REVISTA DESIGN & TECNOLOGIA
ISSN: 2178-1974
2019, Vol. 09, No. 19
DOI 10.23972/det2019iss19pp87-111

PGDESIGN
www.pgdesign.ufrgs.br

A roadmap towards mobile applications with accessibility for visually impaired users: guideline and its evaluation

Walter F. M. Correia[1]; Marcelo Penha[1]; Jefté A. Macedo[1]; Clauirton A. Siebra[2]; Marcelo E. Anjos[3]

1 *Programa de Pós-Graduação em Design, Departamento de Design, Universidade Federal de Pernambuco, Recife, Brasil*

2 *Universidade Federal de Paraíba, Centro de Informática, Joao Pessoa, Brasil*

3 *SIDI, Campinas, Brasil*

ABSTRACT

When applications intend to support accessibility, several aspects of interface design and usability must be reviewed to adapt or extend common functional requirements that are implemented to ensure easy use of applications. However, initiatives to develop guidelines for accessible mobile applications are recent and several approaches present only suggestions rather than a concrete list of requirements. It is becoming increasingly apparent not only the need for social and digital inclusion of people with disabilities but also the urgency to make devices and applications truly accessible to anyone. This work focuses on accessibility requirements for visually impaired users and it is part of a broader effort that involves different types of impairments. Our research method was based on three main steps. First, from an initial set of 1014 scientific and technical articles, 46 were analyzed to identify requirements that are related to visual impairments. Second, we carried out an observation-based analysis to confirm the importance of these requirements, considering a set of accessibility tests with impaired volunteers. Third, the Design Thinking process was used to implement an application prototype, considering the most important requirements that were identified along with the two first steps. This application was submitted to a user-centered evaluation with volunteers, which also supported the evolution of our requirements. We sought to include prior knowledge aligned with all pre-identified requirements, culminating in something that would add to the procedure greater validation. As the main result, this paper presents the complete roadmap of this process, which resulted in a guideline and evaluation method to support the integration of accessibility aspects into mobile applications.

KEYWORDS

Accessibility; Interface requirements; Usability; Mobile applications

Um roteiro para aplicativos móveis com acessibilidade para usuários com deficiência visual: diretrizes e sua avaliação

RESUMO

Quando aplicativos pretendem oferecer suporte à acessibilidade, vários aspectos de design e usabilidade de interface devem ser revisados e analisados para adaptar ou estender requisitos funcionais comuns que são implementados para garantir o seu uso mais fácil. No entanto, as iniciativas para desenvolver diretrizes para aplicativos móveis acessíveis são recentes e várias abordagens apresentam apenas sugestões, e não uma lista concreta de requisitos. Tem-se percebido cada vez mais não apenas a necessidade de inclusão social e digital de pessoas com algum tipo de deficiência, mas a urgência em tornar dispositivos e aplicativos de fato acessíveis a qualquer pessoa. Dentro desse contexto, este trabalho enfoca em requisitos de acessibilidade para usuários com deficiência visual e faz parte de um esforço mais amplo que envolve diferentes tipos de deficiências. Nosso método de pesquisa foi baseado em três etapas principais. Primeiro, de um conjunto inicial de 1014 artigos científicos e técnicos, 46 foram analisados para identificar requisitos relacionados a deficiências visuais. Segundo, realizamos uma análise baseada em observação para confirmar a importância desses requisitos, considerando um conjunto de testes de acessibilidade com voluntários prejudicados. Terceiro, o processo de Design Thinking foi usado para implementar um protótipo de aplicativo, considerando os requisitos mais importantes que foram identificados juntamente com os dois primeiros passos. Procurou-se abarcar o conhecimento prévio alinhado a todos os requisitos pré-identificados, culminando em algo que agregasse ao procedimento uma maior validação. Esse aplicativo foi submetido a uma avaliação centrada no usuário com voluntários, que também apoiou a evolução de nossos requisitos. Como resultado principal, este artigo apresenta o roteiro completo desse processo, que resultou em um método de orientação e avaliação para apoiar a integração de aspectos de acessibilidade em aplicativos móveis.

PALAVRAS-CHAVE

Acessibilidade; Requisitos de interface; Usabilidade; Aplicações Móveis

CONTATO: Walter F. M. Correia – *ergonomia@me.com*

© 2019 – Revista Design & Tecnologia

1. INTRODUCTION

According to the World Health Organization, approximately 15% of the world's population lives with some form of impairment. Out of those, 2-4% experience significant functional difficulties. Furthermore, the number of people with impairments is growing. The main reasons are the population aging (older people have a higher risk of presenting impairments) and the global increase of chronic health conditions associated with impairments. Several governments around the world are aware of such numbers and they have created laws in the area of Information and Communication Technologies (ICT). The Website Accessibility and Equality Act 2010 is an example of those (Giakoumis et al., 2010). According to this document, if a website does not meet certain design standards, then its owners could be sued for discrimination. For example, many visually impaired visitors use speech synthesizer software to read the text in the HTML code of web pages and translate it into audible speech. However, several websites include images that contain text as part of the pre-rendered picture file. These texts may be unreadable by the software. If the text is not embedded in the image properties or alternatively available in the text somewhere on the website, this could render the content inaccessible to visually impaired users and, therefore, be discriminatory for the Equality 2010 Act.

Applications that do not provide accessibility can restrict or even prevent impaired individuals to use them. For example, when interface elements are not properly labeled, visually impaired users will not be able to navigate or locate functionalities on the interface through screen readers. If an application is hard to use, individuals will quickly lose their interest in it. Then, this application may receive negative feedback, which reduces its market value.

In this context, ICT companies have realized that social and digital inclusion is a trend that is expanding and gathering people. Thus, besides its obvious importance, there is an increasingly open market for new products and innovations (Fanucci et al., 2011). One of the fundamental challenges of these new products is to ensure their usability since the support for accessibility affects how traditional aspects of usability are designed. Traditional applications already have default patterns of requirements for usability (Carvajal et al., 2013). However, these patterns must be reviewed if applications intend to offer accessibility support. The current literature on mobile platforms (Khan et al., 2013; Siebra et al., 2015a) already presents some efforts in this direction. Kane et al. (2011), for example, proposed an accessibility guide. However, it is only focused on a particular scenario associated with the elicitation of gestures for visual impairment. Mi et al. (2014) proposed a more generic heuristic checklist for accessible mobile interface design, but they did not apply a classification to better contextualize their requirements. There are also contributions from outside of the academy, such as the Android's Accessibility Developer Checklist (Android Developers, 2015), which lists five requirements as mandatory to ensure a minimum level of accessibility. Another example is the W3C effort (W3C, 2015) to describe how the Web Content Accessibility Guidelines (WCAG) 2.0 and their principles could be applied to mobile web content. However, these efforts have focused on specific accessibility applications (e.g. Web applications) rather than trying to establish a general accessibility design guideline for mobile applications. Thus, such efforts are sparse and they are still far away to present a concrete pattern.

Our research contributes to a better understanding of requirements for visually impaired users of mobile applications by means of an exploratory study, which was based on three research steps. First, a literature review analyzed 1014 scientific articles and relevant sites to identify requirements for mobile applications that are aimed at supporting visually impaired users. Second, an observation-based analysis was conducted to confirm the literature review and stress the main requirements and difficulties of impaired individuals when they are using mobile applications on their Smartphones. Third, the main requirements were consolidated in an initial guideline, which was used to develop a mobile application for credit and debit cards management, currently called *Controle Fácil* (Easy Control in English). The user-centered evaluation of this application was carried out with visually impaired volunteers so that we could evaluate the impact of the requirements associated with accessibility and identify other opportunities that could complement the guideline.

More specifically, the paper's contributions are:

- It presents a broad review of the accessibility requirements that are discussed in both technical sites and academic papers regarding the mobile platform;
- It details several experiments concerning the use of mobile applications for visually impaired individuals in the Android platform, stressing current limitations of the technology and extending the results of our previous review;
- It discusses the implementation of an application that considers the main visual impairment requirements along with its development, and the results regarding the user-centered evaluation of such application, which was conducted with visually impaired volunteers;
- It proposes a guideline for functional requirements, which should be considered over the development of mobile applications to ensure accessibility with usability.

The remainder of this paper is organized as follows: Section 2 presents the background of this research, showing the importance of requirements as the first step to define a set of accessibility design patterns to mobile applications. Examples in such direction are also discussed in this section. Section 3 summarizes our research method, which was divided into 3 parts: literature review, observation-based analysis, and user-centered evaluation. The details and results of the literature review and observation-based analysis are respectively presented in Sections 4 and 5. Section 6 discusses the implementation of an application, which was based on our previous results, and the user-centered evaluation carried out with visually impaired participants. Section 7 concludes this work with the final guideline, main remarks, and research directions.

2. BACKGROUND

According to Carvajal et al. (2013), for over a decade, the software engineering community has been actively pursuing different lines of research targeting the incorporation of usability practices into software development. The final goal of such efforts is to consistently transform the usability recommendations proposed by the HCI community, into software code. However, we do not see these same efforts aimed at accessibility aspects. The first step in this direction is to understand the needs of impaired users so that guidelines can be defined according to such needs. Section 2.1 starts by explaining the importance of guidelines for developers. Then, we introduce (Section 2.2) an example of how accessibility could be integrated into existent applications. The next

sections (2.3 and 2.4) respectively address the current efforts to take accessibility into account in system's design.

2.1 Guidelines for Usability

The use of guidelines or design patterns brings several advantages, since they reduce development time, decrease the complexity of understanding the functionality of interface components and improve the quality of resulting software designs (Tidwell, 2010). The process of mobile interface design already makes use of some patterns, such as the way to present notifications (Figure 1).



Figure 1 Example of notification patterns of popular applications.

Notifications are used to obtain users´ attention on recent activities and they are important because mobile users are commonly consuming lots of information every day but at the same time, they may be busy and cannot promptly answer messages and other requests. However, users still want to know, as soon as possible, if there have been new activities or actions requiring their attention so that they can filter what it is important. Based on these features, mobile applications have implemented almost the same notification patterns, with minor variations. For example:

- Twitter indicates new activity by placing a small dot at the top of the timeline icon (Figure 1, left-hand side);
- Facebook displays notifications regarding new items in the newsfeed with a popup banner that drops down within the app (Figure 1, center);
- LinkedIn places a numbered badge on labels with new updates (Figure 1, right-hand side).

Applications that use such patterns tend to be easily accepted by individuals because they are familiar with this style of interface. Note, however, that this kind of pattern must be modified to attend further needs of impaired users, such as visually impaired users who cannot see dots and badges. Furthermore, as discussed by Wobbrock et al (2011), modifications and other efforts in this direction must consider how systems could be made to fit the abilities of whoever uses them, rather than assisting human users to conform to inflexible computer systems. Based on this idea, systems should be aware of the abilities of their users and provide interfaces that are better suited to such abilities.

2.2 Web Content Accessibility Guidelines

Web Content Accessibility Guidelines (WCAG) 2.0 define how to make Web content more accessible to people with impairments, so that these guidelines involve a wide range of disabilities, such as visual, hearing, motor, cognitive, and neurological disabilities. These guidelines also make Web content more usable by the elderly who lose some abilities due to aging and often improve usability for users in general. In order, studies that intend to support older adults and impaired users could share their findings since they have several features in common.

The initiative to develop the WCAG brings some lessons learned, which are important to other efforts that also intend to provide accessibility. The basic questions from the WCAG initiative are: How do people who cannot move their arms use websites? What about people who cannot see well or at all? Or people who have difficulty hearing or understanding, or have other accessibility needs? This means the first action of such initiative was to understand how people with impairments, including people with age-related impairments, see the Web and the design barriers they encounter in this process. The main resources used in this direction were the Stories of Web Users that are stories of selected scenarios of people with impairments using the Web, which highlight the effect of barriers and the broader benefits of accessible websites and web tools.

2.3 Efforts from Outside of Academy

The majority of proposals regarding accessibility guidelines for mobile applications come from organizations, such as Microsoft, Android Community, and BBC. The main examples are:

- Funka Mobile Accessibility Guidelines;
- Mobile Accessibility at W3C;
- Android Developers Accessibility Guide from Google;
- Accessibility Programming Guide for iOS from Apple;
- Design Guidelines for Windows Mobile from Microsoft;
- BBC Mobile Accessibility Guidelines.

Funka's guidelines for the development of accessible mobile interfaces (Funka, 2016) was a project financed by the *Internet Fund .SE* and based on the WCAG 2.0. According to the authors, WCAG 2.0 is not enough to provide accessibility for mobile applications, so that they developed their own test criteria that supplement the international regulations. The authors also indicate that all recommendations of their guidelines were tested in real life. Unfortunately, details of their method and affirmations are not public and their final document lists a set of 48 recommendations, which are structured into 6 classes: choice of solution, design, layout and design, interaction, content, and user settings. However, several of these recommendations are not directly related to accessibility, while others have a strong relation with web content. This relation could be already expected since their study is based on WCAG 2.0. Some examples of recommendations that support these affirmations are:

- (Recommendation 1) Ensure that your basic website works on mobile devices;
- (Recommendation 2) Do not force the user to use a mobile version, but do offer one of the pages on the basic website are large or the functions are complex;
- (Recommendation 5) Follow WCAG 2.0 except where these guidelines contradict WCAG 2.0;
- (Recommendation 6) When creating applications for specific devices, any design and accessibility guidelines must be followed, provided they do not contradict these guidelines.

The W3C Mobile Accessibility (W3C, 2017) is another effort that is based on the WCAG initiative and focused on Web content. This strong bias to Web is observed in its definition of mobile accessibility which "*refers to making websites and applications more accessible to people with disabilities when they are using mobile phones and other devices*". As stated in the W3C documents, the aspects of mobile accessibility are covered in existing W3C WAI accessibility, so that there are not separate or specific guidelines for mobile accessibility. Thus, developers must carry out information mining to find requirements directly associated with the specification of mobile-accessible applications. These requirements are shared in documents such as: "Accessibility Information for Specific Technologies " (e.g. HTML 4), "*How to define a way to make dynamic web content and web applications developed with Ajax, DHTML, and other Web technologies more accessible*", and "*Mobile*

Web Application Best Practices". This latter document is more structured and explicitly lists 32 best practices, such as "Use Cookies Sparingly" and "*Use Appropriate Client-Side Storage Technologies for Local Data*". These practices are found in a topic called "W3C addresses mobile accessibility". However, the details of such practices do not show the relation between practice and accessibility. The description of "Use Cookies Sparingly" illustrates this affirmation. Cookies are a common and effective resource to store small states on clients. They are appropriate for simple personalization data and commonly used to store a token representing user identity to enable automatic sign-in. Information stored in cookies, however, is sent to the server for every request. Therefore, its excessive use can negatively impact performance, particularly on a mobile network. Furthermore, in the mobile context, cookie support cannot be relied upon since it may be disabled either in the device configuration or by the mobile network. For this reason, applications should try remaining functional even if cookies are unavailable. Similarly to Funka guidelines and any other guideline discussed in this section, the method to define the W3C practices and their evaluation are not discussed. They are only listed and technically explained.

