

Research Article

A Systematic Evaluation of Mobile Applications for Instant Messaging on iOS Devices

Sergio Caro-Alvaro, Eva Garcia-Lopez, Antonio Garcia-Cabot, Luis de-Marcos, and Jose-Maria Gutierrez-Martinez

Computer Science Department, University of Alcala, Alcalá de Henares, Spain

Correspondence should be addressed to Sergio Caro-Alvaro; sergio.caro@uah.es

Received 15 June 2017; Revised 6 August 2017; Accepted 17 August 2017; Published 2 October 2017

Academic Editor: Porfirio Tramontana

Copyright © 2017 Sergio Caro-Alvaro et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nowadays, instant messaging applications (apps) are one of the most popular applications for mobile devices with millions of active users. However, mobile devices present hardware and software characteristics and limitations compared with personal computers. Hence, to address the usability issues of mobile apps, a specific methodology must be conducted. This paper shows the findings from a systematic analysis of these applications on iOS mobile platform that was conducted to identify some usability issues in mobile applications for instant messaging. The overall process includes a Keystroke-Level Modeling and a Mobile Heuristic Evaluation. In the same trend, we propose a set of guidelines for improving the usability of these apps. Based on our findings, this analysis will help in the future to create more effective mobile applications for instant messaging.

1. Introduction

Mobile devices have become an essential tool in our daily lives [1], to the point that the number of mobile users has been increasing more and more in recent years, from 640 million of Android and iOS active devices in 2012 [2] to 2,562 million of devices in 2016 [3]. So considerable is this increase that in the summer of 2016 Apple reported the sale of its billionth iPhone [4]. The increased use of mobile devices has led to a widespread diffusion of the number of applications (henceforth, called *apps*) available in mobile markets in recent years. For example, the Apple App Store reveals an increase in the number of available apps, from 1,200,000 (as of July 2014) to 2,200,000 (as of January 2017) [5, 6]. Among all those applications there are some for written communication that have become ubiquitous in contemporary society [7, 8], such as instant messaging (so called *IM apps*), social networks, or email apps.

IM apps are the newest and most popular evolution of near-synchronous text technologies [9–11]; thus, they are used to facilitate social relationships [12] and are one of the most widely used apps, since this type of app has become an

alternative (usually free or cheaper) to the traditional SMS messages [8, 10, 13]. Since there are so many apps for the same purpose in the mobile markets, users have too many alternatives to choose and therefore they are quite critical about how an app works. Users like simplicity to complete tasks and ease of learning or less time consumption [14–16], so they probably will choose the best app.

Mobile app developers find it difficult to determine, from target users, their needs and potential feedback in order to improve the apps [17, 18]. Thus, a good usability is the key to increase the chances of an app to be chosen by users among many others. Usability is a science that is responsible for studying the interaction between humans and computers (HCI) and has been defined by different organizations and researchers. International Organization for Standardization (ISO) defines usability as “the effectiveness, efficiency, and satisfaction with which specified users achieve specified goals in particular environments” [19] and Nielsen [20] defines it as “a quality attribute that measures the usability of user interfaces and the reference methods to improve ease of use during the design process.” A poor usability is the main discontinuing factor from app usage [11, 17, 21]; hence, it is

important to study the usability in desktop applications, but it is even more important to address it in mobile apps because mobile devices are the first and most used electronic devices in the world [22, 23] with a large number of users [24].

To this end, in this paper, we perform a systematic evaluation of mobile IM apps over the iOS platform to identify some usability issues. Particularly, given the lack of agreement on usability recommendations and difficulties to properly find them [25], this paper is to provide a list of recommendations for improving the usability of these applications, in order to be applied within the development process [25].

Mobile devices have some limitations when compared to personal computers (PCs) [26, 27], such as small-sized screens, limited input mechanisms, low and expensive bandwidth (in some cases), battery life, and a wide variety of devices (including the diversity of hardware and operating systems). Hence, in order to ensure an appropriate usability assessment, mobile devices should be evaluated separately from PCs.

In order to cater for the usability evaluation in existing mobile apps in the online markets, Martin et al. [28] proposed a mechanism for systematic evaluations of mobile apps, which has been successfully applied in different fields, such as diabetes [29, 30] or spreadsheet [31] mobile apps. This evaluation consists of five steps:

- (1) Identify all potentially relevant applications.
- (2) Remove light or old versions of each application.
- (3) Identify the primary operating functions and exclude all applications that do not offer this functionality. According to the agreed definition of usability, this step acts as a measure of effectiveness.
- (4) Identify all secondary functionality.
- (5) Construct tasks to test the main functionality using each of the methods below:
 - (a) Keystroke-Level Modeling (KLM) is used to estimate the time taken to complete each task to provide a measure of effectiveness of the applications [32, 33].
 - (b) Mobile Usability Heuristics (MUH) is used to identify more usability problems and measure user satisfaction.

As we discuss later in detail, this evaluation, considered as a laboratory experiment [34], has some advantages [28, 29], such as platform independence and flexibility [30]; that is, it can be applied on different platforms (e.g., Android, iOS, and Windows Phone).

Previous work applying this evaluation, including diabetes management apps [29, 30] and spreadsheet apps [31], showed that the combination of KLM and MUH allows detecting a larger number of usability problems than when performed independently. KLM results showed significant variations depending mainly on the input method of the app. The main issues detected were related to privacy and security, error management, aesthetics, and learnability.

The paper is organized as follows. Section 2 shows the evaluation carried out and the results obtained in the systematic evaluation. In Section 3, we provide some discussions of the results. Finally, Section 4 draws some conclusions, while presenting the usability recommendations.

2. Evaluation and Results

Here, after having outlined the corresponding need to address the usability issues of instant messaging apps in mobile devices, we show the systematic evaluation carried out and the different results obtained in the steps. To do so, the iOS platform was used for the evaluation and an iPhone 4 was used in the last steps because the evaluation requires testing the applications in a real mobile device.

Overview of the Process. It is important to underline that the systematic evaluation is comprised of five steps. Firstly, in Step 1, all potentially relevant apps are identified from the App Store (see Section 2.1). Consequently for the usability assessment, throughout Step 2, demos and old versions from the whole potential list of apps were discarded (see Section 2.2). A further aspect to be considered, in Step 3, concerns the identification of the main functionalities that characterize an IM app and, next, the exclusion of apps not offering these functionalities (see Section 2.3). With the aim of identifying secondary functionalities, in Step 4, all characterized IM apps are inspected (see Section 2.4). Finally, along with Step 5, the main functionalities are tested with two methodologies: the Keystroke-Level Modeling (KLM) to estimate time to complete tasks and the Mobile Usability Heuristics (MUH) to detect more usability problems with mobile experts (see Sections 2.5 and 2.6).

2.1. Step 1: Identify All Potentially Relevant Applications. To begin with the systematic evaluation, in this first step potential and relevant applications available in the iOS App Store [35] are identified. Indeed, these apps will be used as input in further steps of the evaluation. In order to handle the broad diversity of available apps, and given the lack of an IM category in the store, it was necessary to delimit the sample data applying a proper search term. To do so, it was required to analyze the specifications provided by both the most popular and commercial messaging applications, for example, “WhatsApp Messenger,” “Telegram Messenger,” and “LINE”. Thus, the “instant messaging” term was used to search in the app market. All the preliminary data was crawled from the store in the fall of 2014. As a result, 243 applications were classified as potential applications.

A further aspect to be taken into consideration concerns the rating of the apps by the users. To this end, each app in the iOS market can be rated from a range of 1 to 5 (with midpoints), where a higher number means a better rating. Likewise, users can write a comment about their experience with the app. It is important to highlight that most (51%) of the applications found in the market had no rating or comments by users (Figure 1). To our best knowledge, this may be motivated because they were not well known

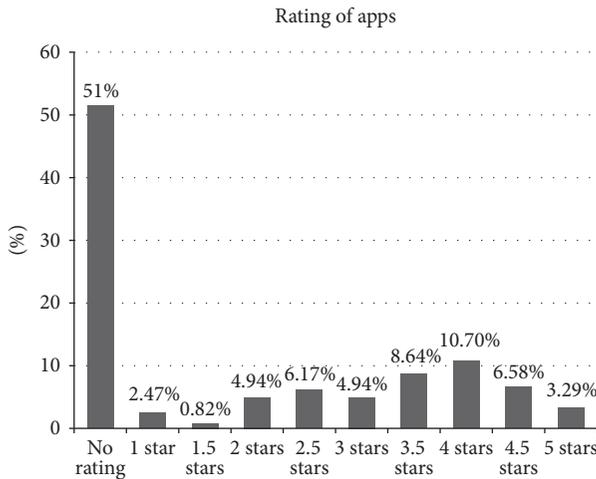


FIGURE 1: Rating provided by users on the applications found in the first step.

and therefore they did not have many downloads or maybe because the app did not have an option or a reminder to rate it.

2.2. Step 2: Remove Light or Old Versions of Each Application. In this step, the applications that were not fully functional were removed from the list of potential applications. Examples of such nonfully functional applications are demo, lite, or trial apps. The decision about which app to remove was based on the title and description of the app. Consequently, all apps' information sheets (i.e., the information provided in the App Store) were analyzed to determine whether or not an app was fully operational.

In all, 20 applications (8%) were removed from the initial list of 243 potential apps. At the end of this step, 223 (potential) applications remained as input for the next step.

2.3. Step 3: Identify the Primary Operating Functions and Exclude All Applications That Do Not Offer This Functionality. In this step, the main functionalities of this type of app were defined as the criteria against which to classify them. Briefly speaking, this characterization is applied to reduce the previous sample data as we only have apps devoted to this particular field, that is, Instant Messaging (IM). To this end, these definitions were made after analyzing the existing literature and reviewing the instant messaging applications discovered on the market from previous steps. Therefore, an application can be considered as instant messaging application when it meets all the main functionalities, which were defined as follows:

(i) Task 1 (T1): Send an Instant Message to a Specific Contact. According to Nardi et al. [36], in IM apps “there is a single individual with whom users communicate, although some IM systems support multiparty chat,” while Tomar and Kakkar [37] describe that “social messaging applications provide user with some very basic features like sending and receiving text messages,” and Cui [38] reported that “when

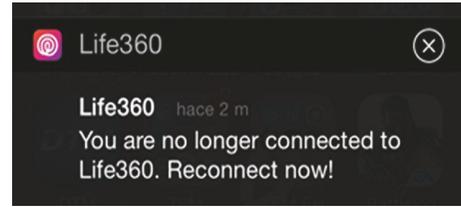


FIGURE 2: Example of continuous notifications from the Life360 app if the user disconnects manually from the system.

asked why they [IM users] began certain conversations at particular points, many interviewees could not give an answer other than <I suddenly wanted to do it.> [...] Spontaneous conversations covered a wide range of topics such as greetings, expressing moods, and sharing experiences, jokes, and gossip”; from this we could deduce that sending a message is an essential functionality.

