Smith and Schaefer’s Bid Tracking System

By

James R. Strayhorn Jr. (J.R.)

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

University of Cincinnati
College of Applied Science

June 2006
Smith and Schaefer’s Bid Tracking System

by

James R. Strayhorn Jr. (J.R.)

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

© Copyright 2006 James R. Strayhorn Jr.

The author grants to the Information Technology Program permission to reproduce and
distribute copies of this document in whole or in part.

________________________________________________________________________
James R. Strayhorn Jr.                                      Date

________________________________________________________________________
Dr. Hazem Said, Faculty Advisor                             Date

________________________________________________________________________
Patrick C. Kumpf, Ed.D. Interim Department Head            Date
Acknowledgements

I would like to give special thanks to James Strayhorn, estimating director of Smith and Schaefer and my dad, for allowing me to take on this project and believing in me to get this solution finished. I would also like to give special thanks to Smith and Schaefer, for allowing me to use their information and logos during the design of the Smith and Schaefer Bid Tracking System. Finally, I give special thanks to my family and friends who have supported me throughout the whole Senior Design process and Professor Hazem Said for continuing to push and motivate me to do my best.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgements</td>
<td>i</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>ii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>iv</td>
</tr>
<tr>
<td>Abstract</td>
<td>vi</td>
</tr>
<tr>
<td>1. Statement of the Problem</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction to Smith and Schaefer</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Description of Current Bid Tracking Process</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Definition of the Need</td>
<td>3</td>
</tr>
<tr>
<td>2. Description of the Solution</td>
<td>4</td>
</tr>
<tr>
<td>2.1 User Profile</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Design Protocols</td>
<td>6</td>
</tr>
<tr>
<td>3. Deliverables</td>
<td>11</td>
</tr>
<tr>
<td>4. Design and Development</td>
<td>12</td>
</tr>
<tr>
<td>4.1 Budget</td>
<td>12</td>
</tr>
<tr>
<td>4.2 Timeline</td>
<td>12</td>
</tr>
<tr>
<td>4.3 Hardware and Software Details</td>
<td>13</td>
</tr>
<tr>
<td>4.4 Testing Plan</td>
<td>14</td>
</tr>
<tr>
<td>5. Proof of Design</td>
<td>14</td>
</tr>
<tr>
<td>5.1 Initial Screens</td>
<td>14</td>
</tr>
<tr>
<td>5.1.1 Shortcut Page</td>
<td>14</td>
</tr>
<tr>
<td>5.1.2 User Login</td>
<td>15</td>
</tr>
<tr>
<td>5.2 Estimating Director (Admin) Console</td>
<td>16</td>
</tr>
<tr>
<td>5.2.1 Admin Console Main Page</td>
<td>16</td>
</tr>
<tr>
<td>5.2.2 View Bid Requests</td>
<td>16</td>
</tr>
<tr>
<td>5.2.2.1 View Bid Request Details</td>
<td>17</td>
</tr>
<tr>
<td>5.2.3 Manage Sales Reps / Estimators</td>
<td>18</td>
</tr>
<tr>
<td>5.2.4 Manage User Accounts</td>
<td>19</td>
</tr>
<tr>
<td>5.2.5 Quick Reports and Crystal Reports</td>
<td>20</td>
</tr>
<tr>
<td>5.3 Sales Rep Console</td>
<td>21</td>
</tr>
<tr>
<td>5.3.1 Sales Rep Console Main Page</td>
<td>21</td>
</tr>
<tr>
<td>5.3.2 Manage Bid Requests</td>
<td>22</td>
</tr>
<tr>
<td>5.3.2.1 View Bid Request</td>
<td>22</td>
</tr>
<tr>
<td>5.3.2.2 Edit Bid Request</td>
<td>23</td>
</tr>
<tr>
<td>5.3.3 Add a New Bid Request</td>
<td>24</td>
</tr>
<tr>
<td>5.4 Estimator Console</td>
<td>24</td>
</tr>
<tr>
<td>5.4.1 Estimator Console Main Page</td>
<td>24</td>
</tr>
<tr>
<td>5.4.2 View Assigned Bids</td>
<td>25</td>
</tr>
<tr>
<td>5.4.3 Export Bid Info to CSV</td>
<td>26</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Bid Tracking System Use Case Diagram</td>
<td>7</td>
</tr>
<tr>
<td>Figure 2, 3</td>
<td>Hardware and Software Budget for Bid Tracking System</td>
<td>12</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Shortcut Page</td>
<td>15</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Login Page</td>
<td>16</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Admin Console Main Page Screen</td>
<td>16</td>
</tr>
<tr>
<td>Figure 7</td>
<td>View Submitted Bids Screen</td>
<td>17</td>
</tr>
<tr>
<td>Figure 8</td>
<td>View Bid Request Details Screen</td>
<td>18</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Manage Sales Reps Screen</td>
<td>18</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Add a New Sales Reps Screen</td>
<td>19</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Edit a Sales Rep Screen</td>
<td>19</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Manage User Accounts Screen</td>
<td>20</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Edit User Account Screen</td>
<td>20</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Quick Report Screen</td>
<td>21</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Crystal Report Screen</td>
<td>21</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Sales Rep Console Main Page Screen</td>
<td>22</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Manage Bid Request Screen</td>
<td>22</td>
</tr>
<tr>
<td>Figure 18</td>
<td>View Bid Request Details Screen</td>
<td>23</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Edit Bid Request Screen</td>
<td>23</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Add a New Bid Request Screen</td>
<td>24</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Estimator Console Main Page Screen</td>
<td>25</td>
</tr>
<tr>
<td>Figure 22</td>
<td>View Assigned Bids Screen</td>
<td>25</td>
</tr>
</tbody>
</table>
Figure 23. View Assigned Bid Request Details Screen 26
Figure 24. CSV Export Dialog Box 26
Figure 25. Exported Bid Request Info in Excel 27
Abstract

