

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

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

Introduction:

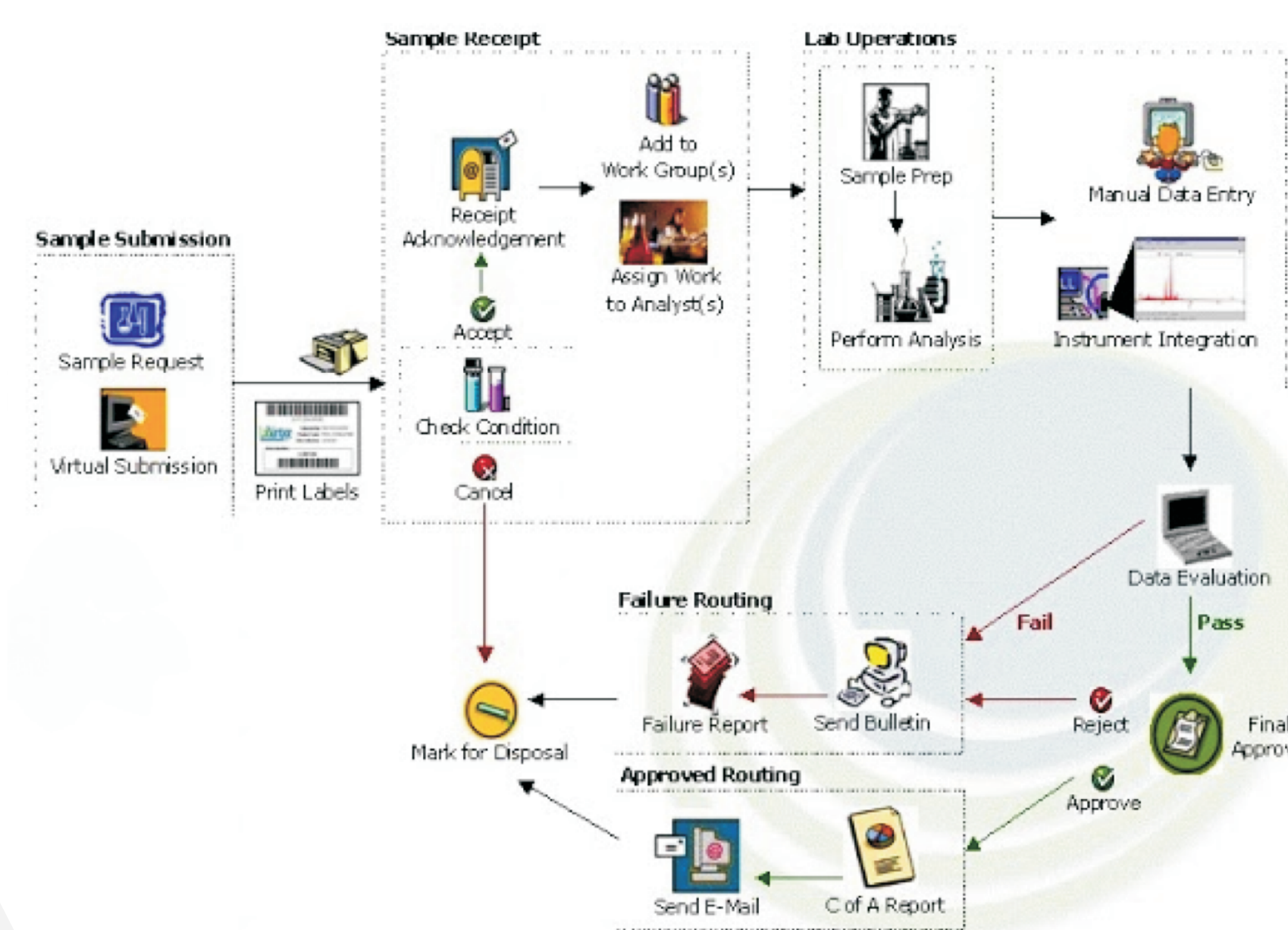


Keywords: Clinical information systems, electronic health record, data search, web-based clinical applications, data quality, user experience

While designing and developing clinical information systems, either from ontology aspect, database architecture, or development environment, it is crucial to understand all relevant points – define clearly 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, should be automated in order to provide maximum efficiency. On the other hand, manual data entry 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 point - that's why we 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 major challenges were facing in their operations, laboratories shared a short list of common issues [2]. These included: meeting turn around times (TAT) and holding times, missed sampling events, certification audits, unprofitable departments, lost time searching for or checking data, and cash flow problems. 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 environment [5], JavaScript language, and standard HTML.

