MOBILE VOTING FOR COLLEGE

PROJECT REPORT

Submitted By

N.SIVASANKARAN

Register No: 0720300044

in partial fulfilment for the award of the degree

of

MASTER OF COMPUTER APPLICATIONS

in

COMPUTER APPLICATIONS

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

MAY 2010

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Coimbatore)

COIMBATORE – 641 006.

Department of Computer Applications

PROJECT WORK

MAY 2010

This is to certify that the project entitled

MOBILE VOTING FOR COLLEGE

is the bonafide record of project work done by

N.SIVASANKARAN

Register No: 0720300044

of MCA (Computer Applications) during the year 2009-2010.

06/05/10

Project Guide

Head of the Department

Submitted for the Project Viva-Voce examination held on 17-05-10.

17/5/10

Internal Examiner

17/5/10

External Examiner

DECLARATION

I affirm that the project work titled **MOBILE VOTING FOR COLLEGE** being submitted in partial fulfilment for the award of **MASTER OF COMPUTER APPLICATIONS** is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

(Signature of the Candidate)

N.SIVASANKARAN
0720300044

I certify that the declaration made above by the candidate is true

Signature of the Guide,

06/05/10

Ms.R.VEENASHREE , Lecturer

MATRIX Cube
Technologies

www.matrixcubes.com

03/05/2010
CHENNAI

TO WHOMSOEVER IT MAY CONCERN

This is to certify that N.Sivasankaran MCA Final year student of Kumaraguru College of Engineering, Coimbatore has completed the Project work in “Mobile Technology – IVR” during the period December 2009 to April 2010. His Performance during this period is good.

For Matrix Cube Technologies

ACKNOWLEDGEMENT

First and foremost I thank the Almighty for his continuous blessings showered on me in completing this project successfully.

I wish to express my profound gratitude to **Dr. S.Ramachandran, Principal, Kumaraguru College of Technology, Coimbatore** for providing an opportunity and necessary facilities in carrying out this project work.

I am very glad to extend my deepest thanks to **Dr. A. Muthukumar, Course Coordinator, Department of Computer Applications, Kumaraguru College of Technology, Coimbatore,** who has always been a source of inspiration.

I wish to express my unending gratitude to my coordinator **Mr. S. Hameed Ibrahim, Senior Lecturer, Department of Computer Applications** for his unrelenting help, support and advice with regards to my project work. I owe him my profound admiration and respect.

I also take this opportunity to thank my Project guide **Ms.R.Veena Shree, Lecturer, Department of Computer Applications, Kumaraguru College of Technology, Coimbatore** for her continuous support.

I also wish to thank **Mr. C.Arun kumar, Project Manager, Matrix Cube Technologies, Chennai** for giving me inspiration and support for doing my project.

I wish to thank all my teaching and supporting staff members for their timely help and guidance to complete the project successfully.

Also I would like to thank my parents, friends, and all those who helped me in this project and whose names are leftover.

TABLE OF CONTENTS

CHAPTER **PAGE NO**

ABSTRACT

College Elections are increasingly popular among college students. Even college institutions encourage the students to participate in college elections since it provides an ideal opportunity for the youth to develop social responsibility.

Mobile communication systems of today provide increased availability, miniaturization and enhanced data rates. This emergence is expected to enhance the deployment of mobile systems in many services. One area of activity is mobile Election (one of the first areas of transaction using mobile devices).

In this project entitled **Mobile Voting for College**, Mobile technology-IVR is used for election polling. Students have to call a number allocated for the voice applications to run. He can choose to vote or result based on his choice. The selection procedure is implemented through Interactive Voice Response system. He has to answer security questions before casting his vote. Once the student is authorized, he can continue with the polling process. After that, IVR will prompt the various candidates contesting in elections. Based on the User inputs, polling will be recorded. User inputs can either be DTMF or Speech Recognition.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IVR - Interactive Voice Response

IVRS - Interactive Voice Response System

XML - Extensible Markup Language

VXML - Voice Extensible Markup Language

TTS- Text-To-Speech.

ASR- Automatic Speech Recognition.

DTMF-Dual-Tone-Multi-Frequency.

CHAPTER-1
INTRODUCTION

1.1 ORGANIZATION PROFILE:

Matrix Cube Technologies is a provider of Information technology services, founded by experienced software professionals with business knowledge and technical expertise. They are specialized in high-end technologies like Embedded Systems, VLSI Design and VoiceXML Applications for education, engineering and industrial users.

The company is geared to offer a broad range of IT services, ranging from advising clients on strategic technology plans to developing and implementing appropriate IT applications. Matrix Cube Technologies will provide custom-tailored solutions based on an assessment of each client's needs.

They provide services in various technologies that includes

- ✓ Embedded systems.
- ✓ Voice XML.
- ✓ VLSI.
- ✓ Open Systems.
- ✓ Web Designing.

Their clients include

- ✓ ACS Automative.
- ✓ KnowSys.
- ✓ Microcode.

1.2 PROJECT OVERVIEW:

College elections are nowadays common among academic institutions. College elections can be conducted within a single department or for a whole college. They may use paper voting system or electronic voting system. Both are time consuming process. Mobile voting system will be accurate enough to address the issues like vote counting and provides high level of security.

Earlier in paper voting system it takes lot of time for vote counting and there is no enough flexibility for students to cast his vote. Voter turnout was also less because earlier system didn't address the issues like confidentiality, security etc.,

The followings are requirements that should be considered for secure and trustworthy e-voting.

- The relation between voter and vote content should not be revealed.
- The result of a poll should be retained as a secret before counting the ballot.
- A voter can cast his vote just one time.
- Only an identified person by an mobile voting device can participate in mobile voting.
- Other people excepting a voter himself should not know the vote content.
- It is not permitted to rig the ballot.

When the user enters the system he will be asked whether he wants to vote or know results. If the user chooses to vote he has to go through various processes like voter authentication and vote validation. The system restricts the user from casting his vote second time for a particular election.

If the user chooses result he will be asked to select College result or Department wise results. If he chooses department wise result he has to provide the system with department name. The system notifies the user with the winner based on his choice with details including total votes secured and his margin of victory.

CHAPTER-2
SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

College elections are conducted through paper voting and department elections winner are selected by a handful of representatives. These systems allow the user to cast his vote but do not address wrong candidate selections and loss of votes, reducing manual work during counting.

2.1.1 DRAWBACKS OF EXISTING SYSTEM:

- ✓ Voters need to wait in a long queue to cast their vote.
- ✓ Voters have to cast their vote within a particular allotted time.
- ✓ This system does not encourage the participation of nationals living in abroad.
- ✓ Lots of manual work and time has to put in to announce the voting results.
- ✓ The system is not secure enough to address the reliability issues like duplicate voting.

2.2 PROPOSED SYSTEM

Our system will be a solution to the existing drawbacks mentioned above and address various issues like vote counting, voter authentication etc.,

2.2.1 FEATURES OF THE PROPOSED SYSTEM:

- ✓ Uses Speech recognition technique to handle user requests and response.
- ✓ The system is designed as such to be interactive and user friendly.
- ✓ The voter is allowed to cast their vote or to know the results depending upon their choice of selection.
- ✓ The system authenticates the user using security information's provided by him.

2.2.2 ADVANTAGES OF THE PROPOSED SYSTEM:

- ✓ Increases voter turnout.
- ✓ Serve as a reliable and secured system.
- ✓ Announcement of results can be made easier.
- ✓ Allows the voter to vote from anywhere.
- ✓ Provides confidential voting.

CHAPTER-3
DEVELOPMENT ENVIRONMENT

3.1 HARDWARE REQUIREMENTS

System Type	: X86-based PC.
Physical Memory	: 512 MB RAM.
Processor	: Intel core 2 duo.
Hard Disk Space	: 160 GB.
Architecture	: 32-bit.

3.2 SOFTWARE REQUIREMENTS

Operating System	: Windows XP or higher, Linux
JAVA	: version-1.5 or higher.
Development IDE	: NETBEANS 6.0 or higher
Database Server	: MySQL 5.1 or higher
Voice Server	: Voxeo Prophecy 8.0 or higher
Soft Phone	: Voxeo Sip Phone.

3.3 PROGRAMMING ENVIRONMENT

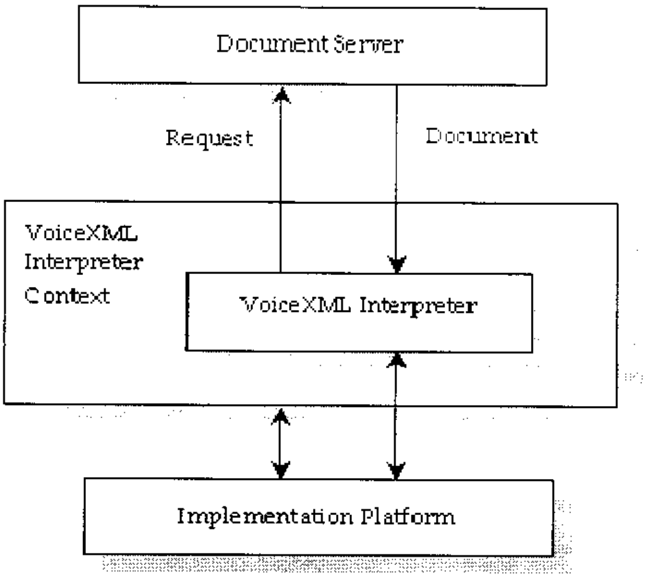
3.3.1 OVERVIEW OF MOBILE TECHNOLOGY - IVRS:

The Interactive Voice Response (IVR) System serves as a bridge between people and computer databases by connecting the telephone network with the database. The telephone user can access the information from anywhere at any time simply by dialing a specified number and following an on-line instruction when a connection has been established.

