Poster: Challenges & Solutions in a Hybrid mHealth Mobile App

Yatharth Ranjan
yatharth.ranjan@kcl.ac.uk
Institute of Psychiatry Psychology & Neuroscience
King's College London, UK

Amos A Folarin
amos.folarin@kcl.ac.uk
Institute of Psychiatry Psychology & Neuroscience
King's College London, UK

Callum Stewart
callum.stewart@kcl.ac.uk
Institute of Psychiatry Psychology & Neuroscience
King's College London, UK

Pauline Conde
pauline.conde@kcl.ac.uk
Institute of Psychiatry Psychology & Neuroscience
King's College London, UK

Richard Dobson
richard.j.dobson@kcl.ac.uk
Institute of Psychiatry Psychology & Neuroscience
King's College London, UK

Zulqarnain Rashid
zulqarnain.rashid@kcl.ac.uk
Institute of Psychiatry Psychology & Neuroscience
King's College London, UK

ABSTRACT

The paper describes the various problems and challenges encountered during the development and remote data collection in a cross-platform hybrid application developed for remote monitoring of participants and what solutions were implemented to mitigate them. These problems and challenges are universal for hybrid applications and this paper digs deep into these in the domain of large-scale, long-duration mHealth research studies. From technical issues to issues with user compliance, this paper discusses the core problems inherent to these types of studies and technologies, and how to mitigate them.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing systems and tools**; • **Computer systems organization** → *Maintainability and maintenance.*

KEYWORDS

mHealth, mobile context sensing, wearable sensors, data collection platform, cross platform mobile application, ionic, cordova, firebase, push notification, hybrid application, compliance

ACM Reference Format:
Yatharth Ranjan, Amos A Folarin, Callum Stewart, Pauline Conde, Richard Dobson, and Zulqarnain Rashid. 2019. Poster: Challenges & Solutions in a Hybrid mHealth Mobile App. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2019 International Symposium on Wearable Computers (UbiComp/ISWC '19 Adjunct), September 9–13, 2019, London, United Kingdom.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3341162.3343778

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
UbiComp/ISWC '19 Adjunct, September 9-13, 2019, London, United Kingdom
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6869-8/19/09…$15.00
https://doi.org/10.1145/3341162.3343778

1 INTRODUCTION

The paper discusses the challenges and solutions for the active remote monitoring component of the RADAR-base platform [24][23]. The RADAR-base platform is a highly scalable and flexible open source mHealth platform for remote assessment of participants. It has in-built functionality to monitor passive high-frequency raw sensor and wearable data alongside the active data (e.g. questionnaires, digital tests) for various disorders. The aRMT (active Remote Monitoring Technology)[9][8] in RADAR-base is the questionnaire application used to collect questionnaires, clinical tasks, active audio and any other measurements requiring user interaction. The aRMT was built with the cross-platform hybrid Ionic framework[16] and integrated with components from the RADAR-base's backend platform. A *hybrid application* is a cross platfrom application which is basically a web page (writtern in HTML, CSS and Javascript) which runs in the device's native browser and interacts with the underlying native device's components (like camera) using a plugin based architecture(Cordova).

1.1 Organisation

Each of the following sections of the paper focus on an issue area of the aforementioned application. The initial two sections discuss technical problems faced with notifications delivery primarily due to longitudinal study and how they were solved. The last section discusses the issue with data collection due to low compliance and how it was mitigated with improved user monitoring and event metrics insight.

2 LOCAL NOTIFICATIONS

Local Notifications[17] was the initial preferred choice as Ionic supports them out of the box and they don't require a network connection (which was favourable as some of the questionnaires had short notification delivery windows (10 minutes)).

2.1 Challenges

One of the major problems encountered with the app was with the Notifications, sent to inform participants that some task was due. Problems arose due in part to the long term data collection periods (e.g. up-to 2 years) of the projects supported by RADAR-base which meant scheduling large amount of notification for the app in advance (about 1100).

Some vendors put a hard limit (e.g. 500 for Samsung) on the number of notifications (in the AlarmManager[2]) that can be scheduled from one app. The same issue has been reported on Stack Overflow(SO)[12] and still persists as reported in this recent issue[13]. Due to the long term study and the amount of notifications to be scheduled, the notifications from the app were blocked by the OS after a period of time. This can be solved if the application is a native android application (as evident from this SO answer[3]) but there is no straight forward way when using Cordova plugins (without a low level re-working of the plugin code, and even then, it may not be possible or may not work as expected).

