

**NEWPORT INTERNATIONAL JOURNAL OF
ENGINEERING AND PHYSICAL SCIENCES (NIJEP)
Volume 3 Issue 1 2023**

**Design and Implementation of a Cross Platform
Document File Reader using Speech Synthesis**

^{1,2}Ogenyi Fabian Chukwudi and ¹Udeani Henrietta

**¹College of Engineering and Engineering Technology. Michael Okpara
University of Agriculture, Umudike. Abia State, Nigeria.**

**²Department of Publication and Extension Kampala International University
Uganda**

ABSTRACT

Document files are files for storing documents on storage media particularly for use by computer systems. These files are viewed by using software which displays the text contents of these files in readable format. Nevertheless, it is necessary to have applications for converting the electronic files into forms appropriate for use by people with certain impairments. This research study proposes the design and implementation of a document reader using speech synthesis technology. When a user provides a document file as an input, he/she will be able to listen to the system as it reads out the content of the document file. There are various options of materials for developing such a system, but the ones chosen in this research study are JDK8, Java, JavaFX framework, NetBeans amongst others. The methods used are structured design analysis, object-oriented design approach and top-down method. These methods were chosen because of their benefits in software development.

Keywords: Design, implementation, cross platform, document, file, reader, speech and synthesis.

INTRODUCTION

Document files are text or binary file for storing documents on a storage media especially used by computers. Currently, there exist a multitude of document file formats with file extensions such as .doc, .docx, .pdf, .txt, .xls and .xlsx. Such files can be viewed by using special software which displays the text contents of these files in readable format. However, there is need for dedicated applications which converts the electronic file into a form suitable for use by people with certain impairment. As a result, this research study proposes the development of a document reader using speech synthesis technology. Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer and can be implemented in software or hardware. A text-to-speech (TTS) system converts normal language text into speech. Therefore, text-to-speech synthesis can simply be seen as the automatic conversion of a text into speech that resembles, as closely as possible, a native speaker of the language reading that text. Text-to-speech synthesizer (TTS) is the technology which lets computer speak to you. [8] The TTS system gets the text as the input and then a computer algorithm called the TTS engine analyses the text, pre-processes the text and synthesizes the speech with some mathematical models. The TTS engine usually generates sound data in an audio format as the output. A text-to-speech engine is composed of two parts: a front-end and a back-end. The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called text normalization, pre-processing, or tokenization. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end, known as the synthesizer then converts the symbolic linguistic representation into sound. In certain systems, this part includes the

© Ogenyi Fabian Chukwudi and Udeani Henrietta

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Publications

computation of the target prosody (pitch contour, phoneme durations), which is then imposed on the output speech.

The text-to-speech synthesis procedure consists of two main phases. The first is text analysis, where the input text is transcribed into a phonetic or some other linguistic representation, and the second one is the generation of speech waveforms, where the output is produced from this phonetic and prosodic information. These two phases are usually called high and low-level synthesis. The input text might be data from a word processor, standard ASCII from e-mail, a mobile text-message, or scanned text from a newspaper. The character string is then pre-processed and analyzed into phonetic representation which is usually a string of phonemes with some additional information for correct intonation, duration, and stress. Speech sound is finally generated with the low-level synthesizer by the information from high-level one. The artificial production of speech-like sounds has a long history, with documented mechanical attempts dating to the eighteenth century. There are several cross-platform and platform-dependent systems for reading document files but most of them do not have text-to-speech synthesizers for reading out the content of document files in audio format. Adobe Acrobat is a family of application developed by Adobe Systems to view, create, manipulate, print and manage files in Portable Document Format (PDF) [1-3]. The basic Acrobat Reader, available for several desktop and mobile platforms, is freeware; it supports viewing, printing and annotating of PDF files.

Microsoft Office is a family of software developed by Microsoft. Initially a marketing term for an office suite, the first version of Office contained Microsoft Word, Microsoft Excel, and Microsoft PowerPoint. Microsoft Office prior to Office 2007 used proprietary file formats based on the OLE Compound File Binary Format [4]. Foxit Reader is a multilingual freemium PDF tool that can create, view, edit, digitally sign, and print PDF files [5]. Foxit Reader is developed by Fremont, California-based Foxit Software. Early versions of Foxit Reader were notable for startup performance and small file size [6]. Ghostscript is a suite of software based on an interpreter for Adobe Systems' PostScript and Portable Document Format (PDF) page description languages. Its main purposes are the rasterization or rendering of such page description language files, for the display or printing of document pages, and the conversion between PostScript and PDF files [7].

Aim of this research

The aim of this research is to design and implement a cross platform document file reader using speech synthesis.