The IVR system uses pre-recorded or computer generated voice responses to provide information in response to an input from a telephone caller. The input may be given by means of touch-tone or Dual Tone Multi-Frequency (DTMF) signal, which is generated when a caller Presses a key of his/her telephone set, and the sequence of messages to be played is determined dynamically according to an internal menu structure (maintained within the IVR application program) and the user input.

3.3.2 VOICEXML

VoiceXML (VXML) is the W3C's standard XML format for specifying interactive voice dialogues between a human and a computer. It allows voice applications to be developed and deployed in an analogous way to HTML for visual applications. Just as HTML documents are interpreted by a visual web browser, VoiceXML documents are interpreted by a voice browser. A common architecture is to deploy banks of voice browsers attached to the Public Switched Telephone Network (PSTN) to allow users to interact with voice applications over the telephone.



3.1 voicexml architecture

A *document server* (eg. a web server) processes *requests* from a client application, *the VoiceXML Interpreter*, through the *VoiceXML interpreter context*. The server produces *VoiceXML documents* in reply, which are processed by the VoiceXML Interpreter.

The VoiceXML interpreter context may monitor user inputs in parallel with the VoiceXML interpreter. For example, one VoiceXML interpreter context may always listen for a special escape phrase that takes the user to a high-level personal assistant, and another may listen for escape phrases that alter user preferences like volume or text-to-speech characteristics.

The *implementation platform* is controlled by the VoiceXML interpreter context and by the VoiceXML interpreter. For instance, in an interactive voice response application, the VoiceXML interpreter context may be responsible for detecting an incoming call, acquiring the initial VoiceXML document, and answering the call, while the VoiceXML interpreter conducts the dialog after answer.

The implementation platform generates events in response to user actions (eg. spoken or character input received, disconnect) and system events (eg. timer expiration). Some of these events are acted upon by the VoiceXML interpreter itself, as specified by the VoiceXML document, while others are acted upon by the VoiceXML interpreter context.

3.3.3 VXML COMPONENTS

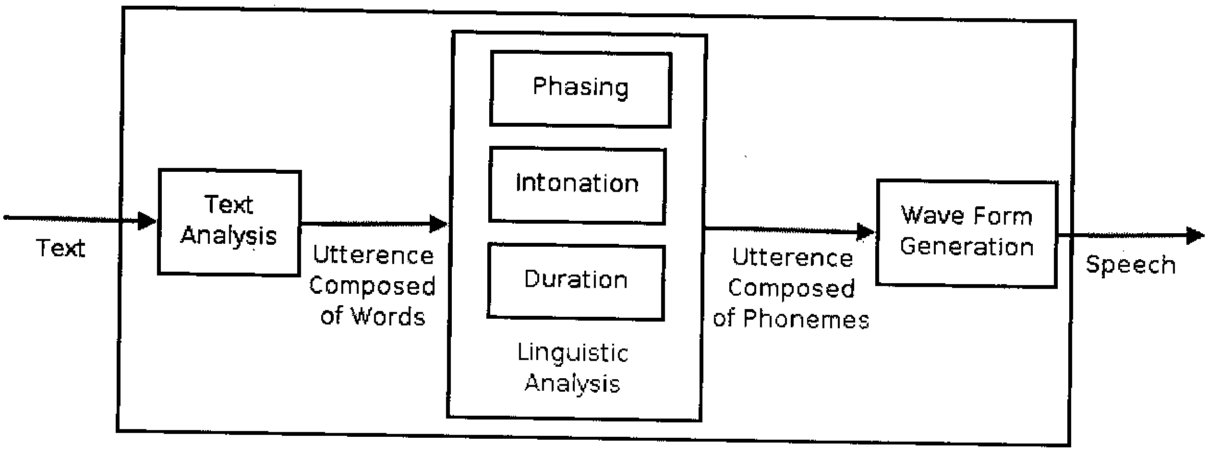
3.3.3.1 PSTN -Public Switched Telephone Network.

The network which is used for connecting telephone calls. PSTN is the world's collection of interconnected voice-oriented public telephone networks, both commercial and government-owned. It's also referred to as the Plain Old Telephone Service (POTS). It's the aggregation of circuit-switching telephone networks that has evolved from the days of Alexander Graham Bell ("Doctor Watson, come here!"). Today, it is almost entirely digital in technology except for the final link from the central (local) telephone office to the user.

3.3.3.2 TTS-Text To Speech Engine

Text to speech is a way of playing dynamic text to user where as the .wav file will play only static (recorded) message.

For example if you want to play standard greeting message "Welcome to Mobile Voting" then we can play greeting.wav to the customer. If we want to play the current balance of your bank account then it is better to go with TTS. Just pass the text and it will convert that to voice and that will be played to the Voter.



3.2 Text-To-Speech Process

3.3.3.3 ASR-Advanced Speech Recognition.

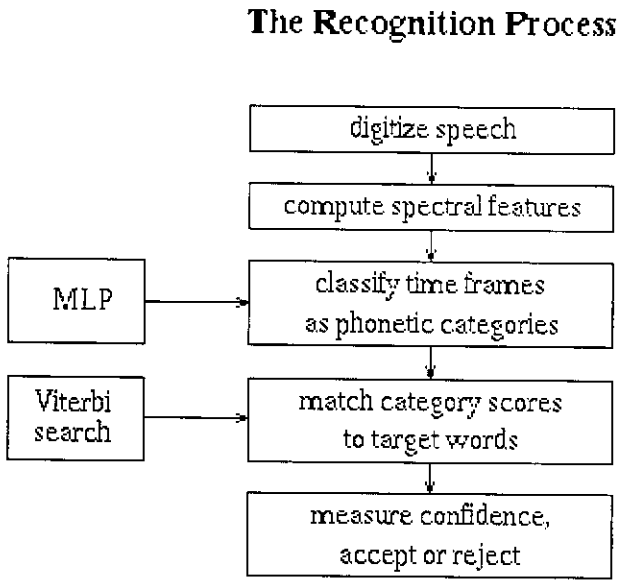
Advanced Speech Recognition (ASR) is what allows your customers to move beyond touch-tone responding and basic verbal yes/no commands. While the idea of ASR seems like a fairly straightforward concept, making that idea a reality is a very involved dynamic process.

Speech Recognition is an innovative technological platform for language processing and intelligent dialogue management. Speech Recognition engine allows you to communicate with your clients virtually, in real time, by chat or by IVR. It has the ability to answer questions and effectively provide appropriate answers as if it was an actual live human. Speech Recognition is an innovative technological platform for language processing and intelligent dialogue management. The Speech Recognition engine allows you to communicate with your clients virtually, in real time, by chat or by IVR. Speech Recognition Engine has the ability to answer questions and effectively provide appropriate answers as if it was an actual live human.

Voice recognition is easier when any of these restrictions is true:

- Limited, predefined vocabulary of possible words and phrases.
- Small number of speakers, with training for each individual's speech and accent.
- Words separated by pauses.

If at least one of these restrictions is true, then voice recognition can be done with reasonable accuracy using a Pentium III class processor. Understanding continuous speech with an unlimited vocabulary, from novel speakers, is beyond the current state of the art.


The Recognition Process
digitize speech
compute spectral features
MLP
classify time frames as phonetic categories
Viterbi search
match category scores to target words
measure confidence, accept or reject

3.3 Speech Recognition Process

3.3.4 JSP OVERVIEW

Architecturally, JSP may be viewed as a high-level abstraction of Java servlets. JSP pages are loaded in the server and operated from a structured special installed Java server packet called a Java EE Web Application, often packaged as a .war or .ear file archive.

JSP allows Java code and certain pre-defined actions to be interleaved with static web markup content, with the resulting page being compiled and executed on the server to deliver an HTML or XML document. The compiled pages and any dependent Java libraries use Java bytecode rather than a native software format, and must therefore be executed within a Java virtual machine (JVM) that integrates with the host operating system to provide an abstract platform-neutral environment.

JSP syntax is a fluid mix of two basic content forms: *scriptlet elements* and *markup.* Markup is typically standard HTML or XML, while scriptlet elements are delimited blocks of Java code which may be intermixed with the markup. When the page is requested the Java code is executed and its output is added, in situ, with the surrounding markup to create the final page. Because Java is a compiled language, not a scripting language, JSP pages must be compiled to Java bytecode classes before they can be executed, but such compilation generally only occurs once each time a change to the source JSP file occurs.

3.3.5 VOICEXML USING JSP

Using Java for outputting VXML offers you several advantages:

- **Flexibility.** You can store your VXML in several different mediums, from a static file to compiled code.
- **Reactivity.** You can react to users' choices by dynamically generating VXML as needed.
- **Extensibility.** It's easy to take a core servlet that outputs basic VXML and extend it with other servlets, each of which provide custom behavior.