(ii) Task 2 (T2): Read and Reply to an Incoming Message. Nardi et al. [36] say that in IM apps “the intended recipient may or may not answer,” while Tomar and Kakkar [37] describe that “social messaging applications provide user with some very basic features like sending and receiving text messages,” and Cui [38] added that in IM systems “general information exchange served the checking-in function. People would only be alerted when they had not received any response from the other party for a long time”; then we could conclude that the second functionality can be reading and replying to an incoming message.

(iii) Task 3 (T3): Add a Contact. Most IM applications provide information about the presence of other users [36], for example, showing a list of contacts, which users can use to choose the receiver of their messages. Moreover, IM communications are mainly established on purpose with specific people [38]. Therefore, the application should allow adding new contacts to that list, because otherwise the messages could only be sent always to the same people.

(iv) Task 4 (T4): Delete/Block a Contact. With respect to ensuring privacy [9] and security, the user should be able to delete internal contacts of the application. However, this is not always possible, for example, because the application accesses the internal agenda of the device. In this case, it should provide at least a blocking-contact feature [37, 39].

(v) Task 5 (T5): Delete Chats. Deleting messages or full conversations is also important because sometimes the users want to clean the text on the screen or just to save some memory space on their device [40, 41].

We would like to underline that, during the execution of this step, some usability issues were detected in the apps. As an example of this issues, if the session is closed (i.e., the user manually logs out from the system), the “Life360” app continually shows notifications to force the user to reconnect (Figure 2). Furthermore, if the app is recommended to the user’s friends (i.e., friends who are invited to use the app using an option of the app), it continuously sends emails to the

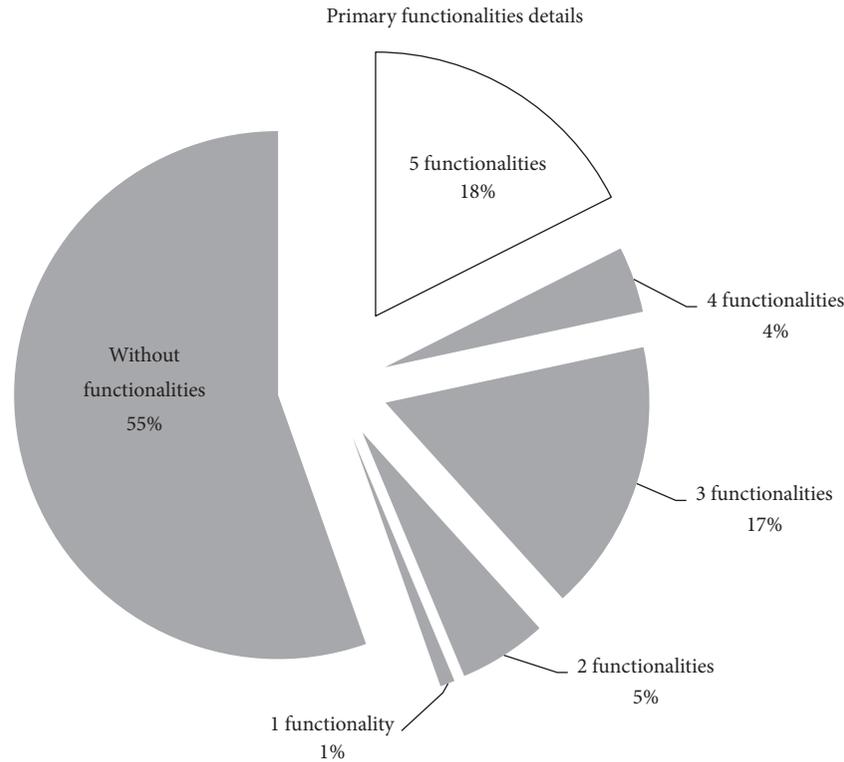


FIGURE 3: Percentages of the applications according to the number of main features met. In grey, discarded applications. In white, the applications that will be used in later steps.

friends until they accept or the user deletes the invitation. Along with this, an additional issue is found because those emails are detected as spam by most email service providers.

Once the main functionalities were detected and defined, the applications that did not meet all these requirements were discarded. Naturally, the availability of each of the aforementioned functions was checked on all the applications individually. As a result, only 39 (18%) applications met the five main functions to be considered as IM applications. However, it is important to note that some applications did not meet all the main functionalities but met three (17%) or four (4%) of them (Figure 3). The main reason is, to our best knowledge, that most applications in these groups were subapplications of social networks in which managing contacts was not possible; that is, the applications were not completely autonomous and they depended on another software.

At the end of this step, 39 (fully IM) applications remained as input for the next step.

2.4. Step 4: Identify All Secondary Functionality. In this step, the secondary functionalities of this type of app are identified. In short, we define secondary functionalities as all other functionalities apart from main functionalities defined in the previous step. It is important to bear in mind that, although this step does not remove applications from the evaluation, this identification of secondary functionality allows detecting

other additional functionalities of these applications, reinforcing the knowledge we could obtain, in this field, from IM apps.

Therefore, in this step, it was necessary to manually test all apps to discover all existing features. The 39 applications resulting after the previous step (Step 3) were installed on the iPhone device and they were tested in depth. Nevertheless, 11 of these applications were automatically discarded from the whole evaluation because they crashed and they run anomaly with critical errors. Hence, only 28 applications (Table 1) were completely examined for discovering the secondary functionalities of instant messaging in mobile devices.

Based on the findings, the most common secondary functionalities implemented in the analyzed applications were (1) including a user profile avatar, (2) sending pictures, (3) sending videos, and (4) group chats. All secondary functionalities and their level of use in the applications are depicted in Figure 4.

At the end of this step, 28 IM applications remained as input for the following step, where the usability tests begin and the usability assessment is performed.

2.5. Step 5(A): Keystroke-Level Modeling. This step is to provide the results of the KLM method on the remaining list of IM apps (i.e., 28 apps in total). Even though the apps were installed and tested in the previous step, at this time we have to ensure their accurate performance in the

TABLE 1: KLM results: number of interactions required for completing each main task.

App	Version	T1	T2	T3	T4	T5	Total
Surespot encrypted messenger	6.00	5	5	3	4	4	21
Hike Messenger	2.5.1	5	5	5	5	4	24
HushHushApp	1.0.3	6	6	4	4	4	24
Hiapp Messenger	1.0.6	6	6	5	5	4	26
Kik Messenger	7.2.1	5	5	5	6	5	26
Touch	3.4.4	7	5	5	5	4	26
WhatsApp Messenger	2.11.8	5	6	5	5	6	27
BBM	2.1.1.64	6	5	7	6	4	28
iTorChat	1.0	7	6	5	5	5	28
XMS	2.31	6	6	6	6	4	28
Hangouts	2.0.2	6	6	5	6	6	29
Tango Text, Voice & Video	3.8.84179	6	6	5	6	6	29
Viber	4.2.1	6	6	5	6	6	29
Google+	4.6.2	8	4	5	8	5	30
LINE	4.3.2	7	6	6	5	6	30
Mxit	7.2.0	7	5	7	6	5	30
Zappzter	2.1	6	6	7	5	6	30
FreePP	3.2.3.250	7	6	5	7	6	31
Telegram Messenger	2.1	6	6	5	8	6	31
IM+ Pro7	8.1.1	5	6	8	8	5	32
Skype	4.17.3	7	6	8	6	5	32
KakaoTalk Messenger	4.1.5	8	6	7	5	7	33
Snapchat	7.0.1	10	4	7	6	6	33
AppMe Chat Messenger	2.0.1	7	7	7	7	6	34
Tuenti	4.3	5	6	10	7	6	34
GroupMe	4.4.3	7	6	9	7	6	35
Keek	4.2.0	9	6	9	7	6	37
Spotbros	3.1.4	8	8	10	6	7	39
<i>Mean</i>		<i>6.54</i>	<i>5.75</i>	<i>6.25</i>	<i>5.96</i>	<i>5.36</i>	<i>29.85</i>

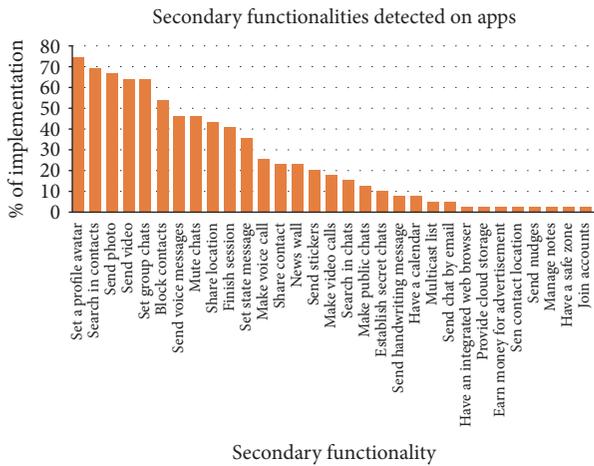


FIGURE 4: Secondary functionalities detected.

actual device (iPhone 4). How to leverage the KLM method is quite simple. To do so, we have to count the number of

interactions required to perform each of the main functionalities established previously (see Step 3). Such a procedure is explained as follows. We start counting once the app is opened (i.e., loaded). We have to count all interactions (e.g., screen touch, slide, scroll, and pushing hardware buttons) required to accomplish the given functionality. The counting procedure finishes as soon as the functionality is completed. Since KLM was originally designed to be used with computer elements (i.e., keyboard and mouse interactions), counting mobile interactions may be constrained by these previously defined interactions. To overcome this, this KLM evaluation applies mobile alike interactions (e.g., button press, tap, or swipe), similar to those defined in TLM [42], a KLM variant for touchscreen devices. It is worth highlighting that keyboard interactions (i.e., introducing letters) count as a unique interaction. By doing so, results are normalized and are easier to analyze. As seen in previous studies [43], this evaluation method is relevant because the interactions in a mobile device are, in some cases, limited and uncomfortable. Table 1 shows the results of the KLM analysis.

