Received: 25 November 2020 | Revised: 3 April 2021 | Accepted: 13 July 2021

IET Software

DOI: 10.1049/sfw2.12039

ORIGINAL RESEARCH PAPER

Applying usability recommendations when developing mobile instant messaging applications

Sergio Caro-Álvaro | **Eva García-López** | **Antonio García-Cabot** | **Luis de-Marcos** | **Adrián Domínguez-Díaz**

Universidad de Alcalá, Departamento de Ciencias de la Computación, Madrid, Spain

Correspondence
Sergio Caro-Álvaro, Universidad de Alcalá, Departamento de Ciencias de la Computación, Madrid, Spain.
Email: sergio.caro@uah.es

Funding information
Universidad de Alcalá, Grant/Award Number: 30400M000.541.A 640.06

Abstract
Mobile Instant Messaging applications noted a growing number of active users recently. Although having millions of users, the previous research results have shown usability flaws in this type of mobile applications. Difficulties when performing tasks, poor design of user interfaces, or lack of information about privacy and security methods were some of the identified flaws. Consequently, a collection of usability recommendations was proposed. The goal of these recommendations is to improve the experience in the use of these mobile applications and, thus, enhancing the usability of this kind of apps. These recommendations were taken into account during the development process of a new instant messaging application. Thus, in this study, the authors present the resulting application, its characteristics, how usability recommendations were covered through design and development phases, and a preliminary usability evaluation performed by conducting two research methods: KLM and Mobile Heuristic Evaluation. Both methods ranked the prototype at top positions (i.e. good usability results) when compared with the results obtained in previous studies about other existing instant messaging applications.

KEYWORDS
human computer interaction, mobile applications, software usability, usability evaluation methods, user interfaces

1 | INTRODUCTION

Nowadays, mobile devices have obtained the consideration of being essential tools for almost all people [1]. It should be borne in mind that they are the most used electronic devices [2], with more than 3 billion smartphones [3]. Along with the use of mobile phones, mobile applications (thereafter called, simply, "apps") are experiencing an exponential growth in their usage: Between 2009 and 2020, an increase in the number of available mobile apps was reported from 16 thousand apps to 2.9 million apps in Google Play Store and from 25 thousand apps to 1.8 million apps in Apple Apps Store [4–6].

One type of mobile applications with a growing usage is Mobile Instant Messaging apps (namely, MIM or IM apps), with a high number of active users [7, 8], as an evolution of SMS messages [9] and as an improvement of social relationships [10].

Nevertheless, when we put together and observe mobile devices, instant messaging apps, and usability, previous studies have found several usability issues in their usage [11–16]. Given the fact that creating and attracting users to use a MIM app does not require too much effort, improving usability should be the key to maintain users using the app [17, 18]. It is true that web usability is highly widespread and, therefore, the use of usability guidelines is quite common in web development and research. However, mobile usability guidelines (sometimes also referred to as recommendations, standards or directives, depending on the authors) are rather infrequent to be found in mobile research studies, as derived from the current literature review [19, 20].

As said before, MIM apps show a large number of users in a wide range of apps. Based on the research of Nouwens, Griggio and Mackay [21], MIM apps present the lowest satisfaction rates among mobile app users. In the work of Naeem and Rafiq [22], their usability evaluation on MIM apps also showed that there is no consistency in the results; as users' opinions significantly change depending on the MIM app (or apps) they are using. Therefore, it could be argued that something is not being properly addressed among MIM apps. Thus, this research will focus these potential problems by proposing a set of usability recommendations to be applied in the design of MIM apps.

This study will, afterwards, present the development of a MIM app for the Android mobile platform that was created by taking into consideration a set of usability recommendations presented in our previous research studies [11, 12]. This study also covers how usability recommendations should be followed to create MIM apps, as a special note for app developers and researchers. Additionally, our prototype is then examined with a couple of usability tests (i.e., KLM and Mobile Heuristic Evaluation) and the results will be compared with other currently existing commercial MIM apps. This comparison will help to determine how usability-friendly (or not) our prototype is, as well as to validate the developing process of a mobile app by being aware of usability, by applying the set of proposed usability recommendations.

The main contribution of this study is the validation of the set of usability recommendations for MIM apps proposed in previous studies [11, 12]. Indirectly, this research will also try to demonstrate that it is possible to improve the usability of this kind of apps by taking into account the proposed set of usability recommendations during the development phase.

The rest of the study is structured as follows: Section 2 presents background studies related to the app presented in this study. Section 3 introduces the set of proposed usability recommendations for MIM apps. Section 4 presents the developed prototype of the MIM app and how usability recommendations were covered during its development. Section 5 presents a preliminary usability evaluation comparing our prototype with a group of the existing commercial apps. Section 6 briefly discusses the results. Finally, Section 7 presents conclusions and future work.

2 | RELATED WORK

The current diffusion of mobile devices and modern development techniques imply that development techniques should be adapted to this new environment [23]. Smartphones' ubiquity (e.g. smartphone users expected to reach 37% out of global population in 2020 [24]) and intense competition on apps markets are conditioning a reduction in time between software releases [25] and more efforts are required to develop for various mobile platforms [26], thus leading apps' design from traditional methodologies to agile methods [25, 27, 28], which are more flexible on development phases.

Developing and releasing an app without usability in mind makes the app less satisfying, unacceptable, and less productive, and it increases learnability efforts [29, 30]. A simple solution could be a usability assessment at the end of the development phase (but before the release), as presented in [29], or a continuous usability assessment on each agile stage, as presented in [28].

2.1 | Instant messaging usability

As mentioned before, previous studies [13–16] have detected several usability issues in IM apps such as, for example, problems with the visibility of the system status or functionality problems within the chat section. However, our previous research studies [11, 12] determined that the main usability issues of IM apps were the following: Issues when performing tasks (e.g. high learning curves, difficulties to follow the actions flow, unrecoverable errors), poor user interfaces (e.g. content not adapted and limited to available UI, issues when rotating the device, usage of non-standard UI elements) and lack of information about privacy policies and security methods.

It should be highlighted that, over the past years, some MIM apps/systems were developed, such as those shown in [13, 15, 16, 31]. Although we can assume that developers commonly apply usability recommendations, there is no scientific evidence of usability recommendations enhancing user experience in mobile apps [19, 20]. However, a MIM app called *YourIM* [32] was created to overcome the usability and accessibility issues of elderly people in the use of MIM apps. It could be highlighted that, in that app, a semi-structured interview/evaluation method was followed to come up with the prototype, while our research followed a systematic evaluation to produce a set of usability recommendations. However, overall, their research did not aim to provide specific-range recommendations but generic usability recommendations for this type of apps.

2.2 | Usability recommendations for mobile apps

Traditional Nielsen heuristics [33], presented in 2005 for desktop interactions, are not suitable enough for mobile environments. Thus, works like Bertini *et al.* [34], Joyce *et al.* [35] or Inostroza *et al.* [36] presented their own set of heuristics for mobile apps in order to fill this research gap. Therefore, there are tools to evaluate the usability of mobile apps. Nonetheless, the literature is rather scarce when looking for usability recommendations to be applied to mobile apps, as highlighted by Swierenga *et al.* [19] and Shitkova *et al.* [20].

Ahmad, Rextin and Kulsoom [37] propose a set of 25 generic guidelines for mobile apps. Their proposal is divided into seven categories: '*Navigation*', '*Content*', '*Error Handling*', '*Input method*', '*Equitable use*', '*Cognitive Load*' and '*Design*'. Examples of their proposed guidelines could be '*use clear consistent navigation*', '*do not use objects which provide different meanings*', '*limit screen and provide title of screen*' or '*error messages should be simple and easy to follow*', among others.

Shitkova *et al.* [20] propose a set of 39 generic usability guidelines for web and native mobile apps (i.e., multiplatform guidelines). Authors' proposal is divided into five categories: '*Layout*', '*Navigation*', '*Design*', '*Content*' and '*Performance*'. Some of the proposed guidelines are '*Avoid horizontal scrolling*', '*Minimise the number of clicks needed to reach each page*', '*Keep design simple, consistent, uniform and clear*', '*Integrate confirmation dialogs for change and edit actions*' or '*Minimise loading times*', among others.

The work of Hussain *et al.* [38] applied a set of heuristics to validate shopping mobile apps. In their results, authors provide a set of 12 usability recommendations for mobile apps (although special emphasis is made on shopping apps). Some of those recommendations are '*the system should provide exit link or button to exit from the application*', '*the application should provide other languages to cater for users in their location instead of only English to prevent user misunderstanding and to enhance communication*' or '*provide the appropriate message to inform user when there is wrong input*', among others.

The research gap that motivates our research should be identified on the few studies about usability recommendations for MIM apps that could be found on the scientific literature.

In the work of Qu *et al.* [39], applying eye tracking methods on MIM apps, 19 design variables (divided into four sections: '*Search*', '*Click*', '*Main interface*', '*Message interface*' and '*Interaction interface*') have been evaluated on a set of commercial MIM apps. The purpose of this study is to determine the implementation technique of each of the design variables in the MIM applications. Their results showed the following conclusions: search navigation location is mainly placed at the bottom of the layout, historic reviews of clicks is mostly not stored, chat elements usually show time indicators or send key is usually reshaped when the messages are sent, among others.

The work of Tzu-Ning *et al.* [40] applied an acceptance model to determine why elder users use '*LINE*' (i.e., a MIM app). The authors analysed 17 variables, divided it into four groups (usefulness, ease to use, willingness to use and inconveniences), such as '*This tool is able to meet my needs for a mobile*', '*Even without operating instructions or human guidance, you can quickly learn to operate this tool*', '*I think using the LINE APP is a wise choice*' or '*I would recommend friends and family to use this tool*'.

2.3 | Usability recommendations from iOS and Android developer guides

Regarding usability of apps, both Google (i.e., Android) [41] and Apple (i.e., iOS) [42] have published some usability guidelines to help developers to create better apps, in line with each host Operating System (OS). Nevertheless, their guidelines are more related to accessibility than usability recommendations; i.e. colour and contrast, visual alternatives to common sounds or how to create an accessible layout, among others. Nonetheless, in the current trend of hybrid apps (i.e. mobile apps created using web technologies), authors like Santos *et al.* [25] point out that native apps provide better look and feel of usability to users, given the direct access to OS development kits. This implies a greater effort in the development of usable hybrid apps.

2.4 | Other challenges of instant messaging apps

Some studies advised that, given that communications travel through insecure channels, IM apps should secure their transmissions (preferably, with end-to-end encryption) and user-generated data [24, 43, 44], along with local encrypted databases [45], as a way to protect the app from unauthorised (either physical or logical) access to the device. It has been detected a lack of user awareness on mobile security, and it cannot be comparable to current computer awareness [46]. Moreover, if an app is intended to be used by several users, isolation features should be implemented in order to separate one user's data from the others [43]. In addition, other studies recommend that IM apps should supply clear enough privacy policies to users, ensuring that sensitive data is protected and encrypted [45, 47]. These aspects have been detected as a major concern by users [48]. Additionally, notifications are considered to be essential for IM apps, although, in many cases, it was detected that they are rather difficult to configure by users [49].

To conclude this section, it has been found that MIM apps present high levels of energy waste derived from network usage, which is essential in this type of apps. To solve this, energy efficiency in MIM apps should balance energy saving and perceived benefits for the user [50, 51]. A possible solution could be bundling messages (instead of performing individual delivery), in the server side (optimally), and serve them periodically to the clients.

3 | PROPOSED USABILITY RECOMMENDATIONS FOR MIM APPS

3.1 | Source of the proposal

In previous works [11, 12], the researchers of this study came up with a series of usability recommendations to be considered in the development of MIM apps. Our previous research studies followed the methodology of Martin, Flood and Harrison [52]; a systematic usability evaluation created to analyse the usability of mobile apps. In those experiments, MIM apps found in the major marketplaces (i.e. iOS and Android platforms) were analysed with an adapted version of the Keystroke Level Model (KLM) for mobile apps. A subset of those apps (i.e. best KLM performing) were tested with a Heuristic Evaluation. As a result of those systematic evaluations, a set of main usability issues on MIM apps was presented. Those usability issues lead the research to propose a set of usability recommendations for MIM apps.

