Group 6

Amanda Chaffin
Alex Godwin
Dimitris Koutsogiorgas
Jann Nguyen
Gurvir Sidhu
Lenny Tran

11/21/2005

Table of Contents

Purpose…….....3

Design Objectives…….....4

System Boundaries…….....6

Environment…….....9

Design Specifications

Prior to Election Day (Ballot Creation Software)……..13

Desk One (Registration Verification)…….…….....15

Desk Two (Ballot Box)…….…….....20

Purpose

The purpose of the Online Voting software system is primarily to provide a secure, efficient, and accessible means of voting to the public with a system that is easily updatable and financially sound. Current voting systems using custom-built hardware are expensive, difficult to maintain, and foreign in appearance to the average voter. We propose to offer a system that uses web-based software designed for a thin-client PC. This system would offer the following benefits:

Software-based updates

Current electronic voting boxes are very expensive. They also need to be replaced every few years as voting legislation changes. They are completely hardware based, and there is little adaptability in their design. Our system is almost completely software-oriented, so that as the times change, the system can adapt with little cost and fewer staff hours.

Few moving parts for easy maintenance

Hardware systems are difficult to maintain and frequently break down when they are most needed: Election Day. Because our Online Voting system contains almost no moving parts, the productivity is drastically increased, allowing you and your employees to focus on other tasks on your busiest days.

Familiarity of Web Brower

Times have changed, and one can find a computer in a large percentage of homes all across the nation. That means that many voters are going to be instantly familiar with a system that uses a web browser. This translates to less time spent trying to explain the often confusing hardware systems currently in-place. In addition, the system is easily portable between voting districts. By using “style sheets” that individually represent each

district, one station can quickly and easily handle voters from any supported area.

Design Objectives

In order for the design of the online voting system to satisfy the expectations of the customer, the system design will need to be broken down into categories. These categories will be the design objectives and they will describe the various areas the design will cover.

Accessibility

According to HAVA (The Help America Vote Act), all features and protection of privacies given to non-handicapped voters will be provided for handicapped persons. The system will support multiple languages including English and Spanish. For those voters who cannot, under their own power, place a vote physically, they will be provided a blow/puff mechanism to place their vote. For vision-impaired voters, the ballot will be read aloud.

Cost-Effective

Throughout the design process, cost management will be kept in the forefront. Cost management consists of balancing the quality of the system and the customer's expectations. Our system will take into account the overhead involved in upgrading a voting system in a county. The system will be upgradeable through software and whenever upgrades are needed, the new software will be loaded onto machines via the server. In addition, we are building all of our software for the servers, the databases, and the web servers on open sourced platforms, which aids greatly in keeping our overhead down.

Functional/Operational

The online voting system will let users use computers to place their votes as opposed to paper ballots or voting machines. Computers will be placed at designated voting locations. The system will be able to sort the candidates according to the local government's guidelines. Straight party voting can be implemented and it will be designed to follow the local government's guidelines and laws. The system will integrate the individual schemes for voting by mail, early voting, and Election Day voting. The system will store authorized votes and functionality will be added to handle provisional ballots. As per current law, the voting system will not keep tallies but will tabulate votes at a specified time. The system will be designed to audit "fall-off" votes. The ballots will be sorted using the appropriate layouts. These layouts will include party order and the sorting of last names within a party ballot. As laid out by HAVA, the system will let the voter verify his/her vote before submission, change his/her vote before submission, and give a warning when more than one candidate is selected in a section. The voter will have a chance for a vote audit before the permanent paper record is produced.

Security

In order for the voting system to be accepted by the public, security will be our focal point and our design will incorporate the most secure technologies available. Our design will protect the voting data from hackers because of the security designs and the platforms we are using. We will incorporate high-level security protocols such as RSA and SSL. Voters will login to the system using Smart-Card technology, which they will receive when they check in, removing the possibility of tampering with the Cards. A paper trail will be kept for recounts and security.

Public Support

In order for this system to be successful, it needs the public's support. The public will need to be convinced of the reliability and security of the system. We believe that our system will meet the public's expectations and put to rest the public's concerns about security. The system will be designed to make it easy for those voters who are not computer savvy to be able to vote with ease. Training will be provided for those that are new to the system.

System Boundaries

Machine malfunctions

In recent elections, electronic voting machines malfunctions include:

1. Giving voters ballot for the wrong district
2. Problems with the Smart Card reader delaying opening times
3. Recording votes incorrectly or breaking down, causing long lines during elections
4. Reversing election outcomes
5. Handing votes to the wrong candidate
6. Passing pre-election testing and failing on election day
7. Changing voters' choices on the screen
8. Subtracting votes cast by voters
9. Adding votes not cast by voters
10. Forcing states to hold new elections

As with any electronic system, unfamiliarity with the equipments, machine failures, printers jammed are normal problems; simple to catch and fix, provided the systems notify the users beforehand as to the possibility of a malfunction.

