

Mobile Aid for Haemophilia Injury Recognition

Richard K. Lomotey^{*1}, Kathy Mulder², JoAnn Nilson³, Candice Schachter⁴, Kristy Wittmeier⁵, Ralph Deters⁶

^{1,6}Department of Computer Science, University of Saskatchewan, S7N 5C9, Saskatoon, Canada

²Children's Hospital, Health Sciences Centre, Winnipeg, Canada

³Saskatchewan Bleeding Disorder Program, Royal University, Hospital, Saskatoon, Canada

⁴School of Physical Therapy, University of Saskatchewan, Saskatoon, Canada

⁵Department of Paediatrics and Child Health, University of Manitoba, Manitoba, Canada

^{*1}richard.lomotey@usask.ca; ⁶deters@cs.usask.ca

Abstract-Haemophilia is a bleeding condition that prevents blood clot during injuries. Previous studies suggest that some persons with the condition often do not report the problem and in some cases, there is the need to take certain critical actions when injuries occur. This has created the need to investigate best approaches that can encourage people with the condition to seek medical help or in the case where the injury is minor, users can assess the injury themselves by following some guidelines. To this effect, our ongoing project seeks to explore the issue through the adoption of mobile computing technology. Thus, an App is developed called, Haemophilia Injury Recognition Tool (HIRT?), which provides the users (especially young men) with a set of instructions on how to assess minor injuries. In cases where the injury is severe, the user is asked to go to the emergency room or contact the nearest Haemophilia Treatment Center which is provided in the App. The paper discusses the architectural flow of the App as well as the user feedback. Overall, the initial group of people who tested the App found it very useful in terms of its content, reminders, and clarity of steps.

Keywords- Mobile Devices; Haemophilia; Proxy; Cloud Computing; Distributed System; NoSQL Database; Injury Assessment

I. INTRODUCTION

People with Haemophilia are persons (mostly men) who have a blood disorder that prevents blood clotting during injury. The condition can be classified into varying degrees depending on the blood clotting factor such as mild, moderate, or severe.

Across Canada, physiotherapists had seen situations where young men with mild haemophilia waited too long to come in for treatment after an injury. The clinicians and the physiotherapists (i.e., the comprehensive care colleagues) had also noticed this. The question therefore is why this was happening? During interactions with some physiotherapists, there were suggestions for the provision of new booklets to people with mild haemophilia; but, we wondered if that was the best approach. In order to arrive at the best possible solution to provide quality information to the young men with the condition, we followed the following steps.

A. Step 1: Understanding the Issue

In the initial work by Nilson et al. [1], young Canadian men (18-30 years old) with mild haemophilia were interviewed to learn more about their knowledge, attitudes and behaviours. Meetings were also held with the focus groups (i.e., physiotherapists) across Canada. This aided the research group to study what the underlying issues were.

B. Step 2: What are the Issues

From the study of the evidence in step 1, we realised that paper-based booklets will not just be used by the target group but they may not be considered relevant.

C. Step 3: Decision Pathway

Several workflows are developed based on the interviews with the young men with mild hemophilia.

D. Step 4: The Results

Clearly, there was a need to adopt a more modern approach to aid the young men with mild hemophilia to assess minor injuries when they occur.

E. The Way Forward

The answer to the final question in step 4 carries a lot of weight. It is one thing developing a mobile application but it is another thing building the application to provide a better option than a booklet. The mobile app must be informative enough to encourage the young men with hemophilia to report the injury to the physiotherapists. Besides, the mobile specific challenges such as intermittent loss of connectivity in distributed architectures must be overcome. Thus, collaboration with computer scientists at the University of Saskatchewan is formed with members of the Canadian Physiotherapists in Haemophilia Care (CPHC) to help develop an electronic and convenient mobile app. This collaboration is sponsored by Bayer Haemophilia

Award Program (BHAP) through the *Bayer Caregiver Award*¹, and *MITACS Accelerate Programme*². The partnership has led to the research and development of the Haemophilia Injury Recognition Tool (HIRT).

This App, the Haemophilia Injury Recognition Tool (*HIRT?*) was developed for, and with help from, young men with MILD haemophilia. Many people with milder haemophilia are able to live a normal and active life and have few episodes of serious bleeding. Minor bumps and bruises that occur with daily life or sports often heal by themselves.

But once in a while, an injury may be more serious, and it is important to be able to tell the difference. *HIRT?* was developed:

- as a self-management tool, to help young men with mild haemophilia assess their symptoms and decide if an injury needs medical attention, and
- to help them contact a Haemophilia Treatment Center (HTC).

NOTE: This self-management App does not replace professional advice. It is designed to prepare users to discuss their decisions with their health care team.

From the interview conducted for the 25+ young men with mild haemophilia, the message was clear. The young men we spoke with felt that HTC staff ‘over-reacted’ when they came in with an injury. They also felt that most of the existing patient education information did not apply to them.

This evidence-based self-management tool helps young men with mild haemophilia assess an injury and decide when to seek medical attention.

It is also important to state that there is a reason why the App is emphasizing on “young men”. Our research included young men 18-30(5) years old; therefore, the evidence on these knowledge gaps included this group [1, 2]. The App addresses the issues specifically for this age group. This entire group had haemophilia A or B and were of the Mild form of haemophilia which is 5-50% of the clotting factor (which they have a deficit of) circulating in their blood. This study was a Canadian study with participants from different provinces.

In an earlier published paper by Lomotey et al. [3], we discussed the process flow of the *HIRT?* architecture. In the current work however, we discuss the design of the mobile system and how our choice of design justifies our research goals. Unlike resident mobile applications, *HIRT?* is designed as a hybrid architecture that is “self-contained” and has a cloud-hosted back-end emulating a distributed architecture. The self-contained capacity of the App means most of the features are resident on the mobile so the users can use it without any demand on network connectivity. Some of the resident features include access to the detailed assessment guide, list of Haemophilia Treatment Centers, push notification (used as reminders), and so on. The back-end includes a middleware application server that retrieves the user feedback and sends the request to the NoSQL database for further analysis. The back-end is provided to enable the users to submit their comments and feedback to the research team which can be used for further analysis.