Differently from the Funka and W3C approaches, the Android developers' community gives a broader perspective to its *Accessibility Developer Checklist* (Android Developers, 2015). This means the requirements checklist is aimed at any kind of Android mobile application. However, such requirements were specified to ensure a minimum level of accessibility and only five steps are required to obtain such level: (1) describe user interface controls, providing content descriptions for user interface components that do not have visible text; (2) enable focus-based navigation, making sure users can navigate along with screen layouts using hardware-based or software directional controls; (3) custom view controls, implementing accessibility interfaces and providing content descriptions; (4) no audio-only feedback, that is, audio feedback must always have a secondary feedback mechanism to support users who are deaf or hard of hearing, and; (5) test accessibility by navigating along the application using directional controls, and using eyes-free navigation with TalkBack enabled. To this last step, the Android document also brings an *Accessibility Testing Checklist* (Android Developers, 2015), which evaluates the correct implementation of the previous steps in the Android platform. Apart from these five items, two other classes of requirements called accessibility recommendations and special cases or considerations, respectively describe further 3 and 10 requirements.

The iOS domain also presents an approach to accessibility that is similar to Android. Its *Accessibility Programming Guide for iOS* (iOS, 2015) provides support to the following accessibility features: (1) Zoom, which magnifies the entire device screen; (2) White on Black, which inverts the colors on the display; (3) Mono Audio, which combines the sound of the left and right channels into a mono signal played on both sides; (4) Speak Auto-text, which speaks the text corrections and suggestions that iPhone makes while users type; (5) Voice Control, which allows users to make phone calls and control iPod playback using voice commands. However, the main iOS efforts rely on the VoiceOver screen reader, which is aimed at visually impaired users.

The accessibility checklist of Windows Mobile is also minimalist and is focused on Microsoft technology. The high-level specification of such checklist only suggests four items: (1) set the accessible name (required) and description (optional) for content and interactive interface elements in your app; (2) implement keyboard accessibility; (3) visually verify your interface to ensure that the text contrast is adequate, elements render correctly in the high-contrast themes, and colors are used correctly; and, (4) run accessibility tools, address reported issues and verify the screen reading experience. This latter item shows that Microsoft technology gives high importance to compatibility issues between their applications and external accessibility tools and products.

Table 1 Comparison of different guidelines for accessibility in mobile applications.

	Requirements clearly presented	Structure for requirements	Source of requirements	Evaluation Discussion	Main feature
Funka	Yes, 48 items	Yes, 6 classes	Not presented	Not presented	Focus on web aspects
W3C	No	No	Not presented	Not presented	Focus on web aspects
Android	Yes, 18 items	Yes, 3 classes	Not presented	Initial demonstration	Minimalist set
iOS	Yes, 5 items	No	Not presented	Not presented	Minimalist set and focused on VoiceOver resource
Microsoft	Yes, 4 items	No	Not presented	Not presented	Minimalist set and based on its own platform
BBC	Yes, 53 items	Yes, 11 classes	Part of the requirements come from other guidelines	Not present, but shows how the evaluation could be carried out	Consolidates and extends previous guidelines

The BBC Mobile Accessibility Guideline presents the most recent list of requirements, which is based on the works of Android Community, iOS, and W3C. Thus, this document consolidates previous guidelines and also shows the relations between them when such relations exist. For example, the requirement "*The color of text and background content must have sufficient contrast*" has the following equivalences:

- Android: use standard Android OS colors for buttons, text, and other user interface elements or ensure the foreground and background colors provide sufficient contrast;
- iOS: use standard iOS colors for buttons, text, and other user interface objects or ensure the

foreground and background provide sufficient contrast;
- W3C: for text or images of text avoid background colors or use background colors that have sufficient contrast from the foreground color. If background colors are used, apply the 4.5:1 contrast ratio to foreground colors.

When a new requirement is identified and such requirement does not have equivalence in some of the other approaches, then the BBC documentation shows how it could be implemented in iOS, Android or Web platforms. Another interesting feature of the BCC documentation is its structure, so the identification of requirements is straightforward. The 53 requirements are divided into 11 audio and video classes (4), design (12), editorial (3), focus (6), forms (5), images (2), links (3), notifications (4), scripts and dynamic content (5), structure (4), and text equivalents (5). While part of the requirements come from other guidelines, it is not clear how other requirements are specified. Thus, the BBC guideline suffers from the same problem that the previous approaches, which do not discuss the methodology and evaluation of its requirements. Table 1 summarizes this discussion.

This table shows that the majority of approaches have a minimalist set of requirements, differently from the BBC approach that has a more concrete suite. Funka´s guideline also presents an extended list of requirements, but several of these requirements are focused on Web aspects. Our final guideline is similar to BBC work. However, BBC and other works neither present a formalism to define the requirements nor their rationale. This formalism is important because it could support, for example, the measurement of the importance of the requirements (e.g. identify when they are essential or desirable) and their process of implementation.

2.4 Academic Contributions to Mobile Accessibility

Differently from the previous efforts, academic contributions to mobile accessibility tend to deeply analyze a specific problem rather than providing an extended guideline. Consider blindness and the problem of generating hearing output to compensate for the lack of visual sense. The generation of hearing output is a requirement cited by the majority of the previous guidelines. However, they do not present details of its importance and specific use/implementation. On the other hand, academic contributions explores this problem in details, such as the next examples:

- Signal processing system that analyses the visual scene to identify objects of interest and encode them into sound (Lescal et al., 2012);
- Multimodal interactive system for non-visual (auditory-haptic) exploration of virtual maps (Geronazzo et al., 2016);
- Benefits and drawbacks of simultaneous spatial sounds in auditory interfaces for visually impaired and blind computer users (Sodnik et al., 2011).

Like these examples, several other studies have focused on specific accessibility applications/requirements rather than trying to establish general guidelines for accessible design (see Appendix A). This means that current efforts are aimed at independently developing and testing single solutions, so that the literature does not present systematic methods to elicit requirements from individuals with impairments or guidelines on accessibility issues. The study of Kane et al. (2013) considered this topic, but it is focused on a particular scenario, which is associated with the elicitation of gestures to identify differences between people with visual impairments and sighted people regarding gesture preferences and various gesture parameters. Based on their results, Kane and colleagues proposed preliminary recommendations on how to design future touchscreen-based applications for both blind and sighted users. Similarly, the work of Moreno et al. (2017) presented recommendations that include indicators that can assist in the design and evaluation of accessible media players, while Grussenmeyer and Folmer (2017) provided a review and recommendations of technologies for touchscreen accessibility for people with visual impairments.

Mi et al. (2014) proposed a more general heuristic checklist for accessible mobile interface design, developed through reviewing existing design standards and guidelines and validating these guidelines with user involvement. However, they did not apply a classification to better contextualize the requirements of these guidelines according to the type of impairment, as proposed in Siebra et al. (2015b). Their classification was based on five general features: (i) Mechanical Control (e.g. keys that prevent slipping), (ii) Display (e.g. screen and menus that are easy to explore without excessive searching), (iii) Speech and General operation (e.g. easy access to voice mail without long key sequences), (iv) Audio Feedback (e.g. a brief, distinct sound must be heard when an item is selected) and (vi) Touch Feedback (e.g. user vibration feedback must be localized at the hand or touching/activation finger rather than vibrating the entire device).

Our work mixes the main features from both commercial and academic sides. While we intend to create practical guidelines, such as the commercial approaches; we also intend to provide a rationale for each requirement. This fact justifies the importance of looking for requirements in the academic literature. Even when academic works only consider a single or a group of related requirements in their analysis, it tends to be based on formal methods and evaluations that provide the rationale required in our work. For example, the results of the paper “Color Transformation for color Blind Compensation on Augmented Reality System” (Ananto et al., 2011) show the importance of adjusting the color scheme and other image features to different levels of blindness. Thus, such results are the basis for one of our requirements, which is stated as “*Applications must offer adjustable brightness/contrast/color controls. This feature changes the foreground/background color of the screen and modifies the brightness to meet individual needs*”. Thus, our work is initially a process of finding and consolidating previous and shared requirements, such as the BBC approach, to create our guideline. However, we carried out this process using a formal protocol and mainly using academic works as a source of information. After that, other research methods are applied to confirm and extend our findings, as detailed in the next section.

3. RESEARCH METHOD

The definition of a requirements guideline, which can ensure proper usability to mobile applications when impaired individuals use them, has not been clearly defined in the literature so that exploratory research was employed. Such kind of research often occurs before we know enough to make conceptual distinctions and it frequently relies on secondary research such as literature reviews or qualitative approaches such as discussions and interviews. Figure 2 illustrates the process and the stages applied over the three steps of our research.

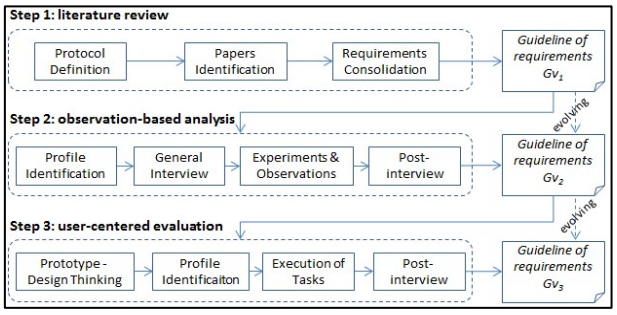


Figure 2 Research method schema.

The details of each of these steps, together with their results, are discussed in the next sections.

4. LITERATURE REVIEW

This section describes the path taken to conduct a literature review on accessibility requirements.

4.1 Method description

The aim of the literature review on requirements aimed to identify and characterize the main accessibility requirements, discussed in the current technical and scientific literature, regarding mobile platforms. The review was based on a protocol that led the process. This first step of this protocol was to define a research question to guide the selection of papers and this question was: *What are the modifications or further resources that must be considered along with the implementation of mobile applications that intend to support impaired users?*

The primary idea of a systematic literature review is to carry out an extensive search for scientific and technical papers to answer pre-defined research questions. To that end, the limits of the review must be set and the second step is to identify keywords for search so that such keywords support the data retrieval on empirical research regarding our domain. The definition of such a string is not simple since the terminology in this subject is rather complex. To facilitate the string construction and ensure a good coverage, we decided to separate the search string in three dimensions (X, Y, Z) and chose general terms (x_1, x_2 ,..., x_n; y_1,y_2,..., y_n; and z_1, z_2, ..., z_n) for each of these dimensions. Then, the logic connector AND was used between dimensions, while the logic connector OR was used between terms inside each dimension. The dimensions and their rationale are detailed in Table 2.

Table 2 Dimensions of the search string and their terms

Dim	Definition	Terms
X	This dimension is composed of terms regarding the main subject of research that we intend to focus on.	Requirement, Recommendation, “Best practices”, Requisite
Y	This dimension qualifies and restricts the subject of research since our interest is on a subgroup of such subjects.	Accessibility, Ability, Capability, Disability, Impairment, Assistive
Z	This dimension indicates the type of computational platform, which is the target of this research.	Smartphone, “Mobile device”, “Mobile platform”, “Mobile applications”, “Mobile interface”

The definition of terms and final search string were also based on the experience obtained from pilot attempts, which were important to refine the string; while variations of the suffixes of words were treated according to the resources of each search engine employing special symbols (e.g. impair* = impairment + impaired + impairments). The final search string format, as a Boolean expression, was:

(Requirement OR Recommendation OR “Best practices” OR Requisite) AND (Accessibility OR Ability OR Capability OR Disability OR Impairment OR Assistive) AND (Smartphone OR “Mobile device” OR “Mobile platform” OR “Mobile applications” OR “Mobile interface”)

The next step was to choose the databases. Initially we decided to use several databases; however, the results showed a large number of duplications in references. Thus, our work focused on four search engines. *ACM Digital Library, IEEE Xplore, and Scientific Direct* are the main search engines to works in technology. PubMed seems to be the main health literature database and its use was intended to verify if accessibility is also considered in some way from the health side.

The final step of the protocol is the paper selection, whose aim is to identify papers that are relevant regarding the pre-defined research question. The use of search strings does not ensure this aim and it is not expected that all studies initially returned would be used in the final phase of analysis. Thus, we have three phases of selection and respective criteria of inclusion/exclusion:

- Selection of studies based on search string (Phase 1): studies that contain the pre-defined search string in the title or abstract, text in English language, date from 2007 to 2019, and they are not editorials, prefaces, discussions, comments, summaries of tutorials, workshops, abstracts, and panels;
- Screening of title and abstract (Phase 2): quick checking of studies to verify if they cover aspects of our research area. We only considered primary studies. This means, informal or systematic reviews are not considered for the next phase. Duplicates are also eliminated in this phase;
- Relevance analysis (Phase 3): the full text is analyzed along the data extraction to verify if the study brings enough information to answer the research question.

It is important to stress that this analysis does not necessarily refer to evaluating the quality of a paper. Rather, the idea is to select papers regarding its relevance to our analysis. All papers in systematic reviews have been published and thus quality should be assured through peer-review. In other words, the relevance evaluation, regarding the objective of our review, was aimed at ensuring that papers finally included were indeed focused on our domain of research and had sufficient details for empirical studies.

4.2 Results

The papers’ selection resulted in 46 from the initial 1014 scientific and technological articles (Table 3). The information extracted from such articles was consolidated in 23 requirements that compose the first version of our guideline (Gv1). From these 23 requirements, 13 are directly associated with visually impaired users and 10 other requirements (R27 to R36) are not exclusive but also important to such users. Requirements from R14 to R26 are associated with other types of impairments (e.g., hearing and motor) and their discussion is out of the scope of this work. T

o each requirement we indicate one or more studies ([Sxx] in Appendix A) that discuss or are related to such a requirement. Papers in this appendix also show how the requirement was validated, since the huge majority of the

papers in our literature revision present empirical validation methods, such as usability tests with volunteers.

