Montana Healthcare Transparency

By
Mason Williamson
Philip Gehde
Michael Roduin
Alyse Mize
Emma Haines
Luke Armijo

Submitted in Fulfillment of Requirements
for the Degree of Bachelor of Arts in Computer Science
Gainforte School of Computing, Montana State University

Senior Team Portfolio

ESOF423 Software Engineering Applications
Spring 2022
Daniel DeFrance

Program

Site Link:
https://main--cranky-banach-68238c.netlify.app/
Repository Link:
https://github.com/423s22/G1

Teamwork

With six members working on the overall project, we split into two groups of three. Team A was responsible for data gathering and consisted of team members 2 and 3. Team B was responsible for designing the website and its components. This was done primarily by members 5 and 6. Team C was in charge of building the website and consisted of members 1, 4 & 5.

Team Member 1 - Philip:

Philip was largely responsible for building the CRUD aspect of the application, setting up and populating the database, writing schemas / data models for our API, and building the majority of the front-end of the application. Introducing Tailwind late in the process, Philip built most of the application from scratch, because of the need for a clean code base for the purpose of maintainability. He was assisted in this process largely by Member 5, Luke, and partially by Member 4, Alyse. In terms of building the website, everyone contributed, but because of the learning curve involved, probably about 80% of what we ended up using was built by Philip. However, overall, I believe that in terms of time invested all members contributed a similar amount. For Team A, both members contributed, with Member 2, Michael, providing the majority of code. 25% of time spent on this project was done by Philip.

Team Member 2 - Michael:

Michael developed the data mining and collection aspect of the project in its entirety along with data management and generation of Json files used to upload into the database and any error catching systems on the data collection side. Machael's contributions made up just over 15% of the github repository code in python files. Also with this, Michael managed finding and solving github and personnel related errors depending on the necessity. In all, Michael contributed an estimated 25% of the project time.

Team Member 3 - Mason:

Mason contributed to the back-end work, providing Michael with the hospital chargemaster information necessary to create the web-scraper. This task was necessary for Micahel to spend less time searching for the necessary data, and more time doing meaningful tasks, like scraping the data. Mason also operated as the Sprint Manager, facilitating most meetings and keeping most scrum artifacts in order. This role was entirely necessary to ensure productivity and that deadlines were being met. Duties included frequently and thoroughly updating the Sprint Backlogs and Burndown Chart, taking notes from any meetings, and ensuring a positive and fruitful work environment. In total, Mason contributed about 10% of the overall percentage of time on this project.

Team Member 4 - Alyse:

Alyse's main contribution was front end design. Working alongside Luke and Philip she utilized the designs created by Emma and converted them into a functioning website. First learning how React worked Alyse would slowly incorporate the Figma prototype into Json files using Css for the styling. As the website became more complicated evolving from the Beta release to using a database to make everything more efficient Alyse would then pivot to learning NextJs. Overall Alyse's contribution was styling the website to look as close to the prototype designs as possible while also trying to make everything more effective.

While there have been bumps in the project because of a lack of communication at one point overall Alyse has provided a good body of work and has helped where she can. Because of the teamwork with Philip and Luke the product has really benefited with each teammate helping each other. Because of this Alyse has spent a total of 10% on this project.

Team Member 5 - Luke:

Luke contributed around 15% of total time to help build and assist the website and its design interface.

Team Member 6 - Emma:

Emma was in charge of the entirety of the UX/UI design material. Team member 5, Luke, also began working on it during the beginning which proved helpful to get a direction to go in, but ultimately, Luke moved on to transfer the design to code. After that, Emma essentially took over ownership over the UX/UI. They roles and tasks associated with doing so included: creating personas, designing paper wireframes, moving them to digital wireframes, creating lo-fi mockups, then lo-fi prototypes, creating hi-fi mockups, and hi-fi prototypes, creating drawings for the design, and conducting a usability study. Overall, Emma made a large contribution to this project because they created how the whole thing looks and how it should work. They contributed to around 15% of the project.

Design Pattern

The software design pattern we used the most was the module pattern. Due to the large number of people we had compared to other projects we had previously experienced we focused on making each small piece of our program extremely modular isolating functions and data from each other. We used this pattern to increase the ease of understanding an expanding program as we developed and to make it as easy as possible to identify and fix bugs in the program with as little risk of affecting other parts of the program as possible. This also allowed us to repeatedly task group members to sets of small and easily quantifiable tasks to help us track progress.

This design pattern was mostly implemented through the separation of files and functions based on what they did and what they interacted with but each individual piece of processing done in the project was isolated to these module pieces. This can be seen more in the UML diagram section of this write up but for that case specifically each action listed as done by the program was isolated into a function or set of functions called then by a larger part of the program.

