NEAR EAST UNIVERSITY

INSTITUTE OF APPLIED AND SOCIAL SCIENCES

DEVELOPMENT OF VOICE INTERACTIVE SYSTEM FOR BANK TRANSACTION

OMAR SAEED

Master Thesis

Department Of Computer Engineering

Nicosia - 2003

Omar Saeed : Development of Voice Interactive System for Bank Transaction

Approval of the Graduate School of Applied and Social Sciences

Prof. Dr. Fakhraddin Mamedov
Director

We certify this thesis is satisfactory for the award of the Degree of Master of Science in Computer Engineering

Examining Committee in charge:

Assoc. Prof. Dr. Doğan Ibrahim, Committee Chairman, Chairman of Computer Engineering Dept., NEU

Assis. Prof. Dr. Doğan Haktanir, Committee Member, Elec. & Elec. Dept., NEU

Assoc. Prof. Dr. Ilham Huseynov, Committee Member, CIS Dept., NEU

Assoc. Prof. Dr. Rahib Abiyev, Supervisor, Computer Engineering Dept., NEU

ACKNOWLEDGEMENTS

First of all, I would like to extend my sincere and humble gratitude to Almighty Allah who endowed me potential and ability to make solid contribution to the already existing ocean of knowledge.

I wish to express my gratitude and sincere appreciation to my supervisor Dr. Rahib Abiyev, Department of Computer Engineering, Near East University for his kind patronage, guidance and concern not only during this thesis but always. I am really thankful to him for being there whenever I needed his advice.

With great respect and gratitude, I wish to thank Dr. Fakhruddin Mamedov, Dean of Engineering faculty, Near East University for providing me necessary atmosphere in the completion of MSc Computer Engineering.

My special note of thanks to all my friends, who made my stay in Near East University possible. I am really thankful to them for bearing with me and my moods during my studies.

No words could express the love and understanding my parents have extended to me during the whole period of my education and throughout my life. The same goes for the rest of my family who always had a word of appreciation and always stood beside me.

I would really like to mention name of Jonathan Eisenzopf member of Ferrum Group, LLC based in Reston, Virginia who give me his precious articles and provided me with consultation that helped me a lot to get through this project.

ABSTRACT

In last few decades, some of computer scientists have realized that speech recognition and output to interactive systems would undoubtedly improve their usability. In recent years, the Internet has dramatically changed people's life styles, and using the power of the Web, various kinds of services can widely be provided. However, most people still use a telephone more often than they use the Internet, and continuous improvement of Automated Speech Recognition (ASR) and text-to-speech (TTS) technology can bridge the gap between the Web and the telephony communication.

VoiceXML is a standard technology to make Internet content and information widely accessible via voice and phone. VoiceXML uses speech recognition and touch-tone keypad (DTMF) for input, and pre-recorded audio and text-to-speech synthesis (TTS) for output. It is based on the XML, and leverages the Web paradigm for application development and deployment.

In response to this reality, in the thesis Voice Interactive System is designed and developed for Bank using Interactive Voice Response (IVR) that is, interactive with the Web and can be used through a regular telephone. Specifically, the system provides 5 kinds of services, namely Check Account Balance, Transfer Money, Activate/Deactivate Credit Card and Bill Payment. The system lets a user to communicate with the application which is voice-enabled providing so-called speech user interface (SUI) or voice user interface (VUI) interactive with a Web server.

TABLE OF CONTENTS

INTRODUCTION

Speech Web, a combination of voice and data standards woven together with XML and other technologies has presented us with new competent choice. But integrating speech web with enterprise/web applications involves much more than simply adding a new voice interface to an existing internet/intranet based solution. Simply, because the users have different sets of needs and expectations for voice applications as opposed to visual ones. Users want voice transactions that are reachable at any time, easy to use, tailored specifically for their requirements, robust fault tolerant and prompt in responding. VoiceXML is one such technology available to satisfy these needs.

VoiceXML plays the role of the language of communication in voice applications. It is a W3C specification that bring together number of useful capabilities, such as: automatic speech recognition, touch-tone keypad recognition, text-to-speech playback, pre-recorded audio playback, telephony features, dialog features, event handling, scripting, platform features and performance features. It is a domain-specific language that defines dialogs between humans and machines in terms of audio files to be played, text to be spoken, speech to be recorded or recognized, and touch-tone input to be collected. VoiceXML can describe traditional voice response services as well as newer, mixed-initiative services. These capabilities can be employed to develop rich user experiences that allow callers to access information and transaction services through a telephony device.

In a common use of VoiceXML, a caller dials a telephone number that is routed to an IVR platform with a VoiceXML client. The platform translates the telephone number to a URL, and the client places an HTTP request to the specified URL. The Web server responds with a VoiceXML document that contains a dialog to be conducted with the caller. The client interprets the document to interact with the caller, plays prompts, collects input, and eventually submits the collected information to the URL designated by the document itself. The Web server processes the input and responds with a subsequent document to continue the session.

Aim of this thesis was to facilitate a bank customer with 'Voice' as Interactive Voice Response application with Automated Speech Recognition and the web technology. By keeping in mind VoiceXML features, I have designed and developed a Voice Interactive System for Bank Transaction. So that one can interact with his account from any where, any time. Just he has needed to call a provided number. It gives satisfaction and service to customer and also saves much of customer time.

Voice Interactive System for Bank Transaction is fully integrated with database, and all data is requesting through Active Server Pages. Application is giving a complete Voice User Interface environment. For the designed and development of application I have used Voice Application Development Life Cycle. In which I have passed this application through each phase. Facilities which I have provided are Customer can check his account balance, he can transfer money from one account to another, pay utility bills instantly, can get his account detail through fax, customer can activate or deactivate his credit card, he can get bank news, and customer can get update about the stock rates.

I have run and tested the application as client/server architecture using IBM WebSphere Voice Server 3.1, IBM WebSphere Voice Toolkit 3.1 and the Microsoft Windows 2000 operating system. Language used to develop Voice Interactive System for Bank Transaction is VoiceXML. The database is designed in Microsoft Access which integrated with VoiceXML using Active Server Pages and database connection made with ActiveX Data Objects. Application tested and simulation results were successful.

Chapter 1 defines the state of application problem. It gives introduction to interactive voice response systems and discuss the environment and design characteristics of voice user interface. Second, it tells the brief introduction of VoiceXML and after this the demand for voice applications.

In chapter 2, I have defined the architecture of Voice Activated System with brief discussion of VoiceXML gateway, VoIP gateway, VoiceXML Browser/Interpreter.

Third Chapter is about the Design and Development of Voice Interactive Bank System for bank customers. In this chapter I have given Voice Application Development Life

Cycle (VADLC), which passes through each development phase from analysis of problem domain to the deployment and maintenance phase.

Last chapter of this thesis is about the working prototype of Voice Interactive System for Bank Transaction using IBM WebSphere Voice Sever SDK. It shows the desktop simulation of voice application using the voice commands through microphone and DTMF as keypad tone.

CHAPTER 1

STATE OF APPLICATION PROBLEM

1.1 Overview

Everyone agrees on one thing: someday, in the not-too-distant future, we'll be talking with computers as easily as we do with humans, on the telephone, over the Web, or through a variety of embedded devices.

This chapter gives the introduction about voice applications, interactive voice response systems, visual user interface characteristics, voice extensible markup language and the demand of voice applications in market.

1.2 State of Art Understanding

As Internet access becomes a basic necessity, alternative ways to go on-line without a computer will be required by mobile professionals, the visually impaired, and people without access to computers.

Also, more companies will look for new ways to expand access to their web sites, using voice technology. The last decade brought an incredible convergence of communications and computers, with the World Wide Web arguably being the single most important development of our times. With the advent of easy access to the Internet, vast resources of information, both public and private, have already become readily available. As the pool of accessible information continues to grow, and better methods of selecting and filtering the desired portions are developed, the Internet is becoming an irresistible magnet for all variety of users. In fact, being connected is fast becoming a basic necessity.

The primary method of access today continues to be the computer, which has certain advantages as well as some limitations. Computers offer a visual Internet experience that is usually rich in content.

Some basic computer skills and knowledge are needed to access the Internet. But, computer-based access is proving insufficient for the professional on the move. When in the car or away from the office or computer, accessing the Web is difficult, if not impossible. And, an increasing number of people prefer an interface that allows them to hear and speak rather than see and click or type.

The computer-based Internet experience also does not meet the needs of another segment of the population – the visually impaired. Neither visual displays of information, nor keyboard-based interactions naturally meet their needs and this segment is often unable to benefit from all that the Information Age has to offer.
Some existing Internet users have also identified problems with the visual Internet experience.

Pages are increasingly full of graphics, advertisement banners, etc., which move, flash, and blink as they vie for attention. Some find this "information overload" annoying, and lament the delays it creates by severely taxing the available bandwidth.

While computers and their use are on the rise, they're not ubiquitous yet. A large segment of the population still doesn't have access to the Internet. In some cases, the barrier is cost, although the price of a computer has come down significantly in recent years. Other consumers have a basic distaste for complex technology, which prevents them from accessing Web-based information via a computer. A more natural, less cumbersome way to interface with the net would provide them an opportunity to experience the Internet as well.

1.2.1 Voice Applications

Voice applications are applications in which the input and/or output are through a spoken, rather than a Graphical User Interface. The application files can reside on the local system, an intranet, or the Internet. Users can access the deployed applications anytime, anywhere, from any telephony-capable device.

Until recently, the World Wide Web has relied exclusively on visual interfaces to deliver information and services to users via computers equipped with a monitor, keyboard, and

pointing device. In doing so, a huge potential customer base has been ignored: people who (due to time, location, and/or cost constraints) do not have access to a computer. Many of these people do, however, have access to a telephone. Providing "conversational access" (that is, spoken input and audio output over a telephone) to Web-based data will permit companies to reach this untapped market. Users benefit from the convenience of using the mobile Internet for self-service transactions, while companies enjoy the Web's relatively low transaction costs. And, unlike applications that rely on dual tone multi-frequency (DTMF) (telephone keypress) input, voice applications can be used in a hands-free or eyes-free environment, as well as by customers with rotary pulse telephone service or telephones in which the keypad is on the handset.

1.2.2 Interactive Voice Response

An automated telephone information system that speaks to the caller with a combination of fixed voice menus and real time data from databases. The caller responds by pressing digits on the telephone or speaking words or short phrases. Applications include bank-by-phone, flight-scheduling information and automated order entry and tracking.

IVR systems allow callers to get needed information 24 hours a day. They are also used as a front end to call centers in order to offload as many calls as possible to costly human agents. In such cases, it does not replace the agent, but helps to eliminate the need for them to constantly answer simple, repetitive questions.

Most IVR systems reside in Wintel PCs equipped with special ISA or PCI board-level products that contain DSP chips. These specialized processors connect to the telephone system, which actually switches the calls. IVR systems are also networked on LANs and WANs.

1.2.3 IVR Systems - Voice Applications

The promise of the information revolution has been achieved via the Internet-centric model of access to *any information, anytime, anywhere, and with any device.* Using a Web browser, without being tied down to proprietary user interfaces, one can connect to

cations and services like never before. However, Internet-centric computing has two drawbacks:

- **Lack of Mobility**, as one is still tied to the physical network to access information, and
- **Use of a Single Type of Device**, as computers are the only way to access devices.

present generation of phones and PDAs has the power and sophistication of puters. The interlinking and interfacing of various technologies in concert with these ces makes the vision of pervasive computing a reality. Telephones (wireless and ed) and PDAs are becoming the most pervasive of devices, as they're within reach for t consumers. Even though the network and browser technology in these devices are erent, it makes sense to extend Internet content to these devices because of their quity and convenience.

ce applications are an important part of the pervasive computing vision. In these lications, the input and/or output occur through a spoken medium, rather than a GUI. so many people have access to telephones and mobile devices, companies can use ce applications to reach this huge customer base of users who do not have access to a nputer due to time, location, and/or cost constraints. Users have the convenience of the bile Internet for self-service transactions, while companies benefit from the low- nsaction costs.

.4 Voice User Interface and Graphical User Interface

UIs may be a new concept to some who have been used to GUIs, character-based erfaces, etc. Major differences between VUIs and other types of UIs are:

- VUIs are invisible. VUIs exist only in the mind, and so must be designed to make the lightest possible demand on the user's memory and cognitive processes. This is the most critical difference between VUIs and GUIs.

- VUIs are single-mode interfaces where the medium of sound is the only input and output mode. Such single mode interfaces cannot deliver as much information or communicate with a user as effectively as other UIs.

- VUIs demand a simple task workflow with minimal branching and should maintain user interest and awareness while the application is being employed. The user task analysis drives the workflow of a VUI, whereas a GUI can be as elegant or complicated as the designer chooses it to be. A user generally trains himself to the workflow of the GUI.
- The user's environment plays a critical role in the functioning and performance of the voice application. VUIs are used in environments where there is competition for the user's attention and cognitive processing. Voice applications can be used while walking or driving, for instance. Consequently, they should be more fault-tolerant and have very good error handling features. In contrast, a computer is used to interact with a GUI and drive the workflow in the UI at the user's own pace. The user's environment usually does not impact the functioning of the GUI as the user drives the workflow.
- There are currently no standard VUI elements that correlate to GUI elements such as "Home", "Back", "Refresh", etc. VUIs have workarounds for these corresponding GUI features and associated application processing.

The Table 1.1 compares and contrasts GUI components and characteristics with their VUI counterparts.

Table 1.1 Comparison between GUI and VUI

GUI Component or Characteristic	VUI Counterpart
Screen layout, color, graphics, and style.	Recorded audio, TTS voices, voice gender, and tones.
Pop-ups or windows to indicate error messages or error recovery.	Tones, TTS, or recorded audio that indicate error messages and recovery
Help links or online context sensitive help.	Tiered help messages as voice messages.
Links to other Web pages.	Programmed functions such as mixed initiative grammars that let users

	jump to other applications or modules within a voice application.
Form input, selection lists, and radio buttons.	VoiceXML forms with fields and variables, acquiring field input for the form elements.
In-progress indicators of other user feedback.	Audio hourglass tones, music, or voice messages to communicate system functioning.

.2.5 VUI User Characteristics

Before a voice application is conceptualized and developed, it is important to study and document the characteristics of the user base that is the target for the voice application. The target user base for any voice application is more or less the same as for PC/GUI users; however, there are some characteristics of the VUI users that are worth mentioning.

The first point to note is that people typically are not going to be using a PC or a computer to access the voice application. The telephone or a cellular phone is going to be the mode of access for the majority of users. Some platforms support computer-based telephony using VoIP and SIP, but such clients are few. Although most users of computer applications are voice application. Even then, it is important to educate users about voice applications, what they can and can't do, and their salient features. For example, users should not assume that their mental model of human-to-human conversational models cannot be carried over to a voice application. Users of previous-generation IVR systems should not let their experiences with those systems spill over to today's newer and better voice applications. Internet users that have been accustomed to browsers would find that they don't have the concepts of Back, Refresh, hyperlinks, and field-based forms. All said, it has to be stressed again that users new to voice application should be made aware of the characteristics of voice applications and their usefulness to the user's expectations of task fulfillment.

1.2.6 Environment

The environment plays a critical role in the design of voice applications and the VUI, more so than other software applications. People are going to be using both wireline and wireless telephone devices to access voice applications. They may be talking into their phones while doing other tasks, such as walking or driving, and may not fully devote their attention and concentration to using the voice application. The environment may have extraneous background noise which may interfere with the speech or recognition function of the voice application. People using voice applications are often in a hurry and want task-specific applications that gets the job done quickly. This is more so with "expert" users, such as those familiar with the voice application, who want to get to the point in the application where they can get their task done quickly. Using telephones to access voice applications makes it difficult or impossible for any significant amount of data entry using a keypad. Voice applications have a single input/output mode using only the medium of sound. The result is that they cannot convey as much information as other graphical applications. Also, users cannot remember everything that is presented by the voice application when sound is the only delivery system. All these factors, along with the above user characteristics, make for challenges unique to voice application development.

1.2.7 Design Characteristics

A well designed VUI should be tightly focused on the activities or tasks that the users want to perform. This implies that the design of the VUI is intimately connected with user and task analysis.

A good VUI design should exceed the user's expectation and generate customer satisfaction. The voice used for prompts and dialogs should engender trust and likeability. The dialog, prompts, and recorded announcements should be designed to align with the culture of the target user base. The voice application should have system timers and application pauses that can be tuned so that the application operates at a speed best suited to the task, environment, and typical user.

is important that the VUI has a clear and uncomplicated application workflow so it pears to be "user-friendly." To achieve this goal, the VUI should be designed in cordance with best practices for user interface design and usability. The VUI should ve a well-designed workflow with no non-standard features and with terminology most rs would know.

good VUI design should shield the constraints of the voice application platform from users. A voice application has constraints associated with speech recognition, tform capabilities, and network latencies. The VUI design should lessen the straint's impact to the users' perception of the application by employing prompt design l DTMF input. This is to overcome ASR limitations, using "audio hourglass" to icate system processing, using system timers and messages to lessen negative ceptions associated with external system latency or failures.

.8 Layers of Voice Application

ere are three layers of technology required to implement a voice application: the phony layer, the voice platform layer and the integration layer. The telephony layer wers incoming calls, performs call management, and connects the caller with a ning instance of an application. This involves the installation and management of rier connections, switches, call distributors, and the software necessary to keep them and running. Features include simple connection controls (call transfer, add 3rd party, disconnect) and telephony information like Automatic Number Identification (ANI) Dialed Number Information Service (DNIS).

voice platform layer provides the environment in which the voice application is run. responsible for providing the following functionality:

- Speech recognition. Interprets callers' spoken input.
- Streaming audio. Plays audio files for prompting callers and providing information.
- Text-to-speech. Automatically generates speech when pre-recorded audio isn't available.

- Voice application interpreter. Coordinates playing of prompts, invocation of the speech recognizer, and implementing application logic according to callers' responses.

The integration layer links the voice application with computing infrastructure external to the application. This includes resources such as databases, call-center management systems, transaction processing systems, and legacy applications. The specific technologies to do this vary based the systems to be integrated.

1.3 Voice eXtensible Markup Language

VoiceXML is the emerging standard for voice-based markup on the Web. It was released as a 0.9 draft in August, 1999, with formal release and acceptance for submission by the W3C in March, 2000.

A joint effort by AT&T®, IBM®, Lucent™ and Motorola® has created an XML vocabulary that marks up information for speech synthesizers, which enable computers to speak to users. This technology, called VoiceXML, has tremendous implications for people with visual impairments and for the illiterate. VoiceXML-enabled applications read content to the user, and understand words spoken into a telephone device through speech recognition technology.

VoiceXML is defined and promoted by an industry forum, the *VoiceXML Forum*, founded by AT&T®, IBM®, Lucent™ and Motorola® and supported by around 500 member companies. VoiceXML was designed to create audio dialogs that feature text-to-speech, digitized as well as prerecorded audio, recognition of both spoken and DTMF key input, recording of spoken input, telephony, and mixed-initiative conversations. Its goal is to provide voice access and interactive voice response (e.g. by telephone, PDA, or desktop) to Web-based content and applications.

A VoiceXML interpreter and VoiceXML browser process VoiceXML. When a VoiceXML document is loaded, a voice server sends a message to the VoiceXML browser and begins conversation between the user and the computer.

Users call voice application by telephone. They listen to spoken instructions and questions instead of viewing a screen display; they provide input using the spoken word and the touchtone keypad instead of entering information with a keyboard or mouse.

The VoiceXML language enables Web developers to use a familiar markup style and Web server-side logic to deliver applications over telephone lines. The resulting VoiceXML applications can interact with your existing back-end business data and logic. Using VoiceXML, application developers can create Web-based voice applications that users can access by telephone or other pervasive devices. VoiceXML supports dialogs that feature:

- Recognition of spoken input
- DTMF input
- Recording of spoken input
- Synthesized speech output ("text-to-speech")
- Pre-recorded digitized audio output
- Dialog flow control
- Scoping of input
- Automatic Number Identification (ANI)
- Dialed Number Identification Service (DNIS)
- Call transfer

VoiceXML brings the power of Web development and content delivery to voice response applications, and frees the authors of such applications from low-level programming and resource management. It enables integration of voice services with data services using the familiar client-server paradigm, and it gives users the power to seamlessly transition between applications.

1.3.1 Advantages of VoiceXML

VoiceXML provides several important capabilities:

- VoiceXML is a markup language that makes building voice applications easier and also reduces the amount of speech expertise that developers must have.

- VoiceXML applications can use the same existing back-end business logic as their visual counterparts, enabling voice solutions to be introduced to new markets quickly. Customers can benefit from a consistency of experience between voice and visual applications.
- VoiceXML implements a client/server paradigm, where a Web server provides VoiceXML documents that contain dialogs to be interpreted and presented to a user. The user's responses are submitted to the Web server, which responds by providing additional VoiceXML documents, as appropriate. VoiceXML allows to request documents and submit data to server scripts using Universal Resource Identifiers (URIs). VoiceXML documents can be static, or they can be dynamically generated by CGI scripts, Java Beans, ASPs, JSPs, Java servlets, or other server-side logic.
- Unlike a proprietary Interactive Voice Response (IVR) system, VoiceXML provides an open application development environment that generates portable applications. This makes VoiceXML a cost-effective alternative for providing voice access services.
- Most installed IVR systems today accept input from the telephone keypad only. In contrast, VoiceXML is designed predominantly to accept spoken input, but it can also accept DTMF input, if desired. As a result, VoiceXML helps speed up customer interactions by providing a more natural interface that replaces the traditional, hierarchical IVR menu tree with a streamlined dialog using a flattened command structure.
- VoiceXML directly supports networked and Web-based applications, meaning that a user at one location can access information or an application provided by a server at another geographically or organizationally distant location. This capitalizes on the connectivity and commerce potential of the World Wide Web.
- Using a single VoiceXML browser to interpret streams of markup language originating from multiple locations provides the user with a seamless conversational experience across independent applications. For example, a voice portal application might allow a user to temporarily suspend an airline purchase

transaction to interact with a banking application on a different server to check an account balance.

- VoiceXML supports local processing and validation of user input.
- VoiceXML supports playback of prerecorded audio files.
- VoiceXML supports recording of user input. The resulting audio can be played back locally or uploaded to the server for storage, processing, or playback at a later time.
- VoiceXML defines a set of events corresponding to such activities as a user request for help, the failure of a user to respond within a timeout period, and an unrecognized user response. A VoiceXML application can provide catch elements that respond appropriately to a given event for a particular context.
- VoiceXML supports subdialogs, which are roughly the equivalent of function or method calls. Subdialogs can be used to provide a disambiguation or confirmation dialog, and to create reusable dialog components.

1.3.2 VoiceXML Structure

VoiceXML is an XML-based application, meaning that it is defined as a set of XML tags.

Dialog Structure

VoiceXML documents are composed primarily of top-level elements called dialogs. There are two types of dialogs defined in the language: <form> and <menu>.

Forms and Form Items

Forms allow the user to provide voice or DTMF input by responding to one or more <field> elements.

Fields

Each field can contain one or more <prompt> elements that guide the user to provide the desired input. The count attribute can use to vary the prompt text based on the number of times that the prompt has been played.

Fields can also specify a type attribute or a <grammar> or <dtmf> element to define the valid input values for the field, and any <catch> elements necessary to process the events

that might occur. Fields may also contain <filled> elements, which specify code to execute when a value is assigned to a field. <clear> element can use to reset one or more form items.

Subdialogs

Another type of form item is the <subdialog> element, which creates a separate execution context to gather information and return it to the form. If form requires prompts or computation that do not involve user input (for example, welcome information), you can use the <block> element. This element is also a container for the <submit> element, which specifies the next URI to visit after the user has completed all the fields in the form. You can also jump to another form item in the current form, another dialog in the current document, or another document using the <goto> element.

There are two types of form dialogs:

- **Machine-Directed Forms** - traditional forms where each field or other form item is executed once and in a sequential order, as directed by the system.
- **Mixed-Initiative Forms** - more robust forms in which the system or the user can direct the dialog. When coding mixed-initiative forms, you can use form-level grammars (<form scope="dialog">) to allow the user to fill in multiple fields from a single utterance, or document-level grammars (<formscope="document">) to allow the form's grammars to be active in any dialog in the same VoiceXML document; if the document is the application root document, then the form's grammars are active in any dialog in any document within the application. You can use the <initial> element to prompt for form-wide information in a mixed-initiative dialog, before the user is prompted on a field-by-field basis.

Menus

A menu is essentially a simplified form with a single field. Menus present the user with a list of choices, and associate with each choice a URI identifying a VoiceXML page or element to visit if the user selects that choice. The grammar for a menu is constructed dynamically from the menu entries, which you specify using the <choice> element or the

shortcut <enumerate/> construction; you can use the <menu> element's <scope> attribute to control the scope of the grammar.

The <enumerate> element instructs the VoiceXML browser to speak the text of each menu <choice> element when presenting the list of available selections to the user. If you want more control over the exact wording of the prompts (such as the ability to add words between menu items or to hide active entries in your menu), simply leave off the <enumerate> tag. Menus can accept voice and/or DTMF input; you can specify the acceptable type(s) of input using the construct <property name="inputmodes" value=" mode">, where modeis "dtmf", "voice", or "dtmf voice" (the default). If desired, you can implicitly assign DTMF key sequences to menu choices based on their position in the list of choices by using the construct <menu dtmf="true">.

Flow Control

When the VoiceXML browser starts, it uses the URI you specify to request an initial VoiceXML document. Based on the interaction between the application and the user, the VoiceXML browser may jump to another dialog in the same VoiceXML document, or fetch and process a new VoiceXML document. VoiceXML provides a number of ways for managing flow control. For example, the <link> element specifies a control common to all dialogs in the link's scope.

Recorded Audio

VoiceXML supports the use of recorded audio files for output. The VoiceXML browser plays an audio file when the corresponding URI (<audio src="URI of the audio prompt">) is encountered in a VoiceXML document.

Document Fetching and Caching

The VoiceXML browser uses caching to improve performance when fetching documents and other resources (such as audio files, grammars, and scripts).

1.3.3 Voice Browsing from Telephone Devices

Here's how a voice-enabled application works. Telephone calls come in on an ordinary telephone line to a connection environment. The primary purpose of the connection

environment is to transfer the telephone voice data to Voice Server. The connection environment can be IBM WebSphere Voice Response, Cisco, or Intel Dialogic.

Next it is received by the Voice Server. The server runs the voice recognition and text-to-speech software, multiple instances of the voice browser, and call management software that stacks and controls the incoming calls as they go through the system.

The recognition engine analyzes the audio stream and converts it to digitized text. The digitized text is then sent to the voice browser, which creates HTTP requests as necessary, and accesses the target information over the network. This is analogous to a visual browser, except that speech-enabled requests look for pages written in VoiceXML code. The application resides on a Web application server such as IBM's WebSphere Application Server, which contains pages of VoiceXML code. Data is accessed from various databases as needed. As each HTTP request is received, information is returned to the requesting server in the form of VoiceXML pages, which the IBM Text-To-Speech engine reads back to the caller.

1.4 The Demand for Voice Applications

Voice applications are growing in certain key markets, in particular, banking, finance, securities, and in the communications industry, especially customer call centers.

1.4.1 Electronic Relationship Management

For many years, the primary point of customer interaction, or Customer Relationship Management (CRM), was the traditional, telephone-based call center. In today's business environment, the Internet has rapidly assumed a role as an alternative to the call center. This shift has not replaced the call center, but instead has levied a whole new set of requirements on the enterprise, as CRM moves to eRM, or electronic Relationship Management. Voice technologies have begun to change the way self-service systems work. No longer are these systems limited by the rigidly-structured, keypad-driven dialogs that characterize traditional interactive voice response (IVR) systems, or for that matter, most Web sites. Voice recognition frees customers from these constraints,

enabling them to interact with an automated, self-service environment in a way that is very similar to a conversation with a human agent.

1.4.2 Travel Agency

Callers dial into the 800 number for a large travel agency call center. Around 20% of those callers are calling to check the status of their reservation. Rather than connecting callers to expensive call center agents or to a restrictive touch tone application, the agency ask the callers to say their reservation number. The system then queries the agency's existing Web-based reservation database and receives XML instructions to inform the user of the reservation status

Benefit: Direct reduction in call center costs and an increase in customer satisfaction.

1.4.3 Outbound Voice Surveys

A large fast food restaurant chain notices a decline in sales. Rather than launching a slow and costly outbound survey with call center agents, it uses an automated Voice Survey application that calls customers to ask opinions on new foods the chain plans to sell. When called, the restaurant's customers can respond conversationally, saying things such as, "That sounds great" or "I wouldn't buy that". In less than 24 hours, the restaurant knows which of the new products its customers desire most.

Benefit: Automated Outbound Survey, No expensive call center costs, rapid deployment, easy integration.

1.4.4 Inbound Voice Surveys

A large fast food restaurant chain has customer response cards at the dinning tables with free food incentives on them. The customer must fill it out and mail it into the main office. The office personnel must enter the information into their database and then mail the coupons back to the customer. A VoiceXML Survey would allow the user to call into the system, enter the unique number on the customer response card and answer the survey questions. The system would then queue the coupons to be mailed to the customer address obtained from the Caller ID (CID). This same survey could also be completed on the Internet.

Benefit: Automated Inbound Survey, Direct reduction in reservation agent costs and an increase in customer satisfaction

1.4.5 Service and Dispatch

A high volume service maintenance company needs an efficient way to track real time service statuses, inventory, and completion times. They need to be able to provide real time results and reporting to their clients. Rather than having expensive wireless laptops in the large fleet of service vehicles, the service technician can use their Cellular phones to interact with the database via a VoiceXML application. The technician can simply speak their name and job number and get the status of the job, complete the job, check for any new jobs, order inventory, etc. The company's clients can access the service results via the telephone or the company's web site.

Benefit: No expensive wireless laptops, real time data, no paper trail, companies client's are satisfied.

1.4.6 Online Vacation Registration

A vacation resort chain issues vacation vouchers to customers. When the customer is ready to travel they must fill out the voucher with their mailing address information along with the dates and location of their vacation and the mail it to the resort's reservation center. The resort reservation agent would enter the customer's information and mail them a confirmation letter. Using a Voice XML application the Company now allows the customer to register via the Internet or over the telephone. The automated telephony IVR system uses the caller id information to obtain the customer's mailing address. It allows them to speak the destination and travel dates and verifies the room availability. A confirmation number is given to them, followed with a confirmation letter in the mail or email.

Benefit: Direct reduction in reservation agent costs and an increase in customer satisfaction.

1.5 Market Forecasts

The voice market is expected to grow. Often cited estimate by researcher International Data Corp. is that by 2004, some 600 million people worldwide will hook up to the Net

via PCs, but 1.4 billion will connect through cell phones and another 1.4 billion will get on through wired phones. Kelsey Group estimates that revenues from voice commerce are over $30 billion by 2005 and from speech portals about $4.6 billion by year 2003. Comverse, a voice technology provider, reports that 52 percent of users claim they want voice recognition-based services within the next six months.

According to Cahner's In-Stat, the market for voice portals and services could generate $1.6 billion annually by 2005, and reach $6 billion by 2006 in the U.S. alone. The overall market for network based speech products will reach $12 billion by 2006. The Kelsey Group projects that worldwide revenue from voice applications will reach $41 billion by 2005.

These forecasts will help to drive the development of specific voice applications. Allied Business Intelligence predicts that the market for voice recognition applications will grow from $2.3 billion this year to $50 billion by 2005. Cahner's In-Stat projects that sales of speech engines—the server software that runs voice applications will grow to $2.7 billion by 2005.

The number of users accessing voice services also will grow exponentially, particularly among mobile users. According to Allied Business Intelligence, by 2006 there will be more than 291 million voice portal users worldwide. Of this number, 25 percent will be infrequent users, 40 percent will be frequent users and 35 percent will be habitual users of voice portal services.

Many voice applications are currently operating successfully, fueling expectations for rapid future growth. Research also indicates that users may be ready for the new services. According to a recent study by the Yankee Group, about half of the 120 million wireless subscribers in the United States would like to use voice-enabled services, and more than 40 million are also interested in voice enabled personal digital assistants and other handhelds. Research conducted by Nuance Communications and Evans Research found that 87 percent of the users of speech systems are satisfied with their interactions, and often prefer the automated systems to live agents.

VoiceXML leverages the existing skills of Web developers, thus taking advantage of a large pool of developers. And because it's an open standard, applications can be more portable between platforms. The end result is the expansion of voice applications in the marketplace.

1.6 Deployed Voice Services

Interactive Voice Response systems are being deployed efficiently in the area of bank services, voice portals, traveler information, updates about weather, stock exchange. For detail check Appendix C listing.

1.7 Summary

Unification of voice power and universality of telephony network with the matured web technologies has opened new vistas of voice-based solutions and an extensive new market for commerce. VoiceXML 2.0 empowers the developers with incisive tools for defining and fine-tuning the behavior of the voice interface and for grammar building. VoiceXML makes it possible for enterprises to replace legacy, proprietary IVR platforms with a unified architecture for delivering voice web services from any telephony device.

Presently, the voice world is in its incipient stage. Development of more and more voice based enterprise applications and extensive technological innovation will lead to evolution of the voice technologies to incorporate many unanticipated and application specific features. And eventually it would uncover and establish some robust design patterns for the voice solutions to capitalize on the very nature of voice interface.

CHAPTER 2

ARCHITECTURE OF VOICE ACTIVATED SYSTEM

2.1 Overview

The end user in voice system uses a standard telephone, such as a regular home phone or a cell phone. The phone doesn't have to be WAP-enabled or have any type of screen. The user simply speaks into it and receives audio back through the earpiece. Here, this is accomplished with the Voice Server and a Voice Over IP (VOIP) gateway. The user makes a phone call and the VOIP gateway turns the audio coming over the telephone into IP packets, which travel over the network to the voice server. The voice server runs instances of the voice browser.

When the VOIP gateway accepts the incoming telephone call, it finds a free instance of a voice browser to connect to. That voice browser becomes the virtual person with whom the user talks. The Voice browser plays audio prompts and synthesizes speech as directed by the VoiceXML markup. The voice browser accepts input from the user and determines what the user has said. The Voice Browser executes the application as directed by the VoiceXML.

Outside this infrastructure, the Web server receives an HTTP request from the client (the Voice Browser running inside the voice server) and determines how to get markup (VoiceXML) back to the client. This markup could be static VoiceXML files, dynamic markup generation using CGI, Perl, ASP, JSP servlets, and so on.

In this chapter I have defined the complete architecture of Voice Interactive System from telephone device to VoiceXML gateway and WebServer through its components such as VoiceXML platform, VoiceXML gateway, VoIP Gateway, Automatic Speech Recognition, VoiceXML Browser/Interpreter and VoiceXML Application Server.

