Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2018) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

The 2nd International Workshop on Healthcare Interoperability and Pervasive Intelligent Systems (HIPIS 2018)

# Step Towards Progressive Web Development in Obstetrics

Patrícia Loreto[a], Jorge Braga[b], Hugo Peixoto[c,*], José Machado[c], António Abelha[c]

[a] *University of Minho, Campus Gualtar, Braga 4710, Portugal*
[b] *Centro Hospitalar do Porto, Centro Materno Infantil, Porto 4099, Portugal*
[c] *Algoritmi Center, University of Minho, Campus Gualtar, Braga 4710, Portugal*

**Abstract**

The aim of this paper is to develop a Personal Health Record (PHR) for the support of pregnant women. With this goal in mind, concepts such as PHR and their importance in the obstetrics field are overviewed, as well as mobile development strategies. The system was developed with the support of a medical institution and taking into account what pregnant women find useful. The developed app is a Progressive Web App (PWA). This is a recent technology that allows the same app to work on most devices, gives a native feel to it when using on mobile devices and enables offline support. Further testing is necessary to understand the impact that this system may have in the engagement of pregnant women and in birth outcomes.



*Keywords:* Personal Health Records, Obstetrics, Mobile Development, Progressive Web Apps

## 1. Introduction

For the past years, with the developments in the health information systems field, there has been an increasing interest in Personal Health Records [1]. A Personal Health Record (PHR) is an electronic document that contains medical related information maintained by the patient himself or by a family member. It provides a more detailed summary of the patient's health history and turns the patient a vital stakeholder in his own health management [2, 3].

PHRs have shown to improve medical decisions and therefore the services delivered by health professionals, while also reducing their cost [2, 3]. Most PHRs provide ways of recording clinical values, as well as other features, such as alarming the patient when abnormal values are detected, giving personalized information or displaying appointments [1, 4]. Although PHRs have proven their value in improving healthcare services, there are still barriers to a more wide adoption. One of the main challenges is usability and availability, since it is difficult to develop a system that can be used by all patients no matter their age, medical condition, economic status or technologies they use [3, 4, 5].

* Corresponding author. Tel.: +351 253 604 430 ; fax: +351 253 604 471.
*E-mail address:* hpeixoto@di.uminho.pt

PHRs can be desktop-based, Internet-based or mobile [3]. However, given the previously presented barriers to PHRs adoption, it would be of great interest to find a solution that reaches as many people as possible. To do this, it is necessary to develop a system that can adapt itself to work properly in as many devices as possible.

In this paper a PHR for pregnant women is proposed. This system will be developed using the Progressive Web App (PWA) methodology. Given this, in the next section a state-of-the-art overview of PHRs for the monitoring of pregnant women and of the current mobile development strategies is presented. Then, the PWA methodology and its main characteristics and advantages are explained. Finally, the developed PHR is presented and discussed.

## 2. State of the Art

### *2.1. PHRs for the monitoring of pregnant women*

One of the areas where PHRs have shown to be a relevant tool for the monitoring of health indicators is the Obstetrics field. PHRs are specially important in the monitoring of pregnant women, since pregnancy is a delicate medical condition that needs extensive monitoring and where detecting problems in time is crucial [6, 7, 8]. PHRs may save the woman's medical history and other parameters, such as weight or blood pressure, making it possible for caregivers to access these parameters in case of emergency [6]. However, PHRs can have other features like giving information about the fetus and the changes in the woman's body and a calendar for appointments [6, 9]. If the PHR is connected to the institution that provides care to the pregnant woman, other advantages arise. Decision Support Systems (DSS) help health professionals in the decision-making process and PHRs could be a way of providing more information and more quality information to DSS, which may improve results [10, 11].

Bachiri, Idri, Aléman & Toval (2016) conducted a Systematic Literature Review to analyze the features of mobile PHRs for pregnancy monitoring. 33 PHRs were analyzed and all of them estimated the due date, making this the only feature present in all PHRs. The second most frequent functionality was the monitoring of haelth indicators, such as weight or waist measurements. As for other features, 94% of the analyzed applications displayed information related to the pregnancy state and 87% had a countdown until the due date or a progress bar [12].

To examine the impact that a mobile health application may have in birth outcomes and user engagement, Bush et al. (2017) developed a study with two groups of pregnant women: one that used a mobile application provided by them and one that did not. The group that used the application was small compared to the other group, not making it possible to generalize the conclusions, but the use of the application was associated with a small increase in the attendance of prenatal visits in the 6 months previous to birth and a lower incidence of low-birth weight newborns [8].

