

Integrated Analytics

Ben Jasinski

Madison Klein

Michael Niemeyer

Felipe Silva

Data Verity

David Flammer

June 20, 2017

I. Introduction

Our group was assigned to work with Data Verity, a company that specializes in the management and analysis of data for multiple financial institutions. The company provides various statistical analysis with its clients' data and presents the analysis with graphics and visual tools that aid in managerial and investment decisions. Each customer's data is stored in a database, but the information is displayed and accessed online through Data Verity's web application. The project "Integrated Analysis" focused on developing a more polished interface for easier and more effective usability, as well as presenting the data in a cleaner and more informative way. Although the web application already possessed some interfaces and visual representations of the data, the client wished to further integrate the powerful data analysis tools with the visual capabilities.

The project broke down into two major tasks. The first of these tasks revolved around the development of specific data visualization tools. The web application provided by Data Verity presented the user with numerous visual representations of their respective database. The task, however, was mostly focussed on developing the hierarchical and historical charts. It aimed at creating comprehensive items where the user could view and interact with these two specific representations. The "Hierarchical Viewer" for example had a very robust sunburst diagram which contained information that ranged from the different branches of a financial institution to specific transactions made daily. The "Historical Viewer" on the other hand contained mostly line charts presenting the selected data.

The second major task for the project involved expanding upon the statistical tools and functionality provided in the table manager system. This system allowed the user to view its data in a standard table form and run statistical analysis on it. The group was tasked with developing additional statistical options (sum, standard deviation, etc.) for users to apply to the necessary components of their data.

II. Requirements

Functional

The functionality requirements applied to both major tasks. The functionality of the visual representation items revolved around the user's ability to interact with the representations. Our clients' web application already contained a sunburst diagram and a line chart, nonetheless, integration was necessary for users to understand how to navigate them and what information they were actually looking at. To achieve this outcome we were required to:

- Include a panel for the user to edit the settings of the representations.

- Include a panel that held the actual representation and minor information about it.
- Include a third panel that displayed more specific details about elements shown in the representation.

The settings panel was to contain drop down menus for information such as which period and which group to segment by. These drop down menus were expected to change and react as the form was being filled in. Once enough information was put in, the panel should send the settings to generate the specific graphical representation (sunburst diagram or line chart).

Both the hierarchy and history representations should have the ability to interact through mouse actions such as:

- Hovering over specific elements to bring up minor details about the element in both representations.
- Double clicking on elements to navigate the hierarchy representation.
- Clicking in both representations to send the element id to the details panel.

The details panel should be responsible for displaying all the information about the elements. When it receives a specific id, it must send a request for the data and display the information it receives back. The details panel should also host the option for the user to view the opposing visual representation for that element, i.e. if in a history viewer, the user can request to see the hierarchy viewer for a specific element and vice versa.

The second major functionality requirement was in regards to the ease and accessibility of the statistical options added to the table manager system. Upon viewing the table manager system, the user is able to select the specific rows to analyze. In order to access the statistical tools, the user must click the drop-down menu on the respective column category they desire. There, we were expected to add a statistical analysis button which should be composed of two options, to analyze only the selected rows or all rows in the table manager.

A pop-up window should then be displayed to the user specifying the different statistical information. This window should provide the user with statistics of the selected data including average, sum, minimum, maximum, population standard deviation, and sample standard deviation.

Non-Functional

This project was primarily written in HTML5 and JavaScript using the ExtJS framework. The new dynamic graphs utilized for data visualization were written using D3.js data visualization tools. The communication to the servers to access data and files for the various

interfaces and tools was achieved using AJAX. The entirety of the development and testing of these interfaces and tools was done on a server that the client setup for field session.

The team's software must be able to run with other existing software on the client's server. Finally, due to the fact that Data Verity's clients are primarily financial institutions, the software for this project must be secured and would only run after a successful login.