3.2 | Usability recommendations

Based on our previous works, there is a proposal of a set of 11 usability recommendations to be applied in the design of MIM apps. An overview of the recommendations is presented in Table 1.

- **Status top-bar is always visible.** Each interactive element of the app (e.g. panels or pop-ups) should not hide the top-bar status of the smartphone. While the implementation of the *UI* is taking place, the use of full-screen options should be avoided.
- **Automatic display of the keyboard at new chats.** When the user wants to start a new chat, the app should display the keyboard to start the conversation. This could be accomplished while implementing the logic of the chat: When an access to an empty chat is detected, the keyboard should be automatically launched.
- **Main features should be easy to access.** The number of interactions in the app to access the main features should be as *least* as possible. Based on Miller's study [53], 7 ± 2 should be the optimum number of interactions for each task; since more interactions impacts on user's learnability. Prototyping would help, but it is necessary to verify it with user tests and inspection methods.
- **UI adapted to and limited by the OS.** User interface should follow the design recommendations of a particular OS/platform. Prototyping would help to detect which are the majority of the *UI* elements that would be required on the app, but final implementation should follow platform-specific design rules (e.g. design recommendations for Android [41] or iOS [42]).
- **Design the interface carefully and accurately.** The design of the app should fit different screen sizes and layouts, so the elements of the app are properly displayed. This recommendation would be applied through the entire development process, especially on design and testing iterations. While designing, any prototyping method (e.g. sketching, paper interfaces, storyboards or mock-ups) would help to discover what it is needed on each screen; also helping to discover minimum contents, in order to homogeneously fit in different screens and layouts. Finally, during the testing stage, the design should be validated and, if needed, refined.
- **Visual distinction between individual and group chats.** While the chat list is being shown, use visual clues (e.g. multiple tabs or icons) that help users to differentiate individual and group chats. Prototyping the *UI* would be an answer to this recommendation.
- **Avoid half-translations.** Vocabulary of the app should not mix different languages. This recommendation would be followed during the design phase (it involves creating bigger *UI* elements to allow different languages to be displayed accurately) and the implementation stage (when the texts are coded). The easiest solution should be the localisation (using a different file for each language, containing all the texts of the app) and, hence, internationalisation (i.e. translating those files, one per required language) of any text present in the app.
- **Provide security mechanisms and information to the user.** Security methods should be applied, but also information about those security and privacy methods in the app should be accessible and relevant. As it was discussed previously, the main security feature that should be provided is securing the transmissions (essentially, through *HTTPS*). Additionally, encrypting the local database of messages could be a bonus. Obviously, the server-side should implement measures to ensure data protection. When referred to privacy information, the app should state how (and when) users' private information is managed by the app ecosystem (server side and third parties, if any, included).
- **Do not tolerate unrecoverable errors.** As stated in previous studies, this obvious recommendation is not followed on all apps. In case of an error, it is better to display an error message than forcing an unexpected closing. Error messages should not be the error notice itself, but explicit, human-readable, polite, precise, constructive, and highly noticeable [54]. The main solution to follow this recommendation is performing an in-depth testing of the final product.
- **Add a new contact only with the ID.** When adding a new contact, only one item, the *identifier*, should be enough to register it in the app. This identifier could be either a username, an email or a phone number. The ID should be enough, while other information should be optional and, if desired, added afterwards (e.g., in the mobile phone book or within the contact list details). This should be particularly taken into account while designing the structure of the back-end's database.
- **Account recovery feature.** The app should allow the user to recover his/her account, at least for profile details and contacts. This feature is easier to accomplish if an email or a phone number is stored. The server side should send a regeneration link to the stored email/phone, so the user can follow that link and change the password by him/herself. Another alternative could be the use two-factor

TABLE 1 Summary of recommendations for MIM apps

MIM Recommendations for MIM apps
Status top-bar is always visible
Automatic display of the keyboard at new chats
Main features should be easy to access
UI adapted to and limited by the OS
Design the interface carefully and accurately
Visual distinction between individual and group chats
Avoid half-translations
Provide security mechanisms and information to the user
Do not tolerate unrecoverable errors
Add a new contact only with the ID
Account recovery feature

authentication. Accessing archived messages might not be possible since the server side should not always store exchanged messages for a long time. This could be solved by driving the user to use custom backups (either locally or in the cloud).

3.3 | Comparing proposal with current literature

Proposing and applying usability recommendations in web environments is widely disseminated in the current scientific literature. Nevertheless, usability recommendations for smartphones and its applications are rather unusual to be found in scientific studies, as noticed by Swierenga *et al.* [19] and Shitkova *et al.* [20]. Not only MIM apps present the lowest satisfaction rates among mobile app users [21], but user perception changes depending on the MIM app [22]. Thus, usability recommendations for MIM apps seem to be necessary to enhance usage and users' perception of this type of mobile apps. As said, literature is scarce when searching for usability recommendations for mobile apps. In the following Table 2, we present some research studies that could support our proposal of usability recommendations for MIM apps.

4 | PROTOTYPE APPLICATION: INCORPORATING USABILITY RECOMMENDATIONS INTO MIM APPS

4.1 | System architecture

As depicted in Figure 1, the system architecture for this proposal was built upon three main elements: the MIM app, the Web Service, and the IM Web Server. The MIM app (i.e. the client in this architecture) had, essentially, an encrypted

TABLE 2 Studies that support our proposal of usability recommendations for MIM apps

MIM Recommendations for MIM apps	Supported by	Technique applied
Status top-bar is always visible	Ahmad, Rextin and Kulsoom [37]	SLR + user experiment
	Bertini *et al.* [34]	Heuristic evaluation
	Joyce *et al.* [35]	Heuristic evaluation
Automatic display of the keyboard at new chats	*Specifically proposed in this research*	
Main features should be easy to access	Ahmad, Rextin and Kulsoom [37]	SLR + user experiment
	Hussain *et al.* [38]	Heuristic evaluation
	Joyce *et al.* [35]	Heuristic evaluation
	Naeem and Rafiq [22]	Heuristic evaluation
	Qu *et al.* [39]	Eye-tracking
	Shitkova *et al.* [20]	SLR + custom methodology
	Tzu-Ning *et al.* [40]	TAM model
UI adapted to and limited by the OS	Ahmad, Rextin and Kulsoom [37]	SLR + user experiment
	Apple guides for iOS apps [42]	Undisclosed
	Google guides for Android apps [55]	Undisclosed
	Hussain *et al.* [38]	Heuristic evaluation
	Joyce *et al.* [35]	Heuristic evaluation
	Mendoza [56]	Undisclosed
	Rauch [57]	Review of published usability guidelines
	Shitkova *et al.* [20]	SLR + custom methodology
Design the interface carefully and accurately	Ahmad, Rextin and Kulsoom [37]	SLR + user experiment
	Apple guides for iOS apps [42]	Undisclosed
	Google guides for Android apps [55]	Undisclosed
	Hussain *et al.* [38]	Heuristic evaluation
	Joyce *et al.* [35]	Heuristic evaluation
	Mendoza [56]	Undisclosed
	Qu *et al.* [39]	Eye-tracking

(Continues)

TABLE 2 (Continued)

MIM Recommendations for MIM apps	Supported by	Technique applied
	Shitkova *et al.* [20]	SLR + custom methodology
	Tzu-Ning *et al.* [40]	TAM model
Visual distinction between individual and group chats	*Specifically proposed in this research*	
Avoid half-translations	Ahmad, Rextin and Kulsoom [37] (partially)	SLR + user experiment
	Hussain *et al.* [38]	Heuristic evaluation
	Rauch [57]	Review of published usability guidelines
	Shitkova *et al.* [20]	SLR + custom methodology
Provide security mechanisms and information to the user	Christin *et al.* [58]	Contextual integrity review + user experiment
	Dye y Scarfone [43]	Proposal of best practices
	La-Polla *et al.* [59]	Review
	Vaziripour *et al.* [60]	Pilot study
Do not tolerate unrecoverable errors	Ahmad, Rextin and Kulsoom [37]	SLR + user experiment
	Inostroza *et al.* [36]	Heuristic evaluation
	Joyce *et al.* [35]	Heuristic evaluation
	Nielsen [54]	Undisclosed
	Shitkova *et al.* [20]	SLR + custom methodology
Add a new contact only with the ID	*Specifically proposed in this research*	
Account recovery feature	Bertini *et al.* [34]	Heuristic evaluation
	Nielsen [33]	Heuristic evaluation

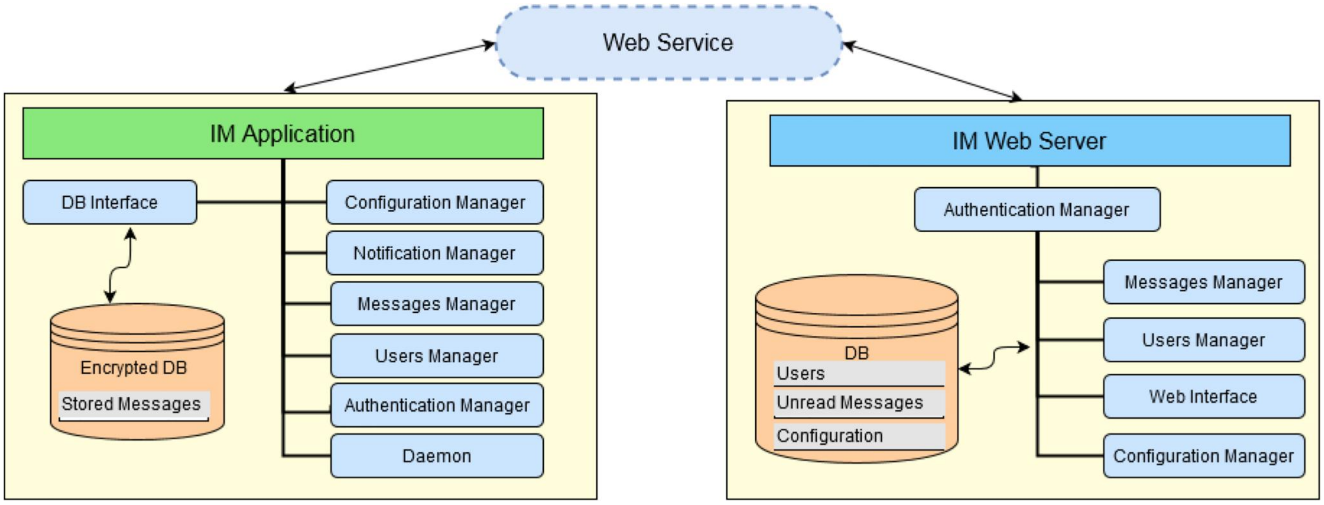


FIGURE 1 System Architecture of the proposed prototype

database to store messages and a daemon service to perform the connections with the server side. Meanwhile, the IM Web Server had, basically, the messages' manager and users' manager (that is, they connect with the server's database) and a web UI (which is a front-end access service through a web browser, similar to what those famous apps apply on their own systems, such as *Whatsapp Web* or *Telegram Web*, that allows access to the MIM app by using an Internet browser).

4.2 | Mobile platform

In order to create a mobile application, it was first required to choose the platform where the app will be deployed. As the required elements to create this mobile app (i.e. access to an internal database, availability to use Internet connectivity, running background services and accessing notifications) were available in the main mobile platforms (that is, iOS, Android and Windows Phone), any of them can be chosen.

Windows Phone was not chosen because its support officially ended in December 2019, so it was unnecessary to deploy in that platform. Between iOS and Android, the latter was chosen. This decision was made given that Android is the most used mobile OS in the market [61] and the software development kit (*SDK*) is open sourced and cross-platform software.

4.3 | Technology

This project required one back-end server and some mobile devices with Android OS. To develop for the Android platform, programing with Java language using the *Android Studio IDE* is required, along with the Android SDK tools. The server side had, as a minimum deploy requirement, an Apache *HTTP* web server coded with *PHP* that answered the incoming petitions through *JSON* notation.

