2011

Cross-Language Mobile Communication And Pervasive Language Learning Using Multimedia Cellular Phone Messaging And Online Translation

Daniel Valencia
North Carolina Agricultural and Technical State University

CROSS-LANGUAGE MOBILE COMMUNICATION AND PERVASIVE LANGUAGE LEARNING USING MULTIMEDIA CELLULAR PHONE MESSAGING AND ONLINE TRANSLATION

by

Daniel Valencia

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Department: Electrical & Computer Engineering
Major: Electrical Engineering
Major Professor: Dr. Corey A. Graves

North Carolina A&T State University
Greensboro, North Carolina
2011

School of Graduate Studies
North Carolina Agricultural and Technical State University

This is to certify that the Master's Thesis of

Daniel Valencia

has met the thesis requirements of
North Carolina Agricultural and Technical State University

Greensboro, North Carolina
2011

Approved by:

_____
Dr. Corey A. Graves
Major Professor

_____
Dr. John C. Kelly
Department Chairperson

_____
Dr. Sanjiv Sarin
Dean of Graduate Studies

DEDICATION

I dedicate my thesis to my parents, wife and my son. They have motivated me to work hard and persevere in my goals. If I had to give support and love a meaning I would say it’s them.

BIOGRAPHICAL SKETCH

Daniel Valencia was born on May 15, 1985, in West Palm Beach, Florida. He received the Bachelor of Science degree in Electrical Engineering from North Carolina Agricultural and Technical State University. He has been affiliated with Eta Kappa Nu, Tau Beta Pi and Society of Hispanic Professional Engineers. His professional experience includes internships with Hewlett Packard and full time employment with Northrop Grumman. He is a candidate for the Master's of Science in Electrical Engineering.

ACKNOWLEDGMENTS

Dr. Corey Graves has inspired me to be an excellent engineer. His dedication to his students is outstanding and will never be forgotten. He has shown me how exciting engineering is and that there are no limits. I want to thank Dr. Christopher Doss and Dr. Jose Bravo De Rueda for being supportive and showing enthusiasm about my thesis, it motivated me to finish and work hard. Without all of my advisors support this would have been impossible.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES **PAGE**

LIST OF TABLES

TABLES **PAGE**

ABSTRACT

Valencia, Daniel. CROSS-LANGUAGE MOBILE COMMUNICATION AND PERVASIVE LANGUAGE LEARNING USING MULTIMEDIA CELLULAR PHONE MESSAGING AND ONLINE TRANSLATION. (**Advisor: Dr. Corey Graves),** North Carolina Agricultural and Technical State University.

Today within the US there is a gap between cultures, mainly Spanish speakers and English speakers. Communication is the main barrier and learning each other's language would be a solution. With this being said, the question is "How can technology be implemented to create a solution?" The proposed system is offered that allows people with different native languages to communicate using cell phones. Besides communication, learning is also a goal. This system implements the 3 most common styles of learning which are; visual, auditory and kinesthetic learning. These learning styles are aided by the ubiquitous and mobile learning paradigms. The system operates such that the users not only communicate with each other, but also can learn while doing so.

This system allows two users two speakers of different languages to communicate via basic cell phone SMS and MMS messaging, using text and audio. The proposed tool, Motilingual, is basically an intercessor that translates messages from both languages. It allows a user, who speaks a particular language, to communicate with another person who speaks a different language, by simply sending a text message to the other written in the sender's native language. The message is then translated to the receiver's language by an online translation engine. Additionally, an audio file of the message in the sender's language is generated using an open-source speech synthesizer. Finally, the translated

text message is sent to the receiver, along with the text and audio in the sender's language.

The advantage of using basic SMS and MMS messaging approach, rather than a smartphone application approach, is that it makes the tool accessible to a broader range of cell phone users. In other words, not everyone has a smartphone, but a majority of cell phone users have the option of using SMS and MMS services. The proposed system allows users to communicate with one another even though they do not speak each other's language. This system may help students learn in foreign language courses and allow professors to be more involved.

CHAPTER 1

Introduction

Today, Spanish is the second most used language in the United States [1]. Communication between English and Spanish speakers is a necessity, however, many recent immigrants and families living in isolated communities have a difficult time learning English, which is the language of Commerce and Education. Alternate reasons why Spanish-speakers have not been able to learn English sufficiently can be due to lack of resources (such as community-based classes) or lack of time due to long work weeks and since we live in a fast-paced society it is hard to complete tasks that are time consuming. This problem will continue to grow, since the Census Bureau forecasts continued growth of the Hispanic population from 47.8 million in 2010 to 102.6 million in 2050 [2].A language barrier exists which needs a solution. The acquisition of Foreign-language skills must be more available and affordable to the Native English and Spanish speakers, in order to foster better communication. This is why, in this research we propose a mobile to Multimedia Messaging Service-based language learning tool to facilitate the ubiquitous learning of a new language by two people engaging in a routine text communication dialogue.

1.1 Problem Statement

Throughout the United States, globalization has brought a wide variety of cultures together in a wide variety of settings. The ways in which people interact with one another

has greatly increased due to the ubiquitous use of electronic communications. “Different cultures” often implies “different languages”, which may prevent electronic communication from being used to its full potential at times for intercultural communication. Many people want to learn different languages but don’t have the time or money to do so. To learn different languages one can take classes at colleges, immerse oneself into different cultures or learn on one’s own through the use of resources like books and computer technology. To do any of these requires extra resources and time of a person.

Technology provides many advantages like mobility, ease of access, user-friendly interfaces, and affordability. These advantages make technology one of the best resources for learning, but not necessarily the most effective means of learning. In most cases it is still an extra task because the users need to take extra time to learn on their own time. Mobile devices are an attractive technological solution, since virtually everybody in the US has access to a cell phone (i.e. they are ubiquitous). In 2010, according to Cellular Telephone Industries (CTIA Wireless Association), there were over 290 million cell phone subscribers in the United States [3]. Integrating learning applications in smart phones would benefit new-language learners because applications are cheap and easy to access, the user has an opportunity to learn on the go, and the cell phone is already being used for communication. However, smart phones are expensive and often have expensive coverage plans, which can be one out of several reasons why only 29.7% of the U.S cell phone users has a smart phone, according to the Nielsen Company [4]. Every other user has a non-smart cellular phone (i.e. feature phone). Additionally, the fact that there are

many types of smart phones means many software development efforts. The question arises: “What is the best solution to communicate and learn at the same time?” Since Spanish is the second-most frequently spoken and studied language in the United States [5], Spanish will be the main focus of the proposed tool, which will allow the user to communicate and learn Spanish or English pervasively, without making it an extra assignment or chore. This can be achieved by using the cell phone’s multimedia messaging services. Both smart phones and feature phones have this capability. According to CTIA, over 1 trillion text messages were sent by people to communicate in 2009 [6]. The proposed tool will send messages in the sender’s language as text and as the audio. The translated text in the receiver’s language would also be sent. This may enable users to subliminally learn since they would read and hear the other language. The proposed tool can be used by any person that wants to communicate with others. It has been given the name “Motilingual” which is derived from motile and multi-lingual.

Current automated translation tools can only translate proper English or Spanish. As we know, in text messaging there is a short-hand lingo. This is used to shorten and speed up text messaging. In order to improve Motilingual’s translating capabilities an automatic translator will be created that converts the lingo to proper English or Spanish. The lingo will be translated so that users don’t have to change the way they communicate.

1.2 Ideas to Tackle the Problem

As discussed in the problem statement, the main idea of the project is to introduce a new way of learning a language using the cell phone. The cell phone is widely used all over the world and is gaining popularity. The cell phone allows people to communicate with each other, but only if they know the same language. Motilingual will allow people to communicate across two languages. The system should be compatible with most cell phones available in the market today. The key component of the cell phone that we will use is the multimedia messaging capability. This allows 290 million users in the US to have access to the proposed system. Some key features of the system are that, (1) it translates English to Spanish and vice versa, (2) it translates texting lingo in both languages to proper words in order to get an accurate translation, (3) it creates synthesized speech that it sends to the user to support learning not just by reading but also by hearing, (4) it also lets the user have access to dictionary definitions of words, (5) and allows users to message themselves for self learning.

This project proposes the design of an innovative system that allows people to learn English or Spanish as they use their cell phones for their normal, daily communication. The system allows an English and Spanish speaker to communicate with one another. The communication occurs by using Multimedia Messaging Service (MMS) and Short Messaging Service (SMS). These services are used by cell phones to transmit messages. Each user will send a message in their respective language and the other user will receive the translated message as text, as well as synthesized speech. To perform the translation, Google Translate is used [7]. Google Translate is a service provided by

Google Inc. to translate a section of text, or a webpage, into another language. To perform the speech synthesis, Espeak is used. Espeak is an open-source text-to-speech tool [8]. The idea is that while communicating, the user has the opportunity to learn ubiquitously. The system's goal is to help people learn a different language. The different styles of learning are incorporated into Motilingual and believed to make it a more effective system for learning.

Current commercial and researched applications will be analyzed to do a comparison. Some qualitative and quantitative tables will be developed for easy comparison of two systems. In this thesis the system will be compared to BabelWithMe since it is the most similar application. The applications specific components and usage will also be discussed. The remainder of this document is laid out as follows; Chapter 2 gives a background about different learning styles, commercial solutions and researched solutions, Chapter 3 talks about the proposed solutions and in depth analysis of the components of the system, Chapter 4 provides an experimental setup and data comparison to similar communication systems and Chapter 5 concludes the results and discusses future work.

CHAPTER 2

Background

Learning can occur anywhere and at anytime. It can be anything from how to cook or how to tie a shoe. The most common styles of learning are visual, auditory and kinesthetic [9]. These styles are considered traditional learning styles. Since technology is everywhere it provides leverage for learning. Technology can be used to learn and this is why Ubiquitous learning is still being researched today. Ubiquitous learning is composed of Mobile and Electronic learning [10]. Ubiquitous learning is important because this style integrates technology to everyday life of people and helps them learn without interruption.

2.1 Characteristics of Traditional Learning Styles

Traditional learning styles are considered to be visual, auditory and kinesthetic learning. The three styles are very different and each one can be characterized separately. A person can usually relate to one or more style than the others, but this does not mean he/she is limited to only one. Table 1 shows the characteristics of each different learning style. The idea is that the application covers the different styles to ensure it is capable of teaching every user. If the application only targeted one style then other users might not learn anything. Learning about the traditional styles was important because these guidelines are going to be followed.

Table 1. Traditional Learning Characteristics [11]

Learning Style	Characteristics
Visual Learning	• Uses visual objects such as, graphs, charts, pictures, and seeing information • Can read bodily language well and has a good perception of aesthetics • Able to memorize and recall various information • Tends to remember things that are written down • Learns better in lectures by watching them
Auditory Learning	• Retains information through hearing and speaking • Often prefers to be told how to do things and then summarizes the main points out loud to help with memorization • Notices different aspects of speaking • Often has talents in music and may concentrate better with soft music playing in the background
Kinesthetic Learning	• Likes to use the hands-on approach to learn new material • Generally good in math and science • Would rather demonstrate how to do something rather than verbally explain it • Usually prefers group work more than others

2.2 Characteristics of Ubiquitous Learning

Ubiquitous learning is a topic that is still under research. A proposed definition is a learning paradigm which takes place in a ubiquitous computing environment that enables learning the right thing at the right place and time in the right way [10]. Ubiquitous computing is a post-desktop model of human-computer interaction in which information processing has been thoroughly integrated into everyday objects and activities [11]. The authors in [10] propose Ubiquitous learning is a combination between electronic and mobile learning. Mobile learning has been defined as learning that takes place via wireless devices such as mobile phones, personal digital assistants (PDAs), or

laptop computers [12]. Electronic learning is simply an expansion of traditional learning buy using electronics.

Ubiquitous learning has several different characteristics: permanency, accessibility, immediacy, interactivity and context-awareness. Permanency means that the information remains unless the user removes it [10]. The learners do not lose their work, the system stores all of the user's data and can change personal files with the users input. Accessibility means that when the information is always available whenever the learner needs to access it [10]. The system is accessed via ubiquitous computing technologies and wireless networks. These technologies allow the users to access the application at any time. Immediacy implies when the information can be immediately retrieved by the user [10]. Saved messages can be accessed immediately by cell phones at any time. Interactivity is when learners can interact with peers, teachers and experts efficiently and effectively through the different media [10]. These users are required to interact with themselves or other users so that they can engage in learning. Context-awareness is when the system can adapt to the real situation to provide adequate information for the learners [10]. The proposed system is identity context aware because the system recognizes the user and each user has different information. Depending on the user, different information is accessed and provided. To better explain the characteristics Table 2 shows the comparison of these styles of learning. All the characteristics are important because the goal of the application is to make it as Ubiquitous as possible while implementing traditional learning styles.

Two examples of Ubiquitous learning environments are the "Japanese Polite Expressions Learning Assisting System (JAPELAS)" and "Tag Added Learning Objects (TANGO)". The JAPELAS system provides learners the appropriate polite expressions depending on the learner's situation and personal information. The TANGO system detects the objects around learner using RFID tags, and provides the learner the educational information [13].

Table 2. Comparison of Learning Paradigms [10]

Criteria	Ubiquitous learning	Mobile learning	Electronic learning
Concept	Learn the right thing at the right place and time in the right way.	Learn at the right place and time.	Learn at the right time.
Permanency	Learners can never lose their work.	Learners may lose their work.	Learners can lose their work.
Accessibility	System access via ubiquitous computing technologies.	System access via wireless networks.	System access via computer network.
Immediacy	Learners get information immediately.	Learners get information immediately in fixed environments with specified mobile learning devices.	Learners cannot get information immediately.
Interactivity	Learner's interaction with peers, teachers, and experts effectively through the interfaces of u-learning systems.	Learners can interact with peers, teachers, and experts in specified learning environment.	Learner's interaction is limited.
Context-awareness	The system can understand the learner's environment via database, location and situation.	The system understands the learner's situation by accessing the database.	The system cannot sense the learner's environment.