Image 1. WHAT IS A LIMS?



## Tools and Methods – Angular JS Framework and Two-Way Data Binding

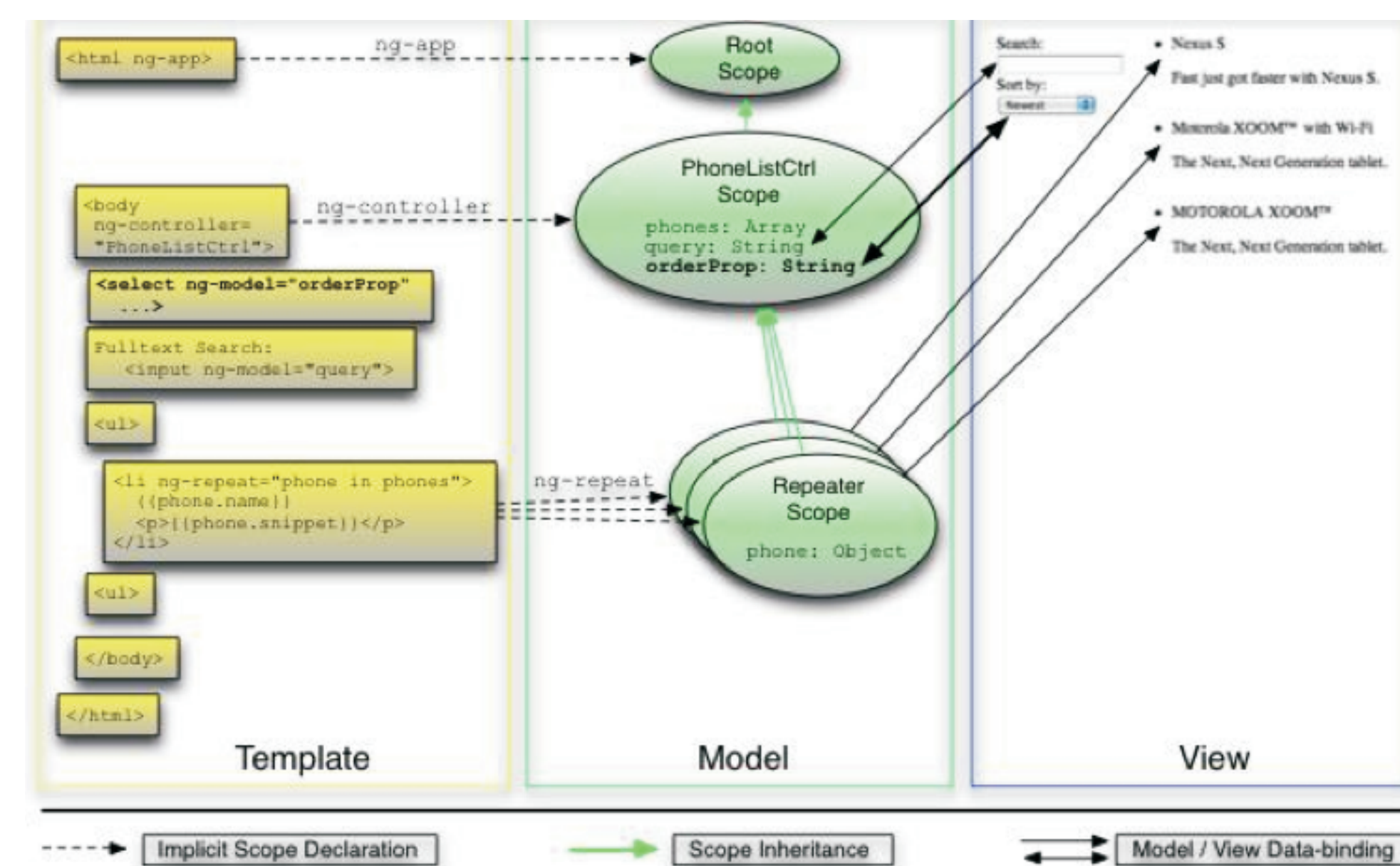
### Why AngularJS?