4.4 | App features

Based on previous studies [11, 12], the following are the main MIM features that should be covered on the prototype:

- Message exchange. This is a basic MIM feature, which should allow sending messages to contacts. For ease of implementation, the prototype will only implement one-to-one conversations. Each message would show, additionally to the message itself, the date and time when the message was sent.
- Chat management. This should cover listing active chats and the possibility to delete them (i.e. remove exchanged messages). The list section would show a visual indicator with the number of new messages (if any) in each chat.
- Contact management. This should include listing all friends, adding new contacts to the agenda (upon request) and deleting (or blocking) contacts. For ease of implementation, this prototype will implement the deleting feature, but not the blocking one.
- Account management. This feature should cover creating an account within the app and logging in and out.

In addition to these characteristics, the app could cover the following secondary features:

- Types of messages. In addition to plain text messages, the app would also allow sending emojis and images to a chat.
- Profile management. This should cover changing the login's password and setting a profile photo and a status message. These features would be optional for the user.
- Notifications. When a new message arrives or a new contact relationship is accepted, a notification is shown in the device, even if the app is closed.
- Configuration. A section where the user can change the settings of the app (such as font size or allowing notifications) and that provides frequently asked questions.

4.5 | Description and development of the application

The application developed was named '*Multiplatform Messaging*' (the icon is shown in Figure 2). The primary goal of the mobile app was, given that it was an instant messaging application, establishing a conversation with friends (in other words, contacts). Those friends were required to be managed by the user within the app (i.e. adding and deleting them). As previously stated, the main objective of this project was creating an instant messaging app that meets a set of usability recommendations. Furthermore, this section focus on how the mobile app was created, leaving aside the development of the server side.

The main functional requirements were defined according to the results of our previous research [11, 12] and included in the following features:

- Managing contacts: Adding and deleting contacts.
- Managing messages: Sending, reading, replaying, and deleting messages.
- Including, as a main section, a list with all the active chats.
- Including a background service that was intended to periodically connect to the server side and receive new incoming messages and friendship requests, if any.
- Showing notifications when new messages or friendship requests are received.

Additionally, the main non-functional requirements for the app could be defined as follows:

- Good usability. Following a set of usability recommendations during design and development.
- Reliability of the system. Try to avoid unrecoverable crashes (i.e. manage all errors that could cause those crashes). Also, avoid unexpected behaviours.

FIGURE 2 Icon of the application, as seen in the Android dashboard

- Stable performance. Reduce phone's freezing events. If it is not a game, an app should avoid doing too much *CPU* workload.
- Security methods and information. Wrapping network communications through *HTTPS* protocol. Besides, if necessary, use an encrypted database to store information in the app's device. In addition, information about this should be provided to the user.
- Privacy methods and information. Create a policy that determines how users' private information is managed by the app (and the server side, if any). Also, provide access to this privacy policy to the user.

The purpose of the system was to support the five main tasks that define a Mobile Instant Messaging app, as described in [11,12]: Send a message (T1), read and reply incoming messages (T2), add a new contact (T3), delete or block a contact (T4), and delete a chat (T5).

Hence, the app was planned to have five main sections, as seen in Figure 3: A login screen, a friend list, a section with friendship requests, a chat list, and a settings section. The login screen was in charge of the user authentication in order to access the system (as specified in a non-functional requirement). The friend list was proposed to present all the friends the user may have, including options to delete existing friendships and sending new requests. The friendship requests section was designed to harbour all incoming friendship requests to the user (including accepting and rejecting them, if any), as well as the option to send new ones. The chat list was planned to present all active chats the user may have, if any, including options to open (this leads to the chat section) and delete them. Finally, the settings section was designed to change configuration parameters of the app (chat's globe colour, font size, enable/disable showing the date of messages in chat, enable/disable 'enter' as a keyboard key to send messages, enable/disable notifications and the *LED* colour for notifications).

Once the use cases and requirements were defined, a set of mock-up prototypes were created, regarding the usability recommendation '*design the interface carefully and accurately*,' so usability was considered at a very early development stage. Those mock-ups helped create a flow of actions that are easy to follow by the user (i.e. meeting the usability recommendation '*main features should be easy to access*') before developing, and set up *UI*s with minimal contents and related to the host system (meeting the usability recommendation '*UI adapted to and limited by the OS*').

Those mock-up screens were then translated into Android screens (technically speaking, into *XML* files for the *UI* layouts and Java-code files to set the logic to those layouts), as depicted in Figure 4.

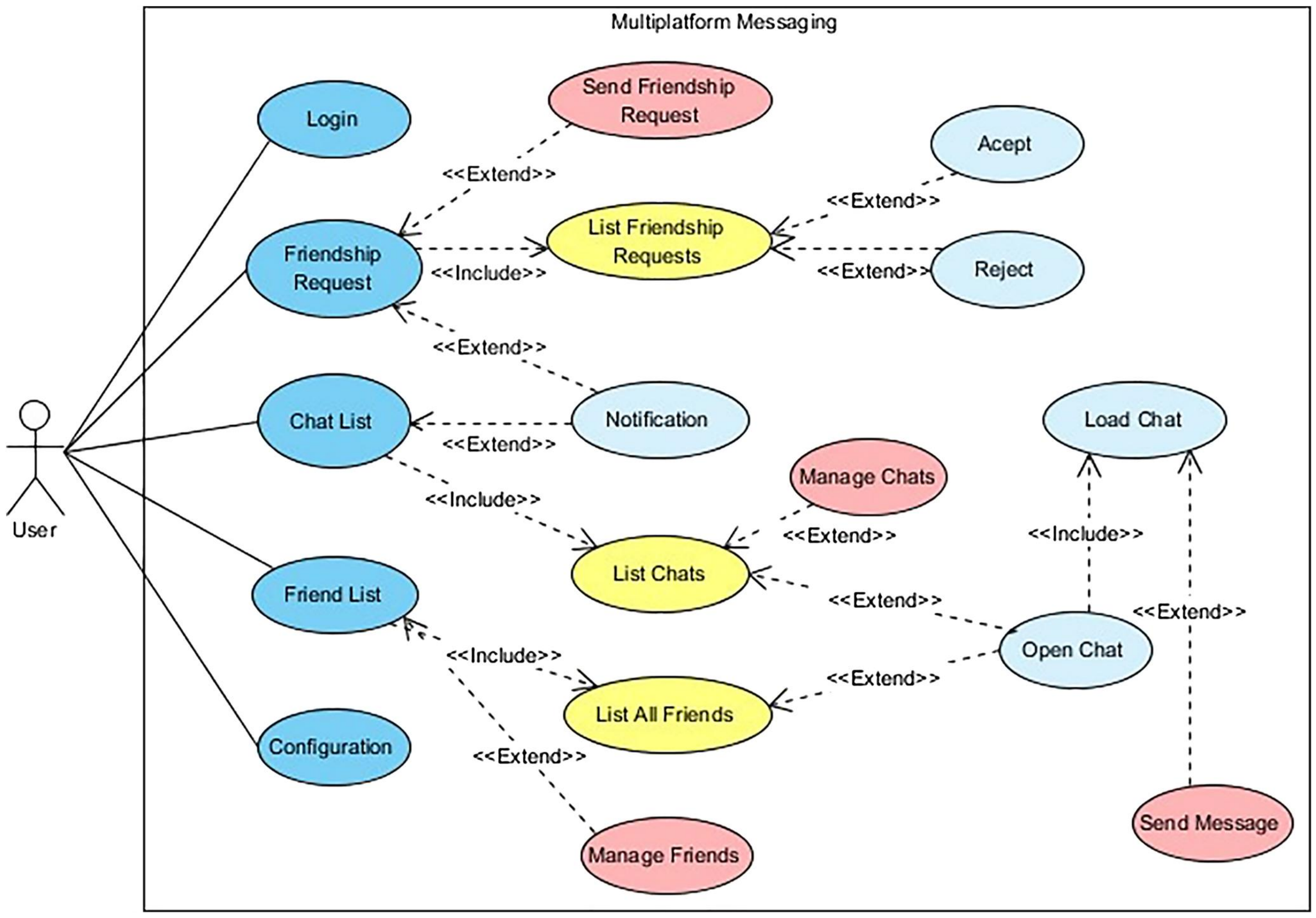
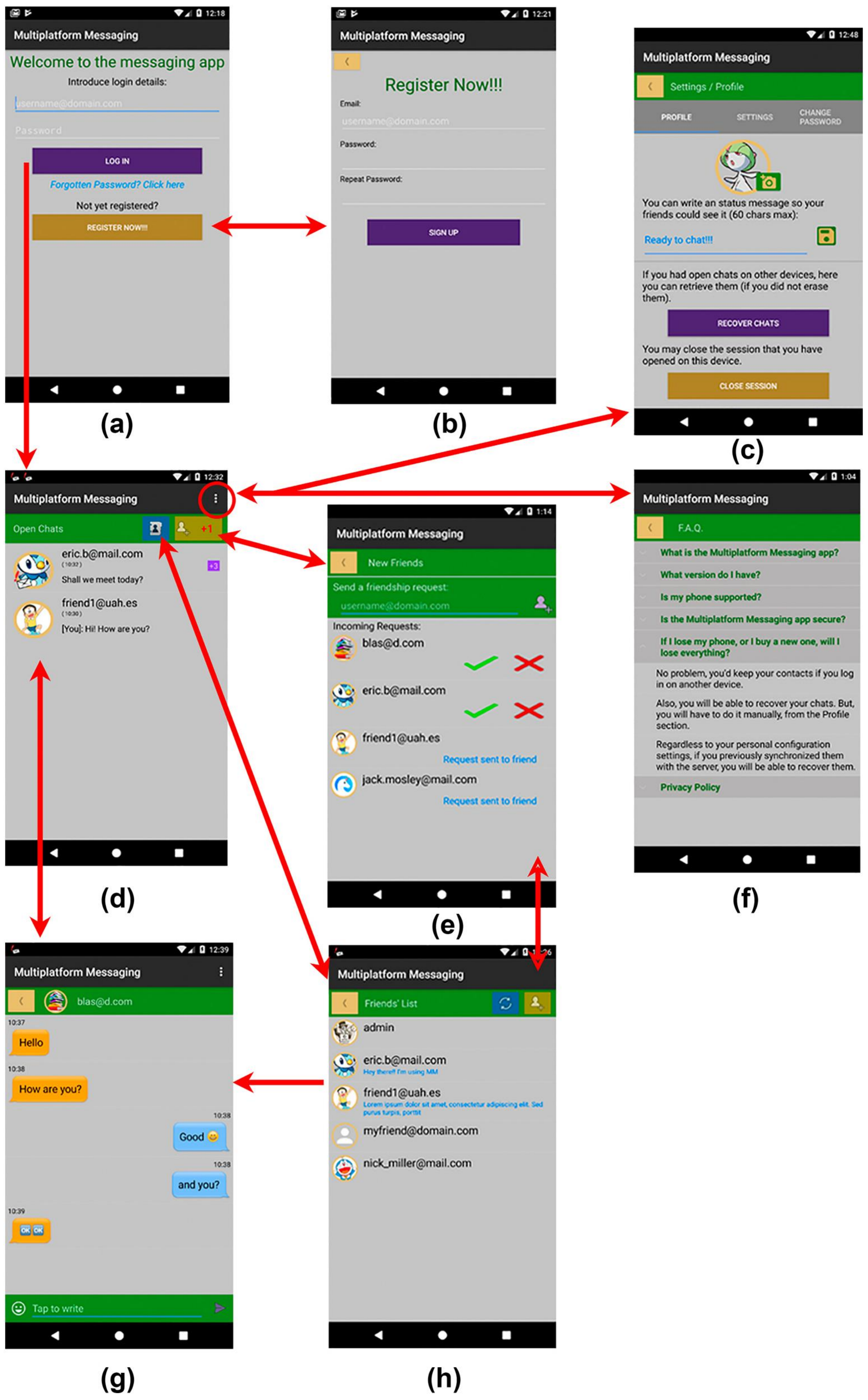


FIGURE 3 Use case diagram of the proposed system



You can write an status message so your friends could see it (60 chars max):

If you had open chats on other devices, here you can retrieve them (if you did not erase them).

You may close the session that you have opened on this device.

Shall we meet today?

[You]: Hi! How are you?

What is the Multiplatform Messaging app?

What version do I have?

Is my phone supported?

Is the Multiplatform Messaging app secure?

If I lose my phone, or I buy a new one, will I lose everything?

No problem, you'd keep your contacts if you log in on another device.