Smith and Schaefer’s Bid Tracking System is a web application that will provide Smith and Schaefer with a powerful data warehousing system that will allow users to access bid request information online, create reports automatically and help to eliminate their redundant and inefficient paper-based bid tracking process. This application is coded in VB.NET, utilizing ASP.NET and MS Access technologies. Smith and Schaefer’s Bid Tracking System will have three levels of users who benefit from this system; the Estimating Director, Sales Representatives and Estimators. The Estimating Director will use this system to oversee the entire bid tracking process from a single “admin” console. Sales Representatives will use this system to initiate the bid tracking process by creating bid requests. Estimators will use this system to view and retrieve any assigned bid requests.
Smith and Schaefer’s Bid Tracking System

1. Statement of the Problem

1.1 Introduction to Smith and Schaefer

Smith & Schaefer, Inc. is a regional sales company specializing in furnishings and equipment for the laboratory, healthcare facility, and educational institutions. They have been doing business for more than 50 years. Their main areas of business are Ohio, Kentucky, Indiana and Western Pennsylvania. Their corporate office is in Cincinnati, Ohio. They maintain a full service branch office in Lakewood, Ohio (Cleveland) and have resident sales personnel in Columbus, Toledo, Louisville, Indianapolis, Pittsburgh and State College, Pennsylvania. (1)

Smith & Schaefer, Inc.’s mission is to be dedicated to the sale and servicing of capital goods and products. Along with this mission, Smith & Schaefer also support their selling efforts with complete pre-sales planning and post-sale installation and warranty services. For the past 50 years, Smith & Schaefer has been providing complete floor plan design services for laboratories in the healthcare, educational and industrial research markets. Their design engineers work closely with laboratory personnel, architects and construction contractors to create professional design drawings for each phase of new or renovation construction projects. (1)

In other words, Smith & Schaefer, Inc. is a one-stop shop for laboratory, healthcare and educational equipment. They work with many suppliers from Fisher Hamilton (metal, wood and plastic laminate casework) to Consolidated Stills & Sterilizers (steam sterilizers and water stills) to provide quality products to their laboratory, healthcare and educational clients. (1)
1.2 Description of Current Bid Tracking Process

Smith & Schaefer accomplish their sales currently through 10 sales reps that are scattered across their Midwestern offices. These sales reps meet with their markets (laboratory, healthcare and educational institutions) and complete bids/proposals for their clients against their competition. They routinely complete around 260 to 300 bids a year totaling on average **$30 millions dollars in revenue.** This dollar amount in sales revenue is impressive but their current way of tracking this revenue information is most certainly not.

The company is manually capturing sales information as follows: sales rep would fill out a bid/quotation request form (OOR) and fax this form to the Estimating Director at the main corporate office in Cincinnati. The Estimating Director would then view the bid request form, add any additional info and assign the “estimating” of the bid to either one of the employees in the Estimating Department or to himself. From there, the assignee in the estimating department would manually copy the information from the faxed form and input the information into excel or into “Labworks” (“Labworks” is a 3rd party tools from one of their suppliers that they use to calculate price estimates). (2)

The estimating department would then run calculations and produce estimated pricing either in excel or “Labworks”. They would then send this information in the form of a scope letter (summarizing the bid) and price breakdown (gives details of how price estimate was calculated) to the Sales Rep. From there, the Sales Rep would then call the project lead for whom the bid was created and get the results of the bid rather win / loss. If the results are not updated, the bid should still be given the status of pending. The Estimating Director can also receive bid status updates and therefore update bid requests.
The bid outcome (win or loss) is sent back to the Estimating Director. The Estimating Director would then enter the post-bid information into a spreadsheet and use this spreadsheet to generate reports. (Total bids, win/loss %, dollar amounts, etc)

1.3 Definition of the Need

There are some major flaws with the current way that they are processing information. First and foremost there is no real good communications between sales reps and the estimating department. Once the estimating department receives the bid quote, the sales reps are out of the loop on any updates happening to the bid such as wins, losses, or changes to the initial quote. Secondly, there are bound to be human mistakes such as messy handwriting on the faxed forms or incorrect calculation of numbers associated with the bid quote. Thirdly, after the bid has been won or loss the estimating department would then by hand create a yearly bid summary to total millions of dollars worth of revenue. This amount of data should not be counted merely by hand due to errors.