Public Opinion - *The inherent distrust*

One of the main problems with an electronic voting system is assure the public that it works, free of the dramatic software and hardware problems of the past. Those problems have cast increasing doubts about the accuracy and integrity of voting equipment, especially in a Presidential Election year. Issues such as "Phantom" votes (votes added by electronic voting machines, where some candidate candidates gained votes in the manual recount of the paper tapes printed by the machine or the accumulation software simply tallied wrong) proves that there are always glitches. These glitches can be caught because of extensive testing and close monitoring of the Election process however, we still have to prove that to the public.

Current Security

The booth machines operate independently, or connect only to a local election judge's workstation. They lack the hardware to connect to the Internet. To report results, most systems collect votes onto one voting machine or PC at the polling place. That machine then dials in to a PC at election headquarters and transfers that precinct's tallies over an encrypted modem-to-modem connection. Later, poll workers deliver the memory cards along with a printout of the results. The PCs used to collect and transmit results are not supposed to be connected to the Internet while tallying results.

Any programmer can write code that displays one thing on a screen, records something else, and prints yet another result. There is no known way to ensure that this is not happening inside of the current voting system. No current electronic voting system has been verified as secure even at the lowest level of the U.S. government or international computer security standards.

Machine Vulnerabilities

The machines' biggest vulnerability is that there is no way for a voter to know what the machine records when they cast their vote. Currently, there is no voter-verified physical record available for recounts. If the software goes awry or is tricked into flipping votes, no one will be able to tell as long as the total ballot count matches that of the number of voters while allowing for fall-off.

Voting Laws

There are now at least three bills in the Senate (S.1980, S.1986, and S.2045) and another in the House of Representatives (H.R.2239) that address the problems with e-voting by advocating voter-verified paper trail. That will not get rid of the problems per se, but it will help voters spot a miscount of their votes as they are cast, and will raise confidence in case a recount is needed. According to these laws, ballot creators must take into consideration (and it can vary between ballots):

- Party order
- Federal order, then State
- Sort by last name within party (and means of sorting)

The Cost

The federal Help America Vote Act of 2002 supplied nearly $4 billion in funding to replace punch card and lever machines, but mandated that states receiving the funds must replace those machines by this November. The cost for replacing the current system (which will be undertaken after this year's election) will be approximately 7-8 million dollars for Mecklenburg County alone. However, it would be beneficial if a system could be developed, in which only software is updated, and none of the systems currently up for

government contract are based solely on software updates. Optical Scan/Ballot Marking DeviceSolution will cost $10,000 per precinct. The total cost for new voting equipment that is disabled accessible and has Voter Verified Paper Ballots will cost $45 million. Finally, Direct Record Electronic (touch screen or pushbutton digital voting machines) will cost $145 million.

Accessibility

HAVA's primary importance to voting system developers is that it requires Election Boards to provide the means for disabled voters to take part in an election without assistance. Any system implemented would need to compensate for their disability (e.g. read aloud to the blind; allow puff/blow systems for those without the strength to push buttons). However, some voting systems might not provide the accessibility they promise such as asking blind voters to press a yellow button.

Environment

Our number one concern with the environment that we will use on all systems is that of security. We know that we simply cannot sell a solution that concerns something as heavily regulated as the America vote with security holes that should have been addressed before the system was released. Secondary to that is the cost effectiveness of the systems. Like the need for security, we simply will not be able to get political support for our system if we cannot prove that our system is more cost effective than both the current system and any other system in the making. Our third and final concern is making sure that the voting process remains within the bounds of the laws while making it as easy as possible to vote for the user and for the systems' administrators to maintain the system. If the solution is not one that is user friendly for the voter, the politicians will

not want to spend the money on it and, as a professional courtesy, it behooves us to make the system as easy to maintain as possible.

Servers

The servers that store all of the data for voting will run Linux Red Hat. The reason that we choose Linux is that Linux, as opposed to Windows, is a more secure operating system and is harder to hack. In addition, by using Linux, we can control what ports are open at what time. Since we plan to shut down all of the ports other than HTTP (hypertext transfer protocol) during Election Day, telnet access, for example, will not be accessible. That solves a lot of the security problems right there, as people will not be able to use a port scanner to find points of access to the server other than what we want them to have. Red Hat also has a thin client bundled into their operating system, which we will be using in order to keep the amount of information the thin clients have to parse through to a bare minimum. This will both speed up the process for the voter as well as insure the continuing security of the thin clients themselves. The next reason that we choose Red Hat is that is a free operating system. We will have to follow the open source rules and the GNU general public license laws but that will not be a problem and, in fact, may actually aid us in insuring the system is as tightly controlled and secure as we can make it. As for the final reason, ease of use, while it is true that there is a slightly steeper learning curve with Linux as opposed to Windows, we believe that we can write the documentation for both the user and the systems' administrators, which will be very effective in teaching them how to use the system.

