Genesys Lab Application

Project Engineer: Ryan Pastelak
Bachelor of Science in Computer Science
University of Evansville

Project Sponsor: Jim Heidecker, Ascension Information Services
Project Advisor: Dr. Hwang
25 April 2018

ABSTRACT

Ascension Information Services is the technical portion of Ascension. Ascension is the largest healthcare system in the United States. As part of the healthcare system, labs are used to test fluid samples. Part of the test process requires the lab workers to record the results of each test on paper and then file those papers. Certain types of lab documents have different rules regarding how long they each must be kept. These documents take up lots of space over time. To improve the storage and efficiency of each lab test, an online web tool is being built for them. It allows lab workers to record the information of each test and stores the document for them along with tracking the progress of each individual fluid test.
ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation towards my project sponsor, Jim Heidecker, and AIS senior developer, Richard Livingston, for taking the time to find a senior project for me and their support throughout the whole process. I would like to thank Dr. Hwang as both my project advisor and academic advisor for her valuable advice.

LIST OF FIGURES

Figure 1: System Overview
Figure 2: MVC Pattern [2]
Figure 3: Database Tables
Figure 4: Home Page
Figure 5: Search Screen
Figure 6: Record Submission
INTRODUCTION

Ascension is a Catholic-based healthcare organization whose headquarters are located in the Greater St. Louis metropolitan area. It is the largest non-profit health system with 2,500 sites of care in 22 different states and the District of Columbia. With more and more hospitals joining the Ascension network and the current size of the company, structural changes take a while to accomplish. With the structural changes that have occurred recently, the idea of One Ascension was formed. With One Ascension, the idea from a development standpoint is to create the applications that can be reusable by hospitals and have developers create applications at a higher level in order for them to be distributed as needed to hospitals without redevelopment of a whole new application.

With this idea, hospitals had meetings with development teams to decide what applications could be reused and what applications could be written to support this idea. The Genesys fluids testing lab has a slow manual process with additional storage costs. With large amounts of storage space being used to file records and the slow process of having to fill paperwork out and do every part of the testing process manually, cost requirements are rising with the increasing size of the company and the volume of people who come to use Ascension’s services. The other problem facing the Genesys testing lab, is a lack of quality controls. The sample was not being tracked along the process and if mistakes or errors arise, it is extremely hard to pinpoint where they occurred.

Developing a new web tool that runs on the Ascension network is a solution to these problems. The Genesys Lab Application is a web tool that will allow lab technicians and other authorized individuals on the Ascension network to be able to quickly key in important information and log files that are created by the testing equipment, which is written to a database. This web application will use many security features already in place in order to guarantee the security of the data.
PROBLEM STATEMENT AND BACKGROUND

Ascension is a healthcare organization that provides personalized care to all individuals. Although Ascension has had numerous applications written to decrease the amount of time it takes for lab results to be completed, there are still many applications that can be written to decrease the time and space it takes to get lab results. For example, when an individual goes in to get bloodwork done, a nurse takes the patient into a backroom and withdraws a few vials of blood. Those vials are then sent to a testing lab. In this testing lab, machines are used by lab technicians to examine the blood sample. During the testing, lab technicians are given paperwork to manually fill out and record results based on the tests they are performing. These pieces of paper are then taken to the administration staff and stored with all the other lab results for that day. To further the problem, the lab test is unknown by the hospital at what stage of testing it sits in, making it hard to estimate the time to completion. This lack of quality controls makes it harder to track problems if a result were to get lost or if tests had a human error.

The Health Insurance Portability and Accountability Act (HIPPA) has requirements for the privacy and security of the application. This policy dictates that the application must follow national standards for protecting health information and safeguards to ensure that the confidentiality, integrity, and availability of the information are followed. Based on these and other government policies, records must be kept for the patient for the lifetime of that the patient lists Ascension as the primary care provider or given as the patient and HIPPA policies dictate.