III. System Architecture

Figure 1: Client/Server Interaction and Components

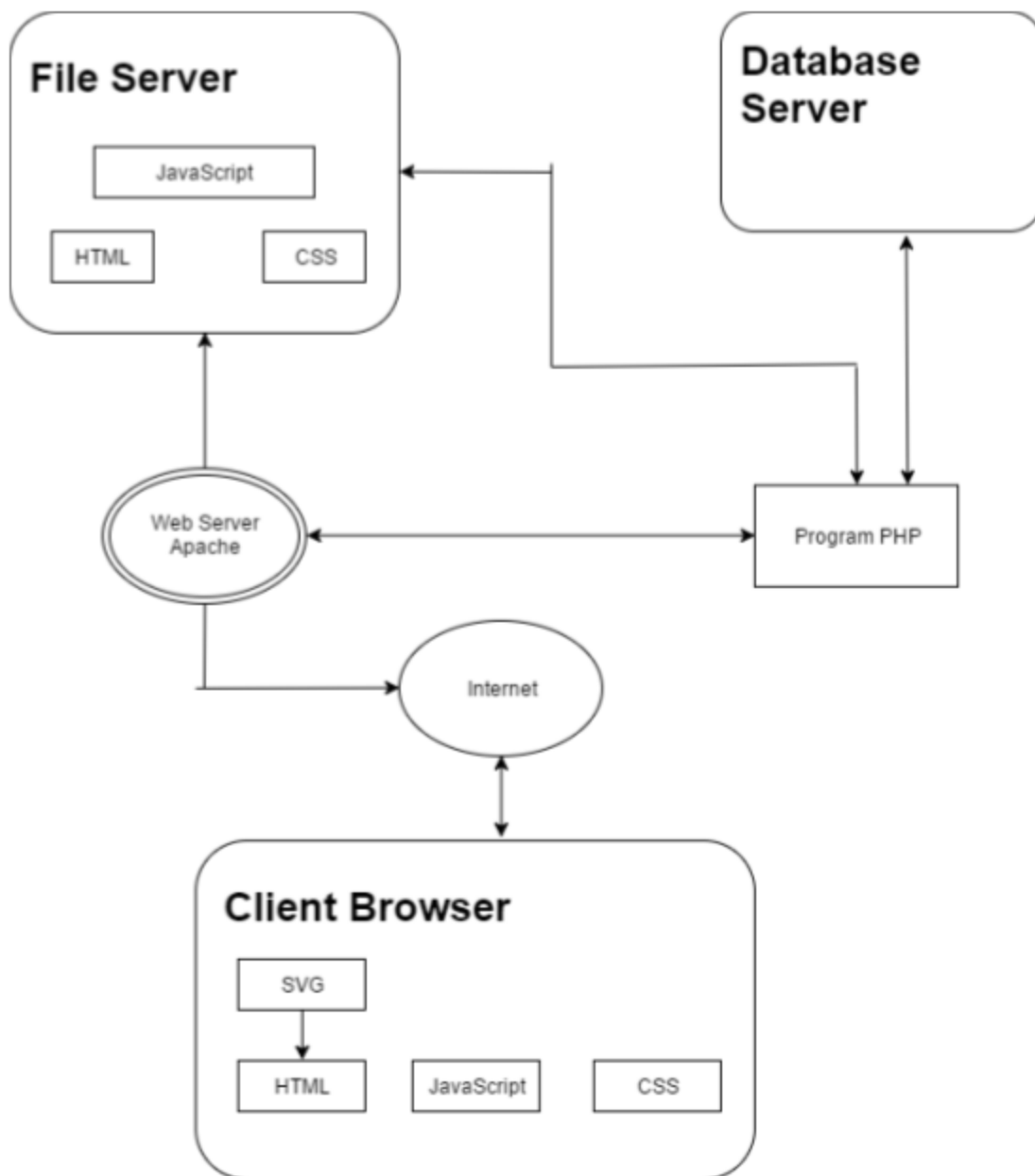


Figure 1, describes the overarching system architecture for the application that the team worked with throughout the duration of the project. The majority of the team's work remained within the Client Browser section of the architecture. This was due to the team working primarily with JavaScript front-end interfaces which the users utilize. The team interacted very little with the back-end database portion of the system.

