


## Article

# An Intelligent and Interactive Interface to Support Symmetrical Collaborative Educational Writing among Visually Impaired and Sighted Users

Mirza Muhammad Waqar <sup>1,\*</sup> , Muhammad Aslam <sup>1</sup> and Muhammad Farhan <sup>2</sup> 

<sup>1</sup> Department of Computer Science and Engineering, University of Engineering and Technology, Lahore 54890, Pakistan; maslam@uet.edu.pk

<sup>2</sup> Department of Computer Science, COMSATS University Islamabad, Sahiwal Campus 57000, Pakistan; farhansajid@gmail.com

\* Correspondence: mirzamwaqar@hotmail.com; Tel.: +92-321-4264457

Received: 11 December 2018; Accepted: 13 February 2019; Published: 15 February 2019



**Abstract:** It is often uncomfortable for disabled individuals, especially those with vision impairment, to conduct educational activities in collaboration with people that have perfect vision. This can be because of the former's lack of confidence, vision capability, and acceptance. Information and communications technology (ICT) has played a vital role in giving support to people with visual impairments so that they can overcome their issues. This study proposes innovative solutions that address the challenges faced by partially or completely visually impaired people. It provides an interactive and intelligent interface, which they may use to perform educational activities, such as editing, writing, or reviewing documents, in collaboration with people without visual impairments. The system provides high-quality awareness features by sending them instant voice notifications about the actions and events occurring in the shared environment. A speech-recognition engine has been integrated into the system to allow users to interact with the application through voice commands. The system is evaluated through experiments, where people with visual impairment and people without visual impairment were engaged in collaborative writing. The obtained results are encouraging. The users showed curiosity in the system and were able to focus on the productive task instead of their disability.

**Keywords:** collaborative environment; human-computer interaction; speech-based application; educational technology; coauthoring for blinds

## 1. Introduction

It is worth mentioning that more than 253 million people are visually impaired according to the World Health Organization (WHO) [1]. Given that they make up a substantial part of society, it is imperative to ensure that they may actively participate in social activities/kinetics and interact with others effectively. The objective should be to help them become self-reliant and confident. Existing information and communications technology (ICT) tools have been designed particularly for users with normal vision. Users with visual impairment have very few interactive user interface (UI) components available to them, which in turn are not very useful to them. In order to interact with this kind of application, users with such a disability employ assistive technologies and add-ons, such as a braille translator, voice recognition tools [2], speech synthesizers/screen readers [3,4], and so on.

Computer supported cooperative work (CSCW) [5] allows users to work collaboratively with each other on a single goal or task. CSCW helps users to obtain diverse knowledge and skills, which is not easily achieved if they work alone. Thus, individuals can be more productive if they work

collaboratively in a group [6]. However, individuals with visual impairments struggle when they interact with such applications [7], as they are not particularly designed for them. Add-ons [8] and interface wrappers [9] have been developed to make applications accessible to them, however, they are still not fully utilizable.

In CSCW applications, to make the work worthy and effective, all aspects of cooperative work, i.e., interaction, cooperation, collaboration, awareness, and coordination, are necessary [10]. The most important factors are an awareness of and interaction with the applications themselves [11,12]. The system must provide an interactive interface that is easily operable by users and can give complete awareness about the actions and events happening in the collaborative environment. Normally, awareness features are built as popup notifications and color formatting over contents. These mechanisms are visuals and do not work for users with extreme visual impairment. Similarly, to interact with the application, the standard input/output devices (mouse, keyboard, liquid crystal display (LCD) monitor, and so on) are ones that a user without visual impairments can easily use and are hard or impossible to use for individuals who cannot see. Existing solutions require the introduction of advanced components and features (audible alert, sound beeps, speech-based input, and so on) specifically designed for people with visual impairments to let them participate equally in collaborative work.

The focus of this research is to enable persons with visual impairments to become self-reliant, confident, and independent, and facilitate them to participate equally, alongside individuals was developed. This framework includes speech-based inputs and awareness functions particularly designed for users who are blind. The proposed system allows visually impaired users to work in a collaborative environment with sighted users on a single goal and has a special feature of voice command input for blind users to interact with the application efficiently. Interaction through voice allows them to use the application without any hustle and assistance. Both of these complement each other with the aim of making an interactive and intelligent interface. The application is embedded with well-structured information/notification components and has an easy-to-use interface. The occurrence of every event is shared with the users to keep every participant on the same track. Moreover, a communication service is added so that information may be easily exchanged between individuals without visual impairments and those with visual impairments.

## 2. Literature Review

When a group of users works on a single task, the productivity and the quality of the work increases and decision making becomes faster and more efficient [13]. This is achieved through a groupware environment that has the functions of collaboration and coordination in it [7]. To achieve a high coordination and collaboration quality, group awareness and the support of coordination are essential. This is related to the activities of each user working in the shared platform, the information of the participating authors, and the effect of each author's activity on the activity of others [14].

A survey was conducted [15] with blind people to gauge whether they have any past experience of group activity and if they had done any group activity, then what were the issues and deficiencies they faced. Contrastingly, if no issues were faced, what were their expectations for such an environment. A questionnaire was distributed among global special education institutes and individuals that were classified as visually impaired through email. Around 150 responses were received, and the results showed that the blind community has a need for a system that is particularly designed for them and can help them to work on a single task in collaboration with sighted people.

### 2.1. Overview of Computer Supported Cooperative Work (CSCW) Systems

CSCW systems are used by many big organizations. They help collaborators because they manage asynchronous as well as synchronous communication [16]. Java applets made multiuser (JAMM) [17] and synchronous asynchronous structured shared editor (SASSE) [18] are examples of these collaborator systems. Inconsistency is considered one of the major issues in these systems.

Inconsistency refers to the duplication of the same content in different parts of the document due to collaborative work. Yang et al. [19] presented a consistency model solution to avoid inconsistency problems. The Clay [20] system was proposed by Locasto et al. It allows users to work in a synced manner, from different geographic locations.

Gutwin and Greenberg [21] developed a descriptive theory of awareness that guides the developers of groupware applications about the importance of and need for awareness functions in a shared workspace. The proposed framework helps the designers to understand the concept for the purposes of designing awareness support and improving the quality of group awareness in a collaborative environment. The group awareness knowledge-based system (GAKS) [5] is a web-based application that enhances the coauthors' document writing abilities by providing them with elaborated and innovative awareness functionalities. The proposed system provides synchronous/asynchronous contextual communication tools and a work proximity detector for the users to efficiently produce and coordinate their actions. Big Watch (BW) [22] is a framework that provides flexible and extensible awareness functionalities to its users. The proposed framework can be integrated into any application to enhance its event-based awareness functions. The framework reduces the development cost and extends the awareness information in a unified way.

An extensible markup language (XML) based co-authoring platform was created by Qingzhang et al. [23] for collaborative working. Another XML based framework was also presented by Ho, Leong et al. [24]. In XML-based systems, shared documents are converted and stored in XML format. The advantage of using XML is that it stores the information in a structured format that is easy to read for machines and humans. Thus, the processing of the documents becomes easier and it helps to manage resources and access, as well as locking the content of documents. Another system with the name of WoTel [25] allows its authors to conduct video conferences to share different ideas while working on a shared document. To conduct group communications, multimedia systems are integrated for collaboration.

An asynchronous co-authoring system named TeNDeX was developed by Hodel et al. [26], which allows its users to edit documents synchronously. In the proposed system, a document's content is saved in a database, despite the existing conventional co-authoring platforms, which increased the efficiency of data retrieval. Joeris et al. [27] also proposed another application of synchronous collaboration, which supports the engineering domain. The work done in [28] allows its users to write mathematical expressions collaboratively. It provides, within the interface, the option of writing formulae, obtaining suggestions from old written formulae, reusing them, and evaluating them. In the case of any difficulty, it also suggests to currently available expert authors to its user so that they may get help from them.

## 2.2. Technologies and Applications for the Visually Impaired to Help in Document Writing

Assistive applications contain special user interface/user experiences (UI/UX) components, which enable blind users to produce work through them. There might be three ways to manage UI/UX for these kinds of users. Well formatted and good quality visual content for partially blind or color blind people as they have sight, but are unable to see clearly; text to speech function and special assistive hardware devices for the completely blind as they cannot see; and sound/speech based alerts and voice input controls for both are the best ways to interact with applications for persons who are visually impaired. All these things are a challenge to achieve in one framework. However, some frameworks support some individual features. For instance, nowadays, speech-based assistants [29–31] are already used, and are specifically designed for visually impaired people; and a similar kind of application may be designed to help said people with their educational and learning activities. A lot of chat bots have arrived, which use natural language processing (NLP) to communicate with users and do not even let the communicator know that there is a bot behind the screen [32]. One good application of such systems is for the elderly and disabled people, who suffer from loneliness

and do not have a vast social life. The system can provide them with the benefit of a distributed network, and help them stay updated with their surroundings [33].

Google Docs UI developed by Mori et al. [9] encountered the major problems faced by people who are blind while using Google Docs [34] via screen readers. The proposed UI has the same look and feel as the original Google Docs, but the accessibility to interactive elements was improved by integrating a new standard of (X)HTML interactive widgets (links, menu, buttons, and so on). To improve the orientation for the blind, the accessible rich internet applications (ARIA) [35] landmarks and hidden labels were added in the modified layout. The TinyMCE (Tiny Moxiecode Content Editor) editor was used to replace the existing one. This is more accessible through the keyboard and screen reader. In addition, to provide quick information about the document list, summary attributes were added to the document list tables. The real-time informative message issue was solved by using Ajax scripting.