Table 3: Quantitative summary from the literature review. See the 46 final papers in Appendix A.

Database	Number of Paper Selected		
	Phase 1	Phase 2	Phase 3
ACM	724	57	19
IEEE Xplore	143	16	4
ScienceDirect	136	39	23
PubMed	11	0	0

The first set of requirements is associated with the generation of hearing output to compensate for the lack of visual sense. These requirements are:

- R_1 - The name of a character that is being entered must be heard;
- R_2 - Names of items and images on the screen must be heard as they are touched or selected;
- R_3 - Clear feedback must be provided for all actions/interactions, via tactile feedback, voice and/or event sounds;
- R_4 - Presence of Screen Reader strategies.

Mi *et al*. [S15] explicitly cite R_1 and validate this and other requirements by means of usability tests with volunteers. Al Mourad and Kamoun [S02] discuss the lack of textual descriptions in no-textual interface elements (R_2) as one of the main accessibility problems. Ahmetovic [S01] shows the importance of feedback that uses different sensorial channels when changes in interface elements are performed (R_3). To that end, experiments with an application for spatial navigation were carried out. Byerley and Chambers [S03] and Coonin [S04] also stress this importance, using different scenarios (library access and e-journal applications). Byerley and Chambers [S03], in particular, conduct comparisons with applications with and without feedback resources and show the superior usability of the first group when blind individuals use them. The importance of screen readers (R_4) is discussed by De Rosa and Justice [S26], who show the limitations of current approaches, mainly in terms of portability.

Alerts are a special case of hearing the output. Important alerts of the system must be generated and recognized by users and in some cases, they must confirm the visualization of these alerts to resume the execution of the system. To ensure easy identification of this alert, the next requirement should be implemented:

- R5 - Provide information alerts by other communication channels beyond visual (e.g. voice).

Several studies demonstrate the importance of the use of alerts by means of real experiments. The main examples are associated with applications for navigation, such as the studies of Faria *et al*. [S06], Gallagher *et al*. [S07], Ganz *et al*. [S08] and Fernandes *et al*. [S27]. However, the studies are very diverse, presenting information alerts for changes in weather (Saravanan and Vineet [S17]) and UV intensity (Puente-Mansilla *et al*. [S44]). Skulimowski *et al*. [S45] employ a combination of text-to-speech and sonification techniques to ensure clarity of alert messages as well as the high responsiveness of the application. One of the important contributions from these investigations is to show that individuals feel more comfortable when they are sure about the existence of such alerts. This requirement is already implemented in most of the traditional systems, where sound alerts and text messages are delivered in case of important events such as errors. However, it is important to stress the mandatory feature of this requirement in the accessibility context.

The use of tactile information is also important, mainly as a resource to create a “save point” command, which leads the navigation of the device to a known place. Thus, applications must consider that the Smartphone home button is a resource to reset the application and start again the interaction process. This is useful, for example, when users found themselves lost along with navigations. Such fact motivated the next requirement:

- R6 - Provide a discernible tactile “home” that ensures there is one key that can be easily and quickly identified on any tactile control pad, to allow users to orientate themselves within the interface.

Mi *et al*. [S15] directly cite this requirement as one of the results of its experiments with twelve participants, who were asked to complete 6 tasks regarding common basic digital music player operations. According to this work, this requirement allows users to better orientate themselves within the interface. Feng [S24] also discusses the problem of finding specific controls, such as the “home” option, in the interface. However, their experiments are focused on tactile design for on-body placements that seem to be valid alternatives to control Smartphones.

The touchscreen of devices does not provide tactile information to users since their screens are flat. Thus, visually impaired users are not able to find specific types of elements, which are used to initiate the devices’ operations. Thus, this initialization function should be modified by the next requirement:

- R_7 - Users must be able to start their devices in any position on the touchscreen.

According to McGookin *et al*. [S14], a particular problem, identified by the visually impaired participants, was the awareness of his location on the touchscreen concerning its borders. Therefore designers should avoid gestures that require users to interact with specific spatial locations of the screen.

The literature also discusses requirements for visually impaired users that have strong visual limitations but are still able to see (R_8 to R_{13}). The main idea of these requirements is to adapt the visual information delivery, such as increasing the text fonts or the color contrast between text and background. Thus, users will be able to use computational applications in traditionally way. These requirements are:

- R_8 - Provide documentation in alternative forms, using big fonts, without a serigraph, with good legibility, providing alternatives to information presented in colors.

Motiwalla and Qin [S16] investigated alternative forms to present information. Their studies were focused on educational environments to demonstrate that alternative forms to deliver information and adapt such information may improve the accessibility of such applications. This demonstration was carried out with 10 participants from two different blind institutions in Massachusetts who used a

mobile prototype that was able to deliver document content in different forms.

- R_9 - Ensure that the perception of text and figures can be also comprehended without the visual sense of colors.

Mascetti *et al*. [S32] investigated the problem of applications whose interface depends on the use of colors. To address this problem, they proposed to use resources that could read the color of the object pointed to by the user's fingertip. Issues related to the possible confusion between foreground and background colors are also discussed. Its results were based on the evaluation of prototypes with nine volunteers.

- R_{10} - Applications must support customizations and avoid modifications in user-defined configurations. This means, the application must maintain the font, size, color and other configurations, which were already defined by the user;

Kane *et al*. [S10] used a survey with visually impaired volunteers to stress the most commonly requested interface features, such as screen magnification and larger buttons. Participants of their experiments used devices with some adjustable settings, but they complain that the allowable ranges for these settings were often limited. For example, some devices allowed users to increase the text size, but only to a certain point. According to their conclusions, mobile devices should be designed to be configurable to arbitrary settings. Furthermore, such settings must be saved for later interactions.

- R_{11} - Present canvas amplifier and the possibility to limit the maximum zoom, without loss of the characteristics of the canvas configuration (e.g. position of icons, etc.).

Krainz *et al*. [S30] show, with the execution of experiments with 12 volunteers, the importance of maintaining the difference between the various interface elements, which must be prominent even when frames change their dimensions.

- R_{12} - Provide clear textual equivalences with semantics to avoid mistakes when texts are read on the screen.

The experiments of Ludi *et al*. [S12] present the scope of the usability needs in terms of understandability and other aspects related to interface design. Technical terms or works that are not common to the traditional vocabulary, for example, must be avoided.

- R_{13} - Adjustable brightness/contrast/color controls. This feature changes the foreground/background color of the screen and modifies the brightness to meet individual needs.

Waite *et al*. [S19] discuss the importance of supporting the adaptation of interface features, such as the contrast between foreground and background, to attend the huge diversity of individuals. Its experiments are based on usability experiments that use applications to support visits in museums. Ananto *et al*. [S46] also show the importance of adjusting the color scheme and other image features to different levels of blindness. The last set of requirements (R_{27} to R_{36}) is not exclusively associated with visual impairments. Rather, they are also important to other types of impairments, such as motor and hearing impairments. These requirements are:

- R_{27} - Provide adaptable layouts without loss of information (customization resources);

Lin *et al*. [S13] describe an example of a user interface, which allows changes in its parameters to support an easy-to-distinguish layout and shows that it is more easy-to-use and acceptable for visually impaired. Similarly, Chiara *et al*. [S42] also discuss the importance to reorganize and adapt layouts for 2-dimensional data sets, such as geographical data, showing its better usability.

- R_{28} - Signalize changes in the interface via sounds, images and tactile sensors (e.g. vibration);

The importance of signalization of interface changes, using different resources, is one of the most discussed topics in the accessibility literature. Experiments that show this importance can be found in several studies such as Sun *et al*. [S18] (notification with predefined vibrating patterns), Wise *et al*. [S20] (notifications for navigation-based interfaces), Götzelmann [S28] (proposal for notifications regarding interface for maps) and Kim *et al*. [S11] (context-dependent notifications).

- R_{29} - Guide and provide feedback to users about what is going on (state of the system);

Adams and Kurniawan [S23] show that participants were pleased with being able to locate photos more quickly and share the stories of these photos when the application was able to guide and provide feedback along the process. The use of speech is almost compulsory for this requirement since the aim is to describe situations or events for users.

- R_{30} - Provide touch interactions;

The use of touchscreen is already considered as the main interface pattern for all mobile users and researches try to advance its features to impaired users. For example, Guerreiro *et al*. [S29] carried out experiments with 15 impaired participants to show that accuracy and precision of touch interactions vary across different areas of the screen and directions, in a way that these features are directly dependent on target sizes. Mascetti *et al*. [S31] investigated special forms of touch interaction, such as braille-based typing applications for touchscreen devices. Oliveira *et al*. [S33] and Kalmaç and Diri [S43] also carried out usability experiments regarding touchscreen interactions.

- R31 - Provide form instructions;

Park *et al*. [S35] showed the importance of instructions when forms present features that are not usual to users. Such importance was demonstrated with the 28 participant experiments using a prototype of a taxi reservation system.

- R32 - Provide linear and intuitive navigation;

Kiat *et al*. [S21] explicitly demonstrated the confusion generated when applications do not present a liner and intuitive navigation. Their experiments observed the execution of simple operations in both WhatsApp and Facebook Messenger applications, and they concluded that the flow of some operations was problematic for participants. To create more intuitive navigation, Penev and Wong [S36] proposed an on-demand automatic clustering of a page's hyperlinks to assist a blind user in quickly locating the desired link. Clustering related links into cohesive groups allowed

users to focus their attention on a subset of content, providing more intuitive navigation. Rodrigues [S38] also showed the importance of linear operations. Its approach automatically generates macros (sequence of commands) when repeated sequences of interactions are detected. Thus, a linear sequence of operations is transformed into a single selection.

- R_{33} - Provide configuration of feedback speed.

The Honk Kong Accessibility Handbook [S09] discusses the need for speed configuration regarding feedback of actions such as navigation through the screen pages and pressing buttons. Dim and Ren [S05] also identified issues during usability tests, where participants complained about the speed of interactions.

- R34 - Provide assistance to find content and guide users through the system.

The experiments of Mi *et al*. [S15] showed that it is still a challenge for blind people to discover functions on the mobile screen and to input commands so that the use of assistance or more natural form of interaction is required. This is especially important when users are interacting with new applications.

- R_{35} - Avoid limits of time control to users' reading or interactions;

The importance of avoiding limits of time controls was demonstrated by Bossini and Moreno [S22] using practical experiments with real users. The range of time to complete the same operation is broad, considering visually impaired users, and it depends on several factors, such as their experience and type of impairment.

- R_{36} - Provide coverage of users' errors and suggestions about their corrections.

Dim and Ren [S05] explicitly list the question "Are users notified of errors?" in their heuristic checklist. Another question on the same list is "Does the phone allow error corrections?" These and other questions were consolidated from a survey with visually impaired users, which indicated the importance of these features.

The idea of enabling configurations is the main point of this last set. The level of impairment is not, in fact, a discrete feature. Rather, such level is represented inside a continuous range, so that the configuration of parameters, such as layouts features (R_{27}) and feedback speed (R_{36}) must be customized in accordance with different users' needs.

5. OBSERVATION-BASED ANALYSIS

5.1 Method description

The Observation-based Analysis was mainly used to observe in loco the type of problems that real users have when they interact with four mobile applications. The aim was to use very popular applications since they provide the main patterns of interaction, which are followed by several other applications. These applications are Facebook, WhatsApp, GloboNews and common embedded mobile accessories. Facebook and WhatsApp are the most downloaded Smartphone applications, according to sites such as Google Play Store. GloboNews is the main news application in Portuguese, similar to BBC in the UK or CNN in the USA. Embedded mobile accessories include the calendar and contact list applications, which are very common in any kind of Smartphone. These applications were previously analyzed so that we identified their main accessibility problems and this information was used to lead the definition of the test scripts.

The analysis was carried out with five volunteers (U1 to U5). As the scripts (see Table 6) are very contextualized and present a strict set of activities, the results exposed an exponential decay behavior in terms of new findings to each new volunteer. This experimental behavior corroborates conclusions of Nielsen (2000) who says, "The best results come from testing no more than 5 users and running as many small tests as you can afford". According to Nielsen, 5 users are enough in the context of a website which reduces the context where usability issues may appear. Similarly, we reduced this context employing our scripts and usability issues tend to be identified because we already know they exist. By the way, we must be aware that this low number of volunteers and the impairment diversity avoid any kind of requirements evaluation because it is not statistically representative. However, this is not our intention since the identified requirements (Gv1) were already evaluated in their original studies. Differently, we already know the accessibility problems of the listed applications we aim to see how this group of volunteers reacts to such issues so that we could be able to ponder over our initial list of requirements and evolve it.

Table 4: Volunteers profile

U_i	Gender	Age	Education	Job	Impairment type	Impairment cause
U_1	Male	24	Technician	Lecturer	Blindness	From birth
U_2	Female	37	Ongoing High school	-	Low vision (3%)	Retinal detachment (age of 30)
U_3	Male	48	Graduated	Seller	Low vision (Left eye: 10%; Right eye: 2%)	Medical error (age of 26)
U_4	Male	60	Post-graduated	Administrator	Low vision (5%)	From birth
U_5	Female	28	Ongoing graduation	Trainee	Low vision (less than 10%)	From birth

Along with the experiments, three researchers observed each volunteer and each experiment was also audio/video-recorded. Data collection was carried out after a presentation of the research and motivations, together with an initial open conversation to relax the participants. The sequence of activities was: (1) identification of the volunteer's profile; (2) interview regarding the general use of Smartphones and their applications; (3) observation of the volunteer behavior along with interactions with Smartphones and pre-defined applications; and (4) post-interaction interview.

The identification of a volunteer's profile was obtained via a structured guide with personal questions related to gender, age, educational level, job, type, and cause of impairment. Previous experience in the use of Smartphones

was mandatory to all participants. Table 4 summarizes the information collected from these 5 volunteers.

The interview about the use of Smartphones also followed a structured guide (Table 5) with 8 questions (Q_1 to Q_8) aimed at obtaining a general view about the use of mobile devices and their applications from the volunteers' perspective.

Table 5: Questions to obtain a general view on the use of Smartphones and their applications

Q_i	Questions
Q_1	How do you use your mobile and which are the main problems regarding this use?
Q_2	Is there any special reason to buy the mobile phone model that you are using?
Q_3	Did you configure your mobile by yourself, or someone assists you?
Q_4	Are you using any useful accessibility configuration?
Q_5	Do you use some external hardware to assist you?
Q_6	Which are the main applications that you use in your mobile device?
Q_7	Do you know other applications that present the same functions? If so, why do you choose the previous one?
Q_8	Do you remember some application that you tried to use but were not able to? What did happen?

