mHealthSwarm: A Unified Platform for mHealth Applications

Ben Philip[1][a], Yasmeen Alshehhi[1][b], Mohamed Abdelrazek[1][c], Scott Barnett[1][d], Alessio Bonti[1][e] and John Grundy[2][f]

[1]*Deakin University, Burwood, VIC, Australia*
[2]*Monash University, Clayton, VIC, Australia*

Keywords: mHealth apps, mHealth platform, micro-mHealth apps

Abstract: Mobile health (mHealth) applications are ubiquitous and offer several benefits such as easier access to one's health and wellness data through smartphones. However, their growth in popularity has also introduced several challenges for both end-users and developers. Users face challenges around the poor user experience (UX) introduced by the need to install several apps and limited customizability which is exacerbated by the limited control over app functionality. While features common across different apps may not directly affect individual developers, the fact that one app may provide a better implementation of features leads to wasted developer effort. A single platform that can satisfy all user needs does not currently exist, and given the diversity of the mHealth domain, a single app would be far too complex if it existed. In this paper, we present a new approach and a platform for mHealth apps - micro-mHealth apps, and we discuss them as an alternative to the current mHealth app development model, which we believe could improve the current state of mHealth app development and adoption. We are currently evaluating our prototype with several micro-mHealth apps built using common features in the mHealth apps available in commercial app stores.

1 INTRODUCTION

mHealth has been defined as the "medical and public health practice supported by mobile devices, such as mobile phones, patient monitoring devices, personal digital assistants (PDAs), and other wireless devices" (WHO, 2011). Today, several health, fitness and wellness applications for uses such as drug dosage and medical reference (Ventola, 2014; Berauk et al., 2017; Sezgin, Özkan-Yildirim, and Yildirim, 2017) and weight management (Higgins, 2015) are popular among users as part of their daily routines (Berauk et al., 2017; Higgins, 2015).

Despite the increase of mHealth apps and interest in adopting them from a great many end users, there are many limitations around existing mHealth app usability, feature relevance, and data sharing. There are an increasing number of users needing to install many mHealth apps, a key challenge as there is no app that delivers a complete set of features that a user might need (Berauk et al., 2017). Users dislike downloading

---

[a] https://orcid.org/0000-0001-9438-4532
[b] https://orcid.org/0000-0002-9432-9477
[c] https://orcid.org/0000-0003-3812-9785
[d] https://orcid.org/0000-0002-3187-4937
[e] https://orcid.org/0000-0003-2240-0454
[f] https://orcid.org/0000-0003-4928-7076

multiple apps as they unnecessarily increase the number of applications on their smartphones (van Velsen, Beaujean, and van Gemert-Pijnen, 2013). Similarly, many users stop using health apps due to the burden of repeated data entry, hidden costs and confusion around several apps (Krebs and Duncan, 2015; Wiechmann, Kwan, Bokarius, and Toohey, 2016).

Data visualization is also a key component of health tracking apps that helps users to track their goals, habits, and achievements (Lee, Choe, Isenberg, Marriott, and Stasko, 2020). However, studies have shown diversity in data visualization audiences (Jena et al., 2021), which in turn introduce challenges related to the user experience around mobile data visualization (Gu, Mackin, and Zheng, 2018) (Katz, Dalton, Holland, O'kane, and Price, 2017). For example, users generally desire more control over charts to better monitor their health progress and goals (Y. Alshehhi, Abdelrazek, and Bonti, 2022). Further, another user survey around the strengths and weaknesses of mHealth app visualizations highlighted the lack of interaction, such as zooming capabilities, as a major limitation (Y. A. Alshehhi, Abdelrazek, and Bonti, 2022). The ability to customize personal goals on charts was also highlighted as one of the desired features. This, in turn, indicates that the current data visualization guidelines lack support for inclusive de-

sign, resulting in a poorer experience.