Smith and Schaefer soon realized after years of manually creating bid tracking summaries that a more modern and automatic system needed to be implemented. The company is now interested in making this manual process paperless and accessible through the Web. In addition, they would like for the application to do the following:
1) Track the win/loss/pending status of bids
2) Track the name of project, location of project, sales rep of the bid, amount of the bid, profit from the bid and other information as outlined in the company’s OOR form for a particular bid.
3) Track bid results such as total profit and manufacturer (Estimating Director)
4) Create sales rep. profiles for sales reps in the Cincinnati and Cleveland offices (these profiles would give specific sales rep information as outlined by the Estimating Directory)

5) Give breakdown of sales activity by sales reps by location over a certain time frame

6) Export data to excel for charts and graphs

7) Have login capabilities for Sales Reps and Estimating Department

2. Description of the Solution

The Smith and Schaefer Bid Tracking System is a Web based application, utilizing ASP.NET technology and MS Access database technology to provide a powerful data warehousing application that will allow for Smith and Schaefer to more readily make their bid information available via online reporting and paperless workflows.

**Centralized area for tracking of bids (Data Warehousing)**

As outlined in the project description, Smith and Schaefer gain most of their revenue through the gaining and winning of bids. This application will allow the company to have a centralized location where their entire bid related activities will take place, such as the updating of bid status and initial assigning of bids to estimators. With this centralizing of bid data, it will be even easier to access information needed to make key business decisions.

**Internet available for accessibility (Web Application)**

Everyone involved in the bidding process will be able to login to the application to their individual jobs when it comes to bidding. Sales reps will be able to add, update and delete bids. Estimators will be able to access bid information online. The Estimating
Director will be able to create meaningful metric reports to manage the bidding process more effectively.

*Elimination of their current paper workflow (Completely Paperless)*

The beauty of the Bid Tracking System is in the fact that this application will eliminate the possibility of data loss through missed communication and data error through human processing. For the amount of revenue that is produced through their bids, the elimination of data corruption is a primary concern.

*Ability to create reports (Robust Reporting)*

With the data warehousing capabilities of this application, Smith and Schaefer’s management teams can now create reports quickly and effectively to provide metrics on their bid tracking activities such as reporting on win / loss / pending status of bids, overall bid results such as total profit, percentage of loss by dollars, etc.

2.1 User Profile

Smith and Schaefer expect to have three user-roles for their Bid Tracking System.

(3)

*Sales Representative (Sales Rep)*

The Sales Rep will talk with potential clients and project owners. Once a partnership has been made, a Bid Request will be submitted through the Bid Tracking System to the Estimating Director. The Sales Rep will then wait for a final status from the Estimating Department. Based on the final bid status (Won or Loss), the Sales Rep would follow up with the client or project owners accordingly.
Estimating Director (Admin)

The Estimating Director will look over the bids that have been submitted by the Sales Rep. After making sure that the Bid Request form has been submitted and completed properly, the Estimating Director will assign the Bid Request to an estimator. The Estimating Director will then wait for a Labworks file that will have the price quote and supplement information. The Estimating Director will send this information to the client and project owner. The Estimating Director would then receive a final bid status either Won or Loss. After receiving this final bid status, the Estimating Director would update the Bid Request with this information.

Estimator

The Estimator would retrieve the bid information for the Bid Requests that have been assigned to them. The Estimator would then take the Bid information and export an excel file from this information. After downloading this excel document, the Estimator would then import it into Labworks. Once the import and calculations have been done by Labworks, then a Labworks file that contains all of the pricing and supplement information would be sent to the Estimating Director.

2.2 Design Protocols

Organizational Scheme and Role-Based Security Logins

Smith and Schaefer’s Bid Tracking System is divided into three sections based on the three user roles explained earlier. The user interface is Web-based in nature. In order to use the system, each user will have to login with a username and password. The Estimating Director will be given a system password that he can use to log onto the system. On a new install of the Bid Tracking System, there will be one Admin user but
no Sales Rep or Estimator users. The Admin will have to create at least one Sales Rep and one Estimator user for the Bid Tracking Process to take place. Once logged in, the user will have access to a variety of functions based on the role that they have been granted by the Admin. The functionality of each user is outlined by Figure 1. The use case diagram shows each user and the functions that they will be able to access. Having role based security via a unique username and password combination was a much needed security enhancement that will ensure the safety of the bid information.

Figure 1. Bid Tracking System Use Case Diagram
**Database and Class Design**

In the Smith and Schaefer Bid Tracking System, there are a variety of tables that hold all of the important bid request information. The main focus of this database is the Bids table: linked to it via a Foreign Key are the Dealers, Projects, Companies, Employees, Architects, Manufacturers, Orders and WorkTopMaterials tables. The database design is what brings the whole application together. In order to get all of the Bid information, there will be various queries that the application will run in the background. The whole purpose of the application is for the process to be paperless. With this database design in place, simple queries can now be run to create any custom report that the Estimating Director has in mind.

