Drug Label Explorer

Spring 2022

Software Engineering Capstone, CSCI E-599 Section 2

Group Members

Ken Brown, Agi Kajanaku, Leo Landau, Sam Negassi, Ky Nguyen

Customer

David Edelen

Teaching Staff

Peter Henstock, Roman Burdakov

Table of Contents

1. Introduction

1.1 Background

A Drug Label contains important information about a medication such as dosage and administration, clinical pharmacology, boxed warning, indication, and pharmacokinetics. Before a drug goes into the market, a drug label document is prepared and submitted by the drug Marketing Authorization Holder to a regulatory agency, who are responsible for reviews and approvals of the text in drug labels.

The names and contents of drug label documents differ by type of drug and country or jurisdiction. In the US, drugs are categorized under the following: prescription drugs, biological products, and over-the-counter drugs. In this project, we have limited our scope only to human prescription drugs that are approved for marketing in the United States and European Union.

In the United States, drug labels for human prescription drugs are called Prescribing Information (PI). The Food and Drug Administration (FDA) approves the PI per the regulatory requirements established by a Code of Federal Regulations (CFR) [1]. Similarly, in the European Union, drug labels for human prescription drugs are known as a Summary of Product Characteristics (SmPC). The European Medicines Agency (EMA) approves the SmPC per the regulatory requirements issued through A Guideline on Summary of Product Characteristics [2].

Both PI and SmPC documents contain a summary of the essential scientific information for the safe and effective use of a drug. The information contained in them is formatted differently but generally includes information such as indications, dosage and administration, contraindications, warnings and precautions, adverse reactions, drug interactions, information about use in specific populations, storage and handling and other pertinent information.

The FDA requires drug labels to be “informative and accurate and neither promotional in tone nor false or misleading”. There are over 51,000 human prescription drugs and biological products approved for use in the United States [25], and that number is steadily growing by a few hundred newly approved additions every year.

Pharmaceutical companies, researchers, and regulators are increasingly looking for ways to leverage the wealth of existing drug label data to ensure the safety of existing products, reduce the cost and time to discover new drugs, and to improve the quality and standardize the data in the various drug label documents used by healthcare professionals and the public. One recent trend to leverage the available data is the use of machine learning.

Classification of existing data can help predict the potential adverse reactions of a certain drug. For example, a study developed a scheme to assess a drug's potential for drug induced liver damage by systematically classifying approved drug label data [27].

Drug development is a time-consuming and expensive endeavor that takes 10 years and an estimated $2.6 billion on average per drug [26]. One way to shorten the time and reduce the cost of drug development is the use of repurposing, or formally known as repositioning, of existing drugs for new use. Repositioning involves exploring new indications for existing, discontinued, or investigational drugs by taking advantage of available safety, pharmacokinetic and manufacturing information.

The ability to do semantic search through the corpus of approved drug labels as well as the ability to compare two drug labels will help drug makers draft better drug labeling for future drugs.

However, computational repositioning and safety assessment schemes will result in low quality predictions if the drug label data used is of low quality. Therefore, having a drug label data repository with an easy to use user interface will help the existing corpus of approved drug label data more useful.

1.2 Literature Review

1.2.1 Drug Label Data Sources

Labels for all the approved drugs in the US are made publicly available by the FDA through Drugs@FDA databases [16], and by the National Institute of Health (NIH) through the DailyMed website [17] in XML and PDF file formats. Similarly, labels for drugs approved in the EU are available from the EMA website [18] in a PDF file format.

While many sections of the PI and SmPC have similar section headings for similar content, some sections use different naming for the same content between the two documents. In one

study, the SmPC from EMA and the PI from FDA were examined for 32 biopharmaceutical products approved in both regions [9]. This study compared the labels across several factors including therapeutic indications, contraindications, warnings & precautions, adverse reactions, pregnancy and lactation, pharmacodynamics, and efficacy, as well as pediatric use. The study found that the EU-approved SmPC was more conservative compared to the FDA-approved PI with regards to factors such as contraindications and warnings [9]. In addition, the SmPC had more detailed instructions about drug efficacy for different stages of the disease as well as interaction with other therapies.

Previous work at categorizing drug label resources has had success utilizing sources such as Drugs@FDA, RxNorm, and the SRS-UNII [4]. These data sources or data repositories make a lot of information accessible that would otherwise be hard to find or compile. For example, Structured Product Labeling (SPL) from the FDA is a good resource, but often too vast to be utilized easily. A 2016 report showed that of only 1600 human prescription drugs there were approximately 31,000 SPLs associated with them [4]. This is because one prescription drug can have many drug products because of the differences in regulatory application, dosage forms, routes of administration, and manufacturers.These large numbers of labels need search features before the information is useful or actionable.

1.2.2 Drug Label Update Frequency and Impact

Once a drug is approved and is out on the market, government regulations require the manufacturers to submit updated drug labels whenever they learn new information that can cause the existing labels to become inaccurate, misleading, or false. As a result, a given drug label evolves over time and the regulatory agencies review, approve and make the updated labels available to the public. A study found approximately 450 drug label updates are published by the FDA every week [4]. These updates can originate from a variety of sources such as spontaneous reports (52%), clinical trials (16%), and pharmacokinetic studies (11%) [3]. The FDA reported that in 2009 there were 181 major safety regulatory actions, including 25 new boxed warnings and 19 contraindications [5]. Minor changes are even more numerous and widespread. Moreover, these changes are not limited to new drugs. They are widely distributed among both drugs recently approved and drugs with longer tenure. In a sample of 181 updated drugs, less than 75% were recently approved [5]. Of this sample size, a safety action occurred in 61 drugs after they were in the market for at least 15 years, with the median time for action after initial approval being 11 years. Actions within the first 5 years after approval occurred in only 36 drugs.

Some updates to drug labels are also associated with 2 unfortunate consequences: non-proportional risk mitigation and delayed drug approvals due to regulatory agency's resource limitations. While updates highlight adverse effects and additional warnings, recommending additional monitoring usually led to decreased drug use overall [6]. This is likely due to poor risk communication as opposed to actual risk. And while drug labels change frequently, the official process for approving changes can take a long time, which can result in delaying safety/corrective actions for drugs already on the market. These two consequences can be very dangerous for patients as it often results in labels not being fully incorporated into related documentation [7].