When creating a dynamic VoiceXML, you include three types of files:

- **Static VoiceXML page**

  The VoiceXML page acts like any other VoiceXML page, except it contains the <submit> tag file that calls the JavaServer Page.

- **Dynamic JavaServer Page**

  The JSP is a VoiceXML file with dynamic content, meaning it can change, depending on the input of the user. The JSP sends its dynamic content back to the VoiceXML file. Although you can embed any amount of Java™ code within a JSP, it is more efficient to include the processing logic within reusable JavaBeans components.

  Note: If you develop JSPs, these pages still delegate the bulk of the processing to component beans (described next), but they also contain some conditional logic to respond to a actions of the user. But these pages do not contain presentation logic to display UI elements. This task is externalized into separate JSPs, which is invoked as needed by the controller.

- **JavaBeans**

  JavaBeans are created with a Java development tool. They are reusable software programs that store processing logic and process data allowing you to build applications. JavaBeans are a Java class that follow a set of naming and design conventions outlined by the JavaBeans Specification. You have the option of referencing beans and their methods directly through JSP scripting elements.

  - The design conventions include implementing a constructor that includes no arguments, which is used by the JSP to instantiate the bean. Also, access methods are necessary for the JSP to interact with bean objects. These access methods must conform to appropriate naming conventions and begin with either "set" or "get". For example: public int getInfo(); and public void setInfo(String info); are access methods used in a JSP and defined in a bean.
  - Naming conventions do not require that "bean" be at the end of the class name, but it is recommended.

CHAPTER-4

SYSTEM DESIGN

The most important and challenging phase of the system life cycle is system design. The design focuses on the detailed implementation of the system. The first step in system design phase is to determine how the outputs are produced and in what format. Secondly, input data and the tables have to be designed to meet the requirements of proposed system.

System Design is the process of developing specifications for a system that meet the criteria established in system analysis. The major step in design is the preparation of input design and design of output reports in a form acceptable by the user. System Design includes input to the system and the speed of retrieval of data.

Design is the first step of the development of the system. Design will be perfect only if the data collection is done properly with out errors. Design is the base of the development of the system. While designing all possibilities of the system is taken into consideration.

This chapter describes the data model and process model of the application.

4.1 SERVER MODULE DESCRIPTION:

4.1.1 Voting Device (Mobile/Land Phone):

In our scheme, the voting device corresponds to the mobile equipment (ME), which consists of a SIM card. The device needs to provide a platform to run the voting application, which consists of the candidate information, the key storage and generation functions.

4.1.2 Authentication Centre (AC):

AC is an entity within the GSM network. As described in Section 2.1, AC generates the authentication parameters and authenticates the mobile equipment. Apart from authenticating the mobile equipment, AC is also an important information distribution server in the proposed scheme. AC needs to be trusted to transfer the messages as required.

4.1.3 Verification Server (VS):

VS belong to the voting authority, who organizes the voting event. It verifies the legitimacy of the voter and issues a voting token to the voter. VS also publish a list of voter information.

4.2 Overview

In this section, we outline our mobile voting scheme.

4.2.1 Voters Authentication Phase:-

In this phase Election commission is responsible to verify the authenticity of the voters. If the voter is authentic then only he will be allowed to participate in the next steps of voting.

4.2.2 Voting Phase:

In this phase, the voter dial a designated phone number and follow the instructions using a password already provided.

- The voter selects the appropriate option to vote the particular candidate.

- The voting machine checks whether the student has already casted his vote for this election and then allow to cast his vote if he has not casted already.

4.2.3 Counting Phase:

Counting Phase checks the database of vote cast details and checks whether the voting token is valid or not. If it is valid it will be counted else it will be rejected.

4.2.4 Security Analysis

In this section, we discuss how and to what extent the security requirements.

- Authentication only the authorized voters can vote.

First, voters are authenticated through student ID, which assures that voters are who they claim to be.

Second, the eligibility of voters is checked by VS. This prevents voters from voting more than once.

- Privacy all votes remain secret while the voting takes place and each individual vote cannot be linked to the voter who casts it. The proposed scheme is divided into three phases, and they are separated in time.

4.3 MOBILE VOTING SPEECH PROCESS

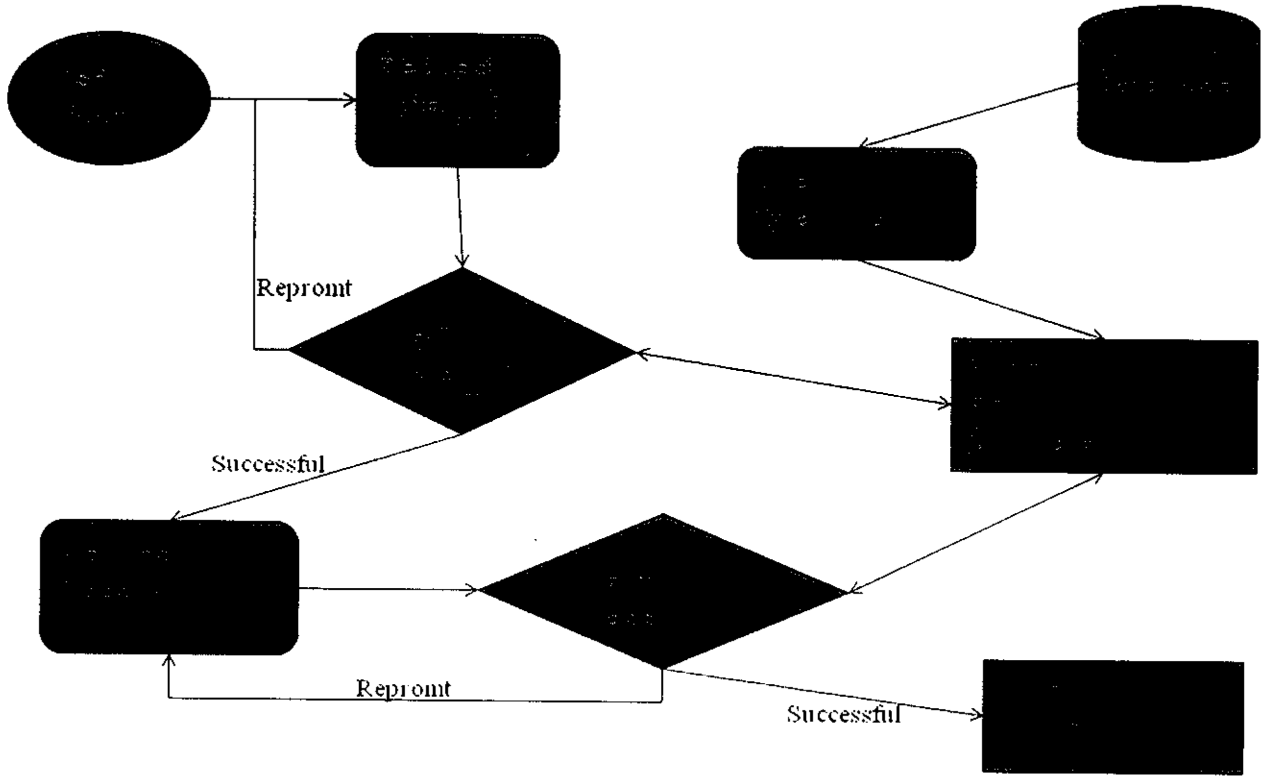
In the below diagram voter authentication process is diagrammatically explained about how the speech recognition software loads the grammar from the database and matches it with the user input.

The system will prompt the user for his voter Id and simultaneously loads the grammar from the database to the speech recognition system. Once the grammar is loaded, the speech recognizer starts to look for the exact utterance from the user so that it can find a match in the grammar.

Interaction with the users will be handled by voicexml documents. The dynamic behavior of the user interaction like loading grammar from the database, responding user queries based on the user input will be handled by java server pages.

After it finds voter id match, the system now prompts the user for his password. If his voter id and password matches then the user will be asked some security questions to provide voter authentication.

Finally he will be allowed to cast his vote in two different modes via touch tone or speech recognition.



4.1 SPEECH RECOGNITION PROCESS IN MOBILE VOTING

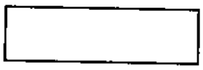
4.4 DATA FLOW DIAGRAM-VOTE CASTING

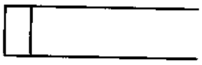
Data flow diagrams are commonly used during problem analysis and design. A DFD shows the flow of data through a system. If views the system as a function that transforms the input into desired outputs. A DFD aims to capture the transformation that takes place within a system into output data so that eventually the output data is produced.

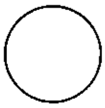
The agent that performs the transformation of data from one state to another is called a process (Bubble). The processes are shown by named circles and dataflow are represented by named arrows. A square defines a source or destination of system data. An open rectangle is a data source.

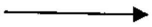
The dataflow diagram can be used as a tool for business planning and strategic planning. It is know as 'bubble chart' o 'data flow graphs' or 'program graph'. The components of the dataflow diagram are the process, the flow, the store and the terminator.

The representation of the four components of dataflow diagram is:

 - The terminator of the dataflow diagram.

 - The tables that are used in the system.

 - The process that are used in the system.

