Information and Software Technology 146 (2022) 106841

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

An evaluation of the effectiveness of personalization and self-adaptation for e-Health apps

Eoin Martino Grua [a,*], Martina De Sanctis [b], Ivano Malavolta [a], Mark Hoogendoorn [a], Patricia Lago [a,c]

[a] *Vrije Universiteit Amsterdam, The Netherlands*
[b] *Gran Sasso Science Institute, L'Aquila, Italy*
[c] *Chalmers University of Technology, Sweden*

ARTICLE INFO

Keywords:
Self-adaptive systems
Personalization
Reference architecture
Mobile apps
e-Health

ABSTRACT

Context. There are many e-Health mobile apps on the apps store, from apps to improve a user's lifestyle to mental coaching. Whilst these apps might consider user context when they give their interventions, prompts, and encouragements, they still tend to be rigid *e.g.,* not using user context and experience to tailor themselves to the user.
Objective. To better engage and tailor to the user, we have previously proposed a Reference Architecture for enabling self-adaptation and AI personalization in e-Health mobile apps. In this work we evaluate the end users' perception, usability, performance impact, and energy consumption contributed by this Reference Architecture.
Method. We do so by implementing a Reference Architecture compliant app and conducting two experiments: a user study and a measurement-based experiment.
Results. Although limited in the number of participants, the results of our user study show that usability of the Reference Architecture compliant app is similar to the control app. Users' perception was found to be positively influenced by the compliant app when compared to the control group. Results of our measurement-based experiment showed some differences in performance and energy consumption measurements between the two apps. The differences are, however, deemed minimal.
Conclusions. Our experiments show promising results for an app implemented following our proposed Reference Architecture. This is preliminary evidence that the use of personalization and self-adaptation techniques can be beneficial within the domain of e-Health apps.

1. Introduction

In recent years e-Health apps have become more popular and have now a projected market growth to US$102.3 Billion by 2023 [1]. E-Health mobile apps are designed to aid the end user by providing a wide range of services that can help improve users' lifestyles [2]. Considering their spread and popularity, they might play a relevant role in the context of Decision Support Systems (DSSs) for health care, whose aim is to help patients manage their health care by providing accessible and reliable health services [3]. E-Health apps have some components that make them unique compared to other health-related systems *i.e.,* (i) can take advantage of smartphone sensors, (ii) can reach an extremely wide audience with low infrastructural investments, and (iii) can leverage the intrinsic characteristics of the mobile medium (*i.e.,* being always-on, personal, and always-carried by the user) for providing timely and in-context services [4]. However, even with all of these tools available to them, e-Health apps still tend to be *rigid* and not tailored in their interventions and prompts to the user, *e.g.,* the apps are using a fixed rule set to construct their interventions and not considering unique traits and behaviors of the individual user. In the literature, several works exploit self-adaptation techniques or personalization techniques to keep users engaged, in e-Health mobile apps. For instance, self-adaptation policies are used to dynamically self-configure the internal behavior of a wearable patient-monitoring system for tele-rehabilitation, based on the current context of the patient [5]. Self-adaptation is also exploited for delivering persuasive messages stimulating the medication adherence, by using real-time

* Corresponding author.
E-mail addresses: e.m.grua@vu.nl (E.M. Grua), martina.desanctis@gssi.it (M. De Sanctis), i.malavolta@vu.nl (I. Malavolta), m.hoogendoorn@vu.nl (M. Hoogendoorn), p.lago@vu.nl (P. Lago).

https://doi.org/10.1016/j.infsof.2022.106841
Received 30 April 2021; Received in revised form 29 October 2021; Accepted 7 January 2022
Available online 5 February 2022

physiological data (*e.g.,* heart rate) [6]. Other self-adaptive applications are intended for healthcare professionals and caregivers to deliver the right patient's information at the right time under variable connectivity and limited resource availability [7,8]. Personalization, instead, has been mainly exploited for clustering users according to specific variables about their physical and psychological well-being [9], and for reducing the drop-out rates and increase the patience adherence to treatment [10,11]. However, none of existing approaches exploit both self-adaptation and personalization techniques to get the most from their combination. Moreover, according to [3], DSSs show some open issues about (1) the exploitation of big data analytics to gain knowledge from users/patients for healthcare professionals, (2) the exploitation of IoT to extract insights from remote monitoring data, and (3) the provisioning of patient-centered analytics to improve treatments, by making them more accurate, efficient and personalized.

To address these problems, we previously proposed a *reference architecture (RA) that combines data-driven personalization with self-adaptation* [12,13]. In this paper we extend on this research line by:

- utilizing our RA to guide the implementation of an app.
- designing and conducting a user study to investigate end users' concerns related to both usability and perception of an app complying to our RA.
- designing and conducting a measurement-based experiment to investigate the impact on performance and energy consumption that an app complying to our RA has.
- discussing the newly found results and frame them in the broader context of e-Health mobile apps and the usage of personalization and self-adaptation techniques in this domain.

We conduct two experiments that investigate some concerns that developers and end users of our implemented app would have. To this end, we have formulated four main research questions to empirically assess the impact of personalization and self-adaptation from (i) the users' perspective and (ii) the system perspective. Our experiment results show that for the user perspective personalization and self-adaptation techniques have an overall positive impact on the end users' perception of e-Health mobile apps. We saw no apparent impact of these techniques on usability of e-Health mobile apps. From the system perspective our results have found some statistically significant differences in app performance. These differences are too small to realistically impact the user experience of an Android app. Furthermore, our experiments provide evidence that the impact of personalization and self-adaptation on energy consumption of e-Health mobile apps is negligible.

The paper is structured as follows. The next section briefly outlines the RA presented in our previous work. Section 4 describes our study design, with Section 3 showing how an app was implemented following the guidelines of our RA, and Sections 4.1 and 4.2 describing the design of our experiments. Section 5 explains the results for both experiments. In Section 6 we discuss the results. Section 7 explains the threats to validity. Section 8 describes the related work. Lastly, Section 9 concludes the paper.

2. Reference architecture

In this work we evaluate the implementation of our RA for personalized and self-adaptive e-Health apps, whose preliminary version was presented in [12] and then extended in [13]. The RA combines *personalization* [14] and *self-adaptation* [15] as effective instruments for getting users continuously engaged and active, eventually leading to better physical and mental conditions. Specifically, the RA simultaneously supports (i) *personalization* for the different users, by exploiting the users' smart objects and preferences to dynamically get data about *e.g.,* their mood and daily activities, and (ii) *self-adaptation* to the user-needs and context, such that to improve the usability of e-Health apps thus keeping users engaged and active. The RA, exploits an **online clustering algorithm** [16] for efficiently managing the evolution of the users behavior, a **dedicated goal model** [13] for representing health-related goals, and multiple *Monitor - Analyze - Plan - Execute* (**MAPE**) loops [17] managing adaptation at different levels and for different purposes.

Fig. 1 shows our RA [12]. For clarity and to make this study self-contained, we summarize hereby the architecture components and their main functionalities. The RA is made by two main macro-components, the *e-Health app* running on the *User*'s smartphone and the *Back-end* hosting components supporting the general functioning of the app. The back-end is managed by a *Development team* and it further exposes an interface to the *Domain Expert* (*e.g.,* psychologists), whose smooth participation is supported.

For personalization and self-adaptation reasons, our RA takes into account the *Environment* in which the user lives and uses the e-Health app as well as possible *Smart Objects* that the user owns and adopts (*e.g.,* smart bracelet, smart-watch). Precisely, the environment represents the physical location of the user, exploited by the e-Health app to make self-adaptations according to its current operational context (*i.e.,* environmental conditions and weather) and to the user's scheduled activities. Smart objects, instead, are devices that the app can communicate with, used to gather additional data about the users, thus augmenting the data collected by the smartphone sensors (*e.g.,* a smart-watch providing extra information about the user's heart-rate).

Through the Internet, the e-Health app sends collected data to the *AI Personalization* back-end component. The communication from the back-end to the e-Health app, instead, is performed by the *User Process Handler* which is in charge of sending the *User Process* to the e-Health app via push notifications. In fact, our RA exploits a particular type of *User Process* that is composed of *Abstract Activities*, namely a vector of *Activity categories* (one per weekday) and an associated goal. Each Activity category identifies the kind of activity the user should perform (*e.g., Cardio* or *Strength* Activity category), whereas each goal labeling an abstract activity is defined in terms of a dedicated *goal model* for representing health-related goals via a descriptive concise language accessible by healthcare professionals (*e.g.,* fitness coaches, psychologists). Then, for each user, the Activity categories are converted to *Concrete Activities* at runtime via the use of the *User Driven Adaptation Manager* and based on the user's preferences. For instance, a cardio Activity category can be instantiated into different *Concrete Activities* such as running, swimming and walking.

The *User Process Handler* is, then, in charge of providing *User Processes*, the same User Process to all users of the same cluster. In fact, our RA exploits an online clustering algorithm (implemented by the *AI Personalization* component) for efficiently managing the evolution of the behavior of users as multiple time series evolving over time [16]. Clustering allows the RA to group similar users together, in a real-time and online fashion; where similarity is determined by the data gathered from the apps. This gives our RA a more sustainable, scalable, faster and data efficient method of *personalization*, without requiring to create individual personalization strategies (requiring more data from each individual, gathered over a longer time period), maintaining a suitable degree of personalization [16,18].

Lastly, our RA has five components used for self-adaptation. Each of these components implements a MAPE loops [17] operating at different levels of granularity and for different purposes. Specifically:

- the *AI Personalization Adaptation* monitors the evolution of clusters, detects if any change occurs and enables the creation of new User Process (*e.g.,* a new cluster is generated);
- the *User Driven Adaptation Manager* receives the User Process and refines the Abstract Activities into Concrete ones. It suggests users the most suitable and timely activities according to their (evolving) health-related characteristics (*e.g.,* active vs. less active), and to technical aspects (*e.g.,* smart objects own by the user);

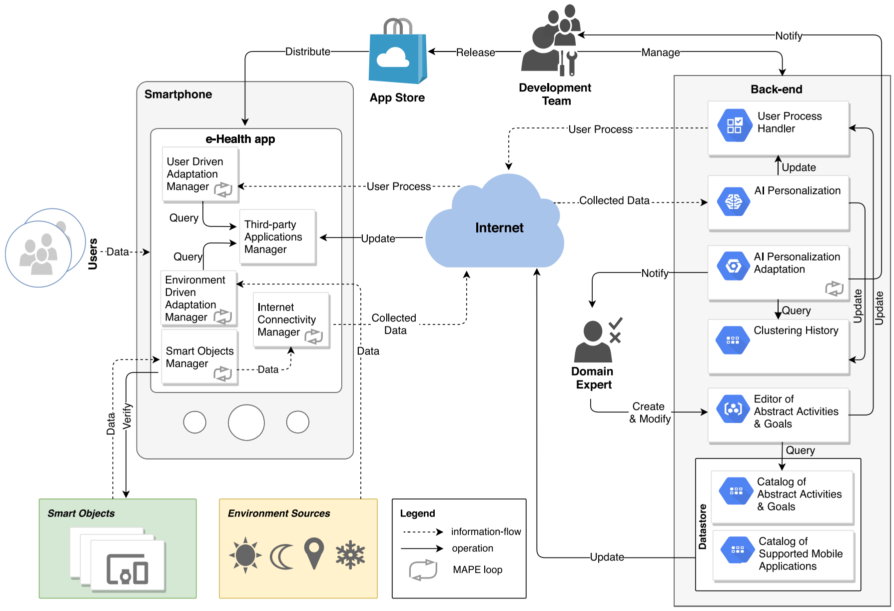


Fig. 1. Reference architecture for personalized and self-adaptive e-Health apps [12].