After these interviews, the volunteers were asked to perform four scripts regarding types of interactions with different applications, as described below:

- Interaction with Mobile applications:
  1. Open the calendar and identify the weekday related to the date 13/Jul/2015;
  2. Add a notification to this date;
  3. Access the list of contacts and localize the contact "X";
  4. Modify the accessibility option "Background light duration" to 5 minutes.
- Interaction with WhatsApp:
  1. Open the WhatsApp;
  2. Update the contact list;
  3. Start a conversation with "X";
  4. Take a photo from the local where this experiment is being performed and send it to "X".
- Interaction with News application (GloboNews):
  1. Download the application;
  2. Open the application;
  3. Access the option "News at 10'O Clock";
  4. Play the fourth available video;
  5. Go back to the initial screen (home) of the application.
- Interaction with Facebook:
  1. Open the Facebook;
  2. Find user "X";
  3. Access the last notifications.

Finally, a further post-interaction interview was carried out just to enrich the information from our observations. The guide for this final interview is shown in Table 6.

Table 6: Guide for post-interaction interview.

P_i	Post-interaction questions
P_1	Do you have any problem along with the use of this application? Which?
P_2	What do you think could improve in such applications? Is there some additional function they should have?
P_3	What is your favorite feature of this application?
P_4	Are you able to understand what is written in (or spoken by) the application? In the negative case, what is the problem?
P_5	Did you already have an idea of an application but did not find it? In which context do you think that idea could be important?
P_6	Is there anything else that you would like to comment on?

All data acquired in the form of audio was transcribed and coded. As exploratory work, we employed an iterative analysis approach using a mixture of inductive and deductive codes, such as in (Lee et al., 2017). The deductive code used a template approach, as outlined by Crabtree and Miller (1999), which involved a template in the form of codes from a codebook to be applied as a means of organizing text for subsequent interpretation. In our case, this codebook is the Gv_1 and each requirement is a code of this codebook. However, the analysis of the text at this stage was guided, but not confined, by the preliminary codes. During the coding of transcripts, inductive codes (requirements) were assigned to segments of data that described a new theme observed in the text (Boyatzis, 1998). The results of all this process were then employed to evolve the initial guideline (Gv_1) to a second version Gv_2.

5.2 Results

This section first consolidates the interviews with volunteers, giving a general idea on common types of interactions and problems faced by visually impaired users (Section 5.1.1). Then, we present our observations about the execution of the tasks, which were specified in Table 6, together with the results of the post-interaction interview (Section 5.1.2). Both sections employed the text coding to deductively relate information from our analysis to requirements, as explained in Section 5.1. New requirements, which were not discussed in the literature review, were inductively coded as [**NR_**]. Finally, we redefine the first version of our guideline of references (Section 5.1.3), considering the results of Sections 5.1.1 and 5.1.2.

5.2.1 Consolidation of Interviews

The interviews showed that visually impaired users do not complain about the touch screen style of interfaces. They did not demonstrate the desire of trying other approaches and they think this kind of interface should be maintained [**R_{30}**]. We discussed other types of interaction interfaces (e.g. Braille input devices), which were our object of study in other works (Siebra *et al.*, 2015c). However, they are not keen on such resources due to their costs and practicality ("*they are further devices that need to be carried around*"). Exceptions were related to situations that require a big amount of data entry, which is usually carried out at home or work. In such cases,

they agree that the support for this kind of resource could be a good aspect and bring advantages to applications [**NRl**].

The interviews make clear that the touch screen only works well when the Smartphone has a good screen reader [R_4] ("*Obviously, it is not possible to locate interface elements such as buttons and text fields without the screen reader*"). Volunteers discussed their bad experiences when their Smartphones did not speak or incorrectly speak the name of interface elements ("*I hear the 'click' when we touch the element, but no information is given or such information does not make sense for me*"). This is certainly an issue of applications that do not correctly label their interface elements [R_2]. Volunteers also said that touch screen sliding gestures are difficult when several interface elements are on the same screen due to cognitive load. However, this is again a particular problem of applications that do not respect a proper distribution and distance amongst interface components, which must be customized to each user [R_{27}].

Volunteers with very limited vision (U_2, U_4 *and* U_5) said that they could identify icons and buttons if their contrast with the background is high. Thus, it is important to provide ways to promote such adjustments [R_{13}]. However, the same users are still not able to identify details of images and texts without zoom [R_{11}] ("*zoom is fundamental for me*"). For U_1 who is blind, his main complaint was associated with applications that do not speak the text he is entering in text input components [R_1] ("*it is frustrating when you are not sure if what you have written is correct*"). In this context, the use of the word/sentence suggestion (word complete function) was cited as an interesting resource to facilitate the text entry [**NRc**].

Another problem discussed in the interviews was the lack of support when problems occur. In some cases, users need to reset the Smartphone to start again. They suggested a common and easy-to-find command that could only reset the current application rather than the whole system. The home button seems to be the best candidate for this function [R_6]. However, rather than starting again, volunteers indicated that they would prefer to figure out the current situation and try to continue. In this case, they need assistance from their application [R_{34}]. In this and other scenarios, users indicated the audible information as a fundamental form of feedback [R_3][R_{12}] ("*if I do some action, I need to know if such action was done*"). They complained, for example, that some events, such as screen rotation, are not indicated in several Smartphone models.

Users said it is easier to interact with buttons located at the borders of the screen since such positions are easier to memorize. According to our observations, U_1 used to directly go to Apps and Contact-buttons, which were located on the board, rather than sliding their fingers to find such buttons. Thus, the arrangement of components is an important feature [**NRf**] for interface design. Furthermore, we also observed that U_1, U_2, and U_4 do not only memorize the position of the components, but also the sequence of commands to perform some operation. U_5 related that he has problems when an application does not present an intuitive or "traditional" interface (interface that employs an already consolidated pattern, which is used in the majority of applications) (Tidwell, 2010) since he needs to navigate for several paths to become familiar to the application template before its use [R_{32}]. This navigation without utility, or without following the standard interface, is very stressful and, according to users, it could be improved if application forms could provide some type of assistance [R_{31}]. This scenario is more stressful when there are timelines (list of events in chronological order) to complete some type of interaction [R_{35}]. Another stressful situation happens when the "back" button of a Smartphone is accidentally pressed because users became lost since this button is not read by the screen reader and users do not realize that it was pressed [**NRj**]. Regarding the use of the back button function, volunteers related that a better use for this button is to return to the home form of the application rather than leaving the applications and going back to the home screen of the device [**NRi**].

Users presented problems to navigate in an application when it presents sequential forms or sequential components, but the screen reader does not inform where the user is in such sequence. For example, "video 1 of 4", "item 1 of 9" [**NRe**]. They also tend to sequentially search for items in a list by jumping blocks of items when these items are alphabetically ordered. After jumping a block, an aleatory item is selected to know where they are on the list. If the list was not alphabetically organized, then they needed to look item by item [**NRg**].

5.2.2 Observations and post-interaction interview on the execution of tasks

As previously presented in our research method, a post-interview was individually carried out after the conclusion of all tasks. The next paragraphs consolidate the main comments of the volunteers on the use of each of the applications.

Globonews (news application). This application forces the change of screen orientation and this feature created different levels of difficulty to all five users. Furthermore, this option is not customized [R_{10}]. All users also had problems identifying some items in this application since they did not present labels with clear descriptions [R_{12}]. Some components were identified as "Button 50" and "Button 90". The application did not also warn about changes in the screen orientation [R_{28}] and the numeric ordering of available videos [**NRe**]. Thus, the screen reader was not able to identify where users were on the video sequence. During the experiments with this application, some buttons became hidden while videos were in execution. This feature created problems for users because it was hard to find such controls. For example, users spent a long time to find the pause function [**NRh**]. To access the fourth available video (requested task), all users needed to count the screens while they were passed along. Another problem was related to return to the initial application form, as asked in the script. As this application did not present a HOME button [R_6], all users closed the application to execute this step.

WhatsApp. The most difficult task was to find a new contact in the application since there is a system delay in updating the list and the application does not warn users about this delay [R_{28}]. A second problem observed was related to the task of sending photos since some required buttons did not present labels [R_2]. Third, the lack of sound/tactile feedback [R_3] to confirm the image capture also affected the correct sequence of activities. Finally, users did not know WhatsApp has a search function. Thus, rather than use such function, they tried to find a contact through the contact search function [**NRk**].

Facebook. Four of five users were able to complete the tasks since they quickly found the text field associated with the search function. Furthermore, the use of a list was an appropriate navigation option for them. Only one volunteer had problems because characters were not entered in the text field [R_{29}] since this text field was not selected and the application did not warn her about this problem [R_{36}]. Thus, she thought the text field was correctly filed and continued with other operations. At this moment, our team interfered in

the process and concluded the experiment since she was not able to identify the problem.

Agenda: the first problem in using this application was to open it since its label (SPlanner, name of the application) does not bring a proper meaning to users [**R_{12}**]. Furthermore, it is an English term rather than some name in the native language of users. In the end, the identification of such an application was carried out with the assistance of the team. The next task (finding a date and identify the weekday of such date) also presented problems since the agenda does not inform the weekday when a date is selected in the calendar [**R_2**]. The identification of the date was hardly carried out since the dates were very close and small [**R_{11}**]. U_3 was the unique volunteer able to perform the second task (schedule a meeting in the indicated date) since his Smartphone (iPhone) allowed voice command via Siri (personal assistant for Apple Smartphones) [**NR_a**]. The technical identification of interface elements [**R_{12}**] and several elements that are not important to a task [**NR_b**] did not assist the process of finding the correct elements such as buttons and text fields. Along the process of reading phone numbers on the agenda, the screen reader concatenated the number before saying it. For example, 2234323 was read as "two million two hundred thirty-four thousand..." rather than "two, two, three ..." [**NR_d**]. In order, the agenda was the less accessible application due to the complexity in presenting data and related calendar information to visually impaired users.

5.2.3 Redefining the Requirements

The results of the interviews and observations, described in the previous sections, were useful to confirm the importance of some requirements discussed in our literature review, while also indicated necessary extensions in some of these requirements. Furthermore, we could also identify new requirements, which were not discussed in the literature. Table 7 presents the second version (Gv_2) of the requirements, which were consolidated after the observation-based analysis. The table organizes the requirements by their type (e.g. function, color, screen orientation, etc.) and nature (interface, interaction, navigation, and others). The *Code* indicates the coding symbols used in the previous sections. This code came from the literature review [R_1..R_{13}, R_{27}..R_{36}], or the observation-based analysis [NR_] when this process was the source of the requirement. Requirements can have the same description as presented in the first requirements version (Gv_1), a description derived/evolved from Gv_1, or be completely new [NR_]. The table also shows an attribute called *F-code*, which is used as the final codification in the third version of the guideline (Gv_3).

Table 7: Second version of the Guideline of requirements (Gv2).

Type	Code	Fcode	Requirement Description
			INTERFACE
FUNCTION	NRb	R01	Adopt components and information in the interface that directly contribute to the application's functionalities. Avoid components with an only aesthetic purpose.
COLOR	R_8	R02	Ensure that components can be understood without the visual sense of colors.
	R_{12}	R03	Use high-contrast colors in the interface elements concerning the background.
SCREEN ORIENTATION	R_{28}	R05	Do not change the screen orientation pattern without prompting users. If the horizontal screen orientation is more suitable for some application functionality, allow users to choose which orientation they prefer.
TEXTUAL AND NUMERICAL DESCRIPTION	R_{11}	R06	Use terminology in the native language with simple terms, without technical words.
	R_2	R07	Ensure that all visual elements (including non-textual components) of the application interface are properly labeled in the implementation of the code.
	R_{11}	R08	Ensure that the subtitled content description clearly and succinctly expresses the full functionality/meaning of screen elements. E.g.: "S-Planner - Diary," "Button 'Back'", "Field 'Insert your name'".
	NRd	R09	Provide textual/numerical description according to the context.
	NRe	R11	Inform and identify the ordering of sequential components. ("Page 01 of 03"; "Five items listed", "Item 1 of 5").
	R_2	R12	Describe images and figures textually, if they exist, ensuring that they can be read by a screen reader.
COMPONENTS SIZE	R_{10}	R13	Adopt components with sizes that facilitate user access to all components - Minimum dimensions of the buttons; - Minimum height of the components; - Minimum space between components.
COMPONENTS LOCATION	NRf	R14	Set the location/distribution of components on the interface so that the user can easily access to all components. - When possible, place the most important/usual functions at the extremity of the interface. - Avoid interaction components with very different sizes on the same screen. - Avoid interfaces with many interacting elements on the same screen.
	R_{27}	R15	Set the location/distribution of components on the interface so that the user can easily interact with all components. - Avoid placing drop-down lists below the middle of the screen, especially when there are many items listed. - Avoid placing interaction components very close to the screen edges. - Prioritize the use of quadrants for each selectable component. - Prioritize the provision of large amounts of information in list forms, sorted in ascending / alphabetical order. - The toolbar should be static on the screen with scrolling.
	NRg	R16	Prioritize the arrangement of components forms in one item per line, avoiding, whenever possible, multiple columns in the same row.

Table 7: (Cont.)

			INTERACTION
VOICE COMMAND	NRa	R21	Allow interaction via voice command.
UNLIMITED TIME	R_{35}	R22	Provide unlimited time, without disabling the screen, for user interaction.
REDUCE EFFORT	NRc	R23	Offer features that reduce the effort to enter data (e.g. "autocomplete").
VISIBILITY	NRh	R26	Avoid hiding application interaction elements when given screen/functionality is in use.
FEEDBACK	R_1 R_{29}	R27	Inform everything that is happening in the application.
	R_4	R28	Ensure that when the user touches any interface component, the content of this element is immediately read, cutting off any possible reading in progress.
	R3 R_{13}	R29	Give sonorous feedback about all user interactions.
	R3 R_{13}	R30	Give tactile feedback about all user interactions.
			NAVIGATION
HOME	R_5	R32	Provide access to app "Home" on all screens to allow users to more easily navigate along with the application.
BACK	NRi	R33	Offer the "Back" button, programming it to return to the previous screen of the application.
	NRj	R34	Make the "Back" button on the device recognizable by the screen reader.
SEQUENTIAL	R_{32}	R36	Provide sequential navigation between screens, also considering the trigger of the "Back" button.
PROVIDE HELP	R_{36}	R38	Provide coverage of user's errors. (i.e.: not fill a required field or fill incorrectly, cancellation of information without saving, etc.).
	NRk	R39	Provide search function in the app, especially for items localization tasks in endless scrolling lists.
	R_7	R42	Guide users along with the navigation of an application when this is their first access.
	R_{34}	R43	Provide help to find content.
	R_{31}	R45	Provide instructions for data input fields in the interface.
			OTHERS
CUSTOMIZATIONS	R_{33}	R48	Allow users to configure the application's notifications and speed.
COMPATIBILITY	NRl	R49	Support the integration of input and output devices.
	R_4	R50	Provide full compatibility with screen readers.