Objectives of this Research

To accomplish this aim, there are many objectives to be achieved which include the following:

1. To develop an algorithm for the document file reader using speech synthesis.
2. To develop a source code based on the algorithm above using the Java programming language.
3. To integrate all necessary modules and libraries.
4. To test-run and implement the cross-platform system in different operating systems.

Scope and Limitations of the Research

The scope of this study covers the design and implementation of a cross platform document file reader using speech synthesis. The system can extract text from document files with extensions such as .doc, .docx, .pdf and .txt. The text is passed as input into the text-to-speech engine which reads it out through the computer's speakers. Optical Character Recognition (OCR) is not employed in this project work. All implementations are achieved using Java programming language. The limitation of this work is more on time available to conduct and design the system in the light of some difficulty encountered in getting or having access to classified document.

MATERIALS AND METHODS**Materials**

There are numerous options and choices in terms of software development tools, platform, system and technology. The materials used in this project include the following:

1. JDK8 (Java Development Toolkit version 8)
2. Java Programming Language
3. JavaFX framework (GUI development tool)
4. NetBeans (Integrated Development Environment)
5. Laptop (Development Machine)

JDK: is an abbreviation for **Java Development Toolkit**. It is an implementation of either one of the Java Platform: Standard Edition (SE), Enterprise Edition (EE), or Micro Edition (ME) platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on various operating system. The JDK includes a private Java Virtual Machine (JVM) and a few other resources to finish the development of a Java Application. Since the introduction of the Java platform, it has been by far the most widely used Software Development Kit (SDK). In 2006, Sun announced that they would release it under the GNU General Public License (GPL), thus making it free software. This happened in large part on 8 May 2007, when Sun contributed the source code to the OpenJDK.

© Ogenyi Fabian Chukwudi and Udeani Henrietta

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Publications

Java: is the programming language selected for implementing the cross-platform document file reader. Java is chosen because it supports the creation of cross-platform applications and provides the necessary Application Programming Interface (API) needed in the system implementation. It is a general-purpose programming language which is concurrent, object-oriented and specifically designed to have as few implementation dependencies as possible. It is intended to let software developers "write once, run anywhere" meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).

JavaFX: is preferred because it is a software platform that enables creating and delivering of desktop applications, as well as rich Internet applications (RIAs) that can run across a wide variety of devices. JavaFX is intended to replace Swing as the standard GUI library for Java SE. JavaFX has support for desktop computers and web browsers on various operating system such as Microsoft Windows, Linux, and macOS. Before version 2.0 of JavaFX, developers used a statically typed, declarative language called JavaFX Script to build JavaFX applications. Because JavaFX Script was compiled to Java bytecode, programmers could also use Java code instead. JavaFX applications could run on any desktop that could run Java SE or on any mobile phone that could run Java ME.

NetBeans: is an integrated development environment (IDE) for developing Java applications. In this project work, it is used for writing and editing program source code. NetBeans allows applications to be developed from a set of modular software components called modules. NetBeans runs on different operating system. In addition to Java development, it has extensions for other languages like C, C++, PHP, HTML5 and JavaScript. Applications based on NetBeans, including the NetBeans IDE, can be extended by third party developers. The NetBeans Team actively supports the product and seeks feature suggestions from the wider community. Every release is preceded by a time for Community testing and feedback.