From the engineering side, developing apps requires writing large amounts of boilerplate code (Barnett, Vasa, and Grundy, 2015) and doing so for several different platforms greatly increases cost and development effort (Pinto and Coutinho, 2018), which cross-platform frameworks aim to simplify. Given the fragmentation among the mHealth apps and current platforms, efforts have been made to overcome these challenges. This includes frameworks for cross-platform development such as Ionic, Cordova and React Native. Although these (limited) frameworks can be used for developing mHealth apps (Banos et al., 2015), they still do not solve the problem of data sharing between apps, integration of wearables and peripherals, a uniform UX/UI and app features customization (Anastasiadou, Folkvord, Serrano-Troncoso, and Lupiañez-Villanueva, 2019). While recent works (Inupakutika, Kaghyan, Akopian, Chalela, and Ramirez, 2020) have attempted to develop cross-platform mHealth solutions that focus on simplifying app development, they fall short on customizability and integration with other services. The development of a "super-app" that covers all major aspects of health was previously suggested by Higgins as a solution to such problems. However, this has its own set of challenges, such as many unused features, diverse user needs and great maintenance issues (Higgins, 2015).

There is thus a need for a new approach and platform to accelerate and support building better collaborative mHealth apps with focused capabilities that can be customized as needed. We propose an approach we term "micro-mHealth apps", and we hypothesize that they will offer a better experience for both end users and mHealth app developers saving a lot of effort, time and money. This paper discusses our motivation and past work supporting this position, a high-level solution architecture and platform details, and discusses the details of developing a micro-mHealth app prototype.

2 MOTIVATION

2.1 Motivating Example

Consider a blood pressure management scenario. Many blood pressure tracker apps are available in commercial app stores. Unfortunately, users often end up downloading different apps with several unused features, and feature overlaps between them (B. Philip, Abdelrazek, Barnett, Bonti, and Grundy, 2022; Y. A. Alshehhi et al., 2022). In the case of data

visualization, users can find that their apps present similar visualizations to track their blood pressure (figure 1) – an example of redundant features. In addition, related information such as weight and glucose measurements are also often presented separately. In such cases, this case, users may need to switch between two or more apps to find meaningful insights derived from different metrics. A previous study highlighted these challenges when users need to use more than one mHealth app, and user expectations, suggesting the need for feature customization that would allow end users to tailor mHealth apps to their needs (B. Philip et al., 2022). Our solution to this problem proposes combining several separate app *features* – micro mHealth apps – into one, to provide a highly customizable, unified mHealth platform for end users.
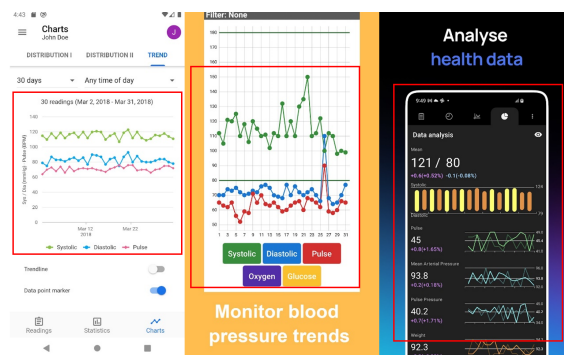

Figure 1: App screenshot showing overlapping charts with single unique feature

2.2 User Challenges and Expectations

Alshehhi et al. conducted a review of user perspectives around visualizations in mHealth apps that indicated several difficulties in chart customizability, interactivity and functionality (Y. Alshehhi et al., 2022). A subsequent anonymous survey allowed them to further investigate these issues, which indicated that the needs of several users have to be considered to improve the acceptance of mHealth data visualizations (Y. A. Alshehhi et al., 2022). Similarly, Philip et al. (B. Philip et al., 2022) conducted an online survey of people having experience with mHealth apps to understand challenges when using several mHealth apps and expectations from future apps. While poor design was highlighted as one reason for poor app adoption, the participants indicated several other challenges such as additional, unnecessary features, feature overlaps between different apps, and repeated manual data entry that can inadvertently drive users away The survey showed that users expect a lot more

from current apps and prefer a single platform that allows feature customization to build an application meeting all their needs without any additional bloat. These studies motivated us to consider the development of a unified mHealth platform to combine together micro mHealth app features, and we hypothesize that such platforms can solve both the developer and end-user challenges.