As a result of this analysis, it could be observed that the minimum number of interactions for completing all

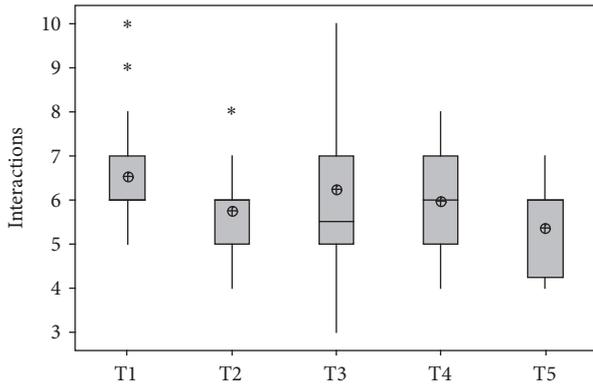


FIGURE 5: Boxplot with interactions per main functionality.

tasks (computed as aggregation) was 21 (in the form of “Surespot encrypted messenger” app) and the maximum was 39 interactions (“Spotbros” app). In this regard, this difference is because all tasks, individually, required more interactions in the latter than in the former.

Task 1: Sending a New Message. Briefly, the apps with fewer interactions (i.e., 5 interactions) in order to send a new message obtained these results by showing the keyboard automatically. This functionality is quite interesting, although only 6 apps (WhatsApp, Tuenti, IM+ Pro7, Hike, Surespot, and Kik) have implemented this feature. As seen in Figure 5, Keek (with 9 interactions) and Snapchat (with 10 interactions) are outliers. This could be produced because Keek requires recording a short video and Snapchat requires taking a photo (and, optionally, adding a message) to start a new chat.

Task 2: Reply an Incoming Message. Our measurements from examining the number of interactions when answering a given message show us that all analyzed apps required between 4 and 6 interactions (the exception was found in Spotbros, which required 8 interactions). Consequently, this similarity in the number of interactions is because almost all applications have a section in which the active chats are grouped. As aforementioned, Spotbros had atypical number of interactions because it required more navigation steps to answer a chat. This was because the individual chats were showed after a button was pressed to display the active chats.

Task 3: Adding a Contact. In order to address the results of this functionality, we assert that most variations, in number of required interactions, were observed in this task, as depicted in Figure 5. Thereby, we found apps from 3 interactions (Surespot encrypted messenger) to 10 interactions (Tuenti and Spotbros). To elaborate, this could be produced because some applications (specifically, 11 out of all apps analyzed) used the internal agenda of the mobile device, while other apps used their own contact list, causing alternative implementations of this process. Even if they used an alternative agenda/procedure, some applications required more information to register a contact than the average (some of them

only required a username, while others required a name and a phone number).

To go further on this point, we have observed that some applications required a very high number of keystrokes for adding a contact when we compared it with others or even when compared with the average value (Figure 5). Examples of this are Tuenti, Spotbros (both apps with 10 required interactions), or Keek (with 9 interactions). Particularly, these apps are characterized by being social networks in which public chats are common and, in addition, in which adding contacts becomes a poorly intuitive operation. To be more specific, GroupMe (with 9 interactions) only allowed adding contacts in the process of creating chat rooms (i.e., new contacts could be added when choosing the chat members). As a final note on adding a contact, we should highlight the applications with less number of interactions required to complete the task: Surespot encrypted messenger (with 3 interactions) and iTorChat and HushHushApp (both apps with 4 interactions). These apps achieve these lower values by only using usernames as required information to add a new contact. Thus, the process of adding contacts becomes a quite short task.

Task 4: Delete/Block a Contact. The results show that most of the analyzed applications required a number of interactions between 5 and 7 in order to delete/block a contact from the system. On the basis of the presented results, we may conclude that all apps used the same (or very similar) method to remove a contact from the internal system and, therefore, no major deviations were found.

Task 5: Delete a Chat. After the analysis of the results, most of the examined apps required from 4 to 6 keystrokes to complete this task. As we have already pointed out in the previous task, the reason of the similarities in the number of interactions is due to the widespread use of a similar method in the majority of the apps.

For the next step (i.e., the Heuristic Evaluation), not all apps were selected to continue in the process. To further understand this, as seen in previous studies of Flood et al. [29–31], the four applications with fewer interactions were selected for the next step. However, our KLM results present multiple apps with the same total number of interactions. In this case, applications with the same number of interactions were considered as one in order to avoid the challenge of choosing one from various apps. Hence, 7 applications (i.e., the 4 lowest values from the KLM results) were selected: Surespot (21 interactions), Hike Messenger and HushHushApp (24 interactions), Kik Messenger, Hiapp Messenger and Touch (the three with 26 interactions), and WhatsApp Messenger (27 interactions).

2.6. Step 5(B): Mobile Heuristic Evaluation. At this point, the final step of the evaluation, the mobile usability evaluation using heuristics was performed (i.e., the Mobile Heuristic Evaluation). To this end, six mobile experts carried out the evaluation with the 7 applications selected in the previous step (i.e., the KLM evaluation). Based on the previous experimental results of Bastien [44] and Hwang and Salvendy

TABLE 2: Results of Heuristic Evaluation on applications.

App	Mobile usability heuristics results								Total
	A	B	C	D	E	F	G	H	
WhatsApp Messenger	0.08	0.28	0.92	0.67	0.13	0.42	1.58	0.00	4.08
HushHushApp	0.92	0.61	0.67	1.00	0.23	1.17	0.50	0.39	5.48
Hiapp Messenger	0.00	1.17	1.67	1.42	0.70	0.50	1.25	0.89	7.59
Surespot	2.54	2.17	1.75	0.58	1.03	1.50	0.50	0.61	10.69
Kik Messenger	2.63	1.61	2.08	1.17	1.33	1.25	1.67	1.00	12.74
Touch	0.00	2.00	1.67	1.92	1.07	2.08	3.17	0.89	12.79
Hike Messenger	1.00	2.39	1.75	2.08	1.57	0.67	2.33	1.44	13.23
<i>Mean</i>	<i>1.02</i>	<i>1.46</i>	<i>1.50</i>	<i>1.26</i>	<i>0.87</i>	<i>1.08</i>	<i>1.57</i>	<i>0.75</i>	<i>9.51</i>

[45], 5 to 10 evaluators participating in the MHE evaluation are enough to detect, at least, 80% of the usability issues in a given software. As the common setting of the Mobile Heuristic Evaluation (MHE), which is described in detail in [26], this method is based on a study in which each mobile expert analyzes a series of usability guidelines for each application, which includes indicators about usability features of the application that the expert has to evaluate if the application incorporates it or not.

Briefly speaking, the participants of this step were characterized as follows: 6 evaluators (aged between 18 and 24 years with, at least, a Bachelor's Degree), all of them mobile experts with more than three years of background experience with smartphones and applications. It is worth highlighting that 3 of them stated that they had never used an iOS device and the rest said that they often used an iOS device.

Previous to heuristics evaluation, experts were asked to perform a series of tasks to familiarize themselves with the application. These tasks were as follows: add a contact, send a message to a contact, reply messages, delete conversations, and try to (re)configure the app, in other words, trying to become familiar with the interface and navigation of the app. It should be noted that the experts did not have problems or disagreements during the execution of the interview. Experts were also encouraged to comment out what were their in-use impressions. Also note that they were not time-limited to evaluate the apps.

The eight heuristics used were, as described in [26], the following: A (visibility of system status and losability/findability of the mobile device), B (match between system and the real world), C (consistency and mapping), D (good ergonomics and minimalist design), E (ease of input, screen readability, and glancability), F (flexibility, efficiency of use, and personalization), G (aesthetic, privacy, and social conventions), and H (realistic error management).

To go further on this point, the directives (i.e., the heuristics) to evaluate apps consist of eight sections (one section per heuristic), in which experts evaluate a number of indicators (also called subheuristics) about the usability of the application. Thus, the evaluators express their opinion within a numeric value from a range of 0 to 4, which is known as Nielsen's five-point Severity Ranking Scale [19], which is structured as follows: 0 to indicate that there is no problem, 1 to indicate a cosmetic problem, 2 to indicate a minor problem,

3 to indicate a major problem, and 4 to indicate a catastrophic problem. In addition, they also have to justify the scores by commenting their impression about each topic.

As an example, these are some of the subheuristics that were analyzed: "A4: the previously logged data and personal settings can be recovered if the device is lost," "B1: the information appears in a natural and logical order," "C2: there are not any objects on the interface that you would not expect to see," "D1: the screens are well designed and clear," "E2: it is easy to see what the information on each screen means," "F1: the user can personalize the system sufficiently," "G2: there are suitable provisions for security and privacy," or "H1: users can recover from errors easily," among others.

Table 2 shows the results of the Heuristic Evaluation. The values of a particular heuristic (each app' row) were obtained by averaging the user rating for all subheuristics. The last column on the right contains the aggregation of the values of all heuristics (calculated as the sum of the columns' value for each app), individually. As a result, in the bottom row, the average scores for each heuristic were calculated. Therefore, based on the results, the applications with lower values are better in terms of usability. It should be highlighted that applications with lower values (i.e., more usable applications) had mostly cosmetic problems or no problems. On the other hand, applications with higher values had mainly minor and major problems. Such problems affect the functionality of the application in a regular use, while other problems (cosmetic problems) mean errors involving small obstacles in the interface, which does not affect the regular use of the application.

Then, the way to interpret the results of Table 2 goes as follows. Take, for example, "WhatsApp Messenger" row and Heuristic "A" (i.e., visibility of system status and losability/findability of the mobile device) column, presenting 0.08 as a result (computed as the mean value from provided evaluations of the experts). As, here, lower is better; this value implies a quite positive result, nearly a 0 (i.e., indication of no problem detected). As another example, take "Touch" row (6th row) and Heuristic "G" (aesthetic, privacy, and social conventions) column presenting, as a result, a value of 3.17. This value implies a rather negative usability result, a little more than 3 (i.e., indication of a major problem).