2.3 Commercial Technology Solutions for Improving Bilingual Communication

Current Technology solutions that address the multilingual communication problem include Rosseta Stone, BabelWithMe, Navita Translator, Lingtastic, and other applications that translate text. Rosetta Stone produces revolutionary software that teaches new languages by complete immersion in the language, without translation or explicit grammar explanations. With this solution, the user learns and practices a new language using interactive software on a traditional computer. The user can practice online with native speaking tutors or an online language learning community. To reinforce learning the users can listen to companion CD's [14]. Rosetta Stone is an excellent learning tool for another language. It does not do direct translations but it immerses you directly into the language using a computer. Rosetta Stone follows most of the guidelines of an Electronic learning system. The goal of the proposed solution is not to prove it is better than Rosetta Stone but instead to develop a Ubiquitous Learning System that can be advanced one day to be a viable option.

Mobile Lingtastic includes the ability to transparently translate SMS (text messages) to other cell phones. A text message can be sent and will be translated and sent to the proper recipient. Customers can use their cell phone web browser or computer based web browser to have a live interpreter on their cell phone within 60 seconds at rates starting at $17 per hour. This system also lets you create a conference call on the cell phone, land line, Google talk or Skype with a live language interpreter at a moment's notice [15]. This sounds like a very promising tool but it is international and does not create a personal one on one interaction. The main focus is not to teach but to provide

interpreters. The interpreters will forward a text message to the other recipient. The process is not automated and it is more of a service. It has some aspects of Ubiquity but it is not a learning environment.

Navita Translator is a free application specially developed for BlackBerry smartphones. It translates and speaks words and phrases from e-mails as well as from the BlackBerry Browser, SMS or typing in the software. The Navita Translator translates more than 50 languages and is able to speak languages like English, Portuguese, Spanish, Italian, French, German and Russian [16]. This tool does not create a teaching environment, it is more of a self learning tool and it is specific to Blackberry.

BabelWithMe is a simple, free online group chat tool that lets you communicate in one language or multiple languages (up to 45). It has real-time conversations without language barriers and it automatically translates each message after sending it. BabelWithMe makes it easy to invite people to join the conversation by given them a special URL address. This is a web-based tool, that doesn't need any software [17]. It is a compelling solution, but it does not meet the qualities of a ubiquitous learning system. BabelWithMe requires an online connection at all time and its main purpose is not to teach but to communicate. It is not a completely mobile learning system unless you have a smart phone with full access to internet, which excludes many people. It mainly fits the criteria of an electronic learning system.

2.4 Currently Researched Solutions for Improving Bilingual Communication

Multilingual and Mobile translation Query System presents a system that takes a message through a mobile service and the message is sent to a server for translation. The system relies on Google translate or any third party translator using the internet. The message is translated regardless of its length, and then sent back to the server. The message is delivered to the user of interest. The system provides a query service where the message itself could contain information about a product or service. The system has the capability to translate up to 15 languages including Korean and Chinese. The system performance shows that a query and translation cycle needs 10 to 15 seconds depending on how long the message's are and the efficiency of the service provider [18].

Multilingual Mobile-phone Translation Services for World Travelers is a translation service that provides text-to-text translations of. It is also a speech translation service between Japanese and English for real environments. It is based on distributed speech recognition with noise suppression. Flexible interfaces between internal and external speech translation resources ease the portability of the system to other languages and enable real-time location-free communication world-wide [19].

Both these systems are similar to the proposed system but they do not implement all of the different learning styles. Both are good products, but they only send translated messages back to the sender. They do not provide interaction with peers and friends. SMS/MMS users use texting lingo most of the time, these services do not translate improper language. Lingo is abbreviated text for words or phrases in the text messaging world. By not having this feature it does not fully integrate the user with the application.

2.5 Summary

Traditional learning consists of three different types of learning: visual, auditory and kinesthetic. Some visual learner characteristics are learning by using visual objects, reading bodily language, memorizing information and learning with lectures. Auditory learner's retain information through hearing, notice different aspects of hearing and often have talents with music. Kinesthetic learner's like to use a hands-on approach to learn and would rather demonstrate how to do something rather than verbally explain it. Ubiquitous learning is integrating technology in everyday life to support learning. Ubiquitous learning is composed of Mobile and Electronic learning. A proposed definition of Ubiquitous learning is a learning paradigm which takes place in a ubiquitous computing environment that enables learning the right thing at the right place and time in the right way. Five different learning characteristics of Ubiquitous learning are: permanency, accessibility, immediacy, interactivity and context-awareness. BabelWithMe is the closest commercial competitor to Motilingual. BabelWithMe was found to be a good tool for communication in the same way as the Ubiquitous learning tool proposed by this research.

CHAPTER 3

Methodology and System Details

3.1 Overview of Proposed Solution

Motilingual has been designed to help students learn another language by seeing, hearing and experiencing communication with a native speaker of that language. The system has several components which have been selected to facilitate learning while performing the daily routine of text messaging. The system involves the use of emails, Google translate, Espeak, speech synthesis and a custom lingo-to-text conversion process. Each component plays an important role in order to maximize learning potential. An environment has also been created to provide user friendliness and to give the user options and control.

3.2 Block Diagram

The proposed solution needs to follow the guidelines of Traditional and Ubiquitous Learning. The goal is to fuse technology and learning to communicate as one. The three different major styles of traditional learning were considered. The question is "what kind of system can be designed to ensure the user would learn with technology at the same time?" The answer is "a mobile-to-mobile translation system that converts a message from one language to another without interrupting a daily routine". This system will allow people to communicate and learn a different language using standard

SMS/MMS technology. To better explain how all the components of the proposed Motilingual system are connected, Figure 1 shows a Block Diagram of the system.

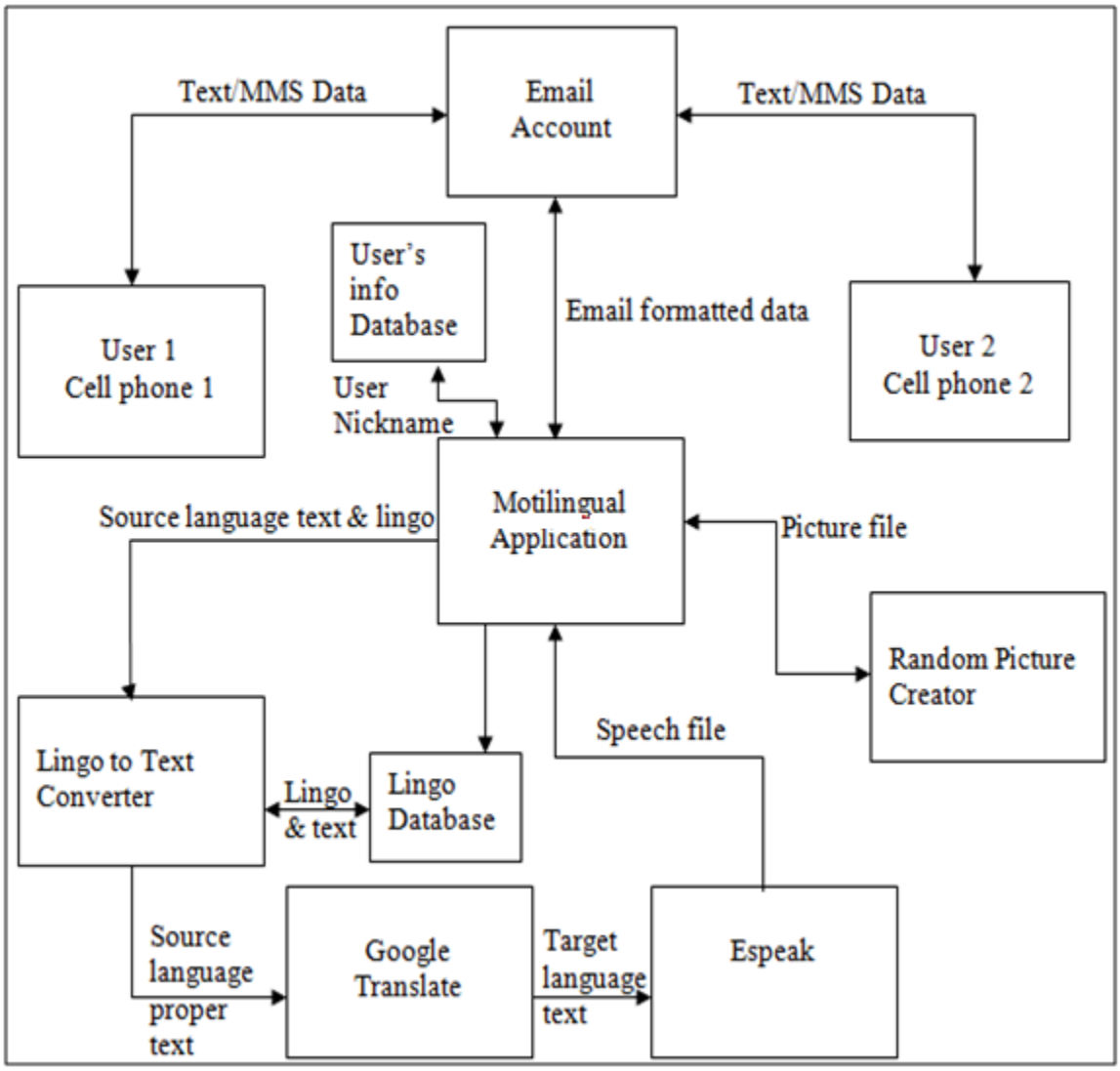

Figure 1. Block Diagram of the Motilingual System

In order to receive the messages, the Motlingual application checks an email account periodically. The application pops an email account and extracts one email at a time. It determines the sender, the subject, the recipient and the text message that needs to be sent. Once the message received is extracted, it checks a database and replaces any texting lingo with proper syntax in the target language. A database is used to convert the lingo to the appropriate word. This must be done since Google translate does not recognize improper words. Google translates application programming interface (API) is used to translate the message. The API returns a translated message back to the computer application. Once a proper sentence is available, it is sent to a text to speech synthesizer program called Espeak. This is a command line executable program that can be used to create a wave file from text input. This is needed since a audio message is to be sent to the other user. Once all these operations are performed, the non-lingo message plus the translation will be sent along with the synthesized speech to both the sender and recipient. These results are sent back to the sender so that the sender can see what message translation was and to help the sender learn both when sending and receiving messages.

3.3 Description of Each Component

The flow of the system is explained in the section above. To recap, two users are using their mobile devices to communicate. The environment encompasses all the components except for the cell phones. To use the Motilingual system, users need MMS based cell phones and only have to understand the syntax expected by the application. An

effort was made to make the system as user friendly as possible and to resemble everyday texting.

3.3.1 Email Account

The email account has only one purpose and that is to send and receive messages, making it one of the key components of the system. It receives and sends messages from and to the users. This component allows the user to send text messages. The short messaging service of the cell phone allows you to send text messages to, as well as receive messages from, email accounts. The email account is the main connection between the user and the application.

3.3.2 Google Translate

Google Translate purpose in the Motilingual system is to translate messages from English to Spanish and vice versa. This component is key when providing a good translation of a message for the user. Like all automated translators, this is not the best means of translation or for learning. When Google Translate generates a translation, it looks for patterns in hundreds of millions of documents to help decide on the best translation for you. By detecting patterns in documents that have already been translated by human translators, Google Translate can make intelligent guesses as to what an appropriate translation should be. This process of seeking patterns in large amounts of text is called "statistical machine translation". Since the translations are generated by machines, not all translation will be perfect. The more human-translated documents that Google Translate can analyze in a specific language, the better the translation quality will be. This is why translation accuracy will sometimes vary across languages [7].

Motilingual supports visual learning because the users will be able to see the messages in its appropriate translation.

3.3.3 Motilingual Lingo-to-Text Converter

Google translate only works with proper English and Spanish. Anything else is not recognized. Most people that text use texting lingo. Texting lingo is used for abbreviations of words to speed up text messaging. For example, “lol” stands for “laughing out loud” and “b4” stands for “before”. A special lingo-to-text converter was designed so that Google translate would be able to translate messages more accurately. It was also implemented to ensure the users does not change their way of texting. The lingo converter has access to a database that holds the lingo abbreviations and words. This data never gets lost. It keeps track of all the words that the user input. When the user registers to the application, he/she gets added to a list and two standard databases get created. One contains Standard English lingo and the other is Standard Spanish lingo. The purpose of the databases is to allow the users to customize them, since each person may use different lingo.

3.3.4 Text-to-Speech Synthesizer

Espeak is a compact open source software speech synthesizer for English and other languages. Espeak uses a "formant synthesis" method. This allows many languages to be provided in a small size. The speech is clear, and can be used at high speeds, but is not as natural or smooth as larger synthesizers which are based on human speech recordings [8]. Adding a speech file was necessary to support audio learning for the user. Every message gets an attached audio file to promote learning not only by seeing the

translation but also by hearing. Espeak was chosen because it can convert text to speech and is open source for multiple languages. It can also be interacted with by Microsoft Visual Studio using command line programming. So it was simple to adapt to the system.

3.3.5 Random Picture

An extra component was added to increase visual learning. This component sends a random picture to the user with an image as well as the text. For example, if a picture of an apple was sent, then the words "apple" and "manzana" would be sent along with it. The reason for this is to help the user remember. The program only sends the picture messages between 9am and 8 pm and at random times. This is to be respectful and not invade privacy.