The remaining sections of the paper are structured as follows. Section 2 discusses some works on data synchronization using middleware. Sections 3 and 4 discuss our work and design justifications. We report some preliminary results after evaluating the work in Section 5 and the paper concludes in Section 6 with our contribution and future directions.

II. MIDDLEWARE-BASED SERVICES SYNCHRONIZATION

To understand how middleware has been adopted to solve mobile related challenges, this section reviews some works. In highly distributed systems such as enterprise information systems with cloud back-ends, the major concern is data state (update) propagation and management according to Lindholm et al. [4] and Whang et al. [5]. Update management and synchronization of data is crucial for business continuity, real-time and accurate decision making. Data synchronization which aims at ensuring consistency on every node in the distributed system follows the following processes: database or data source access, data capture or retrieval, data conversion or transformation, and data transmission and security verification [6]. The mode of data exchanges during the synchronization process can be asynchronous (where one party sends a request to the other party and without receiving a response sends other requests) or synchronous (where a response has to be received for every request before another request can be made) [7].

The success of state management and synchronization from many sources can be advanced by deploying middleware frameworks. Lv and Zheng [6] put forward a middleware layer that takes files from multiple sources and applies a rule-based policy that combines those files into a single database. Rules are defined to capture changes that are occurring from different sources and later the changes can be synchronized in a single repository (specifically, XML flat file storage). Yang [8] also shares similar opinions by proposing a middleware layer (the author called it a mediator) that performs mediation duties between front-end applications and back-end services. However, data synchronization issues in mobile networks are more

¹<http://www.bayer-hemophilia-awards.com/awards/>

²<https://www.mitacs.ca/en>

challenging due to the limitations imposed on bandwidths in wireless networks and the intermittent loss in connectivity. These challenges lead to high communication latency during update managements and sometimes updates are unsuccessfully propagated when the mobile is in a disconnected state [9, 10].

Xue [11] therefore proposed a middleware framework (called synchronization server) that aids in reliable and real-time data synchronization in mobile networks. The mobile/client-side application stores the application logic and a cache of the replica data from the back-end data source. The middleware then stores the business logic and the synchronization logic that facilitates the update propagation between the mobile and the back-end. The main objective of the mobile cache is to provide offline data access. Further, the middleware aids in pushing only update from the backend to the front-end; an attempt that is aimed at reducing network overload. The middleware further facilitates the detection of conflicting updates.

Latency reduction is a very important aspect of mobile services design. For example, in mobile multi-player game systems, latency can lead to inconsistency in the game state [12]; while, in mission critical systems such as mHealth [13], data propagation delays can lead to undesired circumstances. The concern for latency reduction and real-time data update has been our research focus over the past years. We proposed a real-time data propagation middleware that aids in data routing among mobile providers and consumers using the health domain as a use case [14]. There are other works that also employed middleware to support end-to-end mobile communication in the health domain. For example, Arunachalan and Light [15] proposed a mobile agent communication protocol that transmits medical data from the back-end server to the mobile clients. The agent communication protocol interfaces the middleware which performs roles such as: data synchronization, data segregation, data distribution, power management, and geo-location detection. Further, the MUHIS (Middleware for Ubiquitous Healthcare Information System) framework which is designed by Sain et al. [16] provides group collaboration and synchronization of data in the distributed medical domain.

Also, Boukerche and Owens [17] investigated packet (data) transmission algorithms that enforce quality of service (QoS) in mobile distributed systems that transmit multimedia packets. In such environment, N servers are facilitated to send multimedia data to M mobile clients as streams where a base station is introduced in-between as a middleware. The question is how to ensure proper arrangement of the fragmented data when they arrive on the client node. The authors therefore focused on the QoS by investigating the impact of the following algorithms: First-In-First-Out (FIFO)— the arrival of packet streams are stored in a single queue as and when they arrive on the mobile node in a buffer, Priority Queuing (PQ)— priorities are set based on which packet stream should be sent in the order of importance, Round Robin (RR)— the buffer is based on FIFO scheduling and queuing and reading from the buffer is also based on FIFO, and Weighted fair queuing (WFQ)— the combination of PQ and RR.

A. Core Research Questions

As shown in the two major sections, our research goals are divided into two streams. We need to answer the questions on usability of the App by the haemophiliacs who may not be techies and the questions on system design to overcome latency during feedback propagation. To overcome the issues of network latency as a barrier for a crucial information access, the app is designed to run without a network (i.e., completely housed on the mobile) and only the feedback requires network. But the answered questions which require further investigations are:

- How do we clearly deliver the feedback to the healthcare practitioners (i.e., the physiotherapists) for further analysis?
- How do we ensure scalability of the middleware?
- How do we ensure hybrid architecture that will support users without network interference?
- How do we address additional concerns (in the form of questions) of the users of the App?

The answers to these questions are offered in the upcoming sections. We detail the architectural design, the workflow execution of the app, the integration into social media techniques, and so on. Further discussion of our latest results regarding the App will also be provided.

III. THE ARCHITECTURAL DESIGN OF THE HAEMOPHILIA INJURY RECOGNITION TOOL (HIRT?)

The proposed architectural design of the App is shown in Fig. 1. Overall, the architecture comprises seven (7) components. The young men with mild haemophilia are enabled to use their mobile phones to access the information which are deployed on several platforms including the iOS and Android devices. The mobile device connects to the proposed cloud-hosted middleware through Wi-Fi or 3.5G. We opt to host the middleware on a privately owned cloud platform to ensure both security and availability at the same time. The middleware has one centralized gateway through which all the communications are routed. The architecture also shows the social media network layer that allows the users to share the App with friends. Moreover, the NoSQL layer is proposed to store the feedback since it is more convenient for textual storage and file attachment. The details of the components are discussed below.

