BellChat: An Inclusive Web Application for Messaging

Gleiston Guerrero-Ulloa, Víctor Romero-Castro, Janer Torrales-Peralta, Tyrone Tocta-Bonilla, and Orlando Erazo

Abstract The goal of the web is universal access, but minority groups of web users, such as people with disabilities and elderly people, are limited in their ability to communicate through chat applications. These types of applications include social networking, which are especially used in communication among their users. However, social networks such as Facebook (Messenger), Twitter and WhatsApp are not adapted for these groups, and therefore, they do not fully meet the needs for which they were created. Moreover, for people with disabilities to use web-based messaging applications, they are required to use third-party tools, such as screen readers. For these reasons, this paper presents BellChat, a responsive web application for communication between everybody, adapted for people with disabilities, especially for those ones with visual or hearing disabilities. It was developed following the Scrum development framework with adaptations to very small groups, where the roles were modified to be according to the project team. In addition, developers worked remotely and meetings were held in person twice a week. BellChat is the only application adapted to convert text messages into speech, or speech messages into text, depending on the disability of the user who is connected. The evaluation results confirm the acceptance of the application by users.

G. Guerrero-Ulloa (✉) · V. Romero-Castro · J. Torrales-Peralta · T. Tocta-Bonilla · O. Erazo
Universidad Técnica Estatal de Quevedo, Quevedo 120301, Los Ríos, Ecuador
e-mail: gguerrero@uteq.edu.ec

V. Romero-Castro
e-mail: victor.romero2016@uteq.edu.ec

J. Torrales-Peralta
e-mail: janer.torrales2016@uteq.edu.ec

T. Tocta-Bonilla
e-mail: tyrone.tocta2016@uteq.edu.ec

O. Erazo
e-mail: oerazo@uteq.edu.ec

G. Guerrero-Ulloa
University of Granada, 18071 Granada, Spain

X.-S. Yang et al. (eds.), *Proceedings of Eighth International Congress on Information and Communication Technology*, Lecture Notes in Networks and Systems 695,
https://doi.org/10.1007/978-981-99-3043-2_35

Keywords Visual impairment · Hearing impairment · Web accessibility · Text-to-speech · Speech-to-text

1 Introduction

Although inclusivity goes beyond accessibility, its basis is accessibility. An inclusive web application must first be an accessible web application. This means that an application should be thought of from the design to the building and deployment as an application that is accessible to all people without exception [1]. To achieve this goal, the Web Accessibility Initiative (WAI) of the World Wide Web Consortium (W3C) has been established to raise awareness of universal accessibility [2]. The WAI provides developers with guidelines that can help make web pages widely accessible [3]. Considering that websites are designed to serve different purposes such as information, entertainment, advertising, to name a few, and that to achieve this goal they present a wide range of information to meet user needs and they must be accessible to everybody. However, most of these websites do not comply with the web accessibility standards designed by the W3C, making it difficult for part of the population to access to web content [4].

The web is the place where everybody should feel their full right to equality, regardless of their conditions or disabilities [5]. However, what happens with people with disabilities? It is very common to find websites that are not adapted for people with disabilities; for example, they do not have their own content reader, or they do not have a voice command interface. Likewise, not all websites take care of people with hearing disabilities, i.e. the voice files generally do not have the transcription of the content into text to allow for people with hearing disabilities understanding the content [6].

Given these disadvantages, people with visual or hearing impairments face the difficult task of using communication resources. Visually impaired people lessen this difficulty by using screen readers provided by third parties [7], while tools to reduce the difficulty faced by the hearing-impaired people are poorly suited for instant transcription of voice messages by downloading the message and uploading it to the applications for transcription. Some of these tools and the most popular ones can be found at González [8]. Two authors of the present work share with hearing-impaired people, who have shared with them that, when they receive voice messages, they must ask for help from third parties to know the content. This situation is also affirmed by Pereira [9]. In addition, this group of people are joined by older adults who, due to their condition are diminished in their abilities, and therefore do not enjoy all the benefits offered by the web applications for communication [6].