"The main disadvantage of interpreted application is that the development is dependent on the feature set provided by the selected framework"[19] And that feature set may not be rich or evolve at the same pace as the underlying operating systems. Furthermore, Because of the variety of devices, hardware, vendors and custom software in the Android ecosystem, each of the device can behave differently even if the version of Android is same, especially when using a hybrid application instead of a natively built application.

The Cordova Local Notification plugin[17] was not up to date and during which Android had made some drastic changes to their notification architecture. Most of the problems associated with the plugin on Android 8.x have already been heavily discussed on the official Github repo issues[18], some of which are closed and some still open. Most of the early issues involved not getting notifications when the app was closed or running in the background, not receiving notifications when device was restarted or issues relating to new notification channels introduced in Android 8.0 Oreo.

2.2 Solutions

Various different tweaks and hacks to work-around these problems were tried[22] but although better, they were still not working as intended. The final solution was to discard the entire local notification system from the application and use Push Notifications to make the notification delivery independent of the application, device and software. Push notifications were chosen to replace local notifications as they were industry standard for reliability. Push notifications rely on a server to send notifications to the device.

3 PUSH NOTIFICATIONS

3.1 Challenges

We decided to use Firebase CloudMessaging (FCM)[6] with XMPP protocol[14] as the underlying service for notification delivery and our own XMPP client application server acting as a scheduler for notifications but there were a few caveats with this approach-

- The already available Cordova plugin for FCM[15] did not implement the upstream functionality present[11] in the native FCM SDK (Software Development Kit) for Android and IOS and we needed this to send messages to the server from the app for scheduling notifications in an easy and reliable way.
- There was no proper server side java library for FCM XMPP protocol (so it can consume upstream messages).
- Push notifications need an internet connection on the device to be delivered.

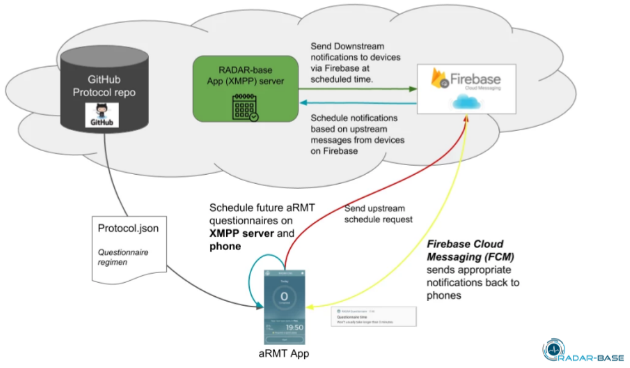
GitHub Protocol repo
GitHub
RADAR-base App (XMPP) server
Send Downstream notifications to devices via Firebase at scheduled time.
Firebase
Cloud Messaging
Schedule notifications based on upstream messages from devices on Firebase
Protocol.json
Questionnaire regimen
Schedule future aRMT questionnaires on XMPP server and phone
Send upstream schedule request
Firebase Cloud Messaging (FCM) sends appropriate notifications back to phones
aRMT App
RADAR-BASE

Figure 1: Push notification architecture.

- There are two types of messages[1] that can be used to deliver notifications using FCM and the challenge was to decide which type of message to use. *Data messages* are handed over to the app to be handled. If we sent a notification this way, FCM will hand over all the content in the data payload to the app and then the app will handle the display of the notification. *Notification Messages* on the other hand are automatically displayed as a notification on the device by FCM without any callback to the application.

3.2 Solutions

The solutions for the above problems although in appearance not trivial were comparatively straight forward to implement.

- The Cordova plugin for FCM was extended and added the upstream functionality[20].This was easier than fixing the local notification plugin because it just had to call a function of the underlying FCM SDK. This also meant that no problems with evolution as one can just increment the version of FCM SDK used in the plugin.
- Fortunately, there was a very basic implementation[26] of FCM XMPP protocol on GitHub. We forked the repository and added multiple extensions to it to cater to our needs. These improvements are listed on the README of our implementation of the FCM XMPP server[5]. Figure 1 shows the architecture of the notification server.
- The stats show that only a small amount of notifications were not received principally as a result of network connectivity, currently 10% though we expect this to decrease as connectivity improves in the future(see Figure 2). Hence, the pros outweigh the cons by a fairly large margin.
- Notification Messages were chosen as a mode of sending notifications as we wanted to minimise any dependency on Cordova for the new notification system. Also to avoid any complications in handling incoming messages if the app was not allowed to run in the background.