1.2.3 Currently Available Solutions

The main sources of the approved prescription drug labels are the FDALabel [16] and the DailyMed [17] databases for the U.S. and the EMA [18] database for the E.U. approved drugs. In addition to making the raw data of all approved drug labels available, these web-based sites provide basic search functionality. The FDA provides a drug Safety-related Labeling Changes (SrLC) database [19] which does allow a user to search for drug label changes; however this is missing the important feature of being able to search for specific text in the drug label document. However, the data is only accessible as XML or PDF files with no feature to search in specific sections and no features to compare between drug labels. Moreover, the user interfaces are not intuitive.

Other available web-based resources we surveyed include RxList [20], ReedTech [21], WizMed [22], Cerner Website [23], and Drugs.com [24]. These commercially available websites provide searchable drug label databases with varying levels of content, search features, and accessibility. Some of these commercially available databases are subscription fee-supported [21, 22], some are ad-supported [20, 24], while others are available for healthcare enterprise use only [23]. Most of these commercially available sites state the drug label data in their databases originated with the FDA, while others include data from the EMA and other countries, with one claiming to have data from 9 countries [23].

While we were unable to verify the functionality of the fee-based and private sites, the ability of the ad-supported sites to analyze the text of drug labels fell short in allowing us to easily find the drug labels that matched our query. For example, RxList.com and Drugs.com allow users to search the text of drug labels. But they do not allow for "exact match" searches and sometimes it is confusing why the labels are included in the search results. In addition, these sites do not allow us to easily compare the text of two drug labels to find the similar text.

1.3 Drug Label Explorer (DLE)

Since we were unable to find an existing solution that met what we consider essential functionalities, we built Drug Label Explorer (DLE). DLE is a web-based application that provides robust search functionality that supports a wide variety of queries, including data filtering and aggregation using several different attributes.

We have extracted, transformed, and loaded Drug Label data from both the FDA and EMA source repositories. Our website is set up to regularly load new data after it is published by these regulatory agencies. DLE provides the following functionalities:

- Search using MedDRA synonym terms
- Upload and share private drug labels
- Save and share queries
- Compare two labels
- See the version history of a label
- Export a drug label or comparison

DLE has a search capability that provides robust results by looking up query terms and their synonyms. This is achieved by combining the drug label data with data from MedDRA, a rich and standardized medical terminology repository.

As discussed previously, drug label changes are very frequent. Our DLE application provides functionality to help users track and compare different versions of a drug label and show the evolution of the drug label information over time. This feature is one of the distinguishing features of our DLE application and can help to inform decision-making for drug manufacturers and regulators as they try to leverage existing data to make future safety action determinations or drug repositioning proposals.

DLE includes the ability to upload custom drug label data for labels that are not yet approved and available publicly. This unique feature is useful to drug manufacturers who want to better formulate the wording of their unpublished drug labels by comparing their labels with those that are already in the system. This custom drug label uploading functionality is only available to registered users to make the data only available to them and to their collaborators who are also registered users and granted access by the originator of the data so registered users can share drug labels with others users on the platform.

Beside the features embedded in our DLE application, the database itself is openly available for anyone who wants to leverage it. The availability of such a well organized drug label database that is broken down into the various sections and subsections makes it attractive for researchers who want to access a specific section's corpus to train a machine learning model.

DLE is re-deployable, is backed by a MariaDB database and is open source and freely available on GitHub (https://github.com/DrugLabelExplorer/dle). Any individual, team, or institution can modify and improve upon the functionalities we provide to meet their desired use case, if not already available in our application. The open-source nature of our application, along with the easily accessible and regularly updated drug label database, makes it desirable for small teams of government regulators & policy makers, pharmaceutical & health care researchers, as well as academic institutions that operate with limited budgets and resources.

1.4 References

[1] FDA: Code of Federal Regulations, Title 21, Vol.4, Chapter 1, Part 201-Labeling.
Source:
https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=201
[2] European Commission: A Guideline on Summary of Product Characteristics (SmPC), September 2009.
Source: https://ec.europa.eu/health/system/files/2016-11/smpc_guideline_rev2_en_0.pdf
[3] Lester, Jean, et al. (2013). Evaluation of FDA safety-related drug label changes in 2010. Pharmacoepidemiology and Drug Safety, vol. 22.3, p302-305.
Source:
https://onlinelibrary-wiley-com.ezp-prod1.hul.harvard.edu/doi/pdfdirect/10.1002/pds.3395
[4] Fang, Hong, et al. (2016). FDA drug labeling: rich resources to facilitate precision medicine, drug safety, and regulatory science. Drug Discovery Today, vol 21.10, p1566-1570.
Source:
https://www-sciencedirect-com.ezp-prod1.hul.harvard.edu/science/article/pii/S1359644616302240
[5] Moore, Thomas J., Sonal S., and Curt D. F. (2012). The FDA and new safety warnings. Archives of Internal Medicine, vol. 172.1, p78-80.
Source:
https://jamanetwork-com.ezp-prod1.hul.harvard.edu/journals/jamainternalmedicine/fullarticle/1108624
[6] Dusetzina, Stacie B., et al. (2012). Impact of FDA drug risk communications on health care utilization and health behaviors: a systematic review. Medical Care, vol. 50.6, p466.
Source:
https://oce-ovid-com.ezp-prod1.hul.harvard.edu/article/00005650-201206000-00002/HTML
[7] Seminerio, M. J., and M. J. Ratain. (2013). Are drug labels static or dynamic? Clinical Pharmacology & Therapeutics, vol 94.3, p302-304.
Source:
https://ascpt-onlinelibrary-wiley-com.ezp-prod1.hul.harvard.edu/doi/full/10.1038/clpt.2013.109?sid=vendor%3Adatabase
[9] O. Nieminena, P. Kurkib, K. Nordstro. (2005). Differences in product information of biopharmaceuticals in the EU and the USA: implications for product development. European Journal of Pharmaceutics and Biopharmaceutics, vol. 60.3, p319–32