### *2.2. Mobile Development Strategies*

According to the 2016 edition of the Statistical Yearbook from the United Nations, in 2015, there were 110 cellular mobile telephone subscriptions per 100 inhabitants in Portugal [13]. A report from the Portuguese National Institute of Statistics (Instituto Nacional de Estatística) from the same year, found that 75% of families have access to the Internet at home, 70% of the people between 16 and 74 years used the Internet in the 3 months previous to the interview and the devices that are most used to access the Internet were mobile phones or smartphones (78%) [14].

Nowadays, strategies for mobile development can be divided in two main groups: native development and web-based development. Native mobile apps are developed only for one platform and use programming languages and tools specific of that platform. For example, an Android native app is developed in Java and uses the Android Software Development Kit (SDK). Native apps have high performance and good User Experience (UX). However, since code from a platform can not be reused in another one, it leads to heavy development and high maintenance costs. On the other hand, web-based development uses web technologies, such as HTML5, CSS and JavaScript, to develop mobile apps that can work in any mobile platform, overcoming this problem [15, 16, 17].

Web-based development strategies can be divided into three groups: Mobile Web Apps, Hybrid Mobile Apps and PWAs. Mobile Web Apps are websites optimized for mobile use that are accessed in browsers. Since standard web technologies are used, the user experience is the same no matter the platform. This allows faster development and simpler maintenance. Even though these apps can access the camera or the microphone, for example, they fail at accessing other features. They also struggle with heavy graphics and cannot be distributed in app stores [16]. Hybrid

Mobile Apps have a wrapper that makes them available for distribution in any platform. This wrapper is achieved by using a hybrid development framework (e.g. PhoneGap) and provides an Application Programming Interface (API) that matches the requests to their equivalents in the corresponding platform. In this methodology, the same code is reusable for all platforms allowing simple development and maintenance, but not all native APIs are provided by the wrapper [16, 18, 19]. PWAs are enhanced Mobile Web Apps. They run as well in browsers, but support other features, such as push notifications or offline support. They are opened in the browser and ask the user if he wants to install the app. If the user accepts, the app is added to the home screen and is accessed and used just like a native app [15, 16, 18].

PWAs are better than web apps and can be an alternative to hybrid apps. They have the potential to work in any mobile device or computer, which neither native apps or hybrid apps can. In fact, some known technological companies have switched to this methodology. One of these companies is Pinterest. Pinterest realized that their website only converted 1% of users into signups, logins or native app installs and they wanted to improve these values. The change from their old mobile site to the new one improved the time spent by users in their website, as well as the core engagements. The results were even better than the results from their native apps. Aside from this, loading times on mobile were drastically reduced. By combining other techniques to optimize the app size (e.g. image optimization), the PWA was much smaller than the native apps as well [20].

Given this, PWAs have the potential to overcome one of the barriers to the adoption of PHRs presented above, usability and availability. Therefore, the PWA methodology was chosen to develop this PHR.

## 3. Progressive Web Apps

Progressive Web Apps were first introduced in 2015 and are websites optimized to work on mobile devices [21]. PWAs are available for everyone no matter the browser or device, are installable in the user's home screen like a regular app and their content is always displayed properly. They are served through a HTTPS connection and do not depend on Internet connectivity, because the service worker will handle the app behaviour when offline or on low-quality networks. The service worker will also make sure the app is always updated and combined with the app manifest makes the app discoverable in search engines [15, 21, 22].

The first time a PWA is visited it is accessed like any other web app through the browser. Then, the user is prompted to install the app. This prompt is only fired if the app is not already installed, the user interacted with the app for at least 30 seconds, it includes a web app manifest with the minimum parameters, is served over HTTPS and has a service worker registered [23]. If the user accepts the prompt, the app is added to the home screen [15].

The core of a PWA is the service worker. It is a JavaScript file responsible for caching resources and serving cached resources, which decreases loading time and makes the app capable of working offline, and makes it possible to use advanced features such as notifications or push messages [21, 24]. The first step in a service worker lifecycle is Registration. During Registration, the browser is told where the service worker is and to start installing it. When Registration ends, Installation happens. During the Installation event it is possible to perform some tasks, such as precaching resources. Finally, Activation happens and the service worker controls all pages and resources within its scope and listens to events [24].

One of the biggest advantages of PWA is the offline support they provide. As previously explained, this is possible due to service workers, because they have the ability to use the Cache interface. The storage from the Cache interface is independent from the browser's HTTP cache and, therefore, it is available offline. Aside from the cache, service workers can also use IndexedDB to store data. Using the offline capabilities of PWA, the overall UX improves, since it results in faster loading [25]. A possible way of enabling offline support is to cache the static data during Installation and have the service worker listening for resource requests. When a resource is requested, the service worker checks if it is in the cache and only if the resource is not available, it uses the network. When the last option happens, the service worker may copy the resource and cache it for later use [25]. There are other strategies to deal with resources, such as cache only, network only, cache falling back to network, network falling back to cache and stale while revalidate [26].