2.1.1 Telephony Network

This can be a PSTN (Public Switched Telephony Network), a regular analog line or lines coming through a PBX (Private Board Exchange) system, ISDN (Integrated Services

Digital Network) lines or VoIP (Voice over IP) network. The telephony network is connected to the VoiceXML gateway. The telephones can be regular phones or IP (Internet Protocol) phones if connected to the VoIP network.

2.1.2 VoiceXML Platform

A VoiceXML platform typically consists of a gateway and an application server. The gateway almost always resides on the same machine as the voice hardware and the application server interfaces with any data and control sources, and houses the programming logic. In most cases, all programming takes place on the application server side. We can treat the VoiceXML gateway as a black box that interfaces with the phone network, the caller, and the caller's telephone.

2.1.3 Voice Application Connection

Access to the voice application is through a wireline or wireless telephone device. Users dial a telephone number associated with the voice application. The VoIP Gateway receives the incoming call over a voice interface card (VIC). The telephony interface establishes the voice connection between the telephone switch and the VoIP Gateway and provides echo cancellation, DTMF tone detection, and audio prompt playback/record.

2.1.4 VoiceXML Gateway

Interpretation of the script and the interaction with the user is controlled by the VoiceXML gateway. Gateways are special collections of hardware and software which form the core of VoiceXML technology. Essentially they provide the presentation services component of VoiceXML, analogous to the web browser in conventional HTTP service.

Incoming calls are answered by the Telephony Services and Signal Processing component. Gateway systems are provisioned in a manner similar to IVR systems and can in fact be located "downstream" of a PBX or automatic call director. This architecture allows callers to request transfer to a live operator if they encounter problems.

2.1.5 VoIP Gateway

The VoIP gateway serves as an interface between the voice application gateway and external telephony networks, such as PBXs (Public Branch Exchange), PSTNs (public switched telephone network), and voice traffic over VoIP. The VoIP gateway translates voice data received from the telephony network into IP packets and maps the incoming call number to an IP address and port number of the VoiceXML browser.

2.1.6 Text-To-Speech

The TTS engine synthesizes human speech from a text stream by combining pronounceable phonemes into human-discernible words and phrases. TTS engines can use speech synthesis techniques such as articulator synthesis, formant synthesis, and concatenative synthesis to generate a very human and natural sounding speech instead of robotic-sounding speech. They can be implemented through firmware or on a host-based solution. These engines also support barge-in.

2.1.7 Automatic Speech Recognition

Automated speech recognition (ASR) is a technology that allows users of information systems to speak entries rather than punch numbers on a keypad. ASR is used primarily to provide information and to forward telephone calls.

Speech recognition works in a manner similar to human speech. Specifically, it works like the ear and human brain; the ear takes in the sounds in the form of vibrations, and the brain decodes the signals and determines meaning. In a basic sense, this is how speech recognition works.

At the core of all modern high-performance speech recognition systems are complex statistical models that are able to characterize the properties of the sounds of the language to be recognized. This is an especially challenging problem because of the large degree of variability introduced along the way. Things such as speaker differences (male Vs female, tone of voice etc), accents, speaking rate, background noise, telephone handset variability, transmission channel differences, etc. all affect the properties of the signals that reach the speech recognition system. Because of this variability, there is no simple way to model what a "b" sound or a "d" sound should be.

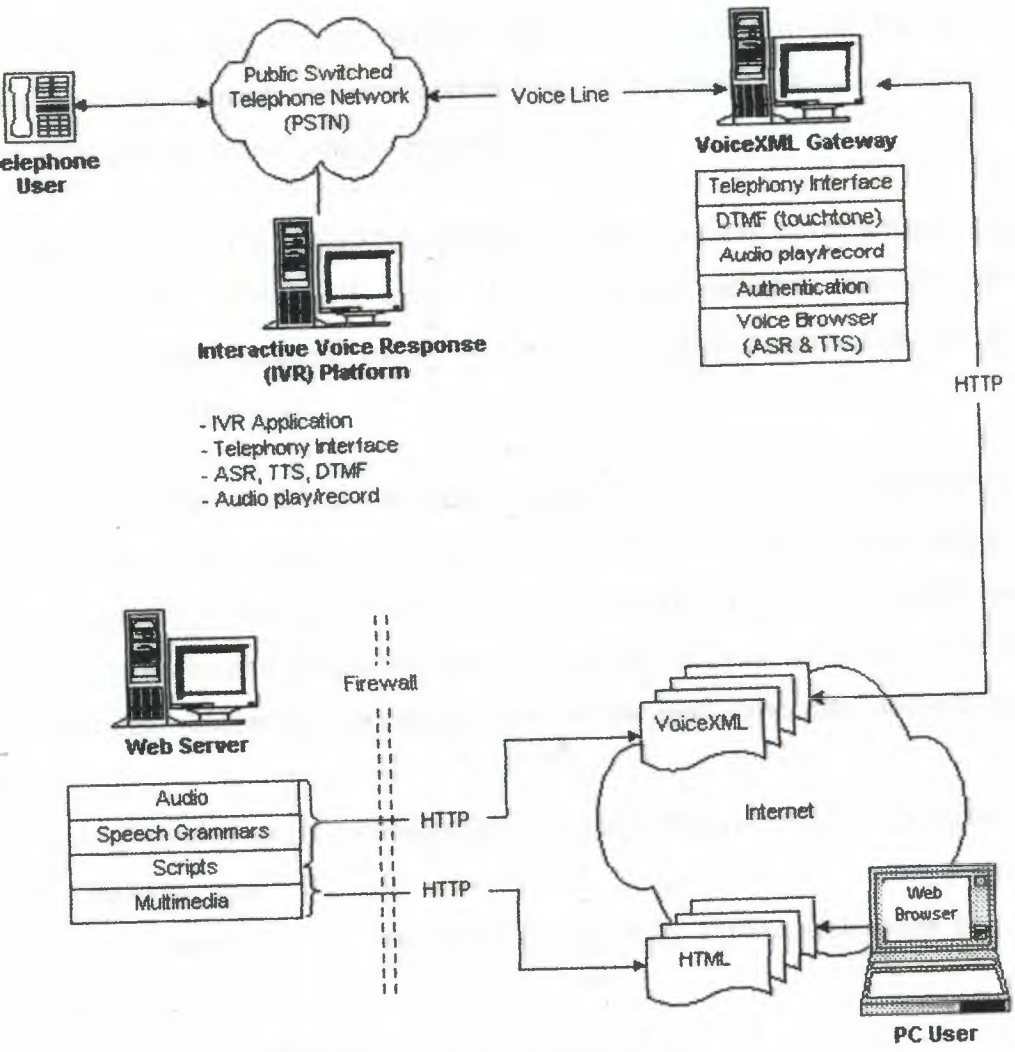


Figure 2.1 Architecture of voice-based interactive system

he job of the statistical models, which are trained on large amounts of real speech, to into account all of the factors described above. These models judge the probability a given segment of speech is a particular phoneme (the basic unit of the sounds of a uage).

3 Browser/Interpreter

ser connects with voice application by dialing the appropriate phone number. The ceXML interpreter answers the call and starts executing your VoiceXML document. VoiceXML browser/interpreter is the heart of the voice application gateway. The bined browser and interpreter are referred to as the VoiceXML engine. This is the ware that fetches VoiceXML documents using HTTP and renders the markup as a

Writing powerful documents is easier when use SpeechObjects. These are pieces of software that are pre-written, tested, and packaged in a form that is easy for a VoiceXML document to use. SpeechObjects conduct dialogs for common functions such as accepting credit card numbers, times and dates, and dollar amounts.

2.1.9 Audio

The audio resource is responsible for the playing of audio files. Most voice application platforms support raw audio data without headers and .wav format files with an 8 KHz sample rate and 8-bit mu-law encoded data. They currently do not support the GSM encoding for audio, which is the dominant format for cellular phones worldwide.

2.1.10 TCP/IP

The TCP/IP resource provides the backbone connection between the voice application gateway and the Web/applications server. It is also responsible for the packet assembly/disassembly between the voice application gateway and other machines on the network.

2.1.11 Application Server

Finally, there is the Web/application server where the application pages are found, consisting of static VoiceXML files and programs that generate dynamic VoiceXML pages. The server can interface with a data store such as a DBMS to retrieve/update enterprise data.

2.1.12 Voice Recognition

The voice recognition process is performed by a core software component known as the *voice recognition engine*, which translates spoken words into text in a format that an application can understand. Of course the application could simply leave the words as text, as in a dictation application. But usually the application interprets the text as an instruction to do something, as in a *command and control* application, where the caller might say *"sell all shares"* and a transaction takes place.

Input comes into the voice recognition engine from a microphone as an audio stream, over the Internet, or from a telephone. The recognition engine may have to adapt to low

volume or background noise, matching the input against an *acoustic model*. It then uses data, statistics, and software algorithms to convert the incoming audio signal into a data format that is suitable for further analysis. Once the audio data is ready, the engine searches for the best match, using the words and phrases it already has (active grammars and vocabularies), returning a text string.

2.2 Summary

The architectural design of VoiceXML is very flexible and integrates seamlessly with the existing web technologies. The general model of voice application architecture comprises a VoiceXML gateway or server that sits between the wired / wireless phone network and the HTTP server and is primarily responsible for actual translation process between voice and VoiceXML contents.

CHAPTER 3

DESIGN AND DEVELOPMENT OF VOICE INTERACTIVE BANK SYSTEM

Overview

VUI is any part of a voice application that the user has to interact with to perform the ks the application supports. On a high level, an application developed using iceXML represents a voice application to the user. At a basic level, a VUI:

- Provides users with a mental model of how the application works and what functionality it provides
- Collects user input in the form of speech or DTMF (dual tone multi frequency) sounds generated by pressing buttons on a telephone keypad
- Outputs synthesized speech or prerecorded audio
- Supports users in task completion
- Supports recovery from user or system errors

cause they can't communicate as much information as high-bandwidth multimedia Is, VUIs have limited capabilities. They are also "invisible," as they exist only in the rs' minds. VUIs are also single-mode interfaces in which the only input medium is nd, which is also the output medium. Such single mode interfaces cannot mmunicate as effectively as multimode GUI interfaces. Thus, voice applications are suited for situations in which:

- The user's environment is very noisy
- Large amounts of data need to be presented for review or comparison
- The output content is large and visually complex
- It is easier to accomplish the task using another means of device input, such as keyboard or mouse

wever, voice applications are best suited for situations in which:

- Voice is the most convenient mode of device input (for example, when driving a car)
- Users save time and money and improve productivity through self-service
- Corporations or call centers can minimize customer service expenses
- Users have physical disabilities

e components of a VUI that support user input include:

- Spoken utterances that match a word or phrase included in an active grammar
- A DTMF key press that provides a single input to the application
- Other speech directed to the VUI that can be recorded for future playback by the same or another application

components of a VUI that support user output include:

- A prompt that gives an auditory message to users, cueing them to provide input to the application
- Playback of digitized recorded audio within prompts and elsewhere
- TTS (text to speech) to synthesize speech content from text
- Menus within prompts that give a list of choices to the user
- Forms that function behind the VUI to collect information for the application

Voice Application Development Life Cycle

uccessful voice application development project is the result of carefully executing nst all states of a software development cycle. The primary phases of a voice ication project are the following:

- Requirements
- Architecture
- Prototype
- Design
- Testing
- Validation and tuning
- Deployment and maintenance

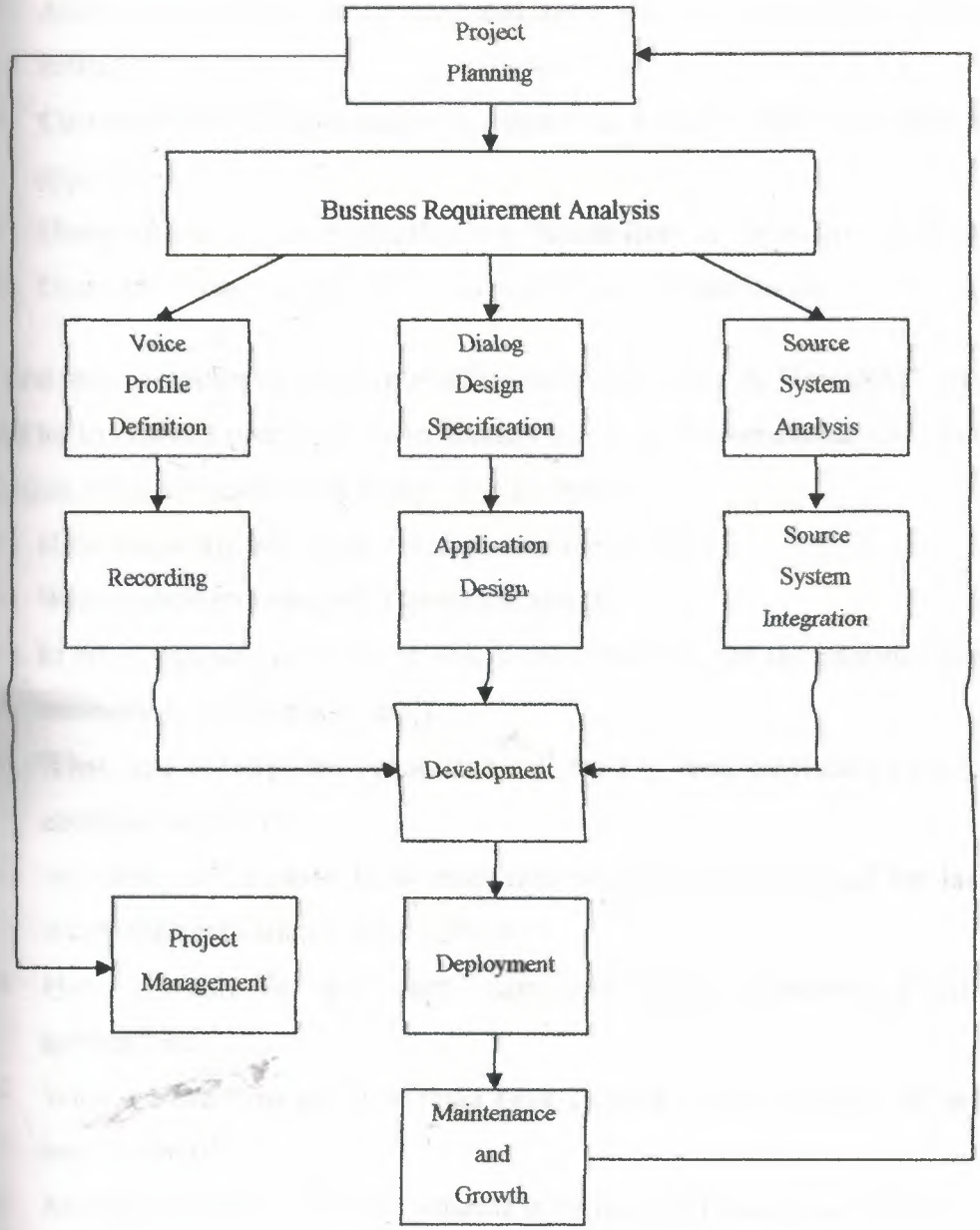


Figure 3.1 Voice Application Development Life Cycle

3.2.1 Requirement Phase

3.2.1.1 Analyzing Bank Customers

There are following characteristics which are motivated customers to use the voice interactive bank transaction system

- Saves them time or money
- It is available 24 hours a day
- Provides access to features not available through other means

- Allows them remain anonymous and avoid discussing sensitive subjects with a human
- Customers will not have access to a computer keyboard when they want to use the application
- Users want to use the application in a "hands-free" or "eyes-free" environment
- Users are visually impaired or have limited use of their hands

The first step in designing voice interactive system for bank is, VoiceXML applications should be to conduct user analysis to identify any user characteristics and requirements that might influence application design. For example:

- How frequently will bank customers use the system?
- What is their motivation for using the system?
- In what type of environment will bank customers use the system (quiet office, outdoors, noisy shopping mall)?
- What type of telephone connection will most of bank customers have (land-line, cordless, cellular)?
- Are many of intended bank customers non-native speakers of the language in which the application will be written?
- How comfortable are bank customers with automated ("self-service") applications?
- What are the most common tasks bank customers will perform? What tasks are less common?
- Are bank customers familiar with the tasks they will need to complete?
- Will bank customers be able to perform these tasks by other means (in person, using a visual Web interface, by calling a customer service representative, etc.)?
- What words and phrases do bank customers typically use to describe the tasks and items in voice interactive system for bank application?

Table 3.1 illustrates some of the ways in which customer characteristics can impact voice application design.

Table 3.1 Influence of customer profile on Application design

Bank Customer Characteristics	**Application**
Frequency of use	Daily – Users will become experienced and may want to be able to cut through the interface to complete common tasks more quickly.
Access hours	24 hours a day, 7 days a week – Users may relay on this "self service" application and therefore be more motivated to use it.
Expertise in application domain	Expert – Users know what to do and use a common terminology (check balance, money transfer, bill payment); prompts should use this terminology.

3.2.1.2 Analyzing Bank Customer Tasks

After identified who customers are, the next step is to determine what tasks voice application should support, and assign them to the application or customer based on the strengths of each. For example, customer tasks in a bank could ask such types of facilities and features with voice application.

- What is bank history?
- What are current news at bank?
- What products bank is offering?
- What is account balance?
- Customer wants to transfer money to another account
- Customer wants to pay a utility bill
- Customer wants to know about trade stocks
- Customer wants contact and mailing address
- Customer wants to reach bank

Voice application might allocate the functions as follows:

- Have the application locate, sort, and store account information, and make any routine decisions. For example, if the customer attempts to transfer more money than is available, the application plays a warning message.
- Have the customer confirm transactions and make any non-routine or elective decisions. For example, ask the customer to confirm the amount of money and account number before the application submits a form that initiates a monetary transfer.

3.2.1.3 Defining Information Flow

Next, you will want to outline an information flow that maps the interaction between voice interactive system and the bank customer. For example:

- What questions do voice application need to ask the bank customer?
- When the customer answers, how should the application respond?

Voice application interaction should have a logical progression that takes into account typical responses, unusual responses, and any error conditions that might occur. For example, in Voice interactive system for bank, application can ask

- What is customer account number?
- What is customer pin number?
- To which account customer wants to transfer a money?
- For which bill id customer wants to pay?

3.2.1.4 Identifying What a Caller Can Say

One of the first steps in the voice application development process is deciding what the callers will say. Are there different parts of the application where different things will be said? Are there things callers will always want to say, regardless of where they are in the application (for example, "*help*" or "*exit*")? Do you want to support synonyms - more than one way of saying a command? Do you want to support a more natural way of saying something versus providing specific command sequences (for example, "*Can I get my account balance please*" versus "*account balance*")? These are just some of the considerations which should make when identifying what callers will be able to say to an application. Once decided what words and phrases the caller will be allowed to say at

each point in voice application, define the grammar(s) for these words and phrases. You can and probably will have multiple grammars for your application. This helps group and combine things the user will say to voice application. Some typical questions caller can ask,

- Please tell me about bank
- What is current news?
- What products bank is offering?
- How much money left in my account?
- I want to transfer money
- How can I pay my bills?
- Can I get my account detail through fax?
- I want to change my account pin number
- I want to activate/deactivate credit card
- Tell me about trade stocks
- I want to talk to customer care center
- Where is bank ATM?
- What is the mailing address of bank?
- How can I reach bank?

3.2.1.5 Identifying What the Application Will "Say"

A voice application is an audio-only interface. Now need to identify what voice application is going to say at any given point in the dialog with the caller. This is called defining voice application prompts.

Prompts can be prerecorded and played to the user during the phone call. They can also be played using text-to-speech. Text-to-speech is a critical component of your application when you need to present "unbounded" information to the.

3.2.1.6 Creating the Initial Scripts

After defining the information flow, be ready to create an initial draft of the scripts for the dialog between the application and the customer. The script should include all of the text that will be spoken by the application, as well as expected valid bank customer utterances.

.2.1.6.1 Script Bank Information

'aller: Please tell me about bank

ystem: First opened in 1961 in Pakistan, today Citibank Pakistan serves the financial eeds of the entire range of customers from individuals to the world's largest ıultinational companies. Across consumer banking products, it enjoys the trust and ›yalty of more than 200,000 customers belonging to multiple income segments. Not only oes Citibank offer the widest range of products, it has traditionally been the first to ınovate and offer new products and services.

.2.1.6.2 Script News:

aller: Tell me news at bank

ystem: In keeping with its reputation and commitment to its Cardmembers and ierchants, Citibank Pakistan has joined hands with 35 top of the line restaurants ›untrywide, to provide value added benefits under the umbrella of Privilege Plus rogram.

.2.1.6.3 Script Products:

aller: What are new products

ystem: With over 40 years of experience in Pakistan, Citibank has been a pioneer in ıtroducing new products and services that have been emulated across the financial ervices industry.

.2.1.6.4 Script Account Balance:

aller: What is my account balance

ystem: Welcome to the account balance inquiry service.

ystem: What is your account number?

aller: 239

ystem: Your PIN?

aller: 7825

ystem: (database interaction)

ystem: You are the authorized customer and your current balance is 500 dollar.

Caller: 239

System: Your PIN?

Caller: 7825

System: What is the fax number?

Caller: 8813247

System: (database interaction)

System: In short time u will receive fax. Thank you for using fax service.

3.2.1.6.8 Script Tools:

Caller: I want to deactivate my credit card

System: Welcome to the bank tool section. Here u can change deactivate your credit card.

System: What is your account number?

Caller: 239

System: Your PIN?

Caller: 7825

System: Your credit card number?

Caller: 8813247

System: (database interaction)

System: Now your credit card is deactivated, you can not use it more.

3.2.1.6.9 Script Trade Stocks:

Caller: I want to know about New York stock exchange

System: Welcome to the bank update stock rates.

System: It is New York stock exchange. Today stock rate is

System: (database interaction)

System: 3000

System: and change is stock is

System: +15 Percent

3.2.1.6.10 Script Center Operator:

Caller: Please transfer my call to center operator

System: Your call is going to transfer. Wait for a while.

System: (call transferred)

3.2.1.6.11 Script Contact Address:

Caller: What is contact address of bank

System: The mailing and contact address of bank is

System: (fetching data from database)

3.2.1.6.12 Script Driving Directions:

Caller: Tell me driving directions

System: (fetching data from database)

3.2.1.6.13 Script Authentication Check:

System: What is your date of birth

System: What is your month of birth

System: What is your year of birth

System: What is your social security number

System: What is city zip code

System: What is your secret number

System: What is your life insurance number

System: What is your car license number

System will ask one of them question from the caller.

3.2.1.7 Deciding Level of Information

To keep from overloading the user's short-term memory, information presented in a speech user interface must generally be more concise than information presented visually. Often, only the most essential information should be presented initially, with the opportunity for the user to access detailed information at a lower level.

For example, in banking application in which a user can request a list of recently cleared checks. In a visual interface, the application might return a table showing the check number, date cleared, payee name, and amount. A similar application with a speech interface might return only the check number and date cleared, and then permits the user to select a specific check number to hear the payee name and amount, if desired.

So keeping this in mind, I have made the scripts in concise manner, so that customer can interact easily and efficiently.

3.2.1.8 Full-Duplex versus Half-Duplex Implementation

Full-duplex implementations allow the application and the user to speak at the same time, permitting the user's speech to interrupt system prompts as the machine plays them, a feature known as barge-in. Full-duplex implementations require good echo cancellation, which you must configure in the telephony hardware for deployment.

On the surface, it might seem that full-duplex is always being preferable to half-duplex. It is easy to imagine experienced users wanting to interrupt prompts (especially lengthy ones) when they know what to say. There are situations, however, in which a half-duplex system will be as easy as or easier to use than a full-duplex system.

Table 3.2 Full Duplex versus half duplex implementation

Implementation	Advantages	Disadvantages
Full Duplex	Experienced users can interrupt system prompts to speed up the interaction. Users can say "Exit" to terminate the call and "Home" to replay the application.	Inexperienced users may inadvertently interrupt the prompt before hearing enough to form an acceptable response. You can minimize this problem by keeping system prompts short, to lessen the user's need to bargein; if your prompts are long, you should try to present key information early in the prompt.
Half Duplex	Guarantees that the entire prompt text plays. This maybe useful for the applications with lots of legal notices, advertisements or other information that you want to make sure always gets presented to the user. Creates a "my turn – your turn" rhythm for the application.	Experienced users cannot interrupt prompts; however if the prompts are short enough, users should not need to interrupt. Users my experience turn-taking errors. Keeping prompts short helps minimize this.

So by the advantages of full duplex, as bank customers usually know the process of bank transactions, here I am providing full duplex. So that customer can interact with application in less time as time is key factor in bank transactions.

2.2 Prototype Phase

The value of iterative design and incremental development has been written about extensively in software engineering practice. Particularly for voice application development, the iterative testing of detailed call flows with potential callers is very valuable.

The goal of this phase is to create a prototype of the application, leaving the design flexible enough to accommodate changes in prompts and application flow in subsequent states of the design phase.

In the prototype stage, I have produced detailed call flows and scripts for every prompt for the application. Based on the initial scripts, I work on the initial grammar designs for the prompts and determine the application behavior for every allowed state of the application.

For the first iteration in the prototype phase, I use a technique called "Wizard of Oz Testing." This technique can be done before coding, as it involves paper documentation of the prompts and application along with two humans - one to play the role of the user, and a "human wizard" to play the role of the computer system. The Wizard is familiar with the documentation while the user has no idea of the application flows. The User and the Wizard are physically separated and communicate with each other through a telephone. Each is given a call script that has the proposed introduction, list of prompts, list of active commands, and help options. The user then calls the Wizard who begins to read the script aloud. The user responds to the prompts and the Wizard continues the dialog based on the scripted response to the user's utterances. Wizard of Oz testing helps application developers fix problems in the script and task flow before code development.

3.2.2.1 Iterations of Voice Interactive System for Bank

After the Wizard of Oz testing phase, a functioning prototype is used to provide a more realistic assessment of how users will interact with the application. I also recommend that Wizard of Oz testing be performed in a lab with appropriate test users. I will say here that

each test is one iteration of application. Below is the Wizard of Oz testing with iterations of voice interactive system for bank based on initial scripts.

3.2.2.1.1 Iteration 1: Bank Information Call Script

Bank Information Call Script shows that if customer wants to get information about the bank then what dialogs will take place between him and voice application. First the voice application will say the intro message. Then on request application will fetch information about bank. At the end application will ask from the customer that whether he want to listen again or want to go back to main menu.

Table 3.3 Iteration 1 based on Initial Script

Message Type	Prompts and Responses	Titles and System Actions
System	To exit any time say exit. To go to the home page any time say home.	Play only at start.
About Bank	First opened in 1961 in Pakistan, today Citibank Pakistan serves the financial needs of the entire range of customers from individuals to the world's largest multinational companies. Across consumer banking products, it enjoys the trust and loyalty of more than 200,000 customers belonging to multiple income segments. Not only does Citibank offer the widest range of products, it has traditionally been the first to innovate and offer new products and services.	
Prompt	To repeat say yes. Say no, to go back.	
Caller	<Says 'yes' or 'no'>	
System		Do the requested task

3.2.2.1.2 Iteration 2: Current News Call Script

Current News Call Script shows that if customer wants to get update news or new schemes which are launched by the bank. Then he can ask for this by saying 'Current News'.

Table 3.4 Iteration 2 based on Initial Script

Message Type	Prompts and Responses	Titles and System Actions
System	To exit any time say exit. To go	Play only at start.

	to the home page any time say home.	
Current News	Citibank was the first to launch telephone banking capabilities and CitiPhone Banking provides customers with 24 by 7 access to a wide variety of services.	Database interaction
Prompt	To repeat say yes. Say no, to go back.	
Caller	<Says 'yes' or 'no'>	
System		Do the specific task

3.2.2.1.3 Iteration 3: Products Call Script

Products Call Script shows the dialog session between customer and voice application if he wants to get information about bank products such as Loans, Car financing or if he want to apply for a job.

Table 3.5 Iteration 3 based on Initial Script

Message Type	Prompts and Responses	Titles and System Actions
System	To exit any time say exit. To go to the home page any time say home.	Play only at start.
Products	Intro about bank products.	
Prompt	Please say one of the following products or services.	
Prompt	Car Financing	
Caller	<if says car financing>	
System		Fetching the requested.
Prompt	City Loans	
Caller	<if says city loans>	
System		Fetching the requested.
Prompt	Human resources	
Caller	<if says human resources>	
System		Fetching the requested.
Prompt	To repeat say yes. Say no, to go back to home.	
Caller	<says 'yes' or 'no'>	
System		Do the requested.

3.2.2.1.4 Iteration 4: Account Balance Call Script

Account Balance Call Script shows the dialogs between customer and voice application, if he wants to get balance of his account. For this voice application will ask his account number and pin number and upon authentication it will say to the user his account balance.

Table 3.6 Iteration 4 Based on Initial Script

ssage Type	Prompts and Responses	Titles and System Actions
tem	To exit any time say exit. To go to the home page any time say home.	Play only at start.
count Balance	Intro.	
mpt	What is your account number?	
ller	<will say account number>	
mpt	Pin Number?	
ller	<will say pin number>	
stem		Verification
stem		Fetching the requested.
stem	Display to the user.	
stem	To repeat say yes. Say no, to go back to home.	
ller	<says 'yes' or 'no'>	
stem		Do the requested.

2.2.1.5 Iteration 5: Money Transfer Call Script

oney Transfer Call Script will be held between customer and voice application when he nts to transfer money from one account to another account. For this voice application ill ask his account number, pin number, amount of money to be transfer and other count number. After authentication and verification money shall be transfer.

Table 3.7 Iteration 5 Based on Initial Script

essage Type	Prompts and Responses	Titles and System Actions
stem	To exit any time say exit. To go to the home page any time say home.	Play only at start.
oney Transfer	Intro.	
ompt	What is your account number?	
ller	<will say account number>	
ompt	Pin Number?	
ller	<will say pin number>	
ompt	How much money you want to transfer?	
ller	<will say amount to transfer>	
ompt	To which account number you want to transfer?	
ller	<will say account number>	
stem		Verification
tem		Transaction take placed.
tem	Result display to the user	
tem	To repeat say yes. Say no, to go back to home.	
ler	<says 'yes' or 'no'>	
tem		Do the requested.

3.2.2.1.6 Iteration 6: Bill Payment Call Script

Bill Payment Call Script shows the dialog session between customer and voice application when he wants to pay a bill. For this voice application will ask his account number, pin number, bill id. Upon authentication and verification bill amount will be paid.

Table 3.7 Iteration 6 Based on Initial Script

Message Type	Prompts and Responses	Titles and System Actions
System	To exit any time say exit. To go to the home page any time say home.	Play only at start.
Account Balance	Intro.	
Prompt	What is the bill id?	
Caller	<will say bill id>	
Prompt	What is your account number?	
Caller	<will say account number>	
Prompt	What is your pin number?	
Caller	<will say pin number>	
System		Verification
System		Do the requested
System	Result display to the user.	
System	To repeat say yes. Say no, to go back to home.	
Caller	<says 'yes' or 'no'>	
System		Do the requested

3.2.2.1.7 Iteration 7: Fax Call Script

If customer wants to get his account detail through fax then Fax Call Script dialog will be take place. Voice application will ask customer account number, pin number and fax number.

Table 3.8 Iteration 7 Based on Initial Script

Message Type	Prompts and Responses	Titles and System Actions
System	To exit any time say exit. To go to the home page any time say home.	Play only at start.
Fax	Intro.	
Prompt	What is your account number?	
Caller	<will say account number>	
Prompt	What is your pin number?	
Caller	<will say pin number>	
Prompt	What is the fax number?	
Caller	<will say fax number>	
System		Do the requested
System	Result display to the user.	
System	To repeat say yes. Say no, to go back to home.	

ller	<says 'yes' or 'no'>	
stem		Do the requested

2.2.1.8 Iteration 8: Credit Card Call Script

redit Call Script shows the dialog session if customer wants to activate or deactivate edit card. Voice application will ask customer account number, pin number and credit rd number. Upon authentication and verification system will do the requested.

Table 3.9 Iteration 8 Based on Initial Script

lessage Type	**Prompts and responses**	**Titles and system actions**
ystem	To exit any time say exit. To go to the home page any time say home.	Play only at start.
redit Card	Intro.	
rompt	What is your account number?	
aller	<will say account number>	
rompt	What is your pin number?	
aller	<will say pin number>	
rompt	What is your credit card number?	
aller	<will say credit card number>	
ystem		Verification
rompt	Do you want to activate/deactivate credit card?	
aller	<will say Activate/Deactivate>	
ystem		Do the requested
ystem	Result display to the user.	
ystem	To repeat say yes. Say no, to go back to home.	
aller	<says 'yes' or 'no'>	
ystem		Do the requested

3.2.2.1.9 Iteration 9: Trade Stocks Call Script

Trade Stocks Call Script shows the dialog session between user and application when ones want to know the stock rates of Karachi, New York or Doe Jones.

Table 3.10 Iteration 9 Based on Initial Script

Message Type	**Prompts and responses**	**Titles and system actions**
System	To exit any time say exit. To go to the home page any time say home.	Play only at start.
Trade Stocks	Intro.	
Prompt	Please say one of the following.	
Prompt	Karachi	
Prompt	New York	
Prompt	Doe Jones	
Caller	<says any>	
System		Do the requested.

Prompt	Result display to the user.	
System	To repeat say yes. Say no, to go back to home.	
Caller	<says 'yes' or 'no'>	
System		Do the requested

3.2.2.1.10 Iteration 10: Call Transfer Call Script

Call Transfer Script shows the dialog session when customer didn't satisfy or he didn't get the required information from the voice application. On request call will be transfer to the customer care center operator. Customer will be remained in session, so if he want to come back to voice application environment again he can.

Table 3.11 Iteration 10 Based on Initial Script

Message Type	Prompts and responses	Titles and system actions
System	To exit any time say exit. To go to the home page any time say home.	Play only at start.
Call Transfer	Intro.	
Prompt	Your call is going to transfer	
System		Do the requested

3.2.2.1.11 Iteration 11: Contact Address Call Script

Contact Address Call Script will be held between user and voice application when he wants to know the contact or mailing address of bank.

Table 3.12 Iteration 11 Based on Initial Script

Message Type	Prompts and responses	Titles and system actions
System	To exit any time say exit. To go to the home page any time say home.	Play only at start.
Contact Address	Intro.	
System		Fetch the requested
Prompt	Result display to the user.	
System	To repeat say yes. Say no, to go back to home.	
Caller	<says 'yes' or 'no'>	
System		Do the requested

3.2.2.1.12 Iteration 12: Driving Directions Call Script

Driving Directions Call Script will be held between customer and voice application when he wants to know the driving directions of bank to reach it. Customer can say "Tell me driving directions".