Communication between human beings is what makes people live in society. Currently around 3.4 billion users actively use social media platforms daily for an average of 2.5 h [10]. Social media have become an effective means of communication between people, including for companies with their customers. However, the lack

of inclusive platforms may make people with disabilities feel excluded. As a contribution to achieve the inclusion of this group of people, this paper presents BellChat, a responsive web application that ensures accessibility to the greatest number of people, regardless of their conditions (visual, hearing, physical). Its virtual assistant ensures interaction between the user and the application. It enables the actions allowed on each page to be performed by voice commands; it also allows any user to send messages in any format to people with or without disabilities. In addition, to reach as many people as possible, keyboard commands have been implemented for those who have problems with the mouse and for people who may have problems with the pronunciation of voice commands [11].

BellChat implements the W3C standards and complies with the success criteria to make web pages accessible to as many people as possible [1]. Based on these standards, the communication between people with disabilities and without disabilities is facilitated and communication accessibility problems are reduced. Other communication websites comply with certain W3C standards, but they do not achieve the objective of maximizing the number of people because they do not provide ease of use for people with disabilities. Thus, this document details the development and evaluation of BellChat.

2 Related Work

In digital libraries, such as Web of Science (WOS), ACM, IEEE, Scopus and Elsevier, few papers have been found on the development of accessible web applications. However, there are some articles that deal with the evaluation of web accessibility in public interest websites.

Among the works that present the development of accessible web applications, we can find Web-ALAP, a web-based application for writing mathematical documents in LaTeX developed by Arooj et al. [12]. Web-ALAP supports users with low vision through assistive functions and the manifestation of error indications by voice. To support users with the same disability, Lee et al. [13], introduce TableView, which focuses on solving the problem of users with low vision who have difficulty using the on-screen magnifier. TableView extracts content and information from the page and presents it in a more compact form to make the most of the expanded space. Despite their contribution, these works are focussed on a specific problem: people with low vision.

Likewise, the works that have been concerned with people with profound deafness problems, Lyall et al. [14], developed a smartphone application that recognized six sentences dictated by doctors for post-operative patients. These messages, delivered by voice by doctors, were converted to text to be read by the patients. The results shown in this work, with a clear extension, can be used for telephone conversations and even in face-to-face conversations between people without disabilities and with hearing loss.

Along the same lines, we find the work of Shadiev et al. [15], in which they apply speech-to-text technology in a virtual classroom. Their results were a motivation to propose BellChat. Although Shadiev et al. [15] apply it directly, BellChat does it in a deferred way, in text messages and speech-to-text over voice files. Moreover, a mobile application that enables communication between hearing impaired and non-disabled people (or those who do not understand sign language) was presented by Ali et al. [16], which via Bluetooth allow for exchanging information over short distances.

Alsaif et al. [17] present a very useful application for people with speech problems. It predicts through statistics the words that could be pronounced in a conversation. In addition to converting text-to-speech, the system provides different categories of frequently used phrases that are labelled with a representative image for ease of use. The user can also add images to the system and record the human voice or use the automatic text-to-speech synthesizer. The limitation of this application is that it is smartphone-only and converts speech-to-text and text-to-speech using only Arabic language.

All the papers read before and while BellChat was being designed present solutions that help people with specific disabilities. This is the reason why the authors of this paper present BellChat as an inclusive application as a means of communication between the greatest number of people, supporting the right of equality for which the web was created.

3 Proposal

It can be said that the web is one of the ways to facilitate communication between users and access to services. In this way, it alleviates the disadvantages people face when accessing services in the real world, especially people with visual or hearing impairments [18]. Given this landscape, BellChat is a smart web application capable of converting text-to-speech and speech-to-text depending on the disability registered in the user's profile. Its responsiveness gives it the feature of being easily accessed from a smartphone. This quality allows for reaching a much larger group of users, thus integrating into society those users who have been relegated by their situation. Moreover, the text-to-speech and the speech-to-text conversion help not only people with any of the disabilities; they also help people with full capabilities to perform other tasks while learning about the messages they receive or while sending messages to their contacts [19]. In addition, BellChat has an innovative design to ensure its goal of being accessible to everyone.

