Faculty Code: EAR
Project Number: 1902

ConText, A Tool For the FDA's Adverse Event Reporting System

A Major Qualifying Project Report:
Submitted to the Faculty of the
WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the
Degree of Bachelor of Science

Submitted by:
Jacob Kaplan
Peter Nolan
Logan Romero

Date: March 21, 2018

Approved by:
Professor Elke A. Rundensteiner

Acknowledgments

We would like to thank all of those who assisted us in this project.

We thank our advisor, Professor Elke Rundensteiner, for providing us with guidance over the course of this project.

We thank Tabassum Kakar and Xiao Qin for assisting us in communicating with the Food and Drug Administration, for helping us review our project materials, and for providing guidance on our development.

Finally, we thank the Food and Drug Administration for providing us with the needed report materials for our testing.

We thank all of these individuals for their support and critique, providing valuable insight for our project.

All code and deliverables can be accessed at:
https://github.com/peternolan/MEV-MQP/tree/pnolan

Abstract

This project is a continuation of the development of ConText, a Web Application that’s purpose is to assist FDA Agents in their work of analyzing FDA's Adverse Event Reporting System reports and finding potential patterns between Drug Related incidents. Our focus was on updating numerous previously built functions of the project while also providing brand new features that would make navigating through reports more efficient. Our changes have improved upon the functionality of the previous prototype, making it easier for an agent to sort through the thousands of reports in their workload. This brings the FDA one step closer in realizing their goal in streamlining their pharmacovigilance and potentially saving countless lives.

Table of Contents

1. Introduction

The United States Food and Drug Administration, or FDA, is responsible for “assuring the safety, effectiveness, quality, and security of human and veterinary drugs, vaccines and other biological products...” (Center for Drug Evaluation and Research). As part of these efforts, one of the main tasks of the FDA is to receive and analyze medical reports submitted to the FDA's Adverse Event Reporting System, or FAERS. These reports can originate from healthcare professionals, consumers, and manufacturers, and can detail any manner of drug related incident, such as the assignment of the wrong dosage or accidental exposure to a particular drug. In addition to informing the FDA about such an incident, these reports serve a special purpose of providing FDA agents with further understanding of the nature of drugs that are currently out in the market. Even with the extensive testing that the FDA requires, there is no way for a manufacturer to be able to test a drug against every possible scenario. To this end, the FDA collects the information provided by the FAERS reports, and uses this information to see if they can determine any previously unforeseen adverse reactions or any patterns of drug misuse.

As it stands now, however, the FDA’s workflow for analyzing FAERS reports is in need of an update. Currently, FDA agents are required to go through each report they are provided one by one, analyzing and keeping notes all on their own. Each year, the FDA can receive over a million FAERS reports, with nearly 17 million reports currently being processed within the system. In addition, since an individual FDA agent can potentially receive thousands of reports as part of their workload, going through a significant number of reports is, by itself a very tedious task that requires a large amount of time to complete. It is in the hopes of making this process more efficient that our project first originated. (*FDA Dashboard*)

Last school year, a group of WPI students were tasked with providing a prototype for a new web application for the FDA. This web application would serve as a new tool for the FDA, with the purpose of providing the FDA with a new means of organizing and analyzing the information from thousands of FAERS reports in an efficient and user friendly manner. This web application was given the name Medication Error Visualizer, or MedVis. By the end of their MQP term, this team had provided a working prototype for the FDA that would serve as the first step in the development of the FDA’s new analysis tool. (Murphy et al, 2018)

For our project, our group was tasked with continuing the development of the MedVis prototype. We were to look at what the previous MQP had accomplished, determine what areas could use improvement, and implement as many of these improvements into a new design as would be feasible within our allotted time, as well as provide new features that would expand on the web application’s capabilities.

From our analysis, discussions, and development, we have produced what we hope will act as the next iteration of the FDA’s analysis tool, ConText.

2. Background

2.1 Medication Error Visualizer

As stated above, ConText is the second iteration of the web application first developed by a previous MQP, the *Adverse Reaction Reports Analysis Tool* project submitted in March 2018, and was originally called the Medication Error Visualizer, or MedVis. In this project, the students were tasked with creating the initial prototype for the web application, establishing the basic structure of the application, and beginning development on the primary features.

It is from MedVis that our project inherited its general structure, which is focused around three major components, a front-end for visualizations and UI, a database which contains user data and report data modeled after the FAERS reports, and a back-end that connects the two together. Both the front-end and back-end run with Node JS as their runtime environments, utilizing the numerous additional packages Node JS provides to increase functionality. Together, they are responsible for retrieving the information from the database, and providing the user with effective visualizations that summarize the amassed data. In order to accommodate the influx of data, ConText makes use of the PostgreSQL Database System for its database, as it allows for developers to load large portions of data in real time. Most of the database is divided into User data, which contains information on the application's users, report data, which contains the mock FAERS reports, and Case data, which contains reports that an FDA agent has determined to be similar. (Murphy et al, 2018)

Normally, a user will only be looking at a relatively small portion of the total number of reports, which represents the current workload for an FDA agent. However, even this portion can contain thousands of reports that need to be analyzed and interacted with. In order to accommodate this, the previous MQP constructed numerous features that an FDA agent could use to make searching through and analyzing their workload more efficient. For our project, our task was to assess these features and provide updates wherever necessary.

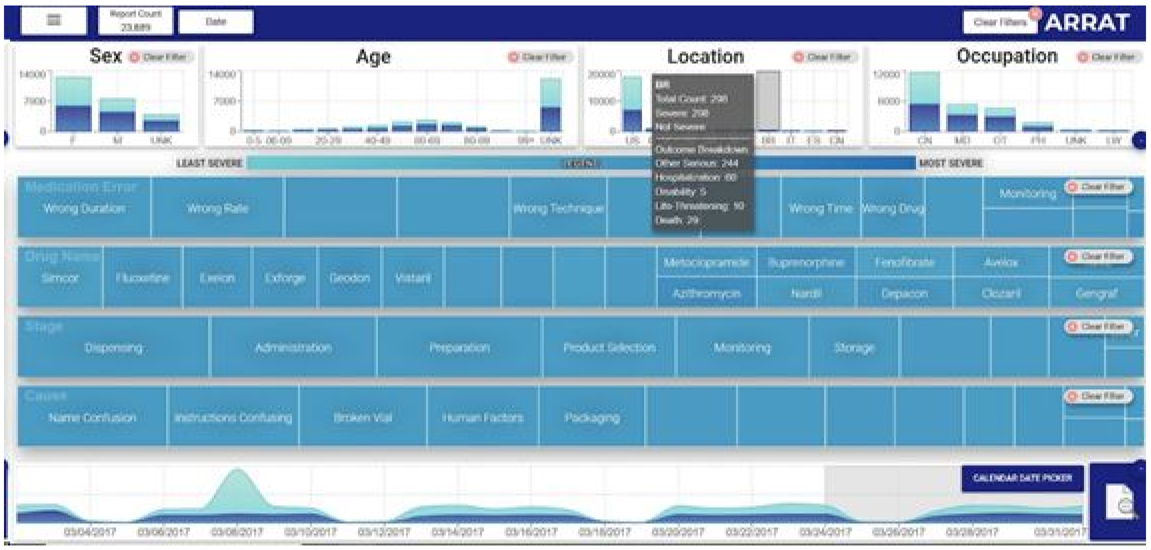


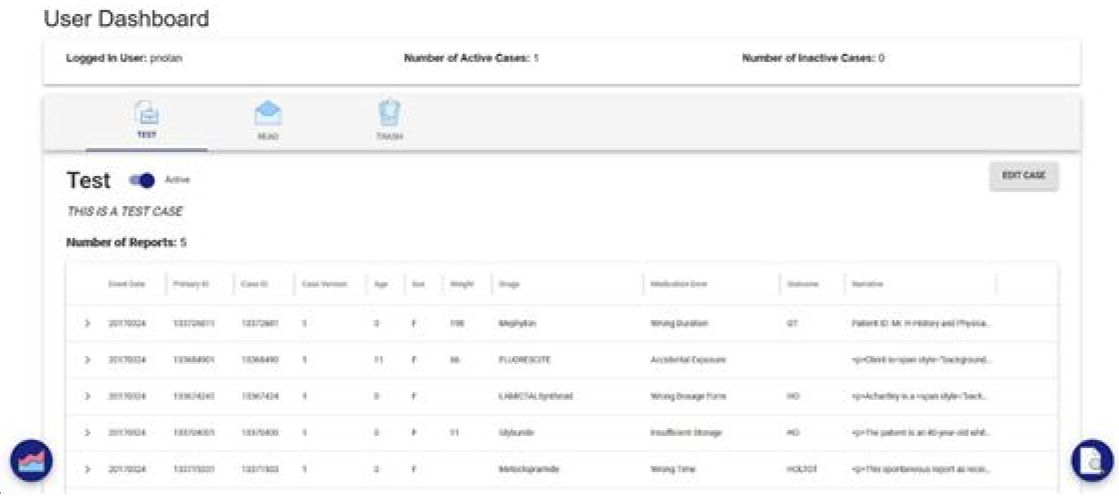
Figure 2.1.1: Original MedVis Visualization Page

2.2 ConText Main Features

ConText contains three major sections: the Visualization Page, the Reports Table, and the Dashboard. The Visualizations Page contains tools that summarized the reports within the user's workload into numerous graphics and charts. These charts included bar charts that display the demographics of report subjects, a timeline of provided reports, and, as its centerpiece, a series of Treemaps which represent fields with a wider variety of articles that can be filtered for, such as the drug name or cause. (Figure 2.1.1)

For our project, we largely left this section unchanged from what was originally in MedVis, as we found that it already suited the needs of our sponsors sufficiently. We did, however, provide some updates to some aspects of the UI, which we will further discuss in the next section.

The Dashboard serves to provide quick links to user created cases, and is the first thing that the user will see when they log into the application. (Figure 2.2.1) Again, we found most of this page to be fine, and only made minor UI changes.


User Dashboard

Logged In User: pnolan **Number of Active Cases:** 1 **Number of Inactive Cases:** 0

Test Active

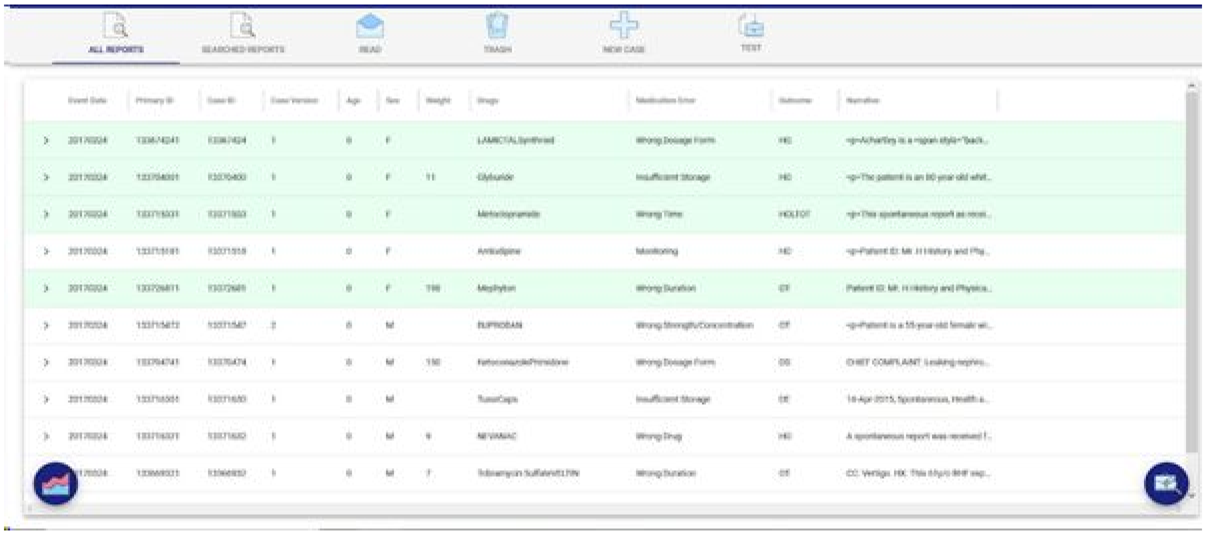
THIS IS A TEST CASE

Number of Reports: 5

	Event Date	Primary ID	Case ID	Case Version	Age	Sex	Weight	Drugs	Medication Error	Outcome	Narrative
>	20170324	133726811	13372681	1	3	F	108	Mephyton	Wrong Duration	OT	Patient ID: Mr. H History and Physica...
>	20170324	133684901	13368490	1	11	F	86	FLUORESCITE	Accidental Exposure		<p>Client is<span style="background...
>	20170324	133674241	13367424	1	3	F		LAMICTAL lamotrigine	Wrong Dosage Form	HO	<p>Achartley is a <span style="back...
>	20170324	133704001	13370400	1	3	F	11	Glyburide	Insufficient Storage	HO	<p>The patient is an 40-year-old whit...
>	20170324	133715331	13371533	1	3	F		Metoclopramide	Wrong Time	HO,OT	<p>This spontaneous report as rece...

Figure 2.2.1: MedVis Dashboard Page

The Reports Page contains a full list of the reports, and provides the user with the ability to separate them into Cases (groups of related reports) and gain access to a report's full contents. It is within the Reports Page where the majority of our changes have taken place, and is where our new features are made available to the user. (Figure 2.2.2)



	Event Date	Primary ID	Case ID	Case Version	Age	Sex	Weight	Drugs	Medication Error	Outcome	Narrative
>	20170224	133674241	13367424	1	0	F		LAMICTAL,Synthroid	Wrong Dosage Form	HO	<p>Acharfley is a <span style="back...
>	20170224	133704001	13370400	1	0	F	11	Glyburide	Insufficient Storage	HO	<p>The patient is an 80 year old whit...
>	20170224	133715031	13371503	1	0	F		Metoclopramide	Wrong Time	HO,LT,OT	<p>This spontaneous report as recei...
>	20170224	133715181	13371518	1	0	F		Amlodipine	Monitoring	HO	<p>Patient ID: Mr. H History and Ph...
>	20170224	133726811	13372681	1	0	F	198	Mephyton	Wrong Duration	OT	Patient ID: Mr. H History and Physica...
>	20170224	133715472	13371547	2	0	M		BUPROBAN	Wrong Strength/Concentration	OT	<p>Patient is a 55 year old female wi...
>	20170224	133704741	13370474	1	0	M	150	Ketoconazole/Primidone	Wrong Dosage Form	DS	CHIEF COMPLAINT: Leaking nephro...
>	20170224	133716301	13371630	1	0	M		TussiCaps	Insufficient Storage	OT	18-Apr-2015, Spontaneous, Health a...
>	20170224	133716321	13371632	1	0	M	9	NEVANAC	Wrong Drug	HO	A spontaneous report was received f...
>	20170224	133669321	13366932	1	0	M	7	Tobramycin Sulfate/[illegible]	Wrong Duration	OT	CC: Vertigo. HX: This 69y/o WHF wa...

Figure 2.2.2: Original MedVis Reports Table Page

According to the previous project, user testing for these features returned mostly positive results. Most participants were able to comprehend what each feature was intended for, how to use the feature, and were able to understand the feedback they received after performing an action, meaning that as it stood, MedVis was functional. However, after our analysis, we determined that there were numerous aspects that could be further improved. The previous MQP set up the foundations for the Reports Table and Dashboard, and put the majority of their efforts into the treemap visualization. As such, while the treemap is almost production quality, other sections had only basic functionality implemented. During our initial analysis of the application, our group was informed that the FDA was most interested in expanding the Reports Table, as it previously featured a user interface for accessing individual reports, but no way of organizing them outside of putting them into Cases or finding reports with specific characteristics. In order to solve these issues, the FDA informed us that they would like to add new features such as a Search Engine for the Table, as well as new means of summarizing reports and making information easier to obtain. Since these were the points that the FDA was most interested in, implementing these two features would be what our project primarily focused on. (Murphy et al, 2018)

In addition to these changes, there were other parts of the project that could be improved upon. While testing, the previous MQP noted that these features performed at desired levels. Most participants were able to complete the assigned tasks in the correct manner, and were able to determine the purposes of the different features. However, there were also several features that participants responded negatively to. For example, were also problems that occurred during review of report annotations, in which a user can highlight a section of a report with a specific color. Participants expressed some confusion over how to add and remove annotations. In addition, while analyzing the application ourselves, we determined that there could be other minor UI elements that could be improved, which we will go into more detail on in later sections.

2.2.A Report Table Interface

As previously mentioned, one of the sections that out project focused on the most was the Reports Page. The primary focus of our predecessors was on the basic structuring and the Visualization Page of the website, and due to the time constraints of an MQP, were unable to allocate as many resources to the reports section. As a result, the report interface lacks the interactivity that the previous MQP team expanded on in section 6 of their report. They mentioned altering the case summary to include "…a series of bar charts to show the frequency of specific highlighted words within the case report narratives…". (Murphy et al, 2018) Alongside the features requested by the FDA, a large portion of this work will be using the ReportList structure designed by the previous group. The ReportList structure is the underlying class for the Reports Table and contains the ReportTable structure, which presents and houses all cases within a pool of reports (All reports, read reports, a case, etc.). Within the ReportsList, we have included numerous new features, and have changed many different aspects of the web application's Front End.

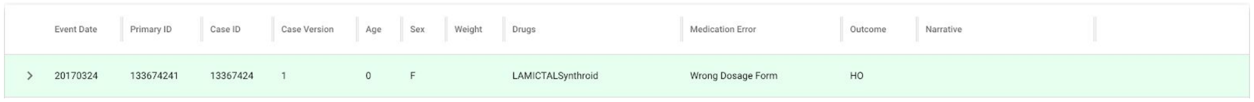
2.2.B Document Searching

As mentioned, in addition to updating the ReportList for viewing and sorting reports, the FDA expressed interest in adding a search function for the Report Section. This feature would allow a user to enter a search text/phrase to further filter through the reports in the Report. Using this tool could improve efficiency by allowing a user to search through 10,000's of reports' texts, which, at present, can only be done by manually reading each report. This facilitates new types of report filtering, for example; antibiotic A may help treat a rash caused by wearing a cast, but while the FDA can see all reports with A present, a search function is needed to further filter based on presence of words like "cast" or "broken bone", etc. Later, we discuss additional features dependent on text search, such as: "similar reports", "tag based filtering", and others.

3. Methods & Implementation

3.1 Interface Redesign & Face Lift

As stated above, the site is split into three separate pages: The Dashboard, The Visualization, and The Reports Listing page. The previous MQP group spent a large portion of their time working on the dashboard and the visualization, as well as many of the common features between pages (such as the navigation bar), but due to time constraints were unable to spend as much time developing the reports page. One of our goals in this project was to polish the system interface as a whole, but we paid especially close attention to the Reports Listing page to make up for the discrepancy in development time. Overall, our time spent redesigning the website would be focused into two categories: Report Table Interactions and Case Summary Interactions. (Murphy et al, 2018)

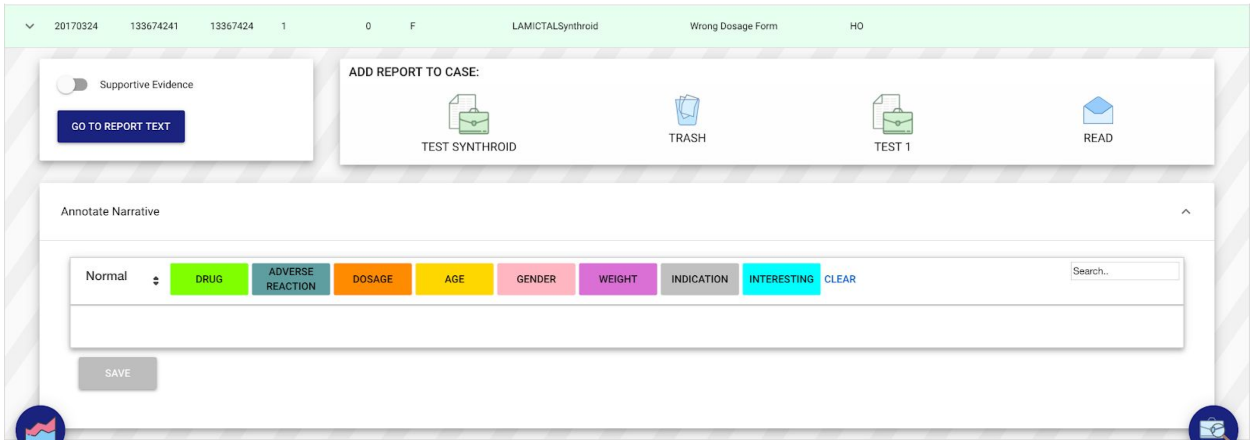


	Event Date	Primary ID	Case ID	Case Version	Age	Sex	Weight	Drugs	Medication Error	Outcome	Narrative
>	20170324	133674241	13367424	1	0	F		LAMICTALSynthroid	Wrong Dosage Form	HO	

Figure 3.1.A1: Original row within ReportList (Collapsed)

3.1.A Report Table Interactions

Probably the largest target of our interface redesign, the general interface encapsulates the reports listing and its many interactions. The initial design featured a very large (almost loquacious) table of reports that allows the user to very quickly sift through the reports on screen by certain parameters expressed by the column headers (see Figure 3.1.A1). However, many of the fields within the structure were superfluous and didn't add meaningful information for the user. For example, there are three separate columns referring to a reports identity: The report ID, the report version, and the primary ID. The primary ID serves as the actual identification of unique elements within the list, and is actually just the concatenation of the case version onto the case ID, thus making the presence of all three redundant. Another element of the listing that serves little purpose is the last element, the narrative, which is the HTML stylized version of the report text.

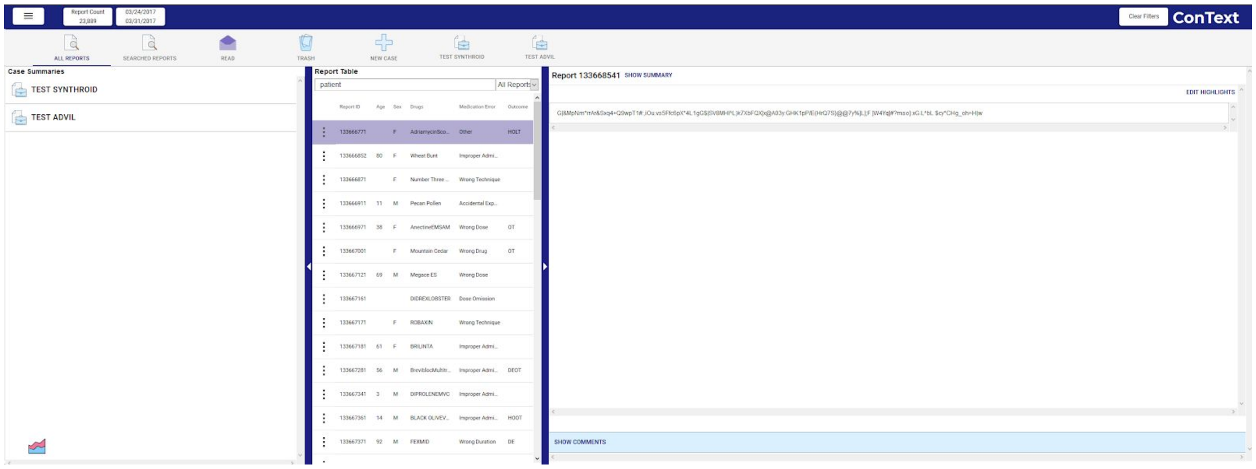


⌄	20170324	133674241	13367424	1	0	F	LAMICTALSynthroid	Wrong Dosage Form	HO

Figure 3.1.A2: Original row within ReportList (Expanded)

Beyond the listing itself, the other issues with the interface come with the contents of the listings. Each row of the report table can be expanded to reveal several functions and to allow the user to access the report text (see Figure 3.1.A2). This method of display very rapidly takes over the user's screen space, and the vertical growth of the contents can make the user lose track of their place if the elements of the list is constantly growing and shrinking. Further, the space within the contents is not being used effectively; all the functions within it are valuable, however they take up a large portion of the screen, making it very hard to perform these operations on different reports in succession. In order to move ten reports from the listing into a case the user would need to expand each report individually and select the operation. However, given that only one report can fit on screen at a time, they will likely lose their place. These same issue bleeds over when viewing the report text.