There follows a review of the heuristics and the usability issues, based on the results that were obtained in our

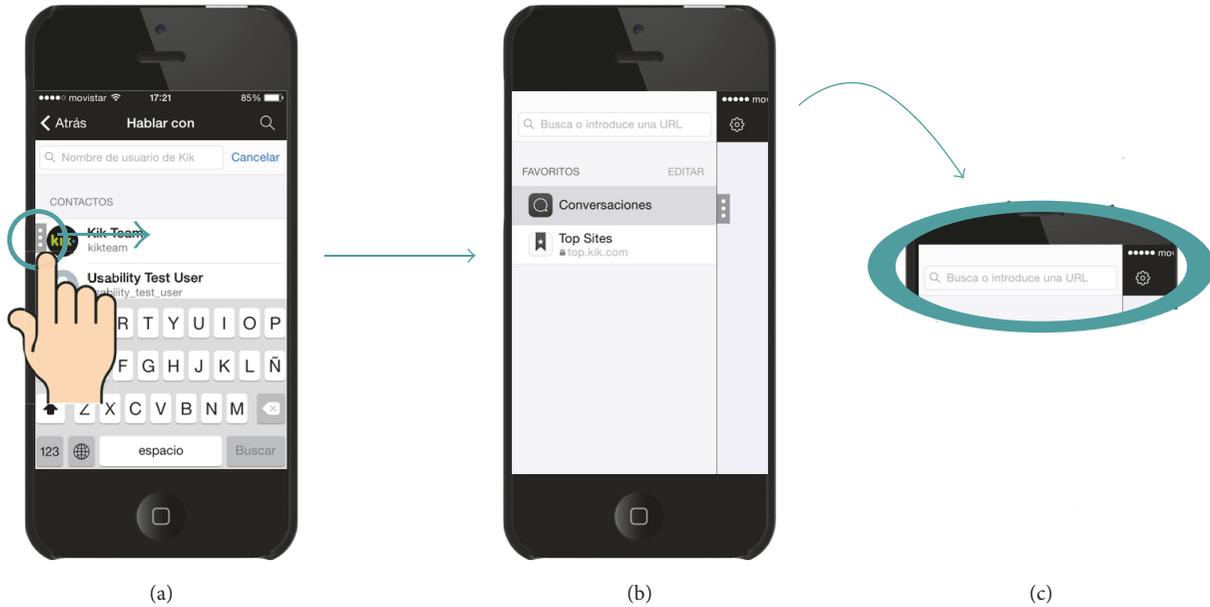


FIGURE 6: The side panel in Kik Messenger hides the top-bar.

evaluation. It is worthwhile to remark that the G and C heuristics got the worst usability assessments, mainly because of a poor UI design and security issues. In view of the results and the reports given by the experts, we propose some recommendations to cope with the usability problems that were found throughout the evaluation.

Heuristic A: Visibility of System Status and Losability/Findability of the Mobile Device. The results present few scenarios (only in 2 of the apps) where the status bar was not visible (Figures 6 and 7) and, additionally, where the app did not provide recovery methods in case of phone loss and using the app in another device.

Heuristic B: Match between System and the Real World. We found issues in which information and objects in the UI were not displayed clearly or in a natural order, such as clipped text elements (Figures 8(a) and 9).

Heuristic C: Consistency and Mapping. The results determined that this was the second worst heuristic, mainly because in some apps some objects were not expected on the interface (Figures 6(c), 8(b), and 10(b)) and because of some difficulties in performing the requested primary tasks, defined in early steps (Figures 10(a) and 10(b)).

Heuristic D: Good Ergonomics and Minimalist Design. Highly related to B heuristic, in this one the experts were asked for clean-design windows and lack of irrelevant information. The results exposed issues with clipped text elements due to (half) translation problems (if the app is not used in English) (Figure 8(c)).

Heuristic E: Ease of Input, Screen Readability, and Glancability. The results showed that this was the second best rated because

of an ease on entering numbers, an inclusion of a back button on the screens and (mainly) the ease on navigation through the screens.

Heuristic F: Flexibility, Efficiency of Use, and Personalization. With low values in the results (that is, good usability), we found issues related to a lack of personalization functionalities on some apps.

Heuristic G: Aesthetic, Privacy, and Social Conventions. The results showed that this was the worst heuristic on these apps. Indeed, this is the one that got the worst score, because of the (generally) bad interface designs (Figures 9, 10(a), 11, and 12). Also, a lack of privacy and information security of the user in almost all apps was reported (i.e., an in-app section where the system will tell the user about privacy policies applied to personal data and what security methods are applied to ensure message encryption over unsecure channels).

The authors would like to point out that, for example, in later versions of WhatsApp (compared with the version discussed here) the issue of security and privacy has been reduced due to the implementation of a visual message in each chat indicating that the application uses end-to-end encryption. This shows, as will be discussed later in the Discussion, that applications vary over time so the focus should be applied on problems in a generic way.

Heuristic H: Realistic Error Management. We would like to highlight that this heuristic got the best results, because of the ease on editing incorrect inputs and, also, ease on recovering from errors (when found, because not much errors were found).

It is important to underline that, based on the above results and previous steps, the reviews given by users (also known as rating) are slightly related to the conclusions

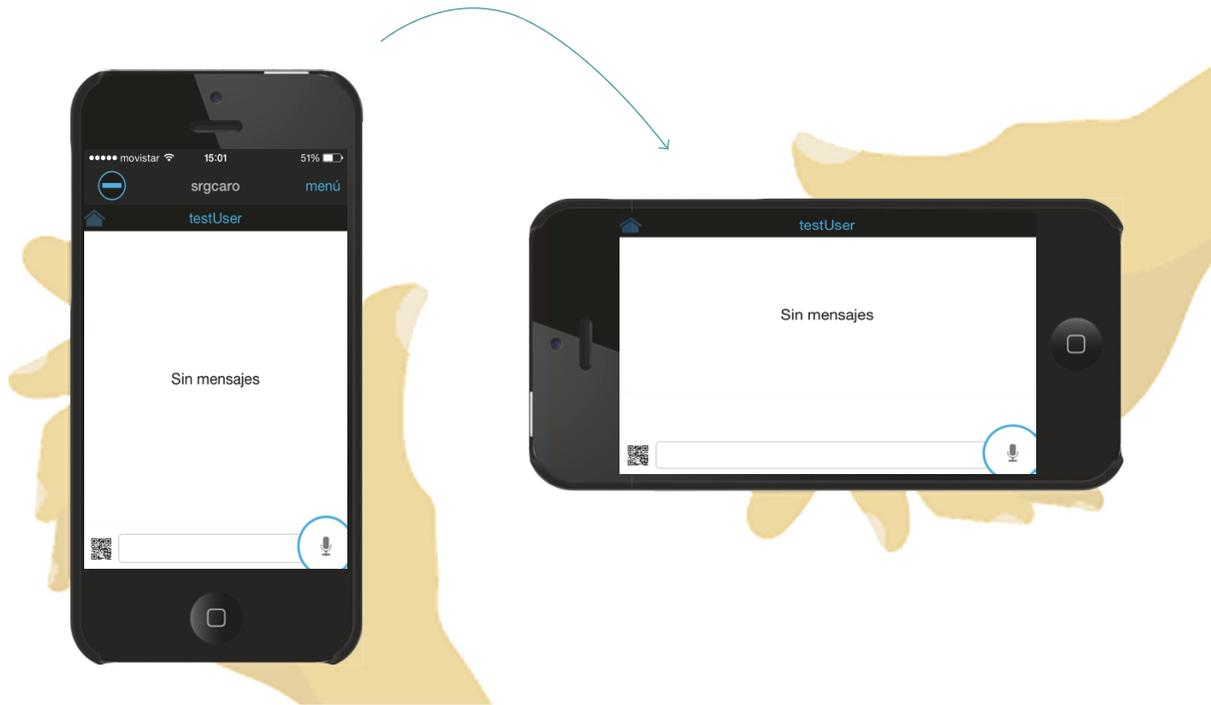


FIGURE 7: The options of Surespot disappear while the device is rotated.

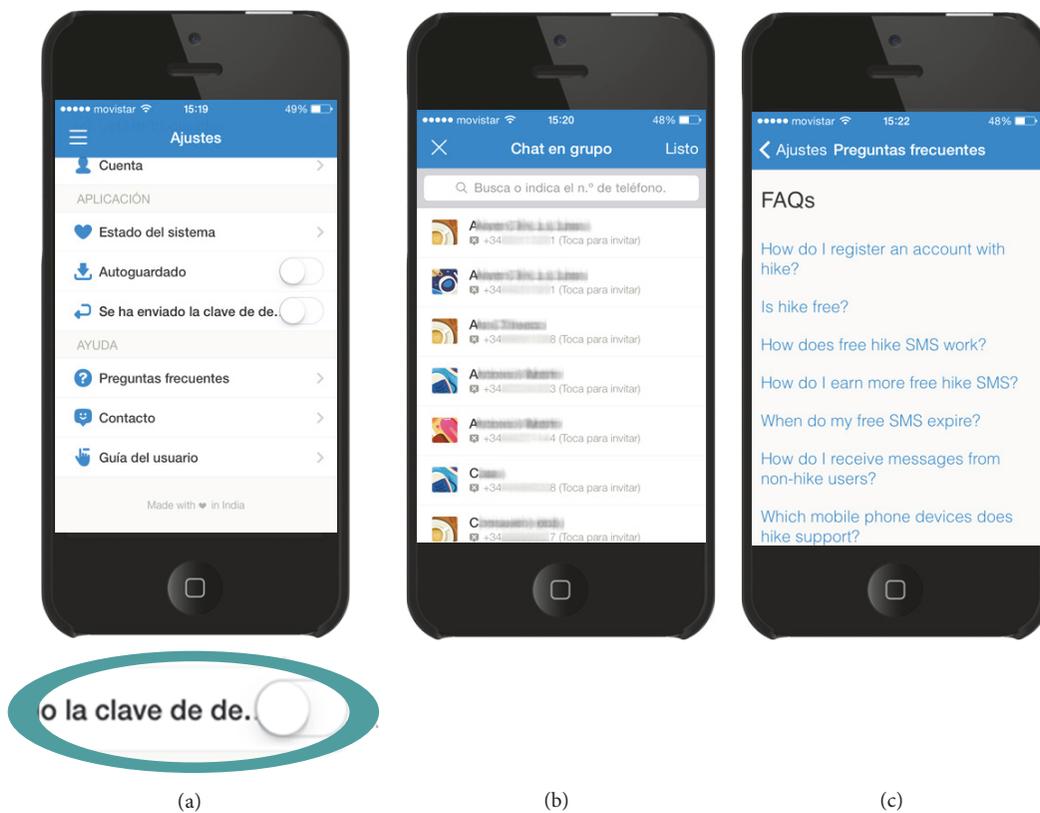


FIGURE 8: Hike clipped elements (a) and Hike contacts' avatars for all contacts, (b) and there are some parts not translated, when the app is not used in English (c).

