A Universally Usable and Anonymous Approach to Voter Write-Ins Using Multimodal Interaction and Prediction Techniques

by

Shaneé Terese Dawkins

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 12, 2011

Keywords: Accessibility, Universally Usable Interfaces, Electronic Voting Systems, Multimodal Interaction, Text Prediction, Privacy

Approved by

Juan E. Gilbert, Professor of Computer Science and Software Engineering
Cheryl Seals, Associate Professor of Computer Science and Software Engineering
Richard Chapman, Associate Professor of Computer Science and Software Engineering

Abstract

The primary objective of the research in this dissertation is to develop a system in which any person, regardless of ability or disability, can efficiently, anonymously, and independently write-in a candidate's name during an election. This research is designed to be beneficial to voters during elections. Not only does this research embrace the needs of the 36 million United States (U. S.) citizens who are disabled [64], but it also affords voters the opportunity to properly write-in a candidate's name as intended. The method presented here utilizes multimodal interaction, i.e. speech, touch, and switch, to allow voters to privately spell the name of the candidate they intend to write-in. This method also implements name prediction, increasing the efficiency and accuracy of the write-in voting process. In accordance with the standards for usability set by the International Organization for Standardization (ISO) [31], the hypothesis of this research is that the system designed would be effective, efficient, and provide user satisfaction. An experiment was performed to evaluate the system against these measures. The results of the experiment show that the system designed is an effective and efficient solution to writing-in a candidate's name, and with minor adjustments, will be more than satisfactory for voters to utilize.

Acknowledgements

First and foremost, I would like to thank my lord and personal savior, Jesus Christ, without whom I would be nothing. I would also like to thank my loving husband, Jeremy Dawkins, for his continuous love and support. I would like to thank all of my family and friends who encouraged me to make it this far. I especially want to thank my mother, Francine Wright, for always pushing me to my full potential, and for being the inspiration for me to pursue my doctoral degree. Additional thanks must be given to Dr. Juan Gilbert for his continuous support, encouragement, mentorship, and guidance throughout this entire process. I wish to thank the remaining members of my committee, Dr. Cheryl Seals and Dr. Richard Chapman, for their reviewing and advising efforts. I want to thank my friends and colleagues who took the time out to review this dissertation. To the members of the Human Centered Computing Lab at Auburn University and Clemson University and other fellow graduate students, I am grateful for your advice, support, and assistance in achieving this goal.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The design of ballots is the foundation of successful election operations. With a usable ballot design, voters can feel certain about their votes cast, and election officials are able to feel confident about interpreting voter intent. Furthermore, with a universally accessible ballot design, disabled voters have the privilege of feeling this same certainty. With a universally accessible ballot design, every voter, regardless of ability or inability, will have a way to vote independently and anonymously, while still maintaining system security and efficiency. Today, a properly designed ballot interface is one of the key aspects to running a successful election; an interface that enables all voters to have independent access to the ballot.

As technology for electronic voting systems continues to develop, there is an increased need for universal design in these systems [2]. A universal design ensures that systems are as usable as possible by as many people as possible regardless of age, ability or situation [16]. By focusing on the voter and their needs, the design of electronic voting systems will satisfy the aforementioned universally usable criteria.

With the ballot privacy constantly being a major concern in the design of voting systems, it is often difficult to implement voting technology that incorporates a private, yet universal, design. Some developers today address this issue through the design of their electronic voting systems [54]; however, these electronic voting systems have yet to integrate universal design into the writing-in of a candidate's name.

1.2 Problem Description

Currently, there is no solution for writing in a candidate's name that is universally accessible. As stated previously, developing systems with a universal design ensures that the system can be used by anyone, regardless of abilities or disabilities. Current mainstream electronic voting systems simply cannot accommodate the range of voter abilities with their current write-in techniques. In most counties, in order for voters with visual or motor impairments to vote, a voting official must enter the voting booth with him or her to write, or type, the candidate on the ballot for which the voter intends to vote. The lack of multimodality and accessibility in these write-in methods only accommodates sighted voters with full mobile ability. This violates the privacy of the voter and the anonymity of the voter's ballot.

1.2.1 Voter Intent

Current voting methods provide the voter with the freedom of writing-in any candidate of their choosing. Since the name of the candidate is not listed directly on the ballot, the voter should know the proper spelling of the candidate's name. In some cases, candidates who intend to get elected through write-in votes have long, complex spelled names. How then, should the vote be counted if the name is spelled incorrectly?

In Maryland, for example, any vote for a write-in candidate can be counted; given voter intent can be determined from the ballot. The Maryland State Board of Elections states that, "any abbreviation, misspelling, or minor variation in the form of the name of a candidate shall be disregarded in determining the validity of the write-in vote as long as the intended candidate can be determined," and that, "writing the last name only will constitute a valid vote, unless there is more than one candidate with the same last name" [42][43]. According to this format, the names "William Johnson," "Wiliam Jonson," "Bill Johnson," "William E. Johnson," "W. E. Johnson," and “Johnson,” could all constitute votes for the same candidate. This method of counting write-in votes would be effective if the election officials could effectively, without bias, determine the intent of the voter. Unfortunately, determining voter intent is not readily welcomed in all states.

The 2010 general election in Alaska was the basis of controversial turmoil. Joe Miller and Lisa Murkowski were candidates for U.S. Senate in the election on November second. Murkowski was registered as a write-in candidate, and upon completion of the vote tallies, was declared the winner of the race. Murkowski obtained more than 97 percent of the write-in votes; more than 11 percent more votes than one of her opponents, Joe Miller [59]. Miller, however, has filed a lawsuit, claiming that write-in votes were not counted properly according to Alaska state law. Alaska code states that, to vote for a write-in candidate, the voter must fill in the designated oval and write the name in the designated space, and that the name written must match the name written on the registration or the last name of the intended candidate [9][44][45]. Miller’s argument is that elections officials accounted for voter intent, from which certain ballots, like those shown in Figure 1.2.1, were counted as votes for Murkowski, violating the aforementioned law. Such a controversy could be avoided by presenting voters with a list of registered write-in candidates, of which they could choose. By using the system presented in Chapter 3, election officials would not need to rely on voter intent to tally the votes.

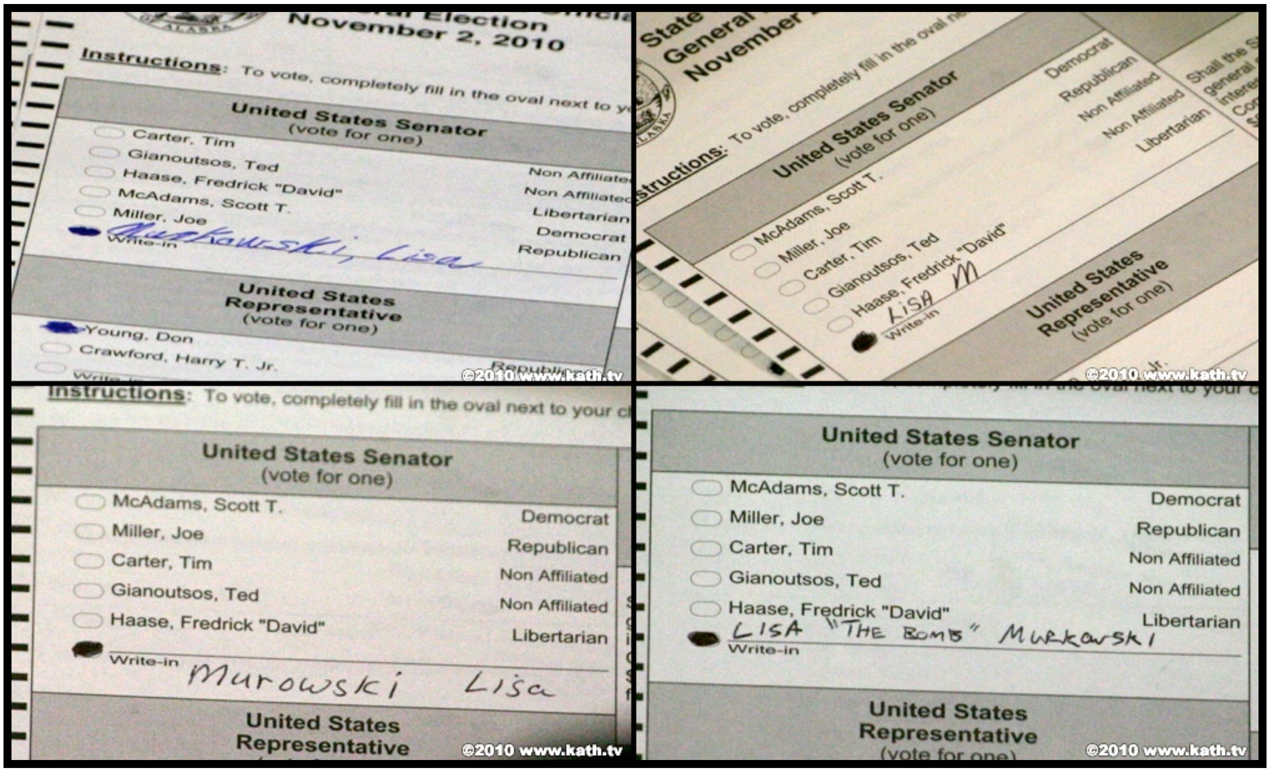

Figure 1.2.1 Ballot Variations for Write-In Candidate, Lisa Murkowski [46]

1.3 Goals, Approach, Contribution

Currently, there is no method to spell a name for writing in a candidate that incorporates a universal design and meets the requirements set forth by the Election Assistance Commission (EAC); no system allows an individual with visual and motor impairments to spell a candidate's name privately and securely. The objective of this research was to develop a system in which a person, regardless of ability or disability, can efficiently, anonymously, and effectively spell a candidate's name through multimodal interaction.

The universally accessible approach to writing in a candidate's name was sparked by the need to provide all voters with the ability to perform voter write-ins independently. This is done through the method presented here; a predictive approach to spelling through multimodal interaction, allowing voters to quickly and anonymously spell a candidate's name for any position or office during the voting process. The research to be conducted intends to capture and analyze the efficiency and effectiveness of writing-in a candidate's name anonymously through multimodal interactions. The hypothesis of this work is that the predictive spelling method will be effective, efficient, and provide user satisfaction.

The immediate contributions of this research will directly benefit disabled voters during elections. In addition, the results of this research could lead to the adaptation of this system in information sensitive search functions for various applications.

1.4 Organization of the Research

The outline of this dissertation began with a brief introduction to the motivation of this research. The remainder of this dissertation is organized as described below.

Chapter 2 provides an overview of relevant areas of research. Following a brief introduction to write-in voting compliance, topics on universal accessibility, current methods of voting interaction for people with disabilities, and usable privacy and security are provided. Also discussed are predictive text methods and information privacy.

Chapter 3 provides a thorough explanation of the system design, including an in depth look at the design solution, and the prediction and clustering algorithm implemented in this research. The chapter concludes with an overall summary of the details of the system.

Chapter 4 presents an initial implementation of the suggested method, along with a recount of the results from the preliminary experiment. Furthermore, experiment details and expected outcomes are discussed.

Chapter 5 reports a detailed analysis of the final experiment and the results of the analysis. It conclusively presents the effectiveness, efficiency, and overall user satisfaction measures.

Chapter 6 summarizes the research performed and concludes with relevant contributions and directions for future research. Conclusively, **Error! Reference source not found.** provides a detailed bibliography.

Chapter 2

Literature Review

2.1 Election Write-In Compliance

In United States' elections, voters have the option to vote for a person who is not listed on the ballot by writing that person's name in a dedicated space on the ballot sheet. Because election law is mandated by each individual state and not federally, laws pertaining to writing in a candidate vary across all states [25]. Most, but not all states only allow write-in candidates for general elections. Similarly, some states require people to pre-register as a write-in candidate for an election, while others do not. Some states do not allow candidates to be written-in at all [25]. Due to the large variance in election law across states, it is impractical to discuss them in their entirety for this dissertation. This section will highlight the election laws of states of Alabama, Maryland, and California to express the variety amongst all states.

The state of Alabama uses paper ballots in their elections rather than electronic voting machines. As such, voters literally have to *write* in a candidate's name on the ballot (see Figure 2.1.1). Alabama only allows voters to write-in a candidate's name for non-municipal general elections. The general election ballots have a space under each office for the voter to write-in any name not printed on the ballot. In order to vote in this manner, voters must write the desired candidate's name in the space provided on the ballot and register the vote by marking the designated write-in space for that office [5].

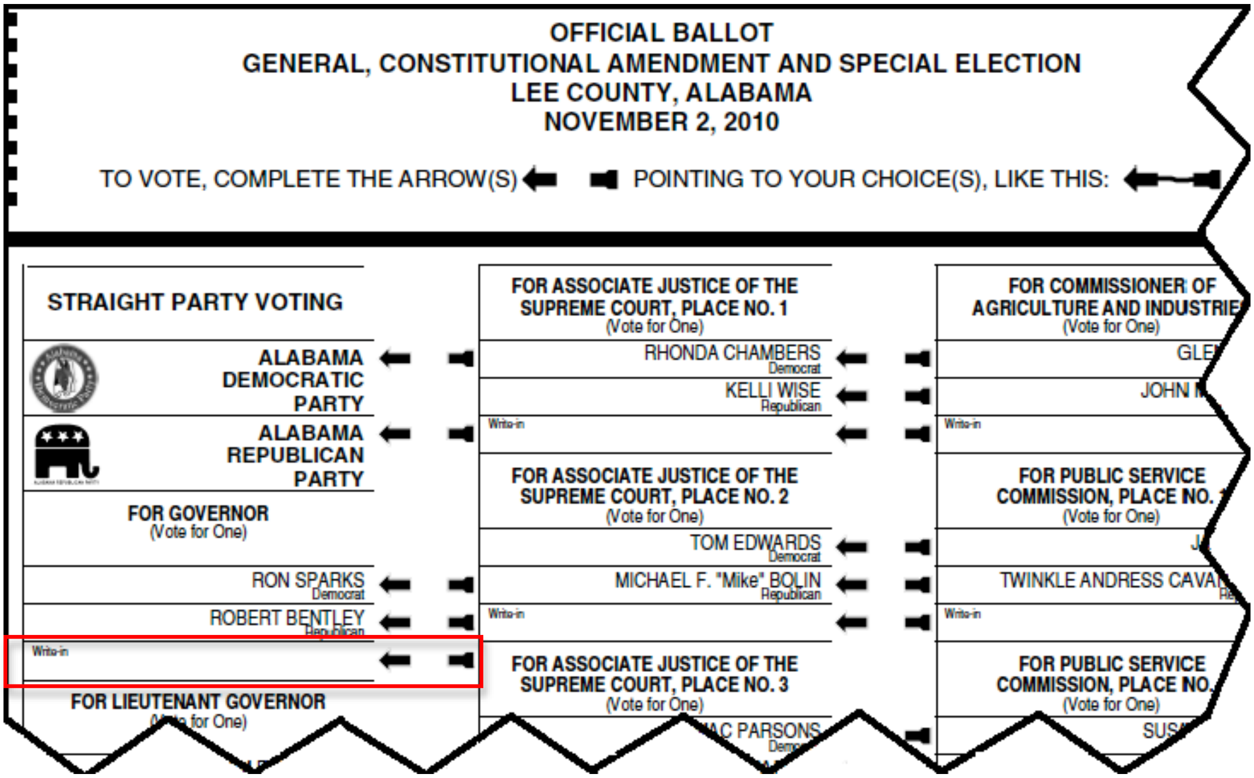

Figure 2.1.1 2010 Alabama General Election Sample Ballot [8]

Polling places in the state of Maryland use electronic voting machines for elections. Voters can vote using a touchscreen ballot, audio ballot, or provisional ballot. Using the touchscreen, the voter must first indicate that he or she wishes to write-in a candidate (see Figure 2.1.2). If indicated, an onscreen keyboard is shown where the voter can type and submit the candidate's name (see Figure 2.1.3). If a voter has a disability that prevents him or her from writing-in a candidate's name, an election official enters the voting booth to assist the voter [71]. If a candidate intends to be elected as a write-in candidate, s/he must file a certificate of candidacy prior to the election [40][41]. However, in Maryland, writing-in a candidate is not allowed in primary elections [39].

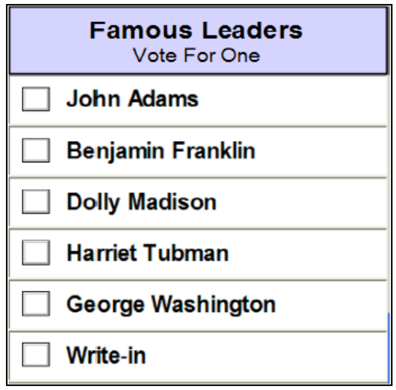

Figure 2.1.2 2010 Maryland Ballot Write-In Indication Screen [43]

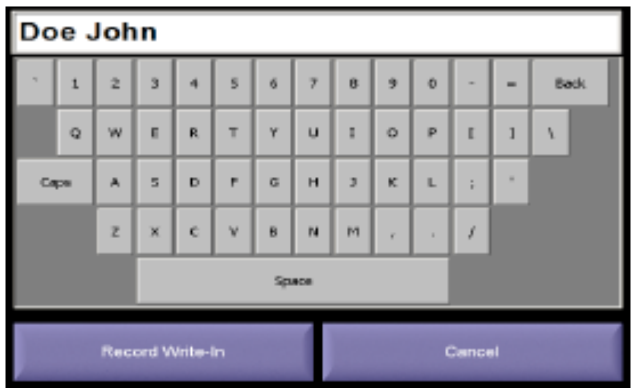

Figure 2.1.3 2010 Maryland Ballot QWERTY Keyboard Write-In Screen [43]

California state election law allows any person to be written in for any public office for any election [15]. The names of write-in candidates can be written on the ballots in the space provided, for the voting systems in California, whether directly beneath the list of candidates or otherwise mentioned in the voting instructions [14]. Some California voting systems do not allow pre-printed stickers, stamps, or other unapproved devices to be used to write-in (or stamp-in) a candidate's name [58]. Like Maryland, people who intend to get elected by means of write-in, need to be certified by filing a statement of write-in candidacy prior to the election [15].

2.2 Universal Accessibility

The following sections are aimed at highlighting accessibility in computing systems. Universal design is first discussed from a general computing perspective, and then discussed as applied to voting, specifically electronic voting systems. Universal Accessibility is the underlying motivation of this research.

2.2.1 Universal Design

Universal design is a key feature that should be incorporated into the design of any computer system. Universal design has been researched by different institutes and organizations [16][30][47], and has been defined similarly amongst them. North Carolina State University's Center for Universal Design states that universal design is an approach to the design of all products and environments to be as usable as possible by as many people as possible regardless of age, ability, or situation, and that it benefits everyone by accommodating limitations [16]. Accordingly, the center adds that the intent of universal design is to simplify life for everyone by making products, communications, and existing environments more usable by as many people as possible at little or no extra cost [47]. The Institute for Human Centered Design poses another context by stating that universal design is a framework for the design of places, things, information, communication and policy that is to be usable without special or separate design, and is an orientation of user experience to any design process [30]. Simply stated, universal design is human-centered design of everything with everyone in mind [30]. An example of an inaccessible design versus a universal design is shown in Figure 2.2.1 and Figure 2.2.2.

Figure 2.2.1 Inaccessible Stairwell [33]

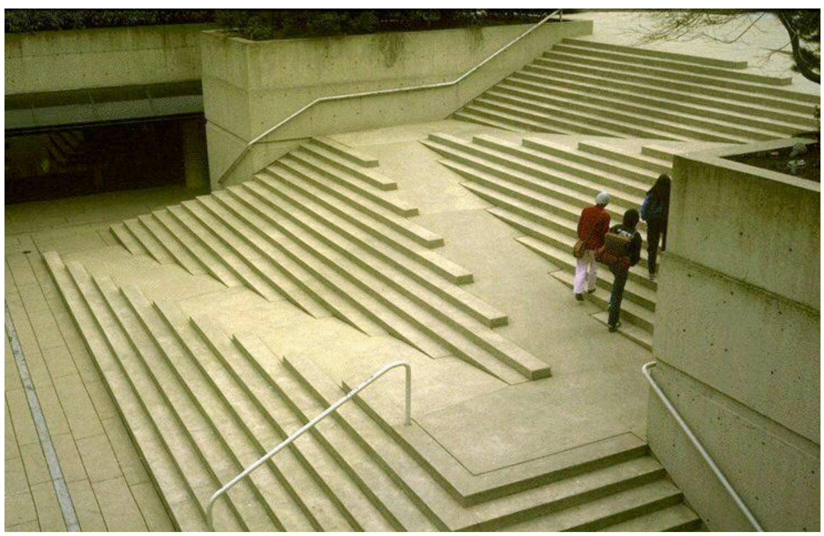

Figure 2.2.2 Stairwell Incorporating Universal Design [21]

2.2.2 Universal Accessibility in Voting

As a result of the major issues faced in the 2000 United States Presidential Election, the Help America Vote Act (HAVA) of 2002 was created [26]. HAVA aimed to prevent these problems from happening in future elections. From HAVA, the United States Election Assistance Commission (EAC) was established. One of the goals of the EAC was to adopt Voluntary Voting System Guidelines (VVSG), which expand access for individuals with disabilities to vote privately and independently [66]. The VVSG is the third revision of voting system standards, following the 1990 and 2002 Voting System Standards (VSS). In 2007, the VVSG was made public. The VVSG now addresses the advancement of technology and provides requirements for voting systems to be tested to ensure functionality, security, and accessibility [53].

It is now necessary for existing and novel electronic voting systems to implement a universal design. Chapter 3 of the 2007 VVSG proposes requirements for the usability and accessibility of electronic voting systems [2]. Due to the diversity amongst people voting in elections, a universal design is essential to the success of electronic voting systems. The VVSG states that all voters must have access to the voting process without discrimination, and that the voting process must be accessible to individuals with disabilities, including non-visual accessibility [66]. It also states that the voting system should be independently accessible to as many voters as possible, which further emphasizes the need for a universal design.

2.3 Predictive Text

Word prediction and word completion are common phrases used to describe suggestion methods for entering text. Many systems are described using the terms "word prediction" and "word completion" interchangeably, whereas others define the two as different techniques [68]. For the purposes of this research, the approach discussed utilizes the term word, or name, prediction.

Today, there are a plethora of different methods used for word prediction. Word prediction is often defined as a design in which systems predict the word the user wants to type, based on what s/he has typed thus far [63]. Word prediction initially was an assistive solution to enter text for people with motor impairments [34]. Word prediction has since evolved into a way for people to expedite their typing rates for text messaging on mobile devices [13]. This section describes popular methods used today.

2.3.1 T9

T9 (Text on Nine Keys) is a mobile text input system, originally introduced by Tegic Communications, designed to make it possible for users to type as fast as they can think [65]. T9 significantly improved text entry on the phone's fixed, 9-button, keypad because it guesses the word desired from the text that the user has already typed by combining the words from its dictionary with the input it received from the user [35]. Using T9 text input, users are able to type words on a mobile phone using just one key press per letter [48]. On most cellular phones with T9 capabilities, when a user presses a key, the most used letter of that key that fits with the keys already pressed is displayed to the user, and s/he indicates that the word is complete by pressing the “space” key. If the desired word is not displayed initially, the user can simply press the ‘next’ key until the desired word appears. For example, to type the word “biking,” the user would press the sequence, “2-4-5-4-6-4” (for b-i-k-i-n-g). The most common word for that sequence of key presses is “ailing”, which is displayed to the user. When the user presses the “next” key, the word “biking” appears, and the user presses the “space” key to accept.

XT9 Smart Input is an enhancement that goes a step further than T9, by predicting the word the user intends to type [49]. As users type, not only is the T9 method applied to the word, but in addition to that, a complete word is suggested to the user. As applied to the previous example, after the user types the sequence “2-4-5-4”, meaning “a-i-l-i”, the system predicts the word to be spelled and displays “ailing” as a possible choice. The user can accept this word by pressing the “space” key, or deny this word by pressing the “next” key or continuing to type the word.

2.3.2 LetterWise

LetterWise is a word prediction system that uses a probability of letter sequences to guess the next intended letter. Unlike the dictionary based T9 Text Input system, LetterWise takes less memory and allows entry of unconventional words [37]. Rather than store full words in a database, this method only needs to store word prefixes. LetterWise suggests letters using probabilities based on language behaviors. As users type, the system selects the most common letter to display based on the letters already selected and the prefixes in the database.

2.4 Information Privacy

When it comes to electronic voting systems, information security is a huge issue because of the necessity of voter and ballot privacy. For the system discussed in this research, it is most relevant due to the need of a database in the design. Most systems that incorporate a database in the design communicate with the database over a network. Transferring voter information over a network is not feasible in this case because it exposes vulnerability to ballot tampering. The Voter Confidence and Increased Accessibility Act of 2009 [67], bans wireless devices and Internet connections in voting and tabulating machines [38]. For this reason, it is necessary for the system presented to utilize a local database. The next section discusses the various types of local databases.

2.4.1 Local Databases

SQLite is an open source embedded SQL database engine. It is an in-process library that implements a serverless SQL database engine [3]. One of the good things about SQLite is that it is a local database that has the option of being loaded in memory. It is written in ANSI-C and is compiled by standard C compilers. Being a C programming language-based system is the primary reason for it being disregarded for the design discussed in this research, since the system presented here was written in JAVA.

H2 is another type of open source database engine [24]. It uses a JAVA Database Connectivity (JDBC) driver, and is JAVA based, which enables it to be incorporated easily into programs written in JAVA. H2 can be run from disk space or in memory, making it a potential candidate for use in secure voting systems. This database was not chosen for the predictive system because it was not robust enough, or as developed as the HSQLDB discussed next.

The HyperSQL Database (HSQLDB) is an open source, SQL relational database engine written in JAVA [28]. HSQLDB also has a JDBC driver, and can be loaded in memory for quick access. It has been tested for stability and reliability, and is the fastest SQL relational database engine available [28]. It has the ability to execute almost every SQL command, including join, count, sum, and max. HSQLDB comes standard with a database GUI tool for database management. For these reasons, this database engine was the optimal choice of the reviewed databases for the project design for the name database.

2.5 Alternative Write-In Methods

2.5.1 Inaccessible Methods

There are several voting systems in use today that have designs inaccessible to those who have visual and/or motor impairments. These systems either require a second party, who is the voter's assistant, or multiple parties, who are election officials, to assist with the voting process [71]. If election officials are used as the third party for assistance, two officials must be used from differing parties. For example, one official is a republican and the other is a democrat. The voter indicates to those assisting for whom s/he intends to write-in. The assisting party/parties then record the desired candidate's name on the ballot of the voter.

As of 2008, the majority of voting systems used in the United States are optically scanned paper ballots and direct recording electronic (DRE) systems (see Table 2.5.1). Although the use of these electronic voting systems has become the standard in most election centers, there are areas that continue to utilize paper ballots (see Figure 2.5.1). Many of these systems, both paper and electronic, require the aforementioned method of assistance for writing-in a candidate's name. For paper ballots that require the voter to hand-write their selections, the assistant writes the voter's choice on the ballot. Similarly, with punch card systems, the voter must hand-write the candidate on a ballot envelope, or in some cases, directly on the ballot [55][56][32]. The state with the highest usage of lever machines, New York, provides instructions on how to write-in a candidate using the machine [20][57]. Using the lever machine, according to their instructions, voters are able to open a slot to physically write a candidate's name on the ballot. For electronic ballots, the assistant types-in the voter's choice, whether using a touchscreen keyboard or a physical keyboard.

Type of Voting Equipment	Counties		Registered Voters	
	Number	Percentage	Number	Percentage
Punch Cards	11	0.35	163,023	0.10
Lever Machines	62	1.99	11,363,178	6.72
Hand-Counted Paper Ballots	56	1.80	280,047	0.17
Optically Scanned Paper Ballots	1,836	55.90	94,926,873	56.17
Electronic (DRE) Systems	1,068	34.26	55,142,920	32.63
Mixed	84	2.69	7,124,765	4.22
TOTAL	**3,117**	**100.00**	**169,000,806**	**100.00**

Table 2.5.1 Equipment Reported for the 2008 Elections [11]

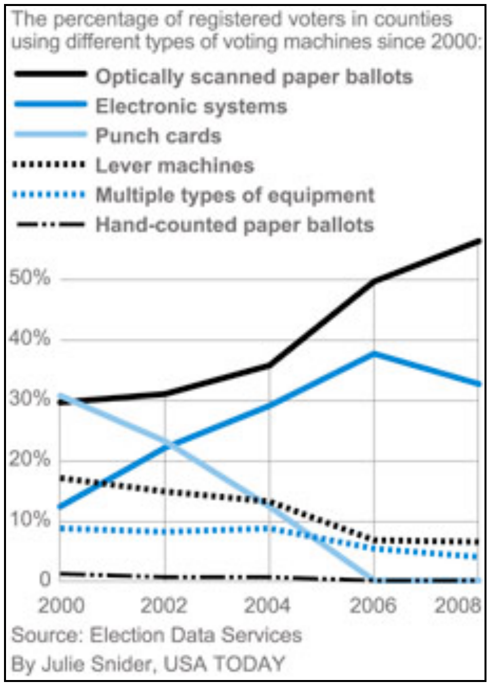

Figure 2.5.1 Voting Technology Usage Trends [69]

2.5.2 Semi-accessible Methods

Due to the need for accessibility in voting, there have been several approaches to the design of accessible write-in voting. The Shoup, AutoMark, and UniLect electronic voting systems provide the option of using headphones to communicate the ballot to the voter. Although these systems use headphones to convey ballot information, the method of input to each system is unique.

2.5.2.1 Shoup Audio Ballot

Although Shoup Voting Solutions, now Advanced Voting Solutions, Inc., is no longer a major force in the election industry [61][4], the company designed a unique approach to accessible voting. Shoup's voting system utilized a touchscreen interface, and in addition, headphones designated to be used by the voter to hear the audio ballot. The touchscreen was divided into four quadrants in order for those who are visually impaired to make selections with the audio ballot. Each of the four quadrants had a specific function. The top left quadrant is used to make a selection, the top right quadrant to go to the next screen or to cast the ballot, the bottom right quadrant to go to the next item in the list, and the bottom left quadrant to go to the previous item in the list. An item can be an office, a candidate, or a referendum question option. Voters using the audio ballot would need to locate these quadrants by following the edge of the frame around the screen in order to make selections. In order to write-in a candidate's name, the voter must tap the bottom right quadrant to enumerate the alphabets and space. To select a letter, the voter taps the top left quadrant of the screen. The top right quadrant is for the voter to confirm the name spelled. [29]

2.5.2.2 UniLect Patriot

The UniLect Patriot Touch-Screen Voting System is a DRE system that utilizes a Freedom Voter Unit for blind and illiterate voters [51]. The Freedom Unit is an additional option that includes headphones, four uniquely shaped selection buttons, and a QWERTY keyboard (see Figure 2.5.2) [51]. The button functions are as follows: a round button to select a candidate, a square button to change a candidate, and two arrow-shaped buttons to navigate the offices on the ballot. Visually impaired voters may utilize the keyboard to write-in a candidate's name. Once the voter has completed typing the desired name, s/he presses the round button to confirm the selection [29].

Figure 2.5.2 UniLect Patriot Freedom Unit Keyboard

2.5.2.3 ES&S AutoMARK

The ES&S AutoMARK Voter Assist Terminal is an optical scan electronic voting system that allows people with various disabilities to vote [22]. The AutoMARK terminal includes a touchscreen interface, headphones to communicate the ballot to the voter, a keypad for input, and a sip/puff tube for input for those who are unable to use the keypad. The keypad consists of 11 buttons; 6 buttons control the output into the headphones, four directional arrow buttons are utilized to navigate the ballot, and a button to make selections. For voters who are unable to use the touchscreen to write-in a candidate's name, they must use the headphones and keypad. For writing-in a candidate, the alphabet is read aloud through the headphones one letter at a time. The voter is able to spell a candidate's name by utilizing the up and down arrows to traverse the alphabet, the select button to choose the current letter, and the right arrow key to confirm (see Figure 2.5.3) [19].