The report text is expanded below the interfaces of the aforementioned functions, and, depending on the size of the user's display and the length of the report, may not be able to be displayed at the same time as those functions. The content display as a whole does not effectively consider the screen real estate of the user, and an overuse of negative space within the design greatly limits its effectiveness. Instances of negative space can be found in the margins surrounding the pieces of the display; the report table has excess padding within the application window, and the contents of the table have excess padding, and the contents of the individual listings have excess padding and so on. While padding can be effective in conveying separations within the interface, too much can prove redundant and in this case detrimental to the function of the interface. Further, the presence of a button to go to the report text is redundant if the report text can also be viewed below that same button.



Report ID	Age	Sex	Drugs	Medication Error	Outcome
133666771		F	AdriamycinSco...	Other	HOLT
133666852	80	F	Wheat Bunt	Improper Admi...	
133666871		F	Number Three ...	Wrong Technique	
133666911	11	M	Pecan Pollen	Accidental Exp...	
133666971	38	F	AnectineEMSAM	Wrong Dose	OT
133667001		F	Mountain Cedar	Wrong Drug	OT
133667121	69	M	Megace ES	Wrong Dose	
133667161			DIDREXLOBSTER	Dose Omission	
133667171		F	ROBAXIN	Wrong Technique	
133667181	61	F	BRILINTA	Improper Admi...	
133667281	56	M	BreviblocMultir...	Improper Admi...	DEOT
133667341	3	M	DIPROLENEMVC	Improper Admi...	
133667361	14	M	BLACK OLIVEV...	Improper Admi...	HOOT
133667371	92	M	FEXMID	Wrong Duration	DE

G|&MpNm*rrAr&Sxq4+Q9wpT1#_iOu.vs5Ffc6pX*4L1gG$|SV8MHI^L)k7XbFQX|x@A03y:GHK1pP/E(HrQ7S)@@7y%|L|;F |W4Yq|#?mso|:xG:L*bL $cy*CHg_oh=H|w

Figure 3.1.A3: New Reports Page Design for ConText

Our redesign represents a shift from the previous vertical listing of the interface to a more commonly used paneled interface (see Figure 3.1.A3). This design draws inspiration from most email software front-ends where there is a selection of folders, a listing of emails (in this case reports), and a panel for the contents of those emails. To make this all work, we've decided to rearrange the pieces of the interface to better convey the control flow of the system. Panel A contains the workings of the case summary system, which we will be explaining in more detail in section 3.1.B, and can control the contents of panel B. Additionally, the contents of panel B can be changed with the use of the tab selection bar at the top of the interface. From there, a line within panel B can be selected and the contents of the report, comment, and summary information for that report will be displayed in panel C.

Moving from a vertically aligned interface to a horizontally aligned interface allows us to convey more information to the user whilst also retaining access to multiple features on the same screen. Displays are almost always wider than they are tall (with more displays featuring a 16:9 display ratio) which will allow us to fit more information on the page at a given time. Further we have added toggles to each panel to allow the user to decide what is displayed on the screen. Panels A and C can be minimized to increase the amount of space available to other elements of the interface.

Panel B will always remain on the screen because it is the link between the other two panels, and it is not feasible that the user would use A or C without B. The inverse also holds true, wherein the user would not often use the table without using one of the other two elements of the interface, therefore we have decided to make our toggles follow 'OR' logic. A can be minimized to show just B and C or C can be minimized to show just A and B. When A or C are minimized the other will grow to fit the new allotment of space, a valuable option when analyzing either the report text or the case summaries.

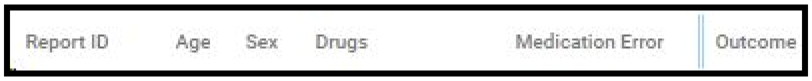


Figure 3.1.A4: Columns Retained for new Report Page Design

This redesign would also require various adjustments to many of the individual pieces of the interface to maximize our space efficiency while maintaining simplicity. Within panel B our primary concern was to maximize the efficiency of our use of horizontal space within the panel, we can stack as many things vertically as we want, but we can only display so much information horizontally. To work around this, we decided that we would need to prune some of the columns on the table listing. The columns we maintained can be seen in Figure 3.1.A4. Once the horizontal dimension had been reduced we needed to address the positioning of functions previously found within expanded listings: The ability to mark reports as supportive, and the ability to move them to and from the system and case bins.

One of the largest issues with the previous system was that all actions and content pertaining to a report required the report to be expanded. To address this the managerial functions of the reports were placed into a side menu within the row, denoted by a vertical ellipsis—a symbol essentially synonymous in modern design with meaning more. Further, in keeping the function on a per line basis we can allow the user to perform the same action on multiple reports in rapid succession without scrolling between reports, thus mitigating the chance that they lose their place.

Another addition to panel B was the report search bar, which would allow the user to search through reports in specific pools of reports ranging from all reports to specific cases. We will be discussing the exact functioning of our system's search later on in this chapter. Accompanying the search bar is a drop down menu for selecting the target pool of the search. The system defaults search to target all of our reports, but if the user were to select another option from the dropdown they could just as easily search one of their cases, or even trash or read.

Once a search query is launched the reports page will enter a loading state until the search returns results—while in the loading state, all interactions with the reports page will be locked. Once the query returns the results the report table will be updated with these results and the individual cells will be expanded to display excerpts from within the text of the report that best match the query. These excerpts are generated by the search system we have in place and are ordered by the score given by that system (detailed more in section 3.3). All interactions with each cell remain the same, the excerpts are just given to allow the user insight as to how the report was chosen.

For insight on how all aforementioned changes operate and appear, refer to the 3.1.A Case Management Walkthrough found at the end of this sub section.

Moving on to panel C, most of our work would be reformatting the report text and some minor additions revolving around user interaction with the report text (see a visual example in section 3.2). The first addition to the interface was the addition of a space for the system to display report summaries, an intended feature for a future release of the system—accompanying this was the ability to toggle the display of the summary. The next and most prominent interaction between the user and the report text was the ability for the user to highlight specific portions of the report with different colors to signify different keys (such as patient age, and drug names). The original design featured the annotation interface as part of the generic display of the report text (seen above). The initial design was not very intuitive and used colored buttons to display the controls for the report text, however it did not in any way explain to the user the functions of these buttons relative to the interface. The actual function of the interface is that any text that you highlight after pressing on of the buttons will be highlighted with that color in the display. More on this is explained in section 3.2.

Beyond our changes to the interface surrounding these functions we have also changed the formatting of the report text itself. Report text will be displayed pretty standardly with the ability to scroll down the page should the text extend beyond the bounds of the display. Further, we make standard use of padding and font weight and size to align different sections of the text and make the different bodies differentiable at a glance. Our intent with this panel was to keep the display as simple as possible without the use of any gimmicks or too many moving parts, so the user could simply parse through the text for what they need. Overall, we believe the general redesign to have addressed most of our concerns with the previous interface while also providing the user with more functionality and ease of use.

3.1.A Case Management Walkthrough

I. Case Management Options

Previously, the management tools for reports were nested inside of a collapsible panel attached to each report entry in the table (see figure 3.1.A2). These tools allowed the user to move reports to different bins and, when accessed from within a bin, remove the report from that bin or mark a report as supportive or primary. These functionalities were all retained in the new system you can access them as follows.

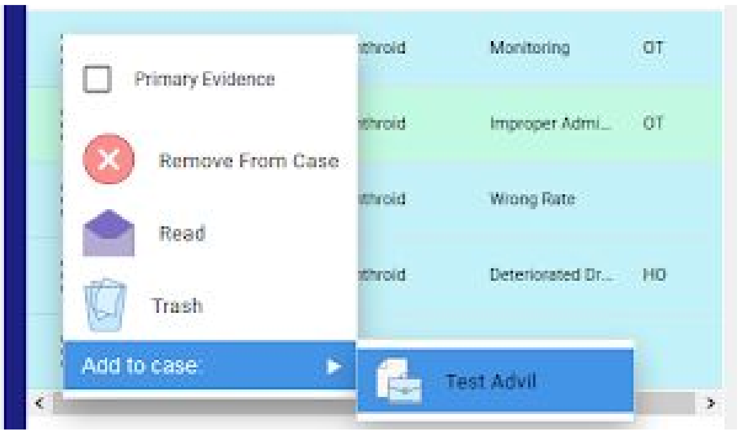
To start, simply click the vertical ellipsis (shown right) to expand the context menu (seen below):

1. To move a report to a bin/case:
    a. Select the name of the desired bin.
        i. If the desired bin is one of your created cases, hover over the "add to case" text to reveal a list of your cases.

As was previously mentioned, there are specific interactions you can perform when opening this menu inside one of the system bins:

1. To remove a report from a bin/case:
    a. Select the "Remove From Case" option from the menu.

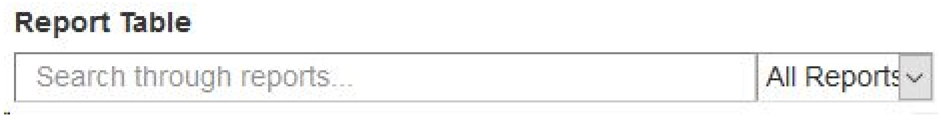
2. To mark a report as Primary or Supportive:
    a. Check or uncheck the “Primary Evidence” box.



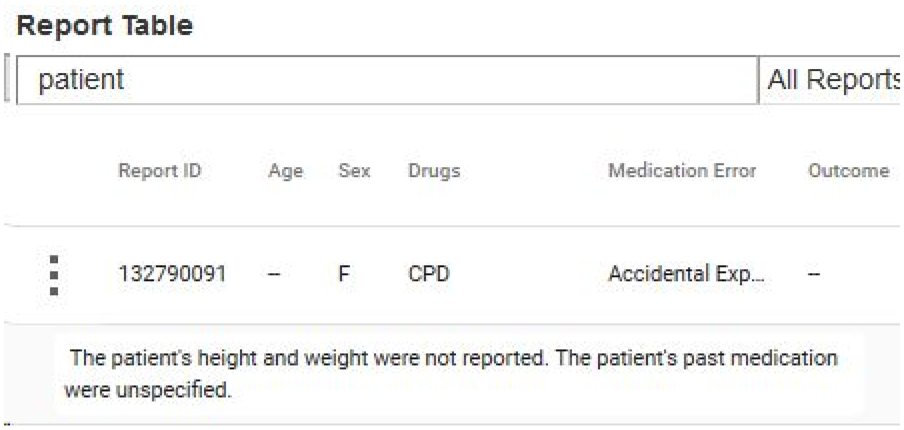
After each action has been successfully completed, the icon associated with that action will update. If you are adding the report to another case the hue of the icon will shift to let you know that it is already in there. Removing a report will simply remove the report. And checking the box will show a visible check within the box.

II. Search Interactivity

To launch a search query, enter the desired text into the search bar found at the top of the Report Table panel. Once you have entered the text into the search bar, press enter to launch the query. Additionally, you may choose to select a target for the search query via the dropdown menu found adjacent to the search bar (see below).



Once the query has been launched, the screen will enter a loading state like the one the site enters when you first venture to the reports page. Once the query has been resolved you will be presented with the results of the search query. These results resemble the entries in the base report table, however they now have a small section that shows the excerpts related to the query you launched (see below).



Report Table

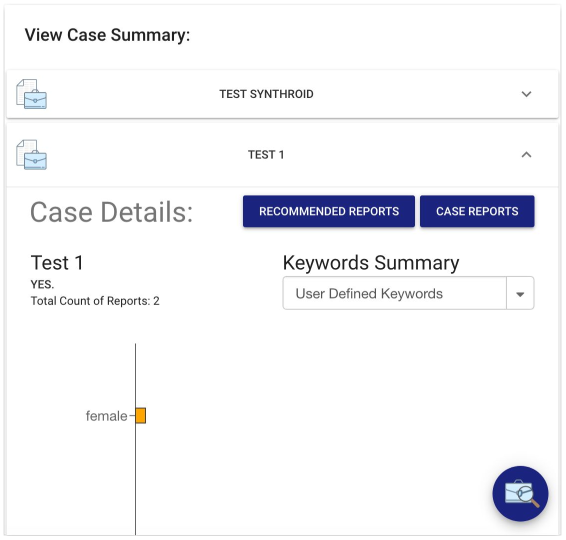
patient | All Reports

	Report ID	Age	Sex	Drugs	Medication Error	Outcome
⋮	132790091	–	F	CPD	Accidental Exp...	–

The patient's height and weight were not reported. The patient's past medication were unspecified.

3.1.B Case Summary Interactions

As previously mentioned, panel A contains an interface that in-part functions semi-laterally to the tab selection bar at the top of the interface. Much like the tab bar, this panel has the ability to change the contents of panel B to display the reports within a case entirely independent of the filters in place on the system. As a result, we thought it best to move its interface to the far left of the site to better convey the control hierarchy of the system (explained in more detail in section 3.1.A).



View Case Summary:

TEST SYNTHROID

TEST 1

Case Details:

RECOMMENDED REPORTS | CASE REPORTS

Test 1

YES.

Total Count of Reports: 2

Keywords Summary

User Defined Keywords

Figure 3.1.B1: Original Case Summary Panel

The original design (see Figure 3.1.B1) was originally on the far right of the site and fell prey to the same design choices as the rest of the system. The panel would open to display a list of bins pertaining to the user's active cases, and when one would select a bin it would expand down beyond the length of the page with large portions of the display not even being effectively utilized. To display word frequency, they utilized a bar chart to convey the occurrence of each word, however the previous group did not account for the scenario wherein there would be a substantial number of keywords. Further, even with a small number of keywords the chart is excessively padded and extends beyond the length of the screen with most horizontal space containing nothing. Accompanying the bar chart was a pie chart to see the ratio of supportive and primary reports within a case—supportive meaning it supports a potential hypothesis, and primary meaning it is a report that falls within the bounds of the hypothesis.

Following basic design principles, humans are less effective at judging the differences in angles than they are at judging the length of a bar along a fixed scale. This discrepancy is well documented and on the of the most popular explorations of this comes from William Cleveland and Robert McGill (Cleveland & McGill 1985). The original interface featured both a pie chart and a bar graph, however both were not implemented properly, as stated above. While the pie chart does its job quite well, making it easy for the user to see the ratio between the two categories, it becomes less effective when the number of categories increases—a feature needed in the new system.

It was decided that the new system would need to allow more exploration of contents of the cases; building off of the supportive vs. primary comparison we also wanted the user to be able to compare across certain fields within the table (such as medication error, outcome, etc.). When implementing these features, it became quite difficult to justify our use of the pie chart as it lacked a balance of accuracy and compactness that would be necessary within the panel. To fix this, we replaced the original pie chart with a horizontal stacked bar chart (see Figure 3.1.B2), a tool commonly used to display partitions. Aside from using an aligned scale to compare length—a method ranking higher than pie charts (Cleveland & McGill 1985)—the bar chart also allowed us more efficient use of vertical space within the panel. The bar chart shows only one category at a time, but the contents of the bar chart can be changed by selecting an option within the Case Breakdown dropdown menu. Further, to allow the user more precision when viewing the partitions, we have added a legend with the names of the categories and the counts within each. Once the pie chart had been remedied we moved onto the bar graph.

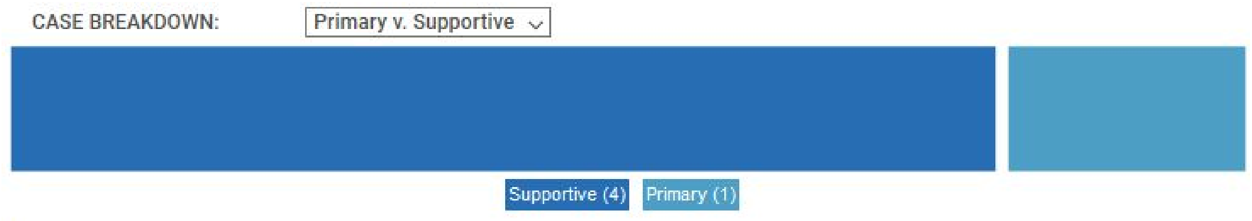


Figure 3.1.B2: New Bar Chart, showing Evidence Type distribution

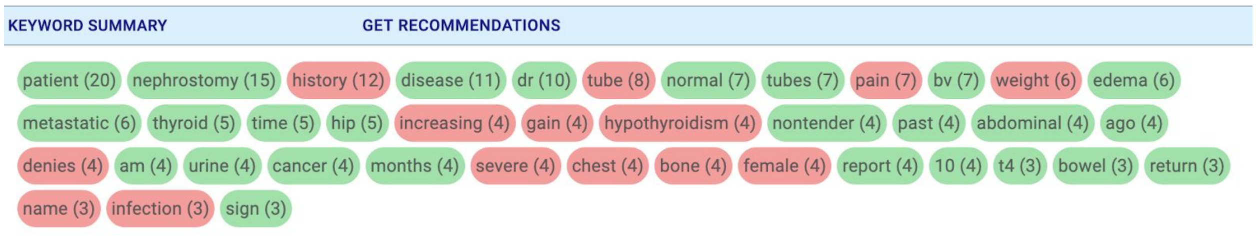


Figure 3.1.B3: New Keyword Display within Case Summary Panel

Much like the pie chart, in our iteration of the system, the original bar graph would need to accommodate a new mechanism for the user. The keywords would now be used in structuring a query for generating report recommendations—a new feature of the system (detailed in section 3.4). As such, it no longer felt proper to use a bar graph to display the information—aside from its poor use of space. In the redesign, keywords are now displayed as a descending-ordered list of buttons displaying both the keyword and the occurrences of said word (see Figure 3.1.B3). Each button is defaulted to a green color and can be clicked to turn the word off when generating recommendations (which will also change its color to red) and clicked again to toggle it back. Once the user is happy with their selection of keywords for generating recommendations they simply press the Get Recommendations text button and the system resolves the request. Once the request has been sent, the site will enter a loading state until the search returns the recommendations in the same form as a generic search from the table.

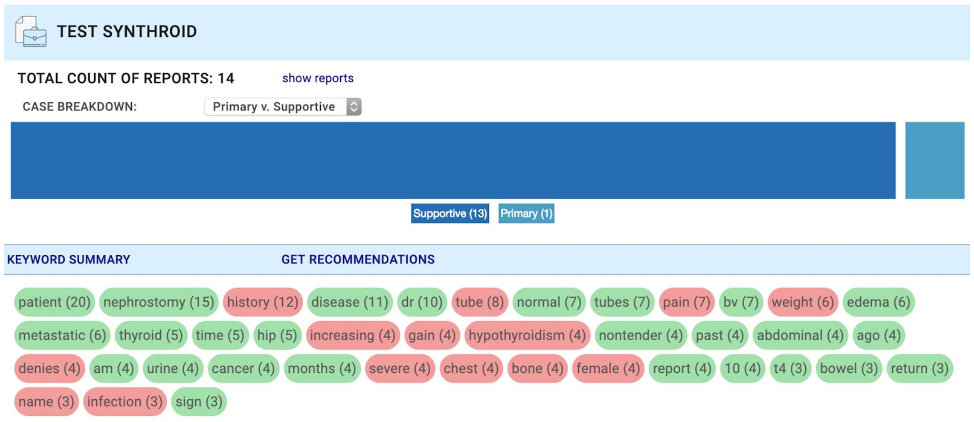


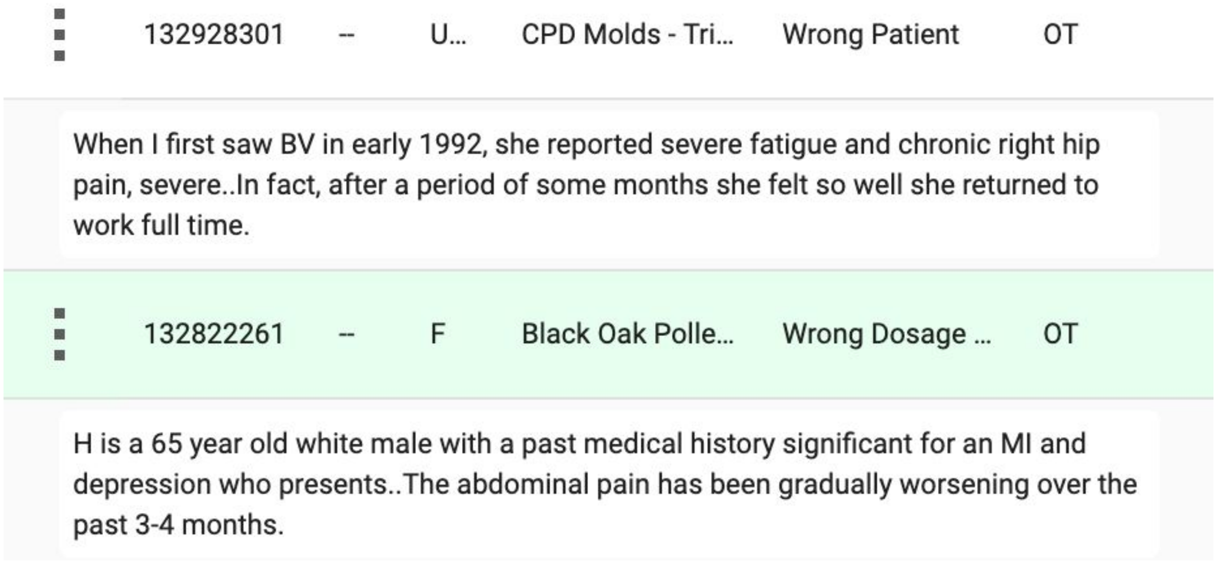
Figure 3.1.B4: New Case Summary View. Example shown is for Case "Test Synthroid"

In addition to these changes, we have rearranged many parts of the case summary interface (see Figure 3.1.B4). The Case Reports button has been renamed to "show reports" and moved to be in-line with the total number of reports and their classifications. The Get Recommendations button has been moved into the keyword sections, as it will be recommending reports based on these keywords. And more broadly, we have reduced a fair amount of the padding within the bin listing and the contents of the bins. With this, the user will be able to see the information within the cases and toggle between them with greater ease and efficiency.