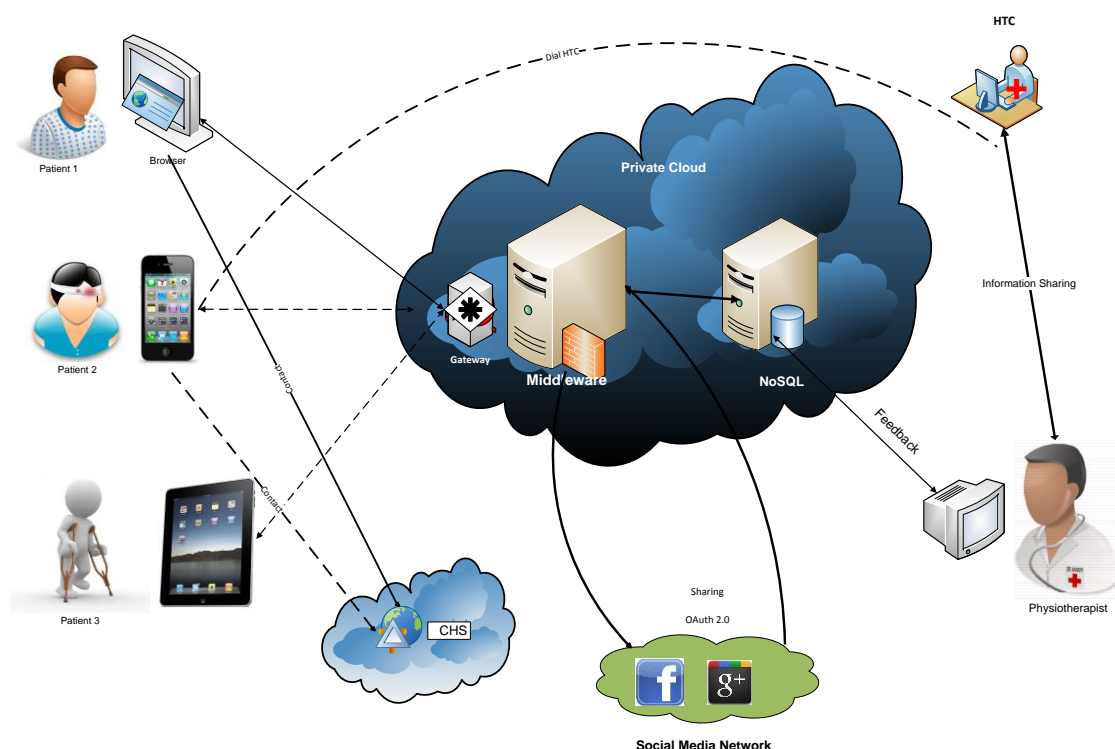


Fig. 1 The architectural design of the N-Screen mHealth environment

B. The NoSQL Storage

The NoSQL storage is proposed to host the feedback from the users and their further comments because the response is text heavy. In this work, we design the data to be transmitted as a JSON data as shown in Fig. 2. This is the data that are made available to the Physiotherapists for further analysis. In this work, we proposed the CouchDB³ storage since the facility enables the storage of text and files as attachment. This meets our design goal since we sort to encourage the users to sometimes send images of their injury when the need arises.

```
{ "_id": "9c0c736e8206577d9acd502592002b02", " _rev": "1-605728c3ac07c87972dc2b58c6f1c241", "Date": "Fri Feb 21 2014 17:49:21 GMT-0600 (CST)", "Device_Version": "6.1.3", "Device_Model": "iPad3,1", "Device_Platform": "iOS", "App_Accessibility": 4, "Using_App_Injury_Accessibility": "Yes", "Using_App_Compression": "No", "Using_App_Rest": "Yes", "Using_App_Elevation": "Yes", "Using_App_Ice": "No", "Using_App_Contact": "No", "Using_App_Ignore_Injury": "No", "Without_App_Compression": "No", "Without_App_Rest": "Yes", "Without_App_Elevation": "Yes", "Without_App_Ice": "No", "Without_App_Contact": "Yes", "Without_App_Ignore_Injury": "No", "Ability_to_Manage_Injury": 4, "App_Reusability": 5 }
```

Fig. 2 Sample JSON data representing user feedback

The NoSQL storage is also stored in the private cloud since that is where the feedback and the survey are stored. When it comes to the analysis, the physiotherapists want to know how the App influenced the young men with mild haemophilia to treat the injury. Other important things to know from the feedback data are how the App aided the users to reach the specified Haemophilia treatment Centres (HTC), What the users would have done differently without the proposed injury recognition tool, and whether all their needs would have been met.

Most of these issues will be explained further in the evaluation section. However, it is important to state that the storage we chose allows us to write map/reduce queries that aid us to determine the answers effectively. In some other cases, the data are exported into CSV or Microsoft Excel for analysis.

C. The Mobile Layer

The mobile is the primary platform on which the young men with mild haemophilia can access the injury recognition tool. At the moment, the App is successfully deployed on the iOS devices (iPhone 5, iPad, iPhone) and Android devices (Samsung Galaxy, Transformer Prime, etc.). In the design, the App is developed in the Xamarin (www.xamarin.com) environment which enables cross-platform deployment of a single code base. This is important for quick fixes and update to the App since one fix can be applied to all other platforms. The mobile design stack is illustrated in Fig. 3. This stack shows the abstract layer of how

³<http://www.couchbase.com>

the functionalities are embedded in the mobile design.