In terms of the Class Design, essentially the Database Design and the Class Design (See Figure B-1, Appendix B) are one in the same. The reasoning behind that is so each database table can be turned into an object.

**3-Tier Object Oriented Application Development**

This application was built with 3 specific tiers: Data Access Layer (DAL), Business Logic Layer (BLL) and a Graphical User Interface Layer (GUI). Each tier breaks down as follows:

- **DAL** – controls all reading, writing and deleting from the Database (For more information on DAL, see Appendix C – Data Access Layer Code Snippet)
- **BLL** – houses all the business rules for each object with the application. (For more information on BLL, see Appendix C – Business Logic Layer Code Snippet).
• GUI – the actual design of the pages, Web forms and report layouts. (ASP.NET and HTML pages)

The beauty of 3-Tier application development is the separation that each layer has from one another. For example, if I have a form that takes submitted bid request information and updates this into the database, I would create a new Bid object

```dim
aNewBid as New Bid,
```

populate its data then call `DAL.CreateNewBid(aNewBid)`.

The GUI is calling a function of the DAL using a BLL object as a parameter. The beauty of this is two fold. First, it allows me to do a very complex database task in two commands and secondly, if I wanted to change how many values are within a Bid, all I’d have to do is change the Bid BLL object. I wouldn’t have to touch the `DAL.CreateNewBid` function because the function knows to use a Bid object and I don’t have to update the GUI. This is a very nice feature of 3-Tier Object Oriented Application Development.

**Color Scheme**

There is a simple color scheme to help users distinguish which area they are in. The Admin area is red. The Sales Rep area is blue. The Estimator area is green. I kept the graphical elements about the same for each area in terms of font type and font sizes since the color code allows users to know which area they are in just by looking at the page.

**Bid Tracking System User Interface**

As stated earlier, only certain users can access certain functionality. Below is an outline of what each user’s main links are and a brief description of each.

The Sales Rep’s main links are as follows:
• **Manage Your Bid Request:** Sales Reps can view the bids that they have submitted. From this view they can either delete their bid request or edit the existing bid request. Here they can also view the status of their bids and which Estimator the bids have been assigned to.

• **Add a New Bid Request:** Sales Reps can create a brand new bid request here.

• **Logoff:** This will log the Sales Rep out of their session and return them to the shortcut start up page.

The estimating director’s main links are as follows:

• **View Bid Requests:** The estimating director can view the bids that have been submitted. There is a dropdown box that will allow the estimating director to view any created bid based on current bid status (New, Pending, Won or Loss). The estimating director will also be able to change bid status and which estimator the bid request has been assigned to.

• **Manage Sales Rep/Estimator:** From here the estimating director will be able to add, update or delete a Sales Rep or Estimator user (respectively).

• **Manage User Account:** From here the estimating director will be able to update username and password information for any user.

• **Logoff:** This will log the estimating director of their session and return them to the shortcut start up page.

The estimator’s main links are as follows:

• **View Assigned Bid Requests:** The estimator can view the bids that have been assigned to them. From here the estimator can view the bid and export the information within the bid into an excel file.
• **Logoff:** This will log the estimator out of their session and return them to the shortcut start up page.

3. **Deliverables**

During the design phase of this project the following deliverables were defined:

• Form Based Authentication for Login and Logout – All Actors
• Add / Update / Delete Sales Rep User – Estimating Director
• Add / Update / Delete Estimator User – Estimating Director
• Add / Update / Delete Bids – Sales Reps
• Update / Delete Bids – Estimating Director
• View Created Bid Requests – Sales Reps
• View Bid Requests By Status – Estimating Director
• View Assigned Bid Requests – Estimator
• Assign Bid to Estimator – Estimating Director
• Update Bid Status – Estimating Director
• Export Assigned Bid Info to CSV – Estimating Director and Estimator
• Quick Report Views – Estimating Director
• Create and View Crystal Repots – Estimating Director
• Auto generate email notifications – Bid Tracking System
• User friendly GUI designs
• Create consistent look and feel through CSS
• Web Application done with 3 Tier Architecture
4. Design and Development

The next section describes the project’s budget, timeline, and hardware and software details.

4.1 Budget

Figure 2, 3 is the budget for the Bid Tracking System.

<table>
<thead>
<tr>
<th>Item</th>
<th>Explanation</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Software</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Macromedia Studio MX</td>
<td>Licensing agreement</td>
<td>$257</td>
</tr>
<tr>
<td>Adobe Systems</td>
<td>Licensing agreement</td>
<td>$202</td>
</tr>
<tr>
<td>(Include Photoshop,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>InDesign and ImageReady)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visual Studio .NET (for</td>
<td>Licensing agreement</td>
<td>$425</td>
</tr>
<tr>
<td>ASP.NET and VB.NET)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MS Access</td>
<td>Own – included with their MS Office License</td>
<td>Free</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Hardware</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purchasing a new server</td>
<td>N/A – only if a STRONG business need arises</td>
<td>Unknown as time of this proposal</td>
</tr>
</tbody>
</table>

Figure 2, 3. Hardware and Software Budget for Bid Tracking System

As you can see, the true cost to Smith and Schaefer for this project is $884. (4)

4.2 Timeline

Over the past year, the timeline for the Bid Tracking System has changed from time to time. The most serious portions of development time were spent in Technical Practicum I last quarter and Technical Practicum II this quarter. I spent the majority of my time working on this project on weekdays and weekends as my school work allowed. The majority of time in Technical Practicum I consisted of me coding the actual 3 tiers of
this system. The breakdown of how my time was spent in Technical Practicum II is shown in Figure B-2 in Appendix B.