When some part of audio or video information is identified as a reason to define a new requirement, we create a reference and link it to such a requirement. For example, Table 8 shows an example where we have identified a problem, created a requirement to cover this problem and linked three references that support the existence of this requirement. The references indicate three different video files and the initial time when the information can be seen. This process was carried out to all requirements in Table 8.

Table 8: Example of references to support the definition of requirements.

Problem/situation	Proposed requirement	References
During the process of reading the phone number on the agenda, the screen reader concatenates the number before saying it. For example, 2234323 is "two million two hundred thirty-four thousand..." rather than "two, two, three ...".	Provide textual/numerical description according to the context.	DV02 (Vídeo USER05A: 15'05") DV04 (Vídeo USER07A: 04'06") DV05 (Vídeo MOV04989: 09'45")

6. USER-CENTERED EVALUATION

This section describes the process to perform accessibility tests with visually impaired users and validate requirements from a developed application.

6.1 Method description

The target of this evaluation is an application called *Controle Fácil*, which intends to provide strong support to visually impaired users. The motivation to develop this application was first based on reports of volunteers' problems in using current financial management applications. According to them, such applications are not accessible at all and our team, which evaluated several applications from this category, confirmed this fact.

Design Thinking (Beckman and Barry, 2007) was the process used to develop our application. Based on this process, we carried out 5 steps, which were: (1) Empathize - work to fully understand the experience of the user for whom we are designing. We conducted this step by observing, interacting, and immersing ourselves in their experiences; (2) Define - process and synthesize the findings from the previous work in order to form a user point of view to address our design; (3) Ideate - explore a wide variety of possible solutions through generating a large quantity of various possible solutions, allowing to step beyond the obvious and explore a range of ideas; (4) Prototype - transform our ideas into a physical form so that we can experience and interact with them and, in the process, learn and develop more empathy; and (5) Test - try out the prototype and use observations and feedback to refine such prototype, learning more about users, and refining our original point of view.

The *Controle Fácil* development considered the main requirements identified in the *Guideline of Requirements – V2* (Gv_2). *Controle Fácil* is currently compatible with Samsung S6 Edge Plus Smartphone, and its function is to support the control of expenses that are carried out with credit and debit cards. The registration of expenses can be performed via both manual insertion and/or SMS messages. In this latter case, the notification of the credit/debit operation is sent via SMS from the bank associated with the card to the Smartphone. The application has three main functional frames: cards, graphics/reports, and expenses. The cards frame provides functions to list, add, and delete cards, where each card is characterized by a set of features (e.g., brand, expenses limit,

etc.). The graphics/reports frame provides visual elements to present the types of expenses according to different classes such as clothes, supermarkets, and others. Users can also track their expenses according to the maximum limit of each card. At last, the expenses frame provides information about individual expenses, which were performed with each card. Information about the date, value and place/service are given for each entry. Examples and screenshots about this application can be seen in Appendix B.

After the development of *Controle Fácil*, it was evaluated by 10 (U_a, ...U_j) visually impaired volunteers, whose profile is summarized in Table 9. According to this table, the majority of the volunteers have a very good educational level and only one has less than two years of experience regarding the use of Smartphones. Thus, we can eliminate the possibility of a lack of experience when an activity is not concluded.

The evaluation sessions were carried out at the usability laboratory of the Samsung/CIn Research and Development unit. Three different and simultaneous videos were recorded to register the events. The first camera was placed beside the volunteer and its focus was on the user interaction with the Smartphone. The second camera was placed in front of the volunteer and it was focused on the user's face. The third camera is, in fact, an application called *AZ Screen Record* that was installed in the Smartphone to capture the sequence of interactions.

The same evaluation session was applied to each volunteer. First, a short description and project goals were presented, as well as the evaluation method. For example, volunteers must know that the experiment is going to be recorded for future references and their data are confidential. After that, a consent form was assigned and the session could start. An initial chat "to relax" the volunteer before the evaluation was very useful in our previous experiences and we also carried out such chat as the first step in this evaluation. The average time of this chat was 15 minutes and 39 seconds (max: 24:22, min: 09:38).

Table 9: Profile of volunteers.

U_i	Gender	Age	Education	Experience with Android	Smartphone	Impairment and cause
U_a	Female	31	Graduated	5 years	LG lite 2	Born totally blind
U_b	Male	41	Postgraduate	3 years	iPhone 6	Total, retinal cancer at a few months of life
U_c	Male	26	Technician	3, 5 years	Samsung J1 mini	Born totally blind
U_d	Male	38	Graduated	6 - 7 years	Moto G4 Plus	Born partially blind
U_e	Male	61	Postgraduate	9 years	Android and iPhone	Partially blind, illness
U_f	Male	50	Postgraduate	2 years	Moto G2	Partially blind, medical error (age of 26)
U_g	Male	28	Ongoing Graduation	3 years	Motorola G4 Play	Partially blind, congenital glaucoma
U_h	Female	21	Ongoing Graduation	4 years	Samsung J5	Born totally blind
U_i	Male	51	Elementary level	3 months	Moto G4	Totally blind, illness (age of 10)
U_j	Female	36	Elementary level	2 years	Samsung J5	Born totally blind

The next step was an exploratory stage where volunteers could freely interact with the application and its functions. Tasks were not given in this stage and the accessibility configurations were active in the Smartphone. The aims of this stage were: (1) observe the initial impressions of volunteers on the application, (2) identify the spontaneous interaction patterns, (3) identify their understanding of the application functions, (4) identify their understanding of the interface components, interface structure and navigation schema, (5) identify the main problems regarding the free interaction, (6) identify their decisions along the interactions, and; (7) get users familiar with specific functions regarding activities 09 and 10. In this stage, the moderator could ask questions to better understand what was going on. Examples of questions were: "What are you trying to do?", "Are you having problems in this interaction?", "Is there something wrong or strange?", "Do you think there is a better way to do that?". The next stage was the execution of 12 tasks, which were:

- TK01 - Accept legal terms of use;
- TK02 - Decide to register a user;
- TK03 - Register a card;
- TK04 - Manually Register an expense (x2);
- TK05 - Edit a card;
- TK06 - Register a user (email and password);
- TK07 - Login (email and password already registered);
- TK08 - Register expense via SMS (x2);
- TK09 - Access information from the history graph;
- TK10 - Access information from the category graph;
- TK11 - Filter expenses by a period;
- TK12 - Remove expenses already registered.

Differently from the exploratory stage, the tasks in this stage have a pre-defined and formal specification of the steps that must be followed. For example, the next schema (Table 10) illustrates the specification of TK5 (Edit a card). It is important to emphasize that during this step we explained to participants that we were evaluating the application rather than its users. Thus, the concept of right and wrong does not exist. Participants were also motivated to talk anything along with the experiment freely. Furthermore, participants could leave a task at any moment.

Table 10: Example of task specification.

TASK 05 (TK05)	
Title	Edit a card
Conditions	The application must have at least a card inserted in the list.
Main tasks	Edit the information of a card already inserted in the list.
Scenario	"Imagine that you lost a credit card xx that is registered in the application. Now you received a card to substitute it and you would like to update the information of the lost card. You know you need to modify the last four digits and the brand of the lost card. The last four digits are 5544 and the new branch is MasterCard".
Resources	- The information must be given to volunteer along with the experiment
Activities to be observed	1. Localize and open the application in the main Smartphone screen; 2. Find the "Cards" tab; 3. Click on the card that must be modified; 4. Choose the feature that must be updated; 5. Modify the four last numbers and brand of the card; 6. Confirm the change.
Conclusion	The task is concluded when the volunteer confirms the modification and returns to the "Cards" tab.
Final orientation	"Now you have concluded this task, please, go back to the Home screen of the device".
Issues to be observed	- The facility/difficult of the volunteer in finding where she/he could edit the card; - The facility/difficult of the volunteer in editing information of the card in the edit field and lists; - The time spent by the volunteer to conclude the task.
Post-task questions	- Do you think this process could be simpler? - Were you expecting this navigation sequence, or is there some divergence that you were not expecting? - Do you think the process is very long or complex at some moment? - Do you feel the lack of some feedback or information from the system? - Do you think some feedback or information is not important?

Two Smartphones were used for these activities. Tasks from TK1 to TK6 used the first Smartphone, whose application database was empty. Activities from TK7 to TK12 used the second Smartphone that had some registers in its database, which were important to the execution of such activities. For example, registers of 40 expenses and 5 credit cards. Such Smartphones were configured under the accessibility profile (e.g., use of a keyboard, screen reader velocity, etc.) of each volunteer before their use. This step was carried out to create a more familiar scenario for them.

Finally, an audio-recorded post-interview was individually carried out after the conclusion of all 12 tasks. These questions were divided into three blocks. The first block is associated with the satisfaction of users regarding application accessibility. The questions were:

- Q_1: How do you feel using this application? Why? Have you considered a positive or negative experience? Did you like to use this application?
- Q_2: Considering a scale from 1 to 10, how could you evaluate your satisfaction in using this application, where 0 is completely unsatisfied and 10 completely satisfied?
- Q_3: How do you evaluate the accessibility of this application?
- Q_4: Which are the best and worst features of this application?

The next four questions are associated with the facility of use or usability of the application:

- Q_5: Could you indicate the situations that you think the use of this application was easy?
- Q_6: Was there any situation where you had problems in using the application along with the tasks? Could you describe this situation?
- Q_7: Do you think any aspect could be implemented differently? Which way?
- Q_8: Do you have any suggestions to improve the accessibility of this application?

The two final questions are associated with feedback and information received from the application:

- Q_9: How do you evaluate the feedbacks provided by the TalkBack/VoiceAssistant along with the use of this application?
- Q_{10}: How do you evaluate the information provided by the text, which was read by the TalkBack/Voice-Assistant along with the use of this application?

At last, the mediator allowed further comments, suggestions or criticisms. The consolidation and analysis of the results were based on rate of completeness (measures if users were able to complete the tasks), completeness time (average time that volunteers spent to complete the task) and errors/problems (description of errors or problems that occurred along the interaction to each of the proposed activities and also the relevance of them).

6.2 Results

The following paragraphs present the requirements derived from the *Controle Fácil* experiments and their respective justifications for validation.

6.2.1 Requirements and their rationale

As discussed in Section 5.2.3, the guideline identified 34 requirements in its second version (Gv_2). A subset of 21 requirements was considered over the implementation of

Controle Fácil. These requirements are listed as follows, together with the rationale of their implementation.

R01 - *Adopt components and information in the interface that directly contribute to the application's functionalities. Avoid components with only aesthetic purposes*. Our experiments showed that visually impaired users are focused on functions. This scenario is different when we present an interface with elements that are not contributing to the real function of the application. When these elements are presented in the interface, users tend to visit them to understand what they are and what kind of contribution they could give to the process in execution. Thus, the team only maintained esthetic elements when they could bring some important information. For example, the welcome screen (Appendix B), where the application trend mark (image) brings an overview of the application when it is touched.

R02 - *Ensure that components can be understood without the use of colors*. One of the limitations of visually impaired users is to discriminate colors, which are used to indicate functions or information. *Controle Fácil*, for example, employs graphs where colors represent information. To consider this requirement, every graph has a dynamic legend that is composed at execution time and read to users when the graph is touched. Initially, the color was not part of this legend. However, one of the volunteers asked the reason to retreat the colors from the textual description, since this information could assist him in imagining the graph in his mind.

R03 - *Use high-contrast colors in the interface elements concerning the background*. The implementation of this requirement was based on a W3C color contrast checker tool, whose function was to verify if the contrast between interface elements and background for all frames was adequate. Furthermore, A/B split tests were carried out with low vision volunteers to validate the color pattern used in the application, also considering color-blindness (daltonism) users. This feature could be configured as a whole. This means changes in the contrast are used in all frames so that they have the same pattern.

R6 – *Use terminology in the native language with simple terms, without technical words*. The language of *Controle Fácil* is very simple and only uses native terms without acronyms. An easier understanding of terms resulted in a more efficient interaction since users did not spend time trying to comprehend what the functions are in fact for what.

R07 - *Ensure that all visual elements (including non-textual components) of the application interface are properly labeled in the implementation of the* code. Visually impaired users simply ignore visual elements that are not labeled. This fact is quite obvious once users navigate along with screens through the feedback that they receive when elements of such screens are touched. Thus, all interface elements of *Controle Fácil* have their label, since element without label represents the same as an empty area inside the interface.

R08 - *Ensure that the subtitled content description clearly and succinctly expresses the full functionality/meaning of screen elements. E.g.: "S-Planner - Diary," "Button 'Back'", "Field 'Insert your name'"*. This requirement seems simple, but it requires careful attention since similar sentences can have different interpretations. For example, we had problems with the title of some screens, such as "Insert Card". When the screen reader-generated (read) this sentence, users thought they could already save the data about the card touching in the screen title. A suggestion was to change this title to "Form to Insert a Card". The option "Save" also had different semantics in different forms and this fact also generated problems. Thus, rather than only use "Save", we modified these functions to include the semantics of their screens. For example, "Save Card" on the screen to insert a card and "Save expense" on the screen to insert an expense.

R09 - *Provide textual/numerical description according to the context*. Our experiments showed that the read of textual information must consider its context to become clear to users. For example, screen readers use to spell the content of the "Email" and "Login" fields. This means character after character so that parts such as ".com" become very strange or hard to understand to users. Thus, *Controle Fácil* reads this and other predefined terms as a word. Other examples in *Controle Fácil* are currency values, which are read as a complete number, rather than digit after digit.

