of their subordinates in the system
- ❑ Ability for managers to reassign subordinates leads



Results of the Application

- ❑ Increase in net resident move-ins
- ❑ Increase in occupancy
- ❑ Dramatic increase in home sales



Lessons Learned

- ❑ Using SDLC to transform a concept to a working application
- ❑ Project management skills
- ❑ Expanded knowledge of PL/SQL, ColdFusion, JavaScript, and Cascading Style Sheets
- ❑ Perseverance



Next Evolution of the Project

- ❑ Implement SSL
- ❑ Port for Blackberry and handheld devices
- ❑ Increase in performance
- ❑ Allow customers to enter their own information at a kiosk or a third party Web site



Conclusion

- ❑ Custom built application may be just as effective as an off the shelf solution
- ❑ Using ColdFusion, Oracle and the Prototyping Life Cycle a successful Web-based application was developed
- ❑ Application will fill a necessary void in the sales department and allow for a central repository of customer information



Questions and Answers

Annotated Bibliography

.NET Framework Developer's Guide (n.d.). Retrieved, January 17, 2006 from
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconaspnetplatformrequirements.asp>

This website discusses the requirements for the .NET framework. It also informs the reader on how to install and configure the .NET framework on different platforms.

Alapati, Sam R. (2003). Expert Oracle 9i Database Administration. New York, NY: Apress.

This book is a valuable resource for DBA's. It covers SQL, PL/SQL and also an introduction to UNIX. A few of the Oracle topics that are covered in this book include creating an Oracle database, performance tuning and security.

Artz, John M. (1996). A top-down methodology for building corporate Web applications. *Internet Research*, 6, 64-74.

In this paper Artz points out that corporate Web pages serve a different purpose than Web pages that are merely to provide a web presence. The paper also discusses that there are two methodologies in building web-based applications. One of these is a top-down methodology and the other is evolutionary. Artz shows that top-down development produces applications which can meet business objectives, be developed by a programming team and be maintained by people other than developers.

Beynon-Davies, Paul (2004). Database Systems. New York, NY: Palgrave Macmillan.

This book focuses on the concepts of modern database technology and the development of database systems. The book covers areas of database fundamentals, data models, database development, database management systems as well as other topics.

Boehm, Barry W. (1988). A spiral model of software development and enhancement. *Computer*, 21(5), 21-42.

This paper goes into depth about the Spiral Model of software development.

Brooks-Bilson, Rob (2003). Programming ColdFusion MX, 2nd Edition. Sebastopol, CA: O'Reilly & Associates, Inc.

This book is about ColdFusion MX. The version covered in the book is version 6. It not only talks about the history of ColdFusion but also discusses programming technique. The book also touches on databases and writing simple queries.

Buehrer, Richard E., Senecal, Sylvain, Pullins, Ellen Bolman (2005). Sales force technology usage – reasons, barriers, and support: an exploratory investigation. *Industrial Marketing Management*, 34, 389-398.

Sales force automation (SFA) technology is becoming more popular in order to increase the productivity and efficiency of salespeople. This paper outlines the reasons why some salespeople fully use or do not use the technology available to them. The paper also discusses possible barriers to SFA usage and what management can do to increase the use of such technologies.

Chen, Kuanchin, Yen, David C. (2004). Improving the quality of online presence through interactivity. *Information & Management*, 42, 217-226.

This article validates the five interactivity dimensions of Ha and James and their relationship to the quality of web design. The five dimensions are playfulness, connectedness, reciprocal communication, information collection and choice. The study found that the playfulness, connectedness and reciprocal communication are the three most important dimensions of a quality web design.

Chen, Mu-Chen, Chiu, Ai-Lun, Chang, Hsu-Hwa (2005). Mining changes in customer behavior in retail marketing. *Expert Systems with Applications*, 28, 773-781.