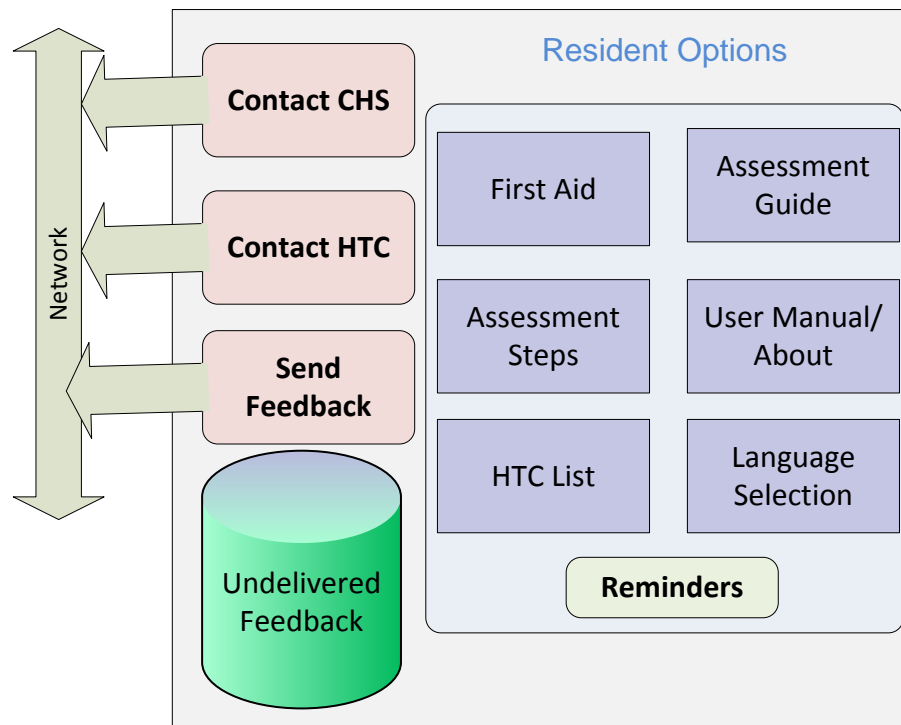


Fig. 3 The mobile design stack of the App

The mobile stack contains the components that are network dependent, resident options, and storage.

- **Network Components:** These are the options that require the users to reach out to the healthcare providers (i.e., the physiotherapists). When the users have questions about the App, they can contact CHS. The Canadian Haemophilia Care (CHS) has the detail and comprehensive list of physiotherapists across Canada. In the event that a haemophiliac has a worsening case, he can also dial the HTC by going through the HTC list provided. The HTC lists the centres by province so the user only needs to choose the province and view the list. There are numbers that are available also for after hours. Also, another network dependent component is the feedback. The users will need to connect to the Internet to be able to send the feedback to the middleware.
- **Resident Options:** These are the components of the App that are on the mobile and are accessible without any network dependencies. The users are enabled to perform initial assessment, go through the first aid, wait and examine the injury later, select language between French and English, access the detail assessment guide, view the HTC list, and the reminders.
- **Storage:** This is where the undelivered feedback is stored for some time until network is restored. Then, the user can re-send the feedback.

The design of the mobile application clearly ensures that network loss in mobile networks does not affect the user when it is time to access crucial information. This is a huge step to ensuring convenient access to the assessment guide at any time and from anywhere.

D. The Middleware Layer and Social Media Networks

The reason for the integration with the social media network is to enable the users to share their experiences with wider audience. This is not mandatory feature for the haemophiliacs especially knowing from the preliminary questions that they shy away from reporting the condition. However, some users will like to contact the CHS and the physiotherapists for further discussions. In this case, users need to provide names, contacts, and so on. To avoid the time of feeling the required information, the social media come handy where users can authorise the accessibility of their information through OAuth 2.0 methodology. This is based on retrieving the user token and using it to identify the user. In this work, two social media networks in use are Facebook and Google+.

Regarding the design of the middleware, the illustration in Fig. 4 shows the various components and the relevance of each component. The *gateway* is the HTTP/S component that allows the mobile to communicate with the middleware over the network. The gateway also interfaces the social media network cloud, and the NoSQL database.

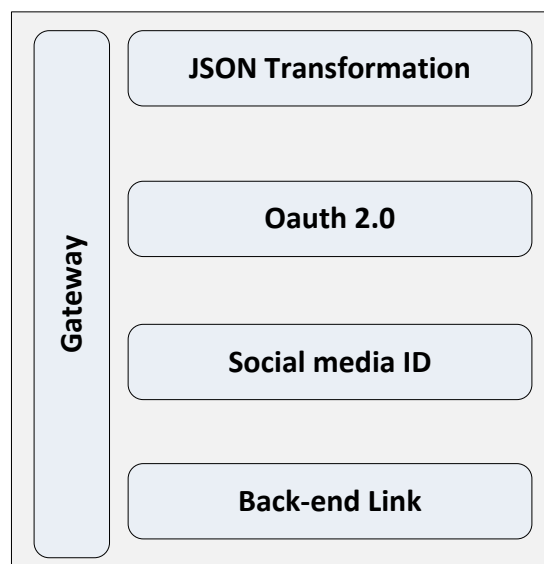


Fig. 4 The anatomy of the provenance-based proxy

Since the NoSQL database under consideration is JSON-oriented, the middleware transforms the actual feedback from the mobile into valid JSON structure. The sample JSON data is shown in Fig. 2. As already posited, the middleware relies on the OAuth 2.0 technique to authenticate users who are having other concerns to be addressed personally by the physiotherapists. Furthermore, the middleware keeps the IDs of the two social media forums and the one that is selected gives the user the middleware the advantage to retrieve the appropriate token. For instance, the selection of Facebook means the middleware needs to select Facebook id so that the tokenization can be done on Facebook and not Google+. Additionally, the middleware relies on the back-end link (which is HTTP interface in Erlang) to connect to the NoSQL (i.e., CouchDB) database.

The middleware is designed in Erlang. In the next section, the process flow is outlined concerning how the haemophiliacs are aided to access the injury over a period of time.

E. Building a Hybrid Mobile-Cloud Architecture

Recently, two general ways of building mobile applications have been adopted which are: the standalone (or resident) architecture and the distributed mobile architecture. The resident architecture is the type of design where the entire application state and all processes are run on the mobile. This style of design is good when the App does not focus on data collection and integration with other services. The distributed mobile architecture is designed in a way where parts of the application runs on the mobile and other components reside on other layers such as external databases, cloud computing platforms, proxies, and so on. In the design of HIRT?, we proposed the hybrid architecture approach (Fig. 5) which combines both architectural design choices.

This is necessary to support the users of the App without any limitation being imposed by network connectivity. In this regard, all the actions required to accomplish the assessment of a minor injury are supported on the mobile as a standalone application. The user interface which is designed as a native project is the interaction point. Some of the screens are presented as embedded HTML (mobile web) and files (e.g., PDF). The major way to notify users to re-assess their injury is through a local push notification technology. Messages are written to the notification center of the phone when it is time to perform assessment. The App is further designed to cause vibration and beep depending on the user settings.