R11 - *Inform and identify the ordering of information and listed and/or sequential components. ("Page 01 of 03"; "Item 1 of 5"; "SMS 1 of 4")*. As previously discussed, visually impaired users navigate along with the interface supported by the feedback that they receive when they touch components. However, interfaces in *Controle Fácil* that have pages present a different kind of navigation. In this case, users need to know the page where they are and the total number of pages, so that they could have a better idea of position and time to conclude the navigation. This concept of page is ample and can be used in screens with several tabs, for example. In this case, the system must read the information "Tab 1 of 5" if users access the first tab of a screen with 5 tabs. This simple requirement gives exactly the notion of space that can still be explored. This same idea can be extended to list components. For example, access to the first element of a menu list with 10 elements could generate a speech like "Item 1 of 10" before the name of the item. The list of expenses is an example in *Controle Fácil* (see Appendix B).

R12 - *Describe images and figures textually, if they exist, ensuring that they can be read by a screen reader*. Images and figures are interface components that require more than a simple label to present some significance to users. They require a minimum description to ensure that visually impaired users can imagine the image. This is not a trivial problem since descriptions must be long enough to pass the meaning of the image, but also not so long to avoid users to become impatient. *Controle Fácil* implements this feature to describe its graphics, which are generated in the report screen.

R13 - *Adopt components with minimum touch target size and spacing that facilitate user access to all components*. This is a simple but very important requirement since a high number of components placed very close in a screen can increase the number of interaction errors and delay the conclusion of tasks. Usability tests showed that one of the menus of our application had a very small touch area and it was exactly one of the main points of problems along with the interaction experiments. Thus, both size and distance between elements must be respected and well dimensioned. Our specification relies on the MIT Touch Lab study (Dandekar *et al*., 2003) to choose a proper size for buttons and other interface elements.

R14 - *Set location/distribution of components on the interface so that the user access to all components can be facilitated*. The pattern is the key work for this requirement. This means the distribution of the components along the screens of the application must follow a unique pattern since

this feature assists the process of learning and lead users to have a better usability experience. Based on this requirement, we decided to use the pattern of *Google Material Design* for the location and distribution of components on a screen, since it is part of the daily activities of a huge number of people. For example, the location of a confirm button must be placed at the right superior corner of insertion screens. This approach takes advantage of past experiences of users so that they feel more comfortable in using the application. Examples of good practices regarding this requirement are: (1) when possible, place the most important/usual functions at the extremity of the interface; (2) avoid interaction components with very different sizes on the same screen, and (3) avoid interfaces with many messy interacting elements on the same screen.

R15 - *Set location/distribution of components on the interface so that the user interaction with all components can be facilitated*. Visually impaired users always explore the interface when they need to carry out some activity but do not know the interface yet. They intend to memorize the location of available components and their functions. Thus, the components must be distributed on the interface in such a way to facilitate the interaction of users with them. Good practices used in *Controle Fácil* were: (1) avoid placing drop-down lists below the middle of the screen, especially, when there are many items listed; (2) avoid placing interaction components very close to the screen edges; (3) prioritize the use of quadrants to each selectable component; (4) prioritize the provision of large amounts of information in list contextualized forms, for example, sorted in ascending/alphabetical order; and (5) fix the toolbar on screens with scrolling.

R16 - *Prioritize the arrangement of forms components in one item per line, avoiding, whenever possible, multiple columns in the same row*. This requirement is very important in forms with several text fields to be filled. Experiments showed that visually impaired users tend to navigate in the vertical way (from top to down). Thus, *Controle Fácil* follows this strategy, since elements that are placed beside others are not quickly identified.

R22 - *Provide unlimited time, without disabling the screen, for user interaction*. The implementation of deadlines to carry out an operation tends to generate stress in users and even irritability since users may be trying to conclude an operation when such operation is interrupted. The best practice was to identify long pauses during an interaction and generate questions to configure out the problem. For example, if users spend a long time to complete the expenses information (expenses insert form), a popup window asks what such users want to do at that moment and offers buttons with interaction options to assist them.

R23 - *Offer features that reduce user effort. Ex.: "autocomplete"*. The autocomplete is a classical resource to accelerate the use of general text fields that must be filled. Our application applied this resource in the "Search Screen" but only 3 volunteers used this resource. However, we observed that the interaction of these volunteers was faster when compared to the interaction of others. Autocomplete is only one of the resources that have this effect. Developers must identify opportunities to implement other approaches that have the same objective and some of these approaches may be dependent on the application domain.

R27 - *Inform everything that is happening in the application*. It is not enough to just generate audible messages regarding direct actions of users, such as touches in components and changes of screens. Some events are generated by their system, such as the identification of errors, which must be informed to users. Some of these messages are already patterns and expected by users. For example, in the process of text entry, the expected behavior is that the system read what was already filled when space is entered. We have not implemented this behavior in *Controle Fácil*. Thus, we observed that users tried to confirm the conclusion through touch in the text field so that the content could be read. However, most of the time this touch deleted the content and users needed to do the process again. This example shows the importance of a good audible information strategy.

R28 - *Ensure that when the user touches any interface component, the content of this element is immediately read, cutting off any possible reading in progress*. This requirement was the subject of several experiments in *Controle Fácil*, so that we could decide the correct action. In the first approach, a current message was not interrupted when users touched on another component or carried out other actions. Rather, new messages were stacked so that they could be started just after the conclusion of the current message. This approach generated several problems of interaction so that we decided to implement the strategy described in R28. Special cases can appear, but they must be analyzed as exceptions.

R29 - *Give sonorous feedback about all user interactions*. Even when text is not part of interactions, such interactions must still have associated audible feedbacks to confirm their execution. For example, *Controle Fácil* provides specific sounds to the operation of scrolling forms, mainly to indicate that the form has reached its final. These types of feedback are essential since visually impaired users are not able to identify situations like that without audible indications.

R33 - *Offer the "Back" button and it should be programmed to return to the previous screen of the application*. The Back button is one of the most used resources in a web or conventional application. This function has already a universal design pattern (left superior corner) that assists its fast identification and returns to the previous screen. This pattern was confirmed by our experiments since volunteers directed their touch to this position rather than trying to find it via successive touches on the screen. An interesting behavior happened when users wanted to return to the previous screen, but the current screen did not present this option. In this case, users closed their applications and reopen it to go to the required screen. This is a clear situation of work overload and justifies the use of the back button on all screens.

R36 - *Provide sequential navigation between screens, also considering the trigger of the "Back" button (physical or logical)*. It is important to create applications with a logical sequence of actions. This sequence can be understood using the semantics of the data. In *Controle Fácil*, if we need to insert an expense, we first need to have a card to associate to this expense. Thus, users intuitively know if they need to go forward or backward in their navigation. Furthermore, related information and functions are closer (e.g., all functions related to expenses are on the same screen) so that users do not need to carry out several steps along with their navigation. Note that the specification of this logical sequence is domain-dependent.

R39 - *Provide search function in the app, especially in items localization tasks in endless scrolling lists*. The search resource, which is available in screens that have a high amount of items (e.g. expenses in *Controle Fácil*) was considered essential to accelerate the conclusion of the activities that needed to find a specific item in a list with several items. The activity "Delete an expense" is an example. As this search function does not have a universal pattern for

its location in the interface, several users did not find and use it.

The other 13 requirements were not implemented due to their technical difficulty or because the application did not create opportunities for their implementation. For example, R50 (Provide full compatibility with screen readers) was not considered because our team was only able to configure the use of screen readers compatible with the Smartphone in use. R21 (Allow interaction via voice command) is also complex since the application must rely on the voice command embedded in the Smartphone, or implement its own strategy.

6.2.2 Temporal analysis

The next results (Table 11) show a quantitative analysis regarding the time to complete the 12 tasks. Apart from TK11 (Filter expenses by period), all other activities had a high rate of conclusion and all volunteers were able to conclude 8 from 12 tasks. This high number of concluded activities is an indication that the application indeed offers good support for visually impaired users.

The bad results regarding TK11 were then better analyzed. The task specification gives the next scenario to each volunteer: “Please, could you access the expenses screen to show us which were your expenses with meals from 1st January 2016 until today?” According to the results of interviews and recorded videos, we observed that the main problems were the difficulty in finding the context menu and the reasoning to map the filtering criteria from this sentence to the correct fields. As this activity could be considered complex and it was carried out at the end of the process, some volunteers were also tired.

Table 11: Analysis regarding the conclusion time of tasks.

Task	Average Time	Longer Time	Shorter Time	Concluded Tasks
TK01	0:03:36	0:14:55	0:00:45	10
TK02	0:01:00	0:04:25	0:00:11	10
TK03	0:13:03	0:18:13	0:08:50	10
TK04	0:18:06	0:26:30	0:07:10	10
TK05	0:08:37	0:17:13	0:03:37	10
TK06	0:06:48	0:10:27	0:04:25	10
TK07	0:06:01	0:10:55	0:01:47	10
TK08	0:08:42	0:21:10	0:00:14	9
TK09	0:01:51	0:03:25	0:00:45	10
TK10	0:01:02	0:02:49	0:00:22	9
TK11	0:07:51	0:21:50	0:07:23	6
TK12	0:03:38	0:06:19	0:01:53	8

Another observation based on Table 11 is about the high differences between the shorter and longer times of each activity. However, this fact was expected since the profiles of volunteers are not homogeneous (Table 9). For example, U_i has an adverse combination of factors. He is a total blind, but he was not born blind. He lost his vision when he was 10 years old. Besides that, he has a short time of practicing with smartphone (3 months). Differently, we had volunteers with 7 years practicing with smartphones. However, rather than presenting a disadvantage, this difference of profiles is important to our research, since we want to provide support for diverse levels of visually impaired users.

6.2.3 Synthesis of the execution of the tasks

Next paragraphs summarize the results regarding the execution of each task.

TK01 (Accept legal terms of use) - 90% of the volunteers considered this activity as easy, while 10% considered its difficulty as being a medium level of difficult. About 30% of participants had problems in scrolling the screen associated with the "Terms of Use" because the text was very long and users needed to reach the bottom of the page to find the interaction components. In order, some users (10%) were expecting that these buttons were already down on the screen. Users (20%) did not also understand that the "Enter" button was only active when the “Accept Box” was selected. This problem mainly happened because users interrupted the ongoing explanation, which says, for example, that such components are at the final of the screen. Based on this problem, interaction components were placed at the bottom of the page and they were always visible.

TK02 (Decide to register a user) – 100% of the volunteers considered this activity as easy and with very good resources of accessibility. Depending on the user’s decision, for example, they could directly be forwarded to TK06.

TK03 (Register a card) - 60% of the volunteers considered that this activity had a medium difficulty, while 40% considered as easy. The main problems were related to the process of filling currency fields (70%) and comprehension regarding the use of the keys "Next" and "Ok" in the keyboard (40%). The first case was approached with the inclusion of an explanation regarding the format of this field. The second case happened because users did not expect the complete explanation of the button functionality when they touched on such a button.

TK04 (Manually register an expense) - 40% of volunteers considered this activity as easy, 40% considered as a medium level and 20% as hard. The currency fields, whose format should be “0,00” were again the main source of problems. The interaction with Android Spinners (DropDown List) was also difficult because this kind of component is not common and they hide (stay over) other components. Thus, users became confused. Similar names between the screen title and buttons of this screen also generated problems because the screen titles are read but they are not interaction components. The lack of feedback about the conclusion of the filling process of all fields of the screen was also a point of critics.

TK05 (Edit a card) – 80% of volunteers considered this activity as intermediate and 20% as easy. Two interesting problems happened in this activity. First, one of the fields (“Type of the card”) of the form was not editable. Rather, uses needed to select a card from a list. As all other previous fields were editable, users were also expecting an editable field for this attribute. Second, the edition process of a card first required the selection of a card. However, this was not obvious to users.

TK06 (Register a user) – the difficulty level of this activity was considered as low and just the location (sublevel of the keyboard) of the key “@” and problems in understanding the phonetics of the screen reader to such special symbols were identified as negative points of this activity.

TK07 (Login) – this activity seems very simple, but 30% of users considered it as hard. Differently, 50% of users considered the activity level of difficulty as easy, while 20% considered such level as moderate. The main problem was the field password since it does not present feedback. Thus, users were not sure about what they have entered in this field. The use of security approaches for visually impaired users is a subject for discussion and requires a better analysis.

Passwords are not considered a good solution since users are not able to see who is around them. Thus, new solutions are required, such as the use of biometry.

TK08 (Register expense via SMS) – this activity was considered as easy. However, as an expense is automatically registered when an SMS is received; users need to know that such expense should be classified. This means users must include the type of expense. The first problem was the notification. An alert must be generated and the notification must be placed in an easy region to be selected. Two options could be implemented when users touch in this notification. First, users can be directly forwarded to this expense, so that it can complete its information. Second, they can be led to a screen where all incomplete expenses are listed. Then, an expense can be selected and its text fields presented on the screen. This process may seem complex at the first moment, but it was considered very straightforward by users.

TK09 (Access information from the history graph) – the level of difficulty of this activity was considered as easy by 80% of the volunteers, while 20% considered it as medium. The main problems were to find the required information among all available options and understand that the report screen was organized in two tabs. The reason for these problems was associated with the non-traditional way to present information via graphs, which are not common to this group of users. However, the majority enjoyed this resource as an option to present quantitative data.

TK 10 (Access information from the category graph) – the level of difficulty was considered low to this activity. This mainly happened because this activity was carried out after TK09. This fact shows that several of the problems reported by users are associated with the first use of the application when they are starting to know the facilities and available resources.

TK11 (Filter expenses by period) – the level of difficulty of this activity was considered as hard by 90% of users. The main reason was the concept of information filtering and context menu, which were new to several volunteers. Thus, they had problems to fill in the required fields, which are a set of search criteria. For example, a search constrained by a period requires the insertion of initial and final dates in the respective fields. However, this is not obvious if the screen structure cannot be seen. Users may better understand this and other new functions if the application provides audible examples of how to use them. This is a situation where we should have considered R42.

TK12 (Remove expenses already registered) – this activity was considered as easy. However, 40% of volunteers did not use the search function, which could accelerate the process. Differently, they manually looked for the specific expenses in the list. Again, this is a problem related to the low familiarity with the application. Like any other application, users will learn more about the resources offered as they explore the screens and get used to their functions.

6.2.4 Post-use interview

The next paragraphs summarize the results of the interviews, considering the questions (Q_1 *to* Q_{10}) introduced in Section 6.1.

Q_1: How do you feel using this application? Why? Have you considered a positive or negative experience? All users related their experiences as positive and comfortable. U_d, for example, said "at the beginning, I had some difficulties, but this is normal when the application is used for the first time". The accessibility was considered a very positive point and several volunteers are motivated to explore in more detail the application.

