MEDISCRIPT- MOBILE CLOUD COLLABRATIVE SPEECH RECOGNITION FRAMEWORK

T.Senthil Kumar
Assistant Professor,
Computer Science and Engineering
Amrita School of Engineering
Amrita Vishwa Vidyapeetham
Coimbatore

Vishnu Gajendran
Amrita School of Engineering
Amrita Vishwa Vidyapeetham,Coimbatore

Rajamanuri Harshad
Amrita School of Engineering
Amrita Vishwa Vidyapeetham,Coimbatore

Sneha Aswani
Deepti Sankara Narayanan
Amrita School of Engineering
Amrita Vishwa Vidyapeetham,
Coimbat

ABSTRACT
Speech recognition is a vital part in medical transciription. The existing speech recognition systems, that run as standalone desktop applications, fall short in many cases due to low accuracy rates and high processing time. The bottleneck in these systems, is the lack of computation power (in terms of processing power and memory) made accessible to them. This paper proposes a mobile-cloud collaborative approach for the automation of speech to text conversion. The model proposed leverages the power of cloud computing and the ubiquitous nature of mobile computing. Computing resthisces can be scaled up/down in the cloud (Elastic Computing) depending on the usage of the system. This kind of speech recognition framework has many real time applications such as IVR systems, Medical Transcription systems, Railway Enquiries, Jthisnalism, Interactive User Interfaces, etc. A generic framework is advantageous, because the speech models in the Automatic Speech Recognizer (ASR) could be trained according to the specific domain required, allowing wide usability. The proposed speech framework is used for medical transcription process. Medical transcription process involves a medical transcriptionist who listens to the recorded speech of a doctor and manually types a transcript file. This process is automated by using the proposed speech framework. With this system, the work of the medical transcriptionist is reduced to error checking in the auto generated transcript file. The entire model is developed for a mobile cloud environment considering the characteristics of cloud delivery models.

Keywords
Framework, cloud, buckets, model, speech

1. INTRODUCTION
Cloud computing refers to the provision of computational resthisces on demand via computer network such as applications, databases, file services, email, etc. In the traditional model of computing, both data and software are fully contained on the user's computer; in cloud computing, the user's computer may containal most no software or data (perhaps a minimal operating system and web browseronly), serving as little more than a display terminal for processes occurring on a network of computers far away[15]. A common shorthand for a provided cloud computing service (or even an aggregation of all existing cloud services) is "The Cloud".

The principle behind the cloud is that any computer connected to the internet is connected to the same pool of computing power, applications, and files. Users can store and access personal files or use productivity applications on a remote sever rather than physically carrying around a storage medium such as a DVD or thumb drive. Almost all users of the internet may be using a form of cloud computing though few realize it. Figure 1 shows the prototype model for a cloud.

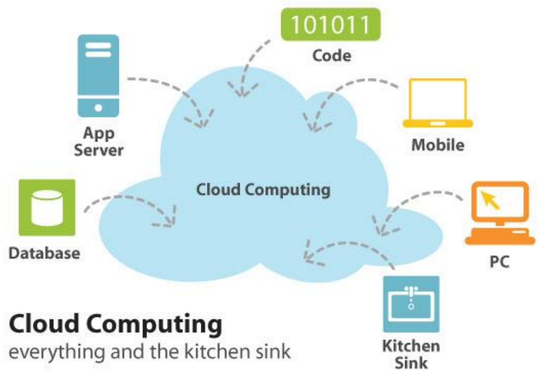

Figure 1. Prototype Model for a cloud

1.1 CLOUD ARCHITECTURES
Cloud Architectures address key difficulties surrounding large scale data processing. In traditional data processing, it is difficult to get as many machines as an application needs. Second, it is difficult to get the machines when one needs them. Third, it is difficult to distribute and coordinate a large scale job on different machines, run processes on them, and provision another machine to recover if one machine fails[1]. Fthisth, it is difficult to auto scale up

and down based on dynamic workloads. Cloud Architectures solve such difficulties. Applications built on Cloud Architectures run in the cloud where the physical location of the infrastructure is determined by the provider[12]. They take advantage of simple APIs of Internet accessible services that scale on demand, that are industrial strength, where the complex reliability and scalability logic of the underlying services remains implemented and hidden inside the cloud. The usage of resthisces in Cloud Architectures is as needed, sometimes ephemeral or seasonal, thereby providing the highest utilization and optimum service for the price paid[13]. Cloud Computing models could be private/public/hybrid cloud. The services could be SoftwareAsaService, PlatformasAsaService, InfrastructureAsaService [15].

1.2 SPEECH RECOGNITION SYSTEM

This systems are used for automatic speech to text conversion. Automatic speech recognition systems today find widespread application in tasks that require a human-machine interface, such as automatic call processing in the telephone network and query-based information systems that do things like providing updated travel information, stock price quotations, weather reports, etc. For example, in this paper, we are using ASR system for Automatic Medical Transcription[16].

This process is very computationally intensive process. It requires more memory and processing power for its execution. The accuracy of the recognition process depends on different factors. It is not practically possible to achieve 100 percent accuracy in the system. But with proper training of the system, the accuracy rate can be improved.