Table 3.13 Iteration 12 Based on Initial Script

Message Type	Prompts and responses	Titles and system actions
System	To exit any time say exit. To go to the home page any time say home.	Play only at start.
Driving Directions	Intro.	
System		Fetch the requested.
Prompt	Result display to the user.	
System	To repeat say yes. Say no, to go back to home.	
Caller	<says 'yes' or 'no'>	
System		Do the requested

3.2.2.2 Voice Application Usability Heuristics

Voice application usability heuristics are guidelines that describe an approach to support established best practices for usability of a voice application. Established best practices for usability should be followed so as to minimize usability issues that may result in application rework later on.

3.2.2.3 User Memory and Cognitive Processes

Users of voice applications rarely devote their full attention and thinking while using the application. There are other constraining factors that a voice application is faced with: environment noise, competition for user's attention, voice recognition limitations, etc. It is imperative that the user should not be subjected to cognitive and memory overloads either from the design of the VUI itself or from the complicated task flows that the application may have. Otherwise, the users may perceive that the application is "difficult" to use and may not use it. The approach to this heuristic is to first emphasize that the requirements specifications for user analysis and task analysis should be done as thoroughly as possible. This gives a good idea as to how many users will use the voice application for performing tasks and under what circumstances. Streamline the task flow in the application; reduce branching while supporting expected functionality without having unnecessary features. When there are problems with the ASR functionality, either

due to the limitations of the ASR engine itself or due to noise in the user's environment, offer DTMF grammar as an alternative to spoken user input. Use effective prompt design to aid user's navigation and tiered responses to error conditions.

3.2.2.4 Maintain a Simple and Natural Dialog

Dialogs within a voice application can range from those that sound like IVR systems to those that approach natural language interaction ("How may I help you?"). The overall structure of the dialogs in voice application develops the users' mental model of what they are doing within a voice application. The design of the individual prompts and responses within a dialog creates the conversational aspects of the voice application. Here are some of the design heuristics that ensure better usability:

- Follow the "Goal->Action" prompt structure consistently
- Construct error responses that specify what constitutes valid input and suggest correct kind or format of response.

- Allow users to select from a list of categories or terms instead of just prompting them for open-ended input.
- Use prompt designs that suggest or limit the logical range on input ("Would you like to save your recording?").
- Use terms in prompts that exactly parallel the terms in the grammar specification.

3.2.2.5 Provide Clear Navigation and Shortcuts

Two primary reasons other than poor design that might frustrate users are:

- They have to rely on their memory, which is fleeting.
- They may use the application infrequently, so they don't really learn it.

1. There is another problem with the frequency of use: frequent users of the voice application might find that navigation designed for novice and infrequent users slows them down. Such users prefer shortcuts to the point in the application where they want to be headed. To handle these two kinds of users:
    - Use the application greeting to indicate important aspects of the navigation, such as commands to access help, start over, etc.

- Help users recognize what they can do in the application.
- Guide users to provide usable input.
- Respond to uncertainties about the meaning of user input.
- Return users to known starting points if they get too confused.
- Provide error recovery based on a best guess at what uncertain user input represents.
- Interleave context sensitive help within error messages.
- Provide global navigation commands that are available anywhere in the application.
- Provide local navigation commands with a specific module of the voice application including error recovery and context-sensitive help.
- Provide shortcuts that bypass the normal navigation structure of the application. This can be unpublished global or local navigation commands, support for barge-in, and DTMF alternatives to spoken commands.

2. Provide consistent feedback and user-friendly error handling. This heuristic is closely related to the one above. It is important to give consistent feedback to make sure that user frustration is minimized. Some of the ways to achieve better user experience are:

    - The voice application should give users feedback while the application is waiting for Internet content and while external systems or the network are creating lag times. This could take the form of a message which says, "Please wait while I get your information".
    - Use more complex error handling routines, including tiered feedback and customized error messages, depending on the user's position in the application and number of error conditions encountered in sequence. Phrase error messages in non-threatening language such as "Sorry, I did not understand your input" followed by the original prompt, or provide context-sensitive help followed by a shorter version of the original

prompt. Suggest a response to the error message or condition such as "For e-mail, press one, or say e-mail; for fax, press two, or say fax."

- Never disconnect the user after a few error conditions are encountered in sequence. Take the user to a logical point in the application or offer the choice to start over. If the user does not want to interact with the voice application, offer to transfer to a live operator, but never make the user hang up.
- Maintain consistency in navigation and terminology. Building consistency into a VUI provides users with a sense of control and minimizes frustration levels. Consistency helps users to have a familiar mental model of the application, which leads to better user experience. Some aspects that can make a voice application and its VUI consistent are:
  - Use the same voice consistently throughout the application.
  - Use consistent commands for global navigation.
  - If you offer the option of DTMF input as an alternative to spoken input in one module of the voice application, implement it uniformly across all modules of the application.
  - Use consistent commands and prompt structure for local navigation when the same function is being performed in different modules. For example, don't prompt, "To exit the application, say quit" in one part of the application and then prompt, "To quit from the application say exit or press *Q" in another part of the application."
  - Assess voice and personality selection. Gathering user preferences regarding voice and personality selection is an important part of the user task analysis of the requirements phase. Based on user feedback, the voice and application personality for the voice application can be selected. Some of the aspects of voice and personality application include:

3. Select a voice type for recorded audio or TTS that is acceptable to the majority of the users and matches the user's requirements. For example, studies of North

American audiences show that a female contralto voice is probably the most acceptable. If the TTS voice is close to lifelike, then almost all content can be delivered using it and the application will appear seamless to users. However, it is important to test the voice type and personality (possibly with a specific terminology, slang, and style) on a sample of the target users and test the prompts. If TTS is unacceptable, then switch to recorded audio.

3.2.3 Design and Development Phase

Up to this point, the development cycle for a voice application has involved the preliminary phases of requirement specifications and exploring "proof of concept" using a prototype. The information gathered from the prototyping phase and, perhaps, from the usability studies leads to the next phase -- that of design. At this point I have develop a clear plan for the application and all of the detailed work necessary to craft the requirements specification into a working voice application. This working application can then be handed to the QA team for system testing and then on to commercial deployment. Now I am going to look at the design and implementation phase from a software engineering perspective.

Typically, the design phase is divided into the *high level design* process where we develop a higher-level plan for the application and define the system-level interfaces; and the *low level design* phase where we develop the details of coding VoiceXML pages, designing prompts and dialogs. The most critical aspect of the design phase is the design of the voice user interface itself. The VUI comprises the "front-end" of the application and this is the aspect of the voice application that the users would be interacting with. The importance of the VUI gives much credence to the necessity to devote more time to the VUI application design than to the actual implementation. Understanding the content that is to be incorporated in the VUI and how it is going to be accessed comprises the "back-end" development of the voice application. The VUI and some of the back-end functionality will be implemented in VoiceXML pages. The back-end data and content retrieval will be implemented according to the interface necessities developed in the requirements phase.

2.3.1 Adopting a Consistent "Sound and Feel"

To promote a consistent sound and feel, I adopt the following scenarios.

2.3.1.1 Designing Prompts

When designing prompts, I choose one prompt style and use it consistently throughout voice interactive application for bank. In general, I use the present tense and active voice for all prompts. For example, use:

> System: How much money you want to transfer? [!!!]

rather than:

> System: Amount to be transferred? [!!!]

2.3.1.2 Standardizing Valid User Responses

Users master applications more quickly when they can predict what responses are valid. For example, I have used positive responses of "Yes," "Okay," and "True" to one yes/no question, and allow the same responses for all yes/no questions in voice application. .

2.3.1.3 Using Consistent Timing

Consistent timing is important in developing a conversational rhythm for the dialog. Activities which I have considered include:

- The amount of user "think time"
- The length of pauses between menu items
- The amount of time the system takes to respond to a user utterance

2.3.1.4 Designing Consistent Dialogs

When users don't know what they can say at a given point in a dialog, the interaction between the user and the application can quickly break down. To help users avoid this "What can I say now?" dilemma, I have adopted consistent sound and feel standards to create dialogs that behave consistently.

2.3.1.5 Writing Directive Prompts

Have written prompts that clearly indicate to users what they can say. For example:

System: To exit any time say exit. To go to the home page any time say home.

or

System: Say Products, Manger, Center Operator.

3.2.3.1.6 Handling the Recognition Results

Once voice application receives text from the speech recognition software, it needs to do something with it. If the caller says "*checking account*," the application should read the caller's current checking account balance back to her. Based on the results of the recognition, voice application may request another VoiceXML page to be fetched from the web application server, or a backend database query will occur and the results will be presented back to the caller, or the dialog may continue so that your application can get more input from the caller. Furthermore, voice application needs to consider "normal" speech recognition errors, such as when the user says something that is not in one of the active grammars, or the caller speaks indistinctly, or the caller doesn't say anything at all.

3.2.3.2 IBM WebSphere Voice Server SDK

I am using IBM WebSphere Voice Server SDK which provides support for bringing VoiceXML to Web application development activities, in essence, providing a spoken equivalent to visual browsing. Second I am using WebSphere Voice Toolkit to create a voice application written in VoiceXML. VoiceXML pages can be static or may be generated dynamically from CGI scripts, Java Beans, ASPs, JSPs, Java servlets, or other server-side techniques.

WebSphere VoiceServer SDK is supported on Microsoft® Windows 2000 Professional with Service Pack 2 and on Windows2000 Server with Service Pack 2. The SDK uses the workstation's speakers to play audio output. Data can be input using the workstation's microphone, prerecorded audio files, or the WebSphere Voice Server SDK's DTMF simulator (to simulate any telephone key input). The SDK also supports text mode and automated testing. When deploy voice applications, users can interact with them through spoken or DTMF commands and audio output.

3.2.1 Contents of the Voice Server SDK

e SDK includes:

- A speech browser that interprets VoiceXML markup. This VoiceXML browser includes a DTMF Simulator to generate simulated telephone keypress input during desktop testing.
- The IBM speech recognition engine which accepts spoken input and recognizes the input as one or more words.
- A text-to-speech (speech synthesis) engine that converts the speech-recognition engine's output to audio. In some WebSphere Voice Server publications, this component is referred to as the formant text-to-speech engine.
- Telephony acoustic models to approximate the speech recognition behavior of applications deployed in a telephony environment.
- Tools for desktop testing of speech applications.
- An audio setup program to configure microphone and speakers for use with this product.

3.2.2 Required Hardware and Software

mputer equipped with the following minimum hardware is required:

- Intel Pentium® 366 MHz processor or equivalent
- 128 MB RAM
- 290 MB disk space for each language to be installed, which includes:
  - 30 MB for installing the Sun® Java Runtime Environment (Sun JRE) 1.3.1
  - 80 MB in the Windows system directory
  - 130 MB disk space in the destination installation directory for installing each language selected
  - 28 MB in the installation destination directory for caching, logging, and tracing

stall IBM WebSphere Voice Server Concatenative Text-to-Speech, the following ional resources are required:

150 MB RAM for each CTTS language

640 MB for each CTTS, of which 300 MB are for temporary storage

- A display adapter with a setting of greater than 256 colors
- A Microsoft Windows 2000 compatible, 16-bit, full-duplex sound card (with a microphone input jack) with good recording quality
- A quality microphone

e WebSphere Voice Server SDK requires the following software:

- Microsoft Windows 2000 Professional or Server with Service Pack 2.
- Networking (for example, an IP network) must be enabled.
- Sun Java Runtime Environment (Sun JRE) Version 1.3.1 (included with the SDK but must be installed
- before the IBM WebSphere Voice Server SDK software)
- HTTP 1.1 client, if user-customized requests to the Web-based server are desired.

2.3.2.3 Architectural Overview

gure 3.2 shows how the WebSphere Voice Server SDK interfaces with information ored on Web application servers and in back-end enterprise databases.

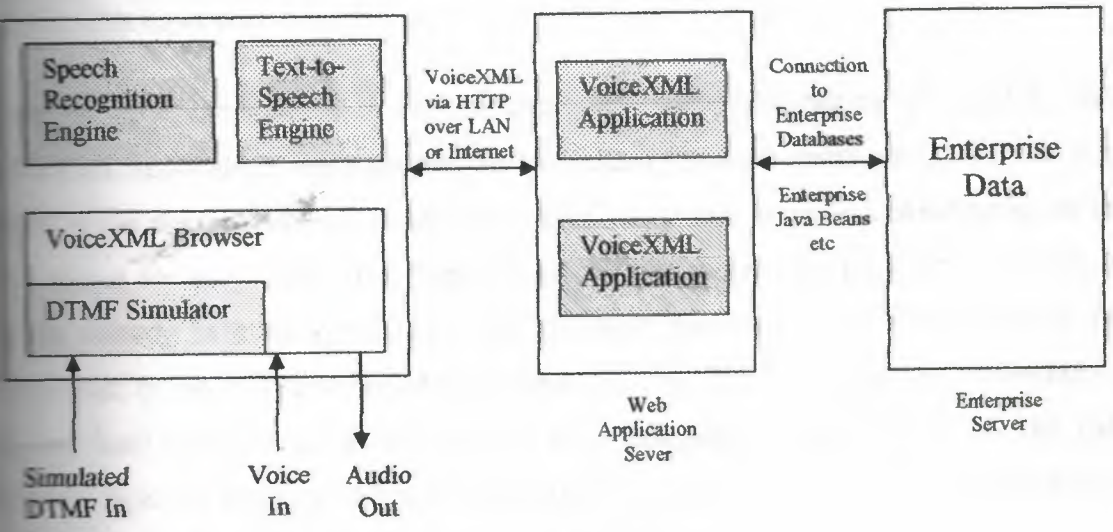


Figure 3.2 WebSphere Voice Server SDK Interaction with Web and Enterprise Servers

he VoiceXML applications will generally reside on a Web application server, although hey can reside on your workstation during early testing; the VoiceXML browser uses TTP over a LAN or the Internet to the fetch the documents.

/oiceXML document specifies an interaction, or dialog, between the user and the tion. The VoiceXML browser interprets and renders the VoiceXML document, es workstation's speakers to play the prompts, instructions, and menu choices ed in the document. The information played can be:

Prerecorded audio files

Synthesized by the text-to-speech engine from text specified in your VoiceXML files

on the state of the dialog, the VoiceXML browser also enables and disables speech ition grammars, as specified by VoiceXML application. Developer will respond to ompts and make menu selections by speaking into a microphone connected to ation. Speech is passed to the speech recognition engine, which compares it to the valid inputs specified in the currently active grammars.

eXML application will support DTMF input when it is deployed in a telephony nment, developer can use the WebSphere Voice Server SDK's DTMF Simulator to te this input on workstation.

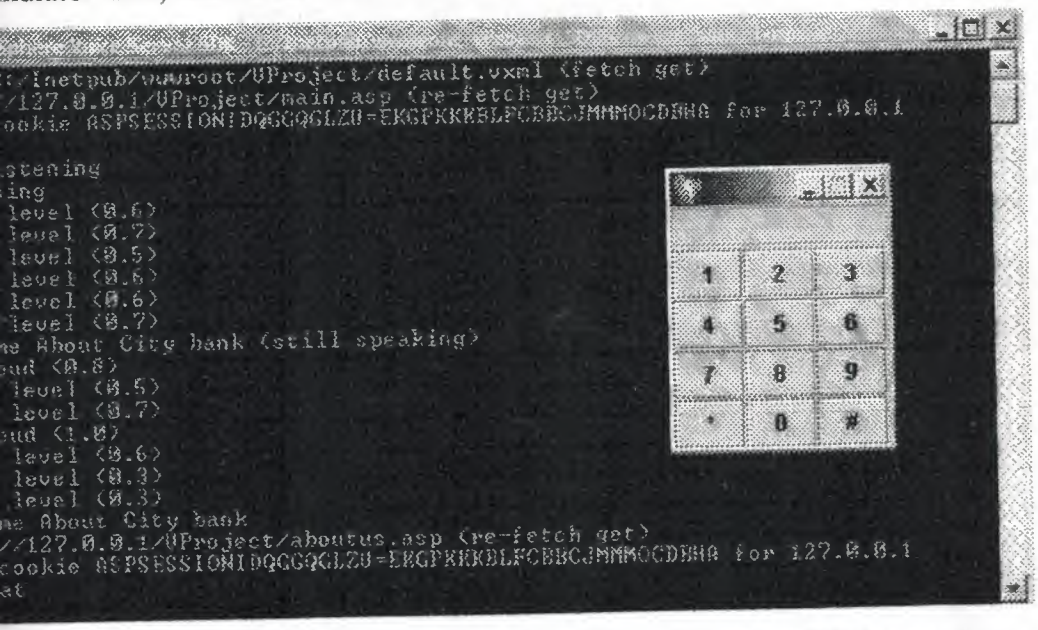
on input, the VoiceXML browser proceeds with the interaction specified by your XML application. For example, the VoiceXML browser might continue to the next in the document, fetch a new VoiceXML document, or submit information to the server for processing. The Web server can use server-side programs to locate or e records in a back-end enterprise database and return the information to the XML browser. The VoiceXML browser presents this information to you either by g back prerecorded audio files or by synthesizing speech based on the data ed from the database. When a dialog does not specify a transition to another dialog, oiceXML browser exits and the session terminates.

2.4 VoiceXML Browser

of the primary functions of the VoiceXML browser is to fetch documents to process. equest to fetch a document can be generated either by the interpretation of a *XML document, or in response to an external event.*

XML browser manages the dialog between the application and the user by udio prompts, accepting user inputs, and acting on those inputs. The action olve jumping to a new dialog, fetching a new document, or submitting user e Web server for processing.

console provides trace information about the prompts played, resource files nd user input recognized by the VoiceXML browser. Other than this and the mulator GUI, there is no visual interface.



ıre 3.3 IBM WebSphere Voice Server SDK Browser running in Audio Mode

Interaction with the DTMF Simulator

oiceXML application is deployed in a telephony environment, developer may allow users to provide DTMF input in addition to spoken input. The DTMF or is a GUI tool that enables to simulate DTMF tones on workstation. The DTMF or plus microphone and speakers take the place of a telephone during testing, to run and evaluate VoiceXML applications without having to connect to y hardware and the PSTN (Public Switched Telephone Network).

ıe DTMF Simulator, one can simulate a telephone keypress event by pressing the onding key on the computer keyboard or clicking on the corresponding button on MF Simulator GUI, shown in Figure.

Figure 3.4 DTMF Simulator GUI

BM WebSphere Voice Toolkit

e Toolkit includes an integrated development environment, a VoiceXML editor, r creating grammars written in JSGF or Speech Recognition Control Language, ciation builder, a basic audio recorder, a set of VoiceXML reusable dialog nts, and a debugger. Developer can use these tools to:

evelop VoiceXML applications

eate grammars and pronunciations for use with other products that employ the M speech

cognition and text-to-speech engines (for example, IBM WebSphere Voice sponse and certain non-IBM IVR platforms)

f the WebSphere Voice Toolkit include:

urce mode VoiceXML editor with support for VoiceXML 2.0 plus IBM ensions, as shown in

urce mode grammar editor for JSGF and Speech Recognition Grammar mats.

lorizing feature in the VoiceXML and grammar editors to highlight element s, attributes, comments, etc.

ntent assist in the VoiceXML editor, to list the elements and attributes that are d in the current location within the source code.

tax checking in the grammar editor to validate code.

ility to generate and tune pronunciations for speech recognition and speech nthesis. (This function is only available if the WebSphere Voice Server SDK stalled.)

basic audio recorder for creating and playing audio files.

usable dialog components that allow you to easily add common functions to ur VoiceXML applications.

ility to launch WebSphere Voice Server SDK to test applications. The SDK is quired if you want to use certain functions of the Voice Toolkit. For example:

- Highlight unknown words in the Grammar editor
- Generate a default pronunciation in the pronunciation builder
- View a list of supported pronunciation phonemes
- Play a pronunciation in the pronunciation builder
- Test a VoiceXML application by launching the SDK's VoiceXML browser

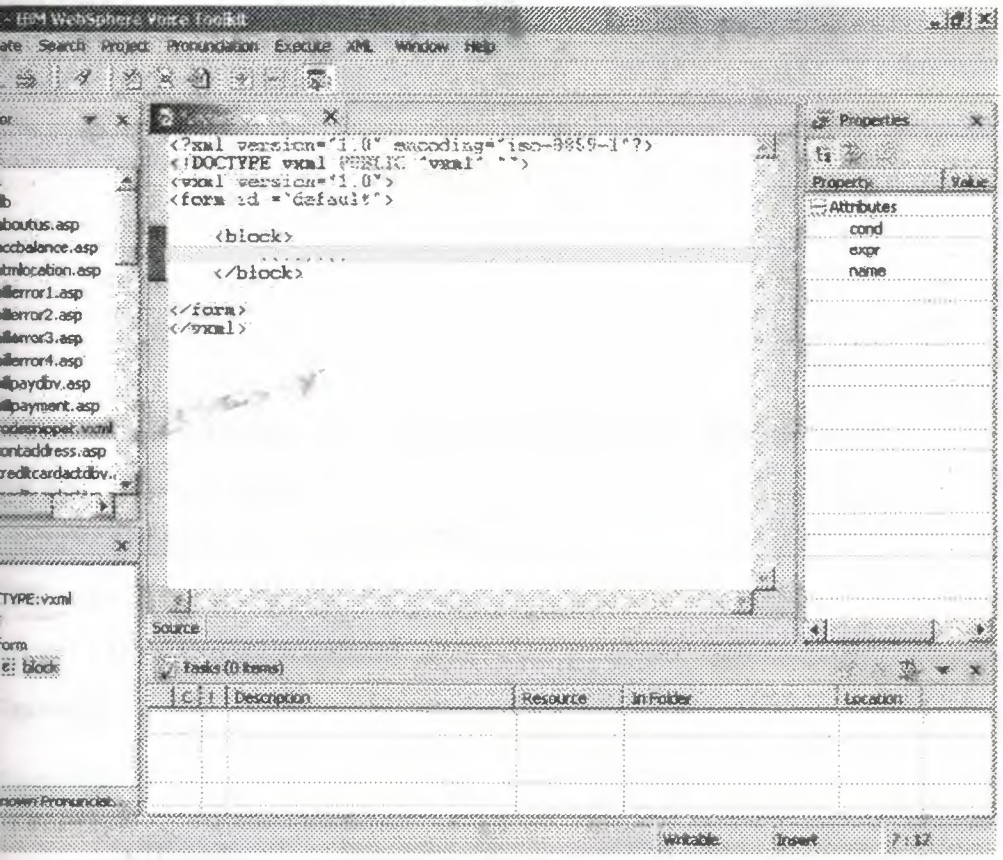


Figure 3.5 IBM WebSphere Voice Toolkit

tive Server Pages

dynamic Web page technology of the Microsoft. ASP code can take advantage braries, DCOM components, and classes developed in the many languages by the framework. ASP Web pages are usually hosted on an Internet n Server (IIS) and Windows platforms. Active Server Pages are processed by X Component called a scripting engine. I have used Active Server Pages to data between clients and servers via the HTTP protocol. When a server receives HTTP request, the server loads the document requested by the client.

pages I have used Visual InterDev 6.0 provided by Microsoft. It gives GUI o write ASP pages. It also validates line by line if error come, and also g facilities. Another powerful feature is database integration. So that it can erver side scripts as well through Visual InterDev tool.

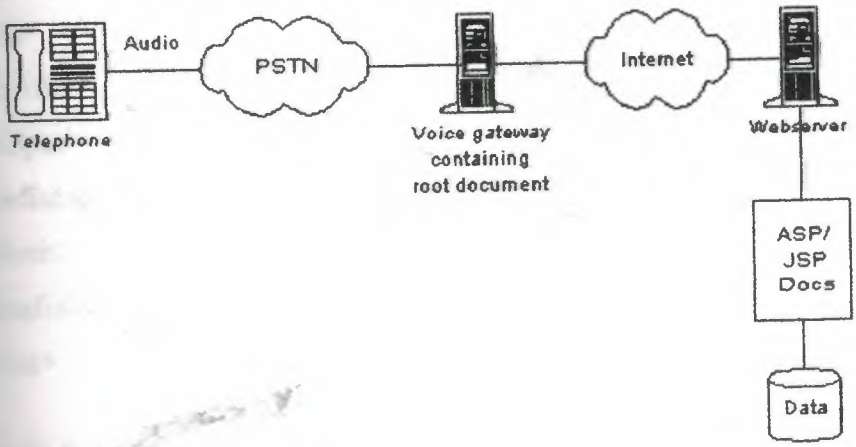


Figure 3.6 Architecture of Voice Activated System with Web Server

nple ASP code will be look like this.

```
ersion="1.0" encoding="iso-8859-1"?>
'YPE vxml PUBLIC "vxml" "">
ersion="1.0">
 <prompt>
        ...
 </prompt>
 <%
        ASP code
 %>
        ...
```

3.5 Dialogue Coding

e I am giving some code snippets from coding, which will tell main functionality
g with syntax. Also I am describing main tags which are used in dialog session.
se are:

- <vxml>
- <block>
- <form>
- <menu>
- <choice>
- <grammar>
- <dtmf>
- <prompt>
- <field>
- <record>
- <noinput>
- <subdialog>
- <submit>
- <transfer>
- <filled>

2.3.5.1 Code Snippet 1

```
?xml version="1.0" encoding="iso-8859-1"?>
!DOCTYPE vxml PUBLIC "vxml" "">
<vxml version="1.0">
<form id ="default">
    <block>
        <goto next = "http://127.0.0.1/VProject/main.asp"/>
    </block>
</form>
</vxml>
```

a voice application root document. Which don't have any dialogue. It just call
er main document which resides on server. The first line in the code
l version="1.0" encoding="iso-8859-1"?>
hat it is validated document and parsed by xml parser. Second line
CTYPE vxml PUBLIC "vxml" "">
he document type definition of the current document (DTD), which is publicly
d. Document Type Definition is actually definition of each tag used in current
ent. It also specifies rules to define new tags if necessary. Third line
version="1.0">
at it is VoiceXML document. These three lines must be include in every document,
it is ASP page.

2 Code Snippet 2

e next = "/VProject/aboutus.asp#aboutus">About City Bank
<grammar type="application/x-jsgf">
[Please] [Tell me] About [City] bank |
[What is] City [Bank]
</grammar>
e>

will prompt "About City Bank", as Choice tag is used to select choice from the
nder the choice tag Grammar tag is defined. So that if caller will say "Please Tell
t bank". System will recognize it and jump to the "aboutus.asp" document.

Code Snippet 3

block>
<emp>Welcome</emp> to the Bank.
/block>

ent is a form item. It contains executable content that is executed if the block's
variable is undefined and the block's cond attribute, if any, evaluates to true.

5.4 Code Snippet 4

type="mime-type">

<!-- inline dtmf grammar -->

f>

dtmf> element is used to specify a DTMF grammar that

defines a set of key presses that a user may use to perform an action or supply information, and

defines the corresponding string value that describes that information or action.

5.5 Code Snippet 5

name="customer_accno" type="digits">

<prompt>

What is your account number?

</prompt>

<catch event="help">

Please type in or spell out your account number.

</catch>

>

field> type attribute is used to specify a built-in grammar for one of the ental types, and also specifies how its value is to be spoken if subsequently used lue attribute in a prompt.

.6 Code Snippet 6

id = "home">

e next = "/VProject/aboutus.asp#aboutus">About City Bank

<grammar type="application/x-jsgf">

[Please] [Tell me] About [City] bank |

[What is] City [Bank]

</grammar>

e>

next = "/VProject/accbalance.asp">Account balance

<grammar type="application/x-jsgf">

I want to check [my] [account] [balance]|

What is [my] [account] balance |

How much [money] left in [my] account

/grammar></choice>

is convenient syntactic shorthand for a form containing a single anonymous field mpts the user to make a choice and transitions to different places based on that

Code Snippet 7

name="greeting" beep="true" maxtime="10s"

inalsilence="4000ms" dtmfterm="true" type="audio/wav">

prompt>

At the tone, please say your words.

/prompt>

noinput>

I didn't hear anything, please try again.

/noinput>

>

cord> element is a field item that collects a recording from the user. The g is stored in the field item variable, which can be played back or submitted to a The user is prompted for a greeting and then records it. The greeting is played d if the user approves it, is sent on to the server for storage using the HTTP ethod.

Code Snippet 8

block>

<emp>Welcome</emp> to the Bank.

To exit any time say exit. To go to the home page any time

say home.

```
</block>
<subdialog src = "#home"/>
rm>
```

ubdialog> element invokes a "called" dialog (known as the *subdialog*) identified by rc attribute. The subdialog executes in a new execution context. The subdialog eeds until the execution of a <return> element which causes the subdialog to return. n the subdialog returns, its execution context is deleted, and execution resumes in the ng dialog with any appropriate <filled> elements. An execution context includes all arations and state information for the dialog, the dialog's document, and the ication root (if present). Subdialogs can permit the reuse of a common dialog such as example of prompting a user for credit card information, or build libraries of reusable ications.

3.5.9 Code Snippet 9

```
ck>
mit next="billpaydbv.asp#billpaydbv" method="post" namelist="list of variables"/>
ock>
```

mit a list of variables to the document server via an HTTP GET or POST request.

3.5.10 Code Snippet 10

```
nsfer name="mycall" dest="phone://180012345" connecttimeout="30s"
ge="false">
    <filled>
        <assign name="mydur" expr="mycall$.duration"/>
        <if cond="mycall == 'busy'">
            <prompt>
                Sorry, our customer support team is busy serving other
                customers. Please try again later.
            </prompt>
        <elseif cond="mycall == 'noanswer'"/>
            <prompt>
```

```
                Sorry, our customer support team's normal hours are 9 am
                to 7 pm Monday through Saturday.
            </prompt>
        </if>
</filled>
fer>
```

onally, it is appropriate to suspend the session between the user and the interpreter tiate a session with another entity. The most common use for this capability in practice is to connect a user in a telephone conversation with a interpreter to a rty through the telephone network. The <transfer> element directs the interpreter e such a third party connection. During a bridge transfer, the platform can listen MF input from the caller. In particular, if a DTMF grammar appears inside the er> element, DTMF input matching that grammar will terminate the transfer and control to the interpreter. A bridge transfer may be terminated by recognition of an ce matching an enclosed <grammar> element; support of this feature is not d. The <transfer> element is modal in that no grammar defined outside its scope is

11 Code Snippet 11

```
d = "main">
>

Dim valuecase
valuecase = 0
Randomize
valuecase = CInt(Int((8 * Rnd()) + 1))
if valuecase = 1 then
        valuecase = 0
        %>
        <goto next = "#main1"/>
        <%
elseif valuecase = 2 then
        valuecase = 0
        %>
        <goto next = "#main2"/>
        <%
```

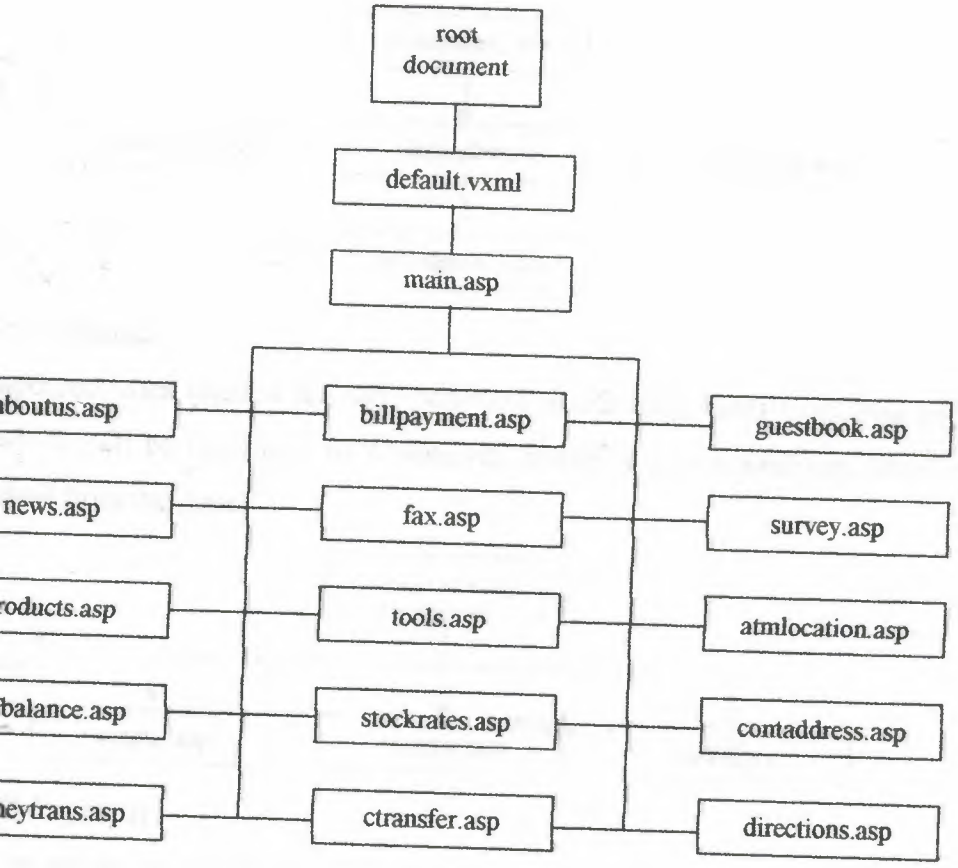
```
elseif valuecase = 3 then
        valuecase = 0
        %>
        <goto next = "#main3"/>
        <%
elseif valuecase = 4 then
        valuecase = 0
        %>
        <goto next = "#main4"/>
        <%
elseif valuecase = 5 then
        valuecase = 0
        %>
        <goto next = "#main5"/>
        <%
elseif valuecase = 6 then
        valuecase = 0
        %>
        <goto next = "#main6"/>
        <%
elseif valuecase = 7 then
        valuecase = 0
        %>
        <goto next = "#main7"/>
        <%
elseif valuecase = 8 then
        valuecase = 0
        %>
        <goto next = "#main8"/>
        <%
end if
```

k>
n>

eck whether user is authorized or not. I am asking one question out of eight ons randomly. After getting value from user, system verifies from the database.

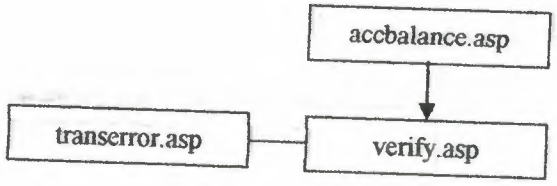
6 Hierarchy of Voice Application Documents

Interactive System for Bank Transaction gives you a menu to select from list of es. Once you make your selection, you are taken to the appropriate document. rchy of documents are as under.