Figure 2: Example Visual Representation User Process for Hierarchy Viewer

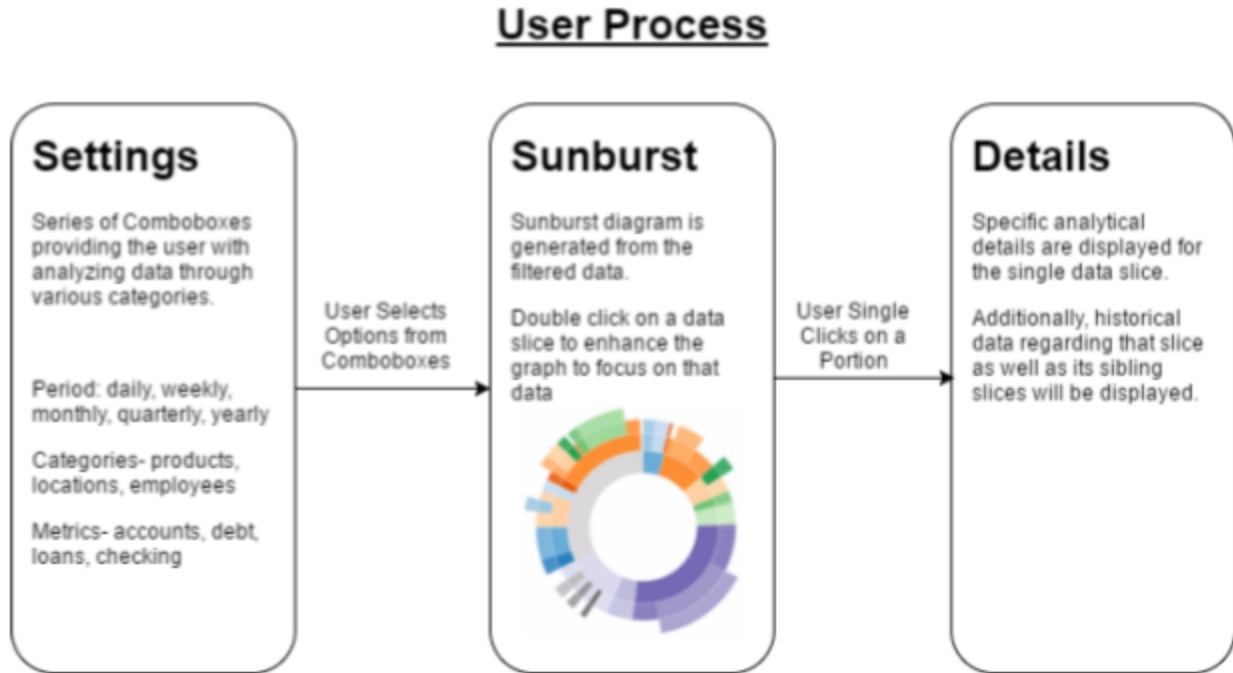


Figure 2 focuses on much fewer technicalities than Figure 1. Figure 2 represents the process that the user experiences as they navigate the hierarchical graph. The diagram describes what the user interacts with in each phase of the hierarchy chart as well as what is provided in response to their input. Additionally, Figure 2 closely describes how the user interacts with the historical chart.

IV. Technical Design

Given the scope of the major goals achieved throughout the duration of the project, it is interesting to note just how much of the overall project depended upon such a minor detail as the tree branch system. This system, utilized within the settings panel for both the hierarchy and history charts, is broken up into two subsystems (Figure 5 in Appendix). The first subsystem appears within the hierarchy chart in the form of a single selection tree in which the user would select something such as a product metric. The latter of these systems is a multi-selection tree that is utilized within the history viewer chart which requires multiple data inputs in order to offer a valuable comparison.

The reason behind a large majority of the project hinging behind this seemingly minor detail is the fact that the majority of the data is handled by tree branch systems. The data that populates the trees depends on previous selections in the settings and impacts options for later selections. The interaction of the combo boxes and tree systems is what supplies the settings panels with enough information to send to the server to receive a full hierarchy or history graph. Due to these systems being the backbone of the hierarchy and history, a large emphasis was placed upon their development in the early stages of the project.