The application was originally written in Lotus notes and functioned for a while, but it had stability issues and was unreliable. Due to the instability of the application, Ascension went back to using paper. This created a demand for a new application to be in place so that manual records would not need to be stored and a manual process would not be in use. This project will be tested by the Genesys Hospital in Grand Blanc, Michigan. From here, the application will be packaged up and easily distributed to other hospitals on a needs basis.
SPECIFICATIONS AND REQUIREMENTS

To gather requirements for the new application and determine all the specifications of the Genesys lab application, one of the lead developers of that hospital had a meeting with the members of each department to determine a list of applications that the hospital would need. It was determined that we would like to replace as many manual and paper-based processes as possible in the One Ascension initiative. One of the ideas brought forth by a lab technician was to replace the paper process that they were doing for all the hospitals lab specimens.

Security Requirements

Applications are required to authenticate users that try to hit any page of the application. Authentication in this case is a process of identifying objects, services and people who want to interact with the application using credentials. By providing this identity to the application, the application is able to safely interact with verified users, administrators, files, and databases. In this specific application, we are required to use Windows Active Directory Domain Services. In Windows Active Directory, we store cryptographic keys in a secure central location tied to the user’s login and be logged into the Ascension secure domain frame. In using Windows Active Directory and Windows computers, we get additional authentication protocols through Kerberos that allow us to secure the whole process of the application. Included in this authentication will be two groups: Administrators and users. To be added to either of these groups, a help desk ticket must be submitted along with the necessary approvals from managers to use the application. Once a user is created, an administrator must add that user to the acceptable listed of people who can submit fluid sample records. If the user is not added to that list, then they will have read-only access.

Functionality Requirements

The application is required to be written using ASP.NET implementing the Model-View-Controller (MVC) framework. In this framework, bootstrap is used for design, Windows Active Directory for authentication, and ADO.NET to work with SQL databases.

One of the biggest problems with having so many manual records can be going through all of them to find an individual’s blood work. By default, when the application first opens up, they are taken to a landing page with the last 5 records they submitted, the status options available, and the list of users who last submitted a record. Depending on the user’s access, they may have access to a page that allows administrators to add and delete users who can submit lab samples. This page allows them to view and search through all records ever submitted. When an individual specimen/record is selected, the user is brought to a detail screen of what was logged and the time at which it was logged at. Administrators should also be able to edit the details of all fields in the application and view the log files while the application is running.

The next biggest piece of the application is the processing of data. There is an input form that requires a department, the way it was tested, certain tools used to test it, and multiple log sheets where the results from the actual tests are stored. This data is interpreted and sent across
the network to a server, where it is stored in a way that allows other applications and tools to easily access it when necessary.
Following the Ascension Information Services standards for development, the application itself is designed around the model-view-controller (MVC) architecture as shown in Figure 2 below. The model is the data piece of the application and is used to directly manipulate the database by receiving specific calls from the controller and will have built in basic data validation for the application. The view is the user interface (UI) of the application and renders a usable output for the user. The controller is backend of the application. The controller is responsible for taking user requests and being the logic behind the request. The controller is also directly responsible for sending changes to the view so that it can update as needed [1]. In Figure 1 above, the application described in this paper is responsible for the three blocks in the middle column. The application is responsible for test submission, logging the errors and results, and pulling information from the server to be written to a server.
**User interface (front-end)**

The view is mainly written in a mix of HTML and Razor. Razor allows us to embed server-based code so that it can create dynamic content on the fly. It uses a basic Genesys Hospital template with a Bootstrap implementation that includes the CSS and JavaScript necessary for the application. JavaScript assists in basic user-side error checking so that there are fewer call backs to the server. CSS with Bootstrap is used to make the application and data easier to read for the users.

The home page is an overview of different features of the application with the last 5 records they submitted, the status options available, and the list of users who last submitted a record. The navigation bar works as part of the layout template so that it does not need to change or regenerated for each page. The individual users authorized to submit data and view data from the Genesys Lab Application have a Home Page menu item, a Submit Record menu item, an About menu item, and a Search Records menu item on the menu bar. The Submit Record menu item takes the user to the fluid test submission page. This page has input fields for the text files that need to be pulled for the individual user along with basic user identification information and the lab testing done for that individual record. After submission, the user is redirected to a view of the submitted information and an acknowledgement that the submission was successful. If anything is wrong on review, the lab technician can select edit and edit the record. The Search Records page allows users to search for a record based on patient, date, type of test, and for administrators, the lab technician that submitted the result. The lab tests can be selected to view...
them, and even edited, if necessary. The about page offers basic instructions to the application and answers to some common questions about the application.