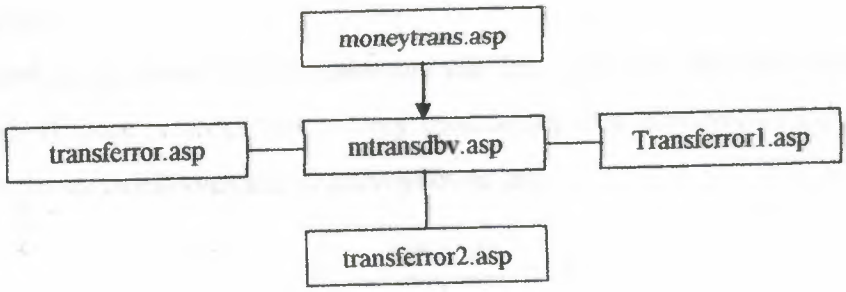
ccount Balance

ccount balance accbalance.asp file will be called, which will take account l pin number and using verify.asp it will verify and fetch the balance from the database. If user didn't give correct account information p will be called.



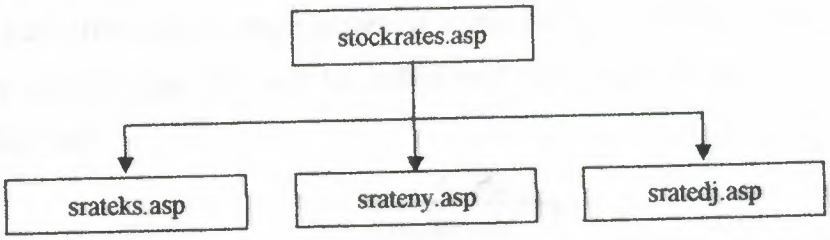
ney Transfer

fer will be take place through moneytrans.asp which will take account number, money to transfer and account number where to transfer money. ransdbv.asp will be called if any information given by customer is not error1.asp or transferror.asp will be called.



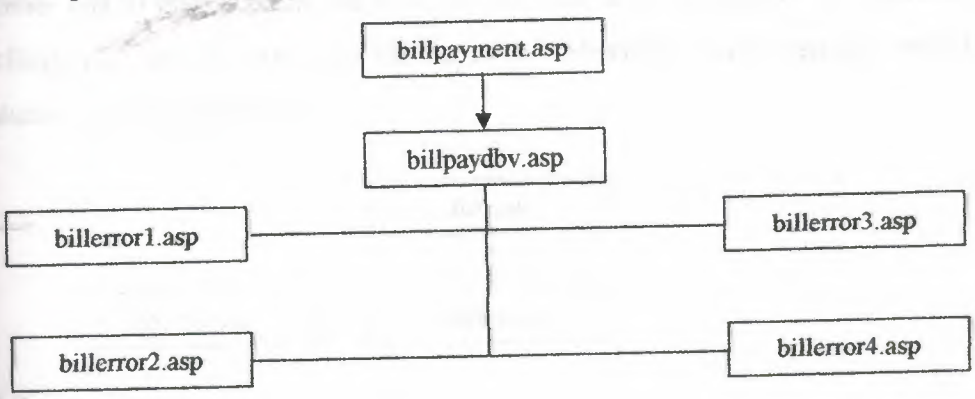
3 Stock Rates

k updated stock rates of Karachi, Newyork and/or Doe Jones stockrates.asp will d which will be redirected to srateks.asp, srateny.asp or sratedj.asp. Stock rates fetched from database.



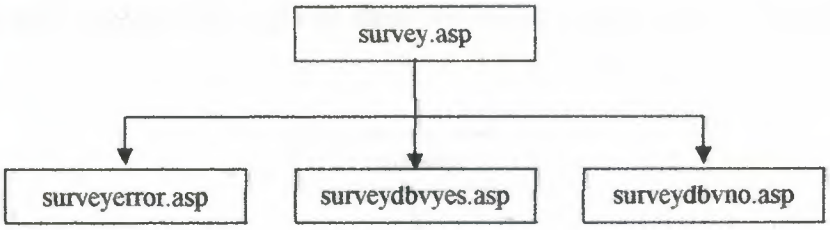
4 Bill Payment

omer wants to pay utility bill billpayment.asp file will work and will get account r, pin number and bill id. After this billpaydbv.asp will be called. If some error or system have to give notification then billerror1.asp, billerror2.asp, billerror3.asp error4.asp will be called.



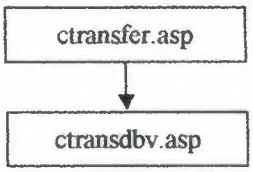
.5 Survey

is conducting some survey customers can take part in it. For this survey.asp file called. If there is no current survey conducting then surveyerror.asp file will be otherwise surveydbvyes.asp or surveydbvno.asp.



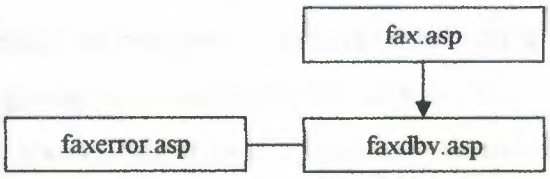
.6 Call Transfer

nsfer call from voice application to customer care center operator or bank ntative ctransfer.asp file will be called and also ctransdbv.asp to store the call ation for bank.



.7 Fax

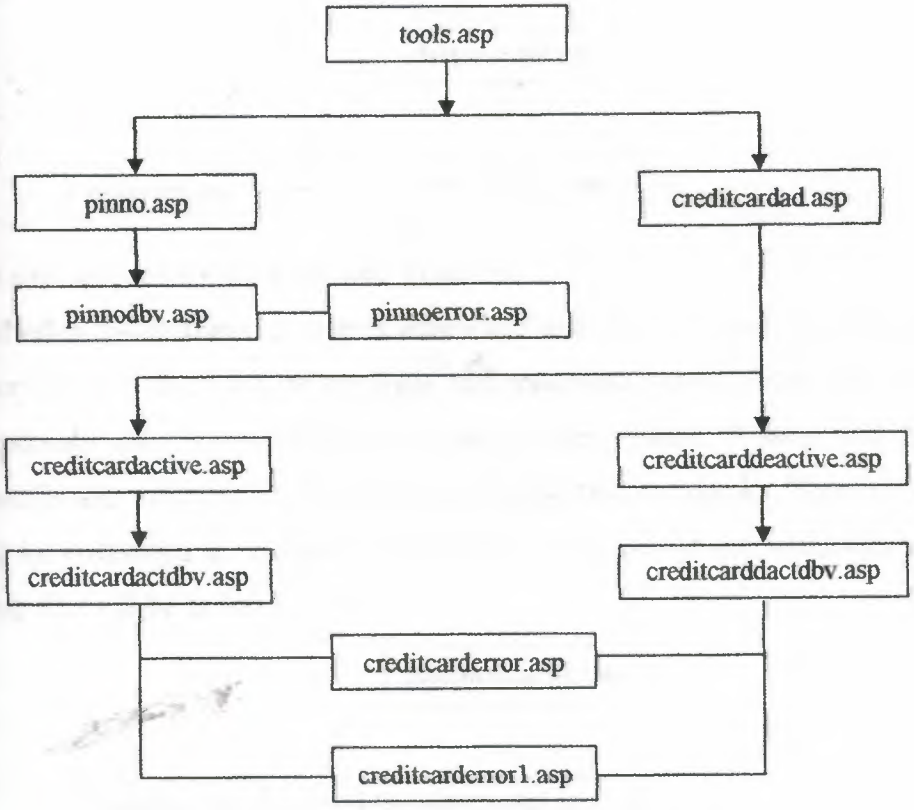
omer wants to get account information through fax, fax.asp file will be called. For stomer has to give account number, pin number and fax number. If customer give information then faxerror.asp file be called otherwise faxdbv.asp file, which will ustomer given information.



.8 Tools

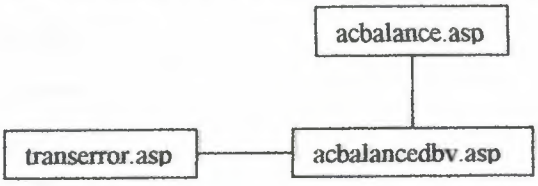
tomer want to change his pin number or if he wants to activate or deactivate credit he can. For this tools.asp file be called. Which then redirected to pinno.asp or card.asp. If user wants to change pin number then voice application will ask mer account number and old pin number and to perform requested task

dbv.asp will be called. If information provided by the user is wrong then he will cted to pinnoerror.asp. To activate credit card user will be directed to cardad.asp. System will ask with customer his account number, pin number and card number. To verify system will use creditcardactdbv or creditcarddactdbv.asp. ne wrong information will be given, then creditcarderror.asp or creditcarderror1.asp be called.



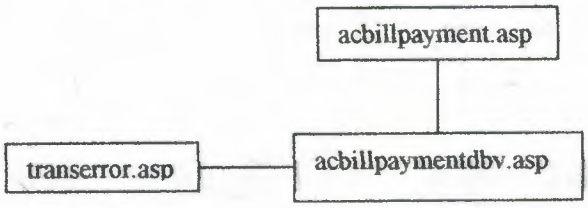
.6.9 Authentication Check Account Balance

heck whether the customer or user is authorized to check account balance is or not. his I am giving eight authentication checks. From which system will ask randomly To check about account balance system. System will first use acbalance.asp, which ask one of eight authentication checks. Then to verify it, system will be redirected to lance.asp. If user is not authenticated then transerror.asp file will be called.



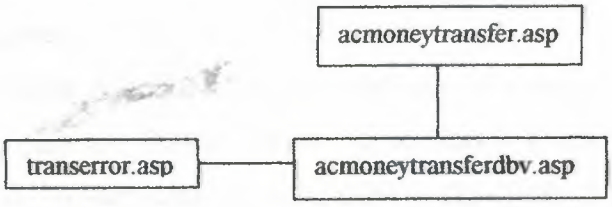
) Authentication Check Bill Payment

whether the customer or user is authorized to pay utility bills or not. For this I g eight authentication checks. From which system will ask randomly one. To out bill payment system. System will first use acbillpayment.asp, which will ask ight authentication checks. Then to verify it, system will be redirected to mentdbv.asp. If user is not authenticated then transerror.asp file will be called.



1 Authentication Check Money Transfer

whether the customer or user is authorized to transfer money from one account r or not. For this I am giving eight authentication checks. From which system randomly one. To check about money transfer system. System will first use transfer.asp, which will ask one of eight authentication checks. Then to verify it, vill be redirected to acmoneytransferdbv.asp. If user is not authenticated then r.asp file will be called.



Database

sed Microsoft Access 2002 to built database for Voice Interactive System for nsaction. Database includes:

Table Account

Table Bank

Table Credit

Table Currency

Table Customer

Table Fax

Table Greeting

Table News

Table Product

Table Stock

Table Survey

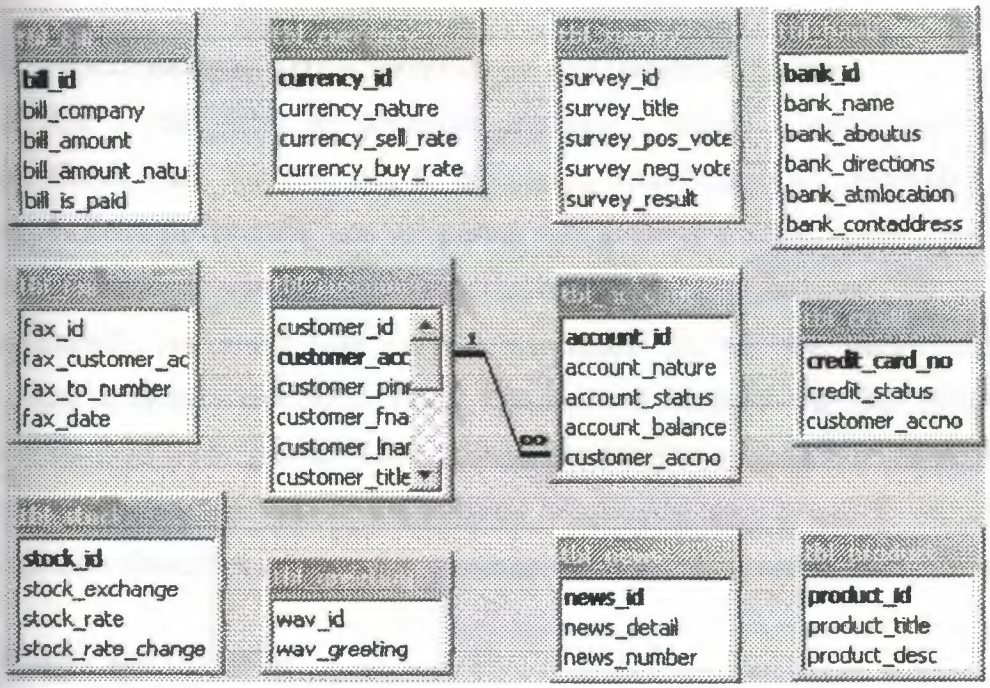


Figure 3.7 Tables

7.1 Description of Tables

Account Table Account holds information about customer account nature, account , account balance. **Table Bank** Table Bank holds information about bank nation, bank contact address, bank driving directions, bank ATM location. **Table** Table Bill shows the information about bill company, bill amount, bill amount nature whether bill is paid already or not. **Table Credit** stores the information about credit number and credit card status. **Table Currency** stores the information about ncy nature, currency sell rate and currency buy rate. Table Customer stores nation about customer account number, customer pin number, customer name, mer title, customer date of birth, customer birth month, customer birth year, mer social security number, customer zip code, customer life insurance number and mer license number. **Table Fax** stores information about customer account number, umber and fax date. **Table Greeting** stores recorded wave files which customer

Table News stores information about news detail. **Table Product** stores on about product title and product description. **Table Stock** stores information ock exchange, stock rate and change in stock rate. Table Survey stores on about survey title, survey positive votes, survey negative votes and survey

Functionality of Active Server Pages and Database System

file is responsible for obtaining user inputs, namely Account Number, Pin Credit Card Number, Bill Id etc passed from VoiceXML, retrieving stored ion from the database, and comparing data to verify a caller. During testing the s and database system, retrieving data from the database and verification of users cessfully performed with voice-based inputs. To test the functionality and ance of those Active Server Pages, I fetched and retrieved ASP files 10 times and d average fetching time. Table 3.14 summarizes the results of the test conducted. ough the average time of file retrieval is currently slow, it will be considerably d once those files are stored in the same server with VoiceXML files.

esting and QA phase

ie implementation of the voice application is complete, the first step is to unit-test ponents. This will ensure that any module level bugs are identified and fixed.

ehensive system testing can then begin to ensure that all reasonable/unreasonable rough the application behave as expected, bugs are fixed, and requirements have et. This is an iterative process involving the application developers and the system so after a few iterations a systematic and comprehensive system test pass of the pplication is performed, including regression tests, to ensure that old bugs do not ce. After the application passes the system tests, it is deployed "live" to an initial callers representative of the production user base. Usability reports of the tion are generated based on the feedback from these users. At the same time, end- performance testing and metrics data collection can be done. Based on the nance data and metrics, the voice application can be tuned/optimized before being

d. The test phase is also when any potential VUI breakdowns are identified. Some actors that need to be analyzed include:

Percentage of users who did not complete the test scenarios

Percentage of users who transferred to an operator

Unexpected user behavior

Points in the application where users experienced most difficulty

Effectiveness of error handling mechanisms

Time to complete typical transactions

User satisfaction surveys

gh the testing of voice applications is similar in concept to practices followed for other software, there is the added complexity that it is a manpower-intensive s and that usability studies must be generated. Automatic QA testing tools, such as ix CallMaster and load-generation tools, can help automate some of the system activities.

1 Identifying Recognition Problems

nost common cause of recognition problems is acoustic confusability among the ntly active phrases. For example, both Madison and Addison are US airports. Thus, potential user inputs to a travel application are highly confusable:

r: Flying from Madison
r: Flying from Addison

rrect the problem by:

Using a synonym for one of the terms. For example, if the system is confusing "no" and "new," you might be able to replace "new" with "recent," depending on the application's context.

Adding a word to one or more of the choices. For the Madison/Addison airport confusion, you could make states optional in the grammar for most cities, but require the state for low-traffic airports that have acoustic confusability with higher-traffic airports.

4.2 Identifying User Interface Breakdowns

Test phase is also where you will identify potential user interface breakdowns. Some ors you may want to analyze include:

- Percentage of users who did not successfully complete your test scenarios
- Percentage of users who transferred to a human operator
- Points in the application where users experienced the most difficulty
- Unexpected user behaviors
- Effectiveness of error recovery mechanisms
- Time to complete typical transactions
- Self-reported level of user satisfaction

first round of user testing typically reveals places where the system's response needs e rephrased to improve usability. For this reason, system responses should be left ble until after the first round of user testing.

Validation and Tuning Phase

validation and tuning phase consists of validating and optimizing prompts, dialogs, mmars, audio, pronunciations, and the VUI, and ensuring that the metrics specified in requirements document and design criteria are met. This is the stage before the cation is turned over for Beta phase field-testing with real and simulated users to ile actual usage data. The main tasks in the validation and tuning process involve aring transcribed actual user utterances with the application's prompt and dialog n, to determine the rate of success of grammar and speech recognition. There are tools, such as Nuance V-Optimizer, that help in the optimization and tuning ess and generate useful analysis reports. Once the tasks in this phase are done and the application deemed to be of production quality, the application is deployed for nercial usage.

.1 Metrics

collection of metrics provides data used to refine the VUI and judge the quality and ility of the voice application. The following list provides some suggested metrics:

1.1 Speech Recognition Rate

peech recognition rate measures the number of successful and unsuccessful speech nition results returned by the ASR engine. The metric is dependent on the design of rammar rules. If this metric is tied to the number of "no-match" events, then bility of utterance records helps to peg the recognition rates with the grammar n. Also, if this metric is tied with the number of "no-input" events with sufficient out" values, then it might be used to gauge how well the user understood the pt.

1.2 Time Between Prompt and Response

uring the amount of time between the end of prompt playback and the user's nse helps to characterize the complexity of the prompt interface. New users would lly allow prompts to play to completion, whereas repeat users would typically in if the prompts were well designed. (Barge-in is what allows the user to interrupt aying of a prompt either through spoken or DTMF input.)

1.3 Barge-In Rate

ent prompts in a voice application may have varying barge-in rates. Barge-in rates, er with call completion rates, can help the application designers analyze whether er interacted fully with the application or disconnected because of frustration. This o distinguish whether the prompt was poorly designed or if users are familiar with mpt information.

.4 Call Duration

se of the nature of the voice user interface, the length of call sessions varies n different users and depends on the user's perspective of the usefulness of the application. The call duration depends on a number of factors such as number of ts, the time to retrieve data, and so on. The call duration tied with the call etion rates can reveal user satisfaction with the voice application.

5 Call Completion Rate

etric measures the rate at which users disconnect from the voice application after ve initiated a call session. Disconnect may be a logical disconnect after the user ained desired information traversing through normal call flow, or it may be anted" from the application's perspective.

6 Number of Prompts

mber of prompts metric is the global count of the number of prompts that a user s through a call session. This metric can be used to characterize application flow enchmark a typical/ideal call flow vs. actual call flows.

7 Prompt Request to Prompt Playback Delay

etric helps to characterize the efficiency of the TTS engine and audio resources. ptable delays for playback may result in unsatisfactory user experiences. ques such as "caching" may be used to tune the prompt playback time.

etric as a separate entity does not yield enough information to be of much use, but combinations of the metrics may give a clearer picture of the average usability nce. Careful analysis of metrics may help in the fine tuning of the application and ing the usability of the voice user interface. Platform level metrics such as call /failure rates, file download rates, and network latency help to supplement the ntered metrics.

eployment and Maintenance Phase

he voice application has been designed, implemented, and tested, it is ready for the production launch. At this stage, the telephone numbers for application access and visioning of the subscriber base is done. After the production launch, for the first eeks, intense collection and analysis of production data, such as accuracy reports, ne for final accuracy tuning and VUI tweaks. After the final round of tweaks, the ation is considered to be in maintenance mode. Any fixes or new enhancements to plication or related components are done periodically while adhering to a ction and maintenance plan. Key tasks that are performed during the maintenance include scheduling ongoing audio/pronunciation updates and periodically tuning

mance to address new performance/usability issues that may arise as a result of the ation being commercially deployed over a period of time.

1 Deploy a VoiceXML Application

A system administrator uses a Web application server program (such as WebSphere Application Server) to configure and manage a Web server.

The developer publishes the VoiceXML application (including VoiceXML pages, grammar files, any prerecorded audio files, and any server-side logic) to the Web server.

A telephony expert configures the telephony infrastructure, for the applicable deployment platform.

The system administrator uses the deployment platform to configure, deploy, monitor, and manage a dedicated WebSphere Voice Server system.

The developer uses a real telephone to test the VoiceXML application running on WebSphere Voice Server.

2 Access the Deployed Application

your voice applications are deployed, users simply dial the telephone number that rovide and are connected to the corresponding voice application. The figure 3.7 a flow chart of a typical call.

A user dials the telephone number you provide. WebSphere Voice Server answers the call and executes the application referenced by the dialed phone number.

WebSphere Voice Server plays a greeting to the caller and prompts the caller to indicate what information he or she wants.

The application can use prerecorded greetings and prompts, or the application can have the greeting or prompt synthesized from text using the text-to-speech engine.

If the application supports barge-in, the caller can interrupt the prompt if he or she already knows what to do.

The application waits for the caller's response for a set period of time.

The caller can respond either by speaking or by pressing one or more keys on a DTMF telephone keypad, depending on the types of responses expected by the application.

the response does not match the criteria defined by the application (such as the ecific word, phrase, or digits), the voice application can prompt the caller to ter the response again, using the same or different wording.

the waiting period has elapsed and the caller has not responded, the application n prompt the caller again, using the same or different wording.

ication takes whatever action is appropriate to the caller's response. For he application might:

date information in a database

trieve information from a database and speak it to the caller

ore or retrieve a voice message

unch another application

y a help message

ter taking action, the application prompts the caller with what to do next.

or the application can terminate the call. For example:

e caller can terminate the interaction at any moment by hanging up. ebSphere Voice Server can detect if the caller hangs up and can then disconnect lf.

he application permits it, the caller can use a command to indicate explicitly t the interaction is over (for example, by saying "Exit").

he application has finished running, it can play a closing message and then connect.

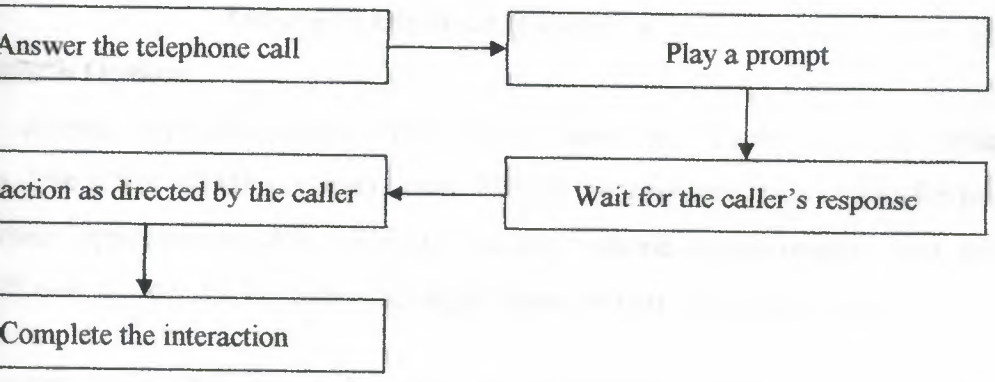


Figure 3.8 Typical Call Flow

ment Platforms

ion is ready to deploy in a telephony environment, system administrators the deployment platforms in table 3.15 to configure, deploy, monitor, and plications on a dedicated WebSphere Voice Server system.

	Description
Server – Use with Response for AIX	A connection environment that exploits the telephony functions and programming methodologies provided by the IBM WebSphere Voice Response for AIX version 3.1. WebSphere Voice Response's telephone network connection provides the audio channels for the VoiceXML browser.
Server – Use with Response for Windows	A connection environment that exploits the telephony functions and programming methodologies provided by IBM WebSphere Voice Response for Windows. WebSphere Voice Response for Windows. WebSphere Voice Response's telephone network connection provides the audio channels for the VoiceXML browser.
Voice Server – Use with Cisco	WebSphere Voice Server is installed in a Cisco – based environment. This environment is a telephony gateway connected to a PBX/PSTN/GSM telephone network that routes audio to and from the IP network.

Table 3.15 Deployment Platform

n Options

eral platform options for development and deployment of voice se a VoiceXML service bureau. This is the most common option for less applications with relatively modest volume requirements. You will ed to host the logic for your application on your own equipment.

n-VoiceXML service bureau. You will have to pay them to develop your on. There are fewer of these available as VoiceXML takes over as the

industry standard, but they may be less expensive, and can provide you with some of the features missing from VoiceXML.

2. Purchase hardware and build your own non-VoiceXML application. This is by far the most difficult path to pursue, and will require significant specialized training in telephony, and speech recognition (if your application requires it).
3. Purchase a VoiceXML system to reside with your equipment. You will probably still treat it largely as a black box, and may need assistance ordering phone lines and connecting the system to the phone network.

Authentication Issue

bank transactions caller authentication should be consider. I have implemented caller entication by using eight attributes which are:

- Date of Birth
- Birth Month
- Birth Year
- Social Security Number
- Zip Code
- Secret Number
- Life Insurance Number
- License Number

em will first prompt one of these questions. Upon verification from database action will continue. Now the technology which is more appropriate to implement rity for account is to store voiceprints of customer in database. First time during mer registration, system will ask with him, his voiceprints. Next time whenever mer will use voice application system will ask his voice sample and then will match voiceprints which are in database using

Caller Authentication Flow

he identification of caller takes place. In which system will ask account number and umber and verify from the database. Recognized caller will then asked by the on which can be date of birth, birth month, birth year, zip code, social security er, life insurance number, secret number or license number.

n authenticated caller is prompted to speak a phrase for a voice sample. System will e this voice sample in database. So when ever next time customer will come. System do biometric verification using voiceprint authentication technology.

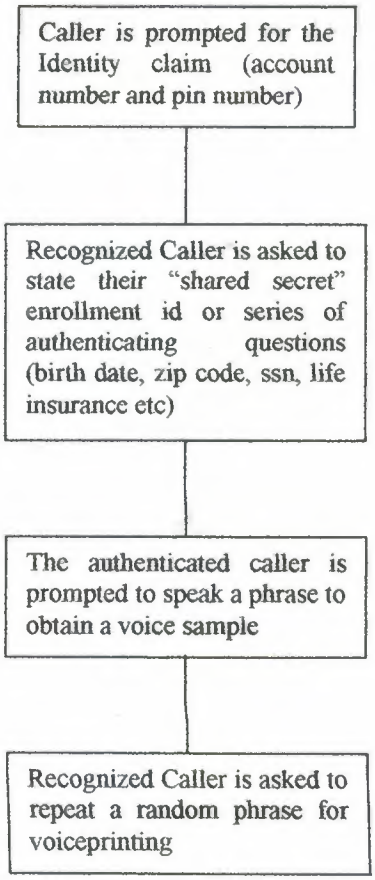


Figure 3.9 Caller Authentication Flow

Summary

ceXML applications are designed to follow a certain pattern of questions and answers rder. Moreover the presentation of transient voice content is generally event driven chronological. VoiceXML is revolutionizing speech-enabled access to Web-based rmation by combining the power of the Internet with the convenience and practicality he telephone. Industry leaders in voice, telephony and Internet technologies support ceXML. It is an open standard that gives businesses the power to deliver powerful sion-critical services quickly and cost-effectively. Businesses can build upon existing

et standards and best practices to deliver a complete voice solution to their ners by integrating VoiceXML with their existing Web infrastructure.

CHAPTER 4

SIMULATION OF VOICE INTERACTIVE SYSTEM FOR BANK TRANSACTION

1 Overview

he WebSphere Voice Server SDK provides support for bringing VoiceXML to Web plication development activities - in essence, providing a spoken equivalent to visual owsing. The SDK uses the workstation's speakers to play audio output. You can input ta using the workstation's microphone, prerecorded audio files, or the WebSphere oice Server SDK's DTMF Simulator (to simulate any telephone key input). The SDK so supports text mode and automated testing.

2 Desktop Simulation

he WebSphere Voice Server SDK includes features to facilitate desktop simulation of iceXML applications.

2.1 Simulating DTMF Input

ou can simulate DTMF input on your desktop by using the DTMF Simulator, which is rted automatically by the VoiceXML browser unless you specify the Java system operty - Dvxml.gui=false.

hen the DTMF GUI is enabled, you can simulate a telephone key press event by essing the corresponding key on the computer keyboard or clicking on the rresponding button on the DTMF Simulator GUI.

2.2 Running Text Mode or Automated Tests

you do not have a microphone on your desktop workstation, you can still invoke the iceXML browser using the vstext script:

IBMVS\bin\vstext_en_US" URL

is script starts the VoiceXML browser in text mode. The VoiceXML browser writes mpts and other output as text in the window from which you started the VoiceXML wser, and accepts input from your keyboard or the DTMF Simulator GUI.

ank Information

trieve information about bank. Caller can say,

Please tell me about City bank

Tell me About City bank

About Bank

What is City Bank

About City Bank

SP page **about.asp** will fetch data from database table **tbl_bank**. Below is the lation using IBM WebSphere Voice Server SDK Browser which is running in audio e. User can interrupt system any time by saying "Exit" or "Home". By saying "Exit" will disconnect and by saying "Home", user will be prompted again by main menu ons.

Figure 4.1. Bank Information

ank Products

ow about the bank products, customer can ask,

What products City bank is offering

What are new products

What products they have

SP file **products.asp** will be called. This ASP file will prompt,

Car Financing

City Loans

Human Resources

has to select one of them. For example, if user says "Car Financing", **services.asp** will fetch the requested product detail. Below is the simulation using IBM Sphere Voice Server SDK Browser which is running in audio mode.

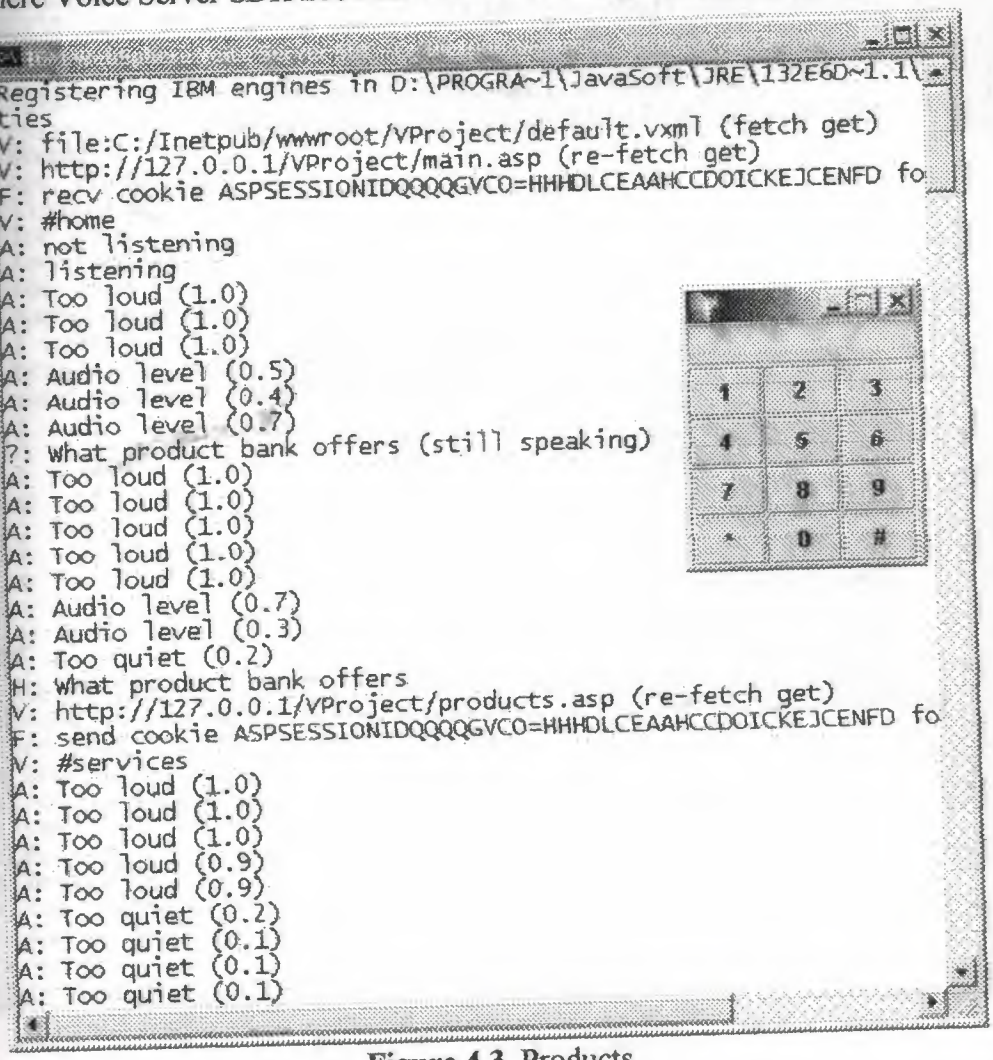


Figure 4.3. Products

nterrupt system any time by saying "Exit" or "Home". By saying "Exit" call nnect and by saying "Home", user will be prompted again by main menu

nt Balance

er wants to check his account balance, he can say,

vant to check my account balance

hat is my account balance

ow much money left in my account

vill call the **accbalance.asp** file. Which will ask account number and pin

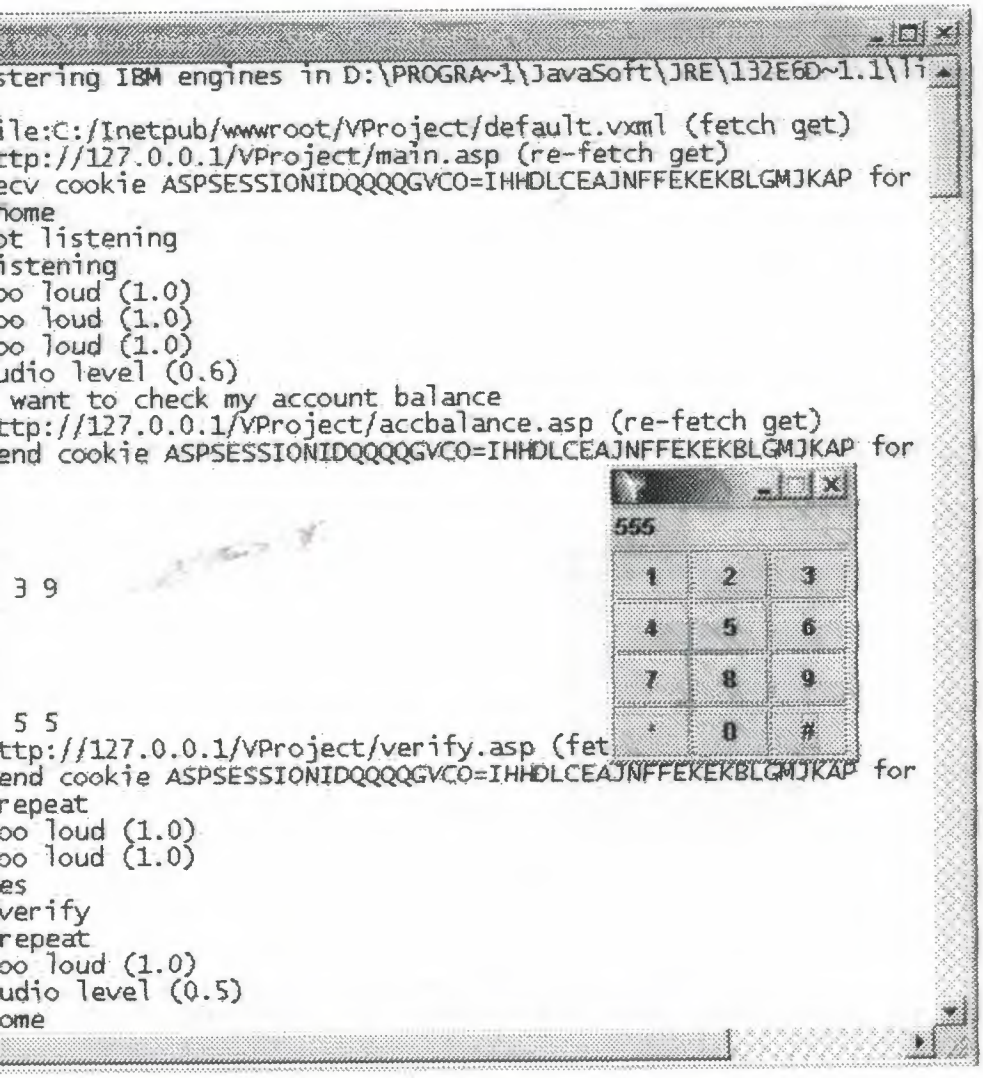


Figure 4.4. Account Balance

the verification and retrieval of account information **verify.asp** file will be call. If ller will give wrong account number and/or pin number, **transerror.asp** file will be lled. User can interrupt system any time by saying "Exit" or "Home". By saying "Exit" ll will disconnect and by saying "Home", user will be prompted again by main menu ptions.

