FAMAP: A Framework for Developing m-Health Apps

Chapter · March 2018

DOI: 10.1007/978-3-319-77703-0_83

Iván García-Magariño
Complutense University of Madrid

Guillermo Palacios
University of Zaragoza

FAMAP: A framework for developing m-health apps

Iván García-Magariño[1,2], Manuel Gonzalez Bedia[1], and Guillermo Palacios-Navarro[3]

[1]Department of Computer Science and Engineering of Systems, University of Zaragoza, Teruel and Zaragoza, Spain
[2]Instituto de Investigación Sanitaria Aragón, University of Zaragoza, Zaragoza
[3]Deparment of Electronic Engineering and Communications, University of Zaragoza, Teruel, Spain

NOTICE: This is the authors' version of an article that was published in *WorldCIST'18 2018: Trends and Advances in Information Systems and Technologies*. The original publication is available at Springer via https://doi.org/10.1007/978-3-319-77703-0_83

PLEASE CITE AS:
García-Magariño, Gonzalez Bedia, M., & Palacios-Navarro, P. (2018). FAMAP: A Framework for Developing m-Health Apps. In WorldCIST'18 2018: Trends and Advances in Information Systems and Technologies, vol. 745 of Advances in Intelligent Systems and Computing (pp. 850-859). Springer, Cham

Abstract. The edge-cutting mobile technologies have allowed the expansion of m-health applications for both patients and doctors. However, the variety of technologies, platforms and general-purpose development frameworks make developers and researchers to spend a considerable amount of time in developing m-health apps from scratch. This paper presents an ongoing research project about the creation of a framework for assisting developers and researchers in creating m-health apps called FAMAP. This framework is presented for the first time in the current article. Among others, this framework contains components for respectively (1) collecting data, (2) visualizing data analytics, (3) automating the definition and management of questionnaires, (4) implementing agent-based decision support systems and (5) supporting multi-modal communication. To show the utility of the proposed framework, this article presents some well-known and in-progress m-health apps developed with this framework. This work is assessed by considering (a) the usage data to show the commitment of users in one of the apps, and (b) the downloads and ranking in stores of another of the apps.

Keywords: m-health, software development, mobile application, app, well-being, multi-agent system

1 Introduction

The last years have witnessed a steep increase in the usage of health and well-being apps. For example, the work about IntelliCare suite [15] showed 10 K downloads of health and well-being apps with their suite in the first public year. In addition, the number of health and well-being apps in specific domains is also increasing. For instance, one can observe the spread of mindfulness-based apps for both Android [20] and Apple iOS [16] platforms.

The general-purpose engines such as Apache Cordova, PhoneGap and Unity solves the development of multi-platform apps that can be deployed for different operative systems such as Android and iOS [18]. However, these engines lack a customized environment that specifically addresses the common needs of mobile health (m-health) and well-being apps.

In this context, the principles of model-driven development (MDD) propose to use domain-specific languages for a high-level definition of systems [22]. Model transformations allows transforming domain-specific models into implementations in the different platforms by means of chains of transformations [8].

Moreover, multi-agent systems (MASs) have proven to be useful in m-health apps such as the ones related with ambient assisted living of Parkinson patients [4]. More concretely, agent-based simulators (ABS) have been developed to support decisions in the health and well-being field. For instance, the recent ABS named ABS-MindHeart allowed experts to simulate different mindfulness programs to determine their possible repercussions in heart-rate variability (HRV) [5] of a group of people. HRV is one of the most common health indicators [23].

The current work proposes a framework for assisting the development of apps for m-health and well-being called FAMAP. This framework contains components that directly support some recurrent operations in these kinds of apps.

2 Related work

In the literature, the development of m-health applications has drawn attention from many perspectives such as (a) the app evaluation, (b) the user-centered models, (c) the information system development, and (d) the security in the health information collected from the Internet of Things (IoT).