HTTP Server

The secondary reason for choosing Linux is because we can use the Apache web server

on the server and, in fact, we will have to. Also under the open source code and GNU general public license laws, the Apache web server is free is the most popular web server on the Internet. Completely configurable by the systems' administrators, Apache couples easily with MySQL (my structured query language) and PHP (Hypertext Preprocessor) and allows for things such as DBMS-based authentication databases (database management software) and highly customized error messages. Apache also has SSL (secure socket layer) support built in as well as the ability to customize the log files and filtering support. The filtering provides a handler, which will process files before they are sent to the client. Security of the Apache server is easy; all we have to do is set the root and user permissions. Since there will be just one root access, they will receive full access – read, write, and execute. All other users will be read only and only in the actual web accessible folders. If we shut down the AllowOverride function (done in the server configuration file), the server will not allow an .htaccess file to work when loaded onto the server. Therefore, even if someone does hack the server web pages, they will not be able to set up their own passwords and gain access to the server. Between the customizable configuration, the cost effectiveness, and the ability to lock the server down completely from outside hackers, we believe that the Apache web server is definitely out best option. Finally, Apache is fairly intuitive and simple to learn once it has been implemented on the server. After compiling the web server with PHP and MySQL support and installing it on the server, the systems' administrators set up the access accounts and shuts down the afore mentioned areas, and the system basically runs itself.

Databases

The tertiary reason for choosing Linux is that it supports MySQL coupled with PHP. We

choose this for several reasons; the most important being our familiarity with the database as well as the scripting language itself. The MySQL database, unlike Microsoft's SQL, is more secure and the programs designed to input data into the database work better than the Enterprise Manager, which tends to be unwieldy and has known security holes. In addition, with the most current release, MySQL is now more functional than ever and the designers made sure to include all of the functions one can use in any other ANSI (American National Standards Institute) based scripting language. With PHP, we can design, implement, and maintain the databases running holding the ballots as well as the database holding the voting results with a minimum amount of human effort. We will write our own PHP database management program to insure that it as easy to use as possible – removing the need for a Database Manager to have to set up each individual district voting pages. The database that holds the ballots will, more than likely, be indexed first by state and then by district to ensure ease of use. The database that holds the voting results will not store anything other than the candidates and their votes. It will not even tabulate the votes, according to current law, until after the poles have been closed down for the day. Finally, both PHP and MySQL are free and easy to configure on Linux Red Hat servers.

Thin Clients Browser

The thin clients that will replace the current voting machines will be able to run any operating system that currently supports internet access. Since we are loading the data onto servers, it does not matter what operating system runs the thin clients. Our only concern at this time is which browser we will use on the thin clients. Since Internet Explorer has known security holes, we are going to download and use Mozilla's Firefox.

Firefox has an automatic popup blocker, prevents spy ware by not loading ActiveX controls, and will work on virtually any system we choose to load it on. Besides being a free download, Firefox supports PHP pages and the current HTML (hypertext markup language) version so we will not have to worry about formatting issues.

Thin Client Options

To give our client options, we have decided to provide support for two versions of thin client. The first is a simple flat-screen monitor and implements only the mouse for a point and click interface. The second thin-client we will support utilizes a touch-screen in place of the mouse for further ease of use at a slightly increased cost. By supporting both, we are allowing the client to choose the option that best fits their budget, and in addition, a single site would support both types of thin client.

Prior to Election Day: Ballot Creation Software

Our ballot creation software combines the ease of using a word processor with the power found in many web-page development applications. Our system presents you with a few simple choices about font, size and style so that you can be in accordance with the ballot requirements of your district. Then, as you are defining the choices for each office, our software allows you to choose quickly and easily ordering options so that your ballot follows state and federal guidelines. Then, once you are done, it exports the HTML in the format recognized by our voting system, and saves a style sheet for future use in your next elections.

Easy HTML and PHP Creation

Our other system components make it easy for you to update your system by implementing easy to use web software. To do this, your ballots need to follow strict

HTML guidelines for implementation into our system. However, to make this easier on you and your office, our ballot creation software allows you to create the ballot you need and takes care of the technical side for you. There is no need to have an HTML or PHP guru on staff, and no worries that your ballot may be incompatible with our ballot box software. It is all done for you, so you can focus on other things.