3.1.B Case Summary Walkthrough

As was previously mentioned, the new case summary allows for the user to request the retrieval of reports similar to those in the case that they are viewing. Accompanying this new feature, we have added two elements to the case summary: the keyword summary section and the get recommendations command button (see below). The former allows the user to view the keywords associated with the case they are examining and additionally allows them to change the query used to find similar reports.

The recommendations are generated via a series of keywords from within the case summary, by default all keywords are selected but the user may toggle any of them off to change the query. To toggle a word on or off simply click on the keyword bubble; a word is considered "on" if it's green, and "off" if it's red (see Figure 3.1.B3). Once the user is satisfied with their selection of keywords, they may press the Get Recommendations button to launch the query. The query will then resolve identically to a search query—for more information regarding that see section 3.1.A Walkthrough, subsection I.



132928301	--	U...	CPD Molds - Tri...	Wrong Patient	OT

When I first saw BV in early 1992, she reported severe fatigue and chronic right hip pain, severe..In fact, after a period of some months she felt so well she returned to work full time.

132822261	--	F	Black Oak Polle...	Wrong Dosage ...	OT

H is a 65 year old white male with a past medical history significant for an MI and depression who presents..The abdominal pain has been gradually worsening over the past 3-4 months.

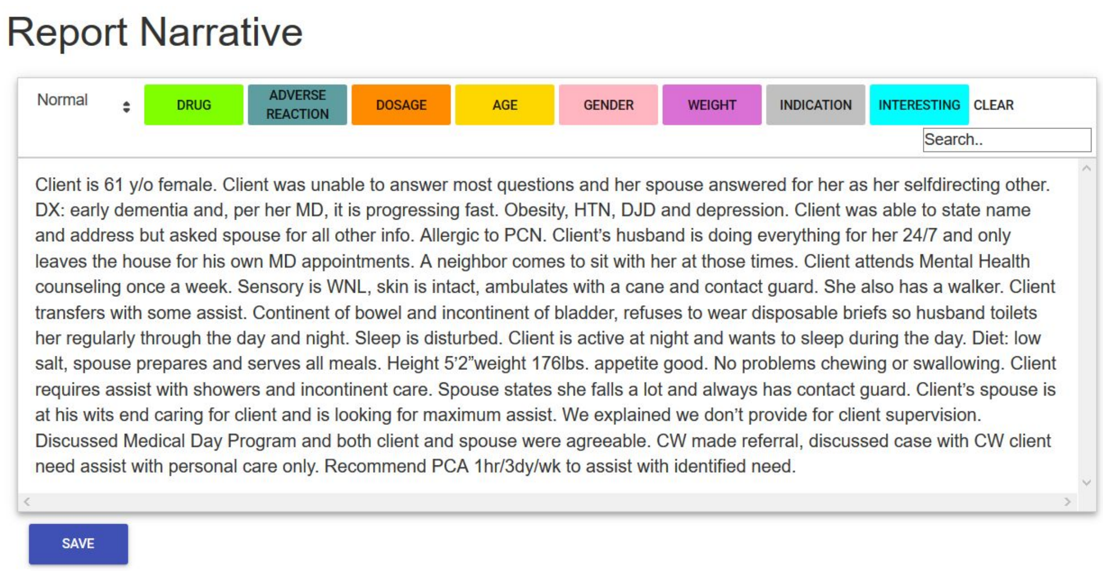
Figure 3.1.B4: Search Results display. Beneath each row, a short excerpt will be provided to preview the narrative

By launching the query formed by the "on" keywords in Figure 3.1.B3 the system will return a series of reports close to this query (the first two are shown in Figure 3.1.B4). The reports are ordered based on closeness to the query; meaning that reports with a higher frequency of keywords from the query will be scored higher and rise to the top. Within our response we can see the presence of several of our keywords, namely: "patient", "BV", "report", "months", and "return". You will also note that some of our "off" keywords appear in the results, such as "pain", this means the off keywords are not exclusively toggled. If you launch a query you *may* receive results containing your "off" keywords. Know that the reports are not graded with these and they do not impact the ranking of the returned reports.

3.2 Narrative Annotation

3.2.A Front End

In order to view and annotate report narratives, ConText utilizes React-Quill, an open source text editor. This library was first implemented during the original project, and was kept by us since it already provided much of the needed functionality for viewing the narrative and making highlights. By itself, React-Quill provides text editor functionality, such as changing font size, bolding, italicizing, etc. However, for our project, we needed to add our own functionality to the system with the goal of allowing users to put their own, written comments into the report, as well as updating the current UI in order to properly convey this functionality to the user.



Client is 61 y/o female. Client was unable to answer most questions and her spouse answered for her as her selfdirecting other. DX: early dementia and, per her MD, it is progressing fast. Obesity, HTN, DJD and depression. Client was able to state name and address but asked spouse for all other info. Allergic to PCN. Client's husband is doing everything for her 24/7 and only leaves the house for his own MD appointments. A neighbor comes to sit with her at those times. Client attends Mental Health counseling once a week. Sensory is WNL, skin is intact, ambulates with a cane and contact guard. She also has a walker. Client transfers with some assist. Continent of bowel and incontinent of bladder, refuses to wear disposable briefs so husband toilets her regularly through the day and night. Sleep is disturbed. Client is active at night and wants to sleep during the day. Diet: low salt, spouse prepares and serves all meals. Height 5'2"weight 176lbs. appetite good. No problems chewing or swallowing. Client requires assist with showers and incontinent care. Spouse states she falls a lot and always has contact guard. Client's spouse is at his wits end caring for client and is looking for maximum assist. We explained we don't provide for client supervision. Discussed Medical Day Program and both client and spouse were agreeable. CW made referral, discussed case with CW client need assist with personal care only. Recommend PCA 1hr/3dy/wk to assist with identified need.

Figure 3.2.A1: Original Narrative Annotator

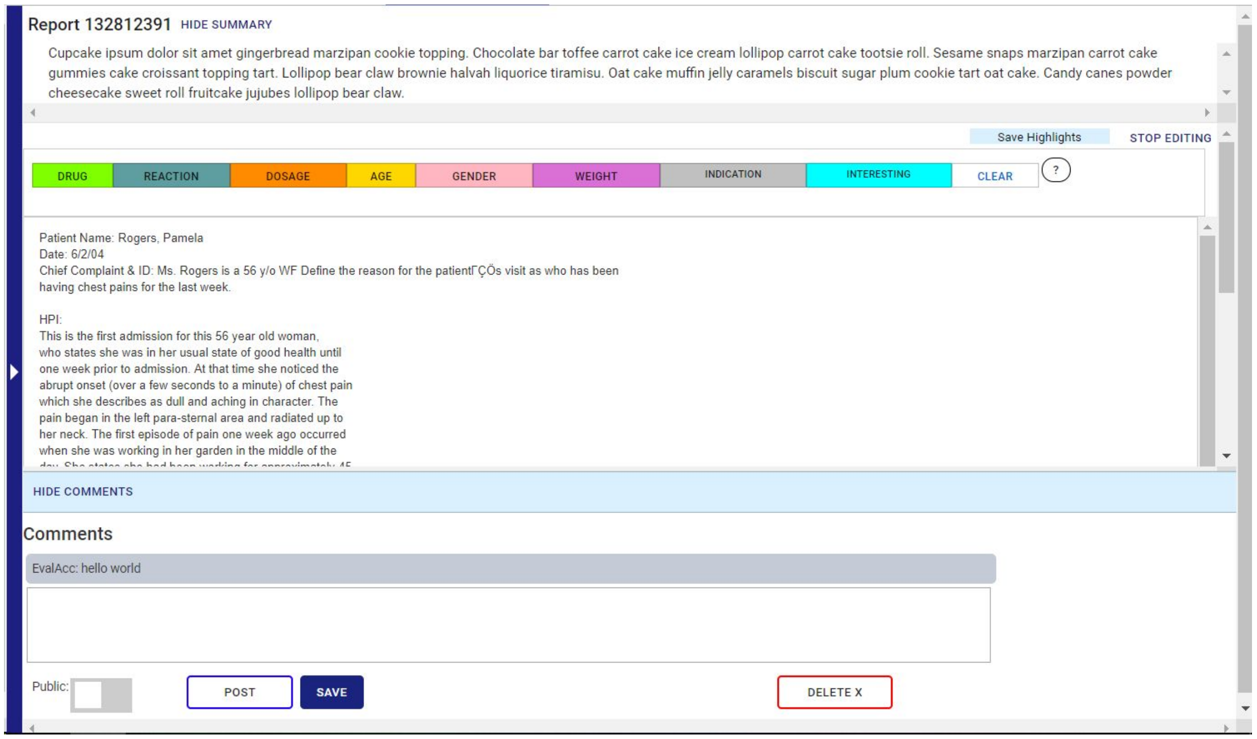
As we can see above, the previous Reports View includes numerous buttons that represent the different highlighting categories that can be used on the report text. In order to highlight part of the text, the user needs to first highlight the desired portion normally with their cursor, then press one of the above buttons while the text is still selected. The text will then be highlighted with the chosen color. After this is done, the user needs to press the save button in order to make their changes permanent.

During our project, we realized that one problem with the current system was that it is difficult for people who are unfamiliar with the system to discover the process needed to annotate the report text, or that there is even a method of editing at all. All they were given is the text and the toolbar at the top, but no instruction on how to use the buttons to highlight the text. In response, our team took several measures in changing the front end to provide the user with more direction. First, in order to confirm that the user is able to edit the text, we added an "Edit Report" button to the top of the editor, which hides the toolbar until the user is ready to edit. This way, the user can clearly see that they are able to edit the report in some way, and when they hit the button, they will know that anything that appears can be used as a tool for them to interact with. For when the user is editing, we also added a small help icon to the toolbar, so that when the user hovers over the text, they will see a short set of instructions on the highlighting process. This will provide them with all of the information that they need in order to properly edit the report text. With these changes, the user is provided with the necessary information on how they can highlight the report text.

In addition to updating the highlighting feature, we also expanded the user's ability to annotate the text through making comments. In conjunction with the search engine, it was decided that for this project, the comments would be saved by having them concatenated with the report text within its own '<comments>' section of the html, which includes an unique id that is based on the id of the report. With each comment placed into its own <comment> line, which includes an id based on the current user's id in order to help keep track of who's comment belonged to who. When a comment is loaded from the system, the report text html is scanned for the <comments> tag. If it appears, it is separated from the rest of the text and placed inside of the middle text box. This way, search results will be able to also search through comments without adding any significant overhead to the system. All together, this system allows for users to quickly make and save annotations to the system in a simple manner.

3.2.A Comments Walkthrough

To allow for comments, we included a new Comments Section into the Report Panel, which always remains on the bottom of the Panel no matter where the user has scrolled to. Within this section, we have included a text box for the user to type out their comment in, and a toggle that can be used to determine if the comment will be private, in which only the user would be able to see it, or public, in which all users would be able to see the comment. In order to make a comment, the user simply needs to enter the desired annotation into the bottom textbox and then press the "Make Comment" button. This will add the comment to the report and present it above the text box. If the comment is public, the comment will be within a bright blue box that also shows the user name of the poster. If Private, the user will see the same, but box will be grey and hidden from other users. As with highlighting, the user then needs to press the save button in order to make this comment permanent.



Cupcake ipsum dolor sit amet gingerbread marzipan cookie topping. Chocolate bar toffee carrot cake ice cream lollipop carrot cake tootsie roll. Sesame snaps marzipan carrot cake gummies cake croissant topping tart. Lollipop bear claw brownie halvah liquorice tiramisu. Oat cake muffin jelly caramels biscuit sugar plum cookie tart oat cake. Candy canes powder cheesecake sweet roll fruitcake jujubes lollipop bear claw.

Patient Name: Rogers, Pamela
Date: 6/2/04
Chief Complaint & ID: Ms. Rogers is a 56 y/o WF Define the reason for the patientΓÇÖs visit as who has been having chest pains for the last week.

HPI:
This is the first admission for this 56 year old woman, who states she was in her usual state of good health until one week prior to admission. At that time she noticed the abrupt onset (over a few seconds to a minute) of chest pain which she describes as dull and aching in character. The pain began in the left para-sternal area and radiated up to her neck. The first episode of pain one week ago occurred when she was working in her garden in the middle of the

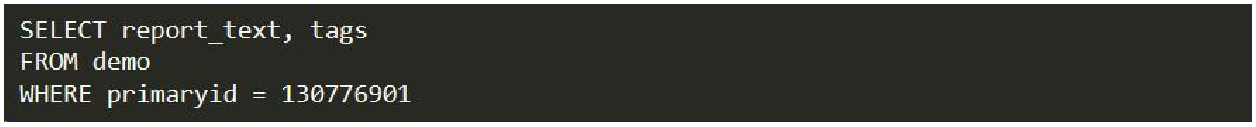
EvalAcc: hello world

Figure 3.2.A2: Current Quill-Editor Annotator

3.2.B Back End

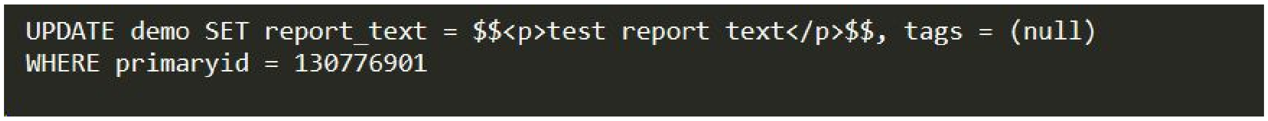
For our project, we decided to keep the majority of the current backend intact. There are only two main queries that we utilize when we are annotating the text. The first is the 'getreporttext' query, which asks for the HTML text, as well as information from the "tags" column, which contains the location and categories of previously made highlights, for a specific report that is identified through the primaryid.

When the user is saving their edited report, we use the 'savereporttext' query. This will save the current report text and tags to the specified report primaryid.



```
SELECT report_text, tags
FROM demo
WHERE primaryid = 130776901
```

Figure 3.2.B1: Example of 'getreporttext' query



```
UPDATE demo SET report_text = $$<p>test report text</p>$$, tags = (null)
WHERE primaryid = 130776901
```

Figure 3.2.B2: Example of 'savereporttext' query

3.2.C Database

Currently, annotations only require two rows within the "Reports" table, 'report_text' and 'tags'. 'report_text' holds the actual narrative and the user comments, all contained in a single block of HTML formatted string, and is null when empty. The 'tags' row contains a json object that holds all of the tag key-value attributes of that particular report, including the locations and type of the highlighted text in the report narrative.

3.3 Document Search

For our project, after considering multiple libraries for report text searching/indexing, we chose Elasticsearch (ES). We considered building the search tool from Lucene, however we changed our plan and chose Elasticsearch as it allows us to save time and still retain the powerful features Lucene provides.

3.3.A Technologies Considered

PostgreSQL Full Text Search

PostgreSQL can support full text searching/indexing natively. However, while it can satisfy our requirements, an implementation in PostgreSQL will need potentially intensive maintenance if the database schema changes, or if the project switches from PostgreSQL to another DBMS. Likewise, development in just SQL would be difficult to debug, and an implementation in a more "typical" programming language has the benefits of quick prototyping, editing, and testing. (PostgreSQL, 2018)

Apache Lucene

Apache Lucene is a powerful tool supporting many different kinds of indexing, searching, and document analysis. For example, it can perform single/multiple keyword searches either for exact matches, or for "fuzzy" matches. Lucene can also perform multiple-index searches, which will be useful for creating a "similar reports" feature, as well as for efficiently updating the search index, as we would only need to index new reports added to the system, as opposed to re-indexing all reports. Additional features of Lucene which will be useful are:

- Ranking of search results, so that we can display the most relevant results first.

- Updating the index *while searching*.
- Low memory profile, and efficient, configurable storage.
- Open Source and free for use. All file formats are even fully documented, and there are easy ways to manipulate/explore the search structures Lucene creates.
- Lucene is fully implemented in Java, increasing portability, but is also available in other programming languages, such as Python, Ruby, and more. (Lucene, 2016)

Xapian

Another open source searching library is Xapian. While it includes many of the features of Lucene, there have been much fewer projects implemented with Xapian, which makes finding code examples of certain features more difficult. Likewise, we are developing on Windows and Unix for a Linux environment, so portability is important for collaboration, as well as for the future of this project, however the Xapian bug list contained multiple features not working properly caused by MSVC compilation on Windows. (Xapian, 2018)

Elasticsearch (ES)

Elasticsearch (ES) is a RESTful search framework built over Lucene. ES provides all of the functionality of Lucene, while simplifying implementation for us, and offering a Domain Specific Language (DSL) library for Python. This is particularly useful for us, as the DSL allows us to perform powerful operations in very few lines of code. In terms of performance speed, Elasticsearch automatically configures itself to run on *shards*, splitting up the document index into different locations which can all be indexed simultaneously; and *nodes,* which are instances of the elastic server that are run on different machines, and which contain their own shards. This automatic, configurable, and scalable deployment will be useful for the FDA, as their document collection is quite large. (Elasticsearch, 2019)

3.3.B Structure of Report Search

Since ES's index is *separate* from the database, and the reports do not have titles, only text, we "titled" each report with its primary key from the database in order to establish a relation between ES ↔ PostgreSQL. When building the index, we store the report text and the unique report "title" in the index, along with age, sex, and report text, and more fields which the user will likely wish to search on.

3.3.C Handling the Search String

To provide a suite of useful features to the user, we implemented a few niceties and specially handled control sequences to be entered in the search box:

Fuzzy Search:

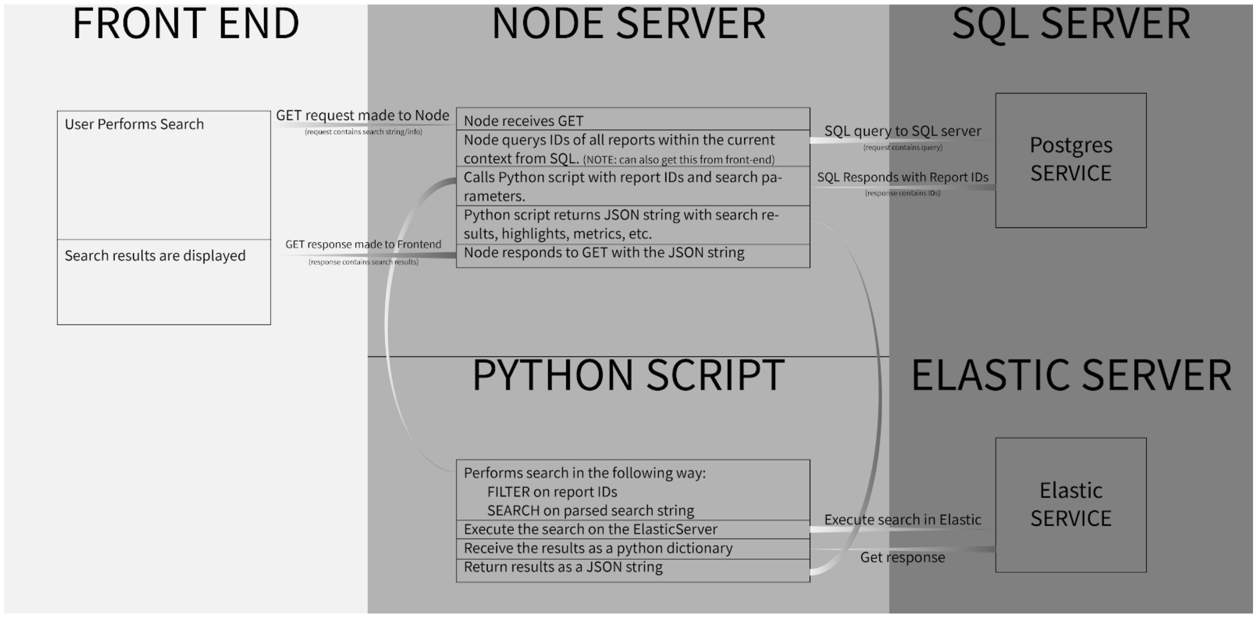
The default search method uses "fuzzy" matching. Fuzzy search helps locate relevant results which do not directly match the search term. For example, with a fuzziness of 2, if a user searches "Treat", then the results would contain some documents which do not exactly contain "Treat", but which may contain "Treats", "Treated", etc., but would not contain "Treatment" (because 'ment' is 4 characters, which surpasses the fuzziness of 2). It is important to set a good value for the fuzziness parameter, which is why we use the “AUTO” parameter for fuzziness, which defaults to:

- 0 for strings of one or two characters
- 1 for strings of three, four, or five characters
- 2 for strings of more than five characters

Negative Search:

Similar to how Google’s search allows you to enter “Fruit” to see all fruit, and “Fruit -banana” to see all fruit EXCEPT for bananas, we implemented negative search as well. A user can enter “Aspirin” to see all documents which mention Aspirin, and “Aspirin topical !oral” to help filter onto the documents in which a patient may have used aspirin without swallowing it. As a design decision, we chose to use the ‘!’ symbol instead of the ‘-’, as we want to allow the FDA to search for words which may begin with a dash; these include some medications with spaces and dashes in their names, or if a doctor lists suffixes, and each of them begin with a ‘-’. Although this case is unlikely, we considered that the ‘!’ will never be intentionally searched for.

3.3.D Architecture of Search Pipeline


FRONT END
NODE SERVER
SQL SERVER
User Performs Search
GET request made to Node
(request contains search string/info)
Node receives GET
Node querys IDs of all reports within the current context from SQL. (NOTE: can also get this from front-end)
Calls Python script with report IDs and search parameters.
Python script returns JSON string with search results, highlights, metrics, etc.
Node responds to GET with the JSON string
SQL query to SQL server
(request contains query)
SQL Responds with Report IDs
(response contains IDs)
Postgres SERVICE
Search results are displayed
GET response made to Frontend
(response contains search results)
PYTHON SCRIPT
ELASTIC SERVER
Performs search in the following way:
FILTER on report IDs
SEARCH on parsed search string
Execute the search on the ElasticServer
Receive the results as a python dictionary
Return results as a JSON string
Execute search in Elastic
Get response
Elastic SERVICE

3.3.D1 Diagram of data flow through the search pipeline when a search is made

When a search is made from the front end, the search string is sent to our Node.js server, which loads the user’s document context (the pool of documents they intend to search on), and then calls our ES Python script with a JSON string containing the search string and context. The script then executes the search on the elastic server, and returns a JSON string containing all of our search results and some metadata (eg, how long the search took on the elastic server) to the node server, which sends this data to the front end to be displayed.

We use a Python *script* instead of a Python RESTful *server*, as the time it takes to do any initializing and connect to the Elastic server is negligible, and using a script allows us to quickly make changes to the script and have it be immediately available to the users - without any overhead.