Also, you will be able to recover your chats. But, you will have to do it manually, from the Profile section.

Regardless to your personal configuration settings, if you previously synchronized them with the server, you will be able to recover them.

Hey there!! I'm using MM

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed purus turpis, porttit

FIGURE 4 Windows workflow of the application: (a) login page, (b) sign-up page, (c) (part of the) settings, (d) chat list, (e) friendship requests' view, (f) (F) A.Q. section, (g) chat view and (h) friend list

It should be highlighted that, when accessing an empty or a new chat, the built-in virtual keyboard was automatically shown, thus helping to reduce the number of interactions required, as noticed by the usability recommendation '*automatically display of the keyboard at new chats*.'

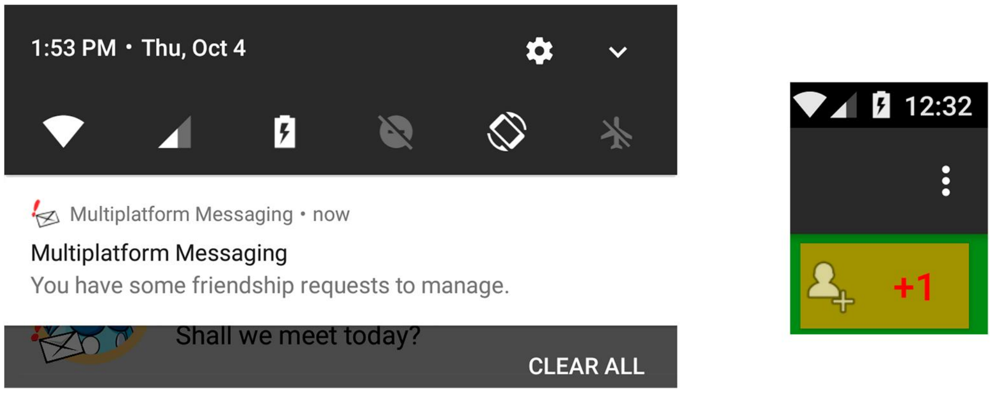


FIGURE 5 Details of a friendship request: Example of notification when new requests are notified (left) and the in-app detail of new requests' counter (right)

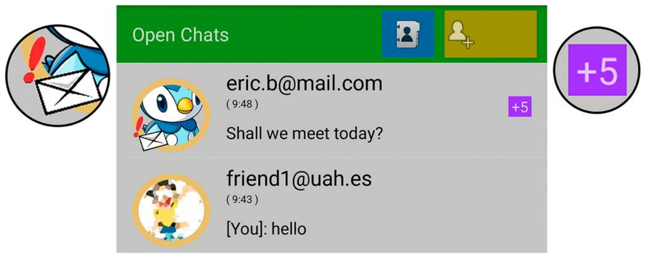


FIGURE 6 In-app detail of visual elements to highlight new incoming messages: (left) closed envelope with an exclamation point and (right) unread message counter

Features to add a user's profile image and a user's status message were added to the app (noticeable in Figure 4h), in order to improve experience in the use of the app, making it possible for users to recognise their contacts at a glance, and their availability in order to start a chat.

Looking to the section where new friends were made (see Figure 4e), the app tried to make it a simple task by only requiring the email of the new contact (as required by the recommendation '*add a new contact only with the ID*'). This section also presents friendship requests, both incoming (in order to accept or reject them) and sent (visually friendly for the user). Besides, incoming friendship requests were shown as Android notifications (see Figure 5, left) as well as a counter indicating the number of incoming requests in the current chat list (seen in Figure 4d, and detail depicted in Figure 5, right).

In-app visual elements to notify new incoming messages (example in Figure 6) were added to the chat list view (Figure 4d) in order to improve glance-ability of unread incoming messages, making a clear distinction from those belonging to already read conversations.

The Frequently Asked Questions (F.A.Q.) section (Figure 4f) presented general details about the app, as well as information to the user about the security measures and privacy policies implemented in the system (to meet the '*security mechanisms and information to the user*' recommendation). In addition, as a privacy measure, Android notifications for new incoming messages only showed the sender's username (which was his/her e-mail, as shown in Figure 7, left), so the message was shown only when accessing to the app. Likewise, incoming friendship request notifications do not show the sender's username (see Figure 5, left). Moreover, those app notifications could be disabled from the settings section (in Figure 7, right).

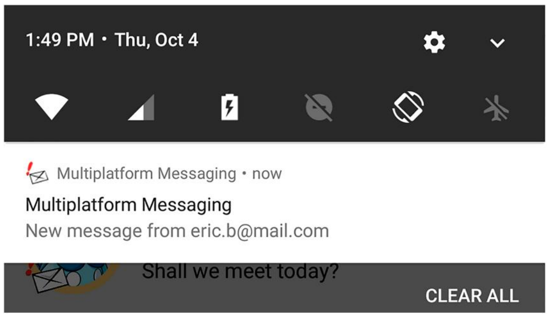
Internationalisation of the app (i.e. turn it into a multi-language application) not only improves the experience of mobile usage with the app but may also help the app to reach more potential users. Although this is theoretically simple, in practice, it requires to avoid the use of any hardcoded strings or any kind of multimedia element with texts in the whole app, which could make the app look as half translated, as highlighted by the '*avoid half-translations*' recommendation. The solution, provided by the Android SDK, consisted in locating those strings in a specific *XML* resource file and then creating translation files, as many as the number of languages required.

According to the recommendation '*account recovery feature*,' this system implemented a mechanism to regenerate the password; upon user request, the system guides the user through the procedure of securely acquiring a new password. In addition, this app had two extra options: one of them to retrieve previous non-deleted conversations and another one to import/export the saved configuration from/to the server side, both available from the *Settings* section (see Figure 4c).

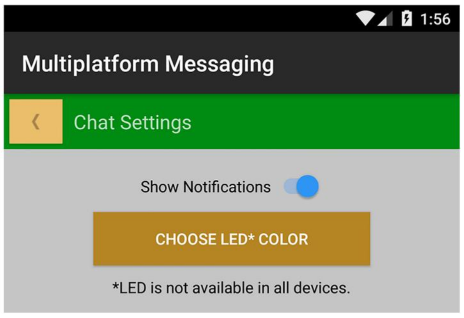
Furthermore, the app was tested both in portrait and landscape orientations, to ensure that the app would always look the same, with elements distributed similarly and adapted to the host system, also verifying that all possible panels or pop-ups did not overlap the status bar. All of this was made according to the '*UI adapted to and limited by the OS*' and the '*status top-bar always visible*' recommendations.

Additionally, the app implemented the accessibility feature of changing the font size of the messages in the chat section (Figure 8), which is customisable via the settings section.

Finally, in order to avoid, as far as possible, any unrecoverable error that could throw an unhandled exception that could force the app to close, all lines of code prone to error (i.e., server requests and responses, I/O operations with the database or contents with dynamic data) were subjected to software testing. All of this derived into information messages


1:49 PM • Thu, Oct 4

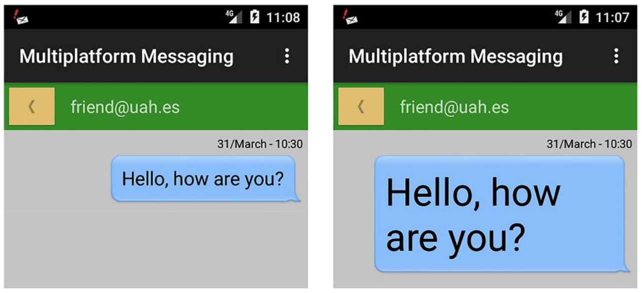
Multiplatform Messaging • now
Multiplatform Messaging
New message from eric.b@mail.com
Shall we meet today?



Show Notifications

*LED is not available in all devices.

FIGURE 7 Detail of an Android notification for new messages (left) and an in-app detail of settings section to disable notifications (right)



Hello, how are you?
Hello, how are you?

FIGURE 8 Accessibility feature of messages (in-app detail): Default font size (left) and custom font size (right) in chat messages

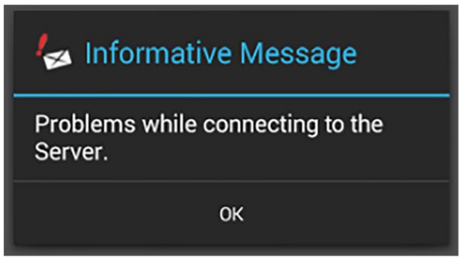

Informative Message
Problems while connecting to the Server.

FIGURE 9 Detail of error management: An information message for the user

to the user, if needed, (as seen in Figure 9), to help him/her to understand what happened and if it was possible to reach a non-error state.

Regarding the security methods applied, the system operated with a username (i.e. the email) and a password as a login method. The password was stored in the server side as the result of applying an *SHA-256* cryptographic hash function with a random salt value. On the client side, as a storage system for all conversations, the app implemented an *SQLite* database with *AES-256* symmetric encryption method, employing the user's password as the encryption key.

4.6 | Usability recommendations: How to design MIM apps

To conclude this section, a brief review on how each usability recommendation was implemented in our prototype is shown.

```
<resources>
    <style
        name="customMMStyle"
        parent="android:Theme.Holo.NoActionBar.Fullscreen"
        >
            <!-- [...] Ommited style code -->
    </style>
</resources>
```

FIGURE 10 Code fragment that should be avoided in order to have the top-bar always visible

4.6.1 | Status top-bar is always visible

When configuring the app's theme, this feature could be achievable avoiding styles of subclasses related to *'NoActionBar'* mixed with *'Fullscreen'* (see an example to be avoided, in Figure 10), which makes the status bar to be hidden by Android. Programmatically, it is possible to achieve the same result by disabling some window manager flags (specifically, the options related to full-screen execution).

4.6.2 | Automatic display of the keyboard at new chats

The easiest way to achieve this recommendation is detecting if the chat is empty, and then, move the focus to the appropriate input text. This way, user's interaction with UI is not forced.

4.6.3 | Main features should be easy to access

It is planned to minimise navigation as much as possible. All main MIM functionalities could be reached by navigating, at most, two windows/screens. Additionally, and trying to make an easy-to-use app, there is no text on almost all buttons. Icon-only buttons are, indeed, a challenge since users' confidence will drop if the icon is not understandable. Nevertheless, MIM apps are usually focussed on the chat, leaving little space in the UI for text buttons, thus forcing to use icon-only buttons. However, universal icons (i.e. those that will be always matched with one specific action) are scarce. That is why we found it necessary to use standard icons (if possible) or, in most cases, icons with equivalent attributes to the triggered action/outcome. Hence, all actions could be reachable at a glance, with distinguishable visual elements (all in line with the OS-specific icons to improve recognisability of the triggered action) (see Figure 11).

4.6.4 | UI adapted to and limited by the OS

To achieve this recommendation, it is highly recommended to use UI elements as similar as possible to those elements in the host OS. Mainly, our prototype was developed using, as the baseline, icons in line with the Android OS (see Figure 12).

4.6.5 | Design the interface carefully and accurately

Physical space is rather limited to assure that all elements will fit on all devices. Our prototype makes use of a scroll view to include all UI elements in the screen. In case the device has a smaller screen than the default, a scroll will be accessible for the user to vertically navigate through the contents. Thus, regardless of the screen size, all contents could be accessible to almost any device. Additionally, at least for Android apps, in order to specify *UI* dimensions, it is a good practice to stop using pixels (or inches or millimetres, in general, avoid any direct value). Instead use *dip* (Density Independent Pixel) or *sp* (Scale Independent Pixel). Different devices have different pixel densities on screen, that is, the equivalence to millimetres (or any other length unit) may differ between devices. Thus, to ensure scalable representation on all devices, pixels have a variable number of units per physical mm/inch, making it virtually impossible to cover the entire spectrum programmatically. To solve this issue, *dip* and *sp* units were created to represent an abstract dimension that automatically scales with regard to the screen density.