Source:
https://www-sciencedirect-com.ezp-prod1.hul.harvard.edu/science/article/pii/S0939641105000780
[10] Rodriguez, T., et al. (2021). Medical Error Reduction and Prevention. National Center for Biotechnology Information
Source: https://www.ncbi.nlm.nih.gov/books/NBK499956/
[11] Tariq, R., et al. (2021). Medication Dispensing Errors And Prevention. National Center for Biotechnology Information
Source: https://www.ncbi.nlm.nih.gov/books/NBK519065/
[12] Delgado, N., etl al. (2019). Fast and accurate medication identification. npj Digital Medicine, vol. 2.10
Source: https://www.nature.com/articles/s41746-019-0086-0#Sec6
[13] Jeetu, G., et al. (2010). Prescription Drug Labeling Medication Errors: A Big Deal for Pharmacists. Journal of Young Pharmacists, vol 2.1, p107-111
Source:
https://www.sciencedirect.com/science/article/abs/pii/S097514831021021X
[14] Davis, T. C., Federman, A. D., Bass, P. F., 3rd, Jackson, R. H., Middlebrooks, M., Parker, R. M., & Wolf, M. S. (2009). Improving Patient Understanding of Prescription Drug Label Instructions. Journal of General Internal Medicine, vol. 24.1, p57–62
Source: https://link.springer.com/article/10.1007/s11606-008-0833-4
[15] Shrank, W., Avorn, J., Rolon, C., & Shekelle, P. (2007). Effect of content and format of prescription drug labels on readability, understanding, and medication use: a systematic review. The Annals of pharmacotherapy, vol. 41.5, p783–801.
Source:
https://journals-sagepub-com.ezp-prod1.hul.harvard.edu/doi/pdf/10.1345/aph.1H582
[16] FDA Databases:
Source (Orange Book): https://www.accessdata.fda.gov/scripts/cder/ob/index.cfm and
Source (Drugs@FDA): https://www.accessdata.fda.gov/scripts/cder/daf/index.cfm
[17] NIH, DailyMed Database:
Source: https://dailymed.nlm.nih.gov/dailymed/index.cfm
[18] EMA, Medicines Database:
Source: https://www.ema.europa.eu/en/medicines/what-we-publish-medicines-when-0
[19] FDA, Drug Safety-related Labeling Changes (SrLC) Database:
Source: https://www.accessdata.fda.gov/scripts/cder/safetylabelingchanges/
[20] RxList Website: https://www.rxlist.com (a WebMD owned product)

[21] ReedTech Website: https://www.reedtech.com
[22] WizMed Website: https://wizmed.com
[23] Cerner Website: https://www.cerner.com/solutions/drug-database (an Oracle owned product)
[24] Drugs.com Website: https://www.drugs.com
[25] FDALabel: Full-Text Search of Drug Product Labeling:
Source:
https://www.fda.gov/science-research/bioinformatics-tools/fdalabel-full-text-search-drug-product-labeling#What%20is%20Included%20in%20Labeling
[26] Krist Shingjergji, Remzi Celebi, Jan Scholtes, Michel Dumontier Relation extraction from DailyMed structured product labels by optimally combining crowd, experts and machines, Journal of Biomedical informatics, Vol. 112, Oct 2021
Source: https://www.sciencedirect.com/science/article/pii/S1532046421002318
[27] Chen M.J., Vijay V., Shi Q., Liu Z.C., Fang H., and Tong W.D. "FDA-Approved Drug Labeling for the Study of Drug-Induced Liver Injury." Drug Discovery Today, vol. 16, p697-703
Source:
https://www.sciencedirect.com/science/article/abs/pii/S1359644611001668?via%3Dihub

2. System Design

2.1 Tech Stack

The project essentially uses a LAMP (Linux, Apache, MariaDB, Python) technology stack which ultimately revolves around a Python application being deployed on AWS. The backend of the application is written in Python utilizing the Django framework with the front-end served via Django templates, effectively reducing most of the application logic within one single framework. The web server used for serving the web requests is Apache, which was ultimately chosen because one of the project members has many years of experience with the library, though many other alternative web servers can be a drop-in replacement. The Python application is served via the mod_wsgi Apache plugin. Lastly, the database the project team decided on is MariaDB. MariaDB was chosen due to the availability of its ColumnStore engine for fast analytics across large datasets as well as its familiar SQL syntax. Ultimately the team did not use the ColumnStore engine; after testing its suitability for the project it was decided to use the default InnoDB engine in MariaDB instead.

2.2 Tool Suite

The team is using Github as a primary tool suite for the project. Github Projects was chosen as the primary planning software. This ultimately leads us to use Github as the hosted git Version Control System. Following this trend, the project uses Github Actions to orchestrate its CI pipeline. And Github Issues are used to track any bugs and action items that arise.

For testing, we are using the Django test harness which is an extension of Python's unittest module. The unit tests are executed using Github Actions on every pull request and on every merge into the main git branch. The results of the tests can be easily seen on Github with a green check mark indicating success and a red X indicating a test failure.

Utilizing all of Github's built-in tools reduces the amount of learning required with other existing 3rd party tools. Asynchronous communication is handled through Slack messages and email is used for coordinating meetings with Zoom conferencing for those who are not a part of the slack organization.

2.3 System Modules

The project requirements were broken up into modules to facilitate developers working on different parts of the application at the same time in a remote environment with team members in different time zones having different work schedules. The original technical requirements for this project, including estimates for when each feature were to be delivered are included in the Appendix. For the requirements, they are categorized into what roughly equates to Base Modules of the product.

- **Non-functional / DevOps**: We have a web application that is accessible on the internet that acts as a gateway to the features of the application. The web application supports encrypted traffic (https) and is able to be easily redeployed by an admin.

- **Data**: The website is backed by public data sources, cleaned, merged, and regularly updated. This data is served in a database that facilitates the queries created by the following features.

- **Users**: The website supports both anonymous and authenticated access, with certain features being restricted to users who have authenticated.