- the *Smart Objects Manager* maintains the connection with the user's smart objects and, if not possible, finds alternative sensors to make the e-Health app able to continuously collect user's data;
- the *Internet Connectivity Manager* sends data to the back-end and provides resilience to the e-Health app's internet connectivity;
- the *Environment Driven Adaptation Manager* to cope with characteristics of the physical environment (*e.g.,* indoor vs. outdoor, weather) with the aim of keeping the users constantly engaged, to ensure that they execute their planned schedule of activities.

The RA can be applied either in the context of a single e-Health app or by integrating the services of third-party e-Health apps (*e.g.,* already installed sport trackers). For a detailed description of the RA, its components and the functioning of each single MAPE loop, we refer the interested reader to [13].

3. Implementation of the e-Health app

In this section we describe the implementation of the RA compliant e-Health app, named RELATE, that we used for our experiments. RELATE is implemented in Android, as Android mobile devices cover the majority of the mobile device sector and the majority of scientific research on mobile software engineering is done on Android [19,20]. For an explanation of the app's flow the reader is directed to the online material in the **replication package**.

Fig. 2b shows RELATE's architecture, whose mapping with the RA components is shown in Fig. 2a. There are three activities in RELATE that together form the UI: the MainActivity, the SettingsActivity and the FirstScreen activity.

– *FirstScreen*. This is the first activity displayed to the user. The sole responsibility of this activity is to present the user with the list of available physical activities and have them choose their preferred ones. After they have made their preference they are redirected to the Main Screen, which is managed by the MainActivity.

– *MainActivity*. This activity is in charge of displaying the Main Screen to the user, as well as instantiating and communicating to most other components present in RELATE. It is also from here that the user can choose to access the settings.

– *SettingsActivity*. This activity is in charge of displaying the app's settings to the user and redirecting them to either adjust their preferred physical activities, read the about page or go back to the Main Screen. Whenever the user makes a change to their preferred activities, the SettingsActivity stores the preferences locally, so that they are available even after the application has been closed by the user.

RELATE contains two services on the user side: the User Driven Adaptation Manager and the Internet Connectivity Manager.

– *User Driven Adaptation Manager*. This service has two main responsibilities: it creates a unique identifier token at installation which it sends to the Back-end and it converts each User Process received from the Back-end. The unique token is used by the Back-end to send the User Process to the correctly paired user. The conversion of the User Process is done by the User Driven Adaptation Manager in accordance to the self-adaptive loop described in Section 2.

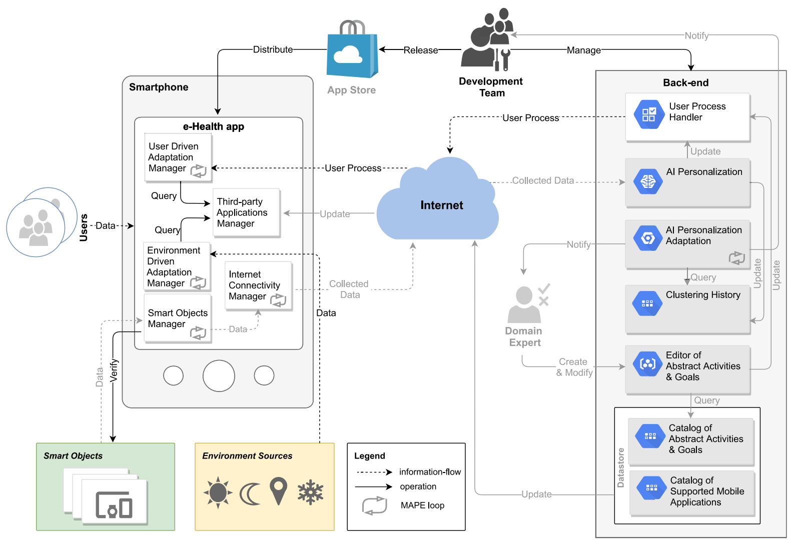
The responsibility of sending the token to the Back-end is a deviation from the RA. In the RA the only component to send information to the Back-end is the Internet Connectivity Manager. This change was made to optimize the information flow of RELATE. As with the current implementation of RELATE we do not have the AI Personalization in the Back-end, we decided to use the already created information flow from the User Process Handler to the User Driven Adaptation Manager and add the task of receiving the user token.

– *Internet Connectivity Manager*. This service is started by the MainActivity whenever the app is opened in the foreground. Its main purpose is to monitor and manage the connection to the internet via its self-adaptive loop as described in Section 2.

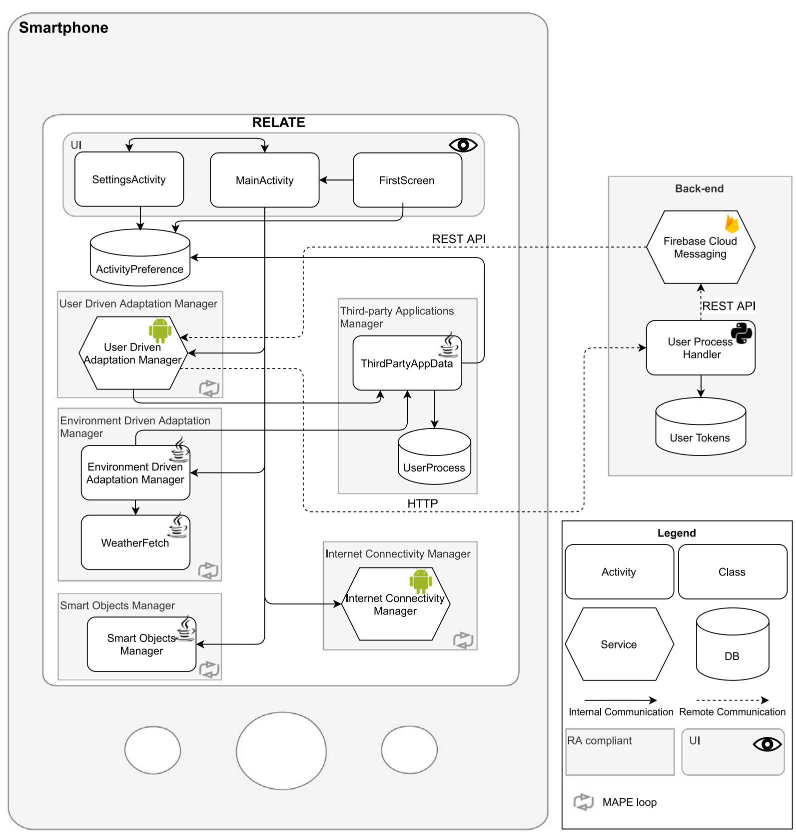
RELATE contains four classes: Smart-Objects Manager, Environment Driven Adaptation Manager, ThirdPartyAppData, and WeatherFetch.

– *Smart Objects Manager*. This class is initialized by the MainActivity whenever the app is launched and has two main purposes: ask the user for the runtime permission for the Bluetooth usage and, monitor and manage the connection to external devices via the self-adaptive loop, in accordance to the RA's description.

– *Environment Driven Adaptation Manager*. This class is also initialized by the MainActivity whenever the app is launched by the user. It has two main purposes: to check what the weather forecast is for the current day and to convert the daily suggested activity if the current user environment calls for it. The class determines the daily weather forecast with the help of WeatherFetch. The Environment Driven Adaptation



(a) All of the RA components implemented by RELATE (the gray ones have been omitted)



(b) RELATE architecture

Fig. 2. Figures describing the components of the RA used and the RELATE architecture.

Manager can also perform a change in suggested activity, as described in the RA, which it then displays to the user via a push-notification.

– *WeatherFetch*. The main responsibility of this class is to determine the weather forecast and deliver that information to the Environment Driven Adaptation Manager. To determine the forecast, it uses the OpenWeather API[1] to retrieve a .json file containing information on the weather forecast for that day. It processes the file and sends the

[1] https://openweathermap.org/api.

parsed information to the Environment Driven Adaptation Manager (*e.g.,* Sunny, Rain, Windy, Storm, etc.).

– *ThirdPartyAppData*. This class is a helper class to the User Driven Adaptation Manager and the Environment Driven Adaptation Manager in the conversion of the received UserProcess from the Back-end to a schedule of concrete activities displayed to the user. In this version of RELATE, this class did not interact with other third party apps as described in the RA.

Lastly, the Back-end *User Process Handler* was implemented using Flask in Python. The Flask server would receive the initial unique identifier token sent by the User Driven Adaptation Manager and store it in the *User Tokens* database. The User Process Handler then sends the weekly user process to the app via the use of Google's Firebase Cloud Messaging.[2] As we did not have a Domain Expert involved in these Experiments, the User Process was fixed and saved in the same class file as the User Process Handler. Furthermore, the other components of the Back-end were not implemented in this version of RELATE. Whilst in the future we would want to include all components to our Experiments, for these Experiments we focused our efforts on the application side of the RA, as it is most relevant to our current research questions.

For both Experiments we also used a BaseApp as a comparison to RELATE. It is identical to RELATE apart from not including: the Environment Driven Adaptation Manager, the Internet Connectivity Manager, and the Smart Objects Manager. The BaseApp is therefore not able to provide the functionalities offered by those components. By not having the Environment Driven Adaptation Manager the BaseApp cannot adapt the daily physical activity to better suit the user's current environment and will not show the related push notification. Without the Internet Connectivity Manager, the BaseApp is unable to detect and automatically resolve a failure to connect to the internet, as well as notify the user of such failure. By excluding the Smart Object Manager, the BaseApp is unable to verify the current status of the Bluetooth connection or amend problems that may occur with said connection. As most of the excluded functionalities work without the user's direct involvement, the two applications are aesthetically identical as they contain all of the same screens.

4. Study design

As shown in Fig. 3, our study is composed of three main phases, namely: the instantiation of the RA, the user study (Experiment 1), and the measurement-based experiment (Experiment 2). We describe each step of all phases, its objective, expected input and output, and number of involved researchers.

The goal of the **RA instantiation** phase is to design and develop an instance of the RA, which is used in the two experiments. This phase is composed of three main steps: the identification of the features of the app (step 1.1), its implementation for the Android platform (step 1.2), and its piloting (step 1.3).

– *Features identification (step 1.1)*. This step is conducted by all five researchers and has the goal of identifying the features that need to be present in the app implementation. This activity is carried out by taking into consideration our need of keeping the app reasonably simple (so to be used by multiple participants without requiring extensive training), while still having room for personalization and self-adaptation at runtime. The main output of this step is the list of the app features:

- **F1.** The app needs to provide a list of weekly physical activities to the user.
- **F2.** The user is able to select their preferred physical activities from a list of available ones.
- **F3.** The app has to be able to determine the environment of the user.
- **F4.** The app needs to change recommended physical activities in accordance to the user environment.

– *App implementation (step 1.2)*. The implementation is named RELATE, standing for peRsonalized sELf-AdapTive E-health. RELATE is implemented in Android and its back-end is implemented in Python. The details of the implementation process are discussed in Section 3.

– *Piloting (step 1.3)*. We pilot the implemented app in order to ensure that it can be successfully used in the two experiments. Two researchers different from the one implementing RELATE are involved in this step and they carry out the piloting activities independently from each other. Each researcher installs the app on their mobile device and simulates typical usage scenarios according to the features identified in step 1.1. During usage, they note down apparent bugs and problems and discuss them with the researchers implementing the app. The app would then go back to implementation, to correct the found issues. This cycle continues until the app is deemed ready to be used for the experiments. This step took a total of 14 days.