Marketing managers would benefit from being able to detect and predict changes in a customers behavior allowing them to have an easier time developing a long term relationship with those customers. The authors of this paper performed a study which integrates customer behavior variables, demographic variables, and transaction database in order to establish a method of mining changes in customer behavior. Their proposed approaches for mining changes in customer behavior would assist the marketing manager in developing better marketing strategies.

Crescenzi, Valter, Merialdo, Paolo, Missier, Paolo (2005). Clustering Web pages based on their structure. *Data & Knowledge Engineering*, 54, 279-299.

This paper discusses a model to describe abstract structural features of HTML pages and based on this model the authors have developed an algorithm which, when given a URL, visits a representative number of pages and produces an accurate clustering of pages based on their structure.

Fioravanti, Fabrizio (2005). *Skills for managing rapidly changing IT projects*. Hershey, PA: Idea Group Publishing.

In this book Fioravanti goes into detail about managing IT projects. One chapter in particular, chapter 8 discusses the possible methodologies one can use in order to build an application. He discusses the different phases of the Waterfall life cycle, the Spiral life cycle, and the Prototyping life cycle. He also discusses risk management for instances when a decision has to be made.

Fernández-Medina, Eduardo, Piattini, Mario (2005). Designing secure databases. *Information and Software Technology*, 47, 463-477.

The authors of this paper developed a methodology for the design of secure databases. Currently there are no database design methodologies that consider security across the entire lifecycle. The author's methodology consists of four stages: requirements gathering; database analysis; multilevel relational logical

design; and specific logical design. The first 3 stage of their methodology define activities to analyze and design a secure database and the last stage is comprised of activities that adapt the general secure data model to Oracle 9i Label Security.

Fitzgerald, Brian (1998). An empirical investigation into the adoption of systems development methodologies. *Information & Management*, 34, 317-328.

Fitzgerald conducted a survey by mail in order to determine the extent that systems development methodologies are used in practice. He also looked at the how they contribute to the development process and possible future implementation of a methodology if none currently exists. The results showed that the majority of those not using any type of systems development methodology have no plans of adopting one.

Fraternali, Piero (1999). Tools and approaches for developing data-intensive Web applications: a survey. *ACM Computing Surveys*, v31 i3, 227.

The process of building a complex web-based application is multifaceted, in that it not only involves technical questions, but also organizational, managerial, and even social and artistic issues. The author of this article goes into detail on what is involved in building a web-based application from the phases involved to the tools that can be used.

Gardner, Phillip, Maffeis, Sergio (2005). Modeling dynamic web data. *Theoretical Computer Science*, 342, 104-131.

Data on web pages is not always static. There are many ways to display dynamic data requiring a complex coordination of data and processes between sites. The authors of this article use calculus to model this coordination by integrating the XML data structure with process orchestration techniques associated with the distributed pi-calculus.

Hassan Ahmed E., Holt, Richard C. (2004). A lightweight approach for migrating Web frameworks. *Information and Software Technology*, 47, 521-532.

The article discusses the process of migrating a web application from one framework, such as Java Server Pages (JSP), to one that is entirely different, like Active Server Pages (ASP). Their process of migration preserves the original structure of the code and comments. This process would allow developers flexibility in building their application in one “language” and allowing them to move it to a possibly better “language” in the future.

Huizingh, Eelko K.R.E. (2000). The content and design of web sites: an empirical study. *Information & Management*, 37, 123-134.

This author has developed a framework to analyze and categorize the capabilities of Web sites. This framework distinguishes content from design. The content portion of the framework refers to information, features, or services that are offered on the site, whereas the design refers to the way the content is made available to the user. The framework is then applied to study how different companies use the Web for commercial purposes.

Kazman, Rick, In, Hoh Peter, Chen, Hong-Mei (2005). From requirements negotiation to software architecture decisions. *Information and Software Technology*, 47, 511-520.