## 4. Pregnant Women Health Record Application

In this section the process of creating a PHR for the monitoring of pregnant women using the PWA methodology is presented. This PHR was developed in partnership with the Centro Materno Infantil do Norte (CMIN), a Portuguese

Maternity. This maternity provided all the information displayed in the app, making it a reliable source of information for pregnant women, and it also allowed us to develop the app taking into account the feedback pregnant women provided of what they enjoy in the apps they use and what they would like to see in a app.

Given this, the main features that the app provides are:

- A calculator for the due date;
- A calendar where the pregnant woman can save important appointments;
- The ability to save documents, such as ultrasounds or analysis;
- A section of frequently asked questions (FAQs) about the pregnancy and labour that women usually have;
- Personalized health information according to health conditions the pregnant woman might have (e.g. diabetes);
- Information updated weekly about the changes occurring in the pregnant woman's body and in the fetus;
- The recording of health indicators such as weight, glucose and heart pressure.

For the development of this application, one of the main goals was to make it available for as many pregnant women as possible so that it could have a wide adoption. Thus, the PWA methodology was chosen. First, a single-page web application was developed and, for the development, JavaScript-based and open source technologies were chosen: React for the front end, Node.js with Express.js for the back end and MongoDB as the database. The workflow of the app is presented in Figure 1. One of the main concerns when developing the web app was to make the front end responsive, so that pages can adapt themselves to different screen sizes and can work properly on all devices. To achieve this, Bootstrap was used in the form of React-Bootstrap, a module that joins Bootstrap and React.

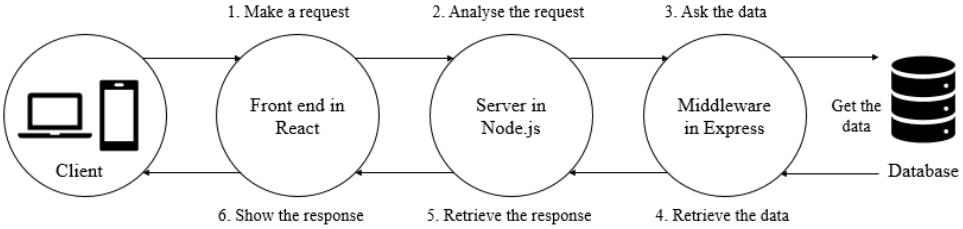
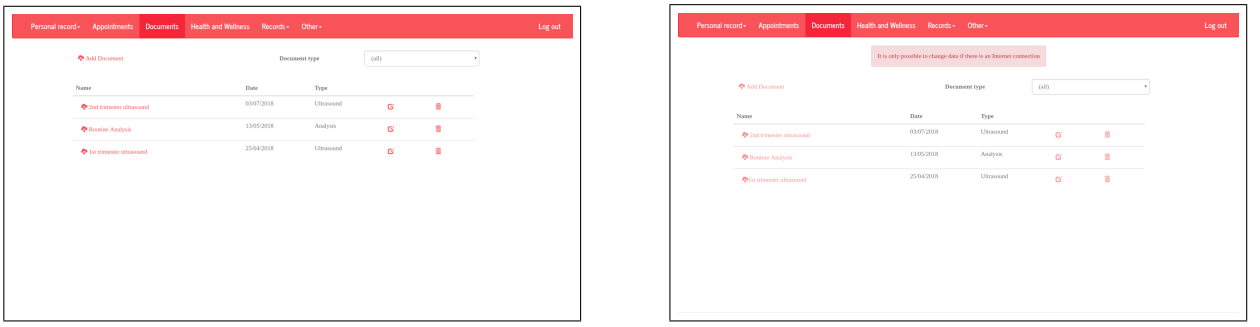



Figure 1: Workflow for the developed web app.

After this, the developed web app was transformed into a PWA. To transform the web app into a progressive one, it was necessary to add a service worker, make it work offline and add an app manifest. The developed service worker caches assets, such as CSS or JS files, and all the URLs from the web app. When the client requests an asset, the request is intercepted by the service worker, instead of going directly to the server. Then, the service worker acts on that request by retrieving data, for example. This makes the app faster. To make sure that assets are updated, a cache version number is defined. When the service worker becomes active, it checks if there are old caches. If there are, it deletes the old ones. Still during the Activation stage, the service worker adds some data to IndexedDB. This data consists of static content that will show up in some pages, for example the FAQs data.