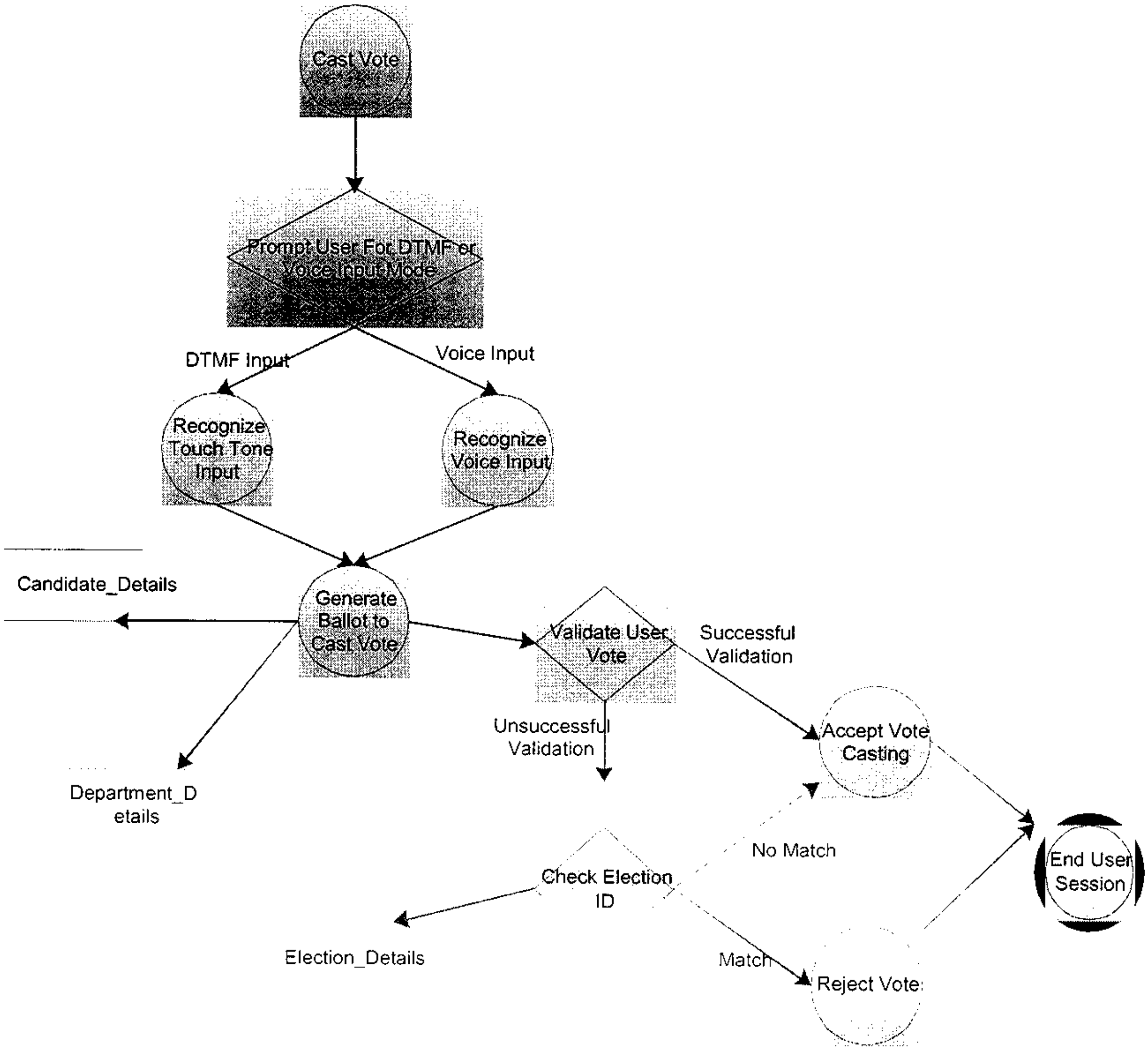
 - The flow of data in the system.

Vote casting process is explained using the data flow diagram. It depicts the data flow between various entities. Once the user is authenticated to cast his vote he will be prompted to choose touch tone mode or speech recognition mode.

Once he entered into a mode he will be asked for his constituency name or constituency number with respect to the mode he chose. For example, he has to know his constituency number if he chooses to vote in touch tone mode.

Depends on the Constituency he chose, the ballot will be generated with the list of candidates contesting in the constituency and their respective symbols. The ballot will be generated with the help of information available in the tables Party Details and Candidate Details.

He can vote for the candidate by saying the symbol of the candidate or saying the candidate number. The system will check his vote validity like duplicate votes and illegal votes before accepting his vote. If the system finds anything wrong with his validation of vote, his vote will be rejected.



4.2 DFD-Vote Casting

4.5 USE CASE DIAGRAM

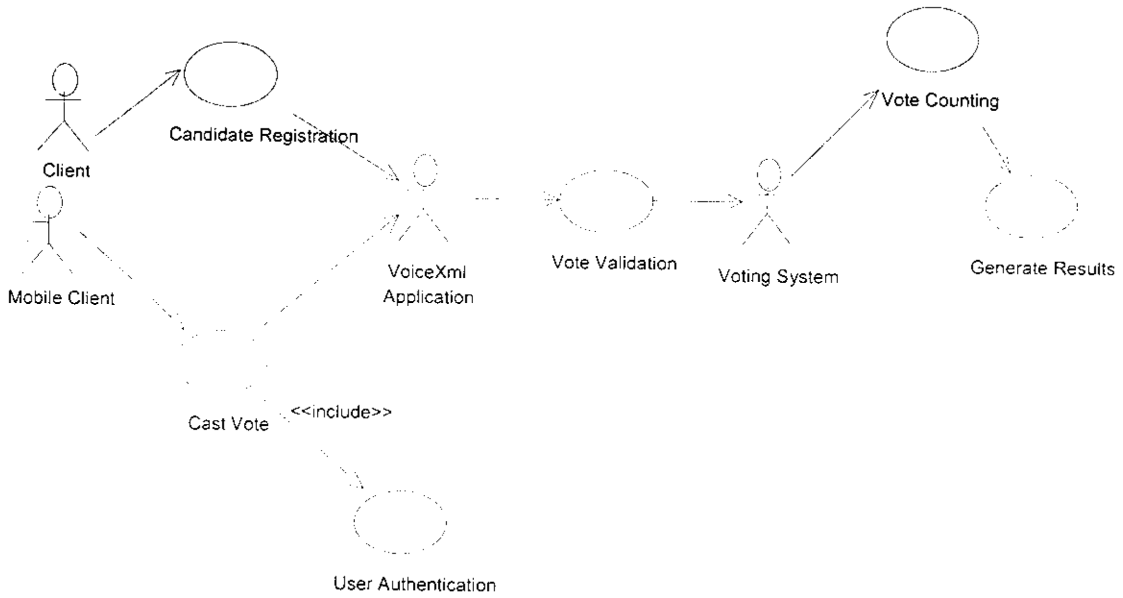
Use case diagram captures the user interaction with the system. The diagram depicts the system overview of how the user interacts with the system and external entities, the system have to depend upon.

When the user enters the system he will be asked whether he wants to vote or know results. Once the user selects his choice he will be directed accordingly.

If the user chooses to vote he has to go through various processes like voter authentication and vote validation as explained in the above diagram. The system restricts the user from casting his vote second time for a particular election and also do not accept illegal votes like forcing someone to cast his vote against their wish.

If the user chooses result he will be asked to select overall result or constituency wise results. Selecting overall result notifies the user about the winner of the current election and his margin of victory with respect to other parties.

If he chooses Constituency wise result he has to provide the system with constituency name. The system notifies the user with the winner of the constituency, total votes secured and his margin of victory with respect to other candidates.



4.3 Voter Login Use Case

4.6 DATABASE DESIGN

Database design is designed to manage large bodies of information. These large bodies of information do not exist in isolation. Database design mainly involves the design of the data base schema. The design of the complete database application environment meets the needs of the enterprise being modeled requires attention to a broader set of issues.

The process of moving from an abstract data model to the implementation of the database proceeds in two design phases. In the logical schema, the designer maps the high level conceptual schema onto the implementation data model of the database system that will be used. The resulting specific database schema will be used in the subsequent physical design phase.

Normalization:

Normalization is a method for designing a relational database design. This will generate a set of relational schemas that allow to store information without unnecessary redundancy, yet also to retrieve information easily. The approach is to design schemas that are in appropriate normal form.

First normal form:

A relation schema will be in first normal form if and only if all the attributes of the relation are atomic in nature.

Second normal form:

A relation schema is said to be in second normal form if and only if it is in the first normal form and no partial dependency exists between key attributes and non key attributes.

Third normal form:

A relation is said to be in third normal form if and only if it is in second normal form and no transitive dependency exists between key attributes and non key attributes.

Boyce-codd normal form:

A relation is said to be in Boyce-codd normal form if and only if all the determinants are candidate keys. Boyce-codd normal form is a strong third normal form, but not every third normal form is Boyce-codd normal form.

Fourth normal form:

A relation is said to be in fourth normal form if and only if it is in third normal form and whenever there exists a multivalued dependency in relation, say A->->B, then all attributes of relation are also functionally dependent on A.

Fifth normal form:

A relation is said to be in fifth normal form if and only if it is in fourth normal form and every join dependency in relation is implied by the candidate key of relation. It is also called projection join normal form.

The table design of this project has been normalized up to second normal form.