The authors of this article discuss how application design and application requirements are closely related but often are performed separately, thereby hindering progress resulting in wasted effort and substantial re-working of the application. The authors look at two development methodologies and concluded that a methodology consisting of portions of both results in the most success.

Kumar, V., Shah Denish (2004). Building and sustaining profitable customer loyalty for the 21st century. *Journal of Retailing*, 80, 317-330.

The authors of this article look at past research to review important findings related to customer attitude and behavior in the context of customer loyalty. The authors also review research that relates customer loyalty to business profitability to propose a conceptual framework for building and sustaining loyalty and profitability at an individual customer level.

Lemieux, Jacques R. (1996). Using RAD tools to develop secure client/server applications. *Computers & Security*, 15, 289-295.

Lemieux shows in this paper that client/server systems can be made secure only if their transaction processing logic is moved to the server. The direct execution of SQL statements on the client can put database integrity at risk and therefore RAD applications can be secured by accessing a RDBMS through a transaction processing server.

Liao, Shu-Hsien (2005). Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert Systems with Applications*, 28, 93–103.

In this paper Liao reviews articles from 1995 – 2004 on how expert systems have developed over this period. This paper also surveys expert system methodologies using the following eleven categories: rule-based systems, knowledge-based systems, neural networks, fuzzy expert systems, object oriented methodology, case-based reasoning, system architecture, intelligent agent systems, database methodology, modeling, and ontology.

Liberty, Jesse, Hurwitz, Dan (2005). *Programming ASP.NET*. O'Reilly & Associates: Sebastopol, CA.

This book provides insight to the ASP.NET technology from Microsoft. It is meant as a guide to teach the reader what they need to know about writing Web applications and web services using C# and Visual Basic .NET.

Loshin, Paul (2001, January 8). *Relational Databases*. Retrieved January 16, 2006, from <http://www.computerworld.com/databasetopics/data/story/0,10801,55918,00.html>

This website defines a relational database. It also discusses database concepts as well as discusses E.F. Codd and how he defined the relational database concept.

McNeil, Sara G., Robin, Bernard R. (2000). Facilitating collaboration, knowledge construction and communication with Web-enabled databases. *WebNet 200 World Conference on the WWW and Internet Proceedings* 7p.

This paper presents an overview of WWW enabled databases which dynamically generate content. The paper also focuses on using this type of technology to support collaboration, knowledge construction and communication. The authors go into detail on how these database driven sites can be used in education, but can be applied to other areas.

Misra, Sanjog, Pinker, Edieal J., Shumsky, Robert A. (2004). Salesforce design with experience-based learning. *IIE Transactions*, 36, 941-953.

This paper uses data collected from the sales force of a large manufacturer and provides evidence for the link between sales person experience and sales. The findings show that worker learning can significantly reduce the effect of rising costs or decreasing margins on staffing levels.

Obrst, Leo, Howard Liu, and Robert Wray (2003). Ontologies for corporate Web applications. *AI Magazine*, v24 i3, 49-63.

The authors of this article discuss the use of ontologies in corporate websites, in particular B2B e-commerce based sites. These authors also went into detail about the ontology architecture as well as the problem of semantic/ontology mapping. They concluded that the use of ontologies is inevitable with the demand for intelligent systems to operate closer to the human conceptual level.

Ratcliffe, Martyn, Budgen, David (2005). The application of use cases in systems analysis and design specification. *Information and Software Technology*, 47, 623-641.

This paper investigates the application of use cases in the design phase of application development. The authors suggest that a state based use case model to each object of the design architecture. It concludes that state based use cases may be used to map analysis models directly into the design process.

Roh, Tae Hyup, Ahn, Cheol Kyung, Han, Ingoo (2005). The priority factor model for customer relationship management system success. *Expert Systems with Application*, 28, 641-654.