Once the implementation of the RELATE app is completed and piloted, we can proceed with the design and execution of the two experiments. The complementary nature of the two experiments allows us to carry them out in parallel.

We organize the **user study** into four main phases: the design of the user study, subjects selection, the execution, and the data analysis. Below we describe how the four phases fit together, whereas their detailed description is given in Section 4.1.

– *Design of user study (step 2.1)*. The goal of this phase is to design a user study that would allow us to understand the impact of personalization and self-adaptation techniques on the usability and end users' perception of our e-Health mobile app. The design is carried out collaboratively by all five researchers. In addition to the formulation of the goal, research questions and other details, the main observable output of this phase is the *Participants guide*. We hand out the Participants guide to each participant. The Participants guide contains instructions on how to install RELATE on their own personal smartphones, links and instructions on how to fill in the participant surveys, where RELATE can be downloaded from, contact e-mail for participants in need of help.

– *Subjects selection (step 2.2)*. After completing the study design we conduct our subjects selection. This step is further detailed in Section 4.1.2.

– *Execution of user study (step 2.3)*. As we are interested in understanding the influence of the introduction of self-adaptation and personalization techniques in our e-Health mobile app, we split the set of participants into two groups. One group uses a baseline version of our RELATE app, whilst the other group uses a version containing the aforementioned techniques. We ask both groups to use their app for *four consecutive weeks*. During this user study, each participant completes three different types of surveys, namely: (i) an initial one-time survey for the demographics, (ii) a daily survey reporting their activities and their perception with respect to their app during the whole four-weeks period, and (iii) a final one-time survey about the overall usability and perception of the two versions of the RELATE app. The details about the structure and contents of the surveys are reported in Sections 4.1.3 and 4.1.5, respectively.

– *Data Analysis (step 2.4)*. The data analysis is carried out once the user study is complete. This phase entails (i) cleaning and organization of all the raw data produced in the previous step and (ii) its qualitative analysis in order to properly answer the research questions. The analysis of the data is further explained in Section 4.1.4.

We organize the **measurement-based experiment** into three main phases: the design of the experiment, its execution, and the data

[2] https://firebase.google.com/docs/cloud-messaging.

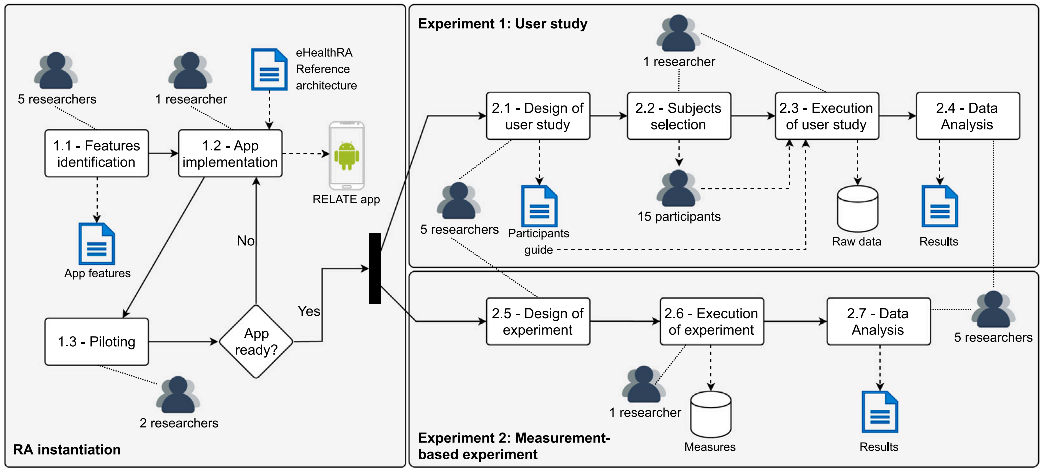


Fig. 3. Study design.

analysis. Below we describe how the three phases fit together and their details are provided in Section 4.2.

– *Design of Experiment (step 2.5)*. The goal of this step is to precisely define the details of the measurement-based experiment, such as its goal, research questions, dependent and independent variables, hypotheses, statistical tests, etc. The experiment is designed as a one-factor-two-treatments experiment, where the main factor is the presence of personalization and self-adaptation techniques. The dependent variables are: the energy consumption, the CPU usage, and the memory consumption of the RELATE app. Similarly to step 2.1, this step is carried out collaboratively by all five researchers.

– *Execution of Experiment (step 2.6)*. In this phase we execute the experiment according to its design. All runs of the experiment are orchestrated automatically and are carried out in a controlled setting. This allows us to isolate the potential effect of the treatments of the main factor of the experiment on the values of the dependent variables, while having minimal bias from external confounding factors. Further details of the experiment execution are reported in Section 4.2.4.

– *Data Analysis (step 2.7)*. In this phase we firstly explore the collected measures by graphically visualizing them and by performing descriptive analyses. Then, we proceed to check for normality and test for statistical significance, so to answer the statistical hypotheses of the experiment. The detailed explanation of the data analysis is given in Section 4.2.3.

A complete **replication package** is publicly available[3] for allowing independent replication and verification of both the experiments presented above.

4.1. Design and execution of Experiment 1 (user study)

4.1.1. Goal and research questions

We formulate the goal of this experiment by using the goal template provided by Basili et al. [21].

> *Analyze* personalization and self-adaptation techniques *for the purpose of* assessing their impact *with respect to* end users' perception *from the viewpoint of* users, developers, and researchers *in the context of* our Android implementation of RA.

The *goal* of this experiment is to study our RA from the end users' perception. In order to gain a better understanding of the combination of AI and self-adaptation, we identified the types of usability concerns that users have whilst using a system complying to our RA, as opposed to an identical, non dynamically tailored, system. As described in Section 3, RELATE is implemented by following our RA and so, in our experiments, is our RA compliant system. As our comparison, non dynamically tailored, system we use the implemented BaseApp (described in Section 3). Specifically, for the scope of these experiments we define dynamic tailoring to be the utilization of the Environment Driven Adaptation Manager, the Smart Objects Manager and the Internet Connectivity Manager. As the two systems are identical, a part from one factor, using them in our experiments allows us to make a fair comparison of systems, that isolates our one investigated factor: dynamic tailoring.

In the following we present and discuss the research questions we translated from the above mentioned overall goal.

> ***RQ*1.1** – What is the impact of personalization and self-adaptation techniques on **end users' perception** of an e-Health mobile app?

The main objective of *RQ*1.1 is to investigate how the inclusion of the aforementioned techniques (personalization and self-adaptation) can influence the perception that a user has on such an app as compared to their perception of apps who are not personalized and self-adaptive. The knowledge gained by answering this question can be of important use to developers as they design and introduce these techniques in their own applications.

> ***RQ*1.2** – What is the impact of personalization and self-adaptation techniques on the **usability** of an e-Health mobile app?

The main objective of *RQ*1.2 is to investigate whether the use of personalization and self-adaptation techniques can influence the usability of an app, as compared to one that does not use such techniques. By studying how users perceive RELATE, we can better understand the usability concerns specifically related to dynamically tailored systems. These findings will allow researchers and developers in this field to have increased awareness on what usability concerns a user of a dynamically tailored system has.

[3] https://github.com/S2-group/self-adaptive-ehealth-apps-replication-package.

Table 1
Table showing the initial subject selection, number of participants whom dropped-out and final group numbers for each user study.

	Participants enrolled	Participant drop out	End size of Group R	End size of Group B
First user study	20	11	5	4
Second user study	9	3	4	2
Total participants	29		9	6

4.1.2. Subjects selection

As shown in Table 1 we recruited 20 participants in the first user study and 9 in the second one. The participants recruited were all either members of staff or students of the university. The advertisement of the user study was posted within the university Canvas groups. No compensation was offered to the participants. The only requirements was that the participant used an Android phone, as our apps could only work on that OS. Participants were split into two groups: Group R used RELATE, group B utilized the BaseApp. During the course of the first user study 11 participants dropped out, whilst 3 participants left in the second user study. The first user study was conducted with weekly reminders to complete the given daily survey. For the second study, as a way to try and diminish participant drop out, the reminders were sent out daily. Another factor that might have played an important role in participant drop out was the ongoing lock-down imposed by the government due to the Covid-19 pandemic. During the first user study, the lockdown was a lot stricter and was, in some cases, the main factor in participants dropping out. This was discovered as we contacted the inactive participants after the trail to inquire on the reasons why they did not complete the study. The Covid-19 restrictions were partially lifted during the run of the second user study. We believe that the combination of less restrictive Covid-19 related policies and the daily reminders, aided to diminish the participants' drop out numbers. In both user studies the majority of the participants identified as male, with six out of nine in the first study and four out of the six in the second study declaring their gender as male. All of the participants were primarily students, a part from one participant in the first user study being a researcher at the university. Overall, seven participants used Samsung model smartphones, three participants used Xiaomi brand smartphones, two used OnePlus brand smartphones and the rest used other brands. Nine of the participants used Android 10, three participants used Android 9 and the rest used either 7, 8 or 11. Lastly, as mentioned in Section 7, given the limited number of participants, the obtained results should be interpreted as an insight into the explored topics not as final generalizations.

4.1.3. Design of the surveys

The initial survey was given to the participants in order to collect general information about them and is formed as shown in Fig. 4a.

The daily survey is presented on Fig. 4b. The goal of the daily survey is to understand the level of engagement of the participants and what their opinion on their daily suggested activity was.

The final questionnaire focuses on usability concerns (Fig. 4c). This questionnaire uses the System Usability Scale (SUS) [22] together with tailored made questions for our particular experiment. All of the questions/state-ments in the daily and final survey, apart from Q13, Q14, Q16 and Q17, are evaluated on a likert scale ranging from 1–5 (1 being strongly disagree and 5 being strongly agree).

4.1.4. Data analysis

For both studies we focused our analysis on the data collected from the final survey, as that most directly addresses RQ1.1 and RQ1.2. For each of the statements counted and classified each of the categorical responses given on the likert scale. We then presented this analysis in the form of tables. Thereafter, we analyzed if a difference was recorded between the users of the BaseApp and those of RELATE.

4.1.5. Experiment execution

Following the initial stage of recruitment, we had the interested participants fill in the initial survey. After, we randomly divided the participants into two groups. Group R used RELATE and Group B used the BaseApp. The participants were then sent an e-mail with attached the .apk file for their system, as well as instructions on how to install it on their own Android devices. Once all participants informed us of the successful installation of the app, we sent them access to the daily survey and sent them their first weekly activity schedule. During the course of the study, we sent e-mail remainders to the participants to fill in their daily surveys. After one month the study was ended. At the end of the study the participants completed the final survey. This designed experiment was executed twice. Once over the month of December 2020 and the second from mid January to mid February 2021. Both executions of the experiment lasted 4 weeks and in both cases the participants were selected via convenience sampling. Due to the Covid-19 pandemic, both studies were conducted with no physical interaction between us and the participants. All correspondence was done via e-mail.

4.2. Design and execution of Experiment 2 (measurement-based experiment)

4.2.1. Goal and research questions

Similarly to the previous experiment, we formulate the goal of this experiment by using the goal template provided by Basili et al. [21].

> *Analyze* personalization and self-adaptation techniques *for the purpose of* assessing their impact *with respect to* resource consumption at runtime *from the viewpoint of* users, developers, and researchers *in the context of* our Android implementation of RA.

In the following we present and discuss the research questions we translated from the above mentioned overall goal.

> ***RQ*2.1** – What is the impact of personalization and self-adaptation techniques on the **performance** of an e-Health mobile app?