Table Name: Student_Details

This table stores information about student details.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
Student_id	Varchar(7)	Not Null	PRI	Null	auto_increment
Student_Name	Varchar(20)	Not Null		Null	
Age	Int	Not Null		Null	
Gender	Varchar(6)	Not Null		Null	
Year_Of_Joining	Int	Not Null		Null	

Degree_Id	Int	Not Null	MUL	Null	
Secret_Question	Varchar(50)	Not Null		Null	
Secret_Answer	Varchar(20)	Not Null		Null	

Table Name: Election_Details

This table stores information about election details.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
Election_id	Int	Not Null	PRI	Null	auto_increment
Election_Type	Varchar(15)	Not Null		Null	
Department_Id	Varchar(4)	Not Null	MUL	Null	
Year_Of_Election	Int	Not Null		Null	

Table Name: Department_Details

This table stores information about department details.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
Department_Id	Varchar(4)	Not Null	PRI	Null	
Department_Name	Varchar(20)	Not Null	Unique	Null	
Total_Students	Int	Not Null		Null	

Table Name: Degree_Details

This table stores information about degree details.

Field	Type	Null	Key	Default	Extra
Degree_Id	Int	Not Null	PRI	Null	Auto_increment
Degree_Type	Varchar(5)	Not Null		Null	
Degree_Name	Varchar(5)	Not Null	MUL	Null	
Branch	Varchar(15)	Not Null	UNI	Null	
Department_Id	Varchar(4)	Not Null	MUL	Null	

Table Name: Candidate_Details

This table stores information about candidate details.

Field	Type	Null	Key	Default	Extra
Candidate_Id	Int	Not Null	PRI	Null	auto_increment
Candidate_Name	Varchar(20)	Not Null		Null	
Gender	Varchar(6)	Not Null		Null	
Symbol	Varchar(12)	Not Null			
Department_Id	Varchar(4)	Not Null	MUL	Null	
Degree_Id	Int	Not Null	MUL	Null	

Votes_Secured	Big Int	Not Null		0	
Student_Id	Varchar(7)	Not Null	MUL	Null	
Year_Of_Election	Int	Not Null		Null	
Election_Type	Varchar(15)	Not Null		Null	

Table Name: Vote_Cast_Details

This table stores information about vote cast details.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
Student_Id	Int	Not Null	PRI	Null	
Election_Id	Int	Not Null	PRI	Null	
Vote_Casted	Varchar(6)	Not Null		No	
Vote_Validity	Varchar(6)	Not Null		True	
Year_Of_Election	Int	Not Null		Null	

Table Name: Symbol

This table stores information about symbol details.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
Symbol	Varchar(12)	Not Null	PRI	Null	

4.7 INPUT AND OUTPUT DESIGN

INPUT DESIGN

Input design is the process of correcting a user-oriented description of the inputs to a computer based one. Inaccurate data is one of the most common causes of data processing errors. If poor input design, particularly where operators enter data from source documents permit wrong data to enter into a computer system, then it will change the entire process in an unpleasant way consists of

- The sequence of field matches the sequence of data or type of data which is going to be entered.
- The data format is well identified for entering or specifying the data

In this project the input design is designed in such a way to accept dtmf and speech input.

PROCESS DESIGN:

The Software process can be identified as a collection of patterns that define a set of activities, actions, work tasks, work products, and related behaviors required to develop computer software. A process provides a method for describe an important characteristic of the software process. By combining patterns, the process that meets the need can be constructed.

The major process in this project is that the user authentication is provided by security questions and vote validity will also be checked to ensure the user to not to vote again for a particular year.

OUTPUT DESIGN

An application is successful only when it can produce efficient and effective reports .A well-formatted report provides a source of reliable information on the website. The outputs are shown to the user as a result of some user input. The output must be shown in a way that a decision can be made easily.

Outputs from a system can be defined as the information being processed and then generated by the system in a specified format. Output design serves the best information source of any system. The results of the college voting can also be accessed with the help of the system.

CHAPTER 5
TESTING

Testing is a critical element of software quality and assurance and represents the ultimate review of specification design and coding. It is a vital activity that has to be enforced in the development of any system. This could be done in parallel during all the phases of system development. The feedback received from these tests can be used for further enhancement of the system under consideration.

The main types of testing carried out on Mobile voting for college are:

- Unit Testing
- Integration Testing
- System Testing

5.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of the software design the module. The local data structure is examined to ensure that data stored temporarily maintains its integrity.

In Mobile Voting for College each page is tested separately as a unit. Initially the flow of control and data through that page is checked. When considering a module as a unit, the flow of data and control through the whole module is tested. The result is stored in the test plan. In a page, each control is further tested in unit testing. The process is done in all the pages of the system. Once the errors are rectified, the testing procedure is repeated with same test cases to ensure this hasn't produced new errors. Hence this is a continuous process.

5.2 Integration Testing

Integration Testing tests the process of integrating the various modules to form the completed system. Integration starts with a set of units each individually tested in isolation and ends when the entire application has been built. Integration testing verifies that the combined units function together correctly.

Mobile Voting for College followed top down integration testing. Modules were linked to the main menu in a sequence as required in the real time operating mode of the system. This process is continued from the page level to module level, finally to the system level. In the final stage, the whole system is taken together and tested for integration. This tests if the change has affected any part of Mobile Voting for College negatively after the change was made. The whole set test cases need to be run again to do the regression testing.

5.3 System Testing

System testing is actually a serious of different tests, whose primary purpose is to fully exercise the computer-based system. This helps in verifying that all the system elements have been properly integrated and perform the allocated functions. It verifies the entire product after having integrated all software and hardware components, and validates it according to the original project requirement.

5.3.1 Security Testing

Security testing is important in system testing. Testing is done to ensure that a user with respective rights will only be allowed to vote in Mobile Voting for College. If other users try to cast vote, the system will not allow the user to cast his vote. In addition user has to answer a security question that he provided during his enrollment.

Another security issues involves the sensitive data in the system. The system is highly secure with authentication fixed at various levels of the hierarchy.

5.3.2 Stress Testing

Stress Testing executes a system in a manner that demands resources in abnormal quantity, frequency or volume. Mobile Voting For College was stress tested by all internal users of authorized service vendors in the project and accessing simultaneously to various modules in the system. All the reports and the screens are tested for their validity and values in the database tables are checked for their correctness and consistency. After successful testing of the system, it was ready for implementation.

5.4 MAINTENANCE

The process of making changes and modifications to the system after it has been delivered implemented and is in use called software maintenance.

Corrective Maintenance

It is concerned with fixing reported errors in software. They are coding errors and design errors.

Adaptive Maintenance

It is concerned with changing the software to source and to adapt to the new and changing environment.

Defective Maintenance

It involves implementing new functional or non-functional system requirements to ensure more effective execution of the system.

Perceptive Maintenance

It mainly deals with accommodating new or changed users requirements. It also includes activities to increase the system performance or to enhance its user interface. The objective of perceptive maintenance should be to prevent failures and optimize the software.

Preventive Maintenance

It concerns activities aimed at increasing the system's maintainability such as updating documentation adding comments, improving modular structure of the system.

CHAPTER 6

SYSTEM IMPLEMENTATION

This chapter describes the implementation process of the application.

6.1 IMPLEMENTATION PROCESS

System Implementation is the part of the software engineering life cycle, where, the design artifacts are converted to a working application. Coding is done in this stage using an framework and programming language, which would solve the specific problem the best way. Once the design is coded into a working application, it has to be verified, validated and tested in detail. The tested product if successful is deployed in the user environment.

6.2 SYSTEM VERIFICATION

In Mobile Voting For College, each module has been verified whether it is working as it is desired. The functionality of the module has been exhibited and verified as it is working properly. Verification also determines whether the system is consistent, adheres to standards, uses reliable techniques and prudent practices, and performs the selected functions in the correct manner. In data access, it verifies whether the right data is being accessed, in terms of the right place and in the right way. For instance the right degree of toughness level question has to be retrieved and displayed during the exam.

6.3 SYSTEM VALIDATION

In this project, validation checks whether the developer is moving towards the right product, whether the development is moving towards the actual intended product that was agreed upon in the beginning. Validation also determines if Mobile Voting for College complies with the requirements and performs functions for which it is intended and meets the goal and user needs. It is traditional and is performed at the end of the project.

Completeness check ensures that all fields in a record are present and are read in the proper sequence. In this project, while entering the student details, all the fields has to be entered, else the system will ask to enter the detail in the unfilled field. This has been done during completeness check.

Sequence check verifies that data records are in sequence prior to processing. A check of duplicate records may also be incorporated in the routine. The same student cannot have a different student id.

CHAPTER-7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

This system allows the user's to cast their vote with high level of confidentiality and keep voters away from the possible threats like booth capturing and it may increase the voter turnouts. People, who are uncomfortable to wait in queues, like physically challenged persons, can benefit from the system. Since the system operates on IVR platform the voter session will be interactive and user friendly which makes the user to feel in a very comfortable zone. The system provides security features by asking security questions. If the answer matches, only then the particular student will be allowed to cast his vote. The system also uses dual tone multi frequency and speech recognition for user inputs. The user inputs will be processed to accept the user vote.