However, the ability to receive feedback, comments, and concerns from the user of the App requires the addition of a database layer as described in previous sections. In the App, there is an interface that enables users to submit answers to a questionnaire. The cloud based database receives this information via wireless networks and the information is stored.

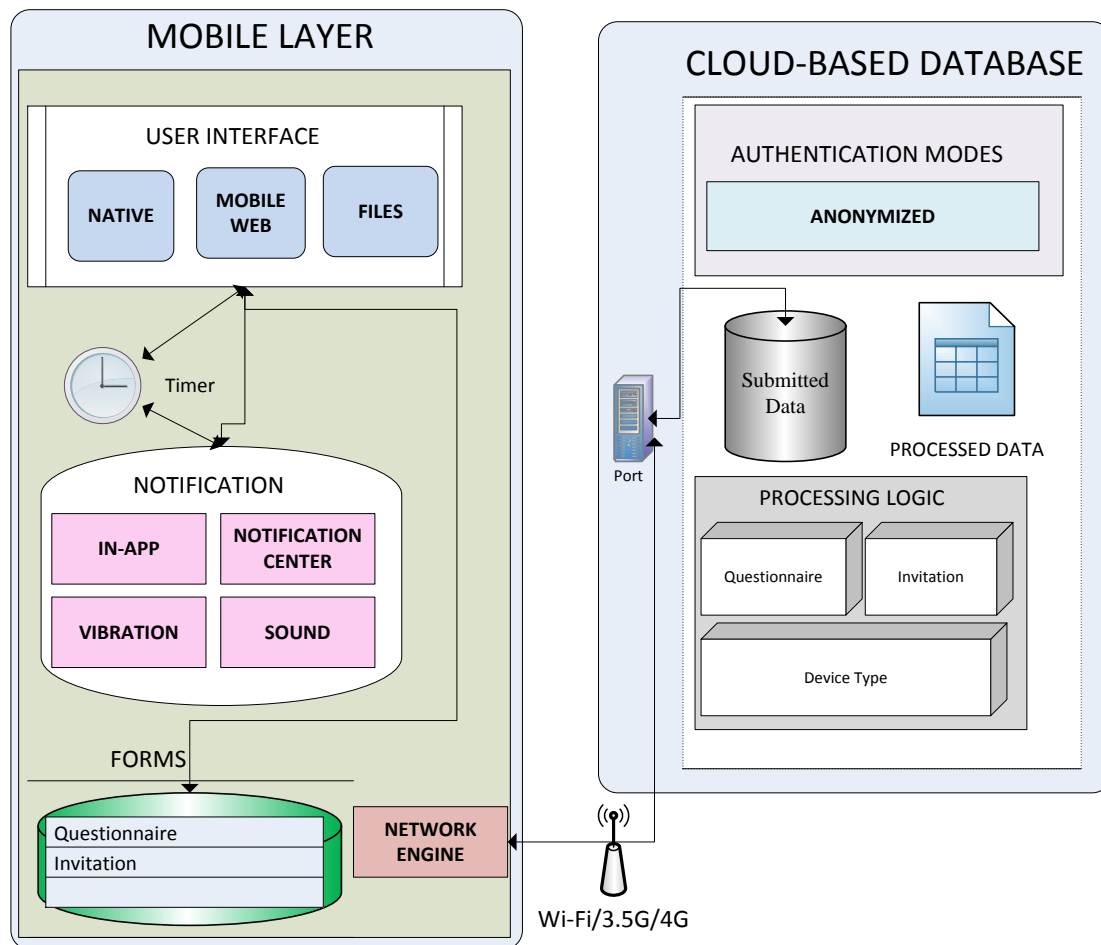


Fig. 5 The enhanced hybrid mobile-cloud architecture of HIRT? showing the individual components and their linkages

There is no authentication required currently as we strive to anonymize the data. However, users who wish to be contacted can provide their contact information such as email or phone. When the data is submitted (in JSON format), a processing logic that runs on the database server separates the information into three – Questionnaire, Invitation, and Device Type. The Questionnaire is a form with set of questions which we require to further analyse the impact of the project on the users. The invitation is for users who wish to be contacted and we use the device type to track which App Store the application is downloaded from (e.g., Google and/or Apple). Once these steps are completed, the data is transformed into processed data in a CSV format.

F. The Process Flow

The App is designed to help users perform assessment of minor injuries by doing the following. These steps are reproduced from [17].

- Looking for pain at the injury site, pain at rest, and pain when moving the injured limb or putting weight through the limb.
- Loss of movement
- Warm to touch, and
- Swelling at injury site

In the application, detail guides are provided on how to examine any of the symptoms listed.

The application of first aid is also series of activities which are summarized below:

- **Compression:** Use tensor bandage or elastic sleeves
- **Rest:** Stop activity such as walking, sports, use sling etc.
- **Elevation:** keep the injured limb above the level of your heart
- **Ice:** Use crushed ice or gel pack

IV. SYSTEM EVALUATION AND IMPLEMENTATION

The proposed system is evaluated in this section based on our research goals. For the purpose of testing the rate of data transfer between the mobile users and the backend, the following devices are put forward: iPad 3 — OS: Apple iOS 5.1.0, Resolution: 2048x1536, Processor: A5X (dual-core, w/ quad-core graphics), Storage: 16GB, RAM: 1GB. The middleware is deployed on a privately owned cloud with the following specifications: Processor: Intel Core i-5, CPU 2330@ 3.12 GHz 3.15 GHz, RAM: 16 GB, System 64-bit operating system, Windows Enterprise Edition. The mobile devices connect to the middleware through 802.11g Wi-Fi 54Mbps connection. In Fig. 6, some screenshots of the App on iPad 3 are shown.