BellChat eliminates the most impactful problems for people with hearing impairment such as those mentioned by Pascual et al. [20]. In addition, it includes visually impaired people, allowing them to interact seamlessly with other users regardless of their abilities [21]. Also, BellChat can adapt to any device (smart phones, laptops, desktop computers) with Internet access in a friendly way.

3.1 Requirements

In the development of BellChat, functional requirements have been considered to try to include as many people as possible and non-functional requirements to provide all users with the security necessary in today's web applications.

Functional Requirements. In the development of BellChat, the following requirements were considered:

- *User identification*: To become a user of the application, a person must register as it. Among the information that the user must register is whether s/he has a disability and which one. BellChat will allow the user to enter his/her username and password, by voice or typing on a keyboard.
- *Self-programmable system*: Once the user has logged into the application, the respective functions will be self-programmed in its configuration, for example: the inbox will be prepared to present the user with the textual content of the messages (speech-to-text), or play the audio content of the messages (text-to-speech), the user's contact search format, voice management control, among others, or simply maintain appropriate options according to the user's profile.

Requirements for visually impaired persons:

- *Voice assistant*: The voice assistant must allow interacting with a browser, being always ready to solve user requests.
- *For each page (screens)*, a set of commands must be defined to help the user to perform the respective actions (in this case, the executor of the voice commands starts its work when Lili is invoked).

Requirements for hearing impaired persons:

- *Voice message to text converter*: If a hearing-impaired user receives an audio message, the application identifies the message format and converts it to text, letting the recipient read the message so that he/she is aware of its content.
- *Writing messages*: The user will be able to write in text format the message he/she wants to send, no matter to whom (recipient capabilities) he/she is going to send it.

Requirements for people with motor disabilities:

- *Voice assistant (Lili)*: The assistant should allow users to interact with a browser, always waiting for user requests. Each time a page is loaded in the browser, the respective commands can be invoked by voice to perform the permitted actions.
- *Sending messages*: The user may decide to send a message in text or in audio format, according to his/her capabilities, without considering to whom (recipient capabilities) he/she is going to send the message.

Non-functional Requirements. BellChat eliminates the most impactful problems for people with hearing impairment such as those mentioned by Pascual et al. [20]. In addition, it includes visually impaired people, allowing them to interact seamlessly

with other users regardless of their abilities. In addition, as the largest number of people access the Internet from their smartphone [22], BellChat should have no restrictions on the type of device from which it can be accessed and should do so in a friendly manner.

4 Methodology

BellChat was developed following the Scrum development framework, which is considered the most widely used in agile software development. The first task after defining the objectives of this project was the assignment of roles [23]. The first author played the role of product owner, the third author played the role of Scrum master and the second and fourth authors played the role of developers (development team). As the team was very small, the roles were expanded and interchanged, with the product owner also playing the role of Scrum master who also was part of the development team.

Another adjustment made to Scrum was the frequency of face-to-face meetings. Scrum specifies that they should be daily, the team of this project considered that the meetings should be less frequent, always considering the needs of the development team to responsibly fulfil the assigned sprint in the estimated time. In addition, the sprints had a maximum duration of two weeks while Scrum estimates that two weeks is the minimum time they should take (two to eight weeks) [23].

The advantage of working with small teams is the level of control in both compliance and attention to prioritization of activities by the developers, and attention to the developers by the product owner. Also, in small teams, the distances between team members can be short improving communication [24]. The daily dedication for the development of BellChat was on average two hours per day for each team member. In addition, developers worked remotely, and meetings were held in person twice a week. The Sprints (Sprint backlog not shown due to lack of space) that had to be executed to achieve the successful development of BellChat determined its duration which was in a total of 326 h. The time spent on the project by the product owner is considered only when the activities are to be executed by the project team. The time spent on the project by the product owner is considered only when the activities are to be executed by the project team.