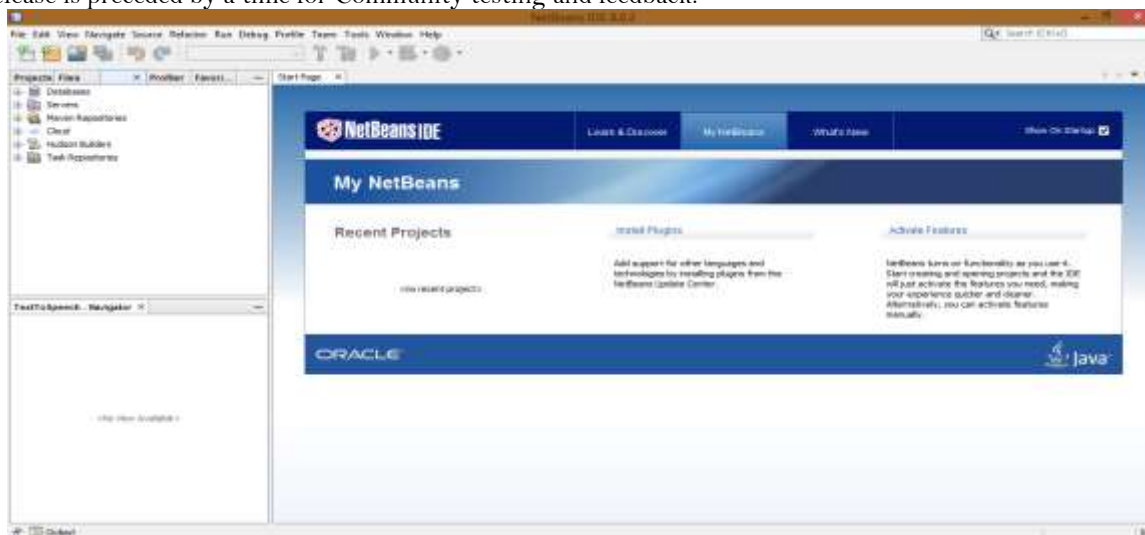


Figure 1: The NetBeans IDE

Laptop: also known as a notebook or "notebook computer", is a small, portable personal computer with a "clamshell" form factor, an alphanumeric keyboard on the lower part of the "clamshell" and a thin LCD or LED computer screen on the upper part, which is opened up to use the computer. PCs are intended to be operated directly by an end-user, rather than by a computer expert or technician. Computers were invented to compute and solve complex mathematical problems, but today, due to media dependency and the everyday use of computers, it is seen that computing is the least important thing computers do. This material served as the development and testing machine during the project implementation.

© Ogenyi Fabian Chukwudi and Udeani Henrietta

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

METHODS

Methodology is a system of methods that is followed in a particular subject area. Several methods exist for analysis and design during the system development process. Each process model follows a Series of steps unique to its type to ensure success in the process of system development.

Choice of Methods Used

The project study applies some approaches to design and implement the cross-platform document file reader system. The reasons for selecting these approaches amongst others stems from the benefits offered by them. These methods are:

- Structured design analysis
- Object-oriented design approach
- Top down method

Structured Design Analysis: is a conceptualization of problem into several well-organized elements of solution. It is basically concerned with the solution design. Benefit of structured design is; it gives better understanding of how the problem is being solved. Structured design also makes it simpler for designer to concentrate on the problem more accurately. Structured design is mostly based on 'divide and conquer' strategy where a problem is broken into several small problems and each small problem is individually solved until the whole problem is solved. The small pieces of problem are solved by means of solution modules. Structured design emphasizes that these modules be well organized in order to achieve precise solution. These modules are arranged in hierarchy. They communicate with each other. A good structured design always follows some rules for communication among multiple modules, namely:

- **Cohesion:** grouping of all functionally related elements.
- **Coupling:** communication between different modules.

A good structured design has high cohesion and low coupling arrangements.

Object-Oriented Design: This works around the entities and their characteristics instead of functions involved in the software system. This design strategy focuses on entities and its characteristics. The whole concept of software solution revolves around the engaged entities. Some important concepts of Object Oriented Design are:

- ❖ **Objects:** All entities involved in the solution design are known as objects. For example, files are considered as objects in this project implementation. Every entity has some attributes associated to it and has some methods to perform on the attributes.
- ❖ **Classes:** This is a generalized description of an object. An object is an instance of a class. Class defines all the attributes, which an object can have and methods, which defines the functionality of the object. In the solution design, attributes are stored as variables and functionalities are defined by means of methods.
- ❖ **Encapsulation:** In object-oriented design, the attributes (data variables) and methods (operation on the data) are bundled together and is called *encapsulation*. Encapsulation not only bundles important information of an object together, but also restricts access of the data and methods from the outside world. This is called information hiding.
- ❖ **Inheritance:** Object oriented design allows similar classes to stack up in hierarchical manner where the lower or sub-classes can import, implement and re-use allowed variables and methods from their immediate super classes. This property of object-oriented design is known as inheritance. This makes it easier to define specific class and to create generalized classes from specific ones.
- ❖ **Polymorphism:** Object oriented design languages provide a mechanism where methods performing similar tasks but vary in arguments, can be assigned same name. This is called polymorphism, which allows a single interface performing tasks for different types. Depending upon how the function is invoked, respective portion of the code gets executed.

Top Down Design: This design pattern takes the whole software system as one entity and then decomposes it to achieve more than one sub-system or component based on some characteristics. Each sub-system or component is then treated as a system and decomposed further. This process keeps on running until the lowest level of system in the top-down hierarchy is achieved. Top-down design is more suitable when the software solution needs to be designed from scratch and specific details are unknown.