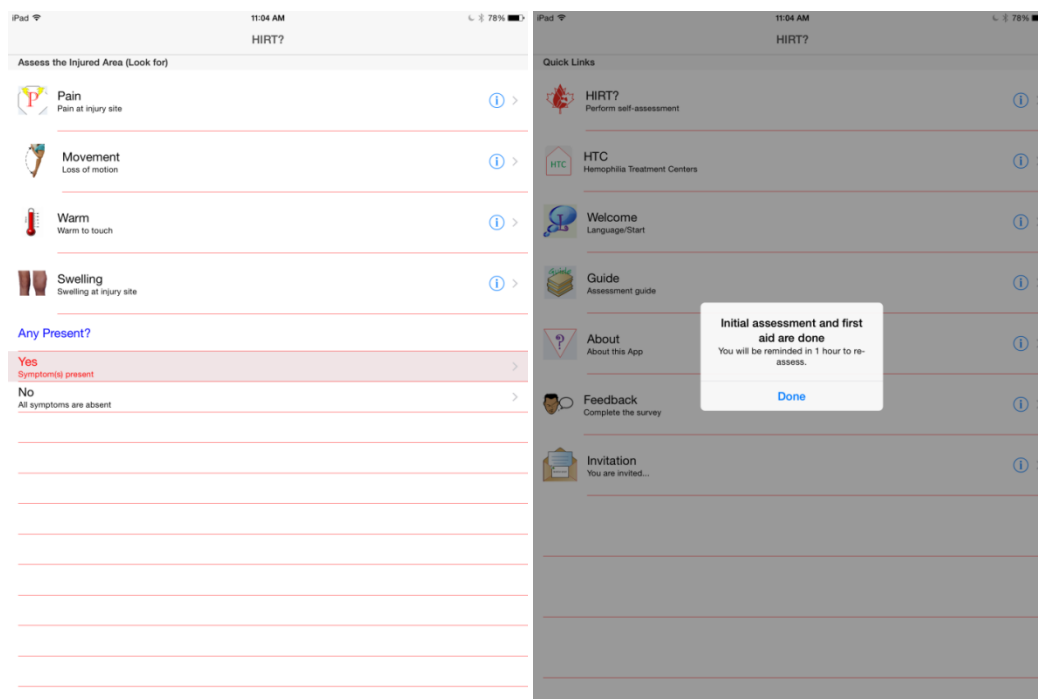


Fig. 6 Some screenshots of the Injury Recognition Tool (HIRT?) on iPad3

The evaluation is conducted in two parts; the system performance perspective and from the user feedback perspective. Focusing first on the system perspective, we analysed the injury recognition tool from the points of view of scalability and data transfer rate. First, we employed several of the test data which are collected in the NoSQL to test the data transfer time. This is important because there are cases where the network signal is weak but if the system is effectively designed, the data transfer will be allowed. Thus, we stored different sizes of data on the mobile starting from 100 MB to 1 GB. In the real situation, the users will send fewer megabytes of data but when the data and comments are stored for a long time due to prolong network loss, the data can grow in size. Our experiment however is testing the extreme cases. We observed the time taken to transfer the JSON payload data (representing user feedbacks) from the mobile through the middleware to the NoSQL. The outcome is plotted in Fig. 7. The experiment is repeated 15 times for each size of data so the plot shows the maximum values, average, and minimum of each set of payload size.

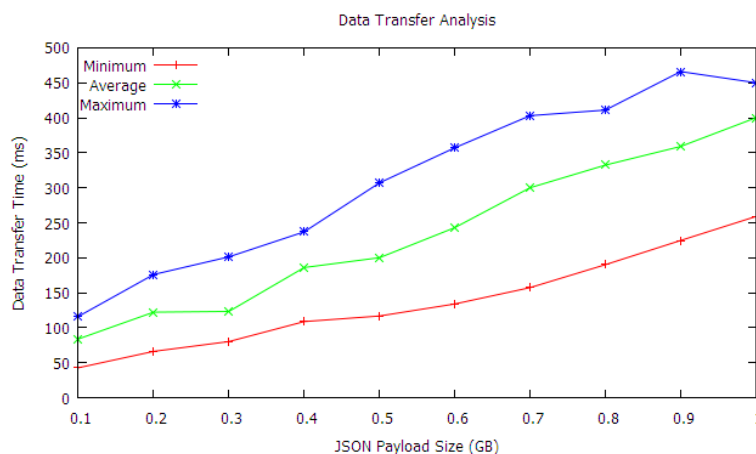


Fig. 7 The data transfer time

We observed that the data transfer time increases linearly as the payload increases. In the case where the file size is 1GB, the maximum time required is 450.00 ms, the average time is 400.21 ms and the minimum time is 258.93 ms. Even when we consider the maximum duration, the transfer time is still encouraging considering the file size. In real life, users will not send 1 GB file size so having achieved a transfer rate of less than half a second is acceptable by the committee. This time is even faster than some of the available works including MUHIS and SOPHRA which were discussed in the background section (i.e., Section 2).

Next, we analysed the scalability of the middleware. Scalability is crucial because when systems reach peak loads, their performance goes down. Currently, the risk of our architecture is the fact that the middleware is centralised. This means, all the users will have their requests issued through the middleware. It is therefore important to understand the behaviour of the system. Since scalability testing requires heavy users, we conducted simulation of the users. We employed a load generator called Apache Benchmark⁴, which sends several concurrent http requests to the middleware. Our primary aim here is to evaluate the throughput which is measured in requests/second. We observed the range of concurrent requests from 500 to 50000. This represents the users' activities and the underlying principle is to determine how the middleware will handle the load. The result is graphed in Fig. 8 and a brief overview in Table 1.

TABLE 1 THE SCALABILITY TEST

Maximum Throughput (requests/second)	Minimum Throughput (requests/second)	Average Throughput (requests/second)
6659.71	148.11	3964.876