Centralising logic on the server means more control and insight into the notification system remotely than you can on individual devices in the case of local notifications. This makes debugging easier and provides individual level metrics. We can know if an

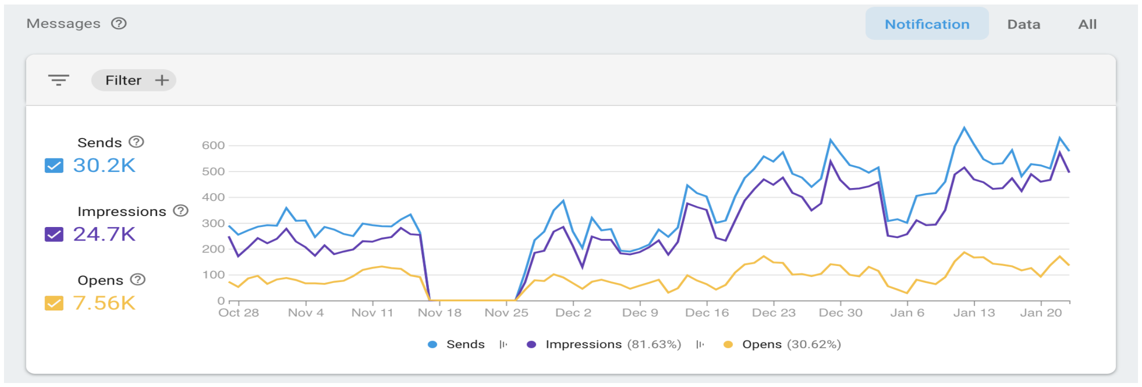

Figure 2: Push notification stats.

individual notification was delivered along with aggregate stats in the Firebase dashboard (Fig [2]). The system is also independent of the hardware and the software of the device. It will work as the components evolve and ensures backwards and future compatibility. This is important for hybrid apps as the underlying frameworks and plugins don't evolve as quickly as the operating system does.

4 LOW COMPLIANCE

4.1 Challenges

While the first iteration of remote notifications was a major step forward, some problems emerged when the above system was running for sometime in production. The major issue was, as you can be seen in Figure [2], that user compliance (% of opened notifications) was low and the project admins needed an individual level insight on which notifications/users are delivered/active so they can intervene and improve the compliance.

We particularly faced the challenging issue of low compliance in the RADAR-CNS multi-disorder study[27][25]. We now set about establishing reasons for the low compliance. We considered if this was:

- due to a technical issue or data leak. Although the components are tested we cannot rule the existence of race conditions and edge cases in such a massive microservices pipeline.
- due to low participant compliance and low motivation to be part of the studies which make sense as there were little or no incentives and value the participants were getting out of it (except for contributing to research)

It was hard to debug this with the limited insight that was available. The system created in the previous section was not flexible. It was great at only one task, i.e. scheduling and sending notifications. The above platform was hard to evolve and it did not integrate well with the management platform of RADAR-base platform and did not provide insight on anything else. The FCM XMPP notification server has 30 days worth of logs (which can be increased) which contain delivery information but it was hard to get, parse and interpret that data. Also it will be very hard to deliver this info in a secure, authorised manner to project admins without integrating this with the management user-interface of RADAR-base, The Management

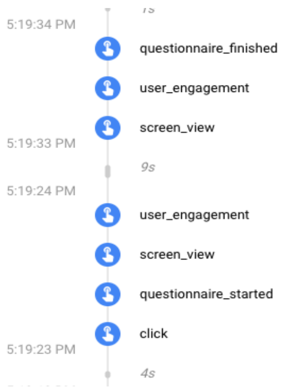
```
5:19:34 PM
        questionnaire_finished
        user_engagement
        screen_view
5:19:33 PM
        9s
5:19:24 PM
        user_engagement
        screen_view
        questionnaire_started
        click
5:19:23 PM
        4s
```

Figure 3: A snapshot of Event stream

Portal[4]. Lastly, this does not guarantee that the notification is displayed, it just denotes that the notification reached the device but individual device level settings may not allow the notification to be displayed (e.g. Do not disturb is on).