A Microsoft Word add-in prototype [36] was developed to improve the usability and accessibility of collaborative writing between visually impaired individuals. The research was initiated with a baseline usability study [37], conducted to identify the accessibility and usability-related issues that stem from collaborative writing features when they are used by visually impaired people while using Microsoft Word. The author proposed a Word add-in prototype [8] that utilized Windows message boxes to present the revisions and comments of the document. It is compatible with the Job Access with Speech (JAWS) screen reader and a standard keyboard. In their next proposal, they used an iterative design approach that was conducted in two rounds of one usability study [38]. A group of blind candidates shared their feedback and suggestions after each iteration to improve the current version of the add-in. Based on the suggested improvements, the authors modified the prototype.

TalkMaths [39] provides blind people with a system that helps them create and edit high precision mathematical formulas. Automatic speech recognition (ASR) and dragon naturally speaking (DNS) is used to recognize speech and give textual output in the form of “parse tree”. Moreover, TalkMaths can recognize and detect syntactic errors. The initial version of TalkMaths [40] was designed for English language users only. An editing mode was also devised for the system, where the user can only delete the last typed digit. Work has started on the “DNS select-and-say” topology, like the mouse “point-and-click” strategy, to improve the editing mode. Users then select a specific label by dictating its position, as each box would have a sequence number attached to it. After selecting the appropriate box, the user then dictates the correct statement to overwrite the existing one.

Writing Mathematics by Speech [41] uses speech input techniques to enable blind students to read, write, and edit their mathematical expressions as quickly and precisely as sighted peers do. The proposed solution is an extension of the linear access to mathematics for braille device and audio-synthesis (LAMBDA) system [42], which is based on the functional integration of a linear mathematical code and an editor to visualize, write, and manipulate the formulas. It was designed to be used with Braille peripherals and the vocal synthesis. For speech recognition, Dragon Naturally Speaking TM 9 is used. The proposed prototype is made up of a script written for the deployment of Dragon Naturally Speaking TM, two dictionaries (one for text input and the second is for mathematic input), and a python script, which enables the LAMBDA editor to perform actions on the mathematical expressions.

A web-based application [43] takes input by speech for the writing of mathematical formulas. It is highly accessible with good usability features. The proposed application is context-sensitive, and its functionality is divided into various categories, where each category is forced to use a specific syntax that reduces the risk of errors in speech recognition and ultimately writes an accurate formula. This application requires prosody to minimize voice readout problems, which affect the desired result. On the development side, the author decided to use the JAVA language and Extensible Hypertext Markup Language (XHTML) + Voice Profile which controls voice processing and supports fast voice recognition. The final mathematical formula expression was written using MathML, which is preferred with regards to the existing standards. A two-layered system was introduced; the bottom layer is

made from JAVA that runs on Jetty, and for the top layer, graphical user interface (GUI) and Opera Browser were chosen because they support XHTML + Voice Profile technology.

A Software Model to Support Collaboration Mathematical Work between Braille and Sighted Users [44] provides an environment in which the people who are blind do cooperative work with sighted people. The system synchronizes two different perspectives of a mathematical formula, one for the person who is blind and the other for sighted people. The expression is presented to the blind by using braille, whereas a graphical illustration is used for sighted people. Support functions were included to allow visionless people to perform calculations easily. A hybrid entering method was also used to insert simple expressions via a keyboard and complicated expressions via speech. The Universal Math conversion library is used to deal with the fact that each math-based software has one associated Braille code. Switching from one Braille code to another is made possible through this library. Universal maths conversion library (UMCL) [45] consists of one major segment whereas the input and output segments depend on the number of mathematical formulae present. The Canonical MathML, a method to unite MathML segments, is used to speed up the evaluation time for the mathematical formulae.

Supporting Cross-Model Collaboration in the Workplace [46] presents a cross-modal tool for collaborative editing of diagrams between visually-impaired and sighted users. Initially, the authors had designed a single user auditory interface [47] to construct nodes-and-links diagrams, such as organizational charts, flow diagrams, unified modeling language (UML), transport maps, etc. The proposed system is an extended form of that system. The system has different views: The Graphical View is similar to a typical diagram editor, having a toolbar, mouse clicks, drag and drop functions, and keyboard shortcuts; in the Hierarchical Auditory view, the diagram is translated into auditory form from a tree-like hierarchical data structure to support non-visual interactions. In the Spatial Haptic representation, the PHANTOM Omni (a 3D mouse with a 'pen') haptic device is used for displaying the contents on a vertical plane where nodes act as a magnetic point. The user simply traces the stylus across the lines. The system allows haptic and auditory hierarchical views to work together in which the user locates its items and gives a command so that PHANTOM locates them on the virtual plane.

An Initial Investigation into Non-Visual Computer Supported Collaboration [48] provides a collaborative environment to visually impaired users to interact with and perform manipulations on simple graphs. This is an advancement of an existing application, Graph-Builder [49], that uses the PHANTOM Omni device to allow browsing and modification of bar graphs via haptic force feedback. The proposed system uses two PHANTOM Omni devices to build a collaborative environment. Two users are allowed to manipulate the same graph simultaneously, but cannot concurrently modify the same bar. Auditory signals are employed for one user to know the other one's location. For interaction, two features are employed: "Come to Me" (a user uses his Omni device to haptically drag the other one's device to its current location) and "Go to You" (a user lets his Omni device get dragged to the other user's proxy).

Some other works have also been conducted along with educational activities to support CSCW between persons who are blind and sighted. Multimodal tools and interfaces [50] are developed to facilitate intercommunication and interaction between a user who is blind and those who cannot listen. It uses the modality replacement function for information transition and enables communication between the users. Stacy et. al. [51] explored the creation and management of accessibility in a shared environment and identified the challenges and solutions of collaborative accessibility. Based on these experiences, they proposed new methodologies and technologies to support collaborative accessibility in the home. Winberg et al. [52] reviewed the collaboration between users with visual impairments and sighted individuals across different modalities. They set up an environment in which both types of users play a game. An auditory interface is provided to the candidates who have visual impairments whereas a visual interface is available for sighted users. The issues regarding the collaborative interface were observed and revised design principles were presented for the users with visual impairments.



A paper [53] presents the methodologies used by users with visual impairments while interacting with computer systems and describes the pros and cons. Based on these analyses, they presented recommendations for user interfaces of groupware and chatting applications designed for persons with visual impairments to enhance their usability interaction without losing their interest. Finally, a prototype with the name, Blind Internet Relay Chat (BIRC), was proposed and its advantages and limitations were discussed.

### 2.3. Limitations and Relevant Recommendations

Some of the systems that we reviewed have implemented basic techniques, such as a screen reader, sound alerts, and popup notifications etc., whereas others have advanced features, like speech-based input, voice alerts, and braille peripherals, etc., to facilitate the visually impaired in performing group activities for document writing, which included text content or mathematical expression. Table 1 presents the characteristics of the discussed systems based on the attributes chosen for comparative study and analysis.

**Table 1.** Comparison of assistive technologies and applications for the blind for document writing.

	Google Doc UI [9]	MS Word Add-In [36]	TalkMaths [39]	WMS/LAMBDA [41]	SRHVIPWMF [43]	MaWEn [44]
Application Platform	Web Application	Microsoft Word Add-in	Desktop Application	Desktop Framework	Web application	Local Machine Application
Application Objective	Collaborative Editing	Collaborative Writing	Mathematical Formulas Writing	Learning Mathematics and Science	Writing Math by Speech	Mathematical Collaborative Writing
Audience Considered	Persons who are blind	Persons who are blind	Persons who are blind	Persons who are blind	Persons who are blind	Persons who are blind and Sighted
Input Mechanism	Tab Keys, Keyboard	Keyboard, Mouse Right Click	Dictating, Select-and-Say	Vocal Synthesis, Braille Peripherals	Speech Input	Keyboard, Speech-Based Commands
Output Mechanism	Auditory via Screen Reader	Message Boxes, Auditory via Screen Reader	Auditory Speech	Auditory Speech	Auditory Speech	Braille Methods, Graphical Illustration
Awareness Mechanism	Warning Messages	Dialogue Boxes	Warning Messages	Sound alerts	Voice Alerts	Pop-up Alerts
Work Space	Multi-user and Shared	Single User	Single User	Single User	Single User	Group Work
Time/Space	Different Time/Different Place	Different Time/Same Places	Same Time/Same Place	Same Time/Same Place	Same Time/Same Place	Different Time/Different Place
Security and Privacy	User authentication, Data security.	Data Privacy	-	-	Basic security, User Authentication	Basic Security

Application platforms are related to whether an application is accessible to its users through a web portal, whether an installation required, or if it is an add-in that needs to integrate with an already installed application. The application objective states exactly what the application does to support its end user. When designing an application, the consideration of its audience is another important factor. It helps to select and design the best input/output approaches for its users instead of using standard mechanisms. An awareness mechanism should be implemented very efficiently to make an application interactive and responsive. Advance interactive components, such as speech-based input, assistive hardware devices, and voice alerts, are needed for blind users to interact with the application, which are not very common input/output methods [54]. The user's workspace describes that either an application is usable by only one user at a time or multiple users are allowed to work simultaneously. Some platforms use the approach where they are used at the same place and time, others differ since they can be used at different times and in different places, while some use a hybrid approach, i.e., a mixture of both [7]. When users are blind, security and privacy must be a property that is defined and handled.