The main objective of *RQ*2.1 is to investigate how the use of personalization and self-adaptation could impact the performance of a e-Health mobile app as opposed to one that does not include such techniques. For the purpose of our experiment, we measure performance impact by measuring the CPU usage and memory consumed by the mobile device whilst operating one of the tested systems (either RELATE or the BaseApp). This knowledge can help developers and users of personalized and self-adaptive e-Health mobile apps as performance problems are easily noticed by a user and impact to their experience. These performance problems can be perceived by the user as app sluggishness and non-responsiveness which can lead to user abandonment as they uninstall the app due to frustration and dissatisfaction.

> ***RQ*2.2** – What is the impact of personalization and self-adaptation techniques on the **energy consumption** of an e-Health mobile app?

Evaluation Goal	Question ID	Question text	RQ
Demographic information	Q1	e-mail address	N/A
	Q2	Name	N/A
	Q3	Surname	N/A
	Q4	Age	N/A
	Q5	Mother-tongue	N/A
	Q6	What job do you do?	N/A
	Q7	How many hours a week do you work-out?	N/A
	Q8	Gender	N/A
	Q9	Android software Version	N/A
	Q10	Phone model/brand	N/A

(a) Initial survey given to the participants

Evaluation Goal	Question ID	Question text	RQ
Daily activity	Q11	e-mail address	N/A
	Q12	I am happy with the daily activity suggested to me today	1.1
	Q13	I performed the daily activity suggested to me today	1.1
	Q14	Provide here if you have performed any other physical activity today	N/A

(b) The daily survey given to the participants

Evaluation Goal	Question ID	Question text	RQ
Usability	S1	Whilst using it, the app changed to better fit my needs and preferences	1.1
	S2	The changes that the app performed influenced my perception of it for the better	1.1
	S3-12	As defined in the System Usability Scale	1.2
	Q15	I am happy with the daily activity suggested to me today	1.1
	Q16	I performed the daily activity suggested to me today	1.1
	Q17	Do you have any further comments or suggestions?	N/A

(c) The end survey given to the participants

Fig. 4. Tables listing all of the surveys used in our user study and each question's relationship to the specified research questions.

The main objective of $RQ2.2$ is to investigate if the use of personalization and self-adaptation can have a significant impact on the energy consumption that a e-Health mobile app draws as compared to an identical e-Health app that does not use these techniques. Answering this research question gives important insight to the developer and the user of such apps. The amount of energy consumed by a single app can have great impact on the users' experience, as it could potentially hinder the users' ability of utilizing their mobile device all together. If the introduction of these techniques would lead to a high enough energy consumption, it could potentially discourage users from choosing e-Health mobile apps containing personalization and self-adaptation, something that would be undesirable to a developer.

4.2.2. *Variables and hypotheses*

This section explains both the independent and dependent variables present in our experiment.

The independent variables in this experiment are two: the type of smartphone used and the type of system installed on it. The type of smartphone used has two treatments: low-end and middle-end. For our low-end device we used a LG Nexus 5X and for the middle-end device we used a Samsung Galaxy J7 Duo (further details on the two smartphones are reported in Section 4.1.5). The type of system installed on the smartphone has also two treatments: the system with no dynamic tailoring (the BaseApp) and the system with dynamic tailoring (RELATE). For each execution of one of these systems we measure the below reported dependent variables.

The dependent variables in this experiment are the energy consumed (reported in Joules), the cpu usage (reported as the percentage amount used over the total amount available) and the memory consumed (reported in kilobytes) by either the BaseApp or RELATE.

For each of the above listed dependent variables we formulate the following hypotheses:

- **H1:** We define CPU_B to be the measured CPU usage of the BaseApp and CPU_R to be the measured CPU usage of RELATE. The null and the alternative hypotheses are formulated as follows:
$H1_0 : CPU_B = CPU_R$
$H1_1 : CPU_B \neq CPU_R$
- **H2:** We define MEM_B to be the measured memory consumption of the BaseApp and MEM_R to be the measured consumption of RELATE. The null and alternative hypotheses are formulated as follows:
$H2_0 : MEM_B = MEM_R$
$H2_1 : MEM_B \neq MEM_R$
- **H3:** We define EC_B to be the measured energy consumption of the BaseApp and EC_R to be the measured energy consumption of RELATE. The null and alternative hypotheses are formulated as follows:
$H3_0 : EC_B = EC_R$
$H3_1 : EC_B \neq EC_R$

H1 and **H2** investigate the dependent variables for answering RQ2.1, while **H3** aims at answering RQ2.2. All three hypotheses are separately assessed for each of the two smartphones used.

4.2.3. *Data analysis*

In this experiment we are going to answer each of our research questions in four phases: exploration, normality checks, hypotheses testing, effect size estimation.

Exploration. In this first phase we get an indication of the data collected via the use of descriptive statistics (*i.e.,* mean, median and standard deviation) and boxplots.

Normality checks. We measure the distribution of each data type collected to understand whether we can apply parametric or non-parametric statistical tests. We check whether the data is normally distributed by first visually analyzing it with a Q–Q plot and the applying a Shapiro–Wilks statistical test [23] with an $\alpha = 0.05$. As we report in Section 5.2, the collected data is not normally distributed.

Hypotheses testing. Given the non normal distribution of the data collected we test our hypotheses by the use of the Mann Whitney U test

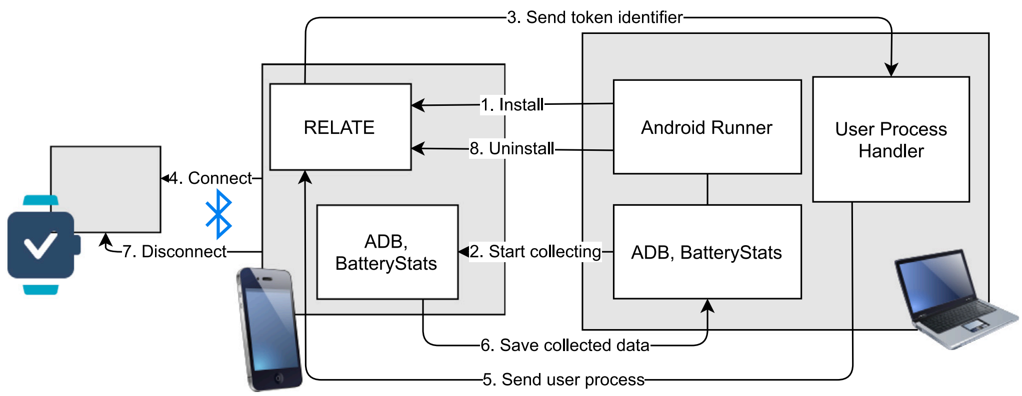


Fig. 5. Execution of one repetition of the experiment.

(with an $\alpha = 0.05$). The Mann Whitney U (also known as the Wilcoxon rank-sum test) is a non-parametric statistical test used to check whether the population of two distributions are statistically equal [24].

Effect size estimation. To statistically test the effect size of the difference found between samples we use the Cliff's Delta statistical test [25]. Cliff's Delta is a non-parametric statistical tool used to calculate the effect size without making assumptions on the distributions compared.

4.2.4. *Experiment execution*

In this subsection we explain how we conducted our experiments to measure CPU, memory and energy consumption. As shown in Fig. 5, for each repetition of our experiments we used: a laptop, one of the two chosen smartphones, a smartwatch, and an internet connection.

– *The Laptop*. It is running Ubuntu 16.04LTS and had the following hardware specifications: RAM 16 GB, CPU i7-6700HQ @ 2.60 GHz * 8, Intel HD Graphics 530. In order to automate our repetitions we installed Android Runner (AR) on the laptop [20]. AR is a framework that allows users to automatically execute measurement-based experiments on both native and web apps running on Android devices.

– *The Smartphones*. The Android devices used are two: a LG Nexus 5X smartphone and a Samsung Galaxy J7 Duo. The LG smartphone has a 1.8 GHz hexacore ARM Cortex A53 & Cortex A57 cpu with 2 GB of RAM running Android 6.0.1. This model is chosen to represent the possible performance impact that these systems can have on an older android smartphone. The Samsung smartphone has a 1.6 GHz octacore ARM Cortex A73 & Cortex A53 cpu with 4 GB of RAM running Android 8.0.0. This smartphone is chosen to represent a mid level android smartphone.

Each repetition starts with installing either the BaseApp or RELATE on the smartphone (step 1). Once installed, AR starts measuring the system's consumption of CPU, memory and energy (step 2). For measuring the CPU and memory consumption AR uses Android Debug Bridge[4] (adb). For measuring the energy consumption it instead uses the Android Batterystats profiler [26]. AR then follows a series of screen taps and gestures that are engineered to be indicative of a worst case scenario. Within the profiling session, the scenario goes through the completion of the initial screen, giving of the necessary runtime permissions, the re-connection to the paired smartwatch (step 4), the receiving of the weekly user activities (step 5). When the scenario is terminated, AR stops profiling the Android device (step 6). After profiling, AR makes the necessary steps to set the device back to how it was before the installation of the system (step 7 and 8). Lastly, AR waits 2 min before running another repetition of the experiment. This wait is introduced to allow the device to 'cool-off' and go back to an idle state; this break minimizes inconsistencies between repetitions. We ran 50 repetitions for each combination of system and smartphone, leading to a total of 200 repetitions.

[4] https://developer.android.com/studio/command-line/adb.

5. Results

In this section we report on the results for both Experiments 1 and 2.

5.1. *Results of Experiment 1 (user study)*

We will now discuss the results of our user studies, organized by research question and following the method and tools described in Section 4.1.4. The results of the final survey for the first user study are shown in Fig. 6a. The participants answering the final survey could rate each statement from agreement (by rating it a 5) to disagreement (by rating it a 1). It is important to note that the scoring can mean something different per statement. For some statements disagreement is desired and for other statements we are looking for user agreement. Participants in Group R used RELATE and Group B participants used the BaseApp. Fig. 6b illustrates the final survey results for the second user study.

5.1.1. *Investigating end users' perception (RQ1.1)*

The statements related to RQ1.1 are S1 and S2 shown in Fig. 6, and Q12, Q15, Q13, and Q16 defined in Figs. 4b and 4c.

– *Whilst using it, the app changed to better fit my needs and preferences (S1)*. In both user studies participants in Group R tended to agree more with this statement. Participants in Group B did instead agree less. This means that generally participants in Group B did not find the app to change for the better, implying that they either did not think the app changed or that it changed for the worse.

– *The changes that the app performed influenced my perception of it for the better (S2)*. In the first user study participants in Group R rated this statement with either disagreement or neutrality. Whilst members of Group B showed more agreement to the statement. In the second user study members of both groups showed their opinion to be either neutral or in agreement. These findings imply that in the first user study participants using RELATE did not believe that the changes performed by the app influenced their opinion of it for the better. This could mean that either the changes they perceived did not impact their opinion of RELATE or shifted their opinion for the worse. On the contrary, members of Group B as well as all participants in the second user study seemed to have received the perceived changes positively. This indicates that, especially for users of RELATE in the second study, the changes offered and perceived by the users are considered a positive aspect of the app.