The speech recognition could be any of the following:

- Interactive Voice Response
- Automatic Speech Recognition
- Advanced Speech Automatic Speech recognition
- Near Natural Language Automatic Speech recognition

2. RELATED WORK

2.1 SPEECH RECOGNITION

Speech Recognition

Presently, Automatic Speech Recognition (ASR) is gaining importance in various industries, because it reduces the manual work done by a person. For example, in an Intelligent Voice Response system, ASR system can be used to automatically provide response to the user according to his/her request. Some of the existing ASR systems are mentioned below.

a. Windows Speech Recognition

Windows Speech Recognition allows the user to control the computer by giving specific voice commands. Programs needing mouse clicks in arbitrary locations can also be controlled through speech; when asked to do so, a "mousegrid" of nine zones is displayed, with numbers inside each. The user speaks the number, and another grid of nine zones is placed inside the chosen zone. This continues until the interface element to be clicked is within the chosen zone. Windows Speech Recognition relies on Microsoft Speech API (SAPI) for functionality..

b. Dragon Dictation

Dragon Dictation is a speech recognition application for Apple's iOS platforms including iPhone, iPod touch and iPad. The App provides automatic speech-totext capabilities. It was developed by Nuance Communications. Dragon Dictation speech recognition is based on Dragon NaturallySpeaking speech recognition technology from Nuance Communications.

c. RWTH ASR

RWTH ASR is an open source speech recognition toolkit. The toolkit includes state of the art speech recognition technology for the development of automatic speech recognition systems. It has been developed by the Human Language Technology and Pattern Recognition Group at RWTH Aachen University. RWTH ASR includes tools for the development of acoustic models and decoders as well as components for speaker adaptation, speaker adaptive training, and unsupervised training. The software runs on Linux.

d. Julius

Julius is an open source speech recognition engine. Julius is a high-performance, two-pass large vocabulary continuous speech recognition (LVCSR) decoder software for speech-related researchers and developers. The main platform is Linux and other Unix workstations, and also works on Windows.

e. CMUSphinx

CMU Sphinx, also called Sphinx, is the speech recognition system developed at Carnegie Mellon University. Sphinx is a continuous-speech, speaker-independent recognition system making use of hidden Markov acoustic models (HMMs) and an n-gram statistical language model. Sphinx featured feasibility of continuous speech, speaker-independent large-vocabulary recognition. Elasticity can be built as part of the cloud service[4].

All these softwares run in Desktop system environment. The problem with all these existing software is that they are restricted by resource constraints. Resources here are the computational power and the memory. If the system needs to be accurate, then more training is needed. When training of acoustic and language models are done for many users, then the computational cost for the recognition process increases.

But, with limited resource available in the normal Desktop environment, this is not very effective and the execution time of the recognition process increases drastically. Furthermore, these are standalone systems and manual accessibility from geographically distant locations is hindered.

The Table 1 shown below shows the comparison of different cloud services[2].

	Amazon AWS	Google App Engine	Windows Azure	Force.com	Rackspace	GoGrid
Cloud Services	*Paas* *Iaas*	*Paas*	*Paas* *Iaas*	*Paas*	*Iaas*	*Iaas*
			Features			
Platforms supported	**Operating systems** • Red Hat Enterprise Linux • Windows Server 2003/2008 • Oracle Enterprise Linux • OpenSolaris • OpenSUSE Linux • Ubuntu Linux • Fedora Gentoo Linux • Debian • Software • IBM DB2 • IBM Informix Dynamic Server • Microsoft SQL Server Standard 2005 • MySQL Enterprise • Oracle Database 11g • Hadoop	**Runtime** • Java Runtime Environment • Python Runtime Environment **Features** • Integration with Google Accounts • URL Fetch • Mail • Memcache • Image Manipulation • Scheduled Tasks and Task Queues • XMPP • Blobstore (which supports objects upto 50MB in size) **Software** External software like AppServers Databases cannot be installed	**Operating systems** • Windows 7 • Windows Server 2008 • Windows Vista	**Software** • Unlimited real-time database customizations • Programmable user interface • Programmable cloud logic • Real-time workflow and approvals • Real-time web sites • Real-time mobile deployment • Integrated content library • Real-time analytics • Granular security and sharing	**Operating systems** • Linux • Mac OS X • Windows	**Operating systems** • Windows server 2008 • Windows server 2003 • CentOS 5.1 • CentOS 5.3 • Redhat Linux 5.1 • Redhat Linux 5.4

Table 1 . Comparison of different cloud services

The comparison is prepared considering the mobile support provided by the cloud platforms. Java and corba could also be integrated to handle medical transcripts[3].

3. PROPOSED MODEL

This proposed framework has three major components. They are:

- **Front end**
- **Cloud Framework**
- **Automatic Speech Recognition system**

The complete model of the proposed approach is shown below:

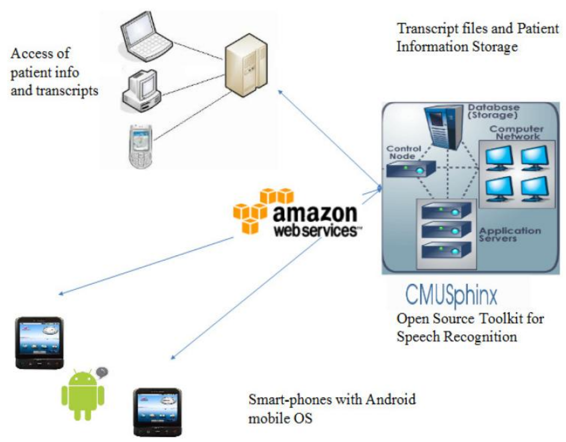

Figure.2 Medscript Application Framework

The architectural Diagram of the MedScript Application is shown below in Figure .3.