Various approaches have been used to integrate user interactions and awareness in the system. This may include dialogue boxes, warning messages, sound beeps, speech alerts, text to speech, and so on. Text to speech, sound beeps, and speech alert techniques are very popular. However, the users have to listen to long speeches to obtain their required information [4], and if, by chance, a user misses a speech or is unable to grasp the information in a timely manner, he must repeat listening to the speech again. Systems must have a function to generate the maximum amount of information from limited speech and have some control over the speech, like stop, repeat, and skip item functions, and so on.

In mathematical writing applications, the major issue is editing an already written formula. There is some basic solutions proposed for this issue, like dictating the position of the content and then updating it [39] or going through the whole expression and then after reaching the particular area, updating the content. The solution to this problem might be the same as discussed earlier; the user must have control over the spoken content. Another limitation is the extensive structure of the mathematical expression that the user needs to speak, such as in the case of very long and complex expressions. Predictions could be added to speed up the process. Also, memory can be added to memorize frequently used expressions and to re-use them at a single call. The system should be interactive with the user to enhance the application's usability. For instance, the system should respond to every action undertaken by a visually impaired user. The response time should also be managed and be small so that the system can work efficiently and meet the requirements of the user.

When we work in a collaborative environment, the authentication, authorization, and user's personal information security becomes an important aspect of that application [55]. Some systems have privacy and security techniques implemented, but their implementation is at a basic level. More work is needed for security, especially when the user of that application is a blind person. Another approach that can be used to authenticate a user is their voice. As the user of the application is a blind person, the authentication can be completed through spoken words. Some systems have voice input features, but none of them have this feature.

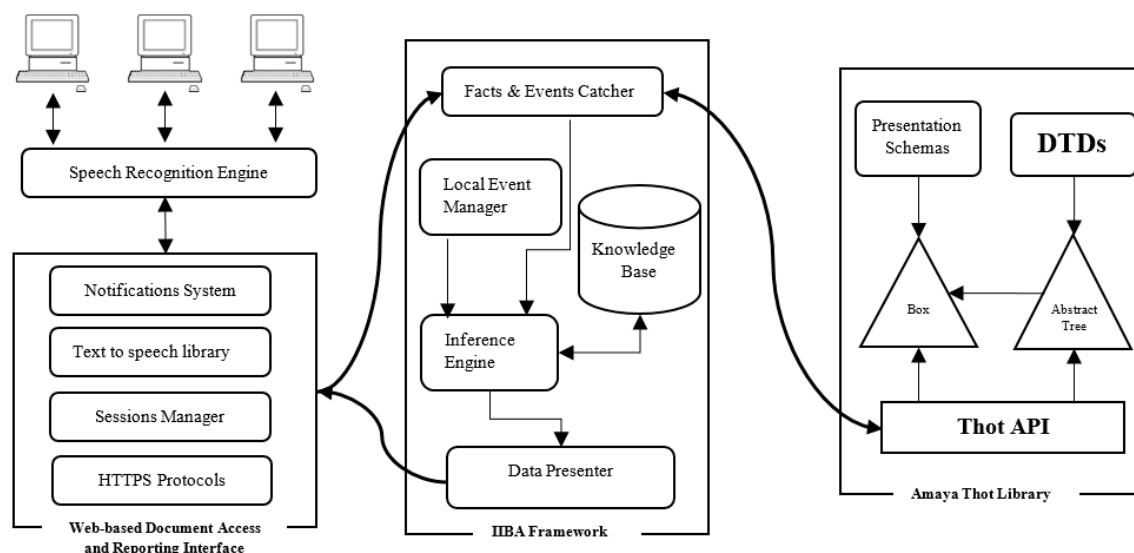
Multiple efforts are in progress to standardize educational software applications for visually impaired users and on the basis that some standards have been developed for the implementation and realization of these applications [56]. However, still, the doors are open for further research opportunities.

### 3. An Interactive Web Co-Authoring Platform for the Visually Impaired

A system, which allows its users to work jointly on a single goal/task even if they are located in different places, is called a collaboration software [11,57]. If the goal is to write a single shared document, then this is called co-authoring [58]. For better results, in a co-authoring system, the users must know about each update, change, and any present and past activity that has taken place in the shared environment [2]. The synchronization of these activities should be proper and the coordination between the users is an important factor. An individual should be able to stay up to date about the group activities; this is the main objective of a co-authoring system.

The comparison of factors described in Table 1 gave us support while designing the proposed system. We analyzed each factor with care and finalized the best options that support both types of users, those who can see and those who cannot. The proposed application is a web-based platform where the users who are visually impaired perform collaborative writing activities with the individuals who have sight. The users with sight give input through a standard keyboard and mouse whereas speech-based commands are available for persons who are blind. There are also shortcut keys available for interactions with the application for both types of users. While interacting with the system, it acknowledges each action performed in the shared environment through speech for its users who are visually impaired whereas the users with normal sight vision do not need any special function for this. Users with visual impairments listen to all the activities happening in the shared document whereas the popup notification and sound beeps are embedded for sighted users. The application

is accessible from different places at different timings, which give its users flexibility to work at any time while sitting at any place without any hurdle. It uses standard web technologies, such as Synchronized Multimedia Integration Language (SMIL), XML, and Hypertext Transfer Protocol Secure (HTTPS), for the development of this system. Amaya [59], an open source library, is used as the kernel. The proposed system, Web-Based Co-authoring Framework for Blind (WCFB), is a three-layered distributed model as shown in Figure 1.



**Figure 1.** Core architecture of the Web-based Co-authoring Framework for the Blind (WCFB).

We integrated the Thot libraries into the system and modified their existing functionalities according to our system's requirements where needed. It helped us to customize the basic library features and adapt them according to our specific user needs. We managed to design an intelligent framework, whose major role is to catch all the activities in the shared, and potentially rather distributed, cloud environment and suggest corresponding notifications and actions as well. We succeeded in designing a localized and distributed inference engine and managed the knowledge base that contained a list of rules. Whenever any activity occurs, all the facts against it are collected, the inference engine finds the best match rule from the knowledge base, and the corresponding action is formulated. We designed and developed web-based document access and reporting modules, which contain some major libraries, like the text to speech library, session manager, notification system, and so on, and integrated them in a managed way so that they can work concurrently within the application and do not create any conflicts. They are responsible for the delivery of access to the document, presentation of well-formatted awareness to its end users, and the management of the authorization against different elements of the system. In addition, we integrated the speech recognition engine into the system, which enables its user to work with the application through speech commands. We developed a trigger model that contains a list of transcribed phrases and corresponding actions against them. We developed a function whose responsibility is to identify the corrected transcribed phrase from the trigger model and propose the correct action against it. A detailed overview of each module of the system is given below in Figure 1.

### 3.1. Amaya's Thot Library

A document may contain different components in it, like headings, sections, pages, images, tables, etc. If the document is stored in a well-structured format, it is easy to manipulate it and track all the activities happening in each component of the document. Similarly, the participants were assigned different roles to work on a different section of the document and thus, they must know about any modification and editions if they occur in their assigned areas. So, the logical structure of



the documents also helps to log this kind of information. Amaya's Thot library was used to keep the documents in a well-structured way. It is an open source library (source code is easily accessible), which has several application program interfaces (APIs) available in its software architecture. They are easy to manage and their functionality can be enhanced with little modification of their existing functions [59]. Many document manipulation libraries are part of Amaya. By the separation of the presentation, structure, and content, the Amaya Thot library keeps documents in a structured way. The abstract tree data structure is used by Amaya's library, whose nodes are the headings, paragraphs, phrases, lists, and so on. The logical structure of the documents is maintained by a set of rules that are a part of the document type definitions (DTDs). To make structured documents, these rules, the attributes of the documents, and the elements of the documents are assembled. The presentation schema is composed of the views to present the documents. A mixture of some sets of rules to present a document to the user is called a view. The box tree is an intermediate structure that uses the abstract tree and the presentation schema from Thot, which presents the views. The outer environment of Amaya is communicated through the API provided by the Thot interface. Several hooks are available to send, receive, and manipulate documents' contents.