Some benefits offered by this design approach are:

- ❖ Easy to understand
- ❖ Readable:
- ❖ Code reuse

Development Procedure

For efficient design and development of this cross-platform document file reader system, well-defined and structured sequence of stages was followed. These series of steps are listed below:

© Ogenyi Fabian Chukwudi and Udeani Henrietta

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Publications

- Requirement Analysis
- Feasibility Study
- System Analysis
- Software Design
- Coding
- Testing
- Integration
- Implementation

Data Gathering

The methods used for data collection are:

- **Website Research:** Different document on the World Wide Web (WWW) were reviewed to find information which was essential in writing the chapters and software design.
- **Library Research:** Books, journals and magazines were reviewed to collect all necessary information which were also used in writing and designing the new system.

Analysis of the Existing System

Most existing systems for reading document files do not have text-to-speech synthesizers which aids in reading out the content of document files through a speaker. With these systems, users can only read the contents of the files through the application's viewer that display the text contents for viewing by the user.

Short Falls of the Existing System

As the current system does not have a speech synthesizer for converting text to speech, it poses challenges to individuals with visual impairment. It also limits the ability of persons with learning and literacy difficulty.

The Proposed System

The proposed system is a cross-platform software system for reading document files developed using the java programming language; it applies speech synthesis technology to convert normal human language text into speech. With the aid of this system, users can listen to the content of document files with extensions such as .pdf, .doc, .docx and .txt as being processed by the speech synthesis engine [8] the system's operation comprises of three phases: the first step is the extraction of text content from the document file using a suitable algorithm. The second is text analysis, where the input text is transcribed into a phonetic or some other linguistic representation, and the third one is the generation of speech waveforms, where the output is produced from this phonetic and prosodic information

System Design

System design covers the evaluation of alternative development solutions and the specification of detailed computer-based solution. While the analysis focuses on the logical, implementation-independent aspects of a system, design deals with the physical or implementation-dependent aspect of the system. The input, output and process design are discussed in this section.

Input and Output Design

This handles the plan on how data get into the cross-platform document file reader and also how processed information get out of the system.

Input Design: Inputs into the system are entered by the user; the major input are document files (text file, PDF file, word document, etc.) which are selected using file dialog box. Other inputs are designed using input controls such as textbox, radio button and slider.

Output Design: System interface is projected using JavaFx framework. As explained earlier, JavaFx is a software platform that enables creating and delivering of desktop applications, as well as rich Internet applications (RIAs) that can run across a wide variety of devices. Outputs contain the result of the processed inputs; they are what the user sees on the screen. These outputs are displayed through the interface, which are interactive with the user. Voice output are presented through the system speakers.

Process Design

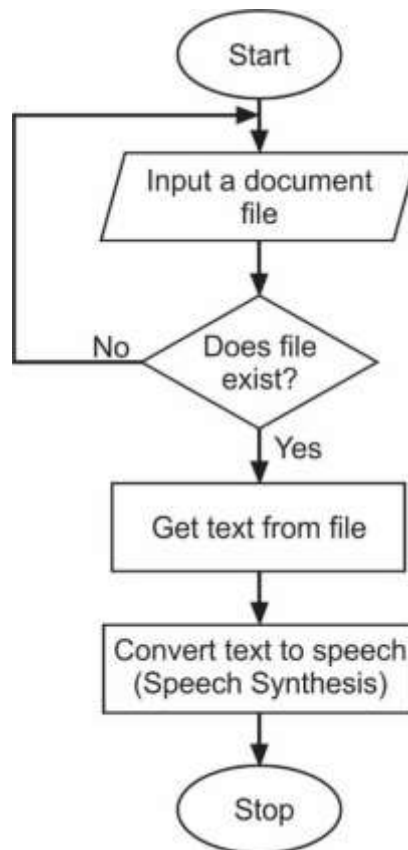
The system process design plays a vital role in the system implementation. In order to understand the working procedure, process design is necessary. In this project work, flow chart and block diagram are the tools used for process design. Flow chart is a graphical representation of the system showing the overall flow of control; it specifies what activities must be done to convert from a physical to logical model. Block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks

Figure2: Flow Chart of the Proposed System

The flow chart of the proposed system is shown below:

© Ogenyi Fabian Chukwudi and Udeani Henrietta

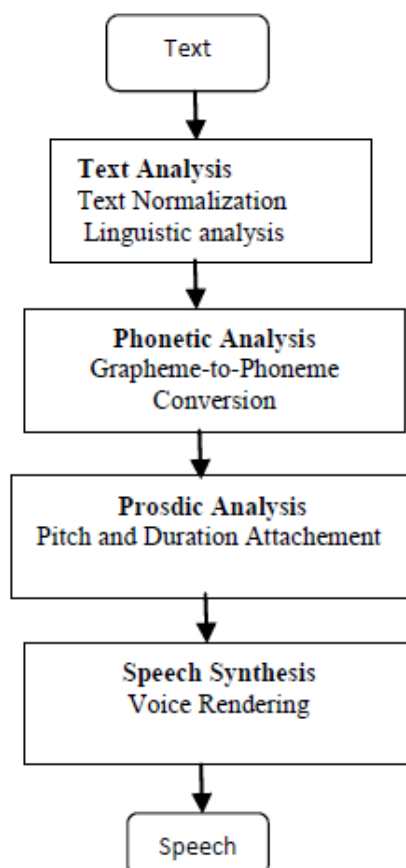
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



From the flow chart, the system user needs to input a document file into the system after which the text contents of the file are extracted using a suitable algorithm. The text is then converted to speech by the speech synthesizer

Figure 3: Flow Diagram of the Speech Synthesizer

The flow diagram of the speech synthesizer is shown below:



Text analysis: consists of normalization of the text wherein the numbers and symbols become words and abbreviations are replaced by their whole words or phrases etc. The most challenging task in the text analysis block is the linguistic analysis which means syntactic and semantic analysis and aims at understanding the context of the text.

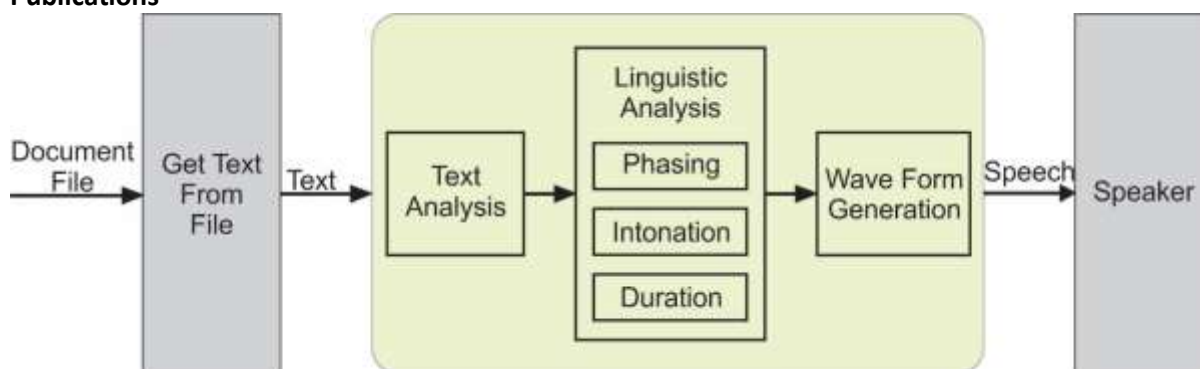
Phonetic Analysis converts the orthographical symbols into phonological ones using a phonetic alphabet. For e.g. the alphabet of the International Phonetic Association contains phoneme symbols, their diacritical marks and other symbols related to their pronunciation.

Prosody is a concept that contains the rhythm of speech, stress patterns and intonation. At the perceptual level, naturalness in speech is attributed to certain properties of the speech signal related to audible changes in pitch, loudness and syllabic length, collectively called prosody.

Speech Synthesis block finally generates the speech signal. This can be achieved either based on parametric representation, in which phoneme realizations are produced by machine, or by selecting speech units from a database. The resulting short units of speech are joined together to produce the final speech signal.

Figure 4: Block Diagram of the Proposed System

The block diagram of the proposed system is depicted below:



From the diagram above, the main input that the system requires is a document file. [9-10]. The text content of this file is extracted using a suitable algorithm. The speech synthesis engine takes in the text as input and then pre-processes and analyzes it into phonetic representation which is usually a string of phonemes with some additional information for correct intonation, duration, and stress. A sound wave is then generated and passed to the computer's speaker. The speaker enables a user to hear the audio output.

RESULTS AND DISCUSSION

System Implementation

System implementation involves the actualization of an application or execution of a plan, idea, model, design, specification, or algorithm. It is also the realization of a technical specification or algorithm as a program, software component or other computer system through programming and deployment. The new system was implemented as a standalone software application with the purpose of making it available to individuals who need a voice-based document file reader for various purpose such as persons with visual impairment. The system is implemented with a user-friendly interface which means that the user needs little or no guidelines on how to use the voice-based document file reader.

System Requirements

System requirement deals with the essential hardware and software requirement that is relevant for the efficient development and execution of the cross-platform document file reader using text-to-speech synthesis.

