

Getting to the Right Patient at the Right Time: An Interoperable Mobile App to Track the ED Journey in Hospital

Audrey P WANG ^{a,1}, David PRYCE ^b and Phillip GOUGH ^c

^a *The University of Sydney Faculty of Medicine and Health*

^b *Western Sydney Local Health District*

^c *The University of Sydney School of Architecture Design and Planning*

Abstract. The current legacy ICT framework structures in healthcare are often siloed and do not allow information to flow easily between business analytics and clinical systems, affecting critical decision making. Western Sydney Local Health District (WSLHD) has numerous electronic database systems for business analytics including tracking individual patients waiting for treatment in the emergency department (ED). Administrators of hospital business data report ED performance measures in a weekly static feedback report to clinical and executive staff due to current legacy systems and manual resource allocation processes. The remit of the project was to prototype a system that could integrate data sources from the current QlikSense Dashboard into an interoperable mobile app with the future intention of direct impact on clinical care decision making for the emergency department. A series of meetings between business analytics unit and clinical staff were used to inform a set of requirements for information workflow systems integration to be used on the project. Stimulated patient data that matched typical data feeds from the system was used to develop a prototype interoperable HL7 messaging mobile app that would report waiting patients in their triage categories in near real time. This working prototype with synthetic scenarios and data will inform a future deployable production system with information for the patient journey from the ED waiting room into available hospital beds. As most applications are either designed for business analytics or clinical workflows, integrating information data sources into one mobile application that could meet the needs of both clinical and business performance was novel and integral. This proof of concept project successfully integrated the information systems necessary for both purposes and informs future requirements for an interoperable and deployable cross-platform mobile app.

Keywords. interoperability, mobile applications, triage, HL7, emergency department

1. Introduction

Timely information in the emergency department (ED) allows oversight of: how many people are waiting; level of critical patients in triage categories; and availability of beds in different hospitals. This assists in managing fluctuating surges in demand and supply of emergency services. An example of this is a website that publishes NSW hospital

¹ Corresponding Author, Dr. Audrey P Wang, Discipline of Biomedical Informatics and Digital Health, School of Medical Sciences, Faculty of Medicine and Health, The University of Sydney. E-mail: Audrey.wang1@sydney.edu.au.

emergency department waiting times to allow the general public an overview of the number of patients assessed by a triage nurse and waiting for treatment at their local hospital. The simplified overview includes how many patients are waiting over a three-hour period from historical data and the general number of beds/treatment spaces in the Emergency Department [1]. Although the information is periodically refreshed throughout the day approximately every 20 minutes; it remains far from real-time.

Metrics from routinely collected hospital data are often siloed in legacy IT systems making it difficult to provide timely insights for frontline clinicians. The local provision of timely information at emergency departments on similar metrics for key ED clinical decision makers could improve feedback loops for resource allocations closer to the real-time pressures experienced. Candidate metrics include waiting times, triage categories and availability of hospital beds in wards for admission post arrivals at the emergency department. These key metrics are usually collected for two different purposes in a busy metropolitan Sydney hospital: first for business analytics of hospital performance and for operational reporting to centrally based NSW Health systems.

The business analytics department and emergency department identified the need for a redesigned data analytics application that is accessible on mobile devices, interoperable, and applicable for clinicians' provision of patient care. The current application is used across the WSLHD's four hospitals to measure key performance indicators (KPI) of the emergency department performance. The KPIs are built from patient and operations data that helps the ED measure how efficiently resources are used and effectiveness of patient care. The current application is powered by Qlik Sense, a proprietary business software tool not easily accessible, nor interoperable on different platforms. We hypothesised it would be ideal to implement an interoperable mobile application for data analytics, for both WSLHD clinicians and business analysts to resolve the timeliness of critical data metrics mentioned to ensure smooth transition of care across multiple settings.

2. Requirements for WSLHD ED Interoperable Mobile App

The WSLHD emergency department required a redesigned data analytics application which is accessible on cross-platform mobile, interoperable, and more useful for clinicians' provision of patient care. The current web-based application is used across the WSLHD's four hospitals to measure KPIs of the emergency department. The KPIs are built from patient and operations data which help the ED measure how efficiently resources are used and how effectively patients are cared for. Overall, the outcome will be a data analytics application available on smartphones which is interoperable and useful for both WSLHD clinicians and business analysts.

2.1. Background to Existing Systems

The current, non-interoperable, Qlik Sense business application is not part of current clinical workflows and exists as a dashboard on an internal web portal. Clinician's current workflow requires access to numbers of patients waiting in triage, their categories and bed flow management from the patient portal. It provides frontline staff with managerial responsibilities in the emergency department information in an easy to use modality via a mobile phone.