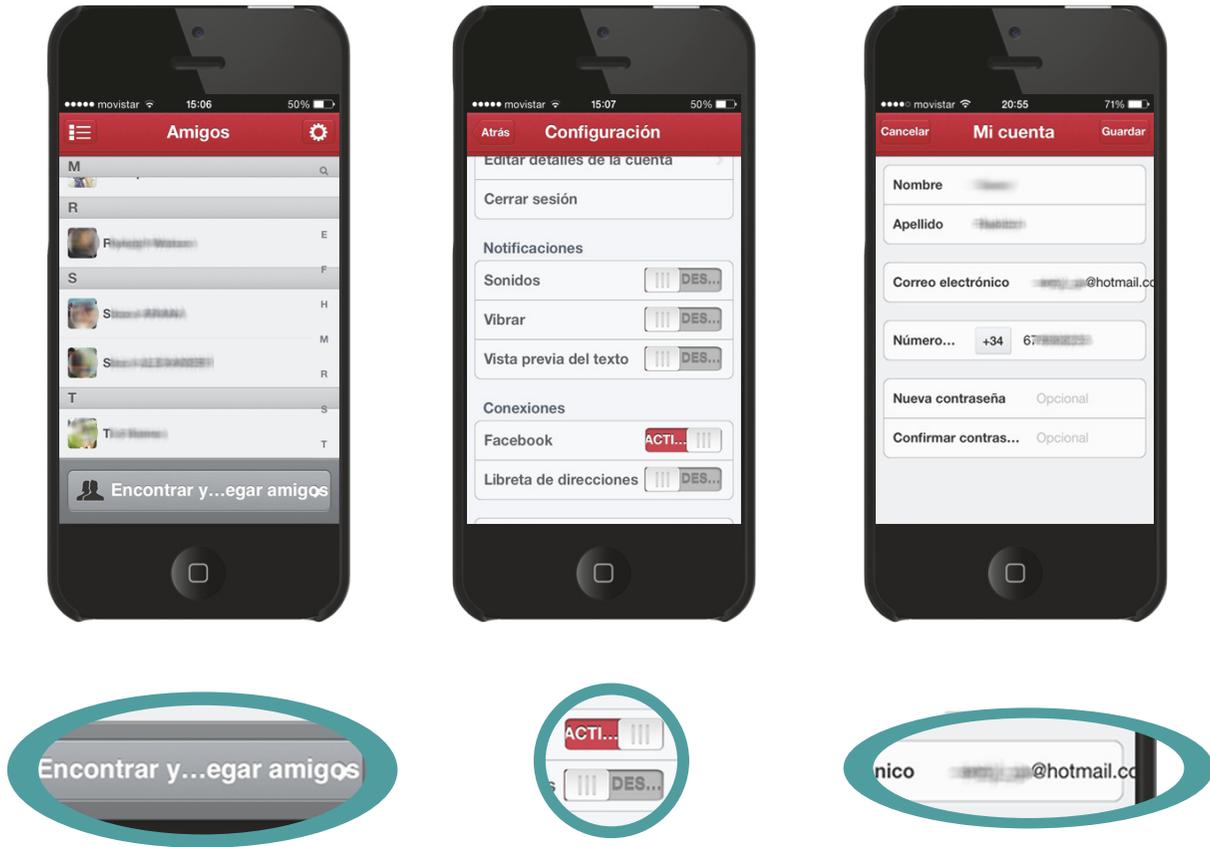


FIGURE 9: Clipped elements in the Touch app.

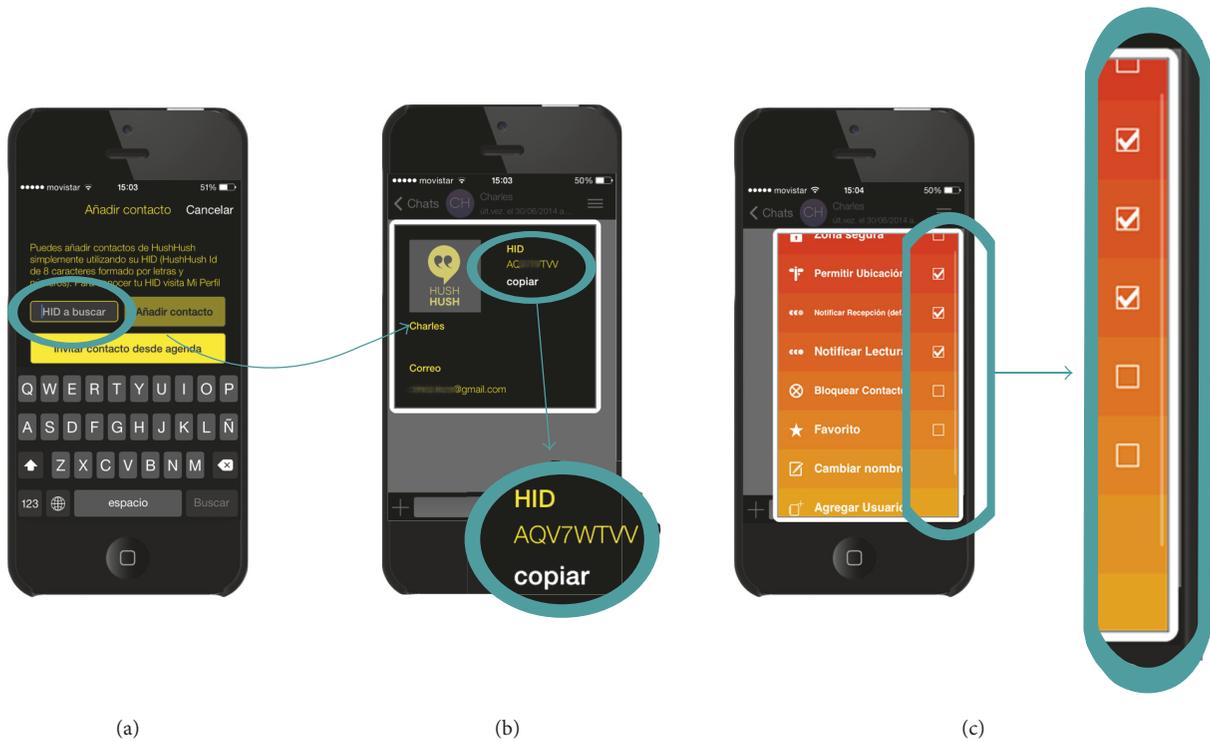


FIGURE 10: Difficulties in HushHushApp when adding a contact (a, b) and showing too many elements in the options button (c).



FIGURE 11: Options in Hiapp that remain hidden.

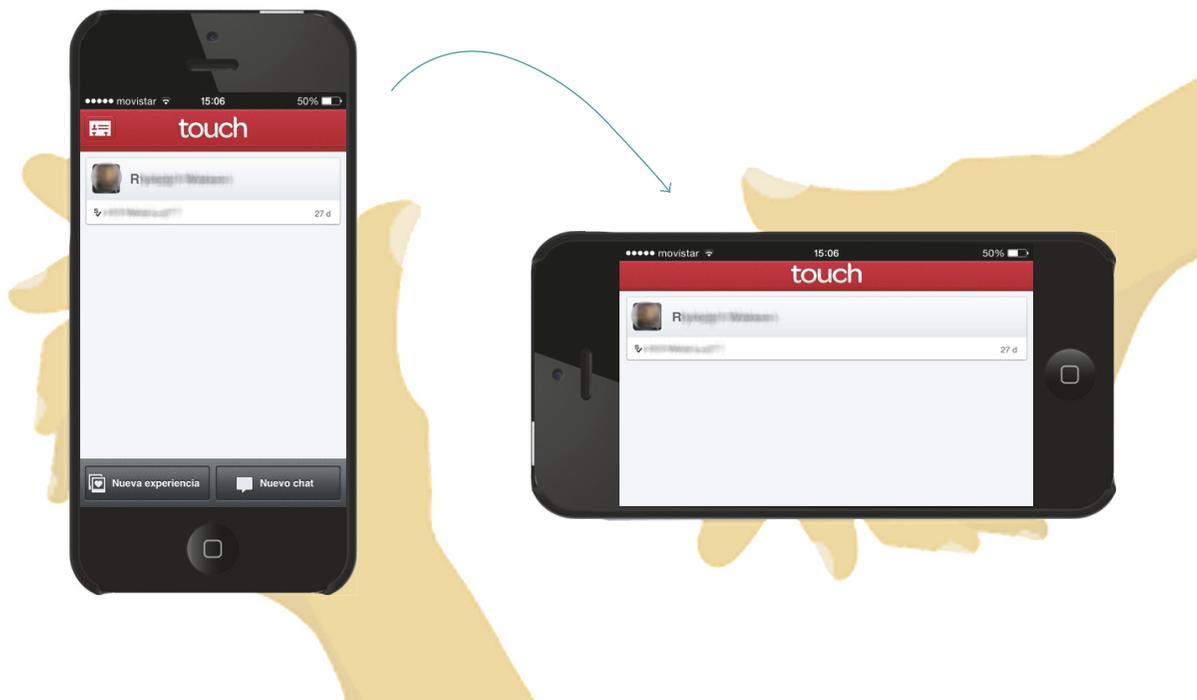


FIGURE 12: Some options are hidden when the device is rotated in the Touch app.

retrieved from the KLM and MHE analyses. Furthermore, HushHush App and Hiapp Messenger, both with 5/5-star rating in App Store by users, had the best results in KLM analysis and MHE test.

On the other hand, Touch and Hike Messenger, both with 0/5-star rating (i.e., nonrated), were the worst apps in the MHE study. Additionally, Surespot, with 0/5-star rating (i.e., nonrated), had a bad result (although it was not the worst) probably because, to our best knowledge, it is not a well-known app. In turn, WhatsApp Messenger, the app with the best result in MHE, had 3.5/5-star rating, which may be because it is a very popular app.