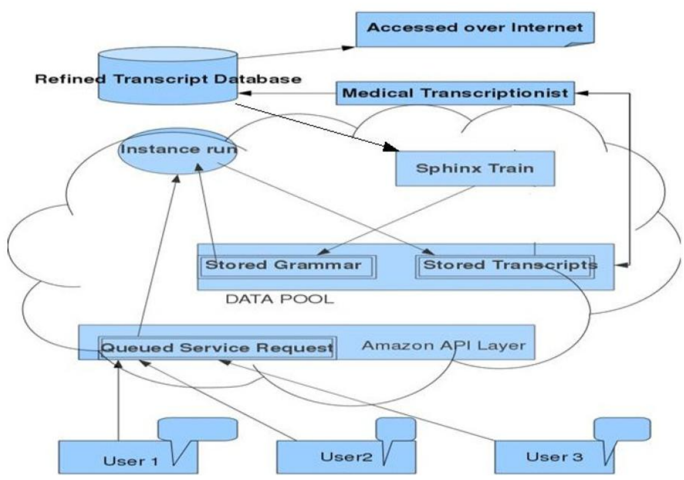

Figure 3.Architectural Diagram

a. Front end

This is the part of the system which is used by the client (Doctor) to record his/her speech. A mobile phone for this purpose in this system. As the name says, the main advantage of using a mobile phone is mobility. It can be used anywhere in the world. The model uses Android as the mobile platform. The front end component of this application is developed using Android-AWS software development kit. With this development platform, the mobile-cloud connectivity is made easier[10].

The functionality is as follows:

The Doctor needs to log into the system to use the functionalities of this product. Every authorized doctor will be provided with a username and password by the system. All details of the doctor will be stored on the cloud. After the successful login operation, the doctor can either choose to record or to search for the transcripts. The record feature is responsible for providing the doctor with the interface to record his/her voice. The doctor speaks at a standard speed. The recorded speech file (.wav) is temporarily saved in the mobile phone and later sent to the cloud. After the doctor affirms to save the file on the cloud, the speech file is saved in the database along with the details of the doctor and the patient. This is followed by the queuing of the speech files on the cloud to be processed by the ASR. The converted transcript is also stored in the database along with all the doctor's details. This transcript can be referred by the doctor from any device. The search option gives the privilege to the doctor to search for the transcript based on the date when the speech file was recorded or by using the patient ID provided.

b.Cloud framework

The entire back-end processing of this application takes place in the cloud. The backend processing includes receiving the audio file from the doctor, generating a transcript file from the audio file and storing the transcript file along with the patient and doctor's information in the database. The model uses Amazon Web Service (AWS) as the cloud for this application[5]. The Amazon Web Services provides a host of services like EC2 (Elastic cloud compute), S3 (Simple Storage Service) and SimpleDB. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as the computing requirements change. By using cloud, this application is given access to enormous resources which can be utilized for its computation.

Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. Amazon SimpleDB is a highly available, scalable, and flexible non-relational data store that offloads the work of database administration.

The functionality is as follows:

The speech recognition software is deployed as an image, which is loaded as an instance and executed on request. The audio files are queued up for processing and are stored in the Amazon Simple Storage Service (Amazon S3). The transfer takes place over HTTP. Each audio file is stored in a bucket and retrieved via a unique, developer-assigned key. Now, each audio file is given to the Speech Recognition System as input which in turn gives the transcript file as output[24]. The grammar and other language models which are used by the Speech Recognition System are stored in Simple Storage Service. The final transcript files are also stored here.

A database is created using SimpleDB ,which stores information related to transcript. The data is automatically indexed, making it easy to find the information needed. The transcription created is not 100 percent accurate. So, a medical transcriptionist manually checks for corrections in the raw transcript file to create a refined transcript file. The refined transcript also replaces the raw transcript in the cloud storage which could be accessible to users over the Internet. SphinxTrain, which is the training module of the Sphinx Speech Recognition system, is used to train the acoustic model (Grammar) in order to improve the accuracy of recognition. This module runs as a separate instance which trains the ASR system for different users.

c. Automatic Speech Recognition System (ASR)

This component does the Speech to Text conversion in this framework. It receives the audio file (Speech file) of the doctor as the input and returns the transcript file as the output[19]. The model uses Sphinx-4 Speech Recognition System developed by Carnegie Mellon University. The Speech Recognition process is computationally intensive and needs more processing power. Hence this system can make good utilization of the resources available in the cloud[17].

The functionality is as follows:

The Speech Recognition System receives an audio file (.wav) as the input. The system cannot directly work with the audio file[20]. The signals are first transformed into a sequence of feature vectors, which are used in place of the actual acoustic signals. The feature vectors are Mel-frequency cepstral coefficients (MFCCs). This feature vector file is used by the system for recognition. There are three important components inside Speech Recognition System. They are Acoustic model, Dictionary, Language model.

An Acoustic model contains acoustic properties for each phoneme. The Dictionary contains all the words in the language (it should at least contain all the words that are spoken by the doctor) with their respective phonemes[22]. It contains the mapping from words to phones. The Language model defines , which word could follow previously recognized word and helps to significantly restrict the matching process by stripping words that are not probable. These components are used together in the system to recognize speech[17].

By using these components, the matching process happens which gives the most probable text for the speech. This text is finally stored in a text file , which is the final output of the Speech Recognition System[23]. As said earlier, this text file (transcript) is later refined and used by the SphinxTrain module for training ASR[11].