4.3 Hardware and Software Details

Since I went with ASP.NET using VB.NET for the front end of the Bid Tracking System, I had to ensure that project resources would be available. First and foremost, Smith and Schaefer is a Windows shop, meaning that a lot of their current implementation of server technology and applications are in Windows. They are currently running Windows Server 2000. We will begin to check to make sure that everything is set up in house to host ASP.NET or if they will continue to go with their current web hosting company. We will contact their hosting company to see if they can host ASP.NET for an affordable price. Right now I believe that their current web hosting company is charging them close to $50 a month. This is a large amount of money. The web hosting will be a key area of cost savings for our project.

With the choice of ASP.NET for our front-end, I’ve gone with MS Access for the back end of the Bid Tracking system. I believe that MS Access is robust enough to serve as the database for the Bid Tracking application and there will be plenty of support from Microsoft on implementation. (5)

I’ve worked with Smith and Schaefer on the resources to also ensure that they will be available. Most of the actual development will not be done in house at Smith and Schaefer but on my machine at home. I’ve meet with Smith and Schaefer off and on for the last year to discuss the project and to make sure that I’m going in the right direction in terms of development. I’ve been using an iterative design process; there has been heavy
prototyping and constant feedback from Smith and Schaefer. Due to the strong backing of Smith and Schaefer, I’ve been able to successfully complete this project.

4.4 Testing Plan

Due to time restraints, the development of the Smith and Schaefer Bid Tracking System has followed an iterative process. This means that as I’ve been developing various modules and functions within the system, I’ve been testing them. In my opinion, I feel that this is the best option available because I can test the many database queries and functions right then and there while I am developing. Waiting until later in the process may cause other functions not to work or crash the application all together. I feel that doing on-going unit testing has allowed me to create a very functional application fairly quickly with minor headaches along the way. As the Senior Design process continued, unit testing was a major form of testing for me. As I was getting closer to a more stable version of the Bid Tracking System, I allowed for user testing focusing primarily on how multiple users can interact with the system and the database backend. This was also shown during the final Senior Design presentation.

5. Proof of Design

The next section shows in detail how deliverables of the project were fulfilled.

5.1 Initial Screens

This section will outline the initial screens that users will first encounter before going into their specific application consoles.

5.1.1 Shortcut Page

The first page that users see is a page with three graphical images representing the three different areas of the web site; Admin Console, Sales Rep Console and the
Estimator console. The users will choose the console that they would like to log into. Ideally these links to the console areas can be placed anywhere within Smith and Schaefer’s intranet it will point them to the correct areas. For simplicity purposes they are all on the same page. (See Figure 4)

![Shortcut Page](image)

**Figure 4. Shortcut Page**

### 5.1.2 User Login

Since form based authentication is being used, users will need to login in order to use the system via the login page (See Figure 5, Page 16) Not only will a user’s username and password have to be correct, they must also have the appropriate role in order to log into the various consoles. There are three user roles; Admin, Estimator, SalesRep. These three roles correspond to the three user profiles for this application. (For more information on the user login, see Appendix A – Login Code Snippet.)
5.2 Estimating Director (Admin) Console

This section will outline the functionality contained within the Admin console of the Bid Tracking System.

5.2.1 Admin Console Main Page

The first page that the Estimating Director will see once they have successfully logged in is the Admin Console Main Page. From here the Estimating Director can access all of the functions necessary to administer the Bid Tracking Process. (Figure 6)
This area of the Admin console will allow the Estimating Director to view all of the Bid Requests that have submitted to the system. Using the drop down, he can select to see Bids based on the Bid Status (Won, Loss, Pending, New). Once the Bid Status that he would like to see has been select. The Estimating Director would then click the “Change View” button to show the Bid Requests that match the criteria selected. (See Figure 7)

![Figure 7. View Submitted Bids Screen](image)

5.2.2.1 View Bid Request Details

Once the Estimating Director has selected a bid to view the View Bid Request Detail page will be displayed. From this screen, the Estimating Director can review and update any Bid Request information. He can also change the Bid Status and assign the Bid Request to an Estimator for further processing. The majority of the time the Estimating Director will be in this area. (Figure 8, Page 18)
5.2.3 Manage Sales Reps / Estimators