3.4 Computer Application Syntax

Motilingual is designed to support user friendliness. The application has certain syntax in order for the user to get access to the program and the features. The user has to send text messages to the email account in order to register and interact with the program. The email account that was created is "motilingual@traid.rr.com". Table 3 explains what options the user has and how to manage and control the program. From the table you can see that the number 304 is used as part of the syntax, this is because it represents a Spanish Phonetics class that is taught at this University. A message syntax feature mentioned in the table, is that when sending a message in Spanish and tildes are needed, the user can capitalize the letters or change the phone settings to Spanish. If this is not done, issues can arrive when translating and they will not be as accurate**.**

Table 3. Motilingual Functionalities and Syntax

Instruction	Purpose
Reg304 your name : 3 letter nickname : provider Ex: reg304 daniel valencia : dva : verizon	This needs to be the first step in order for a user to get access. This is used to register. You need to register your name, a 3 letter nickname and your provider.
Hlp304 Ex: Hlp304	This is a request for help. It sends the user a list that provides the user with the syntax of the program.
Ses304 recipient nickname : message Ex: Ses304 dva: Hola como estas? Ses304 mss This sends mass email to all users.	This stands for send message in Spanish. You need to input the message in Spanish. The message gets translated and sends it to the recipient and the sender with the audio.
Sen304 recipient nickname : message Ex: Sen304 dva: Hello how are you? Sen304 mss This sends mass email to all users.	This stands for send message in English. You need to input the message in English. The message gets translated and sends it to the recipient and the sender with the audio.
Ltu304 Ex: Ltu304	This is a request to list the users. The person receives all the user names and nicknames.
Dtl304 lingo abbreviation Ex: Dtl304 lol	Delete lingo. This deletes the lingo the user inputs.
Adl304 lingo abbreviation : definition Ex: Adl304 lol: laughing out loud	Add lingo. This adds new lingo to each user personal file. The data is available for them at any time and it doesn't get erased.
Brl304 lingo abbreviation Ex: Brl304 amr	This is Spanish for delete lingo. It is the Spanish version of dtl304.
Agl304 lingo abbreviation : definition Ex: amr : amor	This is Spanish for add lingo. It is the Spanish version of adl304.
Lle304 Ex: Lle304	List lingo en English. It sends the user the list of the entire lingo.
Lls304 Ex: Lls304	List lingo in Spanish. It sends the user the list of the entire lingo.
Use capital vowels for Tildes Ex: DIgame, cafE, adOnde This is equivalent to Dígame, café and Adónde	Phones that do not have Spanish settings do not have tildes. This is used to replace the tildes. If not used, translation will get affected.

3.5 Summary

Motilingual has been designed to help students learn another language by seeing, hearing and experiencing communication with a native speaker of that language. In this Chapter Motilingual's major components were described. Some of the systems major components are: email account, Google Translate, lingo-to-text converter, text-to-speech synthesizer and random picture. All the components were described along with the reason why they were selected. Table 3 was also provided Motilingual's functionalities and system syntax.

CHAPTER 4

Results and Analysis

4.1 Motilingual Usage Description

Motilingual was designed for ease of operation by the users. The ease of use is to help the user adapt to the new system without many issues. Chapter 3 has Motilingual's functionality Table 3 which can assist them on how to utilize the system. Motilingual will be expecting this syntax; if it is not used in the appropriate manner software has been implemented to let the user know what type of syntax error occurred. The interaction between a user and Motilingual will be discussed in the next sections. Section 4.1.1 describes registering, asking for help, listing users, listing English and Spanish lingo files and adding and deleting lingo words from the appropriate file. Section 4.1.2 goes over the communication feature which allows users to send a message in Spanish or English. Lastly, section 4.1.3 expands on the extra features that Motilingual has, these are the random picture sender and the dictionary definition of a word instead of a sentence.

4.1.1 User Registration and Query

The first step is to register. To do this, the users must send the command "reg304 Daniel : dva : Verizon" and they receive a registration confirmation message. Figure 2 shows the response the user receives.

Figure 2. Users Registration Response

The next step can be to ask the program for help. To do so the user must then enter the help command “hlp304” this sends them a help confirmation message. Figure 3 shows the response the user receives.

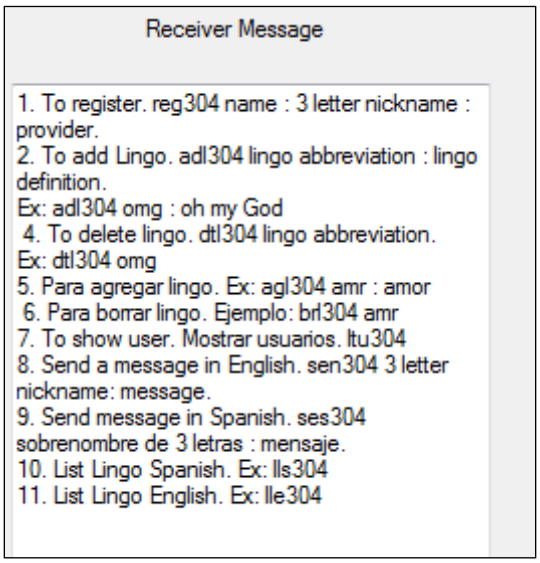

Figure 3. Users Help Query Response

The help message allows the user to understand how to use the system. It is very basic and self explanatory. Once a user is registered and understands how to use the system he/she has the ability to use the other functions. If a user wants to find out if a friend is registered he/she can ask for the user names and the nicknames by using the "ltu304" command. Figure 4 shows what the user will receive in response.

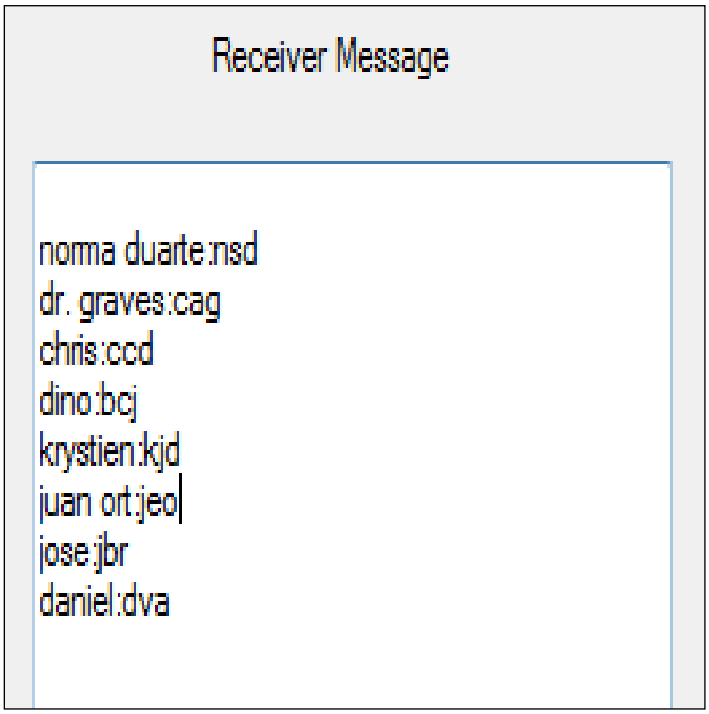

Figure 4. User Names/Nicknames Query Response

Since one of the features of the system is to allow texting lingo the user can request what lingo is available for them both in English and Spanish. Figure 5 shows the English Lingo to the user. The "lle304" command was used.

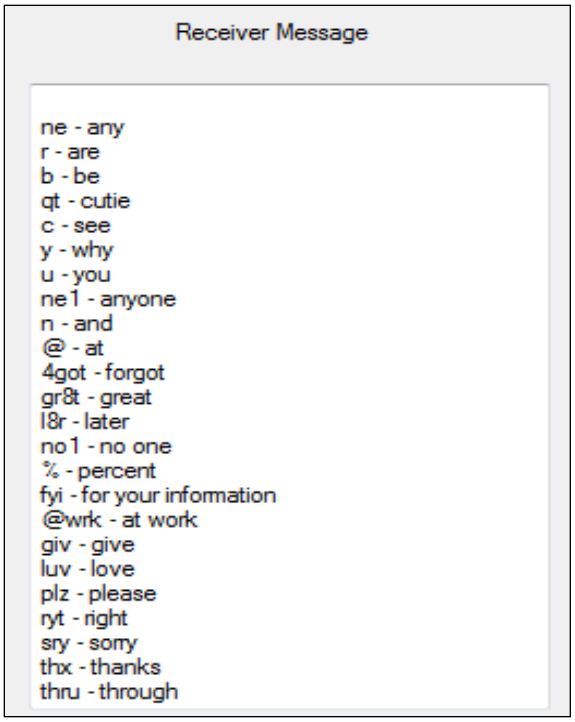

Figure 5. English Lingo Query Response

This lingo database can be modified by the user. For example, the user can add and delete lingo if he/she wants to. Figure 6 shows what the user receives if he/she decided to delete a word by using the "dtl304 lol" command.

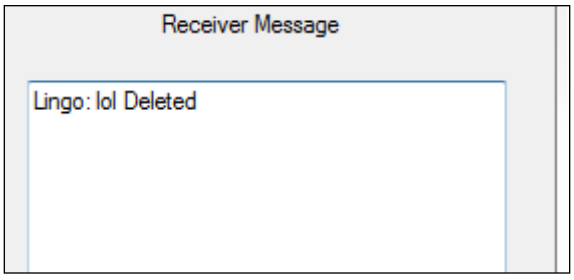

Figure 6. Lingo Delete Query Response

If the person decides to add lingo by using the “adl304 lol: laughing out loud” command a confirmation receipt will be sent to the user, Figure 7 shows what the user receives. The same happens when the user wants to delete or add lingo from the Spanish text file. He/she will use “Brl304 amr” and “Agl304 amr:amor” commands to delete and add lingo respectively. The user will receive similar messages like Figure 6 and Figure 7.

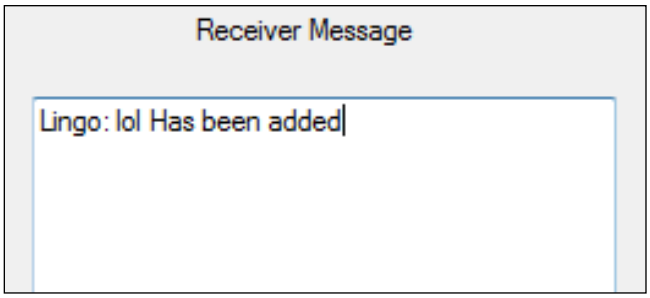

Figure 7. Lingo Add Query Response

4.1.2 Communication Feature

The primary feature of the system is to allow people to communicate even though they do not know each others language. An english speaking person can still send a message and a Spanish speaker will understand it. If one user wants to send a message in Spanish then he/she will send the command “Ses304 dva: hola como estas? Que vas hacer hoy?”. Then a translated message gets sent to both the sender and the receiver. The purpose is so that they both know what the translation is and they can learn more. Figure 8 shows one user greeting another user in a common manner.

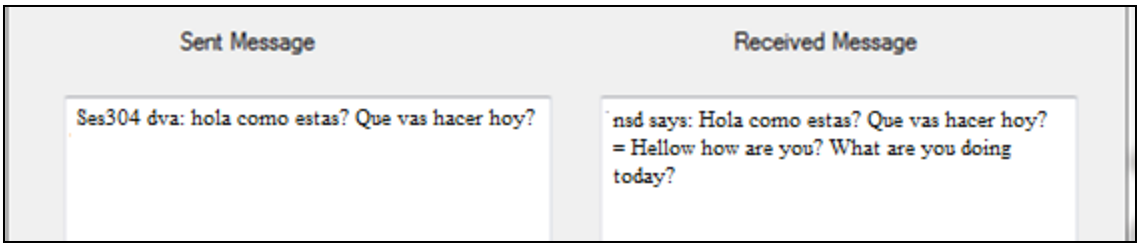

Figure 8. User 1 Asks Question

Figure 9 shows the second user replying the first user by telling him he is perfect and asking if he wants to go to the movies. He/She used the command "Sen304 : Everything is going perfect. Do you want to go to the movies?".

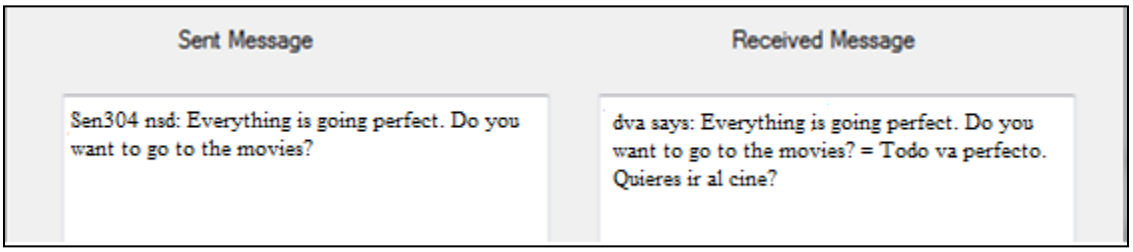

Figure 9. User 2 Replies

This conversation can go on until any user finishes the conversation. To initiate the conversation the user does not need to re-register or start the process over. He can just send a normal Motilingual formatted message to any user that has been previously registered.

One other feature that needs to be discussed is the Lingo to appropriate language conversion. For example, if a sentence is written in the following manner, "Gm. I mu. I wanted to say thx for everything u have done for me. I am sry but I g2g." This is

confusing but the system actually converts that to “Good morning. I miss you. I wanted to say thanks for everything you have done for me. I am sorry but I got to go.” This feature is unique, some online websites do offer the lingo translation but not in the context or an SMS texting system.

4.1.3 Extra Features

Another feature the system has is that it sends random picture messages so the user can learn by relating a picture to text. Figure 10 shows what random picture the user will receive and at a random time in the day.

Figure 10. Random Picture Message

The user can also message themselves either a sentence or a word, maybe they need on demand translation or they have no one to message. If the user sends the command “Sen304 dva: run”, Figure 11 shows what the user will receive.