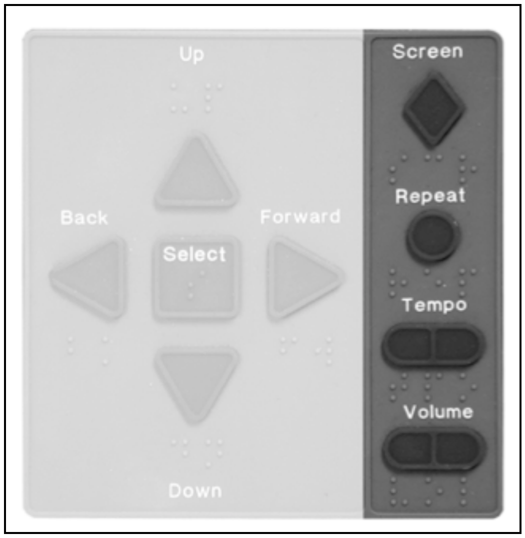

Figure 2.5.3 ES&S AutoMARK Voting Machine Keypad

2.5.2.4 Prime III

Prime III is a research prototype electronic voting system. It is a secure, multimodal electronic voting system that delivers the necessary system security, integrity and user satisfaction safeguards in a user-friendly interface that accommodates all people regardless of ability [54]. The Prime III system is multimodal in that it uses multiple interaction methods. Voters are able to cast their votes with this system through visual interaction and/or through speech interaction; meaning voters can see and touch and/or hear and speak the candidates' names and other options throughout the voting process.

This multimodal approach to electronic voting enables Prime III to incorporate a universal design, as discussed in section 2.2.1, which allows nearly all voters to cast their votes independently and privately. Since Prime III has a universal design, any person who has a visual, cognitive, or motor disability can vote. With Prime III, if a voter has a hearing impairment or disability, s/he can vote using the touchscreen interface (see Figure 2.5.4) to select candidates, navigate the ballot, and to cast and review the final ballot.

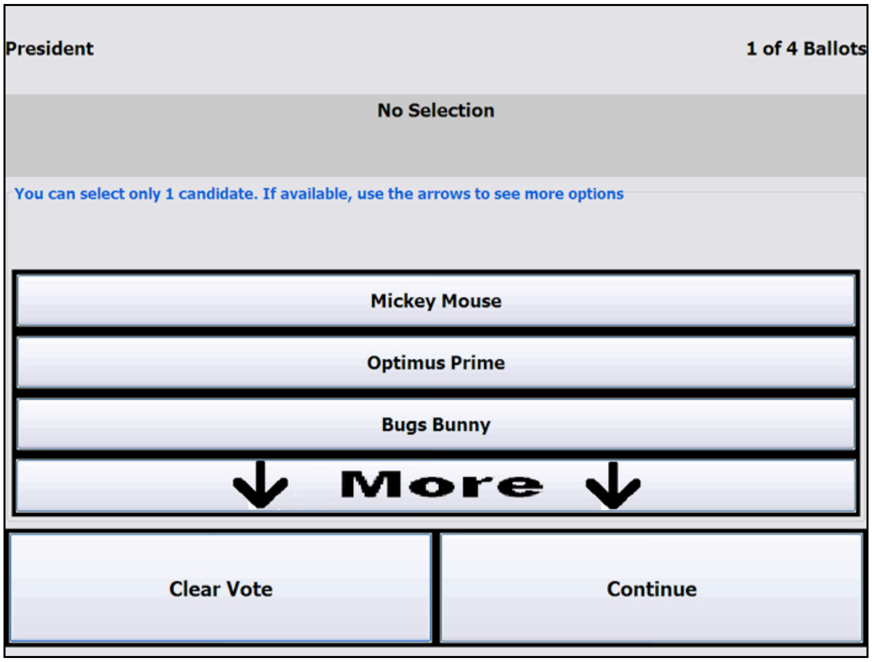

Figure 2.5.4 Prime III Touchscreen interface [54]

Conversely, if a voter is visually impaired or disabled, s/he is able to vote using the speech interface (see Figure 2.5.5) by speaking to the system. Through speech, the voter can select candidates, navigate the ballot, and review and cast the final ballot just as sighted voters are able to do. Alternatively, a voter who may have mild visual, speech, and/or motor impairments may choose to vote using the speech and touch interfaces simultaneously.

Figure 2.5.5 Prime III Speech Interaction Headset

Due to the anonymous nature of voting systems, the candidates that the voter selects must be kept private. Since Prime III integrates speech interaction into the voting process, bystanders may be afforded the opportunity to compromise the privacy of the voter. This presents an issue in voter - ballot anonymity. Bystanders must not be able to hear whom a voter selects for any office, or a voter's decision for any proposition. Therefore, during the voting process, voters cannot simply say the name of the candidates for which s/he wishes to vote. The speech interface of Prime III implements an interaction in which the voter does not need to explicitly verbalize for which candidate they intend to vote.

The Prime III system uses speech to convey the information on the screen to the voter (e.g. candidates listed for a particular office). Each option is presented to the voter in random order, and the system receives input from the voter through speech. When an option is presented, the voter chooses the option by speaking, "Vote," or simply blowing into the microphone. If the voter does not wish to choose the current option, they do not say anything and the system moves on to the next prompt. An example dialogue is as follows:

> Prime III: "To vote for the Democratic Party, say vote <beep>"
> Voter: <says nothing>
> Prime III: "To vote for the Republican Party, say vote <beep>"
> Voter: <says nothing>
> Prime III: "To vote for the Green Party, say vote <beep>"
> Voter: "Vote"

In this example, the voter chose to vote for the green party. With this type of interface, voters make their selections by simply saying, "Vote," or blowing into the microphone. Therefore, instead of a voter's actual choice, bystanders only hear the voter saying "Vote," or hear nothing at all, which ensures the privacy of the voter and the anonymity of the voter's ballot.

The universal accessibility and anonymous nature of electronic voting highlights the incompleteness in the design of writing in a candidate's name with Prime III. Currently, voters have the ability to write-in a candidate's name in one way: using an onscreen keyboard. When a voter chooses not to vote for a predetermined candidate and to write-in a candidate's name, the keyboard is shown, and the user must use the touchscreen to type the candidate's name (see Figure 2.5.6). The next chapter presents the proposed solution to the lack of universal usability in writing-in a candidate's name using the aforementioned systems.

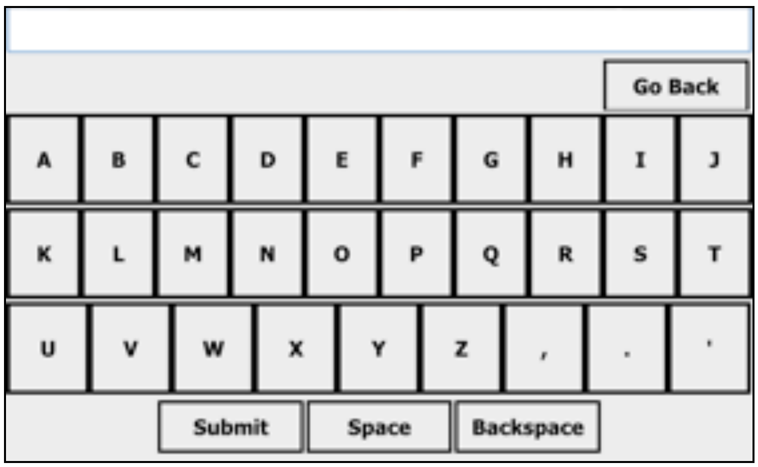

Figure 2.5.6 Prime III Onscreen Keyboard

Chapter 3

System Design

3.1 Design Solution

The most fitting solution to the problem of voter privacy, discussed in section 1.2, is to utilize a multimodal voting system that incorporates speech interaction. With the addition of speech in a multimodal interface, voters, regardless of physical disability, have an option to vote independently. In order to write-in a candidate, a voter could simply speak aloud the name of the person who they intend to write-in. The integration of the speech feature alone enables the system to have a universal design. However, this system is not practical. During election peak times, polling places may have a large voter turnout [52]. With the large number of voters at polling places at any given time, privacy is an enormous issue. In accordance with the EAC, the voting process must preserve the secrecy of the ballot. The voting process should preclude anyone else from determining the content of a voter's ballot, without the voter's cooperation. If such a determination is made against the wishes of the voter, then his or her privacy has been violated [2]. If a voter is required to explicitly say the name of the candidate for which they intend to write-in, any bystanders within the polling place may be able to hear that name, and know for whom that person voted, thereby violating the voter's privacy and ballot anonymity.

In order to secure voter privacy through speech interaction, voters must communicate with the system using the speech interaction method of Prime III. As explained in section 2.5.2, this approach allows a voter to make selections throughout the voting process by simply saying, "Vote" in response to the system's prompts. Using this method for writing in a candidate's name has its challenges. The system cannot simply prompt names to the voter until the system gets to the name the voter intends to write-in. There are an infinite number of names the voter would have to choose from. For example, it would not be viable for the dialogue to be as follows:

Prime III: “To vote for the Bart Smith, say vote <beep>”
Voter: <says nothing>
Prime III: “To vote for the Bill Smith, say vote <beep>”
Voter: <says nothing>
Prime III: “To vote for the Bob Smith, say vote <beep>”
Voter: <says nothing>

.
.
.

If the systems simply made uneducated guesses of the desired name, it would be impossible for the voter to write-in a candidate.

A solution to this problem would be for the voter to spell, rather than say, the desired candidate’s name. However, due to voter privacy, the voter cannot simply spell a name aloud. Spelling a write-in candidate’s name can only be done privately if the Prime III method of getting input data from the voter, through speech, is applied to the design of the system. Using this method, the system would need to prompt the voter to determine the correct letters to spell the desired candidate’s name. This would have to be done for the spelling of the entire name. For example, to spell the name, “Bob,” the dialogue would be as follows:

Prime III: “If the first letter of the candidate’s name is A, say vote <beep>”
Voter: <says nothing>
Prime III: “If the first letter of the candidate’s name is B, say vote <beep>”
Voter: “Vote”
Prime III: “If the second letter of the candidate’s name is A, say vote <beep>”
Voter: <says nothing>
Prime III: “If the second letter of the candidate’s name is B, say vote <beep>”
Voter: <says nothing>
Prime III: “If the second letter of the candidate’s name is C, say vote <beep>”
Voter: <says nothing>

.
.

Prime III: “If the second letter of the candidate’s name is N, say vote <beep>”
Voter: <says nothing>
Prime III: “If the second letter of the candidate’s name is O, say vote <beep>”
Voter: “Vote”

.

.

Prime III: “If the third letter of the candidate’s name is B, say vote <beep>”
Voter: “Vote”

Thus far, this is the best exclusively auditory solution. This approach to spelling a candidate’s name encompasses voter privacy, integrity, and universal accessibility. However, the above example implements a linear search to spell a write-in candidate’s name. For each letter of the candidate’s full name, the voter may have to traverse each of the 26 letters of the alphabet. Spelling using this method would take an extremely long time, especially if the letters of the candidate’s name were at the end of the alphabet (i.e. “Robert Smith”), or if the candidate’s name has several letters (i.e. “Christopher Washington”). Time is a vital factor in voting. Voters want to make their selections and cast their ballots in a reasonable amount of time. The straight linear approach to spell the name of a write-in candidate is long and undesirable.

One method used to assist voters in expediting the time taken to spell a candidate’s name, although not through speech, is for the voter to traverse the alphabet and make selections using a keypad, as presented in section 2.5.2. As the letters of the alphabet are communicated to the voter through the headphones, the voter can press the right arrow to quickly go through the letters. However, a major problem with this method is that persons with motor impairments would not be able to utilize the keypad, and therefore would not be able to use the system without assistance.

Currently, there is no method to spell a name for writing in a candidate that incorporates a universal design and meets the requirements set forth by the EAC; no system allows an individual with visual and motor impairments to spell a candidate's name privately and securely. In order to solve these major issues, a multimodal predictive spelling method was created, including the use of speech interaction. The hypothesis of this work is that the predictive spelling method, will be effective, efficient, and provide user satisfaction.

3.2 Design Overview

The novel approach for writing-in a candidate presented in this research is implemented in a universal design. The approach is usable, private, and time effective. Being that it is multimodal, this approach allows the voter to write-in a candidate's name through speech interaction, touch screen interaction, or switch button interaction. In order to be a more time effective solution for letter and name selection than the linear search method discussed in the section 3.1, the design consists of name prediction and alphabet clustering.

3.2.1 Multimodality

This system is multimodal in that there are various methods in which a voter can write-in a candidate's name. The system produces output via two methods: a touchscreen and headphones. Therefore, if for some reason a voter is unable to use the touchscreen (visual impairment), he or she can hear the ballot through the headphones. In addition to a touchscreen, the system can receive input from a microphone and from switch buttons. Using a touchscreen, a voter can write-in a candidate's name by touching the letters displayed in the onscreen keyboard, as discussed in section 2.5.2.4. Alternatively, when using the speech output option, the voter is able to make selections by speech, via the microphone, or by press, via the switch. The switch utilized in this research has the capability to provide two input measures, one switch button to make selections (as with the speech input), the other to skip speech prompts. To differentiate between the two switch buttons, the surface of one of the buttons is altered to form a rugged surface via a HI MARK (TM) Tactile Pen (black in color). With this alteration, one switch button provides a rugged haptic interaction, while the other switch button remains smooth. Section 3.2.2 provides an overview of the speech output & selection system, on which this dissertation places emphasis.

3.2.2 Speech Output Overview

The speech output through headphones directs the voter to spell a name using alphabet clustering and name prediction. Rather than using linear search to traverse the alphabet, which may take an extensive duration of time to complete, this design breaks down the alphabet into clusters of letters, which are then presented to the voter. The voter then spells a candidate's name by choosing from these letters using the selection method discussed in section 2.5.2.4. During the spelling process, the system performs name prediction similar to the methods used in predictive text technology such as XT9 (discussed in section 2.3.1). Like in XT9, the voters using this system have the option to select from the suggestions made based on the letters spelled. While XT9 utilizes a dictionary database to predict words that the user may intend to type, this system was developed using a database containing names that the voter may intend to spell.

For each letter of the candidate's name, the clusters are presented to the voter for selection using the Prime III interaction method discussed in section 2.5. The voter begins by making the proper selections, through a microphone or switch button, to spell the candidate's last name. The system first prompts the voter with the alphabet clusters. Once the voter selects the desired cluster, containing the first letter of the intended candidate's last name, the system then prompts the voter with the letters contained in that cluster. The voter then chooses a letter, and the system moves on to get the next letter of the desired candidate's name. Once letters have been selected, the name prediction can begin. If the voter does not intend to write-in one of the names suggested, s/he continues the process of selecting clusters, then letters, until the correct name is suggested, or the name has been spelled in full. A visual of this write-in process is depicted in Figure 3.2.1. Further discussion about the depth of the red "Clusters Presented" and the blue "Name Suggested" boxes of this figure are discussed in sections 3.3 and 3.6, respectively.

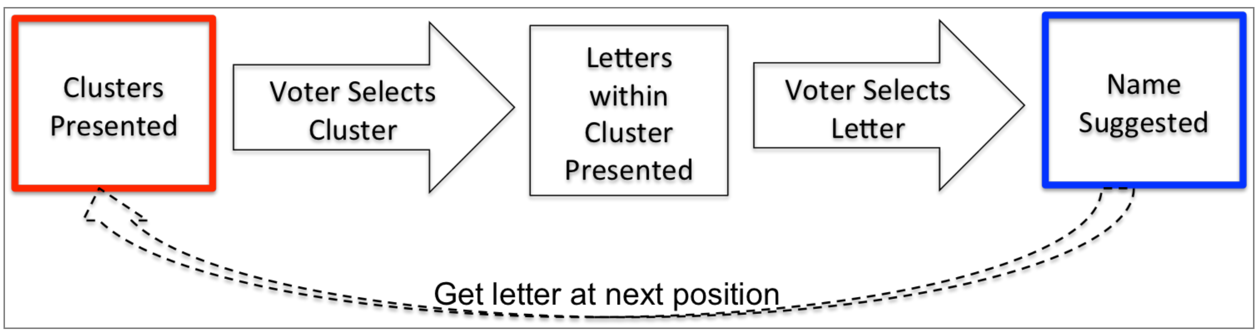

Figure 3.2.1 Auditory Write-in Design Overview

3.3 Cluster Selection

The alphabet is broken down into five standard clusters; four clusters of five letters, and one cluster of six letters (see Table 3.3.1). When selecting the first letter of each of the candidate's names, given name and surname, the voter is first prompted to choose from one of the five standard clusters. The first cluster presented to the voter is chosen at random, with the prompts for the remaining clusters following in alphabetical order, in a round robin fashion. The purpose of this randomization is to secure ballot anonymity by ensuring that bystanders would not be able to decipher for whom the voter voted. The initial cluster is chosen using a weighted random; each cluster may not have an equal chance of being chosen first. The weights for the clusters depend on the letter position of the name being spelled and the names in the database. The cluster weight assignments are discussed in more detail in section 3.5.1. An example of the order in which clusters may be presented is shown in Table 3.3.2.

Cluster Letters
A, B, C, D, E
F, G, H, I, J
K, L, M, N, O
P, Q, R, S, T
U, V, W, X, Y, Z

Table 3.3.1 Standard Letter Clusters

Once the first letter has been selected, the system is able to present a common letter cluster prior to the presentation of the five standard letter cluster prompts (see Table 3.3.2). This common letter cluster consists of the three most common next letters, given the letters previously selected by the voter.

Cluster Type	First Letter Selection	Second Letter Selection
Most Common Letters	-----	R, A, E
Standard	P, Q, R, S, T	F, G, H, I, J
Standard	U, V, W, X, Y, Z	K, L, M, N, O
Standard	A, B, C, D, E	P, Q, R, S, T
Standard	F, G, H, I, J	U, V, W, X, Y, Z
Standard	K, L, M, N, O	A, B, C, D, E

Table 3.3.2 Example Cluster Prompt Order

The most common letter cluster is dependent on the presence of database name matches to the letters already selected. For example, suppose the voter is spelling the intended candidate's last name, and has selected 'C' as the first letter. Of the matching common names in the database, the second letters and the frequencies of names having that second letter, given the first letter is 'C', are as shown in Figure 3.3.1. Therefore, the common letter cluster to be presented to the voter would contain the letters 'A', 'O', and 'H', since those are the letters with the three highest name frequencies. The process of presenting a cluster prompt is depicted in Figure 3.3.2.

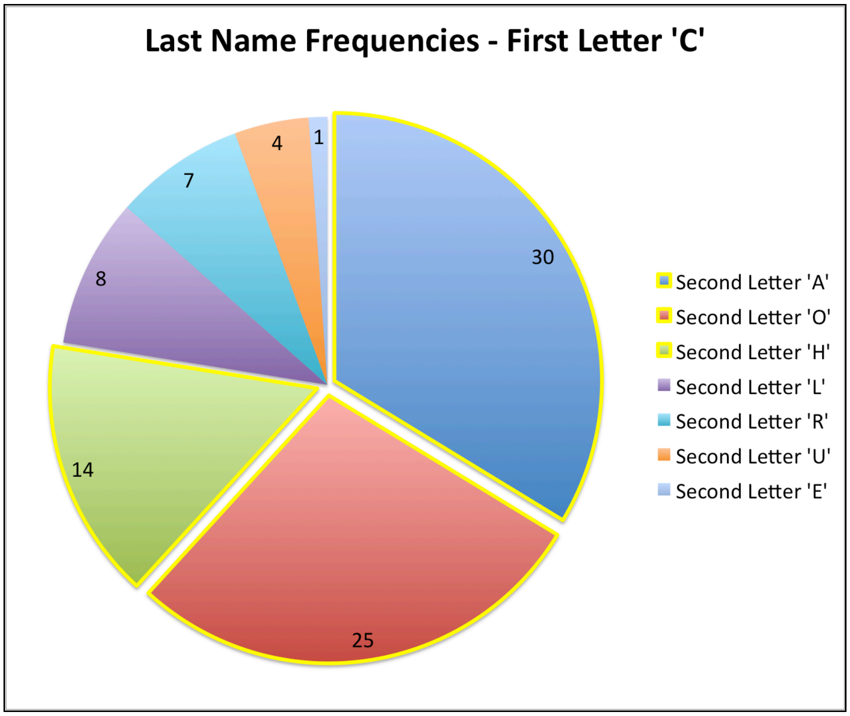

Figure 3.3.1 Common Second Letters - First Letter 'C'

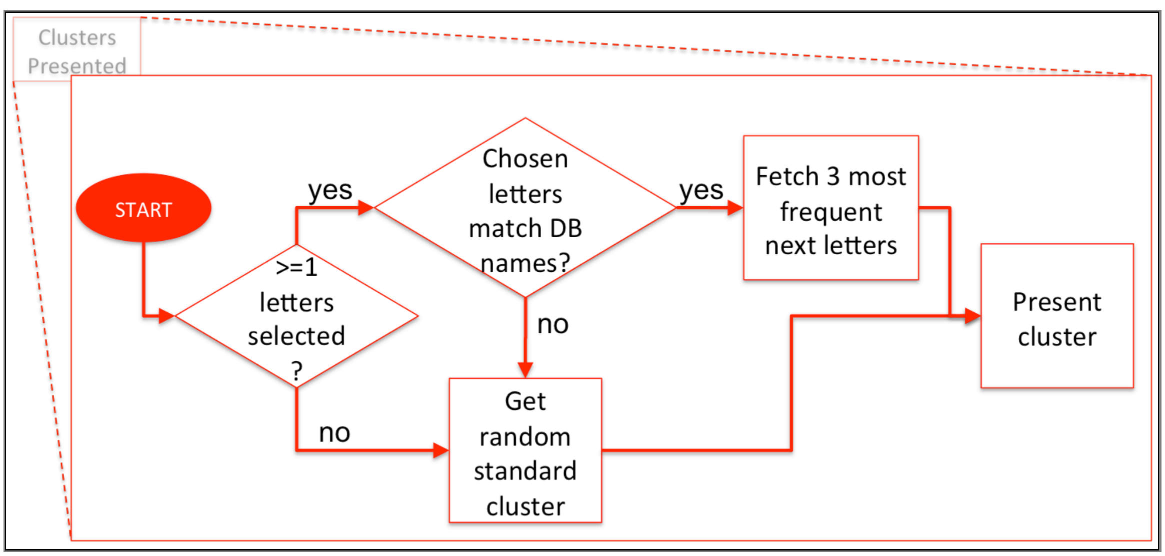

Figure 3.3.2 Auditory Write-in Cluster Prompt Algorithm

These common letter clusters expedite the selection process since the voter is able to make selections at this point, rather than potentially traversing the 5 standard letter clusters. If the next letter of the name is *not* in the most common letter cluster, the voter is then prompted to select one of the five standard clusters (see Table 3.3.2).

3.4 Letter Selection

Once the voter selects the correct cluster containing the next letter of the desired candidate's name, s/he is prompted to choose from those letters. The letters presented by the system are dependent on the cluster the voter selected (see Table 3.6.2). If the voter selects the cluster of letters {A, B, C, D, E}, s/he is prompted to choose from those letters within that cluster. If the voter selects the cluster of the most common letters, for example, {R, A, E}, s/he is prompted to choose a letter from that common letter cluster. Once the desired letter is chosen, the system moves on to the set of prompts for the voter to select the next letter of the write-in candidate's name.

3.5 Name Database

The prediction system for writing in a candidate's name is made possible through the use of a local database of names. The database contains two types of names; common names and names that have the highest probability of being written-in. The most common names were acquired from the 1990 United States census; each name with a given popularity rank. These names include surnames, male given names, and female given names. Within each group of common names, the names that were used most frequently are ranked high, while the names infrequently used have a lower rank. The database used in this design contains a table of the top 1000 ranked surnames and a table containing the top 1000 ranked male names and the top 1000 ranked female names from the 1990 United States Census [12]. Because there is a single database table for given names, the Census rankings of each name needed to be altered to form a single ranking scheme for both male and female names. This new ranking scheme, combining male and female names into a single list, was based on the percent frequency of name popularity rather than the sole rank. For example, the given names “James” and “Mary” are both ranked first on their respective male and female popularity lists. To determine which name is first on the combined list, their percent frequencies – 3.318 and 2.629, respectively – were compared. In the combined given name database table, because of these frequencies, James is ranked first and Mary is ranked second.

The list of names that have a high probability of being written-in varies from county to county. Each county would have names that would have a higher probability than other counties, depending on the election and election year. There are generally two types of names that can be found in this list. The first is based on write-in results from past elections. The second, as discussed in section 2.1, are the names of those who would be registered to be elected as a write-in candidate for that election (required by certain counties). With the former type, for example, the most common full name written-in for past elections held in a particular county could be “Mickey Mouse,” and therefore, it would be stored in the high probability table. In both cases, these names typically refer to an actual person, and therefore are stored in the database and suggested to the voter as full names (given name and surname together). The feature of having names stored in this table also eliminates the write-in spelling issues discussed in section 1.2.1; registered candidates’ names may not be included in the common names tables, but would still be suggested to the voter if in the high probability table.

The relational schema for the name database is shown in Figure 3.5.1. Common names from the 1990 U.S. Census are stored in the “firstNames” and “lastNames” tables. Let us take the surname “Jones” as an example. “Jones” is ranked as the fourth most common surname according to the 1990 U.S. Census. For Jones, the “lastNames” table in the database would have a record where name=’Jones’ and ranking=5. The “lastNameSpellings” table in the database would have five records for the name “Jones”, for each of the letters in the name. The records would be: ranking=5, name=’Jones’, position=0, and letter=’J’; ranking=5, name=’Jones’, position=1, and letter=’O’; ranking=5, name=’Jones’, position=2, and letter=’N’; ranking=5, name=’Jones’, position=3, and letter=’E’; ranking=5, name=’Jones’, position=4, and letter=’S’. Male and female given names are also stored in the database in this fashion.

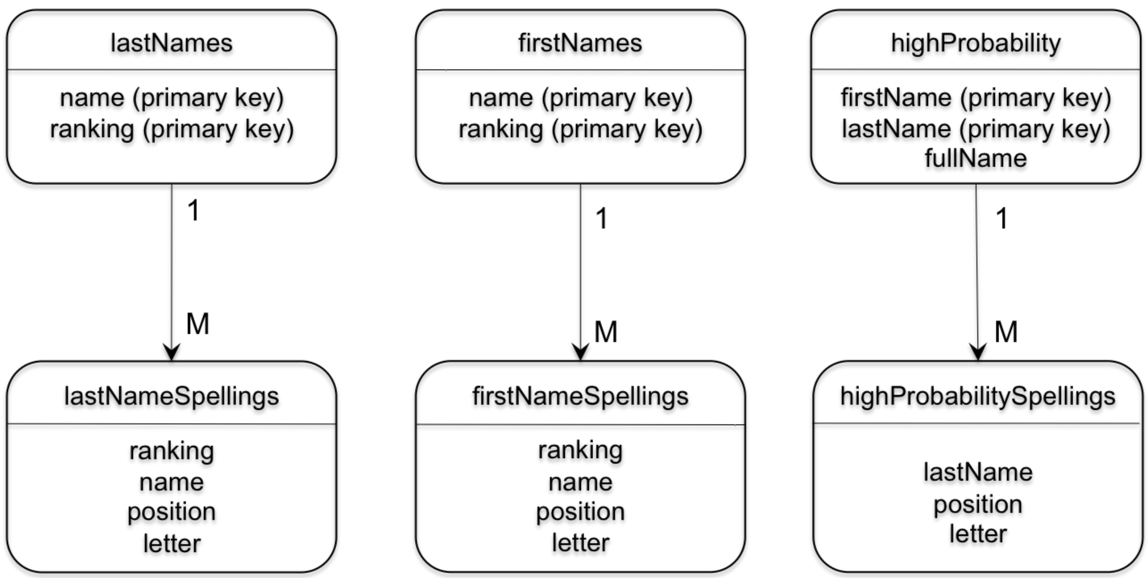

Figure 3.5.1 Name Database Schematic

Names that have a high probability of being written-in are stored in the “highProbability” table. If “Thomas Jones” is registered as a write-in candidate for a given election, the “high Probability” table would have a record where firstName=”Thomas,” lastName=”Jones,” and fullName=”Thomas Jones.” Because the last name is spelled first in the design (see section 3.6.1), the database only needs to store the spelling of the last name to make full name suggestions to the voter. Therefore, the “highProbabilitySpellings” table would have a record for “Thomas Jones,” where name=”Jones,” position=0, and letter=’J’; name=’Jones’, position=1, and letter=’O’; name=’Jones’, position=2, and letter=’N’; name=’Jones’, position=3, and letter=’E’; name=’Jones’, position=4, and letter=’S’.

3.5.1 Weighted Random Clusters

As introduced in section 3.3, the first cluster presented to the voter is chosen at random. This randomization is for privacy purposes when spelling a name. It is weighted in order to increase efficiency and usability while maintaining a private nature. The weights are dependent on which position of the name is being spelled, as well as the names in the database. By default, each cluster has a weight of 1/5, giving it a 20% chance of being selected for a given position. Those weights are increased or decreased based on the names in the database. For instance, suppose the database contained the records, shown in Table 3.5.1, in the “lastNameSpellings” table, and the voter had already selected the first letter. The first standard letter cluster presented to the voter (after the common letter cluster) has a greater chance of being the “K-L-M-N-O” or “A-B-C-D-E” cluster than the other clusters. The cluster weights for the random selection would be: “A-B-C-D-E”, 25%; “F-G-H-I-J”, 12.5%; “K-L-M-N-O”, 37.5%; “P-Q-R-S-T”, 12.5%; “U-V-W-X-Y-Z”, 12.5%. Given the same database records, suppose another voter has already selected the first 5 letters, and is currently selecting the 6th letter of the intended candidate’s last name. In this case, each of the five standard clusters would have an equal weight, because name records with greater than 5 characters do not exist in the database.

Ranking	Name	Position	Letter
1	JONES	2	O
2	ADAMS	2	A
3	MOORE	2	O

Table 3.5.1 Random Cluster Example Database Records

3.6 Name Prediction

In order to effectively reduce the amount of time a voter spends to write-in a candidate's name, this system utilizes a name prediction method built on the name database described in the previous section. The names suggested are fetched from the name database depending on the letters already chosen by the voter. If one of the predicted names is correct, the voter does not need to go through the entire spelling process.

This section describes in detail when during the write-in process name predictions are made, the method in which the names are suggested, and the efficiency of the predictive approach. The section closes with a summary of the prediction, including a diagram illustrating the flow of the prediction and a full example of writing in a name.

3.6.1 Prediction Timing

There are various aspects in the timing of the name predictions. First, name predictions can be presented to the voter after both alphabet cluster selections and letter selections. Second, given the large number of names in the database, name predictions are held until the system gains some knowledge of the name from the voter. Third, since the high probability names have a greater chance of being written in than the common names, those names are given preference in the predictions. Lastly, the names that have a higher probability of being written-in are suggested solely during the spelling of the candidate's last name.

The predictive approach relies on selections from the voter to make name suggestions. Due to this design, the system has the ability to make suggestions to the voter after it receives any information. Therefore, when the voter selects a letter, the system can make a determination based on the letters selected. Likewise, when the voter selects a cluster, the system can base its prediction on the letters already selected and the letters within the cluster selected. For example, if the voter has already selected the letters, “H-A-R”, then selects the “P-Q-R-S-T” cluster, the system can suggest HARRIS as the candidate’s last name. In this case, the system would search the database for names starting with “H-A-R”, and have ‘P’, ‘Q’, ‘R’, ‘S’, or ‘T’ as their fourth letter.

For every position in the intended candidate’s name, the voter selects a cluster followed by a letter. Due to the broad number of names in the database, making predictions after each selection may substantially increase the duration for which a voter is to spell a name. This increase would be especially large if there are several names in the database with the same prefixes (i.e. last names Baker, Bailey, Barnes, Banks, Barnett, Barrett, Bates, etc.). Therefore, predictions are held until the voter has provided sufficient information about the name intended to be written-in (i.e. selected a certain number of letters) to make an educated suggestion.