2.3 Mini Apps

Mobile apps can be classified into three categories – native apps, mobile web apps and hybrid apps (Inupakutika et al., 2020) with their development largely depending on their use. Web and hybrid apps are suggested for better cross-platform compatibility, while native or hybrid apps are suggested for better peripheral support (Serrano, Hernantes, and Gallardo, 2013). mHealth apps can be built using any approach and although the three different formats of mobile apps offer some advantage over each other, they are restricted to single apps - i.e. each package or URL is a single application that runs or renders on the user's smartphone. Hybrid apps offer the benefit of using a web application wrapped inside a native container, which is potentially more beneficial for developers as the web component can be shared across platforms. However, end-users may not see any difference or obtain any direct benefits through such designs.

The concept of "Mini-apps" was introduced in a working draft white paper (*MiniApp Standardization White Paper, W3C First Public Working Draft*, 2019) submitted to the W3C as a new format of mobile apps based on web technologies that integrates with and behave like native apps. Based on hybrid apps, they offer benefits such as being free from installation and storage constraints with the ability to access platform capabilities through native container apps. These apps can be considered across two layers - (1) the Logic Layer responsible for handling the business logic, internal state and integration with system data and services; and (2) the View Layer for rendering the user interface (UI) and handling user interaction.

This structure can be compared to a traditional web application, with the views constructed using HTML and CSS, and the app logic handled using JavaScript. UIs can be rendered using native components such as WebViews which also act as an interface between mini apps and the native container app. Access to resources such as storage and hardware (camera, GPS etc.) cannot be provided directly through the web interface, but through custom APIs provided by the container that allows controlled access to system resources. By extending hybrid applications into

such containers, a whole ecosystem of mini apps can be deployed in a way that facilitates data sharing between them and allows a user to create custom apps with features tailored to their individual needs.

We build on top of this idea for health/fitness/wellness apps by introducing a single, highly customizable super-app and provide a much better health app experience by avoiding the challenges of full apps (Y. Liu, Xu, Ma, and Liu, 2019; Z.-Y. Liu, Liao, and Wang, 2020).

3 OUR PLATFORM

To address the requirements identified in the previous section, we have developed a novel *mHealth-Swarm unified mHealth platform*. This provides a framework for developing health and wellness services and abstracts device features and low level operations into simple high level functionality providing a highly customizable environment that can be personalized by adding/removing single-function mHealth services/apps.

Our mHealthSwarm platform uses a registration mechanism that records app permissions when linked to a user's account. Each time a service app needs user data or access to underlying device hardware, it needs to make a request to the platform using exposed APIs. These requests are then checked, and then the appropriate resource is provided to the requesting application. The following section presents a very high-level overview of the main architectural components.

Our mHealthSwarm platform architecture is outlined in Figure 2. It is divided into four components, which are briefly described below.

Application Manager: The Application Manager combines the view and logic layers and can be considered as a task manager that keeps track of active micro-apps and provides an interface to the container (i.e., our micro-mHealth app platform) APIs.

Data Manager: A gateway for all health data produced and consumed by hosted micro-apps, along with raw data collected from health peripherals. Although this component currently only supports local data storage, future extensions are planned to support remote storage and management.

Device Manager: The component is responsible for handling sensors connected to the smartphone and provides an interface for collecting raw data produced by these sensors.