.7 Money Transfer

f customer wants to transfer money, he can say

- I want to transfer my money
- I want to transfer money from one account to another account
- How can I transfer a money

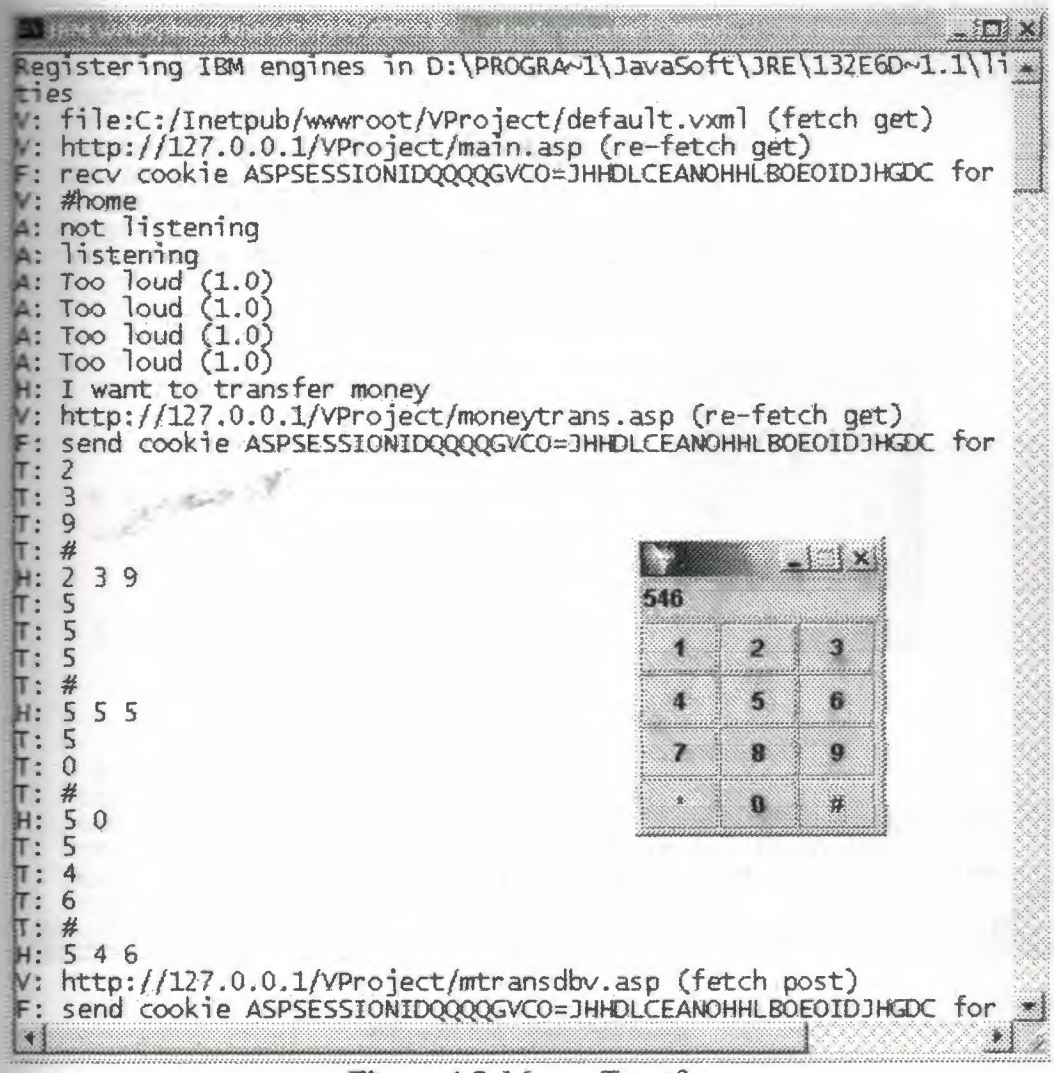


Figure 4.5. Money Transfer

system will fetch **moneytrans.asp** file. Which will ask account number, pin , how much money customer wants to transfer and the account number where he o transfer money. After this for verification and result **mtransdbv.asp** file will be If customer didn't give correct information then **transferror.asp**, **rror1.asp** or **transferror2.asp** file can be called. User can interrupt system any saying "Exit" or "Home". By saying "Exit" call will disconnect and by saying ", user will be prompted again by main menu options.

Payment

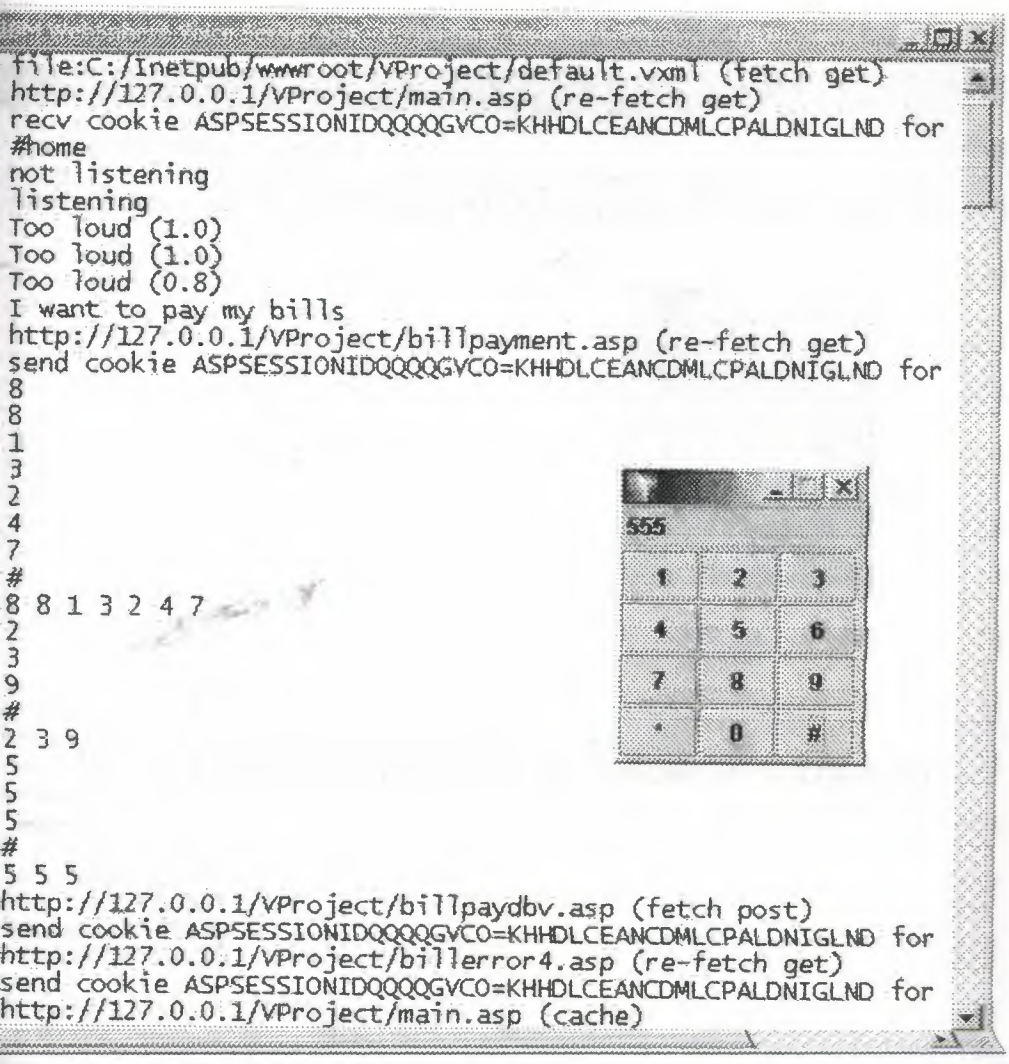


Figure 4.6. Bill Payment

tomer wants to pay bill(s), he can ask,

- I want to pay my bill
- How can I pay my bills

pay bill customer will give bill number, account number and pin number. After this **aydbv.asp** file will do the verification and perform the requested task. If bill ber, account number or pin number will not be correct then **billerror1.asp**, rror2.asp, **billerror3.asp**, **billerror4.asp** can be called. User can interrupt system time by saying "Exit" or "Home". By saying "Exit" call will disconnect and by g "Home", user will be prompted again by main menu options.

ax

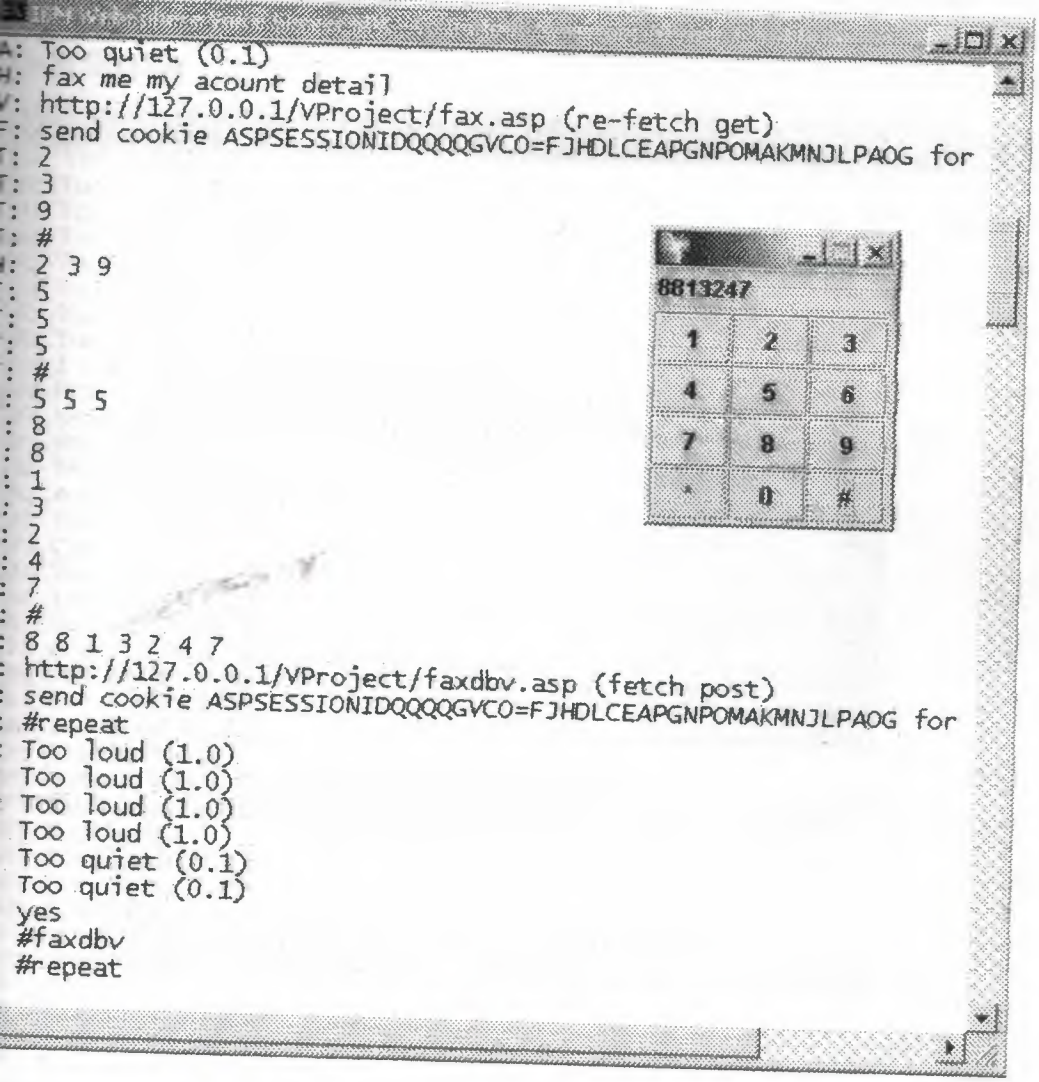


Figure 4.7. Fax

ustomer wants to know his account detail through fax, he can ask,

- Please fax me my account detail
- I want to get my account detail through fax
- Can I get account detail through fax

this **fax.asp** file will do the requested task and **faxdbv.asp** will store customer ormation in database table **tbl_fax**. Customer need to give his account number, pin nber and the fax number, where he wants to get account detail. If customer will not e correct account number or pin number **faxerror.asp** file will be called. User can errupt system any time by saying "Exit" or "Home". By saying "Exit" call will connect and by saying "Home", user will be prompted again by main menu options.

0 Tools Section

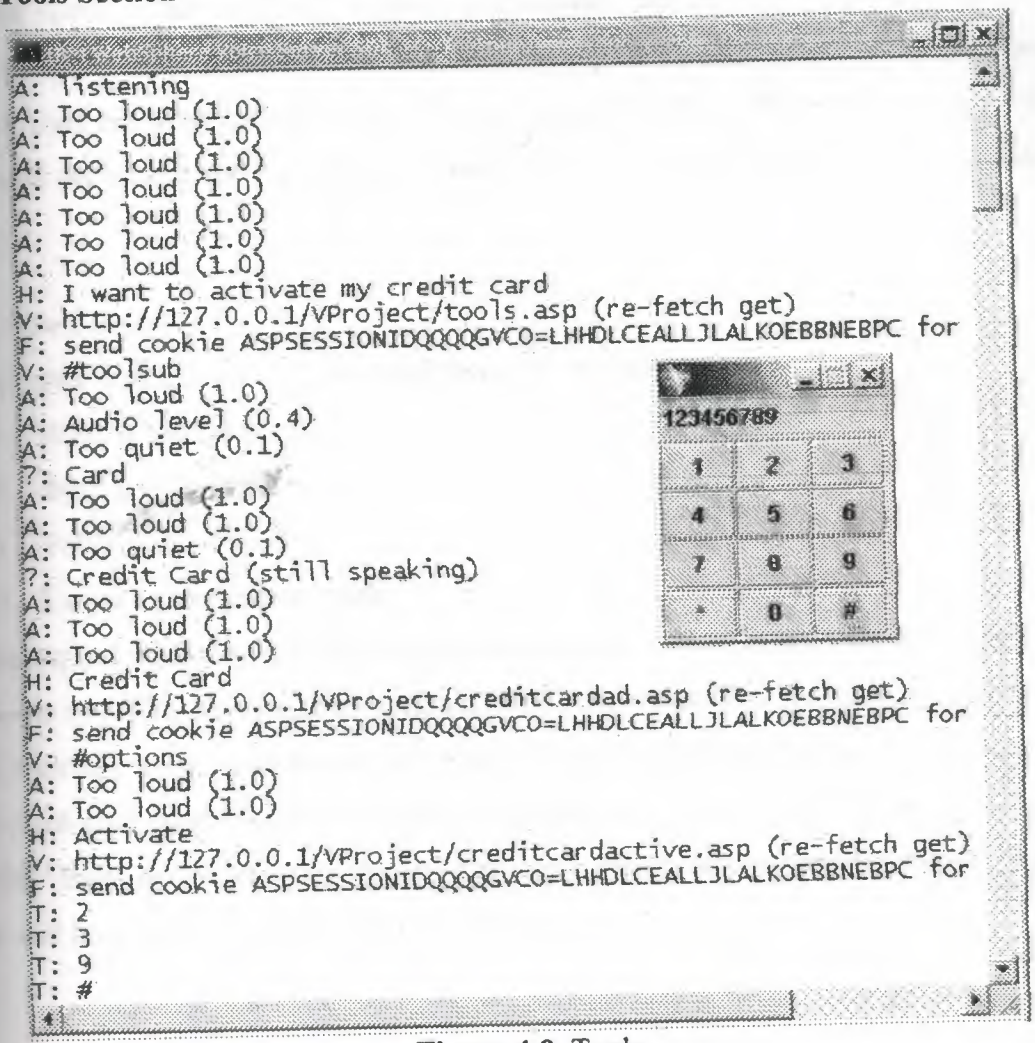


Figure 4.8. Tools

omer wants to Activate/Deactivate Credit Card or want to change pin number, he y,

I want to change my account pin number

Can I change my account pin number

I want to activate/deactivate credit card

will call **tools.asp** file. It will prompt user by the Pin Number and Credit Card. User o select one option. If user will say "Pin Number", **pinno.asp** file be called. It will account number, old pin number, and new pin number. Upon confirmation **odbv.asp** file will do the requested task. If user didn't give correct account number in number **pinerror.asp** file will be called. To activate or deactivate credit card omer can choose "Credit Card" from **tools.asp**, this will call **creditcardad.asp**. For he needs to give account number, pin number and credit card number. If any of rmation is not correct then **creditcarderror.asp** or **creditcarderror1.asp** file can be d. User can interrupt system any time by saying "Exit" or "Home". By saying "Exit" will disconnect and by saying "Home", user will be prompted again by main menu ons.

l Stock Rates

ustomer wants to know about stock rates, he can say

- What is market situation
- What are the stock rates
- Tell me about stocks
- I want to know about trade

on requesting the latest stock rates **stockrates.asp** file will prompt for stock exchanges th the user like Karachi, Newyork and Doe Jones. Customer will say one of them. It ll take to the file **srateks.asp**, **srateny.asp** and **sratedj.asp** respectively. These files ll fetch data from the database table **tbl_stock**. User can interrupt system any time by ying "Exit" or "Home". By saying "Exit" call will disconnect and by saying "Home", er will be prompted again by main menu options.

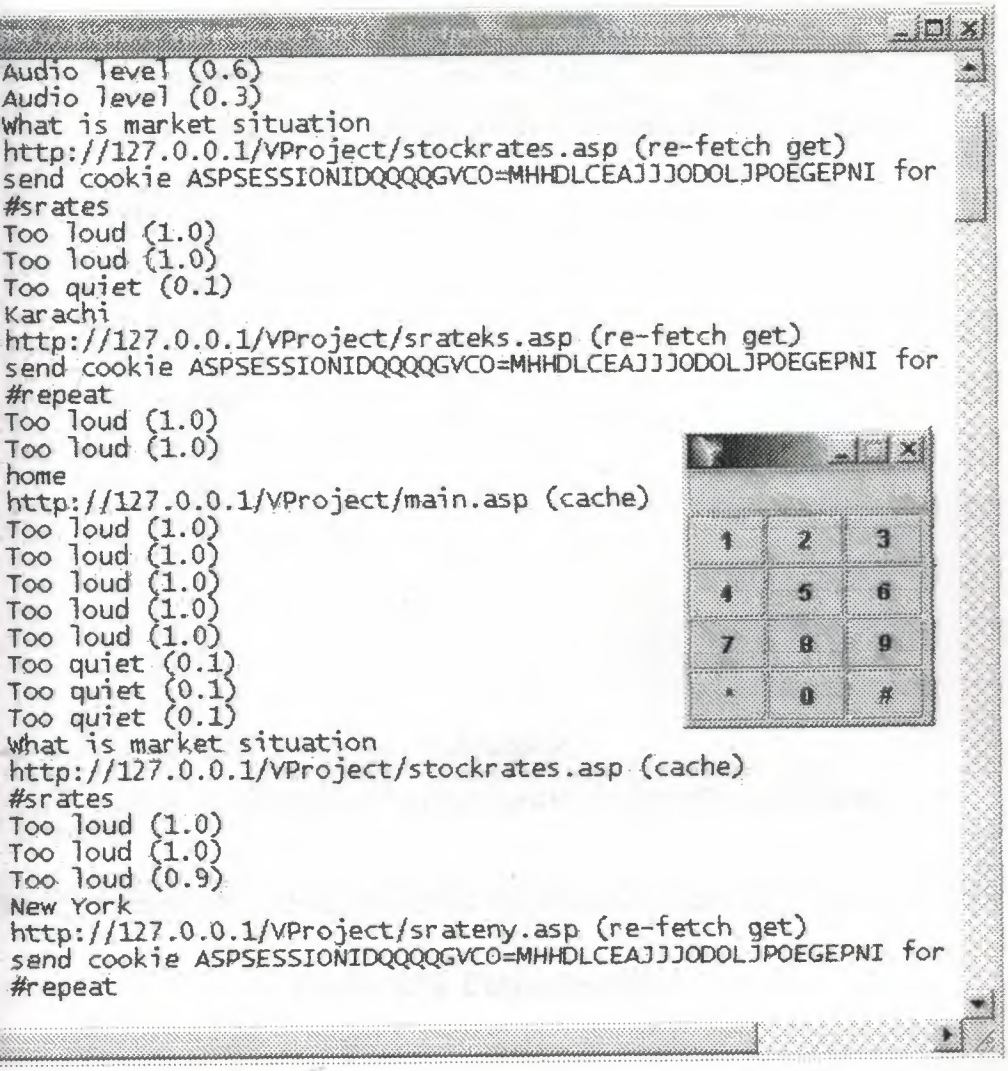


Figure 4.9. Stock Rates

enter Operator

omer didn't get satisfy then he can talk with bank representative by transferring his

r this he can say

Please transfer my call to center operator

Can I talk to city bank representative/person

I didn't find solution please transfer my call to city bank representative/customer

care center

ill call **ctransfer.asp**, which will transfer call to customer care center.

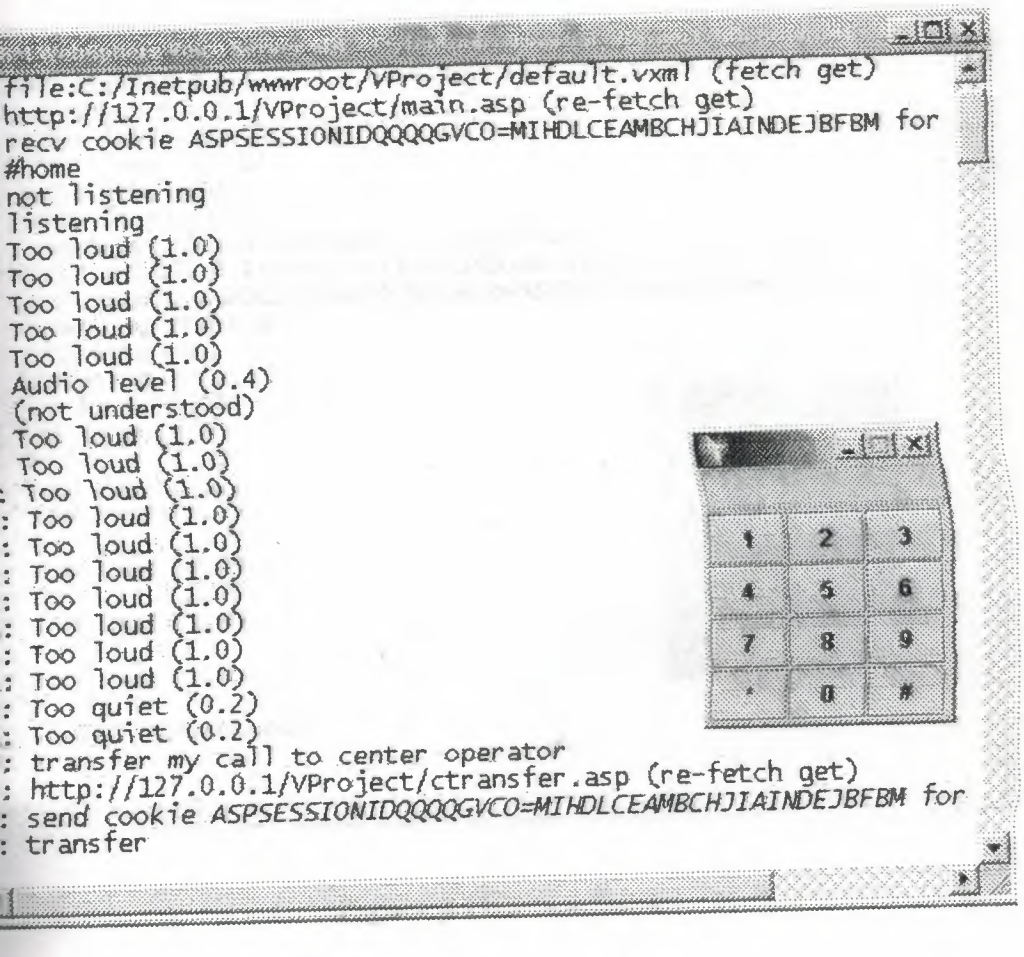


Figure 4.10. Center Operator

Bank Guest book

tomer wants to record his message in guest book, he can ask

I want to record message in guest book

ll call **guestbook.asp**. Here customer can record message or greeting and upon cation **guestbookdbv.asp** will store recorded message to database table **greeting** as a wave file. User can interrupt system any time by saying "Exit" or me". By saying "Exit" call will disconnect and by saying "Home", user will be pted again by main menu options.

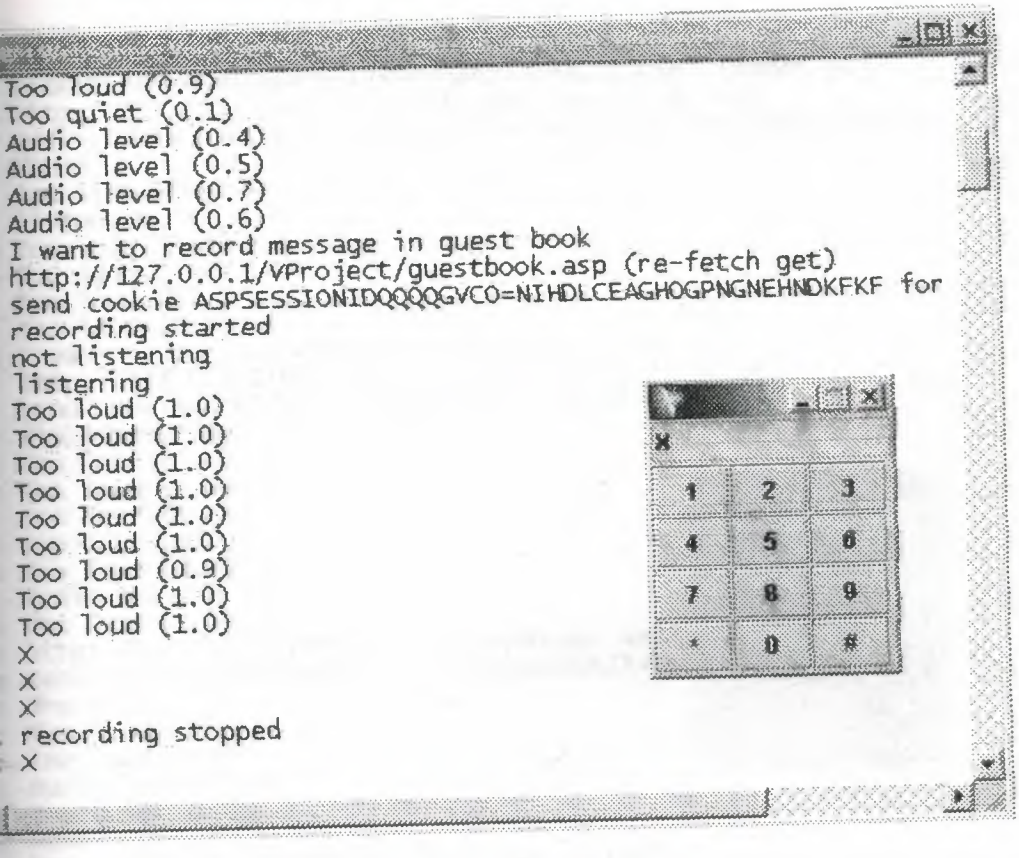


Figure 4.11. Guest Book

Bank Survey

k is conducting a survey, then customer can poll his vote by asking

I want to poll my vote in bank survey

What is new at survey

Poll my vote

I want to give my opinion in survey

l call **survey.asp** file. If no survey question is available then **surveyerror.asp** file e called. To poll a vote customer can say "Yes" or "No". If customer will say "yes" **surveydbvyes.asp** file will be called, in other case **surveydbvno.asp**. User can upt system any time by saying "Exit" or "Home". By saying "Exit" call will nnect and by saying "Home", user will be prompted again by main menu options. lation is as under:

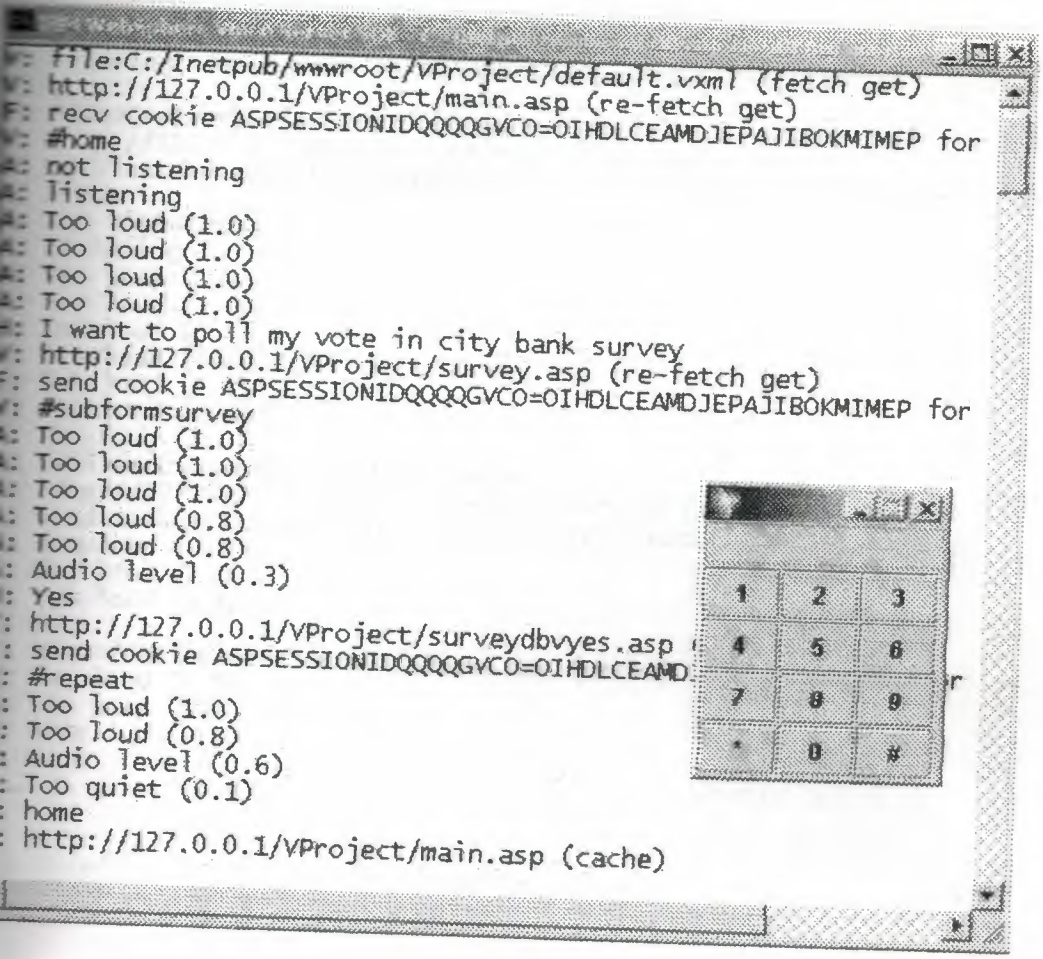


Figure 4.12. Bank Survey

ank ATM

mer wants to know about location of bank ATM, he can ask,

Where is city bank ATM location

Tell me city bank ATM location

ake to **atmlocation.asp** which will fetch the requested data from database table **k**. User can interrupt system any time by saying "Exit" or "Home". By saying all will disconnect and by saying "Home", user will be prompted again by main otions. Simulation is as under:

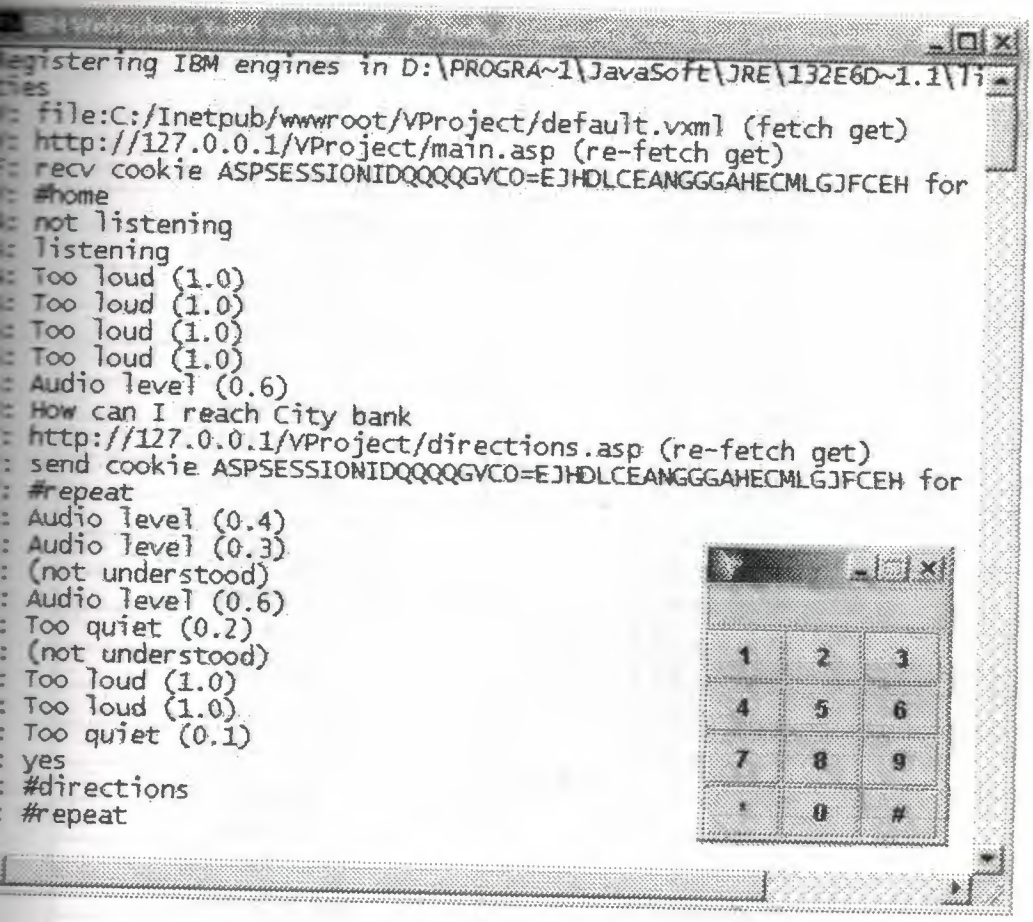


Figure 4.15. Driving Directions

verage Time of File Retrieval

am giving average time of file retrieval. So from it can be shown that how much stomer has to wait for specific job.