CRM systems are used as a customer relationship tool for customer satisfaction and customer retention. The authors of this paper go into detail about how decisions for implementing CRM solutions are hindered by the inconsistency between marketing strategies and information technology (IT). They used a model to determine the success of a CRM system based on CRM initiatives: process fit, customer information quality, and system support, intrinsic success: efficiency and customer satisfaction and extrinsic success: profitability.

Schewe, Klaus-Dieterr, Thalheim, Bernhard (2005). Conceptual modelling of Web information systems. *Data & Knowledge Engineering*, 54, 147-188.

In this paper the authors are focusing on the two most important layers of the abstraction layer model (ALM) for the design of a large-scale data-intensive web information system. The business layer of the ALM deals with user profiling and storyboarding. The conceptual layer addresses links to databases via extended views with the generation of navigation structures, operations supporting the activities in the storyboard, hierarchical presentations, and adaptivity to users.

Sharma, Arun, Sheth, Jagdish N. (2004). Web-based marketing: the coming revolution in marketing thought and strategy. *Journal of Business Research*, 57, 696-702.

This paper offers the suggestion that web-based markets of the future will be quite different than those of today. They also discuss the emergence of reverse marketing, customer-centric marketing, effective-efficiency, adaptation, expectation management, marketing process focus and fixed cost marketing in the context of marketing theory and practice.

Shimomura, Takao (2005). Visual design and programming for Web applications. *Journal of Visual Languages and Computing*, 16, 213-230.

This paper discusses a type of programming methodology which uses graphics as a tool of designing software. This image-oriented programming method allows users to easily develop software according to their image of what they want to develop. The paper also discusses the BioPro system which implements this method for Web-based applications. The BioPro system allows users to develop programs according to their image, verify completeness of components that make up the program, and confirm what they have develop regardless of the stage of development.

Sklar, David (2004). *Learning PHP 5*. Sebastopol, CA: O'Reilly Media, Inc..

This book's main purpose is to teach its readers how to program in PHP. It discusses everything from the concepts of PHP to working with arrays, to creating functions. It also teaches the reader how to store information into a database and how to make that database connection using PHP.

Soh, B.C., Young, S. (1998). Network system and world wide web security. *Computer Communication*, 20, 1431-1436.

The internet is regarded as the information superhighway, however, Soh and Young believe that it is lacking in its security. This paper outlines security aspects of the World Wide Web as well as network systems.

Sohn, So Young, Lee, Ji Soo (2005). Cost of ownership model for a CRM system. *Science of Computer Programming*, 60, 68-81.

This paper covers a proposed Cost of Ownership (COO) model for a CRM system. This proposed model not only covers the initial cost of the system but also the operation cost and the opportunity cost due to customer mismanagement.

Standing, Craig (2002). Methodologies for developing Web applications. *Information and Software Technology*, 44, 151-159.

Standing believes that many of the current development methodologies are inadequate for building Web-based applications. This paper examines a number of e-commerce application development approaches and proposes a methodology which attempts to address a number of issues he identified. Standing proposes the use of the Internet commerce development methodology (ICDM) because it provides a business and strategic focus as well as includes a management structure in addition to covering development aspects.

Suh, Woojong (2004). *Web engineering: principles and techniques*. Hershey, PA: Idea Group Publishing.

This book goes into detail about building applications for the WWW. While the entire book is a great resource of information the chapter on Web development methodologies is the most useful. This chapter also discusses the advantages and disadvantages of Web-based applications compared to non Web-based.