Conversation Manager: Although our platform could reduce the need for several health apps, users may still have to navigate through the micro-apps. The ecosystem of conversational interfaces have
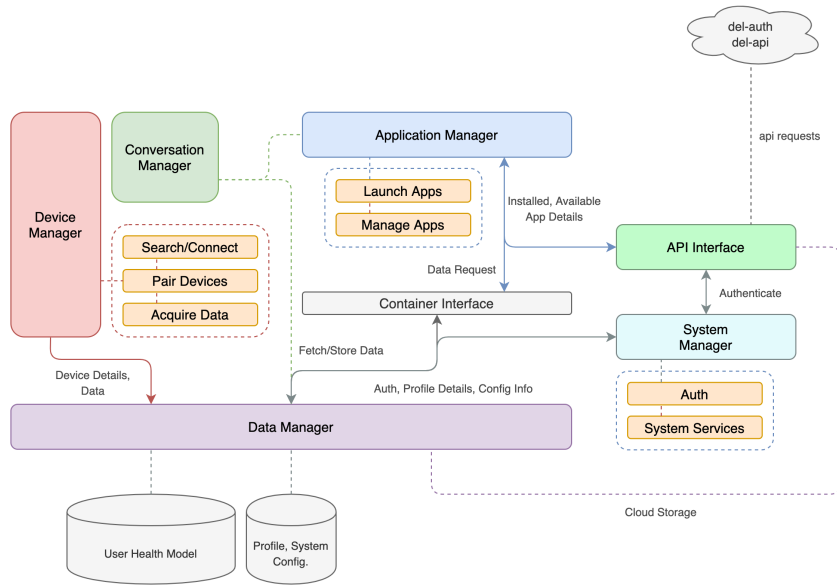
Figure 2: Architecture overview of our platform

been expanding rapidly[7] and studies around their use have also suggested their usefulness in performing simple tasks such as scheduling appointments (Palanica, Flaschner, Thommandram, Li, and Fossat, 2019). Given the availability of numerous health metrics on our platform, we believe it is a suitable environment for a chatbot to help users interact with.

Our current prototype aims to provide end users a better experience by providing complete customizability of functions through micro-health-apps. It is implemented as an Android application, with the supporting framework for micro-mHealth apps implemented as a back end service using Node.js. The platform is designed to facilitate access to device hardware such as Bluetooth and GPS modules along with other embedded sensors (IMUs, camera) through abstract functions. However, given the very limited use of external sensors by currently available apps (B. J. Philip, Abdelrazek, Bonti, Barnett, and Grundy, 2022), the current implementation has limited support for peripheral devices, with plans for supporting more in the future. The micro-mHealth apps do not have any control over the device hardware, but are only passed the data from these sensors via the platform.

Data storage currently uses the Room Persistence Library for predefined data types, with additional support for app-specific storage in separate files. The platform interface allows all registered micro-mHealth apps with the appropriate permissions

to fetch and store the required data. Data transfer between the app and the remote service uses HTTP methods with the data wrapped in JSON objects.

Our mHealthSwarm platform provides a set of mini apps, which form the basis of our micro-mHealth apps. These are structured like web applications, which reduces the learning curve required for developing mHealth apps, while also making them platform-independent. Interaction with the platform is handled through exposed APIs called from the logic layer, a brief example of which is presented in the next section.

While native container apps for different mobile operating systems will have to be developed, they offer the benefit of cross-platform support for the micro-mHealth apps they host. To 'install' services, users will just need to link those micro-mHealth apps available on the platform to their accounts, during which permissions will be established. Launching a micro-mHealth app will launch the feature droplet in the container, which then handles app requests and provides data to the running micro-mHealth app after validating permissions. The overall flow is outlined in brief in figure 3.

4 USAGE EXAMPLE

Consider an app such as MapMyWalk[8] with the core feature of tracking walks. A micro-mHealth app can