Order Ballots by Office

We understand that your district may have a different guideline for ordering of the offices on the ballot. To keep this simple, we allow you to define the order they are presented within as a simple numbered option once you are finished defining each office on the ballot.

Straight Party Voting

We have provided a means to designate which of the offices of your ballot are preset when a voter declares a straight party ticket. A simple click in the “Pre-select for Straight Party” box for each office will allow our ballot box software to pre-select only the required ones, and leaving the rest blank so the voter can then make their decisions.

Party Ordering within Office

In our software, you define the political parties present within your ballot, and then create a rule for their ordering. Then, when defining each office, you can choose whether the office adheres to the rules for the ballot or uses its own. Our software is as flexible as you need it to be.

Ordering by Name within Party

The software will allow the candidate names to be arranged to help reduce the impact of voter bias towards the candidate(s) listed first. Our software provides you with a number

of different options for ordering, including alphabetical, reverse alphabetical, or either beginning at a specific letter. Alternately, if you would rather designate the order of candidates by hand, our software allows you to do so.

Option for Referendums

The software will allow the voter to vote on referendums such as land grants and bonds. Once you have entered a description of the referendum into our software, you need merely to click the "Referendum" option to designate it as such.

Desk One (Registration Verification)

At the first station, the voter will check in and receive the Smart Card. To check in, the voters will be asked for their name and then the user will verify their identity via their address in the database. Once the user has verified the identity, the user will generate a Smart Card for the voter. The name of the voter will be searched in the voter database and his/her eligibility and constituency will be returned, allowing the system to access the correct ballot for the voter. The Smart Card will then be loaded with the ballot for that particular voter as well as a unique identification number that will be generated for each voter. The verification process will ensure that the voter is getting the correct ballot and that voter has cast his/her vote; multiple votes will not be allowed. If the voter is not eligible then the appropriate message will be returned. If the voter has already voted, the system will display the appropriate message. The constituency data will also bring up the available offices and candidates for the voter's constituency. The ballot itself will not contain any of the personal information of the voter other than the unique identification number issued to the voter, which will not be permanently recorded. This will be used in tracing voters if and only if there are any problems or questions that arise concerning

voter eligibility and fraud and only a Judge would have the authority to order the access of the number. A flag will be placed in the voter database indicating that a person has already voted. The voter's information will be kept in a central database, which will be entirely separate of the server that holds the actual votes. This file will interact with other agencies that accept voter registration such as the DMV (Division of Motor Vehicles). In addition, any information that is deemed important to maintain the accuracy of the voter registration database is recorded and sent to the county board of elections.

Technical

The voting system will use PHP in order to retrieve data from the MySQL database. Like MySQL, PHP is an open source programming language used for developing server-side applications, dynamic web content, and other software. The PHP architecture is popular because it is secure, scalable, reliable, and inexpensive. MySQL will be used for this system because it is the best tool for the job. Information retrieval is quick and easy through MySQL. MySQL is also a secure, stable, and a reliable technology. MySQL follows the ANSI/ISO (American National Standards Institute/International Organization of Standardization) standards for SQL and the current standard for MySQL is SQL: 2003. The Apache HTTP server will be used to house the system because of its compatibility with the Linux operating system. This server includes the following features: SSL and Transport Layer support, a proxy module, a useful URL (uniform resource locator) rewriter, custom log files, and filtering support. Other features of the server include highly configurable error messages, DBMS-based authentication databases, and content negotiation. Apache is supported by several GUIs (graphical user interfaces) which makes configuration of the server easier and more intuitive.

Database Build Specifics

The database holding the voter information will be ported into our server database via a comma-delineated text-based file. No matter what the customer is currently using to store the information, every database built around the ANSI standard has the ability to export the data via a text-based file. MySQL allows us to import the files into the database that we will set up (see Figure 1). That database will have a primary key and two secondary keys, which will allow us to search for a particular row in the database. The primary key will be the last name column and the two secondary keys will be the first name and the address. The name keys will be the information that must be input into the database, if the names are not there, the database will be set to reject the message with a warning. Since we have keyed the data to the name, searching for the record of that particular voter will be done easily and simply with the search feature that will be built into our database GUI. After the import of the main data is finished, the Board of Elections will also be able to add, via the same GUI, new voter registration information simply and easily (see Figure 2). In addition, deleting voter records of either the deceased or felons who have had their voter registration removed will not be necessary. Instead, we will flag the records in the database, which will allow the ability to track the names of the deceased, for example, and make sure that voter fraud can be caught immediately at the first station instead of well after the fact. In addition, this will allow the Board of Elections greater ease in dealing with people who have had their right to vote removed. They can remove or add flags with the touch of a button after, of course, cross checking with the police (see Figure 3). Those functions, however, will not be available to anyone outside of the Board of Elections office and will not be accessible on

Election Day to prevent anyone from tampering with the voter database.