Therefore, in conclusion, it could be observed that it may cause a weak correlation between the issues detected in KLM and MHE analyses and the rating given by users in the market.

Finally, it is worth mentioning that a low number of interactions (i.e., KLM results) does not necessarily imply that the application has not usability problems. For example, Hike Messenger had 24 interactions and was the second best app on KLM results while, according to the MHE results, it was the worst app. Furthermore, this implies that both evaluations should be performed in order to categorize an application and address its usability.

3. Discussion

It is interesting to highlight that the need to use heuristics adapted to mobile devices derives from the fact that traditional methods are not intended to be used for touchscreen technology, since they were created for web and desktop applications [46–51]. Thus, they present features that are quite different from mobile devices, as we saw on previous sections. In consequence, this method is able to analyze the key factors of usability, factors discussed under the majority of usability studies: efficiency, effectiveness, and satisfaction [50, 52].

Specifically, this study was performed as a “laboratory evaluation,” a very common technique [51] which is widely applied (47% alone, and 10% in combination with field studies) [52] and, in addition, it is three times less cost consuming than field evaluations [53]. The downside of laboratory evaluations is its lack of context (in terms of real usage and environmental influences) [53, 54], only present in nearly 11% of the studies [52].

In this paper, we set out to address the usability issues of IM apps. Measuring and, hence, improving the usability of IM applications is indeed very important due to the high competition among applications and the very low switch-cost [10, 55]. Attracting users is not a difficult task but encouraging these users to continue using the application is, therefore, really complicated.

In consequence, several studies (2005 [56], 2008 [57], and 2012 [58]) have created IM prototypes for mobile devices. Their usability tests noted problems related to the Nielsen heuristic “*visibility of system status*” (quite similar to our heuristic A), with issues characterized by the need of the user to visualize availability indicators (user status) and message transmission indicators (delivery information). The last one [58] pointed out issues associated with a bad UI, but they

ensured that this was not a problem as their prototype was not in its final version, even though the problems identified by our study are related to the status bar (in some apps, under some circumstances, the status top-bar was not visible), in any case issues related to user status configuration.

Back in 2013, IM usability evaluation on Android devices was conducted [59]. This analysis was conducted using the Cognitive Walkthrough methodology in a laboratory environment with six evaluators. In the authors’ opinion, it was not possible to analyze all existing applications, so they used the “*PC Magazine*” website, in particular a review of the best Android applications (containing a mixture of all kinds of applications). Out of the five applications shown in the magazine, they chose the three most used apps by the users of the review (*WhatsApp*, *Skype*, and *GO SMS Pro*). As in our study, they chose main tasks to analyze but, unlike our study (tasks that characterize an IM application), they chose as tasks the functionalities/features offered by application vendors, with the clear disadvantage of not covering all possible dimensions of the application. These tasks were chatting, file transfer, contacts (adding, updating, and visualizing), and user status. Accordingly, usability problems detected by this study are inability to select multiple emoticons, lack of confirmation message when a file is sent, and inefficient “*search*” feature, among others. Besides, both studies match on the problem of button-icons which are similar to different functionalities that lead to user confusion. In overall terms, the usability problems are detected in the chat environment/section. However, if they had chosen other primary tasks, the problems identified could have been others or at least more diverse issues.

At this point, the results and analysis carried out will be discussed. Firstly, we could compare the results obtained with those from two similar studies performed using a similar method: one for spreadsheet apps [31] and another one for diabetes management apps [30]. It should be noted that the study of diabetes management apps [30] was carried on Android, iOS, and BlackBerry, but we will focus only on the results obtained for the iOS platform. Step 1 (potentially relevant applications) produced 23 spreadsheet apps and 231 diabetes apps and we found 243 IM apps. It would be expected that the number of apps found in the first step had increased because the previous studies were carried out some years ago and the number of apps in the iOS App Store is increasing every year, but surprisingly the apps found in IM apps was quite similar to the diabetes apps. However, too many diabetes apps found in this step were not really diabetes management apps but e-books and, therefore, they were discarded in the next steps. The low number of spreadsheet apps may be because the “*spreadsheet*” term used is more concrete and less confusing than the others used for diabetes and IM apps. Step 2 (delete light or old versions) discarded 9 (4,05%) diabetes apps and we discarded 20 (8,97%) IM apps. The analysis for spreadsheet apps does not indicate the number of discarded apps in Step 2. Based on the presented findings, the difference on the number of discarded apps (more than double) maybe because IM apps are more popular than diabetes apps and there are more evaluation versions.

Regarding Step 3 (identify main functionalities), discussed approaches got 12 (52.17% from Step 1) spreadsheet apps and 8 (3.46%) diabetes apps and we obtained 39 (16.04%) IM apps. This difference can be due, again, to the increasing number of applications in the iOS App Store in the last years but, also, it can be due to the main functionalities chosen, because they are different for each type of application. Specifically, it is important to highlight that 52.17% of spreadsheet apps in Step 1 remained in Step 3, which is a high number compared to the other apps. It can be seen that the “spreadsheet” term is much less confusing and more concrete than the others used for diabetes and IM apps.

It should be highlighted that those five tasks may not cover all possible dimensions of IM apps; therefore, other selections of main tasks could result in completely different results. Thus, we acknowledge this fact as the main limitation of our study.

Nevertheless, results in Step 4 (identify all secondary functionalities) cannot be compared because each app has different functionalities depending on its objectives. In turn, Step 5(A) (KLM analysis) revealed that spreadsheets apps had between 19 and 36 interactions in 9 tasks, diabetes apps had between 19 and 38 (except for an application that reached 50 interactions) in 6 tasks, and in our study IM apps had between 21 and 39 interactions in 5 tasks. Therefore, on average, the tasks in spreadsheets apps took between 2.1 and 4 interactions, in diabetes apps they took between 3.16 and 6.33 interactions, and in IM apps they took between 4.2 and 7.8 interactions. To our best knowledge, the differences in this case can be due to the complexity of tasks, which mostly depends on the type of application.

Regarding the MHE (Step 5(B)), the study over spreadsheet applications was not performed. For evaluation of diabetes management apps, the problems identified were between 0 (not a problem) and 2 (minor usability problem). In our study, the problems on IM apps were between 0 (not a problem) and 3 (major problem). In diabetes apps, the main usability issues detected were related to the heuristics G (aesthetic, privacy, and social conventions) and H (realistic error management), and related to IM apps the main usability problems were about heuristics B (match between system and the real world), C (consistency and mapping), and G (aesthetic, privacy, and social conventions). This suggests that issues related to heuristic G are the most common in all kinds of apps; that is, the design usually does not look good and/or there are no suitable provisions for security and privacy.

As previously mentioned, the method carried out has a number of advantages but it also has some disadvantages and limitations: for example, some steps take a long time and can be tedious to perform, there are a large number of elements in the initial stages, and results are time sensitive (due to the almost daily updates). Furthermore, detecting all usability issues in this kind of experiment is not possible because the context of use is not taken into account [27, 44] but, on the contrary, when applying experiments in the laboratory, there is more control over the usability issues detected.

While these results could be a greatly valuable resource, these results may not be fully generalizable because this study was conducted only on the iOS platform. Indeed, different

results could be obtained on other platforms, mainly because of other interaction methods. For example, Android devices usually have hard buttons, and BlackBerry devices have a physical keyboard and, in earlier versions, they did not have a touch screen. These differences would probably change both the KLM and MHE results.

As an explanation, the iOS platform was chosen (over the Android platform) to perform this study because, in iOS devices, user interface is quite similar on all models. On the other hand, on the Android platform, there are more variety of user interfaces, perhaps due to the wide variety of available brands and models.

As aforementioned, this method is time-dependent; that is, the evaluation has been performed on the apps available in the market at a given time (September 2014) and, hence, the emergence of new versions and/or applications could vary the results of this study. In addition, some applications could have been removed from the store; for example, in our study one application was detected in the early stages but disappeared when performing the fourth step.

4. Conclusion

As highlighted in this paper, the widespread propagation of mobile devices and instant messaging apps has led to addressing their usability issues differently from PC evaluations. To summarize, the methodology aimed at evaluating the usability of mobile apps is based on five steps: (1) identification of potentially relevant apps, (2) discard demos and old versions apps, (3) identification of main functionalities and exclusion of apps not offering all of these functionalities, (4) identification of secondary functionalities, and (5) construction of tasks to test the main functionalities: Keystroke-Level Modeling (KLM) to estimate time to complete tasks and Mobile Usability Heuristics (MUH) to detect more usability problems with mobile experts.

In this work, we presented a systematic evaluation to determine the main usability issues of instant messaging apps over iOS platform (an iPhone 4 device was operated for this study). In addition, these types of studies contribute as an awareness campaign for interface designers and companies, so they understand that what matters are user-based designs (continuous improvements and tests) to enhance the user experience.

Applying a preliminary research, Instant Messaging applications were characterized as applications that met all of these functionalities: (1) sending a message to a contact, (2) reading and replaying an incoming message, (3) adding a contact, (4) deleting (or blocking) a contact, and (5) deleting chats. Only 39 applications, retrieved from the App Store, met these features.

With our results of the KLM method, we were able to confirm that the functionalities of sending a message, replying to a message, and adding contacts were usually faster to complete. On the other hand, the functionalities of deleting a contact and deleting a chat became, usually, a serious problem in many apps because there were no clear indications of how to complete them.

Regarding the Mobile Heuristic Evaluation, a group of mobile experts were called for analyzing the low-score apps from KLM step. This evaluation consists of validating a series of usability directives (i.e., the *heuristics*) with a set of indicators per directive. In agreement with the results, almost all applications had usability problems in performing the primary functionalities, misplaced objects were detected on several interface screens, structuring content problems, similar buttons/icons with different functions, and deficient interface design, or did not provide enough information about safety and privacy, among other problems.

When comparing the results of both methods it is important to highlight that applications that have obtained high values in the Heuristic Evaluation (i.e., a bad result) did it well in the KLM, which suggests that it is necessary to perform both the KLM and Heuristic Evaluation because if the results were based only in KLM the applications chosen would be applications with many usability problems.