Hardware Requirements

The hardware components of a digital computer system are the physical parts that makes up the computer system. The system can be implemented and executed provided that certain hardware components are available. The following hardware requirement are necessary for the efficient work of the system:

1. Processing Unit: Intel Pentium II or later
2. Main memory: Capacity of at least 1G of RAM
3. Hard disk space: Hard disk of at least 20GB
4. Output devices: Monitor and Speaker
5. Input devices: Keyboard and mouse

Software Requirements

Software, is a generic term that refers to a collection of data or computer instructions that tell the computer how to work, in contrast to the physical hardware from which the system is built, that actually performs the work. In other words, software is a set of programs, procedures, algorithms and its documentation concerned with the operation of a data processing system. The following are the software requirement for developing and executing the cross-platform document file reader.

1. Windows operating system (7 or later)
2. Java Development Kit (JDK) version 7 or later

Therefore, a system that meets the hardware and software requirement explained above can be used to efficiently develop and run the system.

RESULTS AND ANALYSIS

With suitable design, implementation and testing, the system was an efficient, functional and reliable cross-platform document file reader. The system operated properly and adequately thereby meeting the minimum expectations that was needed initially. It is expected to be beneficial to individuals with visual impairment owing to the application of speech synthesis. The user interfaces (UIs) of the developed system are presented and described below which consist of various GUI components such as labels, text fields, radio buttons, checkbox, text area, sliders and buttons.

© Ogenyi Fabian Chukwudi and Udeani Henrietta

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

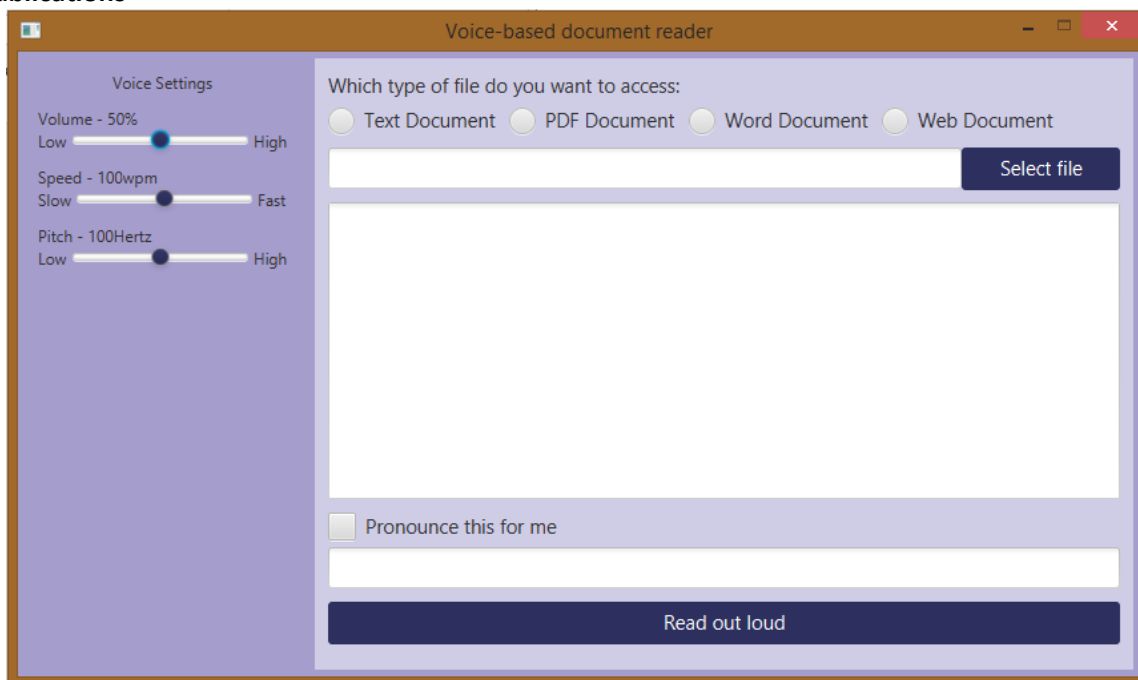


Figure 5: Voice-Based Document File Reader Window

From the cross-platform voice-based document file reader window shown above, some voice parameters can be set using the three sliders on the left-hand side. The sliders are meant for setting values for volume, speed and pitch. The radio buttons are used to determine which type of document file is to be loaded when the “Select file” button is clicked. There are four radio buttons for selecting text documents, PDF documents, word documents and web documents such as HTML files, JavaScript’s files, PHP files etc. Clicking the “Select file” button opens a file dialog box from which a file is chosen. After selecting the file, the text contents of the file are extracted using a suitable extraction algorithm and displayed on the text area. To make the system read out the text content, the “Read out loud” button must be clicked. Reading out of file’s text contents are done in a separate thread which means that there is no blocking in the user interface thread. The user can also type a word or sentence into the text field below the “Pronounce this for me” check box; but the check box must be checked before clicking the “Read out loud” button will read out the typed text