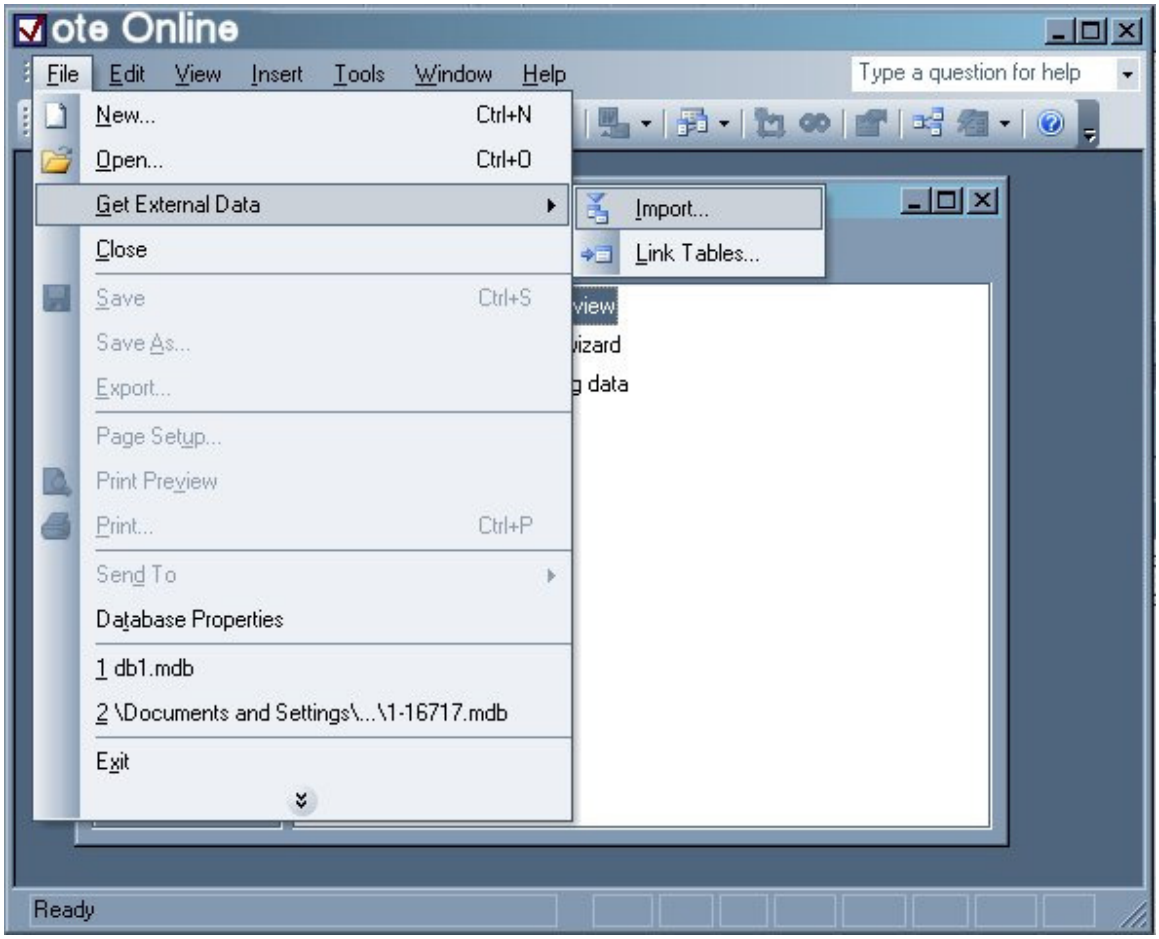

Figure 1 : The import feature of the database manager.