HTML is great for declaring static documents, but it falters when we try to use it for declaring dynamic views in web-applications. AngularJS lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop. AngularJS is a full frontend MVC framework for JavaScript web applications. It was built at Google and provides a way to quickly **build large, single-page web applications**. Like jQuery, it is included into a page using the <script> tag, and is itself written in JavaScript.

```
<!doctype html>
<html lang="en" ng-app="healthApp">
<head>
  <meta charset="UTF-8">
  <title>HealthCare App Demo for SWAT4LS 2014</title>
  <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.2/angular.min.js"></script>
  <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.2/angular-route.js"></script>
</head>
```

Unlike jQuery, it can be used as a framework upon which an entire healthcare web application is built. It actually includes a minimal version of jQuery by default. To show this, we developed a healthApp demo – Patient Search module (while the same method can be easily implemented into wide range of clinical information system requirements, where end user is involved).

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



## Results

### 2.1. Angular JS and Healthapp Demo

We used the `ng-model` directive to bind a model variable to the HTML element that can not only display its value, but also change it. This method can be efficiently used when we need to search data from EHR, LIMS, or wherever there is requirement for efficient filtering of medical data. Figure 1. explains the difference between traditional one-way data binding and new approach in development, two-way data binding model.

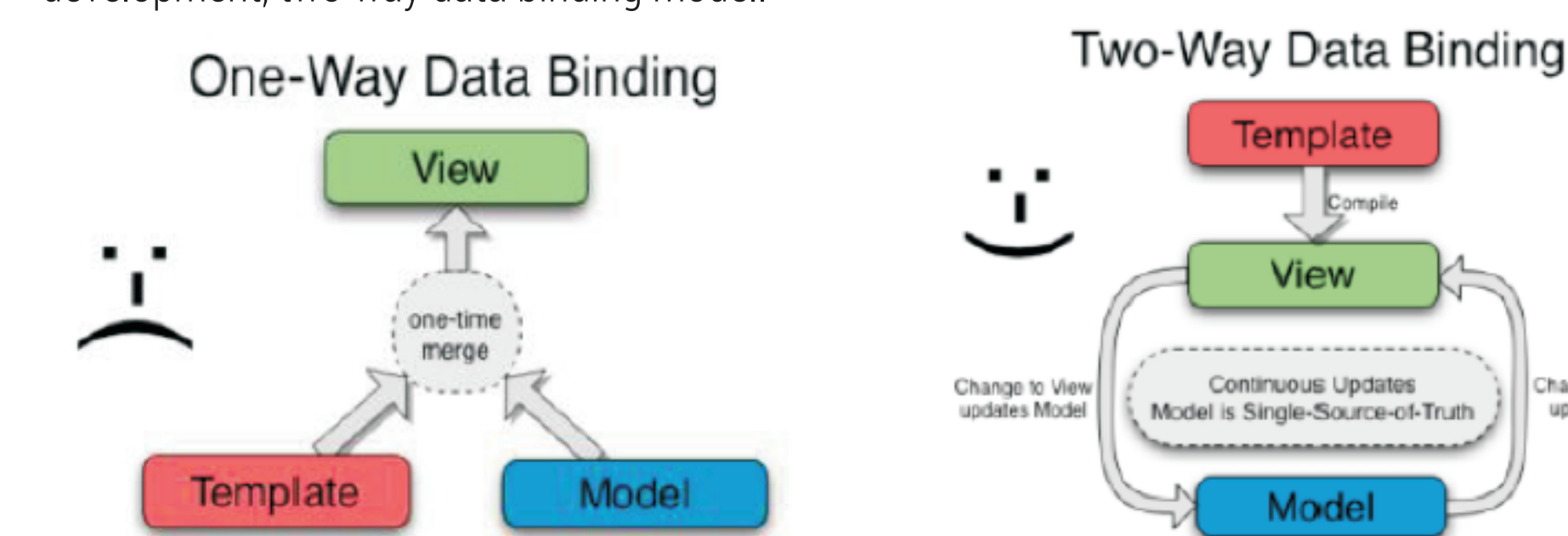


Figure 1. Difference between traditional one-way, and two-way data binding model

In our first example, we bind the `email` and `password` model variables to a couple of HTML elements. When the page is loaded, the value of the `input` elements are initialized to those of the respective model variables and whenever the user types something in an `input`, the value of the model variable is modified as well (two way).

```
<input type="email" class="form-control" id="email"
placeholder="Enter your email" ng-model="email">