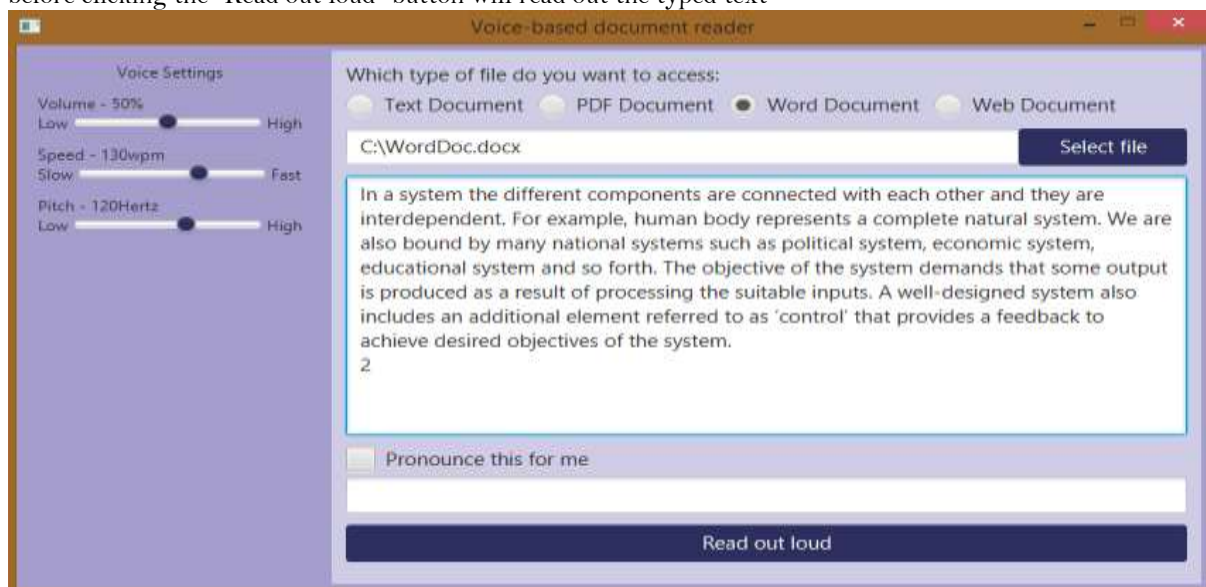
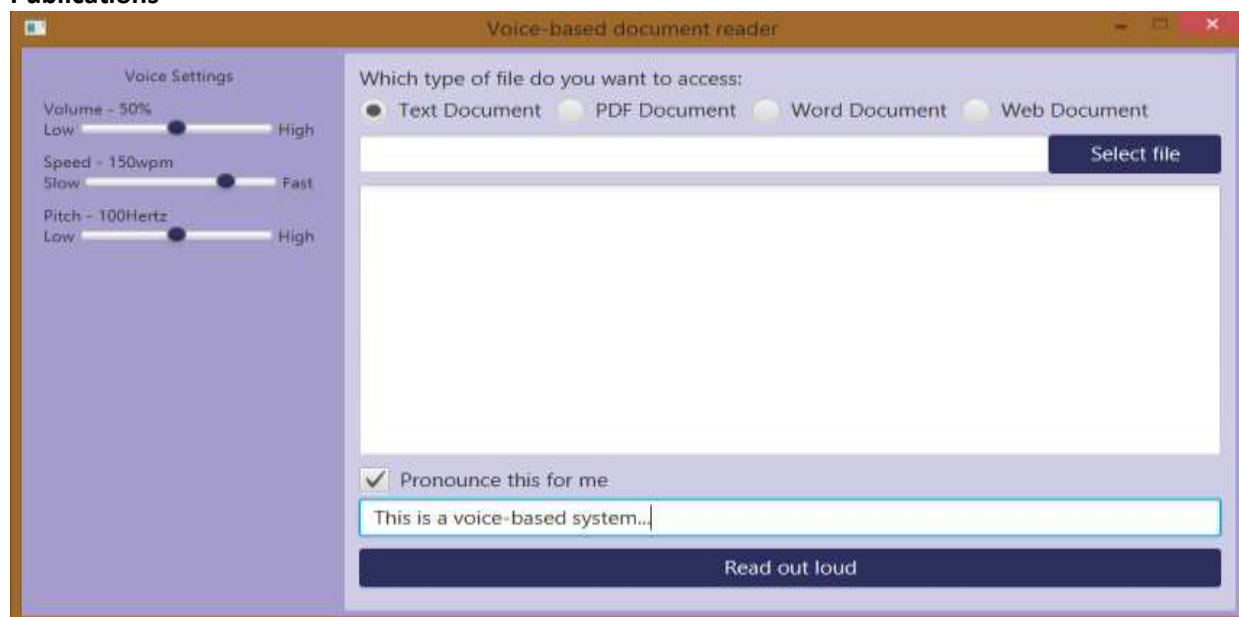