4.IMPLEMENTATION DETAILS

The different parts of the implementation screenshots is presented here. In functionality the implementation starts with a welcome screen for users.

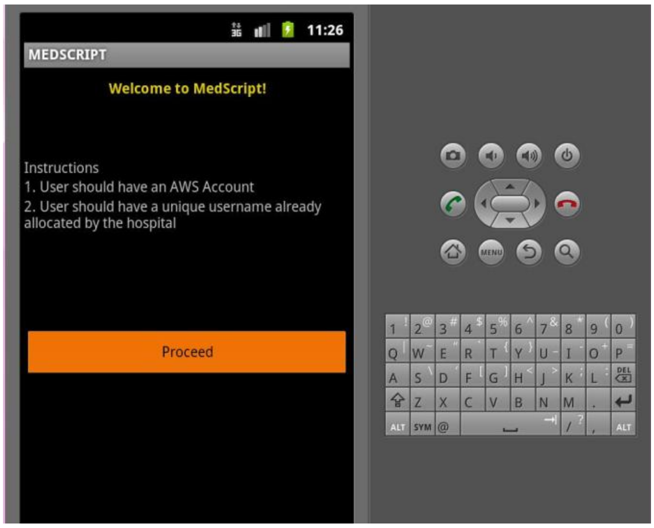

Figure 4. Welcome Screen

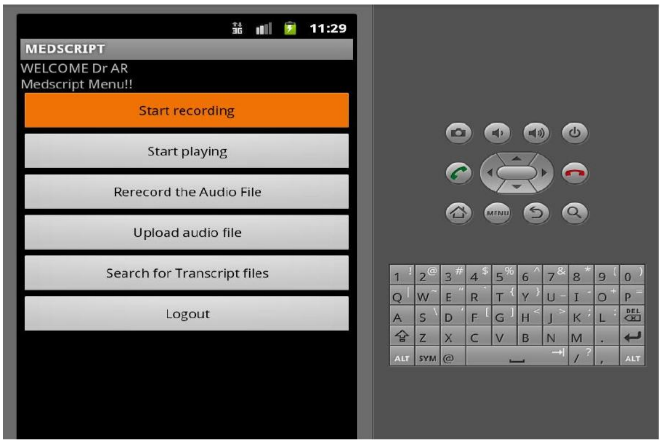

Figure 5. Medscript menu

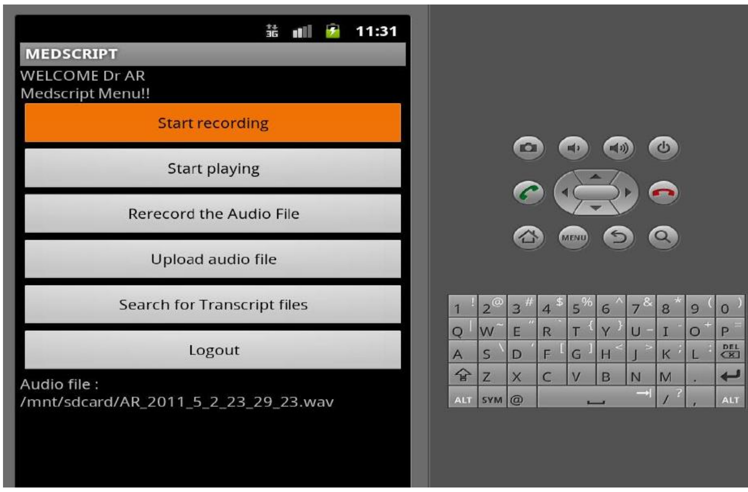

Figure 6.Recording an audio file

Figure 7.Select file to upload

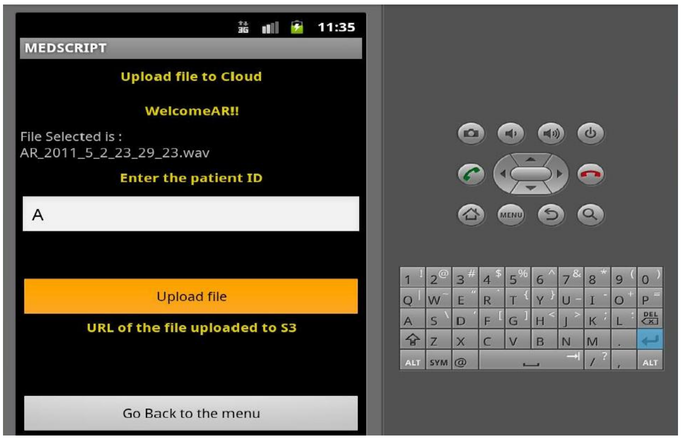

Figure 8. Uploading a file

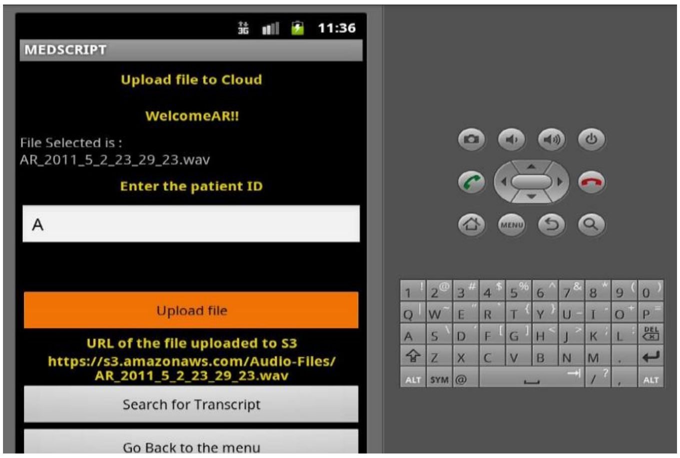

Figure 9. URL of file uploaded