In order to determine what information was sufficient to make a name prediction, the probabilities of a correct prediction, based on the number of letters selected, were determined. Based on the common names in the database (provided the name clustering of section 3.6.2 is used), we can determine the probability that the intended database name is included in the first three names predicted to the voter. For instance, there are 157 records in the database for given names that start with the letter A. If the voter selects the letter ‘A’, and three predictions are made, the system has a 3/157, or 1.91 percent chance of suggesting the correct name from the database. Likewise, there are 13 given names in the database that begin with the letters, “AD”. If these letters are selected, and the system makes three name predictions, there is a 23.08 percent chance of one of those names being the correct name from the database. Finally, with five names in the database that start with the letters, “ADL”, there is a 60 percent chance that one of first three names predicted is of the proper database record. For each three-letter combination that had name records in the database, these probabilities were averaged, resulting in the overall probability that the intended database record would be among the first three names predicted. In this case, for given names, the probability of a correct prediction is 89.68 percent. Similarly, for a candidate's last name, the probability that one of the first three names predicted is correct once three letters have been selected is 97.31%. In summary, this means that the system's prediction accuracy is higher once the voter has selected the first three letters of the intended write-in name (see Table 3.6.1). Therefore, three letters selected by the voter is a sufficient amount of information for the system to make an educated name prediction.

Name Type	Number of Letters Selected	Probability 1st Prediction Correct (3 name cluster)
Given	1	15.93%
Given	2	52.62%
Given	3	89.68%
Surname	1	26.18%
Surname	2	69.23%
Surname	3	97.31%

Table 3.6.1 Information Based Name Prediction Accuracy

The third aspect in the prediction timing is the priority of the types of names predicted. As the "high probability" names are more likely to be written in, they have a suggestion priority over "common" names. Thus, matching "high probability" names are presented to the voter before matching "common" names. As the voter is selecting letters, once the matching "high probability" names have been exhausted, matching "common" names can be suggested. For example, if the voter has selected the letters "M-O-R", the system would potentially suggest the names "Joe Morgan" and "Jane Morton" to the voter before suggesting the common last name "Morris".

Common names are suggested during the spelling of both the given name and the surname. This is possible because those names are individually represented in the database.

However, names that are stored with a high probability of being written-in are represented as a package, with given and surnames together. The predictive system prompts the voter to first spell the candidate's surname, followed by the spelling of the candidate's given name. Because the names are presented as a package, and this ordering of the spelling system, these highly probable names are suggested solely during the spelling of the candidate's last name.

3.6.2 Prediction Presentation

For the common name predictions, the system provides the voter the option of selecting the most common (highest ranked) name from the database. If the selections made by the voter match a record in the database, that name (given name or surname) can be suggested to the voter. For the probable name predictions, the first name and surname are presented to the voter simultaneously. If the first initial of a surname in the database matches the corresponding cluster or letter selected, the system suggests the name to the voter.

Predicted names are suggested to the voter as clusters, similar to the manner in which letters are presented (see section 3.3). If there are names in the database beginning with the letters already selected by the voter, a maximum of three having the highest ranking are suggested to the voter in cluster form. Highly probable names are not stored in a ranking system; however, due to the small number of these names in the database, there are generally less than five matches once three letters of the last name have been selected.

Suppose, for example, that the voter has selected the letters, "N-O-R". Let's assume the "highProbability" database table contains the following records:

firstName = "Jim"	lastName = "Norris"	fullName = "Jim Norris"
firstName = "Bob"	lastName = "Norton"	fullName = "Bob Norton"

The name cluster prompt would be presented as follows: “Say vote if the candidate’s full name is Jim Norris, spelled J-I-M, N-O-R-R-I-S, or Bob Norton, spelled B-O-B, N-O-R-T-O-N.” Likewise, if matching names in the “highProbability” table did not exist, the system would default to suggesting common names, for which the “lastNames” database table had the following records:

name = “Norris”	ranking = 309
name = “Norman”	ranking = 396
name = “Norton”	ranking = 466

The system would speak the following prompt: “Say vote if the candidate’s last name is Norris, spelled N-O-R-R-I-S, Norman, spelled N-O-R-M-A-N, or Norton, spelled N-O-R-T-O-N.” If the voter selects a name cluster, the system then suggests each name of the cluster to the voter individually. If the voter does not select the name cluster, the system proceeds for the voter to continue the spelling of the intended name. This clustering approach reduces time taken to complete the spelling if the system were to suggest undesired names individually. Each name is suggested to the voter only once, to prevent repeated suggestions of the same name.

In the previous Norris-Norman-Norton example, each name was individually spelled within the prompt. Names are spelled aloud to the voter for clarity among homophones in the database. Several groups of names that are pronounced the same by the speech synthesizer may be present in the database. A method to distinguish these names individually is to spell them to the voter. For example, stored in the database are the three first names Sonya, Sonia, and Sonja, each pronounced SOAN-yuh (or SOAN-yah) phonetically. If the voter has selected the letters S-O-N, without spelling the names aloud, the system would speak the prompt, “Say vote if the candidate’s first name is Sonya (SOAN-yuh), Sonia (SOAN-yuh), or Sonja (SOAN-yuh).” This prompt poses an obvious lack of clarity that may confuse the voter. Instead, the names suggested are reiterated letter by letter, and the previous prompt reads, “Say vote if the candidate’s first name is Sonya, spelled S-O-N-Y-A, Sonia, spelled S-O-N-I-A, or Sonja, spelled S-O-N-J-A.”

As one could imagine, spelling a name in this private manner may place a heavy cognitive load on the voter. This load may increase, depending on the existence of the intended name in the database and the length of the intended name. The cognitive load during the write-in process is due to the memory recall of the letters of the name, and the positions of those letters. To reduce the cognitive load, from the selection of the third letter onward, the letters that have already been selected by the voter are reiterated before the selection of the next letter. Suppose the voter has already selected the letters ‘W’ and ‘A’, and is currently selecting the third letter. The system prompt prior to the cluster presentation would read, “You are now selecting the third letter of the candidate’s last name. You have already selected the letters W, A. Say vote if the third letter of the candidate’s last name is A, B, C, D, or E.” The system prompt for letter selection within the prompt would read, “You have already selected the letters W, A. Say vote if the third letter of the last name is A.” From the third letter until the last letter is selected (or the name is predicted), the previously selected letters are repeated to the voter before each cluster prompt and before each individual letter prompt.

3.6.3 Efficiency

This section presents an preliminary analysis of the efficiency of the write-in system’s speech interaction. Best and worst-case scenarios are highlighted, highly probable predictions are analyzed, and an analysis of the system as a whole is discussed.

In a best-case scenario, the voter would only need to select up to three letters. The first system predicted name would be the name the voter intends to write-in, or the intended name is three letters or less. This best-case spelling would be as follows: the first cluster presented contains the first letter of the name (20% chance); the second and third letters are presented within the most common letter cluster (probability depends on the previous letters selected); the correct name is presented in the first name cluster (if name is in the database, 89.7% chance for given names; 97.3 for surnames). For example, the best-case prompt order for the spelling of the first name "Frank," would be the following:

- Present cluster F-G-H-I-J
  - Present letter F (1st letter)
- Present common letter cluster R-E-A
  - Present letter R (2nd letter)
- Present common letter cluster A-E-I
  - Present letter A (3rd letter)
- Present name cluster Frank – Frances – Francis
  - Suggest name Frank

The worst-case scenario for a name in the database differs from the worst-case scenario for a name not included in the database. For a name in the database, the worst-case scenario would be that it is the lowest ranked match, and that desired letters are presented in the last clusters. For example, the worst-case prompt order for the last name "Carney" would be as follows:

- Present cluster F-G-H-I-J
- Present cluster K-L-M-N-O
- Present cluster P-Q-R-S-T
- Present cluster U-V-W-X-Y-Z
- Present cluster A-B-C-D-E
  - Present letter A
  - Present letter B
  - Present letter C (1st letter)
- Present common letter cluster A-O-H

    - Present letter A (2nd letter)
- Present common letter cluster R-S-L
    - Present letter R (3rd letter)
- Present high probability name cluster Tim Carter – Vern Carlson
- Present common letter cluster R-N-V
    - Present name cluster Carroll – Carr – Carrillo
    - Present letter R
    - Present letter N (4th letter)
        - Suggest name Carney

For names without a record in the database, the worst-case scenario is that the voter must fully spell the name. In this scenario, in spelling a name not in the database, the voter may potentially be presented names that are not desired. For example, a worst-case prompt order for the first name "Sanji" would be the following:

- Present cluster U-V-W-X-Y-Z
- Present cluster A-B-C-D-E
- Present cluster F-G-H-I-J
- Present cluster K-L-M-N-O
- Present cluster P-Q-R-S-T
    - Present letter P
    - Present letter Q
    - Present letter R
    - Present letter S (1st letter)
- Present common letter cluster H-A-T
    - Present letter H
    - Present letter A (2nd letter)
- Present common letter cluster N-L-M
    - Present letter N (3rd letter)
- Present name cluster Sandra – Sandy – Santiago
- Present common letter cluster D-T-F

- Present cluster K-L-M-N-O
- Present cluster P-Q-R-S-T
- Present cluster U-V-W-X-Y-Z
- Present cluster A-B-C-D-E
- Present cluster F-G-H-I-J
    - Present name cluster Sanford – Sang
    - Present letter F
    - Present letter G
    - Present letter H
    - Present letter I
    - Present letter J (4th letter)
- Present cluster K-L-M-N-O
- Present cluster P-Q-R-S-T
- Present cluster U-V-W-X-Y-Z
- Present cluster A-B-C-D-E
- Present cluster F-G-H-I-J
    - Present letter F
    - Present letter G
    - Present letter H
    - Present letter I (5th letter)

Fortunately, these worst-case scenarios have a very small chance of occurring. It has been shown that voters who choose to write-in a candidate for a position tend to write-in the name of a registered write-in candidate, and therefore the name would be stored as a highly probable name in the database. For this reason, the previous worst-case scenarios are highly unlikely. Let us study the case of the 2010 Alaska general election for the write-in results of the U.S. Senator race [60]. For this race, there were five candidates on the ballot, and 160 candidates registered as write-in candidates. Of the 103810 total write-in votes counted, 99.42 percent of the votes were for the 165 total candidates; 99.39 percent were for registered write-in candidates. 1.96 percent of the names written-in may have been for the write-in candidate Murkowski, but were thrown out due to lack of clarity in establishing voter intent. Including high probability name predictions in the write-in process *may* increase the duration of time needed to spell individual last names, but the previous case shows that this potential time taken is efficient due to the fact that more than 99 percent of write-in voters intended to write-in a candidate who would have a record in the high probability database table. Therefore, less than one percent of the names written-in *may* result in the aforementioned worst-case scenarios.

3.6.4 Prediction Summary

Summarizing the features of the name predictions, name suggestions are strictly based on the clusters and letters chosen by the voter. There are two methods in which a name can be suggested to the voter: highly probable name suggestions for registered candidates with records in the "highProbability" database table; and common given name and surname suggestions from the "firstNames" and "lastNames" database tables. High probability names have prediction priority over common names, and as such, common names are only suggested to the voter once undesired matching high probability names have been exhausted. Name suggestions may be presented to the voter after cluster and letter selections, only after three letters have been selected. Also at this point in the process, previously selected letters are spoken to the voter to reduce the cognitive load. Each name is suggested only once, to prevent reiterating the same suggestions to the voter. A visual summary of the name prediction operation is shown in Figure 3.6.1. A complete dialogue for the spelling of the full name, "Kathleen Carpenter," can be found in Table 3.6.2. A complete dialogue for the spelling of the full name of the Alaska Senator race registered write-in candidate, "Marsha Markstrom," can be found in Table 3.6.3. The complete list of full names in the "highProbability" database table for this example is listed in Appendix 6.

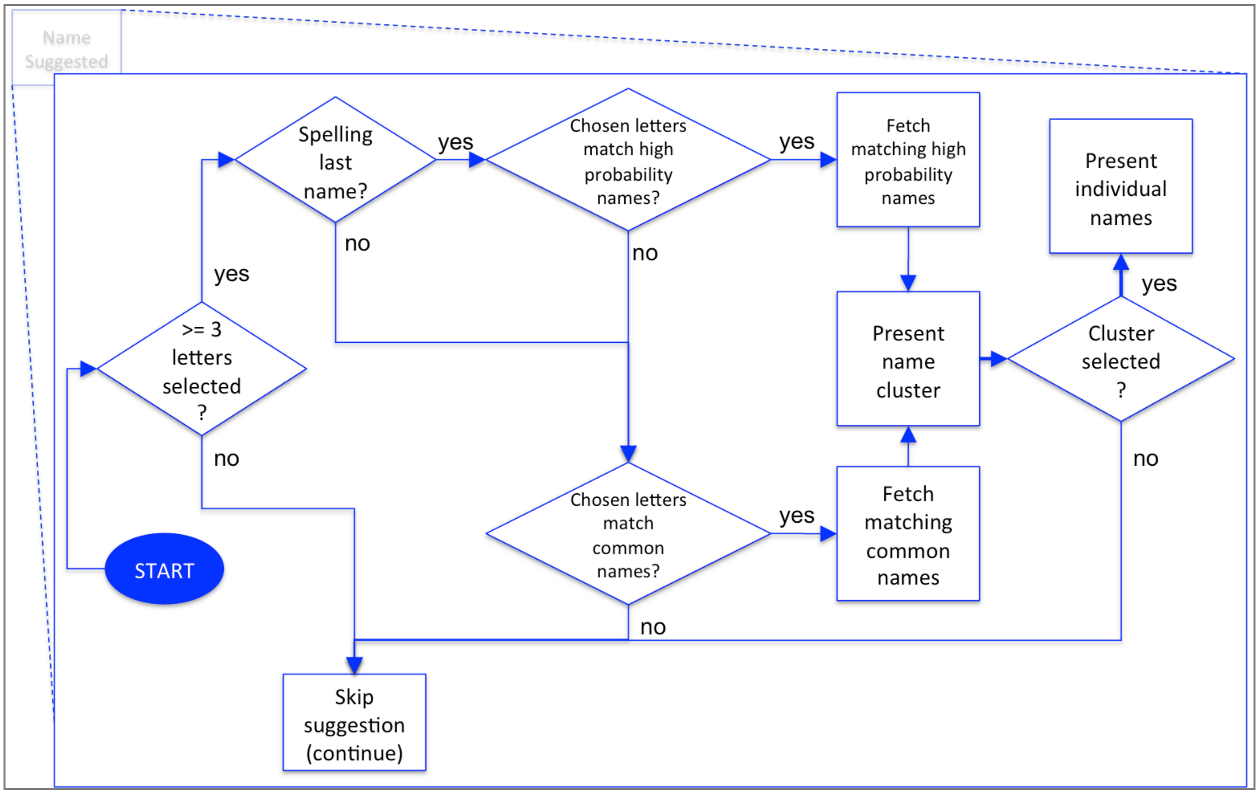

Figure 3.6.1 Auditory Write-in Name Prediction Algorithm

Interaction Mode	Interaction	Letters Already Selected
System	Say vote if the first letter of the candidate's last name is A, B, C, D, or E	--
Voter	*Vote*	--
System	Say vote if the first letter of the candidate's last name is A	--
Voter	*<says nothing>*	--
System	Say vote if the first letter of the candidate's last name is B	--
Voter	*<says nothing>*	--
System	Say vote if the first letter of the candidate's last name is C	--
Voter	*Vote*	C
System	You have selected the letter C. Say vote to delete this letter.	C
Voter	*<says nothing>*	C
System	You have selected C as the candidate's last name. Say vote if you are finished spelling the last name.	C

Voter	*<says nothing>*	C
System	The next letters are the most common letters. Say vote if the second letter of the candidate's last name is A, E, or O	C
Voter	*Vote*	C
System	Say vote if the second letter of the candidate's last name is A	C
Voter	*Vote*	CA
System	You have selected the letter A. Say vote to delete this letter.	CA
Voter	*<says nothing>*	CA
System	You have selected CA as the candidate's last name. Say vote if you are finished spelling the last name.	CA
Voter	*<says nothing>*	CA
System	You will now select the third letter of the candidate's last name. You have already selected the letters, C-A	CA
System	The next letters are the most common letters. Say vote if the third letter of the candidate's last name is M, R, or T	CA
Voter	*Vote*	CA
System	You have already selected the letters, C-A. Say vote if the third letter of the candidate's last name is M	CA
Voter	*<says nothing>*	CA
System	You have already selected the letters, C-A. Say vote if the third letter of the candidate's last name is R	CA
Voter	*Vote*	CAR
System	You have selected the letter R. Say vote to delete this letter	CAR
Voter	*<says nothing>*	CAR
System	Say vote if the candidate's full name is Tim Carter, spelled T-I-M, C-A-R-T-E-R, or Vern Carlson, spelled V-E-R-N, C-A-R-L-S-O-N	CAR
Voter	*<says nothing>*	CAR
System	You will now select the fourth letter of the candidate's last name. You have already selected the letters, C-A-R	CAR
System	The next letters are the most common letters. Say vote if the third letter of the candidate's last name is L, P, or S	CAR
Voter	*Vote*	CAR
System	Say vote if the candidate's last name is Carpenter, spelled C-A-R-P-E-N-T-E-R, Carlson spelled C-A-R-L-S-O-N, or Carson spelled C-A-R-S-O-N	CAR
Voter	*Vote*	CAR
System	Say vote if the candidate's last name is Carpenter	CAR

	spelled C-A-R-P-E-N-T-E-R	
Voter	*Vote*	CARPENTER
System	You have selected Carpenter as the candidate's last name. Say vote if this is incorrect	CARPENTER
Voter	*<says nothing>*	CARPENTER
System	You will now be prompted to spell the candidate's first name.	--
System	Say vote if the first letter of the candidate's first name is K, L, M, N, or O	--
Voter	*Vote*	--
System	Say vote if the first letter of the candidate's first name is K	--
Voter	*Vote*	K
System	You have selected the letter K. Say vote to delete this letter.	K
Voter	*<says nothing>*	K
System	The next letters are the most common letters. Say vote if the second letter of the candidate's first name is A, E, or R	K
Voter	*Vote*	K
System	Say vote if the second letter of the candidate's first name is A	K
Voter	*Vote*	KA
System	You have selected the letter A. Say vote to delete this letter.	KA
Voter	*<says nothing>*	KA
System	You will now select the third letter of the candidate's first name. You have already selected the letters, K-A	KA
System	The next letters are the most common letters. Say vote if the third letter of the candidate's first name is T, R, or Y	KA
Voter	*Vote*	KA
System	You have already selected the letters, K-A. Say vote if the third letter of the candidate's first name is T	KA
Voter	*Vote*	KAT
System	You have selected the letter T. Say vote to delete this letter	KAT
Voter	*<says nothing>*	KAT
System	Say vote if the candidate's first name is Kathleen, spelled K-A-T-H-L-E-E-N, Katherine spelled K-A-T-H-E-R-I-N-E, or Kathy spelled K-A-T-H-Y	KAT
Voter	*Vote*	KAT
System	Say vote if the candidate's first name is Kathleen, spelled K-A-T-H-L-E-E-N	KAT

Voter	*Vote*	KATHLEEN
System	You have selected Kathleen as the candidate's first name. Say vote if this is incorrect	KATHLEEN
Voter	*<says nothing>*	KATHLEEN
System	You have selected Kathleen Carpenter as the candidate's name. Say vote if this is incorrect	KATHLEEN CARPENTER
Voter	*<says nothing>*	KATHLEEN CARPENTER
System	Your selection of Kathleen Carpenter will be written in for the office of president.	KATHLEEN CARPENTER

Table 3.6.2 Example Dialogue for Writing-In "Kathleen Carpenter" for President

Interaction Mode	Interaction	Letters Already Selected
System	Say vote if the first letter of the candidate's last name is K, L, M, N, or O	--
Voter	*Vote*	--
System	Say vote if the first letter of the candidate's last name is K	--
Voter	*<says nothing>*	--
System	Say vote if the first letter of the candidate's last name is L	--
Voter	*<says nothing>*	--
System	Say vote if the first letter of the candidate's last name is M	--
Voter	*Vote*	M
System	You have selected the letter M. Say vote to delete this letter.	M
Voter	*<says nothing>*	M
System	You have selected M as the candidate's last name. Say vote if you are finished spelling the last name.	M
Voter	*<says nothing>*	M
System	The next letters are the most common letters. Say vote if the second letter of the candidate's last name is C, A, or O	M
Voter	*Vote*	M
System	Say vote if the second letter of the candidate's last name is C	M
Voter	*<says nothing>*	M
System	Say vote if the second letter of the candidate's last name is A	M

Voter	*Vote*	MA
System	You have selected the letter A. Say vote to delete this letter.	MA
Voter	*<says nothing>*	MA
System	You have selected MA as the candidate's last name. Say vote if you are finished spelling the last name.	MA
Voter	*<says nothing>*	MA
System	You will now select the third letter of the candidate's last name. You have already selected the letters, M-A	MA
System	The next letters are the most common letters. Say vote if the third letter of the candidate's last name is R, Y, or C	MA
Voter	*Vote*	MA
System	You have already selected the letters, M-A. Say vote if the third letter of the candidate's last name is R	MA
Voter	*Vote*	MAR
System	You have selected the letter R. Say vote to delete this letter	MAR
Voter	*<says nothing>*	MAR
System	Say vote if the candidate's full name is David March, spelled D-A-V-I-D, M-A-R-C-H, Deanna March, spelled D-E-A-N-N-A, M-A-R-C-H, or Edward Martin, spelled E-D-W-A-R-D, M-A-R-T-I-N	MAR
Voter	*<says nothing>*	MAR
System	You will now select the fourth letter of the candidate's last name. You have already selected the letters, M-A-R	MAR
System	Say vote if the fourth letter of the candidate's last name is K, L, M, N, or O	MAR
Voter	*Vote*	MAR
System	Say vote if the candidate's full name is Marsha Markstrom, spelled M-A-R-S-H-A, M-A-R-K-S-T-R-O-M, or Paul Markstrom, spelled P-A-U-L, M-A-R-K-S-T-R-O-M	MAR
Voter	*Vote*	MAR
System	Say vote if the candidate's full name is Marsha Markstrom, spelled M-A-R-S-H-A, M-A-R-K-S-T-R-O-M	MAR
Voter	*Vote*	MARSHA MARKSTROM
System	You have selected Marsha Markstrom as the candidate's full name. Say vote if this is incorrect	MARSHA MARKSTROM
Voter	*<says nothing>*	MARSHA MARKSTROM

System	Your selection of Marsha Markstrom will be written in for the office of senate.	MARSHA MARKSTROM

Table 3.6.3 Example Dialogue for Writing-In registered candidate "Marsha Markstrom" for Alaska Senator

Chapter 4

Experiment Design

The overall objective of this research is to design a method to write-in a candidate's name that addresses the issues of time, privacy, and accessibility. In order to determine if these issues were met, the design is analyzed and compared against alternate methods of writing-in a candidate's name. An experiment is performed to assess two aspects of the multimodal system: speech interaction and switch interaction. The experiment is designed to thoroughly evaluate the efficiency, effectiveness, and usability of the design implementation.

The remainder of this chapter is organized as follows. Presented first are the preliminary research findings and conclusions on the initial implementation studied, along with modifications made to the initial implementation to improve its efficiency. A description of the hypothesis of this research is also provided. Described next are the volunteered participants and the materials used in the experiment. Experimental provisions of the experiment conducted follow, detailing the design and procedure. Finally, the chapter closes with the data collected and an analysis of the data results.

4.1 Preliminary Findings

The initial design of the predictive method was implemented and tested to show proof of concept. The results from this preliminary work served as a basis for design decisions, and a direction for the final implementation. This section discusses the features of the initial implementation, the results from the preliminary experiment, and the modifications made to improve the design upon completion of the experiment analysis. The section concludes with a discussion of how the preliminary results analysis was applied to the statistical plan of the research experiment.

4.1.1 Initial Design Features

Initially, the system was designed as an efficient method for writing-in a candidate's name. The initial system contained letter clustering functionality, and common name prediction functionality for surnames and given names. Also, names were not suggested until the user had selected three letters (as discussed in section 3.6.1). In this initial suggestion phase during the spelling of a name, three suggestions are made sequentially. If those suggestions are incorrect, the system suggests one name after every cluster or letter is selected, if a name match is in the database. The list below is a summary of the functionality of the initial system:

- Name Database
  - 1000 Most Common Surnames
  - 2000 Most Common Given Names
    - 1000 Male Names
    - 1000 Female Names
- Letter Cluster Selections
- Individual Letter Selections
- Name Predictions
  - Made after three letters selected
  - 3 names suggested after the third letter
  - 1 name suggested after every cluster or letter selection following the third letter

4.1.2 Preliminary Experiment

The primary objective of the preliminary study was to test the initial speech interaction predictive write-in system as a proof-of-concept and to observe and analyze how people interacted with the system. The goal of the study was to determine the time it took a voter to use the initial predictive write-in system. The data from this study was analyzed to determine which of two methods, the predictive approach and a linear approach, would be faster for a voter to spell a name. It was expected that the predictive system would perform significantly faster when spelling a name than a linear speech interaction system. Additionally, it was expected that the participants in the study would be able to use the system effectively, meaning they would be able to complete the spelling of a name.

4.1.2.1 Materials

There was a variety of equipment and technology used in this study, including a laptop computer, microphone headset, speakers, and timer. The laptop was a MacBook computer used to run the predictive and linear systems. The software on the MacBook used to run and modify the system code was Eclipse [18]. The speakers were Altec Lansing Technologies and the microphone headset was a Logitech USB headset. A simple stopwatch timer was used to capture the times for the study. The experiment results were analyzed using Microsoft Excel.

4.1.2.2 Participants and Procedure

Various methods were used to recruit participants for this study. The study was advertised through mass email to research labs, word of mouth, and some of those who were Auburn University students were offered extra credit in their courses to participate. Any student, provided they were aged 19 or older, was permitted to participate in the study. A total number of 40 participants participated in this study, 39 of which were undergraduate students. Students receiving extra credit provided their name during the study; this information was not used to identify participant data, and was relinquished to the university course professor upon the completion of the study.

Participants were tasked with spelling one full name, given and surname, using the predictive system. For anonymity, the name spelled could be any name of their choosing, with the exception of their own. The time-to-task was recorded for each participant as the experimental time. The time-to-task was simply the duration of time taken by each participant to spell the name. In order to reduce time strains on the participants, the control time-to-task for this experiment was calculated based on the name chosen by each participant. For example, if the participant chose to write-in the name "Steve Smith" on the experimental system, the corresponding time duration to spell the name on the control system would be calculated. The time-to-task for the control system was calculated as a best-case spelling scenario, and compared against a calculated best-case time-to-task with the experimental approach.

In order to determine how long it would take to spell a name, each interaction cycle of each implementation was isolated and the time for the cycles was recorded. An interaction cycle was considered to be a system prompt followed by an affirmative or negative user response. There is a different sequence of prompts presented to the voter to spell a name for each approach [Appendix 4]. The sequences were determined for each system, and compiled for each name spelled. The sequences for the predictive write-in approach were constructed under the assumption that the names to be spelled were in the system's name database (a best-case scenario).

The following procedure was followed for the experiment:

1. Participants individually signed up to take part in the study during a pre-arranged time slot.
2. Upon arriving at the study location in their designated time slot, each participant agreed to partake in the study after first reading an Institutional Review Board for the Protection of Human Subjects in Research (IRB) stamped information letter. If the participant was a student receiving extra credit for participating in the study, his or her name was recorded on a document separate from the data collection sheet.
3. Participants then completed a pre-questionnaire [Appendix 1] to capture demographic information, and a scenario was given to participants explaining the study. The scenario was to inform the students about the write-in voting process, and to encourage them to treat the study as if it were an actual election.
4. The participants chose a full name for the task and recorded the name on a blank sheet of paper for the researcher. Participants were not permitted to reference this paper during the task completion.
5. Participants then sat at a desk, on which were the laptop and speakers, and were instructed to put on the microphone headset. It was explained to the student that the speech from the system would be coming from the speakers for observational purposes, and that the headset was strictly for the use of the microphone.
6. Participants then completed the task using the experimental system. Prior to leaving the study location, all participants were instructed not to discuss the experiment with friends and classmates to ensure that all participants had an equal knowledge of the study.

7. The best-case time-to-task for each name provided was then calculated and recorded for the control and experimental systems.

4.1.2.3 Data Collection

The first method of data collection for the study was done through the pre-questionnaire [Appendix 1]. Each participant was required to fill out the pre-questionnaire so that demographic information about the participants could be collected. The demographics are informative of what type of disabilities the participants have, if any. It also indicates if English is the participant's native language, their level of education, age, race, and citizenship. Each pre-questionnaire was given a unique identifier so that the information could be paired with the information collected during the experiment.

During the study, information was gathered to analyze the participants' use of the system. A data collection sheet was used to record all information during the study [Appendix 2]. This sheet contained the unique identifier that corresponds with the pre-questionnaire for each participant, the name each participant chose to write-in, and the time taken to spell that name. Also on the data collection sheet was a space for other observations made throughout the study. Best-case times for the experimental and control systems were calculated and stored electronically for analysis. The results of both the pre-questionnaire and time-to-task data are presented in section 4.1.3.

4.1.3 Preliminary Results

This section presents the quantitative data from the pre-questionnaire and the data collection sheet. Also presented are calculated comparisons between the predictive write-in and the linear search approaches.

4.1.3.1 Participant Demographics

The demographic results show that the age range for the participants was 19 to 27, with an average 20.2 years of age (see Figure 4.1.1). As shown in Table 4.1.1, there were 34 males to participate in this study, making up 85% of the participants. Three of the participants listed that they had disabilities; one indicated dyslexia, another indicated loss of hearing in one ear, and another indicated poor vision.

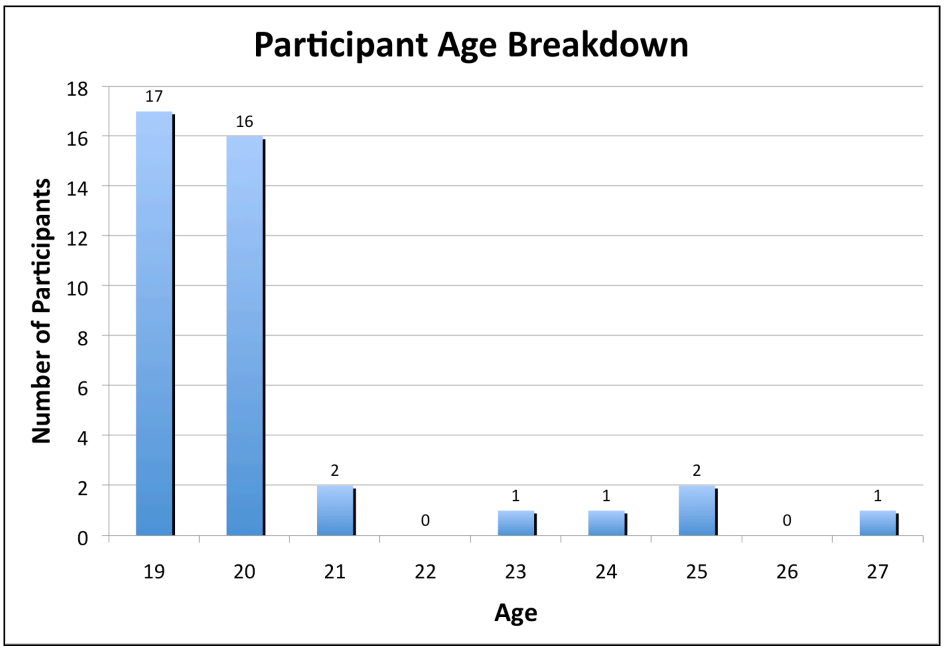

Figure 4.1.1 Participant Demographic Results - Age

	Number of Participants	Percentage of Participants
Male	34	85%
Female	6	15%
Total	40	100%

Table 4.1.1 Participant Demographic Results - Gender

4.1.3.2 Observational Study

Participants were given the task of spelling a full name of their choosing. The average length of the full names chosen was approximately 11 letters (see Table 4.1.2). The shortest full name was 7 letters in length, and the longest full name was 16 letters in length. The average time-to-task was 9.52 minutes, with a standard deviation of 3.83. The average time per letter for the names given was 1.09 minutes, with a standard deviation of 27 seconds. This analysis is shown in Table 4.1.2. Of the 80 given and surnames chosen (one given name and one surname for each of the 40 participants), 71.3% of the names were in the database and suggested to the user. The names chosen, along with the raw time-to-task data, are listed in Appendix 3.