5 Results

5.1 Requirements Elicitation and Analysis

The requirements were taken as the needs of real users (who are related with some of the authors of this paper) with their own visual and hearing impairments. In addition, we analyzed messaging applications currently in use, despite they have poor accessibility and do not comply with W3C standards. In this part of the methodology, use case diagrams and user stories were used for preliminary requirements elicitation and were used as a tool for eliciting the system requirements.

5.2 Bell Chat Architecture

BellChat has been developed following a three-tier architecture: client, application server and database server.

Client layer: The client-side layer is given by the view or interaction with users. Users can interact from any device that has Internet access and a browser (the browser preferred by them). Browsers must be enabled to execute JavaScript and CSS. The browsers that have been tested and have performed very well are Chrome, Opera Gx, Microsoft Edge and Brave (current versions as of October 2022).

BellChat client is a responsive web application, for which CSS was used. Responsiveness gives the application the ability to adapt to any device regardless of screen size. For voice interaction between a user and BellChat, the SpeechRecognition application programming interface (API) was used. In addition, it employs WebSockets for the exchange of messages between users. WebSockets allow the client in client/server applications to receive unsolicited information, which is not normally the case in a web application [25].

BellChat's login interface allows a user to log in with his own account of the application, for which the user had to register beforehand, or with a third-party account (Facebook or Google). Figure 1 shows the screenshot of the chat screen. The logged in user is a hearing-impaired person. The user has sent a text message to his contact *johannapea.* Likewise, he has received an audio message as a reply from this user. BellChat has automatically converted the audio content to text so that the user is aware of its content. BellChat shows both the original message (audio) and the text of its content.

Server layer. In web applications it is usual to have at least a three-tier architecture, which is the case of BellChat. The second layer, i.e. the layer with which a client interacts is the application server (also the web server) where BellChat is hosted. The web application was developed in Java with its JSF 2.3 framework. Moreover, it was developed using the model-view-controller (MVC) model, which makes the web application has all the advantages of this model [26]. In addition, in this layer

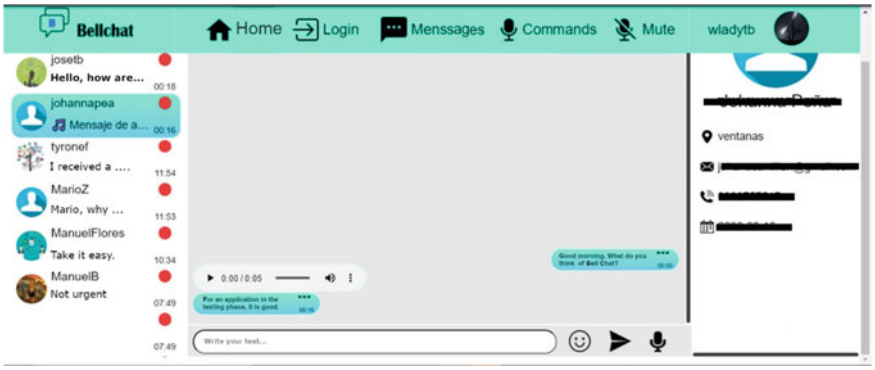

Fig. 1 Screenshot of the message receiving and sending tray in BellChat

we find the sockets endpoint, which makes it possible to send and receive data in real time [25]. All this runs on the Glassfish Server 4.1 application server.

Data layer (database server): The third layer of a three-tier architecture is the data persistence layer [27]. PostgreSQL 13 was the database server on which the BellChat data persistence was implemented.

The tools and technologies used in the development of BellChat are free tools for education and research, so that the cost of producing the application would be the purchase of the domain, server rental and maintenance. In addition, as it is for vulnerable groups of people such as the visually impaired, the hearing impaired, the mobility impaired and the elderly, it is possible to try to obtain all services free of charge.

6 Evaluation of BellChat

Both the elicitation of requirements and the evaluation were carried out through an indirect survey. The answers expressed by the participants were taken directly by the researchers. Among the questions asked to elicit requirements for the implementation of BellChat were demographic questions such as age, disability or not, type of disability, percentage of disability, experience in the use of computer applications, among others. Questions to identify user needs included the difficulties of using third party tools for screen reading (in the case of people with visual impairment), tools for transcribing audio to text and captioning a video for people with hearing impairment.