### 3.2. Intelligent Interface for the Blind Awareness (IIBA) Framework

IIBA is responsible for enhancing awareness functions for its users. The interface and kernel layer are bridged by this layer. The major components of IIBA are: Facts and events catcher, which captures every activity that happens in a co-authoring environment, like a session beginning or ending, opening a document, editing an object, the type of edition, the action performed, and so on. These facts are stored in a dedicated storage space in order to deliver them to a consumer application later on [5]. The local events manager manages all the events generated by local authors and applications. These events are managed in an order list based on the time they occurred in the form of a circular buffer. These events are later sent to other users working in a shared environment to give them awareness about the activities happening in the system. This gives them the visibility of what is going on, on the other side. The events are given as input to the inference engine (IE), which performs analyses using the rules available in the knowledge base and performs actions accordingly. First order predicate logic (without function symbols) [60] is used to write the rules, which consist of two parts, the premise and actions. The rules are well defined and tested very carefully so there are not any syntactical, semantically, and lexical errors. The following rule triggers if any of the authors open any document and notification of the visually impaired author is required of this action:

**If**

```

Author(section_1) = Y
Blind(Y) = "true"
Role(Y) = "writer" /* Y can write section_1 */
Session_Active(Y) = "true"
Author(section_1) = X
Role(X) = "annotator" /* X requested to see the update in section_1 */
Action(X) = "open_document"

```

**Then**

```

SendSpeechAlert(Y) <- "X opened the documents at CURRENT_TIME"

```

**EndRule**

To define a rule, we wrote predefined terminals in boldface. The constants and sequence of characters are enclosed in double quotes (" "). A variable symbol is represented by an identifier. Functions' and actions' names are presented in italics. Comments are enclosed by the symbols /\* \*/. The equal operator (=) is used to compare two items whereas the affectation operator (<-) works as an assignment operator. The semantics of the condition of the above rule is as follows:

1. *Author*(section\_1) = Y, defines the set of Y of coauthors of Section 1 of the document.
2. *Blind*(Y) = "true", checks whether author Y is visually impaired or not.
3. *Role*(Y) = "writer", checks that the role of the author is a writer
4. *Session\_Active*(Y) = "true", checks whether author Y is currently logged in or not.
5. *Action*(X) = "open\_document", defines whether coauthor X is opening the shared document.

Group Awareness by Speech Alerts: Whenever a member joins, edits, or updates the document, all others receive a speech alert, which notifies them about the update. This editing rule of the figure can be explained by highlighting that author X is working on the first section of the document. The role of author Y is that of an editor. Author X has started the session, and Author Y is a visually impaired author. Author Y is editing the figure, and a speech alert is sent to X notifying them that Y has edited the figure. The rule of the figure editing the speech alert triggers as:

**StartRule** "Figure Editing Speech Alert"

**If**

*Author*(First\_Section) = X  
*Blind*(X) = "true"  
*Role*(X) = "writer" /\* X can write section\_1 \*/  
*Session\_Start*(X) = "true"  
*Author*(First\_Section) = Y  
*Role*(Y) = "Editor" /\* Y can review and edit the section\_1 \*/  
*Action*(Y) = "edit\_figure"

**Then**

*SendSpeechAlert*(X) <- "Y edited the Figure just now"

**EndRule**

Interested Users Activity: When a user leaves a comment on a line in the document, the interested users receive a speech alert in case they are visually impaired. This alert is accompanied by a popup notification in case the other user is not a blind person. This rule defines that author X is adding a comment online number (LN) and a popup notification has been sent to Y about this:

**StartRule** "Adding a Comment Speech Alert"

**If**

*Author*(Second\_Section) = X  
*Role*(X) = "writer"  
*Author*(Second\_Section) = Y  
*Session\_Active*(Y) = "true"  
*Blind*(Y) = "false"  
*Role*(Y) = "reviewer" /\* Y can review and add comments in section\_1 \*/  
*Activity\_Interest*(Y) = X /\* Y is interested in activities of X \*/  
*Action*(X) = "Add a Comment on Line No. LN"

**Then**

*PopupAlert*(Y) <- "X added a comment on Line No LN"

**EndRule**

The inference engine automatically starts and keeps gathering information whenever an author logs in. The data presenter gathers information generated by the inference engine, makes it presentable, and transmits it to the document access and reporting interface.

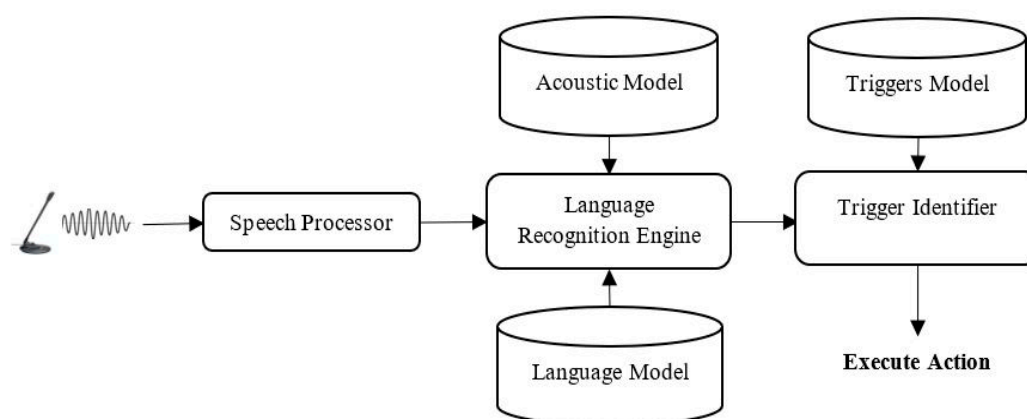
### 3.3. Web-Based Document Access and Reporting Interface

This is an interface layer whose responsibility is to present an interactive and intelligent interface to its user, as they work in a shared environment. It uses HTTPS protocols to send and receive information from the client to the server and the server to the client. HTTPS is a secure protocol and information moves over the net in an encrypted format, which ensures the security of the content.

The session manager keeps track of every login and logout and controls the access of users over the content within the application. It is a security check for the author's verification and protects against unauthorized access. The notification system provides updates to its users in the form of notifications. For the users with visual impairments, it gives notifications in the form of sounds, beeps, along with speech alerts whereas for sighted users, it uses popup and alert boxes. For example, an opening door sound along with the user's name communicates that an author has joined the session. A short beep along with the author's name is an intimation that someone has added comments to the document. Similarly, a long beep with the author's name informs that a new role has been assigned to an author. Whereas, for users with normal sight, all these notifications are sent through popup alerts. Text to speech libraries are embedded into the system to read the content displayed on the screen. They are specifically used by blind users and are activated on demand. To make them compatible with the application, hidden Meta information was added against each element/icon of the screen. This meta content is not visible to its user, but helps the text to speech synthesizer to speak the element/icons' description when a blind user performs navigations over the screen.

### 3.4. Speech Recognition Engine

The speech recognition engine captures the speech spoken by a user and converts it into a digital form from analog sound waves. The speech processor breaks down the speech into small pieces of meaningful words called phonemes. The acoustic model compares the generated phoneme with the standard pronunciation of the words available in the system's dictionary. The language model creates the sequences of the words by the use of grammar and the knowledge of statistical frequencies of words [61]. On the bases of these comparisons, the language recognition engine finds the most likely word sequence based on the probabilities and returns the best-matched phrase. Figure 2 presents the architecture of a speech recognition engine integrated into the WCFB framework.

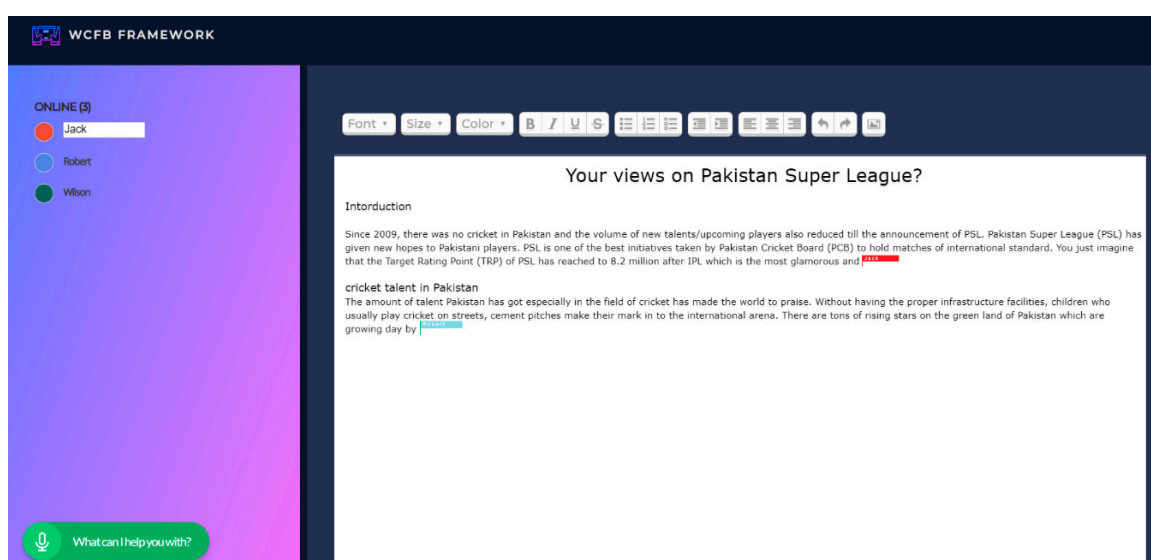


**Figure 2.** The architecture of the speech recognition engine.

The engine used for speech recognition in WCFB is Annyang [62]. It is an open source JavaScript library that enables the web applications to be controlled by voice command. It converts the speech into a transcribed phrase. The triggers model contains a list of action items mapped against a list of transcribed phrases. The trigger identifier compares the generated transcribed phrase with the available list of the action directory stored in the triggers model and performs the corresponding action. The common functions that can be performed through speech input are: Open and close the document; apply textual formatting, like font size, style, color, and so on. The user can ask for the status of other online members. Chat messenger can be opened by a speech command and then can easily be used to communicate using speech. The users can ask for updates made by other members.