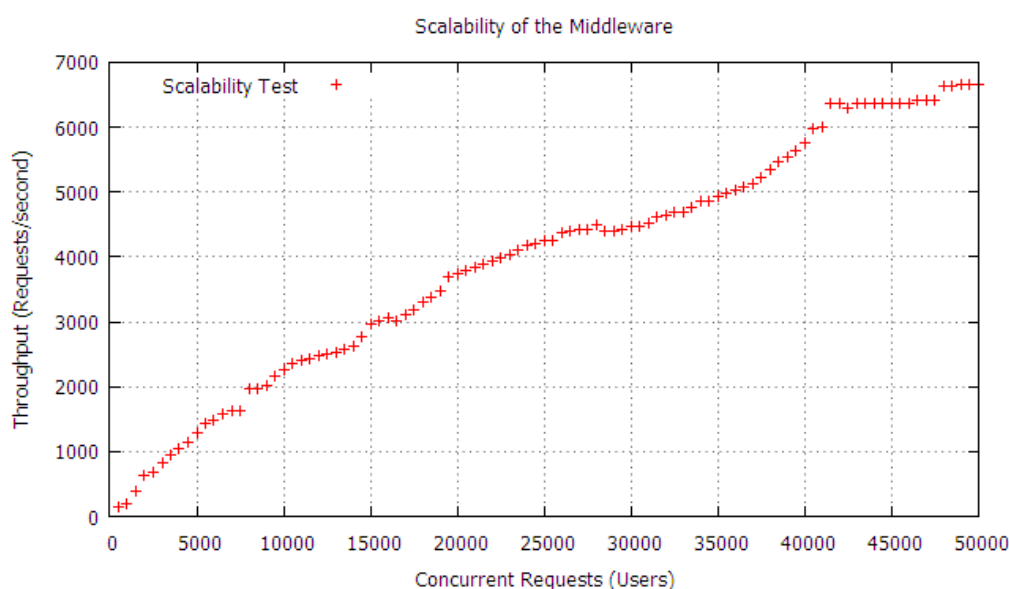


Fig. 8 Scalability test result

As expected, the middleware shows high scalability in term of the throughput analysis. The distribution between the user requests and the throughput follows the exponential distribution of mean 7.9. We observed that the middleware can support up to peak load of 50000 users and at this load, the request processing rate is 6659.71 requests/second. Considering the fact that haemophilia is a rare bleeding disorder, we do not anticipate that 50000 patients will send request concurrently on per second basis. However if they do, the request can be processed. The other observation is that, the middleware crashes after about 51300 requests.

Now that we are comfortable with the performance boost, let us discuss the user feedback aspects.

As part of the injury recognition tool, the users were asked to provide feedback which is in a form of survey. A sample feedback is what is shown in Fig. 2. In the survey, the users were asked to answer a five-part question. These parts are:

- Part A: Accessibility – How easy was the app to access/use?
- Part B: Using the App – Did you assess the injury? Did you apply first aid? Did you contact HTC? And/Or did you ignore the injury?
- Part C: If you did not have the App – Would you have assessed the injury? Would you have applied first aid? Would

⁴<http://httpd.apache.org/docs/2.2/programs/ab.html>

you have contacted HTC? And/Or would you have ignored the injury?

- Part D: Injury Management – How do you feel your ability to manage the injury using this App?
- Part E: Re-usability – Would you use this App again?

From the above set of questions, the test users responded and we analysed their responses. Overall, the test group like the injury recognition tool (HIRT) and its simplicity. Concerning the features, 100% of the respondents like the inclusion of the Haemophilia Treatment Centres (HTC) across the whole of Canada. Some expressed concerns about travelling and not able to contact the right sources when injury occurs. Some of the testers opined that we include the lists of haemophilia clinics in the United States of America (USA) as well but we are limited to do that as of now since this project is a Canadian project and we do not have access to health repositories abroad yet.

Again, 100% of the respondents like the inclusion of the detail assessment guide. One user who gave us his feedback through a conference phone call says he will never use the App again if not for the detail assessment guide. The guide is detailed but simple to understand with images on different injuries to different parts of the body. Another 100% score was received by the reminder feature that informs the users to assess the injury again. There was only one user who expects the sound notification to be a beep rather than a symphonic music and this concern has since been addressed. The users are notified through phone vibration, beep, and alert message on the screen to re-assess the injury. They found this feature new.

On the ease of use (i.e., accessibility) the testers like the simplicity and clarity (100%) but expressed the need to improve the aesthetic to meet the visual grammar of young men. This has also been factored into the App store version release.

Overall, we hope the App will influence users in calling the HTC and avoiding the risk of ignoring the injury. The tool is found useful and this shows how we have progressed on aiding young men with mild haemophilia to assess their injury by empowering them through a mobile App. The development of booklets would have failed even as seen in our preliminary investigations in Section 1.

V. CONCLUSIONS

Across Canada, physiotherapists had seen situations where young men with mild haemophilia waited too long to come in for treatment after an injury. Haemophilia is a bleeding condition where the blood does not easily clot when injury occurs. In order to understand why this is happening, several meetings are held with stakeholders who opined that designing booklets can create awareness among the young men with mild haemophilia. However, our preliminary evaluation of the situation with over 25 young haemophiliacs, points to the fact that mobile App may be a better option. To build the mobile App however calls for several research questions on overcoming mobile system challenges and influencing the attitude of the users. Hence, research collaboration is initiated between the physiotherapists and computer scientist to build a mobile-based injury recognition tool (HIRT?).

The App is designed as a hybrid architecture (integrating standalone architecture features with distributed architecture features) through which the users can access the core functionalities on their mobile devices (smartphones and tablets) without the requirement of network availability. The users found the App very useful and appreciate the provision of the list of Haemophilia Treatment Centres (HTC) across Canada, the detail assessment guide, and the simplicity of the App which makes it easy to use. Further, we can say that the young men with mild haemophilia liked the overall content, and the “re-assess” injury feature - this was a new concept for them.

Overall, we raised some further research questions and they are answered as follows:

- How do we clearly deliver the feedback to the healthcare practitioners (i.e., the physiotherapists) for further analysis? – The JSON data is delivered to the physiotherapists who can query based on what they want.
- How do we ensure scalability of the middleware? – Though the architecture is centralised, the middleware can support about 50000 users. This number probably exceeds the real users at a time.
- How do we ensure fault-tolerance of the middleware? – This question is not addressed in this work and will be discussed in details in future works; dedicated solely to fault-tolerance.
- How do we address additional concerns (in the form of questions) of the users of the App? – The users may contact the Canadian Haemophilia Society (CHS) and have access to all the contact information of the physiotherapists.
- How do we provide satisfactory and dynamic information beyond the booklet approach? – The detail assessment guide is incorporated into the App.
- How do we encourage first time users to inform other friends of the same condition to use the App as aid? – Though social media are proposed, this question will need further review in the future because the haemophiliacs already do not want to report the issue.