Regarding the evaluation of health apps, Amalfitano et al [1] proposed a framework for comparing testing techniques that evaluate mobile apps. They provided a mechanism to assess their user interfaces (UIs). In general, the compared offline and online testing techniques recommending the later ones. In this line of research, Grundy et al [13] had previously reviewed the challenges in assessing m-health apps. They not only considered the technological aspects such as quality and safety of the app, but also took into account the evaluation of their contents. However, these works did not assist the development of apps beyond their evaluation.

Concerning the development of m-health apps for improving the user experience, Schnall et al [21] presented a user-centered model for developing m-health apps focusing on the usability. They tested their approach by developing an app for the HIV prevention. Although their user-centered design was useful to achieve easy-to-use UIs, their approach did not provide the components for facilitating the software development of some recurrent aspects of m-health apps.

Moreover, Miah et al [17] stated that m-health required special attention from the information system viewpoint. Their statement was based on their analysis of the development of m-health applications from 2010 to 2016. They identified eight design issues and they argued their classifications based on their literature review. Nevertheless, they did not provide any software framework for assisting the development of recurrent tasks in these developments.

Bearing in mind the security about health data from IoT, Sicari et al [24] presented a framework for the development of smart health applications based on IoT. Their approach was mainly concerned about guaranteeing security, but it missed other aspects such as the visualization of big data analytics and the automation of questionnaires management.

In conclusion, to the best of the authors' knowledge, the literature lacks a software development framework for assisting developers and researchers in developing m-health and well-being apps by providing support for recurrent aspects such as big data analytics, automation of questionnaires management, multi-modal communication and agent-based decision support systems (DSSs). The next section presents a framework that covers this gap of the literature.

3 FAMAP

A current research project is focused on creating FAMAP for assisting the development of m-health and well-being mobile applications. Figure 1 visually presents the outline of FAMAP. This framework combines de definition of domain-specific models and documents for the creation of multi-platform mobile applications by means of Unity engine. We selected Unity instead of any web-based cross-platform environments with JavaScript for apps, because Unity allows developers to have full control about the interaction patterns with users.

The current framework has components for respectively (a) visualizing big data analytics, (b) automating the definition and management of questionnaires, (c) supporting decisions with ABSs, and (e) collecting information from IoT devices such as wearable sensors. It also allows creating UIs with multi-modal communication.

The collected information is normally sent to a server through Internet. Data are normally associated with the devices, by means of their unique identifiers. When login is required, the information is associated with the corresponding users. These data is normally stored in a database, which can be distributed if the amount of data is expected to be big.

In this framework, the support for ABS is based on the Process for developing ABSs (PEABS) [10]. The framework includes some basic object-oriented classes for supporting the definition of agents, and running the simulation. In addition, the current framework follows the recommendations of TABSAOND

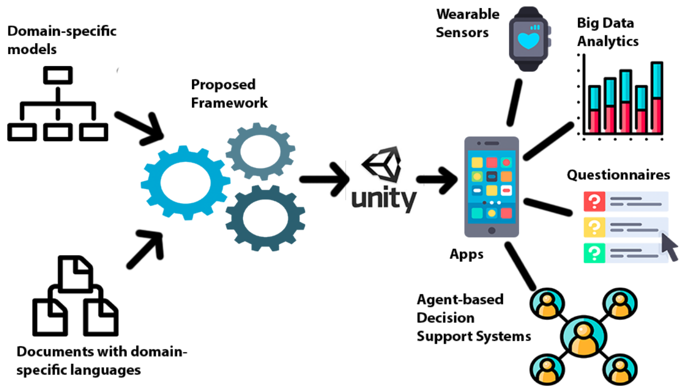


Fig. 1. Outline of FAMAP

(a technique for developing agent-based simulation apps and online tools with nondeterministic decisions) [12]. In addition, it allows one to define coordination mechanisms such as the ones previously proposed for the Delphi method [7]. The underlying framework of PEABS was originally presented for Java and Erlang programming languages, and FAMAP includes its translation into the C# programming languages. Similarly, the underlying necessary functions of TABSAOND were translated from JavaScript to C#. In addition, we considered the common agent-oriented metrics for both architectures [9] and communications [14].