Table 4.1. The Average Time of File Retrieval

Name of an ASP File	Average Time of File Retrieval
Aboutus.asp	1.18 seconds
news.asp	1.36 seconds
products.asp	1.25 seconds
accbalance.asp	1.49 seconds
moneytrans.asp	1.21 seconds
verify.asp	2.08 seconds
tranerror.asp	0.98 seconds
mtransdbv.asp	2.46 seconds
transferror1.asp	0.98 seconds
transferror2.asp	0.98 seconds
Transfer	0.98 seconds

stockrates.asp	1.15 seconds
srateks.asp	1.23 seconds
srateny.asp	1.23 seconds
sratedj.asp	1.23 seconds
billpayment.asp	1.28 seconds
billpaydbv.asp	1.96 seconds
billerror1.asp	0.87 seconds
billerror2.asp	0.87 seconds
billerror3.asp	0.87 seconds
billerror4.asp	0.87 seconds
atmlocation.asp	2.03 seconds
contaddress.asp	2.11 seconds
survey.asp	1.65 seconds
surveyerror.asp	1.62 seconds
surveydbvyes.asp	1.24 seconds
surveydbvno.asp	1.24 seconds
ctransfer.asp	1.01 seconds
fax.asp	1.55 seconds
faxdbv.asp	1.73 seconds
Faxerror.asp	0.86 seconds
tools.asp	1.23 seconds
pinno.asp	1.11 seconds
pinnodbv.asp	1.52 seconds
pinnoerror.asp	0.82 seconds
creditcardad.asp	1.11 seconds
creditcardactive.asp	1.24 seconds
Creditcardactdbv.asp	1.64 seconds
creditcarddactdbv.asp	1.58 seconds
creditcarddeactive.asp	1.24 seconds
creditcarderror.asp	0.74 seconds
creditcarderror1.asp	0.81 seconds
guestbook.asp	1.53 seconds

Results of Voice Interactive System for Bank Transactions

running Voice Interactive System for bank 200 times,

ntage of users who didn't complete their task(s)	0%
ntage of users who transferred to human operator	23%
age time required to check account balance	20sec
age time required to change a pin number	35sec
age time required to activate/deactivate credit card	65sec
age time required to pay a bill	18sec

CONCLUSION

e Web technology is rapidly improving and widely spreading, it is a matter of time people will look for a more convenient way to access the Web and web-based cation. Developing Interactive Voice Response applications that employ Automated ch Recognition technology and VoiceXML to realize the effective use of the Web demonstrate that voice interaction holds the promise of natural dialogs with Web- l services.

main objective of this project was to develop an Interactive Voice Response cation with Automated Speech Recognition and the web technology. The Voice ctive System for Bank is developed using VoiceXML along with Active Server for database integration. Voice application and the database system tested ssfully, performing the following required tasks: recognizing voice-based inputs, ding synthetic outputs, and storing and retrieving data from the database.

Interactive System for Bank Transaction tested successfully and test results shows 7% customers are satisfied and only 23% customers transferred to human operator tra bank information. Average time required to complete task is 33.6 seconds which hat in a less than one minute customer can finish his job. So the application is quite as in terms of time and resource saving.

Interactive System for Bank Transaction can increase customer satisfaction and y, provide quick access to the information and provides a more personalized ner experience that is similar to live customer support.

REFERENCES

ooks

1. H. M. Deitel.; P. J. Deitel.; T. R. Nieto. (2002). Internet & World Wide Web, Prentice Hall, New Jersey: p. 1277-1288
2. Profession WAP. (2000) Wrox Press: p. 655-679.
3. Michael Morrison, et al (1999). XML. Techmedia Sams: p. 883-902

nline Articles

4. VoiceXML Forum. VoiceXML Specification 1.00. March:2000 from the World Wide Web: http://www.alphaworks.ibm.com
5. VoiceXML Programmers Guide version 3.1. IBM Corporation. Sept.:2002 from the World Wide Web: http://www.alphaworks.ibm.com
6. Voice Site Developer's Guide version 1.0. (2000). Nuance Communications from the World Wide Web: http://www.nuance.com
7. VoiceXML Tutorial version 2.1. (2002). BeVocal, Inc. from the World Wide Web: http://cafe.bevocal.com
8. VoiceXML Programmers Guide. (2003) BeVocal, Inc. from the World Wide Web: http://cafe.bevocal.com
9. Priya Kothari. (2002). Voice Portal Solution White Paper [On-line], Wipro Infotech.
10. Gilda Raczkowski. (2002). Voice Portals [On-line], from the World Wide Web: http://www.dash30.com
11. Peter J. Danlelsen, The Promise of a Voice-Enabled Web [On-line], from the World Wide Web: http://www.lucent.com
12. An Introduction to Voice Portals & Voice Technology. (2001) from the World Wide Web: http://www.differentia.com
13. The Business Advantages of VoiceXML. (2001). VoiceGenie Technologies Inc. from the World Wide Web: http://www.voicegenie.com
14. Developing Dynamic VoiceXML Applications. IBM Corporation. From the World Wide Web: http://www.ibm.com/developerworks

Voice Application Middleware. Voxify. From the World Wide Web: http://www.voxify.com

VoiceXML Primer. (2001) Telera. From the World Wide Web: http://www.telera.com

Steve Ihnen, (2000) VoiceXML Overview and System Architecture [On-line], from the World Wide Web: http://www.voice-applications.com

Anon. VoiceXML Planet, (2002) Jupitermedia Corporation [On-line], from the World Wide Web: http://www.voicexmlplanet.com

DeVXchange Tools and Resources from the World Wide Web: http://www.telera.com/developers.html

APPENDIX A

Voice Interactive System for Bank Transaction - Program Listing

LT.VXML

```
ersion="1.0" encoding="iso-8859-1"?>
TYPE vxml PUBLIC "vxml" "">
ersion="1.0">
l ="default">
 <block>
        <goto next = "http://127.0.0.1/VProject/main.asp"/>
 </block>
```

ASP

```
ersion="1.0" encoding="iso-8859-1"?>
TYPE vxml PUBLIC "vxml" "">
ersion="1.0">
ext = "#home">
        <grammar>home</grammar>
 </link>
 <link next = "#end">
        <grammar>exit</grammar>
 </link>
 <var name = "currentoption" expr = "'home'"/>
 <form>
        <block>
        <emp>Welcome</emp> to the City Bank, Islaamaabaad Branch.
        To exit any time say exit. To go to the home page any time
        say home.
        </block>
        <subdialog src = "#home"/>
 </form>
 <menu id = "home">
        <prompt count = "1" timeout = "10s">
                You have just entered the City voice interactive system
                for bank transaction. Please make a selection by
                speaking one of the following options:
                <break msecs = "1000" />
                <enumerate/>
        </prompt>
        <prompt count = "2">
                Please say one of the following.
                <break msecs = "2000"/>
                <enumerate/>
        </prompt>
        <choice next = "/VProject/aboutus.asp#aboutus">About City Bank
        <grammar type="application/x-jsgf">
                [Please] [Tell me] About [City] bank |
                [What is] City [Bank]
 </grammar>
```

```
</choice>
<choice next = "/VProject/news.asp#news">Current News
<grammar type="application/x-jsgf">
        What is [current] news |
        Tell me news [at city bank] |
        Tell me news at bank |
        What is new
</grammar></choice>
<choice next = "/VProject/products.asp#products">Products
<grammar type="application/x-jsgf">
        What (product|products) [city] bank (is offering|offers) |
        What (is|are) new (product|products) |
        What is inside products |
        What products they have
</grammar></choice>
<choice next = "/VProject/accbalance.asp">Account balance
<grammar type="application/x-jsgf">
        I want to check [my] [account] [balance]|
        What is [my] [account] balance |
        How much [money] left in [my] account
</grammar></choice>
<choice next = "/VProject/moneytrans.asp#moneytrans">Money Transfer
<grammar type="application/x-jsgf">
        I want to transfer [my] money |
        I want to transfer money from [one] account to another [account] |
        [How] can I transfer [a] money
</grammar></choice>
<choice next = "/VProject/billpayment.asp#billpayment">Bill Payment
<grammar type="application/x-jsgf">
        (I want to|how can I) (pay|deposit) [my] (bill|bills)
</grammar></choice>
<choice next = "/VProject/fax.asp#fax">Fax
<grammar type="application/x-jsgf">
                [Please] fax [me] [my] [account] [detail] [please] |
                I want to get [my] account detail [through fax] |
                Can I get [my] account detail [through fax]
</grammar>
</choice>
<choice next = "/VProject/tools.asp#tools">Manager
        <grammar type="application/x-jsgf">
                [I] want to change [my] [account] pin [number] |
                Can I change [my] [account] pin [number] |
                I want to (activate|deactivate) [my] [credit] card |
                What is [inside] Manager |
                What manager does |
                How [can] I use Manager
        </grammar>
</choice>
<choice next = "/VProject/stockrates.asp#stockrates">Trade Stocks
<grammar type="application/x-jsgf">
        What is market [situation] |
        What are [the] stock rates |
        Tell me about [Trade] stocks |
        I want to know about [trade] stocks
</grammar></choice>
```

```
<choice next = "/VProject/ctransfer.asp#ctransfer">Center Operator
        <grammar type="application/x-jsgf">
                [Please] transfer my call to [center] [operator] |
                I want to talk to customer [care] center |
                Can I talk to [city] [bank] (representative|person) |
                I did not find [my] solution please transfer [my] [call] to [city] bank
                (representative|person|customer care center)
        </grammar>
</choice>
<choice next = "/VProject/guestbook.asp#guestbook">Guest Book
<grammar type="application/x-jsgf">
        What is [guest] book |
        I want to record [message] [in guest book]
</grammar>
</choice>
<choice next = "/VProject/survey.asp#survey">Survey
        <grammar type="application/x-jsgf">
                I want to poll my vote in [city] [bank] survey |
                What is new at survey |
                What is city bank survey |
                Poll my vote |
                I want to give my opinion in survey
        </grammar>
</choice>
<choice next = "/VProject/atmlocation.asp#atmlocation">ATM
<grammar type="application/x-jsgf">
        (Where|what) is [city] bank A T M [location] |
        Tell me [City] [bank] A T M [location]
</grammar></choice>
<choice next = "/VProject/contaddress.asp#contaddress">Contact Address
        <grammar type="application/x-jsgf">
                what is contact address of [city] bank |
                Tell me [city] [bank] [contact] address |
                What is the mailing address of [city] bank |
                Where is [city] bank
        </grammar>
</choice>
<choice next = "/VProject/directions.asp#directions">Driving directions
<grammar type="application/x-jsgf">
        Where [City] bank is [located] |
        Tell [me] driving directions |
        How can I reach [City] bank
</grammar></choice>

rm id = "end">
    <block>
        Thank you for visiting City Bank voice interactive system for
        bank transaction, Islaamaabaad Branch. Have a nice day.
        <exit/>
    </block>
rm>
```

ASP

```
n="1.0" encoding="iso-8859-1"?>
E vxml PUBLIC "vxml" "">
```

```
version="1.0">
next = "main.asp#home">
        <grammar>home</grammar>
</link>
<link next = "main.asp#end">
        <grammar>exit</grammar>
</link>
<var name = "currentoption" expr = "'home'"/>
id = "aboutus">
<block>
<%
Set conn = Server.CreateObject("ADODB.Connection")
conn.Provider = "Microsoft.Jet.OLEDB.4.0"
conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
conn.open
dim verifycustomerSQL1
'Execute Query to get detail
verifycustomerSQL1 = "select * from tbl_bank"
Set rs1 = conn.Execute(verifycustomerSQL1)
Response.Write rs1("bank_aboutus")
%>
<assign name = "currentoption" expr ="'aboutus'"/>
<goto next = "#repeat"/>
</block>

id ="repeat">
<field name = "confirm" type = "boolean">
        <prompt>To repeat say yes. Say no, to go back to home.</prompt>
                <filled>
                        <if cond = "confirm == true">
                        <goto expr = "'#' + currentoption"/>
                        <else/>
                        <goto next = "main.asp"/>
                        </if>
                </filled>
</field>


WS.ASP

version="1.0" encoding="iso-8859-1"?>
OCTYPE vxml PUBLIC "vxml" "">
version="1.0">
next = "main.asp#home">
        <grammar>home</grammar>
</link>
<link next = "main.asp#end">
        <grammar>exit</grammar>
</link>
<var name = "currentoption" expr = "'home'"/>
id = "news">
<block>
<%
Set conn = Server.CreateObject("ADODB.Connection")
conn.Provider = "Microsoft.Jet.OLEDB.4.0"
```

```
conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
conn.open
dim verifycustomerSQL1
verifycustomerSQL1 = "select * from tbl_news"
Set rs1 = conn.Execute(verifycustomerSQL1)
do until rs1.eof
response.write rs1("news_detail")
%>
<break msecs = "1200" />
<%
rs1.movenext
loop
%>
<assign name = "currentoption" expr ="'news'"/>
<goto next = "#repeat"/>
</block>

d ="repeat">
 <field name = "confirm" type = "boolean">
          <prompt>To repeat say yes. Say no, to go back to home.</prompt>
                    <filled>
                              <if cond = "confirm == true">
                              <goto expr = "'#' + currentoption"/>
                              <else/>
                              <goto next = "main.asp"/>
                              </if>
                    </filled>
  </field>
>
>
```

LLPAYMENT.ASP

```
version="1.0" encoding="iso-8859-1"?>
TYPE vxml PUBLIC "vxml" "">
version="1.0">
ext = "main.asp#home">
  <grammar>home</grammar>

ext = "main.asp#end">
  <grammar>exit</grammar>

ame = "currentoption" expr = "'home'"/>
id = "main">
>

  Dim valuecase
  valuecase = 0
  Randomize
  valuecase = CInt(Int((8 * Rnd()) + 1))
  if valuecase = 1 then
            valuecase = 0
            %>
            <goto next = "#main1"/>
            <%
  elseif valuecase = 2 then
```

```
        valuecase = 0
        %>
        <goto next = "#main2"/>
        <%
elseif valuecase = 3 then
        valuecase = 0
        %>
        <goto next = "#main3"/>
        <%
elseif valuecase = 4 then
        valuecase = 0
        %>
        <goto next = "#main4"/>
        <%
elseif valuecase = 5 then
        valuecase = 0
        %>
        <goto next = "#main5"/>
        <%
elseif valuecase = 6 then
        valuecase = 0
        %>
        <goto next = "#main6"/>
        <%
elseif valuecase = 7 then
        valuecase = 0
        %>
        <goto next = "#main7"/>
        <%
elseif valuecase = 8 then
        valuecase = 0
        %>
        <goto next = "#main8"/>
        <%
end if

k>
>
id = "main1">
>
        Welcome to the account authentication service.
</block>
<field name="authentication_check" type="digits">
        <prompt>What is your day of birth?</prompt>
        <catch event="help">
                Please tell your day of birth.
        </catch>
</field>
        <block>
        <submit     next="/VProject/acbillpaymentdbv.asp#acbillpaymentdbv"     method="post"
st="authentication_check"/>
</block>
>
id = "main2">
>
        Welcome to the account authentication service.
```

```
</block>
<field name="authentication_check" type="digits">
                                        <prompt>What is your month of birth?</prompt>
                                <catch event="help">
                                        Please tell your month of birth.
                                </catch>
</field>
<block>
        <submit     next="/VProject/acbillpaymentdbv.asp#acbillpaymentdbv"     method="post"
ist="authentication_check"/>
</block>
>
id = "main3">
<block>
        Welcome to the account authentication service.
</block>
<field name="authentication_check" type="digits">
        <prompt>What is your year of birth?</prompt>
        <catch event="help">
                Please tell your year of birth.
        </catch>
</field>
        <block>
        <submit     next="/VProject/acbillpaymentdbv.asp#acbillpaymentdbv"     method="post"
list="authentication_check"/>
</block>
>
id = "main4">
<block>
        Welcome to the account authentication service.
</block>
<field name="authentication_check" type="digits">
        <prompt>What is your social security number?</prompt>
        <catch event="help">
                Please tell your 15 digit social security number.
        </catch>
</field>
        <block>
        <submit     next="/VProject/acbillpaymentdbv.asp#acbillpaymentdbv"     method="post"
elist="authentication_check"/>
</block>
>
id = "main5">
<block>
        Welcome to the account authentication service.
</block>
<field name="authentication_check" type="digits">
        <prompt>What is city zip code?</prompt>
        <catch event="help">
                Please tell city zip code which you gave at time of account registration.
        </catch>
</field>
        <block>
        <submit     next="/VProject/acbillpaymentdbv.asp#acbillpaymentdbv"     method="post"
elist="authentication_check"/>
</block>
```

```
</form>
<form id = "main6">
    <block>
        Welcome to the account authentication service.
    </block>
    <field name="authentication_check" type="digits">
        <prompt>What is your secret number?</prompt>
        <catch event="help">
            Please tell your secret number which you gave at time of account registration.
        </catch>
    </field>
        <block>
        <submit  next="/VProject/acbillpaymentdbv.asp#acbillpaymentdbv"  method="post"
namelist="authentication_check"/>
    </block>
</form>
<form id = "main7">
    <block>
        Welcome to the account authentication service.
    </block>
    <field name="authentication_check" type="digits">
        <prompt>What is your life insurance number?</prompt>
        <catch event="help">
            Please tell your life insurance number.
        </catch>
    </field>
        <block>
        <submit  next="/VProject/acbillpaymentdbv.asp#acbillpaymentdbv"  method="post"
namelist="authentication_check"/>
    </block>
</form>
<form id = "main8">
    <block>
        Welcome to the account authentication service.
    </block>
    <field name="authentication_check" type="digits">
        <prompt>What is your car license number?</prompt>
        <catch event="help">
            Please tell your car license number
        </catch>
    </field>
        <block>
        <submit  next="/VProject/acbillpaymentdbv.asp#acbillpaymentdbv"  method="post"
namelist="authentication_check"/>
    </block>
</form>
</vxml>
```

CBILLPAYMENTDBV.ASP

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE vxml PUBLIC "vxml" "">
<vxml version="1.0">
<link next = "main.asp#home">
    <grammar>home</grammar>
</link>
```

```
= "main.asp#end">
grammar>exit</grammar>

= "currentoption" expr = "home"/>
"acbillpaymentdbv">
block>
%
et conn = Server.CreateObject("ADODB.Connection")
onn.Provider = "Microsoft.Jet.OLEDB.4.0"
onn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
onn.open
im ac
=Request.Form("authentication_check")
im verifycustomerSQL1
Execute Query to get detail
acsql
erifyacsql = "select * from tbl_customer where customer_dobday='"&ac&"' or
dobmonth='"&ac&"' or customer_dobyear='"&ac&"' or customer_ssn='"&ac&"' or
zipcode='"&ac&"' or customer_secretno='"&ac&"' or customer_lifenumber='"&ac&"' or
licenseno='"&ac&"'"
et rs5 = conn.Execute(verifyacsql)
f rs5.bof and rs5.eof then
>
goto next="transerror.asp"/>
%
nd if
>
goto next = "billpayment.asp#billpayment"/>
```

TS.ASP

```
sion="1.0" encoding="iso-8859-1"?>
PE vxml PUBLIC "vxml" "">
sion="1.0">
= "main.asp#home">
        <grammar>home</grammar>
/link>
link next = "main.asp#end">
        <grammar>exit</grammar>
/link>
var name = "currentoption" expr = "home"/>
 "products">
block>
        With over 40 years of experience in Pakistan,
        Citibank has been a pioneer in introducing new products
        and services that have been emulated across the
        financial services industry.
/block>
subdialog src = "#services"/>

menu id = "services">
        <prompt count = "1">
                Please say one of the following products or services.
```

```
                <break msecs = "2000"/>
                <enumerate/>
        </prompt>
        <choice next = "services.asp#car">Car Financing
        <grammar type="application/x-jsgf">
                What is [city] car financing |
                What is inside [city] car financing |
                How can I get [car] finance from [city] [bank]
        </grammar>
        </choice>
        <choice next = "services.asp#loans">City Loans
        <grammar type="application/x-jsgf">
                What (is|are) [city] (loan|loans) |
                How can I get city loan |
                What (is|are) the (benefit|benefits) of [city] loan
        </grammar>
        </choice>
        <choice next = "services.asp#human">Human Rescources
        <grammar type="application/x-jsgf">
                Can I get job at [city] bank
                What is [inside] human resources
        </grammar>
        </choice>
  </menu>
 id ="repeat">
  <field name = "confirm" type = "boolean">
  <prompt>To repeat say yes. Say no, to go back to home.</prompt>
                <filled>
                        <if cond = "confirm == true">
                        <goto expr = "'#' + currentoption"/>
                        <else/>
                        <goto next = "main.asp"/>
                        </if>
                </filled>
  </field>
rm>
ml>
```

VICES.ASP

```
ml version="1.0" encoding="iso-8859-1"?>
OCTYPE vxml PUBLIC "vxml" "">
ml version="1.0">
k next = "main.asp#home">
    <grammar>home</grammar>
nk>
k next = "main.asp#end">
    <grammar>exit</grammar>
nk>
r name = "currentoption" expr = "'home'"/>
rm id = "car">
    <block>
    With the City Bank Car Financing u can get following benefits.
    <break msecs = "500"/>
    Number 1. Pre-approved Credit Card with every car.
    Number 2. Citibank Financing Counters at our dealership offer
```

you the convenience of sales officers trained in providing you with
the best financing option.
Number 3. Largest network of authorized dealerships ensures that
you get the car of your choice in the color of your choice.
Number 4. Round the clock Citiphone Banking Service, to help and
assist you regarding any customer queries.
Number 5. Comprehensive insurance at reduced premium rates from
AIG New Hampshire,and EFU.
And Number 6. Special pricing breaks for selected Credit Card members.
<assign name = "currentoption" expr = "'car'"/>
<goto next = "#repeat"/>
</block>

= "loans">
<block>
With over 40 years of experience in Pakistan,
Citibank has been a pioneer in introducing new products
and services that have been emulated across the
financial services industry.
<assign name = "currentoption" expr = "'loans'"/>
<goto next = "#repeat"/>
</block>

l = "human">
<block>
We are committed to attracting, challenging and rewarding the top
professionals in financial services and to creating an environment in
which employees realize their maximum potential. We reward people based
on performance standards which are aligned with our business objectives
and long term goals. The various Training and development tools at
Citibank assist our employees to be fully effective in their jobs and
long term careers. Opportunities at Citibank offer a wide scope of learning
and development while providing with a perspective which is attuned to the
global markets while catering to the local environment. Whether you are
still at your studies, have graduated or ahead in life on a career track
already there could be challenging opportunities at Citibank to suit your
ambitions. Submit your view for current positions being offered at Citibank or
become part of exciting employment opportunities in the future for internships,
entry level, junior level and executive positions.
<assign name = "currentoption" expr = "'human'"/>
<goto next = "#repeat"/>
</block>

d ="repeat">
<field name = "confirm" type = "boolean">
<prompt>To repeat say yes. Say no, to go back to home.</prompt>
<filled>
<if cond = "confirm == true">
<goto expr = "'#' + currentoption"/>
<else/>
<goto next = "main.asp"/>
</if>
</filled>
</field>

BALANCE.ASP

```
l version="1.0" encoding="iso-8859-1"?>
CTYPE vxml PUBLIC "vxml" "">
l version="1.0">
 next = "main.asp#home">
    <grammar>home</grammar>
>
 next = "main.asp#end">
    <grammar>exit</grammar>
>
name = "currentoption" expr = "'home'"/>
 id = "balance_info">
    <block>
            Welcome to the account balance inquiry service.
    </block>
    <field name="customer_accno" type="digits">
            <prompt>What is your account number?</prompt>
            <catch event="help">
                    Please type in or spell out your account number.
            </catch>
    </field>
    <field name="customer_pinno" type="digits">
            <prompt>Your PIN?</prompt>
            <catch event="help">
                    Please enter or spell your four digit personal
                    identification number, PIN, for your account number
            </catch>
    </field>
    <block>
            <submit next="/VProject/verify.asp#verify" method="post" namelist="customer_accno
ner_pinno"/>
    </block>
>
I>
```

FY.ASP

```
 version="1.0" encoding="iso-8859-1"?>
CTYPE vxml PUBLIC "vxml" "">
 version="1.0">
ext = "main.asp#home">
   <grammar>home</grammar>
>
ext = "main.asp#end">
   <grammar>exit</grammar>
>
me = "currentoption" expr = "'home'"/>
id="verify">
   <block>
   <%
   Set conn = Server.CreateObject("ADODB.Connection")
   conn.Provider = "Microsoft.Jet.OLEDB.4.0"
   conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
   conn.open
   dim accno
```

```
dim pinno
accno=Request.Form("customer_accno")
pinno=Request.Form("customer_pinno")
dim verifycustomerSQL1
'Execute Query to get detail
verifycustomerSQL1 = "select * from tbl_customer, tbl_account where
tbl_customer.customer_accno='"&accno&"' and tbl_customer.customer_pinno='"&pinno&"' and
tbl_account.customer_accno='"&accno&"'"
Set rs1 = conn.Execute(verifycustomerSQL1)
If rs1.bof and RS1.EOF Then %>
<goto next="transerror.asp"/>
<%
else
Response.Write rs1("customer_title")
Response.Write rs1("customer_fname")
Response.Write rs1("customer_lname")
Response.Write ("You are the Authorized Customer")
Response.Write ("And your current balance is")
%>
<break msecs = "200"/>
<%
Response.Write rs1("account_balance")
Response.Write rs1("account_nature")
end if
rs1.close
set rs1 = Nothing
Conn.Close
Set Conn = Nothing
%>
<assign name = "currentoption" expr = "'verify'"/>
<goto next = "#repeat"/>
block>
form>
form id ="repeat">
<field name = "confirm" type = "boolean">
<prompt>To repeat say yes. Say no, to go back to home.</prompt>
<filled>
<if cond = "confirm == true">
<goto expr = "'#' + currentoption"/>
<else/>
<goto next = "main.asp"/>
</if>
</filled>
</field>
form>
vxml>
```

RANSERROR.ASP

```
xml version="1.0" encoding="iso-8859-1"?>
DOCTYPE vxml PUBLIC "vxml" "">
xml version="1.0">
orm>
<block>
You are not the authorized customer, please try again
<goto next="accbalance.asp"/>
```

```
>
>
>
```

EYTRANS.ASP

```
version="1.0" encoding="iso-8859-1"?>
TYPE vxml PUBLIC "vxml" "">
version="1.0">
ext = "main.asp#home">
  <grammar>home</grammar>

ext = "main.asp#end">
  <grammar>exit</grammar>

d = "moneytrans">
>
          Welcome to the City Bank money transfer service.
 </block>
 <field name="customer_accno" type="digits">
          <prompt>What is your account number?</prompt>
          <catch event="help">
                Please type in or spell out your account number.
          </catch>
 </field>
 <field name="customer_pinno" type="digits">
          <prompt>Your PIN?</prompt>
          <catch event="help">
                Please enter or spell your four digit personal
                identification number, PIN, for your account number
          </catch>
 </field>
ame="money_to_transfer" type="digits">
          <prompt>How much money you want to transfer?</prompt>
          <catch event="help">
                Please enter or spell that how much money
                you want to transfer to other account
          </catch>
 </field>
 <field name="transfer_to_account" type="digits">
          <prompt>To which account number u want to transfer?</prompt>
          <catch event="help">
                Please enter or spell to which account number
                you want to transfer money
          </catch>
 </field>
 <block>
          <submit next="mtransdbv.asp#mtransdbv" method="post" namelist="customer_accno
r_pinno money_to_transfer transfer_to_account"/>
 </block>
```

NSDBV.ASP

```
ersion="1.0" encoding="iso-8859-1"?>
```

```
CTYPE vxml PUBLIC "vxml" "">
 version="1.0">
next = "main.asp#home">
   <grammar>home</grammar>
>
next = "main.asp#end">
   <grammar>exit</grammar>
>
ame = "currentoption" expr = "'home'"/>
id="mtransdbv">
   <block>
   <%
   dim accno
   dim pinno
   dim mttrans
   dim ttacc
   dim queryselect1
   dim queryselect2
   Set conn = Server.CreateObject("ADODB.Connection")
   conn.Provider = "Microsoft.Jet.OLEDB.4.0"
   conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
   conn.open
   accno=Request.Form("customer_accno")
   pinno=Request.Form("customer_pinno")
   mttrans=Request.Form("money_to_transfer")
   ttacc=Request.Form("transfer_to_account")
   queryselect1 = "select * from tbl_customer, tbl_account where
omer.customer_accno='"&accno&"' and tbl_customer.customer_pinno='"&pinno&"' and
unt.customer_accno ='"&accno&"'"
   queryselect2 = "select * from tbl_customer,tbl_account where
omer.customer_accno='"&ttacc&"' and tbl_account.customer_accno = '"&ttacc&"'"
   Set rs1 = conn.Execute(queryselect1)
   Set rs2 = conn.Execute(queryselect2)
   If rs1.eof Then %>
   <goto next="transferror.asp"/>
   <%
   else
   if rs2.eof then
            %>
            <goto next="transferror1.asp"/>
            <%
else
If cint(mttrans) > cint(rs1("account_balance")) Then
                  %>
                        <goto next="transferror2.asp"/>
                  <%
else
                        dim temp1
                        temp1 = cint(rs1("account_balance")) - cint(mttrans)
                        Response.Write rs1("customer_title")
                        Response.Write rs1("customer_fname")
                        Response.Write rs1("customer_lname")
                        Response.Write ("Now current balance in your account is ")
                        Response.Write (temp1)
                        Response.Write rs2("account_nature")
                        Dim sqlupdate1
```

```
                        sqlupdate1 = "update tbl_account set account_balance = '"&temp1&"'
omer_accno='"&accno&"'"
                        Set rs1 = conn.Execute (sqlupdate1)
                        dim temp2
                        temp2 = cint(rs2("account_balance")) + cint(mttrans)
                        dim sqlupdate2
                        sqlupdate2 = "update tbl_account set account_balance = '"&temp2&"'
omer_accno='"&ttacc&"'"
                        Set rs2 = conn.Execute (sqlupdate2)
                        Response.Write ("And the money have successfully transferred")
        End if
        End if
nd if
        %>
assign name = "currentoption" expr = "'mtransdbv'"/>
goto next = "#repeat"/>


="repeat">
field name = "confirm" type = "boolean">
        <prompt>To repeat say yes. Say no, to go back to home.</prompt>
                <filled>
                        <if cond = "confirm == true">
                        <goto expr = "'#' + currentoption"/>
                        <else/>
                        <goto next = "main.asp"/>
                        </if>
                </filled>
</field>
```

FERROR.ASP

```
rsion="1.0" encoding="iso-8859-1"?>
YPE vxml PUBLIC "vxml" "">
rsion="1.0">

<block>
You are not the authorized customer, please try again
<goto next="moneytrans.asp"/>
</block>
```

FERROR1.ASP

```
ersion="1.0" encoding="iso-8859-1"?>
YPE vxml PUBLIC "vxml" "">
ersion="1.0">

 <block>
 You are going to transfer a money to an account which do not exist.
 Please try again.
  <goto next="moneytrans.asp"/>
```

```
</block>
```

FERROR2.ASP

```
ersion="1.0" encoding="iso-8859-1"?>
TYPE vxml PUBLIC "vxml" "">
ersion="1.0">

  <block>
  You dont have enough money to transfer, check your account balance
  <goto next="main.asp#home"/>
  </block>
>
>
```

AYMENT.ASP