3.3.E Structure of "Recommendations"

We considered various approaches to creating the functionality of "from all reports R, given a set X of reports, return a set Y of similar reports". We implemented this by presenting the user with their highlighted keywords (and their counts) from X, and allowing the user to toggle the keywords in/out of the search string used to yield Y. This affords the user the ability to fine-tune the similar reports, while still being a quick and easy tool. For more details refer to Section 3.1.B above.

3.4 Deploy Management & Continuous Integration

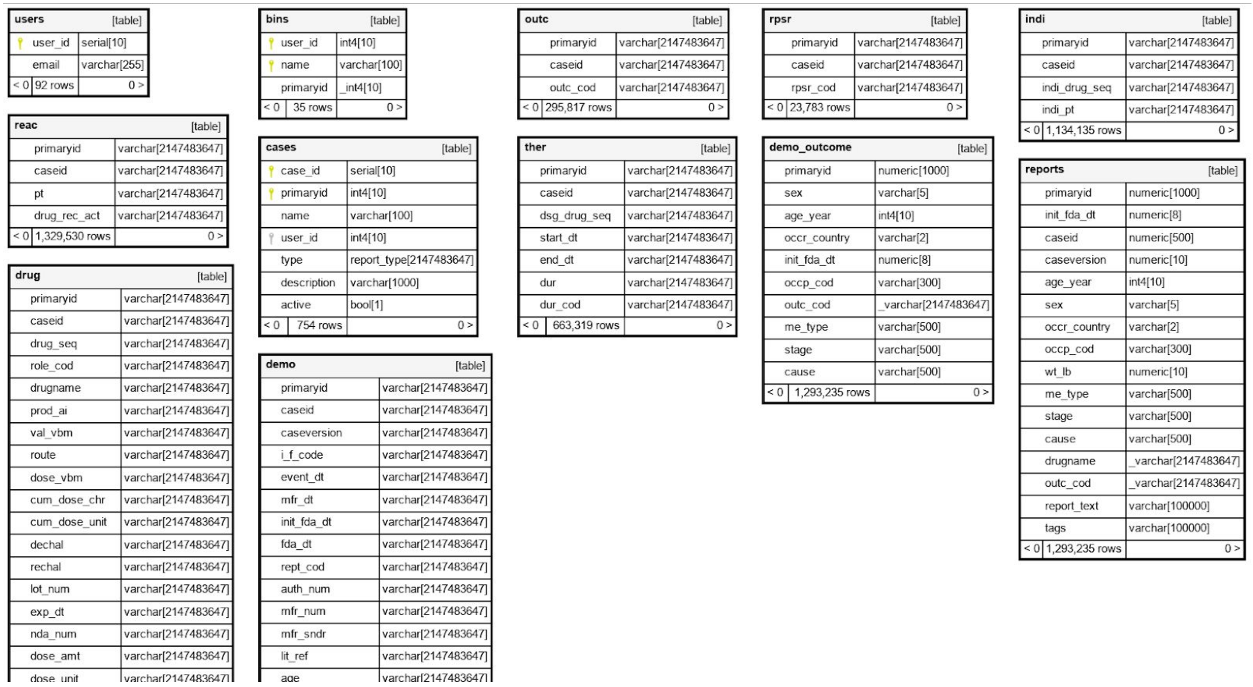
3.4.A Jenkins Integration

We use Jenkins on Linux to quickly update and deploy the website, while also providing a simple way to run and monitor builds from any device. Our Jenkins server, which runs on port 8080, automatically pulls from git and executes the proper commands to cleanly re/launch the website all with the click of a button. Jenkins has been very useful to check the status of each launch both throughout production, and now in deployment. In earlier versions, we also used Cobertura and jest to get code coverage, to spot and clean deprecated code from the previous MQP.

3.5 Performance Optimizations

3.5.A Database Optimization

The database previously consisted of many tables which had keys referencing other entries in other tables, but had no formally declared relationships/constraints. In this way, the 'tables' functioned like lists, which is very slow, and does not benefit from the very many performance optimizations afforded to us with modern DBMS systems.



users	[table]
user_id	serial[10]
email	varchar[255]
< 0 92 rows	0 >

reac	[table]
primaryid	varchar[2147483647]
caseid	varchar[2147483647]
pt	varchar[2147483647]
drug_rec_act	varchar[2147483647]
< 0 1,329,530 rows	0 >

drug	[table]
primaryid	varchar[2147483647]
caseid	varchar[2147483647]
drug_seq	varchar[2147483647]
role_cod	varchar[2147483647]
drugname	varchar[2147483647]
prod_ai	varchar[2147483647]
val_vbm	varchar[2147483647]
route	varchar[2147483647]
dose_vbm	varchar[2147483647]
cum_dose_chr	varchar[2147483647]
cum_dose_unit	varchar[2147483647]
dechal	varchar[2147483647]
rechal	varchar[2147483647]
lot_num	varchar[2147483647]
exp_dt	varchar[2147483647]
nda_num	varchar[2147483647]
dose_amt	varchar[2147483647]
dose_unit	varchar[2147483647]

bins	[table]
user_id	int4[10]
name	varchar[100]
primaryid	_int4[10]
< 0 35 rows	0 >

cases	[table]
case_id	serial[10]
primaryid	int4[10]
name	varchar[100]
user_id	int4[10]
type	report_type[2147483647]
description	varchar[1000]
active	bool[1]
< 0 754 rows	0 >

demo	[table]
primaryid	varchar[2147483647]
caseid	varchar[2147483647]
caseversion	varchar[2147483647]
i_f_code	varchar[2147483647]
event_dt	varchar[2147483647]
mfr_dt	varchar[2147483647]
init_fda_dt	varchar[2147483647]
fda_dt	varchar[2147483647]
rept_cod	varchar[2147483647]
auth_num	varchar[2147483647]
mfr_num	varchar[2147483647]
mfr_sndr	varchar[2147483647]
lit_ref	varchar[2147483647]
age	varchar[2147483647]

outc	[table]
primaryid	varchar[2147483647]
caseid	varchar[2147483647]
outc_cod	varchar[2147483647]
< 0 295,817 rows	0 >

ther	[table]
primaryid	varchar[2147483647]
caseid	varchar[2147483647]
dsg_drug_seq	varchar[2147483647]
start_dt	varchar[2147483647]
end_dt	varchar[2147483647]
dur	varchar[2147483647]
dur_cod	varchar[2147483647]
< 0 663,319 rows	0 >

rpsr	[table]
primaryid	varchar[2147483647]
caseid	varchar[2147483647]
rpsr_cod	varchar[2147483647]
< 0 23,783 rows	0 >

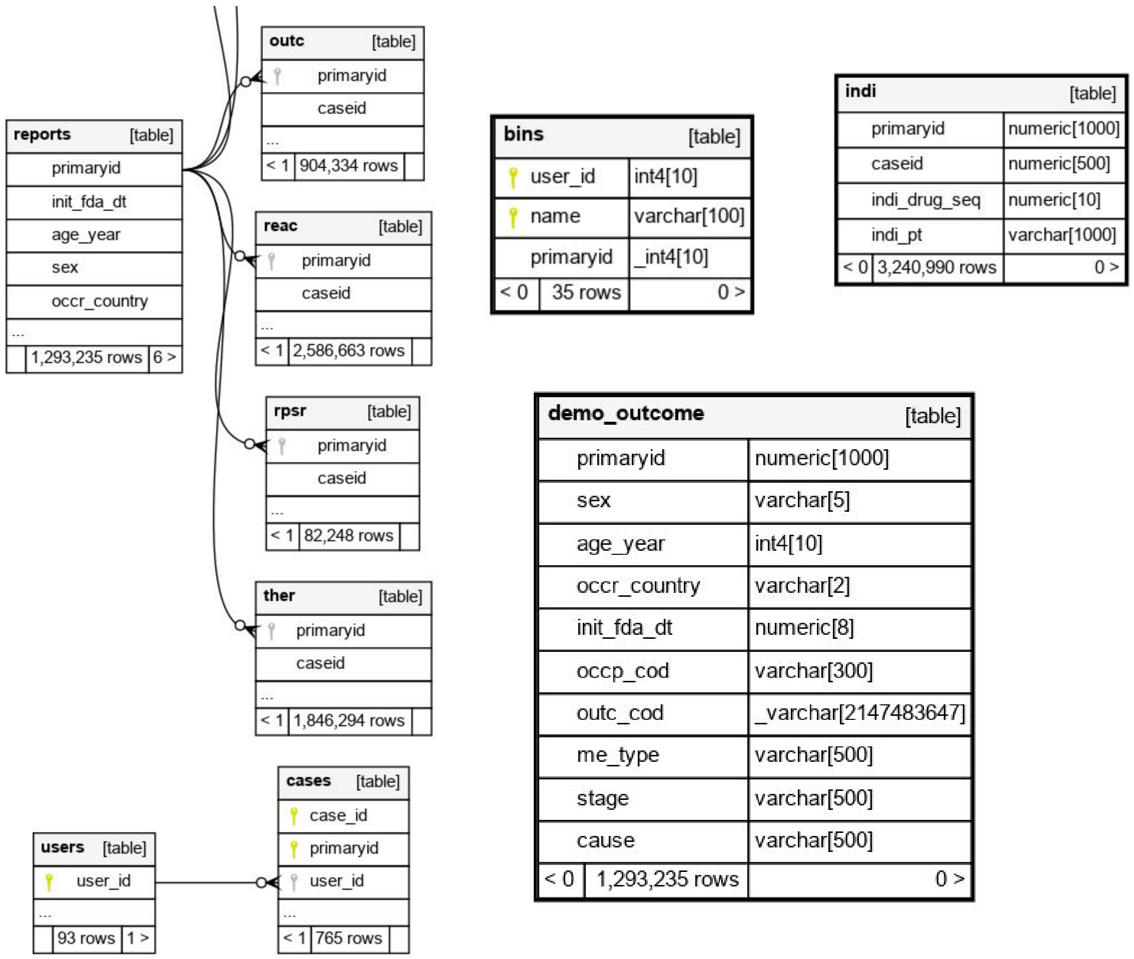
demo_outcome	[table]
primaryid	numeric[1000]
sex	varchar[5]
age_year	int4[10]
occr_country	varchar[2]
init_fda_dt	numeric[8]
occp_cod	varchar[300]
outc_cod	_varchar[2147483647]
me_type	varchar[500]
stage	varchar[500]
cause	varchar[500]
< 0 1,293,235 rows	0 >

indi	[table]
primaryid	varchar[2147483647]
caseid	varchar[2147483647]
indi_drug_seq	varchar[2147483647]
indi_pt	varchar[2147483647]
< 0 1,134,135 rows	0 >

reports	[table]
primaryid	numeric[1000]
init_fda_dt	numeric[8]
caseid	numeric[500]
caseversion	numeric[10]
age_year	int4[10]
sex	varchar[5]
occr_country	varchar[2]
occp_cod	varchar[300]
wt_lb	numeric[10]
me_type	varchar[500]
stage	varchar[500]
cause	varchar[500]
drugname	_varchar[2147483647]
outc_cod	_varchar[2147483647]
report_text	varchar[100000]
tags	varchar[100000]
< 0 1,293,235 rows	0 >

3.5.A1: A screenshot of a graphical representation of the original database. Notice that all tables are orphan tables, and that most tables lack any form of key.

To fix this, we ran various queries to discover the underlying structure and relationships of the data/tables, and used our findings to properly add constraints like uniqueness and/or foreign keys, as seen below.



reports	[table]
primaryid	
init_fda_dt	
age_year	
sex	
occr_country	
...	
1,293,235 rows	6 >

outc	[table]
primaryid	
caseid	
...	
< 1	904,334 rows

reac	[table]
primaryid	
caseid	
...	
< 1	2,586,663 rows

rpsr	[table]
primaryid	
caseid	
...	
< 1	82,248 rows

ther	[table]
primaryid	
caseid	
...	
< 1	1,846,294 rows

users	[table]
user_id	
...	
93 rows	1 >

cases	[table]
case_id	
primaryid	
user_id	
...	
< 1	765 rows

bins	[table]
user_id	int4[10]
name	varchar[100]
primaryid	_int4[10]
< 0 35 rows	0 >

indi	[table]
primaryid	numeric[1000]
caseid	numeric[500]
indi_drug_seq	numeric[10]
indi_pt	varchar[1000]
< 0 3,240,990 rows	0 >

demo_outcome	[table]
primaryid	numeric[1000]
sex	varchar[5]
age_year	int4[10]
occr_country	varchar[2]
init_fda_dt	numeric[8]
occp_cod	varchar[300]
outc_cod	_varchar[2147483647]
me_type	varchar[500]
stage	varchar[500]
cause	varchar[500]
< 0 1,293,235 rows	0 >

3.5.A2: A screenshot of a graphical representation of the new database. All but 3 tables are related, and most tables have a primary and/or foreign key.

The performance improvement of our database changes has been drastic.

3.5.B React Schema Rework

The previous MQP built their system with React, a feature we have maintained in our release, however there were several issues with their React schema that slowed the system. The previous group mixed different techniques of implementation with React. Some components were using a "Will" schema and others were using a "Did" schema. The difference between the two is that the Will schema works via preemption of an update to control the components while the Did schema reacts to actions to control components. When you mix the two you will often get undesirable effects, most commonly taking the form of excessive calls to the same function at different intervals during a process call.

The most rampant example of this was within the report table where they had the methods for retrieving reports in both a Will and Did React function that resulted in several queries to the server to retrieve the same reports. The problem here is that the site very quickly ran into memory issues for the user and could reach CPU usage ranges in the high nineties. This level of rigor on the hardware slowed the system quite drastically. To remedy this, we changed the React Schema to be a thoroughly grounded in a Did schema, using Will components only where it was absolutely necessary. The results is a much more controlled and regulated interaction between the site, the server, and the user’s hardware—it also made the website run much faster.

4. Evaluation

In order to test the usability of the web application's new and improved features, the team decided to hold a User Study. The main goal of this study would be to test how intuitive our new system is when used by a participant with no prior experience with the application.

4.1 User Study Structure

For this study, we determined that the best method of understanding the usability of the system would in the form of a live study in which a participant would go through numerous different features and answer both quantitative and qualitative questions on tasks they were assigned. Since the vast majority of the new features and changes made were within the Reports Page, the study would be focused on this section alone.

Participants were chosen at random from amongst WPI's student population, providing us with a diverse group of individuals from different backgrounds and technical skill. After discussing the project with WPI's Institutional Review Board, it was determined that since we were not requiring the participant to provide any personal information that would be used in the study outside of a short demographic survey, we would not need to provide them with a written consent form before starting the study. Instead, we simply told them that this study was by no means necessary, and that they could choose to discontinue at any time.

When a participant agreed to partake in the study, the first step was to provide them with a short tutorial on ConText, namely what ConText is and providing them with at least some reference for the features that are available within the system. This tutorial would also explain that the user would be focused on the Reports Page, and provide a short overview on the different aspects of the Reports page. (Appendix E) After this tutorial was completed, the participant would then be asked to fill out a short survey in order to get their general demographic, i.e. age, gender, and previous education. (Appendix A)

Finally, the user would then begin the user study itself. The study was broken up into 5 separate sections, or Tasks, each one having a varying amount of tasks and questions to complete.

1. **Building a Case/Getting Started:** This Task is where the participant first enters into the system and is familiarized with building a case and using our new search engine. The tests are meant to see how easily the participant is able to determine how to create new cases, and the different methods by which they can add reports into their case.
2. **Marking Reports:** Here, the participant is asked to make use of different methods of interacting with reports in a case, namely by marking reports as "Primary Evidence" and annotating the narrative of a report. This is to show them how interacting with reports will cause changes not only in the reports themselves, but also in the Case Summary panel, which will change depending on the status of the reports within it.

3. **Summarizing a Case:** This section is meant to show the participant more on how they can alter the case, namely by seeing what happens when they remove a report from a case. Here, the participant will see changes in both the D3 graph and the Keyword Summary as they remove reports from the case.
4. **Dummy Test/Summarizing and Adding to a Larger Case:** This test is focused on furthering the participant's understanding of the Case Summary panel by having them look for specific information and utilizing the Get Recommendations feature.
5. **Commenting On a Report:** Finally, this section tests the usability of the new commenting system, and had the participant interact with an already commented report, and has them leaving their own comment, thus going through all of the workflow paths for working with comments.

Each section contained a list of smaller tasks for the participant to complete, as well as some quantitative questions that could be attached to a particular task and were to be completed in parallel. At the end of each set of tasks, the participant would be asked to answer some qualitative questions, providing their opinion on the effectiveness or ease of use of the tested features on a scale from 1 (Unintuitive/Hard) to 5 (Intuitive/Easy). During examination, in the event that a participant was stuck or asked a question, the investigator would be allowed to provide a small hint in order to help them along. For example, occasionally a user would be unable to locate the Report Count indicator at the top of the screen, so if they were struggling to find it or asked, the investigator would be allowed to say something to the effect of "The count is visible at this time," which usually was enough to expand their view and bring them to the correct answer. In such instances, we would mark down any hints or guidance that were provided, which in turn gave us valuable information on which features were most difficult for participants, as well as insight into potential fixes that could be done in the future. (Appendix B)

In addition to recording the answers to these questions, investigators also kept track of the time that it took for a participant to complete each section, starting from the start of the first task and ending on them beginning the qualitative questions for that section. Investigators also recorded the state of the Reports Page at the end of each section, such as whether or not they had the correct panels open or saved their annotations. (Appendix C)

Once the user ended the study, we offered them the chance to enter their email for a free $30 Amazon Gift Card. For obvious purposes, these emails have been kept confidential and are not included in the notes.

4.2 Evaluation Results

By the end of the Evaluation phase, 15 participants had taken the User Study. For the most part, it appears that participants were able to get a handle on the system fairly quickly.

All of the results of our data collection is available in Appendix C: Data Table.

Time wise, most of our participants were able to get through the examination fairly quickly, although there are fluctuations in terms of how quickly an individual participant completed individual Tasks.

Task	Building a Case /Getting Started	Marking Reports	Summarizing a Case	Dummy Test /Summarizing and Adding to a Larger Case	Commenting on a Report	Total Time
Avg Time (Min:Sec)	5:32	2:15	0:56	3:46	1:19	13:50

Table 4.1: Table of Completion Times. This table shows the averages time that it took participants to complete individual tasks, as well as the total time it took for them to complete the User Study.

For the qualitative questions, most participants were able to come to the correct answer. For example:

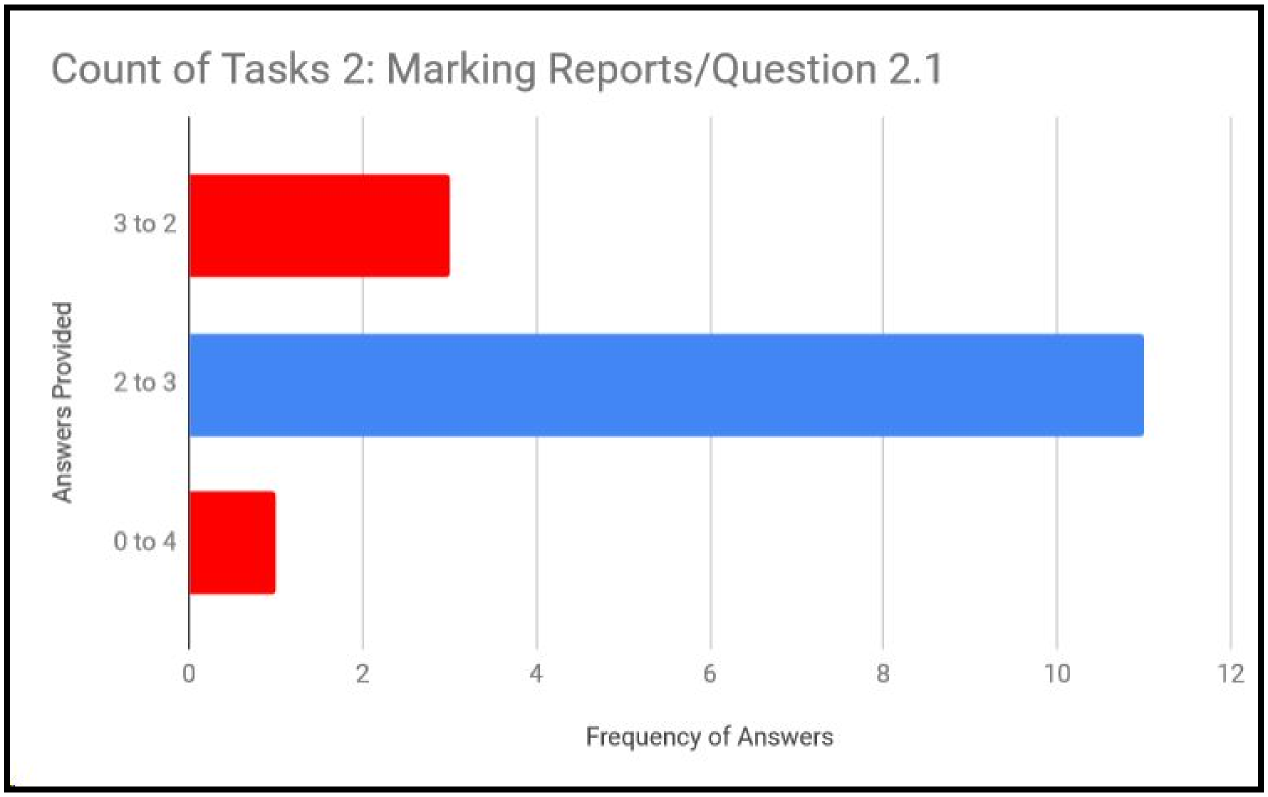


Figure 4.2.1: Results of Task 2 Question 2.1, "What is the current ratio between Primary and Supportive reports as indicated by the graph?" The correct Answer is "2 to 3"

The above graph shows participant's answers to the first question of the Marking Reports Task, in which they are asked to report the ratio of Primary Reports to Supportive Reports after making 2 out of 5 reports Primary. Specifically, the user is meant to view the Case Summary Panel and find the D3 graph that shows the proper ratio. The correct answer to this question is "2 to 3", which we see that most participants (11/15) were able to discern for themselves through the system. In addition, the second most common answer is "3 to 2" (3/15), which while not exactly the correct answer, shows that the user was at least able to discern how to get the needed information, but either misread the question or misinterpreted the graph. (Figure 4.1)