At this point, after finishing the study, we can propose some recommendations to improve the usability of instant messaging apps in mobile devices. All the proposed usability guidelines are a valuable resource for mobile app developers because they will improve the usability of their IM apps in mobile devices, thus achieving more downloads and users using their apps. Thus, the proposed recommendations are as follows:

(i) *Avoid Deep Navigation.* The app should require few interactions or, at least, not deep navigation between windows to complete a task. As seen on the best apps, each task should not exceed more than 5 or 6 interactions. In addition, experts interviewed acknowledged that with more than 8 interactions they found it difficult to follow, in a smooth way, the flow of events of a certain task.

(ii) *Keyboard Displayed Automatically at New Chats.* To reinforce this, when starting a new chat, the automatic display of the keyboard when the user accesses the chat screen is seen as highly positive. To ensure success, this procedure is more comfortable for the user and avoids interactions given that the chat remains empty of messages.

(iii) *Task Similarity When Sending and Replying to Messages.* For learnability purposes, sending a new message should not require more than double of interactions than replying to a message and vice versa. The aim is to make both tasks as similar as possible.

(iv) *Adding a Contact Only with the ID.* Specifying the ID of a contact (in the form of username, phone number, or email) should be enough for adding a contact. Other options (extra data such as name, last name, and location) should be optional and could be added, if any, afterwards.

(v) *Account Recovery.* As far as possible, the app should provide methods to restore the account if the user loses the device or migrates to another device. At least, users could recover their contacts.

(vi) *Visual Distinction between Individual and Group Chats.* Experts pointed out that it should be necessary to make a visual distinction between individual and group chats, for example, with multiple tabs or with an icon.

(vii) *Availability to Use the App Horizontally.* Experts pointed out that it was more comfortable (sometimes) to use the application with the phone horizontally located (to write messages more relaxed). Some apps, however, do not provide this functionality and allow only using the app with the device vertically located.

(viii) *Security Methods and Information to the User.* Since sensitive information travels over insecure channels [60, 61], privacy policies and security methods should be provided to make the app reliable and trusted. Moreover, given the lack of security knowledge of the users [60] and similar to security icons on mobile web-browsers [62], information about security and privacy methods used by the app should be provided to the user.

Along with these IM-specific recommendations we propose a list of recommendations that, although they have been defined for its use with IM applications, they can be generalized to any type of mobile application:

(i) *Primary Tasks Should Be Intuitive.* In order to be agile to achieve and be learnability friendly for the user, primary tasks should be intuitive; otherwise, indications should be provided on how to perform them. These indications, however, are sometimes necessary because some tasks are difficult to perform or even the user could not know that they can be performed. For example, deleting a chat or a contact just by sliding the item (without using/pressing any button) may be hard to perform without preliminary indications.

(ii) *Status Top-Bar Always Visible.* Try to avoid the use of sliding panels or pop-ups that hide totally the status top-bar (battery, time, and network indicators) while performing a task.

(iii) *UI Adapted to Host OS.* Given the fact that apps could be seen as a part of the operating system, it is especially relevant that icons have the same meaning as the host system itself, trying to avoid misunderstandings. As seen in [63], different OS/platforms require different UIs, that is, different approaches when designing the interface.

(iv) *Content Adapted to and Limited by the Screen-Size.* The content should be adapted and limited by the available space on the screen, which is not too big. As seen in [63, 64], the same app differs (in usability) depending on its host screen-size. Some apps (probably because they were designed in other languages) clip elements, making the content unreadable. As a solution, a recent study [65] shows the different techniques to distribute content on mobile devices.

(v) *Interface Centered Design.* The design of the application should pay close attention to the interface, even more than

the functionality itself. In words of [25], the UI design should not be done at the end. The app should be tested in every screen rotation (vertically and horizontally) and with different screen sizes to ensure that all elements of the app are properly displayed, the main problems described in [63].

(vi) *Avoid Half-Translations.* Try to avoid half-translations (i.e., mixed-language applications) because applications become difficult to use for people who are not fluent in several languages, while the UI changes when different-from-original languages are applied, as seen in [63].

(vii) *Testing on Target Devices.* Make sure that the application actually works for the target devices before publishing, especially for those paid apps, because users may have a fraudulent sensation after paying for an app that does not work or does not work properly. As seen in [63], different devices (for example, and iPhone and an iPad) create different environments.

(viii) *Do Not Tolerate Unrecoverable Errors.* Although it may seem trivial, do not tolerate unrecoverable errors. It may be seen as insignificant recommendation, but some of the tested apps presented critical errors. It is always better closing an error message than an unexpected shutdown of the app.

Thus, in the future, a new analysis will be carried out on other existing mobile platforms (e.g., Android or BlackBerry) to compare results between different platforms. Actually, we are in the process of evaluating the Android platform. Moreover, we would like to develop a mobile instant messaging application meeting the recommendations proposed, which will solve the main usability problems identified in existing applications and test the prototype with users.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

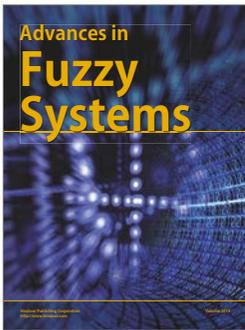
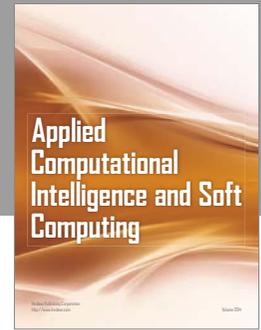
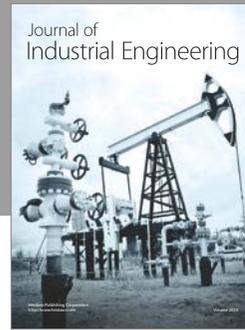
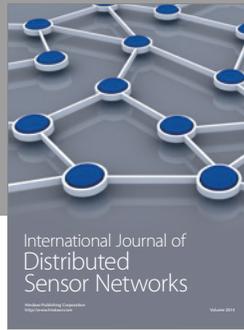
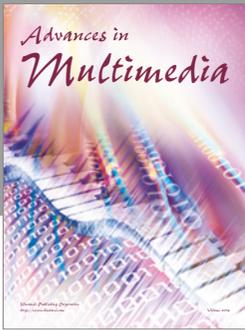
This research was funded by the FPU Research Staff Education Program of the “University of Alcala.” Also, the authors would like to thank the support of TIFYC and PMI research groups.

References

- [1] S. Gao, J. Krogstie, and K. Siau, “Adoption of mobile information services: an empirical study,” *Mobile Information Systems*, vol. 10, no. 2, pp. 147–171, 2014.
- [2] P. Farago, *iOS and Android Adoption Explodes Internationally*, The Flurry Blog, 2013.
- [3] A. Dutta, A. Puvvala, R. Roy, and P. Seetharaman, “Technology diffusion: Shift happens — The case of iOS and Android handsets,” *Technological Forecasting and Social Change*, vol. 118, pp. 28–43, 2017.
- [4] Apple-Inc., *Apple celebrates one billion iPhones*, 2016, <https://www.apple.com/newsroom/2016/07/apple-celebrates-one-billion-iphones/>.
- [5] Apple-Inc., *App Store shatters records on New Years Day*, 2017, <https://www.apple.com/newsroom/2017/01/app-store-shatters-records-on-new-years-day/>.
- [6] T. Statistics Portal, “Number of available apps in the Apple App Store from July 2008 to January 2017,” <https://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store/>.
- [7] Y. Bababekova, M. Rosenfield, J. E. Hue, and R. R. Huang, “Font size and viewing distance of handheld smart phones,” *Optometry and Vision Science*, vol. 88, no. 7, pp. 795–797, 2011.
- [8] Y. Sun, D. Liu, S. Chen, X. Wu, X. Shen, and X. Zhang, “Understanding users’ switching behavior of mobile instant messaging applications: An empirical study from the perspective of push-pull-mooring framework,” *Computers in Human Behavior*, vol. 75, pp. 727–738, 2017.
- [9] R. E. Grinter and L. Palen, “Instant messaging in teen life,” in *Proceedings of the The eight Conference on Computer Supported Cooperative Work (CSCW 2002)*, pp. 21–30, New Orleans, Louisiana, USA, November 2002.
- [10] A. P. Oghuma, C. F. Libaque-Saenz, S. F. Wong, and Y. Chang, “An expectation-confirmation model of continuance intention to use mobile instant messaging,” *Telematics and Informatics*, vol. 33, no. 1, pp. 34–47, 2016.
- [11] A. De Luca, “Expert and Non-Expert Attitudes towards (Secure) Instant Messaging,” *Symposium on Usable Privacy and Security (SOUPS)*, 2016.
- [12] A. Quan-Haase and A. L. Young, “Uses and gratifications of social media: A comparison of Facebook and instant messaging. Bulletin of Science,” *Technology Society*, vol. 30, no. 5, pp. 350–361, 2010.
- [13] K. Church and R. De Oliveira, “What’s up with whatsapp?: comparing mobile instant messaging behaviors with traditional SMS,” in *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '13)*, pp. 352–361, Munich, Germany, August 2013.
- [14] F. Nayebi, J.-M. Desharnais, and A. Abran, “The state of the art of mobile application usability evaluation,” in *Proceedings of the 2012 25th IEEE Canadian Conference on Electrical and Computer Engineering, CCECE 2012*, Montreal, Canada, May 2012.
- [15] N. Tillmann, M. Moskal, J. De Halleux, M. Fahndrich, and S. Burckhardt, “TouchDevelop - App development on mobile devices,” in *Proceedings of the 20th ACM SIGSOFT International Symposium on the Foundations of Software Engineering, FSE 2012*, USA, November 2012.
- [16] T. Petsas, A. Papadogiannakis, M. Polychronakis, E. P. Markatos, and T. Karagiannis, “Rise of the planet of the apps: A systematic study of the mobile app ecosystem,” in *Proceedings of the 13th ACM Internet Measurement Conference, IMC 2013*, pp. 277–290, ESP, October 2013.
- [17] S. L. Lim, P. J. Bentley, N. Kanakam, F. Ishikawa, and S. Honiden, “Investigating country differences in mobile app user behavior and challenges for software engineering,” *IEEE Transactions on Software Engineering*, vol. 41, no. 1, pp. 40–64, 2015.
- [18] K.-W. Hsu, “Efficiently and Effectively Mining Time-Constrained Sequential Patterns of Smartphone Application Usage,” *Mobile Information Systems*, vol. 2017, Article ID 3689309, 2017.
- [19] E. Din, “Ergonomic requirements for office work with visual display terminals (VDTs)—Part 11: Guidance on usability, International Organization for Standardization,” 1998.
- [20] J. Nielsen, *Usability 101: Introduction to usability*, 2003.