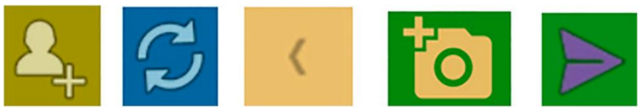
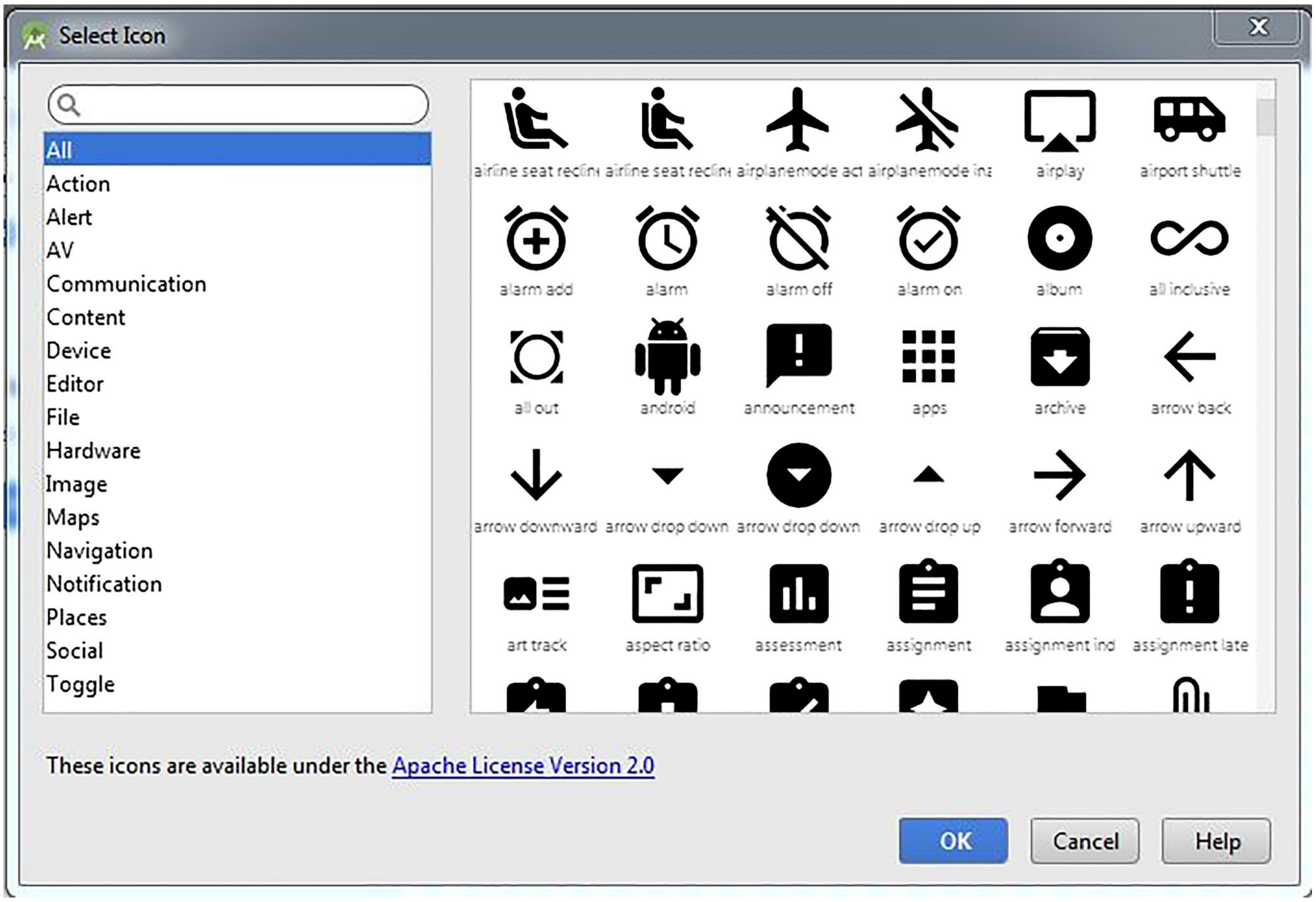
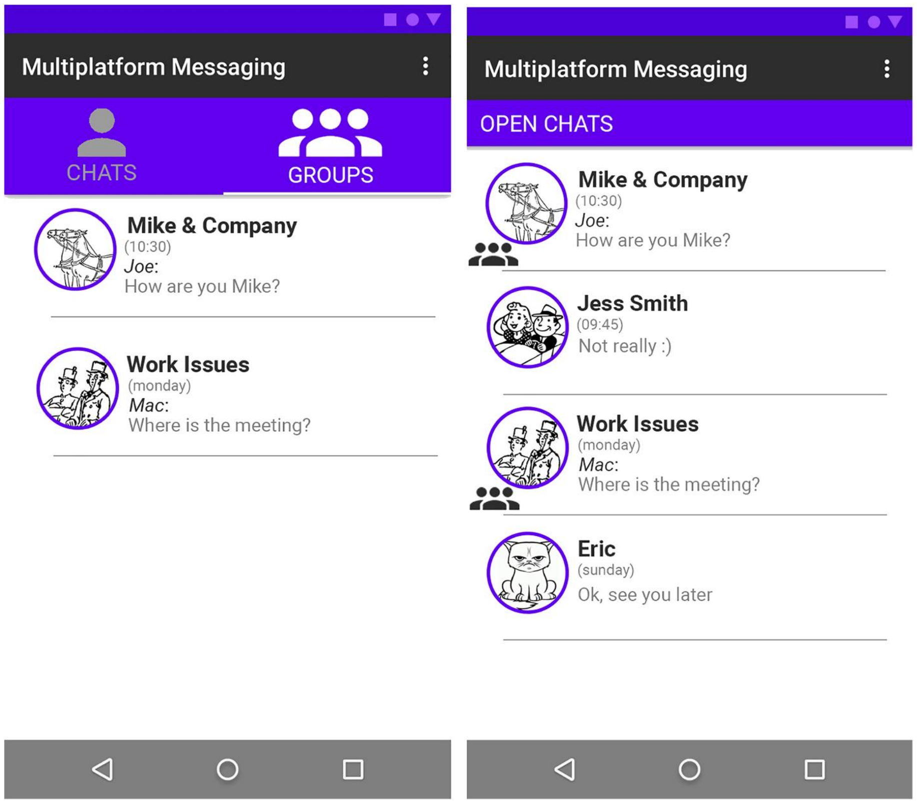


FIGURE 11 Examples of image buttons made to remove text in buttons (from left to right: New contact, refresh content, back to previous screen, take picture and send message)



These icons are available under the Apache License Version 2.0

FIGURE 12 Example of built-in icons on Android's SDK



Mike & Company
(10:30)
Joe:
How are you Mike?

Work Issues
(monday)
Mac:
Where is the meeting?

Multiplatform Messaging

Mike & Company
(10:30)
Joe:
How are you Mike?

Jess Smith
(09:45)
Not really :)

Work Issues
(monday)
Mac:
Where is the meeting?

Eric
(sunday)
Ok, see you later

FIGURE 13 Mock-up ideas to make distinctions between individual and group chats: (left) on each chat and (right) different sections to place chats

4.6.6 | Visual distinction between individual and group chats

Although our prototype does not implement group chats, it was planned on the design phase (as seen in Figure 13). This distinction could be made by adding an icon to group chats (on chat list, if both types are mixed) or dividing the chats into two sections: One-to-one chats and group chats. This recommendation could also be applied to broadcast lists or channels.

4.6.7 | Avoid half-translations

The best way to face this recommendation is by localising all strings in the app (i.e. avoid hardcoding any text that could be read by users) and internationalise those strings with different languages (obviously, the number of languages depends on the target of the app). Hardcoded texts imply that, at running time, the app is not aware that those texts that need to be replaced with an OS-language translation. Hence, the app will have a base strings file (intended for developing and in case the language of the device is not available in the app) and additional strings files, one per language. For example, both Android and iOS platforms support this feature with similar approaches (*.xml* files in the former, *.strings* files for the latter).

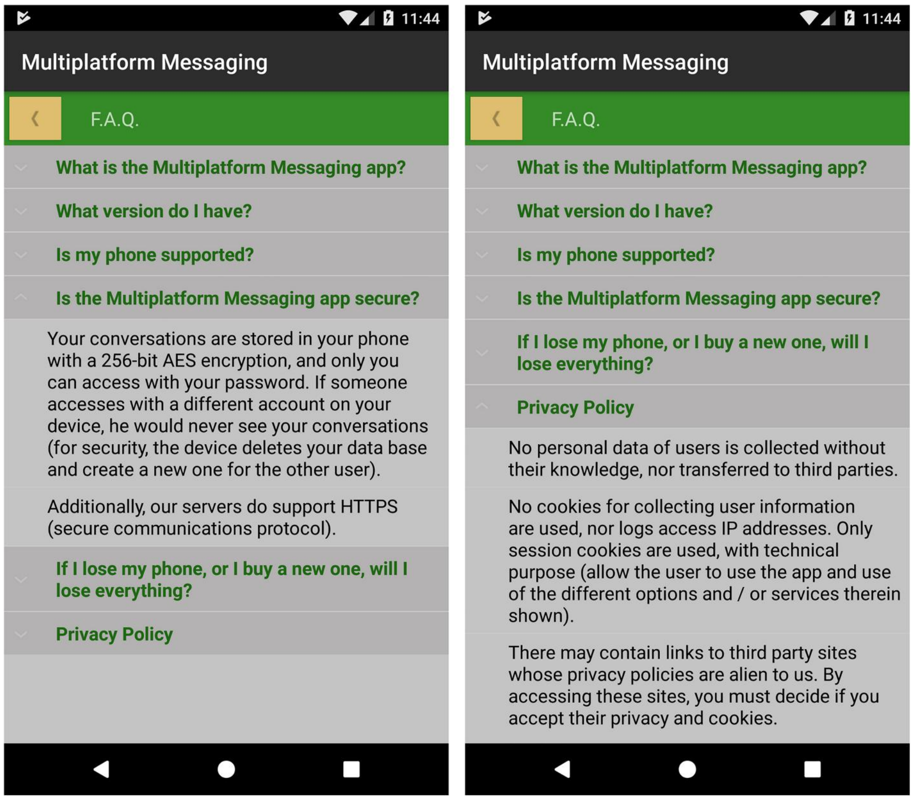
4.6.8 | Provide security mechanisms and information to the user

There are two main dangerous points when we consider security in an app: network connections and stored information. Our prototype applies *HTTPS* (secure network protocol) in all network connections and *SQLite* (database manager available for Android) with *AES* 256-bits encryption to store and encrypt all chats. Not only those methods are applied, but also information to the user is provided, along with a basic privacy policy (see Figure 14).

4.6.9 | Do not tolerate unrecoverable errors

It is not good for an app to crash (Figure 15), because the user is forced to manually relaunch the app, often without being aware of what exactly caused the crash. In general, for any app, any piece of code prone to produce an error (unrecoverable or not) should be wrapped to be executed in the safe mode.

If an error is triggered, actions should be taken, informing the user (see examples in Figure 16), if necessary. For those background errors (i.e. not caused by a previous user interaction), we chose to apply a silent mode error prevention. An unrecoverable error is as dangerous as an error message from which users might not be aware of its cause.


11:44
Multiplatform Messaging
F.A.Q.
What is the Multiplatform Messaging app?
What version do I have?
Is my phone supported?
Is the Multiplatform Messaging app secure?

Your conversations are stored in your phone with a 256-bit AES encryption, and only you can access with your password. If someone accesses with a different account on your device, he would never see your conversations (for security, the device deletes your data base and create a new one for the other user).

Additionally, our servers do support HTTPS (secure communications protocol).

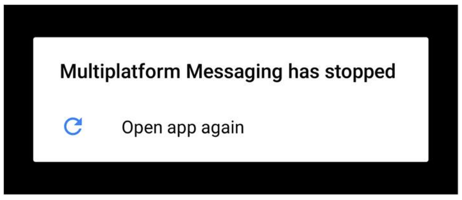
If I lose my phone, or I buy a new one, will I lose everything?
Privacy Policy

No personal data of users is collected without their knowledge, nor transferred to third parties.

No cookies for collecting user information are used, nor logs access IP addresses. Only session cookies are used, with technical purpose (allow the user to use the app and use of the different options and / or services therein shown).

There may contain links to third party sites whose privacy policies are alien to us. By accessing these sites, you must decide if you accept their privacy and cookies.