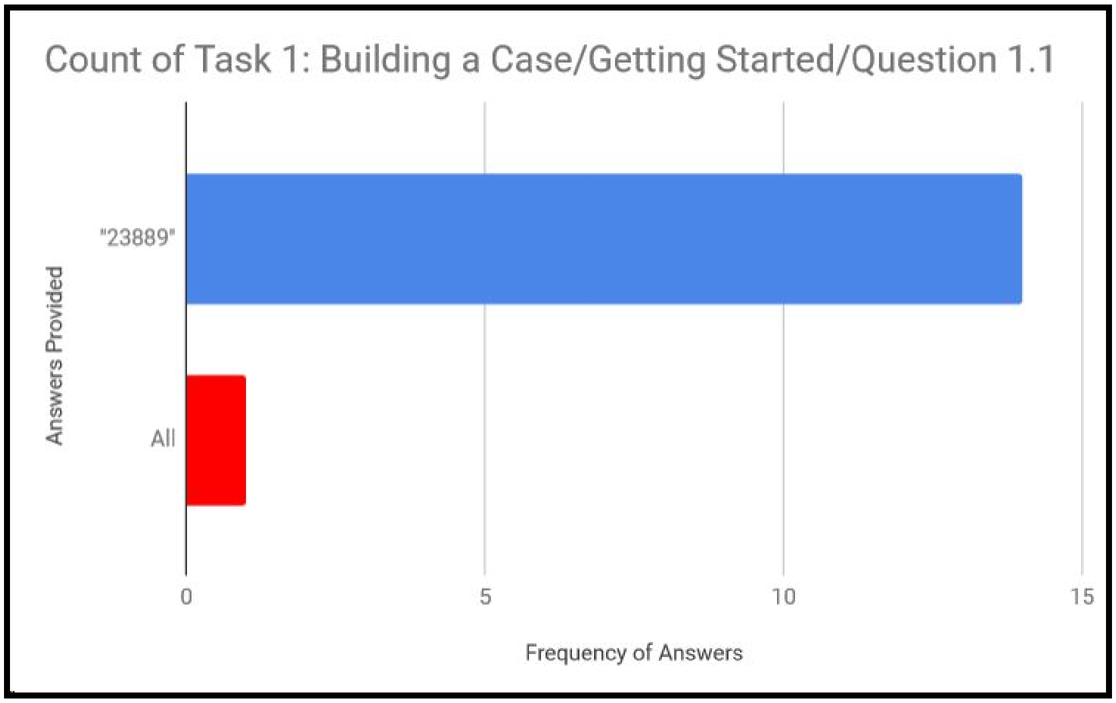


Figure 4.2.2: Results Question of Task 1 Question 1, "How many reports are made available to the user when you first entered the reports page? (hint: there is an indicator on the page)" The correct answer is "23889"

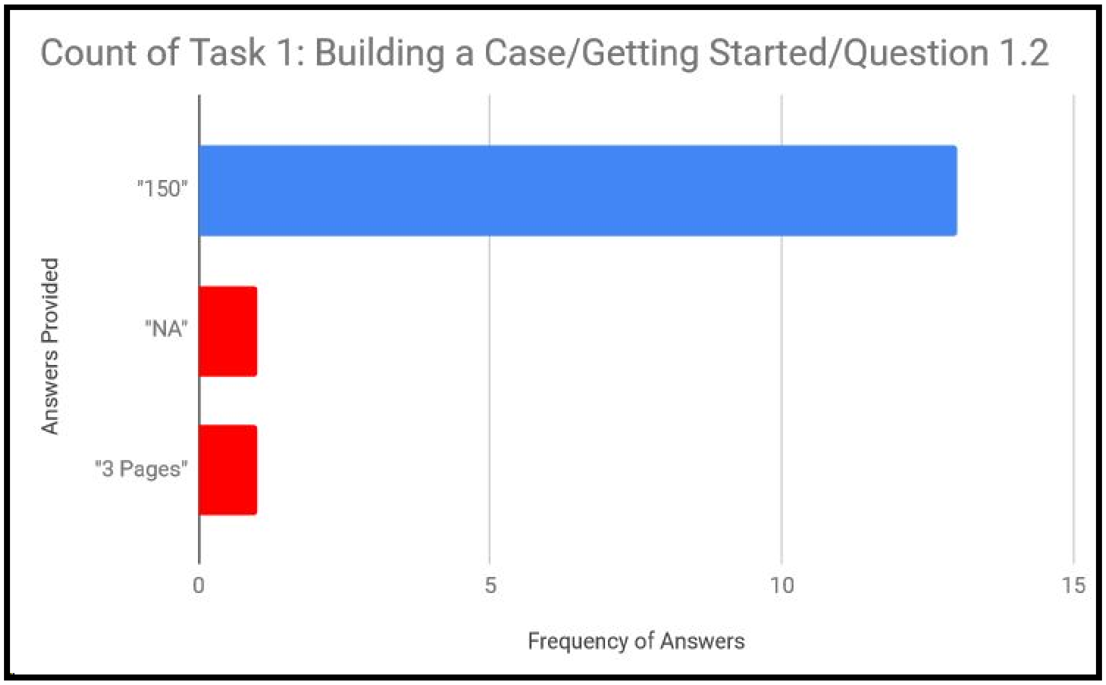


Figure 4.2.3: Results Question of Task 1 Question 2, "How many search results appeared?" The correct answer is "150"

For other questions, participants would occasionally need to be provided a hint for them to find the correct answer. Above, we see the results of two other Qualitative questions, the first of which the user needed to identify how many reports they currently had access to when the first opened the reports page, the second being to identify the number of search results they get when they enter the word "patient" into the search bar. For the first question, the correct answer to this question was "23889", which most participants (14/15) were able to successfully identify. Only one seemed to not understand the question, or did not see the proper count of reports. Ideally, this would have been accomplished by looking at the top of the screen and finding the "Report Count" tab. However, as we discovered and recorded in our notes, some people did not seem to see this count when they first opened the system. When they asked about this, the proctor usually would provide a simple hint, such as reaffirming that the count was visible. In many cases, we found that this almost immediately provided our users with the insight to find the count, and record the correct answer. This was much the same for second question, in which the correct answer was "150." Similarly, the participant would occasionally need a slight hint, but then was able to find the answer on their own, with 13/15 getting the answer correct. (Figure 4.2.2: Figure 4.2.3)

In other examples, we see that the participants, while not finding the exact answer, appears to at least be near the correct answer.

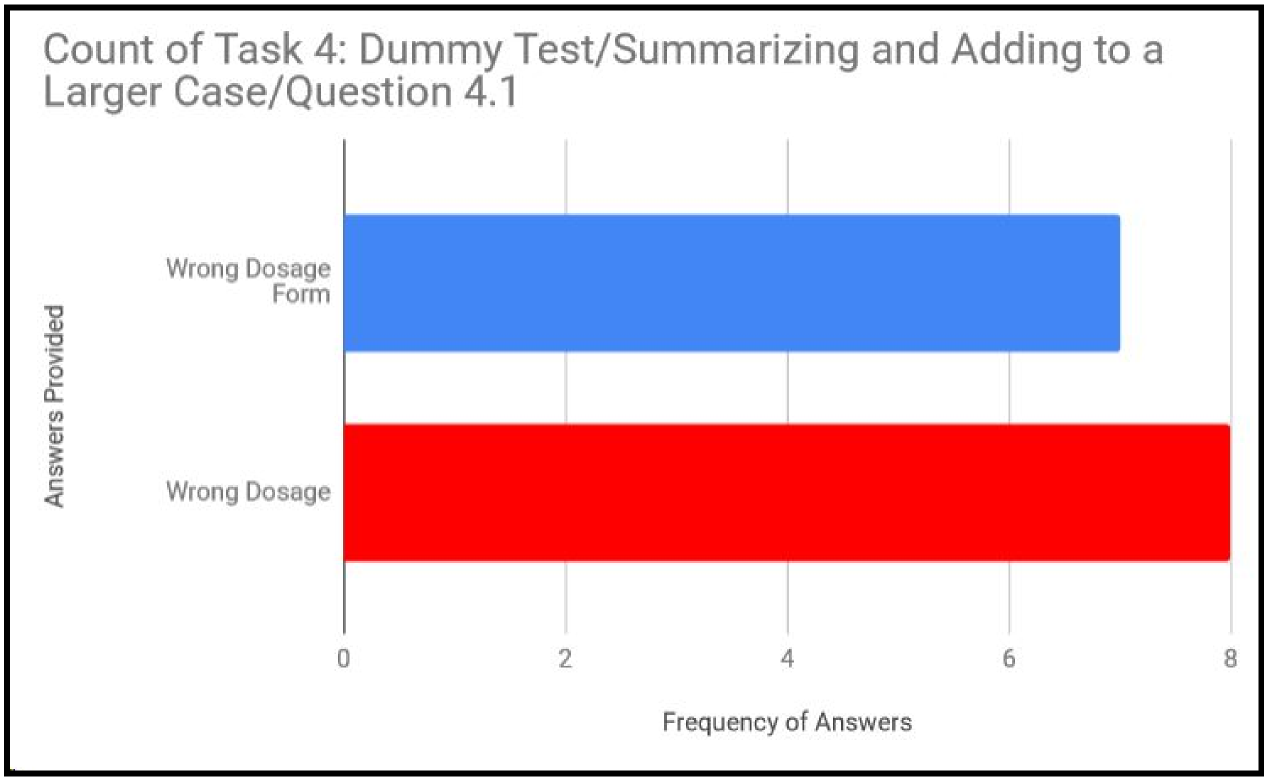


Figure 4.2.4: Results of Task 4 Question 1, "Which medication error is most prevalent?" The correct answer is "Wrong Dosage Form"

In the above graph, we see the answers to Question 1 in Task 4. This question asks the user to use the D3 graph again, this time switching the mode from Primary vs. Supportive to checking the distribution of Medical Errors. The correct answer to this question is "Wrong Dosage Form," and as we can see above, we have received some interesting results. The most common answer to this question was "Wrong Dosage" (8/15) and the second most common answer was "Wrong Dosage Form" (7/15). While "Wrong Dosage" is not entirely correct, it shows us that the participant was able to find the necessary information in the system, and was able to at least provide a partially correct answer. In addition, there were no other answers for Medication Errors that didn't have anything to do with "Wrong Dosage," meaning that there were no completely incorrect answers to this question. (Figure 4.2.4)

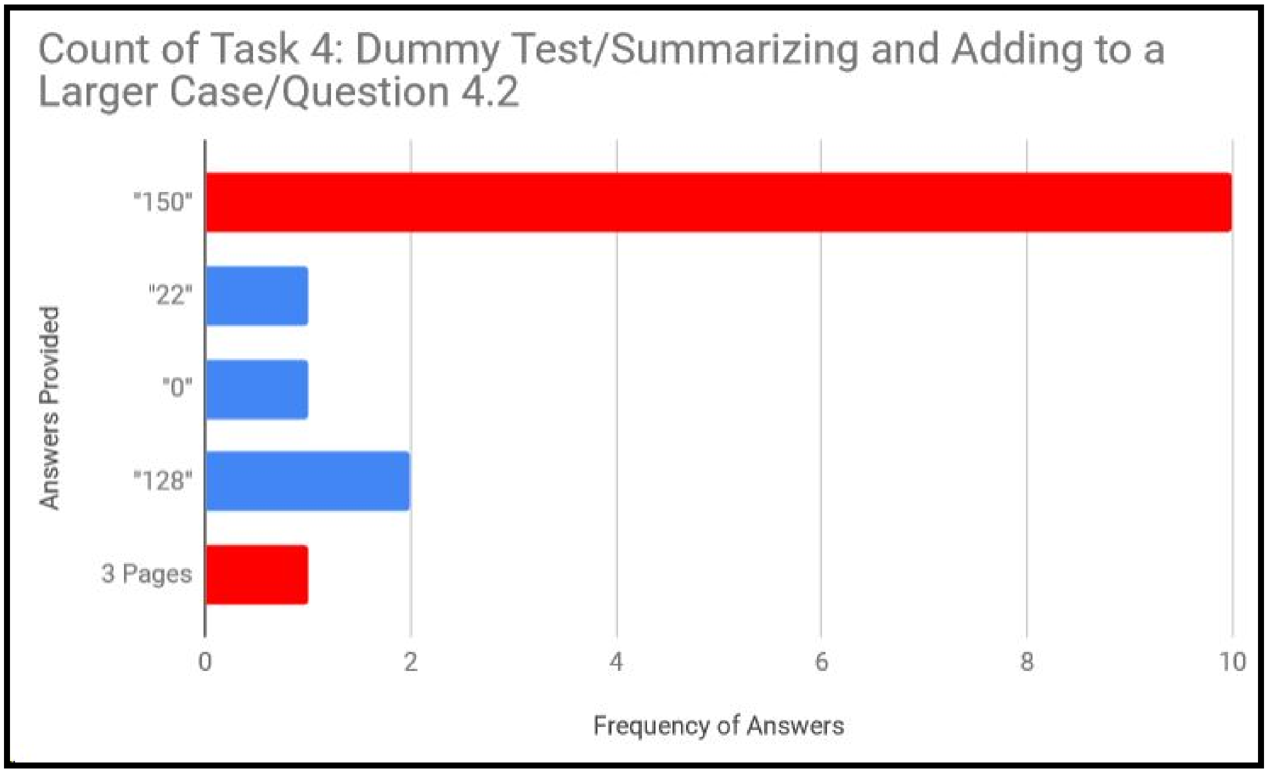


Figure 4.2.5: Results of Task 4 Question 2, "How many Recommended Reports are currently available after removing keywords?" The correct answer was anything less than 150, as long as it is a number.

For question 2 in Task 4, this question was meant to check the search results feature results, only this time when applied with the Get Recommendations Tool. For this question, the goal was to test what would happen when a user reduces the number of keywords involved in a query so as to test whether or not the query can help narrow down reports. This question was to see if a participant could change the search result amount significantly through the Get Recommendations. For the most part, participants seemed to receive a full list of returned reports (10/15), but occasionally they would return less. As we can see, this is indeed the case, showing us that the tool is, in its current state, a potential means of narrowing down related reports and finding potential links. However, we can also clearly see that the tool still needs further tuning, as we see that most participants were not able to reduce the amount results from their search. (Figure 4.2.5)

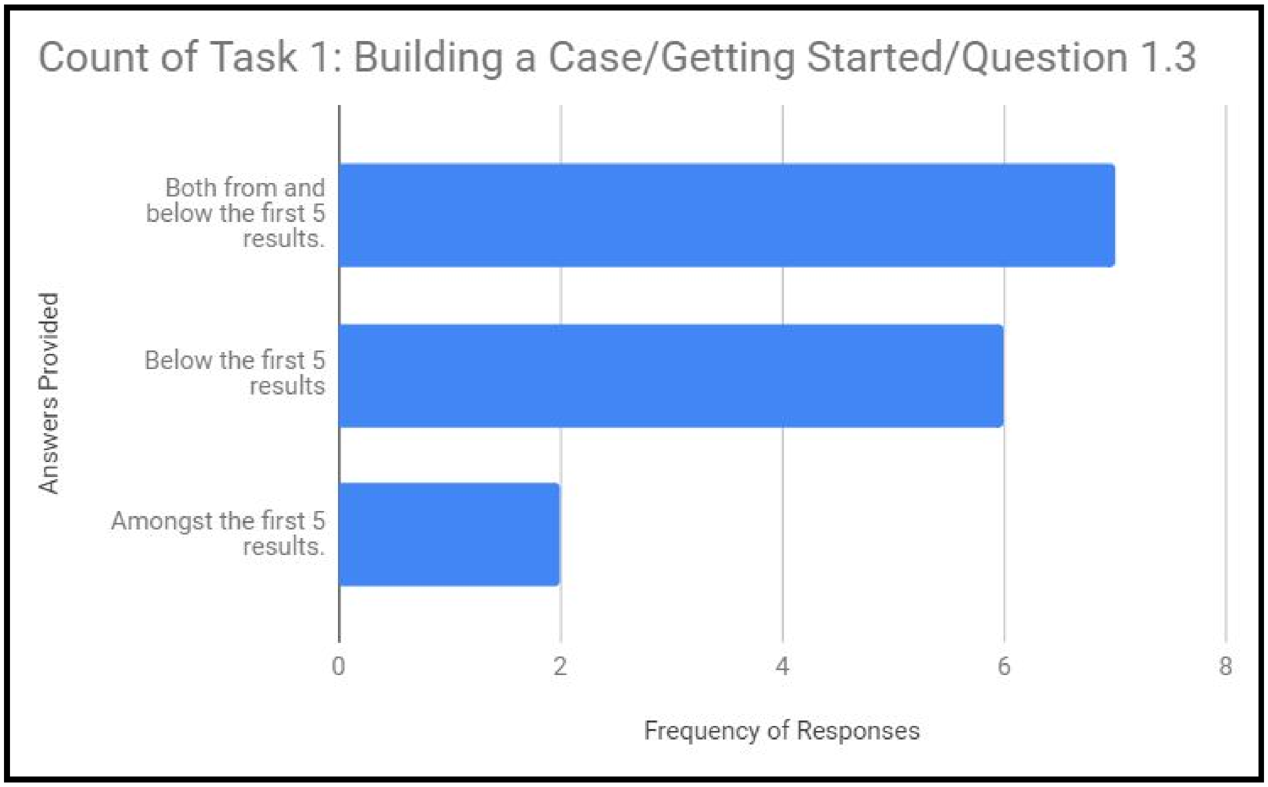


Figure 4.2.6: Results of Task 1 Question 3, "When choosing your reports, from which of the following groups did they belong to?" There is no correct answer to this question, as its purpose was to check to see how participants determined which results were important.

For the rest of the quantitative questions, these were less about providing a correct answer, and more of getting an idea for how the participant thinks. For example, in the table below, we see the answers provided to the question "When choosing your reports, from which of the following groups did they belong to?" that relates directly to the task of searching through the reports with the search engine using the work "patient." When the participant entered this search query, they would receive a list of results, and then have to select five of them to put into case Patients. As we see above, there were a variety of strategies that a participant had for choosing their reports, particularly choosing from both from and outside the first 5 search results. This question was meant to provide us with an idea for what users would do when provided with search results, and how they would go about determining what is relevant and what is not. This is then expanded upon in the next question, where they provide an extended answer. From these answers, we see that most users went off of the information within Report Table, rather than from the Report Panel, or that they simply chose at random. (Figure 4.2.6)

In terms of the qualitative questions, we have found that people responded positively to most features. As said before, qualitative answers were put on a scale of 1 to 5, with 1 representing unintuitive/difficult and 5 representing intuitive/easy. In the table below, the qualitative questions that participants were asked is listed, along with the averages of all 15 participants' responses.

Task 1: Building a Case/Getting Started:	
How intuitive was it to navigate to the reports section?	4.0667
How easy was it to find reports related to the query?	4.1333
How easy was it to add reports to the case?	3.6667
How difficult was it to get to the reports within the case?	3.6
Task 2: Marking Reports	
How intuitive is the primary/supportive graph provided?	4
How intuitive is it to annotate a report?	3.6
Task 3: Summarizing a Case	
How easy was it to remove a report?	4.9333
Task 4: Dummy Test/Summarizing and Adding to a Larger Case	
How intuitive is it to interact with recommended reports?	3.7333
How insightful are the case summary demographic distribution graphs?	3.8
How intuitive is the keyword selector?	3.8
How insightful is the keyword distribution?	4.3333
How relevant are the Recommended Reports to the entered Keywords?	3.7333
How intuitive is it to manipulate the Query?	3.6
How useful are the recommended reports?	4.0667
Task 5: Commenting On a Report	
How easy was it to remove comments?	4.9333
How easy was it to add comments?	4.2
How useful do you think comments would be in this tool?	4.8667

Table 4.2: Table of Qualitative Results. This table shows the different qualitative questions that we asked out participants on the left and shows the average of all of their answers on the right.

For the most part, we were able to keep our grades around 3.5-5, which tells us that most participants found the improved system to be intuitive, but that some improvements to the system are possible. (Table 4.2)

4.3 Evaluation and Discussion

Our results provide us with a great deal of information about how well our participants were able to adapt to ConText, namely that while the current system is functioning as expected, there is certainly room for future improvement.

When building the User Study, it was estimated that it would take approximately 15-20 minutes for someone to complete all of the tasks. For the most part, this estimation was accurate, and we found that many participants were able to complete the examination even before the 15 minute point. However, as we can see above, there are certainly fluctuations in terms of how long it took participants to complete each task. For example, Task 1 was longer than the other tasks as it required the participant to do more so that they could become more familiar with the system. On average, most participants took approximately 5 minutes to complete the task, showing they were able to get a handle on the new system in a fairly short amount of time. However, others took more time to complete Task 1, sometimes even as high as 8-9 minutes. This type of fluctuation occurs across all of the Tasks, showing us that while some were able to get a grasp on the functionality fairly quickly, some had more difficulty understanding parts of the system. (Table 4.1)

During the study, the parts that we found participants to have the most trouble with were in identifying the Case manipulation menu that was available in every Report Row, first in that it may not be too clear that the menu is there and that there was not enough feedback provided when a user moved a report to a Case. The Get Recommendations feature was also a place where participants seemed to struggle, as they would occasionally be confused about where it was located and how to tell what words were selected or not. In the qualitative questions, when asked to rate their feelings on adding reports, participants only gave it a 3.6666, and when asked their feelings about the intuitiveness of the Get Recommendations feature, answers stayed around 3.6-3.8. These were the sections that most participants seemed to take the longest to complete, and also asked the most questions on during the study.

In addition, it would appear that there are still some issues with the highlighting tool in terms of its intuitiveness, as it also received only a 3.6, meaning that further improvements to the annotation workflow will be necessary outside of just adding a tooltip. There are also some improvements that could be made to the workflow of adding comments. While adding comments received a fairly high score of 4.2, one of the most common complaints from participants was that it was unnecessary to have both a "Post" and "Save" button for the system. (Table 4.2)

Also, as said above, while most participants were able to come close to the desired answer, there were still several instances in which participants weren't entirely correct, usually providing an incomplete or partial answer to what we were expecting. This most likely indicates that our main issue is that while we are making a lot of information available to the user, there is still some work to be done in refining that information, and making it easier for the user to understand completely.

In conclusion, while these scores show that people had mostly positive experiences with our system, the overall results of our study show that our newest features are not completely ready, and will most likely require finer tuning in future iterations of the project.

5. Conclusion

5.1 Summary

For this project, our task was to update the MedVis system that had been created by last year's MQP.

We built on the previous project with improvements to:

- The Reports Page — Making a new 3-panel view that centralizes all information.
- The Case Summary Panel — Adding new views to the panel to help summarize Cases.
- The Narrative Annotator — Improving the efficiency and simplicity of viewing and annotating a report.

while adding entirely new functionality via:

- Search capability — Being able to enter a search string, and view relevant reports.
- Report comments — The ability to add public and private comments to reports.
- Recommended reports — Finding more reports relevant to reports currently in the case.

After conducting a study, we evaluated that our changes:

- Excelled in:
  - Simplifying the workflow of viewing and interacting with reports.
  - Improving the simplicity of the Reports Page
- Could be improved by:
  - Working more with the Case Summary Panel, especially its Front-End to improve interactivity.
  - Improving Comments UI functionality.

Through our efforts, the MedVis project has entered the next stage of its development, ConText, and has been brought closer to achieving its full potential as a lifesaving tool for the FDA.

5.2 Future Work

Much like the previous project team, our efforts were focused primarily on one portion of the project and while we believe we have addressed many of the issues with the system as a whole, there is still much to do. Due to time constraints and a lack of expertise in certain areas of development, certain features we had planned had to be pulled from the site. These features included a revision of the visualization page, implementation of Natural Language Processing (NLP) on the report text, and a revision of the Dashboard interface. Additionally, we have also written a subsection on potential avenues for development that we feel would benefit the project as a whole.