**Backend (Server)**

The controller piece of the MVC architecture takes care of the backend logic. In this piece is the manipulation of the data that it will interpret and send to the model. This piece also controls the behavior of the view and how it will interact with the user. In a sense, it is the connection between the database and the view the user is seeing.

The backend is responsible for communicating with security modules and ensuring all requests are properly handled. To protect the data transfer from the front-end to the back-end, we use Windows Authentication (sometimes called Integrated Security). Windows Authentication is specified in the configuration file using the key words Integrated Security. This removes any need for specifying direct login information to the application. Instead, users are authenticated by Windows. This means that to access the resources on the database or server, the user must have been given permissions through Windows users or group rights. These rights are requested through the support team and must be given permission by management before being added to the groups. Windows authentication controls what the user is allowed to see on each individual page that differentiates between what an administrator can do and what an individual user can do.

The controller piece of the application is charged with interacting with the database. The controller issues commands to the database or retrieves records for the user to view. It interprets basic SQL commands to the databases such as Select, Insert, Edit, and etc. Basic JavaScript validations work on the front-end, but for the back-end, the project engineer uses a more heavily implemented validation system that checks queries for errors, data that was not supposed to be sent, or data that should have been sent such as not filling in required fields when submitting a fluid test result.

**Database and Model**

The model is directly in charge of managing the database and validating certain requests. The database will have authentication from the application and users, allowing this specific application to submit data and only certain users to access data based on Windows Authentication Groups and policies. When the controller interacts with the database, it interacts with the ADO.NET Entity Data Model (EDM). EDM is used to describe the structure of the database and does not need to know what form it is stored in. This model allows developers to focus on the conceptual piece of the database using conceptual schema definition language (CSDL) to define what each of the conceptual models look like. Simply put, the controller will issue commands based on the EDM structure that will have its own checks and validations to ensure that data integrity constraints are held.

For the database, the project engineer uses SQL server with certain permissions granted to each user group. Specific users have read-only access to the data as they only need to view.
Those users who need to enter records are split into lab technicians and specified administrators for the application. Lab technicians will have read/write to data submission fields only, while administrators will have read/write to every field in the database.

When looking at the data in a database, the number of individual records is not large, but the number of records in this environment can add up. To deal with this, there are stored procedures that make the larger transactions transition seamlessly with little to no overhead. On top of this, the project engineer separated tables out to be referenced by the main table following fifth normal form (5NF), meaning that there are no wasted join dependencies that are not follow from key constraints.

Due to constraints in privacy rules from HIPPA and the competitive nature of the business, we cannot give exact contents of the entire database. The contents of the database that will not shown include the contents of the files that get picked up from the laboratory testing. Figure 3 shows tables that log what the application does, tracks the state of each individual fluid sample, and keeps track of what was submitted by the lab technicians through this application.

![Database Tables](image)

**Figure 3: Database Tables**
Alternate Designs

The application was going to originally pull some of the less sensitive data from the files and store it in a separate table with the lab request id as the key. This data was deemed to legally still be under HIPPA policies that allowed only certain users to see it. This table and its corresponding models were then changed, and all file data was stored separately. Part of the process that was added to this application, was the ability to track the lab sample throughout the lab testing process so doctors and nurses could see the progress of the lab sample and given patients who call in a more accurate lab testing completion date. This added record types that could be submitted to the log and added additional steps to the system overview.
Results

With further testing pending, the application is sitting on a dev server. The basic requirements of the application have been met and future plans for changes to add to this application have already been proposed. The majority of the initial design plans were adequate to implement this application with some basic legal and process modifications along the way. Items are still being added to the about page including questions that will come up during the testing process so that users can get most of their answers without waiting for a support ticket.

