Organization Name

Raxa

Organization Address

2nd Floor, 12 Hauz Khas Village
New Delhi, Delhi; India

Website

Web: http://raxa.org
Wiki: https://raxaemr.atlassian.net/wiki/display/RAXAJSS
GitHub: https://github.com/Raxa/

Mission Statement

The Raxa JSS EMR is a medical information management system ("Raxa") designed for and initially implemented at the Jan Swasthya Sahyog (JSS), a healthcare non-governmental organization (NGO) working in a largely rural, underserved community in India.

Contact person

Nathan Leiby
Software Developer
nathan.leiby@raxa.org or nathanleiby+raxa@gmail.com
+91 98181 39831
Problem Statement

We are faced with managing the health system of hospitals. We have one particular partner hospital, based in rural India, which is transitioning from paper documents to an electronic medical records (EMR) system. They need to track, organize, and present their data in the most meaningful ways to assist medical doctors at 1) assessing disease trends 2) assessing trends in a specific patient’s care.

Project Proposal

One-sentence summary: We seek flexible, powerful reporting tools (manipulable visualizations of custom selected data/statistics) to use on the medical data we gather in our EMR.

Inspiration for medical data visualization:

- http://www.ted.com/talks/anders_ynnerman_visualizing_the_medical_data_explosion.html
- http://uxmag.com/articles/beyond-the-medical-chart

For a simple example, we could allow doctors to graph blood pressure over time. Or for hospital administrators, they could view rates of different varieties of TB as related to location and time. For any of these, the ideal is that users (doctors, administrators, others) at a given hospital will be able to create custom visualizations that solve their particular issues in care.

If time permits or as a separate sub-project, we'd like to tie custom analytics into the system that doctor's use at point-of-care. For the example of "blood pressure over time", we could cause certain measurements to propagate alerts, for example if an unhealthy trend (delta > X, % change > Y) is present in a patient's history. By propagating alerts in real-time, the doctor will be prompted to investigate this issue immediately when in a patient consultation.

How we can support students

- We have several full-time software developers who can answer questions about our technology and frameworks.
- One of our full-time software developers will serve as an assigned mentor from our organization to the student team.
- We have a vibrant community of health professionals, user experience designers, software developers, project managers, lawyers, students, and others who volunteer with our work. These various contributors can assist the students throughout their tasks.
- Students are invited to visit and work in our office in New Delhi, India, and to explore the challenges of designing, deploying, testing, and iteratively improving their work at a real hospital. (Although it is unlikely students will be able to visit
during the semester, they may wish to visit during their winter holidays. After the course project, successful students would have the opportunity to extend their project or take up another project via further courses, research, or an internship.)

Additional information

Timezones
Our team is in the IST timezone (India), so we must communicate around our timezone difference (IST = EST + 9.5 hrs). For example, we’re most easily available in early morning or late evening EST. We are used to working with international contributors across many timezones, so can assist in coordinating.

Technologies
The data is gathered via our EMR, an open-source web app built in Sencha / ExtJS (JavaScript frameworks), deployed on desktop web browsers, tablets, and phones. Data is stored in the OpenMRS data model. Using i2b2, we should be able to retrieve patient data from OpenMRS in a flexible way.

Open-source
Our present work is entirely open-source. Students must commit up front to allowing their work to be developed according to our open source process and license. (We use the Apache 2.0 license - http://www.apache.org/licenses/LICENSE-2.0)