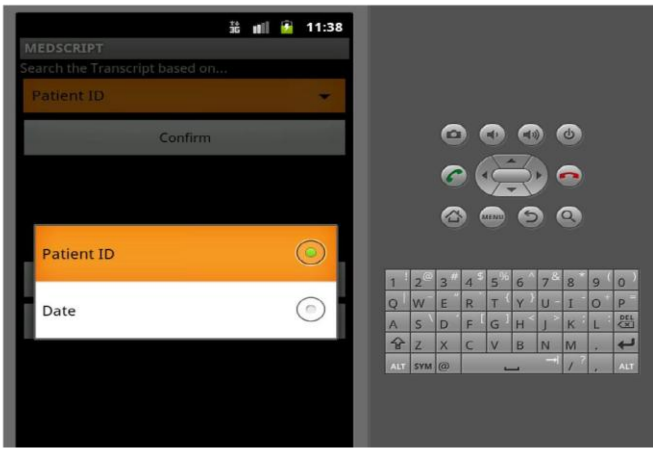

Figure 10.Search for Transcript

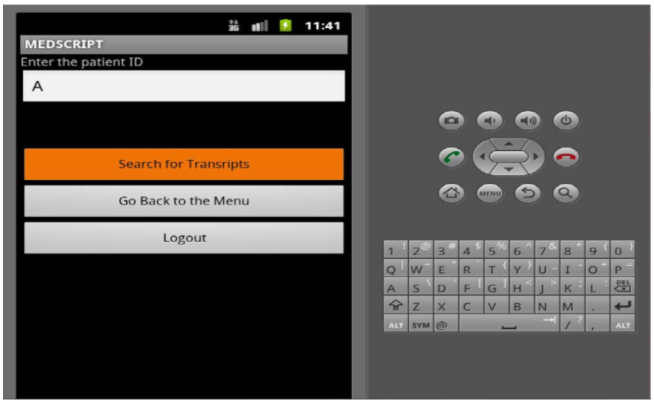

Figure 11.Search based on patient id

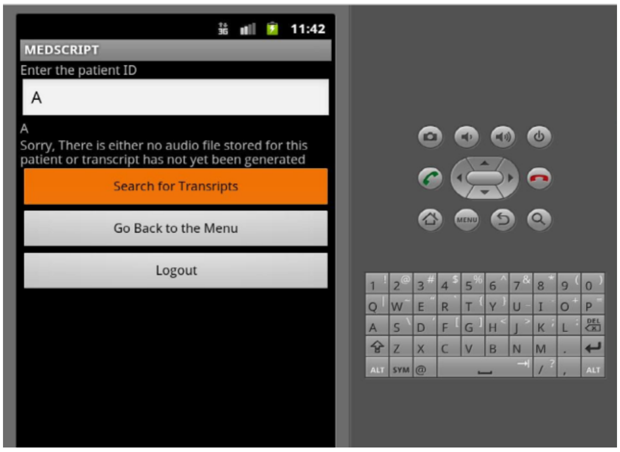

Figure12. Unavailable file error message

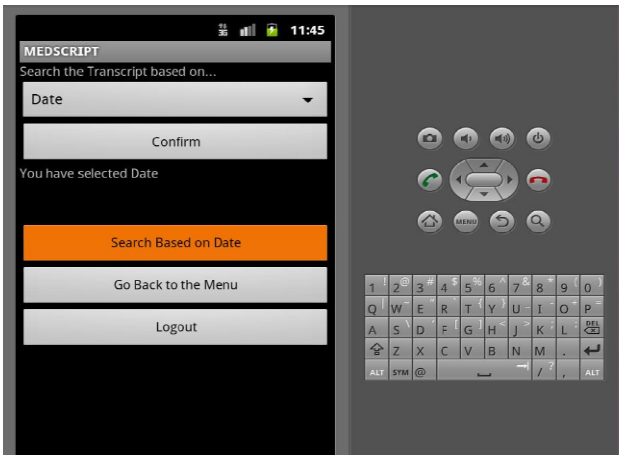

Figure 13. Search based on date

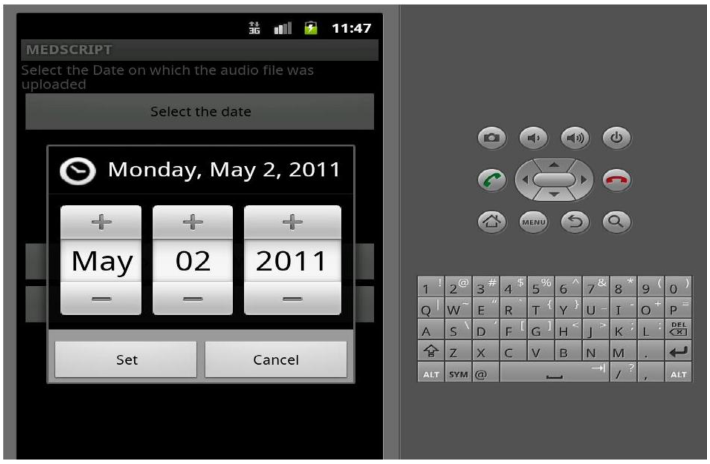

Figure 14.Select date for transcript

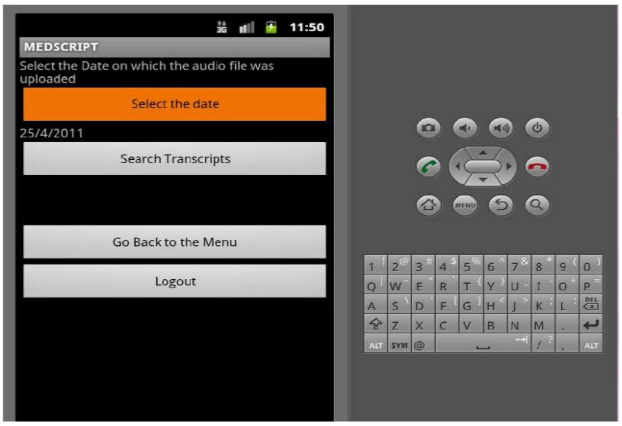

Figure15.Search based on date

The following are the screenshots of the support provided to the doctors.