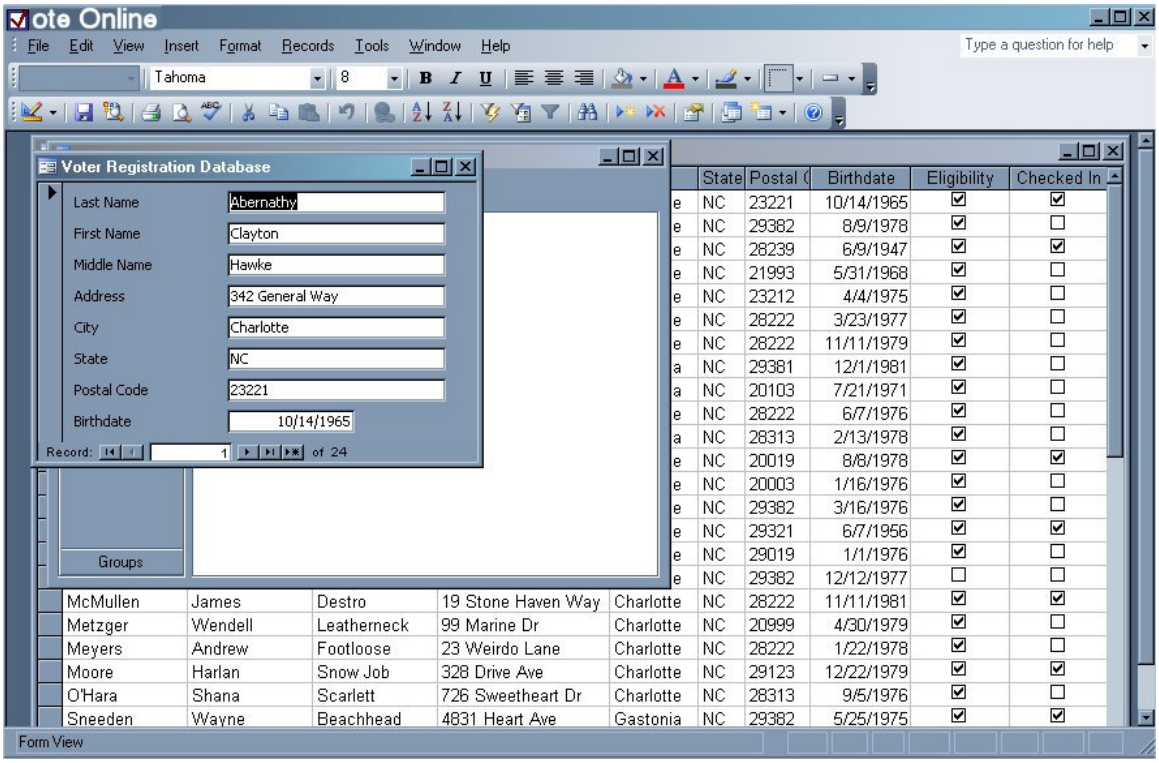

Figure 2: The GUI to insert or remove records from the database.

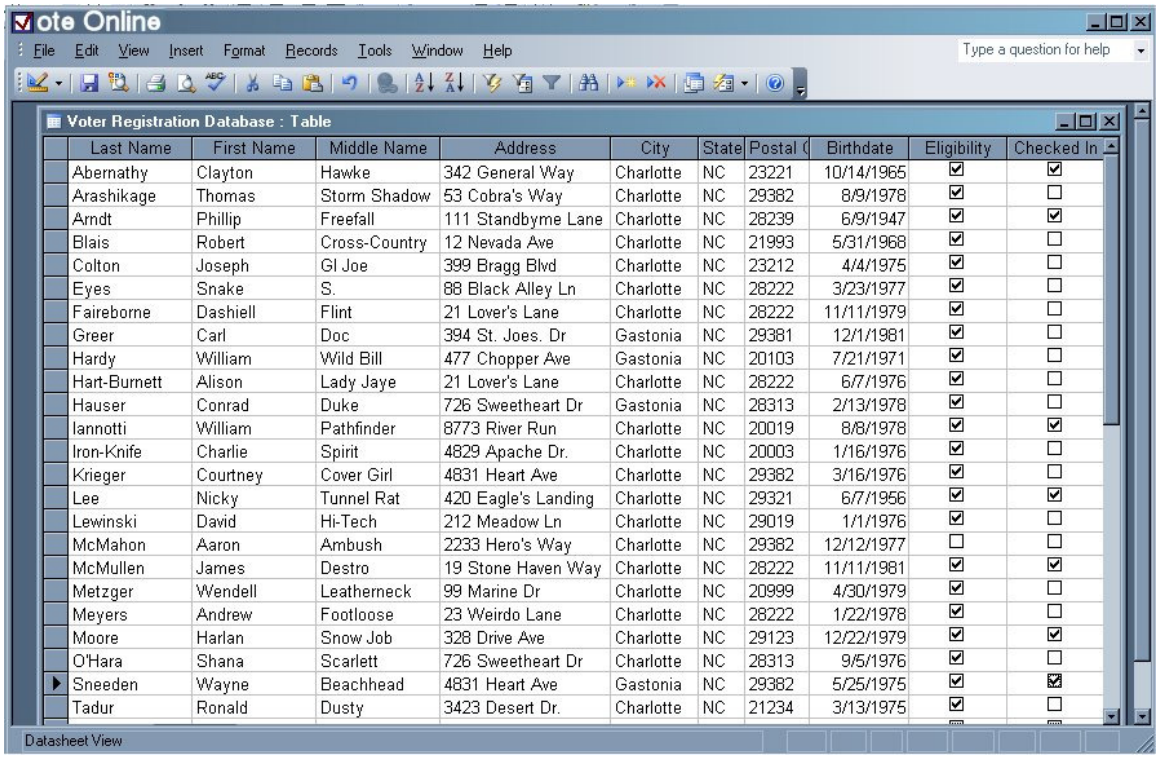

Figure 3: What the database looks like.