7.2 FUTURE ENHANCEMENTS

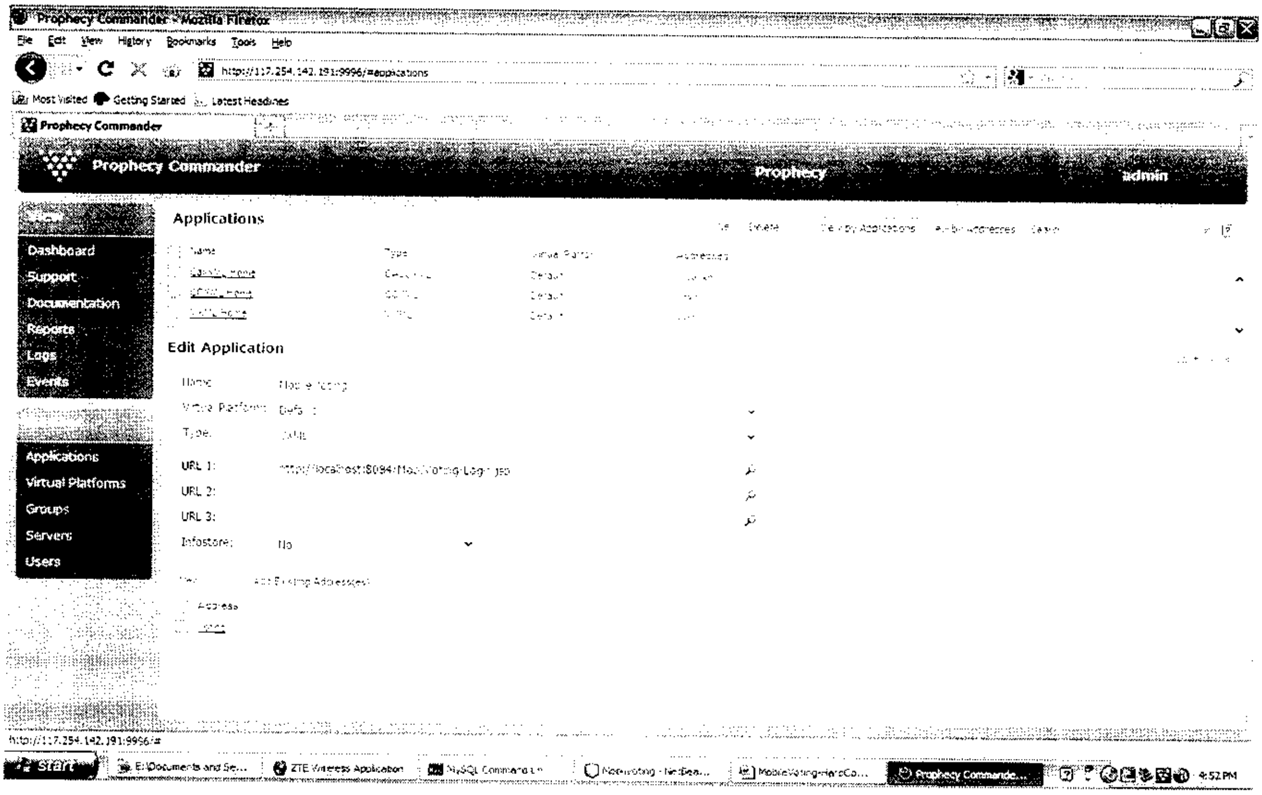
As a measure of security features mobile devices allows the user to make use of biometric information's which are unique to individuals. For example voice biometrics can be used with VOICEXML to authenticate a user. Voice biometrics stores the user's voice by taking samples and verifies it with the database when he tries to login again. Voice biometrics uses the user's unique voice characteristics by eliminating noise to authenticate the user. Similarly finger prints can also be used with mobiles that provide those options. Scalability is another important feature to be addressed.

CHAPTER-8

APPENDICES

APPENDIX-A: SCREENSHOTS

A-1 Deploying VOICEXML Applications:



Name: Mobile Voting

Virtual Platform: Default

Type: VXML

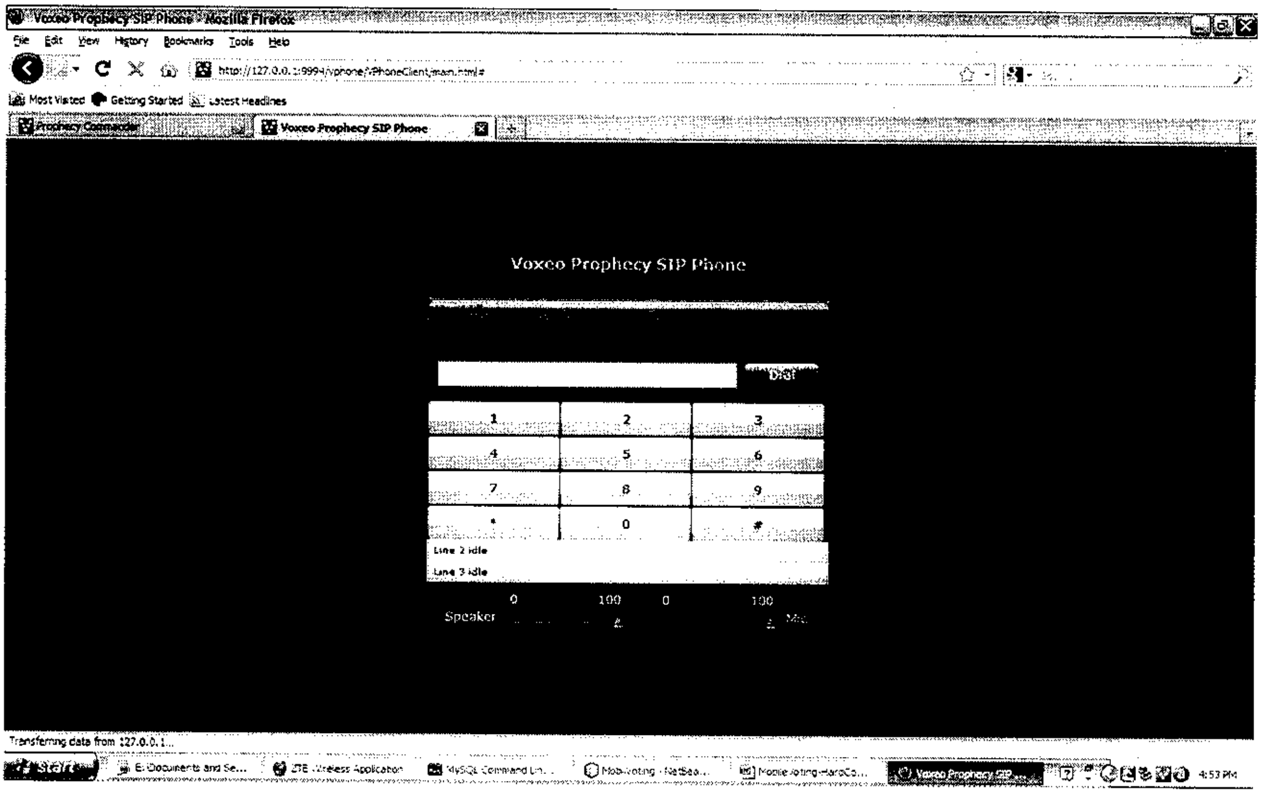
URL 1: http://localhost:8094/MobVoting/Login.jsp

URL 2:

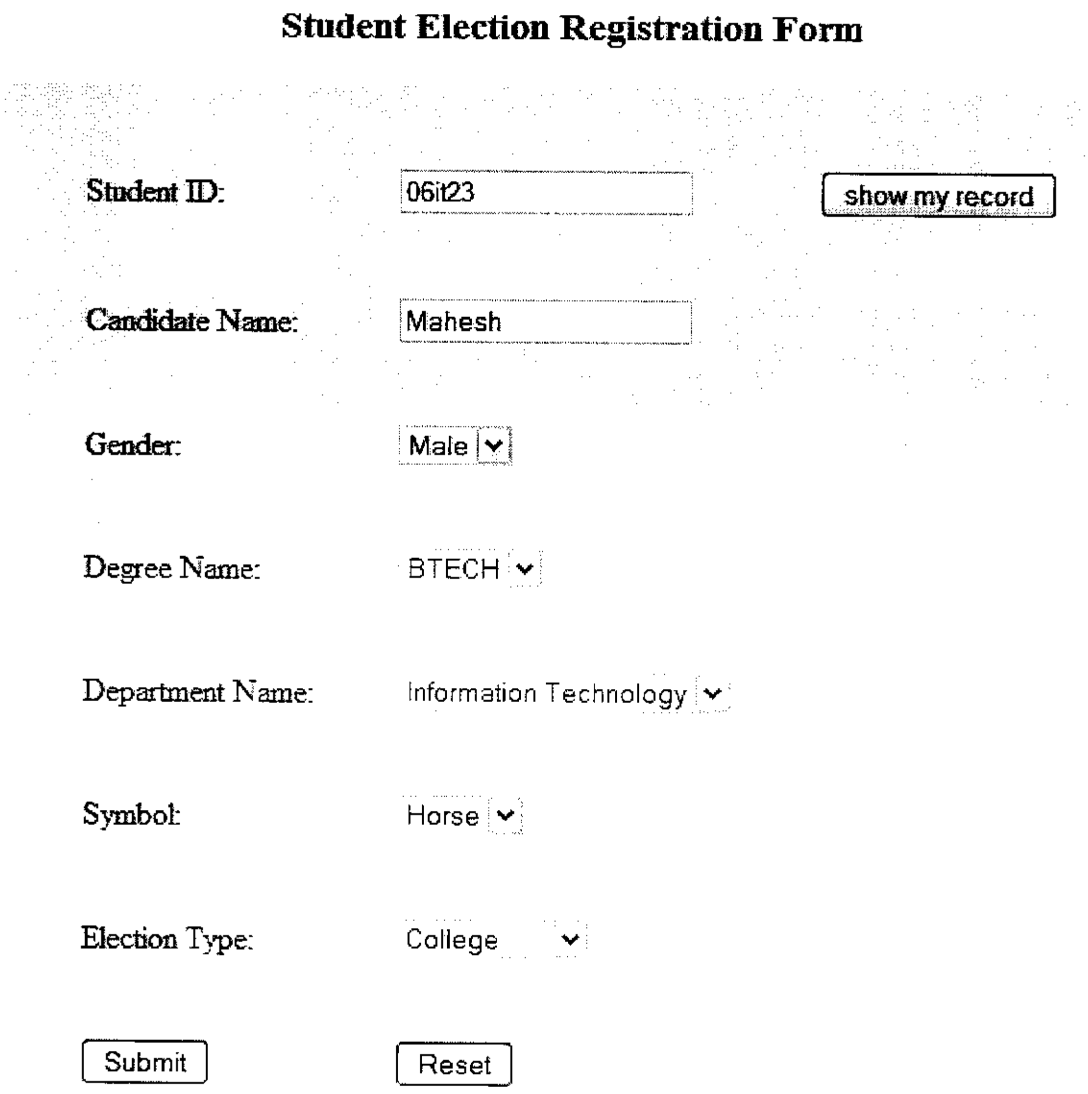
URL 3:

Infostore: No

A-2 Testing VOICEXML Application Using SIP Phone:

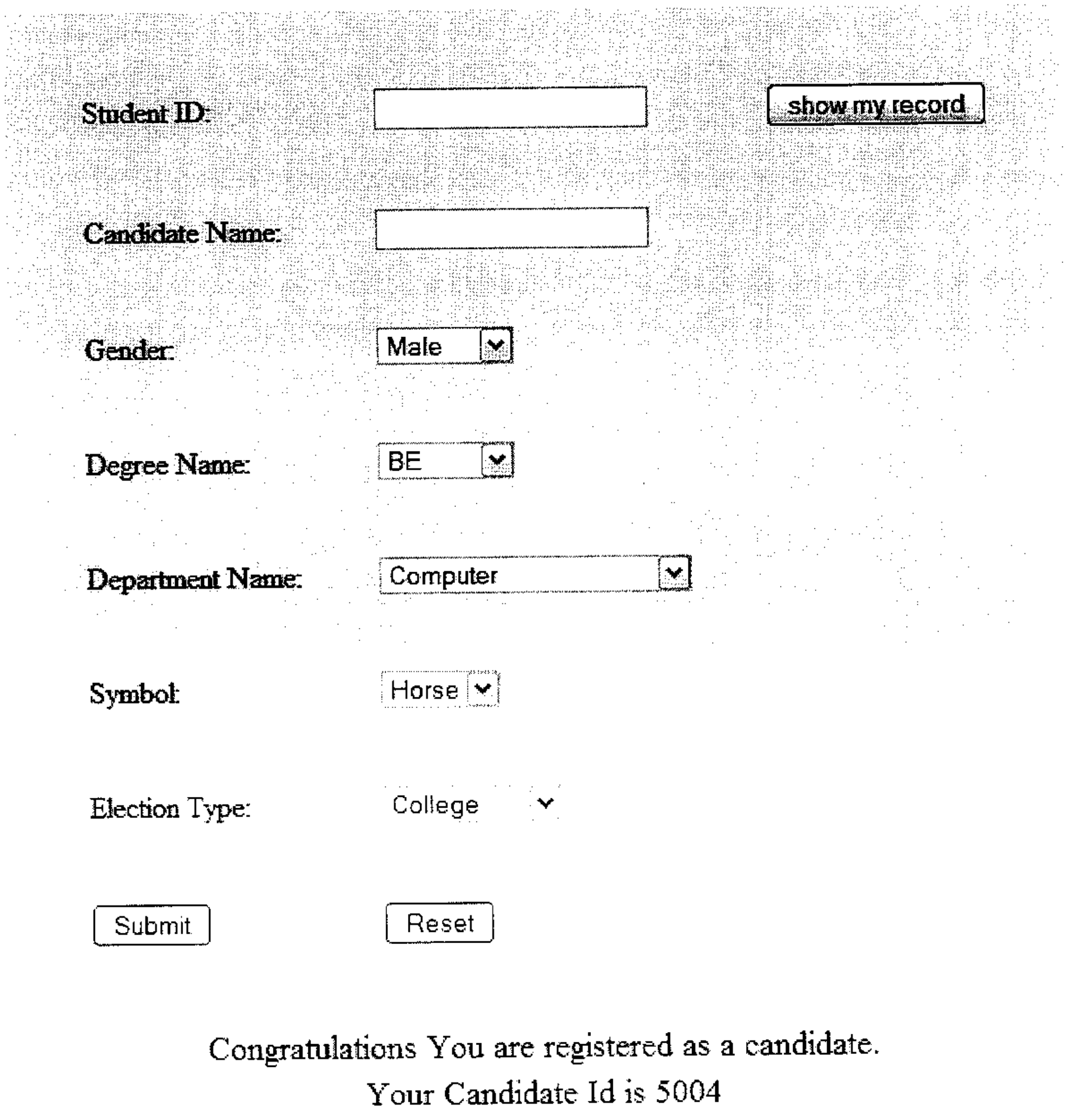


A-3 CANDIDATE REGISTRATION FORM



Student Election Registration Form

Student ID:	06it23
Candidate Name:	Mahesh
Gender:	Male
Degree Name:	BTECH
Department Name:	Information Technology
Symbol:	Horse
Election Type:	College



Student ID:

Candidate Name:

Gender: Male

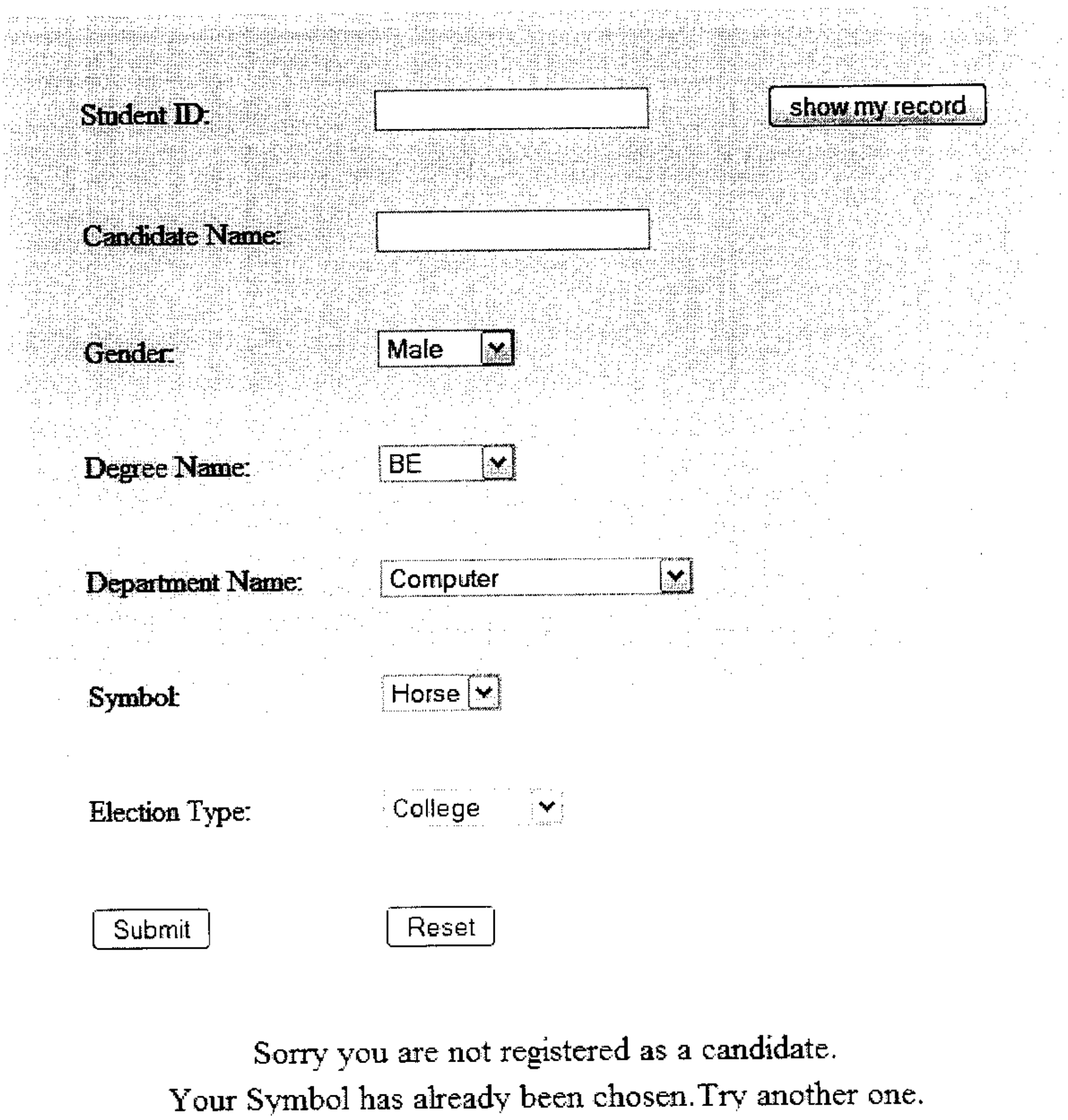
Degree Name: BE

Department Name: Computer

Symbol: Horse

Election Type: College

Congratulations You are registered as a candidate.
Your Candidate Id is 5004



Student ID:	
Candidate Name:	
Gender:	Male
Degree Name:	BE
Department Name:	Computer
Symbol:	Horse
Election Type:	College

Sorry you are not registered as a candidate.
Your Symbol has already been chosen.Try another one.