4.2 Solutions

To counter the challenges faced above, two solutions were designed,

- *App Server*: a new General Purpose Application server[21] which will serve as more than just a notification scheduler was proposed. Irrespective of the internal working of the server, it exposes standardised REST[10] endpoints to deliver messages/notifications and other functionality. This allows other systems to also use these endpoints to utilise the same mechanisms and hence allows the system to be reactive. For example, a machine learning algorithm may alter the timing of delivery of notifications for better compliance, based on some real-time criteria. It uses the same standards and security mechanisms that are already a part of RADAR-base platform. Easy to extend, so we can evolve the App Server and its database as required. The app server also provides pre-built libraries for XMPP and FCM admin SDK support out of the box, so can be used by the community in future projects. Future work involves simplifying the use

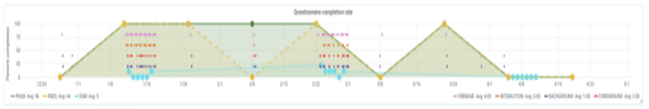

Figure 4: Questionnaire Schedules and corresponding Notification Delivery (Firebase) and Phone User Interaction Events for a participant. The pink points are Delivery receipts from FCM. The orange, red and blue points are user and app interaction. The green, yellow and blue points are different types of scheduled questionnaires. The y-axis denotes the percentage completion of each questionnaire and x-axis is a timeline.

of provided libraries, providing user interfaces for various different tasks and integrating other components in RADAR-base ecosystem to deliver just-in-time information through the App Server.

- *Event Analytics*: Added Firebase analytics[7] with a combination of user properties into the app so as to get insight into the low compliance. The value of these insights cannot easily be understated. This provide event-by-event data and metrics (see Figure [3]) at the individual user level, at the study level and at the global level. Because of increased user-app complexity, we needed granular insight into how it was being used. Firebase Metrics helps investigate different compliance issues encountered by providing delivery and sent stats for each notification. The events are reported by the application to the very reliable and tested Firebase platform (by Google)[6]. These can also be exported to google BigQuery for further analysis on how users are interacting with the app. This provides powerful visualizations (Figure [4]) in data dashboard and helps analyse the workflow and contact appropriate participants e.g. case they are not compliant for long periods or frequently dismiss notifications.

Future work in this area involves running engagement campaigns with the help of the Firebase platform described above. For example, sending notifications to users based on predictions (e.g. groups predicted to dismiss more than 50% of notifications in the coming period of questionnaires, send them a notifications to prompt completing the upcoming questionnaire period), notifying users who have not interacted with the app within a certain threshold of time (eg - 30 days). Other ways of improving compliance like changing the look and feel of notifications for different types of questionnaires (since some of the questionnaires are too long and may decrease motivation for users).

ACKNOWLEDGMENTS

This work has received support from the EU/EFPIA Innovative Medicines Initiative Joint Undertaking RADAR-CNS (grant No 115902) (www.imi.europa.eu). The Authors receive funding support from the National Institute for Health Research (NIHR) Biomedical Research Centre at South London and Maudsley NHS Foundation Trust and King's College London.

Disclaimer. This communication reflects the views of the RADAR-CNS consortium and neither IMI nor the European Union and EFPIA are liable for any use that may be made of the information contained herein.