We received a set of requirements following a series of internal inter-departmental meetings. These included developing a conceptual working mobile application (mApp)

prototype with a new design and framework that would have operationally timely information request systems and updates (< 5 minutes) once deployed with all 32 metrics from the Qlik Sense dashboard in the new cross-platform mobile app. The display would be a clean and user-friendly visualisation, allowing clinicians to access any metrics within 10 seconds of launching the application. It also must have no real patient or hospital data latently stored on the device to ensure the privacy and security of sensitive data. The users would use familiar data security and privacy authentication systems. The mApp should be interoperable and identify capabilities for SMART on FHIR standards [2, 3] and have an application programming interface (API) that could be used to support extended functionality including bed management in the future. Finally, information requests and updates must occur in the timely manner for operational usefulness when deployed.

The mApp includes the following functional requirements: First, users will be able to securely login to the application using their work credentials. Data will be displayed for one hospital location at a time and allow for easy navigation to other locations. Each location will be displayed in a similar format to allow for comparison. Finally, the application will provide additional information on demand about the data being displayed, so users are able to understand the information, following W3C protocols to ensure the app has high usability for a wide base of users (e.g. hearing or vision impaired).

3. Prototype Overview

The prototype included a front end, back end consisting of an interoperable Python Flask framework with interactions into the backend database, with existing HL7 [4] standards server populated with mock patient data, called with HL7 messaging APIs. The current QlikSense web-based application receives data from an existing PostgreSQL database continuously updated from a wide range of legacy IT systems that contain patient related emergency department data. The data includes how long the person has been waiting post front-desk administration registration and the number of patients allocated to each triage category. A diagram of the structure for System Integration implementation is provided in Figure 1.

The mApp prototype system's cross-platform front end was built using the JavaScript-based React Native framework. Other dependencies included Node.js and Expo. The Expo Client App allowed live testing of the application on any mobile phone and can be downloaded from Apple App Store or Google Play Store for Android. A Flask server microframework was implemented to support the functional requirements and create the system's middle API, built using Python code. It has modular dependencies that can be interoperable. The middle API utilised the SQL Alchemy Python module to interact with the dummy version of the database used by the hospital, built using PostgreSQL. Further dependencies for developing a functional API include Flask-Script, Flask-Migrate, Psycopg2 and APScheduler.

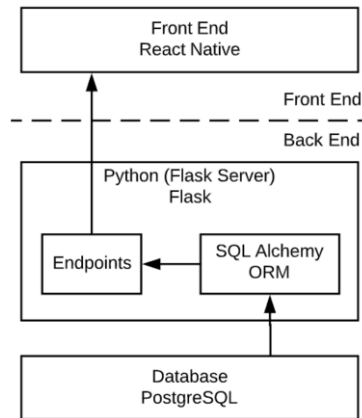


Figure 1. High level architecture diagram of mApp prototype.

4. Testing Results

The development team utilised various tests and tools to verify and validate system integration functionalities. The feedback findings from inter-departmental meetings were heavily relied upon to test the front-end prototype functionalities. A range of tests were instigated to determine if the mApp could be successfully deployed in the future base on the current system integration framework profile:

Unit Tests - Automated tests to check the output for the smaller modules of the program. This has been automated. For the backend, the tests done were to check if the API correctly includes all the ED KPIs for the react code to fetch data from. The frontend tests here were mainly done to establish if the rendered screen exists and comparing them to the snapshots of the determined wireframes.

Integration Tests - Testing the connection to the database.

Functional Tests - Manual tests to check whether system responds correctly according to the different predetermined functions.

The test coverage of the codebase is achieved across three testing areas: the front-end testing of the React Native codebase, back-end testing of the Python codebase, and durability stress testing of the system. The dummy database itself does not need to be tested because the codebase has been provided by WSLHD. The app was tested and able to sustain 100 users simultaneously and was pressured tested to 1000 concurrent pulls on the server for 10 seconds using apache Benchmark. The response rate was 3.5 seconds. Consideration of the code base adherence to SMART on FHIR standards were considered in the project. All changes conforming to HL7 message standards were already in place (from the hospital team prior to this project) for database informatics pulls. All other code base adhered to open source concepts using python language and any API standards used could be modified for REST API standards.

Automated test coverage was 60% for the front end. The rest of the coverage were tested using manual testing. This included testing the app's functional output to deliver login authentication, correct page navigation and appropriate key performance indicator data items. Login in credentials mimic NSW Health methods for authentication tested well in the prototype between the database and login screen. This performed well as a necessary gateway into the app. The application will need integration into the NSW

Health authentication and encryption system during live deployment in the future. Automation of user log out occurs if static use of one page is for more than 5 minutes. Automated testing is shown in Table 1. The stress test is for the cross-platform application when hosted on a local machine allowed a transfer rate of 628.60 Kbytes/sec. As a result, the stress test is largely testing the speed and robustness of the network interactions and system structure. The stress test must be repeated once deployed to a real environment where the system is connected to WSLHD servers. There was no latent cache of data required and therefore allows protection of sensitive patient related data once user logged out.

