Protege: A mobile health application for the elder-caregiver monitoring paradigm

Fábio Ferreira\textsuperscript{a}, Flávio Dias\textsuperscript{a}, João Braz\textsuperscript{a}, Ricardo Santos\textsuperscript{a}, Roberto Nascimento\textsuperscript{a}, Carlos Ferreira\textsuperscript{a}, Ricardo Martinho\textsuperscript{a,b,*}

\textsuperscript{a}School of Technology and Management, Polytechnic Institute of Leiria, Leiria, Portugal
\textsuperscript{b}Computer Science and Communication Research Centre (CIIC), Polytechnic Institute of Leiria, Leiria, Portugal

Abstract

Population in modern societies is ageing, and the number of elders in complete isolation increases with this phenomena. Information technologies are being increasingly used to enhance remote communications and monitoring between elders and caregivers. However, these technologies seldom are adapted to elders’ real needs and functional limitations, and are often discarded by them. They are usually hard to use and lack a set of features that elders perceive as useful to their daily-life communications with their caregivers. In this paper we present Protege: an Android mobile application for the elder-caregiver paradigm, with a fully customized user interface. The integrated features of Protege and its fully customized user interfaces allow for an efficient way of enhancing elder-caregiver communications, and provides elders with a safety-critical device that they can easily use to call for help.

Keywords: elder-caregiver paradigm, mobile health, Android app, remote patient monitoring

* Corresponding author.
E-mail address: ricardo.martinho@ipleiria.pt
1. Introduction

The increased use of assistive Information Communications and Technologies for home based long-term care has great potential to increase both caregiver and elder independence, alleviate the caregiver’s and/or family burden, reduce unmet need, and lower expenditures [1]. Regarding the elder-caregiver paradigm, it is of most importance to the elder to have a mobile device that can assure her/him of reliable communications either to engage in simple conversations or to quickly ask for help. For the caregiver, it is essential to enable remote monitoring of the elder’s activity (movement, position) and status (physical and emotional) through the same mobile device.

The elder’s receptivity towards a device is a common problem [2], which also emerges when developing a dedicated mobile software application. Nevertheless, we observed through interviews that this problem is easily surpassed by the elder’s willingness to overcome isolation, and also by a feeling of extra safety (e.g., to be able to get aid faster).

In this paper we present an Android OS mobile application called Protege. It was developed specially for the elder-caregiver paradigm, with its main features identified and refined within a population of elders and caregivers. Features that require elders’ interaction include a special launcher, an SOS feature and dedicated user interfaces for short messaging (SMS) and phone. These features grouped together compose an integrated elder-customized mobile application developed for a refined, intuitive and adjusted user interaction experience.

Regarding caregiver’s actions, we developed an SMS server that allows her/him to monitor the elder’s device. This includes retrieving battery level, GPS location, movement activity and quick elder’s answers to predefined status questions (e.g., “how are you?”).

Protege also includes alert features, such as “Fall alert”, “No activity” and “Low battery” alerts. The first two are based on readings from a mobile device’s accelerometer, but aimed at different purposes. “Fall alert” matches those readings with fall detection patterns and, in case of a positive match, triggers an SOS request. “No activity” request the elder for a quick status answer, in case of long time periods of inactivity.

These alert-based autonomous features also serve as a mean to increase remote elder-caregiver interactions, as they are complementary regarding the individual actions that each caregiver and elder are able to perform with Protege. They also provide some degree of real-time remote monitoring of an elder, which can be of most importance, especially in emergency situations.

Another achievement that we can report is the fact that Protege comprises features that, although offered by some other applications, cannot be found in a seamlessly integrated fashion, downloadable from “Google play” and installable in any smartphone with Android OS v4.03 or higher.

This paper is organized as follows: section 2 refers to related work and section 3 explains the methodology adopted for the development of our Protege app. Section 4 depicts our system architecture using a UML deployment diagram, and section 5 describes the main features of our app. Finally, section 6 concludes the paper and presents some future work.

2. Related work

Population ageing in developed societies is, nowadays, a well-known fact [3]. In Portugal, recent data shows that by 2012 there were 400,000 elders living by themselves, and another 804,000 living exclusively with other elders [4]. Adding these two figures totals over 1.2 million elders in isolation, representing over 10% of the Portuguese population.