---

[7]https://www.businessinsider.com/chatbot
-market-stats-trends?r=AU\&IR=T

[8]https://play.google.com/store/apps/
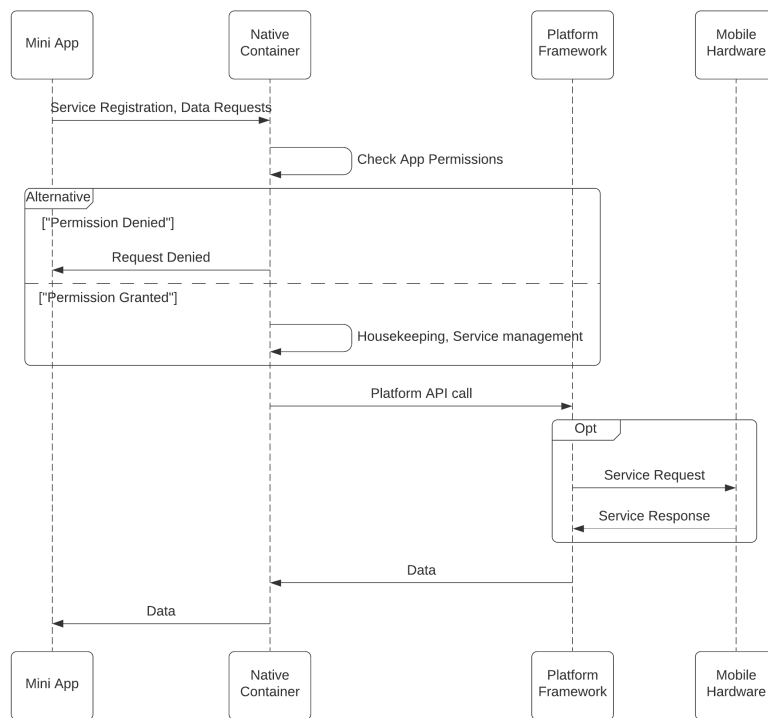details?id=com.mapmywalk.android2

Figure 3: An overview of our micro-application container data-flow.

be built with the same core feature, eliminating additional, unnecessary functionality, allowing users to choose the *app functionality* they need. To demonstrate this, we developed a micro-mHealth app on our platform where the service keeps track of a user's location, step count and heart rate (if available) during a workout session and allows them to review the data later. The following code snippet from the JavaScript logic layer outlines a very basic request for the user's location. It shows a call to the platform API for logging the GPS coordinates in the database and setting a callback reference to the function that handles the location data. This data is then used by the micro-mHealth app to provide the walk tracking feature. Figure 5 (left) shows the micro app implementing this feature, and figure 4 shows the overall control and data flow for this app after it is launched.

```
1   function appInit() {
2     let appRequest = {
3       // injected on launch
4       "appId" : this.appId,
5       "requests" : [{
6           "resource" : "location",
7           "callback" : "processLoc",
8           "toggle": true
9       }]
10    };
```

```
11
12    // Log data in the database
13    DelUtils.setSensorLoggerRequest(
14      JSON.stringify(appRequest));
15
16    // Register callback
17    DelUtils.setCallbackRequest(
18      JSON.stringify(appRequest));
19  }
20
21  // Registered callback for GPS data
22  function processLoc(dataType, loc) {
23    if ("location" == dataType) {
24      // process location data
25    }
26  }
```

Listing 1: App data request registration example

The sample application is built using JavaScript, HTML and CSS with the OnsenUI framework[9] for creating a native-like interface. Although the core business logic for the micro-mHealth apps would require a similar effort as developing native code for similar features, the benefit of this approach is apparent when considering the single-function nature of these app droplets and the overall simplification of access to device resources and shared user data.

The novelty of the platform for mHealth comes
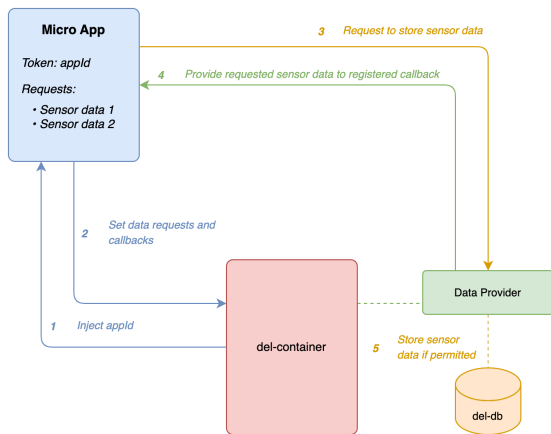
---