</div>
</div>
<div class="form-group">
  <label class="control-label col-sm-2"
for="pwd">Password:</label>
  <div class="col-sm-6">
    <div class="input-group">
      <span class="input-group-addon">
        <i class="fa fa-key fa-fw"></i></span>
      <input type="password" class="form-control" id="pwd" placeholder="Enter your password" ng-model="pass">
    </div>
  </div>
```

## 2.2. Data Filtering

Filtering of data is very simplified: We use `filter` command to define which variable will be used as a filter parameter, while sorting is enabled by `orderBy` command. healthApp Demo is an example of a SPA (single page application), where we are using Template feature – meaning that only necessary part of HTML is rendered. Complete coding of our `patient_list.html` page is provided below:

```
<div class="container" ng-controller="listaController">
  <form class="form-horizontal" role="form">
    <div class="form-group">
      <div class="col-sm-6">
        <label>Filter:</label>
        <input type="text" class="form-control"
placeholder="Filter" ng-model="query">
      </div>
    </div>
    <div class="form-group">
      <div class="col-sm-6">
        <label>Sorting:</label>
        <select class="form-control" id="sel"
ng-model="orderPatient">
          <option value="name">Name</option>
          <option value="address">Address</option>
          <option value="age">Age</option>
        </select>
      </div>
    </div></form><br/>
    <label>Patient list:</label>
    <hr/>
    <table class="table table-striped table-bordered">
      <thead>
        <tr>
          <th>Profile</th>
          <th>Patient Name</th>
          <th>Diagnosis</th>
          <th>Therapy</th>
          <th>Address</th>
          <th>Age</th>
        </tr>
      </thead>
      <tbody>
        <tr ng-repeat="p in patients | filter:query |
orderBy: orderPatient">
          <td><a href="#/sm-{{p.id}}"></a></td>
          <td></a></td>
          <td><a href="#/index/{{p.id}}">{{p.name}}</a></td>
          <td>{{p.disease}}</td>
          <td>{{p.therapy}}</td>
          <td>{{p.address}}</td>
          <td>{{p.age}}</td>
        </tr>
      </tbody>
    </table>
```

Output of the Patient Search or Filter command is shown on the next figure.