There are requirements for future work such as the evaluation of the App on the behavioural impact and changes in the

lifestyle of the haemophiliacs. This evaluation will take a couple of months since behavioural analysis is multi-faceted.

ACKNOWLEDGMENTS

We will like to thank the following agencies for funding:

- BHAP - The Bayer Haemophilia Award Program (BHAP) through the Bayer Caregiver Award, and
- MITACS Accelerate Program.

Further thanks to the Canadian Haemophilia Society (CHS) for their support.

Also, thanks to **Wendy Lane** from the Biomedical Engineering Research (BER), The World Academic Publishing Co., Ltd., and the entire editorial team of BER.

REFERENCES

- [1] J. Nilson, C. Schachter, K. Mulder, M. Hahn, M. Steele, P. Hilliard, and C. Jarock, "A qualitative study identifying the knowledge, attitudes and behaviours of young men with mild haemophilia," *Haemophilia*, vol. 18(3), pp. e120-5, May 2012.
- [2] J. Nilson and K. Mulder, "A new self-assessment pathway tool for young men with mild hemophilia A and B who experience musculoskeletal bleeds," Poster, WFH Congress, France.
- [3] R. K. Lomotey, K. Mulder, J. Nilson, C. Schachter, K. Wittmeier, and R. Deters, "Mobile self-management guide for young men with mild hemophilia in cases of minor injuries," *International Journal of Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 3, iss. 1 (NetMAHIB), pp. 1-12, July 2014, 10.1007/s13721-014-0064-z, Publisher: Springer Vienna.
- [4] T. Lindholm, J. Kangasharju, and S. Tarkoma, "Data Synchronization Middleware for the Mobile Web," *Mob. Network Application*, vol. 14, iss. 5, pp. 661-676, October 2009.
- [5] K. Whang, I. Song, T. Kim, and K. Lee, "The ubiquitous DBMS," *SIGMOD Rec.*, vol. 38, iss. 4, pp. 14-22, June 2010.
- [6] J. Lv and X. Y. Zheng, "Research for a Data Synchronization Model Based on Middleware and Rule Base," *Information Science and Engineering (ICISE), 2009 1st International Conference on*, pp. 2998-3001, Dec. 26-28, 2009.
- [7] L. Mokdad and M. Sene, "Performance measures of a call admission control in mobile networks using SWN," *In Proceedings of the 1st international conference on Performance evaluation methodologies and tools (valuetools '06)*, ACM, New York, NY, USA.
- [8] G. Yang, "Data synchronization for integration systems based on trigger," *Signal Processing Systems (ICSPS), 2010 2nd International Conference on*, vol. 3, pp. V3-310-V3-312, July 5-7, 2010.
- [9] W. Shun-Yan, Z. Luo, and C. Yong-Liang, "A Data Synchronization Mechanism for Cache on Mobile Client," *Wireless Communications, Networking and Mobile Computing, 2006. WiCOM 2006. International Conference on*, pp. 1-5, Sept. 22-24, 2006.
- [10] D. Zhou, N. Islam, and A. Ismael, "Flexible on-device service object replication with replots," *In Proceedings of the 13th international conference on World Wide Web (WWW '04)*, ACM, New York, NY, USA, 131-142.
- [11] Y. Xue, "The Research on Data Synchronization of Distributed Real-Time Mobile Network," *Computer Science and Software Engineering, 2008 International Conference on*, vol. 3, pp. 1104-1107, Dec. 12-14, 2008.
- [12] A. M. Khan, S. Chabridon, and A. Beugnard, "Synchronization medium: a consistency maintenance component for mobile multiplayer games," *In Proceedings of the 6th ACM SIGCOMM workshop on Network and system support for games (NetGames '07)*, ACM, New York, NY, USA, pp. 99-104.
- [13] R. K. Lomotey, S. Jamal, and R. Deters, "SOPHRA: A Mobile Web Services Hosting Infrastructure in Health," *Mobile Services (MS), 2012 IEEE First International Conference on*, pp. 88-95, June 24-29, 2012.
- [14] R. Lomotey, R. Kazi, and R. Deters, "Near real-time medical data dissemination in m-Health," *In Proceedings of the International Conference on Management of Emergent Digital EcoSystems (MEDES '12)*, ACM, New York.
- [15] B. Arunachalan and J. Light, "Agent-Based Mobile Middleware Architecture (AMMA) for Patient-Care Clinical Data Messaging Using Wireless Networks," *In Proceedings of the 2008 12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications (DS-RT '08)*, IEEE Computer Society, Washington, DC, USA, pp. 326-329.
- [16] M. Sain, H. Lee, and W. Chung, "Middleware in ubiquitous computing system with MedRec architecture," *In Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human (ICIS '09)*, ACM, New York, NY, USA, pp. 149-154.
- [17] A. Boukerche and H. Owens, "Media synchronization and QoS packet scheduling algorithms for wireless systems," *Mobile Network Applications*, vol. 10, iss. 1-2, 233-249, February 2005.

Richard K. Lomotey is currently pursuing his PhD in Computer Science at the University of Saskatchewan, Canada, under the supervision of Prof. Ralph Deters where his main work focuses on mobile cloud computing. He has been actively researching topics relating to: ubiquitous cloud computing and the paradigm shift in enterprise mobility workforce support. Over a couple years, most of his works are industry collaboration with the health sector where several mHealth application development is explored.

JoAnn Nilson is a physiotherapist at the Saskatchewan Bleeding Disorder Program, Royal University Hospital, Saskatoon, Canada.

Kathy Mulder is a physiotherapist, Children's Hospital, Health Sciences Center, Winnipeg.

Kristy Wittmeier is currently in Winnipeg, Canada. She is a Physiotherapy Innovations & Best Practice Coordinator, and Assistant

Professor, Department of Pediatrics and Child Health, University of Manitoba. Also she is the Director, Knowledge Translation, Manitoba Centre for Healthcare Innovation.

Candice Schachter is Adjunct Professor, School of Physical Therapy, University of Saskatchewan

Ralph Deters is currently a professor in the department of Computer Science at the University of Saskatchewan (Canada). His research focusses on mobile and cloud computing.