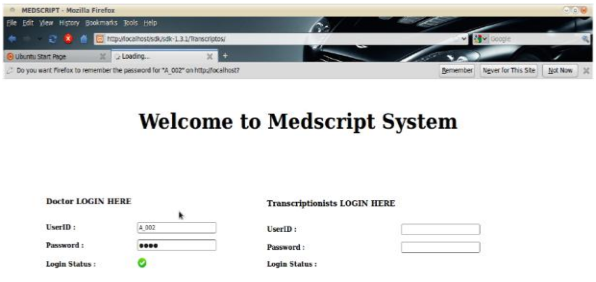

Figure 16.Login page for transcriptionist

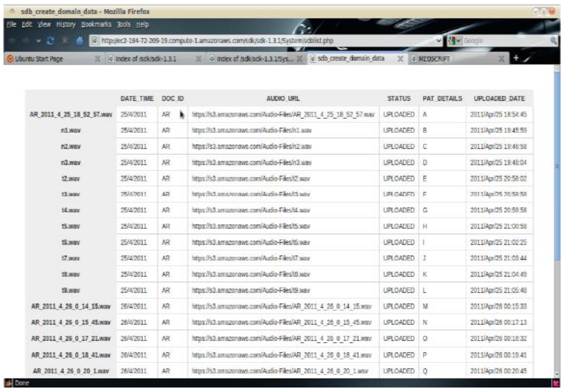

Figure 17. List of Pending Jobs

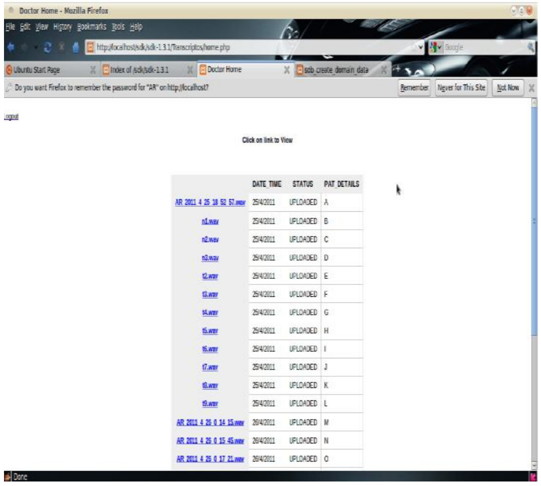

Figure 18. Job Details Per Doctor

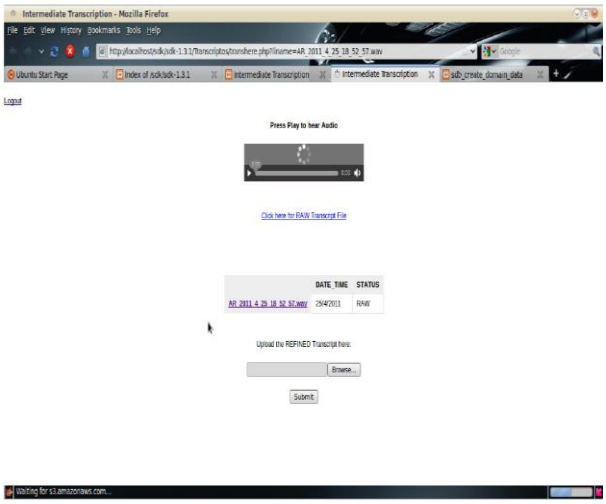

Figure 19. Transcriptionist refined upload

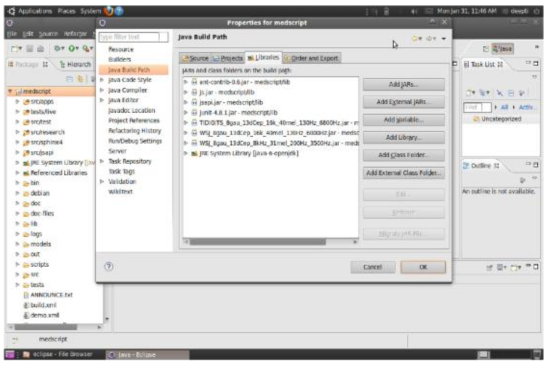

Figure 20. Adding required libraries in Sphinx

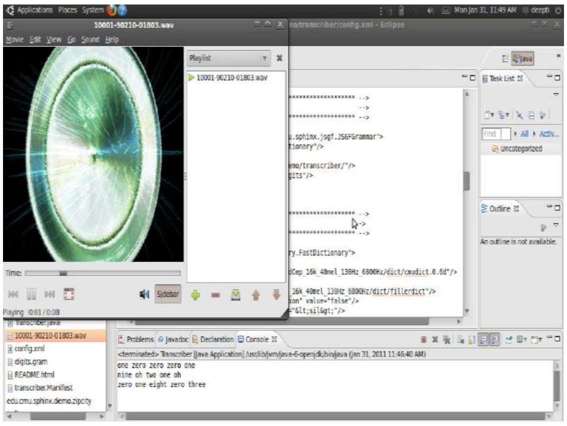

Figure 21. Testing audio file transcription in Sphinx

4. FINDINGS

The Sphinx system was tested in both the desktop system and in the proposed cloud framework. Figure 22. shows the runtime of the ASR process for different length audio files in both the desktop system and cloud. The configurations of both systems are given in Table 2. below:

Test Environment	RAM	Processor
Desktop system	3 GB	Intel Core2 Duo
Cloud instance	15 GB	8 EC2 (4 cores) Compute units

Table 2. Configuration Details

As shown in the graph below in Figure 22, the runtime of the ASR process in the cloud is reduced with the proposed Speech Recognition framework ,when compared to desktop system. The performance of the system in terms of accuracy is not shown as it depends on the training of the acoustic and language models of Sphinx system. With a well trained acoustic and language model, accuracy of the system can be improved. When a speech model (acoustic or language) is trained for a particular domain such as medical transcription, accuracy will be drastically improved as compared to a generic acoustic and language model.

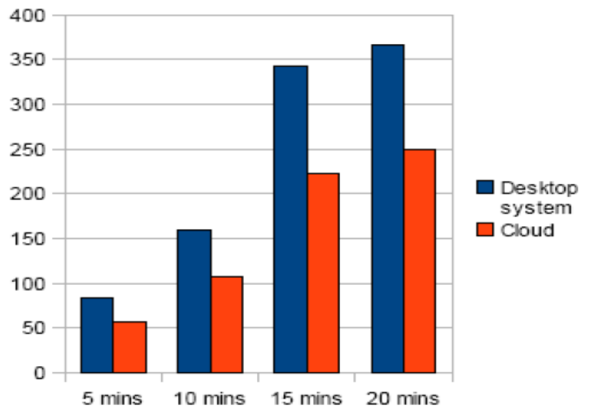

Figure 22.Comparison of ASR Systems on Desktop and Cloud

5. CONCLUSION

In this paper, an open framework for speech to text conversion is proposed . The proposed system involves collaboration between mobile devices and uses the power of cloud computing. The application is a basically an integration of the open source technologies to bring out a solution for the defined model. The paper has also proposed and provided suggestions how the proposed framework can be used to run the Medical Transcription application.

6. SCOPE FOR FUTURE WORK