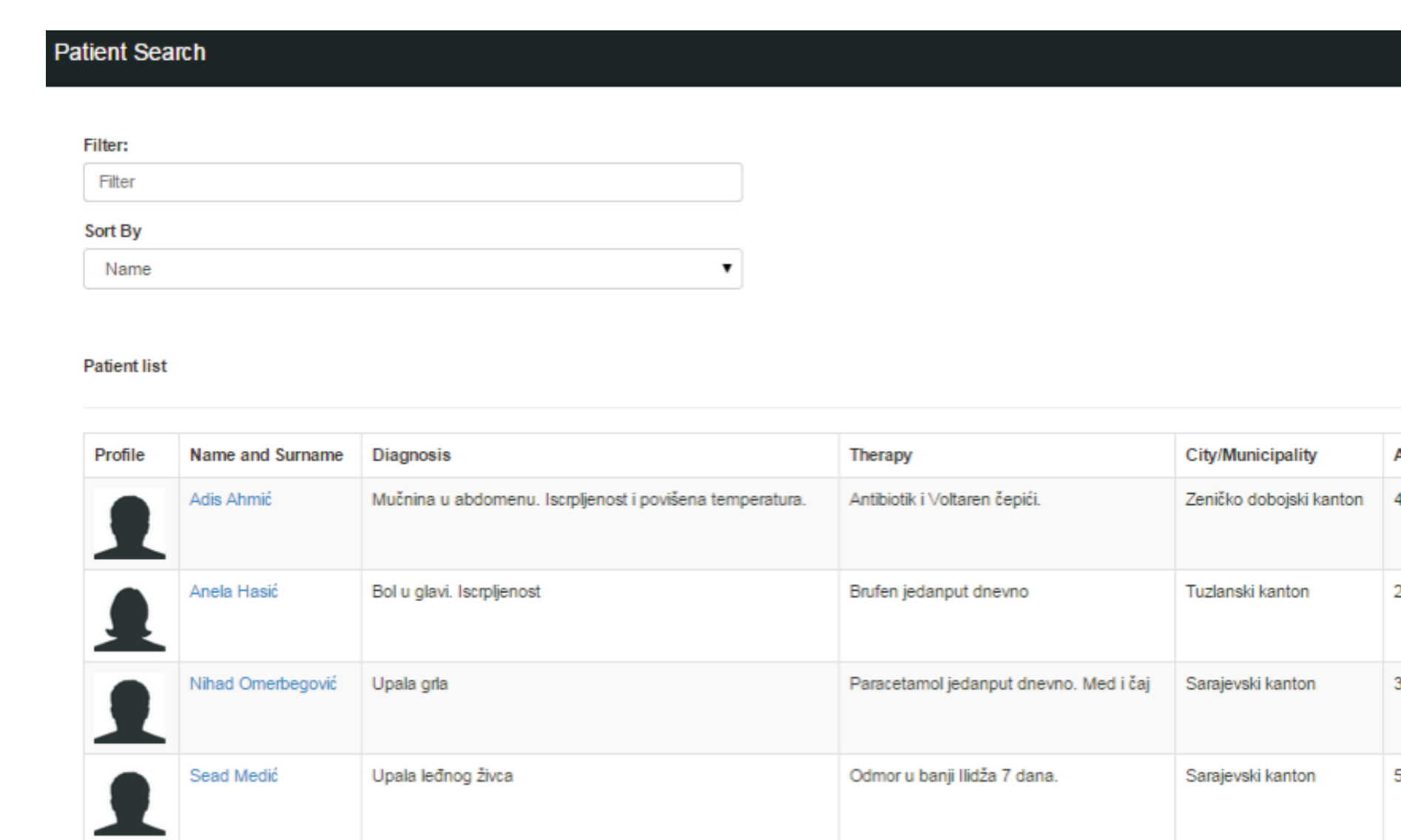


Figure 2.

shows the output of the Patient record search and Filtering of data, based on any criteria. As soon as user starts typing specific term or a word (for ex: disease), results are instantly displayed on the screen.

## 2.3. Usage in Practise

Two-way data binding method can be efficiently used in developing Sample Tracking or Patient Search modules of a web-based LIMS [3]. This method could be also implemented to reduce paper work in a laboratory.