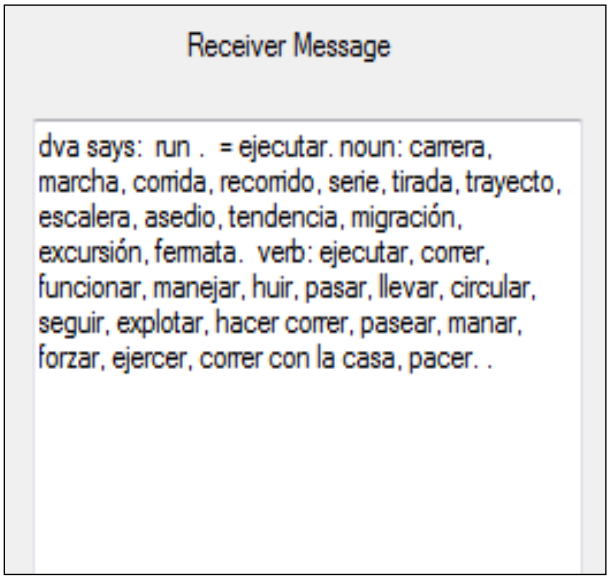

Figure 11. Run Dictionary Definition

4.2 Qualitative Comparison to BabelWithMe

BabelWithMe was previously described and is comparable to Motilingual. BabelWithMe is a cross-language chat room that lets multiple people communicate in different languages at one time. The users have to be logged on to a internet connected PC to get access. Then they can have group conversations with the people they want. This is a really good feature because multiple people can engage in one conversation, no limit is specified. BabelWithMe allows cross-language chatting 45 different languages from all over the world. The conversation initialization is simple. One user has to start the conversation and send a URL link to all the other users. Once they are in the conversation BabelWithMe runs smoothly and quickly. On the other hand, Motilingual allows multiple people to communicate in only two different languages at on time. The users can be

anywhere in the country as long there is a mobile service provider. Only two people can engage in a conversation but multiple separate conversations can occur. A conversation initialization is simple and the only requirement is a one-time registration. Once the user is registered a normal text message can be sent.

In comparison to Motilingual, BabelWithMe has some qualitative and quantitative disadvantages. Motilingual is meant to be used for cell phones, so that the user can chat with friends on the run. The main reason is for students to learn the two most common spoken languages in the United States. To allow users to learn, Motilingual has the random picture message feature which sends an image with text to the user. Every time a message is sent the end user receives an audio file to improve learning by hearing. Since Lingo is a big part of texting, Motilingual has an automatic lingo converter in both languages so the user does not have to change the way he/she texts. Table 4 shows a feature comparison of the two. Up and down arrows will be used to specify which system is better. When it comes to translation from English to Spanish it is the same because they both use Google translate as a translation tool. Motilingual allows lingo to be used when sending messages making it better. Under the multiple users category BabelWithMe is superior because it allows not only multiple user but group chats, Motlingual allows multiple users but not group chats. Motilingual has four features that make it unique compared to BabelWithMe which are text-to-speech, picture with text and it is open source. Motilingual also incorporates different learning styles which are very important, BabelWithMe main purpose is communication.

Table 4. Standard Feature Comparison to BabelWithMe

Standard Features	System Name	
	Motilingual	**BabelWithMe**
Translate English and Spanish	Same	Same
Translate English and Spanish lingo	↑	N/A
Multiple users	↓	↑
Text to speech	↑	N/A
Picture with text	↑	N/A
Incorporates different styles of learning	↑	↓
Open source	↑	↓
Word definitions	↑	N/A
Privacy	↑	↓

Now that the system was compared by the features alone, a comparison is done by deciding which system follows the different styles of learning guidelines. Table 5 shows a comparison of the two. The up arrow shows which system is better. Motilingual surpasses BabelWithMe in almost all the Ubiquitous learning categories as well as the traditional learning categories. In areas like accessibility, permanency, immediacy and context awareness, Motilingual has an advantage. Motilingual is more accessible because the users can access the system anywhere mostly because it is a mobile system and BabelWithMe is not. It is mobile since users will access Motilingual by using their cell phones and BabelWithMe are limited to locations that have wireless internet. The users have immediate access to the application files at any time and BabelWithMe has no access to any files. When it comes to permanency, Motilingual stores databases on a computer, this causes to permanently keep a users file unless they delete the data. BabelWithME does not have the characteristics of permanency because it does not keep

track of any files. Motilingual is identity context aware because it checks for both the sender and receiver of a message to ensure they are members. It also checks their personal lingo file since each member has the potential of having different lingo and provides them with accurate information according to their need. BabelWithMe is not context aware because different information is not provided by identifying each user. BabelWithMe wins in the interactivity class since it allows group chatting. Motilingual only allows two users to chat at a time and it does not allow group chatting.

When comparing traditional learning characteristics Motilingual exceeds BabelWithMe in the visual and auditory characteristics. For visual and auditory learning, Motilingual not only has text messages but it also sends pictures and voices messages. BabelWithMe only shows message translation to the other users but does not include audio and pictures. When it comes to hands on learning they both make the users interact with friends and family by having them message each other. They both translate the messages allowing people with different languages to communicate with each other.

Table 5. Learning Paradigms Comparison to BabelWithMe

Traditional and Ubiquitous Learning Features	**System Name**	
	Motilingual	**BabelWithMe**
Accessibility	↑	↓
Permanency	↑	N/A
Immediacy	↑	N/A
Interactivity	↓	↑
Context aware	↑	N/A
Visual learning	↑	N/A
Auditory learning	↑	N/A
Kinesthetic learning	Same	Same

4.3 Quantitative Comparison to BabelWithMe

To do a quantitative comparison between BabelWithMe and Motilingual an experiment was set up. This experiment reassured both the speed and accuracy of each system when translating messages. Forty messages were sent to Motilingual to check the accuracy and speed. Twenty of the messages were in English and Spanish without lingo and the other twenty messages were with lingo to check accuracy. The same twenty messages in English and Spanish were sent to BabelWithMe without lingo to check for speed. Messages with Lingo were not tested using BabelWithMe because the accuracy would have been zero percent, since this system does not understand lingo. Table 6, 7, 8 and 9 show all the test data for Motilingual.

The first column has the message that was sent by the user. Table 7 and Table 8 have two messages. The first one contains lingo and the second doesn't but the message was not sent twice. In the translation column there are two values. "Actual" represents the appropriate translation by a Spanish speaking person and the other translation is by Google Translate's. The tables have a time column which is measured in seconds. This value represents the total time it took a message to go from one user's cell phone to the other user's cell phone. In between, the message starts at a dedicated email account, then any lingo the message contains is appropriately converted, then the message gets translated using Google and finally the message is sent to the recipient using the dedicated email account with the audio file. Of course, this is assuming both the sender and the receiver are registered user's and he/she used the appropriate syntax for the message.

Table 6. No Lingo Motilingual English Translation Measurements

Message	Translation	Correct?	Time(sec)
Hello, what are you doing today?	Actual: Hola, que vas hacer hoy? Google: Hola, ¿qué vas a hacer hoy?	Yes	28
Do you understand English?	Actual: Entiendes Ingles? Google: Entiende usted Inglés?	Yes	46
Do you want to go to the movies tonight?	Actual: Quieres ir a las peliculas en la noche? Google: Quíeres ír al cíne esta noche?	Yes	50
Today is a very special day.	Actual: Hoy es un día especial. Google: Hoy es un día muy especial.	Yes	30
Let's go to the pool today!	Actual: Nos vamos a la piscina hoy! Google: Vamos a ir a la piscina hoy!	Yes	22
Did you watch the show last night?	Actual: Viste el show anoche? Google: ¿Viste el show de anoche?	Yes	24
I love United States because of freedom.	Actual: Me encanta los Estados Unidos por la libertad. Google: Me encanta Estados Unidos a causa de la libertad.	Yes	40
Where can I find the bathroom?	Actual: Donde puedo encontrar el baño? Google: ¿Dónde puedo encontrar el baño? .	Yes	40
Hello. My name is Daniel and I love to play soccer.	Actual: Hola mi nombre is Daniel y me encanta jugar el futbol. Google: Hola. Mi nombre es Daniel y me encanta el fútbol.	Yes	20
Can you please help me out with the following problem?	Actual: Me puedes ayudar con el siguiente problema? Google: Puede usted por favor me ayude con el siguiente problema?	No	34

Table 7. English with Lingo Translations Measurements

Message	Translation	Correct?	Time(sec)
-I luv n8r. -I love nature.	Actual: Me encanta la naturleza. Google: I aman la naturaleza.	No	26
-Btw, you are ringl8. -By the way, you are running late.	Actual: Por cierto, vienes tarde. Google: Por cierto, se le hace tarde.	No	40
-How is your bf doing? -How is your boy friend doing?	Actual: Como esta tu novio? Google: Cómo es novio haciendo?	No	22
-Have a gn mother! -Have a good night mother	Actual: Que tengas una Buena noche madre. Google: Tener una madre buena noche.	No	12
-Can u plz bring me the Shampoo? -Can you please bring me the Shampoo?	Actual: Puedes traerme el Champu por favor. Google: Puede usted por favor tráeme el champú.	Yes	12
-U really upset me tonight. I will just ttyl. -You really upset me tonight I will just talk to you later.	Actual: Realmente me enojaste esta noche. Hablaremos mas tarde. Google: Que realmente me molesta de esta noche. Sólo voy a hablar con usted más tarde.	No	36
-U are hilarious, you make me rofl. -You are hilarious, you make me rolling on floor laughing.	Actual: Eres muy comico, me haces road en el suelo riendo. Google: Eres gracioso, me haces rodar en el suelo riendo.	Yes	37
-It was a pleasure meeting you, plz kit. -It was a pleasure meeting you, please keep in touch.	Actual: Fue un placer conocerte, mantengase en contacto. Google: Se trataba de una reunión que por favor, por favor, manténgase en contacto.	No	60
-I know you're a busy so tyt. -I know you are busy so take your time .	Actual: Se que estas ocupado entonces toma tu tiempo. Google: Sé que están ocupados así que tómate tu tiempo.	Yes	32
-Fyi my thesis is due Monday so plz edit my paper asap. -For your information, my thesis is due Monday so please edit my paper as soon as possible.	Actual: Para tu information, tengo que entregar mi tesis el Lunes entonces por favor edita me papel lo mas pronto possible. Google: Para tu información, mi tesis se debe Lunes así que por favor editar mi trabajo tan pronto como sea posible.	Yes	39

Table 8. No Lingo Motilingual Spanish Translation Measurements

Message	Translation	Correct?	Time(sec)
¿Cómo está usted?	Actual: How are you? Google: How are you?	Yes	21
Sí, hablo un poquito de español.	Actual: yes, I speak a little Spanish. Google: Yes, I speak a little Spanish.	Yes	30
Yo espero que él me compre unas flores.	Actual: I hope he buy me flowers. Google: I hope he buys me flowers.	Yes	37
Me alegro de que tú sonrías.	Actual: I am happy you smile. Google: I'm glad you smile.	Yes	39
¿Conoce un buen restaurante?	Actual: Do you know a good restaurant? Google: Know a good restaurant?	Yes	20
La chicas están en la clase.	Actual: The girls are in class. Google: The girls are in class.	Yes	24
Está justo detrás de la parada de metro Tacubaya.	Actual: Is right behind the metro stop Tacubaya. Google: Is right behind the metro Tacubaya.	Yes	37
La casa está ubicada a tres cuadras del centro en un callejón tranquilo.	Actual: The house is located three blocks from downtown in a quiet street. Google: The house is located three blocks from downtown on a quiet side street.	Yes	38
Estos dormitorios están bien diseñados.	Actual: These dorms are well designed. Google: These rooms are well designed.	Yes	40
La señora, cuyo hijo está en mi clase, trabaja en la oficina de mi padre.	Actual: The ladies son who is in my class, works in my father's office. Google: The lady whose son is in my class works in the office of my father.	Yes	39

Table 9. Spanish with Lingo Translations Measurements

Message	Translation	Correct?	Time(sec)
-Se que estas muy ocupado, m1ml. - Se que estas muy ocupado, mandamame un mensaje luego.	Actual: I know your busy, send me a message later. Google: I know you're busy, send me a message then.	Yes	53
-Tengo a mi mama en la otra linea nph. - Tengo a mi mama en la otra linea no puedo hablar	Actual: I have my mom on the other line I can't speak. Google: I have my mom on the other line I can not speak.	Yes	22
-Cuando empiezas la uni? - cuando empiezas la universidad?	Actual: When do you start college? Google: When you start college?	Yes	27
-Tranquilo que npn. -Tranquilo que no pasa nada.	Actual: Don't worry, nothing will happen. Google: Assured that nothing happens.	Yes	29
-Gracias a Dios llego el finde. -Gracias a Dios llego el fin de semana.	Actual: Thank god the weekend is here. Google: Thank God the weekend arrived.	Yes	19
-Esta tesis es mim pero no pierdo mis esperanzas. -Esta tesis es mission imposible pero no pierdo mis esperanzas.	Actual: This thesis is mission impossible but I do not lose my hopes. Google: This thesis is mission impossible but I do not lose my hopes.	Yes	42
-Llego mi padre xfa ymam que tqi. -Llego mi padre por favor llamame que tengo que irme.	Actual: My father arrived please call me I have to go. Google: I get my father please call me I have to go.	No	25
-Q tal, como estas? - Qué tal, como estas?	Actual: Hello, how are you? Google: Hello, how are you?	Yes	49
-Perdi tu llamada. Q qrs? - Perdi tu llamada. Que quieres? -	Actual: I lost your call. What do you want? Google: I lost your call. What do you want?	Yes	38
-Porque tienes la kbza grande? Que has exo? -Porque tienes la cabeza grande? Que has hecho?	Actual: Why do you your head big? What have you done? Google: Why have the big head? What have you done?	No	33

Tables 10 and Table 11 show the test data for time when using BabelWithMe. Only the previous messages without lingo were sent because BabelWithMe does not translate lingo. BabelWithMe does not have a time value for messages arriving at the application. Instead the only time that was measured was the time it took for the recipient to receive the translated message.