Q2: Considering a scale from 1 to 10, how could you evaluate your satisfaction in using this application, where 0 is completely unsatisfied and 10 completely satisfied? The average score assigned to *Controle Fácil* by volunteers was 9.00 on a scale from 0 to 10 (min=8.0, max=9.5).

Q3: How do you evaluate the accessibility of this application? Some of the answers to this question were: "I think the application is totally accessible. I do not know other applications that are so accessible"; "The accessibility is very good and the application gives several important feedbacks. However, it could give some other feedbacks that are also important, such as the current amount of characters that were inserted when the text field has a limit. For example, 3 of 5", "Good application. It is different from others that are exclusive to visually impaired users. For example, this one does not have the option to put labels on the buttons"; "It is very good. It could be perfect if the filtering process was easier to use". These answers were important because the volunteers have a larger experience with accessible applications and considered the *Controle Fácil* as the most accessible application they know.

Q4: Which are the best and worst features of this application? Half of the participants indicated the expense reports and graphs functionality as the most interesting and useful. Applications based on quantitative reports are not so common to visually impaired users because such information tends to be presented in the form of graphs. However, if good descriptions regarding these graphs are generated, this style of information delivery may be very useful. On the other hand, the filtering function had the majority of the critics. We consider that the problems with this function were caused because some volunteers were not used with such function and the application does not present a guide about this function. As many other future functions will also be new to visually impaired users, we need to investigate the best ways to provide these guides and facilitate the learning process of users.

Q5: Could you indicate the situations that you think the use of the application were easy? According to the volunteers, the quality of the explanations was essential to facilitate the use of menus, reports, and insertion of expenses. These comments were important because the definition of explanations is a subjective task that must look for short but semantically complete sentences.

Q_6*: Was there any situation where you had problems in using the application along with the tasks? Could you describe this situation?* Again, volunteers highlighted the filtering and search functions as the most problematic parts of the application.

Q_7: Do you think any aspect could be implemented differently? Which way? The majority of the volunteers answered negatively this question. However, one volunteer said that he was expecting the option of searching for keywords in the search option; while another was expecting to find the search function in the toolbar, so that it could be applied in several other screens. Other volunteers also suggested the voice search as an option. Furthermore, one of the volunteers was expecting an organization to expenses similar to a file system, where they could be organized according to their type or period. The filtering function is a resource to dynamically create such an organization in different ways. However, the results are not persistent. Rather, they are generated on the fly.

Q_8: Do you have any suggestions to improve the accessibility of this application? The main suggestions were: (1) adapt the keyboard to the current text field, such as, the case where we need to use the symbol "@" to enter an email;

(2) enable the color inversion to configure a black background with white texts; (3) a better assistance to the search function, which expresses the options of such search; (4) use of more shortcuts, which could be configured in the own application, and (5) implementation of some kind of memory so that the application could present the most used resources in a more direct and easier way.

Q_9: How do you evaluate the feedbacks provided by the TalkBack/Voice-Assistant along with the use of the application? The general evaluation was very positive. Only two main points were observed. First, the pronunciation of the content regarding email fields and the style of how the summary of cards (textual composition) was carried out. This means, how the information of the card was consolidated in a unique text.

Q_{10}: How do you evaluate the information provided by the text, which was read by the TalkBack/Voice-Assistant along with the use of the application? According to a volunteer, "Information is very good in this application. They are clear and this was one of the main advantages of this application". Other volunteers said that the information always correctly comes at the moment that they need it.

6.2.5 Suggestions for further requirements

The observation of the tasks and post-use interviews also resulted in a set of 17 new requirements that will be further investigated in future steps of our research. Table 12 lists such requirements, complementing the *F-code* numeration in Table 7.

Table 12: Further requirements that were identified along with the user-centered evaluation.

Type	Fcode	Requirement Description
		INTERFACE
SCREEN ORIENTATION	R04	Always offer the vertical screen orientation as an interface option.
TEXTUAL AND NUMERICAL DESCRIPTION	R10	Name non-textual components so that they can be understood independently of its context.
READ TEXT SIZE	R17	Evaluate the advantage of providing a long text when touching a component. This may be important to graphic or image components since their description requires more information to be understood.
PATTERNS	R18	Ensure an internal standard in the interface (for terminology, colors, components position, and other attributes), establishing consistency in all application screens.
	R19	Set the location of the components according to already known and established conventions (i.e.: the "Save" button in the upper right corner).
	R20	Keep the same pattern for Form components whenever possible.
		INTERACTION
KEYBOARD	R24	Choose keyboard with keys compatible with the context field. ("Comma" instead "point", on the numeric keyboard to Brazilian language).
	R25	Choose the keyboard with navigation keys appropriate to the next interaction elements. (Ex.: Enter Button "Next" when the next field is data entry).
FEEDBACK	R31	Give visual feedback about all user interactions (e.g. processing action)
		NAVIGATION
SHORTCUTS	R35	Provide navigation via shortcuts, if necessary (i.e.: button FAB).
FOCUS	R37	Enable focus-based navigation.
PROVIDE HELP	R40	When the user insistently touches (x times) on one device button that is not active during the app use, notify the user about its inactivity.
	R41	When the user gives double touch insistently (x times) on one app component that is not available for interaction, notify the user about its inactivity.
	R44	Provide search function in the app, especially in items localization tasks in endless scrolling lists.
		OTHERS
ACTIVATE DEVICE RESOURCES	R46	Inform and request user authorization to activate the device accessibility resources to the best use of the application functionality (when necessary).
	R47	If the app requested user authorization to activate any Smartphone resources during the app use, after using the application, the Smartphone must go back to the previous operating system configuration.
COMPATIBILITY	R51	The application must be compatible with the latest version of Android and the most used older versions.

7. CONCLUSION AND RESEARCH DIRECTIONS

The aim of creating mobile accessible applications to a broad set of users is a complex and demanding work, given the high variety of impairments. This current study focused only on visually impaired users and, ever considering this restriction, such a study is still providing several technical challenges. Like any other complex work, it must start with single steps and our main contribution was to create a concrete basis for further developments. An essential point in this work was to understand that different impairments create different accessibility issues. For example, blind people require the content of the screen, including visual images, to be converted into speech or Braille; partially sighted users may require the screen to be magnified or contrast increased, and color-blind users need alternative means of distinguishing objects. Current mobile applications do not provide the proper support to this group of users and we could attest that fact along with our experiments. One of the main causes is exactly the lack of a guideline to show the requirements that are important and should be considered.

Our work presented a complete roadmap that considered the theoretical analysis of the state of the art regarding requirements for mobile accessibility; the confirmation of the importance of these requirements by means of interviews and observations associated with user's interactions; and the use of the proposed requirements to implement a real application, which was validated by visually impaired users. All this work is going to result in both a concrete guideline and a roadmap for the definition of other accessibility guidelines. An effort in this direction is our ongoing work in carrying out the same steps to specify a guideline for motor-impaired users.

Controle Fácil will be the first real product to use the guideline along with its ongoing development. The application is not ready to be released to the market, but its current version was already evaluated and certified as high accessible along with our experiments with volunteers. The method of evaluation is also a contribution of this work and it can be used to evaluate the accessibility of any other application.

ACKNOWLEDGMENTS

The results presented in this paper have been developed as part of a collaborative project between Samsung Institute for Development of Informatics (Samsung/SIDI) and the Centre of Informatics at the Federal University of Pernambuco (CIn/UFPE), financed by Samsung Eletronica da Amazonia Ltda., under the auspices of the Brazilian Federal Law of Informatics no. 8248/91. Professor Fabio Q. B. da Silva holds a research grant from the Brazilian National Research Council (CNPq), process #314523/2009-0.

REFERENCES

1. ANANTO, B., SARI, R., HARWAHYU, R., 2011. Color Transformation for color Blind Compensation on Augmented Reality System. Proceedings of the International Conference on User Science and Engineering, pp. 129-134.
2. ANDROID DEVELOPERS, 2015. Accessibility Developer Checklist, http://developer.android.com/guide/topics/ui/accessibility/checklist.html
3. BBC, 2017. Mobile Accessibility Guidelines. At http://www.bbc.co.uk/guidelines/ futuremedia /accessibility/mobile
4. BECKMAN, S., BARRY, M., 2007. Innovation as a Learning Process: Embedding Design Thinking. California Management Review, 50(1):25-56.
5. BOYATZIS, R., 1998. Transforming qualitative information: Thematic analysis and code development. Thousand Oaks, CA: Sage
6. CARVAJAL, L., MORENO, A., SANCHEZ-SEGURA, M., SEFFAH, A., 2013. Usability through Software Design. IEEE Transactions on Software Engineering, 39(11):1582-1596.
7. CRABTREE, B., MILLER, W., 1999. A template approach to text analysis: Developing and using codebooks. In B. Crabtree & W. Miller (Eds.), Doing qualitative research (pp. 163-177.) Newbury Park, CA: Sage.
8. DANDEKAR, K., RAJU B. I., SRINIVASAN M. A., 2003. 3-D finite-element models of human and monkey fingertips to investigate the mechanics of tactile sense. Journal of Biomechanical Engineering, vol. 125, pp. 682-691.
9. FANUCCI, L., IACOPETTI, F., RONCELLA, R., 2011. A console interface for game accessibility to people with motor impairments. Proceedings of the IEEE International Conference on Consumer Electronics, pp. 206-210.
10. FUNKA, 2016. Guidelines for the development of accessible mobile interfaces. Available at https://www.funka.com/en/design-for-all/accessibility/
11. GERONAZZO, M., BEDIN, A., BRAYDA, L., CAMPUS, C., AVANZINI, F., 2016. Interactive spatial sonification for non-visual exploration of virtual maps. International Journal of Human-Computer Studies, Vol. 85, pp. 4-15.
12. GIAKOUMIS, D., VOTIS, K., TZOVARAS, D., LIKOTHANASSIS, S., HASSAPIS, G., 2010. Introducing accessibility in the Web services domain. Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology, pp. 18-22.
13. GRUSSENMEYER, W., FOLMER, E., 2017. Accessible Touchscreen Technology for People with Visual Impairments: A Survey. ACM Trans. Access. Comput. 9, 2, Article 6. iOS. 2015. Accessibility for Developers. https://developer.apple.com/accessibility/ios
14. KANE, S., WOBBROCK, J., LADNER, R., 2011. Usable gestures for blind people: understanding preference and performance. Proceedings of the Annual Conference on Human Factors in Computing Systems, pp. 413-422.
15. KHAN, S., TAHIR, M., RAZA, A., 2013. Usability Issues for Smartphone Users with Special Needs–Autism. Proceedings of the International Conference on Open Source Systems and Technologies, pp. 107-113.
16. LEE, T. Y., SMITH, A., SEPPI, K., ELMQVIST, N., BOYD-GRABER, J., FINDLATER, L., 2017. The human touch: How non-expert users perceive, interpret, and fix topic models. International Journal of Human-Computer Studies, vol. 105, pp. 28-42.
17. LESCAL, D., CARON, L., ROUAT, J., 2012. Neural visual objects enhancement for sensorial substitution from vision to audition. Proceedings of the 11th International Conference on Information Sciences, Signal Processing and their Applications, pp. 1074-1079.
18. MI, N., CAVUOTO, L., BENSON, K., SMITH-JACKSON, T., NUSSBAUM, M., 2014. A heuristic checklist for an

accessible smartphone interface design. Universal Access in the Information Society, 2014, 13(4):351–365.

19. MORENO, L., GONZÁLEZ-GARCÍA, M., MARTÍNEZ, P., GONZÁLEZ, Y., 2017. Checklist for Accessible Media Player Evaluation. In Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '17). ACM, New York, NY, USA, 367-368.

20. NIELSEN, J., 2000. Why You Only Need to Test with 5 Users. Research Report Nielsen Norman Group, [https://www.nngroup.com/articles/why-you-only-need-to-test-with-5users/]

21. SIEBRA, C., GOUVEIA, T., MACEDO, J., CORREIA, W., PENHA, M., SILVA, F., SANTOS, A., ANJOS, M., FLORENTIN, F., 2015a. Usability requirements for mobile accessibility: a study on the vision impairment. Proceedings of the 14th ACM International Conference on Mobile and Ubiquitous Multimedia (MUM '15), pp. 384-389.

22. SIEBRA, C., GOUVEIA, T., FILHO, A., CORREIA, W., PENHA, M., ANJOS, M., FLORENTIN, F., SILVA, F., SANTOS, A., 2015b. Usability for Accessibility: A Consolidation of Requirements for Mobile Applications. Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility (ASSETS '15), pp. 321-322.

23. SIEBRA, C., ANJOS, M., FLORENTIN, F., GOUVEIA, T., FILHO, A., CORREIA, W., PENHA, M., SILVA, F. Q. B., SANTOS, A. L. M., 2015c. Accessibility Devices for Mobile Interfaces Extensions: A Survey. In Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '15). ACM, New York, NY, USA, pp. 644-651.

24. SODNIK, J., JAKUS, G., TOMAŽIČ, S., 2011. Multiple spatial sounds in hierarchical menu navigation for visually impaired computer users. International Journal of HumanComputer Studies, 69(1–2): 100-112.

25. SOUSA, A., BARROSO, J., PAREDES, H., FERNANDES, H., FILIPE, V., 2015. Context-aware, Accessibility and Dynamic Adaptation of Mobile Interfaces in Business Environments. Procedia Computer Science, vol. 67, pp. 397–402.

26. TIDWELL, J., 2010. Designing Interfaces: Patterns for Effective Interaction Design, 2nd Edition. O'Reilly Media Publisher.

27. WOBBROCK, J. O., SHUAN K., KRZYSZTOF Z. GAJOS, S. H., FROEHLICH, J. 2011. Ability-based design: Concept, principles and examples. ACM Transactions on Accessible Computing 3(3): 9.

28. W3C. 2017. Web Accessibility Initiative (WAI), http://www.w3.org/WAI

29. W3C. 2015. Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile, http://www.w3.org/TR/mobile-accessibility-mapping

APPENDIX A: PRIMARY STUDIES SELECTED

[S01] AHMETOVIC, D., 2013. Smartphone-assisted mobility in urban environments for visually impaired users through computer vision and sensor fusion. IEEE 14th International Conference on Mobile Data Management (MDM), vol. 2, pp. 15-18.

[S02] AL MOURAD, B., KAMOUN, F., 2013. Accessibility evaluation of Dubai e-government websites: Findings and implications. Journal of E-Government Studies and Best Practices pp. 1-15.