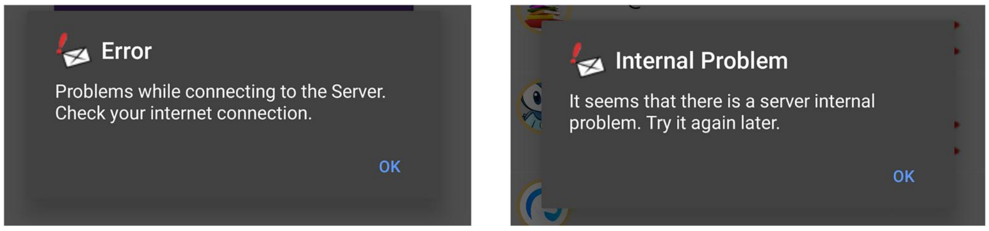
FIGURE 14 Details of the FAQ section, showing security information and a basic privacy policy to the user



Multiplatform Messaging has stopped

Open app again

FIGURE 15 Example of an unrecoverable error message



Error

Problems while connecting to the Server. Check your internet connection.

OK

Internal Problem

It seems that there is a server internal problem. Try it again later.

OK

FIGURE 16 Examples of error messages to the user

4.6.10 | Add a new contact only with the ID

The email (i.e. the ID in our prototype) is the only field required to send a friendship request (see Figure 17). This feature, apart from providing a minimalist design, allows the app to be highly scalable since any extra parameter should be provided in another section.

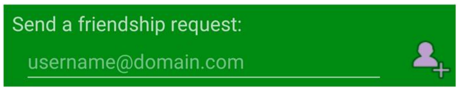
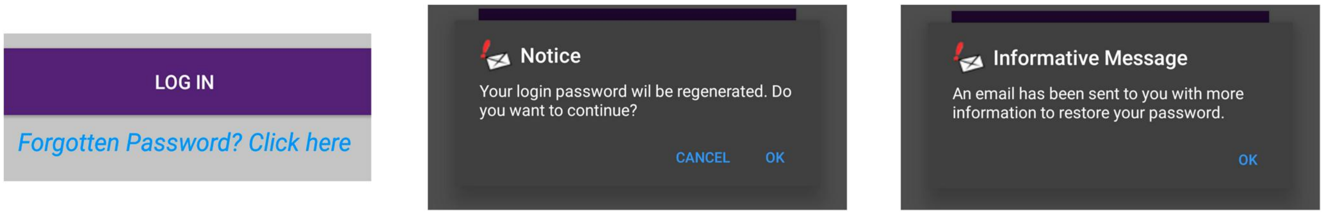


FIGURE 17 Detail of the UI, asking only for the email to send a friendship request


LOG IN

Forgotten Password? Click here

Notice

Your login password wil be regenerated. Do you want to continue?

CANCEL OK

Informative Message

An email has been sent to you with more information to restore your password.

OK

FIGURE 18 Screenshots to show how to ask for password regeneration

4.6.11 | Account recovery feature

This app implemented this feature by sending an email with a token and instructions on how to regenerate the user's password. In order to ask for it, a link is located in the login screen (see Figure 18); the user might find this feature more useful.

5 | PRELIMINARY USABILITY EVALUATION

5.1 | Methodology

At this point, a preliminary usability evaluation was performed. To do so, a couple of usability methods were applied: The Keystroke-Level Model (also known as KLM) and a heuristic evaluation approach specifically designed for mobile devices (also referred to as Mobile Heuristic Evaluation), measuring efficiency and effectiveness; these attributes are seen in most mobile usability evaluations [62, 63].

Both inspection methods have proven to be useful for scientific research on evaluating mobile apps, as seen in previous studies [64–66]. The results obtained with our prototype were compared with those findings obtained in a previous usability study about instant messaging apps [12].

The first usability evaluation was KLM. The model followed was presented by Martin *et al.* [52], and it is a KLM adaptation for mobile interfaces to measure efficiency. This method evaluates by counting the number of interactions required for the main tasks that define a mobile app. Each interaction (e.g., tap, scroll or swipe) is counted as one point. In this particular case, the main tasks of Instant Messaging app were the main tasks that were defined in previous studies [11,12]: (T1) Send a message to a specific contact, (T2) read and reply to incoming messages, (T3) add a new contact, (T4) delete or block a contact, and (T5) delete a chat.

The second usability evaluation was a Mobile Heuristic Evaluation approach. This method evaluates a set of directives (i.e. heuristics) by a group of eight mobile experts determining a degree of whether or not those directives were offered in the evaluated prototype, which complies the suggestion of having at least three to five evaluators that could uncover 74%-87% usability flaws [67]. These mobile experts were aged between 26 and 34 years, five men and three women, all of them with more than four years of background experience with mobile devices and apps. The selected experts were mobile UI practitioners with Mobile Heuristic Evaluation knowledge, given that even the best practitioner in mobile heuristics, but with little or non-mobile UI background, could result in false positive errors. Most of these experts (five out of eight) participated in the previous studies (as shown in [11,12]). Same instructions as in previous studies were provided to all experts. It should be noted that the authors consider that these evaluators could be characterised as both '*regular evaluators*' and '*double specialists*' [67], since they are usability specialists with extended experience working with mobile interfaces, but not fully experienced in working with Instant Messaging interfaces. That is why it was decided to use more than the recommended set of three to five usability evaluators defined by Nielsen [67]. With all of this in hand, the authors think that the results found here could be comparable with those found in previous studies.

Regarding the set of examined heuristics, they were specifically created to inspect mobile apps. These heuristics are depicted in a previous work by Bertini *et al.* [34] as follows: (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 glance-ability*', (F) '*flexibility, efficiency of use and personalisation*', (G) '*aesthetic, privacy and social conventions*' and (H) '*realistic error management*'. Nielsen's five-point Severity Ranking Scale [68] was applied as an evaluation scale of the given heuristics. Each heuristic is assessed with a 0 to 4 scale (the higher the value, the worse is the usability result): no problem at all (0), cosmetic problem (1), minor problem (2), major problem (3) and usability catastrophe (4).

5.2 | Results

Results from KLM evaluation could be observed in Table 3, showing that the application presented here has similar results, on average, to the best-qualified apps. Moreover, when results are compared with all apps, our prototype presents slightly lower values (i.e. better results) than those of the average values for each task.

Outcomes from the Mobile Heuristic Evaluation are shown in Table 4 (results are compared with data of previous studies).

The results for those apps (except prototype), as extracted from a previous study [12], show that the following were the

TABLE 3 KLM results

Apps \ Functionalities	T1	T2	T3	T4	T5	Total
Surespot encrypted messenger	4	4	3	4	4	19
ZOHIB messenger	5	5	4	5	4	23
Multiplatform Messaging	5	5	5	5	4	24
Cnectd messenger	5	4	6	5	4	24
Yak messenger	6	5	6	4	3	24
HushHushApp	5	6	5	4	5	25
Kik messenger	5	5	5	7	3	25
Mean (these apps, except prototype)	5.00	4.86	4.86	4.86	3.86	23.43
Mean (all apps shown in [12])	6.27	5.47	7.62	5.73	4.31	29.40

Note: Data of the MIM prototype is highlighted with a background colour in its row. Note that mean values do not take into account the values of the prototype presented here. Main data were extracted from [12], showing here only the six best apps (i.e. apps with lower keystrokes).

TABLE 4 Mobile Heuristic Evaluation results

APP \ Heuristics	A	B	C	D	E	F	G	H	Total
HushHush APP	0.00	0.40	0.50	0.10	0.28	0.00	0.10	0.33	1.71
Multiplatform Messaging	0.00	0.46	0.25	0.50	0.08	0.44	0.38	0.08	2.18
Yak messenger	1.00	1.27	1.40	0.20	0.56	0.00	2.30	0.33	7.06
ZOHIB messenger	0.00	0.53	1.50	0.30	0.72	1.00	2.30	1.20	7.55
Cnectd messenger	0.05	0.60	1.50	0.60	0.44	0.80	3.30	0.27	7.56
Surespot encrypetd messenger	0.60	1.27	1.30	0.90	0.96	1.90	1.20	0.00	8.13
Kik messenger	0.00	1.47	2.70	2.10	0.96	0.50	2.10	0.00	9.83
Mean (excluding prototype)	0.28	0.92	1.48	0.70	0.65	0.70	1.88	0.36	6.97

Note: For each app, it presents the means of severity ratings for each Heuristic. Main data was extracted from [12]. The last column, '*TOTAL*,' is not in the original source and it is calculated here. The results of the MIM prototype are highlighted with a background colour in their row. Note that the mean values do not take into account the values of the prototype presented here.

main problems (results' summary is broken down by heuristics):

- Heuristic A ('*visibility of system status and losability/findability of the mobile device*'). Status top-bars were not always visible, and some apps did not provide account recovery features.
- Heuristic B ('*match between system and the real world*'). Several problems adapting and limiting UI contents, along with design flaws.
- Heuristic C ('*consistency and mapping*'). As seen in the previous table, the results showed that this was the second worst heuristic. Essentially, this was caused by using too many UI elements in the layouts, together with the reports of designs not in line with the host OS.
- Heuristic D ('*good ergonomics and minimalist design*'). Learnability problems were reported, due to tasks requiring a high number of interactions and clipped translations.
- Heuristic E ('*ease of input, screen readability and glanceability*'). Mainly, difficulties to differentiate individual chats from group chats were found.
- Heuristic F ('*flexibility, efficiency of use and personalisation*'). Mostly, layout adaptability problems were found when the device was rotated. Apart from this, some apps require more than the ID to add a new contact.
- Heuristic G ('*aesthetic, privacy and social conventions*'). As seen in the previous table, this was the worst heuristic, i.e. here is where the main flaws were detected. Mainly, poor UI designs were reported. Also, layouts failed to fit different screen sizes and device orientations. Apart from this, several apps failed to provide information on applied security and privacy methods.
- Heuristic H ('*realistic error management*'). Primarily, unrecoverable errors were found.

In general, the results for our prototype were quite positive, given that the app ranked on first positions and all heuristics' results were below the average values. These results imply that following the proposed usability recommendations really helps to improve usability of these types of apps. Some results of the prototype are a bit higher than those of the leader app, however, it is not perfect. It is true that there are some limitations in the results, but we believe this is because the application is a prototype. The app is mainly focussed on providing the main functionalities of MIM apps, so the deficiencies found by experts can be located outside the context of these main functionalities. The evaluation results will be broken down by each of the examined heuristics:

- Heuristic A ('*visibility of system status and losability/findability of the mobile device*'). No issues related to this heuristic have been detected.
- Heuristic B ('*match between system and the real world*'). Experts pointed out that content distribution on the settings section is slightly irregular.
- Heuristic C ('*consistency and mapping*'). As in other heuristics, our experts pointed out the slightly irregular content distribution on the settings section. Additionally, the experts noted that the input field placed to send new friendship requests should be somehow more highlighted.
- Heuristic D ('*good ergonomics and minimalist design*'). As pointed out by the experts, our prototype could look better using modern Android UI elements. Furthermore, as in previous heuristics, experts highlighted that ergonomics of the app could be improved with a slight redesign of the settings section and new friends screen.
- Heuristic E ('*ease of input, screen readability and glancability*'). Minor improvements were suggested to highlight relevant input fields.
- Heuristic F ('*flexibility, efficiency of use and personalisation*'). The app may include some extra personalisation features (such as a background image on chats) and the possibility to send multimedia files on chats.
- Heuristic G ('*aesthetic, privacy and social conventions*'). Experts noted that, given the basic functionalities the app provided, UI design was accurate. Nonetheless, the design could be improved with the usage of modern Android UI elements. Also, and previously indicated in other heuristics, a design improvement in content distribution on the settings section could enhance the usability results of the whole app.
- Heuristic H ('*realistic error management*'). Minor adjustments to some error notifications were suggested, in order to provide more information to the user about what produced the error.

6 | DISCUSSION

KLM results ranked our prototype (i.e. the system applying our proposed set of usability recommendations) with a similar number of interactions compared to the rest of the MIM applications analysed. It is true that *Surespot encrypted messenger* app required the lowest number of interactions to complete the required tasks. However, it turned out to be an expected result. It is an app with all content in a single window. From the results of the heuristics it was possible to detect usability issues related to efficiency and content distribution.