Table 10. BabelWithMe English Translation Time Measurements

Message	Time(sec)
Hello, what are you doing today?	1
Do you understand English?	1
Do you want to go to the movies tonight?	2
Today is a very special day.	2
Let's go to the pool today!	3
Did you watch the show last night?	8
I love United States because of freedom.	10
Where can I find the bathroom?	4
Hello. My name is Daniel and I love to play soccer.	3
Can you please help me out with the following problem?	2

Table 11. BabelWithMe Spanish Translation Time Measurements

Message	Time(sec)
¿Como está usted?	2
Sí, hablo un poquito de español.	17
Yo espero que él me compre unas flores.	2
Me alegro de que tú sonrías.	6
¿Conoce un buen restaurante?	1
La chicas están en la clase.	8
Está justo detrás de la parada de metro Tacubaya.	3
La casa está ubicada a tres cuadras del centro en un callejón tranquilo.	1
Estos dormitorios están bien diseñados.	6
La señora, cuyo hijo está en mi clase, trabaja en la oficina de mi padre.	4

Table 12 shows the comparison between both systems. By analyzing the data BabelWithMe is quite a bit faster than Motilingual. This is attributed to the fact that the application is checking a dedicated email account every twenty seconds and it sends an MMS message that contains an audio file. The audio file is an attachment and this causes to slow down sending the email message. They both have the same accuracy in translation because they are both using Google Translate. Motilingual has about 60 percent accuracy for translation with lingo. Since lingo phrases are usually figures of speech, this can cause a difficulty in translation.

To register with Motilingual, one simply needs to send a one-time message with the users name, nickname and provider. To register with BabelWithMe, three steps are involved but you have to multiply by n and this represents the number of people in the conversation. Once registered in Motilingual, the user can initialize the conversation at any time. Once registered to BabelWithMe a link has to be sent to all the users they want to invite to their conversation every time. This is why the value gets multiplied by n. Both systems had limitation on how many characters they can translate. Motilingual had a capacity of 1000 characters with spaces. BabelWithMe crashed with a message over 650 characters with spaces. One of the best benefits of BabelWithMe is that it had zero characters in the preamble of the message. In Motilingual the user needs to input nine characters to send the message which can be a little time consuming, depending on the skills and device features of the user.

Table 12. Quantitative Comparison with BabelWithMe

	Motilingual	**BabelWithMe**
Average send/receive time	32.8	4.3
Translation Accuracy without lingo	95%	95%
Translation accuracy with lingo	60%	N/A
Overall translation accuracy	75%	N/A
Number of steps for user registration	1	3*n
Number of steps for conversation initiation	0	1*n
Character limitation with spaces	1000	650
Length of message preamble	9	0

CHAPTER 5

Conclusions and Future Work

5.1 Conclusion

The research work that was performed led to the creation of Motilingual. This is a cross-language mobile communication and pervasive language learning tool that uses multimedia cellular phone messaging and an online translation system. The proposed system offers an learning environment that may help students learn Spanish and English. It does this by allowing two users of two different native languages to communicate using cell phones SMS and MMS capabilities. The system has useful features that can help users learn while maintaining standards of texting. Motilingual exemplifies Ubiquitous computing by integrating the system as part of everyday activities. It is a mobile system that can be accessed by anybody who has a cell phone. This means over 290 million cell phone subscribers that can take advantage of this system. Users have immediate access to their personal files and can have fun while learning.

Motilingual was compared to an online cross-language BabelWithMe both qualitatively and quantitatively. From looking at the standard feature, Motilingual was better because it had several feature like lingo conversion, text to speech, picture with text messages and word definitions which exhibit different styles of learning. When analyzing the different learning paradigms, Motilingual had all of the Ubiquitous learning characteristics incorporated. Some of these include accessibility, permanency, immediacy, interactivity, context-awareness, visual learning, auditory learning and

kinesthetic learning. BabelWithMe only incorporated accessibility, interactivity and kinesthetic learning.

Some quantitative testing was done to compare Motilingual to BabelWithMe. BabelWithMe is quite a bit faster when looking at the average send/translate/receive time of a message. From this testing it was reassured that both systems use Google Translate as a translation tool. Motilingual had fewer registration and conversation initialization steps. Motilingual also had a higher capacity for translating a message.

5.2 Future Work

An advantage of this system is that it has the ability to grow. Many features can be added to enhance learning and interaction with users and to speed up the process. The system can be modified as follows:

- Eliminating the email account so it can speed up translations and sending messages out.
- Allowing a user to send a message using voice and translate that message and send it to the other user.
- Creating smart applications for the different platforms available on the market.
- Sending more random messages versus only picture messages, for example voice messages,
- Allowing the user to upload pictures to add to the profile.
- Handling many more languages by using Google Translate
- Making it a into a web application.

- Incorporating instructor tracking for grading.
- Shortening the preamble character length.

BIBLIOGRAPHY

[1] US Census Bureau. (2011) US Census Bureau. [Online]. http://www.census.gov/prod/2011pubs/11statab/pop.pdf

[2] US Census Bureau. (2006) US Census Bureau. [Online]. http://www.census.gov/population/www/socdemo/hispanic/files/Internet_Hispanic_in_US_2006.pdf

[3] CTIA The Wireless Association. (2010, Mar.) CTIA The Wireless Association. [Online]. http://www.ctia.org/media/press/body.cfm/prid/1936

[4] Nielson Company. (2010) Nielson Wire. [Online]. http://blog.nielsen.com/nielsenwire/online_mobile/us-smartphone-battle-heats-up/

[5] Modern Language Association. (2010, Dec.) Modern Language Association. [Online]. http://www.mla.org/pdf/2009_enrollment_survey_pr.pdf

[6] CTIA The Wireless Association. (2010, June) CTIA The Wireless Association. [Online]. http://files.ctia.org/pdf/CTIA__Survey_Midyear_2010_Graphics.pdf

[7] Google. Google. [Online]. http://translate.google.com/about/intl/en_ALL/

[8] Source Forge. (2007) Sourceforge. [Online]. http://espeak.sourceforge.net/

[9] University of South Dakota. (2009) Three Different Learning Styles. [Online]. http://people.usd.edu/~bwjames/tut/learning-style/styleres.html

[10] Saadiah Yahja, Erny Arniza Ahmad, and Kamarularifin Abd Jalil, "The Definition and Characteristics of Ubiquitous Learning: A discussion," *International Journal of Education and Development using Information and Communication Technology*, vol. 6, 2010.

[11] Judy York and Parag C. Pendharkar, "Human–computer interaction issues for mobile computing in a variable work context ," *International Journal of Human-Computer Studies*, vol. 60, no. 5-6, pp. 771-797, May 2004.

[12] D Dochev and I Hristov, "Mobile Learning Applications: Ubiquitous Characteristics and Technological Solutions," *Cybernetics and Computer Technologies*, vol. 6, no. 3, pp. 63-74, 2006.

[13] Hiroaki Ogata and Yoneo Yano, "Context-Aware Support for Computer-Supported Ubiquitous," in *Proceedings of the 2nd IEEE International Workshop on Wireless and Mobile*, 2004, pp. 27-34.

[14] Rosetta Stone. (1999) About Us: Rosetta Stone. [Online]. http://www.rosettastone.com/personal/faq

[15] Lingtastic. (2008, June) Free Press Release. [Online]. http://www.free-press-release.com/news/200806/1214578824.html

[16] Navita BB Translator. About Us: Navita BB Translator. [Online]. http://www.bbtranslator.com/blackberry-translator/home.html

[17] Life Church Tv. (2007) About Us: Babel With Me. [Online]. http://www.babelwith.me/about

[18] Moussa Abdallah, Ata Alqadi, Ahmad Saleh, and Waleed Khamis, "Multingual Mobile Translation and Query System," in *Proceedings of the 8th International Conference on Applications of Electrical Engineering*, Houston, 2009, pp. 111-116.

[19] Michael Paul et al., "Multilingual Mobile-Phone Translation Services for World Travelers," *International Conference on on Computation Linguistics:*, pp. 165-168, 2008.

APPENDIX

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Net.Mail;
using System.Net.Security;
using System.Net.Sockets;
using System.Net;
using System.Text.RegularExpressions;
using System.Speech.Synthesis;
using System.Media;
using System.Diagnostics;

namespace FinalThesisProject
{
   public partial class Form1 : Form
   {
      //Initilizing global variables
      const int ARRAY_LENGTH = 20000;
      static string[] array = new string[ARRAY_LENGTH];
      static int messagecount = 0, counter = 0;
      static string phoneNum = "";
      static int tempphoneNum = 0;
      static string tempSender = "";

      //Intialize GUI
      public Form1()
      {
         InitializeComponent();
      }//end Form initialization

      //Initialize start button which also initializes two timers
      private void startButton_Click(object sender, EventArgs e)
      {
         popTimer.Enabled = true;
         randomTimer.Enabled = true;
      }//End start Button function
```

```
      //Pop timer function
      private void popTimer_Tick(object sender, EventArgs e)
      {
         //initializing local variables
         counter = 0;
         messagecount = 0;
         int fromwho = 0, fromline = 0;
         string from, tester = "";
         string ListMessage, MessageBody, DialogueText = "";
         int bodyline = 0;
         string bodytest = "";
         string matchingString = "";
         string matchStringLower = "";
         string errorMessage = "";
         // Create a TCP client for a TCP connection
         TcpClient tcpClient = new TcpClient();
         //Get server and authentication information.
         string PopServer = "pop-server.triad.rr.com";
         string PortNumber = "110";
         string UserName = "dvalenci85@triad.rr.com";
         string PassWord = "dv4507888";
         //Begin talking to the POP3 Server using the Post Office
Protocol (POP)
         DialogueText = "I say:\r\nConnect me to " + PopServer +
":" + PortNumber + "\r\n\r\n";
         // Connect this TCP client to the server IP/name and
port specified in the form
         tcpClient.Connect(PopServer,
Convert.ToInt32(PortNumber));
         // Create a network stream to retrieve data from the TCP
client
         // Create a non-secure network stream to retrieve data
from the TCP client
         NetworkStream netStream = tcpClient.GetStream();
         //netStream.AuthenticateAsClient(PopServer);
         // We need a stream reader to be able to read the
network stream
         System.IO.StreamReader strReader = new
System.IO.StreamReader(netStream);
         // If the connection was made successfully
         if (tcpClient.Connected)
         {
            Console.WriteLine("Connected, now attempting to
authenticate!\n\n\n");
            DialogueText += "Server says:\r\n" +
strReader.ReadLine() + "\r\n\r\n";
            // Buffer to which we're going to write the commands
            byte[] WriteBuffer = new byte[1024];
            // We're passing ASCII characters
```

```
            ASCIIEncoding enc = new System.Text.ASCIIEncoding();
            // Pass the username to the server
            WriteBuffer = enc.GetBytes("USER " + UserName +
"\r\n");
            DialogueText += "I say:\r\nHere's the username: " +
UserName + "\r\n\r\n";
            netStream.Write(WriteBuffer, 0, WriteBuffer.Length);
            DialogueText += "Server says\r\n" +
strReader.ReadLine() + "\r\n\r\n";
            // Pass the password to the server
            WriteBuffer = enc.GetBytes("PASS " + PassWord +
"\r\n");
            DialogueText += "I say:\r\nHere's the password: " +
PassWord + "\r\n\r\n";
            netStream.Write(WriteBuffer, 0, WriteBuffer.Length);
            DialogueText += "Server says:\r\n" +
strReader.ReadLine() + "\r\n\r\n";
            // Now that we are (probably) authenticated, list the
messages
            WriteBuffer = enc.GetBytes("LIST\r\n");
            DialogueText += "I say:\r\nPlease list the
messages\r\n\r\n";
            netStream.Write(WriteBuffer, 0, WriteBuffer.Length);
            while (true)
            {
               ListMessage = strReader.ReadLine();
               //If ListMessage says "+OK 0 messages" then break
               //If not GUI will not respond
               if (ListMessage == "+OK 0 messages") { break; }
               else if (ListMessage == ".") // if this is the
last (most recent) message on the server.
               {

                  // It's the last message so display all info
for it and exit the loop
                  // WriteBuffer = enc.GetBytes("RETR " +
Convert.ToString(messagecount - 1) + "\r\n");
                  WriteBuffer = enc.GetBytes("RETR 1\r\n");
                  netStream.Write(WriteBuffer, 0,
WriteBuffer.Length);
                  MessageBody = "";
                  while (MessageBody != ".") //
                  {
                     // Get all information  for the most recent
information, line-by-line
                     MessageBody = strReader.ReadLine();
                     array[counter] = MessageBody;//store every
line in this array.
```

```
                    DialogueText += "Server says\r\n" +
MessageBody + "\r\n\r\n";
                    counter++; //increase array counter to store
on next line
                 }//End while
                 break;
              }//End if
              else
              {
                 // List the message summary (not the entire
message.)
                 messagecount++;// keep a count of how many
messages there are.
                 DialogueText += "Server says:\r\n" +
ListMessage + "\r\n\r\n";
                 continue;
              }//End else
           }//End while
           counter--;//get actual number of lines of message
           //The code below extracts the line of the sender.
           for (int jump = 0; jump <= counter; jump++)
           {
              tester = array[jump];
              if (tester.StartsWith("Return-Path:"))
              {fromline = jump;}//end if
           }//End for loop

           try
           {
              //The next couple of lines extract the actual
sender
              from = array[fromline];
              fromwho = from.Length - 13;
              from = from.Substring(13, fromwho);
              tempSender = from;
              phoneNum = from.Substring(from.IndexOf('<') + 1,
from.IndexOf('@') - from.IndexOf('<') - 1);
              // The following code looks for Matches of the
command sent by user
              for (int bodycount = 0; bodycount <= counter;
bodycount++)
              {
                 bodytest = array[bodycount];
                 matchingString = DisplayMatches(bodytest,
"\\D\\D\\D304");

                 if (matchingString.Length == 6)
                 {
                    bodyline = bodycount;
```