– *I am happy with the daily activity suggested to me today (Q12 and Q15)*. We have grouped Q12 and Q15 as they are the same question but posed in two different surveys (*i.e.,* the daily and the final surveys). For the first user study, we gathered the following average (median) and

Final Survey, First User Study (9 participants in total)		BaseApp (B)					RELATE (R)				
		1	2	3	4	5	1	2	3	4	5
S1	Whilst using it, the app changed to better fit my needs and preferences	1	2	2	0	0	1	0	2	1	0
S2	The changes that the app performed influenced my perception of it for the better	1	0	1	2	0	2	1	2	0	0
S3	I think that I would like to use this app frequently.	1	0	2	1	0	3	2	0	0	0
S4	I found the app unnecessarily complex.	4	0	0	0	0	4	1	0	0	0
S5	I found the app easy to use.	0	0	0	3	1	0	0	0	2	3
S6	I think that I would need the support of a technical person to be able to use this app.	4	0	0	0	0	5	0	0	0	0
S7	I found the various functions in this system were well integrated.	1	0	2	1	0	1	2	1	0	1
S8	I thought there was too much inconsistency in this app.	4	0	0	0	0	0	2	3	0	0
S9	I would imagine that most people would learn to use this app very quickly.	0	0	0	1	3	0	0	1	1	3
S10	I found the app very cumbersome to use.	4	0	0	0	0	2	0	2	1	0
S11	I felt very confident using the app	0	0	1	1	2	0	0	1	2	2
S12	I needed to learn a lot of things before I could get going with this app.	3	1	0	0	0	4	1	0	0	0

(a) The final survey answers entered by the participants of the first user study (1 - disagreement, 5 - agreement).

Final Survey, Second User Study (6 participants in total)		BaseApp (B)					RELATE (R)				
		1	2	3	4	5	1	2	3	4	5
S1	Whilst using it, the app changed to better fit my needs and preferences	0	0	1	1	0	0	0	1	3	0
S2	The changes that the app performed influenced my perception of it for the better	0	0	1	1	0	0	0	2	2	0
S3	I think that I would like to use this app frequently.	0	0	2	0	0	1	1	0	2	0
S4	I found the app unnecessarily complex.	2	0	0	0	0	3	1	0	0	0
S5	I found the app easy to use.	0	0	0	1	1	0	0	0	2	2
S6	I think that I would need the support of a technical person to be able to use this app.	0	2	0	0	0	4	0	0	0	0
S7	I found the various functions in this system were well integrated.	0	0	1	1	0	0	0	2	2	0
S8	I thought there was too much inconsistency in this app.	0	2	0	0	0	0	2	2	0	0
S9	I would imagine that most people would learn to use this app very quickly.	0	0	0	1	1	0	0	0	2	2
S10	I found the app very cumbersome to use.	1	1	0	0	0	2	1	1	0	0
S11	I felt very confident using the app	0	0	1	1	0	0	1	1	2	0
S12	I needed to learn a lot of things before I could get going with this app.	1	1	0	0	0	4	0	0	0	0

(b) The final survey answers entered by the participants of the second user study (1 - disagreement, 5 - agreement).

Fig. 6. Recorded ratings for the final survey for both user studies.

standard deviation for Group B: 4.290 (5) and 1.062. Whilst, for Group R we gathered the following average (median) and standard deviation of 3.9256 (4) and 1.039. In the second user study, we gathered the following results of average (median) and standard deviation from Group B: 3.706 (4) and 0.47. For Group R we got an average (median) and standard deviation of 3.655 (4) and 0.971. In both user studies the results gathered indicate a minimal difference in happiness with the suggested daily activities, with the participants in Group B seemingly happier.

– *I performed the daily activity suggested to me today (Q13 and Q16)*. For these questions we calculated the percentage of times that participants of a group reported performing their suggested physical activity of the day. In the first user study participants of Group B reported following the suggested daily activity 82.26% of the time. Participants of Group R reported performing their suggested activity 73.4% of the time. In the second study, the participants of Group B reported performing their daily suggested activity 58.82% of the time, whilst participants of Group R agreed with the suggested activity 66.37% of the time. These findings show that in the first user study, participants using RELATE recorded performing their suggested activities less often than those not using this app. The opposite can be seen from the results of the second user study.

5.1.2. Investigating usability (RQ1.2)

In this section we report on the data obtained to answer RQ1.2. The statements related to this research question are S3 to S12 shown in Fig. 6. We have grouped the statements together per overarching topic: ease of use of the app, app cohesiveness, and likely hood of using the app in the future. – *Ease of use of the app*. This topic groups the following statements:

- I found the app unnecessarily complex. (S4)
- I found the app easy to use. (S5)
- I think that I would need the support of a technical person to be able to use this app. (S6)
- I would imagine that most people would learn to use this app very quickly. (S9)
- I felt very confident using the app. (S11)
- I needed to learn a lot of things before I could get going with this app. (S12)

For this group of statements, we find that the participants rated their utilized apps easy to use (S5, S6, S9, S11) and understand (S4, S6, S12). This is true no matter if the participants are from Group R or Group B. Furthermore, there is no significant difference in opinion between the participants of the first user study as compared to those of the second user study. As participants from both groups had no significant difference in scoring their version of the app, we also understand that dynamic tailoring did not make it harder for the user to use and understand the app.

– *App cohesiveness*. This group is comprised of the following statements:

- I found the various functions in this system were well integrated. (S7)
- I thought there was too much inconsistency in this app. (S8)

The findings for S7 in the first user study show that, no matter which group the participants belonged to, they were split across the scale. The opposite is seen for the results of the second user study, here the opinion on the integration of system functionalities was more focused and seen more positively. More cohesive are the results of S8. For both user studies the participants found their apps to not have too much inconsistency. In particular users of RELATE were more neutral towards this statement than the users of the BaseApp, which all rated the statement with disagreement.

– *Likelihood of using the app in the future*. Lastly, in this topic we group the following statements:

- I think that I would like to use this app frequently. (S3)
- I found the app very cumbersome to use. (S10)

The findings for S3 are non homogeneous and inconsistent across the two user studies. In the first one, most participants either disagreed with the statement of were neutral about it. In the second one, half of Group R disagreed with the statement, whilst the other half agreed; the members of Group B all rated the statement neutrally. These findings show a non homogeneous consent in the opinions of the participants. We, instead, find cohesion in the results given for S10, as in both user studies the majority of participants did not find their version of the app to be cumbersome to use. We further elaborate on these results in Section 7.4.

5.2. Results of experiment 2 (Measurement-Based Experiment)

In this section we report on the results obtained for the measurement-based experiment. We will be discussing the results per research question answered, following the procedure reported in Section 4.2.3.

5.2.1. Impact on performance (RQ2.1)

Exploration. The performance data measured for the LG smartphone are shown on Fig. 7. For the CPU measurements, we see no clear difference between the two apps. The mean (median) and standard deviation for the BaseApp are: 14.24% (10.55%) and 7.90%, whilst for RELATE they are: 13.69% (11.00%) and 7.15% respectively.

We can observe a difference between the distribution of the memory usage of the two apps, with RELATE consuming more memory. The mean (median) and standard deviation of the BaseApp are 65266.40 kB (64675.35 kB) and 1140.62 kB, respectively, and the descriptive statistics of RELATE are: 67912.26 kB (67859.79 kB) and 519.67 kB, respectively.

The performance measured for the Samsung smartphone are shown in Fig. 8. As shown in Fig. 8a, we have found no difference in CPU usage by the two systems. The mean (median) and standard deviation of the BaseApp are: 13.90% (11.96%) and 9.34%. Similarly, the descriptive statistics for the CPU consumption of RELATE are: 12.61% (11.00%) and 7.93%. Similar to the LG smartphone, in this case RELATE tends to use more memory than the BaseApp (seen in Fig. 8b). The mean (median) and standard deviation for the BaseApp are: 47963.58 kB (44531.24 kB) and 6363.78 kB, respectively. Differently, RELATE reported a mean (median) and standard deviation of: 55492.90 kB (54478.22 kB) and 2898.77 kB, respectively.

Check for normality. Fig. 7 shows the Q–Q plot against the normal distribution for both the CPU and the memory consumption data measured on the LG smartphone. Several measures fall far away from the reference line, indicating that the collected measures are not normally distributed. To further confirm our observation we carried out the Shapiro–Wilks test on all four datasets. For RELATE CPU measurements the test returned a p-value of 3.545e−08 and, for the BaseApp we achieved a p-value of 4.72e−09. For the memory measurements of RELATE we obtained a p-value = 0.03345 and for the BaseApp the p-value is 1.89e−05. Therefore, in all cases, we can reject the null hypothesis stating that these samples come from a normal distribution.

With Fig. 8 we illustrate the Q–Q plots for the performance measurements taken on the Samsung smartphone.

Our Shapiro–Wilks tests confirm the Q–Q plots. With a returned p-value of 6.262e−10 for the CPU measurements of RELATE and a p-value of 4.161e−09 for the BaseApp. For the memory usage RELATE had a p-value = 1.285e−11 and the BaseApp returned a p-value of 4.661e−07. We can therefore reject the null hypothesis stating that the Samsung smartphone CPU usage data comes from a normal distribution.

Hypothesis testing. As stated in Section 4.2.3, we utilize the non-parametric Mann–Whitney U test to determine whether we can reject our stated null hypotheses (formulated in Section 4.2.2). Starting by examining the measurements collected for the LG smartphone; the p-value returned for the comparison between BaseApp and AdaptiveSystem on the CPU consumption is equal to 0.48. As this p-value is above the significance threshold ($\alpha = 0.05$), we cannot reject the null hypothesis $H1_0$. When applying the statistical test to the memory consumption values of the two apps we obtain a returned p-value of 2.54814e−17. As this value is smaller than our chosen α, we can reject our null hypothesis $H2_0$. The above findings are similar for the Samsung smartphone, where the returned p-values for the comparison of the two apps on CPU consumption and memory usage are 0.36 and 9.67143e−13, respectively. This means that we cannot reject the null hypothesis $H1_0$, but we can reject the null hypothesis $H2_0$.

Effect size estimation. As a follow up to the use of the Mann–Whitney U test, we determine the effect size of the differences found. As stated in Section 4.2.3, we use Cliff's Delta to do so. A **large** effect size is found when investigating the difference in memory consumption between the BaseApp and AdaptiveSystem for both the Lg smartphone (0.98) and the Samsung smartphone (0.83).

5.2.2. Impact on energy consumption (RQ2.2)

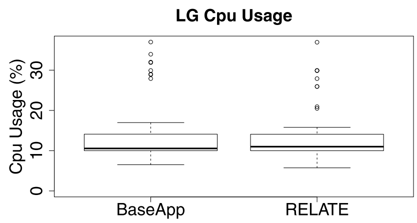
Exploration. Fig. 9b shows the distribution of the energy consumption of the two apps running on the LG smartphone. We see no apparent difference in the energy consumption between the two apps. Indeed the mean (median) and standard deviation for the BaseApp are: 139.54 J (133.53 J) and 15.34 J. For RELATE the mean (median) and standard deviation are: 139.32 J (132.95 J) and 18.78 J.

For the Samsung smartphone we observe a slight difference in energy consumption between the two systems (shown in Fig. 9a). We can observe that RELATE consumes less energy than the BaseApp; we will be further discussing this finding in Section 6. The mean (median) and standard deviation of the energy consumption for the baseline app are: 137.51 J (134.83 J) and 9.99 J; whilst the descriptive statistic for RELATE are: 133.96 J (131.85 J) and 8.28 J respectively.

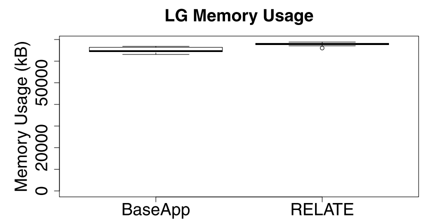
Check for normality. Figs. 9d and 9f show the Q–Q plots against the normal distribution for the energy consumption measured on the LG smartphone. Both plots show that the data collected is not normally distributed.