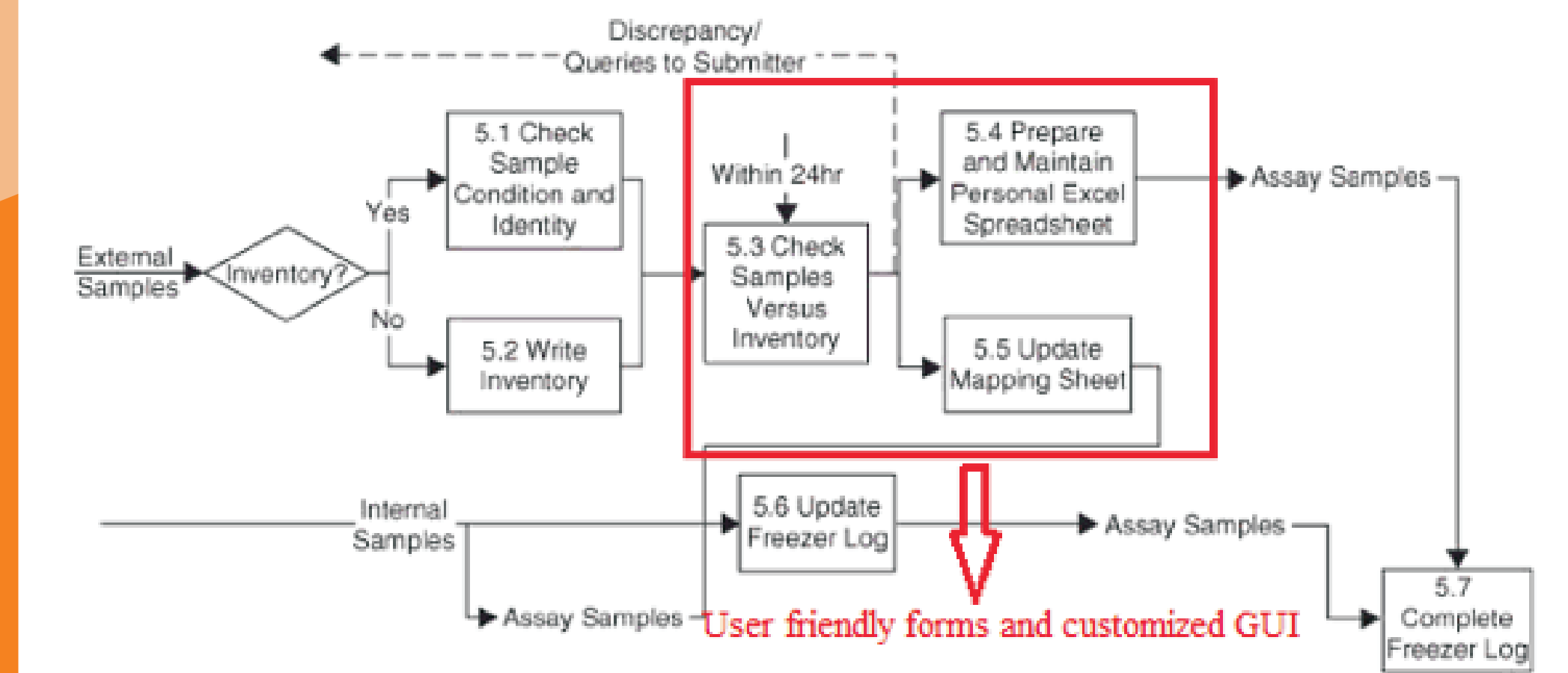


Figure 3. Standard Sample management process in a LIMS

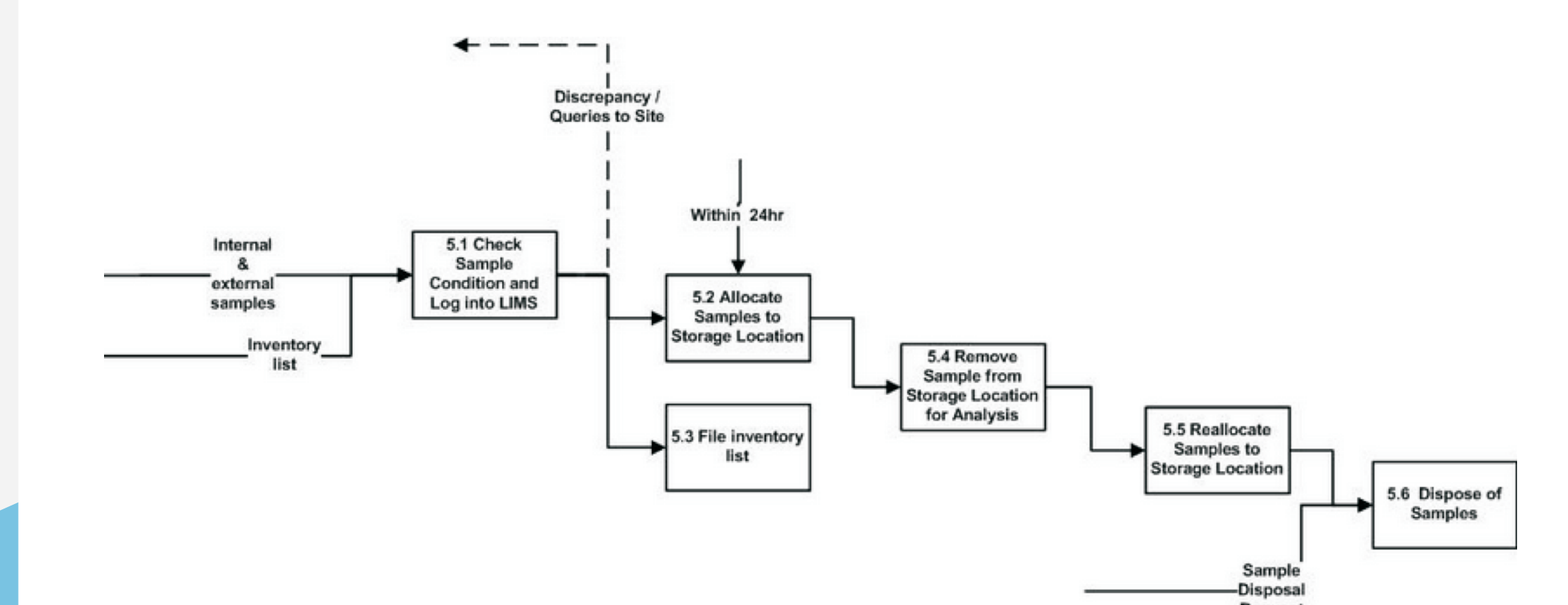


Figure 4. Optimized Sample management process in a LIMS

## Conclusion:

Data interoperability and data quality are some of the main issues in modern clinical information systems today. The main aim of our paper was to suggest developers of clinical information systems an easier method for developing standard features like search for patients, or sorting of sample results by using AngularJS framework two-way data binding model. Also, we intended to show how combining of these techniques and standard web languages like HTML and JavaScript can have effect on better end user experience and increased performance. In this paper, we showed how:

- Optimized data search and filtering can improve user experience with EMR or LIMS apps.
- Drastically reduced programming code lines, comparing to ordinary JavaScript and jQuery.