Thompson, Bob (2002). *What is CRM?* Retrieved December 29, 2005, from [http://crmguru.custhelp.com/cgi-bin/crmguru.cfg/php/enduser/std_adp.php?p_faqid=416&p_created=1018758641&p_sid=YVT1iiYh&p_lva=&p_sp=cF9zcmNoPSZwX3NvcnRfYnk9JnBfZ3JpZHNvcnQ9JnBfcm93X2NudD0xMzAyJnBfcHJvZHM9JnBfY2F0cz0mcF9wdj0mcF9jdj0mcF9zZWZyY2hfdHlwZT1hbnN3ZXJzLnNIYXJjaF9ubCZwX3BhZ2U9MQ&p_li=&p_topview=1](http://crmguru.custhelp.com/cgi-bin/crmguru.cfg/php/enduser/std_adp.php?p_faqid=416&p_created=1018758641&p_sid=YVT1iiYh&p_lva=&p_sp=cF9zcmNoPSZwX3NvcnRfYnk9JnBfZ3JpZHNvcnQ9JnBfcm93X2NudD0xMzAyJnBfcHJvZHM9JnBfY2F0cz0mcF9wdj0mcF9jdj0mcF9zZWZyY2hfdHlwZT1hbnN3ZXJzLnNIYXJjaF9ubCZwX3BhZ2U9MQ**&p_li=&p_topview=1)**

This website provides a definition of Customer Relationship Management (CRM). It also discusses the customer relationship lifecycle and the business strategy of CRM.

Valderrama, Rubén Peredo, Ocaña, Leandro Balladares , Sheremetov, Leonid B. (2005). Development of intelligent reusable learning objects for web-based education systems. *Expert Systems with Applications*, 28, 273-283.

This paper goes into depth about Web-Based education and how there is a need for durable, interoperable and reusable objects in these systems. The concept of reusable objects does not only apply to Web-Based Education systems but can be used in other Web-Based applications.

Wang, Hongbing, Huang, Joshua Zheexue, Qu, Yuzhong, Xie, Junyuan (2004). Web services: problems and future directions. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1, 309-320.

The authors of this paper believe that the current Web services architectures are confronted with problems. One major flaw in Web services is the notion of security. The authors believe that the resolution of the problems with Web services is crucial to their success. Web services are based on existing internet

protocols and open standards, and can provide assistance with application integration.

***What is SQL Server 2005?* (n.d.). Retrieved December 29, 2005, from**

<http://www.microsoft.com/sql/prodinfo/overview/what-is-sql-server.msp>

This site briefly discusses what Microsoft SQL Server is and the requirements that are needed for it to run. The site also illustrates the core components of SQL Server.

Westman, Stephan (2002). Building database-backed Web applications: process and issues. *Information Technology and Libraries*, 21 i2, 63-64.

This article discusses the development of a database backed web application from concept to implementation. Development tools, RDBMS systems and hardware platforms are also discussed. Regardless of the size of the project the article discusses the need for careful planning and distribution of duties among those involved.

Yu, Byeong-Min, Seak-Zoon Roh, and Sungwook Han (2001). The effect of different menu styles on the user's perception and performance on the WWW. *Annual Topics on Distance Learning Conference*.

This paper was presented at the Annual Topics on Distance Learning Conference June 5-6, 2001. The authors discuss how different menu styles can help or hinder a user's ability to efficiently and effectively navigate a website. Through the experiment on undergraduate and graduate students they determined that a pull-down style menu provided the most ease in navigating their experimental website.

Regis University
School for Professional Studies Graduate Programs
Professional Project

Certification of Authorship of Professional Project Work
--

Print Student's Name James M. Darschewski

Telephone 303.886.9691 Email JAMES.DARSCHEWSKI@GMAIL.COM

Date of Submission 04/30/2006 Degree Program MSCIT

Title of Submission Design and Implementation of a Web-Based Lead Management System Using Oracle and ColdFusion Technologies

Advisor/Faculty Name Sam Conn

Certification of Authorship:

I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for the purpose of partial fulfillment of requirements for the MSCIT Degree Program.