Table 1. Automated testing results.

Test	Codebase tested	Type	Result
Front end coverage	React Native	Automated	60%
Back end coverage	Python (Flask)	Automated	87%
Durability and stress testing	System	Automated	mean: 3.7s, max: 3.8s, min: 0.062s

4.1. Limitations

We identified four live-deployment limitations. First, authentication systems were not live tested for NSW Health authentication systems, but the prototype allows for the necessary security protocols to be embedded into the app. Also, validation of HL7 messages will need to have a framework of testing that will ensure quality control of messaging to draw the data from a much larger live regularly updated database. Further, the mock prototype app which is functional and able to call up synthetic aggregated patient data could have extended functionality to individual patient data. Finally, we have a placeholder for patient flow portal services to assist with database updates for beds for WSLHD. This functionality was not prototyped fully due to limited time and stakeholder availability for this function during the period of the project. We also identified one limitation through pressure testing. The Flask server has the ability to integrate more interoperable extensions including Flask-RESTful for building REST APIs or proxy servers, that will allow further adherence to SMART on FHIR standards. This will be part of the proposed development for future deployment.

5. Discussion and Conclusion

Overall, the prototype of across-platform mobile application for Emergency Department data was determined to be largely be a success. The working data analytics network connections were prototyped as useful for both hospital clinicians and business analysts to resolve the timeliness of critical data metrics to ensure smooth transition of care across multiple settings. The success was measured by having a reproducible working code base with documentation that resulted in a final deployable cross-platform mobile app. Sample aggregated patient data could be displayed and lower level data breakdown to individual patient data could be extended for live deployment. The data in the application updates within five-minute intervals or less, as demonstrated by the pressure tests, from the dummy database. The privacy of patient and hospital data was secure by ensuring data is never saved to the individual device. A higher level of authentication protocol could be implemented in the future where the usernames match the current NSW Health systems.

This project had time and scope limitations for project completion. Therefore, product success criterion that were not achievable in this iteration included:

- The application can be successfully uploaded onto a cross-platform phone by anyone.
- The data in the application is read from the WSLHD database directly.
- The application can connect to WSLHD databases offsite.
- Successful in vivo usability tests conducted with actual clinicians to assess the value of data analytics tool.
- The presence of an initial application directory page for additional extensions.

Overall, the system structure served well for the development where the application was kept local to the machine and isolated from the WSLHD servers, using a middle network of APIs. Upon real deployment onto the ED floor using WSLHD databases, a different set of network interactions involving a server proxy is likely to be required if data accessibility issues arise. Future work will be needed to address the limitations of the current interoperable mobile app for live deployment to ensure all implementation success criterion are met. In conclusion, we achieved adequate evidence to support our hypothesis of the required integration of siloed systems to demonstrate success in integration of clinical and business analytics to track the patient journey in ED.

Acknowledgements

We would like to acknowledge the contributions of Katherine Sutarlim, Abhinandan Srinivas, Onam Khan, Antonia Mijatovic, Dugald Shannon in working on the prototype as part of a capstone student software group project. Adhish Panta, Jim Cook and Viji Venkataramani as part of the collaboration with the ICT Tech Lab and School of Computer Science, University of Sydney. Reshma Kolambkar for providing HL7 expertise and codebase, and Margaret Murphy for her clinical emergency department expertise, to build the necessary system interactions into clinically relevant workflows. A demonstration of the app was given in the University of Sydney Westmead Innovation Showcase. Institutional ethical approval was given WSLHD HREC QA2009-16.

References

- [1] Health N. Emergency Department waiting times in major NSW hospitals 2020 [Accessed 14/02/2020]. Available from: <https://www.emergencywait.health.nsw.gov.au/#a>.
- [2] Mandel JC, Kreda DA, Mandl KD, Kohane IS, Ramoni RB. SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. *Journal of the American Medical Informatics Association*. 2016;23(5):899-908.
- [3] Metke-Jimenez A, Harrap K, Conlan D, Gibson S, Pearson J, Hansen D, editors. A SMART on FHIR Prototype for Genomic Test Ordering. *Digital Health: Changing the Way Healthcare is Conceptualised and Delivered: Selected Papers from the 27th Australian National Health Informatics Conference (HIC 2019)*; 2019: IOS Press.
- [4] Lopez DM, Blobel B. Architectural approaches for HL7-based health information systems implementation. *Methods of information in medicine*. 2010;49(2):196-204.