To further corroborate our finding, the Shapiro–Wilks test done on RELATE's dataset returns a p-value of 6.115e−12 and the BaseApp case gives a p-value of 5.904e−09. Therefore we can reject the null hypothesis of the data belonging to a normal distribution.

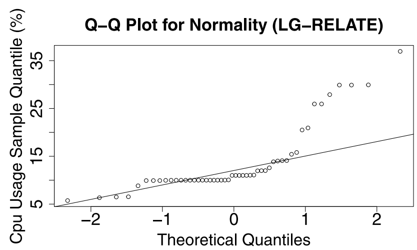
Figs. 9c and 9e illustrate the Q–Q plots against the normal distribution for the energy consumption measured on the Samsung smartphone. As the plots indicates, the data is not normally distributed. This is confirmed by the Shapiro–Wilks test, as RELATE returned a p-value of 1.258e−05 and the BaseApp returned a p-value equal to 3.329e−08.



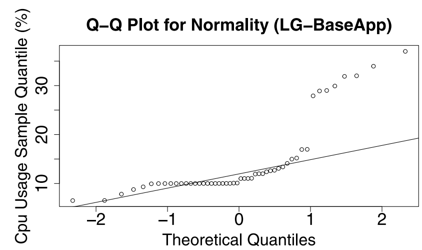
(a) CPU usage for the LG smartphone



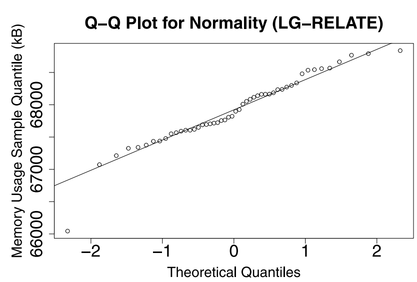
(b) Memory usage for the LG smartphone



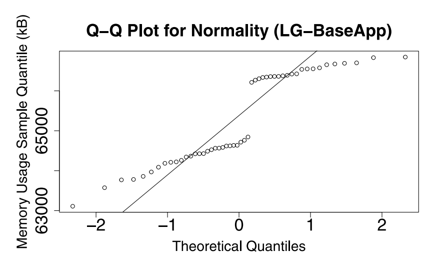
(c) Q-Q plot for the CPU usage on the LG smartphone for RELATE



(d) Q-Q plot for the CPU usage on the LG smartphone for the BaseApp



(e) Q-Q plot for the memory usage on the LG smartphone for RELATE



(f) Q-Q plot for the memory usage on the LG smartphone for the BaseApp

Fig. 7. All plots related to performance measurements for the LG smartphone.

We can therefore reject the null hypothesis that the energy consumption measured from the Samsung smartphone comes from a normal distribution.

Hypothesis testing. We start by using the Mann–Whitney U test on the energy consumption data collected on the Lg smartphone. The p-value returned by the test is 0.73, as this value is above our chosen α, we cannot reject the null hypothesis $H3_0$ and therefore we find that the difference in energy consumption between the BaseApp and RELATE on the Lg smartphone is not statistically significant. When running the test on the energy consumption data for the Samsung smartphone we obtain a p-value of 0.009. As the p-value is below our α threshold, we can reject the null hypothesis $H3_0$ and find that the difference in energy consumption between the BaseApp and RELATE is statistically significant.

Effect size estimation. Here we use Cliff's Delta to follow up on the findings gathered in our hypothesis testing. The difference found on the Samsung smartphone can be classified as small (*i.e.,* −0.30).

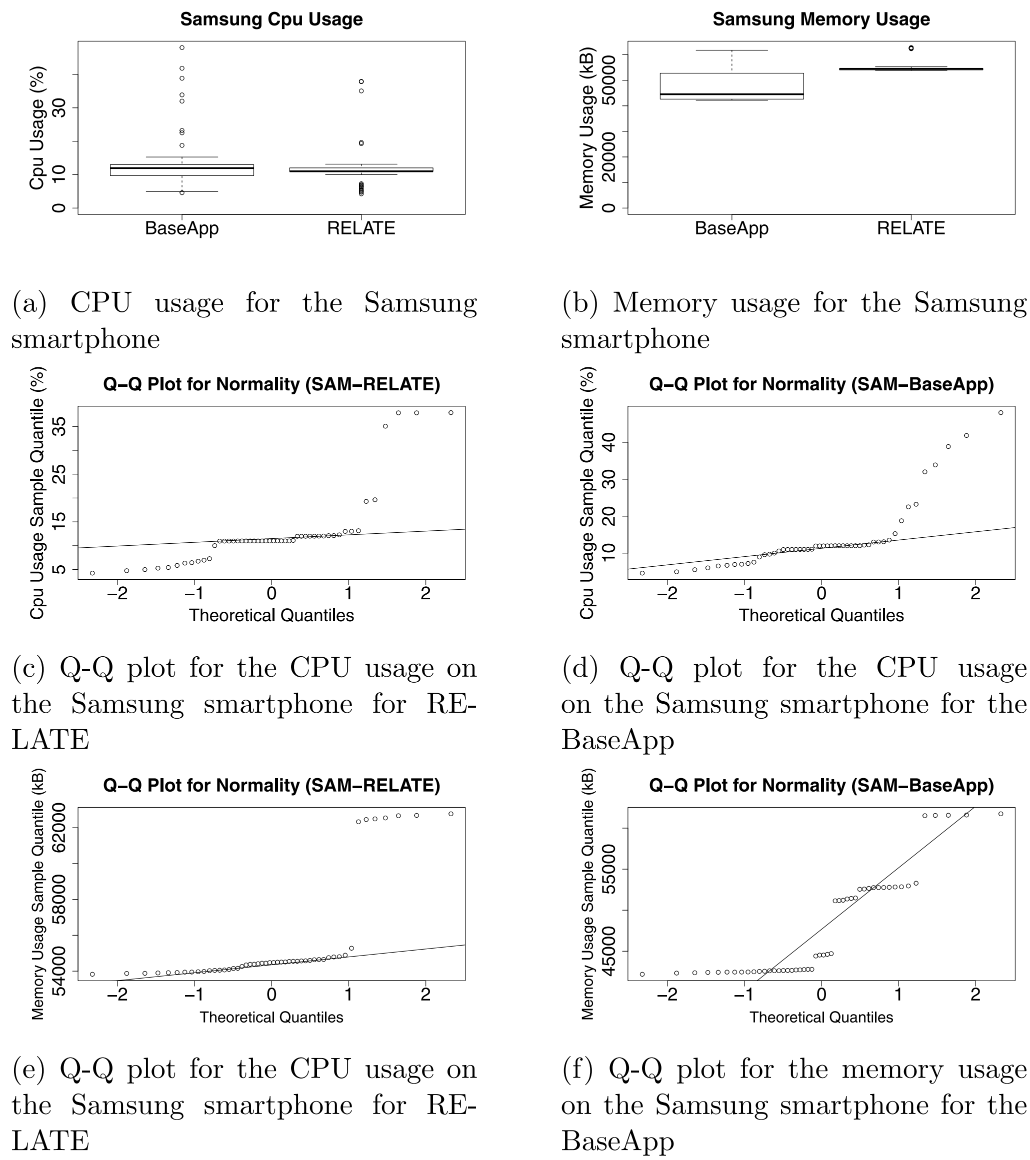
6. Discussion

6.1. Discussion on experiment 1 (User study)

We start by discussing the results on the *end users' perception* (RQ1.1). Participants using RELATE tend to agree more than those using the BaseApp that the app changed to better fit their needs and preference (S1). This result is interesting, as it suggests that users of RELATE (i) noticed the adaptation of the app and (ii) found those changes to be useful.

Most participants rated the statement "the changes that the app performed influenced my perception of it for the better" neutrally or approvingly (S2). Only the users of RELATE in the first user study also stated disagreement with the statement. The disagreement with the statement can either mean that the participants found the changes to modify their perception of RELATE for the worse or that they did not make a difference in their perception of the app. Given the agreement recorded for the previous statement (S1), we find it unlikely for the disagreement on this statement, S2, to have a negative connotation: as this would contradict the positive implications found with S1. Therefore, we can conclude that the changes performed by the app were overall seen as either non impactful to the users' perception of the app or as a positive influence.

Regarding how happy the users were with their daily activities, our results found little difference between users using the BaseApp and those using RELATE in both user studies. The only difference seems to be the fact that participants in Group B appeared to be somewhat happier. This, however, is not reflected in the adherence to performing the suggested activities. Here, the two user studies show opposite results with the first one showing participants in Group B performing their suggested activities more often, whereas the second user study showed Group R more often performing their daily activities. Therefore, these last results seem to be inconclusive. This could be due to the simplicity of the suggested daily activities and the minimal



(a) CPU usage for the Samsung smartphone

(b) Memory usage for the Samsung smartphone

(c) Q-Q plot for the CPU usage on the Samsung smartphone for RELATE

(d) Q-Q plot for the CPU usage on the Samsung smartphone for the BaseApp

(e) Q-Q plot for the CPU usage on the Samsung smartphone for RELATE

(f) Q-Q plot for the memory usage on the Samsung smartphone for the BaseApp

Fig. 8. All plots related to performance measurements for the Samsung smartphone.

dynamic tailoring that is done with them in this current version of the implemented apps. As future work, it would be important to include all of the Back-end components in the RA in order to be able to better personalize the daily suggested activities to the participants using RELATE. This further implementation of dynamic tailoring could lead to a wider observed difference between the two groups of participants as the two apps will be further distinguished from each other.

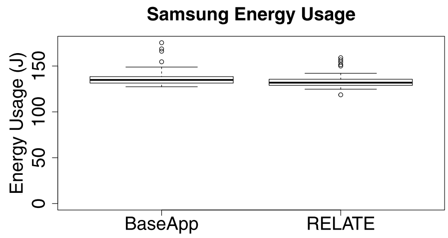
In summary, the results we have obtained indicate that **personalization and self-adaptation techniques have an overall positive impact on the end users' perception of e-Health mobile apps**. Therefore, developers and researchers whom are interested in end users' perception, can successfully adopt these techniques in their own e-Health apps.

Lastly, privacy was a relevant concern during our experiments. We addressed privacy concerns by having all of our participants give us their personal information willingly and understand that it would be saved and used for the purposes of this work. To respect privacy regulations, the data presented to the public, via the replication package, has been anonymized.

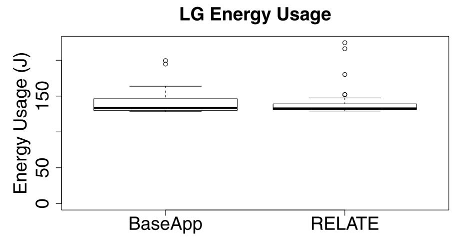
We will discuss now the results related to our investigation on the *usability* of our e-Health mobile app (RQ1.2). Across all of the statements analyzed, we see a pattern of agreement between all of the participants, no matter the group they were assigned to. This is interesting, as it points to dynamic tailoring not being a determining factor to how the participants responded to the survey. We observed only with statements S3 (*i.e.,* "I think that I would like to use this app frequently") and S7 (*i.e.,* "I found the various functions in this system were well integrated") that the participants did not show a clear trend or consensus, and were instead more distributed along the Likert scale. Given these results, we can conclude that **there seems to be no apparent impact caused by personalization and self-adaptation techniques on usability of e-Health mobile apps**. As discussed previously, a future implementation of RELATE containing the complete Back-end components from our RA might help surface differences that were not recorded in this study, as this new version of RELATE would include the full range of dynamic tailoring advocated by our RA and would therefore increase the difference between our two tested systems (RELATE and our BaseApp).

