Submission Id: 222

Title: TECHNICAL DESIGN CONSIDERATIONS FOR A WEB RHD REGISTRY APPLICATION FOR USE IN LOW-RESOURCE SETTINGS

Authors: Riley Morrison, Nicholas Felicelli, Kristen Tillman, Sarah de Loizaga, Andrea Beaton, Judith Dexheimer, Emmy Okello, Jafesi Pulle, Jessica Abrams, Jeremiah Mwangi, Joselyn Rwebembera, Joselyn Rwebembera, Neema W. Minja, Ndate Fall, Emily Chesnut, Karen Lapthorn

Background & Aims: The Active Community Case Management Tool (ACT) is a patient registry web application designed to improve care for patients with Rheumatic Heart Disease (RHD) by providing tools to facilitate disease management, and report qualitative metrics within a single application. We specifically designed the ACT registry to meet the unique needs of low-income countries with features to accommodate unreliable Internet connections and facilitate communication between community clinics and higher-level health centers. Additionally, data security and application versatility to different environments were key. Finally, low cost and adaptable design allowing for future scale up were prioritized.

Methods: We utilized several AWS (Amazon Web Services) technologies to build our registry, which are less expensive than traditional physical hardware and can be deployed anywhere. AWS S3 buckets hosted our web application (built with React). Through AWS RDS we hosted a serverless Aurora MySQL database which contains patient data. Privacy laws were upheld by securing patient data by setting up the registry's web application under HTTPS and controlling access to the registry through AWS Cognito authentication of users. Additionally, within the registry, a users' access to patient data is limited essential tasks only based on their user role. Lastly, user actions are stored in audit logs for review. We built an API in Python and made use of a Python library, Chalice, to deploy that API to AWS Lambda functions. We utilized React's browser caching to create data entry forms that could function entirely without an Internet connection.

Results: The ACT patient registry underwent user testing with volunteers and select clinics before broader implementation. Since then, the ACT application has transitioned to the national RHD registry within Uganda, supporting more than 4, 000 patients and over 100 users across dozens of clinics. As the number of patients grew, the ACT patient registry has maintained low costs within AWS. The serverless nature of our application means our AWS services expand and shrink as needed, rather than using a fixed amount of servers that run up costs. Hence, if many users are using the registry at once then more resources are started but during down times those resources are turned off. We have not encountered any performance issues and all of our API and database calls are under 30 seconds. Lambda functions not only connect our frontend to our database but also run lengthy and complex functions (such as calculating patient adherence to treatment plans and report generation) overnight and storing the results in our database, thus facilitating these shorts database calls.

Conclusions: Developing the ACT patient registry has demonstrated the feasibility of creating a low cost but feature rich application that can meet the unique needs of low-income countries. By incorporating management tools with patient data in a single application, clinicians can make data driven decisions around RHD treatment, with the goal of improving outcomes for those living with RHD. By utilizing AWS's serverless technologies to address specific needs of low resource settings, ACT can be expanded to use for other diseases and additional low resource settings.