```
version="1.0" encoding="iso-8859-1"?>
TYPE vxml PUBLIC "vxml" "">
version="1.0">
ext = "main.asp#home">
mar>home</grammar>

ext = "main.asp#end">
   <grammar>exit</grammar>

ame = "currentoption" expr = "'home'"/>
id = "billpayment">
   <block>
          Welcome to the City Bank quick online bill payment service.
   </block>
   <field name="bill_id" type="digits">
          <prompt>What is the bill i d?</prompt>
          <catch event="help">
                Please type in or spell out your bill id.
          </catch>
   </field>
   <field name="customer_accno" type="digits">
          <prompt>What is your account number?</prompt>
          <catch event="help">
                Please type in or spell out your account number.
                Dont say spaces or dash in between
          </catch>
   </field>
   <field name="customer_pinno" type="digits">
          <prompt>Your PIN?</prompt>
          <catch event="help">
                Please enter or spell your four digit personal
                identification number, PIN, for your account number
          </catch>
   </field>

          <block>
```

```
            <submit next="billpaydbv.asp#billpaydbv" method="post" namelist=" bill_id
omer_accno customer_pinno"/>
      </block>
rm>
ml>
```

LLPAYDBV.ASP

```
ml version="1.0" encoding="iso-8859-1"?>
OCTYPE vxml PUBLIC "vxml" "">
ml version="1.0">
k next = "main.asp#home">
      <grammar>home</grammar>
nk>
k next = "main.asp#end">
      <grammar>exit</grammar>
nk>
r name = "currentoption" expr = "'home'"/>
m id="billpaydbv">
      <block>
      <%
      dim billid
      dim accno
      dim pinno
      dim queryselect1
      dim queryselect2
      dim queryselect3
      Set conn = Server.CreateObject("ADODB.Connection")
      conn.Provider = "Microsoft.Jet.OLEDB.4.0"
      conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
      conn.open
      billid=Request.Form ("bill_id")
      accno=Request.Form("customer_accno")
      pinno=Request.Form("customer_pinno")
      queryselect1 = "select * from tbl_bill where bill_id ='"&billid&"'"
      queryselect2 = "select * from tbl_customer, tbl_account where
customer.customer_accno='"&accno&"' and tbl_customer.customer_pinno='"&pinno&"' and
account.customer_accno ='"&accno&"'"
      Set rs1 = conn.Execute(queryselect1)
      Set rs2 = conn.Execute(queryselect2)
      If rs1.eof Then %>
      <goto next="billerror1.asp"/>
      <%
      else
      if rs2.eof then
      %>
      <goto next="billerror2.asp"/>
            <%
      else
            if (rs1("bill_is_paid")) = true then
            %>
            <goto next="billerror4.asp"/>
            <%
      else
                  If cint(rs1("bill_amount")) > cint(rs2("account_balance")) Then
                  %>
```

```
                    <goto next="billerror3.asp"/>
            <%
                    else
                    dim temp1
                    temp1 = cint(rs2("account_balance")) - cint(rs1("bill_amount"))
                    Response.Write rs2("customer_title")
                    Response.Write rs2("customer_fname")
                    Response.Write rs2("customer_lname")
                    Response.Write ("Now current balance in your account is ")
                    Response.Write (temp1)
                    Response.Write rs2("account_nature")
                    Dim sqlupdate1
                    sqlupdate1 = "update tbl_account set account_balance = '"&temp1&"'
stomer_accno='"&accno&"'"
                    Set rs1 = conn.Execute (sqlupdate1)
                    dim temp3
                    temp3 = 1
                    temp3 = cint(temp3)
                    dim sqlupdate2
                    sqlupdate2 = "update tbl_bill set bill_is_paid = '"&temp3&"' where
'"&billid&"'"
                    Set rs2 = conn.Execute (sqlupdate2)
                    Response.Write ("And the bill is paid. Thankyou for using City bank
nline bill payment service.")
                End if
        End if
        End if
 End if
        %>
 <assign name = "currentoption" expr = "'billpaydbv'"/>
 <goto next = "#repeat"/>
>
>
id ="repeat">
 <field name = "confirm" type = "boolean">
        <prompt>To repeat say yes. Say no, to go back to home.</prompt>
        <filled>
                    <if cond = "confirm == true">
                    <goto expr = "'#' + currentoption"/>
                    <else/>
                    <goto next = "main.asp"/>
                    </if>
                </filled>
  </field>
>
I>
```

ERROR1.ASP

```
 version="1.0" encoding="iso-8859-1"?>
CTYPE vxml PUBLIC "vxml" "">
l version="1.0">
>
  <block>
        It is not a valid bill i d. Please enter a valid bill i d.
  <goto next="billpayment.asp"/>
```

</block>

RROR2.ASP

```
ersion="1.0" encoding="iso-8859-1"?>
TYPE vxml PUBLIC "vxml" "">
ersion="1.0">

  <block>
  You are not a authorized customer. Please try again.
  <goto next="billpayment.asp"/>
  </block>
>
>
```

RROR3.ASP

```
version="1.0" encoding="iso-8859-1"?>
TYPE vxml PUBLIC "vxml" "">
version="1.0">

   <block>
          You dont have enough amount to pay a bill.
   <goto next="main.asp"/>
   </block>
>
>
```

ERROR4.ASP

```
 version="1.0" encoding="iso-8859-1"?>
CTYPE vxml PUBLIC "vxml" "">
 version="1.0">
>
    <block>
          Bill is already paid.
    <goto next="main.asp"/>
    </block>
>
>
```

ONEYTRANSFER.ASP

```
ml version="1.0" encoding="iso-8859-1"?>
CTYPE vxml PUBLIC "vxml" "">
l version="1.0">
next = "main.asp#home">
    <grammar>home</grammar>
>
next = "main.asp#end">
    <grammar>exit</grammar>
>
name = "currentoption" expr = "'home'"/>
 id = "main">
```

```
Dim valuecase
valuecase = 0
Randomize
valuecase = CInt(Int((8 * Rnd()) + 1))
if valuecase = 1 then
        valuecase = 0
        %>
        <goto next = "#main1"/>
        <%
elseif valuecase = 2 then
        valuecase = 0
        %>
        <goto next = "#main2"/>
        <%
elseif valuecase = 3 then
        valuecase = 0
        %>
        <goto next = "#main3"/>
        <%
elseif valuecase = 4 then
        valuecase = 0
        %>
        <goto next = "#main4"/>
        <%
elseif valuecase = 5 then
        valuecase = 0
        %>
        <goto next = "#main5"/>
        <%
elseif valuecase = 6 then
        valuecase = 0
        %>
        <goto next = "#main6"/>
        <%
elseif valuecase = 7 then
        valuecase = 0
        %>
        <goto next = "#main7"/>
        <%
elseif valuecase = 8 then
        valuecase = 0
        %>
        <goto next = "#main8"/>
<%
end if

>

d = "main1">
 <block>
        Welcome to the account authentication service.
 </block>
 <field name="authentication_check" type="digits">
        <prompt>What is your day of birth?</prompt>
```

```
        <catch event="help">
                Please tell your day of birth.
        </catch>
</field>
        <block>
        <submit next="/VProject/acmoneytransferdbv.asp#acmoneytransferdbv" method="post"
="authentication_check"/>
</block>

= "main2">
<block>
        Welcome to the account authentication service.
</block>
<field name="authentication_check" type="digits">
                                        <prompt>What is your month of birth?</prompt>
                                <catch event="help">
                                        Please tell your month of birth.
                                </catch>
</field>
<block>
        <submit next="/VProject/acmoneytransferdbv.asp#acmoneytransferdbv" method="post"
="authentication_check"/>
</block>

d = "main3">
<block>
        Welcome to the account authentication service.
</block>
<field name="authentication_check" type="digits">
        <prompt>What is your year of birth?</prompt>
        <catch event="help">
                Please tell your year of birth.
        </catch>
</field>
        <block>
        <submit next="/VProject/acmoneytransferdbv.asp#acmoneytransferdbv" method="post"
st="authentication_check"/>
</block>
>
id = "main4">
<block>
        Welcome to the account authentication service.
</block>
<field name="authentication_check" type="digits">
        <prompt>What is your social security number?</prompt>
        <catch event="help">
                Please tell your 15 digit social security number.
        </catch>
</field>
        <block>
        <submit next="/VProject/acmoneytransferdbv.asp#acmoneytransferdbv" method="post"
ist="authentication_check"/>
</block>
n>
n id = "main5">
<block>
```

```
        Welcome to the account authentication service.
  </block>
  <field name="authentication_check" type="digits">
        <prompt>What is city zip code?</prompt>
        <catch event="help">
              Please tell city zip code which you gave at time of account registration.
        </catch>
  </field>
        <block>
        <submit next="/VProject/acmoneytransferdbv.asp#acmoneytransferdbv" method="post"
ist="authentication_check"/>
  </block>
>
id = "main6">
  <block>
        Welcome to the account authentication service.
  </block>
  <field name="authentication_check" type="digits">
        <prompt>What is your secret number?</prompt>
        <catch event="help">
              Please tell your secret number which you gave at time of account registration.
        </catch>
  </field>
        <block>
        <submit next="/VProject/acmoneytransferdbv.asp#acmoneytransferdbv" method="post"
list="authentication_check"/>
  </block>
m>
n id = "main7">
  <block>
        Welcome to the account authentication service.
  </block>
  <field name="authentication_check" type="digits">
        <prompt>What is your life insurance number?</prompt>
        <catch event="help">
              Please tell your life insurance number.
        </catch>
  </field>
        <block>
        <submit next="/VProject/acmoneytransferdbv.asp#acmoneytransferdbv" method="post"
elist="authentication_check"/>
  </block>
rm>
m id = "main8">
ck>
        Welcome to the account authentication service.
  </block>
  <field name="authentication_check" type="digits">
        <prompt>What is your car license number?</prompt>
        <catch event="help">
              Please tell your car license number
        </catch>
  </field>
        <block>
        <submit next="/VProject/acmoneytransferdbv.asp#acmoneytransferdbv" method="post"
elist="authentication_check"/>
```

SFERDBV.ASP

```
' encoding="iso-8859-1"?>
PUBLIC "vxml" "">
">
asp#home">
-home</grammar>

asp#end">
>exit</grammar>

ntoption" expr = "'home'"/>
'transferdbv">


= Server.CreateObject("ADODB.Connection")
der = "Microsoft.Jet.OLEDB.4.0"
nectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")


st.Form("authentication_check")
'customerSQL1
Query to get detail

ql = "select * from tbl_customer where customer_dobday='"&ac&"' or
th='"&ac&"' or customer_dobyear='"&ac&"' or customer_ssn='"&ac&"' or
='"&ac&"' or customer_secretno='"&ac&"' or customer_lifenumber='"&ac&"' or
o='"&ac&"'"
conn.Execute(verifyacsql)
and rs5.eof then

xt="transerror.asp"/>


xt = "moneytrans.asp#moneytrans"/>




.0" encoding="iso-8859-1"?>
ml PUBLIC "vxml" "">
1.0">
n.asp#home">
ar>home</grammar>

in.asp#end">
ar>exit</grammar>

>
```

```
<block>
        Welcome to the City bank Fax service.
</block>
<field name="customer_accno" type="digits">
        <prompt>What is your account number?</prompt>
        <catch event="help">
                Please type in or spell out your account number.
        </catch>
</field>
<field name="customer_pinno" type="digits">
        <prompt>Your PIN?</prompt>
        <catch event="help">
                Please enter or spell your four digit personal
                identification number, PIN, for your account number
        </catch>
 </field>
 <field name="fax_number" type="digits">
        <prompt>What is the fax number?</prompt>
        <catch event="help">
                Please enter or spell fax number, where you want to get account transaction
                detail
        </catch>
  </field>
  <block>
        <submit next="faxdbv.asp#faxdbv" method="post" namelist="customer_accno
er_pinno fax_number"/>
  </block>
>
l>
```

BV.ASP

```
 version="1.0" encoding="iso-8859-1"?>
CTYPE vxml PUBLIC "vxml" "">
l version="1.0">
next = "main.asp#home">
   <grammar>home</grammar>
>
next = "main.asp#end">
   <grammar>exit</grammar>
>
name = "currentoption" expr = "'home'"/>
n id="faxdbv">
   <block>
   <%
   dim accno
   dim pinno
   dim faxnumber
   dim queryselect1
   Set conn = Server.CreateObject("ADODB.Connection")
   conn.Provider = "Microsoft.Jet.OLEDB.4.0"
   conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
   conn.open
   accno=Request.Form("customer_accno")
   pinno=Request.Form("customer_pinno")
   faxnumber=Request.Form ("fax_number")
```

```
    queryselect1 = "select * from tbl_customer where customer_accno='"&accno&"' and
mer_pinno='"&pinno&"'"
    Set rs1 = conn.Execute(queryselect1)
    If rs1.eof Then %>
    <goto next="faxerror.asp"/>
    <%
    else
            dim queryinsert1
            queryinsert1 = "INSERT INTO tbl_fax (fax_customer_accno,fax_to_number, fax_date)
UES ('"&accno&"','"&faxnumber&"',#"&Date()&"#)"
            set rs2 = conn.Execute (queryinsert1)
            Response.Write rs1("customer_title")
            Response.Write rs1("customer_fname")
            %>
            <break msec = "150"/>
            <%
            Response.Write rs1("customer_lname")
            %>
            <break msec = "500"/>
            <%
            Response.Write ("In short time u will receive fax. Thankyou for using City Fax service.")
    End if
    %>
    <assign name = "currentoption" expr = "'faxdbv'"/>
    <goto next = "#repeat"/>
lock>
rm>
m id ="repeat">
    <field name = "confirm" type = "boolean">
    <prompt>To repeat say yes. Say no, to go back to home.</prompt>
                <filled>
                    <if cond = "confirm == true">
                    <goto expr = "'#' + currentoption"/>
                    <else/>
                    <goto next = "main.asp"/>
                    </if>
                </filled>
    </field>
rm>
xml>
```

XERROR.ASP

```
ml version="1.0" encoding="iso-8859-1"?>
OCTYPE vxml PUBLIC "vxml" "">
ml version="1.0">
m>
    <block>
    You are not a authorized customer. Please try again.
    <goto next="fax.asp"/>
    </block>
m>
ml>
```

)LS.ASP

```
nl version="1.0" encoding="iso-8859-1"?>
)CTYPE vxml PUBLIC "vxml" "">
nl version="1.0">
 next = "main.asp#home">
    <grammar>home</grammar>
k>
 next = "main.asp#end">
    <grammar>exit</grammar>
<>
n id = "tools">
k>
            Welcome to the City bank tool section, here u can change ur pin number
            and activate or deactivate your credit card.
    </block>
    <subdialog src = "#toolsub"/>
n>
u id = "toolsub" dtmf="true">
            <prompt count = "1" timeout = "10s">
                    Please make      a selection by speaking one of the following options:
                    <break msecs = "1000" />
                    <enumerate/>
            </prompt>
  <prompt count = "2">
                    Please say one of the following.
                    <break msecs = "2000"/>
                    <enumerate/>
            </prompt>
            <choice next = "pinno.asp#pinno">Pin Number</choice>
            <choice next = "creditcardad.asp#creditcardad">Credit Card</choice>
u>
>
```

).ASP

```
ersion="1.0" encoding="iso-8859-1"?>
TYPE vxml PUBLIC "vxml" "">
version="1.0">
ext = "main.asp#home">
  <grammar>home</grammar>

ext = "main.asp#end">
  <grammar>exit</grammar>

d = "pinno">
 <block>
        Welcome to the City bank, change pin number service.
 </block>
 <field name="customer_accno" type="digits">
        <prompt>What is your account number?</prompt>
        <catch event="help">
                Please type in or spell out your account number.
        </catch>
</field>
<field name="customer_pinno" type="digits">
```

```
        <prompt>Your pin?</prompt>
        <catch event="help">
                Please enter or spell your four digit personal
                identification number, PIN, for your account number.
        </catch>
</field>
<field name="new_pinno" type="digits">
        <prompt>What is your new pin number?</prompt>
        <catch event="help">
                Please enter or spell your four digit personal
                identification number, PIN, for your account number
        </catch>
</field>
<field name = "confirm" type = "boolean">
        <prompt>Are u sure, if then say yes. Say no, to go back to tools section.</prompt>
                <filled>
                        <if cond = "confirm == true">
                                <submit next="pinnodbv.asp#pinnodbv" method="post"
t="customer_accno customer_pinno new_pinno"/>
                        <else/>
                                <goto next = "tools.asp"/>
                        </if>
                </filled>
</field>
>
>
```

ODBV.ASP

```
version="1.0" encoding="iso-8859-1"?>
TYPE vxml PUBLIC "vxml" "">
version="1.0">
ext = "main.asp#home">
  <grammar>home</grammar>
>
ext = "main.asp#end">
  <grammar>exit</grammar>
>
ame = "currentoption" expr = "'home'"/>
id="pinnodbv">
  <block>
  <%
  dim accno
  dim pinno
  dim pinnonew
  dim queryselect1
  Set conn = Server.CreateObject("ADODB.Connection")
  conn.Provider = "Microsoft.Jet.OLEDB.4.0"
  conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
  conn.open
  accno=Request.Form("customer_accno")
  pinno=Request.Form("customer_pinno")
  pinnonew=Request.Form("new_pinno")
   queryselect1 = "select * from tbl_customer where customer_accno='"&accno&"' and
mer_pinno='"&pinno&"'"
```

```
    Set rs1 = conn.Execute(queryselect1)
    If rs1.eof Then
    %>
    <goto next="pinnoerror.asp"/>
    <%
    else
    Response.Write rs1("customer_title")
    %>
    <break msec = "150"/>
    <%
    Response.Write rs1("customer_fname")
    %>
    <break msec = "150"/>
    <%
    Response.Write rs1("customer_lname")
    Dim sqlupdate1
    sqlupdate1 = "update tbl_customer set customer_pinno = '"&pinnonew&"' where
omer_accno='"&accno&"'"
    Set rs1 = conn.Execute (sqlupdate1)
    Response.Write ("Your pin number have been successfully changed. Please remember your new
number.")
    End if
    %>
    <assign name = "currentoption" expr = "'pinnodbv'"/>
    <goto next = "#repeat"/>
ock>
rm>
m id ="repeat">
    <field name = "confirm" type = "boolean">
    <prompt>To repeat say yes. Say no, to go back to home.</prompt>
                    <filled>
                        <if cond = "confirm == true">
                        <goto expr = "'#' + currentoption"/>
                        <else/>
                        <goto next = "main.asp"/>
                        </if>
                    </filled>
    </field>
rm>
ml>
```

NOERROR.ASP

```
ml version="1.0" encoding="iso-8859-1"?>
OCTYPE vxml PUBLIC "vxml" "">
ml version="1.0">
m>
    <block>
    You are not the authorized customer, please try again
    <goto next="pinno.asp"/>
    </block>
rm>
ml>
```

REDITCARD.ASP

```
'xml version="1.0" encoding="iso-8859-1"?>
DOCTYPE vxml PUBLIC "vxml" "">
'xml version="1.0">
ink next = "main.asp#home">
            <grammar>home</grammar>
    </link>
    <link next = "main.asp#end">
            <grammar>exit</grammar>
    </link>
orm id = "creditcardad">
    <block>
            You can activate or deactivate credit card here.
    </block>
    <subdialog src = "#options"/>
form>
    <menu id = "options">
        <prompt count = "1">
                Please say one of the following options.
                <break msecs = "2000"/>
                <enumerate/>
        </prompt>
        <choice next = "creditcardactive.asp#activate">Activate</choice>
        <choice next = "creditcarddeactive.asp#deactivate">Deactivate</choice>
    </menu>
'xml>
```

REDITCARDACTIVE.ASP

```
xml version="1.0" encoding="iso-8859-1"?>
OCTYPE vxml PUBLIC "vxml" "">
xml version="1.0">
nk next = "main.asp#home">
    <grammar>home</grammar>
nk>
nk next = "main.asp#end">
    <grammar>exit</grammar>
nk>
rm id = "activate">
    <block>
        Welcome to the City bank, Activate credit card section.
    </block>
    <field name="customer_accno" type="digits">
        <prompt>What is your account number?</prompt>
        <catch event="help">
            Please type in or spell out your account number.
        </catch>
    </field>
    <field name="customer_pinno" type="digits">
        <prompt>Your Pin?</prompt>
        <catch event="help">
            Please enter or spell your four digit personal
            identification number, PIN, for your account number.
        </catch>
    </field>
```

```
    <field name="creditcard_number" type="digits">
            <prompt>Your Credit card number?</prompt>
            <catch event="help">
                    Please enter or spell your 15 digit credit card number.
            </catch>
    </field>
    <block>
            <submit next="creditcardactdbv.asp#creditcardactdbv" method="post"
ist="customer_accno customer_pinno creditcard_number"/>
    </block>
n>
l>
```

DITCARDACTDBV.ASP

```
l version="1.0" encoding="iso-8859-1"?>
CTYPE vxml PUBLIC "vxml" "">
l version="1.0">
next = "main.asp#home">
    <grammar>home</grammar>
>
next = "main.asp#end">
    <grammar>exit</grammar>
>
 id="creditcardactdbv">
    <block>
    <%
    dim accno
    dim pinno
    dim ccnumber
    dim queryselect1
    dim queryselect2
    Set conn = Server.CreateObject("ADODB.Connection")
    conn.Provider = "Microsoft.Jet.OLEDB.4.0"
    conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
    conn.open
    accno=Request.Form("customer_accno")
    pinno=Request.Form("customer_pinno")
    ccnumber=Request.Form("creditcard_number")
    queryselect1 = "select * from tbl_customer where customer_accno='"&accno&"' and
ner_pinno='"&pinno&"'"
    queryselect2 = "select * from tbl_credit where customer_accno='"&accno&"' and
_card_no='"&ccnumber&"'"
    Set rs1 = conn.Execute(queryselect1)
    Set rs2 = conn.Execute(queryselect2)
    if rs1.eof then
    %>
            <goto next="creditcarderror.asp"/>
            <%
            else
            if rs2.eof then
            %>
                    <goto next="creditcarderror1.asp"/>
            <%
            else
```

```
                if (rs2("credit_status")) = false then
                        Response.Write rs1("customer_title")
                        %>
                        <break msec = "150"/>
                        <%
                        Response.Write rs1("customer_fname")
                        %>
                        <break msec = "150"/>
                        <%
                        Response.Write rs1("customer_lname")
                        dim temp
                        temp = 1
                        temp = cint(temp)
                        Dim sqlupdate1
e customer_accno='"&accno&"'"   sqlupdate1 = "update tbl_credit set credit_status = '"&temp&"'

                        Set rs1 = conn.Execute (sqlupdate1)
.")                     Response.Write ("Now your credit card is activated, you can

                %>
                        <goto next = "main.asp"/>
                        <%
                else
                if (rs2("credit_status")) = true then
                Response.Write ("Your credit card is already active")
                %>
                <goto next = "main.asp"/>
                <%
                End if
        End if
    End if
    End if

<
>
>
```

ITCARDERROR.ASP

```
version="1.0" encoding="iso-8859-1"?>
TYPE vxml PUBLIC "vxml" "">
version="1.0">

 <block>
 You are not the authorized customer, please try again
 <goto next="creditcard.asp"/>
 </block>
```

TCARDERROR1.ASP

```
ersion="1.0" encoding="iso-8859-1"?>
YPE vxml PUBLIC "vxml" "">
ersion="1.0">
```

```
        <block>
        You dont have any credit card.
        <goto next="main.asp"/>
        </block>
</form>
</vxml>
```

CREDITCARDDEACTIVE.ASP

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE vxml PUBLIC "vxml" "">
<vxml version="1.0">
<link next = "main.asp#home">
        <grammar>home</grammar>
</link>
<link next = "main.asp#end">
        <grammar>exit</grammar>
</link>
<form id = "deactivate">
<block>
                Welcome to the City bank, Deactivate credit card section.
        </block>
<field name="customer_accno" type="digits">
                <prompt>What is your account number?</prompt>
                <catch event="help">
                        Please type in or spell out your account number.
                </catch>
        </field>
        <field name="customer_pinno" type="digits">
                <prompt>Your Pin?</prompt>
                <catch event="help">
                        Please enter or spell your four digit personal
                        identification number, PIN, for your account number.
                </catch>
        </field>
        <field name="creditcard_number" type="digits">
                <prompt>Your Credit card number?</prompt>
                <catch event="help">
                        Please enter or spell your 15 digit credit card number.
                </catch>
        </field>

        <block>
        <submit next="creditcarddactdbv.asp#creditcarddactdbv" method="post"
namelist="customer_accno customer_pinno creditcard_number"/>
        </block>
</form>
</vxml>
```

CREDITCARDDACTDBV.ASP

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE vxml PUBLIC "vxml" "">
<vxml version="1.0">
<link next = "main.asp#home">
        <grammar>home</grammar>
```

```
>
 next = "main.asp#end">
    <grammar>exit</grammar>
>
n id="creditcarddactdbv">
    <block>
    <%
    dim accno
    dim pinno
    dim ccnumber
    dim queryselect1
    dim queryselect2
    Set conn = Server.CreateObject("ADODB.Connection")
    conn.Provider = "Microsoft.Jet.OLEDB.4.0"
    conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
    conn.open
    accno=Request.Form("customer_accno")
    pinno=Request.Form("customer_pinno")
    ccnumber=Request.Form("creditcard_number")
    queryselect1 = "select * from tbl_customer where customer_accno='"&accno&"' and
mer_pinno='"&pinno&"'"
    queryselect2 = "select * from tbl_credit where customer_accno='"&accno&"' and
_card_no='"&ccnumber&"'"
s1 = conn.Execute(queryselect1)
    Set rs2 = conn.Execute(queryselect2)
    if rs1.eof then
    %>
            <goto next="creditcarderror.asp"/>
            <%
            else
            if rs2.eof then
            %>
                    <goto next="creditcarderror1.asp"/>
            <%
            else
                    if (rs2("credit_status")) = true then
                                    Response.Write rs1("customer_title")
                                    %>
                                    <break msec = "150"/>
                                    <%
                                    Response.Write rs1("customer_fname")
                                    %>
                                    <break msec = "150"/>
                                    <%
                                    Response.Write rs1("customer_lname")

                                    dim temp
                                    temp = 0
                                    temp = cint(temp)
                                    Dim sqlupdate1
                                    sqlupdate1 = "update tbl_credit set credit_status = '"&temp&"'
e customer_accno='"&accno&"'"
                                    Set rs1 = conn.Execute (sqlupdate1)
                                    Response.Write ("Now your credit card is deactivated, you
ot use it more.")
                                    %>
```

```
                    <goto next = "main.asp"/>
                        <%
                else
                if (rs2("credit_status")) = false then
                Response.Write ("Your credit card is already deactivated.")
                        %>
                        <goto next = "main.asp"/>
                <%
                End if
                End if
        End if
  End if

k>
>
l>
```

KRATES.ASP

```
version="1.0" encoding="iso-8859-1"?>
CTYPE vxml PUBLIC "vxml" "">
version="1.0">
next = "main.asp#home">
    <grammar>home</grammar>
>
next = "main.asp#end">
    <grammar>exit</grammar>
>
id = "stockrates">
    <block>
            Welcome to the city bank update stock rates.
    </block>
    <subdialog src = "#srates"/>
m>
u id = "srates" dtmf="true">
            <prompt count = "1" timeout = "10s">
                    You have just entered the stock page. Please make
                    a selection by speaking one of the following options:
                    <break msecs = "1000" />
                    <enumerate/>
            </prompt>
    <prompt count = "2">
                    Please say one of the following.
                    <break msecs = "2000"/>
                    <enumerate/>
            </prompt>
            <choice next = "srateks.asp#kse">Karachi
            <grammar type="application/x-jsgf">
                    I want to know about Karachi [stock] [exchange] |
                    What is the market situation in karachi |
                    Tell me about karachi [stock] [exchange]
            </grammar>
            </choice>
            <choice next = "srateny.asp#nyse">New York
            <grammar type="application/x-jsgf">
                    I want to know about New York [stock] [exchange] |
```

```
                What is the market situation in New York |
                Tell me about New York [stock] [exchange]
        </grammar>
        </choice>
        <choice next = "sratedj.asp#dj">Doe Jones
        <grammar type="application/x-jsgf">
                I want to know about Doe Jones [stock] [exchange] |
                What is the market situation at Doe jones |
                Tell me about Doe Jones [stock] [exchange]
        </grammar>
        </choice>
</menu>
l>
```

TEKS.ASP

```
l version="1.0" encoding="iso-8859-1"?>
CTYPE vxml PUBLIC "vxml" "">
l version="1.0">
next = "main.asp#home">
    <grammar>home</grammar>
>
next = "main.asp#end">
    <grammar>exit</grammar>
>
ame = "currentoption" expr = "'home'"/>
id="kse">
    <block>
    <%
    Set conn = Server.CreateObject("ADODB.Connection")
    conn.Provider = "Microsoft.Jet.OLEDB.4.0"
    conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
    conn.open
    dim verifycustomerSQL1
    verifycustomerSQL1 = "select * from tbl_stock where stock_id='222'"
    Set rs1 = conn.Execute(verifycustomerSQL1)
    Response.Write ("It is Karachi Stock Exchange. Today stock rate is")
    %>
    <break msecs = "200" />
    <%
    Response.Write rs1("stock_rate")
    Response.Write ("And change in stock is.")
    Response.Write rs1("stock_rate_change")
    %>
    <assign name = "currentoption" expr = "'kse'"/>
    <goto next = "#repeat"/>
ck>
n>
n id ="repeat">
    <field name = "confirm" type = "boolean">
        <prompt>To repeat say yes. Say no, to go back to home.</prompt>
            <filled>
                <if cond = "confirm == true">
                <goto expr = "'#' + currentoption"/>
                <else/>
                <goto next = "main.asp"/>
```

```
                    </if>
                </filled>
        </field>
orm>
xml>
```

ATENY.ASP

```
ml version="1.0" encoding="iso-8859-1"?>
OCTYPE vxml PUBLIC "vxml" "">
ml version="1.0">
k next = "main.asp#home">
        <grammar>home</grammar>
nk>
k next = "main.asp#end">
        <grammar>exit</grammar>
nk>
r name = "currentoption" expr = "'home'"/>
m id="nyse">
        <block>
        <%
        Set conn = Server.CreateObject("ADODB.Connection")
        conn.Provider = "Microsoft.Jet.OLEDB.4.0"
        conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
        conn.open
        dim verifycustomerSQL1
        verifycustomerSQL1 = "select * from tbl_stock where stock_id='333'"
        Set rs1 = conn.Execute(verifycustomerSQL1)
        Response.Write ("It is New York Stock Exchange. Today stock rate is")
        %>
        <break msecs = "200" />
        <%
        Response.Write rs1("stock_rate")
        Response.Write ("And change in stock is.")
        Response.Write rs1("stock_rate_change")
        %>
        <assign name = "currentoption" expr = "'nyse'"/>
        <goto next = "#repeat"/>
ock>
rm>
m id ="repeat">
    <field name = "confirm" type = "boolean">
            <prompt>To repeat say yes. Say no, to go back to home.</prompt>
            <filled>
                    <if cond = "confirm == true">
                    <goto expr = "'#' + currentoption"/>
                    <else/>
                    <goto next = "main.asp"/>
                    </if>
                </filled>
    </field>
m>
ml>
```

ATEDJ.ASP

```
ml version="1.0" encoding="iso-8859-1"?>
OCTYPE vxml PUBLIC "vxml" "">
ml version="1.0">
k next = "main.asp#home">
        <grammar>home</grammar>
nk>
k next = "main.asp#end">
        <grammar>exit</grammar>
nk>
r name = "currentoption" expr = "'home'"/>
rm id="dj">
        <block>
        <%
        Set conn = Server.CreateObject("ADODB.Connection")
        conn.Provider = "Microsoft.Jet.OLEDB.4.0"
        conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
        conn.open
        dim verifycustomerSQL1
        verifycustomerSQL1 = "select * from tbl_stock where stock_id='444'"
        Set rs1 = conn.Execute(verifycustomerSQL1)
        Response.Write ("It is Doe Jones. Today stock rate is")
        %>
        <break msecs = "200" />
        <%
        Response.Write rs1("stock_rate")
        Response.Write ("And change in stock is.")
        Response.Write rs1("stock_rate_change")
        %>
        <assign name = "currentoption" expr = "'dj'"/>
        <goto next = "#repeat"/>
lock>
rm>
rm id ="repeat">
        <field name = "confirm" type = "boolean">
                <prompt>To repeat say yes. Say no, to go back to home.</prompt>
                <filled>
                                <if cond = "confirm == true">
                                <goto expr = "'#' + currentoption"/>
                                <else/>
                                <goto next = "main.asp"/>
                                </if>
                        </filled>
        </field>
rm>
xml>
```

RANSFER.ASP

```
ml version="1.0" encoding="iso-8859-1"?>
OCTYPE vxml PUBLIC "vxml" "">
ml version="1.0">
m id = "ctransfer">
        <var name="mydur" expr="0"/>
        <block>
```

```
        Your call is going to transfer. Wait for a while.
        </block>
ransfer name="mycall" dest="phone://180012345" connecttimeout="30s" bridge="false">
illed>
ssign name="mydur" expr="mycall$.duration"/>
f cond="mycall == 'busy'">
rompt>Sorry, our customer support team is busy serving
ner customers. Please try again later.</prompt>
lseif cond="mycall == 'noanswer'"/>
rompt>Sorry, our customer support team's normal hours
e 9 am to 7 pm Monday through Saturday.</prompt>
if>
filled>
transfer>
lock>
ubmit next="ctransdbv.asp#ctransdbv" method="post" namelist="mycall mydur"/>
block>
form>
vxml>
```

TRANSDBV.ASP

```
xml version="1.0" encoding="iso-8859-1"?>
DOCTYPE vxml PUBLIC "vxml" "">
xml version="1.0">
ink next = "main.asp#home">
        <grammar>home</grammar>
link>
ink next = "main.asp#end">
        <grammar>exit</grammar>
link>
ar name = "currentoption" expr = "'home'"/>
orm id="ctransdbv">
        <block>
        <%
        dim var1call
        dim var2dur
        var1call=Request.Form("mycall")
        var2dur=Request.Form("mydur")
        Response.Write ("My call is")
        Response.Write (var1call)
        %>
        <assign name = "currentoption" expr = "'ctransdbv'"/>
        <goto next = "#repeat"/>
block>
form>
orm id ="repeat">
        <field name = "confirm" type = "boolean">
                <prompt>To repeat say yes. Say no, to go back to home.</prompt>
                        <filled>
                                <if cond = "confirm == true">
                                <goto expr = "'#' + currentoption"/>
                                <else/>
                                <goto next = "main.asp"/>
                                </if>
                        </filled>
```