6.2. Discussion on experiment 2 (Measurement-Based Experiment)

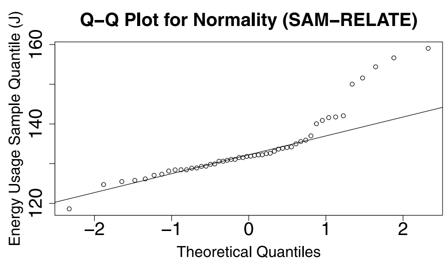
We start the discussion by elaborating on our results for RQ2.1, namely: "What is the impact of personalization and self-adaptation techniques on the *performance* of e-Health mobile apps?". For both



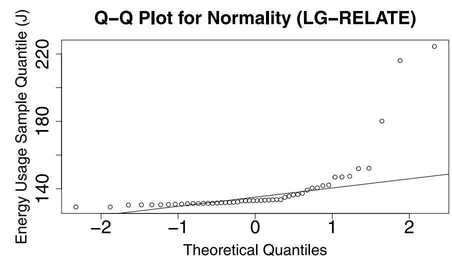
(a) Energy usage for the Samsung smartphone



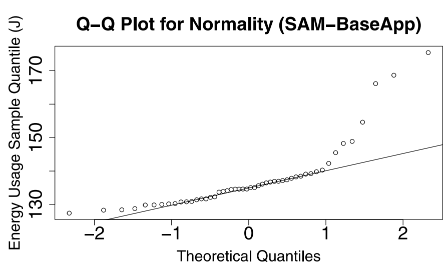
(b) Energy usage for the LG smartphone



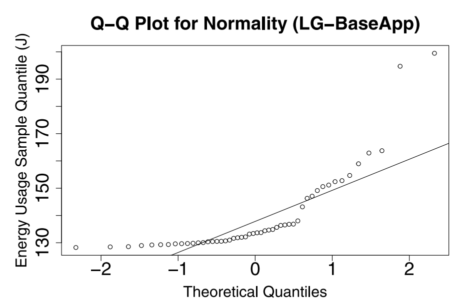
(c) Q-Q plot for the energy usage on the Samsung smartphone for RELATE



(d) Q-Q plot for the energy usage on the LG smartphone for RELATE



(e) Q-Q plot for the energy usage on the Samsung smartphone for the BaseApp



(f) Q-Q plot for the energy usage on the LG smartphone for the BaseApp

Fig. 9. All plots for the energy usage measured.

devices, RELATE tends to use more memory than the BaseApp. This is understandable, as RELATE contains the adaptive components that the BaseApp does not (*i.e.,* Environment Driven Adaptation Manger, Smart Objects Manager, and Internet Connectivity Manager). These components require the utilization of the smartphone's memory in order to carry out their business logic. As an example, the Environment Driven Adaptation Manager needs to assess what current day of the week it is, what the weather forecast for that day is and if it needs to change the currently recommended daily activity. Having said so, the difference of the amounts of used memory is negligible when put into the context of the total amount of memory that these devices have. The difference between the averages for the LG smartphone is 2645.9 kB (over a total of 2 GB available) and for the Samsung smartphone is 7529.3 kB (over a total of 4 GB available). This difference, whilst shown to be statistically significant, has no practical implication over the user experience of the apps. We can therefore conclude that the 'price' paid in terms of memory consumption for the benefits of adding dynamic tailoring is worthwhile.

Our results also show a difference in the CPU usage levels between the two examined apps on the Samsung smartphone. However, RELATE seems to be the one consuming the least amount of CPU. If we take the difference between RELATE CPU usage average and the one of the BaseApp we get a difference of −1.288 (the CPU measurement is quantified as a percentage of the total amount of CPU). Whilst our analysis has shown this difference to be statistically significant, we argue that such a small difference in CPU usage would have no impact on the user experience. In conclusion, **whilst our results have found some statistically significant differences in app performance, these differences are too small to realistically impact the user experience of an Android app**.

Regarding RQ2.2, namely: "What is the impact of personalization and self-adaptation techniques on the *energy consumption* of e-Health mobile apps?". Only for the Samsung smartphone we found a statistically significant difference in energy consumption between the two apps. This difference is, however, not expected as it shows RELATE to be consuming *less* energy than the BaseApp. Upon closer inspection, we notice that the difference in average energy consumption between the BaseApp and RELATE is equal to 3.6 J. Just like with the differences found for the memory and the CPU, this discovered difference is so small that it will not impact the user experience in a practical sense. In conclusion, **our experiments provide evidence that the impact of personalization and self-adaptation techniques on the energy consumption of e-Health mobile apps is negligible**.

Overall, our findings for RQ2.1 and RQ2.2 show that using personalization and self-adaptation techniques in e-Health mobile apps has no adverse effect on both performance an energy consumption. This should encourage app developers and researchers working in this field to adopt these techniques in their own e-Health apps as they provide a great range of extra functionalities with little to no impact on the resources of the user's smartphone.

7. Threats to validity

7.1. External validity

– *Experiment 1 (User study)*. There is a threat to generalizability as the sample of participants for our experiment was limited. Because of this, the presented results are not meant to be final but rather as an exploration of these topics. Further work with a larger sample of participants would be needed to draw more conclusive results.

– *Experiment 2 (Measurement-Base Experiment)*. To minimize the threat to external validity we ran our experiment on two different types of smartphone. The smartphones chosen are intended to be a representation of a low-end and a middle-end device. This diversification of devices should better capture the real world scenario. Having said so, the use of a newer smartphone could possibly lead to different results and conclusions. We therefore encourage further experiments to further minimize this threat to validity.

7.2. Internal validity

– *Experiment 1 (User study)*. To mitigate the threat to internal validity we implemented the two applications to be as close as possible, leaving dynamic tailoring as the sole difference. Furthermore, the participants for both groups R and B were recruited in the same manner and are all of a comparable demographic (therefore mitigating possible selection bias).

– *Experiment 2 (Measurement-Base Experiment)*. There are a number of factors that can influence the measurements we have collected in our experiments *i.e.,* brightness of the screen, distance to the internet router, distance to the Bluetooth smartwatch and background processes. We designed our experiments so to minimize as much as possible these factors. We maintained the brightness of the screen, the distance to the internet router and the distance to the Bluetooth smartwatch fixed across all repetitions. To mitigate the impact of uncontrollable background processes we performed 50 repetitions for each experiment case, mitigating the bias that one spike in background processes can have over our overall readings. Lastly, maturation can influence the data collected in the experiments. In our case maturation is the changes that occur in the smartphone as the experiment is running (*e.g.,* memory usage, CPU heat generation and impact on its performance). In order to mitigate it, we imposed a waiting time of 2 min between each repetition. We also cleared any data that was gathered during the course of a repetition, to maintain the status of the smartphone identical across experiments.

7.3. Construct validity

– *Experiment 1 (User study)*. To minimize the threat to construct validity, we defined all of the details regarding our experiment design a priori (*e.g.,* research questions, data analysis methodology, variables).

– *Experiment 2 (Measurement-Base Experiment)*. Here we also defined everything regarding our experiment design and methodology a priori.

7.4. Conclusion validity

– *Experiment 1 (User study)*. To mitigate the threat to conclusion validity, we had all 5 researchers involved in the data analysis of the results obtained for this experiment. This mitigates an individual bias and interpretation of the results. Furthermore, we offer a complete replication package to the public. Allowing for independent replication of our experiment.

– *Experiment 2 (Measurement-Base Experiment)*. To minimize the threat to conclusion validity we have used statistical analysis to more objectively draw our conclusions on the experiment. Lastly, we offer a complete replication package to the public. Allowing for independent replication of our experiment.

8. Related work

Self-adaptation represents a suitable method to detect and deal with (potentially impactful) unexpected context changes. In the field of *mobile* apps, it is even more challenging due to, *i.e.,* mobile phones resource constraints (*e.g.,* battery level, network traffic). The need for *self-adaptation* is exacerbated in the e-Health domain where adapting to the user-needs and context may be of crucial importance, *i.e.,* to properly and promptly react to monitored patients activities. Ballesteros et al. [5] present a wearable patient-monitoring system for tele-rehabilitation, supporting therapies by providing valuable information for the evaluation, monitoring, and treatment of patients. The system follows a goal-oriented self-adaptation approach based on dynamic software product lines (DSPL) and it uses a set of self-adaptation policies enabling it to dynamically self-configure its internal behavior to the current context of the patient, while maintaining the system efficiency (*e.g.,* optimizing battery consumption). Differently than our RELATE app, the used adaptation policies do not influence the usability or end users' perception, since end users do not directly interact with the system. They only make use of the system's devices and wearable (*e.g.,* knee motion sensor), used to monitor and recognize the activity the user is performing and to collect data to trigger re-configuration.

Mizouni et al. [7] focus on the design and development of self-adaptive applications that sense and react to contextual changes (*e.g.,* environment, device status) to provide a *value-added user experience*. The authors present a framework defining a systematic approach to model dynamic adaptation of mobile apps behavior at runtime by using SPL concepts and offering feature priority based dynamic adaptability. The framework is evaluated through an application supporting doctors on the move to have access to patients' files, report medical conditions, prepare for intervention and communicate with the hospital. A similar application targeting doctors on the move is proposed by Preuveneers et al. [8]. In this study, the authors focus on how to deliver the right patient's information at the right time under variable connectivity and limited resource availability. Probabilistic models and dynamic decision networks are used to improve the user experience and on-device resource utilization. Differently than [7], we have defined our RA by leaving a certain degree of freedom to developers about design decisions and adaptation strategies. Moreover, in contrast to both [7,8], our RELATE app is not intended for healthcare professionals and caregivers, but for end-users. For this reason, usability and users' perception concerns are quite relevant, since they might impact the constant and active commitment of end users.

Lopez et al. [6] make use of non-obtrusive monitoring technology in their context-aware mobile app delivering self-adaptive persuasive messages that stimulate the medication adherence, by exploiting real-time physiological data (*e.g.,* heart rate). In our e-Health app, in contrast, self-adaptation and personalization are used to create the best possible conditions for users to keep active in their activities, by considering their current context and preferences, thus guaranteeing a good level of usability.

Table 2
Comparison with related works.

Approach	Personalization	Usability	Self-adaptation	Energy efficiency
Ballesteros et al. [5]	x	x	✓	✓
Mizouni et al. [7]	x	x	✓	x
Preuveneers et al. [8]	x	x	✓	✓
Lopez et al. [6]	✓	✓	x	x
Raheel [27]	~	✓	✓	x
Sartori et al. [9]	✓	x	x	x
Gamberini et al. [10]	✓	~	x	x
Burley et al. [11]	✓	x	x	x
RELATE	✓	✓	✓	✓

Concerning the user experience provided by the apps' user interface, Raheel [27] proposes a set of adaptive mechanisms able to monitoring the user's behavior w.r.t. a mobile phone (*e.g.,* determining the distance between the user and the screen) and adapting the interface accordingly. The author presents a medical adaptive mobile app aiming to help the elderly remember taking their medicine at specific times. Experiments show the effectiveness of adaptive user interface in improving usability and acceptance of the mobile app. However, our RA brings other instruments in support of usability that go beyond the usability of the user interface (*e.g.,* the goal model, user process adaptation) and, simultaneously, it aims to guarantee that the personalization and self-adaptation techniques we use do not degrade the app usability.