#### 4. Evaluation of WCFB and Usability Testing

In WCFB, users require login credentials to interact with the platform. The users are managed by assigning different roles to them and the role may vary from time to time. Once a member acts as a writer, later on, he/she may have the responsibility of a reviewer. Audio alerts are available for individuals who are blind, whereas sighted users get visual notifications and popup alerts. Visually impaired users can learn the active participant's list by listening and can communicate with them within the application. They can communicate with each other via a chat messenger and thus have discussions when required. The users can join at any time and may work remotely as it is a web-based application. In Figure 3, three users are active in the system and two are working on the same document. Jack produces the "Introduction" section of the document and Robert works on Section 2 of the document. In addition to this, we added a function to give a voice command to the system. For this, a user must have a microphone attached and installed in his machine.



**Figure 3.** The interface of the WCFB application.

To evaluate the framework, experiments were conducted in which sighted and blind candidates participated and were asked to produce an article in a cooperative manner. Two different parameters were selected: The usability of the UI components and the acceptance level of the application by its users.

##### 4.1. Participants

To conduct the experiments in different environments, we visited different special educational institutes and engaged students who are visually impaired and sighted. The selection criteria were being visually impaired and sighted, to have good experience with using computer applications, and to have previously completed collaborative educational activities. Luckily, we were able to find 92 candidates, and all were used to working with a computer on a daily basis to perform educational activities. Out of the 92, 61 were visually impaired, whereas 31 were able to see. The candidates were divided into 11 groups and in each group, there were two teams. Each team contained sighted and visually impaired candidates, so that the blind users experienced working with normally sighted users and vice versa. This also verified the objective of the proposed system, which was to build a collaborative environment for sighted and visually impaired candidates. Table 2 presents the details of each group and team structured for the experiments. The participants were aged between 17 to 31 years of age and were of both genders. They were college and university level students.

**Table 2.** Participants' details of the WCFB testing and experiments.

		Team-1		Team-2	
		Sighted	Blind	Sighted	Blind
Group-1	Participants	3	2	2	2
	Gender (Male/Female)	2/1	2/0	1/1	2/0
	Age Range	19–22	20–25	20–25	20–23
Group-2	Participants	2	2	2	2
	Gender (Male/Female)	1/1	1/1	1/1	2/0
	Age Range	21–26	20–25	21–24	22–27
Group-3	Participants	3	1	2	2
	Gender (Male/Female)	3/0	1/0	2/0	1/1
	Age Range	17–26	23	23–28	24–29
Group-4	Participants	3	1	3	1
	Gender (Male/Female)	2/1	0/1	2/1	1/0
	Age Range	24–29	27	24–31	27
Group-5	Participants	3	1	2	2
	Gender (Male/Female)	1/2	0/1	1/1	0/2
	Age Range	23–26	30	25–29	25–28
Group-6	Participants	3	1	4	0
	Gender (Male/Female)	2/1	0/1	2/2	-
	Age Range	22–29	24	25–31	-
Group-7	Participants	4	1	3	1
	Gender (Male/Female)	2/2	1/0	1/2	0/1
	Age Range	17/22	19	19–26	19
Group-8	Participants	3	1	3	1
	Gender (Male/Female)	2/1	1/0	2/1	1/0
	Age Range	20–24	24	23–28	25
Group-9	Participants	3	2	3	1
	Gender (Male/Female)	2/1	1/1	2/1	0/1
	Age Range	22–28	23–26	22–27	28
Group-10	Participants	2	2	3	2
	Gender (Male/Female)	2/0	1/1	2/1	1/1
	Age Range	23–26	22–29	24–30	26–28
Group-11	Participants	2	2	3	1
	Gender (Male/Female)	1/1	1/1	1/2	1/0
	Age Range	25–27	24–27	22–27	26
<b>Total Participants</b>		<b>31</b>	<b>16</b>	<b>30</b>	<b>15</b>

#### 4.2. Preparation

Before giving the task to the users, a 60 min tutorial session was conducted with all the participants in which a detailed demonstration of the WCFB application and Google Doc UI was given to them. This was required as the WSFB application was new for all the users whereas the Google Doc UI had been used previously by a few of them. Thus, the demo gave the participants knowledge about



both the application's basic features and functions and gave them an understanding of how they would interact with the applications. The tutorial covered an introduction of the application's UI components available for both sets of users to make them familiar with the help module. They were given a demonstration to see how they could interact with the application and operate it. They were also told about the short cut keys available for both sets of users. During the experiment, no assistance was provided to any participant, and they were asked to use the help module in case of any difficulty or problem. Feedback was taken from all participants and was used to further enhance the usability and effectiveness of the WCFB.

#### 4.3. Activity Goals

There were 11 experimental sessions (one with each group) that were conducted to evaluate the system. In each experimental session, there were two activities performed. The candidates were asked to write articles in a collaborative environment on two different topics: (1) The role of the Pakistan Super League (PSL) in bringing cricket back to the country, and (2) the future of politics in the country. The two topics were selected because they are very common discussion points for the age group that participated in the experiment. Moreover, at the time of the experiment, the two topics were heavily discussed in daily news and on social media. For activity 1, topic 1 was selected and team 1 was asked to use an existing system to produce the piece, i.e., they used Google Docs UI for the Blind [9], whereas team 2 was provided with WCFB to produce their piece. For activity 2, team 1 was given the WCFB, whereas team 2 was asked to use Google Docs UI for the Blind to complete a write-up on topic 2. In this way, both the teams used both applications one after another and were able to give feedback about them. The test team was given an hour to complete one topic.

#### 4.4. Post-Experiment Questionnaire

After the completion of the experiments, a questionnaire was given to each user to provide feedback. The responses were answered on a rating scale, ranging from 1 to 10, where 1 meant the lowest and 10 meant the highest.

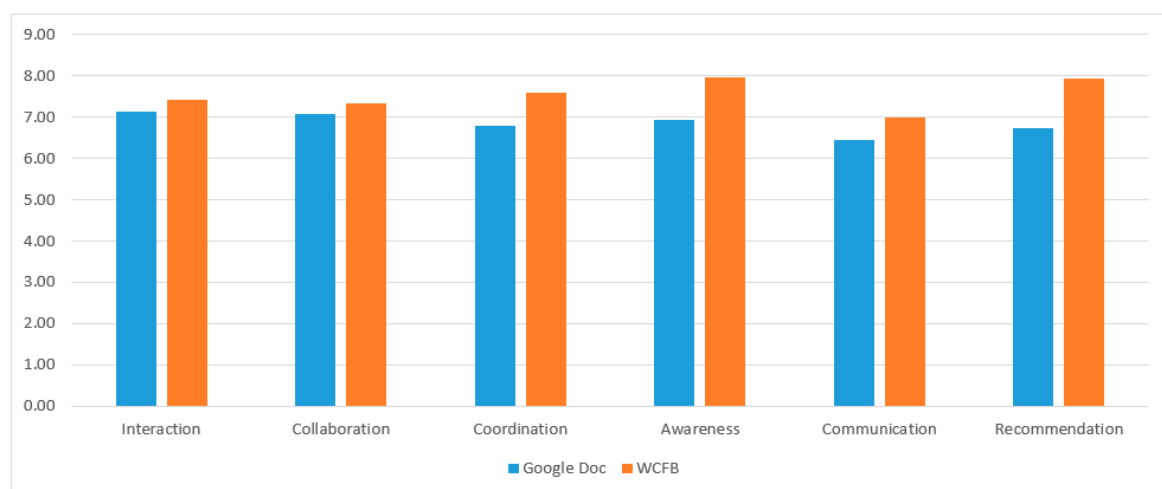
The questions were grouped into six categories: Interaction, collaboration, coordination, awareness, communication, and recommendation. In each group, there were at most two to three questions. Table 3 shows a list of questions alongside the results obtained from the participants. The numbers represent the averages of the scores given by each author against each question.

The results obtained from the questionnaire were qualitatively evaluated. The approach helped us analyze our solution to see whether it is useful for collaborative writing.

Figure 4 presents the graphical representation of the average scores obtained against the attributes used for grouping the questions. The results show that the participants rated the interaction and collaboration factors almost the same, but for coordination, awareness, and recommendation, they preferred WCFB over the Google Doc UI.