```
                  matchStringLower = matchingString.ToLower();
               }// end if
            }//end for l
            errorMessage = array[bodyline];
            //The switch statement decides which command was
input
            switch (matchStringLower)
            {
               case "ses304":
                  spanishToEnglish(from); //Ex: ste304
nickname : mensaje
                  break;
               case "sen304":
                  englishToSpanish(from);//Ex: ets304 nickname
: message
                  break;
               case "hlp304":
                  help(from);//Ex: hlp304
                  break;
               case "reg304":
                  regUser(from);//Ex: reg304 daniel valencia :
dva : verizon
                  break;
               case "ltu304":
                  listUsers();  //Ex:ltu304
                  break;
               case "dtl304":
                  deleteLingo(from);//Ex: dtl304 lol
                  break;
               case "adl304":
                  addLingo(from);//Ex: adl304 omg : oh my God
                  break;
               case "brl304":
                  borrarLingo(from);//Ex: brl304 amr
                  break;
               case "agl304":
                  agregarLingo(from);//Ex: agl304 amr : amor
                  break;
               case "lle304":
                  listEnglLingo();//Ex: brl304 lol
                  break;
               case "lls304":
                  listSpanLingo();//Ex: agl304 omg : oh my God
                  break;
               default:
                  //Check for syntax error
                  if (bodyline == 0)
                  {
                     bodyline = counter - 1;
```

```
                        errorMessage = array[bodyline];
                        instructionReply(from, "Error with: " +
errorMessage);
                    }//End if
                    else
                    {instructionReply(from, "Error with: " +
errorMessage);}//End else
                    break;
                }//End Switch
                //this writes it to a file called usersfile.txt to
see who emailed

                //Write sender to user file
                using (StreamWriter user = new
StreamWriter("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//userfile.
txt", true))
                { user.WriteLine(from); }//End using

                //Write the entire message to text file
                using (StreamWriter swine = new
StreamWriter("C:/Users/daniel/Documents/Visual Studio
2008/Projects/FinalThesisProject/FinalThesisProject/everythingfil
e.txt"))
                {
                   for (int hisdo = 0; hisdo <= counter; hisdo++)
                   { swine.WriteLine(array[hisdo]); }//End for
loop
                }//End using
            }
            catch { }
            //Delete  Message
            WriteBuffer = enc.GetBytes("DELE " +
Convert.ToString(messagecount - 1) + "\r\n");
            DialogueText += "I say:\r\nPlease delete the
messages\r\n\r\n";
            netStream.Write(WriteBuffer, 0, WriteBuffer.Length);
            DialogueText += "Server says:\r\n" +
strReader.ReadLine() + "\r\n\r\n";
            //End connection
            DialogueText += "I say:\r\nThanks, we will disconnect
now\r\n\r\n";
            WriteBuffer = enc.GetBytes("QUIT\r\n");
            netStream.Write(WriteBuffer, 0, WriteBuffer.Length);
            DialogueText += "Server says:\r\n" +
strReader.ReadLine();
         }//End if
         else { } //End else
      }//end pop timer
```

```
      //This function lists Lingo for the user
      private void listEnglLingo()
      {
         //Initialize variables
         string line = "";
         string temp = "";
         bool approved = false;
         string from = "";
         approved = approvedUser(phoneNum);//Check if approved
user
         //If approved create list
         if (approved == true)
         {
            StreamReader file = new
StreamReader("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//" + phoneNum + "Engl.txt");
            while ((line = file.ReadLine()) != null)
            {temp = temp + "\r\n" + line;}//End while
            file.Close();
            from = getUserNumWithPhoneNum(phoneNum);
         }//End if
         else { from = tempSender; temp = "Not registered";
}//End else
         instructionReply(from, temp);//Send list to user
         receiverTextBox.Text = temp;
      }//End listEnglLingo function

      //This function list Spanish lingo for the User
      private void listSpanLingo()
      {
         //Initialize variables
         string line = "";
         string temp = "";
         bool approved = false;
         string from = "";
         approved = approvedUser(phoneNum); //Check if approved
user
         //If approved create list of Spanish Lingo
         if (approved == true)
         {
            StreamReader file = new
StreamReader("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//" + phoneNum + "Span.txt");
            while ((line = file.ReadLine()) != null)
            { temp = temp + "\r\n" + line; }
            file.Close();//close file
```

```
            from = getUserNumWithPhoneNum(phoneNum);
         }//End if
         else { from = tempSender; temp = "Not registered";
}//End else
         instructionReply(from, temp);//Send Spanish lingo text
         receiverTextBox.Text = temp;
      }//End listSpanLingo function

      //This function allows user to add lingo in Spanish
      private void agregarLingo(string from)
      {
         //Initialize variables
         bool lingo_found = false;
         string confirmation = "";
         string line = "";
         string temp = "";
         string LingoToAdd = "";
         string lingo = "";
         string definition = "";
         int bodyline = 0;
         string bodytest, word = "";
         bool approved = false;
         approved = approvedUser(phoneNum);//Check if approved
user
         //If approved add lingo
         if (approved == true)
         {
            try
            {
               for (int bodycount = 0; bodycount <= counter;
bodycount++)
               {
                  bodytest = array[bodycount];
                  bool contains = bodytest.IndexOf("agl304",
StringComparison.OrdinalIgnoreCase) >= 0;
                  if (contains == true) { bodyline = bodycount;
}// end if
               }//End for loop

               //Extract message
               word = array[bodyline];
               lingo =
word.Substring(word.IndexOf("304")+3,word.IndexOf(':')-
word.IndexOf("304")-3).Trim().ToLower();
               definition = word.Substring(word.LastIndexOf(':')
+ 1).ToLower().Trim();
               LingoToAdd = " " + lingo + " - " + definition;
               StreamReader file = new
StreamReader("C://Users//daniel//Documents//Visual Studio
```

```
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//" + phoneNum + "Span.txt");

               while ((line = file.ReadLine()) != null)
               {
                  temp = line.Substring(0, line.IndexOf('-
')).Trim();
                  if (temp.Equals(lingo)) { lingo_found = true;
}//End if
                  else { }//End else
               }//End while
               file.Close();//close file
               if (lingo_found == true) { confirmation = "Lingo:
" + lingo + " Ya existe. No va ser agregado."; }
               else
               {
                  using (StreamWriter sw = new
StreamWriter("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//" + phoneNum + "Span.txt", true))
                  { sw.WriteLine(LingoToAdd); }
                  confirmation = "Lingo: " + lingo + " a sido
agregado.";
               }//End else
            }//End try
            catch { }//End Catch
         }//End if
         else { confirmation = "Usuario no registrado."; }//End
else
         instructionReply(from, confirmation);
         receiverTextBox.Text = confirmation;
      }//end agregarLingo function

      //Allow user to delete lingo in Spanish
      private void borrarLingo(string from)
      {
         //Initialize variable
         int bodyline = 0;
         int lookupword = 0;
         string bodytest, word = "";
         string lingoDelete = "";
         int count = 0;
         string line;
         const int ARRAY_LENGTH1 = 1000;
         string confirmation = "";
         string temp = "";
         bool lingo_found = false;
         bool approved = false;
         string[] array1 = new string[ARRAY_LENGTH1];
```

```
         approved = approvedUser(phoneNum);//Check if approved
user
         //If approved delete lingo in Spanish
         if (approved == true)
         {
            try
            {
               for (int bodycount = 0; bodycount <= counter;
bodycount++)
               {
                  bodytest = array[bodycount];
                  bool contains = bodytest.IndexOf("brl304",
StringComparison.OrdinalIgnoreCase) >= 0;
                  if (contains == true) { bodyline = bodycount;
}// end if
               }//end for loop

               //Extract message
               word = array[bodyline];
               lookupword = word.Length - 7;
               word = word.Substring(7, lookupword);
               lingoDelete = word.Trim().ToLower();

               //Check if lingo in file and Delete
               StreamReader file = new
StreamReader("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//" + phoneNum + "Span.txt");
               while ((line = file.ReadLine()) != null)
               {
                  temp = line.Substring(0, line.IndexOf('-
')).Trim();
                  if (temp.Equals(lingoDelete))
                  {
                     count--; //decrement counter because it will
overwrite previous line
                     lingo_found = true;
                  }//End if
                  else
                     array1[count] = line;
                  count++;
               }//End while
               count--;//decrement counter to get official lines
of file
               if (lingo_found == true) { confirmation = "Lingo:
" + lingoDelete + " Borrado"; }//End if
               else { confirmation = "Lingo " + lingoDelete + "
no fue encontrado."; }//ENd else
               file.Close();//close file
```

```
               //Rewrite file without deleted lingo
               StreamWriter sw = new
StreamWriter("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//" + phoneNum + "Span.txt", false);
               for (int i = 0; i <= count; i++)
               {sw.WriteLine(array1[i]);}//End for
               sw.Close();//close object sw
            }//End try
            catch {}//End catch
         }//End if
         else { confirmation = "Usuario no registrado."; }//End
else
         instructionReply(from, confirmation);//Send confirmation
about lingo
         receiverTextBox.Text = confirmation;
      }// end delete lingo function

      //Allow user to add Lingo in English file
      private void addLingo(string from)
      {
         //Initialize variable
         bool lingo_found = false;
         string confirmation = "";
         string line = "";
         string temp = "";
         string LingoToAdd = "";
         string lingo = "";
         string definition = "";
         int bodyline = 0;
         string bodytest, word = "";
         bool approved = false;
         approved = approvedUser(phoneNum);//Check if approved
user
         //If approved add lingo
         if (approved == true)
         {
            try
            {
               for (int bodycount = 0; bodycount <= counter;
bodycount++)
               {
                  bodytest = array[bodycount];
                  bool contains = bodytest.IndexOf("adl304",
StringComparison.OrdinalIgnoreCase) >= 0;
                  if (contains == true) { bodyline = bodycount;
}// End if
               }//End for loop
```

```
                //Extract message
                word = array[bodyline];
                lingo = word.Substring(word.IndexOf("304") + 3,
word.IndexOf(':') - word.IndexOf("304") - 3).Trim().ToLower();
                definition = word.Substring(word.LastIndexOf(':')
+ 1).ToLower().Trim();
                LingoToAdd = " " + lingo + " - " + definition;
                //Check if lingo found in file
                StreamReader file = new
StreamReader("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//" + phoneNum + "Engl.txt");
                while ((line = file.ReadLine()) != null)
                {
                   temp = line.Substring(0, line.IndexOf('-
')).Trim();
                   if (temp.Equals(lingo)) { lingo_found = true;
}//End if
                   else { }//End else
                }//End while
                file.Close();//close file
                //If already found do not add lingo
                if (lingo_found == true) { confirmation = "Lingo:
" + lingo + " Already exists. Will not add";}//End if
                else
                {
                   //Add lingo to file if not found
                   using (StreamWriter sw = new
StreamWriter("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//" + phoneNum + "Engl.txt", true))
                   { sw.WriteLine(LingoToAdd); } //append number
and carrier to file.
                   confirmation = "Lingo: " + lingo + " Has been
added";
                }//End else
             }//End try
             catch{}//End catch
          }//End if
          else { confirmation = "User not registered"; }//End else
          instructionReply(from,confirmation);//Send confirmation
about adding lingo
          receiverTextBox.Text = confirmation;
       }//End adding lingo function

       //Delete lingo from English lingo file
       private void deleteLingo(string from)
       {
```

```
         //Initialize variables
         int bodyline = 0;
         int lookupword = 0;
         string bodytest, word = "";
         string lingoDelete = "";
         int count = 0;
         string line;
         const int ARRAY_LENGTH1 = 1000;
         string confirmation = "";
         string temp = "";
         bool lingo_found = false;
         bool approved = false;
         string[] array1 = new string[ARRAY_LENGTH1];
         approved = approvedUser(phoneNum);//Check if approved
user
         //If approved delete lingo
         if (approved == true)
         {
            try
            {
               for (int bodycount = 0; bodycount <= counter;
bodycount++)
               {
                  bodytest = array[bodycount];
                  bool contains = bodytest.IndexOf("dtl304",
StringComparison.OrdinalIgnoreCase) >= 0;
                  if (contains == true) { bodyline = bodycount;
}// end if
               }//End for loop
               //Extract message
               word = array[bodyline];
               lookupword = word.Length - 7;
               word = word.Substring(7, lookupword);
               lingoDelete = word.Trim().ToLower();
               //Read file to see if lingo exists
               StreamReader file = new
StreamReader("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//" + phoneNum + "Engl.txt");
               while ((line = file.ReadLine()) != null)
               {
                  temp = line.Substring(0, line.IndexOf('-
')).Trim();
                  if (temp.Equals(lingoDelete))
                  {
                     count--;
                     lingo_found = true;
                  }
                  else
```

```
                    array1[count] = line;
                   count++;
                }
                count--;
                //If lingo found then delete it
                if (lingo_found == true) { confirmation = "Lingo:
" + lingoDelete + " Deleted"; }
                //Else no lingo was found
                else { confirmation = "Lingo " + lingoDelete + "
not found"; }
                file.Close();//close file
                //Rewrite file without deleted lingo
                StreamWriter sw = new
StreamWriter("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//" + phoneNum + "Engl.txt", false);
                for (int i = 0; i <= count; i++)
                {sw.WriteLine(array1[i]);}//End for loop
                sw.Close();//close object sw
             }//End try
             catch {}//End catch
          }//End if
          else { confirmation = "User not registered"; }
          instructionReply(from,confirmation);//Send confirmation
to user
          receiverTextBox.Text = confirmation;
       }//End deleteLingo function