The heuristic results were, also, positive for the prototype. The prototype had an overall result quite similar to that of the top-ranked MIM app, and with a noticeable distance from the rest of the evaluated MIM applications. It should be highlighted that the MIM prototype presented some deficiencies. Usability issues found were detected on non-primary MIM tasks, hence, we believe that these weaknesses may be related to the prototype-state of the app, and they could be solved in future releases. It seems to be a usual pattern among usability-focussed prototypes, as derived from Nawi *et al.* [15] and Perttunen *et al.* [16]. Their research studies showed, when creating usable MIM apps, minor usability errors, as we did on our system, associated to being in a prototype phase can occur. It could be said that they are usability errors that fall within the expected ones for the creation of a prototype software.

A comparison between our prototype and the best-ranked app (*HushHush App*) should be set to determine how our MIM app should be improved to be more usability-friendly. Mainly, our experts pointed out that *HushHush App* had a minimalist design, it was a flexible app with plenty of personalisation features, and it presented a couple of privacy features (e.g., a pin-code to restrict access to certain chats). The main drawback of *HushHush App* was error prevention, since it was not considered as good as expected (e.g. unclear error messages). Furthermore, hidden sections (such as chats visible after inserting a code) reduced its screen readability.

It is usual for the mobile brands themselves to publish what they call usability guides [41, 42]. However, these guides are more focussed on generating content according to the host's interface, than specifically enhancing the usability of mobile apps, as it was performed in this study.

In the work of Qu *et al.* [39], which was carried out by applying eye tracking technology on MIM apps, the authors found some design variables to enhance the use of MIM apps. The authors only focussed their experiments on the message interface and contacts interface, which could match our T1 and T2 tasks (message exchange). Similar to our recommendations of making features easy to access and provide careful UI designs, the authors found that both navigation and design styles of labels have a high impact on the user's interactivity with the app.

In the work of Tzu-Ning *et al.* [40], a usability acceptance model for elderly MIM users is presented. The elements such as problems with steps accuracy (i.e. navigation issues), difficulty to understand the target of certain buttons or interface designs are among the main usability keys for this kind of apps. All of these are issues in line with some of our recommendations.

In the study by Vaziripour *et al.* [60], which evaluates the usability of security mechanisms of MIM apps, the authors showed that most users do not understand why security is a threat to the communication process. Without any kind of information or help from authors, only 14% of participants were able to determine whether or not an app was safe to communicate with. When they were informed and instructed, this ratio rose up to 78% of participants. These results clearly justify our usability recommendation on not only incorporating security methods in MIM apps, but also informing the users about how to verify that an app is safe to use.

Shitkova *et al.* [20] defined and validated a set of usability guidelines for multiplatform (web and mobile) apps. Although their guidelines are focussed mainly on the distribution of content to produce unified web and mobile products, some of their guidelines are similar to those proposed in this study, such as providing uniform distribution of content and little navigation, or have a simple design, among others.

It is somehow clear that MIM apps have usability issues that could be solved by applying the set of usability recommendations presented in this research. Although focussed on MIM apps, it is true that some of the proposed usability recommendations could be applied in a wide range of mobile apps. The lessons we have learnt from our experiment could be twofold. First, usability-focussed prototypes present minor usability issues. Second, usability methods applied (i.e. KLM and Heuristic Evaluation) could be the first approach looking for usability issues, although end users are required for further validation.

Finally, we could conclude from the results that the proposed usability recommendations for MIM apps seem to be helpful to design more usable MIM apps. Nonetheless, the end users will be invited to an experiment with our proposed usability recommendations, which should be validated in subsequent research.

6.1 | Threats to validity

This research presents some threats that could bias the validity of the proposed usability recommendations for the development of MIM apps. First, the results may be seen as they are not generalisable: Are good results related to the proposed usability recommendations? Or are they linked to the developed prototype? The prototype was developed from scratch applying the proposed usability recommendations. The prototype system does not exist prior to this research. The prototype was then tested in comparison with the market-available MIM apps to measure the effects of the proposed usability recommendations. In case of product fidelity concerns of our research conclusions, based on the study of Sauer and Sonderegger [69], there are no apparent effects of high fidelity prototypes on usability evaluation results. Thus, we think that the prototype itself (i.e. as a unique app including our proposed usability recommendations) presents a reduced bias on our usability results, if so. Although, of course, we cannot completely assure whether the good results are related to the proposed usability recommendations or the prototype itself.

Second, the applied usability evaluation required the assistance of mobile usability experts only. There are no tests with end users. Further work will address end user's performance and satisfaction with the proposed set of usability recommendations. It should be highlighted that the usability methods applied in this research (i.e. KLM and Heuristic Evaluation) are part of the same systematic usability evaluation applied in our previous research studies. The applied methodology of Martin, Flood and Harrison [52] considered that applying both KLM and Heuristic Evaluation is an economical alternative to detect a wide range of usability issues on mobile apps. The background of evaluators could be another concern of the Heuristic Evaluation results. As we explained in Section 5.1, a part of the evaluators in this research already assisted in previous research works on this topic. They were self-characterised as mobile usability experts, rather than experts in MIM apps. As explained before, that is why we required the assistance of more evaluators than it is usually suggested. Thus, we consider that results derived from the evaluators are sound.

7 | CONCLUSIONS

Within this study, we were able to argue that instant messaging apps (MIM apps) currently available in market could improve their (inadequate) usability by carrying out a simple and fast evaluation as we did here with our prototype. For the existing apps and those yet to come, they could be (re)designed and created in a usability-friendly way by following a set of usability recommendations presented here and in the previous literature of the authors. As a summary, the following are the usability recommendations for instant messaging apps:

- Status top-bar is always visible.
- Automatic display of the keyboard at new chats.
- Main features should be easy to access.
- UI adapted to and limited by the OS.
- Design the interface carefully and accurately.
- Visual distinction between individual and group chats.
- Avoid half-translations.
- Provide security mechanisms and information to the user.
- Do not tolerate unrecoverable errors.
- Add a new contact only with the ID.
- Account recovery feature.

Towards the improvement of the relationship between current digital society and mobile information systems, this study showed how to create mobile instant messaging applications considering the usability recommendations during the development phase. The main contribution of this study is the evaluation of the usability recommendations in the design and implementation process of IM apps. This has been achieved by creating a prototype of a mobile instant messaging application for the Android platform that was planned to be usability-friendly by applying the proposed set of usability recommendations. Thus, during the development process of the app, we took into consideration our proposal set of usability recommendations and we, thereafter, evaluated the results by means of two usability evaluation methods: KLM and Mobile Heuristic Evaluation.

This study concludes that usability methods can be applied during software development lifecycle to improve user-friendliness of instant messaging apps; due to applied usability methods, our prototype ranked among top-rated applications. Similarly, this preliminary usability evaluation has shown that the proposed prototype and (by extension) the usability recommendations have rather positive usability results.

On the other hand, the prototype also has some small drawbacks resulting in some comments made by the experts.

The deficiencies could be due to two factors: The app is a prototype (as literature shows, prototype apps may present minor usability issues) and the deficiencies found are mainly outside the context of the main functionalities of the app. To summarise, the mean weakness of this app is the irregular content distribution of elements in the settings section (e.g. lack of visual distinctions in the settings) and a slightly outdated usage of Android UI elements (e.g. the app should apply a colour palette and icons in line with modern versions of the Android OS). Furthermore, chats (i.e. the chat window) should allow the user to send multimedia files (such as images, audios or videos). These experts' recommendations will be taken into consideration in future improvements of the mobile application, which is currently only a prototype in its first development stages. Additionally with regard to the app limitations, the main limitation of the study could be that we did not use real users to test the prototype.

As a future work, we plan to further refine the usability of this mobile application through usability enhancements (to improve the application, if necessary) and we will test it with real users, to formally validate the proposed set of usability recommendations. We also plan to redesign the app, especially in the parts where experts pointed out in their recommendations. Upon the evaluation of newer versions of the app, we plan to compare again the evaluation results, which could bring us to new conclusions.

ACKNOWLEDGEMENTS

This research was funded by the FPU research staff education programme of the 'Universidad de Alcalá' (UAH), grant number 30400M000.541.A 640.06. The authors would also like to thank the support of TIFYC and PMI research groups.

DATA AVAILABILITY STATEMENT

The data that supports the findings of this study are available in the article.

ORCID

Sergio Caro-Álvaro https://orcid.org/0000-0002-3192-8499
Eva García-López https://orcid.org/0000-0002-7598-3289
Antonio García-Cabot https://orcid.org/0000-0002-0298-3237
Luis de-Marcos https://orcid.org/0000-0003-0718-8774
Adrián Domínguez-Díaz https://orcid.org/0000-0001-7632-8609

REFERENCES

1. Lv, M., et al.: A collaborative deep and shallow semisupervised Learning framework for mobile app classification. Mobile Inf Syst. 2020, 1–12 (2020). https://doi.org/10.1155/2020/4521723
2. Smith, M.L., Spence, R., Rashid, A.T.: Mobile phones and expanding human capabilities. Inf. Technol. Int. Dev. 7(3), 77–88 (2011). https://itidjournal.org/index.php/itid/article/view/762.html
3. statista, (2017), Smartphones Worldwide Installed Base from 2008 to 2017 (in Millions) [Online]. Available: https://www.statista.com/statistics/371889/smartphone-worldwide-installed-base/. [Accessed: 20-May-2019]
4. Cuadrado, F., Dueñas, J.C.: Mobile application stores: success factors, existing approaches, and future developments. IEEE. Commun. Mag. 50(11), 160–167 (2012). https://doi.org/10.1109/mcom.2012.6353696
5. statista 2020, Number of Available Applications in the Google Play Store from December 2009 to December 2019 [Online]. Available: https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/. [Accessed: 27-Feb-2020]
6. Costello, S.: How many apps are in the app store? [Online]. Available: https://www.lifewire.com/how-many-apps-in-app-store-2000252 (2020). Accessed 27 Feb 2020
7. statista, 2019, Number of Mobile Phone Messaging App Users Worldwide from 2018 to 2022 [Online]. Available: https://www.statista.com/statistics/483255/number-of-mobile-messaging-users-worldwide/ [Accessed: 27-Feb-2020]
8. statista.Most Popular Global Mobile Messenger Apps as of October 2019, Based on Number of Monthly Active Users [Online]. Available: https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/(2019). Accessed 27 Feb 2020
9. Church, K., Oliveira, D.R.: 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. Association for Computing Machinery, New York (2013). https://doi.org/10.1145/2493190.2493225
10. Quan-Haase, A., Young, A.L.: Uses and gratifications of social media: A comparison of facebook and instant messaging. Bull. Sci. Technol. Soc. 30(5), 350–361 (2010). https://doi.org/10.1177/0270467610380009
11. Caro-Alvaro, S., et al.: A systematic evaluation of mobile applications for instant messaging on iOS devices. Mobile. Inf. Syst. 1–17 (2017). https://doi.org/10.1155/2017/1294193
12. Caro-Alvaro, S., et al.: Identifying usability issues in instant messaging apps on iOS and android platforms. Mobile. Inf. Syst 2018, 1–19 (2018). https://doi.org/10.1155/2018/2056290
13. Inbar, O., Zilberman, B.: 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 '08), pp. 453–456. Association for Computing Machinery, New York (2008). https://doi.org/10.1145/1409240.1409312
14. Jadhav, D., Bhutkar, G., Mehta, V.: Usability evaluation of messenger applications for Android phones using cognitive walkthrough. In Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction (APCHI '13), pp. 9–18. Association for Computing Machinery, New York (2013). https://doi.org/10.1145/2525194.2525202
15. Nawi, M., Haron, N.S., Hasan, M.H.: Context-aware instant messenger with integrated scheduling planner, 2012 International Conference on Computer & Information Science (ICCIS), 900–907 (2012). https://doi.org/10.1109/ICCISci.2012.6297154
16. Perttunen, M., et al.: Experiments on mobile context-aware instant messaging, Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems, 305–312 (2005). https://doi.org/10.1109/ISCST.2005.1553327
17. Oghuma, A.P., et al.: An expectation-confirmation model of continuance intention to use mobile instant messaging. Telematics Inf. 33(1), 34–47 (2016). https://doi.org/10.1016/j.tele.2015.05.006
18. Zhou, T., Lu, Y.: Examining mobile instant messaging user loyalty from the perspectives of network externalities and flow experience. Comput Hum Behav. 27(2), 883–889 (2011). https://doi.org/10.1016/j.chb.2010.11.013
19. Swierenga, S.J., et al.: Mobile design usability guidelines for outdoor recreation and tourism. In: Nah, F.F.-H. (ed.) Proceedings in HCIB 2014: First International Conference HCI in Business. Held as Part of HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014, pp. 371–378. Springer International Publishing, Cham (2014)
20. Shitkova, M., et al.: Towards Usability Guidelines for Mobile Websites and Applications. Wirtschaftsinformatik Proceedings 2015, 1603–1617 (2015). https://aisel.aisnet.org/wi2015/107