Future work in this field would involve integration of the Medscript application into a Hospital Management System and how this application runs in a platform independent manner. Other improvements would be in terms of enhancing the accuracy of the system and making the user interface as simple as possible. However some of the major challenges that still remain are, firstly, to make it free from human intervention or nearly perfect transcript system which would not require an intermediate to correct the transcripts. Secondly, the speech recognition system would encounter different accents of the same language (Speakerindependent model) and lastly the issue of uninterrupted and good speed Internet connectivity on mobile devices. The speech data can also be interpreted for understanding the patterns[7].Gaussian Models could also be used for evaluating the speech in the framework[18] .Speech recognition system built on cloud could be extended to people with speech impairment [28][29].

Yet another revolutionary feature would be to transcribe real time. In this case every request is based over a session, and the transcription happens in real time, with the server keeping track of the requesting client and transcribing the incoming requests in bursts. Lot of security challenges would involved in such a framework[9].This can be built as part of a educational system[6][8]. The speech that is recorded could be extended and summarized for user in a concise manner[25][26].Speech compression could also be developed and encoded as part of the mobile standard[27].

7. References

[1] Kwang Mong Sim," Agent-Based Cloud Computing" IEEE Transactions on Services Computing, Vol.5,pp. 564 - 577,2012.

[2] David Chiu,Gagan N.Agrawal,"Evaluating caching and storage options on the Amazon Web Services Cloud" ,IEEE/ACM International Conference on Grid Computing, pp.17 - 24,2010.

[3] V."Juggy" Jagannathan., "The Careflow Architecture-A Case Study in Medical Transcription",IEEE Internet Computing,pp.59-64,2001.

[4] Bernd Grobauer,Tobias Walloschek,Elmar Stöcker, "Understanding Cloud Computing Vulnerabilities", IEEE Security & Privacy,Vol.9,pp.50 - 57,2011.

[5] Alexandru Iosup,Simon Ostermann,Nezih Nezih Yigitbasi,Radu Prodan,Thomas Fahringer,Dick H J Epema,"Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing",IEEE Transactions on Parallel and Distributed Systems, Vol.22,pp.931 - 945,2011.

[6] .A.Alshuwaier,A.A.Alshwaier,A.M.Areshey,"Applications of cloud computing in education",8th International Conference on Computing and Networking Technology (ICCNT),pp.26 - 33,2012.