5.2.A Revisiting the Visualization

There are certain oversights with the implementation of the TreeMap found on the visualization page. The first being the limitation of not being able to fully show the user the categories within the tree map selections, particularly where the number of options is quite large (such as the drugs TreeMap). This results in the user only being able to see the top 15 or so drugs from within a pool of over 100 making the others inaccessible. To remedy this, we proposed the addition of a searchable list field in place of the TreeMap for drugs. While not as elegant as a TreeMap the primary concern here should be function, rather than neatness of the visualization. Unfortunately, due to scope constraints we were unable to address these concerns but feel it necessary to offer up to future teams to work on.

5.2.B Natural Language Processing

The introduction of NLP into the system surrounding reports would greatly bolster the efficiency of the report analysts. Our proposed uses for this technology are to automatically generate summaries of the report text for the user and to automatically recognize and highlight keyword annotations within the report text. The general thought surrounding this is that the primary purpose of the tool is to allow analysts to collect and go through the text of the report and shortening the text they need to review would increase efficiency. However, in order to become a true replacement, it is necessary to properly train the agent to generate effective summaries.

The biggest issue with this feature lies within this training. As it stands, we only have access to a collection of around 50 faux reports given to us by our sponsors so effectively training the model isn't feasible. And given that we lack knowledge of how to write reports that mimic the style of health care professionals and random generation may breed undesirable results. In order to truly implement the proposed NLP, it is imperative to train the model on the actual reports from the database.

5.2.C Dashboard Mechanisms

Another avenue for improvement on the system lies in an overhaul of the current dashboard interface. As it stands, the only unique thing the dashboard offers is the ability to set the user's cases as active or inactive. We believe that the best direction to take this portion of the interface is in a much more in-depth version of the case summary panel found within the reports page. Given that the dashboard has its own dedicated space and mechanism for managing cases it appears to be the optimum candidate for deep exploration of the cases.

In terms of the actual exploration of the reports, it may make more sense to draw on a number of different visualization strategies to follow intended areas of intrigue. To track the interactions between certain drugs within a case (or even the whole body of reports) it might make sense to look to something like the DIVA project. This project uses a force-directed graph to map the interactions between drugs and tracks the frequencies and magnitudes of these interactions. Other areas to look at may be additional views for tracking the distributions of errors, outcomes, etc. for a case. These views would offer more fidelity than would be achievable within the case summary. Without too much thought, a view could be a map of the distributions of errors and outcomes, the data could be run through a neural network to detect certain patterns and distributed with these groups as guides. Another view could encompass multiple cases to check commonalities between the two that may otherwise be overlooked in case specific analysis. There are many other possibilities here and these are just a few, regardless we feel that a revision of the dashboard into an analysis tool would greatly aide analysts.

5.2.D Other Areas of Interest

Aside from those previously mentioned, the system would benefit from additional features, such as:

- ‘Smarter’ search — for example having the search engine to understand that “advil” and “ibuprofen” are the same, and to automatically search on both.
- Security — limiting access to reports, and intelligently keeping track of what data has been delivered to each person. This is important, as although we are using publicly available data, the FDA is using data which should be kept secure.
- Full database rework — the database still contains redundant data, and could be drastically trimmed in size while reducing latencies. Likewise, combining the elastic index with the postgres database is a direction worth exploring. Elastic might be able to fully replace and improve the database currently in use.

References

1. *FDA Dashboard*, fis.fda.gov/sense/app/d10be6bb-494e-4cd2-82e4-0135608ddc13/sheet/7a47a261-d58b-4203-a8aa-6d3021737452/state/analysis.
2. Center for Drug Evaluation and Research. “Questions and Answers on FDA's Adverse Event Reporting System (FAERS).” *U S Food and Drug Administration Home Page*, Center for Drug Evaluation and Research, www.fda.gov/drugs/guidancecomplianceregulatoryinformation/surveillance/adversedrugeffects/default.Htm.
3. Cleveland, William S., and Mcgill, Robert. “Graphical Perception and Graphical Methods for Analyzing Scientific Data.” *Science* 229.4716 (1985): 828–833. Web.
4. “FDA Basics - FDA Fundamentals.” *U S Food and Drug Administration Home Page*, Center for Drug Evaluation and Research, www.fda.gov/AboutFDA/Transparency/Basics/ucm192695.htm.
5. “Get Started in the Cloud.” (2019) *Elastic*, www.elastic.co/.
6. Murphy, D., Spring, O.B., Tapply, C.M., & Yun, D. (2018). *Adverse Reaction Reports Analysis Tool* (Undergraduate Major Qualifying Project No. E-project-032218-172311). Retrieved from Worcester Polytechnic Institute Electronic Projects Collection: https://web.wpi.edu/Pubs/E-project/Available/E-project-032218-172311/
7. PostgreSQL, *PostgreSQL*, 2018: https://www.postgresql.org/
8. PostgreSQL (2018). “Full Text Search”: https://www.postgresql.org/docs/10/static/textsearch-intro.html
9. Lucene (2016). Apache Lucene. “Lucene Features”: https://lucene.apache.org/core/features.html
10. Xapian (2018). The Xapian Project. “Features”: https://xapian.org/features

Appendix A: User Evaluation Script

FDA MQP Evaluation Task List

Script:

1. Thank the participant for their interest, and assure them that this is completely voluntary.
2. Have the participant complete the survey for their background information.
3. Once the survey is complete, we will read an introduction to the participant, introducing them to the system and providing them with an overview of what to expect from the following evaluation.
4. Start the evaluation. While the participant is answering the following question, we will record:
    a. The amount of time it takes for the participant to complete each section.
    b. The number of times the proctor must assist the participant during a task.
    c. Their answers to each of our questions.
    d. The status of the screen at the end of each task. (To check accuracy)
5. When the participant has completed the evaluation, thank them for their time and offer them a chance to enter their email for the gift card.

Survey

What is your age?

1. 18-20
2. 21-29
3. 30-39
4. 40-49
5. 50-59
6. 60 or older

What is you gender?

1. Female
2. Male
3. Other: __________

4. Prefer not to answer

What is the highest level of school you have completed or the highest degree you have received?

1. Less than high school degree
2. High school degree or equivalent (e.g., GED)
3. Some college but no degree
4. Associate degree
5. Bachelor degree
6. Graduate degree

Have timer at ready to jot down times for each task.
Make sure to take note of the number of times the participant requires the proctor's assistance.
Make sure to note the status of the screen when the participant ends their each task.

Evaluation Section 1: Starting From a New Account

Building a Case/Getting Started

1. Create a new account.
2. Go into reports section.
   - **Question:** How many reports are made available to the user when you first entered the reports page? (hint: there is an indicator on the page)
     (**Ans:** 28,339)
3. Create a new case titled "Patients."
4. Enter "Patients" into the search engine and execute search.
   - **Question:** How many search results appeared?
     (**Ans:** 150)

5. Take any 5 reports from the search results that you feel are relevant and add them to case “Patients.”
    - **Question**: When choosing your reports, from which of the following groups did they belong to?

        1. Amongst the first 5 results

        2. Below the first 5 results

        3. Both from and below the first 5

    - **Question:** Why did you choose these particular reports out of the results?

6. Go to the case summary panel.

7. View the reports in the “Patients” case.

Difficult	How intuitive was it to navigate to the reports section?			Easy
1	2	3	4	5

Difficult	How easy was it to find reports related to the query?			Easy
1	2	3	4	5

Difficult	How easy was it to add reports to the case?			Easy
1	2	3	4	5

Difficult	How difficult was it to get to the reports within the case?			Easy
1	2	3	4	5

Marking Reports

1. Select 2 of the reports within case “Patient” and toggle them to primary.

2. Open the Case Summary Tab to the Left.

3. View the distribution of primary to supportive (using d3 graph).

- **Question:** What is the current ratio between Primary and Supportive reports as indicated by the graph?

  ___________ (**Ans:** Should be 2 Primary / 3 Supportive)

4. Navigate to 1st Report.
5. Annotate using the “Drug” highlight in the report.

Uninformative	How intuitive is the primary/supportive graph provided?			Very Informative
1	2	3	4	5

Unintuitive	How intuitive is it to annotate a report?			Intuitive
1	2	3	4	5

Summarizing a Case

1. Keywords should update in the Case Summary panel.
2. Remove a report from the case.
3. Go back to keywords; check that the keywords were removed.

   Question: Can you tell the information within the graphs has changed as a result of the report being removed?

   1. Yes
   2. No

Difficult	How easy was it to remove a report?			Easy
1	2	3	4	5

Section 2; Using an Existing Account

Dummy Test/Summarizing and Adding to a Larger Case

1. Log into a premade account, Username “EvalAcc”

2. Navigate to the premade case called “Target” in reports section.

3. View Case Summaries.

4. View distribution of medication error.

   Question: Which medication error is most prevalent?
   (**Ans**: Wrong Dosage Form)

5. Select Keywords to query for Recommended Reports.

6. Remove Keywords from query.

   Question: How many Recommended Reports are currently available after removing keywords?
   (**Ans**: Depends on which Keywords are removed)

7. Look for recommended reports.

8. Add the top recommended report to the premade case.

Unintuitive	How intuitive is it to interact with recommended reports?			Intuitive
1	2	3	4	5

Not Insightful	How insightful are the case summary demographic distribution graphs?			Very Insightful
1	2	3	4	5

Not Intuitive	How intuitive is the keyword selector?			Very Intuitive
1	2	3	4	5

Uninsightful	How insightful is the keyword distribution?			Very Insightful
1	2	3	4	5

Not Relevant	How relevant are the Recommended Reports to the entered Keywords?			Relevant
1	2	3	4	5

Not Intuitive	How intuitive is it to manipulate the Query?			Very Intuitive
1	2	3	4	5

Not Useful	How useful are the recommended reports?			Very Useful
1	2	3	4	5

Commenting On a Report

1. Go to reports in Case “Target”.
2. Select the report with id ‘132812391’.

- **Question:** How many comments were within the report when you first opened it?

  (**Ans:** Should only be **1**)

3. Delete previous comment.
4. Leave a comment.
5. Save your work.

Difficult	How easy was it to remove comments?			Easy
1	2	3	4	5

Difficult	How easy was it to add comments?			Easy
1	2	3	4	5

Not Useful	How useful do you think comments would be in this tool?			Very Useful
1	2	3	4	5

Enter Email for Amazon Gift Card!

Email:

____

Appendix B: User Evaluation (What a Participant Sees)

ConText User Evaluation

Survey

What is your age?

1. 18-20
2. 21-29
3. 30-39
4. 40-49
5. 50-59
6. 60 or older

What is you gender?

1. Female
2. Male
3. Other: __________
4. Prefer not to answer

What is the highest level of school you have completed or the highest degree you have received

1. Less than high school degree
2. High school degree or equivalent (e.g., GED)
3. Some college but no degree
4. Associate degree
5. Bachelor degree
6. Graduate degree

Evaluation Section 1: Starting From a New Account

Building a Case/Getting Started

1. Create a new account.

2. Go into reports section.

   - **Question:** How many reports are made available to the user when you first entered the reports page? (hint: there is an indicator on the page)

     ____

3. Create a new case titled "Patients."

4. Enter "Patients" into the search engine and execute search.

   - **Question:** How many search results appeared?

     ____

5. Take any 5 reports from the search results that you feel are relevant and add them to case "Patients."

   - **Question**: When choosing your reports, from which of the following groups did they belong to?

     1. Amongst the first 5 results

     2. Below the first 5 results

     3. Both from and below the first 5

   - **Question:** Why did you choose these particular reports out of the results?

____

____

____

6. Go to the case summary panel.

7. View the reports in the “Patients” case.

Difficult	How intuitive was it to navigate to the reports section?			Easy
1	2	3	4	5

Difficult	How easy was it to find reports related to the query?			Easy
1	2	3	4	5

Difficult	How easy was it to add reports to the case?			Easy
1	2	3	4	5

Difficult	How difficult was it to get to the reports within the case?			Easy
1	2	3	4	5

Marking Reports

1. Select 2 of the reports within case “Patient” and toggle them to primary.
2. Open the Case Summary Tab to the Left.
3. View the distribution of primary to supportive (using d3 graph).
    - **Question:** What is the current ratio between Primary and Supportive reports as indicated by the graph?

      ____

4. Navigate to 1st Report.
5. Annotate using the “Drug” highlight in the report.

Uninformative	How intuitive is the primary/supportive graph provided?			Very Informative
1	2	3	4	5

Unintuitive	How intuitive is it to annotate a report?			Intuitive
1	2	3	4	5

Summarizing a Case

1. Keywords should update in the Case Summary panel.
2. Remove a report from the case.
3. Go back to keywords; check that the keywords were removed.

    Question: Can you tell the information within the graphs has changed as a result of the report being removed?

    1. Yes
    2. No

Difficult	How easy was it to remove a report?			Easy
1	2	3	4	5

Section 2; Using an Existing Account

Dummy Test/Summarizing and Adding to a Larger Case

1. Log into a premade account, Username “EvalAcc”
2. Navigate to the premade case called “Target” in reports section.
3. View Case Summaries.
4. View distribution of medication error.

    Question: Which medication error is most prevalent?

    ____

5. Select Keywords to query for Recommended Reports.

6. Remove Keywords from query.

   Question: How many Recommended Reports are currently available after removing keywords?

   ____

7. Look for recommended reports.

8. Add the top recommended report to the premade case.

Unintuitive	How intuitive is it to interact with recommended reports?			Intuitive
1	2	3	4	5

Not Insightful	How insightful are the case summary demographic distribution graphs?			Very Insightful
1	2	3	4	5

Not Intuitive	How intuitive is the keyword selector?			Very Intuitive
1	2	3	4	5

Uninsightful	How insightful is the keyword distribution?			Very Insightful
1	2	3	4	5

Not Relevant	How relevant are the Recommended Reports to the entered Keywords?			Relevant
1	2	3	4	5

Not Intuitive	How intuitive is it to manipulate the Query?			Very Intuitive
1	2	3	4	5

Not Useful	How useful are the recommended reports?			Very Useful
1	2	3	4	5

Commenting On a Report

1. Go to reports in Case “Target”.
2. Select the report with id ‘132812391’.

- **Question:** How many comments were within the report when you first opened it?

  ____

3. Delete previous comment.
4. Leave a comment.
5. Save your work.

Difficult	How easy was it to remove comments?			Easy
1	2	3	4	5

Difficult	How easy was it to add comments?			Easy
1	2	3	4	5

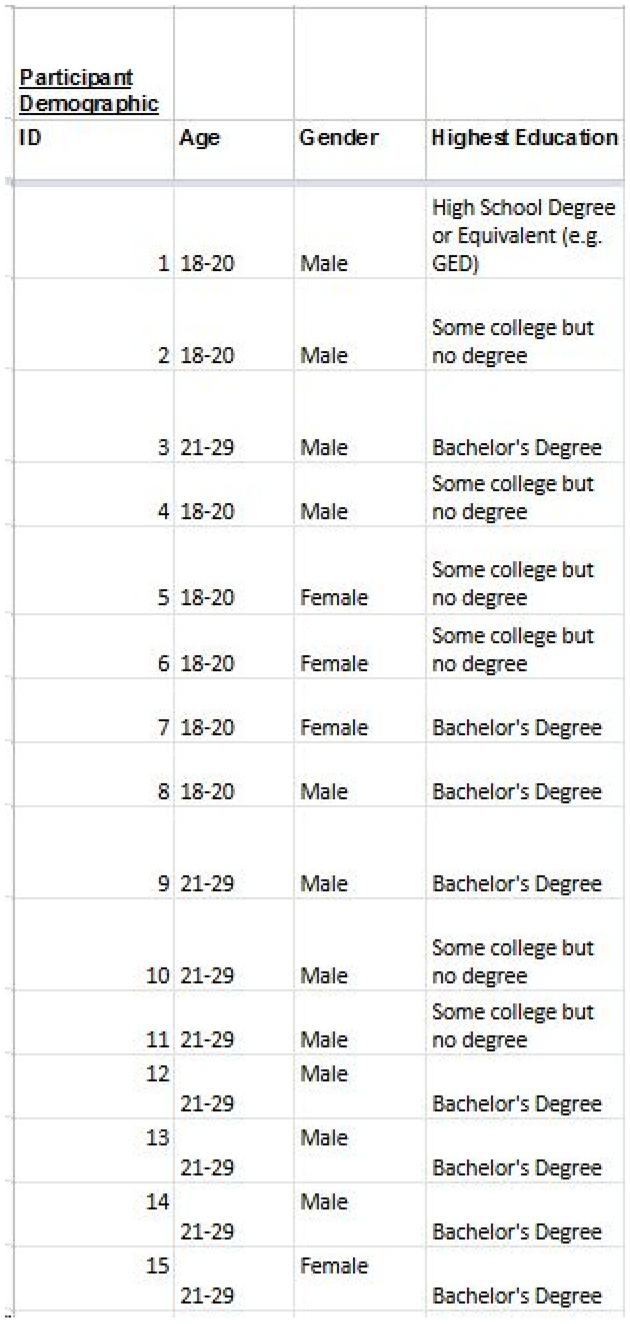
Not Useful	How useful do you think comments would be in this tool?			Very Useful
1	2	3	4	5

Enter Email for Amazon Gift Card!

Email:

____

Appendix C: Data Table



Participant Demographic			
ID	Age	Gender	Highest Education
1	18-20	Male	High School Degree or Equivalent (e.g. GED)
2	18-20	Male	Some college but no degree
3	21-29	Male	Bachelor's Degree
4	18-20	Male	Some college but no degree
5	18-20	Female	Some college but no degree
6	18-20	Female	Some college but no degree
7	18-20	Female	Bachelor's Degree
8	18-20	Male	Bachelor's Degree
9	21-29	Male	Bachelor's Degree
10	21-29	Male	Some college but no degree
11	21-29	Male	Some college but no degree
12	21-29	Male	Bachelor's Degree
13	21-29	Male	Bachelor's Degree
14	21-29	Male	Bachelor's Degree
15	21-29	Female	Bachelor's Degree

Task 1: Building a Case/Getting Started										
Time To Complete (MIN:SEC:MS)	Page Where Participant Ended	Question 1.1	Question 1.2	Question 1.3	Question 1.4	Qualitative 1.1	Qualitative 1.2	Qualitative 1.3	Qualitative 1.4	Comments
"05:14:63"	Stopped on Reports Page with Case Summary Closed	23889	150	Both from and below the first 5 results.	Randomly.	5	4	3	3	Did not see the number of Reports at the top of the page.
"05:32:17"	Stopped on Reports Page with Case Summary Open	23889	150	Below the first 5 results	First couple of results had different medication errors. I was going for "Accidental ex., Wrong Time, etc."	3	2	2	1	Saw the number of Reports at the top of the page. Not enough feedback for adding cases for finding Case Summary Panel
"06:17:21"	Stopped on Reports Page with Case Summary Open	23889	150	Both from and below the first 5 results.	Wrong Patients, Dosage, and Duration seemed valid when dealing with patients case.	3	4	4	5	Did not see the number of Reports at the top of the page. Needed more feedback on moving report into case.
"02:12:84"	Stopped on Reports Page with Case Summary Open	23889	150	Amongst the first 5 results.	It just seemed easiest with a random selection.	4	5	4	3	None
"04:08:32"	Ended back on the Dashboard	23889	NA	Below the first 5 results	Because the medication error for the ones I chose were "Wrong Patient" so it seemed to fit the name.	4	4	2	3	Did not see the number of Reports at the top of the page.
"04:31:76"	Stopped on Reports Page with Case Summary Open	All	3 Pages	Amongst the first 5 results.	They were easily accessible with a random selction.	3	4	5	5	Did not see the number of Reports at the top of the page.
"06:00:09"	Stopped on Reports Page with Case Summary Open	23889	150	Both from and below the first 5 results.	To have a variety of medication errors.	5	5	5	4	Did not see the number of Reports at the top of the page.
"07:30:53"	Stopped on Reports Page with Case Summary Open	23889	150	Both from and below the first 5 results.	Follow my eyes.	5	2	2	5	Needed to be guided to Case Summary. Needed to be given hint on Case move menu.
"08:14:28"	Stopped on Reports Page with Case Summary Open	23889	150	Both from and below the first 5 results.	First 5 to indicate wrong patient.	5	5	5	4	Needed to be given hint on Case move menu. Not enough feedback for adding cases for finding Case Summary Panel
"09:32:90"	Stopped on Reports Page with Case Summary Open	23889	150	Both from and below the first 5 results.	Combination of the medication error column in report table and understanding from skimming the report summary.	5	3	5	2	Not enough feedback for adding cases for finding Case Summary Panel
"06:09:53"	Stopped on Reports Page with Case Summary Open	23889	150	Both from and below the first 5 results.	Somewhat random process.	4	5	3	4	Not enough feedback for adding cases for finding Case Summary Panel
"05:21:00"	Reports Page	23889	150	Below the first 5 results	random	3	4	2	3	Case ellipsis is confusing
"04:14:00"	Reports Page	23889	150	Below the first 5 results	random	4	5	4	4	More responsive case interactions
"02:33:00"	Reports Page	23889	150	Below the first 5 results	random	5	5	5	4	More responsive case interactions
"05:11:00"	Reports Page	23889	150	Below the first 5 results	random	3	5	4	4	More responsive case interactions

Tasks 2: Marking Reports					
Time To Complete (MIN:SEC:MS)	Page Where Participant Ended	Question 2.1	Qualitative 2.1	Qualitative 2.2	Comments
"02:45:59"	Reports Page. Ended task without saving annotation.	3 to 2	4	5	Needed help with identifying D3 graph.
"01:16:99"	Reports Page. Ended task without saving annotation.	2 to 3	5	1	Keywords should be expanded by default.
"02:07:60"	Reports Page. Ended task without saving annotation.	2 to 3	5	4	None
"01:24:88	Reports Page. Ended task with saving annotation.	2 to 3	4	4	None
"2:35:82"	Reports Page. Ended task without saving annotation.	2 to 3	2	3	Needed a hint for opening the Case Summary Panel
"1:23:24"	Reports Page. Ended task without saving annotation.	2 to 3	4	5	None
"03:36:95"	Ended task without saving annotation	2 to 3	5	3	None
"02:30:51"	Ended task with saving annotation	0 to 4	2	3	Needed a hint for the highlighitng help button.
"02:33:79"	Ended task without saving annotation	3 to 2	4	5	None
"01:43:14"	Ended task without saving annotation	3 to 2	4	5	None
"02:18:49"	Ended task with saving annotation	2 to 3	5	4	None
"03:27:00"	reports page	2 to 3	3	2	Needed hint on how to edit
"02:21:00"	reports page	2 to 3	4	3	None
"01:31:00"	reports page	2 to 3	5	4	None
"02:11:00"	reports page	2 to 3	4	3	None