When it comes to working offline, we defined that all URLs should work even if there is no Internet connection, that the most recent data should always be available and that the woman should only update her data if online. The first condition was accomplished by precaching all URLs. This condition is specially important on mobile, because it gives a more native feel to the app if the pregnant woman adds it to her home screen. The second condition, was accomplished by saving the most recent user data and content on her device and updating it when there is an Internet connection. Finally, to make the user only update data when online, all the buttons that lead to data update are disabled and a warning is presented, when there is no Internet connection. An example of the offline capability in action is presented in Figure 4.

Lastly, the app manifest was added, so that the users can add the app to their home screen on mobile. For the app manifest, an icon, a name, short name, start url and the display mode (fullscreen) were provided as well as a theme color and a background color.




(a) Online.

(b) Offline.

Figure 2: The same page when the user is online and offline. When the user is offline, all buttons that edit or add data are disabled and a warning is presented. Still, the data that was available when the user was online for the last time is displayed.

## 5. Discussion

The developed system has both strengths and weaknesses. One of the factors that may difficult the adoption of this app is browser compatibility. This is not an issue if the user uses Google Chrome, because the main driver behind PWAs is Google. For other platforms, this issue tends to decrease over time. For example, with the latest release of iOS Safari in last March (version 11.3), service workers are now compatible. Even though the app works offline, there are some features available only online. This could be solved using the Background Sync API, but it is not available in some browsers (e.g. Safari), so the work that would be applied to enable these functionalities would not pay off at the moment. However, if this API starts having a more wide spread adoption this may be an interesting area to improve.

The app works on all devices as long as there is a browser and adapts itself to different screen sizes. On computers, it works as any other website with the advantage of loading faster after the first visit, since all required resources are cached. On mobile, users can either open it in the browser or add the app to the home screen like a mobile app from the app stores. In both cases, it works offline and has a wide range of features available. The availability in such a wide range of devices may have a positive impact in the adoption of the developed PHR, which was one of the main problems to the adoption of these records. If the user adds the app to the home screen, the access to the app becomes easier and it will look like a native one, since it has a splash screen and is displayed in fullscreen. This, combined with the fact that the app is responsive, improves the UX and gives a more native look to the app.

Since the app was developed taking into account feedback from pregnant women, it was designed with features that they find interesting and useful. The development process also had the help of a medical institution that provided reliable information for the pregnant women. These two conditions combined have the potential to engage pregnant women more and turn them more active in their pregnancy management, which may impact positively birth outcomes.

## 6. Conclusions and Future Work

In this paper a PHR for the support of pregnant women is presented. This PHR is a PWA, which means that it is an enhanced web app. This development methodology makes the same web app available on computers and on mobile phones and has proved to improve user engagement. This was one of the reasons why this methodology was chosen, since PHRs need to find a solution that reaches as many individuals as possible. PWAs can also cache resources, which makes the app faster and able of working offline. The developed PHR was also designed with the support of a medical institution and taking into account feedback provided by pregnant women. This allowed the app to display trustworthy information and to have a set of features that pregnant women find useful. Some of the features provided are a calendar for appointments, weekly information on fetus development and the tracking of some health indicators.

In the future, a usability test should be performed so that the app can be evaluated and improvements can be made. It would also be interesting to study the adoption of this app by pregnant women and the impact that it has in their engagement and birth outcomes.

## Acknowledgements

This work has been supported by Compete: POCI-01-0145-FEDER-007043 and FCT within the Project Scope UID/CEC/00319/2013.