Figure 6: Reading Extracted Text from a Word Document

© Ogenyi Fabian Chukwudi and Udeani Henrietta

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



**Figure 7: Reading Typed Text
System Testing and Integration**

System testing consist of the various activities executed to uncover possible problems that might still be found in the designed system. Program testing involves the testing of the programs designed to see how they function individually. It also involves the testing of different programs, the system and how they interact with one another. System testing uncovers weakness that was not found in earlier testing. The testing normally begins with low volumes of data to the upper bond. The system as a whole is tested for recovery and fall back after various major features to ensure that no data had been lost during the emergency. As a cross-platform system, the system was tested in various versions of windows operating system and Linux.

CONCLUSION

The visually impaired can get outside information only through auditory sense and tactile sense. But in reality, a lot of information still spreads in the form of text document which can be converted to speech using text-to-speech synthesis. Speech synthesis has been developed steadily over the last decades and it has been incorporated into several new applications. For most applications, the intelligibility and comprehensibility of synthetic speech have reached the acceptable level. [11] However, in prosodic, text preprocessing, and pronunciation fields there is still much work and improvements to be done to achieve more natural sounding speech. Natural speech has so many dynamic changes that perfect naturalness may be impossible to achieve. Present speech synthesis systems are so complicated that one researcher cannot handle the entire system. With good modularity it is possible to divide the system into several individual modules whose developing process can be done separately if the communication between the modules is made carefully.

RECOMMENDATIONS

In the future, Optical Character Recognition (OCR) can be used to predict the input text with pre-loaded database template. Both the characters can be compared if it matches and then using text to speech synthesizer, speech output is produced; a hardware system can be built and integrated. For the implementation using hardware, a finger module with camera mounted with a vibrator sensor can be used. The camera will be used to capture the text which can be used as an input and the vibration sensor helping to indicate the effective line by line reading. With the help of this system, the user will find it easier to read the text in the form of speech using OCR and text-to-speech synthesis.

REFERENCES

1. Adobe Acrobat family (2008). Archived from the original on 19 January 2008. Retrieved 19 January 2008.
2. Archana Balyan, S. S. Agrawal, Amita Dev (2013) Speech Synthesis: A Review International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 www.ijert.org IJERT Vol. 2 Issue 6, June – 2013.
3. Lee K. (1989). Hidden Markov Models: Past, Present, and Future. Proceedings of Eurospeech 89 (1): 148-155.
4. Microsoft Office File Formats (2013). MSDN Library. Microsoft. Retrieved 2 February 2013.

© Ogenyi Fabian Chukwudi and Udeani Henrietta

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Publications

5. Popa, Bogdan (2014). Foxit Reader. Softpedia. SoftNews.
6. Foxit Software Foxit Reader 3.0 (2015). PC World. Retrieved 2015-07-24.
7. Ingo, Henrik (2006). Open Life: The Philosophy of Open Source. Lulu.com – via Google Books.
8. Itunuoluwa Isewon, Jelili Oyelade, Olufunke Oladipupo (2014) Design and Implementation of Text to Speech Conversion for Visually Impaired People International Journal of Applied Information Systems (IJAIS) – ISSN: 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 7– No. 2, April 2014 – www.ijais.org.
9. Klatt D. (1987) Review of Text-to-Speech Conversion for English. Journal of the Acoustical Society of America, JASA vol. 82 (3), pp.737-793.
10. Flanagan J., Rabiner L. (Editors) (1973). Speech Synthesis. Dowden, Hutchinson & Ross, Inc., Pennsylvania.
11. The benefits of Speech Synthesis – ReadSpeaker < <https://www.readspeaker.com/benefits-of-text-to-speech/>>Speech Synthesis –Wikipedia < https://en.wikipedia.org/wiki/Speech_synthesis>

Ogenyi Fabian Chukwudi and Udeani Henrietta (2023). Design and Implementation of a Cross Platform Document File Reader using Speech Synthesis. NEWPORT INTERNATIONAL JOURNAL OF ENGINEERING AND PHYSICAL SCIENCES (NIJEP) 3(1):1-11.