Task 3: Summarizing A Case

Time To Complete (MIN:SEC:MS)	Page Where Participant Ended	Question 3.1	Qualitative 3.1	Comments
"00:59:71"	Ended with Case Summary Open	Yes	5	None
"01:16:99"	Ended with Case Summary Open	Yes	5	None
"00:54:75"	Ended with Case Summary Open	Yes	5	None
"00:34:78	Ended with Case Summary Open	Yes	5	None
"00:53:79"	Ended with Case Summary Open	Yes	5	None
"00:39:97"	Ended with Case Summary Open	Yes	5	None
"1:01:06"	Ended with Case Summary Open	Yes	5	None
"01:21:57"	Ended with Case Summary Open	Yes	4	None
"01:15:39"	Ended with Case Summary Open	Yes	5	None
"00:50:43"	Ended with Case Summary Open	Yes	5	None
"00:56:65"	Ended with Case Summary Open	Yes	5	None
"01:05:00"	case summ open	Yes	5	None
"00:45:00"	case summ open	Yes	5	None
"00:32:00"	case summ open	Yes	5	None
"00:57:00"	case summ open	Yes	5	None

Task 4: Dummy Test/Summarizing and Adding to a Larger Case											
Time To Complete (MIN:SEC:MS)	**Page Where Participant Ended**	**Question 4.1**	**Question 4.2**	**Qualitative 4.1**	**Qualitative 4.2**	**Qualitative 4.3**	**Qualitative 4.4**	**Qualitative 4.5**	**Qualitative 4.6**	**Qualitative 4.7**	**Comments**
"5:56:47"	Ended with User on Case Target	Wrong Dosage Form	150	3	4	3	4	3	4	4	Needed help Navigating back to Reports
"03:45:39"	Ended with User on with all reports open	Wrong Dosage Form	22	2	3	2	5	4	2	4	None
"04:03:02"	Ended with User on with all reports open	Wrong Dosage Form	150	4	4	4	4	4	4	4	None
"02:52:33"	Ended with User on Case Target	Wrong Dosage Form	0	4	5	4	5	4	3	4	None
"03:13:34"	Ended with User on Case Target	Wrong Dosage Form	128	3	3	2	4	4	2	4	Needed Help Navigating to Case Summary Panel
"02:51:14"	Ended with User on with all reports open	Wrong Dosage	3 Pages	4	5	5	4	4	5	4	None
"03:32:20"	Ended on the Searched Reports tab.	Wrong Dosage	150	4	5	5	5	5	4	5	Needed Help with using Get Recommendations
"04:21:57"	Ended with User on Case Target.	Wrong Dosage Form	150	4	2	3	3	1	4	3	Needed Help Navigating to Case Summary Panel
"05:10:80"	Ended on the Searched Reports tab.	Wrong Dosage	150	5	5	5	5	5	5	4	None
"02:49:02"	Ended on the Searched Reports tab.	Wrong Dosage	150	5	1	4	4	4	3	4	None
"02:43:04"	Ended with User on Case Target.	Wrong Dosage Form	150	4	4	4	4	5	5	5	Very bad mem issue.
"04:38:00"	in case tab	Wrong Dosage	150	3	3	3	4	3	2	4	needed help getting recommendation
"03:55:00"	in case tab	Wrong Dosage	150	4	4	4	4	3	3	4	None
"03:01:00"	in case tab	Wrong Dosage	128	4	5	5	5	4	4	4	None
"03:27:00"	in case tab	Wrong Dosage	150	3	4	4	5	3	4	4	None

Task 5: Commenting On a Report						
Time To Complete (MIN:SEC:MS)	Page Where Participant Ended	Question 5.1	Qualitative 5.1	Qualitative 5.2	Qualitative 5.3	Comments
"01:42:44"	Ended on Correct Report	1	5	5	5	Needed some help finding report.
"00:40:47"	Ended on Incorrect Report	1	4	4	5	Two Buttons Post and Save unnecessary.
"01:21:46"	Ended on Correct Report	1	5	5	5	Needed some help finding report.
"00:54:52"	Ended on Correct Report	1	5	5	4	None
"1:03:03"	Ended on Correct Report	1	5	3	5	None
"02:10:03"	Ended on Correct Report	1	5	5	5	None
"01:07:77"	Ended on Correct Report	1	5	5	5	None
"01:30:50"	Ended on Correct Report	1	5	5	4	None
"01:20:07"	Ended on Correct Report	1	5	5	5	None
"01:43:14"	Ended on Correct Report	1	5	4	5	Two Buttons Post and Save unnecessary.
"00:59:05"	Ended on Correct Report	1	5	5	5	None
"01:52:00"	correct report	1	5	2	5	post and save confusing
"01:11:00"	correct report	1	5	3	5	post and save confusing
"01:02:00"	correct report	1	5	4	5	post and save confusing
"01:22:00"	correct report	1	5	3	5	post and save confusing

Appendix D: Email to Student Body Requesting Pa

Subject: Request for Participants in User Study

Hello Everyone,

We are an MQP group working with the US Food and Drug Administration, and we are looking for participants for a User Evaluation aimed at testing the usability of our Web Application, ConText. ConText is aimed at assisting Food and Drug Administration Agents in finding correlations between drug-related medical incidents. Participants will be given the option to enter a raffle, in which three Students will win a $50 Amazon gift card.

This test should not take any longer than 20 minutes.

If you would like to participate, please sign up for a time on the following signup sheet:

[LINK REDACTED]

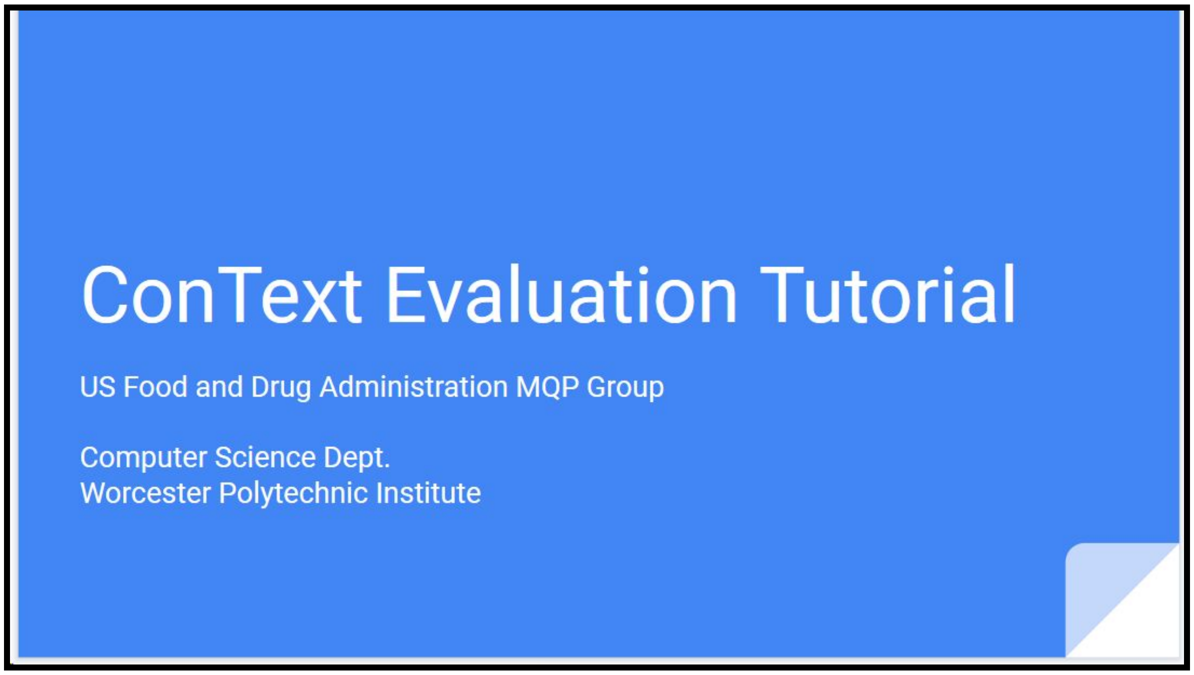
We will be setting up a testing location in the Wedge (between Daniels and Morgan Halls). Please Note, we will only be accepting 20 Evaluation Participants, so sign up quick!

Thank you,

Jacob Kaplan
Peter Nolan
Logan Romero

Appendix E: Evaluation Tutorial (Shown to each Participant)

Slide 1:

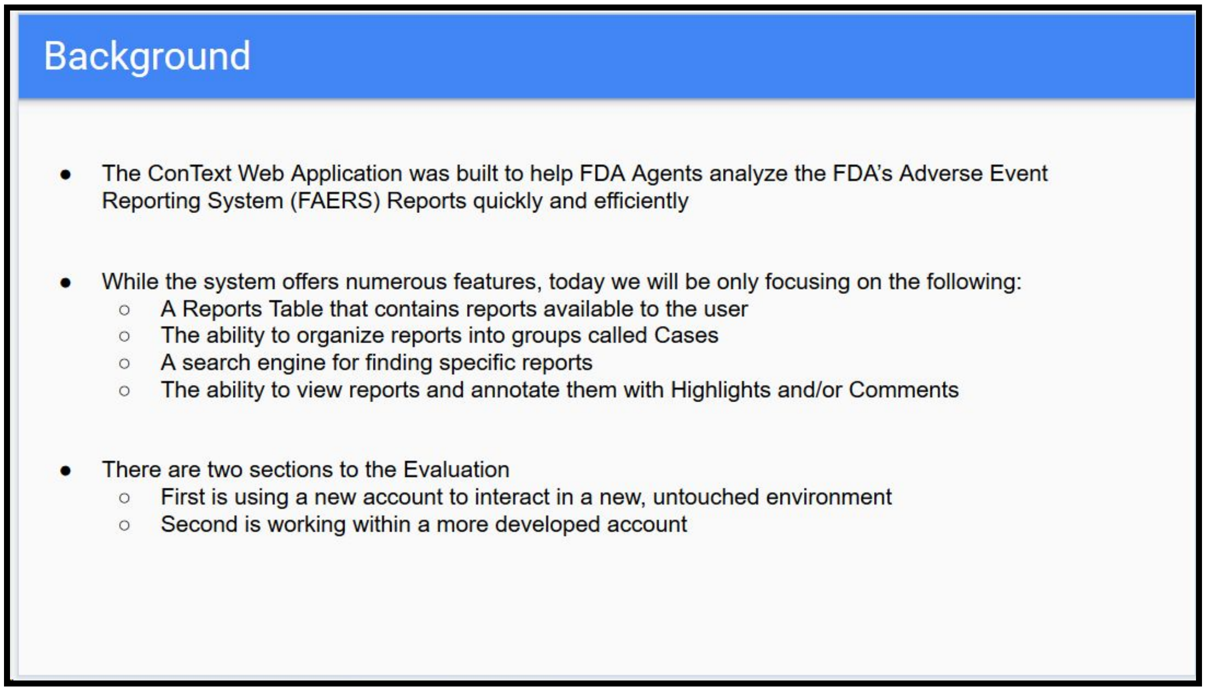


ConText Evaluation Tutorial

US Food and Drug Administration MQP Group

Computer Science Dept.
Worcester Polytechnic Institute

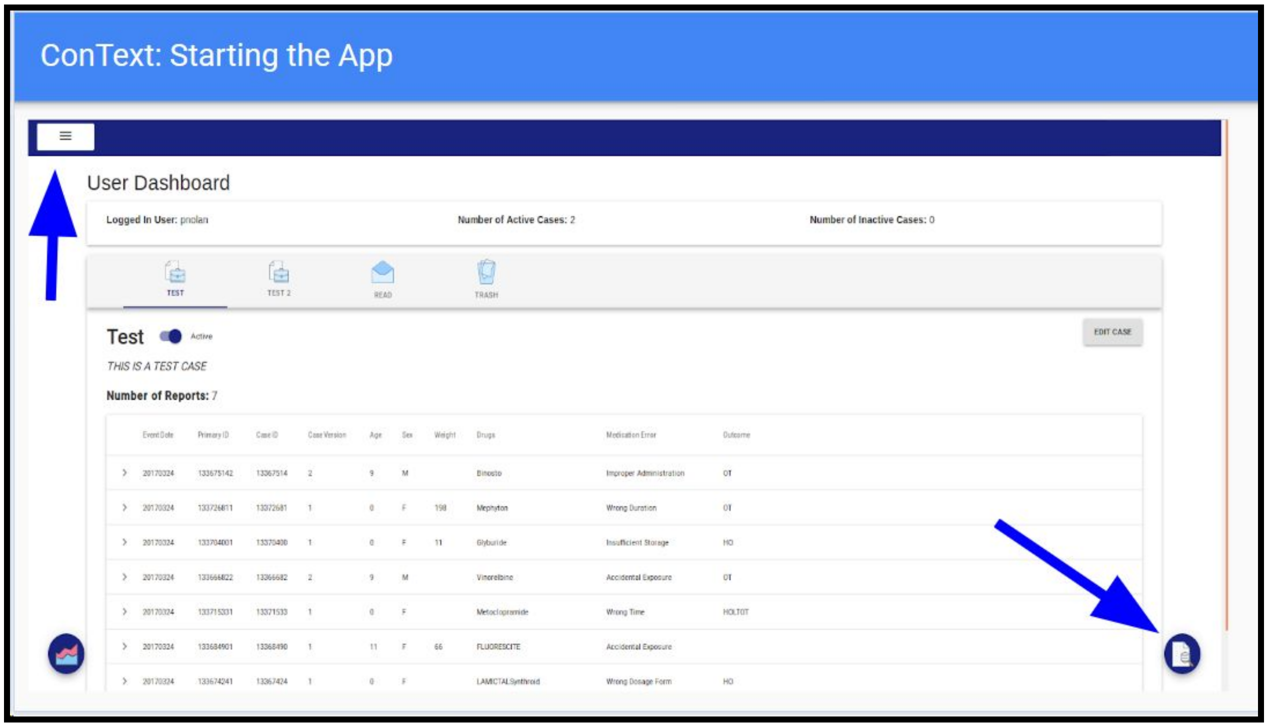
Slide 2:



Background

- The ConText Web Application was built to help FDA Agents analyze the FDA's Adverse Event Reporting System (FAERS) Reports quickly and efficiently
- While the system offers numerous features, today we will be only focusing on the following:
  - A Reports Table that contains reports available to the user
  - The ability to organize reports into groups called Cases
  - A search engine for finding specific reports
  - The ability to view reports and annotate them with Highlights and/or Comments
- There are two sections to the Evaluation
  - First is using a new account to interact in a new, untouched environment
  - Second is working within a more developed account

Slide 3:



ConText: Starting the App

User Dashboard

Logged In User: pnolan **Number of Active Cases:** 2 **Number of Inactive Cases:** 0

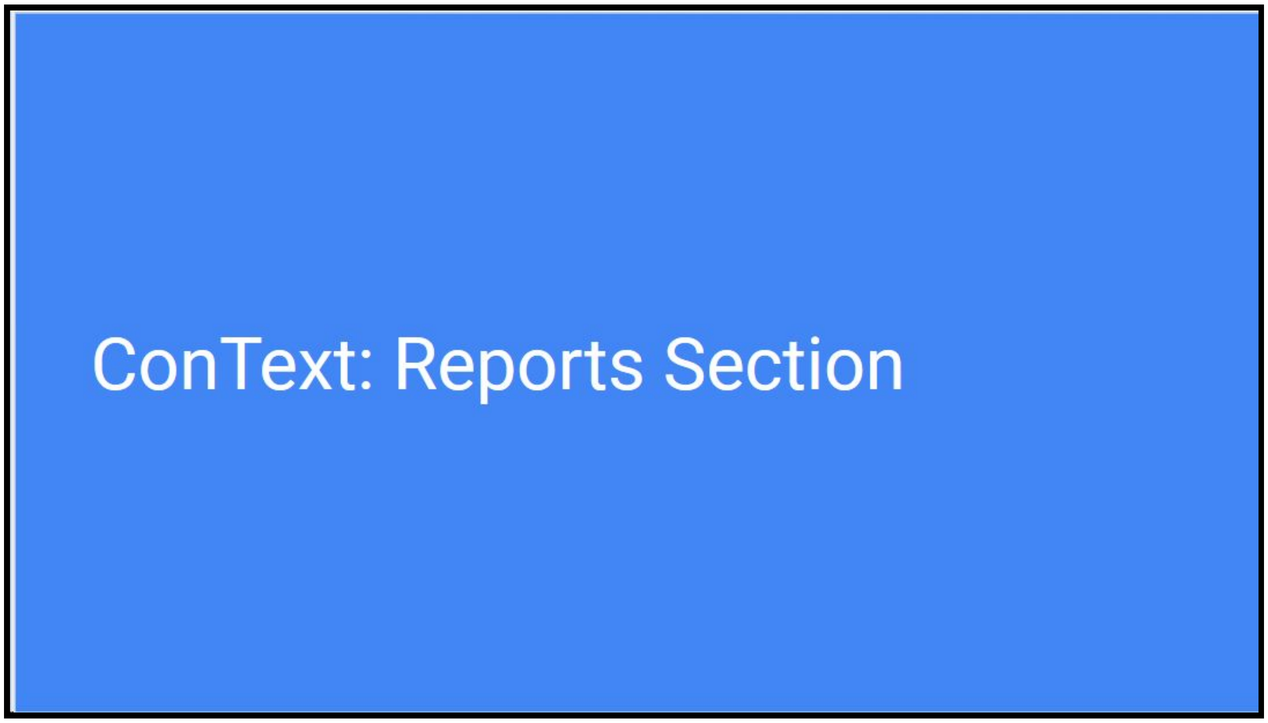
Test Active

THIS IS A TEST CASE

Number of Reports: 7

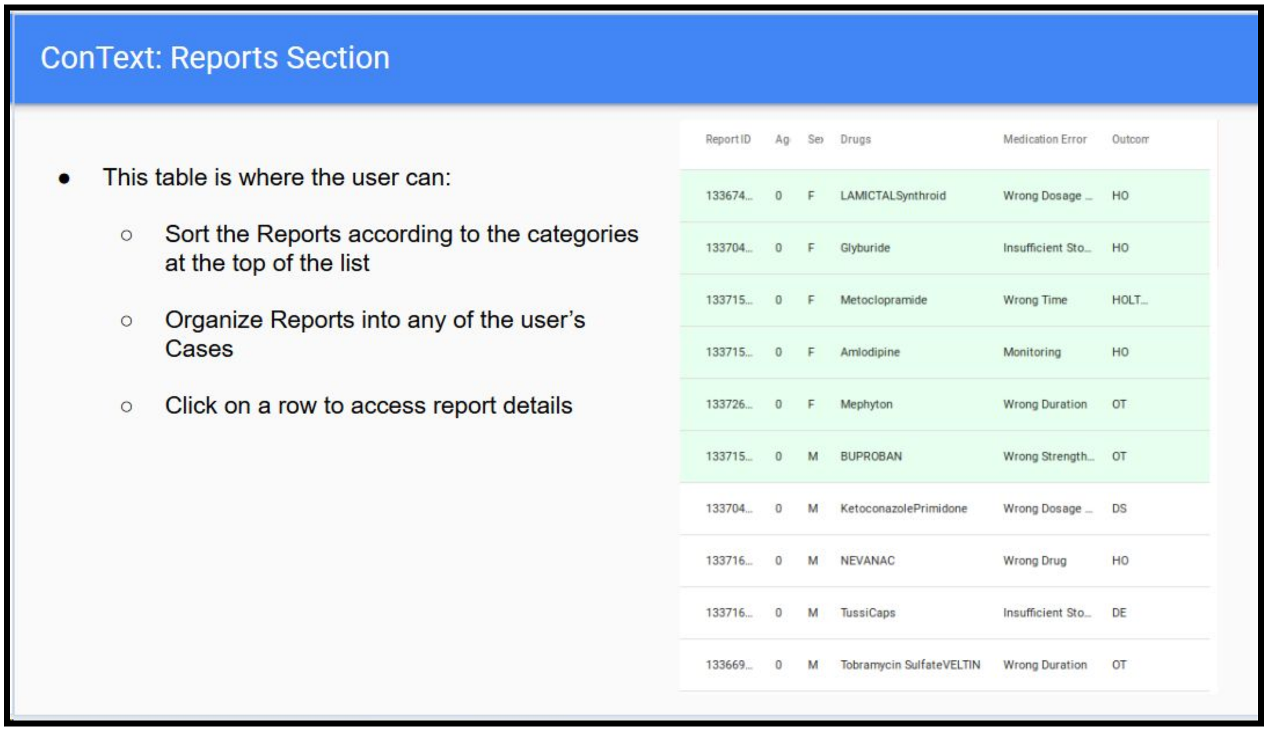
	Event Date	Primary ID	Case ID	Case Version	Age	Sex	Weight	Drugs	Medication Error	Outcome
>	20170324	133675142	13367514	2	9	M		Binosto	Improper Administration	OT
>	20170324	133726811	13372681	1	0	F	198	Mephyton	Wrong Duration	OT
>	20170324	133704001	13370400	1	0	F	11	Glyburide	Insufficient Storage	HO
>	20170324	133666822	13366682	2	9	M		Vinorelbine	Accidental Exposure	OT
>	20170324	133715331	13371533	1	0	F		Metoclopramide	Wrong Time	HOLTOT
>	20170324	133684901	13368490	1	11	F	66	FLUORESCITE	Accidental Exposure	
>	20170324	133674241	13367424	1	0	F		LAMICTALSynthroid	Wrong Dosage Form	HO

Slide 4:



ConText: Reports Section

Slide 5:

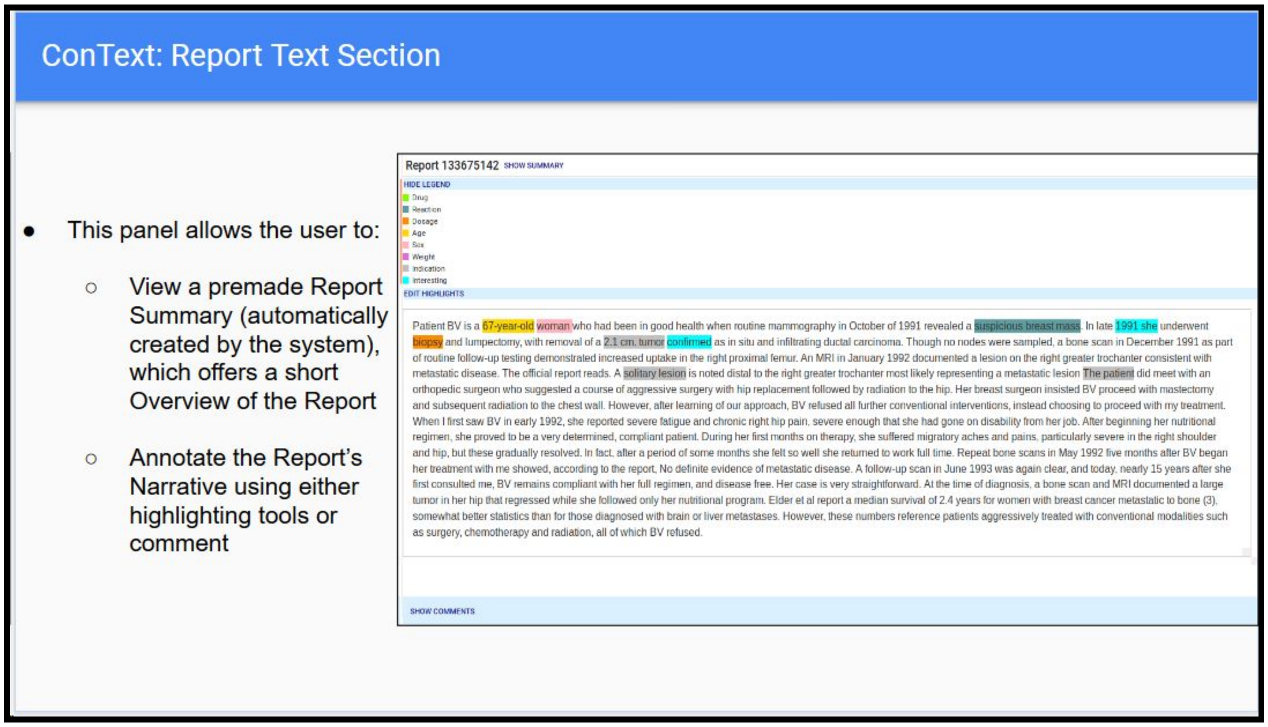


ConText: Reports Section

- This table is where the user can:
  - Sort the Reports according to the categories at the top of the list
  - Organize Reports into any of the user's Cases
  - Click on a row to access report details