- **MyLabels**: When logged in, users can upload their Drug Labels and have them be queryable by the system. These uploaded Drug Labels are parsed and inserted into the Drug Label Explorer database. The user-uploaded Drug Labels are only accessible to the user who uploaded the Drug Label.

- **SearchForm**: One of the main features of the website is search form that allows Drug Labels to be queried. Capabilities for searching include: by drug brand name, by manufacturer, by label section, by agency and by generic drug name. This search form gives the user the ability to fine tune their results to get exactly what they need.

- **SearchResults**: The search results are displayed cleanly and the search terms are highlighted in the results. From this page the user is able to see blurbs from each search result with relevant keywords highlighted. From this page the user will also be able to click into a result to get more information, or select multiple results to compare them side by side.

- **SingleLabelView**: The user can switch into a detailed view of a single Drug Label. From this view the user will be able to see the entire content of the selected Drug Label.

- **CompareView**: The user can select 2 drug labels to view side by side, including separate drugs or different versions of the same drug. The view will clearly highlight areas that are similar and the differences.

2.4 Architectural Diagrams

Diagram showing the latest system architecture in AWS.

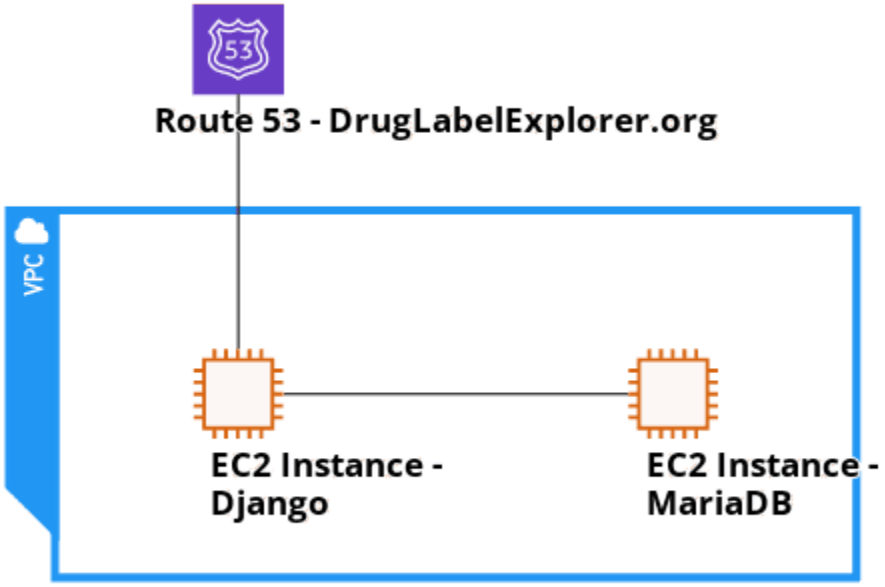
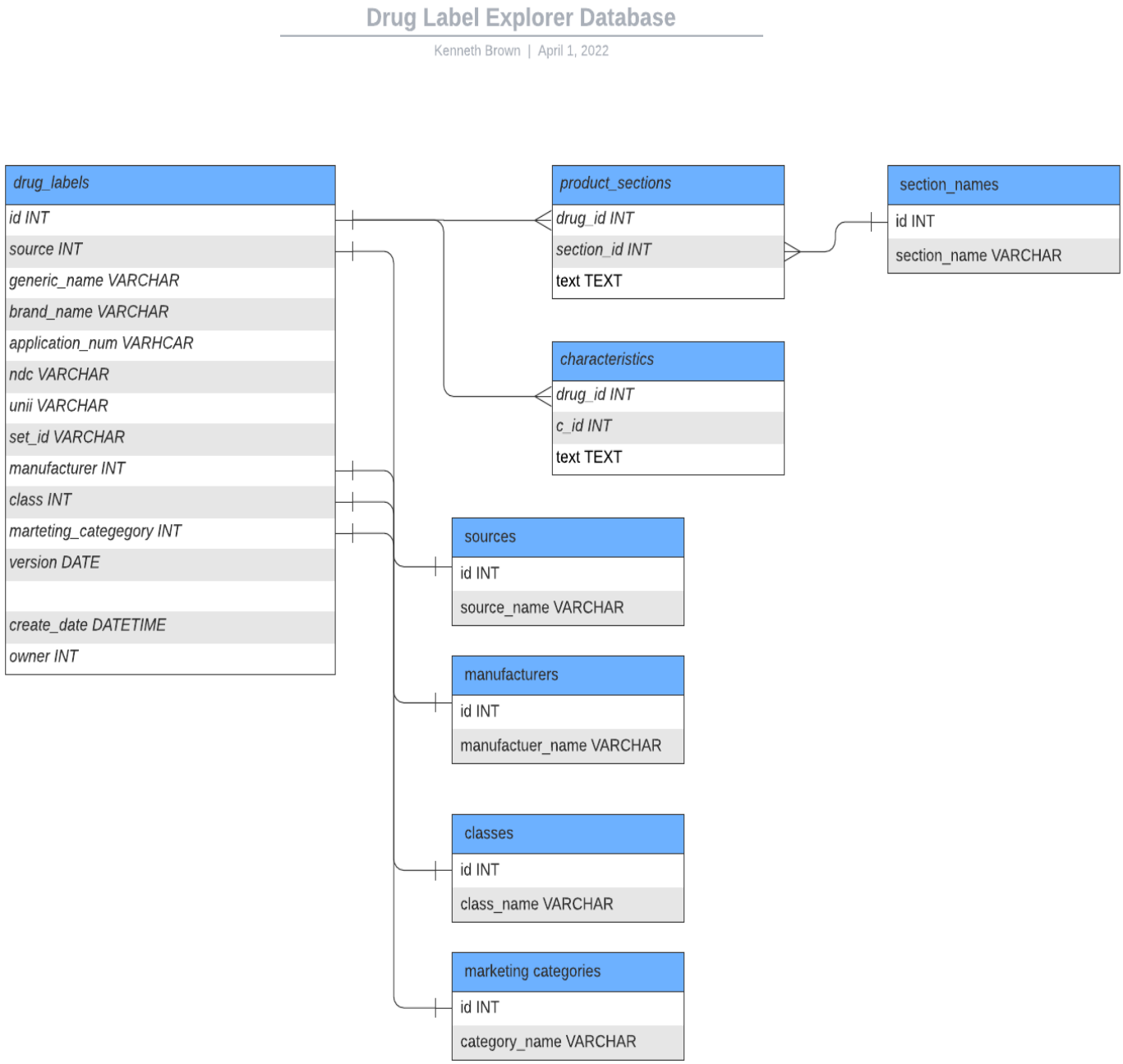


Diagram showing the latest class / database model for the DrugLabels.