In spite of the existence of some software products with features that aim to reduce this isolation and to alert caregivers of emergency situations (such as SOS requests, fall alerts, dedicated user interfaces or even GPS location requests), these products come up as non-integrated features, and pose serious usability difficulties to elders and caregivers.
In [5], the authors present the One Touch SOS app (for Android, Windows Phone, Java ME and Nokia mobile platforms) that enables SOS requests with GPS location. However, it requires users to open the application, and does not include SOS calls or integration with other features (such as fall alerts).

The apps referred in [6] and [7] present adapted user interfaces for elders/people with vision problems. They run as launchers (default user interfaces for mobile devices), and contain large soft buttons and texts. They allow phone calls, texting and quick access to applications, photo album and phone camera. Big Launcher in [6] has also an SOS feature that send GPS location coordinates and calls a caregiver, but none of them present remote monitoring features.

The default Android Launcher/Dialer/SMS features in [8] do not enhance elder’s usability. We’ve observed this during our preliminary interviews with elders, where difficulties in the use of these features came up frequently, even in elders that already had an Android-based mobile device.

The WRU? Lite app in [9] is an Android service that replies to a location request made by SMS. To enable these requests, the user of the (monitored) mobile device has to add permitted (location requesters) phone numbers to a whitelist. It does not allow the monitoring of other indicators such as a mobile device’s movement activity.

In spite of presenting good examples of elder-caregiver support features, these products are very restricted concerning the aim of an integrated and all-purpose mobile app. This is precisely the objective of our Protege Android app. The next sections explain the methodologies we used in finding which features were most valuable together and integrated, and the final results of our integrated mobile app.

3. Methodology

In the beginning of our study, one of our greatest difficulties was the fact that we could not ask elders what they expect from an Android mobile application/device. This is due mainly to their relative level of ignorance regarding the full possibilities of this kind of technologies. Therefore, we began by researching similar products and brainstorming about elder-related limitations towards the usability of Android mobile devices. From this process of research and brainstorming, we concluded the following:

• We need to work on the user interface so that the elder uses the device/app with no limitations and sees it as something useful;
• Caregiver’s remote monitoring features (those not requiring direct interaction by the elder) are only useful if the device/app is effectively used by the elder;
• Our mobile app power-user that will engage the elder to effectively use it will definitely be the caregiver, so we better add it real value for her/him;
• The caregiver will definitely be the power user responsible for promoting the effective use of the app by the elder, so we better add real value to her/him;
• The extra well-being that our app can add to the elder-caregiver relationship will be proportional to its communication, understandability and usability degrees;
• From a communication technology and economic point of view, GSM and texting communications are better in Portugal, due to Wi-Fi/3G/4G insufficient coverage and higher costs, and to SMS wide spread use among the Portuguese.

We then interviewed caregivers and professionals working in elder care. With the results from brainstorming and from these interviews, we came up with a first set of potential features that we validated and prioritized among elders, promoting a close participation of both caregivers and elders in our project.

The results of this overall process, including effort estimates and prioritization are resumed in Table 1.

Table 1. List of features to be integrated in our Protege mobile app.
Table 1’s list of features is based on eXtreme Programming (XP) agile software development method [10], where features are defined and prioritized by “customers”, and effort estimates are performed by the development team. The last (italicized) 5 features were not included for selection by customers (interviewed elders and caregivers) since they were considered as basic dependencies regarding the remaining ones.

As observed, safety-critical features are the most preferred by both caregivers and elders (SOS and FallAlert), which reveals a strong concern about this issue. Therefore, and also part of XP, we performed release planning meetings and, taking into account the fixed resources and time to develop (5 months, 5 team members), decided to separate the product into 2 releases:

- 1st release (2 months): including Launcher, Phone, Settings, and SMS basic features and the SOS feature;
- 2nd release (3 months): including SMSServer, FallAlert, NoActivityAlert, LowBatteryAlert, WhereAreYou and HowAreYou.

The remaining features referred in Table 1 were not included in these 2 releases, due to time/effort-required issues. Release planning had as a strategy to present a first yet quite usable mobile app, in order to assess user acceptance under real-world circumstances and, eventually, to reprioritize features. For each feature, we let our customers to define a first “user story” draft (a requirement written by customers in natural language), and defined all acceptance tests in advance. Following we derived and coded each corresponding unit test and, when finished, validated the feature with our customers, against the previously defined set of acceptance tests. The next section describes the overall architecture of our Protege app, with each of the features identified as dependable components.