Perhaps the most difficult aspect developing the tree branches was their initial value metrics. Real world clients will likely save a specific graph for the purpose of revisiting it later on for further analysis. Due to this fact, the selected metrics must be converted to an efficient string composed of the various item ID's of each individually selected node of the tree. This string is stored within the database and is easily accessible at a later time.

When the request to reload the hierarchy or history chart is processed, the string key is fetched from the database and converted back into a readable form for the tree architecture to interpret. Once interpreted, the corresponding tree node is located and processed. If the node is not processed exactly as needed, it is likely that the visual representation of the tree will differ from the data representation. This sort of error would allow the user to mistakenly select a node that is already processed as a selected node, thus creating a duplicate in the system. If a duplicate is not caught and is fully processed, the entire hierarchy or history chart will ultimately become broken and unusable to the user.

V. Design Decisions

The team's project was characterized by adding functionality to an already created web application. Since the web app was already implemented, most of the technical design decisions were already made prior to the start of the project. The primary decision was to use ExtJS as the javascript framework for the client side work. The code for the existing web app uses ExtJS and the team kept consistent with this in order to easily incorporate the added functionality to the overall web application.

For the Hierarchy and History charts, it was decided that we should use previously existing components to create the full functionality. These items, as described earlier, consist of a settings panel on the left, a graphical display panel in the center, and a details panel on the right. Screenshots for both the hierarchy and history items are shown in Figure 3 and 4 (found in Section VI).

For the settings panel, the team modified an existing update form panel to include the specific functionality needed. This form panel consisted of several combo boxes that allowed the user to select the specific settings that they wanted to visualize. The additional three combo boxes that are shown in Figure 5 (in Appendix) were taken from existing code but modified by the team for a new specific purpose.

For the two different graphical displays, the team modified the existing Sunburst d3 diagram and the existing chart panel. The toolbar seen above the hierarchy representation in Figure 3 and the legend were added by the team.

For the details panel, we used an existing panel that formatted the information and modified the other components to correctly tell the details panel what to display. The only modification the team made on this panel was the addition of the button in the bottom right on Figure 3 and 4 that pops up the opposing visual representation of the specific element.

The statistics tool was also derived from portions of pre-existing architecture. The drop-down menu for the columns in the table manager already existed prior to the start of this portion of the project. These drop-down menus provided a compatible base to build off and allowed the team to simply add onto the pre-existing tools.

The actual pop-up calculations window also allowed the team to utilize already existing architecture from the back-end. Pre-existing statistical calculation tools from the back-end database architecture were also available. Once these tools were implemented, most of the remaining work was simply sending the correct data to the database and then interpreting the returned data.