[7] Arshdeep Bahga,Vijay K. Madisetti, "Analyzing Massive Machine Maintenance Data in a Computing Cloud",IEEE Transactions on Parallel and Distributed Systems, Vol.23,pp.1831 - 1843,2012.

[8] N.R.R.Mohan,E.B.Raj,"Resthisce Allocation Techniques in Cloud Computing -- Research Challenges for Applications", Fthisth International Conference on Computational Intelligence and Communication Networks (CICN),pp.556 - 560,2012.

[9] Khaled M.Khan,Qutaibah M. Malluhi,"Establishing Trust in Cloud Computing",IEEE,Vol.12, pp.20 - 27,2010

[10] Jorge Martins,João Pereira,Sergio M.Fernandes,João Cachopo, "Towards a Simple Programming Model in Cloud Computing Platforms",First International Symposium on Network Cloud Computing and Applications (NCCA),pp.83 - 90,2011.

[11] Lori M.Kaufman,"Data Security in the World of Cloud Computing",IEEE Security & Privacy, Vol.7 ,pp.61 - 64,2009.

[12]. Abu S.Asaduzzaman,Abilash Rao Joseph,Fadi N.Sibai,Nader Mohamed, " Cloud computing: A cloudy future?",International Conference on Innovations in Information Technology (IIT), pp.78 - 82,2012

[13] Qian Wang,Cong Wang, Kui Ren,Wenjing Lou,Jin Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing",IEEE Transactions on Parallel and Distributed Systems, Vol.22,pp.847 - 859,2011

[14] Mohamed Hamdi,"Security of cloud computing, storage, and networking",International Conference on Collaboration Technologies and Systems (CTS), pp.1-5,2012

[15] H. Gilbert Miller,John Veiga,"Cloud Computing: Will Commodity Services Benefit Users Long Term?",IT Professional,Vol.11,pp.57-59,2009

[16] Sven Anderson,Diane Kewley-Port, "Evaluation of speech recognizers for speech training applications",IEEE Transactions on Speech and Audio Processing, Vol.3,pp.229 - 241,1995.

[17] Liang Gu,Yuqing Gao,Fu-Hua Liu,Michael A.Picheny, "Concept-based speech-to-speech translation using maximum entropy models for statistical natural concept generation",IEEE Transactions on Audio, Speech, and Language Processing, Vol.14,pp.377 - 392,2006.

[18] Tomohiro Nakatani, ;Büng-Hwang Hwang Fred Juang,Takuya Yoshioka,Keisuke Kinoshita,Marc Delcroix,Masato Miyoshi, "Speech Dereverberation Based on Maximum-Likelihood Estimation With Time-Varying Gaussian Sthisce Model",IEEE Transactions on Audio, Speech, and Language Processing, Vol.16 ,pp.1512 - 1527,2008

[19] Ji Ming,Ramji Srinivasan,Danny Crookes, "A Corpus-Based Approach to Speech Enhancement From Nonstationary Noise",IEEE Transactions on Audio, Speech, and Language Processing, Vol.19,pp.822 - 836,2011

[20] Jont B. Allen,"How do humans process and recognize speech?",IEEE Transactions on Speech and Audio Processing,Vol.2 , pp. 567 - 577,1994

[21] John H L Hansen,Brian David Womack, "Feature analysis and neural network based classification of speech under stress",IEEE Transactions on Speech and Audio Processing, Vol.4,pp.307 - 313,1996.

[22] John H L Hansen,Vaishnevi Varadarajan, "Analysis and Compensation of Lombard Speech Across Noise Type and Levels With Application to In-Set/Out-of-Set Speaker Recognition",IEEE Transactions on Audio,Speech and Language Processing,Volume: 17,pp.366 - 378,2009.

[23]Jiaohua Hua Tao,Yongguo Kang,Ai-Jun Li,"Prosody conversion from neutral speech to emotional speech",IEEE Transactions on Audio, Speech, and Language Processing, Vol.14,pp.1145 - 1154,2006.

[24] Ki-Seung Lee,Richard V.Cox,"A very low bit rate speech coder based on a recognition/synthesis paradigm",IEEE Transactions on Speech and Audio Processing,Vol.9,pp.482 - 491,2011.

[25] Sadaoki Furui,Tomonori Kikuchi,Yousuke Shinnaka,Chiori Hori, "Speech-to-text and speech-to-speech summarization of spontaneous speech",IEEE

Transactions on Speech and Audio Processing, Vol.12,pp.401 - 408,2004.

[26] E. Bryan George,Mark J T Smith, "Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model",IEEE Transactions on Speech and Audio Processing,Vol.5,pp.389 - 406,1997.

[27] Sorin Dusan, James L. Flanagan,Amod Karve,Mridul Balaraman,"Speech Compression by Polynomial Approximation",IEEE Transactions on Audio, Speech, and Language Processing, Vol.15 ,pp.387 - 395,2007.

[28] M.S.Hawley,S.P.Cunningham,P.D.Green, P.Enderby,R.Palmer,S.Sehgal , P.O'Neill,"A Voice-Input Voice-Output Communication Aid for People With Severe Speech Impairment",IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol.21,pp.23 - 31,2013.

[29] Jiaohua Hua Tao,Yongguo Kang,Ai-Jun Li, "Prosody conversion from neutral speech to emotional speech",IEEE Transactions on Audio, Speech, and Language Processing, Vol.14 ,pp.1145 - 1154,2006.

[30].BrianHayes,"CloudComputing",Communications of the ACM - Web science Magazine Vol.51,pp.9-11,2008 .