## References

[1] L.-H. Lee, Y.-T. Chou, E.-W. Huang, D.-M. Liou, Design of a personal health record and health knowledge sharing system using ihe-xds and owl, Journal of medical systems 37 (2) (2013) 9921.
[2] I. Genitsaridi, H. Kondylakis, L. Koumakis, K. Marias, M. Tsiknakis, Evaluation of personal health record systems through the lenses of ec research projects, Computers in biology and medicine 59 (2015) 175–185.
[3] L. S. Liu, P. C. Shih, G. R. Hayes, Barriers to the adoption and use of personal health record systems, in: Proceedings of the 2011 iConference, ACM, 2011, pp. 363–370.
[4] N. Archer, U. Fevrier-Thomas, C. Lokker, K. A. McKibbon, S. E. Straus, Personal health records: a scoping review, Journal of the American Medical Informatics Association 18 (4) (2011) 515–522.
[5] J. S. Kahn, V. Aulakh, A. Bosworth, What it takes: characteristics of the ideal personal health record, Health affairs 28 (2) (2009) 369–376.
[6] A. Idri, M. Bachiri, J. L. Fernández-Alemán, A. Toval, Experiment design of free pregnancy monitoring mobile personal health records quality evaluation, in: e-Health Networking, Applications and Services (Healthcom), 2016 IEEE 18th International Conference on, IEEE, 2016, pp. 1–6.
[7] C.-W. Chang, T.-Y. Ma, M.-S. Choi, Y.-Y. Hsu, Y.-J. Tsai, T.-W. Hou, Electronic personal maternity records: both web and smartphone services, Computer methods and programs in biomedicine 121 (1) (2015) 49–58.
[8] J. Bush, D. E. Barlow, J. Echols, J. Wilkerson, K. Bellevin, Impact of a mobile health application on user engagement and pregnancy outcomes among wyoming medicaid members, Telemedicine and e-Health 23 (11) (2017) 891–898.
[9] R. Kennedy, L. Mullaney, C. Reynolds, S. Cawley, D. McCartney, M. Turner, Preferences of women for web-based nutritional information in pregnancy, Public health 143 (2017) 71–77.
[10] H. Peixoto, M. Santos, A. Abelha, J. Machado, Intelligence in interoperability with aida, in: International Symposium on Methodologies for Intelligent Systems, Springer, 2012, pp. 264–273.
[11] A. Cabral, C. Pina, H. Machado, A. Abelha, M. Salazar, C. Quintas, C. F. Portela, J. Machado, J. Neves, M. F. Santos, Data acquisition process for an intelligent decision support in gynecology and obstetrics emergency triage, in: International Conference on ENTERprise Information Systems, Springer, 2011, pp. 223–232.
[12] M. Bachiri, A. Idri, J. L. Fernández-Alemán, A. Toval, Mobile personal health records for pregnancy monitoring functionalities: Analysis and potential, Computer methods and programs in biomedicine 134 (2016) 121–135.
[13] U. Nations, Statistical yearbook 2016 edition (2016).
[14] Instituto Nacional de Estatística, Inquérito à utilização de tecnologias da informação e da comunicação pelas famílias (2016).
[15] T. A. Majchrzak, A. Biørn-Hansen, T.-M. Grønli, Progressive web apps: the definite approach to cross-platform development?, in: Proceedings of the 51st Hawaii International Conference on System Sciences, 2018, pp. 5735–5744.
[16] I. Malavolta, Beyond native apps: web technologies to the rescue!(keynote), in: Proceedings of the 1st International Workshop on Mobile Development, ACM, 2016, pp. 1–2.
[17] S. Valente, J. Braga, J. Machado, M. Santos, A. Abelha, The impact of mobile platforms in obstetrics, Procedia Technology 9 (2013) 1201–1208.
[18] A. Biørn-Hansen, T. A. Majchrzak, T.-M. Grønli, Progressive web apps: The possible web-native unifier for mobile development, in: Proceedings of the 13th International Conference on Web Information Systems and Technologies, 2017, pp. 344–351.
[19] M. Ciman, O. Gaggi, An empirical analysis of energy consumption of cross-platform frameworks for mobile development, Pervasive and Mobile Computing 39 (2017) 214–230.
[20] Addy Osmani, A pinterest progressive web app performance case study, https://medium.com/dev-channel/a-pinterest-progressive-web-app-performance-case-study-3bd6ed2e6154, Last accessed on 2018-08-20 (2017).
[21] I. Malavolta, G. Procaccianti, P. Noorland, P. Vukmirovic, Assessing the impact of service workers on the energy efficiency of progressive web apps, in: Mobile Software Engineering and Systems (MOBILESoft), 2017 IEEE/ACM 4th International Conference on, IEEE, 2017, pp. 35–45.
[22] Google Developers, Your first progressive web app, https://codelabs.developers.google.com/codelabs/your-first-pwapp/, Last accessed on 2018-06-18 (2018).
[23] Google Developers, Add to home screen, https://developers.google.com/web/fundamentals/app-install-banners/, Last accessed on 2018-06-18 (2018).
[24] Google Developers, Introduction to service worker, https://developers.google.com/web/ilt/pwa/introduction-to-service-worker, Last accessed on 2018-06-18 (2018).
[25] Google Developers, Offline quickstart, https://developers.google.com/web/ilt/pwa/offline-quickstart, Last accessed on 2018-06-18 (2018).
[26] Google Developers, The offline cookbook, https://developers.google.com/web/fundamentals/instant-and-offline/offline-cookbook/, Last accessed on 2018-06-18 (2018).