       //This function list user for user that request it
       private void listUsers()
       {
          //Initialize variables
          string line = "";
          string temp = "";
          bool approved = false;
          string from = "";
          approved = approvedUser(phoneNum);//Check if approved
user
          //If approved create user message
          if (approved == true)
          {
             StreamReader file = new
StreamReader("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//usersfile.txt");
             while ((line = file.ReadLine()) != null)
             {
                temp = temp + "\r\n" + line.Substring(0,
line.LastIndexOf(':'));
```

```
               if (line.Contains(phoneNum)) { from =
line.Substring(line.LastIndexOf(':') + 1);}//End if
               else { }//End else
            }
            file.Close();//close file
         }//End if
         else { from = tempSender; temp = "User not registered";
}//End else
         instructionReply(from, temp);//Send list of users
         receiverTextBox.Text = temp;
      }//End listUsers function

      //This function check if a user is approved using the
nickname
      public bool approvedUser(string nickname)
      {
         //Initialize variables
         string line = "";
         string temp = "";
         bool nickname_found = false;
         //Check if you found in file
         StreamReader file = new
StreamReader("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//usersfile.txt");
         while ((line = file.ReadLine()) != null)
         {
            temp = line.Substring(line.IndexOf(':'));
            if (temp.Contains(nickname)) { nickname_found = true;
}//End if
            else { }//End else
         }//End while
         file.Close();//close file
         return nickname_found;
      }//End approvedUser function

      //This function returns help syntax for user
      private void help(string from)
      {
         //Initialize variable
         string confirmation = "1. To register. reg304 name : 3
letter nickname : provider. \r\n2. To add Lingo. adl304 lingo
abbreviation : lingo definition.\r\nEx: adl304 omg : oh my
God\r\n 4. To delete lingo. dtl304 lingo abbreviation. \r\nEx:
dtl304 omg \r\n5. Para agregar lingo. Ex: agl304 amr : amor \r\n
6. Para borrar lingo. Ejemplo: brl304 amr \r\n7. To show user.
Mostrar usuarios. ltu304 \r\n8. Send a message in English. sen304
3 letter nickname: message. \r\n9. Send message in Spanish.
```

```
ses304 sobrenombre de 3 letras : mensaje. \r\n10. List Lingo
Spanish. Ex: lls304  \r\n11. List Lingo English. Ex: lle304 ";
         from = getUserNumWithPhoneNum(phoneNum);//Get the user
with the phone number
         instructionReply(from, confirmation);//Send message to
user
         receiverTextBox.Text = confirmation;
      }//end help function

      //This function registers a new user
      private void regUser(string from)
      {
         //Initialize variables
         string confirmation = "";
         string numberAndCarrier = "";
         string provider = "";
         string nickname = "";
         string user = "";
         string line = "";
         string temp = "";
         bool providerNotRecognized = false;
         bool nickname_found = false;
         int firstIndex = 0;
         int bodyline = 0;
         string bodytest, word = "";
         //Check if valid user
         try
         {
            for (int bodycount = 0; bodycount <= counter;
bodycount++)
            {
               bodytest = array[bodycount];
               bool contains = bodytest.IndexOf("reg304",
StringComparison.OrdinalIgnoreCase) >= 0;
               if (contains == true) { bodyline = bodycount; }//
end if
            }//end for loop
            //Extract message
            word = array[bodyline];
            firstIndex = word.IndexOf(':');
            user = word.Substring(word.IndexOf("304") + 3,
word.IndexOf(':')-word.IndexOf("304")-3).Trim().ToLower();
            nickname =
word.Substring(word.IndexOf(':')+1,word.LastIndexOf(':')-
firstIndex-1).Trim().ToLower();
            provider = word.Substring(word.LastIndexOf(':') +
1).Trim().ToLower();
            //If  nickname previously used then do not add
```

```
            StreamReader file = new
StreamReader("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//usersfile.txt");
            while ((line = file.ReadLine()) != null)
            {
              temp = line.Substring(line.IndexOf(':') + 1,
line.LastIndexOf(':') - line.IndexOf(':') - 1).Trim().ToLower();
            if (temp.Equals(nickname)) { nickname_found =
true;}//End if
            else { }//End else
            }//End while
            file.Close();//close file
            if (nickname_found == true) { }
            //If nickname not found that add it to the users text
file
            else
            {
               switch (provider)
               {
                  case "verizon":
                     numberAndCarrier = user + ":" + nickname +
":" + phoneNum + "@vzwpix.com";
                     break;
                  //if second selected from combo box then add
@mms.att.net>
                  case "atnt":
                     numberAndCarrier = user + ":" + nickname +
":" + phoneNum + "@mms.att.net";
                     break;
                  //if third selected from combo box then add
@mms.mycricket.com>
                  case "cricket":
                     numberAndCarrier = user + ":" + nickname +
":" + phoneNum + "@sms.mycricket.com";
                     break;
                  //if fourth selected from combo box then add
@pm.sprint.com>
                  case "sprint":
                     numberAndCarrier = user + ":" + nickname +
":" + phoneNum + "@pm.sprint.com";
                     break;
                  case "cingular":
                     numberAndCarrier = user + ":" + nickname +
":" + phoneNum + "@mms.mycingular.com";
                     break;
                  default:
                     providerNotRecognized = true;
                     break;
```

```
            }//end switch
         }
      }
      catch { }//End catch
      if (providerNotRecognized == true){confirmation =
"Provider was not recognized. Options are verizon, cingular,
sprint, cricket, atnt.";}
      else
      {
         using (StreamWriter sw = new
StreamWriter("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//usersfile.txt", true))
         {
            sw.WriteLine(numberAndCarrier); //append number
and carrier to file.
         }//end writing
         string engSourceFile =
"C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//standardE
nglishLingo.txt";
         string engDestFile =
"C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//" + phoneNum + "Engl.txt";
         System.IO.File.Copy(engSourceFile, engDestFile,
true);
         string spanSourceFile =
"C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//standardS
panishLingo.txt";
         string spanDestFile =
"C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//" + phoneNum + "Span.txt";
         System.IO.File.Copy(spanSourceFile, spanDestFile,
true);
         if (nickname_found == true) { confirmation = "User
with nickname " + nickname + " already exists did not register.";
}
         else { confirmation = "User: " + user + " : " +
nickname + " has been registered."; }
      }//End else
      instructionReply(from, confirmation);//Send confirmation
      receiverTextBox.Text = confirmation;
   }//end register user function

   //This function gets user number with phone number
   public string getUserNumWithPhoneNum(string num)
```

```
      {
         //Initialize variables
         string line = "";
         string temp = "";
         StreamReader file = new
StreamReader("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//usersfile.txt");
         while ((line = file.ReadLine()) != null)
         {
            if (line.Contains(num)) { temp =
line.Substring(line.LastIndexOf(':')+1); }//End if
            else { }//End else
         }//End while
         file.Close();//close file
         return temp;
      }//End getUserNumWithPhoneNum fucntion

      //This function gets user number with nickname
      public string getUserNumWithNickname(string nickname)
      {
         //Initialize variables
         string line = "";
         string temp = "";
         string phonenumber = "";
         StreamReader file = new
StreamReader("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//usersfile.txt");
         while ((line = file.ReadLine()) != null)
         {
            temp = line.Substring(line.IndexOf(':') + 1,
line.LastIndexOf(':') - line.IndexOf(':') - 1).Trim().ToLower();
            if (temp.Equals(nickname)) { phonenumber =
line.Substring(line.LastIndexOf(':') + 1); }//End if
            else { }//End else
         }//End while
         file.Close();//close file
         return phonenumber;
      }// End getUserNumWithNickname function


      public string getUserNicknamewithNum(string num)
      {
         //Initialize variables
         string line = "";
         string temp = "";
         StreamReader file = new
StreamReader("C://Users//daniel//Documents//Visual Studio
```

```
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//usersfile.txt");
         while ((line = file.ReadLine()) != null)
         {
            if (line.Contains(num)) { temp =
line.Substring(line.IndexOf(':') + 1, line.LastIndexOf(':') -
line.IndexOf(':')-1); }//End if
            else { }//End else
         }//End while
         file.Close();//close file
         return temp;
      }//End getUserNicknamewithNum function

      //This function gets all users.
      public string getAllUsers()
      {
         string line = "";
         string phonenumber = "";
         StreamReader file = new
StreamReader("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//usersfile.txt");
         while ((line = file.ReadLine()) != null)
         {
            phonenumber = phonenumber + ";" +
line.Substring(line.LastIndexOf(':') + 1);
         }
         file.Close();//close file
         return phonenumber;
      }//End getAllUsers function

      //This function extracts a message converts it from English
to Spanish
      //and calls a function to send to recipient
      private void englishToSpanish(string from)
      {
         //Initialize variables
         int bodyline = 0;
         string nickname = "";
         string bodytest, word = "";
         string languagePair = "en|es";
         string translation = "";
         string sendNick = "";
         string sender = "";
         bool approved = false;
         bool approvedReceiver = false;
         approved = approvedUser(phoneNum);//Check if approved
user
         //If approved then translate and sned
```

```
         if (approved == true)
         {
            try
            {
               for (int bodycount = 0; bodycount <= counter;
bodycount++)
               {
                  bodytest = array[bodycount];
                  bool contains = bodytest.IndexOf("sen304",
StringComparison.OrdinalIgnoreCase) >= 0;
                  if (contains == true) { bodyline = bodycount;
}// end if
               }//end for loop
               //Extract message
               word = array[bodyline];//puts line in word
               nickname = word.Substring(word.IndexOf("304") + 3,
word.IndexOf(':') - word.IndexOf("304") - 3).Trim().ToLower();
               word = word.Substring(word.IndexOf(':')+1);//get
substring which is the word the want
               word = englishLingoToText(word);//Convert English
Lingo
               translation = translateText(word,
languagePair);//Translate message
               //The following converts text file to audio file
               try
               {
                  // create the ProcessStartInfo using "cmd" as
the program to be run,
                  // and "/c " as the parameters.
                  // Incidentally, /c tells cmd that we want it
to execute the command that follows,
                  // and then exit.
                  System.Diagnostics.ProcessStartInfo
procStartInfo = new System.Diagnostics.ProcessStartInfo("cmd",
"/c " + "espeak -v en -g 10 -w " + Convert.ToString(tempphoneNum)
+ ".wav \"" + word + "\"");
                  // The following commands are needed to
redirect the standard output.
                  // This means that it will be redirected to the
Process.StandardOutput StreamReader.
                  procStartInfo.RedirectStandardOutput = true;
                  procStartInfo.UseShellExecute = false;
                  // Do not create the black window.
                  procStartInfo.CreateNoWindow = true;
                  // Now we create a process, assign its
ProcessStartInfo and start it
                  System.Diagnostics.Process proc = new
System.Diagnostics.Process();
                  proc.StartInfo = procStartInfo;
```

```
                  proc.Start();
                  // Get the output into a string
                  string result =
proc.StandardOutput.ReadToEnd();
               }//End try
               catch { } //End catch
               sendNick =
getUserNicknamewithNum(phoneNum).Trim().ToLower();
               //mss is sent then the message will get sent out
to every user
               if (nickname == "mss")
               {
                  word = sendNick + " says: " + word + " = \n";
                  from = getUserNumWithNickname(nickname);
                  sender = getAllUsers();
                  sender = sender.Substring(1);
                  sender = sender.Replace(";", ", ");
                  receiverTextBox.Text = word + translation;
                  senderTextBox.Text = word + translation;
                  defintionReply(sender, "", word, translation);
               }//End if
               //Else only send to specified user
               else
               {
                  approvedReceiver = approvedUser(nickname);
                  if (approvedReceiver == true)
                  {
                     word = sendNick + " says: " + word + " =
\n";
                     from = getUserNumWithNickname(nickname);
                     sender = getUserNumWithPhoneNum(phoneNum);
                     receiverTextBox.Text = word + translation;
                     senderTextBox.Text = word + translation;
                     defintionReply(from, sender, word,
translation);
                  }//End if
                  else
                  { instructionReply(from, "Invalid nick name " +
nickname); }//End else
               }//End else
            }//End try
            catch {}
         }
         else { instructionReply(from,"Not an approved user."); }
      }//End englishToSpanish function

      //This function changes english lingo to text
      public string englishLingoToText(string b)
      {
```

```
         //Initialize variables
         string line;
         int found = 0;
         string find = "";
         string replace = "";
         //Replace symbols with other.
         b = b + ".";
         b = b.Replace("?", " ? ");
         b = b.Replace("..",".");
         b = b.Replace(".", " . ");
         b = b.Replace("!", " ! ");
         //Finds lingo and replace with regular text
         StreamReader file = new
StreamReader("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//" + phoneNum + "Engl.txt");
         while ((line = file.ReadLine()) != null)
         {
            find = line.Substring(0, line.IndexOf("-"));
            found = line.IndexOf("-");
            replace = line.Substring(found + 1).Trim();
            b = Regex.Replace(b, find, " " + replace + " ",
RegexOptions.IgnoreCase);
         }//End while
         file.Close(); //Close file
         return b;
      }//End englishtext to lingo function

      //This function extracts a message converts it from Spanish
to English
      //and calls a function to send to recipient
      private void spanishToEnglish(string from)
      {
         //Initialize variables
         int bodyline = 0;
         string bodytest, word = "";
         string languagePair = "es|en";
         string translation = "";
         string nickname = "";
         string sendNick = "";
         string sender = "";
         bool approvedReceiver = false;
         bool approved = false;
         //Check if approved user
         approved = approvedUser(phoneNum);
         if (approved == true)
         {
         try
         {
```