	Time to spell full name (minutes)	**Number of letters per full name**	**Average time per letter (minutes)**
Average	9.52	10.43	1.09
Standard Deviation	3.83	2.22	.45
Median	8.42	10.00	1.04

Table 4.1.2 Predictive Write-In Statistics

Figure 4.1.2 shows a breakdown of the average time-to-task for the participants, based on the number of letters in the full name spelled. The averages in this figure are based on the quartiles for number of letters of the full names used in the task. The first quartile was 10 letters in full name length; the median was 11 letters; the third quartile was 12.5 letters; the maximum full name length was 17 letters. Excluding the extreme values of this chart, these averages show that the time-to-task was fairly consistent regardless of the number of letters in the full name. These results also show that in practice, time-to-task took longer than anticipated (see section 4.1.3.3). Subsequently, modifications were made to the predictive method in order to further increase the efficiency of the design (see section 4.1.4).

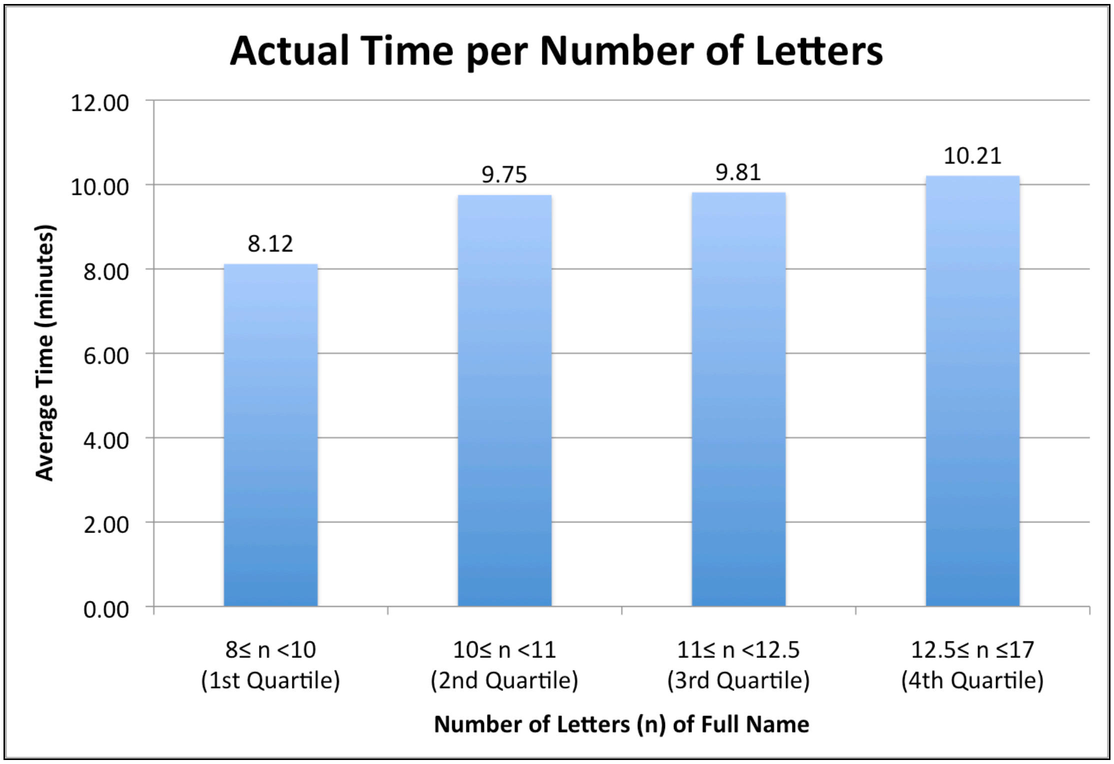

Figure 4.1.2 Average Time to Spell Full Names

4.1.3.3 Best-case Comparison

To compare the predictive and linear approaches to spell a name, without putting time strains on the participants, calculations were made on a best-case basis. These calculations determined the time-to-task for spelling the names provided by participants during study (raw data in Appendix 3). In theory, the predictive approach would only require the participant to select 3 letters of a name, after which the desired name would be suggested to the voter. Therefore, the theoretical best-case time-to-task for the predictive approach was only dependent on the time it takes to select the first three letters. However, if the name provided was not the first one suggested to the voter, the name was not in the database, or the desired letter was not in the common letter cluster, the time-to-task may vary. The actual best-case time was calculated based on the record of names in the database at the time of the study; meaning, the best-possible time a participant could have had given the name chosen.

The average theoretical best-case time-to-task, with the assumption that the name was in the database and was the first name suggested, and the average actual best-case time-to-task for the experimental predictive approach were compared against the best-case time-to-task of the control linear approach, in Figure 4.1.3. The averages in this figure were based on the quartiles for number of letters of the full names used in the task. The first quartile was 10 letters in full name length; the median was 11 letters; the third quartile was 12.5 letters; the maximum full name length was 17 letters.

From this data, it was concluded that in a best-case scenario, the time-to-task for the linear approach was directly related to the length of the name to be spelled; as the length of the name increases, its time-to-task increases. Conversely, the best-case time-to-task for the predictive approach was related to the inclusion of the name in the local database.

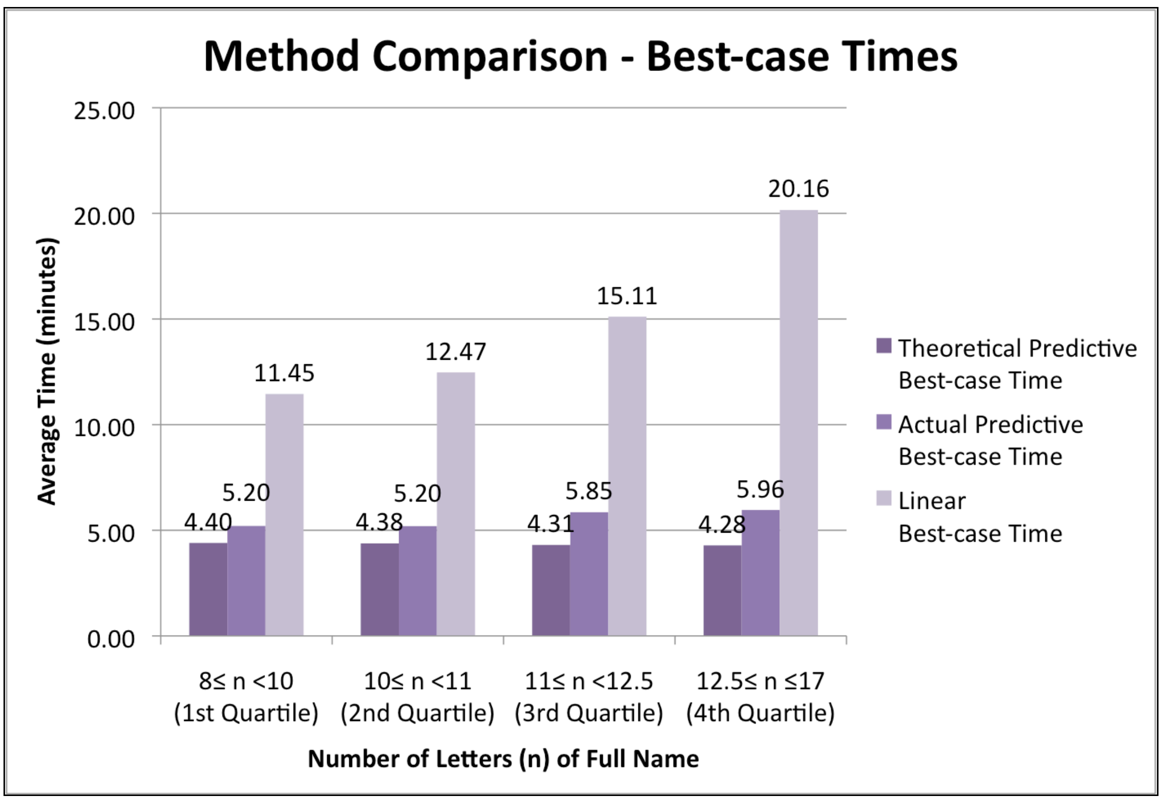

Figure 4.1.3 Best-case Method Comparison of Times to Spell Full Names

The average time-to-task for the full names provided in the study for the calculated best-case linear spelling approach was 15.09 minutes, while the same for the actual calculated best-case predictive spelling approach was 5.62 minutes (see Table 4.1.3). In theory, the best-case average time-to-task for the predictive spelling approach was 4.33 minutes. From these results, it was concluded that at best-case, the average time-to-task for the predictive spelling approach was approximately three times faster than the linear spelling approach. The predictive spelling method was effective in that 100% of the participants were able to complete the spelling of the intended names.

	Predictive Method Theoretical Time-to-task (minutes)	**Predictive Method Actual Time-to-task (minutes)**	**Linear Method Time-to-task (minutes)**
Average	4.33	5.62	15.09
Standard Deviation	0.18	3.87	3.91
Median	4.34	4.79	14.73

Table 4.1.3 Calculated Predictive and Linear Spelling Statistics

Figure 4.1.4 depicts the complete report on the time-to-task averages for the observational study, the theoretical best-cast predictive approach, the actual best-case predictive approach, and the best-case linear approach. The next section discusses the differences in time between the actual observed time and the actual predictive best-case time, and what was altered in the predictive system design to lessen the difference between the two.

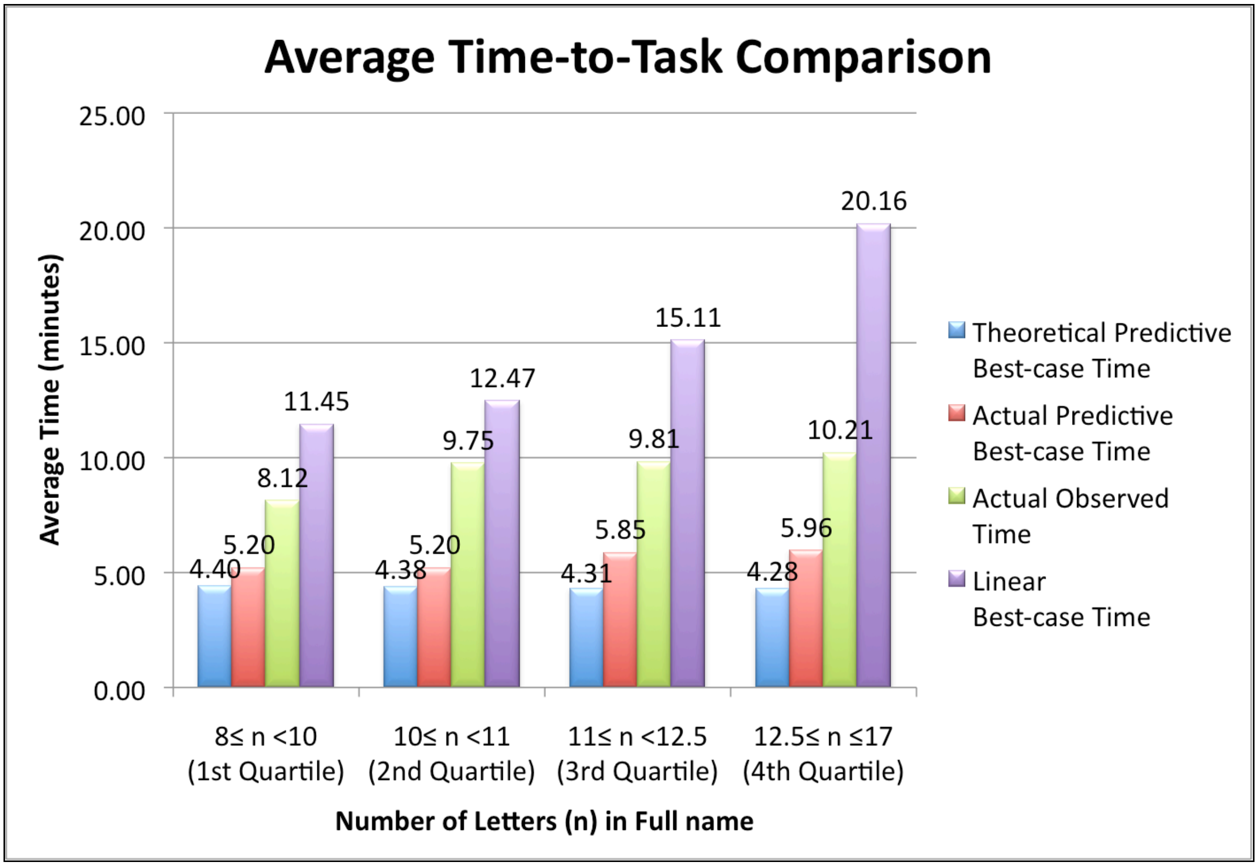

Figure 4.1.4 Preliminary Results - Various Time-to-Task Averages

4.1.4 Design Progression

Due to the large differences in the time-to-task results shown in Figure 4.1.5, several modifications were made to the predictive system. The main goal for these changes was to increase efficiency by decreasing the time a voter would take to spell a desired name. First, the name predictions were modified to be suggested to the voter in clusters rather than individually. Second, during name prediction prompts, the name suggested was also spelled aloud to the voter. Third, a new table was added to the database to include names that had a high probability of being written-in. These changes are included in the system design discussion of Chapter 3.

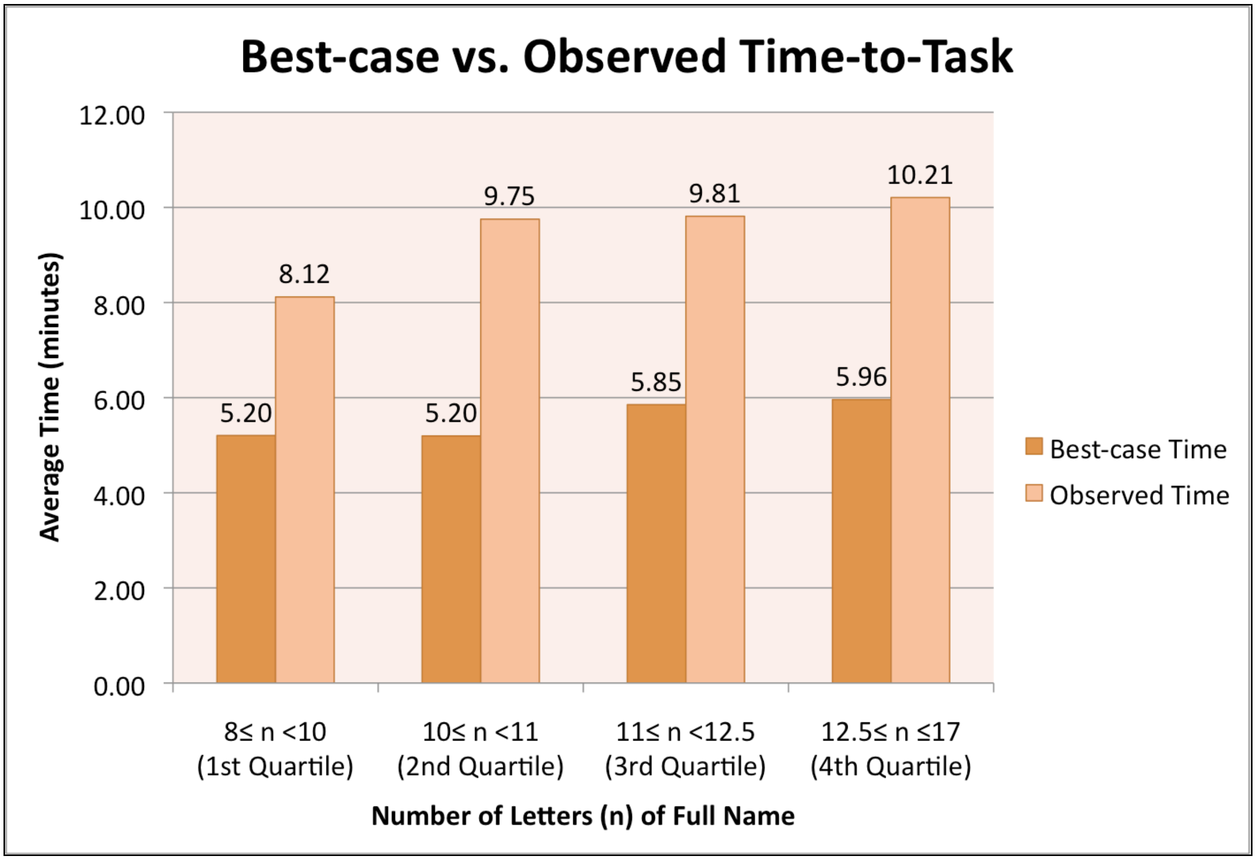

Figure 4.1.5 Best-case vs. Actual Predictive Approach Time-to-Task Averages

Names prediction prompts were modified so that they were suggested in clusters to reduce the interaction cycles in the case that the names suggested were not a name the voter intended to write-in. In the initial design, as discussed in section 4.1.1, three names were suggested individually to the voter in sequential order according to the rank of the name. If the voter had not intended to spell one of those names suggested, unnecessary time would be taken predicting incorrect names. For this reason, the predicted names are suggested in clusters of up to three names, reducing the interaction time if the names are incorrect. Suppose, for example, that the voter wants to spell the last name "Norrison," and has already selected the first three letters, N-O-R. The most common names in the database that match these letters are Norris, Norman, and Norton. Each incorrect name suggestion prompt in the initial design took approximately 5.72 seconds. At this point in the system, each of the three names would be suggested individually, resulting in a total of (5.72*3=) 17.16 seconds until the system can continue on in the spelling of “Norrison.” In the re-designed system, the same three names would instead be presented in a cluster (in addition to the spelling of the names), which has a total approximate duration of 9.46 seconds. This clustering process reduced the prediction time from 17.16 seconds to 9.46 seconds, shaving 7.7 seconds off of the voter’s time-to-task.

The second major modification was made to reduce the selection errors made in spelling a name due to name confusion in the suggestion prompts. The confusion addressed was a result of name homophones in the database. For example, stored in the database are the three first names Sonya, Sonia, and Sonja, each pronounced SOAN-yuh (or SOAN-yah) phonetically. Therefore, if the voter has selected the letters S-O-N, the voter would then hear the prompt (with name clustering), “Say vote if the candidate’s first name is Sonya (SOAN-yuh), Sonia (SOAN-yuh), or Sonja (SOAN-yuh).” This prompt poses an obvious lack of clarity that may confuse the voter. To eliminate name homophone confusion, the names suggested were spelled aloud to the voter, dictating the name letter by letter. The previous prompt would instead now read, “Say vote if the candidate’s first name is Sonya, spelled S-O-N-Y-A, Sonia, spelled S-O-N-I-A, or Sonja, spelled S-O-N-J-A.”

The third major change involves the addition of names to the database in order to further reduce the time-to-task of spelling a full name. This enhancement to the system was not a direct result of the experiment; rather, it was due to the recent events of the US election (see section 1.2.1). The original design included separate tables in the database for the most common US surnames and given names. Added, were tables that consisted of names that were more likely to be written-in. These names are presented to the voter before the common names, to eliminate process of spelling both a surname and a given name if one of them is desired. A detailed description of this feature is presented in section 3.6.

In summary, upon completion of the design progression, the system features are:

- Name Database
    - 1000 Most Common Surnames
    - 2000 Most Common Given Names
        - 1000 Male Names
        - 1000 Female Names
    - Highly Probable Full Names
- Letter Cluster Selections
- Individual Letter Selections
- Name Predictions
    - Made after three letters selected
    - Name Cluster Selections
        - Up to 3 names suggested per cluster
    - Names spelled after suggestion

4.1.5 Statistical Planning

The results from the preliminary study were analyzed to determine the statistical plan for the experimental setup. This statistical analysis is applied to the research experiment in order to determine significance. To perform this analysis, a Java applet for power and sample size was used [36]. The student's t-test is used to determine if the experiment results are statistically significant. With an alpha value of .05, in order to achieve an 80% power ranking for the experiment design discussed in section 4.5, and to show a 3-minute statistical difference of means, a total of 40 participants were needed for the research experiment.

4.2 Hypothesis

The hypothesis of this research, as stated in section 1.2, was that the predictive system will be effective, efficient, and provide user satisfaction. To further expound upon this hypothesis, the study was designed to show effectiveness in that the participants will be able to successfully use the system to spell a candidate's name. In this case, success is defined as the ability of participants to submit the correct spelling of the intended name to the system. Secondly, to measure efficiency, the study was designed to determine if the participants were able to spell a name in a timely manner. Lastly, the participants completed a survey to measure the level of satisfaction they felt using the system. In order to accurately assess these three criteria, the study compared the predictive system with similar methods of accessibly writing-in a candidate's name.

4.3 Participants

The participants in this study were recruited through various methods, including email advertisements and word of mouth. The advertisement stated that the participants would need 45 minutes to an hour to do the study, and how to contact the researcher for more information and to set up a time to participate. All volunteers were accepted to participate in the study, given they were at least 19 years of age. Participants were primarily students at Auburn University, and were offered extra credit in their courses to participate. Ideally, people who have disabilities would be encouraged to volunteer to participate. However, Auburn University has few students with disabilities, and employs strict policies to protect their privacy. A total number of 40 participants participated in this study. In order to reduce unnecessary pressures of time, participants were only tasked with spelling surnames. The methods used by each participant are discussed in section 4.5. The tools used for the experiment are presented in the next section.

4.4 Materials

The study was conducted in the Human Centered Computing Lab of the Shelby Center for Engineering Technology at Auburn University. The system software was run on an Apple Macbook [10]. The following software was installed on the machine for this study:

- Eclipse Integrated Development Environment (IDE) [18]
- USBKeys 2

The database used on the backend was HSQLDB, a SQL relational database engine written in Java [28]. The following hardware was used in this study:

- Altec Lansing Speakers
- Crick USB Switch Box
- AbleNet Rocker Switch
- RadioShack stereo to mono y-adapter
- Macbook Internal Microphone

The experiment results were analyzed using Microsoft Excel and SAS statistical analysis software.

4.5 Experimental Provisions

The participants in this study were randomly divided into two groups, each group having approximately 20 participants. Each group utilized a different aspect of the multimodal system for the experiment. The first group utilized the speech input interaction method, while the second group utilized the switch input interaction method. Within each group, there were two systems, experimental and control, on which two tasks were completed. Due to these configurations, the study had a mixed design of between-subjects and within-subjects. The experimental system was the predictive system discussed in Chapter 3. The control system was designed to guide the user to write- in a name using a linear spelling algorithm. These systems are further discussed in section 4.5.2.1. The tasks performed by each participant are described in section 4.5.1. Table 4.5.1 depicts the experiment design.

Group	Interaction Method	System	Task X	Task Y
1	Speech	Experiment	User Chosen	High Probability
	Speech (Calculated)	Control	User Chosen	High Probability
2	Switch	Experiment	User Chosen	High Probability
	Switch	Control	User Chosen	High Probability

Table 4.5.1 Experimental Design

The independent variables for this study were the system and the method of interaction. The dependent variables were the task completion time, success rate of task completion, and user perception of the systems. The participants utilizing the speech interaction method did not perform tasks on the control system, so that the time strain on the participant was reduced; instead, the time-to-task on the control system will be calculated. For the participants utilizing the switch interaction method, the order in which they use the experimental and control systems was chosen at random, to reduce carryover effects. To represent actual usage scenarios, the switch was not visible to the participants during the experiment.

4.5.1 Tasks

There were two tasks to spell a name for this study. Each participant, regardless of what group of which they were a part, performed both tasks. Which task the participant performed first was chosen at random for each participant, to reduce carryover effects. In the first task, Task X in Table 4.5.1 Experimental Design, the participant is asked to provide any last name of their choosing with the exception of their own last name. The participant is asked to provide the name for this task to eliminate bias by the researcher in choosing names in the database. It was suggested that the name provided be seven letters or less to reduce time stress on the participants.

For the second task, Task Y in Table 4.5.1 Experimental Design, participants were asked to spell a full name from the highProbability database table discussed in section 3.5. The names chosen for the participants were chosen at random. Participants were asked to spell the last part (surname) of a full name from the table to measure the time differences in fully spelling the name versus selecting a suggestion. If the participant did not select the name suggested, they were not required to additionally spell the first part (given name) of the full name. The surnames of the highProbability names selected also were seven letters or less to reduce time stress on the participants.

4.5.2 Systems

As previously stated, two systems – experimental and control, were used in this study to complete the tasks described in section 4.5.1. This section discusses these systems and how they were implemented for the study.

4.5.2.1 Experimental Method

The experimental system implemented the predictive method discussed in Chapter 3. The predictive method discussed in chapter 3 has a multimodal design – taking speech, touch, and switch presses as input. For use in the study, this method had to be modified to accept a single input based on the group in which the participant was a part. Switch presses were solely used as input for the switch input group; speech was the sole input for the speech input group. Additionally, for the switch input participant group, the prompts were modified to read, “Press select,” rather than, “Say vote,” to make selections.

For this experiment, the highProbability database table (see section 3.6) was populated with write-in candidates registered in Alaska for the U.S. Senate race of the 2010 general election. Additional highProbability names, such as “Mickey Mouse,” were also added to the table. The full list of names in the highProbability table for this study is shown in Appendix 6.

4.5.2.2 Control Method

The control system was designed to guide the user to write-in a name using a linear spelling algorithm similar to those used in the DRE systems described in Chapter 2. There is neither clustering nor prediction in the control system. For each letter, the participant is presented with letters in alphabetical order, beginning with the letter ‘A’. In the speech input group, the participants are prompted, “Say vote if the first letter of the last name is ‘A’. Say vote if the first letter of the last name is ‘B’,” and so on. For the switch input group, the participants are prompted in a similar manner, exchanging “Say vote” for “Press select”. The system continues to traverse the alphabet until the voter selects the intended letter. If the voter misses their intended letter, they system continues traversing the alphabet until it reaches the letter ‘Z’, then begins again with the letter ‘A’. This process continues until the voter completes the spelling of the last name intended. Like the experimental system, if the “next” switch button is pressed, the system interrupts the current letter prompt and begins the next sequential prompt.

In order to determine the time-to-task for the speech input group, name spellings were simulated and calculated. For each participant, the names spelled for each task on the experimental system were the exact same names simulated by the control system. The simulation operated as if no mistakes were made during the spelling of a name. For example, the simulated dialogue for the spelling of the name “Abe” is as follows:

"You will now be prompted to spell the candidate's last name."
"You will now select the first letter of the candidate's last name."
"Say vote if the first letter of the last name is A."
<Listening...>
<Interrupting Listener. Letter Selected>
"You have chosen the letter A. Say vote to delete this letter."
<Listening...>
"You have currently selected A, as the candidate's last name."
"Say vote if you are finished spelling the candidate's last name."
<Listening...>
"You will now select the second letter of the candidate's last name."
"Say vote if the second letter of the last name is A."
<Listening...>
"Say vote if the second letter of the last name is B."
<Listening...>
<Interrupting Listener. Letter Selected>
"You have chosen the letter B. Say vote to delete this letter."
<Listening...>
"You have currently selected A, B, as the candidate's last name."
"Say vote if you are finished spelling the candidate's last name."
<Listening...>
"You will now select the third letter of the candidate's last name."
"You have already selected the letters A, B."
"Say vote if the third letter of the last name is A."
<Listening...>
"Say vote if the third letter of the last name is B."
<Listening...>
"Say vote if the third letter of the last name is C."
<Listening...>
"Say vote if the third letter of the last name is D."
<Listening...>
"Say vote if the third letter of the last name is E."
<Listening...>
<Interrupting Listener. Letter Selected>
"You have chosen the letter E. Say vote to delete this letter."
<Listening...>

"You have currently selected ABE spelled A, B, E, as the candidate's last name."
"Say vote if you are finished spelling the candidate's last name."
<Listening...>
<Interrupting Listener. Spelling Finished>

4.5.3 Procedure

A comparison of the experimental system versus the control system was achieved by having the participants spell the exact same name within a task with both the experiment and control systems. To measure efficiency, the computer recorded and calculated the time duration during the experiment for each participant to complete each task.

The success metric was determined upon the completion of a task. The use of the system implementation was deemed successful if the participant submitted the correct spelling of the intended name. In order to determine the usability of the methods used, participants were also asked to fill out a post-questionnaire [Appendix 5]. The post-questionnaire asked questions such as:

- Was the system usable or not usable?
- Was the system easy to use?
- Were the system instructions easy to understand?
- Was it easy for you to correct your spelling mistakes?
- Do you feel you made selections private?
- Do you think this system should be used in actual elections?

To reduce the effects of causal factors, the following controls were applied:

- All participants within the speech input group utilized the same microphone and speakers.
- All participants within the switch input group utilized the same switch and speakers.
- All experiments were performed in similar environments – undisruptive and quiet.

- Each participant received all information to complete tasks through the system.
- The tasks to be completed used the same process for each participant within a group regardless of the name to be spelled.
- Each participant was allowed the same flexibility to complete the experiment in the time needed. The experiment began immediately following the pre-survey, and the post-survey was given immediately afterward.
- Each participant was instructed not to discuss the experiment with anyone to ensure that all participants have equal knowledge of the experiment.
- All participants performed each experiment within their group. To avoid bias, the group and order of experiments will be random for each participant.

The following procedure was followed for each of the experiments:

1. Each participant agreed to participate after first reading an Institutional Review Board for the Protection of Human Subjects in Research (IRB) stamped information letter.
2. Participants completed a pre-questionnaire to capture demographic information.
3. Each participant was randomly assigned to either the speech input group or the switch input group. The premise of the study and the procedure tasks were then explained to the participants.
4. The first task each participant performed, either spelling a highProbability name or a name of their own choosing, was chosen at random. For the user chosen name tasks, the participant was asked to provide a last name less than seven letters. For each task, the researcher wrote the name on a blank index card and showed it to the participant, confirming the intended spelling of the name.
5. Task performance

a. Speech group – Participants completed both tasks using the experimental system. Simulations for the names provided by the control system were then performed, and the times were recorded.

b. Switch group – The first system used by the participants was chosen at random. Both tasks were then performed on the first system before moving onto the next system.

6. Post-questionnaires

a. Speech group – Participants were asked to complete a post-questionnaire on the experimental system.

b. Switch group – Upon completing two tasks on the first system, the participants completed a post-questionnaire on that system. Participants then completed another post-questionnaire on the second system after the final two tasks were carried out.

4.6 Data Collection and Analysis

To achieve the objectives of the experiment, the following data was measured and collected. All performance data was collected via the system. The system calculated and recorded all time data for later analysis. In addition to the performance data, a log of the interaction dialogue and error data was also recorded via the system. Completion data was recorded manually and entered into system after the study. User satisfaction measures were collected during the study through a post-questionnaire, and calculated for analysis afterwards.

Upon the completion of the experiments, the data was analyzed to determine the values for the evaluative measures presented below. This data includes pre and post surveys completed by participants. The types of data collected and measured are:

1. Time-to-task
2. Accurate completion of task
3. User Satisfaction

Task completion for a participant was deemed accurate if the name submitted at the completion of the task is the same as the name given, or assigned, prior to the start of the experiment. If the name submitted sounded the same as the initial name, but was spelled differently, it was considered inaccurate, and therefore the participant's completion of the experiment was unsuccessful. These measures were used to evaluate the interactions and implementations in this study.

4.7 Expected Outcomes

The primary expected outcome of this research was to create a novel approach to writing-in a candidate's name accessibly and efficiently, while providing user satisfaction. The expected outcomes of the experiment are as follows:

H1: The participants will be able to accurately complete the given tasks using both input methods.

H2: Task completion with the clustering and prediction system will yield faster times than completing the tasks with the linear control system.

H3: The spelling of high probability names will yield *extremely* longer times with the control system than the time-to-task for the prediction system.

H4: The efficiency of the switch input method will be greater than that of the speech input method.

H5: On average, the participants will be more satisfied using the interfaces implementing clustering and prediction than without.

Chapter 5

Research Findings

This chapter reports on the analysis of the data from the experiment discussed in Chapter 4. Presented first is the demographic information collected from the participants via the pre-questionnaire. Next, an analysis of the performance of the systems used in the experiment are reported, including the effectiveness and efficiency of the systems. In closing, this chapter details the findings of the user satisfaction metric, obtained from the participant via post-questionnaires.

5.1 Pre- Experiment Questionnaire

Each of the 40 participants completed a pre-questionnaire prior to using the systems, in order for their demographic information to be collected and analyzed. The average age of the participants was 22 years, with a standard deviation of 2.58 years. Figure 5.1.1 shows the number of participants categorized by age. Only eight percent of participants were non-U.S. citizens; these participants indicating they were of Indian or Other citizenship (see Figure 5.1.2). The highest degree completed of a majority of the participants was a high school diploma, at 70 percent, followed by a Master's degree and a Bachelor's degree, at 18 percent and 10 percent, respectively (see Figure 5.1.3).