REFERENCES

[1] [n. d.]. About FCM messages. https://firebase.google.com/docs/cloud-messaging/concept-options
[2] [n. d.]. AlarmManager. https://developer.android.com/reference/android/app/AlarmManager
[3] [n. d.]. Android Native Solution. https://stackoverflow.com/questions/29344971/java-lang-securityexception-too-many-alarms-500-registered-from-pid-10790-u/29610474
[4] [n. d.]. Concepts in Management Portal. https://radar-base.org/index.php/2019/02/13/concepts-in-management-portal/
[5] [n. d.]. Extended XMPP Connection Server for FCM. https://github.com/RADAR-base/fcmxmppserverv2 original-date: 2018-07-25T11:47:45Z.
[6] [n. d.]. Firebase Cloud Messaging. https://firebase.google.com/docs/cloud-messaging
[7] [n. d.]. Google Analytics for Firebase. https://firebase.google.com/docs/analytics
[8] [n. d.]. Questionnaire App. https://radar-base.org/index.php/questionnaire-app/
[9] [n. d.]. Questionnaire mobile application for RADAR-base. https://github.com/RADAR-base/RADAR-Questionnaire original-date: 2016-09-15T13:24:47Z.
[10] [n. d.]. Representational state transfer. https://en.wikipedia.org/w/index.php?title=Representational_state_transfer&oldid=897921704 Page Version ID: 897921704.
[11] [n. d.]. Sending upstream messages on Android | Firebase. https://firebase.google.com/docs/cloud-messaging/android/upstream
[12] [n. d.]. Too many alarms (500) registered issue. https://stackoverflow.com/questions/29344971/java-lang-securityexception-too-many-alarms-500-registered-from-pid-10790-u
[13] [n. d.]. Too many alarms registered from uid. https://github.com/katzer/cordova-plugin-local-notifications/issues/1753
[14] [n. d.]. XMPP. https://en.wikipedia.org/w/index.php?title=XMPP&oldid=892119909 Page Version ID: 892119909.
[15] Ionic. [n. d.]. FCM plugin Ionic. https://ionicframework.com/docs/native/fcm
[16] Ionic. [n. d.]. Ionic Documentation. https://ionicframework.com/docs/
[17] SebastiÃąn Katzer. [n. d.]. Cordova Local-Notification Plugin. https://github.com/katzer/cordova-plugin-local-notifications original-date: 2013-08-10T11:25:59Z.
[18] SebastiÃąn Katzer. [n. d.]. Cordova Local-Notification Plugin Issues. https://github.com/katzer/cordova-plugin-local-notifications/issues original-date: 2013-08-10T11:25:59Z.
[19] C. P. Rahul Raj and Seshu Babu Tolety. [n. d.]. A study on approaches to build cross-platform mobile applications and criteria to select appropriate approach. In *2012 Annual IEEE India Conference (INDICON)* (2012-12). 625–629. https://doi.org/10.1109/INDCON.2012.6420693
[20] Yatharth Ranjan. [n. d.]. Extended FCM Cordova Plugin. https://github.com/yatharthranjan/cordova-plugin-fcm original-date: 2018-07-19T15:01:17Z.
[21] Yatharth Ranjan. [n. d.]. General purpose application server for the radar platform. https://github.com/RADAR-base/RADAR-Appserver original-date: 2018-11-06T16:22:38Z.
[22] Yatharth Ranjan. [n. d.]. RADAR App Server. https://radar-base.org/index.php/2019/04/09/radar-app-server/
[23] Yatharth Ranjan, Maximilian Kerz, Zulqarnain Rashid, Sebastian BÃűttcher, Richard J.B Dobson, and Amos A. Folarin. [n. d.]. RADAR-base: A Novel Open Source m-Health Platform. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers* (2018) *(UbiComp '18)*. ACM, 223–226. https://doi.org/10.1145/3267305.3267579 event-place: Singapore, Singapore.
[24] Yatharth Ranjan, Zulqarnain Rashid, Callum Stewart, Maximilian Kerz, Denny Verbeeck, Sebastian Boettcher, Pauline Conde, Richard Dobson, and Amos Folarin. [n. d.]. RADAR-base: An Open Source mHealth Platform for Collecting, Monitoring and Analyzing Data Using Sensors, Wearables, and Mobile Devices. ([n. d.]), 24. https://doi.org/10.2196/11734
[25] Zulqarnain Rashid, Callum L. Stewart, Sebastian BÃűttcher, Yatharth Ranjan, Richard J.B Dobson, and Amos A. Folarin. [n. d.]. RADAR-base: Epilepsy Case Study. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers* (2018) *(UbiComp '18)*. ACM, 227–230. https://doi.org/10.1145/3267305.3267578 event-place: Singapore, Singapore.
[26] Carlos Becerra Rodriguez. [n. d.]. XMPP Connection Server for FCM. https://github.com/carlosCharz/fcmxmppserverv2 original-date: 2016-10-21T20:04:33Z.
[27] Callum L. Stewart, Zulqarnain Rashid, Yatharth Ranjan, Shaoxiong Sun, Richard J.B. Dobson, and Amos A. Folarin. [n. d.]. RADAR-base: Major Depressive Disorder and Epilepsy Case Studies. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers* (2018) *(UbiComp '18)*. ACM, 1735–1743. https://doi.org/10.1145/3267305.3267540 event-place: Singapore.

View publication stats