On the other hand, the questions to obtain the acceptance criteria of BellChat and the possible improvements that can be implemented sought to know from the respondents the importance for them of the functionalities implemented in BellChat and the ease of use of these functionalities. In addition, we wanted to know what

features BellChat lacks, what features could be improved and a general opinion of BellChat as a whole.

Six people accepted to take part in the evaluation. Three of them have a diagnosed visual disability. An elderly person had reduced vision and tremor in his hands. Two people have a hearing disability. There was no collaboration from people with motor disabilities.

The opinion of 100% of the respondents on the implemented functionalities is they are very useful, both the transcription of text-to-speech messages for visually impaired people and voice-to-text messages for hearing impaired people. In addition to considering the auto-configuration of the application according to user profile to make communication between users with different disabilities more transparent, one suggestion everyone made was BellChat should be implemented as a mobile application. The assessment of the user-friendliness of the functionalities of BellChat is related to the experience in using computer applications and to the percentage of disability of the respondent. People with little experience in using computer applications in general and with a high percentage of visual impairment were a bit self-conscious about using the application, but in the end they found it easy to use. While people with a lot of experience in using computer applications needed a brief guide in the use of the application; in the end they stated that it was very easy to use.

7 Conclusions and Future Work

This paper has presented the development and a first evaluation of BellChat, an application for interpersonal communication aimed specifically at people with visual, hearing and motor disabilities, and older people with some pathologies. It is very useful for people considered by society to have normal abilities, who can be doing other tasks while listening to or sending messages to their contacts. Without a doubt, the social group with disabilities is only recently being considered when developing web applications open to the public. The use of web applications for interpersonal communication makes people feel that they belong to a group or society.

BellChat was developed following the Scrum development and project management framework, with some adaptations. Scrum proved to be a framework capable of adapting to a project with a very small team. For the development of BellChat, the team consisted of four people: product owner, Scrum master and two developers. The experience of developing the same between the Scrum master and the developers makes the developers more collaborative to the project goals.

The evaluation of BellChat has motivated the project team to propose soon an application that uses WhatsApp services and has an inclusive interface especially for people with visual, hearing and motor disabilities, as well as elderly people, without leaving out people without physical disabilities and to try to include people with intellectual disabilities. Also, the authors plan to develop a native Android and iOS application as a future work for the convenience of users. Although the current web application is responsive, the user must use a browser to access it. As BellChat was

developed using the MVC development model, its business logic can be reused in other types of applications.