Technical Writing

The world of Healthcare is complicated. If a patient at a hospital or UrgentCare is seeking medical treatment, chances are that they will go into the procedure not knowing what it will cost them. Even if you ask a nurse or a doctor, most times they will hardly know either. For those who are not blessed with good insurance, they will likely have to pay out-of-pocket for many of these things. How are these patients supposed to know what they might be paying when the time comes?

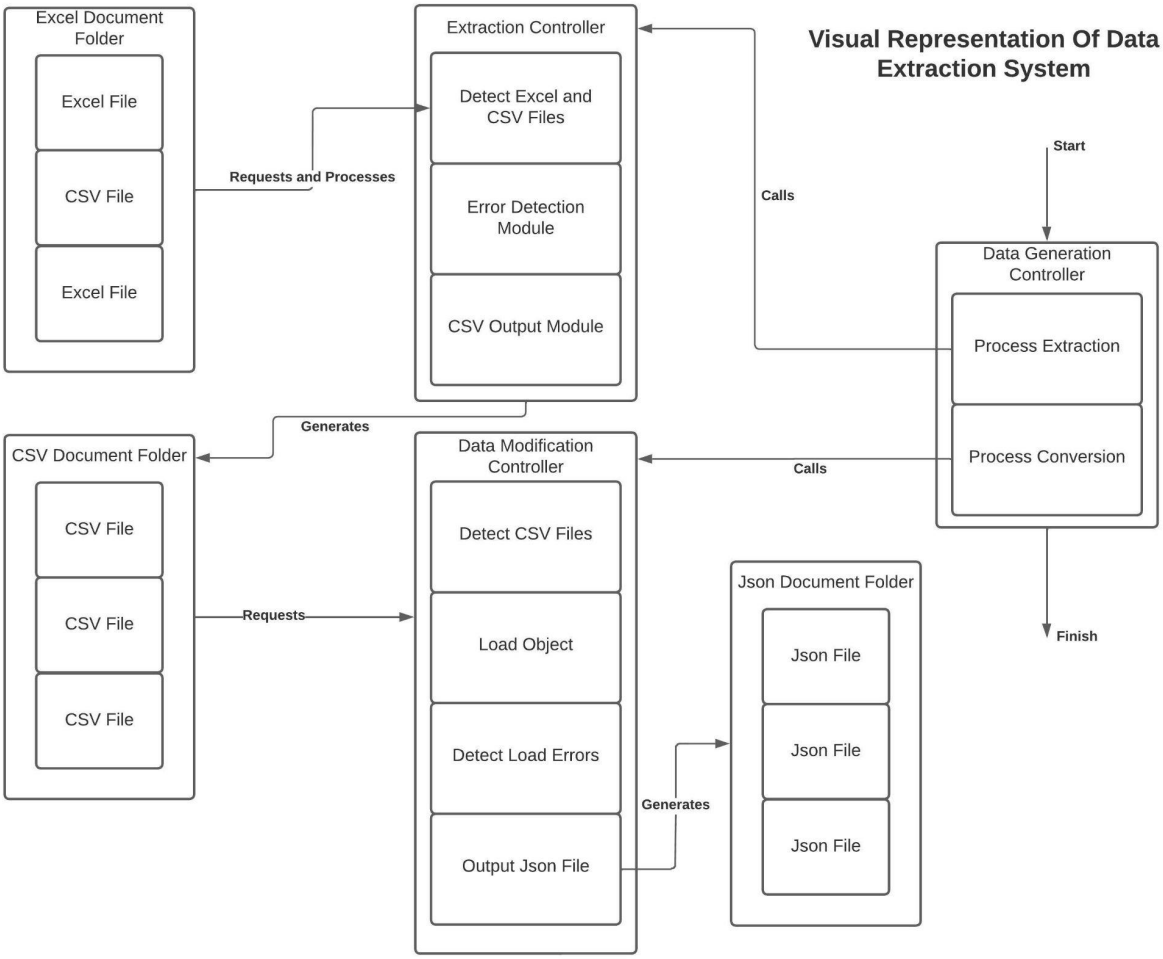
This is where our group comes in. Our client is Dr. Mike DiCello - a certified and practicing Orthodontist located in Bozeman and also an active member of the Montana Board of Medicine. For our capstone assignment, Dr. DiCello asked us to create a software that provides medical transparency information to a user seeking a medical treatment. The specifications were to create a user-friendly environment that when a user inputs a general medical procedure, the software would cross-compare between different hospitals for each procedure and show you all of the available prices. The goal of this software would be to provide more insight to wary hospital patients who want a better understanding of how much a procedure costs at any given hospital near them. They can cross-compare procedure prices between different hospitals and make an educated decision on where they would like to get their procedure done. This kind of knowledge is something that you would have to dig heavily for. Another goal of ours is to make it so that this information is far more easily accessible to the everyday person.