4 Practical applications of FAMAP

We have developed several applications that used the component of FAMAP for presenting big-data analytics. For example, ABS-BedIoT is an ABS that analyzed the big data of the sensors of a simulated smart bed [11]. Figure 2(a) presents the screen of ABS-BedIoT that visualized the big data analytics about time ratios in which the sleeper adopted different sleeping poses. In addition, EmoPaint app also uses this component. EmoPaint allows users to define their bodily sensations maps (BSMs), and the app estimates their emotions from these BSMs. The app visualizes the emotions of users in each month with a starplot using this component (see Figure 2(b)).

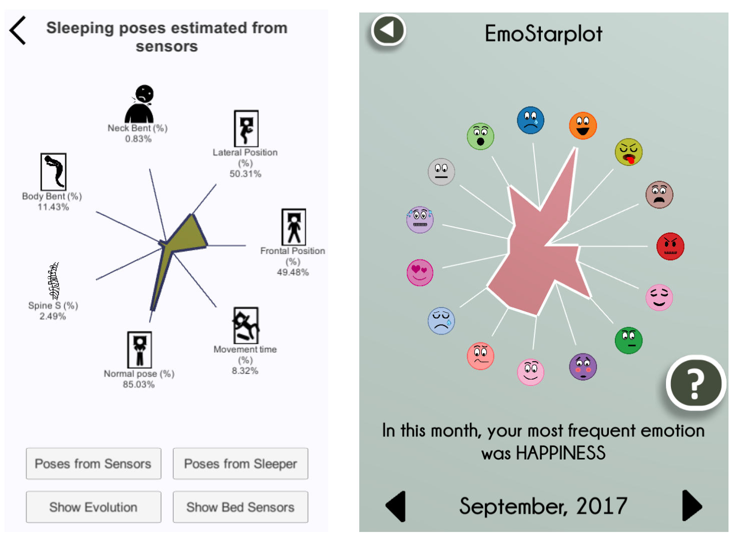
FAMAP allows one to configure usable UIs for introducing the input parameters and configuring some options. Figure 3(a) shows an example of screen for introducing input parameters taken from ABS-MindBurnout app. This app provides an ABS for supporting decisions about the definition of mindfulness programs based on their repercussions on job burnout. In this screen, users can select a default mindfulness programs, or create new ones by selecting a list of activities from the right dropdown. This improves alternative mechanisms of defining mindfulness programs such as the ones used in ABS-MindHeart and ABSEM (an ABS of emotions in mindfulness programs) [6] in terms of usability. In these alternative methods, the mindfulness programs were defined by programming code.

The framework also includes the possibility of defining interactive input UIs based on the interaction with a starplot. In particular, this was used for allowing users to define their emotional states as the combination of the basic emotions of Ekman [2] in the EmoRepair app. The goal of this app was to ameliorate negative emotions of users by recommending them to adopt certain postures based on the principles of embodiment [25].

The current approach supports multi-modal communication. It allows apps not only to present visual information but also to speak aloud information with text-to-speech (TTS) technology. EmoRepair used the corresponding framework module for both presenting a visual countdown and informing the user about it with audio (see Figure 3(b)). The interval of informing the countdown by voice is configurable.

EmoRepair also used the component for automatically saving all the options introduced by the user just when changing any of its values. This component is generic, and is designed for storing any list of numeric options. In particular, EmoRepair used the options to determine the time for adopting each posture and time interval for informing the user of the countdown by audio.

The current framework also includes a component for automatically managing the questionnaires from a document that uses a domain specific language.