The main page has an overview as shown in Figure 4 above that gives the user a basic understanding of how the application works. The record submission page shown in Figure 6 allows lab technicians to quickly submit the results of their testing. This submission allows quality controls through lab sample tracking and reporting. The search page shown in Figure 5 below allows users to be able to lookup previously inserted records and click on them to be edited if need be. The lab users’ page is visible to administrators only and allows administrators to add users to the group if they want them to be able to submit records. The about page is still being added to as stated above, but gives a basic overview of the application and common questions and answers.
### Recent Lab Samples

<table>
<thead>
<tr>
<th>Table ID</th>
<th>Department ID</th>
<th>Log Sheet One</th>
<th>Log Sheet Two</th>
<th>Log Sheet Three</th>
<th>Log Sheet Four</th>
<th>Log Sheet Five</th>
<th>Special</th>
<th>Slides</th>
<th>Box Number</th>
<th>Status</th>
<th>Request Creation Date</th>
<th>Request Creator</th>
<th>Request Last Update Date</th>
<th>Request Updater</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Rad - Mam</td>
<td>5</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>False</td>
<td>False</td>
<td>2</td>
<td>Received</td>
<td>2014/01/04 22:50:45</td>
<td>pastela</td>
<td>2014/01/04 22:50:45</td>
<td>pastela</td>
</tr>
<tr>
<td>6</td>
<td>Lab</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>False</td>
<td>False</td>
<td>1</td>
<td>Received</td>
<td>2014/01/04 22:50:58</td>
<td>pastela</td>
<td>2014/01/04 22:50:58</td>
<td>pastela</td>
</tr>
<tr>
<td>7</td>
<td>Rad - US</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>False</td>
<td>True</td>
<td>1</td>
<td>Received</td>
<td>2014/01/04 22:51:10</td>
<td>pastela</td>
<td>2014/01/04 22:51:10</td>
<td>pastela</td>
</tr>
<tr>
<td>8</td>
<td>L &amp; D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>False</td>
<td>False</td>
<td>2</td>
<td>Received</td>
<td>2014/01/04 22:51:45</td>
<td>pastela</td>
<td>2014/01/04 22:51:45</td>
<td>pastela</td>
</tr>
<tr>
<td>9</td>
<td>L &amp; D</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>False</td>
<td>False</td>
<td>2</td>
<td>Received</td>
<td>2014/01/04 22:51:57</td>
<td>pastela</td>
<td>2014/01/04 22:51:57</td>
<td>pastela</td>
</tr>
<tr>
<td>10</td>
<td>L &amp; D</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>True</td>
<td>False</td>
<td>2</td>
<td>Received</td>
<td>2014/01/04 22:52:33</td>
<td>pastela</td>
<td>2014/01/04 22:52:33</td>
<td>pastela</td>
</tr>
<tr>
<td>11</td>
<td>MPU</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>False</td>
<td>False</td>
<td>2</td>
<td>Received</td>
<td>2014/01/04 22:54:34</td>
<td>pastela</td>
<td>2014/01/04 22:54:34</td>
<td>pastela</td>
</tr>
<tr>
<td>12</td>
<td>Rad - CT</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>False</td>
<td>False</td>
<td>2</td>
<td>Received</td>
<td>2014/01/04 22:54:46</td>
<td>pastela</td>
<td>2014/01/04 22:54:46</td>
<td>pastela</td>
</tr>
</tbody>
</table>

© 2016 - Ascension Information Systems

**Figure 5: Search Screen**

**Figure 6: Record Submission**
CONCLUSION

The model-view-controller framework allows us to easily meet all the specifications necessary for the Genesys Application. When combined with the ADO.NET Entity Data Model, the application can be easily distributed amongst different hospitals quickly. This will also allow easy modifications to the view if we need to switch the application platforms and put it on a tablet instead of a desktop computer.

The view part of the application is an easy to use, but flexible interface that meets the lab technician’s and the administrator’s requirements. It allows them to input file names for the application to grab the lab result text files and read in the contents. There is a search screen that allows them to view/edit their submissions and possibly other submissions based on their security access, removing the amount of time it takes to look up records. The model part of the application allows the records to be stored in a database for the required amount of time and will be backed up with standard Ascension backup policies. The database holds constraints necessary to ensure all data integrity is intact. The controller ensures that security standards are met, and that the application is properly restricted. It is responsible for checking the data on input and communicating with the Razor syntax to dynamically put data on the screen for the user to see. It communicates with the model and tells it what to update to ensure that the records are properly submitted, and the view displays its content properly.
REFERENCES


BIOGRAPHY

The project engineer, Ryan Pastelak, is a senior computer science major with minors in mathematics and engineering management at the University of Evansville. Ryan has strong skills in C# and a good understanding of .Net Framework. He plans to start his career as an embedded software engineer working with communication systems.