Desk Two (Ballot Box)

This section provides a detailed description of the website, the vote-casting portion of the software. By using this system, the voter marks his/her ballot and casts a vote.

After completing the registration verification process at "Desk 1" and attaining a personalized Smart Card, the voter proceeds to the voting booth. Inside the voting booth is a thin client for voting, into which the voter inserts the Smart Card.

The first screen shown is the "Welcome screen" which confirms the Smart Card was read properly. This screen also introduces the voter to the system and displays brief voting instructions. After reading the instructions, the voter is ready to view his/her ballot and begin voting.

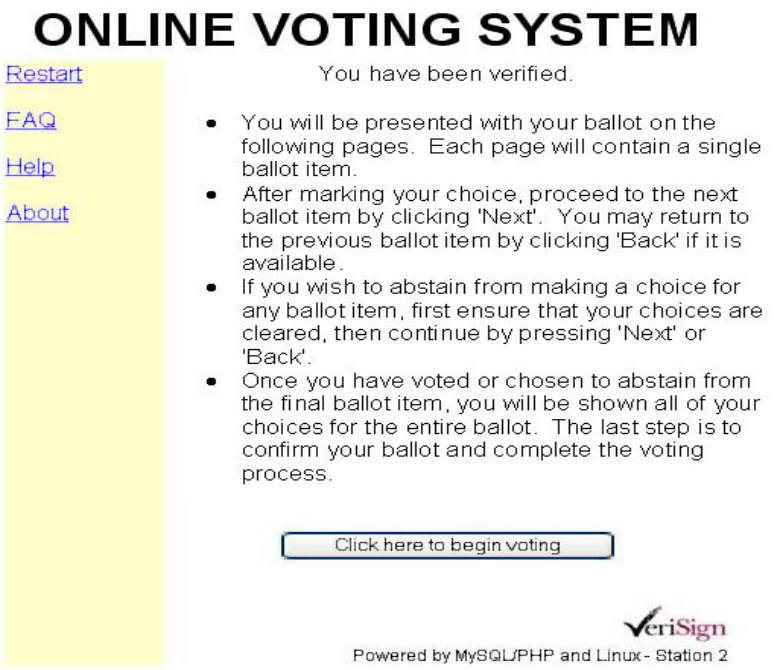

Figure 4: First screen upon loading Smart Card

The first ballot page presented to the voter, in most U.S. elections, is the option to designate a straight-party vote. This page is uniquely designed to mark the rest of the voter's ballot as "Democrat" or "Republican" should he/she choose. In districts that allow split ticket voting, the voter would simply reselect his/her choice on the subsequently presented ballot pages. Some states, such as North Carolina, do not allow the Presidential candidates to be selected via a straight party ticket. Therefore, voters in those states would automatically be taken to the Presidential candidates' page – removing some of the confusion concerning straight-party voting.

Figure 5: Straight party selection.

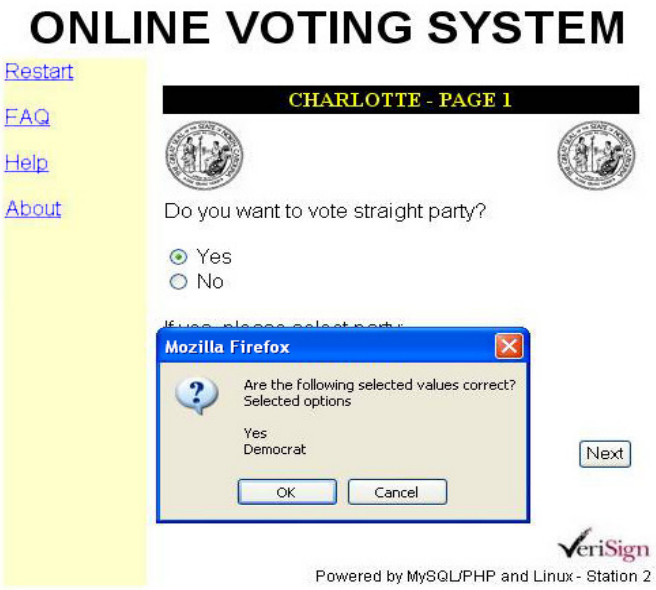

Figure 6: Verify selection.