[9]https://onsen.io

Figure 4: Main interaction between the platform and a micro-app after launch
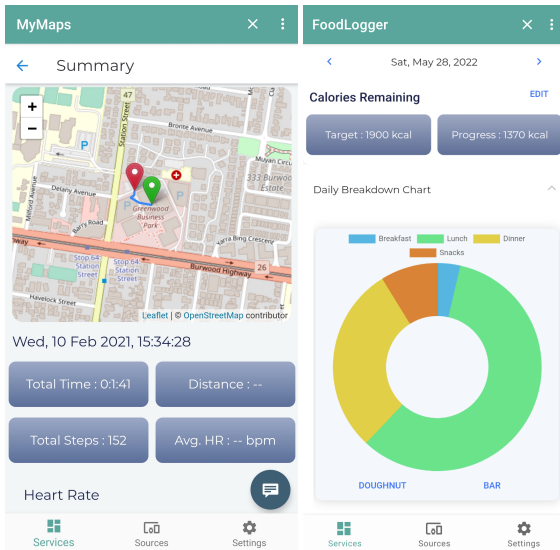

Figure 5: Micro-app implementations of a walk tracker (left) and a meal tracker (right) on our platform prototype

from a highly customizable environment where each micro-mHealth app provides a single function and can be added/removed as required, enabling end users to create personalized health apps. To evaluate our platform and its feature customizability with end users, we implemented core functions (i.e., workout tracking, meal tracking, and mood tracking) inspired from 3 popular apps. Our evaluations are still ongoing, and preliminary results are discussed in the next section. To further enhance the customizability, we also introduced semi-customizable charts for different use-cases such as tracking one's meals (figure 5 right), and while the current implementation is very limited, further extensions to visualization components are planned to extend the range of personalization on the platform.

This design is expected to remove unnecessary development overheads from developers, who no longer have to worry about low level details such as device integration and data management. This may also open up the domain to a wider developer community in addition to mobile app developers, allowing for more innovation in the domain. The hybrid app design is also expected to simplify updates as a verified service update will be made available to all users on release without the need for manual search and installation.

5 DISCUSSION

Although the Android and iOS operating systems offer native support for health data aggregation with frameworks such as Google Fit and Apple Health, they still suffer from a common issue - the reliance on more than one health app. However, given the popularity of hybrid applications and wide commercial success of mini-app frameworks for messaging and eCommerce (e.g., WeChat[10]), they appear to be a suitable alternative to native services. This approach offers the benefits of hybrid applications while promoting an open, customizable platform of mHealth services. Despite the advantages, this customizability comes with some drawbacks, such as an added overhead for learning the platform functionality and slightly lower micro-app performance. However, we believe that these are minor issues considering the overall benefits the platform offers.

In terms of functionality, our mHealthSwarm platform is currently limited to an external heart rate monitor, built-in sensors, GPS and camera. The platform is also currently limited to Android, with a future expansion planned for supporting iOS. We are currently also exploring ways to improve data visualization in micro-mHealth apps by using platform-provided visualization components. This is expected to not only simplify the development aspect, but also in effect provide more flexibility to end-users. Similarly, given the highly specific and limited nature of the platform and its capabilities, a domain-specific language for developing micro-mHealth apps is also being considered as a future extension to further simplify app development by generating micro-mHealth app components.

Our mHealthSwarm platform is currently being evaluated with end-users as well as mobile app developers to get their feedback. Preliminary user evaluations have indicated a preference for this approach, with the participants stating they like the feature cus-

---

[10]https://www.wechat.com/

tomizability offered by the platform. This is highlighted by one participant's comment - *"What I definitely like about the platform is the customization - that I can select the features that I use"*. So far, almost all users have indicated that the platform was easy to learn, with emerging themes from participant interviews highlighting the ease of use, the benefits of feature customization, and the ease of data management on a single platform. Similarly, ongoing developer evaluations have also seen mostly positive feedback, with the main highlight being the reduced development effort. On the topic of ease of use, a developer's comment - *"...this seems to be a bit more like web development, so yeah [it's] quite simple... with less boilerplate code, compared to regular mobile apps"* - highlights the potential benefits of such unified platforms for app developers. However, major concerns such as the limitations of the platform in its current state (including limited support for testing and debugging micro-mHealth apps, and limited documentation) when compared to current mobile app development frameworks were also highlighted.