[S03] BYERLEY, S. L., CHAMBERS, M. B., 2002. Accessibility and usability of Web-based library databases for nonvisual users. Library hi tech 20(2): 169-178.

[S04] COONIN, B., 2002. Establishing accessibility for e-journals: a suggested approach. Library hi tech, 20(2): 207-220.

[S05] DIM, N. K., REN, X., 2014. Designing motion gesture interfaces in mobile phones for blind people. Journal of Computer Science and Technology, 29(5): 812-824.

[S06] FARIA, J., LOPES, S., FERNANDES, H., MARTINS, P., BARROSO, J., 2010. Electronic white cane for blind people navigation assistance. In World Automation Congress (WAC), pp. 1-7.

[S07] GALLAGHER, T., et al., 2012. Indoor positioning system based on sensor fusion for the blind and visually impaired. International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 1-9.

[S08] GANZ, A., et al., 2012. Quantitative and qualitative evaluation of PERCEPT indoor navigation system for visually impaired users. Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 5815-5818.

[S09] HONK KONG OFFICER, 2017. Mobile Application Accessibility Handbook, Office of the Government Chief Information Officer.

[S10] KANE, S., K., JAYANT, C., WOBBROCK, J. O., LADNER, R. E., 2009. Freedom to roam: a study of mobile device adoption and accessibility for people with visual and motor disabilities. In Proceedings of the 11th international ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '09). ACM, New York, NY, USA, pp. 115-122.

[S11] KIM, J., et al., 2014. Mobile applications for assisting mobility for the visually impaired using IoT infrastructure. TRON Symposium (TRONSHOW), pp. 1-6.

[S12] LUDI, S., CANTER, A., ELLIS, L., & SHRESTHA, A., 2012. Requirements gathering for assistive technology that includes low vision and sighted users. First International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE), pp. 25-31.

[S13] LIN, Y. H., HSU, Y. C., KAO, Y. Y., 2012. A Smartphone-Based Chinese DAISY Digital Talking Book Player for Visually Impaired. Sixth International Conference on Genetic and Evolutionary Computing (ICGEC), pp. 296-299.

[S14] MCGOOKIN, D., BREWSTER, S., JIANG, W., 2008. Investigating touchscreen accessibility for people with visual impairments. In Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges (NordiCHI '08). ACM, New York, NY, USA, pp. 298-307.

[S15] Mi, N., CAVUOTO, L. A., BENSON, K., SMITH-JACKSON, T., NUSSBAUM, M. A., 2014. A heuristic checklist for an accessible smartphone interface design. Universal access in the information society, 13(4), 351-365.

[S16] MOTIWALLA, L. F., QIN, J. 2007. Enhancing mobile learning using speech recognition technologies: a case study. Eighth World Congress on Management of eBusiness.

[S17] SARAVANAN, M., VINEET, G., 2014. SWAPN: A dream tool for the visually challenged. IEEE Global Humanitarian Technology Conference-South Asia Satellite (GHTC-SAS), pp. 87-92.

[S18] SUN, M., et al., 2014. eButton: a wearable computer for health monitoring and personal assistance. 51st ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1-6.

[S19] WAITE, T., KIRKLEY, J., PENDLETON, R., TURNER, L., 2005. MUSEpad: Supporting information accessibility through mobile location-based technology. TechTrends, 49(3), 76-82.

[S20] WISE, E., et al., 2012. Indoor navigation for the blind and vision impaired: Where are we and where are we going. International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 1-7.

[S21] BONG WAY KIAT, BONG W., CHEN, W., 2015. Mobile Instant Messaging for the Elderly, Procedia Computer Science, Vol. 67, pp. 28-37.

[S22] BOSSINI, J., MORENO, L., 2014. Accessibility to Mobile Interfaces for Older People Procedia Computer Science, Vol. 27, pp. 57-66.

[S23] ADAMS, D., KURNIAWAN, S., 2014. A blind-friendly photography application for smartphones. SIGACCESS Access. Comput. 108, pp. 12-15.

[S24] FENG, C., 2016. Designing Wearable Mobile Device Controllers for Blind People: A Co-Design Approach. In Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '16). ACM, New York, NY, USA, pp. 341-342.

[S25] CORBETT, E. WEBER, A., 2016. What can I say?: addressing user experience challenges of a mobile voice user interface for accessibility. In Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16). ACM, New York, NY, USA, pp. 72-82.

[S26] ROSA, A., JUSTICE, D., 2016. WebReader: a screen reader for everyone, everywhere. In Proceedings of the 13th Web for All Conference (W4A '16). ACM, New York, NY, USA.

[S27] FERNANDES, H., FARIA, J., PAREDES, H., BARROSO, J., 2011. An integrated system for blind day-to-day life autonomy. In The proceedings of the 13th international ACM SIGACCESS conference on Computers and Accessibility (ASSETS '11). ACM, New York, NY, USA, pp. 225-226.

[S28] GötZELMANn, T., 2016. LucentMaps: 3D Printed Audiovisual Tactile Maps for Blind and Visually Impaired People. In Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '16). ACM, New York, NY, USA, pp. 81-90.

[S29] GUERREIRO, T., NICOLAU, H., JORGE, J., GONÇALVES, D., 2010. Towards accessible touch interfaces. In

Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility (ASSETS '10), ACM, New York, NY, USA, pp. 19-26.

[S30] KRAINZ, E., LIND, V., MOSER, W., DORNHOFER, M., 2016. Accessible way finding on mobile devices for different user groups. In Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '16). ACM, New York, NY, USA, pp. 799-806.

[S31] Mascetti, S., Bernareggi, C., Belotti. M., 2011. TypeInBraille: a braille-based typing application for touchscreen devices. In The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility (ASSETS '11). ACM, New York, NY, USA, pp. 295-296.

[S32] Mascetti, S., Gerino, A., Bernareggi, C., D'Acquisto, S., Ducci, M., Coughlan, J. 2017. JustPoint: Identifying Colors with a Natural User Interface. In Proceedings of the 19th International ACM

SIGACCESS Conference on Computers and Accessibility (ASSETS '17). ACM, NY, USA, pp. 329-330.

[S33] Oliveira, J., Guerreiro, T., Nicolau, H., Jorge, J., Gonçalves, D., 2011. Blind people and mobile touchbased text-entry: acknowledging the need for different flavors. In The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility (ASSETS '11). ACM, New York, NY, USA, pp. 179-186.

[S34] Panëels, S. A., Olmos, A., Blum, J. R., Cooperstock, J. R., 2013. Listen to it yourself!: evaluating usability of what's around me? for the blind. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13). ACM, New York, NY, USA, pp. 2107-2116.

[S35] Park, J., Jeong, B., Jeon, S., Han, S., Cho, J., Ko, J., 2015. Understanding Interactive Interface Design Requirements for the Visually Impaired. In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15). ACM, New York, USA, pp. 881-886.

[S36] Penev, A., Wong, R. K., 2008. Grouping hyperlinks for improved voice/mobile accessibility. In Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A) (W4A '08). ACM, New York, NY, USA, pp. 50-53.

[S37] Rituerto, A., Fusco, G., Coughlan, J. M., 2016. Towards a Sign-Based Indoor Navigation System for People with Visual Impairments. In Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '16). ACM, New York, NY, USA, pp. 287-288.

[S38] RodRIGUes, A., 2015. Breaking Barriers with Assistive Macros. In Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility (ASSETS '15). ACM, New York, NY, USA, pp. 351-352.

[S39]SATO, D., OH, U., NAITO, K., TAKAGI, H., KITANI, K., ASAKAWA, C., 2017. NavCog3: An Evaluation of a Smartphone-Based Blind Indoor Navigation Assistant with Semantic Features in a Large-Scale Environment. In Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '17). ACM, New York, NY, USA, pp. 270-279.

[S40] GUERREIRO, T., NICOLAU, H., JORGE, J., GONÇALVES, D., 2009. NavTap: a long term study with excluded blind users. In Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility (Assets '09). ACM, New York, NY, USA, pp. 99-106.

[S41] TOMLINSON, B. J., SCHUETT, J. H., SHORTRIDGE, W., CHANDRAN, J., WALKER, B. N., 2016. Talkin' about the weather: incorporating TalkBack functionality and sonifications for accessible app design. In Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16). ACM, New York, NY, USA, pp. 377-386.

[S42] CHIARA, G. et al., 2011. The Framy user interface for visually-impaired users. Sixth International Conference on Digital Information Management, Melbourn, QLD, 2011, pp. 36-41.

[S43] KALMAÇ, H., DIRI, B., 2015. moBraille: Mobile framework for visually impaired users. 23nd Signal Processing and Communications Applications Conference (SIU), Malatya, pp. 1477-1480.

[S44] PUENTE-MANSILLA, F., BOZA-QUISPE, G., LAPA-VELASQUEZ, G., MATOS-AVALOS, C., ROSALES-HUAMANI, J., 2016. A wearable UV sensor and accessible smartphone application for blind people. IEEE International Symposium on Consumer Electronics (ISCE), Sao Paulo, pp. 81-82.

[S45] SKULIMOWSKI, P., KORBEL, P., WAWRZYNIAK, P., 2014. POI explorer - A sonified mobile application aiding the visually impaired in urban navigation. Federated Conference on Computer Science and Information Systems, Warsaw, pp. 969-976.

[S46] ANANTO, B., SARI, R., HARWAHYU, R., 2011. Color Transformation for color Blind Compensation on Augmented Reality System. Proc. of the Int. Conference on User Science and Engineering, pp. 129-134.

APPENDIX B: *CONTROLE FÁCIL* APPLICATION

The next illustrations show some screenshots of the *Controle Fácil* application. From the top left corner (clockwise) we have the Welcome screen, which presents traditional functionalities of a login page (login via email, password, and option to recover this password). The second illustration shows the main menu of the Home screen, which is composed of nine functionalities (e.g. Cards, Expenses and Reports). The Cards screen shows all cards that are available to users. For each card, there is a card association (e.g. *Banco do Brasil*), card brand (e.g. Visa), four last card numbers (e.g. 5544), the value of the credit already used and the credit limit. The Cards screen has a menu, which contains functionalities such as "Insert Card" and "Edit Card". The functionality "Insert Card" is shown in the next illustration, which indicates the information that must be filled. A special configuration in this screen "Allow automatic insertion of expenses" indicates if users want to enable the automatic registration of expenses that are received via SMS. The Reports screen provides different views regarding the cards and expenses. The Categories functionality is presented in the next illustration. This view shows the total amount already expended and the categories where this amount was expended. Possible categories are supermarket, education, transports, and health. The last illustration shows the Expenses screen, which gives the individual a description and details of each expense.

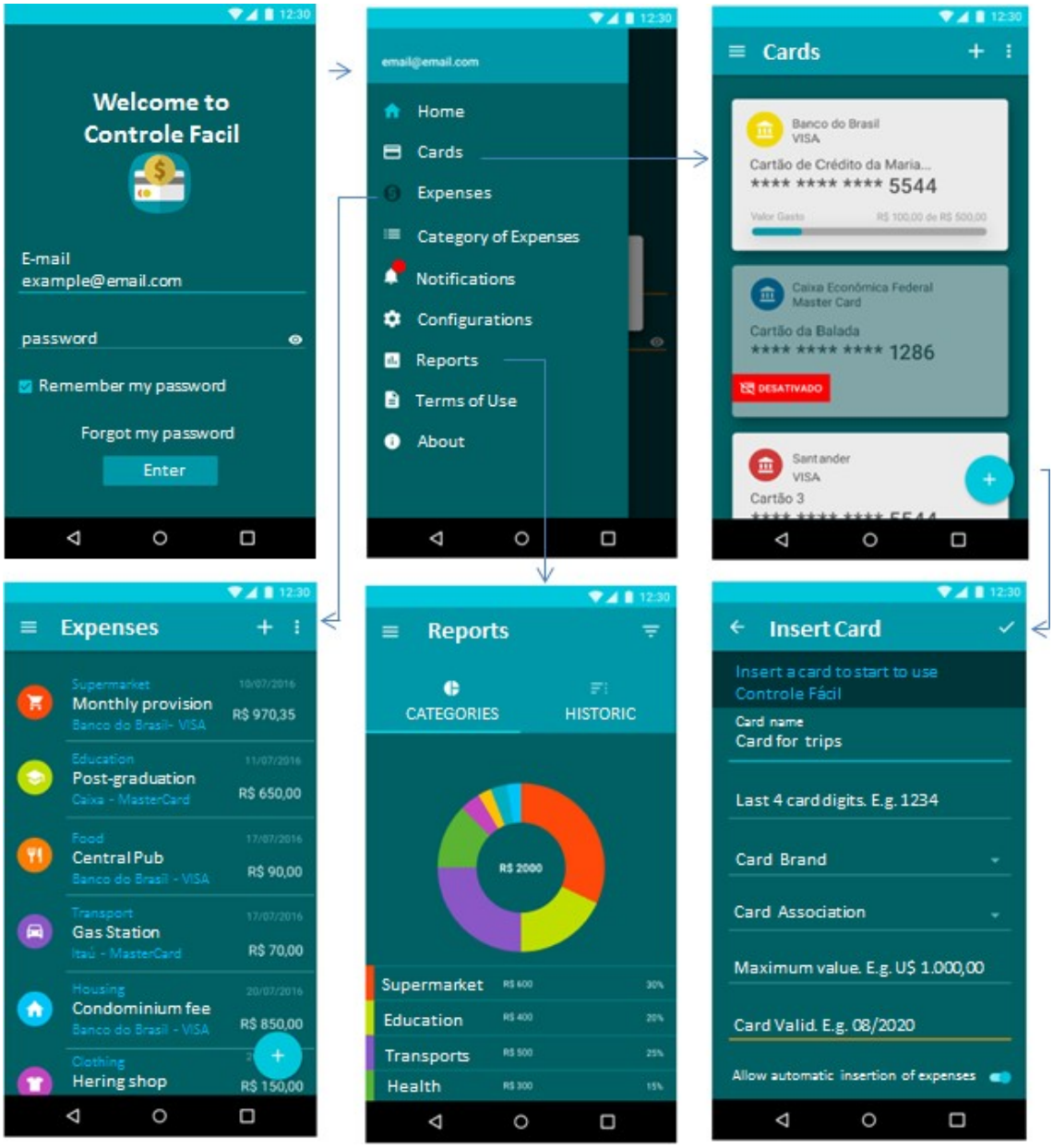


Figure 3: Screenshots of Controle Fácil app