To add, edit or delete a sales rep or estimator user, the Estimating Director will enter the Manage Sales Reps / Estimators area. (For more information on code used here, See Appendix C – Add Estimator Code Snippet.) From here, the Estimating Director can change any information pertaining to a Sales Rep or Estimator. (See Figures 9, 10, 11)
5.2.4 Manage User Accounts

To change user account information such as username and/or password, the Estimating Director will enter the Manage User Accounts area. (See Figures 12, 13, Page 20)
5.2.5 Quick Reports and Crystal Reports

Another key area of the Bid Tracking System is the ability for the Estimating Director to create reports on the Bid Tracking process that can be used as metrics. The Estimating Director has two ways of doing this. 1) He can view “Quick Reports” which will give him the ability to see a quick graph of key bid request information such as Total Number of Bids by Bid Status or Assigned Estimator. 2) He can select “View Reports”
and select a report. This will give me a rich Crystal Report that will allow for him to drill down the report on certain information like Bid Status or Sales Rep. (See Figures 14, 15)

![Figure 14. Quick Report Screen](image1)

![Figure 15. Crystal Report Screen](image2)

### 5.3 Sales Rep Console

This section will outline the functionality contained within the Sales Rep console of the Bid Tracking System.

#### 5.3.1 Sales Rep Console Main Page
The first page that a Sales Rep will see once they have successfully logged in is the Sales Rep Console Main Page. From here a Sales Rep can access all of the functions necessary to submit Bid Requests. (Figure 16)

![Figure 16. Sales Rep Console Main Page Screen](image)

### 5.3.2 Manage Bid Requests

From the Manage Bid Requests area, Sales Reps can view or edit any of the Bid Requests that they have created. (Figure 17)

![Figure 17. Manage Bid Request Screen](image)
5.3.2.1 View Bid Request

Once a Bid Request has been select, a Sales Rep can view any changes that may have been made by the Estimating Director especially the Bid Status and Assigned Estimator which cannot be changed by the Sales Rep. (Figure 18)

Figure 18. View Bid Request Details Screen

5.3.2.2 Edit Bid Request

From the same screen where the Sales Rep views their Bid Request details, they can also edit the information by selecting to edit the Bid Request information. (Figure 19)
5.3.3 Add a New Bid Request

Very simply put, this area will allow a Sales Rep to add a new Bid Request to the system. (Figure 20)

5.4 Estimator Console

This section will outline the functionality contained within the Estimator console of the Bid Tracking System.

5.4.1 Estimator Console Main Page

The first page that an Estimator will see once they have successfully logged in is the Estimator Console Main Page. From here an Estimator can access all of the functions necessary to process and estimate a Bid Request. (Figure 21, Page 25)
5.4.2 View Assigned Bids

From this screen, an Estimator can view all of the Bid Request information. Estimator cannot change Bid Request information. They can only view it. (Figure 22, 23)
5.4.2 Export Bid Information to CSV

Click the Export button will allow for an Estimator to create an Excel document with all the Bid Request information inside of it. This information can then be used to upload into Labworks. (Figure 24, 25)
6. Conclusions and Recommendations

This section will outline my conclusions and recommendations for the Smith and Schaefer Bid Tracking System project.

6.1 Conclusions

This project was created in response to a business need address by Smith and Schaefer’s Estimating Department. I’ve created a powerful Web application that will solve a real world business problem. This project was designed in an object-oriented manner so that it can be easily maintained and expanded upon. I am happy that I was able to complete this project and have it perform well beyond my expectations. Smith and Schaefer, Inc. is interested in the project, and I will be working to get it ready for deployment. I am happy to report that this project won Best of Tech Expo 2006 for Software Application Development. I am proud of this accomplishment and feel it validates the hard work that I put into this project.
6.2 Recommendations

While working on this project, I encountered many challenges and obstacles that I needed to overcome. Solid understandings of application development best practices are essential to have before beginning to create a Web application of this magnitude.

Initially, I spent the majority of my time in the beginning on the design and analysis of this system, especially the underlying architecture and tier separations. Once this was complete, the actual coding of the application was not difficult. I would also suggest having a very good handle on the language of preference chosen to develop in: VB.NET, C#.NET, Java, etc.

I am happy with the results of the project as it stands right now, but for deployment purposes there are a number of enhancements and changes that I feel must be made. Here is a listing of the enhancements that will need to be made:

- An upgrade of IDE from Visual Studio .NET 2003 to Visual Studio .NET 2005 would have to be made in order to stay up to date and current with the available technology.

- An upgrade of frameworks from ASP.NET 1.1 to ASP.NET 2.0 would have to be made in order to take advantage of significant changes in Web technologies.

- Change the programming language from VB.NET to C#.NET. This would be a personal preference. Most applications in the “real-world” are done with C#.

- Change the database from MS Access to SQL Server either Express or Full Blown version. Although MS Access is fine for this version of the Bid Tracking System. I don’t want to get too behind on the database technology as it would be harder to change later.
• With the change in programming language, I would have to have a change in the source code. I would no longer use Get/Set Accessor Methods but rather C#.NET properties and place my get and set logic within there.

• Change from MS Access queries to SQL Server Stored Procedures. This is the preferred method of handling data access on the back end.