Other works specifically focused on personalization to keep e-Health apps users engaged. Sartori et al. [9] exploit fuzzy logic to cluster users according to quantitative and qualitative variables about their physical and psychological well-being. They applied their approach to design and implement *MoveUp*, a e-Health mobile app aiming at increasing the physical activity level in sedentary people. Experiments show that personalization by means of users profiling succeeds in promoting group physical activity among users characterized by similar behaviors. In this work, similarly to our approach, the authors realize a data-driven personalization by clustering similar users, although they do not consider self-adaptation of their app and do not evaluate its usability. Gamberini et al. [10] propose *PatchAi*, an e-Health app exploiting a conversational agent in the form of a chatbot, to reduce the drop-out rates and increase the patience adherence to treatments. The app supports users in inserting several data about their disease, symptoms, drugs assumption, etc. to which doctors may access and evaluate realtime clinical trial patient-related information. Lastly, for its implementation, the authors followed the basic principles of usability and accessibility, although they did not evaluate usability with users. Burley et al. [11] also experienced a high drop-out rate to their *MindTrails* online platform, which aims to reframe the thinking patterns of highly anxious individuals when responding to ambiguous situations that they might interpret as stressful. As a solution they combined personalization and implementation intentions. The former is done through an interactive and informative interface driving users in selecting the domain in which they want to improve their thinking. The latter, instead, is performed by psychologists expert in behavioral interventions that provide intentions scenarios based on the users data to ensure complete their treatments. In both [10,11] the personalization requires the users participation, thus it is not transparent neither exploits users smart objects to collect users e-Health data. Moreover, self-adaptation is not considered at all.

In Table 2 we outline the comparison with the discussed related work. In particular, ✓denotes that the corresponding feature is covered, x means that it is not considered at all, and ∼ denotes that the feature is only partially covered in comparison with our work. The above reviewed studies share with our work the exploitation of self-adaptation techniques or the use of personalization techniques to keep users engaged, in e-Health mobile apps. None of them exploits both approaches to get the most from their combination, as shown in Table 2. Further, works providing personalization solutions, rarely evaluate usability of their e-Health apps, which instead might negatively impact the personalization results. Moreover, in the context of *mobile* apps, adaptation engines must satisfy the energy efficiency requirement. According to Cañete et al. [28], energy consumption also depends on the execution context (environment, devices status) and how the user interacts with the application. Indeed, despite the hardware consumes the energy, the software (*e.g.,* adaptation mechanisms) is responsible for managing hardware resources and its functionality, thus affecting the energy consumption. This demands for energy-efficient adaptation. Although some of the reviewed work (*e.g.,* [5,8]) aim to maintain the system efficiency, differently from them, our study investigates the impact of the used personalization and self-adaptation techniques on the performance and energy consumption of an e-Health app. Results of our empirical evaluation clearly show that applications built on top of our RA, exploiting several MAPE loops and dynamic, personalized user processes, guarantee energy-efficient adaptation.

9. Conclusions and future work

In this work we build upon the RA and test an implemented prototype app that complies to it. We call this prototype RELATE, standing for peRsonalized sELf-AdapTive E-health. We designed and executed two experiments: a user study, to test user concerns regarding usability and app perception, and a measurement-based experiment to test concerns related to performance and energy consumption. Both experiments focused on studying the impact of self-adaptation and personalization on their respective independent variables. To be able to isolate these variables, RELATE was tested again an identical app, lacking dynamic tailoring, which we named BaseApp. In our user study, our results show that end users' usability and perception is not harmed by the introduction of dynamic tailoring and is instead made better for the case of usability whilst for user perception we could not find any significant difference. In our measurement-based experiment, we concluded that for both performance and energy consumption the differences measured were never at such a scale to cause real world usage consequences. As reported in Section 7, there are limitations with our studies. The most impactful to our results are the relatively low number of participants and the limited variety of smartphones used. Whilst we could not fully overcome these limitations in this iteration of the studies, we intend on addressing them in future work.

As future work we are also planning to expand RELATE to implement and use the entirety of the RA components. This would allow for a deeper level of personalization and possible updated results on our experiments. We would also like to conduct a larger user study with the new version of RELATE. Lastly, we are planning to investigate the feasibility of further generalizing the RA and RELATE to other domains outside of e-Health, such as apps for the social good and social networks.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is partially supported by the PON (Programma Operativo Nazionale) AIM1880573 Linea 1, Italian National Project at GSSI.

References

[1] Inc. Global Industry Analysts, mhealth (mobile health) services - market analysis, trends, and forecasts, 2019, URL https://tinyurl.com/rbvdtc3.

[2] M. Paschou, E. Sakkopoulos, E. Sourla, A. Tsakalidis, Health internet of things: Metrics and methods for efficient data transfer, Simul. Model. Pract. Theory 34 (2013) 186–199, http://dx.doi.org/10.1016/j.simpat.2012.08.002.

[3] M.W.L. Moreira, J.J.P.C. Rodrigues, V. Korotaev, J. Al-Muhtadi, N. Kumar, A comprehensive review on smart decision support systems for health care, IEEE Syst. J. 13 (3) (2019) 3536–3545, http://dx.doi.org/10.1109/JSYST.2018.2890121.

[4] B. Fling, Mobile Design and Development: Practical Concepts and Techniques for Creating Mobile Sites and Web Apps, O'Reilly Media, Inc., 2009.

[5] J. Ballesteros, I. Ayala, J.R. Caro-Romero, M. Amor, L. Fuentes, Evolving dynamic self-adaptation policies of mhealth systems for long-term monitoring, J. Biomed. Inform. 108 (2020) 103494, http://dx.doi.org/10.1016/j.jbi.2020.103494.

[6] F.S. Lopez, N. Condori-Fernández, Design of an adaptive persuasive mobile application for stimulating the medication adherence, in: Intelligent Technologies for Interactive Entertainment - 8th International Conference, INTETAIN 2016, Revised Selected Papers, Vol. 178, Springer, 2016, pp. 99–105, http://dx.doi.org/10.1007/978-3-319-49616-0_9.

[7] R. Mizouni, M.A. Matar, Z. Al Mahmoud, S. Alzahmi, A. Salah, A framework for context-aware self-adaptive mobile applications SPL, Expert Syst. Appl. 41 (16) (2014) 7549–7564, http://dx.doi.org/10.1016/j.eswa.2014.05.049.

[8] D. Preuveneers, N.Z. Naqvi, A. Ramakrishnan, Y. Berbers, W. Joosen, Adaptive dissemination for mobile electronic health record applications with proactive situational awareness, in: 2016 49th Hawaii International Conference on System Sciences (HICSS), IEEE, 2016, pp. 3229–3238, http://dx.doi.org/10.1109/HICSS.2016.405.

[9] F. Sartori, L.L. Tonelli, Fuzzy personalization of mobile apps: A case study from mhealth domain, in: Information Systems Development: Crossing Boundaries Between Development and Operations (DevOps) in Information Systems (ISD2021 Proceedings), 2021, URL https://aisel.aisnet.org/isd2014/proceedings2021/hci/4.

[10] L. Gamberini, P. Pluchino, L. Porto, F. Zordan, A. Monterosso, K. Palanivel, A. Fontanari, Patchai: An e-health application powered by an ai virtual assistant to support patients in their clinical trials, in: C. Stephanidis, M. Antona, S. Ntoa (Eds.), HCI International 2021 - Posters, Springer International Publishing, Cham, 2021, pp. 315–322.

[11] C. Burley, D. Anderson, A. Brownlee, G. Lafer, T. Luong, M. McGowan, J. Nguyen, W. Trotter, H. Wine, A. Baglione, et al., Increasing engagement in ehealth interventions using personalization and implementation intentions, in: 2020 Systems and Information Engineering Design Symposium (SIEDS), IEEE, 2020, pp. 1–5.

[12] E.M. Grua, M. De Sanctis, P. Lago, A reference architecture for personalized and self-adaptive e-health apps, in: Software Architecture: 14th European Conference, ECSA 2020 Tracks and Workshops, 2020, Proceedings, Springer, 2020, pp. 195–209, http://dx.doi.org/10.1007/978-3-030-59155-7_15.

[13] E.M. Grua, M. De Sanctis, I. Malavolta, M. Hoogendoorn, P. Lago, Social sustainability in the e-health domain via personalized and self-adaptive mobile apps, in: M. Piattini C. Calero (Ed.), Software Sustainability, Springer International Publishing, Cham, 2021, pp. 301–328, http://dx.doi.org/10.1007/978-3-030-69970-3_13.

[14] H. Fan, M.S. Poole, What is personalization? Perspectives on the design and implementation of personalization in information systems, J. Org. Comput. Electron. Commer. 16 (3–4) (2006) 179–202, http://dx.doi.org/10.1080/10919392.2006.9681199.

[15] D. Weyns, Software engineering of self-adaptive systems, in: Handbook of Software Engineering, 2019, pp. 399–443, http://dx.doi.org/10.1007/978-3-030-00262-6_11.

[16] E.M. Grua, M. Hoogendoorn, I. Malavolta, P. Lago, A. Eiben, Clustream-GT: Online clustering for personalization in the health domain, in: IEEE/WIC/ACM International Conference on Web Intelligence, ACM, 2019, pp. 270–275, http://dx.doi.org/10.1145/3350546.3352529.

[17] IBM, An Architectural Blueprint for Autonomic Computing, Tech. rep., IBM, 2006.

[18] K.-j. Kim, H. Ahn, Using a clustering genetic algorithm to support customer segmentation for personalized recommender systems, in: International Conference on AI, Simulation, and Planning in High Autonomy Systems, Springer, 2004, pp. 409–415, http://dx.doi.org/10.1007/978-3-540-30583-5_44.

[19] L. Baresi, W.G. Griswold, G.A. Lewis, M. Autili, I. Malavolta, C. Julien, Trends and challenges for software engineering in the mobile domain, IEEE Software 38 (1) (2020) 88–96, http://dx.doi.org/10.1109/MS.2020.2994306.

[20] I. Malavolta, E.M. Grua, C.-Y. Lam, R. de Vries, F. Tan, E. Zielinski, M. Peters, L. Kaandorp, A framework for the automatic execution of measurement-based experiments on android devices, in: Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering Workshops, in: ASE '20, 2020, pp. 61–66, http://dx.doi.org/10.1145/3417113.3422184.

[21] R. Van Solingen, V. Basili, G. Caldiera, H.D. Rombach, Goal question metric (gqm) approach, Encyclopedia of software engineering. http://dx.doi.org/10.1002/0471028959.

[22] J. Brooke, et al., SUS-A quick and dirty usability scale, Usability Eval. Ind. 189 (194) (1996) 4–7.

[23] S.S. Shapiro, M.B. Wilk, An analysis of variance test for normality (complete samples), Biometrika 52 (3/4) (1965) 591–611.

[24] P.E. McKnight, J. Najab, Mann-whitney u test, in: The Corsini Encyclopedia of Psychology, 2010, p. 1, http://dx.doi.org/10.1002/9780470479216.

[25] N. Cliff, Dominance statistics: Ordinal analyses to answer ordinal questions, Psychol. Bull. 114 (3) (1993) 494.

[26] D. Di Nucci, F. Palomba, A. Prota, A. Panichella, A. Zaidman, A. De Lucia, Software-based energy profiling of android apps: Simple, efficient and reliable? in: 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), IEEE, 2017, pp. 103–114, http://dx.doi.org/10.1109/SANER.2017.7884613.

[27] S. Raheel, Improving the user experience using an intelligent adaptive user interface in mobile applications, in: 2016 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET), IEEE, 2016, pp. 64–68, http://dx.doi.org/10.1109/IMCET.2016.7777428.

[28] A. Cañete, J.-M. Horcas, I. Ayala, L. Fuentes, Energy efficient adaptation engines for android applications, Inf. Softw. Technol. 118 (2020) 106220, http://dx.doi.org/10.1016/j.infsof.2019.106220.