Student Signature

05/02/2006

Date

Regis University
School for Professional Studies Graduate Programs
MSCIT Program

Graduate Programs Final Project/Thesis

Advisor/Professional Project Faculty Approval Form

Student's Name: James M. Darschewski Program MSCIT
PLEASE PRINT

Professional Project Title: Design and Implementation of a Web-Based Lead
Management System Using Oracle and ColdFusion Technologies

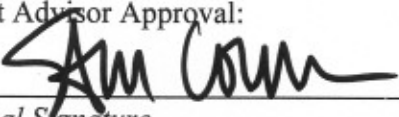
Advisor Name Sam Conn
PLEASE PRINT

Project Faculty Name Joseph Gerber
PLEASE PRINT

Advisor/Faculty Declaration:

I have advised this student through the Professional Project Process and approve of the final document as acceptable to be submitted as fulfillment of partial completion of requirements for the MSCIT Degree Program.

Project Advisor Approval:


Original Signature

5-12-2006
Date

Degree Chair Approval if:

The student has received project approval from Faculty and has followed due process in the completion of the project and subsequent documentation.


Original Degree Chair/Designee Signature

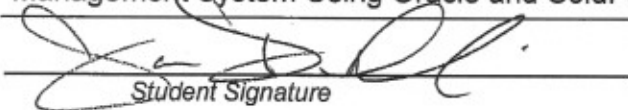
5/30/06
Date

REGIS UNIVERSITY
School for Professional Studies
Graduate Programs Final Project/Thesis

Authorization to Publish Student Work

I, James M. Darschewski, the undersigned student,
Print student name
in the MSCIT Degree Program hereby authorize Regis University to publish through
Degree Program
a Regis University owned and maintained web server, the document described below ("Work"). I
acknowledge and understand that the Work will be freely available to all users of the World Wide
Web under the condition that it can only be used for legitimate, non-commercial academic research
and study. I understand that this restriction on use will be contained in a header note on the Regis
University web site but will not be otherwise policed or enforced. I understand and acknowledge that
under the Family Educational Rights and Privacy Act I have no obligation to release the Work to any
party for any purpose. I am authorizing the release of the Work as a voluntary act without any
coercion or restraint. On behalf of myself, my heirs, personal representatives and beneficiaries, I do
hereby release Regis University, its officers, employees and agents from any claims, causes, causes
of action, law suits, claims for injury, defamation, or other damage to me or my family arising out of or
resulting from good faith compliance with the provisions of this authorization. This authorization shall
be valid and in force until rescinded in writing.

Print Title of Document(s) to be published: Design and Implementation of a Web-Based Lead
Management System Using Oracle and ColdFusion Technologies


Student Signature

05/02/2006
Date

Check if applicable:

☒ The Work contains private or proprietary information of the following parties and their attached permission
is required as well: Affordable Residential Communities

Complete if you do not wish to publish your work on the WWW:

☐ I do not authorize Regis University to publish my work on the WWW.

Student Signature

Date

REGIS UNIVERSITY

SCHOOL FOR PROFESSIONAL STUDIES GRADUATE PROGRAMS FINAL PROJECT


Releasor Authorization to Publish Student Work WWW

I, Michael Brakke the undersigned, Vice President of Information Technology
Print Name of Company/Organization Representative Representative's Title

on behalf of Affordable Residential Communities ("Releasor") do hereby authorize
Company/Organization Name
Regis University to publish through a Regis University owned and maintained web server,
the document described below ("Work") and acknowledges that the Work contains personal
or proprietary information of the Releasor. Releasor further acknowledges and understands
that the Work will be freely available to all users of the World Wide Web under the condition
that it can only be used for legitimate, non-commercial academic research and study but that
this restriction on use will be contained in a header note on the Regis University web site but
will not otherwise be policed or enforced. This authorization shall be valid and in force until
rescinded in writing.

Print Student Name: James M. Darschewski

Title(s) of document(s) to be published: Design and Implementation of a Web-Based Lead
Management System Using Oracle and ColdFusion Technologies

BY:  DATE: 05/02/2006
Company/Organization Releasor Signature

Note: It is the student's responsibility to obtain the necessary release(s) prior to submitting the Final Project for publication. Please print your name and list all applicable documents.