VI. Results

Figure 3: Hierarchy Visual Representation Screenshot

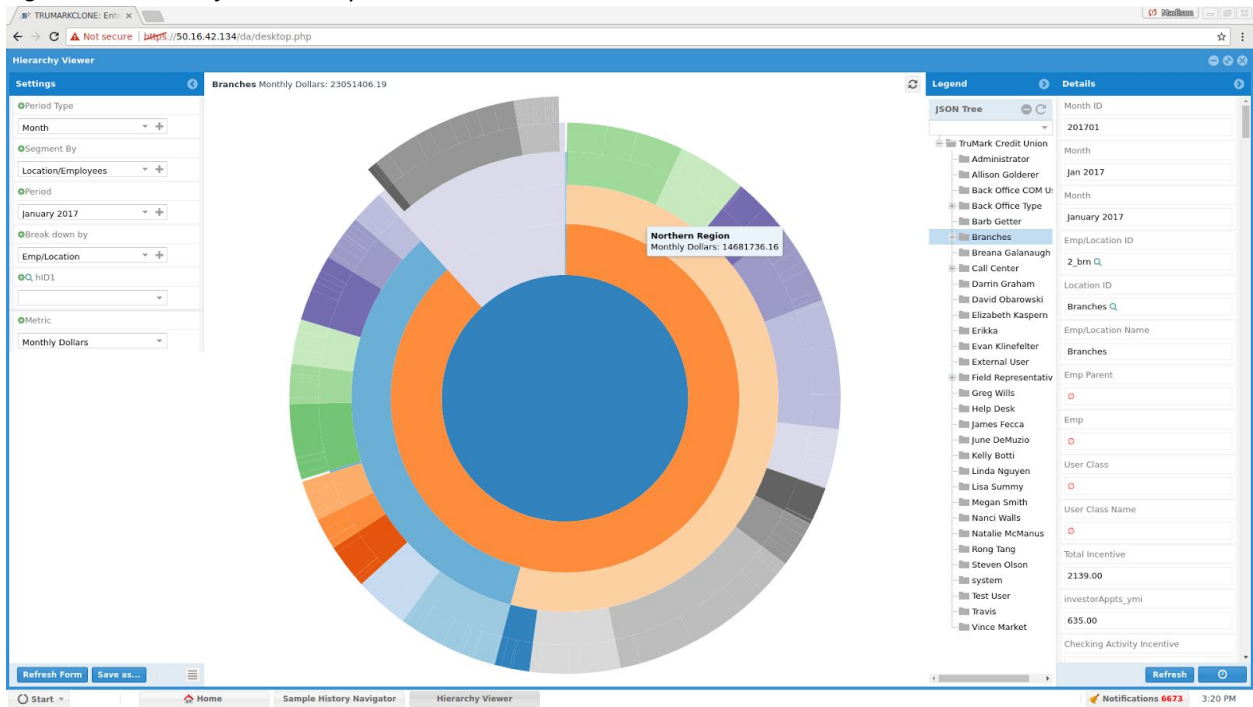
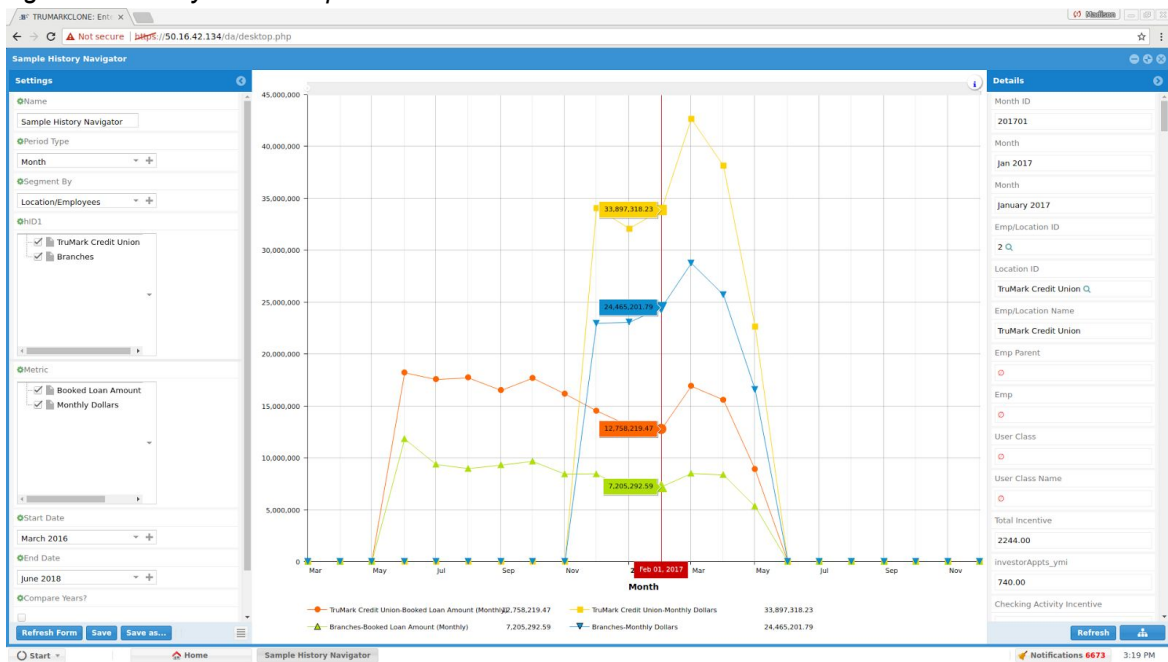