3. Testing Results

For this project, our team conducted unit testing, performance testing, load testing, UI testing along with ad-hoc testing.

Unit testing is automatically performed on every pull request and on every code merge into the main git branch. Current unit test coverage is 63%. A report of the code coverage is included in the Appendix.

Performance testing was conducted periodically through development. The team was able to implement significant performance improvements, getting the average query time from 63 seconds per query to around 10 seconds per query. To facilitate performance testing a script was developed that can run the performance tests automatically with the command `python manage.py performance_tests` the test results are output to a CSV file and a png is created of the results. Some examples of the output are included in the Appendix.

There were some issues the team encountered with the database server in terms of handling a load of multiple queries at the same time. To ensure the reliability of the system, the team created a script that can perform load tests on the system. This is run with the command `python manage.py load_tests`. This helped the team ensure that the system can handle a reasonable load of queries concurrently.

Informal UI testing was performed with the Customer at the team meetings. The team implemented many of the suggestions that occurred at these back and forth sessions. In this way the team had an iterative approach to the User Interface implementation. The team provided the latest (uncompleted) iteration of each feature to the Customer as soon as it was ready. This allowed the Customer to give the team valuable feedback on the progress of the work, even before the features were completed.

Ad-hoc testing was also conducted as a part of the development process. Developers would – in addition to the unit testing – perform ad-hoc tests on the features under development. Developers would perform ad-hoc testing on the features assigned to them, but would also test the features assigned to the other developers during each sprint. The team had the goal to have at least one other developer help test the code before a feature was merged into the main repository branch.

4. Development Process and Lessons Learned

4.1 Meeting the Requirements

The team was able to complete most of the original specified requirements. The main requirements completed include parsing and loading of the data, being able to search the data, displaying the search results, being able to compare the drug labels and being able to upload drug labels that are only available to the individual. Some of the features that we were not able to implement due to time constraints include: being able to search via MedDRA synonyms, being able to save search queries, and being able to export the search results. The team worked with the customer to reprioritize the requirements on an ongoing basis through weekly meetings.

Holding weekly meetings with the customer where we displayed the current (work in progress) status of the project was invaluable in gathering feedback from the customer. This allowed us to focus our efforts on the areas of the project that provided the most value to the customer.

The team originally planned to meet weekly and have 1-week sprints. Partway through the project we added a meeting so we ended up having two meetings a week plus our customer meeting.

There were some changes to our original plans that our team had to deal with as the project progressed. One of the issues was the performance of the database. The team originally planned to use a MariaDB database with the ColumnStore engine as the database for the project. This turned out not to be the best choice after the implementation was not delivering the desired performance.

Another of the issues we encountered was loading of the data was more finicky than initially anticipated. We had in our model the concept of “sections” in the drug labels. But after working with the data and discussing with the customer, there was another level in the hierarchy, “subsections.” The team was able to make progress handling the subsections in the drug label data by working closely with the customer to refine the project requirements as the project progressed.

4.2 Estimates

Our initial time estimates were somewhat inaccurate. In pretty much all the cases, the amount of time spent working on a feature was about 2-3X that of the estimated time. This was mostly due to the time required to deal with “unforeseen” issues. So our initial estimates were somewhat of a “best case scenario” estimate. In the actual development process there were bugs that were introduced that added time to debug and fix. Also the amount of time iterating on a solution was not well captured by the estimates. For the features that we developed, they were improved over time so in some cases our estimates were “time estimates to deliver the first version” of a feature rather than “time estimates to deliver the final version” of a feature.

4.3 Risks

The Drug Label Explorer project enables users to quickly retrieve data on drug labels and their changes over time. This requires a significant amount of data aggregation across multiple data stores and regions to transform into an easily queryable dataset. Two identifiable risks are presented when dealing with data ingestion in this scenario, data mining across varied data stores that change with its region, and modeling the mined data in such a way that querying the data becomes a trivial task.

Effective data mining across multiple regions will enable the project to procure data and provide an effective strategy for data transformation. Originally, the project had requirements that the data be extracted from many different countries’ drug agencies. This presented a large risk because we would have to support parsing and translating an enumerable amount of languages and label formats. The project team was able to de-risk this requirement significantly by scoping the project to 2 drug agencies, FDA and EMA. With the data sources limited to just two agencies, the project team can more effectively mine data from the respective agency’s datastore without having to program the mining tools to take into account different languages and additional label formats.

Once data is extracted from the drug agencies datastore, it will need to enter a transformation process that will parse, clean, and store the data in a queryable format. The risk that comes with ETL on mined data from these datastores is that the parsing step will have to make tradeoffs between acquiring more data points or maintaining high accuracy within the extracted data. This risk is apparent when we compare the FDA XML files against the EMA PDF files. Both of these agencies provide data that are loosely structured and when parsing such data the aforementioned risk is apparent. The project team was able to reduce the risk and complexity of this ETL step by reducing the scope of the features on the dataset

when persisting the data. With the emphasis on high-accuracy and high-quality features on the dataset, the parsing and storing of the data is now more focused and simpler to execute.

In addition to the aforementioned risks, there are also the inherent risks with the limitation of web scraping tools available for working with PDFs. This risk is highlighted by the use of PDF templates that the implementation requires when expecting a PDF of a certain format to be parsed. Inevitably, there will be PDFs that don't align perfectly with the template and will reduce the accuracy of the parsed data. The project team has mitigated this risk by assuming that the accuracy levels of parsed data may not be 100%, and any errors will be logged and categorized internally to further improve the system. The team plans to review the actual results with the customer every week to achieve customer satisfaction within the timeframe of the project.