(a) Example taken from ABS-SmartBedIot (b) Example taken from EmoPaint

Fig. 2. Examples of apps with data analytics using starplots by the proposed framework

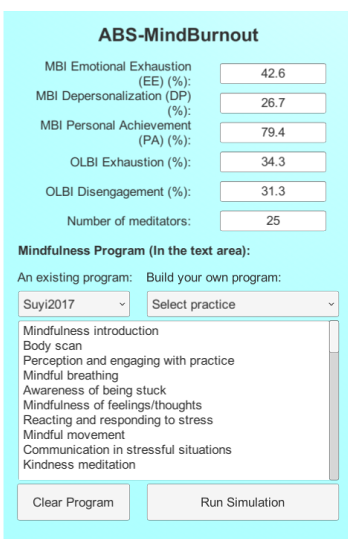
In particular, the documents use the following format for each line representing a question:

<question>[| type] [| <list answers>]

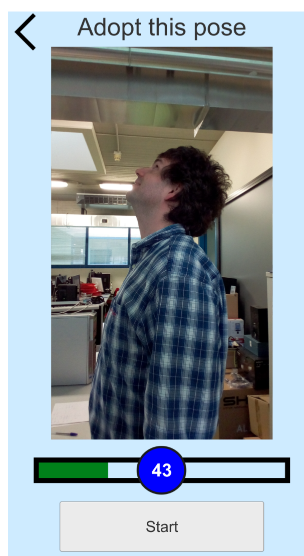
If the “type” of the question is omitted, it is assumed to have the same type as the previous question. This facilitates the fast definition of questionnaires and flexibility. For example, this component allows one to define all the questions in the same Likert scale by just defining the possible answers for the first question. This component also has the flexibility of defining questionnaires that include different types of questions. For example, this framework allows one to define a Likert questionnaire with some open questions at the end by just defining only the type of the first Likert question and the first open one.

For now, the questionnaire module supports (a) multiple-answer questions, including Likert scales or any other question with a list of multiple choices, (b) open questions, in which the user can enter any text, and (c) numeric questions, in which the user can reply with any number. In the multiple-answer questions, the list of answers is defined with the “—” separator character.

The framework can dynamically generate the visual interface of any questionnaire with this text format, generating the scrollable panel with the corresponding labels, text fields, and lists of answers with check boxes. Figure 4 shows an example of a dynamically generated questionnaire with the current



(a) Screen for entering input parameters in ABS-MindBurnout



(b) Example of multi-modal interface from EmoRepair app

Fig. 3. Examples of UIs supported by the current framework

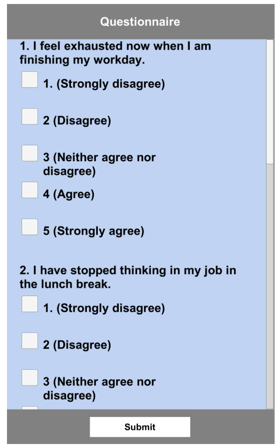


Fig. 4. Example of the visual interface automatically generated for a questionnaire

approach in the Eric app. This app was developed for tracking the repercussion of a mindfulness-based intervention on the health of employees.

The responses of all the questionnaires are separately stored in tables for respectively each kind of answer including the keys (submission date and time, device ID and questionnaire ID) for properly identifying each response. This design allows to maintain, modify and create new questionnaires without altering the database. In order to friendly show a table with all the responses for a given questionnaire, the framework module allows one to automatically generate the query with SQL (Structured Query Language) for creating a database view from the text-format questionnaire definition. The flexibility of this approach partially addresses the challenge of rapid maintainability of the steep growing of data with changeable structures stated by the big data field [19].

This framework also includes a component for associating the questionnaires with a given frequency or specific dates. In this way, apps can schedule notifications for reminding the users to answer the questionnaires. In particular, Eric app schedules daily notifications in working days for one-month interventions. In addition, EmoPaint schedules the notifications of questionnaires for two and four weeks later after installation.

5 Evaluation

In order to assess the current approach, we consider the number of apps developed with the current approach and the usage of some of these apps. It is worth mentioning that four apps have been successfully developed using components of this framework. These apps are ABS-BedIoT, EmoPaint, ABS-MindBurnout, and Eric. The two apps with ABSs for supporting decisions respectively have been tested (or are being tested) with research studies involving comparisons of real and simulated values.