**Table 3.** Results of the questionnaire.

Q #.	Question	Google Doc	WCFB
Interaction			
1.	How flexible was the system in terms of usability needs?	7.13	7.34
2.	How easy was it for you to give input/commands to the system?	7.06	7.35
3.	How much control did you have over the application?	7.21	7.53
Collaboration			
4.	At what rate were the access policy, roles, permissions, and restrictions, effective while working in the shared document?	7	7.42
5.	How would you rate the collaborative feature provided by the application?	7.16	7.27
Coordination			
6.	At what level is the system able to maintain the integrity of the document?	6.44	7.59
7.	How would you rate the syncing functionality of the application in terms of editing and other changes?	7.14	7.56
Awareness			
8.	How much awareness does the application provide about the activities done as a group?	6.94	8.06
9.	How much awareness did you have about the change history of the shared document?	6.97	8.05
10.	How much awareness did you have about your co-authors' presence and absence?	6.86	7.81
Communication			
11.	How easy was it to communicate with other co-authors?	6.98	7.02
12.	What level of interaction with other co-authors has been provided by the system?	5.89	6.97
Recommendation			
13.	How would you rate your impressions while using this application?	5.98	7.97
14.	How much confidence have you gained after using this application?	6.86	7.84
15.	Would you like to recommend other fellows to use this application?	7.33	8.02

**Figure 4.** Averages scores of attributes based on the questionnaire.

## 5. Results and Discussion

The quality of the content was measured by experts, and all the teams wrote both articles very well. The experts were university/college professors who voluntarily completed the review. It was observed in the experiments that in most of the cases the team using WCFB completed their tasks earlier, as compared to the team using the alternative application. To obtain a good accuracy and better results, we used two different testing techniques to perform a comparison between the WCFB and Google Doc.

### 5.1. The *t*-Test Result

The *t*-test is the most commonly used statistical technique that compares the averages of two data sets measured in two different scenarios against a similar unit and ascertains whether they are significantly different from each other [63]. In other words, it tells whether the differences between the two data sets happened by chance or are significantly different.

Table 4 presents the *t*-test results of all experiments based on the completion time of the activities. The participants using Google Doc took an average of 52 minutes to complete the activities whereas when using WCFB they completed their goals in an average of 49 minutes. The *t*-test results show that the difference between the two applications is significant ( $t = 2.2234$ ,  $p = 0.03788 < 0.05$ ). This quantitative evaluation shows that the efficiency of the activity is better when using the WCFB platform.

**Table 4.** The *t*-test results for the completion time of activities.

	Avg. Time to Complete Topics		The <i>t</i> -test Result	
	Google Doc	WCFB	<i>t</i>	<i>p</i>
Group 1	51	46	2.2234 0.03788	
Group 2	52	54		
Group 3	56	48		
Group 4	57	50		
Group 5	49	52		
Group 6	48	51		
Group 7	49	46		
Group 8	57	51		
Group 9	54	51		
Group 10	48	49		
Group 11	58	46		
Overall Avg.	52	49		

The averages of the score given by each author against each category of the questionnaire were noted. Further, these averages were then used to perform the *t*-test evaluation to check whether WCFB was significantly different from Google Doc. Table 5 presents a summary of the *t*-test results obtained from the questionnaire. The results show that there is a significant difference ( $t = 15.305$ ,  $p = 0.00001 < 0.05$ ) in awareness. The users found WCFB better in terms of functionality for awareness, as compared to Google Doc. In the case of collaboration, the variance is not that high ( $t = 1.8472$ ,  $p = 0.06635 > 0.05$ ), which shows that the participants found both applications equally useful in terms of collaboration. Since there is a significant difference in all categories except collaboration, it can be concluded that WCFB is the more effective and impactful application, given the current context.

**Table 5.** Summary of *t*-test results against the questionnaire's categories.

Category	Avg. Scores		The <i>t</i> -Test Results	
	Google Doc	WCFB	<i>t</i>	<i>p</i>
Interaction	7.13	7.41	2.4799	0.01405
Collaboration	7.08	7.34	1.8472	0.06635
Coordination	6.79	7.58	6.4826	0.00001
Awareness	6.93	7.98	15.305	0.00001
Communication	6.44	7.0	5.5581	0.00001
Recommendation	6.73	7.95	12.503	0.00001

### 5.2. Wilcoxon Signed Rank Results

The Wilcoxon signed-rank [64] evaluation is a non-parametric statistical theory test used to evaluate two related samples, matched trials, or replicated measurements on a single sample to evaluate whether their inhabitants implied ranks differ, i.e., a paired difference test [65]. It may be utilized as a substitute for the paired Student's *t*-test, *t*-test for matched pairs, or even the *t*-test for dependent samples once the population cannot be assumed as being normally distributed [66]. Descriptive statistical results obtained from the post-experiment questionnaire are presented in Table 6. The descriptive statistics are not necessary to calculate the Z-value of the Wilcoxon signed-rank but helps with the interpretation of the data.

**Table 6.** Descriptive statistics of the Wilcoxon signed-rank against the questionnaire's categories.

Category	Avg. Scores		Std. Deviation		Min Avg. Score		Max Avg. Score	
	G. Doc	WCFB	G. Doc	WCFB	G. Doc	WCFB	G. Doc	WCFB
Interaction	7.14	7.41	0.828	0.669	5.00	6.00	9.00	8.67
Collaboration	7.08	7.34	1.108	0.828	5.00	6.00	9.00	9.00
Coordination	6.79	7.58	0.862	0.786	5.00	6.00	8.50	9.00
Awareness	6.93	7.98	0.473	0.455	6.00	7.00	8.00	9.00
Communication	6.44	7.0	0.784	0.565	5.00	6.00	8.00	8.00
Recommendation	6.73	7.95	0.794	0.492	4.67	7.00	8.33	9.00

Table 7 presents all the statistics ranks required in the calculation of Wilcoxon signed-ranks, Z, and *p* values. We can see from the table's legend that against interaction, 28 participants have a negative rank, i.e., they gave a higher score to Google Doc over WCFB, a 48 positives rank means the score of WCFB is more than that of Google Doc and 16 participants have given mean equal scores for both interfaces. Ranks obtained from the other categories are also available in Table 7.

**Table 7.** Ranks obtained against the questionnaire's categories.

		<b>N</b>	<b>Mean Rank</b>	<b>Sum of Ranks</b>
Interaction G. Doc - WCFB	Negative Ranks	28	36.57	1024.00
	Positive Ranks	48	39.63	1902.00
	Ties	16		
	Total	92		
Collaboration G. Doc - WCFB	Negative Ranks	33	38.24	1262.00
	Positive Ranks	48	42.90	2059.00
	Ties	11		
	Total	92		
Coordination G. Doc - WCFB	Negative Ranks	24	25.85	620.50
	Positive Ranks	58	47.97	2782.50
	Ties	10		
	Total	92		
Awareness G. Doc - WCFB	Negative Ranks	2	6.25	12.50
	Positive Ranks	83	43.89	3642.50
	Ties	7		
	Total	92		
Communication G. Doc - WCFB	Negative Ranks	19	32.21	612.00
	Positive Ranks	60	42.47	2548.00
	Ties	13		
	Total	92		
Recommendation G. Doc - WCFB	Negative Ranks	5	12.60	63.00
	Positive Ranks	80	44.90	3592.00
	Ties	7		
	Total	92		

Table 8 presents the  $Z$  and  $p$  values obtained after performing the Wilcoxon signed rank test over the questionnaire's scores. The results obtained from the Wilcoxon signed-rank test are almost the same as that obtained from the  $t$ -test analysis. In both tests, only the collaboration's  $p$ -value is greater than 0.05 whereas all the other factors have values less than 0.05. So, both the tests accept that WCFB has a more effective interface for its user as compared to Google Doc.

**Table 8.** Test statistics for the Wilcoxon signed ranks test.

<b>Category</b>	<b>Wilcoxon Signed-rank Results</b>	
	<b><math>Z</math></b>	<b><math>p</math></b>
Interaction	−2.278	0.023
Collaboration	−1.892	0.058
Coordination	−5.024	0.0001
Awareness	−7.971	0.0001
Communication	−4.790	0.0001
Recommendation	−7.741	0.0001



During the feedback session with the participants, users who were blind expressed their appreciation for the input method whereby they could use their voices to insert data. In WCFB, the collaborators had an option to review the document sections accessed by other co-authors and suggest modifications. The object/section on which the communication happens is called “work focus” and this activity is named as context-based focused communication. The user who wants to do a review of any sections of the document takes the initiative to start communication. The other co-author of those sections is notified about the potential communication and has an option to accept/reject the communication. Once the communication is established, it is highlighted in both users’ environments. Whenever a co-author performs any modification in the section, the concerned user is notified about that activity. The user with visual impairments gets speech-based notifications whereas for the sighted user, there are visual notifications. Similarly, in a shared document, there are interlinked components, for example, table, figures, formulas, and their respective legends. In addition, there might be a chance that these interlinked components are composed by different authors. As an example, one author is assigned to compose a table and the other has the responsibility to write up its legend. We called this property “work proximity”. To maintain consistency, when a user completes any modification in an interlink component, the concerned author is notified about the modification so that he can update his assigned section/component accordingly. Unfortunately, both the functionalities are not available in Google Doc UI. Also, WCFB includes an embedded chat messenger, which makes communication between the authors strong. On the other hand, Google Doc only allows users to add comments or send an email for them to communicate with each other. The co-authors must go through the whole document to review the comments. The recommendation of the system is high when compared with the Google Doc UI application. It improves the confidence level of the users according to the kind of participants.

The users also proposed some suggestions and improvements during the feedback. It is observed that, at some time, there is a burst of audio alerts because of multiple users working on a shared document at the same time. The system includes a feature that can log all activities so that the user can listen to them later. The users liked the environment as it required minimum external assistance and the assignment of different roles enhanced their exposure. Privacy is another factor, participants asked for. Besides that, the results obtained from the experiments confirm that the application is acceptable in the community that it is intended for. The feedback proves that the system develops the interest of its user and enhances his writing skills. The overall outcome seems very promising, and the participants’ suggestions encourage us to make it more workable and to add more features and components to the application. We conclude that a web co-authoring platform for visually impaired users can promote enthusiasm and interest for group activities.