Figure 4: History Visual Representation Screenshot



This project consisted of three main overarching goals brought forth by the client. The first goal being the creation of an in depth hierarchical data chart (Figure 3) that allowed the

user to build the chart with the data they desired as well as allowing them to navigate the chart to further analyze the data. The second goal was the inclusion of historical details of certain data through the generation of a historical line chart (Figure 4). Both of these charts rely upon an in depth tree combo box selection system that houses the data for the user to select from (Figure 5 in Appendix). The third goal was to implement a statistical analysis option to the data table system that provided statistical analysis of the selected data (Figure 6).

Figure 6: Statistical Analysis Tool Results

| Average | Sum | Max | Min | Std. Dev. (pop) | Std. Dev. (sample) |
|-------------|---------------|-------------|----------|-------------------|--------------------|
| 5124.224843 | 1779978362.59 | 12021064.13 | -6539.73 | 39161.46537145699 | 39161.52174090976 |

Figure 7: Redesigned Grouping Tool

| Column Name | Data Type | Column Grouping |
|-----------------|-----------|-----------------|
| Amount (amount) | DateTime | Quarter |

| Column Name | Aggregate Function |
|---------------------------|--------------------|
| Booked Loan? (bookedLoan) | Count |

These three goals were all implemented to the pre-existing system and tested with various different user inputs as well as data inputs to ensure the implementations were fully functional. These implementations were accomplished slightly faster than anticipated,

leading to an additional fourth goal to be added regarding cleaning up the UI system for a data grouping tool (Figure 7). This tool had the purpose of allowing the user to group specific factors together to further analyze the relationships between the factors, with the final result resembling Figure 8 (in Appendix). This goal was accomplished before the deadline thus leaving the team with no incomplete tasks.

Nevertheless, the main remaining issue with the team's implementations was in regards to browser functionality. Issues are encountered when running the system on non-chrome browsers as well as linux based systems, but this fault is due to the pre-existing system that the team was building upon, not the team's implementations themselves.

Now that the project is complete, it is rather easy to forget that this entire project was meant to be a learning experience. At the beginning of this project, every team member was essentially new to the JavaScript/Extjs language. Major conceptual difficulties impeded the team's progress early on. Perhaps the most difficult concept to overcome was that of the asynchronous nature of JavaScript, meaning the code is not processed in a linear order such as most object oriented languages. For example, if a server call is processed in Java, the commands following this call must wait until the call returns, but in JavaScript, the rest of the code will be processed without waiting for the return. With time, the team was able to overcome this new concept and apply it within their work. Over the weeks, the team learned extensively about the language, but almost as important, the team has learned to utilize the Extjs documentation. The Extjs documentation is a database containing practically every single aspect and tool contained within each version of Extjs. Knowing how to navigate and use the documentation was vital to become efficient Extjs front-end developers.