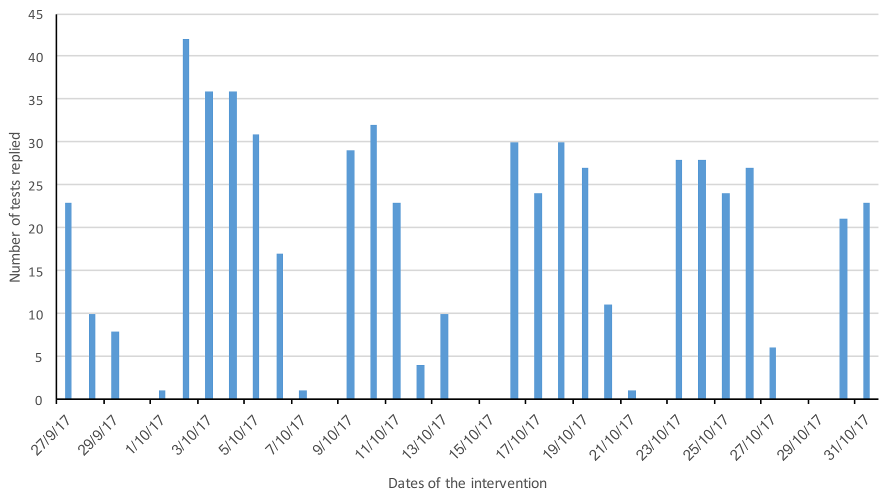


Fig. 5. Number of tests replied per day in the Eric app

EmoPaint and Eric are being tested with users. In particular, Eric has been used with a group of employees. Figure 5 shows the numbers of people that replied the questionnaire in each day of a mindfulness-based intervention. Notice that the app only reminded them to reply the questionnaire in the working days (i.e. from Monday to Friday). One can observe that the participation ratio remained high for a whole month, which is a great success in comparison with other studies where the participation normally drops down [3].

Moreover, EmoPaint was released in Google Play and Apple App Store in October 5, 2017, and had over 100 downloads in its first month. This app was promoted trough a video and snapshots in the stores and social media. In addition, it was ranked with an average of 5.0 out of five by seven people (according to the data accessed on November 15, 2017 from Google Play). This revealed that users may have found this app interesting and useful.

6 Conclusions and future work

The current work presents an ongoing research project about the construction of a framework that facilitates the development of m-health and well-being apps. Most of the components of the FAMAP have been developed, and these have been successfully used for implementing four apps in the context of m-health and well-being. The usage and downloads of two of these apps reveal that these apps are probably useful for the users.

The development of the current framework is planned to continue. First, a module will be developed for communicating the apps with IoT devices such as wearable sensors like smart bands. The communication with near wearable sensors will use Bluetooth. The framework is planned to be packed and distributed as open source. The development time will be analyzed for determining whether de current approach really allows developers to reduce development time in comparison with other alternative approaches.

Acknowledgments We acknowledge the research project “Construcción de un framework para agilizar el desarrollo de aplicaciones móviles en el ámbito de la salud” funded by University of Zaragoza and Foundation Ibercaja with grant reference JIUZ-2017-TEC-03.

Bibliography

[1] Amalfitano D, Amatucci N, Memon AM, Tramontana P, Fasolino AR (2017) A general framework for comparing automatic testing techniques of android mobile apps. Journal of Systems and Software 125:322–343

[2] Ekman P (1992) An argument for basic emotions. Cognition & emotion 6(3-4):169–200

[3] Galesic M (2006) Dropouts on the web: Effects of interest and burden experienced during an online survey. Journal of Official Statistics 22(2):313

[4] García-Magariño I, Palacios-Navarro G (2016) A model-driven approach for constructing ambient assisted-living multi-agent systems customized for parkinson patients. Journal of Systems and Software 111:34–48

[5] García-Magariño I, Plaza I (2017) ABS-MindHeart: An agent based simulator of the influence of mindfulness programs on heart rate variability. Journal of Computational Science 19:11–20

[6] García-Magariño I, Plaza I (2017) ABSEM: An agent-based simulator of emotions in mindfulness programs. Expert Systems with Applications 84:49–57