Overall, our work in this domain focuses on providing a better framework and platform for micro-mHealth apps as a means to achieve the following goals: (1) To provide a consistent platform for better collaboration and communication between the different apps; (2) To foster an ecosystem of micro-mHealth apps built for specific functions, eliminating unnecessary components and bloat; (3) To reduce the increasing device resource requirements and contention of several large health apps on one device; and (4) To provide a better user experience and increase mHealth adoption.

6 SUMMARY

In this paper, we presented our mHeallthSwarm platform of a micro-mHealth platform that addresses major challenges with current mHealth apps such as the need to install more than one app, unnecessary bloat in the applications and challenges with data sharing and management. We discuss implementation details of our platform with a brief data flow sequence diagram showing a high-level flow of app requests and data. We then briefly discuss a sample app and how it can be developed using the platform APIs. Overall, our approach for developing micro-mHealth apps will guide the design and development of novel mHealth apps and platforms, and in the future have a positive impact on the adoption of mHealth services.

REFERENCES

Alshehhi, Y., Abdelrazek, M., and Bonti, A. (2022). Analysis of personal data visualisation reviews on mobile health apps. In *Achi 2022, the fifteenth international conference on advances in computer-human interactions* (pp. 111–118).

Alshehhi, Y. A., Abdelrazek, M., and Bonti, A. (2022, sep). Needs and Challenges of Personal Data Visualisations in Mobile Health Apps: User Survey. Retrieved from `http://arxiv.org/abs/2209.00838`

Anastasiadou, D., Folkvord, F., Serrano-Troncoso, E., and Lupiañez-Villanueva, F. (2019, may). Mobile Health Adoption in Mental Health: User Experience of a Mobile Health App for Patients With an Eating Disorder. *JMIR mHealth and uHealth*, 7(6), e12920. doi: 10.2196/12920

Banos, O., Villalonga, C., Garcia, R., Saez, A., Damas, M., Holgado-Terriza, J. A., ... Rojas, I. (2015). Design, implementation and validation of a novel open framework for agile development of mobile health applications. *BioMedical Engineering OnLine*, 14(2), S6. doi: 10.1186/1475-925X-14-S2-S6

Barnett, S., Vasa, R., and Grundy, J. (2015). Bootstrapping mobile app development. In S. Elbaum and G. Canfora (Eds.), *Proceedings - 2015 ieee/acm 37th ieee international conference on software engineering, icse 2015* (Vol. 2, pp. 657–660). United States of America: IEEE, Institute of Electrical and Electronics Engineers. doi: 10.1109/ICSE.2015.216

Berauk, V., Murugiah, M., Soh, Y., Sheng, Y., Wong, T., and Ming, L. C. (2017). Mobile Health Applications for Caring of Older People: Review and Comparison. *Therapeutic Innovation & Regulatory Science*, 216847901772555. doi: 10.1177/2168479017725556

Gu, J., Mackin, S., and Zheng, Y. (2018). Making sense: an innovative data visualization application utilized via mobile platform. In *2018 ieee 20th international conference on high performance computing and communica-*

tions; ieee 16th international conference on smart city; ieee 4th international conference on data science and systems (hpcc/smartcity/dss) (pp. 1105–1109).

Higgins, J. (2015). Smartphone Applications for Patients' Health & Fitness. *The American journal of medicine*. doi: 10.1016/j.amjmed.2015 .05.038

Inupakutika, D., Kaghyan, S., Akopian, D., Chalela, P., and Ramirez, A. G. (2020, may). Facilitating the development of cross-platform mHealth applications for chronic supportive care and a case study. *Journal of Biomedical Informatics*, *105*, 103420. doi: 10.1016/j.jbi.2020.103420

Jena, A., Butler, M., Dwyer, T., Ellis, K., Engelke, U., Kirkham, R., ... Rajamanickam, V. (2021). The next billion users of visualization. *IEEE Computer Graphics and Applications*, *41*(2), 8–16.