```
            for (int bodycount = 0; bodycount <= counter;
bodycount++)
            {
               bodytest = array[bodycount];
               bool contains = bodytest.IndexOf("ses304",
StringComparison.OrdinalIgnoreCase) >= 0;
               if (contains == true){bodyline = bodycount;}// end
if
            }//end for loop
            word = array[bodyline];//puts line in word
            nickname = word.Substring(word.IndexOf("304") + 3,
word.IndexOf(':') - word.IndexOf("304") - 3).Trim().ToLower();
            word = word.Substring(word.IndexOf(':') + 1);//get
substring which is the word the want
            //Replace Captial vowels with the letter and a tilde
            word = spanishLingoToText(word);
            word = word.Replace('A','á');
            word = word.Replace('E','é');
            word = word.Replace('I','í');
            word = word.Replace('O','ó');
            word = word.Replace('U','ú');
            translation = translateText(word, languagePair);
            //Create the audio file
            try
            {
               // create the ProcessStartInfo using "cmd" as the
program to be run,
               // and "/c " as the parameters.
               // Incidentally, /c tells cmd that we want it to
execute the command that follows,
               // and then exit.
               ProcessStartInfo procStartInfo = new
System.Diagnostics.ProcessStartInfo("cmd", "/c " + "espeak -v es
-g 10 -w " + Convert.ToString(tempphoneNum) + ".wav \"" + word +
"\"");
               // The following commands are needed to redirect
the standard output.
               // This means that it will be redirected to the
Process.StandardOutput StreamReader.
               procStartInfo.RedirectStandardOutput = true;
               procStartInfo.UseShellExecute = false;
               // Do not create the black window.
               procStartInfo.CreateNoWindow = true;
               // Now we create a process, assign its
ProcessStartInfo and start it
               System.Diagnostics.Process proc = new
System.Diagnostics.Process();
               proc.StartInfo = procStartInfo;
               proc.Start();
```

```
               string result = proc.StandardOutput.ReadToEnd();
               // Get the output into a string
            }
            catch {}//End creating audio file
            //Get user nickname with phone number
            sendNick =
getUserNicknamewithNum(phoneNum).Trim().ToLower();
            //If nickname is mss then send mass email to all
users.
            if (nickname == "mss")
            {
               word = sendNick + " says: " + word + " = \n";
               from = getUserNumWithNickname(nickname);
               sender = getAllUsers();
               receiverTextBox.Text = word + translation;
               senderTextBox.Text = word + translation;
               defintionReply(sender, "", word, translation);
            }//End if
            else
            {
               approvedReceiver = approvedUser(nickname);
               if (approvedReceiver == true)
               {
                  word = sendNick + " says: " + word + " = \n";
                  from = getUserNumWithNickname(nickname);
                  sender = getUserNumWithPhoneNum(phoneNum);
                  receiverTextBox.Text = word + translation;
                  senderTextBox.Text = word + translation;
                  defintionReply(from, sender, word,
translation);
               }//End if
               else
               {instructionReply(from, "Invalid nick name " +
nickname);}//End else
            }//End else
         }
         catch { }//End catch
         }
         else { instructionReply(from, "No eres un usario
registrado."); }//End else
      }//end function englishToSpanish

      //This function converts Spanish lingo to regular text
      public string spanishLingoToText(string b)
      {
         //Initialize variables
         string line;
         int found = 0;
         string find = "";
```

```
         string replace = "";
         b = b + ".";
         b = b.Replace("?", " ? ");
         b = b.Replace("..", ".");
         b = b.Replace(".", " . ");
         b = b.Replace("!", " ! ");
         //Look for lingo and replace it with appropiate text
         StreamReader file = new
StreamReader("C://Users//daniel//Documents//Visual Studio
2008//Projects//FinalThesisProject//FinalThesisProject//User//SPA
N304//" + phoneNum + "Span.txt");
         while ((line = file.ReadLine()) != null)
         {
            find = line.Substring(0, line.IndexOf("-"));
            found = line.IndexOf("-");
            replace = line.Substring(found + 1).Trim();
            b = Regex.Replace(b, find, " " + replace + " ",
RegexOptions.IgnoreCase);
         }//end while
         file.Close(); //close file
         return b;//Return text
      }//End spanishLingoToText function

      //This function accesses Google translate API and
translates the message
      public string translateText(string input, string
languagePair)
      {
         dictionaryTextBox.Clear();
         //Initialize variables
         string buffer = "";
         string decodedMessage = "";
         string definition = "";
         int j = 0;
         input = input.Trim();
         string[] split = input.Split(new Char[] { ' ' });
         foreach (string s in split){j++;}
         if (j == 2)
         {input = input.Replace('.', ' ').Trim();}
         string FilePath = "C:/Users/daniel/Documents/Visual
Studio 2008/Projects/FinalThesisProject/resultfile.txt";
         string url =
String.Format("http://www.google.com/translate_t?hl=en&ie=UTF8&te
xt={0}&langpair={1}", input, languagePair);
         WebClient webClient = new WebClient();
         webClient.Encoding =
System.Text.Encoding.GetEncoding("iso-8859-1");
         string result = webClient.DownloadString(url);
         using (StreamWriter write = new StreamWriter(FilePath))
```

```
         {write.WriteLine(result);}
         //Extracts Dictionary definition
         try
         {
            definition =
result.Substring(result.IndexOf("link>View detailed
dictionary</a></span></h3><ol><li>") + 53, 800);
            definition = definition.Substring(0,
definition.IndexOf("</div></div></div> </div>"));
            definition = definition.Replace("<ol><li>", ": ");
            definition = definition.Replace("</li><li>", ", ");
            definition = definition.Replace("</li></ol>", ". ");
            definition = definition.Replace(". ,", ". ");
            dictionaryTextBox.Text = definition;
         }//End try
         catch { }//End catch
         result =
result.Substring(result.IndexOf("this.style.backgroundColor='#fff
'"));
         result = result.Substring(0,
result.IndexOf("</div></div><div"));
         //Extract translation of message
         int i = 0;
         while ((i =
result.IndexOf("this.style.backgroundColor='#fff'", i)) != -1)
         {
            buffer =
result.Substring((i+35),(result.IndexOf("</span>",i)-(i+35)));
            decodedMessage = decodedMessage + buffer;
            i++;
         }
         decodedMessage = decodedMessage.Replace("&#39;", "'");
         decodedMessage = decodedMessage.Replace("&quot;", " ");
         if (j == 2)
         {result = decodedMessage + ". \n" + definition;}//End if
         else { result = decodedMessage; }
         return result;//Return translation
      }// end TranslateText function which translates text

      //This function looks for a match of any command.
      private static string DisplayMatches(string input, string
expression)
      {
         string matchingCode = "";
         foreach (var regexMatch in Regex.Matches(input,
expression))
            matchingCode = Convert.ToString(regexMatch);
         return matchingCode;//Retrun Command
      }//end function DisplayMatches
```

```
      //This function sends an Email to all the users.
      static void defintionReply(string from, string sender,
string word, string definition)
      {
         try
         {
            //create mail message
            //new instance mail of type MailMessage
            MailMessage mail = new MailMessage();
            //set the addresses of who is sending email
            mail.From = new
MailAddress("dvalenci85@triad.rr.com");
            //this sets to who the email is going, gets users
from userfile.txt
            mail.To.Add(from);
            //set the content
            mail.CC.Add(sender);
            //mail.Bcc.Add(sender);
            mail.Subject = "Language Translator";
            mail.Body = word + definition;
            Attachment Attach = new
Attachment("C:/Users/daniel/Documents/Visual Studio
2008/Projects/FinalThesisProject/FinalThesisProject/bin/Debug/" +
Convert.ToString(tempphoneNum) + ".wav");
            mail.Attachments.Add(Attach);
            //set the message
            //sets the smtp server name
            SmtpClient smtp = new SmtpClient("smtp-
server.triad.rr.com");
            //sets the credentials
            smtp.Credentials = new
NetworkCredential("dvalenci85", "dv4507888", "triad.rr.com");
            //send the email
            smtp.Send(mail);
            if (tempphoneNum == 10) { tempphoneNum = 0; }
            else { tempphoneNum++; }
         }//End try
         catch {}//End catch

         //The code below does the same, but on School Campus
         /*try
         {
            MailMessage mail = new MailMessage();
            //set the addresses of who is sending email
            mail.From = new MailAddress("dvalenci@ncat.edu");
            //this sets to who the email is going, gets users
from userfile.txt
            mail.To.Add(from);
```

```
            //set the content
            mail.CC.Add(sender);
            //mail.Bcc.Add(sender);
            mail.Subject = "Language Translator";
            //<User Name>  send a nice message or picture to
Daniel Valencia Please.";
            mail.Body = word + definition;
            Attachment Attach = new
Attachment("C:/Users/daniel/Documents/Visual Studio
2008/Projects/FinalThesisProject/FinalThesisProject/bin/Debug/" +
Convert.ToString(tempphoneNum) + ".wav");
            mail.Attachments.Add(Attach);
            //set the message
            //sets the smtp server name
            SmtpClient smtp = new SmtpClient("smtp.ncat.edu");
            //sets the credentials
            smtp.Credentials = new NetworkCredential("dvalenci",
"dval0550", "ncat.edu");
            //send the email
            smtp.Send(mail);
            if (tempphoneNum == 10) { tempphoneNum = 0; }
            else { tempphoneNum++; }
         }
         catch{}*/

      }//end definitionReply function.

      //This function emails Confirmation for other commands to
users.
      static void instructionReply(string from, string message)
      {
         try
         {
            //create mail message
            //new instance mail of type MailMessage
            MailMessage mail = new MailMessage();
            //set the addresses of who is sending email
            mail.From = new
MailAddress("dvalenci85@triad.rr.com");
            //this sets to who the email is going, gets users
from userfile.txt
            mail.To.Add(from);
            //set the content
            mail.Subject = "Language Translator";
            mail.Body = message;
            //set the message
            //sets the smtp server name
            SmtpClient smtp = new SmtpClient("smtp-
server.triad.rr.com");
```

```
            //sets the credentials
            smtp.Credentials = new NetworkCredential("dvalenci",
"dv45078", "triad.rr.com");
            //send the email
            smtp.Send(mail);
         }//End try
         catch { }//End catch

         //The code below sends confirmation as well but from
School campus
         /*try
         {
            MailMessage mail = new MailMessage();
            //set the addresses of who is sending email
            mail.From = new MailAddress("dvalenci@ncat.edu");
            //this sets to who the email is going, gets users
from userfile.txt
            mail.To.Add(from);
            //set the content
            mail.Subject = "Language Translator";
            //<User Name>  send a nice message or picture to
Daniel Valencia Please.";
            mail.Body = message;
            //set the message
            //sets the smtp server name
            SmtpClient smtp = new SmtpClient("smtp.ncat.edu");
            //sets the credentials
            smtp.Credentials = new NetworkCredential("dvalenci",
"dval0550", "ncat.edu");
            //send the email
            smtp.Send(mail);
          }
         catch{}*/
      }
      //This button stops the entire application
      private void stopButton_Click(object sender, EventArgs e)
      {
         popTimer.Enabled = false;
      }

      //This funcion is initialized when the program starts
      //Its purpose is to send random picture messages to the
user.
      //Only between 9 am and 8 pm.
      private void randomTimer_Tick(object sender, EventArgs e)
      {
         //Initialize variables
         string CCs = "";
         Random defRand = new Random();
```

```
         int num1 = defRand.Next(0, 40);
         int num2 = defRand.Next(0, 20);
         int numFiles;

         string dir = @"C:\Users\daniel\Documents\Visual Studio
2008\Projects\FinalThesisProject\FinalThesisProject\PicturesFile"
;
         string[] files;
         files = Directory.GetFiles(dir);
         numFiles = files.Length;
         int num3 = defRand.Next(0, numFiles);

         //This will only send picture between 9 am and 8 pm
         //Check the time.
         DateTime today = DateTime.Today;
         DateTime start_timer = today.AddHours(9);
         DateTime stop_timer = today.AddHours(20);
         //If the time falls in the boudaries then send picture
         if ((DateTime.Now > start_timer) && (DateTime.Now <
stop_timer))
         {
            if (num1 == num2)
            {
               CCs = getAllUsers();
               CCs = CCs.Substring(1);
               CCs = CCs.Replace(";", ", ");
               //create mail message
               //new instance mail of type MailMessage
               MailMessage mail = new MailMessage();
               mail.From = new
MailAddress("dvalenci85@triad.rr.com");
               mail.To.Add(CCs);
               //set the content
               mail.Subject = "Random Picture = Foto al azar.";
               //<User Name>  send a nice message or picture to
Daniel Valencia Please.";
               mail.Body = "This message was sent to you by learn
in class.";
               Attachment Attach = new
Attachment("C:/Users/daniel/Documents/Visual Studio
2008/Projects/FinalThesisProject/FinalThesisProject/PicturesFile/
" + Convert.ToString(num3) + ".jpg");
               mail.Attachments.Add(Attach);
               //set the message
               //sets the smtp server name
               SmtpClient smtp = new SmtpClient("smtp-
server.triad.rr.com");
               //sets the credentials
```

```
                smtp.Credentials = new
NetworkCredential("dvalenci85", "dv4507888", "triad.rr.com");
                //send the email
                smtp.Send(mail);

                //This send message from School Campus
               /* MailMessage mail = new MailMessage();
                mail.From = new MailAddress("dvalenci@ncat.edu");
                mail.To.Add(CCs);
                //set the content
                mail.Subject = "Random Picture = Foto al azar.";
                //<User Name>  send a nice message or picture to
Daniel Valencia Please.";
                mail.Body = "This message was sent to you by learn
in class.";
                Attachment Attach = new
Attachment("C:/Users/daniel/Documents/Visual Studio
2008/Projects/FinalThesisProject/FinalThesisProject/PicturesFile/
" + Convert.ToString(num3) + ".jpg");
                mail.Attachments.Add(Attach);
                //set the message
                //sets the smtp server name
                SmtpClient smtp = new SmtpClient("smtp.ncat.edu");
                //sets the credentials
                smtp.Credentials = new
NetworkCredential("dvalenci", "dval0550", "ncat.edu");
                //send the email
                smtp.Send(mail);*/

            }//End if
            else { }//End else
         }//End if
         else { }//End else
      }//End randomTimer function
   }//End class
}//End namespace
```