Effects of two-way data binding on better user experience and easier development of Clinical information systems and LIMS

Nihad Omerbegović, International Burch University, Franuske Revolucije bb, Sarajevo, Bosnia and Herzegovina

Apps for Healthcare Professionals

Keywords: Clinical information systems, electronic health record, data search, web-based clinical applications, data quality, user experience

While designing and developing clinical information systems, either from an ontological aspect, database architecture, or development environment, it is essential to understand all relevant points—defining all requirements, and list all possible interactions with other systems, LIMS. For instance, as a software used to manage data and increase control quality in laboratories, it should be automated to provide maximum efficiency. On the other hand, manual processes should be minimized, search of data very efficient, ensuring an easy task for a lab worker. Manual or operator work will always be required to some extent, so these interfaces must aim to optimize this process by usage of efficient tools and modern web technologies, such as AngularJS framework and Javascript.

When surveyed about what the main challenges facing in their operations, laboratories shared a short list of common issues [6]. These included: meeting turn around times (TAT) and testing times, missed sampling events, certification audits, unqualified departments, lost time searching for or checking data, and slow flow processes. These issues can be grouped in two main topics: data quality and costs. In a laboratory, these two topics are usually linked. The aim of this paper is to show how a process of searching for data in a typical electronic patient record can be improved. We developed a simple demo app using AngularJS framework—as a proven and powerful front-end environment [6]. Javascript language and standard HTML.

Image 1: WHAT IS A LIMS?

Tools and Methods – AngularJS Framework and Two-Way Data Binding

Why AngularJS

HTML is great for declaring static documents, but it falls short when you need it for dynamic systems or web applications. AngularJS is an extension of HTML, adding an easy-to-use data binding and two-way data binding syntax, which make it easier to build large, single-page web applications. Using AngularJS, it is easier to develop an app using the same tag, and with the same syntax, JavaScript.

The real power of AngularJS is the dependency injection, which makes it easier to mock or test app components, a big plus for the development process.

One-way Data Binding

In the real world, an AngularJS application can be a collection of HTML markup, each of which represents a function or a state in the application. The two-way data binding is achieved by the use of the $scope.$watch method, which watches for changes in the HTML and updates the JavaScript accordingly.

Two-way Data Binding

Two-way data binding occurs when a value is bound to a HTML element that can both change and display the value of the variable.

Results

2.1. AngularJS and Healthcare Demo

We used AngularJS directives to bind a model variable to the HTML, which can be read or written by using the $scope.$watch method. The method can be efficiently used when we need to search data from LIMS, as where there is a requirement for efficient filtering of medical data. Figure 1 shows the difference between traditional one-way data binding and two-way data binding model.

Conclusions

Data interoperability and data quality are some of the main factors in modern clinical information systems today. The main aim of these works is to develop new tools of clinical information systems in order to develop standard interfaces for researchers and development of new clinical information systems in order to develop standard interfaces for researchers and developers. The work described in this paper is just one step towards this goal. We intend to show how combining these technologies and modern protocols like HTML5 and Javascript can have a greater impact on user experience and information sharing. Future work will be devoted to improve the user experience and efficiency of data science in LIMS apps.

References