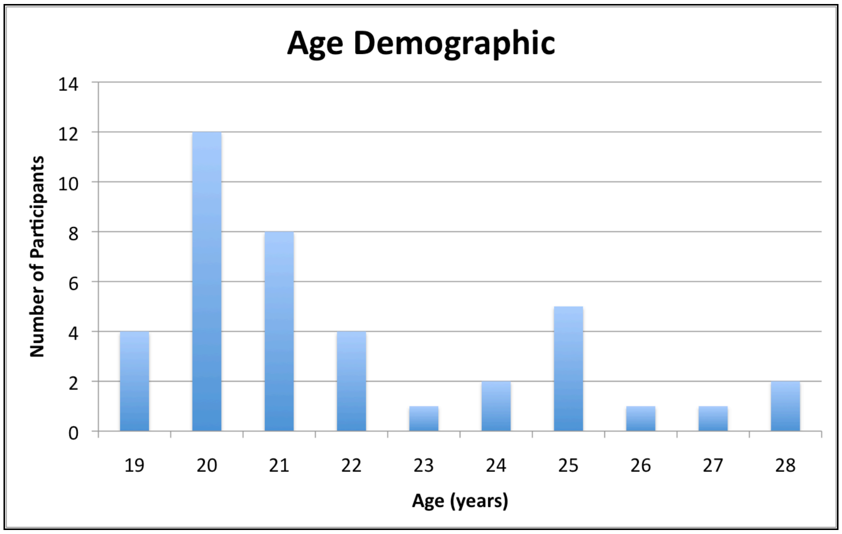

Figure 5.1.1 Number of Participants by Age

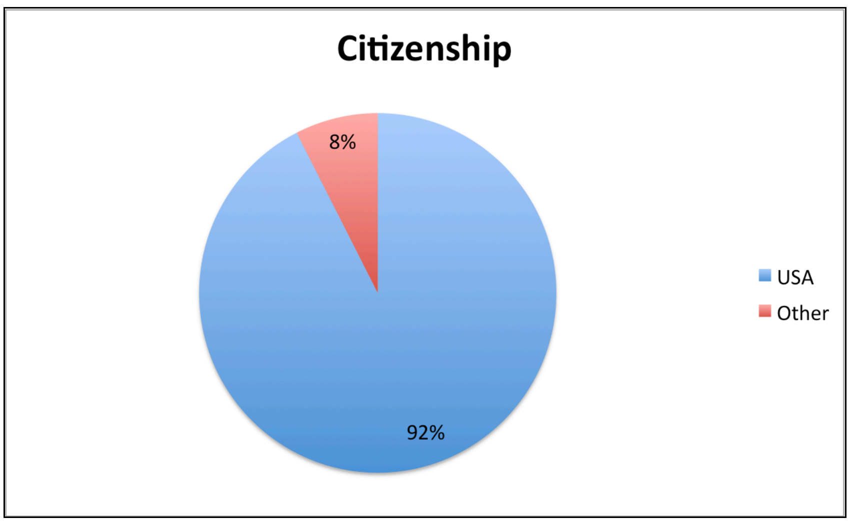

Figure 5.1.2 Participant Indicated Citizenship

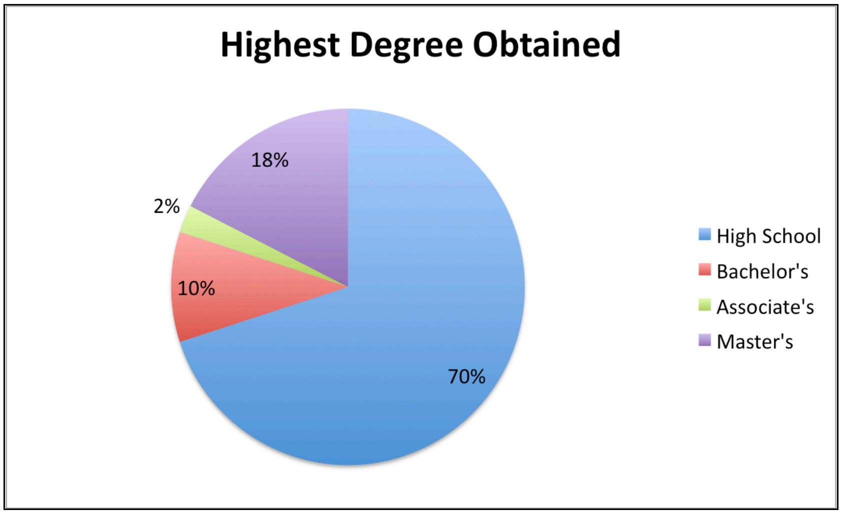

Figure 5.1.3 Highest Degrees Obtained by Participants

While 98 percent of participants indicated they had no form of disability, one participant indicated semi loss of hearing (see Figure 5.1.4). 92.5 percent of the participants indicated U.S. citizenship, and 92.5 percent also indicated that English is their native language (see Figure 5.1.5). Lastly, as reported in Figure 5.1.6, 80 percent of the participants indicated White as their race; Black and Asian both came in at seven percent; Native American and Other were indicated by three percent.

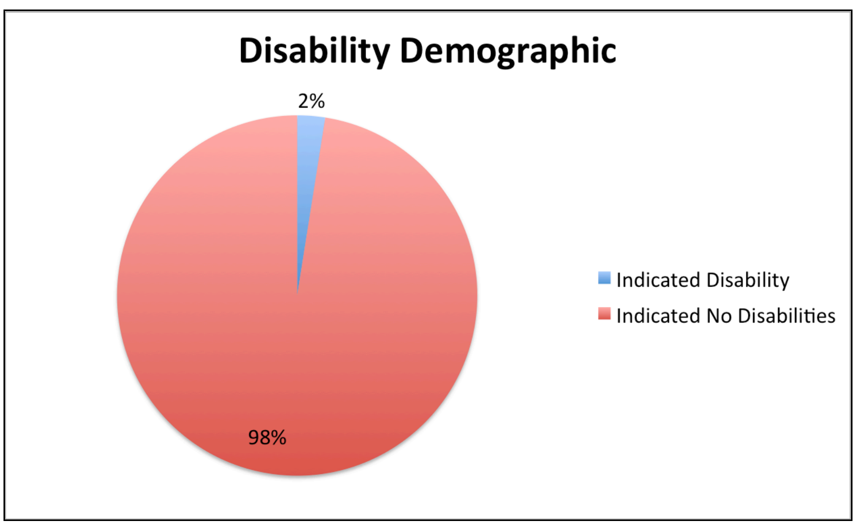

Figure 5.1.4 Participant Indicated Disabilities

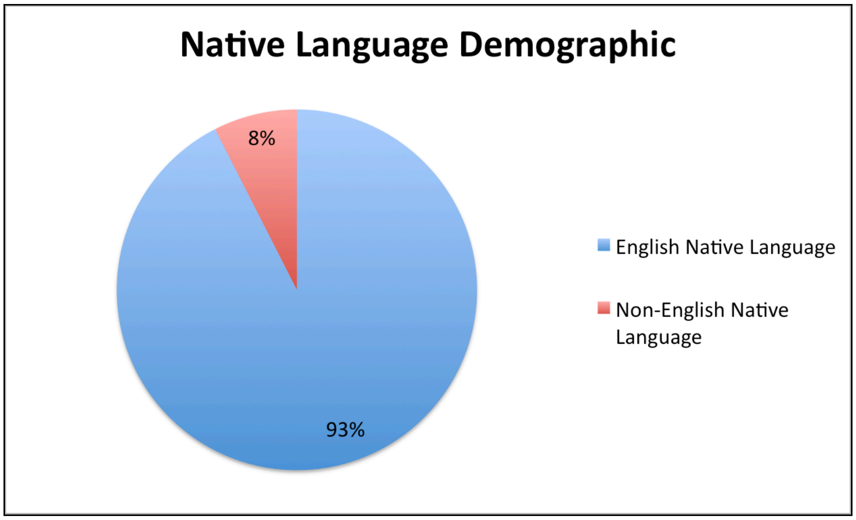

Figure 5.1.5 Participant Indicated Native Language

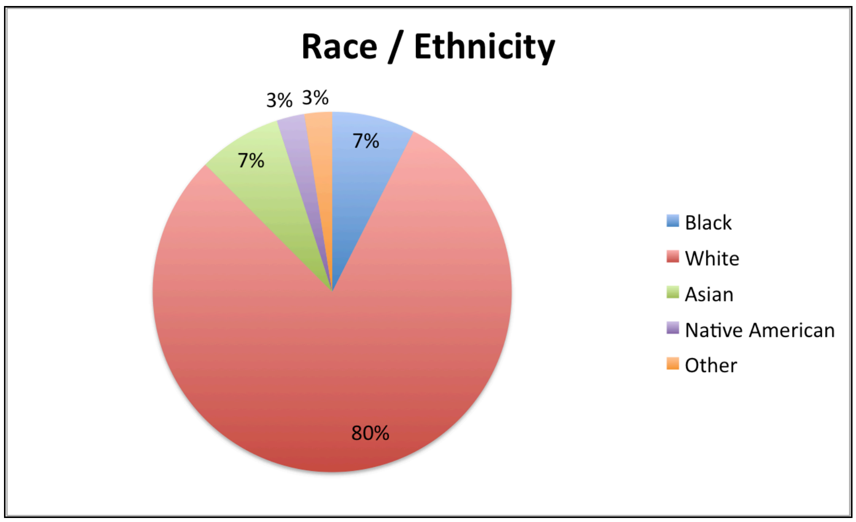

Figure 5.1.6 Participant Indicated Race

5.2 Performance Analysis

The performance metrics evaluated during the experiment were effectiveness and efficiency. The effectiveness is determined by analyzing the accuracy of task completion. Efficiency is determined by calculating the time-to-task completion. The following sections will report on the findings for each of the performance metrics individually.

5.2.1 Effectiveness

Effectiveness was measured discretely as success and failure. A task was deemed successful if the participant completed the spelling of a name correctly. If a participant was unable to complete the spelling, or if upon completion the name spelled was incorrect, the task was declared a failure. As stated in section 4.7, the expected outcome H1 states that the participants will be able to accurately complete the given tasks using both input methods. As shown in Figure 5.2.1, 93.75 percent of the tasks given to participants on the experimental system, including both the speech and switch interaction methods, were completed successfully.

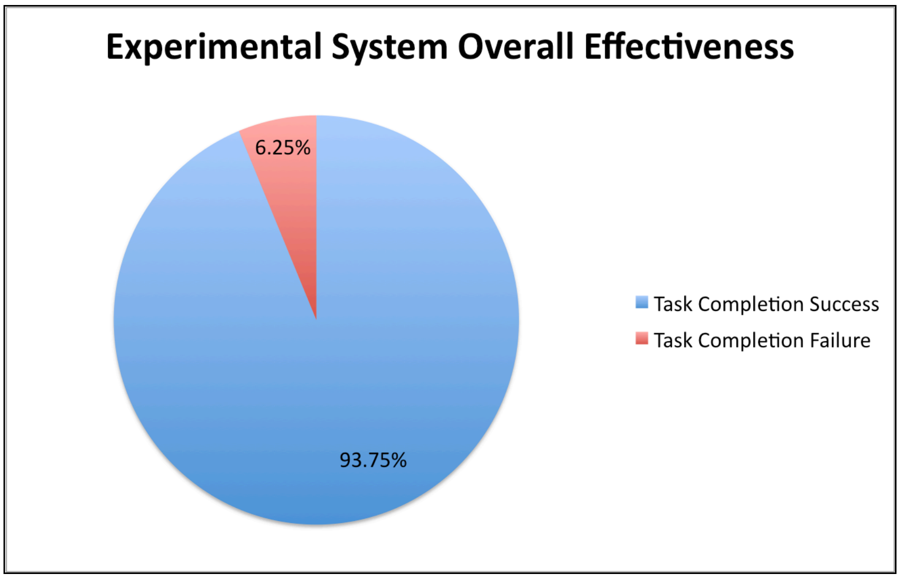

Figure 5.2.1 Effectiveness of Experimental System (Combined Speech and Switch)

Five participants had tasks deemed a failure, and none of these participants had more than one failed task. Of the five failed tasks, one task ended before the spelling of the name was complete. In the remaining four tasks, the name was spelled incorrectly. Of these four incorrect spellings, in three cases, the letter ‘Q’ was unintentionally chosen over the letter ‘U’; in the fourth case, the letter ‘N’ was chosen over the letter ‘M’. Because of these cases, a new feature discussed in section 6.3 will be added to the system implementation to reduce the failures, and therefore increase effectiveness.

5.2.1.1 Switch Input Effectiveness – Experiment vs. Control

The participants who used the switch interaction method performed tasks on both the experimental and control systems. Therefore, a comparison was made of the effectiveness of the two systems. There were a total of four tasks that were failed using the switch interaction method: two failures on the experimental system, and two failures on the control system (see Figure 5.2.2). These failures were independent of one another, that is, a different participant performed each of the four failed tasks.

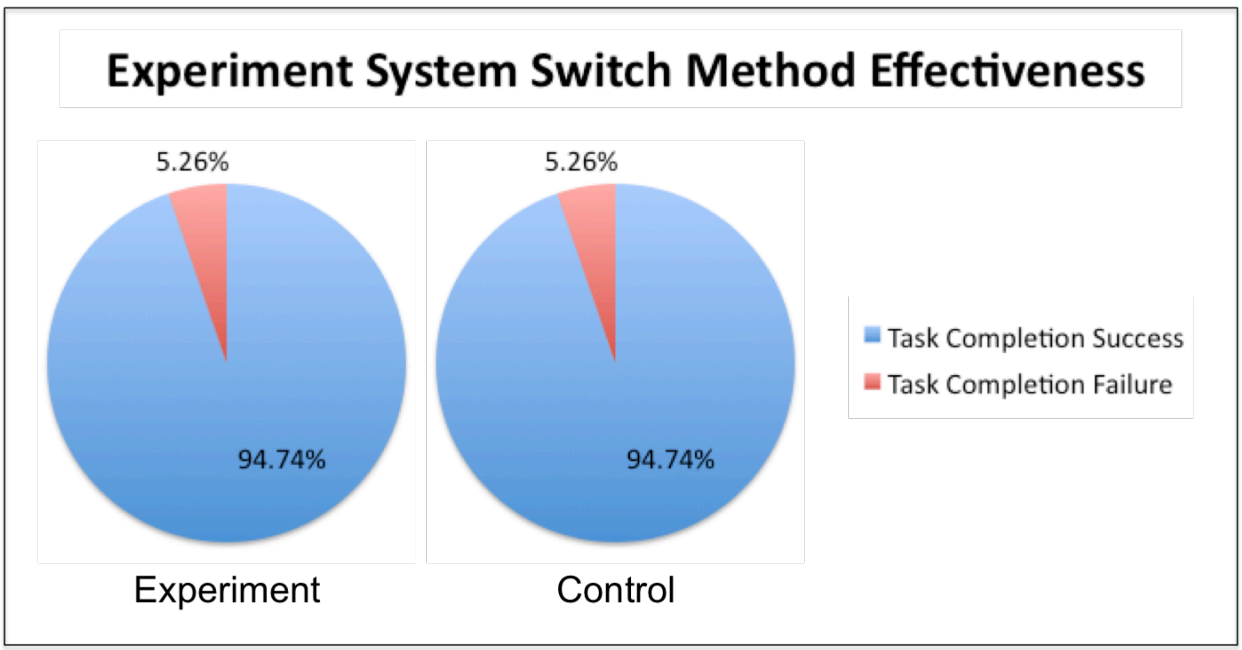

Figure 5.2.2 Switch Interaction Effectiveness

5.2.2 Efficiency

This section will report on the efficiency of the experiment and control systems, separately for both the speech interaction and switch interaction methods. This analysis will address the hypotheses, first discussed in section 4.7, H2, H3, and H4. Sections 5.2.2.1 and

5.2.2.2 presents an analysis of the efficiency of the speech and switch interaction methods, respectively. Hypothesis H4 is the focus of section 5.2.2.3, which provides an investigation of the comparison between the speech and switch interaction methods. Finally, in section 5.2.2.4, a special analysis of the last name, "Smith" is presented due to the large number of participants who chose to write in that name.

5.2.2.1 Speech Interaction – Experiment vs. Control

Hypothesis H2 states that task completion with the clustering and prediction experimental system will yield faster times than completing the tasks with the linear control system. Since the participants only chose surnames (user chosen names), and the high probability names were given *and* surnames, we will analyze at the time-to-task data for the user chosen names and the high probability names individually. The average time-to-task for the user chosen names on the experimental system was 5.19 minutes with a standard deviation of 1.99, while the average time-to-task for the same names on the control system was 8.12 minutes with a standard deviation of 2.07 (see Table 5.2.1).

Measure	Experimental System	Control System
Average Time-to-Task	5.19	8.12
Standard Deviation	1.99	2.07

Table 5.2.1 Speech Interaction Analysis Summary for User Chosen Names

A total of 21 user chosen names were spelled during the study. 81 percent of the names chosen had a record in the database. 88 percent of those names were suggested to the participants. Of those names, the participants selected 87 percent (see Table 5.2.2).

Measure	Value
Number of User Chosen Names Spelled	21
Names with Database Records	80.95%
Predicted Names Suggested to Participant	88.24%
Suggested Names Selected by Participant	86.67%

Table 5.2.2 Common Name Records & Speech Interaction Selection Percentages

Figure 5.2.3 provides further details about both systems with regards to speech interaction. Each name provided was placed in a group based on the number of letters in the surname. Within each group, the experimental system average was then evaluated against the control system average. The length with the fastest time-to-task was the three-letter group on the experimental system, at 2.83 minutes. The length with the slowest time-to-task was the seven-letter group on the control system, at 11.00 minutes. The smallest difference in time-to-task between the two systems was the three-letter group, with a time difference of .85 minutes. The largest difference was the six-letter group, with a time difference of 4.02 minutes. Based on this data, we can say that the experimental system is faster than the control system, but statistical evidence of this claim needs to be found.

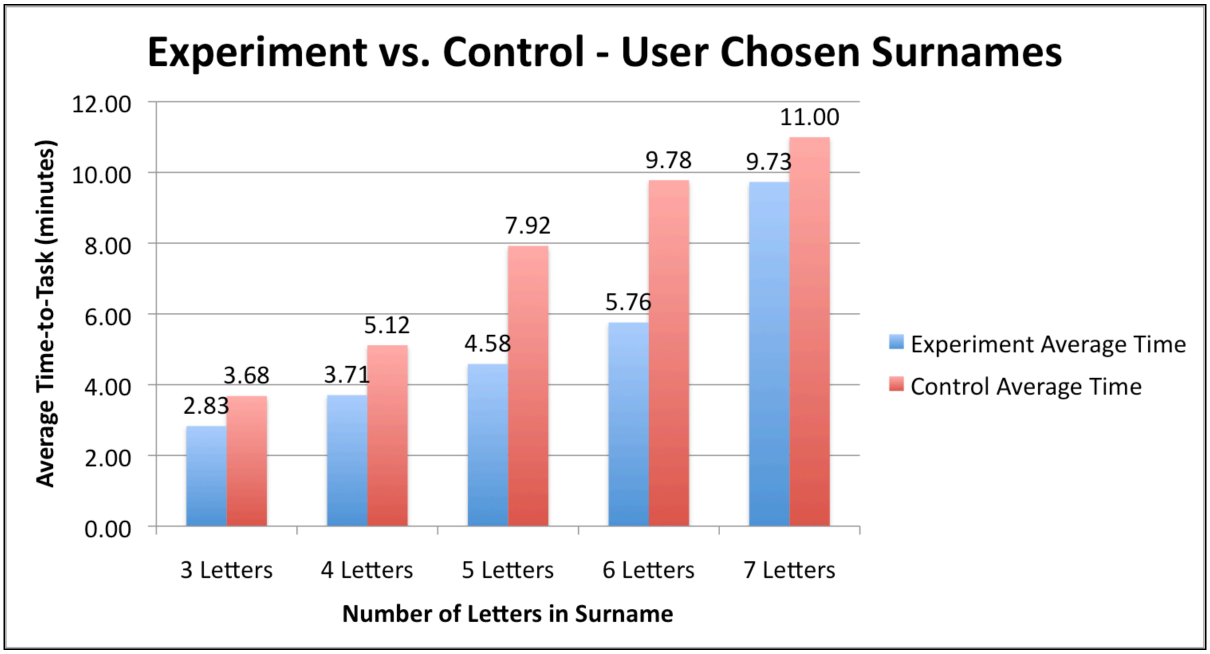

Figure 5.2.3 Time-to-Task by Name Length for User Chosen Names

In order to determine the statistical significance of the hypothesis H2, a t-test would appear to be the appropriate analysis tool. However, the t-test assumes the data is normally distributed, and that the samples have equal variances. Due to these constraints, there is a risk of error when applying a t-test to non-normal distributions of unequal variances. Therefore, tests of variance equality and normality were performed on the data to ensure the t-test would provide an accurate analysis. The results of the equality of variances test on the experimental and control systems yield a p-value of 0.3313, which shows no significant deviation from variance equality. The Shapiro-Wilk test for normality resulted in a p-value of 0.0019, indicating significant evidence that the data does not follow a normal distribution. Therefore, the Wilcoxon Rank Sum test was used to bypass the potential error of using the t-test. The Wilcoxon test yielded a p-value much less than .0001, which is well below the conventional significance value of 0.05. Therefore, we can reject the null hypothesis that the time-to-task for both systems are equivalent, and conclude that there is significant evidence showing that the experimental system is faster than the control system. This confirms the H2 hypothesis that users perform the task of spelling names faster on with the experimental system than on the control system. Figure 5.2.4 depicts a summary of the analysis of the time-to-task for the speech interaction method.

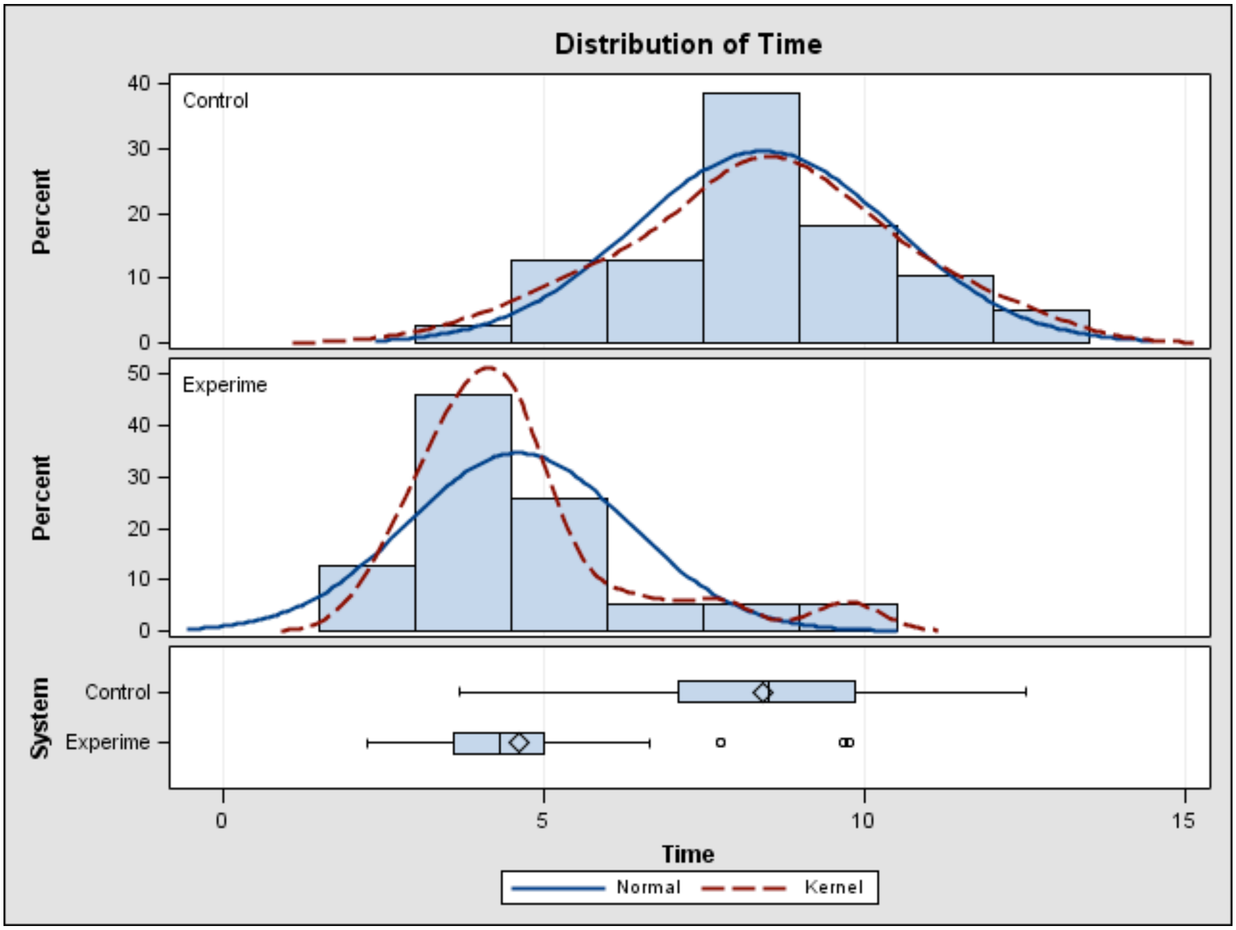

Figure 5.2.4 Speech Interaction Time-to-Task Analysis Summary

Hypothesis H3 states that the spelling of high probability names will yield *extremely* longer times with the control system than the time-to-task for the experimental system. Table 5.2.3 provides a short analysis of this high probability time-to-task data. The average time-to-task for the high probability names on the experimental system was 3.89 minutes with a standard deviation of 1.31, while the average time-to-task for the same names on the control system was 8.71 minutes with a standard deviation of 1.99.

Measure	Experimental System	Control System
Average Time-to-Task	3.89	8.71
Standard Deviation	1.31	1.99

Table 5.2.3 Speech Interaction Analysis Summary for High Probability Names

Figure 5.2.5 shows a more detailed analysis of the time-to-task data, broken down by high-probability name length. The time-to-task for the experimental system was more constant than that of the control system, for which the completion time increased as the name length increased. With these results, it is evident that the experimental system yields faster times than the control system (4.82 minutes difference between the averages), however, does this answer the question of *extreme* differences posed by the hypothesis H3? Based on the constraints placed on the study to reduce time strains on the participants, hypothesis H3 seems to be confirmed. As detailed in section 4.5.1, when using the control system, participants were not required to spell both the given and surnames of the high probability names provided. Therefore, the times listed for high probability names on the control system in this experiment are for the surname only, whereas the time-to-task data for the experimental system is for the full given and surnames. Given that voters would also spell the first name on the control system (the name is predicted on the experimental system), the actual control times would be greater than what is listed here, thus confirming hypothesis H3 for the speech interaction method. However, this does not paint a complete picture of the time-to-task results of the high probability names.

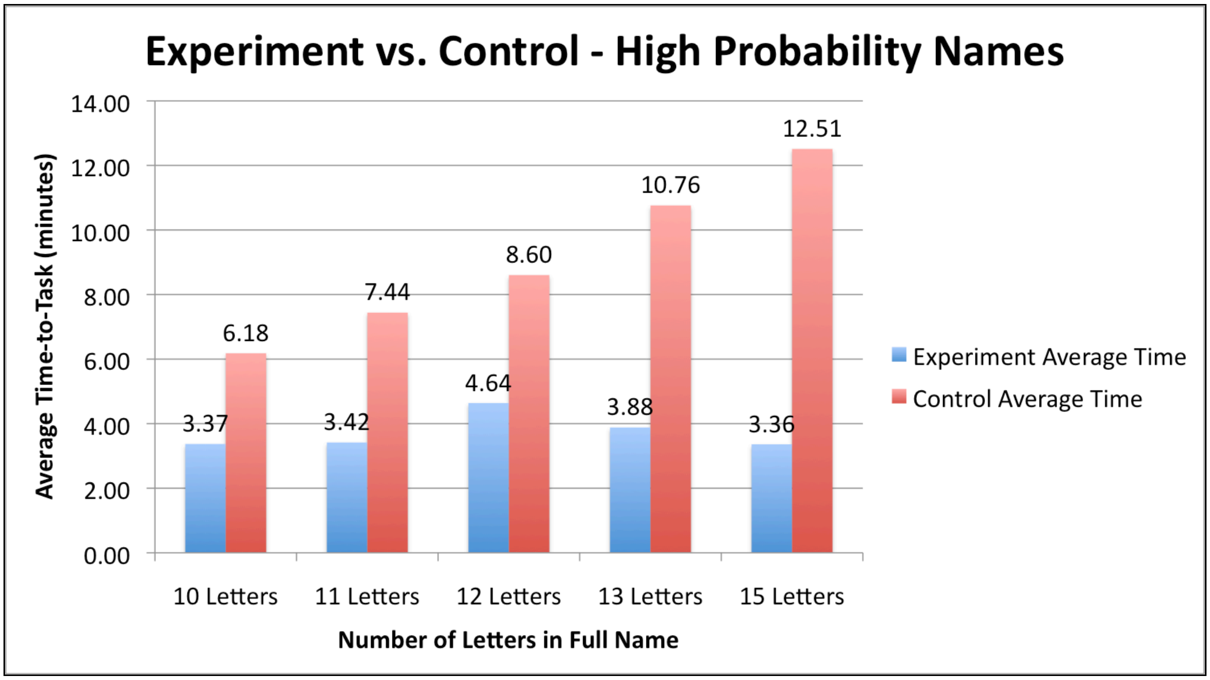

Figure 5.2.5 High Probability Time-to-Task by Name Length

The time-to-task data previously presented for the high probability names included those participants who selected the high probability name when it was suggested. However, if the participant did not select the suggested name, like the control system, they continued to spell the surname, but were not required to additionally spell the given name. As shown in Figure 5.2.6, 25 percent of the participants in the speech group did not select the high probability name when it was suggested. Fortunately, as discussed in section 3.6, common names can still be predicted if a high probability name is not selected. Of the full names not selected, 80 percent of the corresponding surnames were in the database, and suggested to the participant. Of the common surnames suggested, 75 percent were selected. Figure 5.2.7 shows the time-to-task data for these surnames. Even without selecting the full high probability name suggested, the spelling of the surnames yielded faster times on the experimental system than the control system.