The next screen demonstrates the page displayed for a typical ballot item, on which the voter will mark his/her choice for a candidate. Note that in addition to the typical voting method, the voter has two other options similar to paper ballot: write-in and abstain. A box is provided on appropriate ballot items for write-in votes. To abstain from any ballot item, the voter simply submits an unmarked page for the respective item.

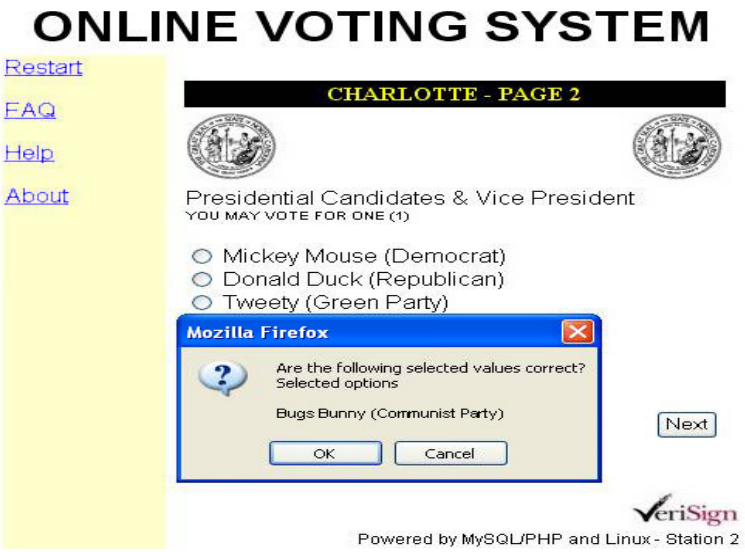

Figure 7: Voter selection for President

The system provides the voter with two methods of confirming his/her ballot. The first comes in the form of a "pop up window" informing the voter of the choices marked on any given ballot item. These appear after every ballot item.

Before completing the voting process and submitting the ballot, a screen displays all of the choices marked on the ballot. From this page, the voter can decide to submit his/her ballot or change any individual ballot item. By providing these additional safeguards, the voter can have confidence the software is properly recording his/her vote.

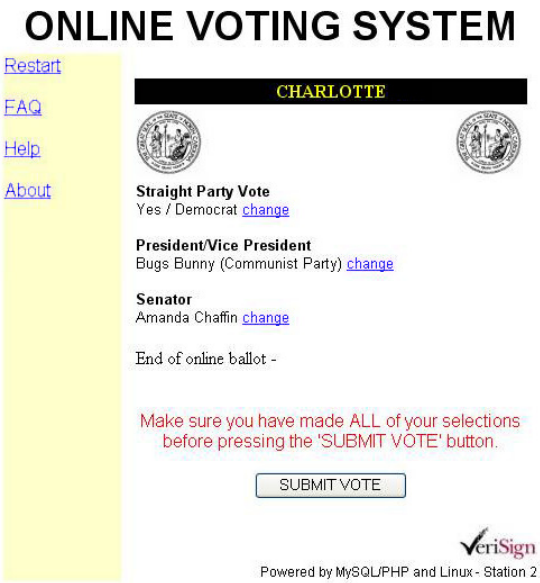

Figure 8: Verify the selection page.

Upon choosing "submit vote", a final screen informs the voter that the voting process is complete and the voter leaves the booth. This page is available for any additional instructions to the voter.

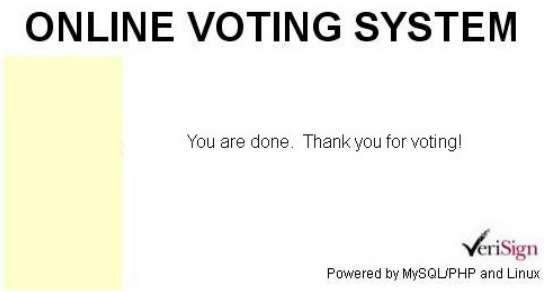

Figure 9: Voting completed.

Once the voter has finished and confirmed their vote, the data will be transmitted via a LAN (local area network) connection to the onsite server. This server will take the recorded votes, store the data in a database, and hold the records without tabulation until the end of the Election Day and the poles close. Once the poles are closed, the server will be connected to the Internet via a secured line and the database holding the votes will be transmitted to the County Board of Elections. After the data is sent and the server is taken offline once more, the votes will be tabulated at the local site and at the County Board of Elections to insure that the integrity of the data has not been compromised. Since the computer is doing the counting, the count will be fast and the redundancy of the system will insure the count is accurate.