References

1. WCAG 2.0 (2012) Web content accessibility guidelines (WCAG) overview. In: Web accessibility initiative, pp 1–29
2. W3C, Web Accessibility Initiative (WAI). https://www.w3.org/WAI/
3. Shawar BA (2015) Evaluating web accessibility of educational websites. Int J Emerg Technol Learn 10:4–10. https://doi.org/10.3991/ijet.v10i4.4518
4. Broccia G, Manca M, Paternò F, Pulina F (2020) Flexible automatic support for web accessibility validation. Proc ACM Hum-Comput Interact 4:1–24. https://doi.org/10.1145/3397871
5. Pelzetter J (2020) A declarative model for accessibility requirements. In: 17th International web for all conference, W4A 2020. Association for Computing Machinery, New York, NY, USA, pp 1–10
6. Lister K, Coughlan T, Iniesto F, Freear N, Devine P (2020) Accessible conversational user interfaces: considerations for design. In: 17th International web for all conference, W4A 2020. Association for Computing Machinery, New York, NY, USA, pp 1–11
7. Morris MR, Johnson J, Bennett CL, Cutrell E (2018) Rich representations of visual content for screen reader users. In: Conference on human factors in computing systems. Association for Computing Machinery, New York, NY, USA, pp 1–11
8. González S (2022) Best apps for deaf people [Ranking 2022] (In Spanish). https://blog.audifono.es/aplicaciones-perdida-auditiva/
9. Pereira J (2010) Handbook of research on personal autonomy technologies and disability informatics. IGI Global
10. Rahman MS, Reza H (2022) A systematic review towards big data analytics in social media. Big Data Min Anal 5:228–244. https://doi.org/10.26599/BDMA.2022.9020009
11. Onay Durdu P, Soydemir ÖN (2022) A systematic review of web accessibility metrics. In: IGI global, pp 77–108. https://doi.org/10.4018/978-1-7998-7848-3.ch004
12. Arooj S, Zulfiqar S, Qasim Hunain M, Shahid S, Karim A (2020) Web-ALAP: a web-based LaTeX editor for blind individuals. In: 22nd International ACM SIGACCESS conference on computers and accessibility, ASSETS 2020. Association for Computing Machinery, New York, NY, USA, pp 1–6
13. Lee HN, Uddin S, Ashok V (2020) TableView: enabling efficient access to web data records for screen-magnifier users. In: 22nd International ACM sigaccess conference on computers and accessibility, ASSETS 2020. Association for Computing Machinery, New York, NY, USA, pp 1–12
14. Lyall FC, Clamp PJ, Hajioff D (2016) Smartphone speech-to-text applications for communication with profoundly deaf patients. J Laryngol Otol 130:104–106. https://doi.org/10.1017/s0022215115003151
15. Shadiev R, Hwang W-Y, Huang Y-M (2014) Investigating applications of speech-to-text recognition to assist learning in online and traditional classrooms. Int J Human Arts Comput 8:179–189. https://doi.org/10.3366/ijhac.2014.0106
16. Ali FABH, Aydah SM (2012) Development of prototype chat system using mobile platform for disable people. Procedia Soc Behav Sci 57:33–39. https://doi.org/10.1016/j.sbspro.2012.09.1154
17. Alsaif A, Albadrani N, Alamro A, Alsaif R (2017) Towards intelligent Arabic text-to-speech application for disabled people. In: International conference on informatics, health and technology, ICIHT 2017. Institute of Electrical and Electronics Engineers Inc

18. Greco A (2020) Social sustainability: from accessibility to inclusive design. EGE-Expresión Gráfica en la Edificación 18. https://doi.org/10.4995/ege.2020.14072
19. Dwivedi AK, Virmani D, Ramasamy A, Acharjee PB, Tiwari M (2022) Modelling and analysis of artificial intelligence approaches in enhancing the speech recognition for effective multi-functional machine learning platform–A multi regression modelling approach. J Eng Res 10. https://doi.org/10.36909/jer.icmet.17161
20. Pascual A, Ribera M, Granollers T (2015) Impact of web accessibility barriers on users with a hearing impairment. DYNA 82:233–240. https://doi.org/10.15446/dyna.v82n193.53499
21. Al Ghurair N, Alnaqi G, Doush IA (2021) An AAC tablet application for children with language impairment. J Eng Res 9:106–123. https://doi.org/10.36909/JER.V9I2.9201
22. Ceci L, Mobile internet usage worldwide—statistics & facts. https://www.statista.com/topics/779/mobile-internet/
23. Rising L, Janoff NS (2000) Scrum software development process for small teams. IEEE Softw 17:26–32. https://doi.org/10.1109/52.854065
24. Obeidat B, Muhammad BB (2021) A study of communication ties among team members in work environments. J Eng Res 9:1–13. https://doi.org/10.36909/jer.v9i3a.10529
25. Gupta Y, Dewan H, Leekha A (2020) Real-time monitoring using AJAX and WebSockets. J Stat Manag Syst 23(1):125–134. https://doi.org/10.1080/09720510.2020.1714154
26. Zhou K (2022) Research and implementation of MVC design pattern on J2EE platform, pp 1122–1127. https://doi.org/10.1007/978-981-19-4132-0_153
27. Marco P, Marcu M (2022) Impact of the persistence layer on performance and architecture for a Java web application. In: IEEE 20th Jubilee world symposium on applied machine intelligence and informatics, proceedings, SAMI 2022, pp 261–266. https://doi.org/10.1109/SAMI54271.2022.9780780