4. System architecture

In Figure 1 we draw an UML Deployment diagram containing the main physical and logical components of our Protege mobile application. We can see that this architecture is composed by two nodes which represent the
caregiver (left) and elder’s devices (right). The caregiver’s device can be any non-customized GSM-enabled mobile device, from low-end text-based interface devices to high-end smartphones.

As for the right node, it represents the elder device where all Protege features will be running, with all dependencies already identified in Table 1. Mainly, most of them depend on the Launcher component that, in turn, is Android OS dependent as well. It should be noted that an elder’s device may be monitored by more than one caregiver’s device and vice-versa.

5. Implemented features

From the list of basic and selected features, we developed our Protege Android mobile app. Due to dependency issues, we began by developing the basic ones, and then followed with the remaining highest prioritized.
5.1. Launcher, SOS, SMS and Phone

In order to evaluate the default (launcher) user interface of our Protege Android mobile app, we developed several functional prototypes to be tested by elders. These tests allowed us to define some useful design guidelines that we applied to the remaining features’ user interfaces. These guidelines included:

- Elders could not successfully distinguish screen “long presses” types of interaction from a normal press. Therefore, we should avoid this kind of distinction in our user interaction design;
- Every clickable interface components that have actions associated (namely buttons), should be large and colorful, using a consistent and easy to follow color code;
- Interface information components (or other non-action ones) should not be colorful, so that they cannot be confused with clickable ones;
- Every clickable component should be at least of the size of a thumb, since we noticed elders lacked precision and were very error-prone when clicking in components of smaller sizes;
-Scrolling actions should not co-exist within the same areas as clickable components (e.g. do not scroll a contact list – add “next” and “previous” buttons above or below each screen);
- Actions classified as very important (e.g. activating/canceling an SOS request), should not be triggered by an accidental press on the screen, in order to avoid false positives;
- Graphics and icons should be preferred over text – they make actions more understandable and do not demand reading abilities, which should be avoided specially for elders with vision problems;
- Navigation should not be supported on the “stack” paradigm of Android, which uses the “back” button to return to the previous activity. This requires an extra abstraction ability and state memory which is not adequate for elders;
- Exceptions to these guidelines apply to caregiver’s interfaces, such as Settings or retrieving the full list of apps installed in the device.

Interface design of Protege reflects, therefore, these guidelines and the results of several tests performed with elders. In Figure 1 we illustrate the interface of the developed Launcher activity. It includes easy access to system information such as time, date, network and battery, and direct access to most used apps such as camera, phone and SMS, as well as to a list of applications that elders can use. The phone and SMS buttons blink whenever there are unseen missed calls and/or SMS messages. Following the guidelines above, the SOS request action is performed by a swipe gesture (and not accidentally by a normal press button), in order to avoid false positives, and to allow for immediate, first front-end availability.

The gray bottom button triggers access to apps that the elder can use, which are configured by the caregiver through a default “Settings” Android interface. This way, we assure that an elder can execute third-party apps, and does not access to sensitive settings that could alter the required behavior of Protege (e.g. turning off the GPS location setting).

Settings access is protected against a PIN code setup by the caregiver. They allow enabling/ disabling/ configuring features, providing a extra adaptability degree for Protege, adjustable to elder-caregiver pairs with distinct user needs.

The SOS request feature development followed Launcher (see Figure 2, on the right). Our algorithm for this SOS feature can be resumed to: 1) tries to obtain GPS location; 2) sends SMS with GPS location to caregiver; 3) makes call to caregiver (after a setup period of time).
SMS sent to the caregiver includes location, time, a customizable text message, a link to Google maps, how location was obtained and its precision in meters. It is possible to enable/disable an automated emergency call in the SOS settings. When enabled, the mobile device will make a call to a predefined number (usually the caregiver’s) in speakerphone mode after the SMS is sent. The device will try to obtaining a GPS location in case of it being unable to do so within the predefined SMS sending time.

Figure 3 shows the adapted interfaces for the Protege Phone feature. The green (phone) button on Launcher’s interface navigates the user to the first left screen with a list of favorite contacts, disposed in large, picture-based layout. The purple phone button on this interface navigates to the next screen of calls history, while the “A-Z” blue button accesses to a list of contacts indexed by initial letter (3rd screen counting from the left). The gray bottom button on the 2nd screen activates a large button dialer, for customized phone numbers (4th screen).
Fig. 3. Screens of the adapted Phone feature of Protege.