We decided on a proof-of-concept application, rather than a full feature-inclusive website for two main reasons: One was the lack of time. We only had so much time to complete a project that could seem to rival price transparency totems such as GoodRx and the like. Second, we did not have the human nor technological resources necessary to go that far. With not everyone having enough coding experience to create what we desired, we could make ends meet by creating something that proves that it can be done and gives a roadmap as to how a real price-transparency software could look.

We set out to scrape hospital chargemaster data from just about every hospital in the state of Montana. Although some hospitals omitted from displaying their chargemasters publicly, we were able to collect 52 out of the 69 hospitals located in Montana. After careful and tedious work, we constructed a live website that allows for users to search for a general procedure and receive price data for all related procedures at various hospitals across Montana. The following link leads to the website we programmed to accomplish the goals outlined for us. Though not built to function like an official user interface, it serves its purpose by providing a proof-of-concept layout of the idea and basic functionality that will help future developers create something that will make a huge impact in the medical world.

UML

For the UML aspect of the projects due to the size and spread of the project across multiple languages in multiple purposes we did not feel the necessity of a standard UML diagram and instead developed sets of program process diagrams to better show the very function based nature of our code. As such we have decided to highlight the procedure diagram for the data mining section of the project for collecting and processing data related to the hospital files.



As you can see we have two major pieces to the data mining section of our program separated into the extraction process and the conversion process as seen on the far right of the above diagram. These processes are further subdivided into the major functions of each process and the order that they are run in as shown in the central two boxes in the diagram. Other than this are just visual representations of the file storage and file types expected in those folders specifically split into an initial unstandardized set in the top left holding initial files as is, a standardized and organized file set in the bottom left for storage and hand off between devices and finally a server located Json document folder to be stored on the server only.

Design Trade-offs

The biggest design trade-off we had to make was the decision between computational efficiency and overall features. Partway through the project, it became clear that we didn't have enough people with quality coding skills to be able to check all of our boxes for a final release. Once our final release was done, we had little to no time to go back in and make the quality tweaks that we should need. We had to decide whether the external quality of our work was more important than the internal quality of our work.

In terms of computational efficiency, our algorithms would need to be checked over multiple times by multiple different group members to assure that it was taking up the least amount of memory while also running in the shortest time possible. This would mainly be in the data scraping portion of our system and the API for our website.

In terms of adding features to our project, with this option, we would be able to dedicate more time and resources to making the website more functional. We would be able to make the design of our website look more like the prototypes we created on Figma, as well as increase the functionality of our searching mechanism.

Overall, I believe that sacrificing computational efficiency for feature additions was the right move. Most users will only see the external portion of our software, and will not bother to scan the codebase for flaws. When it comes to future developers, I hope that they will be able to overlook any flaws in our code and focus on the vision of this project, providing a proof-of-concept software of hospital data prices displayed visually and clearly for users to be more aware and educated.

Software Development Life Cycle Model

In this software engineering team, we used Scrum methodologies to develop our program. Scrum uses iterative and incremental practices to increase productivity and embrace an adaptive system development process. Every two weeks, a new sprint would begin, where our group would track out our goals and assign tasks to be completed by the end of those two weeks. At every class period, every Monday, Wednesday and Friday, we would sit down and have a scrum meeting. The purpose of those meetings was to communicate about the development progress and tie up any loose ends. Common questions to be asked would be “What are you working on right now”, “What have you worked on since our last meeting”, and “Are there any roadblocks or places you might need help”? Also included in scrum methodologies are scrum artifacts. Examples we used in this project were the burndown chart, product backlog, and sprint backlogs for each occurring sprint. Each of us took a few minutes out of the working day to update each of these artifacts to keep track of our hours and progress.

Our scrum master for this project was Team Member 1, Mason, who dedicated himself to enacting scrum values and practices. The scrum artifacts and team meetings were all facilitated and monitored by Team Member 1 and, on top of dedicating themselves to the enactment of these scrum values, also contributed to the working development of the project.

Scrum methodologies helped our group considerably by not only keeping us in touch with one another but organizing our working structure to enable us to work on one aspect of the project at a time. The sprint backlog and product backlog allowed us to input our current and ongoing tasks to ensure that there were no duplicate tasks being worked on, and that everything was all operating on one branch. The burndown chart helped us to track our hours and ensure that our working time was being used diligently and effectively. Our sprint meetings served to help us communicate with each other on a more personal level and keep teamwork to the highest standard. Overall, enacting scrum methodologies into our project was a fantastic idea, and the group seemed to benefit greatly from its use.