Report ID	Ag	Se	Drugs	Medication Error	Outcom
133674...	0	F	LAMICTALSynthroid	Wrong Dosage ...	HO
133704...	0	F	Glyburide	Insufficient Sto...	HO
133715...	0	F	Metoclopramide	Wrong Time	HOLT...
133715...	0	F	Amlodipine	Monitoring	HO
133726...	0	F	Mephyton	Wrong Duration	OT
133715...	0	M	BUPROBAN	Wrong Strength...	OT
133704...	0	M	KetoconazolePrimidone	Wrong Dosage ...	DS
133716...	0	M	NEVANAC	Wrong Drug	HO
133716...	0	M	TussiCaps	Insufficient Sto...	DE
133669...	0	M	Tobramycin SulfateVELTIN	Wrong Duration	OT

Slide 6:



ConText: Report Text Section

- This panel allows the user to:
  - View a premade Report Summary (automatically created by the system), which offers a short Overview of the Report
  - Annotate the Report's Narrative using either highlighting tools or comment

Report 133675142 SHOW SUMMARY

HIDE LEGEND

- 🟩 Drug
- 🟦 Reaction
- 🟧 Dosage
- 🟨 Age
- 🟥 Sex
- 🟪 Weight
- ⬜ Indication
- 🟦 Interesting

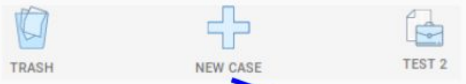
EDIT HIGHLIGHTS

Patient BV is a 67-year-old woman who had been in good health when routine mammography in October of 1991 revealed a suspicious breast mass. In late 1991 she underwent biopsy and lumpectomy, with removal of a 2.1 cm. tumor confirmed as in situ and infiltrating ductal carcinoma. Though no nodes were sampled, a bone scan in December 1991 as part of routine follow-up testing demonstrated increased uptake in the right proximal femur. An MRI in January 1992 documented a lesion on the right greater trochanter consistent with metastatic disease. The official report reads. A solitary lesion is noted distal to the right greater trochanter most likely representing a metastatic lesion The patient did meet with an orthopedic surgeon who suggested a course of aggressive surgery with hip replacement followed by radiation to the hip. Her breast surgeon insisted BV proceed with mastectomy and subsequent radiation to the chest wall. However, after learning of our approach, BV refused all further conventional interventions, instead choosing to proceed with my treatment. When I first saw BV in early 1992, she reported severe fatigue and chronic right hip pain, severe enough that she had gone on disability from her job. After beginning her nutritional regimen, she proved to be a very determined, compliant patient. During her first months on therapy, she suffered migratory aches and pains, particularly severe in the right shoulder and hip, but these gradually resolved. In fact, after a period of some months she felt so well she returned to work full time. Repeat bone scans in May 1992 five months after BV began her treatment with me showed, according to the report, No definite evidence of metastatic disease. A follow-up scan in June 1993 was again clear, and today, nearly 15 years after she first consulted me, BV remains compliant with her full regimen, and disease free. Her case is very straightforward. At the time of diagnosis, a bone scan and MRI documented a large tumor in her hip that regressed while she followed only her nutritional program. Elder et al report a median survival of 2.4 years for women with breast cancer metastatic to bone (3), somewhat better statistics than for those diagnosed with brain or liver metastases. However, these numbers reference patients aggressively treated with conventional modalities such as surgery, chemotherapy and radiation, all of which BV refused.

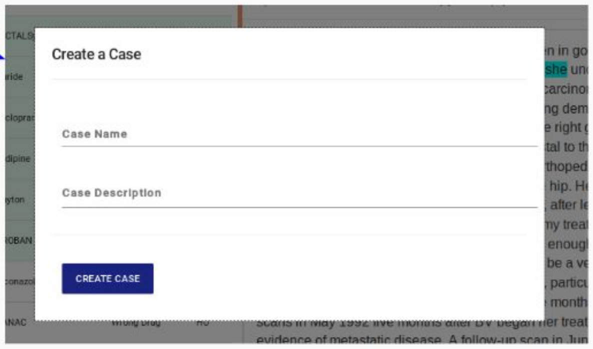
SHOW COMMENTS

Slide 7:

ConText: Add Case



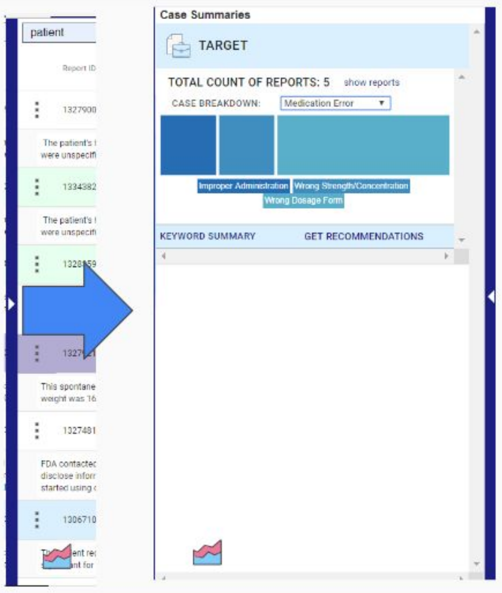
- To create a new case:
  1. Navigate to the toolbar above the reports table
  2. Click on the “NEW CASE” button
  3. Enter information into the bubble menu that appears



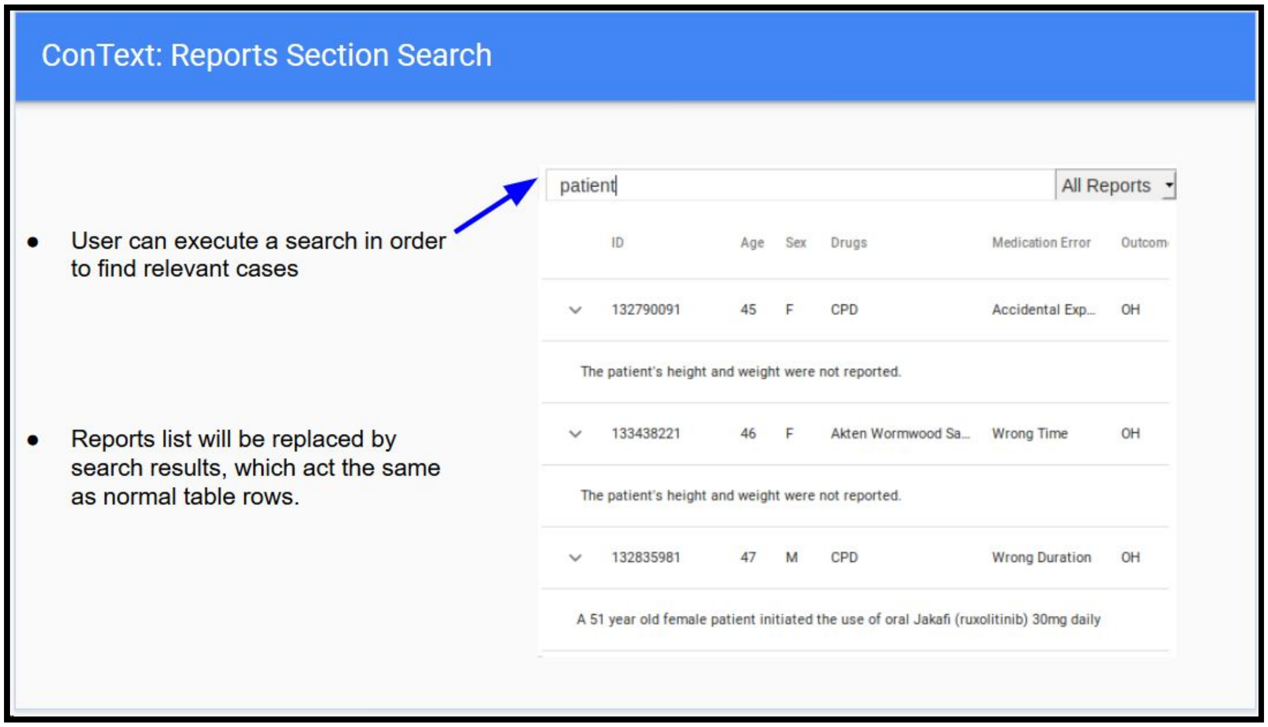
Slide 8:

ConText: Case Summary

- To access Case Summary Panel:
  1. Navigate to the bottom left corner of the page
  2. Click the Case Summary button
  3. The Case Summary panel will expand from the left side



Slide 9:



ConText: Reports Section Search

- User can execute a search in order to find relevant cases
- Reports list will be replaced by search results, which act the same as normal table rows.

patient | All Reports

	ID	Age	Sex	Drugs	Medication Error	Outcom
⌄	132790091	45	F	CPD	Accidental Exp...	OH

The patient's height and weight were not reported.

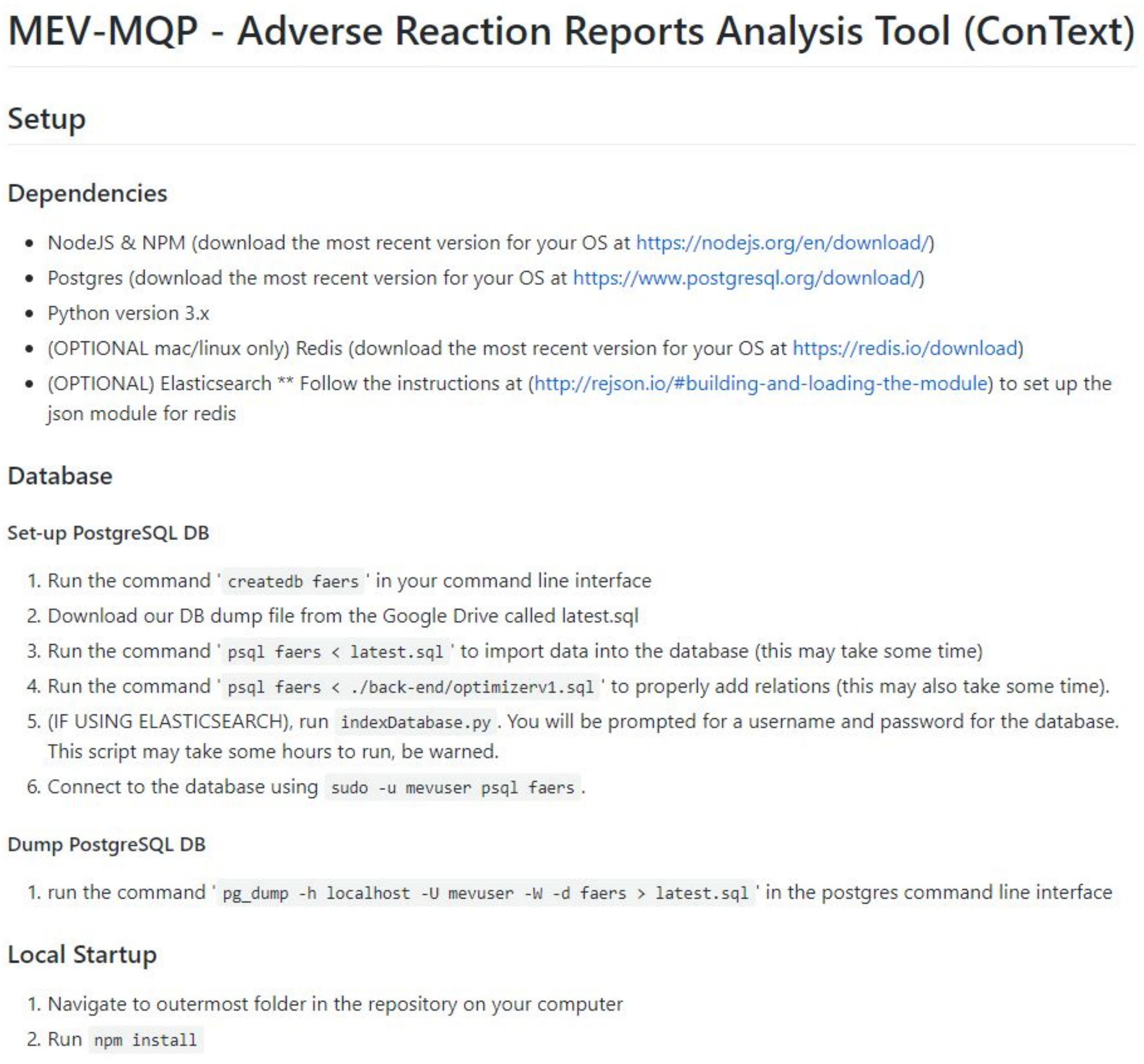
	ID	Age	Sex	Drugs	Medication Error	Outcom
⌄	133438221	46	F	Akten Wormwood Sa...	Wrong Time	OH

The patient's height and weight were not reported.

	ID	Age	Sex	Drugs	Medication Error	Outcom
⌄	132835981	47	M	CPD	Wrong Duration	OH

A 51 year old female patient initiated the use of oral Jakafi (ruxolitinib) 30mg daily

Appendix F: Project Readme



MEV-MQP - Adverse Reaction Reports Analysis Tool (ConText)

Setup

Dependencies

- NodeJS & NPM (download the most recent version for your OS at https://nodejs.org/en/download/)
- Postgres (download the most recent version for your OS at https://www.postgresql.org/download/)
- Python version 3.x
- (OPTIONAL mac/linux only) Redis (download the most recent version for your OS at https://redis.io/download)
- (OPTIONAL) Elasticsearch ** Follow the instructions at (http://rejson.io/#building-and-loading-the-module) to set up the json module for redis

Database

Set-up PostgreSQL DB

1. Run the command ' `createdb faers` ' in your command line interface
2. Download our DB dump file from the Google Drive called latest.sql
3. Run the command ' `psql faers < latest.sql` ' to import data into the database (this may take some time)
4. Run the command ' `psql faers < ./back-end/optimizerv1.sql` ' to properly add relations (this may also take some time).
5. (IF USING ELASTICSEARCH), run `indexDatabase.py` . You will be prompted for a username and password for the database. This script may take some hours to run, be warned.
6. Connect to the database using `sudo -u mevuser psql faers` .

Dump PostgreSQL DB

1. run the command ' `pg_dump -h localhost -U mevuser -W -d faers > latest.sql` ' in the postgres command line interface

Local Startup

1. Navigate to outermost folder in the repository on your computer
2. Run `npm install`

3. Navigate to front-end folder
4. Run `npm install`
5. Navigate to back-end folder
6. Run `npm install`
7. Navigate to the project root
8. Run `npm start` to start the application

Remote Startup and Deploy

1. Open `context.wpi.edu:8080` in your web browser. You will be required to log in.
2. Navigate to the `Context` project
3. Make sure no active build is running. If it is, click on the 'X' on the build to kill it.
4. On the left panel, click "build now" to deploy the website. It will automatically pull your new code from GIT
5. NOTE: if using a different repository, you may have to change the Configuration. This is pretty straightforward.

Starting PostgreSQL

- If you need to restart the database for any reason, type the command `sudo systemctl restart postgresql.service`

Project Layout

|-back-end
|-front-end
|___public
|___src
| |___resources | | |___images | |___components
| | |___visualization
| | | |___components
| | | | |___demographics
| | | | | |___components
| | | | | | |___components
| | | | |___timeline
| | | | | |___components
| | | | | | |___components
| | | | |___treeMap
| | | | | |___components
| | |___portal

| | | | | | |___components
| | | | |___treeMap
| | | | | |___components
| | |___portal
| | | |___userComponents
| | |___cases
| | |___components
| | |___editor
| | | |___components
| | |___reports
| | | |___components
| |___actions
| |___reducers

Working on the Front End

To make changes to the UI and other aspects of the front end, you must edit assets found in the front-end folder.

Working with React + Redux

When editing the global state that Redux manages, you need to make changes in several different locations.

Useful Readings/Videos for Learning React and Redux

- Thinking In React
- State and Lifecycle
- React Component Lifecycle
- JSX in Depth
- Helpful Youtube Channel
- React and Redux Rapid Course Video Series

Outgoing Web Fetch Calls

All asynchronous http requests are done from the `/actions` folder. This is where we talk to the back-end server to get data from our database. These async calls are created using the `fetch()` function which returns a `Promise`. Please read more about javascript promises here.

Adding to the Redux (Global) State

When adding to the global state you must create a function inside of a file in the `/actions` folder, add a case inside of a file in the `/reducers` folder and add an import to the file you are adding to the Redux state from.

In the `timelineActions.js` file we have:

```
export const setTimelineMinimizedToggle = toggle =>
  dispatch => dispatch({ type: 'TOGGLE_TIMELINE_MINIMIZED', timelineMinimized: toggle });
```

In this example we are adding a value `toggle`, labeled as `timelineMinimized` with a type `TOGGLE_TIMELINE_MINIMIZED` this is just a string to know what to listen for in the Reducer. This `dispatch()` function is what we need to call in order to have this information be sent to the reducers and be added to the global state.

In the `timelineReducer.js` file we have:

```
export default (state = initialTimelineState, action = {}) => {
  switch (action.type) {
    case 'SET_ENTIRE_TIMELINE':
      return Object.assign({}, state, { entireTimelineData: action.entireTimelineData });
    case 'TOGGLE_TIMELINE_MINIMIZED':
      return Object.assign({}, state, { timelineMinimized: action.timelineMinimized });
    default: return state;
  }
};
```

We are listening for the same `TOGGLE_TIMELINE_MINIMIZED` type, when we find that type we are setting the state to have the `timelineMinimized` value passed from the actions.

In the `Timeline.jsx` file we use this action like such:

We import the functions from the actions.

```
import { setTimelineMinimizedToggle, getEntireTimeline, setSelectedDate } from '../../../../actions/timelineActions';
```

We then connect these actions to the current component as `props`.

```
export default connect(
  mapStateToProps, // Used for reading the Global state
  { setTimelineMinimizedToggle, getEntireTimeline, setSelectedDate },
)(withStyles(styles)(Timeline));
```

We add these functions to the proptypes of our component.

```
static propTypes = {
  getEntireTimeline: PropTypes.func.isRequired,
  setSelectedDate: PropTypes.func.isRequired,
  setTimelineMinimizedToggle: PropTypes.func.isRequired,
}
```

We can call these functions from our props like this.

```
this.props.setTimelineMinimizedToggle(true);
```

Getting from the Redux (Global) State

To get from the global state, we need to check to make sure the information is in the state properly and then import and connect it into our component.

First make sure we have the proper `case` clause for the information we are looking for. From the example above, we are looking for `TOGGLE_TIMELINE_MINIMIZED` inside of the `timelineReducer.js`.

```
export default (state = initialTimelineState, action = {}) => {
  switch (action.type) {
    case 'SET_ENTIRE_TIMELINE':
      return Object.assign({}, state, { entireTimelineData: action.entireTimelineData });
    case 'TOGGLE_TIMELINE_MINIMIZED': // Here it is
      return Object.assign({}, state, { timelineMinimized: action.timelineMinimized });
    default: return state;
  }
};
```

We also need to make sure we have imported the `timelineReducer` into the index reducer inside of the `/reducers/index.js` file.

```
import demographic from './demographicReducer';
import filters from './filterReducer';
import multiSelectFilters from './multiSelectFilterReducer';
import user from './userReducer';
import mainVisualization from './visualizationReducer';
import timeline from './timelineReducer';

/**
 * Redux Reducer that combines all of the other reducers to build the Redux State
 */
export default {
  demographic,
  filters,
  multiSelectFilters,
  mainVisualization,
  timeline, // Exported as 'timeline'
  user,
};
```

We can see we import the `timelineReducer` and export it with the name `timeline`. **This name is important.**

Now we can go to our component `Timeline.jsx` and import the global state.

```
const mapStateToProps = state => ({
  entireTimelineData: state.timeline.entireTimelineData,
  demographicSexData: state.demographic.sex,
});

export default connect(
  mapStateToProps,
  { setTimelineMinimizedToggle, getEntireTimeline, setSelectedDate },
)(withStyles(styles)(Timeline));
```

Here in the `mapStateToProps` function we are getting from the global state with:

```
entireTimelineData: state.timeline.entireTimelineData,
```

It is `state.timeline` since that is what we exported as inside of the `reducers/index.js` file. And in this case we are getting the `entireTimelineData` from above:

```
case 'SET_ENTIRE_TIMELINE':
  return Object.assign({}, state, { entireTimelineData: action.entireTimelineData });
```

Inside of our component we need to add this to the proptypes.

```
static propTypes = {
  entireTimelineData: PropTypes.arrayOf(
    PropTypes.shape({
      init_fda_dt: PropTypes.string.isRequired,
      serious: PropTypes.number.isRequired,
      not_serious: PropTypes.number.isRequired,
    }),
  ).isRequired,
}
```

This data can now be accessed from the props as such:

```
this.props.entireTimelineData
```

Working on the Visualization page

Assets for the visualization page exist within the visualization folder src/components/visualization. visualization/App.js uses the components for the treemaps, demographics, and timeline which assets are each in their respective folder inside visualization/components.

Working on the Reports Listing page

Assets for the reports listing page exist within the reports folder src/components/reports. reports/ReportsList.jsx uses the components for the report listing grid and Reports Panel which assets are all contained in the folder reports/components.

Working on the Narrative Editor page

All assets for the narrative editor page exist within the editor folder src/components/editor

Working on the Login page, About page, or Dashboard page

All assets for these pages exist within the portal folder src/components/portal
Assets for the dashboard page exist inside the portal/userComponents folder.

Working on the Back End

Our backend is a one file server that receives requests sent from the front end located at back-end/App.js

Adding and modifying endpoints

We use the express library (https://expressjs.com/) for our backend and documentation for use can be found at https://expressjs.com/en/4x/api.html#app.

DB Schema

Link to the FAERS database
PostgreSQL Schema