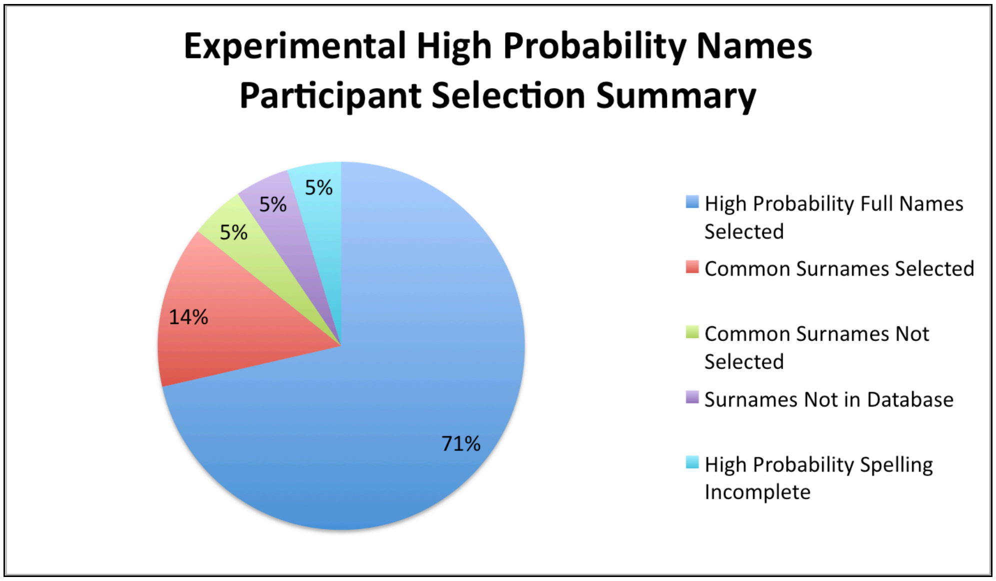

Figure 5.2.6 Participant Speech Selection of High Probability Names

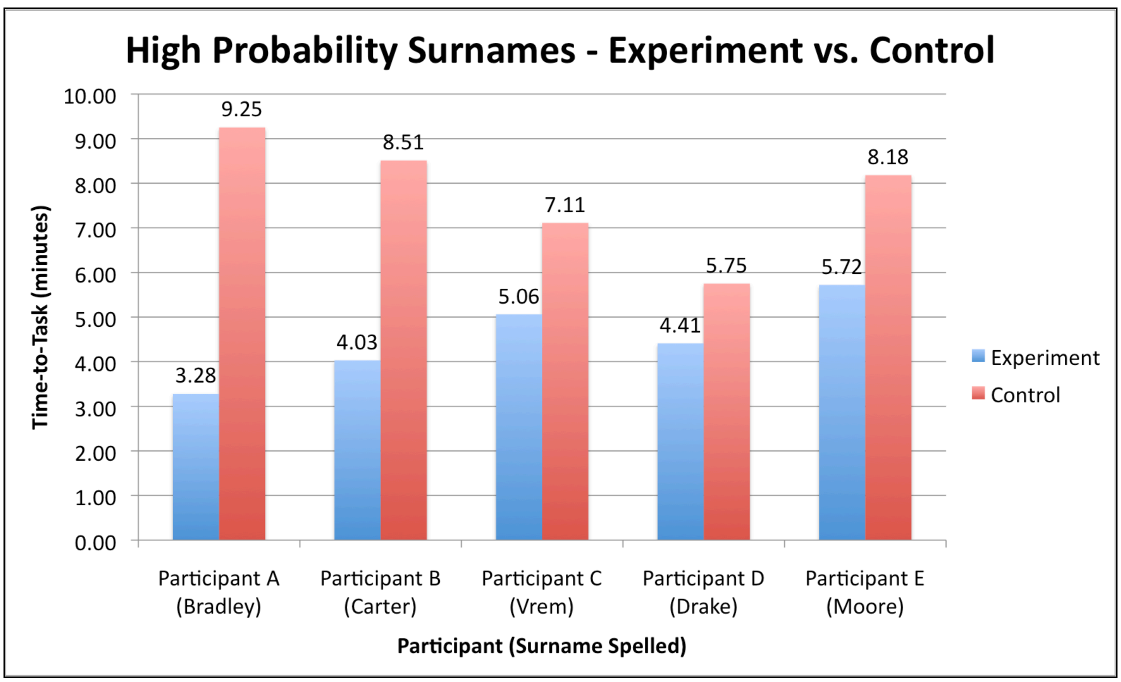

Figure 5.2.7 High Probability Surname Efficiency Comparison

5.2.2.2 Switch Interaction – Experiment vs. Control

Hypothesis H2 states that task completion with the clustering and prediction experimental system will yield faster times than completing the tasks with the linear control system. In a similar format as in section 5.2.2.1, the time-to-task data for the user chosen names and the high probability names will be analyzed individually. As shown in Table 5.2.4, or the user chosen names, the average time-to-task on the experimental system was 3.51 minutes, with a standard deviation of 0.97. On the control system, the average time-to task was 4.9 minutes, with a standard deviation of 1.58.

Measure	Experimental System	Control System
Average Time-to-Task	3.51	4.90
Standard Deviation	0.97	1.58

Table 5.2.4 Switch Interaction Analysis Summary for User Chosen Names

Each of the 19 participants in the switch group chose a surname to be spelled. There were records in the database for 74 percent of these names. 93 percent of the names with database records were suggested to the participants. When names were suggested, 100 percent of the participants selected the name (see Table 5.2.5).

Measure	Value
Number of User Chosen Names Spelled	19
Names with Database Records	73.68%
Predicted Names Suggested to Participant	92.86%
Suggested Names Selected by Participant	100%

Table 5.2.5 Common Name Records & Switch Interaction Selection Percentages

Figure 5.2.8 provides further details, according to name length, about both systems with regards to switch interaction. The shortest surname chosen was "Doe," and three letters in length, while the longest surnames spelled, at six letters, were "Newton" and "Dawkin". There were four four-letter surnames spelled, averaging 2.94 on the experimental system and 3.61 on the control system. Averaging 3.34 and 5.18 on the experimental and control systems, respectively, were 12 five-letter surnames. From this data, it appears as if the experimental system was unmistakably the faster of the two systems, however, further analysis is needed to determine if the results were statistically significant. This analysis is presented later in this section.

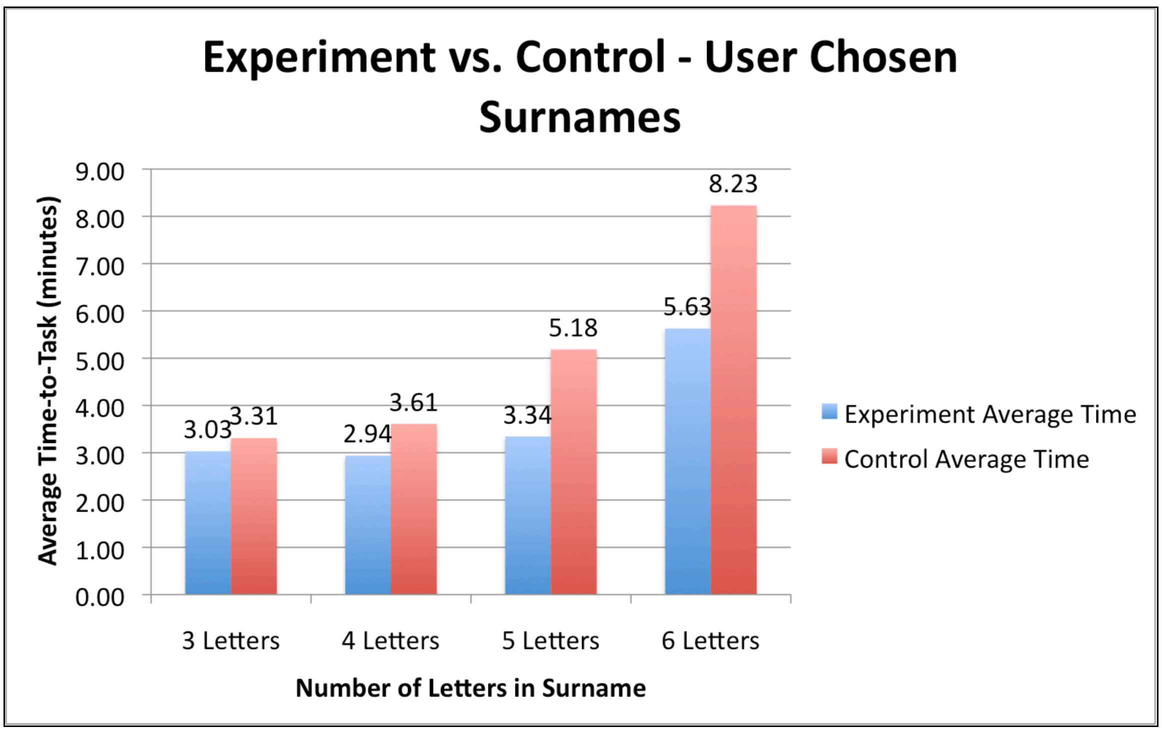

Figure 5.2.8 Switch Interaction - User Chosen Surnames by Name Length

Since the order in which each participant used the experimental and control systems first varied, there may have been some effect on the time-to-task data. Figure 5.2.9 compares the time-to-task averages within each system based on which system was used first. For the experimental system, there did not seem to be much of a difference between the time-to-task data, regardless of which system was used first. With the control system, when used second, participants showed a slightly better performance. This may show that the control system was easier to use once the participant spelled names on the experimental system.

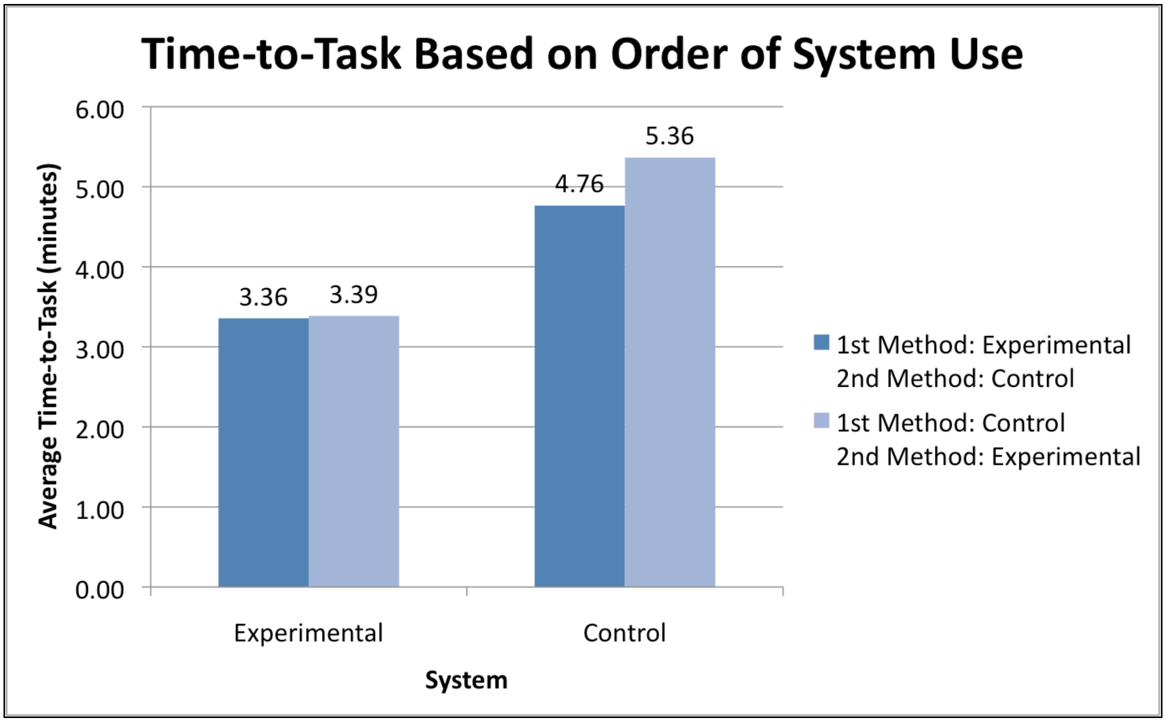

Figure 5.2.9 Switch Interaction Time-to-Task Usage Order Comparison

Like the statistical analysis performed for the speech interaction method (see section 5.2.2.1), a t-test would appear to be the appropriate analysis tool to determine the statistical significance of the hypothesis H2. In the case of the switch interaction, each participant used both the experiment and control systems, which requires the use of a paired t-test to determine significance. Since the paired t-test will evaluate the time-to-task differences between the experimental and control systems, only the participant data for which both times were available were included in the dataset; the data from the two participants who failed to complete the task on the experimental system, and data from the three participants who failed on the control system were omitted.

The paired t-test was used to test the null hypothesis that there is no time difference between the control and experimental systems. Like the pooled t-test used in section 5.2.2.1, the paired t-test works best if the data is normally distributed. Therefore, the data was first tested for normality to ensure the paired t-test would provide an accurate analysis. Results of the Shapiro-Wilk test for normality yield a p-value of .0921, which is greater than the standard significance level of .05; there is not significant evidence to reject the assumption of normality. Proceeding with the paired t-test, we found that the mean difference between times on both systems is 1.9676 minutes. We can say with 95% confidence that the mean time to spell a name with the experimental system is between 1.4413 and 2.4938 minutes faster than the mean time to spell a name using the control system. Figure 5.2.10 depicts a summary of the comparison data distribution. The paired t-test resulted in a p-value much less than .0001. Thus, we can reject the null and conclude that there is significant evidence to suggest that the experimental system is faster than the control system using the switch interaction method. To ensure these results are free of error due to further assumptions made by the t-test, the more robust Signed Rank Wilcoxon test was also used to analyze the data. The Signed Rank Wilcoxon also yielded a p-value much less than .0001, resulting in the same conclusion as the paired t-test, confirming the H2 hypothesis that users perform the task of spelling names faster on with the experimental system than on the control system.

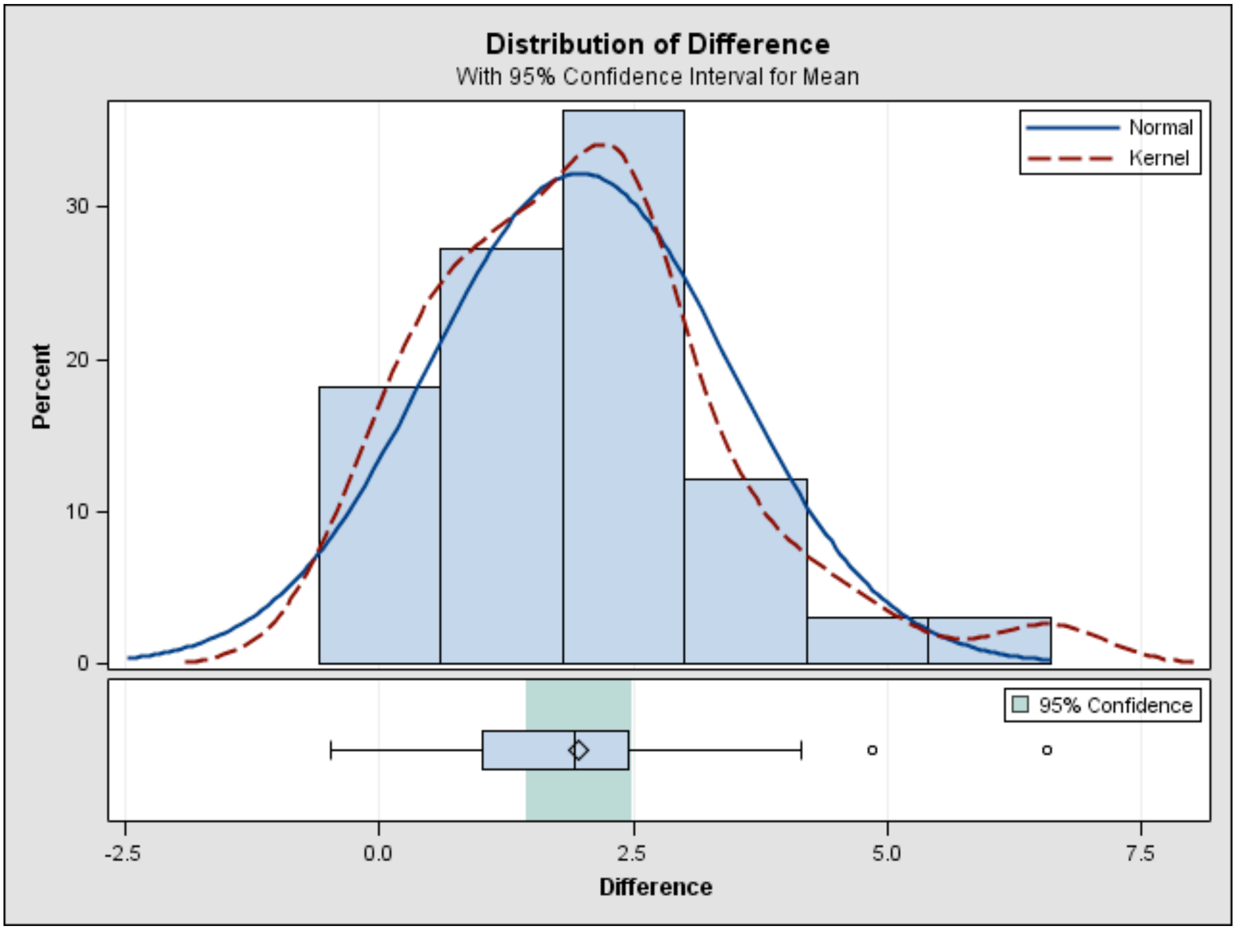

Figure 5.2.10 Switch Interaction Time-to-Task Analysis Summary

Hypothesis H3 states that the spelling of high probability names will yield *extremely* longer times with the control system than the time-to-task for the experimental system. As discussed in section 4.5.1, high probability names were asked of the participants to be spelled in full only for the experimental system. Participants were only required to spell the surname using the control system. Nonetheless, the time-to-task data for the experimental system was much lower than for that of the control system (see Table 5.2.6). The average time-to-task on the experimental system was 3.22 minutes, 2.21 minutes less than the control system, which had an average time-to-task of 5.43 minutes.

Measure	Experimental System	Control System
Average Time-to-Task	3.22	5.43
Standard Deviation	0.87	1.55

Table 5.2.6 Switch Interaction Analysis Summary for High Probability Names

Figure 5.2.11 provides a more detailed analysis of the time-to-task data by high-probability name length. The longest high probability name to be spelled was "Michael Dunton", at 14 letters in length, taking the shortest amount of time on the experimental system, 1.99 minutes compared to 4.84 minutes on the control system. The closest average time-to-task measurement between the experimental and control systems was a difference of 1.01 minutes, for high probability full names ten letters in length. Although, taking into consideration the fact that the control system time-to-task is only averaging the four to six letter surname for this ten-letter group, the actual time difference to spell the full name would be much greater. Therefore, this data combined with the other time-to-task data shows that hypothesis H3 is essentially confirmed. However, as with the other hypotheses, statistical analysis was performed to provide evidentiary support. This additional analysis is presented later in this section.

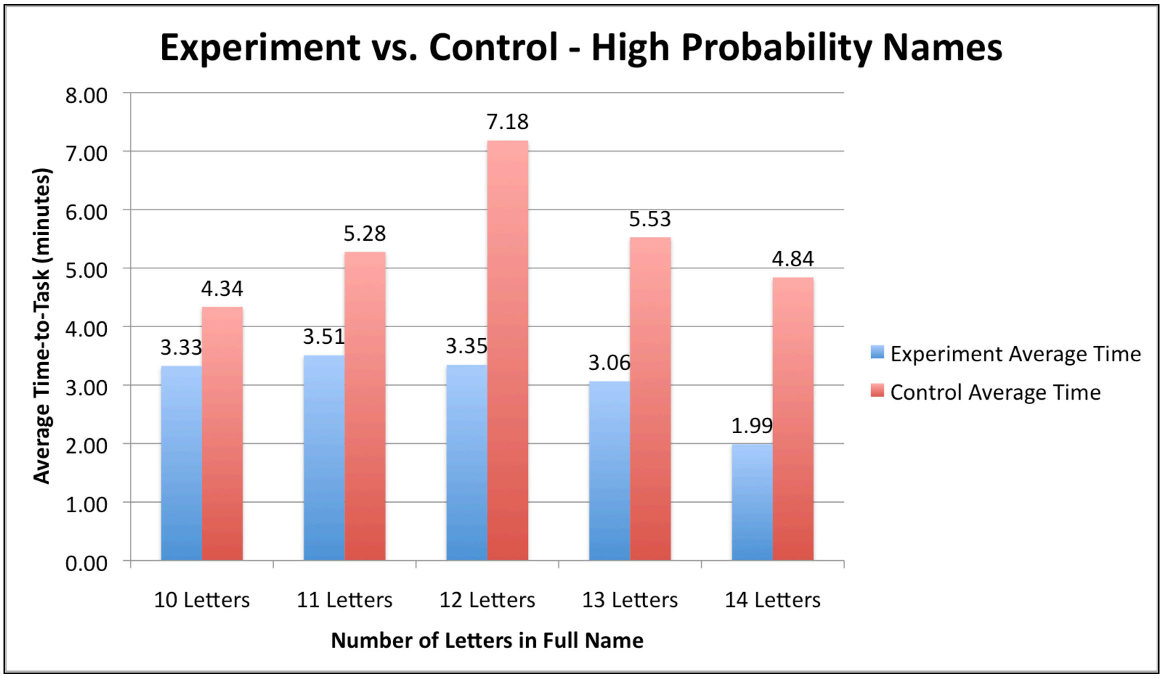

Figure 5.2.11 Switch Interaction - High Probability Time-to-Task by Name Length

Figure 5.2.12 shows the selection efficiency summary for high probability names. 18 participants completed the spelling of spelling of high probability names with the experimental system. Two participants in the switch group did not select the high probability name when it was suggested. One of these two names had a record as a common name in the database, and was selected.

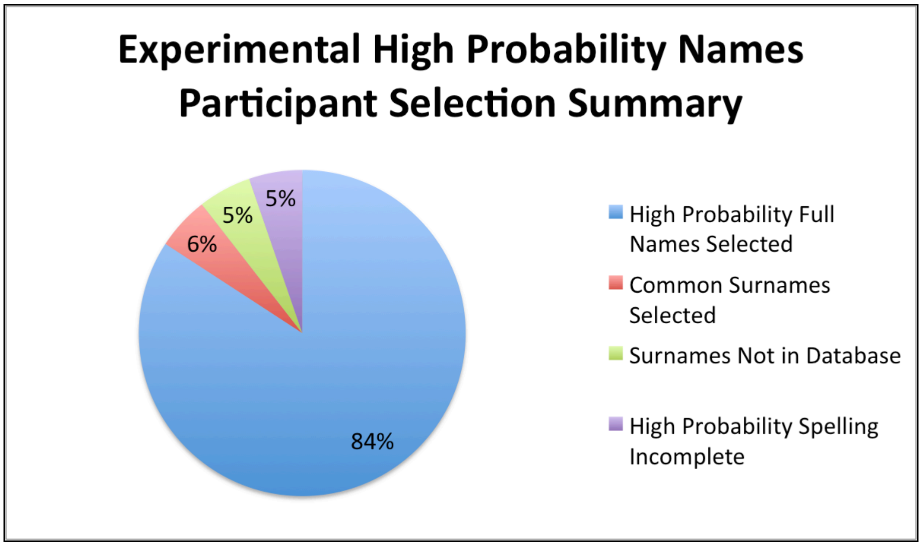

Figure 5.2.12 Participant Switch Selection of High Probability Names

The hypothesis H3 states that the spelling of high probability names will yield *extremely* longer times with the control system than the time-to-task for the prediction system. In order to evaluate this hypothesis, a statistical analysis was performed to determine if a three-minute difference in the control and experimental times for the high probability names using switch interaction was statistically significant. A paired Signed Rank Wilcoxon test resulted in a p-value of .0399, which is less than the conventional .05 level of significance. Therefore, there is significant evidence to reject the null hypothesis, and it can be concluded that statistically, voters will take at least three minutes longer to spell a highly probable name using the control system than the experimental system, with the switch interaction method. This confirms the H3 hypothesis that spelling highly probable names using the control system takes *much* longer than the same on the experimental system.

5.2.2.3 Speech Interaction Method vs. Switch Interaction Method

The hypothesis H4 states that the efficiency of the switch interaction method will be greater than that of the speech interaction method, meaning that, using the experimental system, the time-to-task for the switch interaction method will be faster (lower) than the same for the speech interaction method. The reason that the switch interaction method is expected to be faster is the use of a barge-in feature in the design. Using the switch, participants had the option to interrupt the prompts, skipping to the next sequential prompt. With the speech interaction, barge-in was not permitted in order to prevent the system from recognizing the speech output as input – as mentioned in section 4.5, the spoken prompts were output via speakers for observational purposes.

Results of the study show that, on average, the switch interaction method was in fact more efficient than the speech interaction method. Figure 5.2.13 shows a comparison of the two interaction methods separately for both the experimental and control systems, broken into user chosen and high probability name categories. For each category, the speech method had a higher time-to-task than the switch method. The differences in the average time for the control system were even larger than that of the experimental system. It can be assumed that this is due to the basic linear nature of the control system. Although the experimental system presents clusters and letters in a linear manner, participants were more careful when skipping through prompts because of the system's ability to make letter and name predictions. Since the control system does not make any predictions, it is easier to predict which prompt will be presented next.

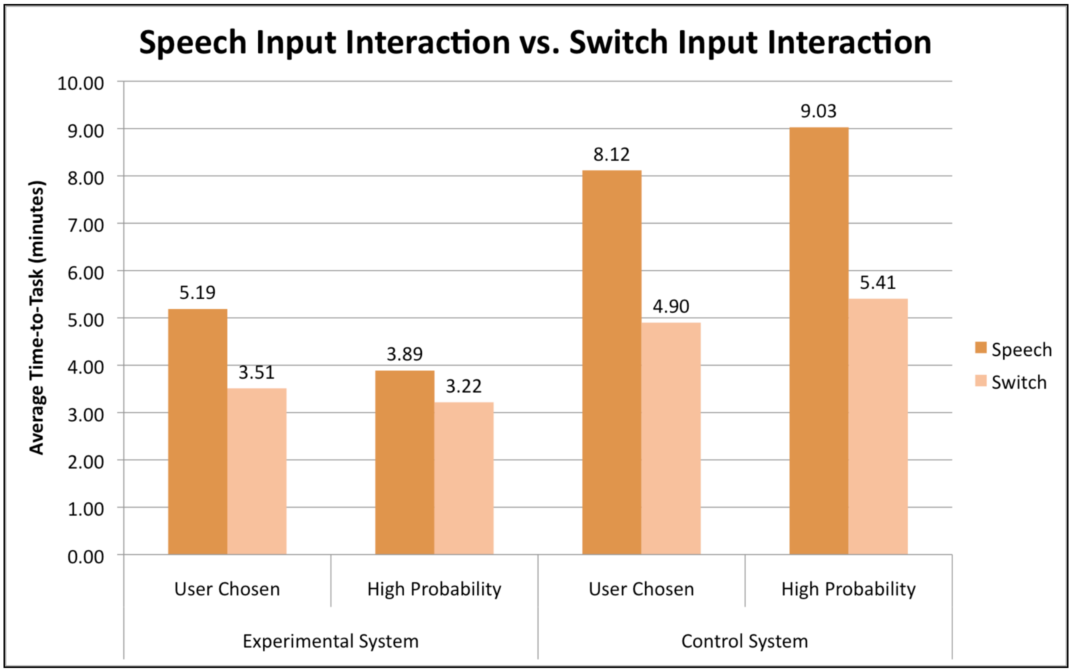

Figure 5.2.13 Speech vs. Switch Interaction Method Comparison

Although these averages appear to be conclusive that the switch interaction method yields faster times than the speech interaction method, they are not sufficient for determining statistical significance. Therefore, statistical analysis was performed. Since the results of the Shapiro-Wilk test show that this data does not follow a normal distribution (p-value much less than .0001), the Wilcoxon Rank Sum test was used to analyze the two interaction methods. The results of the Wilcoxon test yield a p-value of .0001, which shows that there is significant evidence to reject the null that performing tasks on the two methods produces the same times. Figure 5.2.14 depicts a summary of the comparison data distribution. The switch interaction method is significantly slower than the speech interaction method, confirming the H4 hypothesis that the switch interaction method is more efficient than the speech interaction method.

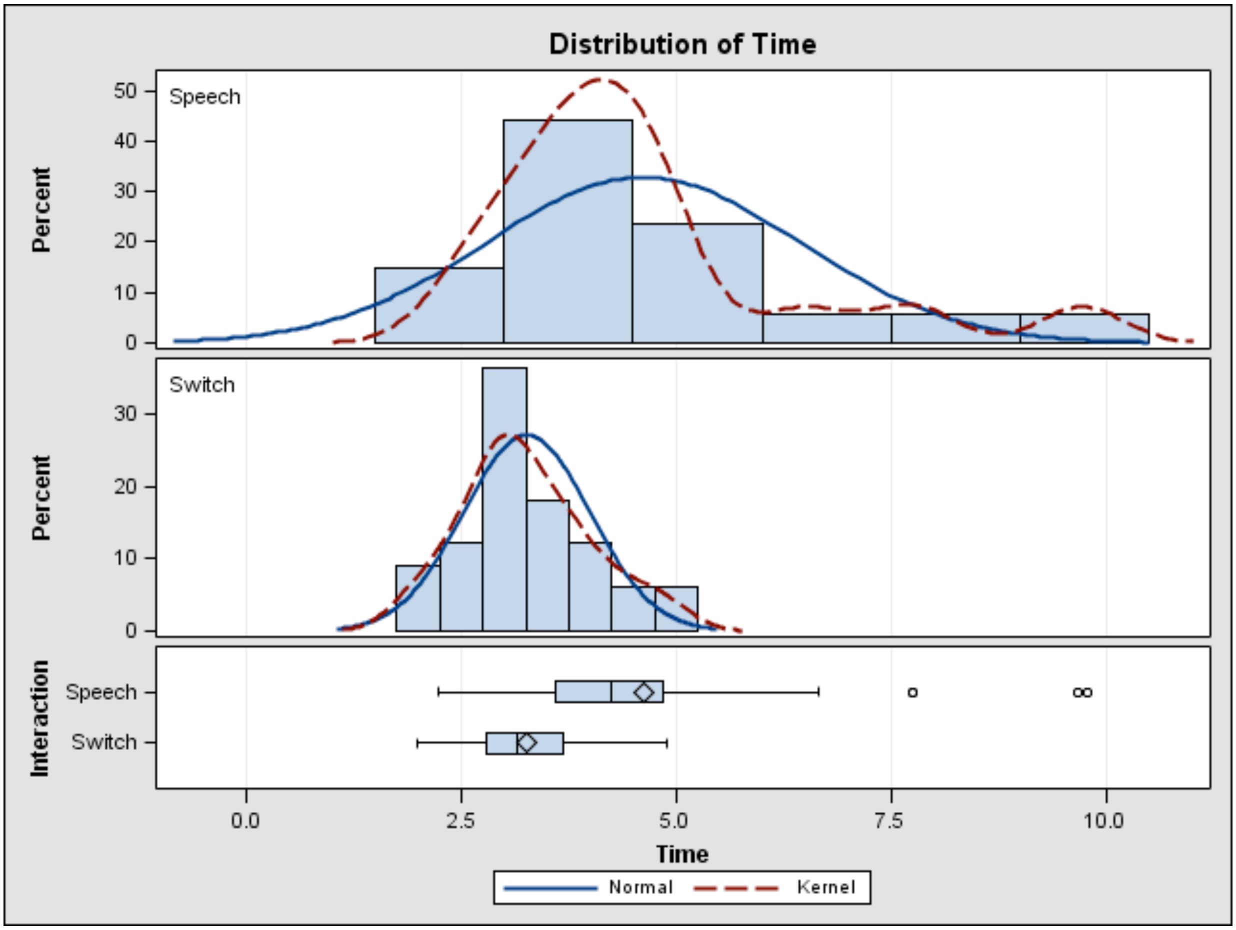

Figure 5.2.14 Distribution Summary of Speech vs. Switch Comparison Data

5.2.2.4 Special Efficiency Analysis Case: "Smith"

During the experiment, the names participants chose to spell were often the same. The most common last name chosen to be written-in was "Smith," at 27.5 percent of the names chosen. Since such a large number of participants chose this name, an individual efficiency analysis was performed as a special case. On average, participants spelled "Smith" in 4.02 minutes (including speech and switch interaction methods) using the experimental system. With the control system, the same task took an average of 6.81 minutes. This produces an average difference of 2.87 minutes. There was only one participant who completed the spelling in less time on the control system than the experimental system, by a measure of 1.2 seconds. Figure

5.2.15 shows the time-to-task for each participant individually, with the overall shortest and longest times highlighted in red.