• Use dOOdads or some other type of software architecture builder to generate the BLL objects and stored procedures from the database design.

• Change the login logic from form authentication using roles to ASP.NET 2.0 User Management API.

• Change the templates and user control includes that I use to ASP.NET 2.0 Master Pages to provide the consistent look and feel that is needed.

• Add SSL (secure login) functionality

• Add system auditing and logging so that this information is captured.

• Add direct import programmatically into Labworks

As you can see, there are many changes that I would like to implement in order to make this project even better and I am excited to take on these new challenges to better my skills and to make Smith and Schaefer’s Bid Tracking System the best application possible!
Appendix A.
Research Information

During my research on how to properly implement a fully functional Web application, I came across many excellent resources that I would like to outline below. Perhaps this could assist another developer.

Appendix B.
Selected Diagrams

Figure B-1. Bid Tracking System Database / Class Diagram
Figure B-2. Technical Practicum II Project Breakdown
Appendix C.  
Code Snippets

C 1. Login Page Code Snippet

This is the ASP.NET code for the Login Page. This login pages is for all users of 
the system; it will authenticate the user against the database and redirect them to the 

'Role-based Security with Forms Authentication
'Special Thanks to Heath Stewart;
'http://www.codeproject.com/aspnet/formsroleauth.asp

'Initialize FormsAuthentication
FormsAuthentication.Initialize()

'Create our connection and command objects
dbConn = New OleDbConnection(connStr)
dbConn.Open()

sqlCmd = ("SELECT Role FROM Users " & _
"WHERE UserName = " & txtUsername.Text & " AND Passwd = " &
txtPassword.Text & ")

dbCmd = New OleDbCommand(sqlCmd, dbConn)

dbReader = dbCmd.ExecuteReader(CommandBehavior.CloseConnection)

If dbReader.Read() Then
  'Set session variables for username for user logging in
  Session("UserName") = txtUsername.Text

  'Create a new ticket used for authentication
  Dim ticket As FormsAuthenticationTicket
  ticket = New FormsAuthenticationTicket(1, txtUsername.Text,
            DateTime.Now, DateTime.Now.AddMinutes(30), True, dbReader.GetString(0),
            FormsAuthentication.FormsCookiePath)

  'Encrypt the cookie using the machine key for secure transport
  Dim strHash As String
strHash = FormsAuthentication.Encrypt(ticket)
Dim cookie As HttpCookie
cookie = New HttpCookie(FormsAuthentication.FormsCookieName, strHash)

'Set the cookie's expiration time to the tickets expiration time
If ticket.IsPersistent Then
    cookie.Expires = ticket.Expiration
End If

'Add the cookie to the list for outgoing response
Response.Cookies.Add(cookie)

'Redirect to requested URL, or homepage if no previous page requested
Dim strReturnURL As String
strReturnURL = Request.QueryString("ReturnUrl")
If strReturnURL Is Nothing Then
    strReturnURL = "login.aspx"
End If

'Don't call FormsAuthentication.RedirectFromLoginPage since it could
'replace the authentication ticket (cookie) we just added
Response.Redirect(strReturnURL)
Else
    lblMessage.Visible = True
    lblMessage.Text = "Username / password incorrect. Please try again."
End If

dbReader.Close()
dbConn.Close()

C 2. Add Estimator Code Snippet

This is the code that is used to add an estimator to the system. This code is a good
example of how the 3-Tier architecture works.

' Create new Estimator object using information from the Web form
' Use DataAccessLayer to add Employee to Database and retrieve
' auto-generated ID
newEstimator = New Estimator(Me.txtFirstName.Text, Me.txtLastName.Text,
Me.txtWorkPhone.Text, Me.txtCellPhone.Text, Me.txtEmail.Text,
Me.txtJobTitle.Text)
newEstimatorID = DAL.CreateNewEmployee(newEstimator)
Create new User object using information from the Web form
Use DataAccessLayer to add User to Database
newUser = New User(Me.txtUserName.Text, Me.txtPassword.Text, "Estimator", newEstimatorID)
DAL.CreateNewUser(newUser)

lblMessage.Text = "Estimator Added"

C 3. Project Business Logic Object Code Snippet

This is an example of how a business logic object is described within the Bid Tracking System.

Project - Business Logic
describes Project object

Public Class Project

attributes
Private ProjectOwner As String
Private ProjectName As String
Private ProjectLocationCity As String
Private ProjectLocationState As String
Private ProjectLocationZip As String
Private ProjectStatus As String
Private CompanyID As Integer
Private Id As Integer

constructor (7 parameters)
Public Sub New(ByVal aProjectOwner As String, ByVal aProjectName As String, ByVal aProjectLocationCity As String, ByVal aProjectLocationState As String, ByVal aProjectLocationZip As String, ByVal aProjectStatus As String, ByVal aCompanyID As Integer)
'invoke setter methods to populate attributes
SetProjectOwner(aProjectOwner)
SetProjectName(aProjectName)
SetProjectLocationCity(aProjectLocationCity)
SetProjectLocationState(aProjectLocationState)
SetProjectLocationZip(aProjectLocationZip)
SetProjectStatus(aProjectStatus)
SetCompanyID(aCompanyID)
'init to 0
SetId(0)
End Sub