## 6. Conclusions and Future Work

The paper summarizes recent developments of document building tools and applications specifically designed for users with visual impairments, who have very limited special functions that restrict them from fully utilizing all the available features of an application. The proposed platform is particularly designed for visually impaired individuals to enhance group awareness among blind collaborators, informing them about all actions and activities happening in the shared environment, and allowing them to interact with the application through voice and speech commands.

The application’s performance was evaluated through multiple experiments, and a questionnaire was disseminated to the participants to gauge their experience and the application’s effectiveness. The responses show that the results are very promising and further research would be fruitful. For the future, we plan to extend our interface to handle even more complex objects, such as multimedia contents, including figures, diagrams, images, and so on. Our next goal is to implement this framework in a commercial context and to evaluate its usability with other differently-abled groups.

**Author Contributions:** Conceptualization, M.A.; methodology, M.M.W.; software, M.M.W.; validation, M.F.; formal analysis, M.F. and M.M.W.; investigation, M.M.W.; writing—original draft preparation, M.M.W.; writing—review and editing, M.M.W.; visualization, M.F.; supervision, M.A.; project administration, M.A.; funding acquisition, M.A.

**Funding:** This research is funded by the National Research Program for Universities (NRPU), Higher Education Commission (HEC), Islamabad, Pakistan, grant number 20- 9649/Punjab/NRPU/R&D/HEC/2017-18.”.

**Acknowledgments:** We are grateful for the staff who planned and implemented the applications and our partners, particularly to Dr. Talha Younas, department of electrical engineering, COMSATS university Islamabad, Sahiwal Campus, Pakistan, for useful discussions and feedback that helped us in designing the system.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. WHO | Vision Impairment and Blindness. Available online: <http://www.who.int/mediacentre/factsheets/fs282/en/> (accessed on 1 April 2018).
2. Kalra, N.; Lauwers, T.; Dewey, D.; Stepleton, T.; Dias, M.B. Iterative design of a Braille writing tutor to combat illiteracy. In Proceedings of the 2007 International Conference on Information and Communication Technologies and Development, Bangalore, India, 15–16 December 2007; pp. 1–9.
3. Reena, F.S.; Geetanjali, S.W. Speech Recognition and Synthesis Tool: Assistive Technology for Physically Disabled Persons. *Int. J. Comput. Sci. Telecommun.* **2012**, *3*, 86–91.
4. Muhammad, A.; Ahmad, W.; Tooba, M.; Anwar, S. Assistive Technology for Disabled Persons. In Proceedings of the Proceedings of the 2015 International Conference on Recent Advances in Computer Systems, Hail, Saudi Arabia, 30 November 2015; pp. 74–80.
5. Decouchant, D.; Escalada-Imaz, G.; Martinez Enriquez, A.M.; Mendoza, S.; Muhammad, A. Contextual awareness based communication and coauthoring proximity in the internet. *Expert Syst. Appl.* **2009**, *36*, 8391–8406. [CrossRef]
6. Rodríguez, G.; Soria, A.; Campo, M. *Advances in New Technologies, Interactive Interfaces and Communicability*; Springer: Berlin, Germany, 2012; pp. 140–150.
7. Schoeberlein, J.G.; Wang, Y. Groupware accessibility for persons with disabilities. In *Lecture Notes in Computer Science*; Stephanidis, C., Ed.; Springer: Berlin, Germany, 2009; Volume 5616, pp. 404–413.
8. Schoeberlein, J.G.; Wang, Y. Providing an accessible track changes feature for persons who are blind. In *Universal Access in Human-Computer Interaction compilation of Lecture Notes in Computer Science*; Stephanidis, C., Antona, M., Eds.; Springer: Berlin, Germany, 2013; Volume 8011, pp. 389–398.
9. Mori, G.; Buzzi, M.; Buzzi, M.; Leporini, B.; Penichet, V.R. Making “Google Docs” User Interface More Accessible for Blind People. In *Advances in New Technologies, Interactive Interfaces, and Communicability SE-4*; Cipolla Ficarra, F.V., de Castro Lozano, C., Pérez Jiménez, M., Nicol, E., Kratky, A., Cipolla-Ficarra, M., Eds.; Springer: Berlin, Germany, 2011; Volume 6616, pp. 20–29.
10. Mori, G.; Buzzi, M.C.; Buzzi, M.; Leporini, B.; Victor, M.R. Collaborative Editing for All: the Google Docs Example. *Univ. Access Hum.-Comput. Interact. Appl. Serv.* **2011**, *6768*, 165–174.
11. Claudia, M.; Buzzi, M.; Giulio Mori, B.L. Designing E-Learning Collaborative Tools for Blind People. In *E-Learning-Long-Distance and Lifelong Perspectives*; Elvis, P., Adilson, G., Sergio, K., Adilson, G., Eds.; InTech: Rijeka, Croatia, 2012.
12. Buzzi, M.C.; Mori, G.; Buzzi, M.; Barbara, L. A Modified Google Docs UI accessible via screen reader. In Proceedings of the ECSCW 2011: The 12th European Conference on Computer Supported Cooperative Work, Aarhus, Denmark, 24–28 September 2011.
13. Ellis, C.A.; Gibbs, S.J.; Rein, G. Groupware: Some Issues and Experiences. *Commun. ACM* **1991**, *34*, 39–58. [CrossRef]
14. Sanchez, J.; Baloian, N. Issues in implementing awareness in collaborative software for blind people. In Proceedings of the 10th international conference on Computers Helping People with Special Needs, Linz, Austria, 11–13 July 2006; pp. 1318–1325.
15. Waqar, M.; Muhammad, A.; Waheed, T.; Enriquez, M. A Group Awareness System for Blinds in Web Co-Authoring Environment. In Proceedings of the 2015 International Conference on Recent Advances in Computer Systems, Hail, Saudi Arabia, 30 November 2015; pp. 88–93.

16. Bower, M. Synchronous collaboration competencies in web-conferencing environments—Their impact on the learning process. *Distance Educ.* **2011**, *32*, 63–83. [[CrossRef](#)]
17. Begole, J.; Struble, C.A.; Shaffer, C.A.; Smith, R.B. Transparent sharing of Java applets. In Proceedings of the 10th annual ACM symposium on User interface software and technology, Banff, AL, Canada, 14–17 October 1997; pp. 55–64.
18. Baecker, R.M.; Nastos, D.; Posner, I.R.; Mawby, K.L. The user-centered iterative design of collaborative writing software. In Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems, Amsterdam, The Netherlands, 24–29 April 1993; pp. 399–405.
19. Yun Yang; Chengzheng Sun; Yanchun Zhang; Xiaohua Jia Real time cooperative editing on the Internet. *IEEE Internet Comput.* **2000**, *4*, 18–25. [[CrossRef](#)]
20. Locasto, M.; Hulme, M.; Gladysiweicz, R.; Tracy, J.; Wolz, U. Clay: Synchronous Collaborative Interactive Environment. *J. Comput. Sci. Coll.* **2002**, *17*, 278–281.
21. Gutwin, C.; Greenberg, S. A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Comput. Support. Coop. Work* **2002**, *11*, 411–446. [[CrossRef](#)]
22. Kirsch-Pinheiro, M.; Valdeni de Lima, J.; Borges, M.R.S. A framework for awareness support in groupware systems. *Comput. Ind.* **2003**, *52*, 47–57. [[CrossRef](#)]
23. Chen, Q.Z.; Huajiang, Z.; Ying, K. The XML-based collaborative documents model design. In Proceedings of the 8th International Conference on Computer Supported Cooperative Work in Design, Xiamen, China, 26–28 May 2004; pp. 24–28.
24. Kei, S.H.; Hong, V.L.; Wai, L. A collaborative word processing system using a CORBA-based workflow framework. In Proceedings of the 3rd International Symposium on Distributed Objects and Applications, Rome, Italy, 17–20 September 2001; pp. 176–185.
25. Weber, M.; Partsch, G.; Höck, S.; Schneider, G.; Scheller-Houy, A.; Schweitzer, J. Integrating synchronous multimedia collaboration into workflow management. In Proceedings of the Proceedings of the international ACM SIGGROUP conference on Supporting group work: The integration challenge the integration challenge—GROUP '97, Phoenix, AR, USA, 16–19 November 1997; pp. 281–290.
26. Hodel, T.B.; Gall, H.; Dittrich, K.R. Dynamic collaborative business processes within documents. In Proceedings of the 22nd annual international conference on Design of communication The engineering of quality documentation—SIGDOC '04, Memphis, TN, USA, 10–13 October 2004; pp. 97–103.
27. Joeris, G. Cooperative and integrated workflow and document management for engineering applications. In Proceedings of the Database and Expert Systems Applications. 8th International Conference, DEXA '97, Toulouse, France, 1–2 September 1997; pp. 68–73.
28. Muhammad, A.; Martinez Enriquez, A.M.; Escalada-Imaz, G. Collaborative Assistant to Handle MathML Expressions. In Proceedings of the International Conference on Intelligent Computer Mathematics, Grand Bend, ON, Canada, 6–12 July 2009; pp. 453–459.
29. Meliones, A.; Filios, C. BlindHelper: A Pedestrian Navigation System for Blinds and Visually Impaired. In Proceedings of the 9th ACM International Conference on Pervasive Technologies Related to Assistive Environments—PETRA '16, Corfu, Greece, 29 June–1 July 2016; pp. 1–4.
30. Farzana, J.; Muhammad, A.; Martinez-Enriquez, A.M.; Afraz, Z.S.; Talha, W. Speech Based Shopping Assistance for the Blind. In Proceedings of the 6th Mexican Conference on Pattern Recognition, Cancun, Mexico, 25–28 June 2014; pp. 361–370.
31. Tanveer, M.S.R.; Hashem, M.M.A.; Hossain, M.K. Android assistant EyeMate for blind and blind tracker. In Proceedings of the 2015 18th International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 21–23 December 2015; pp. 266–271.
32. Bigham, J.P.; Aller, M.B.; Brudvik, J.T.; Leung, J.O.; Yazzolino, L.A.; Ladner, R.E. Inspiring blind high school students to pursue computer science with instant messaging chatbots. *ACM SIGCSE Bull.* **2008**, *40*, 449. [[CrossRef](#)]
33. Jia, S.; Lin, W.; Wang, K.; Takase, K. Network Distributed Multi-Functional Robotic System Supporting the Elderly and Disabled People. *J. Intell. Robot. Syst.* **2006**, *45*, 53–76. [[CrossRef](#)]
34. Google Docs. Available online: <https://www.google.com/docs/about/> (accessed on 20 January 2019).
35. W3C. Accessible Rich Internet Applications (WAI-ARIA) 1.0. Available online: <http://www.w3.org/TR/wai-aria/> (accessed on 25 January 2019).