While we anticipated the risks in working with the PDF data from the EMA datasource, in reality parsing the XML data from the FDA datasource turned out to be more challenging. It turned out that the FDA XML data is structured in a less standardized way than the EMA PDF data, which made parsing the FDA data more complex.

One of the main risks that was unanticipated was the performance of the database. If we were not able to query the data in a reasonably timely manner, the whole project would be a bust. The team did a good amount of planning initially to come up with what we thought was a good solution for how to store and query the data, but testing showed that we needed to change course. As a result, we were able to deliver a project that has a reasonable query time. Unfortunately, this took more development cycles than anticipated.

Another unanticipated risk is if a team member gets sick, takes a vacation or is unable to work on the project for unforeseen personal reasons. This was not a major problem for our team. But there might have been one or more occasions where a developer's absence slowed down the project's progress a little bit.

4.4 Team Dynamic

The Drug Label Explorer Team has a shared goal of developing the best possible outcome to improve the solution for our client. To do so we agreed on utilizing open communication which will be primarily conducted through Slack to account for differing time zones and personal work patterns, and we utilize Google Drive and GitHub for deliverables, we have set

up a shared Google Drive and a GitHub Project to this end. Client meetings are recorded and shared throughout the project. All team members are encouraged to discuss issues and problems that may arise. DLE plans to take advantage of members' unique abilities to maintain efficiency while maintaining a collaborative approach throughout all sections of the project.

In terms of conflict resolution, DLE collectively prefers a minimal contact strategy meaning issues will be raised between conflicting parties and only be brought to the entire group and/or the teaching staff if conflicts cannot be resolved or compromised internally. This strategy aligns with our shared value of open communication. Group decisions made by DLE will have a hybrid of majority rule and guided by members with advanced expertise in a given topic (consensus decision-making).

DLE originally planned to meet twice per week: on Saturday (internally discuss varying relevant agendas) and on Friday (with the Customer to discuss relevant issues/agenda). We ended up adding a meeting on Tuesday to facilitate the project's progress.

The team initially planned to have 9 weekly Sprints, running Thursday to Thursday, showing the latest progress with the Customer on Friday. The goal is to have open communication with the Customer. Each week we sought and gained valuable feedback and discussed any issues as they arose to help facilitate the success of the project.

Having the weekly Sprints did not really help us as a team. Instead we integrated features into the main branch in an ad-hoc manner whenever they were ready.

The team initially decided that each team member would "self manage" meaning that we would create our own Tickets and update our progress on the Github Project Workboard. This did not work out as well as originally hoped as some team members would forget to perform these management tasks. Not having a designated "project manager" might have also hurt the team a little in that there was no one to "keep team members accountable" if they were following through with their tasks in the expected timeframe. Overall this wasn't really an issue for our team. But if this was a larger project or if there were concrete deliverables that were more time sensitive, this might have been a larger issue.

5. Appendix

5.1 Technical Requirements

For these requirements, Week 1 Starts on March 3rd, 2022, and the requirements are expected to be delivered on or before Milestone 2 (April 7th) or Milestone 3 (May 5th) as listed.

Deliverable 1:
There is a website is available on the public internet that allows people to run queries on drug labels
Category: Non-functional
Estimate (hours): 8
Start week: 1
Milestone delivery: 2

Deliverable 2:
Website is protected by industry standard TLS encryption
Category: Non-functional
Estimate (hours): 1
Start week: 1
Milestone delivery: 2

Deliverable 3:
Website can handle a small number of concurrent users - tens
Category: Non-functional
Estimate (hours): 0, because a small number of Users are handled automatically, and if we wanted to handle millions of users, that would take more time.
Start week: 1
Milestone delivery: 2

Deliverable 4:
Instructions provided to deploy / redeploy all System components on Amazon Web. AWS CloudFormation template(s) to assist with the deployment of System components Services (AWS)
Category: Non-functional
Estimate (hours): 5

Start week: 1
Milestone delivery: 3

Deliverable 5:
Database with security measures to protect the data stored in the database including encryption at rest and encryption in transit (via SSL)
Category: Non-functional
Estimate (hours): 3
Start week: 1
Milestone delivery: 2

Deliverable 6:
Database instance is configured to automatically run Snapshot backups daily, keeping a 30 day rolling window of backups
Category: Non-functional
Estimate (hours): 1
Start week: 6
Milestone delivery: 3

Deliverable 7:
Response times of the system are within reason
Category: Non-functional
Estimate (hours): 12
Start week: 6
Milestone delivery: 3

Deliverable 8:
System has access to FDA Drug Label data from DailyMed (SPL/XML), to include the latest versions for all approved prescription drug labels
Category: Data
Estimate (hours): 40
Start week: 1
Milestone delivery: 2

Deliverable 9:
System has access to FDA Drug Label data from DailyMed (SPL/XML), to include ALL historical versions for all prescription drug labels for the previous 3, 5, or 7 years (TBD)
Category: Data (History)
Estimate (hours): 20
Start week: 6
Milestone delivery: 3

Deliverable 10:
System automatically refreshes data from its data sources at specified cadence (daily, weekly, monthly)
Category: Data (Refresh)
Estimate (hours): 20
Start week: 6
Milestone delivery: 3

Deliverable 11:
System accesses data from EU data source (PDFs), to include the latest version for all prescription drug labels
Category: Data
Estimate (hours): 75
Start week: 3
Milestone delivery: 2

Deliverable 12:
Drug label data can be accessed using MedDRA terms
Category: Data
Estimate (hours): 35
Start week: 2
Milestone delivery: 3

Deliverable 13:
Data from all data sources is standardized using the Findable, Accessible, Interoperable, Reusable (FAIR) principles. This will be done with a uniform schema and search tools designed to directly interface with such.
Category: Data

Estimate (hours): 12
Start week: 2
Milestone delivery: 2

Deliverable 14:
Drug label search functionality is available to an unregistered / guest / null User
Category: Users
Estimate (hours): 0
Start week: 1
Milestone delivery: 2