VII. Appendices

Figure 5: Modified Tree ComboBoxes

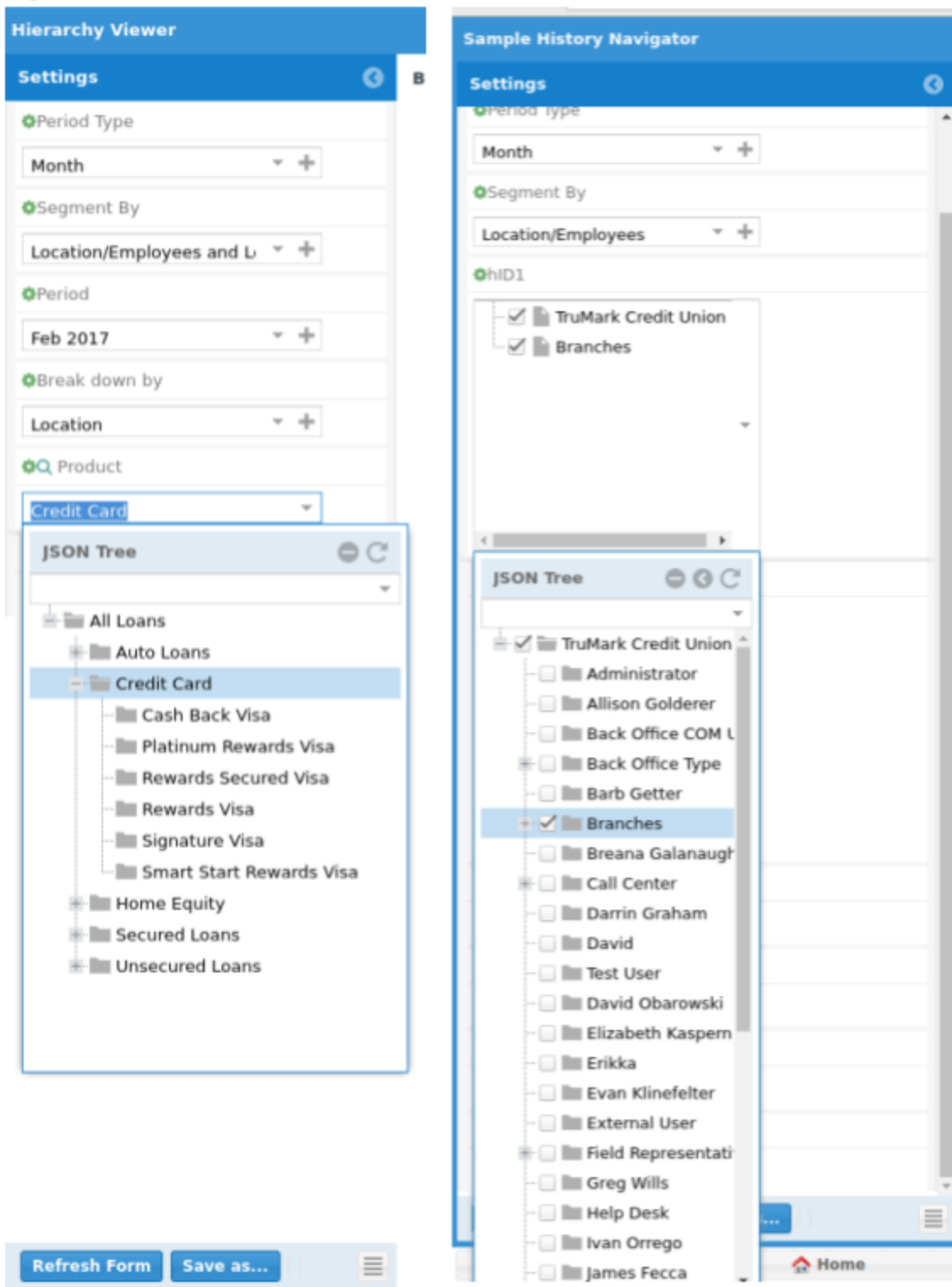






Figure 8: Results of Grouping Tool

| Grouped View | | |
|--|--------------------------------|--|
| Grouped Columns | Add pre-grouping filters here | Results |
|  Record | <input type="text" value="1"/> | of 421   Filters  |
| Row num | Quarter(A... | (Booked L... |
| 1 | ∅ | 0 |
| 2 | 19700 | 1 |
| 3 | 19701 | 0 |
| 4 | 19702 | 0 |
| 5 | 19703 | 0 |
| 6 | 19710 | 1 |
| 7 | 19711 | 0 |
| 8 | 19712 | 0 |
| 9 | 19713 | 0 |
| 10 | 19720 | 0 |
| 11 | 19721 | 0 |
| 12 | 19722 | 0 |
| 13 | 19723 | 0 |
| 14 | 19724 | 1 |
| 15 | 19730 | 1 |
| 16 | 19731 | 0 |
| 17 | 19732 | 1 |
| 18 | 19733 | 0 |
| 19 | 19740 | 0 |
| 20 | 19741 | 1 |
| 21 | 19742 | 0 |