Katz, D., Dalton, N., Holland, S., O'kane, A., and Price, B. A. (2017). Questioning the reflection paradigm for diabetes mobile apps. In *ehealth 360°* (pp. 315–326). Springer.

Krebs, P., and Duncan, D. T. (2015, Nov 04). Health app use among us mobile phone owners: A national survey. *JMIR mHealth uHealth*, *3*(4), e101. Retrieved from http://mhealth.jmir .org/2015/4/e101/ doi: 10.2196/mhealth .4924

Lee, B., Choe, E. K., Isenberg, P., Marriott, K., and Stasko, J. (2020). Reaching broader audiences with data visualization. *IEEE Computer Graphics and Applications*, *40*(2), 82–90.

Liu, Y., Xu, E., Ma, Y., and Liu, X. (2019, jul). A First Look at Instant Service Consumption with Quick Apps on Mobile Devices. In *2019 ieee international conference on web services (icws)* (pp. 328–335). IEEE. doi: 10.1109/ ICWS.2019.00061

Liu, Z.-Y., Liao, K., and Wang, J.-L. (2020). Research on the influence of we-chat applet on improving user experience. In *Proceedings of the korean society of computer information conference* (pp. 613–615).

MiniApp Standardization White Paper, W3C First Public Working Draft. (2019). Retrieved from https://www.w3.org/TR/mini -app-white-paper/

Palanica, A., Flaschner, P., Thommandram, A., Li, M., and Fossat, Y. (2019). Physicians' Perceptions of Chatbots in Health Care: Cross-Sectional Web-Based Survey. *J Med Internet Res*, *21*(4), e12887. doi: 10.2196/12887

Philip, B., Abdelrazek, M., Barnett, S., Bonti, A., and Grundy, J. (2022). Towards Better mHealth Apps: Understanding Current Challenges and User Expectations. In *2022 ieee/acm 9th international conference on mobile software engineering and systems (mobilesoft)* (pp. 33–37). doi: 10.1145/3524613.3527804

Philip, B. J., Abdelrazek, M., Bonti, A., Barnett, S., and Grundy, J. (2022). Data Collection Mechanisms in Health and Wellness Apps:Review and Analysis. *JMIR mHealth and uHealth*, *10*(3). doi: 10.2196/30468

Pinto, C. M., and Coutinho, C. (2018). From Native to Cross-platform Hybrid Development. In *2018 international conference on intelligent systems (is)* (pp. 669–676). doi: 10.1109/IS .2018.8710545

Serrano, N., Hernantes, J., and Gallardo, G. (2013). Mobile web apps. *IEEE Software*, *30*(5), 22-27.

Sezgin, E., Özkan-Yildirim, S., and Yildirim, S. (2017). Investigation of physicians' awareness and use of mHealth apps: A mixed method study. *Health Policy and Technology*(3), 251–267. doi: https://doi.org/10.1016/j.hlpt.2017.07 .007

van Velsen, L., Beaujean, D. J., and van Gemert-Pijnen, J. E. (2013, dec). Why mobile health app overload drives us crazy, and how to restore the sanity. *BMC Medical Informatics and Decision Making*, *13*(1), 23. doi: 10.1186/ 1472-6947-13-23

Ventola, C. L. (2014, may). Mobile devices and apps for health care professionals: uses and benefits. *P & T : a peer-reviewed journal for formulary management*(5), 356–364.

WHO. (2011). *mHealth: New horizons for health through mobile technologies: second global survey on eHealth* (Vol. 3). World Health Organization. Retrieved from https://www.who.int/goe/ publications/goe_mhealth_web.pdf

Wiechmann, W., Kwan, D., Bokarius, A., and Toohey, S. L. (2016). There's an App for That? Highlighting the Difficulty in Finding Clinically Relevant Smartphone Applications. *The western journal of emergency medicine*(2), 191–194. doi: 10.5811/westjem.2015.12.28781