Deliverable 15:
Basic user authentication including sign-up, sign-in, and password reset using email allows for additional features such as uploading labels, saving queries, etc.
Category: Users
Estimate (hours): 20
Start week: 1
Milestone delivery: 2

Deliverable 16:
Ability to upload labels conforming to a supported type: FDA/XML, EU/PDF; labels are only available to the single user (by default)
Category: MyLabels
Estimate (hours): 18
Start week: 4
Milestone delivery: 3

Deliverable 17:
Ability to share saved labels with other registered users; after selecting a label and choosing an email address, the system will send an email with a link to a page in the system that shows the label
Category: MyLabels
Estimate (hours): 5
Start week: 6
Milestone delivery: 3

Deliverable 18:
Sharing user-uploaded drug label, grants access to the registered user with the recipients email address
Category: MyLabels
Estimate (hours): 3
Start week: 6
Milestone delivery: 3

Deliverable 19:
User-uploaded drug labels show up in the user's search results along with other drug labels; only the user who uploaded the label or other users with whom the label was shared have access
Category: MyLabels
Estimate (hours): 4
Start week: 6
Milestone delivery: 3

Deliverable 20:
Ability to Save searches
Category: MyQueries
Estimate (hours): 6
Start week: 7
Milestone delivery: 3

Deliverable 21:
Main page of the application has a SearchForm area that includes the functionality for searching the Drug Labels. In general this can include drop-downs, checkboxes, text fields, etc.
Category: SearchForm
Estimate (hours): 35
Start week: 1
Milestone delivery: 2

Deliverable 22:
Ability to limit searches to FDA Drug Labels, EU Drug Labels or both
Category: SearchForm

Estimate (hours): 2
Start week: 1
Milestone delivery: 2

Deliverable 23:
Ability to Search by Product (Generic and/or Brand Name)
Category: SearchForm
Estimate (hours): 3
Start week: 1
Milestone delivery: 2

Deliverable 24:
Ability to Search by Application number, DEA schedule, NDC, UNI code, SET ID
Category: SearchForm
Estimate (hours): 3
Start week: 4
Milestone delivery: 2

Deliverable 25:
Ability to Search by Product Characteristics (color, imprint, shape, size, scoring, etc)
Category: SearchForm
Estimate (hours): 3
Start week: 4
Milestone delivery: 2

Deliverable 26:
Ability to Search by drug Marketer
Category: SearchForm
Estimate (hours): 3
Start week: 4
Milestone delivery: 2

Deliverable 27:
Ability to Search by Label Section
Category: SearchForm
Estimate (hours): 3
Start week: 4

Milestone delivery: 2

Deliverable 28:
Ability to Search by MedDRA terms
Category: SearchForm
Estimate (hours): 3
Start week: 4
Milestone delivery: 2

Deliverable 29:
Ability to perform wildcard search on drug label data when searching within drug label categories
Category: SearchForm
Estimate (hours): 4
Start week: 6
Milestone delivery: 3

Deliverable 30:
Ability to perform proximity search — a user should be allowed to search for drug label terms that are within a specified distance from each other (e.g. number of words apart, within the same paragraph, or within the same section)
Category: SearchForm
Estimate (hours): 15
Start week: 7
Milestone delivery: 3

Deliverable 31:
Ability to Filter Search Results - Pharmacologic class
Category: SearchForm
Estimate (hours): 1
Start week: 5
Milestone delivery: 2

Deliverable 32:
Ability to Filter Search Results - marketing categories
Category: SearchForm

Estimate (hours): 1
Start week: 5
Milestone delivery: 2

Deliverable 33:
User has some ability to adjust what data columns are displayed from the query results
Category: SearchForm-Results
Estimate (hours): 8
Start week: 3
Milestone delivery: 2

Deliverable 34:
Ability to Group Search Results by Generic Name
Category: SearchForm-Results
Estimate (hours): 2
Start week: 3
Milestone delivery: 2

Deliverable 35:
Ability to Group Search Results by Manufacturer
Category: SearchForm-Results
Estimate (hours): 1
Start week: 3
Milestone delivery: 2

Deliverable 36:
Ability to Group Search Results by Country
Category: SearchForm-Results
Estimate (hours): 1
Start week: 3
Milestone delivery: 2

Deliverable 37:
Ability to Group Search Results by Marketing Category (i.e. Application Type)
Category: SearchForm-Results
Estimate (hours): 1
Start week: 3

Milestone delivery: 2

Deliverable 38:
Ability to specify “latest version” or “all versions” for the drug labels in the search results. Drug label versions are derived from the date the document was last updated.
Category: SearchForm
Estimate (hours): 2
Start week: 2
Milestone delivery: 2

Deliverable 39:
Ability to have multiple search criteria. Ability to apply up to 5 search criteria with AND operators.
Category: SearchForm
Estimate (hours): 5
Start week: 2
Milestone delivery: 2

Deliverable 40:
After the search is executed, the search results are displayed to the user. The search results view should display a list of the matching drug labels. The search results may be paginated when they exceed a specified number of drug labels.
Category: SearchResults
Estimate (hours): 50
Start week: 2
Milestone delivery: 2

Deliverable 41:
The Search query parameters used in the search are highlighted in the SearchResults when present
Category: SearchResults
Estimate (hours): 3
Start week: 6
Milestone delivery: 3

Deliverable 42:

A details page for the drug label is shown after the user clicks on an item from the search results.
Category: SingleLabelView
Estimate (hours): 20
Start week: 4
Milestone delivery: 3

Deliverable 43:
The Search query parameters used in the search are highlighted in the SingleLabelView
Category: SingleLabelView
Estimate (hours): 3
Start week: 6
Milestone delivery: 3

Deliverable 44:
In the SearchResults there is the ability to select two labels. After selecting two labels, the user can then compare the labels.
Category: SearchResults - Compare
Estimate (hours): 4
Start week: 6
Milestone delivery: 3

Deliverable 45:
Side-by-Side Comparison with "Track Changes" View (Two labels). As a user scrolls through the page, both sides of the view should be in sync.
Category: Compare
Estimate (hours): 38
Start week: 3
Milestone delivery: 3