[7] García-Magariño I, Gómez-Sanz JJ, Pérez-Agüera JR (2008) A multi-agent based implementation of a delphi process. In: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3, International Foundation for Autonomous Agents and Multiagent Systems, pp 1543–1546

[8] García-Magariño I, Rougemaille S, Fuentes-Fernández R, Migeon F, Gleizes MP, Gómez-Sanz J (2009) A tool for generating model transformations by-example in multi-agent systems. In: 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009), Springer, Advances in Intelligent and Soft Computing, vol 55, pp 70–79

[9] García-Magariño I, Cossentino M, Seidita V (2010) A metrics suite for evaluating agent-oriented architectures. In: Proceedings of the 2010 ACM Symposium on Applied Computing, ACM, pp 912–919

[10] García-Magariño I, Gómez-Rodríguez A, González-Moreno JC, Palacios-Navarro G (2015) PEABS: a process for developing efficient agent-based simulators. Engineering Applications of Artificial Intelligence 46:104–112

[11] García-Magariño I, Lacuesta R, Lloret J (2017) Agent-based simulation of smart beds with Internet-of-Things for exploring big data analytics. IEEE Access https://doi.org/10.1109/ACCESS.2017.2764467

[12] García-Magariño I, Palacios-Navarro G, Lacuesta R (2017) TABSAOND: A technique for developing agent-based simulation apps and online tools with nondeterministic decisions. Simulation Modelling Practice and Theory 77:84–107

[13] Grundy QH, Wang Z, Bero LA (2016) Challenges in assessing mobile health app quality: a systematic review of prevalent and innovative methods. American journal of preventive medicine 51(6):1051–1059

[14] Gutierrez C, Garcia-Magariño I (2009) A metrics suite for the communication of multi-agent systems. Journal of Physical Agents 3(2):7–14
[15] Lattie EG, Schueller SM, Sargent E, Stiles-Shields C, Tomasino KN, Corden ME, Begale M, Karr CJ, Mohr DC (2016) Uptake and usage of intellicare: a publicly available suite of mental health and well-being apps. Internet interventions 4:152–158
[16] Mani M, Kavanagh DJ, Hides L, Stoyanov SR (2015) Review and evaluation of mindfulness-based iphone apps. JMIR mHealth and uHealth 3(3)
[17] Miah SJ, Gammack J, Hasan N (2017) Extending the framework for mobile health information systems research: A content analysis. Information Systems 69:1–24
[18] Mukherjee S, Mondal I (2014) Future practicability of android application development with new android libraries and frameworks. International Journal of Computer Science and Information Technologies 5(4):5575–5579
[19] Olayinka O, Kekeh M, Sheth-Chandra M, Akpinar-Elci M (2017) Big data knowledge in global health education. Annals of Global Health https://doi.org/10.1016/j.aogh.2017.09.005
[20] Plaza García I, Sánchez CM, Espilez ÁS, García-Magariño I, Guillen GA, García-Campayo J (2017) Development and initial evaluation of a mobile application to help with mindfulness training and practice. International Journal of Medical Informatics 105:59–67
[21] Schnall R, Rojas M, Bakken S, Brown W, Carballo-Dieguez A, Carry M, Gelaude D, Mosley JP, Travers J (2016) A user-centered model for designing consumer mobile health (mhealth) applications (apps). Journal of biomedical informatics 60:243–251
[22] Selic B (2003) The pragmatics of model-driven development. IEEE software 20(5):19–25
[23] Shastry N, Mirajkar AM, Moodithaya SS, Halahalli HN (2017) Resting heart rate variability and cardiorespiratory fitness in healthy young adults. Indian Journal of Medical Specialities 8(1):17–20
[24] Sicari S, Rizzardi A, Grieco L, Piro G, Coen-Porisini A (2017) A policy enforcement framework for internet of things applications in the smart health. Smart Health 3–4:39–74
[25] Wilkes C, Kydd R, Sagar M, Broadbent E (2017) Upright posture improves affect and fatigue in people with depressive symptoms. Journal of behavior therapy and experimental psychiatry 54:143–149