'constructor (8 parameters)
Public Sub New(ByVal aProjectOwner As String, ByVal aProjectName As String, _
    ByVal aProjectLocationCity As String, ByVal aProjectLocationState As String, _
    ByVal aProjectLocationZip As String, ByVal aProjectStatus As String, _
    ByVal aCompanyID As Integer, ByVal aId As Integer)
'invoke setter methods to populate attributes
SetProjectOwner(aProjectOwner)
SetProjectName(aProjectName)
SetProjectLocationCity(aProjectLocationCity)
SetProjectLocationState(aProjectLocationState)
SetProjectLocationZip(aProjectLocationZip)
SetProjectStatus(aProjectStatus)
SetCompanyID(aCompanyID)
'set id from db
SetId(aId)
End Sub

'get accessor methods
Public Function GetProjectOwner() As String
    Return ProjectOwner
End Function
Public Function GetProjectName() As String
    Return ProjectName
End Function
Public Function GetProjectLocationCity() As String
    Return ProjectLocationCity
End Function
Public Function GetProjectLocationState() As String
    Return ProjectLocationState
End Function
Public Function GetProjectLocationZip() As String
    Return ProjectLocationZip
End Function
Public Function GetProjectStatus() As String
    Return ProjectStatus
End Function
Public Function GetCompanyID() As Integer
    Return CompanyID
End Function
Public Function GetId() As Integer
    Return Id
End Function

' set accessor methods
Public Sub SetProjectOwner(ByVal aProjectOwner As String)
    ProjectOwner = aProjectOwner
End Sub
Public Sub SetProjectName(ByVal aProjectName As String)
    ProjectName = aProjectName
End Sub
Public Sub SetProjectLocationCity(ByVal aProjectLocationCity As String)
    ProjectLocationCity = aProjectLocationCity
End Sub
Public Sub SetProjectLocationState(ByVal aProjectLocationState As String)
    ProjectLocationState = aProjectLocationState
End Sub
Public Sub SetProjectLocationZip(ByVal aProjectLocationZip As String)
    ProjectLocationZip = aProjectLocationZip
End Sub
Public Sub SetProjectStatus(ByVal aProjectStatus As String)
    ProjectStatus = aProjectStatus
End Sub
Public Sub SetCompanyID(ByVal aCompanyID As Integer)
    CompanyID = aCompanyID
End Sub
Public Sub SetId(ByVal aId As Integer)
    Id = aId
End Sub
End Class

C 4. Data Access Layer Example Code Snippets

These are some examples of the methods that reside with the Data Access Layer
of the Bid Tracking System. The Data Access Layer controls all database activities.
This following code snippet will take the BLL object User as a parameter and
insert the information within the BLL object into the database.

```vbnet
Public Function CreateNewUser(ByVal newUser As User) As Integer

    ' Validate Parameters
    If newUser Is Nothing Then
        ' Throw exception
        Response.Write("There is no user to add!")
    End If

    ' Execute SQL Command
    Dim sqlCmd As New OleDbCommand
    AddParamToSQLCmd(sqlCmd, "@UserName", OleDbType.VarWChar, 50, ParameterDirection.Input, newUser.GetUserName)
    AddParamToSQLCmd(sqlCmd, "@Passwd", OleDbType.VarWChar, 50, ParameterDirection.Input, newUser.GetPasswd)
    AddParamToSQLCmd(sqlCmd, "@Role", OleDbType.VarWChar, 50, ParameterDirection.Input, newUser.GetRole)
    AddParamToSQLCmd(sqlCmd, "@EmployeeID", OleDbType.Integer, 4, ParameterDirection.Input, newUser.GetEmployeeID)

    SetCommandType(sqlCmd, CommandType.StoredProcedure, SP_USER_CREATE)
    Return ExecuteScalarCmdWithIdentity(sqlCmd)
End Function 'CreateNewUser

The following code snippet will return an ArrayList of Users objects from the
information returned from the database.

Public Shared Function GetAllUsers() As ArrayList

    Dim dbReader As OleDbDataReader
    Dim users As New ArrayList

    ' Execute SQL Command
    Dim sqlCmd As New OleDbCommand
```
SetCommandType(sqlCmd, CommandType.StoredProcedure, SP_USER_GETALLUSERS)

Dim cn As New OleDbConnection(CONN_STRING)

Try
    sqlCmd.Connection = cn
    cn.Open()
    dbReader = sqlCmd.ExecuteReader(CommandBehavior.CloseConnection)
    
    While dbReader.Read()
        Dim newUser As New User(CStr(dbReader("UserName")), CStr(dbReader("Passwd")), CStr(dbReader("Role")), CInt(dbReader("EmployeeID")), CInt(dbReader("UserID")))
        users.Add(newUser)
    End While

Finally
    cn.Dispose()
End Try

Return users

End Function
References