The SMS adapted interfaces follow a similar coherent interaction flow, as it can be observed in Figure 4. Besides being able to SMS in free text (first left “pencil” upmost yellow button on the 2nd screen from the left), we added 3 predefined messages to be sent to the caregiver (remaining yellow buttons): “Call me, please!”, “You can pick me up now!” and “I’m fine, thank you!”.  

Fig. 4. Screens of the adapted SMS feature of Protege.

5.2. **SMSServer, NoActivityAlert, LowBatteryAlert, WhereAreYou, HowAreYou and FallAlert**

For our SMSServer feature, we developed our own protocol over SMS, which resembles a remote shell with commands following a `<PIN> <command>` format, within an SMS message. This feature allows a caregiver to
send SMS commands to an elder’s device without the elder’s intervention (and/or knowledge). The list of commands was refined in interviews conducted among caregivers and elders, and includes the following commands:

- COM – lists the possible commands and associated short descriptions;
- GPS – retrieves a message with the elder’s device GPS location;
- BAT – retrieves a message with the battery level of the elder’s device;
- OK – activates the HowAreYou feature in the elder’s device, and send a confirmation message when the elder answers with a positive action;
- LOC – retrieves a list of the last 5 known GPS locations (stored every hour in the elder’s device), which composes the “WhereAreYou” feature;
- MOV – retrieves a list of the elder’s device movement activity for the past 5 hours.

The HowAreYou feature contains a simple and SOS-like screen (Figure 5), and is activated through an SMSServer command such as 0123 OK, sent through the caregiver’s device by SMS to the elder’s device.

It times out after 30 seconds, and reactivates after a configurable time. This avoids caregivers having to remember to resend the SMS message, and allows a quick interaction between the caregiver and the elder, avoiding accessing SMS texting for a simple answer to a simple question.

The NoActivityAlert and LowBatteryAlert features can also be enabled/disabled in the Settings screen. When enabled, they also use the SMSServer feature to send the caregiver an appropriate alert SMS message.

FallAlert was definitely the feature that elders requested the most, right after the SOS feature. It allows a mobile device to detect when its human carrier has possibly fallen and immediately start an SOS request to the caregiver. For this feature, we first created an isolated application in order to test and identify falling patterns of acceleration, when compared with normal movements of walking, sitting, running, going in and out of vehicles, and other common day-to-life body movements. Figure 6 identifies the four key moments of a fall: Fall, Hit,
Bounce and Rest. These moments derived in about 11 parameters that we analyze in order to detect a possible fall.

![Graph](image)

Fig. 6. Four key acceleration moments of a fall.

After several real-world tests and statistical analysis to accelerometer-measured data, we calculated the limits for the 11 parameters derived. Nevertheless, our strategy regarding this feature included 2 important assumptions: 1) provide a first moment confirmation screen in order for the elder to confirm a fall (Figure 7, left) and; 2) allow for a caregiver to adjust fall sensitivity through simple settings menu (Figure 7, right).

![Confirm Screen and Settings Menu](image)

Fig. 7. FallAlert confirming screen (left) and sensitivity configuration (right).

Being FallAlert a part of Protege integrated mobile app, when Figure 8’s left screen action times out, an SOS request is immediately initiated. Additionally, this confirmation feature puts the need of a 100% accurate fall detection algorithm into 2nd place, since false positives can easily be identified by the feedback provided to the caregiver.
Full use of these features increase battery consumption, since some of them require that the device never enters a “deep sleep” state. We’ve managed to get an elder’s device battery to endure for a full day, with all features turned on. We also identified issues regarding to some devices preventing measurement of accelerometer data during stand-by “off screen” modes, so it is important to take device compatibility into account.

6. Conclusions and future work

We’ve presented in this paper an integrated Android mobile application called Protege, which aims to bridge some communication gaps that can easily occur within the elder-caregiver paradigm. We’ve performed several interviews and tests in order to identify which features should be part of our solution, and to refine usability issues associated with the elders. The final version is in effective use by the elders and caregivers that helped us in this development process, and we’re monitoring their difficulties in using Protege.

Regarding future work, our app can go further by integrating new Bluetooth enabled wearable technologies for dealing with biometric data such as body temperature, heartbeat or any physical movement monitoring. Permanent access to the Internet by 3G/4G connections can also enhance real-time elder monitoring, enabling dashboard web interfaces for caregivers, or even video-calls and remote configuration of the elder’s mobile device.

References