21. Nouwens, M., Griggio, C.F., Mackay, W.E.: WhatsApp is for family; messenger is for friends: Communication places in app ecosystems. (2017). https://doi.org/10.1145/3025453.3025484
22. Naeem, M.M., Rafiq, A.: Usability Analysis of Instant Messaging Platforms in a Mobile Phone Environment using Heuristics Evaluation. International Journal of Scientific & Engineering Research, 10(7), 135–138 (2019). https://www.ijser.org/onlineResearchPaperViewer.aspx?Usability-Analysis-of-Instant-Messaging-Platforms-in-a-Mobile-Phone-Environment-using-Heuristics-Evaluation.pdf
23. Lamhaddab, K., Lachgar, M., Elbaamrani, K.: Porting mobile apps from iOS to android: A practical experience. Mobile. Inf. Syst, 2019. 1–29 (2019). https://doi.org/10.1155/2019/4324871
24. Conti, M., et al.: The dark side(-channel) of mobile devices: a survey on network traffic analysis. IEEE. Commun. Surv. Tutorials. 20(4), 2658–2713 (2018). https://doi.org/10.1109/comst.2018.2843533
25. Jabangwe, R., Edison, H., Duc, A.N.: Software engineering process models for mobile app development: A systematic Literature review. J Syst Software. 145, 98–111 (2018). https://doi.org/10.1016/j.jss.2018.08.028
26. Biørn-Hansen, A., et al.: An empirical study of cross-platform mobile development in industry. Wireless Commun. Mobile. Comput. 1–12 (2019). https://doi.org/10.1155/2019/5743892
27. Santos, A., et al.: Investigating the Adoption of Agile Practices in Mobile Application Development, In: Proceedings of the 18th International Conference on Enterprise Information Systems, pp. 490–497. SCITE-PRESS - Science and and Technology Publications, Rome (2016). https://doi.org/10.5220/0005835404900497
28. Losada, B., et al.: Combining InterMod agile methodology with usability engineering in a mobile application development. In: Proceedings of the 13th International Conference on Interacción Persona-Ordenador-interaccion. 12, pp. 1–8. ACM Press, Elche (2012). https://doi.org/10.1145/2379636.2379674
29. Choi, J., et al.: A framework facilitates development of a mobile app. Stud. Health. Technol. Inform. 250, 97–100 (2018). https://doi.org/10.3233/978-1-61499-872-3-97
30. Hoehle, H., Venkatesh, V.: Mobile application usability: Conceptualization and instrument development. MIS Q. 39(2), 435–472 (2015). https://doi.org/10.25300/misq/2015/39.2.08
31. Fu, C., et al.: Cross-Platform Instant Messaging System, pp. 27–30. (2015). https://doi.org/10.1109/WISA.2015.75
32. Kiat, B.W., Chen, W.: Mobile instant messaging for the elderly. Procedia Computer Science. 67, 28–37 (2015). https://doi.org/10.1016/j.procs.2015.09.246
33. Nielsen, J.: [Online] https://www.nngroup.com/articles/ten-usability-heuristics/
34. Bertini, E., et al.: Appropriating heuristic evaluation for mobile computing. Int. J. Mobile. Hum. Comput. Interact. 1(1), 20–41 (2009). https://doi.org/10.4018/jmhci.2009010102
35. Joyce, G., et al.: 2016, Mobile application usability: Heuristic evaluation and evaluation of heuristics, In: B. Amaba, (ed.), Advances in Human Factors, Software, and Systems Engineering: Proceedings of the AHFE 2016 International Conference on Human Factors, Software, and Systems Engineering, July 27-31, 2016, Walt Disney World®, pp. 77–86. Springer International Publishing, Florida. https://doi.org/10.1007/978-3-319-41935-0_8
36. Inostroza, R., et al.: Developing SMASH: A set of SMArtphone's USability heuristics. Comput. Stand. Interfac. 43, 40–52 (2016). https://doi.org/10.1016/j.csi.2015.08.007
37. Ahmad, N., Rextin, A., Kulsoom, U.E.: Perspectives on usability guidelines for smartphone applications: An empirical investigation and systematic literature review. Inf. Software. Technol. 94, 130–149. (2018). https://doi.org/10.1016/j.infsof.2017.10.005
38. Hussain, A., Mkpojiogu, E.O., Suleiman, K.: A heuristic evaluation of achik. Biz mobile shopping app. Int. J. Recent. Technol. Eng. 8(2S2), 123–126 (2019). https://doi.org/10.35940/ijrte.b1023.0782s219
39. Qu, Q.-X., et al.: 2017, User experience design based on eye-tracking technology: A case study on smartphone APPs, In: V.G. Duffy, (ed.), Advances in Applied Digital Human Modeling and Simulation: Proceedings of the AHFE 2016 International Conference on Digital Human Modeling and Simulation, July 27-31, 2016, Walt Disney World®, pp. 303–315. Springer International Publishing, Florida (2017). https://doi.org/10.1007/978-3-319-41627-4_27
40. Tzu-Ning, W., Po-Liang, C., Po-Lun, C.: Evaluate the usability of the mobile instant messaging software in the elderly. Stud. Health. Technol. Inf., pp. 818–822. (2017). https://doi.org/10.3233/978-1-61499-830-3-818
41. Android.com, Usability | Material Design, Material Desing [Online]. Available: https://material.io/guidelines/usability/accessibility.html# [Accessed: 29-Mar-2017] (2017)
42. Apple-Inc: IOS human interface guidelines. Apple Developer [Online]. Available: https://developer.apple.com/design/human-interface-guidelines/ios/(2019). Accessed 20-May-2019
43. Dye, S.M., Scarfone, K., A.: standard for developing secure mobile applications. Comput. Stand. Interfac. 36(3), 524–530 (2014). https://doi.org/10.1016/j.csi.2013.09.005
44. Johansen, C., et al.: The Snowden Phone: A Comparative Survey of Secure Instant Messaging Mobile Applications. (Authors' Version), arXiv:1807.07952 [cs] (2018). https://arxiv.org/abs/1807.07952
45. Florez, Z.J., et al.: Architecture of Instant Messaging Systems for Secure Data Transmision, pp. 1–7. (2016). https://doi.org/10.1109/CCST.2016.7815685
46. Koyuncu, M., Pusatli, T.: Security awareness Level of smartphone users: An exploratory case study. Mobile. Inf. Syst. 2019. 1–11 (2019). https://doi.org/10.1155/2019/2786913
47. Lai, I.K.W., Shi, G.: The impact of privacy concerns on the intention for continued use of an integrated mobile instant messaging and social network platform. Int. J. Mobile. Commun. 13(6), 641–669 (2015). https://doi.org/10.1504/ijmc.2015.072086
48. Pentina, I., et al.: Exploring privacy paradox in information-sensitive mobile app adoption: A cross-cultural comparison. Comput. Hum. Behav. 65, 409–419 (2016). https://doi.org/10.1016/j.chb.2016.09.005
49. Sahami Shirazi, A., et al.: Large-Scale Assessment of Mobile Notifications (2014). https://doi.org/10.1145/2556288.2557189
50. Vergara, E.J., Andersson, S., Nadjm-Tehrani, S., When mice consume like elephants: Instant messaging applications. (2014). https://doi.org/10.1145/2602044.2602054
51. Ayala, I., Amor, M., Fuentes, L.: An energy efficiency study of web-based communication in android phones. Sci. Program. 2019, 1–19 (2019). https://doi.org/10.1155/2019/8235458
52. Martin, C., Flood, D., Harrison, R.: A protocol for evaluating mobile applications, Information Systems Research and Exploring Social Artifacts: Approaches and Methodologies. 398–414 (2011). https://doi.org/10.4018/978-1-4666-2491-7.ch020
53. Miller, G.A.: The magical number seven, plus or minus two: Some Limits on our capacity for processing information. Psychol Rev. 63(2), 81–97 (1956). https://doi.org/10.1037/h0043158
54. Nielsen, J.: Error message guidelines. [Online]. Available: https://www.nngroup.com/articles/error-message-guidelines/(2001)
55. Google: Android Developers | Material Design. (Undated). https://material.io/develop/android
56. Mendoza, A.: Mobile User Experience: Patterns to Make Sense of It All. Morgan Kaufmann Publishers Inc., San Francisco (2013). https://dl.acm.org/doi/abs/10.5555/2553095
57. Rauch, M.: Mobile Documentation: Usability Guidelines, and Considerations for Providing Documentation on Kindle, Tablets, and Smartphones, IEEE 1–13 (2011). https://doi.org/10.1109/IPCC.2011.6087221
58. Christin, D., et al.: A survey on privacy in mobile participatory sensing applications. J. Syst Software. 84, 1928–1946 (2011). https://doi.org/10.1016/j.jss.2011.06.073
59. La Polla, M., Martinelli, F., Sgandurra, D.: A survey on security for mobile devices. IEEE Commun. Surv. Tutorials. 15, 446–471 (2013). https://doi.org/10.1109/SURV.2012.013012.00028
60. Vaziripuor, E., et al.: Is that you, alice? A Usability Study of the Authentication Ceremony of Secure Messaging Applications. Thirteenth Symposium on Usable Privacy and Security (SOUPS), pp. 29–47. USENIX Association, (2017). https://www.usenix.org/conference/soups2017/technical-sessions/presentation/vaziripour

61. statista: Mobile Operating Systems Market Share Worldwide from January 2012 to December 2019 (2020). [Online]. Available: https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/. Accessed 28 Feb 2020
62. Weichbroth, P.: Usability of mobile applications: A systematic Literature study. IEEE Access. 8, 55563–55577 (2020). https://doi.org/10.1109/ACCESS.2020.2981892
63. Parente Da Costa, R., et al.: Set of usability heuristics for quality assessment of mobile applications on smartphones. IEEE Access. 7, 116145–116161 (2019). https://doi.org/10.1109/access.2019.2910778
64. Flood, D., et al.: A Systematic Evaluation of Mobile Spreadsheet Apps., IADIS International Conference Interfaces and Human Computer Interaction 2011 (part of MCCSIS 2011). IADIS International Conference Interfaces and Human Computer Interaction – IHCI, pp. 217–224. (2011). http://www.iadisportal.org/digital-library/a-systematic-evaluation-of-mobile-spreadsheet-apps
65. Garcia, E., et al.: Systematic analysis of mobile diabetes management applications on different platforms, Information Quality in e-Health. pp. 379–396. (2011). https://doi.org/10.1007/978-3-642-25364-5_27
66. Martin, C., et al.: A systematic evaluation of mobile applications for diabetes management. In: Human-Computer Interaction – INTERACT 2011. IFIP Conference on Human-Computer Interaction, pp. 466–469. Springer (2011). https://link.springer.com/chapter/10.1007/978-3-642-23768-3_59
67. Nielsen, J.: Finding Usability Problems through Heuristic Evaluation. CHI '92: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI pp. 373–380. ACM. (1992). https://doi.org/10.1145/142750.142834
68. Nielsen, J.: Usability Engineering. Elsevier Science & Technology (1994). https://dl.acm.org/doi/abs/10.5555/2821575
69. Sauer, J., Sonderegger, A.: The influence of prototype fidelity and aesthetics of design in usability tests: Effects on user behaviour, subjective evaluation and emotion. Appl. Ergon. 40(4), 670–677 (2009). https://doi.org/10.1016/j.apergo.2008.06.006

How to cite this article: Caro-Álvaro, S., et al.: Applying usability recommendations when developing mobile instant messaging applications. IET Soft. 1–21 (2021). https://doi.org/10.1049/sfw2.12039