- [21] L. Carvajal, A. M. Moreno, M.-I. Sánchez-Segura, and A. Seffah, "Usability through software design," *IEEE Transactions on Software Engineering*, vol. 39, no. 11, pp. 1582–1596, 2013.
- [22] M. L. Smith, R. Spence, and A. T. Rashid, "Mobile phones and expanding human capabilities," in *Proceedings of the Information Technologies International Development*, vol. 7, pp. 77–88, 2011.
- [23] H. Kim, J. Kim, Y. Lee, M. Chae, and Y. Choi, "An empirical study of the use contexts and usability problems in mobile Internet," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences, HICSS 2002*, pp. 1767–1776, USA, January 2002.
- [24] R. Gafni, "Quality metrics for PDA-based m-learning information systems," *Interdisciplinary Journal of E-Learning and Learning Objects*, vol. 5, no. 1, pp. 359–378, 2009.
- [25] N. Juristo, A. M. Moreno, and M.-I. Sanchez-Segura, "Analysing the impact of usability on software design," *Journal of Systems and Software*, vol. 80, no. 9, pp. 1506–1516, 2007.
- [26] E. Bertini, T. Catarci, A. Dix, S. Gabrielli, S. Kimani, and G. Santucci, "Appropriating Heuristic Evaluation for Mobile Computing," *International Journal of Mobile Human Computer Interaction (IJMHCI)*, vol. 1, no. 1, pp. 20–41, 2009.
- [27] D. Zhang and B. Adipat, "Challenges, methodologies, and issues in the usability testing of mobile applications," *International Journal of Human-Computer Interaction*, vol. 18, no. 3, pp. 293–308, 2005.
- [28] C. Martin, D. Flood, and R. Harrison, "A Protocol for Evaluating Mobile Applications," in *Proceedings of the IADIS International Conference on Interfaces and Human Computer Interaction*, Rome, Italy, 2011.
- [29] C. Martin, D. Flood, D. Sutton, A. Aldea, R. Harrison, and M. Waite, "A systematic evaluation of mobile applications for diabetes management," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6949, no. 4, pp. 466–469, 2011.
- [30] E. Garcia et al., "Systematic Analysis of Mobile Diabetes Management Applications on Different Platforms," *Information Quality in e-Health*, pp. 379–396, 2011.
- [31] D. Flood, "A systematic evaluation of mobile spreadsheet apps," in *Proceedings of the in IADIS International Conference Interfaces and Human Computer Interaction*, Rome, Italy, 2011.
- [32] E. Abdulin, "Using the keystroke-level model for designing user interface on middle-sized touch screens," in *Proceedings of the 29th Annual CHI Conference on Human Factors in Computing Systems, CHI 2011*, pp. 673–686, CAN, May 2011.
- [33] D. Kieras, *Using the keystroke-level model to estimate execution times*, University of Michigan, 2001.
- [34] T. Kallio and A. Kaikkonen, "Usability testing of mobile applications: A comparison between laboratory and field testing," *Journal of Usability studies*, vol. 1, pp. 23–28, 2005.
- [35] Apple-Inc, "Apple-Inc. Store, iTunes Store," <http://itunes.apple.com>.
- [36] B. A. Nardi, S. Whittaker, and E. Bradner, "Interaction and outeraction: Instant messaging in action," in *Proceedings of the ACM 2000 Conference on Computer Supported Cooperative Work*, pp. 79–88, ACM, Philadelphia, USA, December 2000.
- [37] A. Tomar and A. Kakkar, "Maturity model for features of social messaging applications," in *Proceedings of the 2014 3rd International Conference on Reliability, Infocom Technologies and Optimization, ICRITO 2014*, India, October 2014.
- [38] D. Cui, "Beyond 'connected presence': Multimedia mobile instant messaging in close relationship management," *Mobile Media and Communication*, vol. 4, no. 1, pp. 19–36, 2016.
- [39] C. Lewis and B. Fabos, "Instant messaging, literacies, and social identities," *Reading Research Quarterly*, vol. 40, no. 4, pp. 470–501, 2005.
- [40] B. S. Gerber, M. R. Stolley, A. L. Thompson, L. K. Sharp, and M. L. Fitzgibbon, "Mobile phone text messaging to promote healthy behaviors and weight loss maintenance: a feasibility study," *Health Informatics Journal*, vol. 15, no. 1, pp. 17–25, 2009.
- [41] M. Rost, C. Kitsos, A. Morgan et al., "Forget-me-not: History-less mobile messaging," in *Proceedings of the 34th Annual Conference on Human Factors in Computing Systems, CHI 2016*, pp. 1904–1908, ACM, Santa Clara, California, USA, May 2016.
- [42] A. D. Rice and J. W. Lartigue, "Touch-Level Model (TLM): Evolving KLM-GOMS for touchscreen and mobile devices," in *Proceedings of the 2014 ACM Southeast Regional Conference, ACM SE 2014*, Kennesaw, Georgia, USA, March 2014.
- [43] K. B. Perry and P. Hourcade, "Evaluating one handed thumb tapping on mobile touchscreen devices," in *Proceedings of the Graphics Interface 2008*, pp. 57–64, Canada, May 2008.
- [44] J. M. C. Bastien, "Usability testing: a review of some methodological and technical aspects of the method," *International Journal of Medical Informatics*, vol. 79, no. 4, pp. e18–e23, 2010.
- [45] W. Hwang and G. Salvendy, "Number of people required for usability evaluation: The 10±2 rule," *Communications of the ACM*, vol. 53, no. 5, pp. 130–133, 2010.
- [46] R. Inostroza, C. Rusu, S. Roncagliolo, C. Jiménez, and V. Rusu, "Usability heuristics for touchscreen-based mobile devices," in *Proceedings of the 9th International Conference on Information Technology (ITNG '12)*, pp. 662–667, Las Vegas, Nev, USA, April 2012.
- [47] A. De Lima Salgado and A. P. Freire, "Heuristic evaluation of mobile usability: A mapping study," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8512, no. 3, pp. 178–188, 2014.
- [48] G. Joyce and M. Lilley, "Towards the development of usability heuristics for native smartphone mobile applications," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8517, no. 1, pp. 465–474, 2014.
- [49] R. Y. Gómez, D. C. Caballero, and J.-L. Sevillano, "Heuristic Evaluation on Mobile Interfaces: A New Checklist," *Scientific World Journal*, vol. 2014, Article ID 434326, 2014.
- [50] R. Baharuddin, D. Singh, and R. Razali, "Usability dimensions for mobile applications—A review," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 5, pp. 2225–2231, 2013.
- [51] O. Machado Neto and M. D. G. Pimentel, "Heuristics for the assessment of interfaces of mobile devices," in *Proceedings of the 19th Brazilian Symposium on Multimedia and the Web, WebMedia 2013*, pp. 93–96, Brazil, November 2013.
- [52] C. K. Coursaris and D. J. Kim, "A meta-analytical review of empirical mobile usability studies," *Journal of usability studies*, vol. 6, no. 3, pp. 117–171, 2011.
- [53] G. Fiotakis, D. Raptis, and N. Avouris, "Considering cost in usability evaluation of mobile applications: Who, where and when," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5726, no. 1, pp. 231–234, 2009.

- [54] J. Kjeldskov and J. Stage, "New techniques for usability evaluation of mobile systems," *International Journal of Human Computer Studies*, vol. 60, no. 5-6, pp. 599–620, 2004.
- [55] T. Zhou and Y. Lu, "Examining mobile instant messaging user loyalty from the perspectives of network externalities and flow experience," *Computers in Human Behavior*, vol. 27, no. 2, pp. 883–889, 2011.
- [56] M. Perttunen, J. Riekkö, K. Koskinen, and M. Tähti, "Experiments on mobile context-aware instant messaging," in *Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems*, pp. 305–312, Saint Louis, Missouri, USA, May 2005.
- [57] O. Inbar and B. Zilberman, "Usability challenges in creating a multi-IM mobile application," in *Proceedings of the 10th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI 2008*, pp. 453–456, Amsterdam, The Netherlands, September 2008.
- [58] M. Nawi, N. Haron, and M. Hasan, "Context-Aware Instant Messenger with Integrated Scheduling Planner," in *Proceedings of the International Conference on Computer Information Science (ICCIS)*, Kuala Lumpur, Malaysia, 2012.
- [59] D. Jadhav, G. Bhutkar, and V. Mehta, "Usability evaluation of messenger applications for Android phones using cognitive Walkthrough," in *Proceedings of the Joint 11th Asia Pacific Conference on Computer Human Interaction, APCHI 2013 and the 5th Indian Conference on Human Computer Interaction, India HCI 2013*, pp. 9–18, ind, September 2013.
- [60] M. La Polla, F. Martinelli, and D. Sgandurra, "A survey on security for mobile devices," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 446–471, 2013.
- [61] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick, "A survey on privacy in mobile participatory sensing applications," *Journal of Systems and Software*, vol. 84, no. 11, pp. 1928–1946, 2011.
- [62] C. Amrutkar, P. Traynor, and P. C. Van Oorschot, "An Empirical Evaluation of Security Indicators in Mobile Web Browsers," *IEEE Transactions on Mobile Computing*, vol. 14, no. 5, pp. 889–903, 2015.
- [63] M. Rauch, "Mobile documentation: Usability guidelines, and considerations for providing documentation on Kindle, tablets, and smartphones," in *Proceedings of the 2011 IEEE International Professional Communication Conference, IPCC 2011*, USA, October 2011.
- [64] R. Budiuh and J. Nielsen, *Usability of iPad apps and websites*, vol. 1, Nielsen Norman Group, 2011.
- [65] E. Garcia-Lopez, A. Garcia-Cabot, and L. de-Marcos, "An experiment with content distribution methods in touchscreen mobile devices," *Applied Ergonomics*, vol. 50, pp. 79–86, 2015.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