APPENDIX-B: SAMPLE CODING

A-1 Votecast.jsp

```
<%--
    Document   : votecast
    Created on : Mar 15, 2010, 4:23:54 PM
    Author     : Administrator
--%>

<?xml version="1.0" encoding="UTF-8" ?>
<vxml version="2.1">
    <%@ page import="java.sql.*,java.util.*"%>
    <form>
        <block>
            <prompt>
                Your vote is being verified.Wait for some time.
                <%
                            try {
                                String vote_cast = request.getParameter("vcast");
                                String dept_name = null, vcast = null, elect_type = null;
                                ArrayList<Integer> elect_id1 = new ArrayList<Integer>();
                                ArrayList<String> electtype = new ArrayList<String>();
                                Set<String> c = new HashSet<String>();
                                String dept_id = null;
                                int deg_id = 0, year = 0, elect_id = 0;
                                Class.forName("com.mysql.jdbc.Driver");
                                Connection                         connection                         =
DriverManager.getConnection("jdbc:mysql://localhost:3306/collegevoting", "root", "root");
                                Statement st = connection.createStatement();
```

```
StringBuffer s = null;
int j = 0;
String sql15="select election_type from election_details";
ResultSet rs15=st.executeQuery(sql15);
while(rs15.next()==true){
    c.add(rs15.getString(1));
  }
if (vote_cast.contains(" ")) {
  String cand_id = "";
  String[] tokens = vote_cast.split("\\s");
  for (String s1 : tokens) {
    cand_id = s1.concat(cand_id);
  }
  s = new StringBuffer(cand_id);
  s = s.reverse();
  StringBuffer Stud_Id = (StringBuffer) session.getAttribute("Stud_Id");
  String sql = "select degree_id from student_details where student_id='" +
Stud_Id + "'";
  ResultSet rs = st.executeQuery(sql);
  while (rs.next() == true) {
    deg_id = rs.getInt(1);
  }
  String sql1 = "select department_id from degree_details where
degree_id=" + deg_id + "";
  ResultSet rs1 = st.executeQuery(sql1);
  while (rs1.next() == true) {
    dept_id = rs1.getString(1);
  }
  String sql2 = "select department_name from department_details where
department_id='" + dept_id + "'";
```

```
                ResultSet rs2 = st.executeQuery(sql2);
                while (rs2.next() == true) {
                    dept_name = rs2.getString(1);
                }
                Calendar calendar = Calendar.getInstance();
                year = calendar.get(calendar.YEAR);
                String sql4 = "select Election_Id from Election_Details where
Department_Id='" + dept_id + "' "
                        + "and year_of_election=" + year + "";
                ResultSet rs4 = st.executeQuery(sql4);
                while (rs4.next() == true) {
                    elect_id = rs4.getInt(1);
                }
                String sql11 = "select * from vote_cast_details where student_id='" +
Stud_Id + "' and year_of_election=" + year + "";
                ResultSet rs11 = st.executeQuery(sql11);
                if (rs11.next() == true) {
                    String sql10 = "select distinct vote_casted from vote_cast_details where
student_id='" + Stud_Id + "' and year_of_election=" + year + "";
                    ResultSet rs10 = st.executeQuery(sql10);
                    while (rs10.next() == true) {
                        vcast = rs10.getString(1);
                    }
                    String Depart_Name = (String) session.getAttribute("Depart_Name");
                    if (Depart_Name.equalsIgnoreCase(dept_name)) {
                        if (vcast.equalsIgnoreCase("no")) {
                            String sql3 = "select votes_secured,election_type from
candidate_details where candidate_id=" + s + "";
                            ResultSet rs3 = st.executeQuery(sql3);
                            while (rs3.next() == true) {
                                j = rs3.getInt(1);
                                elect_type = rs3.getString(2);
                              }
                              String sql6 = "insert into
vote_cast_details(student_id,election_id,year_of_election) values('" + Stud_Id + "'," + elect_id +
"," + year + ")";
                              int i = st.executeUpdate(sql6);
                              if (elect_type.equalsIgnoreCase("Department") && i == 1) {
                                String sql5 = "update candidate_details set votes_secured=" +
(++j) + " where candidate_id=" + s + "";
                                int i1 = st.executeUpdate(sql5);
                                if (i1 == 1) {
              %>
              Your vote has been registered successfully.
              <%
                                        String sql7 = "select election_id from
vote_cast_details where student_id='" + Stud_Id + "' and year_of_election=" + year + "";
                                        ResultSet rs7 = st.executeQuery(sql7);
                                        while (rs7.next() == true) {
                                          elect_id1.add(rs7.getInt(1));
                                        }
                                        for (int k = 0; k < elect_id1.size(); k++) {
                                          String sql8 = "select election_type from
election_details where election_id=" + elect_id1.get(k) + "";
                                          ResultSet rs8 = st.executeQuery(sql8);
                                          while (rs8.next() == true) {
                                            electtype.add(rs8.getString(1));
                                          }
                                          if
((electtype.contains("Department")||electtype.contains("department"))&&(electtype.contains("Co
llege")||electtype.contains("college"))) {
                                        String sql9 = "update vote_cast_details set
vote_casted='yes' where student_id='" + Stud_Id + "'";
                                        st.executeUpdate(sql9);
                                      }
                                    }
                                  }
                                }
                              } else {
        %>
        You have already casted your vote for both elections.Try next time.
        <%                              }
                            } else {
        %>
        Sorry.you do not belong to this department.
        <%                        }
                          } else {
                            String          Depart_Name          =          (String)
session.getAttribute("Depart_Name");
                            if (Depart_Name.equalsIgnoreCase(dept_name)) {
                              String sql3 = "select votes_secured from candidate_details
where candidate_id=" + s + "";
                              ResultSet rs3 = st.executeQuery(sql3);
                              while (rs3.next() == true) {
                                j = rs3.getInt(1);
                              }
                              String sql5 = "update candidate_details set votes_secured="
+ (++j) + " where candidate_id=" + s + "";
                              int i = st.executeUpdate(sql5);
                              if (i == 1) {
```

```
%>
your vote has been registered successfully.
<%
                                    String sql6 = "insert into
vote_cast_details(student_id,election_id,year_of_election) values('" + Stud_Id + "'," + elect_id +
"," + year + ")";
                                    st.executeUpdate(sql6);
                                    String sql7 = "select election_id from
vote_cast_details where student_id='" + Stud_Id + "' and year_of_election=" + year + "";
                                    ResultSet rs7 = st.executeQuery(sql7);
                                    while (rs7.next() == true) {
                                      elect_id1.add(rs7.getInt(1));
                                    }
                                    for (int k = 0; k < elect_id1.size(); k++) {
                                      String sql8 = "select election_type from
election_details where election_id=" + elect_id1.get(k) + "";
                                      ResultSet rs8 = st.executeQuery(sql8);
                                      while (rs8.next() == true) {
                                        electtype.add(rs8.getString(1));
                                      }
                                      if
((electtype.contains("Department")||electtype.contains("department"))&&(electtype.contains("Co
llege")||electtype.contains("college"))) {
                                        String sql9 = "update vote_cast_details set
vote_casted='yes' where student_id='" + Stud_Id + "'";
                                        st.executeUpdate(sql9);
                                      }
                                    }
                                  }
                                } else {
```

```
            %>
            Sorry.you do not belong to this department.
            <%                                  }
                                        }
                                     }
                                  } catch (SQLException se) {
                                     String s;
                                     s = se.toString();
                                     if (s.contains("key 1")) {%>
            You have already casted your vote for department elections.Try next time.
            <%                                    }
                     }
            %>
         </prompt>
      </block>
   </form>
</vxml>
```

CHAPTER-9
REFERENCES

TEXT-BOOK REFERENCES

- VoiceXML: 10 Projects to Voice Enable Your Web Site- Mark Miller
- "Software Engineering, A practitioner's approach", by Roger S. Pressman, Tata McGraw-Hill Edition.
- Head First Servlets and JSP- Kathy Sierra, Bert Bates.
- Fundamentals of Speech Recognition - L.Rabiner.

WEB REFERENCES

- Creating VoiceXML pages within a Java Web developer framework- http://www.ibm.com/developerworks/web/library/wa-voicexml/
- VoiceXML Tutorials- http://www.vxml.org/
- JSP and Servlets Tutorial- http://java.sun.com/javaee/reference/tutorials/