## References

- [1] Christelle Despont-Gros, Henning Mueller, Christian Lovis, Evaluating user interactions with clinical information systems: A model based on human-computer interaction models, Hôpitaux Universitaires de Genève, Service d'Informatique Médicale (SIM), 21, rue Micheli-du-Crest, CH-1211 Genève 4, Switzerland
- [2] Steve Rayburn, Director of Training and Validation and Dr. Christine Paszko, Vice President of Sales and Marketing Accelerated Technology Laboratories, Inc. West End, NC, How a LIMS Can Save Your Laboratory Money and Enhance Data Quality
- [3] Chung Chow Chan, Herman Lam, Xue-Ming Zhang, Practical Approaches to Method Validation and Essential Instrument Qualification, 1st Edition, Wiley (October 4, 2010)
- [4] How to build AngularJS phonecap app Bootstrapping, [https://docs.angularjs.org/tutorial/step\\_00](https://docs.angularjs.org/tutorial/step_00)
- [5] JavaScript Framework Comparison, <https://www.airpair.com/javascript-framework-comparison>
- [6] Two-way data binding in AngularJS framework, <http://www.angularjshub.com/examples/basics/twowaydatabinding/>  
<https://docs.angularjs.org/guide/databinding>
- [7] [www.corelims.com/benefits](http://www.corelims.com/benefits)