Deliverable 46:
Search results are automatically highlighted in the side by side comparison view of the drug labels
Category: Compare
Estimate (hours): 2
Start week: 5
Milestone delivery: 3

Deliverable 47:
Ability to navigate to the VersionHistoryView from the SearchResults view
Category: SearchResults - VersionHistory
Estimate (hours): 2
Start week: 4
Milestone delivery: 3

Deliverable 48:
Ability to navigate to the VersionHistoryView from the SingleLabelView
Category: SingleLabelView - VersionHistory
Estimate (hours): 2
Start week: 4
Milestone delivery: 3

Deliverable 49:
A Version History View page is displayed showing changes to a drug label over time
Category: VersionHistory
Estimate (hours): 32
Start week: 3
Milestone delivery: 3

Deliverable 50:
Search results automatically highlighted in the version history page
Category: VersionHistory
Estimate (hours): 2
Start week: 6
Milestone delivery: 3

Deliverable 51:
Ability to export selected columns from multiple labels from the SearchResults
Category: Export
Estimate (hours): 3
Start week: 8
Milestone delivery: 3

Deliverable 52:
Ability to export Label Comparison to HTML
Category: Export
Estimate (hours): 2
Start week: 8
Milestone delivery: 3

Deliverable 53:
Ability to export the Version History View to HTML
Category: Export
Estimate (hours): 2
Start week: 8
Milestone delivery: 3

Deliverable 54:
All export HTML pages include highlighting of the search parameters, when present
Category: Export
Estimate (hours): 1
Start week: 8
Milestone delivery: 3

5.2 Wireframes

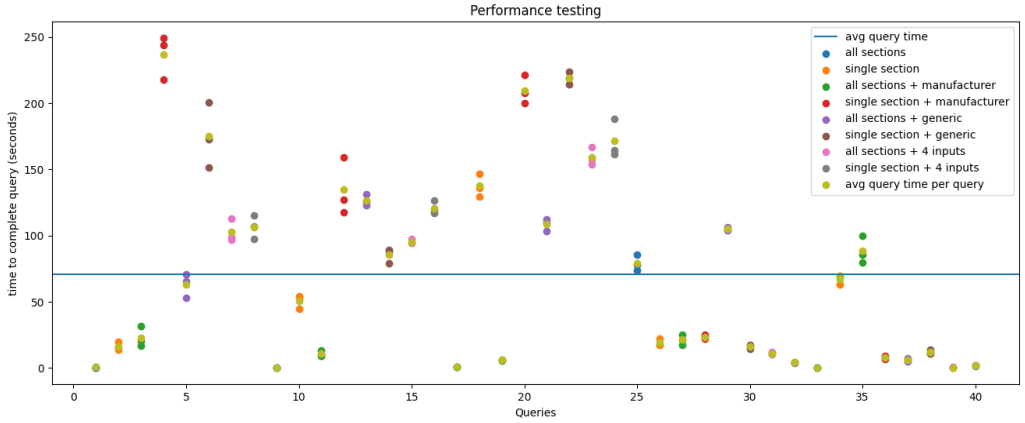
Wireframes from our client, David Edelen

5.3 Unit Test Code Coverage Report

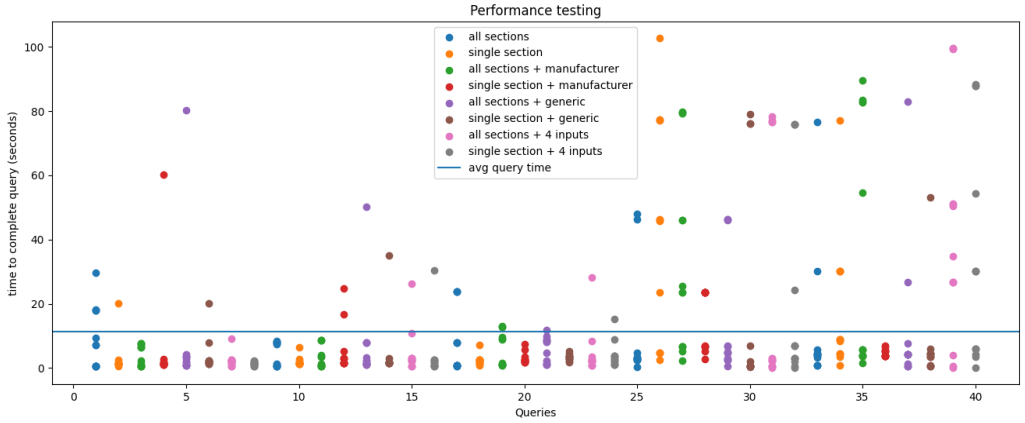
Name	Stmts	Miss	Cover
compare/apps.py	4	0	100%
compare/models.py	2	0	100%
compare/tests.py	1	0	100%
compare/urls.py	4	0	100%
compare/util.py	87	79	9%
compare/views.py	130	120	8%
data/apps.py	4	0	100%
data/constants.py	2	0	100%
data/management/commands/load_ema_data.py	185	44	76%
data/management/commands/load_fda_data.py	235	79	66%
data/management/commands/update_latest_drug_labels.py	27	2	93%
data/models.py	21	0	100%
data/tests.py	62	0	100%
data/urls.py	4	0	100%
data/views.py	22	16	27%
dle/settings.py	29	1	97%
dle/urls.py	7	1	86%
manage.py	12	2	83%
search/apps.py	4	0	100%
search/models.py	18	2	89%
search/search_constants.py	2	0	100%
search/services.py	84	36	57%
search/tests.py	23	0	100%
search/urls.py	3	0	100%
search/views.py	25	16	36%
users/apps.py	4	0	100%
users/forms.py	10	0	100%
users/models.py	13	1	92%
users/tests.py	58	2	97%
users/urls.py	4	0	100%
users/views.py	80	35	56%
TOTAL	1190	436	63%

5.4 Performance Test Results

Initial performance testing indicated an average query time of 63 seconds per query.



Updated hardware in conjunction with query improvements led to an average query time of around 10 seconds per query.



After additional improvements the team was able to get this down to 2.7 seconds average query time per our performance test benchmark tool.