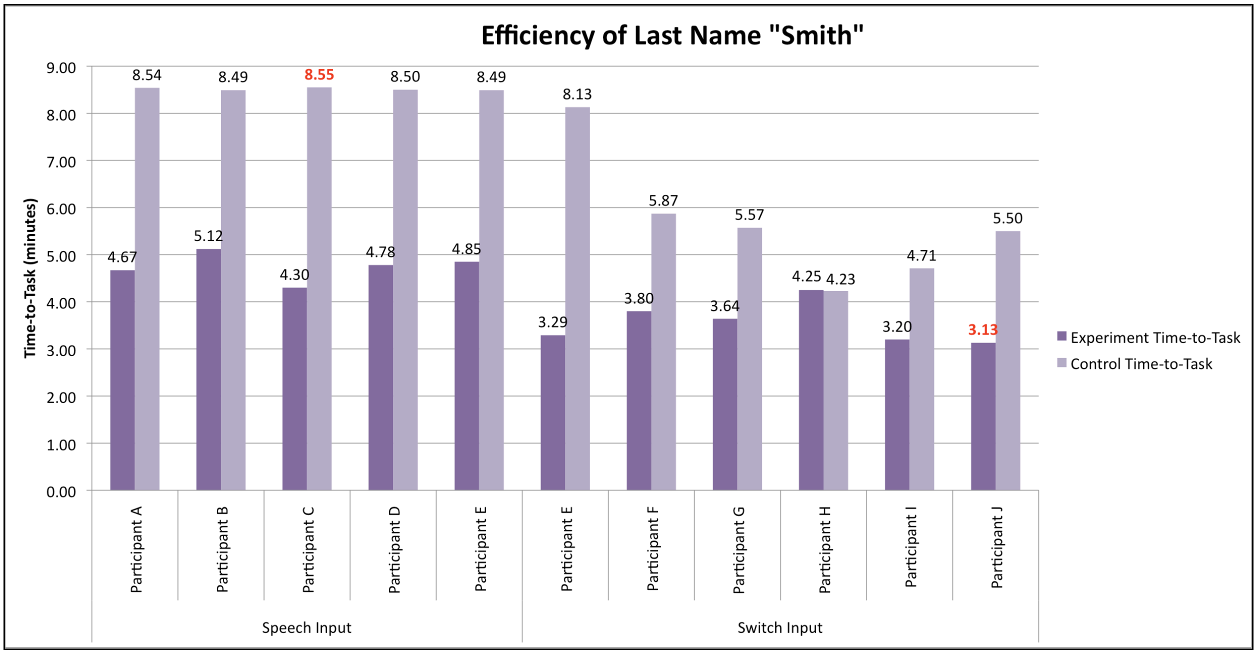

Figure 5.2.15 "Smith" Efficiency by Participant

5.3 Post- Experiment Questionnaire (User Satisfaction)

The post-questionnaire was used to gain knowledge of the participants' opinion of the system designs used in the experiment. User satisfaction was measured based on the participant responses from this questionnaire (see Appendix 5). This section details an overall analysis of the combined user satisfaction of the speech and switch interaction methods on the experimental system. Sections 5.3.1 and 5.3.2 describe these two interaction methods individually. Section 5.3.3 provides insight into what, if anything may have had an effect on the user satisfaction results.

As stated in section 4.7, the expected outcome H5 states that on average, the participants will be more satisfied using the interfaces implementing clustering and prediction than without; meaning that the overall user satisfaction of the experimental system will be greater than that of the control system. Because the participants only performed tasks on the control system using the switch interaction method, user satisfaction was not collected for the control of the speech interaction method. Section 5.3.2 reports the comparative evaluation of the expected outcome, H5.

Figure 5.3.1 depicts the satisfaction levels of the experimental system according to questions asked on the post-survey. Participants were asked to respond to the questions using a Likert scale, to get a quantitative measurement of the user satisfaction. On the Likert scale, 5 indicated “Strongly Agree,” 4 indicated “Agree,” 3 indicated “Neutral,” 2 indicated “Disagree,” and 1 indicated “Strongly Disagree.” Additionally, the participants had the option of omitting a question by writing N/A (not applicable). Of the 40 participants, 85 percent agreed or strongly agreed that the system was usable. 78 percent strongly agreed or agreed that the experimental system was easy to use, while 65 percent felt likewise about the ease of correcting mistakes. 93 percent and 83 percent agreed or strongly agreed, respectively, that selections were made privately and the experimental system should be used in elections. Overall, from the participants’ opinion of the experimental system (including both switch and speech interaction), we can conclude that on average, the results were a positive 4.24 on the Likert scale.

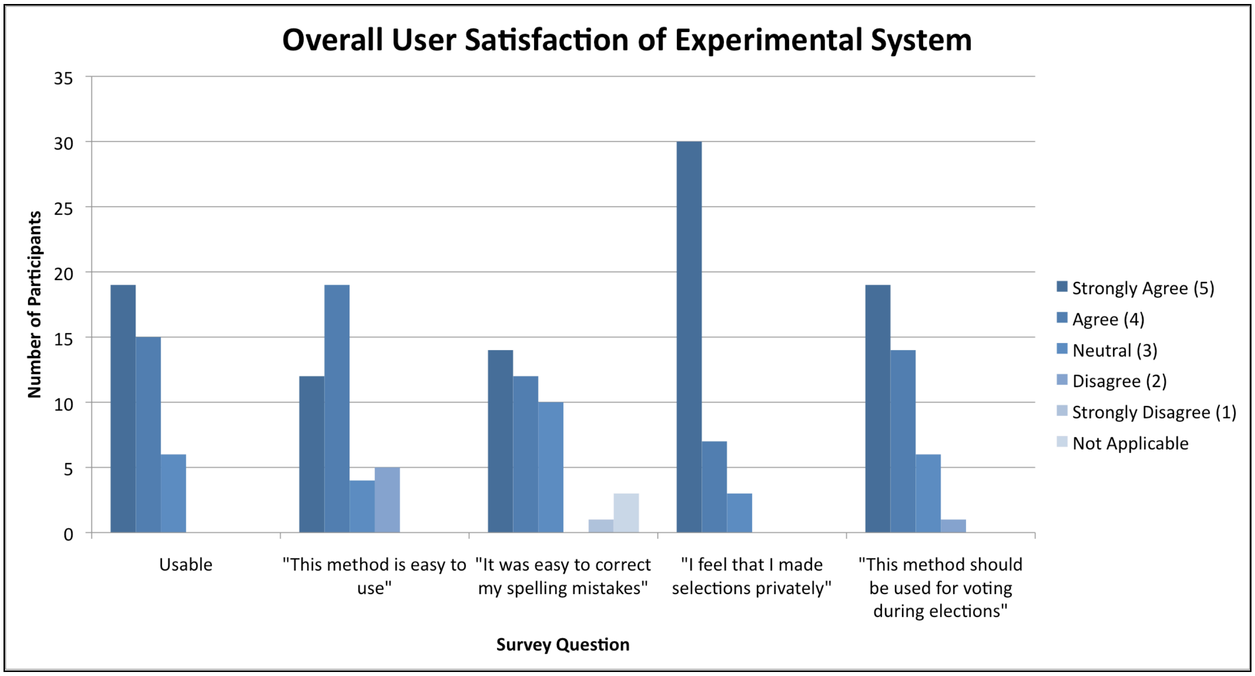

Figure 5.3.1 User Satisfaction of Experimental System

5.3.1 Speech Interaction – Experiment

Participants were asked several questions about their opinion of the experimental system implementing speech interaction via the post-questionnaire. Participants were not asked about their perceptions of the control system with speech interaction because they did not perform tasks using that system. The format of the questions was a statement about the system followed by the Likert scale introduced in section 5.3. Each participant chose a response based on this Likert scale. Among those questions were the following statements:

1. This method is easy to use.
2. It was easy to understand the instructions.
3. It was easy to correct my spelling mistakes.
4. I feel that I made selections privately.
5. This method should be used for voting during elections.

From these questions, the responses were as reported in Figures Figure 5.3.2 through Figure 5.3.7. 72 percent of the participants found the experimental speech method easy to use, while 67 percent thought the instructions were easy to understand. Although nine percent of participants felt they did not make any mistakes, and marked N/A on the post-questionnaire, 67 percent felt that if mistakes were made, they were easy to correct.

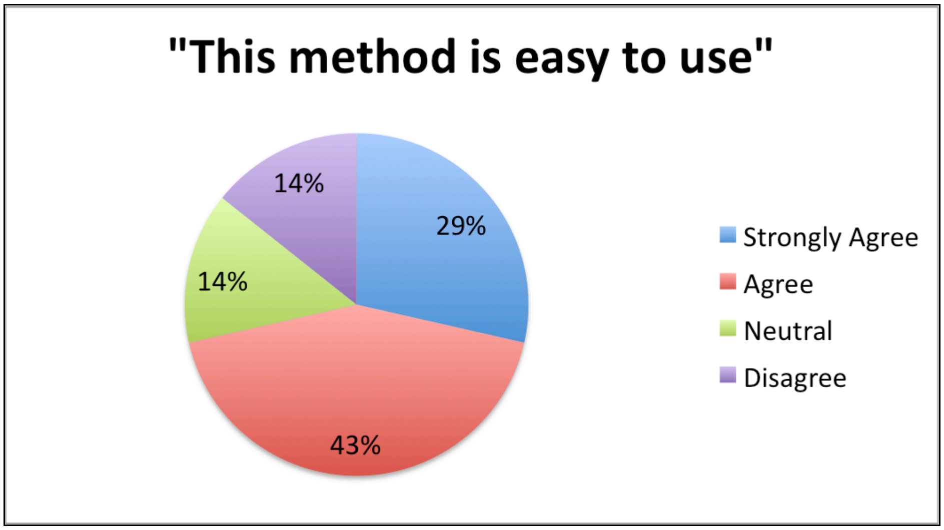

Figure 5.3.2 Ease of Use of Experimental System

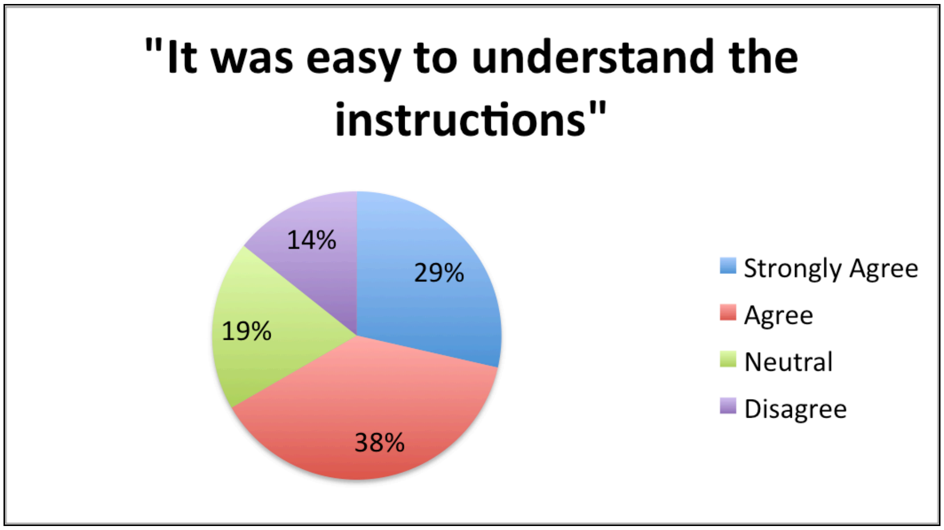

Figure 5.3.3 Experimental System Instruction Understandability

Figure 5.3.4 Ease of Experimental System Error Correction

An overwhelming 90 percent of participants felt they made selections through speech privately, and no participant disagreed with that sentiment. Additionally, 81 percent of participants indicated that the experimental method should be used in the voting process during elections. Above all, 81 percent of the participants in the speech interaction group indicated that the experimental system was usable.

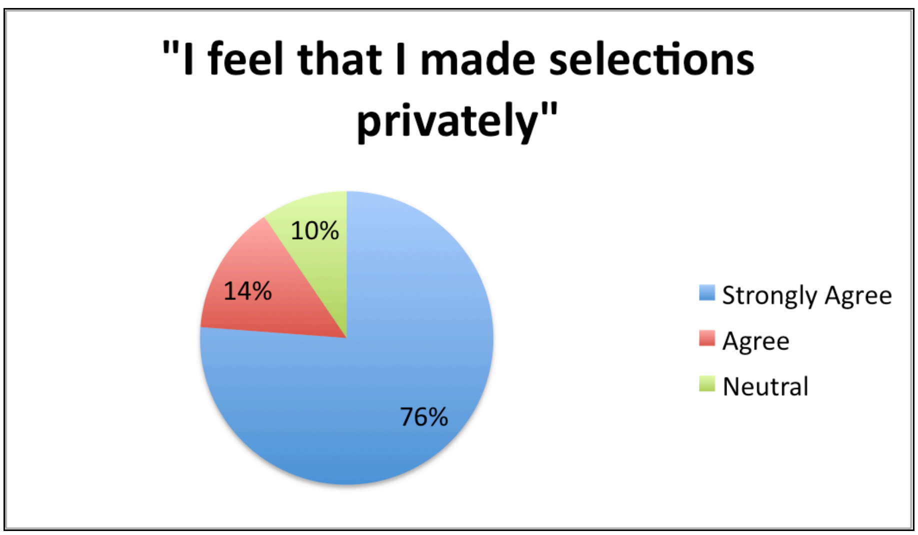

Figure 5.3.5 Experimental Interaction Privacy

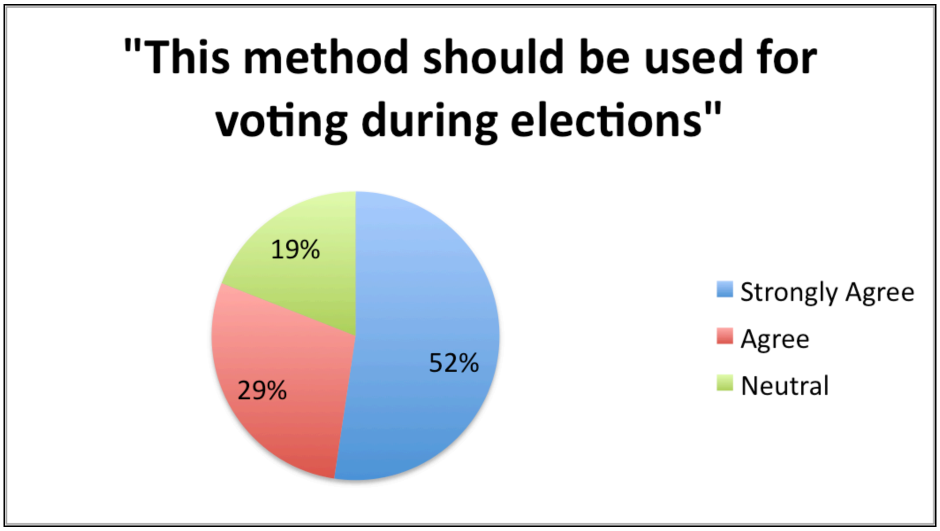

Figure 5.3.6 Experimental System Use in Elections

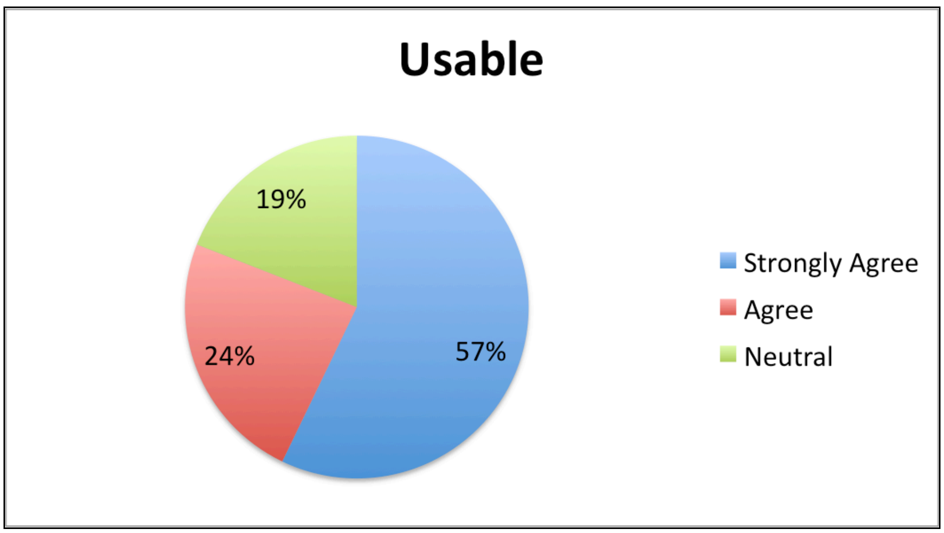

Figure 5.3.7 Usability of Experimental System

This data appears as if the participant responses were completely in support of the experimental system using the speech interaction method. Additional analysis was done to determine if there was evidence that this data showed statistical significance in favor of the experimental system. Table 5.3.1 outlines the results of the analysis of the participant responses to the aforementioned questions. Initially a one-sample t-test was performed on the data, to determine if it was significant that the responses were greater than three on the Likert scale.

However, since the data for each question did not follow a normal distribution (see Shapiro-Wilk normality test column), an additional nonparametric test was needed. The Signed Rank Wilcoxon test was used due to the fact that it is a more robust test than the Student's t-test. Results show that, for each question posed to the participants, the p-value is much lower than the conventional 0.05 level of significance, thus sufficient evidence has been provided to reject the null. This confirms the initial analysis that there is a high probability that the system has a response rating of "Strongly Agree" or "Agree". Table 5.3.1 also shows the analysis results for the overall participant perception of user satisfaction. Using the Signed Rank Wilcoxon test, the resulting p-value is less than 0.0001, indicating that the overall consensus of the experimental system is rated higher than 3.

User Satisfaction Statement	t-test p-value	Shapiro-Wilk (normality test)	Signed Rank Wilcoxon p-value
1. This method is easy to use.	0.0005	0.0035	0.0023
2. It was easy to understand the instructions.	0.0009	0.0067	0.0036
3. It was easy to correct my spelling mistakes.	< 0.0001	0.0008	0.0001
4. I feel that I made selections privately.	< 0.0001	< 0.0001	< 0.0001
5. This method should be used for voting during elections.	< 0.0001	< 0.0001	0.0001
6. Usable	< 0.0001	< 0.0001	< 0.0001
Overall User Perception	< 0.0001	< 0.0001	< 0.0001

Table 5.3.1 Speech Interaction Method - Experimental System User Satisfaction Analysis

5.3.2 Switch Interaction – Experiment vs. Control

Upon completing tasks on each system, the participants of the switch interaction group completed a questionnaire regarding their experience with the system. The data presented here is a comparison of the responses about the control and experimental systems. The survey data reported in this section are responses from the same questions reported in section 5.3.1:

1. This method is easy to use.
2. It was easy to understand the instructions.
3. It was easy to correct my spelling mistakes.
4. I feel that I made selections privately.
5. This method should be used for voting during elections.

A summary of the responses to these questions, for both the experimental and control systems, are depicted in the side-by-side comparison plots of Figures Figure 5.3.8 through Figure 5.3.13. For this analysis, where participants marked "Strongly Agree" or "Agree" to the previous statements are considered affirmative responses. For the first statement (see Figure 5.3.8), the majority of participants thought both systems were easy to use, with 84 percent of participants responding affirmatively about the experimental system, and 89 percent responding affirmatively about the control system. Likewise, a majority of the participants affirmed that it was easy to understand the instructions on both systems; 84 percent for the experimental and 95 percent for the control (see Figure 5.3.9).

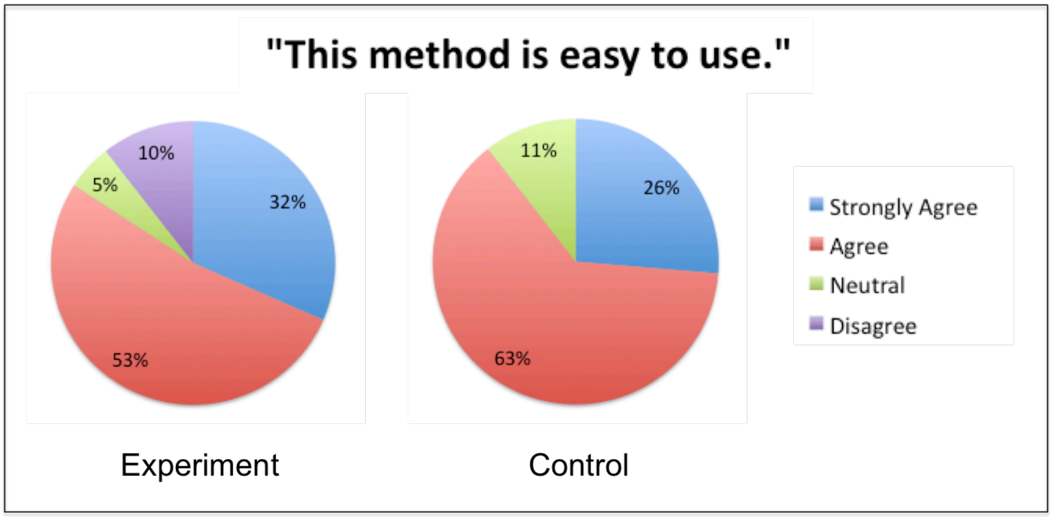

Figure 5.3.8 Ease of Use Comparison

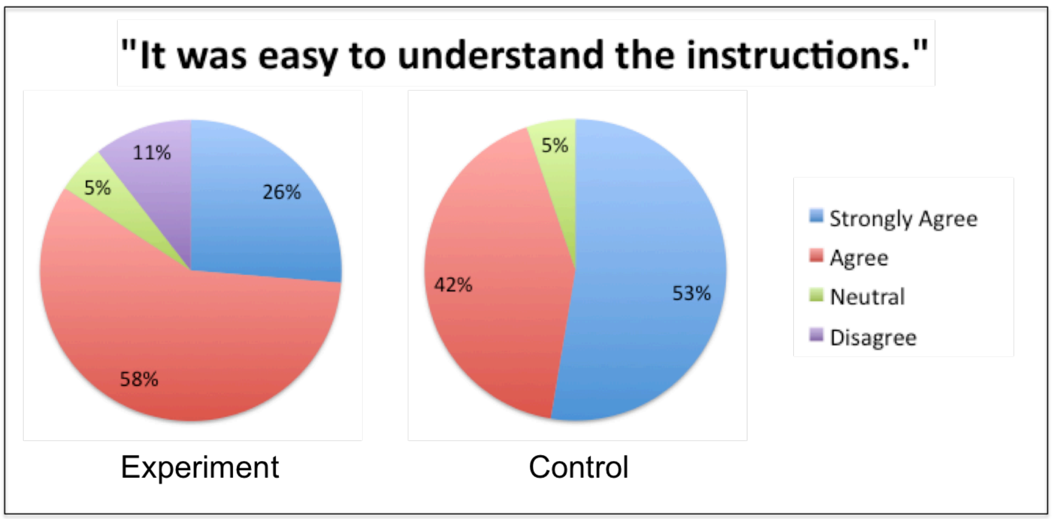

Figure 5.3.9 Comparison of System Instruction Understandability

In response to the third statement, participants thought that it was easier to correct mistakes with the experimental system, 63 percent affirmative, than the control system, 58 percent affirmative (see Figure 5.3.10). Participants indicated that they thought both systems were private (see Figure 5.3.11), but more indicated affirmation that the experimental system should be used in elections, 84 percent, than the control system, 74 percent (see Figure 5.3.12).

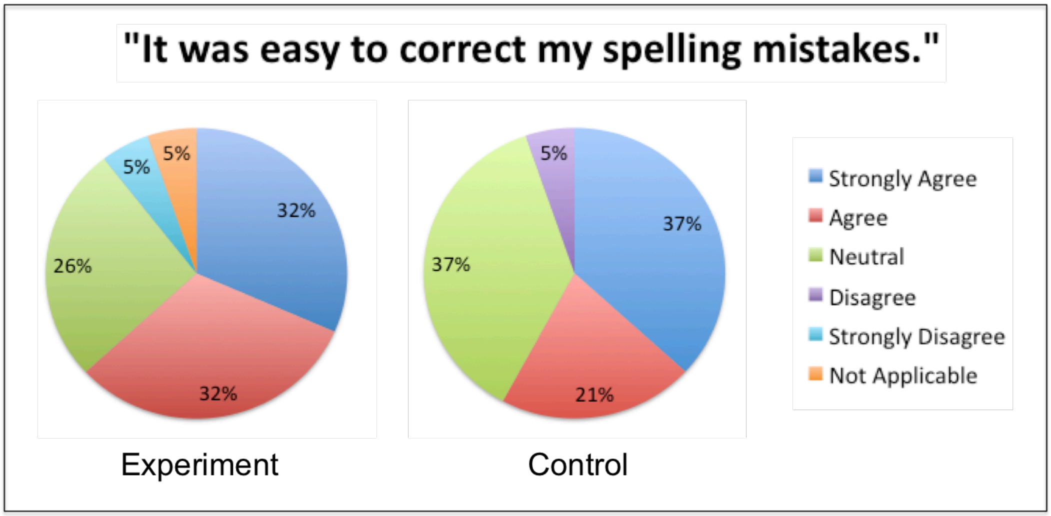

Figure 5.3.10 Ease of Error Correction Comparison

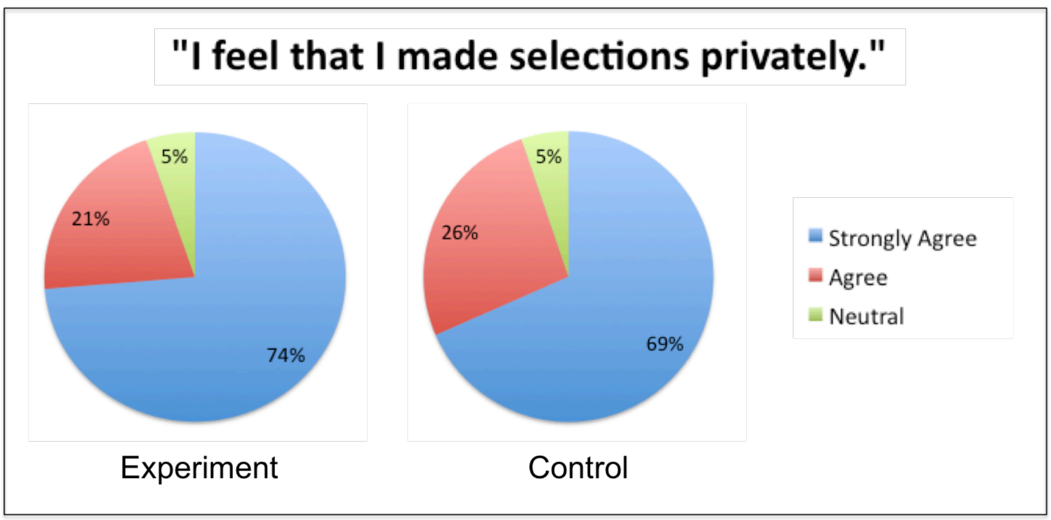

Figure 5.3.11 System Interaction Privacy Comparison

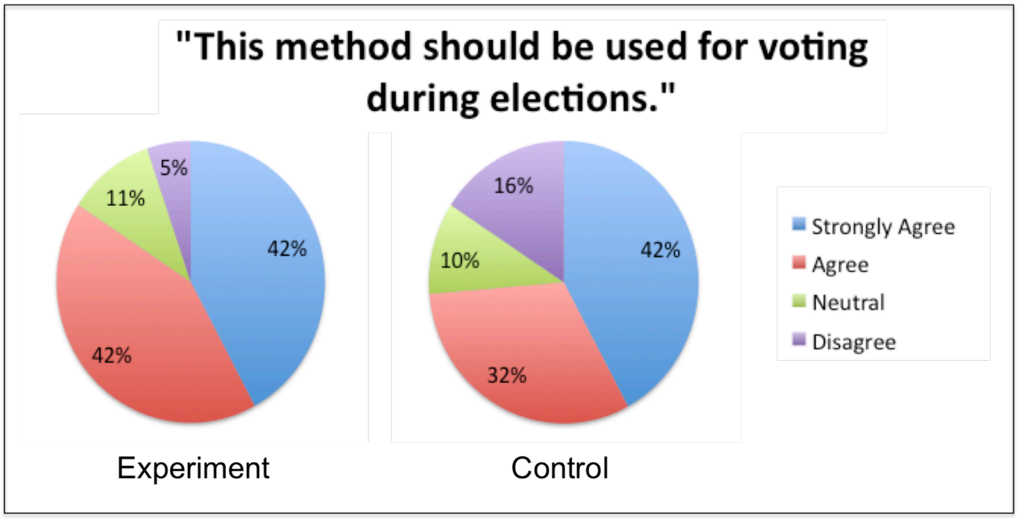

Figure 5.3.12 Opinion of Use in Election Comparison

Lastly, participants were asked on a Likert scale if they felt that the systems were usable; 5 was used to indicate usable, 1 for not usable. None of the participants indicated a negative response to the usability of either system (see Figure 5.3.13). All participants indicated that both systems were usable, with the exception of the experimental system, for which 10 percent indicated neutral.

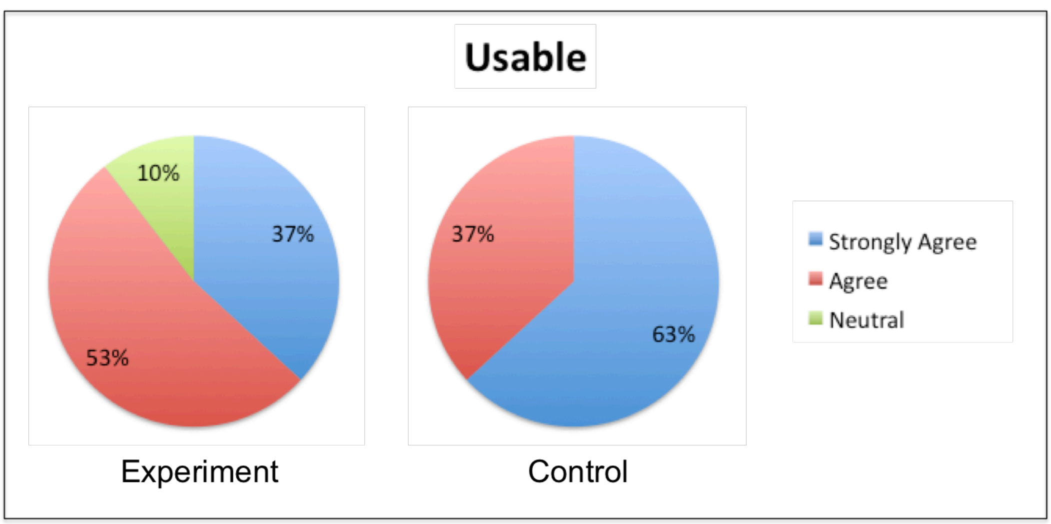

Figure 5.3.13 System Usability Comparison

An initial look at the data has the appearance that there is very little, if any, difference in the overall user perception of the experimental and control systems using the switch interaction method. The experimental system may have some advantages over the control system, but the initial analysis is inconclusive. Therefore, additional analysis was conducted to determine if there was any statistical difference in the user satisfaction between the two systems.

Table 5.3.2 outlines the results of the analysis of the question responses presented previously. Initially, a t-test was performed on the data, resulting in the p-values of column two of the table. These p-values for each observation statement, with the exception of statements two and six, were well above the conventional level of significance, 0.05, and therefore, there was not enough evidence to show a statistical difference between the user perception of the two systems. Statements two and six resulted in p-values of 0.0462 and 0.0150, respectively, which was enough to reject the null and determine that there was a statistical difference in the user perception of the systems. However, as previously mentioned, the t-test is not a robust test, and for that reason, the Shapiro-Wilk test for normality was also performed on the data.

As shown in the third column of Table 5.3.2, the results data for each statement were proven to not follow a normal distribution; each of the p-values is less than the 0.05 level of significance, therefore rejecting the null hypothesis of normality. For this reason, the more robust Signed Rank Wilcoxon test was used to analyze the data. For statements one through five, the resulting p-value was greater than the standard level of significance of 0.05, meaning there was not enough evidence to show a difference between the two systems. However, in response to the usability statement, there is significant evidence to reject the null, indicating that participants felt that the control system was more usable than the experimental system.

In addition to the individual statistical tests for each system perception statement, Table 5.3.2 shows the results of a combined analysis of the statement responses. Overall, there was insufficient evidence to show a difference in the user satisfaction between the two systems. Although hypothesis H5 was not confirmed for the switch interaction method, regarding user satisfaction, the experimental system is a viable alternative to the control system.

User Satisfaction Statement	t-test p-value	Shapiro-Wilk (normality test)	Signed Rank Wilcoxon p-value
1. This method is easy to use.	0.6065	0.0014	0.5547
2. It was easy to understand the instructions.	0.0462	0.0004	0.0781
3. It was easy to correct my spelling mistakes.	0.8344	0.0025	0.9375
4. I feel that I made selections privately.	0.5778	< 0.0001	1.000
5. This method should be used for voting during elections.	0.4477	0.0384	0.3828
6. Usable	0.0150	< 0.0001	0.0313
Overall User Perception	0.1496	< 0.0001	0.1698

Table 5.3.2 Switch Interaction Method – Experiment vs. Control User Satisfaction Analysis

5.3.3 Limitations of Experiment

Due to the observational nature of the experiment, the environmental and setup were not as realistic as intended for the study. For this reason, many things may have attributed to the outcome of the results of the post-questionnaire. The first is that speakers were used for the output of spoken prompts, whereas headphones would be used in an actual election scenario. The participants were informed of this reality during the study; however, their actual experience listening to the voice over the speakers may have negatively influenced their opinion of the privacy of the system. Also explained to the students was that, during an election, the headphones used would have a microphone attachment. For the experiment, those participants who were a part of the speech interaction group used the built-in microphone on the laptop computer, causing some of the participants to project their voice more than would be necessary when using a microphone headset.