36. Schoeberlein, J.G.; Wang, Y. Usability Evaluation of an Accessible Collaborative Writing Prototype for Blind Users. *J. Usability Stud.* **2014**, *10*, 26–45.
37. Schoeberlein, J.G.; Wang, Y. Accessible Collaborative Writing for Persons Who Are Blind: A Usability Study. In Proceedings of the 14th International ACM SIGACCESS conference on Computers and accessibility, Boulder, CO, USA, 22–24 October 2012; pp. 267–268.
38. Schoeberlein, J.G.; Wang, Y. Improving collaborative writing for the blind. *ACM SIGACCESS Access. Comput.* **2013**, *107*, 11–16. [\[CrossRef\]](#)
39. Wigmore, A.M.; Pflügel, E.; Hunter, G.J.A.; Denholm-Price, J.; Colbert, M. Talkmaths better! Evaluating and improving an intelligent interface for creating and editing mathematical text. In Proceedings of the 2010 Sixth International Conference on Intelligent Environments, Kuala Lumpur, Malaysia, 19–21 July 2010; pp. 307–310.
40. Wigmore, A.; Hunter, G.; Pflügel, E.; Denholm-Price, J.; Binelli, V. Using Automatic Speech Recognition to Dictate Mathematical Expressions: The Development of the “TalkMaths” Application at Kingston University. *J. Comput. Math. Sci. Teach.* **2009**, *28*, 177–189.
41. Bernareggi, C.; Brigatti, V. Writing mathematics by speech: A case study for visually impaired. In *Lecture Notes in Computer Science*; Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A., Eds.; Springer: Berlin, Germany, 2008; pp. 879–882.
42. Schweikhardt, W.; Bernareggi, C.; Jessel, N.; Encelle, B.; Gut, M. LAMBDA: A European System to Access Mathematics with Braille and Audio Synthesis. In *Lecture Notes in Computer Science*; Springer: Berlin, Germany, 2006; Volume 4061, pp. 1223–1230.
43. Hanakovi, T.; Nagy, M. Speech Recognition Helps Visually Impaired People Writing Mathematical Formulas. In *Lecture Notes in Computer Science*; Springer: Berlin, Germany, 2006; Volume 4061, pp. 1231–1234.
44. Archambault, D.; Stöger, B.; Batusic, M.; Fahrengruber, C.; Miesenberger, K. A software model to support collaborative mathematical work between braille and sighted users. In Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility, Tempe, AR, USA, 15–17 October 2007; pp. 115–122.
45. Archambault, D.; Fitzpatrick, D.; Gupta, G.; Karshmer, A.I.; Miesenberger, K.; Pontelli, E. Towards a Universal Maths Conversion Library. In Proceedings of the Computers Helping People with Special Needs, Paris, France, 1–9 July 2004.
46. Metatla, O.; Bryan-Kinns, N.; Stockman, T.; Martin, F. Supporting Cross-Modal Collaboration in the Workplace. In Proceedings of the 26th BCS Conference on Human Computer Interaction, Birmingham, UK, 12–14 September 2012; pp. 109–118.
47. Metatla, O.; Tony, N.B. Interactive hierarchy-based auditory displays for accessing and manipulating relational diagrams. *J. Multimodal User Interfaces* **2012**, *5*, 111–112. [\[CrossRef\]](#)
48. McGookin, D.; Brewster, S. An initial investigation into non-visual computer supported collaboration. *CHI '07 Ext. Abstr. Hum. factors Comput. Syst.* **2007**, 2573.
49. McGookin, D.K.; Brewster, S.A. Graph Builder: Constructing Non-visual Visualizations. In *People and Computers XX — Engage*; Springer: London, UK, 2010; pp. 263–278.
50. Moustakas, K.; Nikolakis, G.; Tzovaras, D.; Deville, B.; Bologna, G.; Marras, I.; Pavlek, J. Multimodal tools and interfaces for the intercommunication between visually impaired and “deaf and mute” people. In Proceedings of the eINTERFACE 2006 Workshop, Dubrovnik, Croatia, 17 July–11 August 2006.
51. Branham, S.M.; Kane, S.K. Collaborative Accessibility: How Blind and Sighted Companions Co-Create Accessible Home Spaces. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, Seoul, Korea, 18–23 April 2015; pp. 2373–2382.
52. Winberg, F.; Bowers, J. Assembling the Senses: Towards the Design of Cooperative Interfaces for Visually Impaired Users. In Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, Chicago, IL, USA, 6–10 November 2004; pp. 332–341.
53. Keil-Slawik, R.; Claassen, B.G.; Ploh, F.; Reimann, C.; Moustakas, K.; Tzovaras, D.; Marras, I.; Pavlek, J.; Road, M.E.; Road, M.E.; et al. Pragmatic Solutions for Better Integration of the Visually Impaired in Virtual Communities. In Proceedings of the international ACM SIGGROUP conference on Supporting group work, Phoenix, AR, USA, 14–17 November 1999; pp. 258–266.

54. Karpov, A.; Ronzhin, A. A Universal Assistive Technology with Multimodal Input and Multimedia Output Interfaces. In *Universal Access in Human-Computer Interaction. Design and Development Methods for Universal Access*; Stephanidis, C., Antona, M., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 369–378.
55. Wentz, B.; Pham, D.; Tressler, K. Exploring the accessibility of banking and finance systems for blind users. *First Monday* **2017**, *22*. [[CrossRef](#)]
56. Bocconi, S.; Dini, S.; Ferlino, L.; Martinoli, C.; Ott, M. ICT Educational Tools and Visually Impaired Students: Different Answers to Different Accessibility Needs. In Proceedings of the International Conference on Universal Access in Human-Computer Interaction, Beijing, China, 22–27 July 2007; pp. 491–500.
57. Köhlmann, W. Identifying Barriers to Collaborative Learning for the Blind. In Proceedings of the International Conference on Computers Helping People with Special Needs, Linz, Austria, 11–13 July 2012; pp. 84–91.
58. Tucker, B.P.; Parker, L.D.; Merchant, K.A. With a little help from our friends: An empirical investigation of co-authoring in accounting research. *Br. Account. Rev.* **2016**, *48*, 185–205. [[CrossRef](#)]
59. Guetari, R.; Quint, V.; Vatton, I. Amaya: An Authroing Tool for Web. In Proceedings of the MCSEAI'98 Maghrebian Conference on Software Engineering and Artificial Intelligence, Tunis, Tunisia, 8–10 December 1998; pp. 8–10.
60. Feigenbaum, E.; McCorduck, P. *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World*; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 1983.
61. Karpagavalli, S.; Chandra, E. A Review on Automatic Speech Recognition Architecture and Approaches. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2016**, *9*, 393–404.
62. Ater, T. Annyang! SpeechRecognition that Just Works. Available online: <https://www.talater.com/annyang/> (accessed on 10 January 2019).
63. Raju, T.N.K. William Sealy Gosset and William A. Silverman: Two “Students” of Science. *Pediatrics* **2005**, *116*, 732–735. [[CrossRef](#)]
64. Wilcoxon, F. Individual Comparisons by Ranking Methods. *Biom. Bull.* **1945**, *1*, 80. [[CrossRef](#)]
65. Kerby, D.S. The Simple Difference Formula: An Approach to Teaching Nonparametric Correlation. *Compr. Psychol.* **2014**, *3*. [[CrossRef](#)]
66. Derrick, B.; White, P. Comparing two samples from an individual Likert question. *Int. J. Math. Stat.* **2017**, *18*, 1–13.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).