```
        </field>
orm>
xml>
```

UESTBOOK.ASP

```
xml version="1.0" encoding="iso-8859-1"?>
OCTYPE vxml PUBLIC "vxml" "">
xml version="1.0">
nk next = "main.asp#home">
        <grammar>home</grammar>
ink>
nk next = "main.asp#end">
        <grammar>exit</grammar>
ink>
orm id="guestbook">
 <record  name="greeting" beep="true" maxtime="10s"
   finalsilence="4000ms" dtmfterm="true" type="audio/wav">
  <prompt>
    At the tone, please say your words.
  </prompt>
  <noinput>
    I didn't hear anything, please try again.
  </noinput>
 </record>
 <field name="confirm" type="boolean">
  <prompt>
    Your greeting is <value expr="greeting"/>.
  </prompt>
  <prompt>
    To keep it, say yes.  To discard it, say no.
  </prompt>
  <filled>
    <if cond="confirm">
      <submit next="guestbookdbv.asp#guestbookdbv" method="post" namelist="greeting"/>
    </if>
    <clear/>
  </filled>
 </field>
</form>
vxml>
```

UESTBOOKDBV.ASP

```
'xml version="1.0" encoding="iso-8859-1"?>
'DOCTYPE vxml PUBLIC "vxml" "">
'xml version="1.0">
ink next = "main.asp#home">
        <grammar>home</grammar>
link>
ink next = "main.asp#end">
        <grammar>exit</grammar>
link>
ar name = "currentoption" expr = "'home'"/>
orm id="guestbookdbv">
        <block>
```

```
<%
dim savegreeting
savegreeting = Request.Form ("greeting")
Set conn = Server.CreateObject("ADODB.Connection")
conn.Provider = "Microsoft.Jet.OLEDB.4.0"
conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
conn.open
dim queryinsert1
queryinsert1 = "INSERT INTO tbl_greeting (wav_greeting) VALUES ('"&savegreeting&"')"
set rs2 = conn.Execute (queryinsert1)
Response.Write ("Thank you for visiting City bank guest book.")
%>
<goto next = "main.asp"/>
block>
form>
vxml>
```

URVEY.ASP

```
?xml version="1.0" encoding="iso-8859-1"?>
DOCTYPE vxml PUBLIC "vxml" "">
vxml version="1.0">
ink next = "main.asp#home">
<grammar>home</grammar>
link>
ink next = "main.asp#end">
<grammar>exit</grammar>
link>
ar name = "currentoption" expr = "'home'"/>
orm id = "survey">
<block>
Welcome to the City bank voice survey.
<%
Set conn = Server.CreateObject("ADODB.Connection")
conn.Provider = "Microsoft.Jet.OLEDB.4.0"
conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
conn.open
dim getdata
dim value
value = 1
'Execute Query to get detail
getdata = "select * from tbl_survey where survey_id ='"&value&"'"
Set rs1 = conn.Execute(getdata)
If rs1("survey_title") = "no" Then
%>
<goto next="surveyerror.asp"/>
<%
else
Response.Write rs1("survey_title")
end if
%>
block>
ubdialog src = "#subformsurvey"/>
form>
enu id = "subformsurvey">
<prompt count = "2">
```

```
                    Please say.
                    <break msecs = "2000"/>
                    <enumerate/>
                </prompt>
                <choice next =
ttp://127.0.0.1/VProject/surveydbvyes.asp#surveydbvyes">Yes</choice>
        <choice next = "http://127.0.0.1/VProject/surveydbvno.asp#surveydbvno">No</choice>
        </menu>
vxml>
```

URVEYERROR.ASP

```
'xml version="1.0" encoding="iso-8859-1"?>
DOCTYPE vxml PUBLIC "vxml" "">
'xml version="1.0">
'orm>
        <block>
        There is no survey title yet. Check later. Thanks for visiting
        <goto next="main.asp"/>
        </block>
form>
vxml>
```

URVEYDBVYES.ASP

```
'xml version="1.0" encoding="iso-8859-1"?>
DOCTYPE vxml PUBLIC "vxml" "">
'xml version="1.0">
ink next = "main.asp#home">
        <grammar>home</grammar>
link>
ink next = "main.asp#end">
        <grammar>exit</grammar>
link>
'ar name = "currentoption" expr = "'home'"/>
'orm id="surveydbvyes">
        <block>
        <%
t conn = Server.CreateObject("AD2ODB.Connection")
        conn.Provider = "Microsoft.Jet.OLEDB.4.0"
        conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
        conn.open
        dim value
        value = 1
        dim queryselect1
        queryselect1 = "select * from tbl_survey where survey_id ='"&value&"'"
        Set rs1 = conn.Execute(queryselect1)
        dim temp
        temp = cint(rs1("survey_pos_votes")) + 1
        %>
        You vote successfully
        <%
        Dim sqlupdate1
        sqlupdate1 = "update tbl_survey set survey_pos_votes = '"&temp&"' where
rvey_id='"&value&"'"
        Set rs1 = conn.Execute (sqlupdate1)
```

```
dim queryselect
queryselect = "select * from tbl_survey where survey_id ='"&value&"'"
Set rs1 = conn.Execute(queryselect)
dim tvotes
tvotes = cint(rs1("survey_pos_votes")) + cint(rs1("survey_neg_votes"))
dim pvpercent
pvpercent = (cint(rs1("survey_pos_votes")) * 100) / cint(tvotes)
Response.Write (int(pvpercent))
%>
percent people said yes
<%
dim nvpercent
pvpercent = (cint(rs1("survey_neg_votes")) * 100) / cint(tvotes)
%>
and
<%
Response.Write (int(pvpercent))
rs1.close
set rs1 = nothing
conn.Close
set conn = nothing
%>
percent people said no
<assign name = "currentoption" expr = "'surveydbvyes'"/>
<goto next = "#repeat"/>
block>
form>
orm id ="repeat">
<field name = "confirm" type = "boolean">
<prompt>To repeat say yes. Say no, to go back to home.</prompt>
<filled>
<if cond = "confirm == true">
<goto expr = "'#' + currentoption"/>
<else/>
<goto next = "main.asp"/>
</if>
</filled>
</field>
orm>
xml>
```

RVEYDBVNO.ASP

```
xml version="1.0" encoding="iso-8859-1"?>
OCTYPE vxml PUBLIC "vxml" "">
xml version="1.0">
nk next = "main.asp#home">
<grammar>home</grammar>
nk>
nk next = "main.asp#end">
<grammar>exit</grammar>
nk>
r name = "currentoption" expr = "'home'"/>
rm id="surveydbvno">
<block>
<%
```

```
        Set conn = Server.CreateObject("ADODB.Connection")
        conn.Provider = "Microsoft.Jet.OLEDB.4.0"
        conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
        conn.open
        dim value
        value = 1
        dim queryselect1
        queryselect1 = "select * from tbl_survey where survey_id ='"&value&"'"
        Set rs2 = conn.Execute(queryselect1)
        dim temp
        temp = cint(rs2("survey_neg_votes")) + 1
        %>
        You vote successfully
        <%
        Dim sqlupdate1
        sqlupdate1 = "update tbl_survey set survey_neg_votes = '"&temp&"' where
urvey_id='"&value&"'"
        Set rs2 = conn.Execute (sqlupdate1)
        dim queryselect
        queryselect = "select * from tbl_survey where survey_id ='"&value&"'"
        Set rs2 = conn.Execute(queryselect)
        dim tvotes
        tvotes = cint(rs2("survey_pos_votes")) + cint(rs2("survey_neg_votes"))
        dim pvpercent
        pvpercent = (cint(rs2("survey_pos_votes")) * 100) / cint(tvotes)
        Response.Write (int(pvpercent))
        %>
        percent people said yes
        <%
        dim nvpercent
        pvpercent = (cint(rs2("survey_neg_votes")) * 100) / cint(tvotes)
        %>
        and
        <%
        Response.Write (int(pvpercent))
        rs2.close
        set rs2 = nothing
        conn.Close
        set conn = nothing
        %>
        percent people said no
        <assign name = "currentoption" expr = "'surveydbvno'"/>
        <goto next = "#repeat"/>
block>
form>
orm id ="repeat">
        <field name = "confirm" type = "boolean">
                <prompt>To repeat say yes. Say no, to go back to home.</prompt>
                        <filled>
                                <if cond = "confirm == true">
                                <goto expr = "'#' + currentoption"/>
                                <else/>
                                <goto next = "main.asp"/>
                                </if>
                        </filled>
        </field>
```

```
orm>
xml>
```

MLOCATION.ASP

```
xml version="1.0" encoding="iso-8859-1"?>
OCTYPE vxml PUBLIC "vxml" "">
xml version="1.0">
nk next = "main.asp#home">
        <grammar>home</grammar>
    </link>
    <link next = "main.asp#end">
        <grammar>exit</grammar>
    </link>
ar name = "currentoption" expr = "'home'"/>
rm id = "atmlocation">
ock>
    Here is City Bank A T M location.
    <%
    Set conn = Server.CreateObject("ADODB.Connection")
    conn.Provider = "Microsoft.Jet.OLEDB.4.0"
    conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
    conn.open
    dim verifycustomerSQL1
    'Execute Query to get detail
    verifycustomerSQL1 = "select * from tbl_bank"
    Set rs1 = conn.Execute(verifycustomerSQL1)
    Response.Write rs1("bank_atmlocation")
    %>
    <assign name = "currentoption" expr ="'atmlocation'"/>
    <goto next = "#repeat"/>
    </block>
orm>
rm id ="repeat">
    <field name = "confirm" type = "boolean">
    <prompt>To repeat say yes. Say no, to go back to home.</prompt>
    <filled>
                <if cond = "confirm == true">
                <goto expr = "'#' + currentoption"/>
                <else/>
                <goto next = "main.asp"/>
                </if>
            </filled>
    </field>
orm>
xml>
```

ONTADDRESS.ASP

```
xml version="1.0" encoding="iso-8859-1"?>
OCTYPE vxml PUBLIC "vxml" "">
xml version="1.0">
nk next = "main.asp#home">
        <grammar>home</grammar>
    </link>
    <link next = "main.asp#end">
```

```
                <grammar>exit</grammar>
        </link>
        <var name = "currentoption" expr = "'home'"/>
form id = "contaddress">
block>
        The mailing and contact address of City Bank is
        <%
        Set conn = Server.CreateObject("ADODB.Connection")
        conn.Provider = "Microsoft.Jet.OLEDB.4.0"
        conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
        conn.open
        dim verifycustomerSQL1
        'Execute Query to get detail
        verifycustomerSQL1 = "select * from tbl_bank"
        Set rs1 = conn.Execute(verifycustomerSQL1)
        Response.Write rs1("bank_contaddress")
        %>
        <assign name = "currentoption" expr ="'contaddress'"/>
        <goto next = "#repeat"/>
        </block>
/form>
form id ="repeat">
        <field name = "confirm" type = "boolean">
                <prompt>To repeat say yes. Say no, to go back to home.</prompt>
                <filled>
                                <if cond = "confirm == true">
                                <goto expr = "'#' + currentoption"/>
                                <else/>
                                <goto next = "main.asp"/>
                                </if>
                        </filled>
        </field>
/form>
/vxml>
```

IRECTIONS.ASP

```
?xml version="1.0" encoding="iso-8859-1"?>
!DOCTYPE vxml PUBLIC "vxml" "">
vxml version="1.0">
ink next = "main.asp#home">
                <grammar>home</grammar>
        </link>
        <link next = "main.asp#end">
                <grammar>exit</grammar>
        </link>
ar name = "currentoption" expr = "'home'"/>
orm id = "directions">
        <block>
                <%
        Set conn = Server.CreateObject("ADODB.Connection")
        conn.Provider = "Microsoft.Jet.OLEDB.4.0"
        conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
        conn.open
        dim verifycustomerSQL1
        'Execute Query to get detail
```

```
        verifycustomerSQL1 = "select * from tbl_bank"
        Set rs1 = conn.Execute(verifycustomerSQL1)
        Response.Write rs1("bank_directions")
>
ssign name = "currentoption" expr ="'directions'"/>
        <goto next = "#repeat"/>
        </block>
form>
orm id ="repeat">
        <field name = "confirm" type = "boolean">
                <prompt>To repeat say yes. Say no, to go back to home.</prompt>
                        <filled>
                                <if cond = "confirm == true">
                                <goto expr = "'#' + currentoption"/>
                                <else/>
                                <goto next = "main.asp"/>
                                </if>
                        </filled>
        </field>
form>
vxml>
```

CBALANCE.ASP

```
ml version="1.0" encoding="iso-8859-1"?>
DOCTYPE vxml PUBLIC "vxml" "">
vxml version="1.0">
link next = "main.asp#home">
        <grammar>home</grammar>
link>
link next = "main.asp#end">
        <grammar>exit</grammar>
link>
var name = "currentoption" expr = "'home'"/>
form id = "main">
block>
%
        Dim valuecase
        valuecase = 0
        Randomize
        valuecase = CInt(Int((8 * Rnd()) + 1))
        if valuecase = 1 then
                valuecase = 0
                %>
                <goto next = "#main1"/>
                <%
        elseif valuecase = 2 then
                valuecase = 0
                %>
                <goto next = "#main2"/>
                <%
        elseif valuecase = 3 then
                valuecase = 0
                %>
                <goto next = "#main3"/>
                <%
```

```
        elseif valuecase = 4 then
                valuecase = 0
                %>
                <goto next = "#main4"/>
                <%
        elseif valuecase = 5 then
                valuecase = 0
                %>
                <goto next = "#main5"/>
                <%
        elseif valuecase = 6 then
                valuecase = 0
                %>
                <goto next = "#main6"/>
                <%
        elseif valuecase = 7 then
                valuecase = 0
                %>
                <goto next = "#main7"/>
                <%
        elseif valuecase = 8 then
                valuecase = 0
                %>
                <goto next = "#main8"/>
                <%
        end if
%>
</block>
</form>
<form id = "main1">
        <block>
                Welcome to the account authentication service.
        </block>
        <field name="authentication_check" type="digits">
                <prompt>What is your day of birth?</prompt>
                <catch event="help">
                        Please tell your day of birth.
                </catch>
        </field>
                <block>
                <submit          next="/VProject/acbalancedbv.asp#acbalancedbv"          method="post"
namelist="authentication_check"/>
        </block>
</form>
<form id = "main2">
        <block>
                Welcome to the account authentication service.
        </block>
        <field name="authentication_check" type="digits">
                                        <prompt>What is your month of birth?</prompt>
                                <catch event="help">
                                        Please tell your month of birth.
                                </catch>
        </field>
        <block>
```

```
                <submit          next="/VProject/acbalancedbv.asp#acbalancedbv"          method="post"
namelist="authentication_check"/>
        </block>
</form>
<form id = "main3">
        <block>
                Welcome to the account authentication service.
        </block>
        <field name="authentication_check" type="digits">
                <prompt>What is your year of birth?</prompt>
                <catch event="help">
                        Please tell your year of birth.
                </catch>
        </field>

                <block>
                <submit          next="/VProject/acbalancedbv.asp#acbalancedbv"          method="post"
namelist="authentication_check"/>
        </block>
</form>
<form id = "main4">
        <block>
                Welcome to the account authentication service.
        </block>
        <field name="authentication_check" type="digits">
                <prompt>What is your social security number?</prompt>
                <catch event="help">
                        Please tell your 15 digit social security number.
                </catch>
        </field>
                <block>
                <submit          next="/VProject/acbalancedbv.asp#acbalancedbv"          method="post"
namelist="authentication_check"/>
        </block>
</form>
<form id = "main5">
        <block>
                Welcome to the account authentication service.
        </block>
        <field name="authentication_check" type="digits">
                <prompt>What is city zip code?</prompt>
                <catch event="help">
                        Please tell city zip code which you gave at time of account registration.
                </catch>
        </field>
                <block>
                <submit          next="/VProject/acbalancedbv.asp#acbalancedbv"          method="post"
namelist="authentication_check"/>
        </block>
</form>
<form id = "main6">
        <block>
                Welcome to the account authentication service.
        </block>
        <field name="authentication_check" type="digits">
                <prompt>What is your secret number?</prompt>
```

```
                <catch event="help">
                        Please tell your secret number which you gave at time of account registration.
                </catch>
        </field>
                <block>
                <submit         next="/VProject/acbalancedbv.asp#acbalancedbv"          method="post"
namelist="authentication_check"/>
        </block>
</form>
<form id = "main7">
        <block>
                Welcome to the account authentication service.
        </block>
        <field name="authentication_check" type="digits">
                <prompt>What is your life insurance number?</prompt>
                <catch event="help">
                        Please tell your life insurance number.
                </catch>
        </field>
                <block>
                <submit         next="/VProject/acbalancedbv.asp#acbalancedbv"          method="post"
namelist="authentication_check"/>
        </block>
</form>
<form id = "main8">
        <block>
                Welcome to the account authentication service.
        </block>
        <field name="authentication_check" type="digits">
                <prompt>What is your car license number?</prompt>
                <catch event="help">
                        Please tell your car license number
                </catch>
        </field>
                <block>
                <submit         next="/VProject/acbalancedbv.asp#acbalancedbv"          method="post"
namelist="authentication_check"/>
        </block>
</form>
</vxml>
```

ACBALANCEDBV.ASP

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE vxml PUBLIC "vxml" "">
<vxml version="1.0">
<link next = "main.asp#home">
        <grammar>home</grammar>
</link>
<link next = "main.asp#end">
        <grammar>exit</grammar>
</link>
<var name = "currentoption" expr = "'home'"/>
<form id="acbalancedbv">
        <block>
        <%
```

```
        Set conn = Server.CreateObject("ADODB.Connection")
        conn.Provider = "Microsoft.Jet.OLEDB.4.0"
        conn.ConnectionString = "Data Source=" & Server.MapPath("db/citidb.mdb")
        conn.open
        dim ac
        ac=Request.Form("authentication_check")
        dim verifycustomerSQL1
        'Execute Query to get detail
        dim verifyacsql
        verifyacsql = "select * from tbl_customer where customer_dobday='"&ac&"' or
ustomer_dobmonth='"&ac&"' or customer_dobyear='"&ac&"' or customer_ssn='"&ac&"' or
ustomer_zipcode='"&ac&"' or customer_secretno='"&ac&"' or customer_lifenumber='"&ac&"' or
ustomer_licenseno='"&ac&"'"
        Set rs5 = conn.Execute(verifyacsql)
        If rs5.bof and rs5.eof then
        %>
        <goto next="transerror.asp"/>
        <%
        End if
        %>
        <goto next = "accbalance.asp#main"/>
/block>
/form>
/vxml>
```

APPENDIX B

Terms and Acronyms Related to Voice Technology

Term	Acronym	Definition
Automated Speech Recognition	ASR	The ability of a machine or program to recognize and carry out voice commands.
Computer Telephony Integration	CTI	The use of computers to manage telephone calls. The term is used to describe computerized call center services, such as: recognize a voice, either for authentication or for message forwarding, determine how to process a call using live, recorded voice, or touch tone entered input, and provide interactive voice response (IVR) to callers.
Dial-tone Multiple Frequency	DTMF	A telephone based application that allows a user to input and navigates through the system using the keys on a touch-tone keyboard. Also known as touch-tone-hell (TTH) application.
Interactive Voice Response	IVR	Any telephone-based application that prompts the inbound caller for information using a recorded or synthesized human voice.
Natural Language		Speech-enabled applications in which the call can ask questions or provide information using ordinary sentences.
Public Switched Telephone Network	PSTN	Public telephone networks. Also referred to as "Plain old telephone service". In relation to the Internet, the PSTN furnishes the majority of the Internet's long-distance infrastructure.
Telephony		The technology associated with the electronic transmission of voice, fax, or other information between parties using systems associated with the telephone or a handheld device.
Text-to-speech	TTS	Converts text to audio content. In a voice system, TTS accesses web pages and reads out the data to callers. A mechanical one-way delivery of text information to callers.
Voice Browser		A server-side application that connects the user interface to the portal. Listens for keywords to translate back to the main menu, navigate to URL, or transfer calls.
Voice Portal		The interface between a caller and the information source for delivering web-based data, such as stock quotes, movie listings, and directory assistance from web servers out to callers. Reached through toll-free number or dialing a dedicated network-based access code.
Voice Site		A node on the Voice Web that contains voice-enabled enterprise or e-commerce applications and V-Commerce services.
Voice Over Internet Protocol	VoIP	Voice delivered using the Internet Protocol.

APPENDIX C

Deployed Voice Services

Voice Service	Location	Phone number	Info/Press	Sponsor	Developer / Host / Platform
First Citizen National Bank	Dyersburg, Tennessee		www.syntellect.com		Syntellect
Mashreqbank	United Arab Emirates		www.banktech.com/story/techFocus		Ascent Computing
AT&T Toll Free Directory Assistance	USA	1-800-555-1212	TMAA Speech Recognition Update #100 *(Oct 2001)*	AT&T	Tellme Networks
AT&T Wireless #121 Service	USA	#121	http://www.attws.com/personal/121/	AT&T Wireless Services	Tellme Networks
AT&T Worldnet Web Access By Phone	USA	(demo)	http://download.att.net/WABP	AT&T Worldnet	VoiceGenie *(platform)*
BeVocal Voice Portal	USA	1-800-4BVOCAL	http://www.bevocal.com	BeVocal	BeVocal
Cingular Voice Connect	USA	*8 on Cingular mobile	WirelessDevNet News *(Oct 28, 2001)*	Cingular Wireless	BeVocal
State of Delaware Voice Portal	Delaware, USA	1-866-276-2353	http://www.state.de.us/dti/access_de.htm	Delaware	BeVocal
HeyAnita voice portal	USA	1-800-44-ANITA	http://www.heyanita.com	HeyAnita	HeyAnita
i-TIM Voice Portal	Italy			Telecom Italia Mobile	Loquendo's VoxNauta
LastMinute.com	UK, France	+ 44 870 872 6313		LastMinute.com	Broca Networks
Orange	UK	177 on OrangeUK	Orange Voice Services	Orange	MiLife™ VoiceXML
Qwest Wireless Voice Calling	USA	*WWW on Qwest mobile		Qwest Wireless	BeVocal
Simitel	Latin America		http://www.simitel.com	Simitel	Simitel
Sprint PCS Voice Portal	USA	via *Voice* Command on SprintPCS mobile	HeyAnita press release *(Apr 09, 2001)*	SprintPCS	HeyAnita

Tellme 1-800-555-TELL	USA	1-800-555-TELL	http://www.1-800-555-TELL.COM	Tellme Networks	Tellme Networks
Tiscali by Phone	Italy	Italy	http://byphone.tiscali.it	Tiscali S.p.A	Tiscali S.p.A
Virginia's 511 Traveler Information	Virginia, USA	511	Virginia Department of Transportation press release *(Feb 15, 2001)*	Virginia Department of Transportation	Tellme Networks
Utah's 511 Traveler Information	Utah, USA	511 *(in Utah)* 1-866-511-UTAH *(anywhere in US)*	http://www.utahcommuterlink.com	Utah Department of Transportation	Tellme Networks
Verizon Wireless	USA		Verizon Wireless press release *(Oct 17, 2002)*	Verizon Wireless	HeyAnita
VoiceWeb	Greece	+30 (01) 810-8000	http://www.voiceweb.gr	VoiceWeb S.A.	VoiceWeb S.A.
Voizi	Japan	+81 0088-36-8839	http://www.voizi.net	Japan Telecom	Japan Telecom
Wingcast *(defunct)*	USA		Wingcast press release *(Mar 18, 2002)*	Wingcast	BeVocal
Yahoo by Phone	USA	1-800-MY-YAHOO	http://phone.yahoo.com	Yahoo!	Yahoo! and Nuance

NEAR EAST UNIVERSITY

GRADUATE SCHOOL OF APPLIED
AND SOCIAL SCIENCES

Analysis of the Wireless Communication Systems and Traffic Modeling

Cemal Kavalcıoğlu

Master Thesis

Department of Electrical and Electronic
Engineering

Nicosia-2002

Cemal Kavalcıoğlu: Analysis of the Wireless Communication Systems and Traffic Modeling

Approval of the Graduate School of Applied and Social Sciences

Prof. Dr. Fakhraddin Mamedov
Director

We certify that this thesis is satisfactory for the award of the degree of Master of Science in Electrical and Electronic Engineering

Examining Committee in Charge:

Assoc. Prof. Dr. Adnan Khashman, Chairman of Committee, Chairman of the Electrical and Electronic Engineering Department, NEU

Assist. Prof. Dr. Kadri Bürüncük, Committee Member, Electrical and Electronic Engineering Department, NEU

Assist. Prof. Dr. Doğan Haktanır, Committee Member, Electrical and Electronic Engineering Department, NEU

ACKNOWLEDGEMENTS

First of all, I would like to say how grateful I am to my supervisor, colleagues, friends and family. I could not have prepared this thesis without the generous help of my supervisor, colleaques, friends, and family.

I would like to thank my supervisor Prof. Dr. Fakhreddin Mamedov for his invaluable advice and belief in my work and myself over the course of this MSc Degree. Prof. Dr. Fakhreddin Mamedov supplied the warmth, enthusiasm, and clarity of judgement that every student hopes for. Going beyond the limited role of literary agent, he provided valuable advice at each stage of the preparation of this thesis.

I would like to express my gratitude to Prof. Dr. Şenol Bektaş for him because he provided possibility to me that studying my MSc. Education in Near East University to be Assistant.

I also wish to thank the jury members that consist of Jury Chairman Assoc. Prof. Dr. Adnan Khashman, Jury Member Assist. Prof. Dr. Kadri Büründcük, and Jury Member Assist. Prof. Dr. Doğan Haktanır, because of their acceptance to enter my M.Sc. presentation.

I commenced higher education with Near East University in 1995 and after my graduation; I was employed as an assistant at same University for Programmable Logic Controllers and Power Electronics Laboratories at the Engineering Faculty in 2000. While I was employed with NEU, I took up M.Sc. Degree in "Analysis of the Wireless Communication Systems and Traffic Modeling". Also I wish to thank all engineering staff, friends and colleagues, for their valuable contributes, towards my studies.

Finally, I want to thank my family, especially my parents that is dedicated for them, without their endless support and encouragement, I could never have completed this thesis.

Dedicated to My Father and My Mother

LIST OF ABBREVIATIONS

A	interface between MSC and BSC
Abis	interface between BSC and BTS
Abs	Analysis by synthesis
ADPCM	Adaptive Differential Pulse Code Modulator
AGCH	Access Grant Channel
AIN	Advanced Intelligent Network
AM	Amplitude Modulation
AMPS	Advanced Mobile Phone System
APC	Automatic Power Control
ARC	Automatic Reverse Charge
ARQ	Automatic Request for retransmission
ATIS	Alliance for Telecommunications Industry Solutions
AUC	Authentication Center
BCCH	Broadcast Control Channel
BCH	Broadcast Channel
Bps	Bits per second
BS	Base Station
BSC	Base Station Controller
BSIC	Base Station Identity Code
BSS	Base Station Subsystem
BTS	Base Transceiver System
CAC	Communication Access Channel
CBCH	Cell Broadcast Channel
CCCH	Communication Control Channel
CCF	Call Control Function
CCH	Control Channel
CCITT	Consultative Committee on Telephone and Telegraph
CDL	Coded Digital control channel Locator
CDMA	Code-Division Multiple Access
CDPD	Cellular Digital Packet Data
CDVCC	Coded Digital Verification Color Code
CEPT	Conference Europeenne des Postes et Telecommunications

CNIP	Calling Number Identification Presentation
CNIR	Calling Number Identification Restriction
CRC	Cyclic Redundancy Check
CCIR	International Consultative Committee on Radio
DCC	Digital Color Code
DCCH	Dedicated Control Channel
DECT	Digital European Cordless Telecommunications
DMH	Data Message Handler
DQPSK	Differential Quadrature Phase Shift Keying
EIA	Electronic Industry Association
EIR	Equipment Identity Register
EIRP	Effective Isotropic Radiated Power
ESN	Electronic Serial Number
E-TDMA	Extended Time-Division Multiple Access
ETSI	European Telecommunications Standards Institute
FACCH	Fast Associated Control Channel
FC	Fast Channel
FCC	Federal Communications Commission
FCCH	Frequency Correction Channel
FDD	Frequency-Division Duplex
FDMA	Frequency-Division Multiple Access
FE	Functional Element
FM	Frequency Modulation
FSK	Frequency Shift Keying
FPLMTS	Future Public Land Mobile Telecommunications System
GHz	gigahertz
GMSK	Gaussian Minimum Shift Keying
GSM	Global System for Mobile communications
HAAT	Height Above Average Terrain
HLR	Home Location Register
IAM	Initial Address Message
ID	Identification
IEEE	Institute of Electrical and Electronic Engineers
IMSI	International Mobile Station Identification

IMTS	Improved Mobile Telephone Service
IS	Interim Standard
ISDN	Integrated-Service Digital Network
IWF	Interworking Function
JDC	Japanese Digital Cellular
kHz	kilohertz
LAI	Location Area Identity
LAN	Local Area Network
LPC	Linear Predictive Coding
MAP	Mobile Application Part
MC	Multiple Carriers
Mcps	Million chips per second
MHz	Megahertz
MLPP	Multilevel Precedence and Preemption
MS	Mobile Station
MSC	Mobile Switching Center
MT	Mobile Termination
N-AMPS	Narrowband-Advanced Mobile Phone Service
NSS	Network Switching Subsystem
OAM&P	Operation, Administration, Maintenance, and Provisioning
OMC	Operation Maintenance Center
OMSS	Operation and Maintenance Subsystem
OS	Operations Systems
OSS	Operational Subsystem
PACS	Personal Access Communications System
PAD	Packet Assembler/Disassembler
PBX	Private Branch Exchange
PCC	Power Control Channel
PCH	Paging Channel
PCM	Pulse Code Modulation
PCP	Power Control Pulse
PCS	Personal Communications Services
PCSC	Personal Communications Switching Center
PIN	Personal Identification Network

PLMN	Public Land Mobile Network
PMC	Personal Mobility Controller
PMD	Personal Mobility Data store
PN	Pseudonoise
PS	Personal Station
PSC	PCS Switching Center
PSPDN	Public Switched Packet Data Network
PSTN	Public Switching Telephone Network
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RACF	Radio Access Control Function
RACH	Random Access Channel
RASC	Radio Access System Controller
RCF	Radio Control Function
RELP	Residual Excited Linear Prediction
RES	Radio Equipment System
RF	radio frequency
RP	Radio Port
RPE-LTP	Regular Pulse Excited-Long-Term Predictive
RPI	Radio Port Intermediary
RPT	Radio Personal Terminal
RS	Radio System
SACCH	Slow Associated Control Channel
SAT	Supervisory Audio Tone
SC	Slow Channel
SCH	Synchronization Channel
SDCCH	Stand-alone Dedicated Contol Channel
S/I	Signal-to-Interference ratio
SIM	Subscriber Identity Module
SMR	Specialized Mobile Radio
SRF	Specialized Resource Function
SSD	Shared Secret Data
SSF	Service Switching Function
TACS	Total Access Communications System

TCH	Traffic Channel
TCH/F	Traffic Channel/Full rate
TCH/H	Traffic Channel/Half rate
TDMA	Time-Division Multiple Access
TDD	Time-Division Duplex
TE	Terminal Equipment
TIA	Telecommunications Industry Association
TMC	Terminal Mobility Controller
TMD	Terminal Mobility Data store
TMSI	Temporary Mobile Station Identification
UMTS	Universal Mobile Telecommunications System
UPCH	User Packet Channel
UPT	Universal Personal Telecommunications
USC	User Specific Channel
VLR	Visitor Location Register

ABSTRACT

Wireless Communication was first developed to enable communication of ships on the sea. Wireless Communications technology has evolved along a logical path, from simple first-generation analog products designed for business use to second-generation digital wireless telecommunications systems for residential and business environments. The International Telecommunications Union (ITU) started work a decade ago to create a "framework" for new systems in which true global coverage could be achieved. This scheme is now known as the International Mobile Telecommunications 2000 (IMT-2000). According to the International Telecommunication Union (ITU) International Mobile Telecommunications 2000 initiative ("IMT-2000") third generation mobile ("3G") system services are scheduled to be initiated around the year 2000, subject to market considerations.

This thesis presents "Analysis of the Wireless Communication systems and Traffic Modeling".

Personal Communications Systems (PCS) is a name given to wireless systems that are starting to operate in the 1800 MHz frequency band. Initially the concept was that these systems would be very different than cellular; better, cheaper, simpler.

Global System for Mobile (GSM) is one of the most advanced digital cellular communication systems in the world today. GSM is an integrated system that can support voice communication (mobile telephony), data and Short Message Services.

Personal Digital Cellular (PDC) is the standard digital cellular system in Japan and was built as a high capacity system to replace the Japanese analog cellular system (J-TACS). One of the main differences between this and other digital systems is the use of a frequencies in the 1500 MHz band thus this system requires special radio hardware. PDC was also originally named as a JDC (Japan Digital Cellular).

When designing of any Wireless System as a case study the planning and engineering of a cellular radio system, including engineering philosophy, engineering considerations, quality of service criteria, and types of analyses must be taken into consideration at each design.

TABLE OF CONTENTS

HISTORICAL INVENTIONS AND EVENTS IN

COMMUNICATION SYSTEMS&WIRELESS COMMUNICATION

- **1799** Invention of Telegraph by Samuel Morse.
- **1844** First active telegraph lines between Washington and Baltimore.
- **1858** The first transatlantic cable between US and Europe.
- **1870's** Invention of telephone by Alexander Graham.

Introduction of wireless communication.

- Wireless communication was first developed to enable communication of ships on the sea.
- **1906** Human voice was transferred succesfully over radio for the first time.
- **1915** The invention of mobile radios.
- **1921** In the Detroit police department the first Vehicular mobile radio was used(One way communication system).
- **1930s** First half duplex mobile communication systems were introduced in U.S.
- **1935** Invention of FM (Frequency Modulation).
- **1969** Nordic countries made an attempt to standardise the Telecommunication aspects of that countries.
- **1973** That group (NMT group) specifies a feature allowing mobile telephones to be located within different networks. This is the beginning of the roaming concept.
- **1979** The installation and testing of first cellular systems were authorised by FCC.
- **1981** The installation of first cellular systems in the world which was using an analog system called NMT (North Mobile Telephony).

Introduction of GSM

- **1981** A group of specialists was formed to determine a series of standards for Mobile communications by Conference of European Posts and Telecommunications (CEPT). This group was called Groupe Speciale Mobile.
- The primitive aims of this comitee was as follows:
  * Spectrum efficiency
  * International roaming
  * Low mobile and base stations costs
  * Good subjective voice quality
  * Compatibility with other systems such as ISDN (Integrated Services Digital Network)
  * Ability to support new services
- **1989** The responsibility of GSM was passed from CEPT to ETSI(**European Telecommunications Standards Institute**)