Along the same lines of speech and auditory interaction, one major complaint about both systems used in the experiment was the computer-generated voice. The text-to-speech synthesizer used in the experiment was not the most pleasing to hear, but it was intended to be effective enough to get the point across. Of the 59 post-questionnaires completed by participants, 36 percent indicated in the additional comments that the computer voice needs some sort of change. Participants may have rated the system based on the way it sounded rather than what it did. Not only may this have had an effect on the user satisfaction results, but it may have also affected the effectiveness and efficiency metrics. For example, as discussed in section 5.2.1, participants may have selected the letter 'Q' when intending to select the letter 'U'. When spoken individually by the system, these two letters, and many other phonetically similar letters, sound alike. This incorrect letter selection may have negatively affected the completion rate and efficiency for a given participant.

Combined with a robotic sounding computer voice, a few participants showed concern for the speed of the spoken prompts. This is a downside of requiring participants to utilize their senses in ways that they would not normally use them. People who have visual impairments often rely on their sense of hearing and sense of touch much more than those without any visual impediments. As presented in [17], users who are visual impaired, unlike the participants in this experiment, often want to increase the speech output of the same spoken prompts.

The hardware utilized in this experiment was not a foreseen complication during the design of the experiment. The switch had been tested prior to the study, and seemed to work perfectly with the system. However, a few participants had difficulties properly operating the switch buttons. These participants would hold down the button, skipping through several prompts at once, often skipping over what they intended to select. Those that had this problem spent nearly the first half of their tasks figuring out how to use it properly, despite being told not to hold down the switch button. Unfortunately, this may have also had a negative impact on the data.

The final attributes that may have had a negative impact on the post-questionnaire and performance results were independent of the system and equipment, and were more related to the participants themselves. As reported in section 5.1, there was a participant with partial hearing loss, and participants who indicated that English was not their native language. While the portion of the system used in the experiment was not designed for those with hearing impairments, the participant was still able to complete the given tasks and post-questionnaire. Although this is the case, the results of the post-questionnaire and performance may have been affected. Of the participants that indicated that English was not their native language, two participants explained in the additional comments of the post-questionnaire that they had some trouble with both systems because they are not accustomed to the letter order of the English alphabet. Knowing the order of the letters in the English alphabet is a huge component of both the experimental and control systems. This combined with the hard-to-recognize computer voice may have definitely caused confusion during the tasks.

Chapter 6

Summary and Conclusion

6.1 Summary and Conclusion

The ultimate goal of electronic voting systems today should be to allow anyone to vote privately and independently using a single design. The VVSG (discussed in Chapter 2) provides useful and necessary guidelines to ensure that all eligible citizens have the same access when voting, regardless of a person's disabilities. The primary objective of this research was to embrace these guidelines by developing a system in which a person, regardless of ability or inability, can efficiently, anonymously, and independently write-in a candidate's name during an election. The method designed allows voters to spell a candidate's name discretely through multimodal interaction. This method uses a clustering and predictive approach in order for the voter to get through the voting process of writing-in a candidate's name quickly and accurately.

The objective of this research was evaluated by analyzing different methods of writing-in a candidate's name. The evaluation measures were the time taken to complete write-in tasks, accuracy of the task completion, and user perception of the write-in method used. Analysis of these three measures led to the determination of the predictive system's efficiency, effectiveness, and user satisfaction. The evaluation results suggest that the system is effective, given that 94 percent of all tasks were completed, efficient, with statistically significant evidence showing that voters can write-in names faster with the predictive system than with the linear systems in use today, and provides user satisfaction, with statistical significance that the overall user perception of the system is significantly above a mid-range neutral ranking. Overall, the system design conquers the issues of time, privacy, and accessibility.

6.2 Implications for this Research

Currently, there is no solution for writing in a candidate's name that is universally accessible. Current mainstream electronic voting systems simply cannot accommodate the range of voter abilities with their current techniques for writing-in a candidate's name. The work presented in this dissertation allows the write-in process to be private, secure, accessible, and usable for people with disabilities. As such, the immediate contributions of this research will directly benefit disabled voters during elections.

Not only is this work appropriate for those with disabilities; this design has an impact on those without disabilities as well. As discussed in Chapter 1, the recent voter intent issues in Alaska have been the source for controversy in write-in elections. In using this technology, voters and election officials would not need to deal with the hassles surrounding voter intent. These two major notions alone are reason enough for this system to have a national impact on elections and the voting process.

6.3 Directions of Future Research

There are many directions in which this research has the potential to go. In this dissertation, the system was aimed at making the write-in voting process accessible, while maintaining privacy and efficiency. As this method is further developed, other applications should be explored. This design can be used for applications that require the transfer of information privately and accessibly, e.g. mobile phone applications. Given the database querying aspect, it can also be adapted by certain applications in need of accessible search capabilities. Search applications that utilize a fixed directory will benefit greatly by using this prediction method, e.g. people finder directories, building directories, or telephony search systems.

As for the design of the research, there are some enhancements that can be made to improve the effectiveness and user satisfaction of the system. As first mentioned in section 5.3.3, many letters in the alphabet sound alike; namely, the letters ‘Q’ and ‘U’; ‘M’ and ‘N’; ‘C’ and ‘Z’. Although the system presents letters in alphabetical order, some participants in the experiment had trouble differentiating the letter pairs. Therefore, intended names with records in the database were not suggested to those participants. In these situations, the system can make better predictions if it knows these sound-alike letter groups, and therefore suggest the intended name even if the voter has selected incorrect letters. With this method, the effectiveness increases, because the voter can complete the spelling without starting over, even if one or two letters are incorrect. Another, and perhaps most recognized system enhancement is to employ a different text-to-speech engine. With better, more clear, more understandable speech output, user satisfaction would increase tremendously.

Bibliography

[1] 2005 Voluntary Voting System Guidelines (2005) [online]. http://www.eac.gov/voting%20systems/voluntary-voting-guidelines/2005-vvsg.

[2] 2007 VVSG Chapter 3: Usability, Accessibility, and Privacy Requirements (2007) [online]. http://www.eac.gov/vvsg/part1/chapter03.php/.

[3] About SQLite. http://www.sqlite.org/about.html.

[4] Advanced Voting Solutions, Inc. (FKA: R.F. Shoup Corporation (2008). http://votingmachines.procon.org/view.source.php?sourceID=1114

[5] Alabama Code (1975) [online]. http://alisondb.legislature.state.al.us/acas/CodeOfAlabama/1975/coatoc.htm.

[6] Alabama Election Code 17-6-27 (1975) [online]. http://alisondb.legislature.state.al.us/acas/CodeOfAlabama/1975/17-6-27.htm.

[7] Alabama Election Code 17-6-28 (1975) [online]. http://alisondb.legislature.state.al.us/acas/CodeOfAlabama/1975/17-6-28.htm.

[8] Alabama Official Election Center (2010). 2010 November General Election Sample Ballots. http://alabamavotes.gov/ElectionInfo/2010SampleBallots.aspx?a=voters.

[9] Alaska State Law (2010) AS 15.15.360 (10-11) [online] http://www.legis.state.ak.us/basis/folio.asp.

[10] Apple MacBook. http://www.apple.com/macbook/.

[11] Brace, K. (October 2008). 2008 Voting Equipment Study. Election Data Services. http://www.electiondataservices.com/images/File/NR_VoteEquip_Nov-2008wAppendix2.pdf.

[12] Butler, R., (2005). http://names.mongabay.com/.

[13] Butts, L. and Cockburn, A., (2001). An Evaluation of Mobile Phone Text Input Methods, Third Australasian User Interfaces Conference, Melbourne, Australia, Conferences in Research and Practice in Information Technology, Vol. 7, pp.55-59. [online] http://www.acs.org.au/documents/public/crpit/CRPITV7Butts.pdf.

[14] California Code (2009) 15340-15342 [online]. http://www.leginfo.ca.gov/cgi-bin/displaycode?section=elec&group=15001-16000&file=15340-15342.

[15] California Code (2009) 8600-8605 [online]. http://www.leginfo.ca.gov/cgi-bin/displaycode?section=elec&group=08001-09000&file=8600-8605.

[16] Center for Universal Design, N.C. State University (2004). Universal Design Education Online Web Site http://www.udeducation.org/learn/index.asp.

[17] Cross, E. V., Dawkins, S., McClendon, J., Sullivan, T., Rogers, G., Erete, A. & Gilbert, J. E. **Everyone Counts: Voting Accessibility**. *In Stephanidis, C., editors, Proceedings of Human Computer Interaction International*, Vol. 5616 of Lecture Notes in Computer Science, pp. 324-332. Springer-Heidelberg, 2009.

[18] Eclipse. http://www.eclipse.org/.

[19] Election Inspectors' Guide for AutoMARK Voter Assist Terminal (VAT) (June 2006). Michigan Department of State Bureau of Elections. http://www.michigan.gov/documents/AutomarkElectionInspectorsGuide_163310_7.pdf

[20] Election Preview 2008 (October 2008). PEW Center on the States. http://www.pewcenteronthestates.org/uploadedFiles/Election Preview FINAL.pdf

[21] Erickson, A. C. (1980). Robson Square, Vancouver, British Columbia. Courtesy of Davis, H. http://www.greatbuildings.com/cgi-bin/gbi.cgi/Robson_Square.html/cid_1820069.html

[22] ES&S AutoMARK (2010). Election Systems & Software. http://www.essvote.com/HTML/products/automark.html

[23] Gill, K., An Illustrated History of Voting Methods and Systems. http://uspolitics.about.com/od/elections/ig/History-of-Voting-Systems/.

[24] H2 Database Engine. http://www.h2database.com/html/main.html.

[25] Helm, B., (2004). Ins and Outs of Write-Ins [online]. http://www.businessweek.com/bwdaily/dnflash/nov2004/nf2004112_5680_db038.htm.

[26] Help America Vote Act. (2002). Public Law 107-252, 107th Congress, United States [online]. http://www.fec.gov/hava/law_ext.txt.

[27] History of the Paper Ballot [online]. http://www.fairvote.org/righttovote/pballot.pdf.

[28] HSQL Database Engine. http://hsqldb.org.

[29] Independent, Secret and Verifiable: A Guide to Making Voting an Independent and Accessible Process for People Who Are Blind and Visually Impaired (September 2002). American Council of the Blind. http://www.acb.org/resources/votingbook1.html

[30] Institute for Human Centered Design (2008). http://www.adaptiveenvironments.org/index.php?option=Content&Itemid=3.

[31] ISO 9241-11:1998. International Organization for Standardization. [online] http://www.iso.org/iso/catalogue_detail.htm?csnumber=16883.

[32] Jones, D., (2003). A Brief Illustrated History of Voting [online]. http://www.cs.uiowa.edu/~jones/voting/pictures/.

[33] KDE Construction (2010). http://www.kdeconstruction.com/.

[34] Klund, J., Novak, M. (2001). If Word Prediction Can Help, Which Program Do You Choose? [online]. http://trace.wisc.edu/docs/wordprediction2001/.

[35] Learn to Use T9 (2007). http://www.t9.com/us/learn/.

[36] Lenth, R. V. (2006-9). Java Applets for Power and Sample Size. Retrieved March 2010, from http://www.stat.uiowa.edu/~rlenth/Power

[37] MacKenzie, I.S., Kober, H., Smith, D., Jones, T., Skepner, E., (2001). LetterWise: Prefix-Based Disambiguation for Mobile Text Input, Proceedings of the 14th annual ACM symposium on User interface software and technology, November 11-14, 2001, Orlando, Florida [online]. http://www.eatoni.com/research/lw-mt.pdf.

[38] Making Sure Votes Count and Are Counted (2009). http://holt.house.gov/voting.shtml.

[39] Maryland Election Code 8-205 (2009) [online]. http://www.michie.com/maryland/lpext.dll/mdcode/d676/d8fe/d910/d925?f=templates&fn=document-frame.htm&2.0#JD_el8-205.

[40] Maryland Election Code 5-303 (2009) [online]. http://www.michie.com/maryland/lpext.dll/mdcode/d676/d7a6/d7c8/d7d3?f=templates&fn=document-frame.htm&2.0#JD_el5-303.

[41] Maryland Election Code 5-704 (2009) [online]. http://www.michie.com/maryland/lpext.dll/mdcode/d676/d7a6/d80d/d822?f=templates&fn=document-frame.htm&2.0#JD_el5-704.

[42] Maryland State Board of Elections – Candidacy (2010). http://www.elections.state.md.us/candidacy/

[43] Maryland State Board of Elections. Write-In Candidates (2010). http://www.elections.state.md.us/candidacy/documents/Writein_packet.pdf.

[44] Mauer, R. (2010). Miller Files Suit in State Court Over Senate Vote Count. Anchorage Daily News [online] http://www.adn.com/2010/11/22/1567823/miller-files-vote-suit-in-state.html#ixzz169P30wmY.

[45] Miller v. State of Alaska, Division of Elections (2010) [online] http://media.adn.com/smedia/2010/11/22/16/47-1.source.prod_affiliate.7.pdf

[46] NBC for Southeast Alaska (2010). 2010 Alaska General Election KATH-TV Write-in Ballot Poll. http://www.kath.tv/?page=writeinballots2010.

[47] North Carolina State University Center for Universal Design (2008). http://www.design.ncsu.edu/cud/about_ud/about_ud.htm.

[48] Nuance – T9 Text Input (2009). http://www.nuance.com/t9/textinput/.

[49] Nuance – XT9 Smart Input (2009). http://www.nuance.com/t9/xt9/.

[50] Official Election Site of San Mateo County (2007). http://www.shapethefuture.org/press/2007/090607.asp.

[51] Patriot Voting System (2002). UniLect Corporation. http://www.unilect.com/2002/unilect.html

[52] Polling Place and Vote Center Management (2009) [online]. http://www.eac.gov/election/quick-start-management-guides/docs/09-polling-place-and-vote-center-management.pdf/attachment_download/file.

[53] Press Release: 2005 VVSG Adopted (2005). http://www.eac.gov/voting%20systems/docs/eac-adopts-2005-voluntary-voting-system-guidelines.pdf/attachment_download/file.

[54] Prime III: One Machine, One Vote for Everyone. http://primevotingsystem.org/.

[55] Punch Card Voting. Cape Girardeau County Website. http://www.capecounty.us/ElectionInfo/Punch Card Voting.aspx.

[56] Punchcard Voting Instructions. http://www.idahovotes.gov/VoterGuide/instr_punch.htm.

[57] Registration & Voting (2009). Board of Elections in the City of New York. http://www.vote.nyc.ny.us/pdf/documents/boe/fourlanguages/registrationandvotingenglish.pdf

[58] Summary of Qualifications and Requirements for Write-In Candidates (2003). http://www.sos.ca.gov/elections/gov-qual_write-in.pdf.

[59] State of Alaska Division of Elections (2010). General Election Results. http://www.elections.alaska.gov/results/10GENR/indexWI.shtml

[60] State of Alaska 2010 General Election Official Results (2010) [online]. http://www.elections.alaska.gov/results/10GENR/data/resultsWI.htm

[61] Stewart, W. (November 2007). Pennsylvania: End of the Road for AVS?. Verified Voting Foundation, Inc. http://www.verifiedvotingfoundation.org/article.php?id=6533

[62] Technical Guidelines Development Committee (2007). Voluntary Voting System Guidelines Recommendations to the Election Assistance Commission [online] http://www.eac.gov/files/vvsg/Final-TGDC-VVSG-08312007.pdf.

[63] Trnka, K., McCaw, J., Yarrington, D., McCoy, K., and Pennington, C. 2009. User Interaction with Word Prediction: The Effects of Prediction Quality. ACM Trans. Access. Comput. 1, 3 (Feb. 2009), 1-34. [online]. http://doi.acm.org/10.1145/1497302.1497307

[64] United States Census Bureau (2011) [online]. http://factfinder.census.gov/

[65] Valid Concept (2009). T9: Text on Nine Keys [online]. http://www.validconcept.com/articles-t9.html.

[66] Voluntary Voting System Guidelines Recommendations to the Election Assistance Commission (2007) [online]. http://www.eac.gov/files/vvsg/Final-TGDC-VVSG-08312007.pdf.

[67] Voter Confidence and Increased Accessibility Act of 2009 (2009). Public Law H.R. 2894. [online] http://thomas.loc.gov/cgi-bin/query/z?c111:H.R.2894:.

[68] What is word prediction versus word completion? (2009). http://www.aurora-systems.com/pages/faqpred.html#WPVSWC.

[69] Wolf, R. (October 2008). Voting Equipment changes could get messy on Nov. 4. http://www.usatoday.com/news/politics/election2008/2008-10-28-votingequipment_N.htm - table.

[70] Write-in Candidate. http://dictionary.reference.com/browse/write-in+candidate.

[71] Write In Voting Judges Manual (2008) [online]. http://www.co.ho.md.us/BOE/BOEdocs/Howard_Election_Judges_Manual_2008_Ch05_Approved.pdf.

Appendix 1

Pre-Questionnaire

Pre-Questionnaire

ID: 1

Age: ________

Gender: ☐ Male ☐ Female

Race / Ethnicity:

☐ Black ☐ White ☐ Asian ☐ Native American ☐ Hispanic ☐ Other ________________

Citizenship: ________________________

Highest Degree Obtained (High School, Bachelor's, Master's, Doctorate, etc): __________________

Do you have any disabilities? ☐ Yes ☐ No

If yes, please explain: ____

Is English your native language? ☐ Yes ☐ No

Appendix 2

Data Collection Sheet

Data Collection Sheet			
Generated ID	Name	Time	Comments

Appendix 3

Preliminary Study Results

ID	Name	Actual Times			Calculated Times					
		Prediction Method			Prediction Method			Linear Method		
		Actual Time	# Letters	Average Time per Letter	Time Calcuated - First Name	Time Calcuated - Last Name	Calculated Best Case Time	Time Calcuated - First Name	Time Calcuated - Last Name	Calculated Best Case Time
6	Ciaphas Cain	12.18	11	1.11	2.08	2.08	4.28	8.03	4.13	12.28
7	Brittany Abrams	13.53	14	0.97	1.98	1.89	3.99	13.40	7.41	20.93
8	Jack Jones	11.10	9	1.23	2.27	2.27	4.67	3.93	7.95	12.00
9	Bob Jones	6.28	8	0.79	1.98	2.27	4.38	3.02	7.95	11.08
10	Tony Green	7.06	9	0.78	2.18	1.98	4.28	8.77	6.59	15.48
11	Reese Witherspoon	21.19	16	1.32	2.08	2.10	4.30	6.88	19.53	26.53
12	Lori Woodfin	11.45	11	1.04	1.98	2.10	4.20	6.74	10.84	17.70
13	Bob Billy	11.34	8	1.42	1.98	1.98	4.09	3.02	7.66	10.79
14	Stephen Wilson	4.42	13	0.34	2.18	2.10	4.40	10.94	11.09	22.14
15	Roger Davison	13.48	12	1.12	2.08	2.18	4.38	7.95	10.65	18.71
17	Bob Barker	10.05	9	1.12	1.98	1.98	4.09	3.02	7.04	10.18
18	Bill James	6.32	9	0.70	1.98	2.27	4.38	4.90	6.49	11.51
19	Mickey Mouse	18.16	11	1.65	2.08	2.08	4.28	8.57	8.91	17.60
20	Joe Smith	10.58	8	1.32	2.27	2.18	4.57	4.08	8.53	12.73
21	Jonathan Dunn	11.29	12	0.94	2.27	2.18	4.57	12.14	6.64	18.91
22	James Harmon	7.50	11	0.68	2.27	2.08	4.47	6.49	8.86	15.47
23	Daffy Duck	14.25	9	1.58	2.18	2.18	4.47	5.91	5.29	11.32
24	Daniel Fisher	7.29	12	0.61	2.18	1.89	4.18	6.54	8.47	15.13
25	Guitar Bob	12.26	9	1.36	1.98	1.98	4.09	9.54	3.02	12.68
26	Charles Griffin	7.18	14	0.51	2.08	1.98	4.18	8.90	9.19	18.22
27	Super Man	8.45	8	1.06	2.18	2.08	4.38	9.49	3.89	13.50
28	Daniel John	6.37	10	1.22	2.18	2.27	4.57	6.54	6.06	12.72
29	George Bush	9.47	10	1.35	1.98	1.98	4.09	7.70	6.35	14.17
30	Davis Darwin	11.21	11	1.01	2.18	2.18	4.47	7.17	8.86	16.15
31	Sarah Smith	6.11	10	0.63	2.18	2.18	4.47	6.40	8.53	15.04
32	Kathleen Dulaney	13.16	15	0.47	1.89	2.18	4.18	10.20	10.45	20.78
33	Lane Otto	8.25	8	2.65	1.98	2.27	4.38	4.42	8.29	12.83
34	John Doe	7.35	7	1.64	2.27	2.18	4.57	6.06	3.50	9.68
35	Barack Obama	18.21	11	1.03	1.98	2.27	4.38	5.66	4.94	10.73
36	Charles Haro	9.43	11	0.40	2.08	2.08	4.28	8.90	5.58	14.60
37	Mickey Mouse	8.53	11	1.23	2.08	2.08	4.28	8.57	8.91	17.60
38	Kali Wilson	8.29	10	1.01	1.89	2.10	4.11	4.71	11.09	15.91
39	John Doe	8.38	7	0.90	2.27	2.18	4.57	6.06	3.50	9.68
41	Rob Herring	5.09	10	1.82	2.08	2.08	4.28	4.57	10.16	14.85
42	Taylor Rogers	5.27	12	0.88	2.27	2.08	4.47	10.99	10.12	21.23
43	Wayne Kelley	5.40	11	1.03	2.10	1.89	4.11	7.64	8.96	16.71
44	Bob Jones	6.12	8	0.94	1.98	2.27	4.38	3.02	7.95	11.08
45	Bob Smith	6.31	8	1.78	1.98	2.18	4.28	3.02	8.53	11.66
46	Gabriel McKenzie	5.27	15	0.49	1.98	2.08	4.18	7.74	11.17	19.03
47	John Jones	7.23	9	1.36	2.27	2.27	4.67	6.06	7.95	14.13
	Average Time	9.52	10.43	1.09	2.10	2.12	4.33	6.94	8.03	15.09
	Standard Deviation	3.83	2.22	0.45	0.12	0.11	0.17	2.58	2.88	3.86
	Median	8.42	10.00	1.04	2.08	2.10	4.34	6.64	8.12	14.73

Appendix 4

Prompt Sequences

Prompt Number	**Prompt**	**Prompt Type**	**Interaction Cycle Time (seconds)**
1	You will now be prompted to spell the candidate's {first, last} name.	Which Name	3.90
2	You will now select the {first, second, etc.} letter of the candidate's {first,last} name.	Which Position	3.77
3	The next letters are the most common letters. Say vote if the {first, last} letter of the candidate's {first, last} name is {AEO, CAE, etc}.	Most Common Letter Cluster	8.75
4	Say vote if the {first, last} letter of the candidate's {first, last} name is {ABCDE, FGHIJ, KLMNO, PQRST}.	5 Letter Cluster	6.40
5	Say vote if the {first, last} letter of the candidate's {first, last} name is {UVWXYZ}.	6 Letter Cluster	7.51
6	Say vote if the {first, last} letter of the candidate's {first, last} name is {letter}.	Individual Letter	5.81
7	Say vote to delete this letter.	Delete	6.00
8	You have selected {letter} as the candidate's {first, last} name. Say vote if you are finished spelling the candidate's {first, last} name.	Check if Finished After 1 Letter	10.22
9	You have selected {letter} as the candidate's {first, last} name. Say vote if you are finished spelling the candidate's {first, last} name.	Check if Finished After 2 Letter	10.33

10	You have selected {letters} as the candidate's {first, last} name. Say vote if you are finished spelling the candidate's {first, last} name.	Check if Finished After Letters already spelled (>2)	10.24
11	Say vote if the candidate's {first, last} name is {name}	Suggestion	5.72
12	You have chosen the name {name}. Say vote if this is incorrect.	Name Confirmation	6.68
13	You have chosen the name {first name, last name}. Say vote if this is incorrect.	Full Name Confirmation	7.19
14	You have already selected the letters {letters}	Reminder	2.62 + (.24 for every letter already selected)

Method	**Sequence**
Prediction Method Best Case	[1-2-(4 or 5)-6-7-8-2-3-6-7-9-2-14-3-14-6-7-11-12]* 13 *repeat for first name
Linear Method	{1-(2-6)*-7-8-(2-6)*-7-9-(2-14-6)*-7-10-[(2-14-6)*-7-10]** 12}*** 13 *repeat until prompt for desired letter **repeat until all letters selected ***repeat for first name

Appendix 5

Post-Questionnaire

Post-Questionnaire

ID:

Method Used:

Desired Name:

Time:

In Database (yes/no):

Chose Suggestion (yes/no):

Please mark the number that best reflects your reaction to write-in method:

Terrible..Wonderful

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Frustrating..Satisfying

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Not Usable..Usable

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

This method is easy to use.

☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

It was easy to understand the instructions.

☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

I was able to complete the spelling of the name I intended.

☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

I was able to correctly spell the name I intended.

☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

It was easy for me to correct my spelling mistakes.

☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

I feel that I made selections privately.

☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

I feel that no bystanders would be able to determine the name I spelled.

☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

This method should be used for voting during elections.

☐ Strongly Agree ☐ Agree ☐ Neutral ☐ Disagree ☐ Strongly Disagree

Comments: ____

____

____

____

____

Suggestions: ____

____

____

____

____

Appendix 6

Database Table “highProbability” Contents - Full Names

1. ALAN HUMPHRIES
2. ALLISON DRAKE
3. ANTHONY RICE
4. BEN SAWYER
5. BILL KLEMME
6. BOB GRIFFIN
7. BRADLEY PORTER
8. BRIAN LEON
9. CARITA BACKMAN
10. CARL HARDIN
11. CARL HOLMAN
12. CARL REMLEY
13. CAROLINE MCMULLINS
14. CHAD RENFRO
15. CHAD SOLBERG
16. CHERYL HAAKENSON
17. CHIMENE MILLS
18. CHRIS DILLINGHAM
19. CHRISTOPHER ERICKSON
20. CINDY JOHNSON
21. CLIFF MURRAY
22. CULLEN REECE
23. CURTIS CHAMPAGNE
24. DANA DAVIS
25. DANIEL EZELL
26. DANIEL PIASKOWSKI
27. DARAE CREWS
28. DARRELL HORTON
29. DARYL ANDERSON
30. DAVID ARDUSER
31. DAVID BEEMAN
32. DAVID BOYLE
33. DAVID FOGLER
34. DAVID GEORGE
35. DAVID JENKINS
36. DAVID MARCH
37. DAVID ROBERTS
38. DEANNA MARCH
39. DERRICK HUNZIKER
40. DEVONE HASARA
41. DOMINIC HASARA
42. DONA GROSSMAN
43. DONALD DORSEY
44. EDWARD MARTIN
45. EILEEN RANSOM
46. ELIZABETH CHERRY
47. ELVA BETTINE
48. ERIC FRANKLIN
49. ERIC KELLY
50. ERICA DVORAK
51. ERMA DOGGETT
52. FRED COLSON
53. FRED HANDY
54. FREDRICK HAASE
55. GALE DOGGETT
56. GARRET VANECK
57. GARY COX
58. GEORGE GREENE
59. GEORGIA TOLBERT
60. GLENN GILLETTE
61. GLENN SWAN
62. GREGORIO APODACA
63. GREGORY ERKINS
64. GREGORY PURVIS
65. GROVER JOHNSON
66. GUY CUMMINS
67. HOLLY KJOSTAD
68. HOWARD SHANKS
69. IRENE REPPER
70. JAMES DUFFIELD
71. JAMES GARHART
72. JAMES THATCH
73. JAMES WOOSLEY
74. JANET HENRY

75. JANICE QUINN
76. JASMINE MILES
77. JAY HAMACHER
78. JED WHITTAKER
79. JEFF ARENTZ
80. JEFFREY WOOD
81. JERRY WARD
82. JIMMIE JANEWAY
83. JOE ERICKSON
84. JOE MILLER
85. JOE MORAWITZ
86. JOHN COLLINGE
87. JOHN DAVIS
88. JOHN FRANCIS
89. JOHN KUKLIS
90. JOHN SMART
91. JOSHUA HOLLAND
92. JULIE HOLBROOK
93. KAREN DEARDORFF
94. KAREN HORTON
95. KAREN PERRY
96. KATHERINE HICKS
97. KATHY ZUREK
98. KEITH BARKWOOD
99. KELLY WALTERS
100. KEN BULLARD
101. KEVIN AUSTIN
102. KEVIN HITE
103. KEVIN LUCE
104. KIM THIBODEAUX
105. KIMBERLY WILSON
106. KORY BRADSHAW
107. KRIS CHERNIK
108. KRISTIN HOLLAND
109. LANCE ROBERTS
110. LAURA RASCHAL
111. LAVONNE BOYD
112. LAWRENCE AUSTIN
113. LEE HAMERSKI
114. LINDA BULLARD
115. LINDA EASON
116. LINDA VREM
117. LISA CRUSBERG
118. LISA LACKEY
119. LISA MURKOWSKI
120. LLOYD RUDD
121. LYN MARCUM
122. LYNETTE LARGENT
123. MAGGIE JOYNER
124. MARGARET CROWELL
125. MARJORIE LANDIS-BECK
126. MARK BILLS
127. MARK MCARTHUR
128. MARK RICHARDSON
129. MARSHA MARKSTROM
130. MARTHA MAYER
131. MARTIN WILLIAMS
132. MARTY GROSSMAN
133. MARTY VANDIEST
134. MARVIN MOSER
135. MATT MICHETTI
136. MAUREEN MEEKS
137. MELISSA PIKE
138. MERLE FRANK
139. MICHAEL ALEXANDER
140. MICHAEL AMES
141. MICHAEL BUTLER
142. MICHAEL DUNTON
143. MICHELLE BARNES
144. MICKEY MOUSE
145. NAN HOOPER
146. NICOLA TAYSOM
147. NORMAN STARKEY
148. PAMELA THATCH
149. PATRINA REMLEY
150. PAUL HILLING
151. PAUL MARKSTROM
152. PAULETTE EGGER
153. PETTER JOHNSON
154. RANDY ESTES
155. RED BRADLEY
156. RENE WEBER
157. RICHARD BRAUN
158. RICHARD BURNS
159. RICHARD JOHNSON
160. RICHARD KOLLER
161. RICHARD MCGAHAN
162. RICHARD REPPER
163. RICHARD STILLIE
164. ROBERT FLYNN
165. ROBERT WARTE
166. RODERIC PERRY

167. ROGER EGGER
168. ROGER PEARSON
169. RYAN WATERS
170. SAM PEPPER
171. SANDRA JOHNSON
172. SANDRA WILLIAMS
173. SCOTT MCADAMS
174. SCOTT SMITH
175. SHANNON FECHTNER
176. SID HILL
177. STEPHANIE DEVAULT
178. STEPHEN MOORE
179. STEVE GRAND
180. STEVE KELLY
181. SUSAN APLING-GILMAN
182. TAYLOR MOORE
183. TED GIANOUTSOS
184. THOMAS BAXTER
185. THOMAS DICUS
186. THOMPSON STEVE
187. TIGRAN ANDREW
188. TIM CARTER
189. TIM SOVDE
190. TIMOTHY WEIDENSEE
191. TOM M
192. TOM MOORE
193. TRACY VREM
194. TREVOR TEW
195. TYRAN DEVAULT
196. VERN CARLSON
197. VERONICA KEANAAINA
198. VICKY BEEMAN
199. WILLARD WAMSGANZ
200. WILLIAM AMIDON
201. WILLIAM BROWNFIELD
202. WILLIAM MERRILL
203. WILLIAM NEMEC
204. WILLIAM NILSSON
205. WILLIAM SMITH